計算機組織 Final Project

第17組

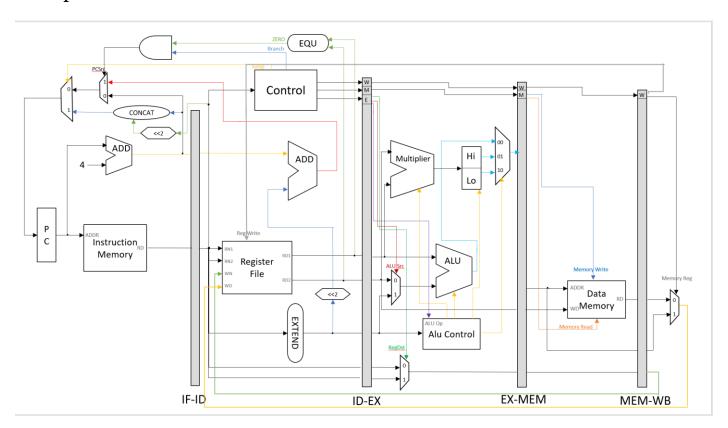
11027211 林芷榆

11027212 黄建閎

11027222 黄彦霖

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。

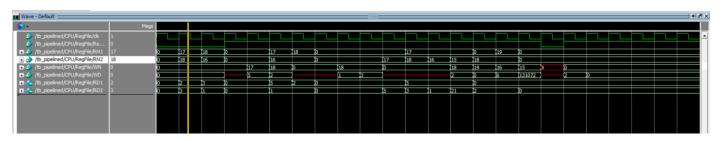
第三個區域放了 ALU, Multiplier, HiLo, ALU Control, EX/MEM register, 而在第三個區域內我們有在 ALU 前面放置一個 2 to 1 的 MUX 去判斷要放 RD2 還是 extend 完的值,還有一個 2 to 1 的 MUX 去選擇 rd 跟 rt,因為要做 MULTU 和 MADDU 所以多給了 HiLo 一個控制去判斷做哪一個動作。

第四個區域裡面放了 Data Memory, MEM/WB register

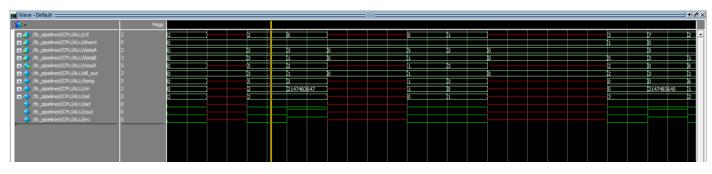
第五個區域裡面放了一個 2 to 1 的 MUX,用來判斷要回傳 ADDR 還是 RD_MEM/WB 的值給 regiser file, 並把要 write back 的東西寫回去。

驗證結果與 Waveform 圖:

add \$s1, s2, s1



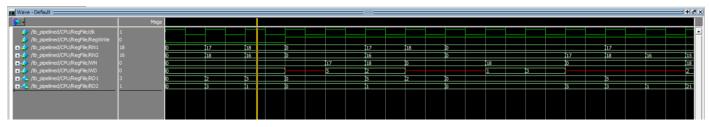
在 regiser 裡面 s1(17)是 2, s2(18)是 3



所以2+3=5正確,並且寫入s1(17)

•	Msgs															
/tb_pipelined/CPU/RegFile/dk	1						i									
/tb_pipelined/CPU/RegFile/RegWrite					_											
/tb_pipelined/CPU/RegFile/RN1		0	17	18	10			17	18	0				17		
/tb_pipelined/CPU/RegFile/RN2	0	0	18	16	10			16		0			17	[18	16	
/tb_pipelined/CPU/RegFile/WN	17	0					17	18	0		18		0			
/tb_pipelined/CPU/RegFile/WD		0			\equiv		5	2			1	3	_			
/ /tb_pipelined/CPU/RegFile/RD1		0	2	3	10			5	2	0				5		
/b_pipelined/CPU/RegFile/RD2	0	0	3	1	10			1		0			5	3	1	

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

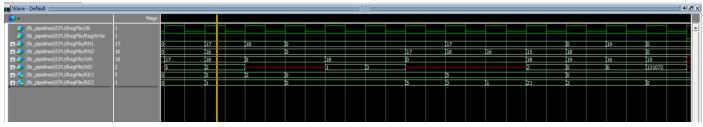


在 regiser 裡面 s2(18)是 3, s0(16)是 1

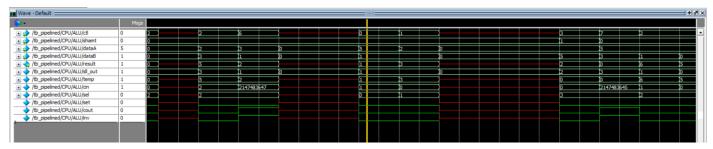
•	Msgs										
<pre>/tb_pipelined/CPU/ALU/ctl</pre>	6	2	3	2	6		(O	1		3	7
/b_pipelined/CPU/ALU/shamt	0	0								1	10
/tb_pipelined/CPU/ALU/dataA	3	0		2	3	0	5	2	Io .		5
/tb_pipelined/CPU/ALU/dataB	1	0		3	1 2	0	1		10	15	3
/tb_pipelined/CPU/ALU/result	2	0	3	5	2		1	3		2	0
/ /tb_pipelined/CPU/ALU/sll_out	1	0		3	12	10	1		10	2	3
/ /tb_pipelined/CPU/ALU/temp	2	0	3	5			1	3		0	0
/tb_pipelined/CPU/ALU/cin	2147483647	0	3	2	2147483647		1	10		0	2147483645
/ /tb_pipelined/CPU/ALU/sel	2	2	3	2			0	1		3	
/ /tb_pipelined/CPU/ALU/set	0										
/tb_pipelined/CPU/ALU/cout	1										
/tb_pipelined/CPU/ALU/inv	1			_							

所以3 - 1 = 2正確,並寫入 s2(18)

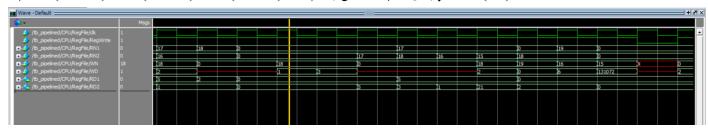
and \$s1, \$s0, \$s2



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

Wave - Default							977777									<u>= + # > ></u>
\$1 •	Msgs															
/tb_pipelined/CPU/RegFile/dk 1																_
/b_pipelined/CPU/RegFile/RegWrite 1																
/tb_pipelined/CPU/RegFile/RN1		17	18	10				17			0	19	 0			_
/tb_pipelined/CPU/RegFile/RN2		16		10			17	18	 16	15	18		 0			
/b_pipelined/CPU/RegFile/WN 0		18	10		18		0			18	19	16	15	X)
/tb_pipelined/CPU/RegFile/WD x		2	3		1	3				2	0	6	131072		- 6	2
/tb_pipelined/CPU/RegFile/RD1 2		5	2	10				5			0					_
/tb_pipelined/CPU/RegFile/RD2 1		11		10			5	3	 1	21	2		 0			

在 register 裡面 s2(18)是 2, s0(16)是 1

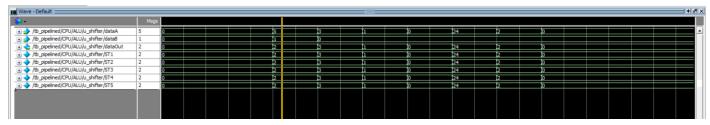
•	Msgs																	
/tb_pipelined/CPU/ALU/ctl	1		<u>0</u>	1				3	7		2							
/tb_pipelined/CPU/ALU/shamt	0	0						1	0									
/tb_pipelined/CPU/ALU/dataA	2	0	5	12	10				5)					
/tb_pipelined/CPU/ALU/dataB	1	0	1		0			5	3		1	0	24	2	_	0		
/tb_pipelined/CPU/ALU/result	3		1	3	_			- 2	0		6	5	24	2	_	0		
/tb_pipelined/CPU/ALU/sll_out	1	0	1		0			2	3		1	0	24	2	_	0		
/tb_pipelined/CPU/ALU/temp	3		1	3				— 0	0		6	5	24	2	_	0		
/tb_pipelined/CPU/ALU/cin	0		1	Ю				— 0	2147483	645	1	0						
/tb_pipelined/CPU/ALU/sel	1		0	1				3			2							
/tb_pipelined/CPU/ALU/set	0	_																
/tb_pipelined/CPU/ALU/cout	0																	
/tb_pipelined/CPU/ALU/inv	0																	

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

srl \$s1, \$s2, 1

Mogs Mogs	Wave - Default													777777										<u> + d</u>
# /tb_propined(CPU_RegFie(RegI)***	≨ 1 •		Msgs																					
□ 4/ (b. preined (CV) MeyFie (RN1 0 17 18 0 17 18 0 17 18 0 18 19 18 18 19 18 19 18 19 18 19 18 19 18 19 18 19 18 19 18 19 18 19 18 19 18 18 19 18 19 18 18 19 18 19 18 18 18 18 18 18 18 18 18 18 18 18 18	/tb_pipelined/CPU/RegFile/dk	1							Πì															
D.★ /fb_pipelined/GPU/Reg/fie/RN2 17 16 0 177 18 16 15 18 0 D.★ /fb_pipelined/GPU/Reg/fie/RD 0 18 0 18 19 16 15 X 0 D.★ /fb_pipelined/GPU/Reg/fie/RD x 1 3 2 0 6 13107Z 2 0 D.★ /fb_pipelined/GPU/Reg/fie/RD 0 5 2 0 5 0 13107Z 2 0									Ħ															
Ds. // (b) prelined/CPU/Regifie(NN) 0 18 0 18 19 15 X 0 Ds. // (b) prelined/CPU/Regifie(ND) x 1 3 2 0 131072 2 0 Ds. // (b) prelined/CPU/Regifie(ND) 0 5 2 0 131072 2 0	/b_pipelined/CPU/RegFile/RN1		17	7 18	0					_	17			0	19	Ю								
C 4 /tb_pipelined/CPURegFie/ND x 1 3 2 0 5 131072 2 0 0 141072 2 0 0 141072 2 0 0 0 141072 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	/tb_pipelined/CPU/RegFile/RN2		16	j	0				17		18	16	15	18		10							بسحا	
c. ✓, /b_pipelined/CPU/RegFile/RD1 0 5 2 0 5	/b_pipelined/CPU/RegFile/WN		0			18	8		0				18	19	16	15	х	0						
	/tb_pipelined/CPU/RegFile/WD					1		3	-				2	10	6	131072	_	2	0					
5 1 10 pelned/CPUReyFieRD2 5 1 2 2 0			5	2	0						5			0										
	/tb_pipelined/CPU/RegFile/RD2		1		0				5		3	1	21	2		10							بسحا	

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

Wave	- Default								 ****								+ 3
\$1 ▼		Msgs															
	/tb_pipelined/CPU/RegFile/dk	1															
	/tb_pipelined/CPU/RegFile/R	1															
	/tb_pipelined/CPU/RegFile/RN1		0			17			0	19	0						
	/tb_pipelined/CPU/RegFile/RN2		0		17	18	16	15	18		0						
	/tb_pipelined/CPU/RegFile/WN		18		0			18	19	16	15	X	0				
	/tb_pipelined/CPU/RegFile/WD		1	3				2	0	6	131072	_	2	ю			
•	/tb_pipelined/CPU/RegFile/RD1	5	0			5			0								
	/tb_pipelined/CPU/RegFile/RD2	3	0		5	3	1	21	2		0						
•																	

在 register 裡面 s1(17)是 5, s2(18)是 3

4	Msgs															
/tb_pipelined/CPU/ALU/ctl /tb_pipelined/CPU/ALU/shamt	7		3	7	2											
/tb_pipelined/CPU/ALU/dataA	5	0		5			0									
/tb_pipelined/CPU/ALU/dataB /tb_pipelined/CPU/ALU/result	3 0	0	2	3 0	6	5	24 24	2	0 0							
/ /tb_pipelined/CPU/ALU/sll_out / /tb_pipelined/CPU/ALU/temp	3	0	2	3	1	0	24	2	0							
/ /tb_pipelined/CPU/ALU/cin	2147483645)	2147 4.	1	0	21	2	, ,							
/tb_pipelined/CPU/ALU/sel /tb_pipelined/CPU/ALU/set	3 0		3		12											
/tb_pipelined/CPU/ALU/cout /tb_pipelined/CPU/ALU/inv	1				_											
In Themenics of working	— i———															

因為5>3,所以輸出結果是0正確,並寫入s3(19)

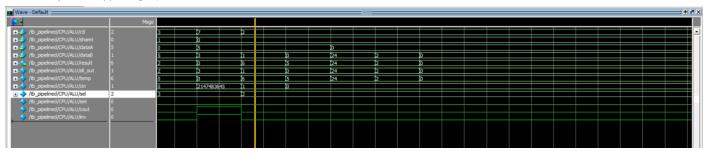
addiu \$s1, \$s0, 1

Wave - Default							*****								# 6
<u>*</u> 1 •	Msgs														
/tb_pipelined/CPU/RegFile/dk	1														
/tb_pipelined/CPU/RegFile/Re	1														
	17	0 17			0	19		0							
- /tb_pipelined/CPU/RegFile/RN2	16	17 18	16	15	18			0							
	0	0		18	19	16		15	X	0					
/tb_pipelined/CPU/RegFile/WD	x			2	0	6		131072		2	0				
-4 /tb_pipelined/CPU/RegFile/RD1	5	0 5			0										
-4 /tb_pipelined/CPU/RegFile/RD2	1	5 3	1	21	2			0							

在 register 裡面 s1(17)是 5, s0(16)是 1

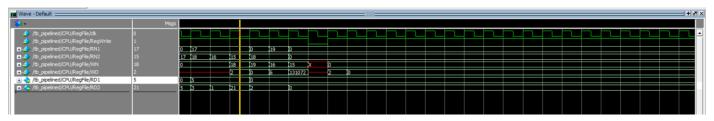
							///i													<u>+</u>	d X
Msgs																					
	0	36930	38954	1			0		24		25		0								_
1	0	4294938690	42949407	14 1			0		24		25		0								
	Msgs 1 1	1 0	1 0 36930	1 0 36930 38954	1 0 36930 38954 1	1 0 36930 38954 1	1 0 36930 38954 1	1 0 36930 38954 1 0	1 0 36930 38954 1 0	1 0 36930 38954 1 0 24	1 0 26930 38954 1 0 24	1 0 36930 38954 1 0 24 25	1 0 36930 38954 1 0 24 25	1 0 36930 38954 1 0 24 25 0	1 0 36930 38954 1 0 24 25 0	1 0 36930 38954 1 0 24 25 0	1 0 36930 38954 1 0 24 25 0	1 0 36930 38954 11 5 24 25 5	1 0 26930 28954 11 0 24 25 0	1 0 26930 28954 11 0 24 25 0	Mega 1 0 36930 38954 1 50 24 25 50

1 先做有號數擴充還是1

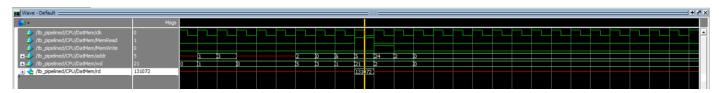


所以5+1=6正確,並寫入s0(16)

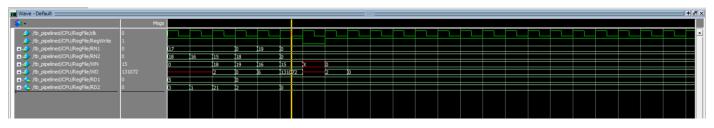
1w \$s1, \$t7, 0



在 register 裡面 s1(17)是 5, t7(15)是 21

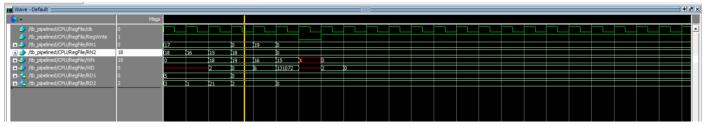


有成功讀到 memory, 並回傳 rd

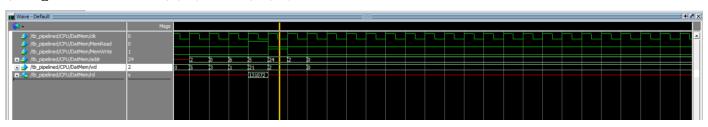


回傳到 register

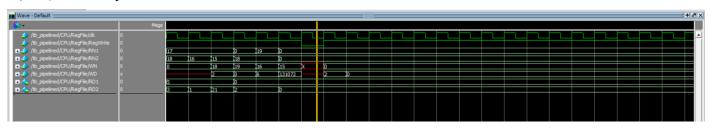
sw \$zero, \$s2, 24



在 register 裡面 zero(0)是 0,s2(18)是 2

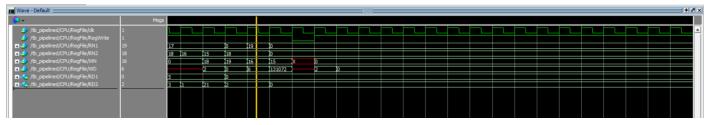


成功寫入 memory

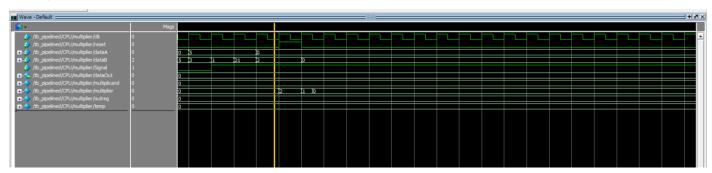


並且 register 不會寫入

multu \$s3, \$s2

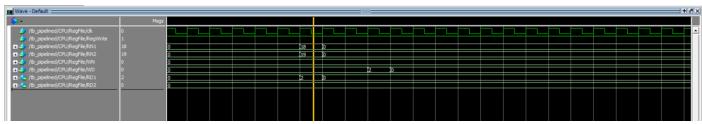


在 register 裡面 s3(19)是 0, s2(18)是 2

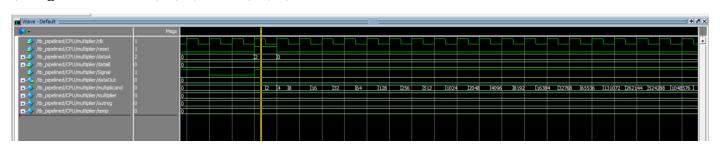


0 * 2 = 0 正確

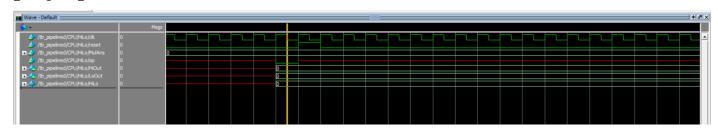
maddu \$s2, \$s3



在 register 裡面 s2(18)是 2, s3(19)是 0



2 * 1 = 2

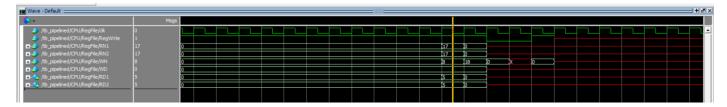


HiLo 的輸出值都為 0,所以 2+0=2 正確

mfhi \$s1

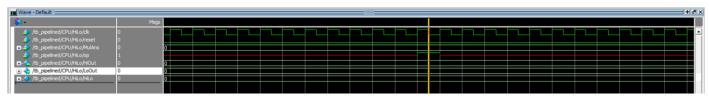
Wave - Default							 W							==
<u>\$</u> 1 •		Msgs												
/tb_pipelined/CPU/HiLo/dk	0													
/tb_pipelined/CPU/HiLo/reset														
/tb_pipelined/CPU/HiLo/MulAns		0												
/tb_pipelined/CPU/HiLo/op								-						
-4 /tb_pipelined/CPU/HiLo/HiOut		0												
💠 /tb_pipelined/CPU/HiLo/LoOut	0	0												
	0	0												

目前的 Hi 值為 0

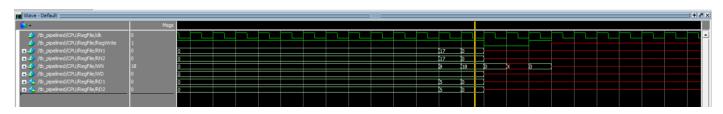


回傳到 register 並把 Hi 值寫入 sl

mflo \$s2

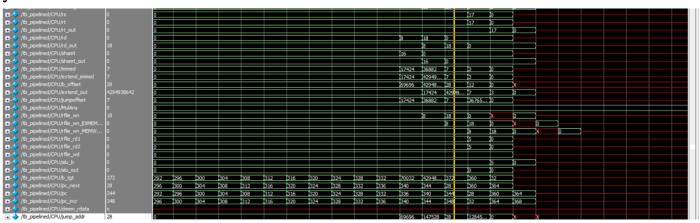


目前的 Lo 值為 0



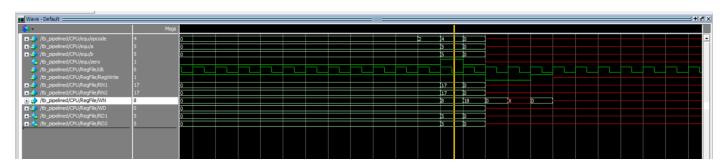
回傳到 register 並把 Lo 值寫入 s2

i 7



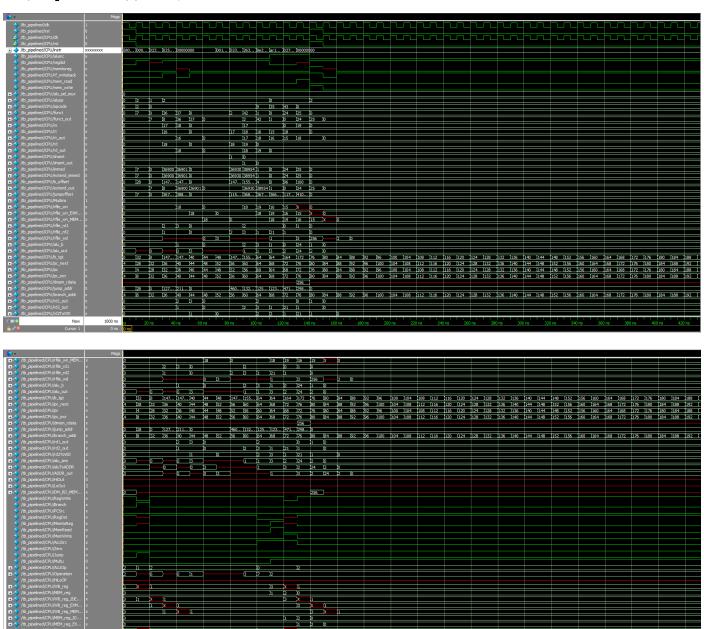
Immed 是 7, 最後 jump 輸出為 28

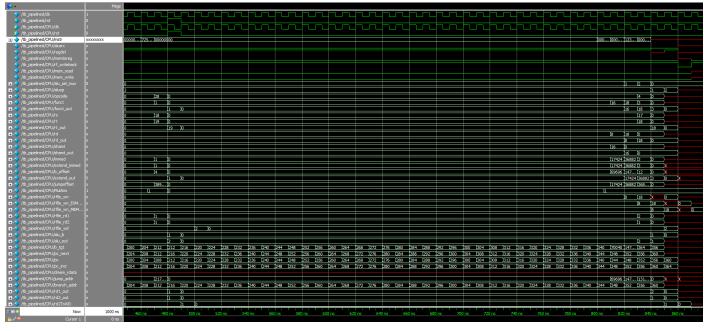
beq \$s1, \$s2, 3

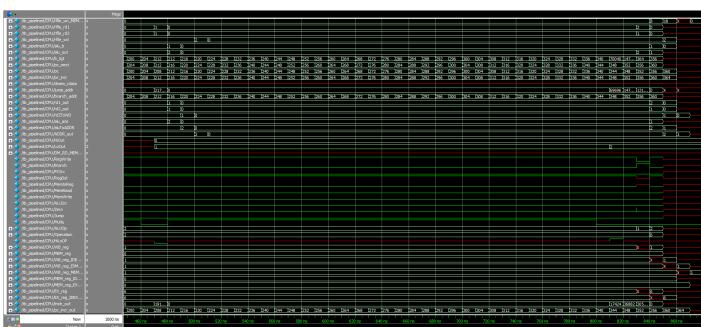


在 register 裡面 s1 是 5, s2 是 5 所以在做 equ 的時候回傳 zero 為 1

全部的 pileline 輸出結果:







```
add wave -position insertpoint sim:/tb_pipelined/CPU/*
                       0, reading data: Mem[
                                                     x1 =>
  18446744073709551615, PC: x
0, reading data: Mem[
                                                  0] => 134217735
0 (Port 2)
0 (Port 1)
                      0, reg_file[ 0] =>
0, reg_file[ 0] =>
0, reg_file[ 0] =>
0, PC: 0
0, wd: 0
                      O, NOP
                                                                                                   44, NOP
                      1, reading data: Mem[
                                                     41 =>
                                                                        0
                                                                                                   45, PC:
                                                                                                                         200
                                                                                                                          0
                                                                                                   45, wd:
                      2, reading data: Mem[
                                                      28] => 36737060
                                                                                                    45, NOP
                      2, PC: 28
2, wd: 0
                      2, NOP
                                                                                                    46, reading data: Mem[
                                                                                                                                              204] => 1918042113
                                                                                                   46, PC: 204
                                                      321 =>
                       3, reading data: Mem[
                                                               38834213
                      3, reg_file[16] =>
3, reg_file[17] =>
3, PC: 32
3, wd: x
                                                     1 (Port 2)
2 (Port 1)
                                                                                                    46, wd:
                                                                                                    46, NOP
                      3, AND
                                                                                                                                                208] =>
                                                                                                    47, reading data: Mem[
                      4, reading data: Mem[
4, reg_file[18] =>
4, PC: 36
4, wd: x
4, OR
                                                      361 =>
                                                                                                    47, reg_file[19] =>
                                                                                                                                                1 (Port 2)
                                                     3 (Port 1)
                                                                                                    47, reg_file[18] =>
                                                                                                                                                 1 (Port 1)
                                                                                                   47, PC:
47, MADDU
                                                                                                                        208
                       5, reg_file[ 0] =>
                                                     0 (Port 2)
                      5, reg_file[ 0] => 5, PC: 40 5, wd: x
                                                     0 (Port 1)
                                                                                                    48, reg_file[ 0] =>
                                                                                                                                                0 (Port 2)
                                                                                                    48, reg_file[ 0] =>
                                                                                                                                                 0 (Port 1)
                       5, NOP
                                                                                                    48, PC:
                                                                                                                         212
                                                                                                    48, wd:
                      7, reg_file[18] <= 6, PC: 44 6, wd: 0 6, NOP
                                                     0 (Write)
                                                                                                    48, NOP
                                                                                                    49, PC:
                                                                                                                         216
                                                                                                   49, wd:
                      8, reg_file[18] <=
7, PC: 48
7, wd: 3
                                                     3 (Write)
                                                                                                    49, NOP
                    8, reading data: Mem[
                                                     52] => 36870186
                    8, reg_file[17] =>
8, PC: 52
                                                    2 (Port 2)
                                                                                                            78, PC:
                    8, wd:
                                                                                                            78, NOP
                                                                                                            79, PC:
                                                                                                                            336
                                                     56] => 640679937
                    9, reading data: Mem[
                                                                                                            79, wd:
79, NOP
                   9, reg_file[18] =>
9, reg_file[17] =>
9, PC: 56
                                                    3 (Port 2)
                                                    2 (Port 1)
                                                                                                            80, reading data: Mem[
                                                                                                                                              340] =>
                                                                                                                                                              17424
                                                                                                           80, PC:
80, wd:
                    9, wd:
                                                                                                                        340
                    9, SLT
                                                                                                            80, NOP
                  10, reading data: Mem[
10, reg_file[16] =>
10, PC: 60
10, ADDIU
                                                     601 => 2385444864
                                                    1 (Port 2)
                                                                                                            81, reading data: Mem[
                                                                                                                                              344] =>
                                                                                                           81, PC: 344
81, wd: 0
                                                                                                            81, MFHI
                                                   64] => 2886860824
21 (Port 2)
                  11, reading data: Mem[
                  11, reg_file[15] =>
12, reg_file[18] <=
                                                                                                            82, reading data: Mem[
                                                                                                                                              3481 => 305266691
                                                                                                            82, PC: 348
82, wd: 0
                  11, PC:
11, LW
                                                                                                            82. MFLO
                  12, reading data: Mem[
                                                      681 => 41025561
                                                                                                            83, reading data: Mem[
                                                                                                                                              352] =>
                                                                                                                                                                   0
                  12, reg_file[ 0] =>
12, reg_file[18] =>
13, reg_file[19] <=
                                                    0 (Port 1)
                                                                                                            83, reg_file[18] =>
83, reg_file[17] =>
                                                                                                                                              1 (Port 2)
                                                    1 (Port 2)
                                                                                                                                              2 (Port 1)
                                                     1 (Write)
                                                                                                            83, PC:
                  12, PC:
                                    68
                                                                                                            83. BEO
                  12. SW
                                                                                                            84, reading data: Mem[
                                                                                                                                               356] =>
                  13, reading data: Mem[
13, reading data: Mem[
                                                     72] =>
2] =>
                                                                                                            84, reg_file[ 0] =>
84, reg_file[ 0] =>
                                                                                                                                              0 (Port 2)
                                                                      256
                                                                                                                                              0 (Port 1)
                  13, reg_file[19] =>
                                                    1 (Port 1)
                  14, reg_file[16] <=
13, PC: 72
                                                                                                            85, reg_file[ 8] <=
                                                    3 (Write)
                                                                                                                        356
                                72
                                                                                                            84, PC:
                                                                                                            84, wd:
                  13, wd:
                  13, MULTU
                  14, reg_file[ 0] => 14, reg_file[ 0] =>
                                                                                     control single unimplemented opcode x
                                                    0 (Port 1)
                                                     0 (Port 2)
                                                                                                            85, reg_file[ x] =>
                                                                                                                                              x (Port 2)
                  15, reg_file[15] <=
                                                  256 (Write)
                                                                                                            85, reg_file[ x] =>
                  15, writing data: Mem[
14, PC: 76
                                                     24] <=
                                                                        1
                                                                                                            86, reg_file[18] <=
85, PC: 360
                                                                                                                                              2 (Write)
                  14, wd:
14, NOP
                                    256
                                                                                                            86, PC:
                                                                                                            87, PC:
                                                                                                            88, PC:
                  15, PC:
15, wd:
15, NOP
                                  80
```

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 model sim 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 黄彥霖:

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