**Amrita Vishwa Vidyapeetham**

**Amrita School of Engineering, Coimbatore**

**Department of Computer Science and Engineering**

**Topic: Periodical Test 2 Lab Evaluation**

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

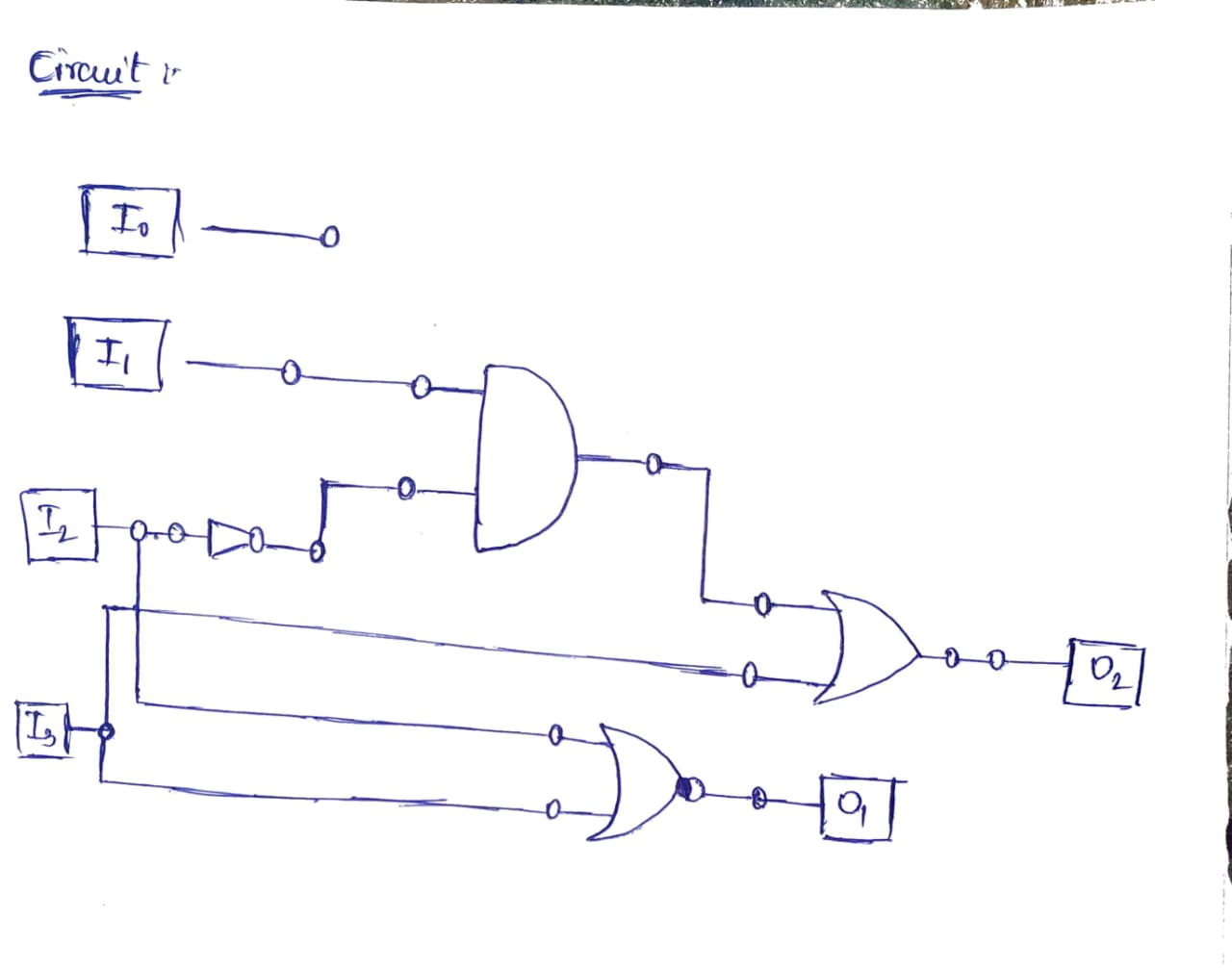
**Sub Code: 19CS211 Sub Title: COA**

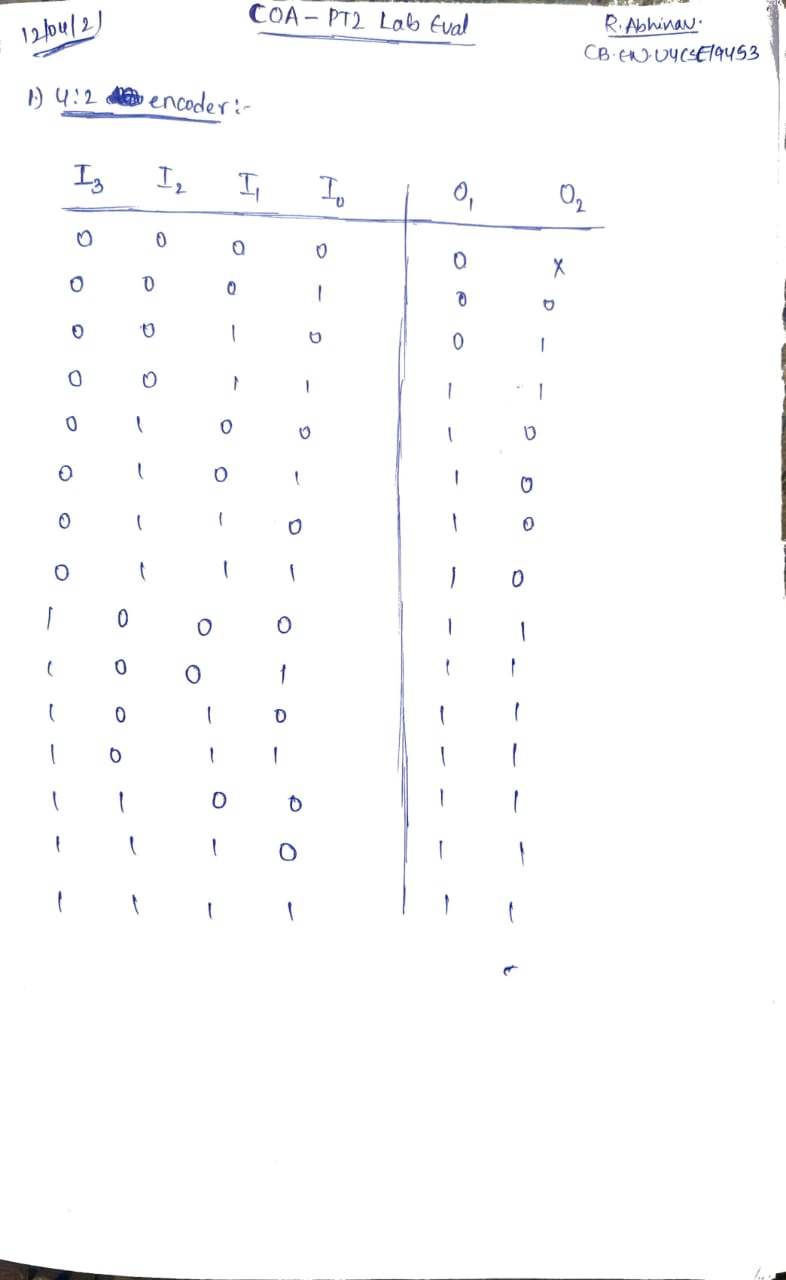
**Roll No: CB.EN.U4CSE19453 Name: R.ABHINAV**

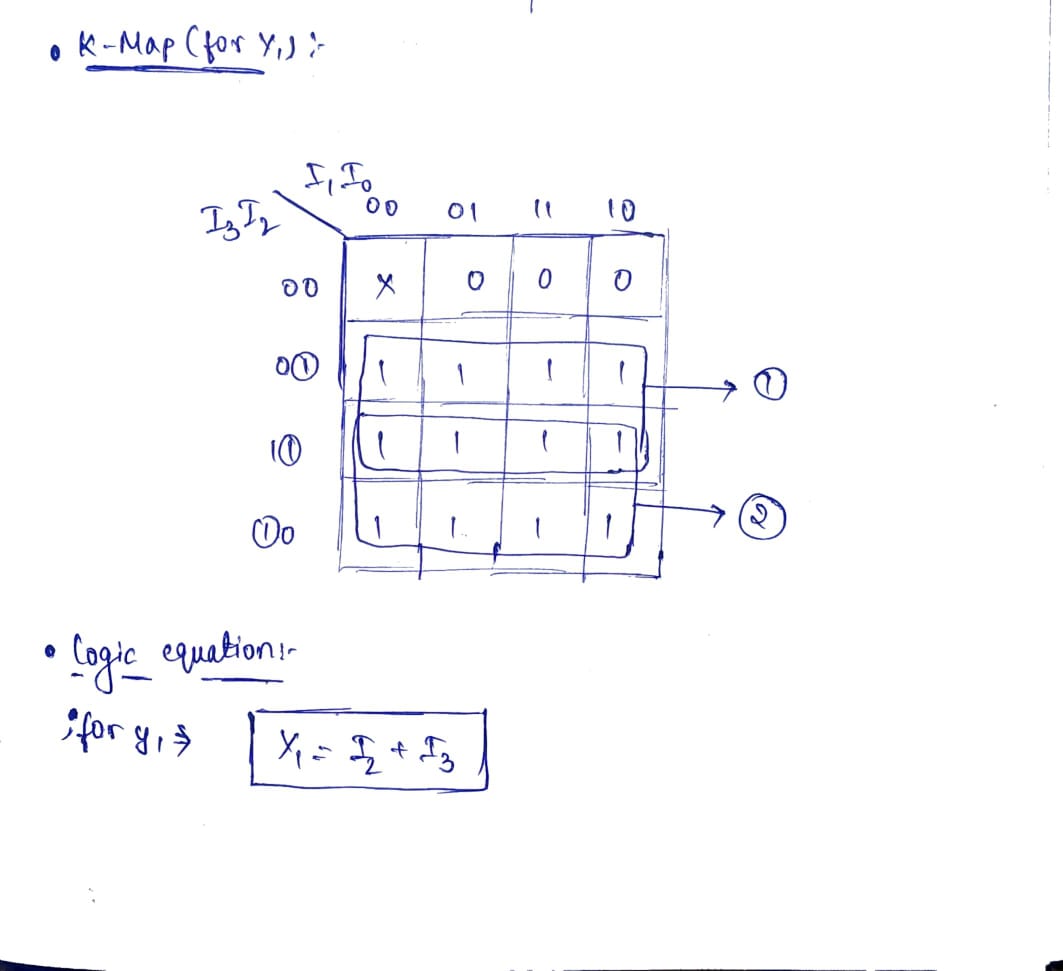
**Lab Evaluation No: PT2 Date: 12-04-2021**

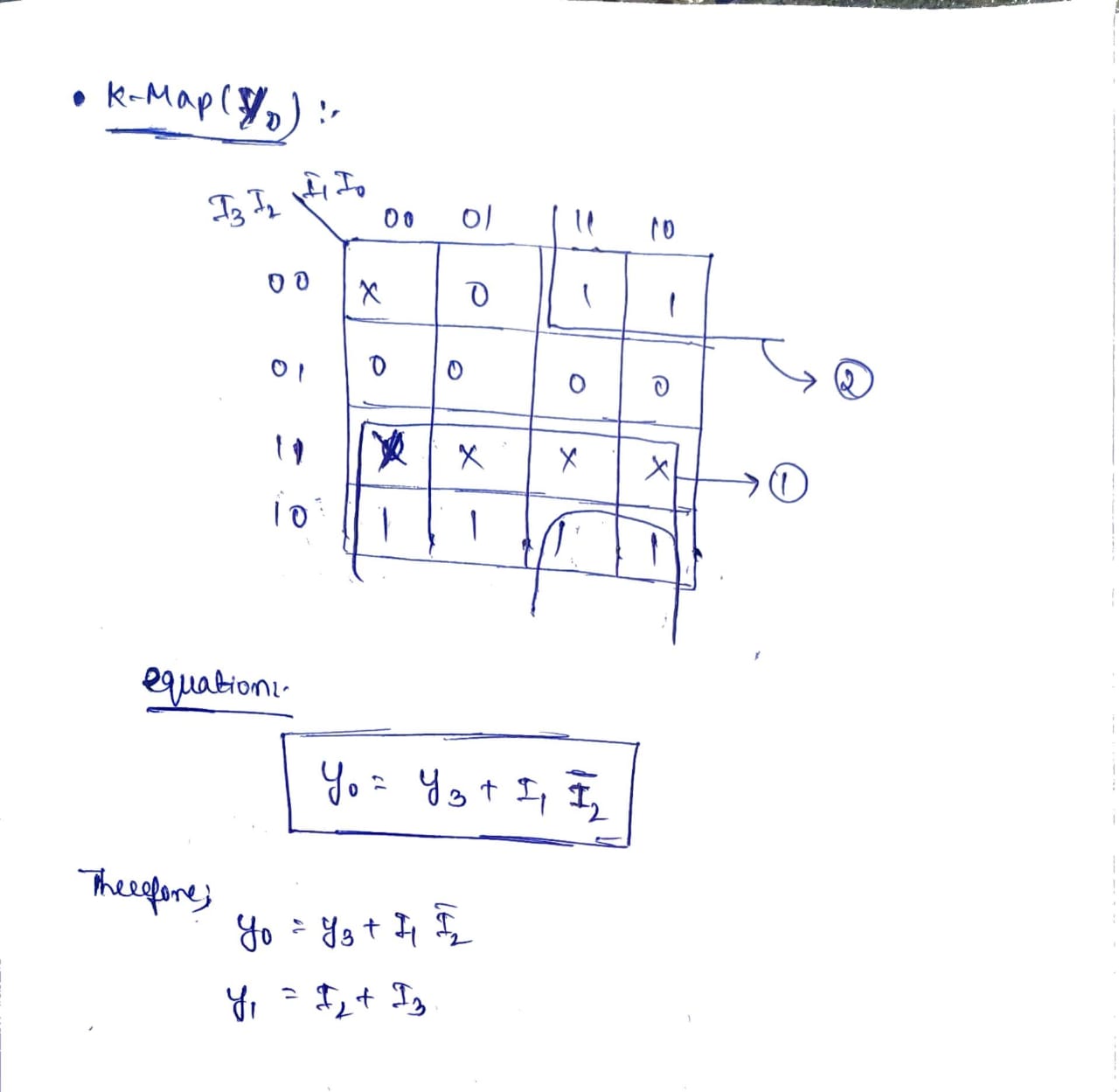
1. **4:2 priority encoder:**

A 4-to-2 priority encoder takes 4 input bits and produces 2 output bits. In this truth table, for all the non-explicitly defined input combinations (i.e. inputs containing 2, 3, or 4 high bits) the lower priority bits are shown as don't cares (X). Similarly when the inputs are 0000, the outputs are not valid and therefore they are XX.









* **iVerilog Code :**

module encoder42(I3,I2,I1,I0,O0,O1);

input I3, I2, I1, I0;

output O0, O1;

and(and\_out, y2bar, y1);

or(O1, I3, I2);

or(O0, (!I2)&I1, I3);

endmodule

* **Test Bench:**

module encoder42\_tb;

reg I3,I2,I1,I0;

wire O0, O1;

encoder42 my\_pr42(.I3(I3),.I2(I2),.I1(I1),.I0(I0), .O0(O0), .O1(O1));

initial

begin

$monitor("I3=%b, I2=%b, I1=%b, I0=%b %b %b", I3, I2, I1, I0, O1, O0);

I3=1'b0;

I2=1'b0;

I1=1'b0;

I0=1'b0;

#10

I3=1'b0;

I2=1'b0;

I1=1'b0;

I0=1'b1;

#10

I3=1'b0;

I2=1'b0;

I1=1'b1;

I0=1'bx;

#10

I3=1'b0;

I2=1'b1;

I1=1'bx;

I0=1'bx;

#10

I3=1'b1;

I2=1'bx;

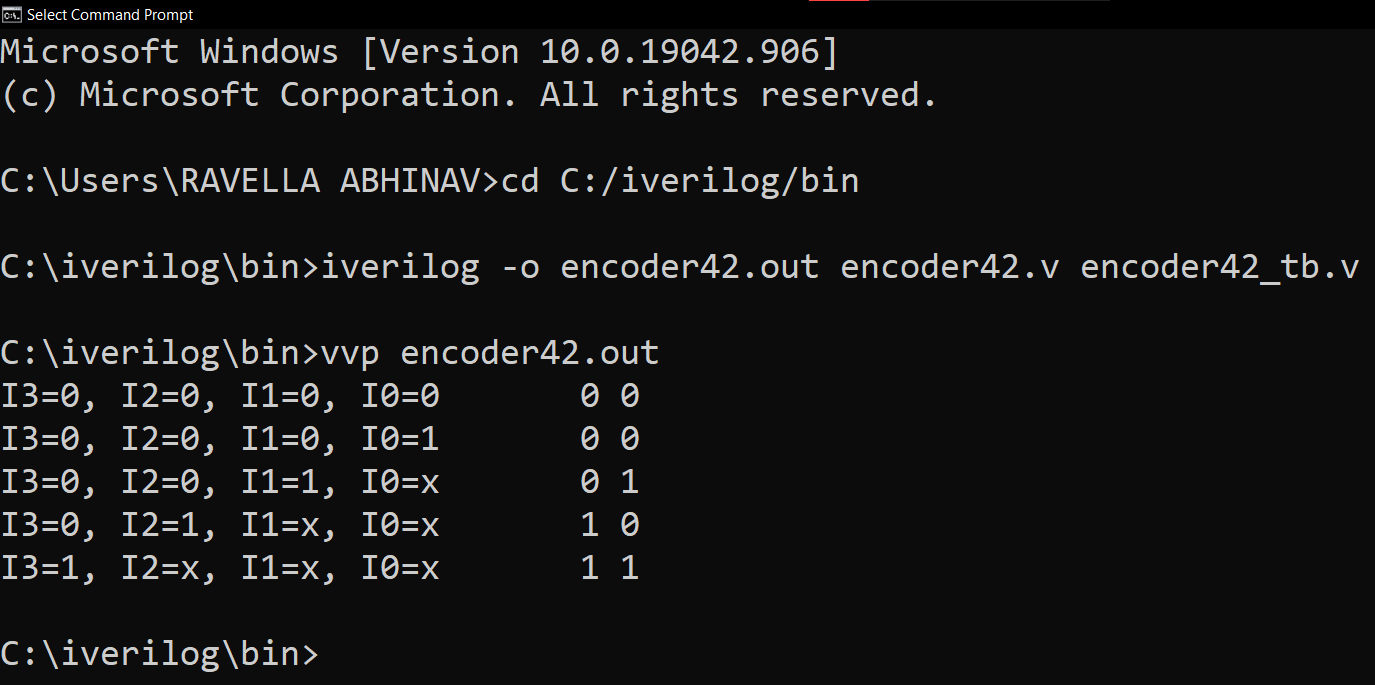
I1=1'bx;

I0=1'bx;

end

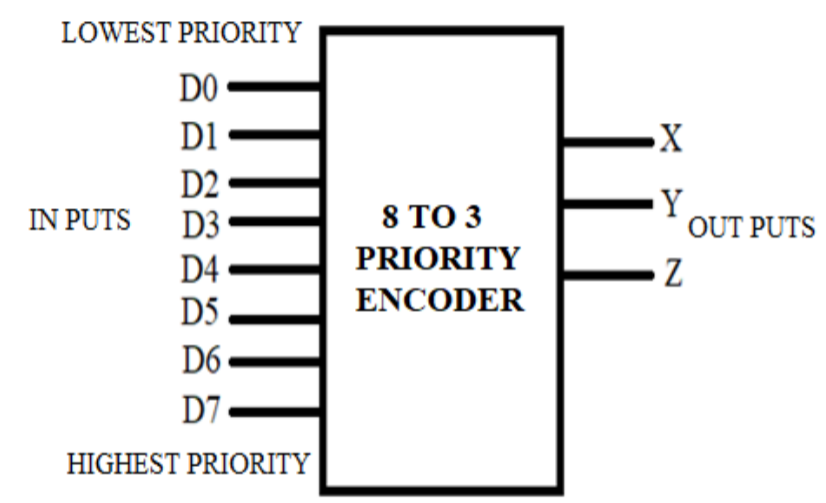
endmodule

**Output :**

****

1. **8:3 priority encoder:**

* **Block Diagram:**

****

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

* **Logic Equation :**
* **x=D4+D5+D6+D7**
* **y=D2+D3+D6+D7**
* **z=D1+D3+D5+D7**
* **Truth Table:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Digital Inputs** | | | | | | | | **Binary Outputs** | | |
| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | **X** | **Y** | **Z** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | X | X | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | X | X | X | X | 1 | 0 | 0 |
| 0 | 0 | 1 | X | X | X | X | X | 1 | 0 | 1 |
| 0 | 1 | X | X | X | X | X | X | 1 | 1 | 0 |
| 1 | X | X | X | X | X | X | X | 1 | 1 | 1 |

**Iverilog code :**

module encoder\_8\_3(o,i);

output reg [2:0]o;

input [7:0]i;

always @ (\*)

case(i)

8'h01: o=3'b000;

8'h02: o=3'b001;

8'h04: o=3'b010;

8'h08: o=3'b011;

8'h10: o=3'b100;

8'h20: o=3'b101;

8'h40: o=3'b110;

8'h80: o=3'b111;

default: o=3'bxxx;

endcase

endmodule

**Test Bench:**

module testbench\_encoder;

//Inputs

reg [7:0] i;

//Outputs

wire [2:0 ] o;

// Instantiate the Unit Under Test (UUT)

encoder\_8\_3 uut (

.o(o),

.i(i)

);

initial begin

i=8'h01;

#2 i=8'h02;

#2 i=8'h04;

#2 i=8'h08;

#2 i=8'h10;

#2 i=8'h20;

#2 i=8'h40;

#2 i=8'h80;

end

initial #18 $finish;

endmodule