Pearson New International Edition

VHDL for Engineers

Kenneth L. Short

ALWAYS LEARNING™

PEARSON®

Pearson New International Edition

VHDL for Engineers

Kenneth L. Short

PEARSON®

VHDL for Engineers

Table of Contents

Cover

Title

Content

List of Figures

List of Programs

Preface

1 Digital Design Using VHDL and PLDs

- 1.1 VHDL/PLD Design Methodology
- 1.2 Requirements Analysis and Specification
- 1.3 VHDL Design Description
- 1.4 Verification Using Simulation
- 1.5 Testbenches
- 1.6 Functional (Behavioral) Simulation
- 1.7 Programmable Logic Devices (PLDs)
- 1.8 SPLDs and the 22V10
- 1.9 Logic Synthesis for the Target PLD
- 1.10 Place-and-Route and Timing Simulation
- 1.11 Programming and Verifying a Target PLD
- 1.12 VHDL/PLD Design Methodology Advantages
- 1.13 VHDLs Development
- 1.14 VHDL for Synthesis versus VHDL for Simulation
- 1.15 This Books Primary Objective

2 Entities, Architectures, and Coding Styles

Pearson

Table of Contents

2.1 Design Units, Library Units, and Design Entities

2.2 Entity Declaration

2.3 VHDL Syntax Definitions

2.4 Port Modes

2.5 Architecture Body

2.6 Coding Styles

2.7 Synthesis Results versus Coding Style

2.8 Levels of Abstraction and Synthesis

2.9 Design Hierarchy and Structural Style

3 Signals and Data Types

3.1 Object Classes and Object Types

3.2 Signal Objects

3.3 Scalar Types

3.4 Type Std_Logic

3.5 Scalar Literals and Scalar Constants

3.6 Composite Types

3.7 Arrays

3.8 Types Unsigned and Signed

3.9 Composite Literals and Composite Constants

3.10 Integer Types

3.11 Port Types for Synthesis

3.12 Operators and Expressions

4 Dataflow Style Combinational Design

4.1 Logical Operators

4.2 Signal Assignments in Dataflow Style Architectures

4.3 Selected Signal Assignment

4.4 Type Boolean and the Relational Operators

4.5 Conditional Signal Assignment

Table of Contents

4.6 Priority Encoders

4.7 Dont Care Inputs and Outputs

4.8 Decoders

4.9 Table Lookup

4.10 Three-state Buffers

4.11 Avoiding Combinational Loops

5 Behavioral Style Combinational Design

5.1 Behavioral Style Architecture

5.2 Process Statement

5.3 Sequential Statements

5.4 Case Statement

5.5 If Statement

5.6 Loop Statement

5.7 Variables

5.8 Parity Detector Example

5.9 Synthesis of Processes Describing Combinational Systems

6 Event-Driven Simulation

6.1 Simulator Approaches

6.2 Elaboration

6.3 Signal Drivers

6.4 Simulator Kernel Process

6.5 Simulation Initialization

6.6 Simulation Cycles

6.7 Signals versus Variables

6.8 Delta Delays

6.9 Delta Delays and Combinational Feedback

6.10 Multiple Drivers

6.11 Signal Attributes

Table of Contents

7 Testbenches for Combinational Designs

- 7.1 Design Verification
- 7.2 Functional Verification of Combinational Designs
- 7.3 A Simple Testbench
- 7.4 Physical Types
- 7.5 Single Process Testbench
- 7.6 Wait Statements
- 7.7 Assertion and Report Statements
- 7.8 Records and Table Lookup Testbenches
- 7.9 Testbenches That Compute Stimulus and Expected Results
- 7.10 Predefined Shift Operators
- 7.11 Stimulus Order Based on UUT Functionality
- 7.12 Comparing a UUT to a Behavioral Intent Model
- 7.13 Code Coverage and Branch Coverage
- 7.14 Post-Synthesis and Timing Verifications for CombinationalDesigns
- 7.15 Timing Models Using VITAL and SDF

8 Latches and Flip-flops

- 8.1 Sequential Systems and Their Memory Elements
- 8.2 D Latch
- 8.3 Detecting Clock Edges
- 8.4 D Flip-flops
- 8.5 Enabled (Gated) Flip-flop
- 8.6 Other Flip-flop Types
- 8.7 PLD Primitive Memory Elements
- 8.8 Timing Requirements and Synchronous Input Data

9 Multibit Latches, Registers, Counters,and Memory

- 9.1 Multibit Latches and Registers
- 9.2 Shift Registers

Table of Contents

9.3 Shift Register Counters

9.4 Counters

9.5 Detecting Non-clock Signal Edges

9.6 Microprocessor Compatible Pulse Width Modulated SignalGenerator

9.7 Memories

10 Finite State Machines

10.1 Finite State Machines

10.2 FSM State Diagrams

10.3 Three Process FSM VHDL Template

10.4 State Diagram Development

10.5 Decoder for an Optical Shaft Encoder

10.6 State Encoding and State Assignment

10.7 Supposedly Safe FSMs

10.8 Inhibit Logic FSM Example

11 ASM Charts and RTL Design

11.1 Algorithmic State Machine Charts

11.2 Converting ASM Charts to VHDL

11.3 System Architecture

11.4 Successive Approximation Register Design Example

11.5 Sequential Multiplier Design

12 Subprograms

12.1 Subprograms

12.2 Functions

12.3 Procedures

12.4 Array Attributes and Unconstrained Arrays

12.5 Overloading Subprograms and Operators

12.6 Type Conversions

13 Packages

Table of Contents

13.1 Packages and Package Bodies

13.2 Standard and De Facto Standard Packages

13.3 Package STD_LOGIC_1164

13.4 Package NUMERIC_STD (IEEE Std 1076.3)

13.5 Package STD_LOGIC_ARITH

13.6 Packages for VHDL Text Output

14 Testbenches for Sequential Systems

14.1 Simple Sequential Testbenches

14.2 Generating a System Clock

14.3 Generating the System Reset

14.4 Synchronizing Stimulus Generation and Monitoring

14.5 Testbench for Successive Approximation Register

14.6 Determining a Testbench Stimulus for a Sequential System

14.7 Using Procedures for Stimulus Generation

14.8 Output Verification in Stimulus Procedures

14.9 Bus Functional Models

14.10 Response Monitors

15 Modular Design and Hierarchy

15.1 Modular Design, Partitioning, and Hierarchy

15.2 Design Units and Library Units

15.3 Design Libraries

15.4 Using Library Units

15.5 Direct Design Entity Instantiation

15.6 Components and Indirect Design Entity Instantiation

15.7 Configuration Declarations

15.8 Component Connections

15.9 Parameterized Design Entities

15.10 Library of Parameterized Modules (LPM)

Pearson

Table of Contents

15.11 Generate Statement

16 More Design Examples

16.1 Microprocessor-Compatible QuadratureDecoder/Counter Design

16.2 Verification of Quadrature Decoder/Counter

16.3 Parameterized Quadrature Decoder/Counter

16.4 Electronic Safe Design

16.5 Verification of Electronic Safe

16.6 Encoder for RF Transmitter Design

Appendix VHDL Attributes

Bibliography

Index

Pearson