VHDL - Expanded

4 days, Basic Level

"One of the best language courses I've taken.

Overview
VHDL is a feature-rich hardware description language, well suited to the synthesis and verification of complex system-on-a-chip ASICs and FPGAs. This course, first in the *Expert Designer* series, is a hardware-oriented VHDL primer for the digital design or verification engineer. Through real-world lecture insights and lab examples, participants in this comprehensive, hands-on course will learn to write synthesis-friendly, simulator-efficient code for progressively more complex logic blocks. They'll acquire confidence in utilizing the more powerful aspects of the language, while gaining mastery over its intricacies. The course focuses 70% on RTL code for synthesis, and 30% on testbench code for simulation.

The course may be customized for company specific topics and areas of focus.

Benefits
Upon completion of this course, students will:
- develop high-level behavioral models to capture design specifications
- refine models to synthesizable register-transfer-level detailed designs
- develop test bench models
- use simulation tools to test and debug models
- identify coding styles that enhance correctness and maintainability of models

Intended Audience
This course is recommended for designers with little or no knowledge of VHDL who need to gain proficiency in using the language for hardware design.

Prerequisites
Students need to be familiar with the basics of digital design such as shift registers, adders, multiplexors and finite state machines. Some background in computer organization would be helpful.

Follow-on course:

Advanced VHDL Coding Styles for Synthesis & Verification

Training Approach
This is an intensive, interactive course, which is approximately 50% lecture and 50% lab. Questions are highly encouraged. The final day concludes with a comprehensive lab project

where the students will have a chance to take a design from beginning to end using most of the information they have learned in the class.

Course Outline

Day 1

Introduction to VHDL modeling concepts
- uses of modeling in the design flow
- basic VHDL concepts through examples
- lexical elements

Scalar data types and operation
- constants and variables
- predefined types and operations
- subtypes
- predefined subtypes
- user-defined types
- integer types
- enumeration types
- expressions and operators

Sequential statements
- if statements
- case statements
- loop and exit statements
- while loops
- for loops
- assert statements

Composite data types and operations
- arrays, aggregates and array attributes
- array indexing, slicing and operations
- unconstrained array types
- predefined array types
- unconstrained array ports

Lab: Pulse-Width Modulator

Behavioral modeling
- review of entities and architectures
- signal assignment with delay
- attributes of signals
- wait statements
- delta delays
- inertial delay
- concurrent assignments

Structural modeling
- direct entity instantiation and port maps
- positional and named association
- association with expression

Design processing
- libraries
- library and use clauses
- analysis order
- elaboration

Lab: ALU for the eLucid-8 Microcontroller

Day 2

Subprograms
- procedures & return statements
- procedure parameters
- default parameter values
- unconstrained array parameters
- concurrent procedure call statements
- functions
- modeling using functions
- the now function
- overloading subprograms and operator symbols

Packages
- package declarations
- use clauses
- subprograms in packages
- package bodies
- predefined packages

Lab: Packages for Timing Checks and Test Utilities

Aliases
- data object alias declarations
- use with unconstrained arrays

Resolved signals
- resolved subtypes and signals
- composite/resolved types
- std_logic and std_logic_vector
- resolved ports
- type conversions in port maps

Generics
- generic lists and generic maps
- generics for delay parameters
- generics for port sizing

Lab: Parallel I/O controller for the eLucid-8 Microcontroller

Day 3

Components & configurations
- component declaration
- component instantiation
- component declarations in packages
- default binding
- configuration declarations
- hierarchical configuration
- direct instantiation of configurations
- remapping of generics and ports

Generate statements
- generating iterative structures
- conditionally generating structures
- configuration of generated statements

Lab: ALU/Shifter for the eLucid-8 Microcontroller

User defined attributes
- attribute declarations
- attribute specifications

Intro to synthesis
- design refinement
- IEEE synthesis standard
- use of data types
- enumeration encoding
- interpretation of metalogic values
- interpretation of timing
- combinational logic
- edge-triggered logic
- latch inference and avoidance
- finite state machines
- other coding guidelines

Lab: Register and Datapath Components for the eLudic-8 Microcontroller

Day 4

Files and input/output
- text files and the package textio
- reading from input
- writing to output

Test benches
- stimulus/response testing
- black box and white box testing
- test coverage
- refinement testing
- checkerboard testing

Conclusion
- overview of additional VHDL features
- further resources

Comprehensive Lab: The eLucid-8 Microcontroller—Putting it Together

For more information contact:

Tom Wille
tw@tm-associates.com
503-656-4457
TM Associates, Inc.
www.tm-associates.com