**Amrita Vishwa Vidyapeetham**

**Amrita School of Engineering, Coimbatore**

**Department of Computer Science and Engineering**

**Topic: Logic Circuit Implemetation**

**--------------------------------------------------------------------------------------------------------------------------**

**Sub Code: 19CS211 Sub Title: COA**

**Roll No: CB.EN.U4CSE19453 Name: R.ABHINAV**

**Lab Evaluation No: 2 Date: 08-02-2021**

1. Implement the concept of 4 x 1 multiplexer by using 2 to 4 decoder as given below.

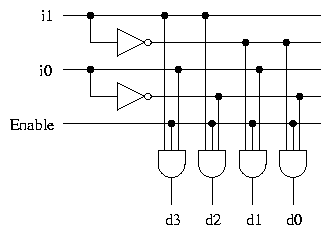
**The Decoder circuit :**

* The following circuit generates all four minterms from two inputs,

and implements the 2-4 decoder.

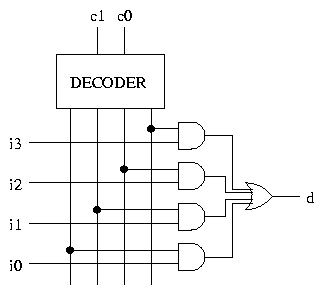
* The Enable input is fed into the AND gates which produce the outputs.
* Many components have an Enable input which works in this way.
* Sometimes the Enable input is active high (as in this case); sometimes

it is active low.



**The Decoder as a Selector for 4 x 1 mux :**

* All the outputs of the decoder are 0, apart from one. The inputs c1c0 determine which of the outputs is non-zero.
* All but one of the AND gates have 0 on one input and therefore output 0. The remaining AND gate has 1 on one input and in (where n is represented in binary by c1c0) on the other input. The output of this AND gate is the value of in.
* The OR gate has 0 on all of its inputs apart from one, and has the value of in on the remaining input. The output of the OR gate is therefore the value of in.



**Logic Equation :**

d = c0'.c1'.i3 + c0'.c1.i2 + c0.c1.'i1 + c0.c1.i0

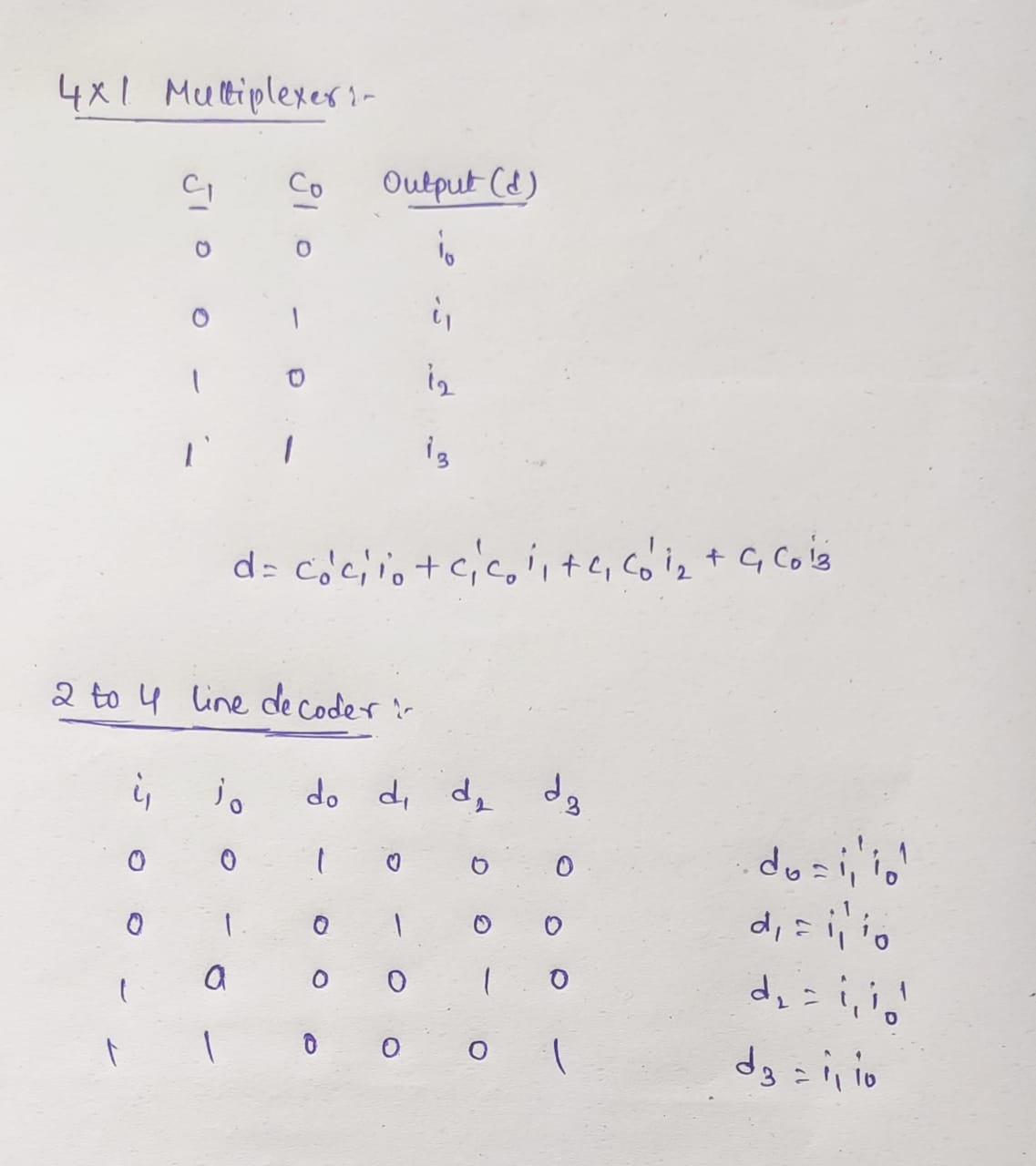
* d is the output.
* For decoder :

d0=i0'i1'

d1=i0i1'

d2=i0'i1

d3=i0i1



**iVerilog Code:**

module muxeval(a,b,c,d,d0,d1,d2,d3,i1,i0,z);

input i1,i0,a,b,c,d;

output d0,d1,d2,d3,z;

assign d0=(!i1)&(!i0);

assign d1=(!i1)&i0;

assign d2=(i1)&(!i0);

assign d3=i1&i0;

assign z=(d0&a)|(d1&b)|(d2&c)|(d3&d);

endmodule

**TestBench :**

module muxeval\_tb;

wire d0,d1,d2,d3,z;

reg i1,i0,a,b,c,d;

muxeval my\_muxeval(.d0(d0),.d1(d1),.d2(d2),.d3(d3),.a(a),.b(b),.c(c),.d(d),.z(z),.i0(i0),.i1(i1));

initial

begin

$monitor("i1=",i1," ","i0=",i0," ","a=",a," ","b=",b," ","c=",c," ","d=",d," ","z=",z);

i1=1'b0;

i0=1'b0;

a=1'b1;

b=1'b0;

c=1'b0;

d=1'b0;

**Test bench:**

#20

i1=1'b0;

i0=1'b1;

a=1'b0;

b=1'b1;

c=1'b0;

d=1'b0;

#5

i1=1'b1;

i0=1'b0;

a=1'b0;

b=1'b0;

c=1'b1;

d=1'b0;

#5

i1=1'b1;

i0=1'b1;

a=1'b0;

b=1'b0;

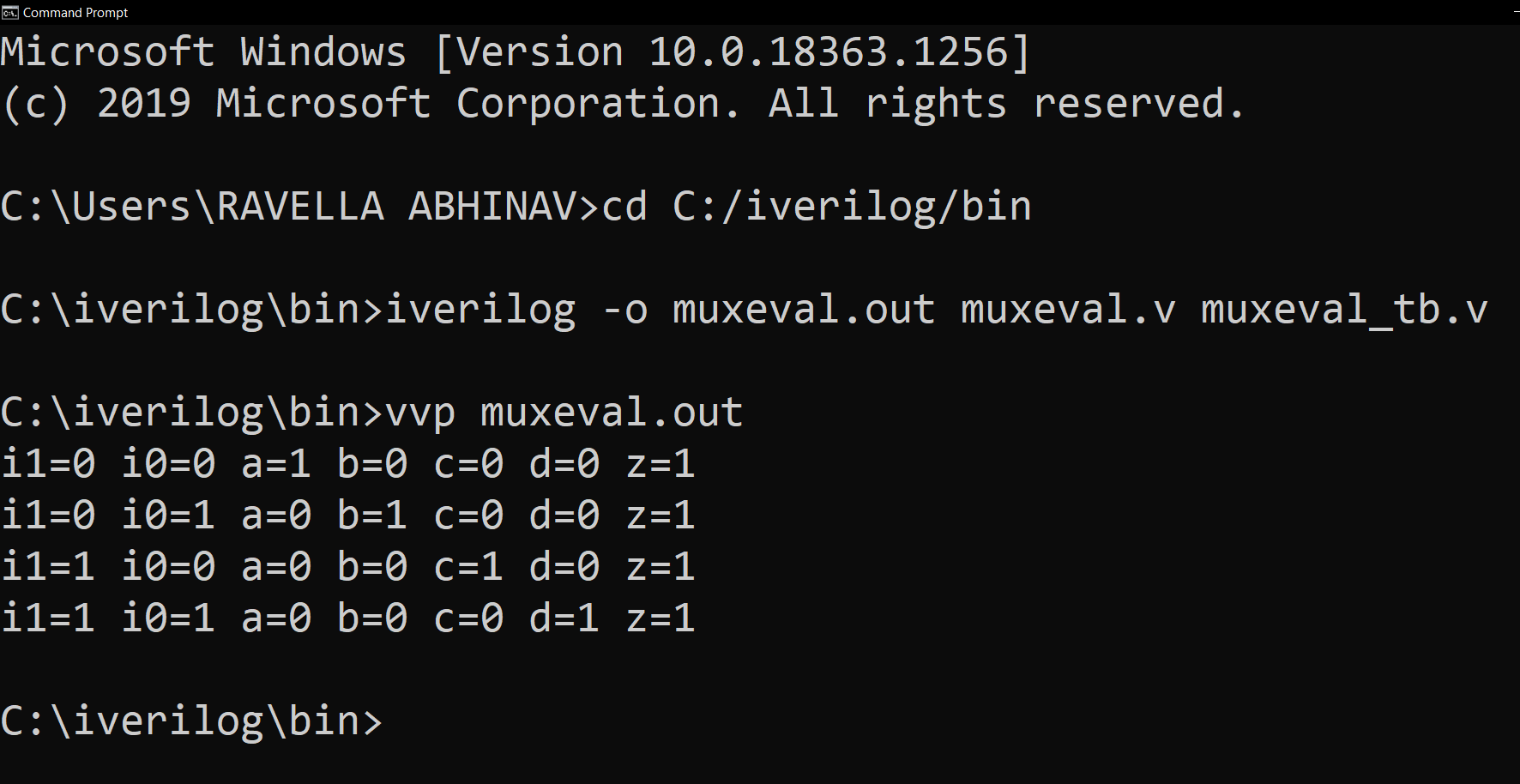
c=1'b0;

d=1'b1;

end

endmodule

**Output:**

****