www.bsc.es

MIRI-PD
Processor Design
Fall 2021

Labs Introduction

Leonidas Kosmidis

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

14/09/2021

Course Logistics

Professors

- Roger Espasa
- Leonidas Kosmidis
  - Email: leonidas.kosmidis@bsc.es
  - Office hours: agreed through email

- https://personals.ac.upc.edu/lkosmidi/PD
- Lab information

Course Structure

- Theory Lectures
  - Thursday 12-14
- Lab Lectures
  - Tuesday 12-14
  - On-line through Zoom
- Project
  - 6 steps to develop your project

Evaluation

- Final mark will be based on your performance at:
  - Project
  - Presentation on a research topic (T)
- Final mark to be computed as:
  - **0.8 x Project + 0.2 x T**

Lab/Project Schedule

Lab Session	Session Dates	Deliverable Due Date
Lab 1. Infrastructure setup and test. Baseline microprocessor selection.	14/09	28/09
	21/09	
Lab2. Module definition and specification. Workplan	28/09	19/10
	No lab 12/10	
	5/10	
Lab3. Module implementation and verification in isolation	19/10	16/11
	26/10	
	02/11	
	09/11	
Lab4. Module review	Week of 16-20/11	Interview
Lab5. Integration with a pipelined CPU	16/11	21/12
	23/11	
	30/11	
	No lab 7/12	
	14/12	
Lab6. Project review	Between 14-23/12	Interview
Extension of original module	21/12	18/01
	11/1	
	18/1	

Module

- Each group will choose an advanced module to extend their baseline processor

Possible examples:

1. Associative cache
2. Branch predictor
3. Store forwarding queue
4. Error detection and correction codes in the memory
5. High performance functional units (i.e. adder, subtracter…)
6. Accelerators (cryptography, neural nets...)

Module

- Each group will choose an advanced module to extend their baseline processor

Possible examples:

7. Multi-issue / superscalar
8. Vector extensions
9. Coherent cache for a multiprocessor
10. Random placement cache
11. Lockstep functionality
12. Floating point Unit subset
13. Other modules possible upon agreement with the instructor and the background of the students. The instructor encourages the implementation of solutions for safety-critical and real-time features

Baseline Processor

- Can be the processor from PA or different
- Can be in Verilog, VHDL or a higher level HDL language such as Chisel or Bluespec
- Can be an open source or commercial CPU core or GPU
- Must not include the selected module

Baseline Processor

- Examples of Available Options
- Open Source or Commercial Open Sourced CPU Cores:
  - RISC-V: Ariane (Verilog), Rocket (Chisel), SHAKTI (BlueSpec), SweRV (System Verilog), NOEL (VHDL), Lagarto (Verilog), Black Parrot (Verilog) …
  - SPARC: LEON3/LEON4 (VHDL), OpenPiton (Verilog), OpenSPARC T1/T2 (Verilog)
  - Power: OpenPower Microwatt (VHDL), Power-A2 (VHDL)
  - X86: Zet (Verilog)
- Commercial CPU Cores (through academic licences/collaboration):
  - ARM, MIPS
- GPU/Accelerator Cores:
  - Commercial: ARM Mali-400, ThinkSilicon Nema Pico
  - Open Source: NVIDIA NVDLA (Verilog)

Tools/Verification

Compete freedom on the tool selection

- Open source tools
  - Verilator, Icarus Verilog, GHDL…
- Commercial
  - Questa Sim, Vivado, ISE, Quartus…
- FPGA
  - 3 kits available: PYNQ Z1, 2 x Kria KV260

Bibliography

- Weste, N. H., & Harris, D. “CMOS VLSI Design : A Circuits and Systems Perspective”. 4th Edition, 2010.
- Kahng AB, Lienig J, Markov IL, Hu J. “VLSI Physical Design: from Graph Partitioning to Timing Closure”. Springer Science & Business Media; 2011 Jan 27.
- Mead, C., & Conway, L. Introduction to VLSI systems (Vol. 1080). Reading, MA: Addison-Wesley, 1980

www.bsc.es

MIRI-PD
Processor Design
Fall 2021

Labs Introduction

Leonidas Kosmidis

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

14/09/2021