CONTENTS

Preface xix

Acknowledgments xxiii

1 Introduction to Digital System Design 1

1.1 Introduction 1
1.2 Device technologies 2
1.2.1 Fabrication of an IC 2
1.2.2 Classification of device technologies 2
1.2.3 Comparison of technologies 5
1.3 System representation 8
1.4 Levels of Abstraction 9
1.4.1 Transistor-level abstraction 10
1.4.2 Gate-level abstraction 10
1.4.3 Register-transfer-level (RT-level) abstraction 11
1.4.4 Processor-level abstraction 12
1.5 Development tasks and EDA software 12
1.5.1 Synthesis 13
1.5.2 Physical design 14
1.5.3 Verification 14
1.5.4 Testing 16
1.5.5 EDA software and its limitations 16

CuuDuongThanCong.com https://fb.com/tailieudientucntt

1.6 Development flow 17
1.6.1 Flow of a medium-sized design targeting FPGA 17
1.6.2 Flow of a large design targeting FPGA 19
1.6.3 Flow of a large design targeting ASIC 19
1.7 Overview of the book 20
1.7.1 Scope 20
1.7.2 Goal 20
1.8 Bibliographic notes 21
Problems 22

2 Overview of Hardware Description Languages 23

2.1 Hardware description languages 23
2.1.1 Limitations of traditional programming languages 23
2.1.2 Use of an HDL program 24
2.1.3 Design of a modern HDL 25
2.1.4 VHDL 25
2.2 Basic VHDL concept via an example 26
2.2.1 General description 27
2.2.2 Structural description 30
2.2.3 Abstract behavioral description 33
2.2.4 Testbench 35
2.2.5 Configuration 37
2.3 VHDL in development flow 38
2.3.1 Scope of VHDL 38
2.3.2 Coding for synthesis 40
2.4 Bibliographic notes 40
Problems 41

3 Basic Language Constructs of VHDL 43

3.1 Introduction 43
3.2 Skeleton of a basic VHDL program 44
3.2.1 Example of a VHDL program 44
3.2.2 Entity declaration 44
3.2.3 Architecture body 46
3.2.4 Design unit and library 46
3.2.5 Processing of VHDL code 47
3.3 Lexical elements and program format 47
3.3.1 Lexical elements 47
3.3.2 VHDL program format 49
3.4 Objects 51
3.5 Data types and operators 53

cuu duong than cong . com
cuu duong than cong . com
CuuDuongThanCong.com https://fb.com/tailieudientucntt

3.5.1 Predefined data types in VHDL 53
3.5.2 Data types in the IEEE std_logic_1164 package 56
3.5.3 Operators over an array data type 58
3.5.4 Data types in the IEEE numeric_std package 60
3.5.5 The std_logic_arith and related packages 64
3.6 Synthesis guidelines 65
3.6.1 Guidelines for general VHDL 65
3.6.2 Guidelines for VHDL formatting 66
3.7 Bibliographic notes 66
Problems 66

4 Concurrent Signal Assignment Statements of VHDL 69

4.1 Combinational versus sequential circuits 69
4.2 Simple signal assignment statement 70
4.2.1 Syntax and examples 70
4.2.2 Conceptual implementation 70
4.2.3 Signal assignment statement with a closed feedback loop 71
4.3 Conditional signal assignment statement 72
4.3.1 Syntax and examples 72
4.3.2 Conceptual implementation 76
4.3.3 Detailed implementation examples 78
4.4 Selected signal assignment statement 85
4.4.1 Syntax and examples 85
4.4.2 Conceptual implementation 88
4.4.3 Detailed implementation examples 90
4.5 Conditional signal assignment statement versus selected signal assignment statement 93
4.5.1 Conversion between conditional signal assignment and selected signal assignment statements 93
4.5.2 Comparison between conditional signal assignment and selected signal assignment statements 94
4.6 Synthesis guidelines 95
4.7 Bibliographic notes 95
Problems 95

5 Sequential Statements of VHDL 97

5.1 VHDL process 97
5.1.1 Introduction 97
5.1.2 Process with a sensitivity list 98
5.1.3 Process with a wait statement 99
5.2 Sequential signal assignment statement 100

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

5.3 Variable assignment statement 101
5.4 If statement 103
5.4.1 Syntax and examples 103
5.4.2 Comparison to a conditional signal assignment statement 105
5.4.3 Incomplete branch and incomplete signal assignment 107
5.4.4 Conceptual implementation 109
5.4.5 Cascading single-branched if statements 110
5.5 Case statement 112
5.5.1 Syntax and examples 112
5.5.2 Comparison to a selected signal assignment statement 114
5.5.3 Incomplete signal assignment 115
5.5.4 Conceptual implementation 116
5.6 Simple for loop statement 118
5.6.1 Syntax 118
5.6.2 Examples 118
5.6.3 Conceptual implementation 119
5.7 Synthesis of sequential statements 120
5.8 Synthesis guidelines 120
5.8.1 Guidelines for using sequential statements 120
5.8.2 Guidelines for combinational circuits 121
5.9 Bibliographic notes 121
Problems 121

6 Synthesis Of VHDL Code 125

6.1 Fundamental limitations of EDA software 125
6.1.1 Computability 126
6.1.2 Computation complexity 126
6.1.3 Limitations of EDA software 128
6.2 Realization of VHDL operators 129
6.2.1 Realization of logical operators 129
6.2.2 Realization of relational operators 129
6.2.3 Realization of addition operators 130
6.2.4 Synthesis support for other operators 130
6.2.5 Realization of an operator with constant operands 130
6.2.6 An example implementation 131
6.3 Realization of VHDL data types 133
6.3.1 Use of the `std_logic` data type 133
6.3.2 Use and realization of the `'Z'` value 133
6.3.3 Use of the `'-'` value 137
6.4 VHDL synthesis flow 139
6.4.1 RT-level synthesis 139
6.4.2 Module generator 141

cuu duong than cong . com
cuu duong than cong . com
CuuDuongThanCong.com https://fb.com/tailieudientucntt

6.4.3 Logic synthesis 142
6.4.4 Technology mapping 143
6.4.5 Effective use of synthesis software 148
6.5 Timing considerations 149
6.5.1 Propagation delay 150
6.5.2 Synthesis with timing constraints 154
6.5.3 Timing hazards 156
6.5.4 Delay-sensitive design and its dangers 158
6.6 Synthesis guidelines 160
6.7 Bibliographic notes 160
Problems 160

7 Combinational Circuit Design: Practice 163

7.1 Derivation of efficient HDL description 163
7.2 Operator sharing 164
7.2.1 Sharing example 1 165
7.2.2 Sharing example 2 166
7.2.3 Sharing example 3 168
7.2.4 Sharing example 4 169
7.2.5 Summary 170
7.3 Functionality sharing 170
7.3.1 Addition–subtraction circuit 171
7.3.2 Signed–unsigned dual-mode comparator 173
7.3.3 Difference circuit 175
7.3.4 Full comparator 177
7.3.5 Three-function barrel shifter 178
7.4 Layout-related circuits 180
7.4.1 Reduced-xor circuit 181
7.4.2 Reduced-xor-vector circuit 183
7.4.3 Tree priority encoder 187
7.4.4 Barrel shifter revisited 192
7.5 General circuits 196
7.5.1 Gray code incrementor 196
7.5.2 Programmable priority encoder 199
7.5.3 Signed addition with status 201
7.5.4 Combinational adder-based multiplier 203
7.5.5 Hamming distance circuit 206
7.6 Synthesis guidelines 208
7.7 Bibliographic notes 208
Problems 208

8 Sequential Circuit Design: Principle 213

cuu duong than cong . com
cuu duong than cong . com
CuuDuongThanCong.com https://fb.com/tailieudientucntt

8.1 Overview of sequential circuits 213
8.1.1 Sequential versus combinational circuits 213
8.1.2 Basic memory elements 214
8.1.3 Synchronous versus asynchronous circuits 216
8.2 Synchronous circuits 217
8.2.1 Basic model of a synchronous circuit 217
8.2.2 Synchronous circuits and design automation 218
8.2.3 Types of synchronous circuits 219
8.3 Danger of synthesis that uses primitive gates 219
8.4 Inference of basic memory elements 221
8.4.1 D latch 221
8.4.2 D FF 222
8.4.3 Register 225
8.4.4 RAM 225
8.5 Simple design examples 226
8.5.1 Other types of FFs 226
8.5.2 Shift register 229
8.5.3 Arbitrary-sequence counter 232
8.5.4 Binary counter 233
8.5.5 Decade counter 236
8.5.6 Programmable mod-m counter 237
8.6 Timing analysis of a synchronous sequential circuit 239
8.6.1 Synchronized versus unsynchronized input 239
8.6.2 Setup time violation and maximal clock rate 240
8.6.3 Hold time violation 243
8.6.4 Output-related timing considerations 243
8.6.5 Input-related timing considerations 244
8.7 Alternative one-segment coding style 245
8.7.1 Examples of one-segment code 245
8.7.2 Summary 250
8.8 Use of variables in sequential circuit description 250
8.9 Synthesis of sequential circuits 253
8.10 Synthesis guidelines 253
8.11 Bibliographic notes 253
Problems 254

9 Sequential Circuit Design: Practice 257

9.1 Poor design practices and their remedies 257
9.1.1 Misuse of asynchronous signals 258
9.1.2 Misuse of gated clocks 260
9.1.3 Misuse of derived clocks 262
9.2 Counters 265

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

9.2.1 Gray counter 265
9.2.2 Ring counter 266
9.2.3 LFSR (linear feedback shift register) 269
9.2.4 Decimal counter 272
9.2.5 Pulse width modulation circuit 275
9.3 Registers as temporary storage 276
9.3.1 Register file 276
9.3.2 Register-based synchronous FIFO buffer 279
9.3.3 Register-based content addressable memory 287
9.4 Pipelined design 293
9.4.1 Delay versus throughput 294
9.4.2 Overview on pipelined design 294
9.4.3 Adding pipeline to a combinational circuit 297
9.4.4 Synthesis of pipelined circuits and retiming 307
9.5 Synthesis guidelines 308
9.6 Bibliographic notes 309
Problems 309

10 Finite State Machine: Principle and Practice 313

10.1 Overview of FSMs 313
10.2 FSM representation 314
10.2.1 State diagram 315
10.2.2 ASM chart 317
10.3 Timing and performance of an FSM 321
10.3.1 Operation of a synchronous FSM 321
10.3.2 Performance of an FSM 324
10.3.3 Representative timing diagram 325
10.4 Moore machine versus Mealy machine 325
10.4.1 Edge detection circuit 326
10.4.2 Comparison of Moore output and Mealy output 328
10.5 VHDL description of an FSM 329
10.5.1 Multi-segment coding style 330
10.5.2 Two-segment coding style 333
10.5.3 Synchronous FSM initialization 335
10.5.4 One-segment coding style and its problem 336
10.5.5 Synthesis and optimization of FSM 337
10.6 State assignment 338
10.6.1 Overview of state assignment 338
10.6.2 State assignment in VHDL 339
10.6.3 Handling the unused states 341
10.7 Moore output buffering 342
10.7.1 Buffering by clever state assignment 342

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

10.7.2 Look-ahead output circuit for Moore output 344
10.8 FSM design examples 348
10.8.1 Edge detection circuit 348
10.8.2 Arbiter 353
10.8.3 DRAM strobe generation circuit 358
10.8.4 Manchester encoding circuit 363
10.8.5 FSM-based binary counter 367
10.9 Bibliographic notes 369
Problems 369

11 Register Transfer Methodology: Principle 373

11.1 Introduction 373
11.1.1 Algorithm 373
11.1.2 Structural data flow implementation 374
11.1.3 Register transfer methodology 375
11.2 Overview of FSMD 376
11.2.1 Basic RT operation 376
11.2.2 Multiple RT operations and data path 378
11.2.3 FSM as the control path 379
11.2.4 ASMD chart 379
11.2.5 Basic FSMD block diagram 380
11.3 FSMD design of a repetitive-addition multiplier 382
11.3.1 Converting an algorithm to an ASMD chart 382
11.3.2 Construction of the FSMD 385
11.3.3 Multi-segment VHDL description of an FSMD 386
11.3.4 Use of a register value in a decision box 389
11.3.5 Four- and two-segment VHDL descriptions of FSMD 391
11.3.6 One-segment coding style and its deficiency 394
11.4 Alternative design of a repetitive-addition multiplier 396
11.4.1 Resource sharing via FSMD 396
11.4.2 Mealy-controlled RT operations 400
11.5 Timing and performance analysis of FSMD 404
11.5.1 Maximal clock rate 404
11.5.2 Performance analysis 407
11.6 Sequential add-and-shift multiplier 407
11.6.1 Initial design 408
11.6.2 Refined design 412
11.6.3 Comparison of three ASMD designs 417
11.7 Synthesis of FSMD 417
11.8 Synthesis guidelines 418
11.9 Bibliographic notes 418
Problems 418

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

12 Register Transfer Methodology: Practice 421

12.1 Introduction 421
12.2 One-shot pulse generator 422
12.2.1 FSM implementation 422
12.2.2 Regular sequential circuit implementation 424
12.2.3 Implementation using RT methodology 425
12.2.4 Comparison 427
12.3 SRAM controller 430
12.3.1 Overview of SRAM 430
12.3.2 Block diagram of an SRAM controller 434
12.3.3 Control path of an SRAM controller 436
12.4 GCD circuit 445
12.5 UART receiver 455
12.6 Square-root approximation circuit 460
12.7 High-level synthesis 469
12.8 Bibliographic notes 470
Problems 470

13 Hierarchical Design in VHDL 473

13.1 Introduction 473
13.1.1 Benefits of hierarchical design 474
13.1.2 VHDL constructs for hierarchical design 474
13.2 Components 475
13.2.1 Component declaration 475
13.2.2 Component instantiation 477
13.2.3 Caveats in component instantiation 480
13.3 Generics 481
13.4 Configuration 485
13.4.1 Introduction 485
13.4.2 Configuration declaration 486
13.4.3 Configuration specification 488
13.4.4 Component instantiation and configuration in VHDL 93 488
13.5 Other supporting constructs for a large system 489
13.5.1 Library 489
13.5.2 Subprogram 491
13.5.3 Package 492
13.6 Partition 495
13.6.1 Physical partition 495
13.6.2 Logical partition 496
13.7 Synthesis guidelines 497
13.8 Bibliographic notes 497

cuu duong than cong . com
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Problems 497

14 Parameterized Design: Principle 499

14.1 Introduction 499
14.2 Types of parameters 500
14.2.1 Width parameters 500
14.2.2 Feature parameters 501
14.3 Specifying parameters 501
14.3.1 Generics 501
14.3.2 Array attribute 502
14.3.3 Unconstrained array 503
14.3.4 Comparison between a generic and an unconstrained array 506
14.4 Clever use of an array 506
14.4.1 Description without fixed-size references 507
14.4.2 Examples 509
14.5 For generate statement 512
14.5.1 Syntax 513
14.5.2 Examples 513
14.6 Conditional generate statement 517
14.6.1 Syntax 517
14.6.2 Examples 518
14.6.3 Comparisons with other feature-selection methods 525
14.7 For loop statement 528
14.7.1 Introduction 528
14.7.2 Examples of a simple for loop statement 528
14.7.3 Examples of a loop body with multiple signal assignment statements 530
14.7.4 Examples of a loop body with variables 533
14.7.5 Comparison of the for generate and for loop statements 536
14.8 Exit and next statements 537
14.8.1 Syntax of the exit statement 537
14.8.2 Examples of the exit statement 537
14.8.3 Conceptual implementation of the exit statement 539
14.8.4 Next statement 540
14.9 Synthesis of iterative structure 541
14.10 Synthesis guidelines 542
14.11 Bibliographic notes 542
Problems 542

15 Parameterized Design: Practice 545

15.1 Introduction 545

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

15.2 Data types for two-dimensional signals 546
15.2.1 Genuine two-dimensional data type 546
15.2.2 Array-of-arrays data type 548
15.2.3 Emulated two-dimensional array 550
15.2.4 Example 552
15.2.5 Summary 554
15.3 Commonly used intermediate-sized RT-level components 555
15.3.1 Reduced-xor circuit 555
15.3.2 Binary decoder 558
15.3.3 Multiplexer 560
15.3.4 Binary encoder 564
15.3.5 Barrel shifter 566
15.4 More sophisticated examples 569
15.4.1 Reduced-xor-vector circuit 570
15.4.2 Multiplier 572
15.4.3 Parameterized LFSR 586
15.4.4 Priority encoder 588
15.4.5 FIFO buffer 591
15.5 Synthesis of parameterized modules 599
15.6 Synthesis guidelines 599
15.7 Bibliographic notes 600
Problems 600

16 Clock and Synchronization: Principle and Practice 603

16.1 Overview of a clock distribution network 603
16.1.1 Physical implementation of a clock distribution network 603
16.1.2 Clock skew and its impact on synchronous design 605
16.2 Timing analysis with clock skew 606
16.2.1 Effect on setup time and maximal clock rate 606
16.2.2 Effect on hold time constraint 609
16.3 Overview of a multiple-clock system 610
16.3.1 System with derived clock signals 611
16.3.2 GALS system 612
16.4 Metastability and synchronization failure 612
16.4.1 Nature of metastability 613
16.4.2 Analysis of MTBF(T_r) 614
16.4.3 Unique characteristics of MTBF(T_r) 616
16.5 Basic synchronizer 617
16.5.1 The danger of no synchronizer 617
16.5.2 One-FF synchronizer and its deficiency 617
16.5.3 Two-FF synchronizer 619
16.5.4 Three-FF synchronizer 620

cuu duong than cong . com
cuu duong than cong . com
CuuDuongThanCong.com https://fb.com/tailieudientucntt

16.5.5 Proper use of a synchronizer 621
16.6 Single enable signal crossing clock domains 623
16.6.1 Edge detection scheme 623
16.6.2 Level-alternation scheme 627
16.7 Handshaking protocol 630
16.7.1 Four-phase handshaking protocol 630
16.7.2 Two-phase handshaking protocol 637
16.8 Data transfer crossing clock domains 639
16.8.1 Four-phase handshaking protocol data transfer 641
16.8.2 Two-phase handshaking data transfer 650
16.8.3 One-phase data transfer 651
16.9 Data transfer via a memory buffer 652
16.9.1 FIFO buffer 652
16.9.2 Shared memory 660
16.10 Synthesis of a multiple-clock system 661
16.11 Synthesis guidelines 662
16.11.1 Guidelines for general use of a clock 662
16.11.2 Guidelines for a synchronizer 662
16.11.3 Guidelines for an interface between clock domains 662
16.12 Bibliographic notes 663
Problems 663

References 665

Topic Index 667

cuu duong than cong . com
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt