

---

# SYNTHESIS AND OPTIMIZATION OF DIGITAL CIRCUITS

---

**Giovanni De Micheli**

*Stanford University*

Technische Universität Darmstadt	
FACHBEREICH INFORMATIK	
<b>B I B L I O T H E K</b>	
Inventar-Nr.:	<u>MO7-00156</u>
Sachgebiete:	<u>Hardware</u>
Standort:	<u>B. 7 / De Mi.</u>

**McGraw-Hill, Inc.**

New York St. Louis San Francisco Auckland Bogotá Caracas  
Lisbon London Madrid Mexico City Milan Montreal  
New Delhi San Juan Singapore Sydney Tokyo Toronto

## Preface

xv

## Part I Circuits and Models

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Microelectronics	3
1.2	Semiconductor Technologies and Circuit Taxonomy	5
1.3	Microelectronic Design Styles	6
1.4	Design of Microelectronic Circuits	12
1.5	Computer-Aided Synthesis and Optimization	14
1.5.1	Circuit Models	15
1.5.2	Synthesis	15
1.5.3	Optimization	21
1.6	Organization of the Book	27
1.7	Related Problems	28
1.7.1	A Closer Look at Physical Design Problems	30
1.7.2	Computer-Aided Simulation	31
1.7.3	Computer-Aided Verification	32
1.7.4	Testing and Design for Testability	33
1.8	Synthesis and Optimization: A Historic Perspective	33
1.9	References	35
<b>2</b>	<b>Background</b>	<b>36</b>
2.1	Notation	36
2.2	Graphs	37
2.2.1	Undirected Graphs	38
2.2.2	Directed Graphs	39
2.2.3	Perfect Graphs	40
2.3	Combinatorial Optimization	42
2.3.1	Decision and Optimization Problems	42
2.3.2	Algorithms	43

2.3.3	Tractable and Intractable Problems	44
2.3.4	Fundamental Algorithms	46
2.4	Graph Optimization Problems and Algorithms	53
2.4.1	The Shortest and Longest Path Problems	54
2.4.2	Vertex Cover	59
2.4.3	Graph Coloring	61
2.4.4	Clique Covering and Clique Partitioning	64
2.5	Boolean Algebra and Applications	67
2.5.1	Boolean Functions	68
2.5.2	Representations of Boolean Functions	72
2.5.3	Satisfiability and Cover	85
2.6	Perspectives	94
2.7	References	95
2.8	Problems	96
<b>3</b>	<b>Hardware Modeling</b>	<b>97</b>
3.1	Introduction	97
3.2	Hardware Modeling Languages	98
3.2.1	Distinctive Features of Hardware Languages	100
3.2.2	Structural Hardware Languages	102
3.2.3	Behavioral Hardware Languages	103
3.2.4	HDLs Used for Synthesis	108
3.3	Abstract Models	115
3.3.1	Structures	115
3.3.2	Logic Networks	116
3.3.3	State Diagrams	118
3.3.4	Data-flow and Sequencing Graphs	119
3.4	Compilation and Behavioral Optimization	126
3.4.1	Compilation Techniques	127
3.4.2	Optimization Techniques	131
3.5	Perspectives	136
3.6	References	137
3.7	Problems	138

## Part II Architectural-Level Synthesis and Optimization

---

<b>4</b>	<b>Architectural Synthesis</b>	<b>141</b>
4.1	Introduction	141
4.2	Circuit specifications for Architectural Synthesis	143
4.2.1	Resources	143
4.2.2	Constraints	145
4.3	The Fundamental Architectural Synthesis Problems	146
4.3.1	The Temporal Domain: Scheduling	146
4.3.2	The Spatial Domain: Binding	150
4.3.3	Hierarchical Models	153
4.3.4	The Synchronization Problem	154
4.4	Area and Performance Estimation	155
4.4.1	Resource-Dominated Circuits	156

4.4.2	General Circuits	156
4.5	Strategies for Architectural Optimization	158
4.5.1	Area/Latency Optimization	159
4.5.2	Cycle-Time/Latency Optimization	160
4.5.3	Cycle-Time/Area Optimization	163
4.6	Data-Path Synthesis	163
4.7	Control-Unit Synthesis	166
4.7.1	Microcoded Control Synthesis for Non-Hierarchical Sequencing Graphs with Data-Independent Delay Operations	167
4.7.2	Microcoded Control Optimization Techniques*	168
4.7.3	Hard-Wired Control Synthesis for Non-Hierarchical Sequencing Graphs with Data-Independent Delay Operations	170
4.7.4	Control Synthesis for Hierarchical Sequencing Graphs with Data-Independent Delay Operations	171
4.7.5	Control Synthesis for Unbounded-Latency Sequencing Graphs*	174
4.8	Synthesis of Pipelined Circuits	178
4.9	Perspectives	181
4.10	References	182
4.11	Problems	183
<b>5</b>	<b>Scheduling Algorithms</b>	<b>185</b>
5.1	Introduction	185
5.2	A Model for the Scheduling Problems	186
5.3	Scheduling without Resource Constraints	187
5.3.1	Unconstrained Scheduling: The ASAP Scheduling Algorithm	188
5.3.2	Latency-Constrained Scheduling: The ALAP Scheduling Algorithm	188
5.3.3	Scheduling Under Timing Constraints	190
5.3.4	Relative Scheduling*	193
5.4	Scheduling with Resource Constraints	198
5.4.1	The Integer Linear Programming Model	198
5.4.2	Multiprocessor Scheduling and Hu's Algorithm	202
5.4.3	Heuristic Scheduling Algorithms: List Scheduling	207
5.4.4	Heuristic Scheduling Algorithms: Force-directed Scheduling*	211
5.4.5	Other Heuristic Scheduling Algorithms*	215
5.5	Scheduling Algorithms for Extended Sequencing Models*	216
5.5.1	Scheduling Graphs with Alternative Paths*	216
5.6	Scheduling Pipelined Circuits*	218
5.6.1	Scheduling with Pipelined Resources*	220
5.6.2	Functional Pipelining*	222
5.6.3	Loop Folding*	224
5.7	Perspectives	225

5.8	References	225
5.9	Problems	226
<b>6</b>	<b>Resource Sharing and Binding</b>	<b>229</b>
6.1	Introduction	229
6.2	Sharing and Binding for Resource-Dominated Circuits	230
6.2.1	Resource Sharing in Non-Hierarchical Sequencing Graphs	233
6.2.2	Resource Sharing in Hierarchical Sequencing Graphs	237
6.2.3	Register Sharing	240
6.2.4	Multi-Port Memory Binding	243
6.2.5	Bus Sharing and Binding	245
6.3	Sharing and Binding for General Circuits*	245
6.3.1	Unconstrained Minimum-Area Binding*	246
6.3.2	Performance-Constrained and Performance-Directed Binding*	249
6.3.3	Considerations for Other Binding Problems*	250
6.4	Concurrent Binding and Scheduling	250
6.5	Resource Sharing and Binding for Non-Scheduled Sequencing Graphs	252
6.5.1	Sharing and Binding for Non-Scheduled Models	253
6.5.2	Size of the Design Space*	255
6.6	The Module Selection Problem	257
6.7	Resource Sharing and Binding for Pipelined Circuits	260
6.8	Sharing and Structural Testability*	262
6.9	Perspectives	263
6.10	References	264
6.11	Problems	265

## Part III Logic-Level Synthesis and Optimization

---

<b>7</b>	<b>Two-Level Combinational Logic Optimization</b>	<b>269</b>
7.1	Introduction	269
7.2	Logic Optimization Principles	270
7.2.1	Definitions	272
7.2.2	Exact Logic Minimization	277
7.2.3	Heuristic Logic Minimization	283
7.2.4	Testability Properties	286
7.3	Operations on Two-Level Logic Covers	288
7.3.1	The Positional-Cube Notation	288
7.3.2	Functions with Multiple-Valued Inputs	289
7.3.3	List-Oriented Manipulation	291
7.3.4	The Unate Recursive Paradigm	294
7.4	Algorithms for Logic Minimization	304
7.4.1	Expand	304
7.4.2	Reduce*	308
7.4.3	Irredundant*	310
7.4.4	Essentials*	313
7.4.5	The ESPRESSO Minimizer	315

7.5	Symbolic Minimization and Encoding Problems	318
7.5.1	Input Encoding	319
7.5.2	Output Encoding*	327
7.5.3	Output Polarity Assignment	333
7.6	Minimization of Boolean Relations*	334
7.7	Perspectives	338
7.8	References	339
7.9	Problems	341
<b>8</b>	<b>Multiple-Level Combinational Logic Optimization</b>	<b>343</b>
8.1	Introduction	343
8.2	Models and Transformations for Combinational Networks	345
8.2.1	Optimization of Logic Networks	348
8.2.2	Transformations for Logic Networks	350
8.2.3	The Algorithmic Approach to Multiple-Level Logic Optimization	356
8.3	The Algebraic Model	360
8.3.1	Substitution	363
8.3.2	Extraction and Algebraic Kernels	365
8.3.3	Decomposition	378
8.4	The Boolean Model	380
8.4.1	<i>Don't Care</i> Conditions and Their Computation	380
8.4.2	Boolean Simplification and Substitution	396
8.4.3	Other Optimization Algorithms Using Boolean Transformations*	408
8.5	Synthesis of Testable Networks	415
8.6	Algorithms for Delay Evaluation and Optimization	418
8.6.1	Delay Modeling	418
8.6.2	Detection of False Paths*	421
8.6.3	Algorithms and Transformations for Delay Optimization	426
8.7	Rule-Based Systems for Logic Optimization	433
8.8	Perspectives	435
8.9	References	436
8.10	Problems	439
<b>9</b>	<b>Sequential Logic Optimization</b>	<b>441</b>
9.1	Introduction	441
9.2	Sequential Circuit Optimization Using State-Based Models	443
9.2.1	State Minimization	444
9.2.2	State Encoding	449
9.2.3	Other Optimization Methods and Recent Developments*	455
9.3	Sequential Circuit Optimization Using Network Models	458
9.3.1	Retiming	462
9.3.2	Synchronous Circuit Optimization by Retiming and Logic Transformations	475
9.3.3	<i>Don't Care</i> Conditions in Synchronous Networks	481
9.4	Implicit Finite-State Machine Traversal Methods	490
9.4.1	State Extraction	491
9.4.2	Implicit State Minimization*	494
9.5	Testability Considerations for Synchronous Circuits	495

9.6	Perspectives	498
9.7	References	500
9.8	Problems	502
<b>10</b>	<b>Cell-Library Binding</b>	<b>504</b>
10.1	Introduction	504
10.2	Problem Formulation and Analysis	505
10.3	Algorithms for Library Binding	509
10.3.1	Covering Algorithms Based on Structural Matching	512
10.3.2	Covering Algorithms Based on Boolean Matching	526
10.3.3	Covering Algorithms and Polarity Assignment	530
10.3.4	Concurrent Logic Optimization and Library Binding*	533
10.3.5	Testability Properties of Bound Networks	536
10.4	Specific Problems and Algorithms for Library Binding	537
10.4.1	Look-Up Table FPGAs	538
10.4.2	Anti-Fuse-Based FPGAs	541
10.5	Rule-Based Library Binding	544
10.5.1	Comparisons of Algorithmic and Rule-Based Library Binding	545
10.6	Perspectives	546
10.7	References	546
10.8	Problems	548

## Part IV Conclusions

---

<b>11</b>	<b>State of the Art and Future Trends</b>	<b>551</b>
11.1	The State of the Art in Synthesis	551
11.2	Synthesis Systems	553
11.2.1	Production-Level Synthesis Systems	554
11.2.2	Research Synthesis Systems	555
11.2.3	Achievements and Unresolved Issues	558
11.3	The Growth of Synthesis in the Near and Distant Future	559
11.3.1	System-Level Synthesis	561
11.3.2	Hardware-Software Co-Design	562
11.4	Envoy	565
11.5	References	565
	<b>Index</b>	<b>567</b>