

M8:1

S	Y
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

Tabla 01

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Mux 4:1 & Compuertas

S	Y
00	C
01	C'
10	C'
11	C

Mux 2:1 & Compuertas

S	Y
0	$B \wedge C$
1	$B \sim \wedge C$

Tabla 02

A	B	C	Y
0	0	0	1
0	0	1	x
0	1	0	0
0	1	1	0
1	0	0	x
1	0	1	1
1	1	0	1
1	1	1	0

Mux 4:1 & Compuertas

S	Y
00	C'
01	0
10	C
11	C'

Mux 2:1 & Compuertas

S	Y
0	$B \sim   C$
1	$B \wedge C$

### Ejercicio 5

Propagation Delay: Es el tiempo de delay máximo en el que una compuerta tarda en cambiar y volver a estabilizarse

Contamination Delay: Es el tiempo mínimo de delay en el que una compuerta tarda en desestabilizarse o mejor dicho de iniciar el cambio

Ruta Crítica: es el tiempo más grande que en tarda de llegar un cambio desde una entrada a una salida en cierta ruta del circuito

Ruta Corta: es el menor tiempo que tarda en llegar un cambio desde una entrada a una salida en el circuito

### Ejercicio 6

Circuito 1	tpd	tcd
T2-F1	80	55
T1-F1	125	85
AB-F1	155	120
AC-F1	155	120
BC-F1	155	120
AB-F2	70	55
AC-F2	70	55
BC-F2	70	55
Ruta Crítica: AB-F1, AC-F1, BC-F1		
Ruta Corta: T2-F1, AB-F2, AC-F2, BC-F2		

---

Circuito 2	tpd	tcd
A-F	75	60
A-F2	85	65
A-F3	85	65
A-G	85	65
A-G2	65	50
BC-F o G	100	80
D-F	70	55
D-G	50	40
Ruta Crítica: BC-F o G,		
Ruta Corta: D-G		

Circuito 3	tpd	tcd
A-T3-F1	80	60
A-T2-F1	145	105
A-T2-F2	85	65
B-T1-T3-F1	125	95
B-T2-T4-F1	130	95
B-T2-F2	70	55
C-T1-T3-F1	110	85
D-T4-F1	100	70
D-F2	55	40
Ruta Crítica: A-T2-F1		
Ruta Corta: D-F2		

Circuito 3	tpd	tcd
A-W	40	30
B-X	70	55
B-Not-X	85	65
B-W	70	55
C-CD-Y	70	55
C-(C+D)'-Y	95	70
C-(C+D)'-X	125	95
C-(C+D)-X	110	85
C-(C+D)-W	110	85
D-Z	15	10
D-CD-Y	70	55
D-(C+D)'-Y	95	70
D-(C+D)'-X	125	95
D-(C+D)-X	110	85
D-(C+D)-W	110	85
Ruta Crítica: C-(C+D)'-X, D-(C+D)'-X		
Ruta Corta: A-W		

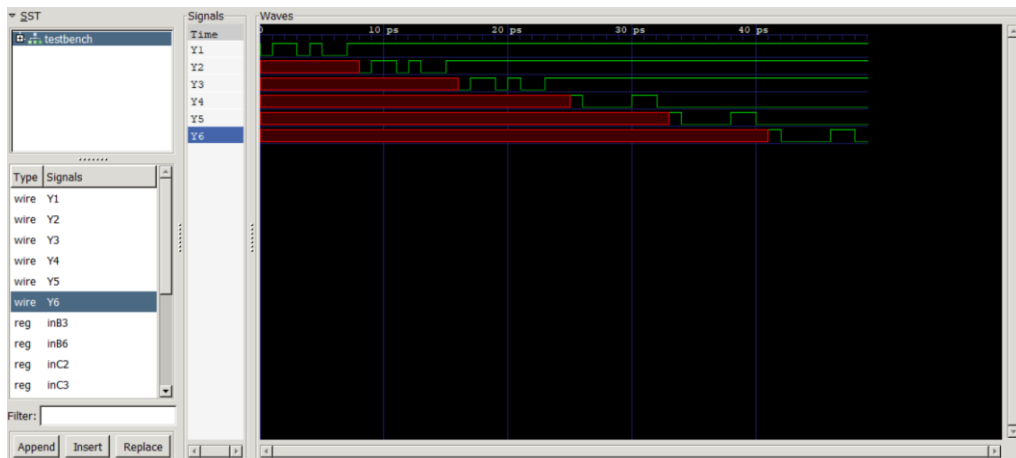


Tabla 1 Mux8 a 1	Tabla 1 Mux4 a 1	Tabla 1 Mux2 a 1	Tabla 2 Mux8 a 1	Tabla 2 Mux4 a 1	Tabla 1 Mux2 a 1
ABC   Y	ABC   Y	ABC   Y	ABC   Y	ABC   Y	ABC   Y
-----	-----	-----	-----	-----	-----
VCD info: d					
000   0	000   0	000   0	000   1	000   1	000   1
001   1	001   1	001   1	001   0	001   0	001   0
010   1	010   1	010   1	010   0	010   0	010   0
011   0	011   0	011   0	011   0	011   0	011   0
100   1	100   1	100   1	100   0	100   0	100   0
101   0	101   0	101   0	101   1	101   1	101   1
110   0	110   0	110   0	110   1	110   1	110   1
111   1	111   1	111   1	111   0	111   0	111   0

```
//Diego A. Méndez
//19673
//Ejercicio 04

module mux2_1(input wire inA, inB, inS, output wire Y);

    assign Y=inS?inB:inA;

endmodule

module mux4_1(input wire inA, inB, inC, inD, input wire[1:0] inS, output wire Y);
//Creacion de un mux 4 a 1 basado
    wire Y1, Y2;
    mux2_1 U1(inA, inB, inS[0], Y1);
    mux2_1 U2(inC, inD, inS[0], Y2);
    mux2_1 U3(Y1, Y2, inS[1], Y);

endmodule

module mux8_1(input wire inA, inB, inC, inD, inE, inF, inG, inH, input wire[2:0] inS, output wire Y);
//Creacion de un mux 8 a 1 basado
    wire Y1, Y2;
    mux4_1 U1(inA, inB, inC, inD, inS[1:0], Y1);
    mux4_1 U2(inE, inF, inG, inH, inS[1:0], Y2);
    mux2_1 U3(Y1, Y2, inS[2], Y);
```

```

module tabla1mux8(input wire[2:0] inS, output wire Y);
//modulo para mux 8 tabla 1

//registrar cables
wire inA1, inB1, inC1, inD1, inE1, inF1, inG1, inH1;

//asignar valores
assign inA1=0; assign inB1=1; assign inC1=1; assign inD1=0;
assign inE1=1; assign inF1=0; assign inG1=0; assign inH1=1;

mux8_1 U1(inA1, inB1, inC1, inD1, inE1, inF1, inG1, inH1, inS, Y);
endmodule

module tabla1mux4(input wire inC, input wire[1:0] inS, output wire Y);
//aplicando la primera tabla con un mux4a1 y compuertas
wire NC;

not N1(NC,inC);
// la compuerta not nos sirve para trabajar unicamente con una entrada C y el mux 4
mux4_1 U1(inC,NC,NC,inC,inS, Y);
endmodule

```

```

module tabla1mux2(input wire inS, inB, inC, output wire Y);
//aplicacion de mux2 en tabla1
    wire Y1, Y2;

    assign Y1= inB ^ inC ;
    assign Y2= inB ~^ inC;

    mux2_1 U1(Y1, Y2, inS, Y);

endmodule

//Tabla2

module tabla2mux8(input wire[2:0] inS, output wire Y);
//modulo para mux 8 tabla 2

//registrar cables
    wire inA1, inB1, inC1, inD1, inE1, inF1, inG1, inH1;

//asignar valores
    assign inA1=1; assign inB1=0; assign inC1=0; assign inD1=0;
    assign inE1=0; assign inF1=1; assign inG1=1; assign inH1=0;

    mux8_1 U1(inA1, inB1, inC1, inD1, inE1, inF1, inG1, inH1, inS, Y);
endmodule

```

```

module tabla2mux4(input wire inC, input wire[1:0] inS, output wire Y);
    //aplicando la segunda tabla con un mux4a1 y compuertas
    wire NC, P;
    assign P=0;
    not N1(NC,inC);
    // la compuerta not nos sirve para trabajar unicamente con una entrada C y el mux 4
    mux4_1 U1(NC,P,inC,NC,inS, Y);
endmodule

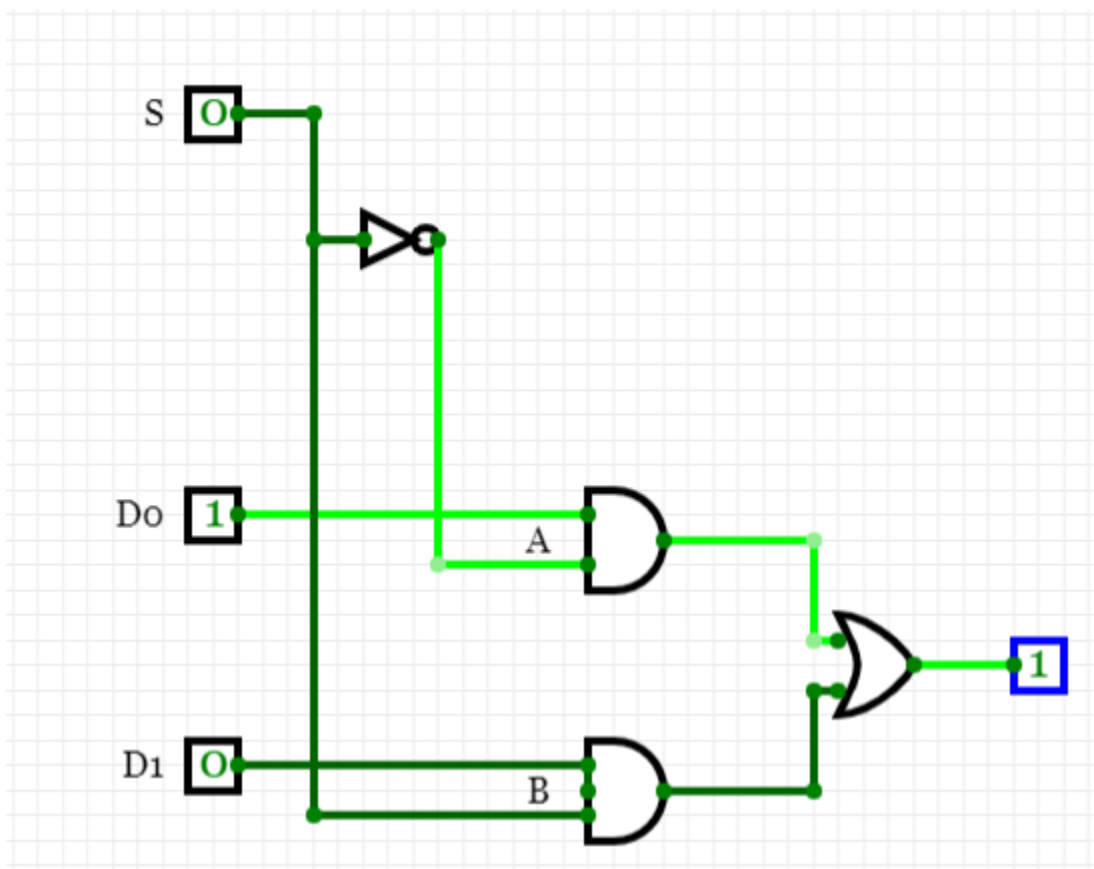
module tabla2mux2(input wire inS, inB, inC, output wire Y);
    //aplicacion de mux2 en tabla1
    wire Y1, Y2;

    assign Y1= inB ~| inC ;
    assign Y2= inB ^ inC;

    mux2_1 U1(Y1, Y2, inS, Y);

endmodule

```



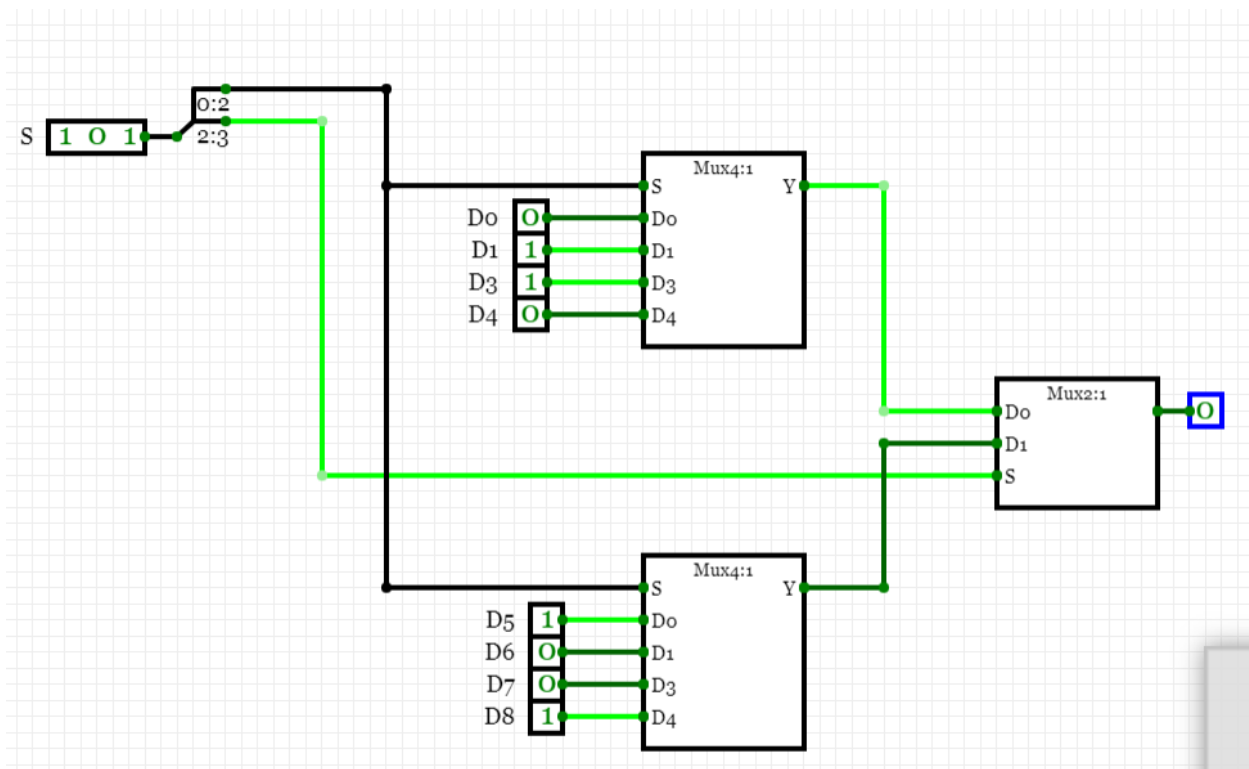
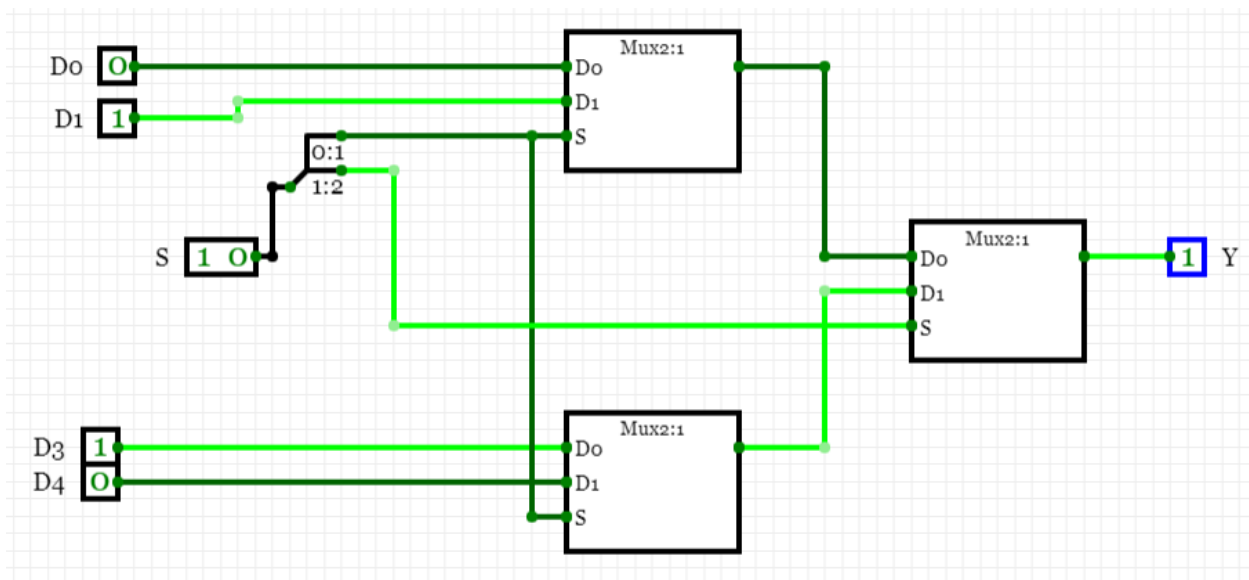




Tabla 01

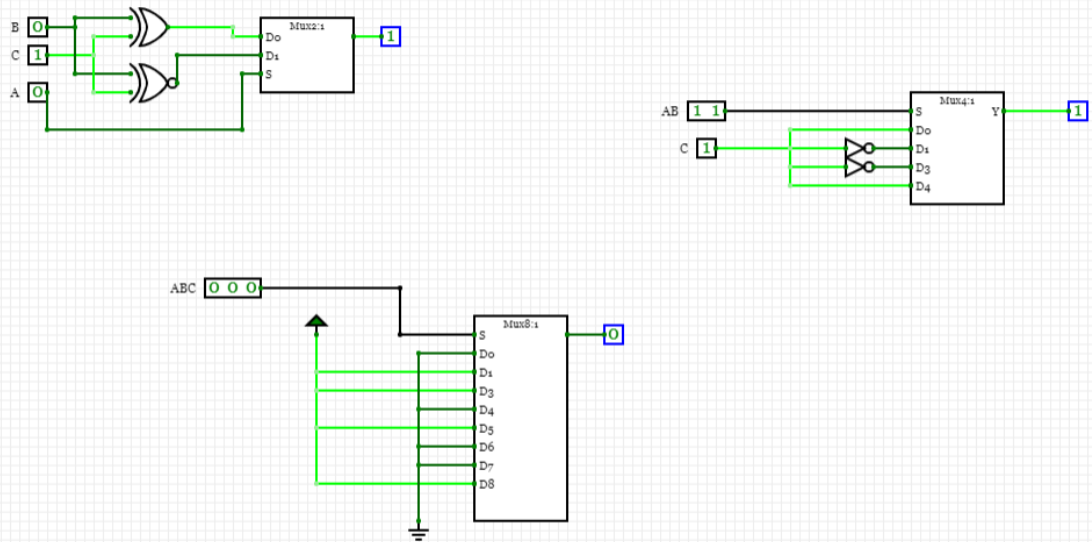
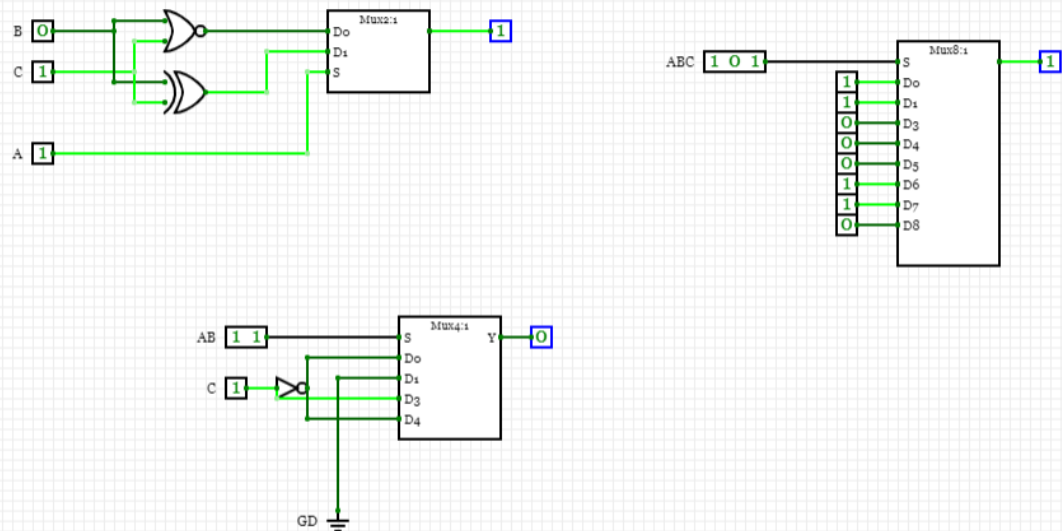
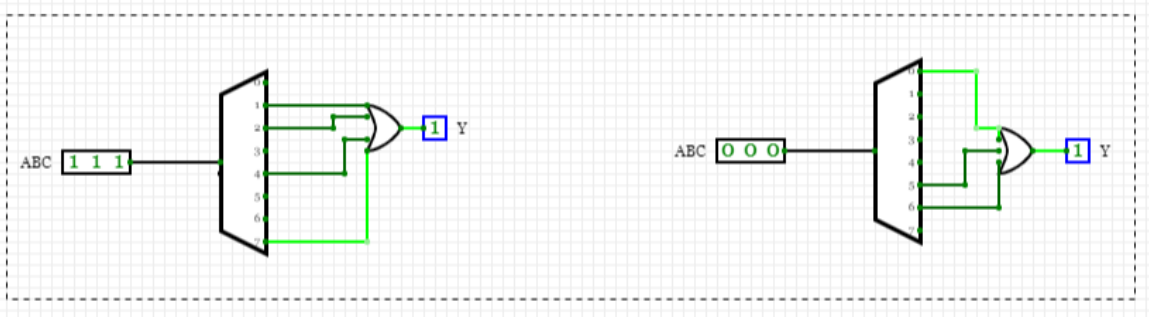


Tabla 02



Ejercicio 03



```
//Diego A. Méndez
//19673
//Testbench

module testbench();
// registrar entradas

    reg inC2;
    reg inB3, inC3;
    reg inC5;
    reg inB6, inC6;

    reg[2:0] inS1, inS4;
    reg[1:0] inS2, inS5;
    reg inS3, inS6;
    wire Y1, Y2, Y3, Y4, Y5, Y6;

    //Llamando a los modulos
        tabla1mux8 U1(inS1, Y1);
        tabla1mux4 U2(inC2, inS2, Y2);
        tabla1mux2 U3(inS3, inB3, inC3 ,Y3);
        tabla2mux8 U4(inS4, Y4);
        tabla2mux4 U5(inC5, inS5, Y5);
        tabla2mux2 U6(inS6, inB6, inC6, Y6);
```

```
initial begin //iniciamos el mux8_1
    $display(" ");
    $display(" Tabla 1 ");
    $display("Mux8 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b | %b",inS1 ,Y1);

        inS1=000;
    #1 inS1=001;
    #1 inS1=010;
    #1 inS1=011;
    #1 inS1=100;
    #1 inS1=101;
    #1 inS1=110;
    #1 inS1=111;
end
```

```
initial begin //iniciamos el mux8_1
#8
    $display(" ");
    $display(" Tabla 1 ");
    $display("Mux4 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b%b | %b",inS2, inC2 ,Y2);
        inS2=00; inC2=0;
    #1 inS2=00; inC2=1;
    #1 inS2=01; inC2=0;
    #1 inS2=01; inC2=1;
    #1 inS2=10; inC2=0;
    #1 inS2=10; inC2=1;
    #1 inS2=11; inC2=0;
    #1 inS2=11; inC2=1;
end
```

```
initial begin //iniciamos el mux8_1
#16
    $display(" ");
    $display(" Tabla 1 ");
    $display("Mux2 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b%b%b | %b",inS3, inB3, inC3 ,Y3);
        inS3=0; inB3=0; inC3=0;
    #1 inS3=0; inB3=0; inC3=1;
    #1 inS3=0; inB3=1; inC3=0;
    #1 inS3=0; inB3=1; inC3=1;
    #1 inS3=1; inB3=0; inC3=0;
    #1 inS3=1; inB3=0; inC3=1;
    #1 inS3=1; inB3=1; inC3=0;
    #1 inS3=1; inB3=1; inC3=1;
end
```

```
initial begin //iniciamos el mux8_1
#25
    $display(" ");
    $display(" Tabla 2 ");
    $display("Mux8 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b | %b",inS4 ,Y4);

        inS4=000;
    #1 inS4=001;
    #1 inS4=010;
    #1 inS4=011;
    #1 inS4=100;
    #1 inS4=101;
    #1 inS4=110;
    #1 inS4=111;
end
```

```

initial begin //iniciamos el mux8_1
#33
    $display(" ");
    $display(" Tabla 2 ");
    $display("Mux4 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b%b | %b",inS5, inC5 ,Y5);
        inS5=00; inC5=0;
    #1 inS5=00; inC5=1;
    #1 inS5=01; inC5=0;
    #1 inS5=01; inC5=1;
    #1 inS5=10; inC5=0;
    #1 inS5=10; inC5=1;
    #1 inS5=11; inC5=0;
    #1 inS5=11; inC5=1;
end

```



```
initial begin //iniciamos el mux8_1
#41
    $display(" ");
    $display(" Tabla 1 ");
    $display("Mux2 a 1");
    $display("ABC | Y");
    $display("-----");
    $monitor("%b%b%b | %b",inS6, inB6, inC6 ,Y6);
        inS6=0; inB6=0; inC6=0;
    #1 inS6=0; inB6=0; inC6=1;
    #1 inS6=0; inB6=1; inC6=0;
    #1 inS6=0; inB6=1; inC6=1;
    #1 inS6=1; inB6=0; inC6=0;
    #1 inS6=1; inB6=0; inC6=1;
    #1 inS6=1; inB6=1; inC6=0;
    #1 inS6=1; inB6=1; inC6=1;
end
```

```
initial
```

```
#49 $finish;
```

```
initial begin
```

```
    $dumpfile("Ejercicio4_tb.vcd");
```

```
    $dumpvars(0,testbench);
```

```
end
```

```
endmodule
```