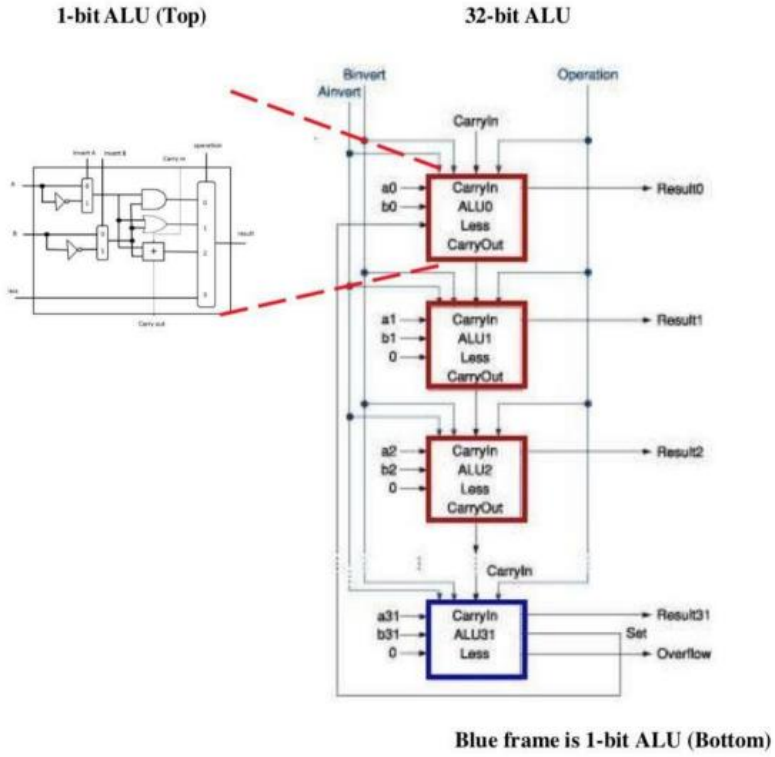
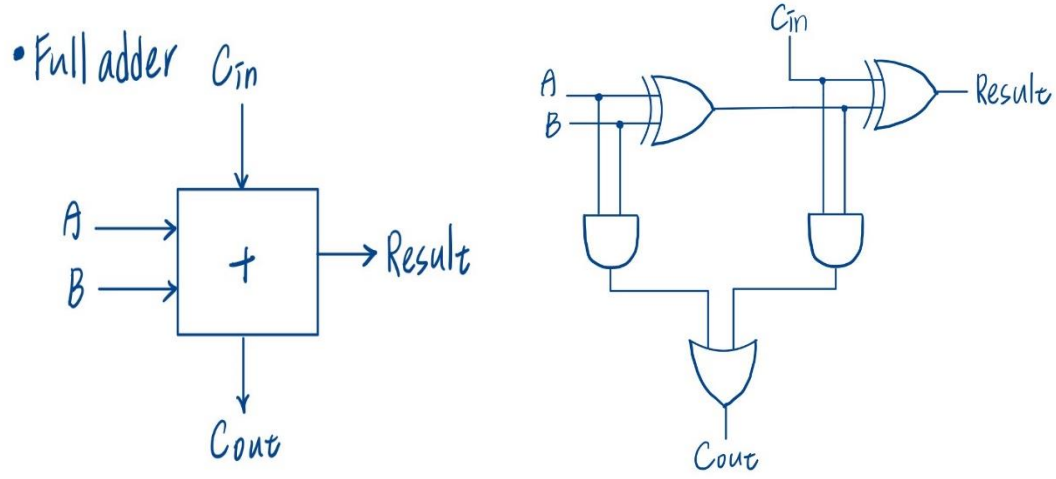
Computer Organization

Architecture diagrams:



紅色部分使用 alu_top module

藍色部分使用 alu_base module



實作此 32-bit ALU，將每一 bit 分別使用 1-bit ALU 處理。

1-bit ALU 包含 AND、OR、ADD、SLT operation，整合成 32-bit ALU 後運用 ADD 延伸出 SUB operation，AND 延伸出 NOR operation。

總共使用了 31 個 alu_top module、1 個 alu_base module。

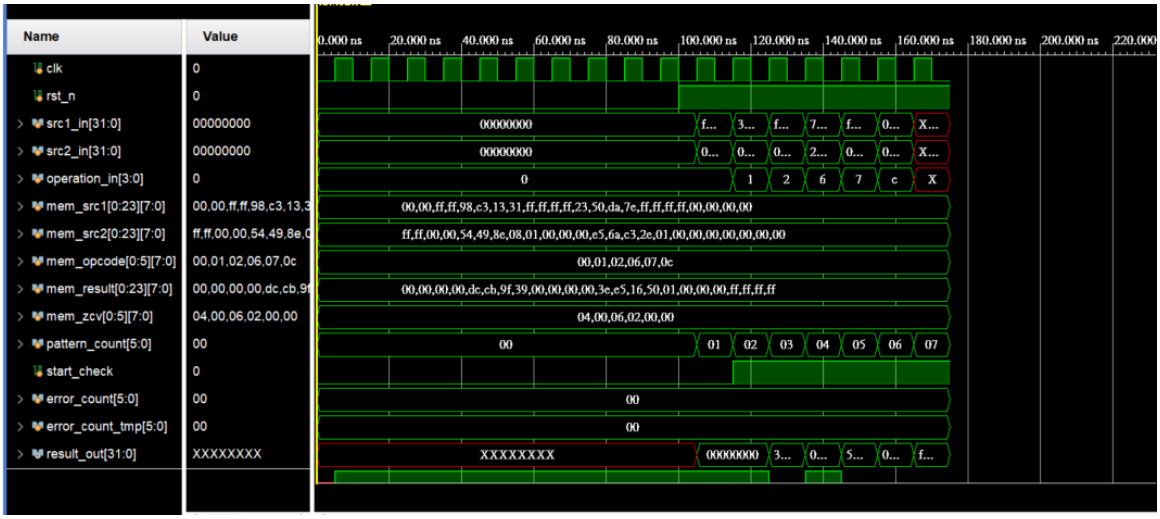
Hardware module analysis:

Module alu_top 用於處理除了 MSB 的其餘 31 個 bit、
Module alu_base 用於處理 MSB、
Module alu 用於將參數運算結果傳入 alu_top 和 alu_base 模組。

Experiment result:

```
**************************************************
 Congratulation! All data are correct!
**************************************************
```

有成功輸出正確訊息，波形也符合預期



Problems you met and solutions:

1.
在 code 方面，首先是 reg 和 wire 的設置，在使用 wire 時才可以用 assign，因為忽略了這點我在 run simulation 時一直接收到 ”concurrent assignment to a non-net 'result' is not permitted” 這樣的報錯。
2.
在觀念理解上面，carryout 和 overflow 使我搞混，最後理解 carryout 是使用於 unsigned，而 overflow 是使用於 signed。
3.
在使用 vivado 執行程式上面我遇到我最大的困難，常常在重啟程式之後要點開 source 的.v 檔，又或者只是在 add source file，vivado 會直接跑很久然後就卡在那邊。關於解決辦法，我透過網路也無從得知如何從根本解決這個問題，而我只能不斷嘗試將電腦重開機又或者不斷重新 create project.
(ps.請問助教知道這類問題如何解決嗎?)

Summary:

經過這次 lab 我更了解 ALU 的細節運作是如何進行的，verilog 硬體語言對我而言屬於較少接觸的語言，透過實作部分需要上網搜尋相關語法，我對 verilog 也更熟悉了。