NYCU-DCS-2024

HW1

Design: Central Processing Unit (CPU) - ALU & ID unit

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~DCSTA01/HW01.tar

Design Description

A Central processing unit (CPU) plays a crucial role in a computer. It is the primary component responsible for executing the instructions of a computer program. This includes performing arithmetic and logic operations, managing control, and handling I/O tasks.

In this homework, you have to design an ALU (Arithmetic Logic Unit) and an ID (Instruction Decoder) unit of a 16-bits CPU with 16 registers. Each instruction will carry out different operations based on ISA. Finally, the result of your operations will be stored in a certain memory location. The instruction format involved in this homework is shown below:

MSB	- Instruction Format - (16-bits)		LSB
opcode (3-bits)	rs (4-bits)	rt (4-bits)	immediate (5-bits)

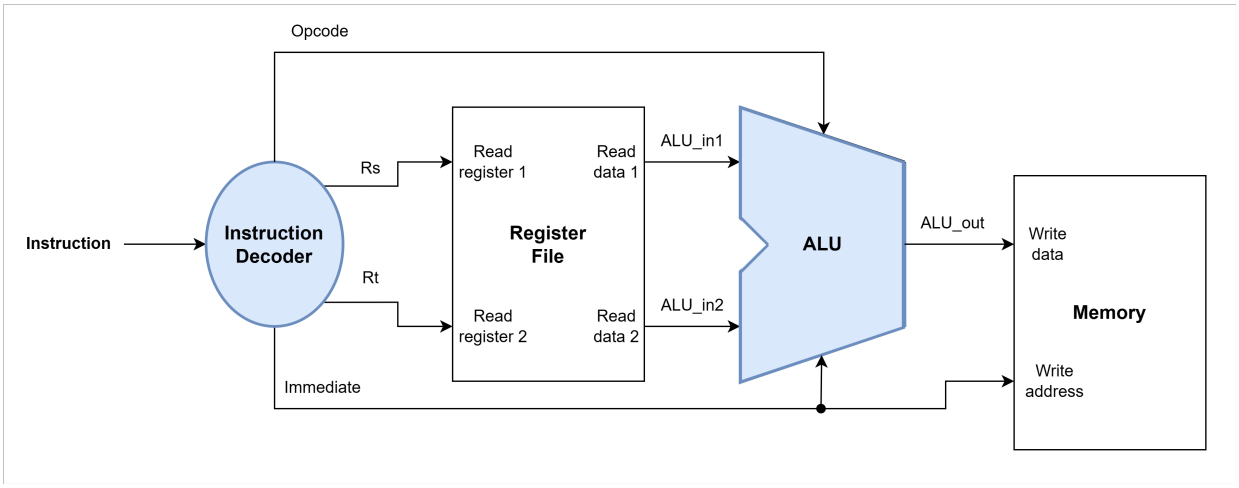
Rs and Rt signify the addresses of specific registers. With 4-bit register addresses, there are 16 available registers. Each of these registers can hold 16 bits of data. The instruction decoder (ID) unit must assign the relevant bits to the register file, enabling the arithmetic logic unit (ALU) to access the necessary data for its operations.

Immediate refers to a memory address. A 5-bit immediate value indicates that there are 32 distinct memory blocks available. In your architecture, the ID unit needs to assign the relevant bits to this memory, which allows the ALU to direct its results to the correct memory location.

Opcode indicates which function ALU will execute; details are as shown below:

Function Name	Operation	op code
Add	Out = R[rs] + R[rt]	3'b000
Mult	Out = R[rs] x R[rt]	3'b001
And	Out = R[rs] & R[rt]	3'b010
Inverse	Out = ~R[rs]	3'b011
Absolute Value	Out = | R[rs] |	3'b100
Minimum	Out = min(R[rs], R[rt])	3'b101
Left Shift	Out = R[rs] << unsigned(Imm)	3'b110
Addi	Out = R[rs] + sign extended(Imm)	3'b111

Block Diagram



Input

Signal	Number of bit	Description
Instruction	16 bits	After Instruction Decoder gets the Instruction, you should separate the Instruction into "opcode"、"rs"、"rt"、"immediate", then send the specific part of Instruction to Register, ALU, and Memory.

Output

Signal	Number of bit	Description
ALU_out	16 bits	ALU would receive data from Register File, then do the function corresponding to the opcode send by Controller. After ALU operation, you should save calculate result to the Memory, and we will check data in the Memory to verify your design.

Homework Upload

1. Use the command we provided to tar whole file into HW1 folder.
2. Upload the tar file and your report file to the new E3.
3. Your report file should be followed the naming rule: report_DCSxxx.pdf.
4. Upload your file before 3/28 (Thu) 15:00.

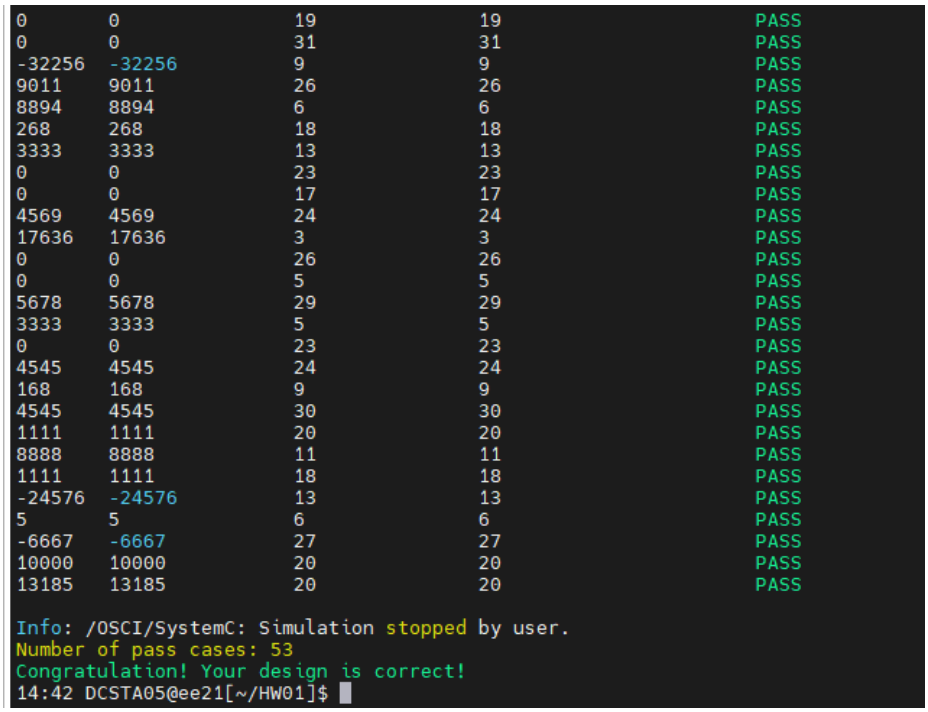
Grading Policy

1. Pass all the pattern we provided. 70%
2. Report 30%

Report

Your report must contain:

1. A Screenshot of output result, the figure must contain your server account information, as shown in the figure below.
2. The problem encountered in this homework and how the problem was solved.
3. Any suggestions about previous Labs and homework.



Command List

1. Compile & Run SystemC source code.
   ./01_systemc
2. Clean SystemC executable file.
   ./09_clean
3. Compress the files to be submitted.
   ./10_tar