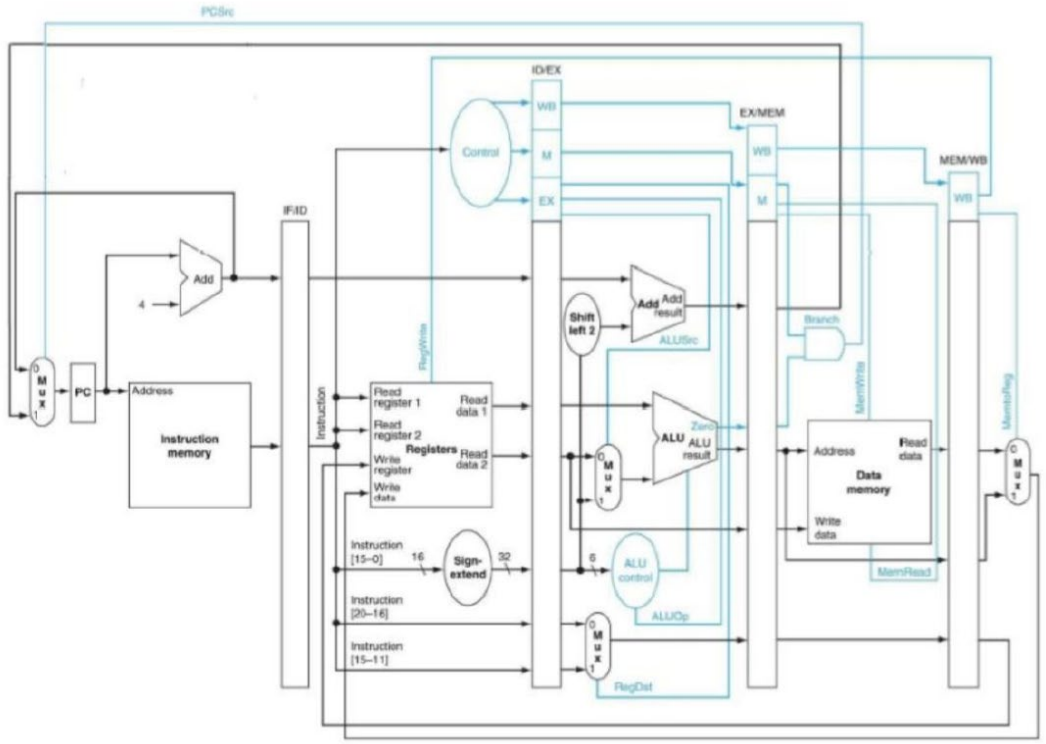
Architecture



Hardware Module Analysis

Pipe_Reg #(.size(64)) IF_ID
sequential address
instruction memory

Pipe_Reg #(.size(148)) ID_EX
Decoder 的全部 output (我把他們包成 WB_ctrl、M_ctrl、EX_ctrl)
sequential address
RS data
RT data
sign extend 的 output
instruction memory[20:16]
instruction memory[15:11]

Pipe_Reg #(.size(107)) EX_MEM
WB_ctrl
M_ctrl
branch address

ALU zero
ALU result
RT data
RD address

Pipe_Reg #(.size(71)) MEM_WB
WB_ctrl
Memory data
ALU result
RD address

Problems and Solutions

1. 不知道原來助教這次的路徑有包含 test/，所以一開始一直讀不到檔案。
2. 一開始不小心把 Mux 3 的 data input 丟反，所以一直跑不出來

Result

Test1

```
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
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 210000 ticks.
>
```

Test2

```
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
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 210000 ticks.
```

我用 stall 的方式避免 data hazard 和 data dependency，在兩個指令之間加了兩個 NOP

```
1   00100000000000010000000000010000    addi $1, $0, 16
2   00000000000000000000000000000000
3   00000000000000000000000000000000
4   00100000001000100000000000000100    addi $2, $1, 4
5   00100000000000110000000000001000    addi $3, $0, 8
6   10101100000000010000000000000100    sw $1, 4($0)
7   10001100000000100000000000000100    lw  $4, 4($0)
8   00000000000000000000000000000000
9   00000000000000000000000000000000
10  00000000100000110010100000100010    sub $5, $4, $3
11  00000000011000010011000000100000    add $6, $3, $1
12  00100000001001110000000000001010    addi $7, $1, 10
13  00000000000000000000000000000000
14  00000000000000000000000000000000
15  00000000111000110100000000100100    and $8, $7, $3
16  00100000000010010000000001100100    addi $9, $0, 100
```

Summary

原本不知道 output 的訊號可以用 concate，還一個一個算是 reg 的第幾個到第幾個 bit，還很容易算錯，改用 concate 就好用很多，不會搞錯。
附上辛苦算還算錯的證明 :)

3. Architecture Diagram