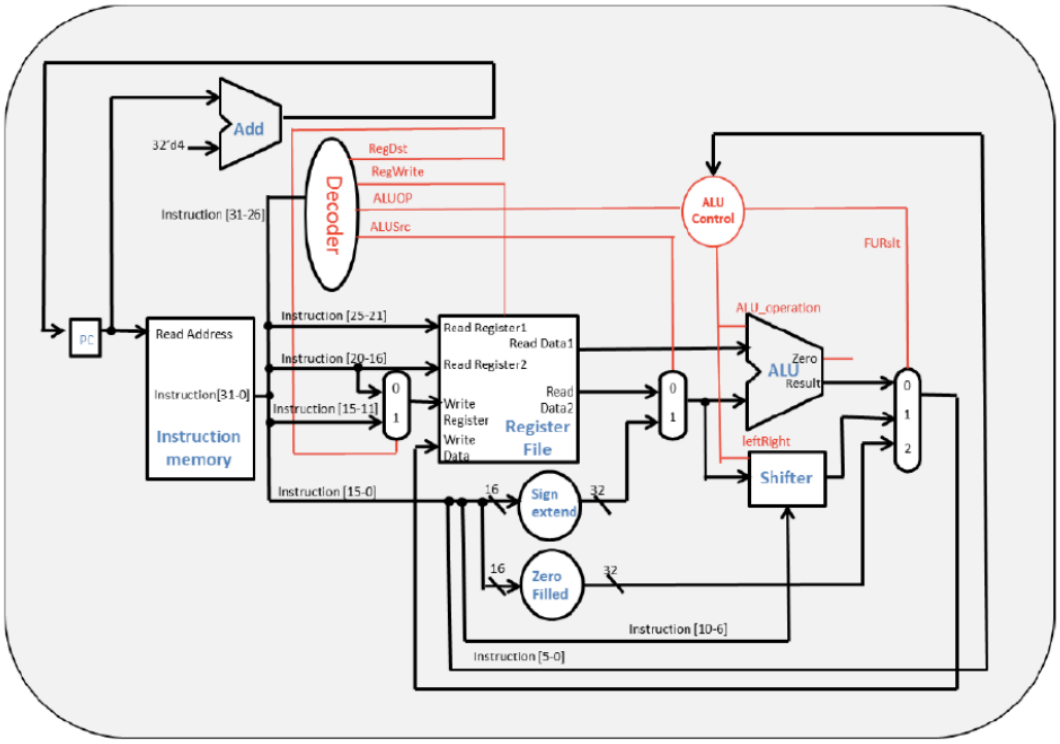
Computer Organization

110550126 曾家祐

Architecture diagrams:



Hardware module analysis:

- Adder.v：實作 PC = PC+4
- ALU.v：from lab 2 依照指令進行運算
- ALU_Ctrl.v：根據 function code 和 ALUOP code 輸出 ALU_operation (leftright)
- Decoder.v：根據 instructio[31:26]輸出 RegDst,RegWrite,ALUOP,ALUSrc
- Instr_Memory.v：讀取指令
- Mux2to1.v：依照輸入的訊號選擇輸出
- Mux3to1.v：依照輸入的訊號選擇輸出
- Program_Counter.v：控制進入下一指令的時間
- Reg_File.v：對 register 的輸入輸出
- Shifter.v：運算 srl sll
- Sign_Extend.v：將 16bits 變成 32bit

- Zero_Filled.v：將 16bits 變成 32bit
- Simple_Single_CPU.v：將前面的所有串接起來

Finished part:

- Adder.v：單純相加兩個輸入
- ALU_Ctrl.v：先檢查是不是 addi 指令(ALUOP_I = 000)
    - 如果不是再依照 function code 來輸出 ALU_operation (leftright)和 FURslt
- Decoder.v：根據 instructio[31:26]輸出 RegDst,RegWrite,ALUOP,ALUSrc
    - (此 lab 只區分 R format 和 ADDI 指令)
- Instr_Memory.v：from TA
- Mux2to1.v：依照輸入的訊號選擇輸出
- Mux3to1.v：依照輸入的訊號選擇輸出
- Program_Counter.v：from TA
- Reg_File.v：from TA
- Shifter.v：from lab2
- Sign_Extend.v：將 16bits 依照正負(正:0 負 1)填完左邊 32bits
- Zero_Filled.v：將 16bits 左邊全部填 0 成 32bits
- Simple_Single_CPU.v：依照 diagram 將所有 module 連起來

Problems you met and solutions:

一開始不知道要從哪邊下手，後來注意到了 architecture diagram 後，知道了這次整個的 lab 目標，但還是有些不清楚元件的實作目標再搭配講義內容去了解每個元件的功能、輸入、輸出，之後就知道為什麼要有這些輸入，和輸出原因了。

最後再依照 diagram 將全部串接起來，也在這個過程將前面的 bug 順便修復。

Summary:

這次的 lab 是要完成一個 single cycle cpu，除了 lab2 的 ALU 外，還做了許多額外的小元件。在成功把每個元件做出來之後再依照 architecture diagram 將完成的元件連起來，讓我更了解整個 cpu 在不同指令時會經過的流程。