

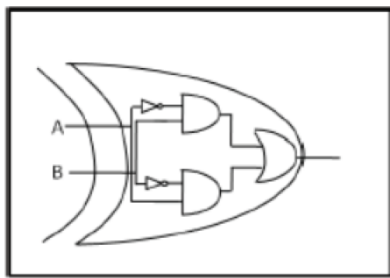
EECS 2070 02 Digital Design Labs 2019

Lab 1

學號：104021215 姓名：熊磊

1. 實作過程

- 首先要實作 myxor，我使用助教提供的設計用兩個 AND Gates 和兩個 NOT Gates，和一個 OR Gate，拚出一個 XOR Gate。



```

23 module myxor(out,a,b);
24     input a, b;
25     output out;
26
27     wire not_a, not_b, and_1, and_2;
28
29     not not0(not_a, a);
30     not not1(not_b, b);
31     and and1(and_1, not_a, b);
32     and and0(and_2, a, not_b);
33
34     or or0(out, and_1, and_2);
35 endmodule

```

- 接著是實作 lab1_1，這個要求只能使用 AND、OR、NOT、myxor、mux4_to_1 等來實作，因此我的作法是寫 4 個 my_add、my_and、my_nor、my_xor 的 module，裡面分別都是用 gate level description 來描述，最後再從 lab1_1 用 mux4_to_1 來選擇 d、e 的值。

```

61 module lab1_1 (a, b, c, aluctr, d, e);
62     input a, b, c;
63     input [1:0] aluctr;
64     output d, e;
65
66     wire d0,d1,d2,d3,e0,e1,e2,e3;
67
68     my_add m0(e0, d0, a, b, c);
69     my_and m1(e1, d1, a, b, c);
70     my_nor m2(e2, d2, a, b, c);
71     my_xor m3(e3, d3, a, b, c);
72
73     mux4_to_1 mux0(d, d0, d1, d2, d3, aluctr);
74     mux4_to_1 mux1(e, e0, e1, e2, e3, aluctr);
75
76 endmodule

```

- lab1_2、lab1_3 因為可以用 data flow modeling 和 behavioral modeling，因此比較簡單，可以直接 assign 值或是用 always block 來 implement。

```

22 module lab1_2 (a, b, c, aluctr, d, e);
23     input a, b, c;
24     input [1:0] aluctr;
25     output d, e;
26
27     assign d = (aluctr == 2'b00) ? a^(b^c) :
28               (aluctr == 2'b01) ? a&b :
29               (aluctr == 2'b10) ? !(a|b) : a^b;
30     assign e = (aluctr == 2'b00) ? (a&b)|(b&c)|(a&c) : 0;
31
32 endmodule

```

```

22 module lab1_3 (a, b, c, aluctr, d, e);
23     input a, b, c;
24     input [1:0] aluctr;
25     output reg d, e;
26
27     always@(*)begin
28         case(aluctr)
29             2'b00:begin
30                 {e, d} = a + b + c;
31             end
32             2'b01:begin
33                 d = a&b;
34                 e = 0;
35             end
36             2'b10:begin
37                 d = !(a|b);
38                 e = 0;
39             end
40             2'b11:begin
41                 d = a^b;
42                 e = 0;
43             end
44         endcase
45     end
46 endmodule

```

- lab1_4 因為要使用 1 bit 來組成 4 bits 的 adder、AND、NOR、XOR，我先在 lab1_4 的外面先做 4 個 four_bit_adder、four_bit_and、four_bit_nor、four_bit_xor，裡面都是 instantiate lab1_3，其中 adder 因為會有 carry out，所以我們要從 LSB 加到 MSB，把前一位的 Carry Out 放到下一位的 input。

| | |
|---|---|
| <pre> 22 module four_bit_adder (d, e, a, b, c); 23 input [3:0] a,b; 24 input c; 25 output [3:0] d; 26 output e; 27 28 wire t1, t2, t3; 29 lab1_3 o1(a[0], b[0], c, 2'b00, d[0], t1); 30 lab1_3 o2(a[1], b[1], t1, 2'b00, d[1], t2); 31 lab1_3 o3(a[2], b[2], t2, 2'b00, d[2], t3); 32 lab1_3 o4(a[3], b[3], t3, 2'b00, d[3], e); 33 endmodule 34 35 module four_bit_and (d, e, a, b, c); 36 input [3:0] a,b; 37 input c; 38 output [3:0] d; 39 output e; 40 41 lab1_3 o1(a[0], b[0], c, 2'b01, d[0], e); 42 lab1_3 o2(a[1], b[1], c, 2'b01, d[1], e); 43 lab1_3 o3(a[2], b[2], c, 2'b01, d[2], e); 44 lab1_3 o4(a[3], b[3], c, 2'b01, d[3], e); 45 endmodule </pre> | <pre> 47 module four_bit_nor (d, e, a, b, c); 48 input [3:0] a,b; 49 input c; 50 output [3:0] d; 51 output e; 52 53 lab1_3 o1(a[0], b[0], c, 2'b10, d[0], e); 54 lab1_3 o2(a[1], b[1], c, 2'b10, d[1], e); 55 lab1_3 o3(a[2], b[2], c, 2'b10, d[2], e); 56 lab1_3 o4(a[3], b[3], c, 2'b10, d[3], e); 57 endmodule 58 59 module four_bit_xor (d, e, a, b, c); 60 input [3:0] a,b; 61 input c; 62 output [3:0] d; 63 output e; 64 65 lab1_3 o1(a[0], b[0], c, 2'b11, d[0], e); 66 lab1_3 o2(a[1], b[1], c, 2'b11, d[1], e); 67 lab1_3 o3(a[2], b[2], c, 2'b11, d[2], e); 68 lab1_3 o4(a[3], b[3], c, 2'b11, d[3], e); 69 endmodule </pre> |
|---|---|

接著我用 always block 來作給值

| | |
|---|--|
| <pre> 71 module lab1_4 (a, b, c, aluctr, d, e); 72 input [3:0] a,b; 73 input [1:0] aluctr; 74 input c; 75 output reg [3:0] d; 76 output reg e; 77 78 wire [3:0] d0,d1,d2,d3; 79 wire e0,e1,e2,e3; 80 81 four_bit_adder m0(d0, e0, a, b, c); 82 four_bit_and m1(d1, e1, a, b, c); 83 four_bit_nor m2(d2, e2, a, b, c); 84 four_bit_xor m3(d3, e3, a, b, c); 85 86 always@(*)begin 87 case(aluctr) 88 2'b00:begin 89 e = e0; </pre> | <pre> 90 d = d0; 91 end 92 2'b01:begin 93 e = e1; 94 d = d1; 95 end 96 2'b10:begin 97 e = e2; 98 d = d2; 99 end 100 2'b11:begin 101 e = e3; 102 d = d3; 103 end 104 endcase 105 end 106 endmodule </pre> |
|---|--|

- Bonus1 的部分是要把其中一個 function 改成 cmpFunc，為了方便 lab1_b2 可以直接使用這個 module，因此我在這個 module 中使用 parameter，當作 input a, b 的 bit 數。後面將 comp_e 這個 wire 接在 always block 中，當 alucr 是 10 時的 e。而 comp_e 如 Spec 所提要求。

```
compareFunc(a, b, c):
  if a > b:
    e = 1
  else if a < b:
    e = 0
  else if a == b:
    if c == 0: e = 0
    if c == 1: e = 1
```

```
22 module lab1_b1 (a, b, c, alucr, d, e);
23     parameter n = 1;
24     input [n-1:0] a, b;
25     input c;
26     input [1:0] alucr;
27     output reg [n-1:0] d;
28     output reg e;
29
30     wire comp_e;
31     assign comp_e = (a>b)? 1 : (a<b)? 0 : (c == 0) ? 0 : 1;
32
33     always@(*)begin
34         case(alucr)
35             2'b00:begin
36                 {e, d} = a + b + c;
37             end
38             2'b01:begin
39                 d = a*b;
40                 e = 0;
41             end
42             2'b10:begin
43                 d = 0;
44                 e = comp_e;
45             end
46             2'b11:begin
47                 d = a*b;
48                 e = 0;
49             end
50         endcase
51     end
52 endmodule
```

- 因為 lab1_b1 已經設計好對於 n bits 都適用的 compFunc，因此我在 lab1_b2 中直接 instantiate lab1_b1。

```
22 module lab1_b2 (a, b, c, alucr, d, e);
23     input [3:0] a, b;
24     input [1:0] alucr;
25     input c;
26     output reg [3:0] d;
27     output reg e;
28
29     wire[3:0] d0,d1,d2,d3;
30     wire e0,e1,e2,e3;
31
32     lab1_b1 #(4) m0(a, b, c, alucr, d0, e0);
33     lab1_b1 #(4) m1(a, b, e0, alucr, d1, e1);
34     lab1_b1 #(4) m2(a, b, e1, alucr, d2, e2);
35     lab1_b1 #(4) m3(a, b, e2, alucr, d3, e3);
36
37     always@(*)begin
38         case(alucr)
39             2'b00:begin
40                 e = e0;
41                 d = d0;
42             end
43             2'b01:begin
44                 e = e1;
45                 d = d1;
46             end
47             2'b10:begin
48                 e = e2;
49                 d = d2;
50             end
51             2'b11:begin
52                 e = e3;
53                 d = d3;
54             end
55         endcase
56     end
57
58 endmodule
```

2. 學到的東西與遇到的困難

第一次 Lab 剛開始，所以還蠻簡單的，遇到的困難就是看到 spec 中有一句話「The c represents the result of the previous bit. If there is no previous bit, c is 0.」有點看不太懂是什麼意思，後來 Demo 時有問助教，所以弄清楚了！

3. 想對老師或助教說的話

謝謝助教/老師花時間幫我們 Demo，第一次 Report 不知道要寫什麼，想問助教有沒有期末 project 的 ideas (?)