Universitat de GIRONA
Estudis d'ETIS / ETIG **Curs: 1^er^** **2005/2006** **8 de Novembre de 2005**
Estructura i Tecnologia de Computadors **Avaluació personal (Temes 1 - 3)** **Grup de Tarda**

Nom i cognoms:..

PROVA DE PROBLEMES
Temps: 1h. La puntuació està posada al costat de cada exercici. La prova puntua un total d'un punt. La revisió de l'examen es farà avui mateix després de l'examen.

1.- L'ORDENADOR. *(3 Punts)*

a. Ordenar de més petit a més gran els següents números naturals. **(1 Punt)**

$3F2_h$, 010111_b, 818_d, 127_8 i 11010_{10}

Per poder-los comparar cal tenir-los tots en el mateixa representació. Una bona representació per ordenar-los és base 10 per tant, es convertiran tots a aquesta base:

$3F2_h$

Pes	**2**	**1**	**0**
Dígit	3	F	2

$$N_d = 3 * 16^2 + F * 16^1 + 2 * 16^0 = 3 * 256 + 15 * 16 + 2 * 1 = 768 + 240 + 2 = \mathbf{1010_d}$$

010111_b

Pes	**5**	**4**	**3**	**2**	**1**	**0**
Dígit	0	1	0	1	1	1

$$N_d = 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 2^4 + 2^2 + 2^1 + 2^0 = 16 + 4 + 2 + 1 = \mathbf{23_d}$$

818_d

$$N_d = \mathbf{818_d}$$

127_8

Pes	**2**	**1**	**0**
Dígit	1	2	7

$$N_d = 1 * 8^2 + 2 * 8^1 + 7 * 8^0 = 64 + 16 + 7 = \mathbf{87_d}$$

11010_{10}

$$N_d = \mathbf{11010_d}$$

L'ordenació serà la següent:

$$\mathbf{010111_b < 127_8 < 818_d < 3F2_h < 11010_{10}}$$

$$\mathbf{(23_d < 87_d < 818_d < 1010_d < 11010_d)}$$

b. Ordenar de més petit a més gran els següents números enters de 8 bits codificats en C_2. **(1 Punt)**
FF_h, 125_d, 11111111_b, 111_8 i 1011111_b

Per poder-los interpretar correctament, primer cal obtenir la seva representació en binari per, posteriorment, calcular quin és el seu valor tenint en compte que es troben codificats amb 8 bits en complement a 2:

FF_h

Conversió a binari:

Hexadecimal	**F**	**F**
Binari	1111	1111

Valor en 8 bits en C_2:

Pes	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Dígit	*1*	1	1	1	1	1	1	1

El bit 7 indica que és negatiu per tant, cal fer el complement a 2:

$$C_2(1111111_b) = C_1(1111111_b) + 1 = 0000000_b + 1 = 0000001_b$$

N = -1

125_d:

Conversió a binari:

Dividend	Divisor	Quocient	Residu
125	2	62	**1**
62	2	31	**0**
31	2	15	**1**
15	2	7	**1**
7	2	3	**1**
3	2	1	**1**
1	2	-	**1**

Valor en 8 bits en C_2:

Pes	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Dígit	*0*	1	1	1	1	1	0	1

En aquest cas, el bit 7 indica que és positiu per tant, el valor és la interpretació del valor binari:

$$1111101_b = 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 =$$
$$= 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 64 + 32 + 16 + 8 + 4 + 1 = 125_d$$

N = +125

11111110_b:
Valor en 8 bits en C_2:

Pes	7	6	5	4	3	2	1	0
Dígit	*1*	1	1	1	1	1	1	0

El bit 7 indica que és negatiu per tant, cal fer el complement a 2:

$$C_2(1111110_b) = C_1(1111110_b) + 1 = 0000001_b + 1 = 0000010_b$$

N = -2

111_8:
Conversió a binari:

Octal	1	1	1
Binari	001	001	001

Valor amb 8 en C_2:

Pes	7	6	5	4	3	2	1	0
Dígit	*0*	1	0	0	1	0	0	1

En aquest cas, el bit 7 indica que és positiu per tant, el valor és la interpretació del valor binari:

$$1001001_b = 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 =$$
$$= 2^6 + 2^3 + 2^0 = 64 + 8 + 1 = 75_d$$

N = +75

1011111_b:
Valor amb 8 en C_2:

Pes	7	6	5	4	3	2	1	0
Dígit	*0*	1	0	1	1	1	1	1

En aquest cas, el bit 7 indica que és positiu per tant, el valor és la interpretació del valor binari:

$$1011111_b = 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 =$$
$$= 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 64 + 16 + 8 + 4 + 2 + 1 = 95$$

N = +95

Per tant l'ordenació serà la següent:

$$\mathbf{11111110_b < FFh < 111_8 < 1011111_b < 125_d}$$

(-2 < -1 < 75 < 95 < 125)

c. Ordenar de més petit a més gran els següents reals <u>codificats en format de coma flotant</u> IEEE 754 de 32 bits. **(1Punt)**

17512301211_8 i $8BFC0990_h$

Cal tenir en compte la representació en forma de **Signe (1 bit) + Exponent (8 bits) + Mantissa (23 bits)** d'aquesta codificació:

17512301211_8:

Octal	**1**	**7**	**5**	**1**	**2**	**3**	**0**	**1**	**2**	**1**	**1**
Binari	001	111	101	001	010	011	000	001	010	001	001

Agafant els 32 primers bits:

	S	Exponent								**Mantissa**																						
Pes	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Dígit	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**

$8BFC0990_h$:

Hexadecimal	**8**	**B**	**F**	**C**	**0**	**9**	**9**	**0**
Binari	1000	1011	1111	1100	0000	1001	1001	0000

Agafant els 32 primers bits:

	S	Exponent								**Mantissa**																						
Pes	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Dígit	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

El número octal té el bit de signe a 0 (positiu) i l'hexadecimal té el bit de signe a 1 (negatiu) per tant, no cal interpretar els valors per ordenar-los:

$8BFC0990_h < 17512301211_8$

NOTA: Cal justificar les ordenacions.

2.- SIMPLIFICADOR. ***(3 Punts)***

Simplifica algebraicament les següents funcions:

a. $Z = f(a, b, c) = a \cdot b' + a \cdot (b \cdot c' + b' \cdot c + a)'$ **(1 Punt)**

Expressió algebraica	Propietat que s'aplica
$Z = f(a, b, c) = a \cdot b' + a \cdot (b \cdot c' + b' \cdot c + a)'$	DeMorgan
$a \cdot b' + a \cdot ((b \cdot c')' \cdot (b' \cdot c)' \cdot a')$	DeMorgan
$a \cdot b' + a \cdot ((b' + c'') \cdot (b'' + c') \cdot a')$	Eliminació de parèntesi
$a \cdot b' + a \cdot (b' + c'') \cdot (b'' + c') \cdot a'$	Commutativa
$a \cdot b' + a \cdot a' \cdot (b' + c'') \cdot (b'' + c')$	$X \cdot X' = 0$
$a \cdot b' + 0 \cdot (b' + c'') \cdot (b'' + c')$	$0 \cdot X = 0$
$a \cdot b' + 0$	$X + 0 = X$
$\mathbf{a \cdot b'}$	

b. $I = f(x, y, z) = (x' \cdot y + y' \cdot z + z' \cdot x + x' \cdot y \cdot z')'$ **(1 Punt)**

Hi ha més d'una possible simplificació vàlida ja que és difícil decidir quina és màxima:

Expressió algebraica	Propietat que s'aplica
$I = f(x, y, z) = (x' \cdot y + y' \cdot z + z' \cdot x + x' \cdot y \cdot z')'$	DeMorgan
$(x' \cdot y)' \cdot (y' \cdot z)' \cdot (z' \cdot x)' \cdot (x' \cdot y \cdot z')'$	DeMorgan
$(x'' + y') \cdot (y'' + z') \cdot (z'' + x') \cdot (x'' + y' + z'')$	$X'' = X$
$(x + y') \cdot (y + z') \cdot (z + x') \cdot (x + y' + z)$	Commutativa
$(x + y') \cdot (x + y' + z) \cdot (y + z') \cdot (z + x')$	$X \cdot (X + Y) = X$
$\mathbf{(x + y') \cdot (y + z') \cdot (z + x')}$	

Una altra simplificació possible (la que dóna simplificant per *Karnaugh*):

Expressió algebraica	Propietat que s'aplica
$I = f(x, y, z) = (x' \cdot y + y' \cdot z + z' \cdot x + x' \cdot y \cdot z')'$	DeMorgan
$(x' \cdot y)' \cdot (y' \cdot z)' \cdot (z' \cdot x)' \cdot (x' \cdot y \cdot z')'$	DeMorgan
$(x'' + y') \cdot (y'' + z') \cdot (z'' + x') \cdot (x'' + y' + z'')$	$X'' = X$
$(x + y') \cdot (y + z') \cdot (z + x') \cdot (x + y' + z)$	Distributiva del Producte Respecte la Suma
$(x \cdot y + x \cdot z' + y' \cdot y + y' \cdot z') \cdot (z + x') \cdot (x + y' + z)$	$X \cdot X' = 0$
$(x \cdot y + x \cdot z' + y' \cdot z') \cdot (z + x') \cdot (x + y' + z)$	Distributiva del Producte Respecte la Suma
$(x \cdot y \cdot (z + x') + x \cdot z' \cdot (z + x') + y' \cdot z' \cdot (z + x')) \cdot$ $\cdot (x + y' + z)$	Distributiva del Producte Respecte la Suma
$(x \cdot y \cdot z + x \cdot y \cdot x' + x \cdot z' \cdot z + x \cdot z' \cdot x' +$ $+ y' \cdot z' \cdot z + y' \cdot z' \cdot x') \cdot (x + y' + z')$	$X \cdot X' = 0$ i $X + 0 = X$
$(x \cdot y \cdot z + y' \cdot z' \cdot x') \cdot (x + y' + z')$	Distributiva del Producte Respecte la Suma
$x \cdot y \cdot z \cdot x + y' \cdot z' \cdot x' \cdot x + x \cdot y \cdot z \cdot y' +$ $+ y' \cdot z' \cdot x' \cdot y' + x \cdot y \cdot z \cdot z' + y' \cdot z' \cdot x' \cdot z'$	$X \cdot X' = 0$ i $X \cdot X = X$
$x \cdot y \cdot z + 0 + 0 + x' \cdot y' \cdot z' + 0 + x' \cdot y' \cdot z'$	$X + 0 = X$ i $X + X = X$
$\mathbf{x \cdot y \cdot z + x' \cdot y' \cdot z'}$	

Simplifica al màxim i representa amb portes NOR la següent funció:

c. $M = f(a, b, c) = a \cdot b + a \cdot b' \cdot c' + a' \cdot b \cdot c$ **(1 Punt)**

Per simplificar al màxim, es pot construir la taula de veritat i utilitzar *Karnaugh*:

Mínterm	Binari a b c	a · b		a · b' · c'		a' · b · c		M = f(a, b, c) a · b + a · b' · c' + a' · b · c
0	000	**000**	$0 \cdot 0 = 0$	**000**	$0 \cdot 1 \cdot 1 = 0$	**000**	$1 \cdot 0 \cdot 0 = 0$	$0 + 0 + 0 = 0$
1	001	**001**	$0 \cdot 0 = 0$	**001**	$0 \cdot 1 \cdot 0 = 0$	**001**	$1 \cdot 0 \cdot 1 = 0$	$0 + 0 + 0 = 0$
2	010	**010**	$0 \cdot 1 = 0$	**010**	$0 \cdot 0 \cdot 1 = 0$	**010**	$1 \cdot 1 \cdot 0 = 0$	$0 + 0 + 0 = 0$
3	011	**011**	$0 \cdot 1 = 0$	**011**	$0 \cdot 0 \cdot 0 = 0$	**011**	$1 \cdot 1 \cdot 1 = 1$	$0 + 0 + 1 = 1$
4	100	**100**	$1 \cdot 0 = 0$	**100**	$1 \cdot 1 \cdot 1 = 1$	**100**	$0 \cdot 0 \cdot 0 = 0$	$0 + 1 + 0 = 1$
5	101	**101**	$1 \cdot 0 = 0$	**101**	$1 \cdot 1 \cdot 0 = 0$	**101**	$0 \cdot 0 \cdot 1 = 0$	$0 + 0 + 0 = 0$
6	110	**110**	$1 \cdot 1 = 1$	**110**	$1 \cdot 0 \cdot 1 = 0$	**110**	$0 \cdot 1 \cdot 0 = 0$	$1 + 0 + 0 = 1$
7	111	**111**	$1 \cdot 1 = 1$	**111**	$1 \cdot 0 \cdot 0 = 0$	**111**	$0 \cdot 1 \cdot 1 = 0$	$1 + 0 + 0 = 1$

Simplificant la funció per *Karnaugh*:

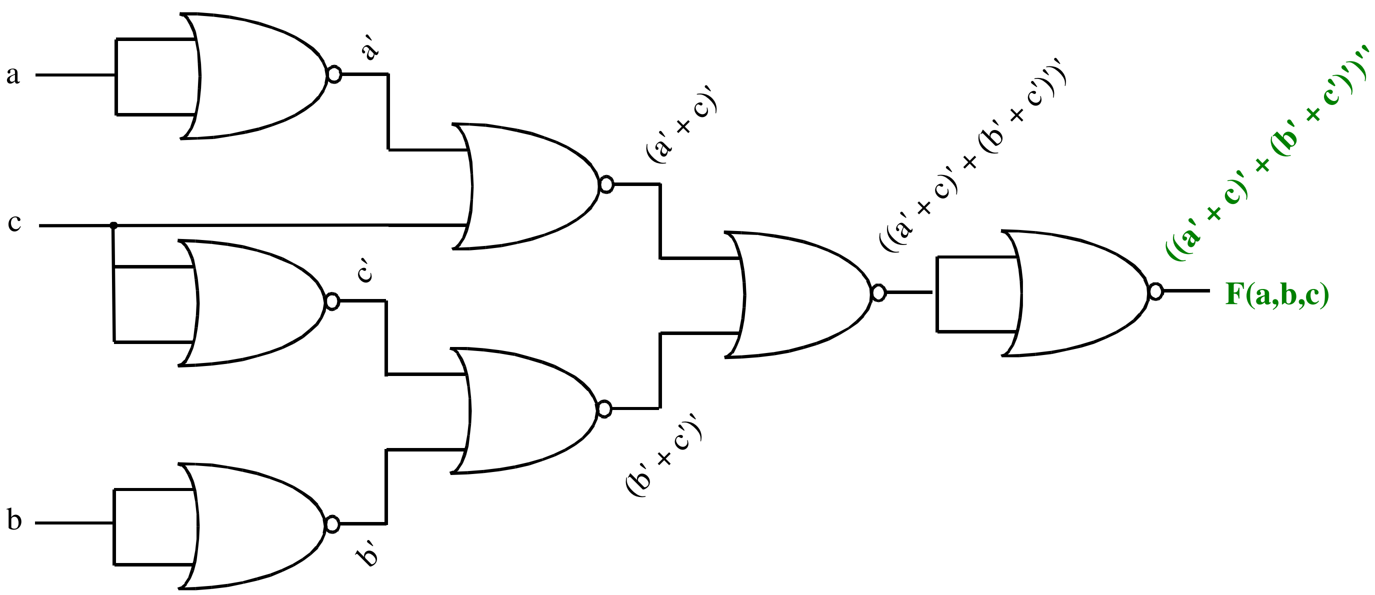
c \ ab	*00*	*01*	*11*	*10*
0	0	2	**1** 6	**1** 4
1	1	**1** 3	**1** 7	5

$$F(a, b, c) = a \cdot c' + b \cdot c$$

Conversió a portes NOR:

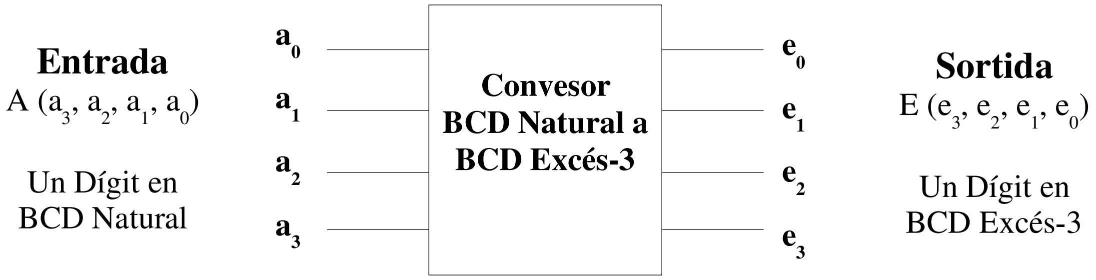
$$F(a, b, c) = a \cdot c' + b \cdot c =$$
$$F(a, b, c) = (a \cdot c')'' + (b \cdot c)'' = (a' + c'')' + (b' + c')' = (a' + c)' + (b' + c')'$$
$$= ((a' + c)' + (b' + c')')''$$

Representació amb portes:



3.- CONVERSOR BCD Natural a **BCD Excés-3.** ***(4 Punts)***

Dissenyar el conversor d'un dígit de codi BCD Natural a un dígit de codi BCD Excés-3 que tindrà la forma següent:



a. Fer la taula de veritat de les 4 funcions de sortida. **(2 Punts)**

Mínterm	**BCD – Natural** $a_3\ a_2\ a_1\ a_0$	**BCD Excés-3** $e_3\ e_2\ e_1\ e_0$
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
10	xxxx	xxxx
11	xxxx	xxxx
12	xxxx	xxxx
13	xxxx	xxxx
14	xxxx	xxxx
15	xxxx	xxxx

b. Expressar les 4 funcions en Suma de Mínterms i Producte de Màxterms numèricament. **(1 Punt)**

En forma de Suma de Mínterms (tenint en compte els termes *don't care*):

$\mathbf{e_3(a_3, a_2, a_1, a_0)} = \Sigma_4(5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_2(a_3, a_2, a_1, a_0)} = \Sigma_4(1, 2, 3, 4, 9) + d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_1(a_3, a_2, a_1, a_0)} = \Sigma_4(0, 3, 4, 7, 8) + d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_0(a_3, a_2, a_1, a_0)} = \Sigma_4(0, 2, 4, 6, 8) + d(10, 11, 12, 13, 14, 15)$

En forma de Producte de Màxterms (tenint en compte els termes *don't care*):

$\mathbf{e_3(a_3, a_2, a_1, a_0)} = \Pi_4(0, 1, 2, 3, 4) \cdot d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_2(a_3, a_2, a_1, a_0)} = \Pi_4(0, 5, 6, 7, 8) \cdot d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_1(a_3, a_2, a_1, a_0)} = \Pi_4(1, 2, 5, 6, 9) \cdot d(10, 11, 12, 13, 14, 15)$
$\mathbf{e_0(a_3, a_2, a_1, a_0)} = \Pi_4(1, 3, 5, 7, 9) \cdot d(10, 11, 12, 13, 14, 15)$

c. Simplificar al màxim les 4 funcions de sortida. **(1 Punt)**

$\mathbf{e_3(a_3, a_2, a_1, a_0)} = \Sigma_4(5, 6, 7, 8, 9) + d(10, 11, 12, 13, 14, 15)$

$a_1a_0 \setminus a_3a_2$	00	01	11	10
00	$_{0}$	$_{4}$	X $_{12}$	1 $_{8}$
01	$_{1}$	1 $_{5}$	X $_{13}$	1 $_{9}$
11	$_{3}$	1 $_{7}$	X $_{15}$	X $_{11}$
10	$_{2}$	1 $_{6}$	X $_{14}$	X $_{10}$

$\mathbf{e_3(a_3, a_2, a_1, a_0)} = a_3 + a_2 \cdot a_0 + a_2 \cdot a_1$

$\mathbf{e_2(a_3, a_2, a_1, a_0)} = \Sigma_4(1, 2, 3, 4, 9) + d(10, 11, 12, 13, 14, 15)$

$a_1a_0 \setminus a_3a_2$	00	01	11	10
00	$_{0}$	1 $_{4}$	X $_{12}$	$_{8}$
01	1 $_{1}$	$_{5}$	X $_{13}$	1 $_{9}$
11	1 $_{3}$	$_{7}$	X $_{15}$	X $_{11}$
10	1 $_{2}$	$_{6}$	X $_{14}$	X $_{10}$

$\mathbf{e_2(a_3, a_2, a_1, a_0)} = a_2 \cdot a_1' \cdot a_0' + a_2' \cdot a_0 + a_2' \cdot a_1$

$\mathbf{e_1(a_3, a_2, a_1, a_0)} = \Sigma_4(0, 3, 4, 7, 8) + d(10, 11, 12, 13, 14, 15)$

$a_1a_0 \setminus a_3a_2$	00	01	11	10
00	1 $_{0}$	1 $_{4}$	X $_{12}$	1 $_{8}$
01	$_{1}$	$_{5}$	X $_{13}$	$_{9}$
11	1 $_{3}$	1 $_{7}$	X $_{15}$	X $_{11}$
10	$_{2}$	$_{6}$	X $_{14}$	X $_{10}$

$\mathbf{e_1(a_3, a_2, a_1, a_0)} = a_1' \cdot a_0' + a_1 \cdot a_0$

$\mathbf{e_0(a_3, a_2, a_1, a_0)} = \Sigma_4(0, 2, 4, 6, 8) + d(10, 11, 12, 13, 14, 15)$

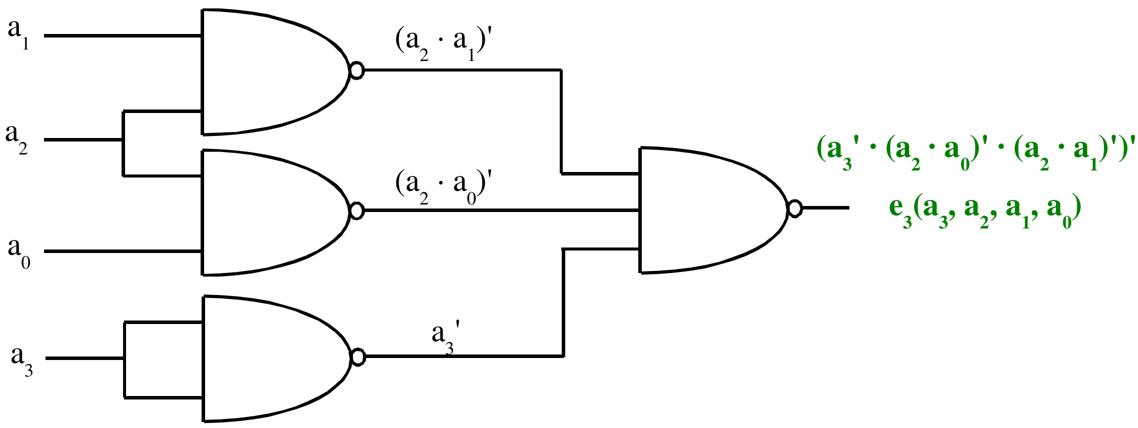
$a_1a_0 \setminus a_3a_2$	00	01	11	10
00	1 $_{0}$	1 $_{4}$	X $_{12}$	1 $_{8}$
01	$_{1}$	$_{5}$	X $_{13}$	$_{9}$
11	$_{3}$	$_{7}$	X $_{15}$	X $_{11}$
10	1 $_{2}$	1 $_{6}$	X $_{14}$	X $_{10}$

$\mathbf{e_0(a_3, a_2, a_1, a_0)} = a_0'$

d. Implementar el conversor amb portes NAND. **(1 Punt)**

$\mathbf{e_3(a_3, a_2, a_1, a_0)} = a_3 + a_2 \cdot a_0 + a_2 \cdot a_1$

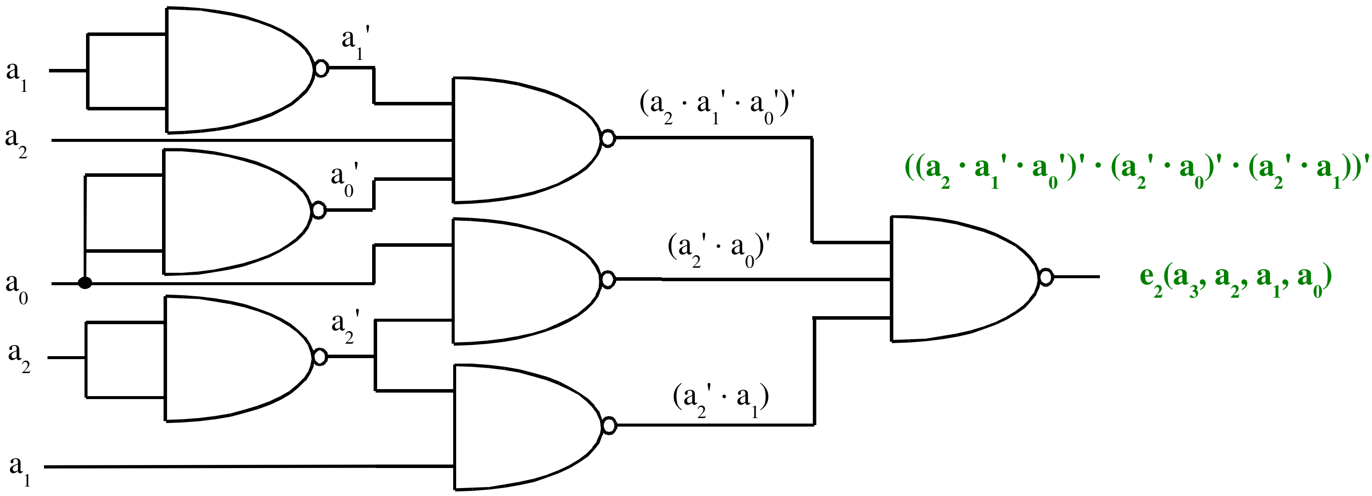
$\mathbf{e_3(a_3, a_2, a_1, a_0)} = a_3 + a_2 \cdot a_0 + a_2 \cdot a_1 = (a_3 + a_2 \cdot a_0 + a_2 \cdot a_1)'' = \mathbf{(a_3' \cdot (a_2 \cdot a_0)' \cdot (a_2 \cdot a_1)')'}$



$\mathbf{e_2(a_3, a_2, a_1, a_0)} = a_2 \cdot a_1' \cdot a_0' + a_2' \cdot a_0 + a_2' \cdot a_1$

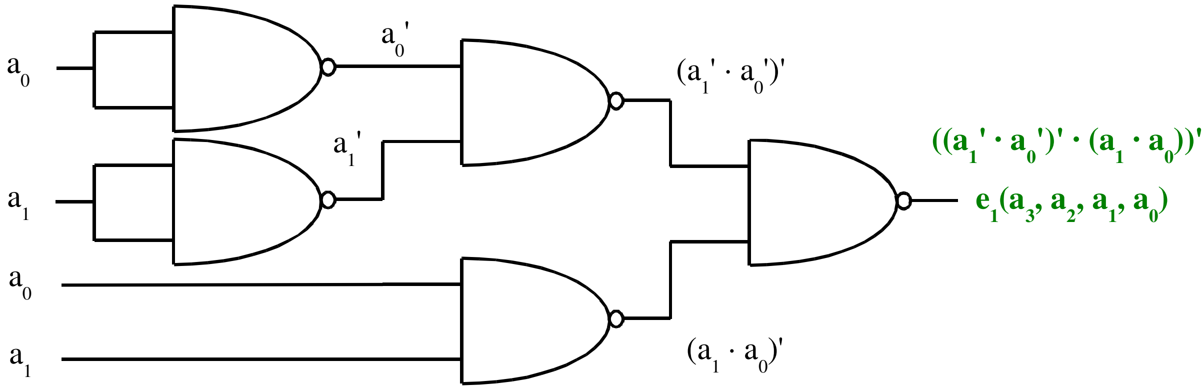
$\mathbf{e_2(a_3, a_2, a_1, a_0)} = a_2 \cdot a_1' \cdot a_0' + a_2' \cdot a_0 + a_2' \cdot a_1 = (a_2 \cdot a_1' \cdot a_0' + a_2' \cdot a_0 + a_2' \cdot a_1)'' =$

$= \mathbf{((a_2 \cdot a_1' \cdot a_0')' \cdot (a_2' \cdot a_0)' \cdot (a_2' \cdot a_1)')'}$



$\mathbf{e_1(a_3, a_2, a_1, a_0)} = a_1' \cdot a_0' + a_1 \cdot a_0$

$\mathbf{e_1(a_3, a_2, a_1, a_0)} = a_1' \cdot a_0' + a_1 \cdot a_0 = (a_1' \cdot a_0' + a_1 \cdot a_0)'' = \mathbf{((a_1' \cdot a_0')' \cdot (a_1 \cdot a_0)')'}$



$\mathbf{e_0(a_3, a_2, a_1, a_0)} = \mathbf{a_0'}$