Fonaments de computadors

TEMA 2. PRINCIPIS DEL DISSENY DIGITAL

Objectius

- Conèixer les funcions lògiques i la representació d'aquestes.
- Dissenyar circuits lògics senzills.
- Fonaments de l'Àlgebra de Boole.
- Mètodes de simplificació. Mapes de Karnaugh.

Recursos d’aprenentatge

- Poliformat, secció “Recursos”
  - Exercicis sense solució.
  - Solucions als exercicis.
  - **Entrenador de Karnaugh.**
  - Exàmens d’anys anteriors.

- Poliformat, secció “Lessons”
  - Mòdul 2: Principios de diseño digital.
    - » *Taules de veritat.*
    - » *Portes lògiques.*

Índex

- Introducció
- Funcions lògiques i taules de veritat
- Portes lògiques
- Àlgebra de Boole
- Anàlisi de circuits
- Formes canòniques de representar una funció lògica
- Simplificació de funcions lògiques
  - Mapes de Karnaugh

Introducció

- Transistor
  - Unitat física mínima de disseny digital
- Porta lògica
  - Unitat lògica mínima de disseny digital

Introducció

- Circuit combinacional
  - Les eixides només depenen del valor de les entrades en el moment actual.
    - Exemple. Selecció de la beguda en una màquina de cafè.
- Circuit seqüencial
  - Les eixides depenen del valor actual de les entrades i de la seqüència de valors anteriors (història) del circuit.
    - Exemple. Magatzem de monedes d'una màquina de cafè.
- Unitat funcional
  - Conjunt de petits circuits que fan una funció definida.

Funcions lògiques i taules de veritat

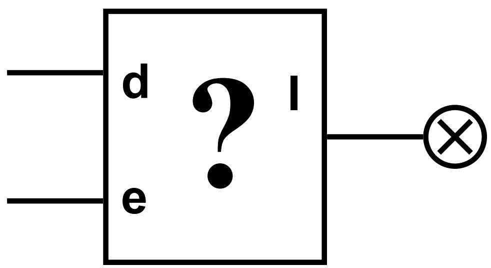
- Funció lògica
  - Expressió formal del comportament d'un circuit lògic
  - Permet determinar l'eixida del circuit en funció de les entrades
  - **Aritat** = nombre de variables lògiques d'entrada
  - **Valoració** = una de les combinacions de valors de les entrades

Funcions lògiques i taules de veritat (ii) FCO

- Taula de veritat
  - Forma tabular d'expressar una funció lògica.
  - Per a cada entrada o eixida s'assigna una columna.
  - Per a cada valoració s'assigna una fila.
  - Entrades a l'esquerra, eixides a la dreta.
  - Valoracions seguint la numeració binària.

Taules de veritat. Exemples

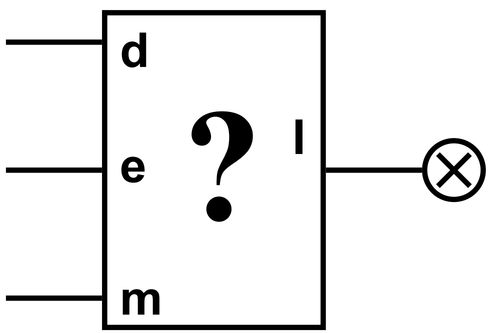
- Llum interior d'un cotxe
  - A partir de dues entrades *d*, *e* (portes dreta i esquerra), dissenyeu un circuit que encenga un llum *l* quan alguna de les portes estiga oberta.



d	e	l
0	0	0
0	1	1
1	0	1
1	1	1

Taules de veritat. Exemples (ii)

- Llum interior d'un cotxe (ii)
  - Afegiu una entrada *m* d'encesa manual: si l'entrada *m* està activada (*m=1*) s'encén el llum independentment de l'estat (obert/tancat) de les portes.



m	d	e	l
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

m	d	e	l
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	X	X	1

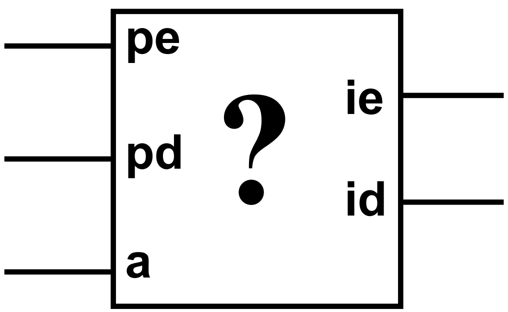
Taula de veritat reduïda

Taules de veritat. Exemples (iii)

- Funcions amb <u>entrades indiferents</u>
  - Aquelles combinacions de valors d'entrada per a les quals no importa el valor de l'eixida, atès que...
    - Es tracta d'una combinació de les entrades per a la qual no s'ha especificat el comportament del circuit.
    - O es tracta d'una combinació de les entrades que és impossible.
  - En la taula de veritat, l'eixida per a aquestes valoracions és X.

Taules de veritat. Exemples (iv)

- Intermitents d'un cotxe
  - A partir de 3 entrades: palanca a l'esquerra (*pe*), palanca a la dreta (*pd*) i avaria (*a*), genereu les eixides que activen els intermitents esquerre (*ie*) i dret (*id*).

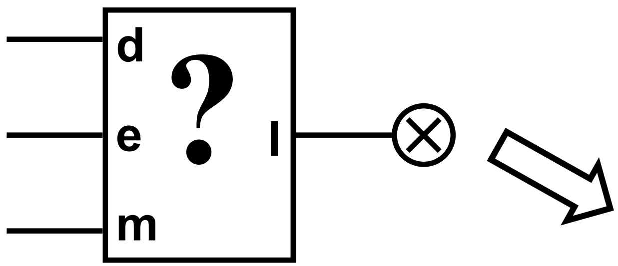


a	pe	pd	ie	id
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	X	X
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	X	X

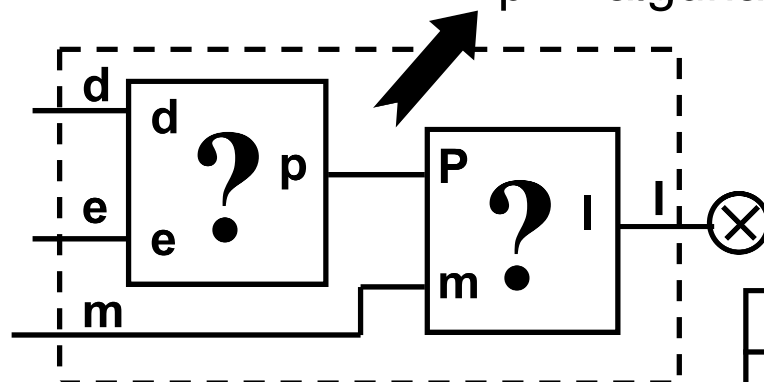
Taula de veritat d'una <u>funció amb entrades indiferents</u>

Composició de funcions

- Funció composta. Aquella en què l'eixida d'una (sub)funció és utilitzada com a entrada d'una altra.
- Exemple: llum interior de cotxe amb encesa manual.



p ≡ “alguna porta oberta”



d	e	p
0	0	0
0	1	1
1	0	1
1	1	1

m	p	l
0	0	0
0	1	1
1	0	1
1	1	1

Portes lògiques

- Porta lògica. Circuit electrònic que implementa una funció lògica elemental.

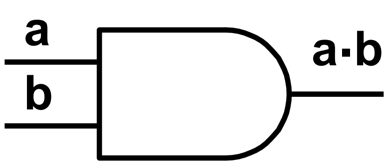
- Tipus
  - Bàsics: AND, OR, NOT
  - Altres: XOR
  - Amb eixida negada: NAND, NOR, XNOR

- Tecnologies. Base física de construcció
  - TTL, CMOS

Portes lògiques (ii)

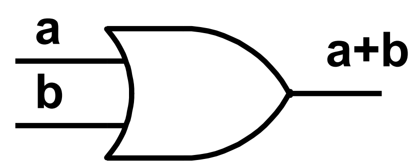
- AND
  - Producte lògic (“i”)
  - Ampliable

b	a	a·b
0	0	0
0	1	0
1	0	0
1	1	1



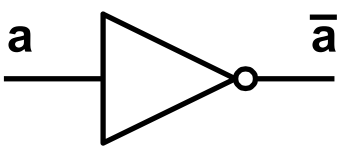
- OR
  - Suma lògica (“o”)
  - Ampliable

b	a	a+b
0	0	0
0	1	1
1	0	1
1	1	1



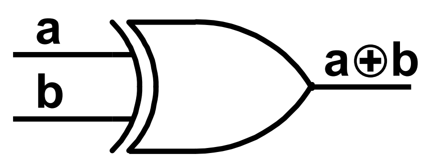
- NOT
  - Negació lògica (“no”)
  - No ampliable

a	$\bar{a}$
0	1
1	0



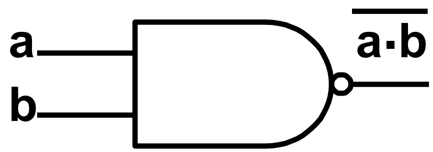
- XOR
  - OR exclusiva
  - No ampliable

b	a	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0



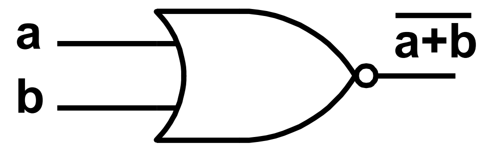
Portes lògiques amb eixida negada

- NAND = NOT (AND)
  - Ampliable



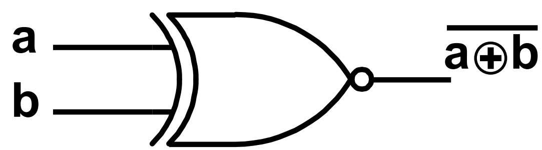
b	a	$\overline{a \cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

- NOR = NOT (OR)
  - Ampliable



b	a	$\overline{a+b}$
0	0	1
0	1	0
1	0	0
1	1	0

- XNOR = NOT (XOR)
  - No ampliable



b	a	$\overline{a \oplus b}$
0	0	1
0	1	0
1	0	0
1	1	1

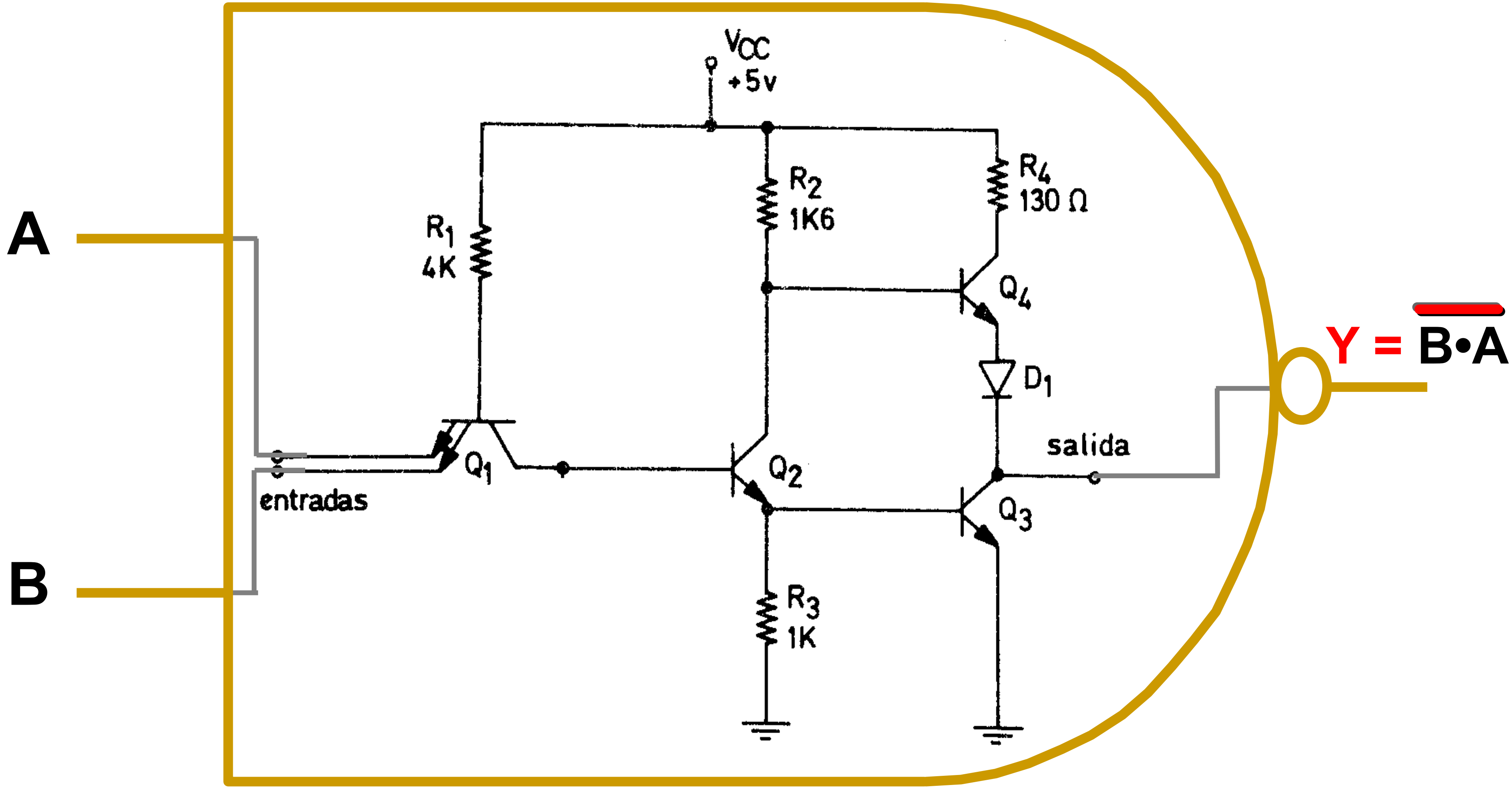
Portes lògiques. Tecnologies

- Cada tecnologia de construcció empra diferents tipus d'elements físics (transistors) i tensions per a representar els valors lògics “0” i “1”

- TTL = *Transistor-Transistor Logic*
  - Basada en transistors bipolars
  - Alta velocitat, alt consum, difícil integració

- CMOS = *Complementary Metal Oxide Semiconductor*
  - Basada en transistors MOSFET
  - Menor velocitat, baix consum, alta escala d'integració

Esquema físic d'una NAND TTL



DISCA

National Semiconductor

54LS00/DM54LS00/DM74LS00 Quad 2-Input NAND Gates

Function Table

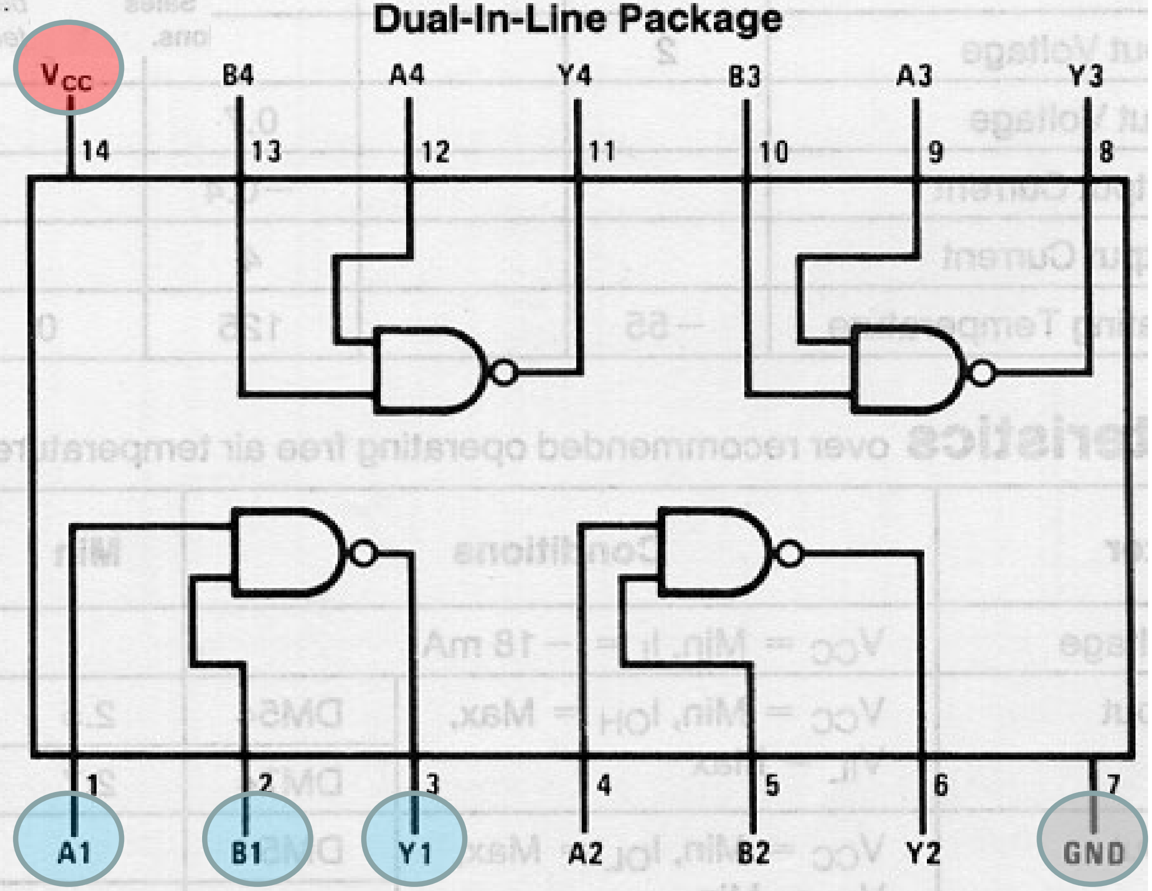
$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = High Logic Level

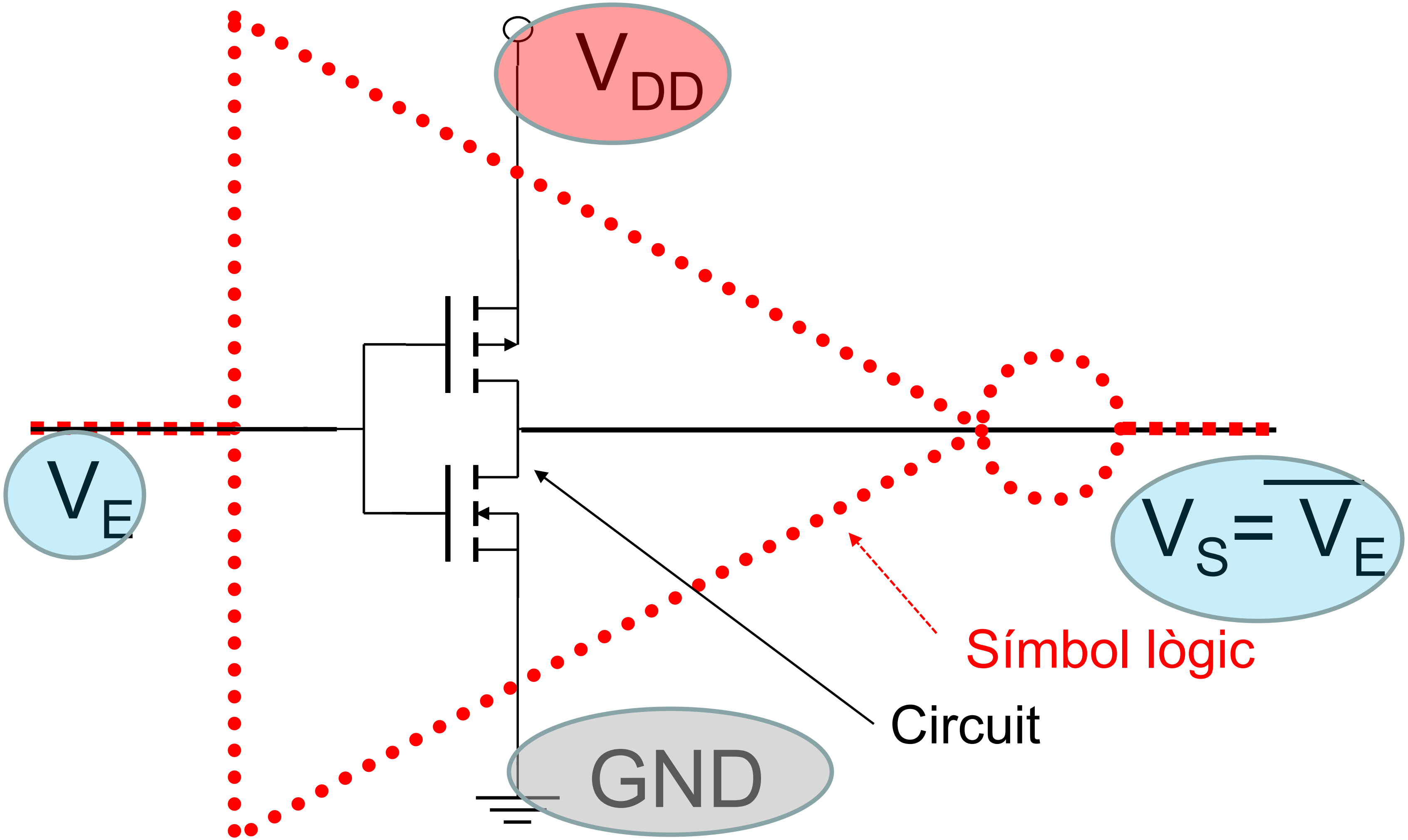
L = Low Logic Level

Dual-In-Line Package

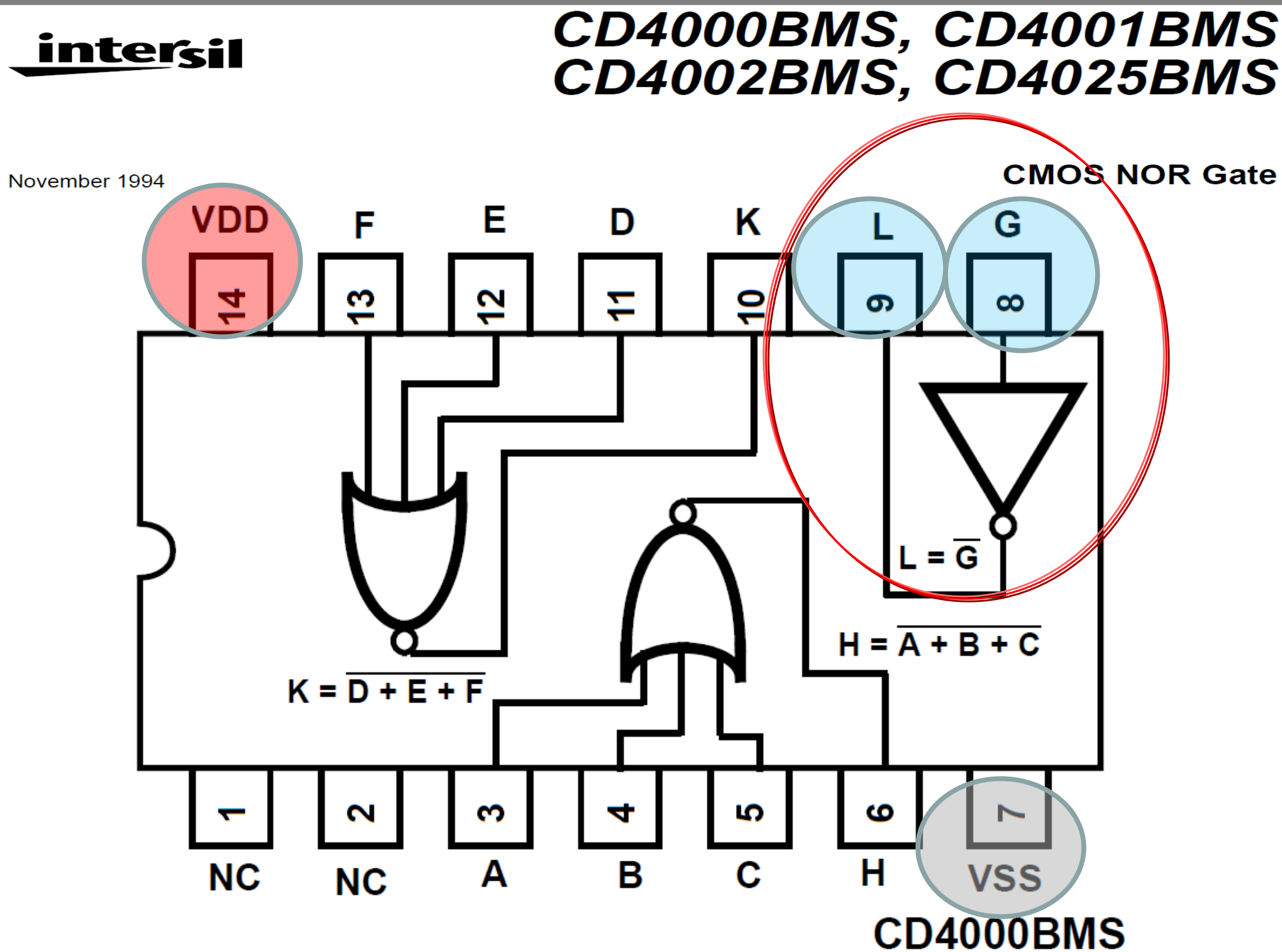


DISCA

Esquema físic d'un inversor CMOS

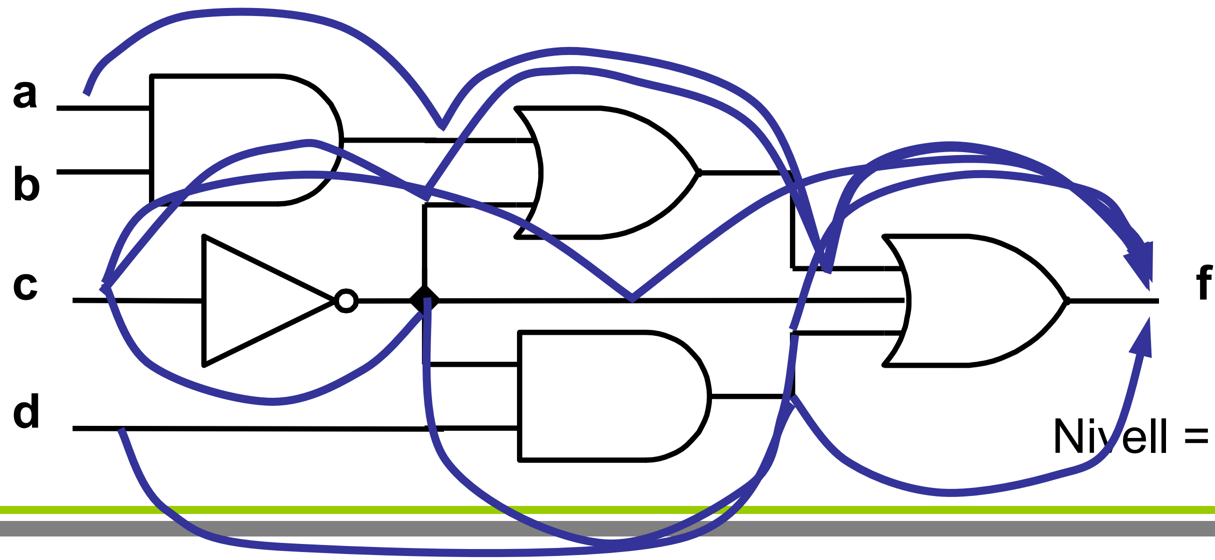


Circuit integrat amb dues NOR i una NOT FCO



Nivell d'un circuit lògic

- Nivell
  - Nombre de portes que cal travessar en el pitjor dels casos des de les entrades fins a les eixides del circuit.
  - És una indicació del retard del circuit.
  - Cada porta té un retard T.
  - Nivell 0 = entrades.



Travessa 2 portes.
Travessa 3 portes.
Travessa 2 portes.
Travessa 3 portes.
Travessa 3 portes.

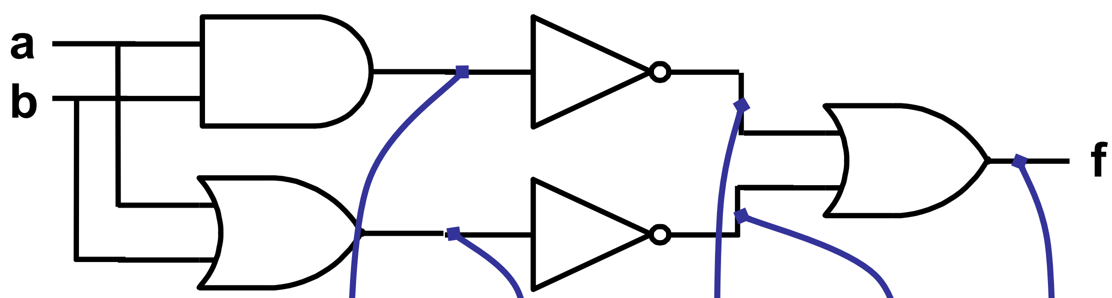
Nivell = pitjor cas = 3 portes

Anàlisi de circuits

- Donat un circuit, es tracta d'obtenir-ne la funció lògica i la taula de veritat.
  - Funció lògica: component les subfuncions corresponents a cada punt del circuit.
  - Taula de veritat: calculant l'eixida per a totes les possibles combinacions d'entrada.

Anàlisi de circuits (ii)

- Exemple



Funció lògica

b	a	$a\cdot b$	$a + b$	$\overline{a\cdot b}$	$\overline{a + b}$	$f = \overline{a\cdot b} + \overline{a + b}$
0	0	0	0	1	1	1
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

$\Rightarrow$

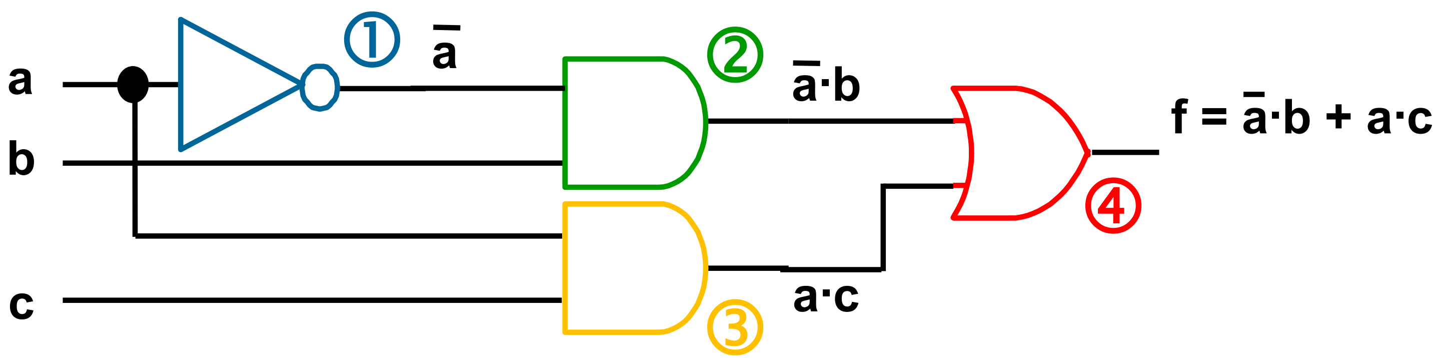
b	a	f
0	0	1
0	1	1
1	0	1
1	1	0

Taula de veritat

Síntesi d'un circuit lògic

- A partir de la funció lògica corresponent:
  - Afegiu i interconnecteu les portes en l'ordre en què s'avaluen els termes de l'expressió lògica.
  - Exemple:

$$f = \boxed{\boxed{\boxed{\overline{a}}\cdot b} + \boxed{a \cdot c}}$$



Àlgebra de Boole

- George Boole (s. XIX)
  - Matemàtic i filòsof anglès.
  - Desenvolupa una estructura algebraica amb dos valors ("vertader", "fals") i dues lleis de composició interna ("i", "o").
  - Permet formalitzar les regles del raonament lògic.

- Claude Shannon (1938, Lab. Bell)
  - Adapta aquest àlgebra a la computació.
    - Valors 0 i 1, lleis de composició AND i OR
  - Permet formalitzar les regles de construcció de circuits digitals.

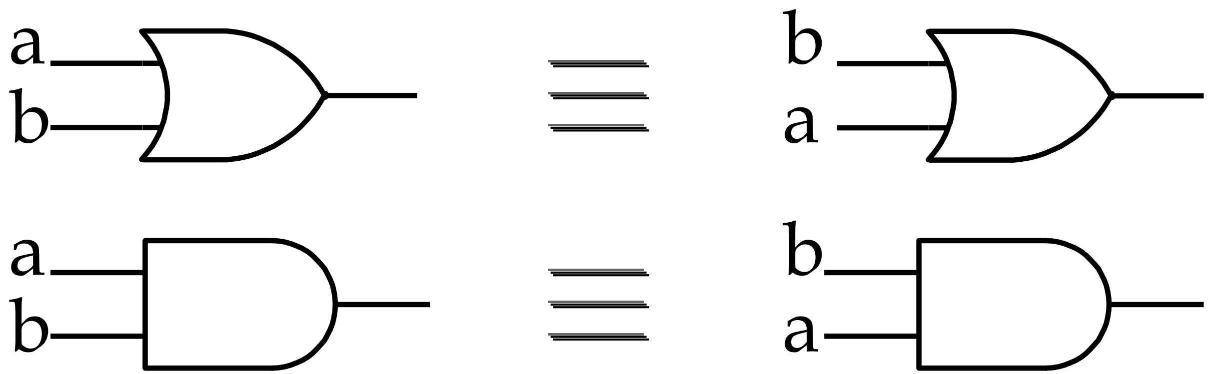
Precedència (si no hi ha parèntesi)

Porta lògica	Símbol estàndard
NOT	$\overline{}$
AND	$\cdot$
OR	$+$
XOR	$\oplus$

- Commutativitat (+ , •)

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$



- Distributivitat (+ ⇔ •)

$$(a + b) \cdot (a + c) = a + (b \cdot c)$$

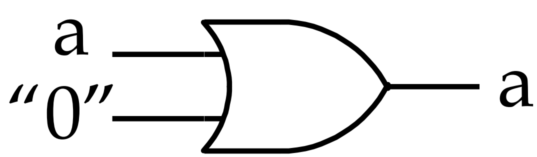
$$(a \cdot b) + (a \cdot c) = a \cdot (b + c)$$

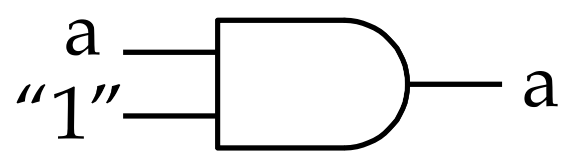
Àlgebra de Boole. Axiomes (ii)

- Existència d'element neutre (+ , •)

$$a + 0 = a$$

$$a \cdot 1 = a$$





- Existència d'element complementari **(+ , •)**

$$a + \overline{a} = 1$$

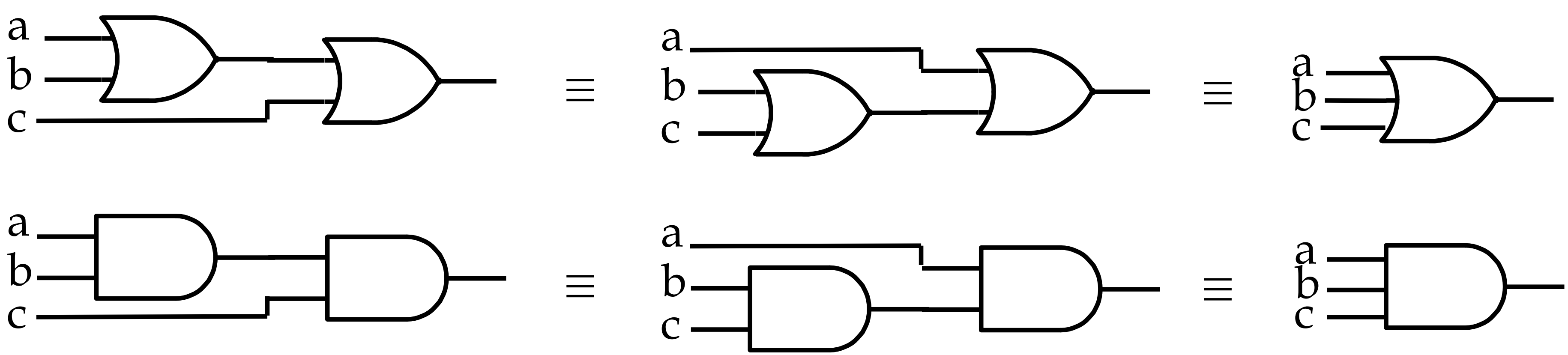
$$a \cdot \overline{a} = 0$$

Àlgebra de Boole. Propietats

- Associativa (+ , •)

$$(a + b) + c = a + (b + c) = a + b + c$$
$$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$$

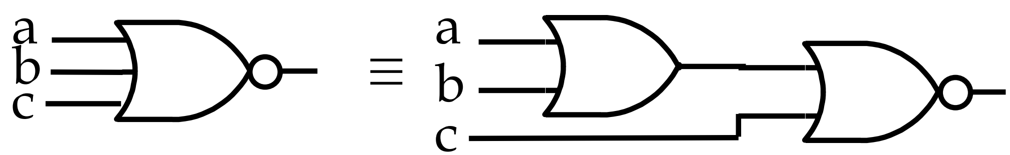
  – Permet construir portes amb un nombre més gran d'entrades a partir de portes amb menys entrades:

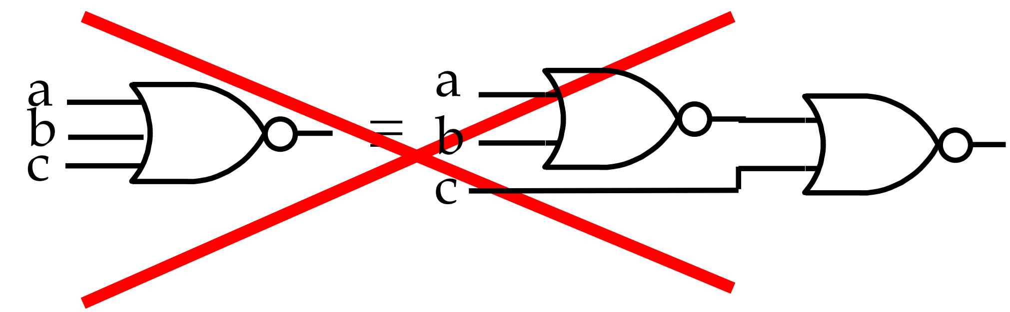


Àlgebra de Boole. Propietats (ii)

- Associativa (ii)
  - ATENCIÓ a les portes amb eixida negada!

$$\overline{a + b + c} = \overline{(a + b) + c}$$

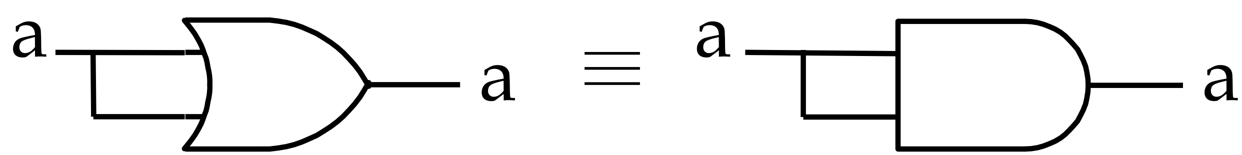




Àlgebra de Boole. Propietats (iii)

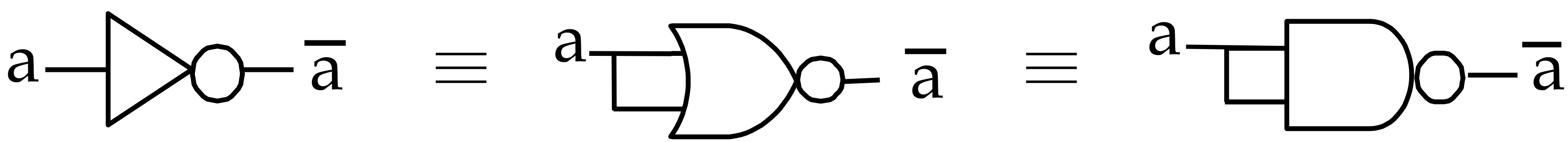
- Idempotència (+ , •)

$$a + a = a \qquad a \cdot a = a$$



  - Permet construir portes NOT a partir de NAND o NOR

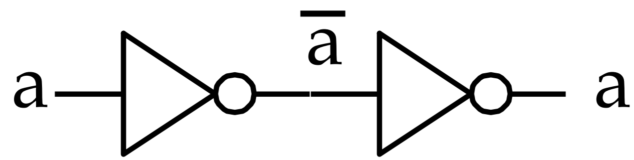
$$\overline{a + a} = \overline{a} = \overline{a \cdot a}$$



Àlgebra de Boole. Propietats (iv) FCO

- Involució

$$\overline{\overline{a}} = a$$



- Lleis de De Morgan

$$\overline{(a + b + \ldots + n)} = \overline{a} \cdot \overline{b} \cdot \ldots \cdot \overline{n}$$

$$\overline{(a \cdot b \cdot \ldots \cdot n)} = \overline{a} + \overline{b} + \ldots + \overline{n}$$

– ATENCIÓ!

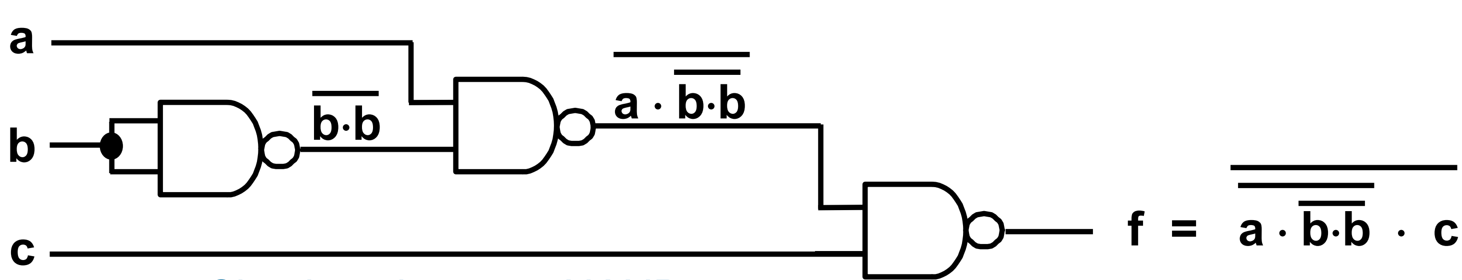
$$\cancel{\overline{(a + b)} = \overline{a} + \overline{b}} \qquad \overline{(a + b)} = \overline{a} \cdot \overline{b}$$

Àlgebra de Boole. Propietats (v)

- Aplicant adequadament les lleis de De Morgan, **qualsevol** circuit lògic es pot **implementar** només amb portes **NAND** o **NOR**.
  - Exemple:

INVOLUCIÓ — DE MORGAN — INVOLUCIÓ — IDEMPOTÈNCIA (aquest últim pas se sol obviar)

$$f = a \cdot \bar{b} + \bar{c} = \overline{\overline{a \cdot \bar{b} + \bar{c}}} = \overline{\overline{a \cdot \bar{b}} \cdot \bar{\bar{c}}} = \overline{\overline{a \cdot \bar{b}} \cdot c} = \overline{\overline{a \cdot \overline{b \cdot b}} \cdot c}$$



Circuit amb portes NAND

Formes canòniques

- Expressió algebraica única d'una funció lògica formulada amb maxitermes o minitermes.

- Miniterme d'ordre n
  - Producte en què apareixen les n variables lògiques d'entrada.
  - Cada variable apareix complementada si el valor n'és 0.
  - Cada valoració dóna lloc a un miniterme distint.
  - Els minitermes es numeren segons la quantitat representada per la valoració corresponent.

Formes canòniques (ii)

- Maxiterme d'ordre n
  - Suma en què apareixen les n variables lògiques d'entrada.
  - Cada variable apareix complementada si el valor n’és 1.
  - Cada valoració dóna lloc a un maxiterme distint.
  - Els maxitermes es numeren segons la quantitat representada per la valoració corresponent.

Forma canònica disjuntiva

- Forma canònica disjuntiva o suma de productes
  - Suma dels minitermes pertanyents a la funció.
  - Pertanyen a la funció els minitermes corresponents a les valoracions per a les quals la funció val 1.

$$\sum_{\text{llista de variables de la funció}} \left(\text{llista numerada dels minitermes de la funció} \right)$$

b	a	f	miniterme	núm.
0	0	0	$\overline{b} \cdot \overline{a}$	0
0	1	1	$\overline{b} \cdot a$	1
1	0	0	$b \cdot \overline{a}$	2
1	1	1	$b \cdot a$	3

Forma canònica

$$f = \sum_{b,a} (1,3) = \overline{b} \cdot a + b \cdot a$$

Expressió algebraica equivalent

Forma canònica conjuntiva

- Forma canònica conjuntiva o producte de sumes
  - Producte dels maxitermes de la funció.
  - Pertanyen a la funció els maxitermes corresponents a les valoracions per a les quals la funció val 0.

$$\prod_{\text{llista de variables de la funció}} \left(\text{llista numerada dels maxitermes de la funció}\right)$$

b	a	f	maxiterme	núm.
0	0	0	$b + a$	0
0	1	1	$b + \overline{a}$	1
1	0	0	$\overline{b} + a$	2
1	1	1	$\overline{b} + \overline{a}$	3

Forma canònica

$$f = \prod_{b,a}(0,2) = (b+a)\cdot(\overline{b}+a)$$

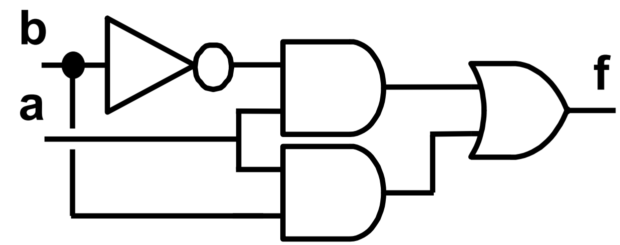
Expressió algebraica equivalent

Formes canòniques. Interès

- Expressió única i compacta d'una funció lògica.
- Primera aproximació a la síntesi de circuits a partir d'una taula de veritat:

b	a	f
0	0	0
0	1	1
1	0	0
1	1	1

$$f = \sum_{b,a}(1,3) = \overline{b} \cdot a + b \cdot a$$



- Qualsevol funció lògica pot implementar-se per mitjà d'un circuit de nivell ≤ 3.

DISCA

Formes canòniques. Entrades indiferents FCO

- Formes canòniques per a funcions amb entrades indiferents
  - Aquestes combinacions s'agrupen per separat en sumatoris o productoris del conjunt buit *Φ*.

	a	pe	pd	id
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	X
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	X

$$id = \sum_{a,\, pe,\, pd} (1, 4, 5, 6) + \sum_{\phi} (3, 7)$$

$$id = \prod_{a,\, pe,\, pd} (0, 2) \bullet \prod_{\phi} (3, 7)$$

Simplificació

- Simplificar una funció
  - Consisteix a trobar una expressió algebraica equivalent a la de partida, però de menor grandària (menys termes, termes amb menys variables).
  - L'objectiu n’és reduir al màxim el circuit amb què s'implementa una funció lògica.
- Metodologia
  - Algebraica. Aplicació d'axiomes i propietats de l'àlgebra de Boole
    - Element complementari, element neutre, distributiva i associativa
  - Gràfica. Mapes de Karnaugh

Simplificació per Karnaugh

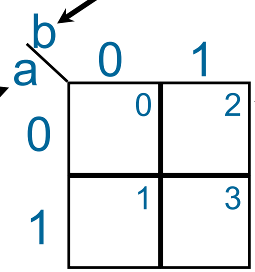
- Mapa de Karnaugh
  - Representació matricial d'una taula de veritat.
  - Una cel·la del mapa de Karnaugh representa una fila de la taula de veritat.
  - En cada cel·la es col·loca el valor d'una eixida de la funció.
  - La disposició espacial de les cel·les és tal que els termes adjacents de la funció lògica estan en cel·les adjacents.
  - Dos termes es diuen adjacents si les valoracions en difereixen en el valor d'una sola variable.
  - Les vores del mapa de Karnaugh han de considerar-se com adjacents.

Simplificació per Karnaugh (ii)

- Mapes per a funcions de 2, 3 i 4 variables

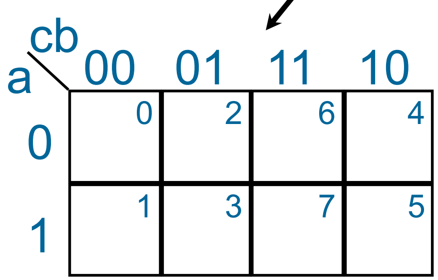
variables de major pes

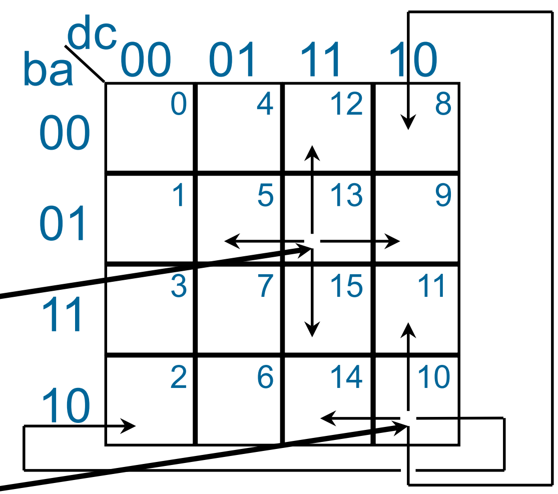
variables de menor pes



número de cel·la / terme (2_{10} => b=1, a=0)

numeració de cel·les en codi Grey





cel·les adjacents a la cel·la 13

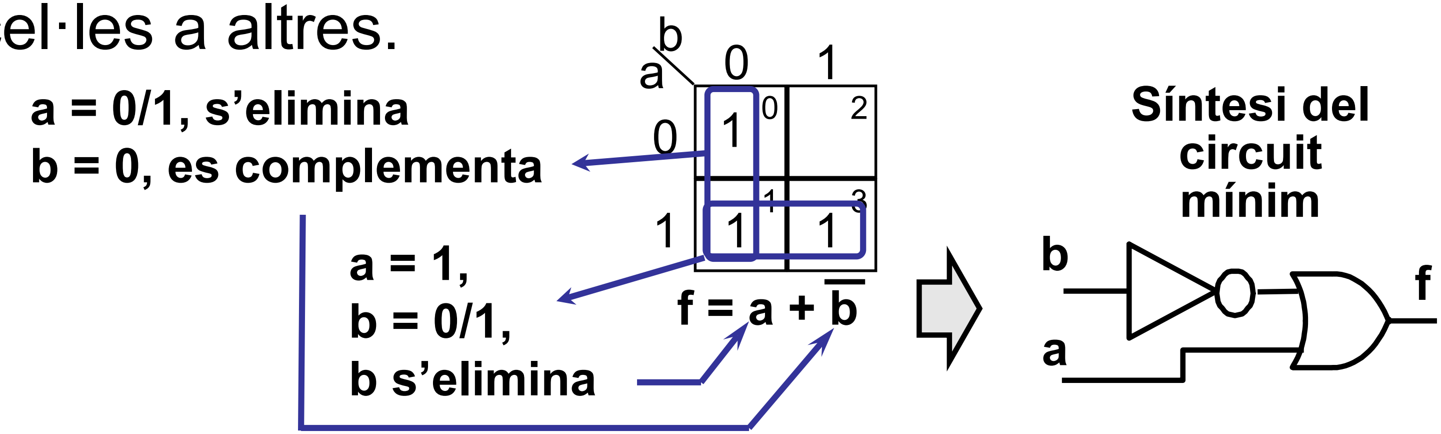
cel·les adjacents a la cel·la 10

Simplificació per Karnaugh (iii)

- Mètode de simplificació
  - Cal agrupar totes les cel·les amb el mateix valor, en un o més grups.
  - Cada grup ha de contenir un nombre de cel·les adjacents potència de 2.
  - Cal fer els grups tan grans com siga possible.
  - El nombre de grups ha de ser mínim.
  - Una cel·la pot estar en un grup o en més d'un.

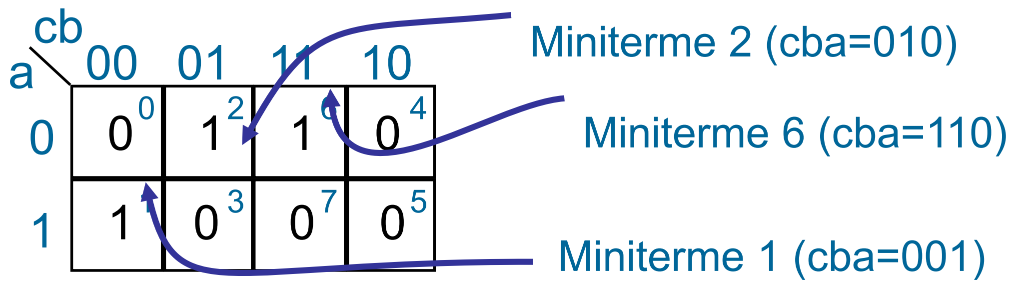
Simplificació per uns. Mètode

- Cal agrupar les cel·les de valor 1.
- Cada grup representa un terme producte (no un miniterme, ja que no apareixen totes les variables de la funció). Les variables a 0 apareixen complementades.
- Un grup de 2^k cel·les elimina k variables del terme resultant, i per tant aquest tindrà n-k variables.
- En cada grup s'eliminen les variables que canvien de valor d'unes cel·les a altres.



Simplificació per uns. Fonament

– Cada cel·la a 1 representa un miniterme que pertany a la funció:



– La funció sense simplificar inclouria tots els minitermes que hi pertanyen:

$$\mathbf{f} = \sum_{c,b,a}(1, 2, 6) = \bar{c} \cdot \bar{b} \cdot a + \bar{c} \cdot b \cdot \bar{a} + c \cdot b \cdot \bar{a}$$

Simplificació per uns. Fonament (ii)

– Els grups de cel·les adjacents detecten minitermes amb un factor comú:

a \ cb	00	01	11	10
0	0^{0}	1^{2}	1^{6}	0^{4}
1	1^{1}	0^{3}	0^{7}	0^{5}

Miniterme 2 (cba=010) $\longrightarrow \bar{c} \cdot \boxed{b \cdot \bar{a}}$

Miniterme 6 (cba=110) $\longrightarrow c \cdot \boxed{b \cdot \bar{a}}$

– La suma n'és simplificable. Algebraicament seria:

$$\bar{c} \cdot b \cdot \bar{a} + c \cdot b \cdot \bar{a} \overset{\text{ASSOCIATIVA}}{=} \bar{c} \cdot (b \cdot \bar{a}) + c \cdot (b \cdot \bar{a}) \overset{\text{DISTRIBUTIVA}}{=} (\bar{c} + c) \cdot (b \cdot \bar{a}) \overset{\text{ELEM. COMPLEM.}}{=} 1 \cdot (b \cdot \bar{a}) \overset{\text{ELEM. NEUTRE}}{=} \boxed{b \cdot \bar{a}}$$

– Karnaugh obté el mateix resultat:

Cel·la 2 (cba=010)
Cel·la 6 (cba=110)

⇨

c = 0/1, s'elimina
b = 1, s'inclou
a = 0, es complementa

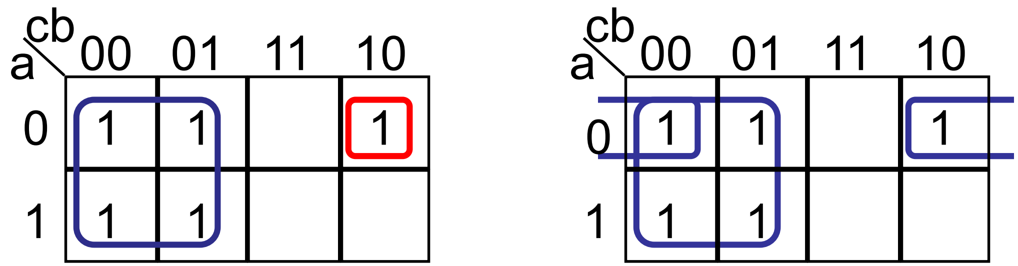
$\boxed{b \cdot \bar{a}}$

Simplificació per uns. Exemples

- Exemples

a \ cb	00	01	11	10
0	1 (0)	1 (2)	0 (6)	1 (4)
1	1 (1)	1 (3)	0 (7)	0 (5)

⇒



$f = \overline{c} + c \cdot \overline{a} \cdot \overline{b}$

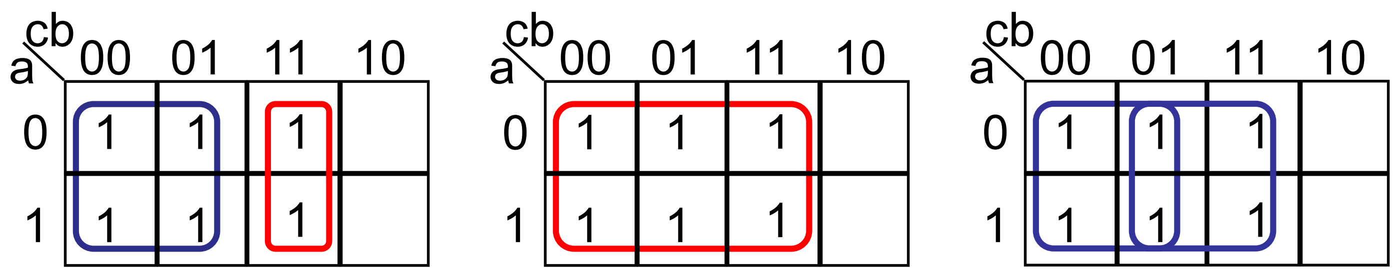
MALAMENT

$f = \overline{c} + \overline{a} \cdot \overline{b}$

BÉ

a \ cb	00	01	11	10
0	1 (0)	1 (2)	1 (6)	0 (4)
1	1 (1)	1 (3)	1 (7)	0 (5)

⇒



$f = \overline{c} + c \cdot b$

MALAMENT

$f = ?$

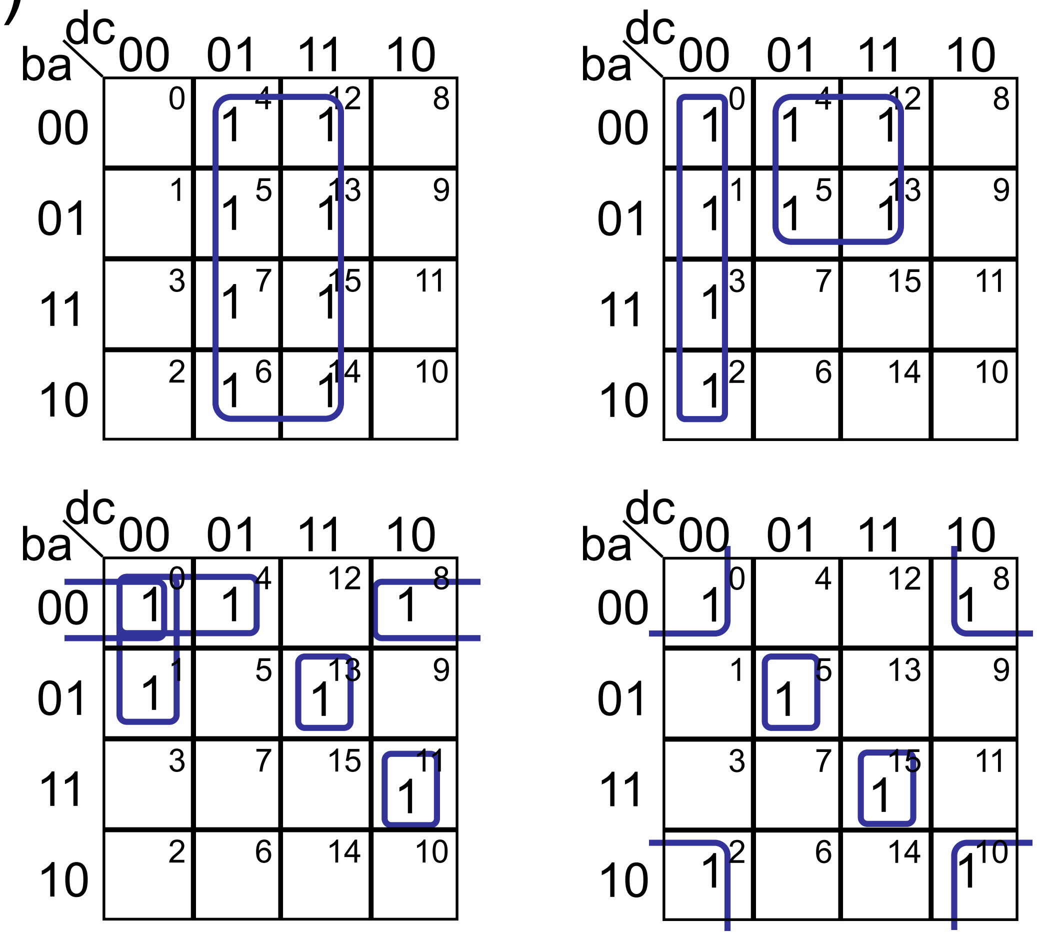
MALAMENT

$f = \overline{c} + b$

BÉ

Simplificació per uns. Exemples (ii)

- Exemples (cont.)



DISCA

Simplificació per zeros

- Cal agrupar les cel·les de valor zero.
- Cada grup representa un terme suma (no un maxiterme, ja que no apareixen totes les variables de la funció). Les variables de valor 1 apareixen complementades.
- Un grup de 2^k cel·les elimina k variables del terme resultant, i per tant aquest tindrà n-k variables.
- En cada grup s'eliminen les variables que canvien de valor d'unes cel·les a altres.

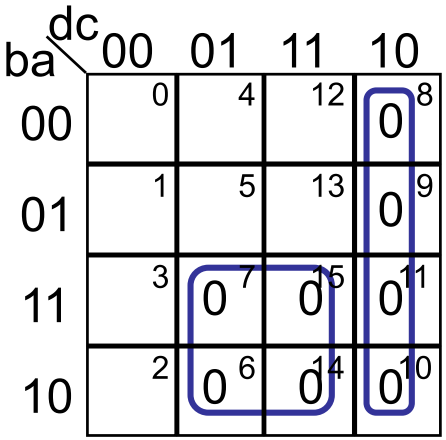
a \ cb	00	01	11	10
0	1^{0}	1^{2}	1^{6}	$\boxed{0^{4}}$
1	1^{1}	1^{3}	1^{7}	$\boxed{0^{5}}$

a = 0/1, s'elimina
b = 0, s'inclou
c = 1, es complementa

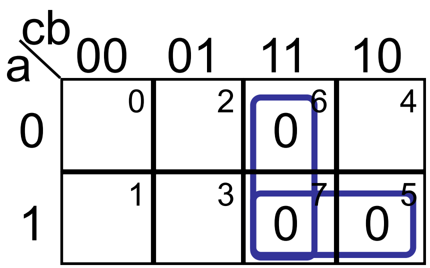
$$f = \overline{c} + b$$

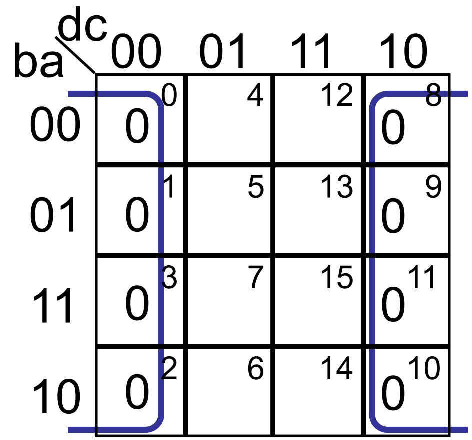
Simplificació per zeros (ii)

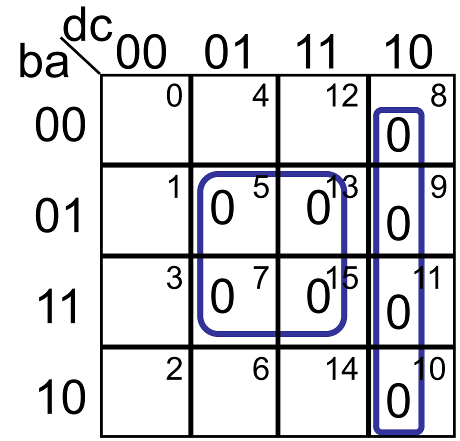
- Exemples

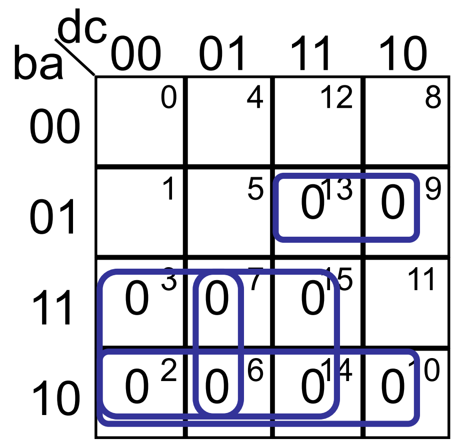


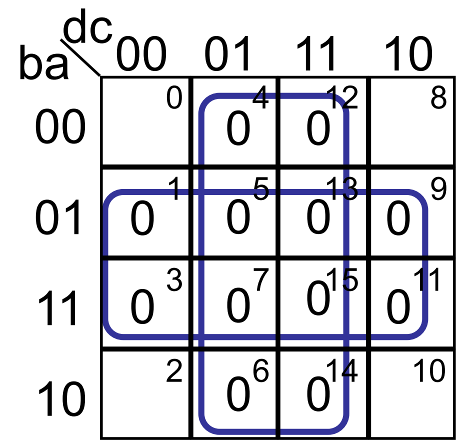
$$f = (\overline{d} + c) \cdot (\overline{c} + \overline{b})$$











Simplificació. Entrades indiferents

– Les cel·les amb "x" es prenen com si tingueren valor 1 o valor 0, cadascuna com més convinga, per maximitzar la simplificació.

d	c	b	a	f
0	0	0	0	0
0	0	0	1	x
0	0	1	0	0
0	0	1	1	x
0	1	0	0	0
0	1	0	1	x
0	1	1	0	x
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	x
1	0	1	1	1
1	1	0	0	x
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$$f = \sum_{d,c,b,a}(7,11,13,14,15) + \sum_{\phi}(1,3,5,6,10,12) =$$

$$\prod_{d,c,b,a}(0,2,4,8,9) \bullet \prod_{\phi}(1,3,5,6,10,12)$$

ba \ dc	00	01	11	10
00	0^{0}	0^{4}	x^{12}	0^{8}
01	x^{1}	x^{5}	1^{13}	0^{9}
11	x^{3}	1^{7}	1^{15}	1^{11}
10	0^{2}	x^{6}	1^{14}	x^{10}

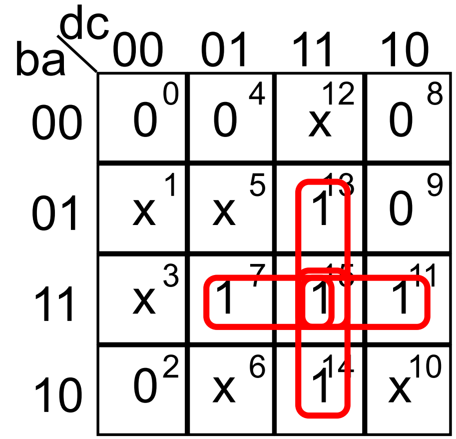
"per uns"

ba \ dc	00	01	11	10
00	0^{0}	0^{4}	x^{12}	0^{8}
01	x^{1}	x^{5}	1^{13}	0^{9}
11	x^{3}	1^{7}	1^{15}	1^{11}
10	0^{2}	x^{6}	1^{14}	x^{10}

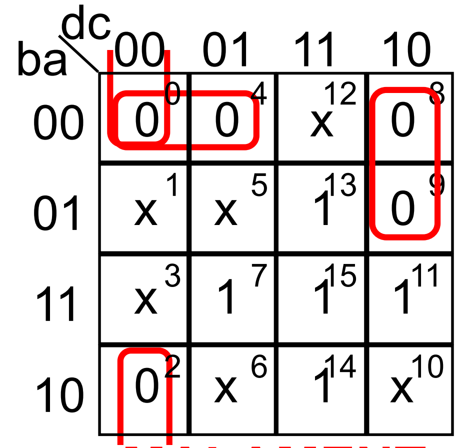
"per zeros"

Simplificació. Entrades indiferents (ii)

- Errors comuns:
  - Prendre totes les “x” per 0 o per 1.



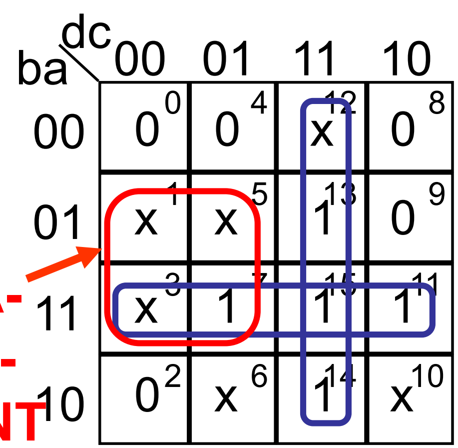
MALAMENT



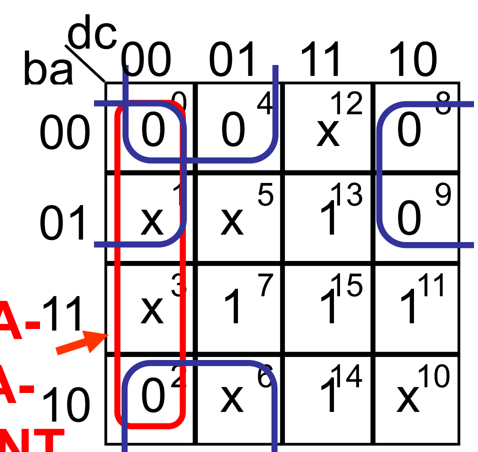
MALAMENT

  - Fer grups amb “x” innecessaris.

MA-LA-MENT



MA-LA-MENT



Recursos d'aprenentatge

- Poliformat, secció "Recursos"
  - Exercicis sense solució.
  - Solucions als exercicis.
  - **Entrenador de Karnaugh.**
  - Exàmens d'anys anteriors.

- Poliformat, secció "Lessons"
  - Mòdul 2: Principios de diseño digital.
    - » *Taules de veritat.*
    - » *Portes lògiques.*

Fonaments de computadors

TEMA 2. PRINCIPIS DEL DISSENY DIGITAL