

計算機組織 Midterm Project

ALU Design

老師：朱守禮老師

組別：第 10 組

學生：11027112 陳逸駿

11027140 陳芄睿

11027147 邱峻彥

11027149 游婕歆

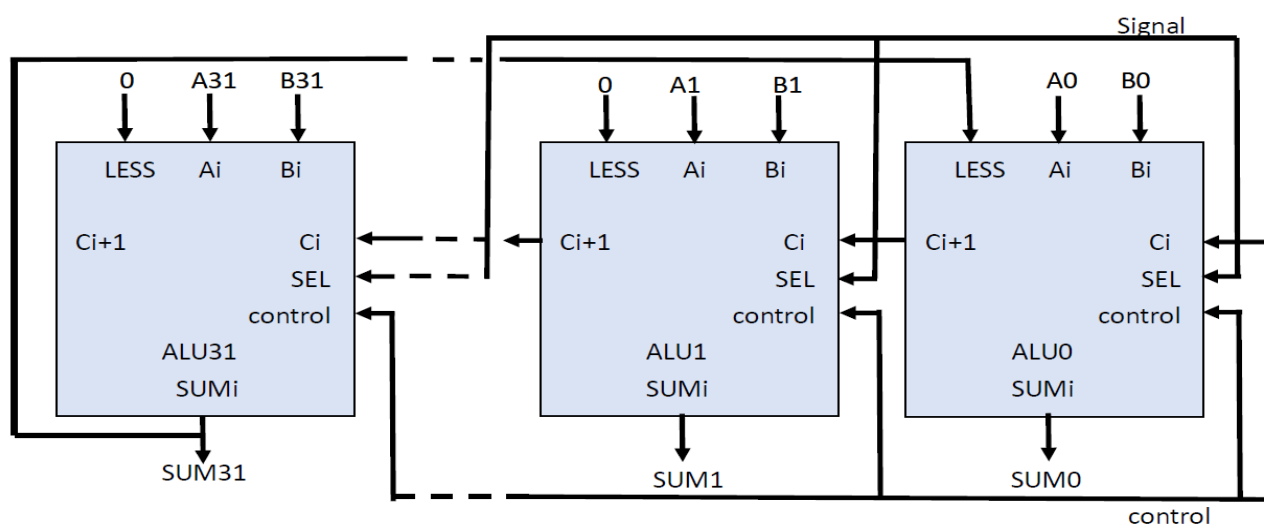
1. 背景

使用 Verilog 和 modelsim 並用課程講義的方法為基礎，設計 ALU 與乘法器。功能包含 AND, OR, ADD, SUB, SLT, SRL(邏輯右移), MULTU(無號數乘法)。

2. 設計重點說明

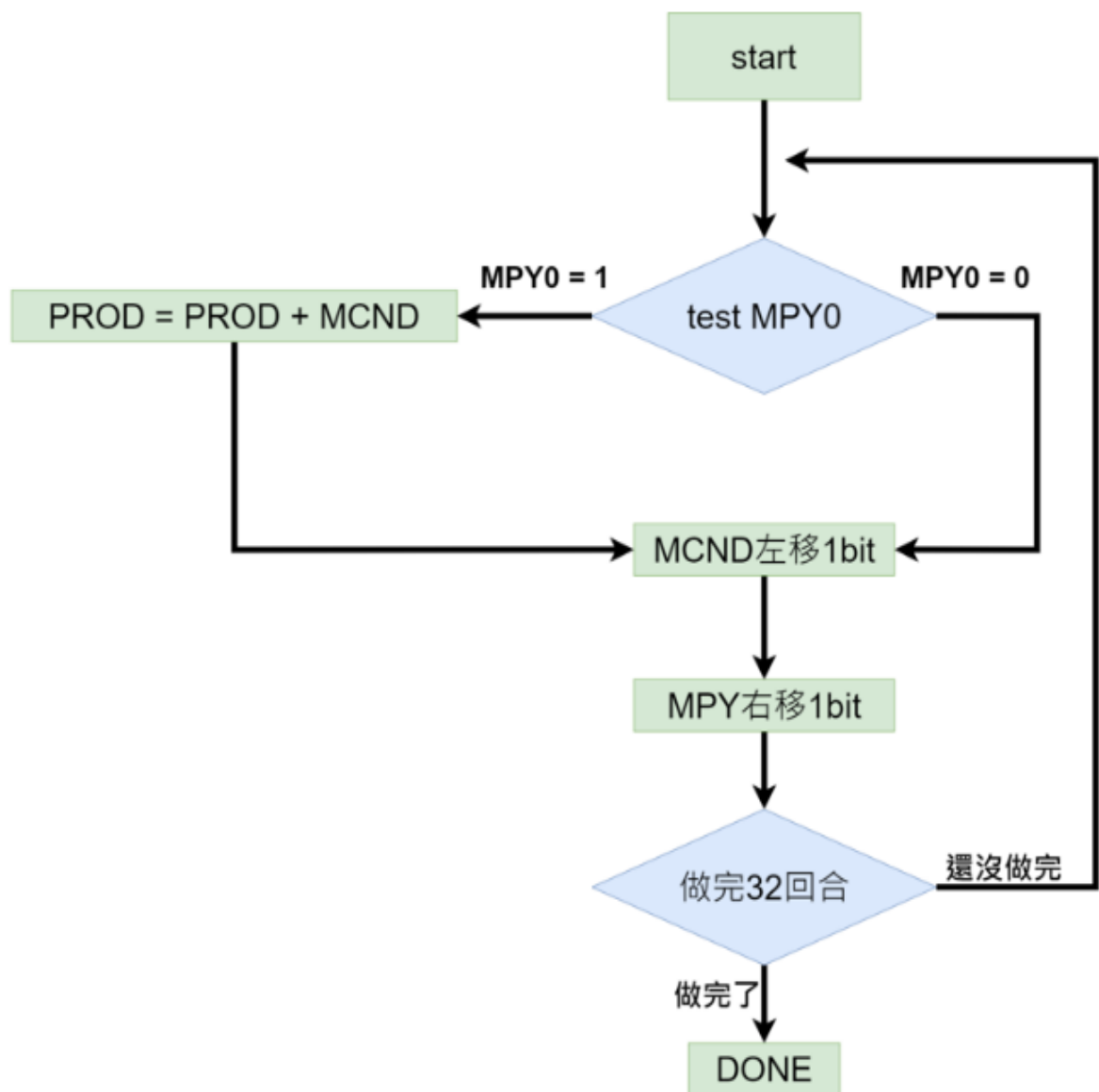
(1)ALU

使用 Gate-Level Modeling 和 Data Flow Modeling 設計包含 32-bits AND, OR, ADD, SUB, SLT 功能，先做出一個 1-bit Full Adder (FA)，並以 Ripple-Carry 的進位方式，連接 32 個 1-bit ALU 就可以組成 32-bit ALU。ALU Control 的 Signal 可以決定要輸出哪種運算結果。第 1 個 ALU 的 Less 是第 31 個 ALU 的 Sum 給它，第 2 到 31 個 ALU 的 Less 輸入都是 0，因為第 31 個 ALU 的 Sum31 是符號位元，如果是 1 代表負也就代表 $A < B$ ，如果是 0 代表正也就代表 $A \geq B$ 。為了解決減法的問題，在 input 中多了一個 Control 來處理減法，利用 $A + (-B)$ 的方式來完成。模組為循序邏輯。



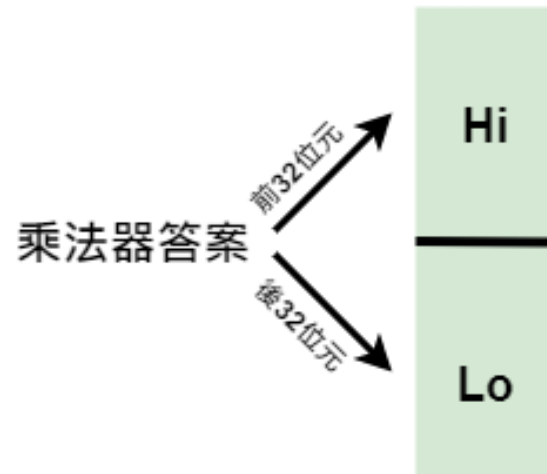
(2)乘法器

32-bits 無號數乘法 Sequential Multiplier，使用課本的第一版乘法器來實作。透過乘數是否為 0 判斷要不要做 $PRODUCT = PRODUCT + \text{被乘數}$ ，判斷完後再把被乘數左移 1bit、乘數右移 1 bit，進行 32 回合後就可以得到答案。模組為循序邏輯。



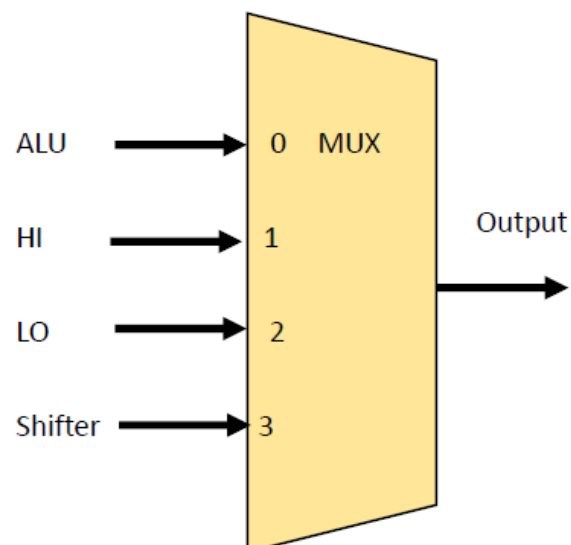
(3) HiLo 暫存器

乘法器計算完後，把乘法器答案儲存在這，答案的前 32bit 存在 Hi 後 32bit 存在 Lo。模組為循序邏輯(Sequential Logic)



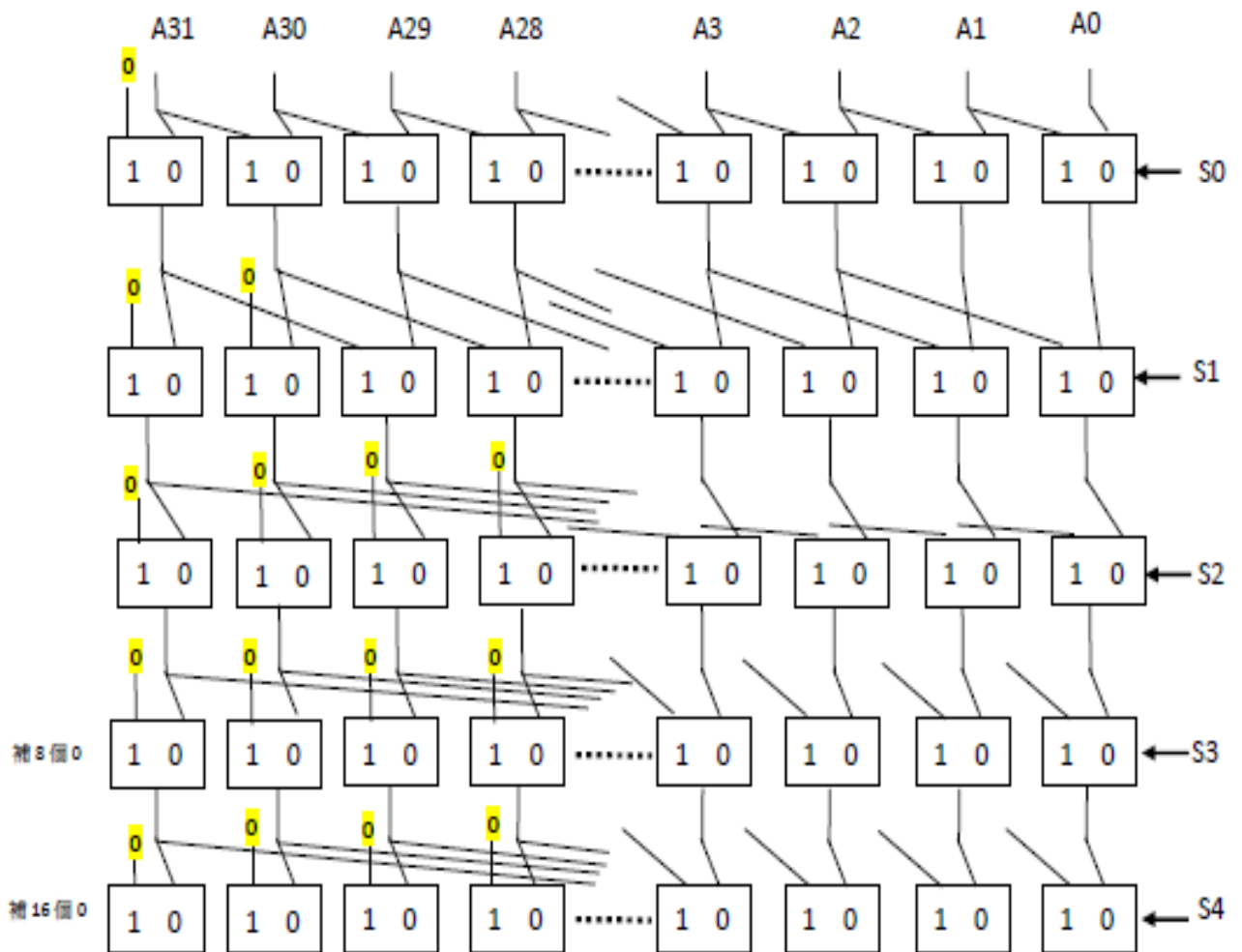
(4) Mux

以 Data Flow Modeling 設計。用(?)條件判斷，做出 if-else 和 case 選擇。
本模組為組合邏輯 (Combinational Logic)。



(5) Shifter

設計 32 bits Barrel Shifter 來完成邏輯右移運算，以課程講義描述方式設計，用 DataFlow Modeling 完成，模組為循序邏輯(Sequential Logic)。總共需要設計出五層，每層 32 個 2 對 1 多工器，共有 160 個 2 對 1 多工器來實現 Shifter 的功能。第一層右移 1 補 1 個 0、第二層右移 2 補 2 個 0、第三層右移 4 補 4 個 0、第四層右移 8 補 8 個 0、第五層右移 16 補 16 個 0。



(6) ALU Control

根據輸入的 6-bit Signal，決定該完成 AND, OR, ADD, SUB, SLT, SRL, MULTU 哪一種運算。控制訊號與功能對應如下：

Signal : 6-bit Value (Decimal)

AND : 36

OR : 37

ADD : 32

SUB : 34

SLT : 42

SRL : 02

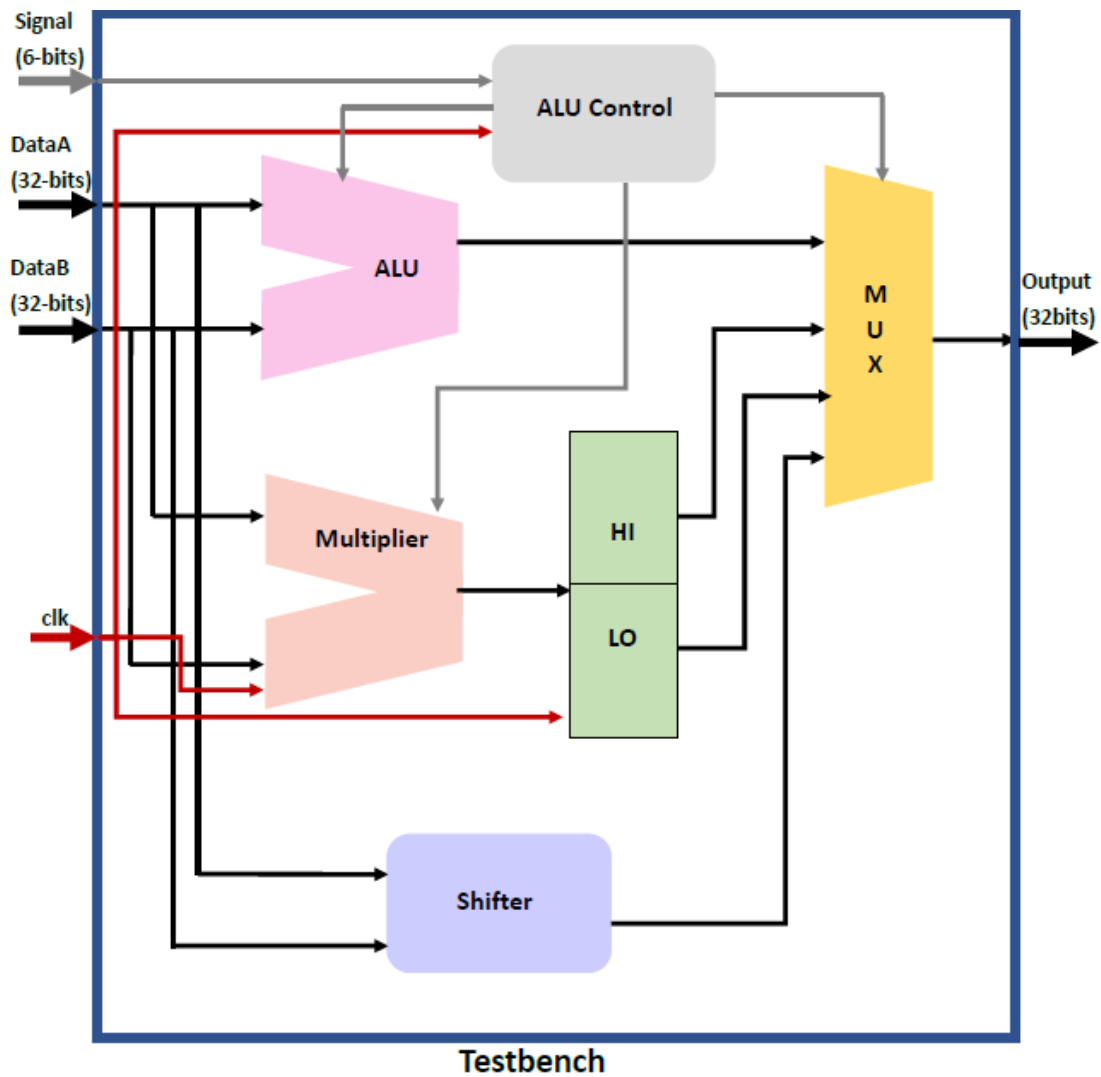
MULTU : 25

(7) Testbench

所有 module 之測試平台，讀入測試資料。以驗證所設計之模組，功能正確性。

(8) TotalALU

用來整合所有 module。

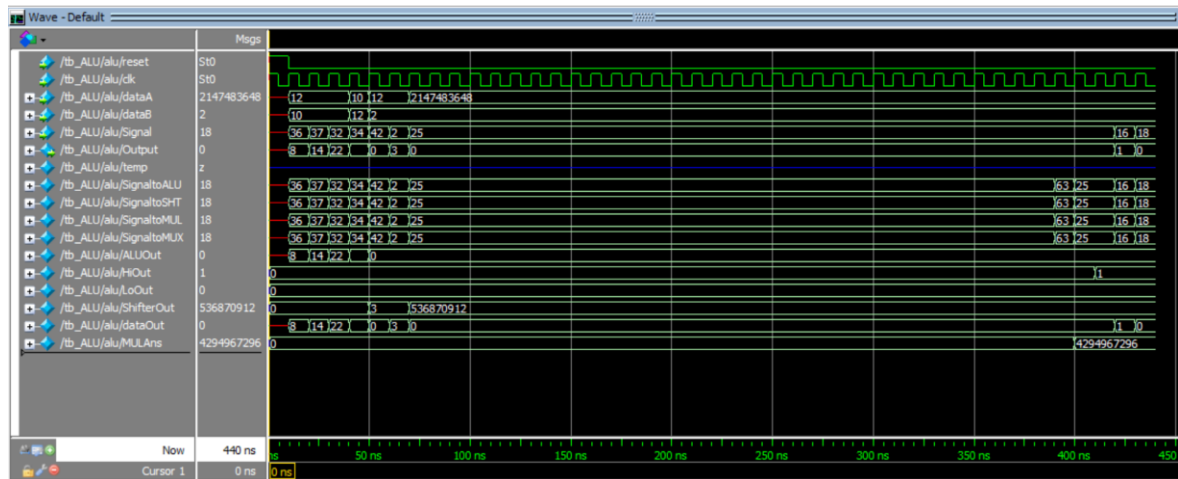


(9)延遲(delay)

除了 Testbench 可包含延遲，以作為 Clock 週期與 Reset 之用外，其餘設計均不可有延遲(clk 同步)。

3. modelsim 執行結果與討論

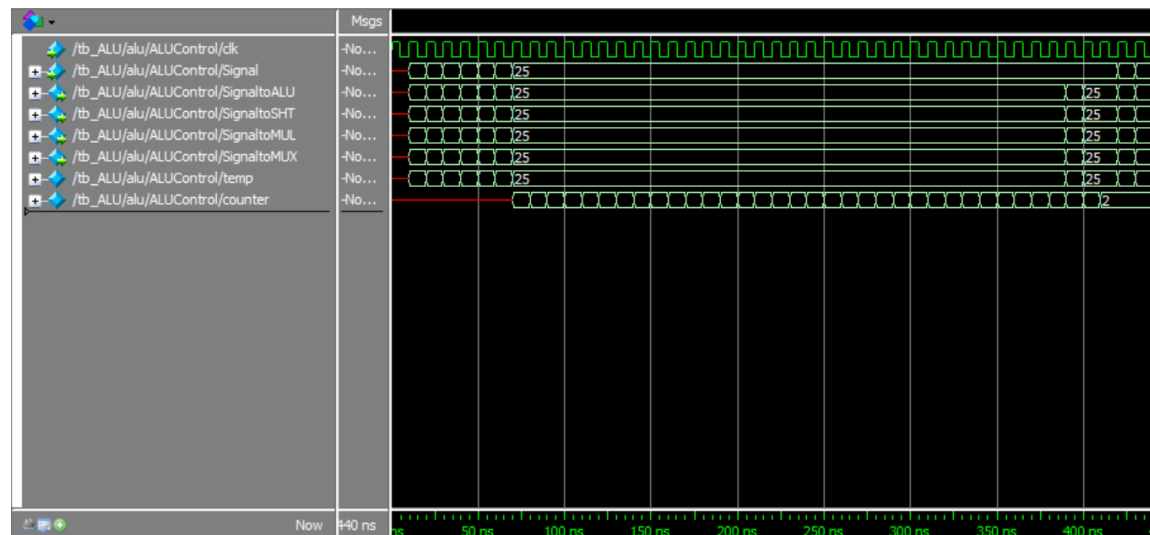
- waveform



- 驗證結果

```
#
#
#      1: Input: AND(36)          12          10
#      2: Correct: Your answer is:      8,
#          Correct answer is:           8
#
#
#      2: Input: OR(37)           12          10
#      3: Correct: Your answer is:     14,
#          Correct answer is:          14
#
#
#      3: Input: ADD(32)           12          10
#      4: Correct: Your answer is:     22,
#          Correct answer is:          22
#
#
#      4: Input: SUB(34)           10          12
#      5: Correct: Your answer is:4294967294,
#          Correct answer is:4294967294
#
#
#      5: Input: SLI(42)           12           2
#      6: Correct: Your answer is:      0,
#          Correct answer is:           0
#
#
#      6: Input: SRL( 2)           12           2
#      7: Correct: Your answer is:      3,
#          Correct answer is:           3
#
#
#      7: Input: MULTU(25) 2147483648          2
#      40: Multu End
#
#
#      Move Hi
#      43: Correct: Your answer is:      1,
#          Correct answer is:           1
#
#
#      Move Lo
#      44: Correct: Your answer is:      0,
#          Correct answer is:           0
#
#
# Simulation End
```


- ALU control



Time	訊號	運算
10ns~20ns	36(AND)	data(12) & dataB(10)
20ns~30ns	37(OR)	data(12) dataB(10)
30ns~40ns	32(ADD)	data(12) + dataB(10)
40ns~50ns	34(SUB)	data(10) - dataB(12)
50ns~60ns	42(SLT)	data(12) < dataB(2)
60ns~70ns	02(SRL)	data(12)向右位移 dataB(2)
70ns~440ns	25(MULTU)	data(2147483648) * dataB(2)

● 結論

這次的 project 整合了第三章的內容，從利用之前學的邏輯閘，組合成 4to1 多工器，再利用 barrel shifter 來實現右移的功能，最後再做乘法器。一步步地把一個一個小功能建立起來再把他們整個整合在一起，做出了一個具有基本運算功能的 ALU，讓我們了解電腦的計算方式。

4. 心得感想

這次 project 雖然和電子實驗一樣都是用 verilog 來寫，最基本的 1bit 加法器電子實驗也寫過，但因為老師有規定不能直接用 operator 所以必須用邏輯閘的概念來做，雖然考試的時候都有考過那些邏輯閘要如何組成，但因為沒有真正實做過，所以一開始腦袋會有點轉不過來。因為老師規定了很多語法不能用，所以一開始真的有點迷茫，可是當寫出一個模組後就更熟悉語法該如何使用，就有種熟能生巧的感覺!但把一個一個模組寫完不代表真的寫完了，還要注意接線有沒有接對，即便編譯過了也不一定可以模擬，modelsim 也不會告訴我們哪裡接錯了必須要自己找，所以在寫得當下必須要很細心。

5. 分工

11027140 陳芃睿 11027147 邱峻彥:撰寫程式碼

11027112 陳逸駿 11027149 游婕歆:書面報告