About the Author xi

1. The Fundamentals
   Why Use FPGAs? 1
   Applications 3
   Some Technology Background 4
      Fusible-link Technology 4
      FPGA Programming Technologies 7
   Instant Summary 12

2. FPGA Architectures
   More on Programming Technologies 14
      SRAM-based Devices 14
      Antifuse-based Devices 16
      E²PROM/FLASH-based Devices 17
      Hybrid FLASH-SRAM Devices 18
   Fine-, Medium-, and Coarse-grained Architectures 18
   Logic Blocks 19
      MUX-based 19
      LUT-based 20
      LUT versus Distributed RAM versus SR 22
   CLBs versus LABs versus Slices 23
      Logic Cells/Logic Elements 24
      Slicing and Dicing 24
      CLBs and LABs 25
      Distributed RAMs and Shift Registers 26
   Embedded RAMs 27
   Embedded Multipliers, Adders, etc. 27
   Embedded Processor Cores 29
      Hard Microprocessor Cores 30
      Soft Microprocessor Cores 31
   Clock Managers 32
      Clock Trees 32
      Clock Managers 33
3. Programming (Configuring) an FPGA

Configuration Cells 50
Antifuse-based FPGAs 51
SRAM-based FPGAs 51
  Programming Embedded (Block) RAMs, Distributed RAMs, etc. 52
  Multiple Programming Chains 53
  Quickly Reinitializing the Device 53
Using the Configuration Port 53
  Serial Load with FPGA as Master 54
  Parallel Load with FPGA as Master 55
  Parallel Load with FPGA as Slave 56
  Serial Load with FPGA as Slave 57
Using the JTAG Port 58
Using an Embedded Processor 59
Instant Summary 60

4. FPGA vs. ASIC Designs

  When You Switch from ASIC to FPGA Design, or Vice Versa 62
Coding Styles 62
Pipelining and Levels of Logic 62
  Levels of Logic 64
Asynchronous Design Practices 65
  Asynchronous Structures 65
  Combinational Loops 65
  Delay Chains 65
Clock Considerations 65
  Clock Domains 65
  Clock Balancing 65
  Clock Gating versus Clock Enabling 66
Contents

PLLs and Clock Conditioning Circuitry 66
Reliable Data Transfer across Multiclock Domains 66

Register and Latch Considerations 67
Latches 67
Flip-flops with both “Set” and “Reset” Inputs 67
Global Resets and Initial Conditions 67

Resource Sharing (Time-Division Multiplexing) 67
Use It or Lose It! 67
But Wait, There’s More 68

State Machine Encoding 68
Test Methodologies 69

Migrating ASIC Designs to FPGAs and Vice Versa 69
Alternative Design Scenarios 69

Instant Summary 73

5. “Traditional” Design Flows 76

Schematic-based Design Flows 81
Back-end Tools like Layout 81
CAE + CAD = EDA 81
A Simple (early) Schematic-driven ASIC Flow 81
A Simple (early) Schematic-driven FPGA Flow 83
Flat versus Hierarchical Schematics 86
Schematic-driven FPGA Design Flows Today 88

HDL-based Design Flows 89
Advent of HDL-based Flows 89
A Plethora of HDLs 96
Points to Ponder 103

Instant Summary 106

6. Other Design Flows 108

C/C++-based Design Flows 110
C versus C++ and Concurrent versus Sequential 110
SystemC-based Flows 112
Augmented C/C++-based Flows 117
Pure C/C++-based Flows 120
Different Levels of Synthesis Abstraction 123
Mixed-language Design and Verification Environments 124

DSP-Based Design Flows 125
Alternative DSP Implementations 126
FPGA-centric Design Flows for DSPs 131
Mixed DSP and VHDL/Verilog etc. Environments 139

www.newnespress.com
Embedded Processor-based Design Flows 140
   Hard versus Soft Cores 142
   Partitioning a Design into Its Hardware and Software Components 145
   Using an FPGA as Its Own Development Environment 147
   Improving Visibility in the Design 147
   A Few Coverification Alternatives 148
Instant Summary 153

7. Using Design Tools 156

Simulation Tools
   Event-driven Logic Simulators 156
   Logic Values and Different Logic Value Systems 158
   Mixed-language Simulation 159
   Alternative Delay Formats 160
   Cycle-based Simulators 163
   Choosing a Logic Simulator 165

Synthesis (Logic/HDL versus Physically Aware) 166
   Logic/HDL Synthesis Technology 166
   Physically Aware Synthesis Technology 167
   Retiming, Replication, and Resynthesis 168

Timing Analysis 169
   Static Timing Analysis 169
   Statistical Static Timing Analysis 170

Verification in General 171
   Verification IP 171
   Verification Environments and Creating Testbenches 173
   Analyzing Simulation Results 174

Formal Verification 174
   Different Flavors of Formal Verification 174
   Terminology and Definitions 176
   Alternative Assertion/Property Specification Techniques 178
   Static Formal versus Dynamic Formal 179

Miscellaneous 182
   HDL to C Conversion 182
   Code Coverage 182
   Performance Analysis 183

Instant Summary 184

8. Choosing the Right Device 185

Choosing Technology 187
Contents

Basic Resources and Packaging 187
General-purpose I/O Interfaces 188
Embedded Multipliers, RAMs, etc. 188
Embedded Processor Cores 189
Gigabit I/O Capabilities 189
IP Availability 189
Speed Grades 190
Future FPGA Developments 191
Instant Summary 196

Index 197