

GABRIEL ALEXANDER FONG PENAGOS

19722

EJERCICIO 01

TABLA 01

MUX 8:1

Entrada	Valor	A	B	C	Y
000	0	0	0	0	0
001	1	0	0	1	1
010	1	0	1	0	1
011	0	0	1	1	0
100	1	1	0	0	1
101	0	1	0	1	0
110	0	1	1	0	0
111	1	1	1	1	1

MUX 4:1

Entrada	A	B	Y
00	0	0	C
01	0	1	$\sim C$
10	1	0	$\sim C$
11	1	1	C

MUX 2:1

Entrada	A	Y	
0	0	$B \wedge C$	XOR
1	1	$(B \sim \wedge C)$	XNOR

TABLA 02

MUX 8:1

Entrada	Valor	A	B	C	Y
000	1	0	0	0	1
001	X	0	0	1	X
010	0	0	1	0	0
011	0	0	1	1	0
100	X	1	0	0	X
101	1	1	0	1	1
110	1	1	1	0	1
111	0	1	1	1	0

MUX 4:1

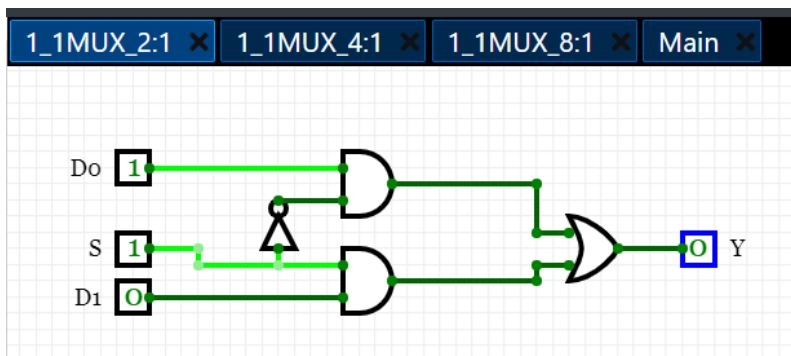
Entrada	A	B	Y
00	0	0	$\sim C$
01	0	1	0
10	1	0	C
11	1	1	$\sim C$

MUX 2:1

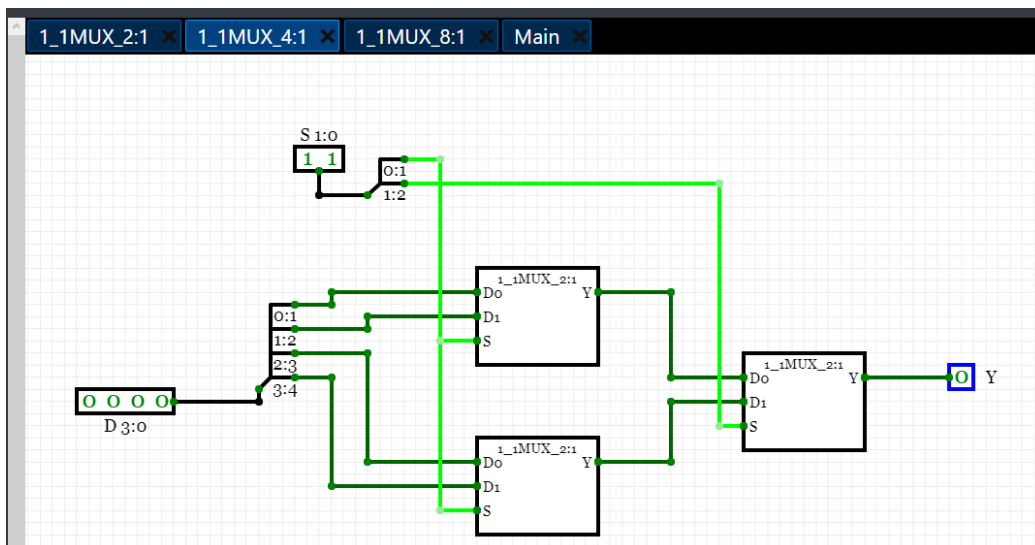
Entrada	A	Y	
0	0	$B \sim C$	NOR
1	1	$B \sim \& C$	NAND

EJERCICIO 02

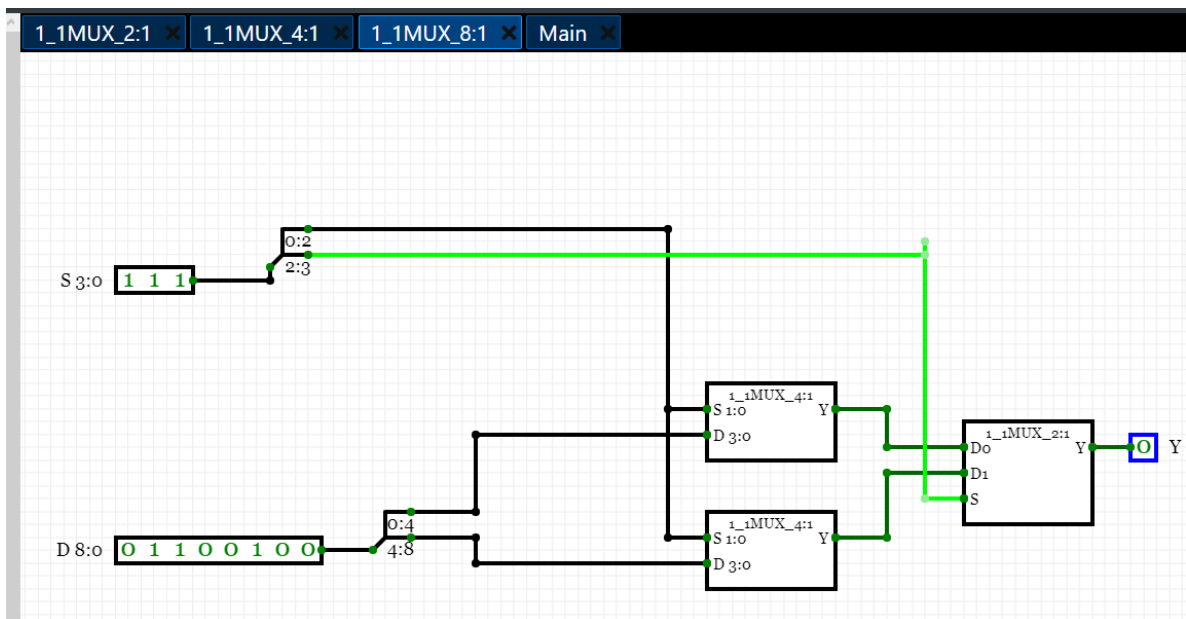
SUB-CIRCUITO MULTIPLEXOR 2:1



SUB-CIRCUITO MULTIPLEXOR 4:1



SUB-CIRCUITO MULTIPLEXOR 8:1



<p>TABLA 01 MUX 2:1</p>	<p>TABLA 01 MUX 4:1</p>
<p>TABLA 01 MUX 8:1</p>	<p>TABLA 02 MUX 8:1</p>
<p>TABLA 02 MUX 4:1</p>	<p>TABLA 02 MUX 8:1</p>

EJERCICIO 03

TABLA 01 DECODER

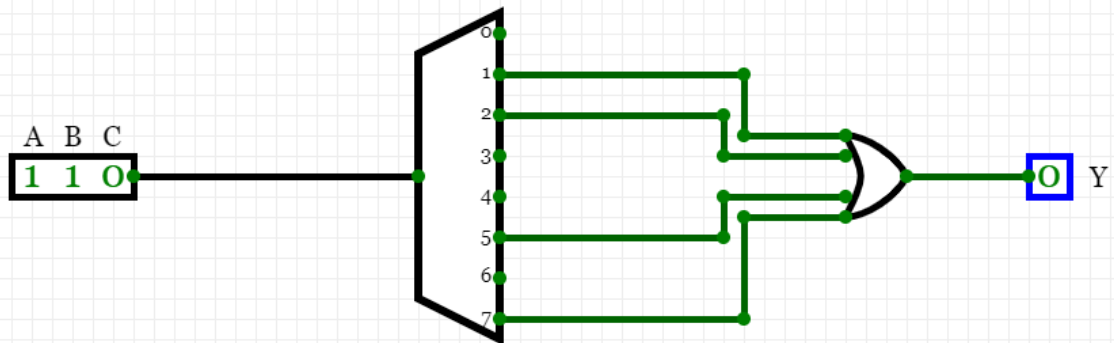
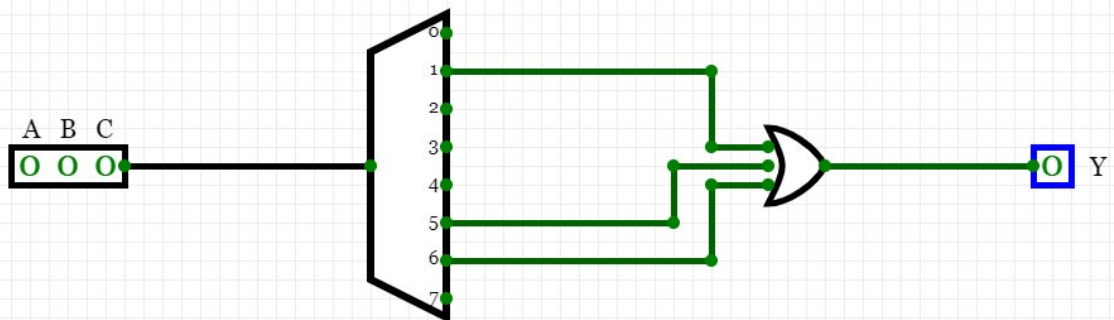
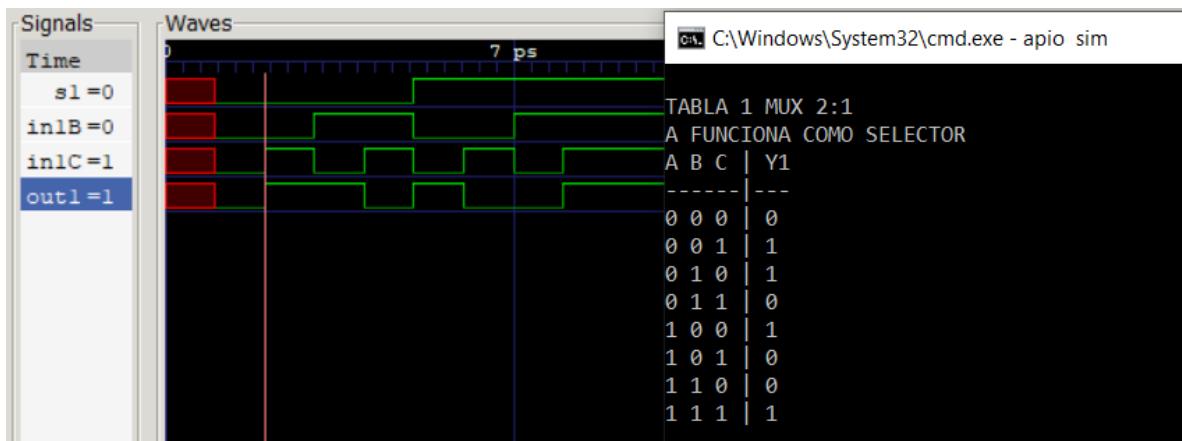


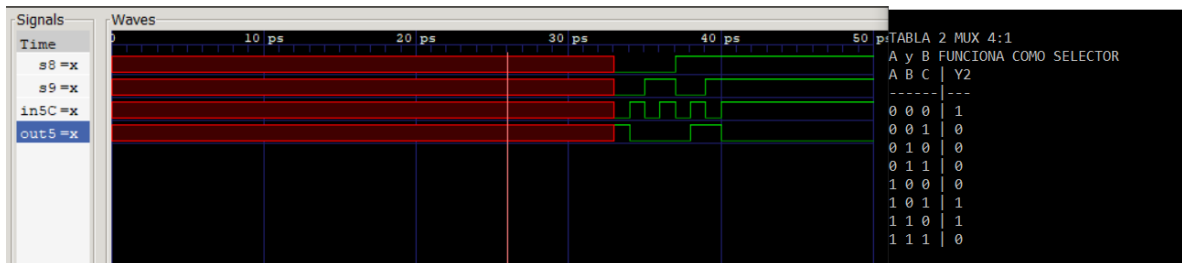
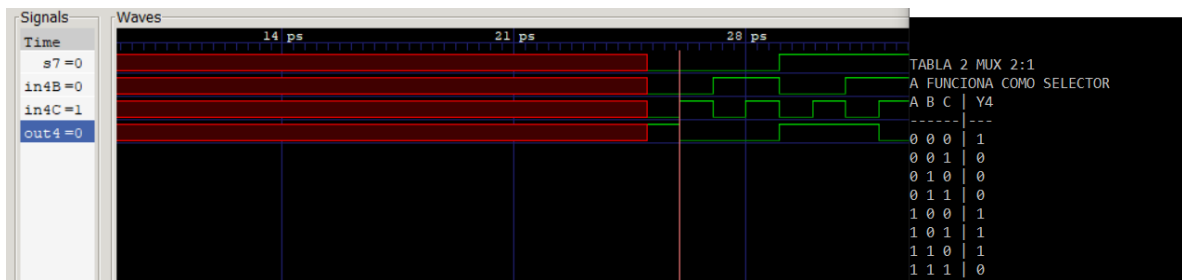
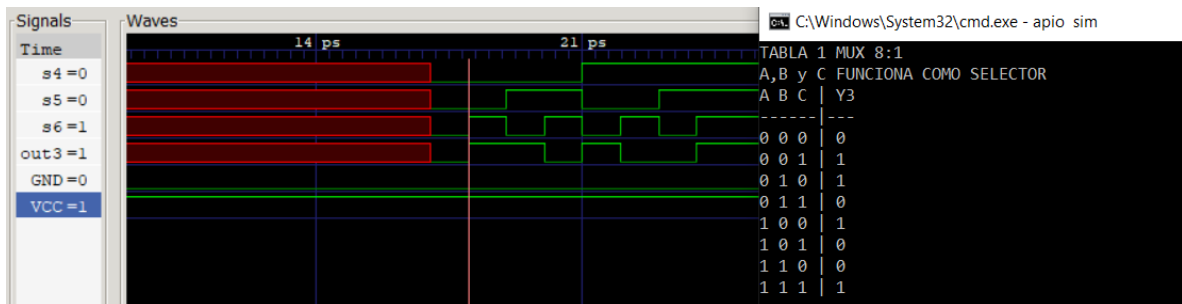
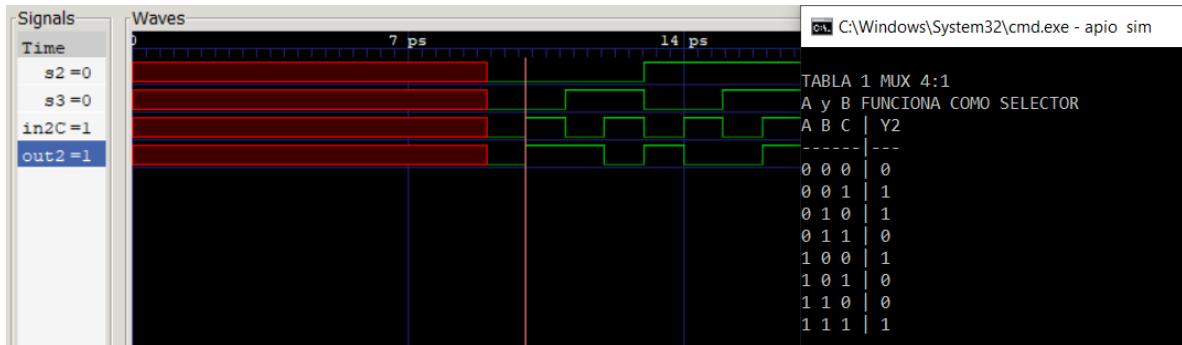
TABLA 02 DECODER

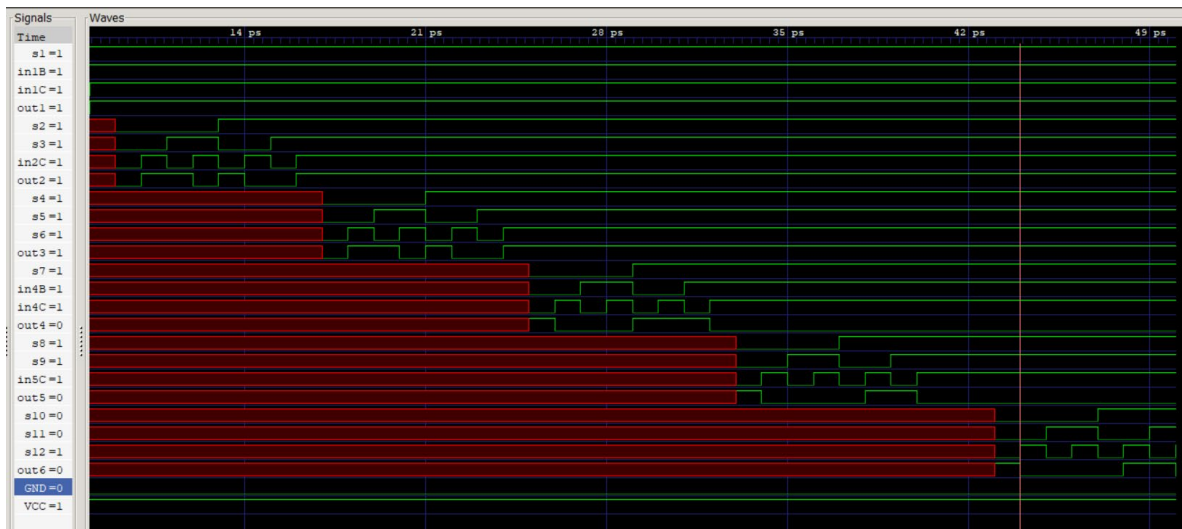
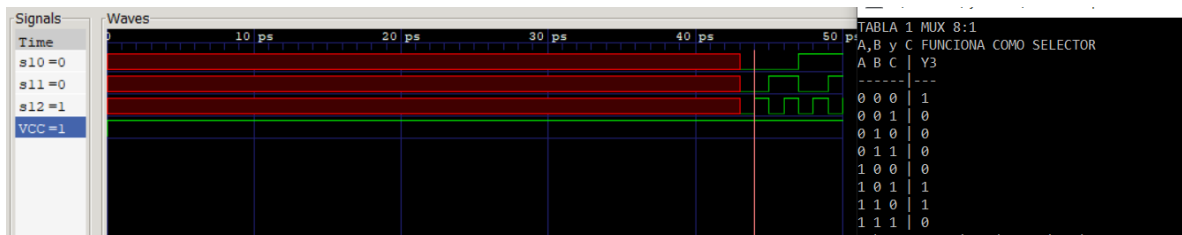
TABLA 02



EJERCICIO 04







PROGRAMACION MUX.v

```

1 //GABRIEL ALEXANDER FONG PENAGOS
2 //CARNE 19722
3 //ELECTRONICA DIGITAL
4 //LABORATORIO 05
5 // MUXES
6 module MUX2_1(input wire A, B, C, output wire Y);
7
8     assign Y = A?C:B;
9
10 endmodule
11
12 module MUX4_1(input wire A, B, C, D, input wire[1:0]S, output wire Y);
13
14     wire OUT1, OUT2;
15
16     MUX2_1 mux1(S[0], A, B, OUT1);
17     MUX2_1 mux2(S[0], C, D, OUT2);
18     MUX2_1 mux3(S[1], OUT1, OUT2, Y);
19
20 endmodule
21
22 module MUX8_1(input wire A, B, C, D, E, F, G, H, input wire [2:0] S, output wire Y);
23
24     wire OUT1, OUT2;
25
26     MUX4_1 mux1(A, B, C, D, S[1:0], OUT1);
27     MUX4_1 mux2(E, F, G, H, S[1:0], OUT2);
28     MUX2_1 mux3(S[2], OUT1, OUT2, Y);
29
30 endmodule

```

```

30 endmodule //EN EL MUX 2:1 SE COLOCA DE PRIMERO EL SELECTOR Y LUEGO LAS 2 ENTRADAS
31
32 module t1_mux2_1(input wire A, B, C, output wire Y); //CREAMOS EL MODULO TABLA 1 CON MUX 2:1 A ES SELECTOR
33 //A SERA NUESTRO SELECTOR. B Y C SERAN LOS QUE DEFINAN EL VALOR DE SALIDA
34 wire RES1, RES2;
35 assign RES1 = (B^C); //REALIZAMOS XOR PARA OBTENER LOS VALORES CUANDO A = 0
36 assign RES2 = ~(B^C); //REALIZAMOS XNOR PARA OBTENER LOS VALORES CUANDO A = 1
37
38 MUX2_1 U1(A, RES1, RES2, Y); //UTILIZAMOS EL MUX 2:1 PARA TERMINAR DE REALIZAR LA TABLA 1
39
40 endmodule
41
42 module t1_mux4_1(input wire A, B, C, output wire Y); //CREAMOS EL MODULO TABLA 1 CON MUX 4:1 A Y B SON SELECTORES
43 //A SERA NUESTRO SELECTOR DE LA SEGUNDA CIFRA Y B DE LA PRIMER CIFRA DE Y C SERA EL
44 wire [1:0]S;
45 assign S[0] = B; //LE ASIGNAMOS LA POSICION A NUESTROS SELECTORES
46 assign S[1] = A;
47 MUX4_1 U2(C, ~C, ~C, C, S[1:0], Y); //UTILIZAMOS EL MUX 4:1 PARA TERMINAR DE REALIZAR LA TABLA 1
48 endmodule
49
50 module t1_mux8_1(input wire A, B, C, GND, VCC, output wire Y); //CREAMOS EL MODULO TABLA 1 CON MUX 8:1 A B C SON SELECTORES
51 wire [2:0]S; //VCC Y GND DEFINIRAN LOS VALORES DE SALIDA, SIENDO GND 0 Y VCC 1
52 assign S[0] = C; //ASIGNAMOS LAS POSICIONES DE LOS SELECTORES
53 assign S[1] = B;
54 assign S[2] = A;
55 MUX8_1 U3(GND, VCC, VCC, GND, VCC, GND, GND, VCC, S[2:0], Y); //UTILIZAMOS EL MUX 8:1 PARA TERMINAR DE REALIZAR LA TABLA 1
56 endmodule
57

```

```

58 module t2_mux2_1(input wire A, B, C, output wire Y); //CREAMOS EL MODULO TABLA 2 CON MUX 2:1 A ES SELECTOR
59 //A SERA NUESTRO SELECTOR. B Y C SERAN LOS QUE DEFINAN EL VALOR DE SALIDA
60 wire RES1, RES2;
61 assign RES1 = ~(B | C); //REALIZAMOS NOR PARA OBTENER LOS VALORES CUANDO A = 0
62 assign RES2 = ~(B & C); //REALIZAMOS NAND PARA OBTENER LOS VALORES CUANDO A = 1
63 MUX2_1 U3(A, RES1, RES2, Y); //UTILIZAMOS EL MUX 2:1 PARA TERMINAR DE REALIZAR LA TABLA 2
64 endmodule
65
66 module t2_mux4_1(input wire A, B, C, GND, output wire Y); //CREAMOS EL MODULO TABLA 2 CON MUX 4:1 A Y B SON SELECTORES
67 //A SERA NUESTRO SELECTOR DE LA SEGUNDA CIFRA Y B DE LA PRIMER CIFRA DE Y C Y GND 0
68 wire [1:0]S; //ASIGNAMOS LA POSICION A NUESTROS SELECTORES
69 assign S[0] = B;
70 assign S[1] = A;
71 MUX4_1 U5(~C, GND, C, ~C, S[1:0], Y); //UTILIZAMOS EL MUX 4:1 PARA TERMINAR DE REALIZAR LA TABLA 2
72 endmodule
73
74 module t2_mux8_1(input wire A, B, C, GND, VCC, output wire Y); //CREAMOS EL MODULO TABLA 1 CON MUX 8:1 A B C SON SELECTORES
75 wire [2:0]S; //VCC Y GND DEFINIRAN LOS VALORES DE SALIDA, SIENDO GND 0 Y VCC 1
76 assign S[0] = C; //ASIGNAMOS LAS POSICIONES DE LOS SELECTORES
77 assign S[1] = B;
78 assign S[2] = A;
79 MUX8_1 U6(VCC, GND, GND, GND, VCC, VCC, GND, VCC, S[2:0], Y); //UTILIZAMOS EL MUX 8:1 PARA TERMINAR DE REALIZAR LA TABLA 2
80 endmodule
81

```

PROGRAMACION MUX_tb.v

```

Welcome Guide X MUX.v MUX_tbv
1 //GABRIEL ALEXANDER FONG PENAGOS
2 //CARNE 19722
3 //ELECTRONICA DIGITAL
4 //LABORATORIO 05
5 //MUXES
6
7 module testbench ();
8
9 wire GND; //GND COMO CABLE Y LUEGO LE ASIGNAMOS DE VALOR 0
10 wire VCC; //VCC COMO GABLE Y LUEGO LE ASIGNAMOS DE VALOR 1
11 assign GND = 0;
12 assign VCC = 1;
13 reg s1, in1B, in1C; //CREAMOS LOS REGISTROS PARA t1_mux2_1
14 reg s2, s3, in2C; //CREAMOS LOS REGISTROS PARA t1_mux4_1
15 reg s4, s5, s6; //CREAMOS LOS REGISTROS PARA t1_mux8_1
16 reg s7, in4B, in4C; //CREAMOS LOS REGISTROS PARA t2_mux2_1
17 reg s8, s9, in5C; //CREAMOS LOS REGISTROS PARA t2_mux4_1
18 reg s10, s11, s12; //CREAMOS LOS REGISTROS PARA t2_mux8_1
19 wire out1, out2, out3, out4, out5, out6; //CREAMOS LAS SALIDAS PARA LOS MODULOS
20
21 t1_mux2_1 U1(s1, in1B, in1C, out1); //s1 ES NUESTRO SELECTOR (A), in1B, in1C son los valores de input de B y C
22 t1_mux4_1 U2(s2, s3, in2C, out2); //s2, s3 SON NUESTROS SELECTORES (A), (B). in2c el valor de input de C
23 t1_mux8_1 U3(s4, s5, s6, GND, VCC, out3); //s4, s5, s6 SON NUESTROS SELECTORES (A), (B), (C)
24 t2_mux2_1 U4(s7, in4B, in4C, out4); //s7 ES NUESTRO SELECTOR (A), in4B, in4C son los valores de input de B y C
25 t2_mux4_1 U5(s8, s9, in5C, GND, out5); //s8, s9 SON NUESTROS SELECTORES (A), (B). in5c el valor de input de C
26 t2_mux8_1 U6(s10, s11, s12, GND, VCC, out6); //s10, s11, s12 SON NUESTROS SELECTORES (A), (B), (C)
27
28

```

```

29     initial begin                                     //INICIAMOS LA TABLA 1 CON EL MUX 2:1
30     #1
31     $display("\n");
32     $display(" TABLA 1 MUX 2:1 ");
33     $display("A FUNCIONA COMO SELECTOR");
34     $display("A B C | Y1");
35     $display("-----|---");
36     $monitor("%b %b %b | %b ", s1, in1B, in1C, out1);
37     s1 = 0; in1B = 0; in1C = 0;
38     #1 s1 = 0; in1B = 0; in1C = 1;
39     #1 s1 = 0; in1B = 1; in1C = 0;
40     #1 s1 = 0; in1B = 1; in1C = 1;
41     #1 s1 = 1; in1B = 0; in1C = 0;
42     #1 s1 = 1; in1B = 0; in1C = 1;
43     #1 s1 = 1; in1B = 1; in1C = 0;
44     #1 s1 = 1; in1B = 1; in1C = 1;
45     end
46
47     initial begin                                     //INICIAMOS LA TABLA 1 CON EL MUX 4:1
48     #9
49     $display("\n");
50     $display(" TABLA 1 MUX 4:1 ");
51     $display("A y B FUNCIONA COMO SELECTOR");
52     $display("A B C | Y2");
53     $display("-----|---");
54     $monitor("%b %b %b | %b ", s2, s3, in2C, out2); //selector s2 = A, s3 = B

```

```

55     s2 = 0; s3 = 0; in2C = 0;
56     #1 s2 = 0; s3 = 0; in2C = 1;
57     #1 s2 = 0; s3 = 1; in2C = 0;
58     #1 s2 = 0; s3 = 1; in2C = 1;
59     #1 s2 = 1; s3 = 0; in2C = 0;
60     #1 s2 = 1; s3 = 0; in2C = 1;
61     #1 s2 = 1; s3 = 1; in2C = 0;
62     #1 s2 = 1; s3 = 1; in2C = 1;
63     end
64
65     initial begin                                     //INICIAMOS LA TABLA 1 CON EL MUX 8:1
66     #17
67     $display("\n");
68     $display(" TABLA 1 MUX 8:1 ");
69     $display("A,B y C FUNCIONA COMO SELECTOR");
70     $display("A B C | Y3");
71     $display("-----|---");
72     $monitor("%b %b %b | %b ", s4, s5, s6, out3); //selector s4 = A, s5 = B, s6 = C
73     s4 = 0; s5 = 0; s6 = 0;
74     #1 s4 = 0; s5 = 0; s6 = 1;
75     #1 s4 = 0; s5 = 1; s6 = 0;
76     #1 s4 = 0; s5 = 1; s6 = 1;
77     #1 s4 = 1; s5 = 0; s6 = 0;
78     #1 s4 = 1; s5 = 0; s6 = 1;
79     #1 s4 = 1; s5 = 1; s6 = 0;
80     #1 s4 = 1; s5 = 1; s6 = 1;
81     end

```

```

83     initial begin                                     //INICIAMOS LA TABLA 2 CON EL MUX 2:1
84     #25
85     $display("\n");
86     $display(" TABLA 2 MUX 2:1 ");
87     $display("A FUNCIONA COMO SELECTOR");
88     $display("A B C | Y4");
89     $display("-----|---");
90     $monitor("%b %b %b | %b ", s7, in4B, in4C, out4);
91     s7 = 0; in4B = 0; in4C = 0;
92     #1 s7 = 0; in4B = 0; in4C = 1;
93     #1 s7 = 0; in4B = 1; in4C = 0;
94     #1 s7 = 0; in4B = 1; in4C = 1;
95     #1 s7 = 1; in4B = 0; in4C = 0;
96     #1 s7 = 1; in4B = 0; in4C = 1;
97     #1 s7 = 1; in4B = 1; in4C = 0;
98     #1 s7 = 1; in4B = 1; in4C = 1;
99     end
100
101     initial begin                                     //INICIAMOS LA TABLA 2 CON EL MUX 4:1
102     #33
103     $display("\n");
104     $display(" TABLA 2 MUX 4:1 ");
105     $display("A y B FUNCIONA COMO SELECTOR");
106     $display("A B C | Y2");
107     $display("-----|---");
108     $monitor("%b %b %b | %b ", s8, s9, in5C, out5); //selector s7 = A, s8 = B

```



```

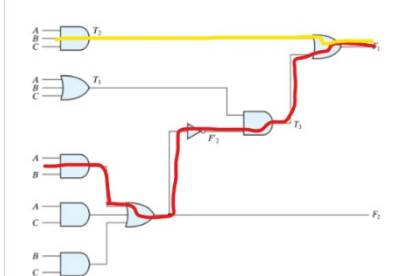
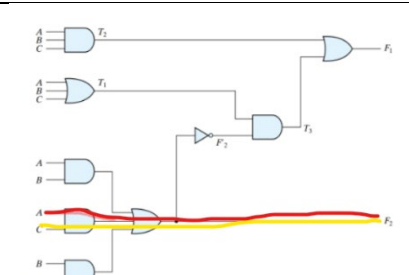
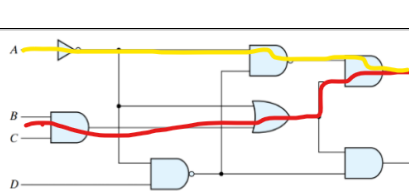
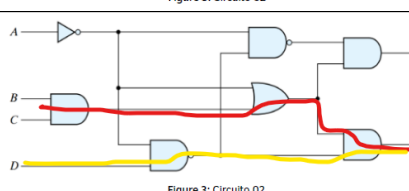
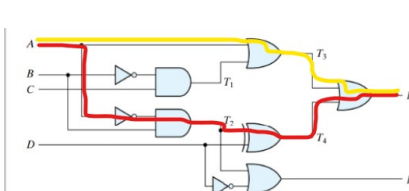
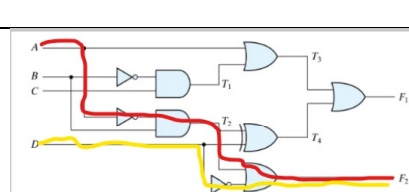
108 ~ $monitor("%b %b %b | %b ", s8, s9, in5C, out5); //selector s7 = A, s8 = B
109     s8 = 0; s9 = 0; in5C = 0;
110     #1 s8 = 0; s9 = 0; in5C = 1;
111     #1 s8 = 0; s9 = 1; in5C = 0;
112     #1 s8 = 0; s9 = 1; in5C = 1;
113     #1 s8 = 1; s9 = 0; in5C = 0;
114     #1 s8 = 1; s9 = 0; in5C = 1;
115     #1 s8 = 1; s9 = 1; in5C = 0;
116 ~ #1 s8 = 1; s9 = 1; in5C = 1;
117     end
118
119     initial begin                                //INICIAMOS LA TABLA 2 CON EL MUX 8:1
120         #43
121         $display("\n");
122         $display(" TABLA 1 MUX 8:1 ");
123         $display("A,B y C FUNCIONA COMO SELECTOR");
124         $display("A B C | Y3");
125         $display("-----|---");
126 ~ $monitor("%b %b %b | %b ", s10, s11, s12, out6); //selector s4 = A, s5 = B, s6 = C
127     s10 = 0; s11 = 0; s12 = 0;
128     #1 s10 = 0; s11 = 0; s12 = 1;
129     #1 s10 = 0; s11 = 1; s12 = 0;
130     #1 s10 = 0; s11 = 1; s12 = 1;
131     #1 s10 = 1; s11 = 0; s12 = 0;
132     #1 s10 = 1; s11 = 0; s12 = 1;
133     #1 s10 = 1; s11 = 1; s12 = 0;
134 ~ #1 s10 = 1; s11 = 1; s12 = 1;
135     end
136
137 ~ initial begin                                //INICIAMOS PARA COLOCAR LOS DATOS EN GTK WAVE
138     $dumpfile("MUX_tb.vcd");
139     $dumpvars(0, testbench);
140     end
141
142 endmodule

```

EJERCICIO 05

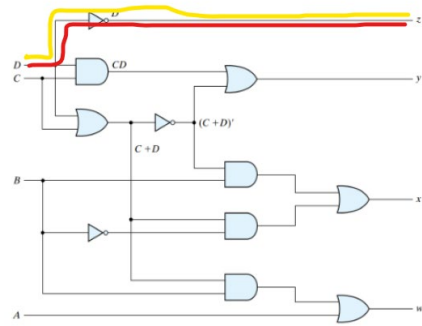
- Propagation Delay: Es el tiempo máximo o retraso entre el cambio de la entrada y el cambio en la salida. Tiempo máximo en que se estabiliza la salida luego de un cambio en la entrada
- Contamination Delay: Es el tiempo mínimo o retraso entre el cambio de la entrada y el cambio en la salida.
- Analogía: Contamination Delay es como cuando tu ex toxica cambia pero el Propagation delay es todo el tiempo en el que se tarda en estabilizarse.
- Ruta Crítica: Es la ruta en donde la suma del Tpd (Propagation Delay) es más alta.
- Ruta Corta: Es la ruta en donde la suma del Tcd (Contamination Delay) es más baja.

EJERCICIO 06

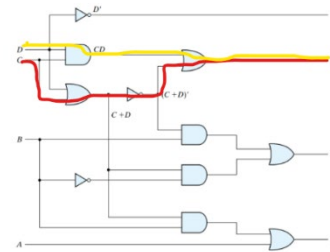
CIRCUITO 1			
F1			
Para F1	Tpd (ps)	Tcd (ps)	
T1	125	100	
T2	80	60	
T3	170	135	
F2			
Para F2	Tpd (ps)	Tcd (ps)	
T4	85	70	
CIRCUITO 2			
Para F	Tpd (ps)	Tcd (ps)	
Ruta 1	65	50	
Ruta 2	85	65	
Ruta 3	100	80	
Ruta 4	70	55	
Para G	Tpd (ps)	Tcd (ps)	
Ruta 5	85	65	
Ruta 6	100	80	
Ruta 7	50	40	
CIRCUITO 3			
Para F1	Tpd (ps)	Tcd (ps)	
Ruta 1	80	60	
Ruta 2	110	85	
Ruta 3	125	95	
Ruta 4	145	105	
Ruta 5	95	75	
Ruta 6	100	70	
Para F2	Tpd (ps)	Tcd (ps)	
Ruta 7	85	65	
Ruta 8	70	55	
Ruta 9	55	40	

CIRCUITO 4

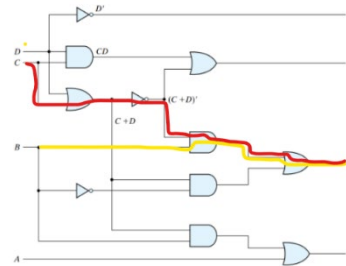
Para Z	Tpd (ps)	Tcd (ps)
Ruta 1	15	10



Para Y	Tpd (ps)	Tcd (ps)
Ruta 2	70	55
Ruta 3	95	70



Para X	Tpd (ps)	Tcd (ps)
Ruta 4	125	95
Ruta 5	70	55
Ruta 6	85	65



Para W	Tpd (ps)	Tcd (ps)
Ruta 7	110	85
Ruta 8	70	55
Ruta 9	40	30

