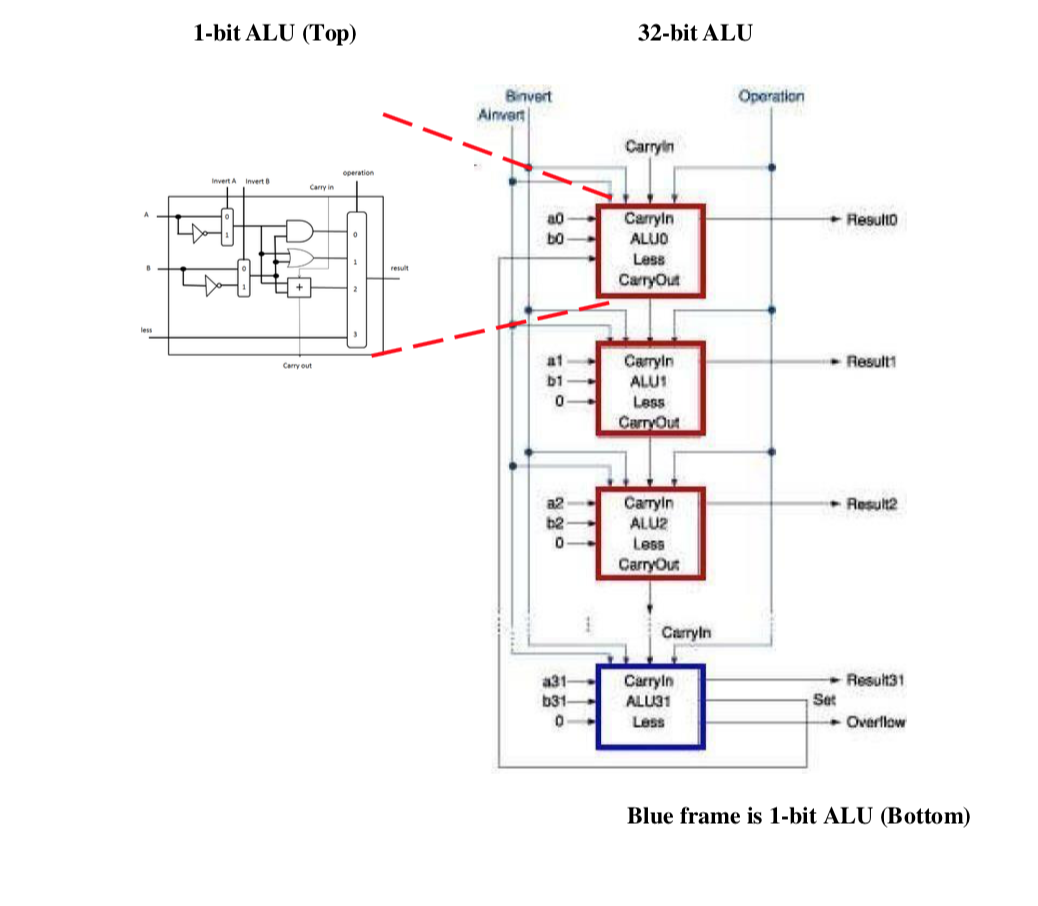
**Computer Organization**

**0516022 吳禹璇**

Architecture diagrams:

Hardware module analysis:

一開始是從1 bit的ALU先寫起(alu\_top.v)，我設了兩個reg (a, b)來表示src1跟src2輸入之後要不要做invert，接著從得到的a, b設計and, or, add, sub, nor, slt。

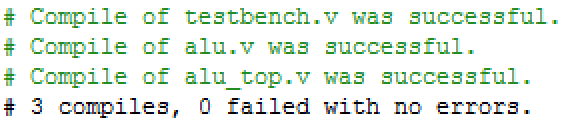
再來就是要寫alu.v的部分，我一共接了32份的alu\_top，並在最低bit的less\_input及first\_cin做修正，接完線之後就可以把32bit的result輸出。最後就是處理zero, cout, overflow這三個結果。

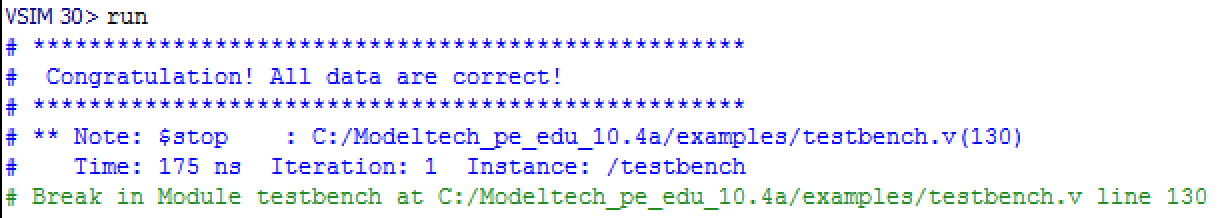
1. zero我是預設為0，如果輸出的result是0，zero就改為1。

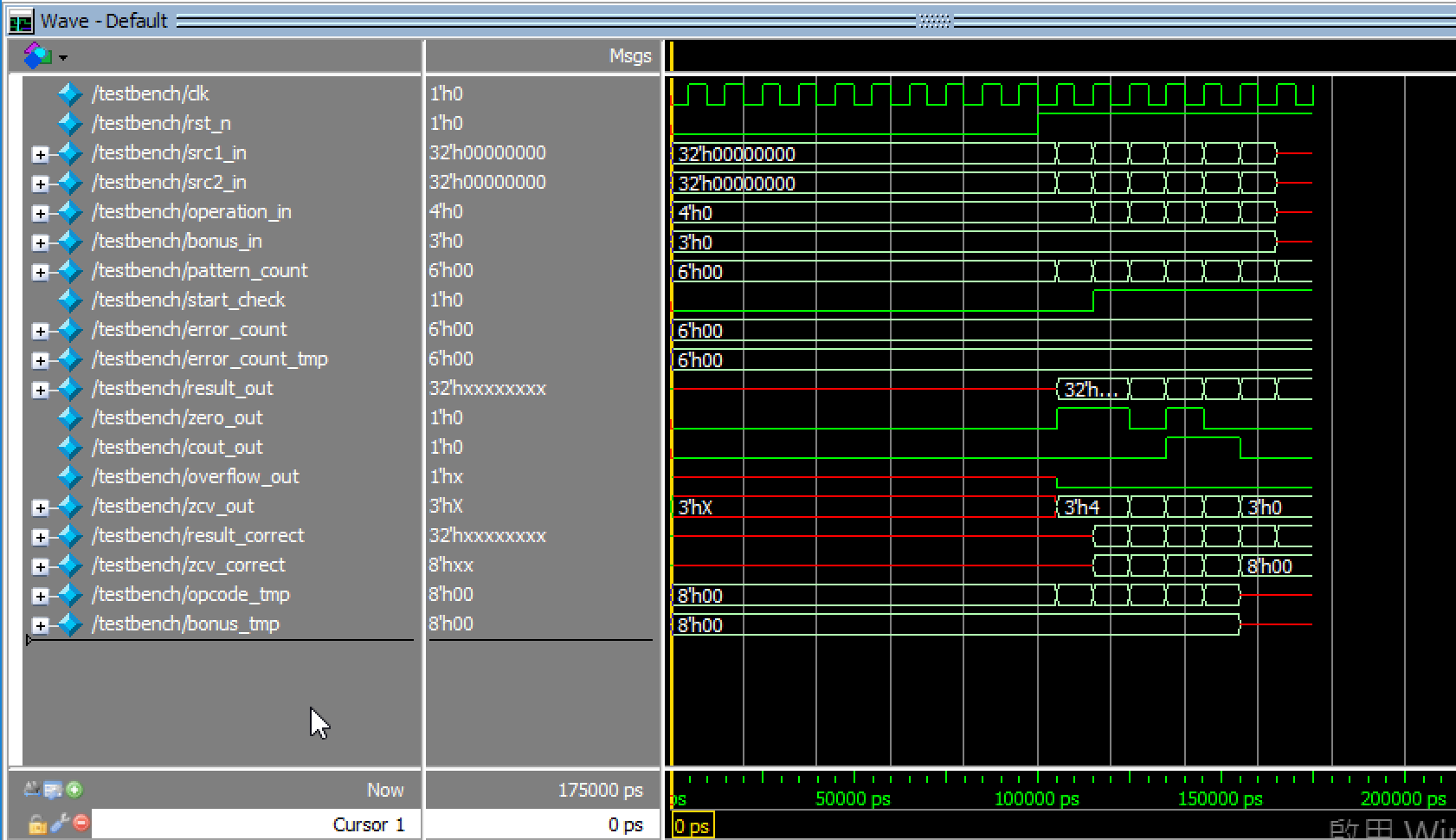
2. cout則是在做加法、減法才會有，所以我給了一個判斷，若是加減法才會去判斷有沒有cout，如果有，就直接讓它等於tmp\_cout。

3. overflow可以由第30及31 bit的tmp\_cout是否相同判斷，若是不同就代表發生了overflow，所以我用xor來做。

Experiment result:







Problems you met and solutions:

雖然去年修數位電路設計的時候就有用verilog寫過比較簡單的小東西，但是這次lab一開始要著手的時候還是很沒頭緒，畢竟對ALU還不是很熟，所以又上網查了有關ALU的介紹，才比較理解要做的內容。另外，verilog的語法也快忘光光了，所以又重新去看了這次作業會使用到的東西，最後才開始動手寫，算是前置準備作業做了比較久。寫完之後simulate跑出warning，但是因為還是可以硬run出結果，所以那部分我到現在還未解決，可能之後再利用助教時間去請問助教。跑測資的時候在slt這個operation一直出錯，也是debug了一陣子才跑出對的結果。

Summary:

因為我是用macbook，所以使用modelsim的話要另外灌windows系統，操作起來覺得有點不方便，加上我是接近死線才開始趕工的，所以整個作業過程好像有點趕，遇到問題也不能直接去找助教問，希望下一次作業出來時可以早點開始動手才不會真的做不出來，遇到問題也才可以面對面直接問助教解決。但是這次作業讓我對verilog熟悉了不少，雖然使用起來還是挺痛苦的，但希望以後的作業都可以順利一點。