Aalto University
School of Electrical
Engineering

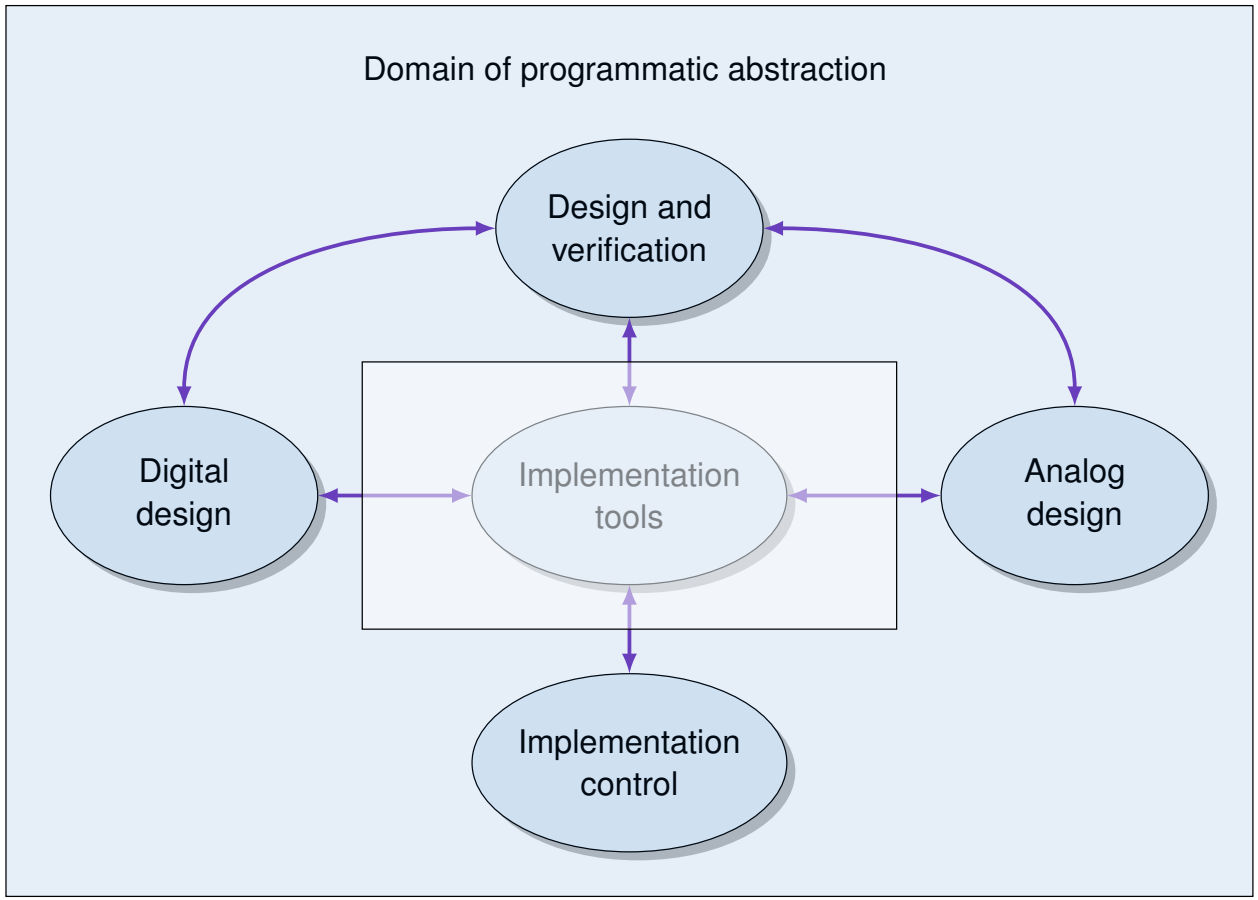
Postgraduate course on electronic circuit design
Memory mapped microcontroller configuration on FPGA for controlling custom things.

Marko Kosunen

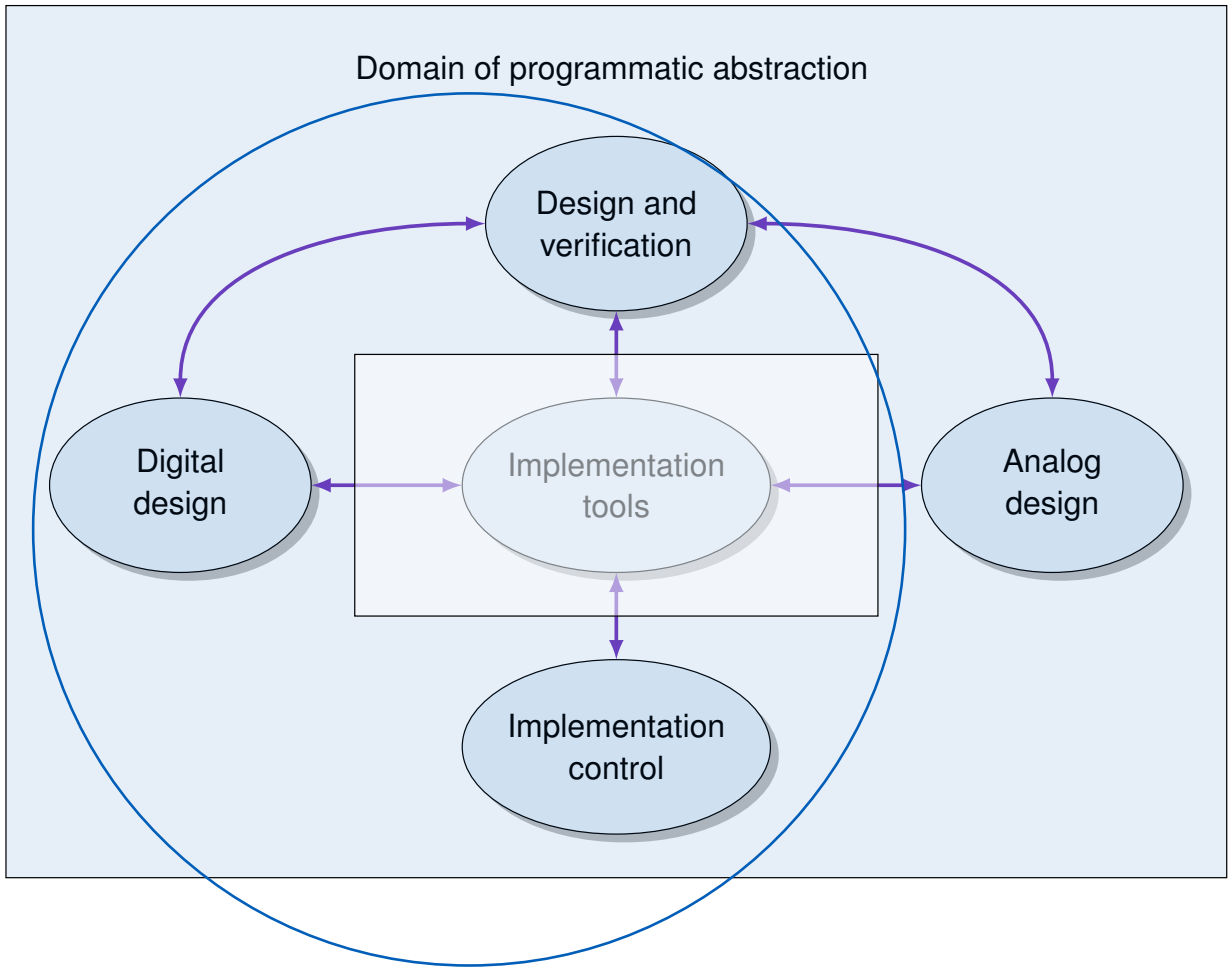
Department of Electronics and Nanoengineering

October 20, 2020

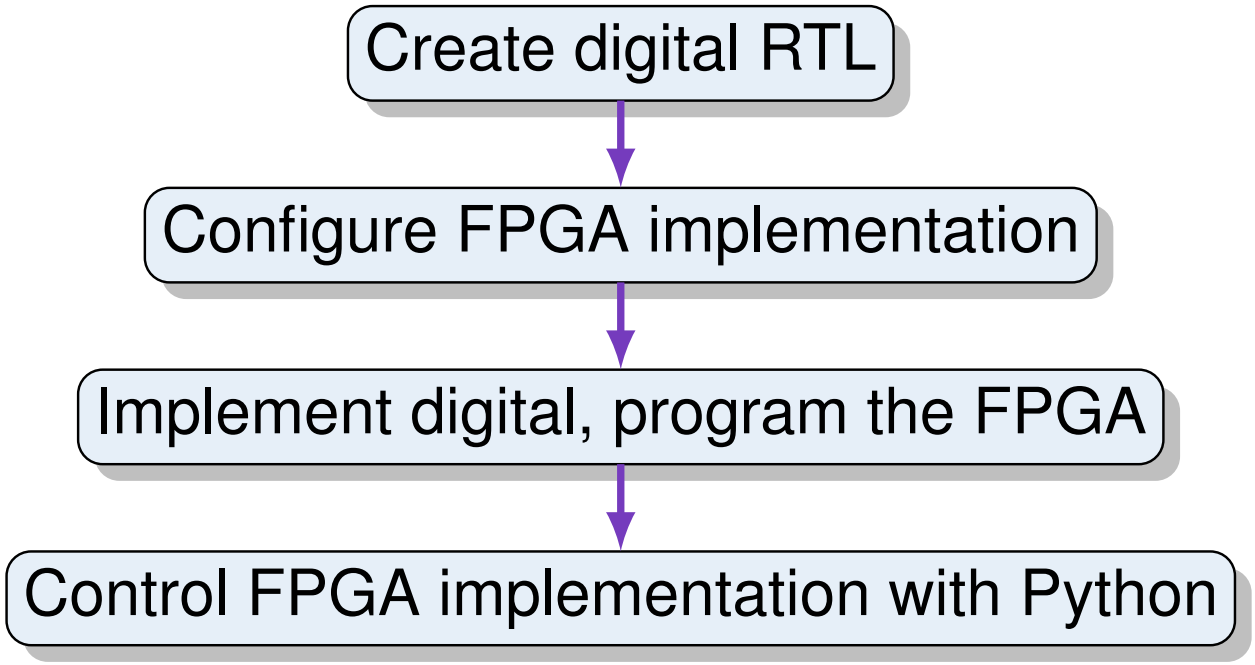
Programmatic circuit design



Programmatic circuit design

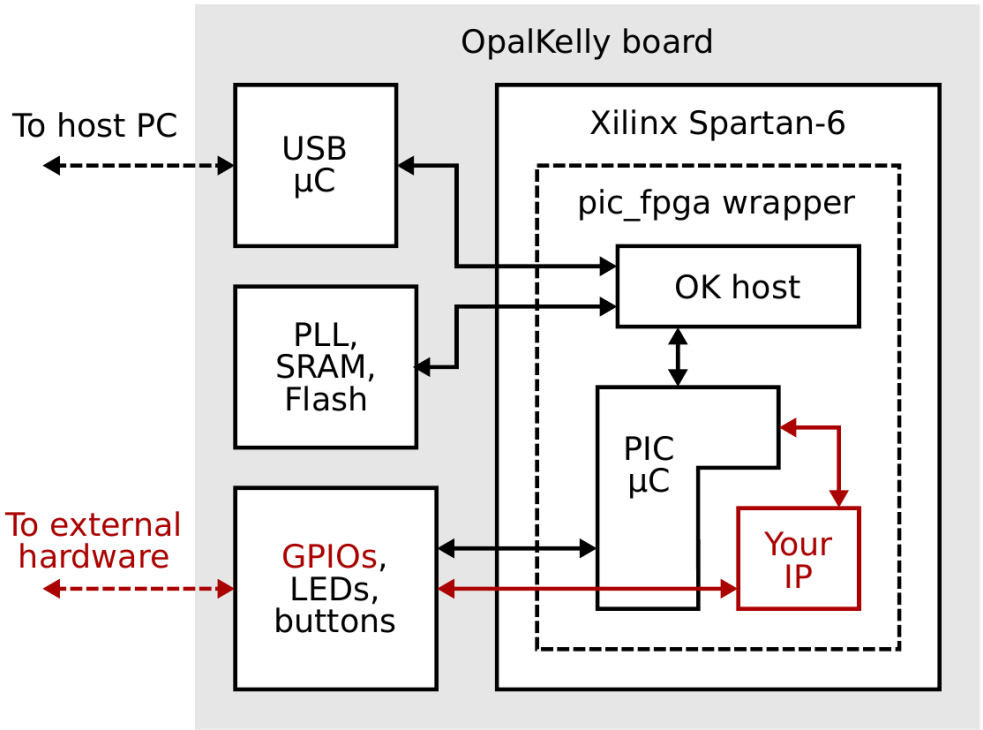


Product of year 2019: seamless FPGA implementation toolchain



Course objective:
Memory mapped microcontroller configuration

- Host runs python
- FPGA contains PIC16F34 microcontroller
- PIC runs assembly
- PIC *may* communicate with host during execution



Course objective

- Course is done in groups of 2
- You are given VHDL for PIC16F84A microcontroller.
- You are given example program in assembly and example how to simulate it on with Modelsim.
- You are given a toolchain to push it and run it on FPGA.
- Task:
  - Implement and control something with microcontroller on FPGA, (E.g. debug SPI slave controller with FPGA).
  - Control something with the FPGA dev board containing microcontroller and program. E.g. Control your chip with SPI-master implemented as PIC on FPGA.
  - Your imagination (and FPGA capacity) is the limit.
- Build process should use Configure && make, GUI for debugging and studies.
- Extend your capabilities by designing an *Exercise* or *Demo* that can be executed based on given instructions.
- Presentation or demo session, $\approx$ 15 min.

Programmatic project management

- We use Git for everything on this course in order to learn how to use it

Programmatic project management

- We use Git for everything on this course in order to learn how to use it
- We use Git issues and Slack for (shared) communication

Implementation platform

- OpalKelly XEM6001 (2 pcs.)
  - Xilinx Spartan-6 (XC6SLX16 FTG256)
- OpalKelly XEM6010 (5 pcs.)
  - Xilinx Spartan-6 (XC6SLX45 FGG484)
- ModelSim (Student edition available at https://www.mentor.com/company/higher_ed/modelsim-student-edition)
- Python 3.6 or newer.

Phase 1-Implementation platform and Toolchain

- OpalKelly FPGA. API's and FrontPanel available at Aalto Version https://version.aalto.fi
- Log in there, and I will add you to the Course Group
- Xilinx ISE required for programming the FPGA. Available:
  Windows 10:
  https://www.xilinx.com/support/download/index.html/content/xilinx/en/dow
  tools/14_7-windows.html
  Other:
  https://www.xilinx.com/support/download/index.html/content/xilinx/en/dow
  tools/v2012_4—14_7.html
- License is "ISE WebPack" it's downloadable during/after the install procedure.

Phase 1-Test the implementation platform and toolchain

- See the Gitlab issue board at https://version.aalto.fi/gitlab/elec-l3510_exec/main/-/boards
- Examples and documentation available:
  - Simulation: https://version.aalto.fi/gitlab/elec-l3510_exec/pic16f84a-vhdl/-/blob/master/README.md
  - FPGA implementation: https://version.aalto.fi/gitlab/elec-l3510_exec/pic-fpga/-/blob/master/README.md
- Fork these projects to you Groups Gitlab group. Instruction provided https://version.aalto.fi/gitlab/elec-l3510_exec/pic16f84a-vhdl
- Study how to program and simulate program the assembly code with VHDL with examples.
- Study how to implement the microcontroller on FPGA by getting acquainted with the code in pic-fpga
- Obtain FPGA board from course personnell and test the flow.

Phase 1-Test the implementation platform and toolchain

- First phase goal is to establish a script-based programming environment where you can push the design and program on FPGA without opening the gui.
- We have multiple groups, and collaboration is OK, but leeching is not. Each groups develops a toolchain of their own, but after X weeks you should (perhaps) converge to use the best one of them.

Phase-2 Implement a memory mapped microcontroller configuration for control of custom blocks.

- Use SPI as a test case. https://version.aalto.fi/gitlab/elec-l3510_exec/spi_slave
- We give you SPI slave VHDL to program on FPGA, but you need to write a Python interface and assembly program to control it (read and Write).

Phase 3-Excercise or demo

- You may skip Phase-II if you can.
- Develop any kind of a memory mapped microcontroller configuration to control anything you want/need in your work or just out of interest.
- Prepare a demo or course exercise out of your work.
- Best works are used in coming Bachelor's course "Programmatic circuit design". Eternal glory will be yours.

Phase 4-Presentations

- Each group gives a 15 minute slide presentation of their demos/exercises

Hints-Modularity is the key

- The most important thing in terms of re-usability:
  - Example templates are a git module that can be used for project initialization.
  - Every design should be a Git module
  - Every sub-design should be utilized as a git sub-module
  - Using sub-modules requires some advanced Git skills, thus we practice.
- On git related problems: *every one of them has been solved by someone*. Do a web search git <myproblem>.