

# Computer Organization, Fall, 2013

## Lab 1: 32bits ALU

**Due: 2013/10/14**

### 1. 作業目的

學習 Verilog 語言，實作出一個 32bits ALU(Arithmetic Logic Unit)，ALU 是 CPU 中運算的元件，例如加、減、AND、OR...等等運算皆由 ALU 來計算，會幫助你了解 CPU 架構，在之後的 Lab 中會一直用到 ALU 元件，可以沿用此 module，testbench 會自動將 input 檔讀入並且告知有幾個 case 沒過，請將解壓縮後的檔案放在同一個資料夾。

### 2. 作業要求

a. 請使用助教教學投影片裡提到的ModelSim PE Student Edition 10.2c作為平台。

b. 一人為一組，並在繳交的每個檔案開頭上以註解方式附上學號、姓名

上傳檔案為：學號.rar （檔名不要有中文）

c. 提供 TestBench modules。

**d. 嚴禁抄襲，被抓到一律 0 分計算。**

**e. Top module 的名稱和 IO ports 的命名請按照以下方式：**

Top module: alu.v

```
module alu (
    rst_n,           // negative reset (input)
    src1,            // 32 bits source 1 (input)
    src2,            // 32 bits source 2 (input)
    ALU_control,     // 4 bits ALU control input (input)
    result,          // 32 bits result (output)
    zero,            // 1 bit when the output is 0, zero must be set (output)
    cout,            // 1 bit carry out (output)
    overflow         // 1 bit overflow (output)
);
```

當 rst\_n 為 1 時 ALU 開始運作，

另外，為了培養良好的 coding style，請務必遵守以下幾點：

- 一個.v 檔內只有一個 module。
- module name 和檔案名稱必須相同。  
例如：檔案”alu.v”中只含有”module alu”;

f. 指令集: 基本的邏輯運算指令。(60%)

ALU action	Name	ALU control input
<b>And</b>	<b>And</b>	<b>0000</b>
<b>Or</b>	<b>Or</b>	<b>0001</b>
<b>Add</b>	<b>Addition</b>	<b>0010</b>
<b>Sub</b>	<b>Subtract</b>	<b>0110</b>
<b>Nor</b>	<b>Nor</b>	<b>1100</b>
<b>Nand</b>	<b>Nand</b>	<b>1101</b>
<b>Slt</b>	<b>Set less than</b>	<b>0111</b>

zcv 三條控制訊號: zero、carry out、overflow (30%)

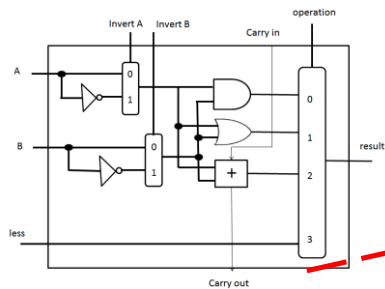
zero must be set when the result is 0

cout must be set when carry out

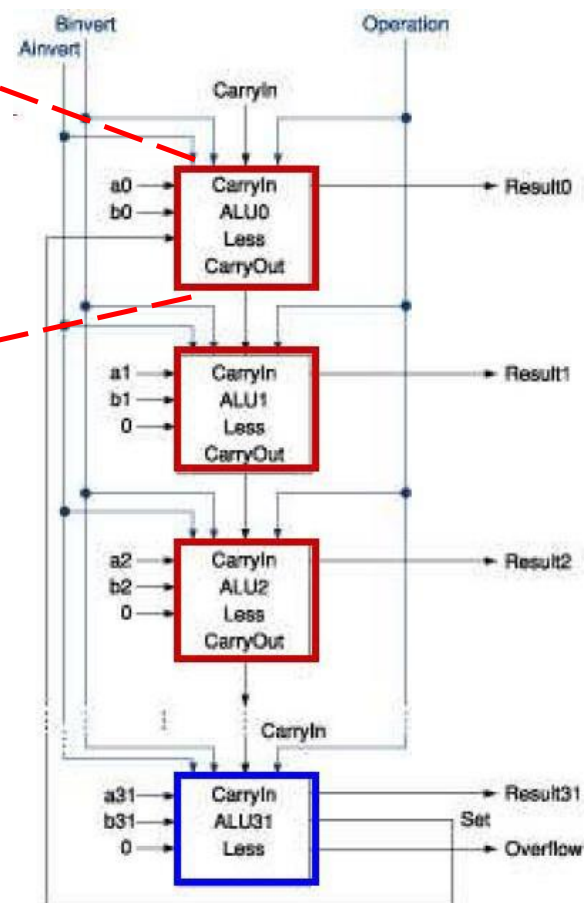
overflow must be set when overflow

### 3. 架構圖

1- bit ALU (Top)



32-bits ALU

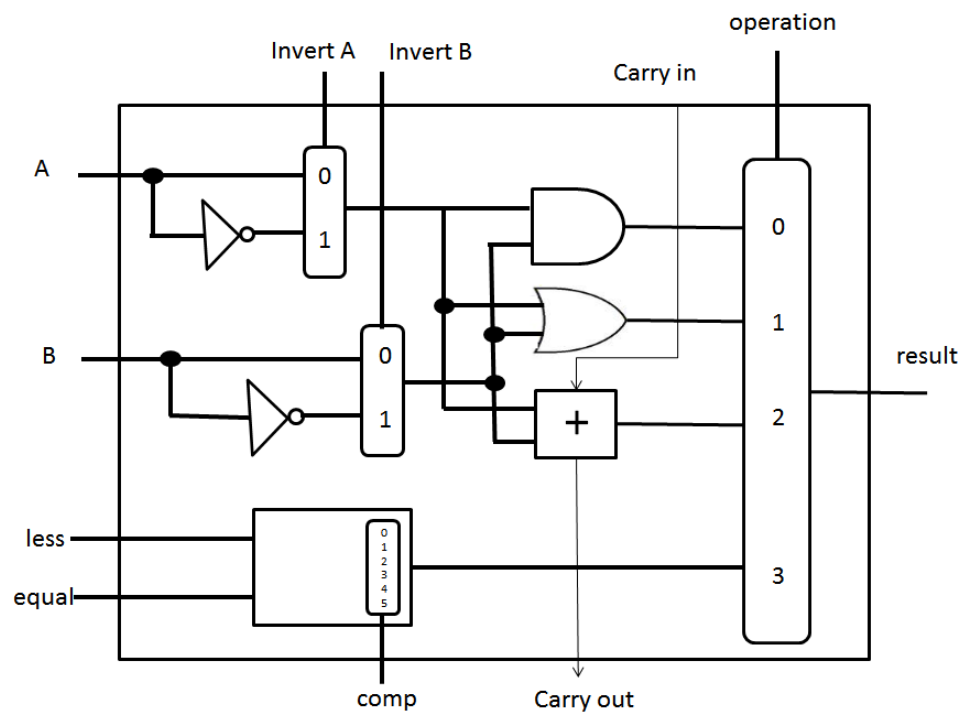


藍色方框為 1-bit ALU(Bottom) 與 1-bit ALU(Top) 不同

**4. Bonus:** 進階指令，視個人意願決定是否實作，成功地實作出來會有額外的獎勵分數。  
(10%)

ALU action	Name	ALU control input
Slt	Set less than	0111_000
Sgt	Set great than	0111_001
Sle	Set less equal	0111_010
Sge	Set great equal	0111_011
Seq	Set equal	0111_110
Sne	Set non equal	0111_100

Hint: 在 1-bit ALU 中加入一個 Compare 的 Module，需額外增加 3 bits 的 Compare\_sel 的 control input  
下圖為參考架構圖。



## 5. 評分方式

- 滿分: 110，抄襲一律 0 分!
- 報告: 10 分
- 每遲交一天作業分數打 9 折!

## 6. 繳交方式

E3 平台

將提供的測資.txt 檔與專案放於同一個資料夾，寫完後開始模擬視窗就會秀出程式是否 正確。  
(模擬方法請看 教學投影片)

```
# vsim -voptargs=+acc work.testbench
# Loading work.testbench
# Loading work.alu
# Loading work.bit_alu_top
# Loading work.bit_alu_bottom
VSI110> run -all
# *****
# No.12 error!
# Current result: 01111100    Current ZCV: 000
# Your result: 7fffffff    Your ZCV: 000
# *****
# Break in Module testbench at C:/Users/kayeoc279/Downloads/9817262/aluu/testbench.v line 130
VSI111>
```

上圖表示部分功能錯誤

下圖為全過的訊息

```
# *****
# Congratulation! All data are correct!
# *****
# Break in Module testbench at C:/Users/kayeoc279/Downloads/9817262/aluu/testbench.v line 130
VSI111>
```