

109-2 計算機組織 Midterm Project: ALU Design

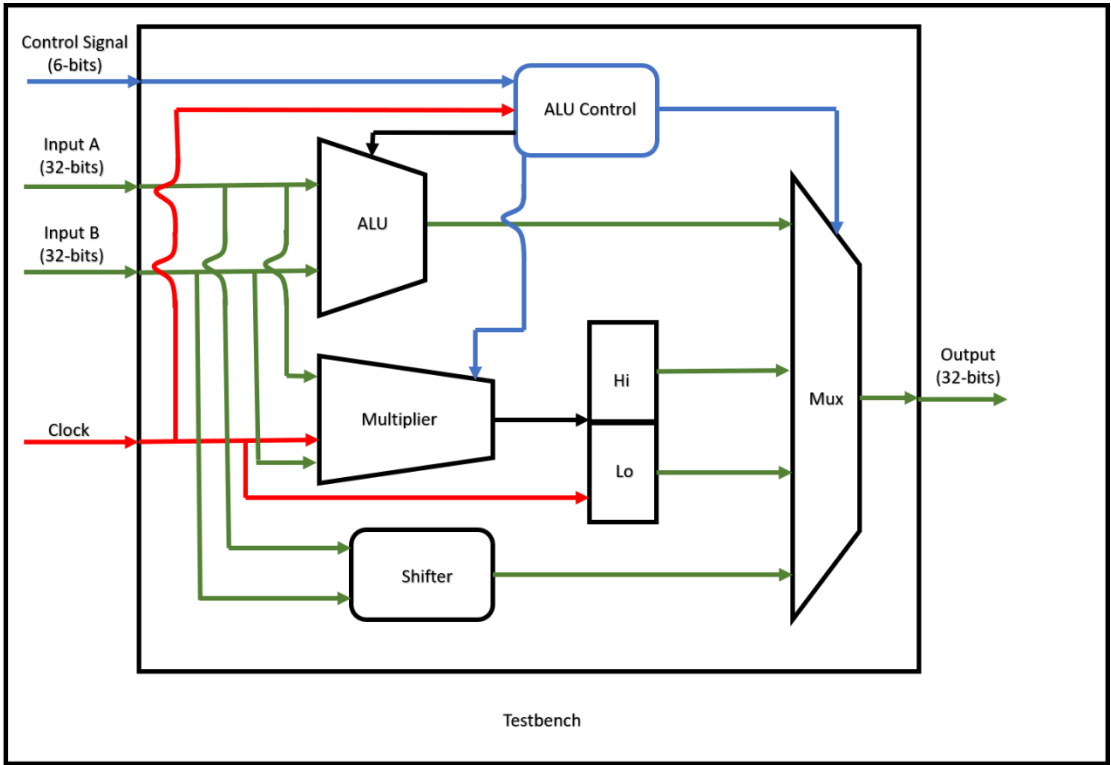
開課班級: 資訊二乙

組別: 第 22 組

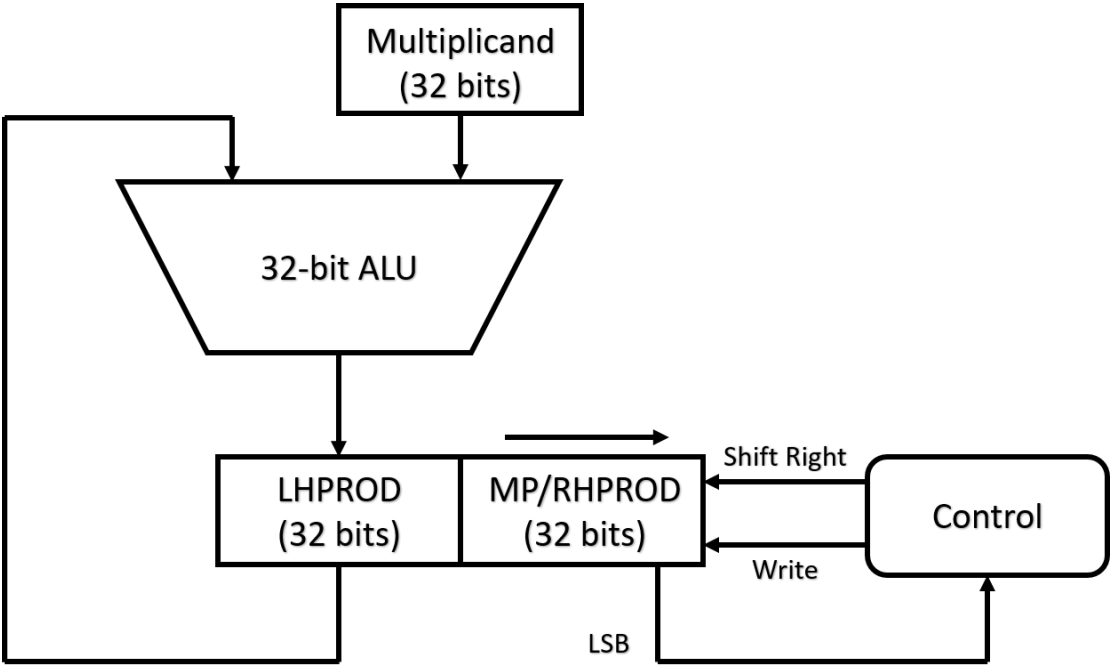
組員 1: 電資四 10520104 莊東翰

組員 2: 資訊四乙 10520131 譚家安

Datapath 和 詳細架構圖:



(圖 1)Mudterm Project 概略架構圖



(圖 2)Multiplier

設計重點說明:

在 tb_ALU 得到題目的 input A 與 B 和 opcode 後 (以一道指令為單位) , 會將這三項資料傳到 TotalALU 執行。TotalALU 會將所有 AND, OR, ADD, SUB, SLT, SLL, MULTU 的電路都跑過一次 , 得到所有指令的運算結果 , 並在最後多工器的地方藉由 6 bits 的 signal (opcode) , 選擇使用者指定的指令結果。

ALU:

ALU 的部分負責 AND, OR, ADD, SUB, SLT 的功能 , ALU 有三個區塊: AND, OR 和 Full_Adder (負責 ADD, SUB, SLT)。

AND 與 OR 的部分重複宣告 32 個元件負責每個 bit 對應的 AND 或 OR 的結果 , 即可得到完整 32 bits 的 AND 與 OR 結果。

Full_Adder 則需要依據 signal 的特性來判斷要如何做運算 , 先觀察:

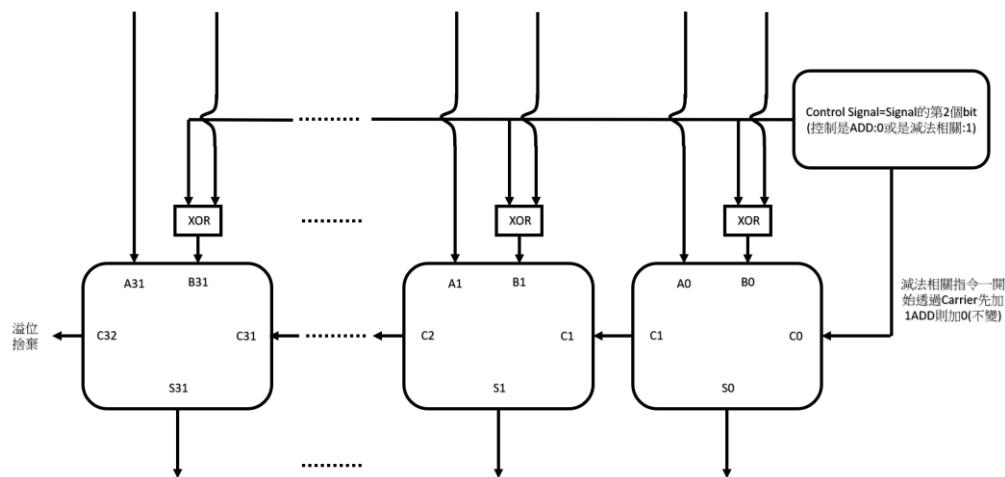
ADD: 32 (100000)

SUB: 34 (100010)

SLT: 42 (101010) (只要相減後最左邊的 bit(符號位元)為 1 則代表結果為負數 , 所以 $A < B$ 。)

會發現只要涉及跟減法(sub 與 slt)有關的指令以 signal 的第二個 bit 做區分 ,

再根據講義的 Full_Adder 設計方法:



由於減法(A-B)的做法就是 $2^{\text{'}}s$ complement($A+(-B)$)，B 的每個 bit 與 Signal 第 2 個 bit 做 xor，如果是 ADD 則 B_i 不變。若是減法相關的指令則 B_i 會 invert 由 0 變 1 或 1 變 0，並且在一開始的 carrier 加上 Signal 第 2 個 bit (ADD:0 減法:1)，即可讓 $2^{\text{'}}s$ complement 最後的+1 在一開始就加入，且此電路也不會破壞 ADD 的結果。

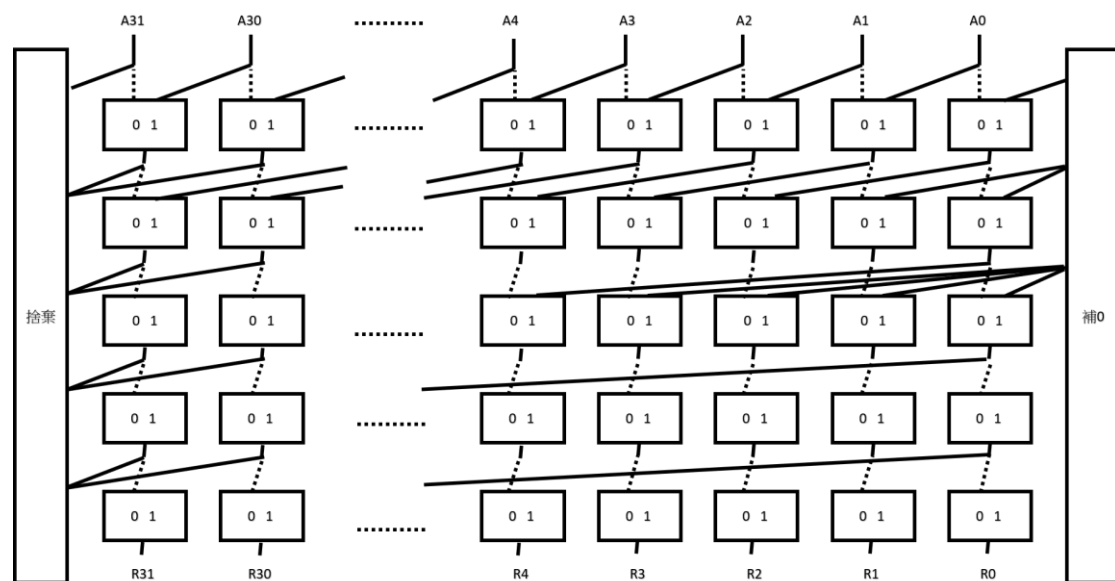
Barrel Shifter (SLL):

由於這次的 input A 與 B 長度為 32bits，所以 Barrel Shifter 最多只能 shift

31(11111)個 bits，也就是最多可以設計五層。如果想左移 3(00011)個 bits，

則在第一層時做移位，第二層也做移位，剩下的三層皆不做移位即可達成。每

一層執行完後都要保留此層運算完的結果 (若沒位移則保持原樣)。



若第一層要位移，則每個 bit 都向左移一位元，最右一位元補 0。

若第二層要位移，則每個 bit 都向左移二位元，最右兩位元補 0。

若第三層要位移，則每個 bit 都向左移四位元，最右四位元補 0。

若第四層要位移，則每個 bit 都向左移八位元，最右八位元補 0。

若第五層要位移，則每個 bit 都向左移十六位元，最右十六位元補 0。

如此設計就能達到移位 0 到 31 個 bits 的所有組合。

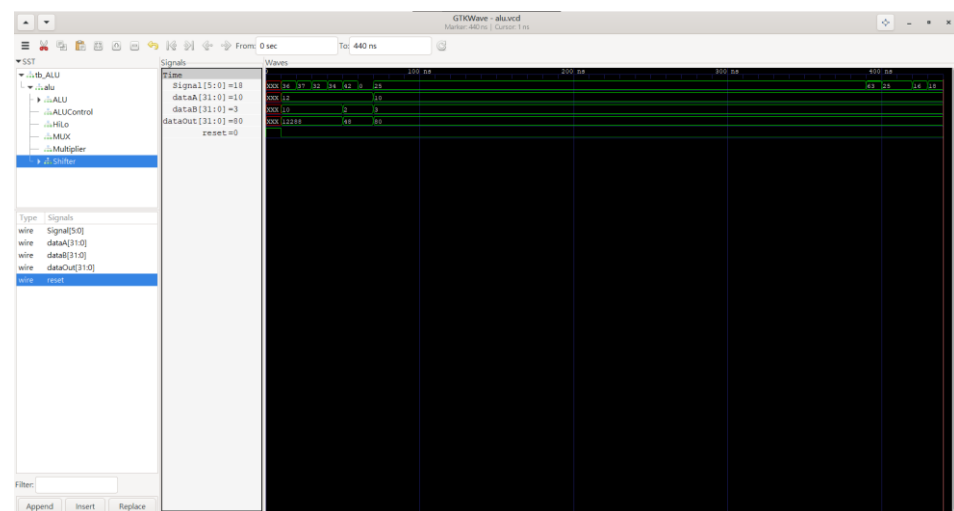
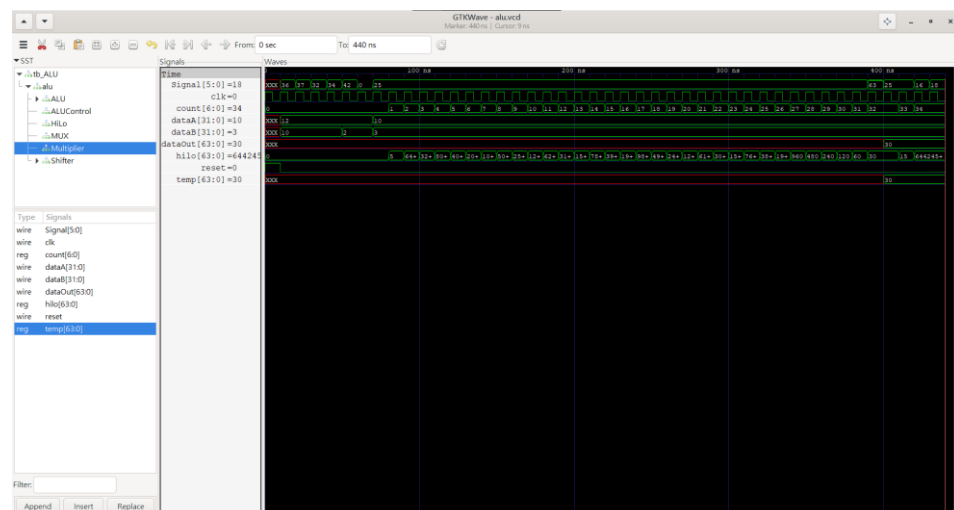
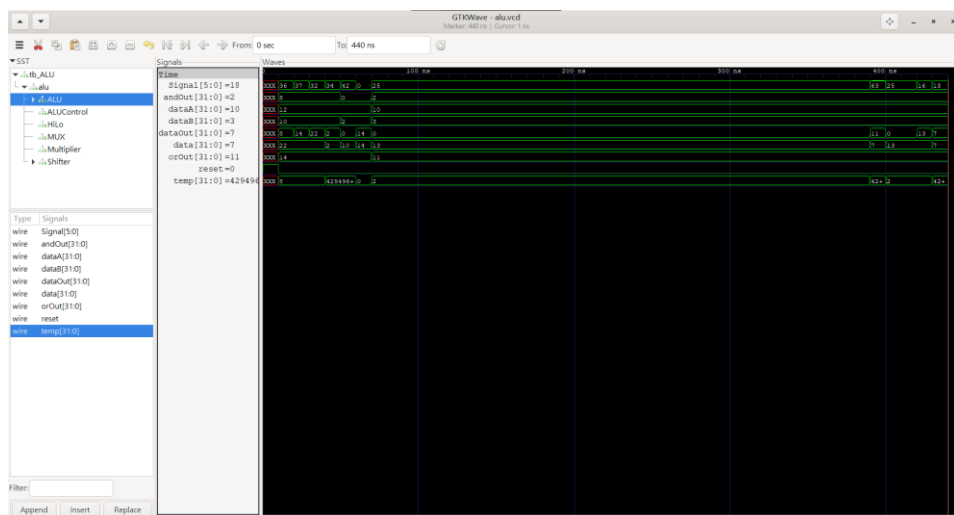
Multiplier (第三版):

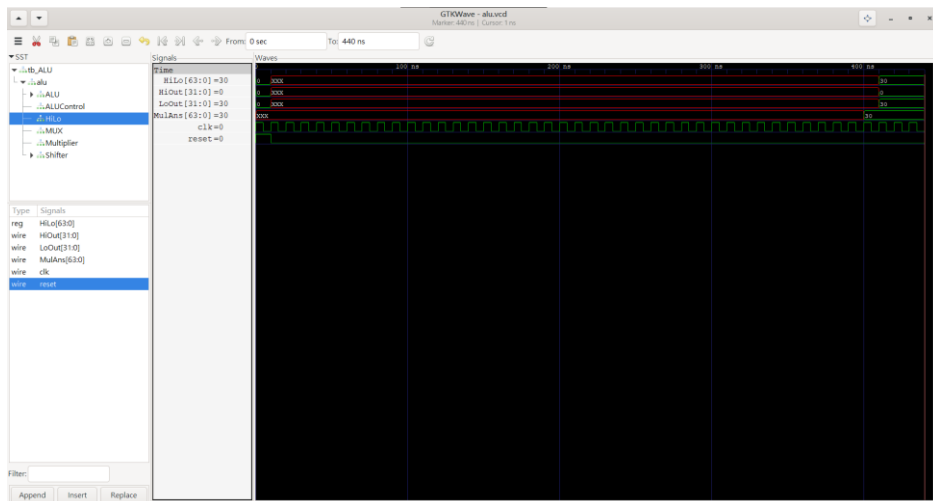
1. 先將 32-bits 的 input A 存入 RHPROD，LHPROD 設為 32-bits 的 0。
2. 若 RHPROD 最後 1bit 為 1 就將 LHPROD 和 inputB 相加結果存入 LHPROD。
3. 將 LHPROD 和 RHPROD 整個右移 1bit。
4. 重複步驟 2. 步驟 3. 32 次。
5. 將 LHPROD 存給 HiLo 暫存器的 Hi，將 RHPROD 存給 HiLo 暫存器的 Lo。

Icarus Verilog, Waveform:

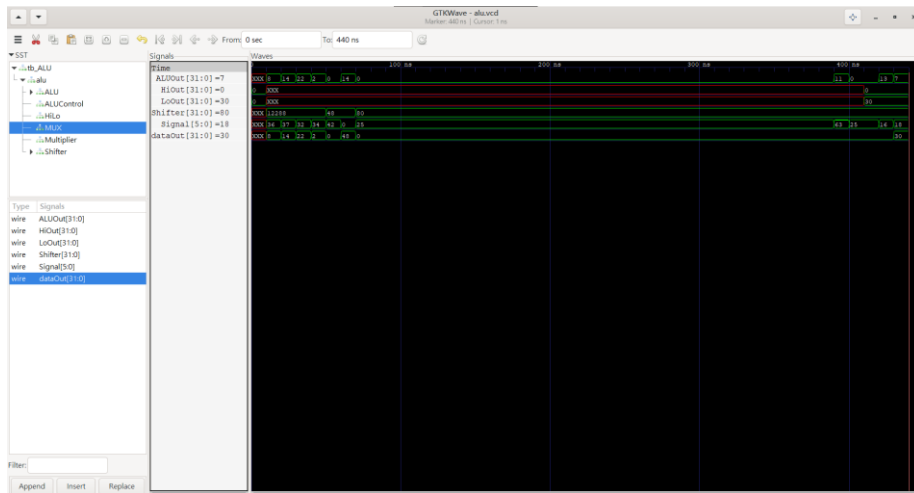
```
Microsoft Windows [版本 10.0.19041.985]
(c) Microsoft Corporation. 所有權保留。 並保留一切權利。
C:\Users\AndreTan\Desktop\期中作業_0517_verilog>verilog -o alu.vvp *.v
C:\Users\AndreTan\Desktop\期中作業_0517_verilog>vvp alu.vvp
VCD info: dumpfile alu.vcd opened for output.
Start
1: Input: AND(36)      12      10
2: Correct: Your answer is: 8, 8
   Correct answer is:
3: Input: OR(37)       12      10
4: Correct: Your answer is: 14, 14
   Correct answer is:
5: Input: ADD(32)      12      10
6: Correct: Your answer is: 22, 22
   Correct answer is:
7: Input: SUB(34)      12      10
8: Correct: Your answer is: 2, 2
   Correct answer is:
9: Input: SLT(42)      12      2
10: Correct: Your answer is: 0, 0
   Correct answer is:
11: Input: SLL( 0)     12      2
12: Correct: Your answer is: 48, 48
   Correct answer is:
13: Input: MULT(25)    10      3
14: Mul End
15: Move Hi
16: Correct: Your answer is: 0, 0
   Correct answer is:
17: Move Lo
18: Correct: Your answer is: 30, 30
   Correct answer is:
Simulation End
tb_alu.v:93: $stop called at 440 (1ns)
** VXP Stop(0) **
** Flushing output streams.
** Current simulation time is 440 ticks.
```

(圖 1)Icarus 執行模擬結果

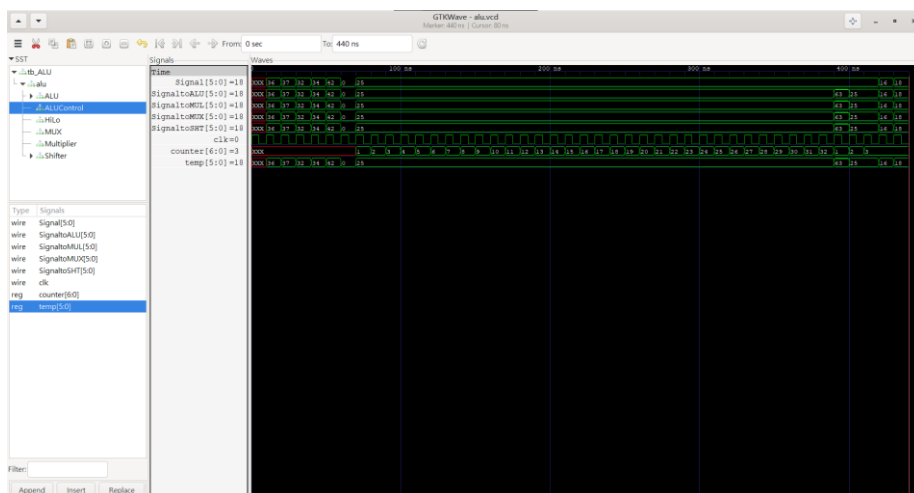




(圖 5)HiLo



(圖 6)MUX



(圖 7)ALU Control

心得感想:

電資四 10520104 莊東翰:

經過這次實作讓我了解 Full Adder 與 Barrel Shifter 的設計方式，特別是從最根本一個一個 bit 的設計讓我更了解為何 opcode 的數字不是隨便規定的，是根據指令的特性來讓 signal 的每個 bit 都有區分指令的功能。本來想把 AND 與 OR 也同時納入 Full Adder 中，但因為一直設計不出同時可以區分 AND, OR 或 ALU 運算的邏輯閘，且用三元運算子時會有不明的錯誤，所以作罷。

資訊四乙 10520131 譚家安:

完成這次 midterm project 後，幫助我對乘法器有更深一步的了解，也更加熟悉 Verilog 的語法，希望 final project 也能順利完成。

分工:

電資四 10520104 莊東翰: ALU(AND, OR, ADD, SUB, SLT), Barrel Shifter(SLL)

資訊四乙 10520131 譚家安: Multiplier, ALU Control, MUX