Digital Design and Computer Architecture II

Module Description

Module code	CS221-F27	**Credits**	7 ECTS
Programme	Foundation programme	**Level**	Semester 3
Semester	Fall 2027	**Prerequisites**	CS102-F26 Digital Design and Computer Architecture I
Module leader	Dr. Ali Khudiyev	**Email**	ali.khudiyev@wosm.academy
Office hours	Monday–Friday, 10:00–17:00 (Baku time, UTC+4)		

1. Module Overview

This module introduces the organisation and design of modern computer systems, from the instruction set architecture to the processor implementation, the memory hierarchy and input/output. It continues Digital Design and Computer Architecture I (CS102-F26), where you built the 8-bit DX8 computer from logic gates, and moves to the scale of a real processor: the RISC-V instruction set.

The first part looks at the computer from the programmer's side: data representation, assembly-language programming in RISC-V, the Von Neumann architecture, other instruction set architectures, system architecture, and memory management with caches. The second part goes down to the hardware: Boolean algebra, adders and multipliers, memory elements and finite automata, and the design of single-cycle, multi-cycle and pipelined processors, together with modern parallel systems and the I/O system. The last part treats the design process itself: logic minimisation with Karnaugh maps and binary decision diagrams, satisfiability (SAT) as a tool for circuit design and verification, and the physical design of chips.

2. Aims

- To explain how software instructions are executed by hardware.
- To develop an understanding of processor design and performance.
- To give practical experience of programming a real instruction set architecture (RISC-V) in assembly language, including its calling convention.
- To explain how the memory hierarchy, caches, parallelism and input/output shape the performance of a computer system.
- To introduce the methods and tools used to design, optimise, verify and physically implement digital circuits.

3. Learning Outcomes

On successful completion of this module, students will be able to:

LO1. Describe the main components of a computer system and how they interact.

LO2. Write and analyse programs at the assembly-language level.

LO3. Explain datapath and control design, including pipelining and hazards.

LO4. Evaluate the effect of cache and memory hierarchy on performance.

LO5. Design and analyse combinational and sequential circuits, including adders, multipliers, memory elements and finite automata.

LO6. Apply logic minimisation (Karnaugh maps, binary decision diagrams) and SAT solving to circuit design, and describe the main steps of physical design.

4. Indicative Content

- Performance metrics and Amdahl's law
- Data representation and computer arithmetic
- Instruction set architecture (RISC-V, compared with other ISAs such as MIPS, ARM and x86)
- Assembly-language programming, calling conventions and recursion
- Von Neumann architecture and system architecture
- Memory hierarchy: caches, virtual memory
- Boolean algebra; adders and multipliers; memory elements; finite automata
- Single-cycle and multi-cycle datapath; control unit
- Pipelining, hazards and branch prediction
- I/O, buses and interrupts; implementation at the physical layer
- Introduction to parallel and multicore architectures
- Logic minimisation: Karnaugh maps and binary decision diagrams
- Satisfiability (SAT) in circuit design and verification
- Physical design: partitioning, placement, routing and timing

5. Weekly Schedule

Each week has two lectures, a central exercise with worked examples and a tutorial in which the solutions of the week's exercise sheet are discussed.

Week	Topic	Activity
1	Introduction and Data Representation	Lectures + Exercise 1
2	Assembly Languages I and II	Lectures + Exercise 2
3	Assembly Example; Von Neumann Architecture	Lectures + Exercise 3
4	Other ISAs; System Architecture	Lectures + Exercise 4
5	Memory Management and Caches I and II	Lectures + Exercise 5
6	Boolean Algebra; Adders	Lectures + Exercise 6
7	Multiplication; Memory	Lectures + Midterm exam
8	Processor Architecture I and II	Lectures + Exercise 7
9	Finite Automata; Multi-cycle Processor	Lectures + Exercise 8
10	Modern Systems and Parallelism I and II	Lectures + Exercise 9
11	The I/O System; Physical Layer Implementation	Lectures + Exercise 10
12	Pipelining I and II	Lectures + Exercise 11
13	Karnaugh Maps; Binary Decision Diagrams	Lectures + Exercise 12
14	SAT; Physical Design	Lectures + Exercise 13

6. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle: recorded lectures with slides, central exercises with worked examples, weekly exercise sheets with tutorial sessions, and lab assignments. Assembly programs are written and run in a RISC-V simulator, and circuits are built in

the Logisim-evolution circuit simulator. Questions are discussed in the module channel on Discord and in office hours.

Activity	Hours
Lectures	41
Labs / Tutorials	39
Independent study	120
Total	**200**

7. Assessment

The lab assignments are RISC-V assembly programs and circuit designs, submitted weekly. The midterm exam covers Weeks 1–6; the final exam covers the whole module.

Component	Weight	Due	LOs
Lab assignments	20%	Weekly	LO2, LO3, LO5
Midterm exam	30%	Week 7	LO1, LO2, LO4, LO5
Final exam	50%	End of module	LO1–LO6
Total	**100%**		

8. Reading List

Essential

- D. A. Patterson, J. L. Hennessy: *Computer Organization and Design: The Hardware/Software Interface, RISC-V Edition*, 2nd ed., Morgan Kaufmann, 2020.

Recommended

- J. L. Hennessy, D. A. Patterson: *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2017.
- W. Stallings: *Computer Organization and Architecture*, 11th ed., Pearson, 2019.
- S. L. Harris, D. Harris: *Digital Design and Computer Architecture: RISC-V Edition*, Morgan Kaufmann, 2021.
- A. B. Kahng, J. Lienig, I. L. Markov, J. Hu: *VLSI Physical Design: From Graph Partitioning to Timing Closure*, 2nd ed., Springer, 2022.

9. Materials and Licence

Course materials are published under the Creative Commons Attribution-ShareAlike 4.0 licence unless stated otherwise. See wosm.academy/licence for details.

10. Policies

- **Attendance:** Attendance is not mandatory. The module is self-paced, so you can study the materials when it suits you.
- **Late submissions:** Work submitted after the deadline is penalised.
- **Academic integrity:** All submitted work must be your own. See the Code of Conduct.