

Computer Engineering and Software Systems Course Assessment Map and Specification

CESS3005: Computer Architecture Major Task: MIPS Design using VHDL

Introduction

This coursework is to be carried out in groups of 5 students. It is designed to meet the module Learning Outcomes as presented in the module proforma.

Project description:

1. A design of MIPS processor using VHDL that illustrates a basic computer system by simulating the data and control paths.
2. The project is done in **two phases**.
3. You should **present the project** to the TA, and be prepared for a 10 minutes discussion.
4. You should submit a fully detailed and described **project report** in soft copy.
5. This coursework is to be carried out in groups of **5 students**.

Phase I

Instructions:

1. Please reference appropriately any documents or URLs or books you use.
2. Assessment weight is 50%.
3. The due date of this Assignment is **29 April 2023**.
4. Only electronic submissions, through the e-learning web site, will be accepted.
5. Submissions are only accepted in form of one zip file containing the vhd files.
6. You are encouraged to ask the module leader and TA for any clarifications.
7. Late submissions will not be accepted.

Requirements:

1- Implement the MIPS register file that reads simultaneously from two registers and write into another. The implementation should follow the internal logic design of the register file:

- The main module should be called "RegisterFile"..
- The entity should look as follows:
 - **read_sel1** : in std_logic_vector(4 downto 0)
 - **read_sel2** : in std_logic_vector(4 downto 0)
 - **write_sel** : in std_logic_vector(4 downto 0)
 - **write_ena** : in std_logic
 - **clk**: in std_logic
 - **write_data**: in std_logic_vector(31 downto 0)
 - **data1**: out std_logic_vector(31 downto 0)
 - **data2**: out std_logic_vector(31 downto 0)

2- Modify the 32 bit full ALU.

- ALU functional specifications:

ALUOp	Function
0000	AND
0001	OR
0010	ADD
0110	SUB
1100	NOR

- Entity should look as follows:
 - **data1** : in std_logic_vector(31 downto 0)
 - **data2** : in std_logic_vector(31 downto 0)
 - **aluop** : in std_logic_vector(3 downto 0)
 - **dataout**: out std_logic_vector(31 downto 0)
 - **zflag**: out std_logic

3- Connect the already-built modules including register file, ALU, to design a simple MIPS CPU Using VHDL. The proposed CPU should be able to perform certain instructions: R-type (AND, OR, ADD, SUB, SLT and NOR).

- The datapath entity should look as follows:
 - **clk, reset**: in STD_LOGIC;
 - **instr**: in STD_LOGIC_VECTOR(31 downto 0);
 - **aluoperation**: in STD_LOGIC_VECTOR(2 downto 0);
 - **zero**: out STD_LOGIC;
 - **regwrite**: in STD_LOGIC
 - **aluout** : buffer STD_LOGIC_VECTOR(31 downto 0));

Marking:

The VHDL code and a report with the following:

- VHDL coding of blocks [40%]
- Simulation of blocks [40%]
- Pass all test cases [20%]

Phase II

Instructions:

1. Please reference appropriately any documents or URLs or books you use.
2. Assessment weight is 50%.
3. The due date of this Assignment is **13 May 2023**.
4. Only electronic submissions, through the e-learning web site, will be accepted.
5. Submissions are only accepted in form of one zip file containing the vhd files **for the whole task (phase I and II)**.
6. You are encouraged to ask the module leader and TA for any clarifications.
7. Late submissions will not be accepted.

Requirements:

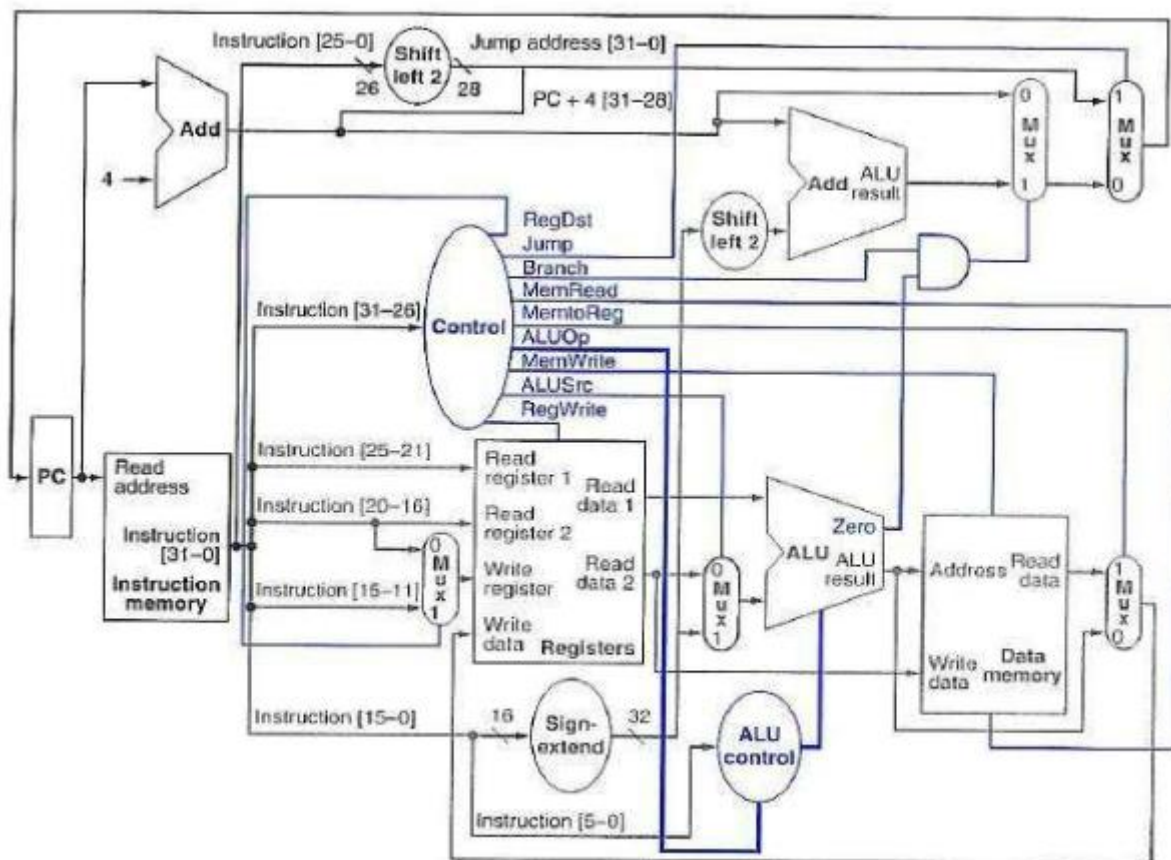
Modify the MIPS CPU as to be able to perform not only R instructions, but also I-type (lw, sw, beq) and J instruction.

Steps:

- Implement the **control** module, which is responsible for all of the control signals.
- Implement the **Mips** module by connecting the datapath with the control module .
- Connect the **Mips** module with instruction and data memory module together.

- Fill the memory module by a simple program.
- The CPU should be able to execute this program.
- Simulate the results and check the final results

The following diagram shows the abstract CPU design



Main_module Entity should look as follows:

- **CLK:** IN STD_LOGIC;
- **RST:** IN STD_LOGIC;
- **Writedata,dataadr:** OUT STD_LOGIC_VECTOR(31 downto 0);
- **memwrite:** OUT STD_LOGIC;

Test Case:

```
main:  addi $2, $0, 5 # initialize $2 = 5 0 20020005
      addi $3, $0, 12 # initialize $3 = 12 4 2003000c
      addi $7, $3, -9 # initialize $7 = 3 8 2067fff7
      or $4, $7, $2 # $4 = (3 OR 5) = 7 c 00e22025
      and $5, $3, $4 # $5 = (12 AND 7) = 4 10 00642824
      add $5, $5, $4 # $5 = 4 + 7 = 11 14 00a42820
      beq $5, $7, end # shouldn't be taken 18 10a7000a
      slt $4, $3, $4 # $4 = 12 < 7 = 0 1c 0064202a
      beq $4, $0, around # should be taken 20 10800001
      addi $5, $0, 0 # shouldn't happen 24 20050000
around: slt $4, $7, $2 # $4 = 3 < 5 = 1 28 00e2202a
      add $7, $4, $5 # $7 = 1 + 11 = 12 2c 00853820
      sub $7, $7, $2 # $7 = 12 - 5 = 7 30 00e23822
      sw $7, 68($3) # [80] = 7 34 ac670044
      lw $2, 80($0) # $2 = [80] = 7 38 8c020050
      j end # should be taken 3c 08000011
      addi $2, $0, 1 # shouldn't happen 40 20020001
end:    sw $2, 84($0) # write mem[84] = 7 44 ac020054
```

Test the MIPS processor.

add, sub, and, or, slt, addi, lw, sw, beq, j

If successful, it should write the value 7 to address 84

Marking:

The VHDL code and a report with the following:

- VHDL coding of blocks [40%]
- Simulation of blocks [40%]
- Pass all test cases [20%]

What you should hand in

Student should submit his/her work as a digital presentation which he/she will present in front of the TA and other students. In addition, student should submit **the project output as a VHDL code and a report.**

The Report should also include the following:

- A brief description of your implementation and design choices.
- The Data path you used including any necessary extensions you added to support all the required instructions.
- Sample output
- Any Assumptions you added that are not standard
- A section showing the contribution of each student in the project (Who did what exactly and the percentage of the overall effort every one made)

Evaluation Criteria

The following criteria may be updated according to the instructor evaluation criteria and the needed level of achievement.

89% and above:

Your work must be of outstanding quality and fully meet the requirements of the coursework specification and learning outcomes stated. You must show independent thinking and apply this to your work showing originality and consideration of key issues. There must be evidence of wider reading on the subject. In addition,

76% - 89%:

Your work must be of good quality and meet the requirements of the coursework specification and learning outcomes stated. You must demonstrate some originality in your work and show this by applying new learning to the key issues of the coursework. There must be evidence of wider reading on the subject. In addition

67% - 76%:

Your work must be comprehensive and meet all of the requirements stated by the coursework specification and learning outcomes. You must show a good understanding of the key concepts and be able to apply them to solve the problem set by the coursework. There must be enough depth to your work to provide evidence of wider reading.

60% - 67%:

Your work must be of a standard that meets the requirements stated by the coursework specification and learning outcomes. You must show a reasonable level of understanding of the key concepts and principles and you must have applied this knowledge to the coursework problem. There should be some evidence of wider reading. In addition

Below 60%:

Your work is of poor quality and does not meet the requirements stated by the coursework specification and learning outcomes. There is a lack of understanding of key concepts and knowledge and no evidence of wider reading.

Academic Misconduct

The University defines Academic Misconduct as 'any case of deliberate, premeditated cheating, collusion, plagiarism or falsification of information, in an attempt to deceive and gain an unfair advantage in assessment'. This includes attempting to gain marks as part of a team without making a contribution. The department takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy. If you are found guilty, you may be expelled from the University with no award.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you are unclear about what is required, please ask.

Exam ILOs according to UEL module specs

Knowledge

1. ~~Explain the concept of combinational and sequential logic circuits. (COI)~~
2. Illustrate how a data-path with or without pipelining is designed. (COI)
3. Infer the effect of the memory hierarchy on performance. (COI)

Thinking skills

4. ~~Simplify logic circuits using Boolean algebra and Karnaugh maps. (COI)~~
5. ~~Analyse and design combinational and sequential logic circuits. (COI)~~
6. Write MIPS assembly programs and translate C programs to assembly as well as assembly programs to machine code. (COI, DP, IC)

Subject-based practical skills

7. Design the digital computer components and processor's data path. (COI, DP, IC)

Skills for life and work (general skills)

8. Work and communicate effectively in team by effective collaboration and task management, working in a constrained stressful environment, and leading and motivating individuals. (SEI, PI, IC)

Course Assessment		2	3	6	7	8	Points	
Phase	I	•		•	•	•	10	
	II	•		•	•	•	10	
							20	