RICE UNIVERSITY

Implementing Energy Parsimonious Circuits through Inexact Designs

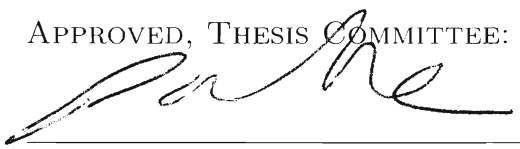
by

Avinash Lingamneni

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

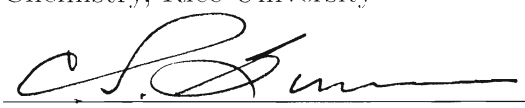
Approved, Thesis Committee:



Krishna V Palem, Chair
Kenneth and Audrey Kennedy Professor of Computing, Rice University

Behnaam Aazhang
J.S. Abercrombie Professor of ECE, Rice University

James M. Tour
T. T. and W. F. Chao Professor of Chemistry, Rice University



C. Sidney Burrus
Maxfield and Oshman Professor Emeritus of Engineering, Rice University

Christian Piguet
Scientific Coordinator, Integrated and Wireless Systems Division, CSEM SA, Switzerland

Houston, Texas
April, 2011

ABSTRACT

Implementing Energy Parsimonious Circuits through Inexact Designs

by

Avinash Lingamneni

Inexact Circuits or circuits in which accuracy of the output can be traded for cost (energy, delay and/or area) savings, have been receiving increasing attention of late due to invariable inaccuracies in nanometer-scale circuits and a concomitant growing desire for ultra low energy embedded systems. Most of the previous approaches to realize inexact circuits relied on scaling of circuit-level operational parameters (such as supply voltage) to achieve the cost and accuracy tradeoffs, and suffered from serious drawbacks of significant implementation overheads that drastically reduced the gains. In this thesis, two novel architecture-level approaches called *Probabilisitc Pruning* and *Probabilistic Logic Minimization* are proposed to realize inexact circuits with *zero overhead.* Extensive simulations on various architectures of datapath elements and a prototype chip fabrication demonstrate that normalized gains as large as 2X-9.5X in Energy-Delay-Area product can be obtained for relative error as low as $10^{-6}\%$ – 1% compared to corresponding conventional correct designs.

Acknowledgments

Please consider this an humble attempt to acknowledge the contributions of all the people who helped me to shape this thesis into its present form and accept my sincere apologies in case I failed to acknowledge the contribution of anyone (blame it on the undue stress of timely thesis completion! :-)).

First of all, I would like to convey my deepest gratitude to my adviser Prof. Krishna Palem for his invaluable support and encouragement throughout my graduate study. His exemplary vision and insights were the foundation for this thesis and I consider myself extremely previleged to have had an exposure to diverse and exciting collaborative environment established by him in many prestigious institutions.

I am extremely grateful to Prof. Christian Piguet and Prof. Christian Enz of CSEM SA, whose encouragement and insights helped mould the physical implementation aspects of this thesis into its current shape. I consider myself very fortunate to have access to industrial standard design tools and framework during my internship at CSEM that helped expedite the validation of my ideas conclusively in physical hardware, typically not feasible at most academic institutions. I would, in particular, like to acknowledge and thank Dr. Jean-Luc Nagel, Marc Morgan and Pierre-Alain Beuchat for their continuous help and support to my work at CSEM.

I would like to express my sincere gratitude to Prof. Chris Bronk for his many insightful discussions and interesting perspectives leading to my first "non-engineering" publication on carbon footprint of the ICT sector (outlined in Chapter 2). I would like to thank my other committee members Prof. James Tour, Prof. Sidney Burrus and Prof. Behnaam Aazhang for taking valuable time out of their busy schedules to help evaluate and provide feedback on my work in an effort to improve it.

Special thanks to my colleague and good friend, Kirthi Krishna Muntimadugu, for his continuing support, friendship and successful collaboration over the past decade. I am thankful to Dr. Lakshmi Chakrapani for his motivation, counseling and persistent encouragement which helped me adjust to a new and highly competitive working environment at Rice. My heartfelt thanks to Aparna Raju Sagi and Sravani Gullapalli for their immense help and support in taking care of all bureaucratic requirements needed for ensuring a successful thesis completion in my absence from Rice.

Lastly, but most importantly, I am eternally indebted to my parents, Kishore Babu Lingamneni and Sravanthi Lingamneni and my sister, Aasritha Lingamneni for their unconditional love, encouragement, sacrifices and support throughout my life.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Thesis Statement

The main focus of this thesis is understanding and analyzing the benefits of trading off accuracy for significant gains in cost (energy, delay and/or area) of electronic circuits through two novel and significantly more cost effective architecture-level approaches called *Probabilistic Pruning* and *Probabilistic Logic Minimization* guided by the overarching philosophy of *value of information based designs* to realize *inexact* circuits. Extensive experimental results, including fabrication of a prototype chip, have been performed to demonstrate and validate the effectiveness and gains that could be achieved by using the proposed techniques.

1.2 A Brief Summary of Important Contributions

The main contributions of this thesis are summarized below:

- A detailed and comprehensive evaluation of the increasing carbon footprint of the Information and Communication Technology (ICT) sector highlighting the increasing need for a radical shift from the present day (electronic) design methodology focused on incremental savings to a methodology which has a potential to achieve orders of magnitude higher savings.

- A comprehensive retrospect of existing inexact circuit design techniques taking

advantage of the principle of trading accuracy for cost (either energy, area or delay) savings, solely using circuit-level (voltage) overscaling techniques while highlighting their significant drawbacks.

- Two novel *zero overhead* architecture-level approaches called *Probabilistic Pruning* and *Probabilistic Logic Minimization* are proposed to realize inexact circuits and are shown to overcome all of the drawbacks of the existing design techniques while offering significantly more savings than any of the conventional approaches.

- A novel logic synthesis based CAD framework incorporating the proposed techniques to design inexact circuits achieving a faster time to design and fabricate than a full-custom design flow needed for most of the previous works in literature.

- Extensive experimental results including fabrication of a test chip, conclusively demonstrating and validating the potential savings in energy, delay and area obtained by the proposed techniques are described. In the context of various datapath elements such as adders and multipliers, savings as large as 2X-9.5X in Energy-Delay-Area product with corresponding relative error percentage as low as 10^{-6}%-1% have been achieved.

1.3 Organization of the thesis

This thesis is organized into eleven chapters. A brief summary of the each chapter is presented below:

- **CHAPTER I. INTRODUCTION:** This chapter gives a brief overview of *inexact* computing based design methodology. Also, the thesis statement and a summary of important contributions of this thesis are provided.

- **CHAPTER II. ENERGY SUSTAINABILITY OF INFORMATION AND COMMUNICATION TECHNOLOGIES (ICT):** In this chapter, the energy sustainability of the ICT sector is analyzed in detail and a need for a shift in the conventional (electronic) design methodology focussed on incremental savings to a design methodology capable of achieving orders of magnitude savings is established owing to the already increasing carbon footprint of the ICT sector. The contents of this chapter have been published in [1].

- **CHAPTER III. PARSIMONIOUS DESIGN - LIVING WITH LESS:** In this chapter, we establish the need and benefits of parsimonious design in a world dominated by error tolerant/resilient applications, in particular the domains of embedded, multimedia and DSP applications, where the end system consuming the output (for example, the human sensory system) could tolerate varying amounts of error in its constituent building blocks. In other words, these applications can synthesize accurate (or sufficient) information even from inaccurate computations. It is shown through illustrative examples, the benefits of parsimonious design approaches in the case of image processing algorithm along with the results of such erroneous outputs of varying magnitudes. Also, a detailed overview of the previously proposed (physical) implementations of the inexact circuits is discussed.

- **CHAPTER IV. UNDERSTANDING THE DESIGN ABSTRACTION LEVELS FOR INEXACT DESIGNS:** In this chapter, the three levels of design abstraction levels at which design optimizations can be done to realize the *optimal* (inexact) design implementations are discussed. It is shown that all of the previous work was focussed at the circuit level using scaling of op-

erational parameters such as supply voltage (V_{dd}) or frequency constrained by the algorithm's error tolerance/resilience at the algorithm level. The significant drawbacks of such an approach are highlighted and the need for innovations at the architecture level is established which will form the focus of rest of the thesis.

- **CHAPTER V. PROPOSED APPROACH I - *PROBABILISTIC PRUNING*:** In this chapter, the first architectural redesign technique called *Probabilistic Pruning* is proposed along with a detailed mathematical foundations and algorithm. The contents of this chapter have been published in [2].

- **CHAPTER VI. PROPOSED APPROACH II - *PROBABILISTIC LOGIC MINIMIZATION*:** In this chapter, the second architectural redesign technique called *Probabilistic Logic Minimization* is proposed along with a rigorous mathematical analysis and algorithm. The contents of this chapter are currently under review at [3].

- **CHAPTER VII. ANALYSIS OF GAINS FROM PROPOSED TECHNIQUES:** This chapter provides an analytic reasoning behind the gains that would be obtained by the proposed architectural redesign techniques. A detailed explanation of the effects on the circuit characteristics such as energy, delay and area by the proposed techniques is given.

- **CHAPTER VIII. APPLICATION OF PROPOSED TECHNIQUES TO DATAPATH ELEMENTS:** In this chapter, the application of the proposed techniques to design critical datapath elements such as arithmetic adders and multipliers is done. A brief overview of the various architectures of the datapath elements, widely studied over the last few decades, is also presented.

- **CHAPTER IX . EXPERIMENTAL RESULTS THROUGH SIMULATIONS:** In this chapter, the results of extensive simulations in a novel logic synthesis based CAD framework are presented. The simulations are performed over a wide variety of technology libraries (TSMC 65nm, IBM 90nm and TSMC 180nm) with varying synthesis constraints targeting both low power and high frequency synthesis.

- **CHAPTER X. PHYSICAL REALIZATION AND VALIDATION:** In this chapter, a physical (CMOS) realization and validation in TSMC 180nm(Low power) technology process of one of the proposed technique (probabilistic pruning) is presented. The framework used for the chip fabrication and testing is presented along with the results from the prototype chip.

- **CHAPTER XI. CONCLUSION AND FUTURE DIRECTIONS:** This chapter provides the conclusions of the thesis along with directions for future research possibilities building upon the work done in this thesis.

Chapter 2

Energy Sustainability of Information and Communication Technologies (ICT)

Nearly three decades ago, some people quipped that trees caused more pollution than automobiles. What they are likely implying was that the emissions of carbon dioxide (CO_2) by trees and other flora exceeds that by the human related emissions, missing the fact that plants and bodies of water also break down CO_2 in a relatively balanced cyclical activity. It is infact this human produced CO_2, from the production of concrete, running of internal combustion engines or generation of electricity from fossil fuels, for example, that is not removed from the global atmospheric stores of the gas (along with other climate-changing greenhouse gases) [4] and is rendering the system out of balance with nature.

Although researchers and engineers in computing domain hold concern regarding the environment, emphasis has been placed on innovation designed to sustain Moore's Law previously, with little thought about the related total energy consumption. With growing concern about global climate change and volatility of energy markets, computing domain has begun to embrace the notion of Green ICT, in which the environmental impact of ICT is taken as a consideration in the design of new technologies and systems.

Assembling a broad sample of the published material, both scholarly and popular on the energy element of ICT, we observed considerable divergence of estimate. Contact with major vendors and market research firms was helpful, but we soon accepted that

we would be wise to make our own attempt to quantify the ICT energy consumption and thus the concomitant carbon emission figure. Some previous studies have aimed to identify the energy consumption and/or carbon footprint of specific IT sectors such as PCs [5] or data centers [6] often confined to a particular region such as Europe [7] or USA [8]. Very few [9] attempt to measure the global ICT energy consumption and resulting carbon footprint. While ambitious in scope, the Smart 2020 report [9] holds several inconsistencies in its estimates (usage of a single carbon conversion number for all the years, no justifications of assumptions or projections through a mathematical framework to name a few). One researcher in particular, Koomey [10] provided ample warning on the hyperbolic nature of assertions regarding the quantity of electricity consumed by ICT. Did the electricity consumption of the network activity supporting the use of a single cell phone equal that of a refrigerator? Was one policy pundit right to assert that about 1 pound of coal [is required] to create, package, store and move 2 megabytes of data? We didn't know. Some detailed data was available, particularly work on data centers [6], but anecdote dominates the discourse.

Another worry is that putting the brakes on carbon emissions will also stymie economic growth in most-developed and developing countries alike. Although there may exist a reduced propensity to emit in most-developed countries, emissions have been and will likely remain closely linked to population and economic growth in the developing world [11]. As C.K. Prahalad argued [12], there is considerable room at the bottom of the pyramid that is the global economy; however this growth will also come at some sort of carbon price.

Hence, building on the work of others such as [8, 6, 13], we have generated estimates of the energy consumption and corresponding carbon emissions of the ICT sector and then, through the metric of Sustainability Innovation Quotient (SIQ), we

correlate the carbon emissions of ICT sector with the impact on national economic growth (measured through the Gross Domestic Product or GDP), thereby setting a strong motivation for a need for radical departure from the conventional ICT design methodologies through sustainable innovations with *Inexact Design Methodology* being the prime focus in this thesis.

For the purposes of this paper, Information Technology (IT or ICT, a commonly used international term which wedges Computing into the definition) is an umbrella that includes all technologies for the manipulation and communication of information.

2.1 Our Methodology for Estimating Carbon Footprint of ICT

2.1.1 A General Methodology for Projections

A central element in any predictive estimate is its ability to project future values accurately based on historical trends. We present a mathematical framework, which specifies the basis of our estimations or projections explicitly. Specifically, whenever the term projection is used by us, we adopt the following approach: estimating the value of a particular variable at a future time instance (for example, the number of PCs in use in the year 2020), the measured historical data to the current time, are compiled first. Regression analysis is then used to determine a function (curve) that fits or best approximates the historical data. The quality of the approximation is quantified through some metric, typically the mean square error. However, as is the case with any regression analysis, one significant pitfall is that all future estimates have a degree of uncertainty, since they are assumed to be some reasonable linear, quadratic or other well-known growth function, of the past. We found that all of

these attempts to extrapolate, sometimes referred to as business as usual trends, are vulnerable to unpredictable extremes such as the economic downturn of 2008-2009, the dot com bubble burst and so on. We wish to acknowledge this potential for being vulnerable to such unpredictable events in our business as usual predictions.

2.1.2 Illustrating our Methodology for Energy Consumption of the ICT Sector

Let us use the personal computer or PC as a driving example to explain our methodology, which consists of two elements. The first element consists of estimating the number of devices in use for a particular year, called the established base (EB), while the second encompasses an estimate of annual energy consumption (AEC) of each PC. Operating under this assumption, the global annual energy consumption ($GAEC$) of the PC sector for a particular year may be expressed as shown in Equation 2.1.

$$GAEC_{PC} = EB_{PC} \times AEC_{PC} \tag{2.1}$$

The annual energy consumption of a PC is estimated by dividing its state into 3 operating modes: active, when the device is on and the processor can be functioning; sleep, when the processor is on a standby mode; and off, when the device is switched off but remains plugged into the electrical socket. The annual energy consumption for a PC in each of these modes can be obtained in turn by multiplying the average power consumption in that mode with the annual usage factor which is defined to be the number of hours per year that a PC is in a particular mode.

This methodology, as summarized in Figure 2.1 can be applied to any type of a device from the ICT space. While we tried to follow this methodology as far as possible in our analysis, scarcity of reliable data forced us to adapt this methodology

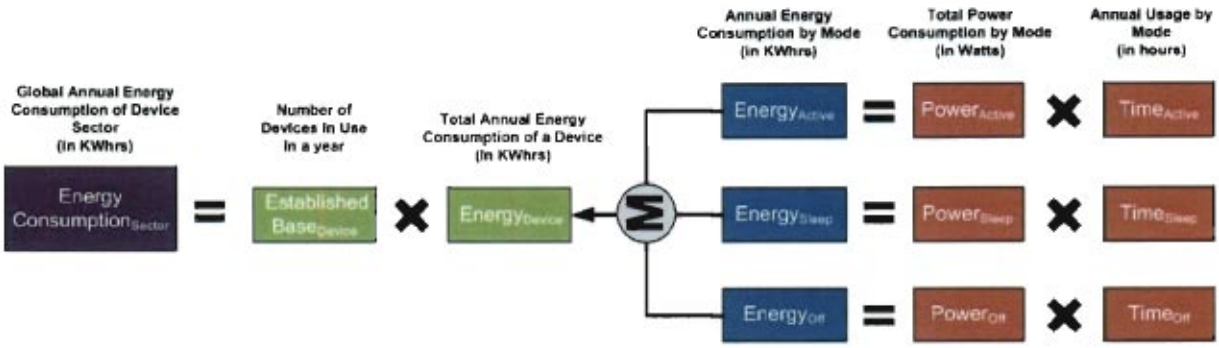


Figure 2.1 : Methodology for Computing the Global Annual Energy Consumption (GAEC) of an ICT Sector

further in a few cases, notably in the energy estimation of mobile phones as we will see in the following section.

2.1.3 Sub-sectors of ICT and their Annual Energy Consumption

We chose to broadly classify ICT into 4 main sub-sectors as shown in Figure 2.2 : Data Centers (DC), Personal Computers (PC), Mobile Devices (M) and Gaming Consoles (GC). In order to produce an estimate of the total global energy consumption of ICT and thus, estimate resulting carbon emissions, we analyzed the trends and projections in each of the sub-sectors separately.

Personal Computers (PCs)

We consider the PC sector to consist of a wide range of devices, from netbooks and laptops to desktop computers, with widely varying energy consumption values. We classify PCs into two broad categories for our estimates: laptops, which consist of all portable and mobile PCs, and desktops, which are fixed PCs connected to an external monitor. For our projections, the historical PC established base numbers were

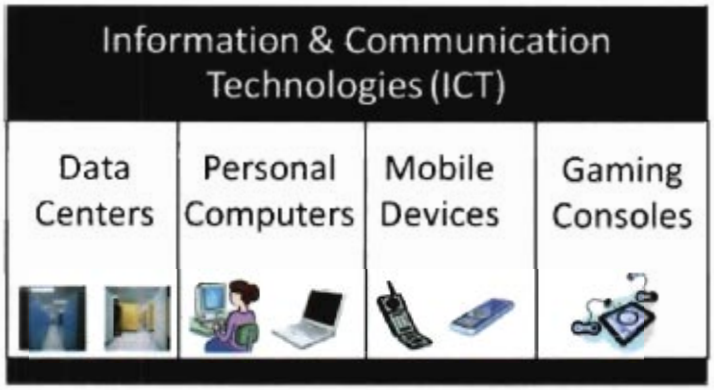


Figure 2.2 : Sub-sectors of ICT

obtained from the sales numbers provided by consultancies Gartner [14] and IDC [15]. The established base was measured based on a sliding window of a life cycle of an individual PC derived from [7]. Annual usage factors as to how the active, idle and off modes quantitatively relate to each other in a day were obtained from estimates published by the U.S. Department of Energy [8]. Lower bound values of the active mode power consumption of representative desktop and laptop models were obtained from [16, 17, 5], while the idle and off mode power consumption values were obtained from the Energy Star ratings [18]. Using these sources, we make the projections on the total PC sector energy consumption using the methodology from Figure 2.1.

Data Centers

A data center is a facility used to house computer systems and associated components, such as telecommunications and storage systems. Using the methodology described in Figure 2.1, the two primary measures that must be ascertained are the established base of components and their average power consumption within a data center. Our estimate for the historical data for the established base of data centers is based on the

sales figures for server computers from market intelligence firm IDC [15]. Using this historical data yielded projections on the established base until 2020. A typical data center is always in the on mode, and therefore, to estimate the energy consumption of a data center, we only consider the active mode of power consumption. In order to accurately estimate the power consumed in this mode, we divide all data center servers into 3 different categories following Koomeys thinking [6]: volume servers, mid-range servers and high-end servers. The power consumption in each category is estimated using a weighted average of the power consumption (Figure 2.1) of the top six server models in each category, determined by the sizes of their respective established bases.

Mobile Devices

Mobile computing, or the evolution of digital cellular communication, has grown to be the most ubiquitous form of computing on the planet, with more than 60 percent of the world's population now using some sort of handheld, connected computing device [19]. Despite their widespread usage, such devices contribute a small share of global ICT energy consumption as they are used, owing to their dependence on battery capacity which hasn't grown significantly over the last few decades when compared to the computing power [20]. In this study, we constrain ourselves to an estimate of the energy consumption and the resulting estimate of the carbon footprint of mobile ICT devices themselves (smartphones, etc.), and will ignore the back-end infrastructure, including antennae, base station computing and fixed links. We do note that not all of the costs associated with cell phone usage are ignored in our study, since the components that are computational tasks being performed on data centers are accounted for through the estimates above. For the projections of energy consumption in this sector, the size of the established base was obtained from the world mobile

usage statistics from the CIA World Factbook [21] and from sales numbers provided by Gartner [14].

Referring back to our overall methodology in Figure 2.1, we need an accurate estimate of the energy consumption per-device. To estimate this, we use the battery capacity (BC) of a mobile device and its charging frequency (N). As mobile devices are seldom used while they are plugged in, we will assume that the annual energy consumption of the mobile device is the product of its battery capacity and it's charging frequency. Favoring a conservative approach that potentially underestimates this value, we will use the anecdotal measure that a typical mobile phone is charged only once in 2 days, a figure many current generation smart phone users will argue to be quite low. Based on this approach, the annual energy consumption due to a single mobile device can be estimated as shown in Equation 2.2.

$$AEC_M = BC_M \times N_M \tag{2.2}$$

Gaming Consoles

Video game consoles are interactive entertainment computers that can be used with a display device, typically a television or a monitor. A rapidly growing component of computing, we believed video game consoles warranted analysis of their carbon footprint, especially owing to their increasing popularity. We found that in isolation, gaming consoles such as Xbox and Playstation without the (television) display consume as much power as a desktop PC and hence, only estimated the energy consumption of the gaming consoles without including the TV energy consumption values.

To estimate the energy consumption of the Gaming consoles sector, the methodology described in Figure 2.1 was used. The projections are built on historical established

	2009	2015	2020
Data Centers	205.28	399.78	660.86
Personal Computers	214.39	386.79	923.91
Mobile Devices	2.61	6.51	11.77
Gaming Consoles	19.00	45.28	71.94
Total	**441.30**	**838.36**	**1668.49**

Table 2.1 : Summary of Projections for ICT Sector Energy Consumption in Billions of KWhrs for Various Years

base data obtained from the reported annual sales numbers [22] with an assumption of a 4-5 year life cycle (based on the average life cycle of a gaming console version before a newer version is released). The annual average energy consumption of a gaming console is computed from the product of the weighted average power consumption of a console in each of the three modes as before, obtained from [23], and the time spent in each mode over the course of a year from [13].

2.1.4 Summary of Energy Consumption Across ICT Sub-sectors

Summarizing our effort from the previous subsection, the energy consumed due to the electricity used by various parts of the of the ICT sector during the year 2009, and projections for the years 2015 and 2020 are given in Table 2.1.

2.1.5 Converting Energy Consumption into a Carbon Footprint

We found that the conversion of energy consumed into the corresponding carbon emitted is not a simple process since the amount of carbon dioxide emitted per unit

energy consumed varies over time, and depends on the source of electricity production ranging over coal, natural gas, nuclear energy and others. To try and consolidate this, we define a metric called the Carbon Conversion Number (CCN) shown in Equation 2.3 through which we intend to denote the amount of carbon emissions (usually in lbs or Kgs) per unit of energy consumption (usually in Joules or KWhrs).

$$CCN = \frac{\text{Carbon Emissions}}{\text{Electricity Consumption}} = \frac{lbs \text{ or } Kgs}{KWhr} \tag{2.3}$$

We note in passing that this metric can be generalized where the source of energy can be more broadly defined and need not be tied to electricity. We base our estimates of CCN on the projections of global electricity consumption and global carbon emissions due to electricity generation until 2030, published by the International Energy Agency [24]. Based on the information available from these sources, we computed the CCN, expressed as lbs of CO_2 per KWhr of electricity consumption. Next, using the CCN values and combining them with the findings summarized in Table 2.1 earlier, we are able to compute the carbon footprint expressed in Mega tonnes of carbon. The results are summarized in Table 2.2, for each of the four ICT sectors, and also as cumulative amount for all of ICT.

2.2 The Sustainability Innovation Quotient (SIQ)

Although there are few fixed factors at play in a global model for ICT's relationship to productivity, we do see the employment of ICT as a major factor in productivity. We also argue for the consideration of a measure that brings to ICT the value of energy efficiency of relevance to both technical and economic thinking. Thus we propose a Sustainability Innovation Quotient (SIQ) which is factored through measurement energy consumption, the concomitant carbon emissions of that consumption and

Table 2.2 : Summary of results for ICT Sectors Carbon Footprint in Megatonnes of CO_2 for Various Years

	2009	**2015**	**2020**
Data Centers	121.30	229.87	369.48
Personal Computers	126.69	222.41	516.55
Mobile Devices	1.54	3.74	6.58
Gaming Consoles	11.23	26.04	40.22
Carbon Conversion(CCN)	**1.3**	**1.265**	**1.23**
Total	**441.30**	**838.36**	**1668.49**

economic output (measured through the Gross Domestic Product or GDP metric) of ICT . The SIQ for any sector (ICT in our case) at a given time (usually a year) may be computed through Equation 2.4 and expressed in $\left(\frac{\text{dollars}}{\text{kg of } CO_2}\right)$

$$SIQ_{sector} = \left(\frac{\text{Change of GDP}_{sector}}{\text{Change in Carbon Emissions}_{sector}}\right) = \frac{\text{d}(GDP)}{\text{d}(CE)} \tag{2.4}$$

2.2.1 An Demonstrative Example for SIQ Computation of the United States

In the previous section, we projected the global carbon emissions of the ICT sector looking ahead until 2020. To compute the Sustainable Innovation Quotient (SIQ) of the ICT sector, we now need to determine the GDP of ICT over the same time period. In the context of global carbon emissions, reliable retrospective data in various forms such as the installed base of computers, their usage patterns and others, was essential. Through our study, we could unearth data we consider reliable, for ICT's share of

	2009	**2015**	**2020**
Data Centers	40.72	68.48	93.92
Personal Computers	27.55	41.48	79.95
Mobile Devices	0.13	0.32	0.58
Gaming Consoles	4.56	12.13	20.61
Carbon Conversion(CCN)	**1.251**	**1.217**	**1.176**
Total	**72.95**	**122.41**	**195.06**

Table 2.3 : Summary of results for ICT Sectors Carbon Footprint in the United States Megatonnes of CO_2 for Various Years

GDP for the US [25], UK and Canada [26]. As a result, we focus our attention on developing and demonstrating the SIQ concept for the USA. In principle, given access to the worldwide GDP data and ICT's contribution to it, our methodology can be extended to the global context. To complete the determination of the SIQ, since we already computed ICT's carbon footprint and its projection to 2020, the GDP component remains to be extrapolated.

Computing the Carbon Emissions of ICT in the United States

To derive the ICT carbon emissions in the United States, we tailor the methodology used in the previous sections. The change involves using US sales data in each sub-sector as opposed to using global sales data. Also, care was taken to specialize the CCN value to be specific to the US. Based on this, the summary of the carbon emissions of the ICT sector in the US is shown in Table 2.3.

	2009	2015	2020
GDP of ICT Sector	461.46	571.13	662.5

Table 2.4 : Summary of Results for ICT Sector's GDP in the United States in Billions of USD

Computing the GDP of ICT in the United States

Using the historical data of the share of ICT sector in GDP of the United States [25], we project the GDP share of the ICT sector until 2020. Due to wide fluctuations in the historical data, it is difficult to project the future values. Hence, GDP values of a subset of the historical data set (years 2002-2008) are taken as the basis to project the future values of the GDP of the ICT sector. (Readers are cautioned that as with majority of economic projections, this projection is only a simple business as usual projection and the actual values might deviate significantly from the projected values due to fluctuations in the economic data analysis). A summary of some of the projections of ICT's future GDP values is given in Table 2.4.

Computing the SIQ of ICT in the United States

Once we projected carbon emissions as well as the GDP of the ICT sector, we computed the SIQ of the ICT sector using the Equation 2.4. While the data representing the change in the carbon emissions as well as the change in GDP are discrete sets, we characterize the trends that they represent by interpolating a continuous function. This allows us to use a differential as a basis for the SIQ and should discrete values be desired, we can replace this with a similar definition using finite differences instead. A summary of SIQ values of the ICT sector over a few years is shown in Table 2.5. This

	2009	2015	2020
SIQ of ICT Sector	2.831	1.657	1.060

Table 2.5 : Summary of Results for ICT Sector's SIQ in the United States

can be interpreted as: in 2009, we are obtaining an economic output of $2.831 per kilogram of CO_2 emitted by the ICT sector, while in 2020 we only obtain an economic output of $1.06 per kilogram of CO_2 emitted.

2.2.2 Innovation as a driver for improving the SIQ

As seen from Table 2.5, the economic output of ICT as represented by GDP for a fixed carbon budget is diminishing. The underlying cause of this can be attributed to the steadily increasing CO_2 emissions of the ICT sector where the efficiencies are lagging the concomitant growth in the GDP. A typical response would be a call for a need to take stringent measures to curb the steadily increasing CO_2 emissions of the ICT sector.

Our notion of SIQ however informs us that this can have a deleterious effect on the continued potential of ICT technologies to be a dominant contributor to the future growth of the GDP, in the context of an acceptable ceiling on the total amount of carbon from any sector. This is shown in Figure 2.3 where, with a hypothetical ceiling of 117.5 Mt of carbon for the ICT sector. As shown in Figure 2.3, with an SIQ of 1.732 and with this ceiling, the contributions from the ICT sector will be limited to $561.99 billion dollars by 2020 and would not grow past this amount starting mid-2014. In stark contrast, in conjunction to innovations and with an improved representing an improvement by a factor of 2.74X – the ICT sector has the improved potential for

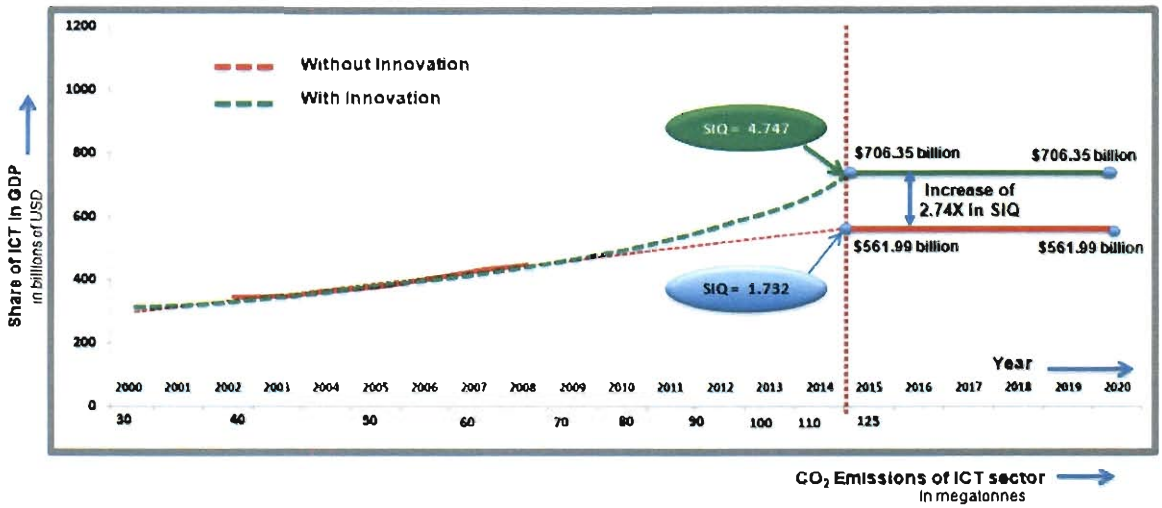


Figure 2.3 : Scenario when ICT's carbon emissions are frozen at 117.5 Mt for the United States and the value of Innovation to improving ICT

contributing an additional $144.36 billion to US GDP by 2020, while respecting an overall ceiling on the expended carbon.

In order to curb this phenomenon and increase the net resulting GDP, we anticipate that the role of innovation in ICT is critical. As demonstrated in Figure 2.3, innovation can play a critical role in increasing the economic value of the ICT sector. Intuitively, a higher value of SIQ is meant to imply that more energy efficient systems are the result, and therefore, for the same carbon budget, more of them can be deployed in the market place. One such innovation which we envision would help achieve energy parsimonious ICT devices is the notion trading accuracy for significant energy savings, popularly known as the Probabilistic CMOS technology (PCMOS) [27], which is the inspiration behind approaches proposed in this thesis.

Chapter 3

Parsimonious Design - Living with Less

Realizing reliable computations from unreliable components has long been a focus of study [28] and is receiving greater than ever prominence today [29] as diminishing transistor sizes driven by Moore's law are leading to increasing process variations. It is due to these process variations, arising as lithographic scaling lags behind device scaling, and the quest for ultra-low energy circuits, particularly in the domain of embedded systems, that *Exact* computing, in which output of the desired circuit is precise, is paving the way for a new domain of *Inexact* computing wherein accuracy of the output of circuits can be traded in for significant savings in cost (energy, area and/or delay parameters).

3.1 Inexact Design and its Usefulness

In a large class of emerging applications, particularly in the field of embedded, multimedia and DSP systems and in application domains of growing interest such as recognition and data mining [30], tolerable amounts of error are still shown to still realize potentially useful computations and hence, in these *inexact* systems, error (either caused probabilistically due to inherent variations/perturbations or introduced deterministically) can be viewed as a commodity that can be traded for significant gains in energy and/or delay, first conceptualized in [27] and subsequently, widely adopted in [31, 32, 33, 34, 35, 36, 37, 38]. The error tolerance or resilience in these

applications can be attributed to many factors : (a) The "cognitive filling" capabilities of the end systems (such as human sensory systems) which consume the output. These systems have an underlying architecture which aids them in realizing *reliable* computations even from *unreliable* components [39]; (b) The underlying algorithms are often statistical or aggregative in that they inherently possess a mechanism through which any output within a particular bound is equally acceptable as the single "golden" output.

These inexact systems are parsimonious in terms of (physical) implementations and cost (in terms of energy, delay and or area) lesser than their conventional correct counterparts. Implementing such parsimonious inexact designs through innovative architectural approaches is the central focus of this thesis. We go on to show that taking advantage of the notion of *value of information* or "significance" in realizing such inexact circuits will help glean further savings in the cost.

3.2 Demonstrative Examples of Benefits of Inexact Design

In this section, examples showing the benefits of inexact systems are presented. Figure 3.1 shows the output of an H.264 video encoding (only one frame has been shown here) for various design approaches presented in [31]. As evident from these figures, the output of the inexact design guided by the principle of *value of information* is barely distinguishable from the output generated by conventional correct design while achieving a significant reduction in cost.

Another example showing the benefits of inexact design is presented in Figure 3.2 [38]. In this example, the cost(energy) gains that could be obtained by varying the quality of the obtained output image (measured through SNR metric) are shown. While multimedia applications could use the top image to consume 1.3X lesser energy,

ERSA(Error Resilient System Architecture) [49] that combines one reliable processor core with a large number of unreliable cores.

The application of this principle have been limited to two classes: (a) Probabilistic circuits, or circuits which are inherently erroneous due to process variations or perturbations such as (thermal) noise [50, 31, 37, 51]; (b) Deterministic circuits, or circuits where error was deterministically introduced exploiting the error tolerance or resilience of the application [36, 35, 34]. While the probabilistic version would be useful for the technologies down the road, it is the deterministic version which by far has been the most popular and widely used technique exploiting the error tolerance of a large number of present day applications to glean significant savings [32, 33, 52, 53].

Furthermore, we can classify most of the proposed approaches in literature for implementing inexact systems to exploit the error tolerance/resilience of the applications into two classes : (a) Application's algorithm modifcations such as [54, 55, 56, 52, 57] , and (b) Circuit level modifications (mostly concentrating on variants of voltage scaling) such as [31, 36, 58, 59, 35, 34, 33]. One striking aspect of the research concentrated in this area is the lack of any innovations at the architecture level of implementing the inexact systems and it is exactly the innovations in this aspect which form the prime focus of this thesis.

Chapter 4

Understanding the Design Abstraction Levels for Inexact Designs

While the unconventional principle of inexact design opens up several novel directions for trading off accuracy for savings, there are serious impediments when one considers integrated circuits based on this principle. As mentioned in Chapter 3, most of the (physical) implementations of this principle so far involved tweaking the operating parameters (such as supply voltage (V_{dd}) or frequency) of the conventional hardware taking advantage of error tolerance of the application. Putting this in the context of global scheme of designing circuits for error tolerant/resilient applications, this approach will not yield optimal circuits with substantial gains (in energy, delay and/or area).

4.1 Design Abstractions for Inexact System Implementations

In general, the hardware implementation of a system/application can be divided into three layers of abstraction: Algorithm, Architecture and Circuit. An optimal implementation of a system involves a cross-layer optimizations across all the layers of abstraction. However, most of the inexact designs realized so far have only tried to optimize designs at the circuit level using operational parameter-scaling (such as supply voltage V_{dd}) approaches guided by the error tolerance/resilience dictated by the algorithm. As shown in Figure 4.1, most of the innovations to realize inexact systems have been either at the algorithm level or at the circuit level. One critical

supply voltage) scaling. The drawbacks of such parameter scaling based approaches are multifold:

1. Fine-tuning of supply voltage at run-time based on the application requirements might not be feasible due to inherent variations present in the power supply routing [60] and by the large overhead generally required to ensure that such an accurate fine-tuning is realized necessitated by the possibility of massive failures that can occur in circuits beyond a critical voltage scaling point [35];

2. One physical realization referred to as Biased Voltage Scaling (BiVOS) [31, 36] is seriously impeded since it involves significant overheads of routing multiple voltage planes, and by necessity for level shifters;

3. Varying supply voltage during circuit operation coupled with the inherent power supply variations might also increase the possibility of timing failures due to metastable conditions and might require metastable tolerant flip-flops or latches adding to the already increasing overhead.

Based on these drawbacks, conventional voltage scaling based approaches might not be a wise option to realize inexact circuits (in particular, datapath elements) as they tend to significantly reduce (or even nullify!) the gains that can obtained by the accuracy tradeoff. This highlights the growing importance of the need to move away from voltage scaling based optimizations at the circuit level to higher abstraction levels to continue to glean substantial gains from the accuracy tradeoff.

In this paper, we overcome all of these drawbacks of conventional approaches to design inexact circuits through a architectural redesign techniques called *Probabilistic Pruning* and *Probabilistic Logic Minimization*, zero-overhead, technology-independent

circuit design techniques for error-tolerant applications yielding significant gains across all 3 dimensions – energy, area and delay – for comparable error.

4.3 Useful Metrics for Designing and Analyzing Inexact Circuits

In this section, we propose some useful metrics needed to analyze and compare the gains obtained by the inexact designs along with a heuristic to guide the architectural optimizations through the notion of "value of information" or significance assigned to portions of the (inexact) system (generally guided by the application algorithm).

4.3.1 Defining Error Metrics

We can broadly classify error resilient applications into two types : ones which have a bound on the total number of erroneous computations (such as number of incorrect memory address computations in a microprocessor) and others (such as computation of the value of a pixel by a graphics processor) which have bounds on the magnitude of error. While in the former type applications, each of the output receives equal importance or "significance" and errors are quantified through the *error rate* metric, the outputs in the latter applications have a certain importance or *weights* depending on the magnitude of error and are quantified through the *relative error magnitude* metric, similar to the ones proposed in [61] and widely used in [36, 38].

$$\text{Error Rate} = \frac{\text{Number of Erroneous Computations}}{\text{Total Number of Computations}} = \frac{\mathcal{V}'}{\mathcal{V}}$$

$$\text{Relative Error Magnitude} = \frac{1}{\mathcal{V}} \sum_{k=1}^{\mathcal{V}} \frac{|\mathcal{O}_k - \mathcal{O}'_k|}{\mathcal{O}_k}$$

where $\mathcal{V}$ is the total number of simulation cycles or test vectors given to the circuit, $\mathcal{O}_k$ is the expected correct output vector and $\mathcal{O}'_k$ is the obtained erroneous output vector for the k^{th} input vector

4.3.2 Computing the "Value of Information" or Significance Value

The notion of assigning significance based on the "value of information" principle to a circuit node is one of the guiding principles for achieving an optimal inexact circuit design. It should be noted that the significance value is generally derived from the application's algorithm and the type of circuit implementation (circuit topology) chosen. Hence, the proposed architectural redesign techniques take this assignment of significance as a parameter which can be modified based on various heuristics to obtain varying (yet significant) amount of savings. For the sake of completeness, we present a simple heuristic to assign significance to circuit nodes depending on the amount of error they can cause at the circuit outputs assuming the rest of the circuit nodes operate correctly. Note that, this is a circuit topology based heuristic and is not limiting in any sense that it could as well be combined with application algorithm's assignment to realize more optimal designs.

We consider the case of a single node that produces an error (averaged over the application's test vectors). Let us consider that for some test vectors, a circuit node i can cause an error at an output node O_t for $t \in \{1, 2, \cdots N_O\}$. Let Er($i$) and Er($O_t$) be the errors at the output of node i and corresponding output nodes O_t. Then, we define the significance of node i as $\sigma(i)$, computed as follows:

$$\sigma(i) = \frac{\sum_{t=1}^{N_O} Er(O_t)}{Er(i)}$$

The heuristic to assign significance described here can be implemented using a

mathematical model for simple circuits such as a ripple carry adder and use simulation based assignment while assigning significance to more complex circuits such as a multiplier.

Chapter 5

Proposed Approach I - Probabilistic Pruning

The central goal of this thesis is to show that introducing an error tradeoff through architectural modifications in the design of error tolerant or resilient circuits would lead to more significant savings in the energy, delay and area dimensions than any conventional circuit-level voltage scaling based design. The first such architectural level approach we propose is called *Probabilistic Pruning*. It is a architectural level design technique wherein we systematically "prune" or delete components and their associated wires along the paths of the circuit that have a lower probability of being active during circuit operation while staying within the error boundaries dictated by the application. As this approach is carried out during the design phase, it can be realized with *zero overhead* on the circuit hardware. In this chapter, we introduce a formal mathematical formulation of the proposed pruning technique along with an algorithm (flowchart) to implement it.

5.1 A Formal Mathematical Formulation of Probabilistic Pruning

A circuit can be represented as a *directed acyclic graph* whose nodes are components such as gates, inputs, or outputs and whose edges are wires. Given a circuit $\mathcal{G}$ with N_{C} components, N_{I} inputs, N_{O} outputs and N_{W} wires, our goal is to prune components in the paths such that the energy, area and speed are reduced while maintaining a

bound on error, say σ. Let **I** be the set of all input nodes, **O** be the set of output nodes, **C** be the set of all components and **W** be the set of all wires.

We now formulate an optimization problem of computing a circuit $\mathcal{G}$', which is a subgraph of $\mathcal{G}$ such that it has the same set of inputs {**I**} and outputs{**O**} but with components {**C**'} where $N_{\mathrm{C}'} \leq N_{\mathrm{C}}$ and wires {**W**'} where $N_{\mathrm{W}'} \leq N_{\mathrm{W}}$ such that given $\mathcal{V}$ randomly chosen inputs, the average error

$$\mathrm{Er}(\mathcal{G}') = \sum_{k=1}^{\mathcal{V}} p_k \times |\mathcal{O}'_k - \mathcal{O}_k| \leq \sigma \tag{5.1}$$

where $\mathcal{O}_k$ and $\mathcal{O}'_k$ correspond to value of final output vectors $< \mathcal{O}_{k,1}, \mathcal{O}_{k,2}, \ldots \mathcal{O}_{k,n} >$ and $< \mathcal{O}'_{k,1}, \mathcal{O}'_{k,2}, \ldots \mathcal{O}'_{k,n} >$ of circuits $\mathcal{G}$ and $\mathcal{G}$' respectively for a given n-bit input vector $\mathcal{I}_k$ which occurs with a probability p_k for $1 \leq k \leq \mathcal{V}$. In the unweighted case, $|\mathcal{O}_i - \mathcal{O}_j|$ is the value of the difference between $\mathcal{O}_i$ and $\mathcal{O}_j$ treated as unary numbers. In the case where the output bits are weighted, without loss of generality, we could assign a weight η_j to the j^{th} output bit O_j. In this case, $\mathcal{O}_i$ and $\mathcal{O}_j$ will be the difference between the corresponding binary numbers. In fact, starting with Section IV, we will be demonstrating the value of probabilistic pruning through circuits for integer addition and therefore, will be considering the case of weights $\eta_j = 2^j$ almost exclusively.

Output: A *pruned* $\mathcal{G}'$ that is optimal in that there is no other $\mathcal{G}''$ satisfying the conditions above such that $N''_{\mathrm{C}} < N'_{\mathrm{C}}$.

The average error computation metric used above is not limiting in any sense that it can conveniently replaced by any other error metric based on the application requirements in using the probabilistic pruning approach. In circuit design, it is considered to be meaningful to evaluate a design using a range of inputs drawn from a distribution and analyze error following approaches to average case analysis [62], and we will adopt this approach. Not surprisingly, it is easy to show that variants of this

problem are NP-hard in general [63, 64]. However, our main goal in this paper is to demonstrate the value of applying probabilistic pruning to circuit design. Therefore, we will not emphasize the algorithmic nuances in this thesis, but will rather use a simple-minded and (almost) brute force heuristic here, which is shown in Figure 5.1.

5.2 Algorithm for Probabilistic Pruner

The algorithm to the proposed probabilistic pruning approach is given in Algorithm 1 along with a flowchart in Figure 5.1. For any given node i in the graph $\mathcal{G}$, we have

- node.significance(i) – denotes the significance of node i, the assignment of which is described in Chapter 4.
- node.activity(i) – denotes the transition probability of node i. It is computed as a relative frequency across the test vectors.
- node.sap(i) – denotes the significance-activity product (SAP) of the node

5.3 Advantages of Probabilistic Pruning

To the best of our knowledge, the proposed probabilistic pruning techniques is the first architectural design technique targeted for Inexact systems for error tolerant or resilient approaches. The advantages of the proposed probabilistic pruning technique are multifold :

1. As the technique is used to realize or "synthesize" inexact circuit architectures to start with, it has *zero overhead* on the circuit hardware in terms of energy, delay and area. In other words, the proposed technique obtains savings in all 3

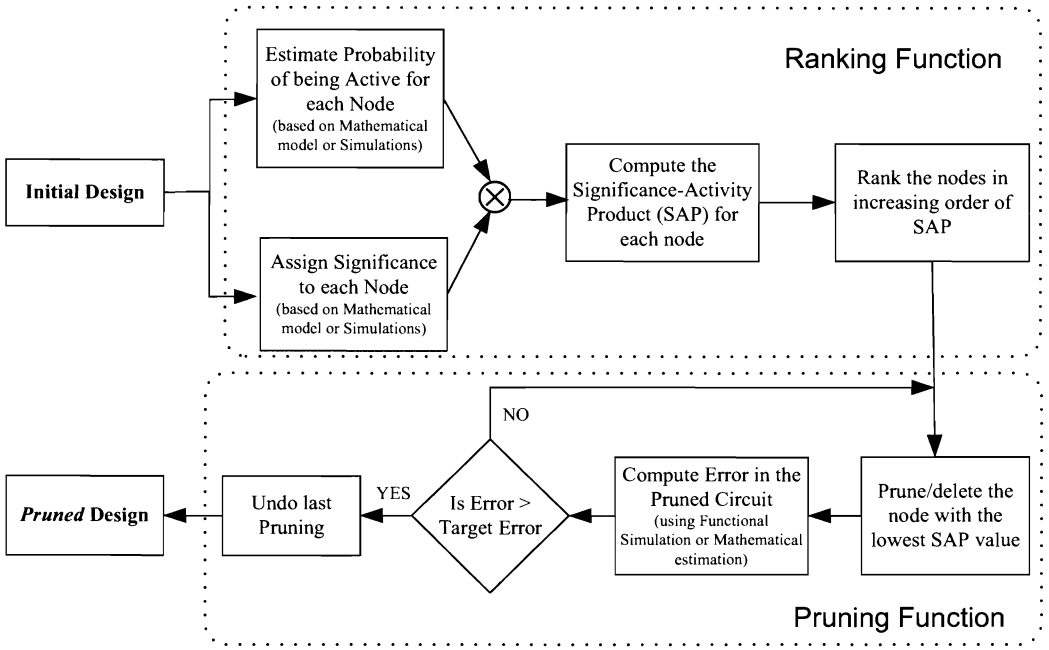


Figure 5.1 : Flowchart for the Probabilistic Pruning

dimensions - energy, delay and area - when compared to its conventional correct counterparts.

2. Since, it is an architectural level implementation, the proposed technique guarantees a bound on the error (average or worst case) for the inexact circuit realization unlike the circuit level scaling techniques (such as voltage scaling). This can be attributed to the fact that the proposed technique is independent of parameter (such as supply voltage) variation and hence, resilient to metastability or timing failures and doesn't have a critical (voltage scaled) point that might cause massive failures [35]!

3. The proposed technique doesn't have the overheads of level shifters, multiple voltage planes or metastability-tolerant latches, typically needed for the operation

of circuit level voltage-scaling based design techniques

4. The proposed technique is technology independent as the amount of gains are only proportional to the amount of nodes pruned and not on the process technology parameters (will be illustrated in Chapter 9) as opposed to the voltage-scaling based schemes in which the amount of gains is limited by the process technology constraints. For example, in the present day deep submicron CMOS technology nodes (65nm and below), the supply voltage is typically around 1V and the threshold voltage is around 0.3-0.5V. Hence, the amount of voltage scaling that could be done is very limited and so are the gains.

5. For the present day and future deep submicron CMOS technologies (45nm and below), leakage power forms a significant portion of the total power consumption. The proposed technique by virtue of its significant reduction in the total number of "leaky" transistors reduces the total leakage power of the system as well.

6. Lastly, the proposed technique can be easily integrated into the traditional system based CAD flows (as shown in Chapter 9), thereby reducing the design effort and time as opposed to most of the circuit-level design techniques, in particular the Biased Voltage Scaling (BiVOS) proposed in [31, 36], that require a custom design flow for (physical) implementation.

Algorithm 1 Pseudo-code for the *Probabilistic Pruning* algorithm on a Circuit or Graph $\mathcal{G}$

```
//Main Function in the Algorithm
function PRUNING(MaxError)
    //Compute the probability of switching or activity of each node
    Benchmark( );
    //Compute the Significance-Activity Product of each node
    for all i ← 1 to N do
        node.sap(i) = node.significance(i) × node.activity(i);
    end for
    //Iteratively prune each node with the lowest SAP value until the error bound is
reached
    while Error ≤ MaxError do
        NodetoPrune = FindMinimum (node.sap);
        PruneNode(NodetoPrune);
    end while
end function

function BENCHMARK
    RunBenchmark();
    for all i ← 1 to N do
        node.activity(i)(j) = ComputeNodeSwitchingProbability;
    end for
end function
```

Chapter 6

Proposed Approach II - Probabilistic Logic Minimization

The second architectural redesign technique for inexact systems that we propose is *Probabilistic Logic Minimization.* It is a design level technique wherein we systematically minimize circuit components (or nodes) guided by the significance and the input combination probabilities of those nodes while staying within the error boundaries dictated by the application. We make an observation that probabilistic pruning can be termed as a "bottom-up" approach to inexact system design wherein, given a conventional circuit architecture, we identify the components which can be pruned or deleted guided by certain heuristics. On the other hand, probabilistic logic minimization is a "top-down" approach wherein we try to *synthesize* an inexact circuit through further logic minimizations of its boolean function based on various heuristics.

6.1 An Example of Applying Probabilistic Logic Minimization

The key to the probabilistic logic minimization algorithm is the notion of introducing bit flips in the minterms of boolean functions to further minimize them, thereby achieving gains (energy/area/delay) through literal reduction while causing an error due of such bit flip(s). However, not all bit flips of minterms would result in expanding the prime implicant (PI) cubes and some of them might result in negative gains. Hence,

it is important to identify the "favorable" bit-flips (or the bit-flips which further minimize the function) and discard the non-favorable ones. To illustrate through an example, Figure 6.1(a) shows a function (Carry logic) that is widely prevalent in most datapath elements. Assuming that the application would only be able to tolerate at most one bit-flip at this logic function (probability of error = 1/8), Figures 6.1(b) and 6.1(c) give an example of favorable 0 to 1 and 1 to 0 bit flips respectively as they minimize the logic function whereas Figure 6.1(d) shows an unfavorable bit flip leading to an increased logic function complexity. Hence, we can conclude that *introduction of favorable bit-flips would lead to further minimization of a logic function owing to the expansion of PI cubes, thereby achieving cost (energy, area and delay) gains at the expense of error which is proportional to the number of such bit flips introduced.*

While the benefit of this imprecise minimization cannot be denied, it gives rise to another interesting question: *Given a circuit node with many favorable bit flip possibilities, each with similar cost gains, how do we select the right minimization for the node?.* In other words, is the error introduced by each of the bit flips equal? While conventional wisdom calls for an assumption of uniform input combination probabilities, it is never the case with most applications, more so with multimedia applications where inputs are highly correlated and hence, our proposed technique takes advantage of such correlation to guide the minimization algorithm and glean further savings.

In general, given a circuit node with n inputs, there are 2^n possible minterms of which we could flip the bits at atmost k minterms (depending on the application's error tolerance) to derive the minimum cost function. We propose a probabilistic extension to the minimization scheme wherein all the favorable bit flips are ranked based on their input combination probabilities and the bit flip(s) having the least

Figure 3.1 : Illustrative example showing the benefits of Inexact design in an image (a) Image processed by conventional correct electronics (b) Image processed by inexact electronics (c) Image processed by "value of information" based inexact electronics

(image) recognition applications would find the bottom image acceptable as well, thereby achieving 3X reduction in energy consumption.

3.3 Survey of Inexact Design based Methodologies in Literature

Several papers in the past have investigated ways of overcoming the challenges to process and parameter variations threatening the sustained evolution of Moore's law. Some of the prominent methods include using multicore architectures to increase parallelism without frequency/voltage scaling, designing for average case operation and using temporal and/or spatial redundancy to correct worst-case errors [40, 41, 42]. Exciting new research into novel materials for realizing circuits such as optoelectronics, memristors [43] and molecular electronics [44, 45] have also been investigated successfully. The question is of such great significance to our daily lives that even the articles in New York Times are actively discussing these issues [46]. One common principle of conventional design approaches has been to ensure that the device always functions

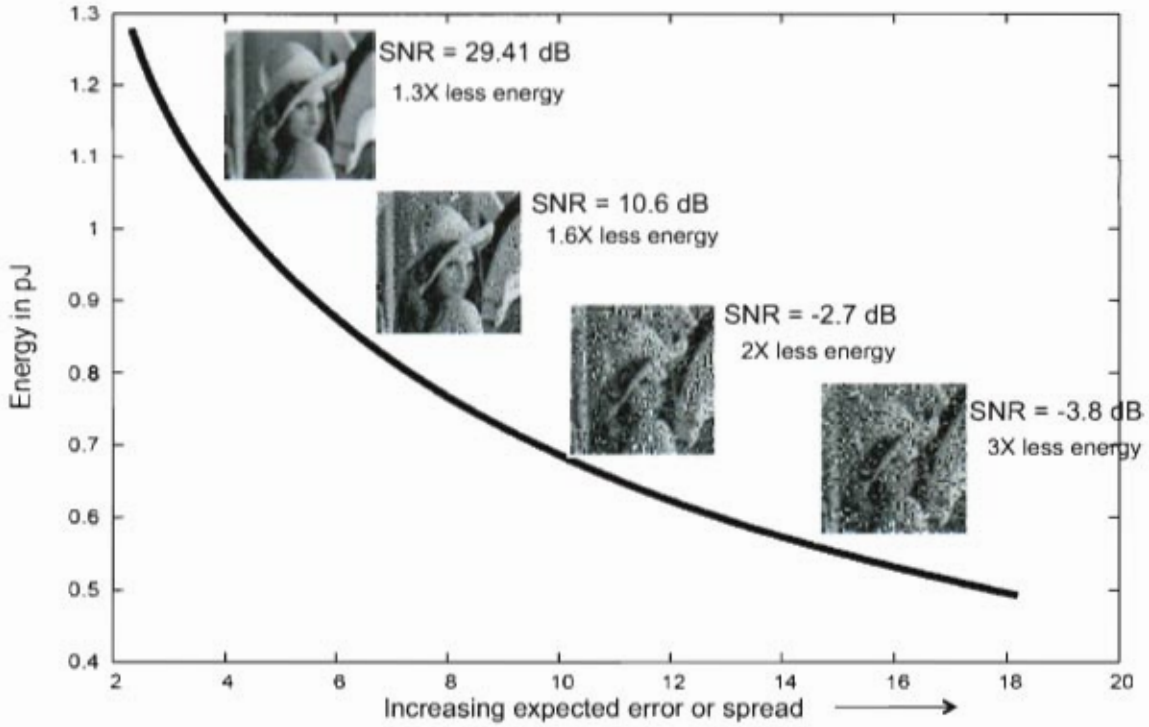


Figure 3.2 : Illustrative example showing the tradeoffs between energy and error involved in Inexact design for an image

correctly, either by design, or through an error-correction mechanism using temporal and/or spatial redundancy to correct worst-case errors [41, 42].

In a radical departure from these conventional approaches, it was shown in [27, 47] that error can be traded as a commodity as opposed to being viewed as an impediment to glean significant savings (typically energy) – in applications that can accommodate error. Fortunately, a large class of emerging applications, particularly in the domain of embedded and mobile systems, can tolerate varying amounts of errors, more so when it results in significant energy savings. A CMOS realization of this principle called PCMOS was given in [48] and was later, extended to realize a system level application through an SoC architecture [37] and a programmable multi-core architecture called

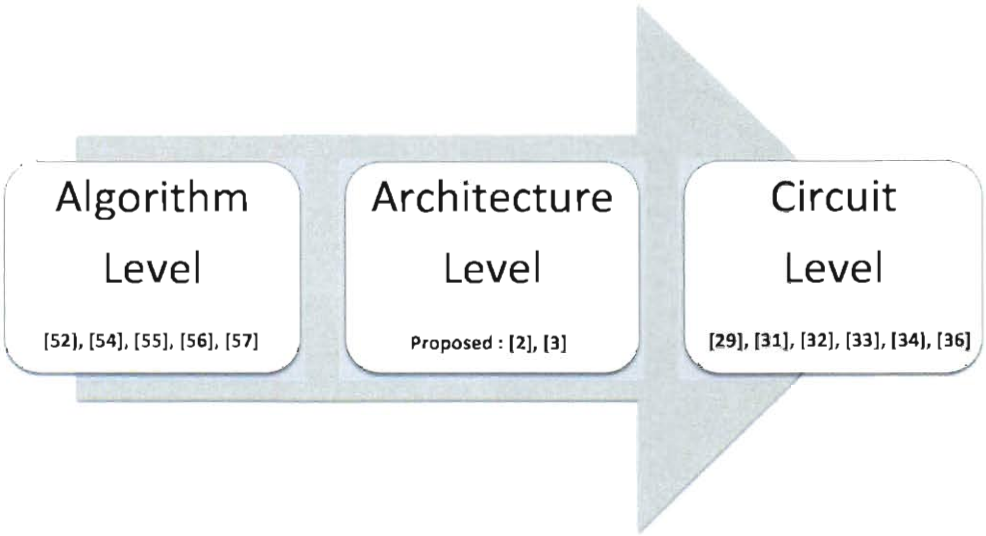


Figure 4.1 : Innovations at various design level abstractions for inexact system implementations

element that has been missing in all the prior realizations of inexact systems is a focussed effort to propose novel approaches at the architecture level and tie-in of those approaches to the other two levels of abstraction to glean further savings in cost for the same accuracy tradeoff. Given that the existing circuit level techniques have significant drawbacks (as outlined in the next subsection) which significantly reduce the possible gains in the inexact systems and that the algorithm level approaches have hit a roadblock, there is a dire need for innovations at the architectural level which could translate to substantial gains for the entire system.

4.2 Drawbacks of Existing Implementations of Inexact Designs

As mentioned previously, most of the existing efforts to realize inexact systems are concentrated at the circuit abstraction level using variants of parameter(particularly

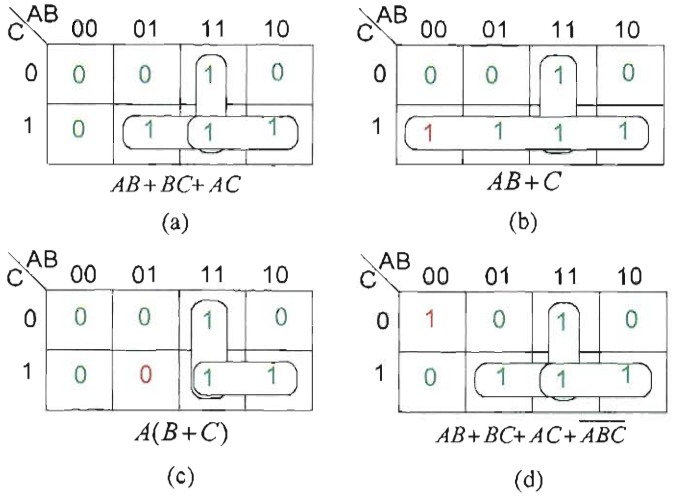


Figure 6.1 : Example of K-Maps of the (a) Initial Correct Function (Carry Logic of a Full Adder) (b) Function with a favorable 0 to 1 bit flip (c) Function with a favorable 1 to 0 bit flip (d) Function with a non-favorable 0 to 1 bit flip

corresponding input combination probabilities are done. For example, in Figure 6.1(b) and (c), the minimized functions have the same gains (3 ORs and 2 ANDs function reduced to 1 OR and 1 AND). But if the probability of input to the logic function being '001' is higher than the input being '011', then a bit flip at '011' would likely cause an error with a lesser probability. Hence, in short, a bit flip occurring at the least likely input combination would result in lesser error for the same amount of savings. With this as background, we propose a general algorithm for application of the Probabilistic Logic Minimization technique in the following section.

6.2 Algorithm for Probabilistic Logic Minimization

A circuit can be represented as a *directed acyclic graph* with nodes representing components such as gates (or even bigger blocks like full adders), input or outputs and with edges representing interconnects. Let a graph $\mathcal{G}$ represent a circuit with

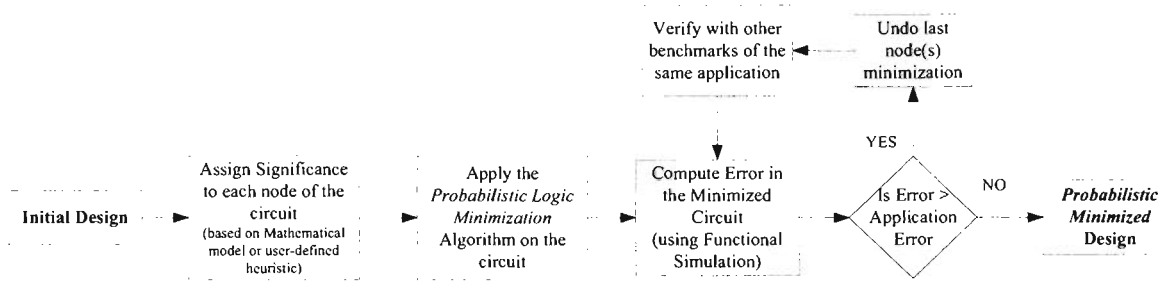


Figure 6.2 : Flowchart to obtain a *Probabilistic Logic Minimized* Design

N nodes and W edges. For any given node i in the graph, we have

- node.function(i) – denotes that function computed by node i,
- node.significance(i) – denotes the significance of node i, the assignment of which is described in section III(c)
- node.fanin(i) – denotes the fanin of node i,
- node.inputprobability(i)(j) – denotes the transition probability of input combination j occurring at node i. The range of values of j are 0 to $2^{fanin} - 1$
- node.functionmin(i) – denotes the minimized function of the node
- node.valuemin(i) – denotes the "value" or normalized cost gains of the minimized function

While function, significance and fanin values of all nodes are derived from the graph structure and user input, the inputprobability, functionmin and costmin are computed during the algorithm execution.

Algorithm 2 Pseudo-code for the *Probabilistic Logic Minimization* (PLM) algorithm on a Circuit or Graph $\mathcal{G}$

```
//Main Function in the Algorithm
function PLM(MaxError)
    //Compute the probability of each input transition at all nodes
    Benchmark( );
    //Compute the most cost-effective minimization at each node
    for all i ← 1 to N do
        ComputeMinimization(node(i));
    end for
    //Iteratively minimize each node based on their "value" until the error bound is
reached
    while Error ≤ MaxError do
        NodetoMinimize = FindMinimum (node.significance × node.costmin);
        MinimizeNode(NodetoMinimize);
    end while
end function

function BENCHMARK
    RunBenchmark();
    for all i ← 1 to N do
        for all j ← 1 to 2^fanin − 1 do
            node.inputprobability(i)(j) = ComputeInputProbability;
        end for
    end for
end function
```

Algorithm 3 Pseudo-code for the *Probabilistic Logic Minimization* (PLM) algorithm on a Circuit or Graph $\mathcal{G}$

function COMPUTEMINIMIZATION(node)

for all $j \leftarrow 1$ to $2^{fanin} - 1$ **do**

//Estimate the gains obtained by the bit flip at the input sequence j in the K-Map through synthesis tools

costgain(j) = **EstimateCostGain**(bitflip(j));

if $costgains(j) > 0$ **then**

ValueofBitFlip(j) = $\frac{costgain(j)}{inputprobability(j)}$;

else

ValueofBitFlip(j) = 0;

end if

end for

MaxValue = **FindMaximum**(ValueofBitFlip(j)):

functionmin $\leftarrow$ **ComputeFunction**(MaxValue);

costmin $\leftarrow$ MaxValue;

end function

function MINIMIZENODE(node)

function $\leftarrow$ functionmin;

end function

6.3 Advantages of Probabilistic Logic Minimization

To the best of our knowledge, the proposed probabilistic logic minimization technique is the only other architectural design technique (after our probabilistic pruning tech-

nique) targeted for Inexact systems for error tolerant or resilient approaches. The advantages of the proposed probabilistic logic minimization technique are multifold (most of them similar to the probabilistic pruning technique but listed here nevertheless for the sake of completeness):

1. As the technique is used to realize or "synthesize" inexact circuit architectures to start with, it has *zero overhead* on the circuit hardware in terms of energy, delay and area. In other words, the proposed technique obtains savings in all 3 dimensions - energy, delay and area - when compared to its conventional correct counterparts.

2. Since, it is an architectural level implementation, the proposed technique guarantees a bound on the error (average or worst case) for the inexact circuit realization unlike the circuit level scaling techniques (such as voltage scaling). This can be attributed to the fact that the proposed technique is independent of parameter (such as supply voltage) variation and hence, resilient to metastability or timing failures and doesn't have a critical (voltage scaled) point that might cause massive failures [35]!

3. The proposed technique doesn't have the overheads of level shifters, multiple voltage planes or metastability-tolerant latches, typically needed for the operation of circuit level voltage-scaling based design techniques

4. The proposed technique is technology independent as the amount of gains are only proportional to the amount of nodes minimized and not on the process technology parameters (will be illustrated in Chapter 9) as opposed to the voltage-scaling based schemes in which the amount of gains is limited by the process technology constraints. For example, in the present day deep submicron

CMOS technology nodes (65nm and below), the supply voltage is typically around 1V and the threshold voltage is around 0.3-0.5V. Hence, the amount of voltage scaling that could be done is very limited and so are the gains.

5. Another advantage in applying the proposed approach to XOR-dominated (datapath) circuits widely prevalent in most applications is that traditional logic synthesizers do a lousy job in minimizing XORs [65] whereas through our logic minimization algorithm, the logic synthesizers can extract further savings as the minimized function is most likely to have primitive gates as opposed to "costly" XORs (as will be shown in Chapter 8).

6. For the present day and future deep submicron CMOS technologies (45nm and below), leakage power forms a significant portion of the total power consumption. The proposed technique by virtue of its significant reduction in the total number of "leaky" transistors reduces the total leakage power of the system as well.

7. Lastly, the proposed technique can be easily integrated into the traditional system based CAD flows (as shown in Chapter 9), thereby reducing the design effort and time as opposed to most of the circuit-level design techniques, in particular the Biased Voltage Scaling (BiVOS) proposed in [31, 36], that require a custom design flow for (physical) implementation.

Chapter 7

Analysis of gains from Proposed Techniques

Today, the most prevalent and widespread transistor technology for building VLSI circuits is the Complementary Metal Oxide Semiconductor (CMOS) which replaced the previously popular Bipolar Junction Transistors (BJTs) and the NMOS transistors. The main advantages the CMOS technology over its counterparts was its negligible static power dissipation (no current flow when the transistors are not switching) and high noise immunity. These advantages coupled with the possibility of high density integration of the CMOS transistors have been the driving reason behind the success of the VLSI industry. In this chapter, we give a brief overview of the underlying models of the energy consumption and delay of CMOS circuits and analyze the reason behind the gains achieved by the proposed techniques.

7.1 Energy Consumption of CMOS Circuits

The main source of energy consumption in CMOS circuits is the switching energy (due to charging and discharging of the capacitance nodes) and the short circuit energy (due to momentary creation of a path enabling current flow from supply voltage (V_{dd}) and ground (V_{ss}) during a node transition) [66, 67, 68]. However, the constant scaling of transistor size driven by the Moore's law and the corresponding reduction of the supply voltage (V_{dd}) needed to maintain a constant electric field strength on the transistor resulted in elevating the previously negligible static energy consumption

to significant levels, particularly in the ultra-deep sub-micron (UDSM) technologies (90nm and below feature sizes). The two main reasons contributing to the increasing static energy consumption are : reduction in the threshold voltage (V_{th}) corresponding to a decrease in the supply voltage (V_{dd}) to maintain an acceptable speed of operation and new electrical effects (often known as the *short channel effects*) arising due to the reduction of the geometrical dimensions of the transistors.

In the following subsection, a brief description of various sources of energy consumption in a CMOS based VLSI circuit is given:

$$E_{total} = E_{dynamic} + E_{static} \tag{7.1}$$

$$= E_{switching} + E_{short-circuit} + E_{static} \tag{7.2}$$

7.1.1 Dynamic/Active Energy Consumption

Dynamic or Active energy consumption occurs during the operation of the circuit i.e. when the circuit nodes are switching due to charging and discharging of the load capacitances as well as the internal node capacitances. The dynamic power consumption has two components : (1) *Switching Energy* which corresponds to the energy required to charge and discharge the node capacitances during transitions, (2) *Short-Circuit Energy* which corresponds to the energy dissipation due to the formation of a conductive path, for a short period of time, between V_{dd} and V_{ss} during the switching of the transistor.

The switching energy consumption of a circuit is given by Equation 7.3.

$$P_{switching} = 1/2 \cdot C_{eff} \cdot V_{dd}^2 \tag{7.3}$$

where C_{eff} is the effective switching and V_{dd} is the supply voltage.

The Short-circuit energy dissipation of a circuit as shown in Equation 7.4 occurs due to direct path currents owing to the non-zero rise and/or fall times of the inputs and are more prominent when the input rise/fall time are much larger than the output rise/fall time. It has been observed that typically, the short-circuit energy only accounts for less than 10% of the total dynamic power consumption and the ratio of the short circuit energy to the dynamic energy is inversely proportional to the ratio of threshold voltage to the supply voltage of the transistors [69].

$$E_{short-circuit} = I_{peak} \cdot \left(\frac{t_r + t_f}{2} \right) \cdot V_{dd} \tag{7.4}$$

I_{peak} is the peak current which is determined by the saturation current of the transistors and hence, is directly proportional to the sizing of the transistors. t_r and t_f are the rising time and falling time of the short circuit current respectively.

7.1.2 Static or Leakage Energy Consumption

The static or leakage energy dissipation occurs when the transistors are in "off-mode" and hence, can be regarded as the energy dissipated without any useful outcome. The static energy dissipation of a circuit is given by Equation 7.5.

$$P_{static} = I_{static} \cdot V_{dd} \tag{7.5}$$

where I_{static} is the current flowing through the transistor in off-mode and V_{dd} is the supply voltage.

Generally, there are three types of static or leakage energy dissipation : (a) Sub-threshold leakage, (b) Gate leakage and (c) Reverse-biased drain-substrate and source-substrate junction band-to-band tunneling (BTBT). Sub-threshold leakage occurs as a result of weak-inversion conduction (when the supply voltage is less than the threshold voltage) of the CMOS transistor and increases exponentially as the

threshold voltage increases. Reverse-based junction currents (dependent on junction currents and doping concentration) cause the reverse-based drain or source-substrate junction BTBT leakage. Gate leakage occurs due to the tunneling current flowing through the gate of the transistor due to the scaling of gate oxide thickness.

7.2 Propagation Delay in CMOS Circuits

The propagation delay t_p of a CMOS inverter is given by Equation 7.6 [67].

$$t_{pHL} = 0.69 R_{eqn} C_L$$

$$t_{pLH} = 0.69 R_{eqp} C_L$$

where t_{pHL} and t_{pLH} are the propagation delays for high to low and low to high transitions respectively, and R_{eqn} and R_{eqp} are the equivalent resistances of the NMOS and PMOS transistors respectively.

$$\begin{aligned} t_p &= \frac{t_{pHL} + t_{pLH}}{2} \\ &= 0.69 C_L \left(\frac{R_{eqn} + R_{eqp}}{2} \right) \end{aligned} \tag{7.6}$$

Expanding the R_{eq} of the NMOS or PMOS transistor yields Equation 7.7 for t_{pHL} and a similar equation for t_{pLH} as well.

$$\begin{aligned} t_{pHL} &= 0.69 \frac{3}{4} \frac{C_L V_{dd}}{I_{DSATn}} \\ &= 0.52 \frac{C_L V_{dd}}{(W/L)_n k_n V_{DSATn} (V_{dd} - V_{tn} - V_{DSATn}/2)} \end{aligned} \tag{7.7}$$

where V_{tn} and V_{DSATn} represent the threshold voltage and the saturation voltage of the NMOS transistor respectively and the rest represent the technological parameters of the transistor. Hence, we can observe that the propagation delay of a CMOS

transistor is inversely proportional to supply voltage V_{dd} as shown in Equation 7.8.

$$\begin{aligned} t_p &\sim \frac{V_{dd}}{V_{dd} - V_{te}} \\ &= \frac{1}{1 - (V_{te}/V_{dd})} \end{aligned} \tag{7.8}$$

where $V_{te} = V_t - V_{DSAT}/2$.

7.3 Analysis of gains obtained by the proposed techniques

From the previous section, it was evident that the most dominants aspects of energy consumption in CMOS circuits are switching energy and for ultra-deep submicron technologies (90 nm and below), the static/leakage energy consumption. From Equation 7.3, it can be observed that the only ways to reduce the energy consumption of a circuit would be to reduce the net effective switching capacitance C_{eff} per clock cycle or by lowering the supply voltage V_{dd}. Effective switching capacitance is generally lowered by using less logic, smaller devices and fewer and shorter interconnects. Lower of supply voltage leads to a quadratic reduction in the switching energy consumption and also, a linear reduction in leakage power consumption. Hence, most of the circuit-level techniques use supply voltage scaling as a means to lower energy energy consumption of CMOS based systems. However, it should be noted that the scaling of supply voltage will have lead to a increased propagation delay in the circuit as evident from Equation 7.8. Therefore, the circuit designers try to find an optimal operating point based on the Energy-Delay product metric or rely on the application needs to optimize for energy or delay.

While this circuit-level (rather transistor level) gains obtained by scaling of supply voltage do translate to proportional system level gains, realizing them in the context of smaller circuits (such as datapath elements or DSP blocks) critical in most error-

tolerant or resilient applications will not translate into substantial gains given the overhead associated with realizing such scaling as mentioned in Chapter 4.

On the other hand, several architecture-level techniques such as resource sharing [70, 71], logic minimizations [72, 73] have been proposed for conventional circuits. These techniques concentrated on reducing the C_{eff} of the overall circuit owing to the decreased number of transistors/gates in the circuit, thereby achieving savings in energy, delay as well as area. However, no such architecture-level optimizations have been attempted for the inexact circuits and this thesis is a first attempt to show the significant impact that architectural redesign techniques could obtain with the accuracy tradeoff.

Chapter 8

Application of Proposed Techniques to Datapath Elements

We choose datapath elements as the first platform for the application of our *Probabilistic Logic Minimization* technique as they are one of the most energy consuming blocks in the targeted error tolerant applications (for example, power consumption of the datapath elements accounts for upto 75% of the total motion estimation block [32]). The main datapath elements commonly used in most applications are arithmetic adders and multipliers, and hence, they will be the prime focus of our study.

8.1 Arithmetic Adders

Binary addition is a fundamental and most frequently used arithmetic operation on microprocessors, digital signal processors and other data-processing ASICs. Apart from their use in binary addition, adders are also used in more complex arithmetic operations like multiplication and division, and also simpler operations like incrementation and magnitude comparison. Many different circuit architectures for binary addition have been proposed over the last 50 years [74, 75, 76, 77, 78, 79] and the usability of these architectures based on the specific constraints (delay/area/fanout/wiring tracks) has been extensively studied [80, 81].

A binary adder adds two n-bit operands, $A = (A_{n-1}\, A_{n-2} \ldots . A_1\, A_0)$ and $B = (B_{n-1}\, B_{n-2} \ldots B_1\, B_0)$ along with an optional carry-in C_{in} and outputs an n-bit Sum

$S = (S_{n-1}\, S_{n-2}\, ...S_1\, S_0)$ and a carry-out C_{out} governed by the equation:

$$A + B + C_{in} = S + 2^n \cdot C_{out}$$

8.1.1 A Retrospect of Conventional Adder Designs

It is widely known that binary addition can be formulated as a prefix problem i.e. every output is dependant on all inputs of equal or lower magnitude, and every input influences all outputs of equal or higher magnitude. In a prefix circuit, N outputs $(y_{n-1}, y_{n-2}, ..., y_1, y_0)$ are computed from N inputs $(x_{n-1}, x_{n_2}, ..., x_1, x_0)$ using an arbitrary associative binary operator '$\circ$' as follows:

$$\begin{aligned} y_0 &= x_0 \\ y_1 &= x_1 \circ x_0 \\ y_2 &= x_2 \circ x_1 \circ x_0 \\ &\vdots \\ y_{n-1} &= x_{n-1} \circ x_{n-2} \circ \cdots \circ x_1 \circ x_0 \end{aligned}$$

Formulating this problem recursively yields :

$$\begin{aligned} y_0 &= x_0 \\ y_i &= x_i \circ y_{i-1} \;\; ; \;\; i = 1, 2, 3, \cdots n-1 \end{aligned}$$

Owing to the associativity of the '$\circ$' operator, the sequence of operations and grouping of bits can be carried out in any order resulting in a large number of possible architectures ranging from serial to highly parallel ones.

Based on this prefix logic, the functioning of a binary adder can be grouped into 3 stages as shown in Figure 8.1.

$$\textbf{Pre Computation: } G_{i:i} = A_i \cdot B_i \quad ; \quad G_{0:0} = C_{in}$$

$$P_{i:i} = A_i \oplus B_i \quad ; \quad P_{0:0} = 0$$

$$\textbf{Prefix Computation: } G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j}$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

$$\textbf{Post Computation: } S_i = P_i \oplus G_{i-1:0}$$

While the precomputation and postcomputation stages are common for almost all adder architectures, the prefix computation stage generally defines the architecture of an adder. The prefix networks of some common adders are shown in Figure 8.2 and the different type of nodes used in a prefix network are described in Figure 8.3. Depending on the order of grouping of bits, the prefix stage is generally classified into 3 types [80] : (a) Serial Prefix (b) Group Prefix (c) Parallel Prefix. As the names suggest, serial prefix structure has a serial propagation of carry, group prefix has a semi-parallel propagation of carry and parallel prefix structures have a highly parallel carry propagation structure. A brief overview of the characteristics of various conventional adders is given in Table I. While most of the previous retrospects concentrated more on speed Vs area tradeoffs/metrics, we incorporate power consumption metric into our adder analysis as well.

8.1.2 A Brief Mathematical Overview of Carry Path Probabilities in an Adder

The analysis of carry propagation in adders is one of the oldest problems in the analysis of algorithms [82]. We use some of the results derived in [82] to model the carry propagation path probabilities which will form the basis of the *pruning* technique. For notational convenience, we will use the symbols S, A and B to denote the Sum

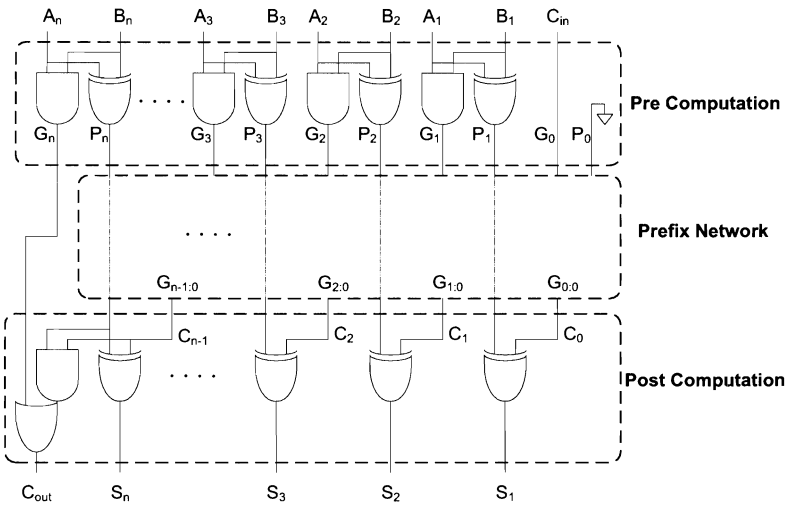


Figure 8.1 : A General Architecture of an Adder based on Prefix Logic

Table 8.1 : Summay of Various Conventional Adder Architecture Characteristics

Type of Adder	**Area**	**Speed**	**Power**	**Prefix Type**
Ripple Carry	Lowest	Lowest	Lowest	Serial
Carry Skip	Low	Low	Low	—
Carry Select	Medium	Medium	Medium	Group
Carry Increment [80]	Medium	Medium	Medium	Group
Sklansky [74]	High	Highest	High	Parallel
Brent-Kung [77]	High	High	High	Parallel
Kogge-Stone [75]	Highest	Highest	Highest	Parallel
Han-Carlson [78]	High	Highest	High	Parallel
Ladner-Fischer	High	Highest	High	Parallel
Sparse Tree [79]	High	Highest	High	Parallel

(output) and the two binary inputs to the adder. As all the paths between output S_i and an input A_j or B_j, ($\forall j \neq i$ and $0 \leq j \leq N$) in existing in an N bit adder are due to the propagation of carry bits, we compute the various path probabilities in an

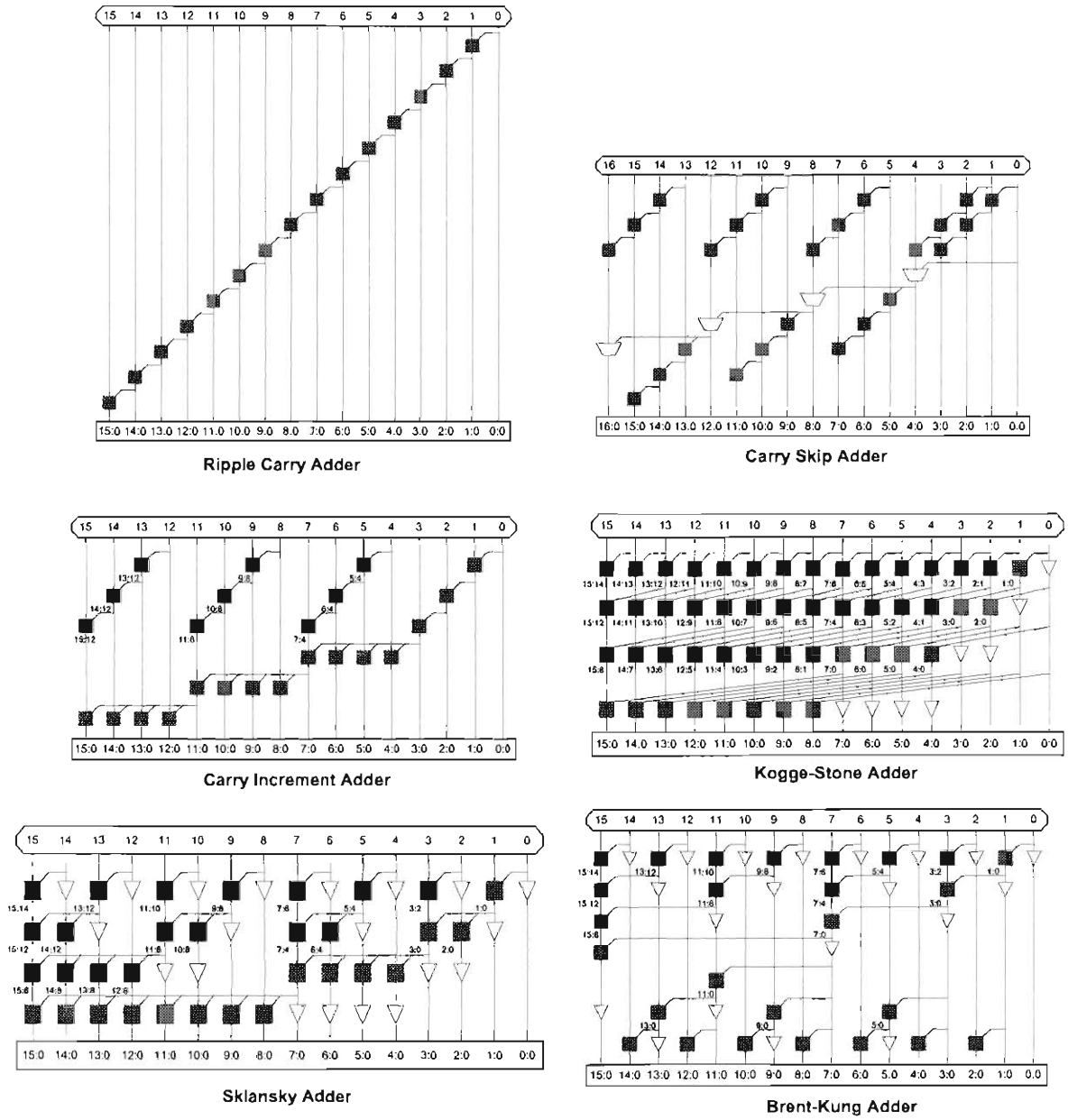


Figure 8.2 : Prefix Networks of Some Adders

adder using a variation of the carry path propagation results derived in [82], to form the basis of the *pruning* technique.

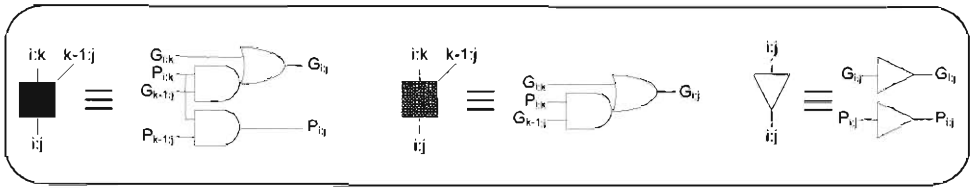


Figure 8.3 : The Composition of the Nodes in the Carry Paths of Prefix Adders

A bit position 'i' is said to *generate* a carry if both A_i and B_i are equal to 1 and *propagate* a carry if exactly one of A_i or B_i is equal to 1. Hence, a sum output S_i is affected by an input A_j or B_j (where $j < i$) only if there is a carry *generated* at j and the rest of the $i - j$ bits *propagate* the carry. For example, if the summands A and B are chosen uniformly at random, the probability that a bit position j *generates* a carry is $1/4$ and the probability that the rest of the $i - j - 1$ *propagate* the carry is $1/2^{i-j-1}$. Hence, the probability of any particular path from an input A_j or B_j to an output Sum S_i being active is $1/2^{i-j+1}$.

8.2 Multipliers

Aside from the adders, multipliers are the other fundamental building blocks present in most signal processing applications [83]. As they have a larger area, delay and energy footprint when compared to the adders, design of low-power and high-performance multipliers has been a subject of research for many decades [84].

The multiplication of a M-bit and N-bit numbers $X = \sum_{i=0}^{M-1} x_i \cdot 2^i$ and $Y =$

$\sum_{j=0}^{N-1} y_j \cdot 2^j$ is computed as

$$\begin{aligned} Z &= X \times Y \\ &= \sum_{k=0}^{M+N-1} z_k \cdot 2^k \\ &= \left(\sum_{i=0}^{M-1} x_i \cdot 2^i \right) \left(\sum_{j=0}^{N-1} y_j \cdot 2^j \right) \\ &= \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_i y_j 2^{i+j} \end{aligned}$$

8.2.1 A Retrospect of Conventional Multiplier Designs

The algorithm for multiplication consists of three steps : (a) Generation of partial products (GPP); (b) Reduction of partial products (RPP) and (c) Final Carry Propagate Addition (CPA) as shown in Figure 8.4. A wide variety of algorithms have been proposed to target the optimization of these stages [84].

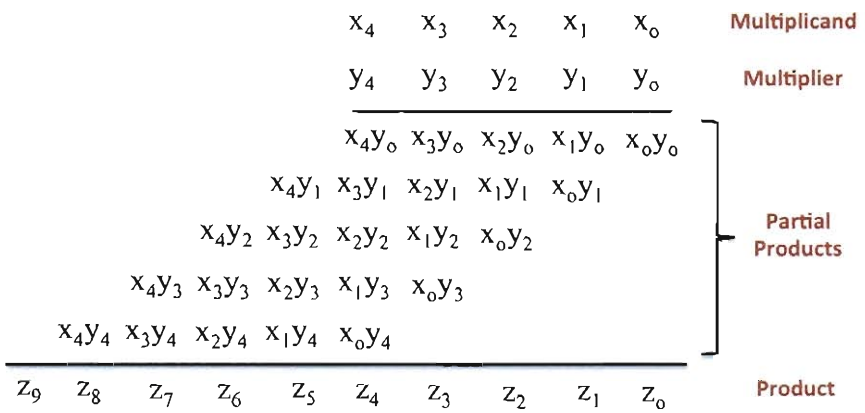


Figure 8.4 : The various states in a typical Multiplier

Various multiplier architectures have been proposed over the past few decades and they can be broadly classified into two types based on the type of partial product

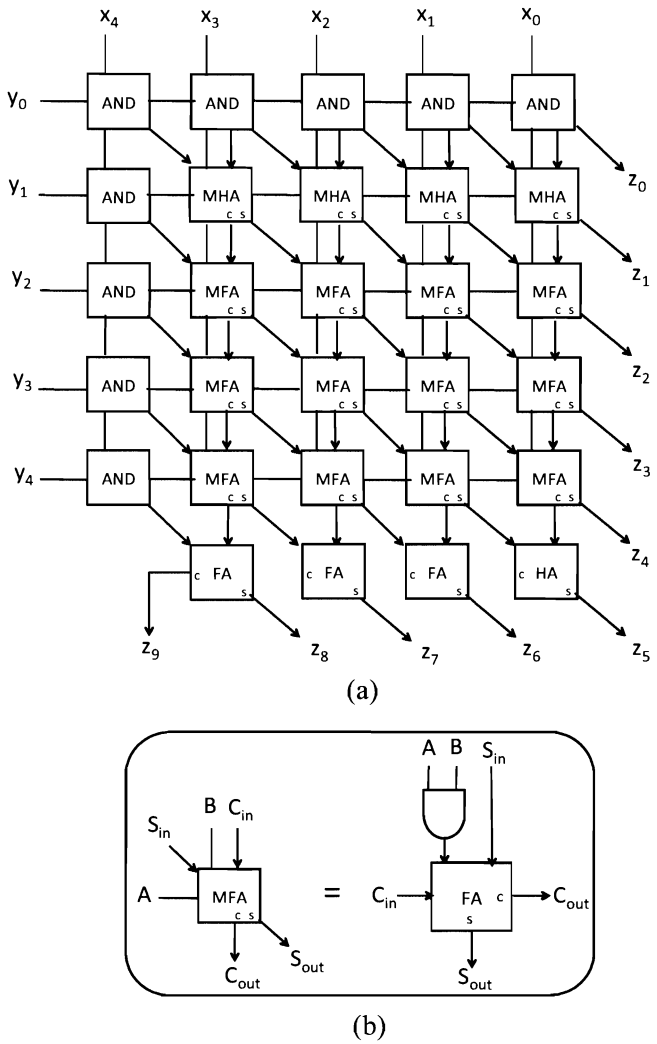


Figure 8.5 : The Architecture of an Array Multiplier

reduction tree: (a) *Serial Multipliers* which use a "shift and add" type of reduction of partial product tree thereby reducing the size of the partial product tree by one in each clock cycle; and (b) *Parallel Multipliers* which employ a more parallel reduction of the partial product tree using Carry Save Adders (CSA). In most applications, parallel multipliers are the obvious choice given the necessity for minimum performance [85].

The Parallel Multipliers are further divided into two types : (1) *Array Multipliers* which employ a network of full adders to reduce the partial product tree as shown in Figure 8.5 and (2) *Tree Multipliers* which employ special type of carry save adders called compressors (Wallace Tree Multipliers) and counters (Dadda Multipliers) to enable a complete parallel reduction of the partial product array [86].

8.3 Applying the proposed techniques on Datapath Elements

In this section, we provide some key information that would be helpful for applying and analyzing the application of proposed techniques on various datapath elements (adders and multipliers). The experimental results demonstrating the gains obtained by applying the proposed techniques on these datapath elements will be shown in Chapter 9.

8.3.1 Probabilistic Pruning based Datapath Elements

Applications for which all bit positions in an adder are treated with equal significance i.e. there is no notion of *Most Significant Bit* (MSB) or *Least Significant Bit* (LSB) and each bit position has equal significance. We use concepts from Section 8.1.2 to calculate the path probabilities in each adder and apply the probabilistic pruning technique as shown in Figure 6.2. Due to regular structure of the prefix networks, all components at the same level (or row) are on the paths with equal probability of being active. For example, the components on the 4^{th} level of the Kogge-stone adder are propagating carry information from inputs A_i and B_i to output S_{i+8} and while the components on the 3^{rd} level are propagating the carry information from inputs A_i and B_i to output S_{i+4}, the path probabilities of which are $1/2^9$ and $1/2^5$ respectively. Hence, we start pruning from the 4_{th} level till we reach an error bound

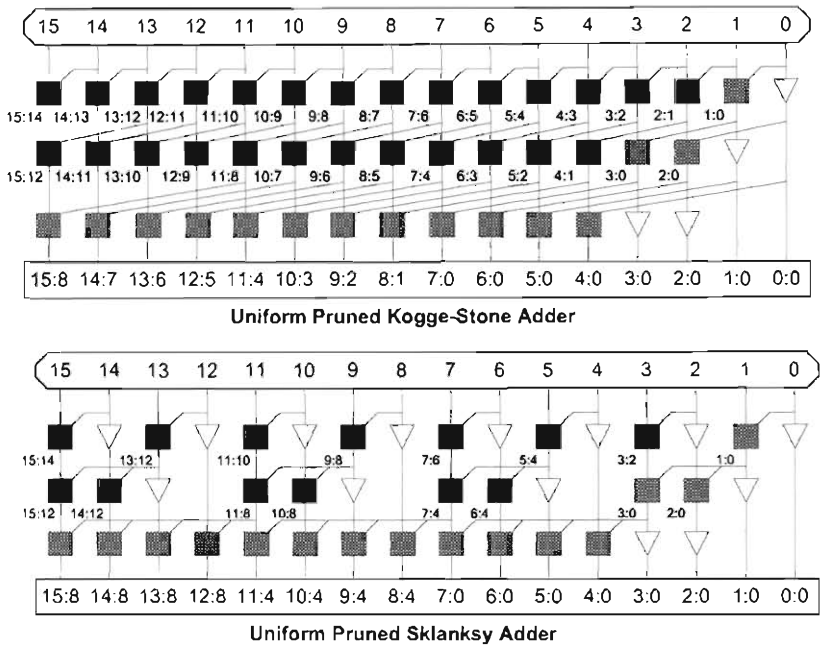


Figure 8.6 : Example of adder architectures designed using Uniform Probabilistic Pruning

of the application. Examples of 16-bit Uniform Pruned Kogge-Stone and Brent-Kung adders are shown in Figure 8.6. It should be noted that while all the outputs of a pruned Kogge-Stone adder have equal amount of carry propagation paths due to its *regular* structure, the outputs of other parallel adders (such as Brent-Kung) have varying number of carry propagation paths.

For applications where the bit positions of adders have unequal significance i.e. each bit position is twice more significant than its previous bit position (based on binary number system representation), we apply the probabilistic pruning with significance value assigned to each node (through heuristics such as those mentioned in Chapter 4) as shown in Figure 6.2. Starting from the LSB and moving towards the MSB, each bit position has a significance of 2 times higher than the previous bit position. While

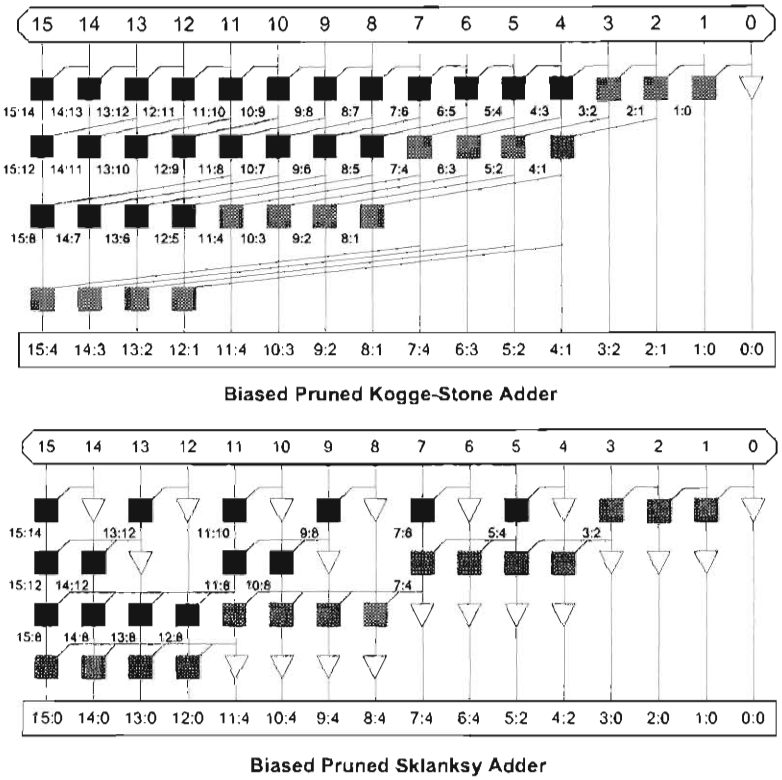


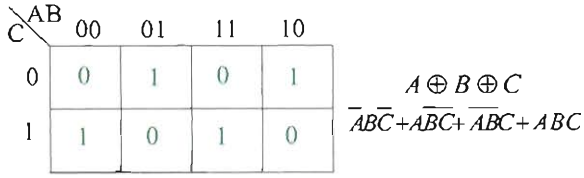
Figure 8.7 : Example of adder architectures designed using Weighted Probabilistic Pruning

it is possible to use the different significance values for each bit position through for the pruning technique, in this work for illustrative purposes , we "bin" the bits into four equal groups with each bin having k consecutive bits. We assume that the bits in the each bin have the same significance and are 2^k times as significant as those in the bin that is immediately following it. Applying our pruning technique, we compute the significance-activity product (SAP) and prune the components with the least product value. Examples of resulting 16-bit Pruned Kogge-Stone and Brent-Kung adders are shown in Figure 8.7.

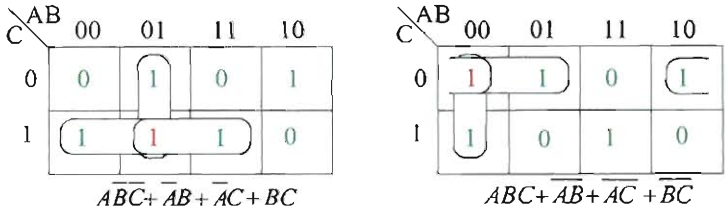
8.3.2 Probabilistic Logic Minimization based Datapath Elements

An example of a probabilistic logic minimization of a XOR-dominated logic (Sum logic of a full adder), widely prevalent in most datapath circuits, for different number of bit-flips is shown in Figure 8.8. To select the optimal bit flips, we will use the input combination probabilities at the full adder nodes. The results obtained for a simulation of different benchmarks on various full adder nodes in datapath elements is shown in Table 8.2. It should be noted that not all full adders used to construct the datapath elements such as array multipliers have similar input transition characteristics. The full adders receiving inputs directly from the circuit inputs are dependent on the application's input correlation while the full adders receiving the inputs from the outputs of other adders are more dependent on the topology of the circuit (specifically the sum and/or carry propagation paths). For the array multiplier, the full adders receiving the inputs directly from the partial products (AND-ed inputs) are denoted as external, the full adders present inside the partial product reduction matrix are denoted as internal and finally, the full adders present in the final carry propagate stage are denoted as CPA. Hence, *the probability of an input combination occurring at a node is generally either (a) only dependent on the input test vectors (such as full adders in ripple carry adder and external full adder in array multiplier) (b) only dependent on the circuit topology (such as internal full adder of array multiplier) (c) a combination of both (such as CPA full adder in array multiplier).*

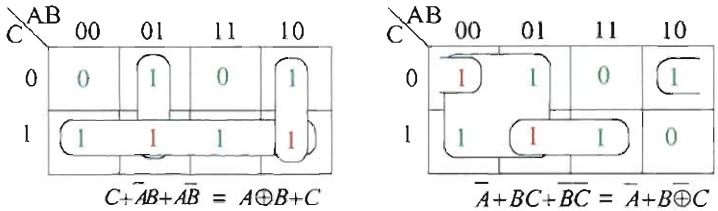
In general, this technique can be extended to higher order counters/compressors [86] with relative ease. While using parallel prefix adders [81], the choice of nodes can be varied between XOR gates in the initial propagate blocks and the PG-blocks in the prefix network tree. Also, it should be noted that, unlike conventional circuits, these datapath circuits are XOR-dominated circuits and traditional logic synthesizers



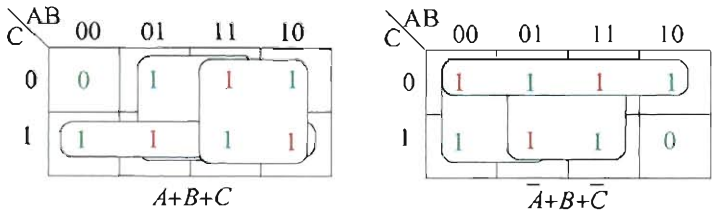
(a) K-Map of Sum Logic of a Full Adder



(b) K-Map of Sum Logic with 1 Bit-Flip



(c) K-Map of Sum Logic with 2 Bit-Flips



(d) K-Map of Sum Logic with 3 Bit-Flips

Figure 8.8 : Example of K-Maps of the Sum Logic of a Full Adder with various Bit-Flip configurations

do a lousy job in minimizing them [65]. Hence, apart from the advantage of savings obtained by our logic minimization algorithm, the traditional logic synthesizers can extract further savings as the minimized function is most likely to have primitive gates as opposed to "costly" XORs.

Table 8.2 : Input Combination Probabilities of Full Adders in various Datapath Elements

Test Vector Suite	Datapath Element	Probability of Various Input Combinations							
		000	001	010	011	100	101	110	111
Uniform	Ripple Carry Adder	0.129	0.121	0.125	0.121	0.125	0.126	0.13	0.124
	Array Multiplier (External)	0.542	0.024	0.167	0.017	0.145	0.04	0.025	0.041
	Array Multiplier (Internal)	0.349	0.047	0.079	0.048	0.314	0.051	0.056	0.055
	Array Multiplier (CPA)	0.388	0.088	0.082	0.01	0.218	0.092	0.091	0.03
Audio [87]	Ripple Carry Adder	0.258	0.02	0.133	0.12	0.129	0.121	0.014	0.205
	Array Multiplier (External)	0.5207	0.0004	0.2209	0.0002	0.2556	0.0005	0.0003	0.0014
	Array Multiplier (Internal)	0.394	0.029	0.041	0.048	0.309	0.032	0.038	0.109
	Array Multiplier (CPA)	0.272	0.073	0.148	0.001	0.291	0.126	0.085	0.004
Image [88]	Ripple Carry Adder	0.355	0	0.382	0	0.148	0	0.115	0
	Array Multiplier (External)	0.846	0	0.132	0	0.024	0	0	0
	Array Multiplier (Internal)	0.38	0.027	0.192	0.015	0.298	0.033	0.037	0.019
	Array Multiplier (CPA)	0.867	0.03	0.013	0	0.081	0.007	0.003	0

Some of the key observations from Table 8.2 are: (a) there is a strong correlation between the input vectors and the type of minimization that can be done at the node; (b) the amount of logic minimization (or the number of bit flips) that can be performed on a node can be determined by the corresponding grouping of input combination probabilities that are close to each other. For example, we could potentially group all the input combinations with values less than one or two standard deviations from the mean of the group and then, favorable bit flips can be done within this group to obtain the most amount of minimization. For example, in the audio benchmark,

input combinations {'001', '011', '101', '110'} for the external full adder of the array multiplier can be grouped together and favorable bit flips identified among them; (c) There can be a strong correlation between the application benchmarks and the possible minimizations. For example, for the image benchmark, input combinations {'001', '011', '101', '111'} can be flipped for full adders in ripple carry adder and array multiplier(external) without any error in the output!

Chapter 9

Experimental Results through Simulations

9.1 Methodology and Framework

The proposed logic-synthesis based CAD methodology for applying the *probabilistic pruning* and *probabilistic logic minimization* technique is based on Figure 9.1. The central object of interest in this CAD methodology is the Probabilistic Pruner/Logic Minimizer which has been seamlessly integrated into the traditional established CAD flow to design and fabricate integrated circuits.

In this CAD flow, the circuits (mostly datapath elements with varying bit-widths in this thesis) are described in a hardware description language (typically VHDL or Verilog) and then, synthesized using industrial logic synthesizers such as Synopsys Design Compiler or Cadence RTL Compiler. This synthesized design is then sent to the *Probabilistic Pruner/Logic-Minimizer* which implements either Algorithm 1 or Algorithm 2-3 described in Chapters 5 and 6 respectively. This pruner/logic-minimizer interacts with an error estimator (implemented in either C/C++/Matlab) and a functional simulator (such as ModelSim) with an application specific benchmarks to determine the final pruned or logic minimized circuit. The application-specific benchmarks used in our simulations include uniform random distributions from Matlab (for generic applications), image test vectors from Mediabench [88] and audio test vectors from NCH Software [87]. The obtained pruned or minimized circuit will be synthesized once again to glean further savings(if any) and then, would be sent to the

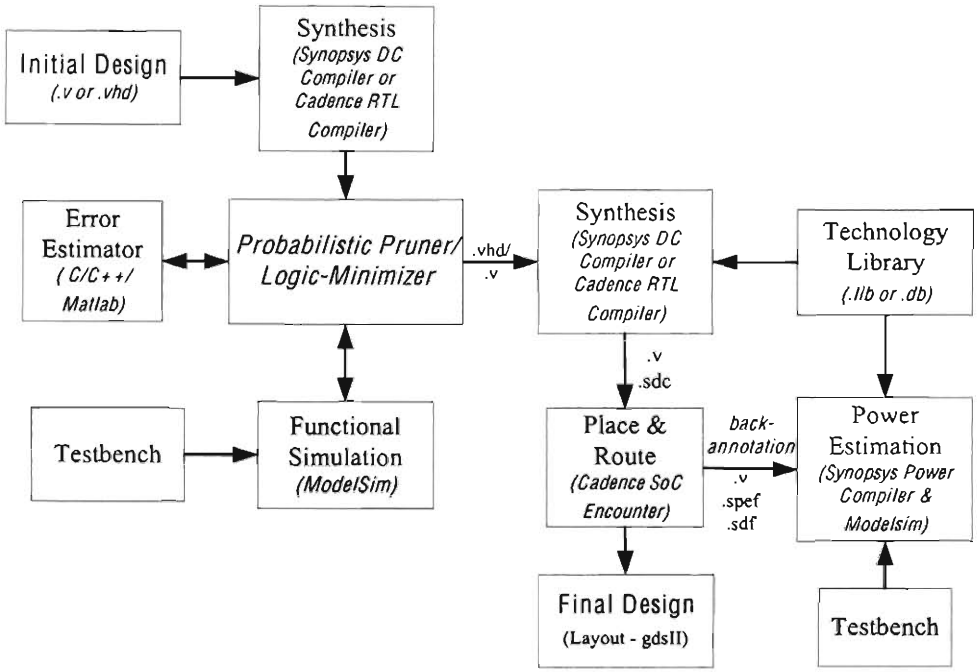


Figure 9.1 : Synthesis based CAD flow integrating the proposed architectural techniques

Place & Route Tools (such as Cadence SoC Encounter) to generate the final layout and the GDSII file that would be sent to the foundry for fabrication. We then perform the post-layout analysis of our resulting circuit by back-annotation of the final netlist and parasitics using Synopsys Power Compiler. The required toggling frequency of each node in the netlist and their corresponding input combination transition characteristics are obtained from the *Switching Activity Interchange Format* (.saif) file derived from the *Value Change Dump* (VCD) file. The VCD file is produced by simulation of the netlist along with the *Standard Delay Format* (SDF) file in ModelSim using the previously created benchmark test vectors.

To establish the technology-independent nature of our architecture-level design techniques, we have implemented the inexact circuits in a variety of CMOS technology

libraries including TSMC 65nm (High V_t) IBM 90nm (Normal V_t) and TSMC 180nm (Low Power). Also, the synthesis of designs was done targeting highest frequency of operation and lowest power consumption (or loose target frequency) separately, to analyze the gains achieved in each case.

9.2 Results and Analysis for Probabilistic Pruning

The central results, namely the normalized gains (Conventional/Pruned) for different metrics (area, delay, energy, energy-delay product and energy-delay-area product) obtained by applying probabilistic pruning technique on various 64-bit adders are summarized in Figure 9.2. From these graphs, it can be observed that: *The (normalized) gains obtained for the applications employing weighted circuits is more than the corresponding uniform circuits for the same error percentage.*

Figure 9.3 outlines the results obtained for a pruned Kogge-Stone adder using three different technology libraries under two different synthesis constraints. From this, we can conclude that: *For similar operating conditions, the gains achieved in the probabilistic pruned circuits is proportional to the ratio of circuit pruned to the original circuit. It is largely independent on the process technology being used and only depends on the logic synthesis constraints.*

Specifically, in addition to the observations highlighted through italicization in the paragraphs above that are of potential value to circuit design in general, we can also observe that probabilistic pruning is a design level technique which does not involve varying of circuit parameters during operation, *the amount of error in a probabilistic pruned circuit is independent of varying of parameters (such as V_{dd}) unlike other inexact circuits and is as robust as conventional circuits to process variations.* The amount of such error is generally fixed at design time based on application requirements. The

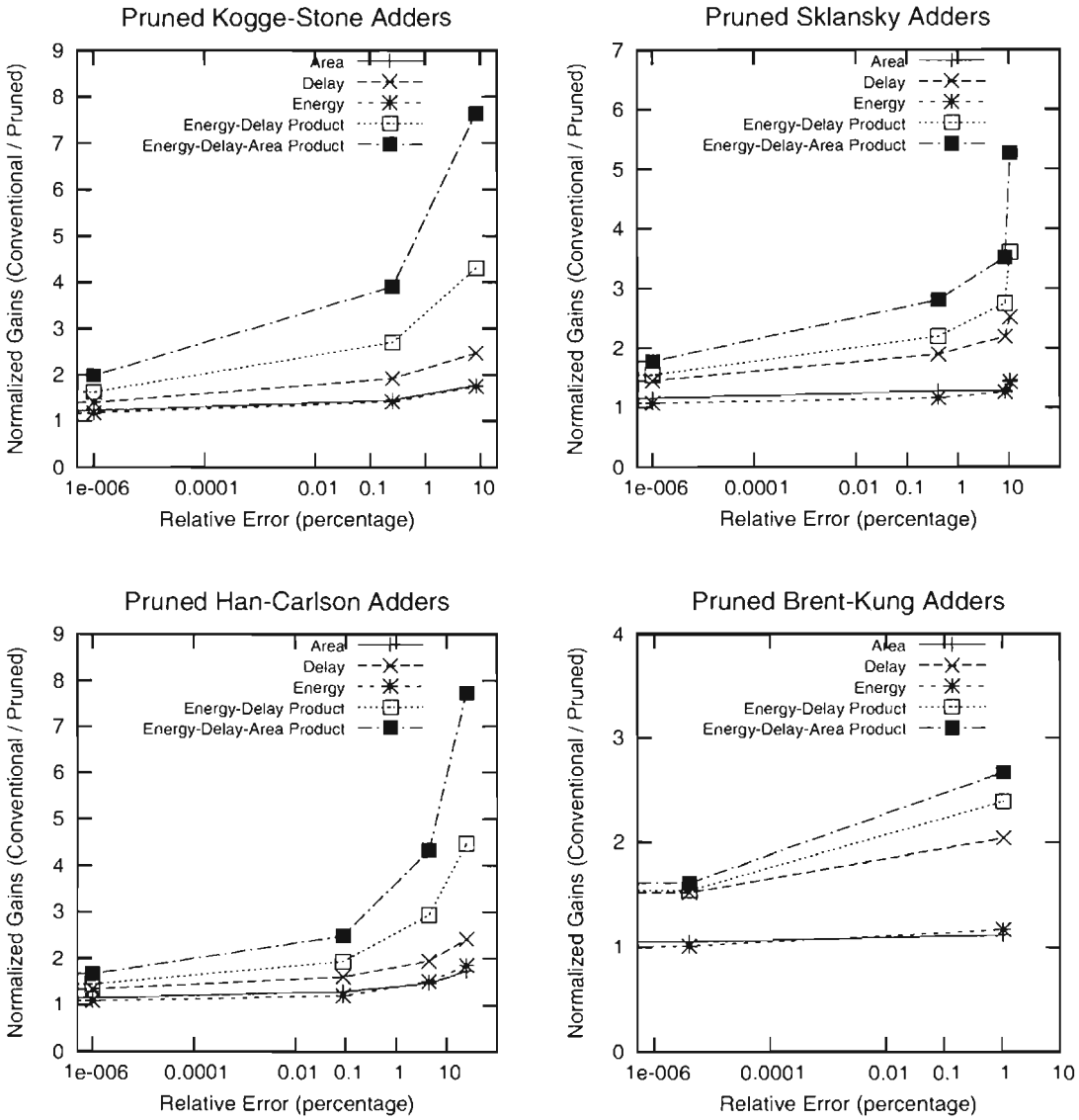


Figure 9.2 : Normalized gains Vs Relative Error percentage of various Probabilistic Pruned 64-bit adders

probabilistic pruning technique can be used in conjunction with techniques such as adaptive body bias [89] to address the effects of parameter variations in the more significant portions of the circuits.

Another observation regarding the probabilistic pruned circuits is that *the error*

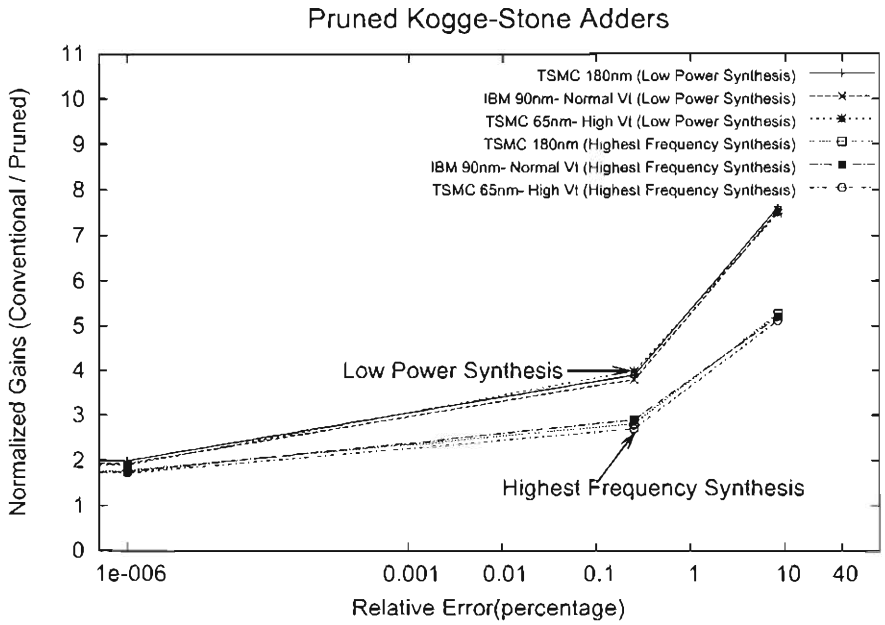


Figure 9.3 : Energy-Delay-Area Product of Weighted Pruned Kogge-Stone adders implemented in different process technologies and synthesis constraints

(both error rate and relative error magnitude) in probabilistic pruned adders rises sharply beyond a critical amount of pruning akin to the critical voltage scaling point problem mentioned in [34]. We anticipate that this can be fixed by combining a parameter variation approach (such as V_{dd} or V_t variation techniques) with our pruning technique.

9.2.1 Comparison to Precision Reduction or Bit-width Reduction

Precision Reduction or Bit-width truncation can be viewed as a special case of the probabilistic pruning algorithm in which (a) Significance of the truncated nodes can be assigned as zero, or (b) Activity of the truncated nodes will be zero. In other words, probabilistic pruning algorithm will yield closer to "optimal circuits than truncation for a given error budget. Our results indicate that the probabilistic pruning technique

outperforms the bit-width truncation by achieving 30-40% more cumulative gains in energy, delay and area for comparable error.

9.3 Results and Analysis for Probabilistic Logic Minimization

The central results, namely the normalized gains (Conventional/Proposed) values for different metrics – Energy, Energy-Delay Product (EDP) and Energy-Delay-Area product (EDAP) – obtained by applying the probabilistic logic minimizations for a 16-bit ripple carry adder and a 16-bit array multiplier for different application benchmarks are given in Figure 9.4 and 9.5 respectively. The choice of the bit-width here is governed by the fact that most of the targeted multimedia applications [32, 33] generally use bit-widths of 16-bits or less for datapath elements. However, we have also implemented other types of adders such as Carry-Select, Kogge-Stone and Sklansky and multipliers such as Wallace-tree and Dadda multipliers, with varying bit-widths (upto 64-bits) and for different application benchmarks, and have obtained similar gains.

As evident from the results, the probabilistic logic minimization approach results in highly energy, delay and area efficient datapath elements. For the uniform test vectors, in the case of ripple carry adders, probabilistic minimization yields savings upto 8X with an relative error of less than 1% compared their conventional correct counterparts while in the case of array multiplier, it resulted in savings of about 7X with a relative error of less than 6.5%. It can be seen that using application specific test vectors (like audio and image), the savings have increased (upto 9.5X in the case of ripple carry adders and upto 8.25X in the case of array multipliers) with comparable

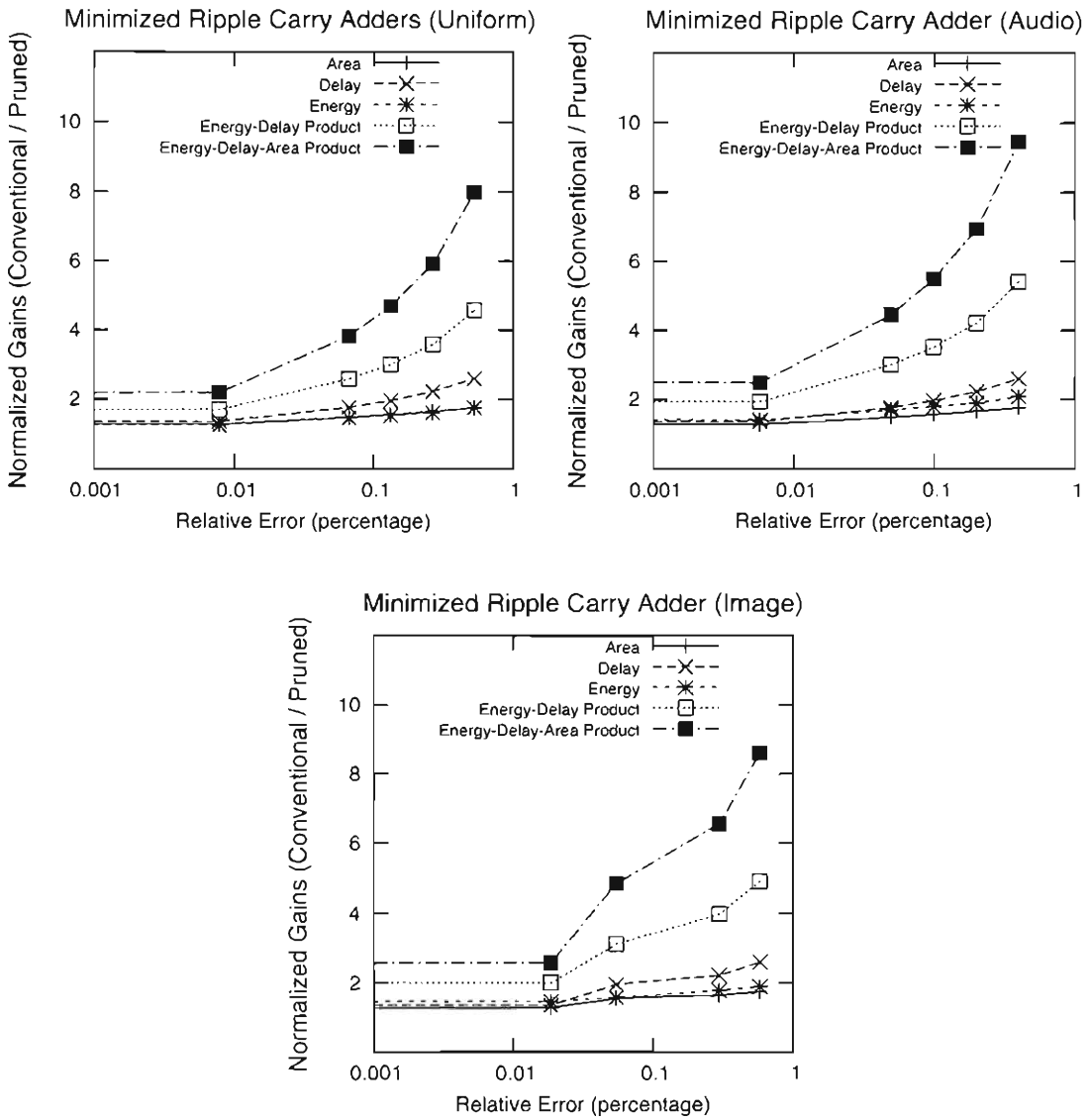


Figure 9.4 : Normalized gains Vs Relative Error percentage of minimized ripple carry adders for different benchmarks

error values.

To summarize, one of the key inferences from the simulation results is that : *the significant gains achieved in a circuit through the proposed probabilistic logic minimization technique are technology-independent, bit-width independent and only proportional*

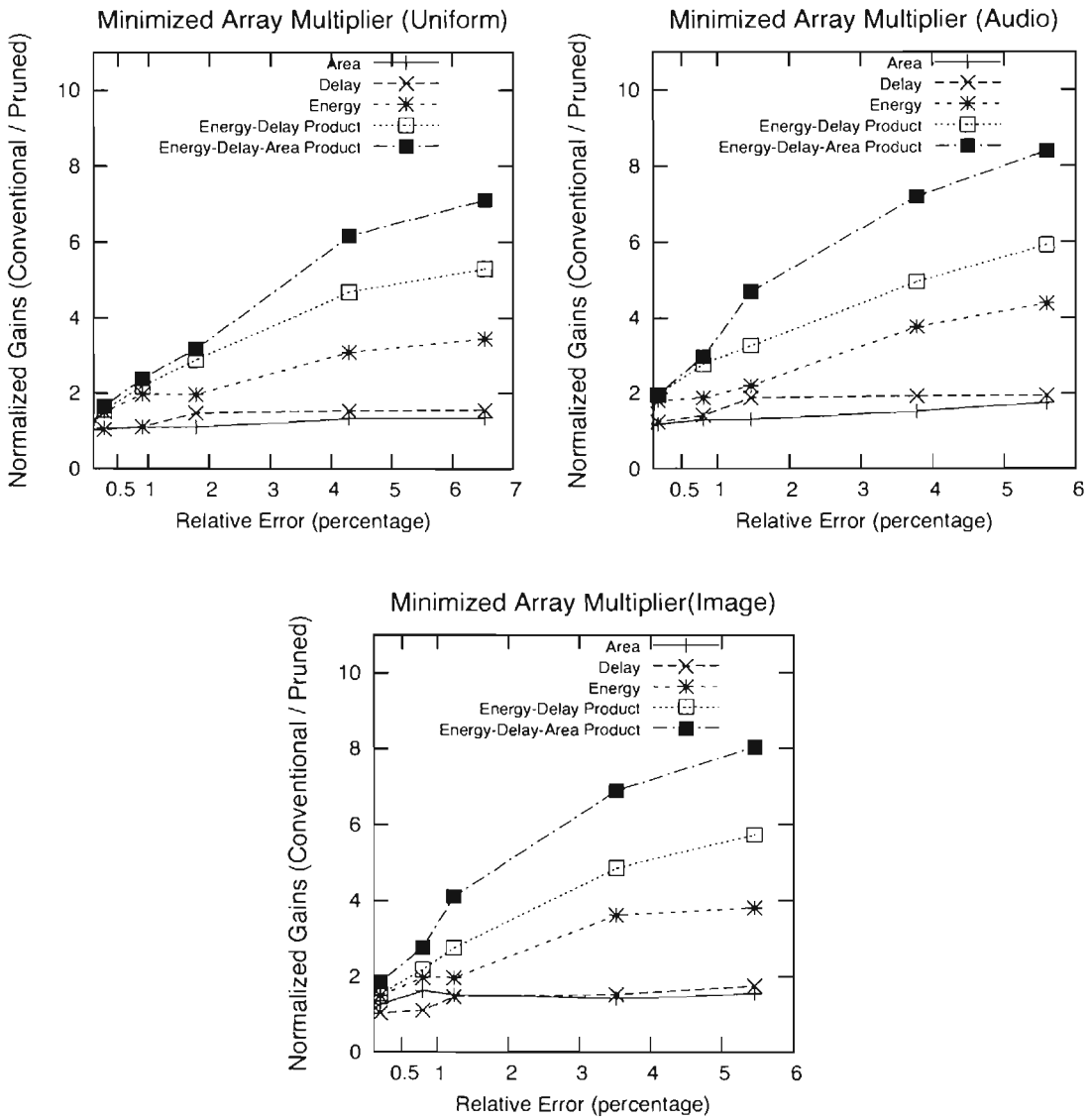


Figure 9.5 : Normalized gains Vs Relative Error percentage of minimized array multipliers for different benchmarks

to the amount of circuit nodes minimized.

Chapter 10

Physical Realization and Validation

10.1 Overview of the Test Chip Specifications

The Probabilistic Pruned Adders (PPA) chip consists of 30 adder circuits (of which 11 are conventional adders while the rest 19 are PPA) along with their peripheral circuitry such as pseudo random number generators. The main goal of the PPA chip is to characterize the power, delay and area values of these 30 adder circuits and establish the gains that can be achieved using PPA over conventional adders in the context of applications involving varying amounts of error tolerance.

10.1.1 Value of Key Features and Operating Conditions

A brief summary of the key features of the test chip is given in Table 10.1. It should be noted that the typical and maximum operating frequency parameters are constrained by the test equipment framework (in our case, *icy*board [90]) and do not necessarily the reflect the typical or maximum frequency of operation of individual adders. The operating conditions are described in Table 10.2, these are parameters the user may act on or has to guarantee.

10.1.2 Architecture of the Chip

The PPA chip circuit is illustrated in Figure 10.1. The functional modes of the PPA chip are controlled by the 6- Select bits ($S_5S_4S_3S_2S_1S_0$). Depending on the value of

Table 10.1 : Key numbers summary

Parameter	Comments	Min	Typical	Max	Unit
Technology	TSMC (low power CMOS)		180		nm
Chip Size			2.96		mm^2
Clock Rate	At VDDC = 1.8V	1	66	200	MHz

Table 10.2 : Operating Conditions summary

Parameter	Comments	Min	Typical	Max	Unit
Operating Temperature		0	25	50	$°C$
VDDIO Supply Voltage	IO supply voltage		3.3		V
VDDCore Supply Voltage	Core supply voltage	0.8	1.8	2	V

the select bits, a demultiplexer is used to choose a particular adder to characterize. The following table summarizes the list of various possible select bit values and the corresponding adder selected.

10.1.3 Power Domain Management

The PPA chip has two power domains as evident from Figure 10.1.

- **Power Domain 1 (PD1)** : This power domain consists mainly of the peripheral circuitry which felicitates the measurement of properties of the probabilistic adders but whose power consumption measurement is not required. It consists of the IO pads along with the two pseudo random number generators and the registers of all the adders.

- **Power Domain 2 (PD2)** : This power domain consists of all the 30 (combi-

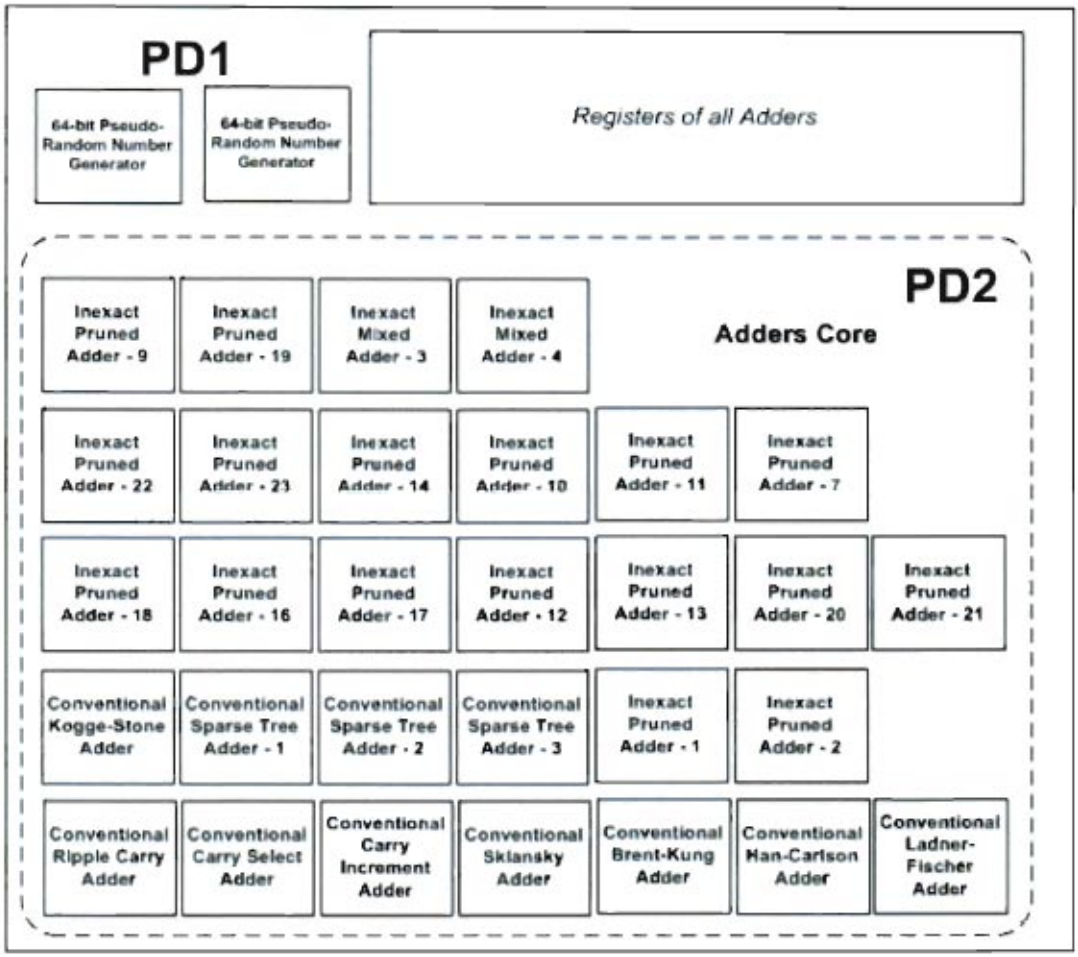


Figure 10.1 : Illustrative example showing the tradeoffs between energy and error involved in Inexact design for an image

national) adder circuits the measurement of whose characteristics (power, delay, area) is desired.

10.1.4 Clocks and Resets

The internal clock of PPA is provided by the clk_pad pad. The clock should have a 50% duty cycle and a nominal frequency of 66 MHz (for typical case operating conditions). However, in the tests the frequency of the clock can be varied and increased to as high

Table 10.3 : Functional modes based on select bits

Select Bits	Adder Selected	Select Bits	Adder Selected
000000	ALL OFF	010000	Pruned17
000001	RCA	010001	Pruned12
000010	CSLA	010010	Pruned 13
000011	CIA	010011	Pruned 20
000100	Sklansky	010100	Pruned 21
000101	Brent-Kung	010101	Pruned 22
000110	Kogge-Stone	010110	Pruned 23
000111	Han-Carlson	010111	Pruned 14
001000	Ladner-Fischer	011000	Pruned 10
001001	Sparse1	011001	Pruned 11
001010	Sparse2	011010	Pruned 7
001011	Sparse3	011011	Pruned 9
001100	Pruned 1	011100	Pruned 19
001101	Pruned 2	011101	Mixed 3
001110	Pruned 18	011110	Mixed 4
001111	Pruned 16	Others	All OFF

as 500MHz-1GHz to obtain the maximum frequency of operation characteristics of some of the adder circuits but the capabilities of the test equipment infrastructure (only handing upto a maximum of 200MHz) limits that possibility in our case.

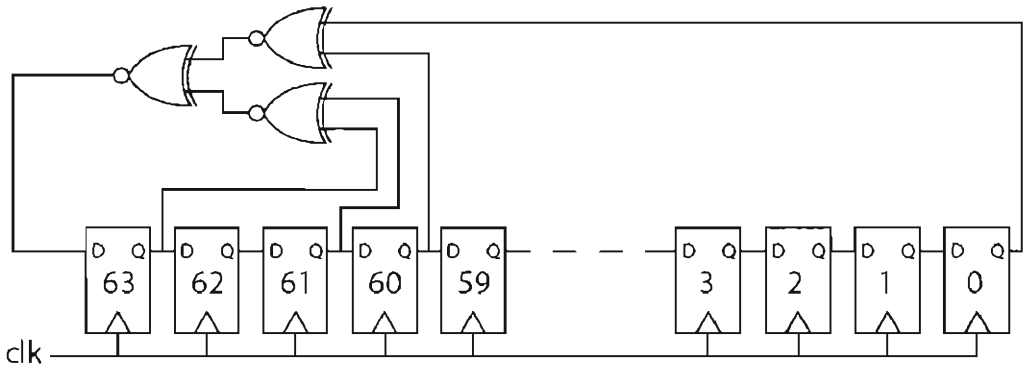


Figure 10.2 : Circuit diagram of the pseudo-random number generator using linear feedback shift register

10.1.5 Test Stimuli

The stimuli applied on the individual adders would be generated by the two on-chip 64-bit PRNGs present in PD1. The circuit of the PRNG which is implemented as a Linear Feedback Shift Register (and hence, has a uniform input distribution) is given in Figure 10.2. The initial seed value given to each of the PRNG are 1010101010101010101010101010101010101010101010101010101010101010 and 1100110011001100110011001100110011001100110011001100110011001100" respectively.

10.2 Physical view

The PPA die size is approximately 2.96 mm^2.

10.2.1 Technology

The circuit is fabricated in the TSMC 180 nanometer low-power CMOS technology using SESAME standard cell library. The circuit has 74 logical pads (of which 9 are

input pads and 65 are output pads), and 12 power/ground pads. The bonding diagram of the chip is shown in Figure 10.3 and the actual screenshot (from the GDSII layout) of the prototype chip is shown in Figure 10.4. As evident from this layout, the portion of particular interest for our results is the PD2(power domain 2) which houses the conventional and proposed pruned adders.

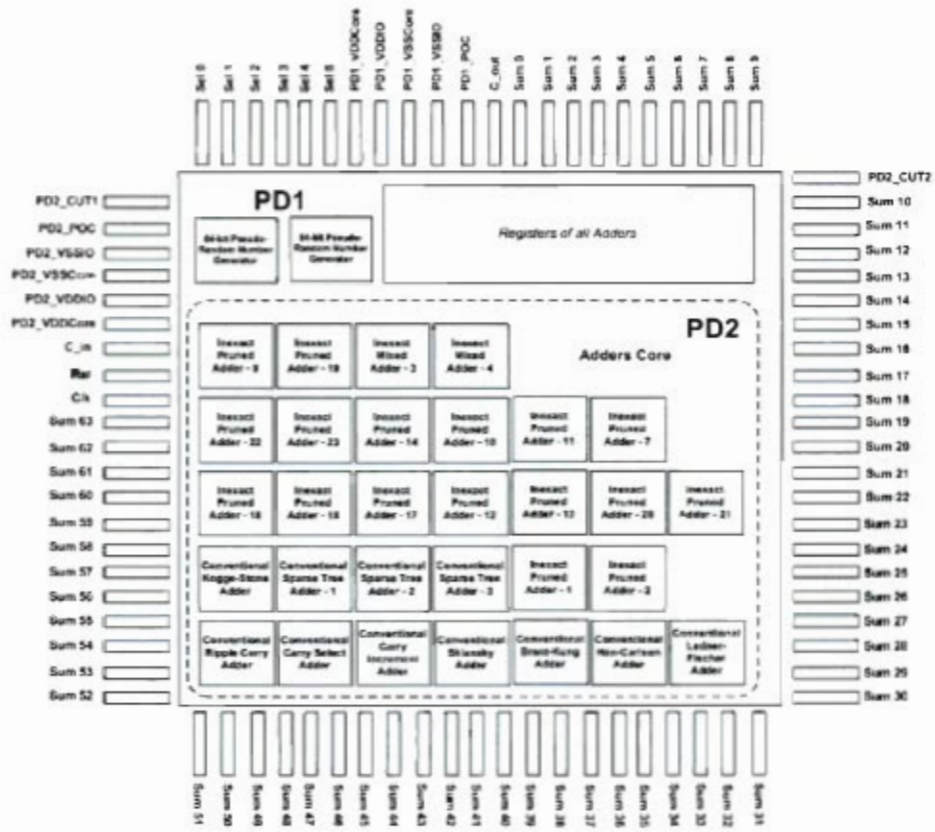


Figure 10.3 : Bonding diagram of the prototype chip

10.2.2 Testing Infrastructure

An important phase in the characterization of the prototype chip is its integration with the testing infrastructure. For the prototype chip, the *icy*board platform [90] has

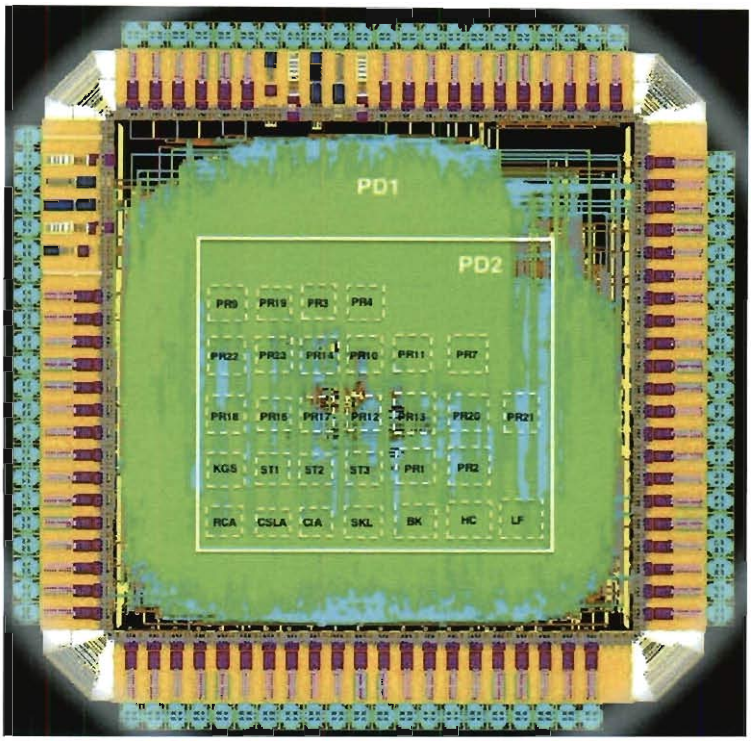


Figure 10.4 : A screenshot of the prototype chip layout

been chosen as the test platform of choice. The *icy*board platform offers features such as switches, LEDs, an RS232 port, a USB port for OCD, a USB port for data transfer, A/D converters (e.g. to measure power consumption) and an Altera MAX IIG CPLD (to route all the digital signals) for up to 3 daughter boards that can be mounted on it. The prototype chip has been bonded onto these daughter boards along with other discrete components such as a voltage regulator(to analyze the effects of voltage scaling). A schematic showing the layout and connections to the *icy*board platform is shown in Figure 10.5 and a snapshot of the integrated test infrastructure with the PCB (housing the pruning chip) mounted on the *icy*board test platform is shown in Figure 10.6. As evident from the figure, the PCB housing the chip also has several

probes to aid the power consumption measurement and for functional verification through a logic analyzer (such as the Saleae Logic Analyzer).

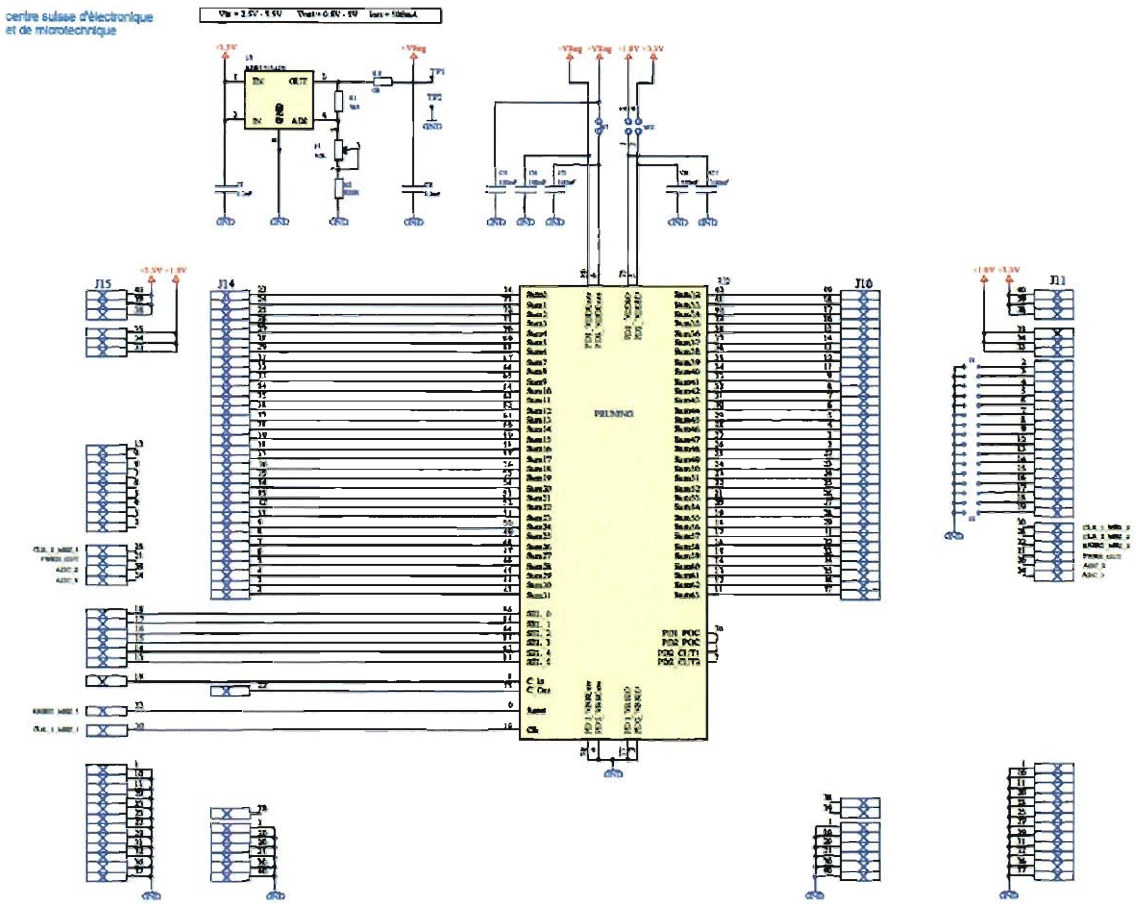


Figure 10.5 : PCB schematic of the prototype chip integrated with the icyboard platform. **Courtesy:** Pierre-Alain Beuchat, CSEM

10.3 Results and Analysis

The measured normalized gains (calculated as Conventional/Proposed) for different metrics such as area, delay, energy, energy-delay product and energy-delay-area product for varying relative error magnitude percentages from the prototype chip obtained

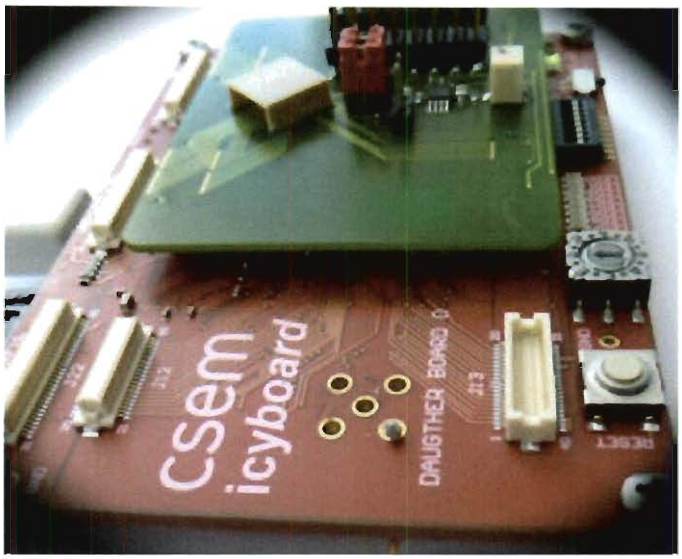


Figure 10.6 : A snapshot of the prototype chip integrated with the icyboard test platform

by applying probabilistic pruning technique on adders are shown in Figure 10.7. From Figure 10.7, it is evident that the pruning technique yields significant savings across all 3 dimensions – energy, delay and area, with the cumulative gains varying from 2X-7.5X for modest error values, conclusively validating the obtained gains through simulations and establishing the significant savings across all dimensions made possible by inexact architecture-level design techniques (such as probabilistic pruning) for acceptable amount of errors.

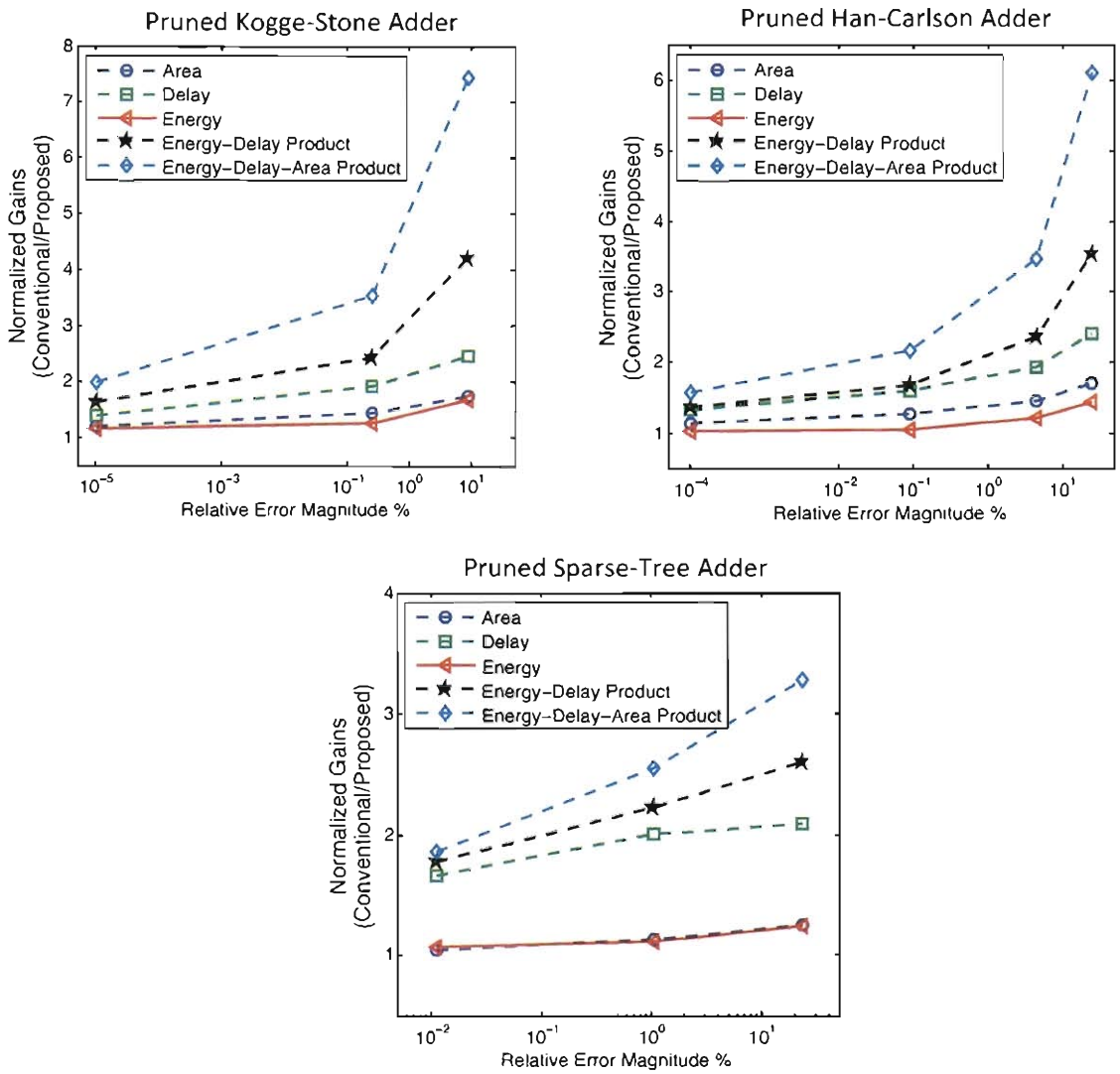


Figure 10.7 : Measured normalized gains Vs relative error magnitude percentage of various 64-bit adders from the prototype chip

Chapter 11

Conclusion and Future Directions

To the best of our knowledge, this is the first attempt at an architectural re-design of circuits to enable Error Vs Energy/Delay tradeoffs. As substantiation, we convincingly show through extensive simulations and through a prototype chip fabrication that our *Probabilistic Pruning* and *Probabilistic Logic Minimization* techniques achieve significantly better savings in all 3 dimensions – energy, delay and area – for the similar error tradeoffs than the conventional circuit-level voltage overscaling schemes. Other benefits of the proposed architecture-level *zero overhead* techniques include non-dependence of amount of error on any particular process technology, faster design time than conventional voltage scaling schemes as a result of easy integration into a logic synthesis based CAD flow and a gurantee to operate within the error bounds of the application at the design level (no timing-errors due to metastable states leading to massive failures).

These gains achieved through our proposed techniques are *relative* in that they can be combined with standard techniques that achieve energy or performance gains or both, through *absolute* approaches. Specically, this means that any technique that uses equal voltage (planes) throughout the datapath and yields correct results or voltage scaling to yield slightly incorrect results can be extended through the insights in this paper to yield *additional* gains simultaneously along the energy, delay and area dimensions by using the proposed architectural techniques.

We view this work as an early validation of a very general principle to datapath

design with the potential to enable novel applications. To start with, we believe that conventional algorithms for (computer) arithmetic and concomitant designs for signal processing will have to be revisited (particularly at the algorithm-level) and will result in innovations if *Inexact* design is considered. Second, we also expect architectural research building on the work of [91, 92] wherein a SoC approach to designing specialized media co-processors is outlined. We anticipate such co-processors as being eminently suited to being designed using the principles outlined in this thesis.

Bibliography

[1] C. Bronk, A. Lingamneni, and K. Palem, "Innovation for sustainability in information and communication technologies (ICT)," tech. rep., James A. Baker III Institute for Public Policy, Rice University, Oct 2010.

[2] A. Lingamneni, C. Enz, J.-L. Nagel, K. Palem, and C. Piguet, "Energy parsimonious circuit design through probabilistic pruning," *in the proceedings of the 14th Design, Automation and Test in Europe (DATE 2011)*, March 2011.

[3] A. Lingamneni, C. Piguet, K. Palem, and C. Enz, "Parsimonious circuit design for error-tolerant applications through probabilistic logic minimization," *in the proceedings of the International Symposium on Low power electronics and design (ISLPED) (under review)*, 2011.

[4] Energy Information Administration (EIA).

[5] Dragon Systems Software Limited, "Review of computer energy consumption and potential savings."

[6] J. Koomey, "Estimating total power consumption by servers in the u.s and the world," tech. rep., Lawrence Berkeley National Laboratory, February 2007.

[7] European Commission DG TREN, "Personal computers (desktops and laptops) and computer monitors." www.ecocomputer.org.

[8] TIAX LLC for the U.S. Department of Energy, "U.S. residential information technology energy consumption in 2005 and 2010," 2006.

[9] The Climate Group, "SMART2020: Enabling the low carbon economy in the information age," 2008.

[10] J. Koomey, C. Calwell, S. Laitner, J. Thornton, R. E. Brown, J. Eto, C. Webber, and C. Cullicott, "Sorry, wrong number: The use and misuse of numerical facts in analysis and media reporting of energy issues," *In Annual Review of Energy and the Environment*, pp. 119–158, 2002.

[11] D. Holtz-Eakin and T. Selden, "Stoking the fires? co_2 emissions and economic growth," *Journal of Public Economics*, vol. 57, pp. 85–101, 1995.

[12] C. Prahlad, *The Fortune at the Bottom of the Pyramid: Eradicating Poverty Through Profit.* Upper Saddle River, NJ: Wharton School Publishing, July 2004.

[13] Final report prepared for the Consumer Electronics Association, "Energy consumption by consumer electronics in U.S. residences," 2007.

[14] Gartner Press Releases.

[15] IDC Press Releases.

[16] Univ. of Pennsylvania, Information Systems and Computing .

[17] Apple Energy and Environment website.

[18] Energy Star Rating of Computers.

[19] Guardian Newspaper.

[20] C. Warwick, "Trends and limits in the talk time of personal communicators," *Proceedings of the IEEE*, vol. 83, April 1995.

[21] CIA World Fact book.

[22] VGChartz.

[23] The Energy Saving Trust, "The ampere strikes back," 2007.

[24] International Energy Agency (IEA), "World energy outlook 2007."

[25] "The CanadaU.S. ict investment gap in 2008: Gains in communications equipment and losses in computers."

[26] Industry Canada, "ICT sector gross domestic product (GDP)," 2008.

[27] K. V. Palem, "Energy aware algorithm design via probabilistic computing: from algorithms and models to Moore's law and novel (semiconductor) devices," in *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture and Synthesis for Embedded Systems*, pp. 113–117, 2003.

[28] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies (C.E. Shannon and J. McCarthy eds.), Priceton Univ. Press, Princeton, N.J.*, 1956.

[29] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.

[30] P. Dubey, "A platform 2015 workload model recognition, mining and synthesis moves computers to the era of tera," *White paper, Intel Corp.*, 2005.

[31] J. George, B. Marr, B. E. S. Akgul, and K. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in *Proc. of the IEEE/ACM Intl. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 158–168, 2006.

[32] G. V. Varatkar and N. R. Shanbhag, "Energy-efficient motion estimation using error-tolerance," *in the proceedings of the International Symposium on Low power electronics and design (ISLPED)*, 2006.

[33] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," *in theproceedings of the International Symposium on Low power electronics and design (ISLPED)*, 2009.

[34] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Proc. of 15th IEEE/SIGDA Asia and South Pacific Design and Automation conference*, January 2010.

[35] S. Narayanan, J. Sartori, R. Kumar, and D. Jones, "Scalable stochastic processors," in *Proc. of the Design, Automation and Test in Europe*, March 2010.

[36] L. N. B. Chakrapani, K. K. Muntimadugu, A. Lingamneni, J. George, and K. V. Palem, "Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation," in *Proc. of the IEEE/ACM Intl.l Conf. on Compilers, Architecture, and Synthesis of Embedded Systems*, 2008.

[37] L. N. Chakrapani, B. E. S. Akgul, S. Cheemalavagu, P. Korkmaz, K. V. Palem, and B. Seshasayee, "Ultra efficient embedded SoC architectures based on proba-

bilistic CMOS technology," in *Proc. of the 9th Design Automation and Test in Europe*, pp. 1110–1115, Mar. 2006.

[38] K. V. Palem, L. N. Chakrapani, Z. M. Kedem, A. Lingamneni, and K. K. Muntimadugu, "Sustaining moore's law in embedded computing through probabilistic and approximate design: retrospects and prospects," in *Proc. of Intl. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 1–10, ACM, 2009.

[39] J. von Neumann, "Probabilistic logics and the synthesis of reliable organizms from unreliable components," *Automata Studies*, pp. 43–98, 1956.

[40] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S.-L. Lu, T. Karnik, and V. De, "Energy-efficient and metastability-immune timing-error detection and recovery circuits for dynamic variation tolerance," *IEEE International Conference on Integrated Circuit Design and Technology and Tutorial*, pp. 155–158, 2008.

[41] J. Ray, J. C. Hoe, and B. Falsafi, "Dual use of superscalar datapath for transient-fault detection and recovery," in *Proceedings of the 34th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 214–224, 2001.

[42] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 7–18, Oct. 2003.

[43] Chua, Leon O, "Memristor—the missing circuit element," *IEEE Transactions on Circuit Theory CT*, vol. 18, pp. 507–519, September 1971.

[44] J. M. Tour and D. K. James, "Molecular electronic computing architectures: A review," in *Handbook of Nanoscience, Engineering and Technology, Second Edition* (I. Goddard, W. A., D. W. Brenner, S. E. Lyshevski, and G. J. Iafrate, eds.), pp. 5.1–5.28, New York: CRC Press, 2007.

[45] J. Yao, Z. Sun, L. Zhong, D. Natelson, and J. M. Tour, "Resistive switches and memories from silicon oxide," *Nano Letters*, August 2010.

[46] "Advances offer path to further shrink computer chips," *New York Times, http://www.nytimes.com/2010/08/31/science/31compute.html*, 2010.

[47] K. V. Palem, "Energy aware computing through probabilistic switching: A study of limits," *IEEE Transactions on Computers*, vol. 54, no. 9, pp. 1123–1137, 2005.

[48] S. Cheemalavagu, P. Korkmaz, and K. V. Palem, "Ultra low-energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship," in *Proc. of the Intl. Conference on Solid State Devices and Materials*, pp. 402–403, Sept. 2004.

[49] J. Bau, R. Hankins, Q. Jacobson, S. Mitra, B. Saha, and A. A. Tabatabai, "Error resilient system architecture (ERSA) for probabilistic applications.," 2007.

[50] P. Korkmaz, *Probabilistic CMOS (PCMOS) in the Nanoelectronics Regime.* PhD thesis, Georgia Institute of Technology, 2007.

[51] L. N. Chakrapani and K. V. Palem, "A probabilistic boolean logic for energy efficient circuit and system design," *in the proceedings of the 15th Asia South Pacific Design Automation Conference*, 2010.

[52] N. Banerjee, G. Karakonstantis, and K. Roy, "Process variation tolerant low power DCT architecture," in *Proceedings of Design, Automation and Test in Europe Conference*, pp. 1 –6, Apr 2007.

[53] S. H. Kim, S. Mukohopadhyay, and W. Wolf, "Experimental analysis of sequence dependence on energy saving for error tolerant image processing," *in the proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, 2009.

[54] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," *In Proc. Int. Symp. on Low Power Electronics and Design*, pp. 30–35, 1999.

[55] N. Shanbhag, "Reliable and energy-efficient digital signal processing," *In Proc. Design Automation Conference*, pp. 830–835, 2002.

[56] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, pp. 813–823, Dec. 2001.

[57] L. Wang and N. R. Shanbhag, "Low-power filtering via adaptive error-cancellation," *IEEE Transactions on Signal Processing*, vol. 51, pp. 575–583, Feb. 2003.

[58] R. Hegde and N. Shanbhag, "A voltage overscaled low-power digital filter ic," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 388–391, February 2004.

[59] I. Chong and A. Ortega, "Dynamic voltage scaling algorithm for power constrained motion estimation," *In Proc. of International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[60] M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: analysis and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 12, pp. 1322 – 1335, 2006.

[61] I. Chong, H. Cheong, and A. Ortega, "New quality metric for multimedia compression using faulty hardware," *In Proc. of International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.

[62] R. Karp, "Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane," *Mathematics of Operations Research*, vol. 2, no. 3, pp. 209–224, 1977.

[63] S. A. Cook, "The complexity of theorem-proving procedures," *Proceedings of the third annual ACM symposium on Theory of computing*, pp. 151–158, 1971.

[64] R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations (Raymond E. Miller and James W. Thatcher (editors))*, pp. 85–103, 1972.

[65] A. Verma and P. Ienne, "Improving XOR-dominated circuits by exploiting dependencies between operands," *Proceedings of the Asia and South Pacific Design Automation Conference*, 2007.

[66] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design.* Addison Wesley Publishing Company, 1993.

[67] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits : A Design Perspective.* Prentice Hall, 2003.

[68] *Low-Power CMOS VLSI Circuit Design.* Wiley and Son, 2000.

[69] K. Nose and T. Sakurai, "Analysis and future trend of short-circuit power," *IEEE Tran. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, Sept. 2000.

[70] L.-Y. Chiou, K. Muhammand, and K. Roy, "Dsp data path synthesis for lowpower applications," *in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, no. 1165-1168, 2001.

[71] A. Raghunathan and N. K. Jha, "An iterative improvement algorithm for low power data path synthesis," *in IEEE/ACM Computer-Aided Design (ICCAD)*, Nov 1995.

[72] S. Iman and M. Pedram, "Two-level logic minimization for low power," *IEEE/ACM Computer-Aided Design (ICCAD)*, Nov 1995.

[73] J.-M. Tseng and J.-Y. Jou, "A power driven two-level logic optimizer," *in Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Jan 1997.

[74] J. Sklansky, "Conditional-sum addition logic," *IRE Transactions on Electronic Computers*, pp. 226–231, 1960.

[75] P. M. Kogge and H. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Transactions on Computers*, vol. 22, pp. 786–793, Aug. 1973.

[76] R. Ladner and M. Fischer, "Parallel prefix computation," *Journal of ACM*, vol. 27, pp. 831–838, Oct. 1980.

[77] R. Brent and H. Kung, "A regular layout for parallel adders," *IEEE Transactions on Computers*, vol. 31, no. 3, pp. 260–264, 1982.

[78] T. Han and D. Carlson, "Fast area-efficient vlsi adders," in *Proceedings of the 8th Symposium on Computer Arithmetic*, pp. 49–56, 1987.

[79] S. Mathew, M. Anders, R. K. Krishnamurthy, and S. Borkar, "A 4-ghz 130-nm address generation unit with 32-bit sparse-tree adder core," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 689–695, May 2003.

[80] R. Zimmerman, *Binary Adder Architectures for Cell-Based VLSI and their Synthesis.* PhD thesis, Swiss Federal Institute of Technolog, 1997.

[81] D. Harris, "A taxonomy of parallel prefix networks," vol. 2, pp. 2213 – 2217, Nov 2003.

[82] N. Pippenger, "Analysis of carry propagation in addition: An elementary approach," *Journal of Algorithms*, vol. 42, pp. 317–313, 2002.

[83] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[84] B. Parhami, *Computer arithmetic: Algorithms and hardware designs.* Oxford, UK: Oxford University Press, 2000.

[85] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective.* Addison Wesley, 2004.

[86] P. Song and G. De Micheli, "Circuit and architecture trade-offs for high-speed multiplication," *in the IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1184–1198, September 1991.

[87] "NCH Software."

[88] "Mediabench."

[89] J. W. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal Of Solid-State Circuits*, pp. 1396–1402, 2002.

[90] S. Gyger, A. Corbaz, and P.-A. Beuchat, "Hardware development kit for systems based on an icyflex processor," *CSEM Scientific and Technical Report*, 2009.

[91] H. Kaul, M. Anders, S. Mathew, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar, "A 320 mv 56 μw 411 gops/watt ultra-low voltage motion estimation accelerator in 65 nm cmos," in *IEEE Journal of Solid-State Circuits*, pp. 107–114, 2008.

[92] R. K. Krishnamurthy, "Ultra-low voltage microprocessor design challenges and solutions," in *Proceedings of the Ultra-low voltage Circuit Design Forum at International Solid State Circuits Conference*, 2009.