Computer Organization
Lab5: Advanced Pipelined CPU

Name: 蔡師睿

Student ID: 110550093

Architecture Diagrams

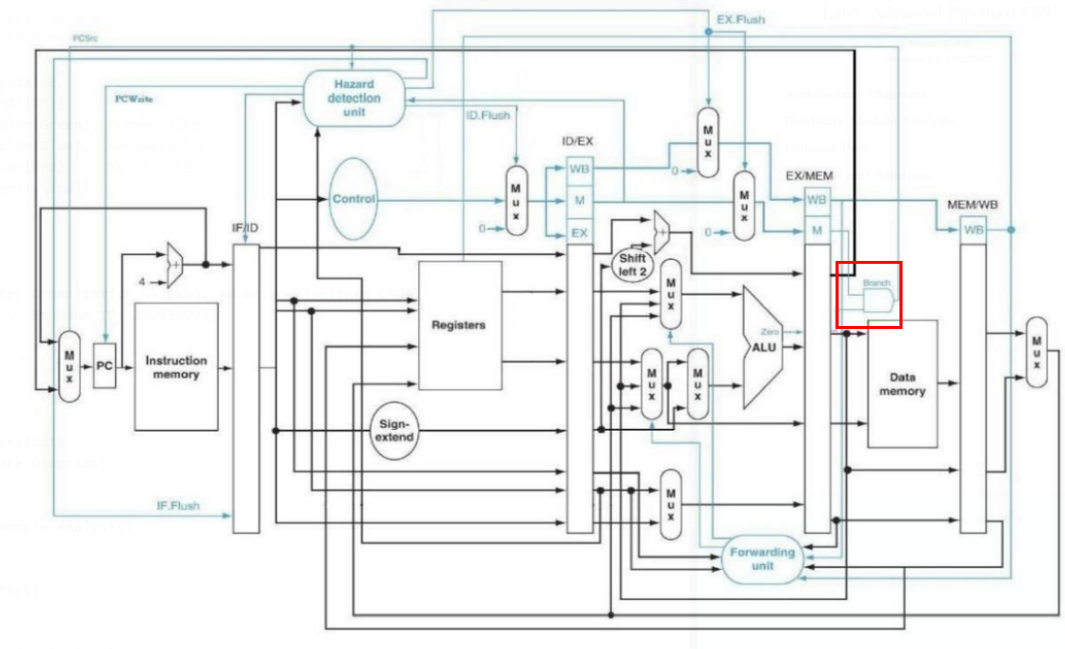


Figure 1

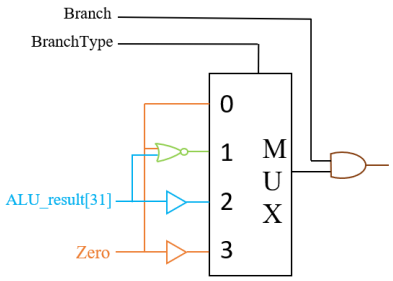


Figure 2

※ 將 Figure 1 中紅框的部分改成 Figure 2 的結構。

Hardware Module Analysis

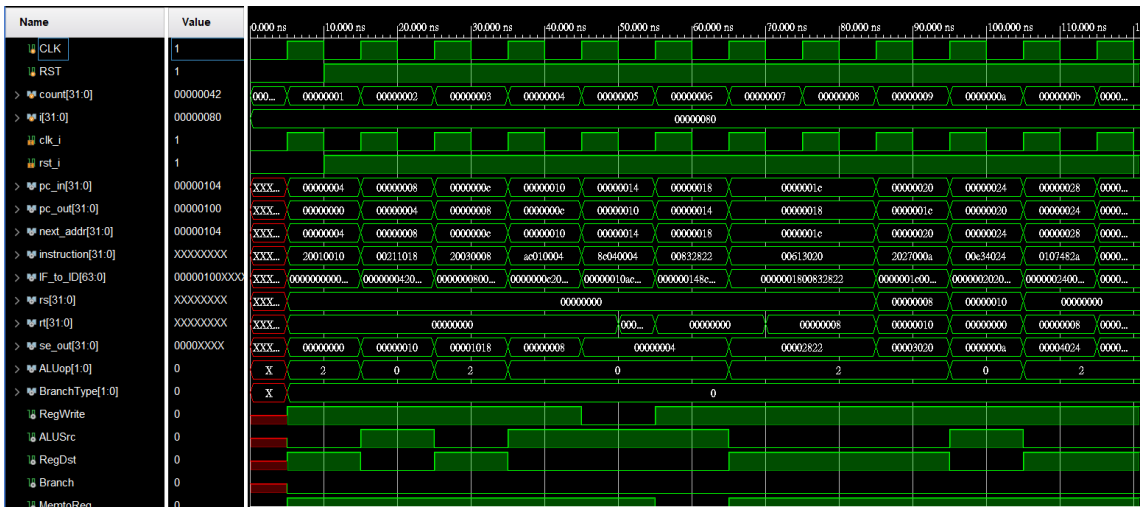
處理不同 branch 指令的方式，我參考了 Lab3 的實作，加入 2 bits 的 BranchType 幫助判斷是 branch 的哪一種指令，並使用 MUX4x1 選擇，MUX4x1 選擇的輸出跟 Branch 做 AND 再拉回去。其餘便都照著 figure 1 的方式接線實作。
Pipelined CPU 加入 forwarding 和 hazard detection 之後，我們的 CPU 便可以根據進來的指令去做 stall 或 forwarding，因此 pipeline 時不用再避免 hazard 的情況也不需要調換 machine code。

Finished Part

Test 1

```
====================================Register====================================
r0 =   0, r1 =  16, r2 = 256, r3 =   8, r4 =  16, r5 =   8, r6 =  24, r7 =  26

r8 =   8, 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

====================================Memory====================================
m0 =   0, m1 =  16, m2 =   0, m3 =   0, m4 =   0, m5 =   0, m6 =   0, m7 =   0

m8 =   0, m9 =   0, m10=   0, m11=   0, m12=   0, m13=   0, m14=   0, m15=   0

m16=   0, m17=   0, m18=   0, m19=   0, m20=   0, m21=   0, m22=   0, m23=   0

m24=   0, m25=   0, m26=   0, m27=   0, m28=   0, m29=   0, m30=   0, m31=   0
```

simulation result of Test 1

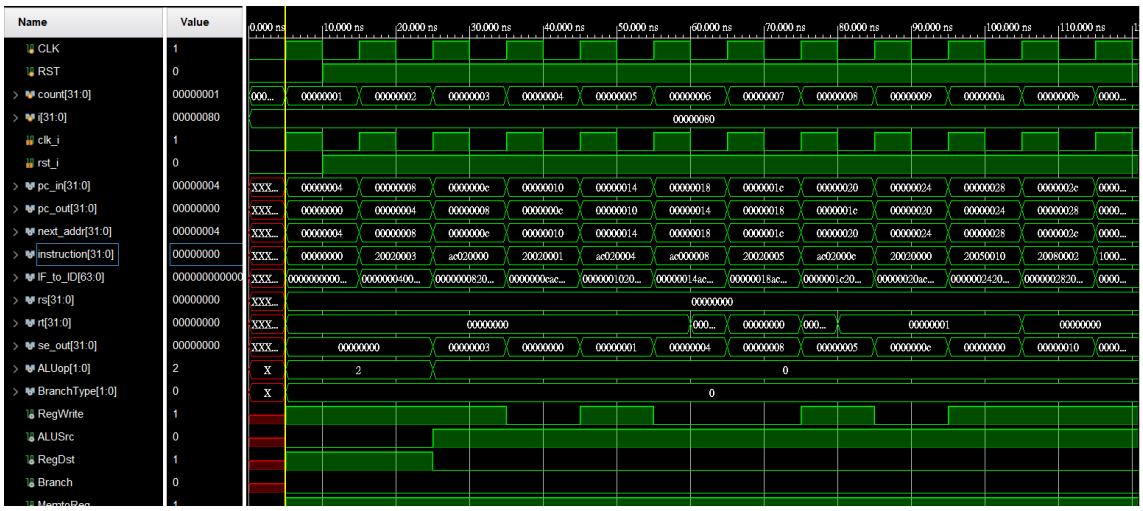


Waveform of Test 1

Test 2

```
=======================================Register=======================================
r0 =    0, r1 =    0, r2 =   16, r3 =    6, r4 =    0, r5 =   16, r6 =    0, r7 =    0

r8 =    2, r9 =    0, 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

=======================================Memory=======================================
m0 =    4, m1 =    1, m2 =    0, m3 =    6, m4 =    0, m5 =    0, m6 =    0, m7 =    0

m8 =    0, m9 =    0, m10=    0, m11=    0, m12=    0, m13=    0, m14=    0, m15=    0

m16=    0, m17=    0, m18=    0, m19=    0, m20=    0, m21=    0, m22=    0, m23=    0

m24=    0, m25=    0, m26=    0, m27=    0, m28=    0, m29=    0, m30=    0, m31=    0
```

Simulation result of Test 2



Waveform of Test 2

Test 1 和 Test 2 的結果與預期的結果一致，故 hazard detection 與 forwarding 實作無誤。

Problems and Solutions

1. 因爲直接拿 Lab4 的 Pipelined CPU 來改，但 testbanch 的 module name 與上次不同，因此 Pipelined_CPU 要改成 Pipe_CPU_1。
2. Simulation Test 2 時，發現 branch 相關的指令無法發揮作用，即使條件成立，依然會繼續執行下一行，debug 許久後才發現在 Hazard_Detection.v 中，寫成 flush 在所有條件下皆爲 1'b0，重設條件後才輸出正確的結果。

Summary

本次的 Pipelined CPU 因爲 hazard 和 forwarding 的關係，結構更爲複雜，尤其是接線時，更是消耗了不少時間，而且 debug 更爲辛苦了，只能從 simulation result 或 waveform 慢慢推算哪個環節出現問題。