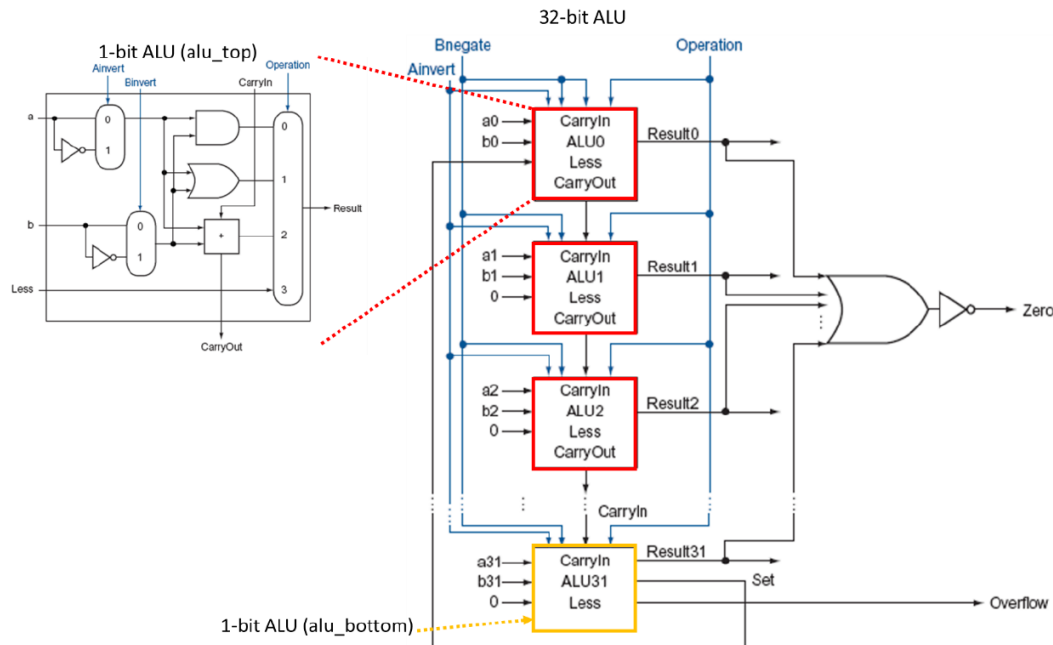


# Computer Organization

## Architecture diagrams:



### Hardware module analysis:

如上圖，用 30 個 alu\_top 和一個 alu\_bottom，alu\_bottom 和 alu\_top 不一樣的地方是有特別處理過 overflow，且第一個 alu\_top 的 cin 要用 B\_negate，less 要用 alu\_bottom 的 set。

此外，除了 Add 和 Sub，其他種運算的 cout 都要設成 0，因為他們不會有 cout

Experiment result:

```
PS C:\Users\ginny\Desktop\計組\CO_LAB1\CO_LAB1\testcase> iverilog -c alu.vvp alu.v alu_top.v alu_bottom.v
testbench.v
PS C:\Users\ginny\Desktop\計組\CO_LAB1\CO_LAB1\testcase> vvp alu.vvp
*****
Congratulation! All data are correct!
*****
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 175000 ticks.
```

### Problems you met and solutions:

1. 不知道 ALU 為什麼有 cout，因為圖上沒有畫

- >>> 上網找資料發現其實就是 alu\_bottom 的 cout
2. 不知道 reg 跟 wire 什麼時候要用哪個  
>>> 向同學請教，發現 assign 要用 wire，always 裡面要用 reg
  3. always 裡面不知道要用很多參數 or 起來的東西還是要用 \*  
>>> 後來用 \* 測資就過了，所以決定用 \*
  4. 以為要手打參數，打 32 次之類的  
>>> 查到可以用 generate 和 for 迴圈

## Summary:

雖然不是第一次寫 verilog，但還是覺得好難，常常有不知道為什麼產生的 bug，而且 2's complement 那些東西有點忘記了，還要再多複習。