



**Nombre:** \_\_\_\_\_

**Grupo:** \_\_\_\_\_

**Apellidos:** \_\_\_\_\_

**Problema 1 (1.75 puntos)**

Dadas las funciones lógicas:

$$f_1 = \sum_4 (2,5,8,10,11,14,15) + \Delta_4 (0,9,13)$$

$$f_2 = ac + cb + ab\bar{c}d$$

se pide:

- a) Implementar  $f_1$  con el mínimo número posible de puertas NAND.
- b) Obtener una expresión simplificada de  $f_2$  en forma de producto de sumas.
- c) Realizar  $f_1$  con un multiplexor de 16 entradas de datos.
- d) Realizar  $f_1$  con un multiplexor de 8 entradas de datos e inversores.

**Nota importante:** se valorará el uso del menor número de componentes en las soluciones.

**Cuestión 1 (0.75 punto)**

- a) Convertir  $499_{10}$  a binario natural, octal y hexadecimal. Determinar razonadamente el número de bits necesarios para expresar números enteros del rango -499 a 499.
- b) Convertir  $F5_{16}$  a BCD.
- c) Expresar  $+27_{10}$  en binario natural y  $+27_{10}$  y  $-27_{10}$  en convenio de complemento a 1 y complemento a 2. Elegir el número de bits apropiado.
- d) Realizar las siguientes operaciones como sumas de 9 bits, con los operandos expresados en complemento a 2:  $27+245$ ,  $-27+245$ . Razonar en qué casos hay acarreo y/o desbordamiento. ( $245_{10}=11110101_2$ ).



$$f_1 = \sum (2, 5, 8, 10, 11, 14, 15) + \Delta (10, 9, 13)$$

a) Primeros 8 valores de  $f_1$

a \ b	00	01	10	11
00	0	0	0	1
01	0	1	0	0
10	0	0	1	1
11	1	0	1	1

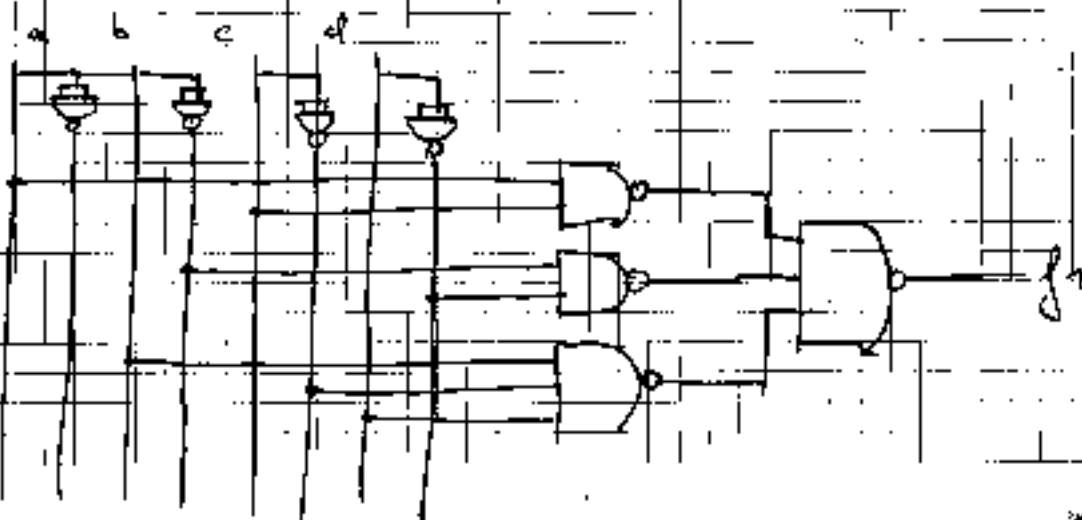
a \ b \ c \ d	d	max
0000	X	0
0001	0	0
0010	1	0
0011	0	0
0100	0	0
0101	1	0
0110	0	0
0111	0	0
1000	1	1
1001	X	1
1010	1	1
1011	1	1
1100	0	0
1101	X	0
1110	1	1
1111	1	1

$$f_1 = ac + bd + b\bar{c}d$$

Para implementar la función solo con puertas NAND es necesario q. aparezca como producto:

$$f_1 = ac + bd + b\bar{c}d$$

$$ac = \bar{b}\bar{d} + b\bar{c}d$$



b)  $f_2 = a + cb + abcd$

abcd	$f_2$
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	1
0111	1
1000	0
1001	0
1010	1
1011	1
1100	0
1101	1
1110	1
1111	1

ab \ cd	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	1	1	1

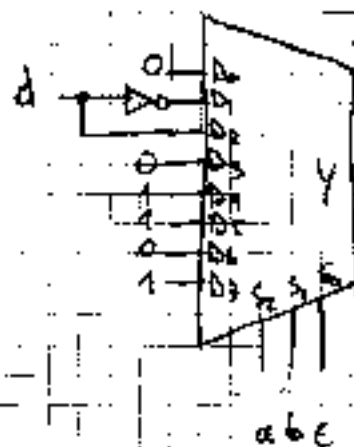
$$f = (a+b)(c+d)(a+c)(b+c)$$

c)



a b c d

d)



En la tabla de verdad del apdo a) está la columna "aux 0"

Directamente en las entradas de datos se coloca la columna  $f_2$  de la tabla de verdad. Los términos andi prestes se forman partiendo como 0 (en este caso).

En las entradas de selección se conectan las variables lógicas en el mismo orden que tienen en la tabla de verdad.



**Cuestión 1 (0.75 punto)**

**a) Convertir  $499_{10}$  a binario natural, octal y hexadecimal. Determinar razonadamente el número de bits necesarios para expresar números enteros del rango -499 a 499.**

a.1)  $499_{10} \rightarrow 111110011_2$   
 $499_{10} \rightarrow 763_8$   
 $499_{10} \rightarrow 1F3_{16}$

a.2) Con 9 bits se puede representar  $2^9 = 512$  números. Por lo tanto, se necesitan 9 bits para los números y 1 bit para el signo. En total hacen falta 10 bits para representar el rango -499  $\rightarrow$  499.

**b) Convertir  $F5_{16}$  a BCD.**

$F5_{16} \rightarrow 245_{10} \rightarrow 0010\ 0100\ 0101_{BCD}$

**c) Expresar  $+27_{10}$  en binario natural y  $+27_{10}$  y  $-27_{10}$  en convenio de complemento a 1 y complemento a 2. Elegir el número de bits apropiado.**

c.1)  $+27_{10} \rightarrow 11011_{BIN}$

c.2.a)  $+27_{10} \rightarrow 011011_{Ca1}$   
 $-27_{10} \rightarrow 100100_{Ca1}$

c.2.b)  $+27_{10} \rightarrow 011011_{Ca2}$   
 $-27_{10} \rightarrow 100101_{Ca2}$

**d) Realizar las siguientes operaciones como sumas de 9 bits, con los operandos expresados en complemento a 2:  $27+245$ ,  $-27+245$ . Razonar en qué casos hay acarreo y/o desbordamiento. ( $245_{10} = 11110101_2$ ).**

d.1)  $+27_{10} \rightarrow 000011011$   
 $+245_{10} \rightarrow 011110101_{Ca2}$   
-----  
 $+272_{10} \rightarrow 100010000_{Ca2}$

No hay acarreo. El resultado sale negativo cuando los operandos son positivos, por tanto, hay desbordamiento. Es debido a que con 8 bits no se puede representar en binario la cifra +272.

d.2)  $-27_{10} \rightarrow 111100101_{Ca2}$   
 $+245_{10} \rightarrow 011110101_{Ca2}$   
-----  
 $+218_{10} \rightarrow 011011010_{Ca2}$

No hay acarreo ni desbordamiento. Sumando dos números de distinto signo no puede haber desbordamiento.