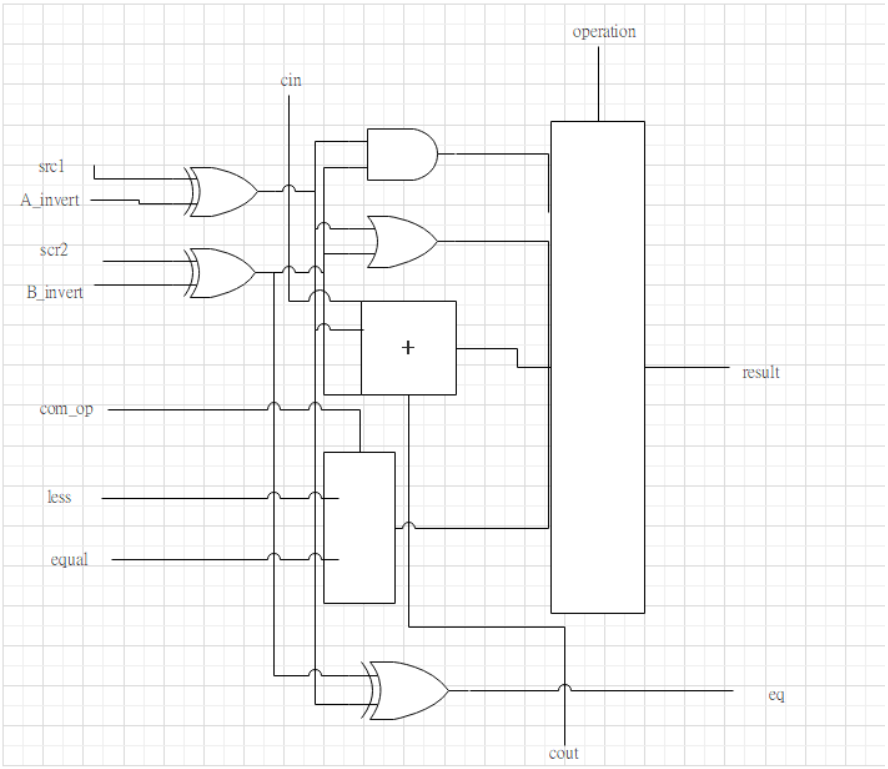
Lab1

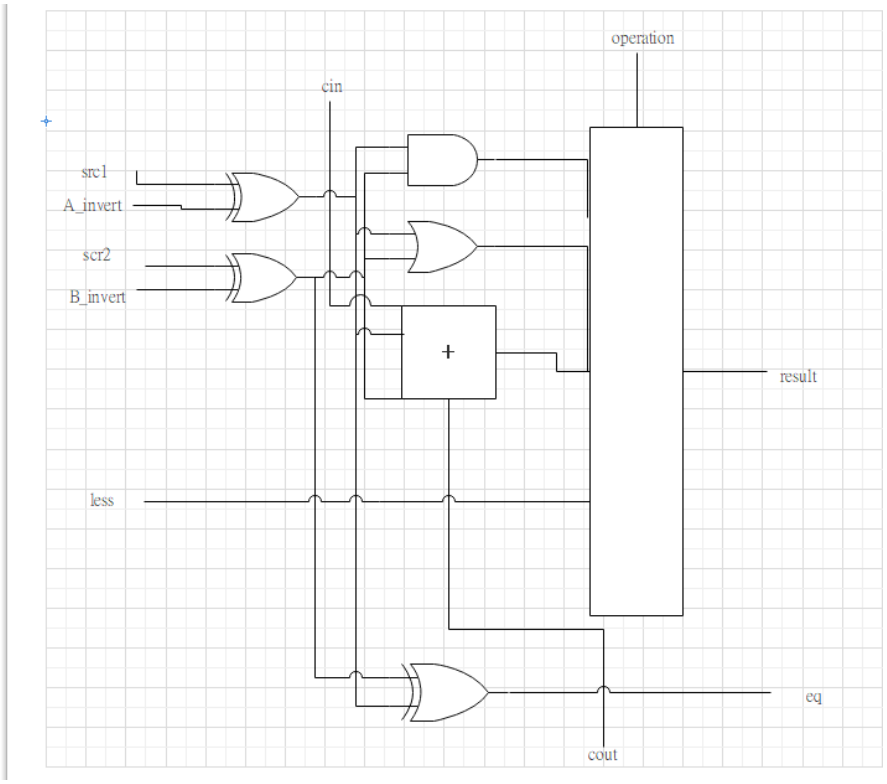
0710028 陳敬諺

1. Diagram

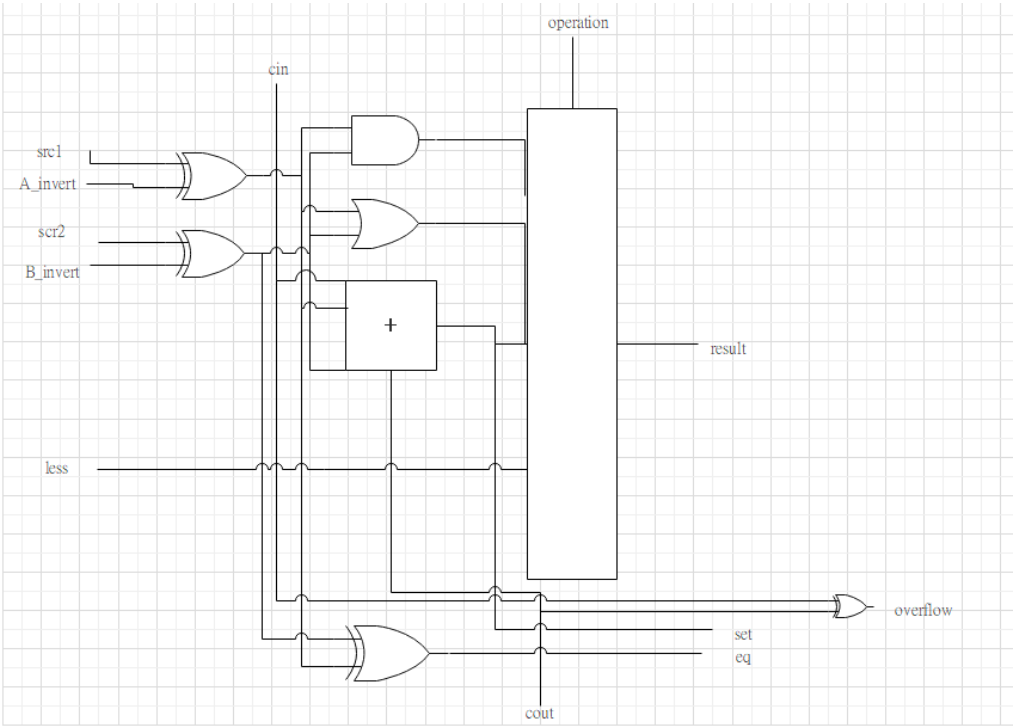
alu_TOP0:



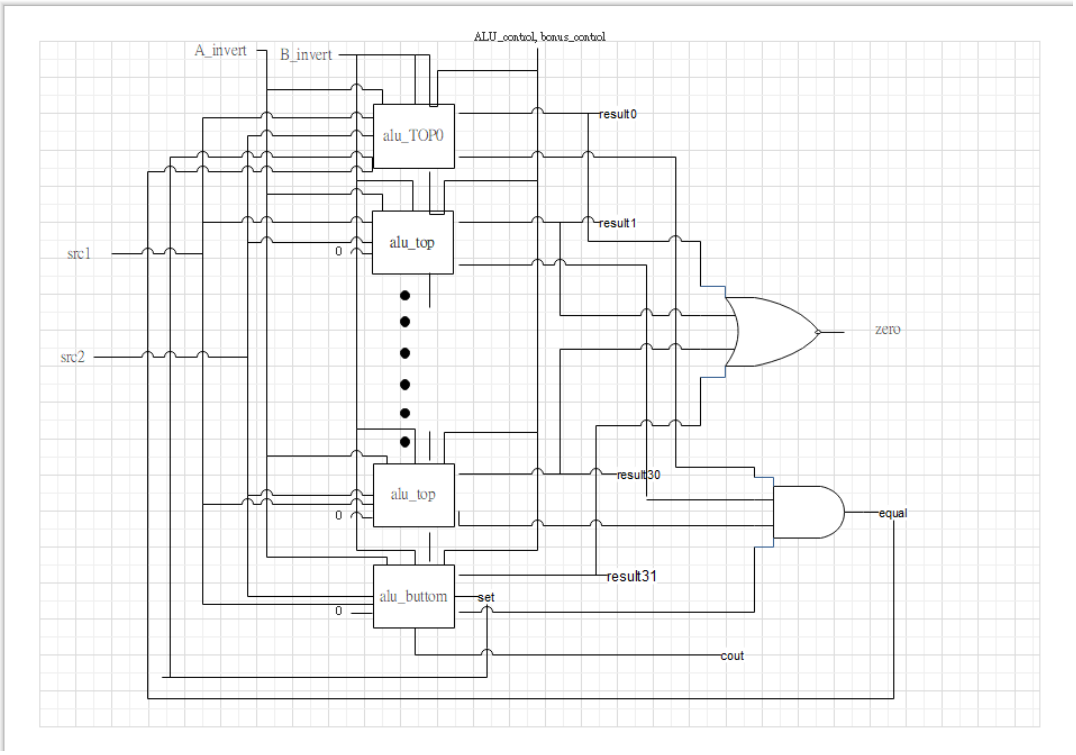
alu_top:



alu_buttom:



alu:



2. Detail

1-bit alu 分成三個部分解釋:

alu_top (for result[1]~[30]):

m1,m2: 讀取 scr1,scr2 是否有 invert 的結果

op1: m1,m2 做 AND 運算

op2: m1,m2 做 OR 運算

op3: m1,m2,cin 做 XOR 運算，及 result[n]做 ADD 或 SUB 的結果

op4: SLT,SGT,SLE,SGE,SEQ,SNE 的結果，由於除 result[0]以外，其餘皆為 0，所以我選擇直接與 less 連接，簡化電路

eq: 比較 sr1,scr2 是否相同

cout: 計算 m1,m2,cin 若有兩個 1 以上則進位

alu_buttom (for result[31]):

與 alu_top 相比，這邊多了以下功能:

set: 表示 scr1 是否小於 scr2，方法是利用 scr1-scr2，當 scr1-scr2 < 0，代表 alu_buttom 的計算結果為 1，可直接作為 set 值

overflow: 表示運算結果是否出現滿溢，以 cin,cout 做 XOR 來判斷

alu_TOP0(for result[0]):

因為 SLT,SGT,SLE,SGE,SEQ,SNE 的 output 只有 result[0]會有影響，所以需要另外做判斷，利用 K-map 可得到 equal,less 與 com_op(bonus_control)的關係式。

alu:

實際完成 32-bits alu，另外還有以下功能

zero: 對所有的 result 做 NOR 運算

equal: 對所以 1-bit alu 的 eq 做 AND 運算，並做完 alu_TOP0 的 equal 輸入

3. command

iverilog –o bonus.vvp testbench.v alu.v alu_TOP0.v alu_top.v alu_buttom.v

4. problems & solution

整個過程中比較棘手的問題主要是 less 的判斷，原本是單純想從高位往低位數比較，不過後來覺得過於麻煩才發現用減法判斷是最好實現的方式。

5. lesson learnt

之前有修過數位電路實驗，算是再次複習吧，不過還是花了不少時間。