

VHDL - Advanced design methodology

Ref : 003343A

Duration : 3 days

OBJECTIVES

- Acquire a strong design methodology
- Get the best out of VHDL language when synthesis and simulate
- A large number of practical labs with demonstrations let you verify the principles of this methodology using an appropriate VHDL write style.
- This methodology give you efficient implementation, reusable code and give you an easy way to debug.

RELATED COURSES

- VHDL - language initiation (003342A)
- VHDL Logical Synthesis and Simulation for Xilinx FPGA design (002572A)

PREREQUISITES

- Intermediate knowledge in digital electronic design.
- Basic to intermediate knowledge in VHDL language.

TRAINING MATERIALS

Software Configuration :

- Xilinx ISE
- Simulation tool : ISIM or Modelsim

Recommended Hardware Configuration :

- Intel Core 2 or equivalent
- Windows XP
- 1 GB Free disk after software installation
- At least 1Go RAM
- Minimum Display resolution : 1024 x 768
- On Site training : video projector

Contact

Tel : 05 62 13 52 32
Fax : 05 61 06 72 60
training@mvd-fpga.com

Course also available
customized

Next sessions, see : <http://www.mvd-training.com/en/schedule.html>

TOPICS

Important reminders on rules and writing recommendations of VHDL code in logical synthesis

- Predefined operators with extended use by using standardized packages
- The process
 - Importance of sensitivity
 - Use of variables
- A few classic tricks to avoid
- Potential interpretation incoherencies between logical synthesis and simulation : how to avoid them ?
- Practical labs

Hardware designing methodology in logical synthesis

- Asynchronous conception and classic tricks
- Metastability and hazards of functioning
- Limits of functional simulation and timing on asynchronous designs : how to avoid them ?
- Synchronous design –advantages-methodology-tuning
- Static analyze of the timing : how to use it ?
- Optimization of performances irrespective of the target
- Pipeline notion
- Asynchronous event management
 - Random
 - Data flows
- Practical labs

Deepening of VHDL language for the optimization and the re-use of the code in logical synthesis

- Notion of variable and example of use
- Genericity and automatic re-configuration or re-usable modules
- Useful predefined allocations in logical synthesis
- Functions and procedures

- Definition of packages and libraries
- Practical labs

Hierarchy management for a better use

- Design organization by functional modules : what routing to choose
- Inference and instancing notions
 - When is it important to instantiate primitives or macros ?
 - Precautions for an evolutionary and or re-usable code
- Importance of modules' name selection and of the nets to facilitates the physical implementation, the simulation and the focusing
- Does the hierarchy have to be preserved during the logical synthesis ?
- Practical labs

Testbenches and simulation

- A few basic rules for the writing of an efficient testbench
- VHDL instructions specific to simulation
 - Wait and its various forms
 - « Loop »
 - Assertions
 - Data types
 - Others
- Writing components models intended to make the simulation more realistic
- Use of existing models and packages of simulation
- Practical labs
- Integration of « pseudo logic » in order to facilitate the interpretation of the simulation results
- Writing and reading of ASCII files
 - Allocation of a data flow from a file
 - Storage of the simulation results in a file
- Command interpreter
- Generating information message
 - Practical labs

DOCUMENTATION

Training manuals will be given to attendees during training in print.