

Activitats

3. Enumera les tècniques de simplificació de funcions lògiques.

4. Simplifica, mitjançant l'àlgebra de Boole, les funcions següents:

$$F = \bar{a} + \bar{b} + \bar{c} + \overline{abc}$$

$$F = a + c\bar{c} + b + \bar{b}$$

$$F = \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc$$

5. Demuestra que:

$$ad + \bar{c} = (a + \bar{c})(\bar{c} + d)$$

6. Representa en una taula de Karnaugh la funció:

$$F = abc\bar{d} + \bar{a}bc\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c}\bar{d}$$

7. Representa en una taula de Karnaugh la funció:

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

8. Simplifica, utilitzant mapes de Karnaugh, la funció:

$$F = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c}$$

9. Simplifica, utilitzant mapes de Karnaugh, la funció:

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

10. A partir d'aquesta taula de la veritat, obtén la funció en la forma més simplificada possible i la seva implementació.

a	b	c	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

2.3. Circuits de tractament numèric

S'anomenen *circuits de tractament numèric* un grup molt important de circuits lògics combinacionals capaços d'efectuar operacions aritmètiques amb valors binaris.

Industrialment es presenten en forma de **xip**, també anomenat *circuit integrat*, una estructura formada per un gran nombre de portes lògiques realitzades i connectades sobre una pastilla de material semiconductor, amb una estructura tan complexa que no entrarem en el seu estudi intern. Representarem els circuits integrats com una mena de *caixa negra*, amb un conjunt de *portes* o connexions accessibles des de l'exterior.

El semisumador, el sumador i el restador

El semisumador

El semisumador és un dels circuits lògics combinacionals més senzills, que permet fer la suma de **dos** dígits binaris.

Per sumar dos dígits binaris, trobem quatre possibilitats:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

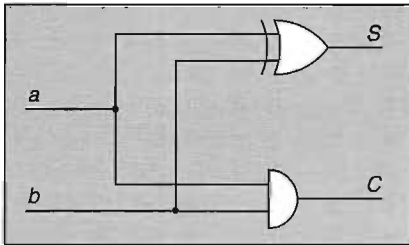
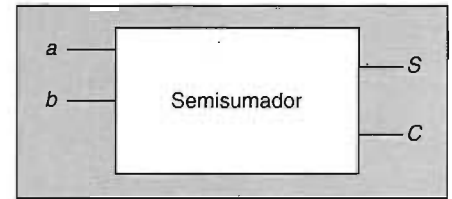
$$1 + 1 = 0$$

amb un 1 de transport o ròssec

El circuit que fa aquesta operació té dos senyals d'entrada a i b , i dos senyals de sortida S (resultat de la suma binària dels dos dígits) i C (valor del ròssec).

La taula de la veritat d'un semisumador és:

Semisumador			
a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



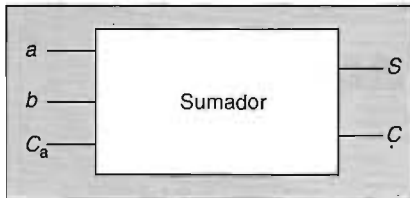
D'aquesta taula es poden deduir les equacions lògiques del circuit:

$$S = \bar{a}b + a\bar{b} = a \oplus b$$

$$C = ab$$

Aquesta funció semisuma pot implementar-se amb portes lògiques.

El sumador

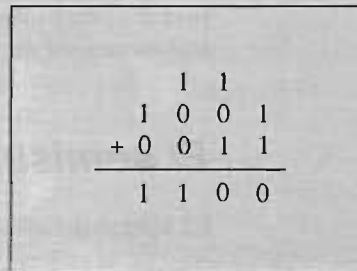


Aquest circuit pot fer la suma de dues xifres expressades en codi binari quan aquestes xifres tenen més d'un bit. Aleshores, cal que el circuit tingui en compte si, de la suma parcial anterior, hi ha ròssec.

Per tenir en compte el ròssec, el circuit té tres variables d'entrada a , b i C_a (valor del ròssec de la suma anterior), i dues de sortida, S (resultat de la suma binària actual) i C (valor del ròssec de la suma actual).

Exemple 5

En la suma de dos nombres binaris de 4 bits, identifica el valor que prenen les variables d'entrada i de sortida en cada etapa:



En la primera columna, com que no hi ha hagut cap suma anterior, el valor de C_a és zero ($C_a = 0$) i les variables d'entrada són $a = 1$ i $b = 1$. El resultat de la suma i del ròssec són $S = 0$ i $C = 1$.

En la segona columna: $C_a = 1$ $a = 0$ $b = 1$ $S = 0$ $C = 1$

En la tercera columna: $C_a = 1$ $a = 0$ $b = 0$ $S = 1$ $C = 0$

I en la suma de l'últim bit: $C_a = 0$ $a = 1$ $b = 0$ $S = 1$ $C = 0$

Sumador				
C_a	a	b	S	C
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

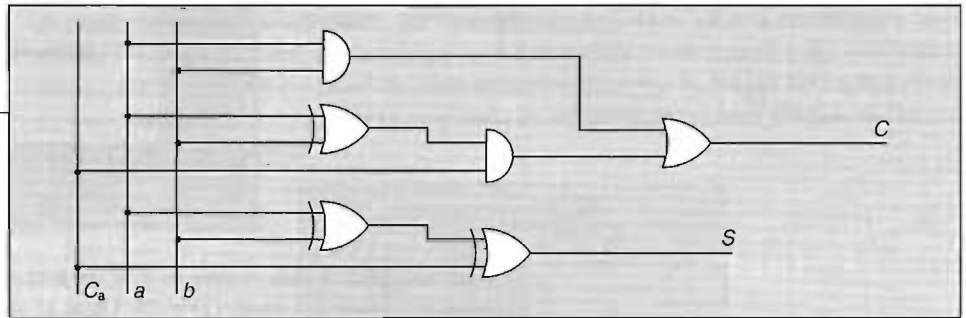
A l'esquerra, es mostra la taula de la veritat d'un sumador i se n'obtenen les funcions lògiques de sortida.

$$S = \bar{a} b \bar{C}_a + a \bar{b} \bar{C}_a + \bar{a} \bar{b} C_a + a b C_a = a \oplus b \oplus C_a$$

$$C = a b \bar{C}_a + \bar{a} b C_a + a \bar{b} C_a + a b C_a = a b + C_a (\bar{a} b + a \bar{b}) = a b + C_a (a \oplus b)$$

Convé observar que en la taula hi ha dues parts. En la primera, el transport o ròssec de la suma anterior C_a és zero, i les variables de sortida tenen els mateixos valors que en el cas del semisumador. En la segona part hi ha ròssec ($C_a = 1$) i els valors a la sortida són ben diferents.

Una possibilitat d'implementació és la que es mostra en la figura, fent servir portes AND i EXOR:

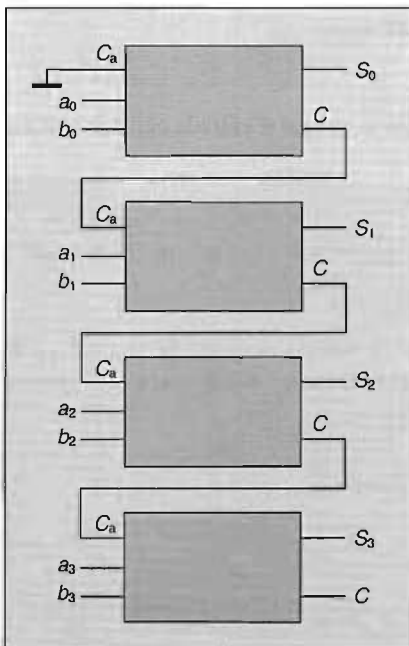


Implementació del sumador

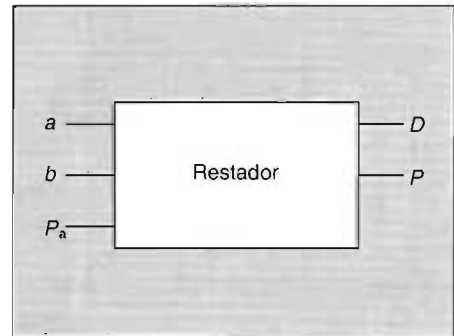
Quan cal sumar nombres formats per més d'un bit són necessaris diversos sumadors d'un bit connectats en cascada; se'ls anomena **sumadors en paral·lel**, ja que tots els bits dels nombres que es volen sumar estan presents alhora a les entrades dels sumadors.

El restador

Aquests circuits són molt semblants als sumadors, però en un restador les entrades són els dígit a i b , que ha de restar, i el valor del préstec (P_a) de la resta feta anteriorment. Com a sortides dóna la diferència (D) entre els dígit i el valor del préstec (P) d'aquesta diferència.



Restador				
P_a	a	b	D	P
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1



Les funcions lògiques d'un restador són:

$$D = a \oplus b \oplus P_a$$

$$P = \bar{a} P_a + \bar{a} b + b P_a$$

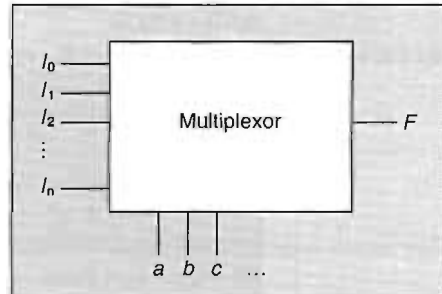
A la pràctica, els restadors es fabriquen fent servir sumadors, així no cal tenir dos circuits diferents. Restar a un número un altre és el mateix que sumar-li l'oposat del nombre que se li vol restar: $A - B = A + (-B)$. La subtracció esdevé, aleshores, una addició de nombres positius i negatius.

Sumador en paral·lel de nombres de quatre bits

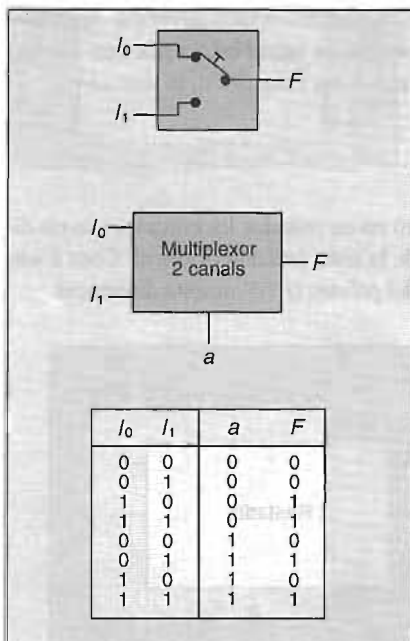
Multiplexors i desmultiplexors

Un multiplexor és un circuit lògic combinacional amb moltes entrades ($I_0, I_1, I_2, \dots, I_n$) i una sola sortida F .

Amb uns **senyals de control** (a, b, c, \dots) es pot seleccionar un dels senyals d'entrada i portar-lo fins a la sortida. Aquest funcionament és comparable al d'un commutador rotatiu, que recull un senyal d'alguna de les seves entrades i el porta fins a la seva sortida.



El multiplexor més senzill és el d'un únic senyal de control, que ha de tenir dues variables d'entrada o *canals* ($2^1 = 2$). Quan el senyal de control té valor 0, el multiplexor dóna a la sortida el valor del senyal d'entrada I_0 , independentment del valor que tingui I_1 . Quan el senyal de control és 1, el senyal que passa a la sortida és I_1 .



Aquest multiplexor de dues variables és comparable amb un commutador, que passa d'una posició a una altra segons el nivell lògic del seu senyal de control. De fet, industrialment poden construir-se amb portes lògiques o amb interruptors electrònics.

El multiplexor més comú és el de 16 canals –s'anomena *16 a 1*–, que té:

- 16 bits d'entrada: $I_0, I_1, I_2, \dots, I_{15}$
- 4 bits de control: a, b, c, d ($2^4 = 16$ combinacions)
- 1 bit de sortida: F

Cada combinació de a, b, c i d deixa passar només un senyal d'entrada cap a la sortida.

Exemple 6

Troba el senyal de sortida d'un multiplexor de 16 entrades que rep els senyals de control $a = 0, b = 0, c = 1$ i $d = 1$.

El nombre binari 0011 correspon al nombre 3 en el sistema decimal: $0011_2 = 3_{10}$, de manera que el circuit mostrarà a la sortida el senyal d'entrada corresponent a la posició I_3 . Si els senyals d'entrada fossin:

$I_0 = 0$ la sortida tindria valor $F = 0$

$I_1 = 1$

$I_2 = 1$

$I_3 = 0$

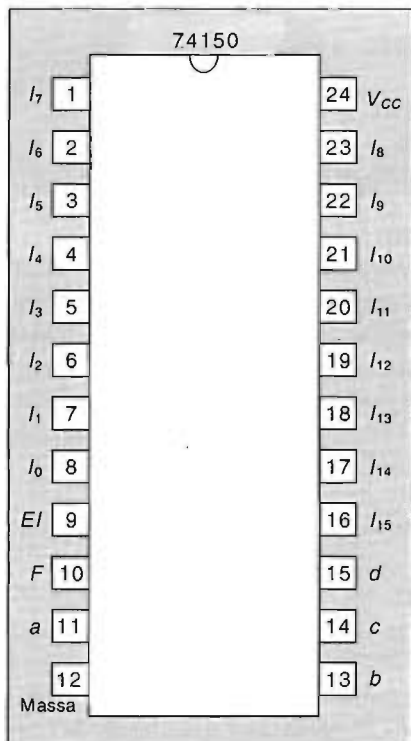
....

Per als casos:

$$a = 0 \quad b = 0 \quad c = 0 \quad d = 0 \quad 0000_2 = 0_{10} \quad \rightarrow \quad F = I_0$$

$$a = 0 \quad b = 1 \quad c = 1 \quad d = 1 \quad 0111_2 = 7_{10} \quad \rightarrow \quad F = I_7$$

$$a = 1 \quad b = 1 \quad c = 1 \quad d = 1 \quad 1111_2 = 15_{10} \quad \rightarrow \quad F = I_{15}$$



El circuit integrat d'un multiplexor real, com ara el 74150, es representa a la figura de l'esquerra.

Des de la pota 1 fins a la 8 i des de la 16 fins a la 23 són les connexions per als 16 senyals d'entrada. A les potes 11, 13, 14 i 15 es connecten els quatre senyals de control i per la pota 10 s'accedeix al senyal de sortida. Les altres potes corresponen a:

- Pota 24: connexió amb el valor de tensió positiva necessari per activar el circuit integrat (+5 V).
- Pota 12: connexió amb el valor de tensió nul (massa).
- Pota 9: és l'entrada d'un senyal d'inhibició (generalment s'anomena *EI: enable input*), que serveix per bloquejar, o deixar en suspensió o espera, el circuit.

Segons el model de multiplexor que s'esculli, les entrades i sortides poden ser les negacions del senyal corresponent; per exemple, aquest 74150 ofereix en realitat a la seva sortida *F* negada, i el senyal d'inhibició també està negat (això significa que el circuit funciona quan el senyal d'inhibició és 0 i es desactiva quan val 1). És per això que cada multiplexor disponible al mercat té la seva taula de la veritat, en la qual s'indica el funcionament de cada multiplexor.

<i>EI</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>F</i>
<i>L</i>	0	0	0	0	\bar{I}_0
<i>L</i>	0	0	0	1	\bar{I}_1
<i>L</i>	0	0	1	0	\bar{I}_2
<i>L</i>	0	0	1	1	\bar{I}_3
<i>L</i>	0	1	0	0	\bar{I}_4
<i>L</i>	0	1	0	1	\bar{I}_5
<i>L</i>	0	1	1	0	\bar{I}_6
<i>L</i>	0	1	1	1	\bar{I}_7
<i>L</i>	1	0	0	0	\bar{I}_8
<i>L</i>	1	0	0	1	\bar{I}_9
<i>L</i>	1	0	1	0	\bar{I}_{10}
<i>L</i>	1	0	1	1	\bar{I}_{11}
<i>L</i>	1	1	0	0	\bar{I}_{12}
<i>L</i>	1	1	0	1	\bar{I}_{13}
<i>L</i>	1	1	1	0	\bar{I}_{14}
<i>L</i>	1	1	1	1	\bar{I}_{15}
<i>H</i>	X	X	X	X	<i>H</i>

L significa *low* i vol dir senyal amb valor baix (0).
H significa *high* i vol dir senyal amb valor alt (1).

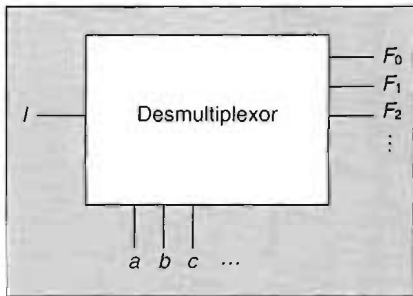
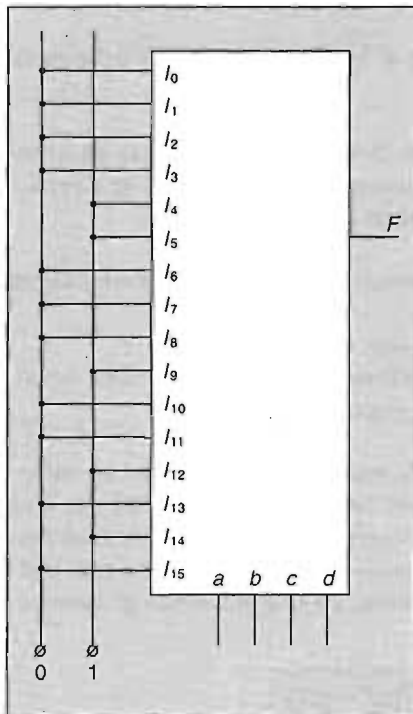
Observeu que, quan el senyal d'inhibició té valor alt (*H*), el circuit està desactivat, aleshores no importen els valors dels senyals de control (per això s'escriu una *X*), i el senyal de sortida per a aquesta situació pren el seu valor alt.

Ús de multiplexors per implementar una funció lògica donada

Estudiem ara el cas d'una funció lògica, la qual es vol implementar:

$$F = \bar{a} b \bar{c} d + \bar{a} b c \bar{d} + a b c \bar{d} + a b \bar{c} d + a \bar{b} \bar{c} d$$

La funció té quatre variables d'entrada, que combinades donen 16 combinacions possibles. Com que s'expressa en forma de *minterms*, en cada terme de la funció les variables són 1 (i 0 si estan negades).



$\bar{a} \bar{b} \bar{c} d = 0101$ (aquest nombre binari correspon a un 5 decimal: $0101_2 = 5_{10}$)

$$\bar{a} b \bar{c} \bar{d} = 0100 \quad (0100_2 = 4_{10})$$

$$a b c \bar{d} = 1110 \quad (1110_2 = 14_{10})$$

$$a b \bar{c} \bar{d} = 1100 \quad (1100_2 = 12_{10})$$

$$a \bar{b} \bar{c} d = 1001 \quad (1001_2 = 9_{10})$$

Com que té forma de *minterms*, cadascun dels cinc termes de la funció lògica són combinacions que tenen valor 1 de la variable de sortida; les combinacions de les variables que no apareixen en la funció lògica corresponen a un 0 de la funció lògica. Per implementar aquesta funció mitjançant un multiplexor, caldrà connectar-lo de la manera següent: les entrades que corresponen a 4, 5, 9, 12 i 14 en codi decimal estan a nivell lògic 1 (així, poden donar un 1 a la sortida F) i la resta s'ha de connectar a 0.

Un multiplexor 16 a 1 pot utilitzar-se per implementar qualsevol funció lògica de quatre variables d'entrada.

El demultiplexor

Es tracta d'un circuit lògic integrat que funciona de forma inversa al multiplexor. Té només un senyal d'entrada i molts de sortida (F_0, F_1, F_2, \dots). Mitjançant uns senyals de control es dirigeix el valor del senyal d'entrada cap a una de les sortides.

Exemple 7

Troba el senyal de sortida d'un demultiplexor de 16 canals de sortida que rep a l'entrada un senyal I (pot entendre's com una dada que el sistema ha de transmetre), si els senyals de control són: $a = 0, b = 0, c = 0, d = 1$.

El canal pel qual farà sortir aquesta dada depèn dels valors dels senyals de control. Així, si:

$$0001_2 = 1_{10}$$

el valor d' I passarà a estar disponible a la sortida F_1 , mentre que la resta de sortides queda en estat baix, per tant:

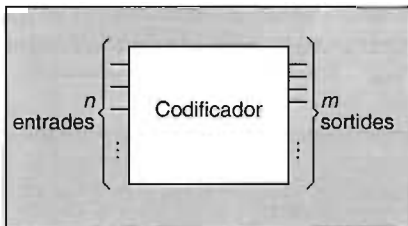
$$\text{si } I = 0 \text{ aleshores} \quad F_0 = 0 \quad F_1 = 0 \quad F_2 = 0 \quad F_3 = 0 \quad \dots \quad F_{15} = 0$$

$$\text{si } I = 1 \text{ aleshores} \quad F_0 = 0 \quad F_1 = 1 \quad F_2 = 0 \quad F_3 = 0 \quad \dots \quad F_{15} = 0$$

Codificadors i descodificadors

Són circuits combinacionals capaços de rebre una informació en un sistema numèric i traduir-la a un altre sistema.

Els **codificadors**, quan reben un senyal d'entrada actiu, ofereixen a la sortida aquest senyal, en qualsevol dels sistemes binaris existents. Els **descodificadors** passen informació d'un codi binari a un altre codi.



El codificador

Aquest circuit integrat té múltiples entrades, de les quals només una és activa, i converteix aquest senyal en una sortida binària d' m bits.

Un dels codificadors més usuals és el *codificador decimal a BCD*. Té deu entrades lògiques i quatre sortides A , B , C i D ; les entrades acostumen a ser pulsadors, com els teclats de les calculadores. Recordeu que un codificador real té més connexions, com ara les d'alimentació. De vegades disposa de més potes de les necessàries, que senzillament no s'utilitzen.

Si es pulsa el botó numerat amb 6, les sortides prenen els valors:

$A = 0 \quad B = 1 \quad C = 1 \quad D = 0$, que corresponen a la forma binària del decimal 6.

Si es pulsa el 5: $A = 0 \quad B = 1 \quad C = 0 \quad D = 1$

Si es pulsa el 9: $A = 1 \quad B = 0 \quad C = 0 \quad D = 1$

Els codificadors sense prioritat són circuits en què no pot activar-se més d'una entrada alhora, ja que provoquen codis binaris erronis a la sortida. En canvi, els codificadors amb prioritat estan preparats per a aquesta eventualitat i, quan s'activa més d'una entrada, el circuit ofereix a la sortida el codi binari de l'entrada de més valor (si s'activen les entrades 4 i 7, la sortida serà 0111, expressió binària del nombre 7).

El descodificador

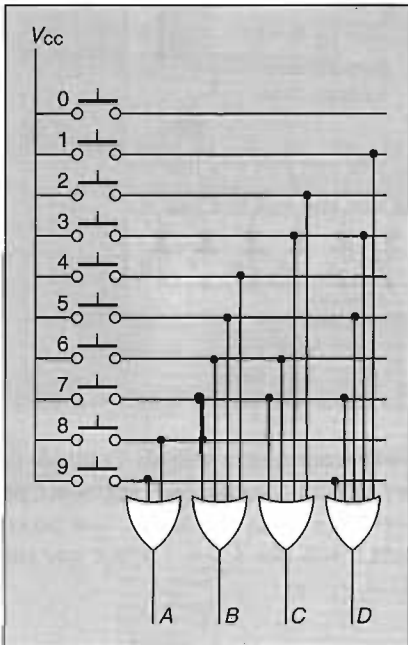
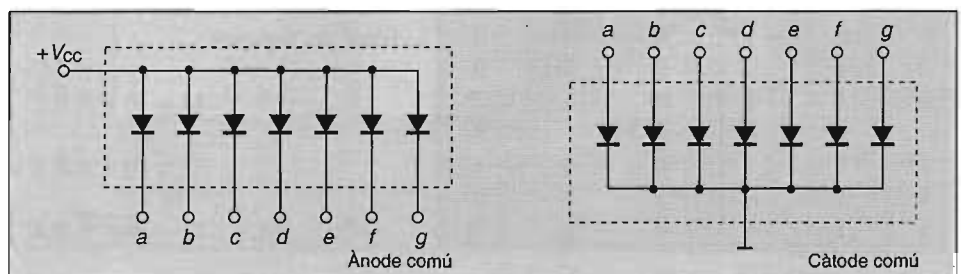
Un descodificador converteix una informació en codi binari a qualsevol altre tipus de codi; té n entrades i un nombre igual o inferior a 2^n sortides.

Els descodificadors poden classificar-se en *excitadors* i *no excitadors*. Els **descodificadors no excitadors** són els que només poden acoblar-se amb altres circuits de la seva mateixa família, perquè donen valors de corrent molt petits a les sortides. Els **descodificadors excitadors** donen prou corrent a les sortides per connectar-se amb circuits integrats d'altres famílies i amb altres dispositius, com ara làmpades, set segments, relès, transductors...

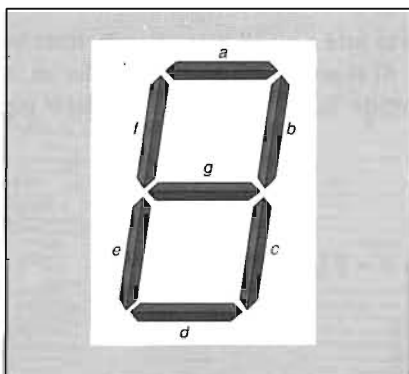
El descodificador excitador d'un set segments

Hi ha uns tipus de descodificadors capaços d'activar diferents visualitzadors. Els visualitzadors són dispositius que permeten una lectura molt fàcil dels senyals de sortida donats pels circuits. Un dels més coneguts i utilitzats és el set segments, també anomenat display.

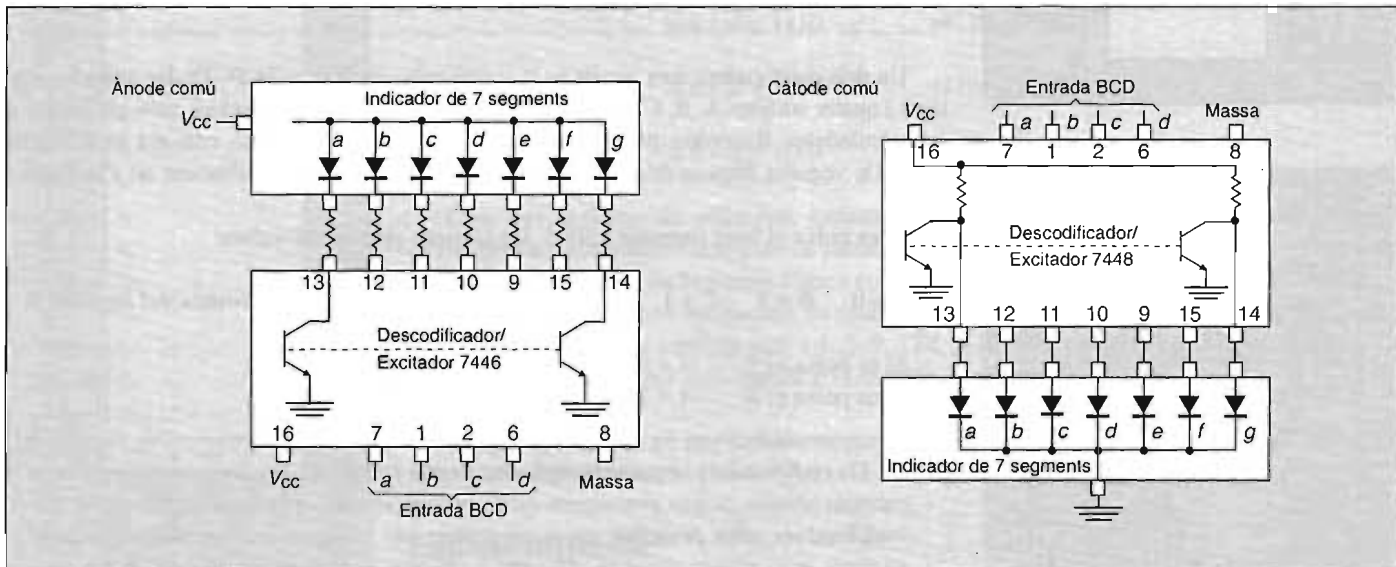
Es tracta d'un dispositiu format per set díodes LED col·locats de manera que puguin representar els deu signes (0, 1, 2,... 9) del nostre sistema decimal. Els díodes es marquen amb les lletres de la a a la g en l'ordre mostrat en la figura.



Esquema d'un codificador decimal a BCD



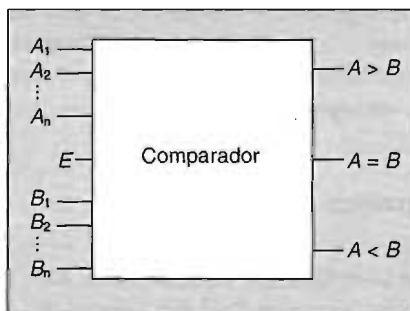
Els set díodes poden muntar-se en *ànode comú* (tots els ànodes dels LED estan connectats entre ells) o en *càtode comú* (tots els càtodes estan connectats entre ells); el fet que hi hagi dos tipus de set segments fa que també existeixin dues classes de descodificadors excitadors de *displays*.



Connexions descodificador-set segments

Aquests descodificadors excitadors de set segments tenen quatre senyals d'entrada (*a*, *b*, *c* i *d*), per on reben informació en codi binari, i set sortides (una per cada segment), per on es mostra la informació en codi binari; hi ha combinacions de les entrades que no engeguen cap segment. Així, una combinació d'entrada $a = 0, b = 1, c = 1, d = 0$ que correspon al nombre decimal 6, engegarà els LED *a*, *c*, *d*, *e*, *f*, *g*.

Comparador digital



Es tracta d'un dispositiu amb dues entrades, que pot comparar-les i donar com a sortida el resultat d'aquesta comparació.

Les entrades són nombres binaris de *n* bits (generalment expressats en binari natural).

Tot comparador té també una entrada d'inhibició, *E* (*enable*). Els seus senyals de sortida són tres, que indiquen els possibles resultats de la comparació feta:

$$A > B, \quad A = B \quad \text{i} \quad A < B$$

En el cas que es vulguin comparar dos nombres d'un sol bit, la taula de la veritat del comparador tindrà l'aspecte següent:

Comparador (de dos nombres d'un bit)					
<i>E</i>	<i>A</i>	<i>B</i>	$A > B$	$A = B$	$A < B$
0	0	0	0	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	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0

Observa que, quan el senyal d'inhibició està a nivell baix ($E = 0$), totes les sortides són zero, no es fa la comparació. Si el senyal d'inhibició té valor alt, el circuit fa la comparació i la mostra a la sortida. Aquesta taula de la veritat té tres funcions lògiques:

$$(A > B) = A \bar{B} E$$

$$(A = B) = \bar{A} \bar{B} E + A B E = E (\bar{A} \bar{B} + A B)$$

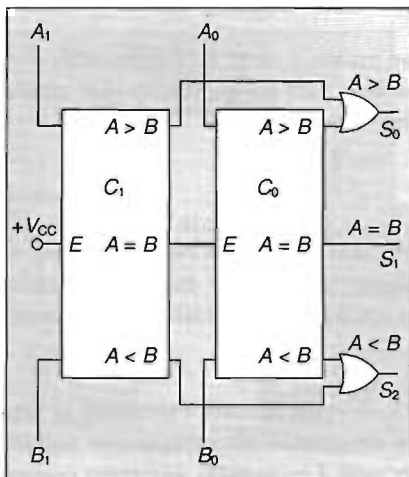
$$(A < B) = \bar{A} B E$$

Comparador de dues entrades de dos bits

Per dur a terme aquesta comparació, amb el mínim nombre de portes lògiques i el disseny més senzill, s'utilitzen dos comparadors de nombres d'un bit, connectats com mostra la figura:

Aquest circuit compara els nombres bit a bit, començant per comparar el bit de més pes. Si $A_1 > B_1$, aquest senyal de sortida alimentarà una porta OR que donarà com a resultat final S_0 ($A > B$). Si $A_1 < B_1$, el senyal passa a una altra porta OR que donarà com a sortida S_2 ($A < B$). Finalment, si $A_1 = B_1$, el circuit passa a comparar el segon bit. Per fer-ho, activa el segon comparador amb el senyal d'inhibició E i el cas es transforma en una comparació entre nombres d'un sol bit.

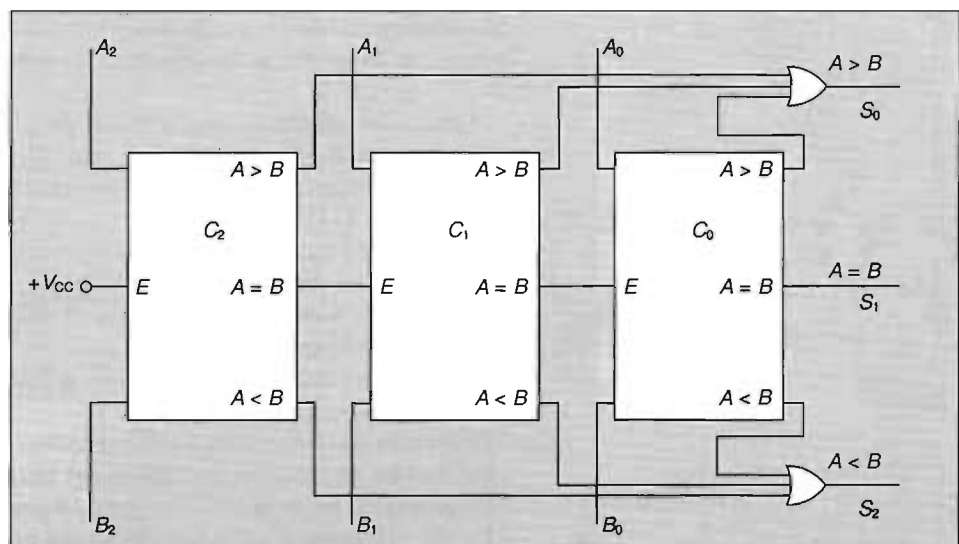
La seva taula de la veritat té l'aspecte següent:



E	A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	X	X	X	X	0	0	0
1	0	0	0	0	0	1	0
1	0	0	0	1	0	0	1
1	0	1	0	0	1	0	0
1	0	1	0	1	0	1	0
1	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1
1	0	1	1	0	0	0	1
1	0	1	1	1	0	0	1
1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	0
1	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0
1	1	0	1	0	0	1	0
1	1	0	1	1	0	0	1
1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	0

Comparador de dues entrades de tres bits

Com en el cas anterior, per comparar dos nombres de tres bits es fa servir un circuit amb comparadors d'un sol bit. El seu funcionament és similar al del cas estudiat anteriorment i el seu aspecte es mostra en la figura:

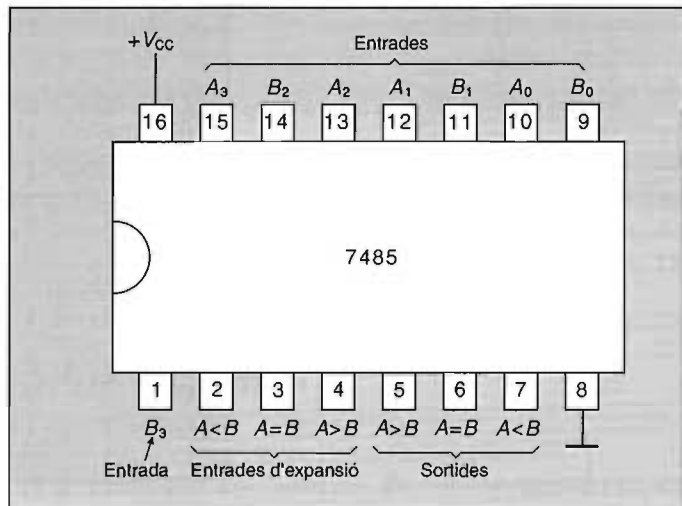


Tot i que els circuits que acabem d'estudiar s'han muntat a partir de comparadors d'un sol bit, comercialment hi ha comparadors de quatre bits, amb els quals es poden muntar fàcilment circuits per comparar paraules de molts bits de longitud.

Comparadors amb entrades d'expansió

Són comparadors especials que es fan servir quan els nombres que es volen comparar tenen un nombre de bits superior al nombre d'entrades disponibles en el comparador. Aquests comparadors tenen, en lloc d'una entrada d'inhibició E , tres entrades d'expansió, anomenades: $A = B$, $A > B$ i $A < B$.

Aquestes entrades permeten connectar en cascada diversos circuits. Per fer-ho, es connecten les sortides d'un comparador a les entrades d'expansió del comparador següent (sempre començant pel que compara els bits de més pes). Les sortides de l'últim comparador són les sortides del sistema, i les entrades $A > B$ i $A < B$ del primer comparador han de connectar-se a 0. L'entrada $A = B$ del primer comparador actua com a entrada d'inhibició del sistema.

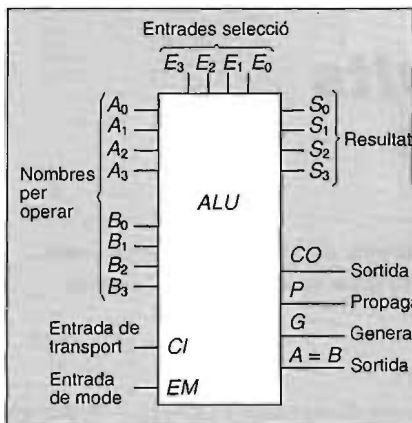


Unitat aritmètica i lògica (ALU)

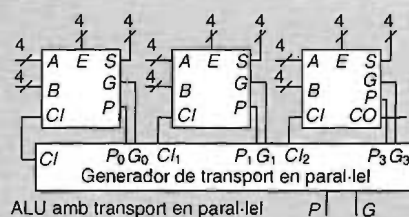
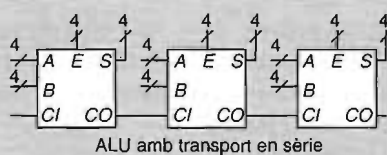
Hi ha una gran varietat de circuits numèrics, i utilitzar-los per fer operacions aritmètiques tal com els hem estudiat resulta complicat i costós, ja que són necessaris molts circuits diferents. Per evitar-ho, es van dissenyar les **unitats aritmètiques i lògiques**.

Una unitat aritmètica i lògica és un circuit que pot efectuar un gran nombre d'operacions diferents (addició, substracció, inversió, comparació, incrementació...); l'operació que es vol efectuar s'escull a través dels seus senyals de selecció. Disposa:

- D'entrades de dades: nombres o paraules binàries amb els quals es fan les operacions o manipulacions desitjades.
- D'entrada de transport: per on arriba el valor de transport d'operacions anteriors si es fan servir diverses ALU alhora.
- D'entrades de selecció: amb les quals es poden escollir les operacions que volem efectuar amb les dades introduïdes.
- De sortides de dades: on queden disponibles els resultats de les operacions.
- De sortides de transport: on s'ofereix el valor de transport en finalitzar l'operació.
- De generació i propagació de transport: per generar transport si es connecten diverses ALU.
- De *flags* (banderes): altres sortides en què hi ha disponible informació sobre les operacions lògiques o aritmètiques efectuades.

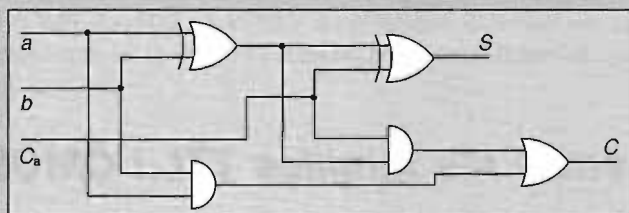


Si les dades que s'han d'operar són molt llargues, cal utilitzar més d'una ALU. Aquestes poden connectar-se en sèrie o en paral·lel (en aquest cas la velocitat de treball és superior, però cal disposar d'un circuit extra, anomenat **generador de transport en paral·lel**).



Activitats

- 11.** Demosta que aquesta implementació és també un sumador complet:



- 12.** Determina les funcions canòniques en forma de *minterms* de la taula de la veritat d'un sumador. Fes-ne la simplificació mitjançant les taules de Karnaugh.

- 13.** Demosta que són certes les funcions obtingudes de la taula de la veritat d'un restador.

- 14.** Defineix *multiplexor* i *desmultiplexor*.

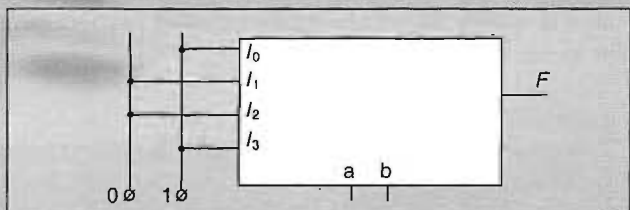
- 15.** De quina manera connectaries un multiplexor per poder obtenir la funció lògica:

$$F = abc + abc + \bar{a}b\bar{c}$$

- 16.** Obtingues, amb un multiplexor, la funció lògica:

$$F = abcd + abcd + \bar{a}bcd + \bar{a}bcd$$

- 17.** Troba la funció lògica que té el multiplexor de la figura, i fes-ne la implementació amb portes lògiques:



- 18.** Defineix *codificador* i *descodificador*.

- 19.** Quins circuits numèrics utilitzaries per passar de:

- Codi decimal a codi binari
- Codi binari a codi decimal
- Codi binari a codi octal
- Codi octal a codi binari

- 20.** Explica què és un descodificador excitador i dóna'n algun exemple.

- 21.** Escribeu la taula de la veritat d'un descodificador excitador d'un visualitzador de set segments.

- 22.** Comprova si aquesta implementació correspon a un comparador, i digues de quin tipus de comparador es tracta:

