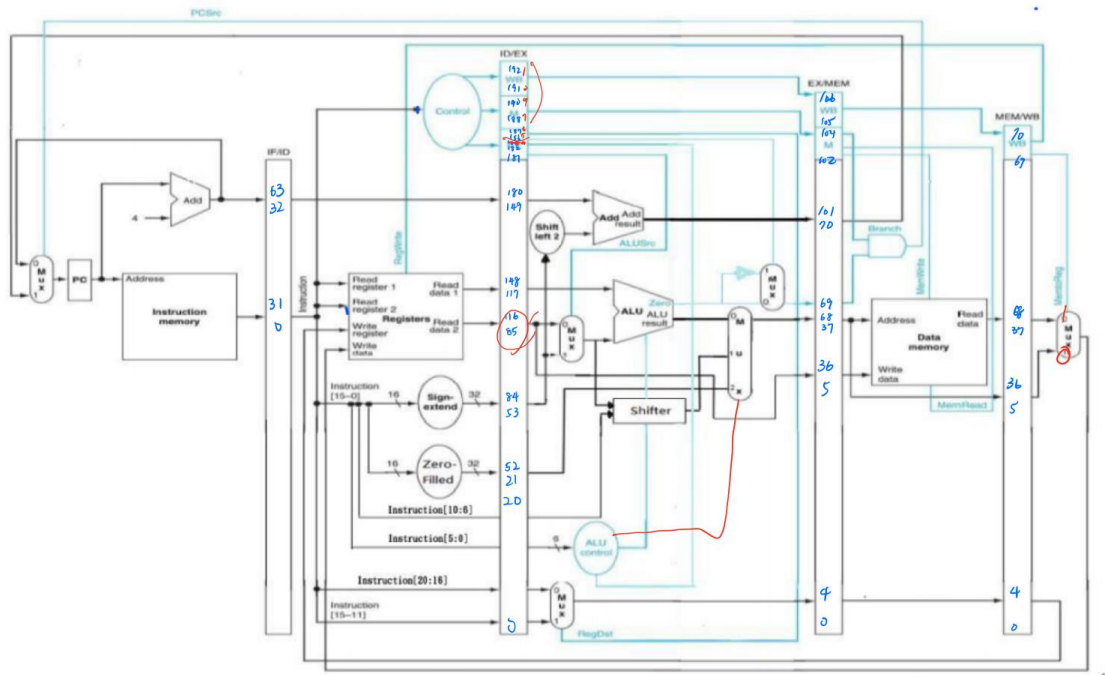


Computer Organization Lab5

The input fields of each pipeline register:

3. Pipelined CPU

A. Architecture diagram



Compared with lab4, the extra modules:

Pipeline_Reg: 4 個存 Pipeline data & signal 的 register

Explain your control signals in sixth cycle (both test data

CO_P5_test_data1 and CO_P5_test_data2 are needed)

CO_P5_test_data1

MemtoReg: 0

RegWrite: 1

MemRead: 0

MemWrite: 0

RegDst:1

PCSrc:0

AluOP:010

ALUSrc:0

CO_P5_test_data2

RegWrite: 1

MemtoReg: 0

RegDst:0

PCSrc:0

AluOP:011

ALUSrc:1

MemRead: 0

MemWrite: 0

Problems you met and solutions:

原本想要分很多個 reg，後來發現題目要求用 4 個 reg，所以要一個一個算每

個 reg 的大小(如圖)，中間有算錯幾次就不能跑

還有最後的 MemWriteReg 的 01 訊號是跟 Lab4 Reference 相反的，也是

找了一段時間才找到

Summary:

這個作業學會了如何在 CPU 中使用 pipeline reg 暫存 data & signal 達到加

速執行的效果

```
Pipeline_CPU.v U  data1_result.txt U  TestBench.v U
2: > data1_result.txt
1  addi r1,r0,31
2  addi r2,r0,17
3  addi r4,r0,14
4  and  r3,r5,r6
5  or   r5,r1,r0
6  and  r6,r2,r1
7
8
9  result
10
11 # Register=====
12 #
13 # r0=0, r1=31, r2=17, r3=0, r4=14, r5=31, r6=1, r7=0
14 #
15 # r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0
16 #
17 # r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0
18 #
19 # r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0
20 #
21 # Memory=====
22 #
23 # m0=0, m1=0, m2=0, m3=0, m4=0, m5=0, m6=0, m7=0
24 #
25 # m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0
26 #
27 # m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0
28 #
29 # m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
30 #
31
32 Pipeline_CPU.v U  data2_result.txt U  TestBench.v U
33: > data2_result.txt
1  addi r2,r0,7
2  addi r3,r0,8
3  addi r4,r0,9
4  addi r5,r0,10
5  sw  r2,r0(4)
6  sw  r3,r3(8)
7  sub r5,r3,r2
8
9  result
10
11 # Register=====
12 #
13 # r0=0, r1=0, r2=7, r3=8, r4=9, r5=1, r6=0, r7=0
14 #
15 # r8=0, r9=0, r10=0, r11=0, r12=0, r13=0, r14=0, r15=0
16 #
17 # r16=0, r17=0, r18=0, r19=0, r20=0, r21=0, r22=0, r23=0
18 #
19 # r24=0, r25=0, r26=0, r27=0, r28=0, r29=128, r30=0, r31=0
20 #
21 # Memory=====
22 #
23 # m0=0, m1=7, m2=0, m3=0, m4=8, m5=0, m6=0, m7=0
24 #
25 # m8=0, m9=0, m10=0, m11=0, m12=0, m13=0, m14=0, m15=0
26 #
27 # m16=0, m17=0, m18=0, m19=0, m20=0, m21=0, m22=0, m23=0
28 #
29 # m24=0, m25=0, m26=0, m27=0, m28=0, m29=0, m30=0, m31=0
30 #
31
32 C:\Users\USER\Desktop\大學學習\109-2 計算機組織\Lab5>iveril
C:\Users\USER\Desktop\大學學習\109-2 計算機組織\Lab5>vvp l5
WARNING: TestBench.v:43: $readmemb(CO_P5_test_data1.txt): N
Register=====
r0=0, r1=31, r2=17, r3=0, r4=14, r5=31, r6=1, r7=0
r8=0, 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=128, r30=0, r31=0
Memory=====
m0=0, m1=0, 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
TestBench.v:50: $stop called at 410000 (1ps)
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 410000 ticks.
> |
C:\Users\USER\Desktop\大學學習\109-2 計算機組織\Lab5>iveril
C:\Users\USER\Desktop\大學學習\109-2 計算機組織\Lab5>vvp l5
WARNING: TestBench.v:43: $readmemb(CO_P5_test_data2.txt): N
Register=====
r0=0, r1=0, r2=7, r3=8, r4=9, r5=1, r6=0, r7=0
r8=0, 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=128, r30=0, r31=0
Memory=====
m0=0, m1=7, m2=0, m3=0, m4=8, 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
TestBench.v:50: $stop called at 410000 (1ps)
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 410000 ticks.
> |
```