

Comprehensive VHDL

Standard Level - 5 days

Comprehensive VHDL is the industry standard 5-day training course teaching the application of VHDL for FPGA and ASIC design. Fully updated and restructured to reflect current best practice, engineers can attend either the individual modules, or the full 5-day course.

- **VHDL for FPGA Design** (days 1-3) 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.
- **Advanced VHDL** (days 4-5) builds on the foundation of the previous module to prepare the engineer for complex FPGA or ASIC design. It focuses on the use of VHDL for large hierarchical designs, design re-use, and the creation of more powerful test benches.

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.

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?

VHDL for FPGA Design

- 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

Advanced VHDL

- The VHDL language concepts constructs essential for complex FPGA and ASIC design
- The VHDL language constructs and coding styles that enable sophisticated test benches
- How to code hierarchical designs using multiple VHDL design libraries
- How to write re-usable, parameterisable VHDL code by exploiting generics and data types
- How to run gate-level simulations

For further information contact your local Doulos [Sales Office](#).



Comprehensive VHDL

Standard Level - 5 days

Pre-requisites

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.

Delegates attending only the **Advanced VHDL** module must have some hardware design experience, and have completed the **VHDL for FPGA Design** module or an equivalent course. We have found that delegates frequently overestimate their own capabilities. If in doubt, you will probably benefit from attending the full **Comprehensive VHDL** course.

Course materials

Doulos course materials are renowned for being the most comprehensive and user friendly available. Their style, content and coverage is unique in the HDL training world and had 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 and solutions 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 Tutorial & Reference— multi-media CD-ROM for optional pre-course preparation
- Design flow guide for ASIC and the leading FPGA/CPLD technologies

Structure and Content

VHDL for FPGA Design (days 1-3)

Introduction

The scope and application of VHDL • Design and tool flow • FPGAs • The VHDL world

Getting Started

The basic VHDL language constructs • VHDL source files and libraries • The compilation procedure • Synchronous design and timing constraints

FPGA Design Flow (Practical exercises using a hardware board)

Simulation • Synthesis • Place-and-Route • Device programming

Design Entities

Entities and Architectures • Std_logic • Signals and Ports • Concurrent assignments • Instantiation and Port Maps • The Context Clause

Processes

The Process statement • Sensitivity list versus Wait • Signal assignments and delta delays • Register transfers • Default assignment • Simple Testbenches

Synthesising Combinational Logic

If statements • Conditional signal assignments and Equivalent process • Transparent latches • Case statements • Synthesis of combinational logic

Comprehensive VHDL

Standard Level - 5 days

Types

VHDL types • Standard packages • Integer subtypes • Std_logic and std_logic_vector • Slices and concatenation • Integer and vector values

Synthesis of Arithmetic

Arithmetic operator overloading • Arithmetic packages • Mixing integers and vectors • Resizing vectors • Resource sharing

Synthesising Sequential Logic

RISING_EDGE • Asynchronous set or reset • Synchronous inputs and clock enables • Synthesisable process templates • Implying registers

FSM Synthesis

Enumeration types • VHDL coding styles for FSMs • State encoding • Unreachable states and input hazards

Memories

Array types • Modelling memories • IP Generators • Instantiating generated components • Implementing ROMs

Basic TEXTIO

TEXTIO • READ and WRITE • Using TEXTIO for testbench stimulus and outputs • STD_LOGIC_TEXTIO

Advanced VHDL (days 4-5)

More About Types

Variables • Loops • Std_logic and resolution • Array and integer subtypes • Aggregates

Managing Hierarchical Designs

Hierarchical design flow • Library name mapping • Component declaration • Configuration • Hierarchical configurations • Compilation order

Parameterised Design Entities

Array and type attributes • Port Maps • Generics and Generic Maps • Generate statement • Generics and generate

Procedural Testbenches

Subprograms • Procedures • Functions • Parameters and Parameter Association • Package declarations • Package bodies • Subprograms in packages • Subprogram overloading • Operator overloading • Qualified expressions • RTL Procedures

Text-File-Based Testbenches

Assertions • Opening and closing files • Catching TEXTIO errors • Converting between VHDL types and strings • Checking simulation results • Initialising memories • Foreign bodies

Continued...

Comprehensive VHDL

Standard Level - 5 days

Gate Level Simulation

Rationale for gate level simulation • VITAL tool flow • Reuse of RTL testbench at gate level • Comparison of RTL and gate level results • Behavioural modelling

Pre-cursor courses

- Essential Digital Design Techniques (2 days)
(VHDL Designer Induction – fast-track pricing package: book Essential Digital Design Techniques at the same time as the full 5-day Comprehensive VHDL course and save money)

Related courses

- Expert VHDL (incorporating Design and Verification modules)
- Essential Perl
- Altera *Professional Designer* courses
- Xilinx *Professional Designer* courses
- Essential Tcl/Tk

Project services

Doulos Project Services enable clients to access our world-leading technical know-how and apply it directly to projects. **Expert-on-call**, **Expert-design** and **Expert-support** options can be flexibly packaged and delivered to provide valuable expertise and additional resource just when it is needed.

How to book a course

To make a provisional booking, or to obtain pricing information, please contact your local Doulos sales team. You will find contact details on our [website](#).

For further information contact your local Doulos [Sales Office](#).

