# NAME: DHVANI PATEL

# ROLL NO : 21BCP116

# Lab 6: Designing of Multiplexer, Demultiplexer, Decoder, Encoder

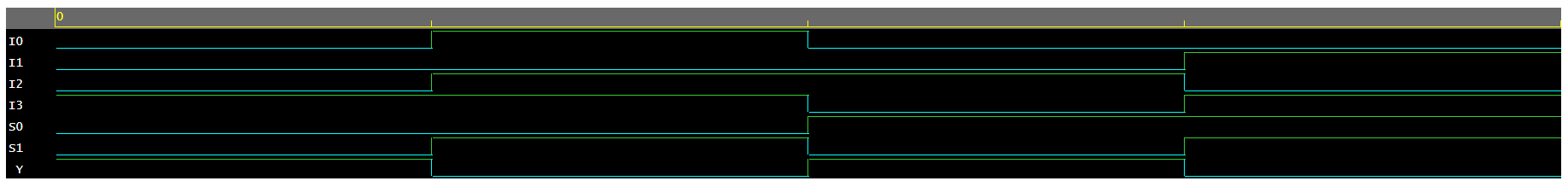
Question 1: Write a Verilog code to design a 4:1 Multiplexer and verify the same.

Code:

|  |  |
| --- | --- |
| // Code your testbench here  // or browse Examples  module tb\_MUX;  reg S0,S1,I0,I1,I2,I3;  wire Y;  MUX M4(S0,S1,I0,I1,I2,I3,Y);  initial  begin  S0=0; S1=0; I0=0 ;I1=0;I2=0;I3=1; #1  $display("S0=%b S1=%b",S0,S1);  $display("Y=%b",Y);  S0=0; S1=1; I0=1 ;I1=0;I2=1;I3=1; #1  $display("S0=%b S1=%b",S0,S1);  $display("Y=%b",Y);  S0=1; S1=0; I0=0 ;I1=0;I2=1;I3=0; #1  $display("S0=%b S1=%b",S0,S1);  $display("Y=%b",Y);  S0=1; S1=1; I0=0 ;I1=1;I2=0;I3=1; #1  $display("S0=%b S1=%b",S0,S1);  $display("Y=%b",Y);  end  initial  begin  $dumpfile("dump.vcd");  $dumpvars(1);  end  endmodule | // Code your design here  module MUX(input s0,s1,I0,I1,I2,I3, output y);  wire s0\_not,s1\_not,a,b,c,d;  not(s0\_not,s0);  not(s1\_not,s1);  and(a,s0,s1,I0);  and(b,s0\_not,s1,I1);  and(c,s0,s1\_not,I2);  and(d,s0\_not,s1\_not,I3);  or(y,a,b,c,d);  endmodule |

Output:

VCD info: dumpfile dump.vcd opened for output.  
S0=0 S1=0  
Y=1  
S0=0 S1=1  
Y=0  
S0=1 S1=0  
Y=1  
S0=1 S1=1  
Y=0



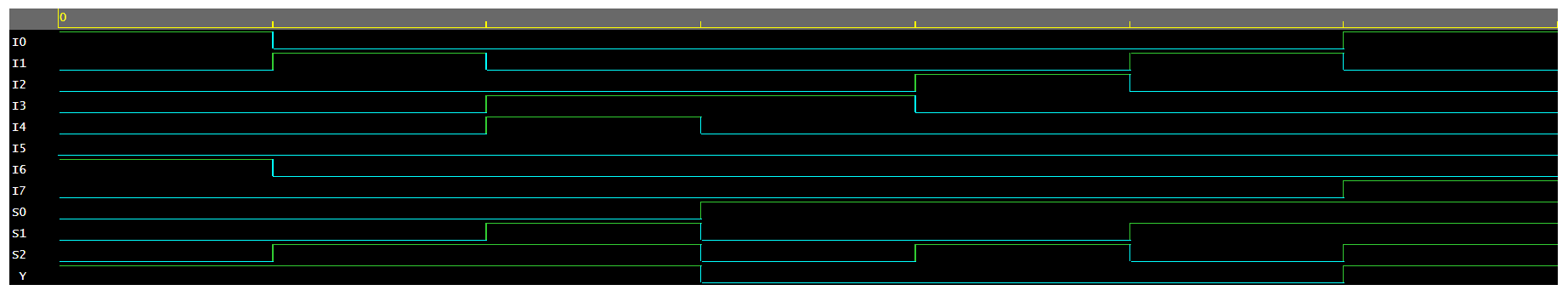
Question 2: Write a Verilog code to design an 8:1 Multiplexer and verify the same.

Code:

|  |  |
| --- | --- |
| // Code your testbench here  module tb\_MUX;  reg S0,S1,S2,I0,I1,I2,I3,I4,I5,I6,I7;  wire Y;  MUX M4(S0,S1,S2,I0,I1,I2,I3,I4,I5,I6,I7,Y);    initial  begin  S0=0;S1=0;S2=0;I0=1;I1=0;I2=0;I3=0;I4=0;I5=0;I6=1;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=0;S1=0;S2=1;I0=0;I1=1;I2=0;I3=0;I4=0;I5=0;I6=0;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=0;S1=1;S2=1;I0=0;I1=0;I2=0;I3=1;I4=1;I5=0;I6=0;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=1;S1=0;S2=0;I0=0;I1=0;I2=0;I3=1;I4=0;I5=0;I6=0;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=1;S1=0;S2=1;I0=0;I1=0;I2=1;I3=0;I4=0;I5=0;I6=0;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=1;S1=1;S2=0;I0=0;I1=1;I2=0;I3=0;I4=0;I5=0;I6=0;I7=0; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  S0=1;S1=1;S2=1;I0=1;I1=0;I2=0;I3=0;I4=0;I5=0;I6=0;I7=1; #1  $display("S0=%b S1=%b S2=%b",S0,S1,S2);  $display("Y=%b",Y);  end  initial  begin  $dumpfile("dump.vcd");  $dumpvars(1);  end  endmodule | // Code your design here  module MUX(s0,s1,s2,A0,A1,A2,A3,A4,A5,A6,A7,y);  input s0,s1,s2;  input A0,A1,A2,A3,A4,A5,A6,A7;  output y;  wire s0\_not,s1\_not,s2\_not;  wire B0,B1,B2,B3,B4,B5,B6,B7;  not(s0\_not,s0);  not(s1\_not,s1);  not(s2\_not,s2);  and(B0,s0\_not,s1\_not,s2\_not,A0);  and(B1,s0\_not,s1\_not,s2,A1);  and(B2,s0\_not,s1,s2\_not,A2);  and(B3,s0\_not,s1,s2,A3);  and(B4,s0,s1\_not,s2\_not,A4);  and(B5,s0,s1\_not,s2,A5);  and(B6,s0,s1,s2\_not,A6);  and(B7,s0,s1,s2,A7);  or(y,B0,B1,B2,B3,B4,B5,B6,B7);  endmodule |

Output:

S0=0 S1=0 S2=0  
Y=1  
S0=0 S1=0 S2=1  
Y=1  
S0=0 S1=1 S2=1  
Y=1  
S0=1 S1=0 S2=0  
Y=0  
S0=1 S1=0 S2=1  
Y=0  
S0=1 S1=1 S2=0  
Y=0  
S0=1 S1=1 S2=1  
Y=1  
Finding VCD file...  
./dump.vcd



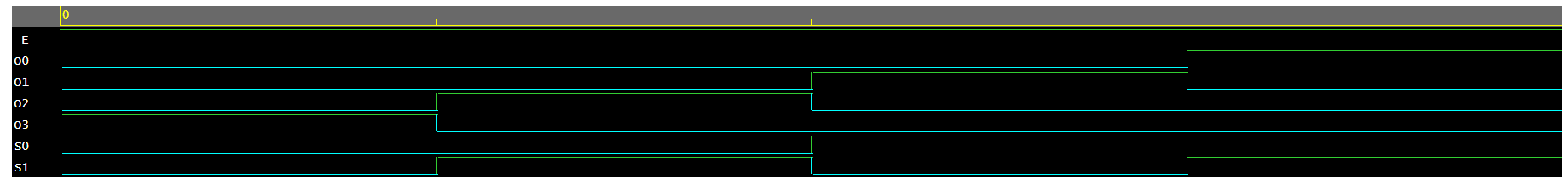
Question 3: Write a Verilog code to design a 1:4 Demultiplexer and verify the same.

Code:

|  |  |
| --- | --- |
| module tb\_DEMUX;  reg S0,S1,E;  wire O0,O1,O2,O3;  DEMUX DM(S0,S1,E,O0,O1,O2,O3);  initial  begin  S0=0; S1=0; E=1; #1  $display("S0=%b S1=%b E=%b",S0,S1,E);  $display("O0=%b O1=%b O2=%b O3=%b",O0,O1,O2,O3);  S0=0; S1=1; E=1;#1  $display("S0=%b S1=%b E=%b",S0,S1,E);  $display("O0=%b O1=%b O2=%bO3=%b",O0,O1,O2,O3);  S0=1; S1=0; E=1;#1  $display("S0=%b S1=%b E=%b",S0,S1,E);  $display("O0=%b O1=%b O2=%b O3=%b",O0,O1,O2,O3);  S0=1; S1=1; E=1;#1  $display("S0=%b S1=%b E=%b",S0,S1,E);  $display("O0=%b O1=%b O2=%b O3=%b",O0,O1,O2,O3);  end  initial  begin  $dumpfile("dump.vcd");  $dumpvars(1);  end  endmodule | // Code your design here  module DEMUX(input s0,s1,E, output o0,o1,o2,o3);  wire s0\_not,s1\_not,x,y,z,w;  not(s0\_not,s0);  not(s1\_not,s1);  and(o0,s0,s1,E);  and(o1,s0,s1\_not,E);  and(o2,s0\_not,s1,E);  and(o3,s0\_not,s1\_not,E);  endmodule |

Output:

VCD info: dumpfile dump.vcd opened for output.  
S0=0 S1=0 E=1  
O0=0 O1=0 O2=0 O3=1  
S0=0 S1=1 E=1  
O0=0 O1=0 O2=1O3=0  
S0=1 S1=0 E=1  
O0=0 O1=1 O2=0 O3=0  
S0=1 S1=1 E=1  
O0=1 O1=0 O2=0 O3=0  
Finding VCD file...  
./dump.vcd



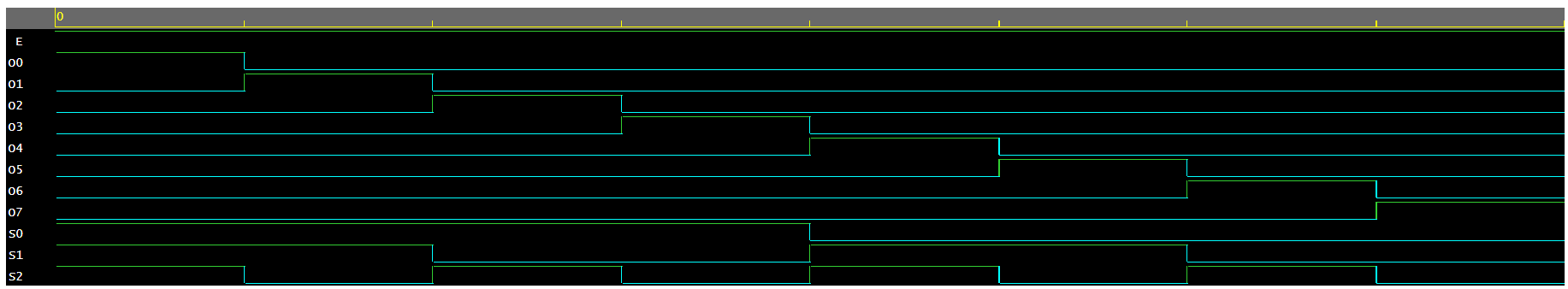
Question 4: Write a Verilog code to design a 1:8 Demultiplexer and verify the same

Code:

|  |  |
| --- | --- |
| module tb\_DEMUX;  reg S0,S1,S2,E;  wire O0,O1,O2,O3,O4,O5,O6,O7;  DEMUX DM(S0,S1,S2,E,O0,O1,O2,O3,O4,O5,O6,O7);  initial  begin  S0=1; S1=1; S2=1; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=1; S1=1; S2=0; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=1; S1=0; S2=1; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=1; S1=0; S2=0; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=0; S1=1; S2=1; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=0; S1=1; S2=0; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=0; S1=0; S2=1; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);    S0=0; S1=0; S2=0; E=1; #1  $display("S0=%b S1=%b S2=0 E=%b",S0,S1,S2,E);  $display("O0=%b O1=%b O2=%b O3=%b O4=%b O5=%b O6=%b O7=%b",O0,O1,O2,O3,O4,O5,O6,O7);  end  initial  begin  $dumpfile("dump.vcd");  $dumpvars(1);  end  endmodule | // Code your design here  module DEMUX(s0,s1,s2,E,A0,A1,A2,A3,A4,A5,A6,A7);  input s0,s1,s2,E;  output A0,A1,A2,A3,A4,A5,A6,A7;  wire s0\_not,s1\_not,s2\_not;  not(s0\_not,s0);  not(s1\_not,s1);  not(s2\_not,s2);  and(A0,s0,s1,s2,E);  and(A1,s0,s1,s2\_not,E);  and(A2,s0,s1\_not,s2,E);  and(A3,s0,s1\_not,s2\_not,E);  and(A4,s0\_not,s1,s2,E);  and(A5,s0\_not,s1,s2\_not,E);  and(A6,s0\_not,s1\_not,s2,E);  and(A7,s0\_not,s1\_not,s2\_not,E);  endmodule |

Output.

VCD info: dumpfile dump.vcd opened for output.  
S0=1 S1=1 S2=0 E=11  
O0=1 O1=0 O2=0 O3=0 O4=0 O5=0 O6=0 O7=0  
S0=1 S1=1 S2=0 E=01  
O0=0 O1=1 O2=0 O3=0 O4=0 O5=0 O6=0 O7=0  
S0=1 S1=0 S2=0 E=11  
O0=0 O1=0 O2=1 O3=0 O4=0 O5=0 O6=0 O7=0  
S0=1 S1=0 S2=0 E=01  
O0=0 O1=0 O2=0 O3=1 O4=0 O5=0 O6=0 O7=0  
S0=0 S1=1 S2=0 E=11  
O0=0 O1=0 O2=0 O3=0 O4=1 O5=0 O6=0 O7=0  
S0=0 S1=1 S2=0 E=01  
O0=0 O1=0 O2=0 O3=0 O4=0 O5=1 O6=0 O7=0  
S0=0 S1=0 S2=0 E=11  
O0=0 O1=0 O2=0 O3=0 O4=0 O5=0 O6=1 O7=0  
S0=0 S1=0 S2=0 E=01  
O0=0 O1=0 O2=0 O3=0 O4=0 O5=0 O6=0 O7=1  
Finding VCD file...  
./dump.vcd



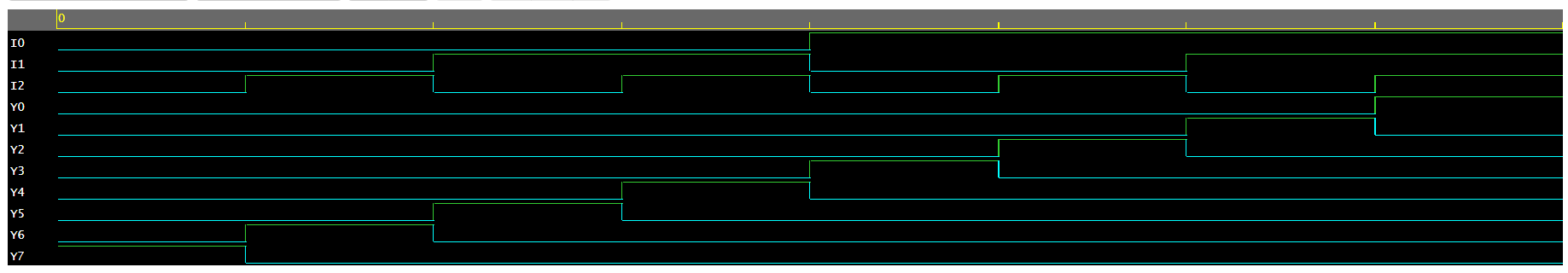
Question 5: Write a Verilog code to design a 3:8 Decoder and verify the same.

Code:

|  |  |
| --- | --- |
| module tb\_Decoder;  reg I0,I1,I2;  wire Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7;  Decoder DM(I0,I1,I2,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    initial  begin  I0=0;I1=0;I2=0; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=0;I1=0;I2=1; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=0;I1=1;I2=0; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=0;I1=1;I2=1; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=1;I1=0;I2=0; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=1;I1=0;I2=1; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=1;I1=1;I2=0; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);    I0=1;I1=1;I2=1; #1  $display("I0=%b I1=%b I2=%b",I0,I1,I2);  $display("Y0=%b Y1=%b Y2=%b Y3=%bY4=%b Y5=%b Y6=%b Y7=%b",Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);  end  initial  begin  $dumpfile("dump.vcd");  $dumpvars(1);  end  endmodule | // Code your design here  module Decoder(B0,B1,B2,A0,A1,A2,A3,A4,A5,A6,A7);  input B0,B1,B2;  output A0,A1,A2,A3,A4,A5,A6,A7;  wire B0\_not,B1\_not,B2\_not;  not(B0\_not,B0);  not(B1\_not,B1);  not(B2\_not,B2);  and(A0,B0,B1,B2);  and(A1,B0,B1,B2\_not);  and(A2,B0,B1\_not,B2);  and(A3,B0,B1\_not,B2\_not);  and(A4,B0\_not,B1,B2);  and(A5,B0\_not,B1,B2\_not);  and(A6,B0\_not,B1\_not,B2);  and(A7,B0\_not,B1\_not,B2\_not);  endmodule |

Output:

VCD info: dumpfile dump.vcd opened for output.  
I0=0 I1=0 I2=0  
Y0=0 Y1=0 Y2=0 Y3=0Y4=0 Y5=0 Y6=0 Y7=1  
I0=0 I1=0 I2=1  
Y0=0 Y1=0 Y2=0 Y3=0Y4=0 Y5=0 Y6=1 Y7=0  
I0=0 I1=1 I2=0  
Y0=0 Y1=0 Y2=0 Y3=0Y4=0 Y5=1 Y6=0 Y7=0  
I0=0 I1=1 I2=1  
Y0=0 Y1=0 Y2=0 Y3=0Y4=1 Y5=0 Y6=0 Y7=0  
I0=1 I1=0 I2=0  
Y0=0 Y1=0 Y2=0 Y3=1Y4=0 Y5=0 Y6=0 Y7=0  
I0=1 I1=0 I2=1  
Y0=0 Y1=0 Y2=1 Y3=0Y4=0 Y5=0 Y6=0 Y7=0  
I0=1 I1=1 I2=0  
Y0=0 Y1=1 Y2=0 Y3=0Y4=0 Y5=0 Y6=0 Y7=0  
I0=1 I1=1 I2=1  
Y0=1 Y1=0 Y2=0 Y3=0Y4=0 Y5=0 Y6=0 Y7=0  
Finding VCD file...  
./dump.vcd

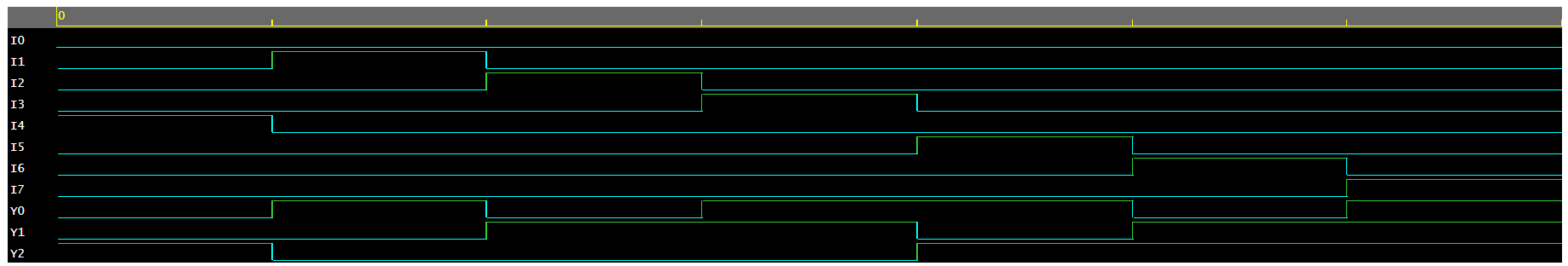


Question 6: Write a Verilog code to design a 8:3 Encoder and verify the same.

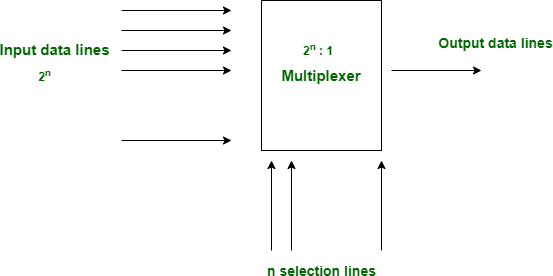
Code:

Output:

VCD info: dumpfile dump.vcd opened for output.  
I0=0 I1=0 I2=0 I3=0 I4=1 I5=0 I6=0 I7=0  
Y0=0 Y1=0 Y2=1  
I0=0 I1=1 I2=0 I3=0 I4=0 I5=0 I6=0 I7=0  
Y0=1 Y1=0 Y2=0  
I0=0 I1=0 I2=1 I3=0 I4=0 I5=0 I6=0 I7=0  
Y0=0 Y1=1 Y2=0  
I0=0 I1=0 I2=0 I3=1 I4=0 I5=0 I6=0 I7=0  
Y0=1 Y1=1 Y2=0  
I0=0 I1=0 I2=0 I3=0 I4=0 I5=1 I6=0 I7=0  
Y0=1 Y1=0 Y2=1  
I0=0 I1=0 I2=0 I3=0 I4=0 I5=0 I6=1 I7=0  
Y0=0 Y1=1 Y2=1  
I0=0 I1=0 I2=0 I3=0 I4=0 I5=0 I6=0 I7=1  
Y0=1 Y1=1 Y2=1  
Finding VCD file...  
./dump.vcd

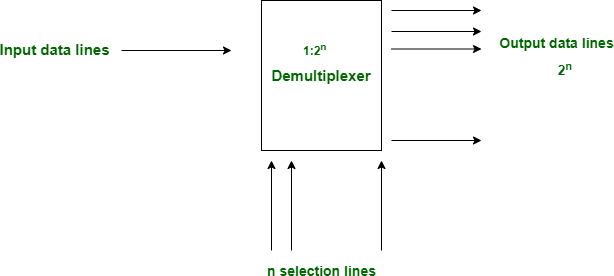


* Multiplexer is a data selector which takes several inputs and gives a single output. In multiplexer we have 2n input lines and 1 output lines where n is the number of selection lines.



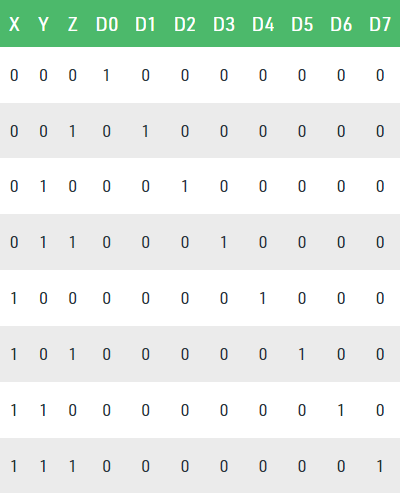
*Figure 1: The Block Diagram of Multiplexer.*

* Demultiplexer is a data distributor which takes a single input and gives several outputs. In demultiplexer we have 1 input and 2n output lines where n is the selection line.



*Figure 2: The Block Diagram of Multiplexer.*

* A decoder does the opposite job of an encoder. It is a combinational circuit that converts n lines of input into 2n lines of output.



*Figure 3: Truth Table of Decoder.*

* An encoder is a combinational circuit that converts binary information in the form of a 2N input lines into N output lines, which represent N bit code for the input. For simple encoders, it is assumed that only one input line is active at a time.

