

Assignment 2
By Amr Mohamed Ahmed Ebrahim

Question 1

DUT Code

```
1  `timescale 1ns/1ps
2  module q1_dut
3      (
4          input [3:0] x,
5          output reg [1:0] y
6      );
7
8      always @(*)
9          casex(x)
10             4'b1xxx: y=2'b11 ;
11             4'b01xx: y=2'b10 ;
12             4'b001x: y=2'b01 ;
13             default: y=2'b00 ;
14          endcase
15
16  endmodule
```

REF Code

```
19  module q1_ref
20      (
21          input [3:0] x,
22          output reg [1:0] y
23      );
24
25
26      always @(*)
27          if(x[3])
28              y=2'b11 ;
29          else if (x[2])
30              y=2'b10 ;
31          else if (x[1])
32              y=2'b01 ;
33          else
34              y=2'b00 ;
35
36  endmodule
```

Test bench Code

```
39 module q1_tb ();
40
41     reg [3:0] x;
42     wire [1:0] y_dut, y_ref ;
43
44     q1_dut dut (x, y_dut) ;
45     q1_ref ref (x, y_ref) ;
46
47     initial begin
48         repeat(100) begin
49             x = $random ;
50             #10;
51             if(y_dut != y_ref) begin
52                 $display ("Fail") ;
53                 $stop ;
54             end
55         end
56         $display("Pass") ;
57         $stop ;
58     end
59
60     initial begin
61         $monitor("x=%b,y_dut=%b,y_ref=%b",x,y_dut,y_ref) ;
62     end
63
64 endmodule
```

Test Bench Simulation

Msgs													
x	1101	0100	0001	1001	0011	1101	0101	0010	0001	1101	0110	1101	
y_dut	11	10	00	11	01	11	10	01	00	11	10	11	
y_ref	11	10	00	11	01	11	10	01	00	11	10	11	

Question 2

DUT Code

```
1  `timescale 1ns/1ps
2
3  module q2_dut
4  (
5      input  d0, d1, d2, d3, d4, d5, d6, d7, d8, d9,
6      output reg y0, y1, y2, y3
7  );
8
9      always @(*)
10         case ({d9, d8, d7, d6, d5, d4, d3, d2, d1, d0})
11             10'b0000000010 : {y3, y2, y1, y0} = 4'd1;
12             10'b0000000100 : {y3, y2, y1, y0} = 4'd2;
13             10'b0000001000 : {y3, y2, y1, y0} = 4'd3;
14             10'b0000010000 : {y3, y2, y1, y0} = 4'd4;
15             10'b0000100000 : {y3, y2, y1, y0} = 4'd5;
16             10'b0001000000 : {y3, y2, y1, y0} = 4'd6;
17             10'b0010000000 : {y3, y2, y1, y0} = 4'd7;
18             10'b0100000000 : {y3, y2, y1, y0} = 4'd8;
19             10'b1000000000 : {y3, y2, y1, y0} = 4'd9;
20             default       : {y3, y2, y1, y0} = 4'd0;
21         endcase
22     endmodule
```

Test Bench Code

```
28 module q2_tb ();
29     reg d0, d1, d2, d3, d4, d5, d6, d7, d8, d9 ;
30     wire y0_dut, y1_dut, y2_dut, y3_dut ;
31     reg y0_exp, y1_exp, y2_exp, y3_exp ;
32
33     wire [9:0] D;
34     wire [3:0] Y_dut ;
35     wire [3:0] Y_exp ;
36
37     assign D = {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0};
38     assign Y_dut = {y3_dut, y2_dut, y1_dut, y0_dut};
39     assign Y_exp = {y3_exp, y2_exp, y1_exp, y0_exp};
40
41     q2_dut dut (d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, y0_dut, y1_dut, y2_dut, y3_dut) ;
42
43     initial begin
44         #0 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000000010 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd1;
45         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000000100 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd2;
46         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b1000001000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
47         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000010000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd4;
48         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0000100000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd5;
49         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0001000000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd6;
50         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0010001100 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
51         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b0100000000 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd8;
52         #10 {d9, d8, d7, d6, d5, d4, d3, d2, d1, d0}= 10'b1011110001 ; {y3_exp, y2_exp, y1_exp, y0_exp}=4'd0;
53         $stop ;
54     end
55
56     initial
57         $monitor("D=%b, Y_dut=%b, Y_exp=%b", D, Y_dut, Y_exp) ;
58 endmodule
```

Test bench simulation

[illegible]

Question 3

DUT Code

```
1  `timescale 1ns/1ps
2  module q3_dut #(parameter N=4)
3  (
4      input [N-1:0] a,b,
5      input [1:0] op,
6      output reg [N-1:0] result
7  );
8
9      always @(*)
10     case(op)
11         0: result=a+b;
12         1: result=a|b;
13         2: result=a-b;
14         3: result= a^b ;
15         default: result= 'bx;
16     endcase
17 endmodule
```

REF Code

```
20 module q3_ref #(parameter N=4)
21 (
22     input [N-1:0] a,b,
23     input [1:0] op,
24     output reg [N-1:0] result
25 );
26
27     always @(*)
28         if(op==0)
29             result=a+b;
30         else if(op==1)
31             result=a|b;
32         else if(op==2)
33             result=a-b;
34         else if(op==3)
35             result=a^b;
36         else
37             result='bx;
38 endmodule
```

Test Bench Code

```
42 module q3_tb #(parameter N=4) ();
43
44     reg [N-1:0] a, b ;
45     reg [1:0] op ;
46     wire [N-1:0] result_dut, result_ref ;
47
48     q3_dut #(N(N)) dut (a, b, op, result_dut) ;
49     q3_ref #(N(N)) ref (a, b, op, result_ref) ;
50
51     initial begin
52         repeat(1000) begin
53             a=$random ;
54             b=$random ;
55             op=$random ;
56             #10;
57             if(result_dut != result_ref) begin
58                 $display("Fail") ;
59                 $stop ;
60             end
61         end
62         $display("Pass") ;
63         $stop;
64     end
65
66     initial
67         $monitor("a=%b, b=%b, op=%b, result_dut=%b, result_ref=%b", a, b, op, result_dut, result_ref) ;
68
69 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

```
1  `timescale 1ns/1ps
2  module q4_dut #(parameter N=4)
3      (
4          input [N-1:0] A,B,
5          input [1:0] op,
6          input enable,
7          output reg a, b, c, d, e, f, g
8      );
9      wire [N-1:0] result ;
10     q3_dut #(.N(N)) dut (A, B, op, result) ;
11
12     always @(*) begin
13         if(~enable)
14             { a, b, c, d, e, f, g} = 0;
15         else
16             case(result)
17                 0: { a, b, c, d, e, f, g}=1111110;
18                 1: { a, b, c, d, e, f, g}=0110000;
19                 2: { a, b, c, d, e, f, g}=1101101;
20                 3: { a, b, c, d, e, f, g}=1111001;
21                 4: { a, b, c, d, e, f, g}=0110011;
22                 5: { a, b, c, d, e, f, g}=1011011;
23                 6: { a, b, c, d, e, f, g}=1011111;
24                 7: { a, b, c, d, e, f, g}=1110000;
25                 8: { a, b, c, d, e, f, g}=1111111;
26                 9: { a, b, c, d, e, f, g}=1111011;
27                 10:{ a, b, c, d, e, f, g}=1110111;
28                 11:{ a, b, c, d, e, f, g}=0011111;
29                 12:{ a, b, c, d, e, f, g}=1001110;
30                 13:{ a, b, c, d, e, f, g}=0111101;
31                 14:{ a, b, c, d, e, f, g}=1001111;
32                 15:{ a, b, c, d, e, f, g}=1000111;
33             endcase
34         end
35     endmodule
```


Test Bench Code

```

38 ▾ module q4_tb #(parameter N=4) ();
39     reg [N-1:0] A,B ;
40     reg [1:0] op ;
41     reg enable ;
42     wire a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut;
43     reg a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp;
44
45     wire [6:0] display_dut, display_exp ;
46     assign display_dut = {a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut};
47     assign display_exp = {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp};
48
49     q4_dut #(N(N)) dut (A, B, op, enable, a_dut, b_dut, c_dut, d_dut, e_dut, f_dut, g_dut) ;
50
51 ▾ initial begin
52     #0 A=4; B=1; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1011011;
53     #10 A=3; B=13; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1000111;
54     #10 A=6; B=5; op=2; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=0110000;
55     #10 A=0; B=10; op=1; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1110111;
56     #10 A=12; B=6; op=0; enable=1; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=1101101;
57     #10 A=14; B=11; op=3; enable=0; {a_exp, b_exp, c_exp, d_exp, e_exp, f_exp, g_exp}=0000000;
58     #10 $stop;
59
60 end
61
62 ▾ initial
63     $monitor("A=%b, B=%b, op=%b, enable=%b, display_dut=%b, display_exp=%b", A, B, op, enable, display_dut, display_exp) ;
64
65 endmodule

```

Test Bench Simulation

		Msgs												
+ A	1110	0100		0011		0110		0000		1100		1110		
+ B	1011	0001		1101		0101		1010		0110		1011		
+ op	11	01				10		01		00		11		
enable	0													
+ display_dut	0000000	1000011		0101111		0110000		1011111		0101101		0000000		
+ display_exp	0000000	1000011		0101111		0110000		1011111		0101101		0000000		

Question 5

```
1  module q5 (  
2      input d, en, clk,  
3      output reg q  
4  );  
5  
6      always @(posedge clk)  
7          if (en)  
8              q <= d ;  
9          else  
10             q <= q ;  
11  
12  endmodule
```