ESTRUCTURA i TECNOLOGIA de COMPUTADORS

Transparències del Tema 4
Sistemes Digitals Combinacionals

J. Freixenet, X. Cufí, J. Martí,
M. Fàbregas i J. Ferrer
Departament d'Electrònica,
Informàtica i Automàtica

ETC - Esquema del tema 4

Sistemes combinacionals
- Disseny d'un sistema combinacional
- Sumadors
  - Semi Sumador *(Half Adder)*
  - Sumador Complert (*Full Adder*)
  - Sumadors d'*n* bits
- Comparador de 2 bits
- Multiplexors
- Decodificadors (*Decoders*)
- Memòries ROM

ETC - Disseny de Sistemes Combinacionals

PAS 1: Representar el problema que es vol resoldre en una ***Taula de Veritat***, és a dir, fer un estudi del comportament d'una funció cas per cas.

Ex: **Half Adder**
Sumador de dos bits

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ex: **Full Adder**
Sumador de dos bits amb *Carry d'Entrada*

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

ATENCIÓ:

La suma utilitzada és 1 + 1 = 0 i en portem una (Bit de Carry = 1).
Aquesta és la **suma** de dos números **en Binari Natural**. **No** és **la suma (*OR*)** de l'**Algebra de Boole.**

ETC - Disseny de Sistemes Combinacionals

PAS 2: ***Simplificar***. Volem escriure les funcions de la forma més simple possible per tal d'estalviar temps i diners a l'hora d'implementar-les amb portes.

Maneres de simplificar: utilitzant els teoremes de l'***Àlgebra de Boole***, utilitzant el mètode tabular de ***Karnaugh***, o bé intentar extreure les equacions directament de la ***Taula de la Veritat***.

A	B	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum = $\overline{A}\,B + A\,\overline{B}$

Carry = A B

RECORDATORI:

NOT X es pot escriure com $\overline{X}$
X AND Y s'escriu com X & Y, X · Y o també X Y
X OR Y s'escriu com X + Y

ETC - Disseny de Sistemes Combinacionals

Obtenció de les funcions a partir de la **Taula de Veritat** d'un **Full Adder**.

$$\textbf{Sum} = \bar{A}\,\bar{B}\,Cin + \bar{A}\,B\,\overline{Cin} + A\,\bar{B}\,\overline{Cin} + A\,B\,Cin$$

A	**B**	**Cin**	**Sum**	**Cout**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\textbf{Cout} = \bar{A}\,B\,Cin + A\,\bar{B}\,Cin + A\,B\,\overline{Cin} + A\,B\,Cin$$

ETC - Disseny de Sistemes Combinacionals

Simplificació de les funcions a partir de la **Taula de Veritat** del **Carry Out** d'un **Full Adder.**

$$\textbf{Cout} = A\,Cin + B\,Cin + A\,B$$

B Cin

A Cin

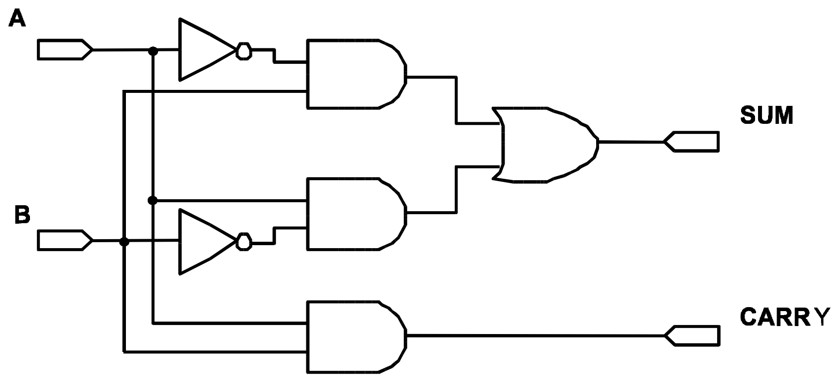
A B

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

ETC - Disseny de Sistemes Combinacionals

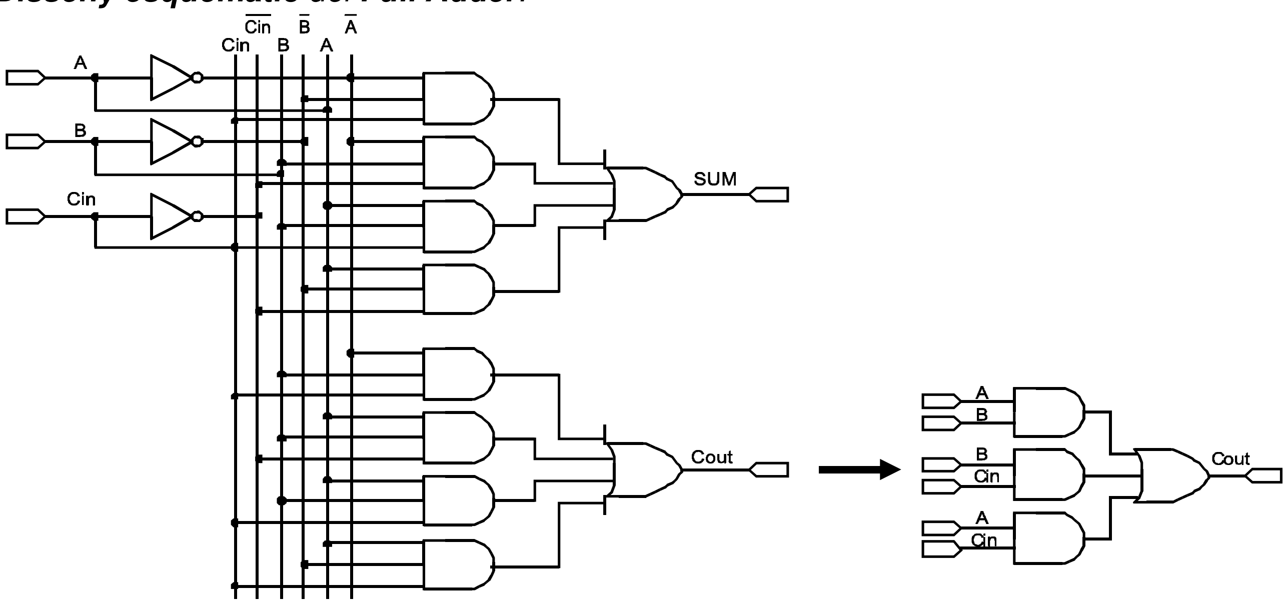
***PAS 3**: **Implementar**, és a dir representar en portes la funció simplificada.*

Disseny esquemàtic** del **Half Adder:



ETC - Disseny de Sistemes Combinacionals

***Disseny esquemàtic** del **Full Adder**:*

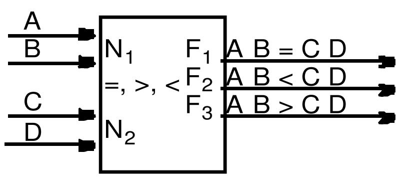


<u>CAL TENIR EN COMPTE</u>:

Temps de propagació.
***Fan - in** i **Fan - out**:* El consum expressat amb número de portes.
Dissenys simplificats = **Menys Cost en temps i diners**

ETC - Disseny de Sistemes Combinacionals

***Exemple de disseny - Comparador de dos bits**: Taula de la Veritat*



A	B	C	D	F_1	F_2	F_3
0	0	0	0	1	0	0
		0	1	0	1	0
		1	0	0	1	0
		1	1	0	1	0
0	1	0	0	0	0	1
		0	1	1	0	0
		1	0	0	1	0
		1	1	0	1	0
1	0	0	0	0	0	1
		0	1	0	0	1
		1	0	1	0	0
		1	1	0	1	0
1	1	0	0	0	0	1
		0	1	0	0	1
		1	0	0	0	1
		1	1	1	0	0

ETC - Disseny de Sistemes Combinacionals

***Exemple de disseny - Comparador de dos bits:** Simplificació I (utilitzant Karnaugh)*

CD \ AB	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

K-map per F_1

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

K-map per F_2

CD \ AB	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

K-map per F_3

$\mathbf{F_1}$ =

$\mathbf{F_2}$ =

$\mathbf{F_3}$ =

Exemple de disseny - Comparador de dos bits: *Simplificació II (utilitzant Karnaugh)*

K-map per F_1

CD \ AB	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

K-map per F_2

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

K-map per F_3

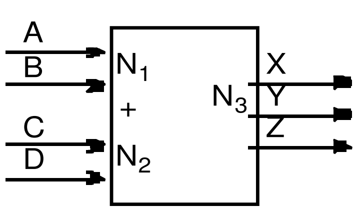
CD \ AB	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$$\mathbf{F_1} = A'\,B'\,C'\,D' + A'\,B\,C'\,D + A\,B\,C\,D + A\,B'\,C\,D'$$

$$\mathbf{F_2} = A'\,B'\,D + A'\,C + B'\,C\,D$$

$$\mathbf{F_3} = B\,C'\,D' + A\,C' + A\,B\,D'$$

Exemple de disseny - Sumador de dos bits*: Taula de la Veritat*



A	B	C	D	X	Y	Z
0	0	0	0	0	0	0
		0	1	0	0	1
		1	0	0	1	0
		1	1	0	1	1
0	1	0	0	0	0	1
		0	1	0	1	0
		1	0	0	1	1
		1	1	1	0	0
1	0	0	0	0	1	0
		0	1	0	1	1
		1	0	1	0	0
		1	1	1	0	1
1	1	0	0	0	1	1
		0	1	1	0	0
		1	0	1	0	1
		1	1	1	1	0

ETC - Disseny de Sistemes Combinacionals

***Exemple de disseny - Sumador de dos bits**: Simplificació I (utilitzant Karnaugh)*

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

K-map per X

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

K-map per Y

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

K-map per Z

X =

Z =

Y =

ETC - Disseny de Sistemes Combinacionals

***Exemple de disseny - Sumador de dos bits**: Simplificació II (utilitzant Karnaugh)*

CD \ AB	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

K-map per X

CD \ AB	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

K-map per Y

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

K-map per Z

1's en diagonal suggereixen XORs!

Pot ser interessant per reduir el número de portes

X = A C + B C D + A B D

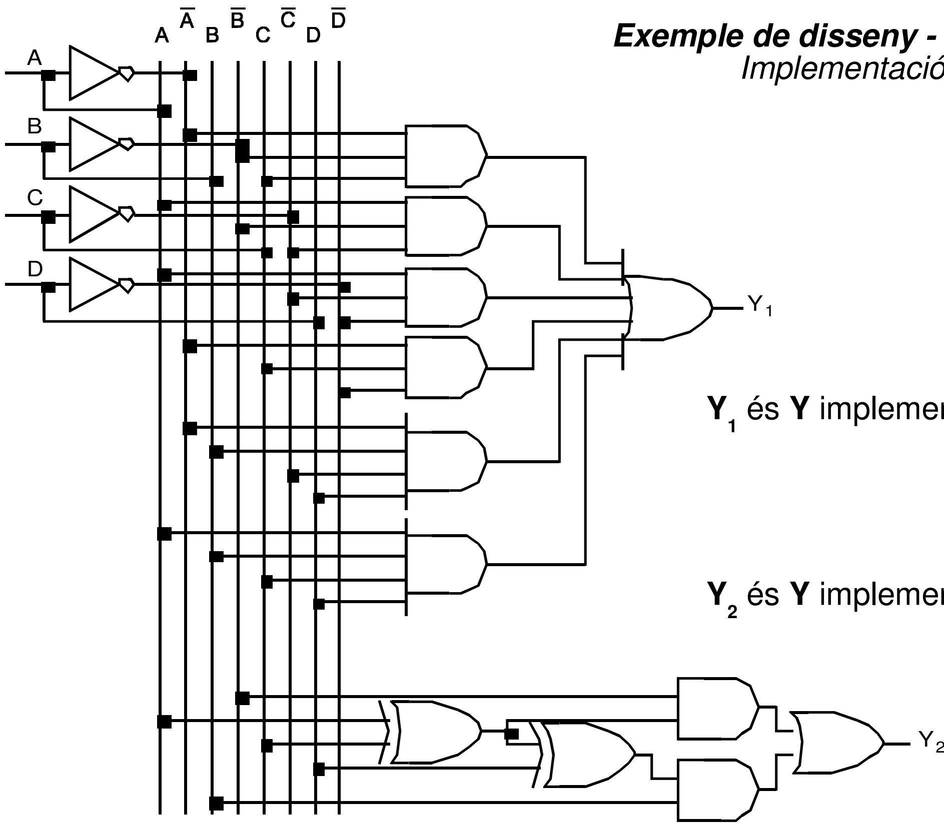
Z = B D' + B' D = B ⊕D

Y = A' B' C + A B' C' + A' B C' D + A' B C D' + A B C' D' + A B C D

= B' (A ⊕C) + A' B (C ⊕D) + A B (C ⊕D)'

= B' (A ⊕C) + B (A ⊕B ⊕C)

***Exemple de disseny - Sumador de dos bits**:*
*Implementació amb portes d'**Y***



$\mathbf{Y_1}$ és **Y** implementada sense utilitzar **XOR**s

$\mathbf{Y_2}$ és **Y** implementada utilitzant **XOR**s

Multiplexor (I)

Sistema digital combinacional amb:

- $\mathbf{2^n}$ Entrades, **n** Bits de Control i **1** Sortida
- S'utilitza per connectar $\mathbf{2^n}$ camins origen a un sol punt
- També es poden utilitzar per implementar funcions

Exemple amb **n = 1**:

$\mathbf{Z} = A'\,I_0 + A\,I_1$

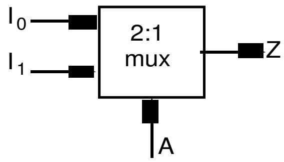
Forma funcional

A	Z
0	I_0
1	I_1

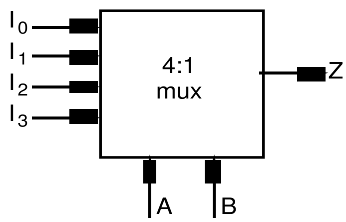
Forma Lògica

I_1	I_0	A	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

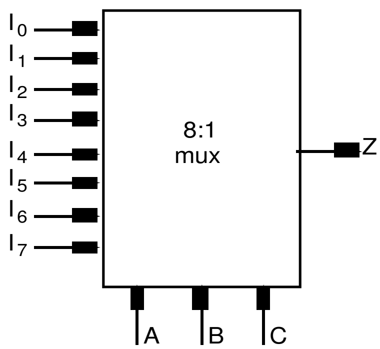
Multiplexor (II)



$\mathbf{Z} = A' I_0 + A I_1$



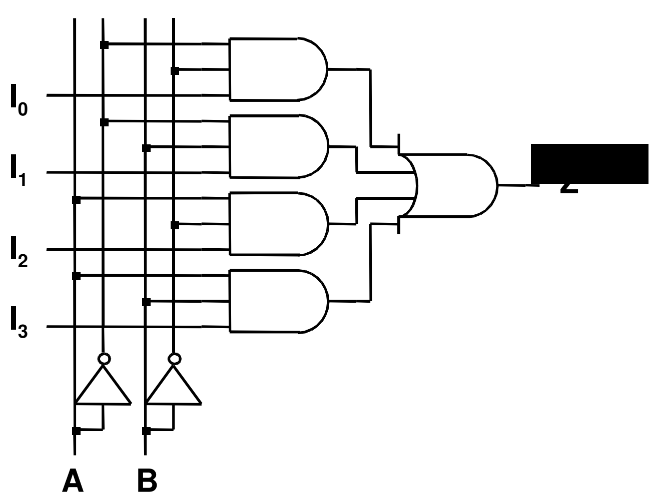
$\mathbf{Z} = A' B' I_0 + A' B I_1 + A B' I_2 + A B I_3$



$$\mathbf{Z} = A' B' C' I_0 + A' B' C I_1 + A' B C' I_2 + A' B C I_3 + A B' C' I_4 + A B' C I_5 + A B C' I_6 + A B C I_7$$

Multiplexor (III)

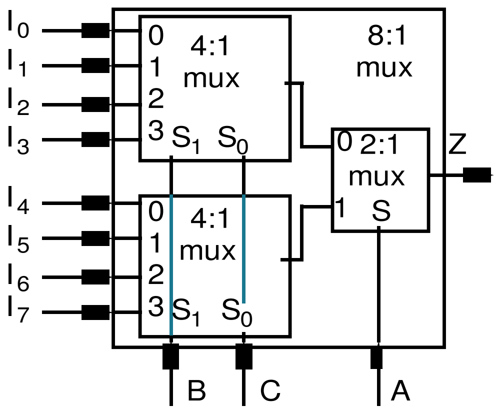
Implementació amb portes d'un **multiplexor** de 4 Entrades (2^2) i **n** = 2 Bits de Control

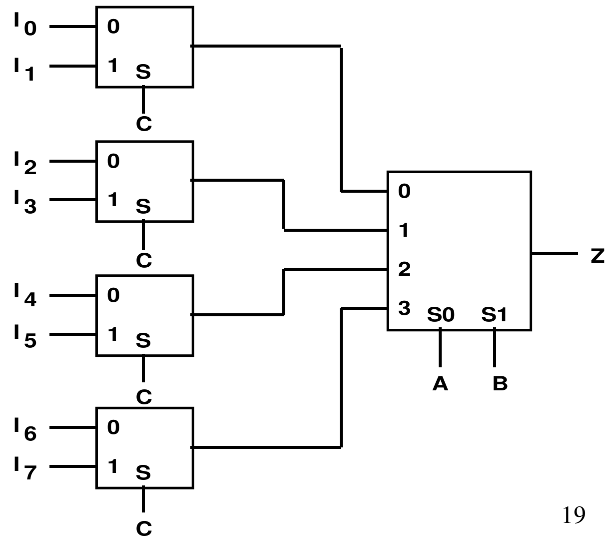


Multiplexor (IV)

Es poden obtenir *multiplexors grans* (de més entrades) a partir de *multiplexors més petits*

Exemple: Implementació d'un **multiplexor 8:1** amb **multiplexors 4:1 i 2:1**





Multiplexor (IV)

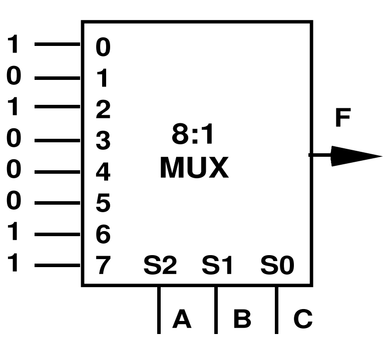
Implementar **funcions lògiques** utilitzant **multiplexors**

Exemple: Implementació de **F(A,B,C) = m0 + m2 + m6 + m7** utilitzant **multiplexors**

F(A,B,C) = A' B' C' + A' B C' + A B C' + A B C (**Opció 1**)

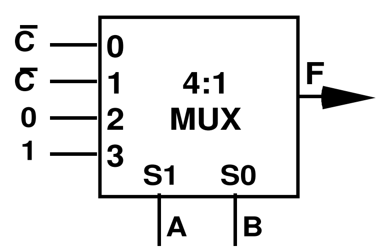
= A' B' (C') + A' B (C') + A B' (0) + A B (1) (**Opció 2**)

Opció 1: LookUp Table (*LUT*)



Opció 2:

A	B	C	F	
0	0	0	1	$\overline{C}$
0	0	1	0	
0	1	0	1	$\overline{C}$
0	1	1	0	
1	0	0	0	0
1	0	1	0	
1	1	0	1	1
1	1	1	1	



ETC - Disseny de Sistemes Combinacionals

Descodificador (I)

Sistema digital combinacional amb:

- **n** Entrades de Control, $\mathbf{2^n}$ Sortides
- Entrada d'activació que sol anomenar-se *Enable* (G)
- Els Bits de Control (Selectors S) representen la sortida que s'ha d'activar si l'entrada d'*Enable* està activada.

Exemples amb **n = 1, 2 i 3**:

1 : 2 Decoder:

$O_0 = G \bullet \overline{S}$

$O_1 = G \bullet S$

2 : 4 Decoder:

$O_0 = G \bullet \overline{S_0} \bullet \overline{S_1}$

$O_1 = G \bullet \overline{S_0} \bullet S_1$

$O_2 = G \bullet S_0 \bullet \overline{S_1}$

$O_3 = G \bullet S_0 \bullet S_1$

3 : 8 Decoder:

$O_0 = G \bullet \overline{S_0} \bullet \overline{S_1} \bullet \overline{S_2}$

$O_1 = G \bullet \overline{S_0} \bullet \overline{S_1} \bullet S_2$

$O_2 = G \bullet \overline{S_0} \bullet S_1 \bullet \overline{S_2}$

$O_3 = G \bullet \overline{S_0} \bullet S_1 \bullet S_2$

$O_4 = G \bullet S_0 \bullet \overline{S_1} \bullet \overline{S_2}$

$O_5 = G \bullet S_0 \bullet \overline{S_1} \bullet S_2$

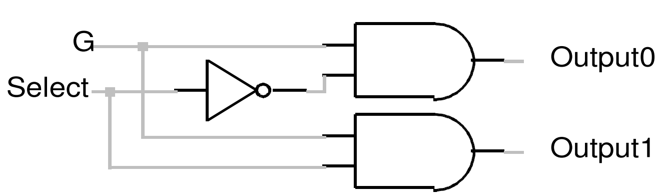
$O_6 = G \bullet S_0 \bullet S_1 \bullet \overline{S_2}$

$O_7 = G \bullet S_0 \bullet S_1 \bullet S_2$

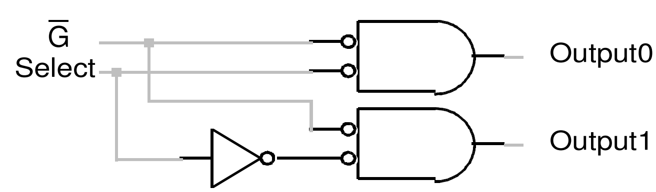
ETC - Disseny de Sistemes Combinacionals

Descodificador (II)

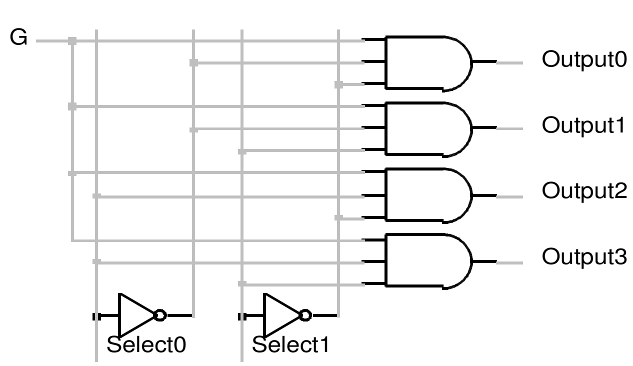
Implementació amb portes de **descodificadors** de 2 i 4 Sortides, **n** = 1 i 2 Bits de Control



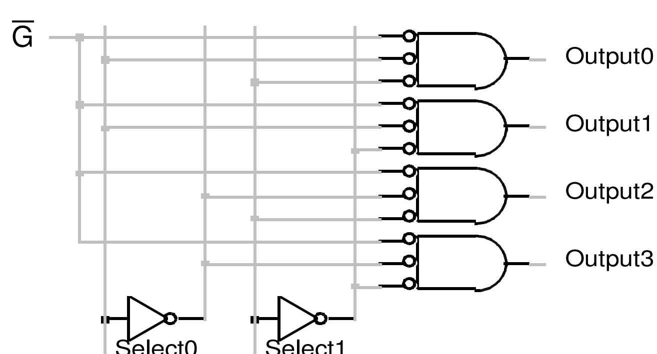
1 : 2 Decoder activat per **Enable High**



1 : 2 Decoder activat per **Enable Low**



2 : 4 Decoder activat per **Enable High**



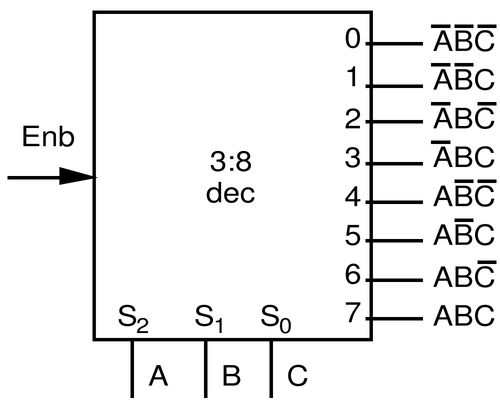
2 : 4 Decoder activat per **Enable Low**

ETC - Disseny de Sistemes Combinacionals

Descodificador (III)

Implementar **funcions lògiques** utilitzant **decodificadors**

El d**escodificador** pot generar **Mínterms** d'acord amb els Bits de Control



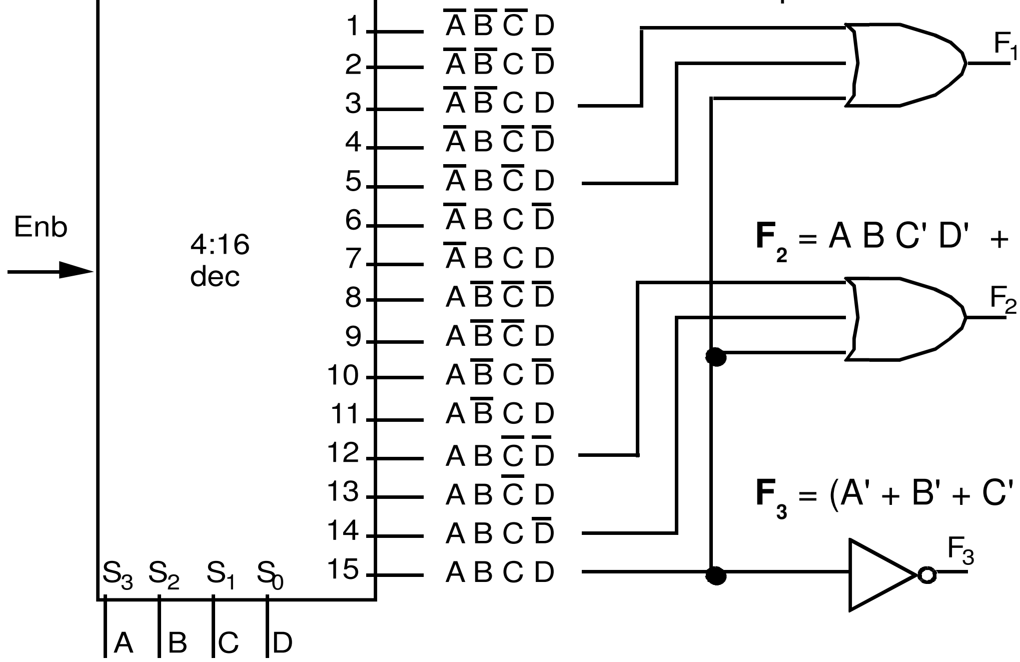
ETC - Disseny de Sistemes Combinacionals

Descodificador (IV)

$\mathbf{F_1}$ = A' B C' D + A' B' C D + A B C D

$\mathbf{F_2}$ = A B C' D' + A B C

$\mathbf{F_3}$ = (A' + B' + C' + D')



ETC - Disseny de Sistemes Combinacionals

Memòries ROM

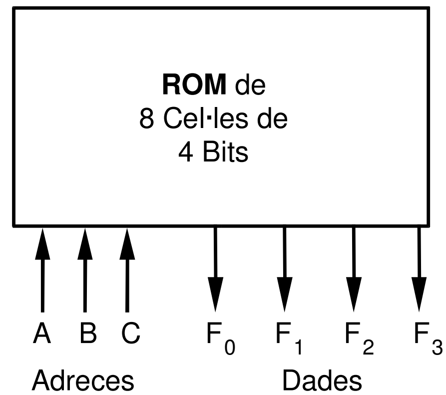
Implementar **funcions lògiques** utilitzant **memòries ROM**

$\mathbf{F_0} = A'\,B'\,C \;+\; A\,B'\,C' \;+\; A\,B'\,C$

$\mathbf{F_1} = A'\,B'\,C \;+\; A'\,B\,C' \;+\; A\,B\,C$

$\mathbf{F_2} = A'\,B'\,C' \;+\; A'\,B'\,C \;+\; A\,B'\,C'$

$\mathbf{F_3} = A'\,B\,C \;+\; A\,B'\,C' \;+ A\,B\,C'$



Adreces

A	B	C	F_0	F_1	F_2	F_3
0	0	0	0	0	1	0
0	0	1	1	1	1	0
0	1	0	0	1	0	0
0	1	1	0	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	1	0	1	0	0

Contingut de la Cel·la
(Valor de les funcions F_i)