

Computer Organization, Fall 2011

Lab 5: Pipelined CPU II

Due: 2011/12/23

1. 作業目的

更改 Lab4 所設計的 CPU, 實作一個進階版本的 Pipelined CPU。Pipeline 為 CPU 設計中一項非常重要的方法, 詳細閱讀此文件並實作此 Lab, 會了解 Pipeline 如何讓 CPU 效能進行提升。

2. 作業要求

a. 基本指令(60%) : ADD, SUB, AND, OR, SLT, ADDI, LW, SW, MULT

- 實作的部分為 Hazard Detection Unit 與 Forwarding Unit
- 當有 data dependency 時必須做 forward
- 當有 load-use 時: PC 與 IF/ID stage 不可被寫入, ID/EX 必須被 flush 以產生 stall
- Mult rd, rs, rt ; rd=rs*rt

0	Rs	Rt	Rd	0	24(0x18)
---	----	----	----	---	----------

b. 進階指令(40%) : BEQ, BNE, BGEZ, BGT

- 修改 Hazard Detection Unit 來 flush 不必要的 registers
- BRANCH 成功與必須 flush IF/ID, ID/EX, EX/MEM registers

c. Demo : 未 demo 視同缺繳。

測資說明:

測試資料內容在 CO_P5_test1.txt, 如下所示, 請用 **forwarding** 解決 I1 和 I2, I5 和 I6, I8 和 I9 以及 I9 和 I10 的 data dependency; I5 與 I6 為 load-use(需 stall 1 cycle)

I1: addi \$1, \$0, 16

I2: mult \$2, \$1, \$1

```

I3:  addi  $3,$0,8
I4:  sw   $1,4($0)
I5:  lw   $4,4($0)
I6:  sub  $5,$4,$3
I7:  add  $6,$3,$1
I8:  addi $7,$1,10
I9:  and  $8,$7,$3
I10: slt  $9,$8,$7

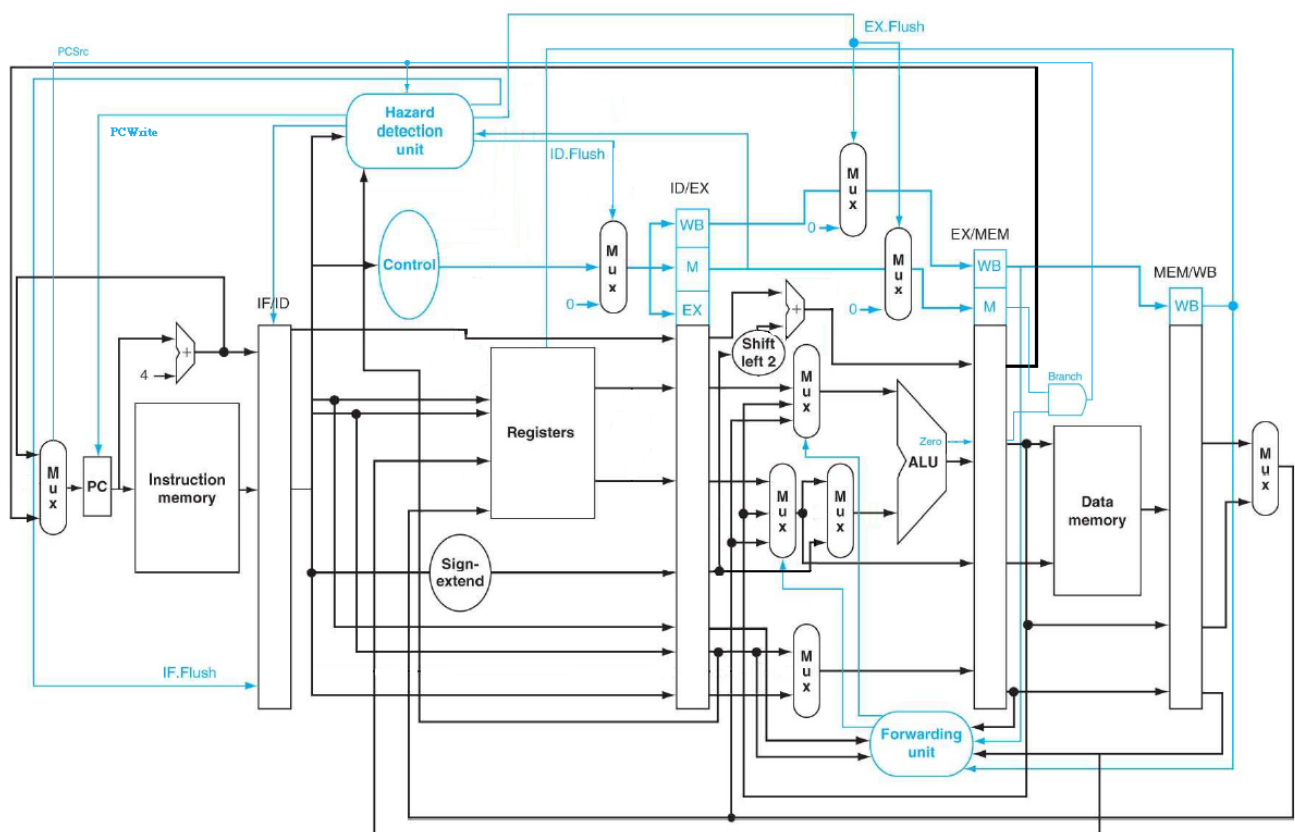
```

結果：r1 = 16; r2 = 256; r3 = 8; r4 = 16; r5 = 8; r6 = 24; r7 = 26; r8 = 8; r9 = 1;

date_mem[1] = 16;

CO_P5_test2.txt 為 Lab3 之 Bubble Sort 但是去掉 **JMP**，其中包含基本指令與進階指令

3. 架構圖.



4. 評分方式

- 滿分 100, 抄襲一律 0 分!
- DEMO 用報告
- DEMO 時間 12/26,27,29
- 遲交一天打九折!

5. 繳交方式

上傳至 E3