計算機組織報告

109學年度第2學期

老師：朱守禮老師

學生： 第十組

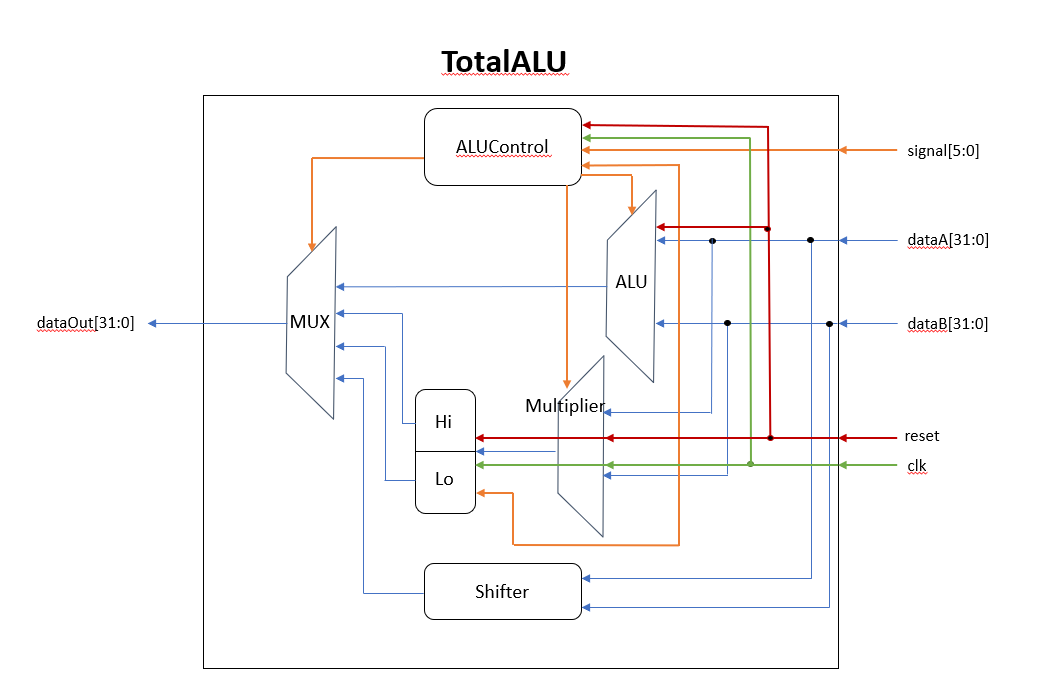
10827117 陳柏宇

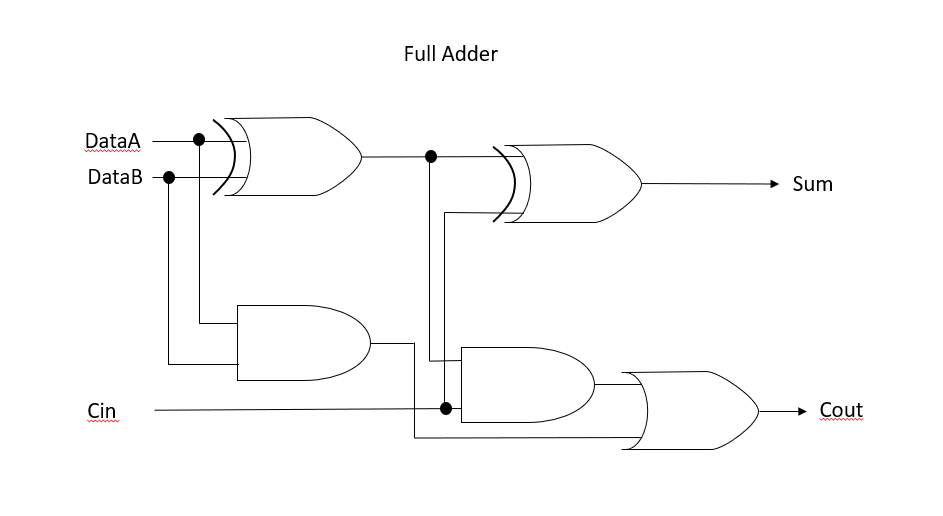
10827129 周暐倫

10827142廖彥蓉

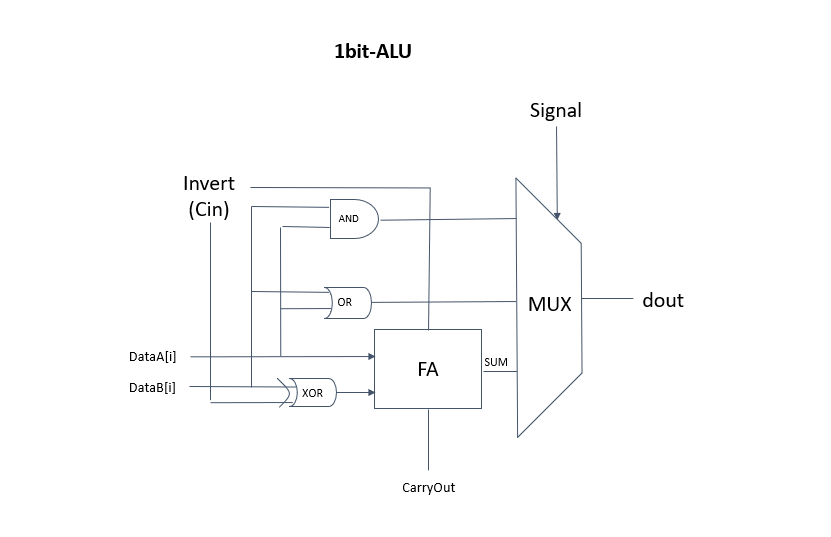
10827157 吳添聖

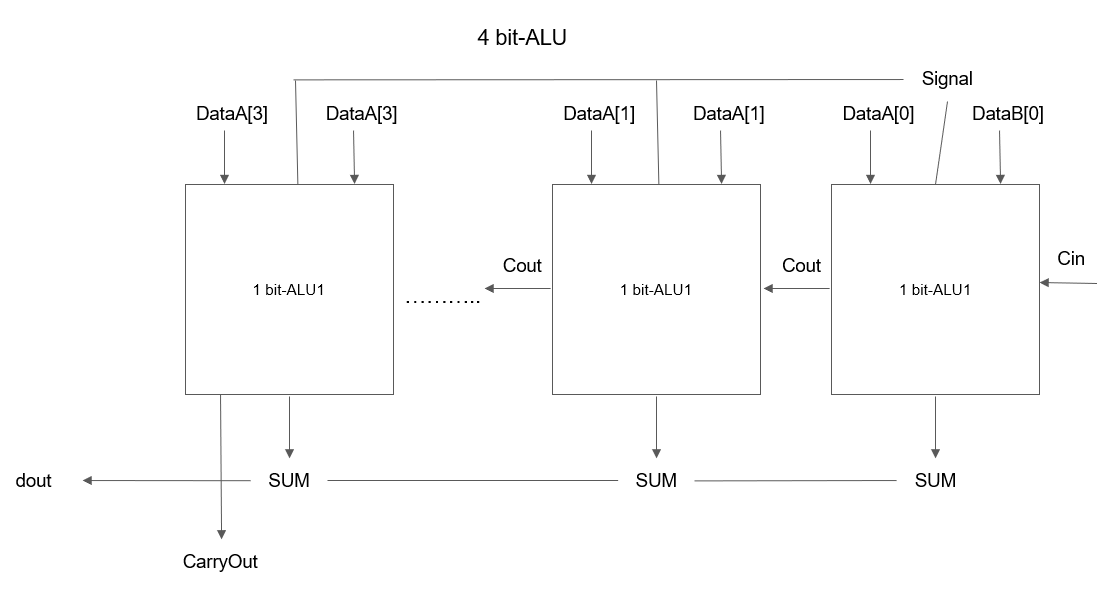
一、Datapath 與詳細架構圖

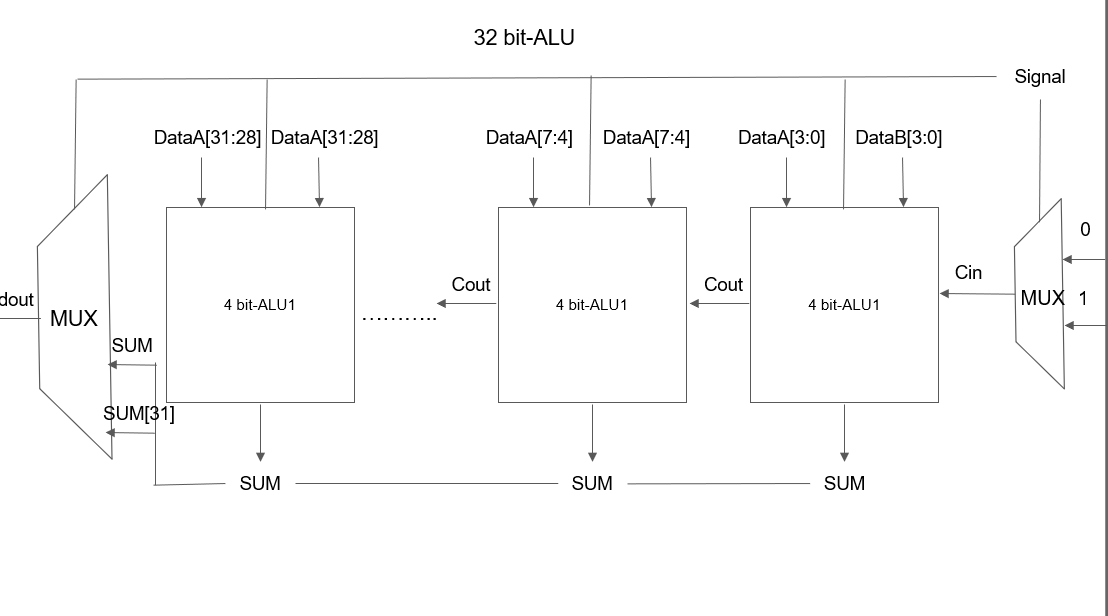


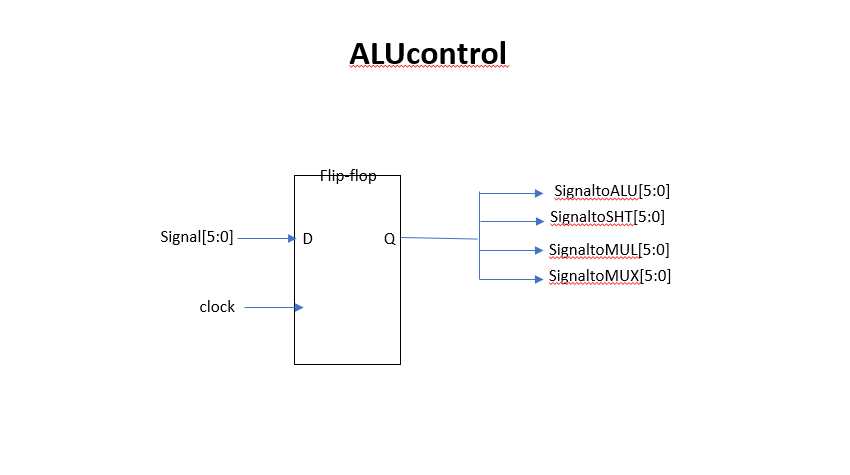
//---------------------------------------------------------------------------------------

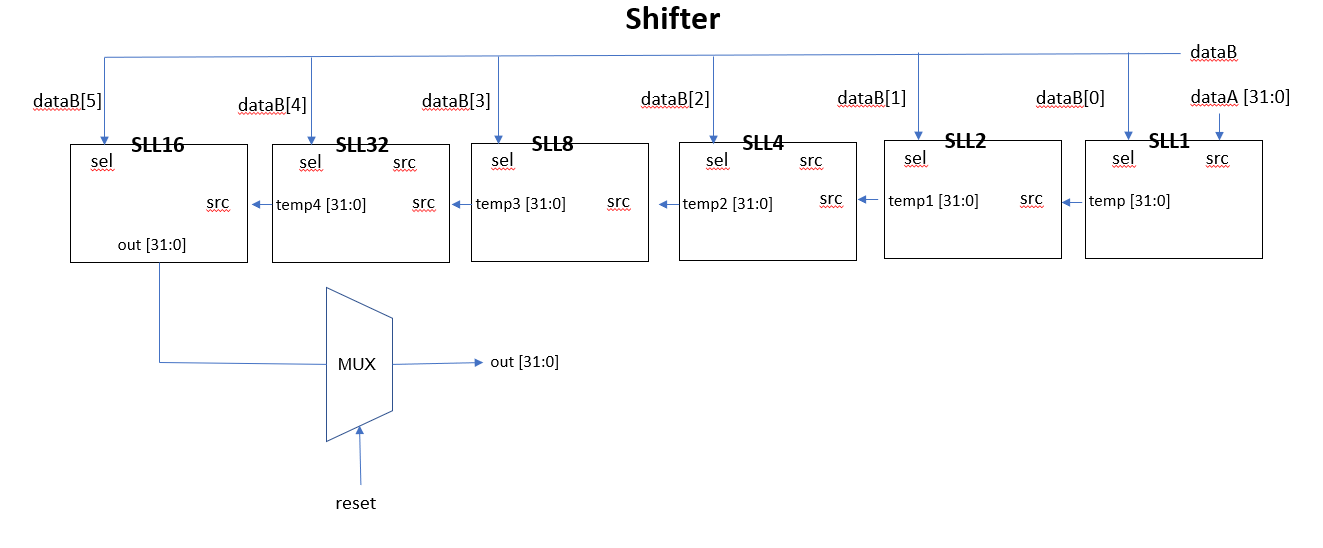
//---------------------------------------------------------------------------------------

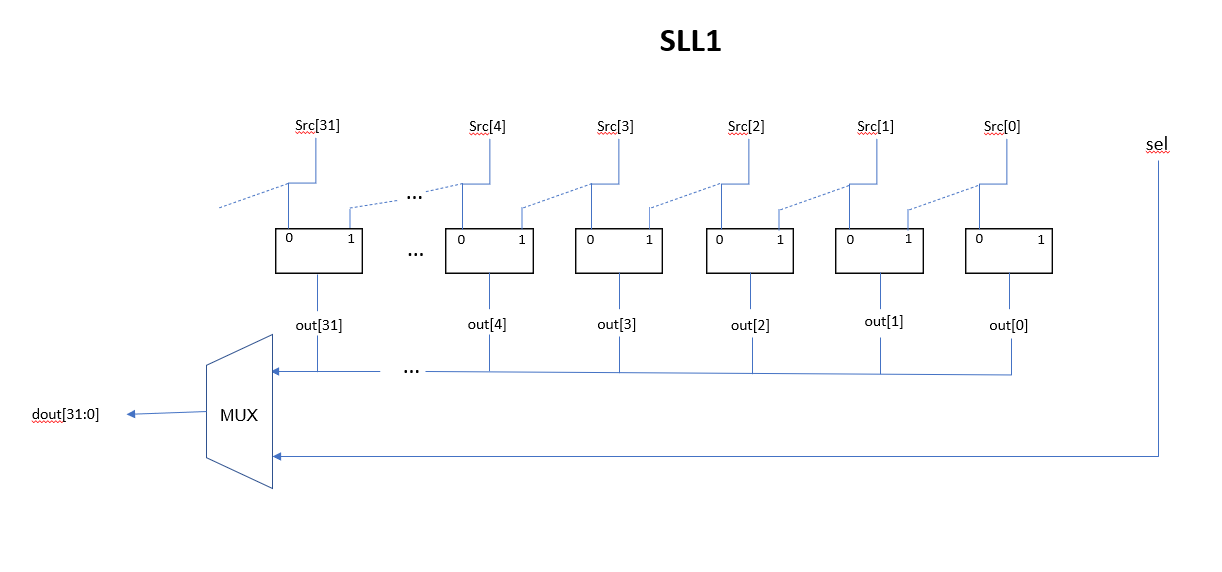
//---------------------------------------------------------------------------------------

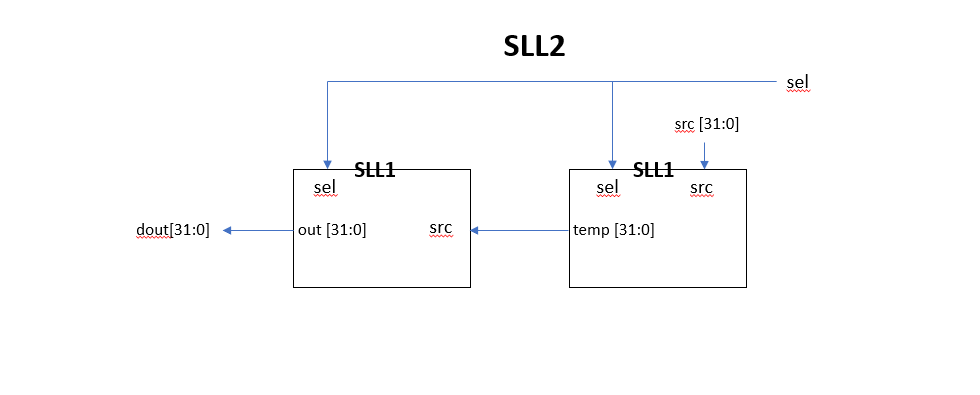
//---------------------------------------------------------------------------------------

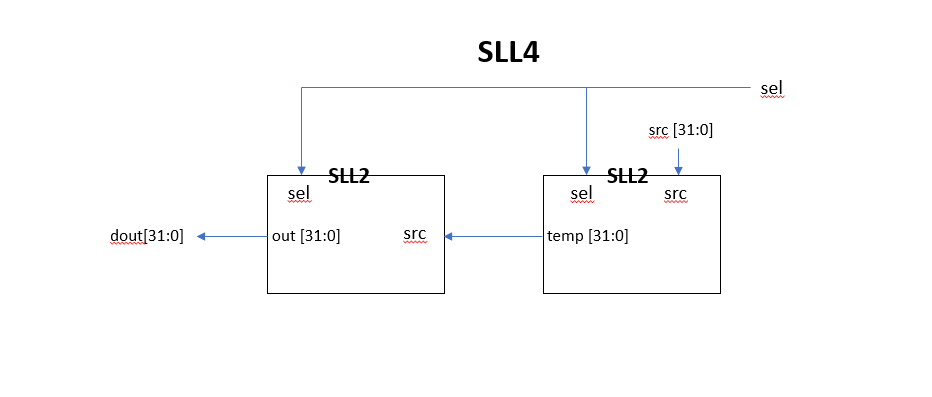
//---------------------------------------------------------------------------------------

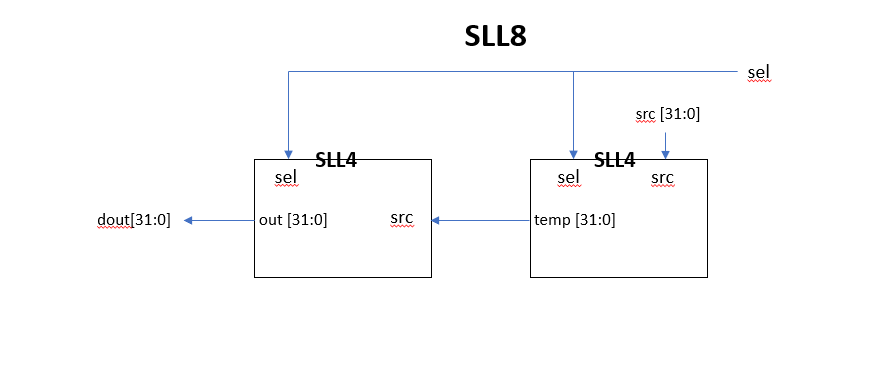
//---------------------------------------------------------------------------------------

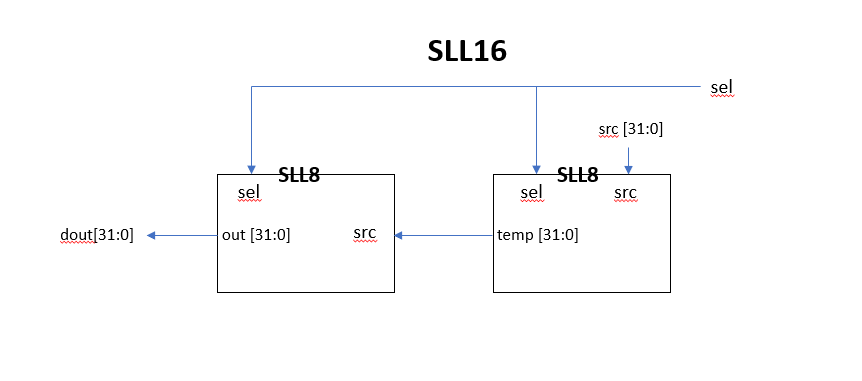
//---------------------------------------------------------------------------------------

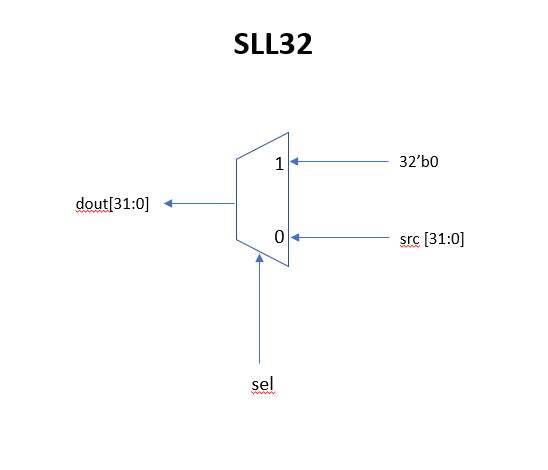
//---------------------------------------------------------------------------------------

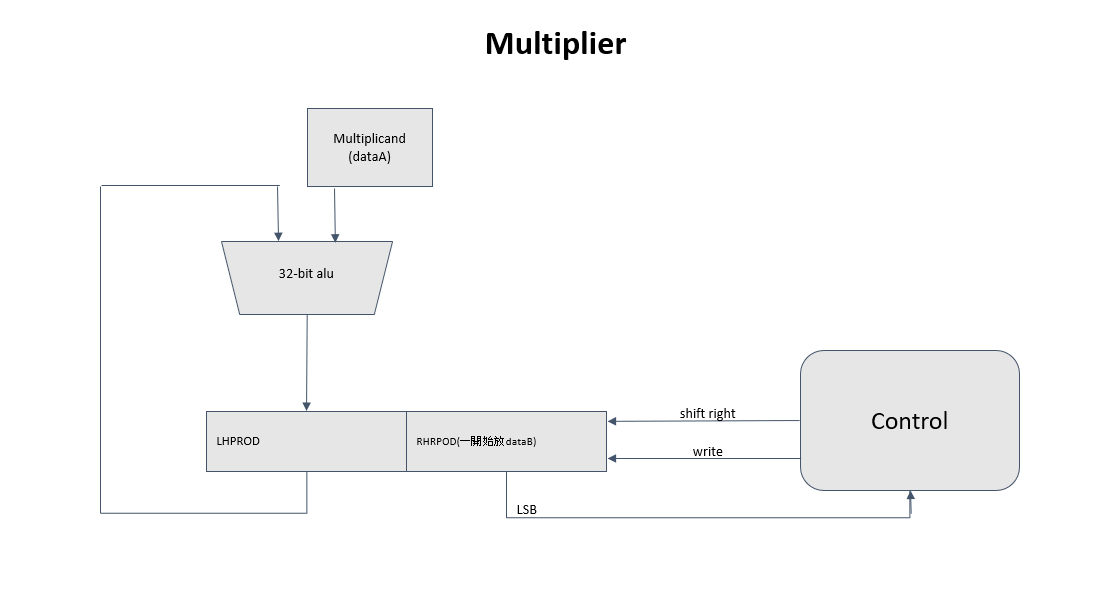
//---------------------------------------------------------------------------------------

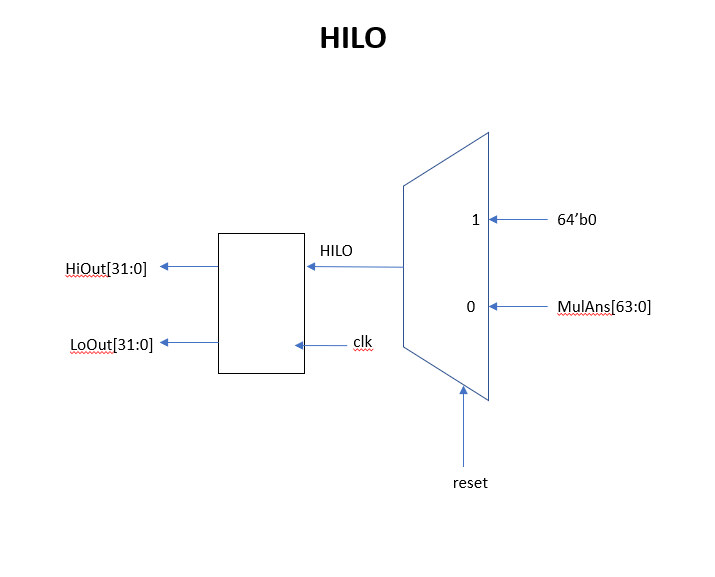
//---------------------------------------------------------------------------------------

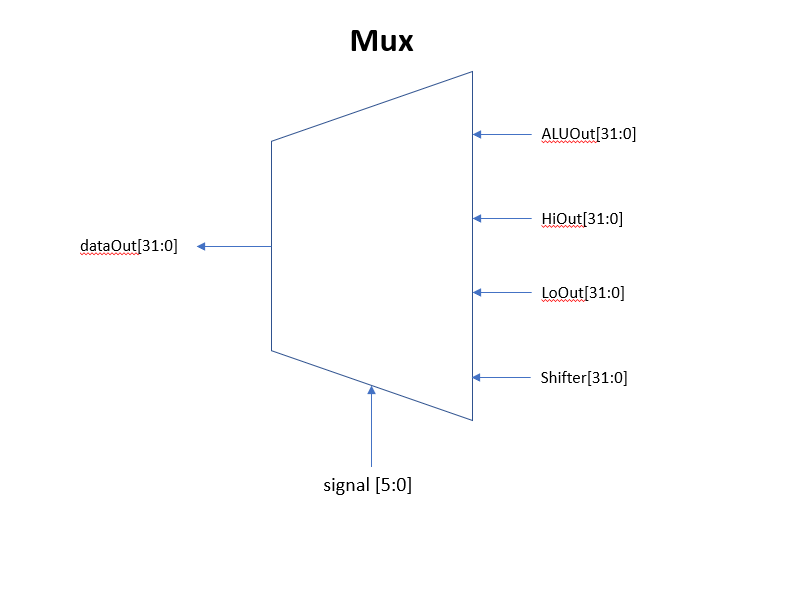
//---------------------------------------------------------------------------------------

//---------------------------------------------------------------------------------------

//---------------------------------------------------------------------------------------

//---------------------------------------------------------------------------------------

//---------------------------------------------------------------------------------------



二、設計重點說明。

ALU：在這一個設計中，ALU的部分我們先做出1 bit-ALU之後，在串接成4 bit-ALU，最後在串接出32 bit-ALU，而在1 bit-ALU中又會呼叫Full Adder，所以在ALU中最大的重點便是在於那1 bit-ALU，在這一個1 bit-ALU中，我們會先決定DataB要不要換成2’s complement，決定好之後，便開始做兩個輸入的AND還有OR和FA，經過這三個過程之後，再依照Signal所給的訊號，來決定要輸出哪一個結果。而在後面的4 bit-ALU，就是呼叫1 bit-ALU四次，32 bit-ALU就是呼叫4 bit-ALU八次，經由這樣的過程，就可以將32個1 bit-ALU串接起來變成一個32 bit-ALU，而在32 bit-ALU中，在呼叫4 bit-ALU之前會先判斷一開始的Cin是要給0或是1，因為在計算時如果DataB要換成2’s complement的話，最一開始的Cin就需要給1，如果不是的話就是給0，而SLT則是判斷DataA和DataB經過Sub之後的結果，最高位元是0還是1，如果是1就是小於，反之則為大於，如此一來就完成了ALU。

Multiplier：在這一個設計中，Multiplier我們決定使用always來運作，在always中，五們分成了三個Cases，分別是第一次進去的時候，因為是第三版的乘法器，所以我們會先把乘數放到temp裡面，而temp是一個64 bit的空間，放進去之後，後面再做運作的時候，就會跑到MULTU的Case這邊，如果當下的temp[0]是1的話，就表示要進行乘法，但因為是乘以1所以直接加上被乘數即可，但如果不是1則不做事，因為是乘以0，所以等於是沒有做事，而加上去後要再向右移，如此一來到了最後乘法器結束，商數就會在temp的[31:0]這半邊，原本一開始放進來的DataB也就是乘數就會因為右移而不見，這樣Multiplier便完成了。

Shifter：在Shifter中我們採取的設計跟ALU很類似，我們要符合各種不同的位移數，所以我們分別設計了幾種SLL，SLL1、SLL2、SLL4、SLL8、SLL16和SLL32，其中最主要的就是SLL1，在裡面，會直接將輸入的DataA放到另一個變數裡，而在放置的過程中，同時將每一個bit往前移一個bit，這樣就可以位移1 bit了，位移結束後，再依照sel的輸入來判斷要輸出原本的DataA還是輸出位移過的變數，剩下的SLL2、SLL4、SLL8、SLL16和SLL32都是彼此呼叫而達到該有的bit數，有了這些module，在shifter就只要將輸入進來的DataB的前面六個位元和DataA分別丟給以上的SLL便可以輸出答案。

HiLo：在這一個設計中，我們使用always block去做判斷，如果reset是0就將暫存的變數設定為0如果不是就將乘法器的答案放進暫存變數的[31:0]位元中，便可以輸出結果。

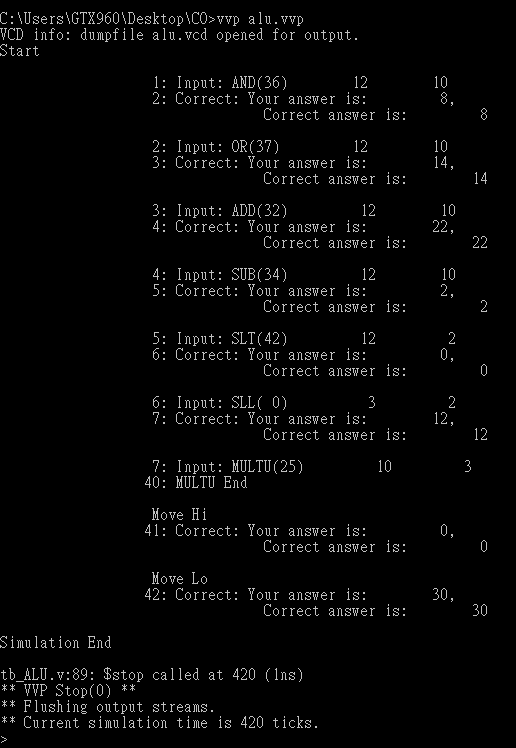
MUX：在這一個設計中，我們會先設定好各種變數的值，然後判斷Signal訊號為何，如果是MFHI就輸出Hiout，如果是MFLO就輸出LoOut，如果是SLL就輸出Shifter，最後如果都不是以上條件就輸出ALUOut。

ALUControl：在control中，便是整理所有要輸出給ALU、Shifter、Multiplier的訊號，其中最需要著墨的就是在Multiplier這邊，因為需要進行多次的運算，所以加上了一個always來進行乘法的回圈。

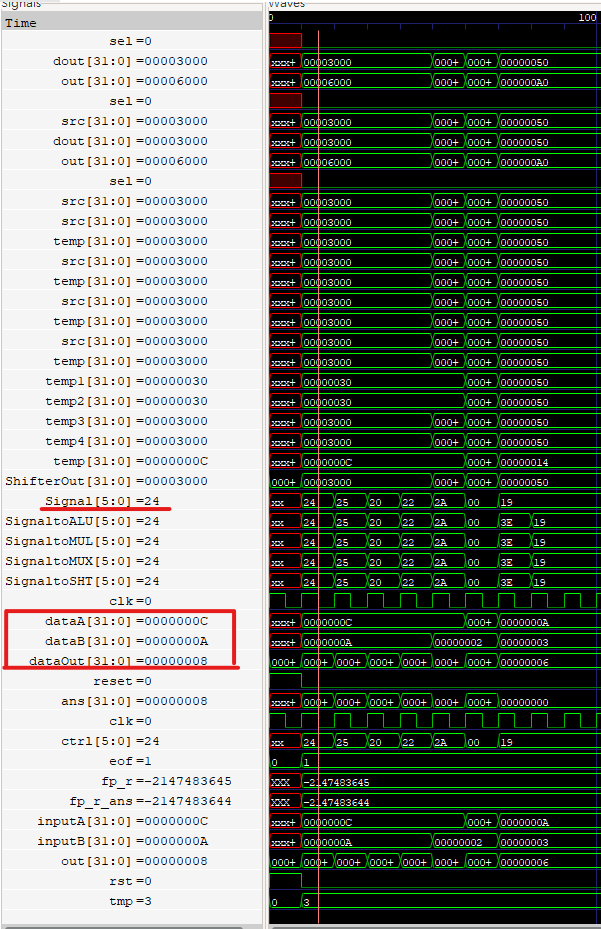
TotalALU： 首先先定義好各指令相對應的六位元二進位訊號，接著定義出用來儲存各個部位答案的線路，最後模擬各個module並在最後經過MUX後，輸出input要求的結果。

TB：我們先設定好時脈的週期，打開一個"alu.vcd"資料庫來記錄訊號，讀取input和ans檔，接著用while迴圈讀入input檔中的數據，並且將輸出的結果和ans檔比對。其中大部分的內容與老師給的相同，主要只是更改了比對後輸出的名稱以及部分input, ans檔案的內容而已。

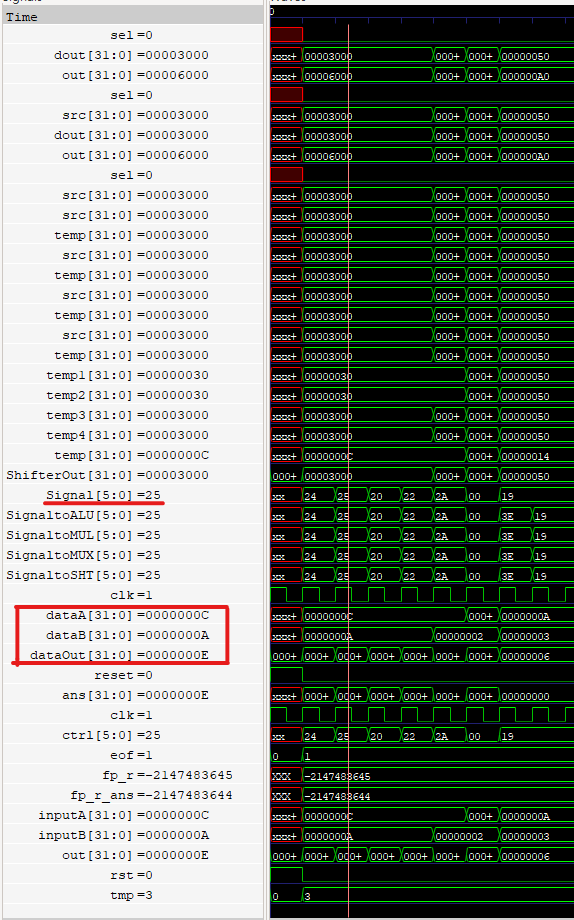
三、Icarus Verilog 驗證結果與 Waveform 輸出圖形，並加以說明。



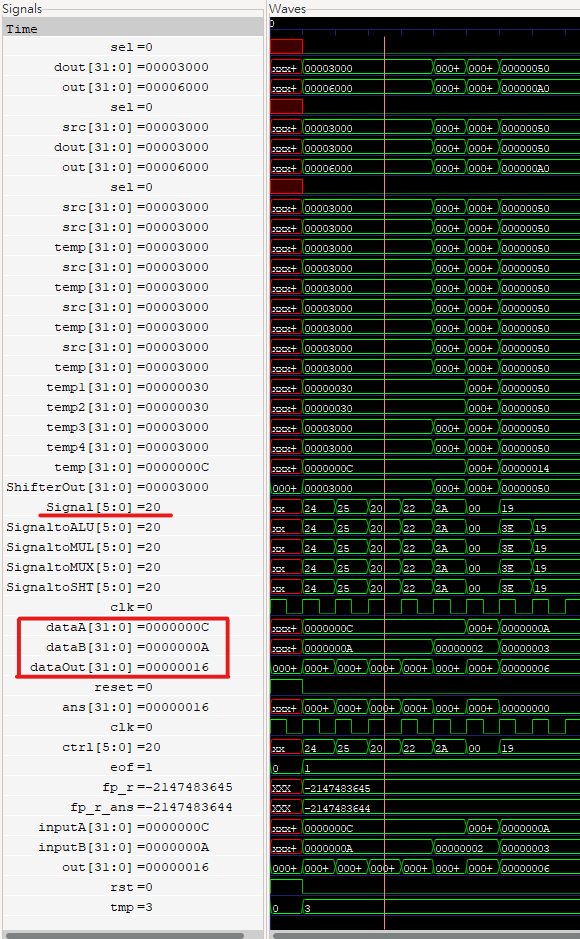
以上是使用Testbanch所測試出來的各項module的結果，答案皆對。

ＡＮＤ

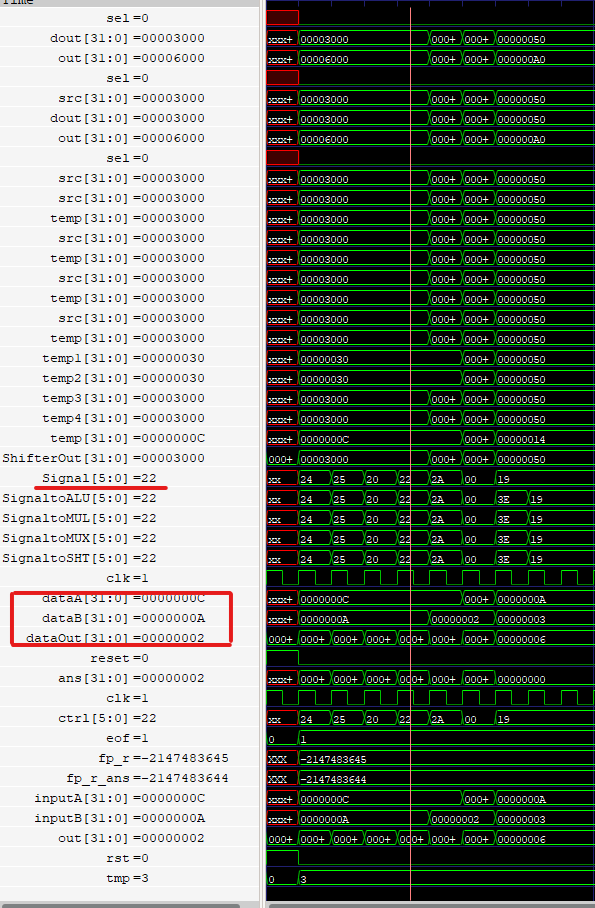
當訊號為十進位的36也就AND的時候，就將dataA和dataB的結果放入於dataOut。（12&10 = 8 二進位的運算）

ＯＲ

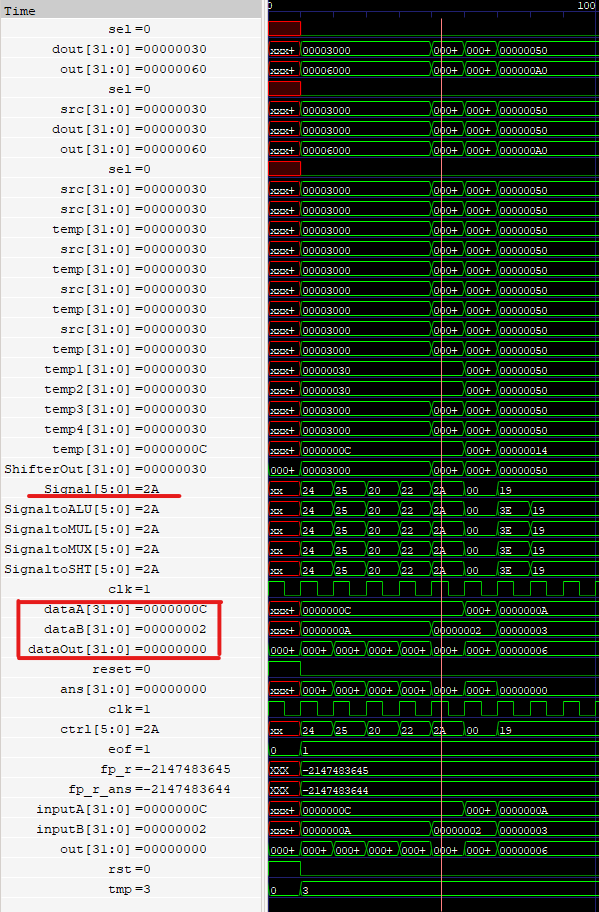
在訊號輸入為十進位37的時候代表要做OR，然後將dataA和dataB的結果放入dataOut。（12|10 = 14 二進位的運算）

ＡＤＤ

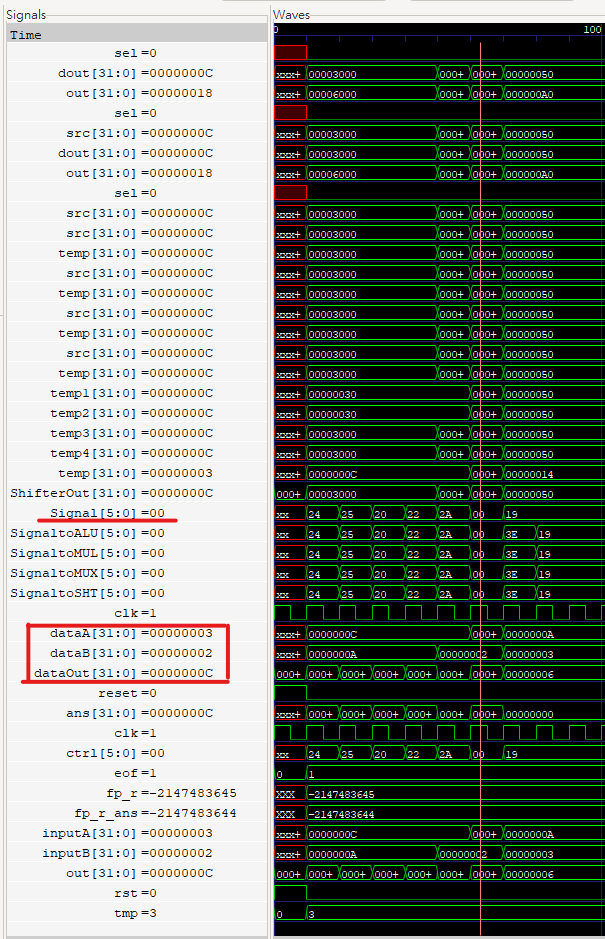
在訊號輸入為十進位32的時候代表要做ADD，然後將dataA和dataB的結果放入dataOut。（12+10=22 二進位的運算）

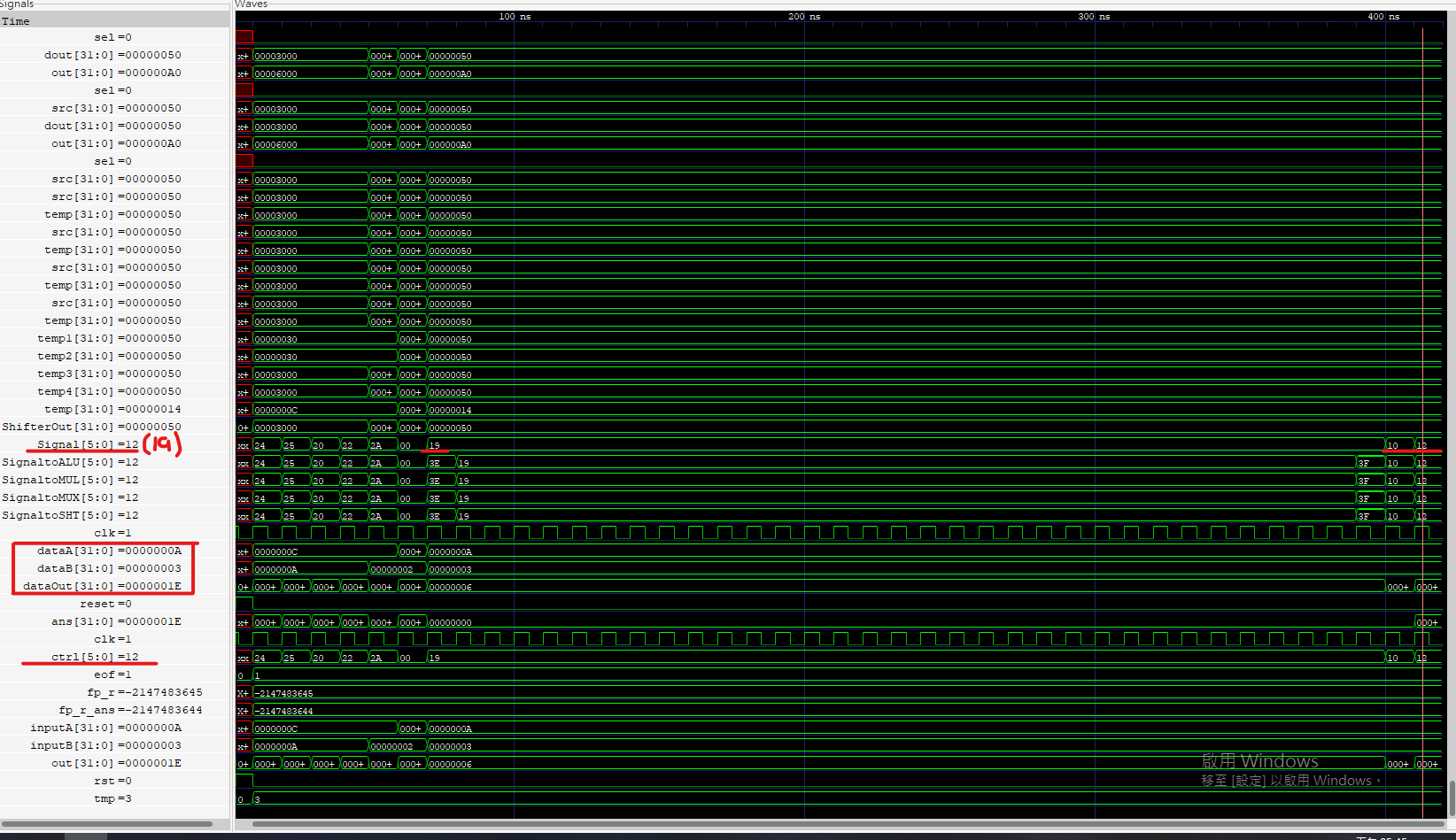
ＳＵＢ

在訊號輸入為十進位34的時候代表要做SUB，然後將dataA和dataB的結果放入dataOut。（12-10=2 二進位的運算）

ＳＬＴ

在訊號輸入為十進位42的時候代表要做SLT，然後將dataA和dataB的結果放入dataOut。（12-2>0 二進位的運算）

ＳＬＬ

在訊號輸入為十進位0的時候代表要做SLL，然後將dataA和dataB的結果放入dataOut。（3左移2位元=3\*4=12）

ＭＵＬＴＩＰＬＩＥＲ

在訊號輸入為十進位19的時候代表要做乘法，然後將dataA和dataB的結果放入dataOut。（10\*3 = 30）

四、心得感想。

在這一次的Project中，得到了許多很實用的經驗，對於日常很習慣在使用的加減法、乘法、還有邏輯位移功能，沒想到實作起來的話是這樣的不容易，再加上作業中的一些規定和限制，寫起來真的是要配上紙筆先將架構畫出來後，才能懵懵懂懂的寫出點所以然。

也許是因為對於硬體描述語言的操作還不夠熟悉，畢竟接觸的時間還不到一個學期。而且硬體描述語言中一個與我們平時再寫的其他語言有一個很特別的不同之處－其指令是水平同步進行的，因為這個不同之處，在一開始設計作業時的構思和理解上就讓我們跌了一大跤。

除此之外，還有關於接線，input和output、reg和wire的差異和不同之處，輸出的時候要用哪一個宣告的變數，輸入的時候要用哪一個宣告的變數，這些都是很重要的，如果用錯的變數或是宣告錯誤的話，執行的結果就會連帶產生很多錯誤，一個module錯了可能就會連帶影響到下一個。

ALU就是最明顯的例子，要從1 bit開始串接到變成32 bit-ALU在這一次的作業中，我們是用1 bit串成4 bit在串接成32 bit-ALU，如果在串接的過程中，只要有一個變數或是輸出錯誤，給的資料錯了，bit範圍給錯，或是輸出到錯誤的變數上面，還有不小心忽略了同步進行的觀念，使用了重複的wire 變數，就會導致程式根本跑不出東西。

此外，因為作業中要求到好幾個元件必須遵守dataflow Modeling設計，因為在設計時就必須去思考如何用更簡單、直接的方式將要求達成，我們的shifter就是基於ALU的經驗製作出來的。

除上述之外，其實每個元件都帶給了我們挑戰和收穫，從錯誤之中學習到許多硬體描述語言的文法、技巧和需要注意的地方。同時，學習硬體描述語言也讓我們切身感受到了硬體和軟體間的聯繫和模擬意義，身為IT人，時刻保持學習的熱情和發現生活中各項事物的設計，也是我們必修的課題之一。

五、各組員分工方式與負責項目。

陳柏宇：撰寫ALU、Shifter、撰寫心得、撰寫重點說明、畫圖。

吳添聖：撰寫Shifter、ALUcontrol、畫圖。

周暐倫：撰寫Multiplier和畫圖、協助Shifter、WaveForm圖。

廖彥蓉：修正報告格式。