CONTENTS

Preface xxi
Acknowledgments xxvii

PART I BASIC DIGITAL CIRCUITS

1 Gate-level combinational circuit 1
   1.1 Introduction 1
   1.2 General description 2
   1.3 Basic lexical elements and data types 3
       1.3.1 Lexical elements 3
   1.4 Data types 4
       1.4.1 Four-value system 4
       1.4.2 Data type groups 4
       1.4.3 Number representation 5
       1.4.4 Operators 5
   1.5 Program skeleton 5
       1.5.1 Port declaration 6
       1.5.2 Program body 7
       1.5.3 Signal declaration 7
       1.5.4 Another example 8
   1.6 Structural description 9
   1.7 Testbench 12
CONTENTS

1.8 Bibliographic notes 14
1.9 Suggested experiments 14

1.9.1 Code for gate-level greater-than circuit 14
1.9.2 Code for gate-level binary decoder 14

2 Overview of FPGA and EDA software 15

2.1 Introduction 15
2.2 FPGA 15

2.2.1 Overview of a general FPGA device 15
2.2.2 Overview of the Xilinx Spartan-3 devices 17
2.3 Overview of the Digilent S3 board 17
2.4 Development flow 19
2.5 Overview of the Xilinx ISE project navigator 21
2.6 Short tutorial on ISE project navigator 24

2.6.1 Create the design project and HDL codes 25
2.6.2 Create a testbench and perform the RTL simulation 26
2.6.3 Add a constraint file and synthesize and implement the code 26
2.6.4 Generate and download the configuration file to an FPGA device 29
2.7 Short tutorial on the ModelSim HDL simulator 31
2.8 Bibliographic notes 35
2.9 Suggested experiments 36

2.9.1 Gate-level greater-than circuit 36
2.9.2 Gate-level binary decoder 36

3 RT-level combinational circuit 39

3.1 Introduction 39
3.2 Operators 39

3.2.1 Arithmetic operators 41
3.2.2 Shift operators 41
3.2.3 Relational and equality operators 42
3.2.4 Bitwise, reduction, and logical operators 42
3.2.5 Concatenation and replication operators 43
3.2.6 Conditional operators 44
3.2.7 Operator precedence 44
3.2.8 Expression bit-length adjustment 45
3.2.9 Synthesis of z and x values 46

3.3 Always block for a combinational circuit 48

3.3.1 Basic syntax and behavior 48
3.3.2 Procedural assignment 49
3.3.3 Variable data types 49
3.3.4 Simple examples 49
3.4 If statement 51
3.4.1 Syntax 51
3.4.2 Examples 52
3.5 Case statement 54
3.5.1 Syntax 54
3.5.2 Examples 54
3.5.3 The casez and casex statements 56
3.5.4 The full case and parallel case 56
3.6 Routing structure of conditional control constructs 57
3.6.1 Priority routing network 57
3.6.2 Multiplexing network 59
3.7 General coding guidelines for an always block 60
3.7.1 Common errors in combinational circuit codes 60
3.7.2 Guidelines 63
3.8 Parameter and constant 64
3.8.1 Constant 64
3.8.2 Parameter 65
3.8.3 Use of parameters in Verilog-1995 67
3.9 Design examples 67
3.9.1 Hexadecimal digit to seven-segment LED decoder 67
3.9.2 Sign-magnitude adder 71
3.9.3 Barrel shifter 73
3.9.4 Simplified floating-point adder 75
3.10 Bibliographic notes 80
3.11 Suggested experiments 80
3.11.1 Multifunction barrel shifter 80
3.11.2 Dual-priority encoder 80
3.11.3 BCD incrementor 81
3.11.4 Floating-point greater-than circuit 81
3.11.5 Floating-point and signed integer conversion circuit 81
3.11.6 Enhanced floating-point adder 81

4 Regular Sequential Circuit 83
4.1 Introduction 83
4.1.1 D FF and register 83
4.1.2 Synchronous system 84
4.1.3 Code development 85
4.2 HDL code of the FF and register 86
4.2.1 D FF 86
4.2.2 Register 89
4.2.3 Register file 90
4.2.4 Storage components in a Spartan-3 device 91
4.3 Simple design examples 91
  4.3.1 Shift register 91
  4.3.2 Binary counter and variant 93
4.4 Testbench for sequential circuits 96
4.5 Case study 99
  4.5.1 LED time-multiplexing circuit 99
  4.5.2 Stopwatch 107
  4.5.3 FIFO buffer 110
4.6 Bibliographic notes 115
4.7 Suggested experiments 115
  4.7.1 Programmable square-wave generator 115
  4.7.2 PWM and LED dimmer 115
  4.7.3 Rotating square circuit 116
  4.7.4 Heartbeat circuit 116
  4.7.5 Rotating LED banner circuit 116
  4.7.6 Enhanced stopwatch 116
  4.7.7 Stack 117

5 FSM 119
  5.1 Introduction 119
    5.1.1 Mealy and Moore outputs 119
    5.1.2 FSM representation 120
  5.2 FSM code development 122
  5.3 Design examples 125
    5.3.1 Rising-edge detector 125
    5.3.2 Debouncing circuit 130
    5.3.3 Testing circuit 133
  5.4 Bibliographic notes 135
  5.5 Suggested experiments 135
    5.5.1 Dual-edge detector 135
    5.5.2 Alternative debouncing circuit 135
    5.5.3 Parking lot occupancy counter 136

6 FSMD 139
  6.1 Introduction 139
    6.1.1 Single RT operation 139
    6.1.2 ASMD chart 140
    6.1.3 Decision box with a register 141
  6.2 Code development of an FSMD 143
    6.2.1 Debouncing circuit based on RT methodology 144
    6.2.2 Code with explicit data path components 146
### PART II  I/O MODULES

#### 8 UART

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Introduction</td>
<td>215</td>
</tr>
<tr>
<td>8.2 UART receiving subsystem</td>
<td>216</td>
</tr>
<tr>
<td>8.2.1 Oversampling procedure</td>
<td>216</td>
</tr>
<tr>
<td>8.2.2 Baud rate generator</td>
<td>217</td>
</tr>
<tr>
<td>8.2.3 UART receiver</td>
<td>217</td>
</tr>
<tr>
<td>8.2.4 Interface circuit</td>
<td>220</td>
</tr>
<tr>
<td>8.3 UART transmitting subsystem</td>
<td>223</td>
</tr>
<tr>
<td>8.4 Overall UART system</td>
<td>226</td>
</tr>
<tr>
<td>8.4.1 Complete UART core</td>
<td>226</td>
</tr>
<tr>
<td>8.4.2 UART verification configuration</td>
<td>228</td>
</tr>
<tr>
<td>8.5 Customizing a UART</td>
<td>230</td>
</tr>
<tr>
<td>8.6 Bibliographic notes</td>
<td>232</td>
</tr>
<tr>
<td>8.7 Suggested experiments</td>
<td>232</td>
</tr>
<tr>
<td>8.7.1 Full-featured UART</td>
<td>232</td>
</tr>
<tr>
<td>8.7.2 UART with an automatic baud rate detection circuit</td>
<td>233</td>
</tr>
<tr>
<td>8.7.3 UART with an automatic baud rate and parity detection circuit</td>
<td>233</td>
</tr>
<tr>
<td>8.7.4 UART-controlled stopwatch</td>
<td>233</td>
</tr>
<tr>
<td>8.7.5 UART-controlled rotating LED banner</td>
<td>234</td>
</tr>
</tbody>
</table>

#### 9 PS2 Keyboard

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 Introduction</td>
<td>235</td>
</tr>
<tr>
<td>9.2 PS2 receiving subsystem</td>
<td>236</td>
</tr>
</tbody>
</table>
9.2.1 Physical interface of a PS2 port 236
9.2.2 Device-to-host communication protocol 236
9.2.3 Design and code 236
9.3 PS2 keyboard scan code 240
9.3.1 Overview of the scan code 240
9.3.2 Scan code monitor circuit 241
9.4 PS2 keyboard interface circuit 244
9.4.1 Basic design and HDL code 244
9.4.2 Verification circuit 246
9.5 Bibliographic notes 248
9.6 Suggested experiments 248
9.6.1 Alternative keyboard interface I 248
9.6.2 Alternative keyboard interface II 249
9.6.3 PS2 receiving subsystem with watchdog timer 249
9.6.4 Keyboard-controlled stopwatch 249
9.6.5 Keyboard-controlled rotating LED banner 249

10 PS2 Mouse 251
10.1 Introduction 251
10.2 PS2 mouse protocol 252
10.2.1 Basic operation 252
10.2.2 Basic initialization procedure 252
10.3 PS2 transmitting subsystem 253
10.3.1 Host-to-PS2-device communication protocol 253
10.3.2 Design and code 254
10.4 Bidirectional PS2 interface 259
10.4.1 Basic design and code 259
10.4.2 Verification circuit 260
10.5 PS2 mouse interface 263
10.5.1 Basic design 263
10.5.2 Testing circuit 265
10.6 Bibliographic notes 266
10.7 Suggested experiments 266
10.7.1 Keyboard control circuit 267
10.7.2 Enhanced mouse interface 267
10.7.3 Mouse-controlled seven-segment LED display 267

11 External SRAM 269
11.1 Introduction 269
11.2 Specification of the IS61LV25616AL SRAM 270
11.2.1 Block diagram and I/O signals 270
## 11.2.2 Timing parameters 270

### 11.3 Basic memory controller

#### 11.3.1 Block diagram 274

#### 11.3.2 Timing requirement 275

#### 11.3.3 Register file versus SRAM 276

### 11.4 A safe design

#### 11.4.1 ASM chart 276

#### 11.4.2 Timing analysis 277

#### 11.4.3 HDL implementation 278

#### 11.4.4 Basic testing circuit 281

#### 11.4.5 Comprehensive SRAM testing circuit 283

### 11.5 More aggressive design

#### 11.5.1 Timing issues 288

#### 11.5.2 Alternative design I 288

#### 11.5.3 Alternative design II 290

#### 11.5.4 Alternative design III 291

#### 11.5.5 Advanced FPGA features

### 11.6 Bibliographic notes 294

### 11.7 Suggested experiments

#### 11.7.1 Memory with a 512K-by-16 configuration 294

#### 11.7.2 Memory with a 1M-by-8 configuration 295

#### 11.7.3 Memory with an 8M-by-1 configuration 295

#### 11.7.4 Expanded memory testing circuit 295

#### 11.7.5 Memory controller and testing circuit for alternative design I 295

#### 11.7.6 Memory controller and testing circuit for alternative design II 295

#### 11.7.7 Memory controller and testing circuit for alternative design III 295

#### 11.7.8 Memory controller with DCM 295

#### 11.7.9 High-performance memory controller 296

## 12 Xilinx Spartan-3 Specific Memory 297

### 12.1 Introduction 297

### 12.2 Embedded memory of Spartan-3 device 297

#### 12.2.1 Overview 297

#### 12.2.2 Comparison 298

### 12.3 Method to incorporate memory modules 298

#### 12.3.1 Memory module via HDL component instantiation 299

#### 12.3.2 Memory module via Core Generator 299

#### 12.3.3 Memory module via HDL inference 300

### 12.4 HDL templates for memory inference 300

#### 12.4.1 Single-port RAM 300

#### 12.4.2 Dual-port RAM 303

#### 12.4.3 ROM 305
12.5 Bibliographic notes 307
12.6 Suggested experiments 307
  12.6.1 Block-RAM-based FIFO 307
  12.6.2 Block-RAM-based stack 307
  12.6.3 ROM-based sign-magnitude adder 307
  12.6.4 ROM-based $\sin(x)$ function 308
  12.6.5 ROM-based $\sin(x)$ and $\cos(x)$ functions 308

13 VGA controller I: graphic 309
  13.1 Introduction 309
    13.1.1 Basic operation of a CRT 309
    13.1.2 VGA port of the S3 board 311
    13.1.3 Video controller 311
  13.2 VGA synchronization 312
    13.2.1 Horizontal synchronization 312
    13.2.2 Vertical synchronization 314
    13.2.3 Timing calculation of VGA synchronization signals 315
    13.2.4 HDL implementation 315
    13.2.5 Testing circuit 318
  13.3 Overview of the pixel generation circuit 319
  13.4 Graphic generation with an object-mapped scheme 319
    13.4.1 Rectangular objects 320
    13.4.2 Non-rectangular object 325
    13.4.3 Animated object 326
  13.5 Graphic generation with a bit-mapped scheme 332
    13.5.1 Dual-port RAM implementation 332
    13.5.2 Single-port RAM implementation 337
  13.6 Bibliographic notes 337
  13.7 Suggested experiments 337
    13.7.1 VGA test pattern generator 337
    13.7.2 SVGA mode synchronization circuit 338
    13.7.3 Visible screen adjustment circuit 338
    13.7.4 Ball-in-a-box circuit 338
    13.7.5 Two-balls-in-a-box circuit 339
    13.7.6 Two-player pong game 339
    13.7.7 Breakout game 339
    13.7.8 Full-screen dot trace 339
    13.7.9 Mouse pointer circuit 340
    13.7.10 Small-screen mouse scribble circuit 340
    13.7.11 Full-screen mouse scribble circuit 340

14 VGA controller II: text 341
14.1 Introduction 341
14.2 Text generation 341
  14.2.1 Character as a tile 341
  14.2.2 Font ROM 342
  14.2.3 Basic text generation circuit 344
  14.2.4 Font display circuit 345
  14.2.5 Font scaling 347
14.3 Full-screen text display 348
14.4 The complete pong game 352
  14.4.1 Text subsystem 352
  14.4.2 Modified graphic subsystem 358
  14.4.3 Auxiliary counters 359
  14.4.4 Top-level system 361
14.5 Bibliographic notes 366
14.6 Suggested experiments 366
  14.6.1 Rotating banner 366
  14.6.2 Underline for the cursor 366
  14.6.3 Dual-mode text display 366
  14.6.4 Keyboard text entry 366
  14.6.5 UART terminal 366
  14.6.6 Square-wave display 367
  14.6.7 Simple four-trace logic analyzer 367
  14.6.8 Complete two-player pong game 368
  14.6.9 Complete breakout game 368

PART III PICOBLAZE MICROCONTROLLER

15 PicoBlaze Overview 371
15.1 Introduction 371
15.2 Customized hardware and customized software 372
  15.2.1 From special-purpose FSMD to general-purpose microcontroller 372
  15.2.2 Application of microcontroller 374
15.3 Overview of PicoBlaze 374
  15.3.1 Basic organization 374
  15.3.2 Top-level HDL modules 376
15.4 Development flow 377
15.5 Instruction set 377
  15.5.1 Programming model 379
  15.5.2 Instruction format 379
  15.5.3 Logical instructions 380
  15.5.4 Arithmetic instructions 381
  15.5.5 Compare and test instructions 382
15.5.6 Shift and rotate instructions 383
15.5.7 Data movement instructions 384
15.5.8 Program flow control instructions 386
15.5.9 Interrupt related instructions 389
15.6 Assembler directives 390
   15.6.1 The KCPM3 directives 390
   15.6.2 The PBlazeIDE directives 390
15.7 Bibliographic notes 391

16 PicoBlaze Assembly Code Development 393
   16.1 Introduction 393
   16.2 Useful code segments 393
      16.2.1 KCPM3 conventions 393
      16.2.2 Bit manipulation 394
      16.2.3 Multiple-byte manipulation 395
      16.2.4 Control structure 396
   16.3 Subroutine development 398
   16.4 Program development 399
      16.4.1 Demonstration example 400
      16.4.2 Program documentation 404
   16.5 Processing of the assembly code 406
      16.5.1 Compiling with KCSPM3 406
      16.5.2 Simulation by PBlazeIDE 407
      16.5.3 Reloading code via the JTAG port 410
      16.5.4 Compiling by PBlazeIDE 410
   16.6 Syntheses with PicoBlaze 411
   16.7 Bibliographic notes 412
   16.8 Suggested experiments 412
      16.8.1 Signed multiplication 412
      16.8.2 Multi-byte multiplication 412
      16.8.3 Barrel shift function 413
      16.8.4 Reverse function 413
      16.8.5 Binary-to-BCD conversion 413
      16.8.6 BCD-to-binary conversion 413
      16.8.7 Heartbeat circuit 413
      16.8.8 Rotating LED circuit 413
      16.8.9 Discrete LED dimmer 413

17 PicoBlaze I/O Interface 415
   17.1 Introduction 415
   17.2 Output port 416
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.2.1 Output instruction and timing</td>
<td>416</td>
</tr>
<tr>
<td>17.2.2 Output interface</td>
<td>417</td>
</tr>
<tr>
<td>17.3 Input port</td>
<td>418</td>
</tr>
<tr>
<td>17.3.1 Input instruction and timing</td>
<td>418</td>
</tr>
<tr>
<td>17.3.2 Input interface</td>
<td>419</td>
</tr>
<tr>
<td>17.4 Square program with a switch and seven-segment LED display interface</td>
<td>421</td>
</tr>
<tr>
<td>17.4.1 Output interface</td>
<td>421</td>
</tr>
<tr>
<td>17.4.2 Input interface</td>
<td>422</td>
</tr>
<tr>
<td>17.4.3 Assembly code development</td>
<td>424</td>
</tr>
<tr>
<td>17.4.4 HDL code development</td>
<td>431</td>
</tr>
<tr>
<td>17.5 Square program with a combinational multiplier and UART console</td>
<td>434</td>
</tr>
<tr>
<td>17.5.1 Multiplier interface</td>
<td>434</td>
</tr>
<tr>
<td>17.5.2 UART interface</td>
<td>435</td>
</tr>
<tr>
<td>17.5.3 Assembly code development</td>
<td>436</td>
</tr>
<tr>
<td>17.5.4 HDL code development</td>
<td>446</td>
</tr>
<tr>
<td>17.6 Bibliographic notes</td>
<td>449</td>
</tr>
<tr>
<td>17.7 Suggested experiments</td>
<td>449</td>
</tr>
<tr>
<td>17.7.1 Low-frequency counter I</td>
<td>449</td>
</tr>
<tr>
<td>17.7.2 Low-frequency counter II</td>
<td>449</td>
</tr>
<tr>
<td>17.7.3 Auto-scaled low-frequency counter</td>
<td>449</td>
</tr>
<tr>
<td>17.7.4 Basic reaction timer with a software timer</td>
<td>449</td>
</tr>
<tr>
<td>17.7.5 Basic reaction timer with a hardware timer</td>
<td>450</td>
</tr>
<tr>
<td>17.7.6 Enhanced reaction timer</td>
<td>450</td>
</tr>
<tr>
<td>17.7.7 Small-screen mouse scribble circuit</td>
<td>450</td>
</tr>
<tr>
<td>17.7.8 Full-screen mouse scribble circuit</td>
<td>450</td>
</tr>
<tr>
<td>17.7.9 Enhanced rotating banner</td>
<td>450</td>
</tr>
<tr>
<td>17.7.10 Pong game</td>
<td>450</td>
</tr>
<tr>
<td>17.7.11 Text editor</td>
<td>451</td>
</tr>
<tr>
<td><strong>18 PicoBlaze Interrupt Interface</strong></td>
<td>453</td>
</tr>
<tr>
<td>18.1 Introduction</td>
<td>453</td>
</tr>
<tr>
<td>18.2 Interrupt handling in PicoBlaze</td>
<td>453</td>
</tr>
<tr>
<td>18.2.1 Software processing</td>
<td>454</td>
</tr>
<tr>
<td>18.2.2 Timing</td>
<td>455</td>
</tr>
<tr>
<td>18.3 External interface</td>
<td>456</td>
</tr>
<tr>
<td>18.3.1 Single interrupt request</td>
<td>456</td>
</tr>
<tr>
<td>18.3.2 Multiple interrupt requests</td>
<td>456</td>
</tr>
<tr>
<td>18.4 Software development considerations</td>
<td>457</td>
</tr>
<tr>
<td>18.4.1 Interrupt as an alternative scheduling scheme</td>
<td>457</td>
</tr>
<tr>
<td>18.4.2 Development of an interrupt service routine</td>
<td>458</td>
</tr>
<tr>
<td>18.5 Design example</td>
<td>458</td>
</tr>
<tr>
<td>18.5.1 Interrupt interface</td>
<td>458</td>
</tr>
</tbody>
</table>