**VLSI系統設計-HW5**

**(共75分)**

**姓名: 李煒權**

**學號: E24045165**

**授課教師:**

**1.系統簡介(共25分)**

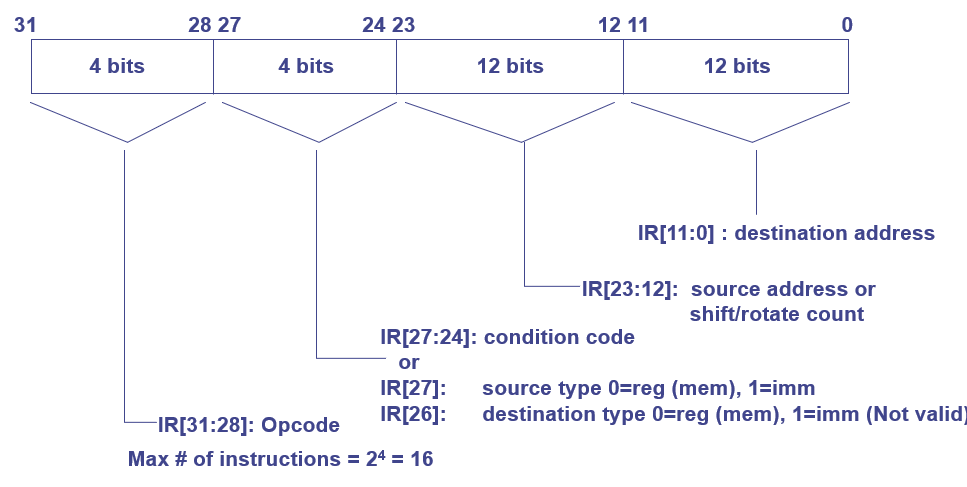
1.指令集格式(5分)

* 13 instructions ( added 2 instruction )
* 32 bits instruction register
* 12 bits address register
* 5 bits PSR (processer status register)
* 33 bits result

2.指令集格式欄位的名稱、長度、說明(5分)

指令有32bits組成

1. OPCODE
   1. 第31～28位元
   2. 主要分辨指令主要功能（例：BRA OPCODE為0001）
2. SRCTYPE
   1. 第27位元
   2. SRCTYPE若為IMD則為1，非IMD值則為0
3. DSTTYPE
   1. 第26位元
   2. DSTTYPE若為IMD則為1，非IMD值則為0
4. CCODE
   1. 第27～24位元
   2. CONDITION CODE，負責分辨當BRA指令時，程式是否要繼續或跳到指定的行列
5. SRC
   1. 第23～12位元
   2. SOURCE 地址，若SHF或ROT指令則是位移位數
6. DST
   1. 第11～0位元
   2. DESTINATION地址



3.BRA指令的定址(5分)

PRE-instruction：上個指令在執行就會完成setccode這個task，這個task主要是設定PSR值。

1. 在fetch階段，IR接收到了指令後，PC依然正常加一

2. 在execute階段，利用IR上的CCODE，比對上條instuction已經設定好的PSR，若條件相符合則跳到指定的指令地址（PC = DST），program counter改變，若條件不符合則PC = PC + 1

3. 在write階段則無須工作

4.架構(共10分)

1.基本零件(5分)

1. CLOCK - 產生cycle以讓系統達到state machine的效果

2. Memory – 記憶體，供系統儲存指令後數值，該位置的值為指令或數值需看指令是怎麼運用的

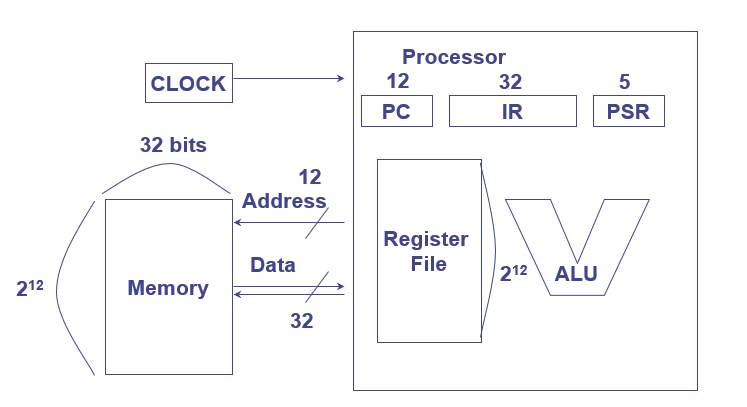
3. Register- Processor裡的記憶體，複製暫存processor裡的一些值

4. ALU – 運算單元，負責運算的部分，例：ADD，MUL等

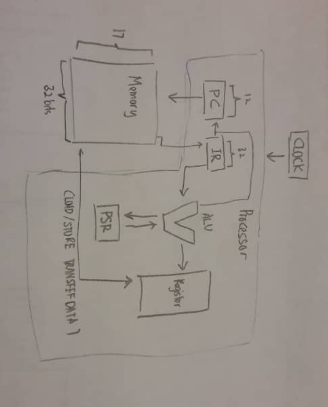
5. PC – program counter, 儲存下一條指令在MEM的地址

6. IR – 儲存指令

7. PSR – 存取判斷result是否有 Negative，Even，Parity，Zero，Carryout的特性，以供BRA指令做判斷



2.資料路徑(系統架構圖)(需自行製作)(5分)



**2.系統目前可執行之指令(5分)(需說明新增之指令)**

**1. NOP**

**2. LD**

**3. STR**

**4. BRA**

**5. ADD**

**6. CMP**

**7. MUL**

**8. DIV（NEW）- 除法ALU**

**9. RMD（NEW）- 餘數ALU**

**10. SHF**

**11. ROT**

**12. JMP（NEW）- JUMP，任何情況下都跳到指定的下個指；。令，改變PC值**

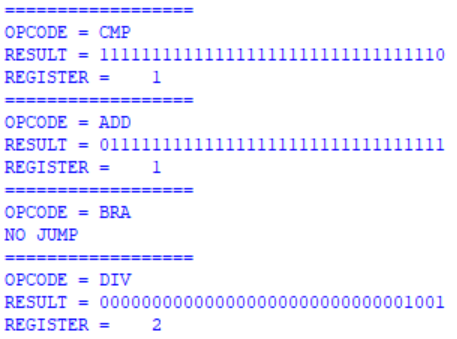
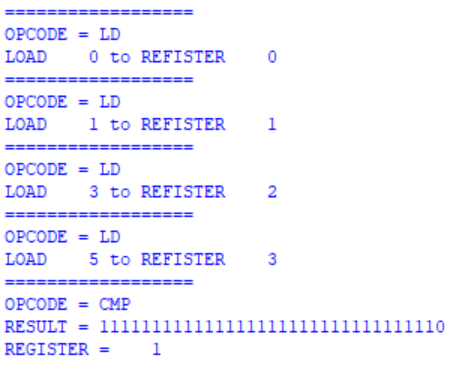
**13. HLT**

**3.系統驗證方法與結果分析(共20分)**

1.驗證方法(需說明Test Program) (10分)

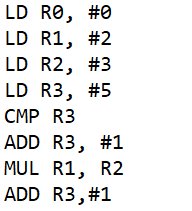
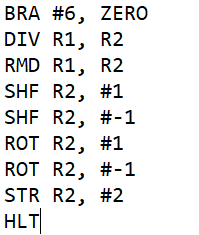
利用display方式，顯示OPCODE，result，result存取的位置等，比對自己寫好的BinaryCode，檢查程式有沒有與理想中有一樣的結果。

一下是我列印出來的結果：

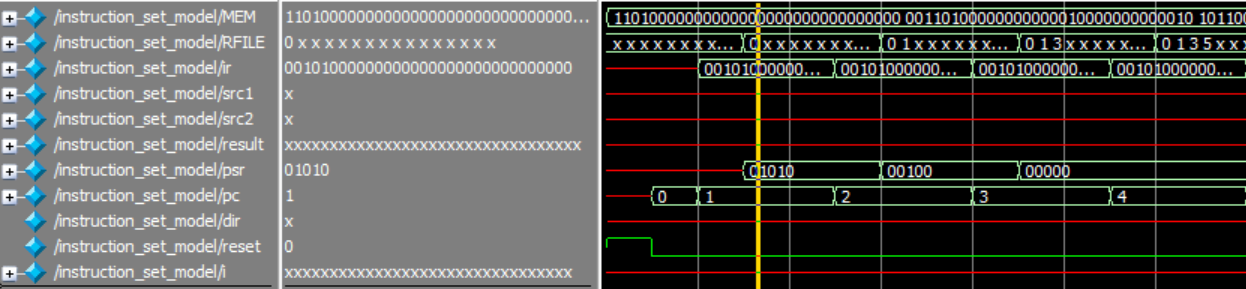


2.結果分析(需附上nWave截圖) (10分)

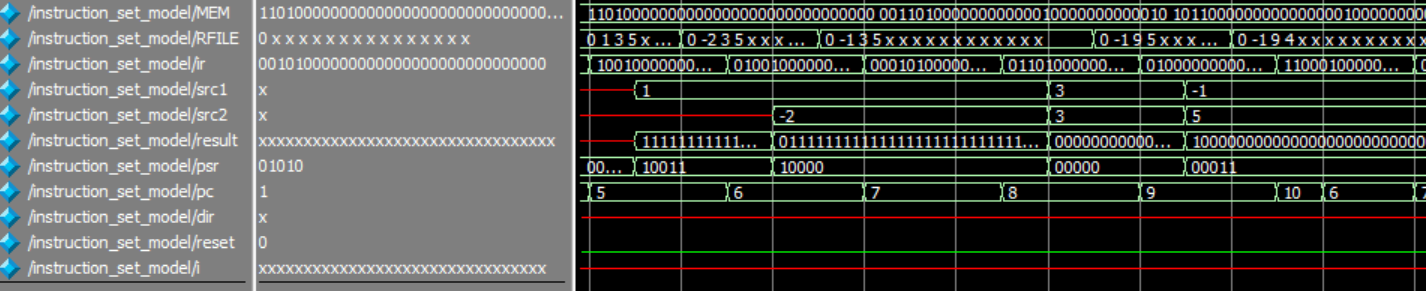
這是我設計的指令，總共有17行指令

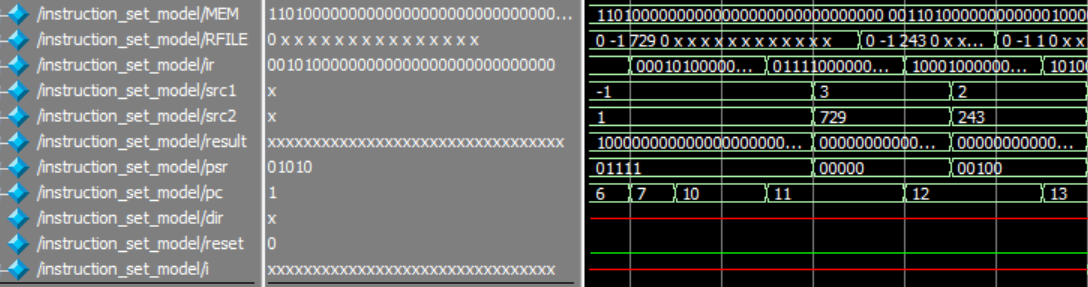
前4個instructions都為LD檔案，RFILE（register）被置入了幾個值



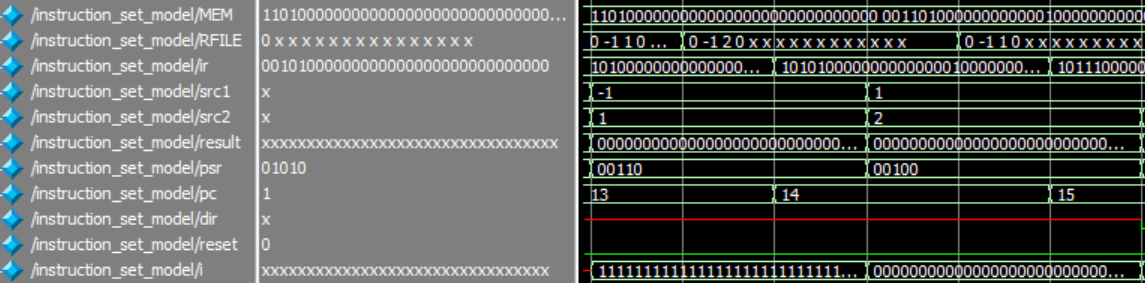
第4個instruction就開始運算，這是一個loop，第九行指令是JMP指令，在波形裡10直接換到6就完成了JMP instruction，JMP回到第6行instuction進行判斷時候要繼續在loop裡面運算。



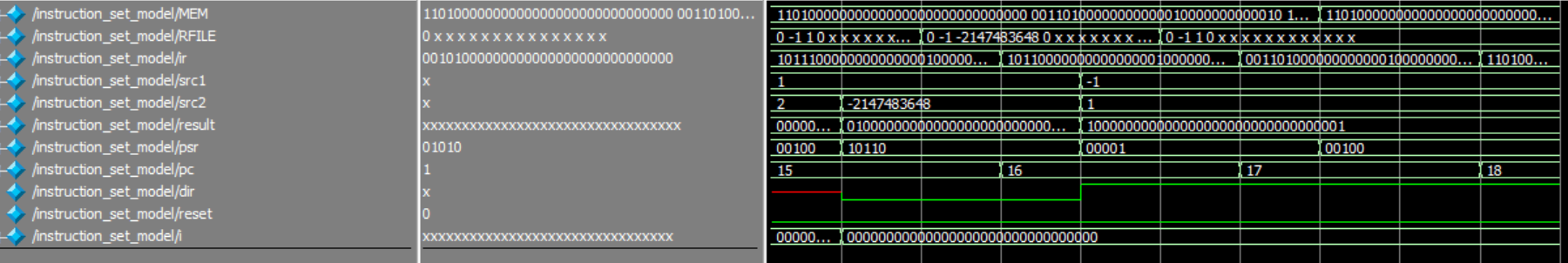
經過了5次的loop，BRA的條件判斷成功直接跳到第十行指令繼續執行程式



這兩個指令為SHF指令，主要在向左或向右位移位元



這兩個指令則是ROT指令，在PC等於18時程式結束



**4.問題與討論(5分)**

**我的問題是我們課堂學的RISC指令是簡易版的嗎？因為我在網絡上查的RISC附上的指令，在IR上面有三個值的輸入(SRC1,SCR2,DST)，兩個值的輸入不太方便，例如說一直要利用兩個值相加，在只能輸入兩個值做加法時必定有一組號碼會改變，這樣會導致不能一直重用這兩個值，要一直load才有辦法達成。**

**5.心得(加分，5分)**

**這次的作業讓我更理解整個code的架構，細節等，透過觀察波形debug，雖然看得很累，但卻是最直接的debug方式。在程式裡原本只加入DIV及RMD新的功能，在在設計loop運算時遇到了一點問題，在BRA判斷必須比較麻煩地用一些數值作為判斷，所以我增加了JUMP指令，能忽視所有情況地JUMP，最後才想起原來在要在BRA指令的CCODE改為ALWAYS就達到JUMP的功能，自己白寫了一些程式。**