

**Name:** - Jayraj Patoriya

**ID Number:** - 21EL065

**Branch:** - Electronics

**Division:** - 04

**Subject:** - Digital System Design (3EL42)

# ASSIGNMENT – 1

Q-1: Write a Verilog code for 2X4 decoder.

CODE:

```
1 // Code your design here SV^
2 module decoder2_4(input en,a,b,output reg [3:0]y);
3
4     always @(*)
5
6         begin
7
8             if(en==0)
9                 begin
10
11                     if(a==1'b0 & b==1'b0) y=4'b0000;
12                     else if(a==1'b0 & b==1'b1) y=4'b0001;
13                     else if(a==1'b1 & b==1'b0) y=4'b0101;
14                     else if(a==1 & b==1) y=4'b1000;
15                     else y=4'bxxxx;
16
17                 end
18
19             else
20                 y=4'b1111;
21         end
22     end
23 endmodule
```

TEST BENCH:

```
1 // Code your testbench here
2 // or browse Examples
3
4 module testbench;
5
6     reg a,b,en; wire [3:0]y;
7     decoder2_4 DUT(en,a,b,y);
8
9     initial
10
11         begin
12
13             $monitor("en=%b a=%b b=%b y=%b",en,a,b,y);
14             en=1;a=1'bx;b=1'bx;#5
15             en=0;a=0;b=0;#5
16             en=0;a=0;b=1;#5
17             en=0;a=1;b=0;#5
18             en=0;a=1;b=1;#5
19             $finish;
20
21         end
22     end
23 endmodule
```

## OUTPUT:

Log

Share

```
[2022-08-07 05:34:07 EDT] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out  
en=1 a=x b=x y=1111  
en=0 a=0 b=0 y=0000  
en=0 a=0 b=1 y=0001  
en=0 a=1 b=0 y=0101  
en=0 a=1 b=1 y=1000  
Done
```

## Q-2: Write a Verilog code for Full subtractor.

### CODE:

```
1 // Code your design here
2
3 module full_subtractor(input a,b,cin, output d,bout);
4
5     assign d = a^b^cin ;
6     assign bout = ~a&b | ~(a^b)&cin;
7
8 endmodule
9
```

### TEST BENCH:

```
1 module testbench;
2
3     reg a,b,cin; wire d,bout;
4
5     full_subtractor DUT(a,b,cin,d,bout);
6
7     initial
8     |
9     begin
10
11         $monitor("a=%b b=%b cin=%b d=%b bout=%b",a,b,cin,d,bout);
12         cin=0;a=0;b=0;#5
13         cin=0;a=0;b=1;#5
14         cin=0;a=1;b=0;#5
15         cin=0;a=1;b=1;#5
16         cin=1;a=1;b=1;#5
17         $finish;
18
19     end
20
21 endmodule
22
```

### OUTPUT:

Log

Share

```
[2022-08-07 05:47:04 EDT] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out
a=0 b=0 cin=0 d=0 bout=0
a=0 b=1 cin=0 d=1 bout=1
a=1 b=0 cin=0 d=1 bout=0
a=1 b=1 cin=0 d=0 bout=0
a=1 b=1 cin=1 d=1 bout=1
Done
```

### Q-3: Write a Verilog code for 2-bit comparator.

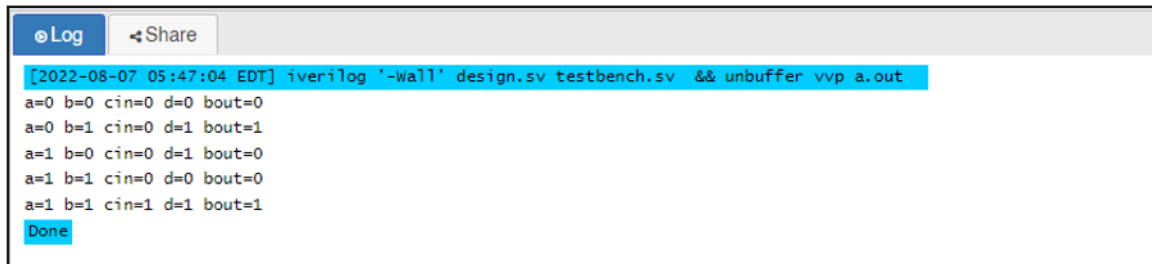
#### CODE:

```
1 module comparator_2bit(a,b,equal,greater,lower);
2
3 output equal ;
4
5 output greater ;
6
7 output lower ;
8
9 input [1:0] a; input [1:0] b;
10
11 assign equal = (a==b) ? 1 : 0;
12
13 assign greater = (a>b) ? 1 : 0;
14
15 assign lower = (a<b) ? 1 : 0;
16
17 endmodule
18
```

#### TEST BENCH:

```
1 module testbench;
2
3 reg [1:0]a,b;
4 wire equal,greater,lower;
5
6 comparator_2bit DUT(a,b,equal,greater,lower);
7
8 initial
9
10 begin
11
12 $monitor("a[0]=%b a[1]=%b b[0]=%b b[1]=%b greater =%b
lower =%b equal=%b",a[0],a[1],b[0],b[1],greater,lower,equal);
13 a=2'b00;b=2'b01;#5
14 a=2'b11;b=2'b10;#5
15 a=2'b00;b=2'b00;#5
16 $finish;
17
18 end
19
20 endmodule
21
```

## OUTPUT:



```
[2022-08-07 05:47:04 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
a=0 b=0 cin=0 d=0 bout=0
a=0 b=1 cin=0 d=1 bout=1
a=1 b=0 cin=0 d=1 bout=0
a=1 b=1 cin=0 d=0 bout=0
a=1 b=1 cin=1 d=1 bout=1
Done
```

**Q-4: Write a Verilog code for 3 bit binary to gray convertor.**

**CODE:**

```
1 module bin_to_gray
2
3 (input [2:0]cin,output [2:0]d);
4
5 assign d[0] = cin[1] ^ cin[0];
6 assign d[1] = cin[2] ^ cin[1];
7 assign d[2] = cin[2];
8
9 endmodule
10
```

**TEST BENCH:**

```
1 module testbench();
2
3 |reg [2:0] cin;
4 wire [2:0] d;
5
6 bin_to_gray DUT(cin,d);
7
8 initial
9
10 begin
11
12 $monitor("cin = %b d = %b",cin,d);
13 cin = 3'b001; #10
14 cin = 3'b101; #10
15 cin = 3'b111; #10
16 cin = 3'b110; #10
17 $finish;
18
19 end
20
21 endmodule
22
```

**OUTPUT:**

LogShare

[2022-08-07 16:23:16 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out

cin = 001 d = 001  
cin = 101 d = 111  
cin = 111 d = 100  
cin = 110 d = 101

Done

**Q-5: Write a Verilog code for BCD to excess 3 convertor.**

**CODE:**

```
1 module bcd_to_excess3(input a,b,c,d,output x,y,z,w)SV/Verilog Design
2
3 assign x = a | (b & c) | b & d;
4 assign y = (~b & c) | (~b & d) | (b & ~c & ~d);
5 assign z = (c & d) | (~c & ~d);
6 assign w = ~d;
7
8 endmodule
9
10
```

**TEST BENCH:**

```
1 module testbench();
2
3     reg a,b,c,d;
4     wire x,y,z,w;
5
6     bcd_to_excess3 DUT(a,b,c,d,x,y,z,w);
7
8     initial
9
10        begin
11
12            $monitor("a = %b b = %b c = %b d = %b, x = %b y = %b z = %b w = %b",a,b,c,d,x,y,z,w);
13
14            a = 1'b0; b = 1'b1; c = 1'b0; d = 1'b0;#50
15            a = 1'b1; b = 1'b1; c = 1'b0; d = 1'b0;#50
16            a = 1'b0; b = 1'b0; c = 1'b0; d = 1'b1;#50
17            a = 1'b1; b = 1'b0; c = 1'b0; d = 1'b1;#50
18
19            $finish;
20
21        end
22    endmodule
23
```

**OUTPUT:**

```
Log Share
[2022-08-07 16:39:34 EDT] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.out
a = 0 b = 1 c = 0 d = 0, x = 0 y = 1 z = 1 w = 1
a = 1 b = 1 c = 0 d = 0, x = 1 y = 1 z = 1 w = 1
a = 0 b = 0 c = 0 d = 1, x = 0 y = 1 z = 0 w = 0
a = 1 b = 0 c = 0 d = 1, x = 1 y = 1 z = 0 w = 0
Done
```



