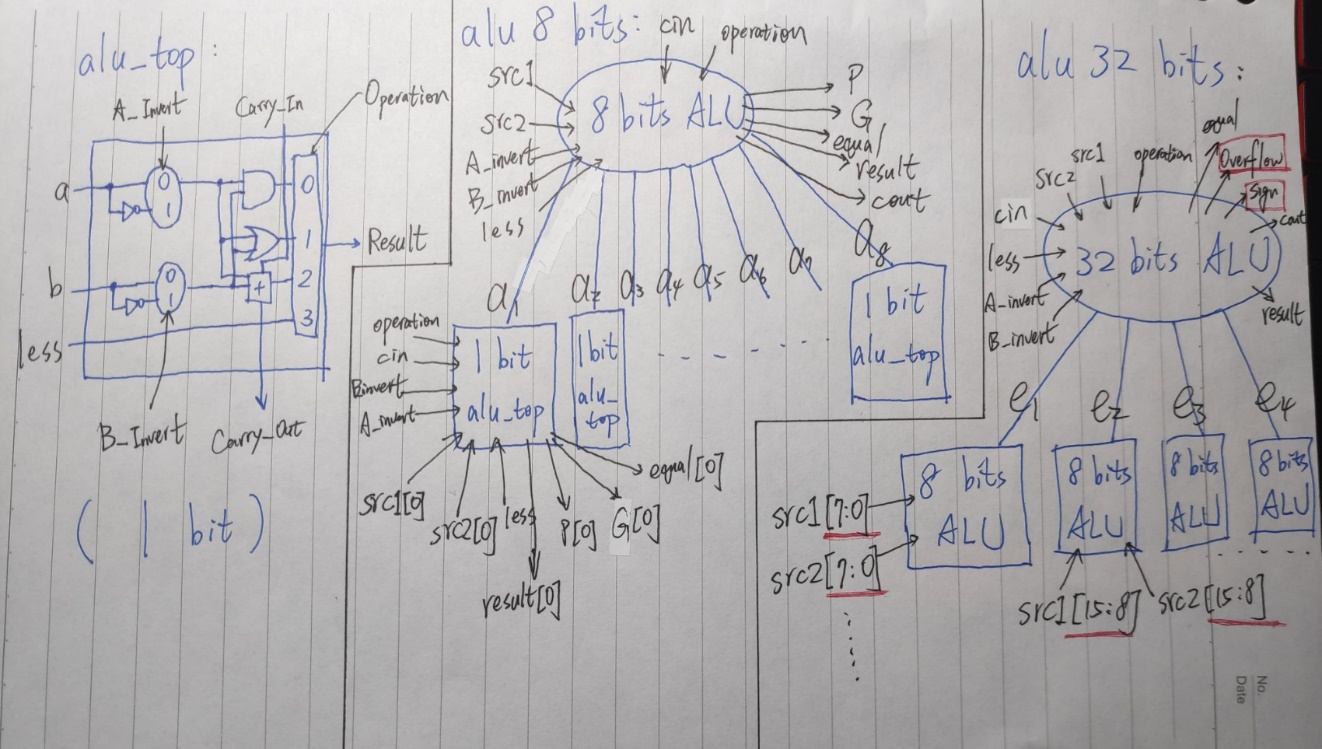
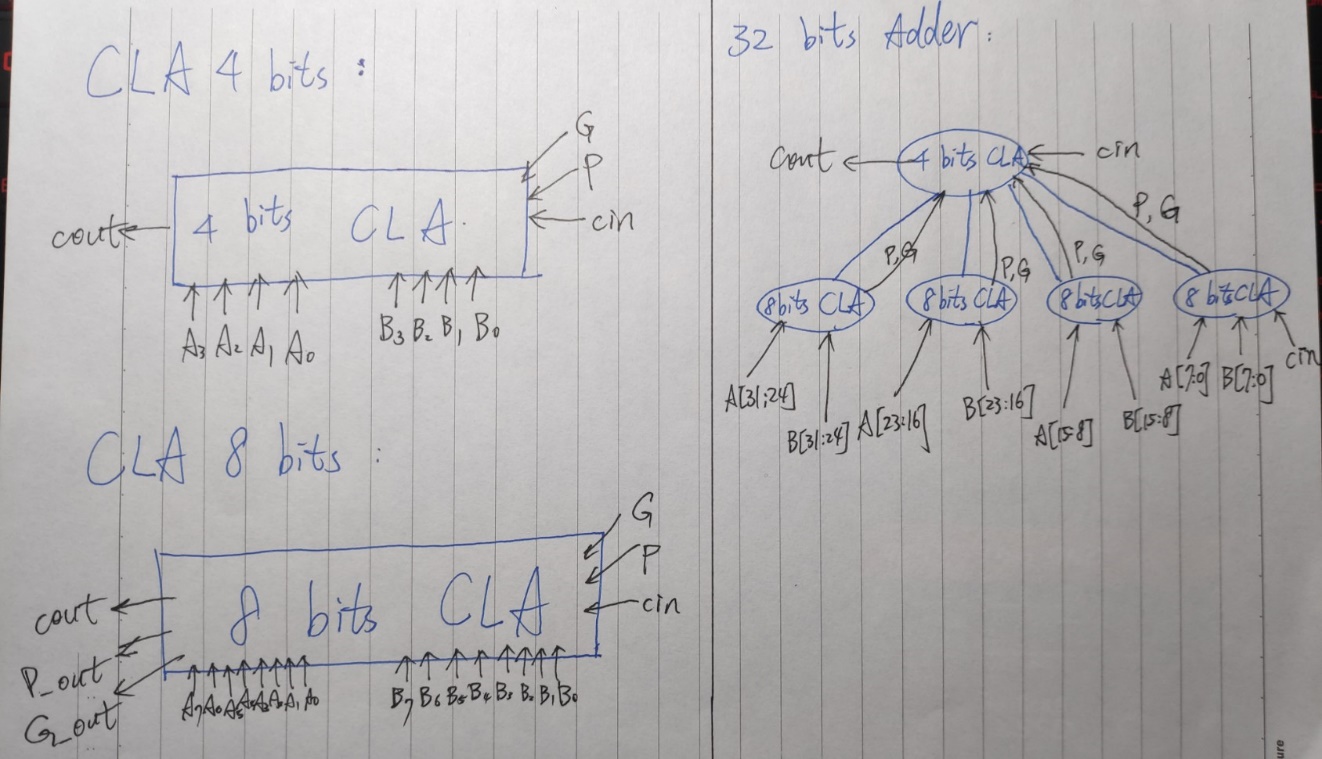
Term Project - Part 1 (Lab 1) Report

1. architecture diagram





1. Detailed description of the implementation

這次ALU要實作And Or Add Sub Nor Nand SLT等七個功能，Bonus則要多做SGT SLE SGE SEQ SNE共12個功能。And和Or屬於比較好實作的部分，而Nand和Nor就是And和Or的相反，透過一個invert就可以完成。而Add做完後，Sub就是把減數換成2補數再相加，因此Sub能透過Add完成。最後SLT就是兩數相減，其結果大於0就是前數大，反之就是後數大。大於或小於0可直接由Sign bit判斷。

加法器Add的部分屬於比較複雜的地方。我們可以做32個1bit ALU再把carry傳給下一個ALU，做成32bits的ripple adder。但這樣實在太沒效率。我們應該carry look ahead的方式同步算出carry，這可以利用公式來取得。但32bits的公式實在太長，寫起來相當不易，而且電路實作出來也會相當複雜、消耗許多成本。因此，我們應該結合上述兩種概念。先把32個bits拆成4個8bits，然後實作8bits CLA。再將4組算出來的結果分別丟進4bits CLA，就能算出答案，如第一部分的下圖所示。

最後Bonus的部分，要先利用Compare來決定出要執行SLT SGT SLE SGE SEQ SNE這六個裡面的哪一個operation，接著用equal結合slt的原理就能實作出六個operation。

另外，我把alu.v拆開成alu\_basic.v和alu\_bonus.v，以方便區別及執行，兩檔案前半部分皆與原本給的alu.v一樣，我有用 //origin above標示。執行Bonus時也要先把testbench.v中的 //`define BONUS的註解拿掉。

1. Commands for executing

Basic data:

先把basic\_data裡的七個檔案(.txt)移動到和.v檔相同的目錄下，再接著輸入以下指令

iverilog -o alu\_basic.vvp alu\_basic.v alu\_top.v alu8bits.v alu32bits.v CLA4bits.v CLA8bits.v Compare.v testbench.v

vvp alu\_basic.vvp

Bonus data:

先把原本basic\_data的txt檔刪除，換成把bonus\_data的七個txt檔移動到相同的目錄下，再執行以下指令

iverilog -o alu\_bonus.vvp alu\_bonus.v alu\_top.v alu8bits.v alu32bits.v CLA4bits.v CLA8bits.v Compare.v testbench.v

vvp alu\_bonus.vvp

Results:

