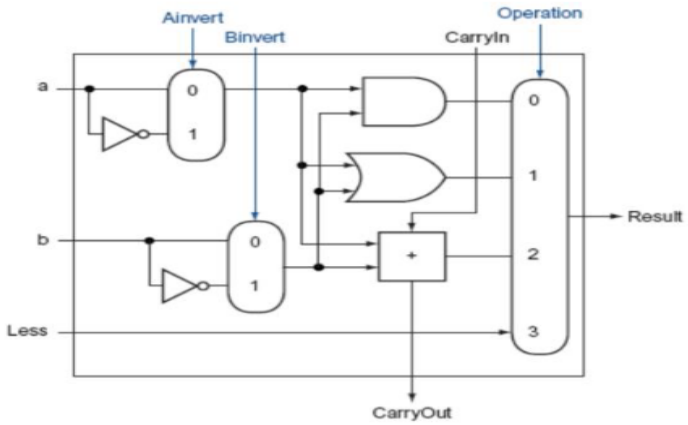
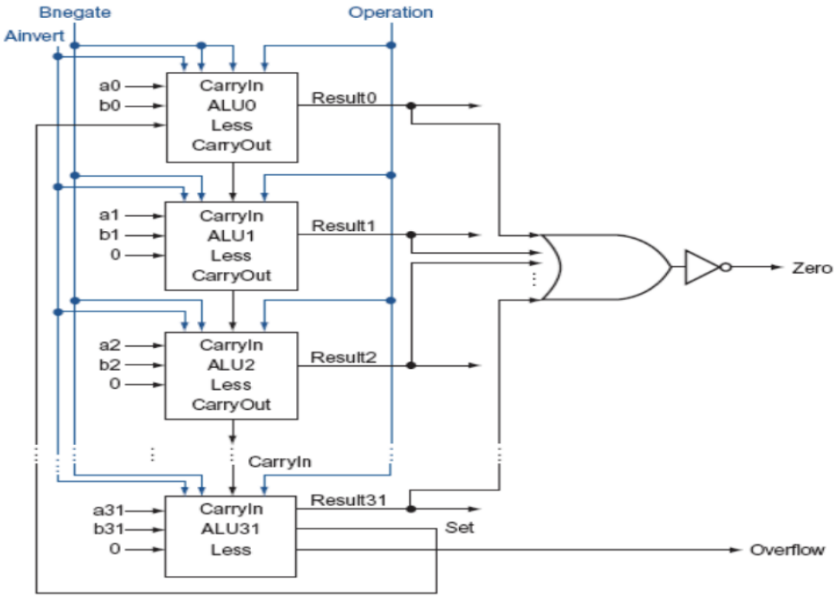
Computer Organization
Lab1: 32-bit ALU

student ID:110550093 Name: 蔡師睿

1. Architecture diagrams



1-bit ALU



32-bit ALU

2. Hardware module analysis

alu_top

首先，設計 1-bit ALU 作爲 32-bit ALU 的 submodule。
1-bit ALU 根據輸入的 A_invert、B_invert 和 Operation 完成相對應的任務，

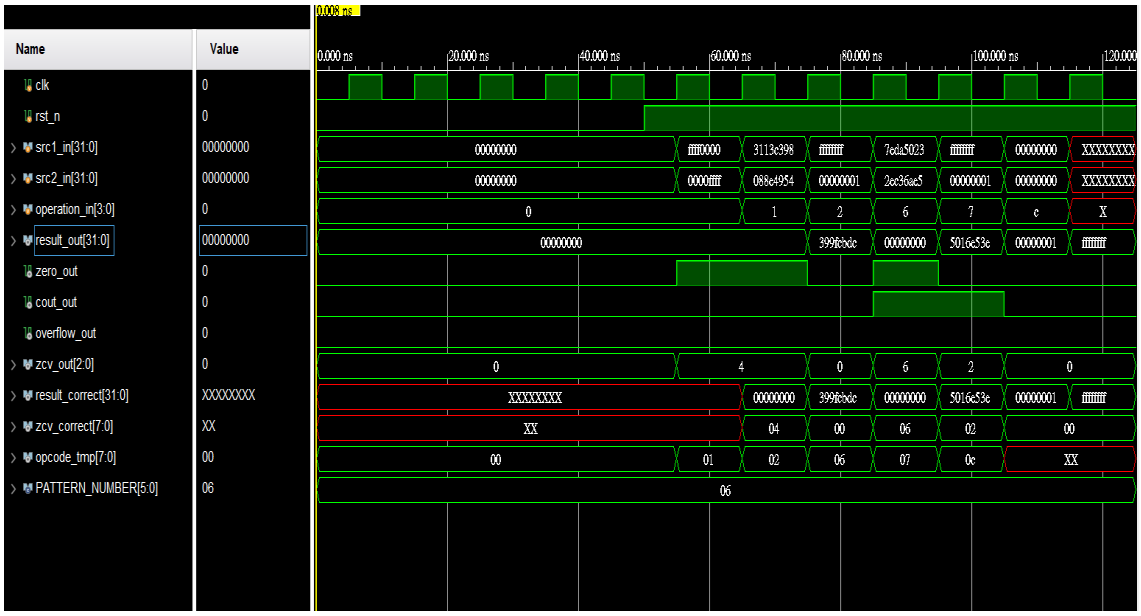
ALU action	**ALU input control ={A_invert, B_invert, Operation}**
And	0000
Or	0001
Add	0010
Sub	0110
Nor	1100
Slt	0111

若 A_invert 或 B_invert 任一爲 1 時，依據 A_invert 和 B_invert 將 A 或 B 做補集，再做相對應的操作。此外，做 Add、Sub 和 Slt 時將 cin 考慮進去，相加後若爲 2 bits，cout 設爲 1，若否則爲 0。最終分別輸出 1 bit 的 result 和 cout。

alu

由 32 個 1-bit ALU 構成，當 rst_n 爲 0 時，輸出 result、cout、zero、overflow 皆爲 0；當 rst_n 爲 1 時，將 32-bit 的 src1、src2 以 1-bit 爲單位輸入到 ALU00 ALU31 中，ALU_control 輸入進 ALU00 ALU31 中，分別對應 A_invert、B_invert 和 Operation。ALU00 的輸入 set 經過眞值表化簡後，得到 $src1[31] \oplus (\sim src2[31]) \oplus (ALU30cout)$。輸入後，將 cout、overflow 設爲 1'b0，若 operation 表示爲 2b'10，cout 爲 ALU31 的 cout，overflow 由 src1[31]、src2[31]、result[31] 判斷是否 overflow。最後若 result 爲 0，zero 設爲 1'b1，反之設爲 1'b0。

3. Experimental result



experimental result

輸出波形與測資所預期的相同，且 console 也出現 Congratulation! All data are correct! 的訊息。

4. Problems and solutions

跑完 testbanch 後，發現我的 slt 和 sub 的部分報錯，檢查後發現 set 的邏輯因爲眞值表化簡錯誤導致，修正後 slt 的輸出便跟測資答案相同，然而，sub 的部分一直找不出 bug，由於減法跟加法相同，只有被減數要先做補數再加法，照理來說加法對了減法也不會出錯，也檢查減法時 ALU00 的 cin 輸入爲 1'b1，然而嘗試了許多次，仍然只有在減法時出錯。最後，發現一個神奇的問題，在 alu_top.v 中，我的 sub 寫法原本是：$\{cout, result\} = cin + src1 + (\sim src2)$，然而改成：$\{cout, result\} = cin + src1 + (!src2)$ 之後，才跟測資答案一樣。由此可知，$\sim$ 跟 ! 在 verilog 加法中得出的結果會不一樣，即使是做只有 1 bit 的加法。

5. Summary

因爲大一下修數位電路設計時就有寫過 verilog，因此只需要回憶一下之前所學的 verilog，並無遭遇到太大的障礙。對於這次作業最大的挑戰莫過於要先了解 ALU 的運作方式，幸好 Lab1 有附 ALU 的電路圖，在理解上不至太過困難，不過由於上述所提到的問題，認爲自己對於 verilog 還未十分熟悉，才導致在減法時出現錯誤。