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

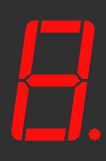
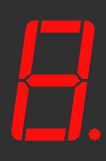
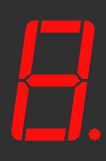
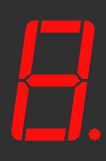
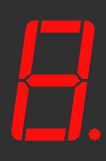
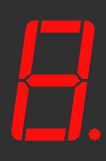
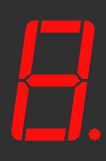
電路系統架構與設計

1. 架構圖

**Op Code**

**register   
number**

**register  
value**



1. 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



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

|  |
| --- |
| Load |
|  |
| 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 |
|  |
| 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 |

|  |
| --- |
| Multi |
|  |
| 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#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 |

|  |
| --- |
| Div |
|  |
| 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 |

|  |
| --- |
| Or |
|  |
| 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 |

|  |
| --- |
| Nor |
|  |
| 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 |

|  |
| --- |
| Slt |
|  |
| 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 :

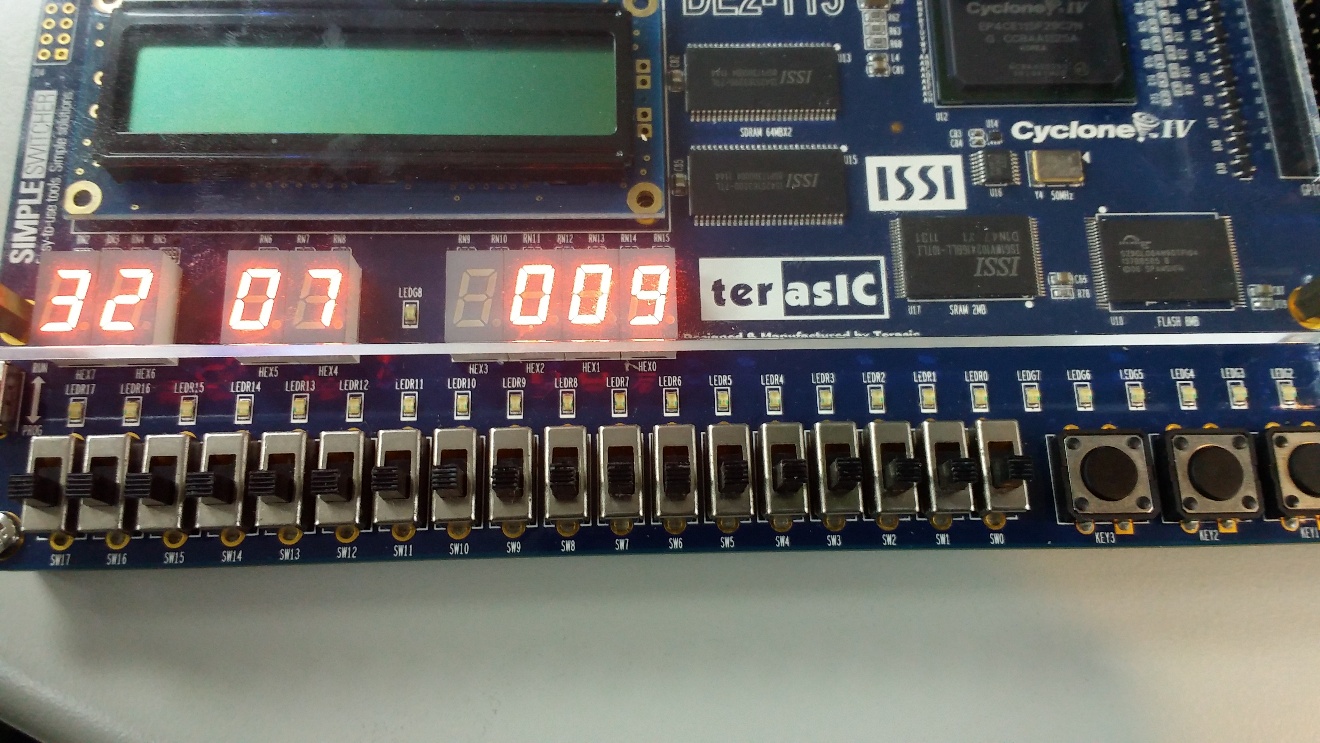
|  |  |
| --- | --- |
| 期末專題程式碼 | 10 % |
| 期末專題電路圖架構 | 10 % |
| 期末專題實驗結果 | 20 % |
| 期末專題時程 | 5 % |
| 期末專題心得 | 5 % |
| 總計 | 50 % |

實驗結果

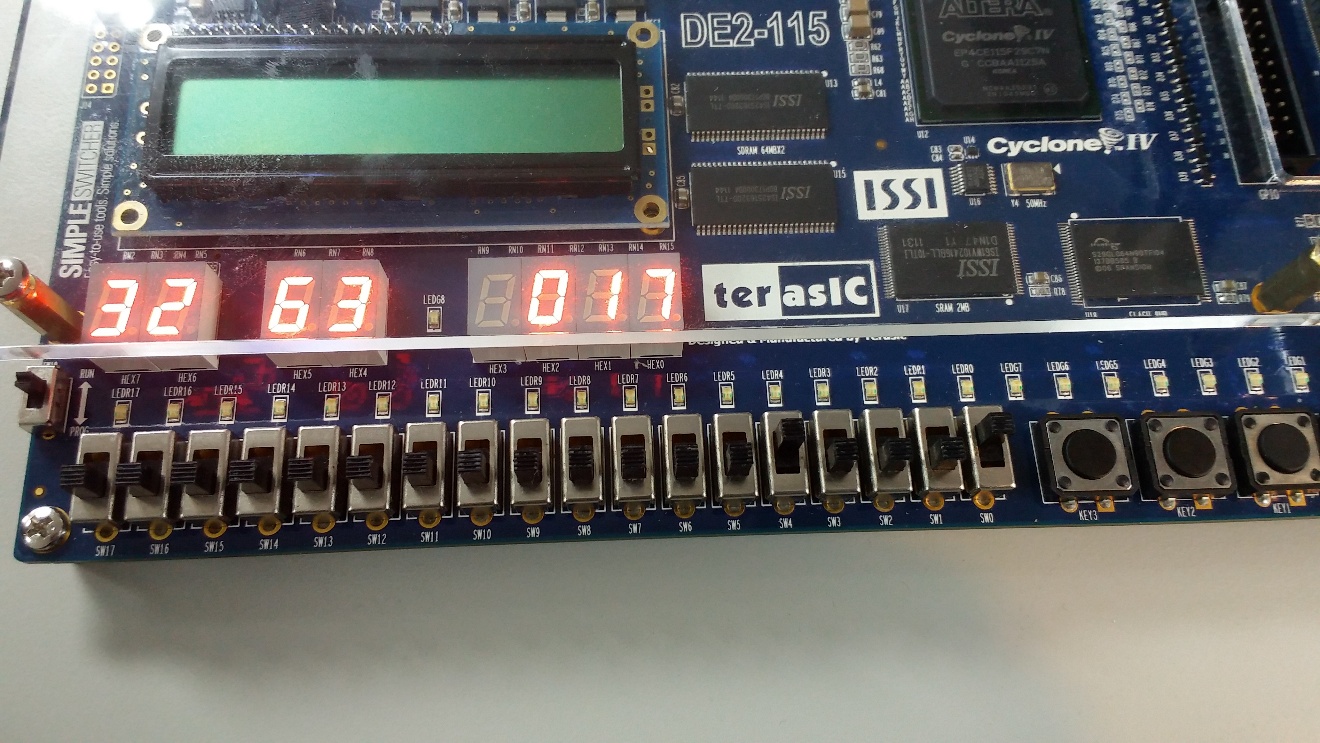
1. 啟動時的狀態

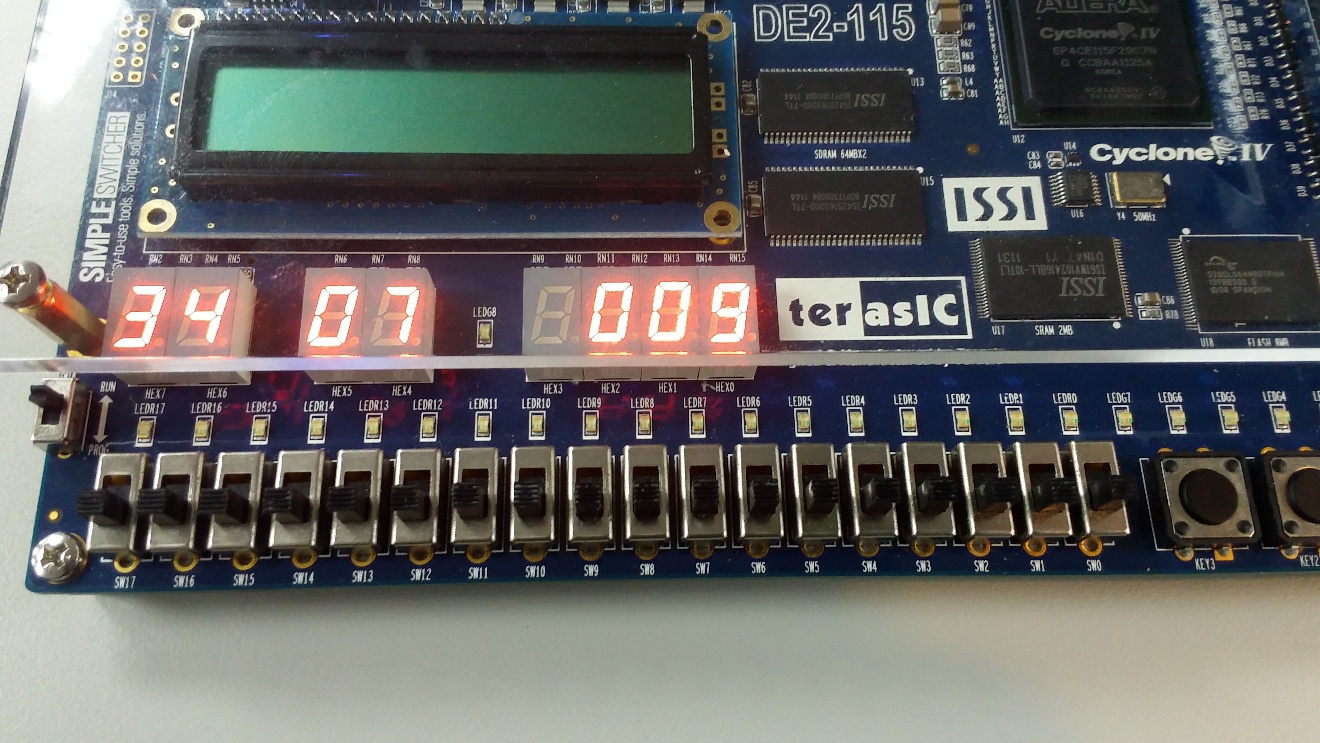


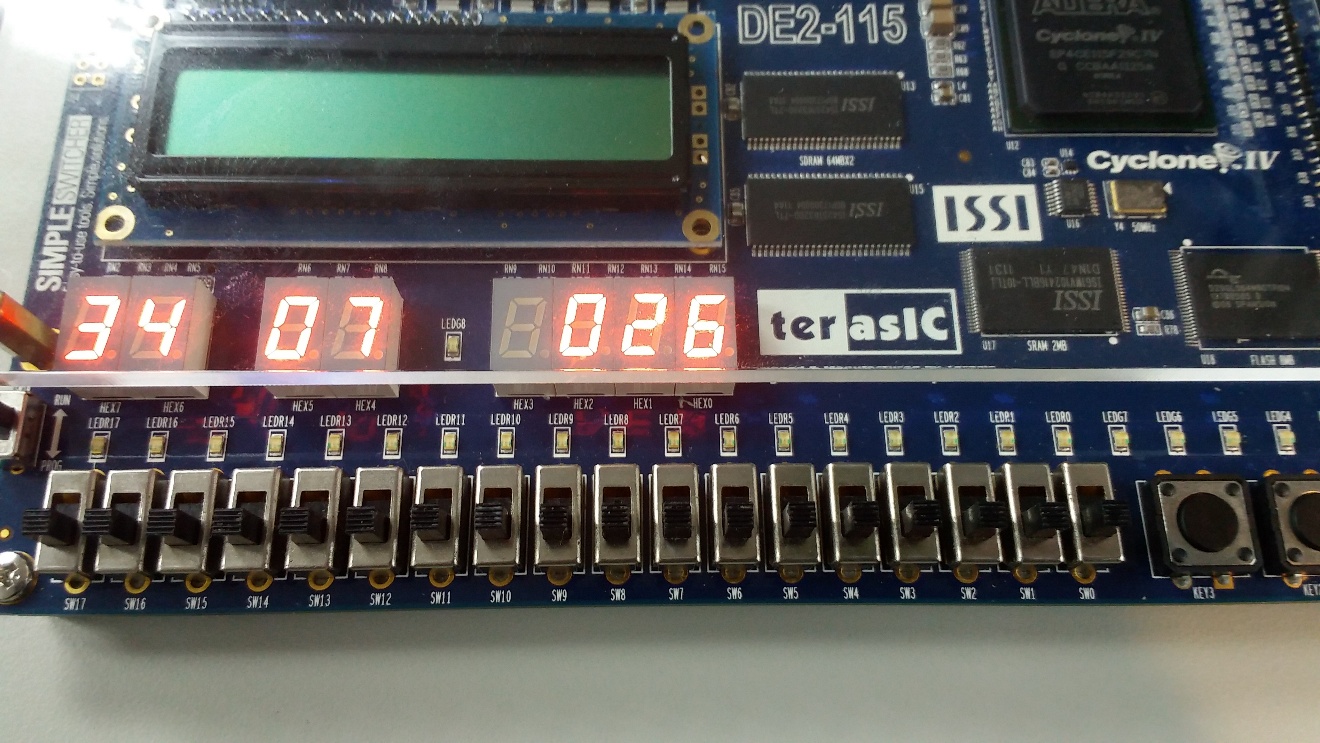
1. 執行Load指令，將910=000010012的資料載入至編號07的暫存器



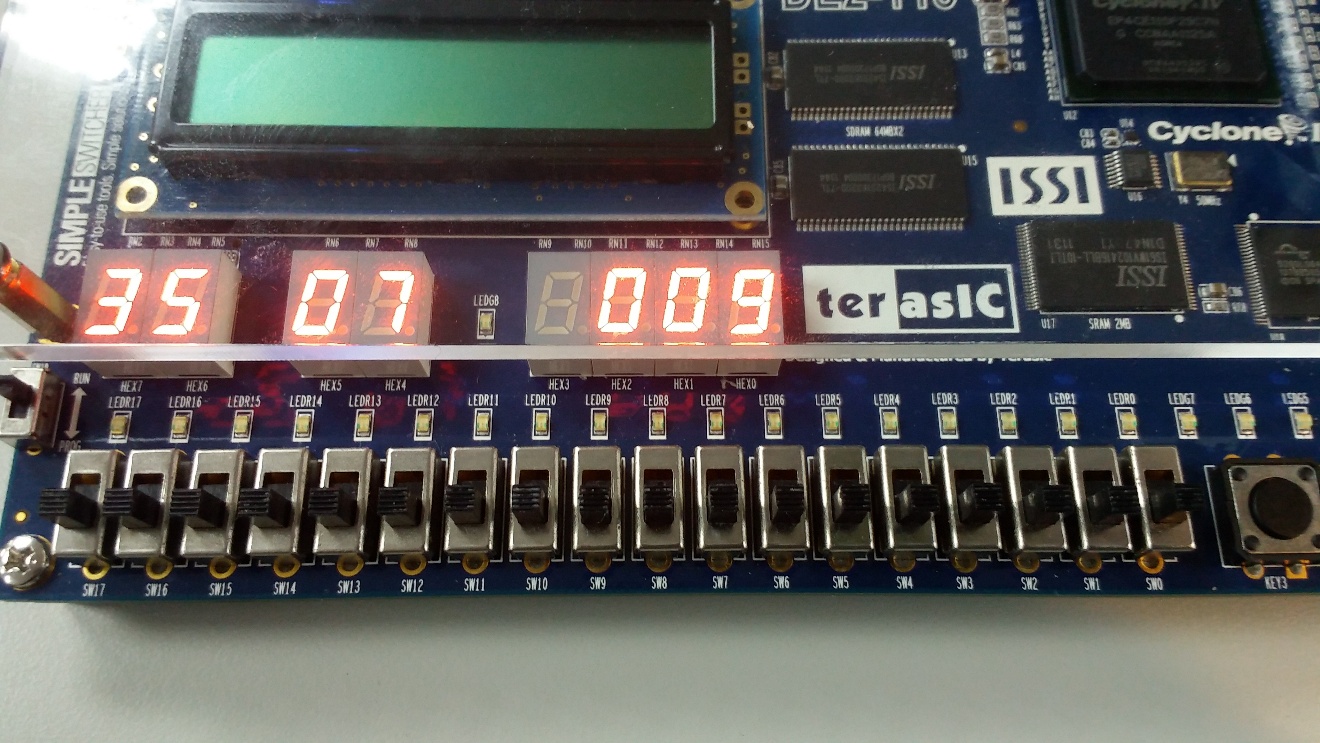
1. 再次Load資料1710=000100012至編號63的暫存器



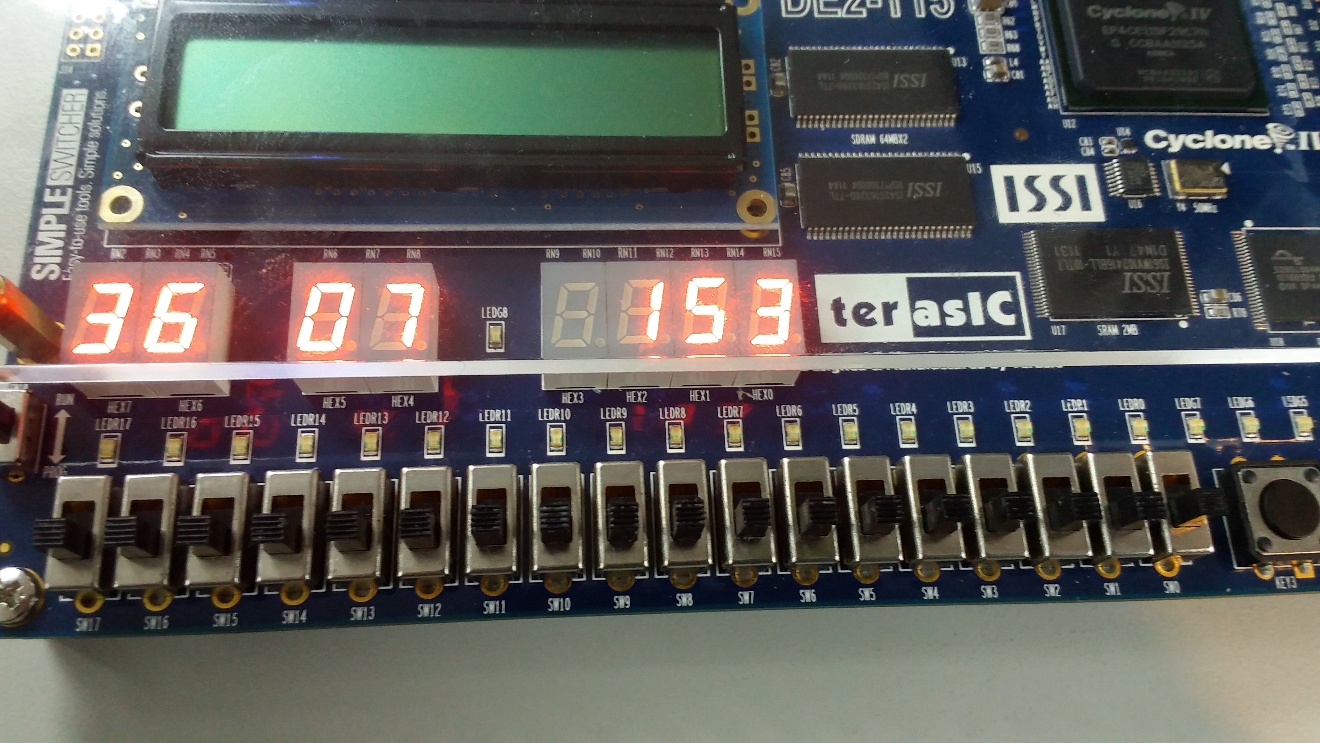
1. 進行Add運算，將暫存器7及暫存器63的資料相加後儲存至暫存器7，第一張圖為相加之前，第二張為相加後的結果。



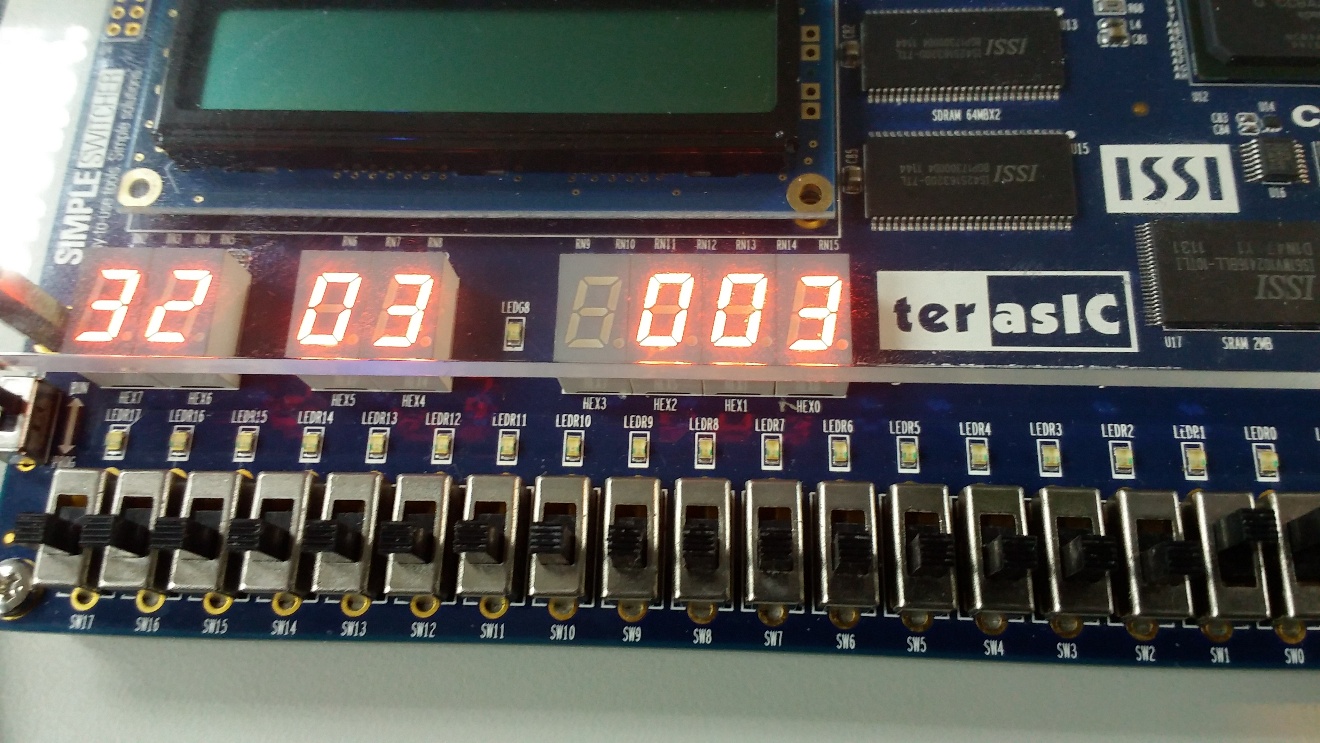
1. Sub運算結果，由暫存器7 (26) 減去暫存器63 (17) 的結果。

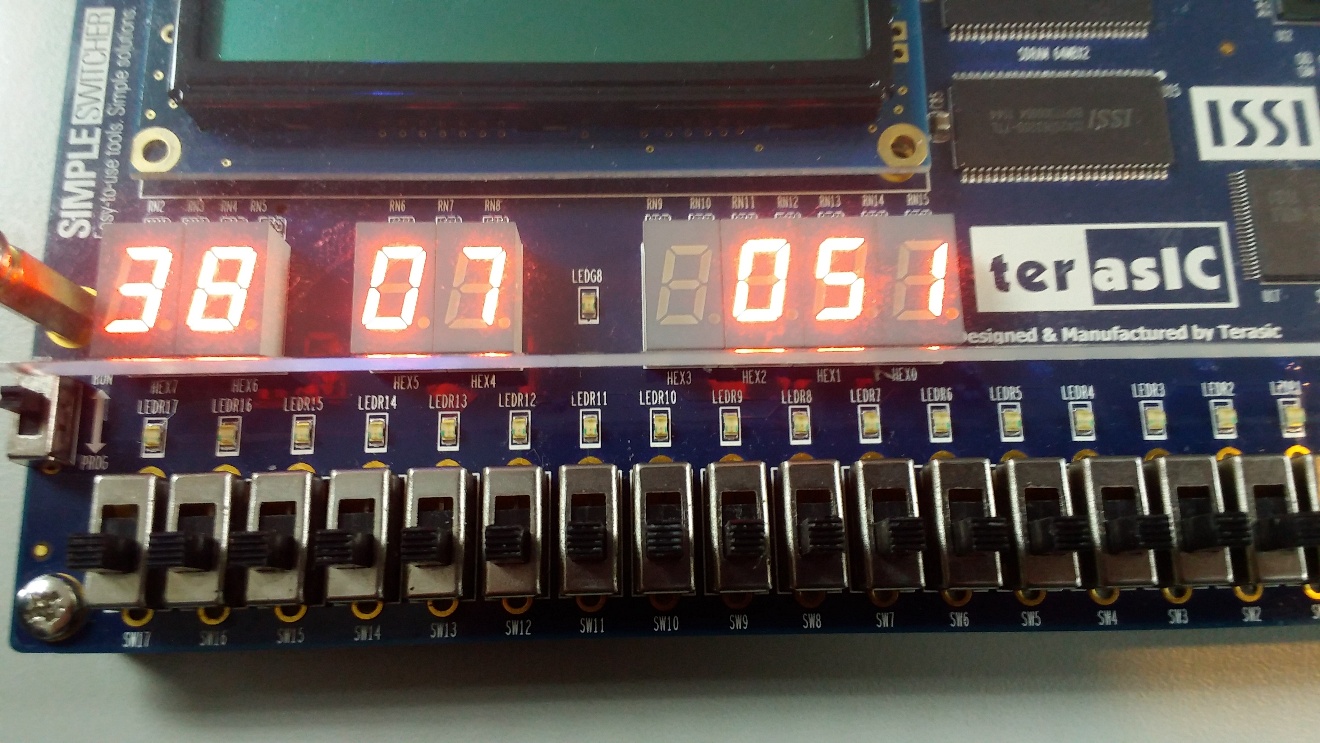


1. Mul運算，暫存器7 (9) 與暫存器63 (17) 相乘的結果

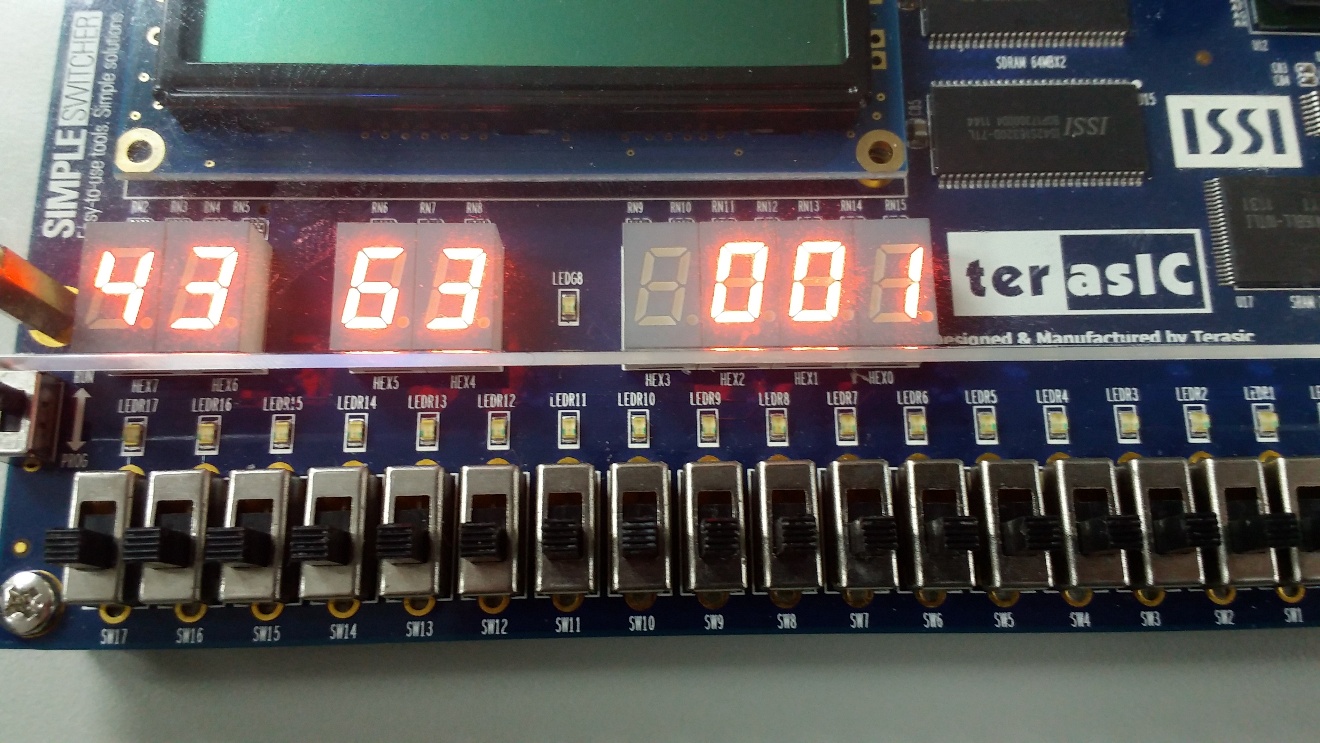


1. Div運算，為確保整數運算，另外Load資料3至暫存器3，再計算暫存器7 (153) 除以暫存器 3 (3) 之結果。

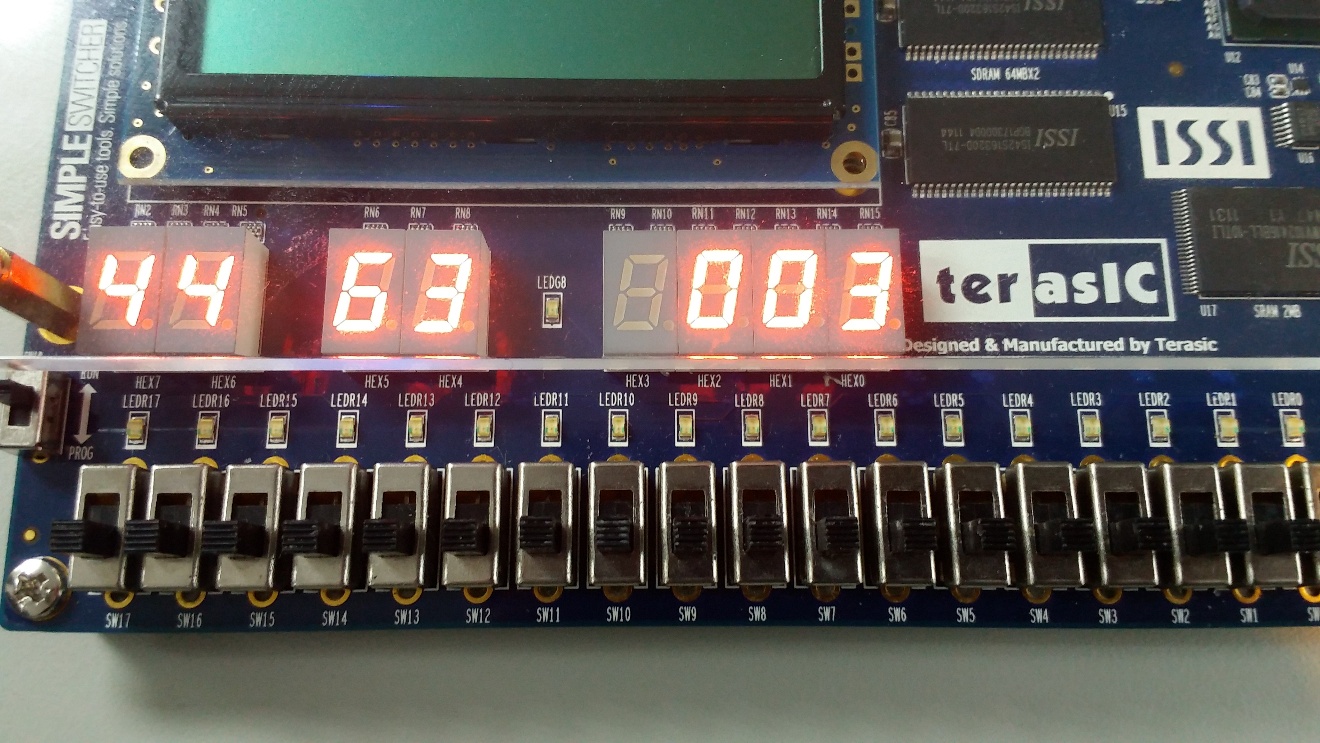




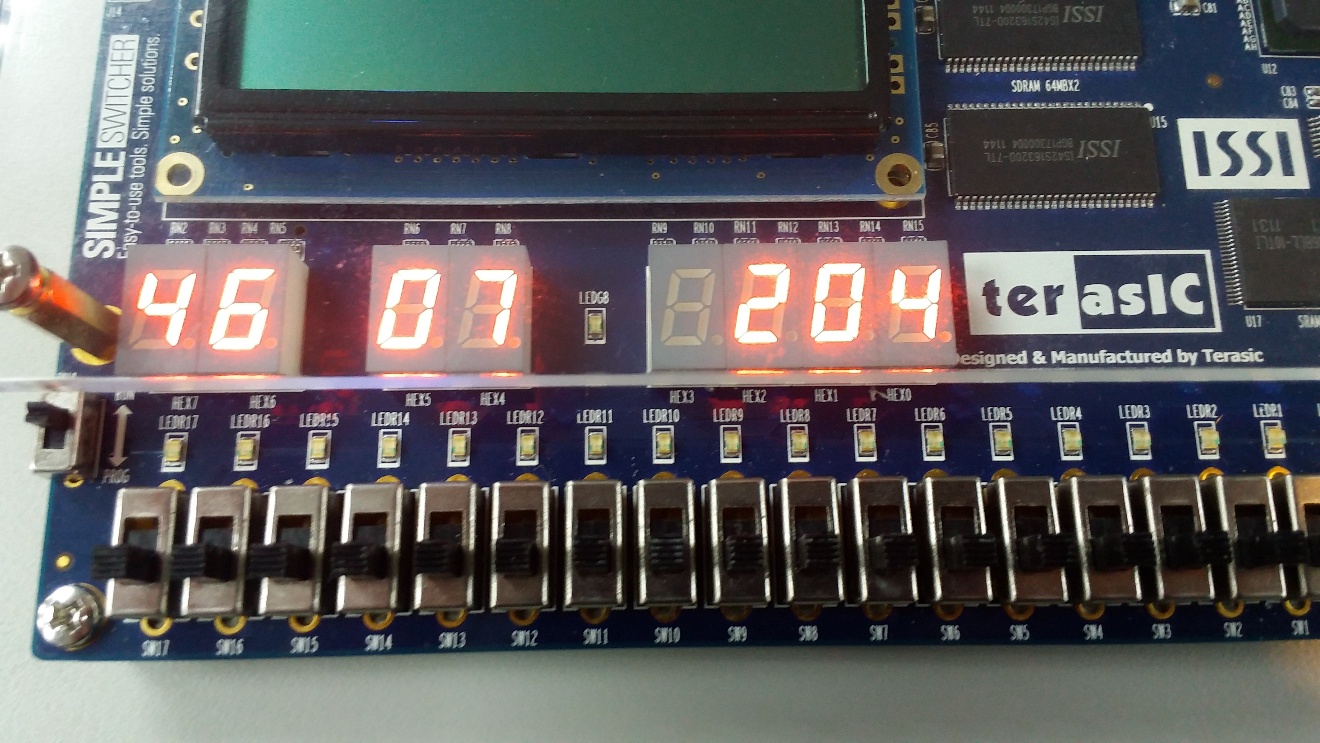
1. And運算，為暫存器63 (1710=000100012) 與暫存器 3 (310=000000112) 運算之結果，並儲存至暫存器 63。



1. Or 運算，為暫存器 63 (110=000000012) 與 暫存器 3 (310=000000112) 運算之結果。

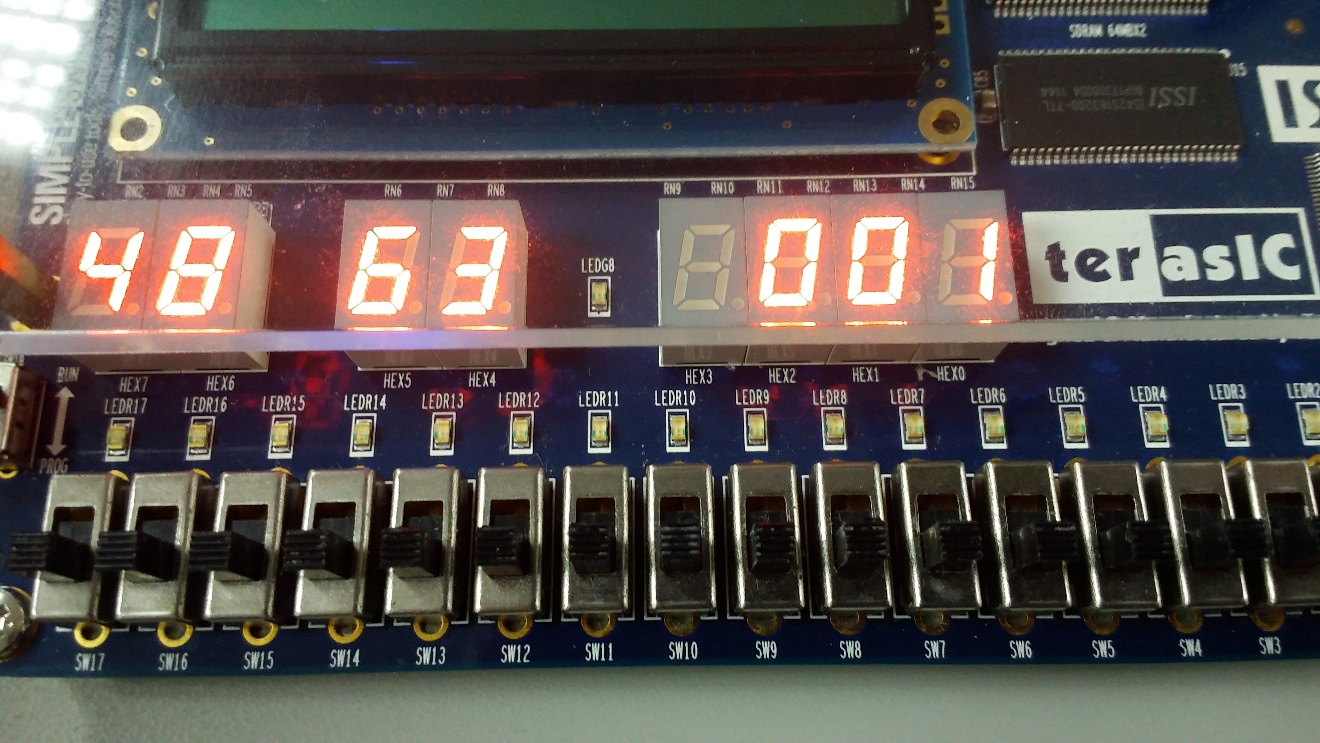


1. Nor運算，為暫存器 7 (5110=001100112) 與暫存器63 (310=000000112) 運算之結果。



1. Slt運算，此指令較為複雜，功能為當register 1 小於 register 2時，將register 1設為1，否則將register 1設為0。

此圖為暫存器63 (310) 作為register 1與暫存器7 (20410) 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。

此外在這一次期未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。

最後在這一次期未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會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。