Soluciones Tema 3

Ejercicio 1.

$$n \geq \frac{\log_{10}(x+1)}{\log_{10}(2)}$$

a) 101011001b
b) 100000000b
c) 11111000011b
d) 1011101001b

Ejercicio 2.

-Si el número finaliza en '1' será impar sino será par.

a) 1001000110011101b ->$2^0+2^2+2^3+2^4+2^7+2^8+2^{12}+2^{15}$=37277
b) 0111000111100011b ->$2^0+2^1+2^5+2^6+2^7+2^8+2^{12}+2^{13}+2^{14}$=29155
c) 1111111100000110b ->$2^1+2^2+2^8+2^9+2^{10}+2^{11}+2^{12}+2^{13}+2^{14}+2^{15}$=65286
d) 0011110011010110b ->$2^1+2^2+2^4+2^6+2^7+2^{10}+2^{11}+2^{12}+2^{13}$=15574

Ejercicio 3.

a) 1001000110011101b -> 110635o -> 919Dh
b) 0111000111100011b -> 70743o -> 71E3h
c) 1111111100000110b -> 177406o -> FF06h
d) 0011110011010110b -> 36326o -> 3CD6h

Ejercicio 4.

$$-\left(2^{n-1}\right) \leq x \leq 2^{n-1}-1 = -32768 \leq x \leq 32767$$

a) 1053 -> 0000010000011101
b) 4017 -> 0000111110110001
c) –356 -> 1111111010011100
d) -8950 -> 1101110100001010

Ejercicio 5.

-Si el bit de la izquierda es un '1' será un número negativo.

a) 0001111001110101 ->$2^0+2^2+2^4+2^5+2^6+2^9+2^{10}+2^{11}+2^{12}$=7797
b) 1111111111111111 ->$2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7+2^8+2^9+2^{10}+2^{11}+2^{12}+2^{13}+2^{14}-(2^{15})$=-1
c) 0001100111100001 ->$2^0+2^5+2^6+2^7+2^8+2^{11}+2^{12}$=6625
d) 1111100000011100 ->$2^2+2^3+2^4+2^{11}+2^{12}+2^{13}+2^{14}-(2^{15})$=-2020

Ejercicio 6.

I.

a) –125-> 10000011
b) 98 -> 01100010
c) 15 -> 00001111
d) -12 ->11110100
e) 85 -> 01010101
f) -93-> 10100011

II.

a) b+e -> overflow

```
  01100010
+ 01010101
----------
  10110111 = -73
```

c) a-f

```
  10000011
+ 01011101
----------
  11100000 = -32
```

e) d-a

```
  11110100
+ 01111101
----------
  01110001 = 113
```

b) e-d

```
  01010101
+ 00001100
----------
  01100001 = 97
```

d) b+c

```
  01100010
+ 00001111
----------
  01110001 = 113
```

f) a+d-> overflow

```
  10000011
+ 11110100
----------
  01110111 = 119
```

Ejercicio 7.

Se desea diseñar un sistema que realice la suma o la resta de dos números de 4 bits codificados en complemento a 2. La operación realizada por el circuito será función de una variable lógica denominada R: R=1 operación de resta, R=0 operación de suma.. La salidas del sistema serán por un lado cuatro bits correspondientes al resultado de la operación, y por otro un bit más que tomará el valor 1 si se produce overflow al realizar la operación. La Fig. 1 representa las entradas y salidas del sistema.

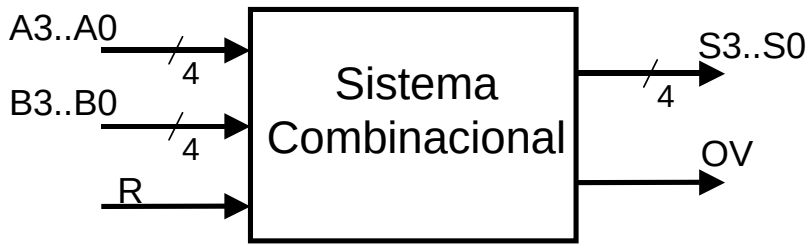
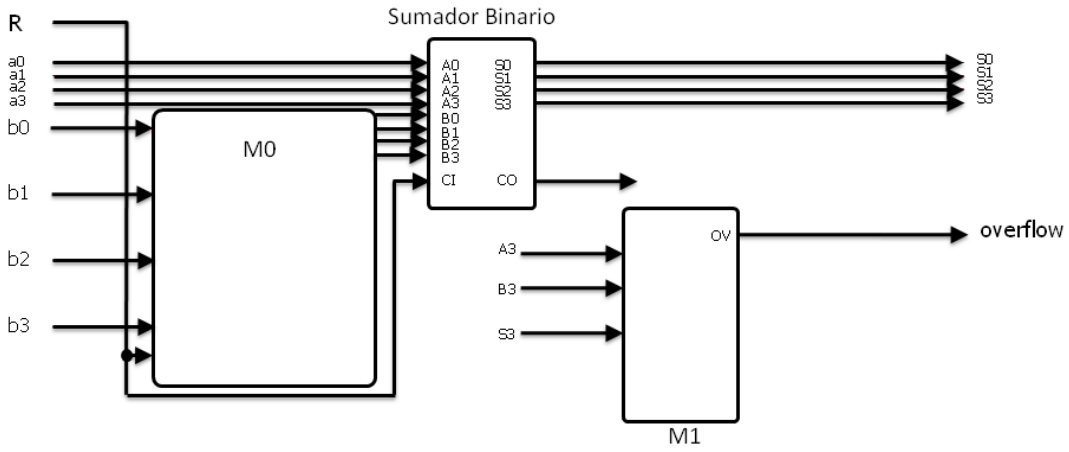


Figura 1 . Definición del sistema lógico a diseñar

a) Implementación del sistema utilizando un sumador binario de dos números de 4 bits y las puertas lógicas que se consideren oportunas.



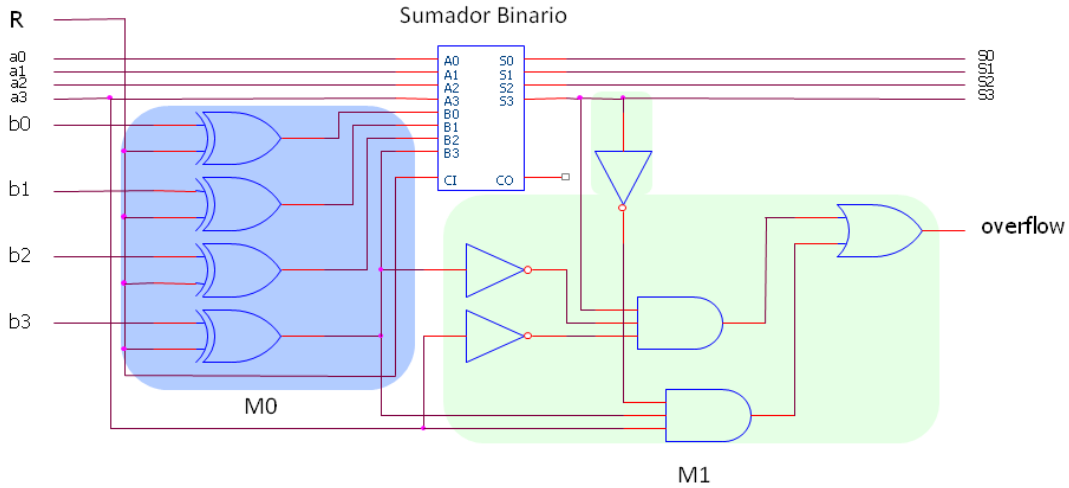
-M0: Si R='1' Calculo del complemento a 1,

$$B0 = \overline{b0},\ B1 = \overline{b1},\ B2 = \overline{b2}\ \ y\ \ B3 = \overline{b3}$$

Si R='0'

$$B0 = b0,\ B1 = b1,\ B2 = b2\ \ y\ \ B3 = b3$$

-M1: Se activará la salida OV siempre que se produzca "overflow".



b)

c.1) a=1011b; b=1100b; R=1

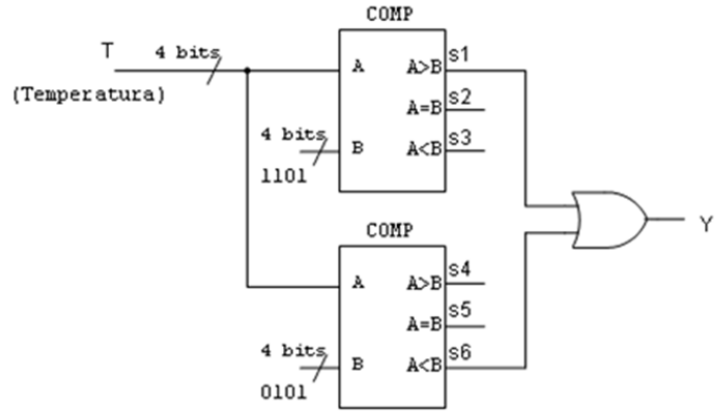
```
  1011
+ 0100
-----------------
  1111 = -1   overflow='0'
```

c.2) a=1110b; b=1111b; R=0

```
  1110
+ 1111
-----------------
  1101= -3   overflow='0'
```

Ejercicio 8.

a) Tendremos que comparar la temperatura que nos proporciona el sensor, con los dos márgenes de temperatura, y si estamos dentro de dichos márgenes activaremos la salida del sistema. Recordar que los comparadores interpretarán los números presentes en la entrada como si estuvieran codificados en binario natural, pero en realidad están en complemento a 2, así que lo tendremos que tener en cuenta a la hora de obtener los dos umbrales de comparación.



b)

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
USE IEEE.std_logic_unsigned.all;

ENTITY Ejercicio_9 IS
```

```
   PORT (T: IN std_logic_vector(3 downto 0);
         Y: OUT std_logic);

END Ejercicio_9;


ARCHITECTURE Estructural OF Ejercicio_9 IS
   -- Declaración del componente COMPARADOR.
   COMPONENT Comparador_4bits IS
         PORT (A, B : IN std_logic_vector(3 downto 0);
               A_mayor, A_igual, A_menor : OUT std_logic);

   END COMPONENT;

   signal s1,s2,s3,s4,s5,s6 : std_logic; --Señales internas.

BEGIN
   U1: Comparador_4bits PORT MAP(A=>T,
                                 B=>"1101",
                                 A_mayor=>s1,
                                 A_igual=>s2,
                                 A_menor=>s3);
   U2: Comparador_4bits PORT MAP(A=>T,
                                 B=>"0101",
                                 A_mayor=>s4,
                                 A_igual=>s5,
                                 A_menor=>s6);

   Y<=s1 or s6;

END Estructural;




--Descripción del comparador de 4 bits

ENTITY Comparador_4bits IS
   PORT (A, B : IN std_logic_vector(3 downto 0);
               A_mayor, A_igual, A_menor : OUT std_logic);

END Comparador_4bits;

ARCHITECTURE comparador OF Comparador_2bits IS
BEGIN
   A_mayor<='1' WHEN (A>B) ELSE '0';
   A_igual<='1' WHEN (A=B) ELSE '0';
   A_menor<='1' WHEN (A<B) ELSE '0';
END comparador ;
```

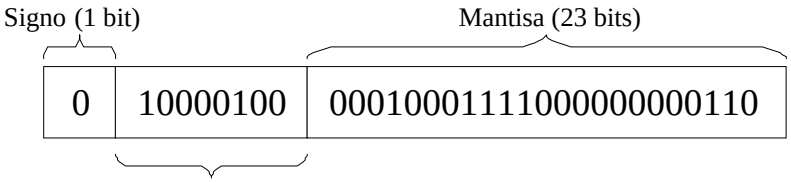
Ejercicio 9.

a) 34,2344-> 1000100011110000
b) 52,1878-> 1101000011000000
c) 15,0625-> 0011110001000000

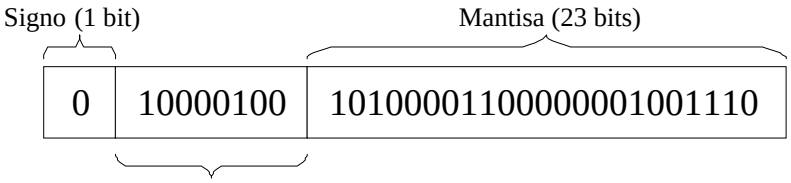
d) 60,7854-> 1111001100100100

Ejercicio 10.

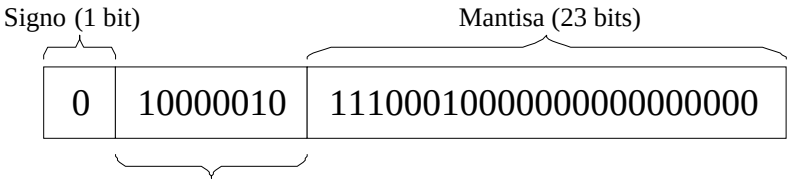
a) 34,2344->01000010000010001111000000000110



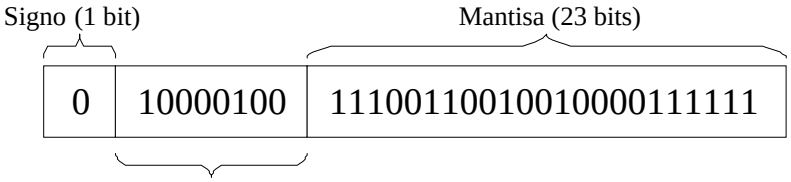
b) 52,1878-> 01000010010100001100000001001110



c) 15,0625-> 01000001011100010000000000000000

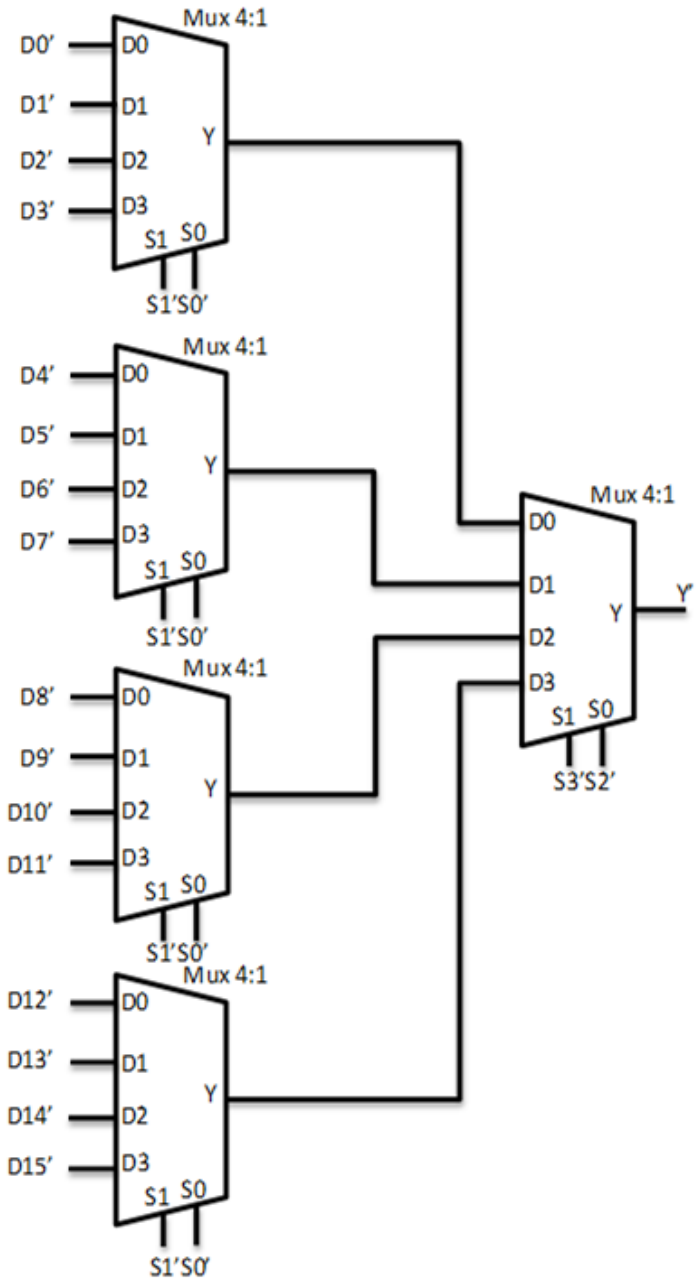


d) 60,7854-> 01000010011100110010010000111111



Ejercicio 11.

- El superíndice " ' " denota las entradas y salidas del multiplexor 16:1 a implementar.



Ejercicio 12.

a)

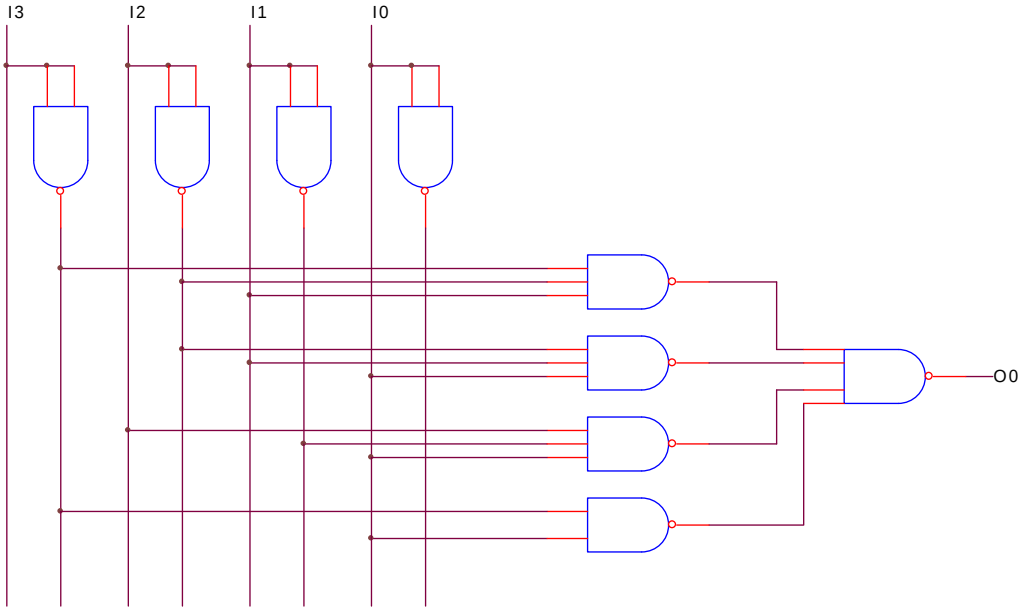
I3	I2	I1	I0	O0	O1
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	1	0	1

b)

I_3I_2 / I_1I_0	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	1	1	0	1
10	1	0	0	0

$$O0 = \overline{I3}\cdot\overline{I2}\cdot I1 + \overline{I2}\cdot I1\cdot I0 + I2\cdot\overline{I1}\cdot I0 + \overline{I3}\cdot I0$$

c)

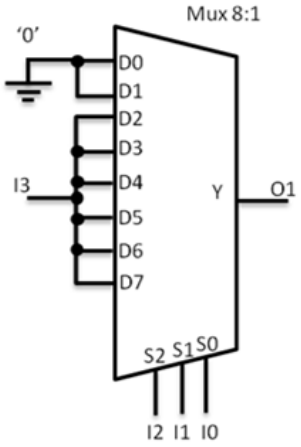


d)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


   entity Ejercicio_13 is
         port(I3,I2,I1,I0 : in std_logic;
                     O0,O1 : out std_logic);
   end Ejercicio_13;

   architecture Behavioral of Ejercicio_13 is

   signal entrada: std_logic_vector (3 downto 0);
   begin
         entrada<=I3&I2&I1&I0;
          O0<=((not I3)and (not(I2) and I1) or ((not I2) and I1 and I0)
               or (I2 and (not I1) and I0) or ((not I3) and I0);

          O1<='1' when (entrada>9) else '0';

   end Behavioral;
```

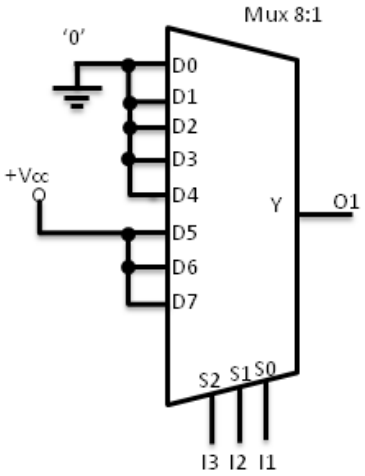
e)

I2	I1	I0	O1
0	0	0	0
0	0	1	0
0	1	0	I3
0	1	1	I3
1	0	0	I3
1	0	1	I3
1	1	0	I3
1	1	1	I3

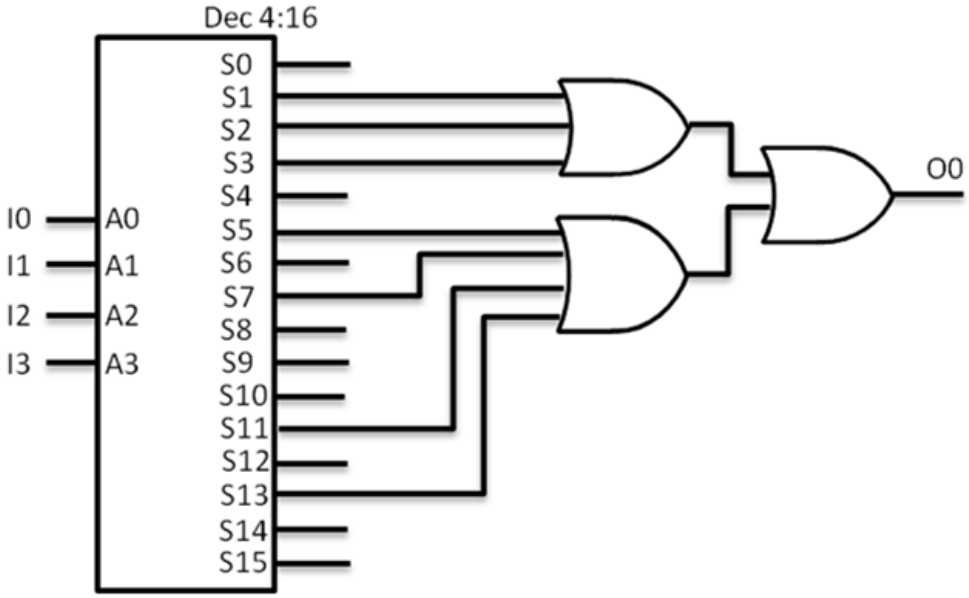


OTRA POSIBLE SOLUCIÓN:

I3	I2	I1	O1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



f)



Ejercicio 13.

a)

d	c	b	a	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

b)

e	y	X0	X1	X2	X3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$X0 = \bar{e} \cdot \bar{y}$

$X1 = \bar{e} \cdot y$

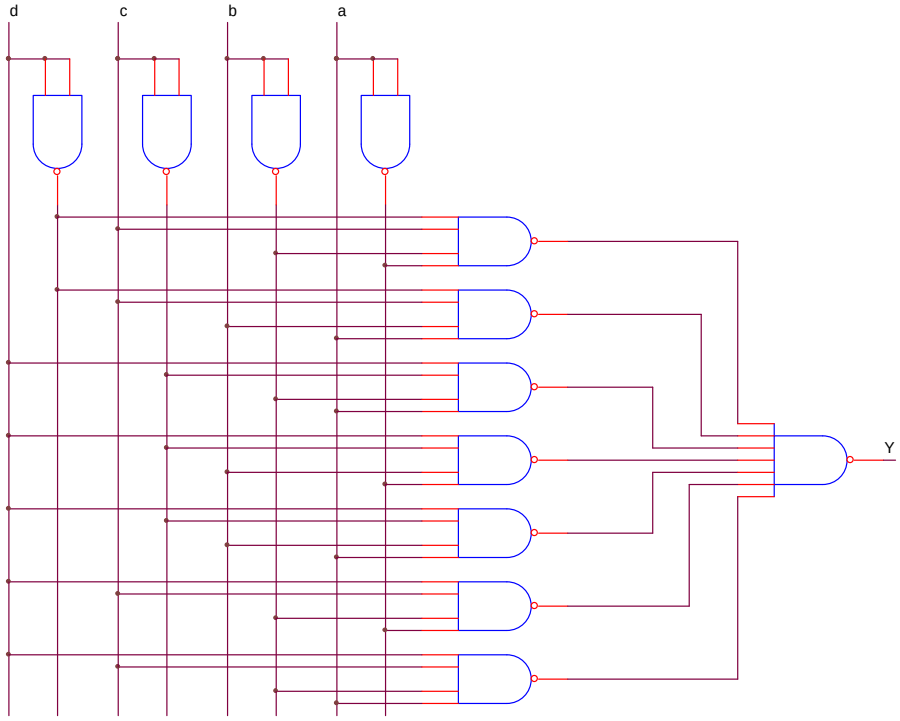
$X2 = e \cdot \bar{y}$

$X3 = e \cdot y$

c)

e	y	g	f	X0	X1	X2	X3	S
0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	0
0	0	1	0	1	0	0	0	0
0	0	1	1	1	0	0	0	0
0	1	0	0	0	1	0	0	0
0	1	0	1	0	1	0	0	1
0	1	1	0	0	1	0	0	0
0	1	1	1	0	1	0	0	0
1	0	0	0	0	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	1	0	0	0	1	0	1
1	0	1	1	0	0	1	0	0
1	1	0	0	0	0	0	1	0
1	1	0	1	0	0	0	1	0
1	1	1	0	0	0	0	1	0
1	1	1	1	0	0	0	1	1

d) $y = \bar{d}\cdot c\cdot\bar{b}\cdot\bar{a} + \bar{d}\cdot c\cdot b\cdot a + d\cdot\bar{c}\cdot\bar{b}\cdot a + d\cdot c\cdot\bar{b}\cdot\bar{a} + d\cdot\bar{c}\cdot b\cdot a + d\cdot c\cdot\bar{b}\cdot\bar{a} + d\cdot c\cdot\bar{b}\cdot a$

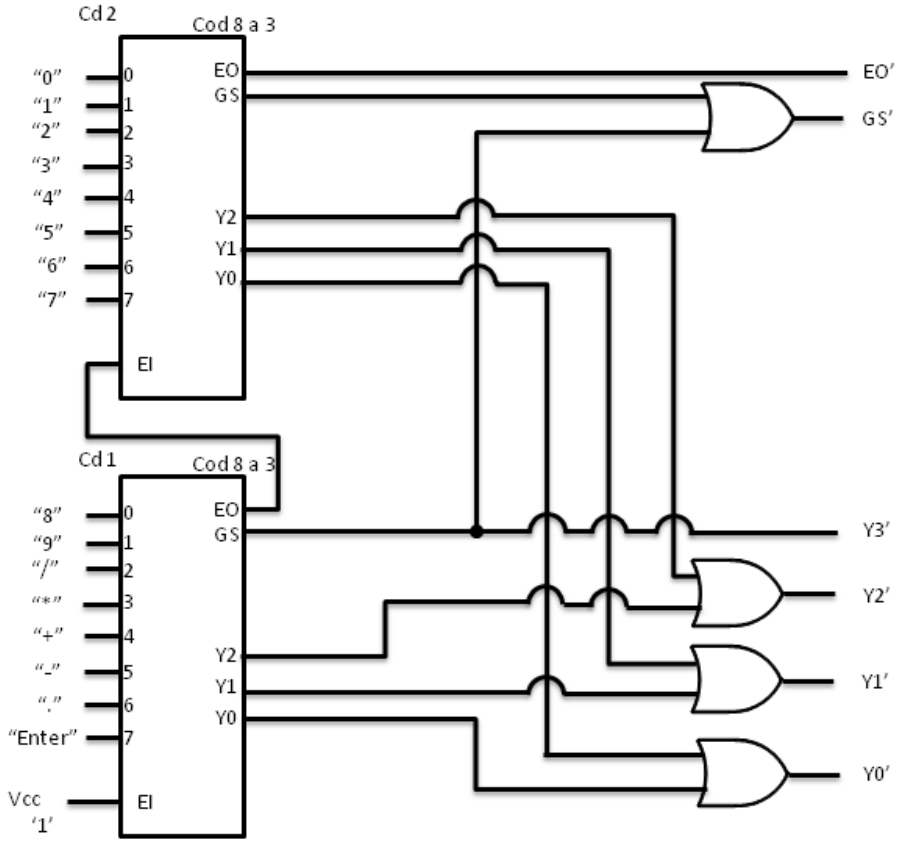


e)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Ejercicio_14 is
   port(g,f,e,d,c,b,a : in std_logic;
                    S : out std_logic);
end Ejercicio_14;

architecture Behavioral of Ejercicio_14 is

signal y: std_logic;
begin
   y<= c and (a xnor b) when (d='0') else ((not c) and (b or a))
        or (c and (not b));

   S<=(not f) and (e xnor g) when (y='0') else f and (e xnor g);

end Behavioral;
```

Ejercicio 14.

- **Si GS'='0' no hay ninguna tecla pulsada.**
- **Si GS'='1' hay alguna tecla pulsada.**



Ejercicio 15.

a)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Comparador is
      port(A,B : in std_logic_vector (3 downto 0);
      AmayorB : out std_logic);
end Comparador;

architecture Behavioral of Comparador is
begin
      AmayorB <='1' when A>B else '0';
end Behavioral;
```

b)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Mux2x4 is
      port(D0,D1 : in std_logic_vector (3 downto 0);
            S : in std_logic;
            Y: out std_logic_vector (3 downto 0));
      end Mux2x4;

architecture Behavioral of Mux2x4 is
begin
      Y<=D1 when S ='1' else D0;
end Behavioral;
```

c)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Convertidor_BCD_7seg is
      port( N : in std_logic_vector (3 downto 0);
      display: out STD_LOGIC_VECTOR (6 downto 0));
end Convertidor_BCD_7seg;

architecture Behavioral of Convertidor_BCD_7seg is
begin
      with N select
              display<="1111110" when "0000",
                       "0110000" when "0001",
                       "1101101" when "0010",
                       "1111001" when "0011",
                       "0110011" when "0100",
                       "1011011" when "0101",
                       "1011111" when "0110",
                       "1110000" when "0111",
                       "1111111" when "1000",
                       "1111011" when "1001",
                       "-------" when others;
end Behavioral;
```

d)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ejercicio16 is
      Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
             B : in STD_LOGIC_VECTOR (3 downto 0);
             C : in STD_LOGIC_VECTOR (3 downto 0);
             display : out STD_LOGIC_VECTOR (6 downto 0));
end ejercicio16;

architecture Behavioral of ejercicio1 is

Component Comparador
      port(A,B : in std_logic_vector (3 downto 0);
      AmayorB : out std_logic);
End Component;

Component Mux2x4
      port(D0,D1 : in std_logic_vector (3 downto 0);
            S : in std_logic;
            Y: out std_logic_vector (3 downto 0));
End Component;

Component Convertidor_BCD_7seg
      port( N : in std_logic_vector (3 downto 0);
      display: out STD_LOGIC_VECTOR (6 downto 0));
End Component;
signal C1 : std_logic;
signal C2 : std_logic;
signal R1 : std_logic_vector (3 downto 0);
signal R2 : std_logic_vector (3 downto 0);

begin
U1: Comparador port map (A=>A,B=>B,AmayorB=>C1);
U2: Mux2x4 port map (D0=>B,D1=>A,S=>C1,Y=>R1);
U3: Comparador port map (A=>R1,B=>C,AmayorB=>C2);
U4: Mux2x4 port map (D0=>C,D1=>R1,S=>C2,Y=>R2);
U5: Convertidor_BCD_7seg port map (N=>R2,display=>display);
end Behavioral;
```