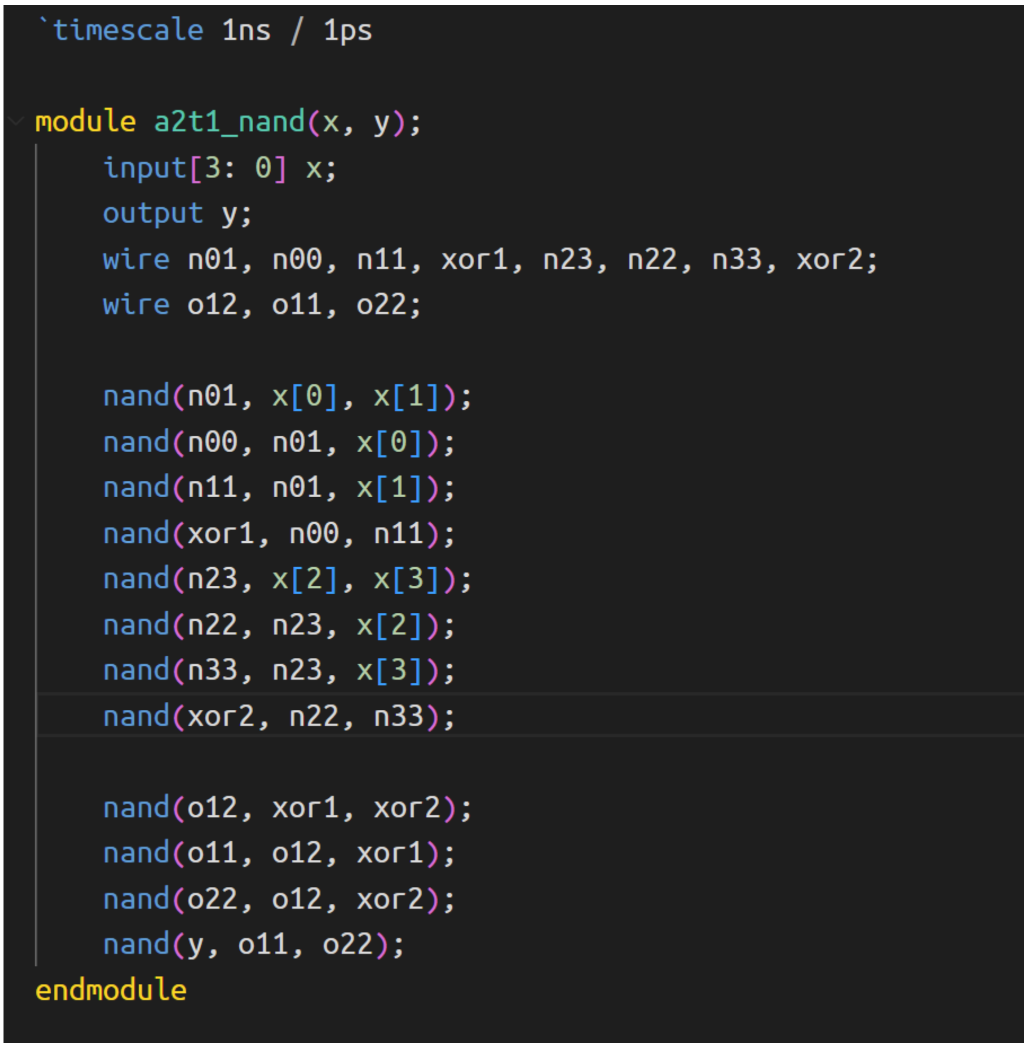
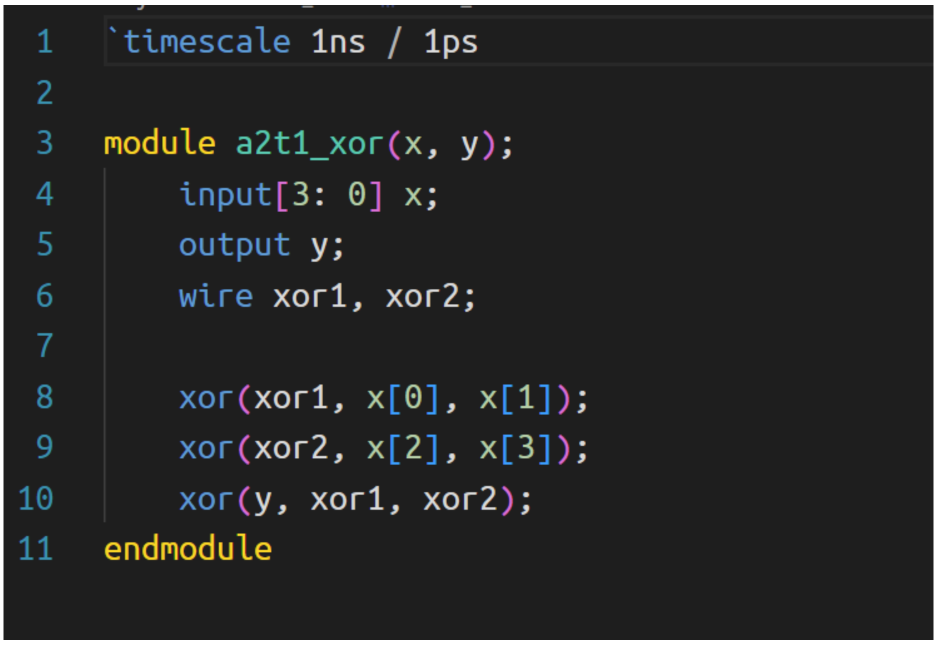
PART 1: DIGITAL DESIGN THEORY

PART 2: DIGITAL DESIGN LAB (TASK1)

1. Design in structure descriptions styles by using xor and nand gates respectively

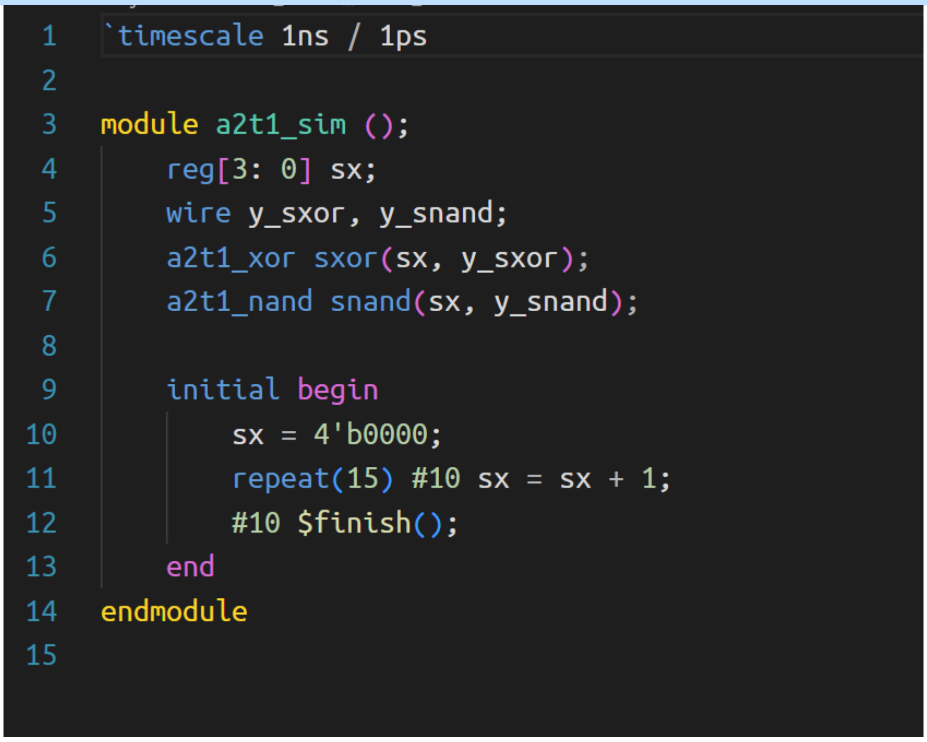


```
`timescale 1ns / 1ps

module a2t1_nand(x, y);
    input[3: 0] x;
    output y;
    wire n01, n00, n11, xor1, n23, n22, n33, xor2;
    wire o12, o11, o22;

    nand(n01, x[0], x[1]);
    nand(n00, n01, x[0]);
    nand(n11, n01, x[1]);
    nand(xor1, n00, n11);
    nand(n23, x[2], x[3]);
    nand(n22, n23, x[2]);
    nand(n33, n23, x[3]);
    nand(xor2, n22, n33);

    nand(o12, xor1, xor2);
    nand(o11, o12, xor1);
    nand(o22, o12, xor2);
    nand(y, o11, o22);
endmodule
```

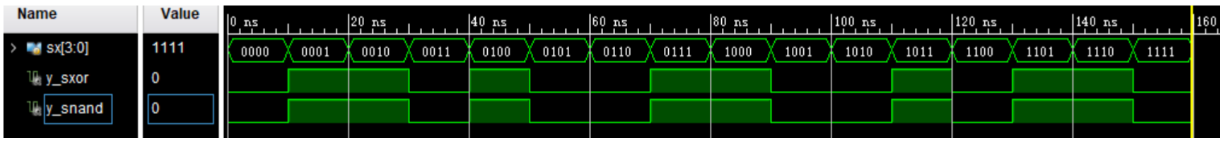


```verilog
`timescale 1ns / 1ps

module a2t1_xor(x, y);
    input[3: 0] x;
    output y;
    wire xor1, xor2;

    xor(xor1, x[0], x[1]);
    xor(xor2, x[2], x[3]);
    xor(y, xor1, xor2);
endmodule
```

2. Test bench in Verilog, simulation result and its description



```verilog
`timescale 1ns / 1ps

module a2t1_sim ();
    reg[3: 0] sx;
    wire y_sxor, y_snand;
    a2t1_xor sxor(sx, y_sxor);
    a2t1_nand snand(sx, y_snand);

    initial begin
        sx = 4'b0000;
        repeat(15) #10 sx = sx + 1;
        #10 $finish();
    end
endmodule

```

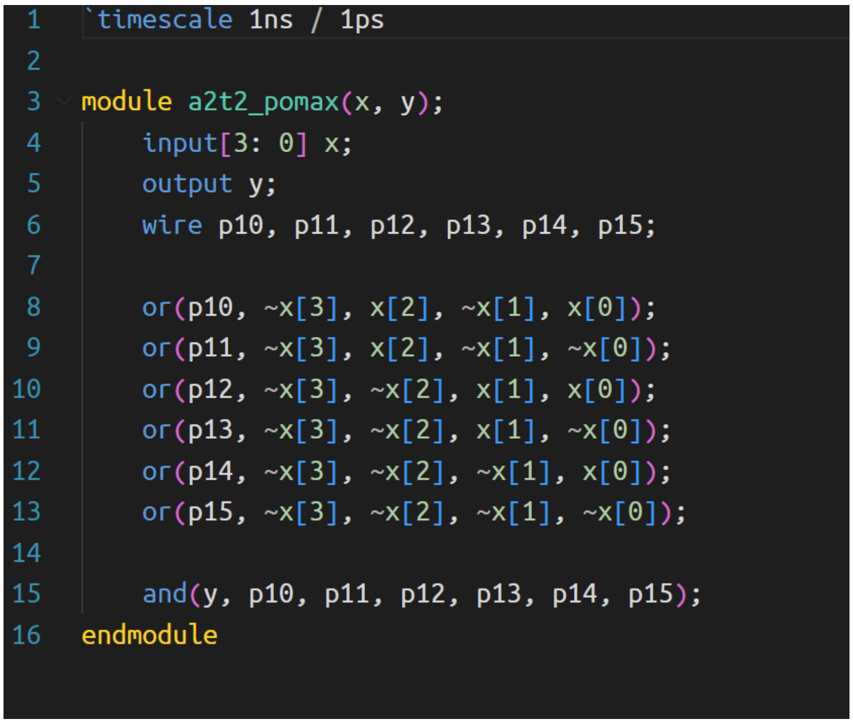


Name	Value
sx[3:0]	1111
y_sxor	0
y_snand	0

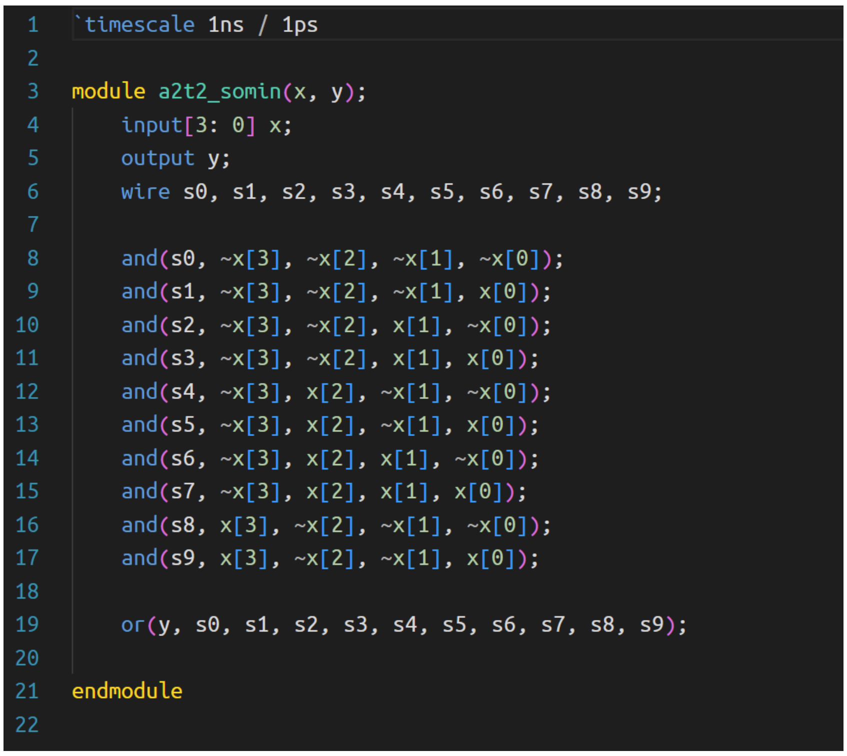
The result is same as we expected. The function of the two designs meets the expectation.

PART 2: DIGITAL DESIGN LAB (TASK2)

1. Design in structure descriptions styles by using primitive gates on Sum-of-Minterms and Product-of-Maxterms respectively

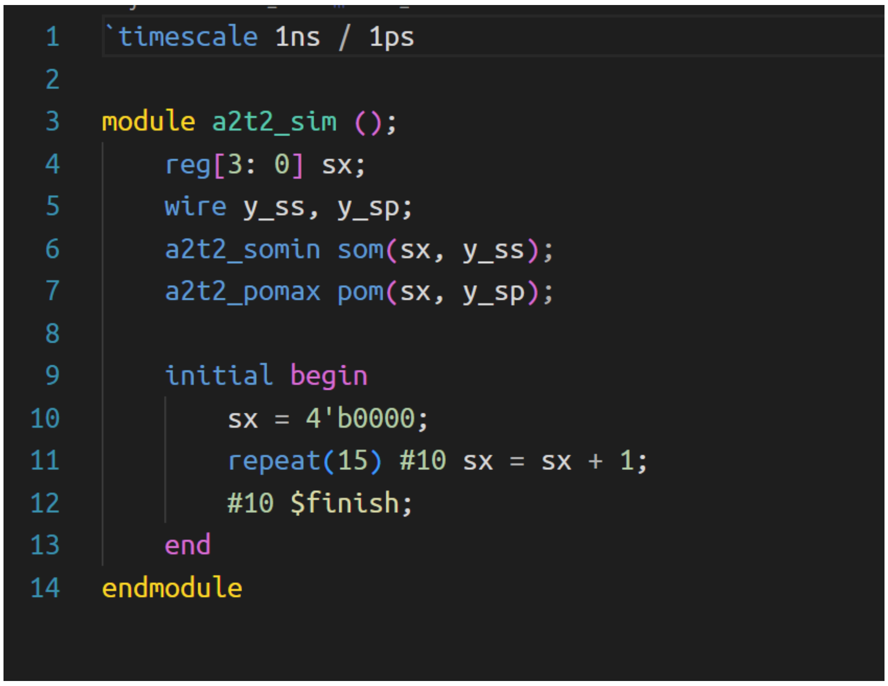


```
`timescale 1ns / 1ps

module a2t2_pomax(x, y);
    input[3: 0] x;
    output y;
    wire p10, p11, p12, p13, p14, p15;

    or(p10, ~x[3], x[2], ~x[1], x[0]);
    or(p11, ~x[3], x[2], ~x[1], ~x[0]);
    or(p12, ~x[3], ~x[2], x[1], x[0]);
    or(p13, ~x[3], ~x[2], x[1], ~x[0]);
    or(p14, ~x[3], ~x[2], ~x[1], x[0]);
    or(p15, ~x[3], ~x[2], ~x[1], ~x[0]);

    and(y, p10, p11, p12, p13, p14, p15);
endmodule
```

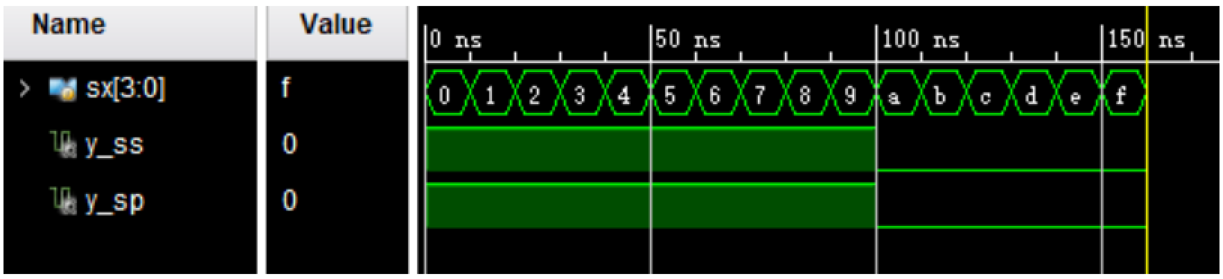


```
`timescale 1ns / 1ps

module a2t2_somin(x, y);
    input[3: 0] x;
    output y;
    wire s0, s1, s2, s3, s4, s5, s6, s7, s8, s9;

    and(s0, ~x[3], ~x[2], ~x[1], ~x[0]);
    and(s1, ~x[3], ~x[2], ~x[1], x[0]);
    and(s2, ~x[3], ~x[2], x[1], ~x[0]);
    and(s3, ~x[3], ~x[2], x[1], x[0]);
    and(s4, ~x[3], x[2], ~x[1], ~x[0]);
    and(s5, ~x[3], x[2], ~x[1], x[0]);
    and(s6, ~x[3], x[2], x[1], ~x[0]);
    and(s7, ~x[3], x[2], x[1], x[0]);
    and(s8, x[3], ~x[2], ~x[1], ~x[0]);
    and(s9, x[3], ~x[2], ~x[1], x[0]);

    or(y, s0, s1, s2, s3, s4, s5, s6, s7, s8, s9);

endmodule

```

2. Test bench in Verilog, simulation result and its description



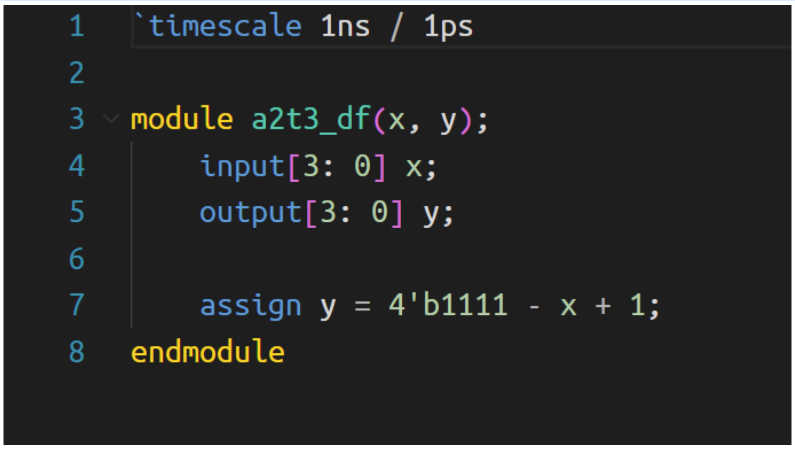
```
`timescale 1ns / 1ps

module a2t2_sim ();
    reg[3: 0] sx;
    wire y_ss, y_sp;
    a2t2_somin som(sx, y_ss);
    a2t2_pomax pom(sx, y_sp);

    initial begin
        sx = 4'b0000;
        repeat(15) #10 sx = sx + 1;
        #10 $finish;
    end
endmodule
```



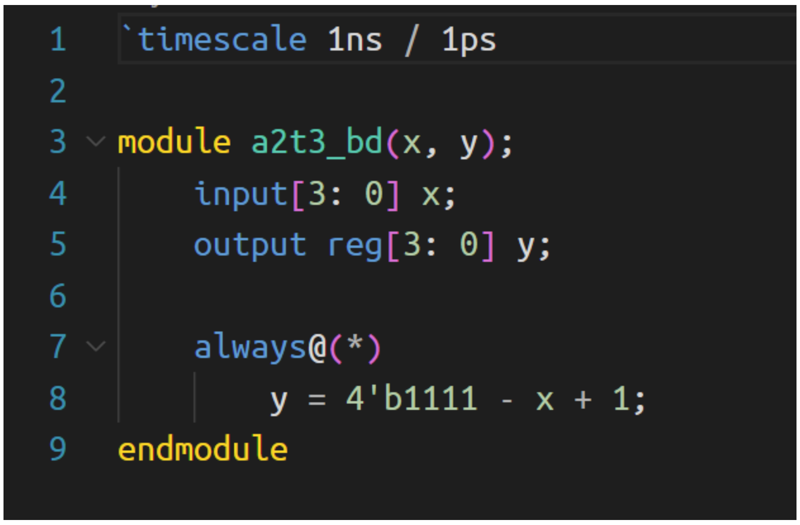
The result is same as we expected. 0-9 is valid. The function of the two designs meets the expectation.

PART 2: DIGITAL DESIGN LAB (TASK3)

1. Design in Verilog in data flow style and behavioral description style.

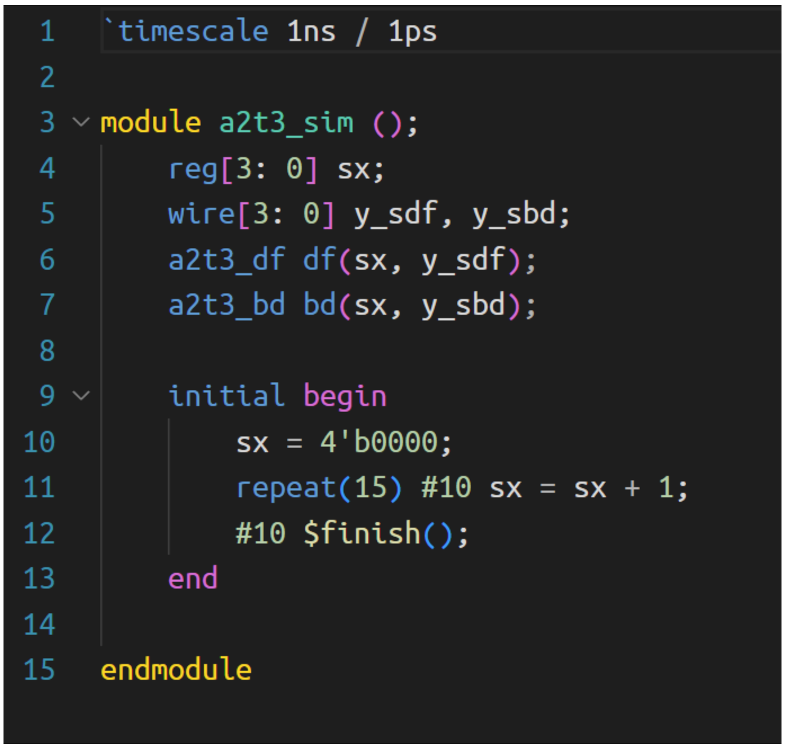


```
`timescale 1ns / 1ps

module a2t3_df(x, y);
    input[3: 0] x;
    output[3: 0] y;

    assign y = 4'b1111 - x + 1;
endmodule
```

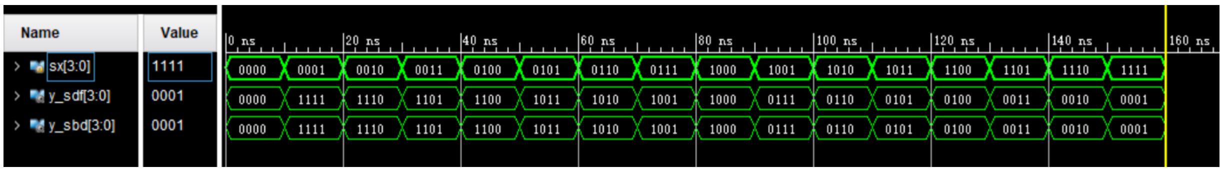


```verilog
`timescale 1ns / 1ps

module a2t3_bd(x, y);
    input[3: 0] x;
    output reg[3: 0] y;

    always@(*)
        y = 4'b1111 - x + 1;
endmodule
```

2. Test bench in Verilog, simulation result and its description



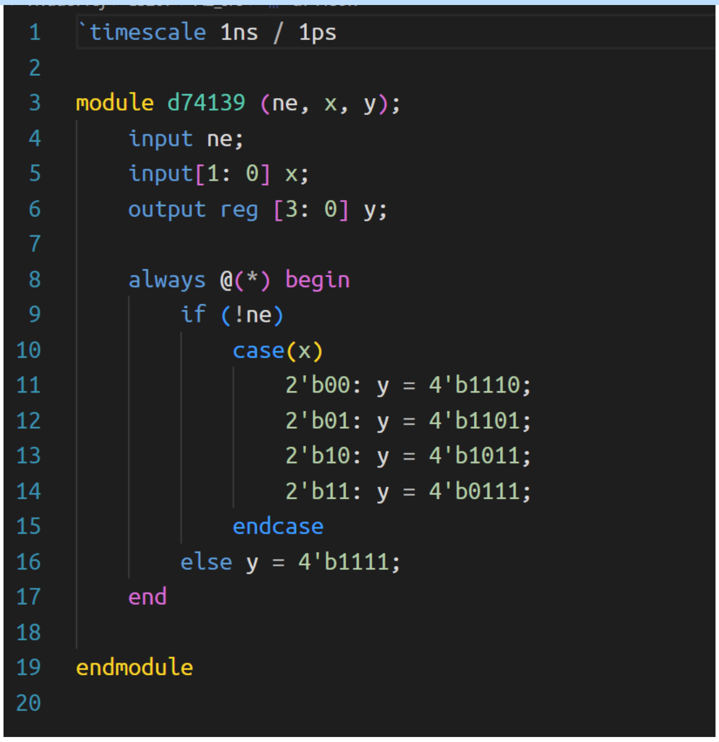
```verilog
`timescale 1ns / 1ps

module a2t3_sim ();
    reg[3: 0] sx;
    wire[3: 0] y_sdf, y_sbd;
    a2t3_df df(sx, y_sdf);
    a2t3_bd bd(sx, y_sbd);

    initial begin
        sx = 4'b0000;
        repeat(15) #10 sx = sx + 1;
        #10 $finish();
    end

endmodule
```



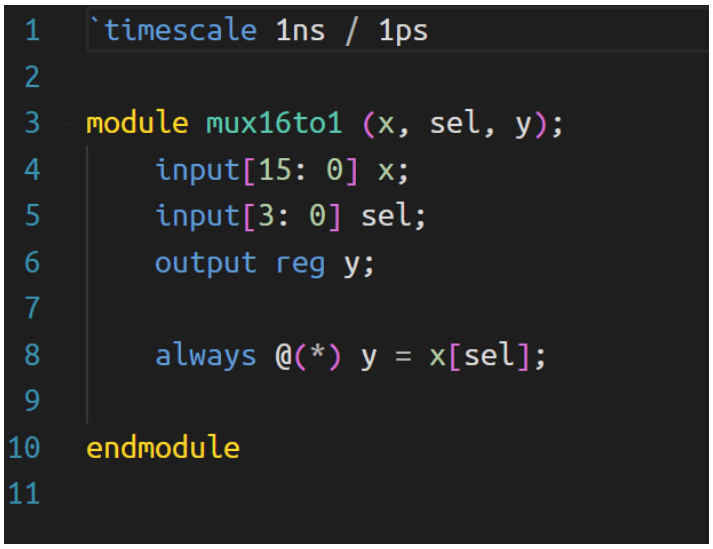
The result is same as we expected. The function of the two designs meets the expectation.

PART 2: DIGITAL DESIGN LAB (TASK4)

1. Design the Decoder(74139) and 16-to-1 MUX which would be used in the circuit

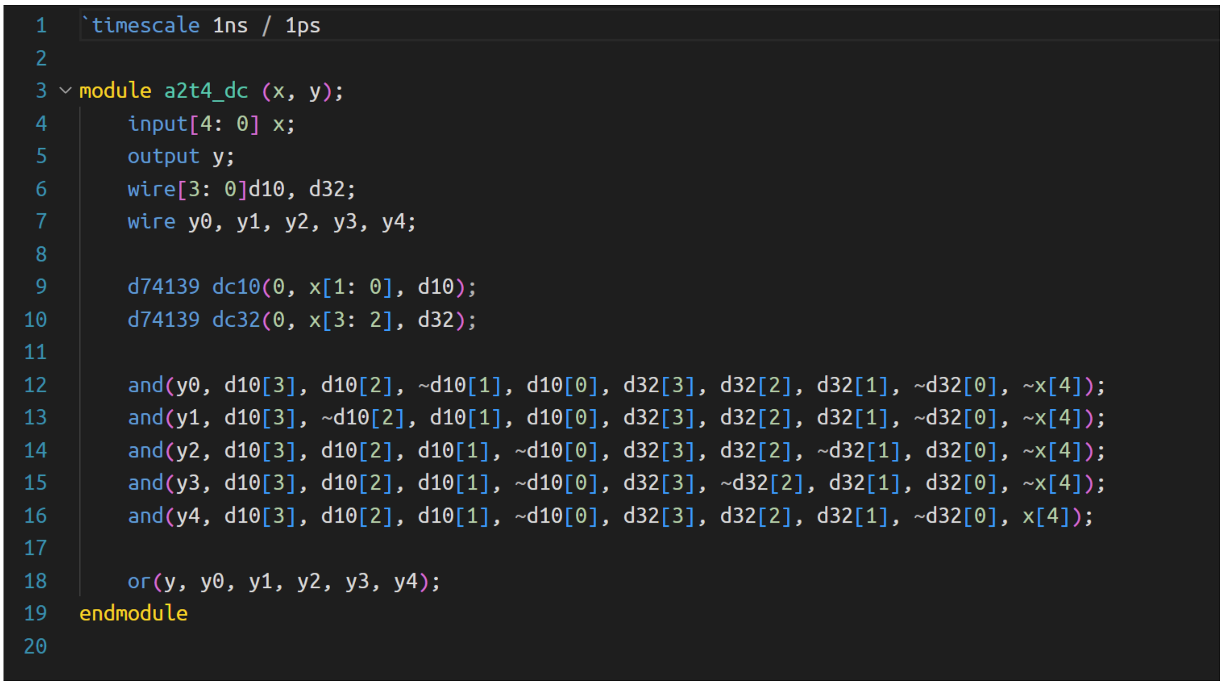


```
`timescale 1ns / 1ps

module d74139 (ne, x, y);
    input ne;
    input[1: 0] x;
    output reg [3: 0] y;

    always @(*) begin
        if (!ne)
            case(x)
                2'b00: y = 4'b1110;
                2'b01: y = 4'b1101;
                2'b10: y = 4'b1011;
                2'b11: y = 4'b0111;
            endcase
        else y = 4'b1111;
    end

endmodule
```

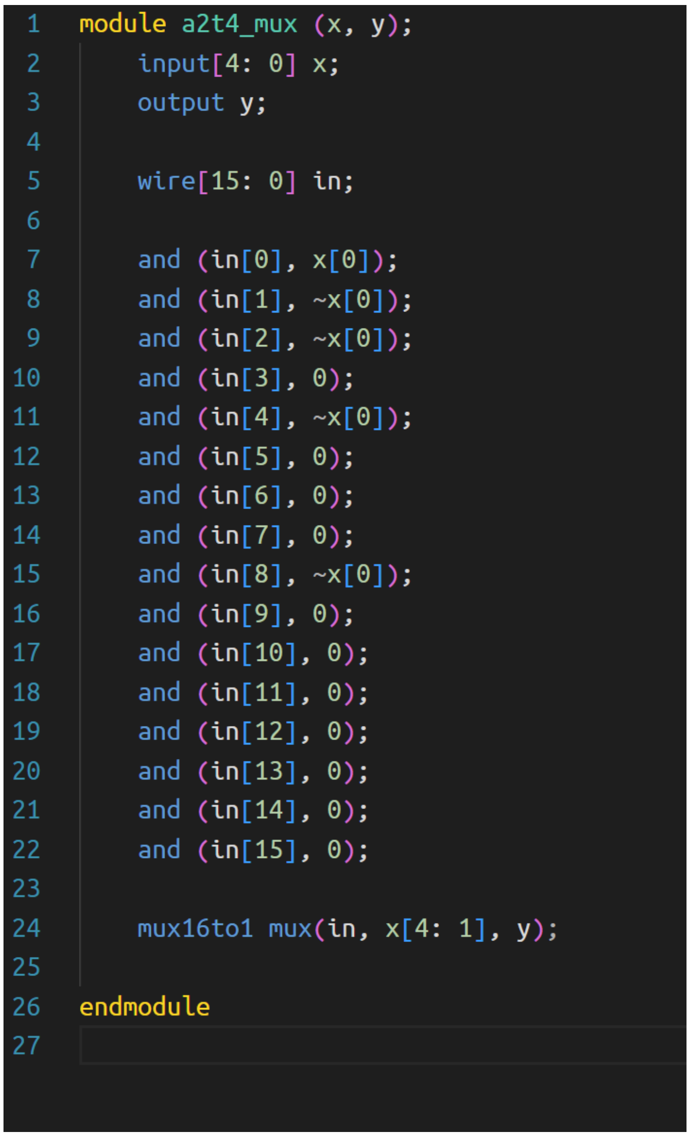


```
`timescale 1ns / 1ps

module mux16to1 (x, sel, y);
    input[15: 0] x;
    input[3: 0] sel;
    output reg y;

    always @(*) y = x[sel];

endmodule
```

2. Design the circuit in structure description style by using the Decoder(74139) and 16-to-1 MUX.

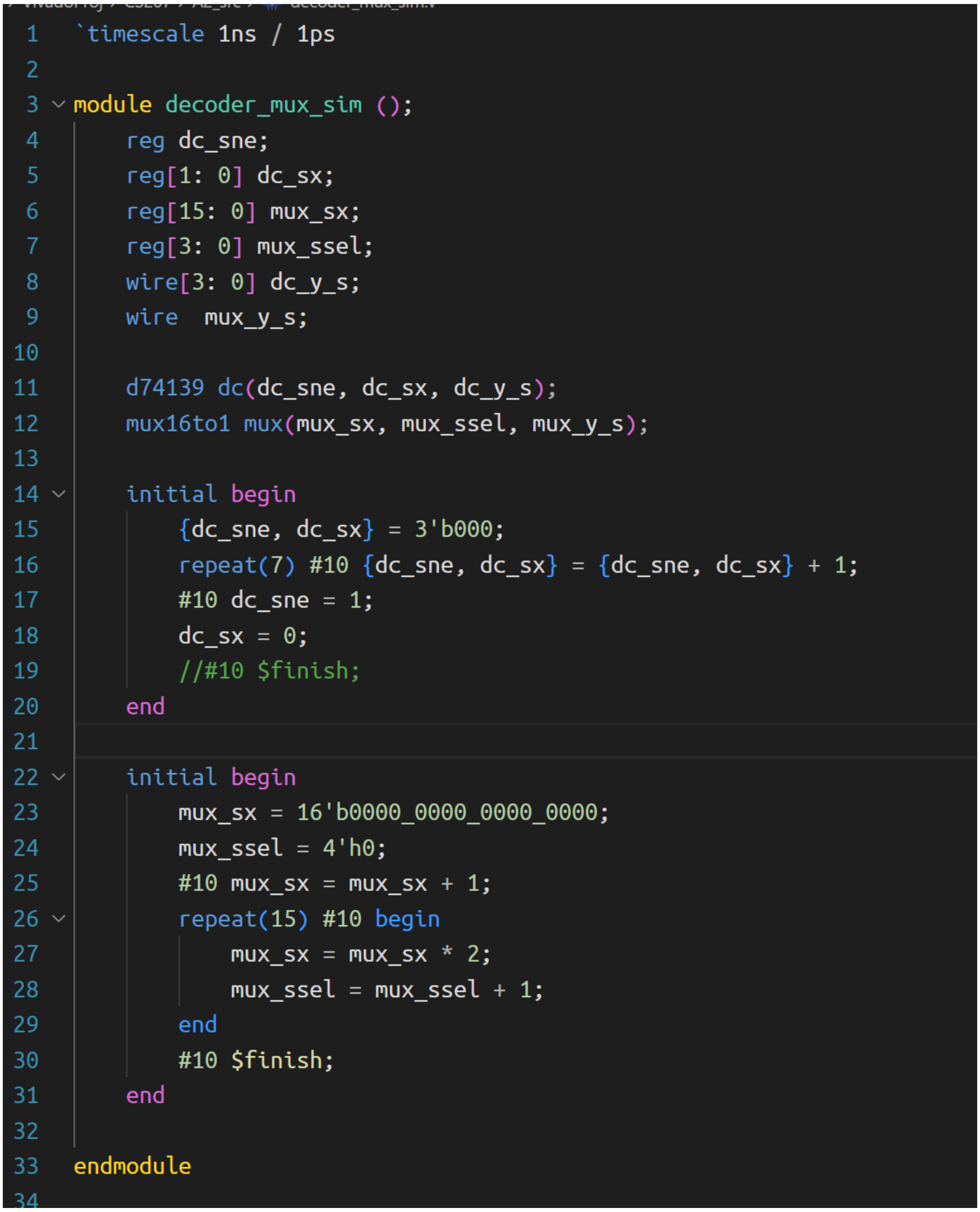


```verilog
`timescale 1ns / 1ps

module a2t4_dc (x, y);
    input[4: 0] x;
    output y;
    wire[3: 0]d10, d32;
    wire y0, y1, y2, y3, y4;

    d74139 dc10(0, x[1: 0], d10);
    d74139 dc32(0, x[3: 2], d32);

    and(y0, d10[3], d10[2], ~d10[1], d10[0], d32[3], d32[2], d32[1], ~d32[0], ~x[4]);
    and(y1, d10[3], ~d10[2], d10[1], d10[0], d32[3], d32[2], d32[1], ~d32[0], ~x[4]);
    and(y2, d10[3], d10[2], d10[1], ~d10[0], d32[3], d32[2], ~d32[1], d32[0], ~x[4]);
    and(y3, d10[3], d10[2], d10[1], ~d10[0], d32[3], ~d32[2], d32[1], d32[0], ~x[4]);
    and(y4, d10[3], d10[2], d10[1], ~d10[0], d32[3], d32[2], d32[1], ~d32[0], x[4]);

    or(y, y0, y1, y2, y3, y4);
endmodule

```

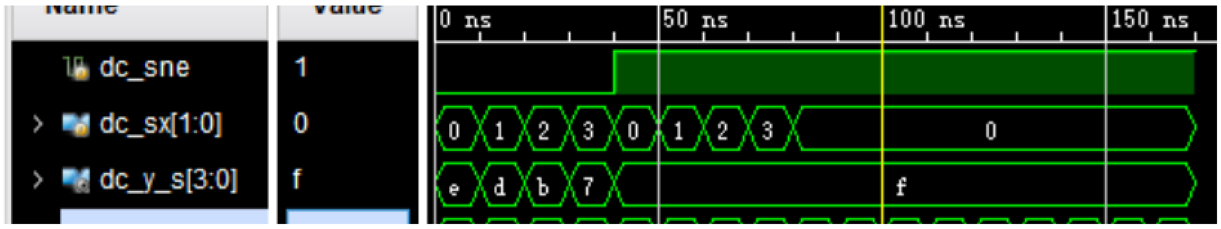


```verilog
module a2t4_mux (x, y);
    input[4: 0] x;
    output y;

    wire[15: 0] in;

    and (in[0], x[0]);
    and (in[1], ~x[0]);
    and (in[2], ~x[0]);
    and (in[3], 0);
    and (in[4], ~x[0]);
    and (in[5], 0);
    and (in[6], 0);
    and (in[7], 0);
    and (in[8], ~x[0]);
    and (in[9], 0);
    and (in[10], 0);
    and (in[11], 0);
    and (in[12], 0);
    and (in[13], 0);
    and (in[14], 0);
    and (in[15], 0);

    mux16to1 mux(in, x[4: 1], y);

endmodule

```

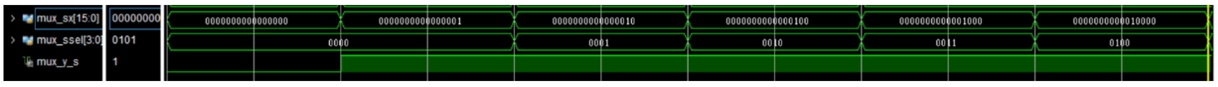
3. Test bench to verify the function of the Decoder(74139) and 16-to-1 MUX in Verilog, simulation result and its description.

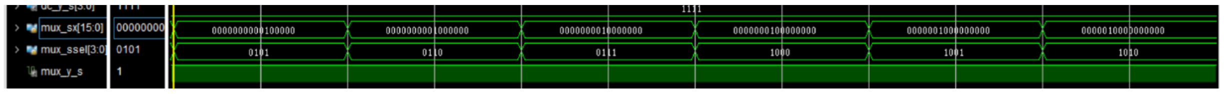


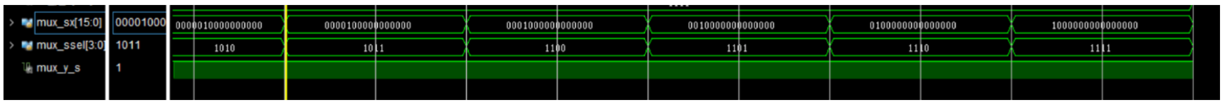
```
`timescale 1ns / 1ps

module decoder_mux_sim ();
    reg dc_sne;
    reg[1: 0] dc_sx;
    reg[15: 0] mux_sx;
    reg[3: 0] mux_ssel;
    wire[3: 0] dc_y_s;
    wire  mux_y_s;

    d74139 dc(dc_sne, dc_sx, dc_y_s);
    mux16to1 mux(mux_sx, mux_ssel, mux_y_s);

    initial begin
        {dc_sne, dc_sx} = 3'b000;
        repeat(7) #10 {dc_sne, dc_sx} = {dc_sne, dc_sx} + 1;
        #10 dc_sne = 1;
        dc_sx = 0;
        //#10 $finish;
    end

    initial begin
        mux_sx = 16'b0000_0000_0000_0000;
        mux_ssel = 4'h0;
        #10 mux_sx = mux_sx + 1;
        repeat(15) #10 begin
            mux_sx = mux_sx * 2;
            mux_ssel = mux_ssel + 1;
        end
        #10 $finish;
    end

endmodule
```



In simulation of decoder, I first set EN to be 0, then the decoder works: 00: e=1110, 01: d=1101, 10: b=1011, 11: 7=1111, then I set EN to be 1, the decoder doesn't work for y=1111



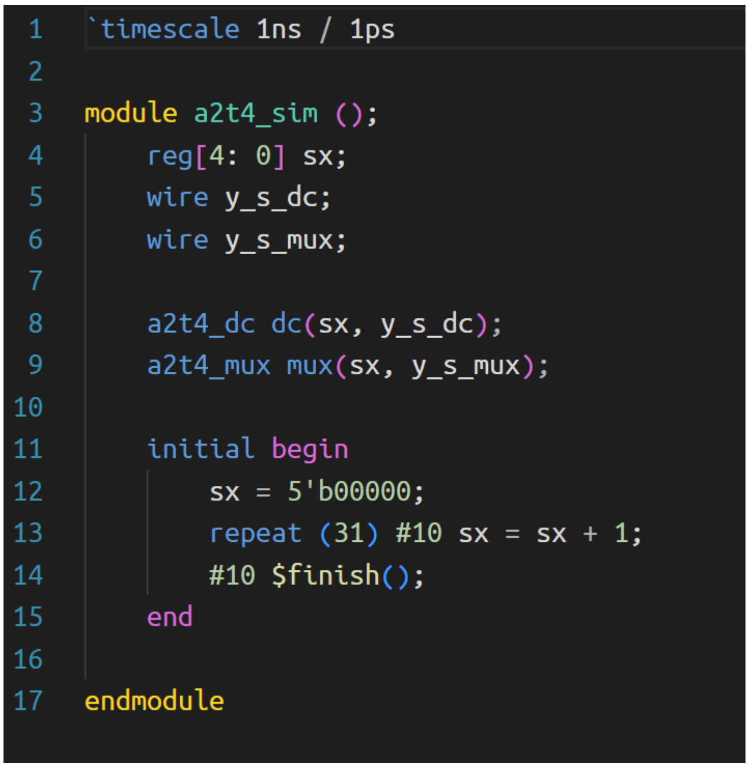




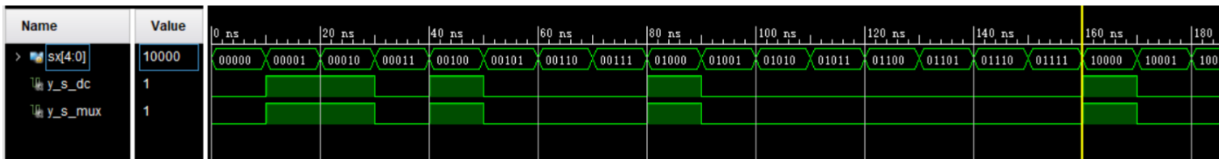
In simulation of MUX, I first set all digits of x to be 0, then y would be 0. Then set sel from 0001 to 1111, set x 1 from MSB to LSB, then all y would be 1.

The result is same as we expected. The function of the two designs meets the expectation.

4. Test bench to verify the function of the circuit, simulation result and its description

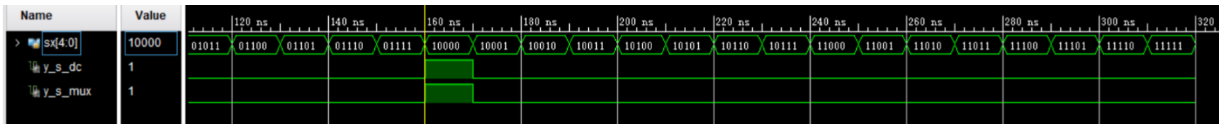


```verilog
`timescale 1ns / 1ps

module a2t4_sim ();
    reg[4: 0] sx;
    wire y_s_dc;
    wire y_s_mux;

    a2t4_dc dc(sx, y_s_dc);
    a2t4_mux mux(sx, y_s_mux);

    initial begin
        sx = 5'b00000;
        repeat (31) #10 sx = sx + 1;
        #10 $finish();
    end

endmodule
```



Name	Value
sx[4:0]	10000
y_s_dc	1
y_s_mux	1

```
00000 00001 00010 00011 00100 00101 00110 00111 01000 01001 01010 01011 01100 01101 01110 01111 10000 10001 100
```



Name	Value
sx[4:0]	10000
y_s_dc	1
y_s_mux	1

```
01011 01100 01101 01110 01111 10000 10001 10010 10011 10100 10101 10110 10111 11000 11001 11010 11011 11100 11101 11110 11111
```

The result is same as we expected. The function of the two designs meets the expectation.