

CIRCUITS COMBINACIONALS

Índex de conceptes

- **Entrades de control (de selecció) i Enable**
- **Convertidors de codi**
- **Comparadors**
- **Memoria ROM**

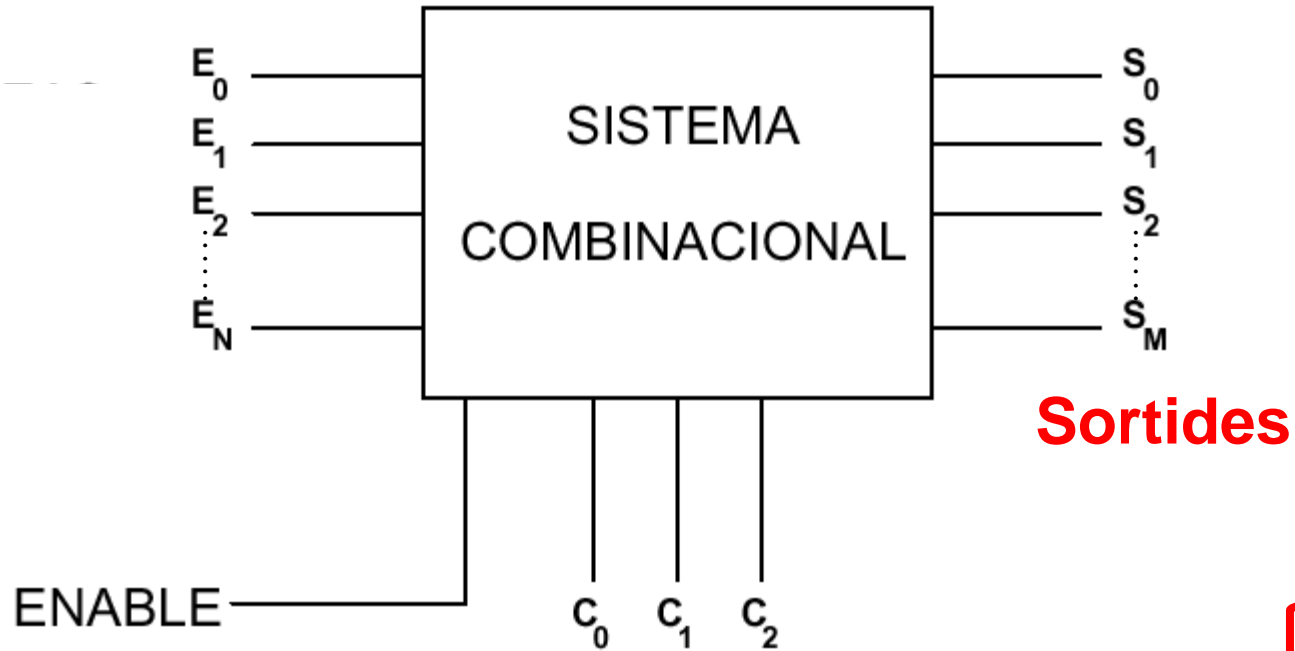
El sistema digital en el qual **la sortida només depèn de l'estat de les variables d'entrada i no depèn d'estats** anteriors s'anomena sistema **combinacional** (combinació de les variables d'entrada).

Qualsevol sistema digital es pot implementar mitjançant portes lògiques a dos nivells, però el circuit resultant pot ser molt complex, amb moltes entrades i difícil de realitzar a la pràctica. Els sistemes complexos es dissenyen de forma jeràrquica o modular: estan formats per subsistemes, que a la seva vegada, poden estar constituïts per mòduls amb funcions ben determinades i, els quals, es poden realitzar amb dos nivells de portes.

Els mòduls poden ser circuits combinacionals ad hoc (dissenyats específicament) i circuits combinacionals estàndard. Entre aquest últim destaquen els que realitzen tasques de **codificació, adreçament, commutació i operacions aritmètiques** entre senyals.

Aquests blocs, a més de terminals de dades (entrades i sortides) i terminals de polarització (alimentació elèctrica), acostumen a tenir **entrades de control**, que modifiquen la funcionalitat del mòdul, inhibint-lo o activant-lo (per exemple les senyals enable or disable)

**Entrades
de dades**



Sortides

Entrades de control

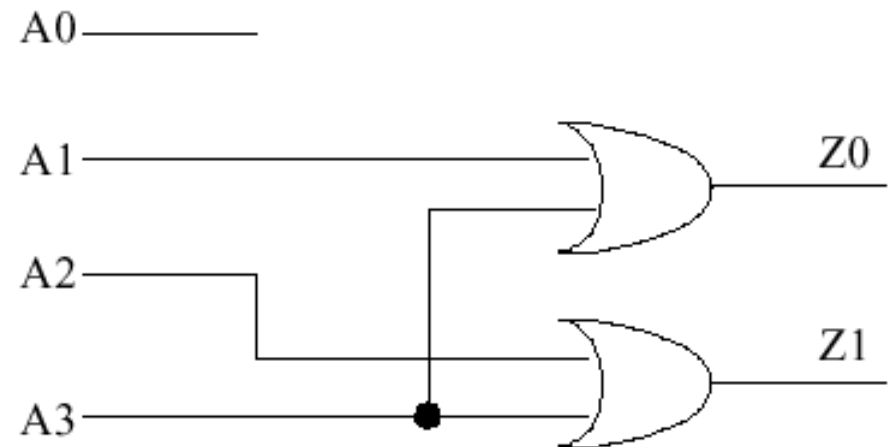
Exemple de sistemes combinacionals

Descodificadors
Multiplexors
Codificadors
Comparadors
ROM
....

Codificadors

Són circuits combinacionals de 2^n entrades i n sortides de manera que quan una de les entrades adopta un estat determinat (0 o 1) diferent a les altres entrades, a la sortida apareix la **combinació binària corresponent al número decimal assignat a la posició de l'entrada que és diferent**.

Entrades				Sortides	
A3	A2	A1	A0	z1	z0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



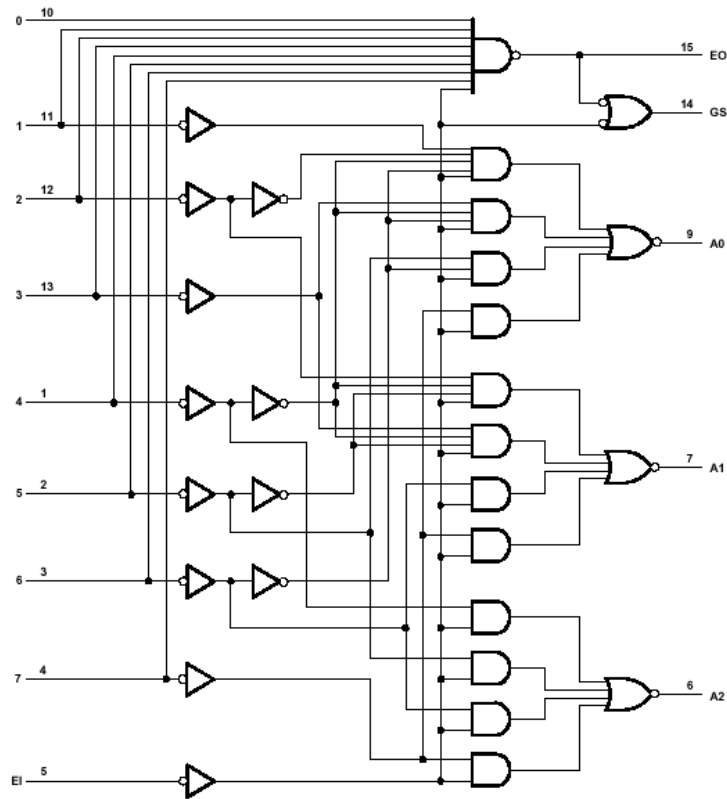
Per resoldre el problema de que arribin, simultàniament, dues entrades diferents, es dissenyen **codificadors amb prioritat**: en cas de que arribin dues entrades el codificador donarà, a la sortida, la combinació binària corresponent a l'entrada de valor decimal més alta (a l'exemple, un cas active-low).

- EI=entrada d'inhibició
- GS=sortida que permet de connectar més d'un codificador

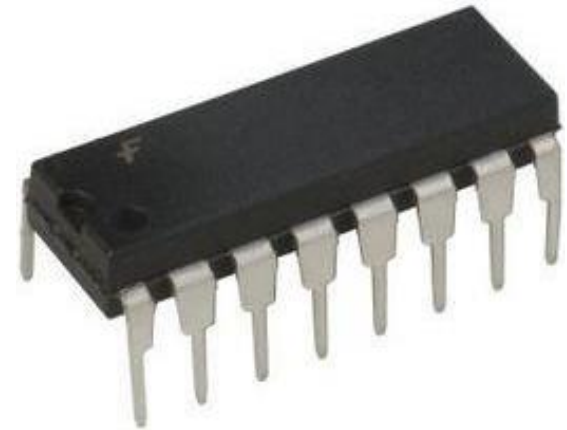
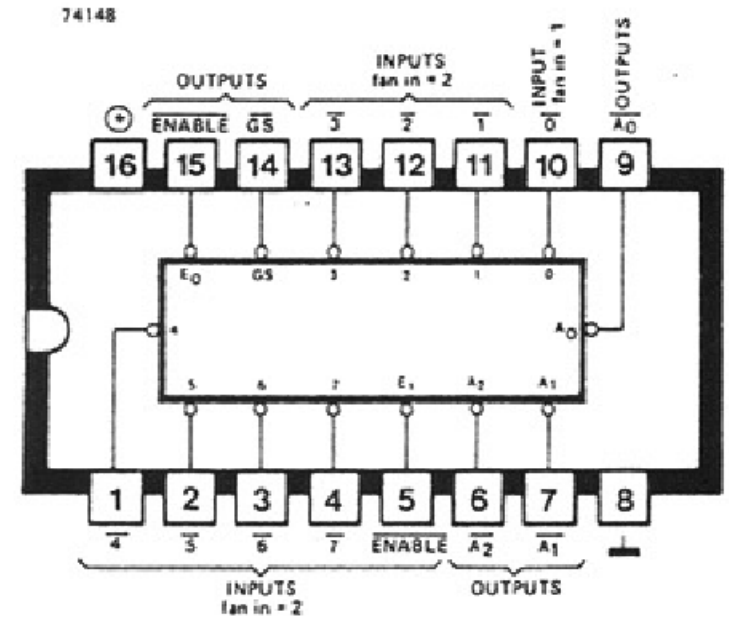
EI	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	Q ₂	Q ₁	Q ₀	GS	EO
1	X	X	X	X	X	X	X	X	0	0	0	1	1
0	1	1	1	1	1	1	1	1	0	0	0	1	0
0	X	X	X	X	X	X	X	0	1	1	1	0	1
0	X	X	X	X	X	X	0	1	1	1	0	0	1
0	X	X	X	X	X	0	1	1	1	0	1	0	1
0	X	X	X	X	0	1	1	1	1	0	1	0	1
0	X	X	0	1	1	1	1	1	0	1	0	0	1
0	X	0	1	1	1	1	1	1	0	0	1	0	1
0	0	1	1	1	1	1	1	1	0	0	0	0	1

FUNCTION TABLE

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

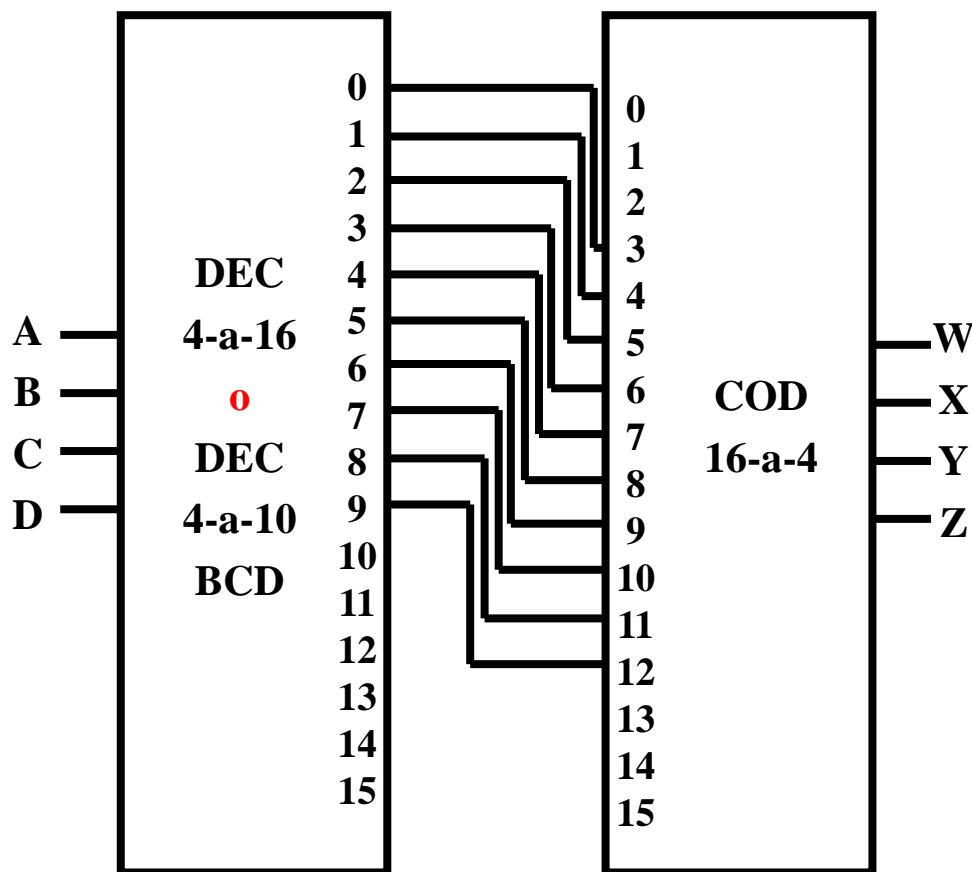


Codificador amb prioritat 74LS148



Convertidors de codi

El convertidor de codi és un circuit combinacional que **tradueix d'un codi a un altre**. Aquest circuit també es pot realitzar utilitzant un descodificador i connectar les seves sortides a un codificador de forma adient.



Convertidor BCD a BCD-excés 3

	BCD				BCD-excés 3 (codi antisimètric)			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

~ Memoria ROM

Comparadors

Són blocs combinacionals que realitzen la comparació de dos números de n bits, entesos com a binaris naturals, i indiquen **la relació entre les dues magnituds** que es presenten. Tenen **3 sortides**: més gran, igual i més petit (**G,E,L**).

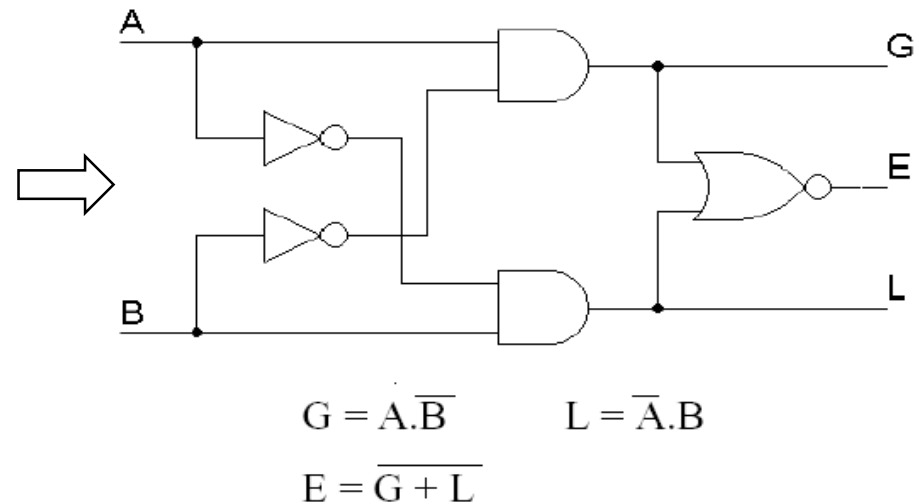
La taula de veritat i un esquema d'un comparador de 2 bits són:

$$G = A > B$$

$$E = A = B$$

$$L = A < B$$

A	B	E	G	L
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0



Comparadors

Per comparar paraules de n bits (no tots els bits tenen la mateixa importància per fer la comparació) tinc dues opcions:

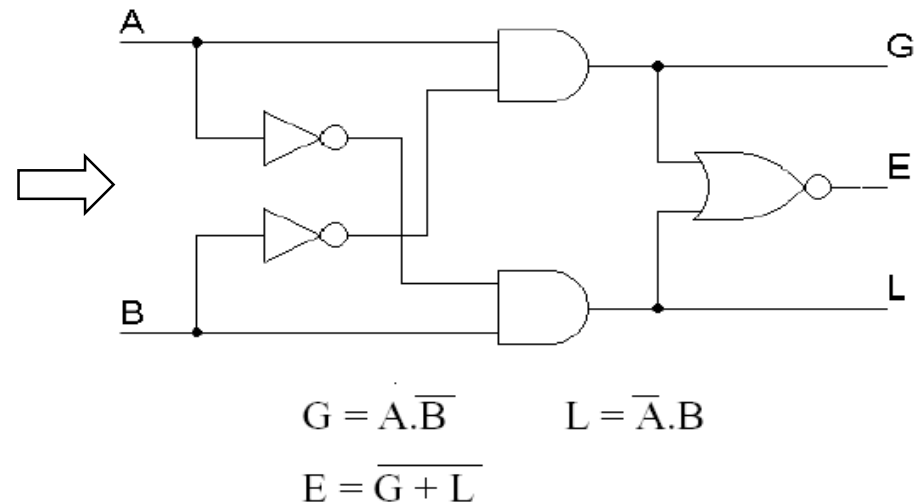
- Utilitzar comparadors de 4 o 5 bits i entrellaçar-los per unes **entrades de control**.
- Dissenyar comparadors de n bits amb un model de **n cel·les iguals (xarxa iterativa)**.

$$G = A > B$$

$$E = A = B$$

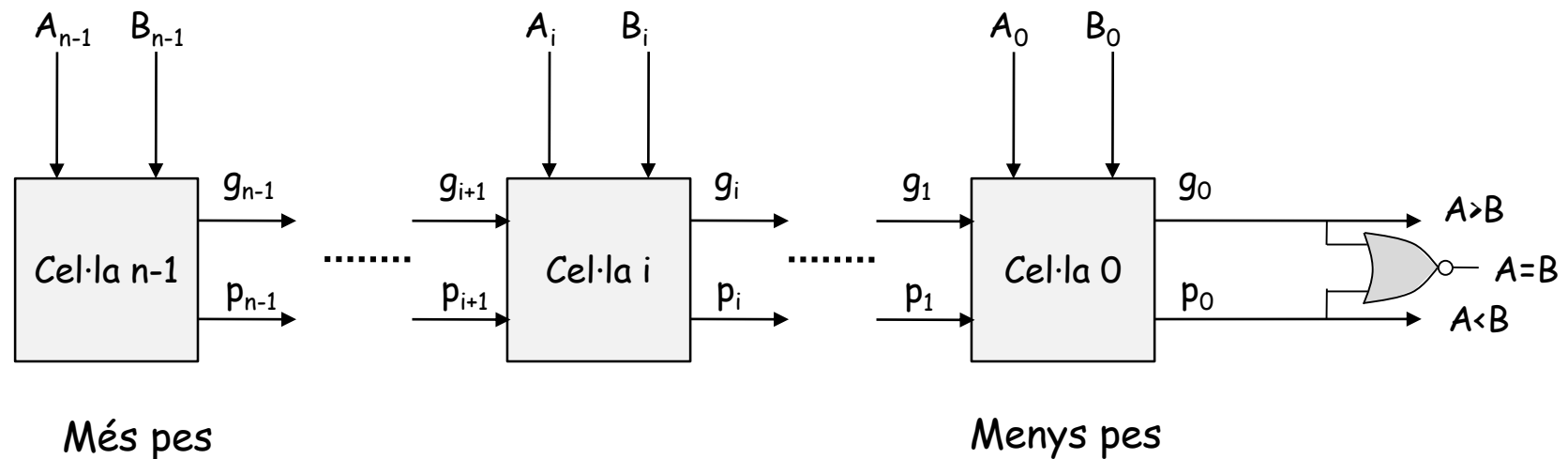
$$L = A < B$$

A	B	E	G	L
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0



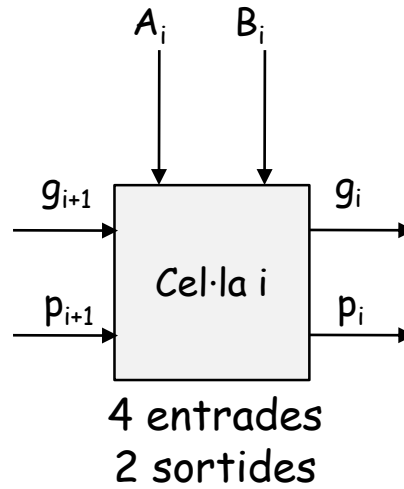
Disseny de Comparadors per Xarxa iterativa

Cada cel·la compara els bits de la seva posició i el resultat de la comparació dels anteriors. Les seves sortides **propaguen el resultat de la comparació a la cel·la posterior**. I totes les cel·les són iguals.



Dissenyem una cel·la

g_{i+1}	p_{i+1}	A_i	B_i	g_i	p_i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X



Funcions-Sortides

$g_i=0$ ↑ propaguen que fins ara
 $p_i=0$ ↓ A=B

$g_i=0$ ↑ propaguen que fins ara
 $p_i=1$ ↓ A<B

$g_i=1$ ↑ propaguen que fins ara
 $p_i=0$ ↓ A>B

Variables entrada

$g_{i+1}=0$ ↑ indiquen que els bits
 $p_{i+1}=0$ ↓ anteriors són iguals

$g_{i+1}=0$ ↑ indiquen que els bits
 $p_{i+1}=1$ ↓ anteriors són A<B

$g_{i+1}=1$ ↑ indiquen que els bits
 $p_{i+1}=0$ ↓ anteriors són A>B

$g_{i+1}=1$ No especificat
 $p_{i+1}=1$

Dissenyem una cel·la

g_{i+1}	p_{i+1}	A_i	B_i	g_i	p_i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

$g_{i+1} p_{i+1}$ $A_i B_i$	00	01	11	10
00			X	1
01			X	1
11			X	1
10	1		X	1

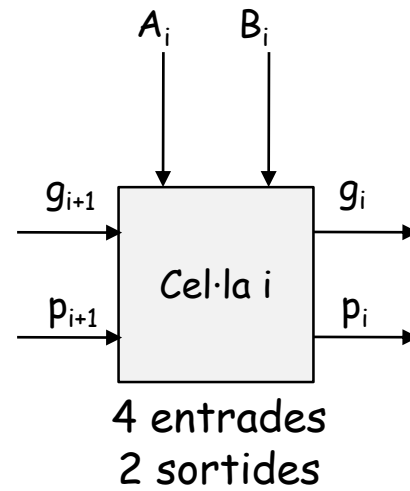
$$g_i = g_{i+1} + \bar{p}_{i+1} \cdot A_i \cdot \bar{B}_i$$

$g_{i+1} p_{i+1}$ $A_i B_i$	00	01	11	10
00		1	X	
01	1	1	X	
11		1	X	
10		1	X	

$$p_i = p_{i+1} + \bar{g}_{i+1} \cdot \bar{A}_i \cdot B_i$$

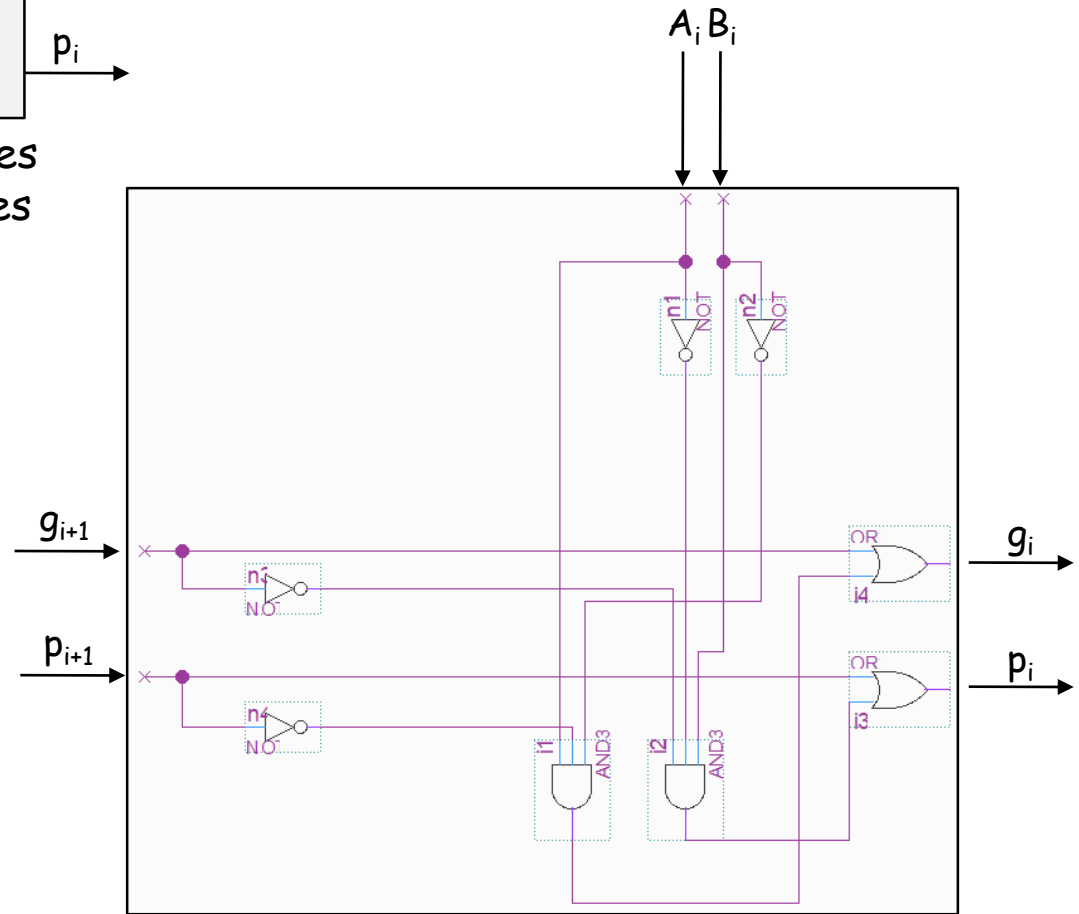
Dissenyem una cel·la

g_{i+1}	p_{i+1}	A_i	B_i	g_i	p_i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X



$$g_i = g_{i+1} + \bar{p}_{i+1} \cdot A_i \cdot \bar{B}_i$$

$$p_i = p_{i+1} + \bar{g}_{i+1} \cdot \bar{A}_i \cdot B_i$$

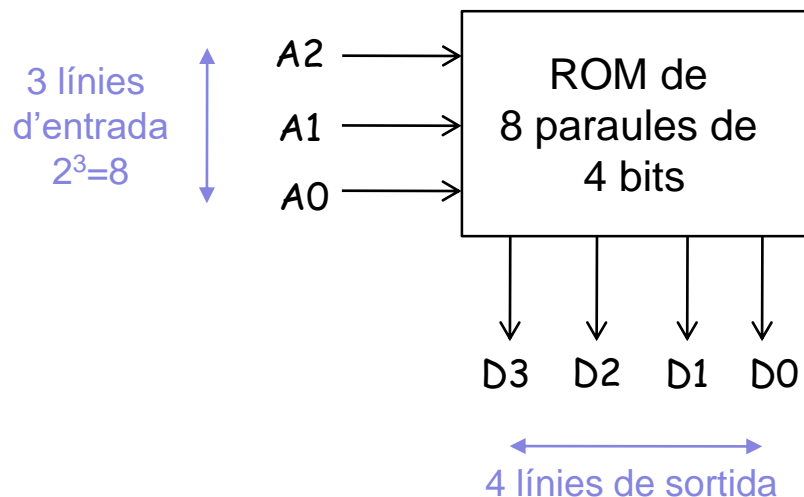


Memoria ROM (Read Only Memory)

Memòria de “Només Lectura”,

- Una gravació; múltiples lectures
- n línies d'entrada (adreça): 2^n posicions de memòria
- m línies de sortida (amplada de la paraula guardada)
- quan es posa una determinada adreça, la sortida **dóna el valor guardat en la posició seleccionada per l'adreça**

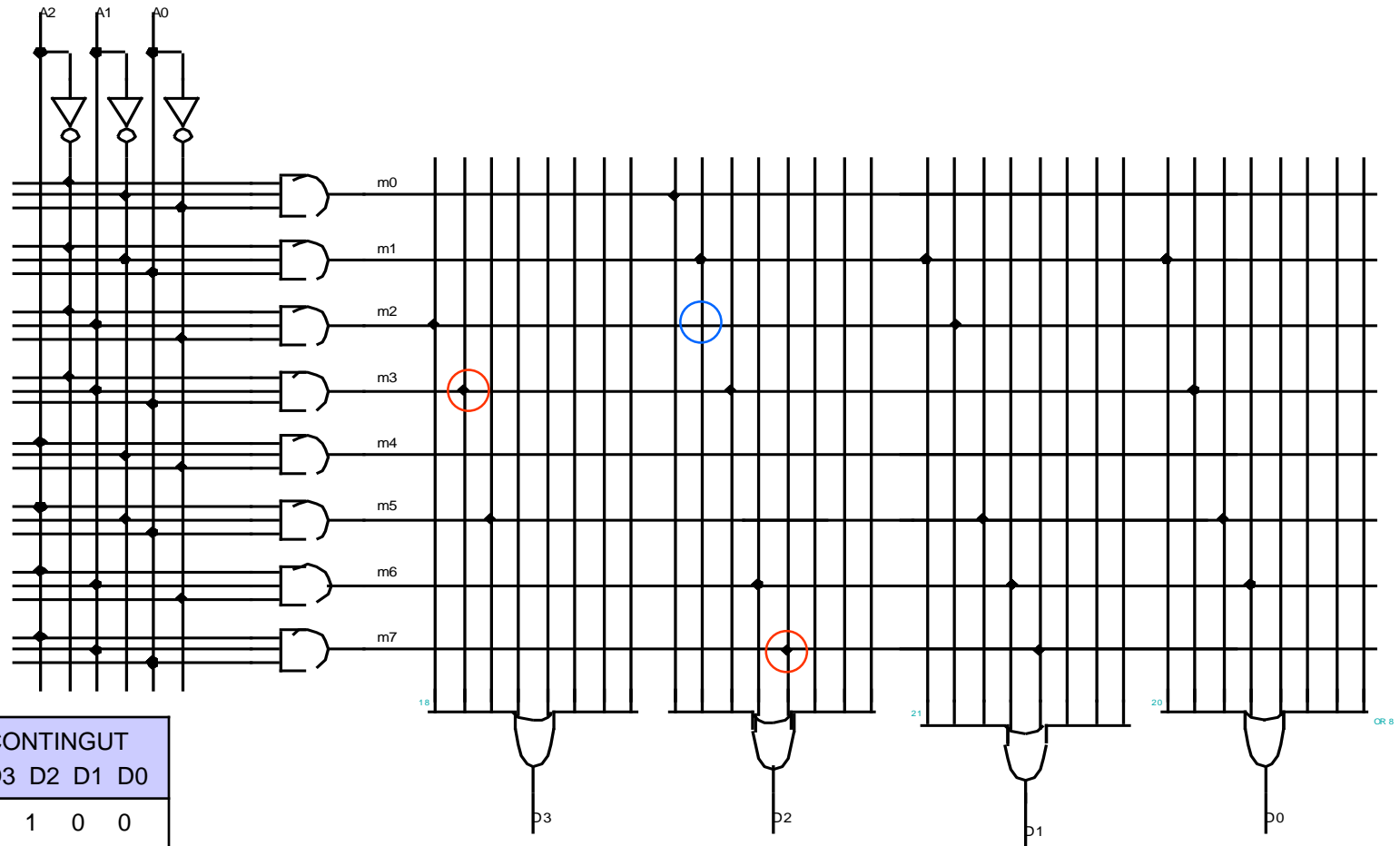
Exemple



Posició de memòria	Adreça			CONTINGUT			
	A2	A1	A0	D3	D2	D1	D0
M0	0	0	0	0	1	0	0
M1	0	0	1	0	1	1	1
M2	0	1	0	1	0	1	0
M3	0	1	1	1	1	0	1
M4	1	0	0	0	0	1	0
M5	1	0	1	1	0	1	1
M6	1	1	0	0	1	1	1
M7	1	1	1	0	1	0	0

ROM = Matriu AND-OR.

(similar a utilitzar Decodificador+Codificador)



Posició de memòria	Adreça A2 A1 A0			CONTINGUT D3 D2 D1 D0			
M0	0	0	0	0	1	0	0
M1	0	0	1	0	1	1	1
M2	0	1	0	1	0	1	0
M3	0	1	1	1	1	0	1
M4	1	0	0	0	0	1	0
M5	1	0	1	1	0	1	1
M6	1	1	0	0	1	1	1
M7	1	1	1	0	1	0	0

Els contactes a les interseccions estableixen el valor guardat (1 o 0)