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

Project No. 2
ENCS4370| Computer Architecture

Deadline: July 2, 2023 at 23:59

---

1. Objectives:

To design and verify a simple RISC processor in Verilog

2. Processor Specifications

1. The instruction size is 32 bits
2. 32 32-bit general-purpose registers: from R0 to R31.
3. A special purpose register for the program counter (PC)
4. It has a stack called control stack which saves the return addresses
5. Stack pointer (SP), another special purpose register to point to the top of the control stack. SP holds the address of the empty element on the top of the stack. For simplicity, you can assume a separate on-chip memory for the stack, and the initial value of SP is zero.
6. Four instruction types (R-type, I-type, J-type, and S-type).
7. The processor's ALU has an output signal called "zero" signal, which is asserted when the result of the last ALU operation is zero.
8. Separate data and instructions memories

3. Instruction Types and Formats

As mentioned above, this ISA has four instruction formats, namely, R-type, I-type, J-type, and S-type. These four types have the following common fields:

a. 2-bit instruction type (00: R-Type, 01: J-Type, 10: I-type, 11: S-type)

b. **5-bit function,** to determine the specific operation of the instruction

c. **Stop bit,** which is the least significant bit of each instruction binary format, and it is used to mark the end of a function code block. In other words, if the value of this stop bit is "1", this means that this instruction is the last instruction of the function, and hence the execution control should return to the return address which is stored on the top of the control stack.

R-Type (Register Type) Format

- **5-bit Rs1:** first source register
- **5-bit Rd:** destination register
- **5-bit Rs2:** second source register
- **9-bit unused**

Function^5	Rs1^5	Rd^5	Rs2^5	Unused^9	Type^2	Stop^1

I-Type (Immediate Type) Format

- **5-bit Rs1:** first source register
- **5-bit Rd:** destination register
- **14-bit immediate:** unsigned for logic instructions, and signed otherwise.

Function^5	Rs1^5	Rd^5	Immediate^{14}	Type^2	Stop^1

J-Type (Jump Type) Format

- **24-bit signed immediate**: jump offset

Function^5	Signed Immediate^{24}	Type^2	Stop^1

S-Type (Shift Type) Format

- **5-bit Rs1:** first source register
- **5-bit Rd:** destination register
- **5-bit Rs2:** second source register. This register stores the shift amount in case the shift amount is variable and it is calculated at runtime
- **5-bit SA:** the constant shift amount.
- **4-bit unused**

Function^5	Rs1^5	Rd^5	Rs2^5	SA^5	Unused^4	Type^2	Stop^1

4. Instructions' Encoding

For simplicity, you are required to implement a subset only of this processor's ISA. The table below shows the different instructions you are required to implement. It shows their type, the function's value, and their meaning in RTN (Register Transfer Notation). Although the instruction set is reduced, it is still rich enough to write useful programs.

No.	Instr	Meaning	Function Value
R-Type Instructions			
1	AND	Reg(Rd) = Reg(Rs1) & Reg(Rs2)	00000
2	ADD	Reg(Rd) = Reg(Rs1) + Reg(Rs2)	00001
3	SUB	Reg(Rd) = Reg(Rs1) - Reg(Rs2)	00010
4	CMP	zero-signal = Reg(Rs) < Reg(Rs2)	00011
I-Type Instructions			
5	ANDI	Reg(Rd) = Reg(Rs1) & $Immediate^{14}$	00000
6	ADDI	Reg(Rd) = Reg(Rs1) + $Immediate^{14}$	00001
7	LW	Reg(Rd) = Mem(Reg(Rs1) + Imm^{14})	00010
8	SW	Mem(Reg(Rs1) + Imm^{14}) = Reg(Rd)	00011
9	BEQ	Branch if (Reg(Rs1) == Reg(Rd))	00100
J-Type Instructions			
10	J	PC = PC + $Immediate^{24}$	00000
11	JAL	PC = PC + $Immediate^{24}$ Stack.Push (PC + 4)	00001
S-Type Instructions			
12	SLL	Reg(Rd) = Reg(Rs1) << SA^5	00000
13	SLR	Reg(Rd) = Reg(Rs1) >> SA^5	00001
14	SLLV	Reg(Rd) = Reg(Rs1) << Reg(Rs2)	00010
15	SLRV	Reg(Rd) = Reg(Rs1) >> Reg(Rs2)	00011

5. RTL Design

You need to design and implement a datapath and control path that support all of the aforementioned instructions. You have three design options

1. **Single cycle** processor. In this case the project will be graded out of 80%
2. **Multi-cycle** processor with five stages: fetch, decode, execute, memory access, and write back.

In this case the project will be graded out of 100%

3. **5-stage pipelined** processor (fetch, decode, execute, memory access, and write back). In this case, the project will be graded out of 120%, i.e., there is a bonus that can reach 20% of the project grade.

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 executed these code sequences.

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 drawings of the component circuits and the overall data path.
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 were implemented properly.

2 – Simulation and Testing

1. Carry out the 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. Also, 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	30
Complete **modular** RTL design of the above microarchitecture that supports all instructions	35
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.	35
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.	20
Discussion	10
Total	**150**