Operating System and Computer Architecture

# RISC-V Assembly Programming

Sami Taktak

2025

## Learning Objectives

— Identify and differentiate RISC-V instruction formats (R-type, I-type, S-type, B-type)

— Encode and decode instructions in hexadecimal machine code

— Manipulate registers and understand function calling conventions

— Implement control structures (loops, conditions) in assembly language

— Use memory operations (load/store) to manipulate arrays

## Work Required

— Exercises must be completed individually on paper

— Justifications and register tracing must be clearly written

— Total indicative duration: 2h00

## 1 Instruction Formats Reference

- **R-type:** `funct7 (7) | rs2 (5) | rs1 (5) | funct3 (3) | rd (5) | opcode (7)`
- **I-type:** `imm[11:0] (12) | rs1 (5) | funct3 (3) | rd (5) | opcode (7)`
- **S-type:** `imm[11:5] (7) | rs2 (5) | rs1 (5) | funct3 (3) | imm[4:0] (5) | opcode (7)`
- **B-type:** `imm[12] | imm[10:5] | rs2 (5) | rs1 (5) | funct3 (3) | imm[4:1] | imm[11] | opcode (7)`

### ** Exercise 1 Instruction Encoding

**Task 1:** Decode the following machine code instructions (given in hexadecimal):

1. `0x00450533` (R-type)
2. `0x00A30313` (I-type)

3. `0xFE010113` (I-type)

**Task 2:** Encode these assembly instructions into machine code:

1. `add x6, x7, x8`
2. `lw x9, 8(x10)`
3. `beq x11, x12, -16`

## * Exercise 2 Code Tracing

Trace through the following code and fill in the register values after each instruction.

```
addi x10, x0, 5       # x10 = ?
addi x11, x0, 3       # x11 = ?
add  x12, x10, x11    # x12 = ?
sub  x13, x10, x11    # x13 = ?
slli x14, x10, 2      # x14 = ?
ori  x15, x10, 8      # x15 = ?
```

Create a table showing the values of registers x10 through x15 after each instruction.

## *** Exercise 3 Algorithm Design

Design RISC-V assembly code (on paper) for the following algorithms:
**Part A:** Array Sum
Write code to sum all elements in an array.

- Array base address is in `x10`
- Array length is in `x11`
- Store result in `x12`

**Part B:** Find Maximum
Write code to find the maximum value in an array.

- Array base address is in `x10`
- Array length is in `x11`
- Store maximum in `x12`

For each algorithm, write pseudocode first, then convert to RISC-V assembly.