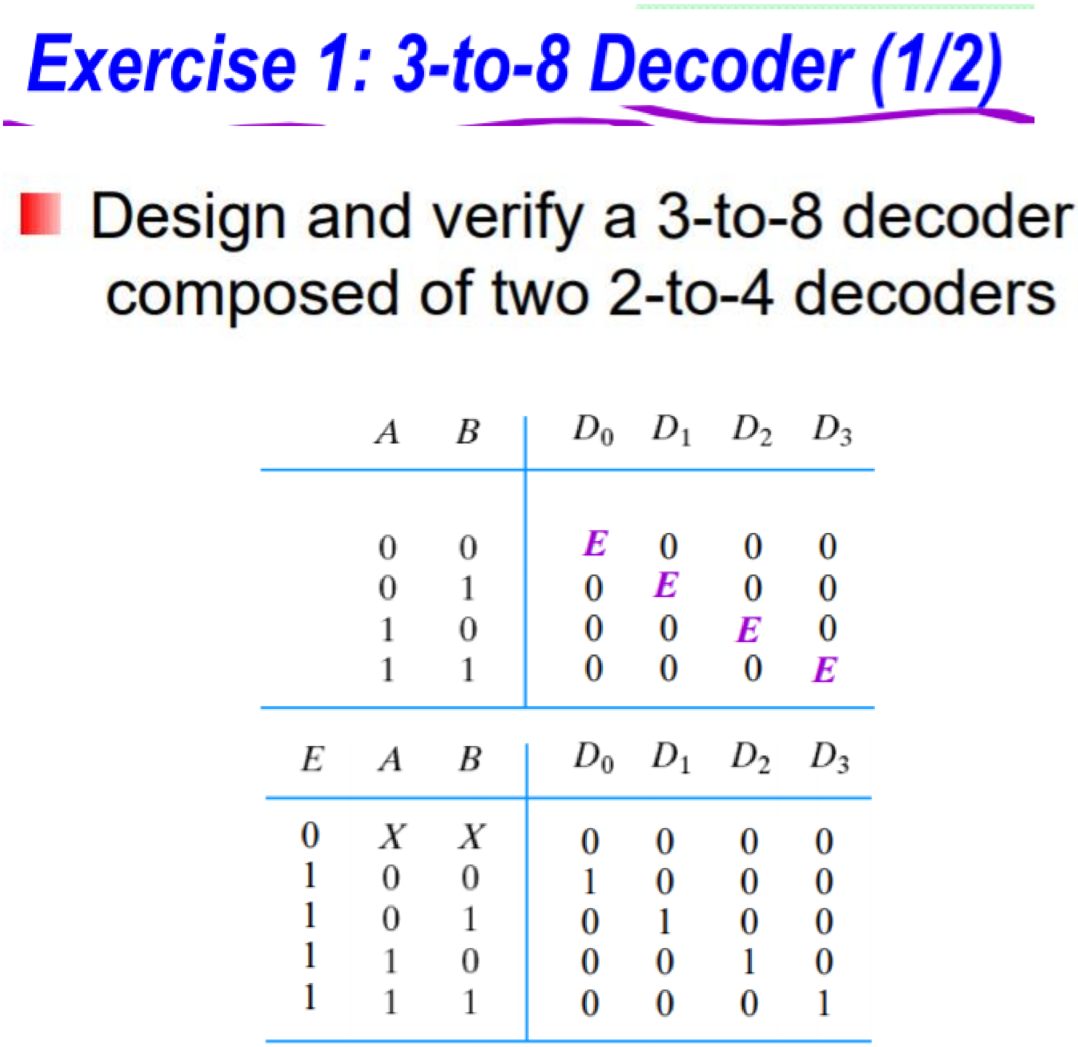
實驗主題(Lab10)

B083022053

黃啟桓

Exercise 1:



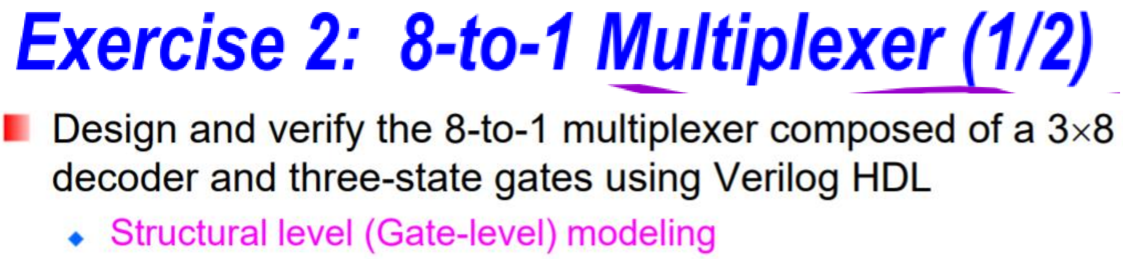
Exercise 1: 3-to-8 Decoder (1/2)

- Design and verify a 3-to-8 decoder composed of two 2-to-4 decoders

A	B	D_0	D_1	D_2	D_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

E	A	B	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

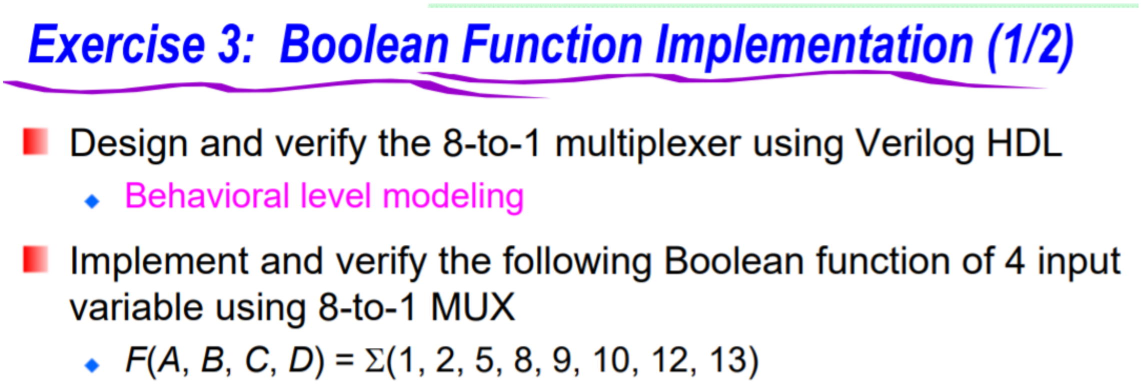
Exercise 2:



Exercise 2: 8-to-1 Multiplexer (1/2)

- Design and verify the 8-to-1 multiplexer composed of a 3×8 decoder and three-state gates using Verilog HDL
  - Structural level (Gate-level) modeling

Exercise 3:

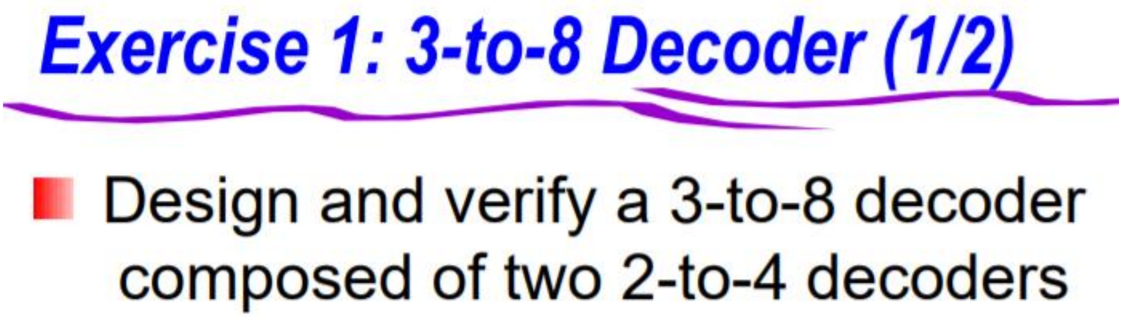


Exercise 3: Boolean Function Implementation (1/2)

- Design and verify the 8-to-1 multiplexer using Verilog HDL
  - Behavioral level modeling
- Implement and verify the following Boolean function of 4 input variable using 8-to-1 MUX
  - $F(A, B, C, D) = \Sigma(1, 2, 5, 8, 9, 10, 12, 13)$

實驗內容

一、 Exercise 1:



Exercise 1: 3-to-8 Decoder (1/2)

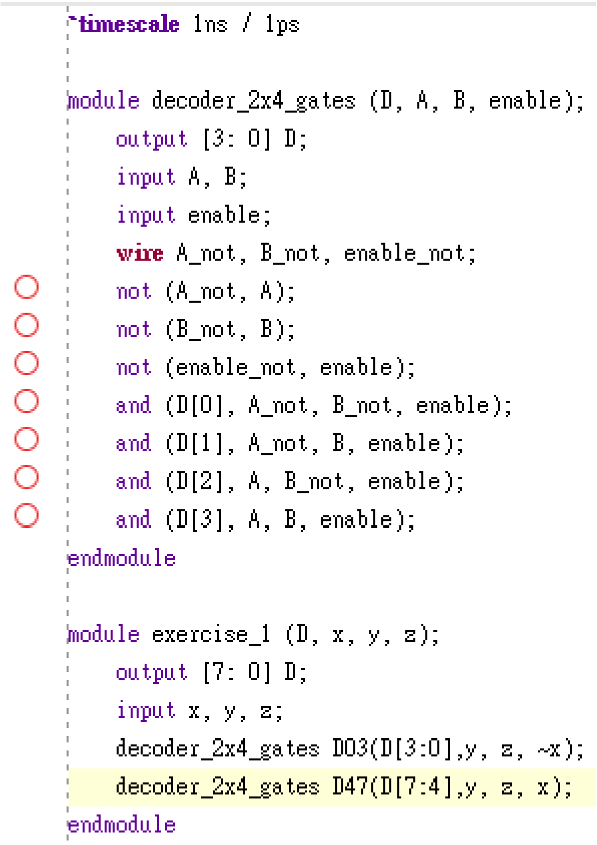
- Design and verify a 3-to-8 decoder composed of two 2-to-4 decoders

3x8 decoder 用 2 個 2x4

decorder 接

2x4 decorder =

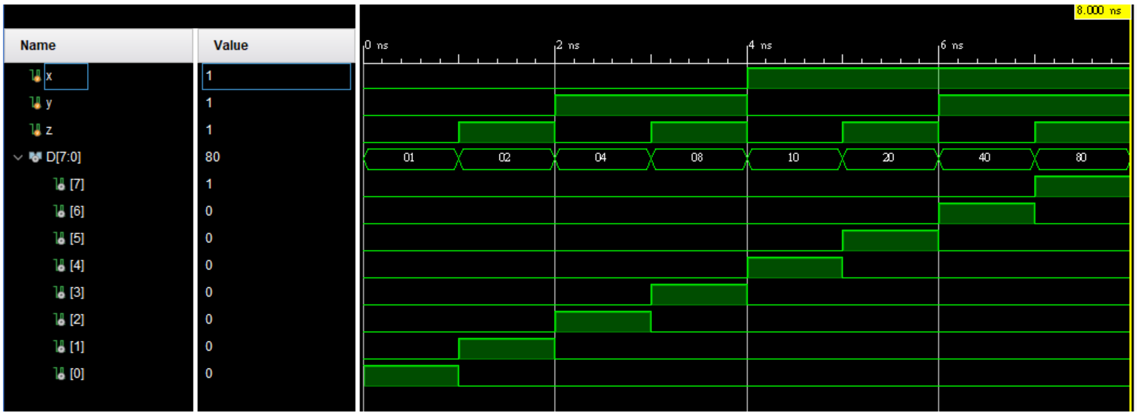
∑(1,2,3,4)



```
`timescale 1ns / 1ps

module decoder_2x4_gates (D, A, B, enable);
    output [3: 0] D;
    input A, B;
    input enable;
    wire A_not, B_not, enable_not;
    not (A_not, A);
    not (B_not, B);
    not (enable_not, enable);
    and (D[0], A_not, B_not, enable);
    and (D[1], A_not, B, enable);
    and (D[2], A, B_not, enable);
    and (D[3], A, B, enable);
endmodule

module exercise_1 (D, x, y, z);
    output [7: 0] D;
    input x, y, z;
    decoder_2x4_gates D03(D[3:0],y, z, ~x);
    decoder_2x4_gates D47(D[7:4],y, z, x);
endmodule
```

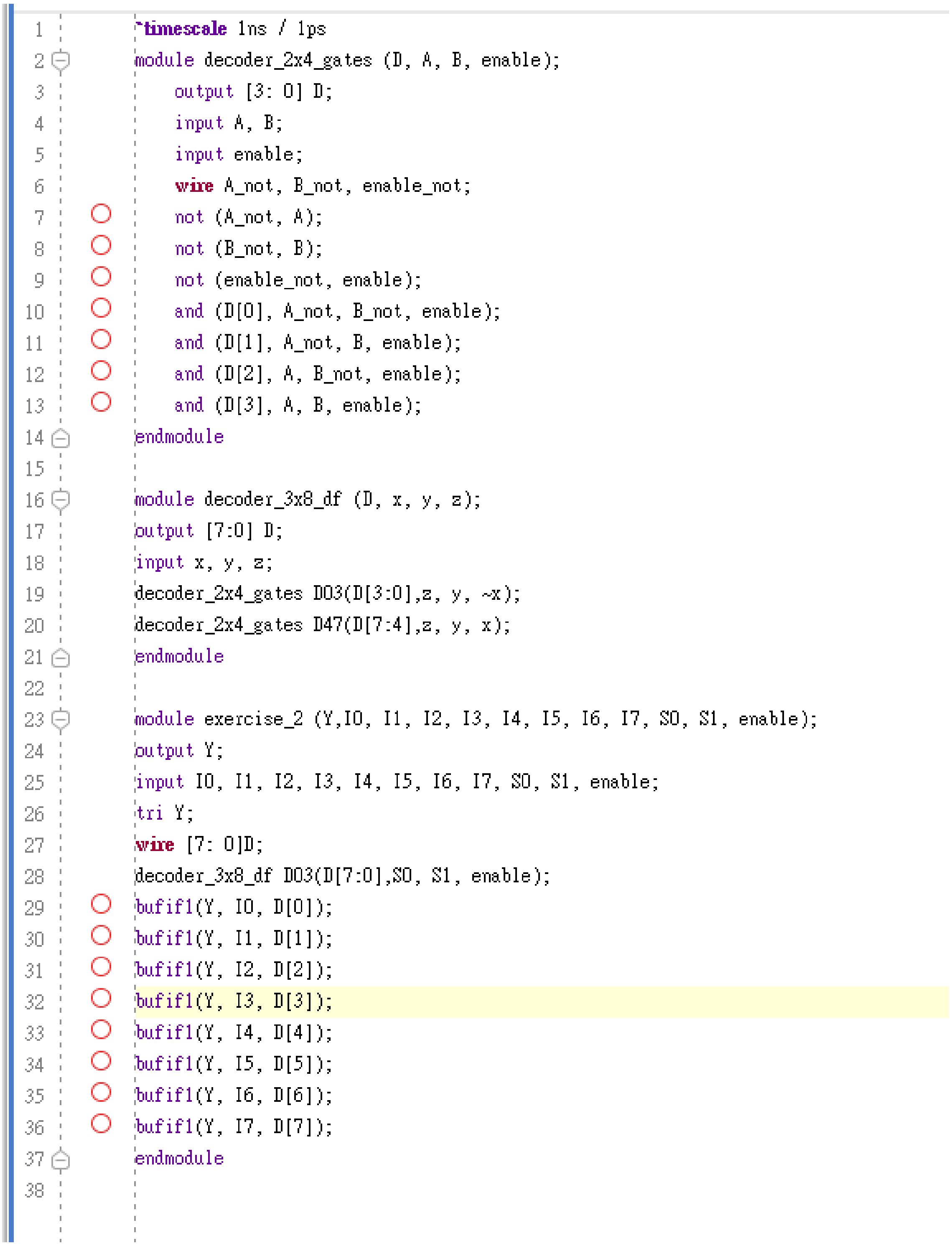


Name	Value
x	1
y	1
z	1
D[7:0]	80
[7]	1
[6]	0
[5]	0
[4]	0
[3]	0
[2]	0
[1]	0
[0]	0

二、Exercise 2:

Exercise 2: 8-to-1 Multiplexer (1/2)

- Design and verify the 8-to-1 multiplexer composed of a 3×8 decoder and three-state gates using Verilog HDL
  - Structural level (Gate-level) modeling



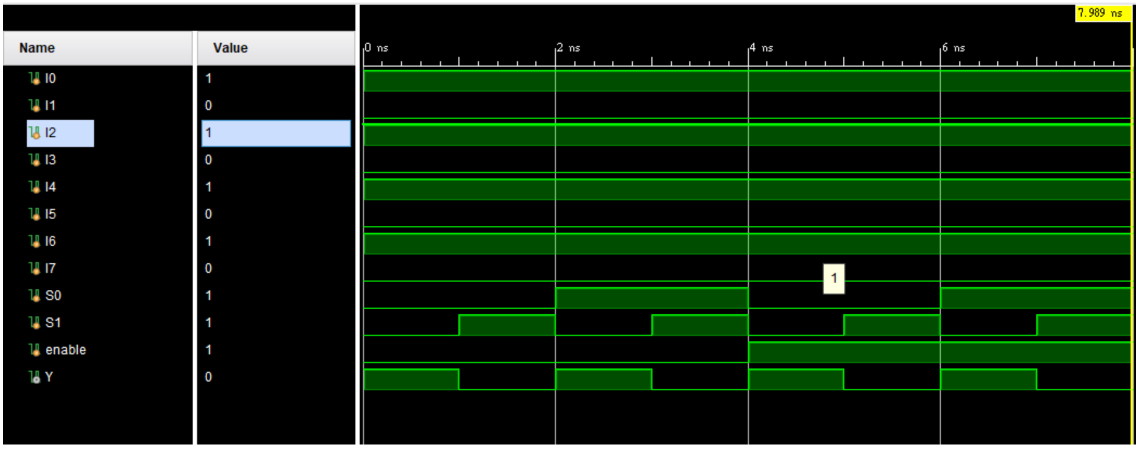
```
`timescale 1ns / 1ps
module decoder_2x4_gates (D, A, B, enable);
    output [3: 0] D;
    input A, B;
    input enable;
    wire A_not, B_not, enable_not;
    not (A_not, A);
    not (B_not, B);
    not (enable_not, enable);
    and (D[0], A_not, B_not, enable);
    and (D[1], A_not, B, enable);
    and (D[2], A, B_not, enable);
    and (D[3], A, B, enable);
endmodule

module decoder_3x8_df (D, x, y, z);
output [7:0] D;
input x, y, z;
decoder_2x4_gates D03(D[3:0],z, y, ~x);
decoder_2x4_gates D47(D[7:4],z, y, x);
endmodule

module exercise_2 (Y,I0, I1, I2, I3, I4, I5, I6, I7, S0, S1, enable);
output Y;
input I0, I1, I2, I3, I4, I5, I6, I7, S0, S1, enable;
tri Y;
wire [7: 0]D;
decoder_3x8_df D03(D[7:0],S0, S1, enable);
bufif1(Y, I0, D[0]);
bufif1(Y, I1, D[1]);
bufif1(Y, I2, D[2]);
bufif1(Y, I3, D[3]);
bufif1(Y, I4, D[4]);
bufif1(Y, I5, D[5]);
bufif1(Y, I6, D[6]);
bufif1(Y, I7, D[7]);
endmodule
```

8bit MUX 拆成 3x8 decoder + three-state gates ，
3x8 decoder 拆成 2 個 2x4 decoder

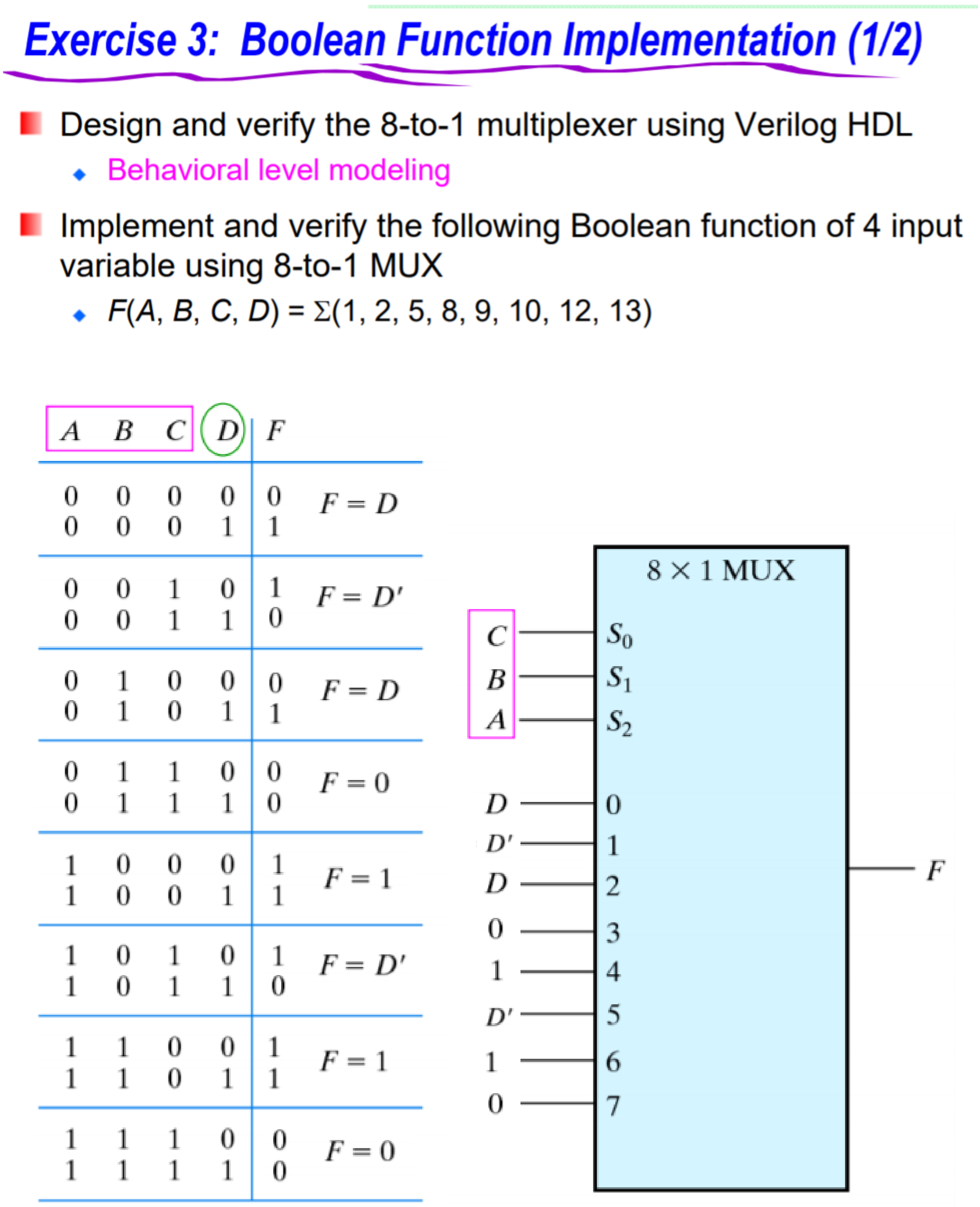
2x4 decorder = Σ(1,2,3,4)



Name	Value
I0	1
I1	0
I2	1
I3	0
I4	1
I5	0
I6	1
I7	0
S0	1
S1	1
enable	1
Y	0

Y=I0 > I1 > I2 > I3 > I4 > I5 > I6 > I7

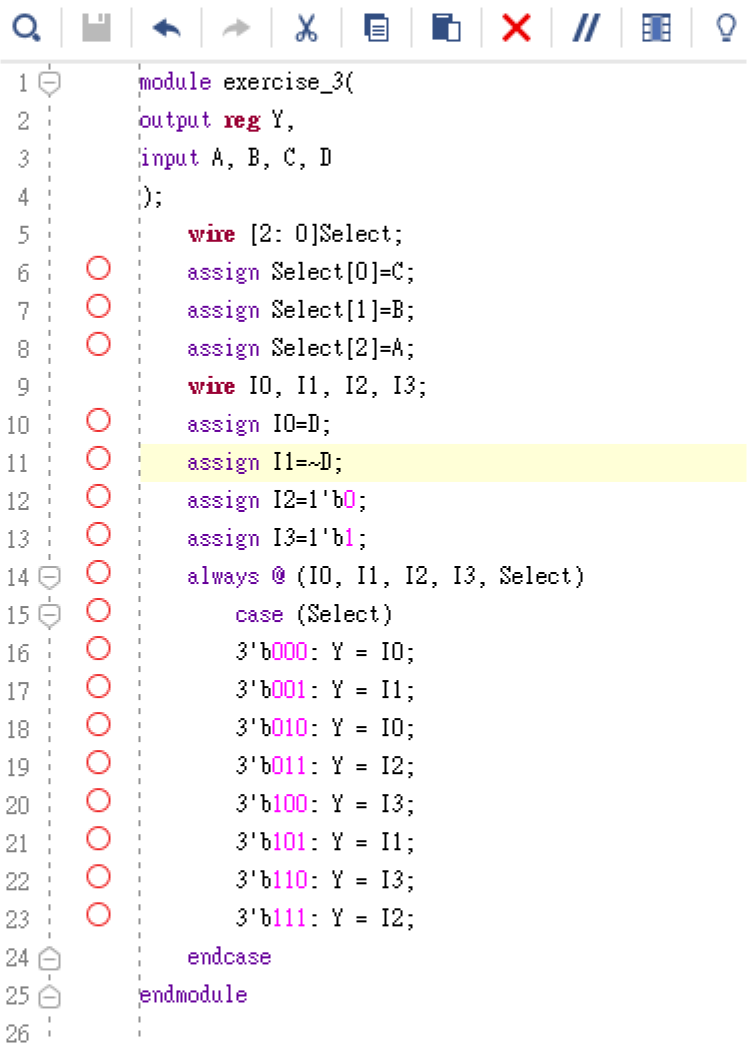
三、Exercise 3:



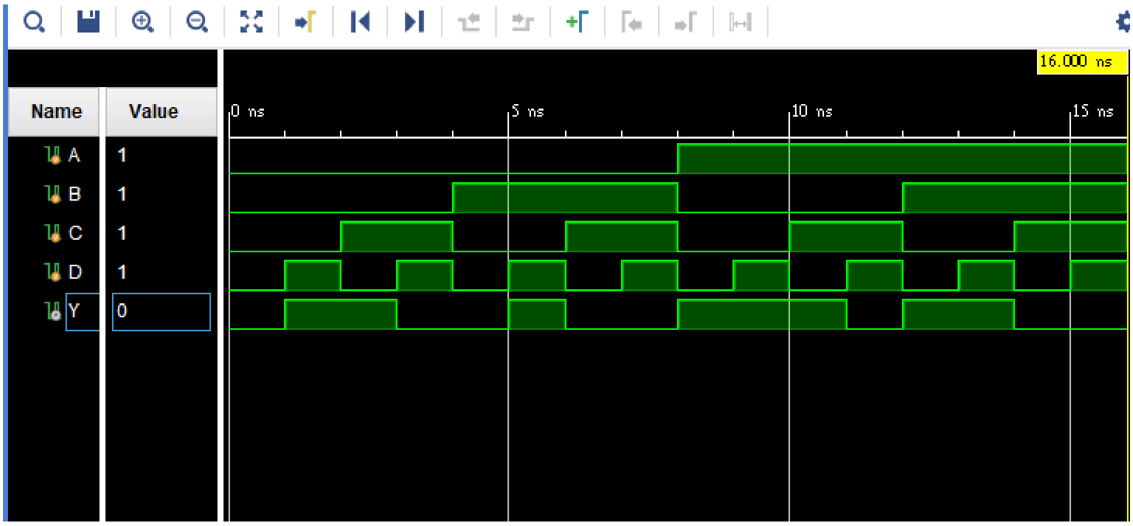
Exercise 3: Boolean Function Implementation (1/2)

- Design and verify the 8-to-1 multiplexer using Verilog HDL
  - Behavioral level modeling
- Implement and verify the following Boolean function of 4 input variable using 8-to-1 MUX
  - $F(A, B, C, D) = \Sigma(1, 2, 5, 8, 9, 10, 12, 13)$

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	1	$F = D'$
0	0	1	1	0	
0	1	0	0	0	$F = D$
0	1	0	1	1	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	1	$F = 1$
1	0	0	1	1	
1	0	1	0	1	$F = D'$
1	0	1	1	0	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	0	$F = 0$
1	1	1	1	0	



```
module exercise_3(
output reg Y,
input A, B, C, D
);
    wire [2: 0]Select;
    assign Select[0]=C;
    assign Select[1]=B;
    assign Select[2]=A;
    wire I0, I1, I2, I3;
    assign I0=D;
    assign I1=~D;
    assign I2=1'b0;
    assign I3=1'b1;
    always @ (I0, I1, I2, I3, Select)
        case (Select)
        3'b000: Y = I0;
        3'b001: Y = I1;
        3'b010: Y = I0;
        3'b011: Y = I2;
        3'b100: Y = I3;
        3'b101: Y = I1;
        3'b110: Y = I3;
        3'b111: Y = I2;
    endcase
endmodule
```



Name	Value
A	1
B	1
C	1
D	1
Y	0

Y=∑(1,2,35,8,9,10,12,13)

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