109 2 計算機組織期末 Project

1.組員資料

組別：23

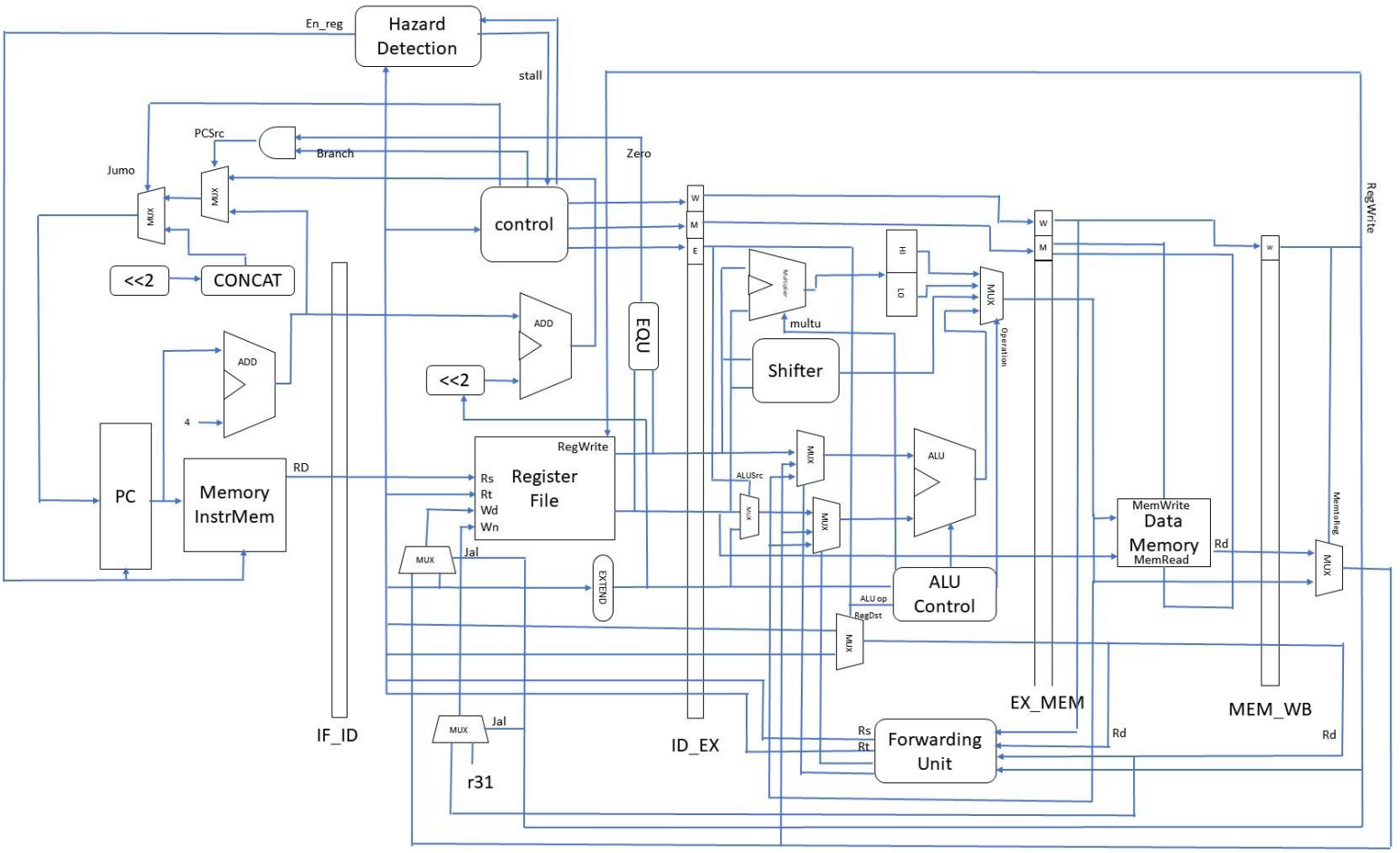
組員：資訊二乙　10827217　蔡佳軒

資訊二乙　10827233　林庭

資訊二乙　10827234　彭桂綺

資訊二乙　10827235　馮信華

2.Datapath 與詳細架構圖



3.設計重點說明

使用single_cycle_CPU將其用5個暫存器(分別是IF_ID、ID_EX、EX_MEM、MEM_WB)切成五份，其中:

第一階(IF)包括

PC、PCADD、PCMUX、JMUX、InstrMem

IF_ID暫存器存了clk, rst, en_reg, instruction, pc的值

第二階(ID)包括

hazardDection、STALLMUX、SignExt、JalMUX、JalWnMUX、RegFile 、CTL、BRADD、branchEqual、BR_AND

ID_EX暫存器存了

clk,rst,WB(from CTL),MEM(from CTL),EX(from CTL),RD1(from RegFile),RD2(from RegFile),immed(from SignExt),rt(from instruction), rd(from instruction), rs(from instruction), Jal(from CTL)的值

第三階(EX)包括

ALU、branchEqual、BR_AND、ALU_CTL、multiplier、hilo、shifter、RESULTMUX、RFMUX、ALUMUX、ALUMUX1、ALUMUX2、Forwarding

EX_MEM暫存器存了

clk, rst,WB(from ID_EX), MEM(from ID_EX), ALU(from ALU), WN(from RFMUX), zero(from ALU), WD(from ID_EX), Jal(from ID_EX)的值

第四階(MEM)包括 DatMem

MEM_WB暫存器存了clk,rst,WB(from EX_MEM),RD(from DatMem), ADDRESS(from EX_MEM), WN(from EX_MEM), Jal(from EX_MEM)的值

第五階(WB)包括 WRMUX

4. Icarus Verilog 驗證結果與 Waveform 輸出圖形

Icarus Verilog 驗證結果：

```
ting to 1364-2005 behavior.
               0, reading data: Mem[         x] =          x
               0, reading data: Mem[         0] => 2385444864
               0, reg_file[ 0] =>          0 (Port 1)
               0, reg_file[ 0] =>          0 (Port 2)
               0, PC:          0
               0, wd:          0
               0, NOP

               1, reading data: Mem[         4] =>  305201155
               1, reg_file[17] =>          2 (Port 1)
               1, reg_file[15] =>         21 (Port 2)
               1, PC:          4
               1, LW

               2, reg_file[17] =>          2 (Port 2)
               2, PC:          4
               2, BEQ

               3, reading data: Mem[         2] =>        256
               3, reg_file[ 0] =>          0 (Port 1)
               3, reg_file[ 0] =>          0 (Port 2)
               3, reading data: Mem[        20] =>  134217738
               3, PC:         20
               3, wd:          0
               3, NOP

               4, reading data: Mem[        24] =>   38834213
               5, reg_file[15] <=         256 (Write)
               4, PC:         24
               4, J

               5, reading data: Mem[        40] =>   38834210
               5, reg_file[18] =>          3 (Port 1)
               5, reg_file[16] =>          1 (Port 2)
               5, PC:         40
               5, wd:          x
               5, OR

               6, reading data: Mem[        44] =>   38834213
               6, PC:         44
               6, wd:          0
               6, SUB

               7, reading data: Mem[        48] => 2886860824
               7, PC:         48
               7, wd:          x
               7, OR

               8, reading data: Mem[        52] =>   45482009
               8, reg_file[ 0] =>          0 (Port 1)
               8, reg_file[18] =>          3 (Port 2)
               9, reg_file[18] <=          3 (Write)
               8, PC:         52
               8, SW
```

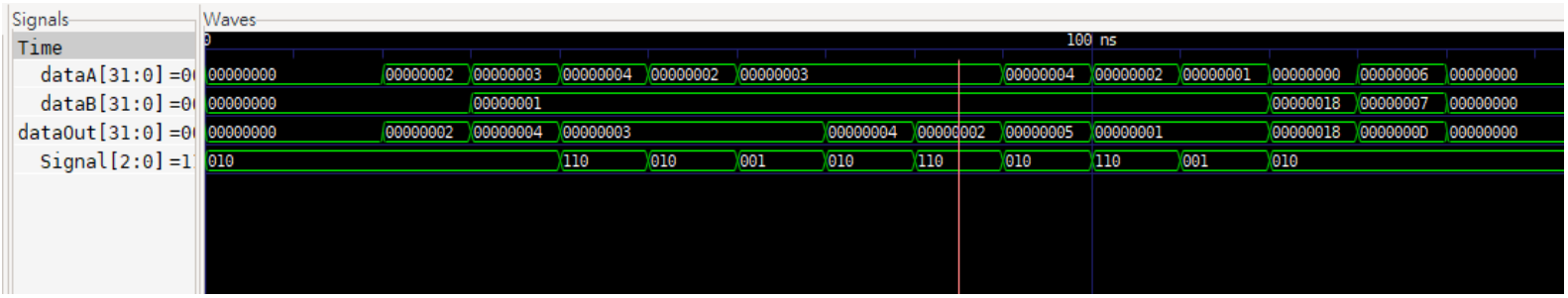
```
 9, reg_file[21] =>          6 (Port 1)
 9, reg_file[22] =>          7 (Port 2)
 9, PC:        52
 9, wd:          2
 9, MULTU

10, reg_file[18] <=          2 (Write)
10, reg_file[ 0] =>          0 (Port 1)
10, reg_file[ 0] =>          0 (Port 2)
11, reg_file[18] <=          3 (Write)
11, writing data: Mem[        24] <=          3
10, PC:        52
10, wd:          3
10, NOP

11, PC:        52
11, wd:          x
11, NOP

12, PC:        52
12, wd:         13
12, NOP

13, PC:        52
13, wd:          0
13, NOP

14, PC:        52
14, wd:          0
14, NOP
```

```
40, PC:         52
40, wd:          0
40, NOP

41, reading data: Mem[        56] =>      43024
41, PC:         56
41, wd:          0
41, NOP

42, reading data: Mem[        60] =>      45074
42, PC:         60
42, wd:          0
42, MFLO

43, reading data: Mem[        64] =>  638648832
43, PC:         64
43, wd:          0
43, MFHI

44, reading data: Mem[        68] =>  201326614
44, reg_file[16] =>          1 (Port 1)
44, reg_file[17] =>          2 (Port 2)
44, PC:         68
44, ADDIU

45, reg_file[ 0] =>          0 (Port 1)
45, reg_file[ 0] =>          0 (Port 2)
45, PC:         68
45, JAL

46, PC:         68
46, wd:          0
46, NOP

47, PC:         68
47, wd:         513
47, NOP

48, reg_file[17] <=        513 (Write)
48, reading data: Mem[        72] =>   38834210
49, reg_file[31] <=         72 (Write)
48, PC:         72
48, wd:          0
48, NOP

49, reading data: Mem[        76] =>   38834213
49, reg_file[18] =>          3 (Port 1)
49, reg_file[16] =>          1 (Port 2)
49, PC:         76
49, wd:          0
49, SUB

50, reading data: Mem[        80] => 2886860824
50, PC:         80
50, wd:          0
50, OR
```

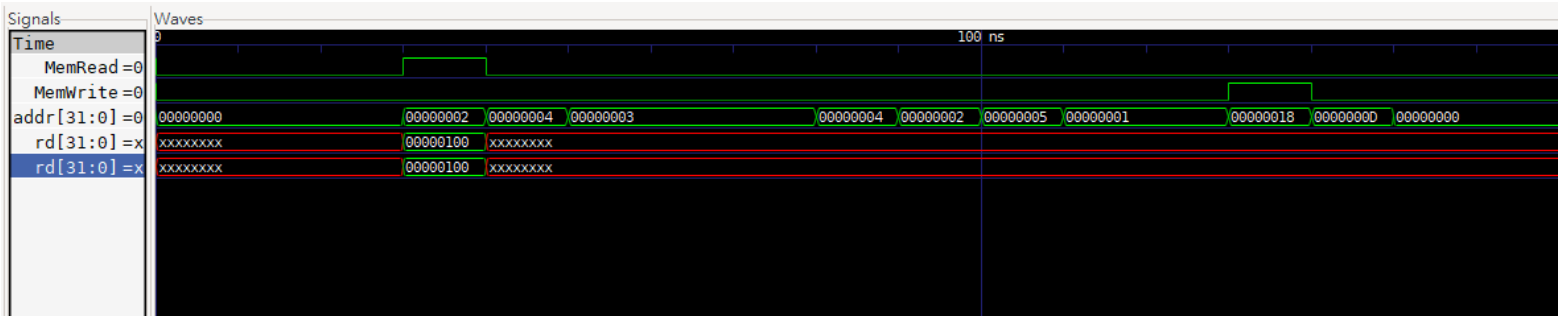
```
                50, OK

                51, reading data: Mem[        84] =>          x
                51, reg_file[ 0] =>          0 (Port 1)
                51, reg_file[18] =>          3 (Port 2)
                51, PC:        84
                51, SW

ntrol_single unimplemented opcode  x
                52, reg_file[ x] =>          x (Port 1)
                52, reg_file[ x] =>          x (Port 2)
                52, PC:        88
                53, PC:         x
                54, writing data: Mem[        24] <=          3
                54, PC:         x
                55, PC:         x
                56, PC:         x
                57, PC:         x
                58, PC:         x
                59, PC:         x
                60, PC:         x
                61, PC:         x
                62, PC:         x
                63, PC:         x
                64, PC:         x
                65, PC:         x
                66, PC:         x
                67, PC:         x
                68, PC:         x
                69, End of Simulation
```

Waveform：

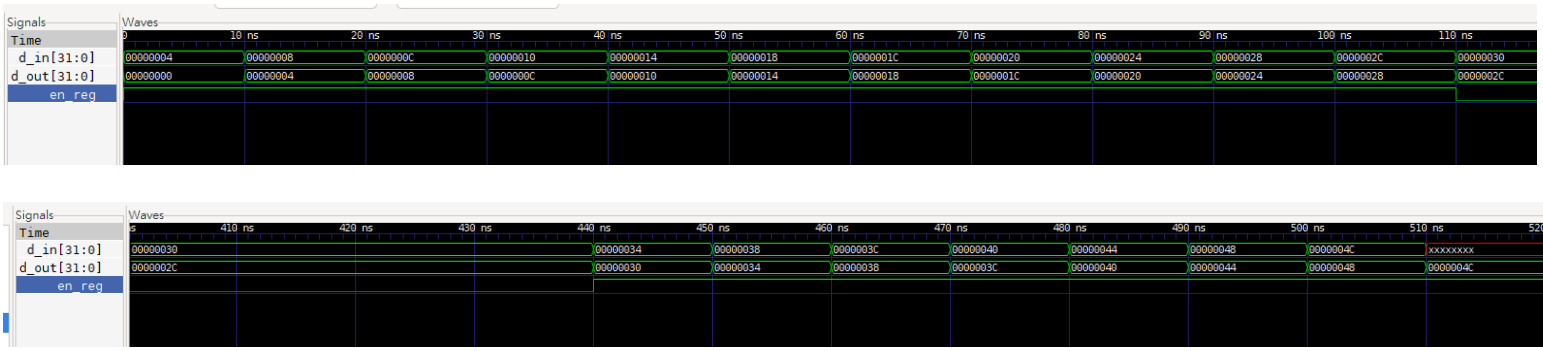
(a)ALU : (包括add, sub, and, or, sll, slt, addiu)



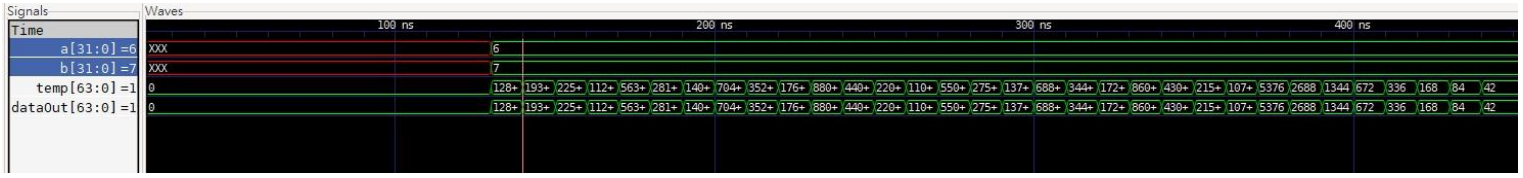
(b)lw, sw :



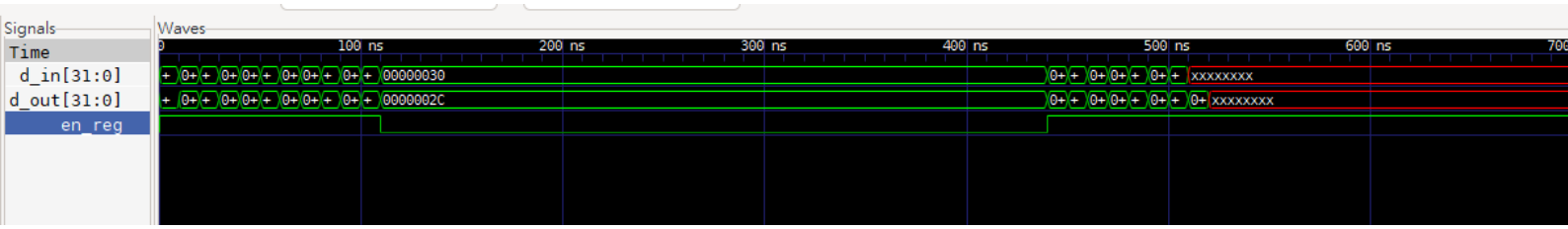
(c)beq, j, jal :



(d)multu :



(e)mfhi, mflo, nop :



5.心得感想

蔡佳軒：

這次的期末project相比其中的難度提升了好幾倍，期中的時候還可以翻翻課本就找得到答案，或是查查網路就會有相對應的方法可以解決，但這次的project卻花了我們好幾倍的時間，不管是新指令的理解，還是變成pipeline的部分，還有Hazard跟forwarding的處理，都讓我們遇到了很大的困難，我們甚至連續七天每天都花至少三個小時開會來寫程式，有一天甚至開了八個小時到凌晨五點天都亮了，雖然很辛苦但的確學習到了很多，也感受到了把這學期整本計算機組織課本的內容利用實際的行動來完成的成果。

林庭：

期末除了難度大幅提升之外，因為少了很多堂課的關係課程也尚未教完，但project難度依舊，導致Hazard部分超級難處理，需要上網找方法自學，學到的東西套用上我們的架構又無法百分之百契合，課本的圖沒有經過老師解說也根本無法看懂，它只給概念而沒有實作方法，所幸在團隊鍥而不捨的努力下forwarding最後有寫出來。期末project準備時間不到三周，所需知識又不夠的狀況下，能做到這樣我覺得已經非常值得嘉許了，連續七天每天超過四小時線上會議，常常開完會外面天色也就亮了，有種肝壽命三周內減十年的感覺，但也感謝有這樣的project，有機會讓我們實踐計算機組織概論所學，不過這次project讓我知道硬體對我來說可能不是這麼適合!未來在方向的選擇上也比較有心理準備，現在的我只希望這些所學是夠用的，也希望有機會把Hazard的課聽完，才不會帶著滿腔的困惑升上大三被大三的教授說:"你們Hazard不是應該學過也寫過了嗎?怎麼還聽不懂"ಥ‿ಥ

彭桂綺：

這堂課一貫的作風，期中project才剛寫完，沒有緩衝期，緊接著就是期末project。期末project的難度真不是期中可以比擬的，寫了期末project才發現原來期中寫的只是期末的小元件。再加上時間緊迫以及課堂沒有完整上完，導致我們欠缺許多應該要有的先備知識，這就讓我們撰寫的難度再增加。我們先是花了不少時間去瞭解 piped line 的原理，再來才能實作它。其中作讓人心累的點是，當我們都想好應該怎麼寫了，也想好應該要改變那些硬體的種類(MUX2 變 MUX3 之類的)才發現要在暫存器增加port是多麻煩的事。不但每一個階段都要新增進入及出暫存器的port ，就連控制訊號都要從1 bit 改成 2 bit，真的超麻煩的。好在最後都順利結束，但是我以後應該是不會想走硬體這條路，除非我想早死吧~~:)。雖然這次project 難度超高，但是我跟組員歷經一個禮拜挑燈夜戰，最終還是順利完成了喔耶，相信我學到很多、也能對於硬體描述語言及 Pipeed line 有更深入的了解。:)

馮信華：

這次的期末project主要是在建立一個pipelined的CPU，而和single cycle的CPU不同的就在於暫存器的使用，所以我們依照著講義，把4個暫存器都建立出來，一開始還不太知道怎麼將暫存器寫成module，把老師給的程式碼都看過後才知道就像PC暫存器一樣，然後我們就把需要的訊號都給接上去，之後的難題就在於，將ALU、Shifter、乘法器這些組件接上去，讓這顆CPU能執行這些指令，這邊花了我們許多時間，而且執行結果不是那麼容易看懂，幸好經過和組員的討論後有接起來，最後我們遇上的大麻煩就是：mul、j、jar、beq指令，這些指令計算完後傳回PC需要時間，而stall的處理我們覺得非常不容易，要怎麼讓PC delay，真的蠻困難的，我和我的組員們花了最多時間在處理這件事上，看講義也找不

到什麼方法，最後我們自己寫了一個module來delay，雖然流程長得和講義最後一頁的圖有點不太一樣，但是這顆CPU有成功delay，最後完成這個project，真的學到很多，感覺自己verilog又增強了許多，也對CPU架構更加理解了。

6.各組員分工方式與負責項目

報告書撰寫：蔡佳軒、林庭、彭桂綺、馮信華

繪製DataPath：蔡佳軒、林庭、彭桂綺、馮信華

程式撰寫：蔡佳軒、林庭、彭桂綺、馮信華