INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

DISEÑO LÓGICO

II SEMESTRE

TAREA 1

PROFESOR: PABLO DANIEL MENDOZA PONCE

ESTUDIANTE: JOSÉ JULIÁN CAMACHO HERNÁNDEZ

CARNÉ: 2019201459

14 DE SETIEMBRE DE 2021

1, 2

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459

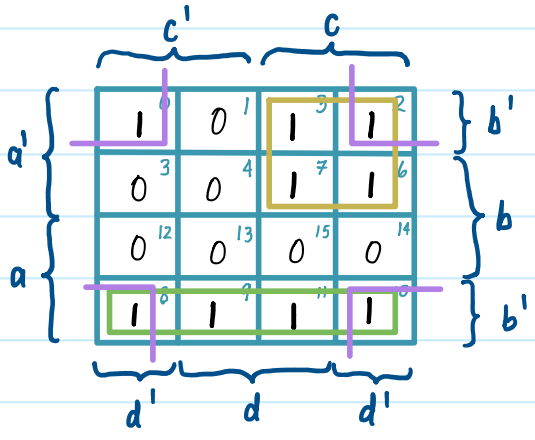
1. Expanda en forma canónica SOP la siguiente expresión

$g(x,b) = \overline{b} + x$

x	b	$g = \bar{b} + x$	Minterm.
0	0	1	$\bar{x}\,\bar{b}$
0	1	0	$\bar{x}b$
1	0	1	$x\bar{b}$
1	1	1	xb

$\Rightarrow$ $g(x,b) = \bar{x}\bar{b} + x\bar{b} + xb$

2. Reduzca el siguiente polinomio usando mapas de Karnaugh
$H(a,b,c,d) = \Sigma_m(0,2,3,6,7,8,9,10,11)$



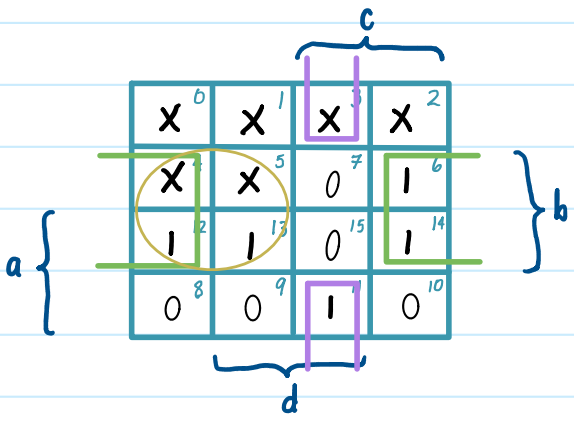
$H = a'c + ab' + b'd'$

3, 4

viernes, 10 de septiembre de 2021 14:07

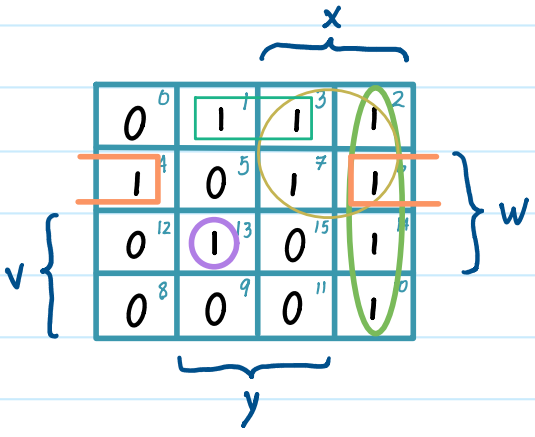
José Julián Camacho Hernández
2019201459

3. En la siguiente función el símbolo Ø indica aquellas posiciones "no importa". Reduzca usando mapas de Karnaugh (exprese en SOP):
$Q(a,b,c,d) = \Sigma_m(6,11,12,13,14) + Ø(0,1,2,3,4,5)$



$Q = bc' + bd' + b'cd$

4. Reduzca la siguiente ecuación (POS) usando mapas de Karnaugh. Exprese su resultado como SOP.
$Q(v,w,x,y) = \Pi_M(0,5,8,9,11,12,15)$ → ceros



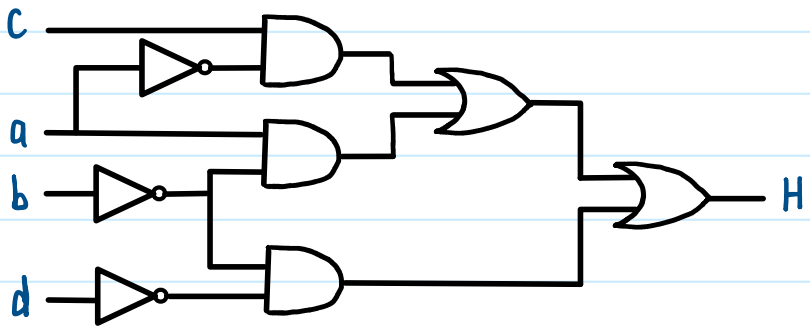
$Q = xy' + v'x + v'wy' + v'w'y + vwx'y$

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459

5. Dibuje el circuito simplificado que se obtiene de la solución del ejercicio 2

$$H = a'c + ab' + b'd'$$



6.a

viernes, 10 de septiembre de 2021 14:07

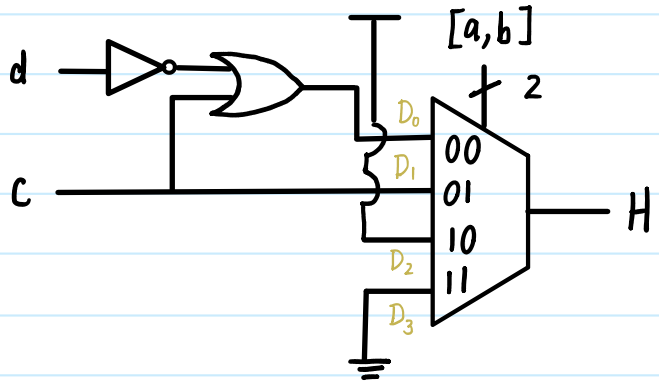
José Julián Camacho Hernández
2019201459

6. Implemente el sistema usado en ejercicio 2 (sin simplificar) usando:
   a. Un multiplexor 4 a 1
   b. Un multiplexor 2 a 1

La tabla de verdad completa del sistema del ejercicio 2 es:

	S							
	a	b	c	d	d'+c	Mintérm.	H	
0	0	0	0	0	1	a'b'c'd'	1	d'+c
1	0	0	0	1	0	a'b'c'd	0	
2	0	0	1	0	1	a'b'cd'	1	
3	0	0	1	1	1	a'b'cd	1	
4	0	1	0	0	0	a'bc'd'	0	C (puede ser también d'+c, pero se elige C para tener menos retrasos)
5	0	1	0	1	0	a'bc'd	0	
6	0	1	1	0	1	a'bcd'	1	
7	0	1	1	1	1	a'bcd	1	
8	1	0	0	0	1	ab'c'd'	1	1 (V_{cc})
9	1	0	0	1	0	ab'c'd	1	
10	1	0	1	0	1	ab'cd'	1	
11	1	0	1	1	1	ab'cd	1	
12	1	1	0	0	1	abc'd'	0	0 (tierra)
13	1	1	0	1	0	abc'd	0	
14	1	1	1	0	1	abcd'	0	
15	1	1	1	1	1	abcd	0	

Entonces el sistema queda como:



Nota:

Intenté hacer algunos cambios en el orden de las columnas.
El que se muestra a continuación fue de las mejores soluciones encontradas.
Pero la mejor solución que encontré fue la que se expuso anteriormente, que es el sistema sin cambiar columnas.

a	b	d	c	d'+c	H	
0	0	0	0	1	1	d'+c
0	0	0	1	1	1	
0	0	1	0	0	0	
0	0	1	1	1	1	
0	1	0	0		0	C
0	1	0	1		1	
0	1	1	0		0	
0	1	1	1		1	
1	0	0	0		1	1 (V_{cc})
1	0	0	1		1	
1	0	1	0		1	
1	0	1	1		1	
1	1	0	0		0	0 (tierra)
1	1	0	1		0	
1	1	1	0		0	
1	1	1	1		0	

6.b

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459

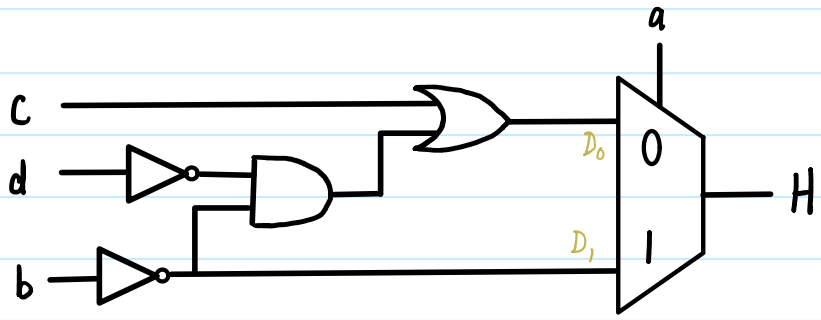
6. Implemente el sistema usado en ejercicio 2 (sin simplificar) usando:
   a. Un multiplexor 4 a 1
   b. Un multiplexor 2 a 1

La tabla de verdad completa del sistema del ejercicio 2 es:

S

a	b	c	d	b'd'	b'd'+c	H	
0	0	0	0	1	1	1	b'd'+c
0	0	0	1	0	0	0	
0	0	1	0	1	1	1	
0	0	1	1	0	1	1	
0	1	0	0	0	0	0	
0	1	0	1	0	0	0	
0	1	1	0	0	1	1	
0	1	1	1	0	1	1	
1	0	0	0	1	1	1	NOT b
1	0	0	1	0	0	1	
1	0	1	0	1	1	1	
1	0	1	1	0	1	1	
1	1	0	0	0	0	0	
1	1	0	1	0	0	0	
1	1	1	0	0	1	0	
1	1	1	1	0	1	0	

Entonces el sistema queda como:



7

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459

7. Escriba el código verilog que implemente un multiplexor 2 a 1, además escriba un "testbench" para su simulación. Muestre su simulación. Recuerde que su código (del módulo) debe ser sintetizable.

testbench.sv

```
`timescale 1ns/1ps

module mux_tb;

  reg dato0, dato1, seleccion;
  wire salida;

  mux DUT (.S(seleccion),
           .D0(dato0),
           .D1(dato1),
           .Q(salida)
  );

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

  initial begin
    dato0 = 0;
    dato1 = 1;
    seleccion = 0;
    #10 //Retardo de 10 unidades de tiempo
    dato0 = 1;
    #10
    dato1 = 0;
    #5
    seleccion = 1;
    #2
    dato0 = 1;
    #10
    dato1 = 0;
    #5
    seleccion = 0;
    #10

    $finish;

  end
endmodule
```

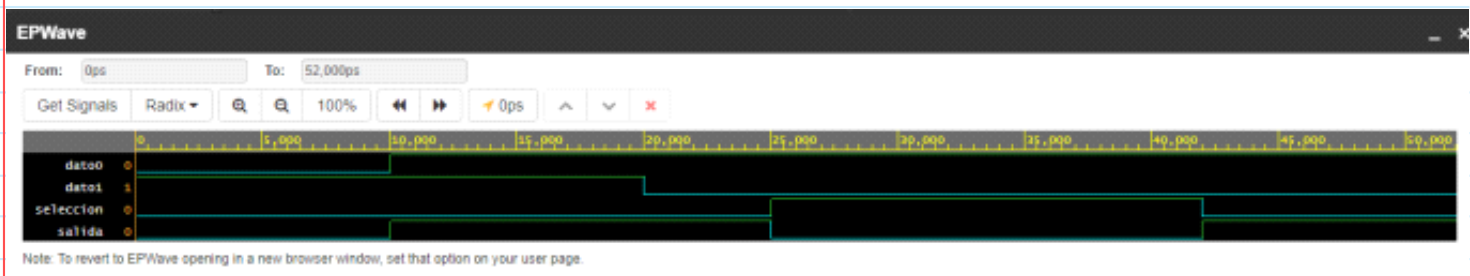
design.sv

```
// Code your design here


//=========== Mux 2 a 1 ===========//

`timescale 1ns/1ps

module mux(
    input D0, D1, S,
    output reg Q
);

  always @ (*) begin
    case(S)
      0: Q = D0;
      1: Q = D1;
    endcase
  end

endmodule
```

7 Simulación

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459



En esta simulación la selección empieza en 0, por lo que la salida refleja el valor de dato0.
A los 10,000 dato0 cambia y la salida también porque la selección no cambió.
A los 25,000 la selección pasa a 1 por lo que la salida es dato1.

Nota:
En la carpeta de ejercicio 7 se adjunta la imagen (png) para mayor resolución

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459

8. Haciendo uso del multiplexor implementado, escriba el código verilog para implementar la solución del ejercicio 6.b (nota: haga una instancia de su multiplexor). Muestre su simulación. Recuerde que su código (del módulo) debe ser sintetizable.

testbench.sv

```
1  //Pruebas Ejercicio 6
2
3  `timescale 1ns/1ps
4
5  module sol6b_tb;
6
7    reg dato0, dato1, dato2, seleccion;
8    wire salida;
9
10   sol6b DUT (.A(seleccion),
11              .B(dato0),
12              .C(dato1),
13              .D(dato2),
14              .H(salida)
15   );
16
17   initial begin
18     $dumpfile("sol6b_tb.vcd");
19     $dumpvars(0,sol6b_tb);
20   end
21
22   initial begin
23     //Combinación 0000 ==> H = 1
24     seleccion = 0;
25     dato0 = 0;
26     dato1 = 0;
27     dato2 = 0;
28     #10
29     //Combinación 0001 ==> H = 0
30     dato2 = 1;
31     #10
32     //Combinación 0010 ==> H = 1
33     dato1 = 1;
34     dato2 = 0;
35     #10
36     //Combinación 1010 ==> H = 1
37     seleccion = 1;
38     #10
39     //Combinación 1110 ==> H = 0
40     dato0 = 1;
41     #10
42
43     $finish;
44
45   end
46 endmodule
```

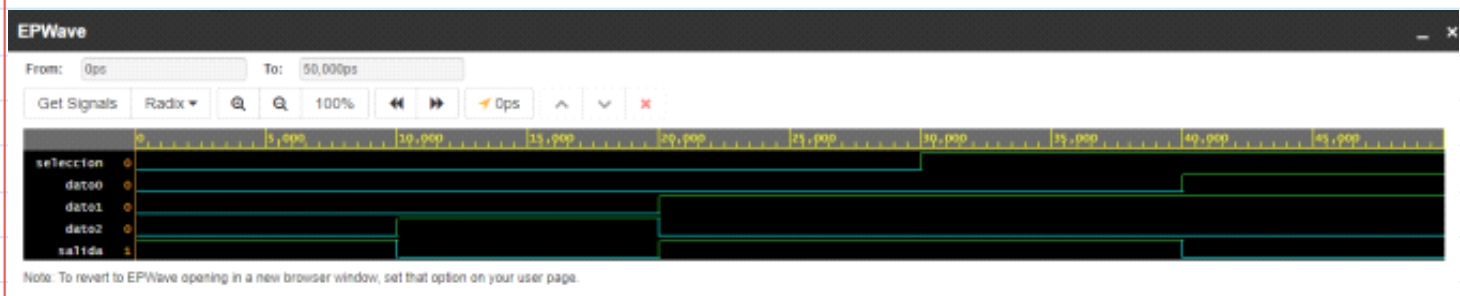
design.sv

```
22
23 //=========== Ejercicio 6b ===========//
24
25 module sol6b(
26     input A, B, C, D,
27     output reg H
28 );
29
30   wire Dneg, Bneg, n0, n1;
31
32   assign Dneg = ~D;
33   assign Bneg = ~B;
34   assign n0 = Dneg & Bneg;
35   assign n1 = n0 | C;
36
37   mux m1 (.S(A),
38            .D0(n1),
39            .D1(Bneg),
40           .Q(H)
41   );
42
43 endmodule
```

8 Simulación

viernes, 10 de septiembre de 2021 14:07

José Julián Camacho Hernández
2019201459



En esta simulación se asignan ciertas combinaciones de selección (a), dato0 (b), dato1 (c) y dato2 (d) para comprobar que su salida es la misma que en la tabla de verdad del ejercicio 6b. Por ejemplo, la combinación inicial (0000) tiene salida igual a 1 tal como se muestra en la tabla.

Nota:

En la carpeta de ejercicio 8 se adjunta la imagen (png) para mayor resolución