

# Low-Power High-Level Synthesis for Nanoscale CMOS Circuits

Saraju P. Mohanty · Nagarajan Ranganathan  
Elias Kougianos · Priyadarsan Patra

# Low-Power High-Level Synthesis for Nanoscale CMOS Circuits

 Springer

Saraju P. Mohanty  
University of North Texas  
Denton, TX  
USA

Nagarajan Ranganathan  
University of South Florida  
Tampa, FL  
USA

Elias Kougianos  
University of North Texas  
Denton, TX  
USA

Priyadarsan Patra  
Intel Corporation  
Hillsboro, OR  
USA

ISBN: 978-0-387-76473-3

e-ISBN: 978-0-387-76474-0

DOI: 10.1007/978-0-387-76474-0

Library of Congress Control Number: 2008920380

© 2008 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

*To Uma, my sisters and my parents.*

*To Radhika, Sheela, Shiva and my parents.*

*To Diane and my family.*

*To Anu, Ashish, Adarsh and my parents  
Harihar and Padmabati.*

# Preface

*Low-Power High-Level Synthesis for Nanoscale CMOS Circuits* shows very-large-scale integration (VLSI) researchers and engineers how to minimize the different types of power consumption of digital circuits. This text provides the fundamentals of high-level (also called behavioral- or architectural-level) synthesis of digital circuits. In addition, this text provides adequate knowledge in the design of low-power CMOS circuits. Students will acquire a sufficient understanding of the different methodologies of power reduction while taking into account process variations during characterization and modeling.

This self-contained text is directed to nanoscale VLSI design engineers, graduate students in electrical engineering and computer scientists who are about to start their research in high-level VLSI design. It is also an important comprehensive source for microelectronics engineers who would like to understand the different aspects of power reduction. The text assumes basic knowledge of algorithms and familiarity with digital circuit design. Since the research focus in this field is continually progressing, students will acquire sufficient knowledge after studying this book so as to be able to understand more specialized literature on their own. Each chapter has simple relevant examples for a better grasp of the principles presented. Several algorithms are given to provide a better understanding of the underlying concepts. However, the goal of this book is to train the students to develop algorithms rationally.

The text provides a sufficient amount of fundamentals to become familiar with the terminology of the field. The main objective is to achieve in-depth knowledge in a few topics instead of a shallow, broader presentation. The purpose and objective of low-power high-level synthesis is explained in Chapter 1. The initial chapters deal with the basics of high-level synthesis (Chapter 2), power (Chapter 3) and power estimation (Chapter 4). In subsequent parts of the text, a detailed discussion of methodologies for the reduction of different types of power is presented: power reduction in Chapter 5, energy or average power reduction in Chapter 6, peak power reduction in Chapter 7 and transient power reduction in Chapter 8. As the dimensions of CMOS devices decrease to nanometer scale to adhere to Moore's law, different leakage mechanisms are observed. Different schemes to minimize various forms of leakage in ICs are presented in Chapter 9.

S.P. Mohanty  
Denton, TX  
USA

# Acknowledgments

Saraju P. Mohanty acknowledges the full support and encouragement from the Department of Computer Science and Engineering at the University of North Texas. This book derives a lot of material from the very-large-scale integration (VLSI) design courses that he teaches there as well as from his doctoral dissertation. The author is very thankful to his students for providing constant valuable feedback. Saraju P. Mohanty also thanks Dr. Krishna Kavi, the Department Chair and Dr. Oscar García, Founding Dean of the College of Engineering. He acknowledges the faculty and staff of the Department of Computer Science and Engineering for their generous support.

Nagarajan Ranganathan would like to express his gratitude to the Department of Computer Science and Engineering at the University of South Florida. He acknowledges with gratitude the research contributions of all his current and past students.

Elias Kougianos expresses his gratitude to the Department of Engineering Technology at the University of North Texas. In particular, he thanks the Department Chair, Dr. Nourredine Boubekri, for the collegial environment that he fosters in the department.

Priyadarsan Patra is grateful to Intel Corporation for fully supporting his authorship and for providing a world-class environment. He thanks his colleagues at the Strategic CAD Labs and the Microprocessor Technology Labs of Intel, where he has worked, and other colleagues from the leading product groups, for providing perspectives that have influenced what is good in this book.

The authors are grateful to Dr. Uma Choppali who has helped immensely in the preparation of figures in the book as well as Ms. Otilia Sánchez, M.A. Both of them have read the entire manuscript thoroughly and provided excellent editorial comments.

The publication of this book would not have been possible without the support of the Springer staff. We would like to thank Alex Greene and Katelyn Stanne for their promptness and patience in answering all our questions.

The authors are greatly indebted to their families for their patience, understanding and support during the innumerable hours of preoccupation with the writing of this book. Authors Saraju P. Mohanty and Elias Kougianos would like to acknowledge NSF award number 0702361.

# Contents

<b>Acronym Definition</b> .....	xxix
<b>1 Introduction</b> .....	1
<b>2 High-Level Synthesis Fundamentals</b> .....	5
2.1 Introduction .....	5
2.2 The Complete Chip Story: From Customers' Requirements to Silicon Chips for Customers .....	5
2.3 Various Phases of Circuit Design and Synthesis .....	7
2.4 High-Level or Behavioral Synthesis: What and Why .....	10
2.5 Various Phases of High-Level Synthesis .....	11
2.5.1 Compilation .....	12
2.5.2 Transformation .....	12
2.5.3 Scheduling .....	13
2.5.4 Selection or Allocation .....	13
2.5.5 Binding or Assignment .....	13
2.5.6 Output Generation .....	14
2.5.7 A Demonstrative Example .....	14
2.6 Behavioral HDL to CDFG Translation or Compilation .....	14
2.7 Scheduling Algorithms .....	16
2.7.1 ASAP and ALAP Scheduling and Mobility .....	19
2.7.2 Integer Linear Programming (ILP) Scheduling .....	19
2.7.3 List-Based Scheduling (LBS) .....	23
2.7.4 Force-Directed Scheduling (FDS) .....	25
2.7.5 Game Theory Scheduling (GTS) .....	26
2.7.6 Tabu Search Scheduling (TSS) .....	28
2.7.7 Simulated Annealing Scheduling (SAS) .....	29
2.7.8 Genetic Algorithm Scheduling (GAS) .....	30
2.7.9 Ant Colony Scheduling (ACS) .....	30
2.7.10 Automata-Based Symbolic Scheduling .....	31
2.7.11 Chaining, Multicycling and Pipelining Data Paths .....	31

2.8	Binding or Allocations Algorithms	32
2.8.1	Clique Partitioning Approach	33
2.8.2	Graph Coloring Approach	33
2.8.3	Left Edge Algorithm for Register Optimization	35
2.8.4	Integer Linear Programming (ILP) Binding	36
2.8.5	Heuristic Algorithm to Solve Clique Partitioning	37
2.8.6	GTS Algorithm	37
2.9	Control Synthesis	38
2.10	High-Level Synthesis Benchmarks	38
2.11	High-Level Synthesis Tools	44
2.11.1	CatapultC from Mentor Graphics	44
2.11.2	CyberWorkBench from NEC	44
2.11.3	PICO Express from Synfora	44
2.11.4	Cynthesizer from Forte Design Systems	44
2.11.5	Cascade from Critical Blue	45
2.11.6	Agility Compiler from Celoxica	45
2.11.7	eXCite from Y Explorations	45
2.11.8	ESEComp from BlueSpec	45
2.11.9	VCS from Synopsys	45
2.11.10	NC-SC, NC-Verilog and NC-VHDL from Cadence	45
2.11.11	Synplify from Synplicity	46
2.11.12	ISE from Xilinx	46
2.11.13	Quartus from Altera	46
2.12	Summary and Conclusions	46
<b>3</b>	<b>Power Modeling and Estimation at Transistor and Logic</b>	
	<b>Gate Levels</b>	47
3.1	Introduction	47
3.2	CMOS Technology Trends	48
3.3	Current Conduction Mechanisms in Nano-CMOS Devices:	
	A Résumé	49
3.3.1	The Ideal ON and OFF States	49
3.3.2	Junction Reverse Bias Current	50
3.3.3	Drain-Induced Barrier Lowering (DIBL)	51
3.3.4	Subthreshold Leakage	51
3.3.5	Gate-Induced Drain Leakage (GIDL)	52
3.3.6	Punch-Through	53
3.3.7	Hot-Carrier Injection	53
3.3.8	Band-to-Band Tunneling (BTBT)	54
3.3.9	Gate-Oxide Tunneling	55
3.4	Power Dissipation in Nano-CMOS Logic Gates	59
3.4.1	Static, Dynamic and Leakage Power Dissipation	59
3.4.2	Case Study: The 45 nm NOT, NAND, NOR CMOS Gates	60
3.5	Process Variation Effects	67
3.5.1	Origins and Sources of Process Variation	67



- 3.5.2 Methodologies to Accommodate Process Variation . . . . . 68
- 3.6 From Gates to Functional Units: A Power Modeling and Estimation Perspective . . . . . 72
  - 3.6.1 SPICE level . . . . . 73
  - 3.6.2 Probabilistic and Statistical Techniques . . . . . 75
- 3.7 Summary and Conclusions . . . . . 78
- 4 Architectural Power Modeling and Estimation . . . . . 81**
  - 4.1 Introduction . . . . . 81
  - 4.2 Architecture-Level Estimation . . . . . 84
  - 4.3 Dynamic Power Modeling and Estimation . . . . . 90
    - 4.3.1 Abstract Data Path Power Estimation . . . . . 91
    - 4.3.2 Capacitance Estimation . . . . . 92
    - 4.3.3 Macro-modeling for Dynamic Power . . . . . 93
    - 4.3.4 Estimation of Bounds on Average Power . . . . . 94
  - 4.4 Leakage Modeling . . . . . 94
    - 4.4.1 Subthreshold and Gate-Oxide Leakage Power Modeling and Estimation . . . . . 95
    - 4.4.2 Methods for Total Leakage Estimation . . . . . 97
  - 4.5 Modeling and Analysis of Architectural Components . . . . . 100
    - 4.5.1 Design-Optimization-Aware Estimation . . . . . 100
    - 4.5.2 Estimating Under Variation Effects . . . . . 103
    - 4.5.3 Estimating Power in Control and Data Path Logic . . . . . 104
    - 4.5.4 Communication Components . . . . . 106
  - 4.6 Register Files . . . . . 108
    - 4.6.1 Methodology . . . . . 109
    - 4.6.2 Basic Power Model . . . . . 109
    - 4.6.3 Pipelined Register Files . . . . . 111
    - 4.6.4 Physical Dimensions and Latency . . . . . 111
    - 4.6.5 Area, Power, Delay Models . . . . . 115
    - 4.6.6 Device Sizing . . . . . 119
  - 4.7 Cache Arrays . . . . . 120
    - 4.7.1 CACTI Dynamic Power Model for Caches . . . . . 120
    - 4.7.2 Leakage Modeling for Arrays . . . . . 123
  - 4.8 Validation and Accuracy . . . . . 125
    - 4.8.1 Model Validation: Arrays as an Example . . . . . 125
    - 4.8.2 Simulator Accuracy . . . . . 127
    - 4.8.3 Power Model Accuracy . . . . . 127
  - 4.9 Effect of Temperature on Power . . . . . 128
  - 4.10 Summary and Conclusions . . . . . 129
- 5 Power Reduction Fundamentals . . . . . 131**
  - 5.1 Introduction . . . . . 131
  - 5.2 Power Dissipation or Consumption Profile of CMOS Circuits . . . . . 131
  - 5.3 Why Low-Power Design? . . . . . 133

5.4	Why Energy or Average Power Reduction? . . . . .	135
5.5	Why Peak Power Minimization? . . . . .	136
5.6	Why Transient Power Minimization? . . . . .	137
5.7	Why Leakage Power Minimization? . . . . .	137
5.8	Power Reduction Mechanisms at Different Levels of Abstraction . . .	138
5.9	Why Power Optimization During High-Level or Behavioral Synthesis? . . . . .	138
5.10	Methods for Power Reduction in High-Level Synthesis . . . . .	139
5.11	Frequency and/or Voltage Scaling for Dynamic Power Reduction . .	140
5.11.1	What Is Voltage or Frequency Scaling? . . . . .	140
5.11.2	Why Frequency and/or Voltage Scaling? . . . . .	142
5.11.3	Energy or Average Power Reduction Using Voltage or Frequency Scaling . . . . .	143
5.11.4	Peak Power Reduction Using Voltage and Frequency Scaling . . . . .	145
5.11.5	Issues in Multiple Supply Voltage-Based Design . . . . .	146
5.11.6	Voltage-Level Converter Design . . . . .	146
5.11.7	Dynamic Frequency Clocking Unit Design . . . . .	148
5.12	$V_{Th}$ Scaling for Subthreshold Leakage Reduction . . . . .	150
5.12.1	The Concept . . . . .	150
5.12.2	Multiple Threshold CMOS (MTCMOS) Technology . . . . .	151
5.12.3	Variable Threshold CMOS (VTCMOS) Technology . . . . .	152
5.12.4	Dynamic Threshold CMOS (DTCMOS) Technology . . . . .	152
5.12.5	Leakage Control Transistor (LECTOR) Technique . . . . .	152
5.12.6	The Issues . . . . .	153
5.13	$T_{ox}$ , $K$ or $L$ Scaling for Gate-Oxide Leakage Reduction . . . . .	153
5.13.1	The Concept . . . . .	153
5.13.2	Multiple Oxide Thickness CMOS (MOXCMOS) Technology . . . . .	154
5.13.3	Multiple Dielectric ( $\kappa$ ) (MKCMOS) Technology . . . . .	154
5.13.4	The Issues . . . . .	154
5.14	Transformation Techniques for Power Reduction . . . . .	155
5.14.1	Operation Reduction . . . . .	155
5.14.2	Operation Substitution . . . . .	156
5.15	Increased Parallelism and Pipelining with Architecture-Driven Voltage Scaling for Power Reduction . . . . .	156
5.15.1	Parallelism with Voltage Scaling . . . . .	157
5.15.2	Pipelining with Voltage Scaling . . . . .	157
5.16	Guarded Evaluation to Reduce Power . . . . .	159
5.17	Precomputation-Based Power Reduction . . . . .	160
5.18	Clock Gating to Reduce Clock Power Dissipation . . . . .	161
5.19	Interconnect Power Minimization . . . . .	161
5.20	Summary and Conclusions . . . . .	161

- 6 Energy or Average Power Reduction** ..... 163
  - 6.1 Introduction ..... 163
  - 6.2 Target Architecture and Data Path Specifications  
for Multiple Voltage ..... 164
  - 6.3 ILP-Based Scheduling for EDP Reduction ..... 165
    - 6.3.1 Introduction ..... 165
    - 6.3.2 EDP Modeling of a DFG ..... 166
    - 6.3.3 ILP Formulations for EDPs ..... 168
    - 6.3.4 ILP-Based Data Path Scheduling Algorithm ..... 170
    - 6.3.5 Experimental Results ..... 172
    - 6.3.6 Conclusions ..... 174
  - 6.4 Heuristic-Based Scheduling Algorithm for Energy Minimization ... 176
    - 6.4.1 Introduction ..... 176
    - 6.4.2 Time-Constrained Scheduling: TC-DFC ..... 177
    - 6.4.3 Resource-Constrained Scheduling: RC-DFC ..... 183
    - 6.4.4 Experimental Results ..... 188
    - 6.4.5 Conclusions ..... 190
  - 6.5 Data Path Scheduling for Energy or Average Power Reduction  
Using Voltage Reduction ..... 191
    - 6.5.1 Time- or Resource-Constrained Scheduling Algorithms ... 191
    - 6.5.2 Time- and Resource-Constrained Scheduling Algorithms .. 193
  - 6.6 Switching Activity Reduction During High-Level Synthesis ..... 194
    - 6.6.1 Scheduling and/or Allocation for Switching  
Activity Reduction ..... 195
    - 6.6.2 Scheduling and/or Binding for Switching  
Activity Reduction ..... 198
  - 6.7 Summary and Conclusions ..... 200
  
- 7 Peak Power Reduction** ..... 201
  - 7.1 Introduction ..... 201
  - 7.2 Peak and Average Power Dissipation Modeling of a Data  
Path Circuit ..... 201
  - 7.3 ILP-Based Scheduling for Peak Power Reduction ..... 204
    - 7.3.1 ILP Formulations ..... 205
    - 7.3.2 ILP-Based Scheduler ..... 207
    - 7.3.3 Experimental Results ..... 211
    - 7.3.4 Conclusions ..... 215
  - 7.4 ILP-Based Scheduling for Simultaneous Peak and Average Power  
Reduction ..... 215
    - 7.4.1 ILP Formulations ..... 215
    - 7.4.2 ILP-Based Scheduler ..... 216
    - 7.4.3 Experimental Results ..... 219
    - 7.4.4 Conclusions ..... 222
  - 7.5 Scheduling or Binding for Peak Power Reduction ..... 222
    - 7.5.1 Scheduling Algorithms ..... 222

7.5.2	Binding Algorithms	223
7.6	Summary and Conclusions	224
<b>8</b>	<b>Transient Power Reduction</b>	<b>225</b>
8.1	Introduction	225
8.2	Modeling for Power Transience or Fluctuation of a Data Path Circuit	225
8.2.1	Model 1: CPF Using Mean Deviation	226
8.2.2	Model 2: CPF Using Cycle-to-Cycle Gradient	229
8.2.3	Minimization of CPF as an Objective Function	230
8.3	Heuristic-Based Scheduling Algorithm for CPF Minimization	232
8.3.1	Introduction	232
8.3.2	Algorithm Flow	232
8.3.3	Pseudocode of the Algorithm Heuristic	234
8.3.4	Algorithm Time Complexity	236
8.3.5	Experimental Results	236
8.3.6	Conclusions	238
8.4	Modified Cycle Power Function (CPF*)	242
8.5	Linear Programming Modeling of Non-linearities	244
8.5.1	Linear Programming Formulation Involving the Sum of Absolute Deviations	244
8.5.2	Linear Programming Formulation Involving Fractions	245
8.6	ILP Formulations to Minimize (CPF*)	246
8.6.1	For MVDFC Operation	246
8.6.2	For MVMC Operation	249
8.7	ILP-Based Scheduling Algorithm for CPF* Minimization	251
8.7.1	Introduction	251
8.7.2	Algorithm	252
8.7.3	Experimental Results	254
8.7.4	Conclusions	259
8.8	Data Monitoring for Transient Power Minimization	259
8.9	Summary and Conclusions	259
<b>9</b>	<b>Leakage Power Reduction</b>	<b>261</b>
9.1	Introduction	261
9.2	Gate-Oxide Leakage Reduction	262
9.2.1	Dual- $T_{ox}$ Technique	262
9.2.2	Dual- $\kappa$ Technique	271
9.3	Subthreshold Leakage Reduction	274
9.3.1	Prioritization Algorithm for Dual- $V_{Th}$ -Based Optimization	274
9.3.2	MTCMOS-Based Clique Partitioning for Subthreshold Leakage Reduction	275
9.3.3	MTCMOS-Based Knapsack Binding for Subthreshold Leakage Reduction	275

- 9.3.4 Power Island Technique for Subthreshold Leakage Reduction ..... 276
- 9.3.5 Maximum Weight-Independent Set (MWIS) Problem Heuristic for Dual- $V_{Th}$ -Based Optimization ..... 276
- 9.4 Summary and Conclusions ..... 276
- 10 Conclusions and Future Directions ..... 277**
- References ..... 281**
- Index ..... 299**

# List of Figures

2.1	The complete chip story: from user specification to user's chip . . . . .	6
2.2	Different levels of design abstractions . . . . .	7
2.3	Description of hardware in different domains and abstractions through the Gajski–Kuhn Y-chart . . . . .	9
2.4	Various synthesis processes that translate a system description to a physical description in silicon. Logic level to physical (layout) level synthesis occurs in one step through physical synthesis, but the transistor level or circuit level is shown because it is a form of description that can take place transparently . . . . .	10
2.5	Various phases of high-level synthesis . . . . .	12
2.6	A synthesis example: step 1 to step 3 . . . . .	15
2.7	Synthesis example: step 4 to step 6 . . . . .	16
2.8	Control and DFG example . . . . .	17
2.9	Steps for converting VHDL to CDFG . . . . .	17
2.10	VHDL to CDFG conversion example . . . . .	18
2.11	Different types of scheduling algorithms . . . . .	18
2.12	DFG and sequencing DFG of HAL differential equation solver . . . . .	20
2.13	ASAP and ALAP schedules with mobility graph . . . . .	21
2.14	Schedule resulting using the ILP-based algorithm. . . . .	23
2.15	Schedule resulting by using a list-based scheduling algorithm assuming unit delay for each operation with a resource constraint of three multipliers and one ALU . . . . .	24
2.16	Sequencing DFGs for chaining, multicycling or pipelining data paths . . . . .	32
2.17	DFG and corresponding compatibility graph and conflict graph for two types of resources . . . . .	34
2.18	Selected high-level synthesis benchmarks as sequencing DFG-1 . . . . .	39
2.19	Selected high-level synthesis benchmarks as sequencing DFG-2 . . . . .	40
2.20	Selected high-level synthesis benchmarks as sequencing DFG-3 . . . . .	41
2.21	Selected high-level synthesis benchmarks as sequencing DFG-4 . . . . .	42
2.22	Selected high-level synthesis benchmarks as sequencing DFG-5 . . . . .	43

- 3.1 Prediction of trends of major sources of power dissipation in nano-CMOS transistors and circuits [17]. It is predicted (area to the right of the dotted line) that gate-oxide leakage and subthreshold leakage will dominate the total power dissipation of nano-CMOS circuits. Note the logarithmic scale for power . . . . . 49
- 3.2 Current flow paths in a nano-CMOS transistor during power dissipation in different states of its operation [308, 211, 55]:  $I_1$  - drain-to-source active current (ON state),  $I_2$  - drain-to-source short-circuit current (ON state),  $I_3$  - subthreshold leakage (OFF state),  $I_4$  - gate leakage (both ON and OFF states),  $I_5$  - gate current due to hot-carrier injection (both ON and OFF states),  $I_6$  - channel punch-through current (OFF state),  $I_7$  - gate-induced drain leakage (OFF state),  $I_8$  - band-to-band tunneling current (OFF state),  $I_9$  - reverse bias PN junction leakage (both ON and OFF states) . . . . . 50
- 3.3 Drain current as a function of gate voltage for an NMOS device with drawn channel length of  $L = 45$  nm, width  $W = 45$  nm, oxide thickness  $T_{ox} = 1.4$  nm and threshold voltage  $V_{Th} = 0.22$  V. The value of  $V_D$  in the upper curve is higher than typical operating values in order to exaggerate the DIBL and GIDL effects for illustrative purposes . . . . . 52
- 3.4 Energy band diagram of a reverse-biased diode exhibiting BTBT current.  $E_C$  and  $E_V$  are the conduction and valence band edges, respectively, and  $E_F$  is the Fermi level.  $\phi_{bi}$  is the built-in potential of the junction and  $V_R$  is the externally applied bias . . . . . 54
- 3.5 Tunneling mechanisms in the gate-oxide region of nano-CMOS [308]. . . . . 56
- 3.6 Gate-oxide tunneling current component flow in the various regions of operation of an NMOS transistor.  $I_{gd}$  and  $I_{gs}$  are the components from the diffusions (drain and source, respectively) directly to the gate.  $I_{gcd}$  and  $I_{gcs}$  are the components from the diffusions (drain and source, respectively) to the gate via the channel. The gate to bulk tunneling current ( $I_{gb}$ ) is negligible and is not shown . . . . . 57
- 3.7 Study of gate-oxide tunneling current ( $I_{gate}$ ) components in response to a pulse input. The input voltage  $V_{gs}$  is given for a complete cycle (low to high and then high to low) to bias the NMOS in the OFF, OFF-to-ON, ON and ON-to-OFF states . . . . . 58
- 3.8 Taxonomy of current components contributing to dynamic and static power dissipation . . . . . 61
- 3.9 Gate-oxide tunneling current paths in various switching states of an inverter logic gate.  $V_{Th}$  voltage drops have not been taken into account. The heavy arrows indicate the direction of the tunneling current flow in each device. In order to provide a clear perspective, individual components like  $I_{gd}$ ,  $I_{gs}$ ,  $I_{gcd}$  and  $I_{gcs}$  as in Fig. 3.6 are not shown . . . . . 61

3.10	Gate-oxide tunneling current paths in various switching states of a two-input NAND logic gate. $V_{Th}$ voltage drops have not been taken into account. The heavy arrows indicate the direction of the tunneling current flow in each device. In order to provide a clear perspective, individual components like $I_{gd}$ , $I_{gs}$ , $I_{gcd}$ and $I_{gcs}$ as in Fig. 3.6 are not shown . . . . .	62
3.11	Transient response of a two-input NAND gate for a fixed load. The left figures show voltages for two input voltages and the resulting output voltage. The gate-oxide tunneling current components in various individual transistors and the total value for the logic gate are presented on the right side figure. . . . .	63
3.12	Transient response of a two-input NOR gate for a fixed load. The left figures show voltages for two input voltages and the resulting output voltage. The gate-oxide tunneling current components in various individual transistors and the total value for the logic gate are presented on the right side figure. . . . .	63
3.13	Change in tunneling current due to oxide thickness and power supply variations. The NAND gate always exhibits lower leakage than the NOR gate . . . . .	65
3.14	Change in tunneling capacitance due to dielectric, oxide thickness and power supply variations. The NOR gate always exhibits lower capacitance than the NAND gate . . . . .	66
3.15	Average gate-oxide leakage and propagation delay for various logic gates. It is evident from the figure that NAND2 has minimal average gate leakage and propagation delay, thus making it most suitable for functional unit realization for low gate leakage data path circuit synthesis. . . . .	67
3.16	Design cube for three process and design corners: gate-oxide thickness ( $T_{ox}$ ), threshold voltage ( $V_{Th}$ ) and power supply ( $V_{DD}$ ). The superscripts – and + indicate the minimum and maximum values of the variable, respectively . . . . .	69
3.17	Monte Carlo simulation methodology to account for process and power supply variation. The inputs are statistical distributions of the variables. The processing engine is SPICE executing a large number of Monte Carlo runs. The outputs are statistical distributions of specification parameters of interest for the entire circuit, such as total gate leakage $I_{gate}$ , total subthreshold leakage $I_{sub}$ , total dynamic current $I_{dyn}$ and propagation delay $T_{PD}$ . . . . .	70



3.18	Effects of statistical process variation on gate-oxide leakage, subthreshold leakage, dynamic current and propagation delay in a two-input NAND gate. It is observed that variability of gate-oxide leakage, subthreshold leakage, dynamic current is lognormal in nature and that variability of propagation delay is normal in nature. A load equal to 10 times the pMOS gate capacitance has been added to the output of the gate [120] . . . . .	71
3.19	Three levels of abstraction in which data path components are realized using two-input NAND gates . . . . .	72
3.20	Nominal results showing individual components of power consumption for different output corners. It may be noted that the total current values are reduced and that the proportions of different components in the total current have changed. Only two corners are shown for brevity. The corners referred to in this figure correspond to those of Fig. 3.16 . . . . .	74
3.21	Variation of gate-oxide leakage current with respect to gate-oxide thickness . . . . .	75
3.22	A simple combinational logic circuit. The fractions indicate the probabilities that the indicated node will be at logic level "1" . . . . .	77
3.23	A NAND implementation of a half-adder. The numbers indicate the probabilities that the indicated net will be at logic level "1" . . . . .	77
4.1	Typical processor cooling setup . . . . .	83
4.2	Mobile platform power breakdown . . . . .	83
4.3	Desktop platform power breakdown . . . . .	84
4.4	Energy delay trade-off in design . . . . .	84
4.5	Accuracy and analysis time trade-off for various design abstractions . . . . .	85
4.6	Simulation flow for RTL power simulation by WattWatcher tool . . . . .	87
4.7	Macro-modeling simulation flow . . . . .	89
4.8	Quick synthesis flow . . . . .	90
4.9	Normalized power versus performance . . . . .	90
4.10	Capacitance versus gate voltage . . . . .	93
4.11	Stack effect on subthreshold current . . . . .	96
4.12	Power gating in cache . . . . .	99
4.13	Power distribution by units . . . . .	100
4.14	Where in a register file is the power? . . . . .	109
4.15	Local read bit line . . . . .	111
4.16	High-level model of the register file . . . . .	112
4.17	Type I RAM cell, single-ended read/single-ended write . . . . .	113
4.18	Type II RAM cell, single-ended read/dual-ended write . . . . .	113
4.19	Type I segment driver, dual pre-charge clock/SDL out . . . . .	114
4.20	Type I segment driver, dual pre-charge clock/SDL out . . . . .	114
4.21	Global bit-line pre-charge . . . . .	115

4.22	Write bit-line pre-charge unit . . . . .	115
4.23	Cell width data for existing register files . . . . .	116
4.24	Cell height data for existing register files . . . . .	116
4.25	Read path in a typical array logic structure . . . . .	119
4.26	Cache segmentation structure . . . . .	121
4.27	Banked cache structure for smaller bit/word lines and bank-specific power gating . . . . .	121
4.28	Typical cache decoder circuitry in a cache core . . . . .	122
4.29	Component of a typical array structure . . . . .	124
4.30	Typical six-transistor cache SRAM cell . . . . .	125
4.31	Typical write circuitry in the cache core . . . . .	126
4.32	Leakage versus temperature . . . . .	129
5.1	Power dissipation or consumption profile of CMOS circuits . . . . .	132
5.2	Driving factors for low-power circuit design . . . . .	133
5.3	Reduction mechanisms at different levels of design abstractions . . . . .	138
5.4	Multiple voltage–frequency versus variable voltage–frequency . . . . .	140
5.5	Variable voltage processor operation: voltage versus frequency [280] . . . . .	141
5.6	Data flow graph in three modes of operation . . . . .	144
5.7	Energy versus peak power efficient schedule . . . . .	145
5.8	Voltage-level converter circuit design and simulation waveform . . . . .	147
5.9	Dynamic frequency generation using dynamic clocking unit . . . . .	148
5.10	Dynamic clocking unit design – 1 . . . . .	149
5.11	Dynamic clocking unit design – 2 . . . . .	150
5.12	Multiple threshold voltage (MTCMOS), variable threshold voltage (VTCMOS), and dynamic threshold voltage (DTCMOS) CMOS techniques for subthreshold leakage reduction . . . . .	152
5.13	Leakage control transistor (LECTOR) technique for subthreshold leakage reduction . . . . .	153
5.14	<b>(a)</b> A NAND logic gate for the dual thickness approach that uses devices with either higher $T_{ox}$ or lower $T_{ox}$ . <b>(b)</b> A NAND logic gate for dual dielectric approach that uses devices with either higher $K$ or lower $K$ . . . . .	154
5.15	Operation reduction for switching capacitance reduction to reduce dynamic power or energy . . . . .	155
5.16	Operation substitution for switching capacitance reduction to reduce dynamic power or energy . . . . .	156
5.17	Example data path – the baseline . . . . .	157
5.18	Example data path – the parallel alternative . . . . .	158
5.19	Example data path – the pipelining alternative . . . . .	159
5.20	Guarded evaluation using operand Isolation . . . . .	159
5.21	Precomputation-based power reduction . . . . .	160
5.22	Clock gating for power reduction . . . . .	161

6.1	Level converters needed for stepping up signal . . . . .	164
6.2	Example DFG for MVDFC . . . . .	172
6.3	Example DFG (for RC2) (MVMC) . . . . .	174
6.4	Reduction for different benchmarks expressed as average percentage . . . . .	176
6.5	HAL differential equation solver (with ASAP labels) . . . . .	177
6.6	Schedules obtained for HAL benchmark for different time constraints using TC-DFC . . . . .	182
6.7	Final schedule of FIR filter DFG (using RC-DFC) . . . . .	188
6.8	Average energy and EDP reduction for benchmarks . . . . .	189
7.1	Example of DFG for MVDFC operation for resource constraint RC1: two multipliers at 3.3 V, one multiplier at 5.0 V, one ALU at 3.3 V and one ALU operating at 5.0 V . . . . .	208
7.2	Example of DFG for MVMC operation for resource constraint RC1: two multipliers at 3.3 V, one multiplier at 5.0 V, one ALU at 3.3 V and one ALU operating at 5.0 V . . . . .	211
7.3	Average peak power and PDP reductions for benchmark circuits . . . . .	214
7.4	Example of DFG for resource constraint RC3 using MVDFC . . . . .	217
7.5	Example of DFG for resource constraint RC3 using MVMC . . . . .	219
7.6	Average reduction for different benchmarks circuits . . . . .	221
8.1	Average reduction for selected benchmarks using CPF model 1 . . . . .	237
8.2	Cycle power consumptions for selected resource constraints – RC1 and RC2 . . . . .	239
8.3	Cycle power consumptions for selected resource constraints – RC3 and RC4 . . . . .	240
8.4	Average reduction for selected benchmarks using CPF model 2 . . . . .	241
8.5	Mobility graph and final schedule for the example of DFG for RC5 using MVDFC . . . . .	253
8.6	Mobility graph and final schedule for the example of DFG for RC5 using MVMC . . . . .	254
8.7	Average percentage reductions for benchmarks using CPF-MVDFC . . . . .	255
8.8	Average percentage reductions for benchmarks using CPF-MVMC . . . . .	256
8.9	Power profile for benchmark for different resource constraints, RC1 and RC2 . . . . .	257
8.10	Power profile for benchmark for different resource constraints, RC3 and RC4 . . . . .	258
9.1	High-level synthesis framework with simultaneous scheduling and binding . . . . .	264
9.2	Flow of the simultaneous scheduling and binding algorithm . . . . .	266

9.3	Illustration of algorithm 1 for simultaneous scheduling and binding for tunneling current reduction for IIR benchmark for time constraint $T_F = 1.0$ . The resource constraint is assumed to be 2 $T_{\text{oxH}}$ multipliers, 2 $T_{\text{oxL}}$ multipliers, 1 $T_{\text{oxH}}$ adder and 1 $T_{\text{oxL}}$ adder . . . . .	269
9.4	Bar charts showing percentage reduction in gate-oxide tunneling current for three cases: (i) 1.4–1.5 nm, (ii) 1.4–1.6 nm and (iii) 1.4–1.7 nm for algorithm 1 . . . . .	270
9.5	Average percentage reduction in gate leakage using dual- $K$ approach for various benchmarks circuits . . . . .	274

# List of Tables

4.1	Process variations . . . . .	103
5.1	How much we save by varying voltage ( $V_{DD}$ ) and/or frequency ( $f$ ) . . . . .	142
6.1	Notations used in the description . . . . .	166
6.2	Notations used in ILP formulations . . . . .	168
6.3	Savings for various scheduling schemes . . . . .	175
6.4	List of functions used in the TC-DFC algorithm . . . . .	179
6.5	List of variables and data structures used in the TC-DFC algorithm description . . . . .	179
6.6	TC-DFC frequency selection: from left $\rightarrow$ right . . . . .	180
6.7	Vertex priority list . . . . .	180
6.8	Cycle priority list : $T_{con} \approx 2 \times T_{cp}$ or $1.75 \times T_{cp}$ . . . . .	181
6.9	Cycle priority list : $T_{con} \approx 1.5 \times T_{cp}$ . . . . .	181
6.10	Frequency selection (from left to right in each step) . . . . .	183
6.11	Resource look-up table (from left to right) . . . . .	183
6.12	List of functions used in the RC-DFC algorithm . . . . .	185
6.13	List of variables and data structures used in the RC-DFC algorithm description . . . . .	185
6.14	Savings for various resource-constrained scheduling . . . . .	190
6.15	Savings for various time-constrained scheduling cases . . . . .	190
6.16	Data path scheduling schemes using multiple supply voltages . . . . .	195
6.17	High-level synthesis schemes using switching activity reduction . . . . .	200
7.1	Notations used in the description . . . . .	202
7.2	Notations used in ILP formulations . . . . .	204
7.3	Peak power reduction for various scheduling schemes . . . . .	214
7.4	Peak and average power reduction for various scheduling schemes . . . . .	222

- 7.5 Relative performance of various schemes proposed for peak power minimization ..... 223
- 8.1 List of notations and terminology used in CPF modeling ..... 226
- 8.2 List of variables used in ILP formulations ..... 247
- 9.1 Notations used in algorithm description ..... 267

# Acronym Definition

ACO	Ant colony optimization
ACS	Ant colony scheduling
AF	Activity factor
AHD	Average Hamming distance
ALAP	As late as possible
ALU	Arithmetic logic unit
AMPL	A mathematical programming language
ARF	Autoregressive filter
ARMA	Autoregressive moving average
ASAP	As soon as possible
ASIC	Application-specific integrated circuit
ATG	Automatic test generation
BFS	Breadth first search
BN	Boolean network
BPF	Band-pass filter
BSIM	Berkeley short channel insulated gate model
BTBT	Band-to-band tunneling
CAD	Computer-aided design
CCLC	Cross-coupled level converter
CDFG	Control data flow graph
CFG	Control flow graph
CG	Compatibility graph
CLT	Central limit theorem
CMOS	Complementary metal oxide semiconductor
CMP	Chemical mechanical polishing
CPF	Cycle power function
CPLD	Complex programmable logic device
CPU	Central processing unit
CVD	Chemical vapor deposition
DCN	Diffusion-connected network
DCT	Discrete cosine transform
DCU	Dynamic clocking unit
DCVS	Differential cascade voltage switch
DFC	Dynamic frequency clocking
DFG	Data flow graph

DFM	Design for manufacturing
DG	Distribution graph
DIBL	Drain-induced barrier lowering
DPCD	Dynamic programmable clock divider
DPM	Dynamic power management
DSP	Digital signal processing
DT	Dynamic threshold
DVS	Dynamic voltage scaling
DWT	Discrete wavelet transform
EDA	Electronic design automation
EDP	Energy delay product
ESL	Electronic system level
ESTG	Extended state transition graph
EFW	Elliptic wave filter
FDS	Force directed scheduling
FET	Field effect transistor
FFT	Fast Fourier transform
FIFO	First in first out
FIR	Finite impulse response
FN	Fowler–Nordheim
FPGA	Field programmable gate array
FSM	Finite state machine
FU	Functional unit
GAS	Genetic algorithm scheduling
GDS	Generalized data stream
GIDL	Gate-induced drain leakage
GIS	Geographic information system
GTS	Game-theory (based) scheduling
HCDG	Hierarchical conditional dependency graph
HDL	Hardware description language
HL	High to low
IC	Integrated circuit
IDCT	Inverse discrete cosine transform
IEEE	Institute of Electrical and Electronics Engineers
IIR	Infinite impulse response
ILP	Integer linear programming
I/O	Input/output
IP	Intellectual property
IR	Infrared
ITRS	International Technology Roadmap for Semiconductors
LBS	List-based scheduling
LECTOR	Leakage control transistor
LCT	Leakage control transistor
LH	Low to high
LP	Linear programming



LUT	Look-up table
MC	Multicycling
MCM	Monte Carlo method
MESVS	Minimum energy schedule with voltage selection
M $\kappa$	Multiple dielectric
MMV	MPEG motion vectors
MOSFET	Metal oxide semiconductor field effect transistor
MOVER	Multiple operating voltage energy reduction
MOX	Multiple oxide
MPEG	Moving Picture Experts Group
MT	Multiple threshold
MU	Memory unit
MV	Multiple supply voltage
MVDFC	Multiple supply voltage and dynamic frequency clocking
MVMC	Multiple supply voltage and multicycling
MVSF	Multiple supply voltage and single frequency
MWIS	Maximum weight independent set
Nano-CMOS	Nanoscale CMOS
NFA	Nondeterministic finite automata
NBTI	Negative bias temperature instability
NMOS	Metal oxide semiconductor, n type
NoC	Network on a chip
NOP	No operation
NP	Nondeterministic polynomial
ODT	On-die termination
PC	Personal computer
PCB	Printed circuit board
PDA	Personal digital assistant
PDP	Power delay product
PDSS	Profile-driven synthesis system
PI	Primary input
PLA	Programmable logic array
PLL	Phase-locked loop
PMOS	Metal oxide semiconductor, p type
PSB	Processor side bus
PSF	Process scaling factor
RAM	Random access memory
RC	Resource constrained (scheduling)
RF	Radio frequency
ROM	Read-only memory
RTL	Register transfer level
SAS	Simulated annealing scheduling
SDL	Set dominant latches
SoC	system on a chip
SOI	Silicon on insulator
SP	Signal probability

SRAM	Static random access memory
STG	State transition graph
SVDFC	Single supply voltage and dynamic frequency clocking
SVMC	Single supply voltage and multicycling
SVSF	Single supply voltage and single frequency
TC	Time constrained
TDP	Thermal design power
TLB	Translation look-aside buffer
TLM	Transaction-level modeling
TSS	Tabu search scheduling
UDFG	Unscheduled data flow graph
VHDL	VHSIC Hardware Description Language
VHSIC	Very-high-speed-integrated circuit
VLSI	Very-large-scale integration
VT	Variable threshold
WDF	Wave digital filter
YACC	“Yet another compiler” compiler