DIGITAL DESIGN

ASSIGNMENT REPORT

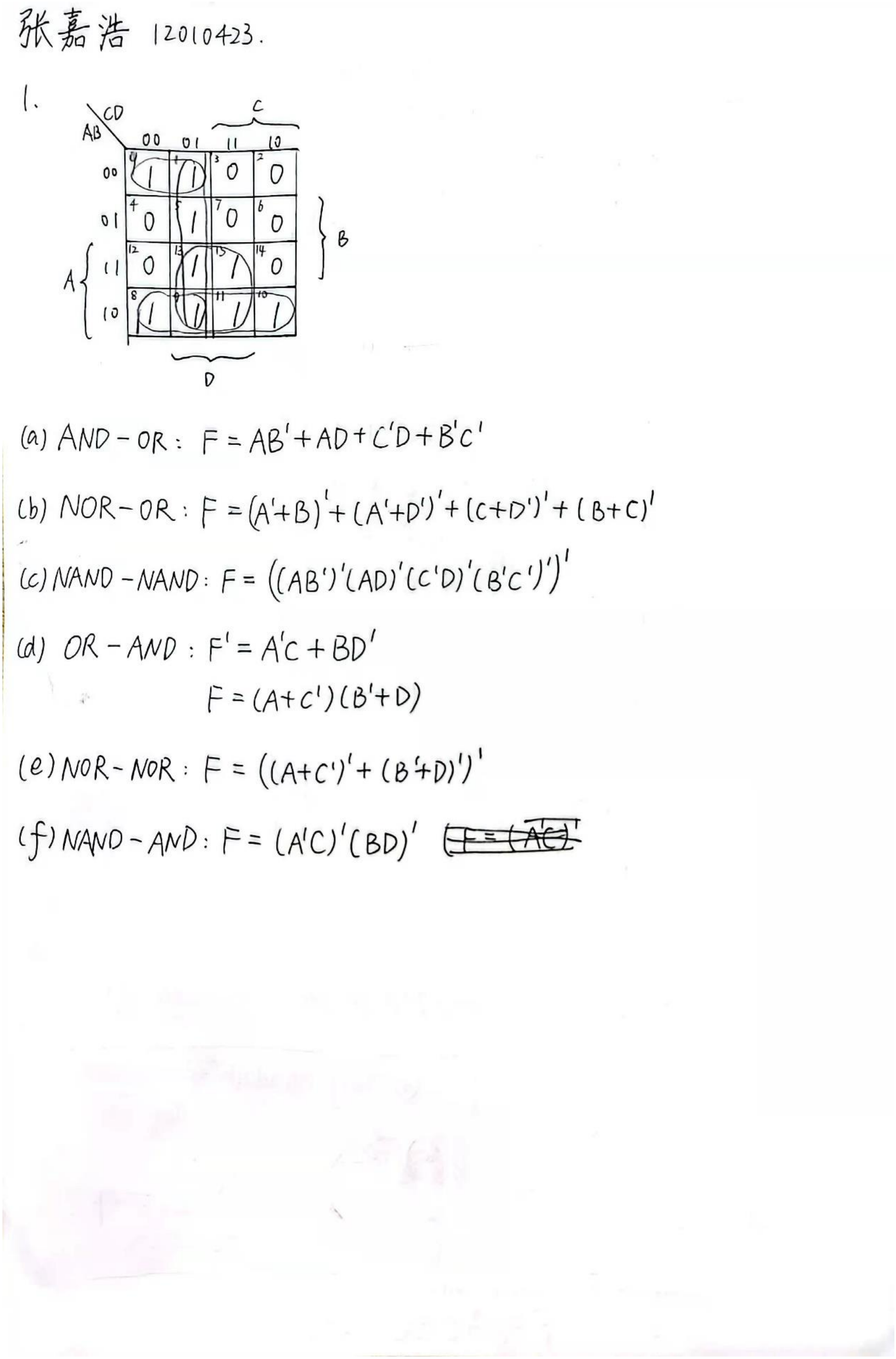
ASSIGNMENT ID: 12010423

Student Name: 张嘉浩

Student ID: 12010423

PART 1: DIGITAL DESIGN THEORY

Provide your answers here:



张嘉浩 12010423.

1.

AB \ CD	00	01	11	10
00	^{0} 1	^{1} 1	^{3} 0	^{2} 0
01	^{4} 0	^{5} 1	^{7} 0	^{6} 0
11	^{12} 0	^{13} 1	^{15} 1	^{14} 0
10	^{8} 1	^{9} 1	^{11} 1	^{10} 1

(a) AND-OR: $F = AB' + AD + C'D + B'C'$

(b) NOR-OR: $F = (A'+B)' + (A'+D')' + (C+D')' + (B+C)'$

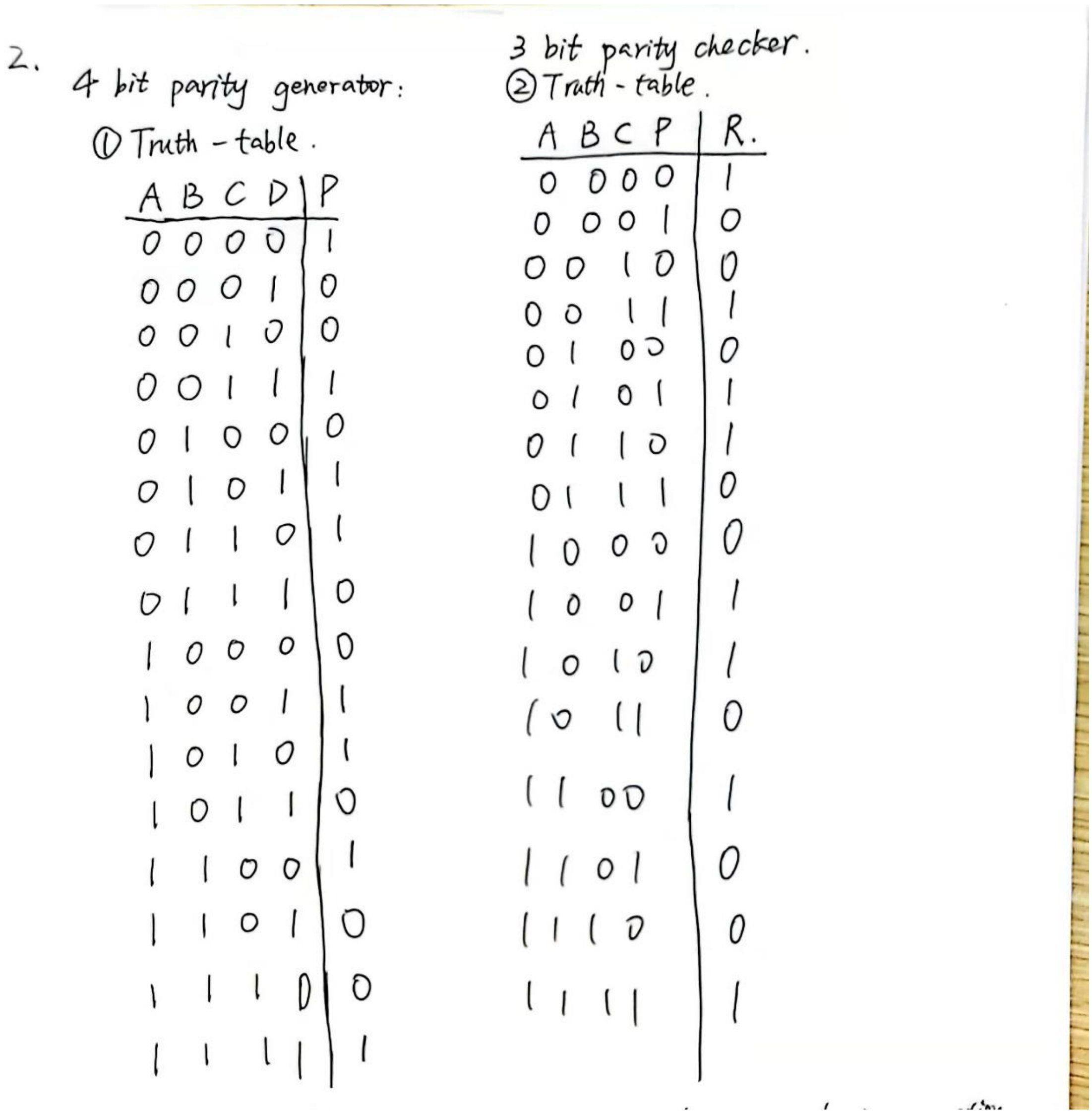
(c) NAND-NAND: $F = ((AB')'(AD)'(C'D)'(B'C')')'$

(d) OR-AND: $F' = A'C + BD'$

$F = (A+C')(B'+D)$

(e) NOR-NOR: $F = ((A+C')' + (B'+D)')'$

(f) NAND-AND: $F = (A'C)'(BD)'$ ~~$F = (A'C)'$~~



2.

4 bit parity generator:

① Truth - table.

A	B	C	D	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

3 bit parity checker.

② Truth - table.

A	B	C	P	R.
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

2、② Simplification .

Both can be simplified to the same logic equation (same Kmap).

C

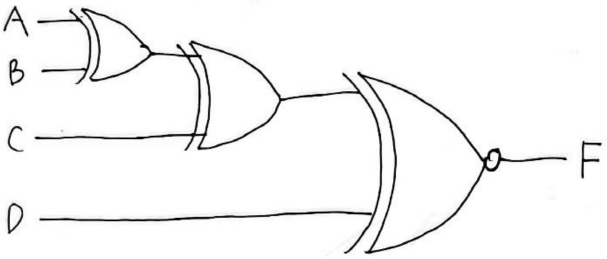
	AB \ CD	00	01	11	10	
	00	1	0	1	0	
	01	0	1	0	1	B
A	11	1	0	1	0	
	10	0	1		1	

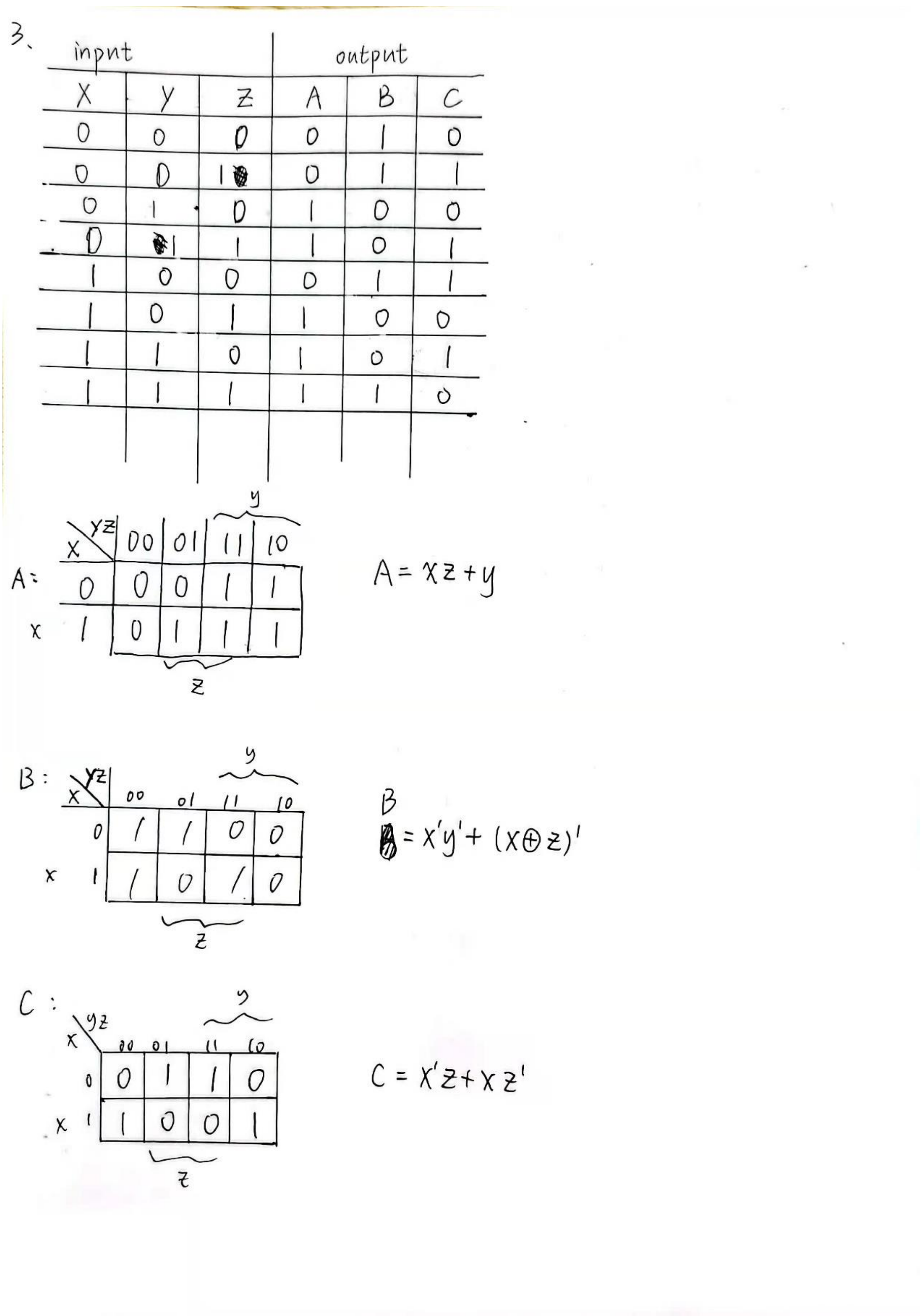
D

thus both equation are:

$F = (A \oplus B \oplus C \oplus D)'$

③ Both shared the same logic diagram as below





3、

input			output		
x	y	z	A	B	C
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

A:

x \ yz	00	01	11	10
0	0	0	1	1
x 1	0	1	1	1

$A = xz + y$

B:

x \ yz	00	01	11	10
0	1	1	0	0
x 1	1	0	1	0

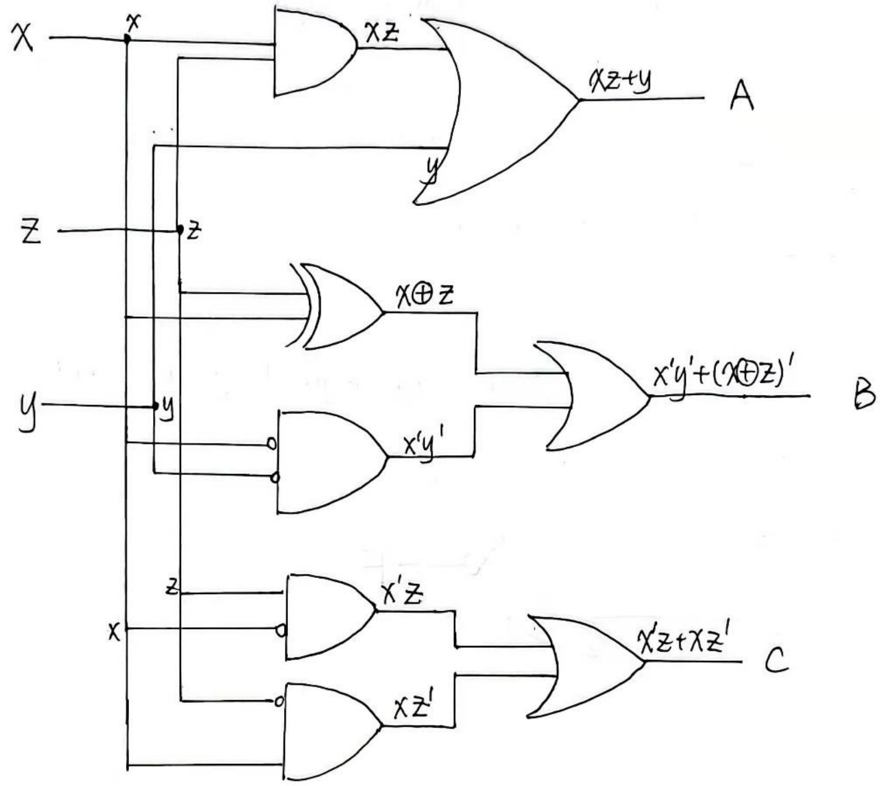
$B = x'y' + (x \oplus z)'$

C:

x \ yz	00	01	11	10
0	0	1	1	0
x 1	1	0	0	1

$C = x'z + xz'$

3. Circuit diagram.



4.

$$
\begin{array}{cccccc}
 & & & A_2 & A_1 & A_0 \\
\times & & & B_2 & B_1 & B_0 \\
\hline
 & & & A_2B_0 & A_1B_0 & A_0B_0 \\
 & & A_2B_1 & A_1B_1 & A_0B_1 & \\
 & A_2B_2 & A_1B_2 & A_0B_2 & & \\
\hline
C_5 & C_4 & C_3 & C_2 & C_1 & C_0
\end{array}
$$

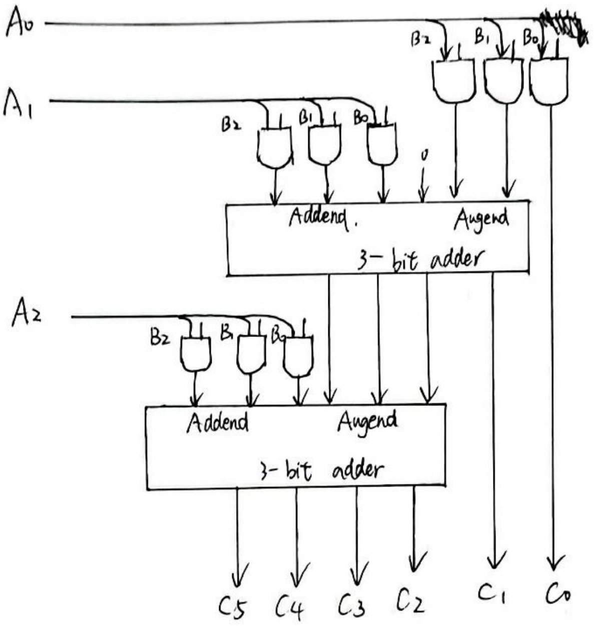
$C_0 = A_0B_0$

$C_1 = A_1B_0 + A_0B_1$

$C_2 = A_2B_0 + A_1B_1 + A_0B_2$

$C_3 = A_2B_1 + A_1B_2$

$C_4 = A_2B_2$



5. (a) $F(A,B,C,D)=\sum(1,3,5,8,10,14)$

$= 0\times A'B'C'D' + 1\times A'B'C'D + 0\times A'B'CD' + 1\times A'B'CD + 0\times A'BC'D' + 1\times A'BC'D$

$+ 0\times A'BCD' + 0\times A'BCD + 1\times AB'C'D' + 0\times AB'C'D + 1\times AB'CD' + 0\times AB'CD +$

$0\times ABC'D' + 0\times ABC'D + 1\times ABCD' + 0\times ABCD$

$$= \sum_{i=0}^{15} D_i m_i$$

$= m_0\times 0 + m_1\times 1 + m_2\times 0 + m_3\times 1 + m_4\times 0 + m_5\times 1 + m_6\times 0 + m_7\times 0 + m_8\times 1 +$

$m_9\times 0 + m_{10}\times 1 + m_{11}\times 0 + m_{12}\times 0 + m_{13}\times 0 + m_{14}\times 1 + m_{15}\times 0$.

(b) $F(A,B,C,D)=\prod(4,7,11)$

$=\sum(0,1,2,3,5,6,8,9,10,12,13,14,15)$

$= 1\times A'B'C'D' + 1\times A'B'C'D + 1\times A'B'CD' + 1\times A'B'CD + 0\times A'BC'D' + 1\times A'BC'D +$

$1\times A'BCD' + 0\times A'BCD + 1\times AB'C'D' + 1\times AB'C'D + 1\times AB'CD' + 0\times AB'CD +$

$1\times ABC'D' + 1\times ABC'D + 1\times ABCD' + 1\times ABCD$

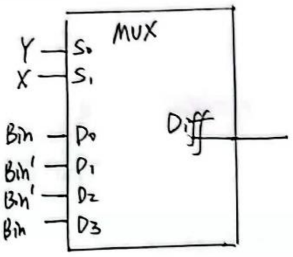
$$= \sum_{i=0}^{15} D_i m_i$$

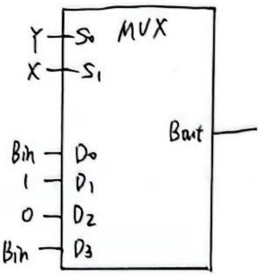
$= m_0\times 1 + m_1\times 1 + m_2\times 1 + m_3\times 1 + m_4\times 0 + m_5\times 1 + m_6\times 1 + m_7\times 0 + m_8\times 1$

$+ m_9\times 1 + m_{10}\times 1 + m_{11}\times 0 + m_{12}\times 1 + m_{13}\times 1 + m_{14}\times 1 + m_{15}\times 1$.

6.

X	Y	Bin	Diff	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1





Simplification:

X \ YBin	00	01	11	10
0	0	1	0	1
1	1	0	1	0

(Kmap for ~~d~~Diff)

thus $Diff = X \oplus Y \oplus Bin$

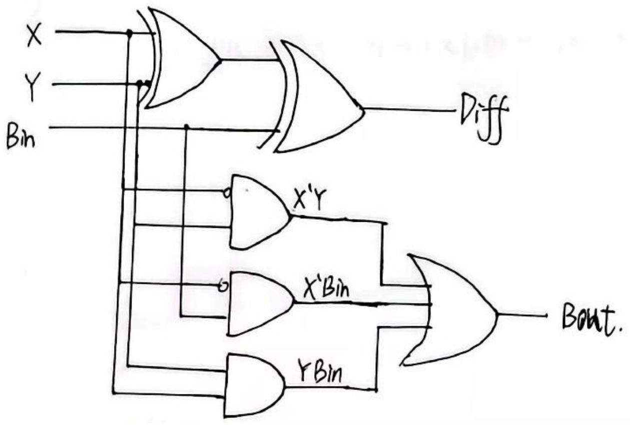
Kmap for Bout

Y

X \ YBin	00	01	11	10
0	0	1	1	1
1	0	0	1	0

Bin

thus Bout $= X'Bin + X'Y + YBin$



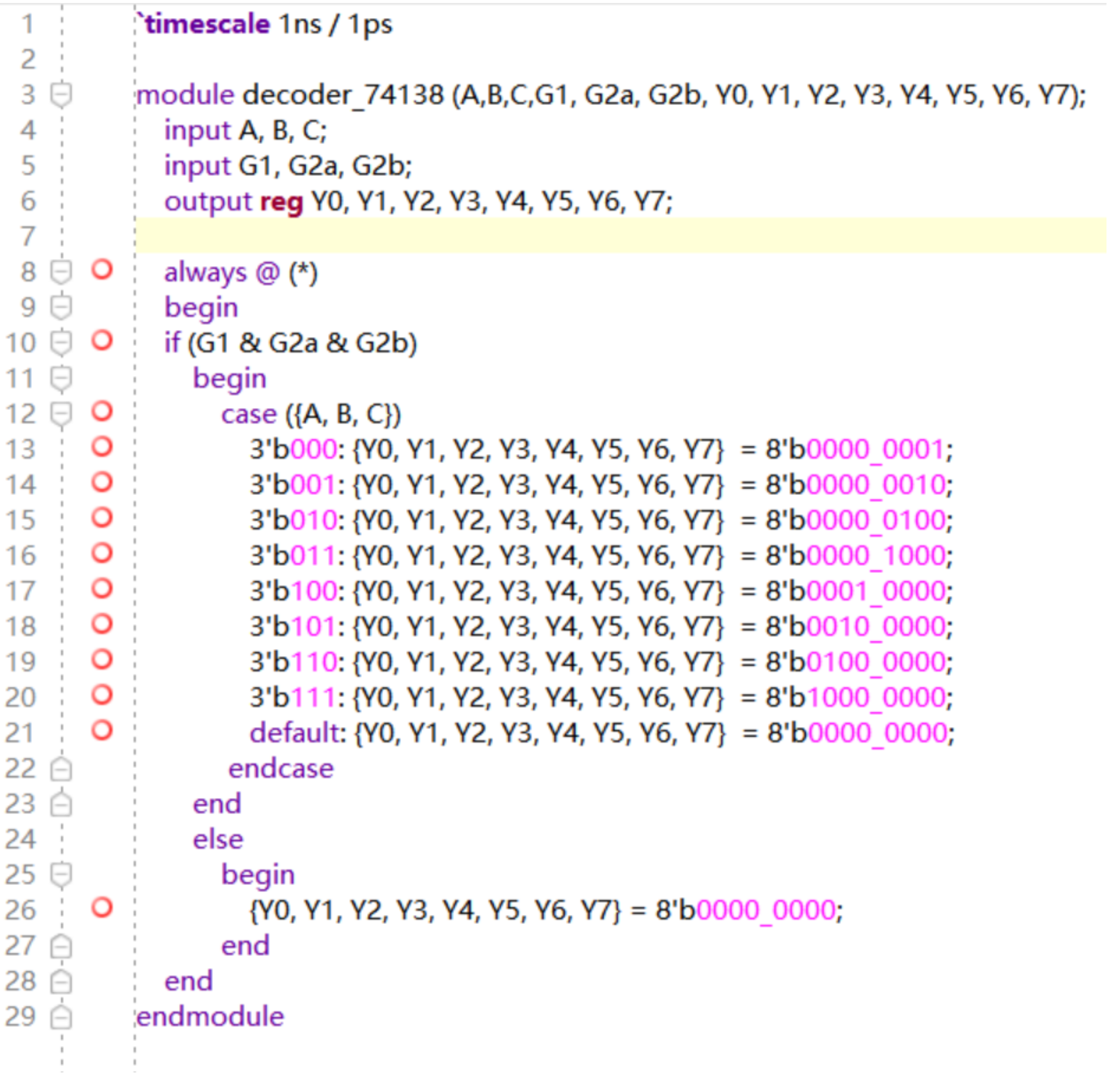
PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

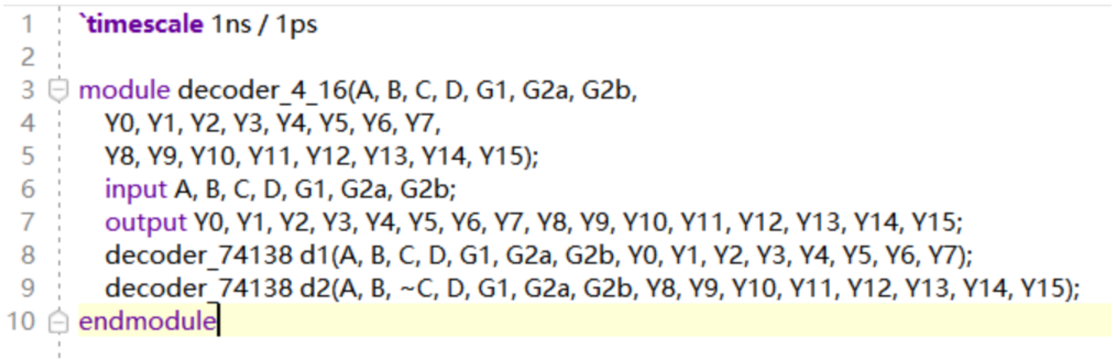
(This part is always REQUIRED)

Describe the design of your system by providing the following information:

- *Verilog design (provide the Verilog code)*



```
`timescale 1ns / 1ps

module decoder_74138 (A,B,C,G1, G2a, G2b, Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7);
  input A, B, C;
  input G1, G2a, G2b;
  output reg Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7;

  always @ (*)
  begin
  if (G1 & G2a & G2b)
     begin
        case ({A, B, C})
           3'b000: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0000_0001;
           3'b001: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0000_0010;
           3'b010: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0000_0100;
           3'b011: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0000_1000;
           3'b100: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0001_0000;
           3'b101: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0010_0000;
           3'b110: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0100_0000;
           3'b111: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b1000_0000;
           default: {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7}  = 8'b0000_0000;
         endcase
     end
     else
        begin
           {Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7} = 8'b0000_0000;
        end
  end
endmodule
```

- 

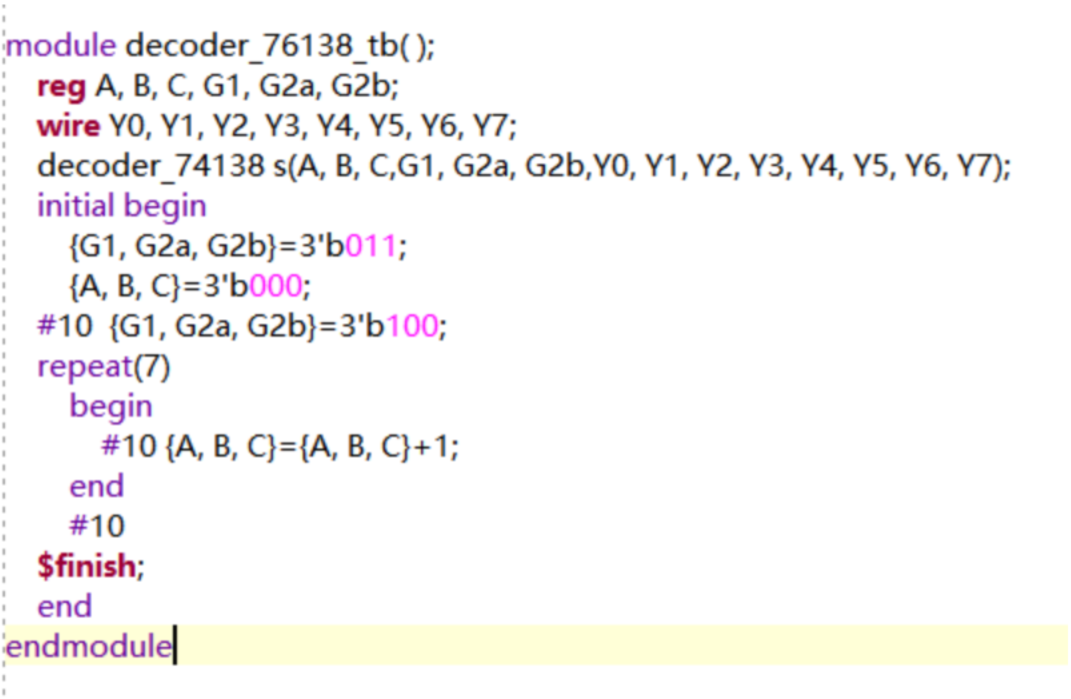
```
`timescale 1ns / 1ps

module decoder_4_16(A, B, C, D, G1, G2a, G2b,
  Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7,
  Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15);
  input A, B, C, D, G1, G2a, G2b;
  output Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15;
  decoder_74138 d1(A, B, C, D, G1, G2a, G2b, Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7);
  decoder_74138 d2(A, B, ~C, D, G1, G2a, G2b, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15);
endmodule
```

SIMULATION

(This part is optional depending on the requirement of the lab task)

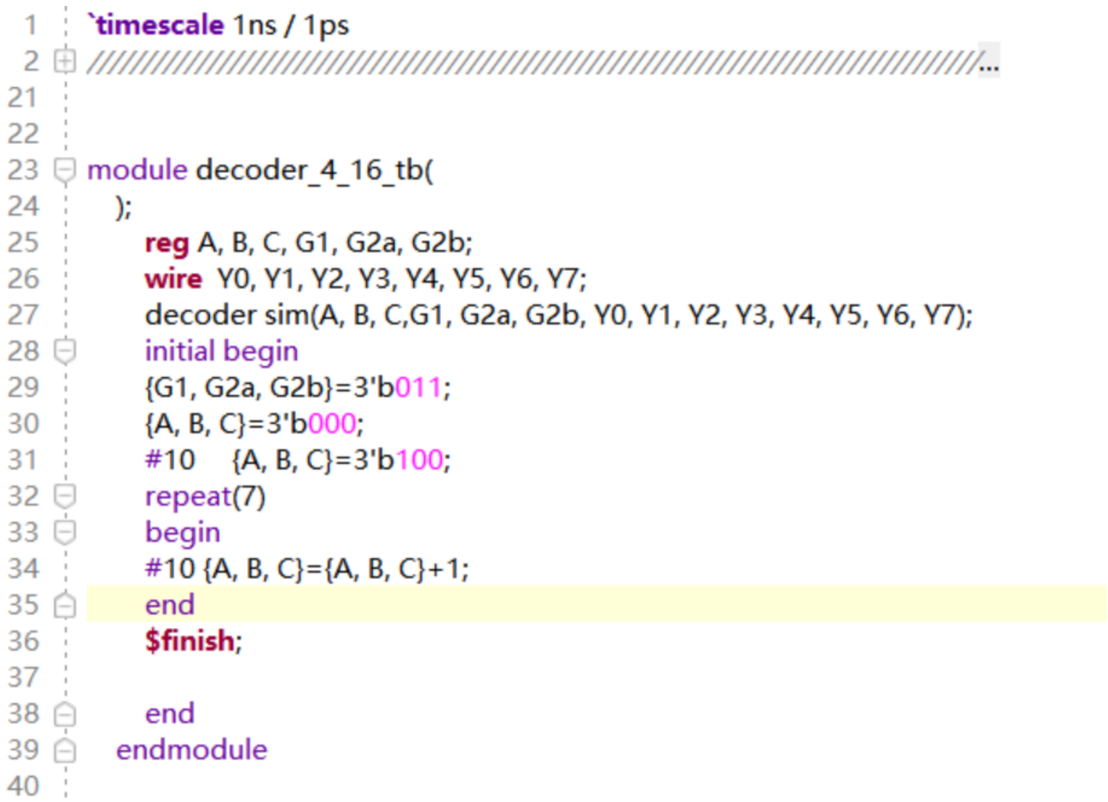
Describe how you build the test bench and do the simulation.

- *Using Verilog (provide the Verilog code)*



```
module decoder_76138_tb( );
  reg A, B, C, G1, G2a, G2b;
  wire Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7;
  decoder_74138 s(A, B, C,G1, G2a, G2b,Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7);
  initial begin
    {G1, G2a, G2b}=3'b011;
    {A, B, C}=3'b000;
  #10  {G1, G2a, G2b}=3'b100;
  repeat(7)
    begin
      #10 {A, B, C}={A, B, C}+1;
    end
    #10
  $finish;
  end
endmodule
```

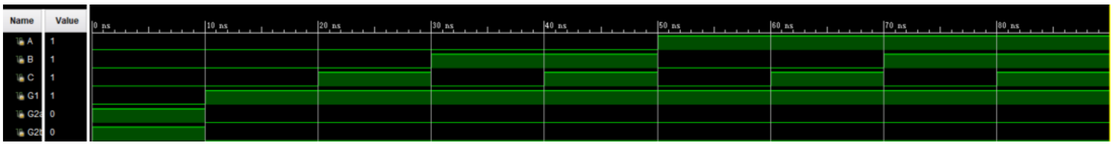
-



```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////...


module decoder_4_16_tb(
    );
        reg A, B, C, G1, G2a, G2b;
        wire  Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7;
        decoder sim(A, B, C,G1, G2a, G2b, Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7);
        initial begin
        {G1, G2a, G2b}=3'b011;
        {A, B, C}=3'b000;
        #10     {A, B, C}=3'b100;
        repeat(7)
        begin
        #10 {A, B, C}={A, B, C}+1;
        end
        $finish;

        end
    endmodule

```

- *Wave form of simulation result (provide screen shots)*

  Really sorry that I failed to get the correct result, I asked my classmates and SA but I still cannot solve it. I'll be so happy if I will not *lose too much mark on this.*



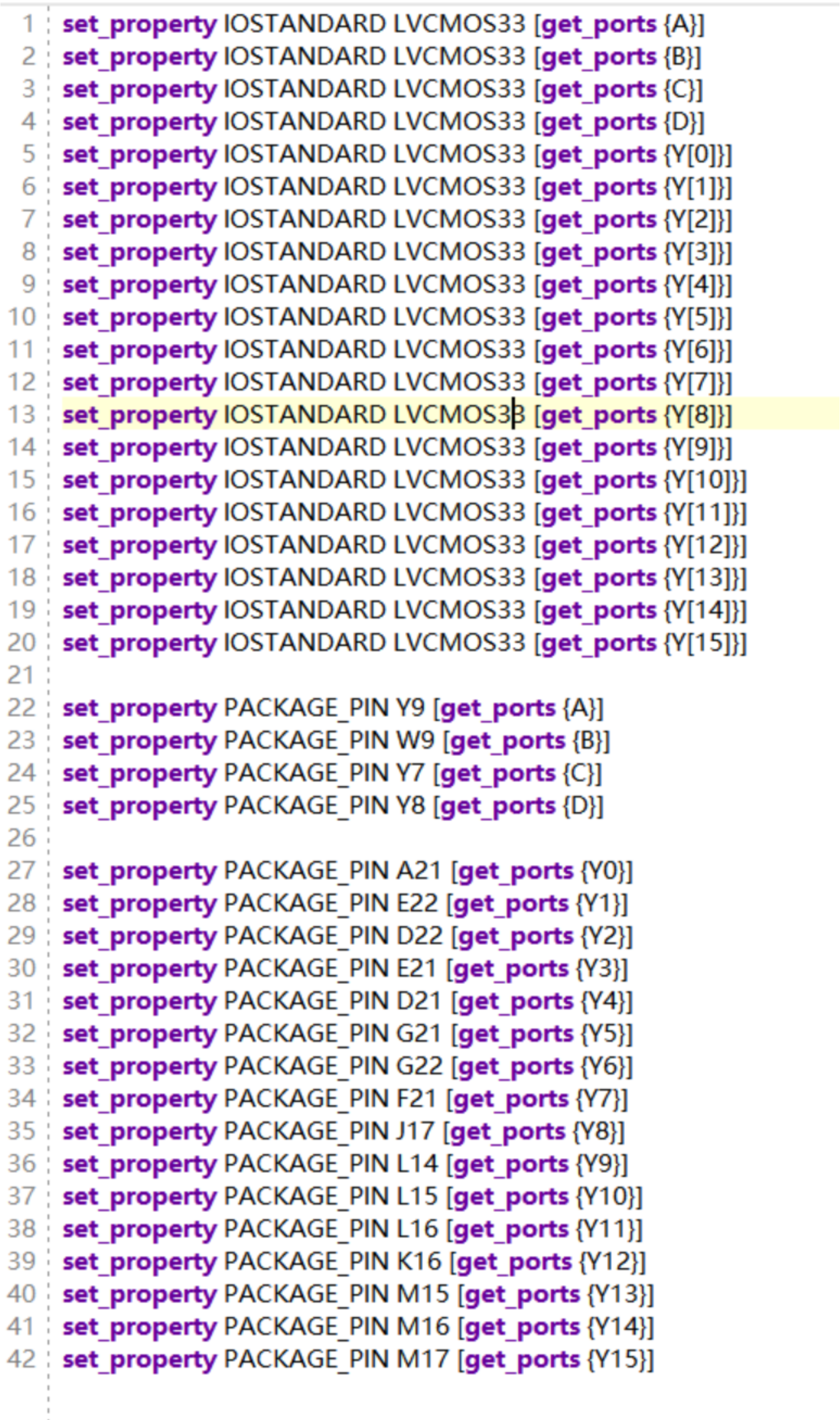
- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation.*

CONSTRAINT FILE AND THE TESTING

(This part is optional depending on the requirement of the lab task)

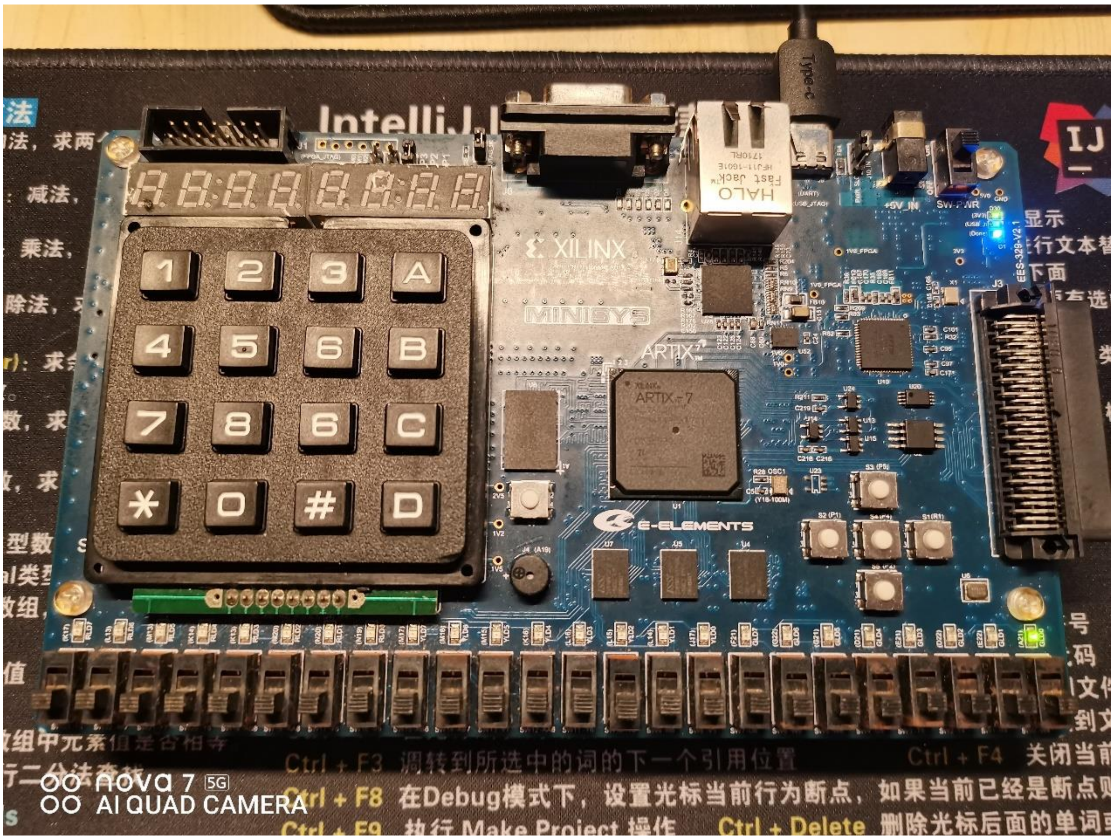
Describe how you test your design on the Minisys Practice platform.

- *Constraint file (provide the screen shots on the feature of a pin and the binding info between pins and the input /output ports)*

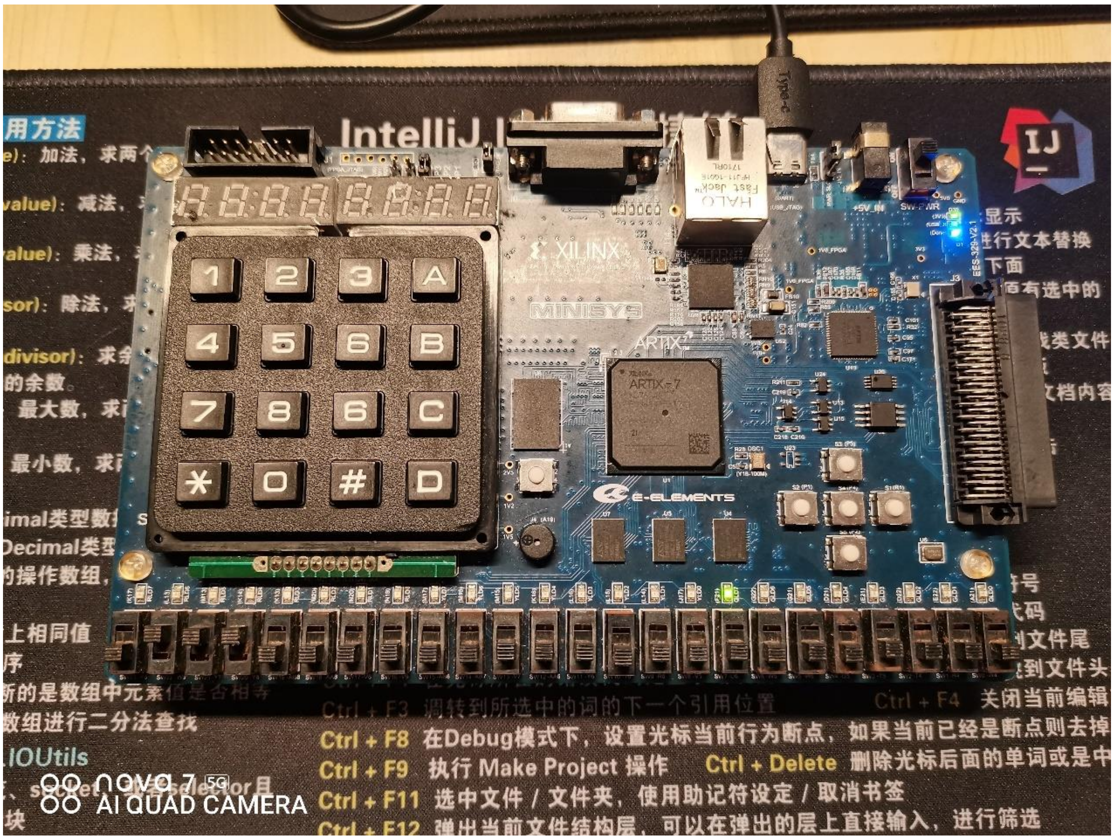
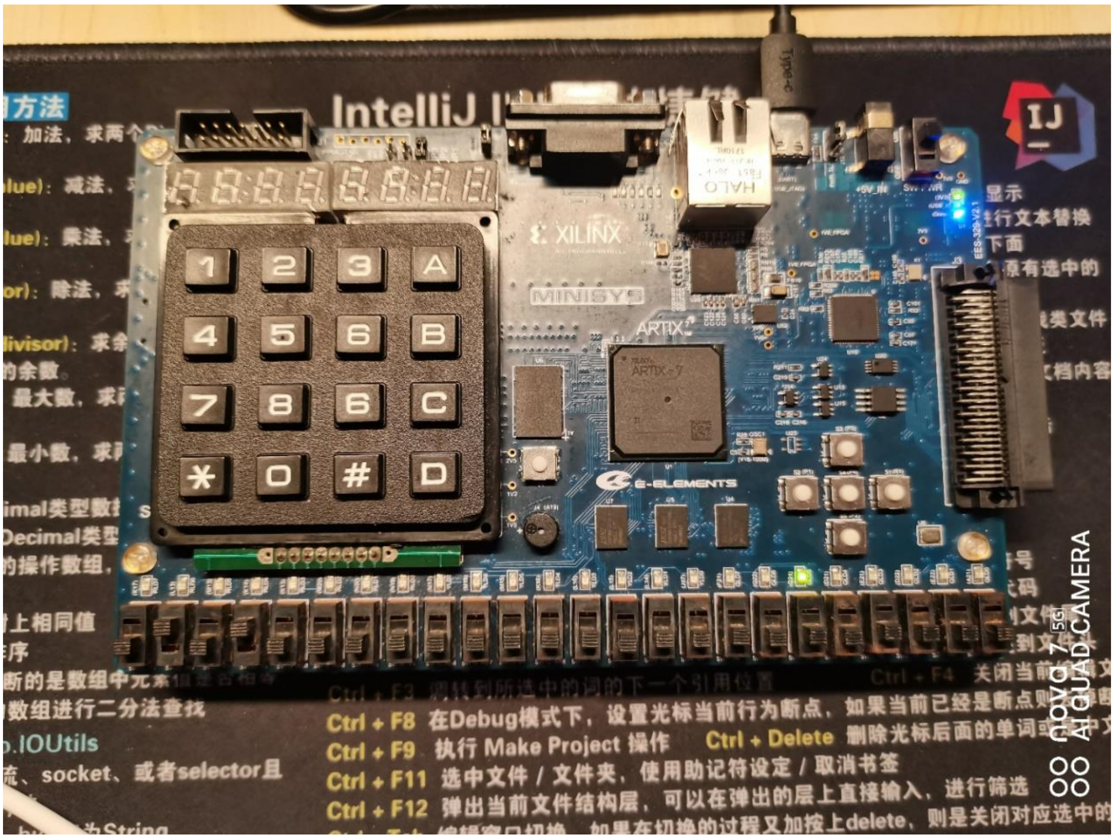


```
set_property IOSTANDARD LVCMOS33 [get_ports {A}]
set_property IOSTANDARD LVCMOS33 [get_ports {B}]
set_property IOSTANDARD LVCMOS33 [get_ports {C}]
set_property IOSTANDARD LVCMOS33 [get_ports {D}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[8]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[9]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[10]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[11]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[12]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[13]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[14]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Y[15]}]

set_property PACKAGE_PIN Y9 [get_ports {A}]
set_property PACKAGE_PIN W9 [get_ports {B}]
set_property PACKAGE_PIN Y7 [get_ports {C}]
set_property PACKAGE_PIN Y8 [get_ports {D}]

set_property PACKAGE_PIN A21 [get_ports {Y0}]
set_property PACKAGE_PIN E22 [get_ports {Y1}]
set_property PACKAGE_PIN D22 [get_ports {Y2}]
set_property PACKAGE_PIN E21 [get_ports {Y3}]
set_property PACKAGE_PIN D21 [get_ports {Y4}]
set_property PACKAGE_PIN G21 [get_ports {Y5}]
set_property PACKAGE_PIN G22 [get_ports {Y6}]
set_property PACKAGE_PIN F21 [get_ports {Y7}]
set_property PACKAGE_PIN J17 [get_ports {Y8}]
set_property PACKAGE_PIN L14 [get_ports {Y9}]
set_property PACKAGE_PIN L15 [get_ports {Y10}]
set_property PACKAGE_PIN L16 [get_ports {Y11}]
set_property PACKAGE_PIN K16 [get_ports {Y12}]
set_property PACKAGE_PIN M15 [get_ports {Y13}]
set_property PACKAGE_PIN M16 [get_ports {Y14}]
set_property PACKAGE_PIN M17 [get_ports {Y15}]
```

-

- *The testing result (provide the screen shots (at least 3 testing scene) to show state of inputs and outputs along with the related descriptions.*
- 

- *0000, default, the first is on*

- 
- *0111, the 8th is on*
- 
- *0101, the sixth is on*

THE DESCRIPTION OF OPERATION

(This part is always REQUIRED unless you can get full marks for this lab task)

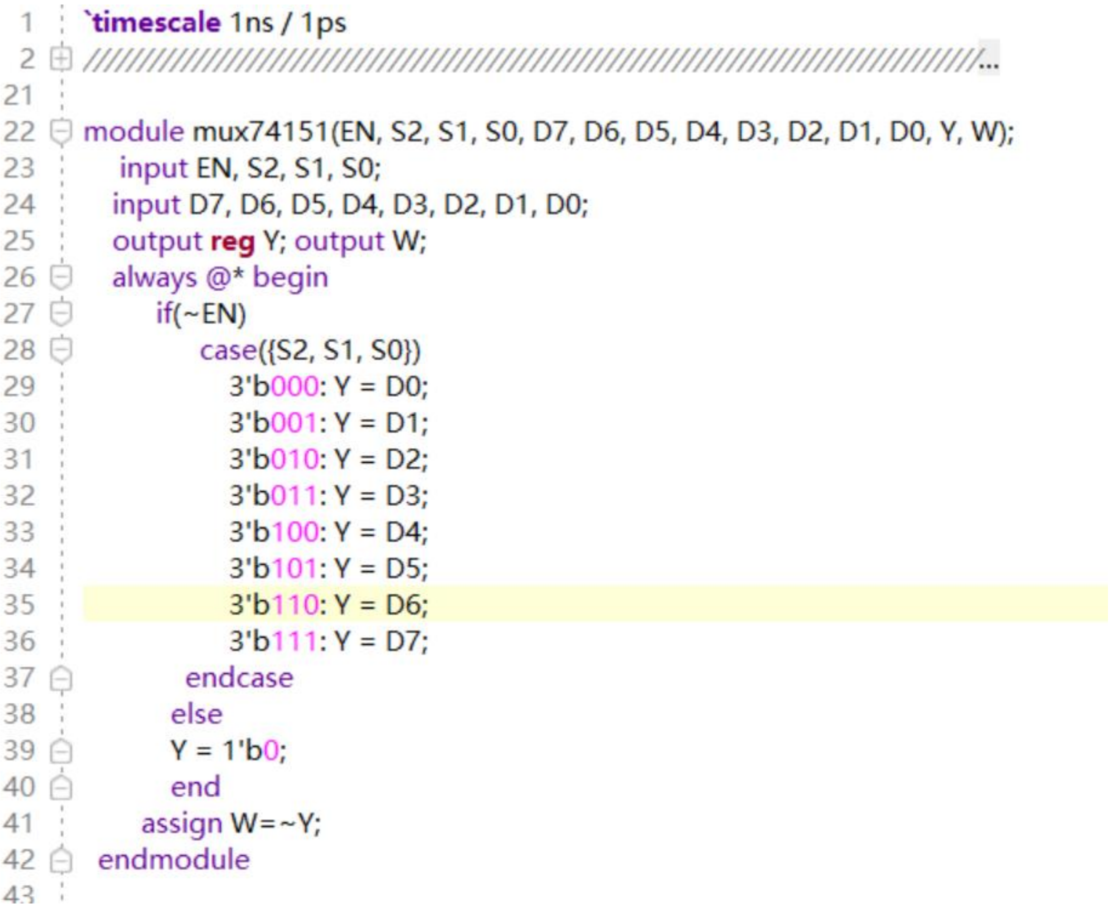
Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*
- *I don’t have enough knowledge for decoder and I failed to design the testbench when I was doing this experiment.*
- *I studied combinational logic and asked my friends for help, and they gave me ideas.*

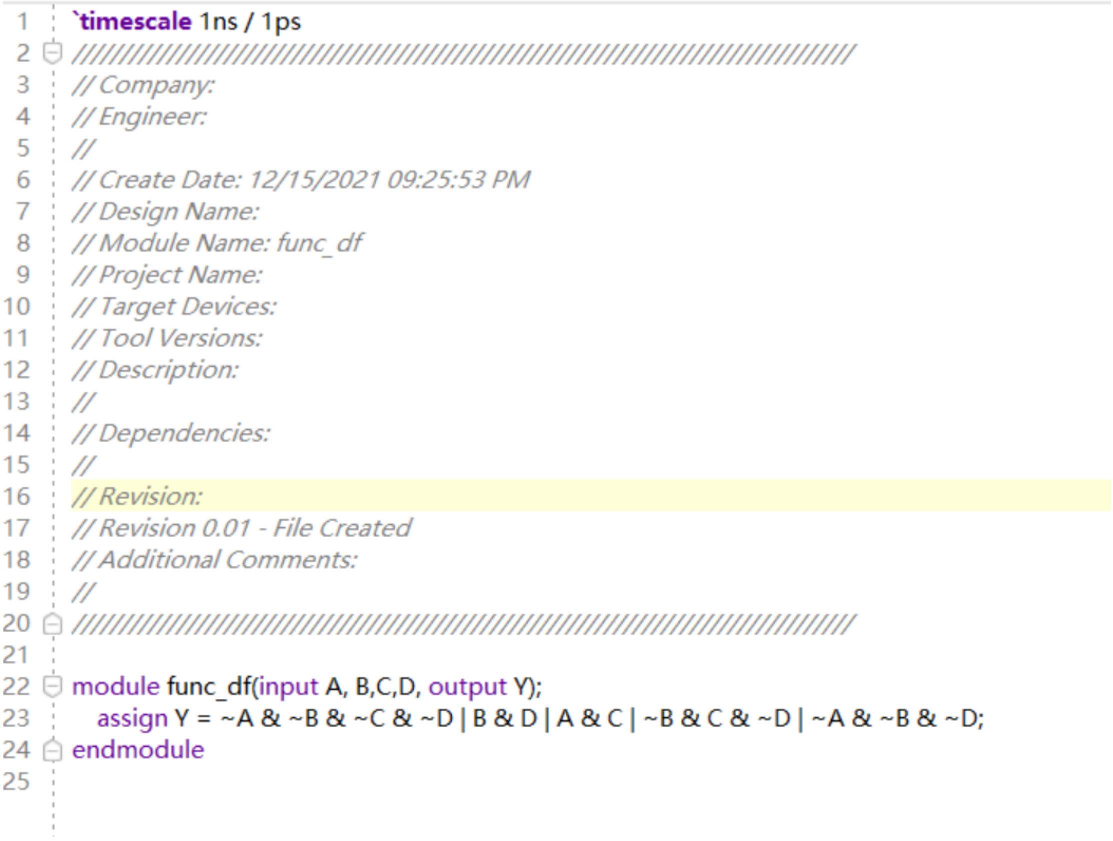
PART 2: DIGITAL DESIGN LAB (TASK2)

DESIGN

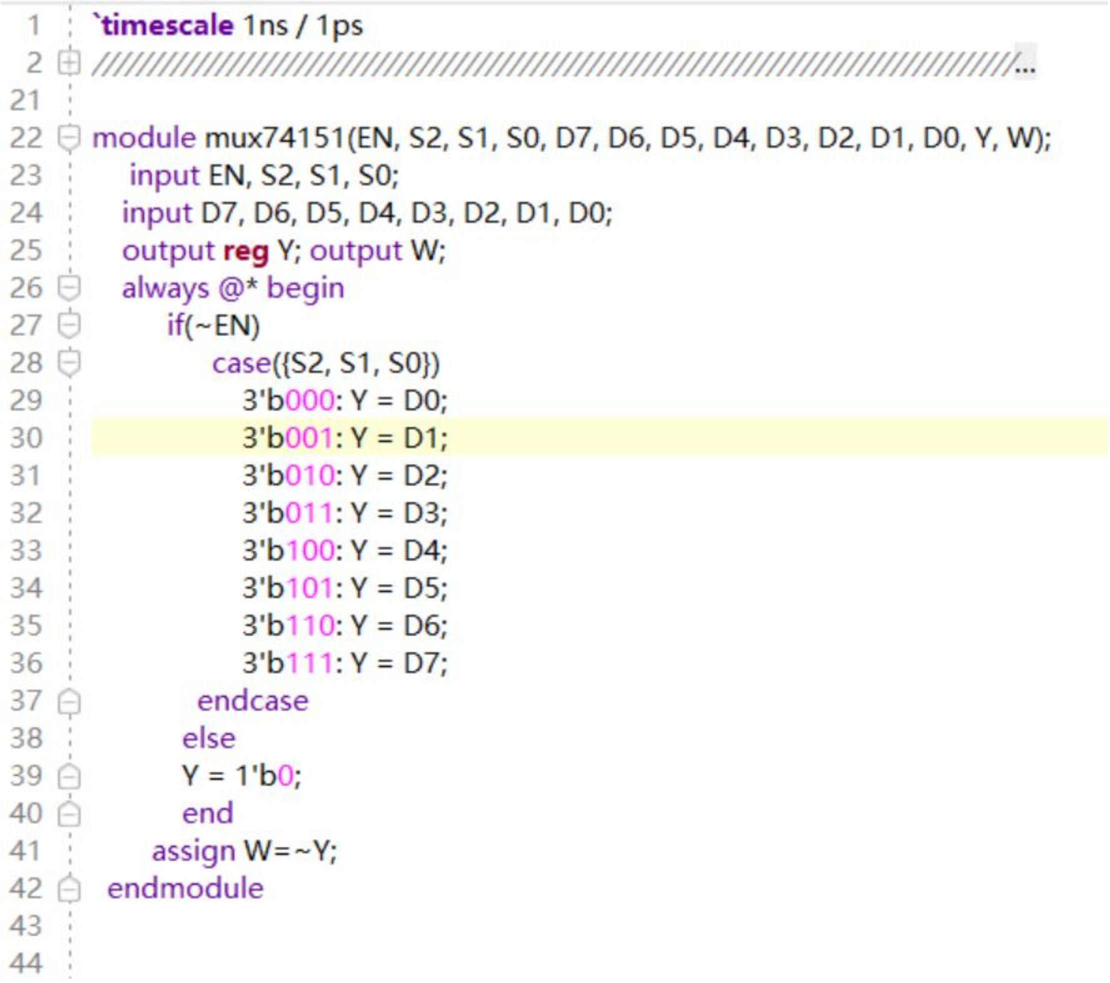
Describe the design of your system by providing the following information:



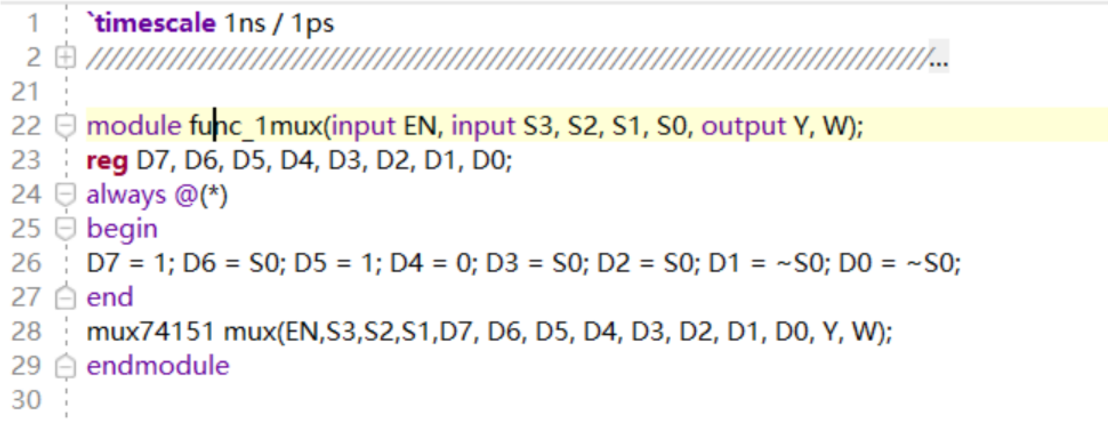
```
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////////////////////...

module mux74151(EN, S2, S1, S0, D7, D6, D5, D4, D3, D2, D1, D0, Y, W);
     input EN, S2, S1, S0;
    input D7, D6, D5, D4, D3, D2, D1, D0;
    output reg Y; output W;
    always @* begin
          if(~EN)
                case({S2, S1, S0})
                    3'b000: Y = D0;
                    3'b001: Y = D1;
                    3'b010: Y = D2;
                    3'b011: Y = D3;
                    3'b100: Y = D4;
                    3'b101: Y = D5;
                    3'b110: Y = D6;
                    3'b111: Y = D7;
              endcase
            else
            Y = 1'b0;
            end
        assign W=~Y;
  endmodule

```



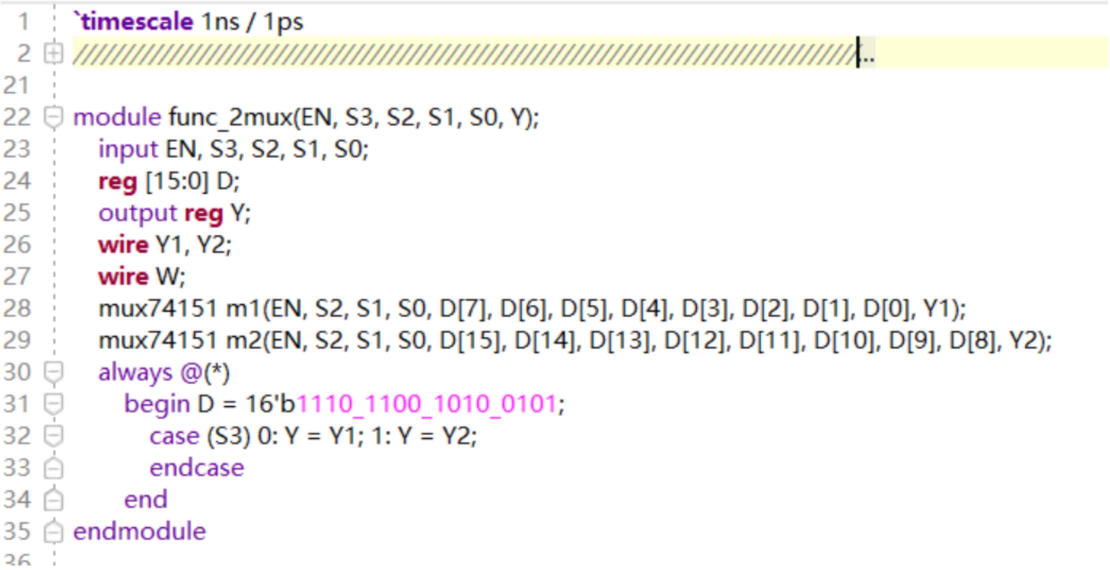
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 12/15/2021 09:25:53 PM
// Design Name:
// Module Name: func_df
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module func_df(input A, B,C,D, output Y);
   assign Y = ~A & ~B & ~C & ~D | B & D | A & C | ~B & C & ~D | ~A & ~B & ~D;
endmodule

```



```
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////////////////////...

module mux74151(EN, S2, S1, S0, D7, D6, D5, D4, D3, D2, D1, D0, Y, W);
     input EN, S2, S1, S0;
    input D7, D6, D5, D4, D3, D2, D1, D0;
    output reg Y; output W;
    always @* begin
        if(~EN)
            case({S2, S1, S0})
                3'b000: Y = D0;
                3'b001: Y = D1;
                3'b010: Y = D2;
                3'b011: Y = D3;
                3'b100: Y = D4;
                3'b101: Y = D5;
                3'b110: Y = D6;
                3'b111: Y = D7;
           endcase
          else
          Y = 1'b0;
          end
       assign W=~Y;
  endmodule


```



```
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////////////////...

module func_1mux(input EN, input S3, S2, S1, S0, output Y, W);
reg D7, D6, D5, D4, D3, D2, D1, D0;
always @(*)
begin
D7 = 1; D6 = S0; D5 = 1; D4 = 0; D3 = S0; D2 = S0; D1 = ~S0; D0 = ~S0;
end
mux74151 mux(EN,S3,S2,S1,D7, D6, D5, D4, D3, D2, D1, D0, Y, W);
endmodule

```

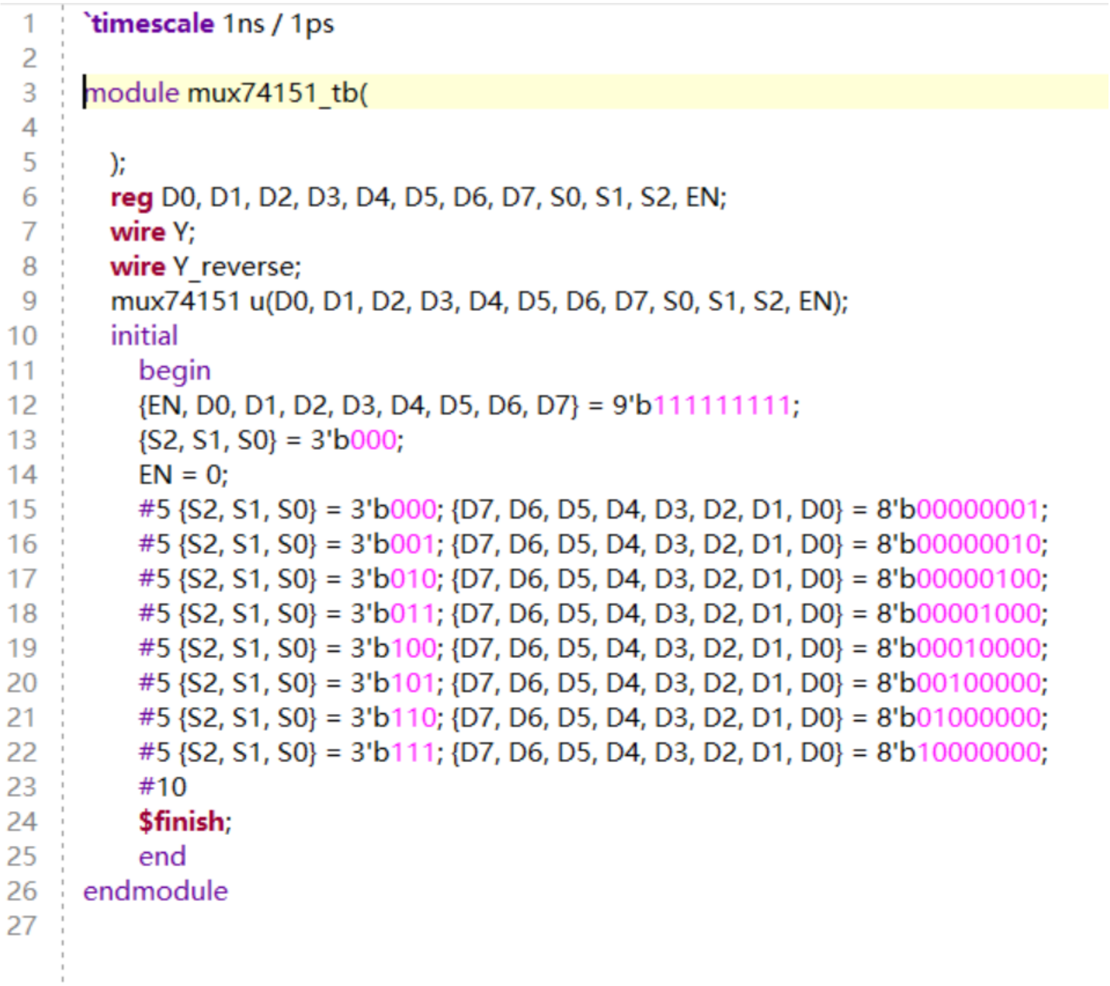


```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////..

module func_2mux(EN, S3, S2, S1, S0, Y);
   input EN, S3, S2, S1, S0;
   reg [15:0] D;
   output reg Y;
   wire Y1, Y2;
   wire W;
   mux74151 m1(EN, S2, S1, S0, D[7], D[6], D[5], D[4], D[3], D[2], D[1], D[0], Y1);
   mux74151 m2(EN, S2, S1, S0, D[15], D[14], D[13], D[12], D[11], D[10], D[9], D[8], Y2);
   always @(*)
      begin D = 16'b1110_1100_1010_0101;
         case (S3) 0: Y = Y1; 1: Y = Y2;
         endcase
      end
endmodule
```

SIMULATION

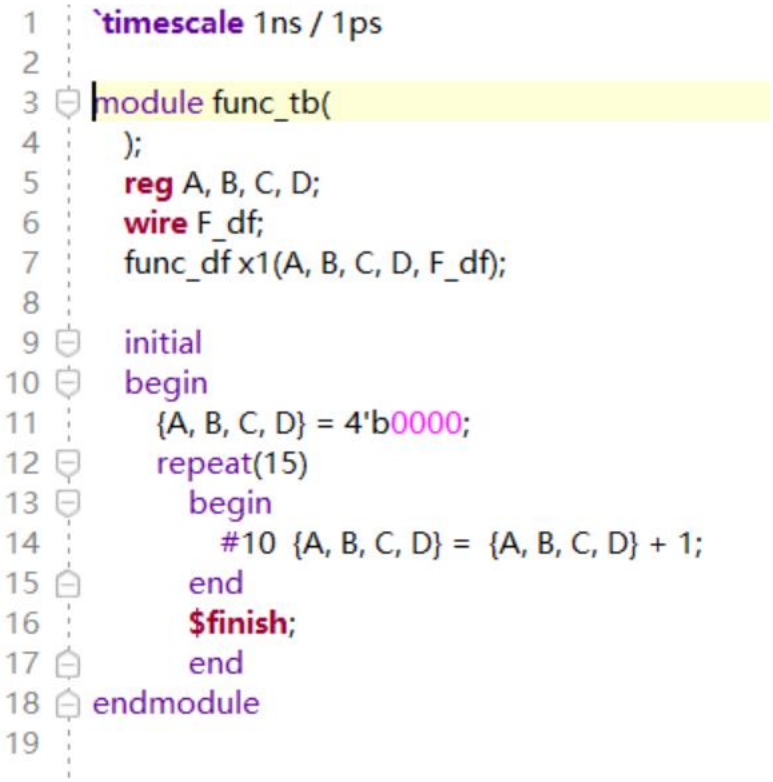
Describe how you build the test bench and do the simulation.

- *Using Verilog (provide the Verilog code)*



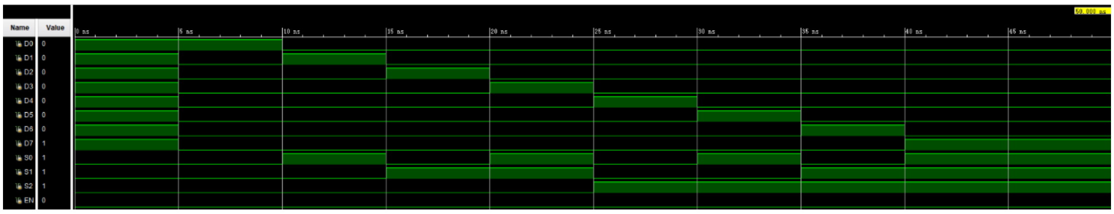
```
`timescale 1ns / 1ps

module mux74151_tb(

  );
  reg D0, D1, D2, D3, D4, D5, D6, D7, S0, S1, S2, EN;
  wire Y;
  wire Y_reverse;
  mux74151 u(D0, D1, D2, D3, D4, D5, D6, D7, S0, S1, S2, EN);
  initial
    begin
    {EN, D0, D1, D2, D3, D4, D5, D6, D7} = 9'b111111111;
    {S2, S1, S0} = 3'b000;
    EN = 0;
    #5 {S2, S1, S0} = 3'b000; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00000001;
    #5 {S2, S1, S0} = 3'b001; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00000010;
    #5 {S2, S1, S0} = 3'b010; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00000100;
    #5 {S2, S1, S0} = 3'b011; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00001000;
    #5 {S2, S1, S0} = 3'b100; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00010000;
    #5 {S2, S1, S0} = 3'b101; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b00100000;
    #5 {S2, S1, S0} = 3'b110; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b01000000;
    #5 {S2, S1, S0} = 3'b111; {D7, D6, D5, D4, D3, D2, D1, D0} = 8'b10000000;
    #10
    $finish;
    end
endmodule
```

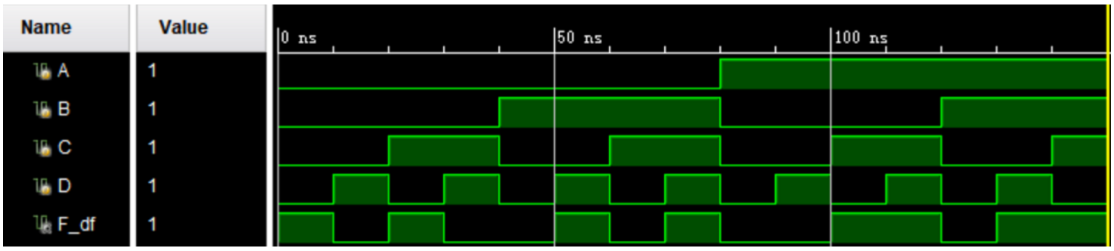
-



```
`timescale 1ns / 1ps

module func_tb(
  );
  reg A, B, C, D;
  wire F_df;
  func_df x1(A, B, C, D, F_df);

  initial
  begin
    {A, B, C, D} = 4'b0000;
    repeat(15)
      begin
        #10  {A, B, C, D} =  {A, B, C, D} + 1;
      end
      $finish;
      end
endmodule
```

-

- *Wave form of simulation result (provide screen shots)*
- *MUX 74151*
- 

- *Func_tb for dataflow(the other two are the same with data flow)*
- 

- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation*
- First, when EN=1, mux is deactivatedThen, when EN=0, mux is activated, 8 cases are tested. When {S2,S1,S0}=3'B111, only D7=1, D0~D6=0, {D7,D6,D5,D4,D3,D2,D1,D0} =8'B1000_0000.The selection works, that it selected the corresponding high level bit in d0~d7.

THE DESCRIPTION OF OPERATION

Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*
- *At first I have no idea how to deal with the function with two 74151;*
- *I asked my classmates and checked the lab slides, and I finally managed to do it.*