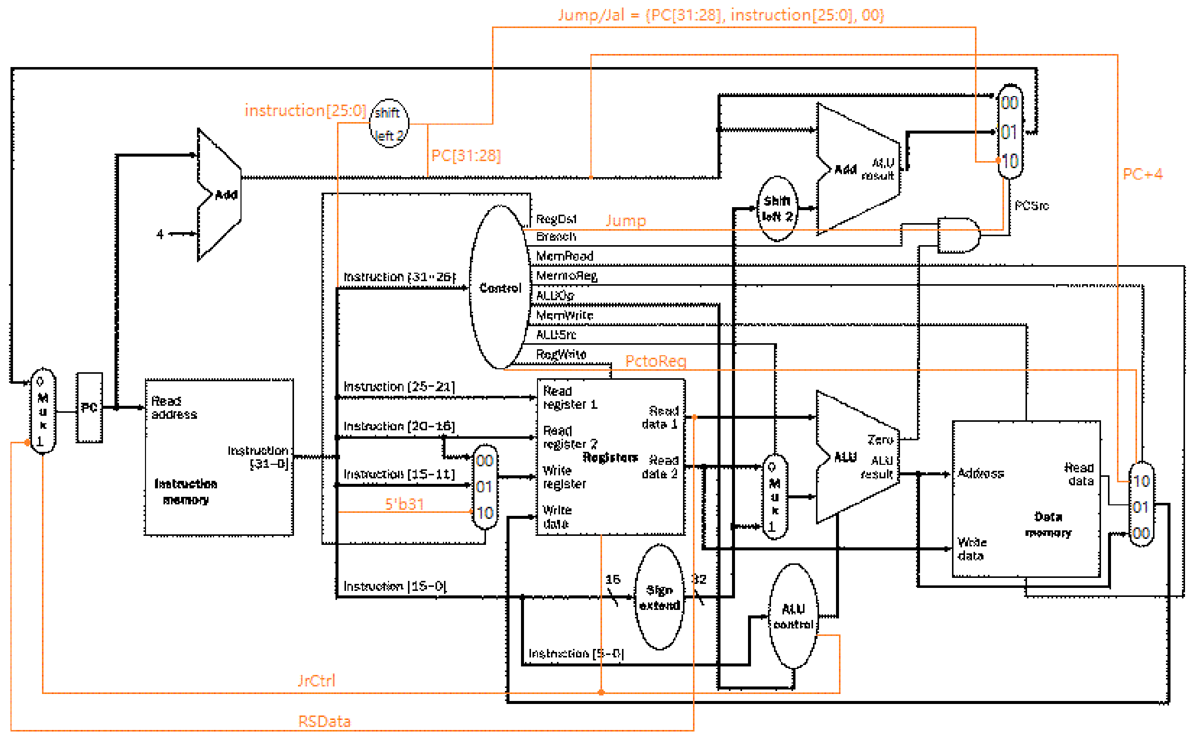
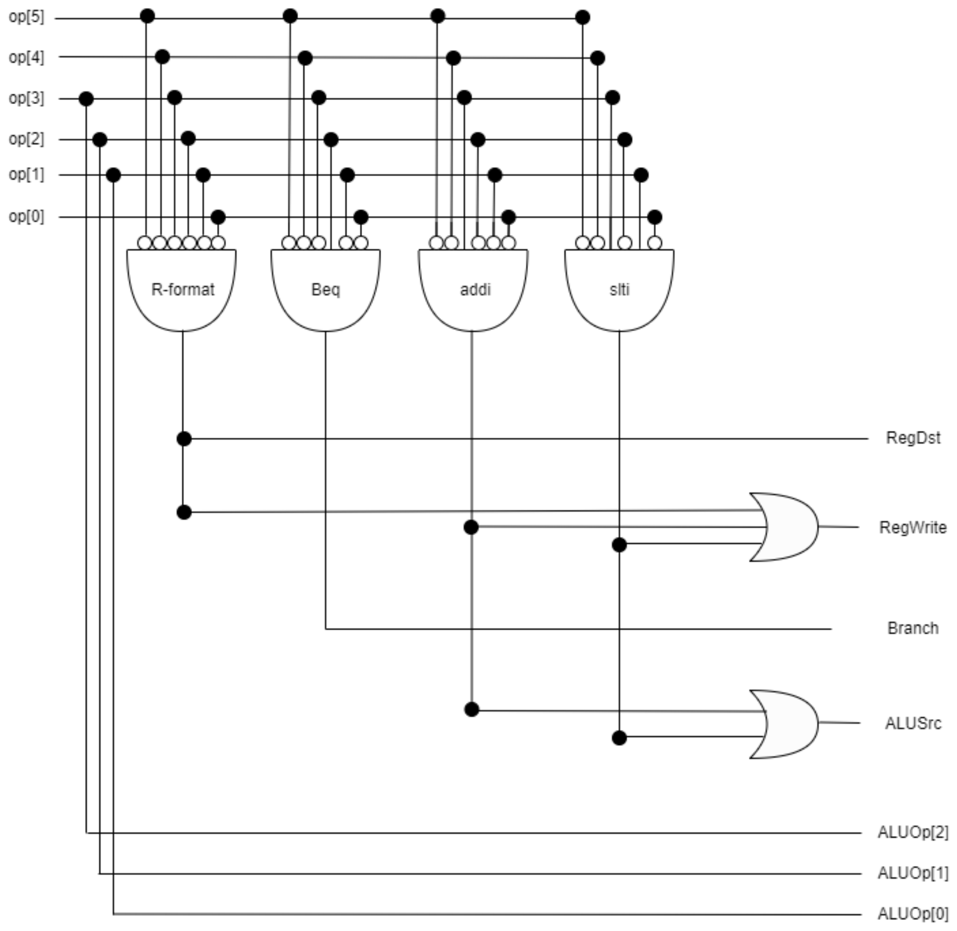
Computer Organization Lab3

Architecture diagrams:



Decoder 概念:



Hardware module analysis:

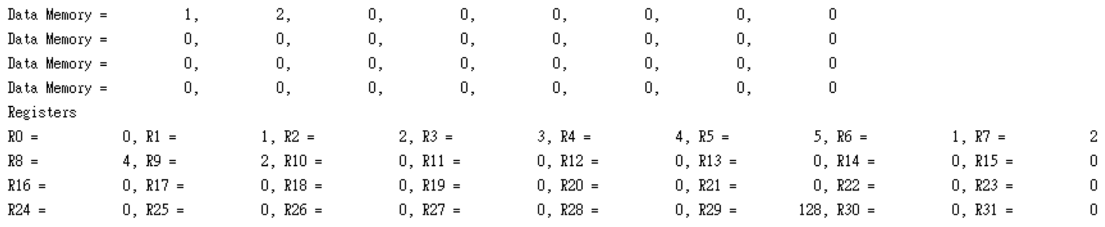
沿用上一次的 single cycle CPU 的設計，並新增了 Jump、PctoReg、JrCtrl 三個 control signal，剩下並無做太多更動

原本要新增成 2 個 2to1 MUX 的地方，改為 3to1 MUX，設計簡單、節省資源

Decoder 沿用上一次的方式，附圖是 lab2 的，概念都是將不同種類的指令分開，設計上較簡單也較不會有錯

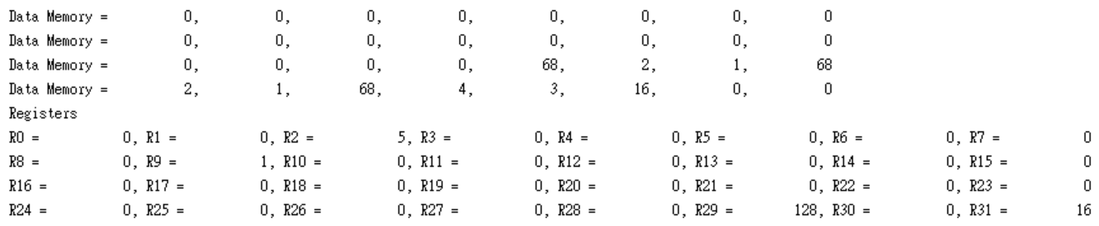
Finished part:

Test1: 和助教提供的結果一致



```
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
```

Test2: 測資輸入的是 f(4)=5 而 R2=5 兩者一致，且 S0=R16=0、S1=R17= 0 也是正確的



```
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 =       128, R30 =         0, R31 =        16
```

Problems you met and solutions:

我遇到一個問題是 JrCtrl 的控制訊號設定有誤
(原本設定 JrCtrl = R-format & ~funct_i[5]，因為 R-format 的指令當中，只有 Jr 的 funct[5]=0)，導致到 test2.txt 跑到最後 32’b0 之後，又跳回 PC=0 的指令，從頭再跑一次
後來改為 JrCtrl=R-format & ~funct[5] & funct[3] 才正常運作

Summary:

本次的 lab 大多可以參考上一次的 lab，但還是有許多地方需要自己仔細思考、設計，這個部分還需要多多練習。