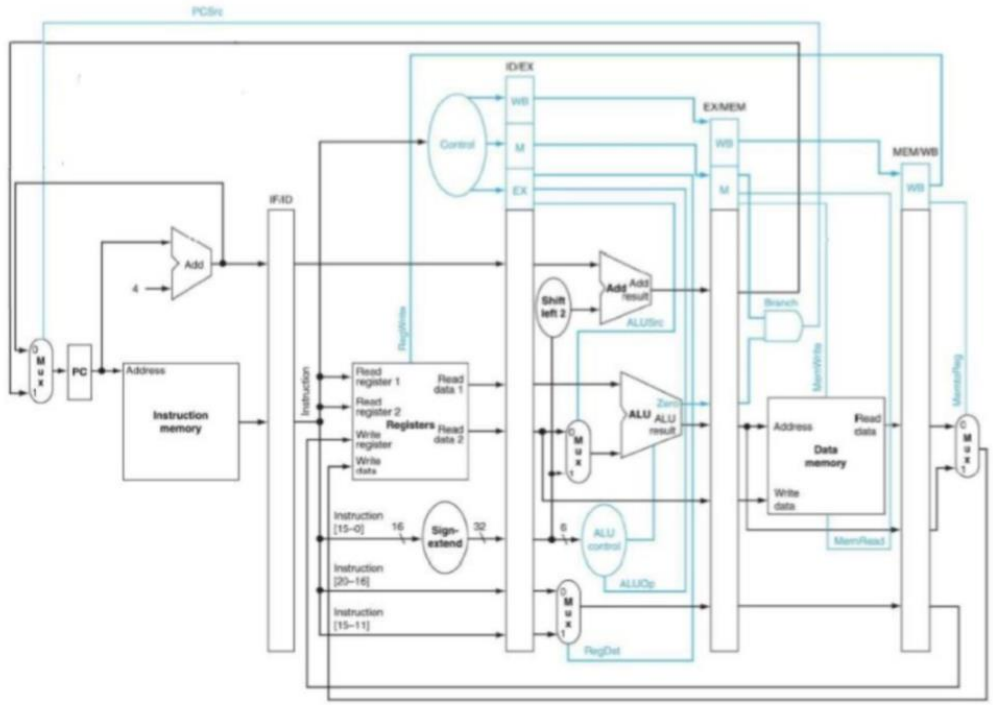
CO_Lab4 Report

0511105 李頤

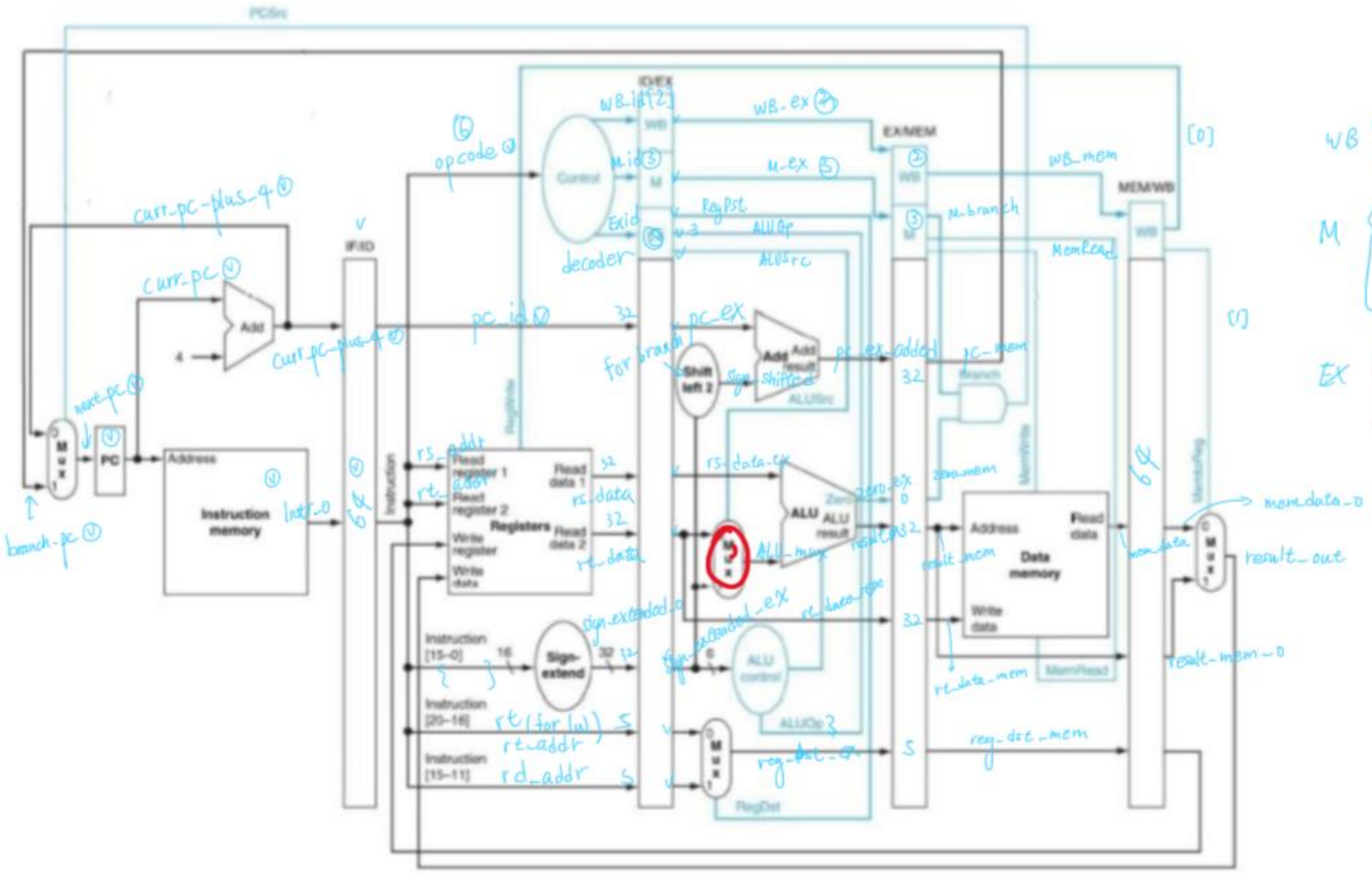
● Architecture

使用題目的架構



● Hardware Module Analysis

這次 lab 主要做的事情就是在各 stage 間加上 pipeline registers
以下是我在 onenote 上每條線名稱及大小的命名及分析
(截圖可能比較模糊但應該還算看得清楚)



WB { RegWrite 0
MemtoReg 1

M { Branch 0
MemRead 1
MemWrite 2

EX { RegDst 0
ALUOp 1 2 3 (EX_id)
ALUSrc 4

● Problems You Met and Solutions

1. 剛接完線完全跑不出結果

後來決定把 reg 的結果印出來，發現各種眼殘手殘
總之有些線的大小設錯了

2. ALU 的結果不對

發現 bus 接線的順序剛好顛倒了
應該是 .ALU_op_o({EX_id[3], EX_id[2], EX_id[1]}),
打成 .ALU_op_o({EX_id[1], EX_id[2], EX_id[3]}),
還好有把 reg 印出來不然找到死

3. 移植 lab3

有些多的線(Jr, Jal)我就把它晾在那，不影響結果

4. 如何解決 data hazard

我把一些沒有 dependency 的指令插進有 hazard 的指令間，有些還需要加 bubble (32'b0)

5. Mult 指令實現

這次兩筆次資都沒有 mult 這個指令我覺得很奇怪，不過我還是加了，
ALU_Ctrl 加個 011000 控制，ALU case 加個相乘的功能

6. Reg 大小調整

原本大小都是自己先加起來在填 N
但發現寫錯要改不太方便就寫成
Pipe_Reg #(.size(2+3+5+32+32+32+32+5+5)) ID_EX(
這樣哪條寫錯比較好調整

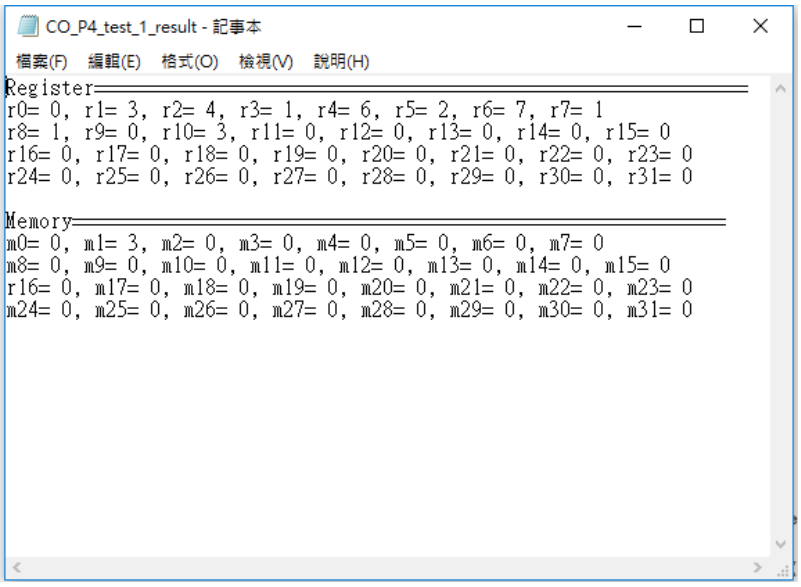
● Result

- **CO_P4_test_1.txt**

My result

```
Register=============================================================

r0=         0, r1=         3, r2=         4, r3=         1, r4=         6, r5=         2, r6=         7, r7=         1

r8=         1, r9=         0, r10=         3, 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=         3, 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

r16=         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
INFO: [USF-XSim-96] XSim completed. Design snapshot 'TestBench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:03 ; elapsed = 00:00:06 . Memory (MB): peak = 792.355 ; gain = 0.000
```

Answer



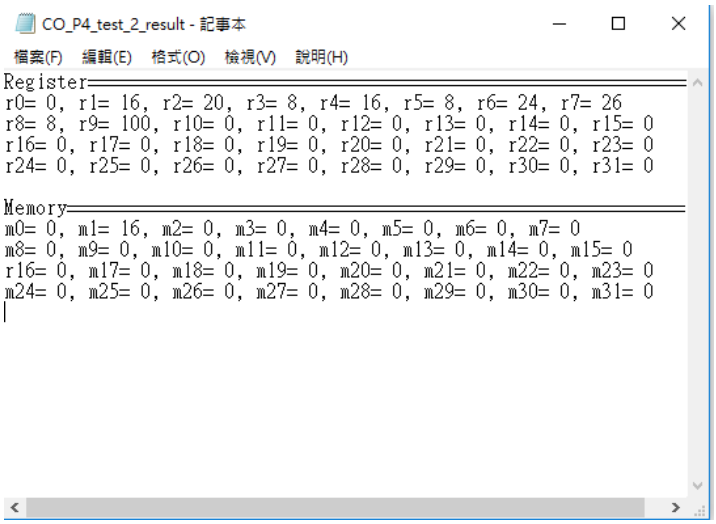
- **CO_P4_test_2.txt**

我不想更動原來的檔案，所以我把更動完的指令存在 CO_P4_test_2_m.txt

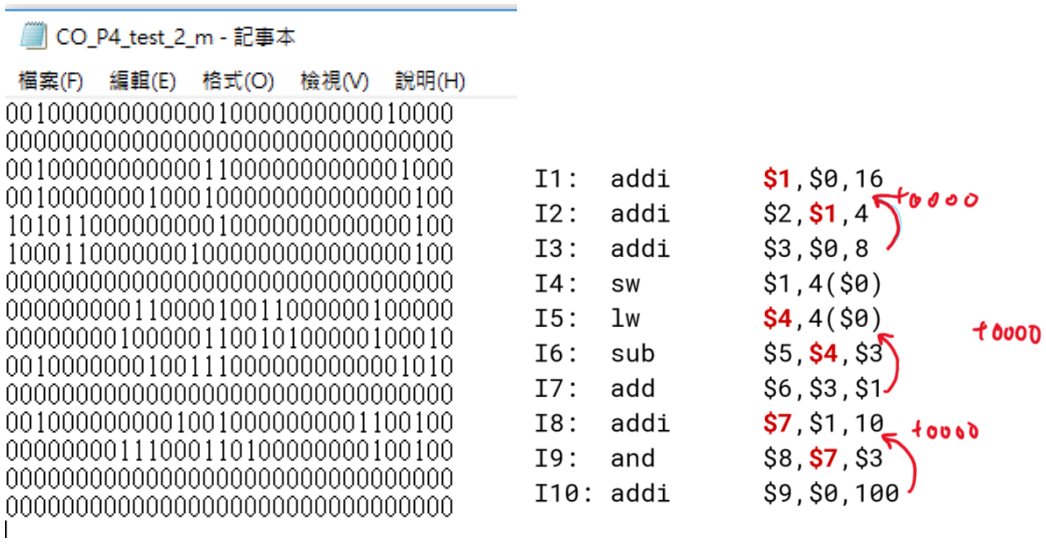
My result

```
Register=============================================================

r0=          0, r1=         16, r2=         20, r3=          8, r4=         16, r5=          8, r6=         24, r7=         26

r8=          8, r9=        100, 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

r16=          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
INFO: [USF-XSim-96] XSim completed. Design snapshot 'TestBench_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:03 ; elapsed = 00:00:07 . Memory (MB): peak = 792.355 ; gain = 0.000
```

Answer

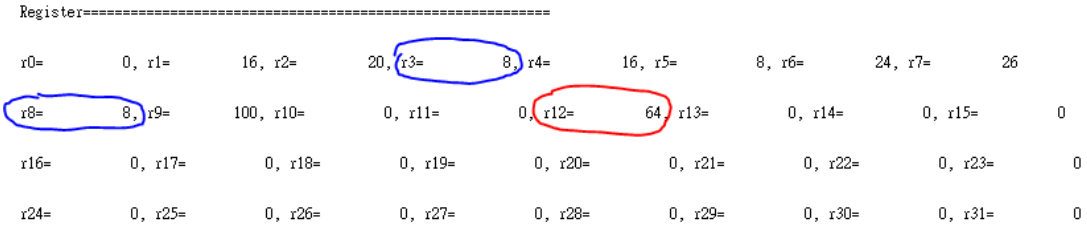


更動的指令



Mult 指令

我直接加 mult 指令在第二筆測資後面測試，r12 = r3 * r8 (= 64)
指令為: 00000000011010000110000000011000
跑出來的結果:



● 心得

剛跑出結果覺得很感動，覺得寫這麼久終於能動了，很有成就感。
但在寫 hardware module analysis
時發現自己做的也只有加個幾個 reg 跟接幾條線而已
突然想不通為什麼搞那麼久
總之寫完還是很開心拉
喔對上次 NewE3 開雷害我壓線失敗，這次有學乖早點開始寫