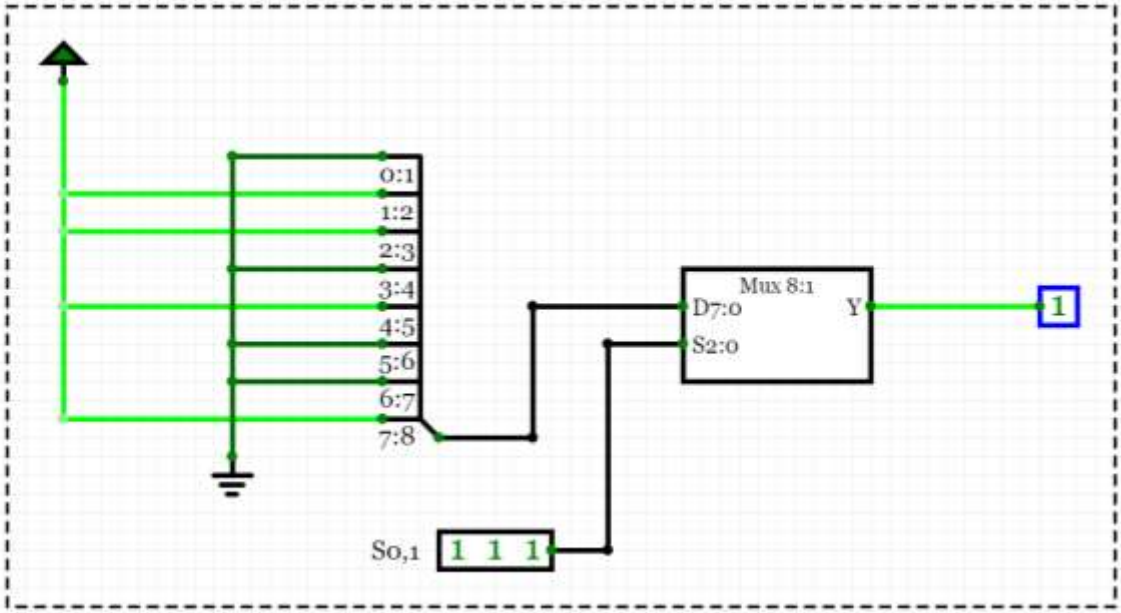
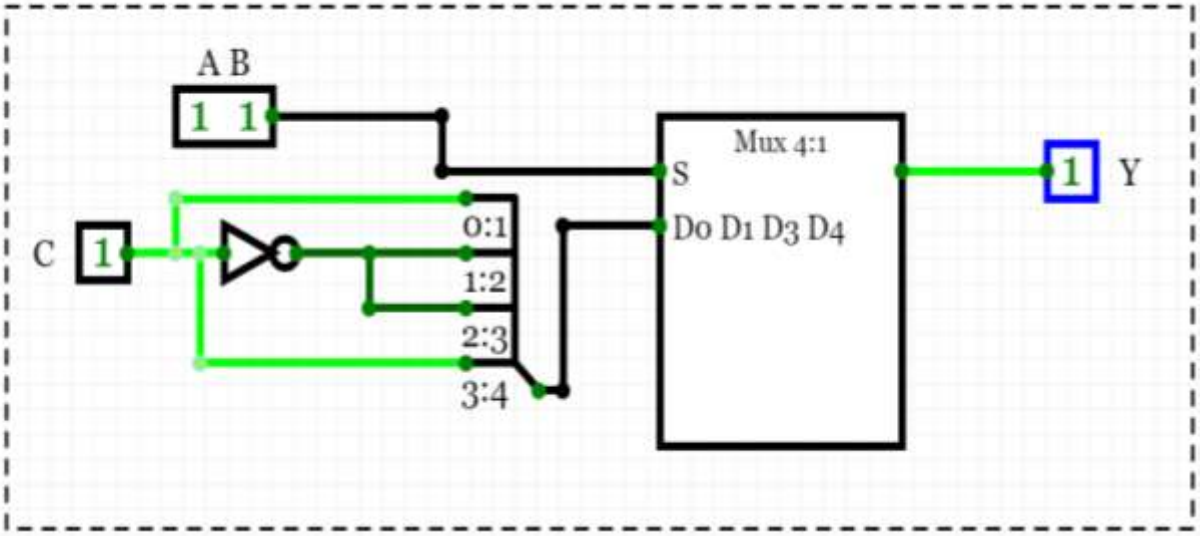
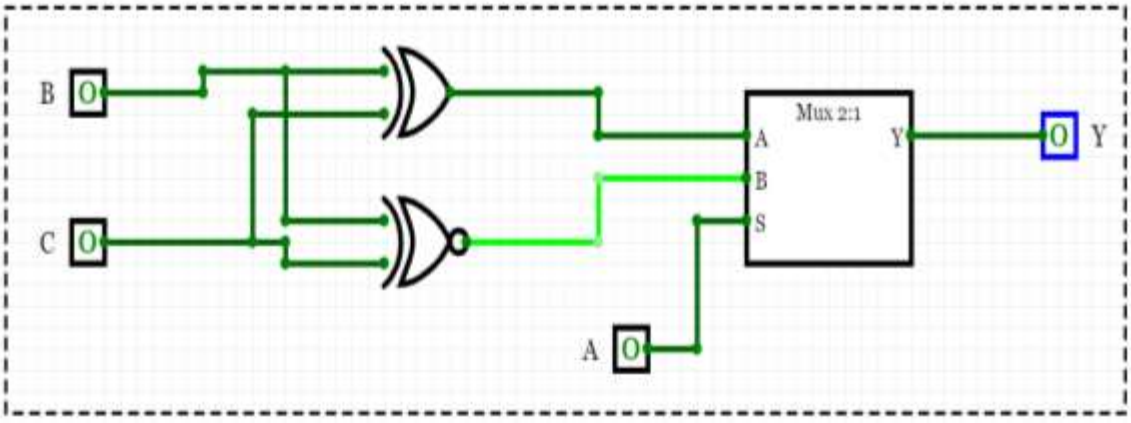
Mux 8:1 Tabla1



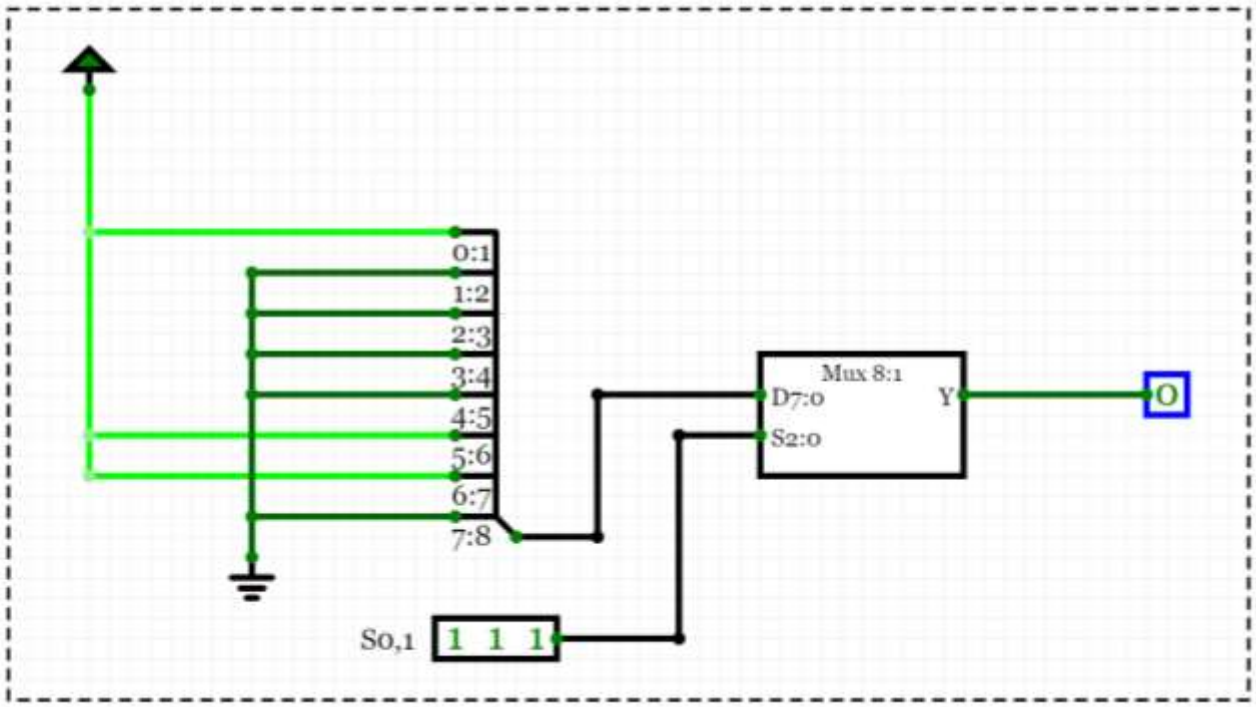
Mux 4:1 Tabla 1



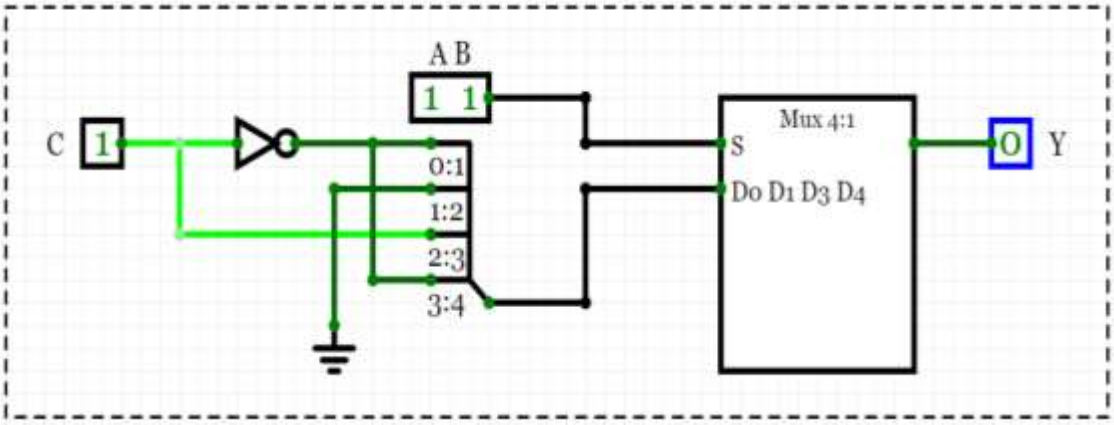
Mux 2:1 Tabla 1



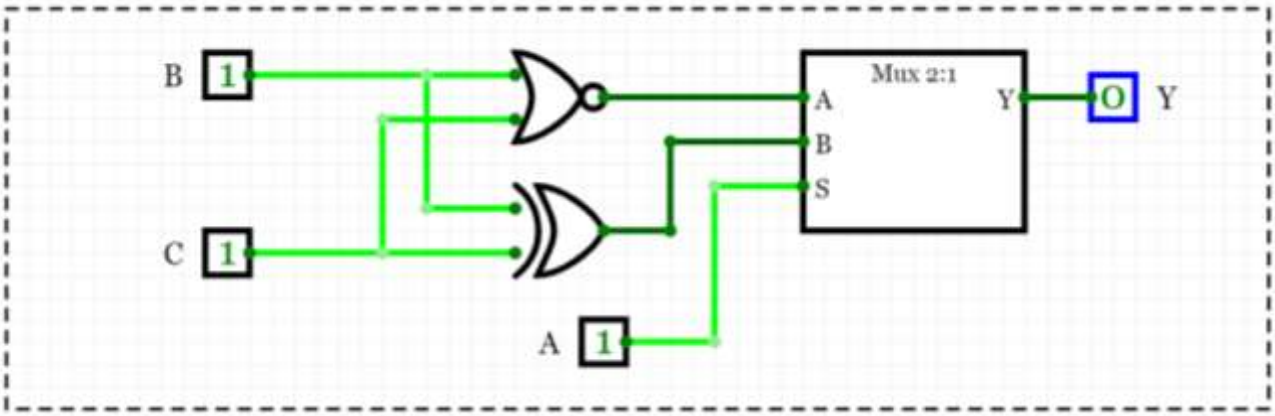
Mux 8:1 Tabla 2



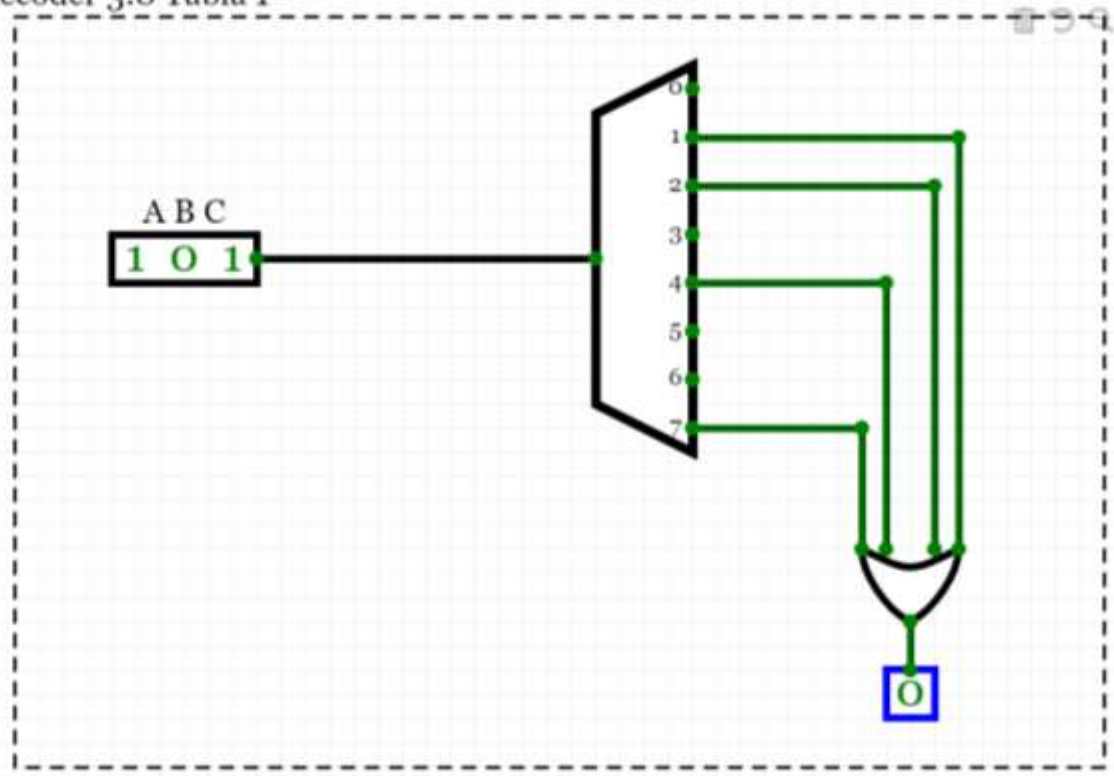
Mux 4:1 Tabla 2



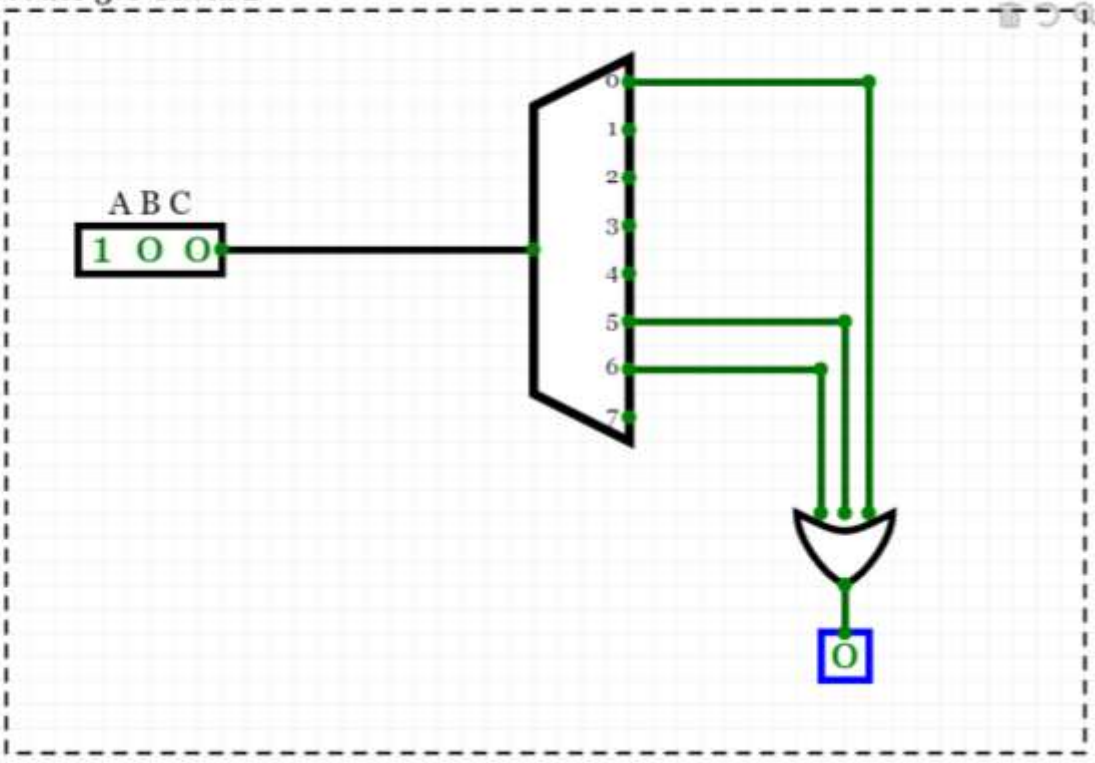
Mux 2:1 Tabla 2



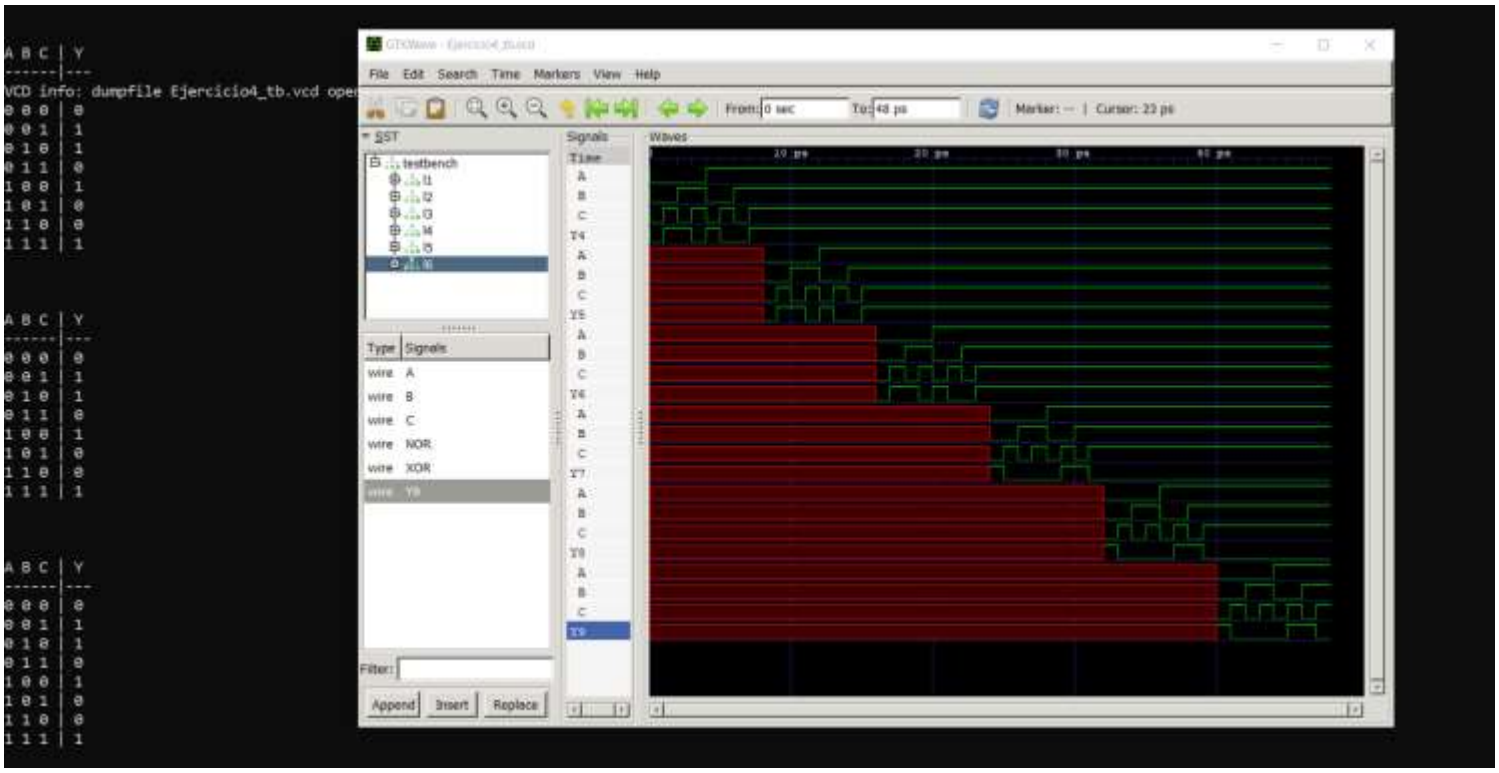
EJERCICIO 3 Decoder 3:8 Tabla 1



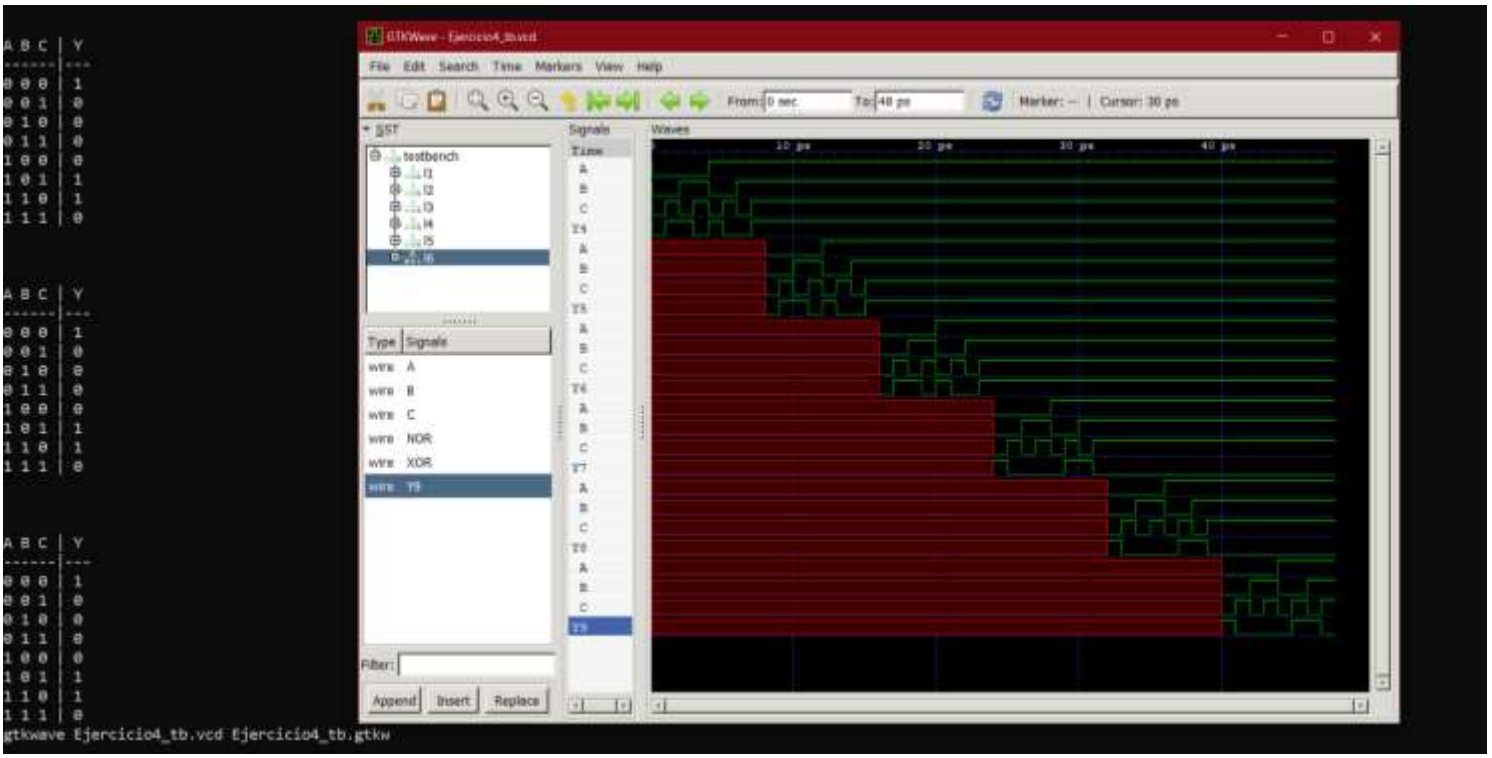
EJERCICIO 3 Decoder 3:8 Tabla 2



GTKWave despliegue de las primeras tablas de tabla 1



GTKWave despliegue de las siguientes tres tablas de tabla 2

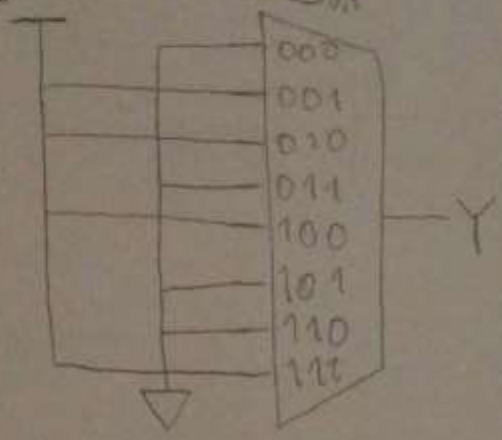


Lab #05

Ejercicio 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

1)

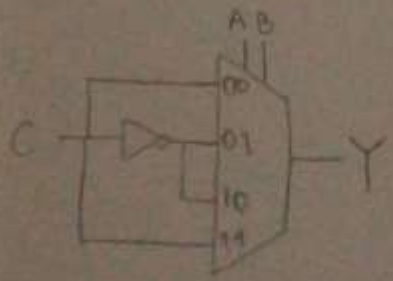


2)

A	B	Y
0	0	C
0	1	$\bar{C}$
1	0	$\bar{C}$
1	1	C

3)

A	Y
0	$B \oplus C$
1	$\overline{B \oplus C}$



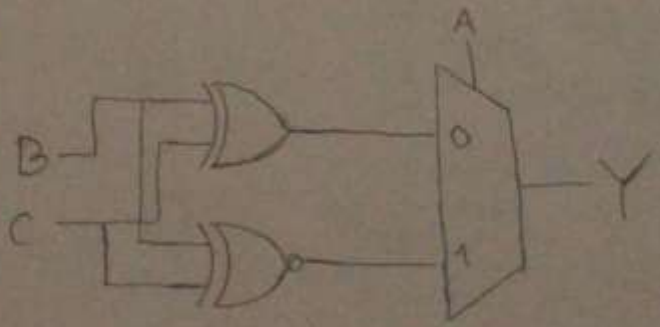
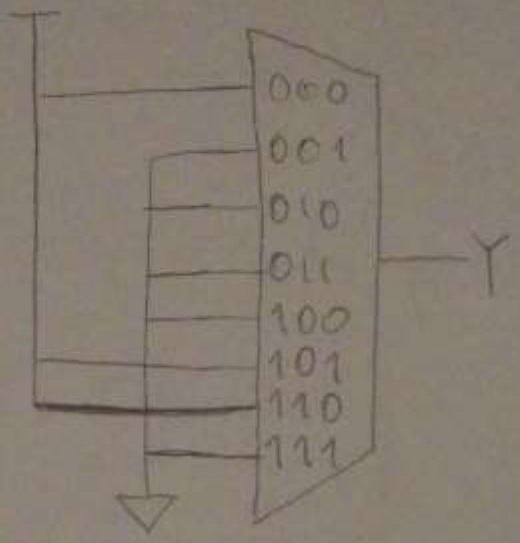


Tabla 2

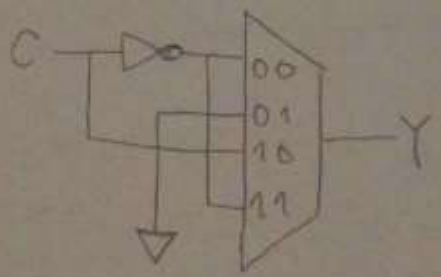
A	B	C	Y
0	0	0	1
0	0	1	X
0	1	0	0
0	1	1	0
1	0	0	X
1	0	1	1
1	1	0	1
1	1	1	0

1)



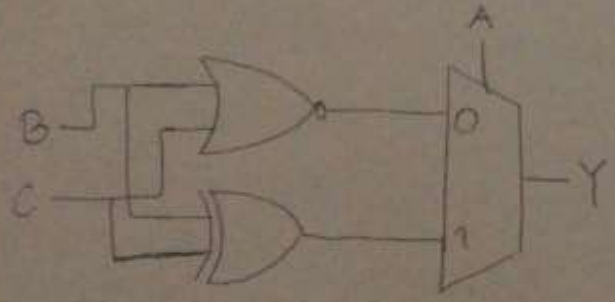
2)

A	B	Y
0	0	$\bar{C}$
0	1	0
1	0	C
1	1	$\bar{C}$



3)

A	Y
0	$\overline{B+C}$
1	$B \oplus C$



Ejercicio 5

Propagation delay

Es lo que tarda en estabilizarse la señal de salida

Contamination delay

Es el tiempo que tarda la señal de salida en desestabilizarse.

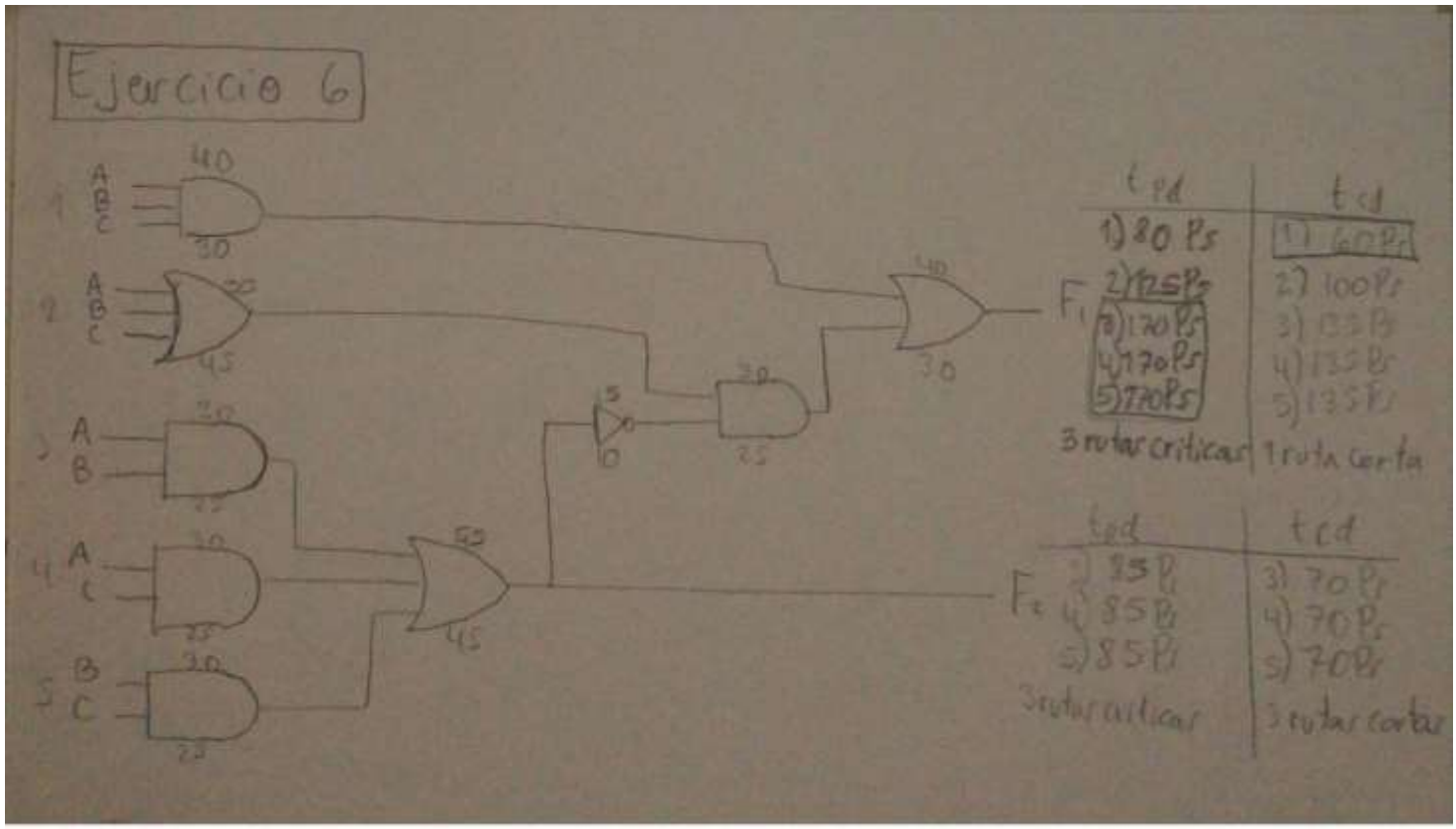
Ruta crítica

Tiene que ver con la Propagtion delay (t_{pd}), y es la ruta donde la sumatoria de t_{pd} es mayor.

Ruta Corta

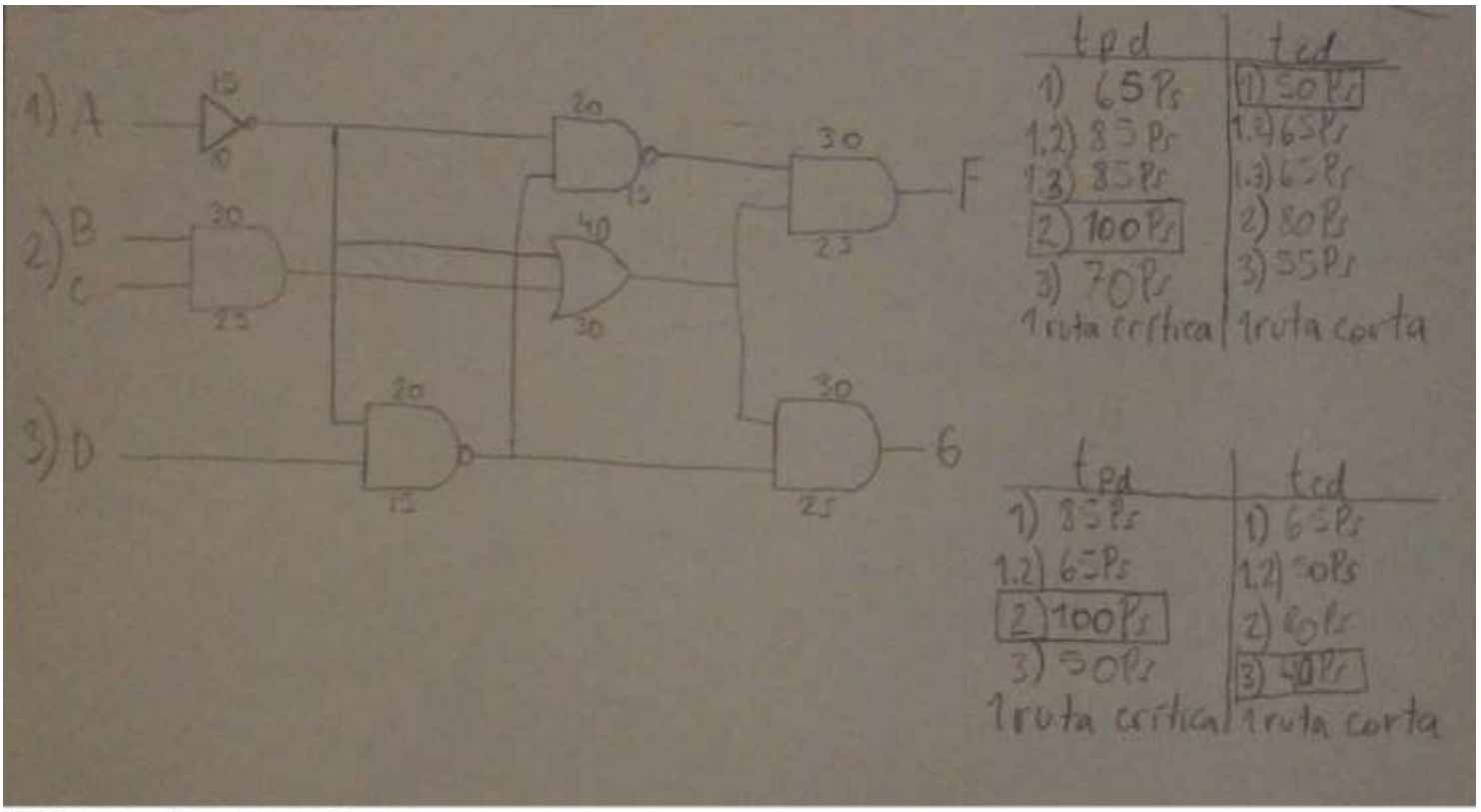
Tiene que ver con Cantamination delay (t_{cd}), y es la ruta donde la sumatoria de t_{cd} es la menor.

Ejercicio 6



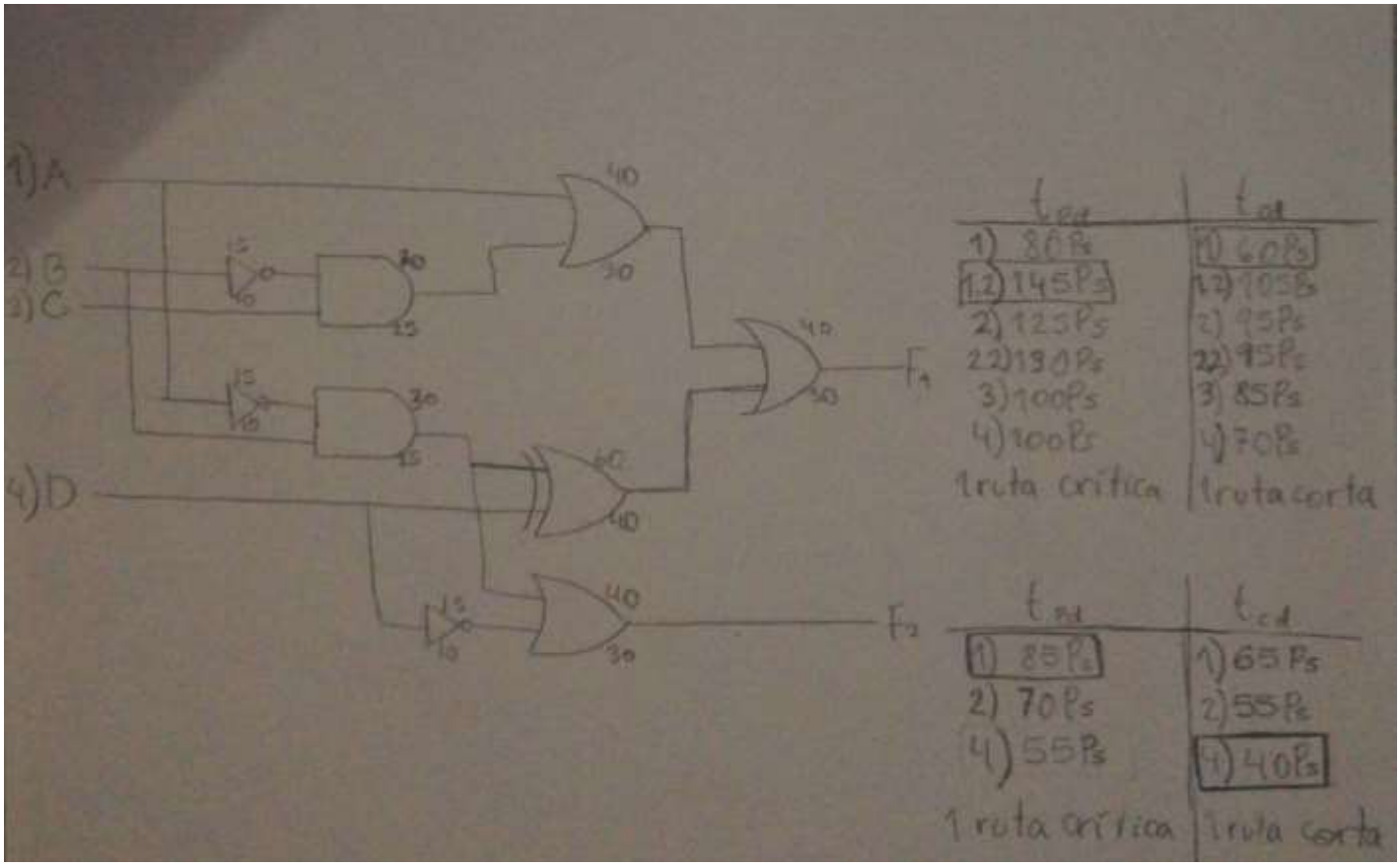
t_{pd}	t_{cd}
1) 80 Ps	1) 60 Ps
2) 125 Ps	2) 100 Ps
3) 170 Ps	3) 135 Ps
4) 170 Ps	4) 135 Ps
5) 170 Ps	5) 135 Ps
3 rutas criticas	1 ruta corta

t_{pd}	t_{cd}
2) 95 Ps	3) 70 Ps
4) 85 Ps	4) 70 Ps
5) 85 Ps	5) 70 Ps
3 rutas criticas	3 rutas cortas



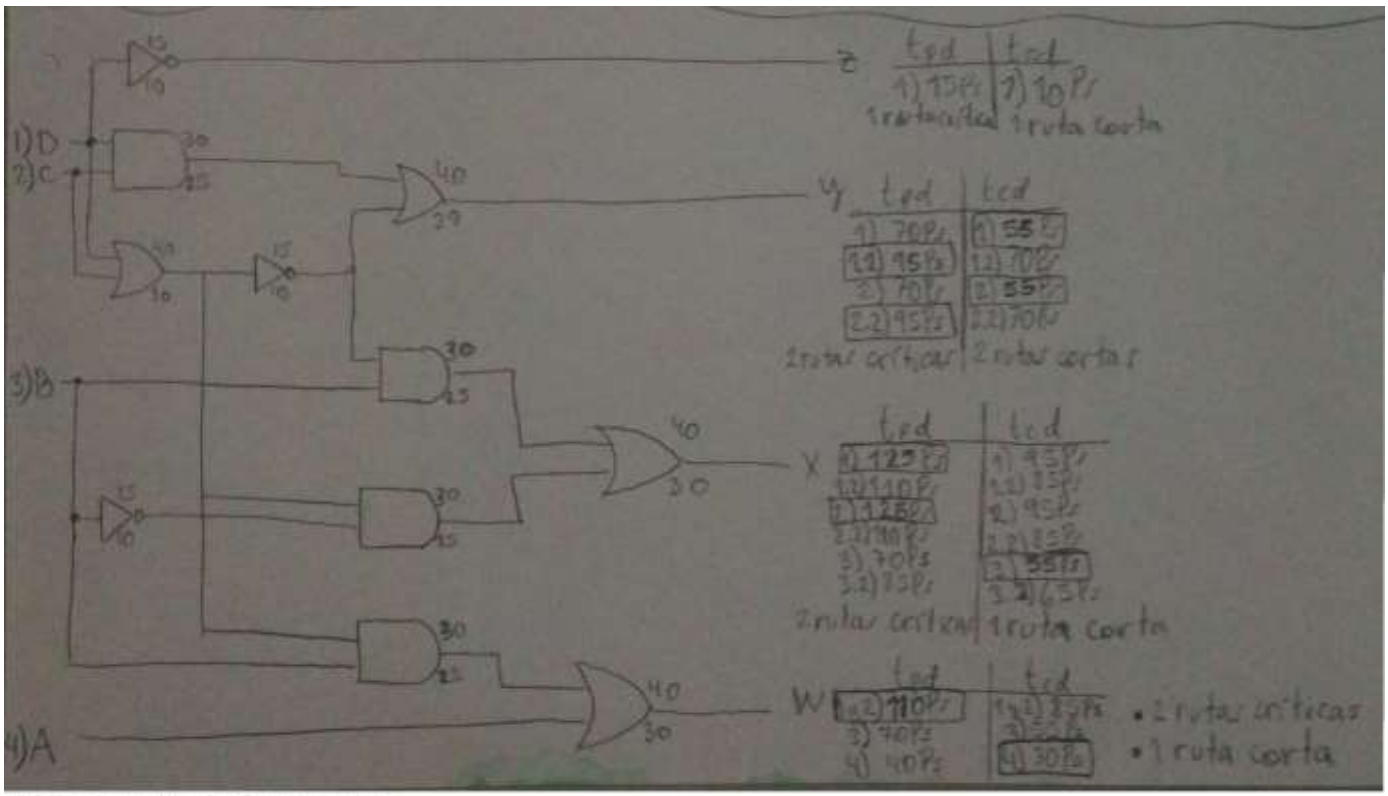
t_{pd}	t_{cd}
1) 65 Ps	1) 50 Ps
1.2) 85 Ps	1.2) 65 Ps
1.3) 85 Ps	1.3) 65 Ps
2) 100 Ps	2) 80 Ps
3) 70 Ps	3) 55 Ps
1 ruta critica	1 ruta corta

t_{pd}	t_{cd}
1) 85 Ps	1) 65 Ps
1.2) 65 Ps	1.2) 50 Ps
2) 100 Ps	2) 80 Ps
3) 50 Ps	3) 40 Ps
1 ruta critica	1 ruta corta



t_{pd}	t_{cd}
1) 80 Ps	1) 60 Ps
1.2) 145 Ps	1.2) 105 Ps
2) 125 Ps	2) 95 Ps
2.2) 190 Ps	2.2) 95 Ps
3) 100 Ps	3) 85 Ps
4) 100 Ps	4) 70 Ps
1 ruta crítica	1 ruta corta

t_{pd}	t_{cd}
1) 85 Ps	1) 65 Ps
2) 70 Ps	2) 55 Ps
4) 55 Ps	1) 40 Ps
1 ruta crítica	1 ruta corta



Z

t_{pd}	t_{cd}
1) 75 Ps	1) 70 Ps
1 ruta crítica	1 ruta corta

Y

t_{pd}	t_{cd}
1) 70 Ps	1) 55 Ps
1.2) 95 Ps	1.2) 70 Ps
2) 70 Ps	2) 55 Ps
2.2) 95 Ps	2.2) 70 Ps
2 rutas críticas	2 rutas cortas

X

t_{pd}	t_{cd}
1.2) 125 Ps	1) 95 Ps
1.2.1) 110 Ps	1.2) 95 Ps
2) 125 Ps	2) 95 Ps
2.2) 190 Ps	2.2) 85 Ps
3) 70 Ps	2) 55 Ps
3.2) 75 Ps	3.2) 65 Ps
2 rutas críticas	1 ruta corta

W

t_{pd}	t_{cd}
1.2.2) 110 Ps	1.2.2) 85 Ps
3) 70 Ps	3) 55 Ps
4) 40 Ps	4) 30 Ps

- 2 rutas críticas
- 1 ruta corta

Ejercicio4.v

```
//Jose Alejandro Duarte Andrade

// Se realiza el modulo del Mux 2:1
module Mux2(input wire S, D0, D1, output wire Y);
    //estableciendo conexiones
  assign Y = S ? D1 : D0;

// Se finaliza el modulo
endmodule

// Se realiza el modulo del Mux 4:1
module Mux4(input wire S0, S1, D0, D1, D2, D3,  output wire Y1);
  // estableciendo conexiones
  wire cbl1, cbl2;
  // Se instancia el Mux2:1
  Mux2 caja0(S0, D0, D1, cbl1);
  Mux2 caja1(S0, D2, D3, cbl2);
  Mux2 caja2(S1, cbl1, cbl2, Y1);

//Se finaliza el modulo
endmodule

// Se realiza el modulo del Mux 8:1
module Mux8(input wire D0, D1, D2, D3, D4, D5, D6, D7, S0, S1, S2, output wire Y2);
  // Estableciendo conexiones
  wire cbl3, cbl4;
  // se instancian el Mux 4:1 y Mux 2:1
  Mux4 caja01(S0, S1, D0, D1, D2, D3, cbl3);
  Mux4 caja02(S0, S1, D4, D5, D6, D7, cbl4);
  Mux2 caja03(S2, cbl3, cbl4, Y2);

//Se finaliza el modulo
endmodule

// Se realiza el Mux 8:1 del ejercicio 1 de la tabla 1
module E1T1M8(input wire A, B, C, output Y4);

// Se realiza el Mux 8:1 del ejercicio 1 de la tabla 1
module E1T1M8(input wire A, B, C, output Y4);

  wire V, G;
  assign V = 1;
  assign G = 0;
  //Se instancia el Mux 8:1
  Mux8 ejc11(G, V, V, G, V, G, G, V, A, B, C, Y4);
  //Finaliza el modulo
endmodule

// Se realiza el Mux 4:1 del ejercicio 2 de la tabla 1
module E1T1M4(input wire A, B, C, output Y5);
  //se asignan variables
  wire NC;
  assign NC = ~C;

  //Se instancia el Mux 4:1
  Mux4 ejc112( A, B, C, NC, NC, C, Y5);
endmodule

// Se realiza el Mux 2:1 del ejercicio 3 de la tabla 1
module E1T1M2(input wire A, B, C, output Y6);

  wire XOR,NXOR;
  assign XOR = B ^ C;
  assign NXOR = B ~^ C;

  Mux2 ejc113(A, XOR, NXOR, Y6);

endmodule

module E1T2M8(input wire A, B, C, output Y7);

  wire V, G;
  assign V = 1;
  assign G = 0;
```

Ejercicio4_tb.v

```
module testbench();

reg r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12,r13, r14, r15;
reg r16, r17, r18;
wire led0, led1, led2, led3, led4, led5;

E1T1M8 l1(r1, r2, r3, led0);
E1T1M4 l2(r4, r5, r6, led1);
E1T1M2 l3(r7, r8, r9, led2);
E1T2M8 l4(r10, r11, r12, led3);
E1T2M4 l5(r13, r14, r15, led4);
E1T2M2 l6(r16, r17, r18, led5);

initial begin //Mux 8:1 del ejercicio 1 de la tabla 1
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r1, r2, r3, led0);
      r1 = 0; r2 = 0; r3 = 0;
    #1r1 = 0; r2 = 0; r3 = 1;
    #1r1 = 0; r2 = 1; r3 = 0;
    #1r1 = 0; r2 = 1; r3 = 1;
    #1r1 = 1; r2 = 0; r3 = 0;
    #1r1 = 1; r2 = 0; r3 = 1;
    #1r1 = 1; r2 = 1; r3 = 0;
    #1r1 = 1; r2 = 1; r3 = 1;
end

initial begin//el Mux 4:1 del ejercicio 2 de la tabla 1
  #8
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r4, r5, r6, led1);
      r4 = 0; r5 = 0; r6 = 0;
    #1r4 = 0; r5 = 0; r6 = 1;
    #1r4 = 0; r5 = 1; r6 = 0;
```

Ejercicio4.v

```
// Se realiza el Mux 8:1 del ejercicio 1 de la tabla 1
module E1T1M8(input wire A, B, C, output Y4);

  wire V, G;
  assign V = 1;
  assign G = 0;
  //Se instancia el Mux 8:1
  Mux8 ejc11(G, V, V, G, V, G, G, V, A, B, C, Y4);
  //Finaliza el modulo
endmodule

// Se realiza el Mux 4:1 del ejercicio 2 de la tabla 1
module E1T1M4(input wire A, B, C, output Y5);
  //se asignan variables
  wire NC;
  assign NC = ~C;

  //Se instancia el Mux 4:1
  Mux4 ejc112( A, B, C, NC, NC, C, Y5);
endmodule

// Se realiza el Mux 2:1 del ejercicio 3 de la tabla 1
module E1T1M2(input wire A, B, C, output Y6);

  wire XOR,NXOR;
  assign XOR = B ^ C;
  assign NXOR = B ~^ C;

  Mux2 ejc113(A, XOR, NXOR, Y6);

endmodule

module E1T2M8(input wire A, B, C, output Y7);

  wire V, G;
  assign V = 1;
  assign G = 0;
```

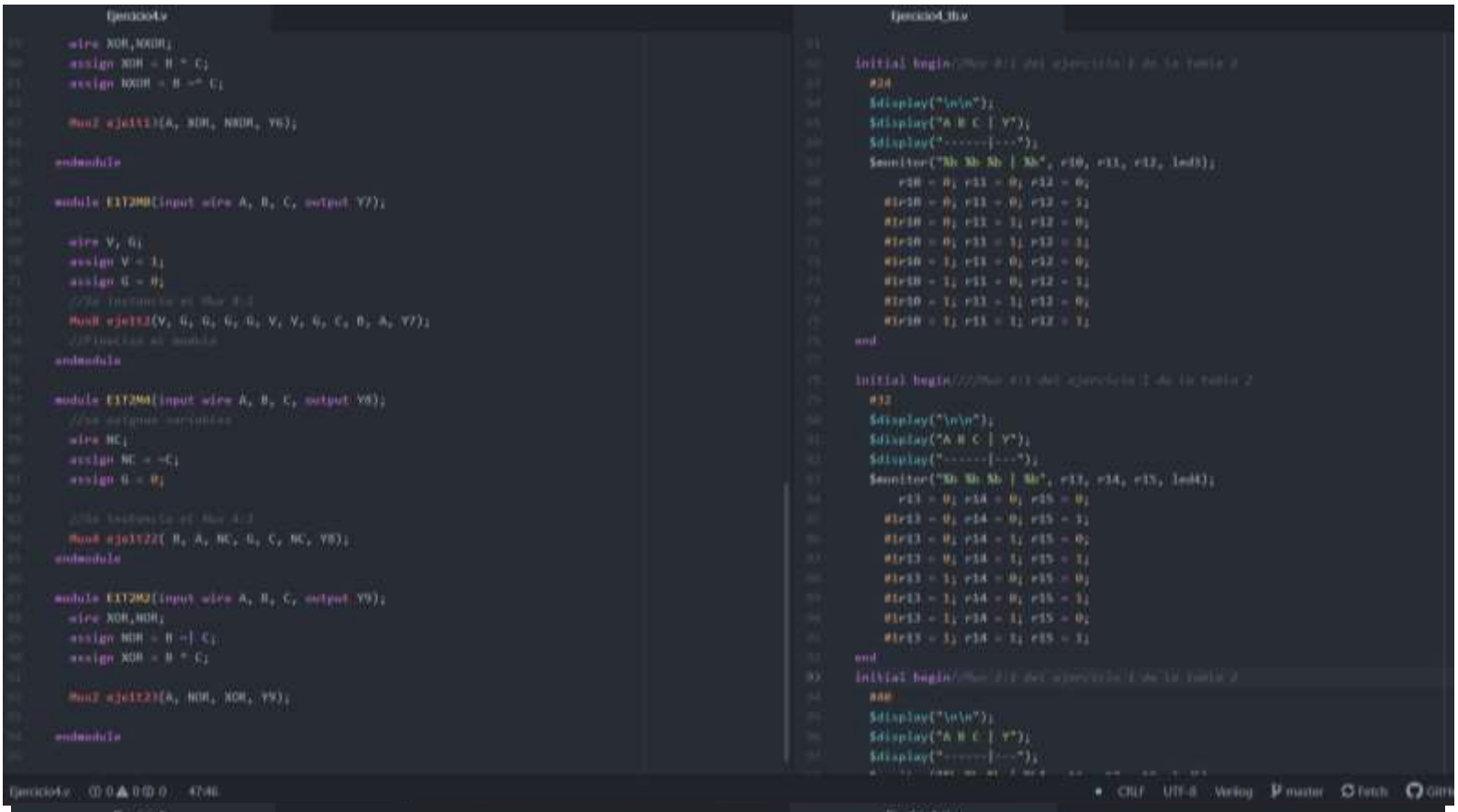
Ejercicio4_tb.v

```
    #1r4 = 0; r5 = 1; r6 = 0;
    #1r4 = 0; r5 = 1; r6 = 1;
    #1r4 = 1; r5 = 0; r6 = 0;
    #1r4 = 1; r5 = 0; r6 = 1;
    #1r4 = 1; r5 = 1; r6 = 0;
    #1r4 = 1; r5 = 1; r6 = 1;

end

initial begin//el Mux 2:1 del ejercicio 3 de la tabla 1
  #16
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r7, r8, r9, led2);
      r7 = 0; r8 = 0; r9 = 0;
    #1r7 = 0; r8 = 0; r9 = 1;
    #1r7 = 0; r8 = 1; r9 = 0;
    #1r7 = 0; r8 = 1; r9 = 1;
    #1r7 = 1; r8 = 0; r9 = 0;
    #1r7 = 1; r8 = 0; r9 = 1;
    #1r7 = 1; r8 = 1; r9 = 0;
    #1r7 = 1; r8 = 1; r9 = 1;
end

initial begin//Mux 8:1 del ejercicio 1 de la tabla 2
  #24
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r10, r11, r12, led3);
      r10 = 0; r11 = 0; r12 = 0;
    #1r10 = 0; r11 = 0; r12 = 1;
    #1r10 = 0; r11 = 1; r12 = 0;
    #1r10 = 0; r11 = 1; r12 = 1;
    #1r10 = 1; r11 = 0; r12 = 0;
    #1r10 = 1; r11 = 0; r12 = 1;
```



```
Ejercicio4.v

    wire XOR,NXOR;
    assign XOR = B ^ C;
    assign NXOR = B ~^ C;

    Mux2 ejelt1(A, XOR, NXOR, Y6);

endmodule

module E1T2M0(input wire A, B, C, output Y7);

    wire V, G;
    assign V = 1;
    assign G = 0;
    //Se instancia el Mux 8:1
    Mux8 ejelt2(V, G, G, G, G, V, V, G, C, B, A, Y7);
    //Finaliza el module
endmodule

module E1T2M4(input wire A, B, C, output Y8);
    //Se asignan variables
    wire NC;
    assign NC = ~C;
    assign G = 0;

    //Se instancia el Mux 8:1
    Mux8 ejelt22( B, A, NC, G, C, NC, Y8);
endmodule

module E1T2M2(input wire A, B, C, output Y9);
    wire XOR,NOR;
    assign NOR = B ~| C;
    assign XOR = B ^ C;

    Mux2 ejelt23(A, NOR, XOR, Y9);

endmodule
```

```
Ejercicio4_tb.v

initial begin//Mux 8:1 del ejercicio 1 de la tabla 2
  #24
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r10, r11, r12, led3);
      r10 = 0; r11 = 0; r12 = 0;
    #1r10 = 0; r11 = 0; r12 = 1;
    #1r10 = 0; r11 = 1; r12 = 0;
    #1r10 = 0; r11 = 1; r12 = 1;
    #1r10 = 1; r11 = 0; r12 = 0;
    #1r10 = 1; r11 = 0; r12 = 1;
    #1r10 = 1; r11 = 1; r12 = 0;
    #1r10 = 1; r11 = 1; r12 = 1;
end

initial begin///Mux 8:1 del ejercicio 1 de la tabla 2
  #32
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r13, r14, r15, led4);
      r13 = 0; r14 = 0; r15 = 0;
    #1r13 = 0; r14 = 0; r15 = 1;
    #1r13 = 0; r14 = 1; r15 = 0;
    #1r13 = 0; r14 = 1; r15 = 1;
    #1r13 = 1; r14 = 0; r15 = 0;
    #1r13 = 1; r14 = 0; r15 = 1;
    #1r13 = 1; r14 = 1; r15 = 0;
    #1r13 = 1; r14 = 1; r15 = 1;
end
initial begin//Mux 8:1 del ejercicio 1 de la tabla 2
  #40
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
```


```
Ejercicio4.v

    wire XOR,NXOR;
    assign XOR = B ^ C;
    assign NXOR = B ~^ C;

    Mux2 ejelt1(A, XOR, NXOR, Y6);

endmodule

module E1T2M0(input wire A, B, C, output Y7);

    wire V, G;
    assign V = 1;
    assign G = 0;
    //Se instancia el Mux 8:1
    Mux8 ejelt2(V, G, G, G, G, V, V, G, C, B, A, Y7);
    //Finaliza el module
endmodule

module E1T2M4(input wire A, B, C, output Y8);
    //Se asignan variables
    wire NC;
    assign NC = ~C;
    assign G = 0;

    //Se instancia el Mux 8:1
    Mux8 ejelt22( B, A, NC, G, C, NC, Y8);
endmodule

module E1T2M2(input wire A, B, C, output Y9);
    wire XOR,NOR;
    assign NOR = B ~| C;
    assign XOR = B ^ C;

    Mux2 ejelt23(A, NOR, XOR, Y9);

endmodule
```


```
Ejercicio4_tb.v

    #1r13 = 0; r14 = 0; r15 = 1;
    #1r13 = 0; r14 = 1; r15 = 0;
    #1r13 = 0; r14 = 1; r15 = 1;
    #1r13 = 1; r14 = 0; r15 = 0;
    #1r13 = 1; r14 = 0; r15 = 1;
    #1r13 = 1; r14 = 1; r15 = 0;
    #1r13 = 1; r14 = 1; r15 = 1;
end
initial begin//Mux 8:1 del ejercicio 2 de la tabla 2
  #48
  $display("\n\n");
  $display("A B C | Y");
  $display("------|---");
  $monitor("%b %b %b | %b", r16, r17, r18, led5);
      r16 = 0; r17 = 0; r18 = 0;
    #1r16 = 0; r17 = 0; r18 = 1;
    #1r16 = 0; r17 = 1; r18 = 0;
    #1r16 = 0; r17 = 1; r18 = 1;
    #1r16 = 1; r17 = 0; r18 = 0;
    #1r16 = 1; r17 = 0; r18 = 1;
    #1r16 = 1; r17 = 1; r18 = 0;
    #1r16 = 1; r17 = 1; r18 = 1;
end

initial begin
    $dumpfile("Ejercicio4_tb.vcd");
    $dumpvars(0, testbench5);
  end

initial
    #48$finish;



endmodule
```