

Expert VHDL

Advanced Level - 5 days

Expert VHDL is an intensive 5-day advanced application class. It teaches engineers how to increase productivity by enhancing their VHDL coding and application skills. Presented in two distinct class modules, Expert VHDL focuses on language and synthesis issues, design maintainability and re-use, test benches and the latest techniques for verification - including an introduction to OVL and modern assertion-based design techniques.

- **Expert VHDL Design (2 days)** is for design engineers wishing to deepen their knowledge of RTL synthesis using VHDL, and to improve their VHDL coding style with maintainability and design re-use in mind. Design for Verification is also covered with an introduction to modern assertion-based techniques
- **Expert VHDL Verification (3 days)** is for design engineers and verification engineers involved in VHDL test bench development or behavioural modelling for the purpose of functional verification

The modules, which may be attended together or independently, follow on from the industry standard class, **Comprehensive VHDL**. Carefully designed workshops comprise approximately 50% of teaching time, and enable engineers to apply their new skills in the context of the latest VHDL design tools, practices and methodologies.

Who should attend?

- Design engineers wishing to improve the efficiency of their hardware designs and increase productivity.
- Design and verification engineers who want to structure and write effective test environments to verify complex designs and systems.

What will you learn?

- A set of VHDL language features that go beyond what is taught on a basic training class
- A deeper understanding of the VHDL language and how to apply it, enabling you to troubleshoot VHDL simulation and synthesis problems with ease
- The principles and details of how to approach the problem of design verification using VHDL
- How to structure and write large and complex VHDL test benches, including implementing functional coverage
- The principles and details of how to write behavioural models of hardware components in VHDL
- How to produce smaller and faster hardware design using VHDL and RTL synthesis tools
- Design techniques for state machines and arithmetic using VHDL packages
- The details of a VHDL coding style to facilitate code re-use and how to 'package' IP for re-use
- An introduction to IEEE 1076-2007c (VHPI) and the proposed VHDL 2008

Pre-requisites

This is an advanced language and methodology training class. Prior attendance of the Doulos **Comprehensive VHDL** class (or equivalent) is required, and at least 6 months of 'live' project experience using VHDL is strongly recommended. Delegates attending the Expert Design module must have knowledge and experience of register transfer level coding and synthesis, using VHDL.

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



Expert VHDL

Advanced Level - 5 days

Class materials

Doulos class materials are renowned as the most comprehensive and user friendly available. Their style, content and coverage is unique in the HDL training world and has made them sought after resources in their own right. Class fees include:

- Fully indexed class notes creating a complete reference manual
- Workbook full of practical examples to help you apply your knowledge
- Doulos VHDL Golden Reference Guide for language, syntax, semantics and tips
- Tour guides (to support the tools and technologies of your choice)

Structure and content

Expert VHDL Design (days 1-2)

RTL

Synthesising combinational and sequential logic • Using variables in clocked processes • Multiple drivers and tri-states • RTL functions and procedures • Kinds of decision-making logic • Using hierarchy to control synthesis • Timing constraints, area constraints, and optimisation options • Using generate and attributes for mapping onto FPGAs • Optimal one-hot decoding • Input hazards and metastability • Multiple clock edges • Synchronisation between clock domains • Synchronising reset signals • Synthesis methodology for large designs

IP and Re-use with VHDL

Using standard packages • Language level re-use • Standard component re-use • General re-use • Economic payback from re-use • Packaging IP for re-use • Impact of IP on the development cycle Writing re-usable RTL VHDL • Readability and maintainability • Seeing generaliseable properties Array attributes, cloning ranges • Arrays of arrays, unconstrained arrays, others • Creating regular structures using loops and generate • Using generics to parameterise widths and structures

VHDL Coding Styles for IP Block Design

Records • Using records and aliases for abstraction • Matrices • Converting between matrices and arrays • Representing register banks • Implementing destructive reads • State machine coding styles Synthesis and hardware encoding of state machines • Recursive instantiation and recursive functions

Design for Verification with Assertions

Reasons for designing with assertions • Properties and assertions • Examples in OVL

Expert VHDL Verification (days 3-5)

VHDL Language

Subprograms, parameters, assigning signals • User defined packages • User defined array types • Record types, selected names, aggregates, arrays of records • Types, subtypes and overloading, conversion functions • Qualified expressions • Generics, string generics, array generics Configurations, binding and dependencies, generic and port maps

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



Expert VHDL

Advanced Level - 5 days

Verification Environments and Methodology

The Verification Plan • Structure of a simple test bench • Structure of a complex test bench • Procedural stimulus generation • Reactive test benches • File I/O; TEXTIO and 'C' • Measuring delays

- Monitoring internal signals • Generating random numbers • Collecting diagnostic data • Scoreboards • Coping with latency and Out-of-Order completion • Control files • Adding a user interface to a test bench • Writing behavioural models • Generic and parameterised test benches • How to implement functional coverage • How to implement run-time parameterisation • A re-usable generic approach to creating verification environments • Example code to take away

How VHDL works

Signal assignments • Events and inertial delay • Deltas Drivers and resolution functions • Wait statements • NOW • Static elaboration, the network model • Dynamic elaboration, elaborating arrays and files in subprograms • VHDL Attributes

Component Modelling

How to structure a behavioural model • Representing state • Example - behavioural modelling of a serial thermometer chip • Giving visibility of internal state • Modelling external timing relationships • Checking timing constraints using signal attributes • 1164 strength strippers • Handling 'X' on the inputs • Modelling memories • Modelling analogue blocks • Bus-functional models • Processor models

- Foreign bodies for including C models for interfacing to emulators

Pre-cursor classes

- Comprehensive VHDL

Related classes

- Assertion Based Verification with PSL
- Altera TechClass
- Xilinx Training
- Essential Perl
- 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 class

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](#).

