VHDL for Designers

Course Description
VHDL training by Doulos is the industry standard training courses teaching the application of VHDL for FPGA and ASIC design. It is fully updated and restructured to reflect current best practice.
This training prepares the engineer for practical project readiness for FPGA designs. While the emphasis is on the practical VHDL-to-hardware flow for FPGA devices, this module also provides the essential foundation needed by ASIC and FPGA designers wishing to apply the more advanced features of VHDL covered in the next module. Delegates targeting FPGAs will take away a flexible project infra-structure which includes a set of scripts, example designs, modules and constraint files to use, adapt and extend on their own projects.
Because Doulos is independent, delegates can usually use their choice of design tools during the workshops. Workshops are based around carefully designed exercises to reinforce and challenge the extent of learning, and comprise approximately 50% of class time.

Training Duration: 4 days

Who should attend?
- Engineers who wish to become skilled in the practical use of VHDL for FPGA or ASIC design
- Engineers who are about to embark on the first VHDL design project
- Engineers who have already acquired some practical experience in the use of VHDL, but wish to consolidate and extend their knowledge within a training environment

What will you learn?
- The VHDL language concepts and constructs essential for FPGA design
- How to write VHDL for effective RTL synthesis
- How to target VHDL code to an FPGA device architecture
- How to write simple VHDL test benches
- The tool flow from VHDL through simulation, synthesis and place-and-route
- How to write high quality VHDL code that reflects best practice in the industry

Prerequisites:
Delegates must have attended Essential Digital Design Techniques or an equivalent course, or have a good working knowledge of digital hardware design. No previous knowledge of VHDL or a software language is required.

Course materials
Course materials are renowned for being the most comprehensive and user friendly available. Their style, content and coverage are unique in the HDL training world and have made them sought after resources in their own right. Course fees include:
- Fully indexed course notes creating a complete reference manual
- Workbook full of practical examples to help you apply your knowledge
- Doulos Golden Reference Guide for VHDL language, syntax, semantics and tips
- Tool tour guides (to support the tools and technologies of your choice)
- PaceMaker Multimedia CD-ROM Tutorial for optional pre-course preparation
- Design flow guide for ASIC and the leading FPGA/CPLD technologies

Course Outline

1. Introduction
- The scope and application of VHDL
- Design and tool flow
- FPGAs
- The VHDL world

2. Getting Started
- The basic VHDL language constructs
- VHDL source files and libraries
- The compilation procedure
- Synchronous design and timing constraints

3. FPGA Design Flow (Practical exercises using a hardware board)
- Simulation
- Synthesis
- Place-and-Route
- Device programming

4. Design Entities
- Entities and Architectures
- Std_logic
- Signals and Ports
- Concurrent assignments
- Instantiation and Port Maps
- The Context Clause

5. Processes
- The Process statement
- Sensitivity list versus Wait
- Signal assignments and delta delays
- Register transfers
- Default assignment
- Simple Testbenches

Continued...
Continued...

6. Synthesising Combinational Logic
   - If statements
   - Conditional signal assignments and Equivalent process
   - Transparent latches
   - Case statements
   - Synthesis of combinational logic

7. Types
   - VHDL types
   - Standard packages
   - Integer subtypes
   - Std_logic and std_logic_vector
   - Slices and concatenation
   - Integer and vector values

8. Synthesis of Arithmetic
   - Arithmetic operator overloading
   - Arithmetic packages
   - Mixing integers and vectors
   - Resizing vectors
   - Resource sharing

9. Synthesising Sequential Logic
   - RISING_EDGE
   - Asynchronous set or reset
   - Synchronous inputs and clock enables
   - Synthesisable process templates
   - Implying registers

10. FSM Synthesis
    - Enumeration types
    - VHDL coding styles for FSMs
    - State encoding
    - Unreachable states and input hazards

11. Memories
    - Array types
    - Modelling memories
    - IP Generators
    - Instantiating generated components
    - Implementing ROMs

12. Basic TEXTIO
    - TEXTIO
    - READ and WRITE
    - Using TEXTIO for testbench stimulus and outputs
    - STD_LOGIC_TEXTIO

For registration and details
Arie Geler
Tel: 03- 924 7780 ext. 207
Fax: (972 3) 9247783
E-mail: arie@logtel.com