-

Department of Electrical and Computer Engineering
Summer Semester, 2022/2023

Project No. 2
ENCS4370| Computer Architecture
Deadline: September 7, 2023 at 23:59

---

1. Objectives:

To design and verify a simple single cycle RISC processor in Verilog

2. Processor Specifications

1. The instruction size is 16 bits
2. Eight 16-bit general-purpose registers: from R0 to R7.
3. The word size is 16-bit.
4. A special purpose 16-bit register for the program counter (PC)
5. Three instruction types (R-type, I-type, and M-type).
6. Separate data and instructions memories
7. Word addressable memory

3. Instruction Types and Formats

The instruction set of this processor has three instruction formats, namely, R-type, I-type, and M-type. These three types have the following formats.

R-Type (Register Type) Format

- **4-bit Opcode**
- **3-bit Rd:** destination register
- **3-bit Rs1:** first source register
- **3-bit Rs2:** second source register
- **3-bit unused**

Opcode^4	Rd^3	Rs1^3	Rs2^3	Unused^3

I-Type (Immediate Type) Format

- **4-bit Opcode**
- **3-bit Rd:** destination register
- **3-bit Rs1:** first source register
- **6-bit immediate:** unsigned for logic instructions, and signed otherwise.

Opcode^4	Rd^3	Rs1^3	Immediate^6

M-Type (Memory Type) Format

- **4-bit Opcode.** It has the value of zero for load and store instructions
- **1-bit M (Mode):** M=0 means memory read, and M=1 means memory write.
- **3-bit Rd:** destination register in the case of load and source register in the case of store.
- **8-bit Absolute Address**

Opcode^4	M^1	Rd^3	Absolute Address^8

4. Instruction Encoding

You are required to implement a subset of this processor's instructions, which are detailed in the table below.

No.	Instr	Meaning	Opcode
R-Type Instructions			
1	AND	Reg(Rd) = Reg(Rs1) & Reg(Rs2)	1
2	ADD	Reg(Rd) = Reg(Rs1) + Reg(Rs2)	2
3	SUB	Reg(Rd) = Reg(Rs1) - Reg(Rs2)	3
4	SEQ	Reg(Rd) = 1 is Reg(Rs) = Reg(Rs2), zero otherwise	4
5	SLT	Reg(Rd) = 1 is Reg(Rs) < Reg(Rs2), zero otherwise	5
6	SGT	Reg(Rd) = 1 is Reg(Rs) > Reg(Rs2), zero otherwise	6
I-Type Instructions			
7	ADDI	Reg(Rd) = Reg(Rs1) + sign_extend (imm)	7
8	ANDI	Reg(Rd) = Reg(Rs1) & zero_extend (imm)	8
9	JZ	• Jump if Reg(Rd) equals zero • Target = Reg(RS1) + sign_extend (imm)	9
10	JNZ	• Jump if not zero • Jump if Reg(Rd) != zero • Target = Reg(RS1) + sign_extend (imm)	10
11	Jump	• Unconditional jump • Target = Reg(RS1) + sign_extend (imm) • Rd is ignored	11
M-Type Instructions			
12	LW	Reg(Rd) = Memory (Absolute Address)	0
13	SW	Memory (Absolute Address) = Reg(Rd)	0

5. RTL Design

You need to design and implement a single cycle processor that fully supports the aforementioned instructions. The processor design should include (1) complete datapath with clear diagrams, (2) control path, control signals truth table and Boolean functions.

6. Verification

To verify the RTL design, write a testbench around it. Moreover, you need to write multiple code sequences in the given ISA and show how the processor you designed executes these code sequences through simulation.

7. Project Report

The report document must contain sections highlighting the following:
Note: Having a complete datapath and control path design with clear description and clear block diagrams is a crucial prerequisite for the project discussion.

1 – Design and Implementation

1. Specify clearly the design giving detailed description of the data path, its components, control, and the implementation details (highlighting the design choices you made and why, and any notable features that your processor might have.)
2. Provide clear block diagrams of the design on both the unit level and the core level.
3. Provide a complete description of the control logic and the control signals. Provide a table giving the control signal values for each instruction. Provide the logic equations for each control signal.
4. Provide a list of sources for any parts of your design that are not entirely yours (if any).
5. Carry out the design and implementation with the following aspects in mind:
   - Correctness of the individual components
   - Correctness of the overall design when wiring the components together
   - Completeness: all instructions should be implemented properly.

2 – Simulation and Testing

1. Run simulation of the processor developed
2. Describe the test programs that you used to test your design with enough comments describing the program, its inputs, and its expected output. List all the instructions that were tested and work correctly. List all the instructions that do not run properly.
3. Provide snapshots of the simulator window with your test program loaded and showing the simulation output results.

3 – Teamwork

1. Work in groups of up to three students.
2. Group members are required to coordinate the work equally among themselves so that everyone is involved in all the following activities:
   - Design and Implementation
   - Simulation and Testing
3. Clearly show the work done by each group member.

8. Submission Guidelines

Attach one zip file containing all the design circuits, the programs source code and binary instruction files that you have used to test your design, their test data, as well as the report document to Ritaj.

9. Grading Criteria

Item	Weight
Control signals generation (truth tables and Boolean equations)	10
Processor Microarchitecture Design (complete datapath and control path) that supports all instructions	25
Complete **modular** RTL design of the above microarchitecture that supports all instructions	25
Verification environment (testbench) around the RTL design such that you can write code sequences in the ISA, store them in the instruction memory, execute them and show results using waveform diagrams.	25
Code organization and documentation	10
Detailed report that includes control signals truth tables, Boolean equations, detailed and clear microarchitecture design schematics, test cases, simulation screenshots, etc.	15
Discussion	10
Total	**120**