数字集成电路设计

[数字集成电路设计_下载链接1_](#)

著者:Hubert Kaeslin

出版者:人民邮电出版社

出版时间:201004

装帧:平装

isbn:9787115223586

《数字集成电路设计:从VLSI体系结构到CMOS制造(英文版)》从架构与算法讲起，介绍

了功能验证、VHDL建模、同步电路设计、异步数据获取、能耗与散热、信号完整性、物理设计、设计验证等必备技术，还讲解了VLSI经济运作与项目管理，并简单阐释了CMOS技术的基础知识，全面覆盖了数字集成电路的整个设计开发过程。

《数字集成电路设计:从VLSI体系结构到CMOS制造(英文版)》既可作为高等院校微电子、电子技术等相关专业高年级师生和研究生的参考教材，也可供半导体行业工程师参考。

作者介绍:

Hubert Kaeslin
1985年于瑞士苏黎世联邦理工学院获得博士学位，现为该校微电子设计中心的负责人，具有20多年教授VLSI的丰富经验。

目录:
Chapter 1 Introduction to Microelectronics 1
1.1 Economic impact 1
1.2 Concepts and terminology 4
1.2.1 The Guinness book of records point of view 4
1.2.2 The marketing point of view 5
1.2.3 The fabrication point of view 6
1.2.4 The design engineer's point of view 10
1.2.5 The business point of view 17
1.3 Design flow in digital VLSI 18
1.3.1 The Y-chart, a map of digital electronic systems 18
1.3.2 Major stages in VLSI design 19
1.3.3 Cell libraries 28
1.3.4 Electronic design automation software 29
1.4 Field-programmable logic 30
1.4.1 Configuration technologies 30
1.4.2 Organization of hardware resources 32
1.4.3 Commercial products 35
1.5 Problems 37
1.6 Appendix I: A brief glossary of logic families 38
1.7 Appendix II: An illustrated glossary of circuit-related terms 40
Chapter 2 From Algorithms to Architectures 44
2.1 The goals of architecture design 44
2.1.1 Agenda 45
2.2 The architectural antipodes 45
2.2.1 What makes an algorithm suitable for a dedicated VLSI architecture? 50
2.2.2 There is plenty of land between the architectural antipodes 53
2.2.3 Assemblies of general-purpose and dedicated processing units 54
2.2.4 Coprocessors 55
2.2.5 Application-specific instruction set processors 55
2.2.6 Configurable computing 58
2.2.7 Extendable instruction set processors 59
2.2.8 Digest 60
2.3 A transform approach to VLSI architecture design 61
2.3.1 There is room for remodelling in the algorithmic domain 62
2.3.2 ...and there is room in the architectural domain 64
2.3.3 Systems engineers and VLSI designers must collaborate 64
2.3.4 A graph-based formalism for describing processing algorithms 65
2.3.5 The isomorphic architecture 66

2.3.6 Relative merits of architectural alternatives 67
2.3.7 Computation cycle versus clock period 69
2.4 Equivalence transforms for combinational computations 70
2.4.1 Common assumptions 71
2.4.2 Iterative decomposition 72
2.4.3 Pipelining 75
2.4.4 Replication 79
2.4.5 Time sharing 81
2.4.6 Associativity transform 86
2.4.7 Other algebraic transforms 87
2.4.8 Digest 87
2.5 Options for temporary storage of data 89
2.5.1 Data access patterns 89
2.5.2 Available memory configurations and area occupation 89
2.5.3 Storage capacities 90
2.5.4 Wiring and the costs of going off-chip 91
2.5.5 Latency and timing 91
2.5.6 Digest 92
2.6 Equivalence transforms for nonrecursive computations 93
2.6.1 Retiming 94
2.6.2 Pipelining revisited 95
2.6.3 Systolic conversion 97
2.6.4 Iterative decomposition and time-sharing revisited 98
2.6.5 Replication revisited 98
2.6.6 Digest 99
2.7 Equivalence transforms for recursive computations 99
2.7.1 The feedback bottleneck 100
2.7.2 Unfolding of first-order loops 101
2.7.3 Higher-order loops 103
2.7.4 Time-variant loops 105
2.7.5 Nonlinear or general loops 106
2.7.6 Pipeline interleaving is not an equivalence transform 109
2.7.7 Digest 111
2.8 Generalizations of the transform approach 112
2.8.1 Generalization to other levels of detail 112
2.8.2 Bit-serial architectures 113
2.8.3 Distributed arithmetic 116
2.8.4 Generalization to other algebraic structures 118
2.8.5 Digest 121
2.9 Conclusions 122
2.9.1 Summary 122
2.9.2 The grand architectural alternatives from an energy point of view 124
2.9.3 A guide to evaluating architectural alternatives 126
2.10 Problems 128
2.11 Appendix I: A brief glossary of algebraic structures 130
2.12 Appendix II: Area and delay figures of VLSI subfunctions 133
Chapter 3 Functional Verification 136
3.1 How to establish valid functional specifications 137
3.1.1 Formal specification 138
3.1.2 Rapid prototyping 138
3.2 Developing an adequate simulation strategy 139
3.2.1 What does it take to uncover a design flaw during simulation? 139
3.2.2 Stimulation and response checking must occur automatically 140
3.2.3 Exhaustive verification remains an elusive goal 142

3.2.4 All partial verification techniques have their pitfalls 143
3.2.5 Collecting test cases from multiple sources helps 150
3.2.6 Assertion-based verification helps 150
3.2.7 Separating test development from circuit design helps 151
3.2.8 Virtual prototypes help to generate expected responses 153
3.3 Reusing the same functional gauge throughout the entire design cycle 153
3.3.1 Alternative ways to handle stimuli and expected responses 155
3.3.2 Modular testbench design 156
3.3.3 A well-defined schedule for stimuli and responses 156
3.3.4 Trimming run times by skipping redundant simulation sequences 159
3.3.5 Abstracting to higher-level transactions on higher-level data 160
3.3.6 Absorbing latency variations across multiple circuit models 164
3.4 Conclusions 166
3.5 Problems 168
3.6 Appendix I: Formal approaches to functional verification 170
3.7 Appendix II: Deriving a coherent schedule for simulation and test 171
Chapter 4 Modelling Hardware with VHDL 175
4.1 Motivation 175
4.1.1 Why hardware synthesis? 175
4.1.2 What are the alternatives to VHDL? 176
4.1.3 What are the origins and aspirations of the IEEE 1076 standard? 176
4.1.4 Why bother learning hardware description languages? 179
4.1.5 Agenda 180
4.2 Key concepts and constructs of VHDL 180
4.2.1 Circuit hierarchy and connectivity 181
4.2.2 Concurrent processes and process interaction 185
4.2.3 A discrete replacement for electrical signals 192
4.2.4 An event-based concept of time for governing simulation 200
4.2.5 Facilities for model parametrization 211
4.2.6 Concepts borrowed from programming languages 216
4.3 Putting VHDL to service for hardware synthesis 223
4.3.1 Synthesis overview 223
4.3.2 Data types 224
4.3.3 Registers, finite state machines, and other sequential subcircuits 225
4.3.4 RAMs, ROMs, and other macrocells 231
4.3.5 Circuits that must be controlled at the netlist level 233
4.3.6 Timing constraints 234
4.3.7 Limitations and caveats for synthesis 238
4.3.8 How to establish a register transfer-level model step by step 238
4.4 Putting VHDL to service for hardware simulation 242
4.4.1 Ingredients of digital simulation 242
4.4.2 Anatomy of a generic testbench 242
4.4.3 Adapting to a design problem at hand 245
4.4.4 The VITAL modelling standard IEEE 1076.4 245
4.5 Conclusions 247
4.6 Problems 248
4.7 Appendix I: Books and Web Pages on VHDL 250
4.8 Appendix II: Related extensions and standards 251
4.8.1 Protected shared variables IEEE 1076a 251
4.8.2 The analog and mixed-signal extension IEEE 1076.1 252
4.8.3 Mathematical packages for real and complex numbers IEEE 1076.2 253
4.8.4 The arithmetic packages IEEE 1076.3 254
4.8.5 A language subset earmarked for synthesis IEEE 1076.6 254
4.8.6 The standard delay format (SDF) IEEE 1497 254

4.8.7 A handy compilation of type conversion functions 255
4.9 Appendix III: Examples of VHDL models 256
4.9.1 Combinational circuit models 256
4.9.2 Mealy, Moore, and Medvedev machines 261
4.9.3 State reduction and state encoding 268
4.9.4 Simulation testbenches 270
4.9.5 Working with VHDL tools from different vendors 285
Chapter 5 The Case for Synchronous Design 286
5.1 Introduction 286
5.2 The grand alternatives for regulating state changes 287
5.2.1 Synchronous clocking 287
5.2.2 Asynchronous clocking 288
5.2.3 Self-timed clocking 288
5.3 Why a rigorous approach to clocking is essential in VLSI 290
5.3.1 The perils of hazards 290
5.3.2 The pros and cons of synchronous clocking 291
5.3.3 Clock-as-clock-can is not an option in VLSI 293
5.3.4 Fully self-timed clocking is not normally an option either 294
5.3.5 Hybrid approaches to system clocking 294
5.4 The dos and don’ ts of synchronous circuit design 296
5.4.1 First guiding principle: Dissociate signal classes! 296
5.4.2 Second guiding principle: Allow circuits to settle before clocking! 298
5.4.3 Synchronous design rules at a more detailed level 298
5.5 Conclusions 306
5.6 Problems 306
5.7 Appendix: On identifying signals 307
5.7.1 Signal class 307
5.7.2 Active level 308
5.7.3 Signaling waveforms 309
5.7.4 Three-state capability 311
5.7.5 Inputs, outputs, and bidirectional terminals 311
5.7.6 Present state vs. next state 312
5.7.7 Syntactical conventions 312
5.7.8 A note on upper- and lower-case letters in VHDL 313
5.7.9 A note on the portability of names across EDA platforms 314
Chapter 6 Clocking of Synchronous Circuits 315
6.1 What is the difficulty in clock distribution? 315
6.1.1 Agenda 316
6.1.2 Timing quantities related to clock distribution 317
6.2 How much skew and jitter does a circuit tolerate? 317
6.2.1 Basics 317
6.2.2 Single-edge-triggered one-phase clocking 319
6.2.3 Dual-edge-triggered one-phase clocking 326
6.2.4 Symmetric level-sensitive two-phase clocking 327
6.2.5 Unsymmetric level-sensitive two-phase clocking 331
6.2.6 Single-wire level-sensitive two-phase clocking 334
6.2.7 Level-sensitive one-phase clocking and wave pipelining 336
6.3 How to keep clock skew within tight bounds 339
6.3.1 Clock waveforms 339
6.3.2 Collective clock buffers 340
6.3.3 Distributed clock buffer trees 343
6.3.4 Hybrid clock distribution networks 344
6.3.5 Clock skew analysis 345
6.4 How to achieve friendly input/output timing 346

6.4.1 Friendly as opposed to unfriendly I/O timing 346
6.4.2 Impact of clock distribution delay on I/O timing 347
6.4.3 Impact of PTV variations on I/O timing 349
6.4.4 Registered inputs and outputs 350
6.4.5 Adding artificial contamination delay to data inputs 350
6.4.6 Driving input registers from an early clock 351
6.4.7 Tapping a domain’ s clock from the slowest component therein 351
6.4.8 “Zero-delay” clock distribution by way of a DLL or PLL 352
6.5 How to implement clock gating properly 353
6.5.1 Traditional feedback-type registers with enable 353
6.5.2 A crude and unsafe approach to clock gating 354
6.5.3 A simple clock gating scheme that may work under certain conditions 355
6.5.4 Safe clock gating schemes 355
6.6 Summary 357
6.7 Problems 361
Chapter 7 Acquisition of Asynchronous Data 364
7.1 Motivation 364
7.2 The data consistency problem of vectored acquisition 366
7.2.1 Plain bit-parallel synchronization 366
7.2.2 Unit-distance coding 367
7.2.3 Suppression of crossover patterns 368
7.2.4 Handshaking 369
7.2.5 Partial handshaking 371
7.3 The data consistency problem of scalar acquisition 373
7.3.1 No synchronization whatsoever 373
7.3.2 Synchronization at multiple places 373
7.3.3 Synchronization at a single place 373
7.3.4 Synchronization from a slow clock 374
7.4 Metastable synchronizer behavior 374
7.4.1 Marginal triggering and how it becomes manifest 374
7.4.2 Repercussions on circuit functioning 378
7.4.3 A statistical model for estimating synchronizer reliability 379
7.4.4 Plesiochronous interfaces 381
7.4.5 Containment of metastable behavior 381
7.5 Summary 384
7.6 Problems 384
Chapter 8 Gate- and Transistor-Level Design 386
8.1 CMOS logic gates 386
8.1.1 The MOSFET as a switch 387
8.1.2 The inverter 388
8.1.3 Simple CMOS gates 396
8.1.4 Composite or complex gates 399
8.1.5 Gates with high-impedance capabilities 403
8.1.6 Parity gates 406
8.1.7 Adder slices 407
8.2 CMOS bistables 409
8.2.1 Latches 410
8.2.2 Function latches 412
8.2.3 Single-edge-triggered flip-flops 413
8.2.4 The mother of all flip-flops 415
8.2.5 Dual-edge-triggered flip-flops 417
8.2.6 Digest 418
8.3 CMOS on-chip memories 418
8.3.1 Static RAM 418

8.3.2 Dynamic RAM 423
8.3.3 Other differences and commonalities 424
8.4 Electrical CMOS contraptions 425
8.4.1 Snapper 425
8.4.2 Schmitt trigger 426
8.4.3 Tie-off cells 427
8.4.4 Filler cell or fillcap 428
8.4.5 Level shifters and input/output buffers 429
8.4.6 Digitally adjustable delay lines 429
8.5 Pitfalls 430
8.5.1 Busses and three-state nodes 430
8.5.2 Transmission gates and other bidirectional components 434
8.5.3 What do we mean by safe design? 437
8.5.4 Microprocessor interface circuits 438
8.5.5 Mechanical contacts 440
8.5.6 Conclusions 440
8.6 Problems 442
8.7 Appendix I: Summary on electrical MOSFET models 445
8.7.1 Naming and counting conventions 445
8.7.2 The Sah model 446
8.7.3 The Shichman–Hodges model 450
8.7.4 The alpha-power-law model 450
8.7.5 Second-order effects 452
8.7.6 Effects not normally captured by transistor models 455
8.7.7 Conclusions 456
8.8 Appendix II: The Bipolar Junction Transistor 457
Chapter 9 Energy Efficiency and Heat Removal 459
9.1 What does energy get dissipated for in CMOS circuits? 459
9.1.1 Charging and discharging of capacitive loads 460
9.1.2 Crossover currents 465
9.1.3 Resistive loads 467
9.1.4 Leakage currents 468
9.1.5 Total energy dissipation 470
9.1.6 CMOS voltage scaling 471
9.2 How to improve energy efficiency 474
9.2.1 General guidelines 474
9.2.2 How to reduce dynamic dissipation 476
9.2.3 How to counteract leakage 482
9.3 Heat flow and heat removal 488
9.4 Appendix I: Contributions to node capacitance 490
9.5 Appendix II: Unorthodox approaches 491
9.5.1 Subthreshold logic 491
9.5.2 Voltage-swing-reduction techniques 492
9.5.3 Adiabatic logic 492
Chapter 10 Signal Integrity 495
10.1 Introduction 495
10.1.1 How does noise enter electronic circuits? 495
10.1.2 How does noise affect digital circuits? 496
10.1.3 Agenda 499
10.2 Crosstalk 499
10.3 Ground bounce and supply droop 499
10.3.1 Coupling mechanisms due to common series impedances 499
10.3.2 Where do large switching currents originate? 501
10.3.3 How severe is the impact of ground bounce? 501

10.4 How to mitigate ground bounce 504
10.4.1 Reduce effective series impedances 505
10.4.2 Separate polluters from potential victims 510
10.4.3 Avoid excessive switching currents 513
10.4.4 Safeguard noise margins 517
10.5 Conclusions 519
10.6 Problems 519
10.7 Appendix: Derivation of second-order approximation 521
Chapter 11 Physical Design 523
11.1 Agenda 523
11.2 Conducting layers and their characteristics 523
11.2.1 Geometric properties and layout rules 523
11.2.2 Electrical properties 527
11.2.3 Connecting between layers 527
11.2.4 Typical roles of conducting layers 529
11.3 Cell-based back-end design 531
11.3.1 Floorplanning 531
11.3.2 Identify major building blocks and clock domains 532
11.3.3 Establish a pin budget 533
11.3.4 Find a relative arrangement of all major building blocks 534
11.3.5 Plan power, clock, and signal distribution 535
11.3.6 Place and route (P&R) 538
11.3.7 Chip assembly 539
11.4 Packaging 540
11.4.1 Wafer sorting 543
11.4.2 Wafer testing 543
11.4.3 Backgrinding and singulation 544
11.4.4 Encapsulation 544
11.4.5 Final testing and binning 544
11.4.6 Bonding diagram and bonding rules 545
11.4.7 Advanced packaging techniques 546
11.4.8 Selecting a packaging technique 551
11.5 Layout at the detail level 551
11.5.1 Objectives of manual layout design 552
11.5.2 Layout design is no WYSIWYG business 552
11.5.3 Standard cell layout 556
11.5.4 Sea-of-gates macro layout 559
11.5.5 SRAM cell layout 559
11.5.6 Lithography-friendly layouts help improve fabrication yield 561
11.5.7 The mesh, a highly efficient and popular layout arrangement 562
11.6 Preventing electrical overstress 562
11.6.1 Electromigration 562
11.6.2 Electrostatic discharge 565
11.6.3 Latch-up 571
11.7 Problems 575
11.8 Appendix I: Geometric quantities advertized in VLSI 576
11.9 Appendix II: On coding diffusion areas in layout drawings 577
11.10 Appendix III: Sheet resistance 579
Chapter 12 Design Verification 581
12.1 Uncovering timing problems 581
12.1.1 What does simulation tell us about timing problems? 581
12.1.2 How does timing verification help? 585
12.2 How accurate are timing data? 587
12.2.1 Cell delays 588

12.2.2 Interconnect delays and layout parasitics 593
12.2.3 Making realistic assumptions is the point 597
12.3 More static verification techniques 598
12.3.1 Electrical rule check 598
12.3.2 Code inspection 599
12.4 Post-layout design verification 601
12.4.1 Design rule check 602
12.4.2 Manufacturability analysis 604
12.4.3 Layout extraction 605
12.4.4 Layout versus schematic 605
12.4.5 Equivalence checking 606
12.4.6 Post-layout timing verification 606
12.4.7 Power grid analysis 607
12.4.8 Signal integrity analysis 607
12.4.9 Post-layout simulations 607
12.4.10 The overall picture 607
12.5 Conclusions 608
12.6 Problems 609
12.7 Appendix I: Cell and library characterization 611
12.8 Appendix II: Equivalent circuits for interconnect modelling 612
Chapter 13 VLSI Economics and Project Management 615
13.1 Agenda 615
13.2 Models of industrial cooperation 617
13.2.1 Systems assembled from standard parts exclusively 617
13.2.2 Systems built around program-controlled processors 618
13.2.3 Systems designed on the basis of field-programmable logic 619
13.2.4 Systems designed on the basis of semi-custom ASICs 620
13.2.5 Systems designed on the basis of full-custom ASICs 622
13.3 Interfacing within the ASIC industry 623
13.3.1 Handoff points for IC design data 623
13.3.2 Scopes of IC manufacturing services 624
13.4 Virtual components 627
13.4.1 Copyright protection vs. customer information 627
13.4.2 Design reuse demands better quality and more thorough verification 628
13.4.3 Many existing virtual components need to be reworked 629
13.4.4 Virtual components require follow-up services 629
13.4.5 Indemnification provisions 630
13.4.6 Deliverables of a comprehensive VC package 630
13.4.7 Business models 631
13.5 The costs of integrated circuits 632
13.5.1 The impact of circuit size 633
13.5.2 The impact of the fabrication process 636
13.5.3 The impact of volume 638
13.5.4 The impact of configurability 639
13.5.5 Digest 640
13.6 Fabrication avenues for small quantities 642
13.6.1 Multi-project wafers 642
13.6.2 Multi-layer reticles 643
13.6.3 Electron beam lithography 643
13.6.4 Laser programming 643
13.6.5 Hardwired FPGAs and structured ASICs 644
13.6.6 Cost trading 644
13.7 The market side 645
13.7.1 Ingredients of commercial success 645

13.7.2 Commercialization stages and market priorities 646
13.7.3 Service versus product 649
13.7.4 Product grading 650
13.8 Making a choice 651
13.8.1 ASICs yes or no? 651
13.8.2 Which implementation technique should one adopt? 655
13.8.3 What if nothing is known for sure? 657
13.8.4 Can system houses afford to ignore microelectronics? 658
13.9 Keys to successful VLSI design 660
13.9.1 Project definition and marketing 660
13.9.2 Technical management 661
13.9.3 Engineering 662
13.9.4 Verification 665
13.9.5 Myths 665
13.10 Appendix: Doing business in microelectronics 667
13.10.1 Checklists for evaluating business partners and design kits 667
13.10.2 Virtual component providers 669
13.10.3 Selected low-volume providers 669
13.10.4 Cost estimation helps 669
Chapter 14 A Primer on CMOS Technology 671
14.1 The essence of MOS device physics 671
14.1.1 Energy bands and electrical conduction 671
14.1.2 Doping of semiconductor materials 672
14.1.3 Junctions, contacts, and diodes 674
14.1.4 MOSFETs 676
14.2 Basic CMOS fabrication flow 682
14.2.1 Key characteristics of CMOS technology 682
14.2.2 Front-end-of-line fabrication steps 685
14.2.3 Back-end-of-line fabrication steps 688
14.2.4 Process monitoring 689
14.2.5 Photolithography 689
14.3 Variations on the theme 697
14.3.1 Copper has replaced aluminum as interconnect material 697
14.3.2 Low-permittivity interlevel dielectrics are replacing silicon dioxide 698
14.3.3 High-permittivity gate dielectrics to replace silicon dioxide 699
14.3.4 Strained silicon and SiGe technology 701
14.3.5 Metal gates bound to come back 702
14.3.6 Silicon-on-insulator (SOI) technology 703
Chapter 15 Outlook 706
15.1 Evolution paths for CMOS technology 706
15.1.1 Classic device scaling 706
15.1.2 The search for new device topologies 709
15.1.3 Vertical integration 711
15.1.4 The search for better semiconductor materials 712
15.2 Is there life after CMOS? 714
15.2.1 Non-CMOS data storage 715
15.2.2 Non-CMOS data processing 716
15.3 Technology push 719
15.3.1 The so-called industry “laws” and the forces behind them 719
15.3.2 Industrial roadmaps 721
15.4 Market pull 723
15.5 Evolution paths for design methodology 724
15.5.1 The productivity problem 724
15.5.2 Fresh approaches to architecture design 727

15.6 Summary 729
15.7 Six grand challenges 730
15.8 Appendix: Non-semiconductor storage technologies for comparison 731
Appendix A Elementary Digital Electronics 732
A.1 Introduction 732
A.1.1 Common number representation schemes 732
A.1.2 Notational conventions for two-valued logic 734
A.2 Theoretical background of combinational logic 735
A.2.1 Truth table 735
A.2.2 The n-cube 736
A.2.3 Karnaugh map 736
A.2.4 Program code and other formal languages 736
A.2.5 Logic equations 737
A.2.6 Two-level logic 738
A.2.7 Multilevel logic 740
A.2.8 Symmetric and monotone functions 741
A.2.9 Threshold functions 741
A.2.10 Complete gate sets 742
A.2.11 Multi-output functions 742
A.2.12 Logic minimization 743
A.3 Circuit alternatives for implementing combinational logic 747
A.3.1 Random logic 747
A.3.2 Programmable logic array (PLA) 747
A.3.3 Read-only memory (ROM) 749
A.3.4 Array multiplier 749
A.3.5 Digest 750
A.4 Bistables and other memory circuits 751
A.4.1 Flip-flops or edge-triggered bistables 752
A.4.2 Latches or level-sensitive bistables 755
A.4.3 Unclocked bistables 756
A.4.4 Random access memories (RAMs) 760
A.5 Transient behavior of logic circuits 761
A.5.1 Glitches, a phenomenological perspective 762
A.5.2 Function hazards, a circuit-independent mechanism 763
A.5.3 Logic hazards, a circuit-dependent mechanism 764
A.5.4 Digest 765
A.6 Timing quantities 766
A.6.1 Delay quantities apply to combinational and sequential circuits 766
A.6.2 Timing conditions apply to sequential circuits only 768
A.6.3 Secondary timing quantities 770
A.6.4 Timing constraints address synthesis needs 771
A.7 Microprocessor input/output transfer protocols 771
A.8 Summary 773
Appendix B Finite State Machines 775
B.1 Abstract automata 775
B.1.1 Mealy machine 776
B.1.2 Moore machine 777
B.1.3 Medvedev machine 778
B.1.4 Relationships between finite state machine models 779
B.1.5 Taxonomy of finite state machines 782
B.1.6 State reduction 783
B.2 Practical aspects and implementation issues 785
B.2.1 Parasitic states and symbols 785
B.2.2 Mealy-, Moore-, Medvedev-type, and combinational output bits 787

B.2.3 Through paths and logic instability 787
B.2.4 Switching hazards 789
B.2.5 Hardware costs 790
B.3 Summary 793
Appendix C VLSI Designer’ s Checklist 794
C.1 Design data sanity 794
C.2 Pre-synthesis design verification 794
C.3 Clocking 795
C.4 Gate-level considerations 796
C.5 Design for test 797
C.6 Electrical considerations 798
C.7 Pre-layout design verification 799
C.8 Physical considerations 800
C.9 Post-layout design verification 800
C.10 Preparation for testing of fabricated prototypes 801
C.11 Thermal considerations 802
C.12 Board-level operation and testing 802
C.13 Documentation 802
Appendix D Symbols and constants 804
D.1 Mathematical symbols used 804
D.2 Abbreviations 807
D.3 Physical and material constants 808
References 811
Index 832
· · · · · · · (收起)

数字集成电路设计_下载链接1_

标签

电子

VLSI

数字集成电路

全定制

IC

EE

CMOS

评论

数字集成电路设计_下载链接1_

书评

这本书是Digital Integrated Circuit Design:From VLSI Architectures to CMOS Fabrication的中文版，数字集成电路设计：从VLSI体系结构到CMOS制造。洋洋洒洒600多页，100多块钱。但是我开始看了三天了。翻译简直就像谷歌翻译的一样，我专门下载了英文电子版来对照。原版是很...

数字集成电路设计_下载链接1_