

MINI PROJECT

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 27.11.2022 16:49:51
// Design Name:
// Module Name: main
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module main(
    input clk,
    input I0,
    input I1,
    input I2,
    input I3,
    output wire data,
    output reg Q0,
    output reg Q1,
    output reg Q2,
    output reg Q3
);

    reg Y0;
    reg Y1;
    reg Y2;
    reg Y3;
    wire [1:0] s;

    // clock
    c_counter_binary_0 counter(
        .CLK(clk),
        .Q(s)
    );
    //vio
    vio_3 myvoi(
        .clk(clk),          // input wire clk
        .probe_out0(I0),    // input wire [0 : 0] probe_in0
```

```

.probe_out1(I1), // input wire [0 : 0] probe_in1
.probe_out2(I2), // input wire [0 : 0] probe_in2
.probe_out3(I3) // input wire [0 : 0] probe_in3
);

//4:1 mux datalow
assign data = s[1] ? (s[0] ? I3 : I2) : (s[0] ? I1 : I0);

//2:4 decoder behavioural modeling
always @(posedge clk)
begin

    // using condition if statement
    // implement the 2:4 truth table
    if (!data)
    begin
        Y0<=0;
        Y1<=1;
        Y2<=2;
        Y3<=0;
    end
    else
    begin
        case (s)
            2'b00: Y3 = 4'b0001;
            2'b01: Y2 = 4'b0010;
            2'b10: Y1 = 4'b0100;
            2'b11: Y0 = 4'b1000;
            default: $display("Error!");
        endcase
    end
end

//latch
// Data input

always @(posedge clk)
begin
    Q0<=Y0;
    Q1<=Y1;
    Q2<=Y2;
    Q3<=Y3;
end
endmodule

```