計算機組織Final Project

第17組

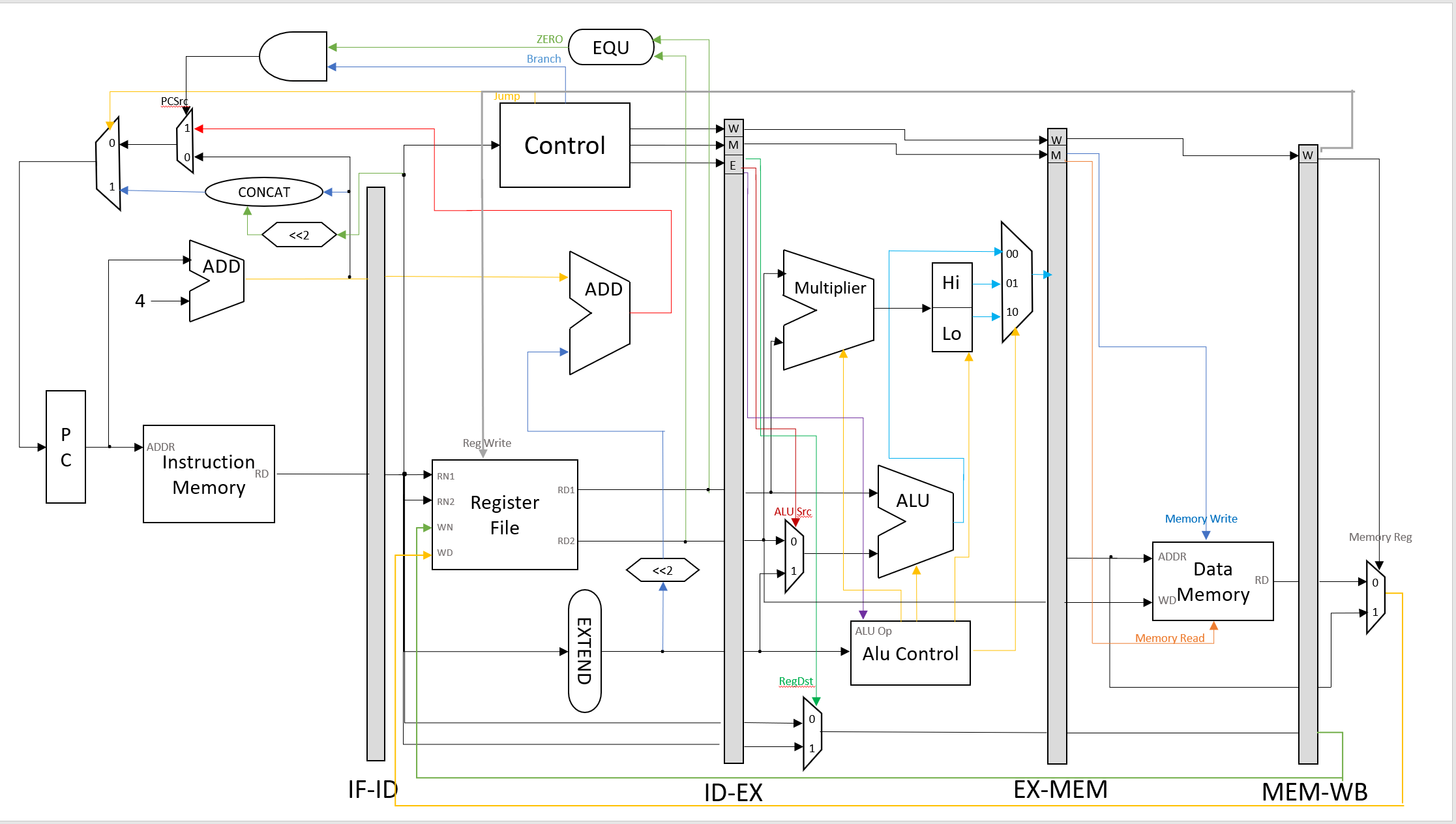
11027211 林芷楡

11027212 黃建閎

11027222 黃彥霖

11027253 盧宸揚

Datapath：



重點說明：

我們總共劃分了5個區域：

第一個區域裡面放了PC，ADD，Instruction Memory，PC的MUX，Jump的MUX，Branch的and gate，IF/ID register，PC的MUX是用來判斷要做branch還是傳入PC+4，Jump的MUX是用來判斷要不要做jump，Branch的and gate是用來判斷第二區域的zero值是不是為1。

第二個區域裡面放了Register file，Branch equ，Signal extend，ADD，Control pipelined，ID/EX register，在這部分要判斷程式是哪個並做出他相應的動作，Branch equ是用來判斷RD1跟RD2的值有沒有相等，在判斷BEQ指令時會使用到。

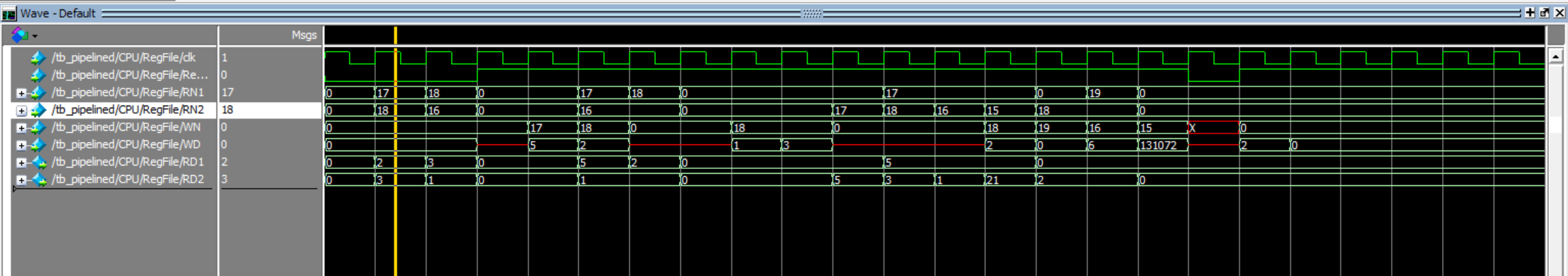
第三個區域放了ALU，Multiplier，HiLo，ALU Control，EX/MEM register，而在第三個區域內我們有在ALU前面放置一個2 to 1 的MUX去判斷要放RD2還是extend完的值，還有一個2 to 1 的MUX去選擇rd跟rt，因為要做MULTU和MADDU所以多給了HiLo一個控制去判斷做哪一個動作。

第四個區域裡面放了Data Memory，MEM/WB register

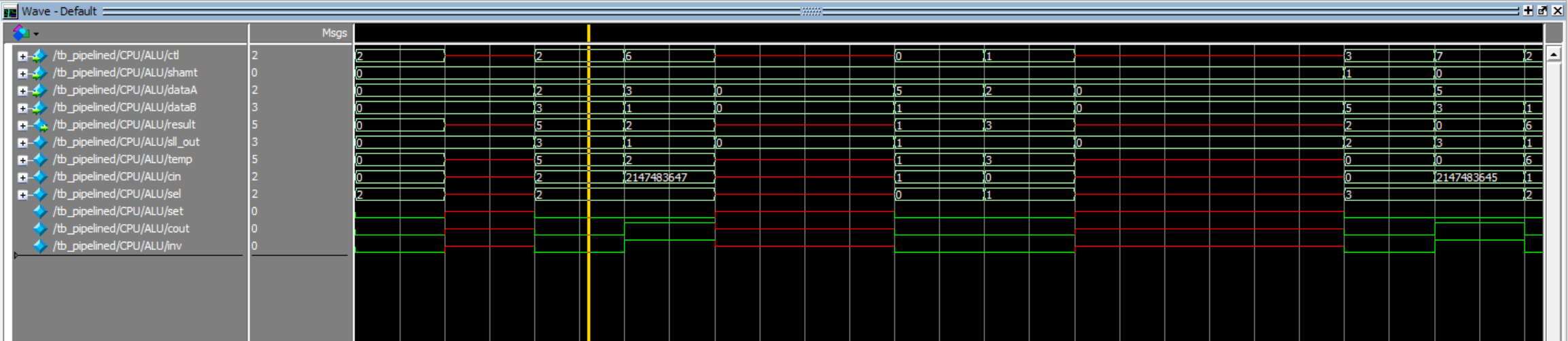
第五個區域裡面放了一個2 to 1 的MUX，用來判斷要回傳ADDR還是RD\_MEM/WB的值給regiser file，並把要write back的東西寫回去。

驗證結果與Waveform圖：

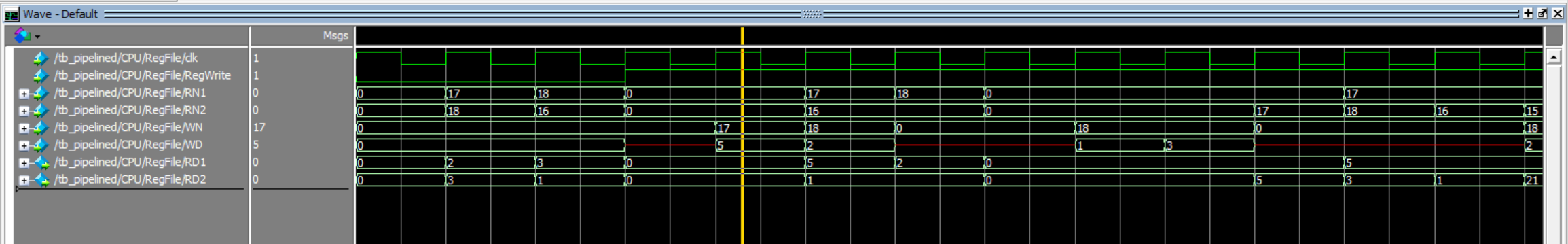
**add $s1, s2, s1**



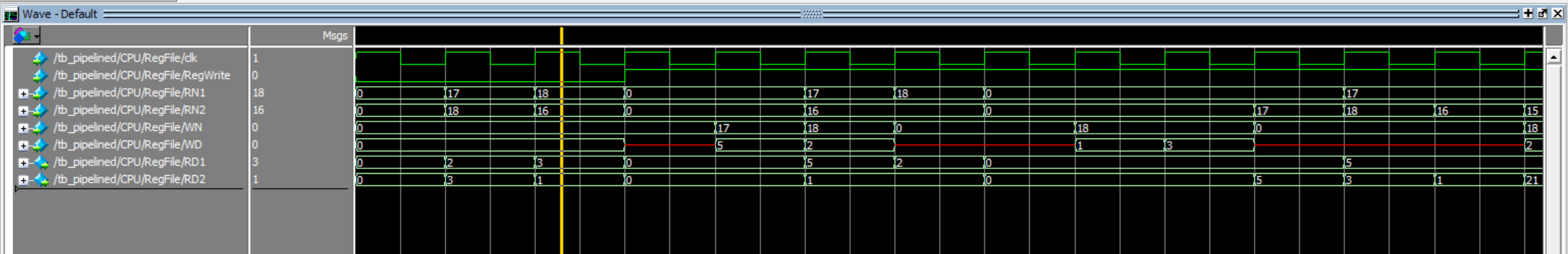
在regiser裡面s1(17)是2，s2(18)是3



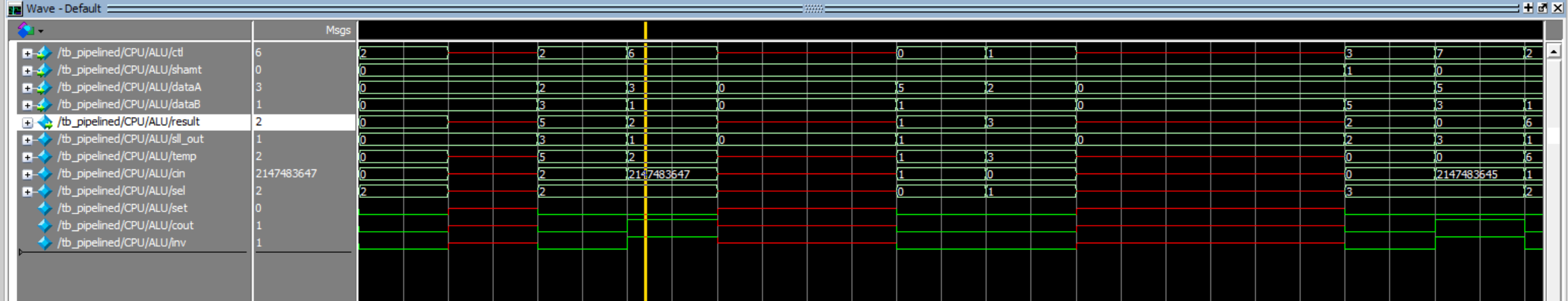
所以2 + 3 = 5正確，並且寫入s1(17)



**sub $s2, $s0, $s2**

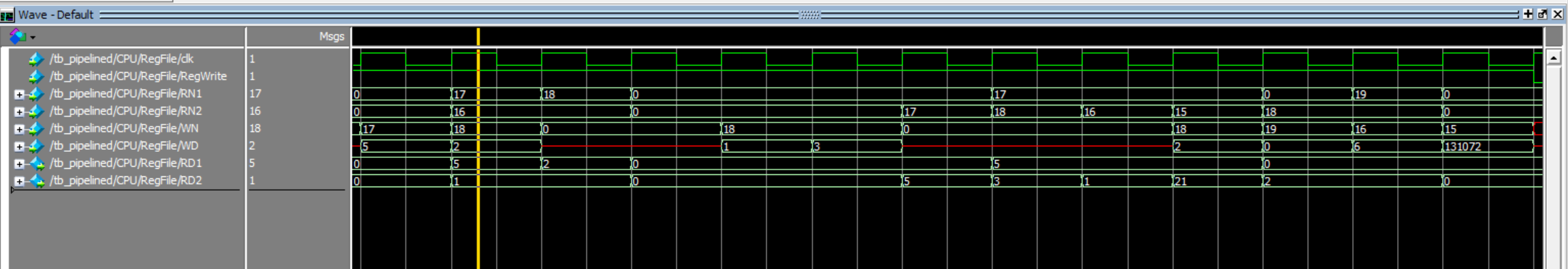


在regiser裡面s2(18)是3，s0(16)是1



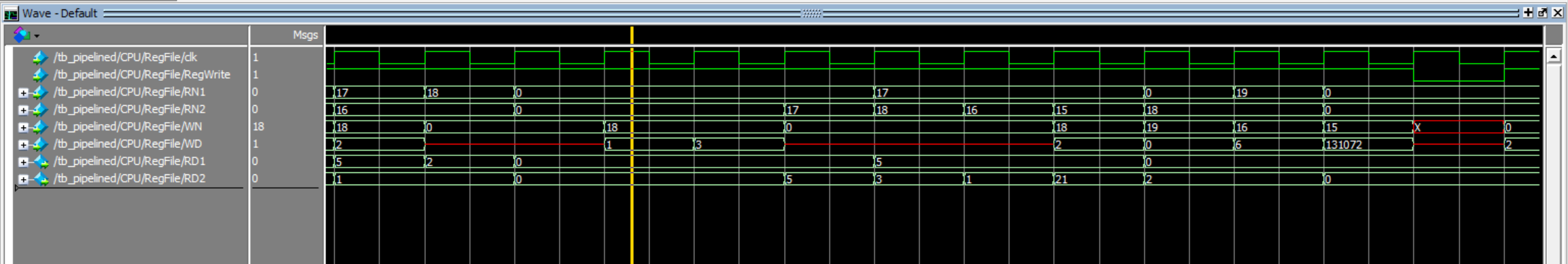
所以3 – 1 = 2正確，並寫入s2(18)

**and $s1, $s0, $s2**

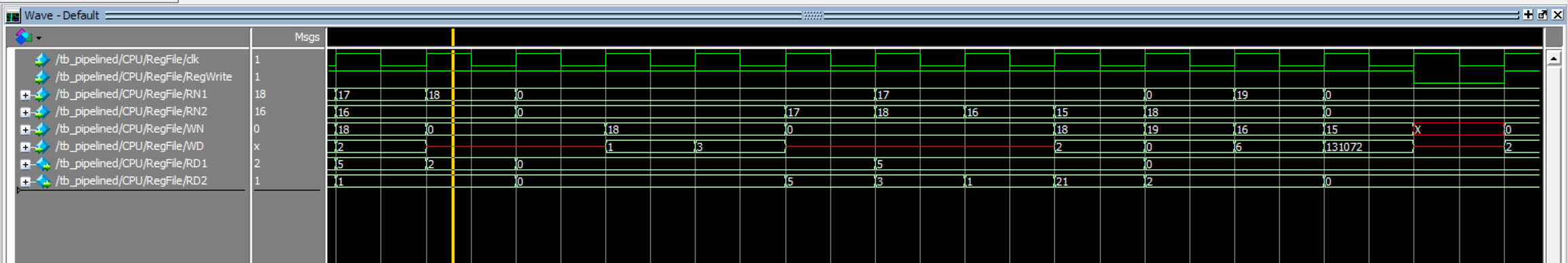


在regiser裡面s1(17)是5，s0(16)是1

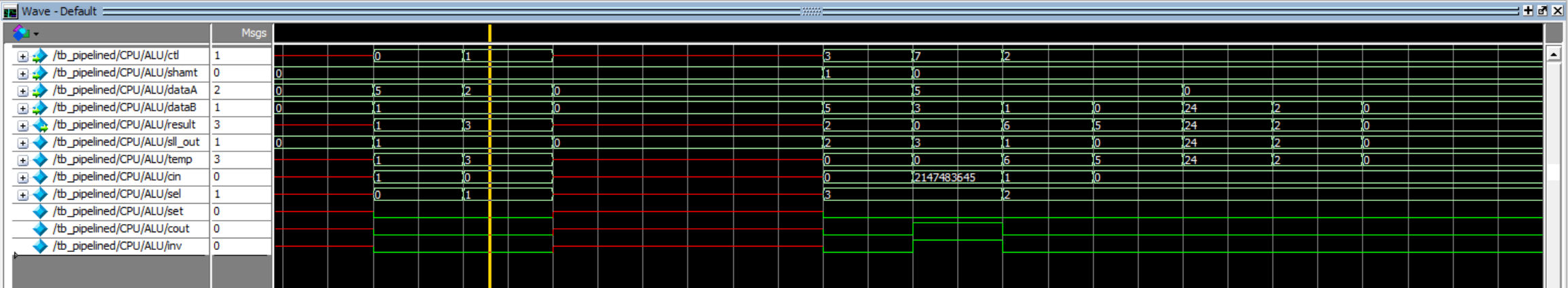
所以5(00101) & 1(00001) = 1(00001)，所以是1正確，並寫入s2(18)



**or $s2, $s0, $s2**

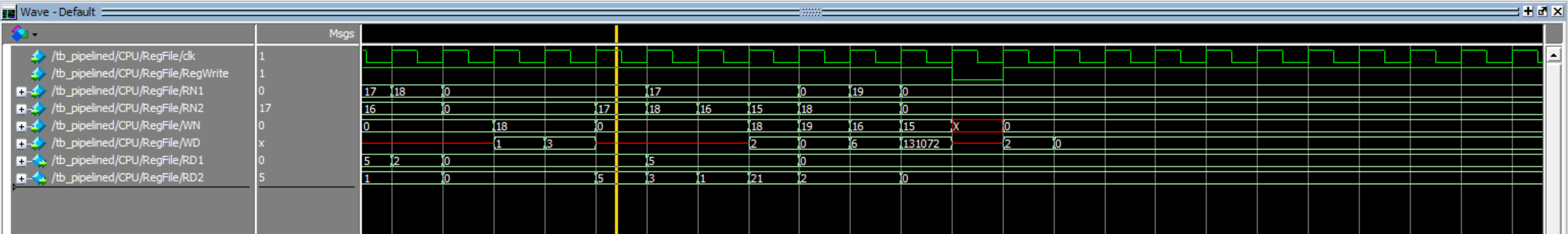


在register裡面s2(18)是2，s0(16)是1

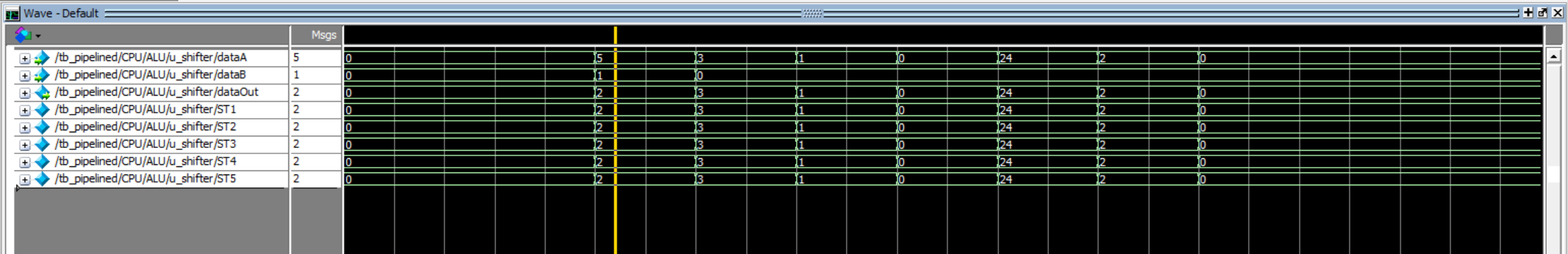


所以2(00010) | 1(00001) = 3(00011)，所以是3正確，並寫入s2(18)

**srl $s1, $s2, 1**

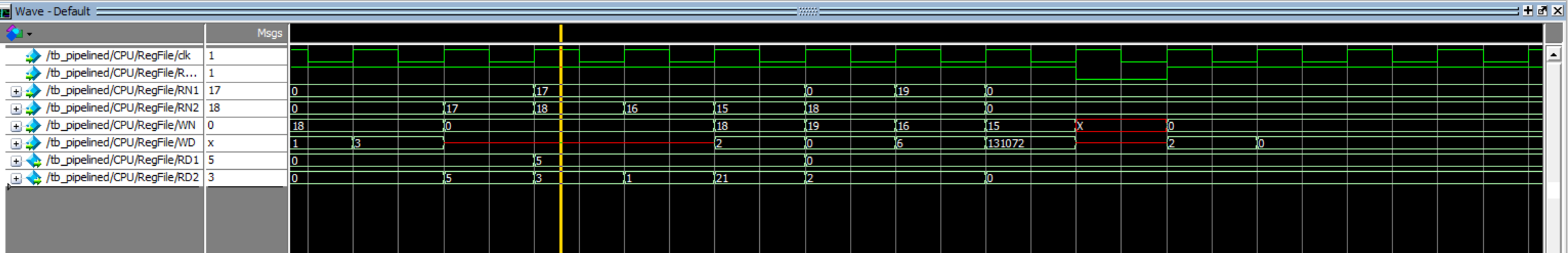
****

在register裡面s1(17)是5

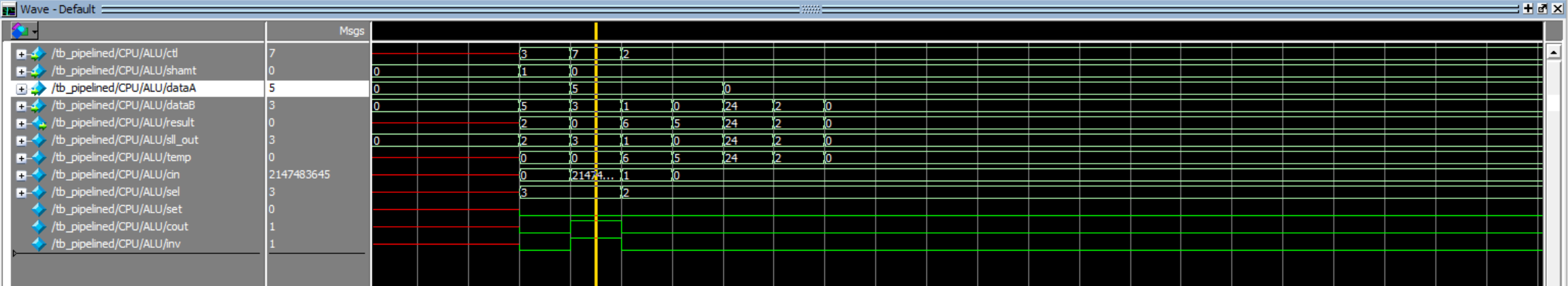
****

5 shift 1 bit 是 2 正確，並寫入s2(18)

**slt $s1, $s2, $s3**

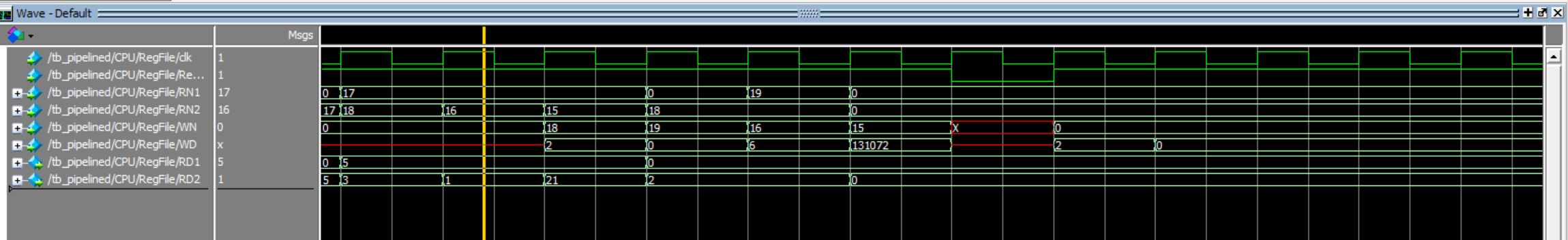


在register裡面s1(17)是5，s2(18)是3

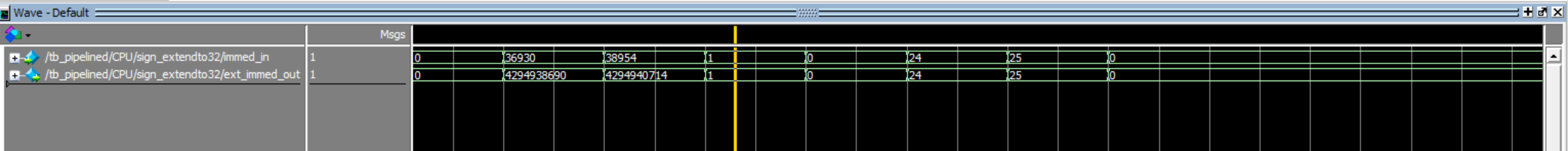


因為5 > 3，所以輸出結果是0正確，並寫入s3(19)

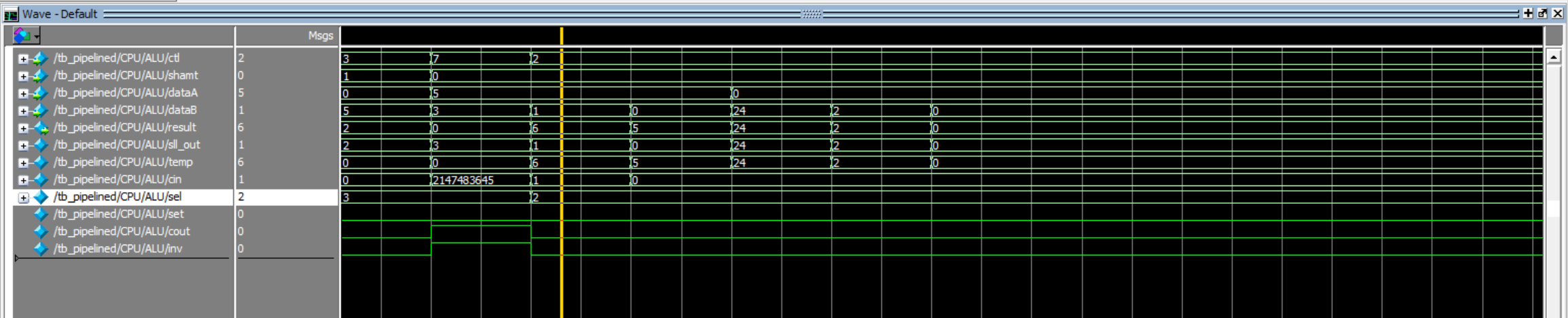
**addiu $s1, $s0, 1**

****

在register裡面s1(17)是5，s0(16)是1

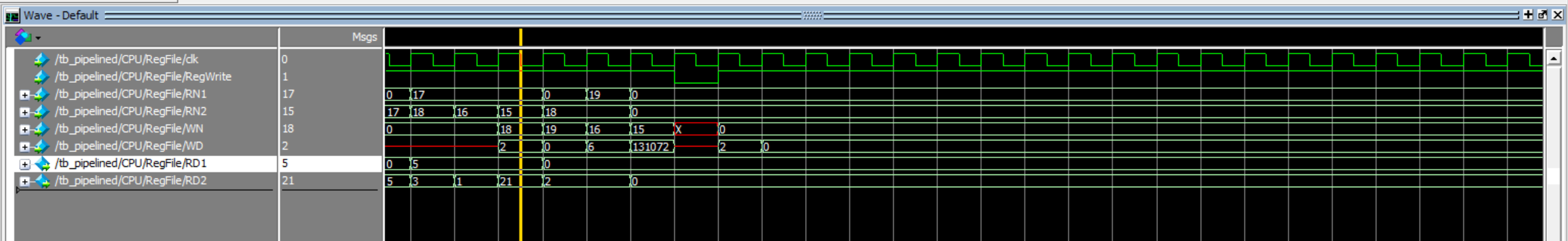
****

1先做有號數擴充還是1

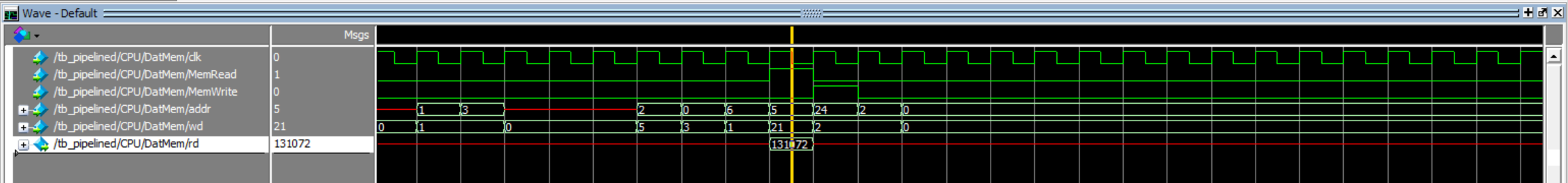


所以5 + 1 = 6 正確，並寫入s0(16)

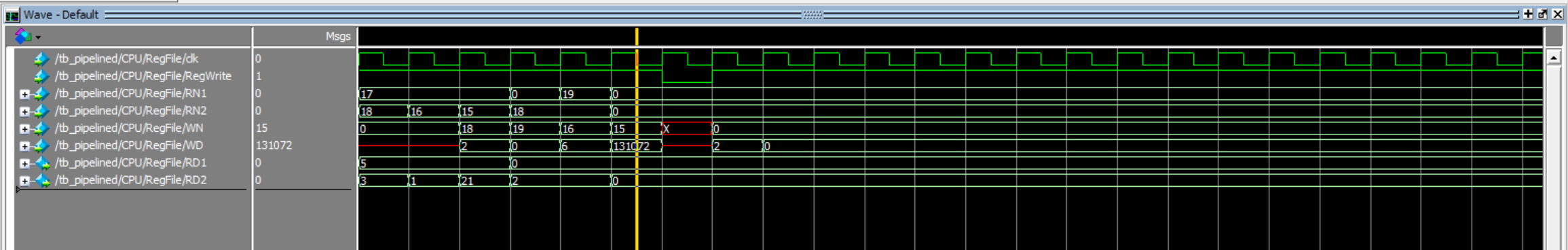
**lw $s1, $t7, 0**



在register裡面s1(17)是5，t7(15)是21

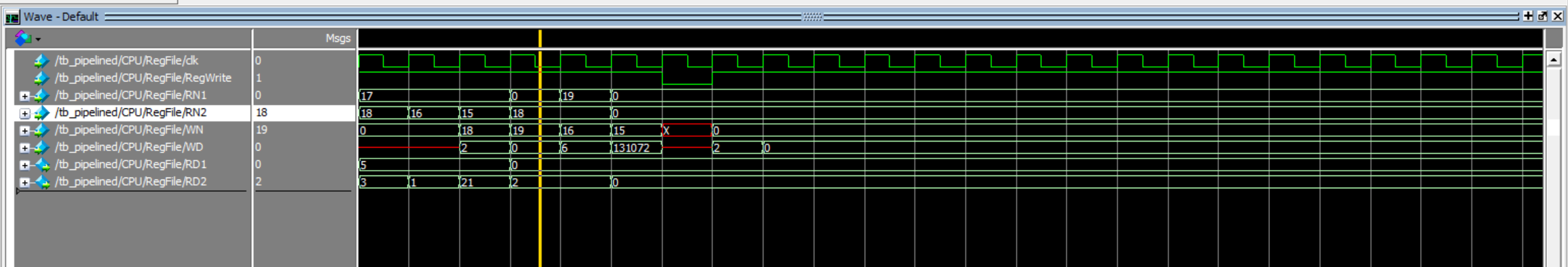


有成功讀到memory，並回傳rd

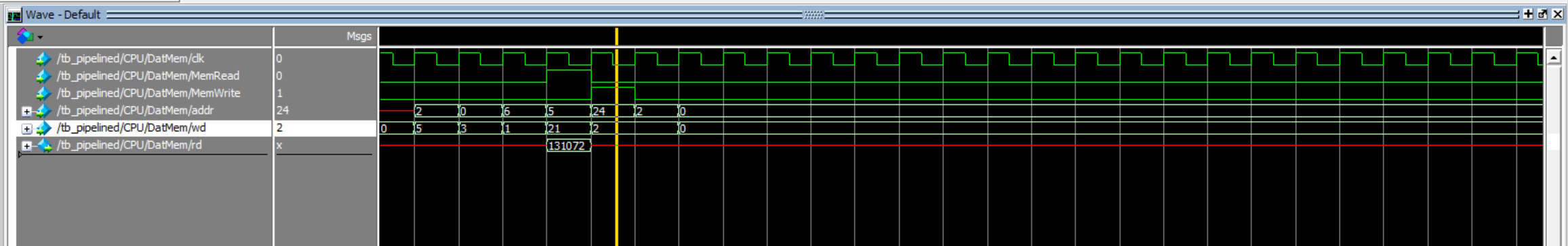


回傳到register

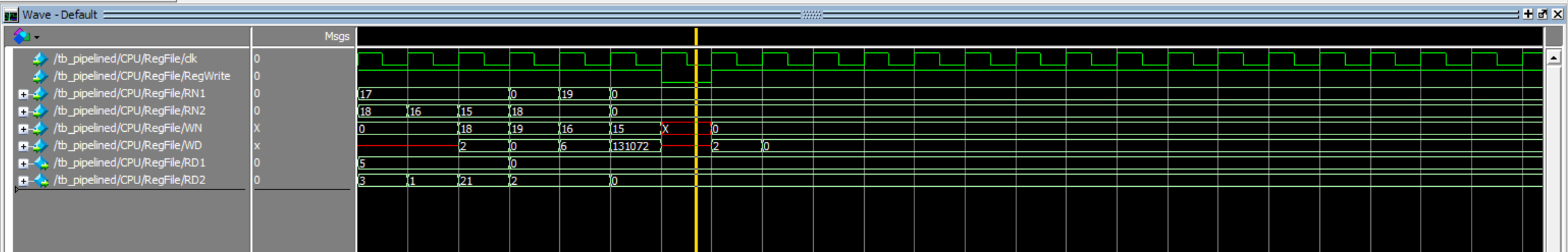
**sw $zero, $s2, 24**



在register裡面zero(0)是0，s2(18)是2

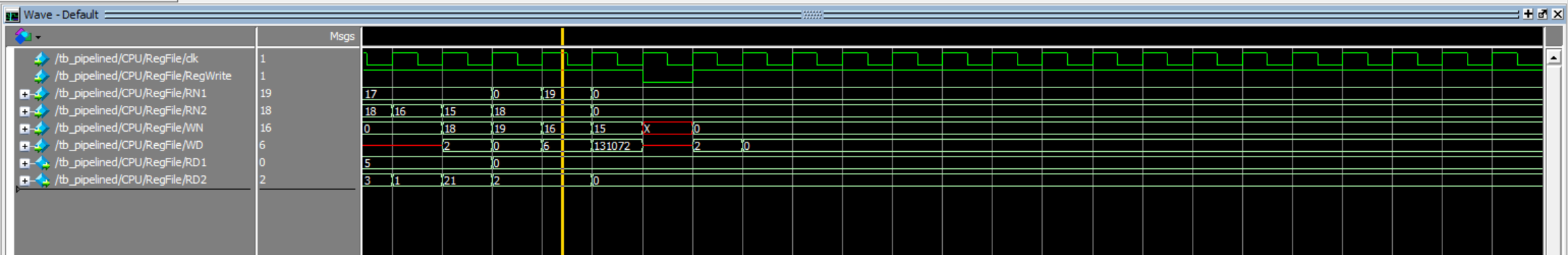


成功寫入memory

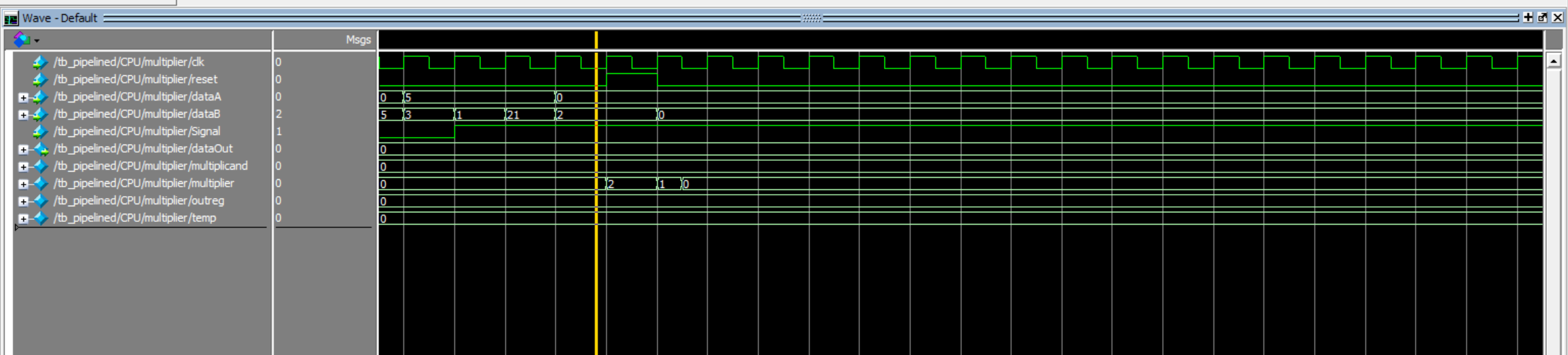


並且register不會寫入

**multu $s3, $s2**

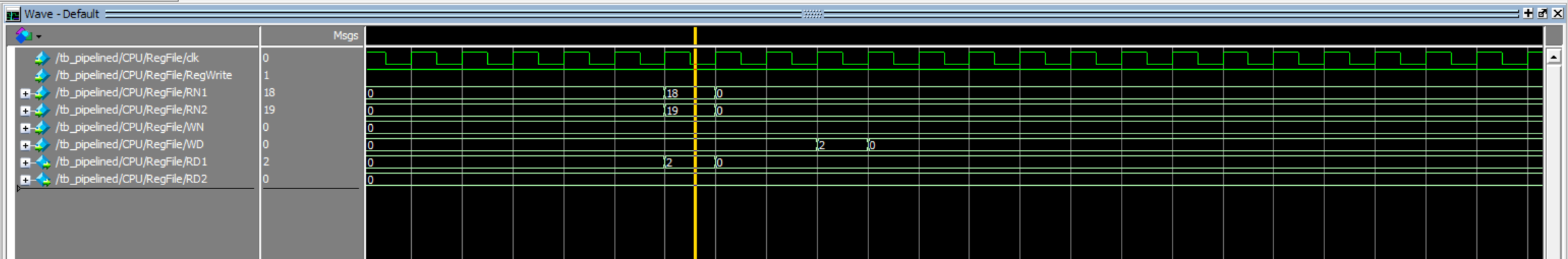


在register裡面s3(19)是0，s2(18)是2

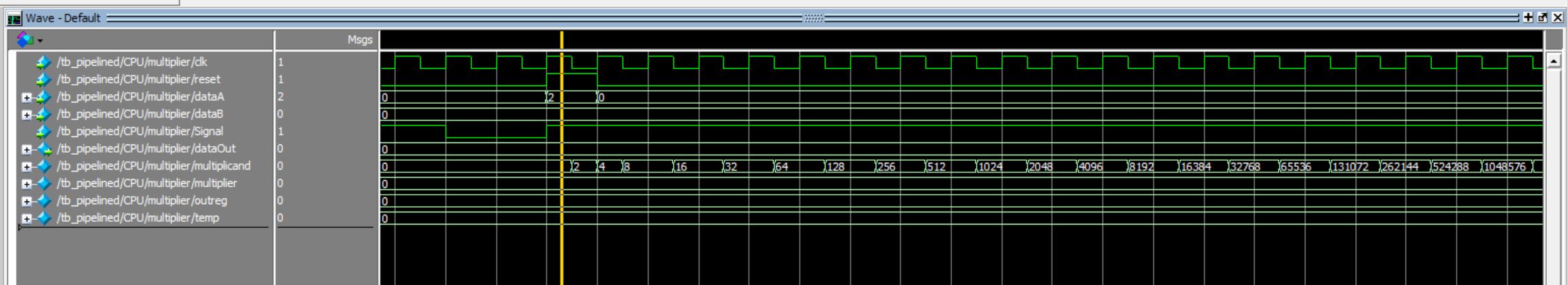


0 \* 2 = 0 正確

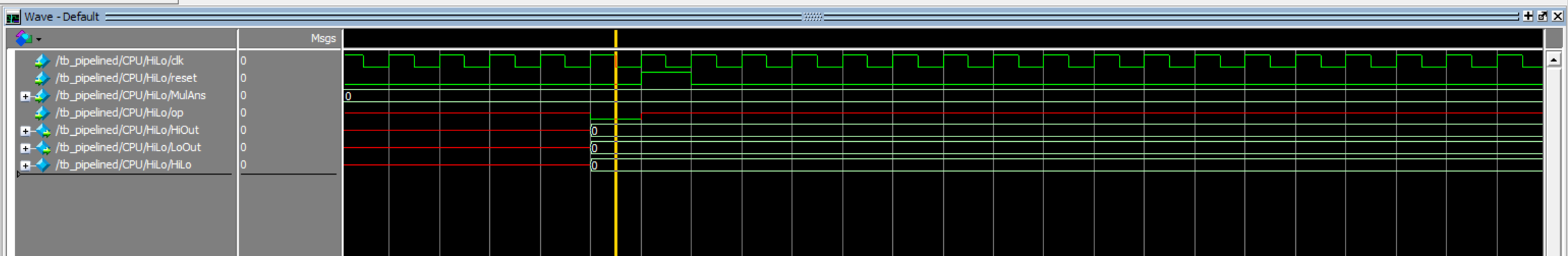
**maddu $s2, $s3**



在register裡面s2(18)是2，s3(19)是0

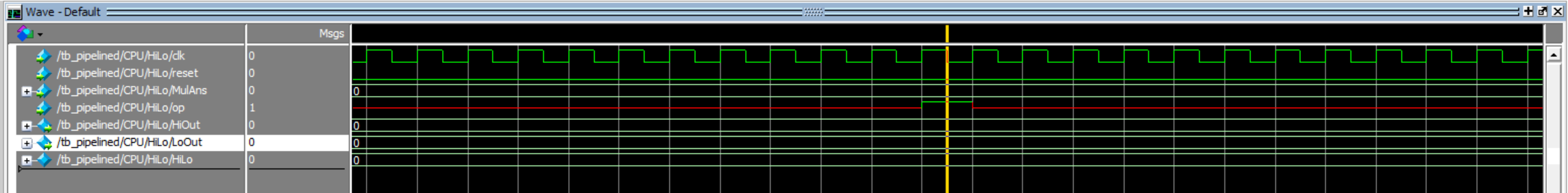


2 \* 1 = 2

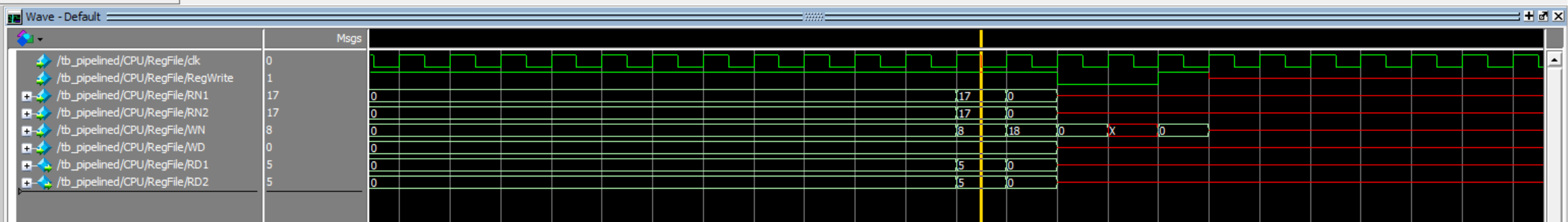


HiLo的輸出值都為0，所以2 + 0 = 2正確

**mfhi $s1**

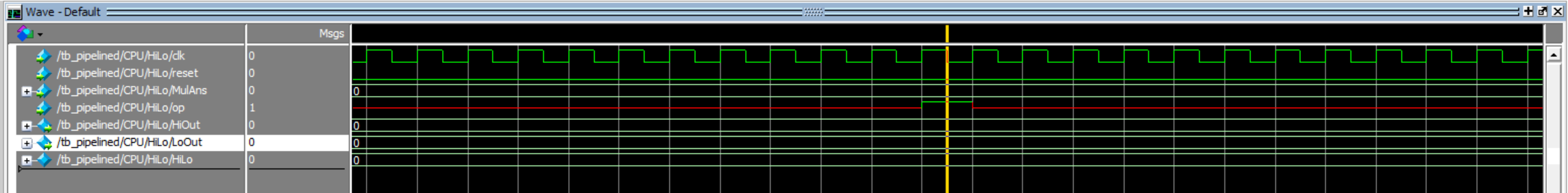


目前的Hi值為0

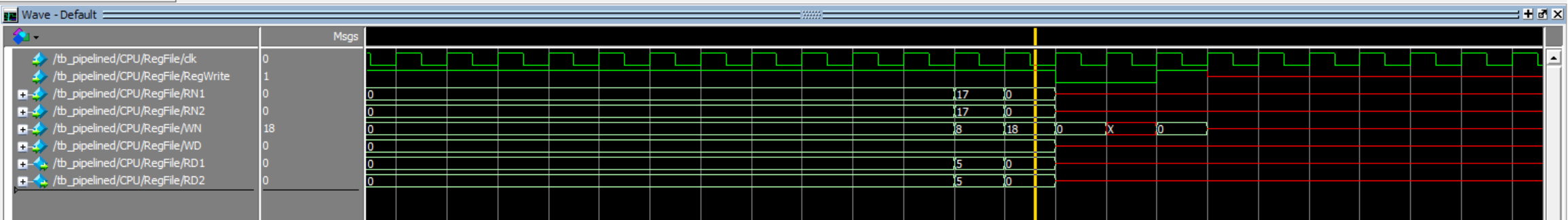


回傳到register並把Hi值寫入s1

**mflo $s2**

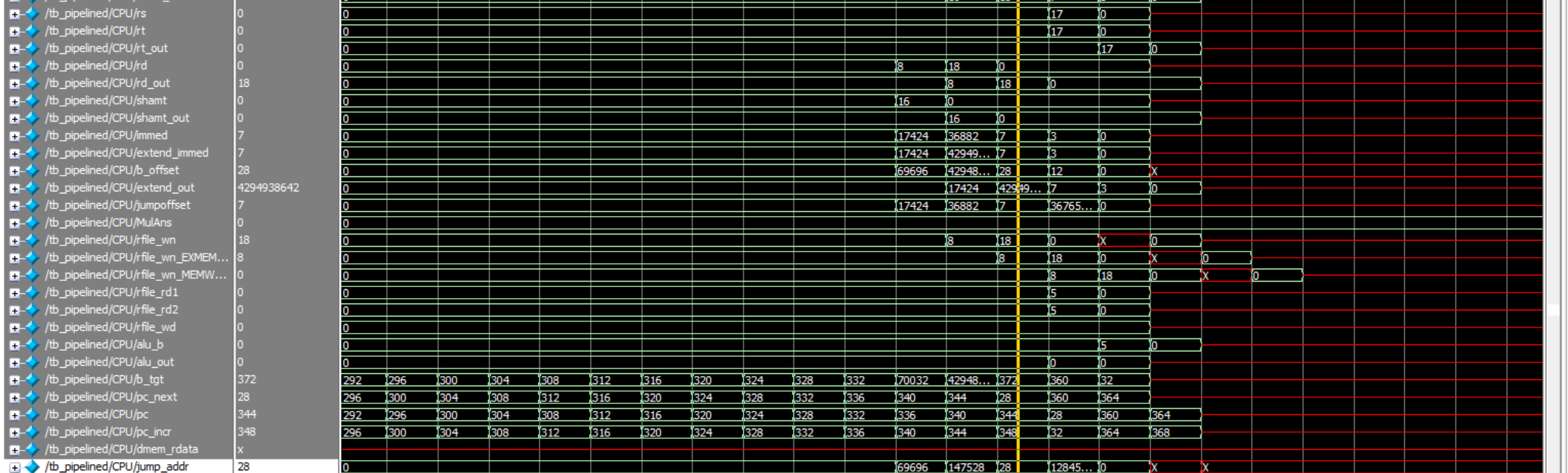


目前的Lo值為0



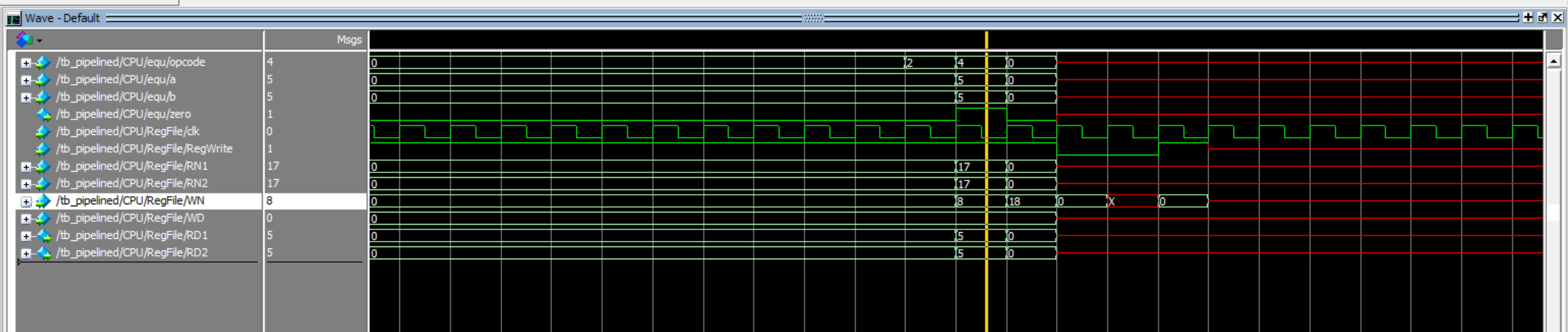
回傳到register並把Lo值寫入s2

**j 7**



Immed是7，最後jump輸出為28

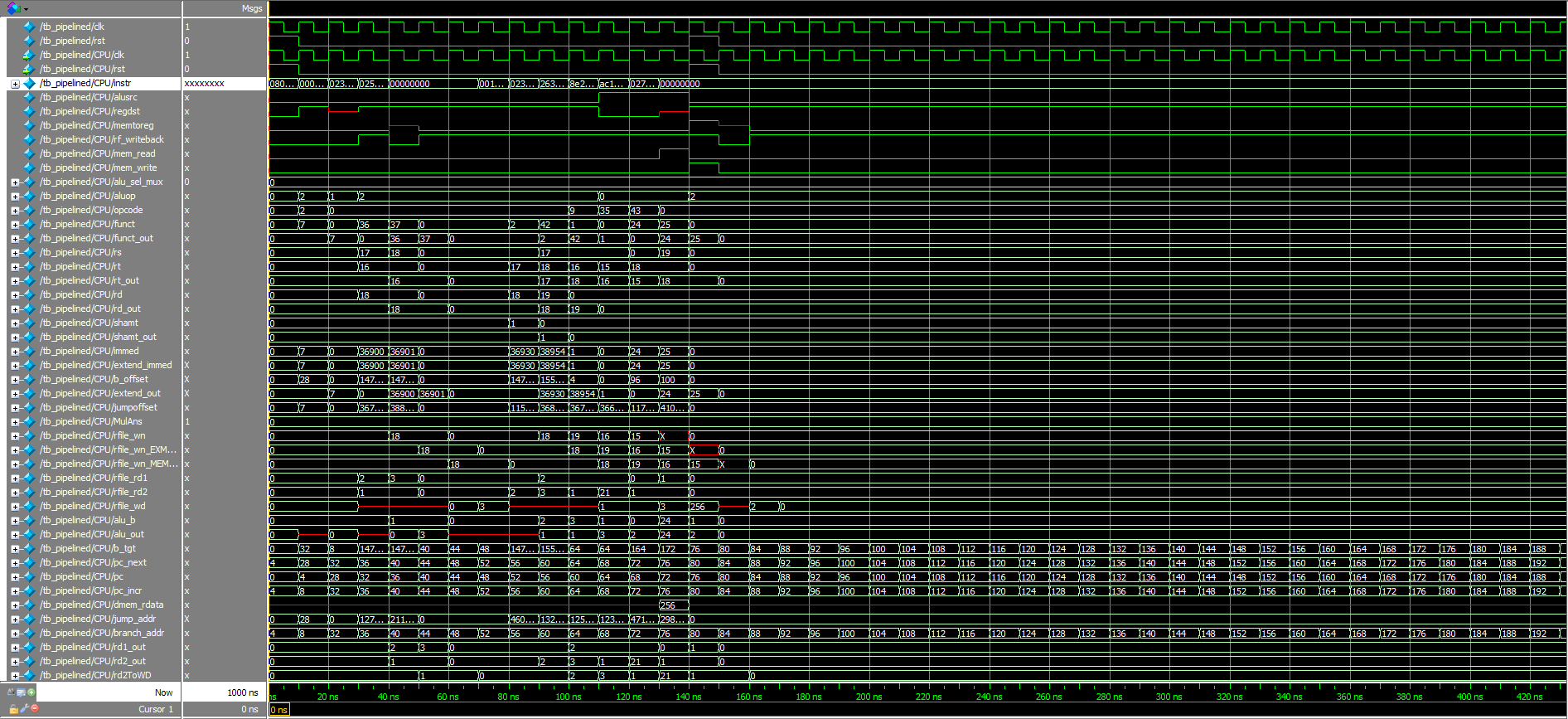
**beq $s1, $s2, 3**

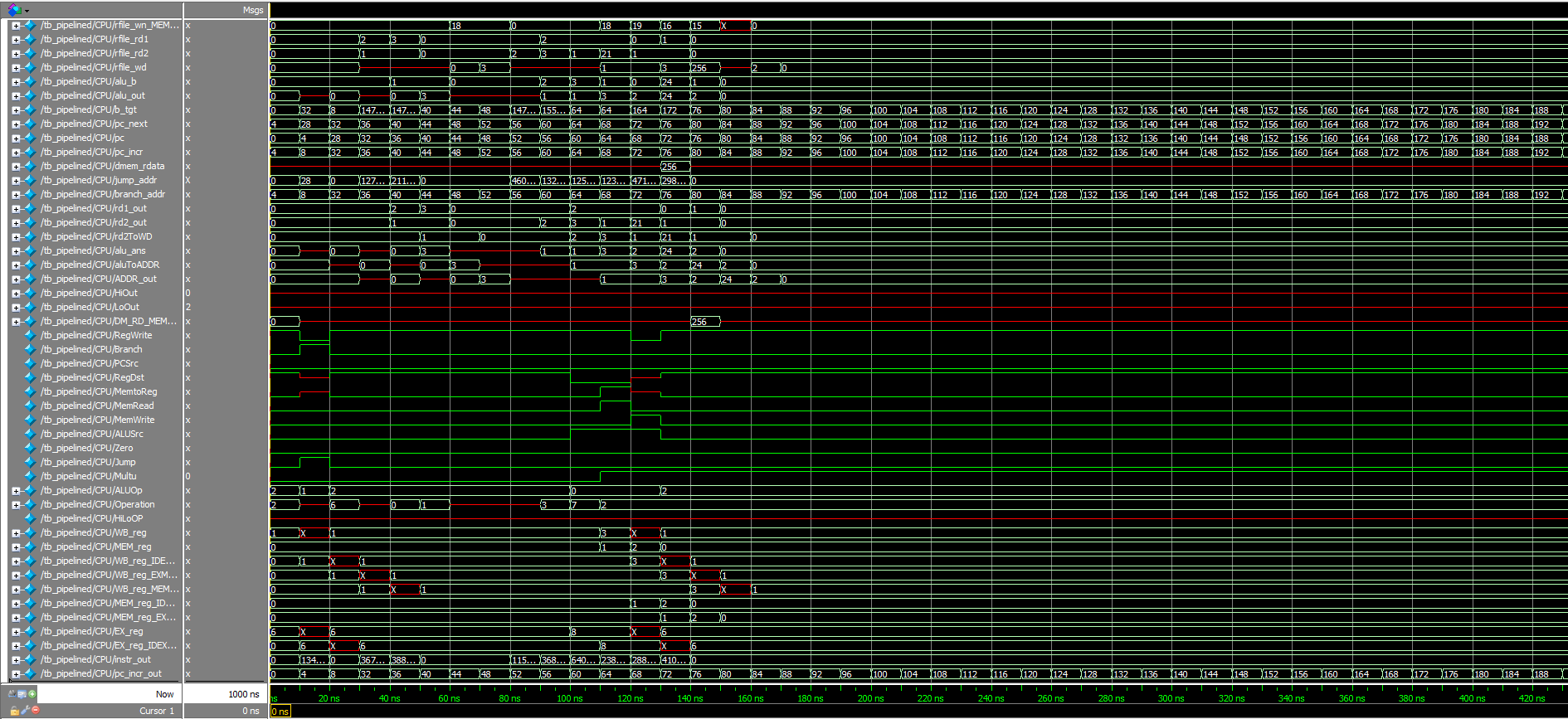


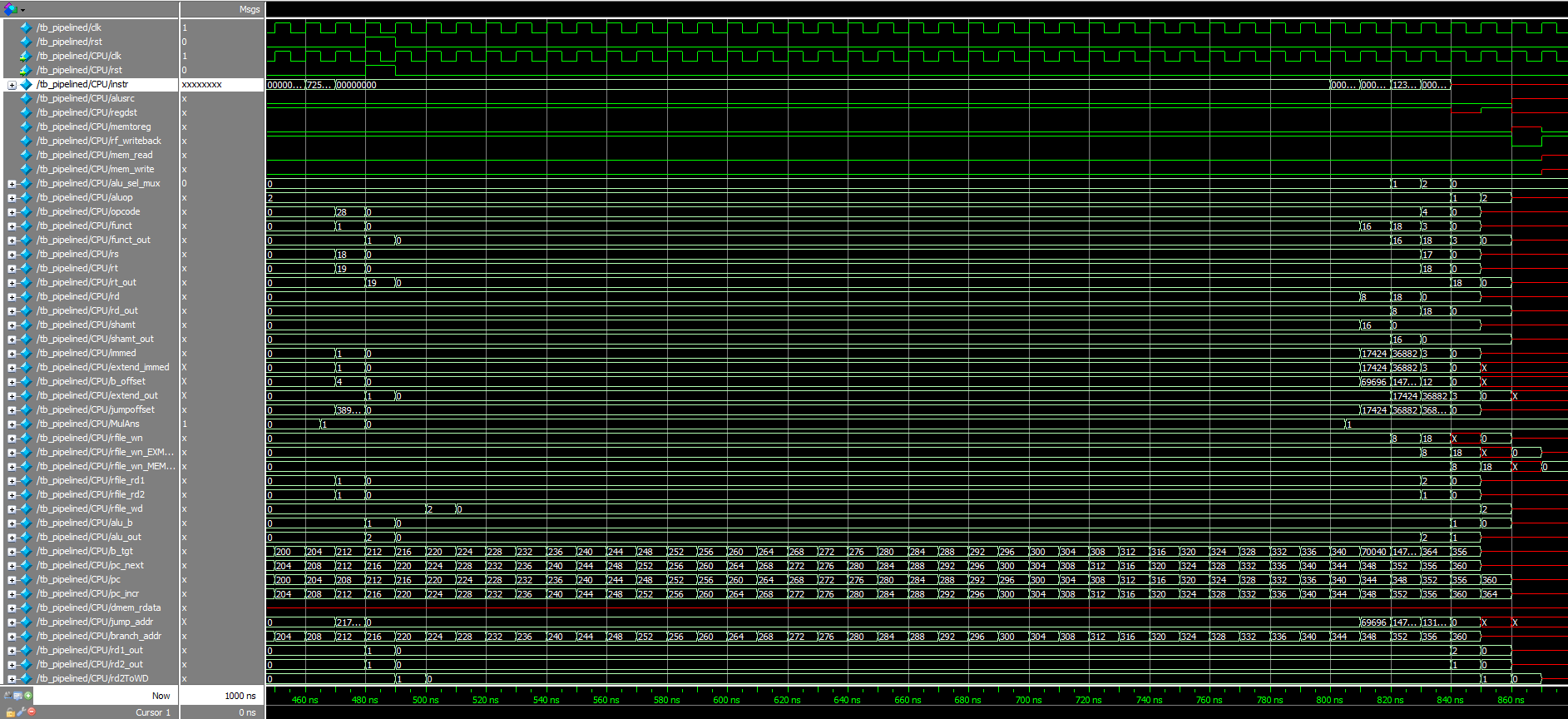
在register裡面s1是5，s2是5

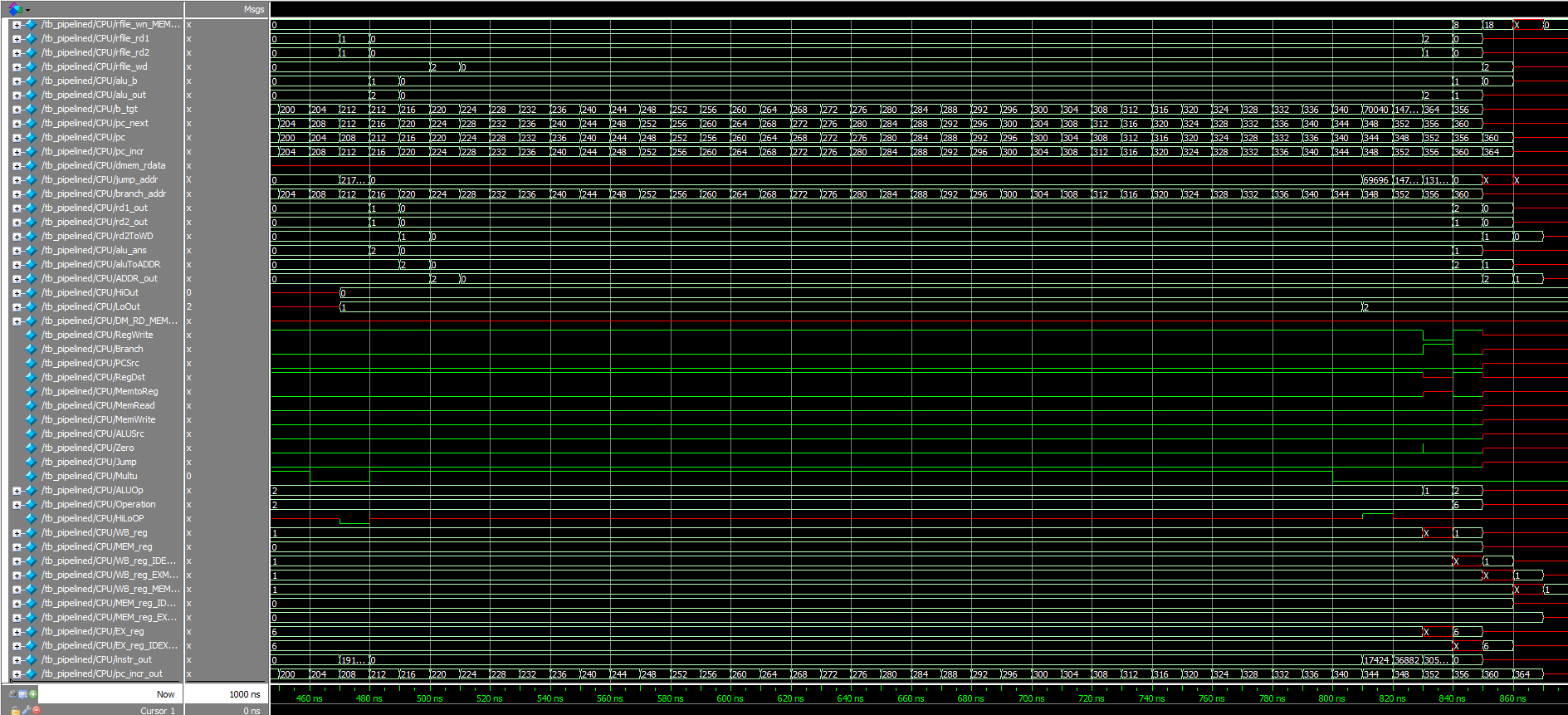
所以在做equ的時候回傳zero為1

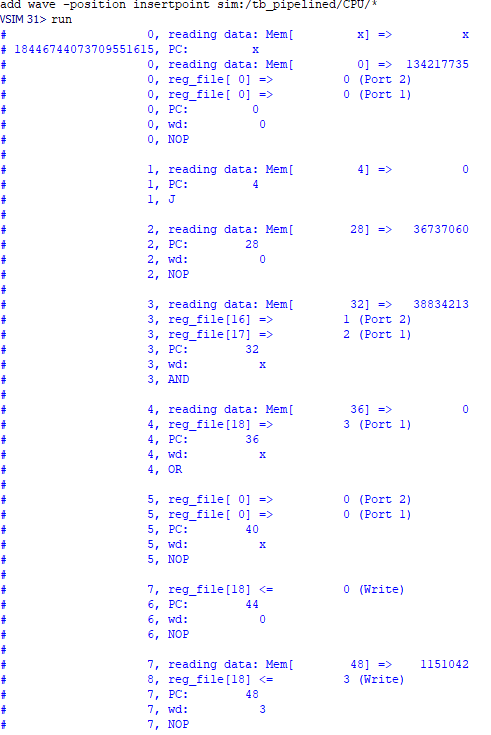
**全部的pileline輸出結果:**

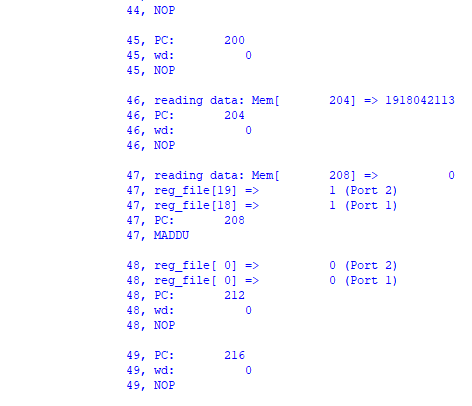


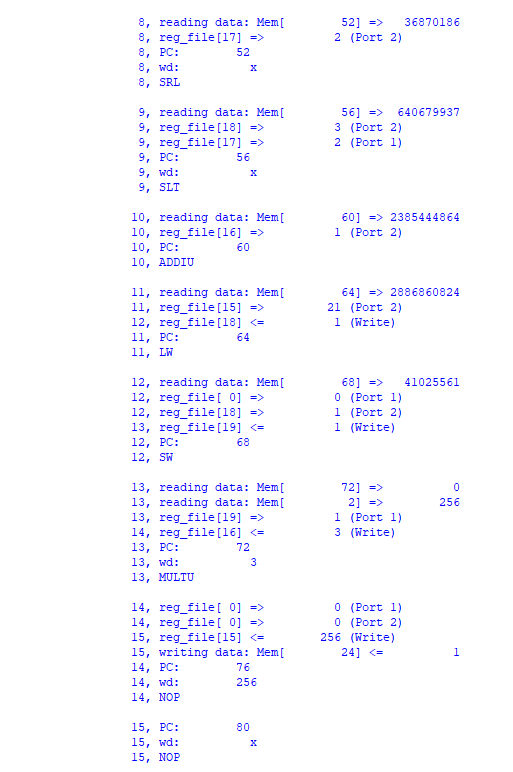
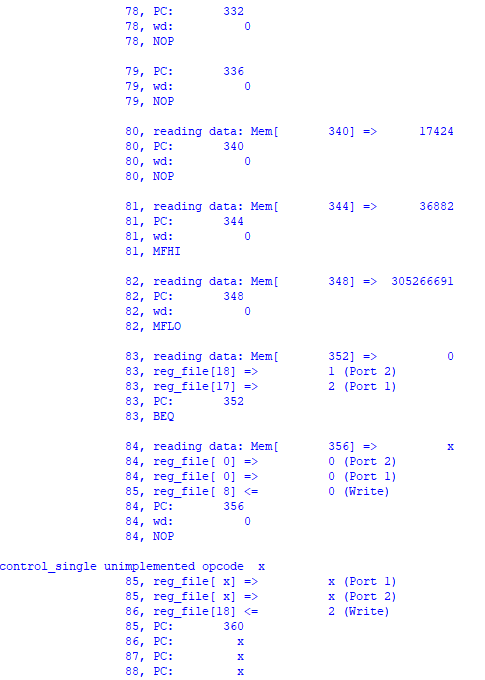












因為乘法後面都是NOP所以不放入

心得感想：

11027212 黃建閎：

在這次期末報告，我是負責寫報告的，在做報告的過程中也越來越熟悉整個single cycle的流程跟Verilog的語法，也對於modelsim更加的熟悉，這次也很感謝組員很認真的去完成作業，雖然我沒有參與到寫程式的地域，但看著室友每天做每天叫，甚至通宵到隔天早上10點才休息，讓我知道能跟這樣子的組員在一組的很棒，我之後也會拿出我的誠意去回饋給他們。

11027211 林芷楡：

這次的final project 複雜程度有種超出我能力範圍的感覺，上課都上過也去了課業守護但我怎麼就是不太懂，也因為這樣所以這次工作就負責畫datapath 等文書工作，整個做project 的過程中就都默默地聽著隊友們討論，也慢慢越來越清楚，真的非常感謝跟佩服他們，最後是到畫完整版的datapath 時，因為要弄清楚每條接線所以才完全了解整個過程。

11027222 黃彥霖：

這是期末作業相比期中難度實在是大很多，花了超級多個夜晚才寫完，還好我的組員很認真，一開始從Single cycle開始看資料怎麼移動的，再想辦法用pipeline的方式切成五個部分，這過程實在是非常折騰，對Verilog的語法沒有很深入的研究，常常接出來的電路不是我們所想的那樣，加上對任何指令的不熟悉，要找出的bug非常多，每個指令都要一個clock慢慢看，慢慢找出錯誤的地方，也要處理好各種hazards的問題。從這次的project的情況來看，有一個好的設計圖和流程，一定有辦法提高更多的效率，有實際的圖在那邊真的好做很多。

11027253 盧宸揚：

這次我是負責撰寫程式，我的計畫是先了解single-cycle CPU與pipelined CPU的區別，熟悉後發覺pipelined CPU因為同時進行多道指令所以容易造成hazard，在debug的部分花費很多時間看波型與數值。我認為事先將圖畫好很有利於程式的撰寫，在接線的步驟看著圖接基本上就不會有太大的問題，會比較有效率。

分工項目：

寫程式：黃彥霖、盧宸揚

Debug：黃彥霖、盧宸揚

做報告：黃建閎、林芷楡

畫Datapath：黃建閎、林芷楡