DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ST JOSEPH ENGINEERING COLLEGE

MANGALURU-575028

Assignment-1

Simulation Assignment

Course Title	**:**	**Digital System Design using Verilog**
Laboratory Code	**:**	**18EC644**
Semester	**:**	**VI**
Student Name	**:**	**Aaron Dsouza**
USN	**:**	**4SO20EC003**
Section	**:**	**A**
Submitted to	**:**	**Ms. Aswathi T**
Submitted on	**:**	**3-06-2023**

AARON DSOUZA
4SO20EC003

1) Develop a Verilog model for a circuit that implements the following Boolean expressions:

a) A+BC+$\bar{B}$D

b) $\bar{B}$C +B$\bar{C}$ + $\bar{D}$

a) A+BC+$\bar{B}$D

Solution:

- **Design code:**

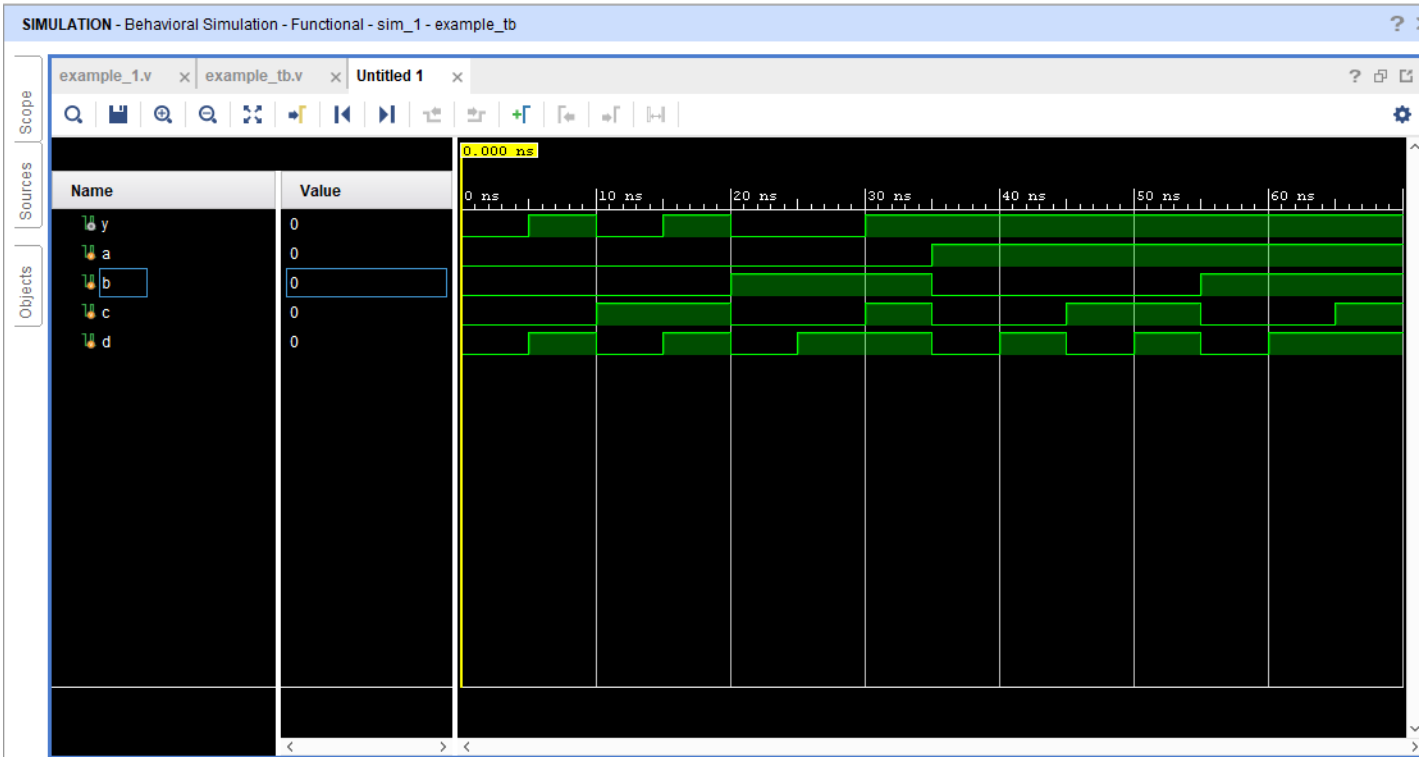
```
module example_1 (y,a,b,c,d);
output y;
input a,b,c,d;
assign y=a|(b&c)|(~b&d);
endmodule
```

- **Test-bench code:**

```
module example_1_tb ();
reg a,b,c,d;
wire y;
example_1 u0 (y,a,b,c,d);
initial
begin
$monitor("a = %b, b = %b, c = %b, d = %b, y = %b", a, b, c, d, y);
    a=0; b=0;c=0;d=0;
#5  a=0;b=0;c=0;d=1;
#5  a=0;b=0;c=1;d=0;
#5  a=0;b=0;c=1;d=1;
#5  a=0;b=1;c=0;d=0;
#5  a=0;b=1;c=0;d=1;
#5  a=0;b=1;c=1;d=0;
#5  a=0;b=1;c=1;d=1;
#5  a=1;b=0;c=0;d=0;
#5  a=1;b=0;c=0;d=1;
#5  a=1;b=0;c=1;d=0;
#5  a=1;b=0;c=1;d=1;
#5  a=1;b=1;c=0;d=0;
#5  a=1;b=1;c=0;d=1;
#5  a=1;b=1;c=1;d=0;
#5  a=1;b=1;c=1;d=1;
```

```
#5 $finish;
end
endmodule
```

Output:



Verification:

```
[2023-05-27 00:30:41 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
a = 0, b = 0, c = 0, d = 0, y = 0
a = 0, b = 0, c = 0, d = 1, y = 1
a = 0, b = 0, c = 1, d = 0, y = 0
a = 0, b = 0, c = 1, d = 1, y = 1
a = 0, b = 1, c = 0, d = 0, y = 0
a = 0, b = 1, c = 0, d = 1, y = 0
a = 0, b = 1, c = 1, d = 0, y = 1
a = 0, b = 1, c = 1, d = 1, y = 1
a = 1, b = 0, c = 0, d = 0, y = 1
a = 1, b = 0, c = 0, d = 1, y = 1
a = 1, b = 0, c = 1, d = 0, y = 1
a = 1, b = 0, c = 1, d = 1, y = 1
a = 1, b = 1, c = 0, d = 0, y = 1
a = 1, b = 1, c = 0, d = 1, y = 1
a = 1, b = 1, c = 1, d = 0, y = 1
a = 1, b = 1, c = 1, d = 1, y = 1
Done
```

b) $\bar{B}C + B\bar{C} + \bar{D}$

Solution:

- **Design code:**

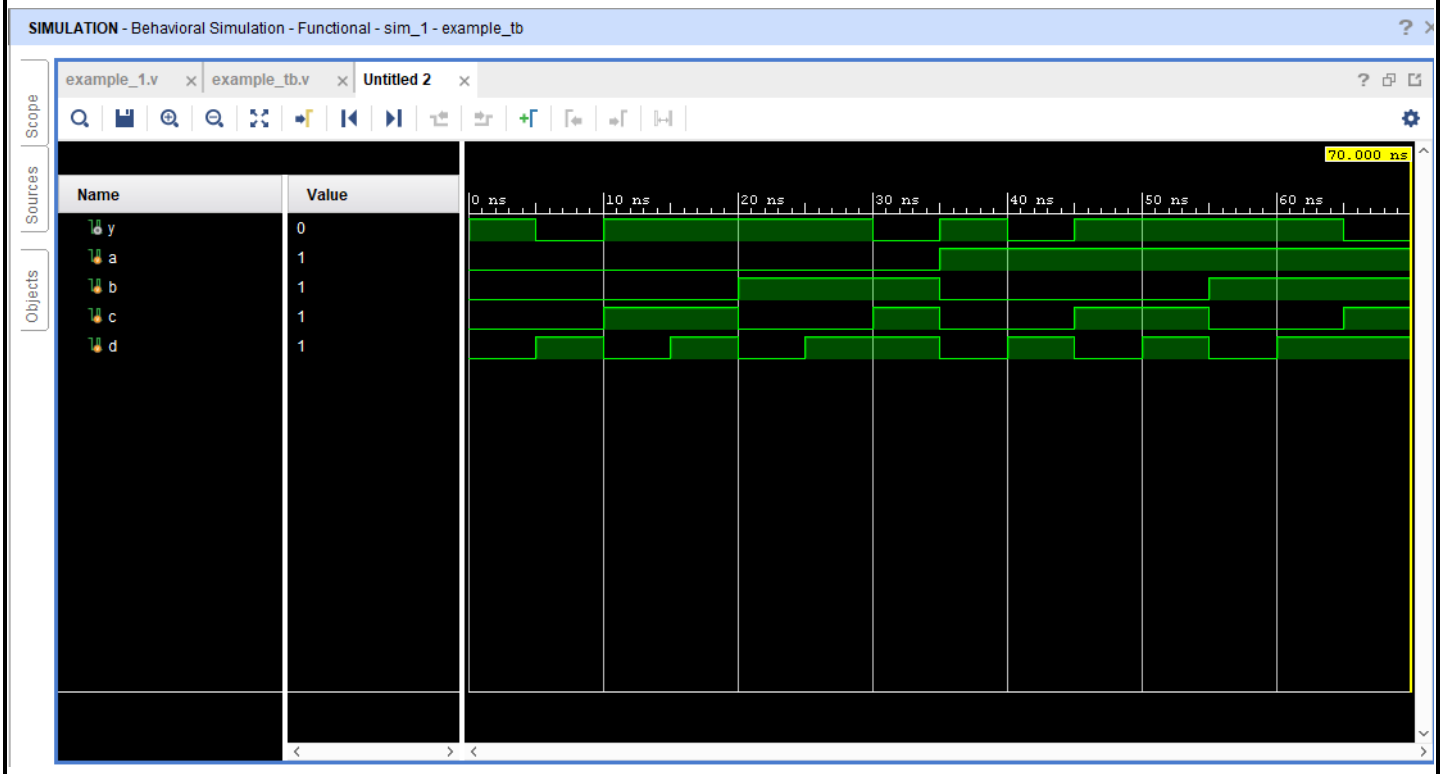
```
module example_1 (y,a,b,c,d);
output y;
input a,b,c,d;
assign y=(~b&c)|(b&~c)|(~d);
endmodule
```

- **Test-bench code:**

```
module example_1_tb ();
reg a,b,c,d;
wire y;
example_1 u0 (y,a,b,c,d);
initial
begin
$monitor ("a = %b, b = %b, c = %b, d = %b, y = %b", a, b, c, d, y);
     a=0; b=0; c=0;d=0;
#5  a=0;b=0;c=0;d=1;
#5  a=0;b=0;c=1;d=0;
#5  a=0;b=0;c=1;d=1;
#5  a=0;b=1;c=0;d=0;
#5  a=0;b=1;c=0;d=1;
#5  a=0;b=1;c=1;d=0;
#5  a=0;b=1;c=1;d=1;
#5  a=1;b=0;c=0;d=0;
#5  a=1;b=0;c=0;d=1;
#5  a=1;b=0;c=1;d=0;
#5  a=1;b=0;c=1;d=1;
#5  a=1;b=1;c=0;d=0;
#5  a=1;b=1;c=0;d=1;
#5  a=1;b=1;c=1;d=0;
#5  a=1;b=1;c=1;d=1;
#5  $finish;
end
endmodule
```

AARON DSOUZA
4SO20EC003

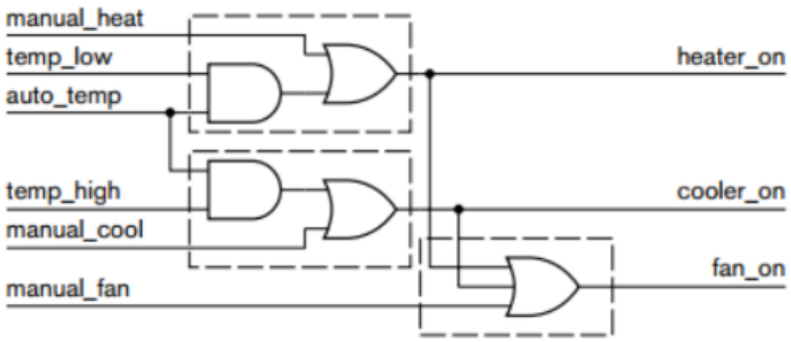
Output:



Verification:

```
[2023-05-27 00:37:21 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
a = 0, b = 0, c = 0, d = 0, y = 1
a = 0, b = 0, c = 0, d = 1, y = 0
a = 0, b = 0, c = 1, d = 0, y = 1
a = 0, b = 0, c = 1, d = 1, y = 1
a = 0, b = 1, c = 0, d = 0, y = 1
a = 0, b = 1, c = 0, d = 1, y = 1
a = 0, b = 1, c = 1, d = 0, y = 1
a = 0, b = 1, c = 1, d = 1, y = 0
a = 1, b = 0, c = 0, d = 0, y = 1
a = 1, b = 0, c = 0, d = 1, y = 0
a = 1, b = 0, c = 1, d = 0, y = 1
a = 1, b = 0, c = 1, d = 1, y = 1
a = 1, b = 1, c = 0, d = 0, y = 1
a = 1, b = 1, c = 0, d = 1, y = 1
a = 1, b = 1, c = 1, d = 0, y = 1
a = 1, b = 1, c = 1, d = 1, y = 0
Done
```

2) Develop a Verilog model for the given combinational circuit that implements the three Boolean equations, representing part of the



Solution:

➢ Design code:

```
module indicator
(heater_on,cooler_on,fan_on,manual_heat,temp_low,auto_temp,temp_high,manual_co
ol,manual_fan);
 output heater_on,cooler_on,fan_on;
 input manual_heat,temp_low,auto_temp,temp_high,manual_cool,manual_fan;
 wire w1,w2;
 and a1(w1,temp_low,auto_temp);
 and a2(w2,auto_temp,temp_high);
 or o1(heater_on,manual_heat,w1);
 or o2(cooler_on,manual_cool,w2);
 or o3(fan_on,cooler_on,heater_on);
 endmodule
```

➢ Test-bench code:

```
module indicator_tb;
 reg manual_heat, temp_low, auto_temp, temp_high, manual_cool, manual_fan;
 wire heater_on, cooler_on, fan_on;
 indicator u0 (.heater_on(heater_on), .cooler_on(cooler_on), .fan_on(fan_on),
              .manual_heat(manual_heat), .temp_low(temp_low), .auto_temp(auto_temp),
              .temp_high(temp_high), .manual_cool(manual_cool),
.manual_fan(manual_fan));
 Initial
 begin
    #10; $display("Input Values: manual_heat=%b, temp_low=%b, auto_temp=%b,
temp_high=%b, manual_cool=%b, manual_fan=%b", manual_heat, temp_low,
auto_temp, temp_high, manual_cool, manual_fan);
```

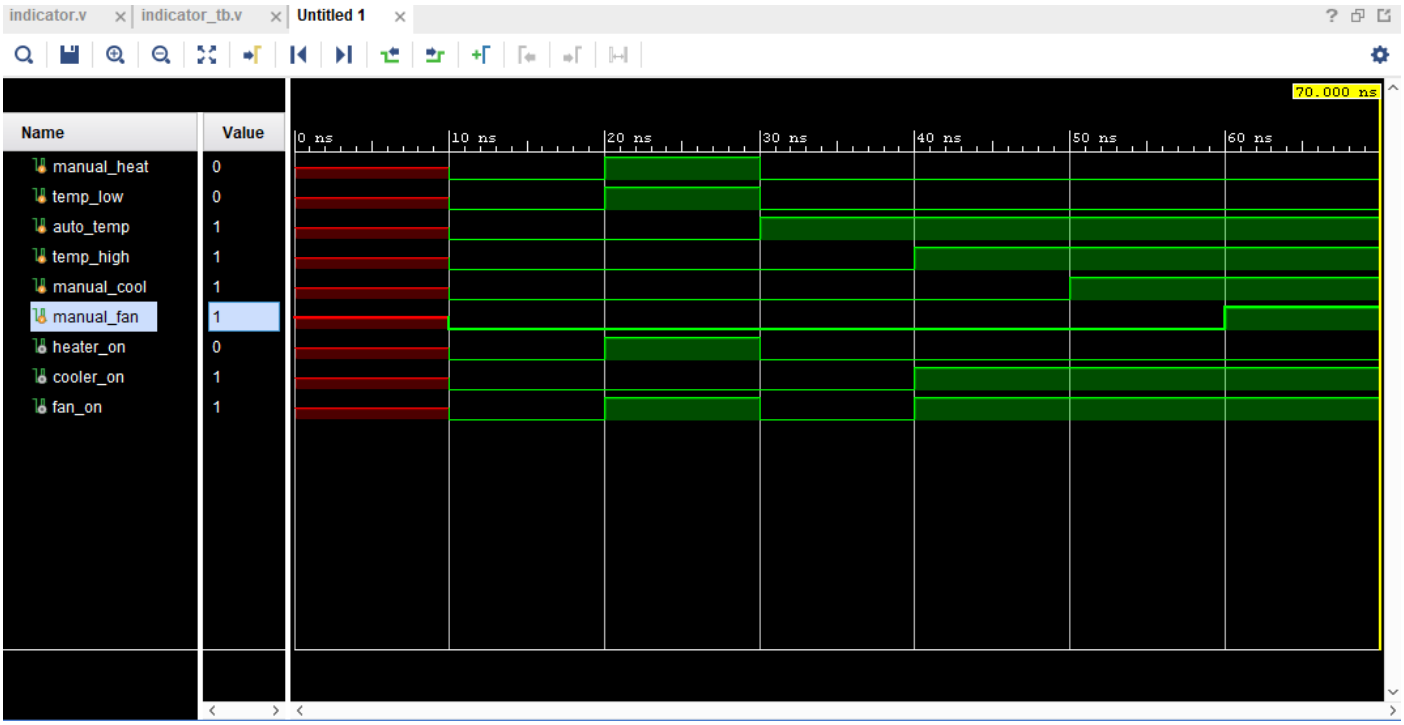
```
    $monitor ("Output Signals: heater_on=%b, cooler_on=%b, fan_on=%b",heater_on,
cooler_on, fan_on);

 manual_heat = 0; temp_low = 0; auto_temp = 0; temp_high = 0; manual_cool = 0;
manual_fan = 0;
    #10;
manual_heat = 1; temp_low = 1; auto_temp = 0; temp_high = 0; manual_cool = 0;
manual_fan = 0;
    #10;
manual_heat = 0; temp_low = 0; auto_temp = 1; temp_high = 0; manual_cool = 0;
manual_fan = 0;
    #10;
manual_heat = 0; temp_low = 0; auto_temp = 1; temp_high = 1; manual_cool = 0;
manual_fan = 0;
    #10;
manual_heat = 0; temp_low = 0; auto_temp = 1; temp_high = 1; manual_cool = 1;
manual_fan = 0;
    #10;
manual_heat = 0; temp_low = 0; auto_temp = 1; temp_high = 1; manual_cool = 1;
manual_fan = 1;
    #10;
 $finish;
 end
endmodule
```

Output:

```
Input Values: manual_heat=x, temp_low=x, auto_temp=x, temp_high=x, manual
_cool=x, manual_fan=x
Output Signals: heater_on=0, cooler_on=0, fan_on=0
Output Signals: heater_on=1, cooler_on=0, fan_on=1
Output Signals: heater_on=0, cooler_on=0, fan_on=0
Output Signals: heater_on=0, cooler_on=1, fan_on=1
```

AARON DSOUZA
4SO20EC003

Output Waveform:



3) Develop a Verilog model for the following:

a) 3 to 8-line Decoder

b) 8 to 3-line Priority Encoder

a.)3 to 8-line Decoder:

Solution:

Design Code:

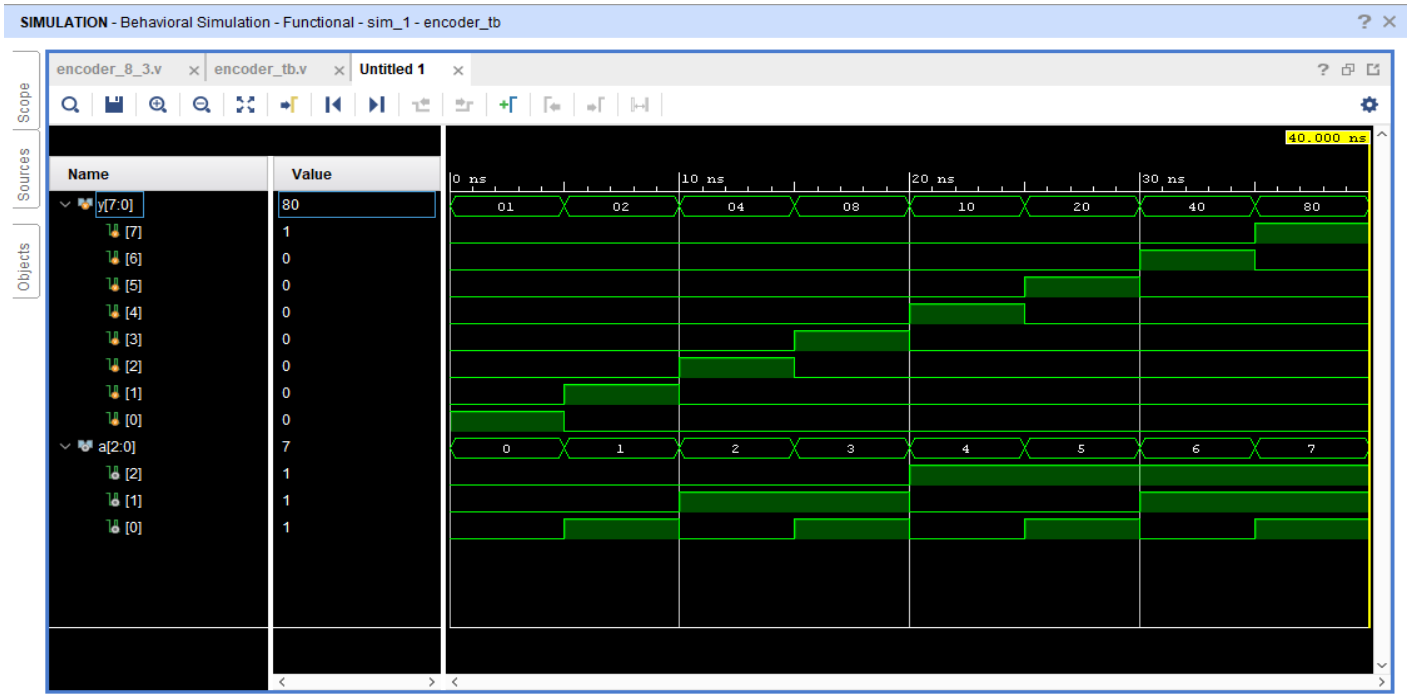
```
module encoder_8_3(a,y);
output reg [2:0]a;
input [7:0]y;
always@(y)
begin
case(y)
8'b00000001:a=3'b000;
8'b00000010:a=3'b001;
8'b00000100:a=3'b010;
8'b00001000:a=3'b011;
8'b00010000:a=3'b100;
8'b00100000:a=3'b101;
8'b01000000:a=3'b110;
8'b10000000:a=3'b111;
default:$display ("ERROR");
endcase
end
endmodule
```

Test-bench Code:

```
module encoder_tb();
reg[7:0]y;
wire[2:0]a;
encoder_8_3 u0(a,y);
initial
begin
y=8'b00000001;
#5 y=8'b00000010;
#5 y=8'b00000100;
#5 y=8'b00001000;
#5 y=8'b00010000;
#5 y=8'b00100000;
#5 y=8'b01000000;
```

```
#5 y=8'b10000000;
#5 $finish;
end
endmodule
```

Output:



b) 8 to 3-line Priority Encoder:

Solution:

Design Code:

```
module encoder_8_3(a,y);
output reg [2:0]a;
input [7:0]y;
always@(y)
begin
casex(y)
8'b00000001:a=3'b000;
8'b0000001x:a=3'b001;
8'b000001xx:a=3'b010;
8'b00001xxx:a=3'b011;
8'b0001xxxx:a=3'b100;
8'b001xxxxx:a=3'b101;
8'b01xxxxxx:a=3'b110;
8'b1xxxxxxx:a=3'b111;
```

```
default:$display ("ERROR");
endcase
end
endmodule
```

Test-bench Code:

```
module encoder_tb();
reg[7:0]y;
wire[2:0]a;
encoder_8_3 u0(a,y);
initial begin
y=8'b00000001;
#5 y=8'b0000001x;
#5 y=8'b000001xx;
#5 y=8'b00001xxx;
#5 y=8'b0001xxxx;
#5 y=8'b001xxxxx;
#5 y=8'b01xxxxxx;
#5 y=8'b1xxxxxxx;
#5 $finish;
end
endmodule
```

Output:

4) Implement the given Boolean expression using 4:1 MUX and develop a Verilog model the designed MUX.

$Y=\sum m\ (0,2,6,9,11,13)$

Solution:

Binary Numbers:	**Input A**	**Input B**	**Input C**	**Input D**	**Output Y**
0	0	0	0	0	**1**
1	0	0	0	1	0
2	0	0	1	0	**1**
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	**1**
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	**1**
10	1	0	1	0	0
11	1	0	1	1	**1**
12	1	1	0	0	0
13	1	1	0	1	**1**
14	1	1	1	0	0
15	1	1	1	1	0

AARON DSOUZA
4SO20EC003

Simplification:

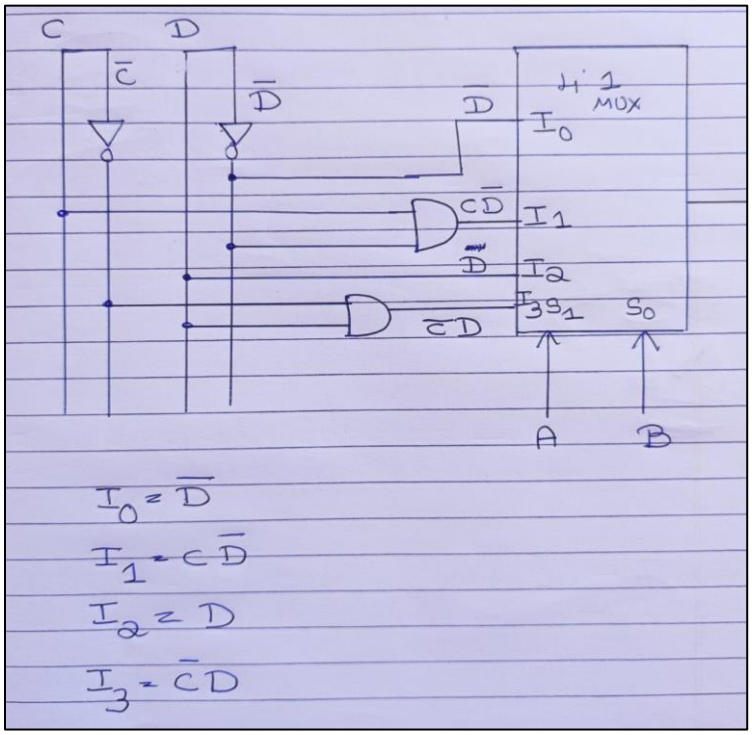
	I0	I1	I2	I3
00	0	4	8	12
01	1	5	9	13
10	2	6	10	14
11	3	7	11	15

- **I0 = C'D' +CD'**
  = D'(C'+C)
  I0 = D'

- **I1 = CD'**

- **I2 = C'D +CD**
  = D (C'+C)
  I2 = D

- **I3 = C'D**



Design Code:

```
module mux4_1(y, sel,d,c);
input d, c;
input [1:0] sel;
output y;
reg y;
wire in_0, in_1, in_2, in_3;
  assign in_0 = ~d;
  assign in_1 = c & ~d;
  assign in_2 = d;
  assign in_3 = ~c & d;
  always @*
```

```
begin
case (sel)
      2'b00: y = in_0;
      2'b01: y = in_1;
      2'b10: y = in_2;
      2'b11: y = in_3;
      default: y = 1'bx;
     endcase
  end
endmodule
```

Test-bench Code:

```
module mux4_1_tb;
reg d, c;
reg [1:0] sel;
wire y;
mux4_1 uut (.d(d),.c(c),.sel(sel),.y(y));
initial
begin
  d = 0;
  c = 0;
  sel = 2'b00;
  #10;
$display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 0;
  c = 1;
  sel = 2'b01;
  #10;
$display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 1;
  c = 1;
  sel = 2'b10;
  #10;
$display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 1;
  c = 0;
  sel = 2'b11;
  #10;
$display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
$finish;
  end
endmodule
```

AARON DSOUZA
4SO20EC003

Output:

testbench.sv

```
module mux4_1_tb;
  reg d, c;
  reg [1:0] sel;
  wire y;
  mux4_1 uut (.d(d),.c(c),.sel(sel),.y(y));
  initial begin
  d = 0;
  c = 0;
  sel = 2'b00;
  #10;
    $display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 0;
  c = 1;
  sel = 2'b01;
  #10;
    $display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 1;
  c = 1;
  sel = 2'b10;
  #10;
    $display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  d = 1;
  c = 0;
  sel = 2'b11;
  #10;
    $display("d = %b, c = %b, sel = %b, y = %b", d, c, sel, y);
  $finish;
  end
endmodule
```

design.sv

```
module mux4_1(y, sel,d,c);
  input d, c;
  input [1:0] sel;
  output y;
  reg y;

  wire in_0, in_1, in_2, in_3;
  assign in_0 = ~d;
  assign in_1 = c & ~d;
  assign in_2 = d;
  assign in_3 = ~c & d;

  always @* begin
    case (sel)
      2'b00: y = in_0;
      2'b01: y = in_1;
      2'b10: y = in_2;
      2'b11: y = in_3;
      default: y = 1'bx;
    endcase
  end
endmodule
```

Log | Share

```
[2023-06-03 04:36:05 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
d = 0, c = 0, sel = 00, y = 1
d = 0, c = 1, sel = 01, y = 1
d = 1, c = 1, sel = 10, y = 1
d = 1, c = 0, sel = 11, y = 1
Done
```

Output Waveform: