VHDL-2008

Course Description

This course introduces the new and changed features of VHDL-2008, in order to improve and enhance hardware engineer's skills.

The course goes into great depth and teaches the advanced features of the VHDL-2008 language through code examples, shows how they improve the language as a tool for design and verification, and guides how to employ them in new designs.

The course combines 50% theory with 50% practical work in every meeting. The practical labs cover all the theory and also include practical digital design.

This course also enriches digital engineers with many years of experience.

Course Duration

3 days

Goals

1. Write generic models at the highest level for synthesis
2. Use generic functions and procedures for synthesis include operator and function overloading
3. Write VHDL programs with VHDL2008 standard
4. Write advanced test-benches using VHPI and PSL
5. Apply advanced arithmetic/logic operations on arrays and scalars
6. Write advanced test-benches that read and write from/to files
7. Become familiar with the expand IEEE Packages
8. Advanced use of TYPE

Intended Users

Hardware engineers who develop FPGAs and would like to enhance their skills, in order to become experts with VHDL language for design and verification

Previous Knowledge

FPGA design, VHDL

Course Material

1. Simulator: Modelsim or ActiveHDL
2. Synthesizer and Place & Route: Quartus II (ALTERA) or Precision (Mentor Graphics)

Table of Contents

Day #1

        - Conditional and selected signal assignments
        - Forcing assignments
        - Matching case assignments
        - If and case generate
    - Modeling enhancements
        - Signal expressions in port maps
        - All signals in sensitivity list
        - Reading OUT mode ports and parameters
        - Slices in aggregates
        - Bit string literals
    - Improved I/O
        - TO_string functions
        - Justify function
        - Newline formatting
        - Read & Write operations
        - The TEE procedure
        - The Flush procedure

Day #2

- **Design Reuse & Parameterized Design**
    - Requirements for reusable design
        - Design document
        - Verification strategy
        - Synchronous design techniques
        - Coding style
        - Test benches
        - Parameterizable
        - Reuse pitfalls
    - Design Reuse Components
        - Generic
        - Constant
        - Alias
        - Functions and Procedures
        - Package
        - Packaging IP for reuse
        - Array attributes

        - Unconstrained arrays
        - Records
        - Array of records
        - Array attributes
    - Generate statements for multiple architecture implementation
        - Component generate
        - For generate
        - If generate
        - Signal declaration in generate
        - Process declaration in generate
        - Controlling generate behavior from a package
        - Pipeline generate
    - Configuration and recursive configuration
    - Shared variables and protected types

- **Advanced Subprograms Implementation**
    - Review of functions and procedures
        - Rules
        - Variables behavior
        - Side effects
        - Common and impure functions
        - Synthesis implementation of subprograms
        - Actual and formal parameters
        - Resolution function
        - Conversion function
        - Function versus Procedure design consideration
        - Subprograms in package versus in architecture design consideration
    - Coding style for fully generic subprograms
        - Unconstrained arrays
        - Constant arrays
        - Attributes
        - Array of arrays
        - Array slices
        - General loops
        - Signals & Variables length match in logical and arithmetic operations
        - Qualified expressions
        - Using signed & unsigned

        - log2 implementation
    - Subprogram overloading
        - Function overloading
        - Procedure overloading
        - Operator overloading
        - Overloading in a package
        - Using dot operator with same declaration in different packages

Day #3

- **Hands-On: Design & Implement a Small Project with VHDL 2008**
    - Project introduction & Theory
    - Project design stage
        - Using VHDL 2008 enhancements
        - Using functions & procedures
        - Design for reusability
    - Project verification stage
        - Writing test vectors
        - Reading and writing files with text I/O and std_logic_textio packages
        - Using PSL