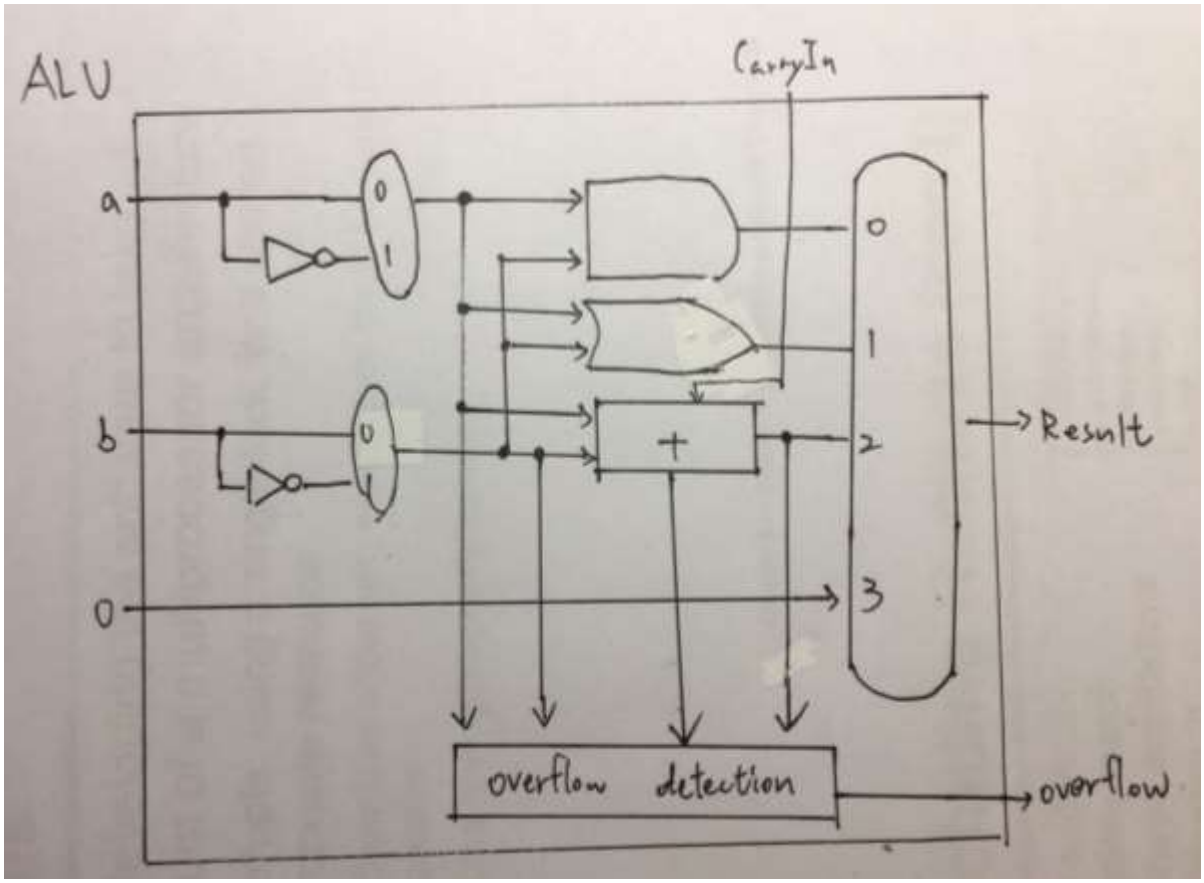
Computer Organization

0413118_0610016

Architecture diagram:



Detailed description of the implementation:

本次 lab 是透過 alu 輸入的指令來判斷執行的運算模式，輸出運算完的結果以及是否有造成 overflow 或是 carry out 的情況。

首先，and 和 or 這兩項指令只需用 verilog 的運算式&或是|就可以比較兩運算元的二進位數；反之可以套用在 nor 以及 nand 上，只需要在答案前給一個(~)就可以。

再來加減運算雖然只需將兩輸入運算元做相加或相減的動作就好，但是需要在此偵測是否出現了 overflow 或是 carryout 的情況，overflow 只需偵測 result 的 signed 值是否超出 2147483647 或是-2147483647 的範圍就可以判斷是否有 overflow。再來是 carryout 的部分，我是藉由判斷 src1 和 result 運算元的變化，其中像是在加法中 src1 某位元的值從 1 變回 result 中的 0，就代表有進位的產生；反之減法也適用。

Slt 是透過將 src1 和 src2 存入 32bits signed 的暫存器中，來比較兩者大小，src1 小於 src2 則輸出 1 反之 0。

至於答案是否為 0 只需用一個條件式來判斷是否 result 為 0 就好。

Problems encountered and solutions:

1.我可以在 vivado 上執行 testbench smulation(照片 1)，但在助教指定的程式上執行沒有顯示文字。

2.無法理解為何需要 alttop 的參與，因為我已經可以在 alu 上直接解決基本上的結構運算。

3.關於 basic 的 number 7 SLT，為何

ffffffff 00000001

結果會是 1，是因為這是 signed 的關係嗎?

解決方法：唯一合理的解釋就是這兩項運算元是運用 signes 的方式計算大小，所以兩項的十進位顯示就是-1 和 1，要在 verilog 裡比較 signed 的大小需要在變數宣告時在前面加上 signed 的指示。像是 reg signed [31:0] test。

Lesson learnt (if any):

1. 關於 signed 和 unsigned 的運算差異。
2. 練習撰寫 Verilog