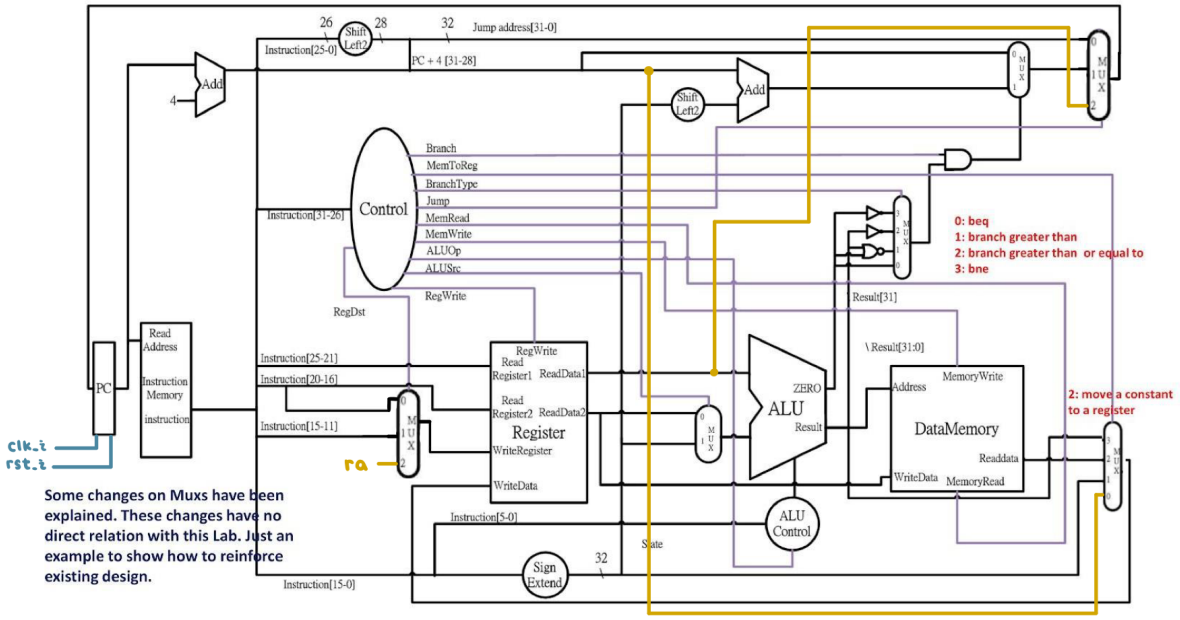
Computer Organization Lab3

Name: 莊婕妤　ID: 109550182

Architecture Diagram



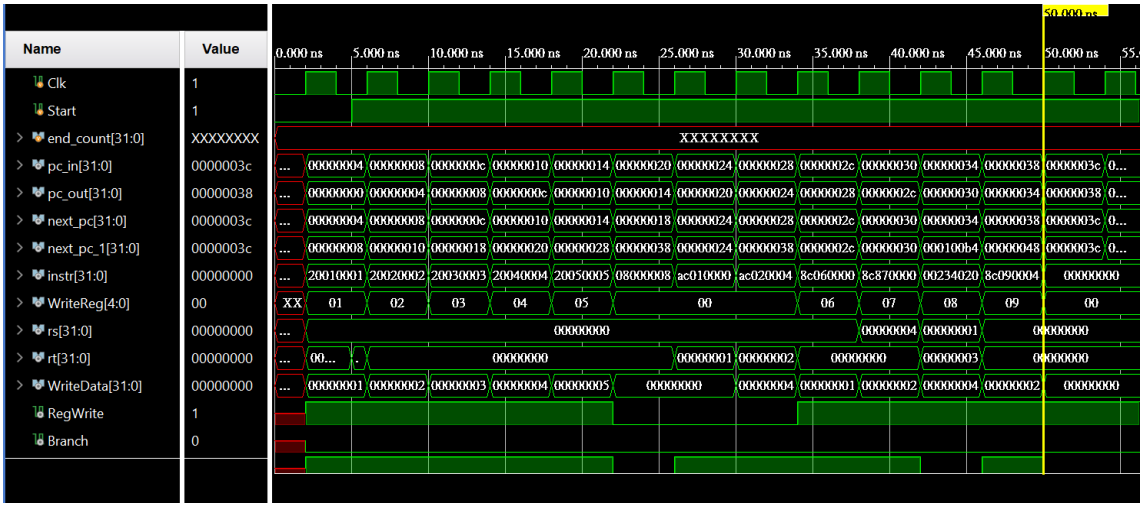
Hardware Module Analysis

這次的 Lab 因為要多實作 jal、jr 兩個比較特別的指令，所以我多拉了上圖中的三條黃色的線當做各自的 MUX 的 input，像是多拉了 ReadData1 到 PC source MUX 讓 jr 可以跳到指定的 register，以及多拉了 ra 到 Register File MUX 和 PC + 4 到 Memory to Register MUX 讓 jal 較好實作。

為了處理更多樣的指令，我也加了很多的 control signal 給我的 decoder，也多設計了 3-to-1 MUX 跟 4-to-1 MUX 的 module。

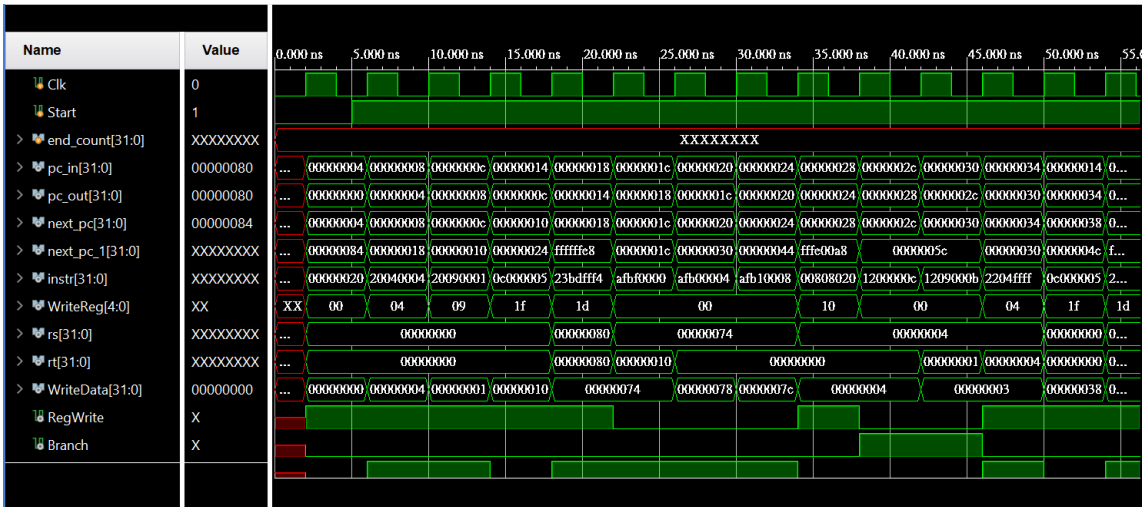
Finished Part

Test Data 1




```
PC =          128
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 =          0, R30 =        128, R31 =          0
```

Test Data 2




```
PC =          128
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 =        128, R31 =         16
```

Finished Part Explanation

上面 Test Data 1 & Test Data 2 波形圖跑出的結果跟我自己 trace 過整個 assembly code 的結果一模一樣，因此我認為我的電路圖實作是正確的。

Problems You Met and Solutions

在實作的過程中，我遇到的問題主要是不太熟悉 Jump、Jump and Link、Jump Register 是如何運作的，以及他們相對應的 control signals 該如何去設定。但在理解他們的 Instructions，並一條線一條線 trace 過整個電路，讓我在設計 decoder 時更加順手。

Summary

這次的 Lab 讓我更了解一個能夠處理如此多種指令的 CPU 內部長什麼樣子，透過各個 Module 的設計，再整個接起來合成一個 Simple Single CPU，幫助我更熟悉他們背後運作的原理，像是如何運用 control signals 讓其可以執行許多不一樣的 Instructions。經過此次作業後，我對於整個 CPU 的觀念都更加清楚了。