Teaching Chip Design with Open-Source Tools

Syed Anas Alam[1], Jakob Furbo Enevoldsen[1], Andreas Alkjær Eriksen[1], Niels William Hartmann[1], Ulrik Helk[1], Jørgen Kragh Jakobsen[2], Christa Skytte Jensen[1], Nicolai Dyre Bülow Jespersen[1], Karl Herman Krause[1], Mads Rumle Nordstrøm[1], Tjark Petersen[1], Luca Pezzarossa[1], Simon Winther Rasmussen[1], **Martin Schoeberl**[1], Jonas Ingerslev Sørensen[1]

[1]Department of Applied Mathematics and Computer Science
Technical University of Denmark
Kgs. Lyngby, Denmark

[2]IC Works - Open Source Chip Design

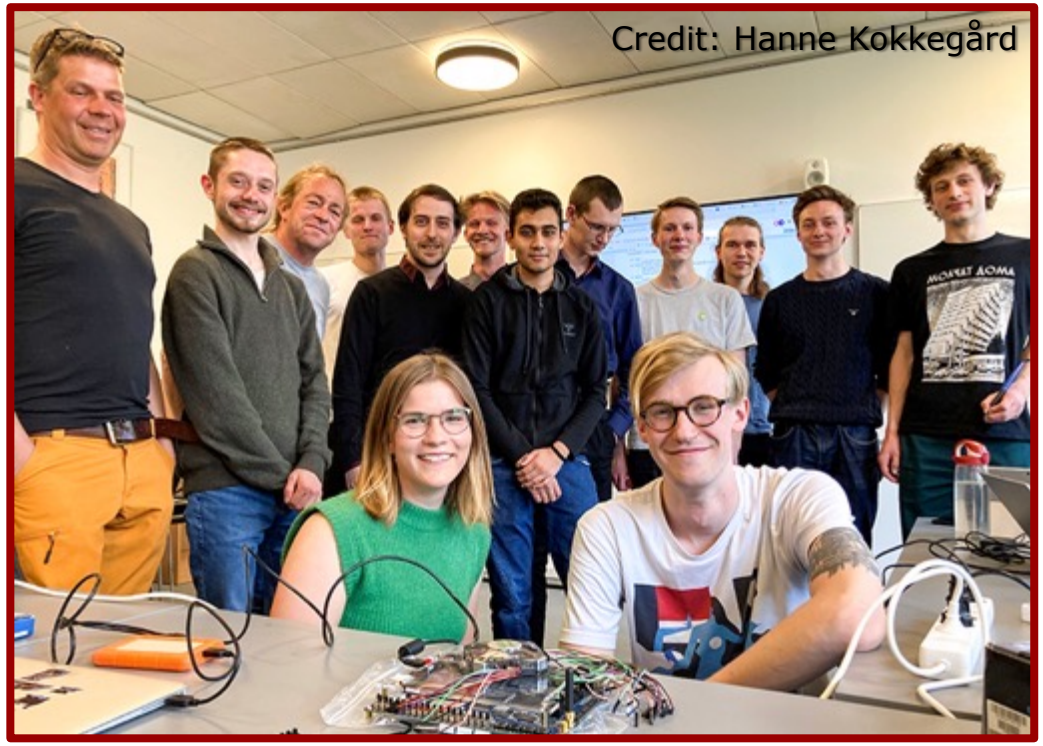
Overview

Focus: Using open-source chip design tools in academic education

Content: Report on a special course at the Technical University of Denmark

- Toolchain
- Hardware architecture
- Challenges and learning values

The team



Outline

- Introduction
- The open-source toolchain
- The hardware design
- Challenges and learning values

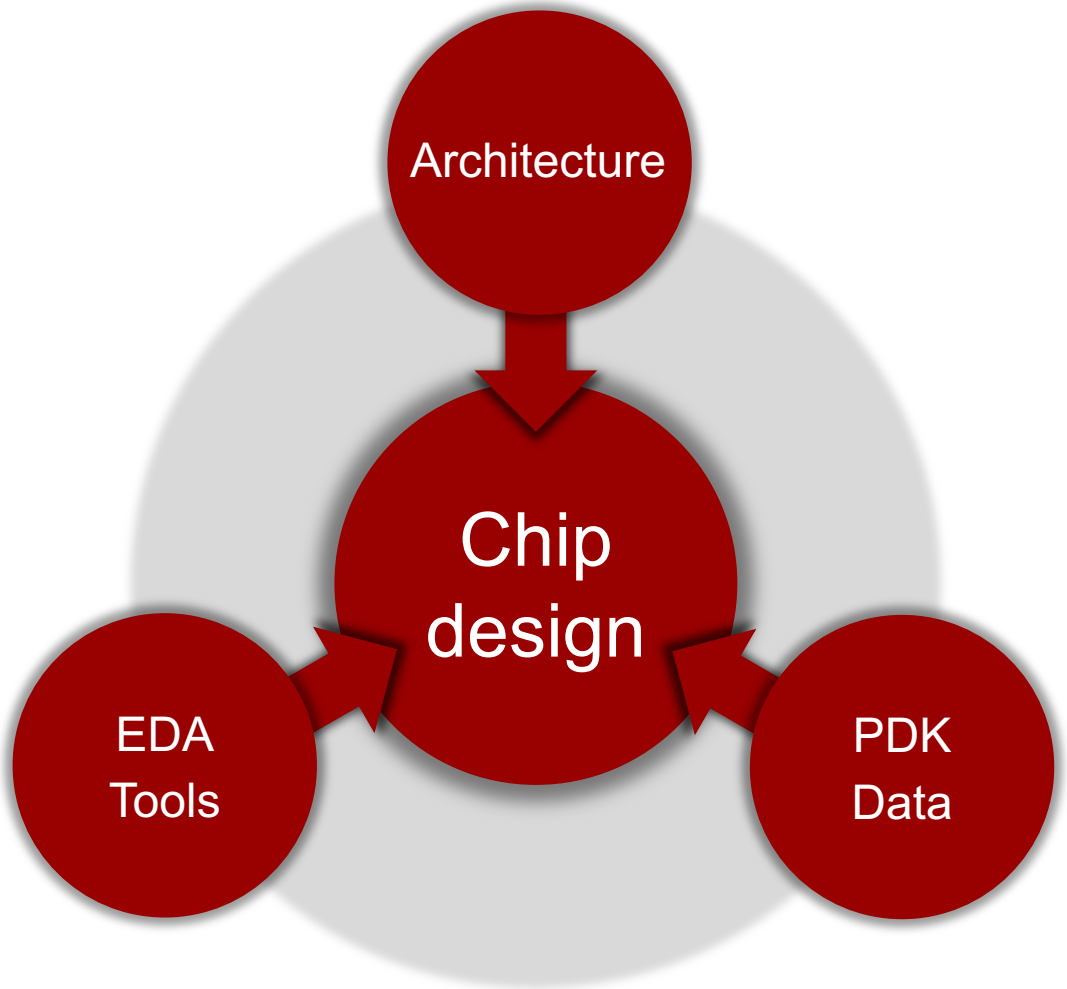
Outline

- **Introduction**
- The open-source toolchain
- The hardware design
- Challenges and learning values

Motivation

- Current global chip shortage crisis started in 2020
  - COVID-19 pandemic
  - Weather incidents
  - Complex geopolitical situation
- Many industries strongly depend on semiconductor chips
- The European Chip Act is Europe's response to the crisis
  - Strengthen Europe's resilience and competitiveness
  - Encourage chip design and production within the countries of the union
- Universities should produce candidates acquainted the chip design processes

The *Open-Source Chip Design* special course



- 13 weeks course - 5 ECTS
- Learning objectives
  - Define precise specifications of a computing system
  - Develop (in Chisel) and test the hardware architecture
  - Use EDA tools and solve related challenges
- 12 students signed up for the course

Enabling technologies

Open source 130nm Skywater PDK

- Full open source toolchain
- From RTL down to the GDS

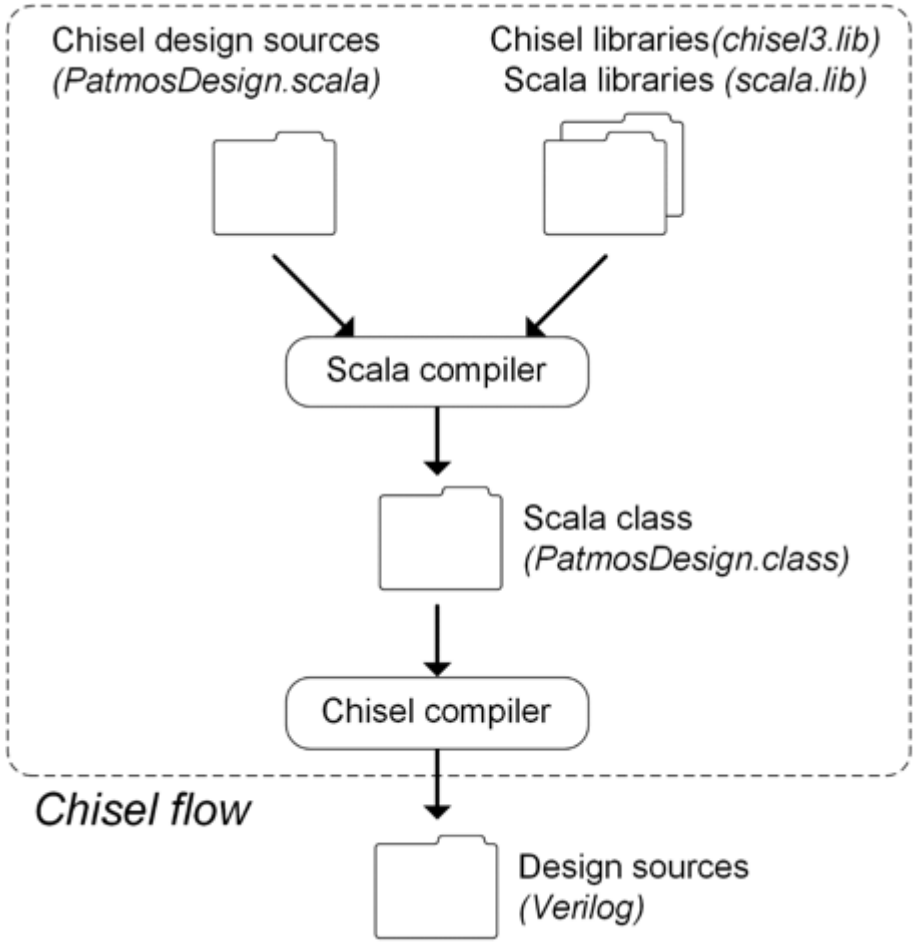
Open source shuttle program

- Free chip fabrication
- Design has to be open source

Outline

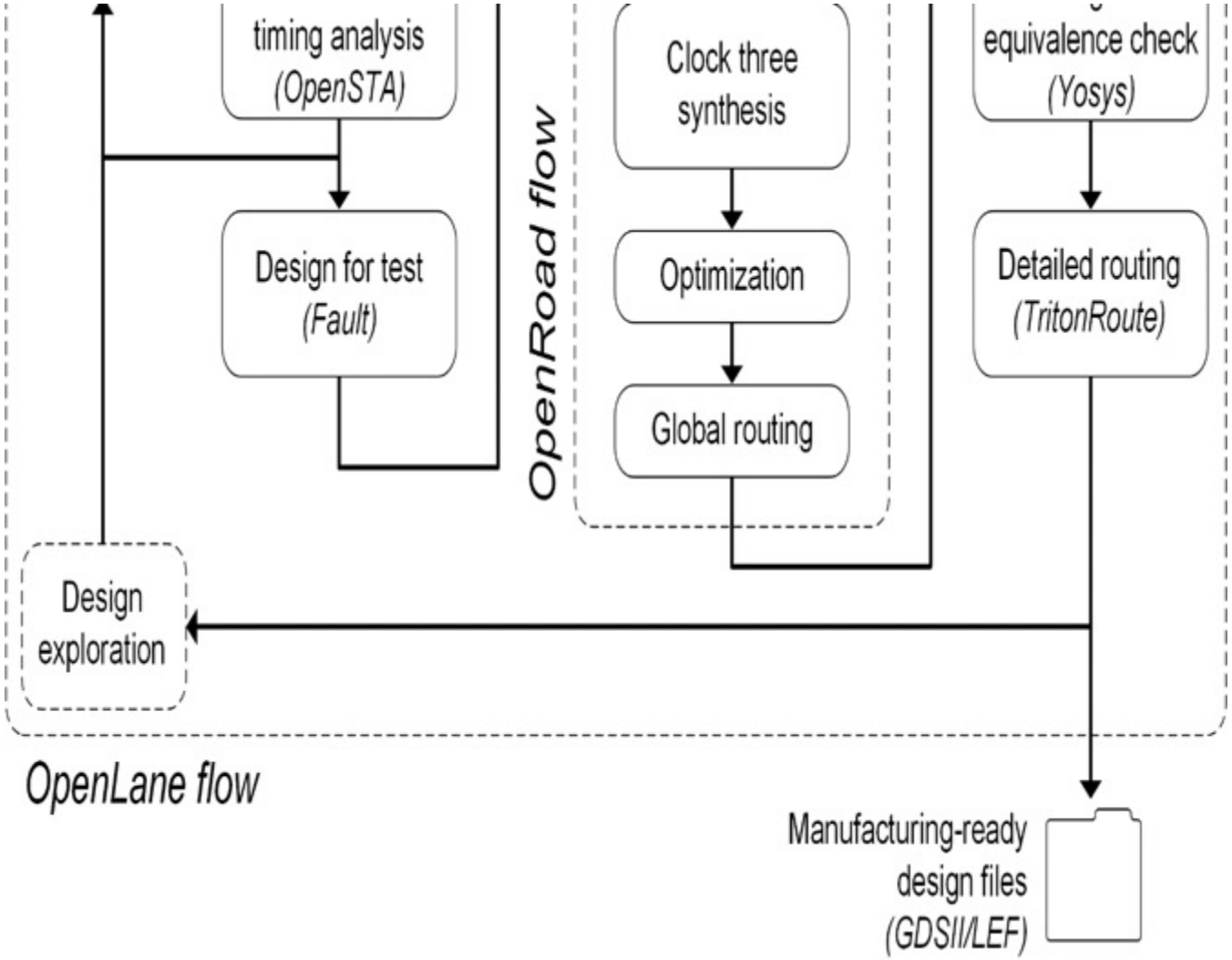
- Introduction
- **The open-source toolchain**
- The hardware design
- Challenges and learning values

The Chisel hardware construction language



- Chisel is a domain specific language, embedded in Scala
- Allows RTL description
- Two steps in the compilation
- Generates Verilog
- Enables co-simulation of Scala testers, Chisel designs, and Java or C-based golden models

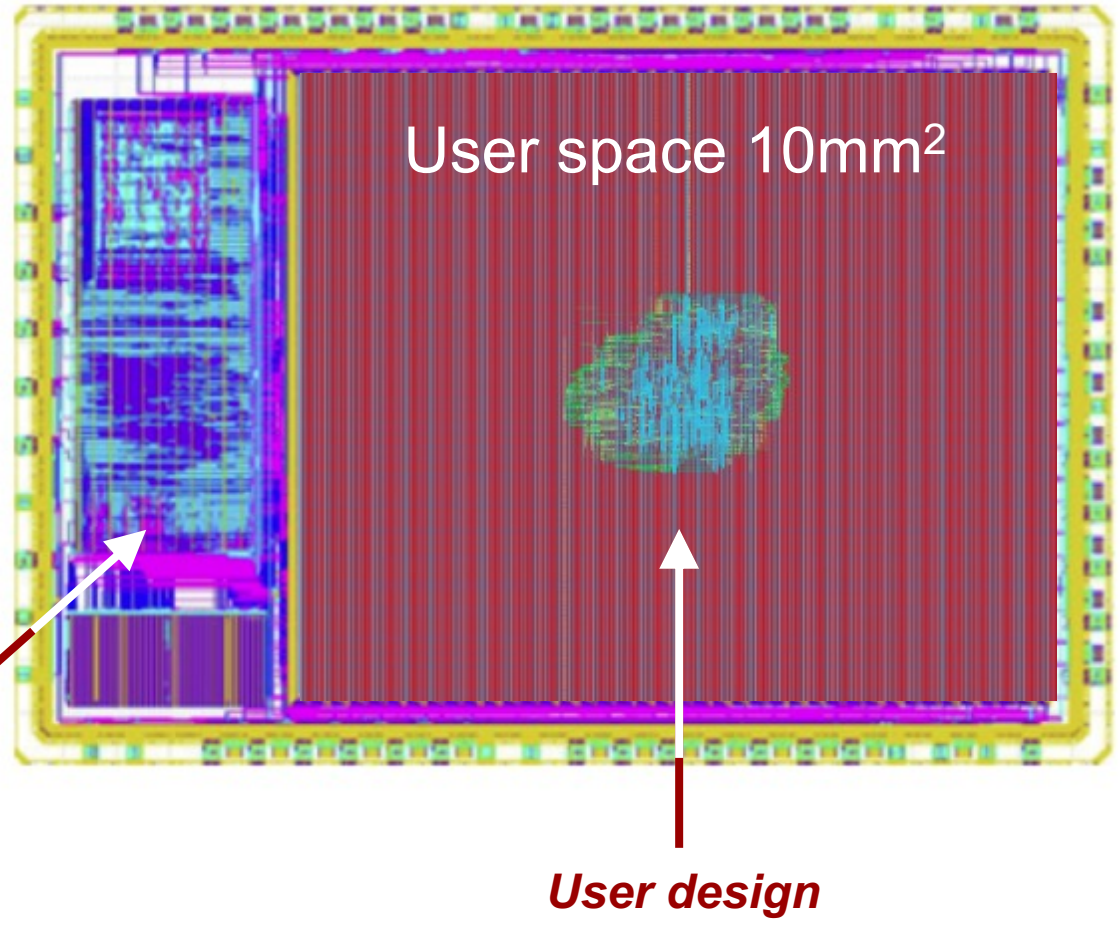
OpenLane Flow



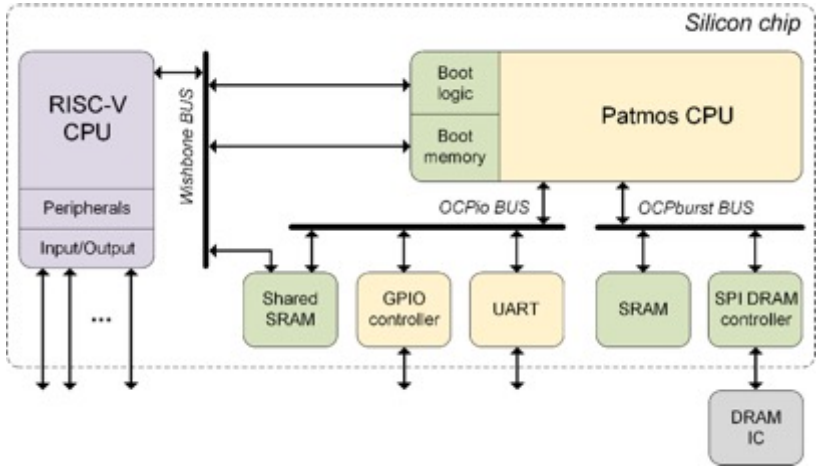
- Start form our Verilog design
- The SkyWater PDK is used
- Chain of different open-source tools
  - Yosys for synthesis
  - OpenRoad (end-to-end silicon compiler)
  - Others
- Produces GDS layout files
- Allows for Intermediate simulations

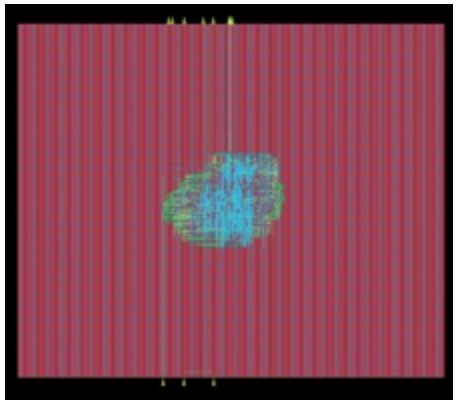
Caravel project

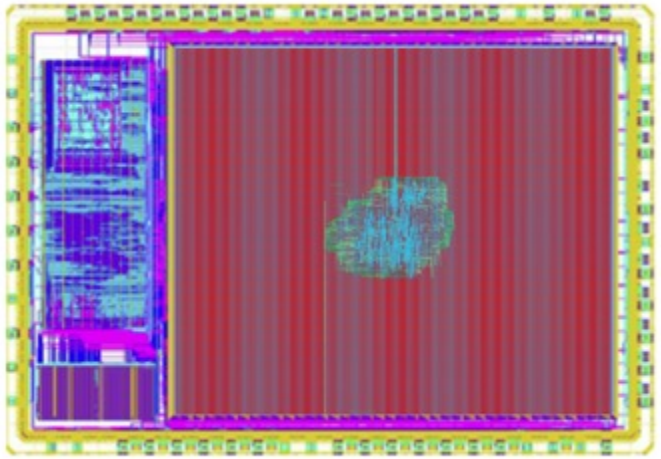
- Pre-made harness including I/Os
- Small RISC-V CPU for managements
- Scripts for using the OpenLane toolchain
- 10mm^2 of user space
- Starting sample project

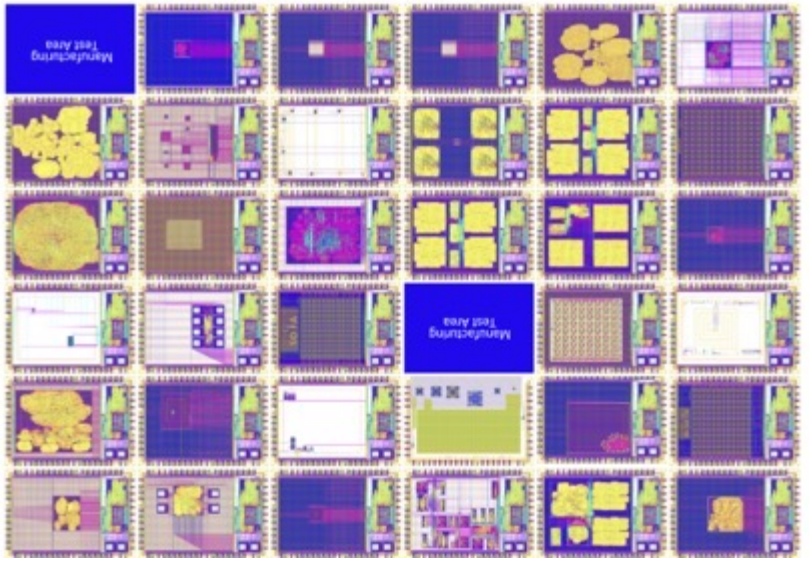


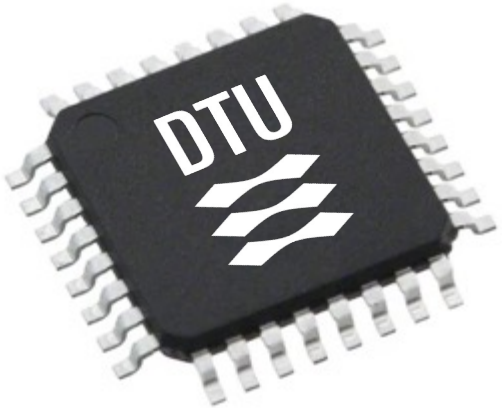
The full process from RTL to chip









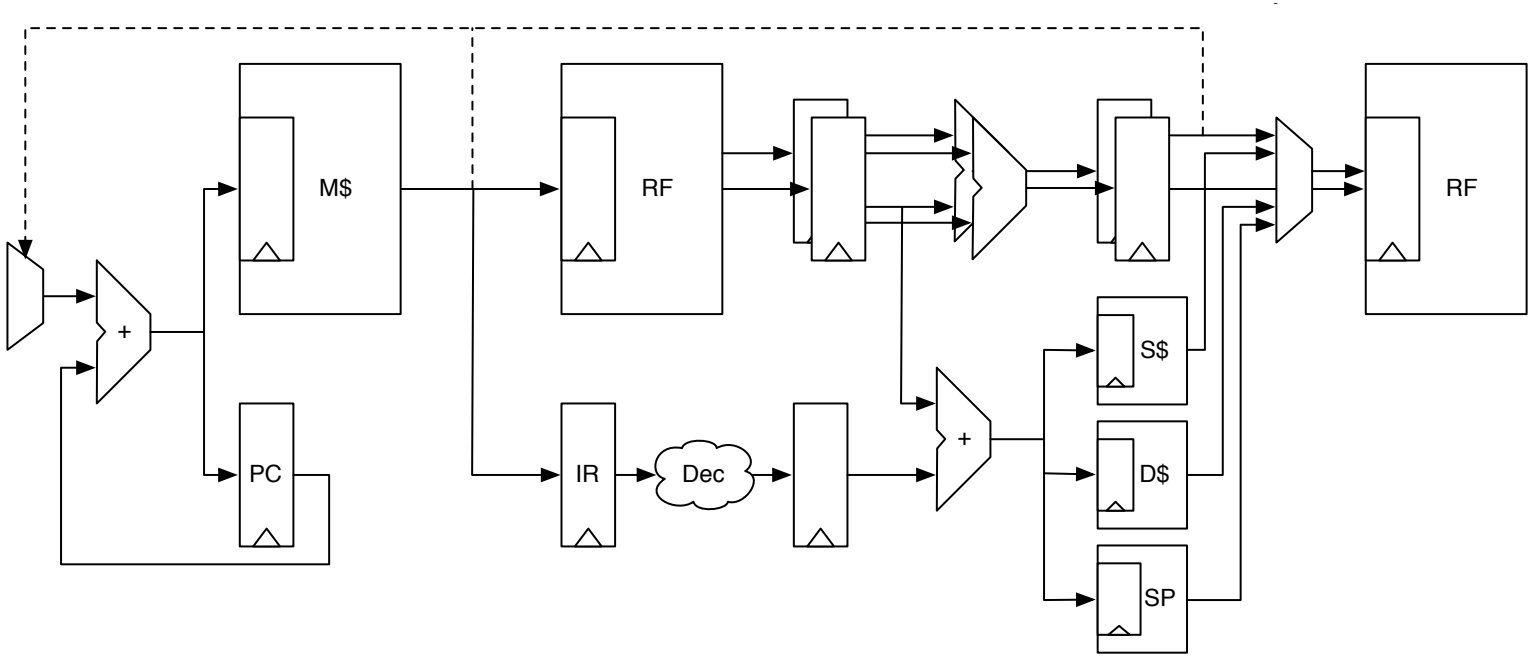


Outline

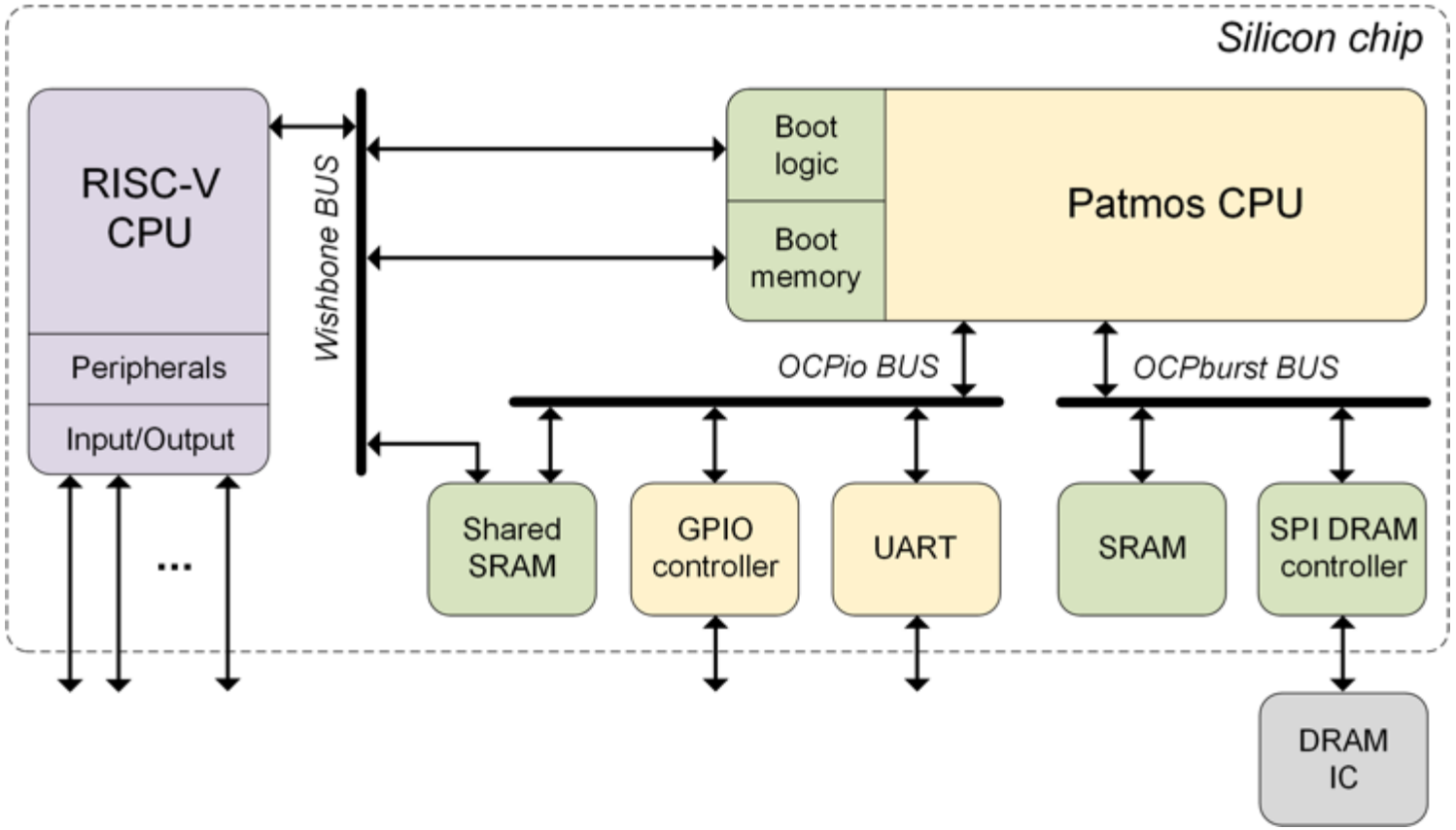
- Introduction
- The open-source toolchain
- **The hardware design**
- Challenges and learning values

The Patmos processor

- A research-oriented time-predictable RISC-style processor written in Chisel
- Part of the open-source T-CREST multicore project
- The LLVM compiler is available for compiling C



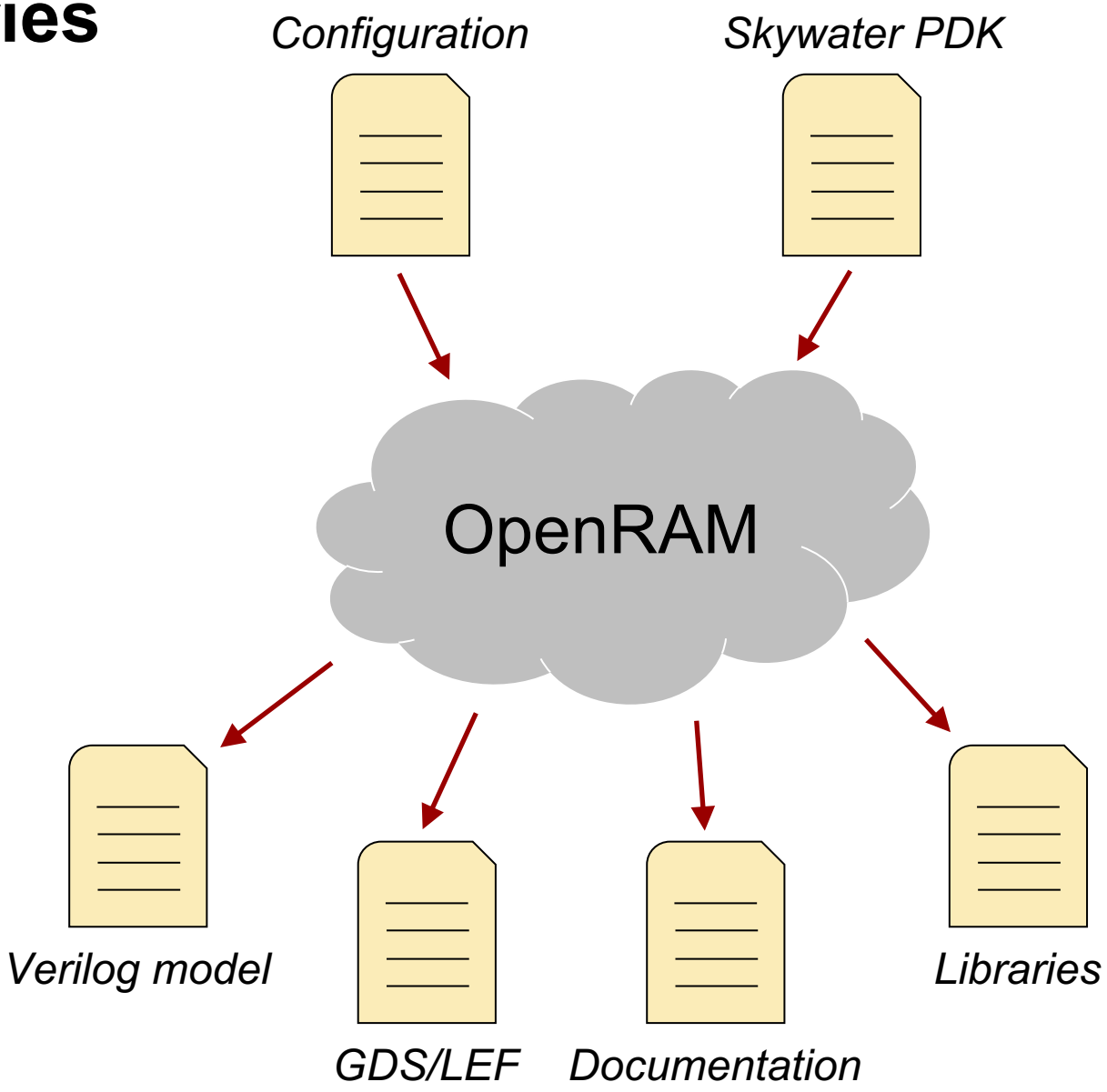
Our architecture



- Includes the Patmos CPU
- Multiple booting modes
  - Boot memory
  - Boot from other memories
  - Make sure we can boot in case of problems
- Peripherals
  - SPI controller for off-chip DRAM
  - On-chip SRAM
  - UART and GPIOs

Internal (on-chip) memories

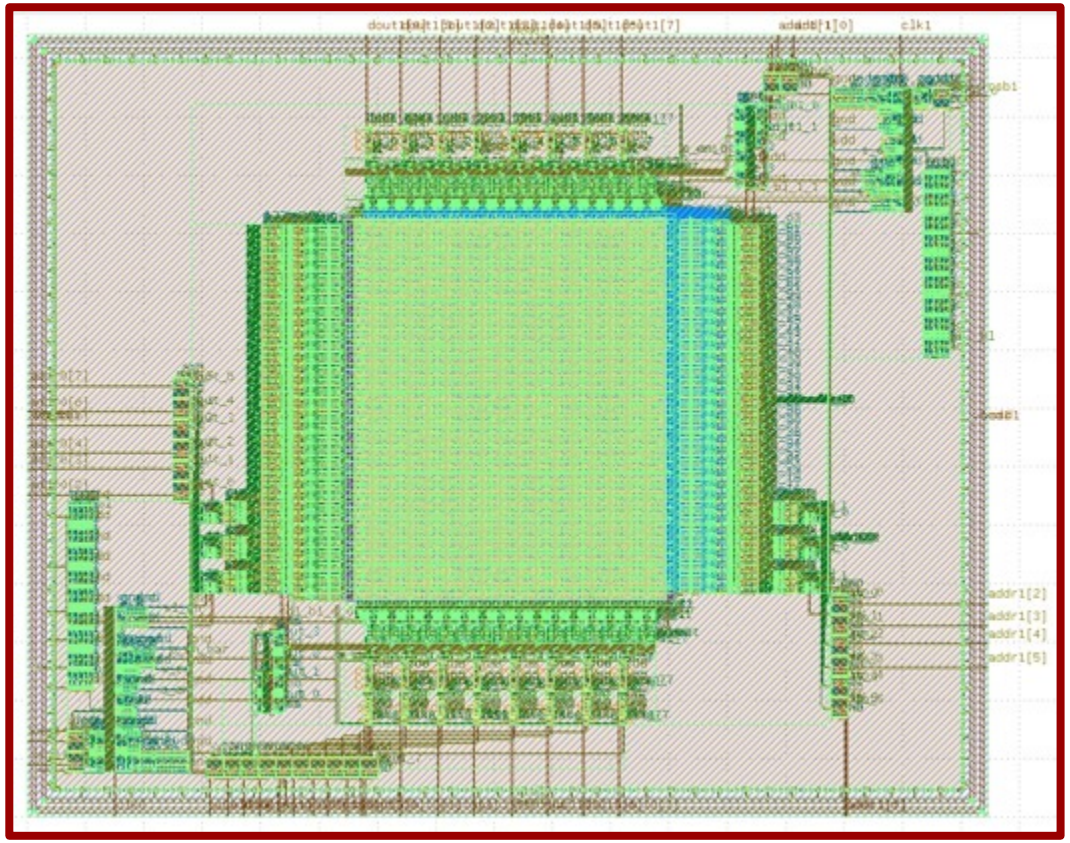
- Memories need to be generated
- We use OpenRAM
  - Precompiled macros
  - Custom memories



Internal (on-chip) memories

- Memories need to be generated
- We use OpenRAM
  - Precompiled macros
  - Custom memories

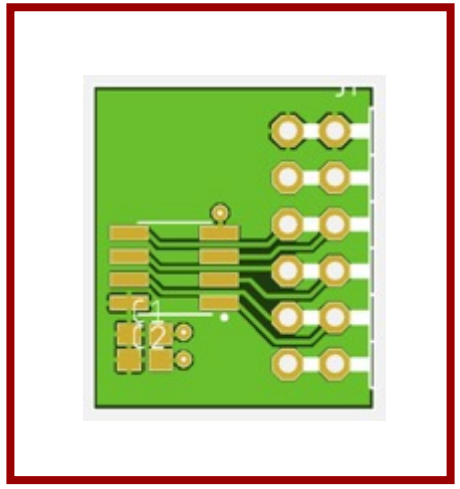
SRAM - sky130 - 1k x 8bit

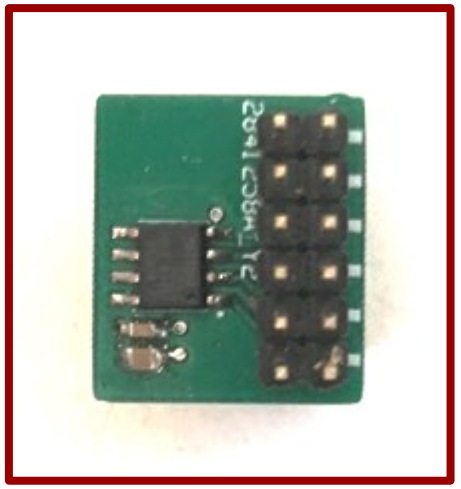


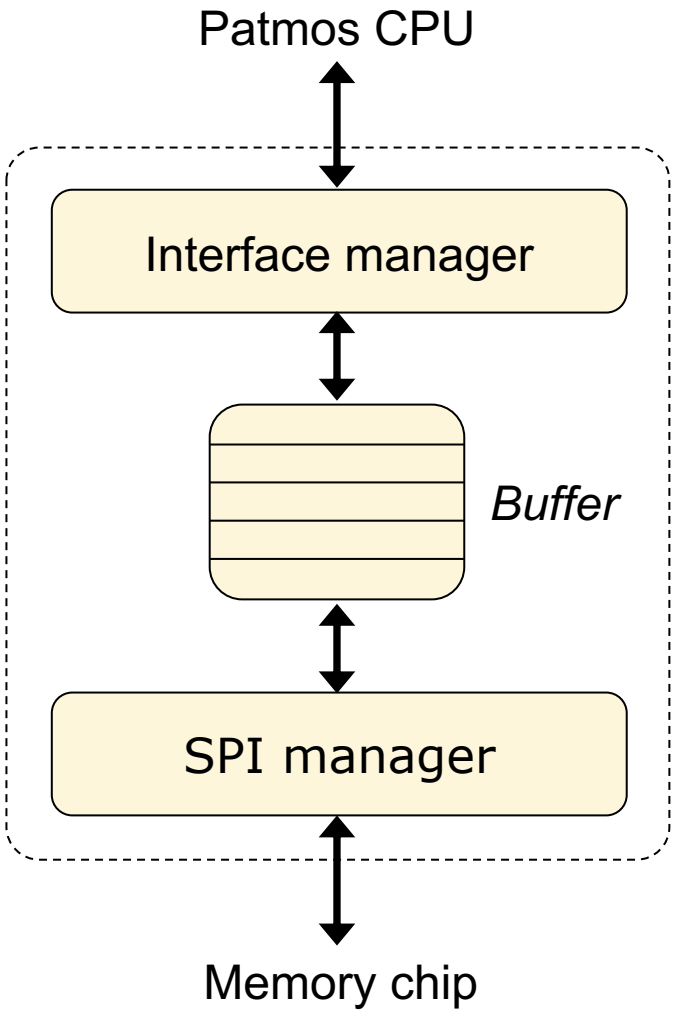
External memories

- 10mm^2 cannot fit large memories
- Off chip DRAM (or Flash)
- We developed a memory controller

- We tested on the real chips



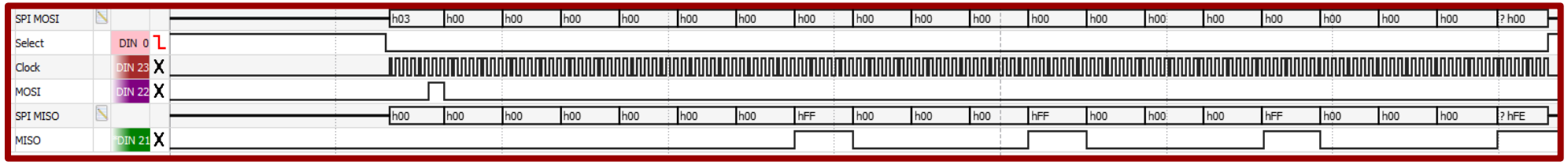




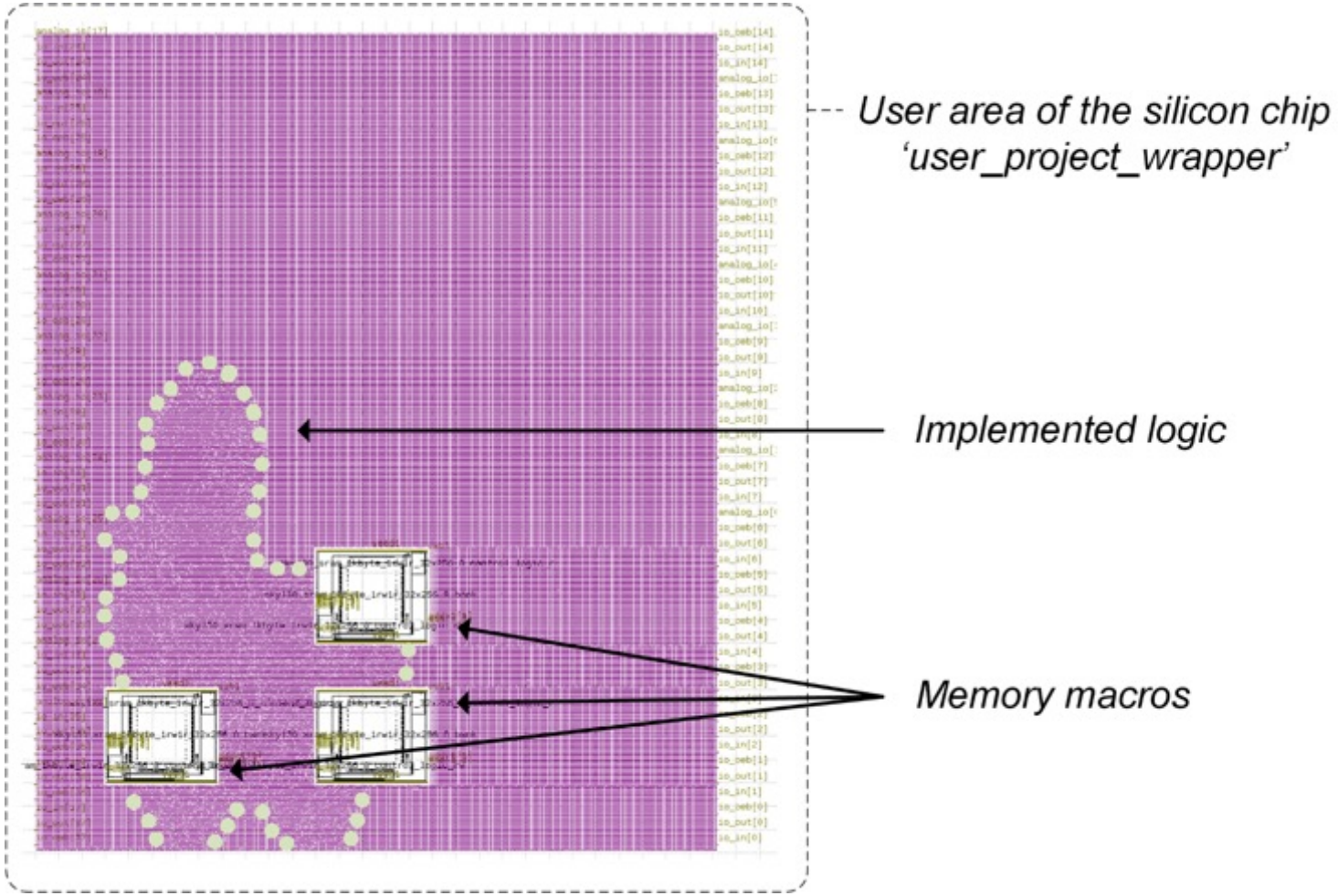
Testing

- Simulation with test-benches
- Simulation running programs

- Test on FPGA + logic analyzer



Final user space layout



Outline

- Introduction
- The open-source toolchain
- The hardware design
- **Challenges and learning values**

Challenges

- The open-source project moves quite fast and the documentation falls behind
  – Use of the Slack community was needed

- High memory usage during synthesis and long synthesis times (8 to 10 hours)
  – Use of multiple servers to have multiple runs in parallel

- Multiple teams working on different parts of the project led to challenging integration
  – Weekly coordination, fast integration and testing

Learning values

- The possibility to have a chip manufactured free of charge is a strong motivating factor
  - Attend the course meetings
  - Actively participate in the project work
  - Overcome the challenges and frustrations

- Open-source tools mitigates the entry barrier of tools
  - No monetary or other licensing is required
  - Support of the community

- Team-work is needed
  - Complex tasks cannot be performed in isolation
  - Promotes strongly interact between students and teams

Conclusion

- The first open-source chip developed at DTU
- The chip is now taped-out on the MPW7 shuttle program
  – 40 chips + 5 evaluation boards including the chips
  – But there was a after deadline change in Caravel
- Additional work (BSc projects, special courses) for the verification of the produced chip
- The open-source toolchain is challenging, but also had very good learning values

---

- The project is hosted on GitHub (see link on the web site)
- The project also got press coverage (see link in the paper)