FPGA Design
Part II - Xilinx Design Tools

Thomas Lenzi
Objective

- In this lecture we will learn to design for FPGAs using the Xilinx Development Tools.
Development Tools
Xilinx Development Tools

- ISE Design Suite: for Spartan3, Spartan6, and Virtex6 FPGAs
- Vivado Design Suite: for new generations of FPGAs
- Impact: programming of the FPGA
- ISim: simulate VHDL and Verilog code
- ChipScope: in-system debug and analysis. Allows you to monitor and control signals inside the FPGA
- Core Generator: gives you access to pre-build components
- PlanAhead: floorplanning, pin assignment, ...
- Xilinx Software Development: develop C/C++ code for embedded systems
- etc
Design workflow

1. ISE: write the VHDL and Verily code
2. *ISim: perform software simulations
3. *PlanAhead: allocate resources on the FPGA
4. Impact: program the FPGA
5. *ChipScope: in-system debug of the code
ISE Design Suite
Workflow

- **Your code (VHDL)**
- **Synthesis**
- **Translate**
- **Constrains (UCF)**
- **Map**
- **Place & Route**
- **Generate**
- ***.bit file**

Transforms your VHDL code into hardware components

Include the user constraints in the synthesised code

Map the translated code to the device resources

Place the design in the FPGA and form the routes

Generate the BIT file that can be loaded on the FPGA
Structure of the project

- Every design includes two things:
  - VHDL files that describe the behaviour of the code;
  - UCF (user constraints) files that constrain the design (IO pins for signals, timing requirements, ...).
The structure of a VHDL code can be compared to the structure of an FPGA.

The Top-Level file describes the FPGA itself. It regroups the signals that will leave or enter. That module is connected to the world via the IO pins.

In that module, you can create sub-modules to use regions of the FPGA.

There is no limitation to the number of sub-modules you can create.
Create a project

- Go to “File > New Project…”
- Give it a name and a location, set the “Top-Level source type” to “HDL”, and click “Next”
- Select the attributes corresponding to your board and click “Next”
  - Family: Spartan3E
  - Device: XC3S100E
  - Package: CP132
  - Speed: -4
  - Preferred language: VHDL
- Finally click “Finish”
Create a VHDL file

- Go to “Project > New Source…”
- Select “VHDL Module” and give it a name with a .vhd extension, then click “Next”
- On the next window, leave everything empty and click “Next” and then “Finish”
Create a UCF file

- Go to “Project > New Source…”
- Select “Implementation Constraints File” and give it a name with a .ucf extension, then click “Next”
- On the next window, click “Finish”
ISE interface
Design management

- This region allows you to generate your design and flash it on the FPGA.

- Various tools are also present to analyse and optimise your design.

- To compile a design, simply double-click on “Generate Programming File”. All the necessary steps will be done automatically.
Simulation environment

- ISE makes a difference between Implementation (code that will run on the FPGA) and Simulation (code that will be run on the software side only).

- Simulation often re-uses implementation code but provides it with signals that are generated in the simulation code.

- For example, instead of being connected to a button on the PCB, the signal would be manually defined (first high, then low after x ns, …) by a VHDL simulation code.
ISim
PlanAhead
Pin placement

- PlanAhead can be used for pin placement in the design. It gives a visual representation of the FPGA and allows you to point and click to place signals.
PlanAhead interface

FPGA package

IO pins and standards
Impact
JTAG chain

Set up

Program

FPGA Flash

Identify Succeeded