Digital Design and Computer Architecture I

Module Description

Module code	CS102-F26	**Credits**	7 ECTS
Programme	Foundation programme	**Level**	Semester 1
Semester	Fall 2026	**Prerequisites**	None
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 shows how a computer is built, starting from nothing more than switches that are on or off. It follows the module leader's book *A Student's Computer Architecture Book I*: you first learn the fundamentals of digital design, and then use them to build a working 8-bit computer, the DX8, piece by piece.

The first part covers the foundations: numeral systems, Boolean algebra and Boolean functions, logic gates, combinational circuits and sequential circuits, including finite state machines, latches, flip-flops, counters and timing. The second part puts these building blocks together. You design an instruction set architecture, build an arithmetic logic unit, registers and random access memory, and connect them with a stepper and a controller into a multi-cycle processor that fetches, decodes and executes instructions.

The module is for anyone who is curious about how computers really work, and no prior knowledge of electronics is needed. Digital Design and Computer Architecture II (Semester 3) continues with real instruction set architectures, memory systems, input/output and processor performance.

2. Aims

- Show how a working computer is built up from logic gates, one layer of abstraction at a time.
- Give a mathematical foundation in Boolean algebra for describing, designing and simplifying digital circuits.
- Develop practical skill in designing, simulating and testing combinational and sequential circuits.
- Give you the understanding and confidence to design and build computing devices of your own.

3. Learning Outcomes

After completing this course, you will be able to:

LO1. Represent numbers in positional numeral systems, in particular binary, octal and hexadecimal, and convert between them.

LO2. Manipulate Boolean expressions with the laws and theorems of Boolean algebra, derive canonical sum-of-products and product-of-sums forms from truth tables, and simplify Boolean functions with Karnaugh maps.

LO3. Implement Boolean functions with logic gates, including universal gate sets, and design and analyse combinational circuits such as multiplexers, decoders, encoders, adders and comparators, including their propagation and contamination delays.

LO4. Model sequential behaviour with Moore and Mealy finite state machines, build latches, flip-flops, registers and counters, and explain the setup and hold time constraints of sequential

circuits.

LO5. Explain the Von Neumann architecture and the role of an instruction set architecture, and design an arithmetic logic unit, a register file and byte-addressable memory from basic components.

LO6. Explain how a processor executes instructions in fetch, decode, execute and load stages, compare single-cycle, multi-cycle and pipelined designs, and build and program a working multi-cycle 8-bit computer in a circuit simulator.

4. Syllabus

Unit	Topic
1	The essence of computing: computation as change of state; a short history of computing tools
2	Numeral systems: additive and positional systems; binary, octal and hexadecimal
3	Boolean algebra: laws and theorems
4	Boolean functions: truth tables; canonical sum-of-products and product-of-sums forms
5	Simplifying Boolean functions with Karnaugh maps
6	Logic gates and universal gate sets
7	Combinational circuits I: multiplexers and demultiplexers
8	Combinational circuits II: encoders, priority encoders and decoders
9	Timing of combinational circuits: propagation and contamination delays, glitches
10	Finite state machines: Moore and Mealy machines; FSM simplification
11	The clock; memory: SR latch and D latch
12	Flip-flops: D (master–slave, asynchronous reset), T and JK flip-flops
13	Counters: synchronous up counter and asynchronous ripple counter
14	Timing of sequential circuits: setup and hold time constraints
15	Instruction set architecture: Von Neumann architecture; the DX8 instruction format and instruction set
16	Arithmetic: half adder, full adder, multi-bit adders; two's complement
17	Logic and comparison: bitwise operations and comparators
18	The arithmetic logic unit: shifts and rotations; selecting the operation
19	Memory: registers, addressing, byte-addressable RAM and the register file
20	Processor design: fetch, decode, execute and load; single-cycle, multi-cycle and pipelined designs
21	The stepper, the data path and the controller
22	Putting it all together: building and programming the DX8 computer
23	Course Review & Exam

5. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle. The module follows the module leader's book *A Student's Computer Architecture Book I*. Each unit has a lecture and exercises in which you build and test the circuits of the unit in the Logisim-evolution circuit simulator; circuit files for the examples in the book are provided. Verilog models of many of the circuits are also available for students who want to go further. 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.

Activity	Hours
Lectures	34
Labs / Tutorials	36
Independent study	130
Total	**200**

6. Assessment

Your understanding and knowledge of the concepts presented will be assessed through quizzes, a circuit-design project and the final exam. In the project you build the DX8 computer, or an extension of it, in Logisim-evolution and run your own programs on it. Any permitted support materials will be provided during the examination; no additional aids or resources are allowed.

Component	Weight	Due	LOs
Quizzes	20%	Throughout	LO1–LO5
Circuit-design project: the DX8 computer	40%	Throughout	LO3–LO6
Final exam	40%	End of module	LO1–LO6
Total	**100%**		

7. Reading List

Essential

- A. Khudiyev: *A Student's Computer Architecture Book I*, 2026. Free online at alikhudiyev.com. Circuit files and Verilog models: DX8 course material.

Recommended

- N. Nisan, S. Schocken: *The Elements of Computing Systems: Building a Modern Computer from First Principles*, 2nd ed., MIT Press, 2021.
- Sarah L. Harris, David Harris. 2022. *Digital Design and Computer Architecture: RISC-V Edition.* Elsevier.
- David A. Patterson, John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface.*
- C. Petzold: *Code: The Hidden Language of Computer Hardware and Software*, 2nd ed., Microsoft Press, 2022.

8. 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.

9. 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.