一、 背景

大二下學期接觸到了”計算機組織”這門課，課堂中了解了硬體和晶片的運作及各元件的運作原理，期中考後進而開始接觸這支 Project。

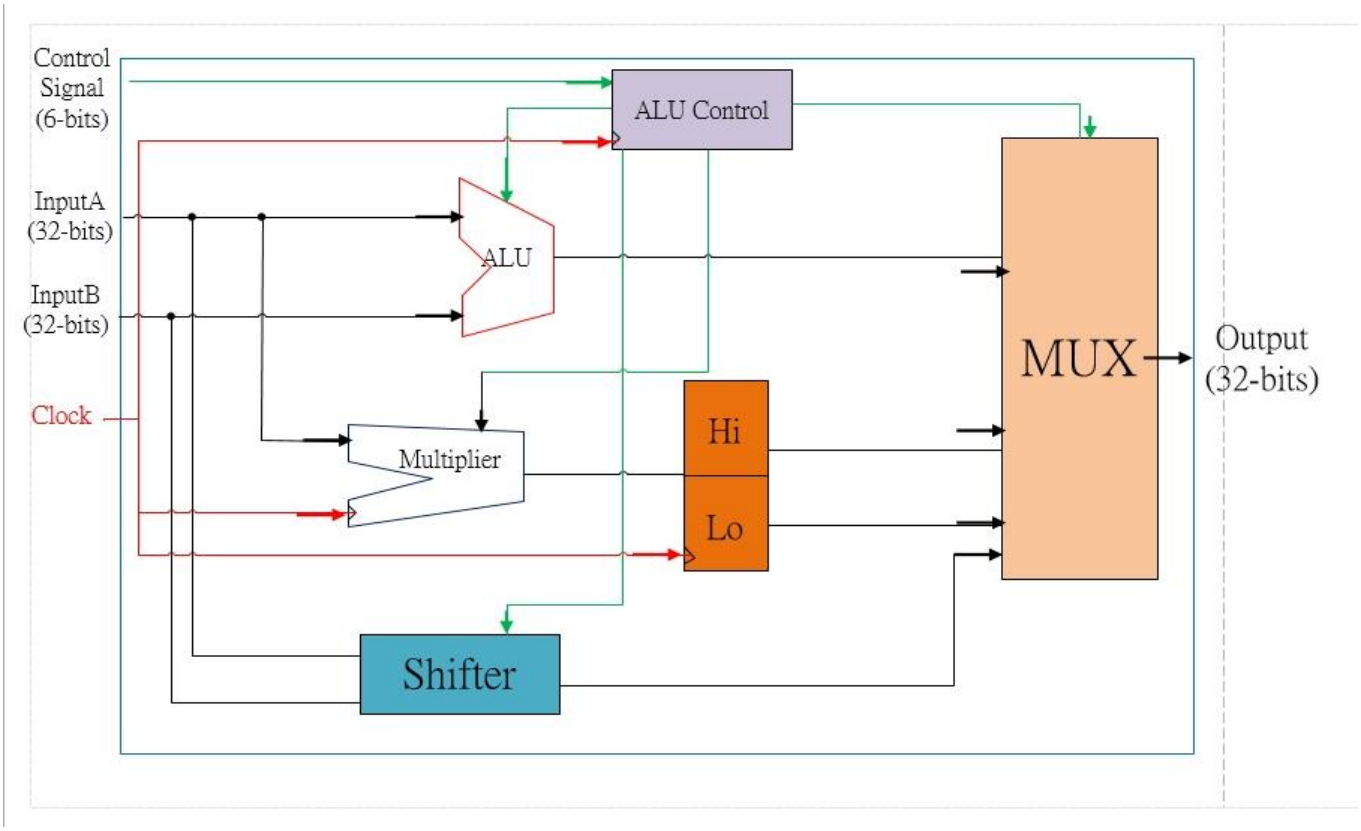
參考資料:

講義和 verilog 書本

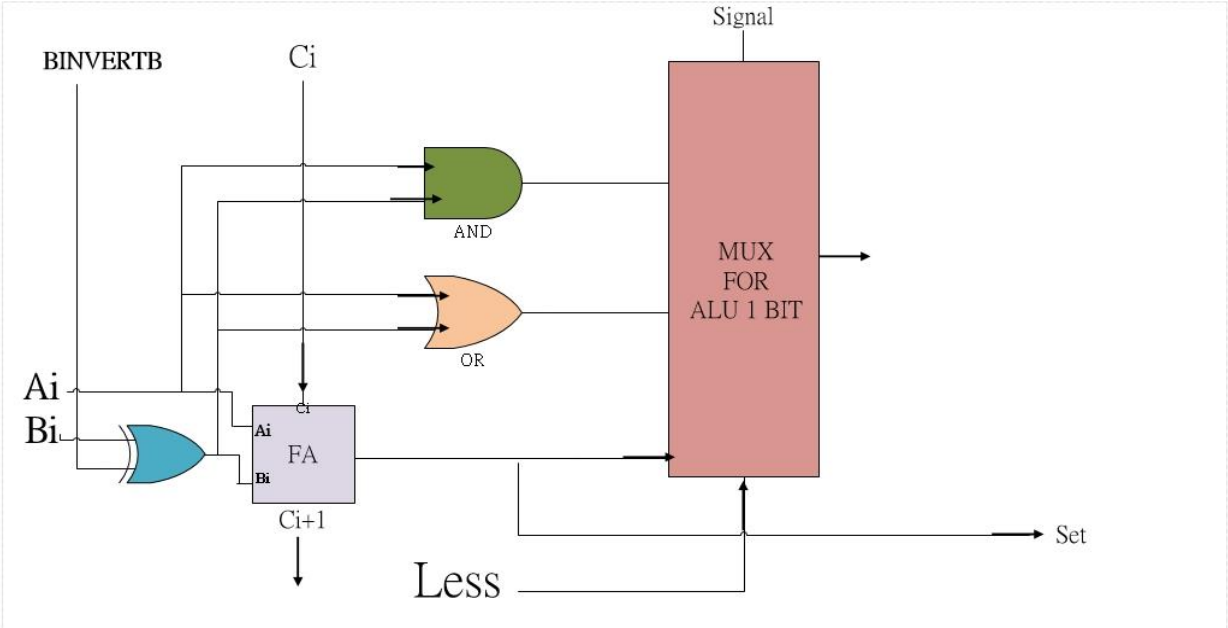
以這些為基礎來進行 ALU、乘法器、Shifters、HiLo Reg.、Mux、ALU Control 之開發。

二、 方法

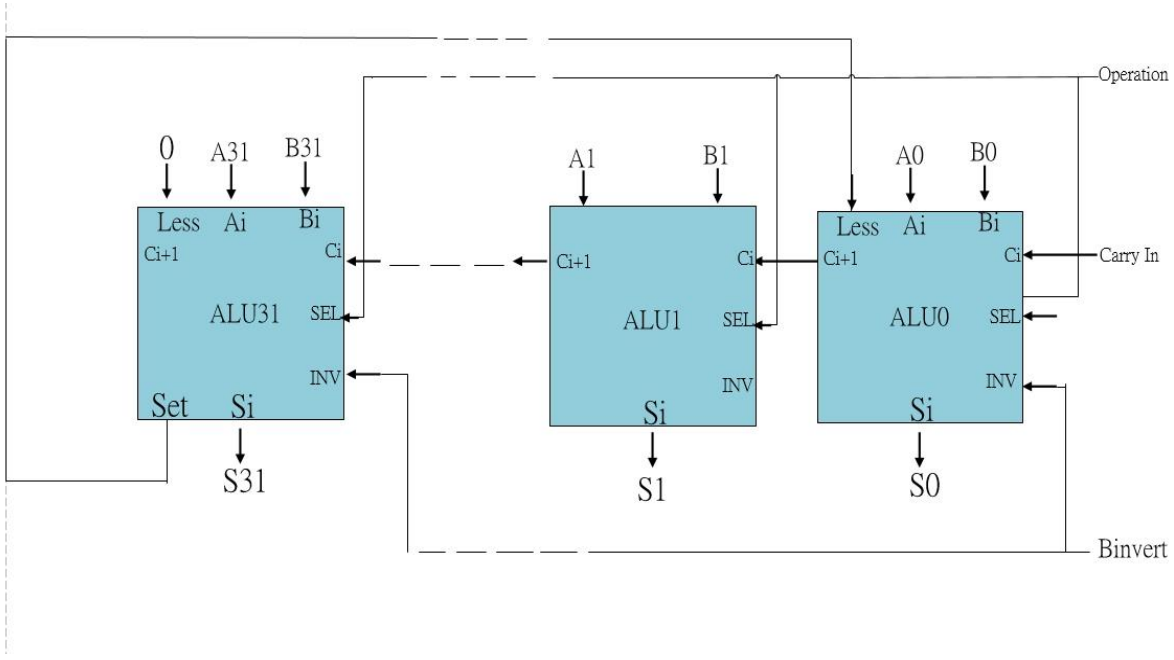
ALU total

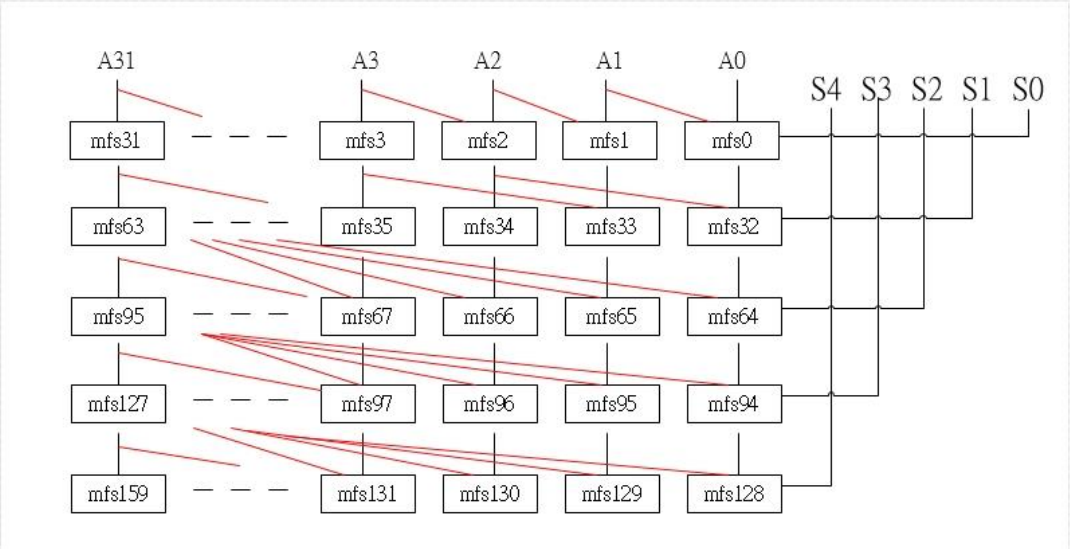


1bit ALU

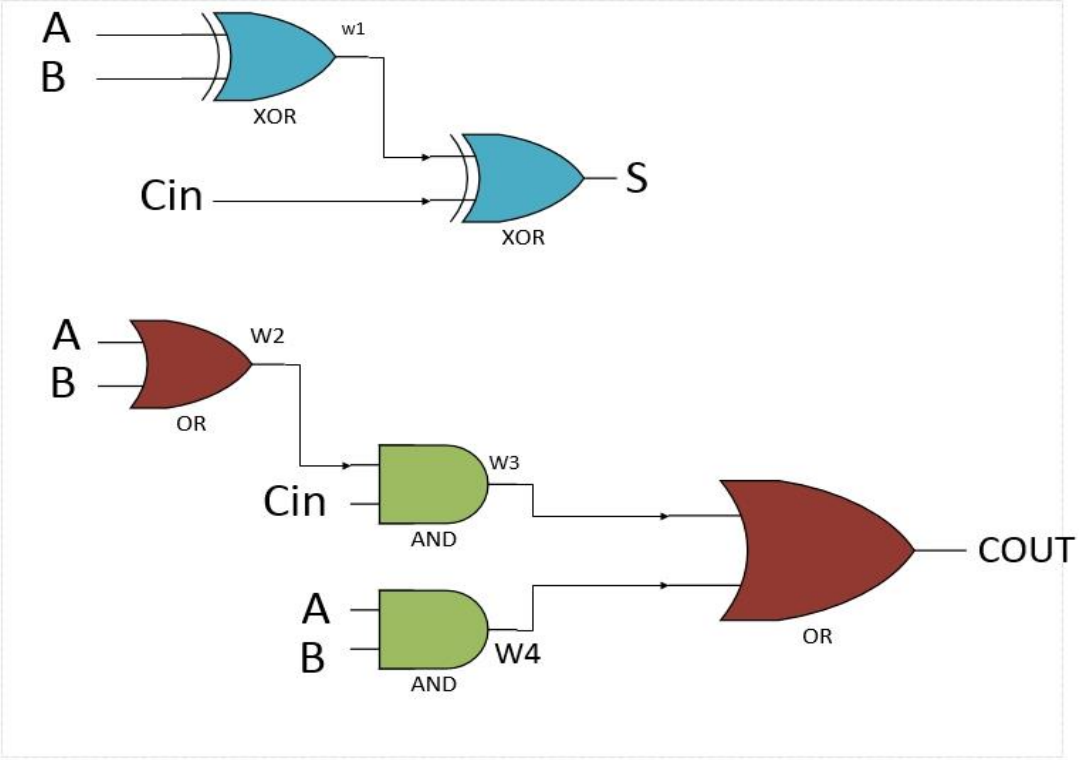


32bit ALU

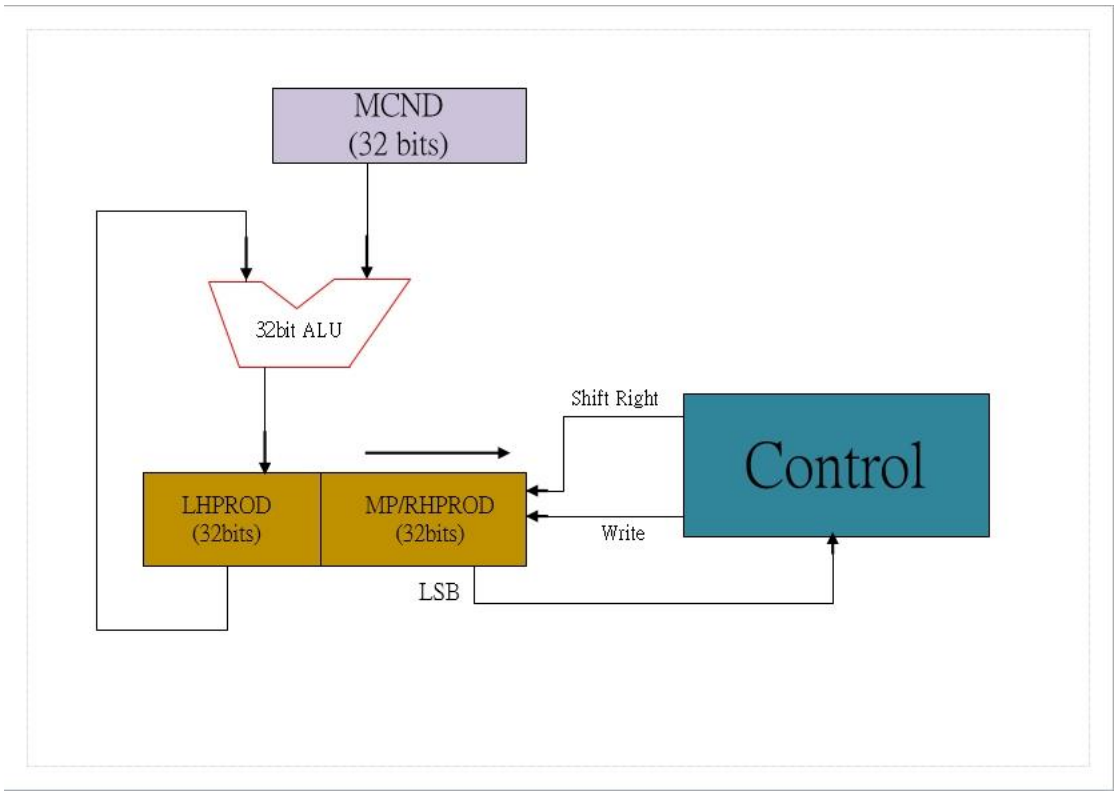



A31
A3
A2
A1
A0
S4
S3
S2
S1
S0
mfs31
mfs3
mfs2
mfs1
mfs0
mfs63
mfs35
mfs34
mfs33
mfs32
mfs95
mfs67
mfs66
mfs65
mfs64
mfs127
mfs97
mfs96
mfs95
mfs94
mfs159
mfs131
mfs130
mfs129
mfs128

FA


A
B
w1
XOR
Cin
S
XOR
A
B
W2
OR
Cin
W3
AND
A
B
W4
AND
OR
COUT

Multiplier


MCND
(32 bits)
32bit ALU
Shift Right
Control
LHPROD
(32bits)
MP/RHPROD
(32bits)
Write
LSB

1. ALU :
   此 ALU 有以下功能:32-bits AND, OR, ADD, SUB, SLT，以 Full-Adder 做起並以 Ripple-Carry 的進位方式連接 32 個 1-Bit ALU Bit Slice，使之變為 32-bits ALU。
2. Multiplier:
   使用第三版乘法器來設計，先將乘數放入 PROD 右半部，然後檢查 PROD 最右邊位數是否為 0，如果為 0 則直接將 PROD 向右位移 1bit，反之則被乘數加 PROD 左半部並放入左半部,然後 PROD 再右位移 1bit，之後檢查是否重負 32 次，最後再進行一次將值傳送給 HiLo Reg.。
3. Shifters :
   以 Data Flow Modeling、組合邏輯的方式設計之 32 bits 的 Barrel Shifter。以 2 對 1 多工器(Mux for Shifter)來組合 2^5 個 Gate 進行移位。
4. HiLo Reg. :
   在乘法器計算完畢之後，將結果儲存到兩個 32 bits 暫存器。
5. Mux :
   以 Data Flow Modeling、組合邏輯的方式設計，整合四組訊號後則一輸出。

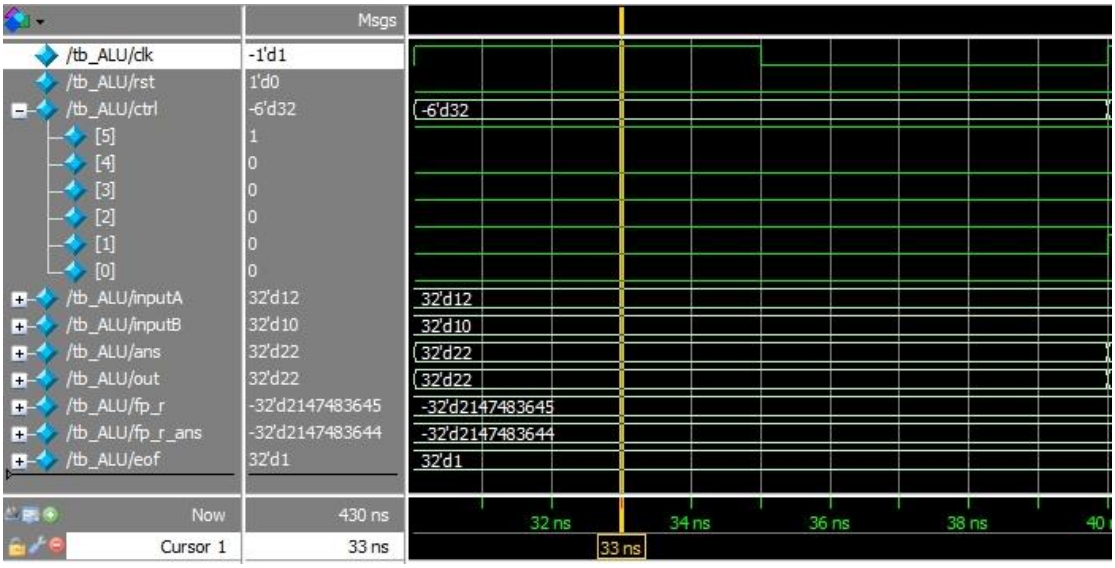
6. ALU Control :
   根據輸入的 6 bits 訊號，決定要執行哪個運算，並將 6 bits 的訊號解析成兩組 2 bits 訊號(SignalToALU_Operation<ALU 功能限定>、SignalToMUX_Operation) 及一組 1 bit 訊號 SignalToALU_Binvert <ALU 功能限定> 。
7. Testbench :
   以讀檔將測試資料讀入，驗證各項 module 之功能正確性。
8. TotalALU :
   將各項 module 建立並執行。

三、結果

1. 加法

輸入:12, 10

輸出:22

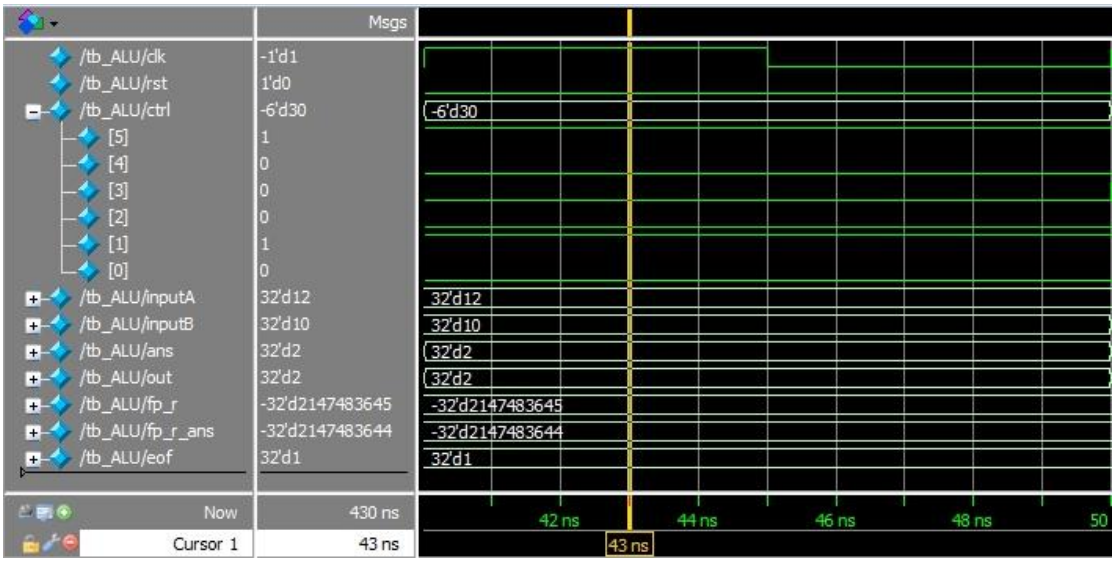


驗證:

當 ctrl 訊號為 32(ADD), 執行加法。

輸入 A 為 12、 B 為 10 而結果 ans 輸出值為 22，故執行正確。

2. 減法



輸入:12,10

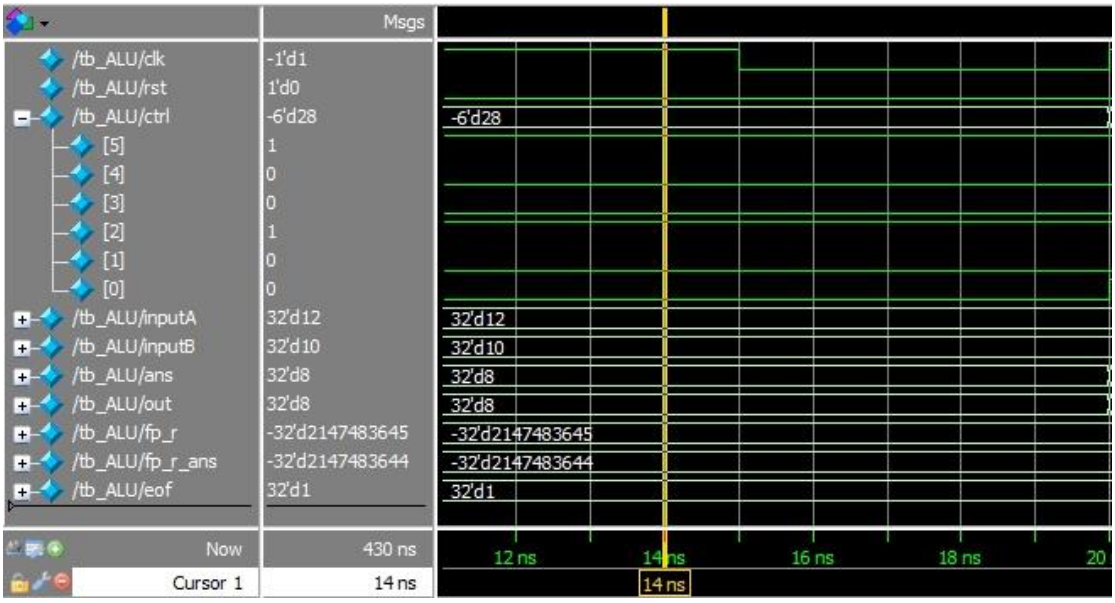
輸出:2

驗證:

當輸入訊號為 34(減法),ALU 執行減法動作。

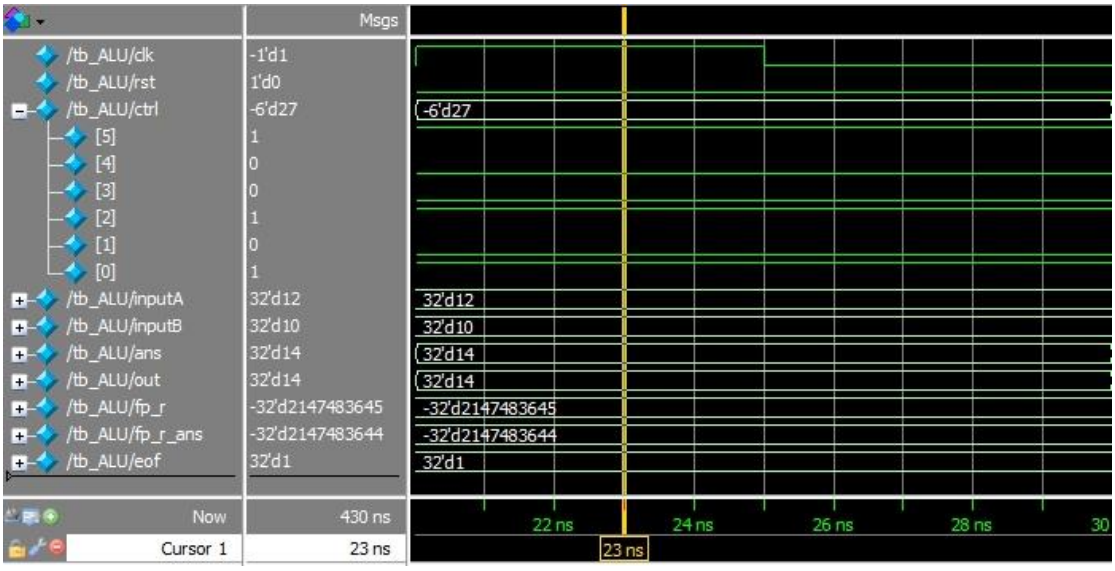
理論結果應該是 12-10 = 2, 而實際檢查 ans 值確實是 2，
故結果正確。

3. AND



輸入:12, 10
輸出:8
驗證:觀察 ctrl 訊號為 32，表示 AND 操作。
12 的 2 進位表示為 1100，10 的 2 進位表示為 1010。
將每一個 bit 各取 AND 結果為 1000，答案為 8。
觀察 and 訊號結果為 8，故執行正確。

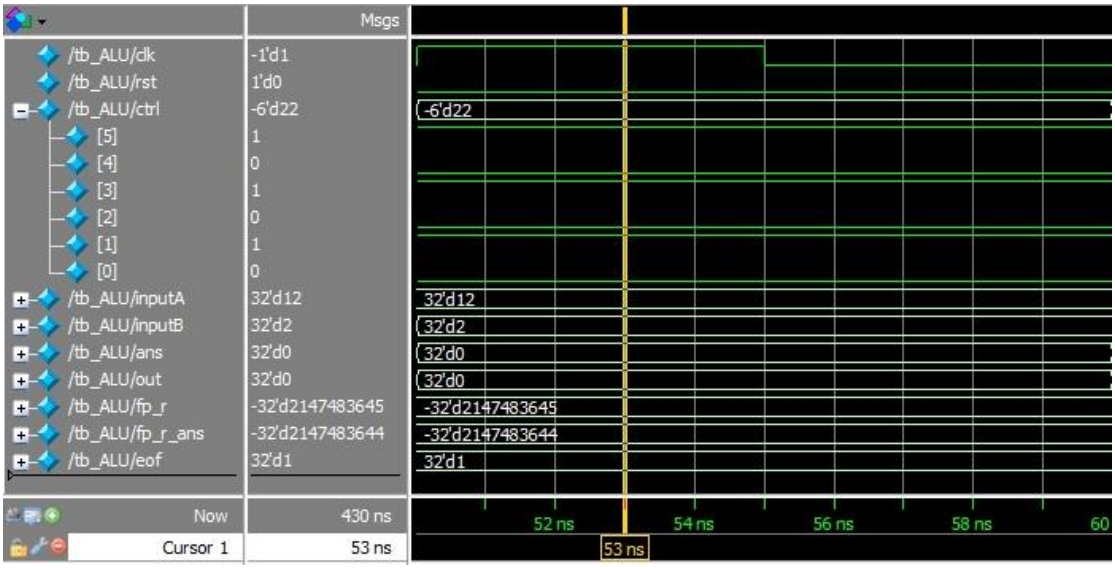
4. OR



輸入:12, 10
輸出:14

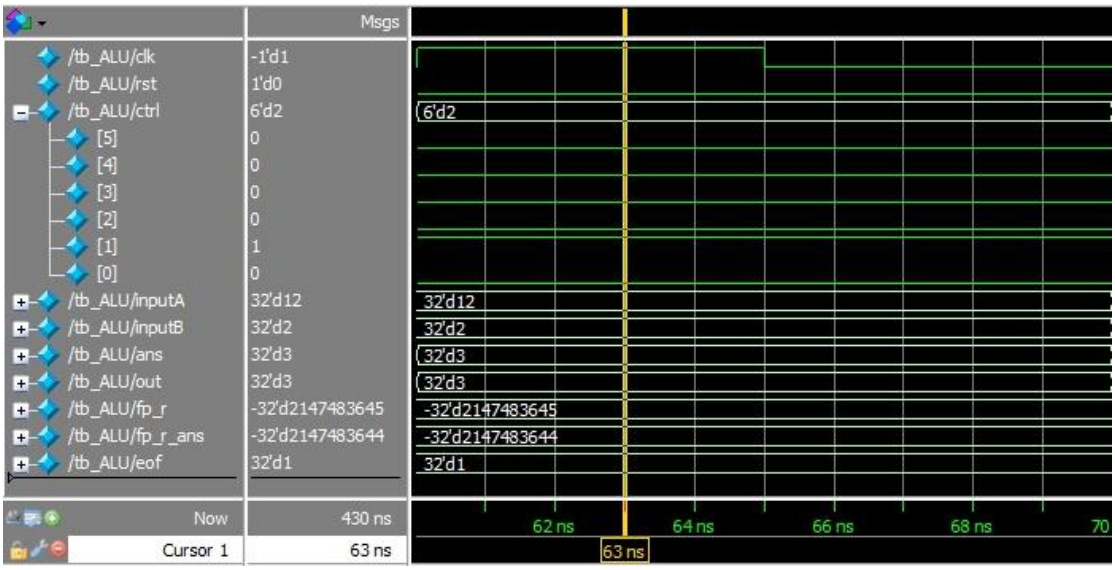
驗證: 觀察 ctrl 訊號為 37，執行 OR 運算。
12 的 2 進位表示為 1100，10 的 2 進位表示為 1010。
將每一個 bit 取 OR 結果為 1110，答案為 14。
觀察 ans 訊號為 14，故結果正確。

5. SLT



輸入:12, 2
輸出:0
驗證:觀察 ctrl 訊號為 42，執行 SLT 運算。
輸入為 12、2，12 小於 2 理論輸出應為 0。
觀察 ans 訊號確實為 0，故執行結果正確。

6. SRL



輸入:12,2

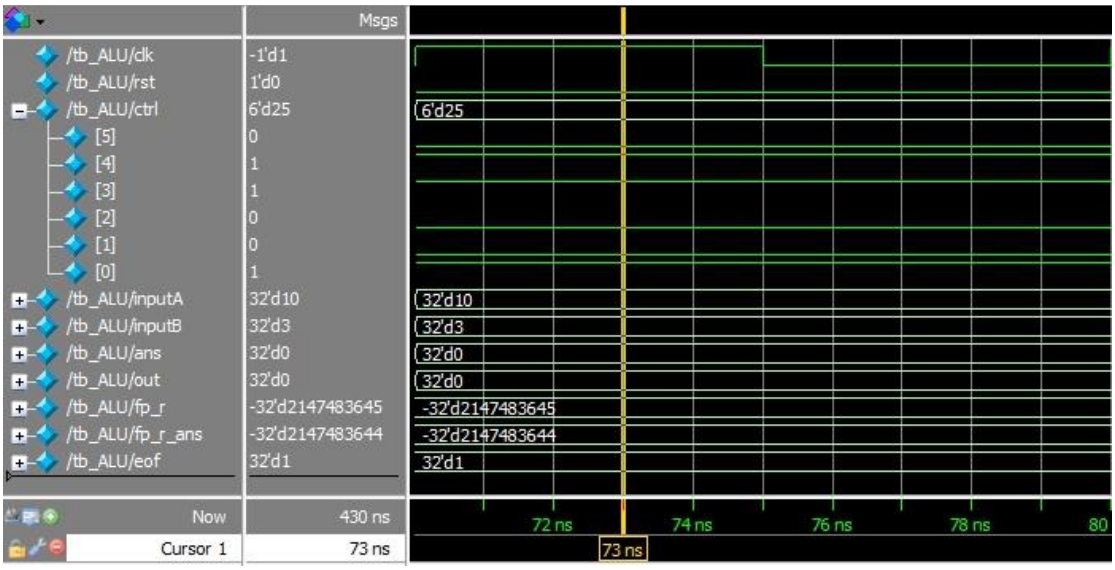
輸出:3

驗證: 觀察 ctrl 訊號為 2，執行 SRL 運算。

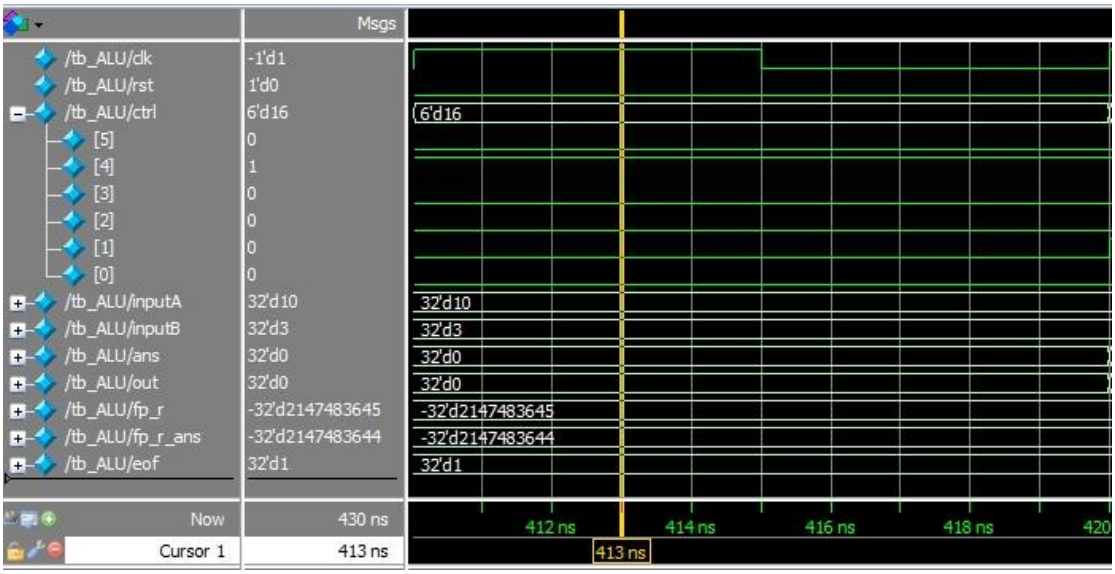
輸入 12、2，執行邏輯右移理論答案應為 3。

觀察 ANS 訊號確實為 3，故結果正確。

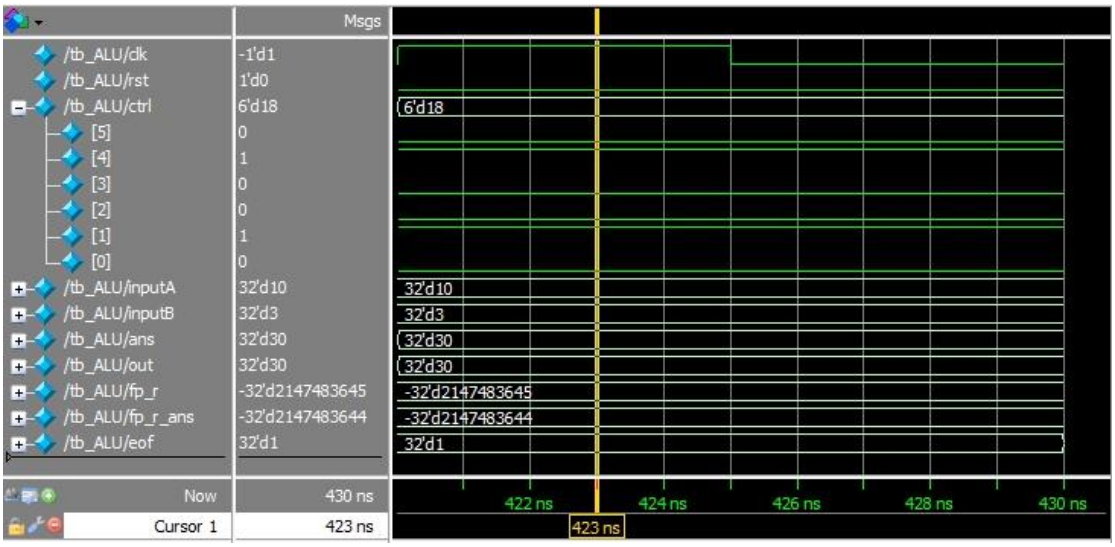
7. MULTU



Hi



Lo



輸入:10,3

輸出:30

驗證:先觀察 ctrl 訊號，一開始為 25，執行乘法運算。

之後乘法運算結束後將結果放入 HiLo 暫存器內。

之後,ctrl 訊號分別為 16(Move from Hi)與 18(Move from Lo)所以分別取出 HiLo 暫存器的值。

觀察 HiLo 的 ans 訊號分別為 0 與 30。理論上 Lo 暫存器應存放乘法結果，比對輸出發現相符，故執行正確。