



## H2 - Lattice Diamond

*Mastering Diamond for FPGA optimisation and debug*

### Objectives

- Learn the use of the Lattice Diamond toolset
- Learn how to optimize a design to meet
  - A mandatory working frequency
  - Constraints of clock synchronization
- Learn how to debug a design by
  - Simulation
  - Insertion of the Reveal trace kernel

All exercises are conducted on Lattice FPGA-based boards.

### Hardware

- A Windows PC for two trainees with the Lattice Diamond toolset
- A Lattice target board
- Printed course material with ample space to take notes
- Hands-on labs manual and detailed solutions

### Prerequisites

- Good knowledge of VHDL or Verilog languages and of the structure of FPGAs ([V1 - VHDL Language Basics](#) course)

### Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

### Target Audience

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

## Course Outline

### First day

#### Basic Diamond design flow

- Diamond projects
- Tools and environment options
- Assigning pin locations and timing preferences
- Map the design
- Perform post-map static timing analysis
- Place and Route the design

- select PAR options
- Perform post-route static timing analysis
  - View device implementation
  - identify critical paths

## Enhance timing characteristics of a design

- Perform static, post-PAR, timing analysis
  - Max frequency analysis
  - Setup and Hold time analysis
  - Clock to output delay analysis
- Optimize Mapping, Placement and Routing
  - avoid using IO blocks in critical paths
  - use VHDL signal attributes
- Use a PLL to minimize clock delays
  - The IPexpress tool
  - Using IPexpress to add a PLL to a design
  - possible impact on clock-to-out delays
- PAR optimisation strategies
  - Use PAR effort settings to enhance timings
  - Use Multi-placement mode
  - Use routing-only (reentrant) PAR
- Use PLL parameter to further compensate for clock delays
- Use guided placement
  - create floorplanning directives in the HDL source
  - create floorplanning directives using the design planner
  - create floorplanning directives in the preferences file
- Look at device implementation after optimization
  - look at routing congestion

## Second day

### Power consumption estimation

- Using the Power Calculator
  - Estimating the Activity Factor
  - Estimate temperature impact
  - Estimate the device selection impact

### Design simulation

- Creating the test-bench with Lattice Diamond
  - using the automatically generated test-bench template
  - completing the testbench
- Simulation with Aldec Active-HDL
  - functional simulation
  - post PAR dynamic timing simulation

### Debugging the device

- The Reveal on-chip debugger
- The Reveal Inserter
  - Add the Reveal core to your design
  - Add signals to trace
  - Add triggers and trigger expressions
- The Reveal Analyzer
  - Creating a Reveal Analyzer project

- Attach to the device
- Run the device and fetch traces
  - Display signal waveforms
  - single trigger capture mode