

計算機組織 Final Project

第 17 組

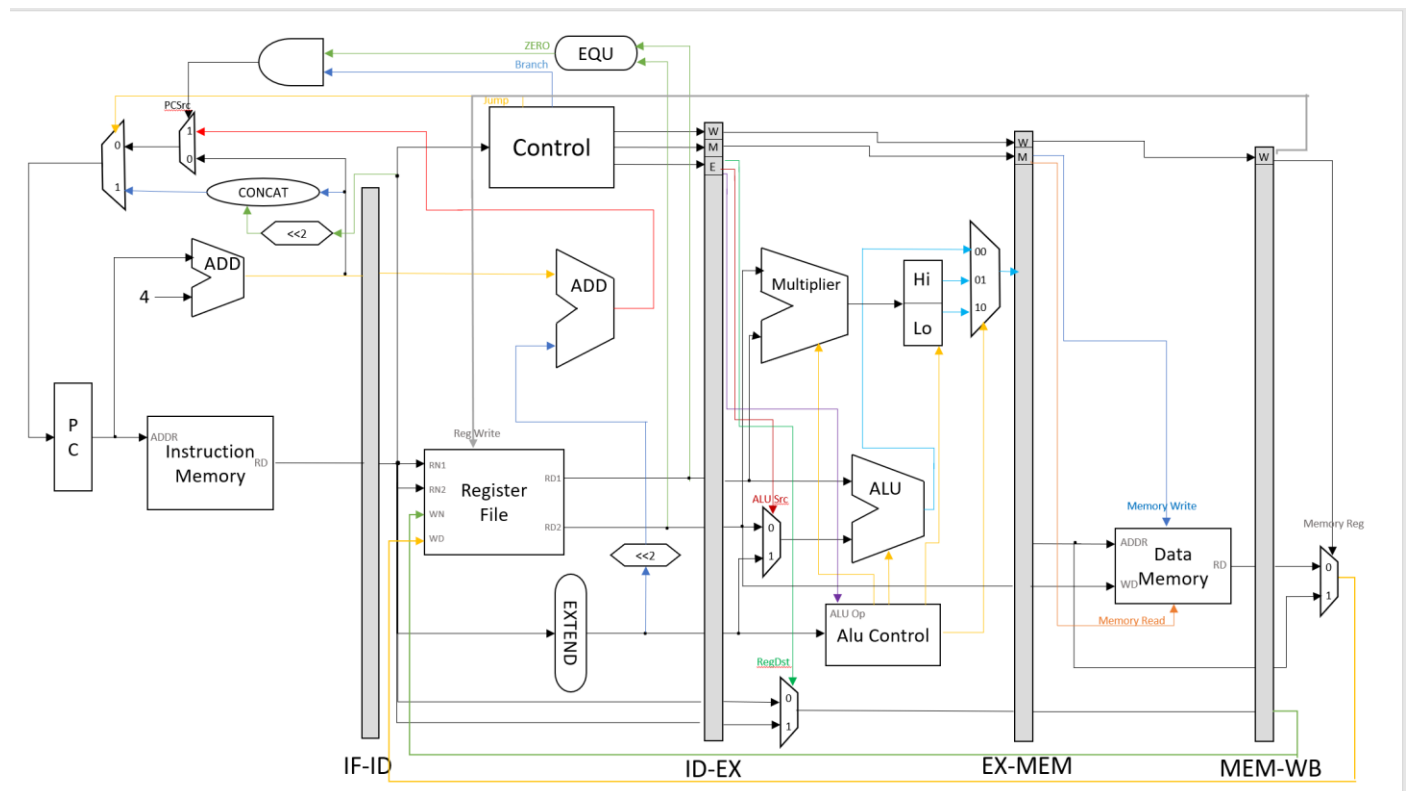
11027211 林芷榆

11027212 黃建閔

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。

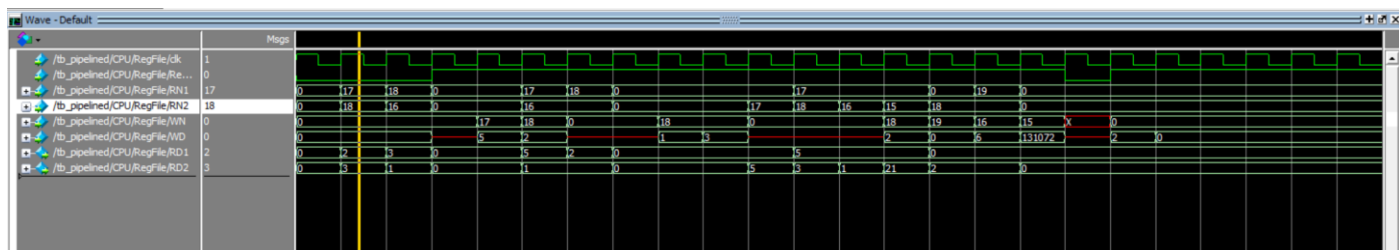
第三個區域放了 ALU，Multiplier，HiLo，ALU Control，EX/MEM register，而在第三個區域內我們有在 ALU 前面放置一個 2 to 1 的 MUX 去判斷要放 RD2 還是 extend 完的值，還有一個 2 to 1 的 MUX 去選擇 rd 跟 rt，因為要做 MULTU 和 MADDU 所以多給了 HiLo 一個控制去判斷做哪一個動作。

第四個區域裡面放了 Data Memory，MEM/WB register

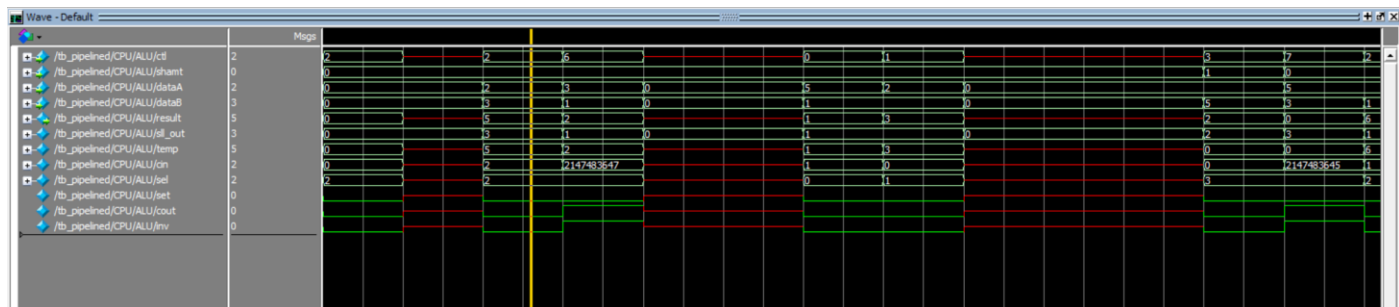
第五個區域裡面放了一個 2 to 1 的 MUX，用來判斷要回傳 ADDR 還是 RD_MEM/WB 的值給 register file，並把要 write back 的東西寫回去。

驗證結果與 Waveform 圖：

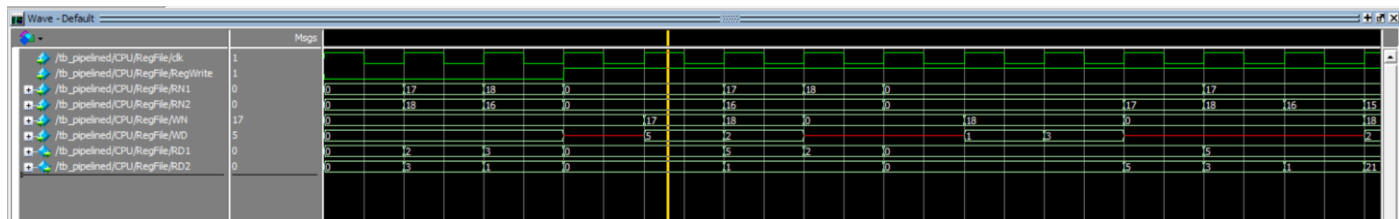
add \$s1, \$s2, \$s1



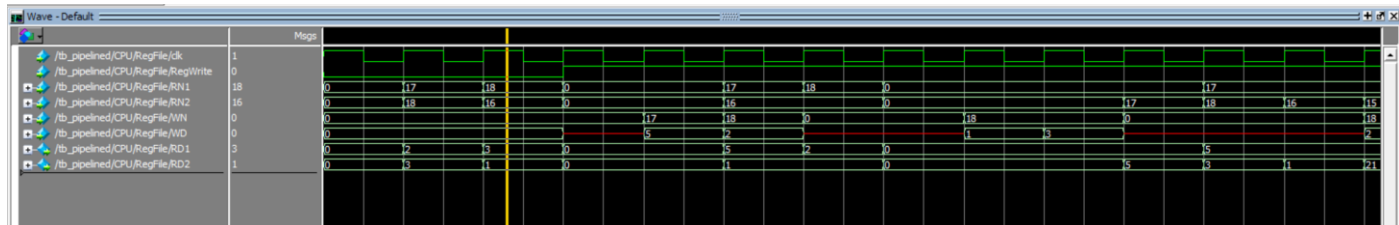
在 regiser 裡面 s1(17)是 2，s2(18)是 3



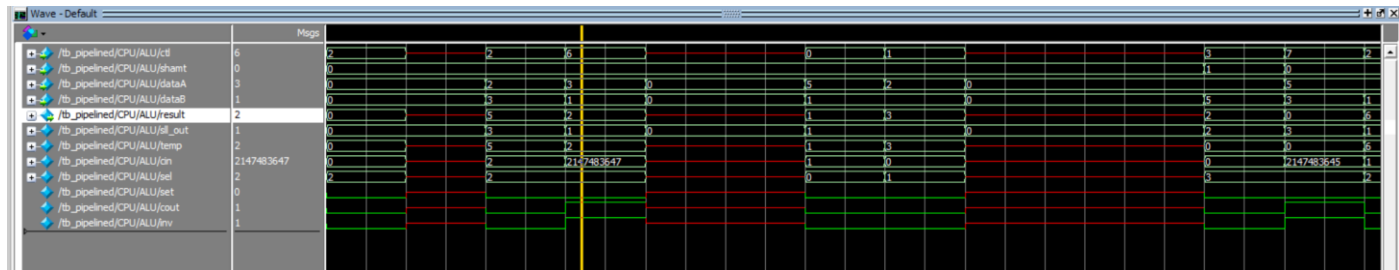
所以 $2 + 3 = 5$ 正確，並且寫入 s1(17)



sub \$s2, \$s0, \$s2

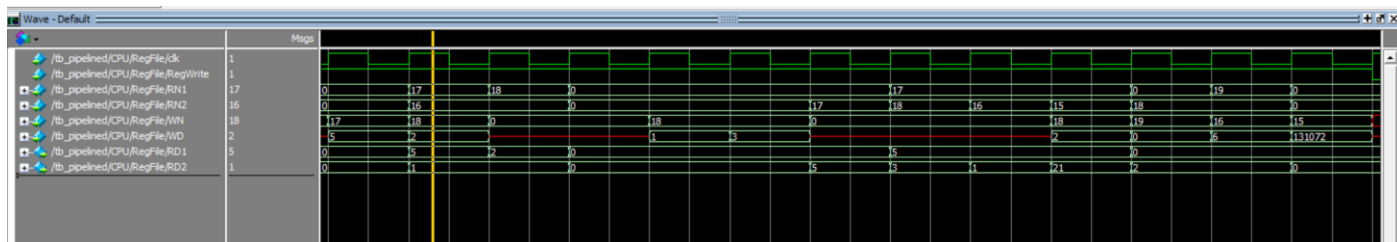


在 regiser 裡面 s2(18)是 3，s0(16)是 1

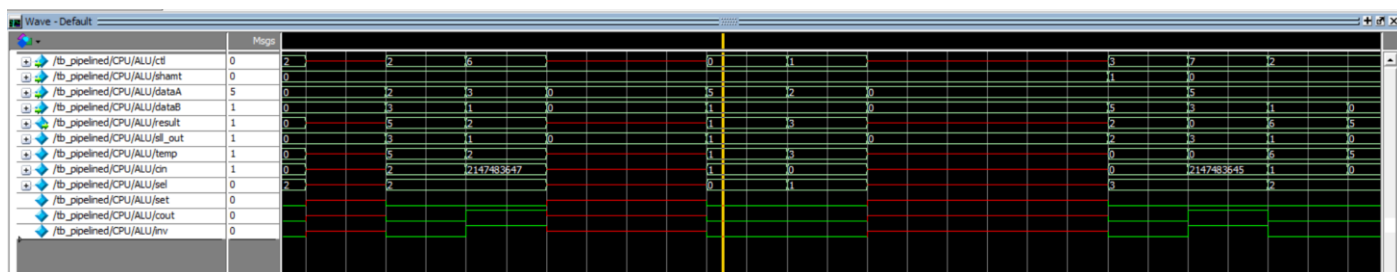


所以 $3 - 1 = 2$ 正確，並寫入 s2(18)

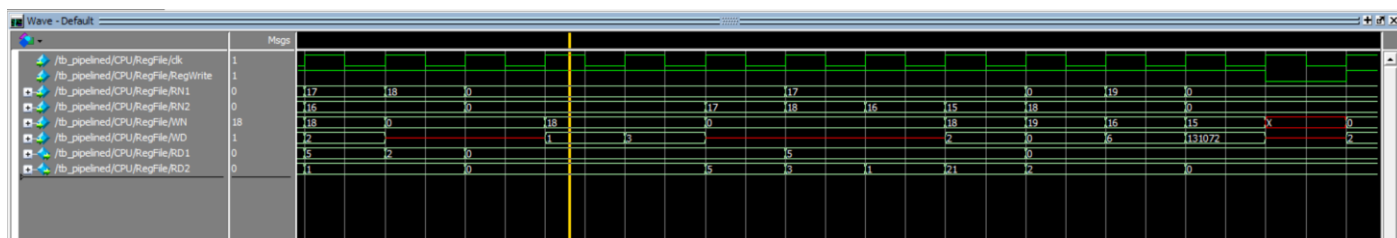
and \$s1, \$s0, \$s2



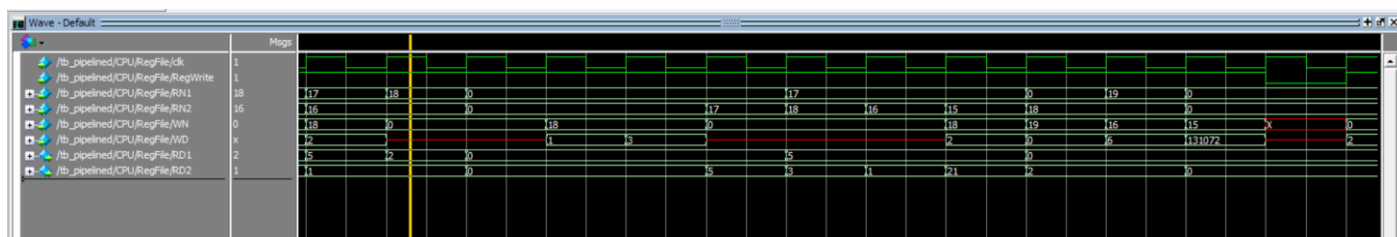
在 register 裡面 s1(17)是 5，s0(16)是 1



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



or \$s2, \$s0, \$s2

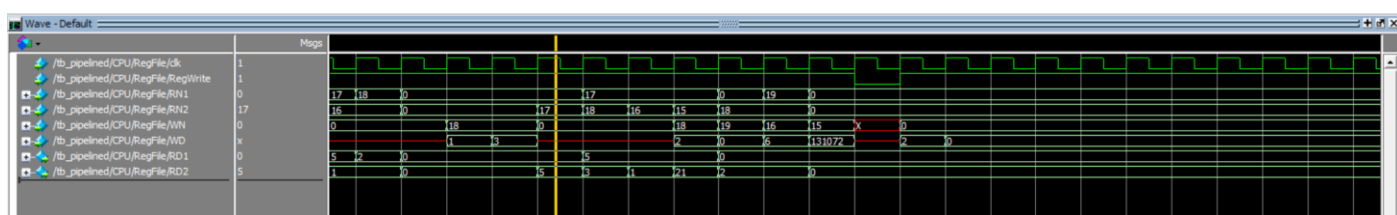


在 register 裡面 s2(18)是 2，s0(16)是 1



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

sr1 \$s1, \$s2, 1

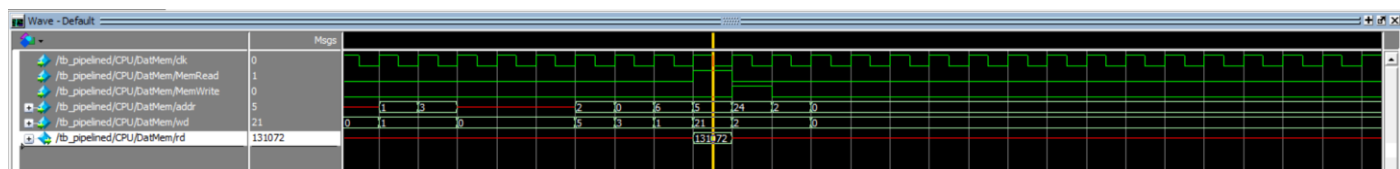


在 register 裡面 s1(17)是 5

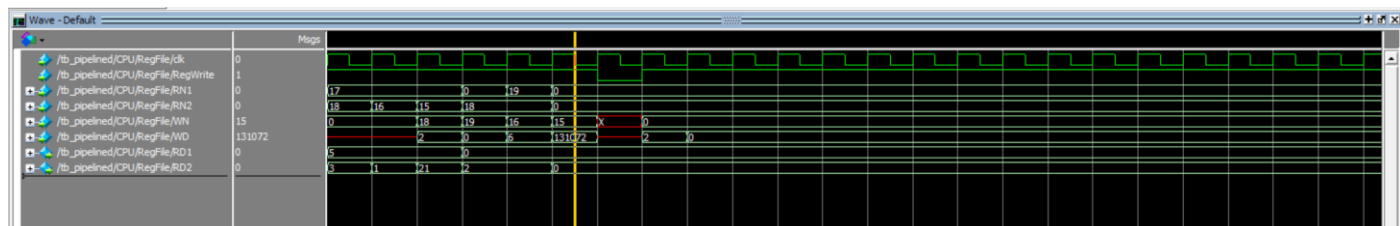

```
lw    $s1, $t7, 0
```



在 register 裡面 `s1`(17)是 5，`t7`(15)是 21

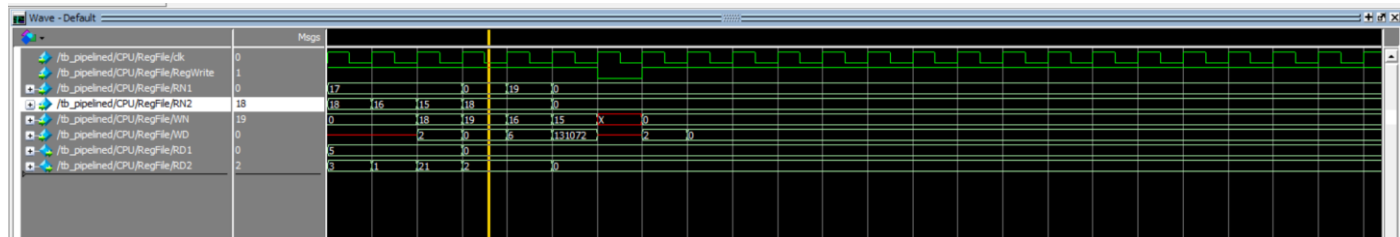


有成功讀到 memory，並回傳 rd

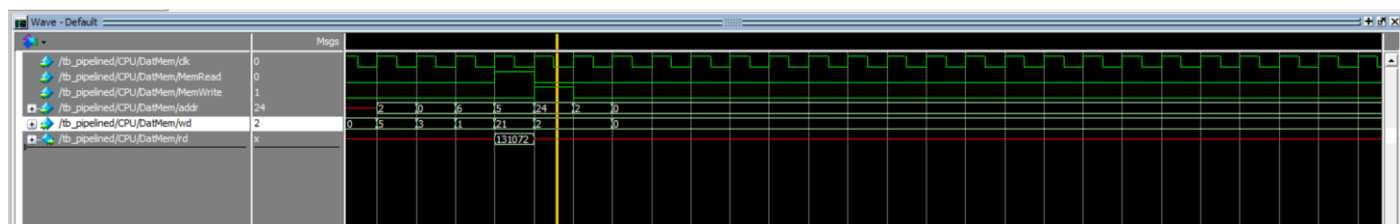


回傳到 register

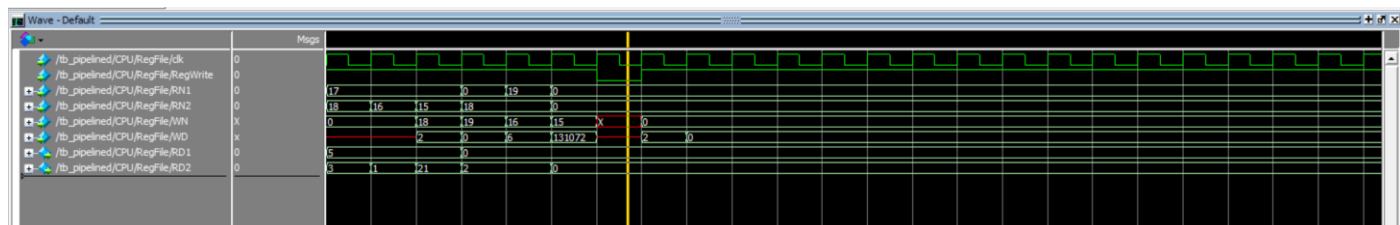
```
sw    $zero, $s2, 24
```



在 register 裡面 `zero`(0)是 0，`s2`(18)是 2

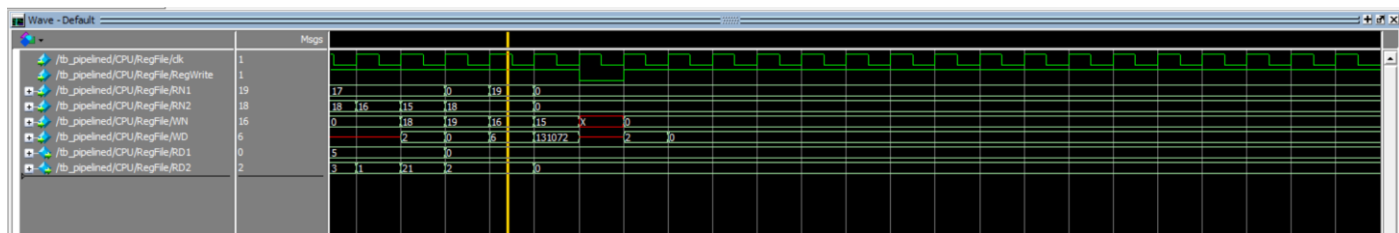


成功寫入 memory

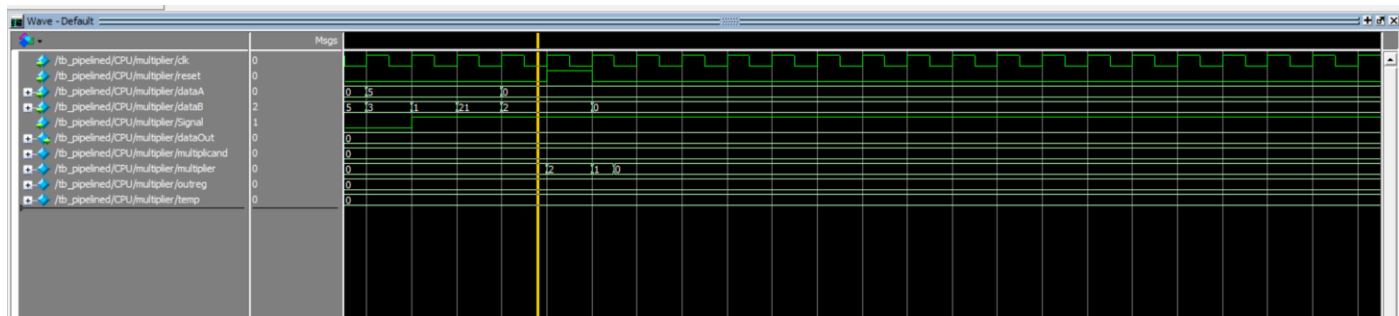


並且 register 不會寫入

multu \$s3, \$s2

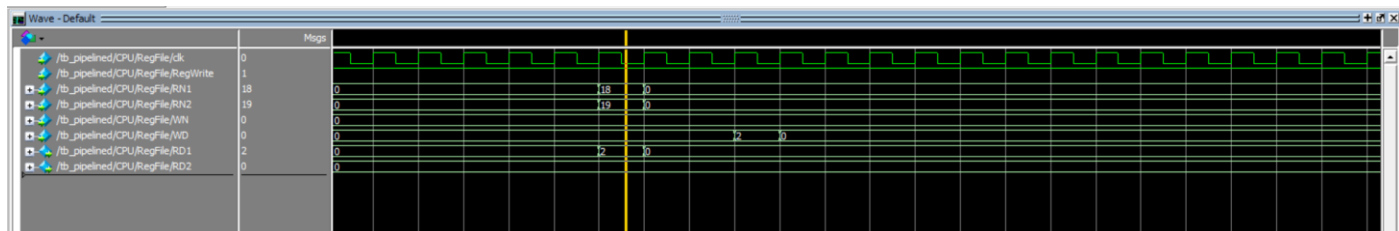


在 register 裡面 s3(19)是 0，s2(18)是 2

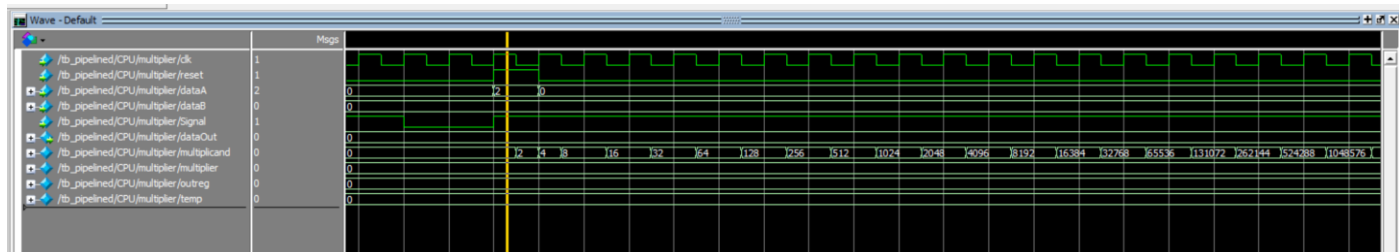


$0 * 2 = 0$ 正確

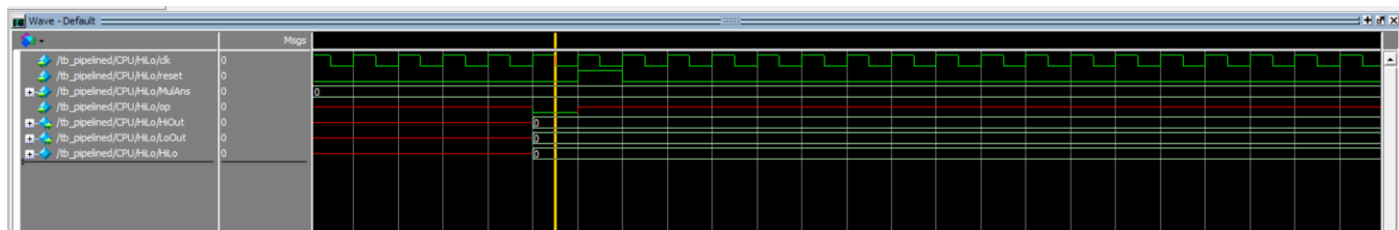
maddu \$s2, \$s3



在 register 裡面 s2(18)是 2，s3(19)是 0

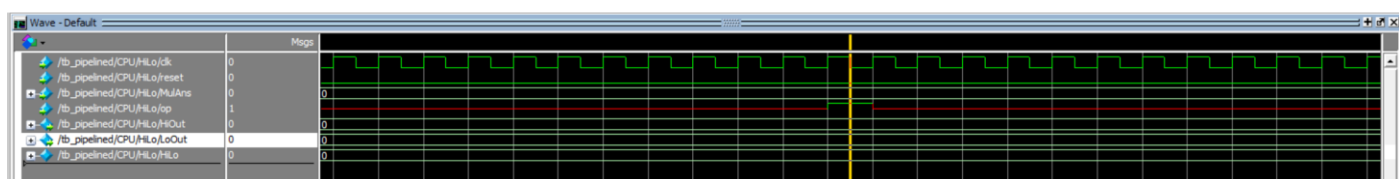


$2 * 1 = 2$



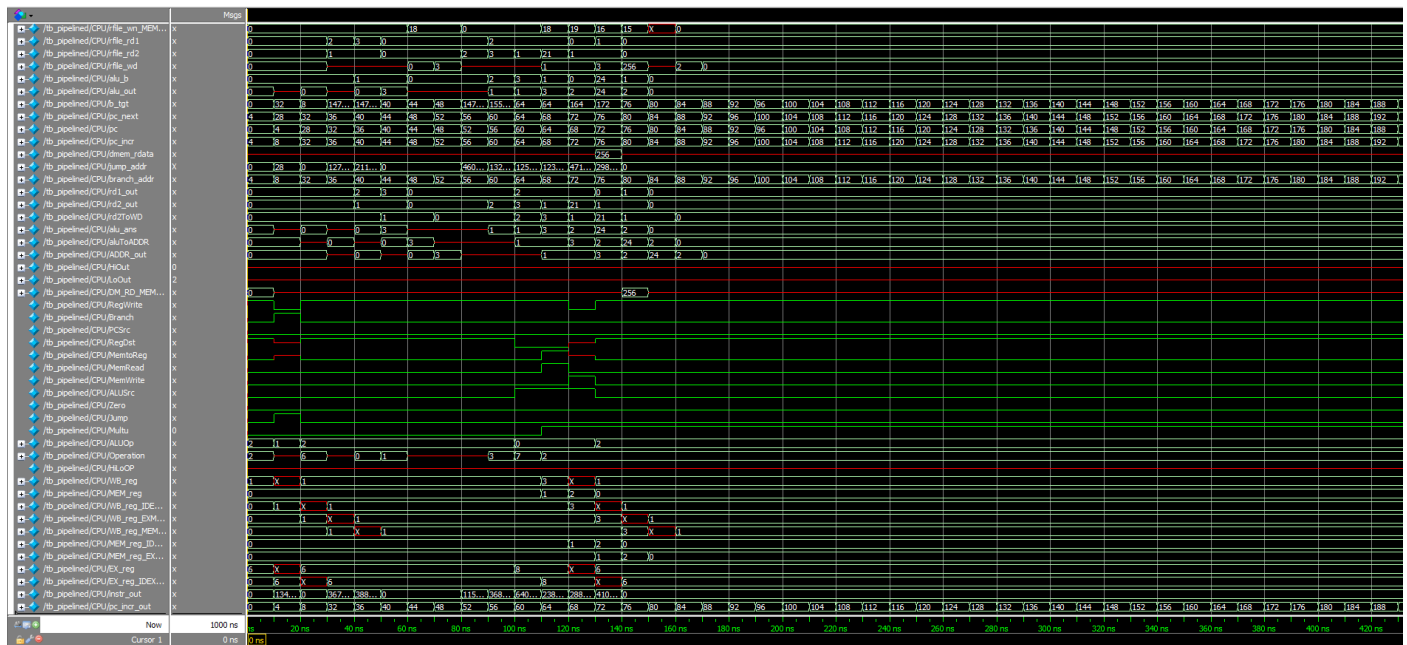
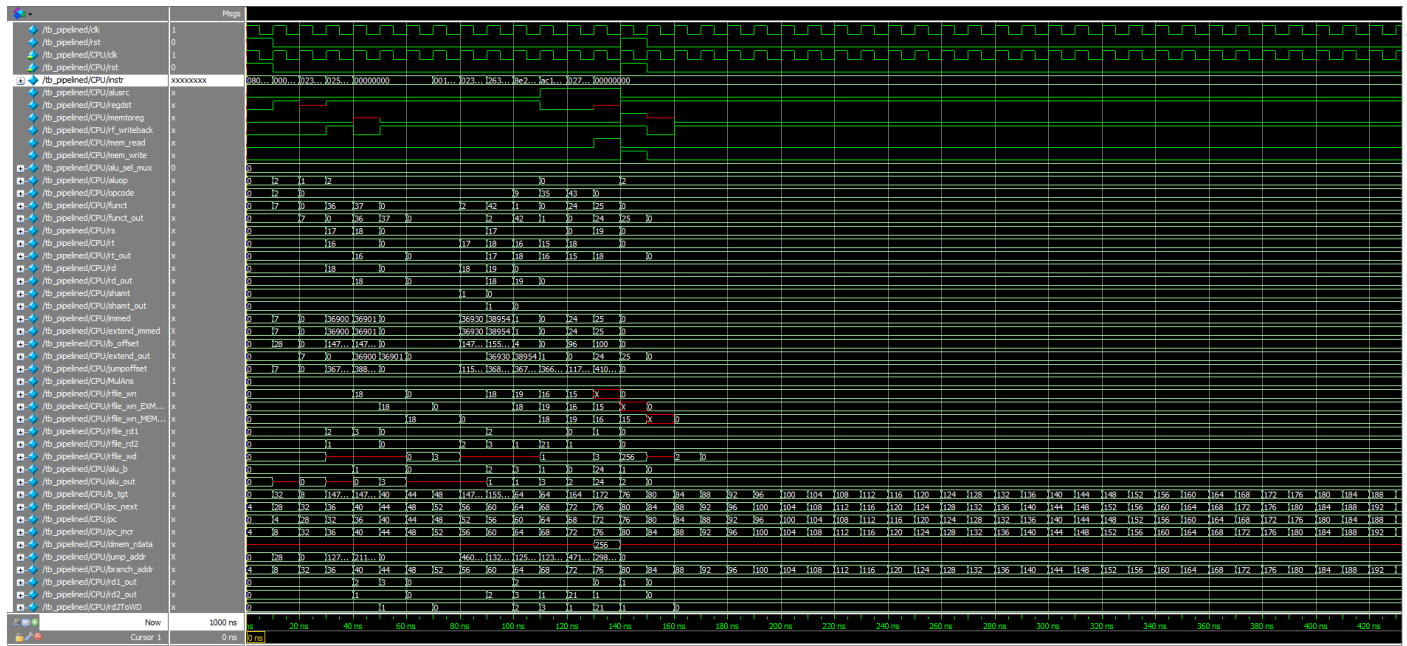
HiLo 的輸出值都為 0，所以 $2 + 0 = 2$ 正確

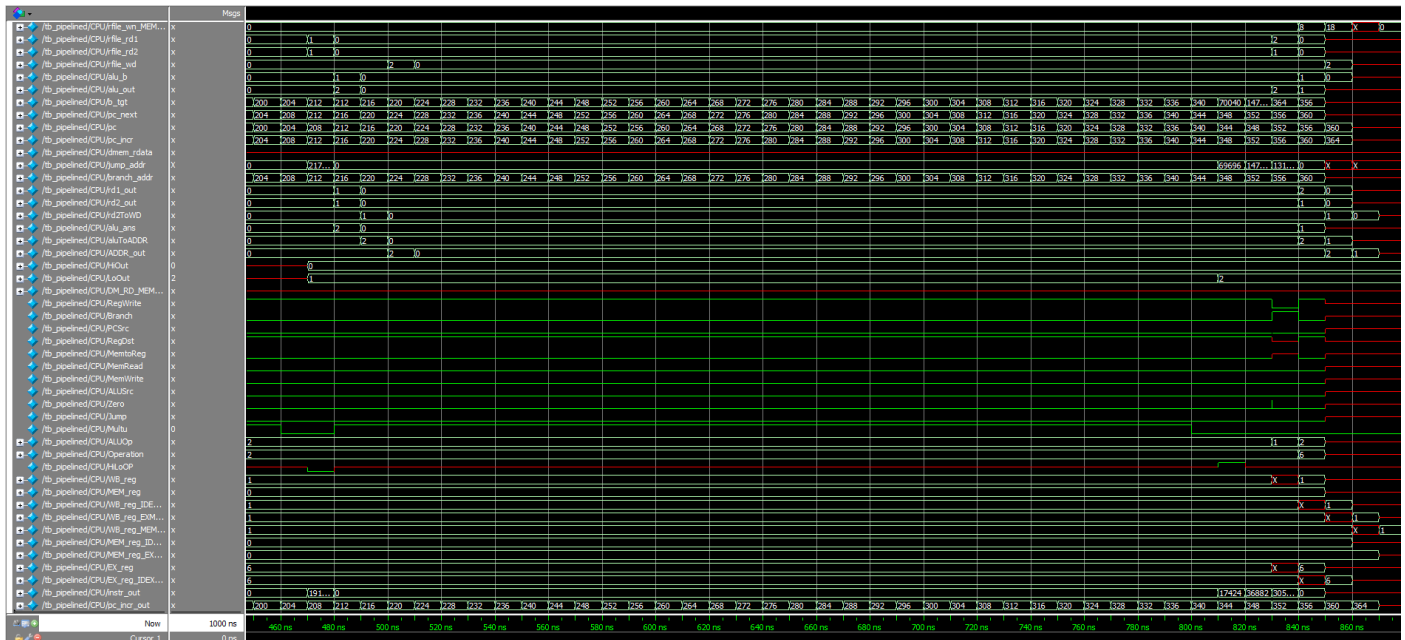
mfhi \$s1



目前的 Hi 值為 0

全部的 pipeline 輸出結果:





心得感想：

11027212 黃建閔：

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 林芷榆：

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：黃建閔、林芷榆

畫 Datapath：黃建閔、林芷榆