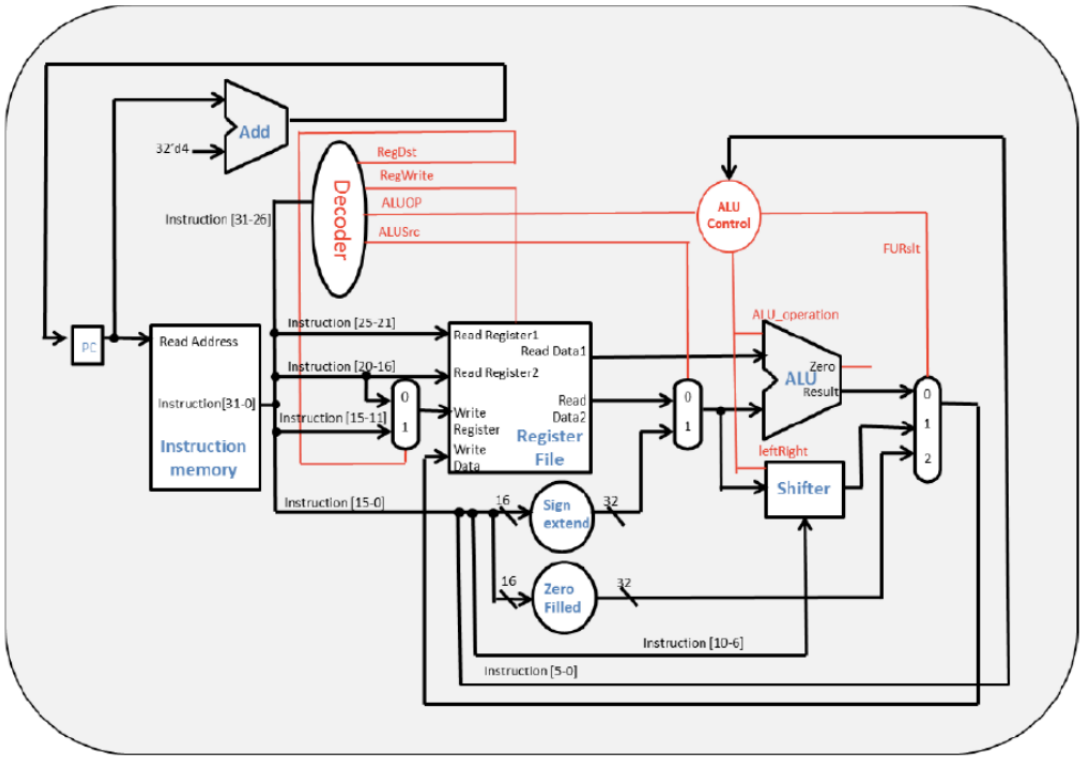
Computer Organization

Architecture diagrams:

3. Architecture Diagram



Hardware module analysis:

Program Counter:

將當前存取的位址送往 Instruction Memory 以讀取資料，並將數值加四更新下一次的 Program Counter。

Adder:

簡單的加法器，在本次作業中用來實現 PC = PC + 4。

Instr_Memory:

讀取當前位址對應的 machine code。

Mux2to1:

在本次作業中使用兩次 Mux2to1，第一次是在決定資料寫回的暫存器類型(rt 或

rd)，第二次是在決定送往 ALU 的運算元類型(rt 或是 sign extension)。

Reg_File:
讀取各暫存器儲存的資料，若資料需要寫回暫存器的話也在此寫回。

Decoder:
根據 machine code 決定多個 multiplexer 和 ALU Control 所需的 control signal。

ALU_Ctrl:
依照不同的指令設定 ALU_operation，以告訴 ALU 該執行什麼指令。若指令為 shift(sll 或 srl)，將 FURslt 設為 1。

Sign_Extend:
對 instruction 最後的 16 個 bits(instruction[15:0])做 sign extension。

Zero_Filled:
對 instruction 最後的 16 個 bits 前面填 0。

ALU:
根據 ALU_operation，執行 or、and、add、sub、slt 等各種運算。

Shifter:
根據 shamt 的數值將輸入左移/右移並輸出結果。

Mux3to1:
在本次作業中，用來決定寫回 register 的結果來自 ALU、shifter 或 zero_filled。

Finished part:

本次作業要求全部完成。

Problems you met and solutions:

1. ALU_operation 不知為何要 3 個 bit

解: 其實到最後也只用到 2 個 bit，首位隨便填就好。

2. ALU Control 在作業圖上有三條輸出，sample code 卻只有兩條 output wire

解: 因為 sll, srl 兩個指令不經過 ALU，所以他們的 ALU_operation 是 don't care bits，可以自行設定 ALU_operation 的值用來當作 leftRight 的判斷。

3. Simple_Single_CPU.v 沒有任何/*your code here*/的註解

解:看完 E3 討論區，發現要自己接線，接完就可以執行了。

Summary:

本次作業實作一個功能尚未齊全的 single-cycle CPU。