

微算機系統期末報告

組別：14

組員：梁皓鈞 (104360098)

洪晟毅 (104590048)

2017.01.15

作品介紹

本次作品是以 VHDL 實作出一個可以輸入 opcode, rx, rt/data 並且執行要求的 ALU。由於我們作品除了做基本分部份外，另外還做了加分項目：乘法，除法，64 個暫存器，因此我們的輸入是會分為三段式輸入，分別是先輸入 opcode，再輸入暫存器編號，再輸入另一個暫存器的編號（假如 opcode 是 load，則輸入一個 8bits 數值）。每一個動作需要用 clock 進行啟動。因此作品中有大量使用狀態機進行控制。

作品中利用了 VHDL 使用程式邏輯編寫硬體邏輯的特性，大大減少了硬體邏輯的麻煩，透過上網查找 IEEE 的 library 功能後，製作速度極快。其中 VHDL 中的 unsigned 以及 numeric library 幫助甚大，簡化了很多已知的東西，不需要重造輪子。

由於硬體電路接線上會有延遲，因此在某些狀態下的大量接線會出現延遲，因此我們在某些部份的功能上拆多幾個小狀態進行，多做一個 clock 便可以完成。解決了一直以來出現的數值無法同步全部輸進去 signal 的問題。作品原本一開始是 4 個 state，後來增加到 6 個 state 用以改善延遲問題。

本次作品大量地運用了先前 6 次實驗的成果。特別應用多在於 Binary to BCD，seven-segment，finite state machine。此外由於 DE2 板子會有彈跳現象令 clock 的輸入出現不穩定，以致無法順利測試，因此我們也應用了 de-bounce 的程式用以解決彈跳問題。

這次作品總共有 10 個功能，分別是輸入，行動，加，減，乘，除，And，Or，Nor，Slt。而狀態則有 A, B, Bhalf, C, Chalf, D, E。詳細 Opcode，狀態以及操作解說會在系統功能介紹中列明。

系統功能介紹

Instruction	Opcode
Load	100000
Move	100001
Add	100010
Sub	100011
Mult	100100
Div	100110
And	101011
Or	101100
Nor	101110
Slt	110000

操作：

作品的狀態機狀態：A, B, Bhalf, C, Chalf, D, E

A：要輸入 Opcode

B：輸入 Rx index code

Bhalf：存入 signal

C：如果 Opcode 是 load, 就要輸入 data, 其他則是輸入 Rt index code

Chalf：存入 signal

D：由 ALU 把資料拿回來

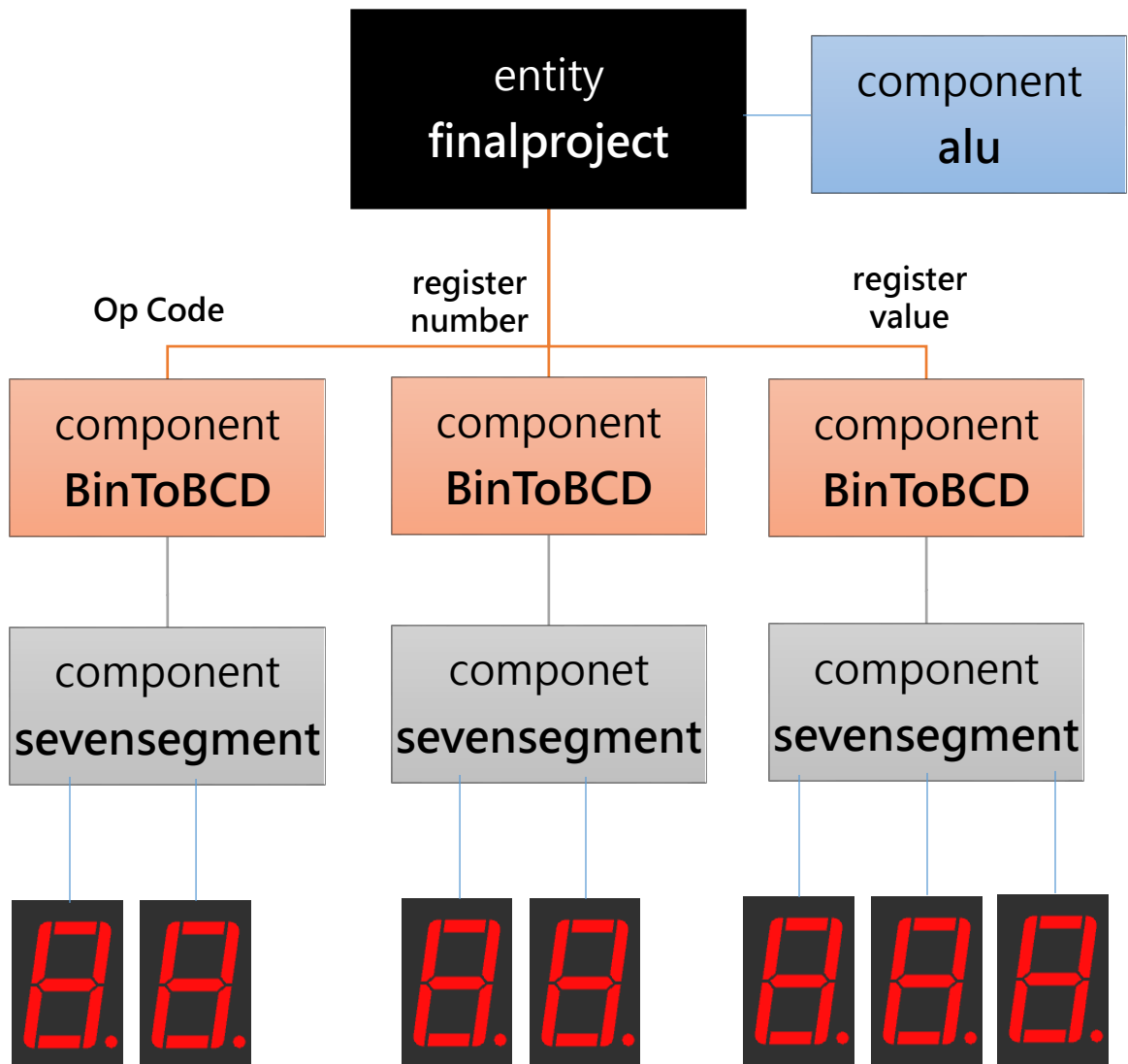
E：把從 ALU 資料輸入回到 Rx

Load 操作：A -> B -> Bhalf -> C -> A

其他操作：A -> B -> Bhalf -> C -> Chalf -> D -> E -> A

電路系統架構與設計

1. 架構圖



2. 架構設計說明

finalporject.vhd 檔案中有一個 finalproject 的 entity，屬於我們整體電路中最主要的控制電路，其中包含了控制指令輸入的狀態機、儲存資料的 registers，由 finalproject 將資料訊號、Op Code 訊號接入作為 ALU 的 component alu 之中，再將運算結果接回 finalproject 內部的 register。

同時 finalproject 會將訊號接往輸出的電路中，因為 register 內部儲存都是二進位的資料，需要先將二進位轉換為 BCD 碼，才能將 BCD 碼接至 Seven-Segment 電路的輸入訊號，最後將 Op Code、register number、register value 輸出，其中後兩個 register 均為各指令中的 register 1。

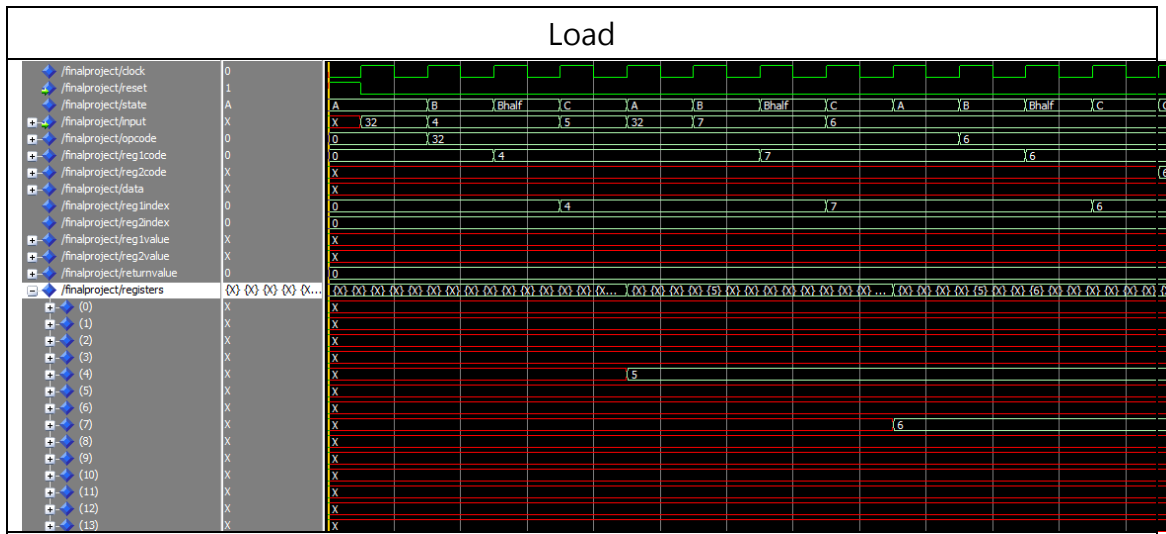
專題時程規劃

		名称	工期	开始	结束
1		架構討論、設計	5天	12/28/2016	01/02/2017
2		工作分配	1天	01/02/2017	01/03/2017
3		finalproject entity 實作	3天	01/02/2017	01/05/2017
4		指令輸入的FSM 實作	3天	01/04/2017	01/07/2017
5		alu component 實作	4天	01/06/2017	01/10/2017
6		BinToBCD component 實作	1天	01/09/2017	01/10/2017
7		sevensegment component 實作	1天	01/09/2017	01/10/2017
8		測試、修正、驗收	5天	01/09/2017	01/14/2017
9		報告製作	2天	01/14/2017	01/16/2017

名称	12月 25 - 12月 31 '16							1月 1 - 1月 7 '17							1月 8 - 1月 14 '17							1月 15 -	
	日	一	二	三	四	五	六	日	一	二	三	四	五	六	日	一	二	三	四	五	六	日	一
架構討論、設計																							
工作分配																							
finalproject entity 實作																							
指令輸入的FSM 實作																							
alu component 實作																							
BinToBCD component 實作																							
sevensegment component 實作																							
測試、修正、驗收																							
報告製作																							

ModelSim 模擬流程

以下分別是 Load, Move, Add, Sub, Multi, Div, And, Or, Nor, Slt 的 Modelsim 模擬以及 Modelsim 操作指令



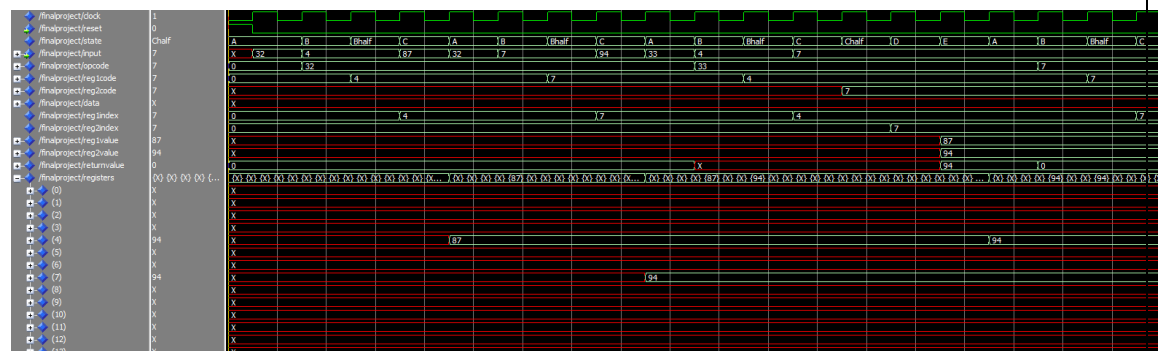
```
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
```

```

force -freeze sim:/finalproject/input 10#5 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#6 0
run 100ps
run 200ps
radix unsigned

```

Move



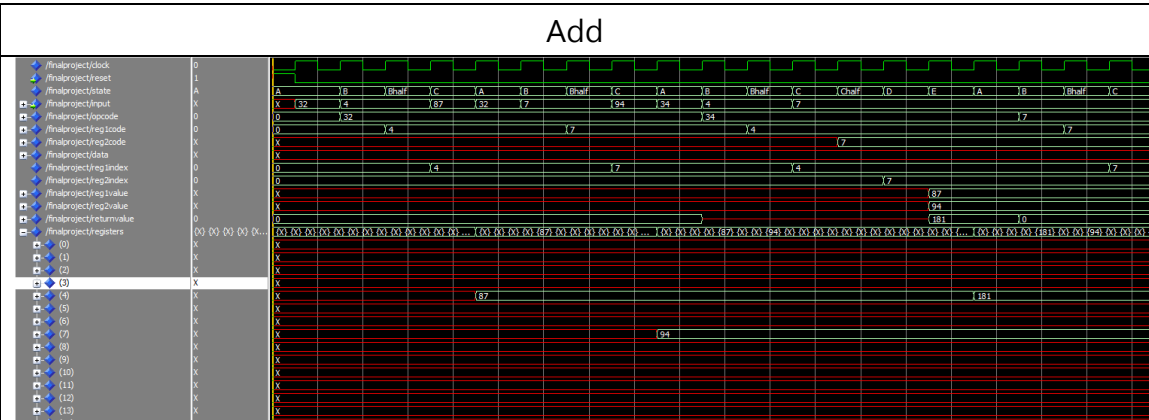
```

add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0

```



```
run 200ps
force -freeze sim:/finalproject/input 10#87 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#94 0
run 100ps
force -freeze sim:/finalproject/input 10#33 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned
```



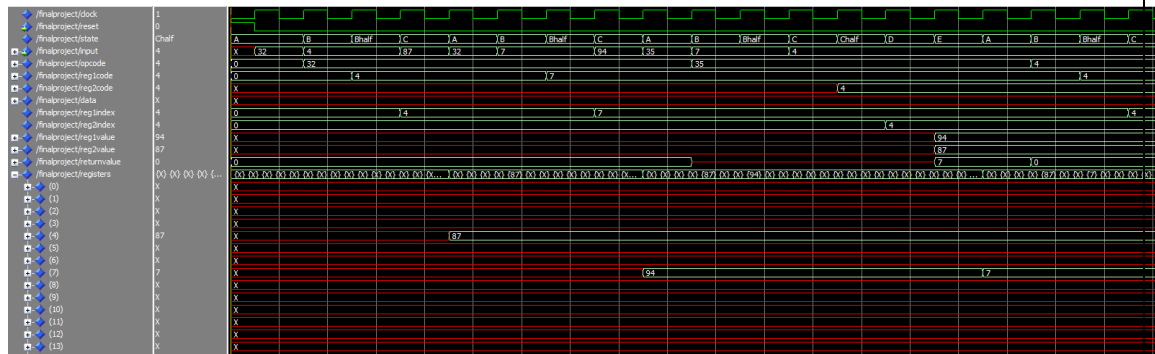
```
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
```

```

run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#87 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#94 0
run 100ps
force -freeze sim:/finalproject/input 10#34 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned

```

Sub



```

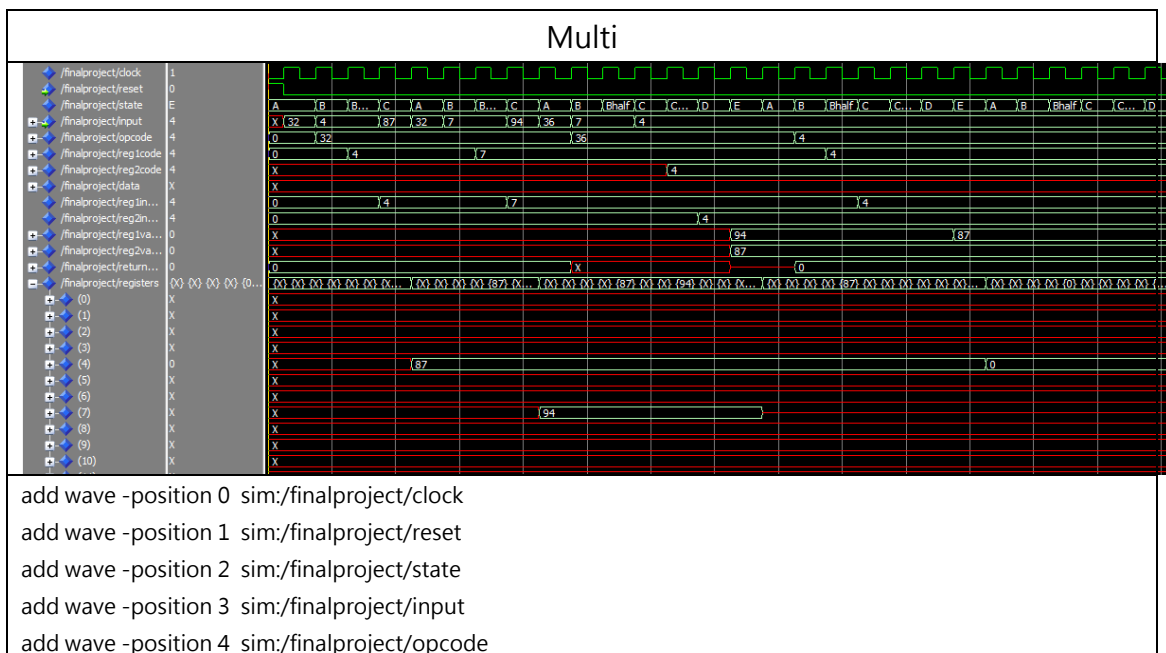
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value

```

```

add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#87 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#94 0
run 100ps
force -freeze sim:/finalproject/input 10#35 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
radix unsigned

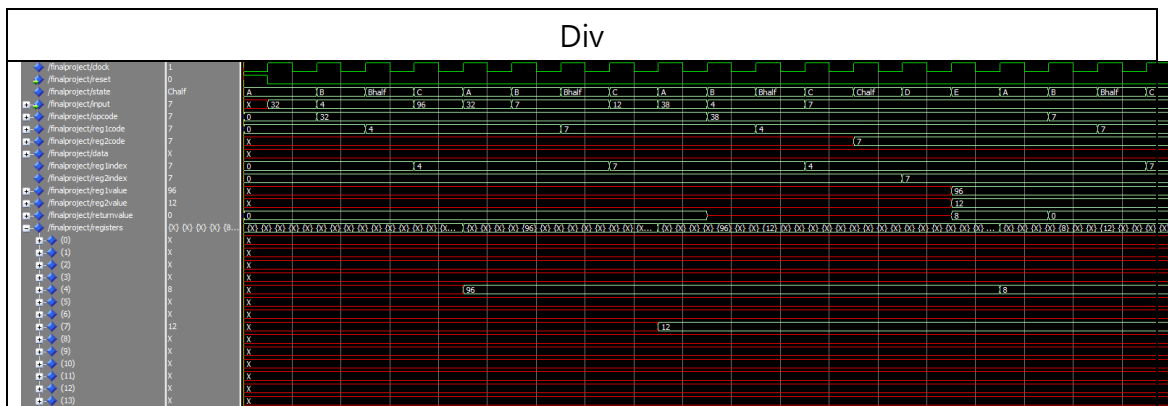
```



```

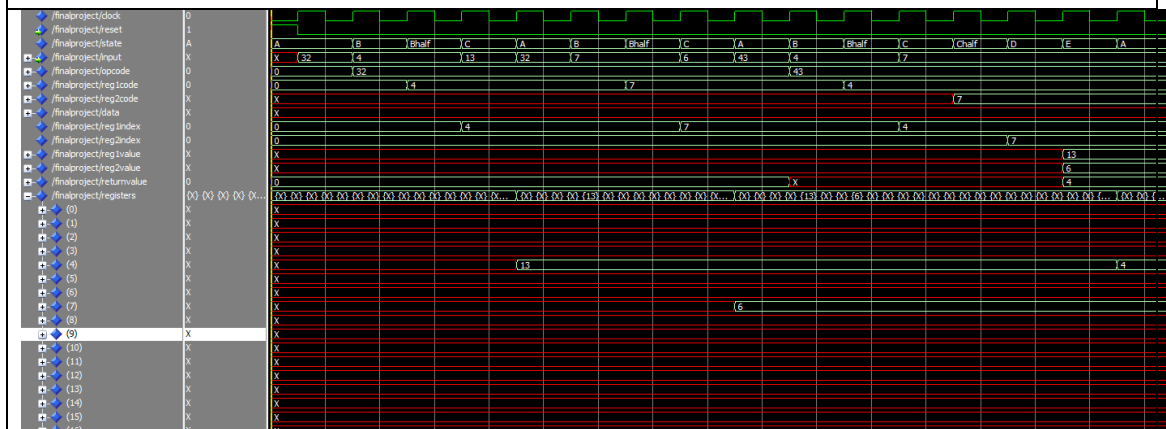
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#96 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#12 0
run 100ps
force -freeze sim:/finalproject/input 10#36 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned

```



```
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#96 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#12 0
run 100ps
force -freeze sim:/finalproject/input 10#38 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned
```

And



```

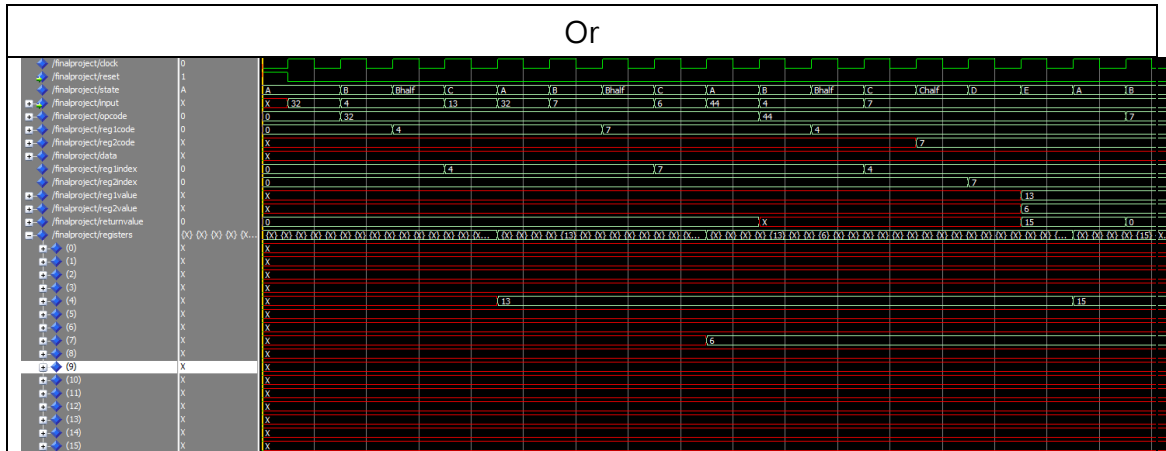
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#13 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#6 0
run 100ps
force -freeze sim:/finalproject/input 10#43 0
run 100ps

```

```

force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned

```



```

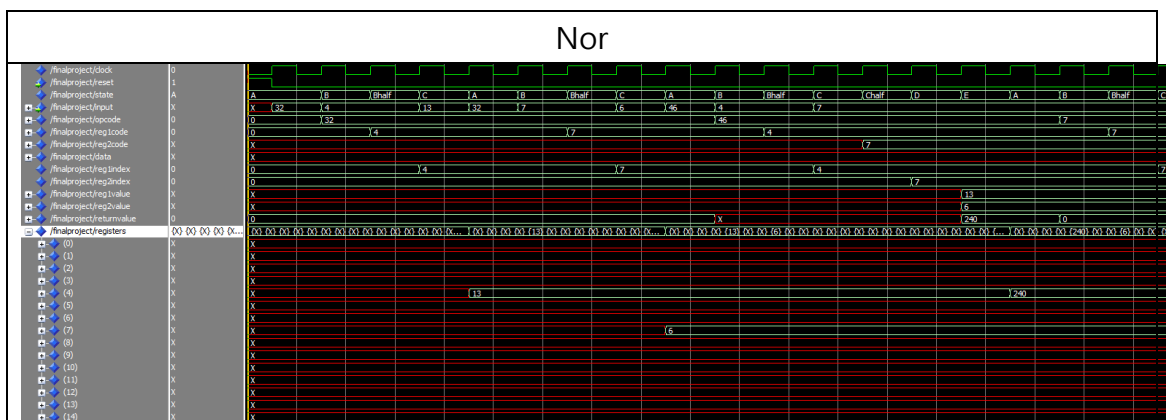
add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#13 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps

```

```

force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#6 0
run 100ps
force -freeze sim:/finalproject/input 10#44 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned

```



```

add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0

```



```

run 200ps
force -freeze sim:/finalproject/input 10#13 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#6 0
run 100ps
force -freeze sim:/finalproject/input 10#46 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned

```



```

add wave -position 0 sim:/finalproject/clock
add wave -position 1 sim:/finalproject/reset
add wave -position 2 sim:/finalproject/state
add wave -position 3 sim:/finalproject/input
add wave -position 4 sim:/finalproject/opcode
add wave -position 5 sim:/finalproject/reg1code
add wave -position 6 sim:/finalproject/reg2code
add wave -position 7 sim:/finalproject/data
add wave -position 9 sim:/finalproject/reg1index
add wave -position 10 sim:/finalproject/reg2index
add wave -position 11 sim:/finalproject/reg1value
add wave -position 12 sim:/finalproject/reg2value
add wave -position 13 sim:/finalproject/returnvalue
add wave -position end sim:/finalproject/registers
restart -force
force -freeze sim:/finalproject/clock 0 0, 1 {50 ps} -r 100

```

```
force -freeze sim:/finalproject/reset 1 0
run 50ps
force -freeze sim:/finalproject/reset 0 0
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#13 0
run 100ps
force -freeze sim:/finalproject/input 10#32 0
run 100ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
force -freeze sim:/finalproject/input 10#6 0
run 100ps
force -freeze sim:/finalproject/input 10#48 0
run 100ps
force -freeze sim:/finalproject/input 10#4 0
run 200ps
force -freeze sim:/finalproject/input 10#7 0
run 200ps
radix unsigned
```

工作分配

梁皓鈞：

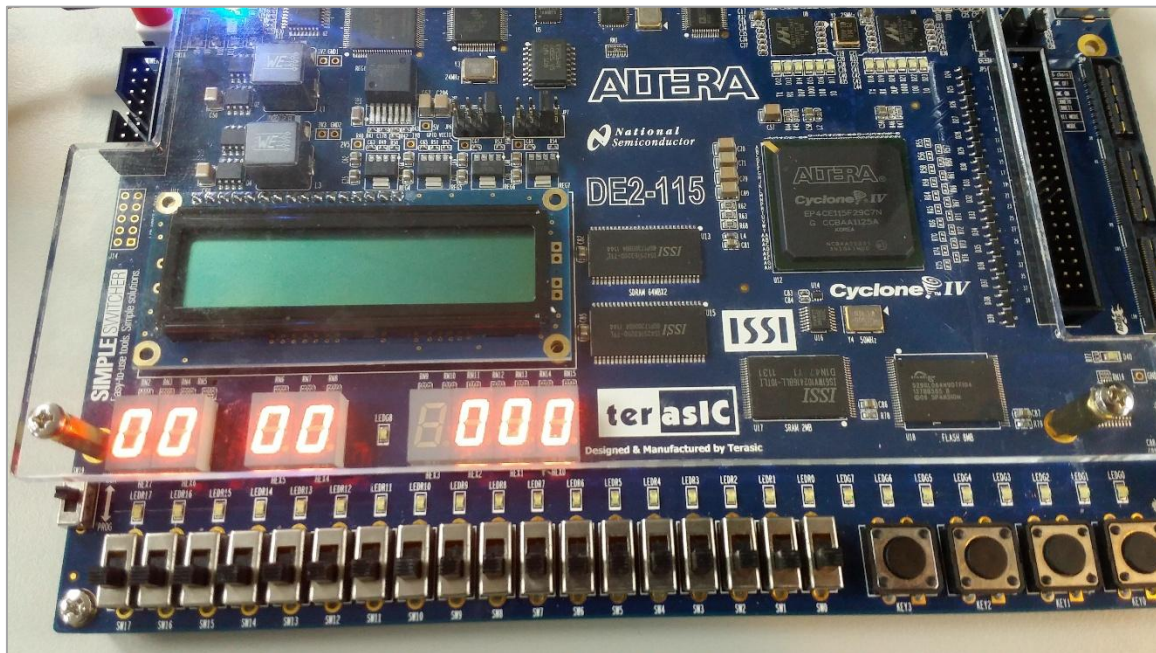
期末專題程式碼	30 %
期末報告作品介紹	3 %
期末報告系統功能介紹	5 %
期末報告 ModelSim 模擬流程	6 %
期末報告結論	3 %
期末專題心得	3 %
總計	50 %

洪晟毅：

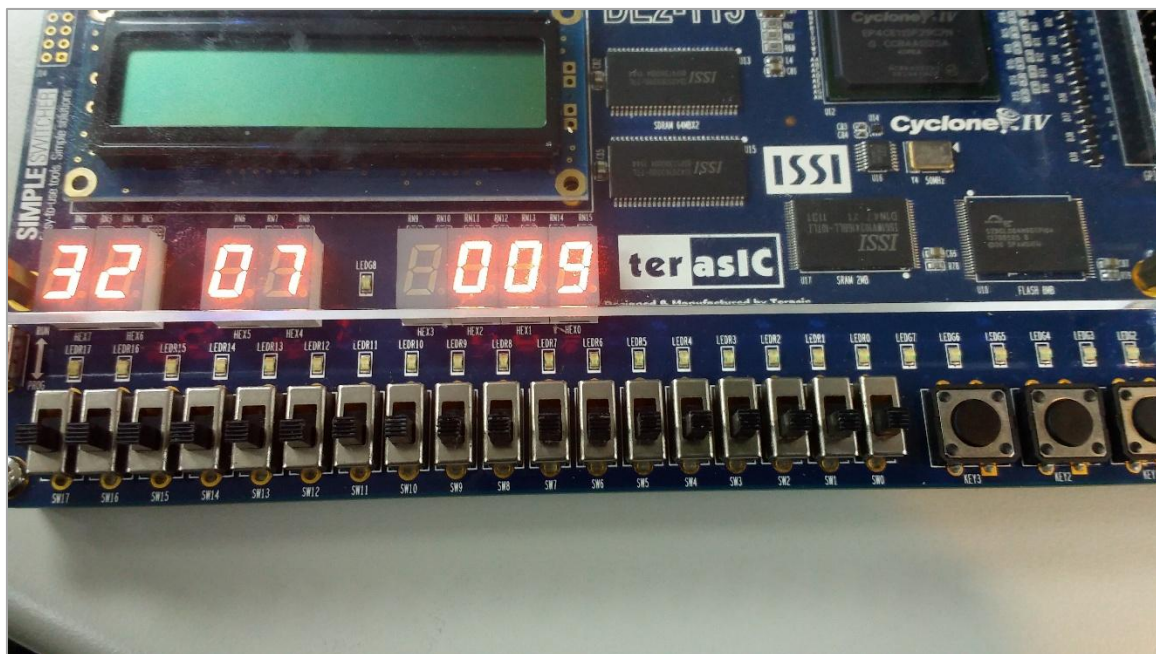
期末專題程式碼	10 %
期末專題電路圖架構	10 %
期末專題實驗結果	20 %
期末專題時程	5 %
期末專題心得	5 %
總計	50 %

實驗結果

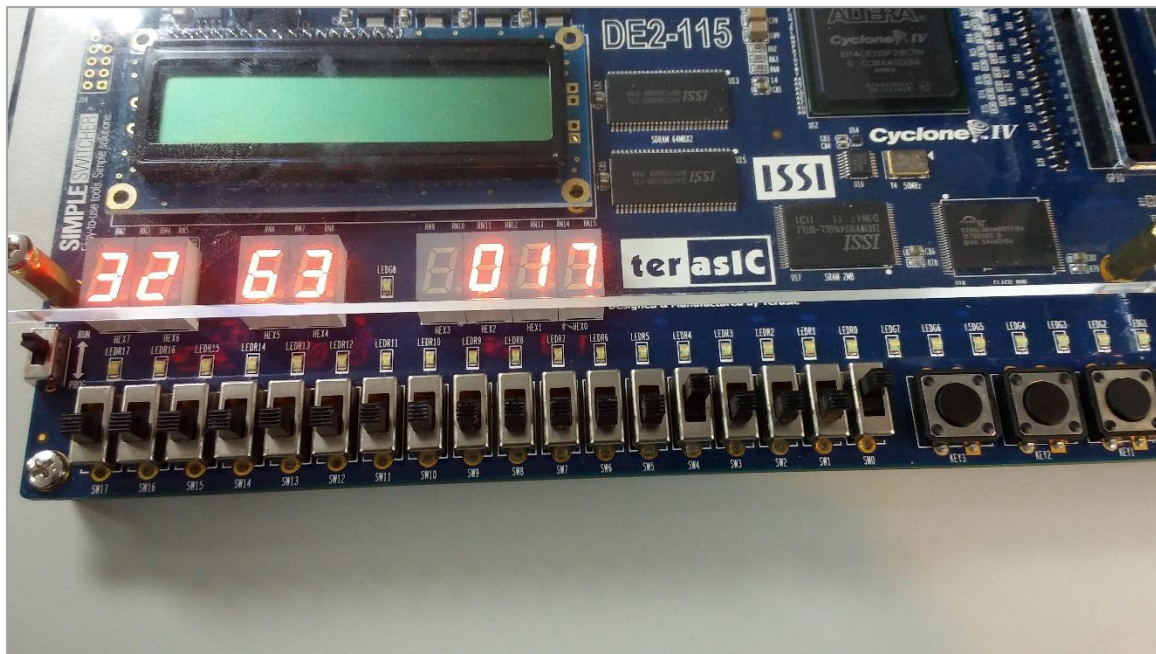
1. 啟動時的狀態



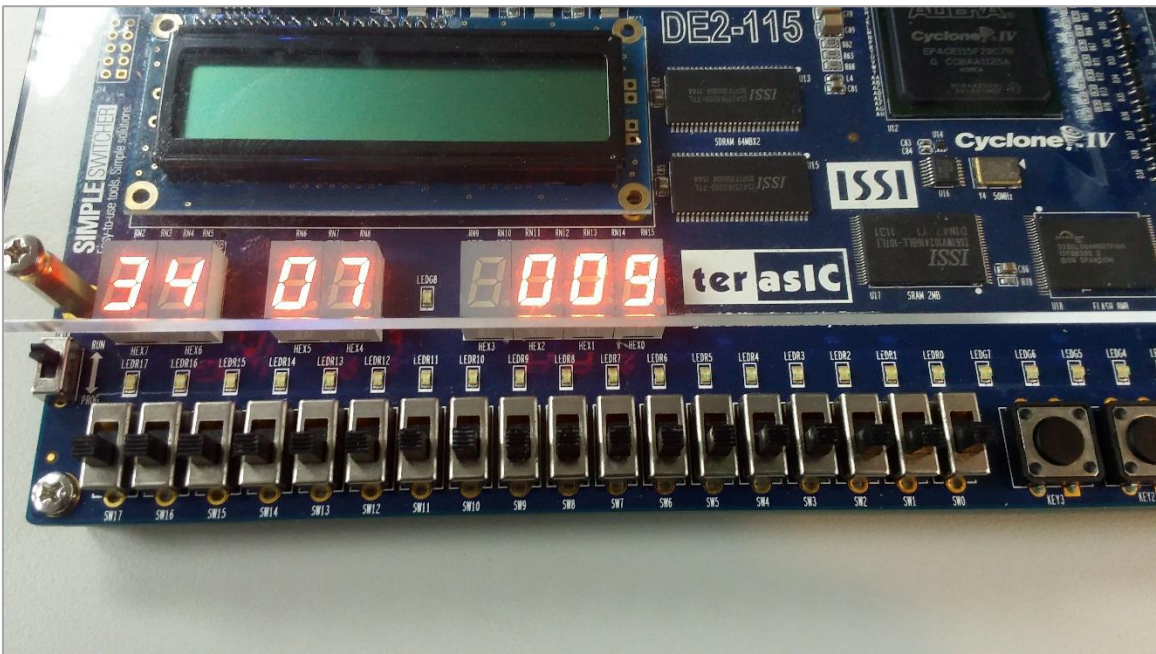
2. 執行 Load 指令，將 $9_{10}=00001001_2$ 的資料載入至編號 07 的暫存器

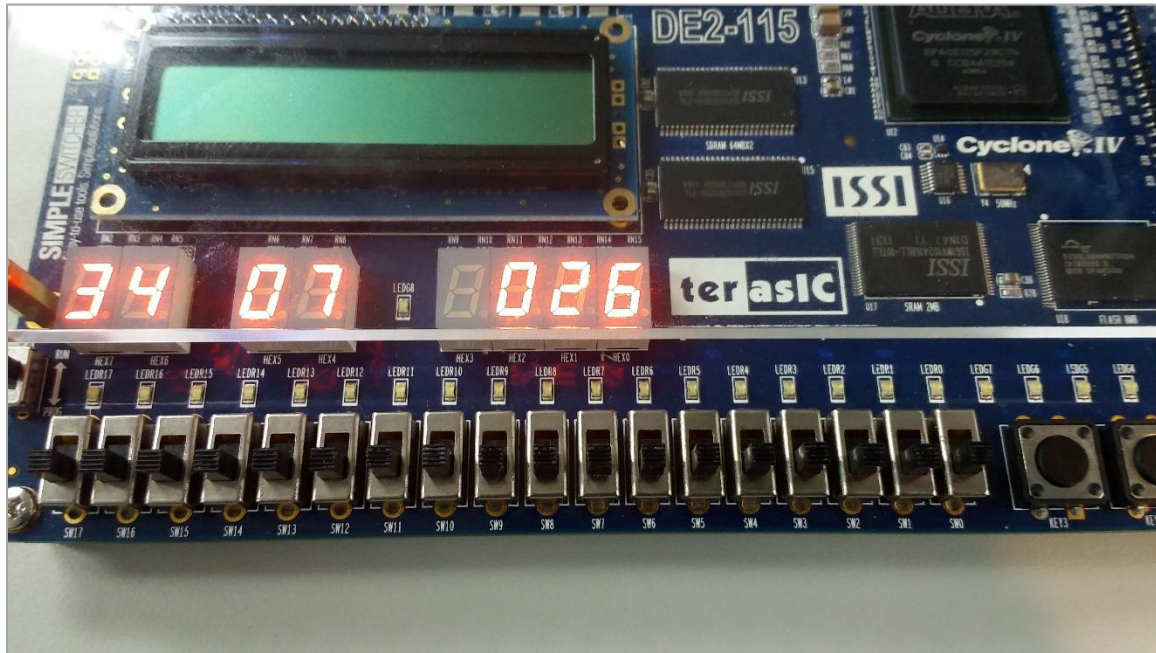


3. 再次 Load 資料 $17_{10}=00010001_2$ 至編號 63 的暫存器

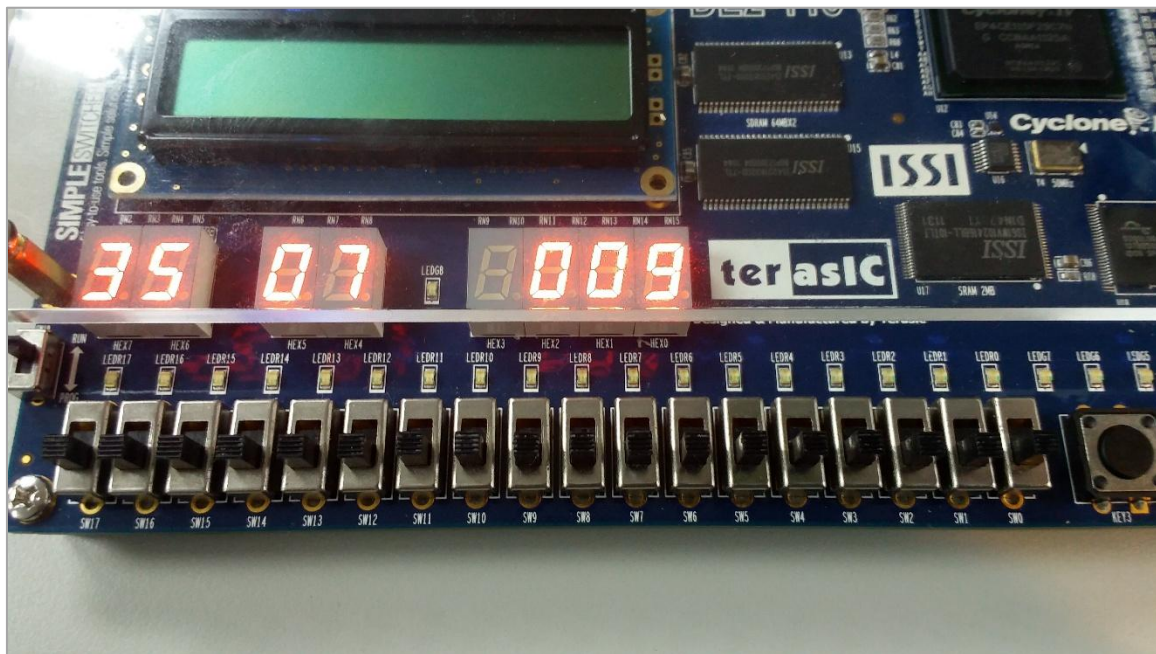


4. 進行 Add 運算，將暫存器 7 及暫存器 63 的資料相加後儲存至暫存器 7，第一張圖為相加之前，第二張為相加後的結果。

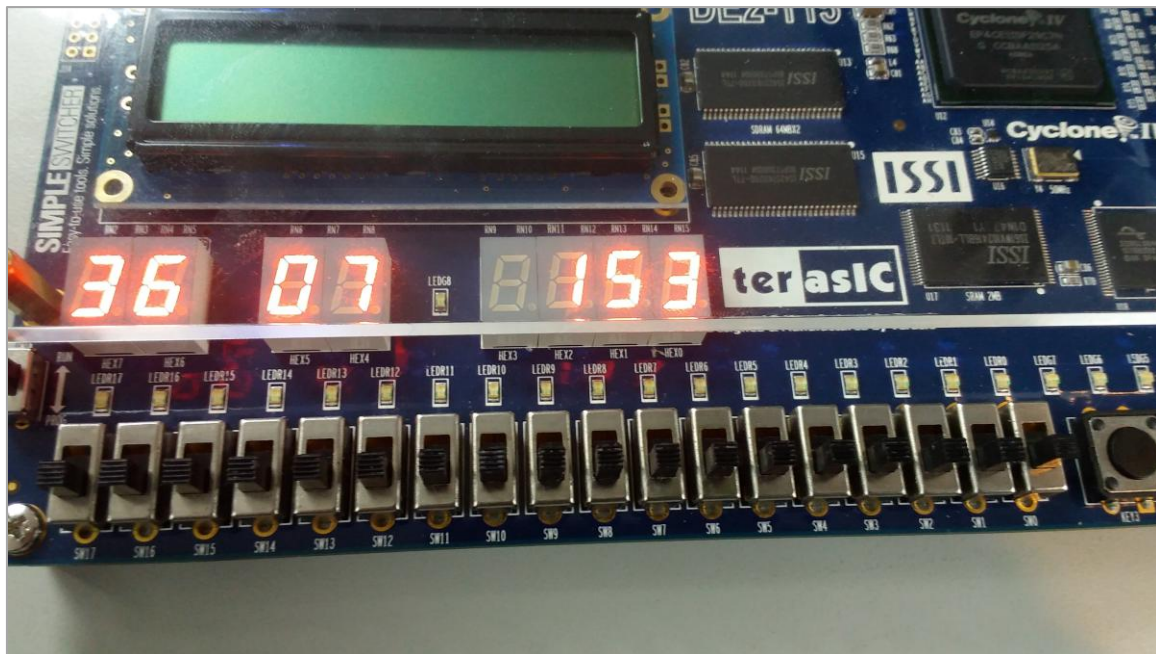




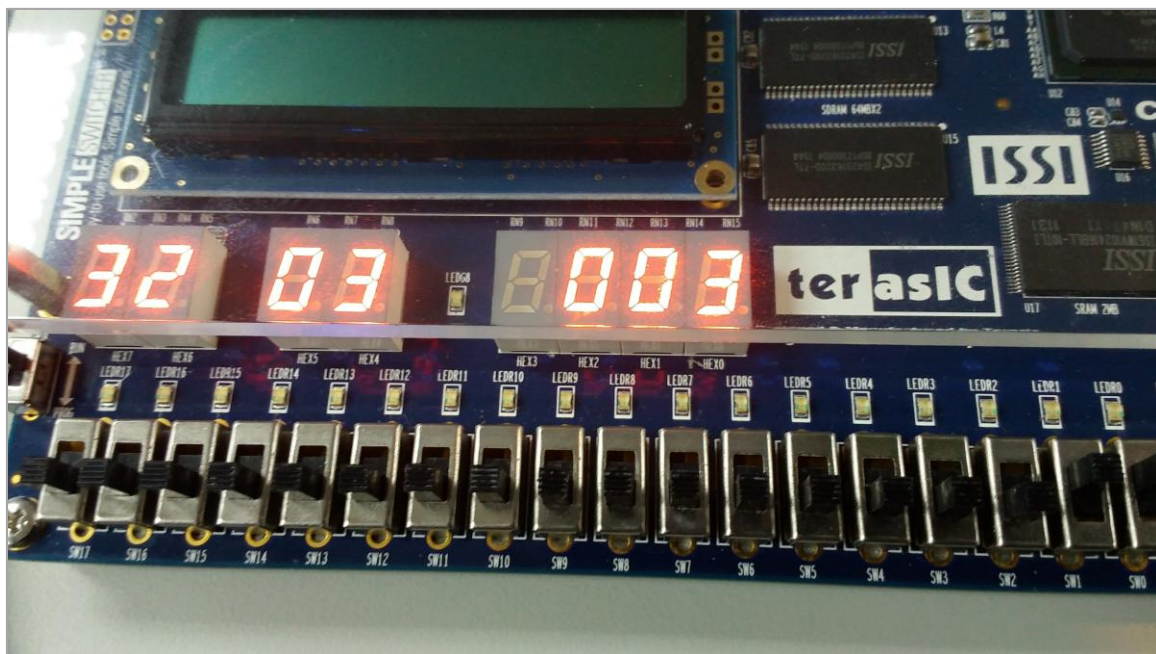
5. Sub 運算結果，由暫存器 7 (26) 減去暫存器 63 (17) 的結果。

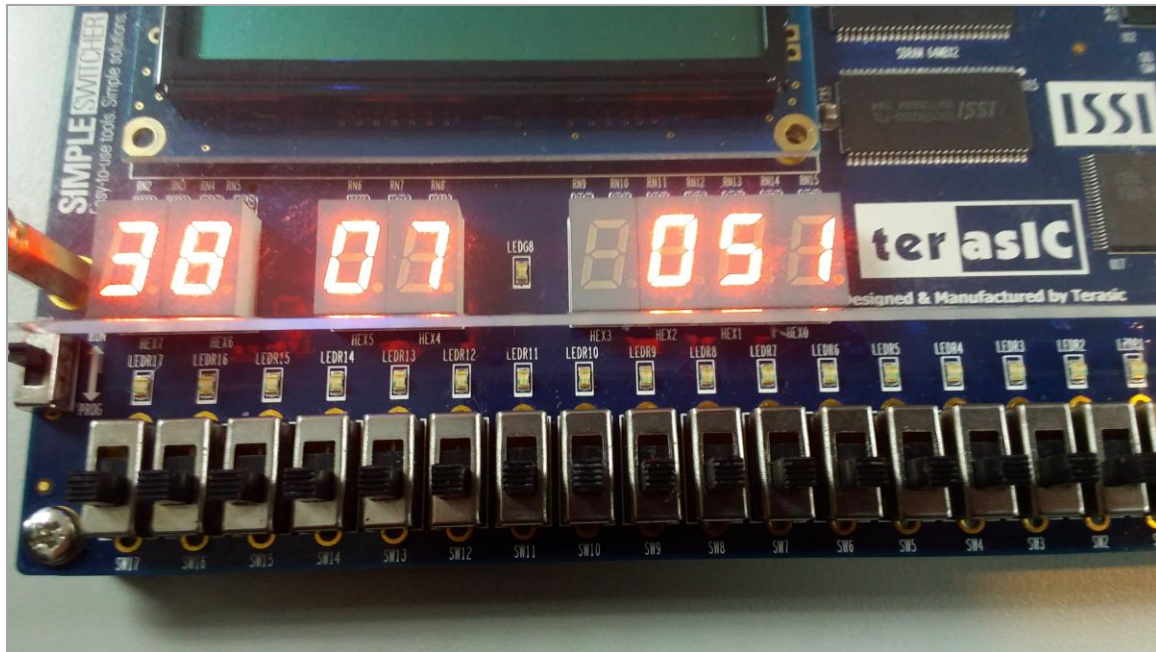


6. Mul 運算，暫存器 7 (9) 與暫存器 63 (17) 相乘的結果

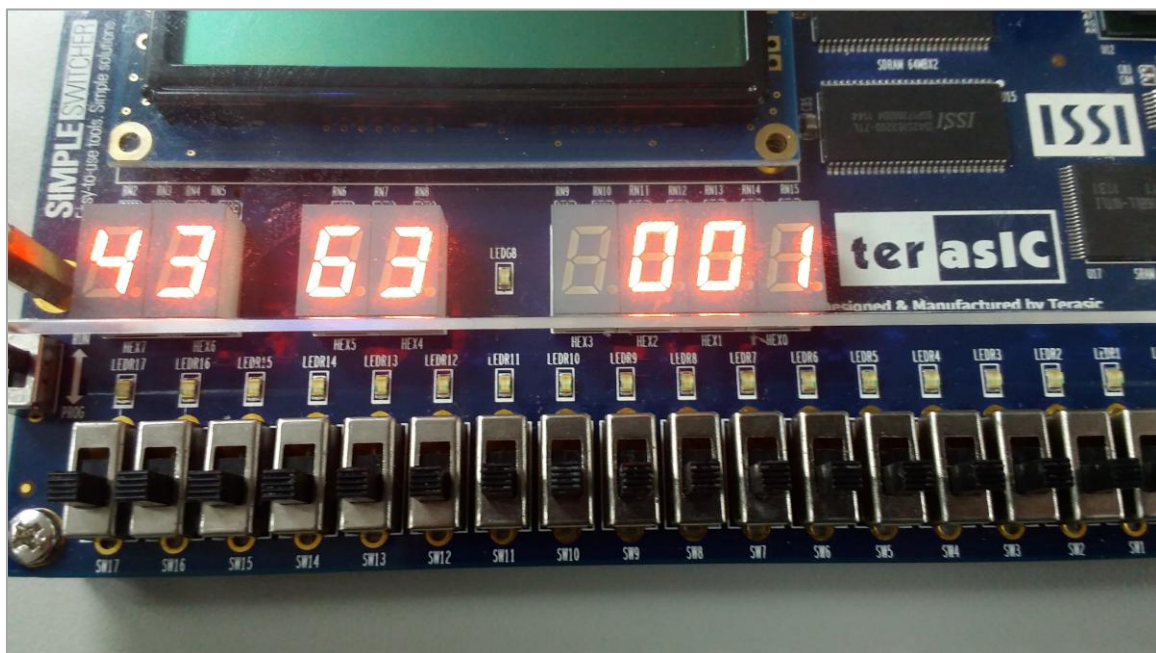


7. Div 運算，為確保整數運算，另外 Load 資料 3 至暫存器 3，再計算暫存器 7 (153) 除以暫存器 3 (3) 之結果。

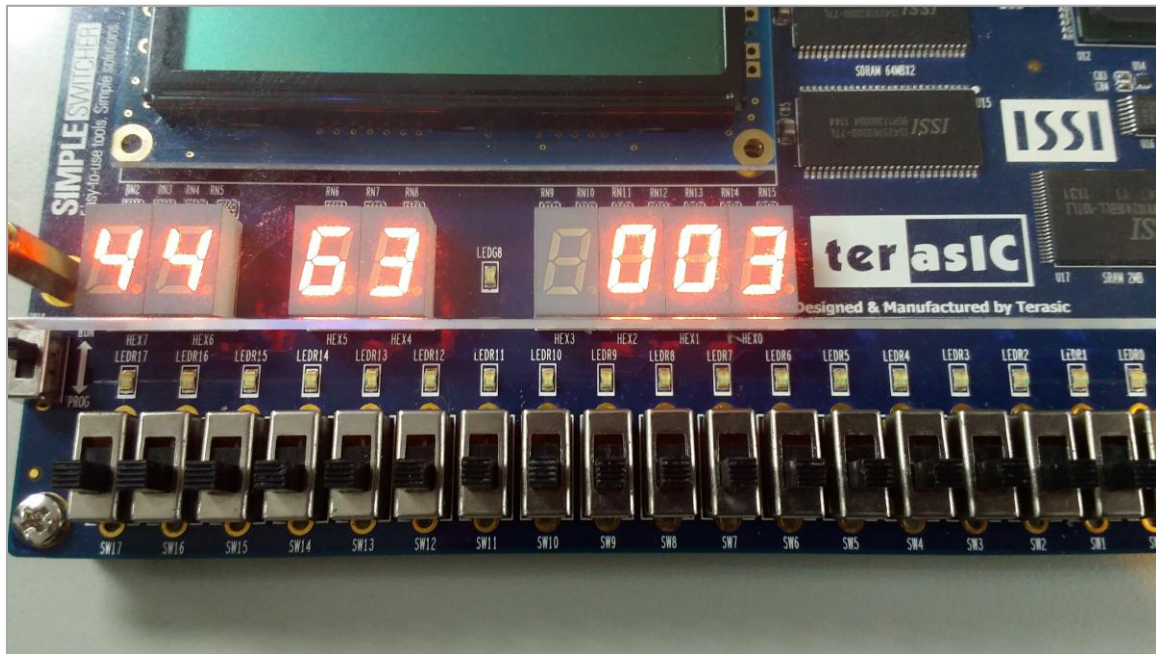




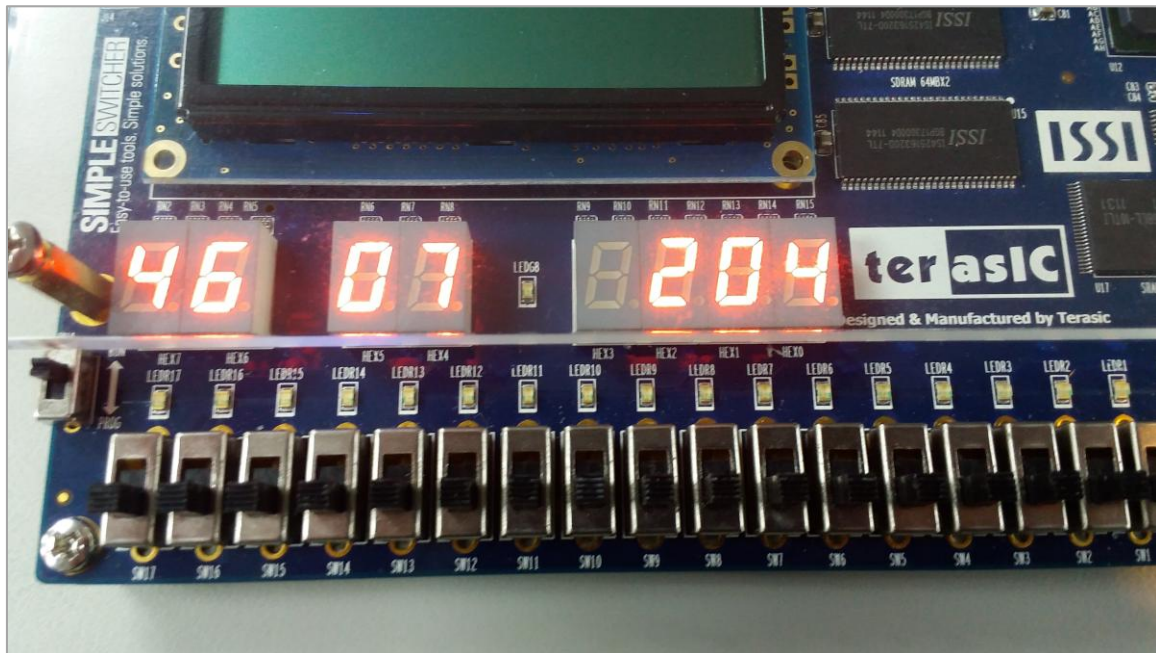
8. And 運算，為暫存器 63 ($17_{10}=00010001_2$) 與暫存器 3 ($3_{10}=00000011_2$) 運算之結果，並儲存至暫存器 63。



9. Or 運算，為暫存器 63 ($1_{10}=00000001_2$) 與 暫存器 3 ($3_{10}=00000011_2$) 運算之結果。



10. Nor 運算，為暫存器 7 ($51_{10}=00110011_2$) 與 暫存器 63 ($3_{10}=00000011_2$) 運算之結果。



此圖為暫存器 63 (3_{10}) 作為 register 1 與暫存器 7 (204_{10}) 作為 register 2 進行 slt 的運算結果。



心得

梁皓鈞 (104360098) :

這一次實驗在程式碼部份，大部份都是由我編寫，所以對於程式的整體問題了解比較深入。這一次問題所在在於延遲問題。在同一個 State 下大量接線會出現延遲問題。

例如：

```
code <= input;  
  
reg1index <= to_integer(unsigned(code));
```

在 clock 那一瞬間，input 的確有給 code，但 reg1index 卻沒有馬上接受到 to_integer(unsigned(code))

有見於這樣的情況，所以我在每一個有需要到這種大量接線的地方都拆開兩個 state 進行，這樣便成功解決這種問題。這樣的問題在其他組別也有出現，包括先前數個完成的組別都有這樣的情況。因此最後認為是延遲導致的問題。

基本上引用了 IEEE 的 Library 在 ALU 的製作上是非常快的。不過這一次也了解到原來在 VHDL 中，std_logic_vector 是一種儲存型態，而不是一個數值。因此 std_logic_vector 不能直接進行數值上的操作，因此例如 A std_logic_vector 加 B std_logic_vector, 需要寫成 std_logic_vector(unsigned(A) + unsigned(B)) 才行。

其他基本上只要了解狀態機就可以完成，先前我有打算使用 Mealy State Machine 進行編寫，可是因為還是不知道 Mealy State Machine 在做什麼，所以仍然選擇了 Moore State Machine。我認為在這一次實驗中，使用 Moore State Machine 與 Mealy State Machine 是幾乎相同的，因此採用了 Moore State Machine。

此外在這一次期末專題因為操作量太大，因此 ModelSim 發揮了很大作用。原本我沒有用 ModelSim 時非常難進行 Debug，幾乎是完全找不出問題所在在哪。直到下定決心使用 ModelSim 時才明白到原來 ModelSim 在 Debug 作用上起了很重要的角色。在我尋找解決延遲問題的時候，可以從 ModelSim 馬上找出問題所在進行測試。

最後在這一次期末專題中我認為是一個大整合，把所有由學期初到現在所有知識的一個整合作業。透過這一次實驗我把之前對於 VHDL 的一些誤解也一同消除了，也對於大學一年級的 Digital Logic Design 加深了熟悉。總括而言，微算機系統的課程讓我喚醒了 Digital Logic Design 的知識以及加深了對於 VHDL 語言的操作。

洪晟毅 (104590048)：

此次期末專題中，有礙於我在本學期的課業中稍重，恰好碰上期末考週而無法專心投入專題之中，因此主要的架構及重要的程式碼均由皓鈞撰寫完成的，我僅有參與到前半的架構規劃、討論及後半的部分 components 串接、測試、報告製作，總體算是蠻可惜的，但幸虧回顧程式碼時發現都是利用先前所使用的語法、概念進行組合，所以也不算損失太大。

後半在進行測試時，發現執行 Load 指令時輸入的資料會亂跳，我與皓鈞兩人不斷重新回顧程式碼、檢查架構及各 components，卻遲遲無法找到錯誤的部分，但因為錯誤僅發生在 Load 指令輸入要儲存至暫存器的資料此步驟，且有時正常有時失敗，這幾點讓我們感到詭異且百思不得其解。

最初以為是按鈕的彈跳問題，故特地尋找一個簡易的 debounce components，快速將其串接至電路中，但狀況並沒有改變。之後懷疑可能是 VHDL 的特性問題，嘗試著將一些可能影響的部分分離，包含將更改狀態機的下一次狀態步驟分離出新的 process 處理、引入新的 enable 訊號來作為控制 ALU 啟用的開關，結果還是發生相同錯誤。

最後是由皓鈞嘗試利用 ModelSim 作檢查，發現其問題出現在 process 的延遲特性，如果將某個 signal 訊號 A 接至訊號 B，再由訊號 B 接至訊號 C，此時訊號 C 獲得的會是舊的 B 訊號，而非新的訊號 B (來自訊號 A)，我進行反思之後認為，process 的設計為了避免衝突才会有這個延遲的特性在，而且發現之前所使用的 FSM 之所有能實現也是因為 process 具有此延遲的特性。

一直以來，電路與時序圖都是我的弱項，所以我盡可能地避開接觸這些部分，但在此次期末專題中碰上的問題，若不嘗試使用 ModelSim 的模擬時序圖來檢測，恐怕根本不可能發現問題所在。透過此專題，讓我體悟到即使我只是撰寫 VHDL，我也應該試著去理解電路的原理、嘗試去接觸不擅長的部分，而非一直逃避，否則下次碰上類似的問題，可能我會永遠也解決不了。

結論

總結來說，這次實驗十分順利，雖然中途因為板子延遲問題而出現，但因為 ModelSim 的功能強大關係，在短短兩天就已經找出問題所在並且計劃好 debug 流程。在使用 IEEE 的 Library 上也比較熟悉，因此在整個實驗中都非常順利。

以往實驗中教授都不容許我們使用其他功能，需要每一個都自己製作，是為了要我們把基礎打好。而在這一次終於解放了，可以讓我們自行學習 VHDL 進行編寫，沒有限制。因此在這一次實驗中，從 Altera 的 Documentation 中我們學習到更多的 VHDL 語法以及一些比較 tricky 的應用。

最後我們完成了基本部份的 75 分以及三項加分題目，分別是乘法, 除法, 64 暫存器，各加 10 分。獲分 105。