Computer Organization, Spring 2019

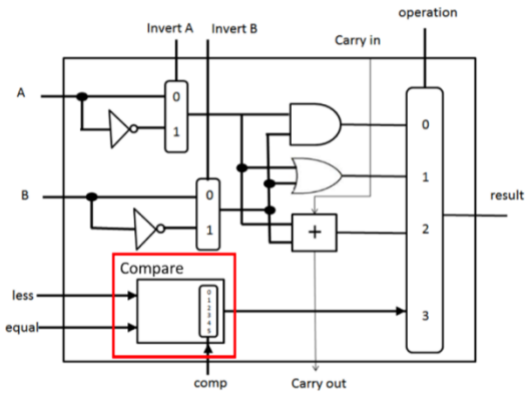
Lab 1: 32-bit ALU

Name: 王耀德

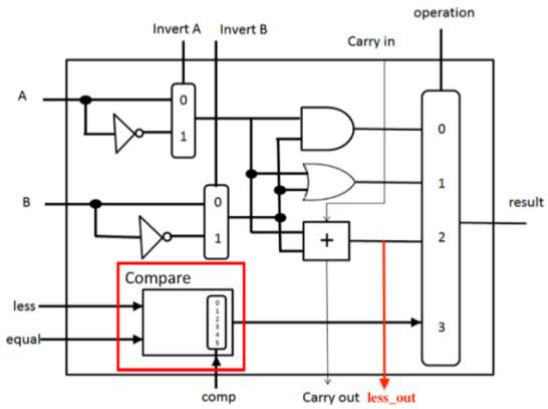
Student ID: 0716055

(1) My architecture diagram

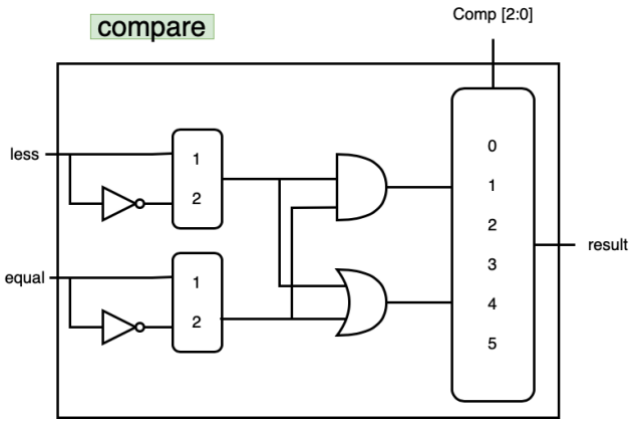
1.alu_top:



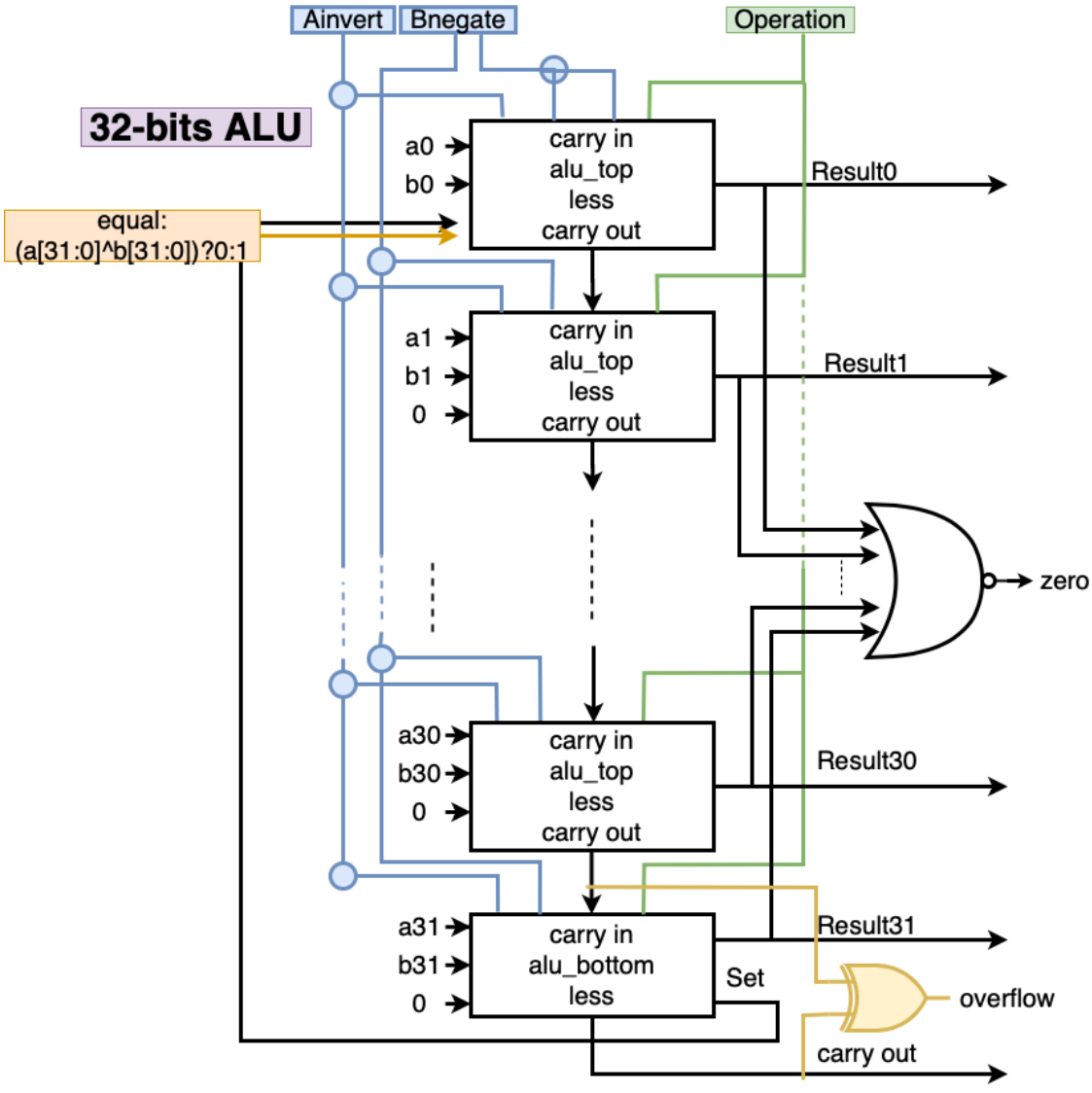
2. alu_bottom:



3. compare:



4. 32-bit alu:



(2) Detailed description of the implementation

這次作業主要包含四個 module，分別是 1 bit 的 alu_top, alu_bottom、compare，並組成最後的 32-bit alu。先在 alu_top 中做好 and, or, adder，搭配 invert 就可以完成題目的前六種功能。

而 slt 則是將減法差值的最後一位，也就是差值的正負傳入作為 less，然後與 equal 進入 compare 做判斷，所以 alu_bottom 多出一個 output 就是減法的輸出，傳入第一個 alu_top 作為 less。比較要注意的是 slt 的 result 只有在第一位，所以其他位數的 slt result 都輸出 0 即可。

在 32-bit alu 中，前三十一個使用 alu_top，最後使用 alu_bottom，最後會得到 32-bit 的 result 跟 carry out，並把結果拿來判斷 zcv。Cout 是最後位 carry out，Overflow 是把第 30 跟 31 個 alu 的 carry out 做 xor 判斷，如果不同則 overflow。Zero 是將所有 result 做 nor，代表只有 result 全零 zero 才會為 1。

(3) Command for compiling your source codes

```
iverilog -o bonus.vvp testbench.v alu.v alu_top.v alu_bottom.v compare.v
```

(4) Problems encountered and solutions

已經很久沒寫 verilog 了，花很多時間想 module 要怎麼用，還有 register 跟 wire 的用法，這部分是看編譯出來的錯誤訊息慢慢修正。除此之外，一開始測試時減法一直有部分錯誤，中間都差一位數，後來是檢查 alu input 的欄位有一格 13 填成 12，細心檢查、修改就解決了。再來是第六筆測資一直失敗，檢查 SLT 的部分也沒有寫錯，後來發現是 equal 的地方沒有寫好，最後修正 equal 的判斷，就完成 lab1 了。

(5) Lesson learnt (if any)

學到 alu 的組成，還有熟悉 verilog，對之後的 lab 應該會更容易上手。