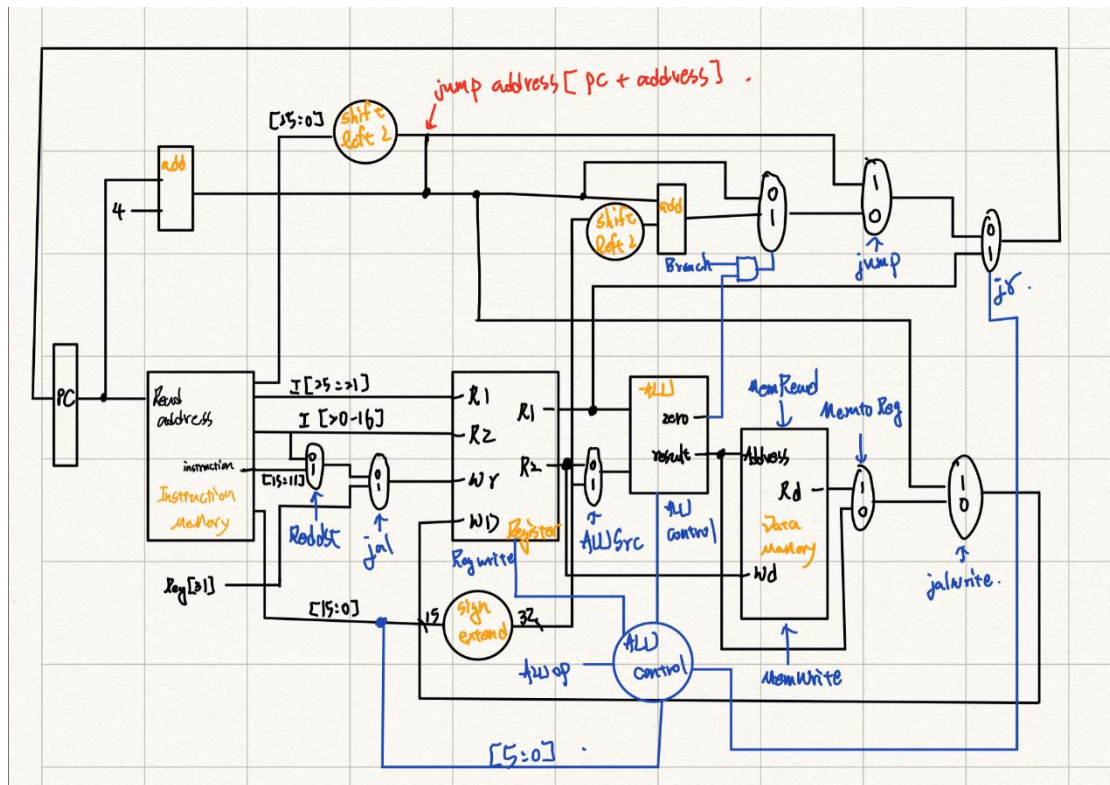


Architecture Diagram



Hardware Module Analysis

1. decoder: 是利用 instruction[31:26] 的 mintern 來做 and or structure 控制 control signal
2. ALU_Control: 是利用 instruction[5:0] 和 ALUOp 來控制但是這次不同於 lab2 有多一些 signal 需要放到 level 2 才能產生正確的輸出，例如: RegWrite, jr
3. 其餘的 multiplexer 和 ISA 和 Regfile 和 data memory 都照著 control signal 和 instruction 的要求接上而已

Result:

1. test1:

```

Data Memory =      1,      2,      0,      0,      0,      0,      0,      0
Data Memory =      0,      0,      0,      0,      0,      0,      0,      0
Data Memory =      0,      0,      0,      0,      0,      0,      0,      0
Data Memory =      0,      0,      0,      0,      0,      0,      0,      0
Registers
R0 =      0, R1 =      1, R2 =      2, R3 =      3, R4 =      4, R5 =      5, R6 =      1, R7 =      2
R8 =      4, R9 =      2, R10 =      0, R11 =      0, R12 =      0, R13 =      0, R14 =      0, R15 =      0
R16 =      0, R17 =      0, R18 =      0, R19 =      0, R20 =      0, R21 =      0, R22 =      0, R23 =      0
R24 =      0, R25 =      0, R26 =      0, R27 =      0, R28 =      0, R29 =      128, R30 =      0, R31 =      0
    
```

2. test2:

```
Data Memory =      0,      0,      0,      0,      0,      0,      0,      0
Data Memory =      0,      0,      0,      0,      0,      0,      0,      0
Data Memory =      0,      0,      0,      0,      68,      2,      1,      68
Data Memory =      2,      1,      68,      4,      3,      16,      0,      0

Registers
R0 =      0, R1 =      0, R2 =      5, R3 =      0, R4 =      0, R5 =      0, R6 =      0, R7 =      0
R8 =      0, R9 =      1, R10 =      0, R11 =      0, R12 =      0, R13 =      0, R14 =      0, R15 =      0
R16 =      0, R17 =      0, R18 =      0, R19 =      0, R20 =      0, R21 =      0, R22 =      0, R23 =      0
R24 =      0, R25 =      0, R26 =      0, R27 =      0, R28 =      0, R29 =      0, R30 =      0, R31 =      0
```

Summary:

這次的 lab 其實和上一次的差不多，但這次我遇到的最大困難是對於 ALU_Control 的畫簡，因為這次的 instruction 比較多在畫簡時很常有畫簡錯誤的問題，我的解決方法是利用 display 將 ALU_Control 的 output 都印出來檢查在相對應的 instruction 下所輸出的 output 是否正確