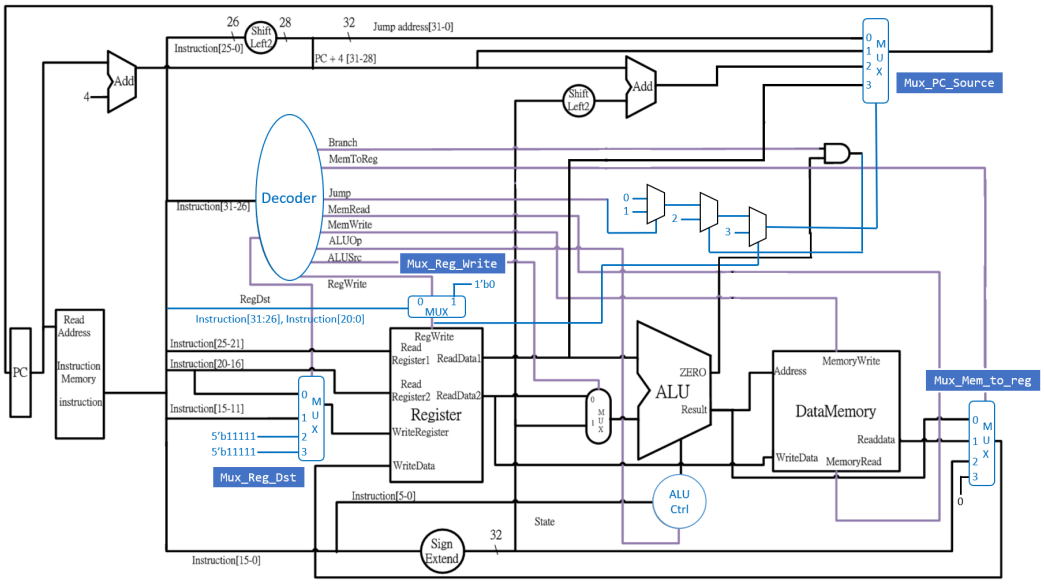
❖ Architecture Diagram



❖ Hardware Module Analysis

解釋和 Lab 2 不同的藍色部分

⌦ Decoder

新增了 Jump、MemRead、MemWrite 這三個控制訊號

⌦ Mux_Reg_Write

因為 Jr 的 Instruction[31:26]和 R-type 一樣，Decoder 輸出的 RegWrite 會是 1，所以再用 Instruction[31:26] == 6'b000000 && Instruction[20:0] == 8 判斷是否為 jr，是的話輸出的 RegWrite_selected 選 1’b0，不是的話選原本的 RegWrite。

⌦ Mux_Reg_Dst

lw 是 0，R-type 是 1，Jal 是 2，3 是隨便塞的。

⌦ Mux_PC_Source

0 是 Jump 和 Jal，1 是 PC+4，2 是 Branch，3 是 Jr (拉 RS_Data)。控制訊號是由一堆 if else 產生的

```
assign pc_select_r = (regWrite_select_r == 1'b1) ? 3:
                     (bz_w == 1) ? 2:
                     (Jump_w == 1) ? 0:
                     1;
```

⌦ Mux_Mem_to_reg

0 是 ALU result，1 是 MemData，2 是 PC+4 (jal)，3 隨便塞 0

⌦ ALU Ctrl

要新增 ALU op 是 3’b000 的時候 ALU control 是 4’b0010 (add)

❖ Result

Test 1

```
PC =         44
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
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 16050000 ticks.
```

Test 2

```
PC =         24
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,         56,          4,          0,         16,          0,          0
Registers
R0 =          0, R1 =          0, R2 =          5, R3 =          0, R4 =          2, R5 =          0, R6 =          0, R7 =          0
R8 =          0, R9 =          1, R10 =         0, R11 =         0, R12 =         0, R13 =         0, R14 =         0, R15 =         0
R16 =         3, 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 =        92, R30 =         0, R31 =        56
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 16050000 ticks.
```

❖ Summary

這次發現架構要自己想，有點慌張，還好有同學的幫忙，才慢慢知道該怎麼做，jump、jal、jr 一開始還搞不太懂怎麼運作。還遇到了取名的問題，Reg_File 一定要叫 Registers，Data_Memory 一定要叫 Data_Memory，不然 testbench 跑不出來。Test 2 還以為全部的質都要一樣，找 bug 找很久，後來才發現只要 R2 = 5 就可以了，還浪費時間找 bug ~~~，不過總算是好好地完成了。