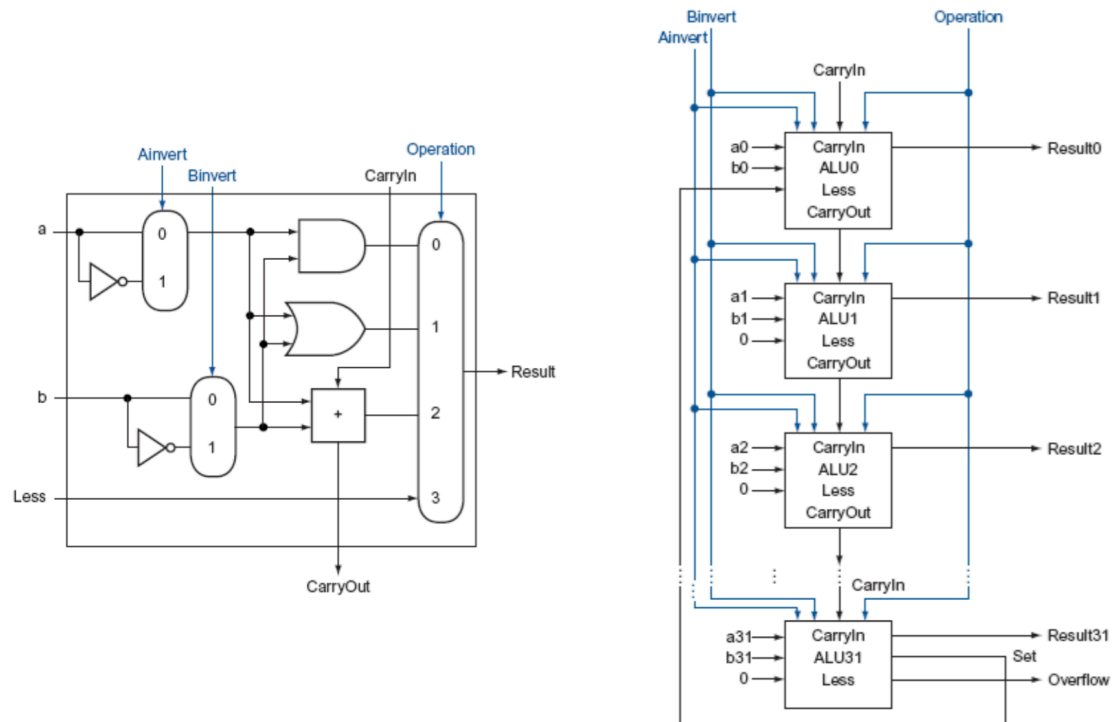


# Computer Organization

0316055許庭嫣 / 0316313張逸群

## Architecture diagram:



## Detailed description of the implementation:

ALU 算術邏輯單元，在另外架設的 `alu_top` 的 module 裡面做基本的運算得出每 1 bit 的結果，最後再將 32 bits 串接起來，得出真正的答案。

AND、OR 是基本的運算，結果直接參考兩個 source 當個 bit 比較的結果。NAND、NOR 則是先讓兩個 source 做 invert 再代入做基本的 AND、OR 運算。加法是用類似 full adder 的算法，由前一個進位和兩個 source 做運算得出結果，並產生新的進位影響下一個 bit 的結果；減法是將第二個 source 做 invert，代入加法並讓第一個進位為 1(減法等同加上第二個數的補數)，使得結果仍為兩數相減。Set less than 是做兩個數相減的運算，若相減的做高位(sign bit)為 1，則將 LSB 設為 1，反之則為 0，其他 bits 則常設 0，所以在做 32 bits 的連接時有些線不會影響結果。

例外處理上，overflow 出現在兩正數相加 sign bit 為負、兩負數相加為正、正負相減為負、負正相減為正的情況下；zero 在當結果為 0 時將值設為 1；cout 的結果就依最後一個 bit 做運算得出的進位。

## Problems encountered and solutions:

Verilog 對我們來說仍有些陌生，尤其在缺少經過一段時間接觸的情況下，對於語法的處理會有些遺忘，不過在翻看之前的程式碼後，就能些微想起了做法。然而仍會有一些小問題出現，需要同組同學互相合作、幫忙檢查與提醒。

第一次寫完程式碼跑測資時就直接通過了，但以為萬無一失的程式碼在之後仔細檢查又發現了一些漏洞和錯誤，所以需要多做檢查和多筆測資作為勘誤比較保險。

## Lesson learnt (if any):

雖然是老師上課講過的東西，但自己寫一次時就會對做運算的方式更為理解，也發現了 ALU 的簡單與好用之處，雖然速度可能會慢一些。

Set less than 的做法一直搞不太懂意思，在同組同學的互助之下，終於弄懂了意思，才發現他真的只是基本的觀念而已。