

# Fonaments dels Computadors

## Tema 3: Circuits Combinacionals

Maria Teresa Signes

[teresa@dtic.ua.es](mailto:teresa@dtic.ua.es)



**Docència en valencià**

# Objectius

---

- ◆ Donar a conèixer els circuits combinacionals més habituals.
- ◆ Adquirir destresa en la resolució de problemes mitjançant l'ús de circuits combinacionals.
- ◆ Aprendre a utilitzar la informació facilitada pels fabricants en els fulls de característiques.

# Continguts

---

1. Introducció
2. Codificadors
3. Descodificadors
4. Multiplexors
5. Desmultiplexors
6. Circuits comparadors
7. Circuits aritmètics
  - Sumadors
  - Restadors
8. Convertidors de codi
9. Generadors/detectors de paritat
10. Fulls de característiques
11. Referències

# Direccions URL

---

◆ <http://www.datasheet4u.com/>

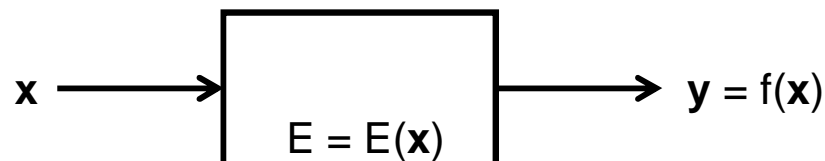
Web des d'on es poden descarregar de forma gratuïta els fulls de característiques de nombrosos components electrònics.

◆ <http://www.alldatasheets.com/>

Web dedicada a la recopilació de fulls de característiques.

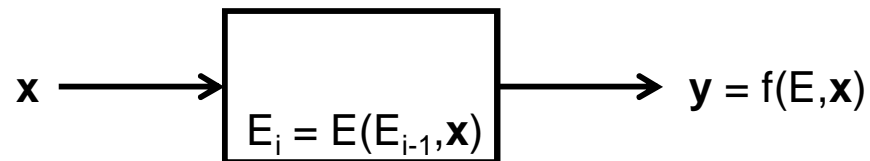
# Introducció

- ◆ Els sistemes digitals es classifiquen en dos grans grups segons el paper que juga el temps sobre els valors de les variables d'eixida.
  - Sistemes combinacionals.
  - Sistemes seqüencials.
- ◆ En un **circuit combinacional** les eixides depenen únicament del valor de les entrades en un instant determinat.



# Introducció

- ◆ En un **circuit seqüencial les eixides** no depenen exclusivament dels valors de les entrades en un moment determinat, sino també dels valors en instants anteriors.
- ◆ En aquest tema ens centrem en l'estudi dels sistemes combinacionals.



# Codificadors

◆ **Definició:** Un **codificador**  $m:n$  es un circuit combinacional que té  $m$  entrades i  $n$  eixides (amb la condició  $m \leq 2^n$ ). La seua funció consisteix a presentar a l'eixida el codi binari natural que correspon a l'entrada activada.

◆ **Classificació:**

- Segons la relació entre entrades i eixides:

Un codificador es **complet** si  $m = 2^n$ . En qualsevol altre cas s'anomena **incomplet**.

- Segons si tenen o no la capacitat de respondre davant activacions simultànies de les entrades:
  - Codificadors sense prioritat.
  - Codificadors amb prioritat.

# Codificadors

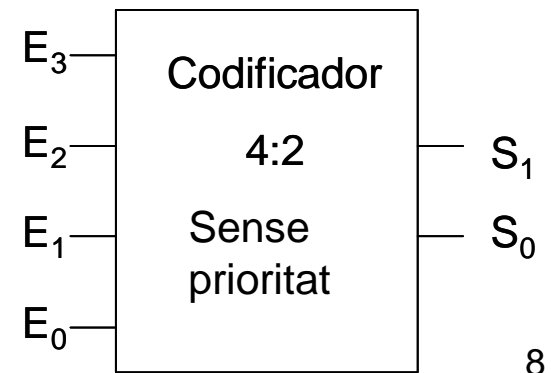
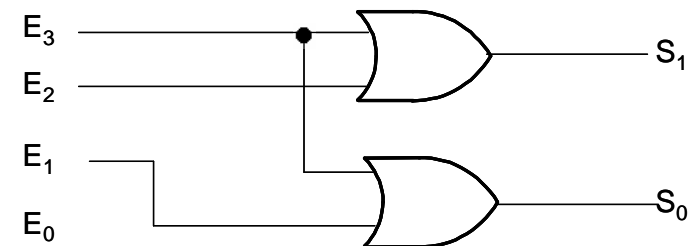
## ◆ Codificadors sense prioritat:

- En cas que s'activen simultàniament dos o més entrades, la codificació que s'obté a l'eixida queda indeterminada.
- **Example:** Codificador 4:2 sense prioritat.

E3	E2	E1	E0	S1	S0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$S_1 = E_3 + E_2$$

$$S_0 = E_3 + E_1$$





# Codificadors

---

## ◆ Entrades i eixides addicionals:

- **EI (Enable Input)** - Entrada d'habilitació: si no està activa, l'eixida es mantindrà en un estat predeterminat i invariant davant possibles canvis a l'entrada.
- **EO (Enable Output)** – Eixida d'habilitació: s'activa quan el codificador està habilitat i no se selecciona ninguna entrada.
- **GS (Group Selection)** – Eixida de selecció de grup: s'activa quan el codificador està habilitat i se selecciona alguna de les seues entrades.

# Codificadors

## ◆ Codificadors sense prioritat:

- Exemple: Codificador 4:2 sense prioritat, amb entrades i eixides addicionals per a l'habilitació i selecció de grup:

EI	E3	E2	E1	E0	S1	S0	EO	GS
0	X	X	X	X	0	0	0	0
1	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	0	1
1	0	1	0	0	1	0	0	1
1	1	0	0	0	1	1	0	1

# Codificadors

## ◆ Codificadors sense prioritat:

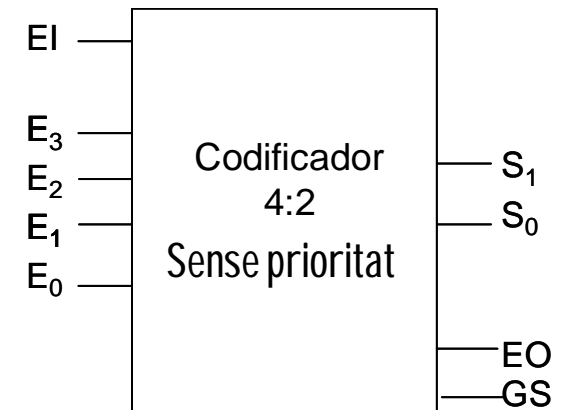
- **Exemple:** Codificador 4:2 sense prioritat amb entrades i eixides addicionals per a l'habilitació i selecció de grup:

$$S_1 = EI \cdot (E_3 + E_2)$$

$$S_0 = EI \cdot (E_3 + E_1)$$

$$EO = EI \cdot \overline{E_3} \cdot \overline{E_2} \cdot \overline{E_1} \cdot \overline{E_0}$$

$$GS = EI \cdot (E_3 + E_2 + E_1 + E_0)$$



# Codificadors

## ◆ Codificadors sense prioritat:

- Exemple: Codificador 10:4 sense prioritat, amb **entrada** i **eixides** **addicionals** per a l'habilitació i selecció de grup:

EI	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0	S3	S2	S1	S0	EO	GS
0	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1
1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1

# Codificadors

## ◆ Codificadors sense prioritat:

- **Exemple:** Codificador 10:4 sense prioritat, amb entrades i eixides addicionals per a l'habilitació i selecció de grup:

$$S_3 = EI \cdot (E_9 + E_8)$$

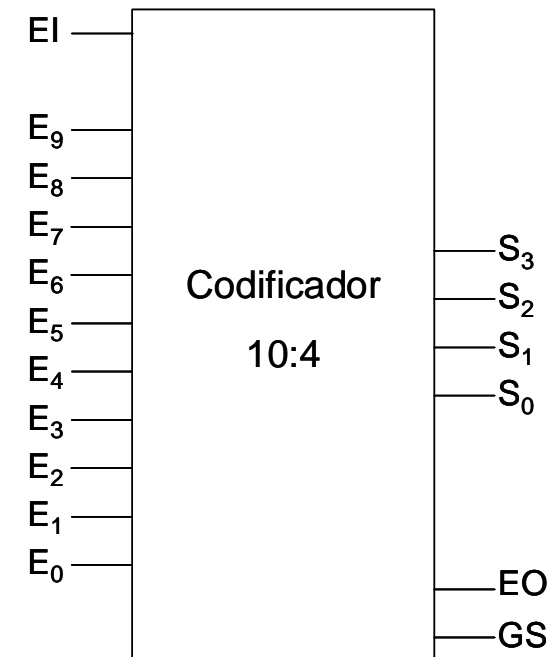
$$S_2 = EI \cdot (E_7 + E_6 + E_5 + E_4)$$

$$S_1 = EI \cdot (E_3 + E_2)$$

$$S_0 = EI \cdot (E_1 + E_0)$$

$$EO = EI \cdot \overline{E_9} \cdot \overline{E_8} \cdot \overline{E_7} \cdot \overline{E_6} \cdot \overline{E_5} \cdot \overline{E_4} \cdot \overline{E_3} \cdot \overline{E_2} \cdot \overline{E_1} \cdot \overline{E_0}$$

$$GS = EI \cdot (E_9 + E_8 + E_7 + E_6 + E_5 + E_4 + E_3 + E_2 + E_1 + E_0)$$



# Codificadors

## ◆ Codificadors amb prioritat:

- En cas d'activació simultània de dos o més entrades la codificació obtinguda a l'eixida correspon a l'entrada de major prioritat, que generalment és l' entrada de més pes.
- **Exemple:** codificador 4:2 amb prioritat, entrada d'habilitació, eixida d'habilitació i eixida de selecció de grup.

EI	E3	E2	E1	E0	S1	S0	EO	GS
0	X	X	X	X	0	0	0	0
1	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	1
1	0	0	1	X	0	1	0	1
1	0	1	X	X	1	0	0	1
1	1	X	X	X	1	1	0	1

# Codificadors

## ◆ Codificadors amb prioritat:

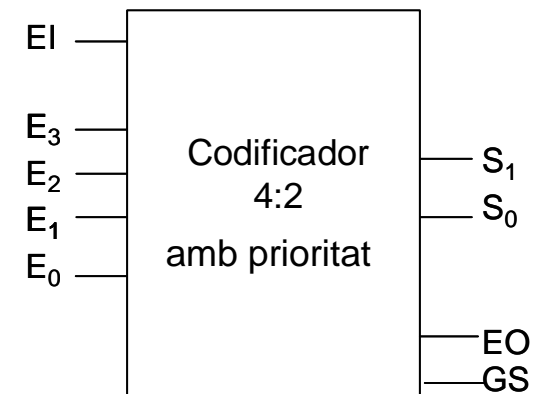
- En cas d' activació simultània de dos o més entrades la codificació obtinguda a l'eixida correspon a l'entrada de major prioritat que generalment és l'entrada de més pes.
- **Exemple:** codificador 4:2 amb prioritat, entrada d'habilitació, eixida d' habilitació i eixida de selecció de grup.

$$S_1 = EI \cdot (E_3 + E_2)$$

$$S_0 = EI \cdot (\overline{E_3} + \overline{E_2} \cdot E_1)$$

$$EO = EI \cdot E_3 \cdot E_2 \cdot E_1 \cdot E_0$$

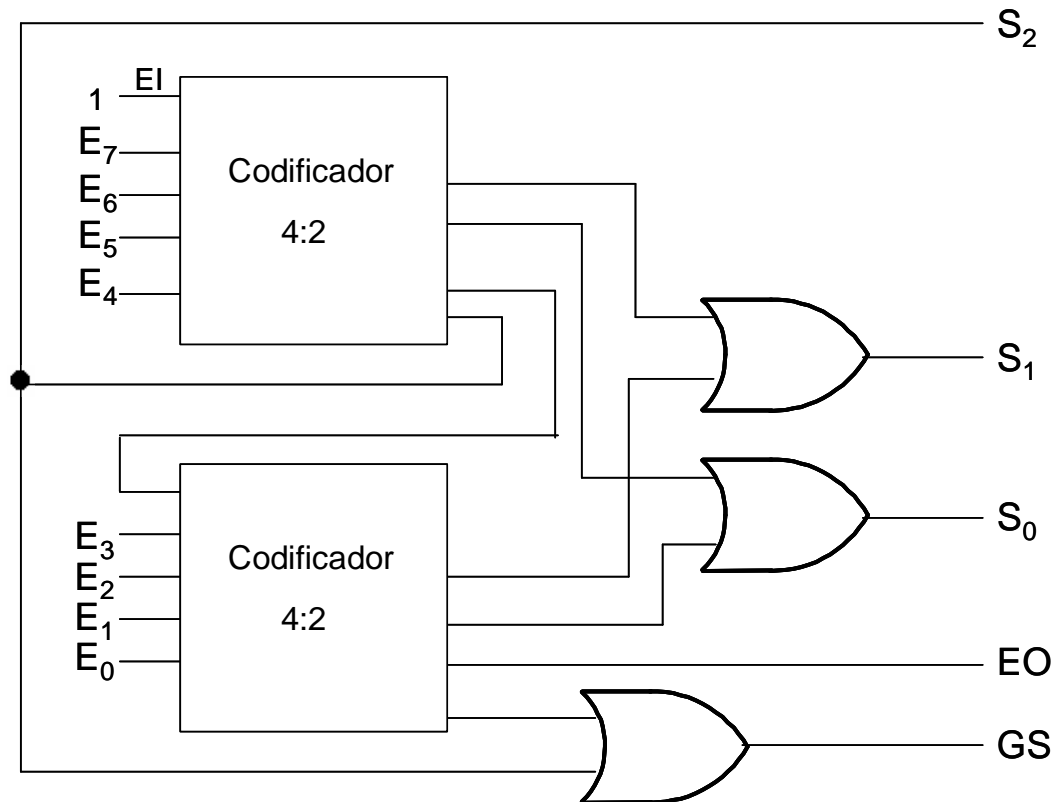
$$GS = EI \cdot (E_3 + E_2 + E_1 + E_0)$$



# Codificadors

## ◆ Extensió de la capacitat d'un codificador:

- Exemple: disseny d'un codificador 8:3 mitjançant codificadors 4:2.





# Descodificadors

---

- ◆ **Definició:** Un **descodificador**  $m:n$  és un circuit combinacional amb  $m$  entrades i  $n$  eixides ( $n \leq 2^m$ ), i la seua funció bàsica és detectar la presència d'un determinat codi binari en les seues entrades i assenyalar la presència d'aquest codi mitjançant un cert nivell d'eixida.

# Descodificadors

- **Exemple:** descodificador 2:4 amb entrada d'habilitació.

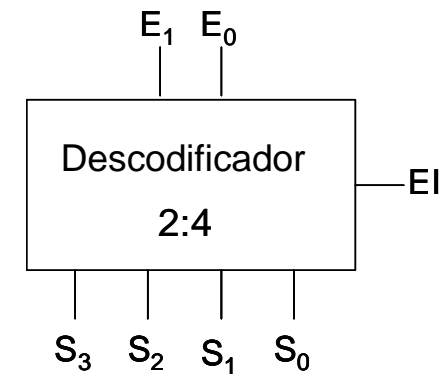
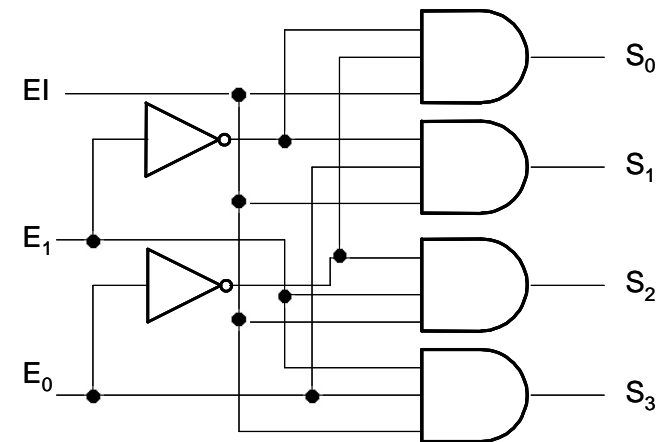
EI	E1	E0	S0	S1	S2	S3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$S_0 = \overline{E_1} \overline{E_0} EI$$

$$S_1 = E_1 \overline{E_0} EI$$

$$S_2 = \overline{E_1} E_0 EI$$

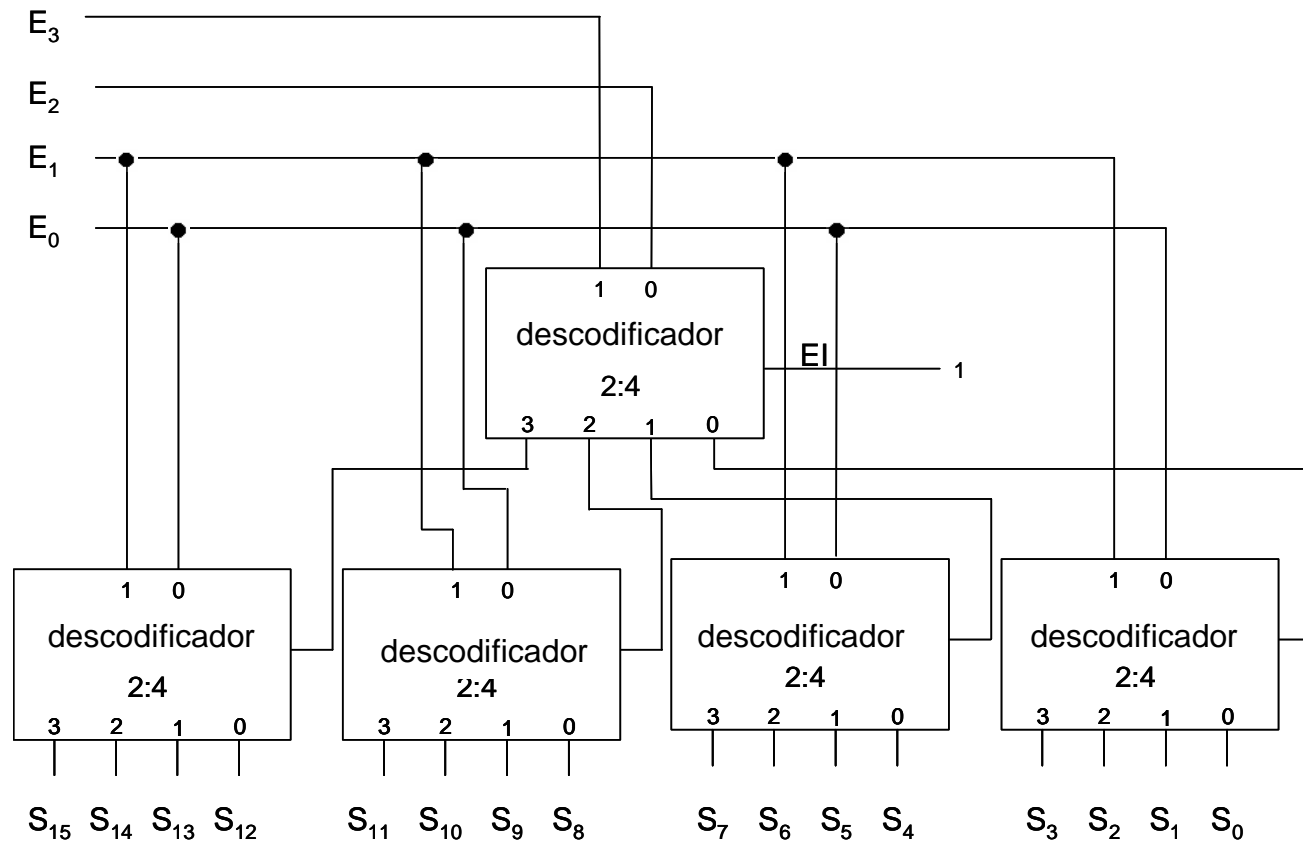
$$S_3 = E_1 E_0 EI$$



# Descodificadors

## ◆ Extensió de la capacitat d'un descodificador:

- **Exemple:** disseny d'un descodificador 4:16 amb descodificadors 2:4.



# Descodificadors

---

## ◆ Aplicacions del descodificador:

### ◆ Aplicacions usals dels descodificadors:

- Visualització de resultats.
- Adreçament d'unitats de memòria, registres, busos i perifèrics.
- Descodificació de les instruccions a executar per part de la Unitat de Control.
- Implementació de funcions booleanes.
  - Qualsevol funció booleana de  $n$  variables pot implementar-se fent ús d'un descodificador  $n:2^n$  i una porta lògica (OR o NAND).

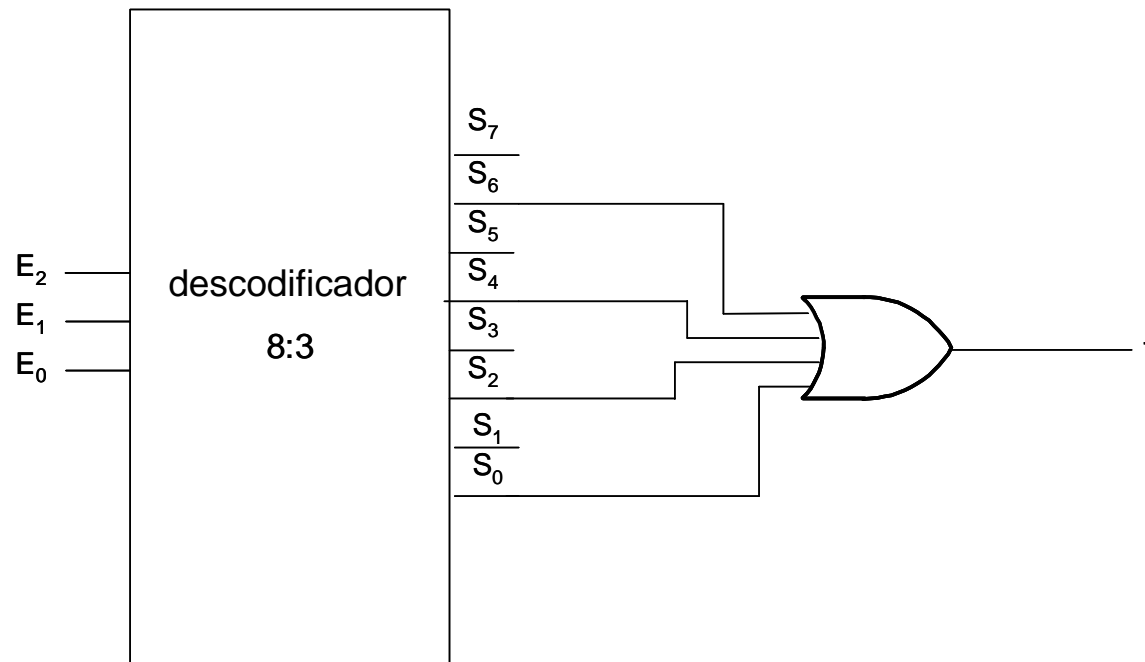
# Descodificadors

## ◆ Aplicacions del descodificador:

### ◆ Aplicacions usals dels descodificadors:

- Implementació de funcions booleanes.

❖ **Example:** Implementació de la funció booleana  $f = \sum_3 m(0,2,4,6)$  fent ús d'un descodificador.



# Multiplexors

- ◆ **Definició:** Un **multiplexor** o **selector de dades**  $2^n:1$  és un dispositiu amb  $2^n$  entrades de dades,  $n$  senyals de control i 1 eixida. Els senyals de control seleccionen una única entrada, que passa a l'eixida.

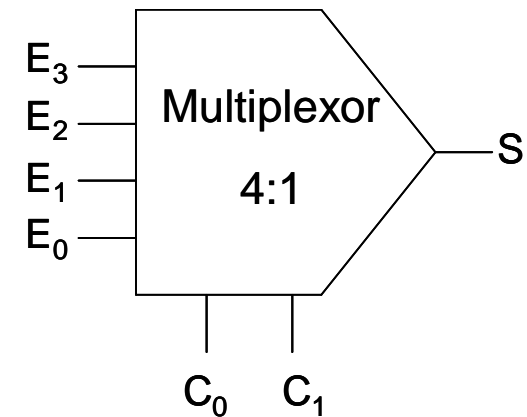
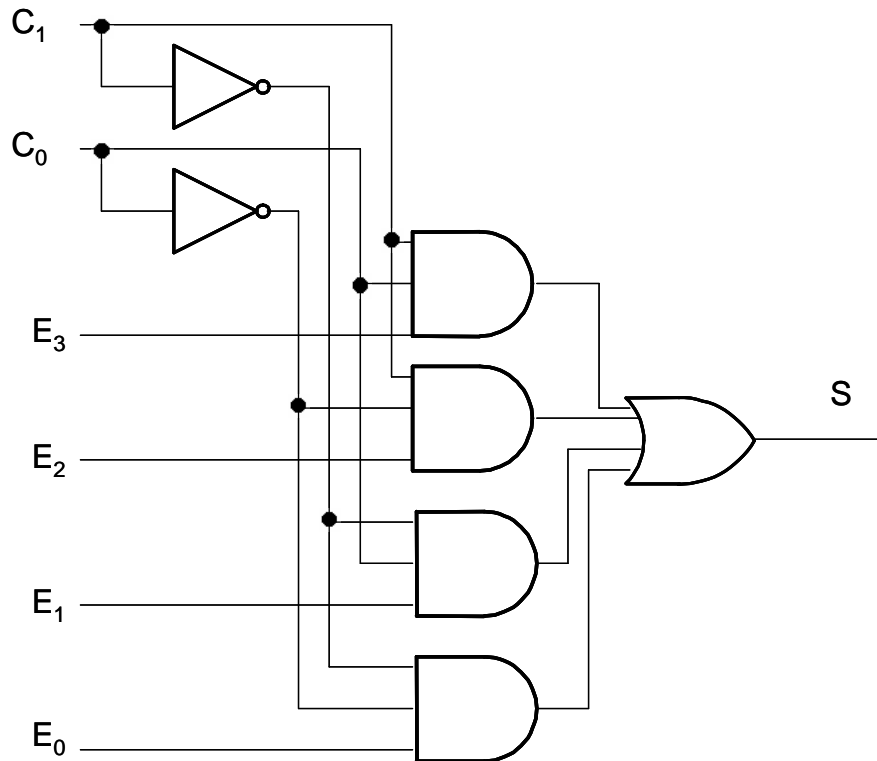
- ◆ **Exemple:** Multiplexor 4:1.

E3	E2	E1	E0	C1	C0	S
X	X	X	0	0	0	0
X	X	X	1	0	0	1
X	X	0	X	0	1	0
X	X	1	X	0	1	1
X	0	X	X	1	0	0
X	1	X	X	1	0	1
0	X	X	X	1	1	0
1	X	X	X	1	1	1

$$S = E_0 \bar{C}_1 \bar{C}_0 + E_1 \bar{C}_1 C_0 + E_2 C_1 \bar{C}_0 + E_3 C_1 C_0$$

# Multiplexors

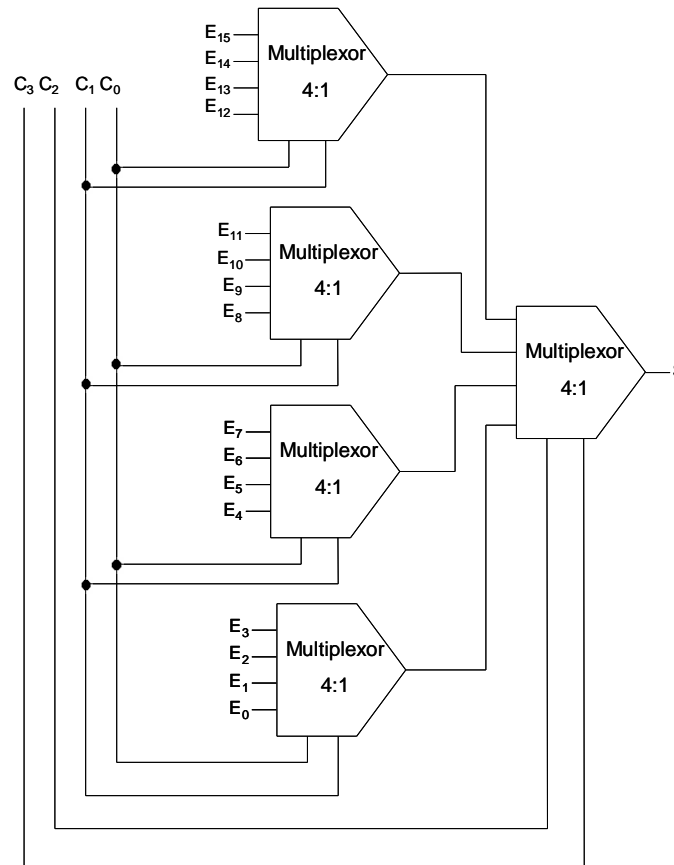
- Example: Multiplexor 4:1.



# Multiplexors

## ◆ Extensió de la capacitat d'un multiplexor:

- Exemple: Construcció d'un multiplexor 16:1 amb multiplexors 4:1.





# Multiplexors

---

## ◆ Aplicacions del multiplexor:

### ◆ Aplicacions usals dels multiplexors:

- Adreçament de registres.
- Adreçament de busos i perifèrics.
- Implementació de funcions booleans.

# Multiplexors

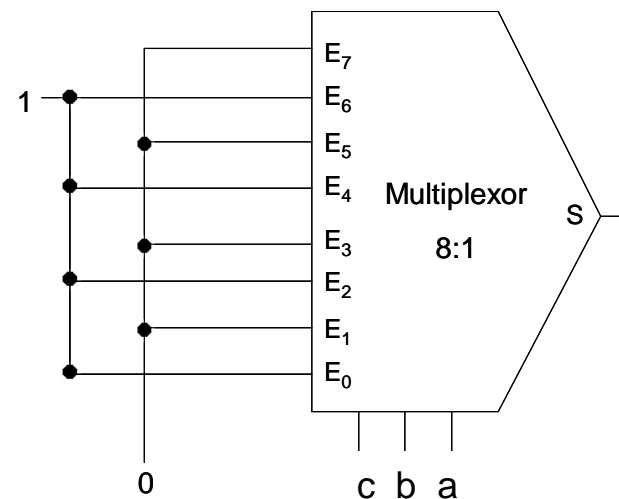
## ◆ Aplicacions del multiplexor:

### ◆ Implementació de funcions booleanes de $n$ variables:

- **Implementació 1:** Mitjançant un multiplexor de  $n$  entrades de control.
- **Implementació 2:** Mitjançant un multiplexor de  $n-1$  entrades de control.

### ◆ Exemple1: Implementació de la funció booleana $f = \sum_3 m(0,2,4,6)$ fent ús d'un multiplexor 8:1

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



# Multiplexors

◆ **Implementació 2:** Implementació de la funció booleana de  $n$  variables mitjançant un multiplexor de  $n-1$  entrades de control:

- S'associen les entrades de control a les variables lògiques exceptuant-ne una que s'utilitza per a configurar l'entrada de dades.
- Es construeix una taula que reorganitza les variables de la funció.

n-1 variables de control				1 variable de dades		
				0	1	
$2^{n-1}$	0	0	...	0		→ selecciona entrada 0
	0	0	...	1		→ selecciona entrada 1
	1	1	...	1		→ selecciona entrada $2^{n-1}$

- Casos possibles per a cada fila en la variable de dades:

Combinació en l'entrada i	0	0	→ Sempre 0 en $E_i$
Combinació en l'entrada i	0	1	→ $\text{variable\_dades}$ en $E_i$
Combinació en l'entrada i	1	0	→ $\overline{\text{variable\_dades}}$ en $E_i$
Combinació en l'entrada i	1	1	→ Sempre 1 en $E_i$

# Multiplexors

a	b	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$C = \sum_2 (1)$$

$$D = \sum_2 (1, 2)$$

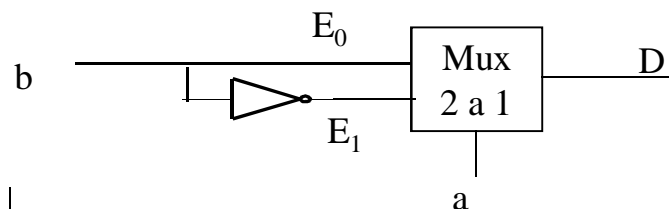
2 variables: 1 variable de selecció + 1 variable de dades : a = variable de selecció ; b = variable de dades

## Implementació de D

a	0	1	
0	0	1	selecciona entrada 0
1	1	0	selecciona entrada 1

$$E_0 = \overline{b}$$

$$E_1 = b$$

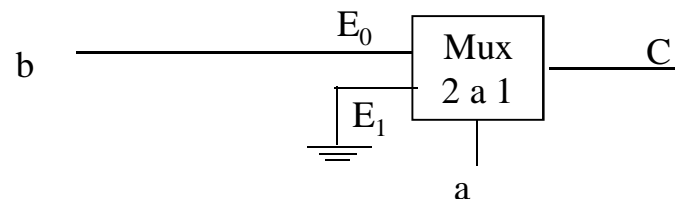


## Implementació de C

a	0	1	
0	0	1	selecciona entrada 0
1	0	0	selecciona entrada 1

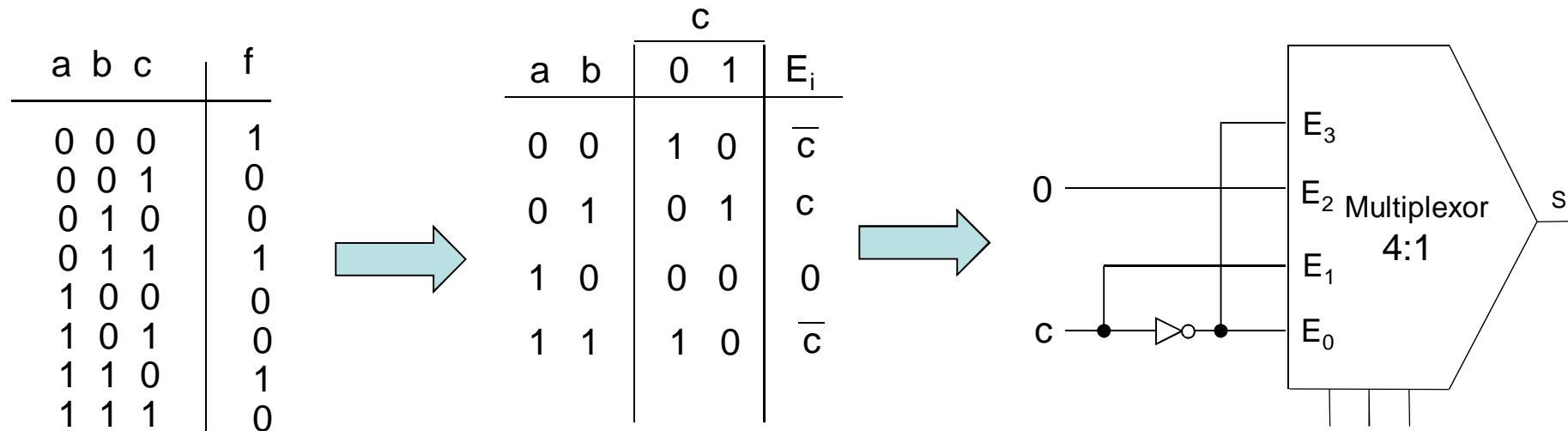
$$E_0 = b$$

$$E_1 = 0$$



# Multiplexors

- ◆ **Exemple2:** Implementació de la funció booleana  $f = \sum_3 m(0,3,6)$  fent ús d'un multiplexor 4:1



# Demultiplexors

- ◆ **Definició:** Un demultiplexor  $1:2^n$  és un dispositiu amb 1 entrada de dades,  $n$  senyals de control i  $2^n$  eixides. Mitjançant els senyals de control, seleccionem l'eixida que presentarà el contingut que té a l'entrada.

- **Exemple:** Demultiplexor 1:4.

E	C1	C0	S0	S1	S2	S3
0	0	0	0	0	0	0
1	0	0	1	0	0	0
0	0	1	0	0	0	0
1	0	1	0	1	0	0
0	1	0	0	0	0	0
1	1	0	0	0	1	0
0	1	1	0	0	0	0
1	1	1	0	0	0	1

$$S_0 = E \overline{C_1} \overline{C_0}$$

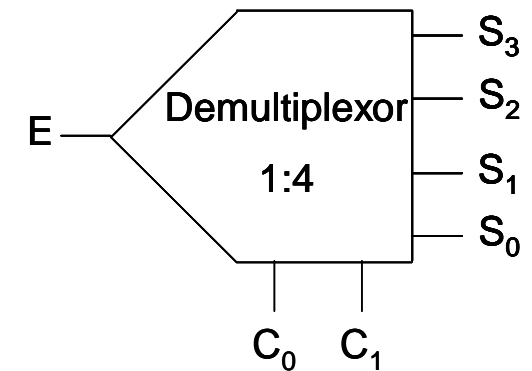
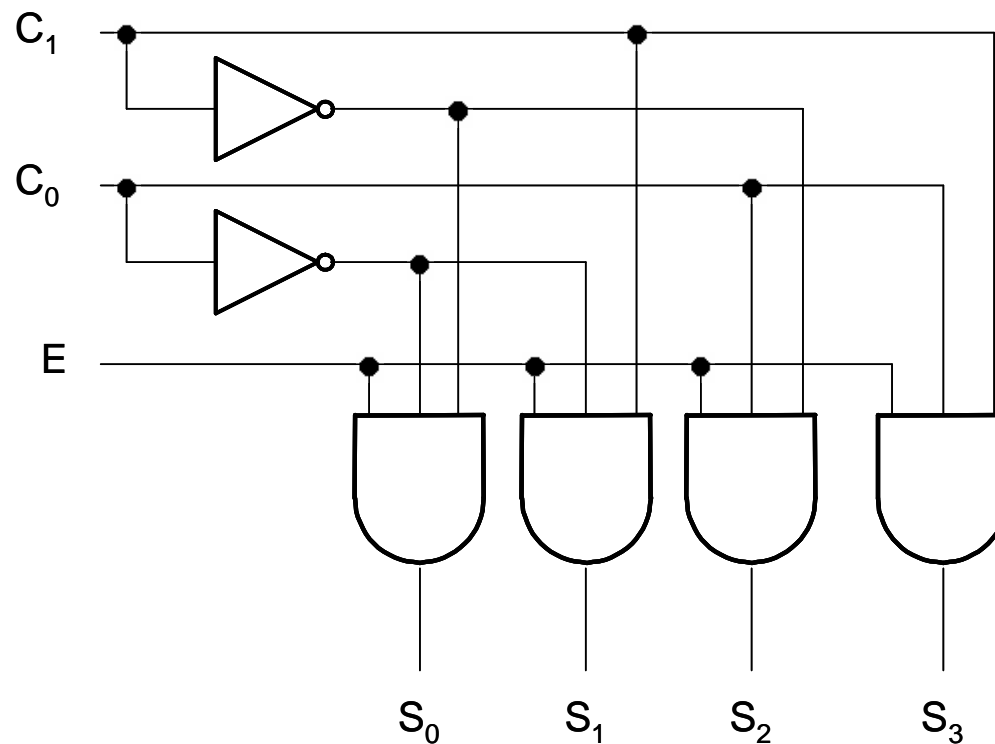
$$S_1 = E \overline{C_1} C_0$$

$$S_2 = E C_1 \overline{C_0}$$

$$S_3 = E C_1 C_0$$

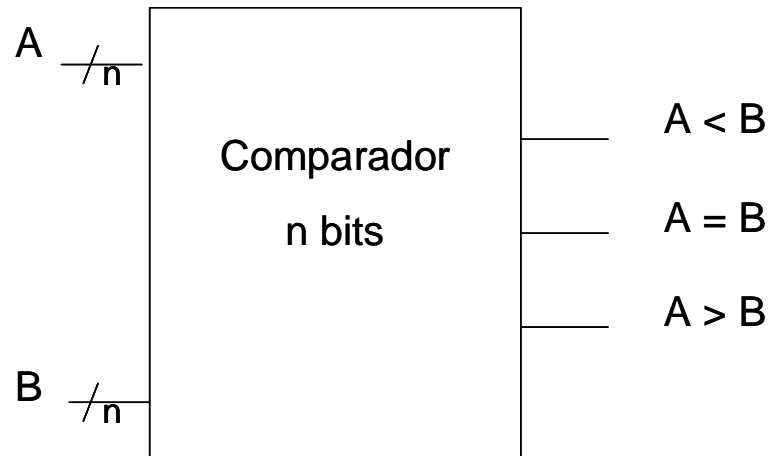
# Demultiplexors

- Example: Demultiplexor 1:4.



# Comparadors

- ◆ **Definició:** Un **comparador** és un circuit combinacional que rep com a entrada dos números A i B de n bits. En l'eixida s'indica la relació que les magnituds de A i B guarden entre sí. El nombre d'eixides pot variar segons la implementació.





# Comparadors

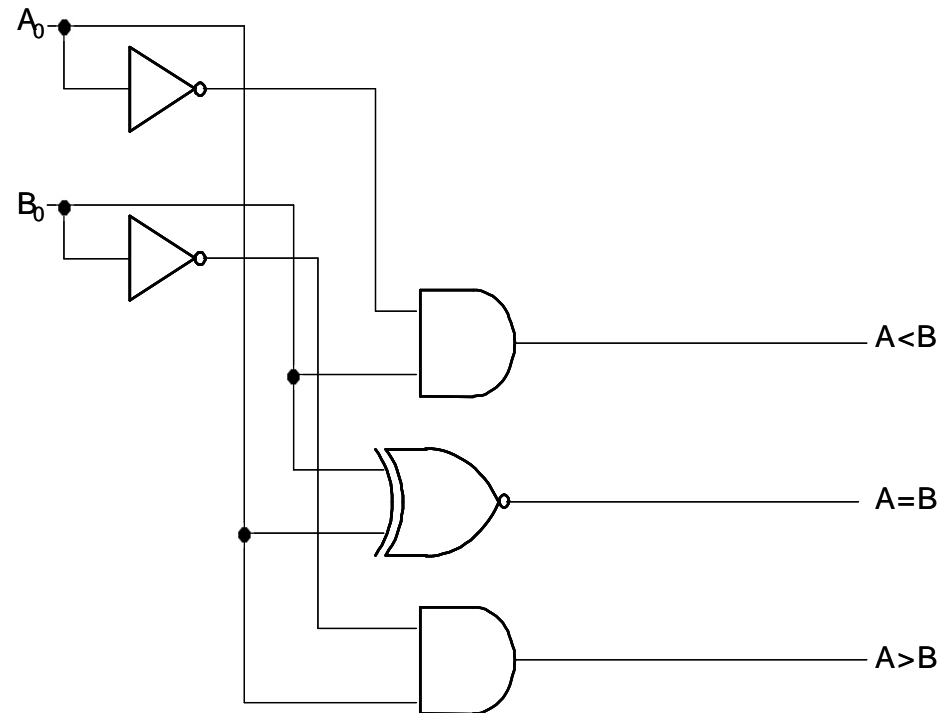
- **Exemple:** Comparador de dos nombres binaris positius d'un bit, amb tres eixides actives a nivell alt:  $A = B$ ,  $A < B$ ,  $A > B$ .

$A_0$	$B_0$	$S_{A<B}$	$S_{A=B}$	$S_{A>B}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

$$S_{A=B} = A_0 \oplus B_0$$

$$S_{A<B} = \overline{A_0} \cdot B_0$$

$$S_{A>B} = A_0 \cdot \overline{B_0}$$



# Circuits aritmètics

---

- ◆ **Definició:** Un **circuit aritmètic** és un circuit capaç de realitzar operacions amb nombres. **Exemple:** un sumador, un restador, un multiplicador, un divisor o una unitat aritmèticològica.
- ◆ La **aritmètica digital** s'encarrega de l'estudi dels circuits aritmètics.
- ◆ En aquest tema ens centrem en l'estudi d'algunes implementacions de circuits aritmètics realitzades amb sistemes combinacionals.

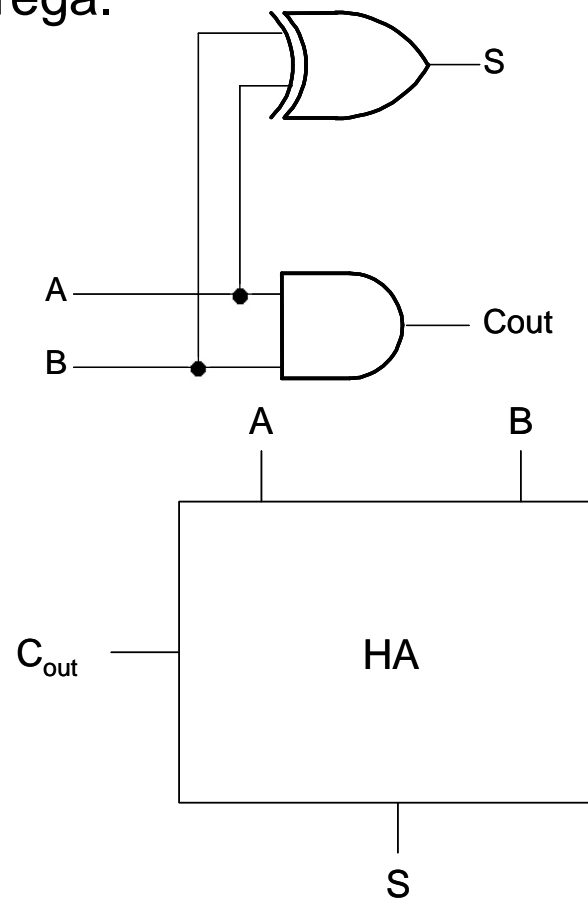
# Sumadors

- ◆ **Definició:** Un **semisumador (half-adder, HA)** d'un bit és un circuit combinacional que realitza la suma de dos bits i proporciona com a eixides un bit de suma i un altre de càrrega.

A	B	S	C <sub>out</sub>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \bar{A}B + A\bar{B} = A \oplus B$$

$$C_{out} = AB$$



# Sumadors

- ◆ **Definició:** Un **sumador complet** (full-adder, FA) d'un bit és un circuit combinacional que realitza la suma de dos bits i una càrrega d'entrada, que proporciona com a eixides el bit de suma i un bit de càrrega.

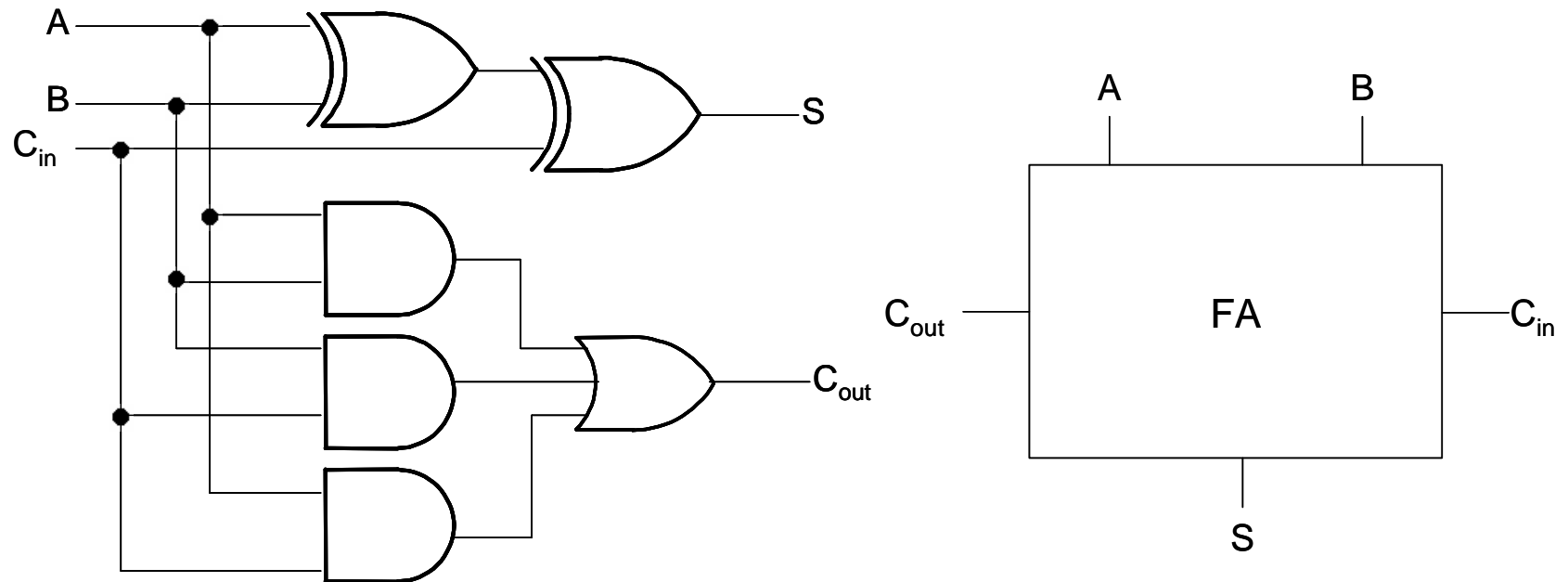
A	B	C <sub>in</sub>	S	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in} = A \oplus B \oplus C_{in}$$

$$\begin{aligned} C_{out} &= \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} = \\ &= AB + AC_{in} + BC_{in} = AB + (A \oplus B)C_{in} \end{aligned}$$

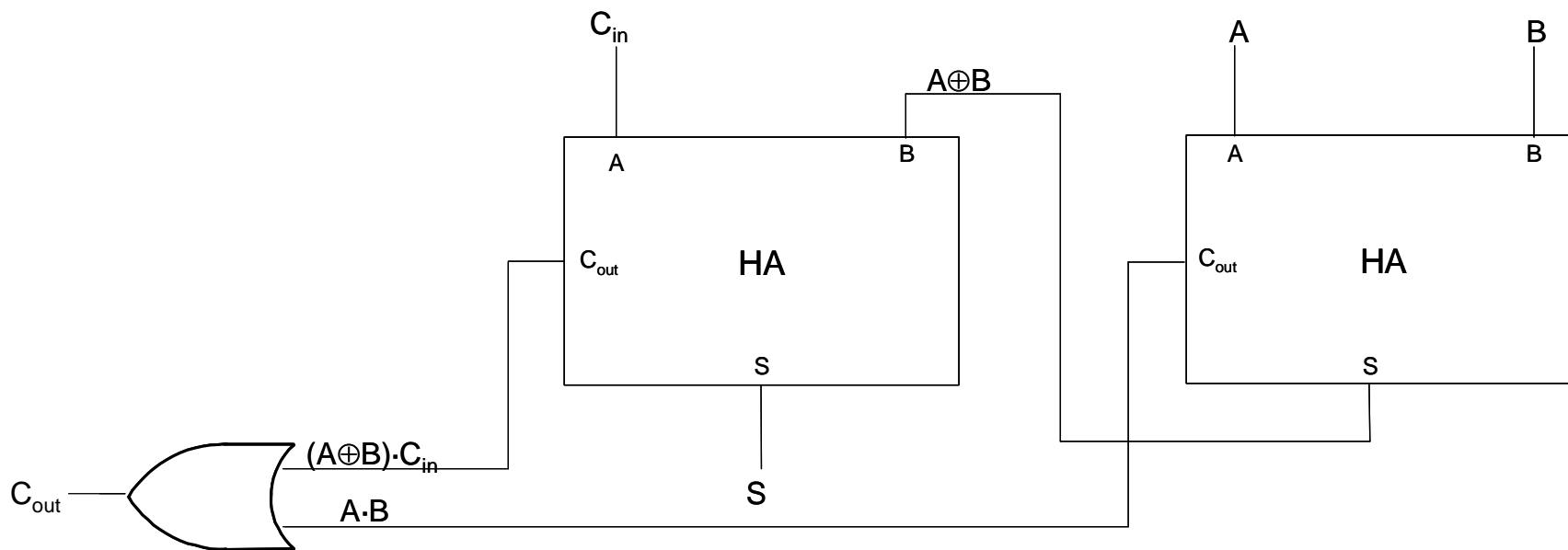
# Sumadors

- ◆ Implementació del sumador complet amb portes lògiques.



# Sumadors

- ◆ Implementació del sumador complet amb semisumadors.

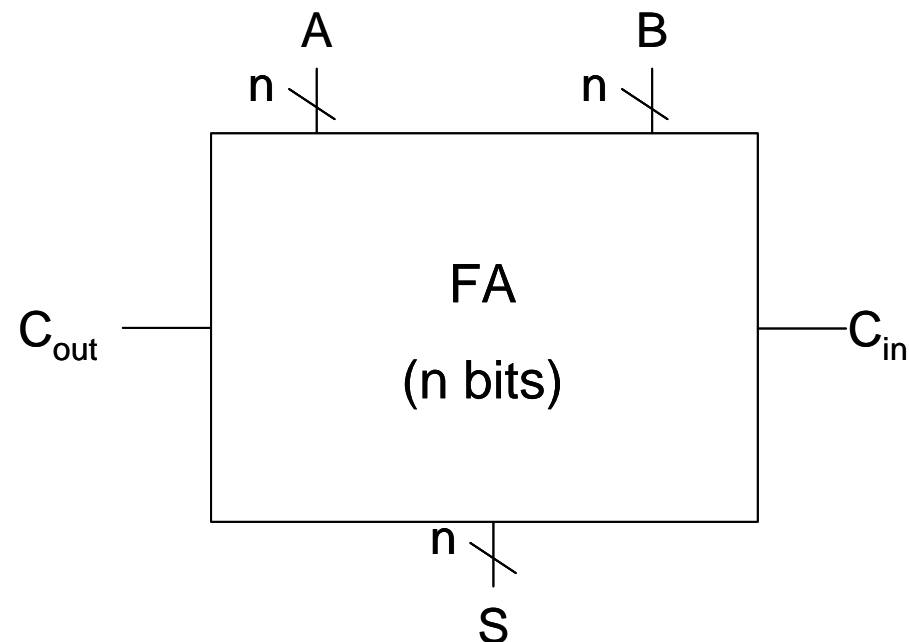


$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + (A \oplus B)C_{in}$$

# Sumadors

- ◆ **Definició:** Un **sumador de n bits** és un circuit combinacional que realitza l'operació de suma sobre operands d'una longitud de n bits.



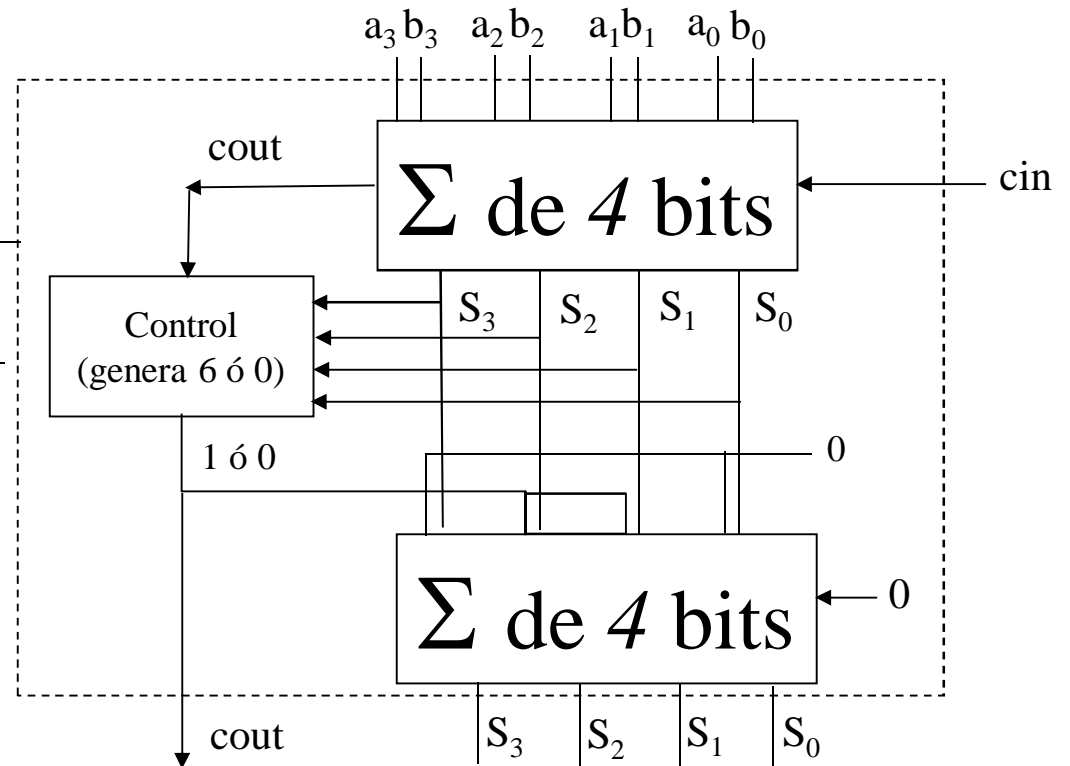
# Sumadors

**Suma en BCD:** Sumar en binari cada grup de 4 bits i corregir sumant-hi quan el resultat és major que 9 o es produïxca ròssec

**Exemple: 9 + 8 = 17**

en BCD:

	1001	( 9)
+	1000	( 8)
	0001 0001	(1 1)
	+ 0110	
	0001 0111	(1 7)





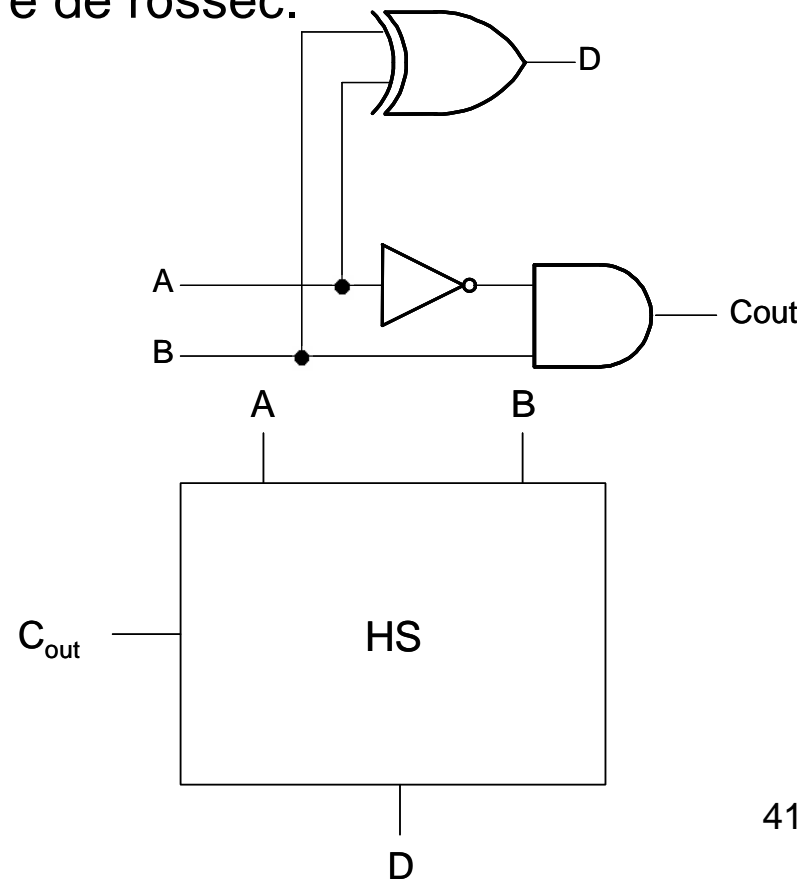
# Restadors

- ◆ **Definició:** Un **semirestador** (**half subtractor, HS**) en el nivell bit és un circuit combinacional capaç de restar dos bits, que proporciona com a eixides un bit de diferència i un altre de ròssec.

A	B	D	C <sub>out</sub>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \bar{A}B + A\bar{B} = A \oplus B$$

$$C_{out} = \bar{A}B$$



# Restadors

- ◆ **Definició:** Un **restador complet** (full subtractor, FS) en el nivell bit és un circuit combinacional capaç de restar dos bits i un ròssec d' entrada, que proporciona com a eixides el bit diferència i un bit de ròssec.

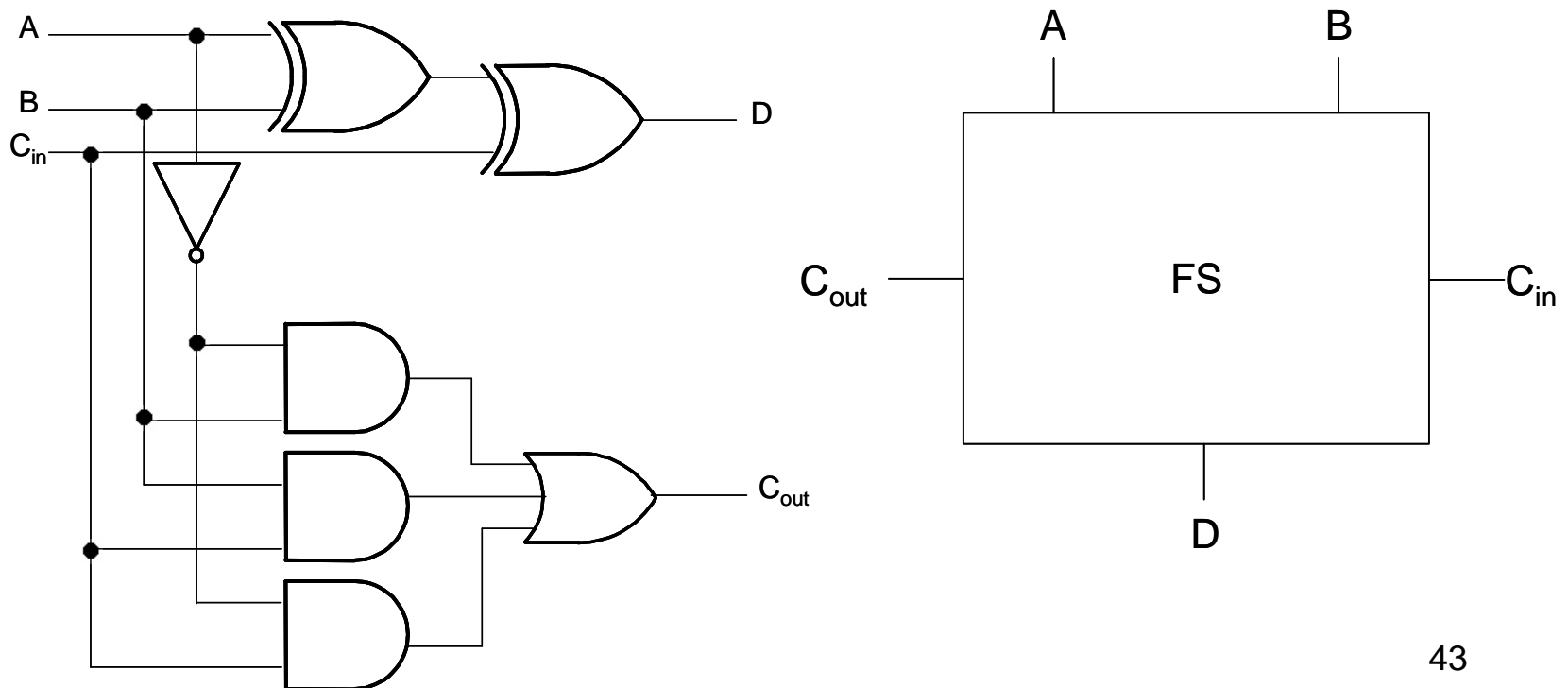
A	B	C <sub>in</sub>	D	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in} = A \oplus B \oplus C_{in}$$

$$C_{out} = \bar{A}B + \bar{A}C_{in} + BC_{in}$$

# Restadors

- ◆ **Definició:** Un **restador complet** (full subtractor, FS) en el nivell bit és un circuit combinacional capaç de restar dos bits i un ròssec d'entrada, que proporciona com a eixides el bit diferència i un bit de ròssec.



# Restadors

---

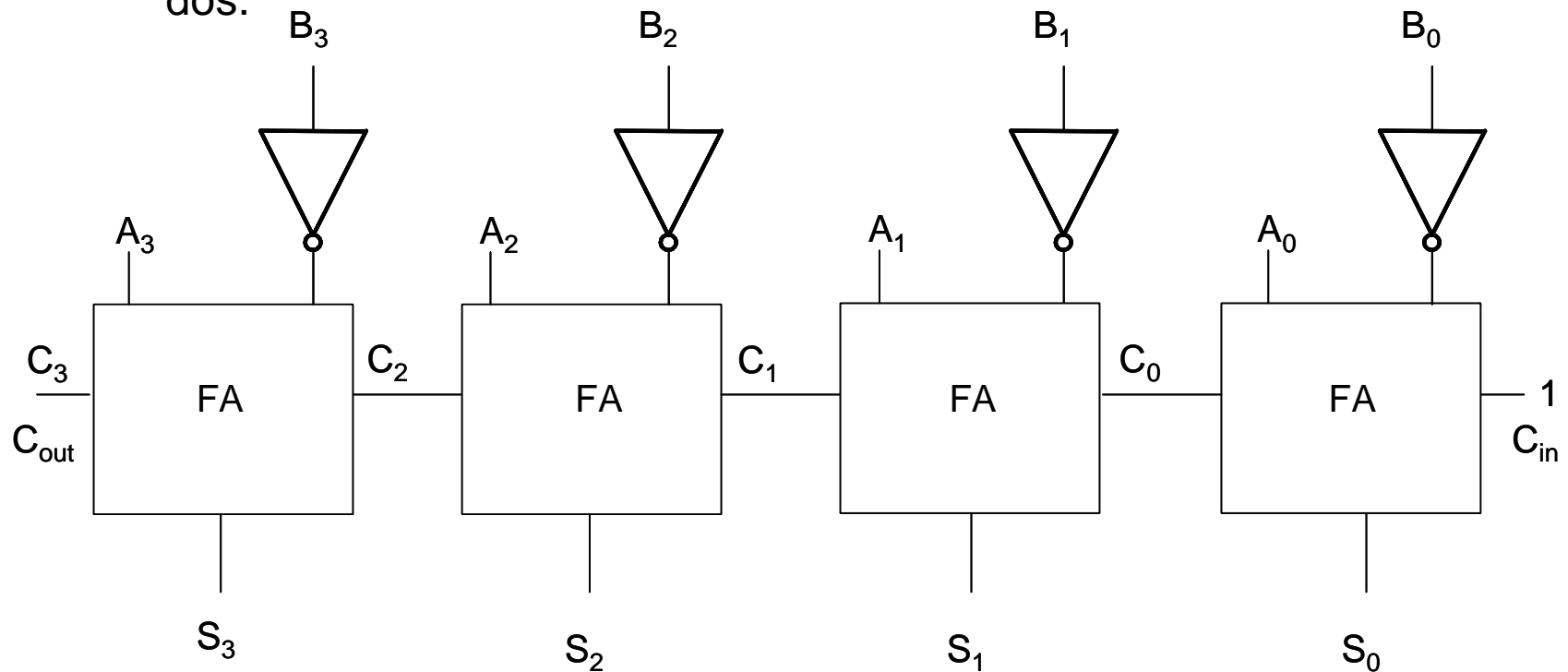
## ◆ La resta, un cas particular de la suma.

- No és necessari un circuit restador per a restar.
- La resta en complement es realitza de la mateixa forma que la suma.
- $A - B = A + (-B) = A + B_{C2} = A + (B_{C1} + 1) = A + \overline{B} + 1$

# Restadors

◆ La resta, un cas particular de la suma.

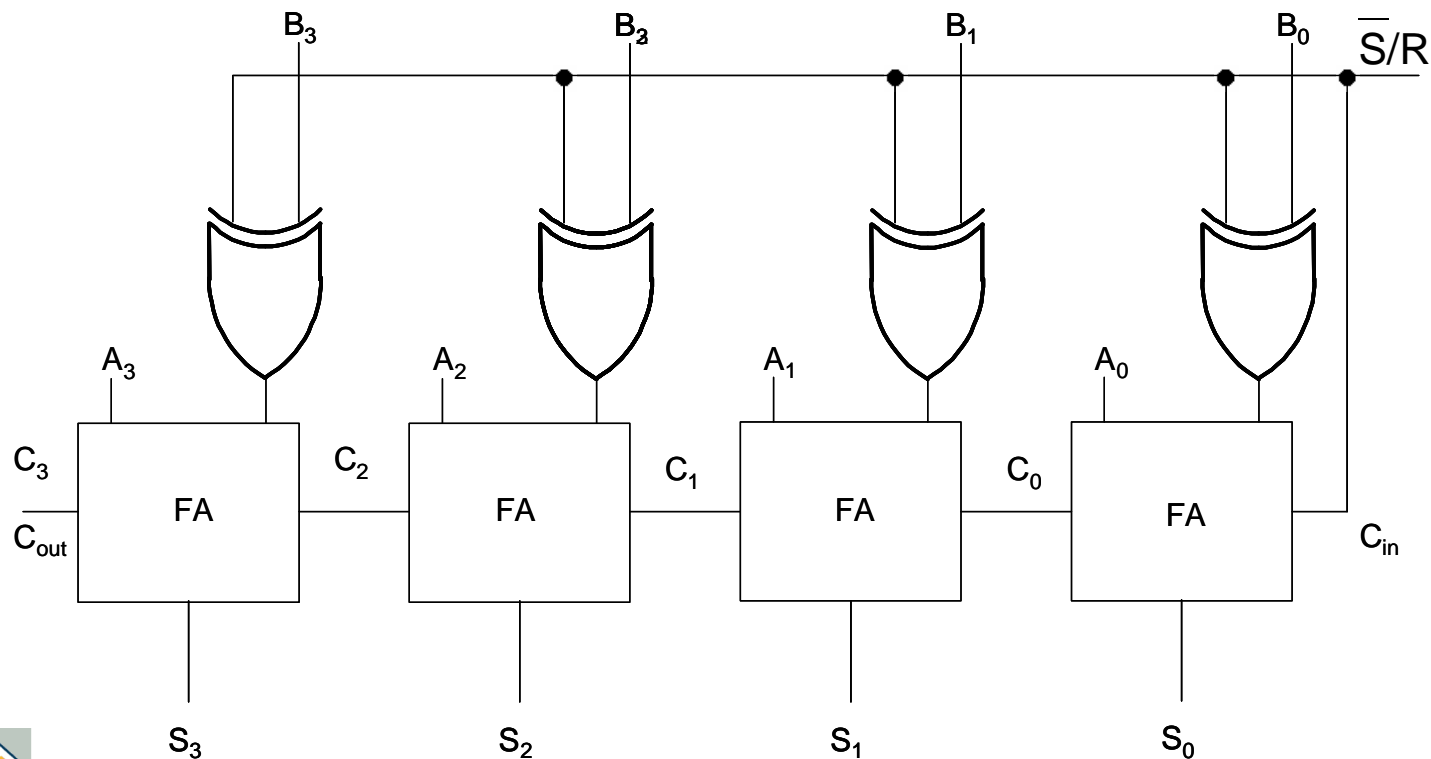
- **Exemple:** Resta de dues paraules de 4 bits, amb ús del complement a dos.



# Restadors

## ◆ Sumador/Restador.

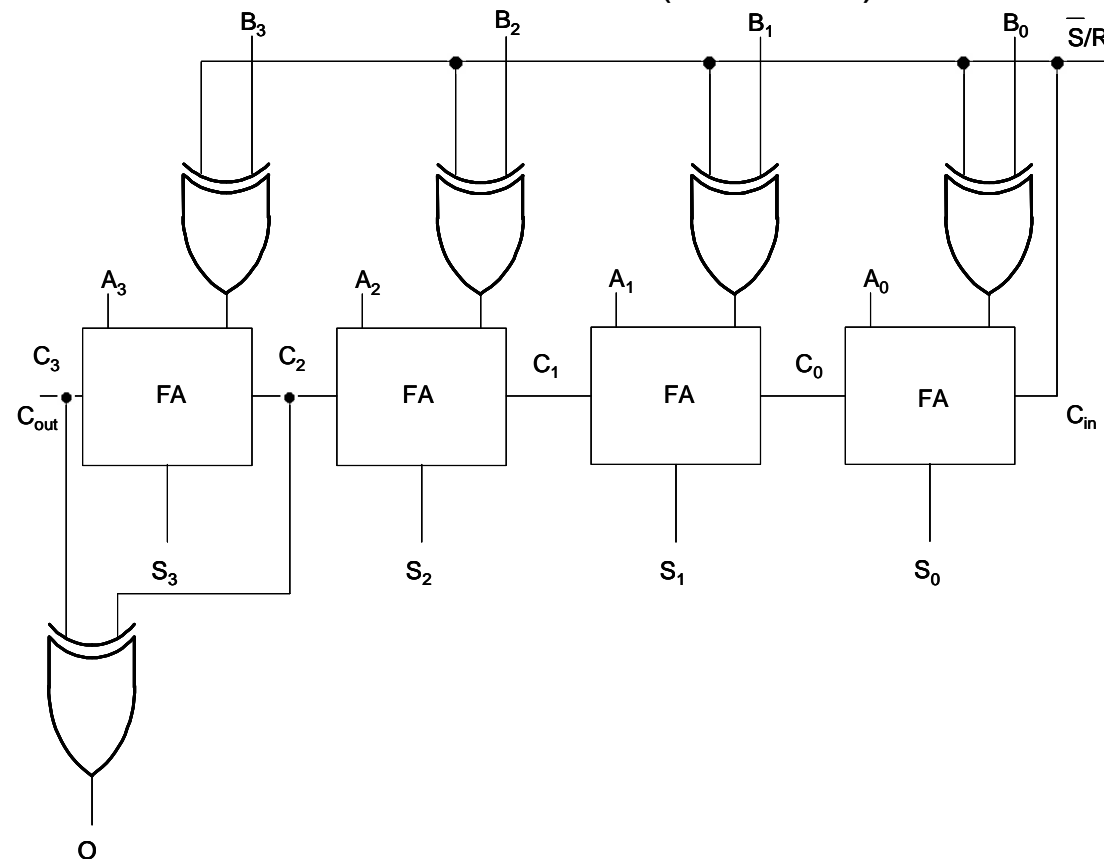
- **Exemple:** Circuit sumador/restador de paraules de 4 bits en complement a 2.



# Restadors

◆ Sumador/Restador.

- **Example:** Circuit sumador/restador de paraules de 4 bits en complement a 2, amb detecció de desbordament (*Overflow*) .



# Convertidors de codi

---

- ◆ **Definició:** Un **convertidor de codi** és un dispositiu capaç de transformar informació numèrica expressada en un determinat codi a un altre.
- ◆ Exemples de convertidors són:
  - binari-BCD / BCD-binari.
  - Gray-binari / binari-Gray.
- ◆ Formes habituals d'implementar un convertidor:
  - Utilitzant circuits sumadors.
  - Emprant portes lògiques.



# Convertidors de codi

---

## ◆ Exemple: Convertidor BCD-binari.

- Per a construir un convertidor BCD-binari utilitzant sumadors, els passos que cal seguir són:
  - ❖ Establir una taula que relaciona els pesos dels bits d'entrada i eixida.
  - ❖ Obtenir les equacions de conversió, a partir de la taula anterior.
  - ❖ Implementar el circuit amb blocs sumadors.

# Convertidors de codi

---

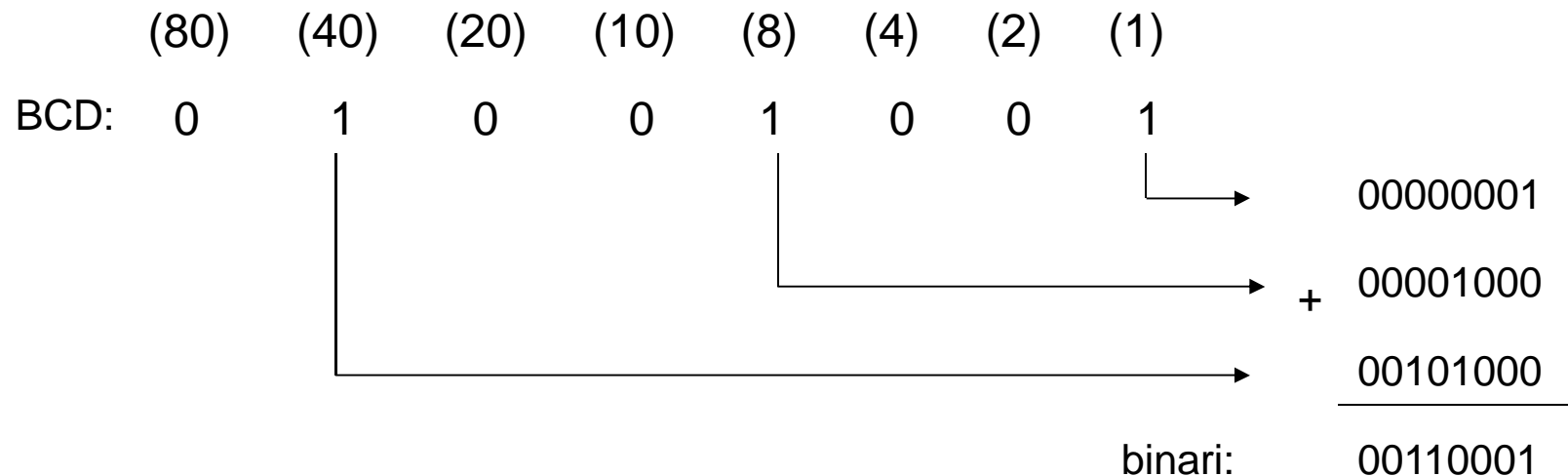
## ◆ Exemple: Convertidor BCD-binari.

- Per a construir un convertidor BCD-binari utilitzant sumadors, els passos que cal seguir són:
  - ❖ El valor, o pes de cada bit d'un número BCD es representa mitjançant un número binari.
  - ❖ Sumem totes les representacions binàries dels pesos dels bits del número BCD que són "1".
  - ❖ El resultat de la suma és l' equivalent binari del número BCD.

# Convertidors de codi

## ◆ Example: Convertidor BCD-binari.

- Per a obtenir el número binari complet cal sumar els números binaris que representen els bits del número BCD.
  - Example: Conversió del número  $49_{10}$ , de BCD a binari.



# Convertidors de codi

## ◆ Exemple: Convertidor BCD-binari.

- La taula següent indica quins bits BCD activen cada bit binari (BN).

BCD	BN6(64)	BN5(32)	BN4(16)	BN3(8)	BN2(4)	BN1(2)	BN0(1)
A0 (1)	0	0	0	0	0	0	1
A1 (2)	0	0	0	0	0	1	0
A2 (4)	0	0	0	0	1	0	0
A3 (8)	0	0	0	1	0	0	0
B0 (10)	0	0	0	1	0	1	0
B1 (20)	0	0	1	0	1	0	0
B2 (40)	0	1	0	1	0	0	0
B3 (80)	1	0	1	0	1	0	0

# Convertidors de codi

## ◆ Exemple: Convertidor BCD-binari natural.

- A partir del mètode descrit i la taula anterior, obtenim les equacions que relacionen el codi BCD amb el binari natural.

$$A0 = BN0$$

Ci: ròssecs generats en la suma de bits

$$A1 + B0 = (C1) BN1$$

$$A2 + B1 + C1 = (C2) BN2$$

$$A3 + B0 + B2 + C2 = (C3) BN3$$

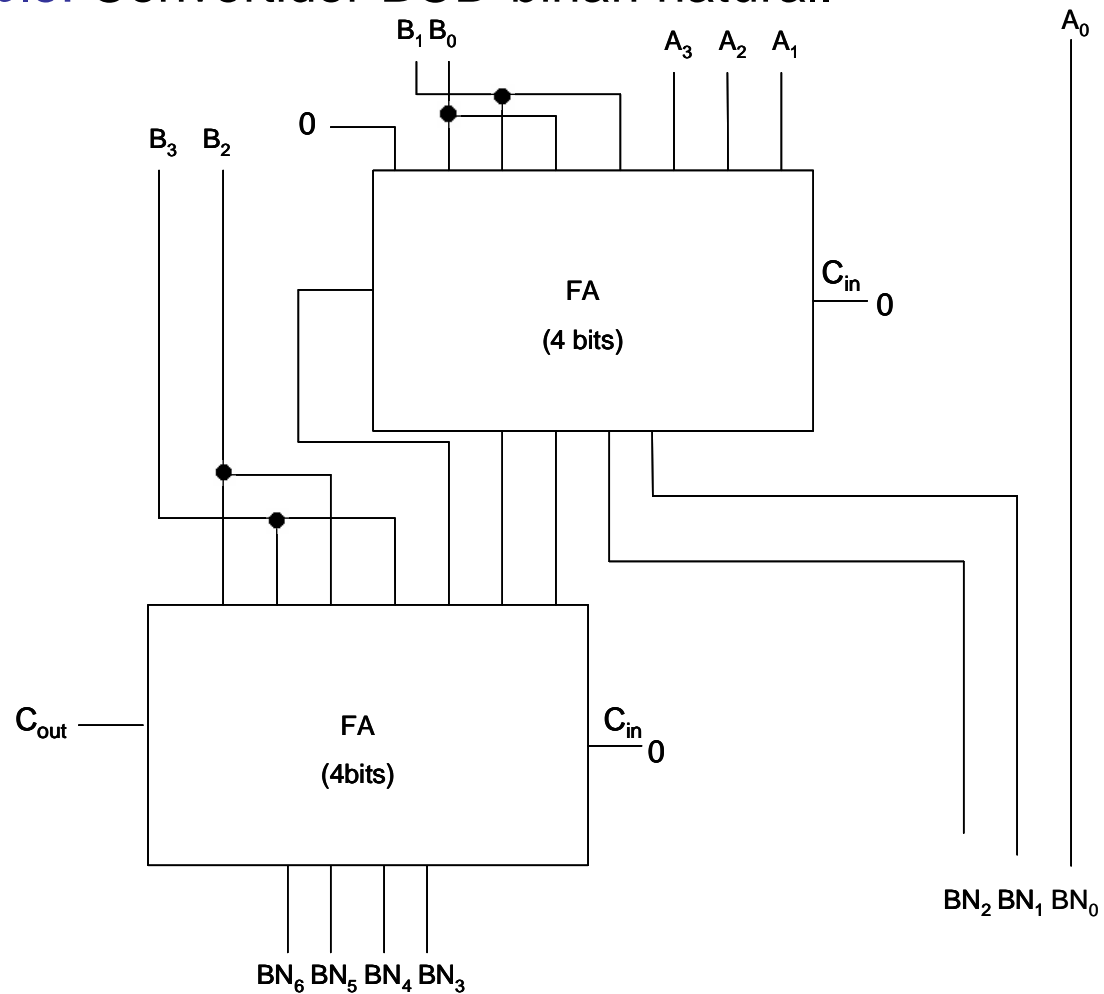
$$B1 + B3 + C3 = (C4) BN4$$

$$B2 + C4 = (C5) BN5$$

$$B3 + C5 = (C6) BN6$$

# Convertidors de codi

◆ Exemple: Convertidor BCD-binari natural.



# Convertidors de codi

◆ **Exemple:** Convertidor binari-Gray.

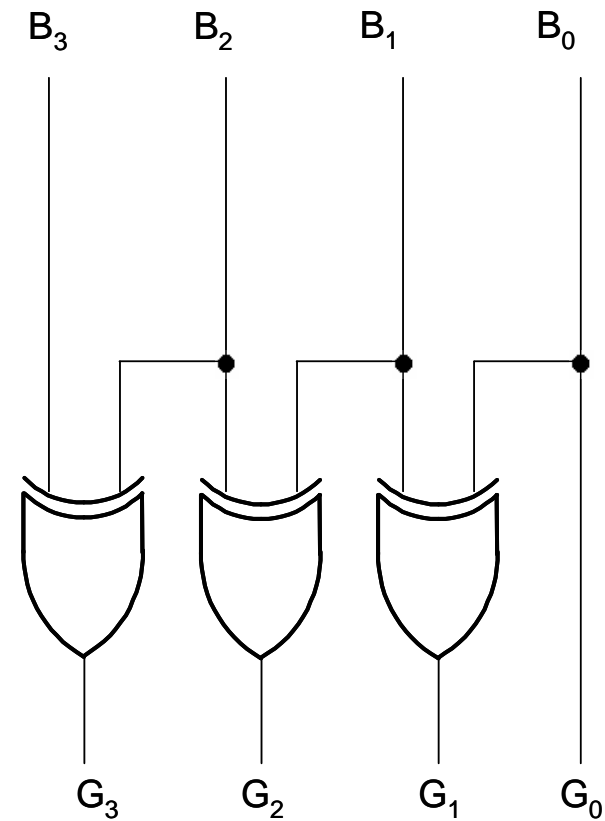
$$G_0 = B_0$$

$$G_1 = B_0 \oplus B_1$$

$$G_2 = B_1 \oplus B_2$$

$$G_3 = B_2 \oplus B_3$$

$$G_n = B_{n-1} \oplus B_n$$



# Convertidors de codi

## ◆ Exemple: Convertidor Gray-binari

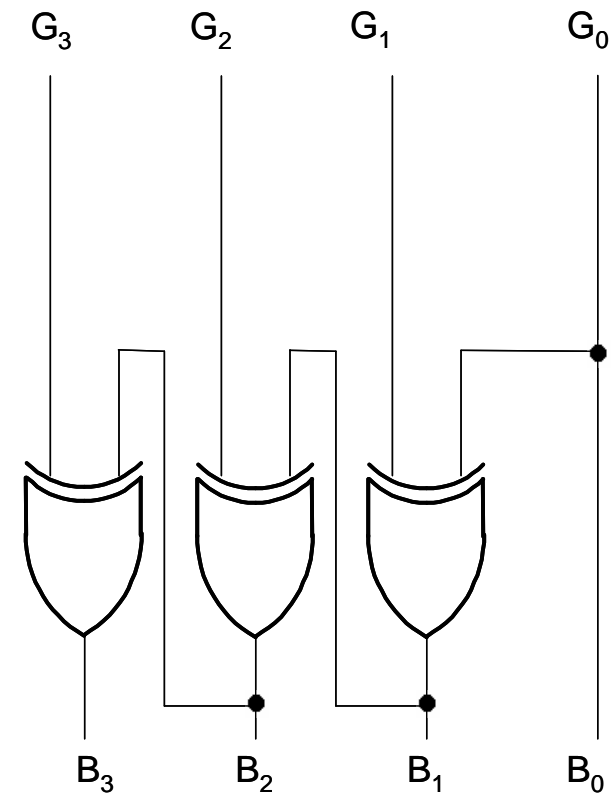
$$B_0 = G_0$$

$$B_1 = G_0 \oplus G_1 = G_1 \oplus B_0$$

$$B_2 = G_0 \oplus G_1 \oplus G_2 = G_2 \oplus B_1$$

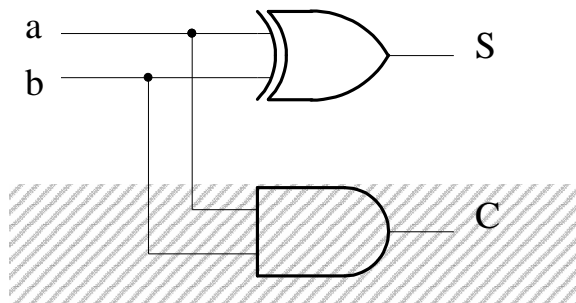
$$B_3 = G_0 \oplus G_1 \oplus G_2 \oplus G_3 = G_3 \oplus B_2$$

$$B_n = G_0 \oplus G_1 \oplus \dots \oplus G_n = G_n \oplus B_{n-1}$$





# Generadors/detectors de paritat

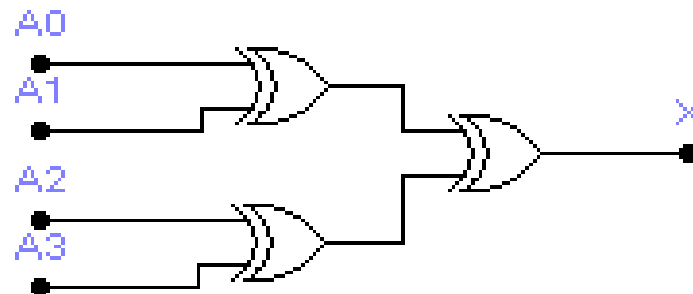


Són sistemes combinacionals que compten el nombre d' "1" presents en les seues entrades. Si aquest nombre és par l'eixida és 0, si és senar és 1.

**Procediment:**

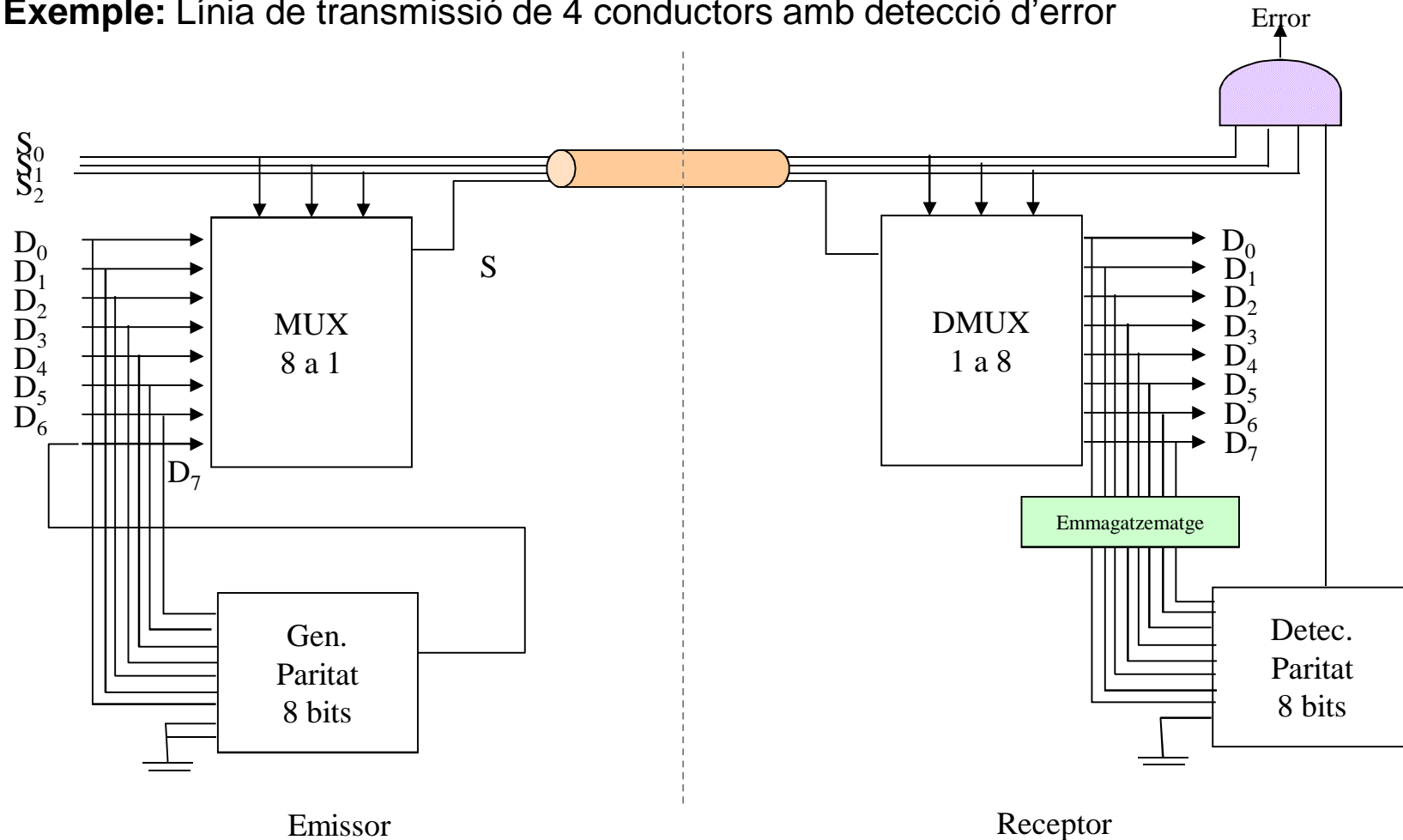
Sumar descartant el ròssec.

Associant més portes XOR es construeixen detectors de més bits

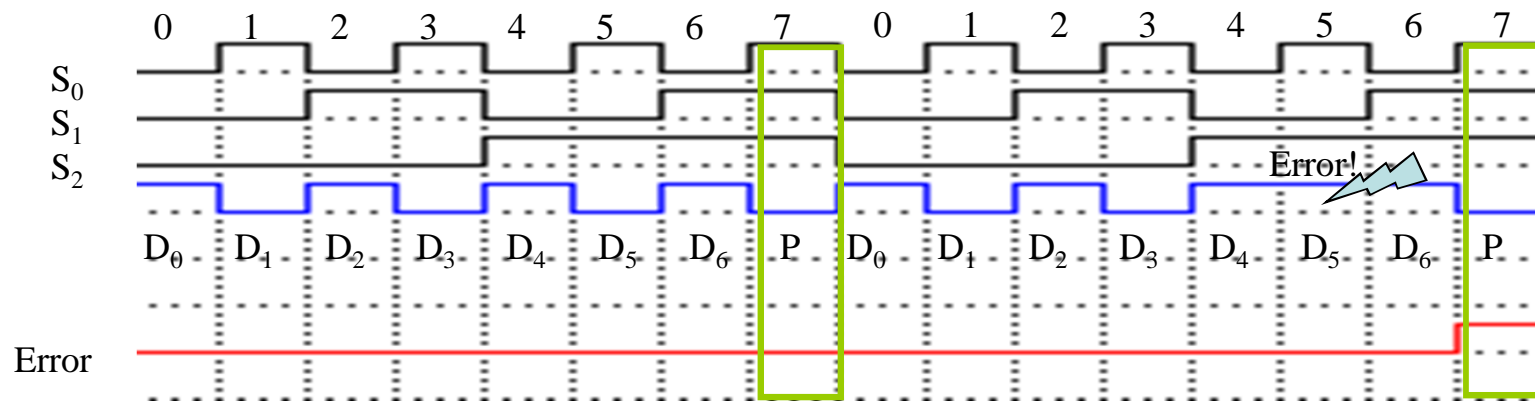


# Generadors/detectors de paritat

**Exemple:** Línia de transmissió de 4 conductors amb detecció d'error



# Generadors/detectors de paritat



Transmissió de dues paraules de 8 bits. Error en la 2<sup>a</sup> paraula

# Full de característiques

---

- ◆ **Definició:** Un **full de característiques** (*data sheet*) és el document que explica detalladament el funcionament i les característiques d'un dispositiu electrònic. Normalment, s'hi troba informació referent a:
- Nom del fabricant.
  - Número i nom del producte.
  - Descripció funcional del dispositiu.
  - Condicions de funcionament (temperatures, voltatges...).
  - Recomanacions d'ús.

# Full de característiques

---

## ◆ Alguns exemples de components combinacionals:



74F148. 8-to-3 line priority encoder



74HC147. 10-to-4 line priority encoder



74HC151. 8 input multiplexer.



74HC157. Quad 2-Input Data Selectors/Multiplexer



74HC138. 3-to-8 line decoder/demultiplexer



74HC154. 4-to-16 line decoder/demultiplexer



74LS139. Dual 2-line to 4 line decoder/demultiplexer



74HC42. BCD to decimal decoder

# Full de característiques

---

## ◆ Alguns exemples de components combinacionals:



74LS147. BCD to 7-segment decoder



DM74HC85. 4 bit Magnitude Comparator



74LS83. 4 bit Binary Adder with Fast Carry



SN74LS183. Dual Carry-Save Full Adder



DM74184/DM74185A. BCD to Binary and Binary to BCD Converters



74LS280. 9 bit odd/even parity generators/checkers

# Full de característiques

---

## ◆ Alguns exemples de components combinacionals:



74F148. Codificador amb prioritat 8:3



74HC147. Codificador amb prioritat 10:4



74HC151. Multiplexor de 8 entrades



74HC157. Multiplexor 2:1 (Quàdruple o 4 en 1)



74HC138. Descodificador/Demultiplexor 3:8



74HC154. Descodificador/Desmultiplexor 4:16



74LS139. Descodificador/Desmultiplexor 2:4 (Dual o 2 en 1)



74HC42. Descodificador BCD-Decimal

# Full de característiques

---

## ◆ Alguns exemples de components combinacionals:



74LS147. Decodificador BCD-7 segments



DM74HC85. Comparador de magnituds, 4 bits



74LS83. Sumador binari amb anticipació de ròssec (CLA), 4 bits



SN74LS183. Sumador complet amb salvaguarda de ròssec (CSA), (Dual o 2 en 1)



DM74184/DM74185A. Convertidors BCD-Binari i Binari-BCD



74LS280. Generador/Detector de paritat (parell/imparell), 9 bits



# Referències

---

- T.L. Floyd. *Fundamentos de sistemas digitales*, Prentice-Hall, 2000.
- P. De Miguel Anasagasti. *Fundamentos de los computadores*. Paraninfo, 2004.
- A. Prieto et al. *Introducción a la informática*, McGraw-Hill, 2006.
- M. Morris Mano y C.R. Kime. *Fundamentos de diseño lógico y computadoras*. Prentice-Hall, 2005.
- J.M. Angulo. *Fundamentos y estructura de computadores*, Paraninfo, 2001.
- A. Lloris, A. Prieto, L. Parrilla. *Sistemas Digitales*. MC Graw Hill, 2003.
- M. D. Ercegovac, T. Lang. *Digital Arithmetic*, Morgan Kaufman, 2004.
- B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 2000.