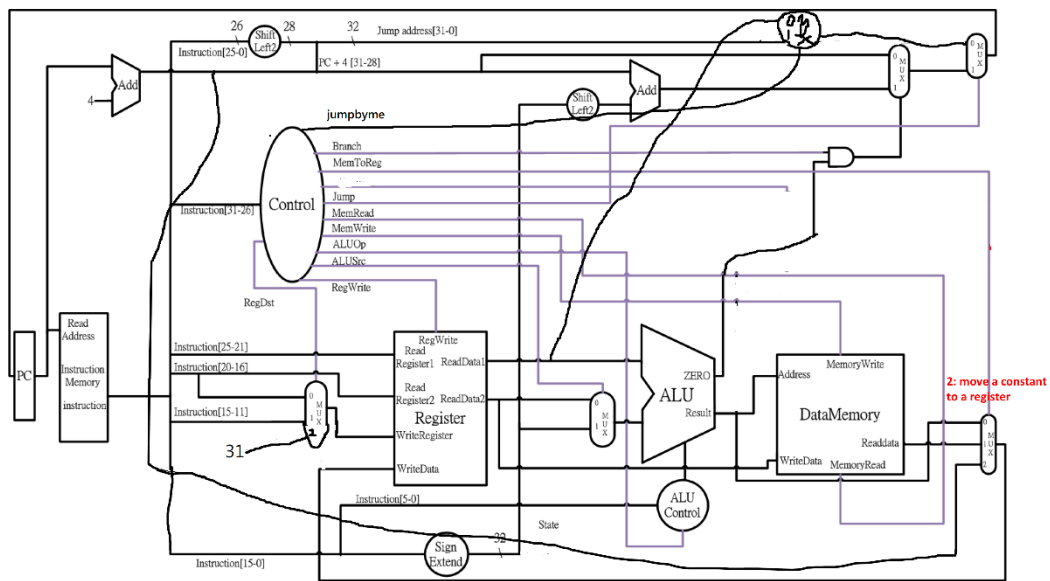


Architecture Diagram:



Hardware Module Analysis:

這次主要改動的是下列四個模組

ALU_Ctrl:與 lab2 相比因為新增了一些指令，因此指令的判斷是也要重新再改動，像是判斷是否為 **R-type** 的指令就變得更複雜了一點。

Decoder:新增了許多 control signal，像是 memwrite memread 等等的指令，以前的也因為新指令的加入需要修改判斷的條件。

Simple_Single_CPU:因為新增的像 **jump** 指令所以多了很多的線路，舊有的線路也有重接。

MUX_3to1:此次 lab 的重要腳色，在許多地方都有用上像是 jump 的選擇上。

Result:

```
命令提示字元 - vvp a
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
PC = x
ALUctrl = 0xxx
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
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
PC = x
ALUctrl = 0xxx
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
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
PC = x
ALUctrl = 0xxx
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
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
PC = x
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 16050000 ticks.
-
```

上面展示的是 **test1** 的結果，可以看到結果與正確解答一樣。

Summary:

這次的 lab 是上次的 lab 的進階版，新增了許多上課教過的指令，像是 **jr jal j** 三兄弟，這次主要的實作就是在他們三個上面，而過程中不意外的又是有非常多的 **bug**，這次最難 **de** 的 **bug** 就是 **decoder** 和 **alucrtl** 的判斷式一直 **derive** 錯，還有接線的部分也是因為多了那三個 **jump** 所以也有一些小地方沒想清楚接錯了，最後才好不容易讓 **cpu** 正常的跑起來，十分的感動，但這次因為整體的架構並沒有很大的改變，所以跟上次相比是沒有寫那麼久的，**Bug** 也因為只有改三個模組範圍縮小了好 **de** 很多，總體而言我覺得這次的 lab 讓我實作了一些上課教過的指令，有一種學以致用的感覺，我覺得是滿好的一次練習！