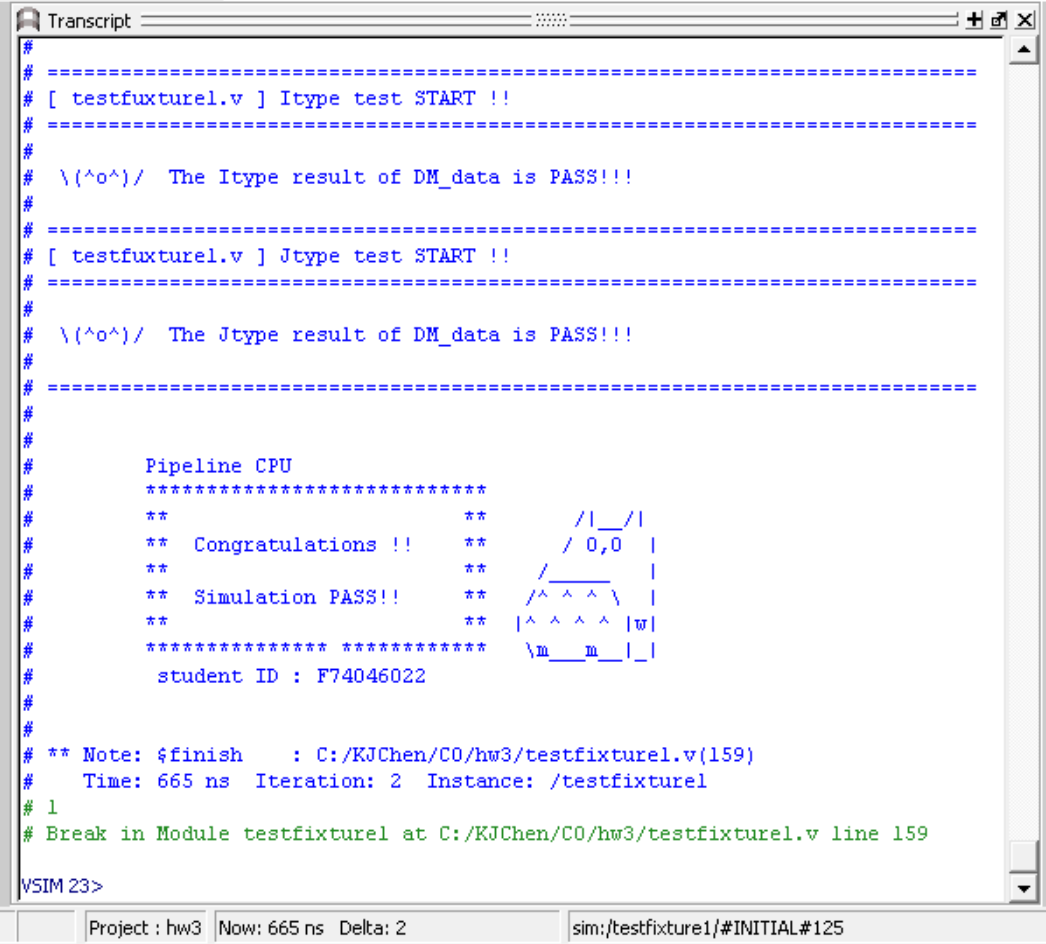
**Computer Organization 2017**

**HOMEWORK III**

系級: 資訊108 學號: F74046022 姓名: 陳冠仁

**實驗結果圖(snapshot of the results)**

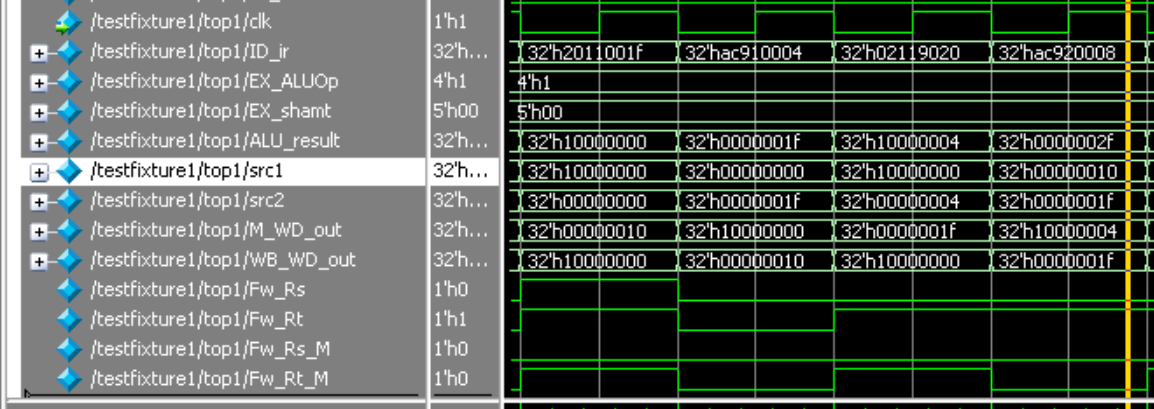
****

**指令波型圖(Snapshot of the instruction execution waveforms.)**

(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:



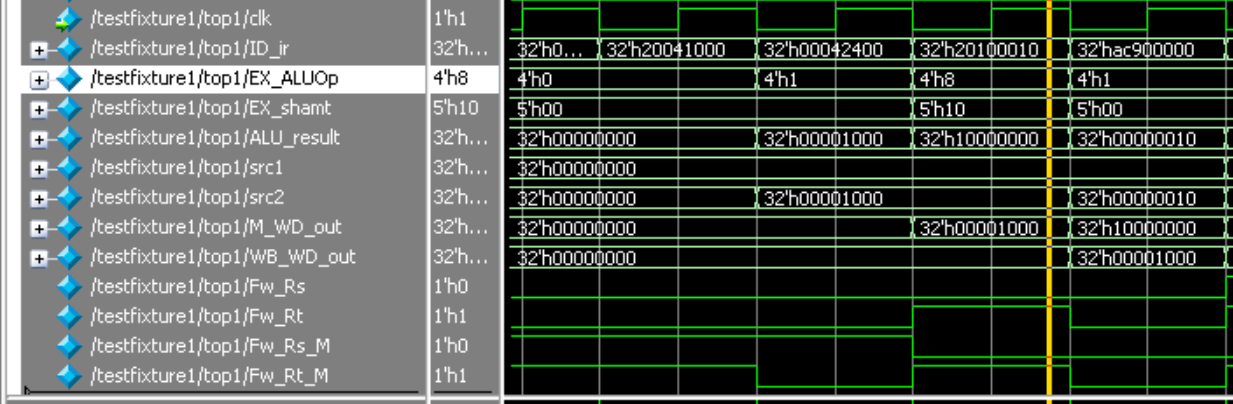
Description:

1. 2011001f //addi $17, $0, 31
2. ac910004 //sw $17, 4($4)
3. 02119020 //add $18, $16, $17

(3)=>(1) RAW

src2 forwarded from WB to EX when (3) in EX

I-type:



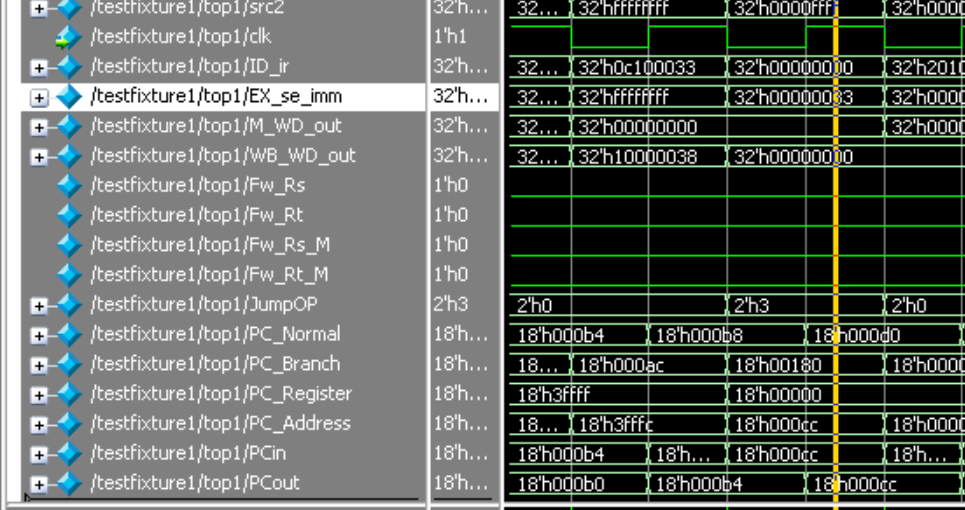
Description:

1. 20041000 //addi $4, $0, 4096
2. 00042400 //sll $4, $4, 16

(2)=>(1) RAW

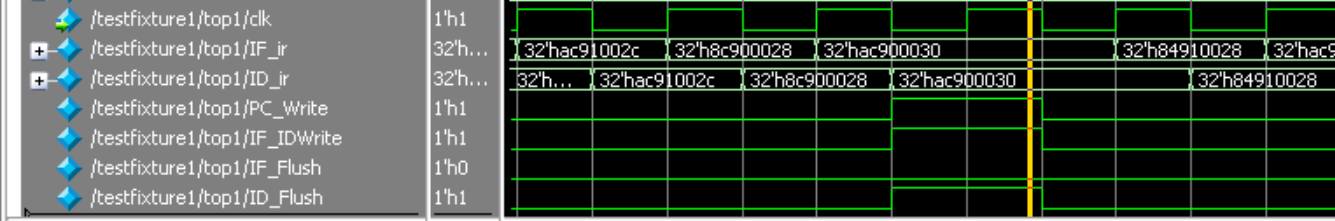
src1 forwarded from M to EX when (2) in EX

J-type:



Description:

1. 0c100033 //jal 0x004000cc [fun1]
2. 00000000 //nop
3. Load Stall:



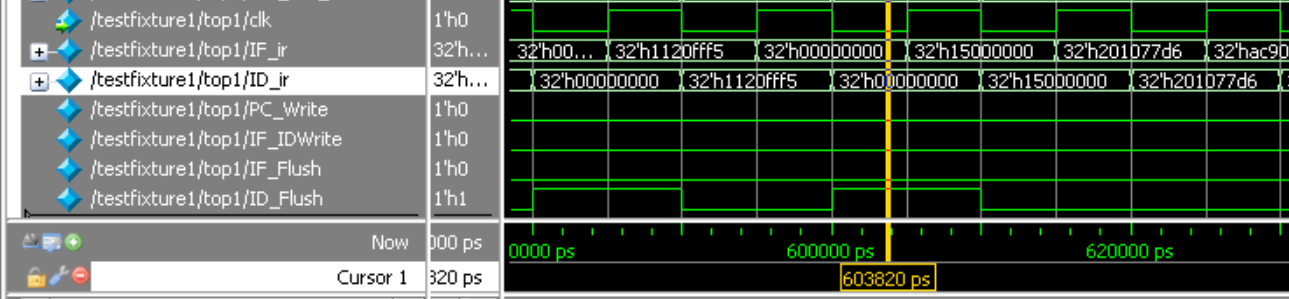
Description:

1. ac91002c //sw $17, 44($4) ; 35: sw $s1, 44($a0)
2. 8c900028 //lw $16, 40($4) ; 37: lw $s0, 40($a0) # $s0 = 65535
3. ac900030 //sw $16, 48($4) ; 38: sw $s0, 48($a0)

(2)=>(1) RAW (memory)

detected and stalled when (2) in ID, (1) in EX

1. Branch Delay (& Flush):



Description:

(a1) 1120fff5 //fun2: beq $9, $0, -44 [fun3-0x00400100] branch to fun3

(a2) 00000000 //nop ; 70: nop

(b1) 15000000 //fun3: bne $8, $0, 0 [fun4-0x004000d8]; Jump to fun4

(b2) 201077d6 //fun4: addi $16, $0, 30678; 59: addi $s0, $0, 30678 # $s0 = 0 + 30678

(b3) ac900040 //sw $16, 64($4) ; 60: sw $s0, 64($a0) # store 30678 to memory

Flush ID when (a1) branched in EX

If you CPU data is from Figure 5 in the home, show your CPU datapath and explain why you want to design your CPU this way.

Reason:

**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)

這次作業大部分只是把模組的線分區重接而已，並不算太難。不過在除錯時因為線太多所以有點麻煩。