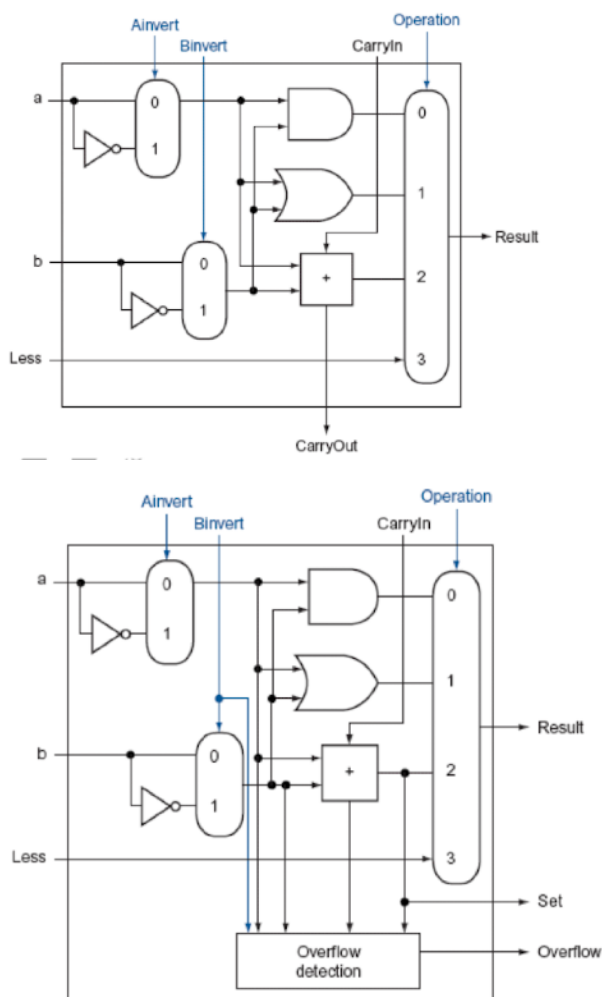


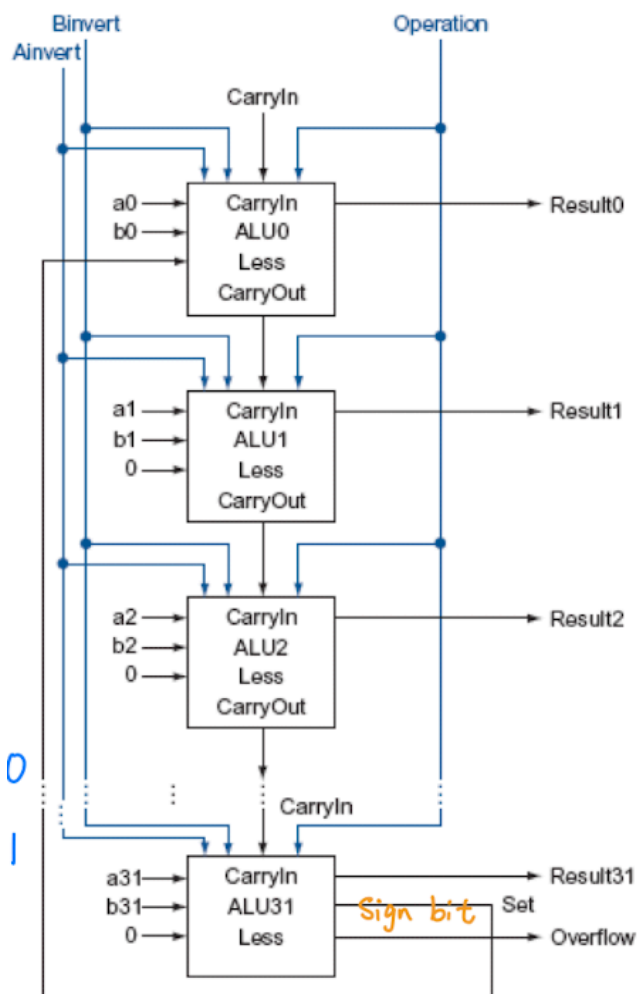
## Computer Organization - Lab1 ALU

### Architecture diagrams

1'bit ALU -> 0 ~ 30



1'bit ALU with Overflow Detection



### Overflow Detection

會發生 overflow 只有兩個狀況，就是當正加正，會是負加負，整理出右表，可以發現只要看 *Carry\_In* 跟 *Carry\_out*，  
 $Overflow = Carry\_In \wedge Carry\_Out \circ (XOR)$

<Note>

正加正的 *Carry\_Out* 一定是 0

負加負的 *Carry\_Out* 一定是 1

### Overflow Detection

MST-A	MST-B	Carry		Overflow
		In	out	
0 (正)	0 (正)	1	0	1
1 (負)	1 (負)	0	1	1

$$overflow = Carry\_In \wedge Carry\_Out$$

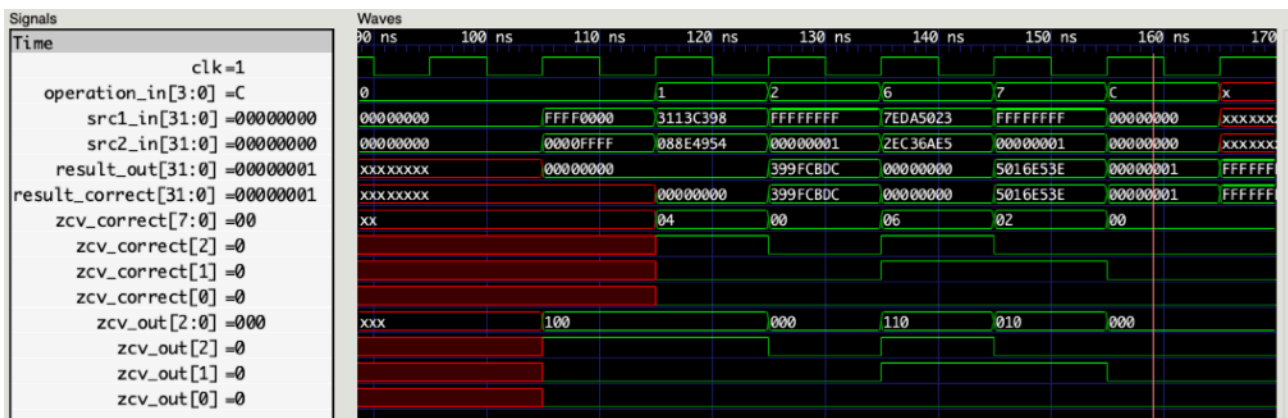
## Hardware module analysis

ALU Action	Name	ALU Control Input
And	And	0000
OR	Or	0001
Add	Addition	0010
Sub	Subtraction	0110
Nor	Nor	1100
Slt	Set less than	0111

這次的共有 6 個操作（左圖），我們可以先把減法當作加上負的，然後將 NOR Operation 觀察一下可以發現： $\overline{(A \vee B)} = \bar{A} \wedge \bar{B}$ ，所以 NOR 可以用 AND OPERATION，把 A\_INVERT 跟 B\_INVERT 設成 1，即可以用 AND 完成。SLT 以  $A + (-B)$  的正負號去判斷是否要將 Le

經過以上分析，可以看到我們可以藉由 ALU Control Input 的最後兩位去決定 And, Or, Add, Slt，然後將第一二位分別當作 A\_INVERT 以及 B\_INVERT 傳入 1-Bit ALU，最後根據結果和操作，去決定 ZCV 值。

## Experiment Result



## Problems & Solutions

這次作業的最大問題就是 Verilog 不會寫，所以在語法上發很多時間，除了上網找資料外，還翻過去寫數位電路概論時候寫的 Code。

## Summary

這次作業的 ALU 是基本的計算單元，這次要以 1-bit ALU 串成 32-bit，藉由這次的 Lab，我對於 ALU 有更清楚的認識，理解其中電路的運作。