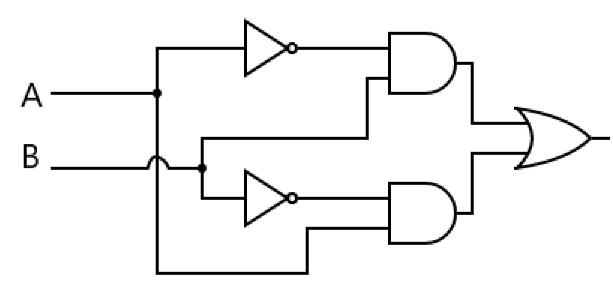
|  |
| --- |
| **EECS 2070 02 Digital Design Labs 2019**  **Lab 1** |
| **學號：107062314 姓名：陳柏均** |

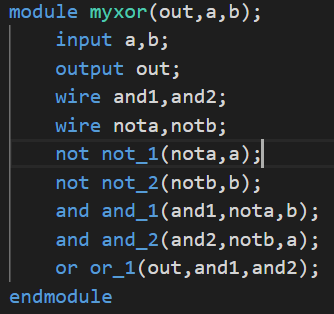
1. 實作過程

這次的lab主要要我們實作ALU的問題,我覺得這次lab不只幫我複習了verilog的主要語法, 更是幫助我訓練嚴謹的邏輯思考。前三題的testbench皆相同, 但卻分別用了gate level, dataflow level以及behavior level, 由於有一段時間沒又去碰verilog了, 一開始有一點生疏, 但在經過認真思考和查資料後, 慢慢地才拾回打code的熟悉感。

第零題我覺得真的出的很好, 畢竟後面要用到, 很感謝有這題並且提供了testbench, 雖說這題很簡單也不是主要要學習的部分, 但我認為這題很關鍵地讓我熟悉了gate level 的用法以及如何看著圖拼出一個自己想要的module, 我相信邏輯閘的用法和連結在硬體中是不可或缺的部分,而這題雖是基礎,但卻是十分重要且必要的。

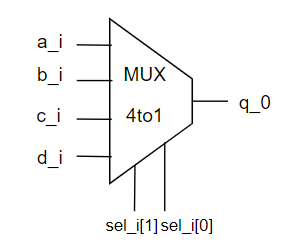


(註:第零題我選擇了用兩個not,兩個and及一個or拼出xor)

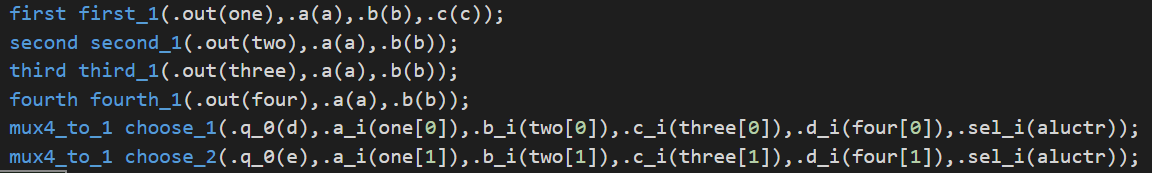


(註:用and,or,not所拼出的myxor)

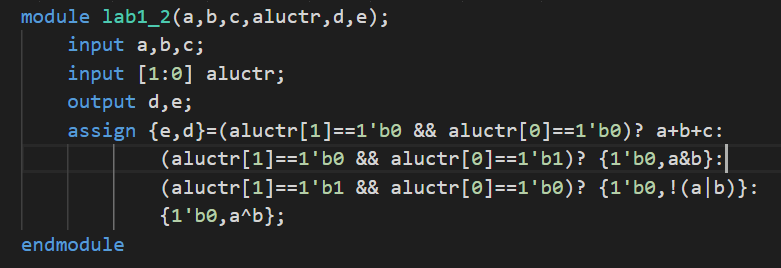
第一題我用了四個module去計算然後在主要的module(lab1\_1)用「one, two, three, four」這幾個wire去記住四種題目條件分別算出來的答案, 然後aluctr再去選出應該給的答案。而在選的部分,我便是利用了助教所給的mux4\_to\_1去做選擇, 如下圖。



(註:mux4\_to\_1示意圖)

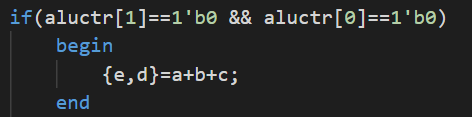
而由於q\_0只有一個bit, 因此我用了兩次的mux4\_to\_1分別去判斷及給d,e值; 另外在這一題的加法中我也使用了之前在邏輯設計課所學的fulladder去進行相加的動作。

(註:用MUX去選擇的程式)

第二題我自己認為比第一題直覺多了, 我用了assign 並且使用verilog 中 「?:」的功能去直接判斷並給值, 我認為這樣十分地像在打C語言, 對於我個人而言, 軟體還是比硬體好打, 軟體比較像用人類的語言去命令電腦去做甚麼, 而硬體語言則比較偏向人類去用電腦的方式思考, 但此題我就有一種回到打軟體語言的感覺, 不用想太多, 直接做自己想要編譯器去跑的程式,如下。

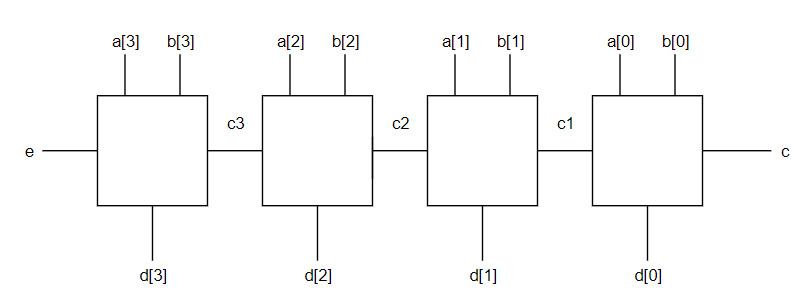
(註:直接使用 ?: 去判斷及給值)

第三題我用了always block 並且在之中使用了if else去給值, 跟上一題的解法差不多, 主要差別是在always裡面我才可以使用if else 這一個功能,

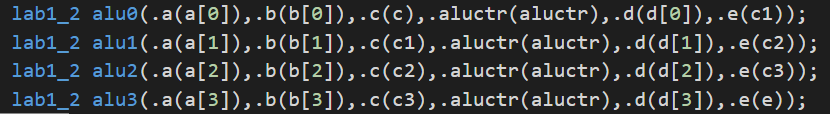


(註:lab1\_4跟上一題差不多,只有差在always及if else 的使用方式)

第四題的題目有稍做修正,跟前三題明顯不同, 但只要仔細的思考,便可以想出此題要使用串接的方式去做4個 bits的題目, 就舉加法而言第一個結果的cout是第二個的cin而以此類推。

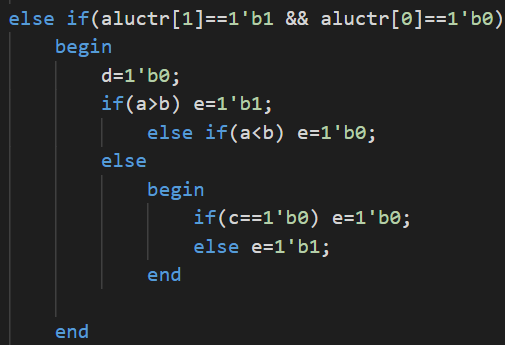


(註:lab1\_4想法示意圖)



(註:使用lab1\_2進行指令的動作及運算,而這個module只負責做串接)

Bonus1一開始我看了很久搞不清楚要幹嘛, 後來才發現只要將之前的code稍微改一下並加上一個判斷式去給值就好, 而bonus2則是依照lab1\_4依樣的方法去做。



(註解:bonus1 使用lab1\_3依題目要求稍做修改)

1. 學到的東西與遇到的困難

(i)reg使用的問題

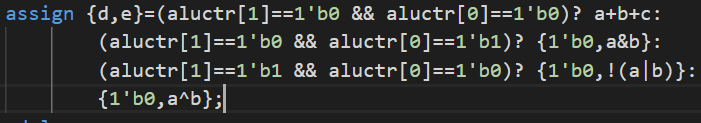
在lab1\_3中,我一度忘記在always block 之中左邊的值必須要為reg, 而產生了不該犯的錯誤, reg 跟wire的用法無疑是verilog 之中一個很重要的觀念,而經過數次的犯錯,希望下次能更快發現錯誤或是說不會再遇到這樣的語法不正確。

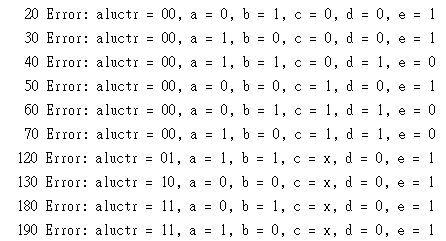
C:\Users\t2378\Desktop\邏輯設計\lab1\圖片\reg_error.png

(註:沒有使用reg導致的錯誤)

(ii)給值的錯誤

在寫第二題的時候,本來以為會很快就過了或者是很少錯誤,沒想到丟上Vivado編譯後竟然跑出了一大堆errors, 我從testbench給的值看了許久還是不懂自己為何而錯。到後來當我再去看一次題目的時候才發現自己d,e的值給相反了, 原本以為e是比較低的那一位,結果是相反的。另外在這裡我也學到了大括號的使用技巧, 可以寫成 {d,e} 然後去給值,十分的方便。

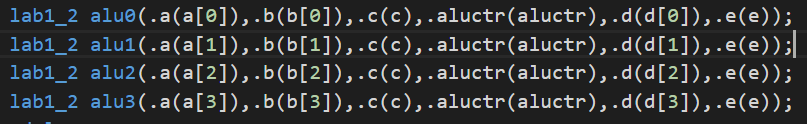


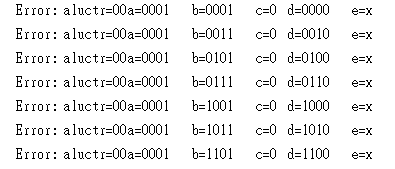


(註:d,e給值錯誤導致出現諸多errors)

(iii)串接的問題

剛開始讀第四題的題目的時候,我其實沒有很懂題目想要表達的意思, 還以為是一個一個bit做並且去給output e值,後來怎麼想都不合理, 思考了許久後才了解到此題要我所做的事; 同時也發現自己原本想法的荒謬,原本我的做法竟然是把它當作四個獨立的alu去給值,而非將其串接起來。





(註: 一開始錯誤的思考方向成了一個e=x的結果)

1. 想對老師或助教說的話

個人一直以來其實對硬體語言都不太熟悉, 尤其邏輯設計的課剛開始打verilog code的時候,每次看到題目後,我幾乎都是很慌張的狀態。對於剛學完軟體程式的那時候的我而言,這個語言真的是難上加難,常常我重複閱讀了指示卻仍坐在電腦前面, 不知從何下手。對於我而言,軟體真的比較直覺且熟悉多了,我可以想到什麼就打什麼,不用以機器的角度去思考。

但同時我不能反對的是熟能生巧,之前打了數次的lab, 自己學習能力真的比較差, 很多次都是不斷地詢問助教及同學才能完成的。而這次卻比較不一樣,我發現其實我認真的去思考還是能打出一個成果的,但我相信這當然也是之前所累積下來的結果。

這次的lab1我複習了不少verilog的應用技巧及學習的一些新的用法,這次的難度對我個人而言是剛剛好的, 其實也花了滿多時間在debug的。這次的題目對大部分人來講應該偏簡單, 我還有很大的進步空間, 相信以後一定會有更佳有挑戰性的題目, 即使難免會擔心打不出來,但我知道這就是學習的過程;沒有犯錯,豈能讓下次的的自己變更好?

我認為這次的出題方式十分地清晰有條理,我很喜歡,幫我複習了許多東西,同時也讓我更加地運用上課所學。未來無可避免的會有許多瓶頸,但我同時也希望透過這些困難及挑戰獲得一定的收穫,讓自己不斷地突破自我。