Universidad Carlos III de Madrid
Grado en Ingeniería Informática. Grupos 84,85,86
Tecnología de Computadores. 1er parcial. Octubre de 2009

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).

Asignatura TECNOLOGÍA DE COMPUTADORES

Nombre del Alumno

Fecha Curso Grupo

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

a) Primero se simplificará f_1:

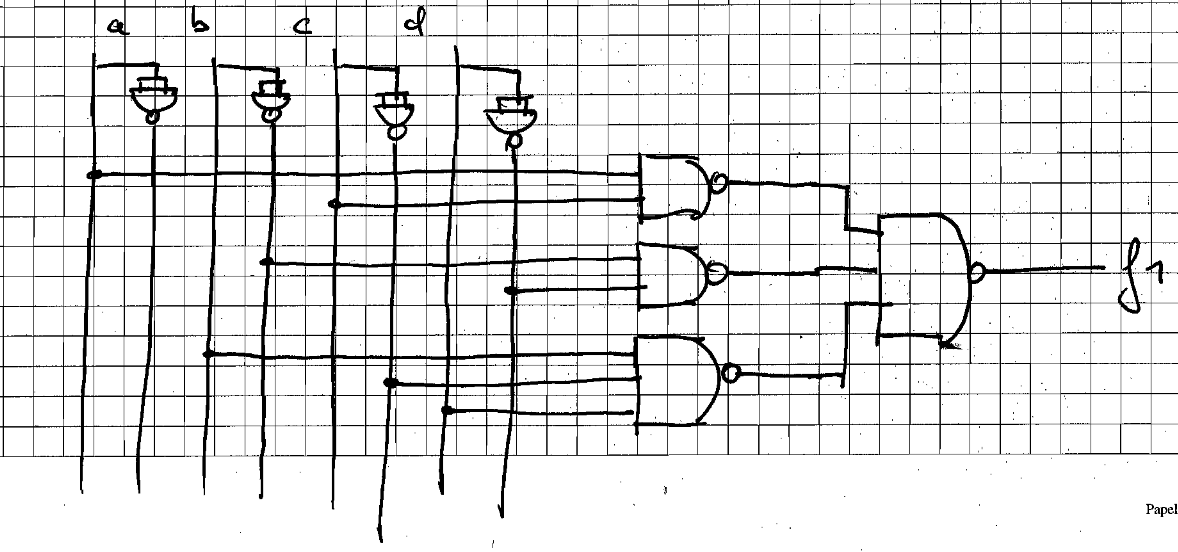
cd \ ab	00	01	11	10
00	X	0	0	1
01	0	1	X	X
11	0	0	1	1
10	1	0	1	1

abcd	f	MUX 8
0000	X	0
0001	0	
0010	1	$\bar{d}$
0011	0	
0100	0	d
0101	1	
0110	0	0
0111	0	
1000	1	1
1001	X	
1010	1	1
1011	1	
1100	0	0
1101	X	
1110	1	1
1111	1	

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

Para implementar la función sólo con puertas NAND es necesario q. aparezca sólo productos:

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



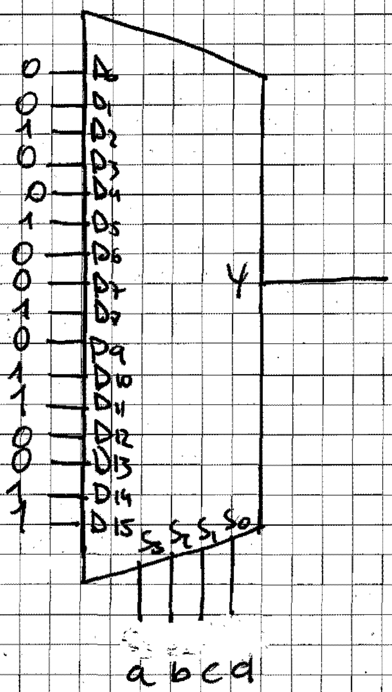
b) $f_2 = ac + cb + a\bar{b}\bar{c}d$

a b c d	f_2
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	1
0 1 1 1	1
1 0 0 0	0
1 0 0 1	0
1 0 1 0	1
1 0 1 1	1
1 1 0 0	0
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

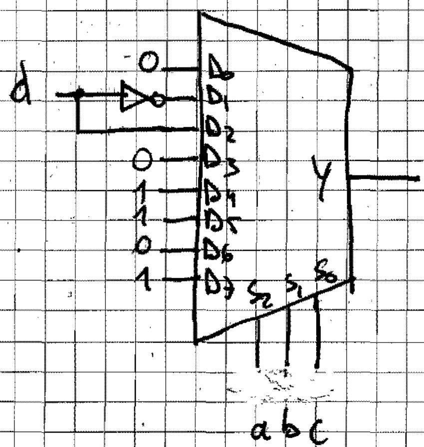
cd \ ab	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	1	1	1

$$\boxed{f = (a+b)\,(c+d)\,(a+c)(b+c)}$$

c)



d)



En la tabla de verdad del apto a) está la columna "MUX 8"

Directamente en las entradas de datos se coloca la columna "f_1" de la tabla de verdad. Los términos indiferentes se toman arbitrariamente como 0 (en este caso)

En las entradas de selección se conectan las variables lógicas en el mismo orden q. tienen el 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.