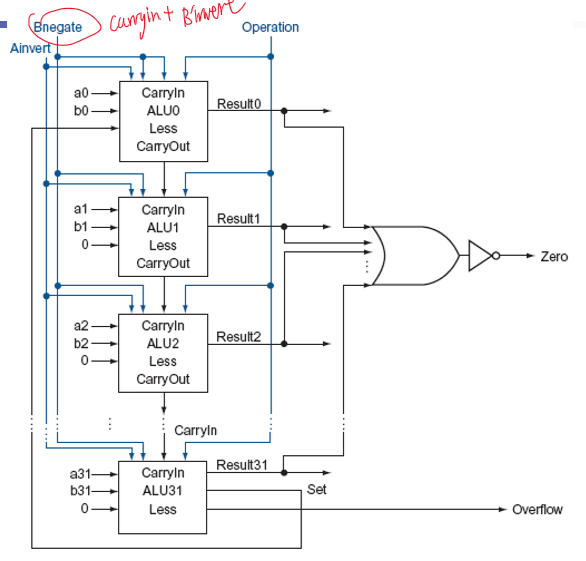
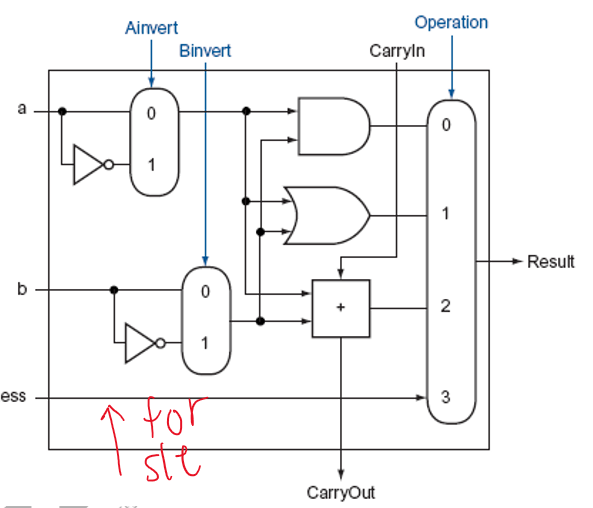
**Computer Organization**

**0511105 李頤**

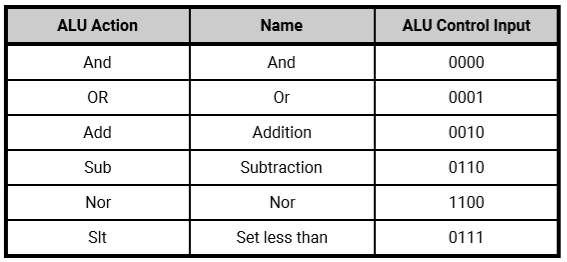
**Architecture diagrams:**

****

* 基本上都是用講義上的架構，左邊為alu\_top(1bit)，右邊為32bit
* 使用的模組: mux2, alu\_top, alu
* Environment : Xilinx vivado
* 因為專案路徑的問題:我資料夾裡的CO\_LAB1專案可能不會動

(but I included it anyway)

**Hardware module analysis:**



* **ALU Control Input – Format Explained**

總共有4 bit，但alu單元的op code 只有2 bit，所以這裡的格式

拆分成左右半邊(各2 bit)來實現上述operation

右半邊: 送進每個alu的op code

左半邊: 為A\_invert 和 B\_invert

所以

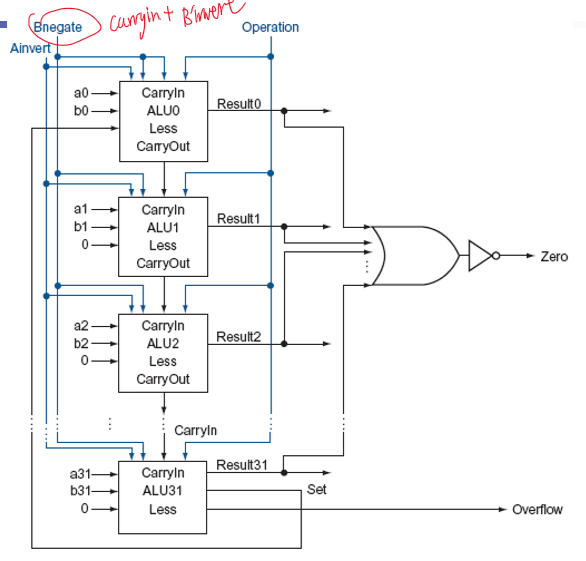
Sub : b invert (01) 後加上 1 ，然後與A相加(10)，所以是0110

Nor : a, b, invert(11) 後做and (00)，等於Nor(1100)

* **ALU Action – Hardware Implementation Detail**
  + **And** 把輸入訊號src1, src2丟入and gate
  + **Or** 丟入or
  + **Add** src1 ^ src2 ^ cin (xor)
  + **Sub**  lsb 的cin 接 b\_invert，達到flip bits + 1 (二補數)
  + **Nor**  a nor b = !a and !b
  + **Slt**  拿a, b相減結果的msb(sign bit)如果是1表示a<b

拉線回lsb alu的less 輸出1

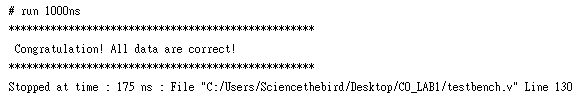
* **ZCV**

****

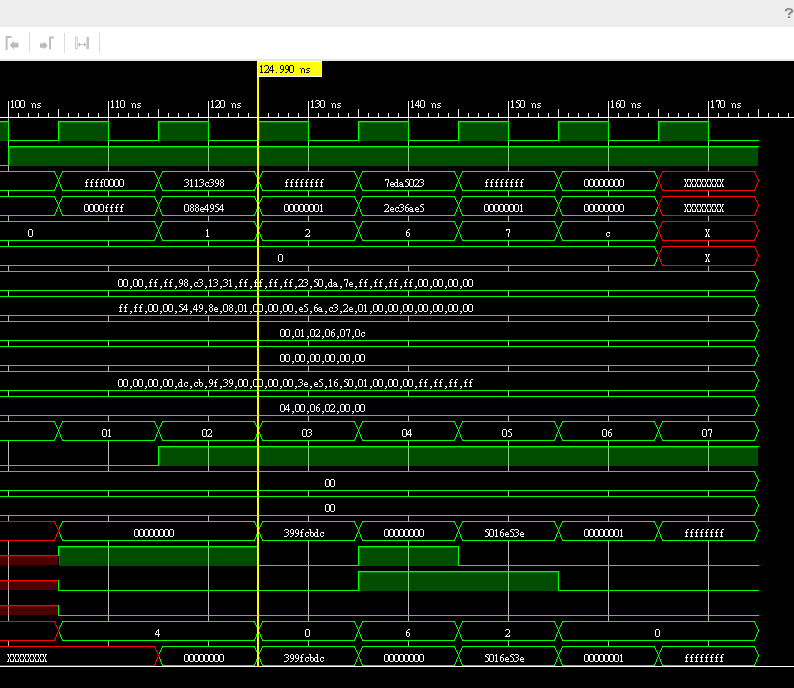
* + **Zero :** 把result or起來，然後再flip，如圖
  + **Carry :** 加或減時輸出carry，其他operation固定輸出0
  + **Overflow :** 參考此YouTube 教學
    - <https://www.youtube.com/watch?v=p4yVpZGZ9tA>
    - 將最後兩個alu的cout 做 xor

**Experiment result:**

* **Console Result**

****

* **Simulation Result**



**Problems you met and solutions:**

1. **結果早一個clock輸出**

我多加了一個ans\_temp register來儲存答案，再posedge時寫入

上一個時脈計算的結果的結果

1. **Add 的結果不正確**

carry 一直不對，是因為後面的alu沒有讀到前面alu新算出的carry，

解決辦法為在alu單元的always增加cin，

always@( ……cin)

這樣 cin更動時便會重新計算

1. **Slt 實現方式**

因為slt需要a, b相減取得msb的result，但op code已經固定，我目前想不到有什麼比較好的方法讓alu減法完後再選擇less輸出，所以我用了比較直接的方法，多加一條wire temp\_diff = src1 - src2，也就是相減的動作是在我的alu外完成的，取得相減結果的msb拉回lsb的less

此方法精神上應該差不多，希望不算太暴力@@

1. **Nor zcv的c輸出錯誤**

忘記除了加減會有carry，其他operation都是零，在alu單元specify 一下就好了

**Summary:**

* 學到基本alu的運作方式
* 學到opcode 的運作方式
* 學到各operation的實現方式