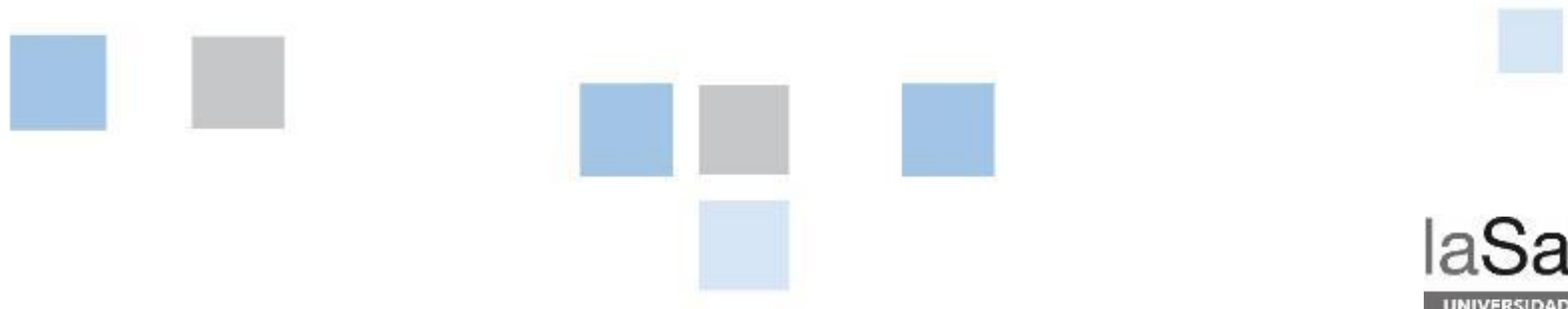


IO – Introducció als Ordinadors

GRUP A – Sessió 13-14

Tema 2. Àlgebra booleana i portes lògiques (II)



Tema 2:

Àlgebra booleana i portes lògiques

2.1. Àlgebra de Boole (4 POSTULATS)

2.2. Funcions booleanes

2.3. Operacions booleanes (portes lògiques)

2.4. Formes canòniques

2.5. Teoremes booleans

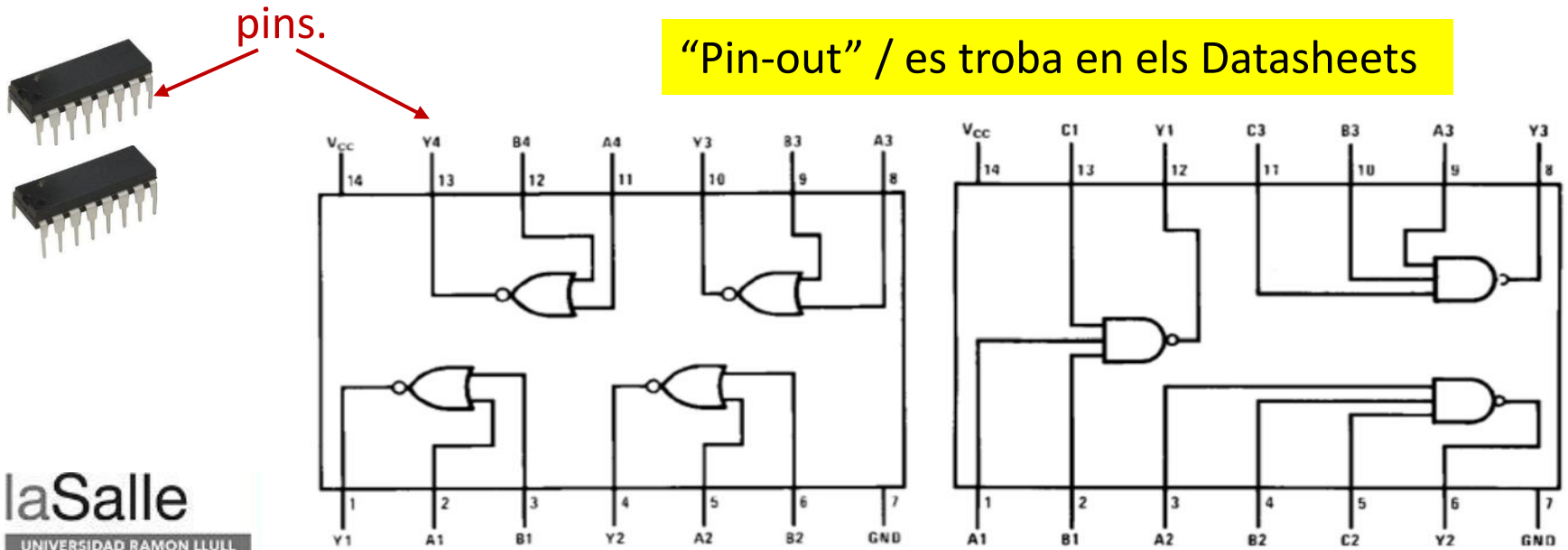
2.6. Implementació de funcions amb portes lògiques

**2.7. Disseny i implementació de sistemes mitjançant
portes lògiques**

Tema 2. Àlgebra booleana i portes lògiques

2.7. Disseny i implementació de sistemes mitjançant portes lògiques

- Manera d'implementar les funcions lògiques.
- Hi ha una sèrie de **circuits integrats (C.I.)** disponibles al mercat que implementen portes de tipus NAND, NOR, AND, OR, XOR, etc.
- La seva interconnexió permet implementar funcions lògiques.
- Els circuits integrats:
 - S'alimenten amb un voltatge de 5 volts / 0 volts (massa).
 - Disposen d'un conjunt de pins d'entrada i sortida.

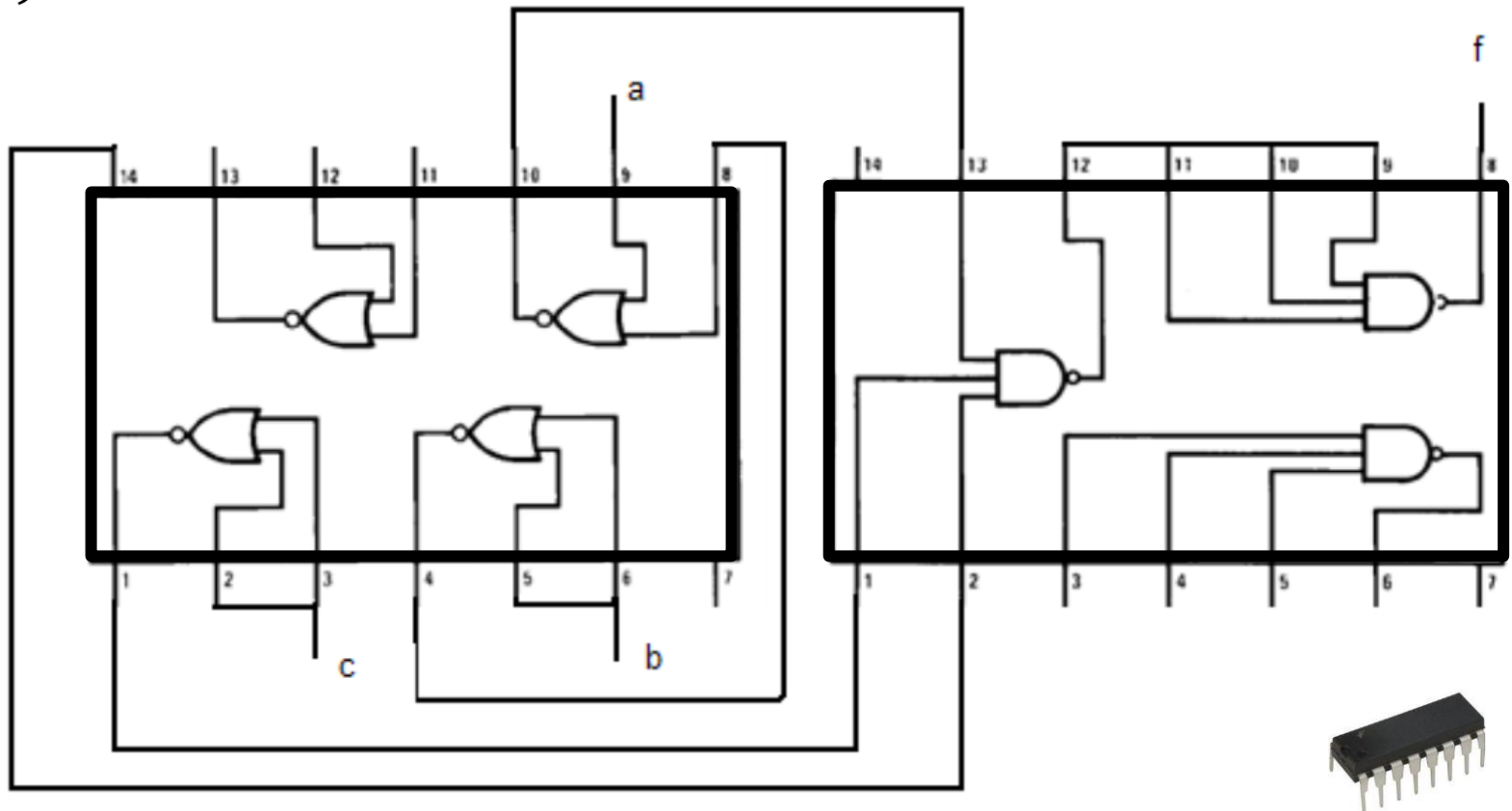


Tema 2. Àlgebra booleana i portes lògiques

2.7. Disseny i implementació de sistemes mitjançant portes lògiques

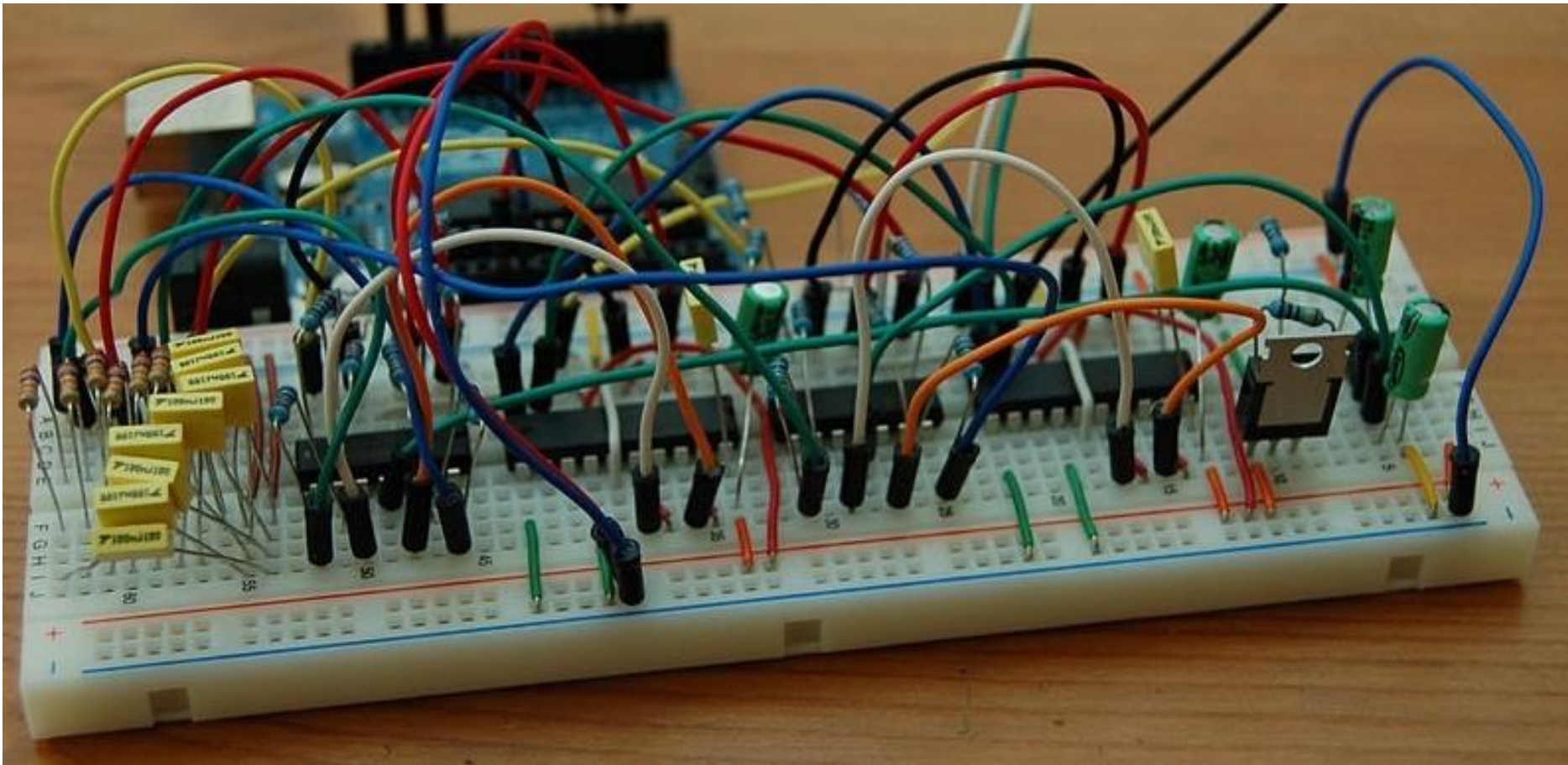
Exemple d'implementació per la funció:

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



Tema 2. Àlgebra booleana i portes lògiques

2.7. Disseny i implementació de sistemes mitjançant portes lògiques



Tema 2. Àlgebra booleana i portes lògiques

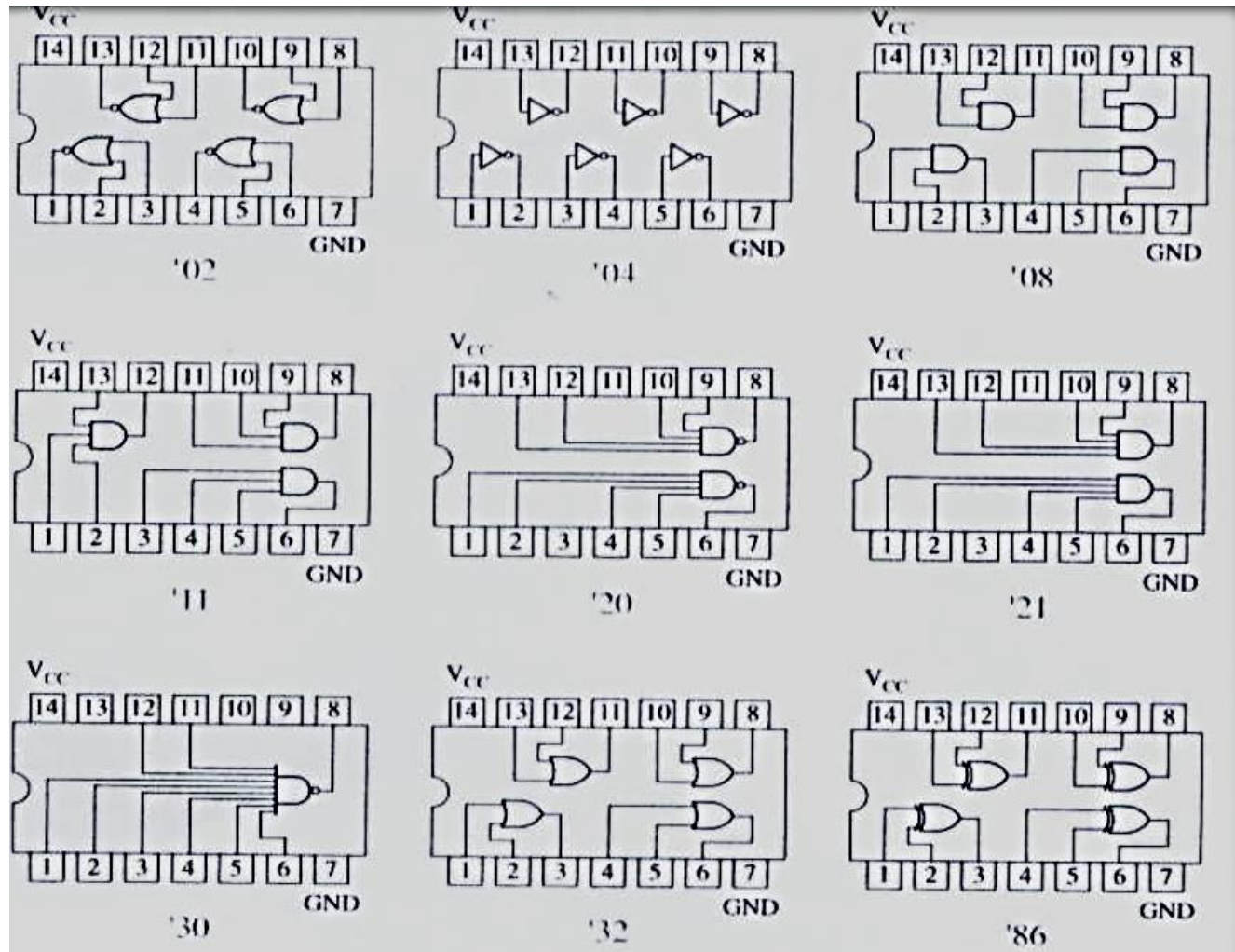
2.7. Disseny i implementació de sistemes mitjançant portes lògiques

Taula de circuits integrats més utilitzats per implementar funcions amb les portes lògiques que fins al moment hem estudiat. La majoria es troben a la **sèrie 74** del fabricant **Texas Instruments**:

Dispositivo	Descripción
7400	4 puertas NAND de 2 entradas cada una
7402	4 puertas NOR de 2 entradas cada una
7404	6 puertas NOT de 1 entrada cada una
7408	4 puertas AND de 2 entradas cada una
7432	4 puertas OR de 2 entradas cada una
7486	4 puertas XOR de 2 entradas cada una
7410	3 puertas NAND de 3 entradas cada una
7427	3 puertas NOR de 3 entradas cada una
7411	3 puertas AND de 3 entradas cada una

Tema 2. Àlgebra booleana i portes lògiques

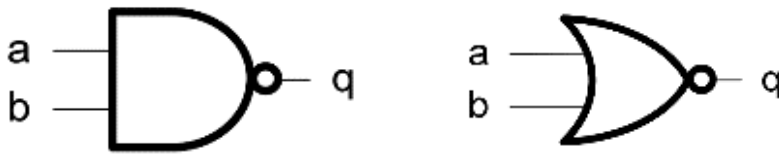
2.6. Implementació de funcions amb portes lògiques de tipus NAND o NOR




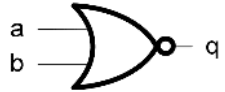
Tema 2. Àlgebra booleana i portes lògiques

2.6. Implementació de funcions amb portes lògiques de tipus NAND o NOR

- Qualsevol funció booleana es pot implementar **NOMÉS** amb combinacions de portes de tipus **NAND** o **NOR**.



Un sol tipus de porta
=
Un sol tipus de XIP

- NAND:** per aconseguir-ho, hem d'aplicar operacions algebraïques fins arribar a obtenir una expressió a on només apareguin **productes negats de dues o tres variables**.
$$\overline{a \cdot b \cdot \overline{a} \cdot c \cdot \dots}$$

- NOR:** per aconseguir-ho, hem d'aplicar operacions algebraïques fins arribar a obtenir una expressió a on només apareguin **sumes negades de dues o tres variables**.
$$\overline{a + b + \overline{a} + c + \dots}$$

- Les transformacions s'acostumen a fer mitjançant **Morgan (T-7)**.

Tema 2. Àlgebra booleana i portes lògiques

2.6. Implementació de funcions amb portes lògiques de tipus NAND o NOR

Resum dels passos:

1. Fer transformacions per obtenir la **funció simplificada**.
2. Un cop simplificada, realitzar transformacions segons el cas:
 - **NAND:** **Si** l'operació més general és una suma \rightarrow negar l'expressió sencera 2 vegades (fer la doble complementada) per tal de poder aplicar la llei de Morgan que permeti convertir la suma en un producte.

$$\overline{a + b + \overline{a + c}} \rightarrow \overline{\overline{\overline{a + b + \overline{a + c}}}} \rightarrow (\text{Morgan}) \rightarrow \overline{\overline{\overline{a + b}} \cdot \overline{\overline{\overline{a + c}}}}...$$

- **NOR:** **Si** l'operació més general és un producte \rightarrow negar l'expressió (fer la complementada) 2 vegades per tal de poder aplicar la llei de Morgan que permeti convertir la multiplicació en una suma.

$$\overline{a \cdot b \cdot \overline{a \cdot c}} \rightarrow \overline{\overline{\overline{a \cdot b \cdot \overline{a \cdot c}}}} \rightarrow (\text{Morgan}) \rightarrow \overline{\overline{\overline{a \cdot b}} + \overline{\overline{\overline{a \cdot c}}}}...$$

Tema 2. Àlgebra booleana i portes lògiques

2.6. Implementació de funcions amb portes lògiques de tipus NAND o NOR

Resum dels passos:

1. Fer transformacions per obtenir la **funció simplificada**.
2. Un cop simplificada, realitzar transformacions segons el cas:
 - **NAND:** Si l'operació més general és una suma \rightarrow negar l'expressió sencera 2 vegades (fer la doble complementada) per tal de poder aplicar la llei de Morgan que permeti convertir la suma en un producte.
 - **NOR:** Si l'operació més general és un producte \rightarrow negar l'expressió (fer la complementada) 2 vegades per tal de poder aplicar la llei de Morgan que permeti convertir la multiplicació en una suma.
3. Repetir el procés de forma iterativa fins que només quedin multiplicacions pel cas de les NANDs i sumes pel cas de les NORs.

No puc deixar cap bloc sense negar com a mínim una vegada

Per negar una variable



Tema 2. Àlgebra booleana i portes lògiques

Exemple 2.14: Implementar la següent funció mitjançant portes NAND

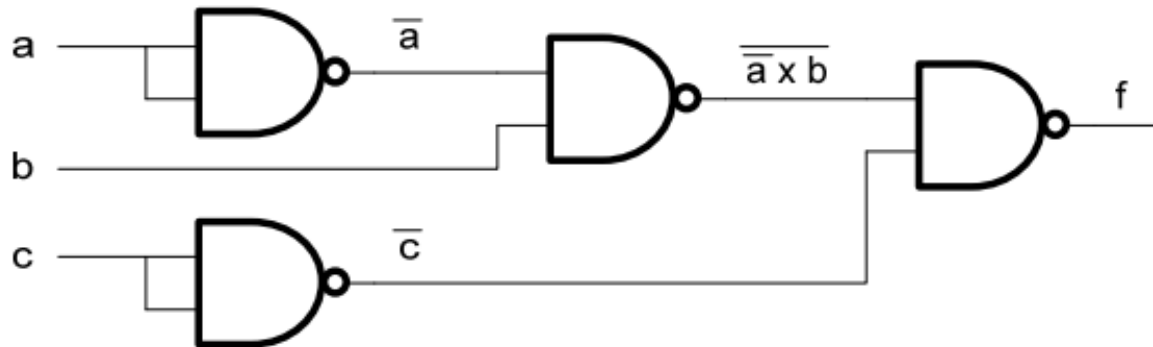
$$f(a, b, c) = \bar{a} \times b + c \quad \leftarrow \text{Suposem que ja no es pot simplificar més}$$

Pas 1. Neguem dues vegades la funció: $\overline{\overline{\bar{a} \times b} + c}$.

D'aquesta manera podem utilitzar Morgan per canviar la suma per un producte:

$$\overline{\bar{a} \times b \times \bar{c}}$$

Ara ja tenim tots els termes negats i multiplicant \leftarrow CONDICIÓN



↑ Per negar una variable ho fem connectant les dues entrades a la porta

EXERCICI 5 (30 minuts)

1. Implementar la següent funció mitjançant portes NAND

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

2. Implementar la següent funció mitjançant portes NOR

$$f(a, b, c) = a \times b + b \times \bar{c}$$

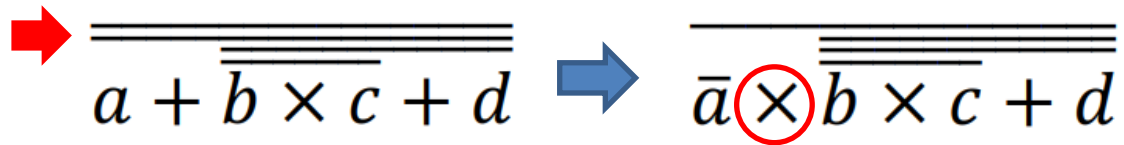
$$f(a, b, c, d) = \overline{\bar{a} \times \bar{b} + \bar{c} \times (\bar{d} \times (a + b))}$$

SOLUCIÓ

Tema 2. Àlgebra booleana i portes lògiques

1. Portes NAND $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

- Neguem dues vegades per convertir la suma en multiplicació: **CONVOLUCIÓ** + **MORGAN**


$$\overline{\overline{a + \overline{\overline{b \times c} + d}}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} + d}}$$

Tema 2. Àlgebra booleana i portes lògiques

1. Portes NAND $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

- Neguem dues vegades per convertir la suma en multiplicació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{a + \overline{\overline{b \times c} + d}}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} + d}}$$

- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{a} \times \overline{\overline{b \times c} + d}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} \times \overline{d}}}$$

Tema 2. Àlgebra booleana i portes lògiques

1. Portes NAND $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

- Neguem dues vegades per convertir la suma en multiplicació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{a + \overline{b \times c} + d}} \rightarrow \overline{\overline{a} \times \overline{b \times c} + d}$$

- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{\overline{a} \times \overline{b \times c} + d}} \rightarrow \overline{\overline{\overline{a} \times \overline{b \times c} \times \overline{d}}}$$

Si NANDs de 2 entrades \rightarrow $\overline{\overline{\overline{a} \times \overline{b \times c} \times \overline{d}}}$

Tema 2. Àlgebra booleana i portes lògiques

1. Portes NAND $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

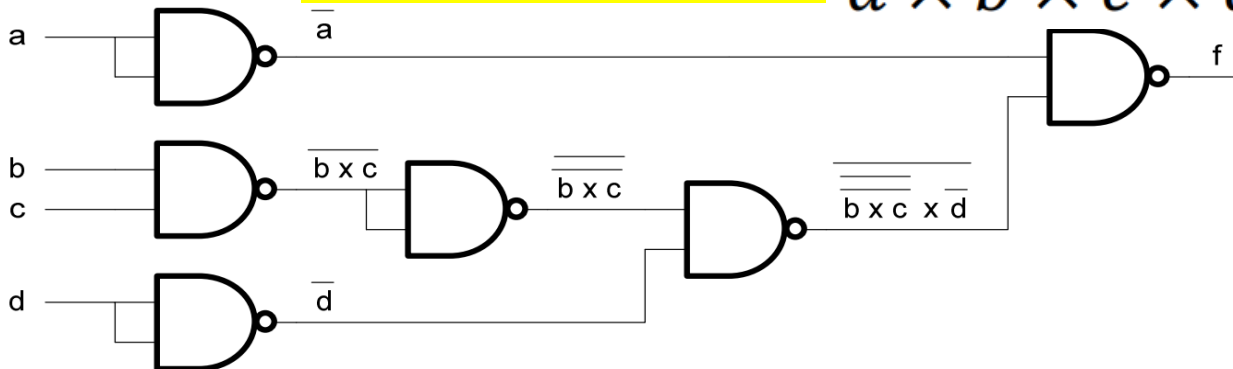
- Neguem dues vegades per convertir la suma en multiplicació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{\overline{a + \overline{b \times c} + d}}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} + d}}$$

- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{a} \times \overline{\overline{b \times c} + d}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} \times \overline{\overline{d}}}}$$

Si NANDs de 2 entrades $\rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} \times \overline{\overline{d}}}}$



Tema 2. Àlgebra booleana i portes lògiques

1. Portes NAND $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

- Neguem dues vegades per convertir la suma en multiplicació: **CONVOLUCIÓ** + **MORGAN**

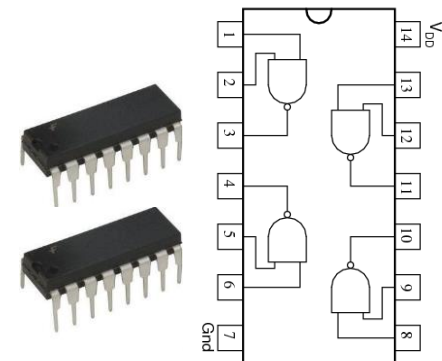
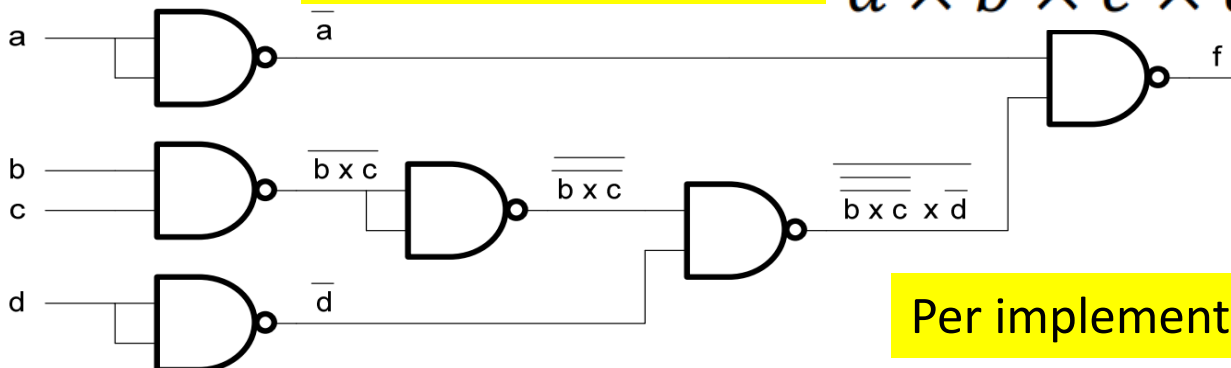
$$\overline{\overline{a + \overline{b \times c} + d}} \rightarrow \overline{\overline{a} \times \overline{\overline{b \times c} + d}}$$

- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **MORGAN**

$$\overline{\overline{a} \times \overline{b \times c} + d} \rightarrow \overline{\overline{a} \times \overline{b \times c} \times \overline{d}}$$

Si NANDs de 2 entrades \rightarrow

$$\overline{\overline{a} \times \overline{\overline{b \times c}} \times \overline{d}}$$



SN74132
(Portes NAND)

Per implementar-ho ens farien falta 2 xips

Tema 2. Àlgebra booleana i portes lògiques

1. Portes NOR $\rightarrow f(a, b, c, d) = a + \overline{\overline{b \times c} + d}$

- Quina és la primera operació que hem de fer?
- Què és el que volem?

Passos bàsics (no els únics):

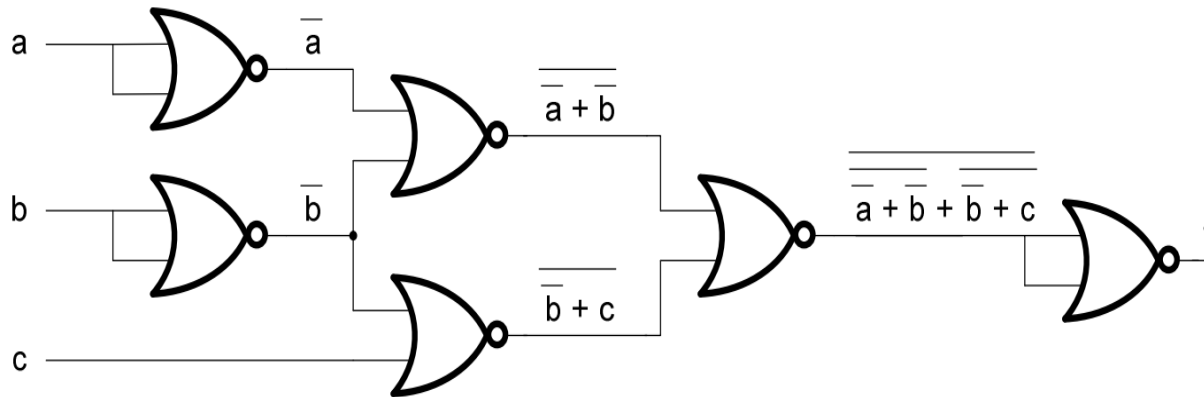
1. Trobar l'operador més general a canviar.
2. Aplicar Morgan segons convingui.
3. Iterar fins arribar al context més profund.

Tema 2. Àlgebra booleana i portes lògiques

2. Portes NOR $\rightarrow f(a, b, c) = a \times b + b \times \bar{c}$

- Neguem dues vegades per convertir les multiplicacions en suma: **CONV.** + **MORGAN**

$$\overline{\overline{a \times b + b \times \bar{c}}} \rightarrow \overline{\overline{\bar{a} + \bar{b} + \bar{b} + \bar{c}}}$$



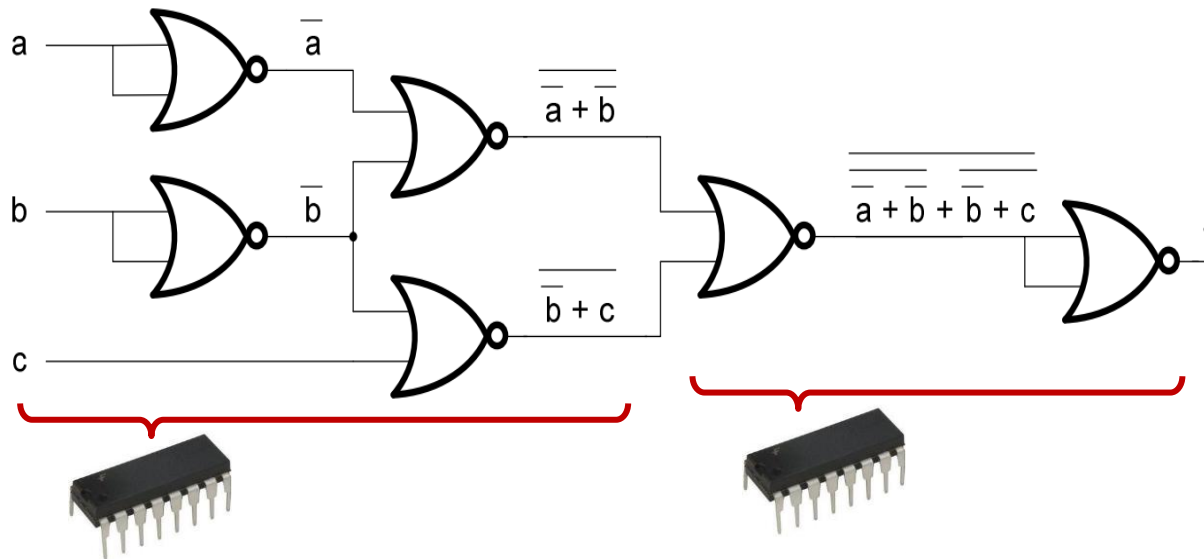
Ho podíem haver fet millor ?

Tema 2. Àlgebra booleana i portes lògiques

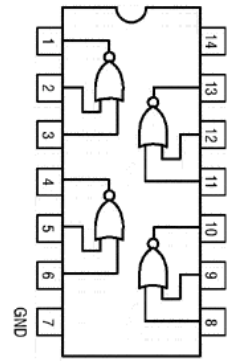
2. Portes NOR $\rightarrow f(a, b, c) = a \times b + b \times \bar{c}$

- Neguem dues vegades per convertir les multiplicacions en suma: **CONV.** + **MORGAN**

$$\overline{\overline{a \times b + b \times \bar{c}}} \rightarrow \overline{\overline{\bar{a} + \bar{b} + \bar{b} + \bar{c}}}$$



SN7402
(Portes NOR)



Sí \rightarrow No hem simplificat

Tema 2. Àlgebra booleana i portes lògiques

2. Portes NOR $\rightarrow f(a, b, c) = a \times b + b \times \bar{c}$

- Neguem dues vegades per convertir les multiplicacions en suma: **CONV.** + **MORGAN**

$$\overline{\overline{a \times b + b \times \bar{c}}} \rightarrow \overline{\overline{a} + \overline{b} + \overline{b} + \overline{\bar{c}}}$$

- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **ORGAN**

$$a \times \textcircled{b} + \textcircled{b} \times \bar{c} = b \times (a + \bar{c}) \quad \leftarrow \text{aplicant P3 + Convolució}$$

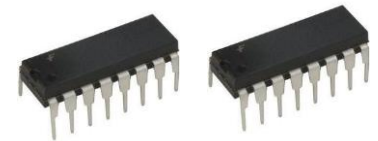
$$= \overline{\overline{b \times (a + \bar{c})}} = \overline{\overline{b}} + \overline{a + \bar{c}}$$

Tema 2. Àlgebra booleana i portes lògiques

2. Portes NOR $\rightarrow f(a, b, c) = a \times b + b \times \bar{c}$

- Neguem dues vegades per convertir les multiplicacions en suma: **CONV.** + **MORGAN**

$$\overline{\overline{a \times b + b \times \bar{c}}} \rightarrow \overline{\overline{\bar{a} + \bar{b} + \bar{b} + \bar{\bar{c}}}}$$



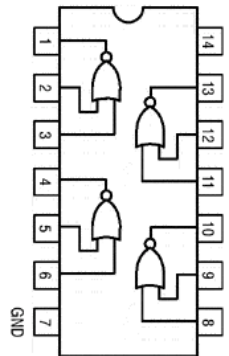
- Operem el terme de dins de la doble negació: **CONVOLUCIÓ** + **ORGAN**

$$a \times b + b \times \bar{c} = b \times (a + \bar{c}) \quad \leftarrow \text{aplicant P3 + Convolució}$$

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



SN7402
(Portes NOR)



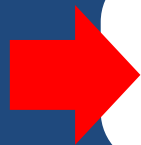
EXERCICI 5 (30 minuts)

1. Implementar la següent funció mitjançant portes NAND

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

2. Implementar la següent funció mitjançant portes NOR

$$f(a, b, c) = a \times b + b \times \bar{c}$$



$$f(a, b, c, d) = \overline{\bar{a} \times \bar{b} + \bar{c} \times (\bar{d} \times (a + b))}$$

Tema 2. Àlgebra booleana i portes lògiques

3. Portes NOR $\rightarrow f(a, b, c, d) = \overline{\bar{a} \times \bar{b} + \bar{c} \times (\bar{d} \times (a + b))}$

Tema 2. Àlgebra booleana i portes lògiques

3. Portes NOR $\rightarrow f(a, b, c, d) = \overline{\bar{a} \times \bar{b} + \bar{c} \times (\bar{d} \times (a + b))}$

- Doble neguem el primer terme amb multiplicació:

$$\begin{array}{ccc} \overline{\underbrace{\bar{a} \times \bar{b}}_{1} + \underbrace{\bar{c} \times (\bar{d} \times (a + b))}_{2}} & \xrightarrow{\quad} & \overline{\underbrace{\bar{a} \times \bar{b}}_{1} + \bar{c} \times (\bar{d} \times (a + b))} \\ & & \downarrow \\ \overline{(a + b) + \underbrace{\bar{c} \times (\bar{d} \times (a + b))}_{2}} & \xleftarrow{\quad} & \overline{(a + b) + \underbrace{\bar{c} \times (\bar{d} \times (a + b))}_{2}} \\ & & \downarrow \\ \overline{(a + b) + c + \underbrace{(\bar{d} \times (a + b))}_{3}} & \xrightarrow{\quad} & \overline{(a + b) + c + \underbrace{d + a + b}_{3}} \end{array}$$

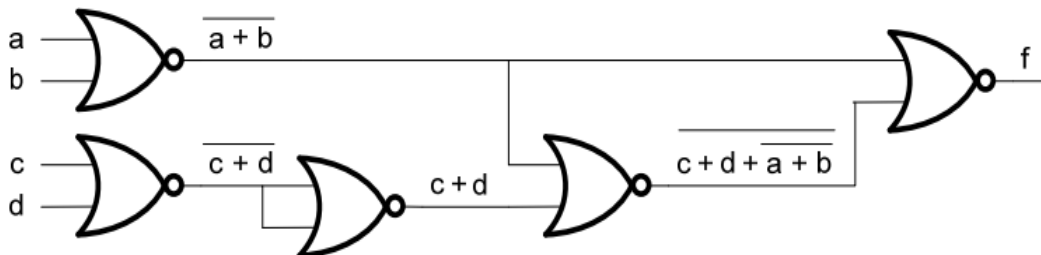
Tema 2. Àlgebra booleana i portes lògiques

3. Portes NOR $\rightarrow f(a, b, c, d) = \overline{\overline{a} \times \overline{b} + \overline{c} \times (\overline{d} \times (a + b))}$

- **Doble neguem el primer terme amb multiplicació:**

The diagram illustrates the simplification of the Boolean expression $(a+b)c + d(a+b)$ through several steps:

- Initial Expression:** $(a+b)c + d(a+b)$
- Step 1:** Apply De Morgan's law to $(a+b)c$: $\overline{\overline{a+b} \times \overline{c}}$
- Step 2:** Apply De Morgan's law to $\overline{a+b}$: $\overline{\overline{a} \times \overline{b}}$
- Step 3:** Apply De Morgan's law to \overline{c} : $\overline{\overline{a} \times \overline{b} + c}$
- Step 4:** Apply De Morgan's law to $d(a+b)$: $\overline{\overline{d} \times \overline{a+b}}$
- Step 5:** Apply De Morgan's law to $\overline{a+b}$: $\overline{\overline{a} \times \overline{b} + c + \overline{d}}$
- Step 6:** Apply De Morgan's law to \overline{d} : $\overline{\overline{a} \times \overline{b} + c + d}$
- Step 7:** Apply De Morgan's law to $\overline{\overline{a} \times \overline{b} + c + d}$: $\overline{\overline{a} \times \overline{b}} \times \overline{c} \times \overline{d}$
- Step 8:** Apply De Morgan's law to $\overline{\overline{a} \times \overline{b}}$: $(a+b) \times \overline{c} \times \overline{d}$
- Step 9:** Apply De Morgan's law to $\overline{c} \times \overline{d}$: $(a+b) + c + d$
- Step 10:** Apply De Morgan's law to $(a+b) + c + d$: $\overline{\overline{a+b} \times \overline{c} \times \overline{d}}$
- Step 11:** Apply De Morgan's law to $\overline{a+b}$: $\overline{\overline{a} \times \overline{b} \times \overline{c} \times \overline{d}}$
- Step 12:** Apply De Morgan's law to $\overline{c} \times \overline{d}$: $\overline{\overline{a} \times \overline{b} + c + d}$
- Step 13:** Apply De Morgan's law to $\overline{a} \times \overline{b}$: $\overline{\overline{a+b} \times \overline{c} \times \overline{d}}$
- Step 14:** Apply De Morgan's law to $\overline{a+b}$: $\overline{\overline{a} \times \overline{b} \times \overline{c} \times \overline{d}}$
- Step 15:** Apply De Morgan's law to $\overline{a} \times \overline{b} \times \overline{c} \times \overline{d}$: $a + b + c + d$



Per defecte, la implementació és amb portes NOR de dos entrades, però ens ho haurien d'especificar

Tema 2. Àlgebra booleana i portes lògiques

Errades: Procediment → MORGAN

1. Respectar el { context } → Parèntesis
2. Mantenir l'ordre de les operacions.
3. Dos negats s'anul·len, però si ens demanen una implementació amb **NANDs** o **NORs**, potser ens interressi conservar-los o afegir-los.

$$\bar{a} \times \overline{\overline{b \times c}} + d$$

$$\overline{\overline{b \times c}} + d$$

$$\overline{\overline{b \times c}} \quad ?$$

→ Una operació a cada pas!

Errades: Implementació

- Implementació només amb les portes indicades, en aquest cas Portes NANDs or NORs (no es poden usar inversores).

PROBLEMES DE DISSENY

Tema 2. Àlgebra booleana i portes lògiques

Problemes de disseny

En la vida real ens demanaran solucions a una necessitat i no que implementem una funció algebraica, per tant, hem de saber com afrontar el desenvolupament de problemes reals.

PAUTES

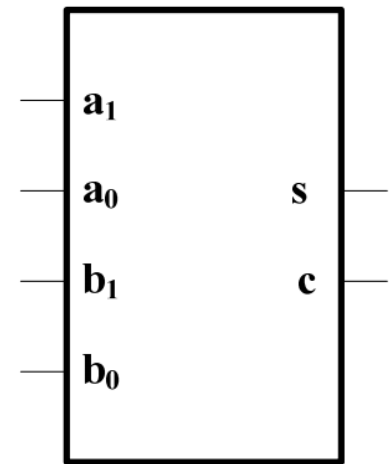
1. **Identificació de la funció booleana** que resol el problema plantejat. En general aquesta funció s'expressa mitjançant **taula de veritat**, en tant que per a cada combinació de les variables d'entrada hem d'anar "raonant" el valor que ha de prendre la sortida de la funció segons els requisits indicats en l'enunciat.
2. **Obtenció de la funció booleana simplificada** (o adaptada al tipus d'implementació a partir d'aquesta: portes NAND, NOR).
3. **Implementació de la funció amb portes lògiques**. Aquesta pot ser, o usant portes NAND o NOR, o bé, fent servir qualsevol tipus de porta lògica.

Tema 2. Àlgebra booleana i portes lògiques

Problemes de disseny

EXEMPLE 1

- Volem dissenyar un sistema que compari 2 nombres digitals.
- El sistema tindrà **2 entrades** 'a' i 'b', cadascuna de **2 bits**, de manera que realment tindrem 4 variables d'entrada **a1, a2, b1, i b2**. El sistema tindrà 2 sortides anomenades 's' i 'c'.
- Quan les dues entrades 'a' i 'b' presentin el mateix valor, és a dir que $a_0 = b_0$ i $a_1 = b_1$ llavors activarem $\rightarrow s = 1$ i $c = 0$.
- En canvi, si són diferents llavors activarem $s = 0$ i $c = 1$ quan el valor binari de 'a' sigui més gran que el de 'b', o bé, les sortides valdran $s=0$ i $c=0$ quan el valor binari de 'a' sigui menor que el de 'b'.



a) Obtenir la expressió algebraica per mitjà de MINTERMS

b) Un cop obtingudes, implementar-les per mitjà de portes lògiques

Tema 2. Àlgebra booleana i portes lògiques

Problemes de disseny

EXEMPLE 1

PAS 1: Construir la taula de la veritat del sistema.

a_1	a_0	b_1	b_0	s	c
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

PAS 2: Obtenir la expressió algebraica per mitjà de MINTERMS

$$s = \overline{a_1} \times \overline{a_0} \times \overline{b_1} \times \overline{b_0} + \\ \overline{a_1} \times a_0 \times \overline{b_1} \times b_0 + \\ a_1 \times \overline{a_0} \times b_1 \times \overline{b_0} + \\ a_1 \times a_0 \times b_1 \times b_0$$

$$c = \overline{a_1} \times a_0 \times \overline{b_1} \times \overline{b_0} + a_1 \times \overline{a_0} \times \overline{b_1} \times \overline{b_0} + \\ a_1 \times \overline{a_0} \times \overline{b_1} \times b_0 + a_1 \times a_0 \times \overline{b_1} \times \overline{b_0} + \\ a_1 \times a_0 \times \overline{b_1} \times b_0 + a_1 \times a_0 \times b_1 \times \overline{b_0}$$

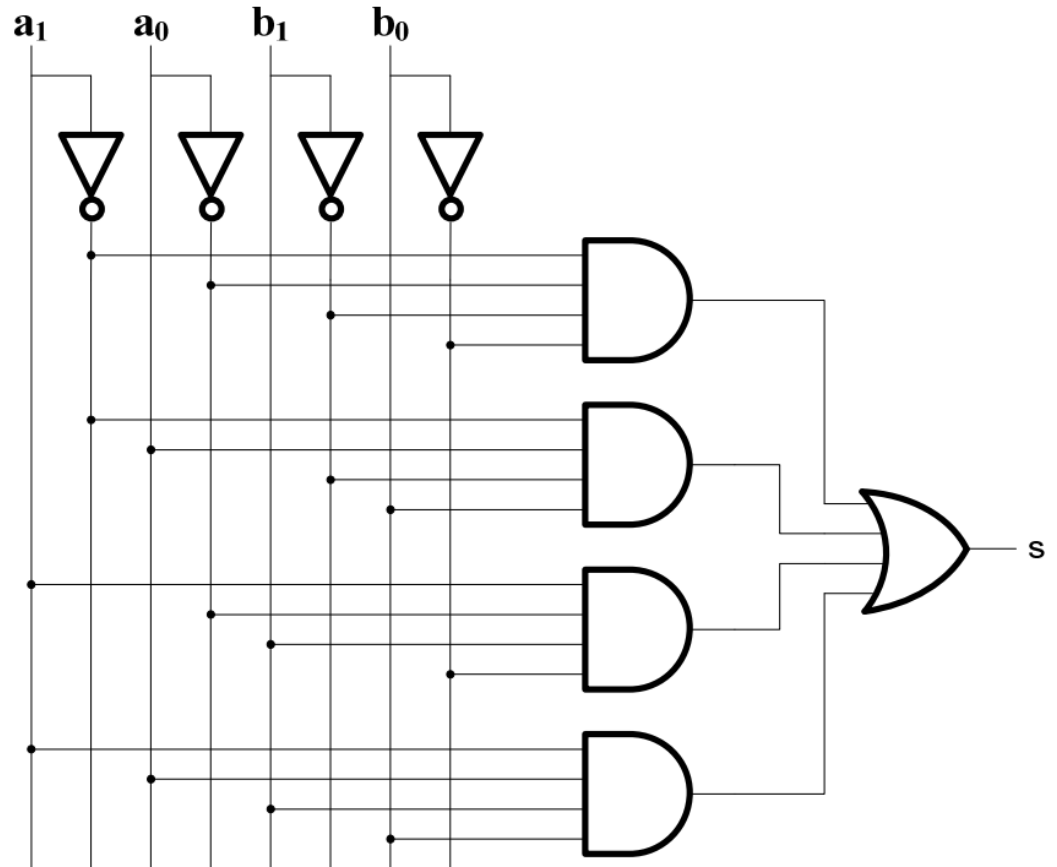
PAS 3: Un cop obtingudes, implementar-les per mitjà de portes lògiques

Tema 2. Àlgebra booleana i portes lògiques

Problemes de disseny

EXEMPLE 1

PAS 3: Implementació de 's':



VHDL

Què és el VHDL?

Problemàtica:

- Necessitat d'un mètode per dissenyar sistemes complexos de forma ràpida.
- Esquemes i equacions booleanes són insuficients per circuits amb milions de portes.

Solució:

- Un llenguatge de descripció de hardware (HDL) per expressar el disseny.
- Dispositius lògics programables per una ràpida implementació de hardware.

Dos llenguatges HDLs utilitzats en l'actualitat:

- **VHDL** (Very High Speed Integrated Circuit HDL) (estàndard IEEE des de 1987).
- **Verilog HDL** (Cadence Design Systems, estàndard IEEE en l'actualitat).

Què és el VHDL?

Plantilla genèrica VHDL

- **Llibreries**
- **Entitat:** defineix els ports de connexió amb l'exterior (entrades i sortides).
- **Arquitectura:** descriu el funcionament d'una entitat.

```
library <library_name>;  
use <library_name>.<package_name>.<object_name>;
```

```
entity <entity_name> is  
    port (  
        -- Input ports  
        <name> :in <type>  
        <name> :in <type>  
        ...  
        -- Output ports  
        <name> :out <type>  
        <name> :out <type>  
        ...  
    );  
end <entity_name>
```

```
architecture <arch_name> of <entity_name> is  
begin  
    Concurrent Statement;  
    Concurrent Statement;  
    Concurrent Statement;  
    ...  
end <arch_name>
```

Què és el VHDL?

Plantilla genèrica VHDL

Entitat

- Es podria entendre com la **caixa negra** que conté el sistema que es vol implementar, amb les seves entrades i sortides.
- Parts: entity <entity_name> is -- Nom de la entitat

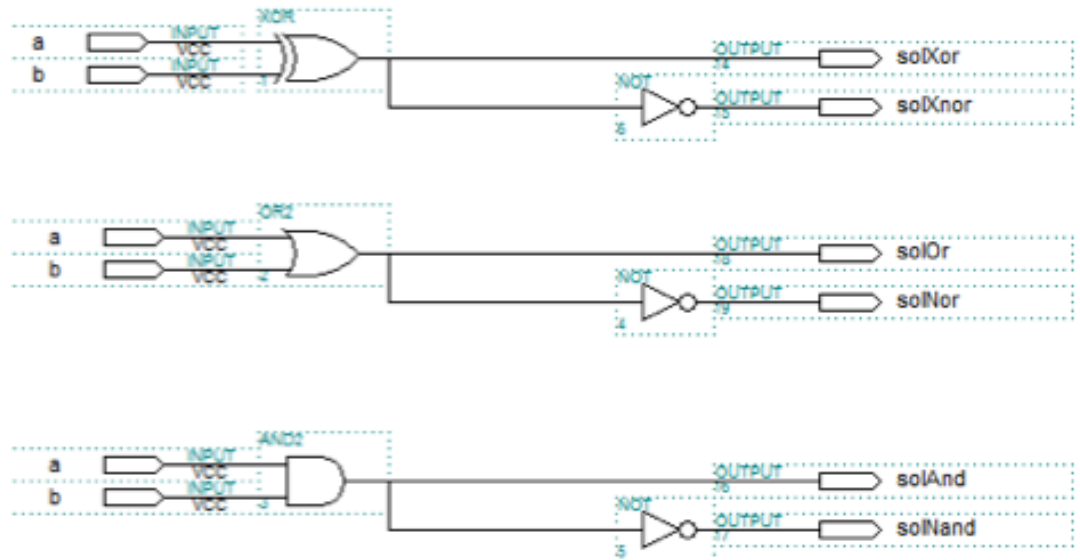
```
port(  
    -- Input ports  
    a, b : in std_logic;  
  
    -- Output ports  
    solAnd   : out std_logic;  
    solOr    : out std_logic;  
    solNand  : out std_logic;  
    solNor   : out std_logic;  
    solXor   : out std_logic;  
    solNxor  : out std_logic  
);
```

Què és el VHDL?

```
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;
```

```
entity myGate is  
  port(  
    -- Input ports  
    a, b : in std_logic;  
  
    -- Output ports  
    solAnd : out std_logic;  
    solOr : out std_logic;  
    solNand : out std_logic;  
    solNor : out std_logic;  
    solXor : out std_logic;  
    solNxor : out std_logic  
  );  
end myGate;  
  
architecture solGate of myGate is  
  begin  
    solAnd <= a and b;  
    solNand <= not (a and b);  
    solOr <= a or b;  
    solNor <= not (a or b);  
    solXor <= a xor b;  
    solNxor <= not (a xor b);  
  end solGate;
```

→ Aquest codi equivaldria amb el següent circuit



Què és el VHDL?

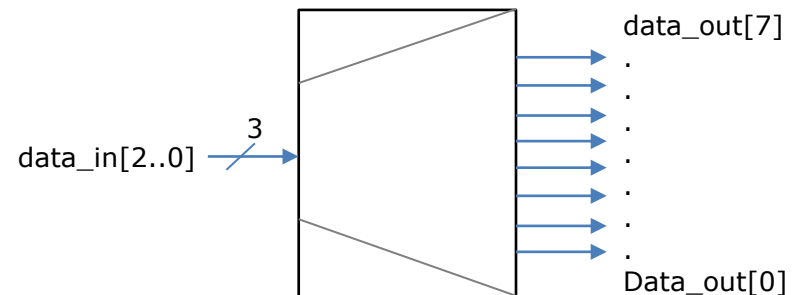
Programació d'un descodificador

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity myDecode is
    port(
        data_in : in std_logic_vector(2 downto 0);
        data_out : out std_logic_vector(7 downto 0)
    );
end myDecode;

architecture solDecode of myDecode is
begin
    data_out <=
        "00000001" when data_in="000" else
        "00000010" when data_in="001" else
        "00000100" when data_in="010" else
        "00001000" when data_in="011" else
        "00010000" when data_in="100" else
        "00100000" when data_in="101" else
        "01000000" when data_in="110" else
        "10000000" when data_in="111";
end solDecode;
```

data_in		data_out
000		00000001
001		00000010
010		00000100
011		00001000
100		00010000
101		00100000
110		01000000
111		10000000

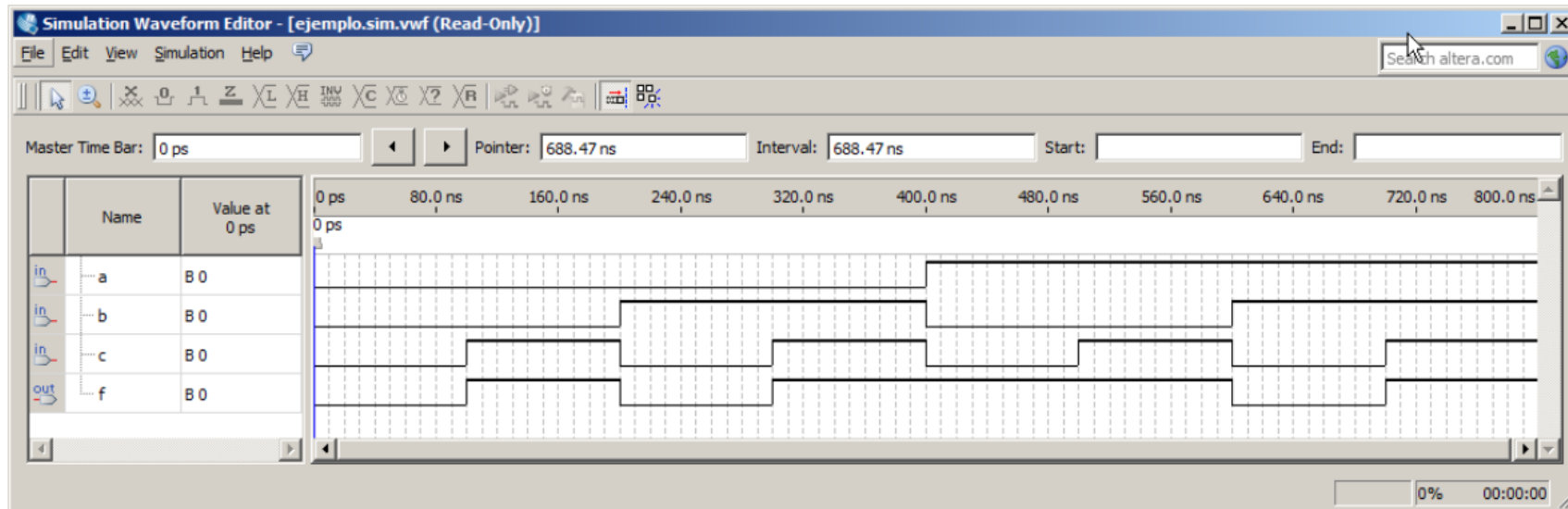
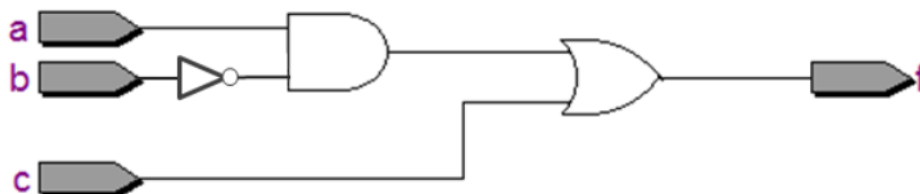


Sessió *QUARTUS*

Tabla de la verdad:

<u>a</u>	<u>b</u>	<u>c</u>	<u>f</u>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

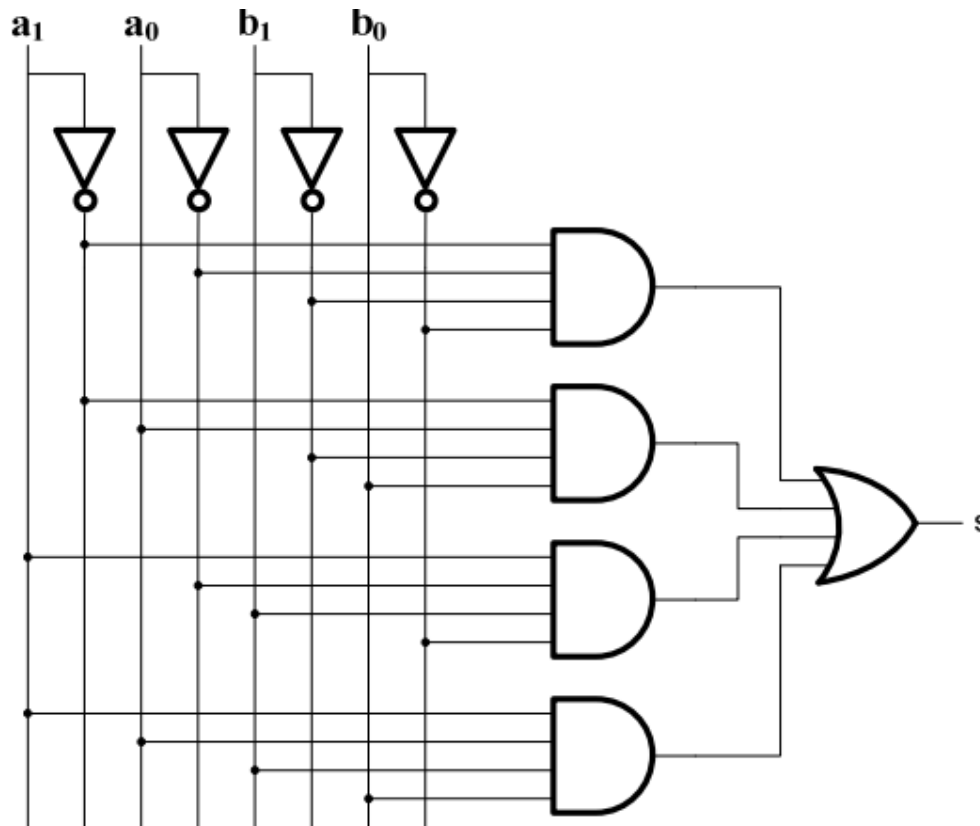
$$f = (a \cdot \bar{b}) + c$$



EXERICI QUARTUS: Disseny en esquemàtic + simulació del següent circuit (A Entregar)

Entregar tot el contingut de dins la carpeta del projecte en un zip.

Exemple: c:\projectes\EX02\ → ZIP



a_1	a_0	b_1	b_0	s
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Aspecte d'una carpeta de projecte del QUARTUS

Nombre	Fecha de modificación	Tipo	Tamaño
db	13/10/2021 12:59	Carpeta de archivos	
incremental_db	07/10/2020 9:00	Carpeta de archivos	
output_files	06/10/2021 9:31	Carpeta de archivos	
simulation	07/10/2020 9:12	Carpeta de archivos	
c5_pin_model_dump.txt	07/10/2020 9:01	Documento de texto	5 KB
PR01.bdf	06/10/2021 9:21	Archivo de texto	4 KB
PR01.bsf	05/10/2021 21:29	Archivo de texto	2 KB
PR01.qpf	07/10/2020 8:59	Archivo de texto	2 KB
PR01.qsf	06/10/2021 9:45	Archivo de texto	3 KB
PR01.qws	13/10/2021 12:59	Archivo de texto	1 KB
Waveform.vwf	07/10/2020 9:12	Archivo VWF	5 KB
Waveform1.vwf	05/10/2021 21:39	Archivo VWF	5 KB
Waveform2.vwf	06/10/2021 9:31	Archivo VWF	5 KB

