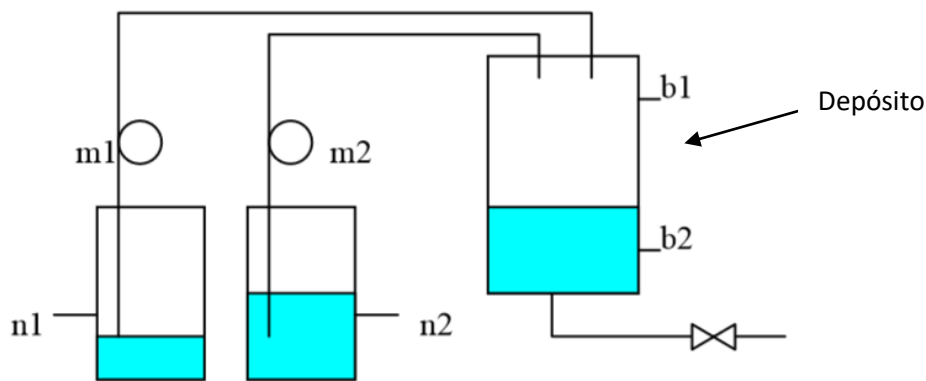


Problema 1 (3.5 puntos)

Mediante dos bombas (m1 y m2) se controla el nivel de un depósito. El depósito tiene dos boyas (b1 y b2). Cuando el nivel está por debajo de la boya el contacto correspondiente está abierto (0 lógico). Las bombas sacan agua de dos pozos. Para controlar esto, cada pozo tiene una boya (n1, n2).

El sistema funciona de la siguiente forma:

- Si no hay agua en el pozo las bombas no funcionan.
- Si el nivel del depósito supera la boya b1, las bombas están paradas.
- Si el nivel del depósito está entre la boya b1 y la b2, funciona la bomba m1, si hay agua suficiente en el pozo 1. Si no hay agua en el pozo 1 pero la hay en el 2, funciona la bomba m2.
- Si el nivel del depósito está por debajo de la boya b2, se activa la bomba m2, además de la m1.



Nota: En los casos en que las boyas (b1 y b2) generen combinaciones imposibles, las bombas (m1 y m2) se pararán.

- a. Describa en VHDL la **entidad** del circuito de control del sistema. (0.5p)

```
entity llen_depos is
  port (
    b1, b2, n1, n2: in    std_logic;
    m1, m2:         out   std_logic
  );
end llen_depos;
```

b. Rellene la tabla de verdad del sistema (1p)

b1	b2	n1	n2	m1	m2
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

c. Determine la primera forma canónica de m1 y la segunda de m2. (0.5p)

$$m1 = \sum_4(2,3,6,7) = \overline{b1} \cdot \overline{b2} \cdot n1 \cdot n2 + \overline{b1} \cdot \overline{b2} \cdot n1 \cdot \overline{n2} + \overline{b1} \cdot b2 \cdot n1 \cdot n2 + \overline{b1} \cdot b2 \cdot n1 \cdot \overline{n2}$$

$$m2 = \prod_4(0,2,4,6,7,8,9,10,11,12,13,14,15) = (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2) \cdot (b1+b2+n1+n2)$$

- d. Describa en VHDL la **arquitectura** que implementa la funcionalidad del sistema.(1.5p)

Se admiten varias soluciones tanto con sentencias concurrentes como secuenciales dentro de un proceso, una de ellas sería la siguiente.

```
architecture control of llen_depos is
    -- Agrupamos las entradas en una sola señal
    signal temp: std_logic_vector(3 downto 0);
begin

    temp <= b1 & b2 & n1 & n2;

    m1 <= '1' when temp="0010" or temp="0011" or temp="0110" or temp="0111" else
    '0';

    m2 <= '1' when temp="0001" or temp="0011" or temp="0101" else
    '0';

end control;
```

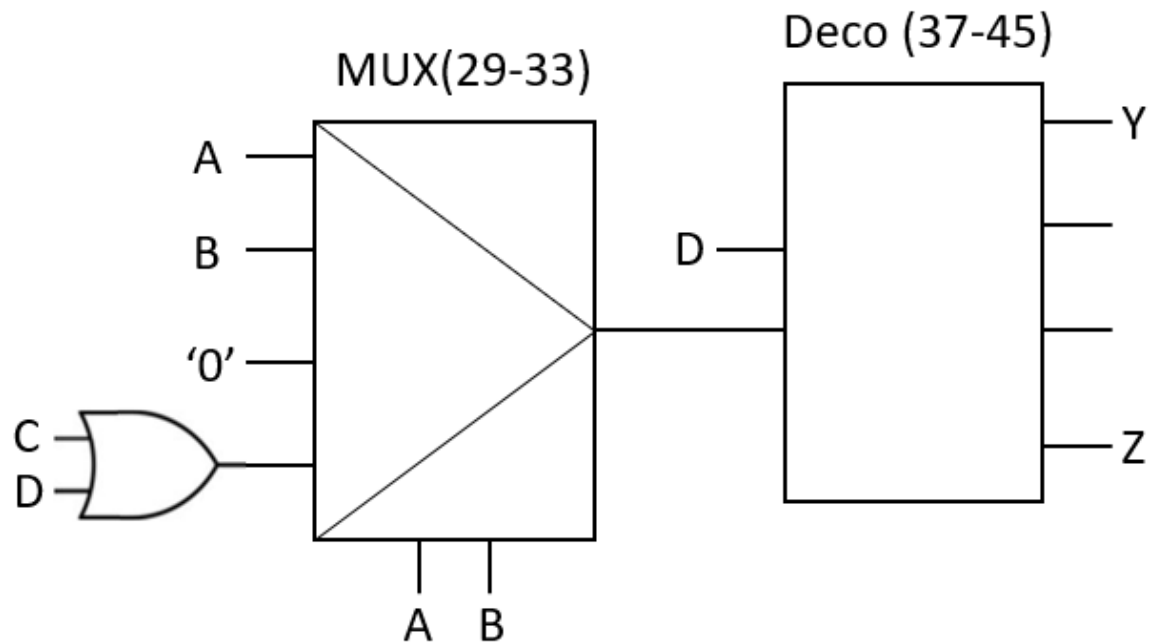
Problema 2 (3 puntos)

Dado el siguiente código VHDL, obtenga el esquemático equivalente. Nombre los bloques obtenidos y diga que líneas del código implementan dichos bloques. (2p)

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  ENTITY exam IS
5      PORT
6      (
7          A : IN  STD_LOGIC;
8          B : IN  STD_LOGIC;
9          C : IN  STD_LOGIC;
10         D : IN  STD_LOGIC;
11         Y : OUT  STD_LOGIC;
12         Z : OUT  STD_LOGIC
13     );
14  END exam;
15
16  ARCHITECTURE bdf_type OF exam IS
17
18      SIGNAL AUX1,AUX2: STD_LOGIC;
19      SIGNAL S,T:STD_LOGIC_VECTOR (1 downto 0);
20      SIGNAL V:STD_LOGIC_VECTOR (3 downto 0);
21
22  BEGIN
23
24      S<=A & B;
25      AUX1<= C OR D;
26
27
28      WITH S SELECT
29      AUX2 <= A    WHEN "00",
30             B    WHEN "01",
31             '0'  WHEN "10",
32             AUX1 WHEN OTHERS;
33
34      T <=D & AUX2;
35
36  PROCESS (T)
37  BEGIN
38      CASE T IS
39      WHEN "00" => V <= "0001";
40      WHEN "01" => V <= "0010";
41      WHEN "10" => V <= "0100";
42      WHEN "11" => V <= "1000";
43      END CASE;
44  END PROCESS;
45
46      Y<=V(0) ;
47      Z<=V(3) ;
48  END bdf_type;

```

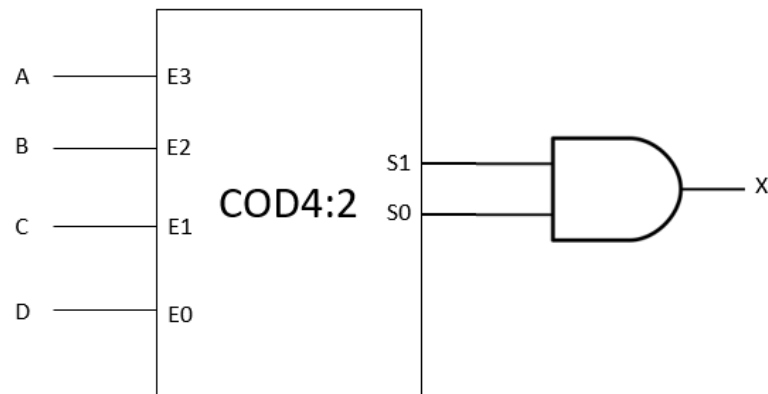


Rellene el siguiente cronograma (AUX2,Y,Z) para el código **anterior** (1p)

A	A 0			
B	A 0			
C	A 0			
D	A 0			
AUX2	A 0			
Y	A 1			
Z	A 0			

Problema 3 (1 punto)

Implemente en un proceso el siguiente circuito.



```

ARCHITECTURE circuito OF problema3 IS
BEGIN

PROCESS (A,B,C,D)
BEGIN
IF (D='0' AND C='0' AND B='0' AND A='1') THEN
X<='1';
ELSE
X<='0';
END IF;
END PROCESS;
END circuito;
  
```

Problema 4 (2.5 punto)

4.1) Convierta la siguientes cantidades decimales a binario, Octal y Hexadecimal(0.5p)

Decimal	Binario	Octal	Hexadecimal
256,125			

4.2) Si A= -17 y B =20 están expresados en base decimal, (1.5p)

- a) Represente A,B,-A,-B, en Complemento a dos con 6 bits
- b) Realice las operaciones A+B , A-B y B-A, con ambos números expresados en la notación Complemento a 2 y señale justificadamente los casos en que se produce desbordamiento (overflow).

4.3) Dado el siguiente código VHDL: (0.5p)

```

signal A,B,SUM: unsigned(7 downto 0);
signal Cy: std_logic;
signal result: unsigned(8 downto 0);
A <= "11010010";
B <= "10101110";
result <= resize(a,9)+resize(b,9);
Cy <= result(8);
SUM<= result(7 downto 0);

```

Obtenga el valor de Cy y de SUM

Solución4.1) 10000000,001₂, 100,2₁₆, 400₈

4.2) 4.2.a) A= -17, +17 → 010001 , pasando a Ca 2 : -17= 101111

-A=17 → 010001 ,, B= 20→ 010100 , -B → 101100

4.2.b) A+B→ 101111

+

010100

1)000011 => +3 Luego No hay Overflow

-A+B→010001 + 010100= 100101 => Overflow, El signo debería ser "+" y sale "-"

4.3) SUM= 10000000 y Cy= 1