EECS 2070 02 Digital Design Labs 2019

Lab 1

學號：107000115　姓名：林珈卉

1) 實作過程

   a) Module myxor

      i) 概念

         (1) 使用 Gate-Level 組合電路，使得將訊號 a 及 b 輸入後，可以得到 a XOR b 的 output

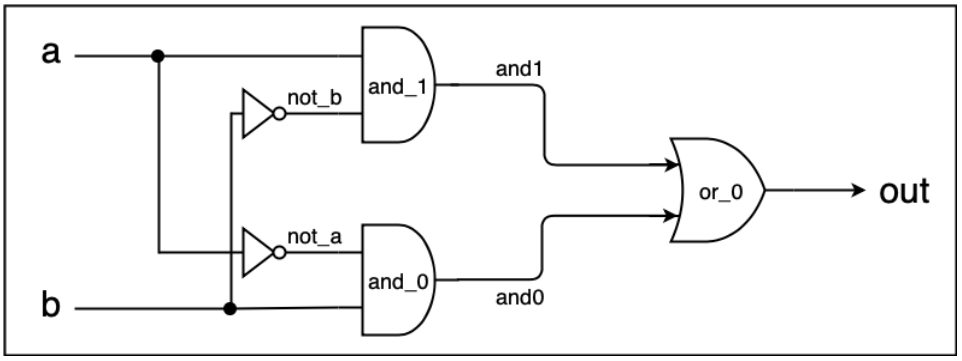
         (2) 只能使用 AND gate、OR gate 及 NOT gate

         (3) 使用 Design 1

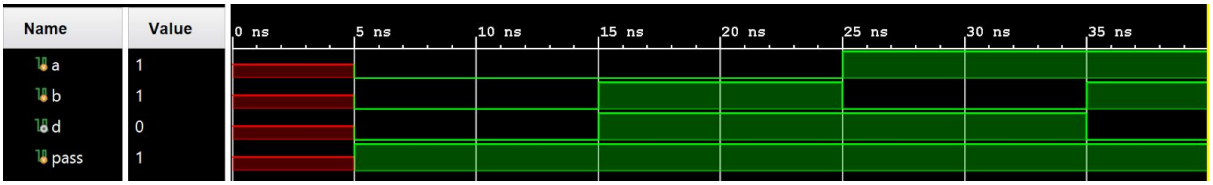
      ii) Boolean Expression

      out = (a & b') | (a' & b)

      iii) Logic Diagram



      iv) Waveform



b) Module lab1_1

i) 概念

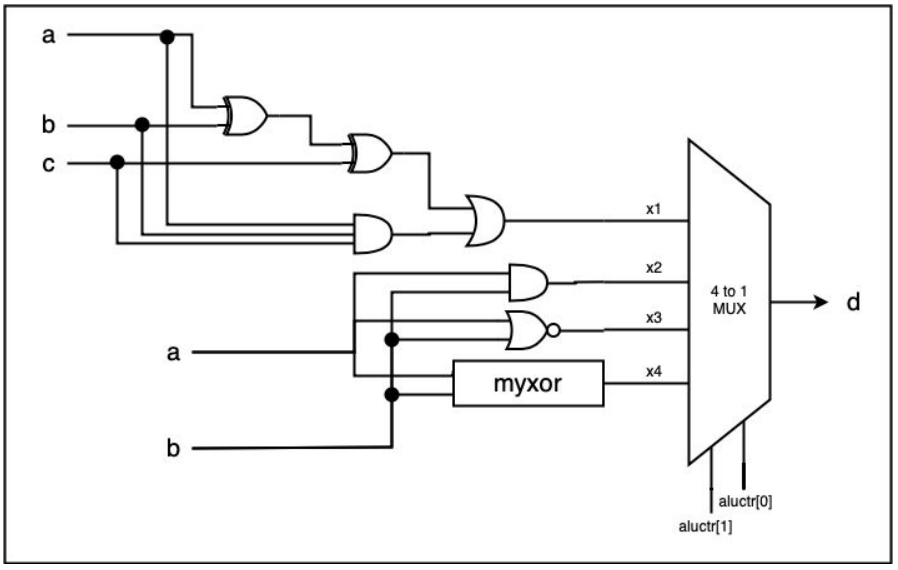
(1) 在 aluctr (input signal) 數值的不同狀況下，執行四種 function，以 AND, OR, NOT Gate 組合電路

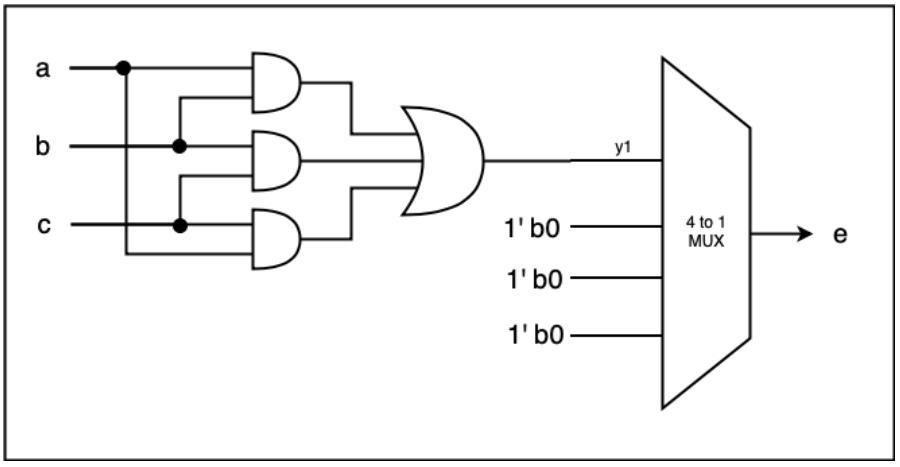
aluctr[1]	aluctr[0]	fucntion
0	0	{e,d} = a + b + c
0	1	d = a and b e = 0
1	0	d = a nor b e = 0
1	1	d = a xor b e = 0

(2) Input Signal : a, b, c, [1:0] aluctr

Output Signal : d, e

ii) Block Diagram





iii) K-map of function {e, d} = a + b + c

(1) Output signal e

a \ bc	00	01	11	10
0	0	0	1	0
1	0	1	1	1

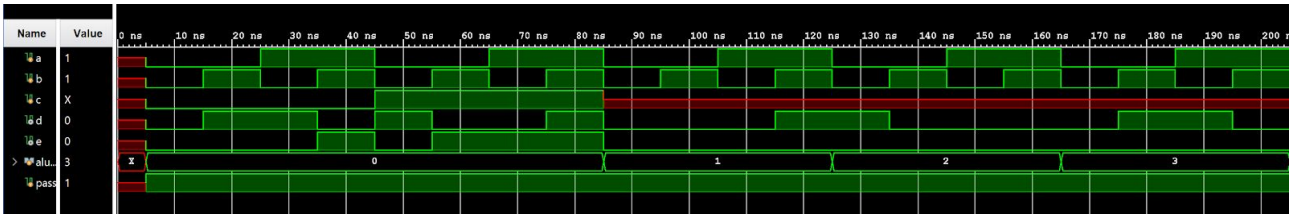
e = (a & b) | (b & c) | (a & c)

(2) Output signal d

a \ bc	00	01	11	10
0	0	1	0	1
1	1	0	1	0

d = ((a XOR b) XOR c) | (a AND b AND c)

iv) Waveform



c) Module lab1_2

i) 概念

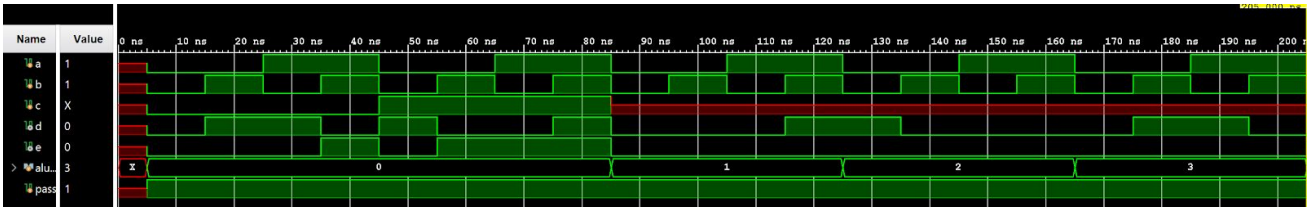
(1) 以 Data Flow Modeling 表示出上述的 4 個 function

(2) Expressions of outputs d and e

d = !aluctr[0] & !aluctr[1] & (!a&!b&c | a&b&c | a&!b&!c | !a&b&!c) |
!aluctr[1] & aluctr[0] & a & b |
aluctr[0] & aluctr[1] & (a&!b | !a&b) |
aluctr[1] & !aluctr[0] & !a & !b

e = !aluctr[0] & !aluctr[1] & (a&b | a&c | b&c)

ii) Waveform



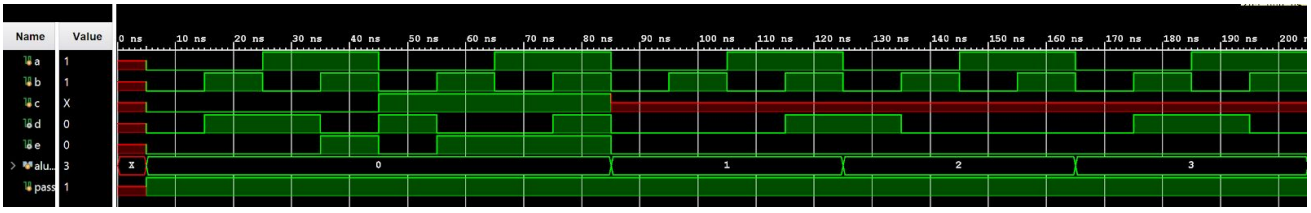
d) Module lab1_3

i) 概念

(1) 以 Behavioral Modeling 表示出上述的 4 個 function

(2) 在 always block 中，使用case (aluctr)，分為 0, 1, 2, 3

ii) Waveform



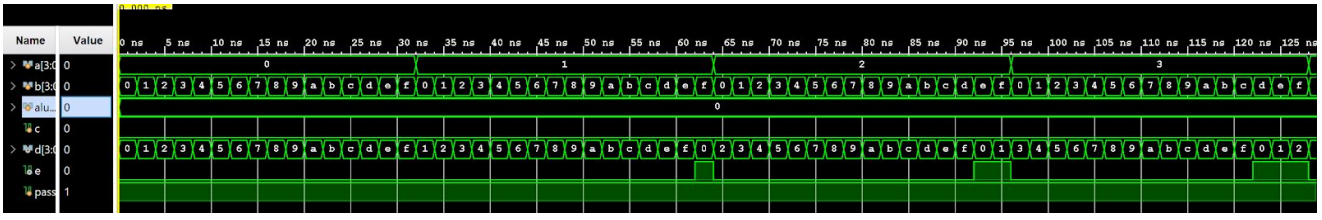
e) Module lab1_4

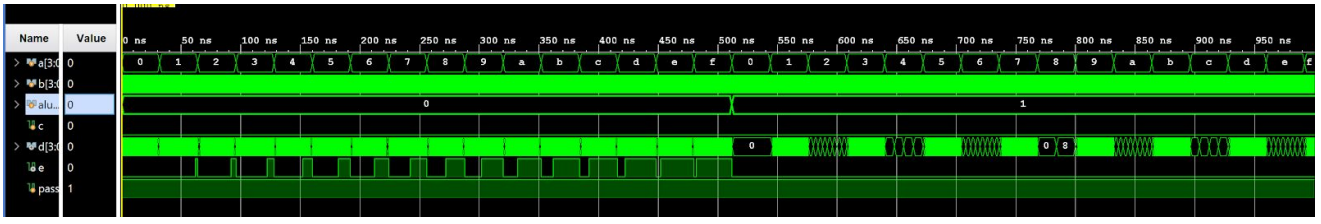
i) 概念

(1) 使用 lab1_1, lab1_2, lab1_3 其中一種 1-bit ALU 來組合成 4-bit ALU

(2) 將每個 bit 的 output signal e，宣告一個 wire 變數連接至下一個 bit 的 input signal c

ii) Waveform





f) Module lab1_b1

i) 概念

(1) 實作出 Compare Function

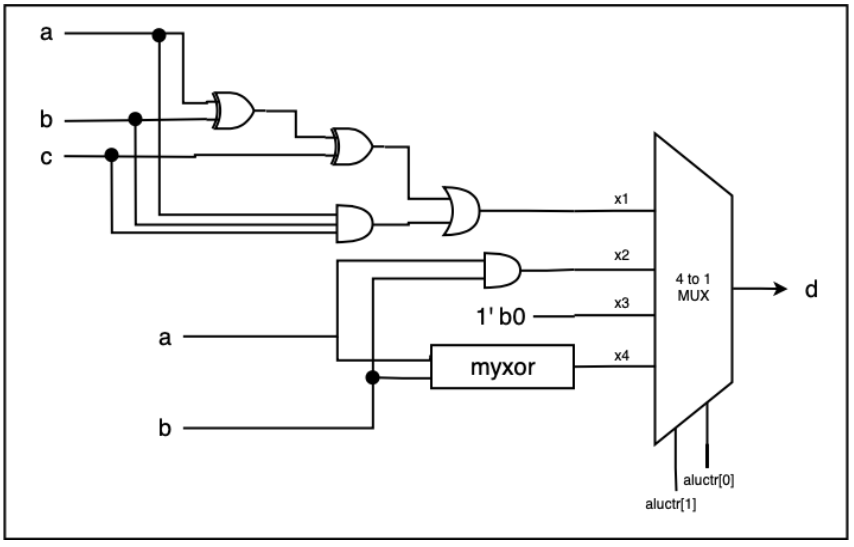
(2) input signal 為 a, b, c

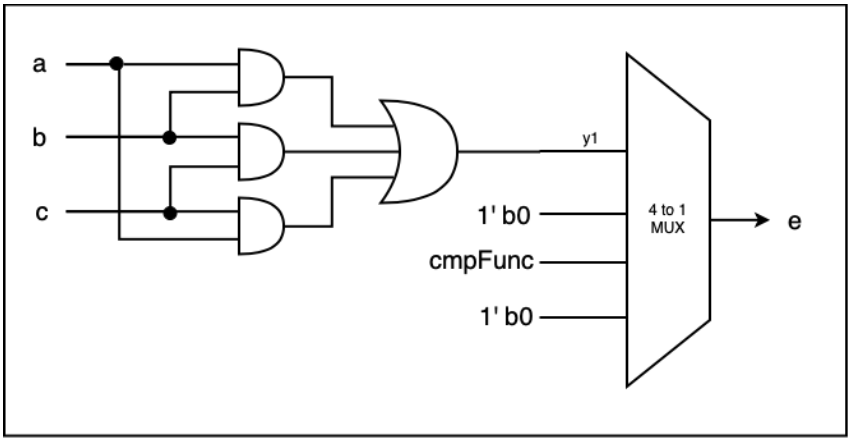
(3) 使用 if else 判斷式

(4) 將輸出結果替換掉 e 的 MUX 的第三個 input

aluctr[1]	aluctr[0]	fucntion
0	0	{e,d} = a + b + c
0	1	d = a and b e = 0
1	0	d = 0 e = cmpFunc(a,b)
1	1	d = a xor b e = 0

ii) Block Diagram





g) Module lab1_b2

i) 概念

(1) 使用 lab1_b1 的 1-bit ALU 來組合成 4-bit ALU

(2) 將每個 bit 的 output signal e，宣告一個 wire 變數連接至下一個 bit 的 input signal c

2) 學到的東西與遇到的困難

a) 在實作 lab1_3 時學到，在 always block 中 assign 值的變數必須為 reg,

b) 學會如何在 Verilog 中寫 Function

原本以為寫 Function 時要像寫 Module 一樣，列出所有的 input 及 output，但其實只要把 input 列出，function 的 output 只有一個，也就是 function 本身，直接將要輸出的值送給 function 的名稱即可

c) 當 MUX 其中有 input 的值恆為定值時，不需要宣告一個 wire 接上去，只要直接將數值寫在括號內

3) 想對老師或助教說的話

a) 希望老師可以提供歷屆學長姐 report 的模範範本給我們可以參考，以了解如何掌握硬體程式設計的重點。

b) 在上機 Demo 時，助教是否能在 Demo 前先告訴學生們要開哪些畫面，才能加快 Demo 的速度。