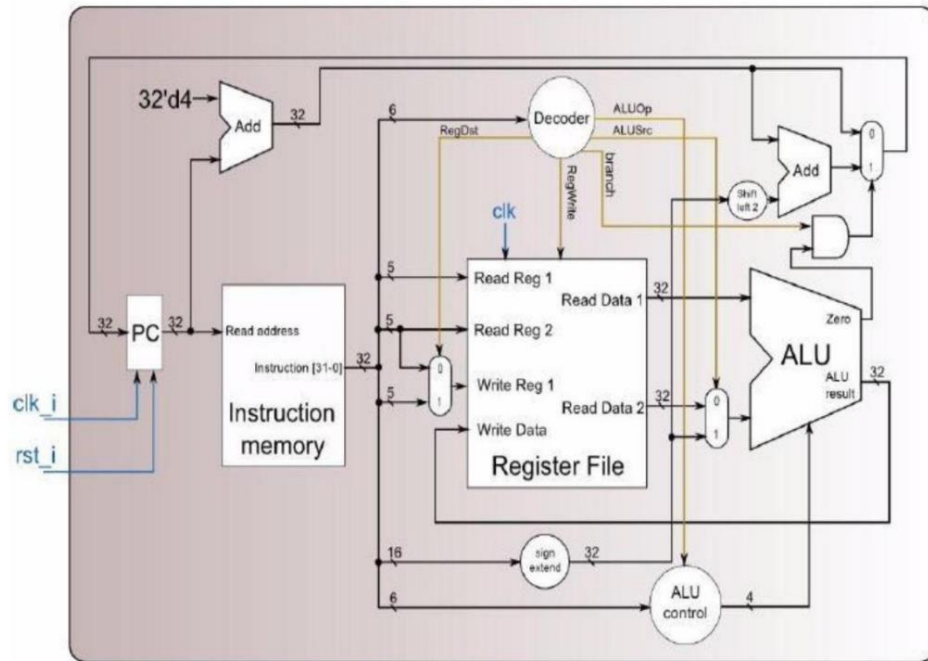


Computer Organization

Architecture diagrams:



Hardware module analysis:

Simple_Single_CPU: 整個 CPU 的架構，將 modules 用對應的線連接起來。

Adder: 兩個輸入等於輸出，用來算地址

MUX_2to1: 依照輸入的 select 從兩個 data 中選擇一個輸出。

Decoder: 把得到的訊息分成 ALUOp, ALUSrc, branch, RegWrite, RegDst

ALU_Ctrl: 把輸入的訊息換成 ALU 要做的運算方式

ALU: 依照 ALU_Ctrl 的輸出做運算

Sign_Extend: 將 16bits 變成 32bits

Shift_Left_Two_32: 往左一兩個 bits

Finished part:

case1

r0=	0
r1=	10
r2=	4
r3=	0
r4=	0
r5=	6
r6=	0
r7=	0
r8=	0
r9=	0
r10=	0
r11=	0
r12=	0

case2:

r0=	0
r1=	1
r2=	0
r3=	0
r4=	0
r5=	0
r6=	0
r7=	14
r8=	0
r9=	15
r10=	0
r11=	0
r12=	0

Problems you met and solutions:

輸出與預期不符:經過檢查後，發現問題出在 ALU_Ctrl。把 test case 改成少一點的指令，檢查輸出，並對照 CO_Lab_2. pdf 給的表格，發現 beq 的的狀況下有誤，並修改一些細節，最後終於輸出預期的答案。

Summary:

這次的 lab 較上次更加複雜，花了許多時間思考。這次 lab 讓我比較了解 CPU 的運行方式和結構。