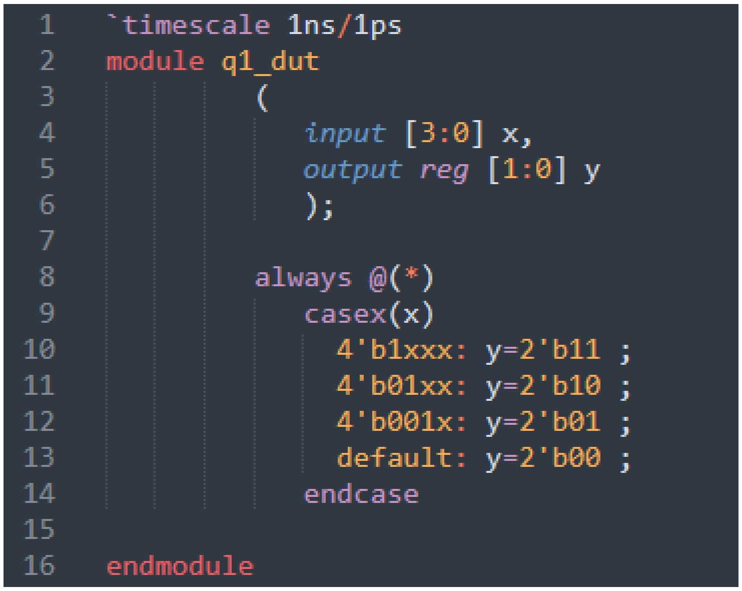
Assignment 2

By Amr Mohamed Ahmed Ebrahim

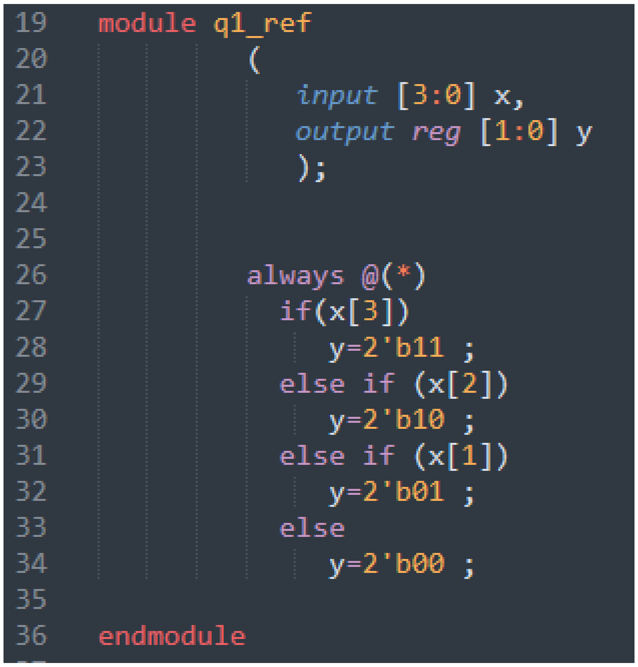
Question 1

DUT Code



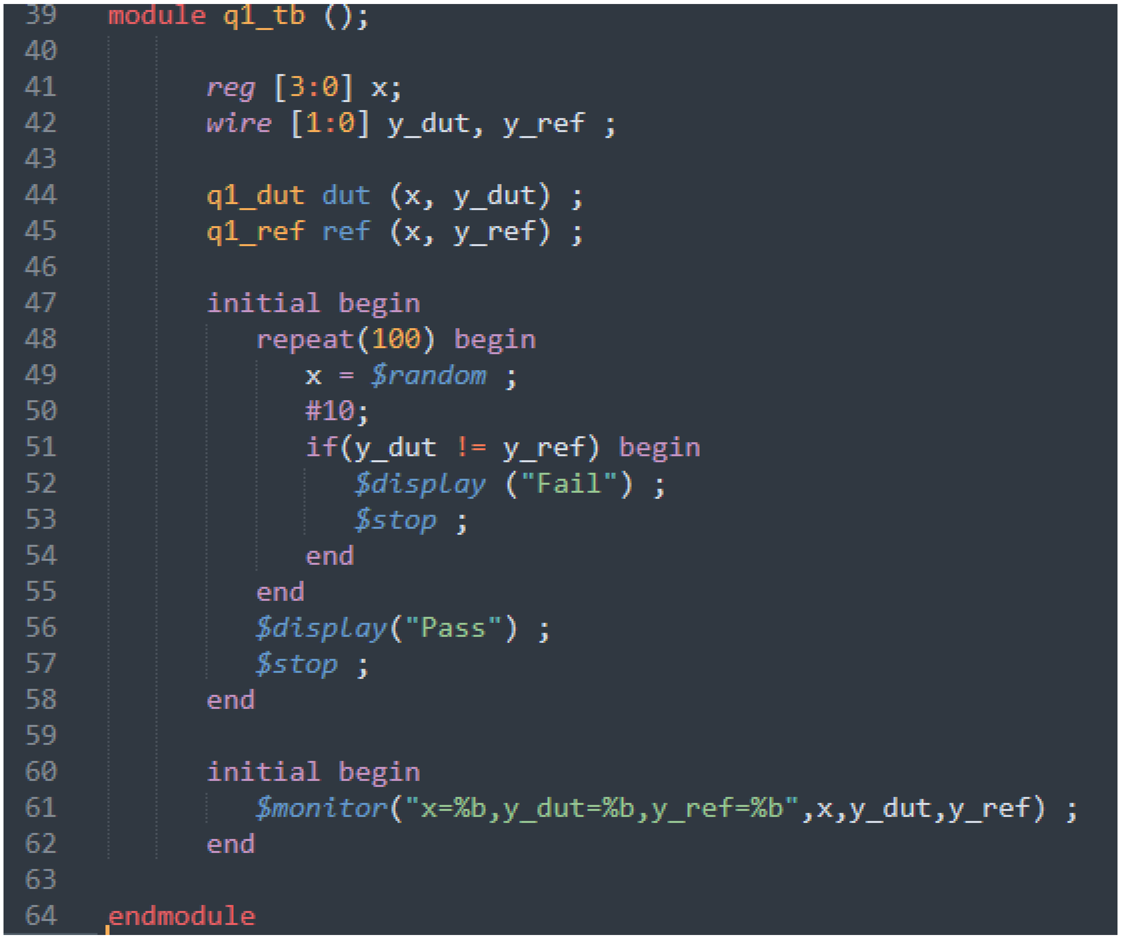
```
`timescale 1ns/1ps
module q1_dut
         (
            input [3:0] x,
            output reg [1:0] y
            );

         always @(*)
            casex(x)
              4'b1xxx: y=2'b11 ;
              4'b01xx: y=2'b10 ;
              4'b001x: y=2'b01 ;
              default: y=2'b00 ;
            endcase

endmodule
```

REF Code



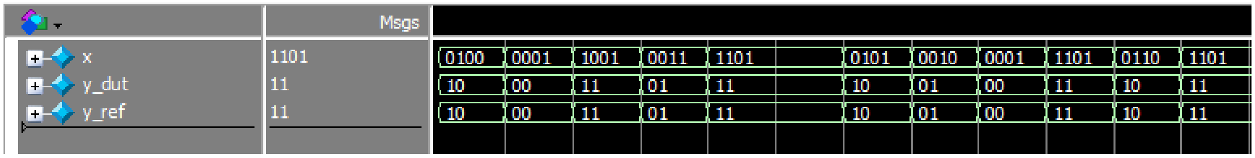
```
module q1_ref
         (
            input [3:0] x,
            output reg [1:0] y
            );


         always @(*)
           if(x[3])
              y=2'b11 ;
           else if (x[2])
              y=2'b10 ;
           else if (x[1])
              y=2'b01 ;
           else
              y=2'b00 ;

endmodule
```

Test bench Code



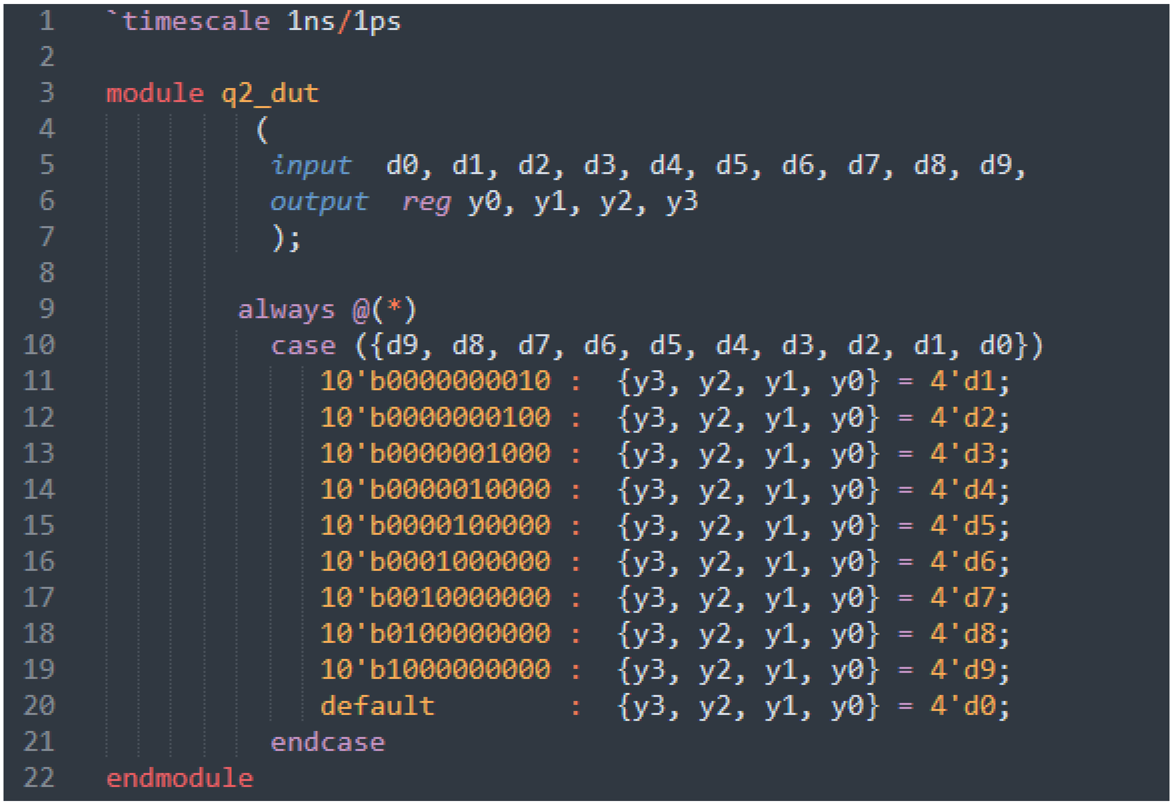
```
module q1_tb ();

        reg [3:0] x;
        wire [1:0] y_dut, y_ref ;

        q1_dut dut (x, y_dut) ;
        q1_ref ref (x, y_ref) ;

        initial begin
            repeat(100) begin
                x = $random ;
                #10;
                if(y_dut != y_ref) begin
                    $display ("Fail") ;
                    $stop ;
                end
            end
            $display("Pass") ;
            $stop ;
        end

        initial begin
            $monitor("x=%b,y_dut=%b,y_ref=%b",x,y_dut,y_ref) ;
        end

endmodule
```

Test Bench Simulation



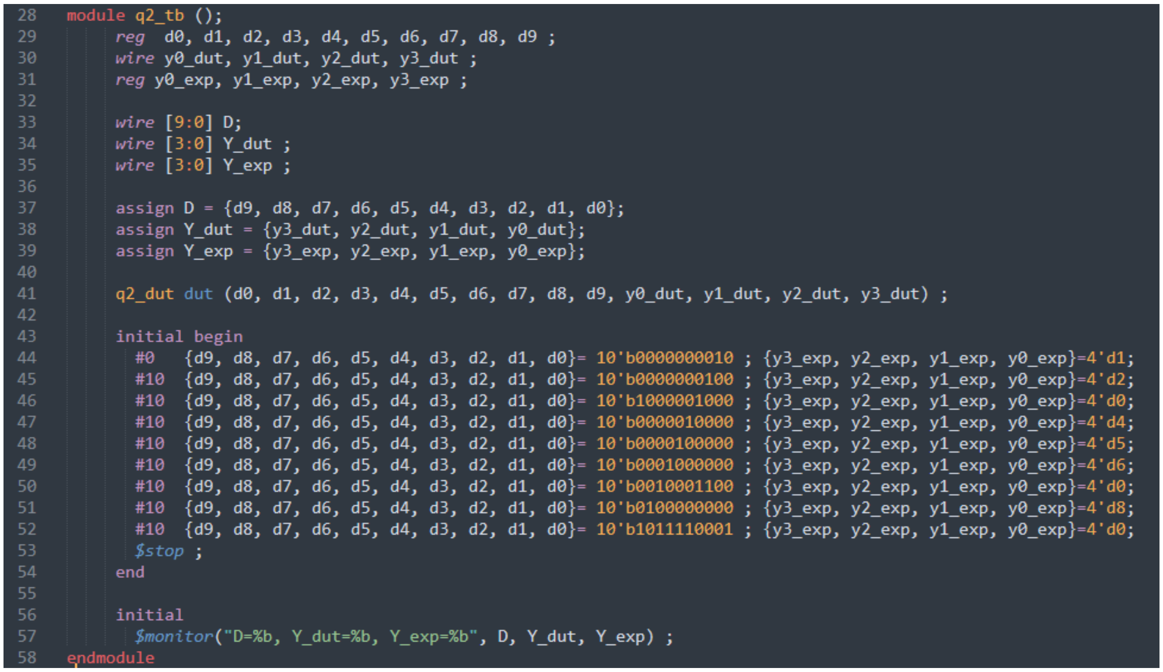
Question 2

DUT Code



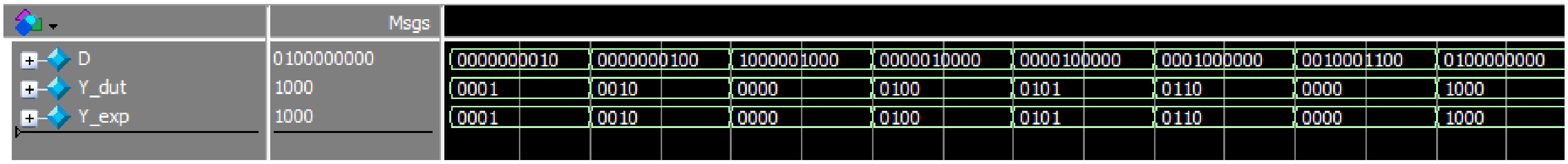
```
`timescale 1ns/1ps

module q2_dut
        (
         input  d0, d1, d2, d3, d4, d5, d6, d7, d8, d9,
         output  reg y0, y1, y2, y3
         );

        always @(*)
         case ({d9, d8, d7, d6, d5, d4, d3, d2, d1, d0})
            10'b0000000010 :  {y3, y2, y1, y0} = 4'd1;
            10'b0000000100 :  {y3, y2, y1, y0} = 4'd2;
            10'b0000001000 :  {y3, y2, y1, y0} = 4'd3;
            10'b0000010000 :  {y3, y2, y1, y0} = 4'd4;
            10'b0000100000 :  {y3, y2, y1, y0} = 4'd5;
            10'b0001000000 :  {y3, y2, y1, y0} = 4'd6;
            10'b0010000000 :  {y3, y2, y1, y0} = 4'd7;
            10'b0100000000 :  {y3, y2, y1, y0} = 4'd8;
            10'b1000000000 :  {y3, y2, y1, y0} = 4'd9;
            default        :  {y3, y2, y1, y0} = 4'd0;
         endcase
endmodule
```

Test Bench Code



```
module q2_tb ();
    reg  d0, d1, d2, d3, d4, d5, d6, d7, d8, d9 ;
    wire y0_dut, y1_dut, y2_dut, y3_dut ;
    reg y0_exp, y1_exp, y2_exp, y3_exp ;

    wire [9:0] D;
    wire [3:0] Y_dut ;
    wire [3:0] Y_exp ;

    assign D = {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0};
    assign Y_dut = {y3_dut, y2_dut, y1_dut, y0_dut};
    assign Y_exp = {y3_exp, y2_exp, y1_exp, y0_exp};

    q2_dut dut (d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, y0_dut, y1_dut, y2_dut, y3_dut) ;

    initial begin
      #0   {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000000010 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd1;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000000100 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd2;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b1000001000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000010000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd4;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000100000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd5;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0001000000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd6;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0010001100 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0100000000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd8;
      #10  {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b1011110001 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
      $stop ;
    end

    initial
      $monitor("D=%b, Y_dut=%b, Y_exp=%b", D, Y_dut, Y_exp) ;
endmodule
```

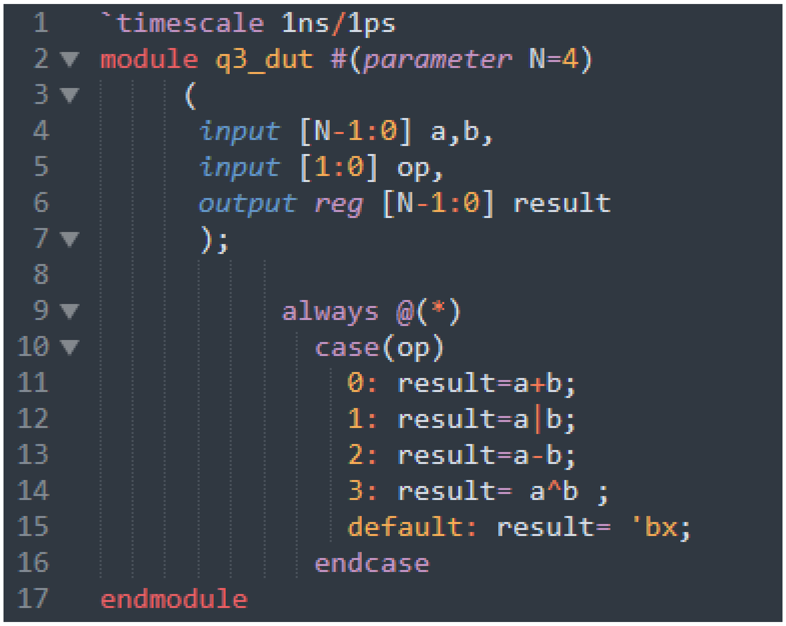
Test bench simulation



```
         Msgs
D        0100000000   0000000010  0000000100  1000001000  0000010000  0000100000  0001000000  0010001100  0100000000
Y_dut    1000         0001        0010        0000        0100        0101        0110        0000        1000
Y_exp    1000         0001        0010        0000        0100        0101        0110        0000        1000
```

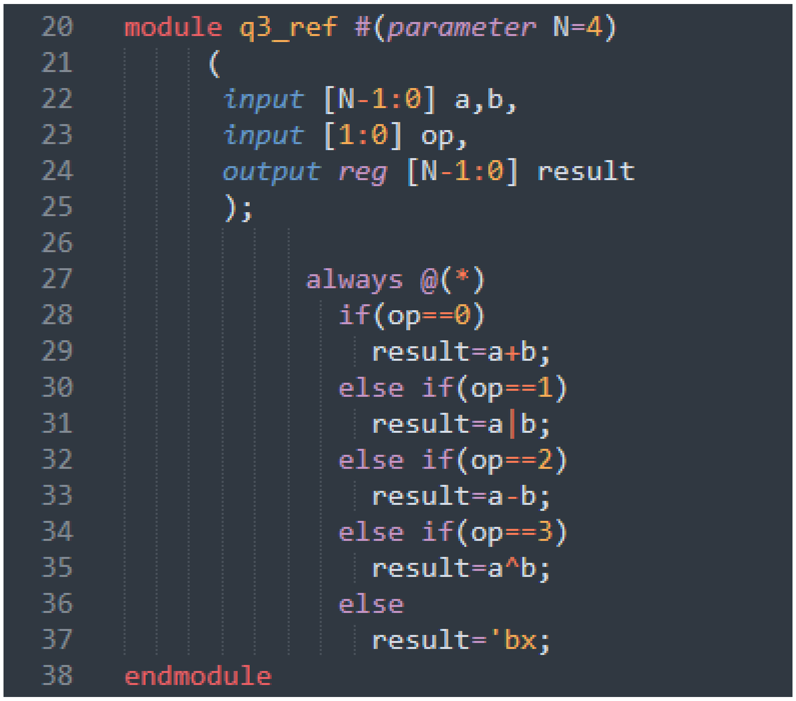
Question 3

DUT Code



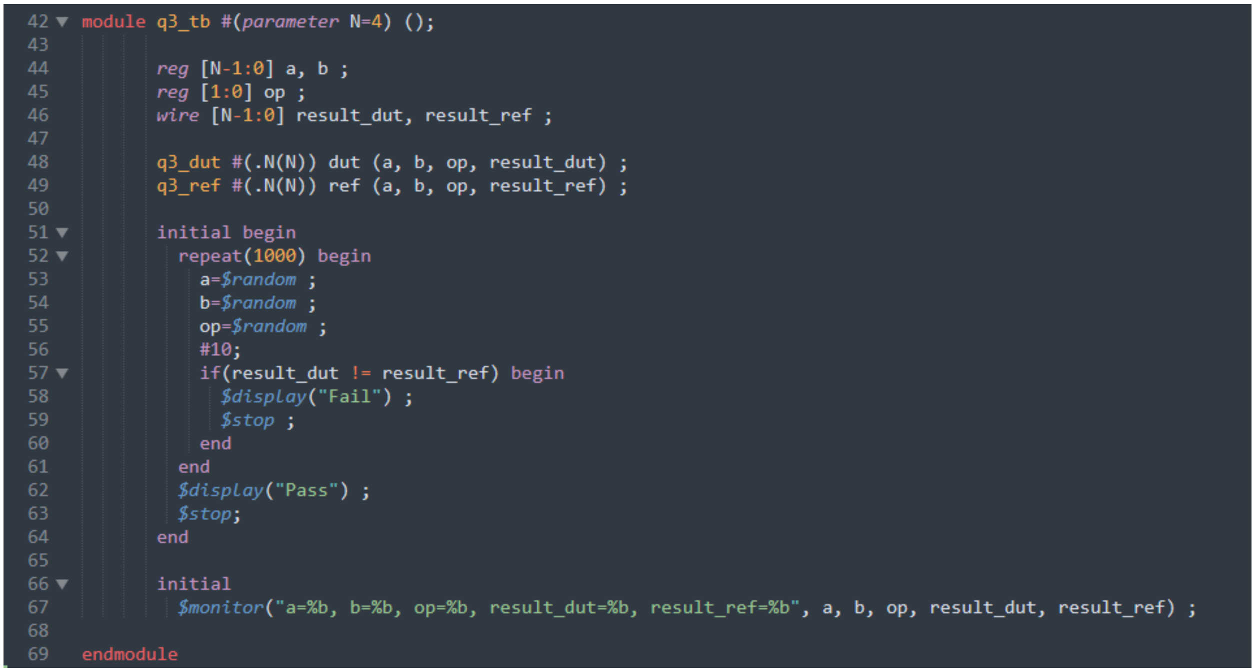
```
`timescale 1ns/1ps
module q3_dut #(parameter N=4)
     (
      input [N-1:0] a,b,
      input [1:0] op,
      output reg [N-1:0] result
      );

           always @(*)
             case(op)
               0: result=a+b;
               1: result=a|b;
               2: result=a-b;
               3: result= a^b ;
               default: result= 'bx;
             endcase
endmodule
```

REF Code



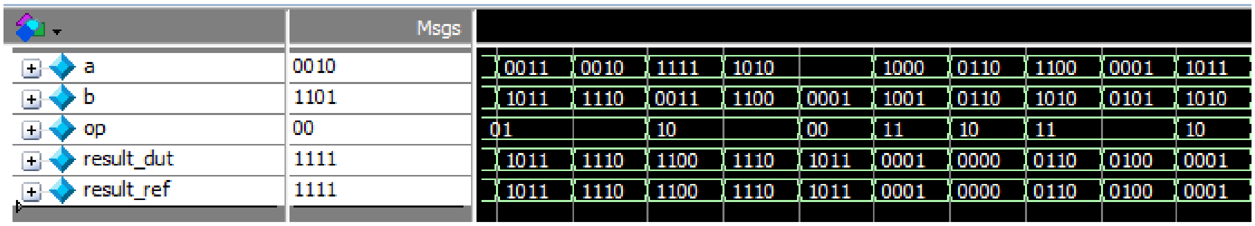
```
module q3_ref #(parameter N=4)
     (
      input [N-1:0] a,b,
      input [1:0] op,
      output reg [N-1:0] result
      );

           always @(*)
             if(op==0)
               result=a+b;
             else if(op==1)
               result=a|b;
             else if(op==2)
               result=a-b;
             else if(op==3)
               result=a^b;
             else
               result='bx;
endmodule
```

Test Bench Code



```
module q3_tb #(parameter N=4) ();

       reg [N-1:0] a, b ;
       reg [1:0] op ;
       wire [N-1:0] result_dut, result_ref ;

       q3_dut #(.N(N)) dut (a, b, op, result_dut) ;
       q3_ref #(.N(N)) ref (a, b, op, result_ref) ;

       initial begin
         repeat(1000) begin
           a=$random ;
           b=$random ;
           op=$random ;
           #10;
           if(result_dut != result_ref) begin
             $display("Fail") ;
             $stop ;
           end
         end
         $display("Pass") ;
         $stop;
       end

       initial
         $monitor("a=%b, b=%b, op=%b, result_dut=%b, result_ref=%b", a, b, op, result_dut, result_ref) ;

endmodule
```

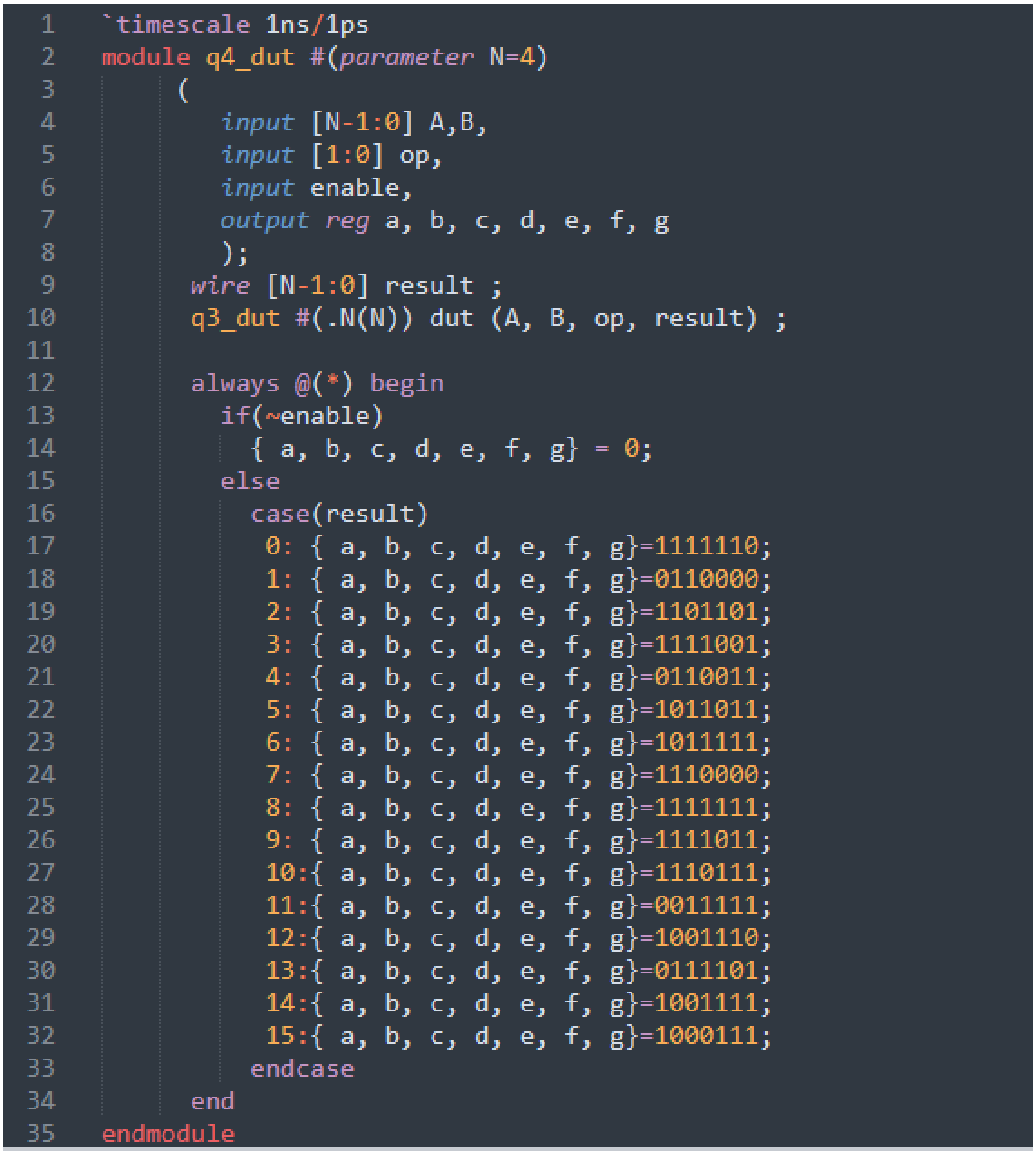
Test Bench Simulation



	Msgs										
a	0010	0011	0010	1111	1010		1000	0110	1100	0001	1011
b	1101	1011	1110	0011	1100	0001	1001	0110	1010	0101	1010
op	00	01		10		00	11	10	11		10
result_dut	1111	1011	1110	1100	1110	1011	0001	0000	0110	0100	0001
result_ref	1111	1011	1110	1100	1110	1011	0001	0000	0110	0100	0001

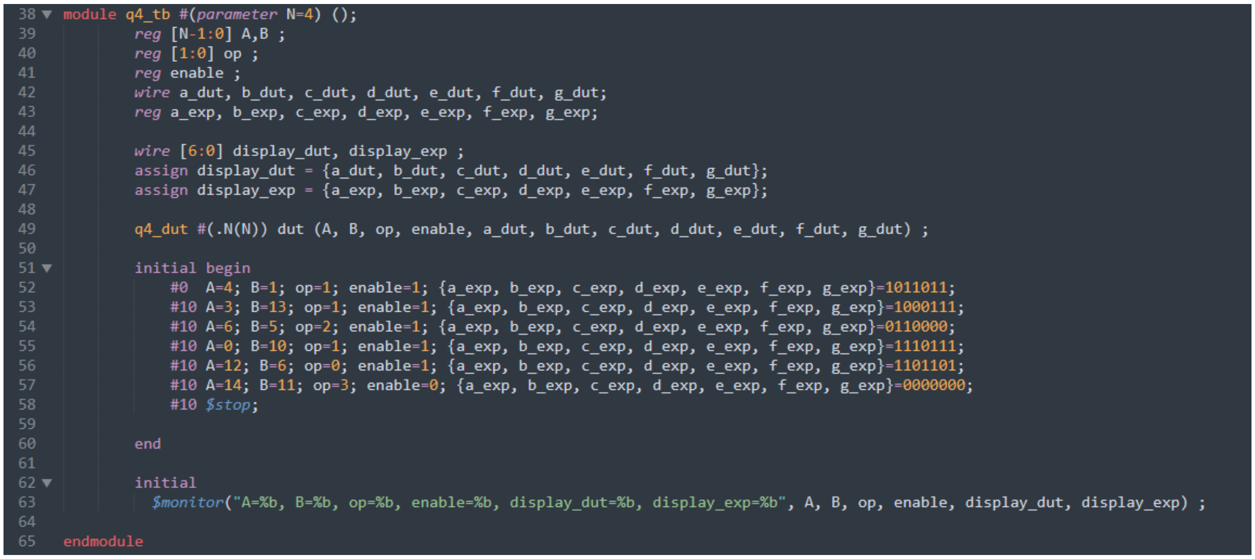
Question 4

DUT Code



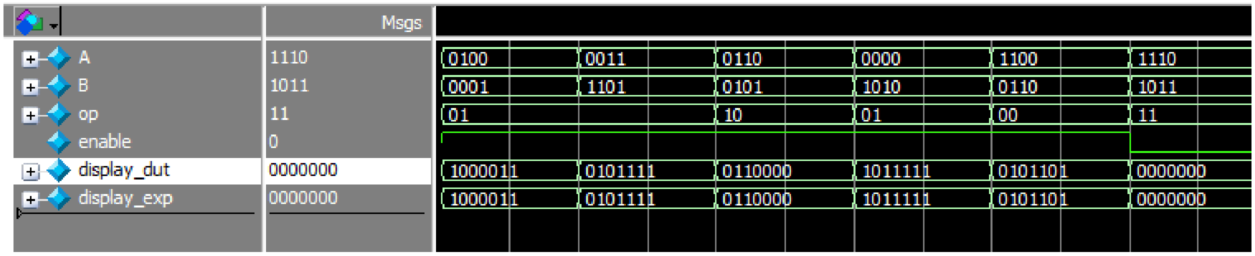
```verilog
`timescale 1ns/1ps
module q4_dut #(parameter N=4)
     (
        input [N-1:0] A,B,
        input [1:0] op,
        input enable,
        output reg a, b, c, d, e, f, g
        );
      wire [N-1:0] result ;
      q3_dut #(.N(N)) dut (A, B, op, result) ;

      always @(*) begin
        if(~enable)
          { a, b, c, d, e, f, g} = 0;
        else
          case(result)
           0: { a, b, c, d, e, f, g}=1111110;
           1: { a, b, c, d, e, f, g}=0110000;
           2: { a, b, c, d, e, f, g}=1101101;
           3: { a, b, c, d, e, f, g}=1111001;
           4: { a, b, c, d, e, f, g}=0110011;
           5: { a, b, c, d, e, f, g}=1011011;
           6: { a, b, c, d, e, f, g}=1011111;
           7: { a, b, c, d, e, f, g}=1110000;
           8: { a, b, c, d, e, f, g}=1111111;
           9: { a, b, c, d, e, f, g}=1111011;
           10:{ a, b, c, d, e, f, g}=1110111;
           11:{ a, b, c, d, e, f, g}=0011111;
           12:{ a, b, c, d, e, f, g}=1001110;
           13:{ a, b, c, d, e, f, g}=0111101;
           14:{ a, b, c, d, e, f, g}=1001111;
           15:{ a, b, c, d, e, f, g}=1000111;
          endcase
      end
endmodule
```

Test Bench Code

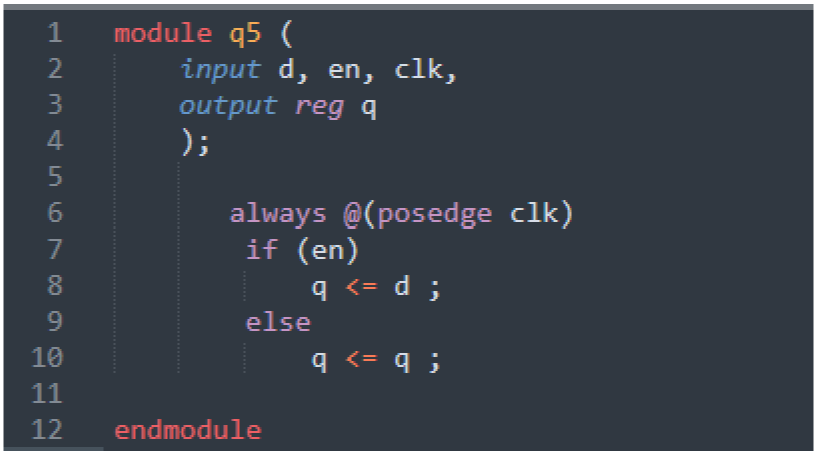


```
module q4_tb #(parameter N=4) ();
        reg [N-1:0] A,B ;
        reg [1:0] op ;
        reg enable ;
        wire a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut;
        reg a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp;

        wire [6:0] display_dut, display_exp ;
        assign display_dut = {a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut};
        assign display_exp = {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp};

        q4_dut #(.N(N)) dut (A, B, op, enable, a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut) ;

        initial begin
            #0  A=4; B=1; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1011011;
            #10 A=3; B=13; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1000111;
            #10 A=6; B=5; op=2; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=0110000;
            #10 A=0; B=10; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1110111;
            #10 A=12; B=6; op=0; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1101101;
            #10 A=14; B=11; op=3; enable=0; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=0000000;
            #10 $stop;

        end

        initial
            $monitor("A=%b, B=%b, op=%b, enable=%b, display_dut=%b, display_exp=%b", A, B, op, enable, display_dut, display_exp) ;

endmodule
```

Test Bench Simulation



Question 5



```
module q5 (
    input d, en, clk,
    output reg q
    );

       always @(posedge clk)
        if (en)
            q <= d ;
        else
            q <= q ;

endmodule
```