POWER EFFICIENT DESIGN OF SRAM ARRAYS AND OPTIMAL DESIGN OF SIGNAL AND POWER DISTRIBUTION NETWORKS IN VLSI CIRCUITS

by

Behnam Amelifard

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2007

DEDICATION

To my adorable mom and lovely wife

whom unconditional love and support made this work possible.

ACKNOWLEDGEMENTS

I am most grateful to my advisor, Professor Massoud Pedram, for inviting me to join his research group and providing invaluable support and guidance throughout my Ph.D. studies at USC. He has been a continuous source of motivation for me and I want to sincerely thank him for all I have achieved. His multi-disciplinary approach and global vision of research problems have been instrumental in defining my professional career.

I would also like to thank my other committee members, Professor Jeff Draper and Professor Aiichiro Nakano for their insightful suggestions and for their valuable time.

I am sincerely grateful to Dr. Farzan Fallah for his guidance in some parts of my Ph.D. research and his help and support during my internship at Fujitsu Laboratories of America. I would also like to extend my appreciation to Dr. Amir H. Ajami for his support and guidance during my summer internship at Magma Design Automation.

I would like to express my gratitude to Professor Ali Afzali-Kusha for his teaching, feedback, and advice during my graduate studies at University of Tehran.

I would like to thank my parents for their unconditional love and support. I would have not been able to accomplish my goals without their support and encouragement. I would also like to thank my sisters, Elham and Elnaz, who I love dearly. No matter how far away they may be physically, they are never far from my heart and mind.

I am much indebted to my uncle, Rahmat Rahnama, for believing in me and

encouraging me to pursue my studies; his strong support and guidance has been crucial in achieving my goals.

Words can not express my gratitude to my beloved wife, Taraneh. Not only is she my adorable wife and closest friend, but also one of the smartest colleagues, technically helping me with fruitful discussions. I would like to thank Taraneh for her constant love, support, and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASIC Application-Specific Integrated Circuit

BJT Bipolar Junction Transistor

CMOS Complementary Metal-Oxide Semiconductor

CPU Central Processing Unit

DIBL Drain-Induced Barrier Lowering

DPM Dynamic Power Management

DRV Data Retention Voltage

DSP Digital Signal Processing

DVS Dynamic Voltage Scaling

EMI Electromagnetic Interference

FB Functional Block

FCS Feasible Configuration Set

HCA Heterogeneous Cell Assignment

HCS Heterogeneous Cell SRAM

IC Integrated Circuit

ITRS International Technology Roadmap for Semiconductors

LDO Low Dropout

LPFO Low-Power Fanout Optimization

MOSFET Metal-Oxide Semiconductor Field Effect Transistor

MTCMOS Multi-Threshold CMOS

NICS Non-Inferior Configuration Set

NIRCS Non-Inferior Robust Configuration Set

NMOS n-channel Metal-Oxide Semiconductor

PCB Printed Circuit Board

PCN Power Conversion Network

PDA Personal Digital Assistant

PDN Power Delivery Network

PFM Pulse-Frequency Modulation

PMOS p-channel Metal-Oxide Semiconductor

PPS Power-Performance State

PSC Power Switch Network

PWM Pulse-Width Modulation

RCS Robust Configuration Set

RDF Random Dopant Fluctuation

RGF Restricted Growth Function

SER Soft Error Rate

SNM Static Noise Margin

SoC System-on-a-Chip

SRAM Static Random-Access Memory

UDSM Ultra Deep Sub-Micron

VLSI Very Large Scale Integration

VRM Voltage Regulator Module

ABSTRACT

In today's IC design, one of the key challenges is the increase in power dissipation of the circuit which in turn shortens the service time of battery-powered electronics, reduces the long-term reliability of circuits due to temperature-induced accelerated device and interconnect aging processes, and increases the cooling and packaging costs of these circuits. This dissertation investigates different techniques for low-power design of VLSI circuits. First, power minimization of on-chip caches is investigated. In particular, a technique is proposed to reduce the active power consumption of on-chip caches by utilizing dual threshold voltages and dual oxide thicknesses. Subsequently, a novel gating technique is presented to reduce the standby leakage current in the SRAM arrays. Next, the focus of the dissertation is shifted to power minimization in signal distribution networks. First, a low-power fanout optimization technique is presented which can be utilized to reduce the power dissipation cost of distributing a signal from source to multiple destinations. Subsequently, a methodology is presented for repeater insertion for global buses which enables low-power on-chip communication. Finally, the focus of the dissertation is shifted to power delivery network design for multiple-voltage-domain circuits. First, a technique is presented to optimally select the voltage regulator modules in the power delivery network of a SoC to achieve minimum power loss in the system. Next, a novel technique is described for power delivery network design to enable dynamic voltage scaling in a SoC.

Chapter 1
Introduction

Integrated circuit (IC) design has always been driven by the demand for having more functionality integrated on a single chip. In Today's System-on-a-Chip (SoC) designs, this functionality includes multiple processor cores, on-chip memory, audio/video encoder/decoder, various I/O controllers, RF front-end, signal processing engines, and multiple voltage regulators. To meet this demand, the semiconductor industry has successfully followed the Moore's Law, resulting in tremendous advances in CMOS manufacturing processes. The accompanying down scaling of the minimum feature sizes has enabled us to double the number of transistors every 15-18 months.

One side effect of technology scaling is that the Critical Dimension[1] (CD) has become so small that the atomicity of the physical features and dopant levels is becoming assessable. This results in large variations in the physical and electrical characteristics of interconnect and transistors which in turn affect the performance and power consumption of the circuit. Traditionally, process variations have been modeled by considering the worst-case process corners in order to evaluate the

[1] The critical dimension of a semiconductor technology is the smallest geometrical features (width of interconnect line, poly width, etc.) which can be formed during semiconductor manufacturing

performance of the design. Nevertheless, designing at the worst-case process corner leads to excessive guard-banding which in turn wastes lots of die resources and leaves silicon performance untapped; therefore, in recent years much research has been conducted on statistical modeling of variations [67, 78, 94, 138].

A further unfortunate consequence of technology scaling is that the short-term reliability (a.k.a. integrity) of VLSI circuits is reduced due to increase in various types of noise, e.g., crosstalk coupling noise, power supply noise, and radiation induced transient faults (a.k.a. soft errors). The increase in crosstalk noise is due to the fact that as technology scales down, the wire aspect ratios (height to width ratios) are increased to minimize the wire sheet resistance while at the same time wires are laid out closer to each other. As a result of these two trends, the capacitive coupling noise between interconnect lines is increased. The increase in power supply noise is due to the down-scaling of the supply voltages and the increase in current demand from the power supply network in successive CMOS technology nodes. Finally, the increase in transient faults in new technology generation is due to lower noise margins and lower nodal capacitances in the modern circuits. Many researchers have been working on development of effective techniques for analysis and optimization of VLSI circuits in the presence of these noise sources [31, 39, 53, 56, 57, 68, 84, 111, 139, 146].

Yet another consequence of technology scaling is that integrated circuit densities and operating frequencies are continuing to go up. The result is that chips are becoming larger, faster, and more complex, therefore, consuming ever larger

amounts of dynamic power [2]. At the same time, CMOS scaling toward Ultra-Deep Submicron (UDSM) technologies requires very low threshold voltages and ultra-thin gate oxides to retain the current drive and alleviate the short-channel effects. The side effect of threshold voltage and oxide thickness scaling is an exponential increase in both subthreshold and tunneling gate leakage currents, which adds to total power consumption of the chip.

The increase in the power consumption results in shorter battery lifetime for battery-operated portable devices such as laptops, cell phones, and PDA's. As a result, the primary objective of low-power design for battery-operated electronics is to extend the battery lifetime while meeting the performance demands. It is known that only a 30% improvement in battery performance can be achieved in five years [26]; therefore, unless power optimization techniques are applied at different levels of granularity, the capabilities of future portable systems will be strictly limited by the weight and size of the batteries required for an acceptable service duration [2]. In high performance desktop systems, on the other hand, the packaging cost and power reliability issues associated with high power consumption also has made the low-power design a primary design objective.

In the last decade, numerous research efforts have been made to address various techniques for power reduction at different levels of granularity. Reducing the capacitance [11, 12, 29, 109], the switching activity [61-63, 113], the frequency [26], and the supply voltage [9, 27, 134, 135] of the circuit are the bases of proposed techniques for reducing the dynamic power consumption. Reducing the supply

voltage [5, 43, 101, 144], utilizing multiple threshold voltages [4, 38, 48, 49, 71, 91, 119] or multiple gate oxide thicknesses [79, 80, 118], and power and/or ground gating [1, 60, 114, 133, 143] are the bases of proposed solutions to suppress leakage power consumption.

Given the importance of low-power design, this dissertation is focused on developing different techniques at circuit, logic, and system level for low-power design of CMOS VLSI circuits.

1.1 Dissertation Contributions

In this dissertation, we target three major sources of power consumption in modern integrated circuits:

- Caches
- Signal distribution network
- Power delivery network

As microprocessors are becoming larger and more complex, a larger portion of the die is dedicated to caches for data and code storage. Since leakage power consumption is roughly proportional to the area, the leakage current of caches is one of the major sources of power consumption in high performance microprocessors. Given that caches are made of static random-access memory (SRAM) blocks, low-leakage SRAM design is crucial to achieve a low-power microprocessor; therefore, in the first section of this dissertation, we investigate efficient techniques to reduce the leakage power consumption of SRAM's.

With the increase in the die size and gate count of integrated circuits, more

buffers are used in fanout trees and global buses to distribute signal on the die. Fanout trees are used for local signal propagation and they are constructed during logic synthesis when an output signal must be distributed to several destinations. Global buses, on the other hand, are used to enable global data transfer between different functional blocks on a die. Usually a large number of buffers are used in fanout trees and global buses to minimize delay from the source to sink(s). As a result, energy consumption of buffers used in the fanout tree and global buses is another major component of power consumption in a modern chip. Consequently, in the second part of this dissertation we address the problem of low-power fanout optimization and low-power global bus design.

Power delivery network has the responsibility of delivering the required current at appropriate voltage levels to different functional blocks on a die. If improperly designed, this network can be a major source of noise, and a major contributor to the chip power dissipation. Therefore, in the last part of this dissertation we concentrate on the problem of optimizing power delivery network for multi-supply-voltage designs.

1.2 Outline of the Dissertation

In Chapter 2 we provide some general background on leakage power dissipation and soft errors which will be used in consequent sections of this dissertation.

In Chapter 3 we present the heterogeneous cell SRAM for active leakage power reduction of caches. Heterogeneous cell SRAM is based on the observation that read and write delays of a memory cell in an SRAM array depend on physical location of

the cell in the array. Therefore, the idea is to deploy different configurations of six-transistor SRAM cells corresponding to different threshold voltages and oxide thicknesses for the transistors. We show that designing a heterogeneous cell SRAM requires only a minor change in the SRAM design flow and does not incur any hardware or delay overheads. We study the effect of different design parameters, including the size of SRAM array, the number of allowed cell configurations, and values of high threshold voltage and oxide thickness, on the power consumption of heterogeneous cell SRAM. It is demonstrated that compared to a conventional SRAM, where all cells have the same threshold voltage and oxide thickness, heterogeneous SRAM reduces the total leakage by 20%-40%.

In Chapter 4 we present PG-gated SRAM cell for standby leakage power reduction of caches. PG-gated SRAM is based on the idea that both leakage and hold static noise margin of a cell vary with the exact values of its supply and ground voltages. As a result, given a fixed value of the voltage difference on the power rails of the SRAM cell during the standby mode, optimum ground and supply voltage levels exist for which the SRAM leakage is minimized subject to a hold static noise margin constraint. Therefore, the idea is to bias the SRAM cell in the standby mode to the optimum values of ground and supply voltages to achieve minimum leakage power consumption. We show that PG-gated technique not only improves the leakage power consumption of the SRAM, but also enhances the hold static noise margin and soft error immunity. Moreover, it improves the leakage and static noise margin variability under process and temperature variations. Compared to a

conventional gating technique, PG-gated SRAM reduces the total leakage by more than 60%.

In Chapter 5 we address the problem of low power fanout optimization for near-continuous size inverter libraries. We show that by neglecting the short-circuit currents, previous techniques proposed to optimize the area of a fanout tree may result in excessive power consumption. We formulate the problem of optimizing the total power consumption of a fanout tree with only one sink (i.e., a fanout chain) as a convex optimization problem to solve it efficiently. Then, we show how to construct a low power fanout tree from power-optimized fanout chains. Simulation results demonstrate that the proposed technique can reduce the power consumption of the fanout trees by an average of 11.17% over SIS fanout optimization program.

In Chapter 6 we present a technique for power-optimal repeater insertion for global buses in the presence of crosstalk noise. We accurately model the effect of crosstalk coupling capacitance not only on propagation delay, but also on different components of power dissipation. Furthermore, we utilize sleep transistors to reduce the leakage power consumption of the bus in idle mode. The problem of simultaneously calculating the repeater sizes, repeater distances, and size of the sleep transistors to minimize the power dissipation subject to a delay constraint is modeled as a mathematical problem and solved efficiently. Additionally, we show how to design the sleep signal delivery circuitry for buses to incur minimum area and power overheads. Compared to a delay-optimal bus line, the proposed technique can reduce the power consumption by 50% with a small delay penalty of 5%.

In Chapter 7 we address the problem of optimal selection of voltage regulator modules (VRM's) in a power delivery network (PDN) of a multiple-supply-voltage SoC. We show that by using a tree topology of suitably chosen VRM's, the power efficiency of the system can be improved. We present a dynamic programming technique to select the best set of VRM's in a fixed VRM tree. Furthermore, we describe how to efficiently generate the set of all VRM tree topologies. Compared to the conventional way of putting one VRM for each functional block, the proposed technique can reduce the power consumption of PDN by an average of 17%.

In Chapter 8 we present a new technique to design a power delivery network for a complex SoC so as to enable dynamic power management through assignment of appropriate voltage level to each function block in the SoC. In this technique the PDN is composed of two layers. In the first layer of the PDN, which is called the power conversion network, fixed-V_{out} VRM's are used to generate all voltage levels that may be needed by different functional blocks in the SoC design. In the second layer of the PDN, a power switch network is used to dynamically connect the power supply terminals of each functional block to the appropriate VRM output in the PCN. Compared to the conventional way of putting a variable-V_{out} VRM for each functional block, the proposed technique reduces the power loss of the power delivery network by an average of 34% while reducing its cost by an average of 8%.

Chapter 2
Preliminaries

2.1 Introduction

CMOS scaling beyond the 90nm technology node requires not only very low threshold voltages (V_t) to retain the device switching speeds, but also ultra-thin gate oxides (T_{ox}) to maintain the current drive and keep threshold voltage variations under control when dealing with short-channel effects [126]. Low threshold voltage results in an exponential increase in the subthreshold leakage current, whereas ultra-thin oxide causes an exponential increase in the tunneling gate leakage current. As a result, reducing the subthreshold and tunneling gate leakage currents has become one of the most important challenges in the design of VLSI circuits [86]. Any technique which attempts to reduce the leakage power consumption should be able to accurately model different components of leakage currents in modern CMOS devices.

Another unfortunate consequence of technology scaling is that the susceptibility of integrated circuits to radiation induced transient faults has been increased. These transient faults, which are also known as soft errors, are more critical for memory elements [53, 68, 111]; therefore, it is crucial to investigate the impact of any power

optimization technique for memory elements on soft error susceptibility.

In this chapter we provide some background on leakage currents and soft errors which will be used in subsequent chapters.

2.2 Leakage Current Components

The leakage current of a deep submicron CMOS transistor consists of three major components: junction tunneling current, subthreshold current, and tunneling gate current [38] which are depicted in Figure 2.1. In this section, each of these three components is briefly described.

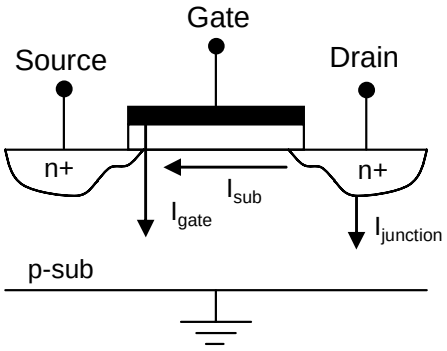


Figure 2.1: Major leakage current components in an NMOS transistor.

2.2.1 Junction Leakage Current

The junction leakage occurs from the source or drain to the substrate through the reverse-biased diodes when a transistor is OFF. The reversed biased P-N junction leakage has two main components: one corresponds to the minority carriers' diffusion near the edge of the depletion region and the other is due to electron-hole pair generation in the depletion region of the reverse biased junction [38]. The tunneling junction leakage current is an exponential function of the junction

doping and reverse bias voltage across the junction. It is known that junction leakage has a rather high temperature dependency (i.e., as much as 50–100 x/100^{o}C) but it is generally insignificant except in circuits designed to operate at high temperatures (>150^{o}C) [40]. Since in present technologies tunneling junction leakage current is quite small compared to other sources of leakage in state-of-the-art CMOS devices [38], in this manuscript we do not attempt to reduce this component of leakage; however, it should be noticed that by applying a forward substrate biasing, tunneling junction current can be reduced [4].

2.2.2 Subthreshold Leakage Current

The subthreshold leakage is the drain-source current of a transistor operating in the weak inversion region. Unlike the strong inversion region in which the drift current dominates, the subthreshold conduction is due to the diffusion current of the minority carriers in the channel of a MOS device. The subthreshold leakage is modeled as [38],

$$I_{sub} = A_{sub}\mu_0 C_{ox}\left(\frac{w}{L_{eff}}\right)\exp\left(\frac{q}{n'kT}\left(V_{gs} - V_{t0} - \gamma' V_{sb} + \eta V_{ds}\right)\right)\times \left(1-\exp\left(-\frac{q}{kT}V_{ds}\right)\right) \tag{2.1}$$

where $A_{sub} = (kT/q)^2 \exp(1.8)$, μ_0 is the zero bias mobility, C_{ox} is the gate oxide capacitance per unit area, w and L_{eff} respectively denote the width and effective length of the transistor, k is the Boltzmann constant, T is the absolute temperature, and q is the electrical charge of an electron. In addition, V_{t0} is the zero biased threshold voltage, V_{gs}, V_{sb}, and V_{ds} are respectively the gate-to-source,

source-to-bulk, and drain-to-source voltages of the transistor. Furthermore, γ' is the linearized body-effect coefficient, η denotes the drain-induced barrier lowering (DIBL) coefficient, and n' is the subthreshold swing coefficient of the transistor.

As transistor supply voltage is scaled down, the threshold voltage must also be reduced to retain the switching speed of the transistor. From (2.1) one can see that this trend results in an exponential increase in the subthreshold leakage. One effective way of reducing the subthreshold leakage is to use higher threshold voltages in some parts of a design. There are different ways to achieve a higher threshold voltage [118], chief among them are adjusting the channel doping concentration and applying a body bias.

2.2.3 Tunneling Gate Leakage Current

The other major source of the leakage power dissipation is due to the tunneling gate leakage current. In NMOS transistors, the tunneling gate current happens because of the electron tunneling from the conduction band (ECB), which is significant in accumulation region. In PMOS transistors, on the other hand, the hole tunneling from the valence band (HVB) gives rise to the tunneling gate leakage. The tunneling current is composed of three major components: (1) gate-to-source and gate-to-drain overlap currents, (2) gate-to-channel current, part of which goes to the source and the rest goes to the drain, and (3) gate-to-substrate current. In bulk CMOS technology, the gate-to-substrate leakage current is several orders of magnitude lower than the overlap tunneling current and gate-to-channel current [79]. While the overlap tunneling current dominates the gate leakage in the OFF state, the gate-to-channel

tunneling dictates the gate current in the ON state. Since the gate to source and gate to drain overlap regions are much smaller than the channel region, the tunneling gate leakage in the OFF state is much smaller than the gate leakage in the ON state [79].

If SiO_2 is used for the gate oxide, a PMOS transistor will have about one order of magnitude smaller gate leakage current than an NMOS transistor with identical T_{ox} and V_{dd} [50, 79]. Based on the above analysis, it is concluded that the major source of tunneling gate leakage in CMOS circuits is the gate-to-channel tunneling current of the ON NMOS transistors, which can be modeled as [38],

$$I_{ox} = A_{ox} w_N L_{eff} \left(\frac{V_{ox}}{T_{ox}} \right)^2 e^{-B_{ox} \frac{T_{ox}}{V_{ox}}} \tag{2.2}$$

where A_{ox} and B_{ox} are technology constants, and V_{ox} is the potential drop across the oxide. When the transistor is ON, $V_{ox} = V_{gs} - \psi_s$, where ψ_s is the surface potential of the transistor.

As transistor length and supply voltage are scaled down, gate oxide thickness must also be reduced to maintain effective gate control over the channel region. From Equation (2.2) one can see that this trend results in an exponential increase in the tunneling gate leakage. An effective approach to overcome the tunneling gate leakage current while maintaining gate control over the channel is to replace the currently-used SiO_2 gate insulator with high-*k* dielectric material [40]. In [85, 88] a comparative study of using high-*k* dielectric and dual oxide thickness on the leakage power consumption has been presented and an algorithm for simultaneous high-*k* and high-T_{ox} assignment has been proposed. Although some investigation has been done

on Zirconium- and Hafnium-based high-*k* dielectrics [28], there are unresolved manufacturing process challenges in way of introducing high-*k* dielectric material under the gate (e.g., related to the compatibility of these materials with Silicon [79] and the need to switch to metal gates); hence, high-*k* dielectrics are not expected to be used before 45nm technology node [28, 65], leaving multiple gate oxide thicknesses as the one promising solution to reduce tunneling gate leakage current at the present time. To achieve multiple oxide thicknesses Arsenic can be implanted into the Silicon substrate before thermal oxidation is done [130].

2.3 Soft Error

A high-energy alpha particle or an atmospheric Neutron striking a capacitive node of a circuit deposits charge which leads to a time-varying voltage pulse at the node. In the case of atmospheric Neutrons, the current flow created by the charge deposited into the node is modeled as [53](similar models exist for alpha-particle related soft errors):

$$I(Q,t) = \frac{2Q}{\sqrt{\pi}T_s}\sqrt{\frac{t}{T_s}}\exp\left(\frac{-t}{T_s}\right) \tag{2.3}$$

where Q is the collected charge and T_s is the technology-dependent collection waveform time constant. If the collected charge Q exceeds the critical charge Q_{crit} in an SRAM cell, it will upset the bit value and cause a soft error. In [53] a methodology for estimating the Neutron-induced soft error rate (SER) in SRAM has been proposed, according to which the dependence of SER on circuit and environmental parameters is expressed as:

$$SER \propto N_{flux} A_S \exp\left(-\frac{Q_{crit}}{Q_s}\right) \tag{2.4}$$

where N_{flux} is the intensity of the Neutron flux and A_S is the area of the cross section of the node (i.e., the area of the drain or source region). Moreover, Q_s is the collection slope, which depends strongly on the doping concentration of the drain and source and also the supply voltage level.

Chapter 3
Heterogeneous Cell SRAM

3.1 Introduction

In many modern microprocessors, caches occupy a large portion of the die. For example, in Intel's Itanium 2 Montecito processor [90], more than 80% of the die is dedicated to caches. Since the leakage power dissipation is roughly proportional to the area of a circuit, the leakage power of caches is one of the major sources of power consumption in high performance microprocessors.

In the past, much research has been conducted to address the problem of leakage in SRAM's. In [71], for example, a dynamic threshold voltage method to reduce the leakage power in SRAM's has been utilized. In that technique, the threshold voltage of the transistors of each cache line is controlled separately by using forward body biasing. In [17], on the other hand, by observing the fact that in ordinary programs most of the bits in data-cache and instruction-cache are zero, the authors proposed using asymmetric SRAM cells to reduce the subthreshold leakage. Leakage biased bit-lines [106], and dynamic power gating [5, 6, 35, 43, 98, 143] are other effective techniques for reducing the leakage power in SRAM's.

Although many techniques have been proposed to address the problem of low-

leakage SRAM design, most of them address only the standby leakage power consumption, while it is known that in sub-100nm designs, active leakage comprises more than 20% of the total active power dissipation in memories [132]. On the other hand, many of these techniques result in hardware overhead and hence increase chip's area and reduce the manufacturing yield. Furthermore, many of them try to reduce the subthreshold leakage current only, whereas for sub-100nm technology node, the tunneling gate leakage is comparable to the subthreshold leakage. In this chapter we present a method for reducing both subthreshold and tunneling gate leakage current of an SRAM by using different threshold voltages and oxide thicknesses for transistors in an SRAM cell. The proposed method is based on the observation that read and write delays of a memory cell in an SRAM block depend on the physical distance of the cell from the sense amplifier and the decoder. Thus, the idea is to deploy different configurations of six-transistor SRAM cells corresponding to different threshold voltage and oxide thickness assignments for the transistors. We show that our heterogeneous cell SRAM (HCS) technique has several main advantages over previous techniques in that it:

- reduces both active and standby leakage current including subthreshold and tunneling gate leakage components,
- has no hardware or delay overheads,
- requires only a minor change in the SRAM design flow, and
- has the ability to improve the static noise margin under process variations.

The remainder of this chapter is organized as follows. In Section 3.2 the SRAM design and operation is discussed. Our idea for reducing the leakage power dissipation is presented in Section 3.3. Section 3.4 is dedicated to the experimental results. Finally, we summarize the chapter in Section 3.5.

3.2 SRAM Design and Operation

3.2.1 SRAM Architecture

A typical SRAM block, shown in Figure 3.1, consists of cell arrays, address decoders, column multiplexers, sense amplifiers, I/O, and a control unit. In the following, the functionality and design of each component is briefly discussed.

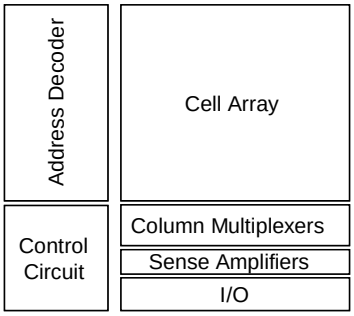


Figure 3.1: An SRAM block.

3.2.1.1 SRAM Cell

Figure 3.2 shows a 6-transistor (6T) SRAM cell. The bit value stored in the cell is preserved as long as the cell is connected to a supply voltage whose value is greater than the Data Retention Voltage (DRV) [101]. In an SRAM cell, the pull-down NMOS transistors and the pass-transistors reside in the read path. To achieve a high read stability, the pull-down transistors are made stronger than the pass-transistors.

The pull-up PMOS transistors and the pass-transistors, on the other hand, are in the write path. Although using strong PMOS transistors improves the read stability, it degrades the write-margin [143]; hence, a proper sizing of pass-transistors is required to achieve an adequate write margin.

Traditionally all cells used in an SRAM block are identical (i.e., corresponding transistors have the same width, threshold voltage, and oxide thickness) which results in identical leakage characteristic for all cells. However, as we will show in this chapter, by using non-identical cells, which have the same layout footprint, one can achieve more power efficient designs.

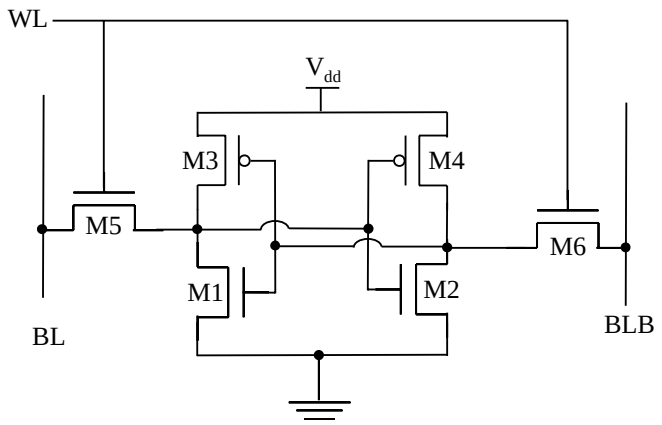


Figure 3.2: A 6T SRAM cell.

3.2.1.2 Cell Array

Most SRAM designs consist of multiple cell arrays. The size of the cell array depends on both performance and density requirements. Generally speaking, as technology shrinks, cell arrays are moving from tall to wide structures [143] [51]. However, since wider arrays need more circuitry for column multiplexers and sense

amplifiers, if a small area overhead is desirable (e.g., large L3 caches), the number of rows is kept high [144] [141].

3.2.1.3 Address Decoder

Although the logical function of an address decoder is very simple, in practice designing it is complicated because the address decoder needs to interface with the core array cells and pitch matching with the core array can be difficult [100]. To overcome the pitch-matching problem and reduce the effect of wire's capacitance on the delay of the decoder, the address decoder is often broken into two pieces. The first piece, called pre-decoder, is placed before the long decoder wires and the second part, row decoder, which usually consists of a single NAND gate and buffers for driving the word-line capacitance, is pitch-matched and placed next to each row as shown in Figure 3.3.

3.2.1.4 Column Multiplexers and Sense Amplifiers

Column multiplexing is inevitable in most SRAM designs because it reduces the number of rows in the cell array and as a result increases the speed. Since during a read operation one of the bit or bitbar lines is partially discharged, a sense amplifier is used to sense this voltage difference between bit and bitbar lines to create a digital voltage. Although sense amplifiers can operate by very small voltage differences such as 50mV, to make the circuit more robust to noise, the sense amplifier is typically switched when the voltage difference between bit and bitbar lines becomes 100-200mV.

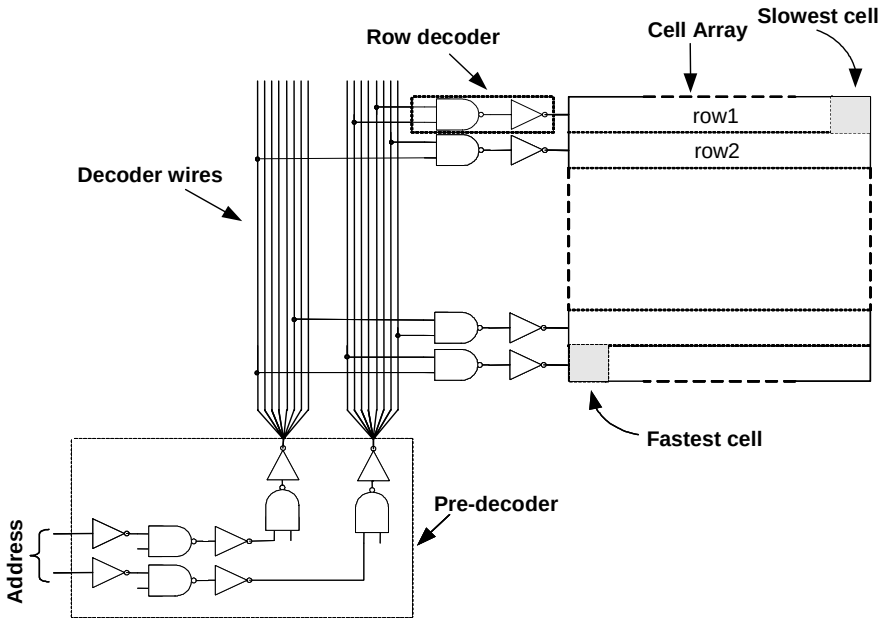


Figure 3.3: An SRAM block with its decoder.

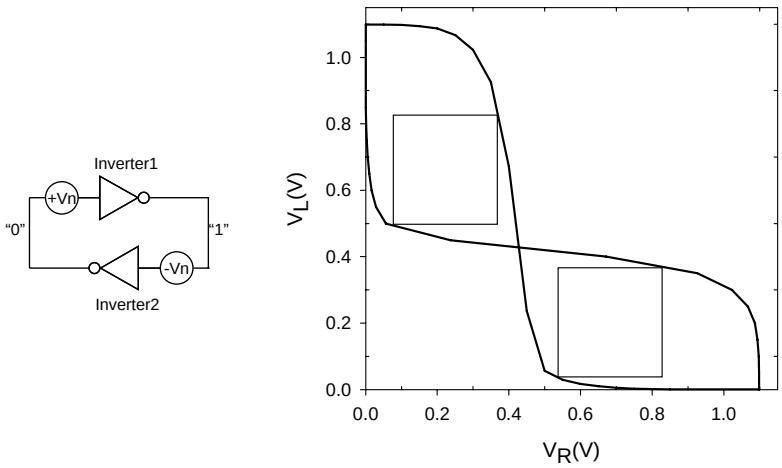
3.2.1.5 Control Unit

The control unit generates internal signals of the SRAM, including the write and read enable signals, the pre-charge signal, and the sense amplifier enabler.

3.2.2 Static Noise Margin

The Static Noise Margin (SNM) of a CMOS SRAM cell is defined as the minimum DC noise voltage necessary to flip the state of a cell [110] (c.f. Figure 3.4.a). SNM can be graphically computed from the butterfly curve. The butterfly curve of an SRAM cell, shown in Figure 3.4.b, is obtained by drawing and mirroring the DC characteristic of the cross-coupled inverters. By measuring the size of the largest square that can be embedded in the lopes of the butterfly curve, the static noise margin can be found.

SRAM cells are especially sensitive to noise during a read operation because the "0" storage node rises to a voltage higher than ground due to a resistive voltage divider comprised of the pull-down NMOS transistor and the pass transistor. If this voltage is high enough, it can change the cell's value.



(a) (b)

Figure 3.4: Measuring the static noise margin.

3.2.3 Leakage Paths in SRAM

There are two dominant subthreshold leakage paths in a 6T SRAM cell: 1) V_{dd} to ground paths inside the SRAM cell and 2) the bit-line (or bit-bar line) to ground path through the pass transistor. To reduce the first type of leakage, the threshold voltages of the pull-down NMOS transistors and/or pull-up PMOS transistors can be increased, whereas to lower the second type of leakage, the threshold voltages of the pull-down NMOS transistors and/or pass transistors can be increased. If the threshold voltage of the pull up PMOS transistors is increased, the write delay increases while the effect on the read delay would be negligible. On the other hand, if the threshold voltage of the pull down NMOS transistors is increased, the read

delay increases while the effect on the write delay would be marginal. By increasing the threshold voltage of the pass transistors, both read and write delays increase.

The major contributor to the tunneling gate leakage current in a 6T SRAM cell is the gate-to-channel leakage of the ON pull-down transistor. To weaken this leakage path, one needs to increase the gate-oxide thickness of the pull-down transistors. To reduce other (minor) tunneling gate leakage currents in the SRAM cell, one only needs to increase the gate oxide thickness of the pass transistors, because from the discussion in Section 2.2.3, one can see that the gate leakage saving achieved by increasing the oxide thickness of the PMOS transistors would be quite small. Increasing the oxide thickness of a transistor not only increases the threshold voltage, but also reduces the drive current of the transistor. So, the effect of applying this technique to an SRAM cell is an increase in the read/write delay of the cell.

3.3 Heterogeneous Cell SRAM

Due to the non-zero delay of the interconnects of the address decoder, word-lines, bit-lines, and the column multiplexer, read and write delays of different cells in an SRAM block are different. Simulations show that for typical SRAM blocks, depending on the number of rows and columns, the read time of the closest cell to the address decoder and the column multiplexer may be 5-15% less than that of the furthest cell from the address decoder and the column multiplexer. This provides an opportunity to reduce the leakage power consumption of an SRAM by increasing the threshold voltage or oxide thickness of some of the transistors in the SRAM cells. The resulting SRAM is called *heterogeneous cell SRAM* (HCS). In this section, it is

shown how to design an HCS without degrading the performance or robustness.

3.3.1 Technology

All results presented in this chapter are obtained by HSPICE [58] simulations using a predictive 65nm technology model [99] with 1.1V for the supply voltage, 0.18V for the threshold voltage, and 12Å as the gate oxide thickness. Moreover, unless otherwise stated, it is assumed that the value of the high threshold voltage is 0.28V and the value of the thicker gate oxide is 14Å. The SiO_2 layer in the gate stack is assumed to be 2Å thicker than the thin oxide so as to achieve one order of magnitude reduction in tunneling gate leakage. All simulations are performed at a die temperature of 100 ^{o}C.

The SRAM module used in these simulations is a pre-designed 64Kb SRAM with a 64-bit word and comprised of two cell arrays, each of which containing 64 rows and 512 columns. All local and global interconnects, including bit and bit-bar lines, word line, and decoder wires have been modeled as distributed RC circuits. In this SRAM, the read delay difference between the slowest cell and the fastest one is about 9%.

Although the simulation results we present in this section are specific to the aforesaid technology and design parameters, the general methodology is applicable to any SRAM block designed in any technology. In Section 3.4 we show how the results change with the change of the values of high-T_{ox} and high-V_t, and also as a function of the SRAM cell array size.

3.3.2 Library Generation

It is known that each additional threshold voltage or oxide thickness requires one additional mask layer in the fabrication process, which increases the manufacturing cost and reduces the yield [119, 130]. As a result, in many cases, only two threshold voltages and/or two oxide thicknesses are utilized in circuits. That is also why we shall concentrate on the problem of low-leakage SRAM design in a dual-V_t and dual-T_{ox} technology in this chapter. Clearly, it is possible to extend the results to handle more than two threshold voltages and two oxide thicknesses. In the next section it is shown how the results are changed if only the option of dual-V_t is available in the technology. We show that in this case, although the efficacy of our technique is reduced, the leakage reduction still remains significant.

The maximum reduction in the subthreshold leakage currents in a SRAM cell is achieved by increasing the threshold voltage of all transistors in the cell. Unfortunately, this scenario also results in the largest read delay penalty for the cell. Therefore, we also consider other configurations which result in lower subthreshold leakage reductions, but also smaller delay penalties. On the other hand, as mentioned in Section 2.2.3, to reduce the tunneling gate leakage of an SRAM cell, only the oxide thickness of the pull-down NMOS transistors and the pass-transistors must be increased. Although this is seemingly desirable from a low power point of view, it is not applicable for all cells in the cell array, i.e., thinner oxide thicknesses needs to be used in the cells that are far from the address decoder and the sense amplifiers. It is worth mentioning that due to roll-off effect, increasing the oxide thickness also raises the threshold voltage, resulting in a decrease in the subthreshold leakage. In the

following, high-V_t transistors refer to the devices whose threshold voltages have been modified by increasing the channel doping only. Furthermore, our simulations show that when the gate oxide thickness of the PMOS transistors is increased, the reduction in subthreshold leakage due to roll-off effect is very small. That is, the overall leakage reduction achieved by using a thicker gate oxide for the PMOS transistor is negligible.

To make the memory cells more manufacturable, unlike [17], we use a symmetric cell configuration, which means that symmetrically located transistors within an SRAM cell will have the same threshold voltages and oxide thicknesses. Thus, there are 32 different possibilities for assigning high and low threshold voltages and oxide thicknesses to the transistors within a cell. Since increasing the oxide thickness increases the threshold voltage of a transistor as well, we do not increase both the oxide thickness and threshold voltage for a transistor because the delay penalty will be too high. Therefore, the number of different configurations is reduced to eighteen (there are two choices for the pair of PMOS transistors, three choices for the pull-down NMOS pair, and three choices for the pass-transistor pair). Each configuration is shown by a triplet (x, y, z) where the first entry x in the triplet corresponds to the pair of pull-down transistors M1 and M2, the second entry y corresponds to the pair of pull-up transistors M3 and M4, and the third entry z corresponds to the pass-transistors M5 and M6 as shown in Figure 3.2. Each entry is zero, one, or two, if the corresponding transistors are respectively normal, high-V_t, or high-T_{ox}. For example, $(0,0,0)$ corresponds to the original configuration where all transistors in the cell

assume default (low) V_t and (low) T_{ox} values whereas $(0,1,2)$ corresponds to a configuration with nominal pull-down transistors, high-V_t pull-up transistors, and high-T_{ox} pass-transistors.

It should be emphasized that our technique does not require all configurations to be used in the optimization process. If a configuration cannot be manufactured due to process restriction or if it has a high manufacturing cost, it can be excluded from the library. Since using eighteen configurations in the optimization process is too expensive, we next show how to eliminate some 'inferior' configurations.

Each configuration has a specific delay and leakage characteristics. We denote the leakage power of the configuration C with $P(C)$ and its read and write delays with $D_R(C)$ and $D_W(C)$, respectively. More specifically, $D_R(C)$ is the difference between the time the address bit's voltage reaches $1/2V_{dd}$ and the time the output of the read buffer reaches 90% of its final value. On the other hand, $D_W(C)$ is the write delay, defined as the difference between the time the address bit's voltage reaches $1/2V_{dd}$ and the voltage of bitbar inside the cell reaches 90% of their final values.

Due to the delay of sense amplifiers and output buffers in a read path, the read delay of a cell is higher than its write delay. Therefore, the read delay specifies the performance of an SRAM. Considering the fact that the PMOS transistors in a 6T SRAM cell have a marginal impact on the read delay, it can be seen that increasing the threshold voltage of these transistors increases the write delay without having much effect on its read delay; so one may reduce the leakage power by increasing the threshold voltage of the PMOS transistors as long as the write time is below a target

value.

Definition 3.1: Assume when only the original configuration $(0,0,0)$ is used, the read-delay of the closest and furthest cells to the address decoder and the column multiplexer are $T_{\min}$ and $T_{\max}$, respectively (c.f. Figure 3.3). Configuration C is called *feasible,* if its read and write delays are less than $T_{\max}$. The set of all feasible configurations is called the *Feasible Configuration Set* (*FCS*).

Definition 3.2: Configuration $C_1 \in FCS$ is *inferior* if there exists a configuration $C_2 \in FCS$, whose leakage power and read-delay are no larger than those of C_1, i.e., $P(C_2) \leq P(C_1)$ and $D_R(C_2) \leq D_R(C_1)$.

It should be noted that the inferiority of a cell depends on different parameters, including the size of the transistors in the cell, the size of the array, and the technology library being used. Changing any of these parameters may change the dominancy relation between two cells.

Definition 3.3: The maximum subset of *FCS* which does not contain any inferior configuration is called the *Non-Inferior Configuration Set* (*NICS*).

NICS may be obtained by simulating all configurations and removing the inferior ones. When designing a heterogeneous cell SRAM, instead of using the complete set of configurations, *NICS* can be used without degrading the results. Table 3.1 shows the set of *NICS* along with their leakage power reduction and read delay increase for the technology described in Section 3.3.1. From this table one can see the delay penalty of some configurations is very small while their leakage saving is significant, e.g., $(1,0,0)$. These configurations are ideal candidates for HCS.

Table 3.1: Non-inferior configuration set (NICS)

Cell	% Leakage Reduction over (0,0,0) Cell	% Read Delay Increase over (0,0,0) Cell
(0,0,0)	-	-
(1,0,0)	43.39	3.02
(0,1,0)	7.60	0.00
(1,1,0)	50.96	2.98
(2,0,0)	16.35	0.43
(2,1,0)	23.93	0.41
(1,1,2)	55.57	6.63

3.3.3 Stability

To design an HCS as robust as the conventional SRAM, only configurations that do not degrade the SNM should be used during design.

Definition 3.4: Configuration *C* is *robust,* if its static noise margin is not any smaller than that of the original cell $(0,0,0)$.

Definition 3.5: The maximum subset of *FCS* which contains only robust configurations is called *Robust Configuration Set* (*RCS*). The maximum subset of *RCS* which does not contain any inferior configuration is called *Non-Inferior RCS* (*NIRCS*).

To obtain the robust configurations, we consider three separate criteria for SNM: SNM under nominal conditions, worst-case corner-based SNM, and statistical SNM.

3.3.3.1 Stability under Nominal Conditions

Table 3.2 lists the set of *NIRCS* when the criterion for robustness is the SNM under nominal condition (*NIRCS-NC*). Also shown are the nominal SNM of each configuration in this set along with the percentage of its improvement over the original cell.

Table 3.2: Nominal SNM of configurations in NIRCS-NC

Cell	Nominal SNM (mV)	% Increase over (0,0,0) Cell
(0,0,0)	185	-
(1,0,0)	208	12.43
(1,1,0)	201	8.65
(1,1,2)	208	12.43

3.3.3.2 Worst Case Stability

Since small transistors are typically used in SRAM cells to achieve a compact design, the most significant source of random intra-die variations in SRAM cells is the threshold voltage variation due to the Random Dopant Fluctuation (RDF) and the line width variation [89]. On the other hand, it is known that gate oxides are very well controlled compared to other dimensions such as the effective channel length [79]. Hence, in this section, we only consider threshold voltage variation for transistors in the 6T SRAM cell.

In the presence of RDF, the threshold voltage of the SRAM cell transistors can be considered as independent Gaussian random variables [22, 89] where the standard deviation of each transistor depends on its length and width as well as manufacturing process. In other words [127],

$$\sigma = \sigma_{\min}\sqrt{\frac{W_{\min}L_{\min}}{WL}} \tag{3.1}$$

where σ is the standard deviation of the threshold voltage of a transistor with the channel length and width of L and W, and $\sigma_{\min}$ is the standard deviation of the threshold voltage for the minimum sized transistor in a given manufacturing process [127].

To measure the worst-case SNM, each configuration is tested under all corners of

V_t variation. To limit the yield loss, we consider a large range of parametric variation, i.e., 5σ, for the transistors in each configuration; so, each configuration is tested in all corners of $\{-5\sigma, 0, +5\sigma\}$. The number of these corners for each configuration is 3^6=729. In these simulations, the standard deviation of each transistor is obtained from (3.1) by assuming $\sigma_{\min} = 30mV$ which is a typical standard deviation of the threshold voltage in the 65nm technology node [65]. By simulating all configurations, *NIRCS-WC*, which denotes *NIRCS* with the worst-case SNM robustness condition, is obtained (see Table 3.3.)

Table 3.3: Set of NIRCS-WC

Cell	Worst-Case SNM (mV)	% Increase over (0,0,0) cell
(0,0,0)	25	-
(1,0,0)	44	76.00
(1,1,0)	40	60.00
(1,1,2)	47	88.00

3.3.3.3 Statistical Stability

To measure the statistical stability of each configuration, we used a Monte Carlo simulation of 500 samples to obtain the statistical mean and variance of the SNM for each configuration.

The threshold voltage of each transistor has been modeled as an independent Gaussian random variable whose standard deviation is obtained from (3.1) by assuming $\sigma_{\min} = 30mV$ [65]. By simulating all configurations, *NIRCS-MC*, which denotes *NIRCS* with the statistical SNM robustness condition, is obtained (see Table 3.4.) Here the measure of robustness has been assumed to be $\mu - 5\sigma$. Interestingly, from Table 3.2-Table 3.4, one can see that for the technology we are

using, the three different criteria for robustness result in the same set of configurations. This result may not hold for other technologies or technology nodes.

Table 3.4: Set of NIRCS-MC

Cell	μ_{SNM} (mV)	σ_{SNM} (mV)	μ_{SNM}-5σ_{SNM} (mV)	% (μ_{SNM}-5σ_{SNM}) Increase over (0,0,0) Cell
(0,0,0)	186	24	66	-
(1,0,0)	210	26	80	21.21
(1,1,0)	202	25	77	16.67
(1,1,2)	209	25	84	27.27

3.3.4 Read Stability

The read stability is a transient stability metric which specifies the likelihood of inverting an SRAM cell's stored value during a read operation [17]. It is typically computed as the ratio of I_{trip} / I_{read}, where I_{trip} is the current through the pull-down NMOS transistor on the "0" side of the cell when the state of the cell is inverted by an external current I_{test} injected at the node storing the "0" value. Notice that I_{read} is the maximum current through the pass-transistor during the read operation [51]. The larger the I_{trip} / I_{read} ratio, the higher the read stability of a cell is.

The read stability simulation results on *NICS* configurations are reported in Table 3.5. From this table, it is seen that for different configurations in *NICS*, the maximum reduction in I_{trip} / I_{read} is 7.1%.

Table 3.5: Read stability for NICS cells

Cell	I_{trip}/I_{read}	% Decrease over (0,0,0) cell
(0,0,0)	1.69	-
(1,0,0)	1.63	3.5
(0,1,0)	1.64	3.0
(1,1,0)	1.60	5.3
(2,0,0)	1.62	4.1
(2,1,0)	1.57	7.1
(1,1,2)	1.68	0.6

3.3.5 Writability

The write-trip voltage is a measure of the writability of an SRAM cell [54]. The write-trip voltage is the highest voltage on the bit-line, which can still flip the SRAM cell content. The write-trip voltage is mainly determined by the pull-ups' ratio of the cell [46]. A higher value for the write-trip voltage represents ease of writability, but the write-trip voltage should be sufficiently lower than the supply voltage so noise cannot cause a write failure or a write during a read operation [54].

Table 3.6 shows the write-trip voltage of different configurations in *NICS*. From this table, one can see that the configurations in *NICS* become slightly easier to write, but at the same time write-trip voltage is far enough from the supply voltage to guarantee safe read/write operations.

Table 3.6: Write-trip voltage for NICS cells

Cell	Write Trip Voltage(mV)	% Increase over (0,0,0) cell
(0,0,0)	424	-
(1,0,0)	438	3.3
(0,1,0)	452	6.6
(1,1,0)	466	9.9
(2,0,0)	428	0.9
(2,1,0)	458	8.0
(1,1,2)	443	4.5

3.3.6 Soft Error

The soft error rate of an SRAM cell is obtained from (2.4). In this section we concentrate on Q_{crit} when investigating the effect of increasing the threshold voltage and gate-oxide thickness on SER, since the other parameters in (2.4) are not affected by our proposed technique.

We have used SPICE simulation to measure Q_{crit} of each SRAM cell

configuration. In these simulations, Equation (2.3) is used to model the collection waveform, and T_s is assumed to be 20ps [53]. Table 3.7 reports Q_{crit} for configurations of *NICS*. From this table one can see that Q_{crit} of an SRAM cell is only marginally affected by increasing the threshold voltage or oxide thickness.

Table 3.7: Qcrit for NICS cells

Cell	$Q_{crit}\,(fC)$	**% Decrease over (0,0,0) cell**
(0,0,0)	7.87	-
(1,0,0)	7.40	5.9
(0,1,0)	7.83	0.6
(1,1,0)	7.44	5.4
(2,0,0)	7.56	3.9
(2,1,0)	7.56	3.9
(1,1,2)	7.44	5.4

3.3.7 Cell Type Assignment

To design an HCS, we need to find out the slowest read and write delay starting with all low-V_t SRAM cells (configuration $C_0 = (0,0,0)$). Next, all remaining configurations are sorted in decreasing order of their leakage reduction. Starting from the configuration that results in the highest leakage reduction among all configurations, say (x,y,z), we replace as many $(0,0,0)$ cells as possible with cell (x,y,z) subject to the condition that the access delay of the replaced cells does not exceed the slowest access delay of the SRAM array. Next we try to replace the remaining $(0,0,0)$ cells with the remaining configurations according to the aforesaid order. As long as design rules are met modifying V_t and T_{ox} (i.e., assigning a cell type) does not change the footprint of a cell. Therefore, the cell type assignment does not change the layout of the SRAM cell array.

Figure 3.5 shows the pseudo-code of the heterogeneous cell assignment (HCA).

In this figure, ROW and COL denote the number of rows and columns of the cell array, respectively. If $robustness = 1$, only robust configurations are used in the optimization process of HCA. The fastest cell is denoted by index $[0,0]$, while the slowest one is denoted by index $[COL-1, ROW-1]$. Subroutines $ReadDelay(col, row, C)$ and $WriteDelay(col, row, C)$ return the read and write delays of cell with index of $[col, row]$ when configuration *C* is used. If configuration *C* fails for cell $[col, row]$, then it will fail for all cells $[i, j]$, where $i \geq col$ and $j \geq row$. Therefore, a large number of cells can be pruned as soon as a configuration fails for a given cell. In the pseudo-code, $flag[col, row, C]$ is a flag that specifies if *cell*$[col, row]$ can work with configuration C. Initially all flags are set to 1.

```
HCA(ROW, COL, robustness){
  T_max = ReadDelay(COL − 1, ROW − 1, C_0);
  If (robustness == 1) ConfigSet = NIRCS;
  Else ConfigSet = NICS;
  Sort ConfigSet in decreasing order of leakage saving;
  For each C in ConfigSet do
    For (0 ≤ col < COL, 0 ≤ row < ROW) do
      flag[col, row, C] = 1;
  For col = 0 to COL − 1 do
    For row = 0 to ROW − 1 do
      For each C in ConfigSet do
        If ( flag[col, row, C] == 1)
          If ( ReadDelay(col, row, C) < T_max & WriteDelay(col, row, C) < T_max )
            Replace cell[col][row] with C;
            Break;
          Else
            For (i ≥ col, j ≥ row)
              flag[i, j, C] = 0;
}
```

Figure 3.5: Pseudo-code for the heterogeneous cell assignment.

To speed up the process, instead of checking for possible replacements of each SRAM cell, one can select $2^n \times 2^m$ cell blocks and do the checking for the slowest cell in the block. If the slowest cell passes the delay test, the whole block will be uniformly optimized based on the current configuration; otherwise, the next configuration for the block is examined (in the case that the block fails the delay test for all configurations, it will remain unchanged and the next block will be taken up). Evidently, choosing a larger value for n or m decreases the design time, but may degrade the quality of the final result.

It is noteworthy that using the configurations where the pass transistors have thick gate oxides decreases the word-line capacitance, and thereby, reduces the delay of the word-line. To avoid short-circuit power consumption in the SRAM cell- array (which could occur due to simultaneous activation of the pre-charge and WL drivers), one may have to redesign the timing of these two signals for the cell array. The required modification will, however, be minor.

3.4 Simulation Results

To study the efficiency of the proposed technique, we performed extensive simulations. To reduce the simulation time, all simulations were done on a simplified version of the memory circuit comprising only of elements in the read/write path of a cell; this included the critical path of the decoder, all cells in corresponding row and column of the SRAM array, the corresponding pre-charge devices, column multiplexers, sense amplifiers, write drivers, and the output buffer.

In the first set of experiments, we applied the proposed technique on the SRAM

block described in Section 3.3.1. Table 3.8 shows the leakage power reduction achieved and the percentage utilization of each configuration by the HCA algorithm for two cases *NICS* and *NIRCS* (i.e., non- robust and robust cases) denoted by HCS and RHCS, respectively. As mentioned in Section 3.3.3, for the technology parameters described earlier, the three different criteria that we defined for the robustness resulted in the same set of configurations for RHCS, as shown in Table 3.2-Table 3.4. From Table 3.8 it is seen that the power reduction in HCS and RHCS are 32.6% and 21.2%, respectively.

Figure 3.6 shows the share of subthreshold and tunneling gate currents in the total leakage power dissipation of the conventional SRAM, HCS and RHCS.

Table 3.8: Leakage reduction and the utilization of each configuration in the heterogeneous cell SRAM

	% Leakage Reduction	**% Utilization of Each Configuration**				
		(0,0,0)	(0,1,0)	(1,1,0)	(2,1,0)	(1,1,2)
HCS	32.6	5.0	10.9	21.5	44.3	18.3
RHCS	21.2	60.2	-	21.5	-	18.3

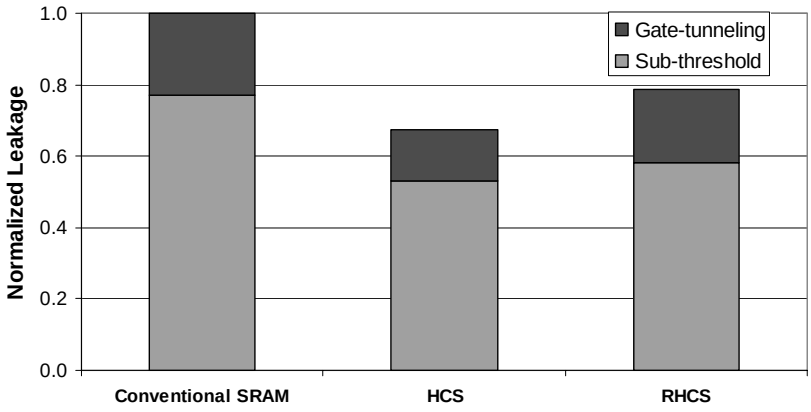


Figure 3.6: Subthreshold and tunneling gate leakage in the conventional and heterogeneous cell SRAM's.

3.4.1 Effect of high-Vt and high-Tox Selection

To study the effect of specific values of high-V_t and high-T_{ox} on the efficacy of heterogeneous cell SRAM technique, we invoked the HCA algorithm with different values of high-V_t and high-T_{ox}. In these experiments, whose results are reported in Table 3.9, we considered three values for high-V_t (i.e., 0.23V, 0.28V, and 0.33V) and three values for high-T_{ox} (i.e., 13Å, 14Å, and 15Å) parameters. For each pair of high-V_t and high-T_{ox}, we ran the HCA algorithm with and without the robustness option. From this table one can see that up to 33% leakage power reduction is achieved by using the HCA algorithm. Furthermore, the power reduction is a weak function of the value of high-T_{ox}. On the other hand, for very high values of high-V_t, power reduction drops. The reason is that in this case the delay overhead of high-V_t configurations becomes too high and these configurations are used less frequently in the SRAM block, which in turn results in less power reduction.

Table 3.9: The Leakage reduction in heterogeneous cell SRAM for different values of high-Vt and high-Tox

(high-V_t, high-T_{ox})	% Leakage Reduction	
	HCS	RHCS
(0.23V, 13Å)	30.3	26.1
(0.23V, 14Å)	31.3	26.1
(0.23V, 15Å)	30.8	25.7
(0.28V, 13Å)	30.8	23.7
(0.28V, 14Å)	32.1	21.2
(0.28V, 15Å)	33.4	20.7
(0.33V, 13Å)	19.1	13.1
(0.33V, 14Å)	19.1	13.1
(0.33V, 15Å)	19.1	13.1

To further study the effect of the specific values of high-V_t and high-T_{ox}, we repeated the simulations for the case that only the dual threshold option is available

in the technology. Table 3.10 shows the power reduction achieved by using the HCA algorithm for three different values of high-V_t. From this table it is seen that the power reduction in this case is still significant and is as high as 24%.

Table 3.10: The Leakage reduction in heterogeneous cell SRAM for a dual-Vt technology

High-V_t	% Leakage Reduction	
	HCS	RHCS
0.23V	22.7	21.0
0.28V	24.5	20.3
0.33V	19.1	13.1

Table 3.11: Leakage reduction in heterogeneous cell SRAM for different values of high-Vt and high-Tox

(high-V_t, high-T_{ox})	% Leakage Reduction			
	HCS		RHCS	
	Two Configs	Three Configs	Two Configs	Three Configs
(0.23V, 13Å)	23.9	27.9	23.9	24.8
(0.23V, 14Å)	23.9	28.3	23.9	24.8
(0.23V, 15Å)	23.9	24.5	23.9	24.5
(0.28V, 13Å)	17.8	22.8	15.0	19.0
(0.28V, 14Å)	20.3	26.0	20.3	21.2
(0.28V, 15Å)	22.0	29.3	20.3	20.8
(0.33V, 13Å)	13.1	19.1	13.1	13.1
(0.33V, 14Å)	13.1	19.1	13.1	13.1
(0.33V, 15Å)	13.1	19.1	13.1	13.1

3.4.2 Effect of the Number of Configurations

Table 3.11 reports the power reduction of the SRAM block for different values of the high-V_t and high-T_{ox} when the number of configurations allowed to be used in the optimized SRAM, including the original configuration, is limited to two or three. As one can see the power reduction is substantial even when only a small number of configurations are used. More precisely, when only two configurations are allowed in the design, 20% power reduction can be achieved; if three configurations can be

used in the optimization process, the quality of the results is comparable with the case that all configurations are used in the cell assignment.

3.4.3 Effect of the Array Size

To further study the efficacy of the HCA algorithm, we conducted another set of experiments for different sizes of the SRAM cell array whose results are reported in Table 3.12. As discussed in Section 3.2, as technology scales, cell arrays are moving from tall to wide structures; so, here we have considered cell array sizes of 32×256, 32×512, 64×256, and 64×512. In all these simulations the values of high threshold voltage and thick oxide are set to 0.28V and 14Å, respectively.

Table 3.12: Summary results for leakage reduction and percentage of replaced cells in HCS for different array sizes

Cell Array Size	% Leakage Reduction	% Replaced Cells
64×256	20.6	90.3
64×512	32.6	95.1
32×256	25.8	94.3
32×512	40.7	96.4

From Table 3.12 one can see that based on the size of cell array, the leakage power reduction resulted from HCA algorithm ranges from 20% to 40%. Moreover, it is seen that the leakage power reduction for a 64×256 cell array is less than that for the 32×256 array. This counter-intuitive result may be explained by noting that when 32 cells are connected to the bit-line, the bit-line becomes less capacitive compared to a 64-cell bit-line. As a result, in a 32-cell bit-line, the delay overheads of some configurations will be less than the delay overheads of them in the 64-cell bit-line (if we use a simple RC model for the delay, changing the threshold voltage of

transistors of a cell, changes the R. Now for a 64-cell bit-line the value of C is higher, therefore, the change in the delay is larger. On the other hand, increasing the length of the bit line due to doubling the number of cells connected to it, has a small effect on the delay difference between the fastest cell and the slowest one. This is because of the fact that SRAM arrays are wide structures and the length of the word line has a higher impact on the delay difference) and hence these configurations will be used more frequently, which in turn results in more power reduction.

3.5 Summary

In this chapter we have presented a novel technique for low-leakage SRAM design. Our technique is based on the fact that due to the non-zero delay of interconnects of the address decoder, word-line, bit-line and the column multiplexers, cells of an SRAM have different access delays. Thus, the threshold voltage or gate oxide thickness of some transistors of cells can be increased without degrading the performance. We showed by using this technique significant power saving can be achieved without scarifying performance or area. We have showed that this leakage saving is a function of the value of high threshold voltage and oxide thickness, as well as the number of rows and columns in the cell array. By applying the proposed technique to a 64Kb SRAM in 65nm technology node, the total leakage power dissipation of the SRAM has been reduced by up to 40%.

Chapter 4
PG-Gated Data Retention SRAM

4.1 Introduction

Aggressive CMOS scaling has resulted low threshold voltage and thin oxide thickness for transistors manufactured in UDSM regime. The leakage power dissipation is roughly proportional to the area of a circuit. As the memories in current technologies occupy a large portion of chip area, their static power dissipation is one of the major components of power dissipation in chips.

In the past much research has been conducted to address the problem of leakage power consumption in SRAM's. Some of these techniques have been reviewed in Chapter 3. Among the dynamic techniques, gating techniques [5, 6, 35, 43, 98, 143] have been proven very effective in power reduction. The key idea of the gated SRAM cell is to disconnect the cell from the supply voltage or ground in the standby mode. If this is done by using a footer NMOS sleep transistor, the technique is called *gated-ground* SRAM; alternatively, if a PMOS sleep transistor is used as header, the technique is called *gated-power-supply*. To retain the value stored in the SRAM cell, it is necessary to strap the gated node, which is either virtual ground or virtual power supply node, to a voltage such that the voltage difference between the rails of SRAM becomes greater than the data retention voltage (DRV) [101]. Source biasing,

dynamic V_{dd}, and drowsy caches are alternative names for data retention gated-ground and gated-power-supply techniques. In these techniques, address decoder can be used to control the gating transistor.

In this chapter we show that although data retention gated-ground and gated-power-supply are effective techniques to suppress the leakage power, there is still much room for improvement. More precisely, we show that by utilizing two sleep transistors instead of one and by selecting appropriate voltage values for biasing the virtual ground and virtual supply nodes, we can achieve higher power saving. We show that, given a fixed value of the voltage difference on the power rails of the SRAM cell during the standby mode, the proposed power-ground (PG)-gating solution achieves significantly higher leakage power savings compared to either power supply (P) gating or ground (G) gating techniques while improving the static noise margin and soft error rate. More precisely, we show that both leakage and hold SNM of a cell vary with the exact values of its supply and ground voltages. As a result, optimum ground and supply voltage levels exist for which the SRAM cell leakage is minimum subject to a hold SNM constraint. When the PG-gated cell is not accessed for read or write operations, it is biased to the optimum values of ground and supply voltages, resulting in minimum leakage power consumption. We show that the PG-gating technique has a higher hold and read SNM, lower soft error rate, and also higher leakage saving compared to single P or G gating techniques at the expense of an increase in the area overhead. Moreover, PG-gated cell exhibits less leakage variability under process and temperature variations compared to single P or G gating techniques. Additionally, its hold SNM is more robust to process variations.

The remainder of this chapter is organized as follows. In Section 4.2 gated-ground and gated-power-supply techniques are reviewed. Section 4.3 introduces the idea of using two sleep transistors to reduce the leakage power consumption of memories. Section 4.4 presents the experimental results, while Section 4.5 summarizes the chapter.

4.2 Single Sleep Transistor Gating Techniques

4.2.1 Gated-Ground SRAM Cell

Based on the discussion presented in Section 3.2.3, one can see that the major leakage currents of an SRAM cell storing "0" are the ones shown in Figure 4.1.

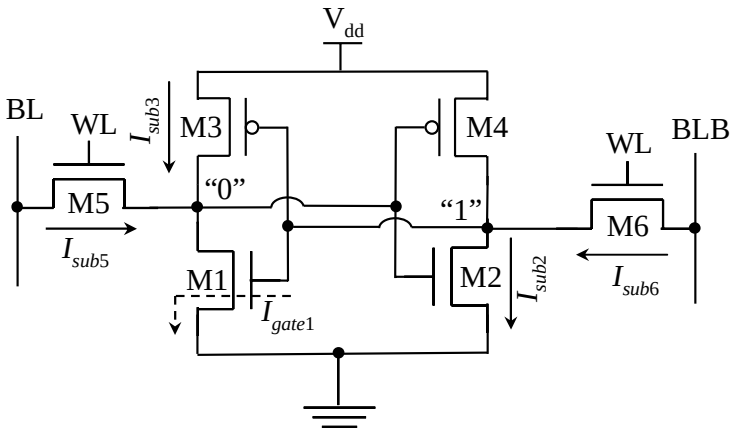


Figure 4.1: Major leakage currents in an SRAM cell storing "0".

Therefore, the total leakage current of an SRAM cell is calculated as,

$$I_{Leak} = I_{sub2} + I_{sub3} + I_{sub5} + I_{sub6} + I_{gate1}\,. \tag{4.1}$$

Notice that if bit and bitbar lines are precharged to V_{dd}, the drain-to-source voltage of M6 becomes zero and according to (2.1) the subthreshold current through this transistor, i.e., I_{sub6}, will be zero.

In a gated-ground data-retention SRAM cell, which we call it *G-gated* cell and is shown in Figure 4.2, an NMOS sleep transistor is used as the footer to disconnect the cell from the ground. In this technique, the bulks of the NMOS transistors are connected to ground to utilize reverse-body biasing effect for reducing subthreshold leakage current. To mitigate the problem of soft errors and coupling noise which may result in losing the value of the bit, an NMOS transistor is added to strap the virtual ground node to a supply voltage $V_G > 0$ when it is not accessed. Both sleep and strapping transistor are controlled by the address decoder (i.e., $\overline{SLP} = WL.$) When $WL = 0$ and the cell is not accessed (i.e., cell is in the *standby mode*), the virtual ground node *GNDV* as well as the node storing "0" are charged to V_G[1]. The increase in the voltage of the virtual ground node as well as the voltage of the left node, compared to the original cell, results in an exponential decrease in I_{sub5}. The reason is threefold: (1) the gate-to-source voltage of M5 becomes negative, which is known as the *stacking effect* (2) the source-to-bulk voltage of M5 increases, resulting in a higher threshold voltage due to the *body biasing effect*, and (3) the drain-to-source voltage of M5 decreases resulting in a higher threshold voltage due to the *DIBL effect*. From (2.1) one can see that each of these effects by itself results in an exponential decrease in the subthreshold leakage of the transistor. The exponential reduction in I_{sub2} is due to the body biasing and DIBL effect because the gate-to-source voltage of the OFF NMOS transistor in both the original cell and G-gated cell

[1] The ground node charges to V_G if $V_G < V_{dd}-V_{t,8}$, where $V_{t,8}$ is the threshold voltage of M8. If V_G is greater than $V_{dd}-V_{t,8}$, M8 should be replaced with a PMOS transistor.

is zero. On the other hand, the reduction in the subthreshold leakage of the OFF PMOS transistor, I_{sub3}, is only due to the DIBL effect only. Since the DIBL coefficient is usually small, the reduction in I_{sub3} is not significant and it will be the main component of leakage current in the G-gated SRAM cell. On the other hand, since the gate-to-source and gate-to-drain voltages of the ON NMOS transistor M1 is reduced, there is also exponential decrease in I_{gate1}.

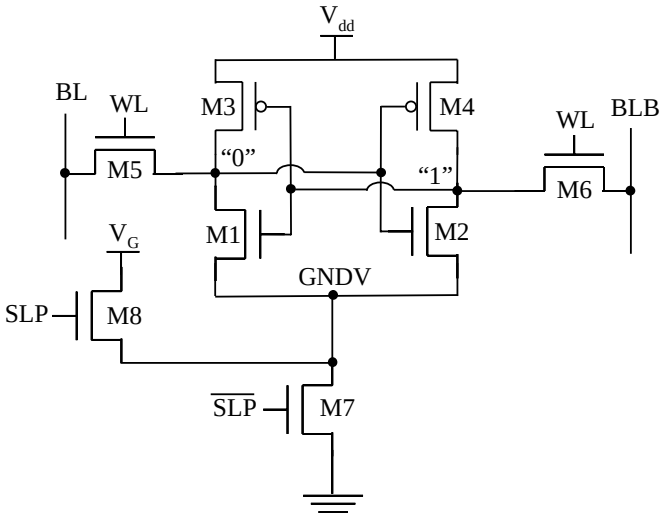


Figure 4.2: G-gated SRAM cell.

From the above discussion it is clear that selecting a higher value for V_G results in a more power efficient cell. However, increasing the voltage of the virtual ground node adversely impacts the hold SNM and in the presence of intra- and inter-die process variation increases the hold failure probability.

4.2.2 Gated-Power Supply SRAM Cell

The second method to reduce the power dissipation of an SRAM cell is to use a PMOS sleep transistor to gate the supply (c.f. Figure 4.3.) Bulks of the PMOS

transistors are connected to V_{dd} to reduce subthreshold leakage of the PMOS transistors as the result of the body effect. In this technique, a PMOS transistor is added to strap the virtual supply node to a supply voltage $V_P < V_{dd}$ when it is not accessed. In the remainder of the chapter this cell is called *P-gated* SRAM cell. If a smaller value is selected for V_P, although tunneling gate current I_{gate1} and subthreshold currents I_{sub2} and I_{sub3} are reduced, I_{sub6} increases and I_{sub5} does not change. However, it should be noticed that since the reduction in I_{sub2} is due to the DIBL effect and the DIBL coefficient is usually very small compared to body-biasing coefficient, the amount of reduction in I_{sub2} is much less than the case of G-gated. From the above discussion one can conclude that the P-gated technique is usually less effective than the G-gated technique (note in the P-gated technique, I_{sub5} does not change.)

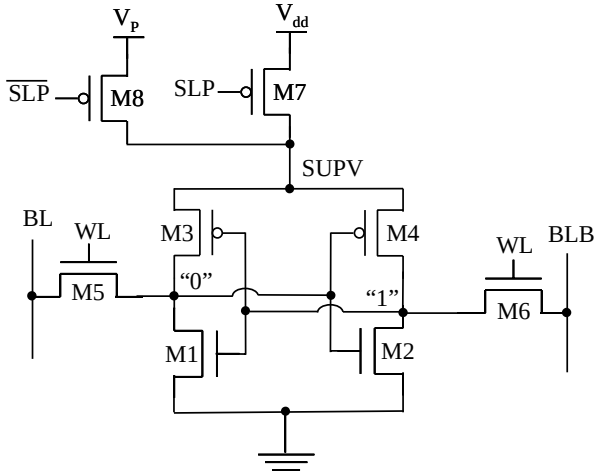


Figure 4.3: P-gated SRAM cell.

Using a P-gated or a G-gated technique involves a tradeoff between area overhead, leakage reduction, and impact on performance [98]. To maintain high

SRAM cell speed, the NMOS sleep transistor in the G-gated cell needs to be sufficiently wide which incurs high area overhead. However, using G-gated technique substantially reduces standby energy dissipation through the self-body biasing, stacking and the DIBL effect of the transistors. On the other hand, using a P-gated technique significantly reduces the required transistor width. But the main advantage of this technique is that since the PMOS transistors of a cell are not contributing in the read operation, a P-gated technique has a marginal impact on the read time of the cell. The disadvantage of this technique is that it does not have any self body biasing or stacking effect on the pull down or pass transistors which translates to a smaller leakage saving compared to the G-gated technique.

4.3 PG-Gated SRAM Cell

In this section we show that in the gating technique, maximum leakage reduction is achieved only when both NMOS and PMOS sleep transistors are used. Figure 4.4 shows a *PG-gated* SRAM cell used in our technique. For the cell to be accessed for a read or write operation, the SLP signal (which is controlled by the address decoder) becomes zero, which causes the voltage of the virtual supply and virtual ground nodes to become V_{dd} and 0, respectively. As soon as the operation is completed, the WL goes to 0, which means that the SLP signal becomes one, and the corresponding cell row enters the standby state. In this state, the strapping transistors M9 and M10 turn on and the voltage of virtual ground and virtual supply become V_G and V_P, respectively. The SRAM cell leakage power is lowered due to source body biasing of the pull-up, pull-down, and access transistors. In this technique, V_P and

V_G are generated by high-efficiency DC-DC converters. In Section 4.3.1 it is shown how to account for the efficiency of the converters.

In PG-gated SRAM cell, like G-gated and P-gated cells, having a smaller potential difference between the two rails of the cell, i.e., $\Delta V = V_P - V_G$, in the standby mode results in lower leakage; however, it also makes the cell more susceptible to noise in the standby mode. To conduct a fair comparison among G-gated, P-gated, and PG-gated SRAM cells, we compare their leakage power reduction and hold SNM when they have equal ΔV in the standby mode.

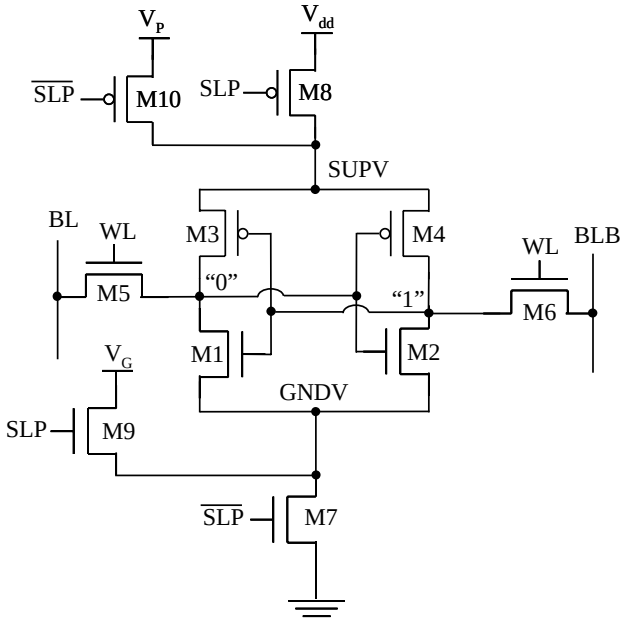


Figure 4.4: PG-gated SRAM cell.

Notice that if ΔV is too low, all six transistors will work in the subthreshold region, but as long as ΔV is greater than the Data Retention Voltage (DRV) of the cell, the data is retained [101].

Consider Figure 4.4 and assume a fixed $\Delta V = V_P - V_G$. If both V_P and V_G are increased such that their difference remains fixed, I_{sub2}, I_{sub5}, and I_{sub6} are

lowered because of the stacking and the body effect, yet I_{sub3} is increased because of the lower bulk-to-source voltage, which results in a more positive threshold voltage for PMOS transistor M3. At the same time, I_{gate1} remains constant because the voltage potential across its gate oxide is constant. On the other hand, if both V_P and V_G are decreased such that their difference remains fixed, I_{sub3} is decreased because of the larger bulk-to-source voltage, while I_{sub2}, I_{sub5}, and I_{sub6} are increased because of the lower bulk-to-source voltage. This shows that when ΔV is fixed, there are optimum values for V_P and V_G for which the leakage power dissipation of the cell is at the minimum. For each value of ΔV, optimal values of V_P and V_G can be found by minimizing (4.1) subject to $V_P - V_G = \Delta V$. To study the effectiveness of the PG-gated SRAM compared to G-gated and P-gated cells in scaled technologies, we simulated the cell using the Predictive Technology Model [99] for 130nm, 90nm, and 65nm technology nodes with 1.3V, 1.2V, and 1.0V as the supply voltage. All results are extracted at 50^{o}C which is the typical temperature for an L2 cache. In these simulations we have assumed both BL and BLB are pre-charged to V_{dd}. Figure 4.5 shows the cell leakage current saving of PG-gated cell compared to G-gated and P-gated cells for different values of ΔV. From this figure, it can be seen that PG-gated is more efficient compared to G-gated and P-gated cells, especially for small values of ΔV (and hence, small values of V_{dd}.) This is useful for ultra low-power applications, which need lower noise immunity. Moreover, from these simulations, one can see that power saving advantage of the PG-gated cell compared to G-gated and P-gated cells is reduced with technology scaling. This

unexpected result occurs because for the predictive technology model that we are using [99], the subthreshold swing is smaller in 130nm technology compared to 90nm and 65nm. As a result, the stacking effect achieves a higher subthreshold leakage saving for the 130nm node.

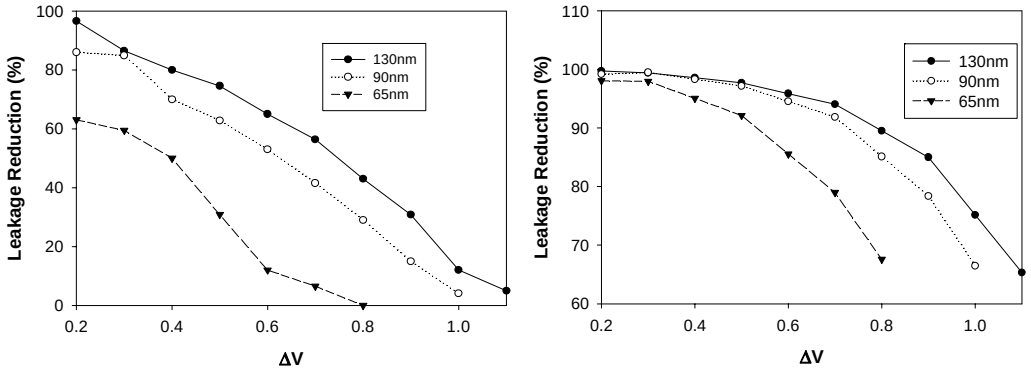


Figure 4.5: Cell leakage current reduction of PG-gated SRAM cell compared to (a) G-gated and (b) P-gated cells.

4.3.1 Optimum PG-Gated SRAM Cell Design

Although minimizing (4.1) subject to $V_P - V_G = \Delta V$ results in the minimum leakage SRAM cell, the equation does not consider the leakage currents of the additional circuitry, nor does it account for the non-ideal efficiency of the DC-DC converters used to generate V_G and V_P from V_{dd}. In this section, we accurately model the leakage power consumption of the PG-gated cell architecture with all these factors properly accounted for.

We start by deriving the current drawn from each power supply. By writing a KCL at the virtual supply node, the current flow from the source to drain of M10, which is the current drawn from V_P, may be written as,

$$I_P = I_{sub2} + I_{sub3} + I_{gate1} - I_{sub6} - I_{sub8}\,. \tag{4.2}$$

Similarly, by writing a KCL at the virtual ground node, the current flow from drain to source of M9, which is the current drawn from V_G, may be written as,

$$I_G = I_{sub7} - (I_P + I_{sub5} + I_{sub6} + I_{sub8})\,. \tag{4.3}$$

On the other hand, the current drawn from V_{dd} is simply,

$$I_{dd} = I_{sub8} + I_{sub5} + I_{sub6} + I_{gate9}\,. \tag{4.4}$$

From (4.2)–(4.4), the total power leakage power consumption of PG-gated SRAM cell in the standby mode may be expressed as,

$$P_{cell} = \frac{I_P V_P}{\delta_P} + \frac{I_G V_G}{\delta_G} + I_{dd} V_{dd} \tag{4.5}$$

where δ_P and δ_G are the efficiency of DC-DC converters used to generate V_P and V_G from V_{dd}, respectively. The efficiency of a DC-DC converter depends on different parameters (including the type of converter, its actual operating point compared to the optimum operating point, type of components used, etc.), but it is usually between 80% and 90% [122].

Now, the problem of minimizing the leakage power consumption of the PG-gated SRAM cell, considering the efficiency of DC-DC converters, can be expressed as,

$$\begin{cases} \min & P_{cell}(V_P, V_G) \\ s.t. & V_P - V_G = \Delta V \end{cases} \tag{4.6}$$

Since the difference of V_P and V_G is constant, to solve (4.6) the objective function may be expressed as unconstrained minimization of $P_{cell}(V_G + \Delta V, V_G)$. This problem can be solved by using standard unconstrained optimization

techniques, such as the Newton-Raphson technique.

To be able to solve (4.6), the size of the sleep and strapping transistors should be known. The NMOS sleep transistor is in the read path should be sufficiently large to result in low delay penalty. Since the PMOS sleep transistor is in the write path and the write delay of SRAM cell is usually lower than the read delay, the PMOS sleep transistor can be made smaller than the NMOS sleep transistor. On the other hand, the strapping transistors M9 and M10 are turned on during the standby mode and because entering the standby mode is not a time-critical step, this transistor can be made very small [70]. Figure 4.6 shows the leakage power reduction of PG-gated cell compared to G-gated and P-gated cells for different values of ΔV in different technology nodes. To have fair comparisons, the size of the sleep transistors in G-gated and P-gated cells have been selected to be equal to the size of the NMOS and PMOS sleep transistors of the PG-gated cell, respectively. Moreover, minimum sized transistors which are shared among eight cells are used as the strapping transistors. Additionally, the efficiency of DC-DC converters to generate V_P and V_G assumed to be 80%.

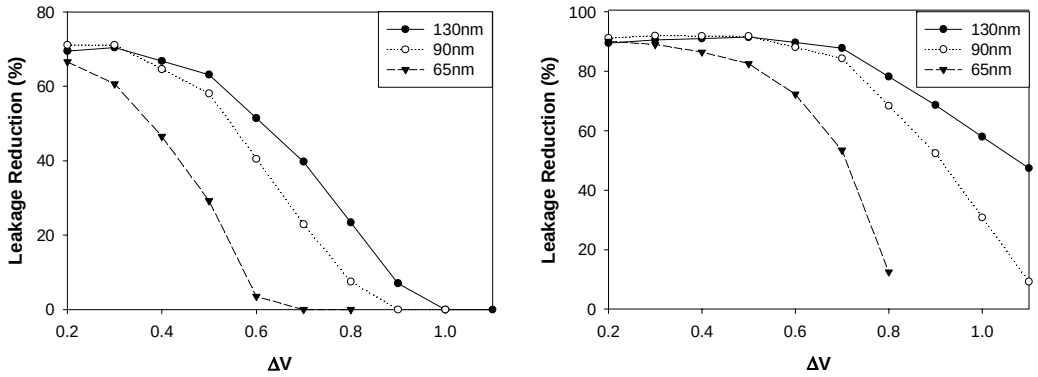


Figure 4.6: Power reduction of PG-gated SRAM cell compared (a) G-gated and (b) P-gated cells.

From Figure 4.6 one can see that despite the overhead of DC-DC converters, the efficiency of PG-gated cell compared to P- and G-gated cells is quite high, specially when $\Delta V \leq 0.5V$ which is greater than the DRV.

Since the power saving of P-gated is poor compared to PG-gated and G-gated cells, in the remainder of this chapter, we focus on G-gated technique.

4.3.2 Static Noise Margin

To investigate the hold SNM of the PG-gated cell and compare it with those of P-gated and G-gated cells, notice that in the PG-gated cell the hold SNM is not only a function of the difference between V_P and V_G, but also depends on their absolute values. The reason is that in this case, by changing the values of V_P and V_G, the threshold voltage of transistors change as a result of the body effect. Since the SNM is a function of the threshold voltage of transistors inside the cell, it is also affected by tuning V_P and V_G. Figure 4.7 shows the hold SNM of the PG-gated cell as a function of ΔV. In each curve, the rightmost point corresponds to data value for a G-gated cell while the leftmost point corresponds to data value for a P-gated cell. For each ΔV, $V^*_{G,G-gated}(\Delta V)$ is defined as the minimum V_G in interval $[0, V_{dd} - \Delta V]$ for which the hold SNM of the PG-gated cell is greater than that in the G-gated cell. Similarly, $V^*_{G,P-gated}(\Delta V)$ is defined as the maximum V_G in interval $[0, V_{dd} - \Delta V]$ for which the hold SNM of PG-gated cell is greater than that in the P-gated cell. Values of $V^*_{G,G-gated}(\Delta V)$ and $V^*_{G,P-gated}(\Delta V)$ can be obtained from hold SNM versus V_G curves (c.f. Figure 4.7). For example, from

Figure 4.7, it is seen that $V^*_{G,G-gated}(0.3) = 150mV$ and $V^*_{G,P-gated}(0.3) = 1.0V$. With these definitions, it is clear that if the PG-gated cell is designed in such a way that its virtual ground voltage level V_G is greater than $V^*_{G,G-gated}(\Delta V)$ and less than $V^*_{G,P-gated}(\Delta V)$, then its hold SNM will become larger than those of P- and G-gated cells.

To guarantee that the hold SNM of the resulting PG-gated cell is not lower that those of the P-gated or G-gated cells, a second constraint should be added to (4.6):

$$V_G \leq V^*_{G,P-gated} \quad \text{(for P-gated cell)}$$
$$V_G \geq V^*_{G,G-gated} \quad \text{(for G-gated cell)} \tag{4.7}$$

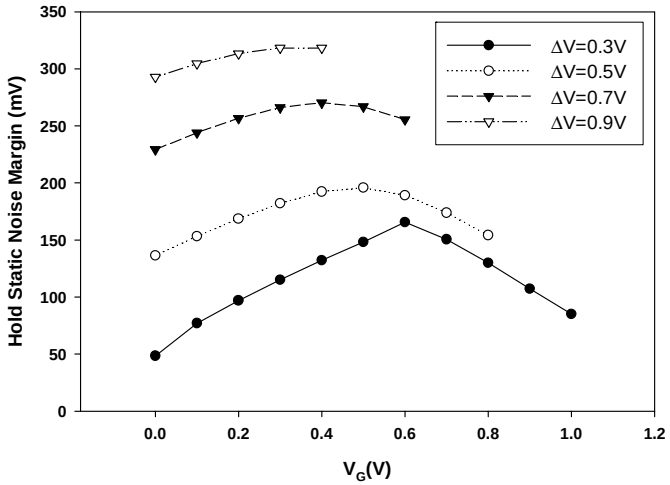


Figure 4.7: Hold static noise margin of PG-gated cell as a function of VG.

Values of $V_G \geq V^*_{G,G-gated}$ and $V_G \leq V^*_{G,P-gated}$ are extracted from the hold SNM versus V_G curves as shown in Figure 4.7. The new mathematical program to minimize leakage becomes a constrained optimization problem. Since there is only

one parameter V_G in this formulation, the optimization problem can be solved efficiently by using standard numerical optimization techniques.

4.3.3 Soft Error

In this section we concentrate on Q_{crit} when investigating the effect changing the voltage of virtual ground and virtual supply nodes of an SRAM cell, since the other parameters of (2.4) are not affected by utilizing the gating technique. Notice that virtual ground and virtual supply nodes are shared among some cells in a row. Therefore, these nodes are highly capacitive which make them soft error immune; therefore, in this section we investigate SER in the internal nodes of SRAM cells which are susceptible to soft error.

We have used SPICE simulation to measure Q_{crit} of PG-gated SRAM cell for different values of V_G when ΔV is fixed. In these simulations, Equation (2.3) is used to model the collection waveform, and T_s is assumed to be 30ps [53].

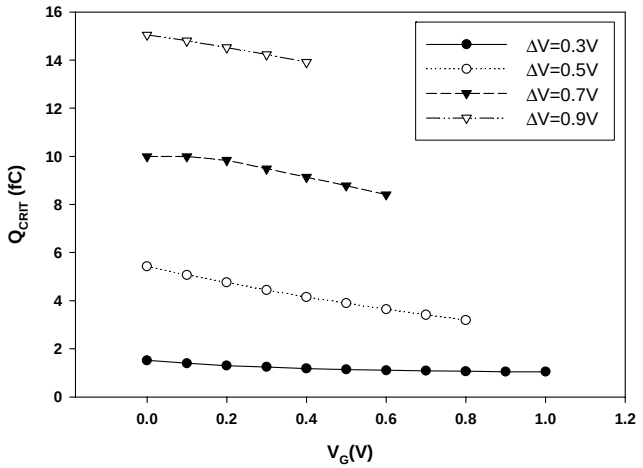


Figure 4.8: Critical charge of PG-gated cell as a function of V_G.

Notice that in each curve, the rightmost point corresponds to data value for a G-gated cell while the leftmost point corresponds to data value for a P-gated cell. From these curves one can see that when ΔV is fixed, Q_{crit} is a decreasing function of V_G; therefore, Q_{crit} of a PG-gated SRAM cell is larger than that in the G-gated cell and consequently SER of a PG-gated SRAM cell is lower than SER in a G-gated cell.

4.3.4 Effect of Temperature

It is known that the subthreshold leakage current is an exponential function of the temperature. To study the effect of temperature on gated SRAM cells, we have simulated the PG-gated and G-gated cells for temperatures ranging from 20°C to 100°C in 130nm node when $\Delta V = 0.5V$. From the results, which are presented in Figure 4.9, it can be seen that PG-gated SRAM cell has much lower sensitivity to temperature variations compared to the G-gated Cell.

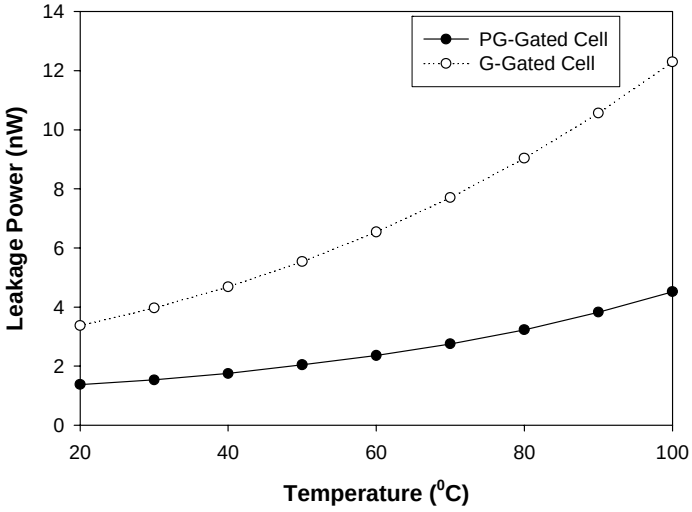


Figure 4.9: Leakage variation of PG-gate and G-gated SRAM cells as a function of chip temperature.

4.3.5 Effect of Process Variation

To study the effect of the process variation on PG-gated and G-gated SRAM cells, we modeled the threshold voltage of each transistor, including the sleep and strapping ones, as independent Gaussian random variable whose standard deviation is obtained from (3.1) by assuming $3\sigma_{\min} = 100mV$.

We performed a Monte Carlo simulation of 5000 samples to obtain the leakage power consumption and hold SNM under these variations. Figure 4.10 and Figure 4.11 show leakage power and hold SNM distribution of PG-gated and G-gated SRAM cells when $\Delta V = 0.5V$.

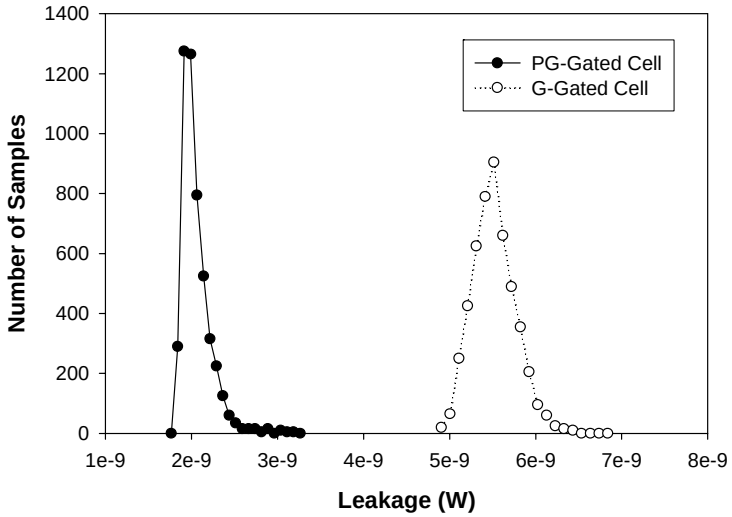


Figure 4.10: Leakage variation of PG-gate and G-gated SRAM cell.

From Figure 4.10 it can be seen that the mean and standard deviation of leakage power consumption in PG-gated cell are 2.1nW and 0.17nW, whereas those values for G-gated cell are 5.57nW and 0.25nW, respectively; so, using PG-gated technique results in 63% reduction in the mean and 32% reduction in the standard deviation of the leakage power consumption. On the other hand, from Figure 4.11, it is clear that

under process variation PG-gated cell is more robust than the G-gated cell, resulting in less hold failures.

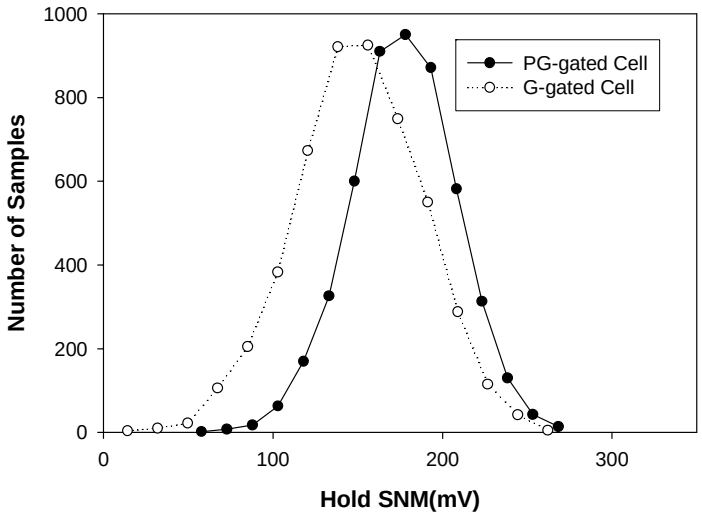


Figure 4.11: Hold static noise margin variation of PG-gate and G-gated SRAM cells under process variations.

4.4 Experimental Results

To study the efficacy of the proposed technique and its tradeoffs, we designed and simulated three 64Kb SRAM modules in 130nm technology, comprising of: (1) a conventional SRAM cell, (2) a data retention G-gated SRAM cell, and (3) a data retention PG-gated SRAM cell.

Each SRAM module is composed of two blocks of 256×128 cells each. The voltage of the power supply in all cases is 1.3V. Both gated SRAM's were designed to have $\Delta V = 0.5V$, which results in about 150mV SNM in the standby mode for the G-gated cell (c.f. Figure 4.7.) Based on this predefined ΔV, the optimum values of V_P and V_G for the PG-gated cell have been obtained by solving (4.6) considering the constraint (4.7).

The final design parameters of G-gated and PG-gated SRAM are shown in Table 4.1. In this table, λ=65nm. Moreover, $W_{sleep,N}$ and $W_{sleep,P}$ denote the widths of the NMOS and PMOS sleep transistors, respectively. The NMOS sleep transistors in both cells are equally sized to yield equal read static noise margin. Since the WL ($\overline{WL}$) is used to drive the sleep and strapping transistors in a gated SRAM, the load of the decoder in a gated SRAM is more than a conventional SRAM. Therefore, the decoders of the gated SRAM's have been resized to produce minimum delay for the new load. To amortize the area overhead of the sleep transistors and also to have lower read access penalty, the sleep transistors of a row are shared as a single transistor. The metal line, which is used as a ground line in a conventional SRAM, is used to connect the drain of the G-gated transistor to the SRAM cells. The drain of the gated-supply transistor is connected to the SRAM cells in a similar manner. Strapping transistors are also shared and their size is 10% of the total size of the sleep transistors.

Table 4.2 compares the area, delay, read SNM, hold SNM, leakage, and Q_{crit} of PG-gated cell with those of G-gated cell. The values of area and delay are normalized to those of the conventional SRAM. The values of the hold SNM have been extracted from Figure 4.7. It can be seen that using the PG-gated cell results in 18% increase in the hold SNM. From the table, one can see that PG-gated SRAM incurs 7.4% area overhead, whereas the overhead of the G-gated SRAM is 3.5%. This is due to using both NMOS and PMOS sleep and strapping transistors in the PG-gated SRAM.

Also it can be seen that the mean and standard deviation of leakage power

consumption in PG-gated cell are 2.1nW and 0.17nW, whereas those values for G-gated cell are 5.57nW and 0.25nW, respectively; so, using PG-gated technique results in 63% reduction in the mean and 32% reduction in the standard deviation of the leakage power consumption. On the other hand, from this table it can be seen that the delay overhead of PG-gated SRAM is slightly worse than its counterpart in G-gated SRAM (e.g., 3.2% versus 2.7%.) This is because in the PG-gated cell, the word line drives more capacitance which results in higher delay in the decoder.

Table 4.1: Design parameters of the G-gated and PG-gated SRAM's

Parameter	**G-gated**	**PG-Gated**
V_P	1.3V	0.8V
V_G	0.8V	0.3V
$W_{sleep,N}$	3.5 λ	3.5λ
$W_{sleep,P}$	–	2.0 λ

Table 4.2: Comparison of G-gated and PG-gated SRAM's

Parameter	**G-Gated SRAM**	**PG-Gated SRAM**	**Improvement (%)**
Area (Normalized)	1.035	1.074	-3.7
Delay (Normalized)	1.027	1.032	-0.5
SNM_{read}	185mV	186mV	0.5
SNM_{hold}	154mV	182mV	18.2
Leakage (mean)	5.57nW	2.1nW	62.3
Leakage (std. dev.)	0.25nW	0.17nW	32.0
Q_{CRIT}	3.19fC	4.44fC	39.2

From Table 4.2 it also can be seen that the read SNM of the PG-gated and G-gated SRAM are marginally better than that of conventional SRAM cell (the read SNM of conventional SRAM is 179mV). This is because in the gated SRAM's, the stacking effect due to the NMOS sleep increases the threshold voltage of the pull-

down NMOS transistors. Higher V_t in NMOS transistor makes it more difficult for the access transistor to destroy the data, which translates into a higher SNM [72].

If we take into account the leakage overhead of using a larger address decoder in the gated SRAM's, total power saving of the G-gated technique is 10X while the saving of the PG-gated SRAM is 25X. In other words, the power consumption of a PG-gated SRAM is 40% of the power consumption of a G-gated SRAM design. Finally one can see that the proposed PG-gated technique improves Q_{crit}, which is an important parameter in SER, by 39.2%.

4.5 Summary

In this chapter we presented a novel gating technique, called PG-gating to reduce the leakage power consumption of SRAM cells. We showed that previously proposed gating techniques do not achieve the optimum leakage saving for a fixed value voltage difference between the power rails and that the maximum saving can be achieved when both NMOS and PMOS sleep transistors are used. We demonstrated the efficacy of our technique, compared to other gating techniques, in different technology nodes. We also showed that a PG-gated cell potentially has a larger hold static noise margin. To further study the proposed technique, we designed two 64Kb SRAM modules based on G-gated and PG-gated techniques. Our simulation results show that with small area and read access delay penalties, PG-gated technique achieves 60% lower leakage compared to that of G-gated technique while improving the hold SNM by 18% and Q_{crit} by about 39%.

Chapter 5
Low-Power Fanout Optimization

5.1 Introduction

Very often in VLSI circuits, a signal needs to be distributed to several destinations under a required timing constraint at each destination. In practice, there may also be a limitation on the load that can be driven by the source signal. Fanout optimization is the problem of building an inverter tree topology between a source and some sinks and sizing the inverters so that the driving capacitance at the source is less than an upper bound and the timing constraints at sinks are met, while an objective function is minimized [11, 104, 108]. Different objective functions have been considered for the fanout optimization problem, such as minimizing area [104, 116, 147], minimizing power consumption [12, 147], and minimizing load on the source [75].

Unlike buffer insertion which is a back-end process and is performed after the global routing when the interconnect information is available, fanout optimization is performed during logic synthesis often interleaved with the technology mapping process in order to provide the global placer with accurate information about the number and sizes of the logic gates in the netlist.

The fanout optimization problem to achieve minimum area for libraries with discrete sizes has been proven to be NP-complete [21, 131]. However, it has been

shown that using an inverter library with near-continuous sizes greatly simplifies the problem [73]. More precisely, the assumption of near-continuous library allows one to model the problem as a mathematical optimization problem with continuous variables and solve it efficiently. With utilizing a near-continuous library, the mapping of optimized continuous variables to discrete ones in the library will be near optimal.

Several techniques have been proposed to address the fanout optimization problem using simplified delay models. In [124], for example, the delay of a single path has been minimized by assigning equal delay budgets to each buffer on the path. While it is known this approach minimizes the delay from the source to any sink, it does not necessarily result in an optimal solution in terms of other objective functions such as area or power dissipation. Reference [75] introduced two transformations, namely *merging* and *splitting*, used to convert any fanout tree to a set of inverter chains. It was shown that these transformations maintain the area, delay, and input capacitance. Using the transformation introduced in [75], reference [104] proposed a logical effort-based fanout optimizer for area which attempts to minimize the total buffer area under the required time and input capacitance constraints.

Although much research has been done to address fanout optimization problem, there is little work on low-power fanout optimization. More specifically, since both capacitive and leakage power dissipation of a fanout chain are proportional to its area, it has been widely accepted that power minimization of the fanout tree is equivalent to its area optimization [12, 147]. In this chapter, however, we show that

due to short-circuit power dissipation, minimizing area does not necessarily result in a minimized power dissipation solution. In particular, the solution obtained from an area optimized fanout tree may dissipate excessive short-circuit power. We formulate the problem of minimizing the power dissipation of a fanout chain and show how to build a fanout tree out of these power-optimized chains. Additionally, to suppress the leakage power dissipation in a fanout tree, we use multi channel length (L_{Gate}) [118] and multi-V_t techniques. In the presence of multi-L_{Gate} and multi-V_t options, we accurately model the delay and power dissipation of inverters as posynomials; therefore, our proposed problem formulation results in a convex mathematical program comprising of a posynomial objective function with posynomial inequality constraints which can be efficiently solved.

When there is only one sink, the fanout tree is reduced to a chain of inverters between the source and sink and the fanout optimization problem becomes that of finding the number and sizes of the inverters to satisfy the input capacitance and timing constraints while minimizing some objective function such as area or power dissipation. For multiple sinks, on the other hand, by using the split and merge transformations [75] or by limiting the types of the fanout trees to the so called LT-trees [131], a fanout tree can be constructed from the inverter chains. In this chapter we use *fanout chain* to describe the fanout topology with one sink and *fanout tree* to describe it when there are multiple sinks.

The remainder of the chapter is organized as follows. Section 5.2 describes logical effort technique and its extension for handling multi-V_t and multi-L_{Gate} circuits. It further describes the power model that will be used throughout the chapter.

Section 5.3 investigates the problem of minimizing the area of a fanout chain and shows that a minimized area fanout chain may dissipate excessive short circuit power. Section 5.4 formulates the problem of low-power fanout chain optimization (i.e., when there is only one sink) and shows how to optimize the power consumption of the fanout chain by utilizing multi-V_t and multi-L_{Gate} techniques. Section 5.5 shows how a low-power fanout tree can be constructed from the fanout chains. Simulation results are given in Sections 5.6 and Section 5.7 summarizes the chapter.

5.2 Delay and power Models

5.2.1 The Delay Model

The delay model we use in this chapter is based on logical effort [124]. The logical effort is a technique for modeling and analyzing delay in CMOS circuits and has been widely used to solve a variety of synthesis problems including technology mapping [59, 69], gate sizing [32], and fanout optimization [12, 75, 104]. Additionally, it has also been incorporated in some industry synthesis tools [81, 121]. In this section we first review this model and then describe its extension to handle multi-V_t and multi-L_{Gate} techniques.

Using the notion of logical effort, the delay of a gate with input capacitance C_{in}, which drives the load capacitance C_L, is modeled as,

$$D = \tau_0(p + gh) \tag{5.1}$$

where τ_0 is a conversion coefficient that characterizes the semiconductor process being used and converts the unit-less part, $p + gh$, to a time unit. For the sake of simplicity, in the remainder of this chapter, we set τ_0 to one. Parameter p denotes

the parasitic delay of the gate. The major contributor to the parasitic delay is the capacitance of the source/drain regions of the transistors that drive the output. Parameter g denotes the *logical effort* of the gate which depends only on the topology of the gate and its relative ability to produce output current. More precisely, the logical effort of a gate shows how worse it is at producing output current than an inverter if each of its inputs has the same input capacitance as the inverter. Finally, parameter h denotes the *electrical effort* of the gate and is defined as the ratio of the output capacitance of the gate to its input capacitance, i.e., $h = C_L / C_{in}$. The electrical effort describes how the electrical environment of the logic gate affects performance and how the size of the transistors in the gate determines its load-driving capability.

For an inverter, the value of logical effort g equals one and can be shown that p is the ratio of output diffusion capacitance to input gate capacitance of the template inverter, denoted by $p_0 = C_{diff,T} / C_{in,T}$. Notice that since both input gate and diffusion capacitances of an inverter are scaled linearly by changing the inverter's size, for a scaled inverter, the ratio of diffusion-to-gate capacitance remains constant, i.e.,

$$C_{diff} / C_{in} = p_0 \tag{5.2}$$

where C_{diff} is the diffusion capacitance at the output and C_{in} is the gate capacitance at the input. In the following, we show how to extend the concept of logical effort to handle multi-V_t and multi-L_{Gate} technologies.

It is known that when the threshold voltage of a gate is changed, the new delay

can be obtained from the alpha-power law [107] by the following equation,

$$d_i = d_0 \frac{\left(V_{dd} - V_{t,0}\right)^{\bar{\alpha}}}{\left(V_{dd} - V_{t,i}\right)^{\bar{\alpha}}} \tag{5.3}$$

where $\bar{\alpha}$ is a technology parameter which is around 2 for long channel devices and 1.3 for short channel devices, V_{dd} is the supply voltage, $V_{t,0}$ is the nominal threshold voltage, d_0 is the delay under the nominal threshold voltage, $V_{t,i}$ is an arbitrary threshold voltage, and d_i is the delay under the arbitrary threshold voltage. Using Equations (5.1) and (5.3) one can verify that in a multi-V_t technology, the values of the logical effort and parasitic delay change as follows,

$$g_i = \frac{(V_{dd} - V_{t,0})^{\bar{\alpha}}}{(V_{dd} - V_{t,i})^{\bar{\alpha}}},\ p_i = p_0 \frac{(V_{dd} - V_{t,0})^{\bar{\alpha}}}{(V_{dd} - V_{t,i})^{\bar{\alpha}}} \tag{5.4}$$

where g_i and p_i are the logical effort and parasitic delay for an arbitrary threshold voltage, $V_{t,i}$.

Equations (5.1) and (5.4) are based on the assumption that the channel length of the gate, L, is equal to the nominal channel length of the technology, L_{nom}. In a multi-L_{Gate} technology, however, the delay of a logic gate is an increasing function of the channel length. Our SPICE simulations show when the channel length of an inverter is increased, the new delay can be obtained from the following equation,

$$d_l = d_0 l^{\beta_d} \tag{5.5}$$

where l is the normalized channel length, i.e., $l = L_{Gate} / L_{nom}$ and β_d is a fitting parameter. Moreover, d_0 is the delay under the nominal channel length, while d_l is the delay of the gate with the normalized channel length l. Figure 5.1 demonstrates

the validity of this delay model. Using Equation (5.5), one can easily establish that in a multi-L_{Gate} technology, values of the logical effort and parasitic delay change as follows,

$$g_l = l^{\beta_d}\,,\; p_l = p_0 l^{\beta_d}\,. \tag{5.6}$$

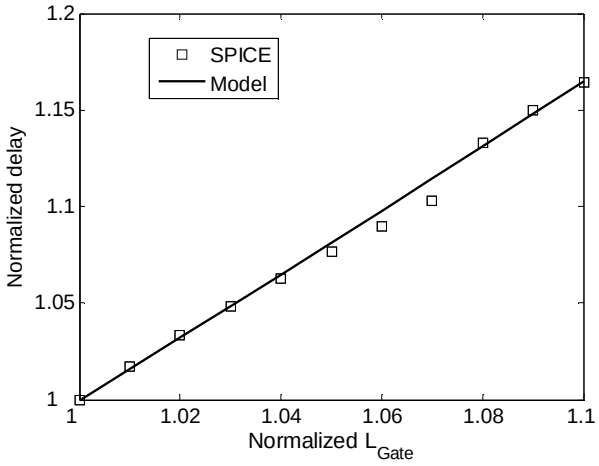


Figure 5.1: Delay as a function of channel-length.

5.2.2 Power Dissipation Model

The power dissipation of a CMOS gate has three components: capacitive power, short circuit power, and leakage power.

5.2.2.1 Capacitive Power Dissipation

The capacitive power dissipated in inverter capacitances, i.e., input gate capacitance and output diffusion capacitance, is equal to,

$$P_{cap} = \alpha f V_{dd}^2 C \tag{5.7}$$

where α is the switching activity of the inverter, f is the frequency, V_{dd} is the supply voltage, and C is the sum of the input gate capacitance and output diffusion capacitance of the inverter, i.e., $C = C_{diff} + C_{in}$. By using (5.2), Equation (5.7) can

be re-written as,

$$P_{cap} = \alpha f V_{dd}^2 (1 + p_0) C_{in} = k_{cap} C_{in} \tag{5.8}$$

In a multi-L_{Gate} technology, the input gate capacitance of the inverter increases as a result of biasing the channel length, while the diffusion capacitance remains unchanged. Therefore, the capacitive power dissipation is obtained from,

$$P_{cap,l} = k_{cap} \frac{l + p_0}{1 + p_0} C_{in} \tag{5.9}$$

where C_{in} denotes the input capacitance of the inverter under nominal gate-length.

5.2.2.2 Short-Circuit Power Dissipation

The second source of power dissipation in digital circuits is short-circuit current. Short circuit power is consumed by the current flow between the power rails through a direct path which is temporarily established during an input transition [92]. If a circuit is *well-designed*, its short-circuit power dissipation is about 10%-20% of the capacitive power dissipation [96]. Several analytical techniques have been proposed to address the problem of short circuit power estimation [3, 92, 96, 125, 136], but due to their complexity, their use tend to be impractical during gate-level optimization. In this chapter, by observing the fact that short-circuit power dissipation of an inverter is a linear function of its size and input transition time [96] and also the fact that input transition time itself can be approximated as a linear function of the electrical effort of its fanin gate (see Figure 5.2), the short-circuit power dissipation of the i^{th} inverter in a chain is calculated as,

$$P_{sc} = \alpha A_{sc} h_{i-1} f V_{dd} C_{in} = k_{sc} h_{i-1} C_{in} \tag{5.10}$$

where A_{sc} is the short-circuit factor which is a technology-dependent parameter,

h_{i-1} is the electrical effort of the $(i-1)^{\text{th}}$ inverter and C_{in} is the input capacitance of the i^{th} inverter. From Figure 5.2 one can see that this technique, despite its simplicity, is accurate enough to be used in gate-level optimization.

From Equations (5.8) and (5.10), one can see the ratio of the short-circuit to the capacitive power dissipation of an inverter can be expressed as,

$$\frac{P_{sc}}{P_{cap}} = \frac{k_{sc}}{k_{cap}} h_{i-1}. \tag{5.11}$$

For various values of h_{i-1} this ratio is plotted in Figure 5.2.

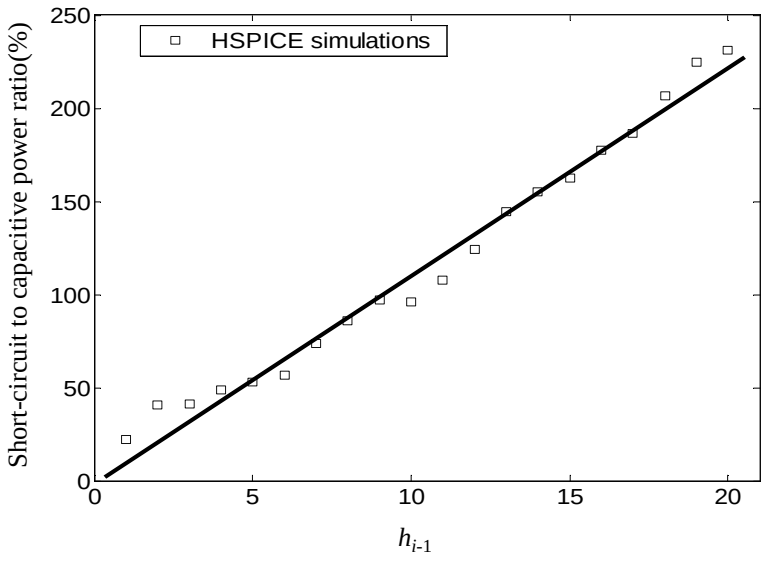


Figure 5.2: Percentage ratio of short-circuit to capacitive power dissipation of i'th inverter, as a function of electrical effort of previous stage.

It should be noted that in a multi-V_t inverter chain, the short-circuit power dissipation, and consequently, k_{sc} of the i^{th} inverter (henceforth, denoted as $k_{sc,i}$) is a function of the threshold voltages of the i^{th} inverter and its driver (i.e., the $(i-1)^{\text{th}}$ inverter). If there are m threshold voltages in the library, then there will be

m^2 distinct values for $k_{sc,i}$'s.

Utilizing longer channel length for PMOS and NMOS transistors in a CMOS inverter increases the threshold voltage of both transistors; therefore, the time during which both NMOS and PMOS transistors are ON during the output transition is decreased. Thus, the short-circuit power consumption of the inverter is reduced. On the other hand, since the output slew time of an inverter increases when using a longer channel length, the short circuit power of the fanout gate increases. Therefore, in an inverter chain, the short-circuit power dissipation of the i^{th} inverter is inversely proportional to the channel length of the inverter, i.e., l_i, and directly proportional to the channel length of its driver, i.e., l_{i-1}. Based on these observations, we model the short-circuit power dissipation of the i^{th} inverter in a chain as,

$$P_{sc} = k_{sc} h_{i-1} l_i^{-\beta_{sc1}} l_{i-1}^{\beta_{sc2}} C_{in} \tag{5.12}$$

where β_{sc1} and β_{sc2} are technology constants found by fitting (5.12) to data extracted from SPICE level simulations.

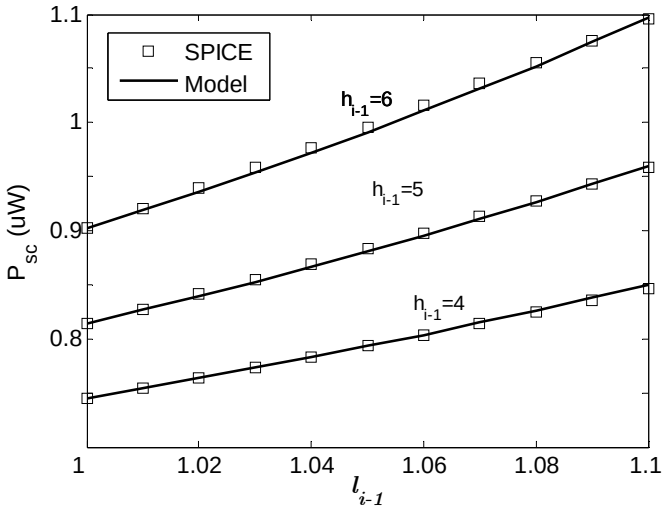


Figure 5.3: Short-circuit power dissipation as a function of driver channel length.

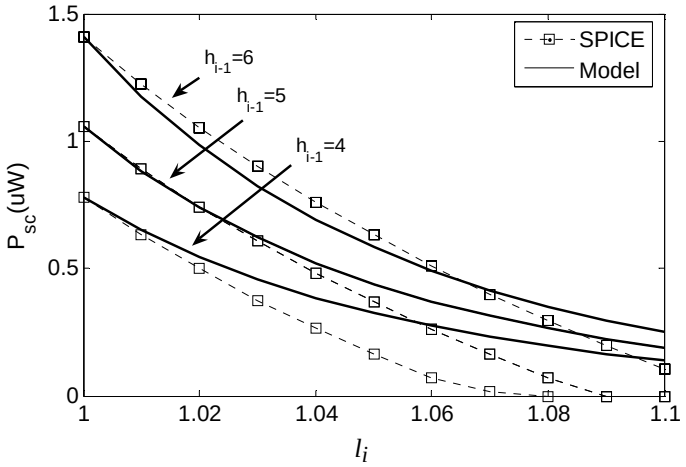


Figure 5.4: Short-circuit power dissipation as a function of channel length.

Figure 5.3 and Figure 5.4 compare (5.12) with the actual SPICE data for various values of l_{i-1} and l_i. It should be mentioned that although the accuracy of the model is reduced for large l_i 's, since for these values of l_i the short-circuit power dissipation becomes quite small compared to the capacitive power, the error in the total power consumption model remains small.

5.2.2.3 Leakage Power Dissipation

The subthreshold leakage current of an MOS transistor is obtained from (2.1). Let C_N denote the input capacitance of an NMOS transistor. Since V_{ds} of the OFF transistor is V_{dd} which is more than a few $kT/q \approx 26mV$ and noting that in an NMOS transistor $w_N = C_N/(L_{eff}C_{ox})$, the subthreshold leakage power of an NMOS transistor as a function of its gate capacitance can be written as,

$$P_{sub,N} = A'_{sub}\, C_N \mu_N e^{-\lambda\, V_{t0,n}} \tag{5.13}$$

where $\lambda = q/n'kT$ and $A'_{sub} = A_{sub}V_{dd}/L_{eff}^2 \exp(\lambda\eta V_{dd})$ are technology

constants. A similar formula can be derived for the subthreshold leakage power of a PMOS transistor. From the subthreshold leakage power expressions for the NMOS and PMOS transistors, the subthreshold leakage power dissipation of an inverter, P_{sub}, can be written as,

$$P_{sub} = pP_{sub,P} + (1-p)P_{sub,N} \tag{5.14}$$

where p is the probability that the input of the inverter is at logic 1. If the ratio of the width of the PMOS transistor to that of the NMOS transistor is β, , i.e., $w_P / w_N = \beta$, by considering the fact that for an inverter $C_{in} = C_N + C_P$, Equation (5.14) can be re-written as,

$$P_{sub} = \frac{A'_{sub}}{1+\beta}\left(p\beta\mu_P e^{-\lambda V_{t0,p}} + (1-p)\mu_N e^{-\lambda V_{t0,n}}\right)C_{in} = k_{sub}C_{in}\,. \tag{5.15}$$

From (5.15) one can see increasing the threshold voltage results in an exponential decrease in subthreshold leakage current. Based on this observation, multi-V_t and gate-length biasing techniques have been proposed to reduce the leakage power dissipation. Without losing generality, we assume the threshold voltage of the NMOS and PMOS transistors are equal. In this case, when the threshold voltage of an inverter is changed to $V_{t,h}$, the new subthreshold leakage power consumption is obtained as,

$$P_{sub,h} = k_{sub}\exp\left(-\lambda\left(V_{t,h} - V_0\right)\right)C_{in} = k_{sub,h}C_{in}\,. \tag{5.16}$$

Utilizing a longer channel length for an inverter increases the threshold voltage of both PMOS and NMOS transistors, which in turn reduces the subthreshold leakage. Based on these observations, we model the subthreshold power dissipation of the i^{th}

inverter in an inverter chain as,

$$P_{sub,l} = k_{sub} l^{-\beta_{sub}} C_{in} \tag{5.17}$$

where β_{sub} is a technology constant. As one can see from Figure 5.5, despite its simplicity, this model is quite accurate.

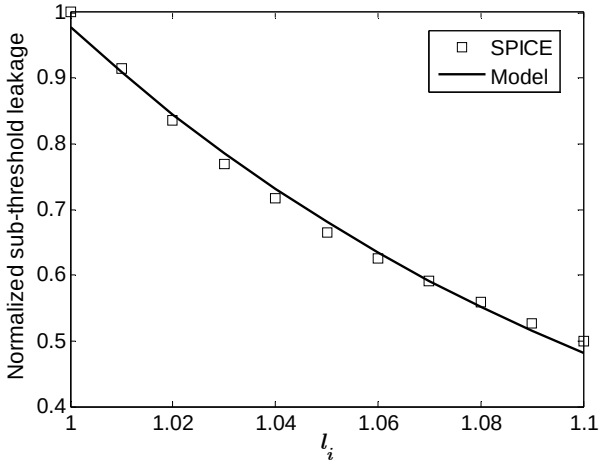


Figure 5.5: Subthreshold power dissipation as a function of channel length.

The tunneling gate leakage current of an NMOS transistor is obtained from Equation (2.2). Ignoring the gate leakage of the PMOS transistor, the tunneling gate leakage power dissipation of an inverter, P_{ox}, can be calculated by,

$$P_{ox} = \frac{A'_{ox}}{1+\beta} p C_{in} = k_{ox} C_{in} \tag{5.18}$$

where $A'_{ox} = A_{ox} V_{dd} (V_{dd} - \psi_s)^2 \exp(-B_{ox} t_{ox} / (V_{dd} \psi_s)) / (t_{ox} \varepsilon_0 \varepsilon_{ox})$ is independent of the size and the threshold voltage of the inverter. From (2.2) one can see that the tunneling gate leakage is proportional to the area of the gate; therefore, in a multi-L_{Gate} technology, (5.18) should be modified as,

$$P_{ox,l} = k_{ox} l C_{in} . \tag{5.19}$$

5.3 Minimum Area Fanout Chain

In minimizing the area of a fanout chain, shown in Figure 5.6, the goal is to find the number of inverters in the chain and their corresponding sizes so that the delay constraint for the sink and the load capacitance constraint for the source are satisfied, while the area of the chain is minimized:

$$\begin{cases} Min & Area \\ s.t. & (i)\ Delay \le T \\ & (ii)\ C_1 \le C_{in,\max} \end{cases} \tag{5.20}$$

where T is the required time at the sink, C_1 is the input capacitance of the first inverter and $C_{in,\max}$ is the maximum tolerable load at the source.

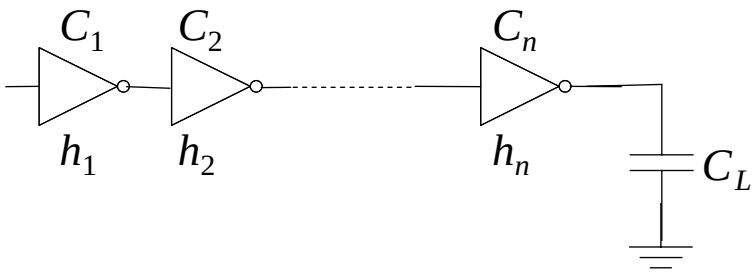


Figure 5.6: A fanout chain driving a lumped capacitance.

In [104], based on the fact that the area of an inverter chain is proportional to the sum of input capacitance of the inverters in the chain and noticing that in an inverter chain with n inverters, the input capacitance of the i^{th} inverter can be expressed as $C_i = C_L / \prod_{j=i}^{n} h_j$, it is shown that the problem of minimizing the area of the chain with n inverters can be formulated in the logical effort notion as,

$$\begin{cases} Min & Area(\vec{h}) = \sum_{i=1}^{n} \dfrac{C_L}{\prod_{j=i}^{n} h_j} \\ s.t. & (i) \quad \sum_{i=1}^{n} p_0 + h_i \le T \\ & (ii) \quad H = \prod_{i=1}^{n} h_i \ge \dfrac{C_L}{C_{in,\max}} \end{cases} \tag{5.21}$$

where C_L is the load capacitance and $\vec{h} = (h_1,...,h_n)$.

Problem stated in (5.21) is called the Fanout Chain Optimization for Area with n inverters, $FCOA(n)$.

The minimized area fanout chain can be found by solving $FCOA(n)$ for different values of n. However, depending on the polarity of the sink, only even or odd values for n should be considered. On the other hand, it can be shown that [104] for a fixed number of inverters in the chain (i.e., a fixed n), (5.21) will have a solution when $n\left(C_L / C_{in,\max}\right)^{1/n} + np_0 \le T$. This inequality defines a lower bound and an upper bound for the values of n satisfying the constraints of (5.21) and limits the number of $FCOA(n)$ instances needed to be solved to find the minimum area fanout chain [104].

Lemma 5.1: In the optimum solution of $FCOA(n)$, the delay of the fanout chain is exactly equal to the required time T, i.e., [104]

$$\sum_{i=1}^{n} p_0 + h_i = T. \tag{5.22}$$

5.3.1 Convex Representation

In the following, we show one important property of $FCOA(n)$ which guarantees the problem of minimizing area of a fanout chain has an optimal polynomial-time

solution. More precisely, we show with a slight modification, the problem shown in (5.21) is converted to a convex program. A convex optimization problem is one of the form [24],

$$\begin{cases} Min & f_0(\vec{x}) \\ s.t. & f_i(\vec{x}) \le b_i, \qquad i = 1,...,m \end{cases} \tag{5.23}$$

where the functions $f_0,...,f_m : \Re^n \to \Re$ are convex, $b_1,...,b_m$ are some positive real numbers, and $\vec{x} = (x_1,...,x_n)$ is a vector. One important property of convex optimization problem is that a local optimal solution is also the global optimum solution.

Lemma 5.2: Function f defined as $f(\vec{x}) = 1/\prod_{i=1}^{m} x_i$ is convex on $dom(f) = \Re_{++}$.

Proof: We use the fact that f is convex if and only if its domain is convex and its Hessian is positive semi-definite [24], i.e., for all x belonging to $dom(f)$, $\nabla^2 f \ge 0$. One can see that,

$$\nabla^2 f(\vec{x}) = \frac{1}{\prod_{i=1}^{m} x_i}\left(\text{diag}\left(1/x_1^2,...,1/x_m^2\right) + zz^T\right) \tag{5.24}$$

where z is a vector such that $z_i = 1/x_i$ and diag(.) is a diagonal matrix. To verify $\nabla^2 f \ge 0$ we should show that for any vector v,

$$v^T \nabla^2 f(\vec{x}) v \ge 0\,. \tag{5.25}$$

However, it can be verified that,

$$v^T \nabla^2 f(\vec{x}) v = \frac{1}{\prod_{i=1}^{m} x_i}\left(\sum_{i=1}^{m} (v_i/x_i)^2 + \left(\sum_{i=1}^{m} v_i/x_i\right)^2\right) \ge 0\,. \tag{5.26}$$

Therefore, f is convex. ■

Theorem 5.1: By changing the second constraint of $FCOA(n)$ as

$$\frac{1}{\prod_{i=1}^{n} h_i} \leq \frac{C_{in,\max}}{C_L} \tag{5.27}$$

$FCOA(n)$ becomes a convex optimization problem for all values of n.

Proof: According to Lemma 5.2 the objective function of $FCOA(n)$ is a summation of convex functions and because the summation operation preserves the convexity property [24], the objective function of the problem given by (5.21) is convex. On the other hand, the first constraint of (5.21) is a linear function of h_i's; hence, it is convex. The function $f(\vec{x}) = \prod_{i=1}^{n} x_i$ is neither convex nor concave [24]. However, according to Lemma 5.2, by re-writing it as (5.27) it becomes convex. Since the objective function and constraints of (5.21) are convex on $\Re_{++}$, the mathematical problem stated in (5.21) is convex. ■

Since $FCOA(n)$ is a convex program, it can be efficiently solved by using standard mathematical program solvers.

5.3.2 Minimum Area versus Minimum Power Fanout Chain

Since both capacitive and leakage power dissipation of a fanout chain are proportional to its area, it has been widely accepted that power minimization of a fanout chain is equivalent to its area optimization [12, 147]. In the following, however, we show that due to short-circuit power dissipation, minimizing area does not necessarily result in a minimized power dissipation solution and the solution

obtained from an area optimization technique may dissipate excessive short-circuit power.

First, note if the constraints of (5.21) do not intersect at any point, i.e., $n\left(C_L / C_{in,\max}\right)^{1/n} + np_0 > T$ there is no solution for the problem. On the other hand, if the intersection of the constraints of (5.21) results in only one point, i.e., when $n\left(C_L / C_{in,\max}\right)^{1/n} + np_0 = T$, the only solution to $FCOA(n)$ is when all h_i's are equal to $T/n - p_0$. In other cases the optimization problem (5.21) can be solved by using the Lagrangian relaxation technique [24, 44]. In this technique, the constraints are relaxed and summed up in the objective function after multiplying them by non-negative coefficients, called the Lagrange multipliers. The new objective function is called the Lagrangian. In $FCOA(n)$, the Lagrangian is written as,

$$L\left(\vec{h},\lambda_1,\lambda_2\right) = Area\left(\vec{h}\right) + \lambda_1\left(\sum_{i=1}^{n} h_i - T_0 + np_0\right) + \lambda_2\left(H_0 - \prod_{i=1}^{n} h_i\right) \tag{5.28}$$

where λ_1 and λ_2 are non-negative Lagrange multipliers, $\vec{h} = (h_1,...,h_n)$, and $H_0 = C_L / C_{in,\min}$.

The set of Kuhn-Tucker conditions implies that at the optimal solution of $FCOA(n)$,

$$\frac{\partial L}{\partial h_i} = 0; \qquad i = 1,...,n \tag{5.29}$$

and

$$\lambda_1 \left(\sum_{i=1}^{n} h_i - T_0 + np_0 \right) = 0 \tag{5.30}$$

$$\lambda_2 \left(H_0 - \prod_{i=1}^{n} h_i \right) = 0 . \tag{5.31}$$

Now, considering the first set of conditions shown in (5.29), from $\partial L / \partial h_1 = 0$, it is concluded that,

$$-\frac{1}{h_1 \pi_1} + \lambda_1 - \frac{\pi_1}{h_1} \lambda_2 = 0 \tag{5.32}$$

where π_i is defined as,

$$\pi_i = \prod_{i}^{n} h_i . \tag{5.33}$$

Similarly, since $\partial L / \partial h_i = \partial L / \partial h_{i+1} = 0$, we have $h_i \partial L / \partial h_i = h_{i+1} \partial L / \partial h_{i+1}$, which results in,

$$\lambda_1 h_i = \lambda_1 h_{i+1} - \frac{1}{\pi_{i+1}} . \tag{5.34}$$

One immediate result of (5.34) is that in the optimal solution of $FCOA(n)$, the values of h_i's are increasing, i.e.,

$$h_1 \leq h_2 \leq ... \leq h_n . \tag{5.35}$$

The equality happens if and only if the required time and input capacitance constraints intersect at exactly one point.

Going back to the remaining Kuhn-Tucker conditions, from Lemma 5.1, one can see (5.30) is already satisfied. The remaining condition, as given in (5.31), implies that one of its terms is zero. If the input capacitance constraint of the optimization problem is "loose", i.e., in the optimal solution $H_0 < \prod_{i=1}^{n} h_i$, it is necessary that $\lambda_2 = 0$. In this case, (5.31) implies that $\lambda_1 = 1/(h_1 \pi_1)$ and (5.32) may be re-written as,

$$\frac{1}{h_1\pi_1}h_i = \frac{1}{h_1\pi_1}h_{i+1} - \frac{1}{\pi_{i+1}}. \tag{5.36}$$

Similarly,

$$\frac{1}{h_1\pi_1}h_{i-1} = \frac{1}{h_1\pi_1}h_i - \frac{1}{\pi_i} \tag{5.37}$$

and since $\pi_i = h_i\pi_{i+1}$, from (5.36) and (5.37), it is concluded that,

$$h_{i+1} = h_i(h_i - h_{i-1} + 1) \tag{5.38}$$

where $h_0 = 0$.

Equation (5.38) is a recursive equation from which the values of all h_i's may be found as functions of h_1. Some of these values are shown in Table 5.1.

Table 5.1: Some terms of recursive Equation (5.38)

i	h_i
1	h_1
2	$h_1^2 + h_1$
3	$h_1^4 + h_1^3 + h_1^2 + h_1$
4	$h_1^8 + 2h_1^7 + 2h_1^6 + 2h_1^5 + 2h_1^4 + h_1^3 + h_1^2 + h_1$

Plugging the values of h_i's as functions of h_1 into (5.22) and solving the polynomial equation, the value of h_1 which minimizes the objective function is found. To the best of our knowledge, there is no closed form solution to (5.38); however, one important property of this recurrence equation may be expressed by the following Lemma.

Lemma 5.3: In the recurrence given by Equation (5.38),

$$h_i > {h_1}^{2^{i-1}}. \tag{5.39}$$

Proof: We first show that all coefficients in polynomial $\Delta_i(h_1) = h_i - h_{i-1}$ are

positive. We do this by using mathematical induction. First we note that $\Delta_1(h_1) = h_1$ is a positive-coefficient polynomial. Next, assuming $\Delta_k(h_1)$ is a positive coefficient for $k \geq 1$ (induction hypothesis), $\Delta_{k+1}(h_1)$ can be written as,

$$\Delta_{k+1}(h_1) = h_{k+1} - h_k = h_k(h_k - h_{k-1} + 1) - h_k = h_k \Delta_k(h_1) \tag{5.40}$$

hence, it is a positive-coefficient polynomial. Now, since for every i, $\Delta_i(h_1)$ is a positive-coefficient polynomial and $h_i = h_{i-1}(\Delta_{i-1}(h_1) + 1)$, it follows that h_i is also a positive-coefficient polynomial with variable h_1; i.e.,

$$h_i = \sum_{j=1}^{ub} a_j h_1^j \tag{5.41}$$

where $a_j \geq 0$. It is easily verified that in Equation (5.41), $ub = 2^{i-1}$ and $a_{ub} = 1$; hence, (5.39) holds. ■

From Lemma 5.3, one can see when the input capacitance constraint of $FCOA(n)$ is loose, in the optimal solution of (5.21) the values of h_i 's grow exponentially and based on (5.11) and Figure 5.2, the ratio of short circuit to capacitive `power dissipation of the inverters grows accordingly. For example, if $T = 23$, $C_L / C_{in,\max} = 90$, $p_0 = 1$, and the polarity of the sink is positive, it can be verified that the optimum values for h_i's in $FCOA(2)$ are 6 and 15, and in $FCOA(4)$ the optimum values are 1, 2, 4, and 12, respectively. From Figure 5.2 one can see that both these scenarios result in excessive short-circuit power dissipation in the last stage of the chain.

5.4 Low-Power Fanout Chains

The discussion in Section 5.3 establishes that minimizing the area of a fanout chain

will not minimize its power consumption. In this section, we generalize the problem and propose a mathematic program for low-power fanout chain design in multi-V_t and multi-L_{Gate} technologies. More precisely, we assume m discrete threshold voltages are available to be used in the inverters of the chain. In addition, we assume the channel length of inverters can be increased up to $L_{\max}$. The objective is to find the optimal number of inverters and their corresponding threshold voltages, channel lengths, and sizes to achieve the minimum power consumption in the active mode. When $m = 1$ and $L_{\max} = L_{nom}$, this problem simply becomes that of finding the optimal number of inverters and their corresponding sizes.

5.4.1 Problem Formulation

A multi-V_t and multi-L_{Gate} fanout chain is shown in Figure 5.7. In this figure, h_i's denote the electrical efforts of the inverters, C_i's are the input capacitances, l_i's denote the channel lengths, and v_i's are the threshold voltages of the inverters. The goal is to find the number of inverters, n, h_i's, l_i's, and v_i's to minimize the total power dissipation while meeting both a timing constraint and an input capacitance upper bound constraint. Moreover, there is an upper bound on the length of the channel and the threshold voltage of each inverter should be selected from a given set of available threshold voltages.

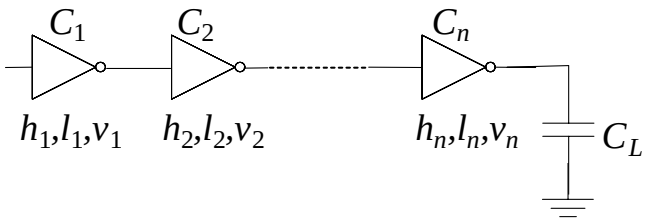


Figure 5.7: A multi-Vt fanout chain.

Since increasing the channel length increases the threshold voltage of a transistor as well, we do not consider increasing both the channel length and threshold voltage of an inverter because the delay penalty tends to be too high. Moreover, we assume a multi-V_t design is achieved by ion implantation in the channel of the gate. Since changing the channel doping has no effect on gate-oxide thickness and negligible effect on the diffusion and gate capacitances, this assumption implies the tunneling gate leakage and capacitive power consumptions are not affected by changing threshold voltages. However, changing the threshold voltage of an inverter alters its delay and subthreshold leakage according to Equations (5.4) and (5.15). On the other hand, as discussed in Section 5.2.2, this change also has an effect on the short-circuit power consumption of the fanout chain. Changing the channel length, on the other hand, alters delay and all components of power dissipation, as described in Section 5.2.

To simplify the equations, without loss of generality, we assume the driver and load of the chain are fixed-sized inverters. The driver is called the 0$^{\text{th}}$ inverter, while the load is called the $(n+1)^{\text{th}}$ inverter.

Using the formulation derived in Section 5.2, the power dissipation of the i^{th} inverter in the chain with the normalized channel length l_i can be expressed as,

$$P_i = \frac{C_L\left(\gamma_i k_{cap} + k_{sub,i} l^{-\beta_{sub}} + k_{ox} l_i + k_{sc,i} h_{i-1} l_i^{-\beta_{sc1}} l_{i-1}^{\beta_{sc2}}\right)}{\prod_{j=i}^{n} h_j} \tag{5.42}$$

where $\gamma_i = (l_i + p_0)/(1+p_0)$. Moreover, $k_{sub,i}$ is obtained from Equation (5.16) and $k_{sc,i}$ is the short-circuit factor for the i^{th} inverter.

Therefore, the problem of optimizing the fanout chain for power dissipation becomes,

$$\begin{cases} Min & P(\vec{h}) = \sum_{i=1}^{n} P_i + k_{sc,n+1} h_n C_L \\ s.t. & (i) \quad \sum_{i=1}^{n} (p_i + g_i h_i) l_i^{\beta_d} \le T \\ & (ii) \quad H = \prod_{i=1}^{n} h_i \ge \frac{1}{l_1} \frac{C_L}{C_{in,\max}} \\ & (iii) \quad 1 \le l_i \le \frac{L_{\max}}{L_{nom}} \\ & (iv) \quad v_i \in \{V_1,...,V_m\} \end{cases} \tag{5.43}$$

where p_i and g_i are the parasitic delay and logical effort of the i^{th} inverter which operates with the threshold voltage of v_i. The first two constraints in (5.43) are the delay and input capacitance constraints while the third constraint of (5.43) imposes that there is an upper bound on the length of the channels. Finally, the forth constraint of (5.43) enforces the threshold voltages of the transistors of the inverters to be from the set of available threshold voltages $\{V_1,...,V_m\}$, where V_1 is the nominal threshold voltage and $V_1 \le ... \le V_m$. The size and threshold voltage of the load are fixed; therefore, the capacitive and leakage power dissipations of the load inverter are constant. However, the short-circuit power dissipation of the load inverter is a function of the electrical effort of the last stage in the chain, i.e., h_n; thus, we include the short-circuit power dissipation of the load into the objective function.

Problem stated in (5.43) which is the Fanout Chain Optimization for minimum Power with n inverters, m threshold voltages, and an upper bound $L_{\max}$ for the channel length will be called $FCOP(n, m, L_{\max})$ in the rest of this chapter. To find

the minimum-power fanout chain, $FCOP(n, m, L_{\max})$ should be solved for different values of n. Based on the polarity of the sink, only even or odd numbers should be considered for n.

Lemma 5.4: In the $FCOP(n, m, L_{\max})$ problem, the total electrical effort, H, is maximized when all v_i's are equal to V_1 and all l_i's are 1, and all h_i's are equal.

Proof: The geometric mean of a number of positive numbers is less than or equal to their arithmetic mean. The equality holds if and only if all values are equal. From the first constraint of (5.43) it can be seen that,

$$\begin{aligned} T &\geq \sum_{i=1}^{n} p_i l_i^{\beta_d} + \sum_{i=1}^{n} g_i h_i l_i^{\beta_d} \\ &\geq \sum_{i=1}^{n} p_i l_i^{\beta_d} + n\prod_{i=1}^{n} \left(g_i h_i l_i^{\beta_d} \right)^{1/n} \end{aligned} \tag{5.44}$$

From (5.44) it is concluded that in order to have a solution to $FCOP(n, m, L_{\max})$, the following relation must hold,

$$\frac{T - \sum_{i=1}^{n} p_i l_i^{\beta_d}}{n\prod_{i=1}^{n} \left(g_i l_i^{\beta_d} \right)^{1/n}} \geq \prod_{i=1}^{n} (h_i)^{1/n} = H^{1/n}. \tag{5.45}$$

Since $p_i \geq p_0$, $l_i \geq 1$ and $g_i \geq 1$, the maximum of H happens when all h_i's are equal, all l_i's are equal to 1, and all p_i's and g_i's assume their minimum values at p_0 and 1, respectively. The latter condition implies that all v_i's are equal. In this case, the maximum value of $H = \prod_{i=1}^{n} h_i$ is $H_{\max} = (T/n - p_0)^n$. ■

According to Lemma 5.4, there is a maximum value for H, $H_{\max}$, for any given buffer count; on the other hand, since $l_1 \leq L_{\max} / L_{nom}$, the second constraint of (5.43) implies that H must be greater than $C_L / C_{in,\min} \times L_{nom} / L_{\max}$. Therefore,

the only feasible buffer counts are those for which $H_{\max}$ is not less than $C_L / C_{in,\min} \times L_{nom} / L_{\max}$.

One important property of $FCOP(n, m, L_{\max})$ is that in its optimal solution, the delay of the fanout chain may not be equal to the specified required time T. To see why this is true, notice the objective function of $FCOP(n, m, L_{\max})$ is not a decreasing function of h_i's or l_i's; therefore, increasing h_i's or l_i's up to the point that $\sum_{i=1}^{n}(p_i + g_i h_i) l_i^{\beta_d} = T$ may not result in the minimum objective function.

If the design is not multi-L_{Gate}, i.e., $L_{\max} = L_{nom}$, then the third constraint in (5.43) will be eliminated from the problem and values of all l_i's become 1. Similarly, if the design is not multi-V_t, i.e., $m = 1$, the fourth constraint in (5.43) is eliminated and the values of all p_i's and g_i's become p_0 and 1, respectively. In this case, one can verify that constraints of $FCOP(n, m, L_{\max})$ are the same as $FCOA(n)$.

If the design is multi-V_t, i.e., $m \geq 2$, due to discrete values of v_i's in $FCOP(n, m, L_{\max})$, a posynomial problem solver needs to enumerate all possible assignments of the threshold voltages, i.e., m^n assignments, and solve the resulting mathematical program to find the minimum-power fanout chain by optimally selecting h_i's and l_i's. Due to its exponential runtime, such an enumeration is not possible. Hence, we use the same approach as in [12] to assign the threshold voltages. In this approach, the assignment of the threshold voltages is done as follows: starting from the source and going to sink, the values of the threshold

voltages are increased. This heuristic called *monotone assignment* of the threshold voltages, greatly simplifies the problem and reduces the number of possible candidates to nm.

It is known that each additional threshold voltage needs one more mask layer in the fabrication process which results in increasing the fabrication cost. As a result, in many cases, only two threshold voltages are utilized in the circuit. At the same time, there are studies that show the benefit of having more than two threshold voltages is small [119]. So, in the sequel we concentrate on the problem of dual-V_t low-power fanout optimization, i.e., $FCOP(n,2,L_{\max})$. The results can be extended to handle more threshold voltages.

The pseudo-code for the $BestChain$ algorithm is provided in Figure 5.8. First, by using the result of Lemma 5.4, for a given $C_{in,\max}$, C_L, and T, the $BestChain$ algorithm finds the lower and upper bounds of n. Based on the polarity of the sink node, only even or odd numbers of inverters between these bounds are considered when searching for the optimum solution. For a given n, the $BestChain$ algorithm attempts to solve the $FCOP(n,2,L_{\max})$ problem with all threshold voltages set to V_1, i.e., the nominal threshold voltage. If there is no feasible solution, then the timing and/or input capacitance constraints are too tight. The algorithm goes through a number of iterations where in each iteration, the threshold voltages of the last m inverters in the chain are set to V_2. This process is repeated until we find $\widetilde{m}$ such that there exists a feasible solution to the $FCOP(n,2,L_{\max})$ with $\widetilde{m}$ inverters, but not with $\widetilde{m}+2$ inverters. In the pseudo-code, function FVT finds the optimum

solution to the $FCOP(n,2,L_{\max})$ problem with known threshold voltage values as captured by the assignment vector, $\vec{v}$. More precisely, FVT algorithm finds l_i's of the first $n-m$ inverters, which have the nominal threshold voltage, and also h_i's of all inverters. Note since the FVT function is called for fixed $\vec{v}$'s; this optimization problem is the minimization of a posynomial function with posynomial inequality constraints. This posynomial formulation is translated into a convex one by a change of variables $h_i = \exp(x_i)$ and $l_i = \exp(y_i)$ and is solved in polynomial time [24].

```
BestChain(C_in,max, C_L, T, pol){
  (ñ1, ñ2) = solution (C_L / C_in,min · L_nom / L_max) = (T / n − p0)^n;
  n1 = ⌊ñ1⌋ or ⌊ñ1⌋ + 1(depending on pol);
  n2 = ⌊ñ2⌋;
  (pwr*, h*, l*, v*) = (+∞, ∅, ∅, ∅);
  For n = n1 to n2 step 2{
   For i = 1 to n
     v(i) = V2;
   (h, l, pwr) = FVT(n, T, C_in,max, C_L, v);
   If h = ∅
     continue;
   If pwr < pwr*
     (pwr*, h*, l*, v*) = (pwr, h, l, v);
   For m = n to 1 step -1{
    v(m) = V2;
    (h, l, pwr) = FVT(n, T, C_in,max, C_L, v);
    If pwr > pwr*
      (pwr*, h*, l*, v*) = (pwr, h, l*, v);
   }
  }
  Return (pwr*, h*, l*, v*);
}
```

Figure 5.8: BestChain algorithm.

5.5 Building a Fanout Tree

In this section we show how to build a fanout tree with more than one sink. Reference [75] introduced two transformations that could be performed on a fanout tree, namely merging and splitting, and showed these transformations preserve area, delay, and input capacitance of the fanout tree. We have extended the merging and splitting transformations to handle multi-V_t and multi-L_{Gate} fanout trees, as depicted in Figure 5.9.

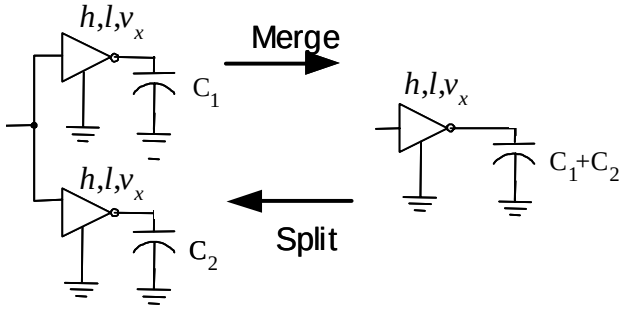


Figure 5.9: Extended split/merge transformations for multi threshold voltage and multi channel length inverters.

Theorem 5.2: The extended split/merge transformations applied to a multi-V_t and multi-L_{Gate} fanout tree as depicted in Figure 5.9 preserve the delay, input capacitance, and power dissipation values of the tree.

Proof: We provide the proof for the split transformation. Before splitting, the delay of the inverter is $(p_x + g_x h) l^{\beta_d}$ while the input capacitance is $(C_1 + C_2)/h$. After splitting the original inverter into two inverters with equal electrical efforts of h and equal channel length l and threshold voltages of v_x, the delay through the inverter in either branch will be $(p_x + g_x h) l^{\beta_d}$ while the input capacitances will be C_1/h and C_2/h which sum up to $(C_1 + C_2)/h$. Therefore, this transformation preserves

the delay and input capacitance values. Since this transformation does not change the input capacitance, the electrical effort of the previous stage, which characterizes the short-circuit power dissipation of two inverters before the merge transformation, does not change; it is easy to see the capacitive and leakage power consumption of the tree remains the same after the transformation. Moreover, since this transformation does not change the channel length of the inverter transistors, the short circuit power dissipations of C_1 and C_2 remain the same. Hence, the total power dissipation of the fanout tree before and after the split transformation remains the same. ■

Since extended split/merge transformations preserve the delay, input capacitance, and power dissipation values, by using these transformations, any fanout optimization problem with m sink nodes, can be converted to m fanout chain optimization problems, whose respective power dissipations will be the same.

To apply these transformations, two issues should be addressed. The first issue is the input capacitance allocation to different chains in a decomposed fanout tree and the second issue is the validity of a continuous-size inverter library. In the following we address these questions.

5.5.1 Input Capacitance Allocation

The Input Capacitance Allocation to achieve minimum Power (ICAP) problem is defined as follows: Given a number of sinks, each with a required time, polarity, and capacitive load, and a total budget on input capacitance $C_{in,tot}$, allocate portions of $C_{in,tot}$ to each fanout chains such that the total power is minimized while the given

constraints for all sinks are satisfied. In this section we show the ICAP problem is NP-complete and we use a heuristic to allocate the input capacitance to different chains in a decomposed fanout tree.

Lemma 5.5: For a fixed number of inverters in a multi-V_t and multi-L_{Gate} fanout chain, the power cost is a decreasing function of the input capacitance bound, $C_{in,\max}$.

Proof: From the second constraint in (5.43), it is seen that increasing the input capacitance constraint of a fanout chain expands the feasible space of the optimization problem. Therefore, there exists either a better solution with lower power consumption or one with the same power consumption; that is, the power cost in a fanout chain is a decreasing function of the input capacitance bound. ■

Theorem 5.3: The ICAP problem is NP-Complete.

Proof: To prove that ICAP is NP-Complete, we show the 0-1 Knapsack problem may be reduced to the ICAP problem. In the 0-1 Knapsack problem, there are some items, each with its own value and weight; the objective is to select some items such that the total value of the selected items is maximized while their total weight is not more than a given budget. In the ICAP problem, however, the objective is to minimize power. To make ICAP a maximization problem, we consider the negative of power as the objective function. According to Lemma 5.5, the power cost is a decreasing function of the input capacitance constraint; therefore, the graph of the maximum of negative power over all inverter counts looks like Figure 5.10. Notice this graph exhibits a piecewise behavior because power is represented by different functions for different inverter counts. The piecewise nature of power versus input

capacitance helps us to reduce the 0-1 Knapsack problem to the ICAP problem.

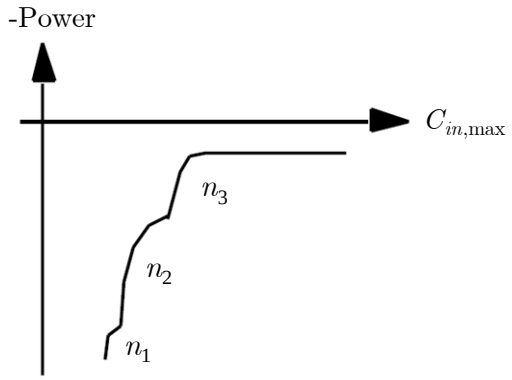


Figure 5.10: Negative of power dissipation versus the input capacitance curve.

This reduction is similar to the reduction of the Knapsack problem to the problem of input capacitance allocation for minimum area, hence, it is omitted here. Interested readers may refer to [104] for details. After proving the ICAP is NP-Hard, we show the decision version of the ICAP can be tested in polynomial time. This is clearly true because one can add up the input capacitances of each branch and compare it with the input capacitance budget in linear time. Therefore, the ICAP is in NP; since it was shown that the ICAP is NP-Hard, therefore, the ICAP problem is NP-Complete. ■

The heuristic we use for solving the ICAP problem is similar to that of [104] and starts by allocating the minimum input capacitance required for each branch to have a feasible fanout chain solution. Next, the remaining total input capacitance is divided between the chains in proportion to the positive slopes of $H_{\max,i}$ versus n_i for each branch i.

5.5.2 Discrete-Size Inverter Library

The second issue to address is the assumption of the availability of a continuous-size

inverter library. In reality, in the ASIC libraries, although many different inverter sizes are available, these sizes are discrete (there are typically 8-16 different inverter sizes in an industrial state-of-the-art ASIC library.) So the solution needs to be mapped onto one of the available inverters in the library. The main problem when rounding the inverter sizes is that it may result in significant errors. To address this problem, reference [104] defined a constant ε_h and merged two inverters on different chains if the difference between their electrical efforts was less than or equal to ε_h. Notice, in general, two inverters are merged if the rounding error after merging is smaller than the sum of the rounding errors of inverters before the merge operation. We adopt the same heuristic with the additional requirement that, in the multi-V_t tree, the two candidate inverters should also have the same threshold voltage, whereas and in the multi-L_{Gate} inverters, the difference between l_1 and l_2 should be smaller than a constant ε_l. Merging is performed starting at the source of the signal and proceeds toward sinks.

Table 5.2: Technology parameters used in simulations

Parameter	Value	Parameter	Value
$L_{\max} / L_{nom}$	1.1	k_{ox}	0.096
$V_{t,low}$	0.2V	$k_{sc,LL}$	0.069
$V_{t,high}$	0.3V	$k_{sc,LH}$	0.006
β	3.5	$k_{sc,HL}$	0.099
τ_0	8.6e-12	$k_{sc,HH}$	0.014
p_0	1.33	β_{sub}	7.4
k_{cap}	1.000	β_{sc1}	22.5
$k_{sub,low}$	0.343	β_{sc2}	4.4
$k_{sub,high}$	0.078	β_d	1.6

5.6 Simulation Results

The proposed technique in Section IV, which we call *LPFO*, has been developed in the SIS framework [112]. The MOSEK convex optimization tool [87] has been used to solve the mathematical problems. To extract the parameters used in the optimization problems, we performed transistor level simulation of devices in HSPICE [58] on a 65nm technology node [99]. The simulations have been done at the frequency of 1GHz, supply voltage of 1.1V, and die temperature of 100^{o}C. Moreover, we assumed the switching activity of the source node is 5% and the probability of this node being at logic one is 0.5 in all circuits. The parameters of this technology node are shown in Table 5.2. In this table, $k_{sc,LH}$ is the short-circuit factor of an inverter whose threshold voltage is high while the threshold voltage of its driver is low. $k_{sc,LL}$, $k_{sc,HL}$, and $k_{sc,HH}$ are defined similarly. The values of short circuit factors as well as $k_{sub,low}$, $k_{sub,high}$, and k_{ox} are normalized with respect to k_{cap}. In this set of experiments, a standard cell library consisting of sixteen different inverters was used to map the fanout trees.

To study the efficiency of our technique in reducing the power consumption of the fanout trees, we conducted two sets of experiments. In the first set of experiments we assumed the options of multi-V_t and multi-L_{Gate} are not available in the library and compared the results of *LPFO* with the results of low-area fanout optimization (*LEOPARD*) [104] for a few random problems in the form of fanout chains whose specifications are shown in Table 5.3. In this table $C_{in.\max}$ denotes the maximum allowed capacitance at the input of the fanout chain, C_{out} is the load capacitance,

and pol is the polarity of the sink. In each fanout chain, first the path delay was minimized using the technique proposed in [124] and delay and power consumption of each chain was measured by SPICE simulations. Next, each chain was given some additional slack and either *LPFO* or *LEOPARD* algorithm was invoked to minimize the power dissipation or the area of the fanout chain. Each optimized chain was mapped to a library of inverters, and detailed SPICE simulation was carried out on the circuit to measure the power consumption. The results of these simulations are shown in Table 5.4. From this table, one can see minimizing the area of the fanout chains in many cases increases the total power consumption. On the other hand, when the fanout chains are optimized for power, by increasing the available slack in the chain, the power reduction saturates at some point. From the table, the power consumption of the minimum power fanout chains is not always a decreasing function of available slack. This is due to round-off error in mapping the continuous-size inverters to discrete-size inverters in the library.

Table 5.3: Specification of fanout chain problems

Circuit	**Circuit Specification**			**Min Delay Circuit**	
	$C_{in,\max}$	C_{out}	pol	**Power (µW)**	**Delay (ps)**
FC1	1	64	+	20.9	140.0
FC2	1	100	+	14.3	129.8
FC3	15	200	–	22.5	65.2
FC4	20	100	+	23.9	61.2
FC5	20	50	–	9.4	69.0
FC6	2	100	–	48.4	94.6
FC7	30	80	+	7.5	36.9
FC8	8	50	–	7.5	115.2
FC9	10	150	–	19.1	42.2
FC10	50	200	+	7.6	52.3

Table 5.4: Comparison of total power consumption in minimum delay fanout chains, LEOPARD, and LPFO

Circuit	Power Reduction over Minimum Delay Circuit (%)							
	LEOPARD				LPFO			
	Slack 10%	Slack 20%	Slack 30%	Slack 40%	Slack 10%	Slack 20%	Slack 30%	Slack 40%
FC1	5.94	-31.51	-55.90	-55.90	10.30	10.17	7.10	7.10
FC2	-2.54	-12.85	-41.79	-72.30	3.81	4.52	2.57	2.59
FC3	15.04	14.72	12.69	-27.62	15.92	17.84	18.32	18.05
FC4	13.13	16.44	16.68	15.95	13.25	17.20	18.04	18.62
FC5	5.02	7.32	7.98	7.70	5.02	7.32	7.98	7.70
FC6	-7.23	-20.06	-35.59	-47.64	0.00	0.00	0.00	0.00
FC7	21.11	28.50	33.49	35.14	21.61	28.77	33.58	36.14
FC8	-7.06	-17.61	-33.83	-33.83	0.00	0.00	0.00	0.00
FC9	13.48	12.17	9.46	5.00	14.63	15.95	15.94	15.18
FC10	17.16	24.20	28.37	29.84	18.52	25.65	29.75	31.33
Average	**7.40**	**2.13**	**-5.84**	**-14.37**	**10.31**	**12.74**	**13.33**	**13.67**

The second set of experimental results compares *LPFO* with *LEOPARD* and the SIS fanout optimization program for a set of problems in the form of fanout trees. SIS runs different fanout optimization algorithms, namely *Two-Level*, *Bottom-Up*, *Balanced*, *LT-Tree*, and reports the best one [131]. In this set of experiments, the same standard cell library used for *LPFO* and *LEOPARD* has been utilized as the SIS library. For each inverter $\tau_{intrinsic}$ and R_{out} were specified for the SIS library delay model and p_0 and τ_0 were specified for the logical effort delay model. A very close match between the SIS delay and logical effort delay model values was enforced.

The fanout optimization programs of SIS were first used to perform fanout optimization for a set of problems. Next the delay and input capacitance resulting from SIS were used as constraints for *LPFO* and *LEOPARD*. After performing the fanout optimization, the SPICE netlist for each circuit was generated and detailed HSPICE simulation was performed to measure the delay and the power consumption of the circuit. The results of these experiments are reported in Table 5.5. The first

column is the name of the problem instance, the second column denotes the number of sinks in the fanout problem, columns 3 and 4 respectively show the area and power consumption of each fanout problem achieved by running the SIS fanout optimization and the remaining columns show the area and power reduction of *LEOPARD* and *LPFO* algorithms over corresponding values of SIS program. From Table IV one can see fanout trees resulting from *LEOPARD*, on average, consume 11.79% more power than those achieved by SIS. Utilizing *LPFO*, on the other hand, reduces not only the power consumption of fanout trees by an average of 11.17% but also their area by an average of 29.64%.

The runtime of our algorithm for the largest problem with 30 sinks is about 5 seconds when the options of multi-V_t and multi-L_{Gate} are not available, 7 seconds when only the multi-L_{Gate} option is available, 21 seconds when only the multi-V_t option is available, and 24 seconds when both multi-V_t and multi-L_{Gate} options are available.

Note in our problem setup and in the simulation results, we ignored the interconnect power dissipation and delay costs. The reason is that we do the fanout optimization during logic synthesis and prior to generating layout. Therefore, locations of the source and the sinks are not known. As a result the interconnect delay information cannot be accurately modeled. It is thus reasonable to assume the expected values of delay and power dissipation per wire in the inverter chain or the fanout tree are nearly the same. This constant contribution can, thus, be taken out of the problem formulation by properly adjusting the required time constraints on the sinks and adding a constant term to the total power equation.

Table 5.5: Comparison of SIS, LEOPARD, and LFPO fanout optimization algorithms

Circuit	Sink	SIS		LEOPARD		LPFO	
		Area	Power (μW)	Area Reduction over SIS	Power Reduction over SIS	Area Reduction over SIS	Power Reduction over SIS
FT1	5	304	14.4	47.70	11.81	43.09	16.67
FT2	7	1082	119.0	62.38	-16.81	9.89	6.72
FT3	8	1026	63.3	48.34	-18.17	42.01	12.48
FT4	10	1139	68.3	79.54	-16.40	53.99	13.47
FT5	20	1347	105.0	54.94	-28.57	18.63	2.76
FT6	12	928	64.4	45.37	-8.07	26.51	12.73
FT7	14	1490	109.1	67.92	-22.82	45.97	17.60
FT8	14	838	86.3	34.01	-9.04	-7.28	9.15
FT9	25	2853	150.0	78.48	-18.00	56.78	15.33
FT10	30	2496	160.0	60.10	-15.63	27.92	6.88
FT11	10	715	46.7	52.73	-0.86	30.91	13.49
FT12	12	1465	73.4	59.73	3.00	50.17	13.62
FT13	15	1218	92.8	38.83	-11.31	16.67	13.15
FT14	16	1099	94.1	38.31	-7.76	8.64	8.29
FT15	22	1334	115.0	48.20	-18.26	20.69	5.22
Average				**54.44**	**-11.79**	**29.64**	**11.17**

5.7 Summary

In this chapter we showed the fanout optimization with area and power objective functions are not the same and a fanout tree optimized for area may dissipate excessive short-circuit power. By modeling all components of power dissipation, i.e., capacitive, short-circuit, subthreshold leakage and tunneling gate leakage, we formulated the fanout optimization problem as a geometric program for a circuit with one sink and showed how to build a fanout tree from power optimized fanout chains. To reduce the leakage power consumption, we proposed using multi-V_t and multi-L_{Gate} inverters in the fanout trees. Experimental results show the proposed technique is effective in reducing the total power consumption of fanout trees.

Chapter 6
Power Optimal MTCMOS Repeater Insertion

6.1 Introduction

As the CMOS technology continues to scale down toward UDSM technologies, more functionality is being integrated on a single die. This drastic integration results in increase in the size of the die, and consequently in the number of long global interconnects and in their length. The interconnect delay becomes the dominant factor to determine the overall performance of the integrated circuits. Since the delay of an interconnect is quadratic in its length, repeater insertion has been widely used to reduce the delay. As shown in [18] the repeaters can be optimally sized and separated to minimize the interconnect delay. The size of an optimal repeater is typically much larger than a minimum-sized repeater. Since millions of repeaters will be inserted to drive global interconnects, significant power will be consumed by these repeaters, particularly if delay-optimal repeaters are used [30]. Several works used the extra tolerable delay for power saving in interconnects. Authors in [19] and [30] provided analytical methods to compute unit length power optimal repeater sizes and distances. The power analysis should consider capacitive, leakage, and short circuit accurately. As the technology scales down, wires are laid out closer to each other which in turn increases the capacitive coupling noise on the

interconnection lines. This will affect both delay and power consumption in interconnects. In addition to switching power on the coupling capacitances, the authors of [42] showed that the short circuit power consumption is increased significantly in the presence of crosstalk noise. Therefore, one should also consider this effect in the design of power optimal repeaters. Moreover, the technology scaling has resulted in large increase in leakage current. Leakage power has grown exponentially to become a significant fraction of the total chip power consumption [65]. Authors in [103] studied the applicability of MTCMOS to repeater design for leakage power saving, however they did not provide a mathematical solution for the simultaneous optimal sizing of the sleep transistors and repeaters and the insertion length. In addition the effect of crosstalk on delay and power has not been taken into account for the optimal design.

This chapter studies the opportunity of minimizing the average power consumption during both active and standby mode of the bus lines by simultaneously computing repeater sizes, repeater insertion lengths, and the size of the sleep transistors subject to a delay constraint in the presence of crosstalk noise. We consider the worst case crosstalk for the delay constraint. However the assumption of worst case crosstalk is not realistic for power optimization. More precisely, the objective is to minimize the average power (in contrast to minimizing the maximum power). Therefore, we show how to estimate the average power as a function of probability of different types of transitions on the coupled lines. We will also discuss the delivery circuitry of the sleep signals to the sleep transistors.

The remainder of this chapter is organized as follows. Section 6.2 presents the

delay and power models in the presence of crosstalk. Power optimization of bus lines by utilizing sleep transistors is presented in section 6.3. Experimental results are given in Section 6.4 and Section 6.5 summarizes the chapter.

6.2 Preliminaries

This section describes our delay and power model. We also explain the delay-optimal buffer size and insertion length in the presence of crosstalk noise.

6.2.1 Delay Model

Consider a uniform interconnection line of resistance r per unit length and capacitance c per unit length, and total length of L. Suppose the line is divided into L/l segments and identical repeaters of unit driving resistance r_s, unit input capacitance c_g, unit output capacitance c_p, and size s are inserted at each segment (c.f. Figure 6.1 for a pictorial). Figure 6.2 shows one stage of the repeater chain with the interconnect model in between. The delay and the transition time of a segment comprising of a repeater driving an interconnect segment of length l terminated with a repeater of the same size and driven by a step input are $\ln 2 \cdot \tau$ and $\ln 9 \cdot \tau / 0.8$, respectively. Note that $\tau = r_s(c_g + c_p) + r_s cl/s + rlsc_g + \frac{1}{2}rcl^2$. With a finite input slew rate, the contribution of the input transition time t_r to the repeater delay can be represented by γt_r [107] where, for a rising input, γ is calculated as: $\gamma = \frac{1}{2} - (1 - V_{tn}/V_{dd})/(1 + \alpha_n)$ where V_{tn} is the threshold of the NMOS and α_n is the NMOS alpha-power parameter. Similarly, for a falling transition, γ is calculated from the PMOS parameters. An average value for γ is used. Therefore the delay of

one repeater stage is given by $\left(\ln 2 + \gamma \cdot \ln 9/0.8\right)\tau$.

Figure 6.3 shows the delay model for two adjacent bus lines. c_c is the coupling capacitance per unit size. We assume zero skew between the transitions launched into the lines. The worst case delay occurs when transitions on these two lines are in opposite directions.

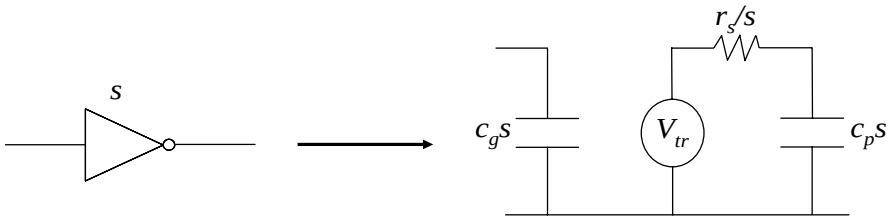


Figure 6.1: Buffer model.

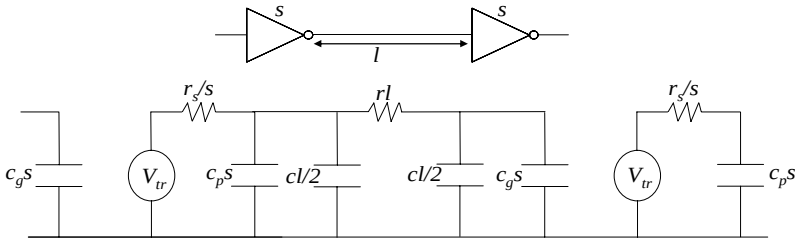


Figure 6.2: One stage of repeaters with interconnect model.

We model the Miller effect in coupling capacitance (to create the worst case delay conditions) by rewriting the formula for the time constant τ as follows:

$$\tau = r_s\left(c_g + c_p\right) + \frac{r_s}{s}(c + 2c_c)l + rlsc_g + \left(\tfrac{1}{2}c + c_c\right)rl^2 \qquad (6.1)$$

The total delay of the interconnection line is equal to $\tau \cdot (L/l)$. Therefore, minimizing the total delay is equivalent to minimizing the time constant per unit length i.e., τ/l:

$$\frac{\tau}{l} = \frac{1}{l}r_s\left(c_g + c_p\right) + \frac{r_s}{s}(c + 2c_c) + rsc_g + \left(\tfrac{1}{2}c + c_c\right)rl \qquad (6.2)$$

With a derivation similar to that given in [18], the worst case delay per unit length

of interconnect line (in the presence of crosstalk) is minimized when:

$$l_{opt} = \sqrt{\frac{2r_s(c_g + c_p)}{r(c + 2c_c)}},\ s_{opt} = \sqrt{\frac{r_s(c + 2c_c)}{rc_g}} \tag{6.3}$$

and,

$$\left(\frac{\tau}{l}\right)_{opt} = 2\sqrt{r_s c_g r(c + c_c)}\left(1 + \sqrt{\frac{1}{2}\left(1 + \frac{c_p}{c_g}\right)}\right) \tag{6.4}$$

It has been shown in [19] and [30] that the optimal delay per unit length (and therefore the optimal total delay) is insensitive to both the size of the repeaters and the distance between repeaters. Hence, significant power and area can be saved by allowing a small delay penalty. Therefore, one can use repeaters with sizes smaller than s_{opt} and segment lengths longer than l_{opt}, and achieve a significant power saving. To accurately address this power optimization problem, we first present the power dissipation model of the global buses and then introduce our power optimal repeater design methodology.

6.2.2 Power Dissipation Model

The power dissipation of a global bus line has three components: capacitive power, short circuit power, and leakage power.

6.2.2.1 Capacitive Power Dissipation

The capacitive power for one stage of the bus can be calculated as:

$$P_{cap} = \alpha f V_{dd}^2 (s(c_g + c_p) + lc) \tag{6.5}$$

where α is the switching activity of the inverter, f is the frequency, and V_{dd} is the supply voltage. Note that equation (6.5) does not consider the capacitive power

consumed on the coupling capacitances. When only one of the lines switches, the coupling capacitance $c_c \cdot l$ charges or discharges with a voltage level change of V_{dd}. Therefore, its coupling energy consumption is $0.5 c_c l V_{dd}^2$. When two adjacent bus lines are simultaneously switching in the opposite directions, the coupling capacitance ($c_c \cdot l$) charges or discharges with a voltage level change of $2V_{dd}$. Therefore, the total energy consumption by the drivers of both lines is $0.5 c_c l (2V_{dd})^2$ [117]. Finally when two adjacent bus lines make transitions in the same direction, no coupling energy is consumed. To estimate the average capacitive power consumption on a single stage of the repeater chain, we make the following assumptions: i) Assume that there is no temporal and spatial correlation between the data which is being transmitted through the two adjacent bus lines. ii) The probability of transmitting a '1' is equal to p. As a result, the probability of the transition between two consecutive data bits on a single bus line can be calculated as $k_1 = p(1-p)$. To calculate the average coupling power, we need to calculate the probability of each type of transition on the coupling capacitance between two adjacent lines. Table 6.1 presents these probabilities for all possible scenarios. Note that $\sum_{i=2}^{5} k_i = 1$. Using the values of k_1 to k_5, we can write the average capacitive power consumption for one stage of two adjacent bus lines (Figure 6.3):

$$\begin{aligned} P_{cap} &= 2 \times 0.5 k_1 f V_{dd}^2 \left(s\left(c_g + c_p \right) + lc \right) \\ &\quad + 0.5 k_2 f (2V_{dd})^2 lc_c + 0.5 k_3 f {V_{dd}}^2 lc_c \end{aligned} \tag{6.6}$$

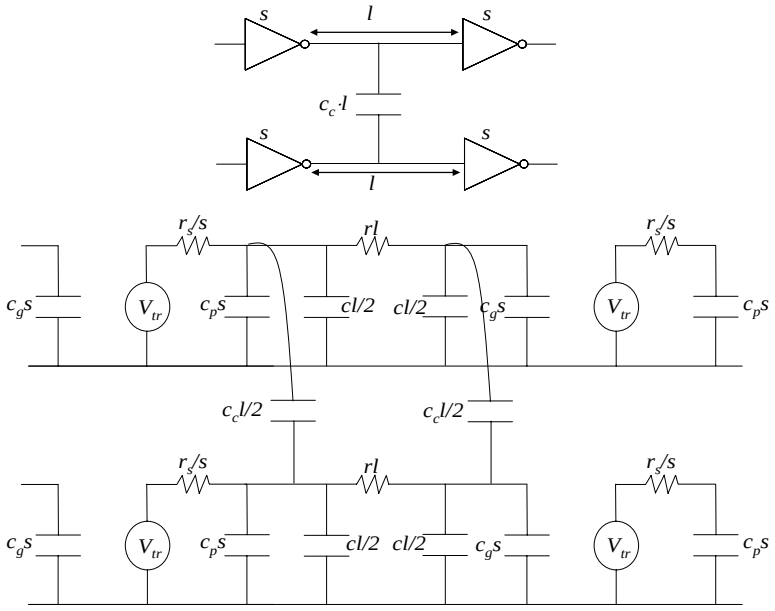


Figure 6.3: The model for one stage of two adjacent coupled bus lines.

Without loss of generality and for the sake of the presentation, we will limit ourselves to only two adjacent lines. The analysis for three (and more) bus lines is similar.

In general, if the input pattern and the spatial-temporal correlation between the data bits of a single line or two adjacent lines are available, a number of probabilistic techniques such as [82, 83, 142] can be used to estimate k_1 to k_5. Furthermore, several encoding techniques have been proposed for minimizing coupling effect for static on chip bus structures [20, 102, 115]. Some approaches were also introduced to find a permutation for the bus lines for minimizing the crosstalk effects [66, 148]. The impact of these optimization techniques can be captured by appropriately revising the equations for k_1 to k_5. The rest of the analysis remains the same.

Table 6.1: Probability of different switching scenarios on the coupling capacitances

Transition Type	Occurrence Probability
Opposite direction	$P(\uparrow\downarrow) = k_2 = 2p^2(1-p)^2$
One switches and other is quiet	$P(\uparrow -) = k_3 = 4p(1-p)\left(p^2 + (1-p)^2\right)$
Both quiet	$P(--) = k_4 = p^4 + (1-p)^4 + 2p^2(1-p)^2$
Same direction	$P(\uparrow\uparrow) = k_5 = 2p^2(1-p)^2$

The coupling power is also dependent on the relative switching time of the line drivers [117]. For global buses, we can safely assume zero skew between the drivers' switching times. However, one can consider the relative delay between the transitions of the two lines and use a similar approach as [117] to compute the effect of relative delay on coupling power.

6.2.2.2 Short-Circuit Power Dissipation

Most of the previous works on power optimal repeater design either ignore the short-circuit power consumption or use an inaccurate approximation of the short-circuit power consumption. In Chapter 5 we developed a simple model to short circuit power dissipation during a gate-level optimization process. That model, however, did not account for the effect of interconnect and crosstalk noise. Therefore, in this chapter we use the closed form formula presented in [92] which captures the dependence of the short-circuit power consumption on the circuit parameters.

The short-circuit power consumption is increased significantly in the presence of crosstalk noise [42]. Therefore similar to capacitive power, we formulate the average short circuit power consumption based on the transition type probability on adjacent

bus lines (Table 6.1). As shown in [92], the short-circuit energy consumption of an inverter during a full signal switch (such as a falling transition followed by a rising) can be approximated as

$$E_{SC} = \frac{4s^2 I_{d0}^{\ 2} t_r^{\ 2} V_{dd}}{V_{dsat} G C_{out} + 2s \cdot H I_{d0} t_r} \tag{6.7}$$

where H and G are technology dependent parameters and I_{d0} is the average saturated drain current of the NMOS and PMOS transistors of the minimum sized inverter. Due to the shielding effect of the interconnect resistance, the repeater sees a capacitance less than C_{total}, where C_{total} is the summation of repeater parasitic capacitances, interconnect capacitance and the coupling capacitances (considering the miller effect based on the transition type), e.g.,

$$C_{total}(\uparrow\downarrow) = (c_p + c_g)s + (c + 2c_c)l \tag{6.8}$$

Using the effective capacitance approach, the capacitance seen by the repeater for opposite direction transitions is written as:

$$C_{out}(\uparrow\downarrow) = C_{eff}(\uparrow\downarrow) = \left(c_p s + {cl}/{2} + c_c l\right) + \delta \cdot \left(c_g s + {cl}/{2} + c_c l\right) \tag{6.9}$$

where $\delta < 1$ and depends on l and s. The ratio of C_{eff} to C_{total} is also a function of l and s. Similar to [30], we calculate ω, the average ratio of C_{eff} to C_{total} for different types of transitions. This average ratio is used for short circuit evaluation. In addition, due to the impact of crosstalk on transition time, different values for t_r are used (by considering different τ values due to different coupling capacitances). Therefore, the average short circuit power consumption of the repeater (for one falling or rising transition) can be estimated as:

$$P_{SC} = k_2 \cdot \frac{2fs^2 {I_{d0}}^2 \cdot {t_{r(\uparrow\downarrow)}}^2 \cdot V_{dd}}{V_{dsat} G \cdot \omega_{(\uparrow\downarrow)} C_{total_{(\uparrow\downarrow)}} + 2s \cdot HI_{d0} t_{r(\uparrow\downarrow)}} + \frac{k_3}{2} \cdot \frac{2fs^2 {I_{d0}}^2 \cdot {t_{r(\uparrow-)}}^2 \cdot V_{dd}}{V_{dsat} G \cdot \omega_{(\uparrow-)} C_{total_{(\uparrow-)}} + 2s \cdot HI_{d0} t_{r(\uparrow-)}} + k_5 \cdot \frac{2fs^2 {I_{d0}}^2 \cdot {t_{r(\uparrow\uparrow)}}^2 \cdot V_{dd}}{V_{dsat} G \cdot \omega_{(\uparrow\uparrow)} C_{total_{(\uparrow\uparrow)}} + 2s \cdot HI_{d0} t_{r(\uparrow\uparrow)}} \quad (6.10)$$

6.2.2.3 Leakage Power Dissipation

From (2.1) one can see that the subthreshold leakage of an NMOS transistor is obtained as,

$$P_{sub,nmos} = A"_{sub} W_{nmos} \mu_{nmos} e^{-\lambda V_{tn}} \quad (6.11)$$

where $\lambda = q / n'kT$ and $A"_{sub} = A_{sub} V_{dd} C_{ox} / L_{eff} \exp(\lambda \eta V_{dd})$ are technology constants. A similar formula can be derived for a PMOS transistor. Therefore, subthreshold leakage power dissipation of a repeater can be written as,

$$P_{sub} = p \cdot P_{sub,P} + (1-p) \cdot P_{sub,P} \quad (6.12)$$

where p is the probability that the input of the inverter is at logic 1. If the ratio of the width of the PMOS transistor to that of the NMOS transistor is β, equation (5.14) can be re-written as:

$$P_{sub} = \frac{A"_{sub} \cdot s W_{\min}}{1+\beta} \left(p\beta \mu_{pmos} e^{-\lambda V_{tp}} + (1-p) \mu_{nmos} e^{-\lambda V_{tn}} \right) = K_{sub} \cdot s \quad (6.13)$$

where $W_{\min}$ is the minimum size of the inverter. Similarly, from (2.2), the tunneling gate leakage of a repeater with size s can be modeled as,

$$P_{ox} = \frac{A"_{ox}}{1+\beta} p \cdot s \cdot W_{\min} = K_{ox} \cdot s \quad (6.14)$$

where $A"_{ox} = A_{ox} L_{eff} V_{dd} (V_{dd} - \psi_s)^2 / {t_{ox}}^2 \exp(-B_{ox} t_{ox} / (V_{dd} - \psi_s))$ is a coefficient

independent of the size and threshold voltage of the inverter.

6.2.2.4 Average Power Dissipation

Having obtained the equations for different components of the power dissipation in equations (6.6), (6.10), (5.15) and (6.14), the total average power dissipation for one stage of two adjacent bus lines in the active mode of circuit operation can be written as,

$$P_{active} = P_{cap} + 2P_{sc} + 2P_{sub} + 2P_{ox} \tag{6.15}$$

The factor 2 is due to the presence of two repeaters in one stage of two adjacent lines. Note that we have already considered the two repeaters on adjacent lines in the case of P_{cap} in equation (6.6). In the standby mode, however, the only sources of the power dissipation are the subthreshold and tunneling gate leakage; so,

$$P_{standby} = 2P_{sub} + 2P_{ox} \tag{6.16}$$

The average power consumption can be obtained as a weighted sum of the power consumption in the active and standby modes:

$$P_{total} = \chi P_{active} + (1 - \chi)P_{standby} \tag{6.17}$$

where χ is the *active mode factor* of the circuit, i.e., the percentage of the time the circuit is in the active mode.

6.3 Power Optimization for MTCMOS Design

6.3.1 Power and Delay Modeling

MTCMOS technology provides low leakage and high performance operation by utilizing high speed, low V_t transistors for logic cells and low leakage, high V_t devices as sleep transistors [1]. Sleep transistors disconnect logic cells from the

supply and/or ground to reduce the leakage in the standby mode. The bus lines spend large percentage of the time in the standby mode. Therefore sleep transistors can be used for total power saving. The drawback is the increase in the delay in the active mode due to the additional resistance of the sleep transistors. Since repeaters are inserted at identical distances, we can share the sleep transistors between repeaters on different data lines. Figure 6.4 shows the case for only two adjacent bus lines. Similarly we can share the sleep for more than two bus lines.

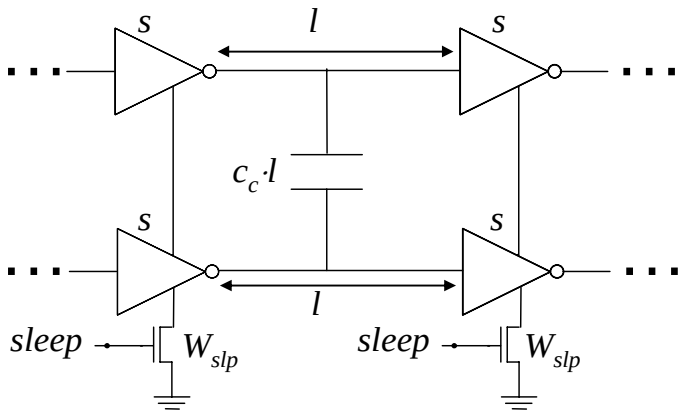


Figure 6.4: Sharing of sleep transistors among different bus lines.

In the presence of sleep transistor both leakage components are substantially smaller in the standby mode. In the standby mode the virtual ground node (i.e., the drain terminal of sleep transistor) charges to a voltage near V_{dd} [1]; hence, the potential drop across the oxide of the ON NMOS transistors becomes very small and, from equation (2.2), the tunneling gate leakage of the inverter becomes negligible. The subthreshold leakage current and power dissipation can be calculated from equation (2.1) as,

$$
\begin{aligned}
P_{standby,MTCMOS} &= V_{dd} \cdot I_{sub,standby} \\
&= V_{dd} \cdot A_{sub}\mu_0 C_{ox} V_{dd} \frac{W_{slp}}{L_{eff}} e^{\lambda(-V_{t,high}+\eta V_{dd})} \\
&= K_{standby,MTCMOS} \cdot s_{slp}
\end{aligned}
\tag{6.18}
$$

where W_{slp} and $V_{t,high}$ denote size and threshold voltage of the sleep transistor, $K_{standby,MTCMOS}$ is the subthreshold current for the minimum size sleep transistor and s_{slp} is the size of the sleep transistor normalized to that of the minimum size transistor.

Using the MTCMOS technique, the total power of one stage of two adjacent bus lines can be written as:

$$
P_{total,MTCMOS} = \chi P_{active} + (1-\chi) P_{standby,MTCMOS} \tag{6.19}
$$

In order to consider the effect of the MTCMOS on the worst case delay constraint, we need to consider two cases:

I) Adjacent bus lines are switching in the opposite direction; therefore, the sleep transistor is contributing to a single falling transition. Using equation (6.1), the time constant for one stage can be written as:

$$
\begin{aligned}
d_1 = {} & r_s\left(c_g + c_p\right) + \frac{r_s}{s}(c + 2c_c)\,l + rlsc_g + \left(\tfrac{1}{2}c + c_c\right) rl^2 \\
& + \frac{r_{slp}}{W_{slp}}\left[s \cdot \left(c_g + c_p\right) + (c + 2c_c)\,l\right]
\end{aligned}
\tag{6.20}
$$

II) Adjacent lines are switching in the same direction; when there are two simultaneous falling transitions, twice as much current has to be sunk through the sleep transistor. Therefore, the resistance of the sleep transistor should be doubled for the delay estimation. More precisely,

$$d_2 = r_s\left(c_g + c_p\right) + \frac{r_s}{s}cl + rlsc_g + \tfrac{1}{2}crl^2 + \frac{2r_{slp}}{W_{slp}}\left[s\cdot\left(c_g + c_p\right) + cl\right] \tag{6.21}$$

Note that the sleep transistors result in the delay increase only in the case of falling transitions at the output node of the repeaters. Therefore we introduce a new time constant as $d_1' = (\tau_1 + d_1)/2$ and $d_2' = (\tau_2 + d_2)/2$ where τ_1 (as in equation (6.1)) and τ_2 are the time constants for opposite and same direction transitions without any sleep transistors, respectively. The worst case delay per stage is equal to $\max\{d_1', d_2'\}$.

6.3.2 Sleep Signal Delivery Circuitry

An important issue in the design of MTCMOS circuits is how to deliver the sleep signal to all MTCMOS transistors in the design. The sleep signal should be fast enough to minimize the transition time of the system from the standby mode to active mode [1]. If the sleep signal driver circuit is improperly designed, it will result in unnecessary switching and leakage power consumption. To minimize the delay of the system for transition from the standby mode to active mode and also to reduce the power consumption of the sleep signal delivery circuit, we use asymmetric inverters in this network as depicted in Figure 6.5. In this figure, weak transistors are minimum-sized and have high threshold voltages. The rationale is that only the rise delay of the sleep signal plays a role in determining the wake-up delay of the circuit. The fall delay of the sleep signal, on the other hand, determines the active to standby mode delay which is not a critical factor. The sleep signal delivery circuit shown in Figure 6.5 not only minimizes the sleep signal propagation delay, but also linearly

reduces the capacitive power dissipation of the sleep signal delivery circuit due to selective use of minimum-size transistors. At the same time, it exponentially reduces the leakage power of the sleep signal delivery circuit during the active mode of circuit operation by using high threshold voltage transistors in each inverter (which are OFF in the active mode).

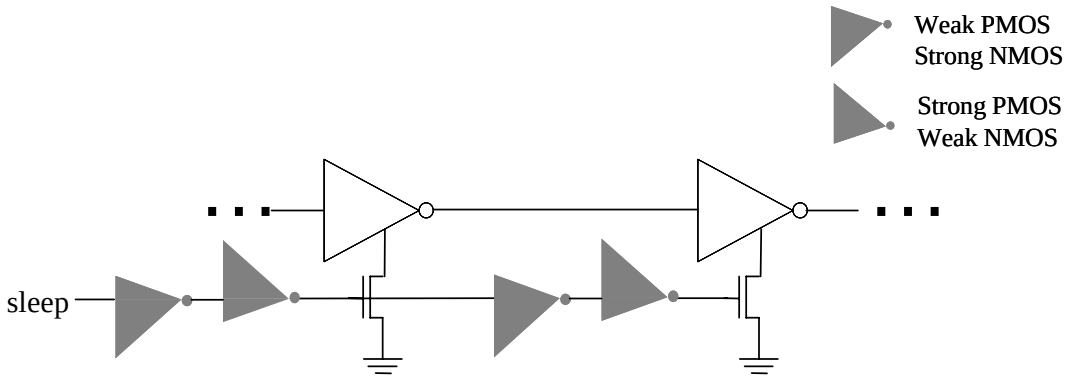


Figure 6.5: Using asymmetric inverters in the sleep signal delivery circuitry.

6.3.3 Problem Formulation

Equation (6.4) gives the optimal worst-case delay per unit length for non-MTCMOS bus lines, i.e., $(\tau / l)_{opt}$. In this section we consider the problem of power optimal design of MTCMOS bus lines. Suppose a target end-to-end delay per unit length of interconnection line is given, which is expressed as $\Delta\%$ more than $(\tau / l)_{opt}$. Given this target delay, we need to calculate the values of l, s, and W_{slp}, which minimize the total power dissipation. The total power for an interconnect of length L is equal to $P_{total-MTCMOS} \cdot (L / l)$ where $P_{total-MTCMOS}$ was given in equation (6.19). Therefore, a constrained minimization problem for $P_{total-MTCMOS} / l$ should be solved:

$$\begin{cases} Min \quad P(l,s,W_{slp}) \\ s.t. \quad (1)\ Q(l,s,W_{slp}) \le T_{req} \\ \qquad (2)\ R(l,s,W_{slp}) \le T_{req} \end{cases} \tag{6.22}$$

where $P \equiv \frac{P_{total-MTCMOS}}{l}; \quad Q \equiv \frac{d'_1}{l}; \quad R \equiv \frac{d'_2}{l}$

and $T_{req} \equiv (1+\Delta)\left(\frac{\tau}{l}\right)_{opt}$

The optimization problem can be solved by using the Lagrangian relaxation technique. The Lagrangian of problem (6.22) can be written as:

$$F = P + \lambda_1 \cdot (Q - T_{req}) + \lambda_2 \cdot (R - T_{req}) \tag{6.23}$$

From the Lagrange method, the solution of the optimization problem (6.23) should satisfy the following set of conditions:

$$\begin{cases} \frac{\partial F}{\partial s} = 0; \quad \frac{\partial F}{\partial l} = 0; \quad \frac{\partial F}{\partial W_{slp}} = 0; \\ \lambda_1 \cdot (Q - T_{req}) = 0; \quad \lambda_2 \cdot (R - T_{req}) = 0; \end{cases} \tag{6.24}$$

These equations are solved numerically and the triplet (l,s,W_{slp}) which results in minimum $P_{total-MTCMOS}/l$ is selected.

6.4 Experimental Results

To study the efficacy of the proposed technique, we conducted a comprehensive set of experiments. To extract the parameters which are used in the optimization problems, we performed transistor level simulation of devices in HSPICE [58] on a 45nm predictive technology model [99]. All simulations were carried out at the frequency of 1GHz, supply voltage of 1.1V, and die temperature of 100°C. The extracted technology parameters are reported in Table 6.2.

Table 6.2: Technology Parameters Used in the Simulation Setup

Parameter	Value	Parameter	Value
$V_{t,low}$	0.25V	K_{ox}	273 µW/µm
$V_{t,high}$	0.35V	K_{MTCMOS}	58 µW/µm
β	2.2	c_c	53.68 fF/µm
$K_{sub,N}$	881 µW/µm	c	19.41 fF/µm
$K_{sub,P}$	301 µW/µm	r	1099.99

MOSEK optimization toolbox [87] was used to solve the mathematical problem. Two coupled bus lines as described in the chapter are used for our experiments. The length of each bus line is 10mm. After optimizing the bus lines, the corresponding values of the design were extracted to SPICE netlist and detailed HSPICE simulations were performed to measure the worst-case delay and the average power consumption of the buffer chain. We first calculated the average power consumption when the worst case delay is optimized. These values are reported in Table 6.3 as P_D. The measurements were done for different active mode factors, χ. The power-optimal solutions with 10% delay penalty and for different χ, without using MTCMOS sleep transistors and with only two degrees of freedom, s and l, are reported as P_P in the table. Finally, the power optimal solutions with MTCMOS sleep transistors are reported as P_M in the table. When the percentage of the time that the circuit is in the active mode (i.e., χ) is small, the dominant component of the power consumption is the standby leakage. Therefore, MTCMOS technique results in significant power savings compared to P_D and P_P. As χ increases, the power saving diminishes. Since the active mode factor of global buses is usually very small, one can see that the power saving achieved by applying our technique is high. Note

that the sleep signal delivery was achieved by the circuit shown in Figure 6.5 and its power dissipation overhead was considered in the total power consumption results.

Table 6.3: Power consumption results for different designs activity mode factor χ.

χ (%)	P_D (μW)	P_P (μW)	P_M (μW)	P_M reduction over P_D (%)	P_M reduction over P_P (%)
1	59.1	24.2	9.9	83.3	59.3
2	66.1	28.0	11.6	82.4	58.6
5	87.3	39.4	22.4	74.4	43.2
10	122.6	58.4	46.3	62.2	20.7
20	193.1	96.3	89.3	53.8	7.3
30	263.7	134.2	132.9	49.6	1.0

Table 6.4: Power consumption results for different delay penalties.

Δ	P_P (μW)	P_M (μW)	P_M reduction over P_D (%)	P_M reduction over P_P (%)
5%	73.1	56.1	54.2	23.2
10%	58.4	46.3	62.2	20.7
15%	51.2	41.1	66.5	19.7
20%	49.1	36.7	70.0	25.3
25%	43.0	36.1	70.5	15.9
30%	38.0	32.7	73.4	14.0
35%	37.7	29.3	76.1	22.3
40%	33.2	29.0	76.4	12.7

In the second set of our experiments, where results are presented in Table 6.4, we compared the efficacy of the proposed technique for different values of delay penalty. More precisely, here the value of χ assumed to be 10% and the delay penalty Δ was varied from 5% to 40%. For each case, P_P and P_M were measured by HSPICE simulation. As we increase the delay penalty, the power reduction in both P_P and P_M increases. This power saving saturates as we increase Δ. Table 6.5 reports the optimal parameter values for the power-optimized design using the MTCMOS technique. The design parameters are normalized with respect to the

delay-optimized repeater size (s_{opt}) and insertion length (l_{opt}). It is observed that by increasing Δ, both repeater and sleep sizes are decreasing. However, decrease in the sizes diminishes as the delay budget increases.

Table 6.5: Design parameters for the optimized MTCMOS design.

Δ	s/s_{opt}	l/l_{opt}	W_{slp}/s_{opt}
5%	0.79	1.21	3.89
10%	0.70	1.43	2.90
15%	0.63	1.57	2.47
20%	0.57	1.71	2.20
25%	0.53	1.82	2.01
30%	0.51	1.93	1.88
35%	0.48	2.07	1.77
40%	0.45	2.14	1.68

Finally, we compared our results with a two-step approach to design MTCMOS repeaters. In this two-step approach, first the power-optimal solution with no sleep transistor is found; then the size of sleep transistors is calculated based on the power-optimal l and s values of the first step. We assume equal $\Delta\%$ in each step of this approach. Therefore, for a fair comparison we have to compare the two-step approach results with our solution with $\left(2\Delta + \Delta^2\right)\% \approx 2\Delta\%$ delay penalty.

Table 6.6: Comparing the proposed technique with a two-step approach to design MTCMOS repeaters

Delay Penalty	P_T (µW)	P_M (µW)	P_M reduction over P_T (%)
5%	56.7	56.1	0.9
10%	49.6	46.3	6.8
15%	44.6	41.1	8.0
20%	40.2	36.7	8.7
25%	39.7	36.1	9.0
30%	35.8	32.7	8.7
35%	35.3	29.3	17.1
40%	34.8	29.0	16.8

Table 6.6 compares the average power consumption achieved by our technique with that of two-step approach, denoted as P_T. It is seen that on average, our approach gives about 9.5% improvements in average power consumption over the two-step solution.

6.5 Summary

This chapter addressed the problem of power-optimal repeater insertion for global buses in the presence of crosstalk noise. We used MTCMOS technique by inserting high-V_t sleep transistors to reduce the leakage power consumption in the idle mode. By accurately modeling different components of the power consumption and the delay, a mathematical problem was formulated for minimizing the average power under a timing constraint. Detailed HSPICE simulation showed that by considering the effect of crosstalk on both delay and power consumption, and by using MTCMOS technique, the average power consumption of the bus lines can be reduced by more than 50% with a small delay penalty of 5%.

Chapter 7
Optimal Voltage Regulator Module Selection in a Power Delivery Network

7.1 Introduction

Utilizing multiple voltage domains (also known as voltage islands [76]) is one of the most effective techniques to minimize the overall power dissipation–both dynamic and leakage– while meeting a performance constraint. In a system designed with multiple voltage domains, the power delivery network (PDN) is responsible for delivering power with appropriate voltage levels to different functional blocks (FB's) on the chip. Voltage regulator modules (VRM's) which are in charge of voltage conversion and regulation are inevitable components in this network. The selection of appropriate VRM's plays a critical role in the power efficiency of the PDN. Typically a star configuration of the VRM's, where only one VRM resides between the power supply and each FB, is used to deliver currents with appropriate voltage levels to different loads in the circuit. In this chapter we show that using a tree topology of suitably chosen VRM's between the power source and FB's yields higher power efficiency in the PDN. We formulize the problem of selecting the best set of VRM's in a tree topology as a dynamic program and efficiently solve it.

The remainder of this chapter is organized as follows. Section 7.2 and Section 7.3

respectively provide some background on power delivery network design and voltage regulator modules. Our idea for optimal selection of VRM's in PDN is presented in Section 7.4. Section 7.5 is dedicated to experimental results, while Section 7.6 summarizes the chapter.

7.2 Power Delivery Network Design Methodology

The power delivery network is a critical design component in large designs, especially for high-speed systems. A robust PDN is required to achieve a high level of system signal integrity [36]. If improperly designed, this network could be a major source of noise, such as IR-drop, ground bounce, and electromagnetic interference (EMI) [23, 33, 34]. In today's high-performance microprocessors, it is typical for the circuit to draw over 100A current from the PDN in a fraction of nano-second, yielding the derivative of the current over 100GA/s. However, with careful design, a PDN can tolerate large variations in load currents while maintaining the supply voltage level across the chip within a desired range [36].

Emerging low-power design techniques have made the design of PDN an even more challenging task. More precisely, multiple voltage domains are being introduced on the SoC in order to minimize the overall power dissipation of the system while meeting a performance constraint. In these systems, it is required that the PDN delivers power at appropriate voltage levels to FB's while incurring the minimum power loss. Consequently, PDN design for a high-performance SoC comprises of three steps:

- Establishing PDN target impedance,
- Designing a proper system-level decoupling network,
- Selecting the right voltage regulator modules.

A methodology for designing a good PDN is to define a target impedance for the network that should be met over a broad frequency band [120]. This parameter can be computed by assuming $\alpha\%$ allowable ripple in the voltage supply and 50% switching current in the rise and fall time of the processor clock. The target impedance can then be calculated as [120]:

$$Z_{target} = \frac{\alpha\% \times V_{dd}}{50\% \times I} \tag{7.1}$$

where V_{dd} is the core voltage of the processor and I is the current drawn by the microprocessor from the PDN. For the 65nm node, $I = 100/1.1 = 91A$. If 5% ripple is allowed on the voltage supply, the calculated target impedance will be $1.2m\Omega$. With the general scaling theory, following the Moore's law, the current I is increasing, while the power supply voltage V_{dd} is decreasing. Therefore, to satisfy the power supply noise constraint, the impedance of the power supply should be decreased.

Since the current drawn by digital circuits can change suddenly with different frequencies, the target impedance should be met over a broad frequency range to guarantee the ripple on the voltage supply does not exceed the allowable value. To meet this requirement, on-chip and off-chip decoupling capacitors (decaps) need to be suitably placed in the design. Decaps play an important role in the PDN as they act as charge reservoirs providing instantaneous current for switching circuits.

Current surface-mount ceramic capacitors provide good IC decoupling up to around 100-300MHz. Decoupling in higher frequencies can be achieved by deploying on-chip capacitors. The amount of on-chip capacitance that can be added is limited to the real estate on-chip. Fabrication data demonstrate that for 90nm technologies, tunneling gate leakage of a 1nF decap is in the order of milliamperes [52] and for more advanced CMOS process technologies it is expected to be even higher. Therefore, the leakage current of the decaps adds to the total power consumption of the circuit and shortens the battery lifetime. These facts emphasize that to achieve a low-power and low-cost design, the added decap should be minimized. In the past, much research has been conducted to address the problem of decap allocation. In [146], for example, the problem of decap allocation during initial floorplanning stage was formulated as a linear program. In [123] the authors proposed a technique for sizing and placing decaps in a standard cell layout. In [140] the authors presented a multigrid-based technique for simultaneously optimizing the power grid and decap. With the aid of macromodeling and the concept of an effective radius of a decap, the authors of [145] proposed an efficient charge-based method for decap allocation.

Every electronic circuit is designed to operate off of some supply voltage, which is usually assumed to be constant. A voltage regulator module (VRM) provides this substantially constant DC output voltage regardless of changes in load current or input voltage (this statement assumes that the load current and input voltage are within the specified operating range for the part). A switching power supply is a device transforming the voltage from one level to another. Typically voltage is taken from the AC power lines or unregulated DC power lines and transformed to the

regulated DC levels that logic circuits require. Each IC specifies its voltage regulator configuration in its datasheets or comes with a companion document that defines the power delivery feature set necessary to support that IC within a larger electronic system. For example, the Intel's VRM version 10.2 describes the Intel® processors' V_{cc} power delivery requirements for desktop computer systems using socket 478. This includes design recommendations for DC-DC regulators which convert the 12V supply to the processor consumable V_{cc} voltage along with specific feature set implementation such as thermal monitoring and Dynamic Voltage Identification.

In a large PCB design or equivalently in a complex SoC design, there are many functional blocks (FB's) providing various functionalities. Examples of processing elements are DSP or CPU cores. Examples of other FB's are random logic or interface blocks, MPEG encoder/decoder blocks, RF front-end, on-chip memory, and various controllers. The V_{cc} regulator design on a specific platform (PCB or SoC) must meet the specifications of all FB's supported in that platform.

Another low power design trend is emerging that makes the design of the *VRM tree*[1] even more important. More precisely, multiple voltage domains are being introduced on the same SoC in order to meet a performance constraint while minimizing the overall power dissipation of the system. This means that it is possible to have multiple logic blocks operated at different, yet fixed, voltages [97] (the question of VRM tree design to support dynamic voltage scaling based on workload monitoring will be addressed in Chapter 8). This is also known as the multiple

[1] The graph representation of the VRM network will have a tree structure, that is, no VRM can be driven by more than one other VRM.

voltage island approach [76]. Figure 7.1 depicts the role of the VRM's in providing appropriate voltage levels to different FB's on a single chip.

Traditionally, off-chip VRM's have been utilized to provide appropriate voltage levels to different FB's on a chip. In a multi-supply-voltage SoC, however, keeping the VRM's off-chip not only increases the total cost of he system, but also requires valuable board space, lowers the system reliability, and creates more rigid requirements on the VRM due to losses on the board. On the other hand, one of the main advantage of deploying on-chip regulator is that because the VRM is located close to the load, the impedance between the VRM and load is small, resulting in minimum noise on the power supply [36]. Consequently, utilizing on-chip voltage regulators have become popular for low-power applications, particularly in compact handheld devices [52] [95].

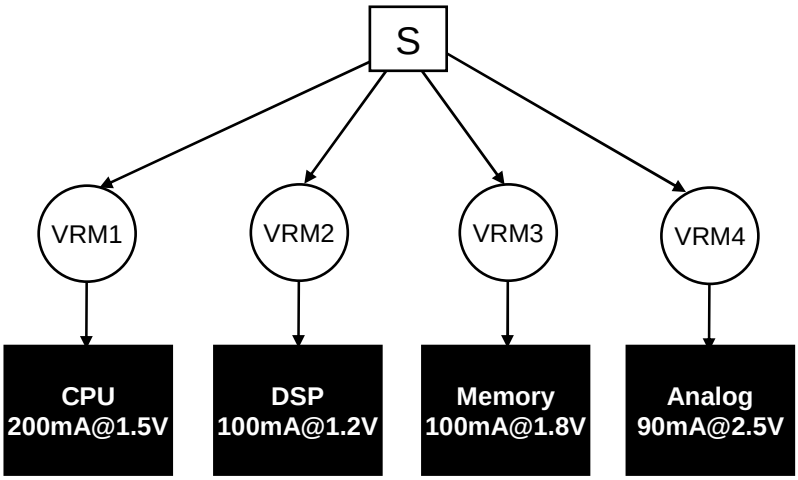


Figure 7.1: The role of VRM tree in providing appropriate voltage level for each FB.

7.3 Voltage Regulators

A voltage regulator module is an electrical device designed to automatically maintain a constant voltage level, regardless of changes in input voltage or output current. The

output voltage of a VRM may not be equal to the DC of the input voltage. If the output voltage of the VRM is smaller than the input voltage, the VRM is called *step-down* (*buck*) and if the output voltage is greater than the input voltage, it is called *step-up* (*boost*). Let the range of input voltages and load currents over which a regulator can maintain a target voltage level within the specified tolerance band (e.g., 1.3V with ±5% ripple) be specified. The VRM's *power efficiency* may be calculated as the ratio of the power that is delivered to the load to the power that is extracted from the input source, i.e.,

$$\eta = \frac{V_{out} I_{out}}{V_{in} I_{in}}. \tag{7.2}$$

Power efficiency is one of the most important figures of merit for a VRM and is a function of the input voltage and output current of the VRM. Figure 7.2 shows the efficiency of a commercial VRM as a function of input voltage and output current.

Another important figure of merit of a VRM is its *load regulation* which is a measure of the ability of the VRM to keep its output voltage fixed in spite of load current variations. More precisely, load regulation is defined as percent of change in the output voltage relative to the change in the output current, i.e.

$$\varphi = \frac{\Delta V_{out}}{\Delta I_{out}}. \tag{7.3}$$

Fast load regulation is important when the VRM is used to power-up digital CMOS circuits with rapidly changing load current demands, for example a power-gated circuit which transits between sleep and active modes. From the definition, one can see that load regulation of a VRM is equal to its output impedance. Usually a feedback loop is utilized in the VRM to keep the output voltage fixed; in this

scheme, to meet the load regulation requirements, the loop gain of the feedback should be high across all operating frequencies.

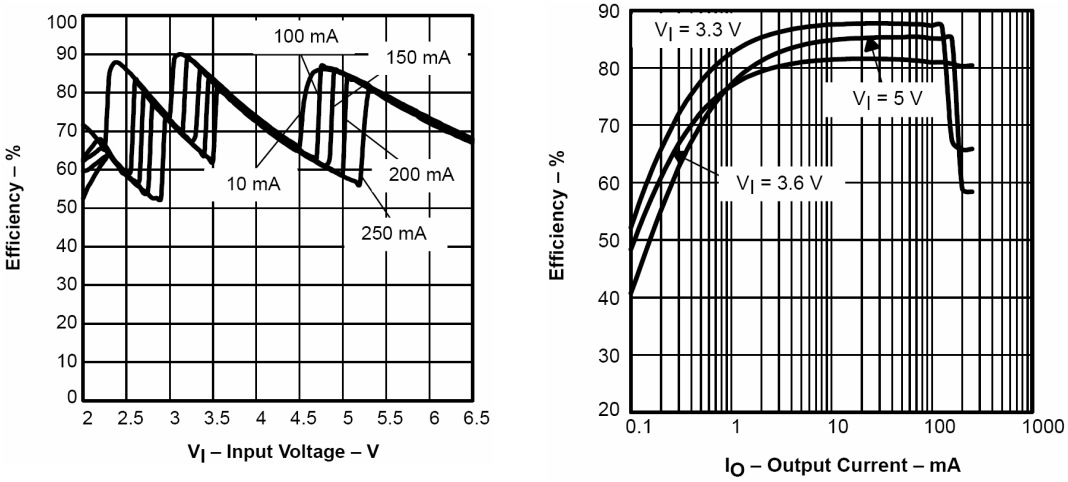


Figure 7.2: The efficiency of TPS60503 as a function of input voltage and output current [129].

Each VRM has an associated cost which depends on its complexity, silicon area, and passive element costs. For example, because of their inductors, regulated inductor-based VRM's are usually the most expensive type of DC-DC converters. Linear regulators, on the other hand, are typically the least expensive ones.

7.3.1 Voltage Regulation Topologies

Based on how voltage conversion is achieved, VRM's are classified into two main categories: *linear regulators* and *switching regulators*. A linear regulator is based on an active device, such as a BJT or a MOSFET, continuously adjusting a voltage divider network to maintain a constant output voltage. A switching regulator is a device transforming the voltage from one level to another with utilizing low-pass components such as capacitors, inductors, or transformers – and switches that are in one of two states, ON or OFF. In *charge-pump switching regulators* (also known as

switched-capacitors[1]), capacitors are utilized as energy storage elements, whereas in inductor-based switching regulators, inductors are the energy storage components. The advantage of using switching regulator is that the switch dissipates very little power in either of these two states and power conversion can be accomplished with minimal power loss, which equates to high power efficiency.

Typically a switching regulator is a circuit that operates in a closed loop system to regulate the power supply output, for example through *pulse-width modulation* (PWM) or *pulse-frequency modulation* (PFM). Since in PWM control the switching frequency is fixed, the noise spectrum is narrow, allowing simple low pass or notch filter techniques to reduce the peak-to-peak ripple voltage. However, having a fixed switching frequency has the disadvantage that the losses are independent of load current; therefore, when the load current is low, the regulators based on PWM control dissipate a large amount of dynamic power relative to the output power [122].

In recent years, inductor-based switching regulators integrated in standard foundry-available digital CMOS processes have been demonstrated [37, 77]; however, this kind of regulators tends to be bulky and the quality of fabricated on-chip inductors is still low, resulting in low efficiency of this kind of regulators. Charge-pump regulators are an alternative to inductor-based regulators and offer comparable power efficiency while reducing the cost and physical volume of the VRM. Since the solution we present in this chapter is targeted toward the PDN

[1] Charge-pump and switched-capacitor are used interchangeably throughout this manuscript.

design of complex SoC's, in the remainder of this manuscript, we do not consider the class of inductor-based VRM's for the design of PDN and limit the class of regulators to linear and charge-pump ones.

7.4 VRM Selection for Minimum Power Loss

The VRM tree optimization (RMTO) problem is defined next.

RMTO Problem: Given is

- A library $\boldsymbol{\mathcal{R}}$ of VRM's and for each $r \in \boldsymbol{\mathcal{R}}$, its output voltage $\upsilon_{r,out}$, the minimum and maximum input voltages $\upsilon_{r,in}^{\min}$ and $\upsilon_{r,in}^{\max}$, the maximum load current $\iota_{r,out}^{\max}$, and its efficiency η_r as a function of load current and input voltage,
- A power source P, with the nominal voltage of V_P,
- A set $\boldsymbol{\mathcal{F}}$ of FB's, and for each $f \in \boldsymbol{\mathcal{F}}$ its required voltage V_f and average current demand I_f.

The goal is to build a tree topology of VRM's that connects P to all FB's and minimizes the PDN power loss from the power source to the loads while meeting the voltage and current constraints.

In this remainder of this chapter, we focus on this RMTO problem statement. An interesting variant of the problem, which we do not address here, may be defined as follows. Given a cost δ_r associated with each regulator r, minimize the power loss in the PDN while ensuring that the total cost of the VRM tree does not exceed a cost budget.

It should be noted that the power delivered to the FB's is independent of the topology of the VRM tree and is calculated as,

$$P_{FBs} = \sum_{f \in \mathcal{F}} V_f I_f \,. \tag{7.4}$$

Therefore, to minimize the power loss in the PDN from the power source to the loads, one needs to minimize the power drawn from the power supply. Given that the voltage of the power supply is assumed to be fixed, the objective of RMTO problem is to minimize the current drawn from the power supply. We assume that each VRM can provide only one output voltage (multi-output VRM's are considered as multiple VRM's, each with its own fixed voltage output).

Although RMTO problem definition does not put any constraints on the depth of the VRM tree that drives the loads, in practice, such a constraint is useful. The reason is that utilizing a VRM tree with a large number of internal levels tends to increase the number of regulators, which in turn increases their cost and chip area overhead with little (if any) benefit in terms of improving the power efficiency of the PDN. For this reason, in this work, we only consider up to two levels of regulators in the VRM tree, i.e., the (node) depth of the tree is 4, with one corresponding to the power source P, one corresponding to the loads and up to two internal levels dedicated to VRM's. Our solution, however, can be easily extended to handle VRM trees with higher depth.

To improve the efficiency of our solution technique by implicitly considering a large class of tree topologies under one class representative, it is convenient to introduce an *ideal VRM* whose efficiency is 100% and whose output voltage and thus output current are equal to its input voltage and current, respectively. This ideal VRM (really a lossless buffer) is added to library $\mathcal{R}$ of VRM's. Note that ideal

VRM's are inserted on every path from the tree root to a leaf node in the tree so that the logical depth of each such path is exactly four (c.f. Figure 7.3).

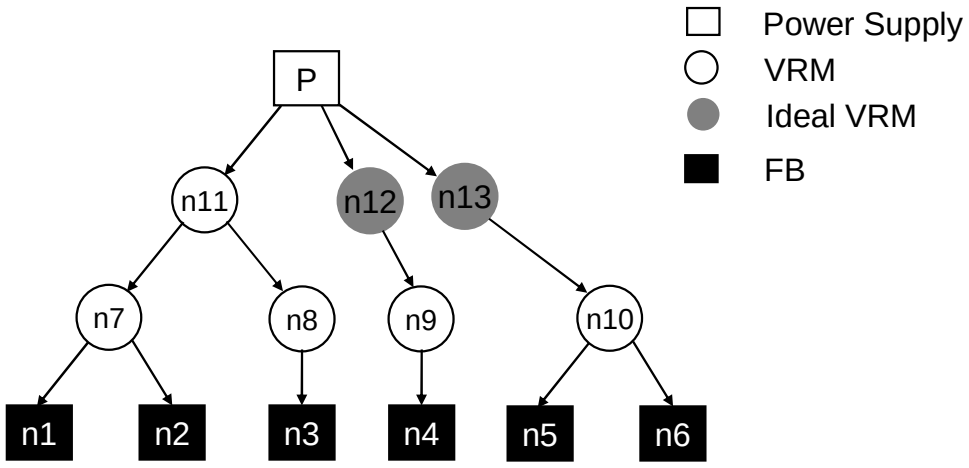


Figure 7.3: A VRM tree after inserting ideal VRM's.

Definition 7.1: A VRM satisfies *monotone input current property* if its input current is a monotone increasing function of its output current independent of the input voltage.

Notice that the monotone input current property may hold in spite of the non-monotone power efficiency characteristics for a VRM. This is because of the way that power efficiency is defined and its relation to input and output voltages and currents. In the next session we elaborate on this assumption in more detail.

If the tree topology is *fixed* (-F option) and the *monotone input current property* holds for all VRM's in the library (-M option), then the selection of the appropriate regulator for each node can be done optimally by using dynamic programming starting from the leaf nodes. This algorithm, called RMTO-FM, will be presented in the next section. Table 7.1 introduces notations which will be used in RMTO-FM algorithm.

Table 7.1: Notation used in RMTO algorithm

$\mathcal{R}$	Set of all VRM's including the ideal VRM
$\mathcal{F}$	Set of all FB's
$\mathcal{U}$	Set of all output voltages of the VRM's
$\mathcal{V}_n$	Set of candidate input voltages for node n
$\mathcal{V}_{n,r}$	Set of candidate input voltages for n when r is the VRM of n
$\mathcal{C}_n$	Set of candidate VRM's for internal node n of the tree
T	Topology of VRM tree
$\pi(n)$	Optimum VRM selection for node n
L_i	Set of all level i internal nodes, $i = 1,2$
V_f and I_f	Voltage level and current demand of FB $f \in \mathcal{F}$
$\upsilon_{r,out}$	Output voltage level of regulator r
$\upsilon_{r,in}^{\min}$ and $\upsilon_{r,in}^{\max}$	Minimum and maximum input voltage levels of regulator r
$\iota_{c_n,out}^{\max}$	Maximum output current of regulator r
$V_{out}(n)$	Output voltage of a node n
$I_{out,r}(n)$ and $I_{in,r}(n)$	Output and input current of node n given that regulator r is assigned to this node
$\mu_r(\upsilon_{in}, i_{out})$	Efficiency of regulator r as a function of its input voltage υ_{in} and output current I_{out}
$\Psi_n(\upsilon_{in})$	One dimensional table in node n with the key υ_{in} and the value of input current of node.

7.4.1 RMTO for Fixed-Tree Topology

The algorithm for solving RMTO-FM problem is shown in Figure 7.4. This algorithm starts with the nodes in the second internal level of the tree T. If any such node is connected to two FB's with different input voltage requirements, then the tree will not be a feasible VRM tree (a precise definition is provided later) and the algorithm terminates; otherwise, the output current of the node is calculated as the sum of the current demands of all leaf nodes (FB's) that are connected to it. Next all candidate VRM's with compatible output current and voltage are evaluated. Since the input voltage of the node is not known at this time, the power efficiency of each

candidate VRM for the node in question cannot be calculated directly. Based on the fact that this second-level node is driven by any first-level VRM node, all voltage values in $\boldsymbol{u}$ must be enumerated. Thus the power efficiency of the candidate second-level VRM is obtained from the efficiency curves for each regulator. This information is then used to compute the input current of the second-level node as the minimum of the input currents of the candidate VRM's which take the specific input voltage level for the second-level node. The calculated input current is stored in a one dimensional table with the key set to the input voltage of the second-level node and the value set to the input current of that same node.

The first-level nodes are visited next. For each such node n, all candidate output voltages $v_{out}(n)$ (defined as the voltages in the intersection of all $\boldsymbol{v}_m$'s, where m denotes a fanout of n) are considered. Next a set of output voltages are identified where each of these output voltages show up in every input current vs. input voltage table stored at each fanout of n. For every such output voltage, the sum of the input currents of the driven second-level nodes is computed and set as the target output current of the first-level node. Next based on the output current of that first-level node and the known input voltage of the same node (which is the same as the output voltage of the power source for the VRM tree), the optimum VRM assignment for the first-level node is determined by enumerating all possible VRM matches at that node, i.e., a VRM assignment is chosen that minimizes the input current of the first-level node (and hence the output current demand on the power source along the edge that leads to that node) while providing the output current needed by driven second-level nodes under the selected output voltage assignment for the first-level node.

$RMTO\text{–}FM(\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{L}},T,V_P)\{$
  For each second level node $n\{$
    If$\left(V_{f_1} \neq V_{f2} : f_1, f_2 \in FO(n), f_1 \neq f_2\right)$
      Exit(0);
    $V_{out}(n) = V_f : f \in FO(n);$
    $I_{out}(n) = \sum_{f \in FO(n)} I_f;$
    $\boldsymbol{\mathcal{C}}_n = \{r \in \boldsymbol{\mathcal{R}} | \upsilon_{r,out} = V_{out}(n),\ \iota^{\max}_{r,out} > I_{out}(n)\};$
    $\boldsymbol{\mathcal{V}}_n = \varnothing;$
    For each $v_{in} \in \boldsymbol{\mathcal{U}}\{$
      For each $r \in \boldsymbol{\mathcal{C}}_n$
        $I_{in,r}(v_{in}) = \dfrac{V_{out}(n) \times I_{out}(n)}{\upsilon_{in} \times \eta_r(v_{in}, I_{out}(n))};$
      $r = \arg\min_{r \in \boldsymbol{\mathcal{C}}_n}\left(I_{in,r}(v_{in})\right);$
      If $r \neq \varnothing\{$
        $\Psi_n(v_{in}) = I_{in,r}(v_{in});$
        $\boldsymbol{\mathcal{V}}_n = \boldsymbol{\mathcal{V}}_n \bigcap v_{in};$
      $\}$
    $\}$
  $\}$
  For each first level node $n\{$
    $\boldsymbol{\mathcal{C}}_n = \{r \in \boldsymbol{\mathcal{R}} | \upsilon_{r,out} \in \bigcap_{i \in FO(n)} \boldsymbol{\mathcal{V}}_i,\ \upsilon^{\min}_{r,in} \leq V_P \leq \upsilon^{\max}_{r,in}\};$
    For each $r \in \boldsymbol{\mathcal{C}}_n\{$
      $V_{out}(n) = \upsilon_{r,out};$
      $I_{out,r}(n) = \sum_{m \in FO(n)} \Psi_m(V_{out}(n));$
      $I_{in,r}(n) = \dfrac{V_{out}(n) \times I_{out,r}(n)}{V_P \times \eta_r\left(V_P, I_{out,r}(n)\right)};$
    $\}$
    $(\pi(n), \pi(m) : m \in FO(n)) = \arg\min_{r \in \boldsymbol{\mathcal{C}}_n}\left(I_{in,r}(n)\right);$
  $\}$
$\}$

Figure 7.4: RMTO-FM algorithm for VRM tree optimization when tree topology is fixed.

7.4.2 RMTO for Varied-Tree Topology

The optimal solution of VRM tree problem when the tree topology may be varied

(-V option) is found by enumerating all *feasible* trees with exactly two internal nodes and $|\mathcal{F}|$ leaf nodes.

Definition 7.2: A VRM tree topology is *feasible* when (i) it has an exact depth of 4, i.e., every path from the root to a leaf node comprises of the zeroth level node corresponding to the tree root, the third-level node corresponding to the leaf node, with two levels of internal nodes in between; (ii) the leaf nodes under any second-level internal node in the tree have the same voltage assignments.

Since each VRM can only provide one output voltage level, the number of VRM's in a feasible VRM tree topology cannot be less than the number of distinct voltage levels of the FB's. The number of possible combinations for the first level of the tree is the power set of the number of second-level nodes in that tree. After generating each feasible tree instance T, the RMTO-FM algorithm is used to find the optimum solution for the corresponding T.

7.4.3 Efficient Generation of Feasible Trees

One issue with RMTO-VM procedure is that the number of feasible trees with n leaves appears to be quite large; fortunately, in the RMTO problem, many of the generated trees are isomorphic (c.f. Figure 7.5).

Definition 7.3: Two VRM trees T_1 and T_2 are called *inter-isomorphic* if by a change of labeling in the intermediate vertices of one tree, it becomes equal to the other; otherwise, they are called *non-inter-isomorphic*. The set of all non-inter-isomorphic trees comprising of exactly two internal levels and n leaf nodes is denoted by $\mathcal{T}_2(n)$.

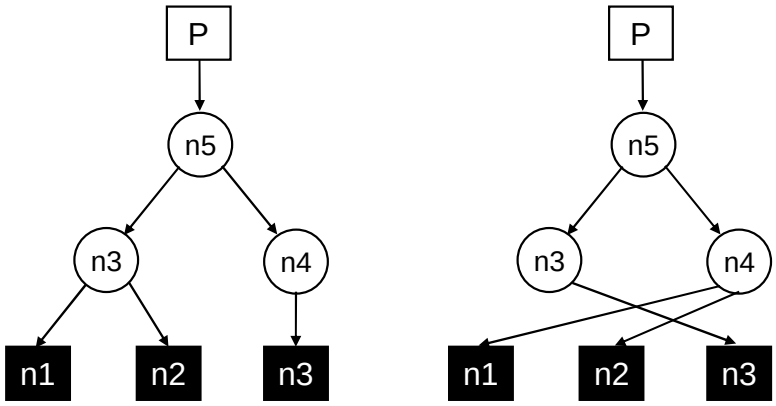


Figure 7.5: Two inter-isomorphic trees.

It is clear that to find the optimal solution of VRM tree problem when the tree topology may be varied, only the set of non-inter-isomorphic feasible trees should be enumerated. In the remaining of this section we provide a mathematical framework to efficiently generate the set of non-inter-isomorphic trees.

Definition 7.4: A *partition* of set X is a collection of disjoint nonempty subsets of X whose union is X. Each of these subsets is called a *part*.

The number of partitions of a set with n elements is the n'th Bell number which can be computed from the following recurrence [45],

$$B_n = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k;\ B_0 = 1. \tag{7.5}$$

Definition 7.5: For each n and $m \leq n$, the Stirling number of the second kind, denoted as $\left\{ {n \atop m} \right\}$, is the number of ways of partitioning a set of n elements into m nonempty sets. These numbers can be computed from the following recurrence [45]:

$$\left\{ {n \atop m} \right\} = \left\{ {n-1 \atop m-1} \right\} + m \left\{ {n-1 \atop m} \right\};\quad \left\{ {n \atop 1} \right\} = 1 \text{ and } \left\{ {n \atop n} \right\} = 1. \tag{7.6}$$

From the definition, one can see Bell numbers are related to Stirling numbers of the second kind by the following relation:

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}. \tag{7.7}$$

Definition 7.6: A restricted growth function (RGF) of length n is a function $\phi : \{1,...,n\} \to \mathbb{N}$ which satisfy the following conditions [74]:

$$\begin{aligned} &\phi(1) = 1 \\ &\phi(i) \le \max\{\phi(1),...,\phi(i-1)\} + 1 \quad (2 \le i \le n). \end{aligned} \tag{7.8}$$

Each RGF is represented as an n-tuple $[f(1),...,f(n)]$, where in entry in the tuple is a positive integer. The set of all RGF's of length n is denoted as Φ^n.

For example, both $[1,1,2,1,3]$ and $[1,2,3,4,2]$ are RGF's of length five, while $[1,2,4,3,1]$ and $[1,1,2,2,4]$ are not. Moreover, it can be easily verified that $\Phi^3 = \{[1,1,1],[1,1,2],[1,2,1],[1,2,2],[1,2,3]\}$.

Lemma 7.1: There is a one-to-one correspondence between Φ^n and the set of all partitions of $\{1,...,n\}$ [74].

In particular, each $\phi \in \Phi^n$ represents a partition into $m \le n$ groups, where 1 by convention belongs to the first part, i belongs to the $\phi(i)^{\text{th}}$ part, and $\max\{\phi(1),...,\phi(n)\} = m$. For example, the RGF $[1,1,2,3,2,4,1]$ represents the partition $\{\{1,2,7\},\{3,5\},\{4\},\{6\}\}$. For each RGF ϕ, $\nabla\phi$ is defined as $\nabla\phi = \max\{\phi(1),...,\phi(n)\}$. From Lemma 7.1 one can see that $|\Phi^n| = B_n$. Moreover, it is seen that the number of $\phi \in \Phi^n$ with $\nabla\phi = m$ is exactly $\left\{ {n \atop m} \right\}$.

Lemma 7.2: There is a one-to-one correspondence between the set of $\Lambda^n = \bigcup_{\phi \in \Phi^n} \{\phi\} \times \Phi^{\nabla\phi}$ and the set of all non-inter-isomorphic trees with exactly two internal levels and n leaf nodes, i.e., $\mathfrak{T}_2(n)$.

Proof: We show that each $T \in \mathfrak{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$ and vice versa. Without loosing generality we assume leaf nodes of T are numerically labeled from left to right. The labeling starts from one and monotonically increments. By using the algorithm shown in Figure 7.6, we assign new labels for the intermediate nodes.

```
VRM_tree_labeling{
  For l = 3 to 2{
    largest_label = 0 ;
    For i = 1 to |L_l|{
      c_node = level-l node with label i ;
      p_node = parent (c_node) ;
      If p_node does not have a label
        label(p_node) =++ largest_label;
      p_{l,i} = label(p_node);
    }
  }
}
```

Figure 7.6: VRM_tree_labeling algorithm.

In this algorithm, $|L_l|$ is the number of nodes at level l. By using this notation, each $T \in \mathfrak{T}_2(n)$ can be represented by the canonical form $\left([p_{3,1},\ldots,p_{3,|L_3|}],[p_{2,1},\ldots,p_{2,|L_2|}]\right)$. Figure 7.7 shows the labeling of a VRM tree after applying $VRM\_tree\_labeling$ algorithm. It is seen that the canonical representation of this tree is $([1,2,2,1,3,4],[1,2,3,1])$.

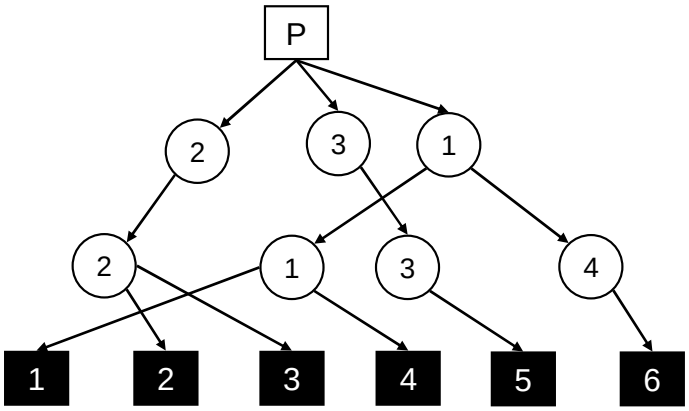


Figure 7.7: An example of VRM tree labeling.

It can be easily verified that both $p_3 = [p_{3,1},\ldots,p_{3,|L_3|}]$ and $p_2 = [p_{2,1},\ldots,p_{2,|L_2|}]$ are RGF and $(p_3, p_2) \in \Lambda^n$. Moreover, if two trees are mapped to one $\lambda \in \Lambda^n$, according to Definition 7.3 they are inter-isomorphic and therefore they both cannot be in $\mathfrak{T}_2(n)$. This means that each $T \in \mathfrak{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$. Next, we show that each $\lambda \in \Lambda^n$ can be mapped to one and only one $T \in \mathfrak{T}_2(n)$. Assume $\lambda = ([p_{3,1},\ldots,p_{3,n}],[p_{2,1},\ldots,p_{2,\nabla p_3}]) \in \Lambda^n$, the $build\_VRM\_tree$ algorithm given in Figure 7.8 shows how a $T \in \mathfrak{T}_2(n)$ can be constructed from λ. It can be easily verified that if $\lambda_1, \lambda_2 \in \Lambda^n$ and $\lambda_1 \neq \lambda_2$, they cannot be mapped to a unique $T \in \mathfrak{T}_2(n)$; therefore, each $\lambda \in \Lambda^n$ can be mapped to one and only one $T \in \mathfrak{T}_2(n)$ and since it was shown that each $T \in \mathfrak{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$, it is concluded that there is a one-to-one correspondence between Λ^n and $\mathfrak{T}_2(n)$. ■

```
build_VRM_tree{
  Construct a rooted tree with |L3| = n, |L2| = ∇p3, and |L1| = ∇p2;
  Label nodes in each level-l from 1 to |Ll|;
  For l = 3 to 2{
    For i = 1 to |Li|
      parent(level−l node i) = p_{l,i};
  }
  For i = 1 to |L1|
    parent(level−1 node i) = root;
}
```

Figure 7.8: Build_VRM_tree algorithm.

Lemma 7.3: The number of all non-inter-isomorphic trees with exactly 2 internal levels and n leaf nodes is obtained from

$$|\mathfrak{T}_2(n)| = \sum_{m=1}^{n} B_m \left\{ {n \atop m} \right\}. \tag{7.9}$$

Proof: According to Lemma 7.2, there is a one-to-one correspondence between the set of $\Lambda^n = \bigcup_{\phi \in \Phi^n} \{\phi\} \times \Phi^{\nabla\phi}$ and $\mathfrak{T}_2(n)$; therefore, $|\mathfrak{T}_2(n)| = |\Lambda^n|$. Since the number of $\phi \in \Phi^n$ with $\nabla\phi = m$ is $\left\{ {n \atop m} \right\}$, the number of elements in Λ^n is calculated as,

$$|\Lambda^n| = \sum_{\phi \in \Phi^n} \left|\Phi^{\nabla\phi}\right| = \sum_{m=1}^{n} \left\{ {n \atop m} \right\} |\Phi^m| = \sum_{m=1}^{n} \left\{ {n \atop m} \right\} B_m \,. \tag{7.10}$$

Therefore, Equation (7.9) holds. ■

Table 7.2 shows the number of non-inter-isomorphic trees with 2 internal levels and n leaves. From the table data, it is seen that by using only non-inter-isomorphic trees, the number of enumerations required to find the optimal solution becomes more manageable.

Table 7.2: Number of non-inter-isomorphic trees with *n* leaves

n	1	2	3	4	5
$\lvert\mathfrak{T}_2(n)\rvert$	1	3	12	60	358

An algorithm for solving the RMTO-VM problem is presented in Figure 7.9. It should be noted that although the time complexity of RMTO-VM algorithm is exponential in the number of leaf nodes, because the number of different voltage domains is small, the runtime of the algorithm is quite reasonable.

$RMTO{-}VM\,(\,\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{F}},V_P\,)\{$
  $\text{For each } T \in \mathfrak{T}_2(n)\{$
    $\text{If } T \text{ is feasible}$
      $RMTO{-}FM\,(\,\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{F}},T,V_P\,)\,;$
  $\}$
  $\text{Return best } RMTO{-}FM\,(\,\boldsymbol{\mathcal{R}},\boldsymbol{\mathcal{F}},T,V_P\,)\,;$
$\}$

Figure 7.9: RMTO-VM algorithm for VRM tree optimization.

7.4.4 Practical Issues

7.4.4.1 Non-Monotone Input Current

The monotone input current property holds as long as the VRM has a single mode, where the basic feedback loop in the regulator which performs the output and line regulation does not change its parameters (reference voltage levels, sensing network parameters, switch configuration, etc) in response to applied input voltages. There are, however, VRM's that may operate as say 2X charge pump or 1.5X charge pump or even an LDO depending on the applied input voltage. Such VRM's tend to exhibit a non-monotone input current vs. output current behavior, which will then break the principle of dynamic programming and require an exhaustive search mechanism to produce the optimum VRM tree solution.

Two changes in the RMTO-FM algorithm are needed to solve the RMTO-FN problem (-*N* option means some of the VRM's have non-monotone input current property). The first is that current look-up tables Ψ that are generated and stored in level-2 nodes should be made 2-D, where the key into the table entries is a pair of values: input voltage of the second-level node and the candidate VRM for that node. The second change occurs when level-1 nodes are traversed. In this case, for each candidate set of a level-1 node, all candidate VRM's in its fanouts should also be enumerated in order to find the best assignment of VRM's. The RMTO-VN algorithm is the same as the RMTO-VM problem except that it calls RMTO-FN.

7.4.4.2 Effect of the Current Profile of the Loads

Current profiles of the loads play a key role in the design of an efficient VRM tree. To motivate the need for considering the load profile of the FB's, consider the following example. Assume that to provide a FB with a desired voltage level, a buck converter is needed and the only candidate converters are those shown in Figure 7.10. Now, if the load profile of the FB is $\{(200mA, 90\%), (100mA, 10\%)\}$, i.e., in 90% of the time the FB consume 200mA and in 10% it consumes 100mA current, then using the VRM (b) is more efficient whereas with a load profile of $\{(200mA, 10\%), (100mA, 90\%)\}$ VRM (a) is a better choice.

In the following, we describe how to account for the effect of load profiles in the RMTO-FM algorithm. To begin with, for simplicity, we assume that the profiles of different FB's are independent of one another. In the next section, we show how to account for the correlations among load profiles.

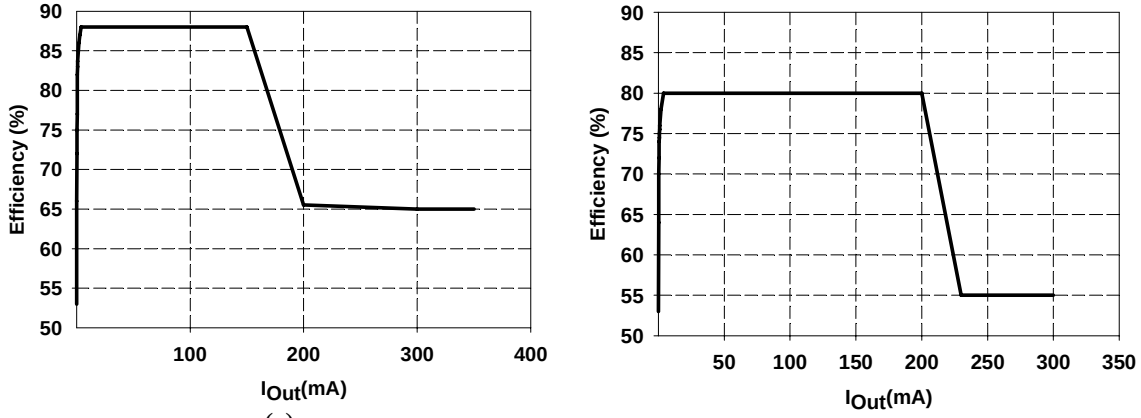


Figure 7.10: The efficiency curves of two commercial buck VRM (TPS60502 [128] and TPS60503 [129]).

Assume that m FB's, f_1, f_2, …, f_m, with the same required voltage level V are connected to a node n. The current profiles of the FB's are expressed as $\{(I_i^j, \alpha_i^j)\}$ where I_i^j and α_i^j are the current demand and the probability of f_i being in its j^{th} state. Notice that for every i, $\sum_{j \in S(i)} \alpha_i^j = 1$, where $S(i)$ is the set of states of the load profile of f_i. When calculating the efficiency and input current of a candidate regulator c_n for n, $i_{out}(n)$ becomes a piecewise-linear function; so, instead of having a constant value for the efficiency and input current of node n, we need to model both of them as piecewise-linear functions. That is,

$$\eta^{k_1,k_2,\ldots,k_m}(c_n, v_{in}(n)) = \mu_r\left(v_{in}(n), I_1^{k_1} + I_2^{k_2} + \ldots + I_m^{k_m}\right)$$

$$i_{in}^{k_1,k_2,\ldots,k_m}(c_n, v_{in}(n)) = \frac{v_{out}(n) \times \left(I_1^{k_1} + I_2^{k_2} + \ldots + I_m^{k_m}\right)}{v_{in} \times \eta^{k_1,k_2,\ldots,k_m}(c_n, v_{in}(n))} \tag{7.11}$$

$$\Pr(S(k_1,k_2,\ldots,k_m)) = \alpha_1^{k_1}\alpha_2^{k_2}\ldots\alpha_m^{k_m}, \text{ for } k_i \in S(i), 1 \le i \le m$$

where $\eta^{k_1,k_2,\ldots,k_m}$ and $i_{in}^{k_1,k_2,\ldots,k_m}$ are the efficiency and input current when f_i is in

state k_i and $\Pr(S(k_1,k_2,...,k_m))$ is the probability of such an event. Notice that the number of states in node n is the product of the number of states in its fanout nodes. An example of generating the piecewise linear input current for the fanin node is shown in Figure 7.11. In this figure it is assumed that the VRM shown in Figure 7.10(a) has been used and $V_{out}/V_{in} = 0.5$.

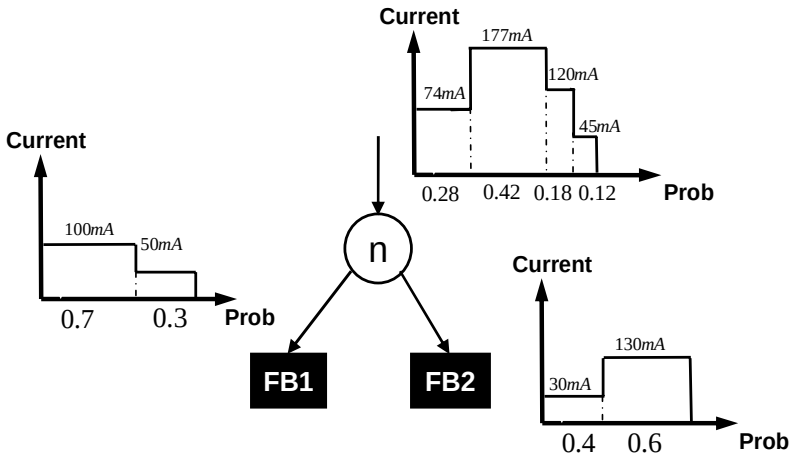


Figure 7.11: Piecewise-linear modeling of the input current of a VRM

The average input current of node n, which is used in optimization, can be obtained from

$$i_{in}^{avg} = \sum_{k_i \in S(i), 1 \le i \le m} i_{in}^{k_1,k_2,...,k_m}(c_n, v_{in}(n)) \times \Pr(S(k_1,k_2,...,k_m)). \tag{7.12}$$

The candidate VRM c_n at node n should satisfy the constraint that,

$$\iota_{c_n,out}^{\max} \ge \max_{k_i \in S(i), 1 \le i \le m} \left(I_1^{k_1} + I_2^{k_2} + ... + I_m^{k_2} \right). \tag{7.13}$$

7.4.4.3 Effect of Correlations among Current Profiles

The correlation between the load profiles of FB's could be used to design a more efficient VRM tree. To motivate the problem, consider two corner case examples. In

the first case, the load currents of the FB's are *positively correlated* in the sense that both FB's have the same peak and off-peak load intervals. An example of such a case is two processor cores that work in parallel. In this case both processors achieve their minimum and maximum currents at the same intervals (c.f. Figure 7.12(a)). On the other hand, in some cases, the load profiles of the FB's are *negatively correlated*, i.e., when one FB is in its low-load state, the other one is in the high-load state and vice versa (c.f. Figure 7.12(b)). An instance of such a scenario occurs by using activity migration technique for dynamic thermal management in which the peak junction temperature is controlled by moving computation between multiple replicated units [55].

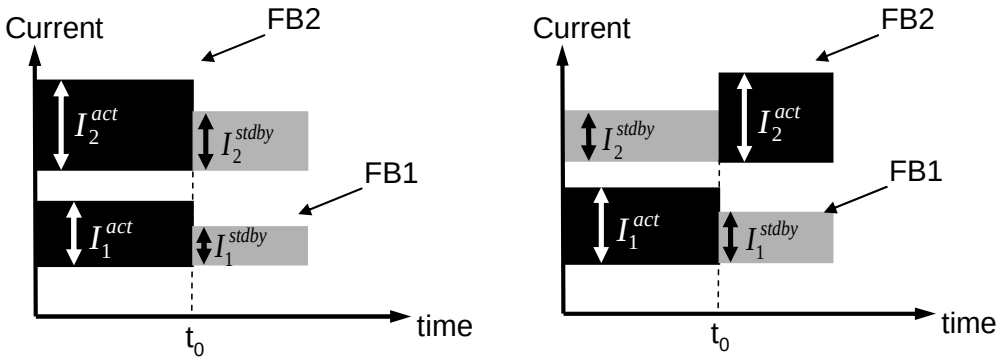


Figure 7.12: (a) Positively correlated FB's (b) negatively correlated FB's.

It is clear that these two scenarios put different constraints on the VRM tree design. For example, when two FB's are negatively correlated, it is more likely that by sharing a single VRM for both of them, a more power-efficient VRM network can be achieved. Rather minor changes need to be made to the RMTO-FM algorithm so that it can handle the effect of load profile correlations. These changes are similar to those that have been discussed in the previous section; so, we do not provide their

details here.

7.5 Experimental Results

The algorithms proposed in this chapter have been implemented in C++ and evaluated on a set of test-benches. A set of thirty DC-DC commercial regulators from Texas Instruments and National Semiconductors were used to create a library of VRM's. This set consists of ten variants of buck, boost, and LDO regulators. Two of these regulators are those shown in Figure 7.10. The power conversion efficiency of each VRM was modeled as a piecewise-linear function of input voltage and output current based on the data sheets for the VRM.

We compared the results of our RMTO-VM with the results of the optimal VRM assignment in a star topology. Table 7.3 summarizes the specifications of our benchmarks along with the reduction of power loss in the VRM tree achieved by applying our algorithm. In Table 7.3, TB is the name of the testbench, V_P is the voltage of power supply, N is the number of FB's in the problem statement, $I_{\min}$ and $I_{\max}$ denote the minimum and maximum required current by any FB's, while $V_{\min}$ and $V_{\max}$ are the minimum and maximum required voltages of any FB's. To illustrate how RMTO-VM algorithm selects the topology of the VRM tree, we depict the VRM tree for the first test-bench in Figure 7.13. In this figure, VRM1 and VRM5 are two LDO's used at the output of two buck regulators to decrease their voltage levels.

Table 7.3: Simulation results for a few test cases

TB	V_P	N	$I_{\min}$	$I_{\max}$	$V_{\min}$	$V_{\max}$	**VRM Tree Power Loss Reduction (%)**
1	2.5	6	50m	100m	1.1	1.8	21.9
2	2.5	7	50m	200m	1.3	1.8	23.6
3	2.5	5	60m	200m	1.3	3.0	14.5
4	2.5	8	50m	200m	1.3	3.3	9.4
5	3.3	6	30m	100m	1.2	1.8	14.6
6	3.3	10	50m	300m	1.1	2.7	21.5
7	3.3	12	30m	350m	1.1	3.0	12.3
8	3.3	8	50m	200m	1.3	3.3	17.9

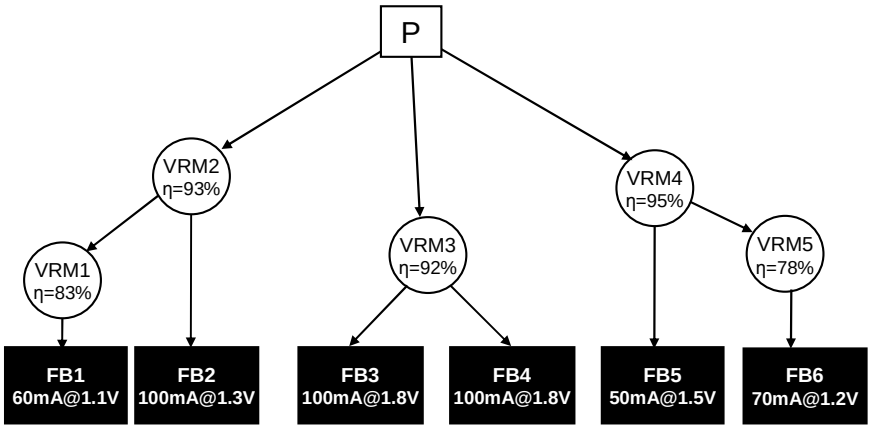


Figure 7.13 : VRM tree topology for TB1.

7.6 Summary

In this chapter we showed that by using a tree topology of suitably chosen voltage regulators between the power source and loads, one can achieve higher power efficiency in the power delivery network. We formulated the problem of optimizing the VRM tree as a dynamic program and solved it efficiently. The experimental results demonstrate the efficacy of proposed problem formulation and solution. Experimental results showed that by using the proposed technique, the power loss in the VRM tree can be reduced by an average of 17%.

Chapter 8
Design of an Efficient Power Delivery Network to Enable Dynamic Power Management

8.1 Introduction

In order to minimize the overall power dissipation of the system while meeting the performance constraints, multiple voltage domains (also known as voltage islands [76]) and dynamic voltage scaling (DVS) [25, 93] are being introduced on the SoC. This means that it is possible to have multiple relatively-small logic blocks operated at different and dynamically changing voltages based on workload monitoring. In these systems, it is required that the PDN delivers power at appropriate voltage levels to different functional blocks (FB's) while incurring the minimum power loss when the voltage level of a FB is changed in response to a change in its workload; therefore, efficient low-voltage DC-DC conversion is a key enabler for the design of any DVS technique. In Chapter 7 we proposed a dynamic programming technique to address the problem of optimal selection of VRM's in a power delivery network [15]; however we did not address the question of VRM selection to enable dynamic voltage scaling.

In this chapter we show how to design an efficient power delivery network for a complex SoC so as to enable DVS through assignment of appropriate voltage level

(and the corresponding clock frequency) to each function block in the SoC.

The rest of this chapter is organized as follows. Section 8.2 provides some background on dynamic voltage scaling and its PDN requirements. In Section 8.3 we propose an efficient power delivery network to enable DVS while incurring minimum power loss in the system. Simulation results are given in Section 8.4, while Section 8.5 summarizes the chapter.

8.2 Background

Dynamic power management (DPM) is a feature of the run-time environment of a system that dynamically reconfigures itself to provide the requested services and performance levels with a minimum activity level on its FB's. The fundamental principle for the applicability of DPM is that systems (and their FB's) experience non-uniform workloads during operation time. Such an assumption is valid for most systems, both when considered in isolation and when inter-networked. A second assumption of DPM is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload [64]. DPM is usually performed through assignment of appropriate voltage levels and corresponding clock frequencies to different FB's of the system. This is also known as dynamic voltage scaling (DVS). In a SoC with DVS option, an on-chip power manager decides when to switch the *SoC power-performance state* (*PPS*), where each PPS corresponds to a particular combination of voltage level (and associated clock frequency) assignments to various FB's in the SoC.

The PDN of a DVS-enabled SoC is required to deliver power at appropriate

voltage levels to different functional FB's while incurring the minimum power loss in the PDN. In the conventional technique to support DVS for different FB's, which is depicted in Figure 8.1, each FB has its own VRM with multiple output voltage levels [25, 93]. The power manager selects the supply level that VRM_i provides to the FB_i.

This architecture, despite its simplicity, has several shortcomings: i) the number of VRM's used in the PDN is equal to the number of FB's i.e., when the number of FB's that can accept multiple voltage levels becomes large, the number of VRM's increases, which in turn increases the chip area and cost, ii) design of variable output voltage VRM is quite challenging and its cost is correspondingly higher than that of a fixed output voltage VRM, iii) unlike the VRM's with fixed-V_{out} where the power conversion efficiency is highly optimized for a specific output voltage level, the power conversion efficiency of the multiple-V_{out} VRM varies as a function of the chosen V_{out} and may sometimes degrade severely from one V_{out} to next [122].

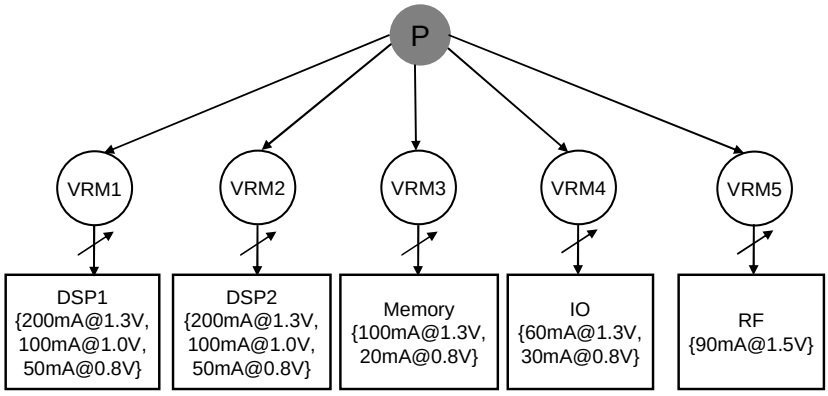


Figure 8.1: The role of VRM tree in providing appropriate voltage level for each FB. The output voltage of each VRM is changed dynamically.

Based on these observations, in the next section we propose a new technique to address the problem of PDN design to support dynamic voltage scaling.

8.3 Power Efficient PDN to enable DVS

In our technique, which is depicted in Figure 8.2, the PDN is composed of two layers. In the first layer of PDN, which is called the *power conversion network* (PCN), VRM's are used to generate all voltage levels that may be needed by different FB's in the SoC design. This is accomplished by using fixed-V_{out} VRM's; so, if $\mathcal{U}$ is the set of all voltage levels required by any FB's, then there must be at least $|\mathcal{U}|$ VRM's in the PCN. Usually this number is small since many of the FB's share the same set of allowed voltage levels. In the second layer of PDN, a *power switch network* (PSN) is used to dynamically connect the power supply terminals of each FB to the appropriate VRM output in the PCN.

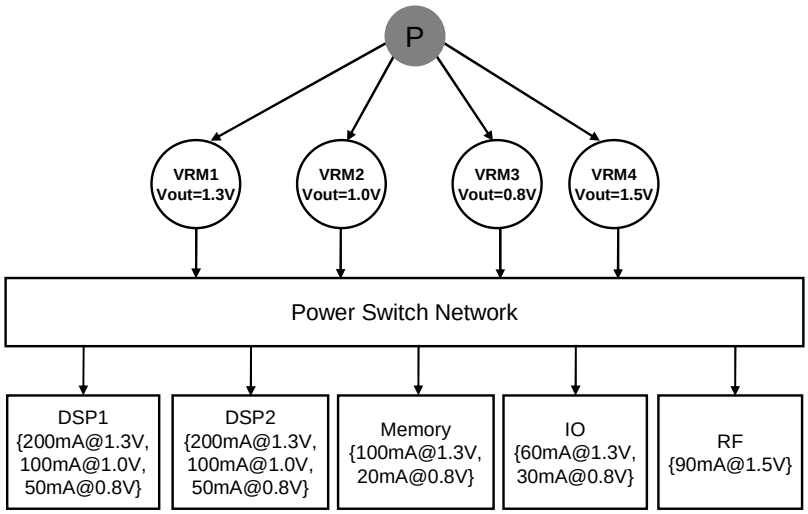


Figure 8.2: The proposed architecture of PDN to support dynamic voltage scaling. The output voltage of each VRM is fixed.

In our system modeling framework, it is assumed that the transition of the system into different PPS's can be described as a time-homogenous Markov chain, and hence, PPS transitions can be captured by a stationary time-independent transition matrix $[p_{ij}]$ (c.f., Figure 8.3). In each state of this Markov chain, the supply voltage

level of all FB's is specified. Clearly, no two states will have the same supply voltage assignments. Let π_i denote the probability of being in state i of this Markov chain. In vector $\pi = [\pi_i]$ entries π_i sum to one and satisfy

$$\pi_i = \sum_{j \in S} \pi_j p_{ji} \,. \tag{8.1}$$

Additionally, for simplicity, in this section it is assumed that the current demands of every FB when it is working with each of its voltage levels is specified and is constant. In the next section it will be shown how to change the problem formulation to handle the general case when the current demands of FB's follow some probability distribution function around a mean value. Moreover, it is assumed that level shifters have been included in the SoC to enable communication among FB's operating on different supply voltages. Now, the question becomes how to design the PCN to achieve minimum power loss in the power distribution network, and how to design the PSN to make sure that all FB's receive the desired supply voltage levels.

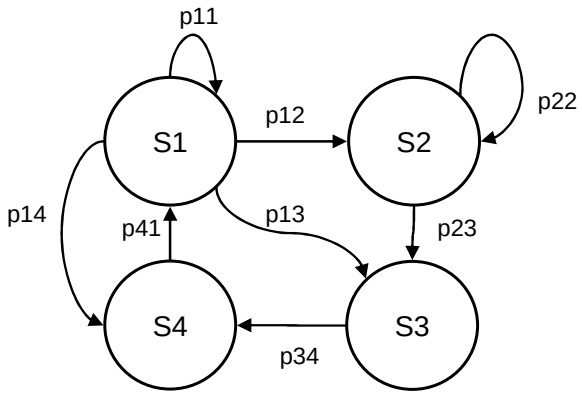


Figure 8.3: Operating states and state transition of a system.

8.3.1 Power Conversion Network Optimization

The PCN optimization supporting dynamic voltage scaling (PCODS) problem is

defined next.

PCODS problem: Given is

- A library $\boldsymbol{\mathcal{R}}$ of VRM's and for each $r \in \boldsymbol{\mathcal{R}}$, its cost c_r, output voltage $\upsilon_{r,out}$, the minimum and maximum input voltages $\upsilon_{r,in}^{\min}$ and $\upsilon_{r,in}^{\max}$, the maximum load current $\iota_{r,out}^{\max}$, and the VRM's power conversion efficiency η_r as a function of the load current and input voltage,
- A power source P, with the nominal voltage of V_P,
- A set $\boldsymbol{\mathcal{F}}$ of FB's, and for each $f \in \boldsymbol{\mathcal{F}}$, the required voltages and the corresponding current demands,
- A Markov chain model $\boldsymbol{S}$ of the system, where in each state of the Markov chain the supply voltage level of each FB is specified.

The objective is to build a network of VRM's that connects P to all FB's and minimizes a weighted sum of total power consumption and total cost of the VRM's used in the PCN, i.e., $V_P I_P + \lambda \sum_{r \in PCN} c_r$, while meeting the voltage and current constraints.

In PCODS problem, λ is a parameter which defines the tradeoff between power-efficiency and cost of the PCN. For example, if $\lambda = 0$, then PCODS optimizes the power efficiency while $\lambda = \infty$ results in the lowest-cost PCN.

Before giving details of how PCODS can be solved, in Table 8.1 we define the notation used in the remainder of the chapter. Some of these notations are similar to those in Chapter 7 but for convenience they are repeated here.

We assume that if a FB requires the same voltage V in two different states, it is always powered up by an identical VRM. This assumption implies that the number of power switches in PSN to deliver power to FB $f \in \boldsymbol{\mathcal{F}}$ is exactly $\left|\boldsymbol{\mathcal{V}}_f\right|$ and hence

reduces not only the complexity of PSN, but also the power loss of PSN during PPS transitions.

Table 8.1: Notation used in RMTO algorithm

$\mathcal{R}$	Set of all VRM's, r
$\mathcal{F}$	Set of all FB's, f
$\mathcal{S}$	Set of all states of the Markov chain model of the system
$\mathcal{V}_f$	Set of required voltage levels by FB $f \in \mathcal{F}$
$\mathcal{U}$	Set of voltage levels required by all FB's; i.e., $\mathcal{U} = \bigcup_{f \in \mathcal{F}} \mathcal{V}_f = \{V_1, V_2, ..., V_m\}$
$V_{f,s}$	Required voltage of FB $f \in \mathcal{F}$ in state $s \in \mathcal{S}$
$I_{f,s}$	Required current of FB $f \in \mathcal{F}$ in state $s \in \mathcal{S}$
$I_{f,v}$	Required current of FB $f \in \mathcal{F}$ when its required voltage level is $v \in \mathcal{V}_f$ $(I_{f,v} = I_{f,s} : V_{f,s} = v)$
$\eta_r(V, I)$	Power conversion efficiency of regulator $r \in \mathcal{R}$, with the input voltage V and output current I
$I_{r,s}^{in}$	Input current of regulator *r* in state $s \in \mathcal{S}$
$I_{avg,r}$	Average input current of regulator r over all states

It should be noted that the power delivered to the FB's is independent of the topology of PCN and can be calculated as,

$$P_{FBs} = \sum_{f \in \mathcal{F}} \sum_{s \in \mathcal{S}} \pi_s V_{f,s} I_{f,s} . \tag{8.2}$$

Definition 8.1: For each voltage level $V_i \in \mathcal{U}$, the corresponding *voltage domain* $\mathcal{D}_i$ is defined as the set of all FB's that require V_i in some state, i.e.,

$$\mathcal{D}_i = \left\{ f \in \mathcal{F} : V_i \in \mathcal{V}_f \right\}. \tag{8.3}$$

Since each FB may have more than one voltage level, $\mathcal{D}_i$'s may be overlapping. For each voltage level $V_i \in \mathcal{U}$, one of more VRM's should be used to deliver power to the corresponding voltage domain $\mathcal{D}_i$. Assume that the topology of the VRM tree

delivering power to $\mathcal{D}_i$ is known. In this case, when the system is in state s, the output current of a VRM r with output voltage V_i that delivers power to a subset $\mathcal{D}_i^j \subseteq \mathcal{D}_i$ can be computed as,

$$I_{r,s}^{out} = \sum\nolimits_{f \in \mathcal{D}_i^j, V_{f,s}=V_i} I_{f,s} \,. \tag{8.4}$$

Therefore, the input current of VRM r in state s is obtained as,

$$I_{r,s}^{in} = \frac{V_i \times I_{r,s}^{out}}{V_P \times \eta_r \left(V_P, I_{r,s}^{out}\right)} \tag{8.5}$$

and the average input current of r which is drawn from the power supply is,

$$I_{avg,r} = \sum\nolimits_{s \in \mathcal{S}} \pi_s I_{r,s}^{in} \,. \tag{8.6}$$

The average current drawn from the power supply by voltage domain $\mathcal{D}_i$ is then computed as,

$$I_{avg}(\mathcal{D}_i) = \sum\nolimits_{r \in \mathcal{R}_i} I_{avg,r}^{in} \tag{8.7}$$

where $\mathcal{R}_i$ is the set of all VRM's used to power up $\mathcal{D}_i$. The total cost of the VRM's used in this topology to deliver power to $\mathcal{D}_i$ is,

$$C_{\mathcal{D}_i} = \sum\nolimits_{r \in \mathcal{R}_i} c_r \,. \tag{8.8}$$

Therefore, the average current drawn from the power supply by this PCN and the total cost of VRM's in the PCN can be written as,

$$I_{avg} = \sum\nolimits_i I_{avg}(\mathcal{D}_i) \tag{8.9}$$

$$C_{PCN} = \sum\nolimits_i C_{\mathcal{D}_i} \,. \tag{8.10}$$

To deliver power to FB's in each $\mathcal{D}_i$, different options are available (c.f., Figure 8.4 for a pictorial elaboration). In the first option, which is the lowest-cost one, only one VRM is used to deliver power to all FB's in each $\mathcal{D}_i$. The other option is to use one VRM per FB. The drawback of this solution is that the number of VRM's increases with the number of FB's. Because of the non-monotone dependency of power conversion efficiency on the delivered output current, neither solution may be the best from the power-efficiency point of view and a design in between the two extremes may be the best one. Furthermore, because objective function in the general formulation of the PCODS problem is a weighted sum of the power consumption and the cost of the PCN, by enumerating other solutions a better tradeoff between power-efficiency and cost may be achieved. Therefore, all possible "set partitions" of $\mathcal{D}_i$ (as defined by Definition 7.4) should be enumerated when searching for the optimal VRM assignment to $\mathcal{D}_i$.

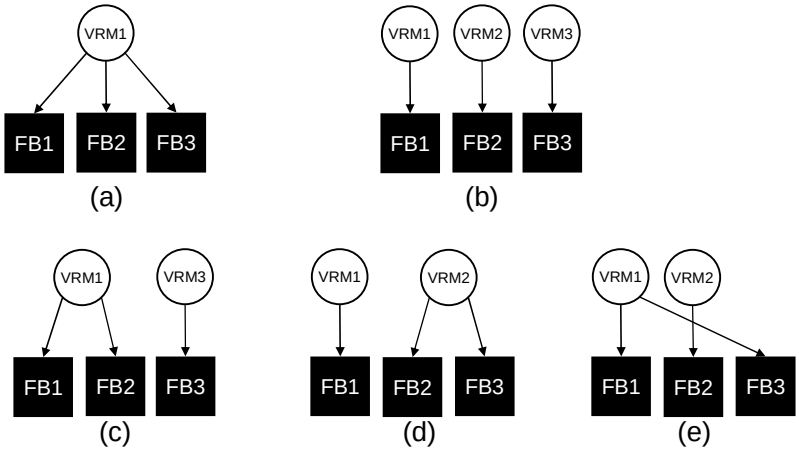


Figure 8.4: Different options for delivering power to three FB's which require the same voltage at some states. The output voltages of all VRM's are the same.

Definition 8.2: In a partition of $\mathcal{D}_i$ the *required voltage* of each part is V_i. The

current demand of a part in a state is the summation of the current demands of all FB's in that part in the specified state.

Definition 8.3: A *valid VRM assignment* to a partition of $\mathcal{D}_i$ is the assignment of one VRM to each part such that the constraints of each VRM are satisfied, i.e., for each VRM r the input voltage of VRM is between $\upsilon_{r,in}^{\min}$ and $\upsilon_{r,in}^{\max}$, the required voltage of the part is $\upsilon_{r,out}$, and the maximum current demand of the part over all states is lower than $\iota_{r,out}^{\max}$.

Definition 8.4: An *optimum VRM assignment* to a partition of $\mathcal{D}_i$ such as $\{\mathcal{D}_i^1,...,\mathcal{D}_i^n\}$ is a valid VRM assignment which minimizes $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$, where $I_{avg,j}$ and c_j are the input current and associated cost of designated VRM to part $\mathcal{D}_i^j$, respectively.

Theorem 8.1: A valid VRM assignment to a partition of $\mathcal{D}_i$ is optimum, if and only if in each of its parts such as $\mathcal{D}_i^j, V_P I_{avg,j} + \lambda c_j$ is minimized.

Proof: Assume $\mathcal{D}_i$ is partitioned into n nonempty subsets such as $\{\mathcal{D}_i^1,...,\mathcal{D}_i^n\}$. Each valid VRM assignments to a part is shown as a pair of input current of the corresponding VRM and its associated cost, i.e., (I_{avg}, c). The set of all valid VRM assignments to part $\mathcal{D}_i^j$ is shown as $\mathcal{Z}_j = \{(I_{avg}, c)\}$. Optimum VRM assignment to partition $\mathcal{D}_i$ is the selection of one tuple $(I_{avg,j}, c_j)$ from each $\mathcal{Z}_j$ such that $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$ is minimized. It can be seen that $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$ is minimized if and only if for each tuple $(I_{avg,j}, c_j)$, the value of $V_P I_{avg,j} + \lambda c_j$ is

minimum over all tuples in $\mathcal{Z}_j$. ■

The result of Theorem 1 is that to find the optimum VRM assignment to set $\mathcal{D}_i$, all partitions of $\mathcal{D}_i$ should be enumerated. In each partition, the best VRM r that satisfies the constraints and minimizes $V_P I_{avg} + \lambda c$ for every part is found. The partition that results in the minimum $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$ is the optimum one.

Based on the above discussion, Figure 8.5 shows $optPCN$ algorithm to solve PCODS problem. Basically it starts by constructing $\mathcal{D}_i$ sets and for each $\mathcal{D}_i$ it finds the best VRM assignment by using Theorem 8.1.

```
optPCN(R, F, S, V_P){
  For each V_i ∈ U = {V_1,...,V_m}{
    D_i = { f ∈ F : V_i ∈ V_f };
    ψ(V_i) = sub − optPCN(R, F, S, V_P, V_i, D_i);
  }
}
═══════════════════════════════════════════
sub−optPCN(R, F, S, V_P, D_i) {
  optCost = ∞;
  optVRMs = {};
  For each non−empty partition of D_i such as {D_i^1,...,D_i^n}
    For each D_i^j, 1 ≤ j ≤ n{
      Select best VRM r that minimizes V_P I_avg,r + λc_r;
      cost_j = V_P I_avg,r + λc_r;
      VRM_j = r;
    }
    newCost = Σ_j cost_j;
    If (newCost < optCost)
      optCost = newCost;
      optVRMs = { VRM_j };
  }
  Return (optCost, optVRMs);
}
```

Figure 8.5: The optPCN algorithm for solving PCODS.

Theorem 8.2: The $optPCN$ algorithm described in Figure 8.5 finds the optimum solution to the PCODS problem.

Proof: The optimality of $optPCN$ algorithm is immediate from Theorem 8.1 and the fact that for each $\mathcal{D}_i$, all partitions are enumerated. ■

Theorem 8.3: The worst-case running time of $optPCN$ algorithm is $O\left(|\mathcal{R}||\mathcal{S}||\mathcal{F}|B_{|\mathcal{F}|}\right)$, where $|\mathcal{R}|$, $|\mathcal{S}|$, and $|\mathcal{F}|$ are the cardinalities of corresponding sets.

Proof: For each of voltage levels required by FB's such as $V_i \in |\mathcal{U}|$, we need to partition the corresponding voltage domain $\mathcal{D}_i$ into non-empty subsets and for each partition find the best set of VRM's. The number of these partitions is $B_{|\mathcal{D}_i|}$. For any of such partitioning into k-subsets, we need to find the best set of k VRM's for the parts of the partition. The best VRM for a part is found by enumerating all VRM's with output voltage V_i. The number of these VRM's is denoted as R_i. For each such VRM, we need to calculate the average input current of the VRM among all states of Markov chain; so, the numbers of operations to find the best set of k VRM's for a partition is $|\mathcal{S}||\mathcal{F}|R_i$. Therefore, the number of operations to find the best set of VRM's for V_i is $|\mathcal{S}||\mathcal{F}|R_iB_{|\mathcal{D}_i|}$ and the complexity of the whole procedure is

$$\sum\nolimits_i |\mathcal{S}||\mathcal{F}|R_iB_{|\mathcal{D}_i|}. \tag{8.11}$$

For each i, $\mathcal{D}_i \subseteq \mathcal{F}$; therefore $|\mathcal{D}_i| \le |\mathcal{F}|$ and it follows that

$$\sum\nolimits_i |\mathcal{S}||\mathcal{F}|R_iB_{|\mathcal{D}_i|} \le |\mathcal{S}||\mathcal{F}|B_{|\mathcal{F}|}\sum\nolimits_i R_i. \tag{8.12}$$

In (8.14) equality holds if and only if for each i, $\mathcal{D}_i = \mathcal{F}$. This condition requires that for each $f \in \mathcal{F}$, the set of required voltage levels is equal to $\mathcal{U}$, i.e., $\mathcal{V}_f = \mathcal{U}$. On

the other hand, since each VRM in the library can provide only one output voltage, $\sum_i R_i \leq |\boldsymbol{\mathcal{R}}|$. Therefore, the worst case running time of the algorithm is $O\left(|\boldsymbol{\mathcal{R}}||\boldsymbol{\mathcal{S}}||\boldsymbol{\mathcal{F}}|B_{|\boldsymbol{\mathcal{F}}|}\right)$. ■

From Theorem 8.3, one can see that $optPCN$ algorithm has exponential complexity in the number of FB's; however, since the number of FB's is small, in practice the runtime of the algorithm is quite reasonable.

8.3.1.1 Effect of non-constant current

In the formulation of PCODS problem, it is assumed that the current demand of each FB is a constant value independent of the system PPS. In this section it is shown how to modify the problem formulation to handle the case when the current demands of various FB's follow some probability density function (pdf).

We assume the current demands of different FB's can be modeled as independent Gaussian distribution functions (the case that the demands follow some other probability distribution function can be addressed in a similar manner). In this case, because the output current of a VRM which is connected to a number of FB's is a sum of independent Gaussian random variables (c.f., Equation (8.4)), it will also be a Gaussian random variable, whose mean and variance respectively are the sum of means and sum of variances of the current demand distributions in the corresponding FB's. This continuous-time random variable is approximated with a discrete-time random variable function which has the probability $\Pr(j)$ in interval $[I_{\min}+j\times\Delta I,\ I_{\min}+(j+1)\times\Delta I)$ (for $0\leq j<(I_{\max}-I_{\min})/\Delta I$) as shown in Figure 8.6.

Since the efficiency of the VRM and hence its input current are functions of the

output current, Equation (8.5) should be modified to account for this dependency,

$$I_{r,s}^{in} = \sum_{j=0}^{L} \Pr(j) \frac{V_i \times (I_{\min} + j \times \Delta I)}{V_P \times \eta_r (V_P, I_{\min} + j \times \Delta I)} \tag{8.13}$$

where $L = (I_{\max} - I_{\min}) / \Delta I - 1$. Selecting a smaller value for ΔI results in a better approximation for input current of the VRM, but also increases the algorithm runtime.

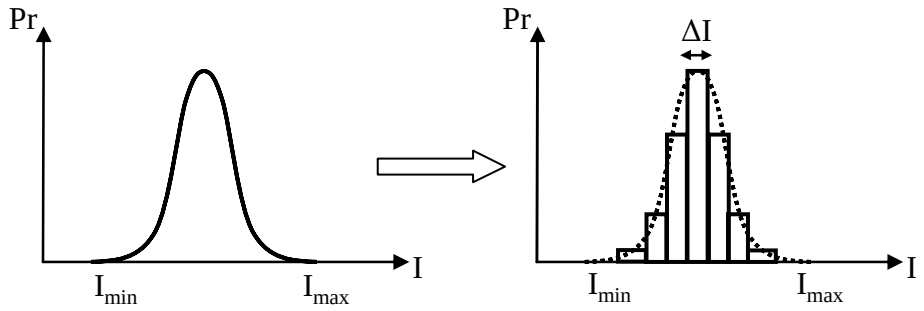


Figure 8.6: Approximating the continuous distribution with a discrete one.

8.3.2 Power Switch Network Optimization

Power switch network (PSN) performs the function of switching the supply voltage level of the FB's when a new PPS is commanded by the power manager. Figure 8.7 depicts a PSN for delivering three different voltage levels to a FB. The switches in the PSN are controlled by a *power switch controller* (PSC) which is zero-hot coded, i.e., at any given time only one of its outputs is zero, and hence, only one PMOS transistors in ON.

The number of PMOS transistors needed for each FB f in the PSN is $|\boldsymbol{\mathcal{V}}_f|$. The PMOS transistor which is required to deliver voltage level $v \in \boldsymbol{\mathcal{V}}_f$ to an $f \in \boldsymbol{\mathcal{F}}$ and its width are respectively denoted as $M_{f,v}$ and $W_{f,v}$. This PMOS transistor should be large enough so that the voltage-drop between its drain and source does not

exceed a tolerable value.

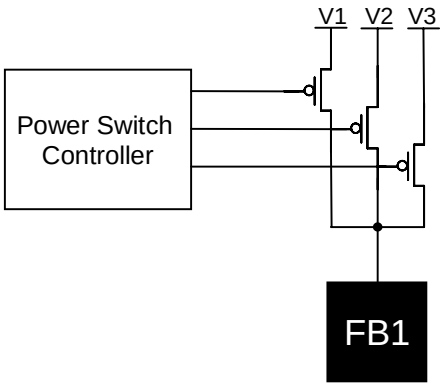


Figure 8.7: A PSN for delivering three different voltage levels to a FB.

In the steady state, when FB f is supplied with $v \in \boldsymbol{\mathcal{V}}_f$, the current that flows through the ON PMOS transistor $M_{f,v}$ is the current demand of f at voltage v, i.e., $I_{f,v}$. Since this transistor is in triode region, its current can be derived from the alpha-power model [107] as,

$$I_{f,v} = I_{M_{f,v}} = k\frac{W_{f,v}}{L_{eff}}\left(\frac{V_{gs}-V_t}{v-V_t}\right)^{\alpha/2} V_{ds} \tag{8.14}$$

where L_{eff} is the effective length of the transistor, V_{gs}, V_{ds}, and V_t are the gate-to-source, drain-to-source, and threshold voltage of the transistor, respectively. Note that k and α are technology. Now, if the maximum tolerable voltage-drop at the supply of the FB is ΔV, the minimum required width for $W_{f,v}$ will be computed as,

$$W_{f,v}^{\min} = \frac{I_{f,v}L_{eff}}{k\Delta V} \tag{8.15}$$

8.3.2.1 PSN Power Consumption

When the state of the system changes from PPS i to j, some energy is consumed to

turn ON/OFF some of the PMOS switches. Assume that the power manager changes the state of the system at regular time intervals with a frequency of f_{PM}. If C_{PMOS} is the total capacitance which is charged or discharged during this transition, then the power consumption for this transition is calculated from

$$P_{dyn,i\to j} = p_{i\to j} V_{dd}^2 f_{PM} C_{PMOS} \tag{8.16}$$

where $p_{i\to j}$ denotes the transition probability from PPS i to j which can be computed as,

$$p_{i\to j} = \pi_i p_{ij} \tag{8.17}$$

So, the power consumption of the PMOS switches is calculated as

$$P_{overhead} = \sum_{i,j}\left(\frac{1}{2} p_{i\to j} V_{dd}^2 f_{PM}\left(\sum_{f:V_{f,i}\neq V_{f,j}}\left(C_{f,V_{f,i}} + C_{f,V_{f,j}}\right)\right)\right) \tag{8.18}$$

where $C_{f,v}$ is the input capacitance of $M_{f,v}$, i.e., $C_{f,v} = W_{f,v} L C_{ox}$.

Equation (8.18) is the power consumption overhead of our solution compared to the conventional one, where one multiple-output VRM is used for each FB to provide it with appropriate voltage levels.

8.4 Simulation Results

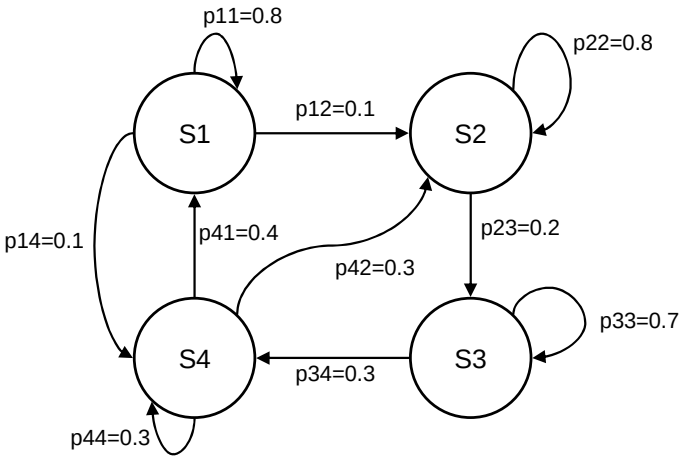
The algorithms described earlier in this chapter have been implemented in C++ and evaluated on a set of test-benches. A collection of thirty DC-DC commercially available regulators from Texas Instruments and National Semiconductors were chosen to create the library of VRM's. The power conversion efficiency of each VRM was modeled as a piecewise-linear function of input voltage and output current based on the data sheets for the VRM. The cost of each VRM was assumed to be its

dollar cost for a 1000-unit purchase. Note that we did not have access to the efficiency curves and cost of the unpackaged DC-DC converters.

We performed two experiments to compare the performance of the proposed technique with the conventional VRM assignment to support dynamic voltage scaling in a system. In the first experiment, we used $optPCN$ algorithm with $\lambda = 0$ to find the most power-efficient PCN based on our solution. The best multiple-output VRM assignment to minimize the power consumption of the system based on the conventional solution was also generated for comparison purposes. The results of this experiment are reported in Table 8.2, where the first column gives the name of the test-bench (Details of the first test-bench are provided in Figure 8.8.), the second column gives the number of FB's in the problem, and the third column gives the number of states in the Markov chain model of the system. Column 4 and 5 show PDN power loss and cost reduction in the proposed solution compared to those of the conventional solution (power loss in the PDN is the difference between the power delivered to FB's and the power drawn from the power source *P*). Finally, the last column shows the runtime of $optPCN$ algorithm for finding the optimal set of VRM in the PCN.

Table 8.2: Power and cost reduction of PDN in the proposed technique compared to those of the conventional technique

TB	$\lvert\mathcal{F}\rvert$	$\lvert S\rvert$	**PDN Power Reduction (%)**	**PDN Cost Reduction (%)**	**Runtime (sec)**
C1	5	4	38.5	1.1	<1
C2	6	4	40.4	5.0	<1
C3	8	5	34.2	–2.8	<1
C4	10	10	30.1	29.7	13
C5	12	10	27.9	8.1	70



S1: {V_{DSP1}=1.3, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5}
S2: {V_{DSP1}=1.0, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5}
S3: {V_{DSP1}=0.8, V_{DSP2}=1.0, V_{MEM}=1.3, V_{IO}=0.8, V_{RF}=1.5}
S4: {V_{DSP1}=0.8, V_{DSP2}=0.8, V_{MEM}=0.8, V_{IO}=0.8, V_{RF}=1.5}

Figure 8.8: Test-bench TB1. The current demands of FB's are similar to those in Figure 8.1.

From Table 8.2, one can see that the proposed technique reduces the power loss of PDN by an average of 34%. Additionally, in most cases it also reduces the PDN cost. The average PDN cost reduction is 8%. Finally, from Table 8.2 one can see that the runtime of $optPCN$ algorithm is quite reasonable.

In the second experiment, we studied the tradeoff between the power-efficiency of the PDN and its cost. More precisely, in addition to designing the optimal PCN for $\lambda = 0$ by running $optPCN$ algorithm, the algorithm was invoked for other values of λ for which the PCN power loss does not increase beyond 10% of its optimal value. The cost reduction of the PDN for this set of test-benches is reported in Table 8.3. It is seen that on average by allowing about 8.6% increase in the PDN power loss, the cost of PDN can be lowered by 47%.

Table 8.3: Trading off power for cost of PDN in the proposed technique

TB	PDN Power Increase (%)	PDN Cost Reduction (%)
C1	10.0	53.0
C2	4.3	46.9
C3	8.9	57.9
C4	9.6	26.1
C5	10.0	52.9

8.5 Summary

In this chapter we presented a new technique to design an efficient power delivery network for systems with dynamic voltage scaling capability. In this technique, the PDN is composed of two layers: PCN and PSN. In PCN, fixed-V_{out} VRM's are used to generate all voltage levels that may be needed by different FB's in the system. PSN is used to dynamically connect the power supply terminals of each FB to the appropriate VRM output in the PCN. We showed that this technique not only reduces the cost of the power conversion network, but also results in a more power-efficient power delivery network. We further described an algorithm to select the best VRM's to achieve a design target in the new PDN. By means of simulation results, it was demonstrated that the proposed technique reduces the power loss of PDN by an average of 34% while reducing its cost by an average of 8%.

Chapter 9
Conclusion

9.1 Summary of Contributions

This thesis has contributed to new circuit, logic, and system techniques to address the problem of low-power design in VLSI circuits.

In Chapter 3 we presented heterogeneous cell SRAM to reduce the active leakage power consumption of on-chip memory arrays. The proposed method was to deploy different configurations of six-transistor SRAM cells corresponding to different threshold voltage and oxide thickness assignments for the transistors. We showed that the proposed solution can achieve up to 40% leakage power reduction without having any hardware or delay overheads. Portions of this work have been published in [7, 10, 13].

In Chapter 4 we described the PG-gated SRAM cell for standby leakage reduction of SRAM's. We showed that our technique not only improves the leakage power consumption of SRAM cells, but also enhances the hold static noise margin and soft error immunity. The PG-gated SRAM cell can also be combined in an orthogonal fashion with the heterogeneous cell SRAM to lower both runtime and standby leakage currents.

In Chapter 5 we addressed the problem of low power fanout optimization. It was

shown that previous techniques proposed to optimize the area of a fanout tree may increase the power consumption of the circuit. We showed how to accurately model the problem of low-power fanout chain design and described how to construct a fanout tree from power-optimal fanout chains. Moreover, we described an efficient method to minimize the total power consumption of a fanout tree by utilizing multi channel length and multi threshold voltage techniques. Simulation results show that the proposed technique can reduce the power consumption of the fanout trees by an average of 11.17% over SIS fanout optimization program. Portions of this work have been published in [8, 11, 12].

In Chapter 6 we investigated the problem of power-optimal repeater insertion for global buses in the presence of crosstalk noise. We utilized MTCMOS technique to reduce the leakage power consumption of the bus in idle mode. We simultaneously calculated the repeater sizes, repeater distances, and the size of the sleep transistors to minimize the power dissipation subject to a delay constraint. Related SPICE simulation showed that by considering the effect of crosstalk on both delay and power consumption, and by using MTCMOS technique, the average power consumption of the bus lines can be reduced by more than 50% with a small delay penalty of 5%. This work has been published in [41].

In Chapter 7 we tackled the problem of optimal selection of voltage regulator modules in a power delivery network. It was demonstrated that using a tree topology of suitably chosen VRM can reduce the power consumption of power delivery network by an average of 17%. We proposed a dynamic programming technique to efficiently select the best set of VRM’s in a power delivery network. This work has

been published in [15].

In Chapter 8 we provided a novel technique to design an efficient power delivery network for a complex SoC so as to enable dynamic power management through assignment of appropriate voltage level to each function block in the SoC. In this technique the PDN is composed of two layers. In the first layer of PDN, fixed-V_{out} VRM's are used to generate all voltage levels that may be needed by different FB's in the SoC design. In the second layer of PDN, a power switch network is used to dynamically connect the power supply terminals of each FB to the appropriate VRM output in the PCN. The related simulations show that the proposed technique reduces the power loss of the power delivery network by an average of 34% while reducing its cost by an average of 8%. This work has been published in [14].

9.2 Future Work

9.2.1 Low-Power SRAM Design

In this section we provide some directions for future research based on the material presented throughout this dissertation.

9.2.1.1 Statistical Optimization of PG-Gated SRAM

One possible extension of the PG-gated SRAM is to minimize the mean of leakage power dissipation under process variations, such as variation in the number of dopants in the channel, variation in the line width, and variation in the oxide thickness. By modeling the subthreshold and tunneling gate leakage current as functions of these variations, it is possible to minimize the mean of leakage subject to a fixed voltage potential between the power supply and ground rail in the standby

mode.

9.2.1.2 FinFET-based HCS and PG-Gated SRAM Design

FinFET-based SRAM [16, 47] has shown to be very effective in reducing the short channel effects. One interesting extension of our research on low-leakage SRAM design is to evaluate the efficacy of heterogeneous cell SRAM and PG-gated SRAM solutions on the FinFET-based designs.

9.2.2 Signal Distribution Network Design

9.2.2.1 Power-optimal Repeater Insertion with Bus Encoding

Several extensions can be developed on top of our buffer insertion technique. Applying a bus encoding technique such as that in [137] to eliminate the worst-case crosstalk between adjacent bus lines and utilizing the resulting extra slack for power reduction is an interesting extension.

9.2.3 Power Delivery Network Design

9.2.3.1 Concurrent VRM Selection and Decap Allocation in a PDN

As stated in Chapter 7, for high-performance microprocessors, VRM's and decaps are essential components of the PDN. Typically VRM selection and decap allocation for the PDN are performed sequentially, one after the other. One interesting future work is to carry out these assignments concurrently to achieve a more power-efficient design. To motivate the need for simultaneous VRM assignment and decap allocation, recall that to have a high efficiency LDO regulator, the quiescent current I_{quies} must be minimized. The problem with reducing the quiescent current is that it increases the transient response time of the LDO. It can be shown that with

increasing the transient response time, a larger decap is required to limit the ripple of the load voltage under load current variation [105]. As asserted in Chapter 7, in sub-100nm technology processes, tunneling gate leakage of a typical decap is in the order of milliamperes [52]. Therefore, the additional leakage current from increasing the size of decap could diminish any benefit of having a small quiescent current and potentially even increase the total power consumption in the pair of VRAM and decap. From this discussion, one can conclude that if VRM assignment and decap allocation are performed independently, the solution to the PDN design may not be power-optimal.

9.2.3.2 Speedup VRM Selection Algorithms

The proposed algorithms to optimally select the best set of VRM's in a VRM tree (Chapter 7) and in a power conversion network (Chapter 8) have exponential complexity in the number of FB's. Usually the number of FB's on a SoC is small, and therefore, in practice the runtime of the algorithms is quite reasonable. But in a very complex SoC with tens of FB's, the runtime of these algorithms tends to be slow. One interesting future direction is to explore new techniques that can achieve near-optimal solutions for these problems in polynomial time in FB count.

BIBLIOGRAPHY

[1] A. Abdollahi, F. Fallah, and M. Pedram, "An effective power mode transition technique in MTCMOS circuits," in *Proc. of Design Automation Conference*, 2005, pp. 37-42.

[2] A. Abdollahi and M. Pedram, "Power minimization techniques at the RT-level and below," in *SoC: Next Generation Electronics*, B. M. Al-Hashimi, Ed. New York, NY: IEE Press, 2005.

[3] E. Acar, R. Arunachalam, and S. R. Nassif, "Predicting short circuit power from timing models," in *Proc. of Asia and South Pacific Design Automation Conference*, 2003, pp. 277-282.

[4] A. Agarwal, C. Kim, S. Mukhopadhyay, and K. Roy, "Leakage in nano-scale technologies: mechanisms, impact and design considerations," in *Proc. of Design Automation Conference*, 2004, pp. 6-11.

[5] A. Agarwal, H. Li, and K. Roy, "DRG-cache: A data retention gated-ground cache for low power," in *Proc. of Design Automation Conference*, 2002, pp. 473- 478.

[6] A. Agarwal and K. Roy, "A noise tolerant cache design to reduce gate and sub-threshold leakage in the nanometer regime," in *Proc. of International Symposium on Low Power Electronics and Design*, 2003, pp. 18-21.

[7] B. Amelifard, F. Fallah, and M. Pedram, "Leakage minimization of SRAM cells in a dual-Vt and dual-Tox technology," to appear in *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2008.

[8] B. Amelifard, F. Fallah, and M. Pedram, "Low-power fanout optimization using multi threshold voltages and multi channel lengths," submitted to *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*.

[9] B. Amelifard, A. Afzali-Kusha, and A. Khademzadeh, "Enhancing the efficiency of cluster voltage scaling for low-power application," in *Proc. of IEEE International Conference on Circuits and Systems*, 2005, pp. 1666-1669.

[10] B. Amelifard, F. Fallah, and M. Pedram, "Low-leakage SRAM design with dual Vt transistors," in *Proc. of International Symposium on Quality Electronic Design*, 2006, pp. 729 -734.

[11] B. Amelifard, F. Fallah, and M. Pedram, "Low-power fanout optimization using MTCMOS and multi-Vt techniques," in *Proc. of International Symposium on Low Power Electronics and Design*, 2006, pp. 334 -337.

[12] B. Amelifard, F. Fallah, and M. Pedram, "Low-power fanout optimization using multiple threshold voltage inverters," in *Proc. of International Symposium on Low Power Electronics and Design*, 2005, pp. 95-98.

[13] B. Amelifard, F. Fallah, and M. Pedram, "Reducing the sub-threshold and gate-tunneling leakage of SRAM cells using dual-Vt and dual-Tox assignment," in *Proc. of Design, Automation and Test in Europe*, 2006, pp. 995 -1000.

[14] B. Amelifard and M. Pedram, "Design of an efficient power delivery network in an SoC to enable dynamic power management," in *Proc. of International Symposium on Low Power Electronics and Design*, 2007, pp. 328-333.

[15] B. Amelifard and M. Pedram, "Optimal selection of voltage regulator modules in a power delivery network," in *Proc. of Design Automation Conference*, 2007, pp. 168-173.

[16] H. Ananthan, A. Bansal, and K. Roy, "FinFET SRAM - device and circuit design considerations," in *Proc. of International Symposium on Quality Electronic Design*, 2004, pp. 511-516.

[17] N. Azizi, F. Najm, and A. Moshovos, "Low-leakage asymmetric-cell SRAM," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 4, Aug. 2003, pp. 701-715.

[18] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Trans. on Electron Devices*, vol. ED-32, no. 5, May 1985, pp. 903–909.

[19] K. Banerjee and A. Mehrotra, "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *IEEE Trans. on Electron Devices*, vol. 49, no. 11, Nov. 2002, pp. 2001–2007.

[20] L. Benini, G. D. Micheli, E. Macii, M. Poncino, and S. Quer, "Power optimization of core-based systems by address bus encoding," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 4, Dec. 1998, pp. 551-562.

[21] C. L. Berman, J. L. Carter, and K. F. Day, "The fanout problem: from theory to practice," in *Proc. of Decennial Caltech Conference Advanced Research in VLSI*, 1989, pp. 69-99.

[22] A. J. Bhavnagarwala, X. Tang, and J. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 4, Apr. 2001, pp. 658-665.

[23] D. Blaauw, R. Panda, and R. Chaudhry, "Design and analysis of power distribution networks," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001, pp. 499-522.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2003.

[25] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proc. of International Symposium on Low Power Electronics and Design*, 2000, pp. 9-14.

[26] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, Apr. 1992, pp. 473-484.

[27] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, Dec. 1997, pp. 436-443.

[28] R. Chau, S. Datta, M. Doczy, and J. Kavalieros, "Gate dielectric scaling for high-performance CMOS: From SiO2 to high-k," in *Proc. of International Workshop on Gate Insulator*, 2003, pp. 124–126.

[29] D. Chen and M. Sarrafzadeh, "An exact algorithm for low power library-specific gate re-sizing," in *Proc. of Design Automation Conference*, 1996, pp. 783-788.

[30] G. Chen and E. Friedman, "Low power repeaters driving RC and RLC interconnects with delay and bandwidth constraints," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 2, Feb. 2006, pp. 161-172.

[31] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proc. of Design Automation Conference*, 1997, pp. 638-643.

[32] W. Chen, C. Hsieh, and M. Pedram, "Simultaneous gate sizing and fanout optimization," in *Proc. of International Conference on Computer-Aided Design*, 2000, pp. 374-378.

[33] S. Chun, "Methodologies for modeling simultaneous switching noise in multi-layered packages and boards," Ph.D. dissertation, Georgia Institute of Technology, 2002.

[34] J. J. Clement, "Electromigration reliability," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001, pp. 429-448.

[35] S. Cserveny, L. Sumanen, J. Masgonty, and C. Piguet, "Locally switched and limited source-body bias and other leakage reduction techniques for a low-power embedded SRAM," *IEEE Trans. on Circuits and Systems-II: Express Briefs*, vol. 52, no. 10, Oct. 2005, pp. 636-640.

[36] W. Dally and J. Poulton, *Digital Systems Engineering*. New York, NY: Cambridge University Press, 1998.

[37] A. Dancy and A. Chandrakasan, "Ultra low power control circuits for PWM converters," in *Proc. of Power Electronics Specialists Conference*, 1997, pp. 21-27.

[38] V. De, A. Keshavarzi, S. Narendra, and J. Kao, "Techniques for leakage power reduction," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001.

[39] A. Devgan, "Efficient coupled noise estimation for on-chip interconnects " in *Proc. of International Conference on Computer-Aided Design*, 1997, pp. 147-151.

[40] F. Fallah and M. Pedram, "Standby and active leakage current control and minimization in CMOS VLSI circuits," *IEICE Trans. on Electronics, Special Section on Low-Power LSI and Low-Power IP*, vol. E88-C, no. 4, Apr. 2005, pp. 509-519.

[41] H. Fatemi, B. Amelifard, and M. Pedram, "Power optimal MTCMOS repeater insertion for global buses," in *Proc. of International Symposium on Low Power Electronics and Design*, 2007, pp. 98-103.

[42] H. Fatemi, S. Nazarian, and M. Pedram, "A current-based method for short circuit power calculation under noisy input waveforms," in *Proc. of Asia and South Pacific Design Automation Conference*, 2007, pp. 774-779.

[43] K. Flautner, N. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," in *Proc. of International Symposium on Computer Architecture*, 2002, pp. 148-157.

[44] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York, NY: Academic Press, 1981.

[45] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics : A Foundation for Computer Science*. Reading, MA: Addison-Wesley, 1990.

[46] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene, "Read stability and write-ability analysis of SRAM cells for nanometer technologies," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, Nov. 2006, pp. 2577-2588.

[47] Z. Guo, S. Balasubramanian, R. Zlatanovici, T.-J. King, and B. Nikolić, "FinFET-based SRAM design," in *Proc. of International Symposium on Low Power Electronics and Design*, 2005, pp. 2-7.

[48] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Gate-length biasing for runtime-leakage control," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 8, Aug. 2006, pp. 1475-1485.

[49] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Selective gate-length biasing for cost-effective runtime leakage control," in *Proc. of Design Automation Conference*, 2004, pp. 327-330

[50] F. Hamzaoglu and M. Stan, "Circuit-level techniques to control gate leakage for sub-100nm CMOS," in *Proc. of International Symposium on Low Power Electronics and Design*, 2002, pp. 60-63.

[51] F. Hamzaoglu, Y. Te, A. Keshavarzi, and K. Zhang, "Dual Vt-SRAM cells with full-swing single-ended bit line sensing for high-performance on-chip cache in 0.13μm technology generation," in *Proc. of International Symposium on Low Power Electronics and Design*, 2000, pp. 15–19.

[52] P. Hazucha, T. Karnik, B. A. Bloechel, C. Parsons, *et al.*, "Area-efficient linear regulator with ultra-fast load regulation," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, Apr. 2005, pp. 933-940.

[53] P. Hazucha and C. Svensson, "Impact of CMOS technology scaling on the atmospheric neutron soft error rate," *IEEE Trans. on Nuclear Science*, vol. 47, no. 6, Dec. 2000, pp. 2586–2594.

[54] R. Heald and P. Wang, "Variability in sub-100nm SRAM designs," in *Proc. of International Conference on Computer-Aided Design*, 2004, pp. 347-352.

[55] S. Heo, K. Barr, and K. Asanovic, "Reducing power density through activity migration," in *Proc. of International Symposium on Low Power Electronics and Design* 2003, pp. 217-222.

[56] P. Heydari and M. Pedram, "Capacitive crosstalk noise in high speed VLSI circuits," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, Mar. 2005, pp. 478-488.

[57] P. Heydari and M. Pedram, "Ground bounce in digital VLSI circuits," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 2, Apr. 2003, pp. 180-193.

[58] HSPICE: The gold standard for accurate circuit simulation, [online] http://www.synopsys.com/products/mixedsignal/hspice/hspice.html

[59] B. Hu, Y. Watanabe, A. Kondratyev, and M. Marek-Sadowska, "Gain-based technology mapping for discrete-size cell libraries," in *Proc. of Design Automation Conference*, 2003, pp. 574-579.

[60] C. Hua, T. Cheng, and W. Hwang, "Distributed data-retention power gating technique for column and row co-controlled embedded SRAM," in *Proc. of IEEE International Workshop on Memory Technology, Design, and Testing*, 2005, pp. 129-134.

[61] S. Iman and M. Pedram, "An approach for multi-level logic optimization targeting low power," *IEEE Trans. on Computer Aided Design*, vol. 15, no. 8, Aug. 1996, pp. 889-901.

[62] S. Iman and M. Pedram, "Logic extraction and decomposition for low power," in *Proc. of Design Automation Conference*, 1995, pp. 248-253.

[63] S. Iman and M. Pedram, "POSE: Power optimization and synthesis environment," in *Proc. of Design Automation Conference*, 1996, pp. 21-26.

[64] A. Iranli and M. Pedram, "System-Level Power Management: An Overview," in *The VLSI Handbook*, W.-K. Chen, Ed. New York, NY: CRC Press, 2006.

[65] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2003 edition, [online] http://public.itrs.net/.

[66] I. Jiang, Y. Chang, and I. Jou, "Crosstalk-driven interconnect optimization by simultaneous gate and wire sizing," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 19, no. 9, Sep. 2000, pp. 999-1010.

[67] H. Jyu, S. Malik, S. Devadas, and K. W. Keutzer, "Statistical timing analysis of combinational logic circuits," *IEEE Trans. on Very Large Scale iNtegration (VLSI) Systems*, vol. 1, no. 2, Jun. 1993, pp. 126-137.

[68] T. Kamik, B. Bloechel, K. Soumyanath, V. De, and S. Borkar, "Scaling trends of cosmic rays induced soft errors in static latches beyond 0.18um," in *Proc. of Symposium on VLSI Circuits*, 2001, pp. 61-62.

[69] S. Karandikar and S. Sapatnekar, "Logical effort based technology mapping," in *Proc. of International Conference on Computer-Aided Design*, 2004, pp. 419-422.

[70] C. Kim, J. Kim, I. Chang, and K. Roy, "PVT-aware leakage reduction for on-die caches with improved read stability," in *Proc. of International Solid-State Circuit Conference*, 2005, pp. 482-483.

[71] C. Kim and K. Roy, "Dynamic Vt SRAM: a leakage tolerant cache memory for low voltage microprocessor," in *Proc. of International Symposium on Low Power Electronics and Design*, 2002, pp. 251-254.

[72] C. H. Kim, J. Kim, S. Mukhopadhyay, and K. Roy, "A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 3, Mar. 2005, pp. 349-357.

[73] K. Kodandapani, J. Grodstein, A. Domic, and H. Touati, "A simple algorithm for fanout optimization using high-performance buffer libraries," in *Proc. of International Conference on Computer-Aided Design*, 1993, pp. 466-471.

[74] D. L. Kreher and D. R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*. Boca Raton, FL: CRC Press, 1999.

[75] D. S. Kung, "A fast fanout optimization algorithm for near-continuous buffer libraries," in *Proc. of Design Automation Conference*, 1998, pp. 352-355.

[76] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, *et al.*, "Managing power and performance for System-on-Chip designs using voltage islands," in *Proc. of International Conference on Computer Aided Design*, 2002, pp. 195-202.

[77] W. Lau and S. Sanders, "An integrated controller for a high frequency buck converter," in *Proc. of Power Electronics Specialists Conference*, 1997, pp. 246-254.

[78] J. Le, X. Li, and L. T. Pileggi, "STAC: statistical timing analysis with correlation," in *Proc. of Design Automation Conference*, 2004, pp. 343-348.

[79] D. Lee, D. Blaauw, and D. Sylvester, "Gate oxide leakage current analysis and reduction for VLSI circuits," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, Feb. 2004, pp. 155-166.

[80] D. Lee, W. Kwong, D. Blaauw, and D. Sylvester, "Analysis and minimization techniques for total leakage considering gate oxide leakage," in *Proc. of Design Automation Conference*, 2003, pp. 175-180

[81] Magma Design Automation. Gain Based Synthesis: Speeding RTL to Silicon, 2002.

[82] R. Marculescu, D. Marculescu, and M. Pedram, "Probabilistic modeling of dependencies during switching activity analysis," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 17, no. 2, Feb. 1998, pp. 73-83.

[83] R. Marculescu, D. Marculescu, and M. Pedram, "Switching activity estimation considering spatiotemporal correlations," in *Proc. of International Conference on Computer Aided Design*, 1994, pp. 294-299.

[84] K. Mohanram and N. A. Touba, "Cost-effective approach for reducing soft error failure rate in logic circuits," in *Proc. of International Test Conference*, 2003, pp. 893-901

[85] S. P. Mohanty, R. Velagapudi, and E. Kougianos, "Dual-k versus dual-T technique for gate leakage reduction: a comparative perspective," in *Proc. of International Symposium on Quality Electronic Design*, 2006, pp. 564-569.

[86] C. Molina, C. Aliagas, M. Garcia, A. Gonzalez, and J. Tubella, "Non redundant data cache," in *Proc. of International Symposium on Low Power Electronics and Design*, 2003, pp. 274-277.

[87] Mosek Optimization Software, [online] http://www.mosek.com

[88] V. Mukherjee, S. P. Mohanty, and E. Kougianos, "A dual dielectric approach for performance aware gate tunneling reduction in combinational circuits," in *Proc. of International Conference on Computer Design*, 2005, pp. 431-437.

[89] S. Mukhopadhyay, H. Mahmoodi-Meimand, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 12, Dec. 2005, pp. 1859-1880.

[90] S. Naffziger, B. Stackhouse, T. Grutkowski, D. Josephson, *et al.*, "The implementation of a 2-core, multi-threaded Itanium family processor," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, Jan. 2006, pp. 197-209.

[91] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar, "Scaling of stack effect and its application for leakage reduction," in *Proc. of International Symposium on Low Power Electronics and Design* 2001, pp. 195-200.

[92] K. Nose and T. Sakurai, "Analysis and future trend of short circuit power," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 19, no. 9, Sep. 2000, pp. 1023-1030.

[93] K. J. Nowka, G. D. Carpenter, E. W. MacDonald, H. C. Ngo, *et al.*, "A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, Nov. 2002, pp. 1441- 1447.

[94] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. of Design Automation Conference*, 2004, pp. 337-342.

[95] G. Patounakis, Y. W. Li, and K. L. Shepard, "A fully integrated on-chip DC-DC conversion and power management system," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, Mar. 2004, pp. 443-451.

[96] M. Pedram, "Power minimization in IC design: principles and applications," *ACM Trans. on Design Automation of Electronic Systems*, vol. 1, no. 1, Jan. 1996, pp. 3-56.

[97] M. Pedram and J. Rabaey, *Power Aware Design Methodologies*. Boston, MA: Kluwer Academic Publishers, 2002.

[98] M. D. Powell, S. Yang., B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-Vdd: a circuit technique to reduce leakage in cache memories," in *Proc. of International Symposium on Low Power Electronics Design*, 2000, pp. 90–95.

[99] Predictive Technology Model, [online] http://www.eas.asu.edu/~ptm/

[100] R. Preston, "Register files and caches," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001, pp. 285-308.

[101] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, "SRAM leakage suppression by minimizing standby supply voltage," in *Proc. of International Symposium on Quality Electronic Design*, 2004, pp. 55- 60.

[102] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A Coding framework for low-power address and data busses," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 2, Jun. 1998, pp. 212-221.

[103] R. Rao, K. Agarwal, D. Sylvester, R. Brown, *et al.*, "Approaches to run-time and standby mode leakage reduction in global buses," in *Proc. of International Symposium on Low Power Electronics and Design*, 2004, pp. 188-193.

[104] P. Rezvani and M. Pedram, "A fanout optimization algorithm based on the effort delay model," *IEEE Trans. on Computer Aided Design*, vol. 22, no. 12, Dec. 2003, pp. 1671-1678.

[105] G. A. Rincon-Mora, "Current efficient, low voltage, low drop-out regulators," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, 1996.

[106] S.Heo, K. Barr, M. Hampton, and K. Asanovic, "Dynamic fine-grain leakage reduction using leakage-biased bitlines," in *Proc. of International Symposium on Computer Architecture*, 2002, pp. 137-147.

[107] T. Sakurai and A. R. Newton, "A simple MOSFET model for circuit analysis," *IEEE Trans. on Electron Devices*, vol. 38, no. 4, Apr. 1991, pp. 887-894.

[108] A. Salek, J. Lou, and M. Pedram, "Hierarchical buffered routing tree generation," *IEEE Trans. on Computer Aided Design*, vol. 21, no. 5, May 2002, pp. 554-567.

[109] S. S. Sapatnekar, "Power-delay optimization in gate sizing," *ACM Trans. on Design Automation of Electronic Systems*, vol. 5, no. 1, Jan. 2000, pp. 98-114.

[110] E. Seevinck, F. J. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE Journal of Solid-State Circuits*, vol. 22, no. 5, Oct. 1987, pp. 748-754.

[111] N. Seifert, D. Moyer, N. Leland, and R. Hokinson, "Historical trend in alpha-particle induced soft error rates of the alpha microprocessor," in *Proc. of International Reliability Physics Symposium*, 2001, pp. 259-265.

[112] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, *et al.*, "SIS: A System for Sequential Circuit Synthesis," University of California, Berkeley, Report M92/41, May 1992.

[113] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On average power dissipation and random pattern testability of CMOS combinational logic networks," in *Proc. of International Conference on Computer Aided Design* 1992, pp. 402-407.

[114] K. Shi and D. Howard, "Challenges in sleep transistor design and implementation in low-power design," in *Proc. of Design Automation Conference*, 2006, pp. 113-116.

[115] Y. Shin, S. Chae, and K. Choi, "Partial bus-invert coding for power optimization of application-specific systems," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 9, no. 2, Apr. 2001, pp. 377-383.

[116] K. J. Singh and A. Sangiovanni-Vincentelli, "A heuristic algorithm for the fanout problem," in *Proc. of Design Automation Conference*, 1990, pp. 357-360.

[117] D. Sinha, D. Khalil, Y. Ismail, and H. Zhou, "A timing dependent power estimation framework considering coupling," in *Proc. of International Conference on Computer Aided Design*, 2006, pp. 401-407.

[118] N. Sirisantana, L. Wei, and K. Roy, "High performance low power CMOS circuits using multiple channel length and multiple oxide thickness," in *Proc. of International Conference on Computer Design*, 2000, pp. 227-232.

[119] A. Sirvastava, "Simultaneous Vt selection and assignment for leakage optimization," in *Proc. of International Symposium on Low Power Electronics and Design*, 2003, pp. 146-151.

[120] L. Smith, R. Anderson, D. Forehand, T. Pelc, and T. Roy, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Trans. on Advanced Packaging*, vol. 22, no. 3, Aug. 1999, pp. 284-291.

[121] L. Stok, D. S. Kung, D. Brand, and A. D. Drumm, "BooleDozer: logic synthesis for ASICs," *IBM Journal of Research and Development*, vol. 40, no. 4, Jul. 1996, pp. 407-430.

[122] A. Stratakos, "High-efficiency low-voltage DC-DC conversion for portable applications," Ph.D. dissertation, University of California, Berkeley, 1998.

[123] H. Su, S. S. Sapatnekar, and S. R. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in *Proc. of International Symposium on Physical Design*, 2002, pp. 68-73.

[124] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. San Francisco, CA: Morgan Kaufmann, 1999.

[125] M. Taherzadeh-S., B. Amelifard, H. Iman-Eini, F. Farbiz, *et al.*, "Power and delay estimation of CMOS inverters using fully analytical approach," in *Proc. of IEEE Southwest Symposium on Mixed-Signal Design*, 2003, pp. 112-115.

[126] Y. Taur, "CMOS scaling and issues in sub-0.25 μm systems," in *Design of High-Performance Microprocessor Circuits*, Chandrakasan, W. J. Bowhill, and F. Fox, Eds. Piscataway, NJ: IEEE, 2001, pp. 27–45.

[127] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. New York, NY: Cambridge University Press, 1998.

[128] Texas Instruments, "TPS60502 Datasheet," [online] http://www.ti.com/lit/gpn/tps60502

[129] Texas Instruments, "TPS60503 Datasheet," [online] http://www.ti.com/lit/gpn/tps60503

[130] M. Togo, K. Noda, and T. Tanigawa, "Multiple-thickness gate oxide and dual-gate technologies for high-performance logic embedded DRAMs," in *Proc. of IEDM Technical Digest*, 1998, pp. 347–350.

[131] H. Touati, "Performance-oriented technology mapping," Ph.D. dissertation, University of California, Berkeley, 1990.

[132] Y. Tsai, D. Duarte, N. Vijaykrishnan, and M. J. Irwin, "Characterization and modeling of run-time techniques for leakage power reduction," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 11, Nov. 2004, pp. 1221-1233.

[133] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, *et al.*, "Dynamic sleep transistor and body bias for active leakage power control of microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, Nov. 2003, pp. 1838-1845.

[134] K. Usami and M. Horowitz, "Cluster voltage scaling technique for low-power design," in *Proc. of International Symposium on Low Power Design*, 1995, pp. 3-8.

[135] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, *et al.*, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, Mar. 1998, pp. 463-472.

[136] H. Veendrick, "Short circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 4, Aug. 1984, pp. 468-473.

[137] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Proc. of International Conference on on Computer Aided Design*, 2001, pp. 57-63.

[138] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. of Design Automation Conference*, 2004, pp. 331-336.

[139] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 16, no. 3, Mar. 1997, pp. 290-298.

[140] K. Wang and M. Marek-Sadowska, "On-chip power-supply network optimization using multigrid-based technique," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, Mar. 2005, pp. 407-417.

[141] D. Weiss, J. Wuu, and V. Chin, "The on-chip 3MB subarray based 3rd level cache on an Itanium microprocessor," in *Proc. of International Solid-State Circuits Conference*, 2002, pp. 112–113.

[142] M. Xakellis and F. Najm, "Statistical estimation of the switching activity in digital circuits," in *Proc. of Design Automation Conference*, 1994, pp. 728-733.

[143] K. Zhang, U. Bhattacharya, Z. Chen, F. Hamzaoglu, *et al.*, "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, Apr. 2005, pp. 895-901.

[144] K. Z. Zhang, U. Bhattacharya, Z. Chen, and F. Hamzaoglu, "A 3-GHz 70Mb SRAM in 65nm CMOS technology with integrated column-based dynamic power supply," in *Proc. of International Solid-State Circuits Conference*, 2005, pp. 474–475.

[145] M. Zhao, R. Panda, S. Sundareswaran, S. Yan, and Y. Fu, "A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming," in *Proc. of Design Automation Conference*, 2006, pp. 217-222.

[146] S. Zhao, K. Roy, and C. Koh, "Decoupling capacitance allocation for power supply noise suppression," in *Proc. of International Symposium on Physical Design*, 2001, pp. 66-71.

[147] D. Zhou and X. Liu, "Minimization of chip size and power consumption of high-speed VLSI buffers," in *Proc. of International Symposium on Physical Design*, 1997, pp. 186-191.

[148] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," in *Proc. of Design Automation Conference*, 1998, pp. 374-377.