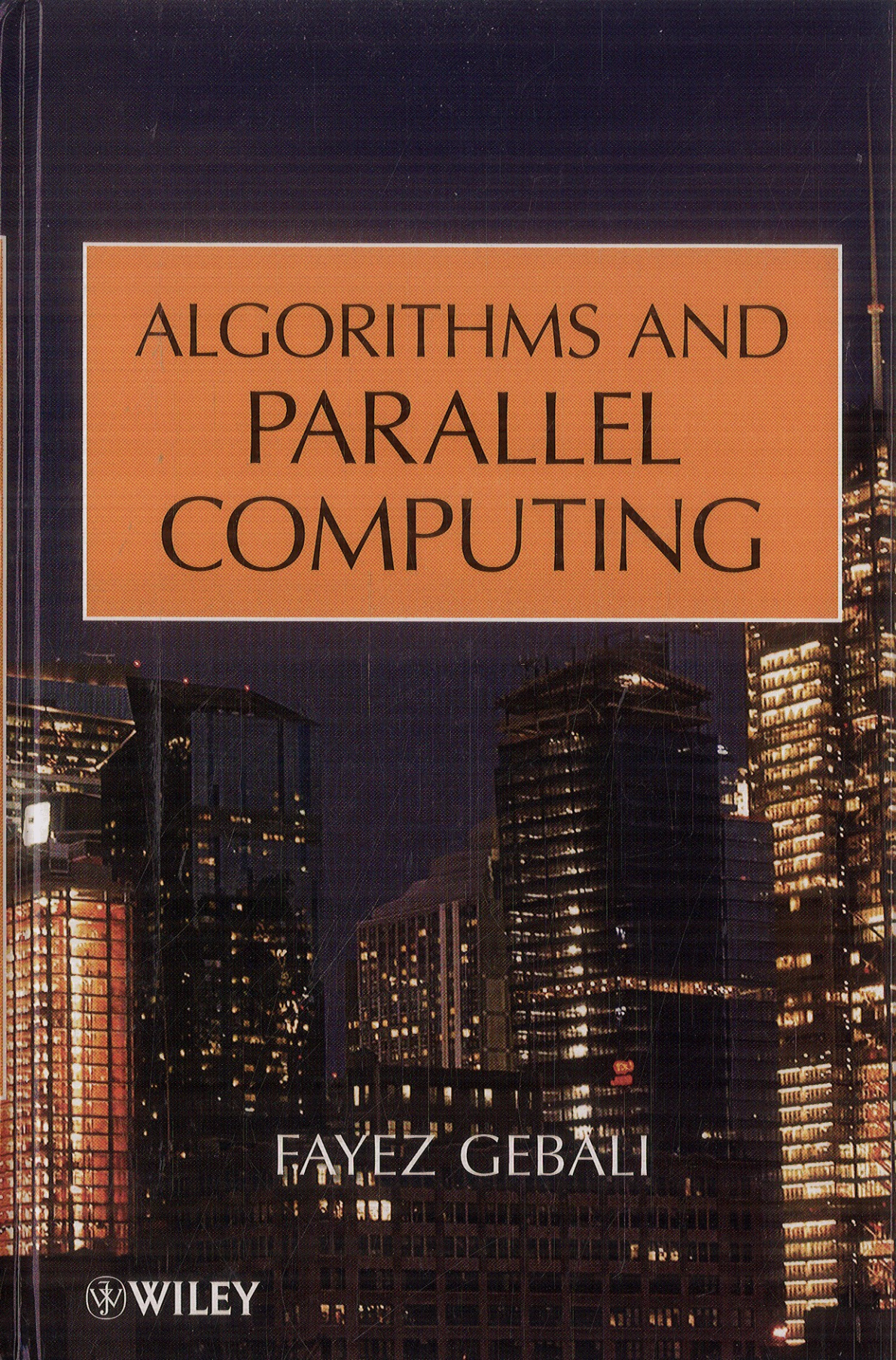


A nighttime photograph of a city skyline, likely New York City, with several skyscrapers illuminated by warm yellow and orange lights. The lights create a grid-like pattern of windows across the buildings. The sky is dark, and the overall atmosphere is urban and modern.

ALGORITHMS AND PARALLEL COMPUTING

FAYEZ GEBALI

 WILEY

Contents

Preface xiii

List of Acronyms xix

1 Introduction **1**

- 1.1 Introduction 1
- 1.2 Toward Automating Parallel Programming 2
- 1.3 Algorithms 4
- 1.4 Parallel Computing Design Considerations 12
- 1.5 Parallel Algorithms and Parallel Architectures 13
- 1.6 Relating Parallel Algorithm and Parallel Architecture 14
- 1.7 Implementation of Algorithms: A Two-Sided Problem 14
- 1.8 Measuring Benefits of Parallel Computing 15
- 1.9 Amdahl's Law for Multiprocessor Systems 19
- 1.10 Gustafson–Barsis's Law 21
- 1.11 Applications of Parallel Computing 22

2 Enhancing Uniprocessor Performance **29**

- 2.1 Introduction 29
- 2.2 Increasing Processor Clock Frequency 30
- 2.3 Parallelizing ALU Structure 30
- 2.4 Using Memory Hierarchy 33
- 2.5 Pipelining 39
- 2.6 Very Long Instruction Word (VLIW) Processors 44
- 2.7 Instruction-Level Parallelism (ILP) and Superscalar Processors 45
- 2.8 Multithreaded Processor 49

3 Parallel Computers **53**

- 3.1 Introduction 53
- 3.2 Parallel Computing 53
- 3.3 Shared-Memory Multiprocessors (Uniform Memory Access [UMA]) 54
- 3.4 Distributed-Memory Multiprocessor (Nonuniform Memory Access [NUMA]) 56

3.5	SIMD Processors	57
3.6	Systolic Processors	57
3.7	Cluster Computing	60
3.8	Grid (Cloud) Computing	60
3.9	Multicore Systems	61
3.10	SM	62
3.11	Communication Between Parallel Processors	64
3.12	Summary of Parallel Architectures	67

4 Shared-Memory Multiprocessors 69

4.1	Introduction	69
4.2	Cache Coherence and Memory Consistency	70
4.3	Synchronization and Mutual Exclusion	76

5 Interconnection Networks 83

5.1	Introduction	83
5.2	Classification of Interconnection Networks by Logical Topologies	84
5.3	Interconnection Network Switch Architecture	91

6 Concurrency Platforms 105

6.1	Introduction	105
6.2	Concurrency Platforms	105
6.3	Cilk++	106
6.4	OpenMP	112
6.5	Compute Unified Device Architecture (CUDA)	122

7 Ad Hoc Techniques for Parallel Algorithms 131

7.1	Introduction	131
7.2	Defining Algorithm Variables	133
7.3	Independent Loop Scheduling	133
7.4	Dependent Loops	134
7.5	Loop Spreading for Simple Dependent Loops	135
7.6	Loop Unrolling	135
7.7	Problem Partitioning	136
7.8	Divide-and-Conquer (Recursive Partitioning) Strategies	137
7.9	Pipelining	139

8 Nonserial-Parallel Algorithms 143

8.1	Introduction	143
8.2	Comparing DAG and DCG Algorithms	143
8.3	Parallelizing NSPA Algorithms Represented by a DAG	145

8.4	Formal Technique for Analyzing NSPAs	147
8.5	Detecting Cycles in the Algorithm	150
8.6	Extracting Serial and Parallel Algorithm Performance Parameters	151
8.7	Useful Theorems	153
8.8	Performance of Serial and Parallel Algorithms on Parallel Computers	156

9 z -Transform Analysis 159

9.1	Introduction	159
9.2	Definition of z -Transform	159
9.3	The 1-D FIR Digital Filter Algorithm	160
9.4	Software and Hardware Implementations of the z -Transform	161
9.5	Design 1: Using Horner's Rule for Broadcast Input and Pipelined Output	162
9.6	Design 2: Pipelined Input and Broadcast Output	163
9.7	Design 3: Pipelined Input and Output	164

10 Dependence Graph Analysis 167

10.1	Introduction	167
10.2	The 1-D FIR Digital Filter Algorithm	167
10.3	The Dependence Graph of an Algorithm	168
10.4	Deriving the Dependence Graph for an Algorithm	168
10.5	The Scheduling Function for the 1-D FIR Filter	171
10.6	Node Projection Operation	177
10.7	Nonlinear Projection Operation	179
10.8	Software and Hardware Implementations of the DAG Technique	180

11 Computational Geometry Analysis 185

11.1	Introduction	185
11.2	Matrix Multiplication Algorithm	185
11.3	The 3-D Dependence Graph and Computation Domain $\mathcal{D}$	186
11.4	The Facets and Vertices of $\mathcal{D}$	188
11.5	The Dependence Matrices of the Algorithm Variables	188
11.6	Nullspace of Dependence Matrix: The Broadcast Subdomain B	189
11.7	Design Space Exploration: Choice of Broadcasting versus Pipelining Variables	192
11.8	Data Scheduling	195
11.9	Projection Operation Using the Linear Projection Operator	200
11.10	Effect of Projection Operation on Data	205
11.11	The Resulting Multithreaded/Multiprocessor Architecture	206
11.12	Summary of Work Done in this Chapter	207

12 Case Study: One-Dimensional IIR Digital Filters	209
12.1 Introduction	209
12.2 The 1-D IIR Digital Filter Algorithm	209
12.3 The IIR Filter Dependence Graph	209
12.4 z-Domain Analysis of 1-D IIR Digital Filter Algorithm	216
13 Case Study: Two- and Three-Dimensional Digital Filters	219
13.1 Introduction	219
13.2 Line and Frame Wraparound Problems	219
13.3 2-D Recursive Filters	221
13.4 3-D Digital Filters	223
14 Case Study: Multirate Decimators and Interpolators	227
14.1 Introduction	227
14.2 Decimator Structures	227
14.3 Decimator Dependence Graph	228
14.4 Decimator Scheduling	230
14.5 Decimator DAG for $s_1 = [1 \ 0]$	231
14.6 Decimator DAG for $s_2 = [1 \ -1]$	233
14.7 Decimator DAG for $s_3 = [1 \ 1]$	235
14.8 Polyphase Decimator Implementations	235
14.9 Interpolator Structures	236
14.10 Interpolator Dependence Graph	237
14.11 Interpolator Scheduling	238
14.12 Interpolator DAG for $s_1 = [1 \ 0]$	239
14.13 Interpolator DAG for $s_2 = [1 \ -1]$	241
14.14 Interpolator DAG for $s_3 = [1 \ 1]$	243
14.15 Polyphase Interpolator Implementations	243
15 Case Study: Pattern Matching	245
15.1 Introduction	245
15.2 Expressing the Algorithm as a Regular Iterative Algorithm (RIA)	245
15.3 Obtaining the Algorithm Dependence Graph	246
15.4 Data Scheduling	247
15.5 DAG Node Projection	248
15.6 DESIGN 1: Design Space Exploration When $s = [1 \ 1]^t$	249
15.7 DESIGN 2: Design Space Exploration When $s = [1 \ -1]^t$	252
15.8 DESIGN 3: Design Space Exploration When $s = [1 \ 0]^t$	253
16 Case Study: Motion Estimation for Video Compression	255
16.1 Introduction	255
16.2 FBMA's	256

16.3	Data Buffering Requirements	257
16.4	Formulation of the FBMA	258
16.5	Hierarchical Formulation of Motion Estimation	259
16.6	Hardware Design of the Hierarchy Blocks	261

17 Case Study: Multiplication over $GF(2^m)$ **267**

17.1	Introduction	267
17.2	The Multiplication Algorithm in $GF(2^m)$	268
17.3	Expressing Field Multiplication as an RIA	270
17.4	Field Multiplication Dependence Graph	270
17.5	Data Scheduling	271
17.6	DAG Node Projection	273
17.7	Design 1: Using $d_1 = [1\ 0]'$	275
17.8	Design 2: Using $d_2 = [1\ 1]'$	275
17.9	Design 3: Using $d_3 = [1\ -1]'$	277
17.10	Applications of Finite Field Multipliers	277

18 Case Study: Polynomial Division over $GF(2)$ **279**

18.1	Introduction	279
18.2	The Polynomial Division Algorithm	279
18.3	The LFSR Dependence Graph	281
18.4	Data Scheduling	282
18.5	DAG Node Projection	283
18.6	Design 1: Design Space Exploration When $s_1 = [1\ -1]$	284
18.7	Design 2: Design Space Exploration When $s_2 = [1\ 0]$	286
18.8	Design 3: Design Space Exploration When $s_3 = [1\ -0.5]$	289
18.9	Comparing the Three Designs	291

19 The Fast Fourier Transform **293**

19.1	Introduction	293
19.2	Decimation-in-Time FFT	295
19.3	Pipeline Radix-2 Decimation-in-Time FFT Processor	298
19.4	Decimation-in-Frequency FFT	299
19.5	Pipeline Radix-2 Decimation-in-Frequency FFT Processor	303

20 Solving Systems of Linear Equations **305**

20.1	Introduction	305
20.2	Special Matrix Structures	305
20.3	Forward Substitution (Direct Technique)	309
20.4	Back Substitution	312
20.5	Matrix Triangularization Algorithm	312
20.6	Successive over Relaxation (SOR) (Iterative Technique)	317
20.7	Problems	321

xii Contents

**21 Solving Partial Differential Equations Using Finite
Difference Method** **323**

- 21.1 Introduction 323
21.2 FDM for 1-D Systems 324

References 331

Index 337