Brief Contents

List of Figures xv
List of Tables xxvii
Preface xxix

1 Computer Fundamentals and Performance Evaluation 1
2 A Brief Introduction to Logic Circuits and Verilog HDL 19
3 Computer Arithmetic Algorithms and Implementations 63
4 Instruction Set Architecture and ALU Design 111
5 Single-Cycle CPU Design in Verilog HDL 143
6 Exceptions and Interrupts Handling and Design in Verilog HDL 170
7 Multiple-Cycle CPU Design in Verilog HDL 192
8 Design of Pipelined CPU with Precise Interrupt in Verilog HDL 212
9 Floating-Point Algorithms and FPU Design in Verilog HDL 266
10 Design of Pipelined CPU with FPU in Verilog HDL 323
11 Memory Hierarchy and Virtual Memory Management 353
12 Design of Pipelined CPU with Caches and TLBs in Verilog HDL 386
13 Multithreading CPU and Multicore CPU Design in Verilog HDL 425
14 Input/Output Interface Controller Design in Verilog HDL 443
15 High-Performance Computers and Interconnection Networks 509
Contents

List of Figures xi
List of Tables xxvii
Preface xxix

1 Computer Fundamentals and Performance Evaluation 1
  1.1 Overview of Computer Systems 1
     1.1.1 Organization of Computer Systems 1
     1.1.2 A Brief History of the Computer 2
     1.1.3 Instruction Set Architecture 4
     1.1.4 CISC and RISC 7
     1.1.5 Definitions of Some Units 8
  1.2 Basic Structure of Computers 8
     1.2.1 Basic Structure of RISC CPU 8
     1.2.2 Multithreading and Multicore CPUs 10
     1.2.3 Memory Hierarchy and Virtual Memory Management 11
     1.2.4 Input/Output Interfaces and Buses 12
  1.3 Improving Computer Performance 13
     1.3.1 Computer Performance Evaluation 13
     1.3.2 Trace-Driven Simulation and Execution-Driven Simulation 14
     1.3.3 High-Performance Computers and Interconnection Networks 15
  1.4 Hardware Description Languages 16
     Exercises 18

2 A Brief Introduction to Logic Circuits and Verilog HDL 19
  2.1 Logic Gates 19
  2.2 Logic Circuit Design in Verilog HDL 22
  2.3 CMOS Logic Gates 25
     2.3.1 CMOS Inverter 25
     2.3.2 CMOS NAND and NOR Gates 26
  2.4 Four Levels/Styles of Verilog HDL 28
     2.4.1 Transistor Switch Level 29
     2.4.2 Logic Gate Level 30
     2.4.3 Dataflow Style 31
     2.4.4 Behavioral Style 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Combinational Circuit Design</td>
<td></td>
</tr>
<tr>
<td>2.5.1</td>
<td>Multiplexer</td>
<td>34</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Decoder</td>
<td>35</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Encoder</td>
<td>36</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Demultiplexer</td>
<td>39</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Barrel Shifter</td>
<td>39</td>
</tr>
<tr>
<td>2.6</td>
<td>Sequential Circuit Design</td>
<td></td>
</tr>
<tr>
<td>2.6.1</td>
<td>D Latch and D Flip-Flop</td>
<td>42</td>
</tr>
<tr>
<td>2.6.2</td>
<td>JK Latch and JK Flip-Flop</td>
<td>46</td>
</tr>
<tr>
<td>2.6.3</td>
<td>T Latch and T Flip-Flop</td>
<td>47</td>
</tr>
<tr>
<td>2.6.4</td>
<td>Shift Register</td>
<td>49</td>
</tr>
<tr>
<td>2.6.5</td>
<td>FIFO Buffer</td>
<td>49</td>
</tr>
<tr>
<td>2.6.6</td>
<td>Finite State Machine and Counter Design</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>59</td>
</tr>
<tr>
<td>3</td>
<td>Computer Arithmetic Algorithms and Implementations</td>
<td>63</td>
</tr>
<tr>
<td>3.1</td>
<td>Binary Integers</td>
<td></td>
</tr>
<tr>
<td>3.1.1</td>
<td>Binary and Hexadecimal Representations</td>
<td>63</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Unsigned Binary Integers</td>
<td>64</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Signed Binary Integers (2's Complement Notation)</td>
<td>64</td>
</tr>
<tr>
<td>3.2</td>
<td>Binary Addition and Subtraction</td>
<td></td>
</tr>
<tr>
<td>3.2.1</td>
<td>Ripple Adder and Subtractor Design</td>
<td>65</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Carry-Lookahead Adder Design</td>
<td>70</td>
</tr>
<tr>
<td>3.3</td>
<td>Binary Multiplication Algorithms</td>
<td></td>
</tr>
<tr>
<td>3.3.1</td>
<td>Unsigned Multiplier Design</td>
<td>74</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Signed Multiplier Design</td>
<td>75</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Wallace Tree</td>
<td>79</td>
</tr>
<tr>
<td>3.4</td>
<td>Binary Division Algorithms</td>
<td></td>
</tr>
<tr>
<td>3.4.1</td>
<td>Restoring Division Algorithm</td>
<td>84</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Nonrestoring Division Algorithm</td>
<td>87</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Goldschmidt Division Algorithm</td>
<td>90</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Newton–Raphson Division Algorithm</td>
<td>92</td>
</tr>
<tr>
<td>3.5</td>
<td>Binary Square Root Algorithms</td>
<td></td>
</tr>
<tr>
<td>3.5.1</td>
<td>Restoring Square Root Algorithm</td>
<td>95</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Nonrestoring Square Root Algorithm</td>
<td>99</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Goldschmidt Square Root Algorithm</td>
<td>104</td>
</tr>
<tr>
<td>3.5.4</td>
<td>Newton–Raphson Square Root Algorithm</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>Instruction Set Architecture and ALU Design</td>
<td>111</td>
</tr>
<tr>
<td>4.1</td>
<td>Instruction Set Architecture</td>
<td></td>
</tr>
<tr>
<td>4.1.1</td>
<td>Operand Types</td>
<td>111</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Little Endian and Big Endian</td>
<td>112</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Instruction Types</td>
<td>113</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Instruction Architecture</td>
<td>115</td>
</tr>
<tr>
<td>4.1.5</td>
<td>Addressing Modes</td>
<td>116</td>
</tr>
<tr>
<td>4.2</td>
<td>MIPS Instruction Format and Registers</td>
<td></td>
</tr>
<tr>
<td>4.2.1</td>
<td>MIPS Instruction Format</td>
<td>117</td>
</tr>
<tr>
<td>4.2.2</td>
<td>MIPS General-Purpose Registers</td>
<td>117</td>
</tr>
</tbody>
</table>
4.3 MIPS Instructions and AsmSim Tool 118
  4.3.1 Some Typical MIPS Instructions 118
  4.3.2 Supporting Subroutine Call and Pointer 121
  4.3.3 AsmSim – A MIPS Assembler and Simulator 125
4.4 ALU Design 136
  Exercises 140

5 Single-Cycle CPU Design in Verilog HDL 143
5.1 The Circuits Required for Executing an Instruction 143
  5.1.1 The Circuits Required by Instruction Fetch 143
  5.1.2 The Circuits Required by Instruction Execution 143
5.2 Register File Design 148
  5.2.1 Register File Schematic Circuit 149
  5.2.2 Register File Design in Dataflow-Style Verilog HDL 150
  5.2.3 Register File Design in Behavioral-Style Verilog HDL 153
5.3 Single-Cycle CPU Datapath Design 154
  5.3.1 Using Multiplexers 154
  5.3.2 Single-Cycle CPU Schematic Circuit 156
  5.3.3 Single-Cycle CPU Verilog HDL Codes 158
5.4 Single-Cycle CPU Control Unit Design 160
  5.4.1 Logic Design of the Control Unit 160
  5.4.2 Verilog HDL Codes of the Control Unit 163
5.5 Test Program and Simulation Waveform 164
  Exercises 166

6 Exceptions and Interrupts Handling and Design in Verilog HDL 170
6.1 Exceptions and Interrupts 170
  6.1.1 Types of Exceptions and Interrupts 170
  6.1.2 Polled Interrupt and Vectored Interrupt 171
  6.1.3 Return from Exception/Interrupt 173
  6.1.4 Interrupt Mask and Interrupt Nesting 174
  6.1.5 Interrupt Priority 175
6.2 Design of CPU with Exception and Interrupt Mechanism 176
  6.2.1 Exception/Interrupt Handling and Related Registers 177
  6.2.2 Instructions Related to Exceptions and Interrupts 177
  6.2.3 Schematic of the CPU 178
  6.2.4 The Verilog HDL Codes of the CPU 180
6.3 The CPU Exception and Interrupt Tests 187
  6.3.1 Test Program and Data 187
  6.3.2 Test Waveform and Explanations 191
  Exercises 191

7 Multiple-Cycle CPU Design in Verilog HDL 192
7.1 Dividing Instruction Execution into Several Clock Cycles 192
  7.1.1 Instruction Fetch Cycle 192
  7.1.2 Instruction Decode and Source Operand Read Cycle 193
  7.1.3 Execution Cycle 195
  7.1.4 Memory Access Cycle 197
  7.1.5 Write Back Cycle 197
7.2 Multiple-Cycle CPU Schematic and Verilog HDL Codes
  7.2.1 Multiple-Cycle CPU Structure
  7.2.2 Multiple-Cycle CPU Verilog HDL Codes

7.3 Multiple-Cycle CPU Control Unit Design
  7.3.1 State Transition Diagram of the Control Unit
  7.3.2 Circuit Structure of the Control Unit
  7.3.3 Next State Function
  7.3.4 Output Function of Control Signal
  7.3.5 Verilog HDL Codes of the Control Unit

7.4 Memory and Test Program
  7.4.1 Memory Design
  7.4.2 Test Program
  7.4.3 Multiple-Cycle CPU Simulation Waveforms

Exercises

8 Design of Pipelined CPU with Precise Interrupt in Verilog HDL
  8.1 Pipelining
    8.1.1 The Concept of Pipelining
    8.1.2 The Baseline of Pipelined CPU
  8.2 Pipeline Hazards and Solutions
    8.2.1 Structural Hazards and Solutions
    8.2.2 Data Hazards and Internal Forwarding
    8.2.3 Control Hazards and Delayed Branch
  8.3 The Circuit of the Pipelined CPU and Verilog HDL Codes
    8.3.1 The Circuit of the Pipelined CPU
    8.3.2 Verilog HDL Codes of the Pipelined CPU
    8.3.3 Test Program and Simulation Waveforms
  8.4 Precise Interrupts/Exceptions in Pipelined CPU
    8.4.1 Exception and Interrupt Types and Related Registers
    8.4.2 Dealing with Exceptions in Pipelined CPU
  8.5 Design of Pipelined CPU with Precise Interrupt/Exception
    8.5.1 The Structure of the Pipelined CPU
    8.5.2 The Verilog HDL Codes of the Pipelined CPU
    8.5.3 Test Program for Exception and Interrupt
    8.5.4 Simulation Waveforms of Exception and Interrupt

Exercises

9 Floating-Point Algorithms and FPU Design in Verilog HDL
  9.1 IEEE 754 Floating-Point Data Formats
  9.2 Converting between Floating-Point Number and Integer
    9.2.1 Converting Floating-Point Number to Integer
    9.2.2 Converting an Integer to a Floating-Point Number
  9.3 Floating-Point Adder (FADD) Design
    9.3.1 Floating-Point Addition Algorithm
    9.3.2 Circuit and Verilog HDL Codes of FADD
    9.3.3 Pipelined FADD Design
  9.4 Floating-Point Multiplier (FMUL) Design
    9.4.1 Floating-Point Multiplication Algorithm
    9.4.2 Circuit and Verilog HDL Codes of FMUL
    9.4.3 Pipelined Wallace Tree FMUL Design
Contents

9.5 Floating-Point Divider (FDIV) Design 302
  9.5.1 Floating-Point Division Algorithm 303
  9.5.2 Circuit and Verilog HDL Codes of FDIV 303

9.6 Floating-Point Square Root (FSQRT) Design 312
  9.6.1 Floating-Point Square Root Algorithm 312
  9.6.2 Circuit and Verilog HDL Codes of FSQRT 314

Exercises 321

10 Design of Pipelined CPU with FPU in Verilog HDL 323
  10.1 CPU/FPU Pipeline Model 323
    10.1.1 FPU Instructions 323
    10.1.2 Basic Model of CPU/FPU Pipeline 324
  10.2 Design of Register File with Two Write Ports 326
  10.3 Data Dependency and Pipeline Stalls 328
    10.3.1 Internal Forwarding 329
    10.3.2 Pipeline Stall Caused by lwz1 and swz1 Instructions 331
    10.3.3 Pipeline Stall Caused by Division and Square Root 334
  10.4 Pipelined CPU/FPU Design in Verilog HDL 335
    10.4.1 Circuit of Pipelined CPU/FPU 335
    10.4.2 Circuit of the Floating-Point Unit 336
    10.4.3 Circuit of the Integer Unit 337
    10.4.4 Verilog HDL Codes of the Pipelined CPU/FPU 337
  10.5 Memory Modules and Pipelined CPU/FPU Test 345
    10.5.1 Instruction Memory and Data Memory 345
    10.5.2 Test Program of Pipelined CPU/FPU 346
    10.5.3 Test Waveforms of Pipelined CPU/FPU 348

Exercises 351

11 Memory Hierarchy and Virtual Memory Management 353
  11.1 Memory 353
    11.1.1 Static Random Access Memory 354
    11.1.2 Dynamic Random Access Memory 354
    11.1.3 Read-Only Memory 355
    11.1.4 Content Addressable Memory 356
    11.1.5 Asynchronous Memory and Synchronous Memory 358
    11.1.6 Memory Hierarchy 359
  11.2 Cache Memory 359
    11.2.1 Mapping Methods of Caches 360
    11.2.2 Cache Block Replacement Algorithms 362
    11.2.3 Cache Write Policies 364
    11.2.4 Data Cache Design in Verilog HDL 365
  11.3 Virtual Memory Management and TLB Design 367
    11.3.1 Relationship of Main Memory and Virtual Memory 368
    11.3.2 Segmentation Management 370
    11.3.3 Paging Management 370
    11.3.4 Translation Lookaside Buffer Design 372
    11.3.5 Accessing Cache and TLB in Parallel 376
  11.4 Mechanism of TLB-Based MIPS Memory Management 377
    11.4.1 MIPS Virtual Memory Space 378
    11.4.2 Organization of MIPS TLB 379
    11.4.3 TLB Manipulation by Software 381

Exercises 384
## Design of Pipelined CPU with Caches and TLBs in Verilog HDL

### Overall Structure of Caches and TLBs

- **12.2 Design of Circuits Related to Caches**
  - Instruction Cache and Its Verilog HDL Codes
  - Memory Interface to Instruction and Data Caches
  - Pipeline Halt Circuit for Cache Misses

### Design of Circuits Related to TLB

- Instruction TLB (ITLB) and Data TLB (DTLB)
- Generating Exception Signals during TLB Misses
- Registers Related to TLB Miss Exceptions
- Circuit Dealing with TLB Miss Exceptions

### Design of CPU with Caches and TLBs

- Structure of CPU with Caches and TLBs
- Verilog HDL Codes of CPU with Caches and TLBs

### Simulation Waveforms of CPU with Caches and TLBs

### Exercises

## Multithreading CPU and Multicore CPU Design in Verilog HDL

### Overview of Multithreading CPUs

- Basic Concept of Multiple Threads
- Basic Structure of Multithreading CPUs

### Multithreading CPU Design

- Thread Selection Method
- Circuit of the Multithreading CPU
- Verilog HDL Codes of the Multithreading CPU
- Simulation Waveforms of Multithreading CPU

### Overview of Multicore CPUs

- Basic Concept of Multicore CPUs
- Basic Structure of Multicore CPUs

### Multicore CPU Design

- Overall Structure of the Multicore CPU
- Verilog HDL Codes of the Multicore CPU
- Simulation Waveforms of the Multicore CPU

### Exercises

## Input/Output Interface Controller Design in Verilog HDL

### Overview of Input/Output Interface Controllers

- Input/Output Address Space and Instructions
- Input/Output Polling and Interrupt
- Direct Memory Access (DMA)
- Synchronization Schemes of Buses

### Error Detection and Correction

- Parity Check
- Error Correction Code (Extended Hamming Code)
- Cyclic Redundancy Check

### Universal Asynchronous Receiver Transmitter

### PS/2 Keyboard/Mouse Interface Design

- PS/2 Keyboard Interface
- PS/2 Mouse Interface

### Video Graphics Array (VGA) Interface Design

- VGA Interface Design in Verilog HDL
- Displaying Scan Codes of Keyboard Keys on VGA

### Exercises
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.6 Input/Output Buses</td>
<td>483</td>
</tr>
<tr>
<td>14.6.1 I2C Serial Bus and I2C EEPROM Access</td>
<td>483</td>
</tr>
<tr>
<td>14.6.2 PCI Parallel Bus</td>
<td>500</td>
</tr>
<tr>
<td>Exercises</td>
<td>507</td>
</tr>
<tr>
<td><strong>15 High-Performance Computers and Interconnection Networks</strong></td>
<td>509</td>
</tr>
<tr>
<td>15.1 Category of High-Performance Computers</td>
<td>509</td>
</tr>
<tr>
<td>15.1.1 Shared-Memory Parallel Multiprocessor Systems</td>
<td>509</td>
</tr>
<tr>
<td>15.1.2 Message-Passing Distributed Multicomputer Systems</td>
<td>509</td>
</tr>
<tr>
<td>15.2 Shared-Memory Parallel Multiprocessor Systems</td>
<td>510</td>
</tr>
<tr>
<td>15.2.1 Centralized Shared-Memory (SMP) Systems</td>
<td>510</td>
</tr>
<tr>
<td>15.2.2 Distributed Shared-Memory (DSM) Systems</td>
<td>512</td>
</tr>
<tr>
<td>15.3 Inside of Interconnection Networks</td>
<td>514</td>
</tr>
<tr>
<td>15.4 Topological Properties of Interconnection Networks</td>
<td>516</td>
</tr>
<tr>
<td>15.4.1 Node Degree</td>
<td>516</td>
</tr>
<tr>
<td>15.4.2 Network Diameter</td>
<td>517</td>
</tr>
<tr>
<td>15.4.3 Average Distance</td>
<td>518</td>
</tr>
<tr>
<td>15.4.4 Bisection Bandwidth</td>
<td>518</td>
</tr>
<tr>
<td>15.5 Some Popular Topologies of Interconnection Networks</td>
<td>518</td>
</tr>
<tr>
<td>15.5.1 2D/3D Mesh</td>
<td>518</td>
</tr>
<tr>
<td>15.5.2 2D/3D Torus</td>
<td>518</td>
</tr>
<tr>
<td>15.5.3 Hypercubes</td>
<td>519</td>
</tr>
<tr>
<td>15.5.4 Tree and Fat Tree</td>
<td>520</td>
</tr>
<tr>
<td>15.5.5 Summary</td>
<td>521</td>
</tr>
<tr>
<td>15.6 Collective Communications</td>
<td>521</td>
</tr>
<tr>
<td>15.6.1 One-to-All Broadcast</td>
<td>522</td>
</tr>
<tr>
<td>15.6.2 All-to-All Broadcast</td>
<td>522</td>
</tr>
<tr>
<td>15.6.3 One-to-All Personalized Communication</td>
<td>523</td>
</tr>
<tr>
<td>15.6.4 All-to-All Personalized Communication</td>
<td>523</td>
</tr>
<tr>
<td>15.7 Low-Node-Degree Short-Diameter Interconnection Networks</td>
<td>524</td>
</tr>
<tr>
<td>15.7.1 Cube-Connected Cycle (CCC)</td>
<td>525</td>
</tr>
<tr>
<td>15.7.2 Hierarchical Cubic Networks (HCNs)</td>
<td>525</td>
</tr>
<tr>
<td>15.7.3 Dual-Cube</td>
<td>525</td>
</tr>
<tr>
<td>15.7.4 Metacube</td>
<td>528</td>
</tr>
<tr>
<td>15.7.5 Recursive Dual-Net (RDN)</td>
<td>531</td>
</tr>
<tr>
<td>Exercises</td>
<td>535</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>536</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>539</td>
</tr>
</tbody>
</table>