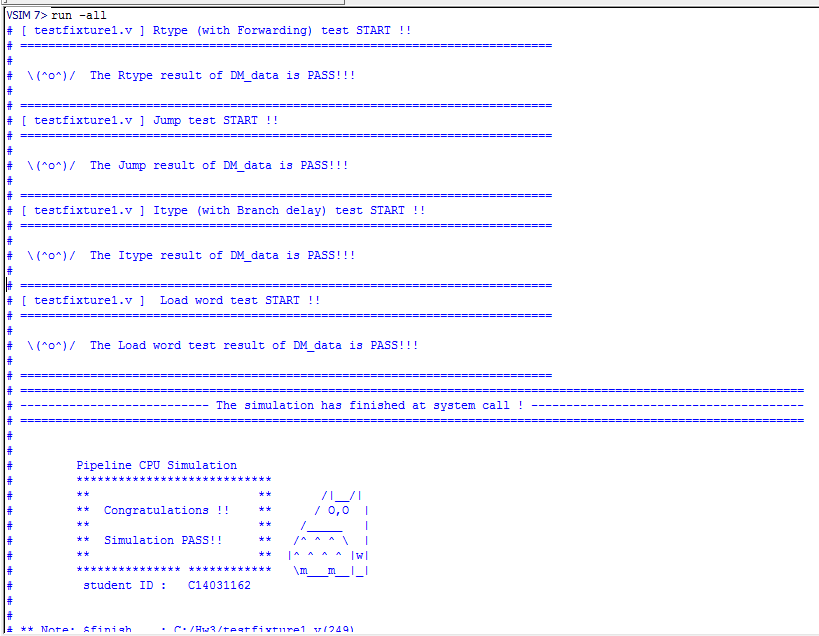
**Computer Organization 2016**

**HOMEWORK III**

系級: 資訊107 學號: C14031162 姓名: 洪正皇

**實驗結果圖(snapshot of result)**

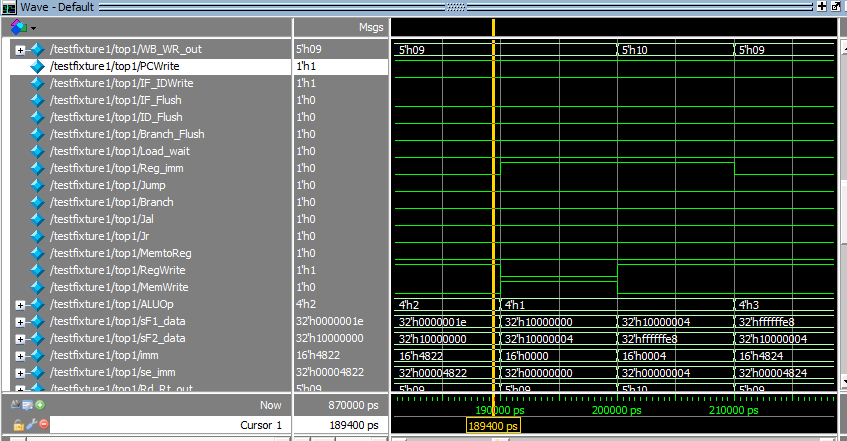
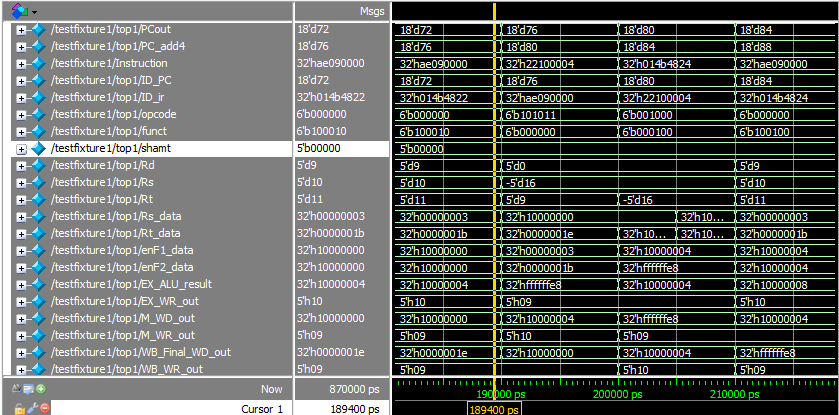
****

**指令波型圖( Instruction waveform of some situations)**

(Please explain why your snapshot is correct, including the wires, signals. The description should be as detailed as possible, e.g. why this situation occurs, and in waveform where does it occur?)

1. Instruction with Forwarding

R-type:



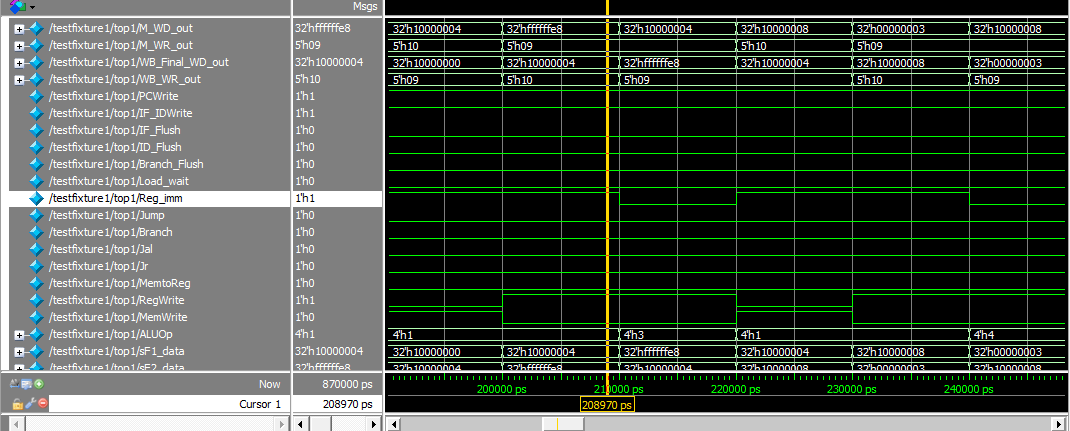
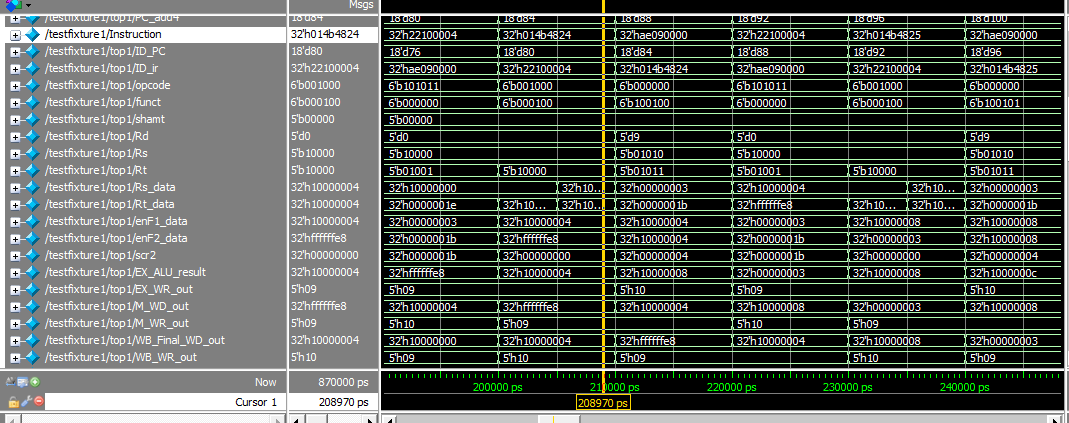
Description:

由ID\_PC和ID\_ir可以看出目前在IM\_Data main裡面的這行， 12: sub $t1, $t2, $t3 #sub

可以看到opcode、funct是sub指令沒錯，Rd,Rs,Rt也是$t1, $t2, $t3沒錯，根據前面code可以算出目前Rs,Rt餵入register後應得到3和27，與Rs\_data, Rt\_data相符，沒有forwarding也不是用imm值，可以看到下個階段的enF1\_data, enF2\_data沒有取錯，而ALU\_result的值也是-24，最後可以看到WR\_out和WD\_out正確的從EX->M->WB傳回register寫入。

控制訊號的部分ALUOp=2代表sub，因為要寫入$t1，regwrite=1，沒有任何的branch，沒有mem存取，jump, mem存取類的指令全部為0，也不是取imm\_data。

I-type:



Description:

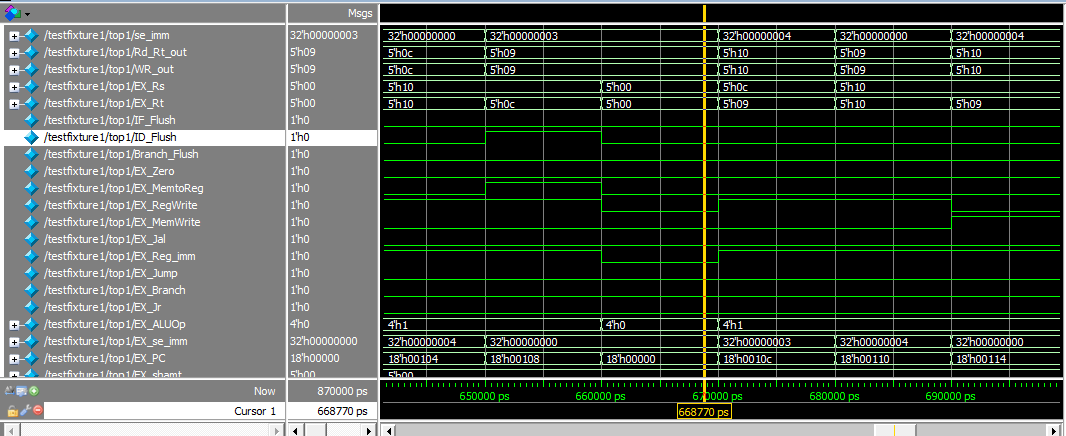
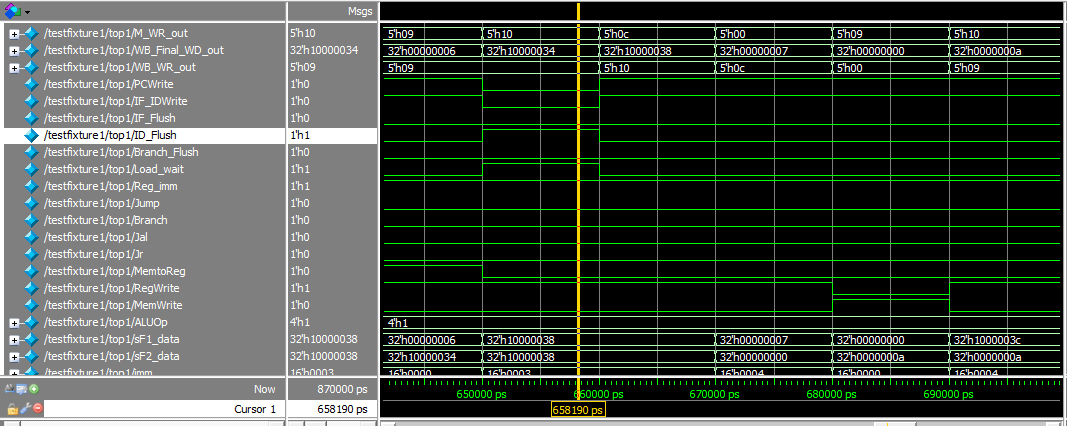
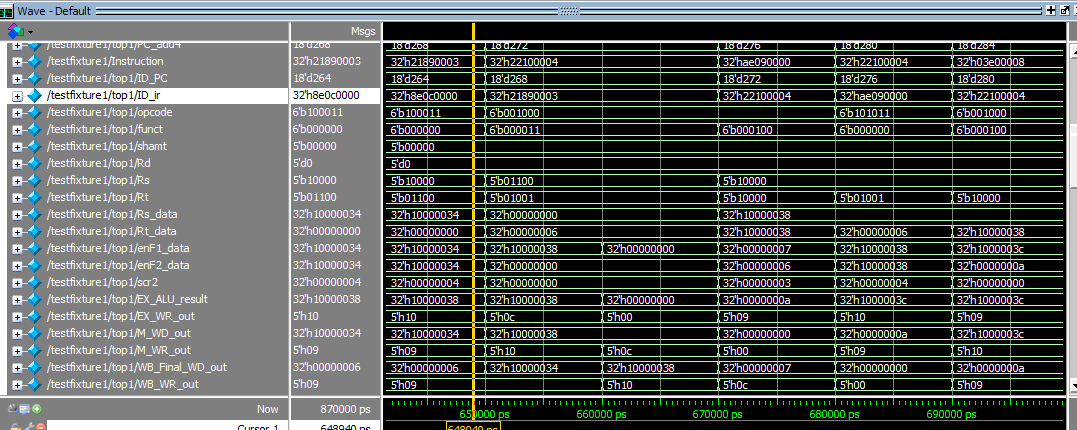
由ID\_PC和ID\_ir可知此為main裡面這行

14: addi $s0, $s0, 4

由opcode可以看出是addi沒錯，Rs=Rt=16=$s0沒錯，不過這時Rt是最後要寫入的位址，Rd是無用的，因此都可以忽略，手算可知目前$s0為2的28次方+4，與Rs\_data相符，在下一個階段(EX階段)的enF1\_data是取沒有forwarding的資料沒錯，scr2則需要取imm的值=4，可以看到ALU\_result為2的28次方+8，EX\_WR\_out為16進位的10，即10進位中的16=$s0沒錯，之後WR\_out和WD\_out就安全的傳回去寫入register了。

控制線的部分與Rtype較不一樣的地方是Reg\_imm，因為Itype的scr2是取用imm區塊的值，故Reg\_imm的值應為1(Rtype為0)，其餘皆與Rtype相同。

1. Load Stall:

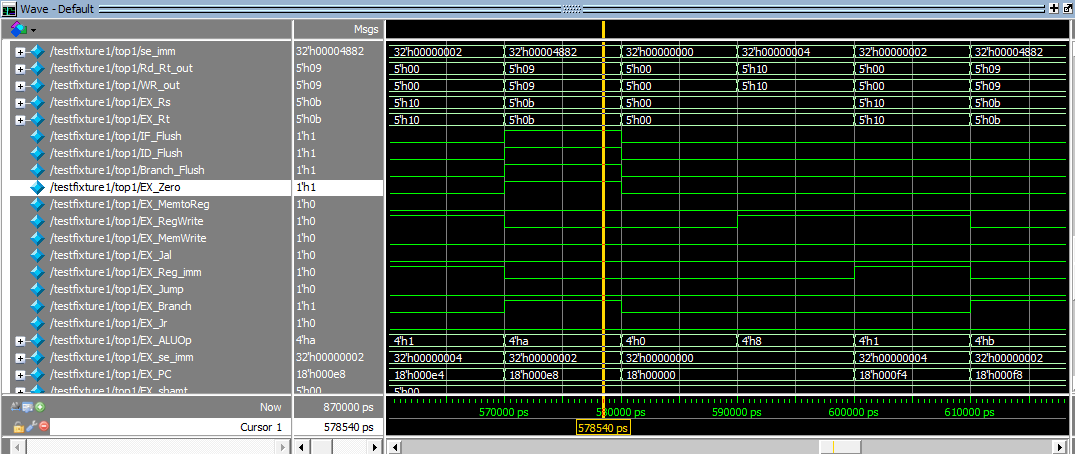
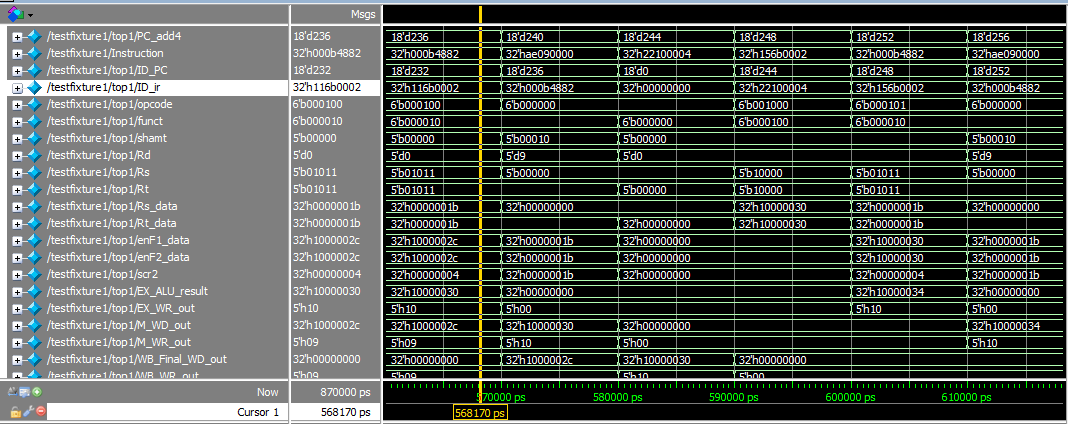


Description:

由ID\_PC及ID\_ir可知此為lw指令，opcode也正確，這時候可以看到下一個cycle的Rs與lw的Rt相同，因lw取值在Mem階段，故需要stall一個cycle，否則forwarding也來不及給正確的值，從第一張截圖可以看到許多資料在兩個cycle間沒有做更動，成功stall了一個cycle。

控制線的部分，因為stall了一個階段，又不能讓重複的ID指令做兩次，故此時將ID\_flush設為1，可以看到下一個cycle(EX)的控制線值全部被設為0，即使其他運算的值往後傳了，但因為控制線為0，不能read也不能write，故此指令不會重複做兩次。

Branch Delay (& Flush):



Description:

由ID\_ir可以看出此為beq指令，取imm值，下一個cycle的enF1\_data==scr2，故可以看到下一個cycle的EX\_Zero為1，所以需要branch，而且目前IF、ID的指令都必須清除，IF、ID\_flush為1，從第一張截圖可以看到IF\_flush後的下一個ID\_PC為0，故輸入指令全為0，成功flash掉IF，第二張截圖可以看到，ID\_flush之後下一個cycle的EX控制訊號都為0，不會寫入也不會讀取，也成功flush掉ID了，在ID\_PC為0以後可以看到下一個ID\_PC為244，而不是0+4，即當初要branch的正確位址，成功清除跑錯的指令，以及跳入正確位址。

**心得(Report)**

(請寫下完成本次作業的心得、學到哪些東西、困難點的部分。大約   
 100~200字 )

(Please write your learned lesson and conclusion, and difficult point. About   
 100~200 words)

在做Hw之前，覺得老師上課教pipeline教得滿輕鬆的，我也學得滿輕鬆的，一開始做Hw的時候發現，好多線不知道怎麼接，搞不清楚哪裡要接哪裡，更頭大的是，還要加入時間的觀點，才發現自己有很多細節沒有仔細想過，常常接錯線，在寫作業的過程中，需要仔細的去看每一條線，完成整份Hw之後我發現CPU所有的線條紋路都刻在我的腦中了，收穫良多。