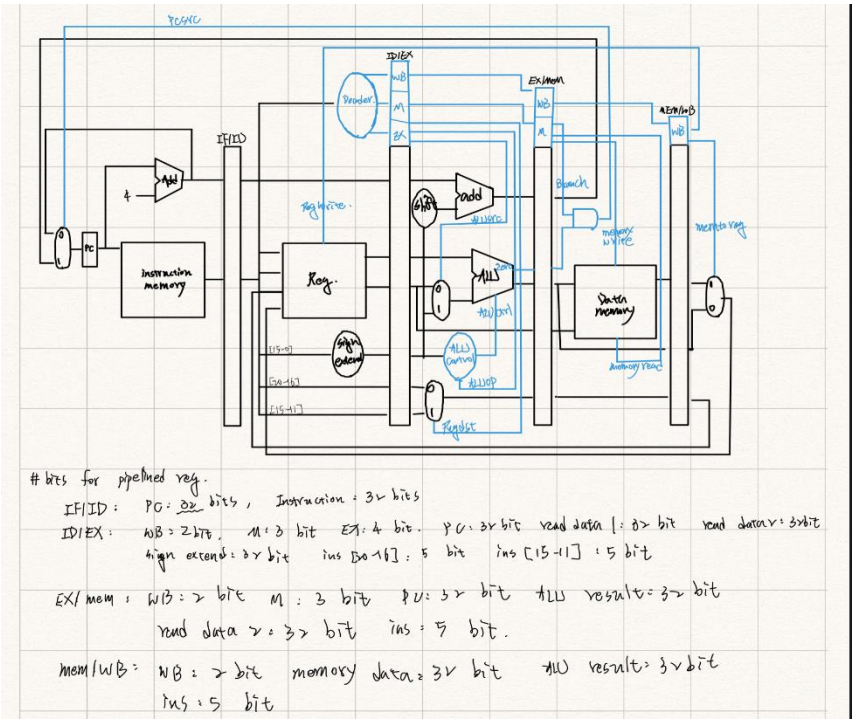
Architecture Diagram:



Hardware Module Analysis:

(1) decoder : 一樣是利用 and or sturcture 先對 instruction[31:26]分析產生 control signal 和 aluop

(2) alu control: 一樣是利用 aluop 和 instruction[5:0] 產生 alu_ctrl 但這次因為有乘法 alu_ctrl 對乘法的運算我將 alu_ctrl 設為 1000

(3) piplined_cpu : 照著 architecture diagram 接

(4) 其餘皆和之前的 lab 實做一樣

Problems I Met and Solutions:

因為 pipreg 的 bit 數太多了在做不同的訊號區分時很容易搞混，所以我另外宣告了很多不同的 wire 在一個 stage 時先將 pipreg 接到 wire 上在接到 module 中

Result:

Text1:

```
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
```

Text2:

```
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
```

Machine code:

00100000000000010000000000010000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00100000001000100000000000000100
00100000000000110000000000001000
10101100000000010000000000000100
10001100000001000000000000000100
00100000001001110000000000001010
00000000011000010011000000100000
00000000100000110010100000100010
00100000000010010000000001100100
00000000111000110100000000100100

Summary:

這次了 lab 和 lab2 差不多只是多了 pipreg 使用但這也使的 architecture diagram 中 wire 的名字突然多了很多種，這也導致一開始在實做時很常有對錯名字的問題發生使的輸入的 input 是錯的，但其實只要細心一點就好了