oV2 - Advanced VHDL for FPGA

Objectives

- Comprehend the various possibilities offered by VHDL language
- Be able to read and test VHDL components
- Understand the logical synthesis notions
- Understand the crucial issue of implementing Finite State Machines (FSMs) in hardware
- Organizing the code developing package and libraries
- Reusing components
- Learning how to write efficient testbenches for simulation
- Knowing the different writing style and their impact on the quality of synthesis results
- Checking Timings
- How to structure and write large and complex VHDL structured verification environments
- The OSVVM and UVVM VHDL verification methodologies

Prerequisites

- Basic knowledge of VHDL, oV1 - VHDL Language basicscourse level

Course Environment

- Theoretical course
  - PDF course material (in English).
  - Course dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - One Online Linux PC per trainee for the practical activities.
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
  - Eclipse environment and GCC compiler.
  - QEMU Emulated board or physical board connected to the online PC (depending on the course).
  - Some Labs may be completed between sessions and are checked by the trainer on the next session.
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Duration

- Total: 18 hours
- 3 sessions, 6 hours each (excluding break time)
- From 40% to 50% of training time is devoted to practical activities
- Some Labs may be completed between sessions and are checked by the trainer on the next session

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

First session

Finite State Machine (1st part)

- The Finite State Machine Approach
  - Sequential Circuits and State Machines
  - State Transition Diagram
  - Transition Types
  - Moore-to-Mealy Conversion
  - Mealy-to-Moore Conversion
  - Exercises
- Hardware Fundamentals
  - Flip-Flops
  - Metastability and Synchronizers
  - Pulse Detection
  - Glitches
  - Pipelined Implementations
  - Exercises
  - Hardware Architectures for State Machines
  - Fundamental Design Technique for Moore Machines
  - Fundamental Design Technique for Mealy Machines
  - Moore versus Mealy Time Behavior
  - State Machine Categories and State-Encoding Options
  - Safe State Machines

Finite State Machine (2nd part)

- Design Steps and Classical Mistakes
  - Classical Problems and Mistakes
  - Design Steps Summary
- Regular State Machines
  - Architectures for Regular Machines
  - Number of Flip-Flops
  - Exercises
- Timed State Machines
  - Architectures for Timed Machines
  - Timer interpretation
  - Transition Types and Timer Usage
  - Timer Control Strategies
  - Time Behavior of Timed Moore and Mealy Machines
  - Examples of Timed Machines

Exercise: *Designing a burstable RAM controller*

Second session

Design Methodology for Synthesis

- Designing for Synthesis
- Metastability
- Memory Synthesis
- Reset Generation
- Crossing Clock domains

Exercise: *Metastability*

Timing analysis and constraints

- Timing Closure challenges
- A methodology for successful Timing Closure
- Common Timing Closure Issues
- Static Timing Analysis
- Role of Timing Constraints in STA
- Common Issues in STA
- Delay Calculation versus STA
- Timing Path
- Setup and Hold
- Slack
- On-Chip Variation
- Clock
- Port Delays
- Completing Port Constraints
- False Paths
- Multi Cycle Paths
- Combinational Paths
- Xilinx Extensions

Exercise: *Design closure*
Exercise: *Analyzing and Resolving timing violations*

Third Session

Introduction to Open Source VHDL Verification Methodology (OSVVM)

- Overview
- Transaction-Level Modeling
- Constrained Random Test Generation
- Functional Coverage
- Intelligent Coverage Randomization Methodology
- Utilities for Testbench Process Synchronization
- Transcript Files
- Error Logging and Reporting – Alerts and Affirmations

Introduction to Universal VHDL Verification Methodology (UVVM)

- Utility Library
- VVC (VHDL Verification Component) Framework
- BFMs (Bus Functional Models
- OSVVM and UVVM

Vivado Debug

- Vivado Integrated Logic Analyzer (ILA)

- Adding debug nets
- Analyzing debug data
- Resources

Renseignements pratiques

Duration : 18 hours
Cost : 2510 € HT