實驗主題(Lab08)

B083022053

黃啟桓

Exercise 1:

Exercise 1: 4-bit by 4-bit Binary Multiplier

- Design and verify the 4-bit by 4-bit binary multiplier

Exercise 2:

Exercise 2: 8-bit Magnitude Comparator (1/2)

- Design and verify the 8-bit magnitude comparator composed of two 4-bit magnitude comparators

實驗內容

一、 Exercise 1:

Exercise 1: 4-bit by 4-bit Binary Multiplier

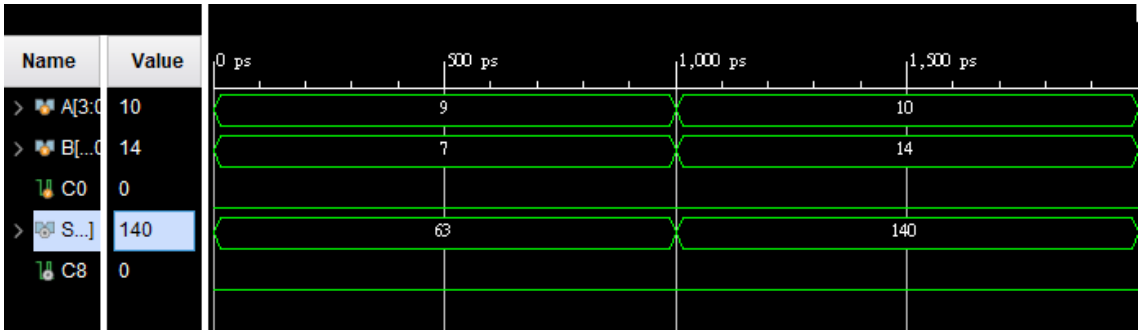
Design and verify the 4-bit by 4-bit binary multiplier

```
`timescale 1ns / 1ps
module half_adder (output S, C, input x, y);
xor (S, x, y);
and (C, x, y);
endmodule

module full_adder (output S, C, input x, y, z);
wire S1, C1, C2;
half_adder HA1 (S1, C1, x, y);
half_adder HA2 (S, C2, S1, z);
or (C, C2, C1);
endmodule

module multiplier (output [3:0]M, input [3:0]A, input B);
and (M[0],A[0],B);
and (M[1],A[1],B);
and (M[2],A[2],B);
and (M[3],A[3],B);
endmodule

module full_adder_4_number (output S, Cout1, Cout2, Cout3, input A,B,C,D,Cin1,Cin2,Cin3);
wire half_sum1, half_sum2, half_sum3;

full_adder FA0 (half_sum1, Cout1, Cin1, Cin2, Cin3);
full_adder FA1 (half_sum2, Cout2, A, B, C);
full_adder FA (S, Cout3, D, half_sum1, half_sum2);

endmodule

module exercise_1 ( output [7: 0] Sum, output C8, input [3:0] A, B, input C0);
wire C1, C2, C3, C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,C17; // Intermediate carries
wire [3:0]M0;
wire [3:0]M1;
wire [3:0]M2;
wire [3:0]M3;
multiplier multiplier0(M0,A,B[0]);
multiplier multiplier1(M1,A,B[1]);
multiplier multiplier2(M2,A,B[2]);
multiplier multiplier3(M3,A,B[3]);

full_adder FA0 (Sum[0], C1, M0[0], 1'b0, C0);
full_adder FA1 (Sum[1], C2, M1[0], M0[1], C1);

full_adder_4_number FA4_2(Sum[2], C3, C4, C5, M2[0], M1[1], M0[2], 1'b0, C2, 1'b0, 1'b0);
full_adder_4_number FA4_3(Sum[3], C6, C7, C8, M3[0], M2[1], M1[2], M0[3], C3, C4, C5);
full_adder_4_number FA4_4(Sum[4], C9, C10, C11, M3[1], M2[2], M1[3], 1'b0, C6, C7, C8);
full_adder_4_number FA4_5(Sum[5], C12, C13, C14, M3[2], M2[3], 1'b0, 1'b0, C9, C10, C11);
full_adder_4_number FA4_6(Sum[6], C15, C16, C17, M3[3], 1'b0, 1'b0, 1'b0, C12, C13, C14);

full_adder FA7 (Sum[7], C8, C15, C16, C17);
endmodule
```

4 位元乘法中，最多會有 7 個 input 要相加，7 個 input 用 3 個 full adder 最多會產生 3 個進位，所以最多會有 7 個 input

$$9 \times 7 = 63 = 00111111_2$$

$$10 \times 14 = 140 = 10001100_2$$



二、Exercise　2:

```
1   `timescale 1ns / 1ps
2   module xi (output x, input A, B);
3   wire n1;
4   xor (n1, A, B);
5   assign x=~n1;
6   endmodule
7
8   module Gi (output G, input [3:0]A, B);
9   wire x0,x1,x2,x3;
10  xi x_0(x0,A[0],B[0]),
11  x_1(x1,A[1],B[1]),
12  x_2(x2,A[2],B[2]),
13  x_3(x3,A[3],B[3]);
14  assign G=(A[3]&(~B[3])) | (x3&A[2]&(~B[2])) | (x3&x2&A[1]&(~B[1])) | (x3&x2&x1&A[0]&(~B[0]));
15  endmodule
16
17  module Li (output L, input [3:0]A, B);
18  wire x0,x1,x2,x3;
19  xi x_0(x0,A[0],B[0]),
20  x_1(x1,A[1],B[1]),
21  x_2(x2,A[2],B[2]),
22  x_3(x3,A[3],B[3]);
23  assign L=(B[3]&(~A[3])) | (x3&B[2]&(~A[2])) | (x3&x2&B[1]&(~A[1])) | (x3&x2&x1&B[0]&(~A[0]));
24  endmodule
25
26  module Ei (output E, input [3:0]A, B);
27  wire x0,x1,x2,x3;
28  xi x_0(x0,A[0],B[0]),
29  x_1(x1,A[1],B[1]),
30  x_2(x2,A[2],B[2]),
31  x_3(x3,A[3],B[3]);
32  assign E=x3&x2&x1&x0;
33  endmodule
34
35  module Magnitude_Comparator_4_bit   ( output E, L, G, input [3:0] A, B);
36  Ei E1(E,A[3:0],B[3:0]);
37  Gi G1(G,A[3:0],B[3:0]);
38  Li L1(L,A[3:0],B[3:0]);
39  endmodule
40
41  module exercise_2 ( output E, L, G, input [7:0] A, B);
42  wire E1,E2,G1,G2,L1,L2;
43  Magnitude_Comparator_4_bit  MC4_1 ( E1, L1, G1, A[7:4], B[7:4]);
44  Magnitude_Comparator_4_bit  MC4_2 ( E2, L2, G2, A[3:0], B[3:0]);
45  assign E=(E1&E2);
46  assign G=G1|(E1&G2);
47  assign L=L1|(E1&L2);
48
49  endmodule
```

8bit 比較器拆成 2 個 4bit 比較器

4bit 比較器拆成 L、G、E 3 個小 module

L、G、E 3 個 module 提出共同 module :x

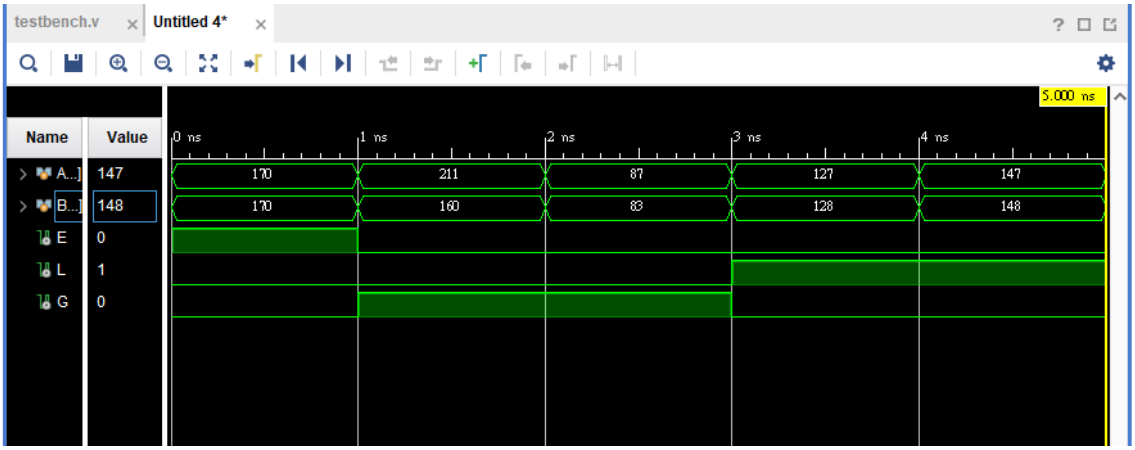
10101010, 10101010

11010011, 10100000

01010111, 01010011

01111111, 10000000

10010011, 10010100



實驗心得

1. Structural level modeling 中邏輯閘使用函數表示
2. Structural level modeling 可以很直觀的轉換成 Schematic
3. Structural level modeling 需要宣告邏輯閘之間的線路
4. 所有的 modeling 都需要注意該邏輯閘是否有交換律或結合律