Digital Design Using VHDL
A Systems Approach

WILLIAM J. DALLY
Stanford University
R. CURTIS HARTING
Google, Inc.
TOR M. AAMODT
The University of British Columbia

CAMBRIDGE UNIVERSITY PRESS
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>Properties</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Dual functions</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Normal form</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>From equations to gates</td>
<td>48</td>
</tr>
<tr>
<td>3.6</td>
<td>Boolean expressions in VHDL</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Bibliographic notes</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>CMOS logic circuits</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Switch logic</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Switch model of MOS transistors</td>
<td>62</td>
</tr>
<tr>
<td>4.3</td>
<td>CMOS gate circuits</td>
<td>68</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Basic CMOS gate circuit</td>
<td>69</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Inverters, NANDs, and NORs</td>
<td>70</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Complex gates</td>
<td>72</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Tri-state circuits</td>
<td>75</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Circuits to avoid</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Bibliographic notes</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>Delay and power of CMOS circuits</td>
<td>82</td>
</tr>
<tr>
<td>5.1</td>
<td>Delay of static CMOS gates</td>
<td>82</td>
</tr>
<tr>
<td>5.2</td>
<td>Fan-out and driving large loads</td>
<td>85</td>
</tr>
<tr>
<td>5.3</td>
<td>Fan-in and logical effort</td>
<td>86</td>
</tr>
<tr>
<td>5.4</td>
<td>Delay calculation</td>
<td>89</td>
</tr>
<tr>
<td>5.5</td>
<td>Optimizing delay</td>
<td>92</td>
</tr>
<tr>
<td>5.6</td>
<td>Wire delay</td>
<td>94</td>
</tr>
<tr>
<td>5.7</td>
<td>Power dissipation in CMOS circuits</td>
<td>98</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Dynamic power</td>
<td>98</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Static power</td>
<td>99</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Power scaling</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Bibliographic notes</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>102</td>
</tr>
<tr>
<td>6</td>
<td>Combinational logic design</td>
<td>105</td>
</tr>
<tr>
<td>6.1</td>
<td>Combinational logic</td>
<td>105</td>
</tr>
<tr>
<td>6.2</td>
<td>Closure</td>
<td>106</td>
</tr>
<tr>
<td>6.3</td>
<td>Truth tables, minterms, and normal form</td>
<td>107</td>
</tr>
<tr>
<td>6.4</td>
<td>Implicants and cubes</td>
<td>110</td>
</tr>
<tr>
<td>6.5</td>
<td>Karnaugh maps</td>
<td>113</td>
</tr>
<tr>
<td>6.6</td>
<td>Covering a function</td>
<td>115</td>
</tr>
</tbody>
</table>
7 VHDL descriptions of combinational logic

7.1 The prime number circuit in VHDL
   7.1.1 A VHDL design entity
   7.1.2 The case statement
   7.1.3 The case? statement
   7.1.4 The if statement
   7.1.5 Concurrent signal assignment statements
   7.1.6 Selected signal assignment statements
   7.1.7 Conditional signal assignment statements
   7.1.8 Structural description
   7.1.9 The decimal prime number function

7.2 A testbench for the prime number circuit

7.3 Example: a seven-segment decoder

8 Combinational building blocks

8.1 Multi-bit notation
8.2 Decoders
8.3 Multiplexers
8.4 Encoders
8.5 Arbiters and priority encoders
8.6 Comparators
8.7 Shifters
8.8 Read-only memories
8.9 Read–write memories
8.10 Programmable logic arrays
8.11 Data sheets
8.12 Intellectual property

9 Combinational examples

9.1 Multiple-of-3 circuit
Part III Arithmetic circuits

10 Arithmetic circuits
10.1 Binary numbers
10.2 Binary addition
10.3 Negative numbers and subtraction
10.4 Multiplication
10.5 Division

11 Fixed- and floating-point numbers
11.1 Representation error: accuracy, precision, and resolution
11.2 Fixed-point numbers
  11.2.1 Representation
  11.2.2 Operations
11.3 Floating-point numbers
  11.3.1 Representation
  11.3.2 Denormalized numbers and gradual underflow
  11.3.3 Floating-point multiplication
  11.3.4 Floating-point addition/subtraction

12 Fast arithmetic circuits
12.1 Carry look-ahead
12.2 Booth recoding
12.3 Wallace trees
12.4 Synthesis notes

13 Arithmetic examples
13.1 Complex multiplication
13.2 Converting between fixed- and floating-point formats
Part IV Synchronous sequential logic

14 Sequential logic
14.1 Sequential circuits
14.2 Synchronous sequential circuits
14.3 Traffic-light controller
14.4 State assignment
14.5 Implementation of finite-state machines
14.6 VHDL implementation of finite-state machines

15 Timing constraints
15.1 Propagation and contamination delay
15.2 The D flip-flop
15.3 Setup- and hold-time constraints
15.4 The effect of clock skew
15.5 Timing examples
15.6 Timing and logic synthesis

16 Datapath sequential logic
16.1 Counters
   16.1.1 A simpler counter
   16.1.2 Up/down/load counter
   16.1.3 A timer
16.2 Shift registers
   16.2.1 A simple shift register
   16.2.2 Left/right/load (LRL) shift register
   16.2.3 Universal shifter/counter
16.3 Control and data partitioning
   16.3.1 Example: vending machine FSM
16.3.2 Example: combination lock
Summary
Exercises

17 Factoring finite-state machines
17.1 A light flasher
17.2 Traffic-light controller
Summary
Exercises

18 Microcode
18.1 Simple microcoded FSM
18.2 Instruction sequencing
18.3 Multi-way branches
18.4 Multiple instruction types
18.5 Microcode subroutines
18.6 Simple computer
Summary
Bibliographic notes
Exercises

19 Sequential examples
19.1 Divide-by-3 counter
19.2 SOS detector
19.3 Tic-tac-toe game
19.4 Huffman encoder/decoder
   19.4.1 Huffman encoder
   19.4.2 Huffman decoder
Summary
Bibliographic note
Exercises

Part V Practical design

20 Verification and test
20.1 Design verification
   20.1.1 Verification coverage
   20.1.2 Types of tests
   20.1.3 Static timing analysis
   20.1.4 Formal verification
   20.1.5 Bug tracking
20.2 Test
   20.2.1 Fault models
Contents xi

20.2.2 Combinational testing 457
20.2.3 Testing redundant logic 457
20.2.4 Scan 458
20.2.5 Built-in self-test (BIST) 459
20.2.6 Characterization 460
Summary 461
Bibliographic notes 462
Exercises 462

Part VI System design

21 System-level design 467
21.1 System design process 467
21.2 Specification 468
21.2.1 Pong 468
21.2.2 DES cracker 471
21.2.3 Music player 472
21.3 Partitioning 473
21.3.1 Pong 474
21.3.2 DES cracker 475
21.3.3 Music synthesizer 475
Summary 476
Bibliographic notes 477
Exercises 477

22 Interface and system-level timing 479
22.1 Interface timing 479
22.1.1 Always valid timing 479
22.1.2 Periodically valid signals 480
22.1.3 Flow control 481
22.2 Interface partitioning and selection 482
22.3 Serial and packetized interfaces 483
22.4 Isochronous timing 486
22.5 Timing tables 487
22.5.1 Event flow 488
22.5.2 Pipelining and anticipatory timing 488
22.6 Interface and timing examples 489
22.6.1 Pong 489
22.6.2 DES cracker 489
22.6.3 Music player 493
Summary 493
Exercises 494
## Contents

### 23 Pipelines
- 23.1 Basic pipelining
- 23.2 Example pipelines
- 23.3 Example: pipelining a ripple-carry adder
- 23.4 Pipeline stalls
- 23.5 Double buffering
- 23.6 Load balance
- 23.7 Variable loads
- 23.8 Resource sharing
- Summary
- Bibliographic notes
- Exercises

### 24 Interconnect
- 24.1 Abstract interconnect
- 24.2 Buses
- 24.3 Crossbar switches
- 24.4 Interconnection networks
- Summary
- Bibliographic notes
- Exercises

### 25 Memory systems
- 25.1 Memory primitives
  - 25.1.1 SRAM arrays
  - 25.1.2 DRAM chips
- 25.2 Bit-slicing and banking memory
- 25.3 Interleaved memory
- 25.4 Caches
- Summary
- Bibliographic notes
- Exercises

### Part VII Asynchronous logic

### 26 Asynchronous sequential circuits
- 26.1 Flow-table analysis
- 26.2 Flow-table synthesis: the toggle circuit
- 26.3 Races and state assignment
- Summary
- Bibliographic notes
- Exercises
### 27 Flip-flops

27.1 Inside a latch 566
27.2 Inside a flip-flop 568
27.3 CMOS latches and flip-flops 571
27.4 Flow-table derivation of the latch 572
27.5 Flow-table synthesis of a D flip-flop 574

Summary 576

Bibliographic notes 577

Exercises 577

### 28 Metastability and synchronization failure

28.1 Synchronization failure 580
28.2 Metastability 581
28.3 Probability of entering and leaving an illegal state 584
28.4 Demonstration of metastability 585

Summary 589

Bibliographic notes 590

Exercises 590

### 29 Synchronizer design

29.1 Where are synchronizers used? 592
29.2 Brute-force synchronizer 593
29.3 The problem with multi-bit signals 595
29.4 FIFO synchronizer 596

Summary 604

Bibliographic notes 605

Exercises 605

### Part VIII Appendix: VHDL coding style and syntax guide

#### Appendix A: VHDL coding style

A.1 Basic principles 611
A.2 All state should be in explicitly declared registers 612
A.3 Define combinational design entities so that they are easy to read 614
A.4 Assign all signals under all conditions 615
A.5 Keep design entities small 617
A.6 Large design entities should be structural 617
A.7 Use descriptive signal names 618
A.8 Use symbolic names for subfields of signals 618
A.9 Define constants 618
A.10 Comments should describe intention and give rationale, not state the obvious 619
A.11 Never forget you are defining hardware 620
A.12 Read and be a critic of VHDL code 620

Appendix B: VHDL syntax guide 622
B.1 Comments, identifiers, and keywords 623
B.2 Types 623
  B.2.1 Std_logic 624
  B.2.2 Boolean 624
  B.2.3 Integer 624
  B.2.4 Std_logic_vector 625
  B.2.5 Subtypes 625
  B.2.6 Enumeration 626
  B.2.7 Arrays and records 626
  B.2.8 Qualified expressions 627
B.3 Libraries, packages, and using multiple files 627
B.4 Design entities 628
B.5 Slices, concatenation, aggregates, operators, and expressions 629
B.6 Concurrent statements 631
  B.6.1 Concurrent signal assignment 632
  B.6.2 Component instantiation 634
B.7 Multiple signal drivers and resolution functions 636
B.8 Attributes 638
B.9 Process statements 640
  B.9.1 The process sensitivity list and execution timing 641
  B.9.2 Wait and report statements 644
  B.9.3 If statements 644
  B.9.4 Case and matching case statements 644
  B.9.5 Signal and variable assignment statements 646
B.10 Synthesizable process statements 648
  B.10.1 Type 1: purely combinational 649
  B.10.2 Type 2: edge-sensitive 649
  B.10.3 Type 3: edge-sensitive with asynchronous reset 650

References 653
Index of VHDL design entities 658
Subject index 660