

## 103-2 Computer Organization Pipeline Simulation Homework

### ➤ 作業要求

請同學以 C/C++程式語言 實作課本 Pipeline 處理器之指令運作模擬程式。同學需實作下列指令：「lw」、「sw」、「add」、「sub」、「and」、「or」、「beq」、「slt」。並且，能夠偵測處理「data hazard」、「hazard with load」以及「branch hazard」三種 hazards。

### ➤ 程式的輸入與輸出

**輸入：**請依照以下公布的 32-bit 機器碼指令，其中包含需要處理 data hazard、hazard with load 以及 branch hazard。每題的指令順序都會放在「\*.txt」檔案中，並由同學們撰寫的模擬程式讀入。一共四題，另有一加分題，每一題都是獨立的，不會相互影響。

1. 請同學的模擬程式從 "General.txt" 檔中，讀出四行指令，分別為：

```
lw    $3, 0x02($7)
sub   $2, $6, $7
and   $8, $5, $4
slt   $1, $5, $6
```

將執行結果寫入 "genResult.txt" 檔案中。

2. 請同學的模擬程式從 "Datahazard.txt" 檔中，讀出四行指令，分別為：

```
sub   $2, $4, $3
and   $4, $2, $3
or    $4, $4, $2
add   $3, $2, $4
```

將執行結果寫入 "dataResult.txt" 中。

3. 請同學的模擬程式從 "Lwhazard.txt" 檔中，讀出四行指令，分別為：

```
lw    $2, 0x04($8)
and   $4, $2, $5
or    $4, $4, $2
add   $3, $2, $4
```

將執行結果寫入 "loadResult.txt" 中。

4. 請同學的模擬程式從 "Branchazard.txt" 檔中，讀出四行指令，分別為：

```
beq   $1, $5, 0x02    #(branch 至 lw 指令)
and   $3, $4, $5
or    $6, $7, $8
lw    $4, 0x00($0)
```

將執行結果寫入 "branchResult.txt" 中。

Bonus: 請同學的模擬程式從 "Branchbonus.txt" 檔中，讀出五行指令，分別為：

```
sub $2, $5, $0
beq $1, $2, 0x02    #(branch 到 lw 指令)
and $3, $4, $5
or  $6, $7, $8
lw  $4, 0x04($8)
```

將執行結果寫入 "bonusResult.txt" 中。

P.S. Branch 指令第三個欄位應為 label，這裡為了方便而表示成數字形式，非正確寫法。

**輸出：**請依照上述每一題規定，將結果分別寫入到「branchResult.txt」、「dataResult.txt」、「genResult.txt」、「loadResult.txt」和「bonusResult.txt」（加分）之中。如有 hazard 需在螢幕輸出提示訊息。輸出文件格式請按照以下範例（Output.txt），也就是列印出在每個 clock cycle 時，各個 pipeline registers 所儲存的值。「Instruction」和「Control signals」的結果已 0/1 表示，其餘皆以十進位表示。「Control signals」的排列順序請參照老師投影片「Chap4\_2, pp.29」所示。

課程網站上提供參考範例輸入：InstrIn.txt，其結果輸出：Output.txt。

```
lw $1, 0x00($8)
add $3, $0, $2
beq $0, $2, 0x06 #(branch PC+4+4×6)
```

CC 3:

Registers:

\$0: 0	\$1: 5	\$2: 8
\$3: 6	\$4: 7	\$5: 5
\$6: 1	\$7: 2	\$8: 4

Data memory:

00:	9
04:	6
08:	4
12:	2
16:	5

IF/ID :

PC	12
Instruction	000100000000000100000000000000110

ID/EX :

ReadData1	0
ReadData2	8
sign_ext	0
Rs	0
Rt	2
Rd	3
Control signals	110000010

EX/MEM :

ALUout	4
WriteData	5
Rt	1
Control signals	01011

MEM/WB :

ReadData	0
ALUout	0
Control signals	00

## ➤ 暫存器、指令記憶體與資料記憶體的初始化

暫存器初始化為以下所示：假設宣告一般暫存器\$0~\$8，共9個暫存器。

暫存器編號	\$0	\$1	\$2	\$3	\$4
初始值	0	5	8	6	7
暫存器編號	\$5	\$6	\$7	\$8	
初始值	5	1	2	4	

資料記憶體初始化為以下所示：

記憶體位址	0	4	8	12	16
初始值	9	6	4	2	5

指令記憶體的初始位址由「0」開始。

## ➔ 作業上傳

請在 **6/30(二) 23:59:59** 前，將程式碼及註解說明報告一併上傳至  
<ftp://msp.csie.ncu.edu.tw>

帳號/密碼：COHW / cohwcchw

檔名命名為：學號\_版本碼.rar 或.zip

例如：975002XXX\_1.rar （若上傳多個版本，以最新的版本評分）

**注意！請將檔案分別上傳至相對應的目錄中，以方便批改，切勿抄襲！**

若有任何問題請寄 email 給助教（張瓊方 [fang177sss@yahoo.com.tw](mailto:fang177sss@yahoo.com.tw)）