

Problemes de circuitis lògics combinacionals

Fermín Sánchez Carracedo
Montse Peiron Guàrdia

1. Avalueu el valor de l'expressió $x + y \cdot (z + x')$ pels casos:
 - $[x \ y \ z] = [0 \ 1 \ 0]$
 - $[x \ y \ z] = [1 \ 1 \ 0]$
 - $[x \ y \ z] = [0 \ 1 \ 1]$
2. Demostreu les lleis de De Morgan en base a tots els possibles valors que poden prendre les variables que hi apareixen.
3. Demostreu la llei 4 de l'àlgebra de Boole a partir de la llei 2 i de l'axioma e).
4. Apliqueu el principi de dualitat a l'expressió $x + (y \cdot 1) + (z \cdot 0 \cdot w)$ i simplifiqueu l'expressió obtinguda aplicant els axiomes i teoremes de l'àlgebra de Boole.
5. Demostreu que les lleis de De Morgan també es compleixen per a 3 i 4 variables. És suficient amb demostrar que $(xyz)' = x' + y' + z'$ i que $(xyzw)' = x' + y' + z' + w'$, perquè les altres lleis s'obtenen directament aplicant el principi de dualitat.
6. Trobeu una expressió algebraica per a les funcions g_1, g_3, g_6, g_7 i g_{10} de la figura 2.
7. Trobeu una expressió més senzilla per aquesta funció:

$$f = wx + xy' + yz + xz' + xy$$
8. Sigui una funció de 3 variables x, y i z . Trobeu-ne una expressió algebraica suposant que la funció ha de valdre 1 quan es compleixi alguna de les següents condicions:
 - $x = 1$ o $y = 0$
 - $x = 0$ i $z = 1$
 - totes tres variables valen 1
9. Donada la funció $xy' + x'y = z$, demostreu que $xz' + x'z = y$.
10. Es disposa de dues caixes fortes electròniques, A i B. Cadascuna de les caixes té un senyal associat, x_A i x_B respectivament, que val 1 quan la caixa és oberta i 0 quan és tancada. Es disposa també d'un interruptor general que té un senyal associat ig , que val 1 si l'interruptor està tancat i 0 si està obert.
Es vol construir un sistema d'alarma contra robatoris, que generarà un senyal de sortida s . Aquest senyal ha de valdre 1 quan alguna caixa forta estigui oberta i l'interruptor no estigui tancat.
Doneu l'expressió algebraica de la funció $s = f(x_A, x_B, ig)$.
11. En Joan s'ha examinat de tres assignatures. Els seus amics han vist els resultat dels exàmens i li han comentat el següent:
 - Has aprovat mates o física, diu el primer
 - Has suspès química o mates, diu el segon
 - Has aprovat només dues assignatures, diu el tercer.
 a) Escriviu les expressions algebraiques de les afirmacions de cadascun dels amics. Tingueu en compte que la "o" del llenguatge natural és una "o exclusiva"
 b) Utilitzant els axiomes i teoremes de l'àlgebra de Boole, deduiu quines assignatures ha aprovat en Joan i quina ha suspès.
12. Escriviu la taula de veritat de les funcions g_1, g_3, g_6, g_7 i g_{10} de la figura 2.
13. Especifiqueu la taula de veritat d'una funció de 4 variables, x_3, x_2, x_1 i x_0 , que val 1 només quan un nombre parell de les variables valen 1.
14. Obteniu la taula de veritat de les funcions següents i determineu si són equivalents:

$$f(x, y, z, w) = xy'zw' + yz'w + x'z + xyw + x'yz'w'$$

$$g(x, y, z, w) = xy + z$$
15. Mitjançant taules de veritat, estudeu si les dues expressions següents són equivalents:

$$f = x'y' + yw + x'w$$

$$g = x'y'z'w' + x'z'w + yz'w + xyw + x'z$$
16. Escriviu la taula de veritat corresponent al problema 10.
17. Escriviu la taula de veritat corresponent al problema 11. Trobeu la solució del problema a partir d'estudiar el contingut d'aquesta taula.

18. Obteniu l'expressió en suma de mintermes de les funcions f_0 , f_1 , f_2 i f_3 :

x_2	x_1	x_0	f_0	f_1	f_2	f_3
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	1	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	1
1	0	1	0	1	1	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

19. Obteniu l'expressió en forma de suma de mintermes de les funcions següents:

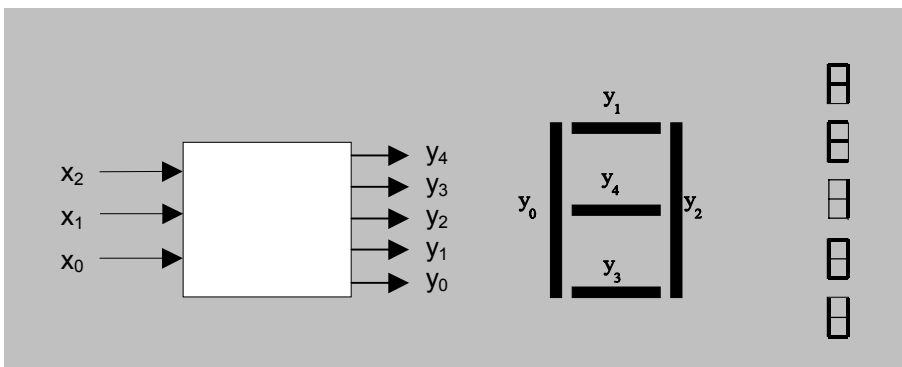
a) $f(x, y, z) = xy + z$

b) $f(x, y, z, w) = x'y'zw + x'y'z' + xy'w + yz'w$

20. Es vol construir un sistema que computi 5 funcions de sortida (y_1, y_2, y_3, y_4 i y_5) de 3 variables (x_2, x_1 i x_0). Aquestes 3 variables codifiquen les cinc vocals, tal com es mostra a la taula següent.

x_2	x_1	x_0	vocal
0	0	0	A
0	0	1	E
0	1	0	I
0	1	1	O
1	0	0	U

Cadascuna de les funcions està associada a un segment d'un *visualitzador de 5 segments* com el que es mostra a la figura. Per exemple, quan y_1 val 1, el segment de dalt del visualitzador s'il·lumina. La figura mostra també quins segments s'han d'il·luminar per a formar les diferents vocals.



En cada moment, el visualitzador ha de formar la vocal que codifiquen les variables d'entrada en aquest moment. Escriviu la taula de veritat de les 5 funcions.

21. Es vol dissenyar un sistema de rec d'una planta amb control de la temperatura i de la humitat de la terra. El sistema té 3 senyals d'entrada (variables) i 2 de sortida (funcions).

Entrades:

- Un sensor de temperatura (t): es posa a '1' si la temperatura de la terra supera un límit prefixat T_0 .
- Dos sensors d'humitat de la terra (h_0 i h_1): es posen a '1' quan la humitat de la terra supera els límits H_0 i H_1 , respectivament. El límit H_0 és inferior al límit H_1 ($H_0 < H_1$).

Sortides:

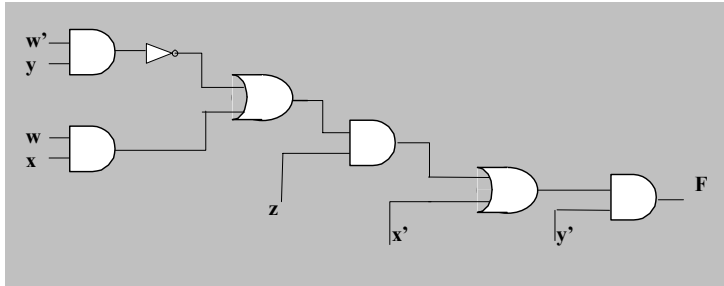
- *Regar* (R): quan es posa a '1' s'activa el rec de la planta.
- *Escalfar* (E): quan es posa a '1' s'activa l'escalfament de la terra.

Les especificacions del sistema són:

- La planta es rega sempre que la terra està seca, és a dir, sempre que no es supera el límit H_0 .
- També es rega quan la temperatura supera el límit T_0 i la humitat de la terra és inferior a H_1 .
- La terra de la planta s'escalfa quan la temperatura és inferior a T_0 i la humitat és superior a H_0 .

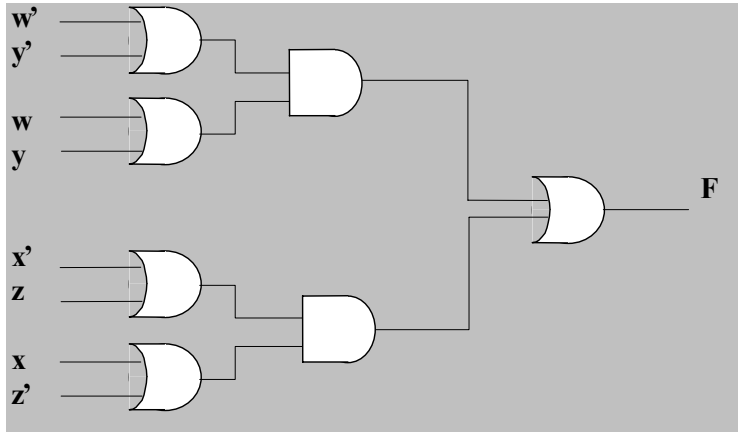
Escriviu la taula de veritat de les funcions R i E.

22. Analitzeu els següent circuit combinacional i expresseu algebraicament en forma de suma de productes la funció que implementa:

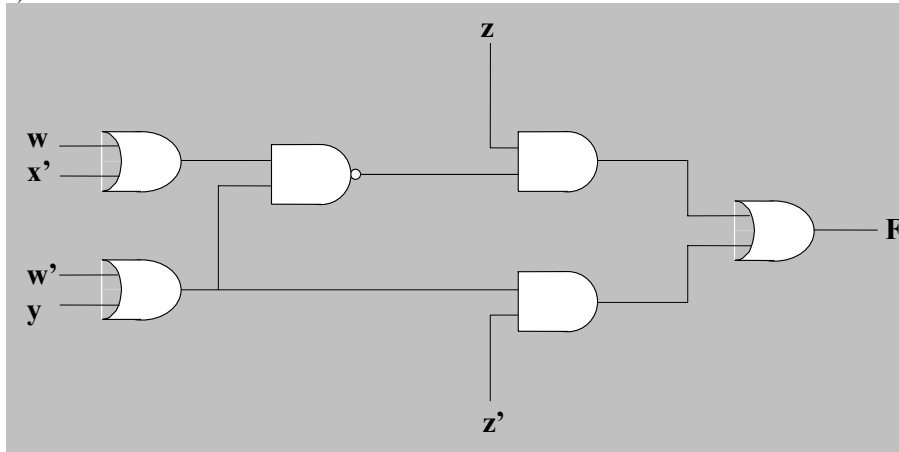


23. Analitzeu els següents circuits:

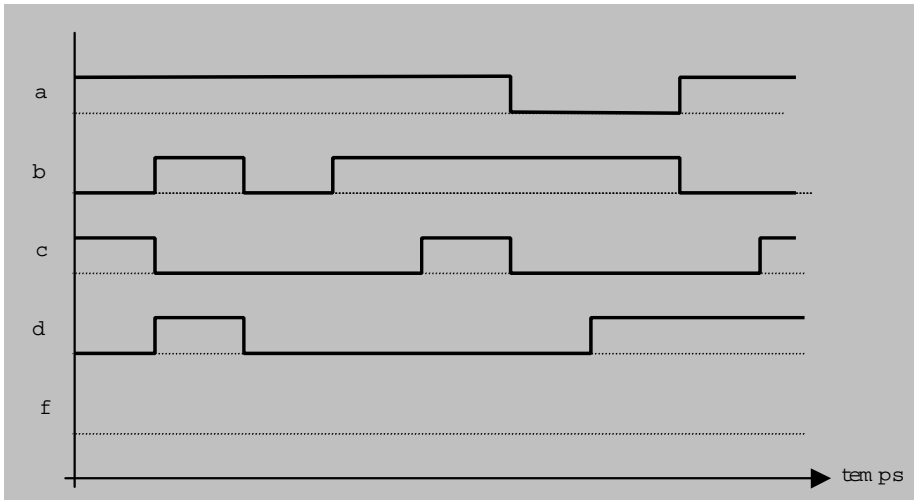
a)



b)



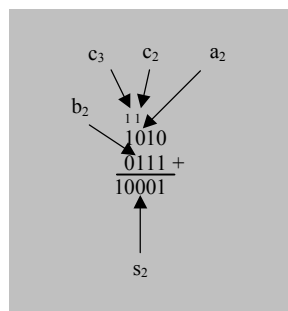
24. Sintetitzeu la funció $f(a,b,c,d) = d \cdot (a' + a \cdot b) \cdot (b' \oplus c)$
25. Completeu el cronograma següent, suposant que el senyal f correspon a la funció de la figura 7b i que les entrades prenen els valors dibuixats. Considereu que els retards introduïts per les portes són 0.



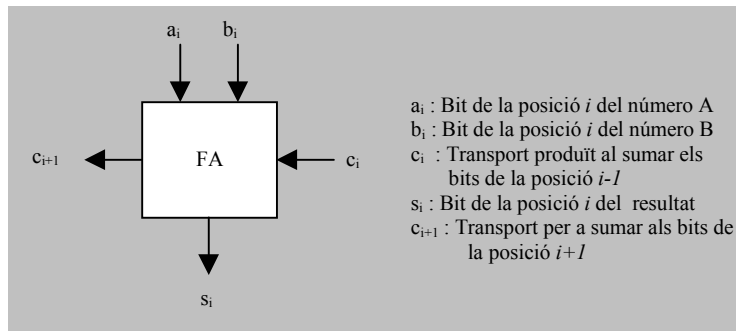
26. Feu la síntesi a 2 nivells de les següents funcions:
- $f(x,y,z,w) = x'y'z'w' + x'yz'w' + xy'zw' + xyzw + x'y'z'w + x'yzw$
 - $f(x,y,z) = xz' + y'z + x'y$
27. Es desitja dissenyar un circuit combinacional que permeti multiplicar dos números naturals de dos bits.
- Indiqueu el nombre de bits de la sortida
 - Escriviu la taula de veritat de les funcions de sortida
 - Implementeu el circuit a 2 nivells.
28. Sintetitzeu de forma mínima a 2 nivells el circuit descrit al problema 27.
29. Dissenyeu a dos nivells un circuit combinacional que permeti multiplicar dos números enters de dos bits representats en complement a 2.
30. Quan fem una suma de números binaris la fem bit a bit (tal com en decimal la fem dígit a dígit). Per obtenir el bit de la posició i necessitem conèixer els bits que són a la posició i en els dos operands i el transport que es genera en la suma dels bits de la posició $i-1$.

La suma d'aquest tres bits dóna com a resultat dos bits: el bit s_i , que correspon al bit de la posició i de la suma, i el bit c_{i+1} , que és el transport que es genera per a la suma dels bits de la posició $i+1$.

A continuació es mostra un exemple per $A=1010$ i $B=0111$. L'exemple mostra concretament la suma dels bits de la posició 2 (recordeu que el bit de més a la dreta és el bit de la posició 0).



El sumador de números d'un bit té tres entrades (a_i , b_i i c_i) i dues sortides (s_i i c_{i+1}). Aquest sumador rep el nom de *full adder* (sumador complet), i s'abreua "FA". La figura següent mostra la seva estructura.



- a) Escriviu la taula de veritat i feu la implementació interna (mitjançant portes lògiques) d'un FA com el descrit.
- b) Utilitzeu 4 FA's per tal de construir un sumador de dos números de 4 bits. Què s'ha de connectar a l'entrada que correspon al bit c_0 ?
31. Sintetitzeu a 2 nivells la funció següent:

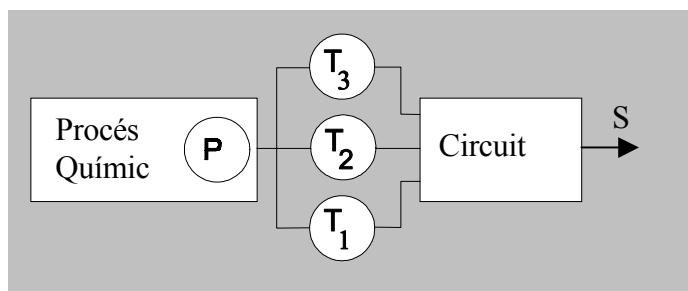
x_2	x_1	x_0	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

32. Sintetitzeu de forma mínima a 2 nivells la funció descrita al problema 10.
33. Sintetitzeu de forma mínima a 2 nivells la funció descrita al problema 26a. Compareu la resposta amb l'obtinguda al problema 26a.
34. Sintetitzeu de forma mínima la funció:
 $f(a,b,c,d) = \sum_4 m(0,4,5,10,13) + \sum_4 d(2,6,7,8,9,12)$
35. Sintetitzeu de forma mínima a 2 nivells la funció descrita al problema 21
36. Un procés químic té tres indicadors de temperatura sobre un punt P determinat, les sortides dels quals T_1 , T_2 , i T_3 , adopten dos nivells de tensió segons la temperatura sigui menor, major o igual que unes determinades constants t_1 , t_2 i t_3 respectivament ($t_1 < t_2 < t_3$).

$$T_1 = 1 \text{ si } t > t_1 \quad T_1 = 0 \text{ si } t \leq t_1$$

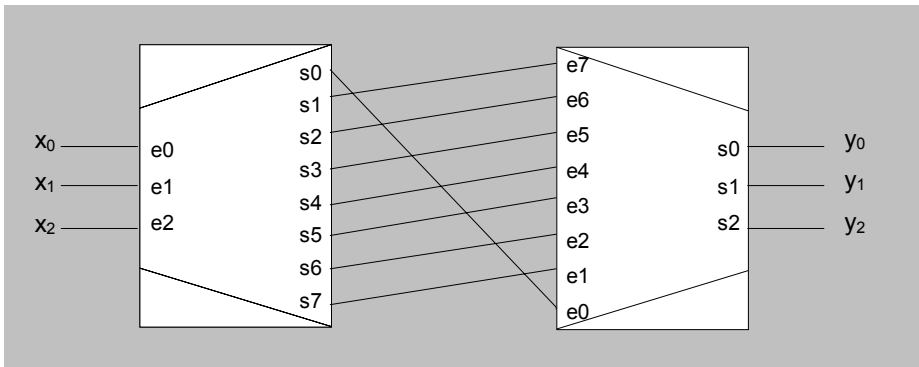
Les sortides T_2 i T_3 tenen el mateix comportament.

Sintetitzeu un circuit combinacional que generi una sortida S que valgui 1 si quan la temperatura a P estigui entre t_1 i t_2 o bé sigui superior a t_3 . Realitzeu la implementació amb portes de forma mínima.

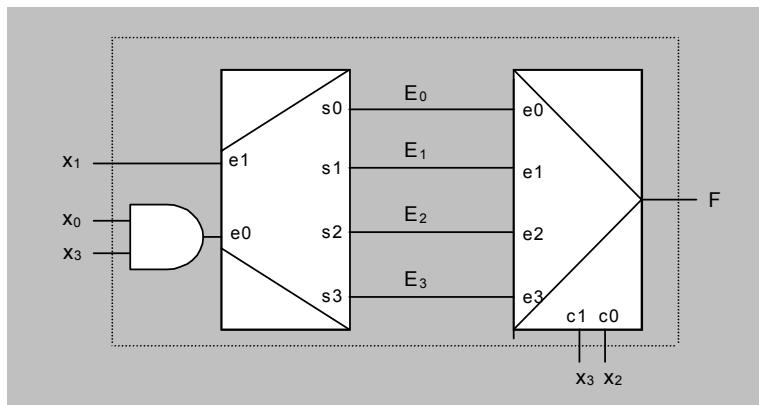


37. Obteniu l'expressió algebraica de la sortida s d'un multiplexor de 4 entrades de dades. Feu la seva implementació interna mitjançant portes lògiques.

38. Implementeu la funció $f(a,b,c) = abc' + a'c$ amb un multiplexor de 3 entrades de control.
39. Implementeu un circuit combinacional que permeti multiplicar dos números enters de dos bits representats en complement a 2 utilitzant només multiplexors.
40. Dissenyeu un multiplexor de 16 entrades de dades, utilitzant 2 multiplexors de 8 entrades de dades i les portes lògiques que facin falta.
41. Dissenyeu un multiplexor de 16 entrades de dades utilitzant únicament multiplexors de 4 entrades de dades (no useu cap porta lògica addicional).
42. Deduïu la implementació interna (mitjançant portes lògiques) d'un codificador 4-2.
43. Deduïu la implementació interna (mitjançant portes lògiques) d'un descodificador 2-4.
44. Implementeu la funció $f(a,b,c) = abc' + a'c$ amb un descodificador 3-8
45. Dissenyeu un descodificador 4-16 utilitzant únicament cinc descodificadors 2-4.
46. Què fa el següent circuit suposant que $x=(x_2,x_1,x_0)$ i $y=(y_2,y_1,y_0)$ són números enters codificats en complement a 2?

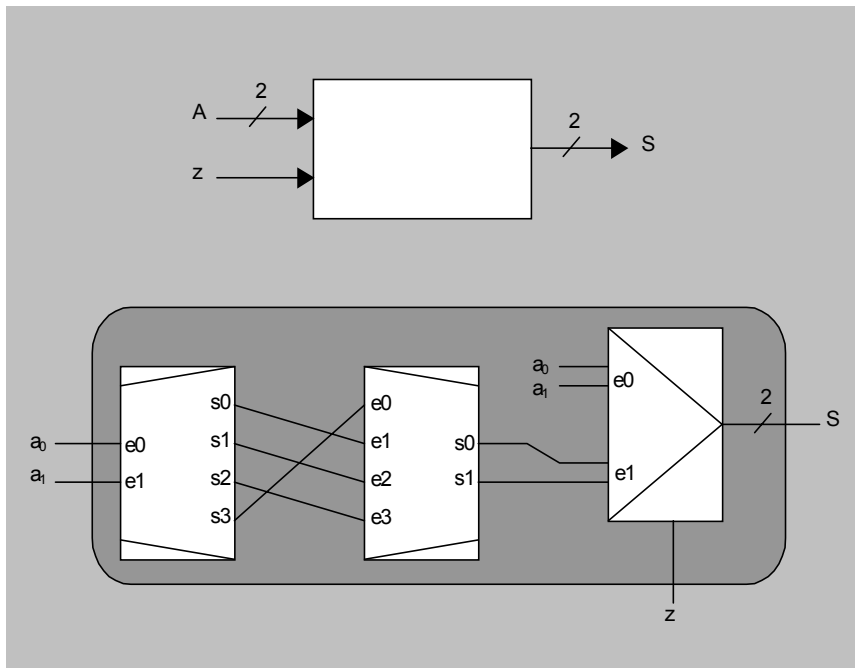


47. Donat el següent circuit:



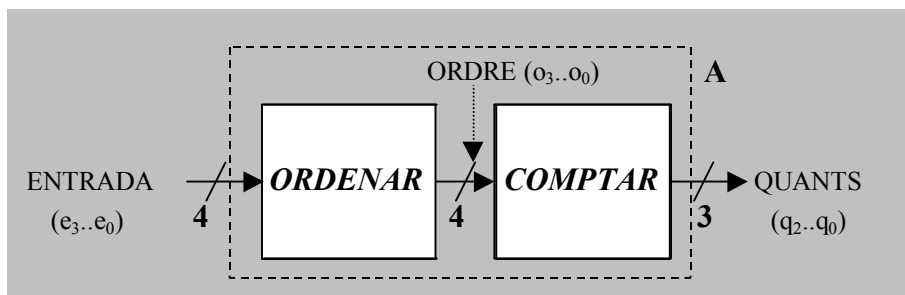
- a) Trobeu les expressions algebraiques d' E_i ($i=0,1,2,3$) en funció d' x_3, x_1 i x_0 .
- b) Escriviu la taula de veritat d' $F(x_3, x_2, x_1, x_0)$.

48. Dissenyeu un descodificador 3-8 utilitzant dos descodificadors 2-4 i les portes lògiques que facin falta.
49. Quina funció realitza el següent circuit si interpretem les entrades i la sortida com a números codificats en binari natural?



50. Suposant que X és un número natural codificat amb 3 bits, implementeu la funció $(3 \cdot X) \bmod 8$ usant només un descodificador 3-8 i un codificador 8-3.
51. Contesteu els següents apartats:
- Implementeu un circuit que pugui actuar com un decalador d'1 bit a l'esquerra de senyals de 4 bits usant un multiplexor de busos de 4 bits de dues entrades de dades. Un senyal de control d'un bit, d , determina si el número d'entrada s'ha de decalar o no.
 - ¿A quina operació aritmètica equival aquest desplaçament?
 - Indica quan es produiria sobreeximent en els casos d'interpretar el senyal d'entrada com un número natural codificat en binari o bé com un número enter codificat en complement a 2.
52. Contesteu els apartats següents:
- Implementeu un *decalador programable a l'esquerra* de números de 8 bits. El decalador té una entrada de control de 3 bits, C , que indica el nombre de bits del decalatge.
 - Si C codifica en binari el valor n , a quina operació aritmètica equival aquest desplaçament?
 - Suposant que $C=010$, indiqueu quan es produiria sobreeximent en els casos d'interpretar l'entrada com un número natural o bé com un enter representat en complement a 2.
- Suposem ara que volem realitzar l'operació $X / 2^n$, usant un decalador a la dreta.
- Quin tipus de decalador hem de fer servir si X és un número natural representat en binari? I si és un número enter representat en complement a 2?
 - En quins casos el resultat de la divisió no és exacte?
53. Un codificador amb prioritat cíclica és un codificador amb n entrades de control addicionals que indiquen quina de les 2^n entrades de dades té la màxima prioritat. A partir d'aquesta entrada, la prioritat és cíclica; És a dir, si $n=3$ i les entrades de control codifiquen un 4, l'ordre en que s'exploraran les entrades de dades per a codificar la sortida és: 4, 3, 2, 1, 0, 7, 6, 5.
- Implementeu un codificador amb prioritat cíclica per a $n=1$
 - Idem per a $n=2$
 - Idem per a $n=3$
54. Indiqueu la grandària i el contingut d'una memòria ROM que implementi la funció descrita al problema 27.

55. Indiqueu la grandària i el contingut d'una memòria ROM que implementi la funció descrita al problema 21.
56. El circuit combinacional de la figura compta el nombre d'1s que té un mot d'entrada de 4 bits.



El bloc *ORDENAR* ordena el mot ENTRADA, col·locant tots els 1s a la dreta i tots els 0s a l'esquerra. Per exemple:

si ENTRADA = 0101 llavors ORDRE = 0011
 si ENTRADA = 0000 llavors ORDRE = 0000

El bloc *COMPTAR* genera a la sortida QUANTS la codificació binària de la quantitat d'1s del mot ORDRE. Per exemple:

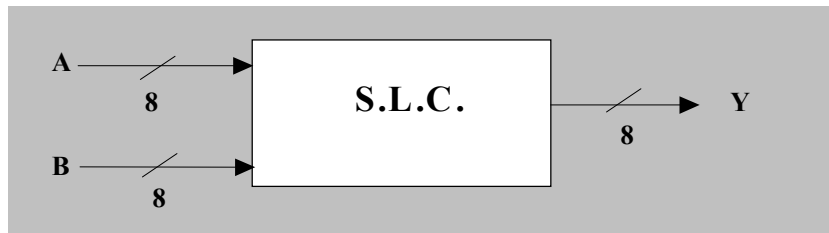
si ORDRE = 0011 llavors QUANTS = 010 (2)
 si ORDRE = 0000 llavors QUANTS = 000 (0)

Respon els següents apartats:

- Feu la taula de veritat del bloc *ORDENAR*.
 - Dissenyu el bloc *COMPTAR* utilitzant només blocs combinacionals (exceptuant memòria ROM), de la grandària que faci falta.
 - Dissenyu el circuit combinacional complet A amb una memòria ROM, indicant la seva grandària i contingut.
57. Deduiu les expressions algebraiques de les funcions de sortida $A = B$ i $A < B$ d'un comparador de números de 2 bits i feu la seva implementació mitjançant portes lògiques.
58. Usant sumadors i restadors de 4 bits i les portes lògiques que calguin, dissenyeu un comparador de números naturals de 4 bits.
59. Sintetitzeu un comparador de números enters de 4 bits codificats en complement a 2, utilitzant comparadors de números naturals de 4 bits i les portes lògiques necessàries.
60. Dissenyu un circuit que obtingui a la seva sortida el màxim de dos números naturals, X i Y, de 4 bits.
61. Suposant que X és un número natural codificat amb 3 bits, implementeu la funció $Z = (3 \cdot X) \bmod 8$ usant només un sumador de 4 bits i un decalador.
62. Dissenyu un circuit que realitzi l'operació $Z = X - Y$ essent X, Y i Z números codificats en complement a 2 amb n bits.
63. Dissenyu un circuit que sigui capaç de sumar o restar dos números codificats en complement a 2 amb 4 bits en funció del què valgui el senyal d'entrada s'/r: si val 0 el circuit ha de realitzar la suma, i si val 1 ha de realitzar la resta. El circuit ha de generar a més un senyal de sortida V que indiqui si s'ha produït sobreiximent en fer l'operació.
64. Dissenyu, a nivell de blocs, un sistema combinacional que faci la funció:

$$Y = INT(0.75 * MAX(A, B))$$

on la funció *INT* calcula la part entera per defecte i la funció *MAX* calcula el màxim dels dos números naturals.

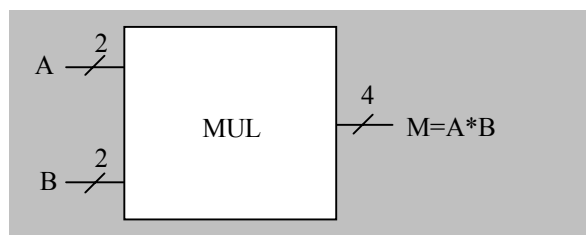


Es demana que ho feu de 3 maneres distintes, considerant:

- $0.75 = 1/2 + 1/4$
- $0.75 = (2+1)/3$
- $0.75 = 1 - 1/4$

Compareu els tres dissenys

65. Es disposa d'un circuit combinacional que implementa un multiplicador de dos números enters, representats en complement a 2 amb 2 bits. La sortida d'un circuit multiplicador de números enters sempre té el doble de bits que les entrades.



- Escriviu la taula de veritat del bloc MUL.
- Implementeu amb blocs combinacionals un circuit que calculi l'operació:

$$Z = A^4 + A^3 + A^2$$

on A és un nombre enter de 2 bits representat en complement a 2. Podeu usar blocs de qualsevol nombre de bits (especifiqueu de quants i justifiqueu la resposta), inclòs el de l'apartat a), que no cal dissenyar. Una possible solució només requereix un sumador i dos multiplicadors.

66. Contesteu els següents apartats:
- Dissenyau una UAL que, a partir de 2 entrades de control c_1 i c_0 , realitzi les següents operacions sobre dos números A i B de 4 bits:

- $c_1=0, c_0=0$: $R=A+B$
- $c_1=0, c_0=1$: $R=A-B$
- $c_1=1, c_0=0$: $R=A$
- $c_1=1, c_0=1$: $R=-A$

b) Afegiu a la UAL dissenyada en l'apartat anterior els circuits necessaris per a calcular els següents bits de condició:

- Vb: Sobreeiximent si s'interpreta que els números d'entrada estan codificats en binari
- V: Sobreeiximent si s'interpreta que els números d'entrada estan codificats en complement a 2
- N: $N=1$ si el resultat de l'operació interpretat en complement a 2 és negatiu, i 0 en cas contrari.
- Z: $Z=1$ si el valor de la sortida és 0, i $Z=0$ en cas contrari

67. Dissenyau una UAL amb sortida R de 4 bits que, a partir de 3 entrades de control c_2 , c_1 i c_0 , realitzi les següents operacions sobre dos números A i B de 4 bits:

- $c_2=0, c_1=x, c_0=x$: $R=B$
- $c_2=1, c_1=0, c_0=0$: $R=A+B$
- $c_2=1, c_1=0, c_0=1$: $R=A-B$
- $c_2=1, c_1=1, c_0=0$: $R=B \gg 1$
- $c_2=1, c_1=1, c_0=1$: $R=A \text{ AND } B$

Solucionari

1.

Per avaluar l'expressió $x + y \cdot (z + x')$ substituïrem les variables per cada combinació de valors:

Pel cas $[x \ y \ z] = [0 \ 1 \ 0]$, tenim:

$$0 + 1 \cdot (0 + 1) = 0 + 1 \cdot 1 = 0 + 1 = 1$$

Pel cas $[x \ y \ z] = [1 \ 1 \ 0]$, tenim:

$$1 + 1 \cdot (0 + 0) = 1 + 1 \cdot 0 = 1 + 0 = 1$$

Pel cas $[x \ y \ z] = [0 \ 1 \ 1]$, tenim:

$$0 + 1 \cdot (1 + 1) = 0 + 1 \cdot 1 = 0 + 1 = 1$$

2.

Per fer la demostració, substituïrem les variables d'entrada per totes les combinacions de valors que puguin prendre. Les igualtats s'han de complir per tots els casos.

▪ Igualtat $(x + y)' = x' \cdot y'$

- Cas $x=0$ i $y=0$

$$(x + y)' = (0 + 0)' = 0' = 1$$

$$x' \cdot y' = 0' \cdot 0' = 1 \cdot 1 = 1$$

- Cas $x=0$ i $y=1$

$$(x + y)' = (0 + 1)' = 1' = 0$$

$$x' \cdot y' = 0' \cdot 1' = 1 \cdot 0 = 0$$

- Cas $x=1$ i $y=0$

$$(x + y)' = (1 + 0)' = 1' = 0$$

$$x' \cdot y' = 1' \cdot 0' = 0 \cdot 1 = 0$$

- Cas $x=1$ i $y=1$

$$(x + y)' = (1 + 1)' = 1' = 0$$

$$x' \cdot y' = 1' \cdot 1' = 0 \cdot 0 = 0$$

Com que les dues expressions valen el mateix per tots els casos possibles, concloem que són equivalents.

Igualtat $(x \cdot y)' = x' + y'$

- Cas $x=0$ i $y=0$

$$(x \cdot y)' = (0 \cdot 0)' = 0' = 1$$

$$x' + y' = 0' + 0' = 1 + 1 = 1$$

- Cas $x=0$ i $y=1$

$$(x \cdot y)' = (0 \cdot 1)' = 0' = 1$$

$$x' + y' = 0' + 1' = 1 + 0 = 1$$

- Cas $x=1$ i $y=0$

$$(x \cdot y)' = (1 \cdot 0)' = 0' = 1$$

$$x' + y' = 1' + 0' = 0 + 1 = 1$$

- Cas $x=1$ i $y=1$

$$(x \cdot y)' = (1 \cdot 1)' = 1' = 0$$

$$x' + y' = 1' + 1' = 0 + 0 = 0$$

Com que les dues expressions valen el mateix per tots els casos possibles, concloem que són equivalents.

3.

La Llei 4 de l'àlgebra de Boole diu que $x + 1 = 1$ i que $x \cdot 0 = 0$

Demostrarem només la primera igualtat; la segona quedarà demostrada pel principi de dualitat.

L'axioma e) diu que $x + x' = 1$. Per tant,

$$x + 1 = x + x + x'$$

La llei 2 (d'idempotència) diu que $x + x = x$. Per tant,

$$x + x + x' = x + x' = 1$$

(hem obtingut l'última igualtat aplicant de nou l'axioma e)). Per tant, $x + 1 = 1$, tal com volíem demostrar.

4.

Per aplicar el principi de dualitat hem d'intercanviar els 0s amb els 1s i els productes lògics amb les sumes lògiques. Així, obtenim l'expressió:

$$x \cdot (y + 0) \cdot (z + 1 + w)$$

Aquesta expressió, obtinguda directament aplicant el teorema de dualitat, es pot simplificar:

- Per la propietat de l'element neutre, tenim que $y + 0 = y$
- Pel teorema 4 (lleï de dominància), tenim que $z + 1 + w = 1$

Substituint en l'expressió, obtenim:

$$x \cdot y \cdot 1$$

i aplicant una altra vegada la propietat de l'element neutre, obtenim l'expressió

$$x \cdot y$$

5.

Hem de demostrar que $(xyz)' = x' + y' + z'$. Apliquem en primer lloc la propietat associativa i després el teorema de De Morgan per a 2 variables, dues vegades:

$$(xyz)' = ((xy)z)' = (xy)' + z' = x' + y' + z'$$

Farem el mateix per a quatre variables, havent demostrat ja que es compleix per a 3 variables:

$$(xyzw)' = ((xyz)w)' = (xyz)' + w' = x' + y' + z' + w'$$

6.

- La funció g_1 val 1 només quan x val 1 i y val 1. Per tant, una possible expressió lògica per aquesta funció es $g_1(x, y) = x \cdot y$
- La funció g_3 val 1 quan x val 1 i y val 0, o bé quan x val 1 i y val 1. Una possible expressió per la funció és, per tant, $x \cdot y' + x \cdot y$. També podem veure que la funció val 1 quan la variable x val 1, independentment del valor que prengui la variable y . Per tant, una possible expressió lògica per aquesta funció és $g_3(x, y) = x$.
- La funció g_6 val 1 quan x val 1 i y val 0, o bé quan x val 0 i y val 1. Per tant, una possible expressió lògica per aquesta funció és $g_6(x, y) = xy' + x'y$
- La funció g_7 val 1 quan x val 0 i y val 1, quan x val 1 i y val 0, o bé quan x val 1 i y val 1. Per tant, una possible expressió lògica per aquesta funció es $g_7(x, y) = x'y + xy' + xy$. També podem dir que la funció val 1 sempre que NO es compleixi que $x = 0$ i $y = 0$. Per tant, una altra expressió per a la funció és $g_7(x, y) = (x'y)'$. Aplicant la segona llei de De Morgan sobre aquesta expressió, i després la llei d'involució, obtenim $g_7(x, y) = x'' + y'' = x + y$.
- La funció g_{10} val 1 quan x val 0 i y val 0, o bé quan x val 1 i y val 0. Es a dir, la funció val 1 quan la variable y val 0, independentment del valor d' x . Per tant, la funció val el contrari del que valgui y , i una possible expressió lògica per aquesta funció es $g_{10}(x, y) = y'$

7.

Tenim l'expressió:

$$wx + xy' + yz + xz' + xy$$

Per simplificar-la aplicarem la propietat distributiva. Agrupem el segon terme amb l'últim, i obtenim:

$$wx + x(y' + y) + y \cdot z + x \cdot z'$$

Per l'axioma de complementació sabem que $y + y' = 1$. Substituint en l'expressió i aplicant l'axioma dels elements neutres obtenim:

$$w \cdot x + x \cdot 1 + yz + xz' = wx + x + yz + xz'$$

Per la llei d'absorció, $wx + x = x$. Per tant,

$$wx + x + yz + xz' = x + yz + xz'$$

Apliquem de nou la llei d'absorció a $x + xz'$, obtenint que

$$x + yz + xz' = x + yz$$

Per tant, concloem que:

$$wx + xy' + yz + xz' + xy = x + yz$$

8.

- Per la primera condició hem de trobar una expressió que valgui 1 quan $x=1$ o $y=0$. Això és el mateix que dir $x=1$ o $y'=1$. L'expressió que val 1 quan es compleix aquesta condició és $x + y'$

- La segona condició és $x=0$ i $z=1$. Això és el mateix que $x'=1$ i $z=1$, i l'expressió que val 1 en aquest cas és $x' \cdot z$.

- La tercera condició, $x=1$ i $y=1$ i $z=1$ només s'acompleix per l'expressió $x \cdot y \cdot z$

- Com que la funció ha de valer 1 en qualsevol dels tres casos, és a dir, quan s'acompleixi la condició 1 o la condició 2 o la condició 3, tenim que l'expressió de la funció és:

$$F = x + y' + x'z + xyz$$

9.

Per demostrar $x \cdot z' + x' \cdot z = y$, substituïrem les zetes de l'expressió pel l'equació que sabem que és certa: $z = x' \cdot y + x \cdot y'$

Una vegada desenvolupada i simplificada l'expressió, hem d'obtenir com a resultat "y".

$$\begin{aligned} x \cdot z' + x' \cdot z &= \\ &= x \cdot (x' \cdot y + x \cdot y')' + x' \cdot (x' \cdot y + x \cdot y') = \end{aligned}$$

Aplicant les lleis de De Morgan:

$$= x \cdot (x' \cdot y)' \cdot (x \cdot y')' + x' \cdot (x' \cdot y + x \cdot y') =$$

i un altre vegada les lleis de De Morgan:

$$= x \cdot (x'' + y') \cdot (x' + y'') + x' \cdot (x' \cdot y + x \cdot y') =$$

Pel teorema d'involució:

$$= x \cdot (x + y') \cdot (x' + y) + x' \cdot (x' \cdot y + x \cdot y') =$$

Aplicant la propietat distributiva:

$$= (x \cdot x + x \cdot y') \cdot (x' + y) + (x' \cdot x' \cdot y) + (x' \cdot x \cdot y') =$$

Simplificant pel teorema d'idempotència:

$$= (x + x \cdot y') \cdot (x' + y) + (x' \cdot y) + (x' \cdot x \cdot y') =$$

i per l'axioma de complementació

$$= (x + x \cdot y') \cdot (x' + y) + (x' \cdot y) =$$

Pel teorema d'absorció:

$$= x \cdot (x' + y) + (x' \cdot y) =$$

Aplicant un altre vegada la propietat distributiva:

$$= x \cdot x' + x \cdot y + x' \cdot y =$$

i simplificant per l'axioma de complementació:

$$= x \cdot y + x' \cdot y =$$

Aplicant la propietat distributiva:

$$= (x + x') \cdot y =$$

i finalment, per l'axioma de complementació:

$$= 1 \cdot y = y$$



10.

La funció ha de valer 1 quan s'hagi d'activar l'alarma. Segons ens diu l'enunciat, l'alarma s'activa quan l'interruptor general no està tancat ($ig=0$) i alguna de les dues caixes fortes és oberta (és a dir, $x_A=1$ o $x_B=1$).

Una expressió per la funció S que val 1 quan $ig=0$ i ($x_A=1$ o $x_B=1$) és:

$$s = ig'(x_A + x_B)$$

11.

Per a saber quines assignatures ha aprovat en J  quina ha suspès, plantejarem les afirmacions dels seus tres amics de forma algebraica 

El resultat de cada assignatura és pot representar amb una variable booleana. Aquesta variable valdrà 1 si l'assignatura està aprovada i 0 en cas contrari.

Les variables que utilitzarem per cadascuna de les assignatures seran m per "Mates", f per Física i q per Química.

La "o" del llenguatge natural és, normalment, una "o exclusiva". És a dir, si diem "Mates o Física", exclouem les dues a la vegada. una altra possible forma d'expressar la frase "has aprovat Mates o Física" és la següent: "o bé has aprovat Mates i has suspès Física, o bé has suspès Mates i has aprovat Física". D'aquesta frase és pot derivar fàcilment l'expressió $m \cdot f' + m' \cdot f$, que només val 1 quan es compleixen les condicions de la frase. Per tant, podem escriure les següents expressions per cadascuna de les frases:

-Has aprovat Mates o Física: $mf' + m'f$

-Has suspès Química o Mates: $qm' + q'm$

-Has aprovat 2 assignatures: $m'fq + mf'q + m'fq'$

Com que s'han de complir les tres afirmacions a la vegada, tenim que:

$$(m'f + m'f)(qm' + q'm)(m'fq + m'f'q + m'f'q') = 1$$

Simplificarem l'expressió aplicant els axiomes i teoremes de l'àlgebra de Boole fins a obtenir la resposta.

$$(m'f + m'f)(qm' + q'm)(m'fq + m'f'q + m'f'q') =$$

(propietat distributiva sobre els 2 primers parèntesis)

$$(m'f'qm' + m'f'q'm + m'f'qm' + m'f'q'm)(m'fq + m'f'q + m'f'q') =$$

(complementació i idempotència)

$$(0 + m'f'q' + m'f'q + 0)(m'fq + m'f'q + m'f'q') =$$

(elements neutres i distributiva sobre els 2 parèntesis)

$$m'f'q'm'fq + m'f'q'm'f'q + m'f'q'm'f'q' + m'f'qm'fq + m'f'qm'f'q + m'f'qm'f'q' =$$

(complementació i idempotència)

$$0 + 0 + 0 + m'fq + 0 + 0 = m'fq$$

Hem obtingut, per tant, l'expressió:

$$m'fq = 1$$

Aquesta expressió només val 1 quan m val 0 i f i q valen 1, és a dir, que en Joan ha suspès Mates i ha aprovat Física i Química.

12.

Les taules de veritat d'aquestes funcions són les següents:

x	y	g ₁	x	y	g ₃	x	y	g ₆	x	y	g ₇	x	y	g ₁₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	1	1	0	1	1	0	1	0
1	0	0	1	0	1	1	0	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1	1	1	1	0

13.

Per resoldre aquesta activitat considerarem que el 0 és un número parell. Per tant, quan cap de les entrades tingui un 1 la sortida valdrà 1.

La taula de veritat és la següent:

x ₃	x ₂	x ₁	x ₀	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

14.

Per a obtenir aquestes taules, el que farem és escriure una columna per cadascun dels termes producte, i després obtindrem la columna corresponent a f fent la OR de les columnes parcials.

x	y	z	w	$x \cdot y' \cdot z' \cdot w'$	$y \cdot z' \cdot w$	$x' \cdot z$	$x \cdot y \cdot w$	$x' \cdot y \cdot z' \cdot w'$	f
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	1
0	0	1	1	0	0	1	0	0	1
0	1	0	0	0	0	0	0	1	1
0	1	0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0	0	1
0	1	1	1	0	0	1	0	0	1
1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	0	1
1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	1

x	y	z	w	$x \cdot y$	$g = x \cdot y + z$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Deduïm que les funcions no són equivalents per què les seves taules de veritat són diferents.

15.

Escriurem les taules de veritat de les dues funcions:

x	y	z	w	$x' \cdot y'$	$y \cdot w$	$x' \cdot w$	f
0	0	0	0	1	0	0	1
0	0	0	1	1	0	1	1
0	0	1	0	1	0	0	1
0	0	1	1	1	0	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	1	1
0	1	1	0	0	0	0	0
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	1	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	0	1

x	y	z	w	$x' \cdot y' \cdot z' \cdot w'$	$x' \cdot z' \cdot w$	$y \cdot z' \cdot w$	$x \cdot y \cdot w$	$x' \cdot z$	g
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	1	0	0	0	0	0	1	1
0	0	1	1	0	0	0	0	1	1
0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	1	1	0	0	1
0	1	1	0	0	0	0	0	1	1
0	1	1	1	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	1	0	1
1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	1

Deduïm que les funcions no són equivalents per què les seves taules de veritat són diferents.

16.

La taula de veritat és la següent:

ig	X_A	X_B	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

17.

Per resoldre aquest problema, assignarem una funció a cadascuna de les afirmacions dels amics d'en Joan. La funció valdrà 1 per les combinacions que fan certa l'afirmació, i 0 en cas contrari. Les funcions que representen les afirmacions dels amics corresponen a les funcions F_1 , F_2 i F_3 respectivament.

Anomenarem m , f i q a les variables que representen les assignatures. Aquestes variables valdran 1 quan l'assignatura estigui aprovada, i 0 si està suspesa.

La solució, representada per la funció F , l'obtidrem fent una AND de les funcions F_1 , F_2 i F_3 , ja que totes tres frases són certes simultàniament. La combinació per la qual F valgui 1 correspondrà a la solució del problema.

m	f	q	F_1	F_2	F_3	F
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	1	1
1	0	0	1	1	0	0
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	1	0	0	0	0

La funció F indica que en Joan ha aprovat Física i Química i ha suspès les Mates.

18.

$$f_0 = x_2'x_1x_0' + x_2x_1x_0$$

$$f_1 = x_2x_1'x_0' + x_2x_1'x_0 + x_2x_1x_0' + x_2x_1x_0$$

$$f_2 = x_2'x_1'x_0 + x_2'x_1x_0 + x_2x_1'x_0 + x_2x_1x_0$$

$$f_3 = x_2'x_1'x_0 + x_2'x_1x_0' + x_2x_1'x_0' + x_2x_1x_0$$

19.

Escriurem la taula de veritat i a partir d'aquesta obtindrem l'expressió en suma de mintermes.

a)

La taula de veritat de la funció és la següent:

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

L'expressió d'f en suma de mintermes és:

$$f = x'y'z + x'yz + xy'z + xyz' + xyz$$

b)

La taula de veritat de la funció és la següent:

x	y	z	w	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
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	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

L'expressió d'f en suma de mintermes és:

$$f = x'y'z'w' + x'y'z'w + x'y'zw + x'yz'w + xy'z'w + xy'zw + xyz'w$$

20.

Les combinacions d'entrada 101, 110 i 111 no poden produir-se, i per tant no ens importarà el valor que prenguin les sortides en aquests casos.

La taula de veritat de les funcions d'aquest sistema és la següent:

x ₂	x ₁	x ₀	y ₄	y ₃	y ₂	y ₁	y ₀
0	0	0	1	0	1	1	1
0	0	1	1	1	0	1	1
0	1	0	0	0	1	0	0
0	1	1	0	1	1	1	1
1	0	0	0	1	1	0	1
1	0	1	x	x	x	x	x
1	1	0	x	x	x	x	x
1	1	1	x	x	x	x	x

21.

a)

A partir de l'enunciat, obtenim que les expressions de les funcions R i E són les següents:

$$\begin{aligned} R &= h_0' + h_1 t \\ E &= t h_0 \end{aligned}$$

Ara bé, les combinacions amb $h_1=1$ i $h_0=0$ no es poden donar (hem de suposar que els sensors funcionen bé). Per tant, la taula de veritat del sistema és la següent:

t	h_1	h_0	R	E
0	0	0	1	0
0	0	1	0	1
0	1	0	x	x
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	x	x
1	1	1	0	0

22.

$$\begin{aligned} F &= ((w' y)' + wx) z + x' y' = ((w'' + y' + wx) z + x') y' = ((w + y' + wx) z + x') y' \\ &= (wz + y'z + wxz + x') y' = wzy' + y'y'z + wxzy' + x'y' = wzy' + y'z + wxzy' + x'y' \end{aligned}$$

23.

Per fer l'anàlisi, cal obtenir l'expressió de la funció a partir del circuit. A partir de l'expressió obtindrem la taula de veritat. Per a omplir-la còmodament, trobarem prèviament una expressió simplificada de la funció.

a) L'expressió que obtenim directament a partir del circuit és la següent:

$$F = (y' + w')(y + w) + (x' + z)(x + z')$$

Simplificant aquesta expressió, obtenim:

$$F = y'w + yw' + x'z' + xz$$

La taula de veritat és la següent:

x	y	z	w	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

b) L'expressió que obtenim directament a partir del circuit és la següent:

$$F = ((x' + w)(y + w'))'z + (y + w')z'$$

Simplificant aquesta expressió, obtenim:

$$\begin{aligned} F &= ((x' + w)' + (y + w')')z + (y + w')z' = \\ &= (xw' + y'w)z + (y + w')z' = \\ &= xzw' + y'zw + yz' + z'w' \end{aligned}$$

La taula de veritat és la següent:

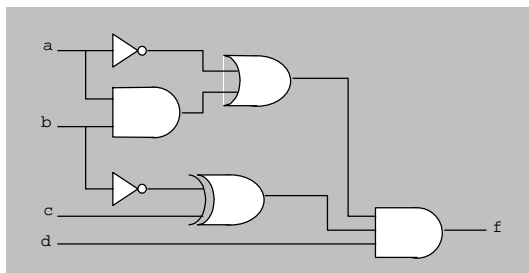
x	y	z	w	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

24.

Per fer la síntesi, simplement dibuixem una línia per cada variable de la funció i substituïm les operacions lògiques de l'expressió per la porta lògica corresponent. La funció és

$$f(a,b,c,d) = d \cdot (a' + a \cdot b) \cdot (b' \oplus c)$$

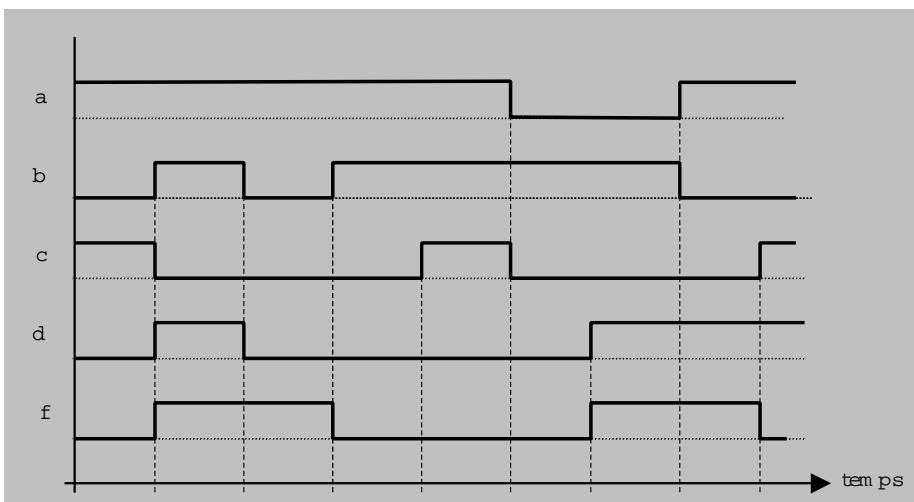
La figura següent mostra el circuit.



25.

A la figura es mostra el cronograma. Com que es tracta d'un circuit combinacional en el que no es consideren els retards (tal com farem habitualment en aquesta assignatura), les variacions en la sortida es produeixen en el mateix moment en què es produeix alguna variació de les entrades (línies puntejades verticals). No obstant, no totes les variacions de les entrades produeixen una variació de la sortida, com es pot veure al cronograma.

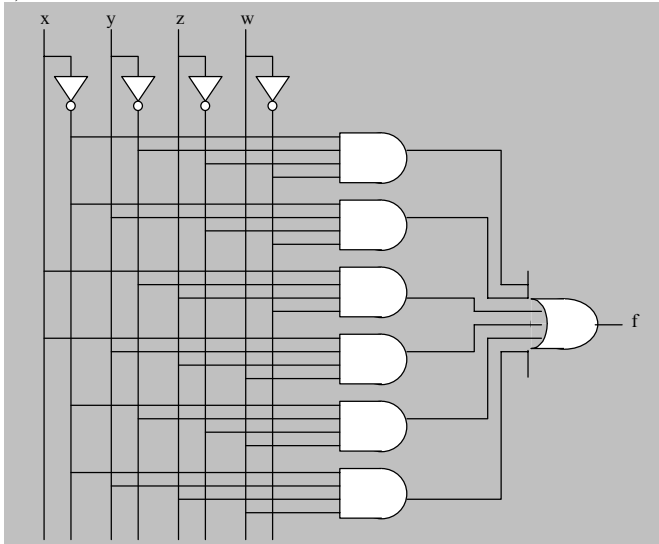
La funció que implementa el circuit és $f = b'c' + c'd$



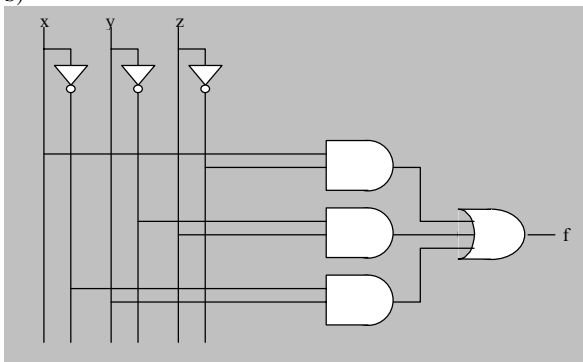
26.

A les figures es mostra la síntesi a 2 nivells de cada circuit. Cada terme producte està implementat per una porta AND de tantes entrades com variables té el terme producte. La suma lògica està implementada per una porta OR de tantes entrades com termes producte té l'expressió.

a)



b)



27.

a) El rang de valors que pot prendre un número de dos bits és 0..3. Per tant, el rang de la multiplicació de dos números serà 0..9. Per representar el 9 necessitem 4 bits. Per tant, la sortida tindrà 4 bits.

b) Anomenarem A i B respectivament als números d'entrada, i C a la seva multiplicació. La taula de veritat d'aquest multiplicador és la següent:

a ₁	a ₀	b ₁	b ₀	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Les expressions en suma de productes de les 4 funcions de sortida són les següents:

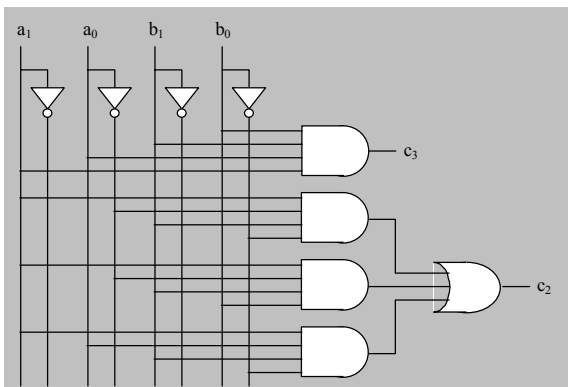
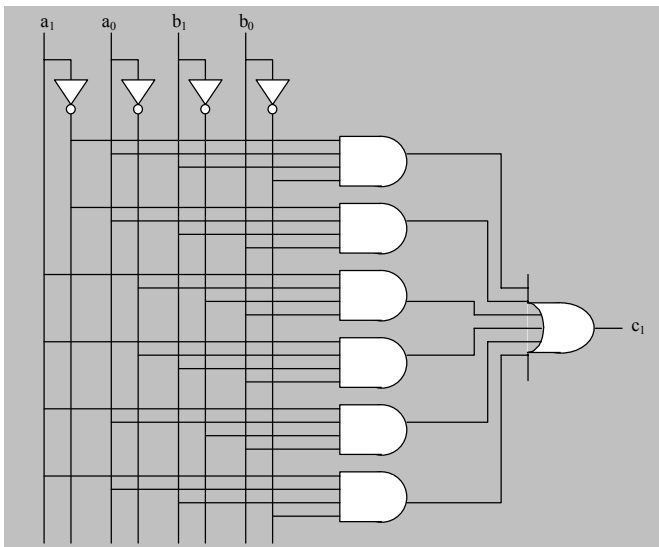
$$c_3 = a_1 a_0 b_1 b_0$$

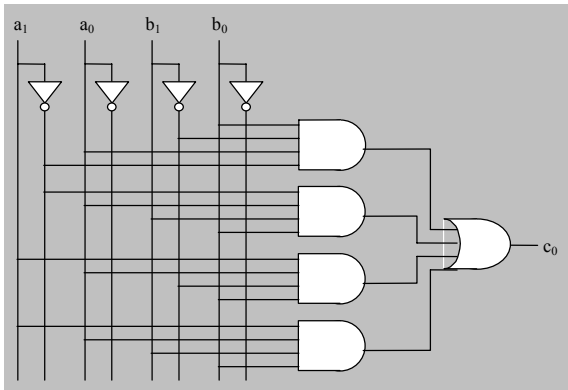
$$c_2 = a_1 a_0' b_1 b_0' + a_1 a_0' b_1 b_0 + a_1 a_0 b_1 b_0'$$

$$c_1 = a_1' a_0 b_1 b_0' + a_1' a_0 b_1 b_0 + a_1 a_0' b_1' b_0 + a_1 a_0' b_1 b_0 + a_1 a_0 b_1' b_0 + a_1 a_0 b_1 b_0'$$

$$c_0 = a_1' a_0 b_1' b_0 + a_1' a_0 b_1 b_0 + a_1 a_0 b_1' b_0 + a_1 a_0 b_1 b_0$$

A continuació es mostren els circuits a 2 nivells corresponents a les quatre funcions de sortida.





28.

Tenim la taula de veritat següent:

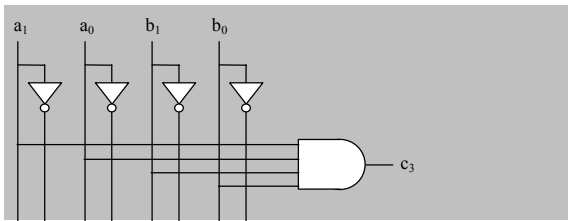
a ₁	a ₀	b ₁	b ₀	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

A continuació simplifiquem per Karnaugh cadascuna de les funcions de sortida, i després implementarem el circuit.

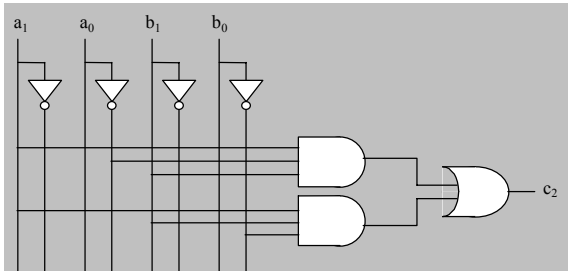
c ₃					c ₂				
a ₁ \ a ₀	00	01	11	10	a ₁ \ a ₀	00	01	11	10
b ₁ \ b ₀	00	01	11	10	b ₁ \ b ₀	00	01	11	10
00	0	0	0	0	00	0	0	0	0
01	0	0	0	0	01	0	0	0	0
11	0	0	1	0	11	0	0	0	1
10	0	0	0	0	10	0	0	1	1

c ₁					c ₀				
a ₁ \ a ₀	00	01	11	10	a ₁ \ a ₀	00	01	11	10
b ₁ \ b ₀	00	01	11	10	b ₁ \ b ₀	00	01	11	10
00	0	0	0	0	00	0	0	0	0
01	0	0	1	1	01	0	1	1	0
11	0	1	0	1	11	0	1	1	0
10	0	1	1	0	10	0	0	0	0

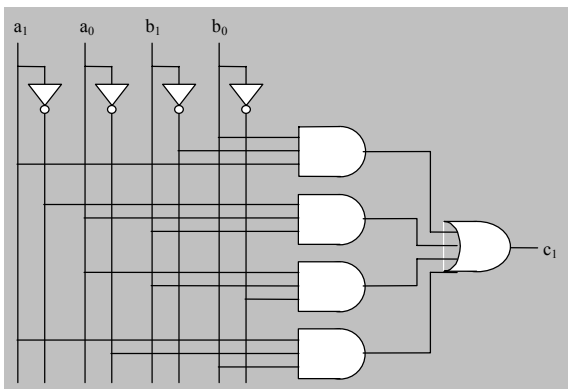
$$c_3 = a_1 a_0 b_1 b_0$$



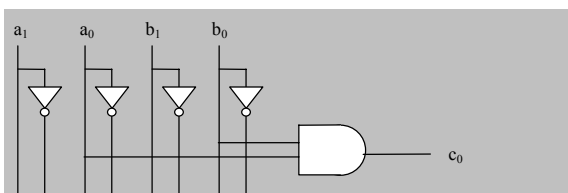
$$c_2 = a_1 a_0' b_1 + a_1 b_1 b_0'$$



$$c_1 = a_1 b_1' b_0 + a_1' a_0 b_1 + a_0 b_1 b_0' + a_1 a_0' b_0$$



$$c_0 = a_0 b_0$$



29.

Primer farem la taula de veritat. Després implementarem el circuit a dos nivells.

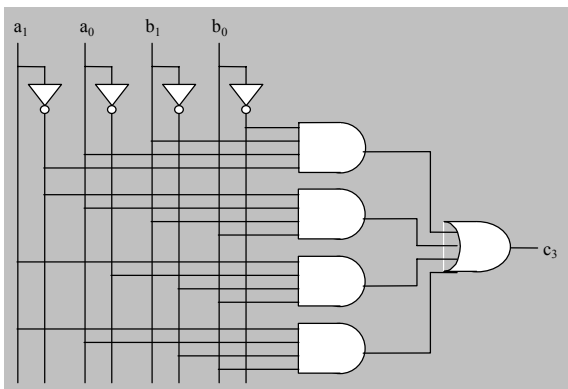
El rang de valors que pot prendre un número enter de dos bits representat en complement a dos és $-2..1$. Per tant, el rang de la multiplicació de dos números serà $-2..4$. Per representar el 4 en complement a dos necessitem 4 bits. Per tant, la sortida tindrà 4 bits.

Anomenarem $[a_1 a_0]$ i $[b_1 b_0]$ respectivament als bits dels dos números d'entrada, i $[c_3 c_2 c_1 c_0]$ als bits de la seva multiplicació. La taula de veritat d'aquest multiplicador és la següent:

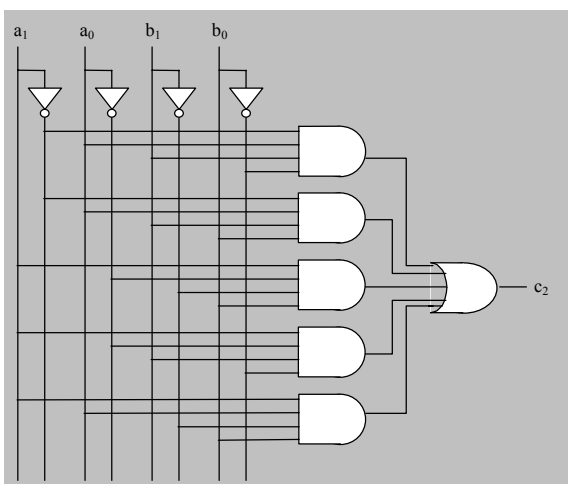
a ₁	a ₀	b ₁	b ₀	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

A continuació es mostren els circuits a 2 nivells corresponents a les quatre funcions de sortida.

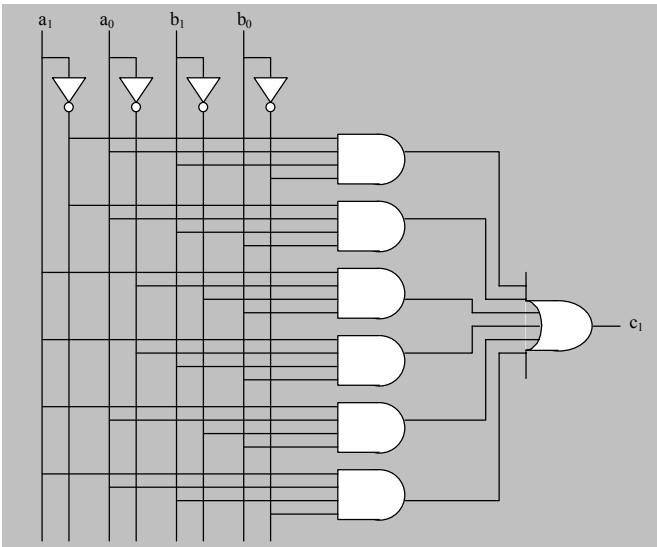
$$c_3 = a_1' a_0 b_1 b_0' + a_1' a_0 b_1 b_0 + a_1 a_0' b_1' b_0 + a_1 a_0 b_1' b_0$$



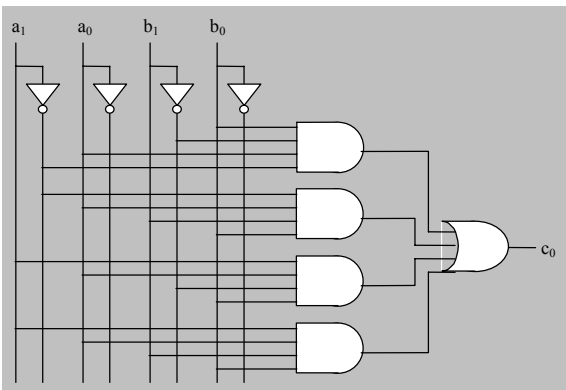
$$c_2 = a_1' a_0 b_1 b_0' + a_1' a_0 b_1 b_0 + a_1 a_0' b_1' b_0 + a_1 a_0' b_1 b_0' + a_1 a_0 b_1' b_0$$



$$c_1 = a_1' a_0 b_1 b_0' + a_1' a_0 b_1 b_0 + a_1 a_0' b_1' b_0 + a_1 a_0' b_1 b_0 + a_1 a_0 b_1' b_0 + a_1 a_0 b_1 b_0'$$



$$c_0 = a_1' a_0 b_1' b_0 + a_1' a_0 b_1 b_0 + a_1 a_0 b_1' b_0 + a_1 a_0 b_1 b_0$$



30.

La taula de veritat d'un *full adder* és la següent:

a_i	b_i	c_i	s_i	c_{i+1}
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

L'expressió en suma de productes de c_{i+1} és la següent:

$$c_{i+1} = a_i' b_i c_i + a_i b_i' c_i + a_i b_i c_i' + a_i b_i c_i$$

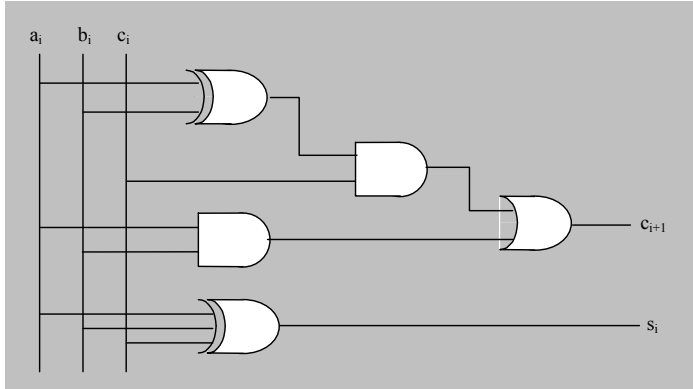
Podem treure factor comú en els dos primers i els dos últims termes, obtenint

$$c_{i+1} = (a_i' b_i + a_i b_i') c_i + a_i b_i (c_i' + c_i) = (a_i \oplus b_i) c_i + a_i b_i$$

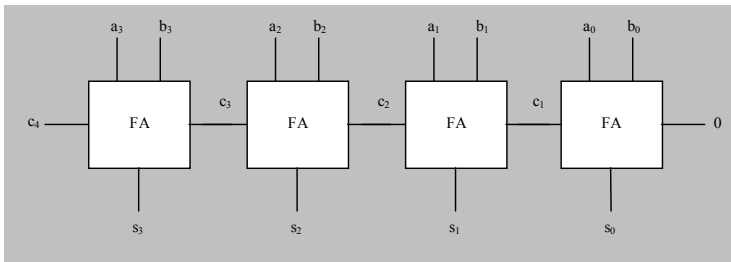
Pel que fa a s_i , veiem que quan $a_i = 0$ val $(b_i \oplus c_i)$ i quan $a_i = 1$ val $(b_i \oplus c_i)'$. Això es pot expressar de la manera següent:

$$s_i = a_i' (b_i \oplus c_i) + a_i (b_i \oplus c_i)' = a_i \oplus b_i \oplus c_i$$

El circuit intern d'un *full adder*, que implementa aquestes expressions, es mostra a la figura següent.



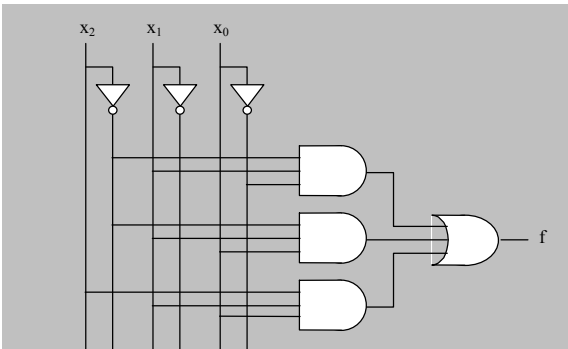
A continuació es mostra com encadenar 4 full adders per tal d'aconseguir un sumador de números de quatre bits. Com es pot veure a la figura, al bit de transport d'entrada, c_0 , s'hi ha de connectar un 0 per tal que la suma sigui correcta.



31.
L'expressió en suma de mintermes de la funció és la següent:

$$f = x_2'x_1x_0' + x_2'x_1x_0 + x_2x_1x_0$$

A continuació es mostra la síntesi a 2 nivells de la funció.



32.
A continuació es mostra la taula de veritat de la funció i la seva minimització utilitzant el mètode de Karnaugh.

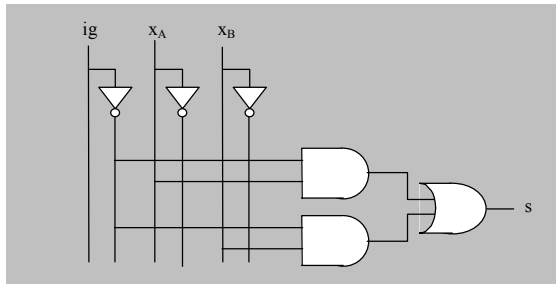
ig	x_A	x_B	s
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

ig x_A	00	01	11	10
x_B 0	0	1	0	0
1	1	1	0	0

Del rectangle vertical se n'obté el terme producte $ig' \cdot x_A$, i del rectangle horitzontal el terme $ig' \cdot x_B$. Per tant, l'expressió de la funció és:

$$s = ig'x_A + ig'x_B$$

A continuació es mostra la síntesi mínima a dos nivells de la funció.



Encara que les portes NOT de les entrades x_A i x_B no s'utilitzen en la funció, les hem inclòs per uniformitat en la metodologia de disseny.

Fixeu-vos que si treiem ig' factor comú en l'expressió mínima d' s , obtenim $s = ig'(x_A + x_B)$, que és l'expressió que vam deduir al problema 10 (però no és una suma de productes).

33.

A continuació es mostren la taula de veritat de la funció i la seva minimització per Karnaugh.

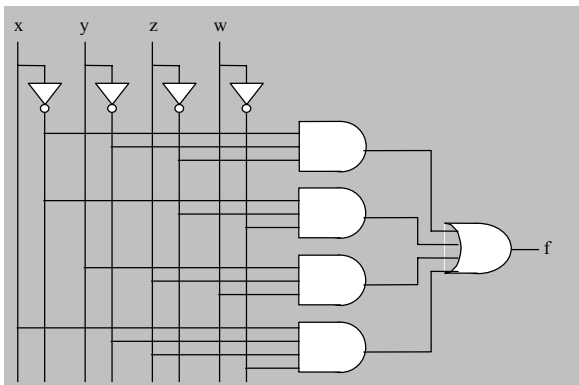
x	y	z	w	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

x y \ z w	00	01	11	10
00	1	1	0	0
01	1	0	0	0
11	0	1	1	0
10	0	0	0	1

Del mapa de Karnaugh se'n dedueix aquesta expressió:

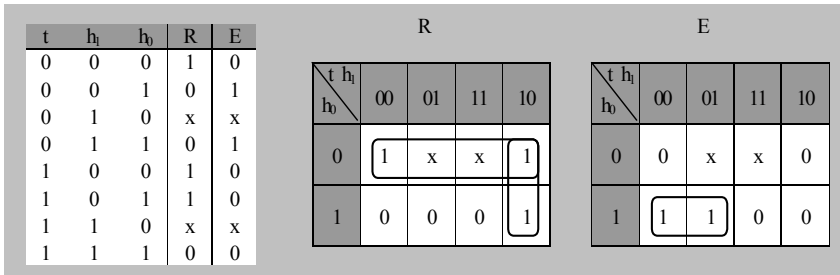
$$f = x'y'z' + x'z'w' + yzw + xy'zw'$$

El circuit mínim a dos nivells es presenta a la següent figura. Es pot comprovar que és més senzill que l'obtingut al problema 26a.



35.

A la següent figura es mostra la taula de veritat de les funcions i la seva minimització per Karnaugh.

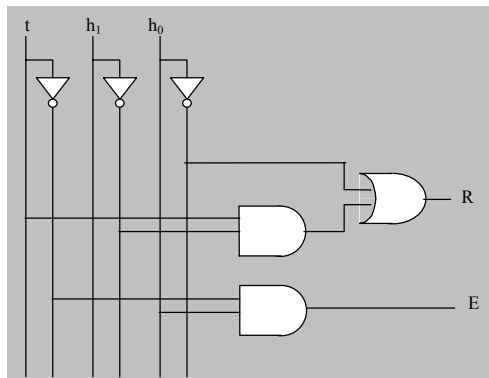


Les funcions obtingudes són:

$$R = h_0' + th_1'$$

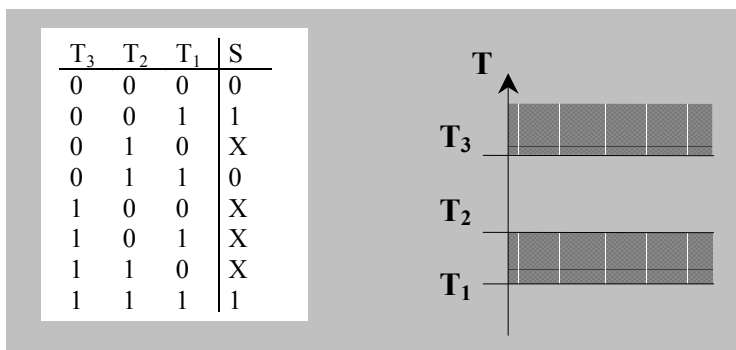
$$E = t'h_0$$

A la següent figura es mostra el circuit minimitzat.

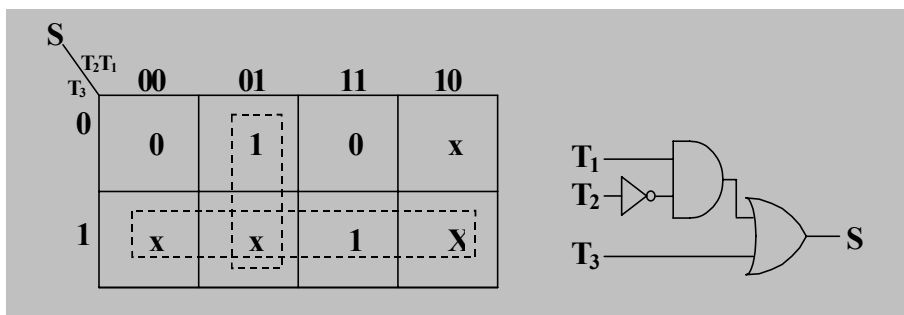


36.

Una X a la taula denota un cas impossible (se suposa que els sensors funcionen correctament).



L'àrea ombrejada indica les zones de temperatura que s'han de detectar.



$$S = T_1 \cdot T_2' + T_3$$

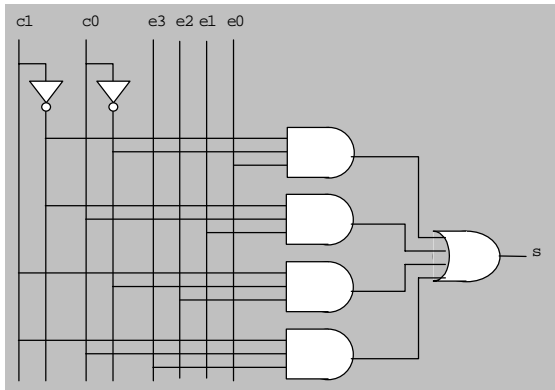
37.

El valor de la sortida s serà el valor de e_3 , e_2 , e_1 o e_0 en funció del què valguin les entrades de control c_1 i c_0 .

Per exemple, si $c_1 = 1$ i $c_0 = 0$ la sortida F valdrà 1 si $e_2 = 1$, i valdrà 0 si $e_2 = 0$ (és a dir, valdrà e_2). En aquest cas podríem escriure que l'expressió de la sortida és $c_1c_0'e_2$. En cada moment, les entrades c_1 i c_0 valen alguna d'aquestes 4 combinacions: [0 0], [0 1], [1 0] o bé [1 1]. Per tant, només un dels productes $c_1c_0'e_0$, $c_1c_0'e_1$, $c_1c_0'e_2$ i $c_1c_0'e_3$ pot valdre 1 en cada moment. Podem concloure que una expressió vàlida per a s és:

$$s = c_1c_0'e_0 + c_1c_0'e_1 + c_1c_0'e_2 + c_1c_0'e_3$$

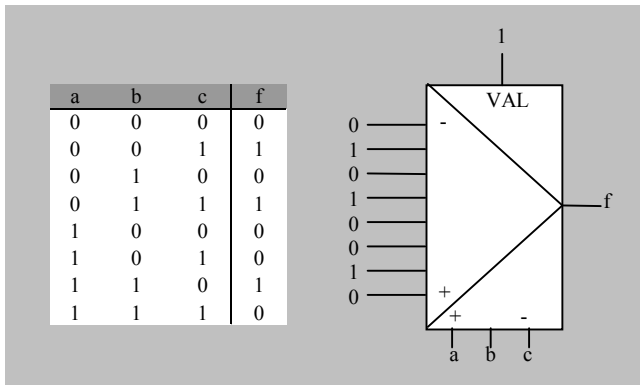
La implementació amb portes d'aquesta expressió és la següent:



38.

Per a implementar aquesta funció construirem la seva taula de veritat. A partir d'aquesta taula sabrem què hem de connectar a les entrades del multiplexor per tal que a la sortida ens aparegui un 0 o un 1 segon indiqui la funció.

En definitiva, es tracta de traspasar la columna f de la taula de veritat a les entrades del multiplexor.



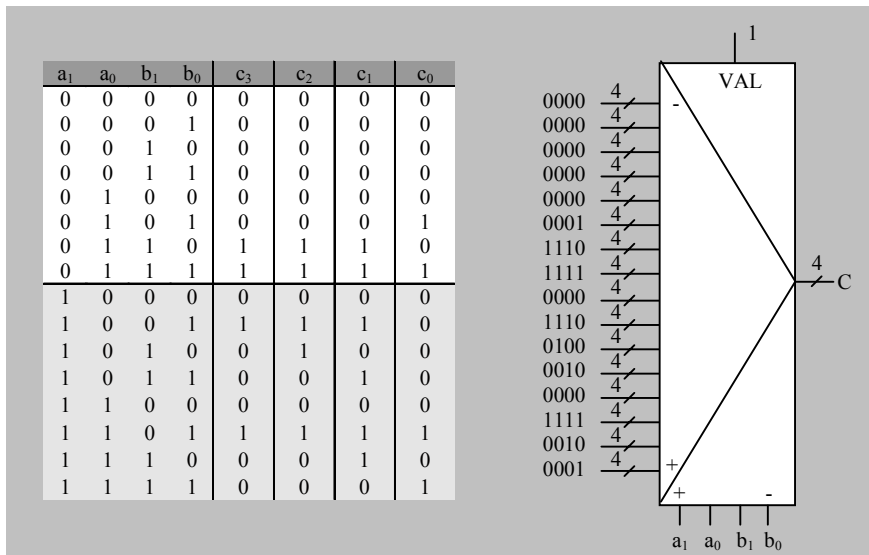
39.

En primer lloc, construirem la taula de veritat del circuit.

El rang d'un número enter representat en complement a 2 amb dos bits és $[-2..1]$. Per tant, el rang del producte de dos d'aquests números és $[-2..4]$. Per a representar números dintre d'aquest rang són suficients quatre bits, ja que amb quatre bits podem representar números en el rang $[-8..7]$.

Per tant, la taula de veritat té quatre entrades i quatre sortides. Les entrades són els dos bits de cadascun dels números A i B, i les sortides són els 4 bits del resultat C.

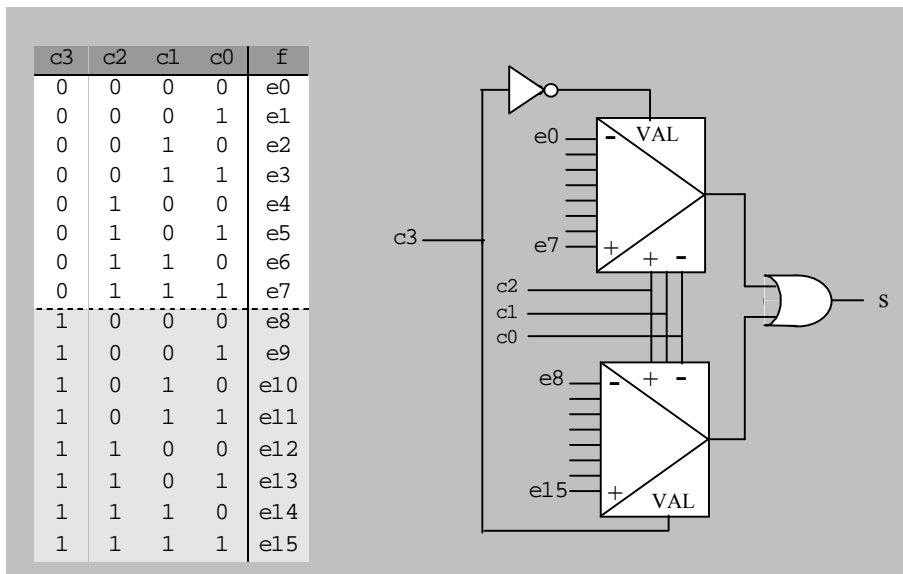
Per implementar el circuit utilitzarem un multiplexor de bussos de 4 bits de 16 entrades de dades. La figura mostra aquesta implementació.



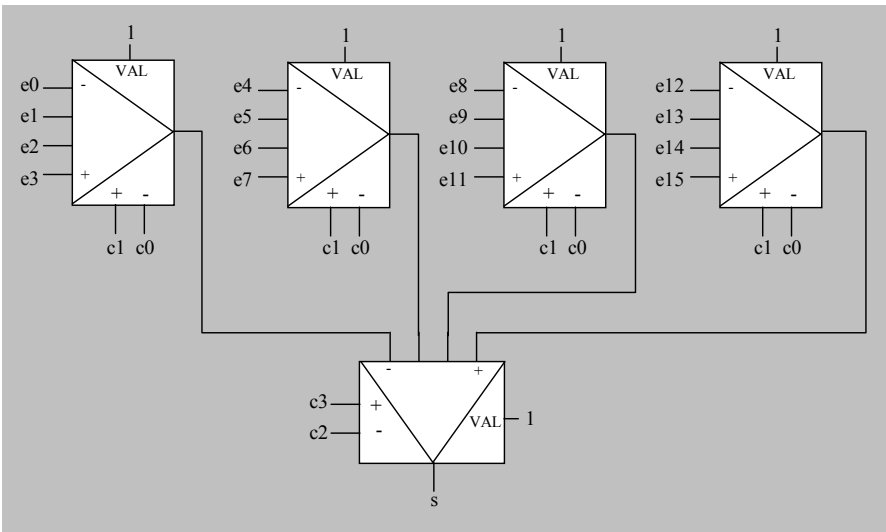
40.

Escrivim primer la taula de veritat corresponent a un multiplexor de 16 entrades de dades. Aquest multiplexor té 4 entrades de control (c₃.. c₀) i 16 entrades de dades (e₁₅..e₀).

La línia discontinua a la taula mostra que es pot descomposar el funcionament del circuit com si es tractés de dos multiplexors de 8 entrades de dades, tots dos controlats pels senyals c₂, c₁ i c₀. El senyal c₃ determina quin dels dos multiplexors funciona en cada moment.



41.



42.

Deduirem les expressions lògiques de cadascuna de les sortides i construirem el circuit a partir d'aquestes. Per a resoldre l'activitat, és convenient tenir al davant la representació gràfica del codificador i la seva taula de veritat (figura 21 dels apunts).

Com que la sortida del codificador ha de codificar en binari l'entrada de més pes que valgui 1, podem fer el raonament següent:

- la sortida s_1 es posarà a 1 quan estiguin a 1 les entrades e_3 o e_2 (en aquests casos s'ha de codificar un 11 o un 10 respectivament).
- la sortida s_0 es posarà a 1 quan estigui a 1 l'entrada e_3 (en aquest cas s'ha de codificar un 11, independentment dels valors de les altres entrades) o bé quan ho estigui l'entrada e_1 i no ho estigui la e_2 (en aquest cas, $[e_3 e_2 e_1] = [0 0 1]$, s'ha de codificar un 01).
- d'altra banda, l'entrada de validació ha d'estar activa perquè el codificador funcioni com a tal.

Per tant, les expressions algebraiques de les sortides són:

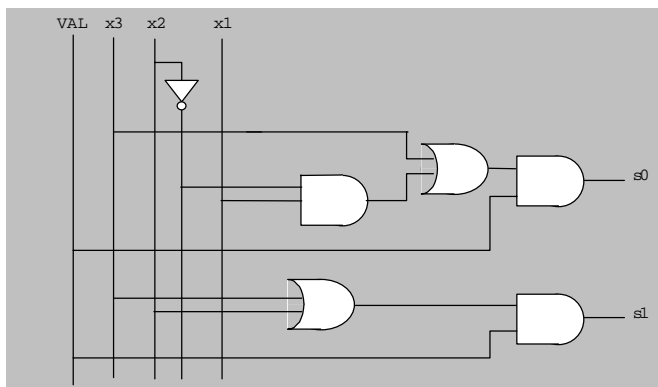
- s_1 valdrà 1 quan $VAL=1$ i e_3 o e_2 siguin 1 (punts 1 i 3):

$$s_1 = VAL (e_3 + e_2)$$

- s_0 valdrà 1 quan $VAL=1$ i e_3 sigui 1, o quan $[e_3 e_2 e_1] = [0 0 1]$ (punts 2 i 3):

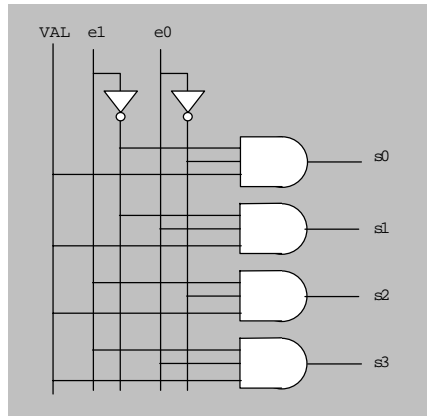
$$s_0 = VAL (e_3 + e_3'e_2'e_1) = VAL (e_3 + e_2'e_1)$$

A continuació es mostra una possible implementació d'aquest codificador. Com es pot veure a la figura, el valor de l'entrada e_0 no té influència en les sortides. Aquest és el motiu que fa que un codificador tingui un 0 a la seva sortida tant quan ha de codificar un 0 com quan cap de les seves entrades té un 1.



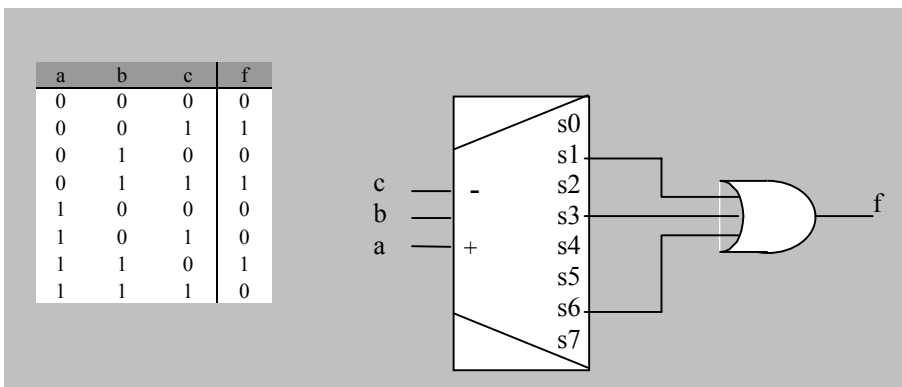
43.

Quan l'entrada de validació està activa, cada sortida d'un descodificador es posa a 1 només quan es dona una combinació determinada de les entrades. Per tant, la implementació de cadascuna de les sortides serà un circuit que detecti que aquesta combinació està present a l'entrada, tal com es mostra a la figura.

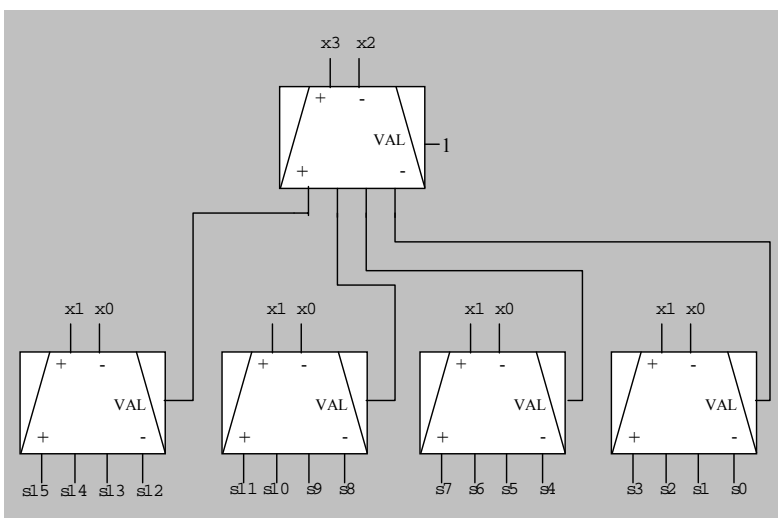


44.

Primer farem la taula de veritat, i a partir de la taula utilitzarem un descodificador per a sintetitzar el circuit. La porta OR fa la suma lògica dels casos en què la funció val 1.



45.



46.

Per tal de deduir què fa el circuit, construirem la seva taula de veritat. La taula mostra també el valor de l'entrada i la sortida quan les interpretem com a números representats en complement a 2 (columnes X i Y).

X	x ₂	x ₁	x ₀	y ₂	y ₁	y ₀	Y
0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	-1
2	0	1	0	1	1	0	-2
3	0	1	1	1	0	1	-3
-4	1	0	0	1	0	0	-4
-3	1	0	1	0	1	1	3
-2	1	1	0	0	1	0	2
-1	1	1	1	0	0	1	1

Com es pot veure a la taula, el què fa el circuit és canviar el signe del número de l'entrada. En el cas particular $X = -4$ això no és possible, perquè el 4 no pot representar-se en complement a 2 amb només tres bits (el resultat és erroni en aquest cas: es produeix sobreiximent).

47.

a) E_0 és 1 quan les dues entrades del descodificador són 0, és a dir, quan $x_1=0$ i $x_0x_3=0$. Per tant, l'expressió algebraica per a E_0 és $E_0 = x_1'(x_0x_3)'$. Seguint el mateix raonament per la resta de sortides, obtenim:

$$E_0 = x_1'(x_0x_3)'$$

$$E_1 = x_1'(x_0x_3) = x_1'x_0x_3$$

$$E_2 = x_1(x_0x_3)'$$

$$E_3 = x_1(x_0x_3) = x_1x_0x_3$$

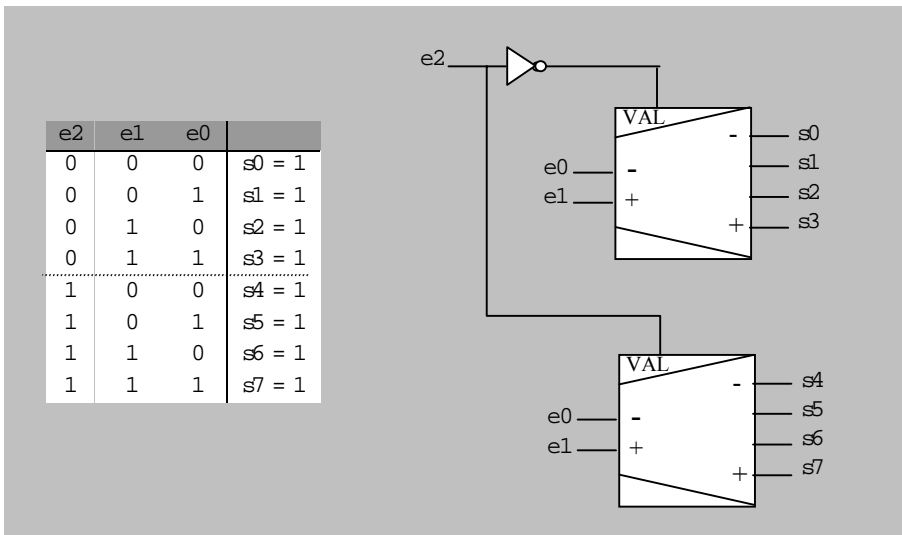
b)

La taula de veritat es troba a la figura següent. Per a obtenir la columna corresponent a F hem fet un pas intermig: hem posat en una columna el què val F en termes de les entrades de dades del multiplexor (E_i), i a la columna final hem substituït les variables E_i pel valor que prenen per cada combinació de x_i .

	x ₃	x ₂	x ₁	x ₀	F	F
	0	0	0	0	E_0	1
	0	0	0	1	E_0	1
	0	0	1	0	E_0	0
	0	0	1	1	E_0	0
	0	1	0	0	E_1	0
	0	1	0	1	E_1	0
	0	1	1	0	E_1	0
	0	1	1	1	E_1	0
	1	0	0	0	E_2	0
	1	0	0	1	E_2	0
	1	0	1	0	E_2	1
	1	0	1	1	E_2	0
	1	1	0	0	E_3	0
	1	1	0	1	E_3	0
	1	1	1	0	E_3	0
	1	1	1	1	E_3	1

48.

Aquesta activitat és molt similar al problema 40. A la figura següent es mostra la taula de veritat del descodificador 3-8, que té 3 entrades (e2..e0), i 8 sortides (s0..s7). Per simplificar la taula, s'ha posat una única columna de sortida en la que hem indicat quina de les sortides val 1; totes les altres valen 0. Com es pot veure a la taula de veritat, l'entrada e2 determina quin dels dos descodificadors 2-4 funciona en cada moment (i per tant controla la seva entrada de validació). A cada descodificador 2-4, les entrades e1 i e0 determinen quina de les sortides ha de valer 1.



49.

Per resoldre aquest exercici construirem la seva taula de veritat i després interpretarem els números com a naturals.

Per claredat en la solució utilitzarem dues variables intermitges, x_1 i x_0 , que corresponen a les sortides del codificador (x_1 a la de més pes i x_0 a la de menys). $X = [x_1 x_0]$.

A	a ₁	a ₀	z	x ₁	x ₀	X	s ₁	s ₀	S
0	0	0	0	0	1	1	0	0	0
1	0	1	0	1	0	2	0	1	1
2	1	0	0	1	1	3	1	0	2
3	1	1	0	0	0	0	1	1	3
0	0	0	1	0	1	1	0	0	1
1	0	1	1	1	0	2	0	0	2
2	1	0	1	1	1	3	0	0	3
3	1	1	1	0	0	0	0	0	0

Tal com podem veure a la taula,

$$X = (A + 1) \text{ mod } 4$$

La variable z selecciona entre les entrades A i X, i per tant la sortida es pot expressar amb la taula següent:

z	S
0	A
1	(A+1) mod 4

Això es pot resumir en aquesta expressió:

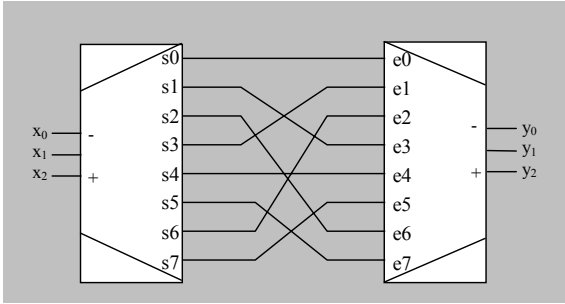
$$S = (A + z) \text{ mod } 4$$

50.

La taula de veritat és la següent:

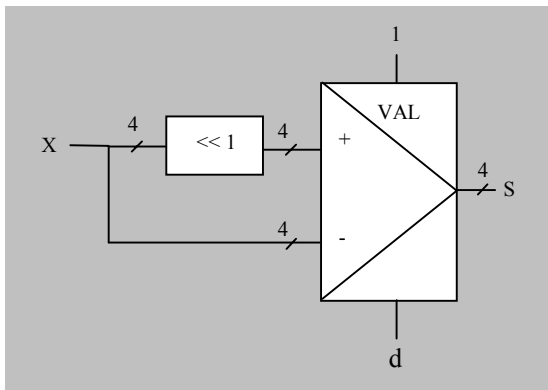
X	x ₂	x ₁	x ₀	y ₂	y ₁	y ₀	Y=3X mod 8
0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	3
2	0	1	0	1	1	0	6
3	0	1	1	0	0	1	1
4	1	0	0	1	0	0	4
5	1	0	1	1	1	1	7
6	1	1	0	0	1	0	2
7	1	1	1	1	0	1	5

A continuació es mostra el disseny.



51.

- a) A la figura es mostra la implementació del circuit. El senyal d determina si la sortida S val el mateix que l'entrada X (d = 0) o X decalat un bit a l'esquerra (d = 1).



- b) Aquest desplaçament d'un bit a la esquerra és equivalent a multiplicar l'entrada per 2. Per tant, $S = 2 \cdot X$.

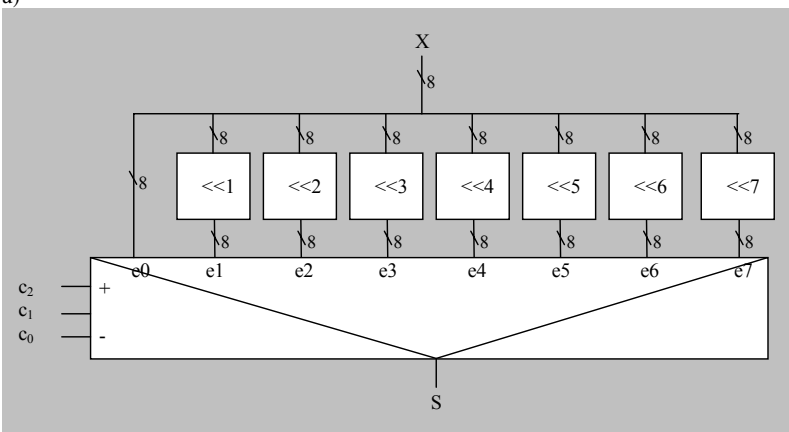
- c) Es produeix sobreeximent quan el resultat no es pot representar amb 4 bits.
 - Si interpretem els números X i S com a números naturals codificats en binari, el rang d'X i S és [0..15]. Per tant, quan X sigui més gran que 7, S no podrà representar el resultat i es produirà sobreeximent.
 Sobreeximent si $X > 7$.

- Si interpretem els números com a enters codificats en complement a 2, el rang d'X i S és [-8..7]. Per tant, es produirà sobreeximent quan X sigui més petit que -4 o més gran que 3.

Sobreeximent si $X < -4$ o $X > 3$

52.

a)



b)

$S = X \cdot 2^n$

c)

Si $C = 010$, el decalatge correspon a l'operació de multiplicar per 4. Si X és natural, el rang representable amb 8 bits és $[0..255]$. Per tant, es produeix sobreiximent quan $X \geq 256 / 4 = 64$.

Si X és enter, el rang de números representables és $[-128..127]$. Per tant, es produeix sobreiximent quan $X \geq 128 / 4 = 32$ o bé quan $X < -128 / 4 = -32$.

d)

Si X és un número natural, els bits de més pes que s'afegeixen al número quan aquest es desplaça a la dreta han de valdre 0. Per tant, s'ha d'usar un decalador lògic.

Si X és un número enter, els bits de més pes que s'afegeixen al número quan aquest es desplaça a la dreta han de valdre el mateix que el bit de més pes del número, per tal de conservar-ne el signe. Per tant, s'ha d'usar un decalador aritmètic.

e)

Un decalador a la dreta no computa la funció $X / 2^n$, sinó la funció $\lfloor X / 2^n \rfloor$. Per tant, si s'usa per a calcular $X / 2^n$, el resultat no serà exacte si algun dels bits que es perden en desplaçar a la dreta valia 1. Dit d'una altra manera, si el desplaçament és de k bits ($C = k$), el resultat és inexacte quan

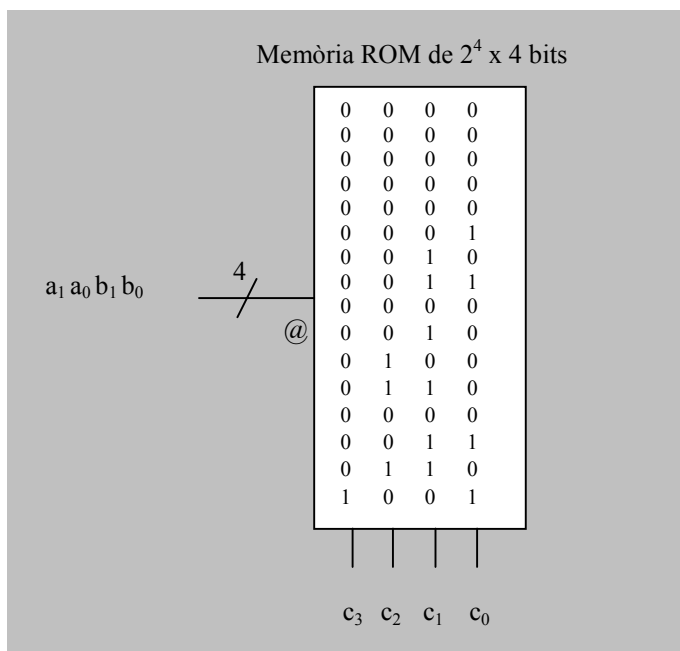
$$X \bmod 2^k \neq 0.$$

54.

El circuit multiplicador de dos números naturals de 2 bits té 4 entrades ($a_1 a_0 b_1 b_0$) i quatre sortides ($c_3 c_2 c_1 c_0$), tal com s'explica al problema 27.

Quatre variables poden prendre $2^4 = 16$ combinacions diferents, per tant la grandària de la ROM és de 16 mots de 4 bits (un per cada funció de sortida).

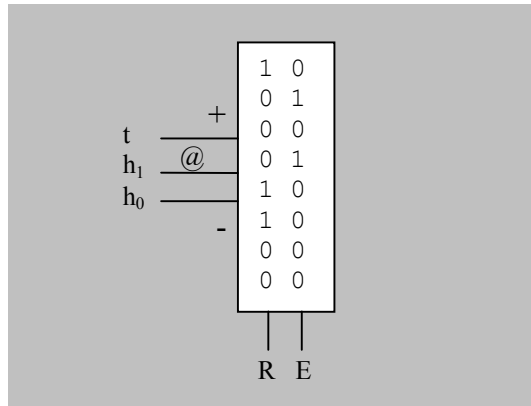
A la figura següent es mostra el seu contingut, que coincideix amb la part dreta de la taula de veritat del problema 27. La variable a_1 està connectada al bit de més pes del bus d'adreces, i la variable b_0 al bit de menys pes (és molt important indicar en el bus d'adreces a quin senyal d'entrada es connecta el bit de més pes per tal que el resultat sigui correcte. Si els bits d'entrada estiguessin en un altre ordre, la taula de veritat seria diferent i per tant també el contingut de la ROM).



55.

Tal com s'explica al problema 21, aquest circuit té 3 senyals d'entrada i 2 de sortida, i per tant la grandària de la memòria ROM és de $2^3 \times 2$ bits (8 mots de 2 bits).

Al contingut de la memòria s'han substituït les "x" de la taula de veritat per 0s, ja que el valor de qualsevol bit en una memòria ROM ha d'estar definit. No obstant, les "x" es podrien haver substituït per qualsevol valor binari.



56.

a) La taula de veritat del circuit és la següent:

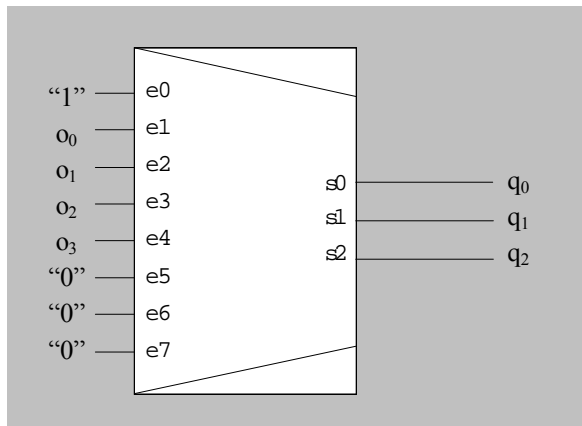
e_3	e_2	e_1	e_0	o_3	o_2	o_1	o_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	1
0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	1
1	0	1	0	0	0	1	1
1	0	1	1	0	1	1	1
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	1
1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	1

b) Les sortides del bloc *ORDENAR* tenen tots els 1s a la dreta, tal com es mostra a la següent taula:

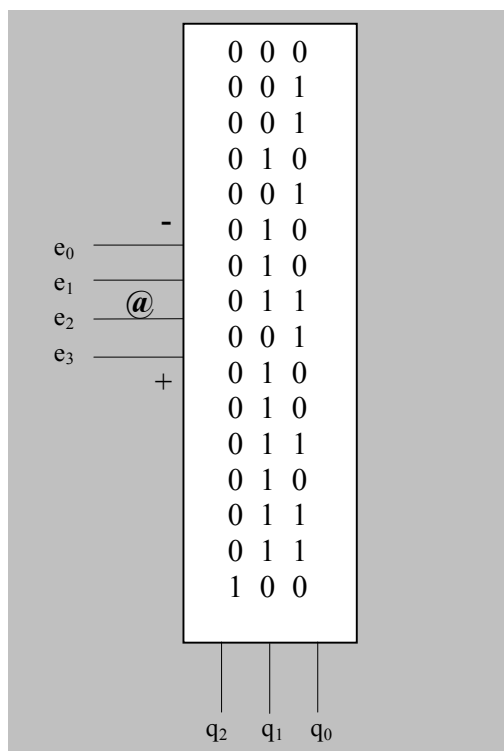
o_3	o_2	o_1	o_0	QUANTS	q_2	q_1	q_0
0	0	0	0	0 (no hi ha 1s a l'entrada)	0	0	0
0	0	0	1	1 (hi ha un 1 a l'entrada)	0	0	1
0	0	1	1	2 (hi ha dos 1s a l'entrada)	0	1	0
0	1	1	1	3 (hi ha tres 1s a l'entrada)	0	1	1
1	1	1	1	4 (hi ha quatre 1s a l'entrada)	1	0	0

Fixem-nos que la sortida QUANTS ha de formar la codificació binària d'un dels números 0, 1, 2, 3 o 4. Per a sintetitzar-la, per tant, podem usar un codificador 8-3 (calen 3 bits per a codificar el 4 en binari). Concretament, tenim que si el bit més alt d'ORDRE que està a 1 és o_i , llavors QUANTS ha de valdre $i+1$. Per tant, connectarem el senyal o_i a l'entrada $i+1$ del codificador, tal com es mostra a la figura següent.

A l'entrada 0 del codificador pot haver-hi tant un 1 com un 0 (en tots dos casos, QUANTS valdrà 0 si no hi ha cap bit d'ORDRE a 1).



c) La grandària de la ROM ha de ser $2^4 \times 3$ bits, i el seu contingut és el següent:



57.

Primer deduirem l'expressió algebraica de cadascuna de les funcions. Anomenarem a_1 , a_0 , b_1 i b_0 als bits dels números d'entrada. Per què els números siguin iguals, ho han de ser bit a bit. Per tant, s'ha de complir que $a_0=b_0$ i $a_1=b_1$.

La porta XOR permet detectar la desigualtat entre dos bits (recordeu la seva taula de veritat). Per tant, una negació de la seva sortida detectarà la igualtat. Així doncs, l'expressió per la sortida "A = B" és

$$(a_1 \oplus b_1)'(a_0 \oplus b_0)'$$

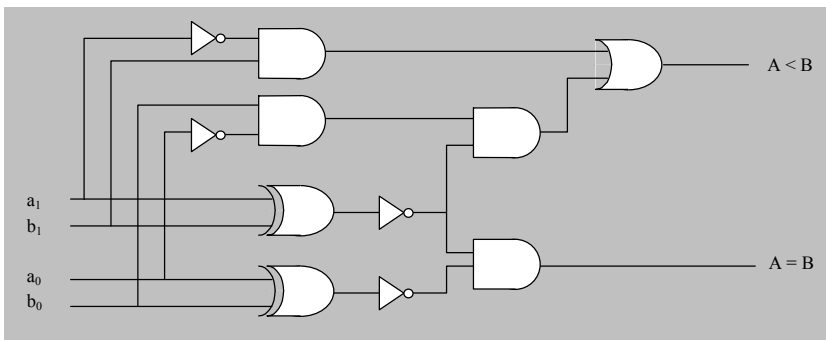
Per fer la sortida "A < B" tindrem en compte que:

- A és més petit que B si ho és el seu bit de més pes: $b_1=1$ i $a_1=0$.
- Si els dos bits de més pes d'A i B són iguals, llavors A és més petit que B si ho és el seu bit de menys pes: $a_1=b_1$ i $b_0=1$ i $a_0=0$.

Per tant, l'expressió per la sortida "A < B" és

$$b_1 a_1' + (b_1 \oplus a_1)' b_0 a_0'$$

A la figura següent es troba la implementació dels dos circuits.



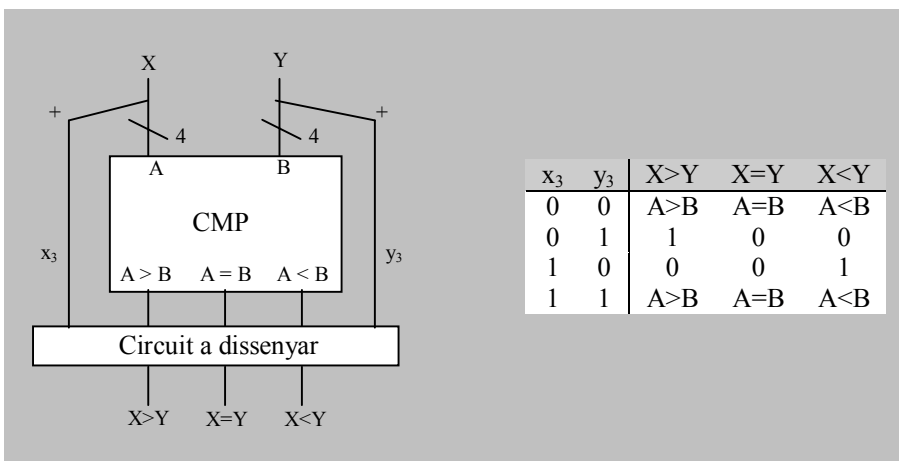
59.

El resultat de la comparació depèn dels signes dels dos números (que vénen donats pel seu bit de més pes: 1 pels negatius, 0 pels positius o zero).

Si els dos números són de signes diferents, llavors el més gran és el positiu.

Quan els dos números són del mateix signe, llavors els podem comparar usant un comparador de números naturals. Fixem-nos que el resultat serà correcte també en el cas que els dos números siguin negatius, perquè si $X > Y$ interpretant X i Y en complement a 2, llavors també es compleix que $X > Y$ si els interpretem en binari. Per exemple, si comparem $X = -1$ (1111) amb $Y = -2$ (1110) el comparador trauria un 1 per la sortida $X > Y$.

A continuació es mostra un primer esquema del circuit, i la taula de veritat que descriu el comportament de la part del circuit que queda per implementar, en funció del signe dels dos números.



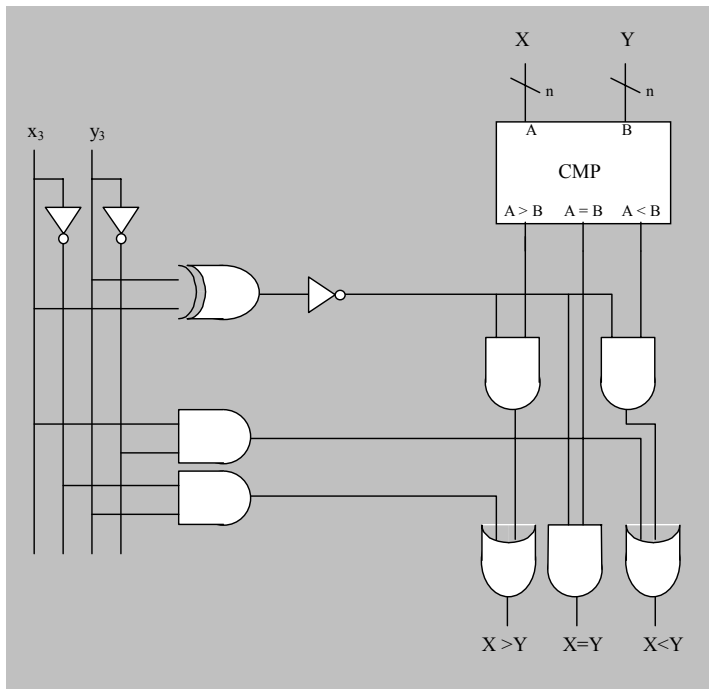
D'aquestes taules es dedueixen les següents igualtats:

$$[X > Y] = x_3' y_3 + (x_3 \oplus y_3)' \cdot [A > B]$$

$$[X < Y] = x_3 y_3' + (x_3 \oplus y_3)' \cdot [A < B]$$

$$[X = Y] = (x_3 \oplus y_3)' \cdot [A = B]$$

A continuació es mostra el circuit complet:

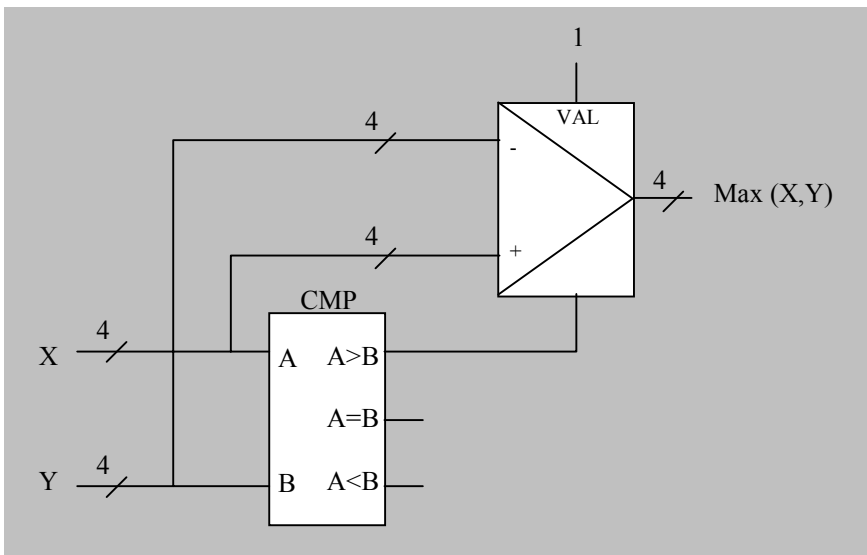


60.

Per a decidir quin dels dos números és més gran utilitzarem un comparador de 4 bits. Connectarem el número X a l'entrada A i el número Y a l'entrada B del comparador. La sortida "A > B" d'aquest comparador valdrà 1 si $X > Y$, i 0 en cas contrari.

Aquesta sortida controlarà un multiplexor de busos que seleccionarà entre X i Y. Així, quan "A > B" = 1 seleccionarem el número X, i en qualsevol altre cas ($X \leq Y$) seleccionarem el número Y.

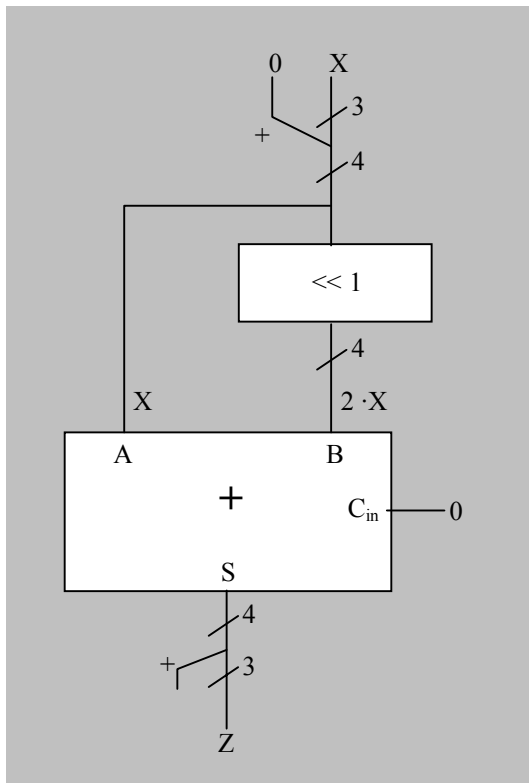
La figura següent mostra aquest disseny.



61.

Realitzar l'operació $X \bmod 2^n$ en binari consisteix en quedar-se amb els n bits de menor pes d'X. Per tant, per implementar aquesta funció serà suficient amb sumar $X + 2 \cdot X$ (X decalat un bit a l'esquerra) i quedar-se amb els 3 bits de menor pes de la sortida del sumador ($[z_2 z_1 z_0]$).

La figura següent mostra aquest disseny.



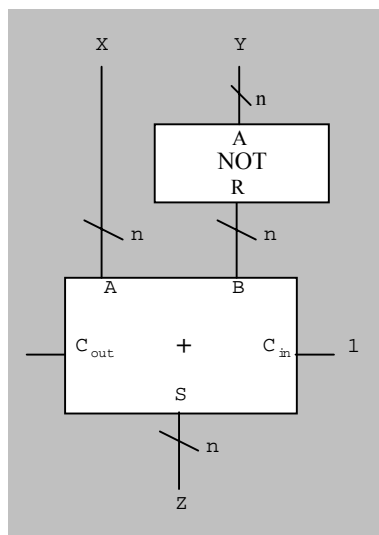
62.

Per fer aquesta operació aprofitarem la següent igualtat, vàlida quan els números estan representats en complement a 2 (tal com es va veure al mòdul “Representació de la informació”):

$$Z = X - Y = X + (-Y) = X + Y' + 1$$

Per tant, per a implementar el circuit ens cal un bloc NOT d'n bits per a obtenir Y' , i un sumador d'n bits per fer la suma $X + Y'$. L'1 que encara queda per sumar es pot connectar a l'entrada de transport del sumador.

La figura següent mostra el disseny proposat.



63.

Es tracta de dissenyar un circuit que faci el següent:

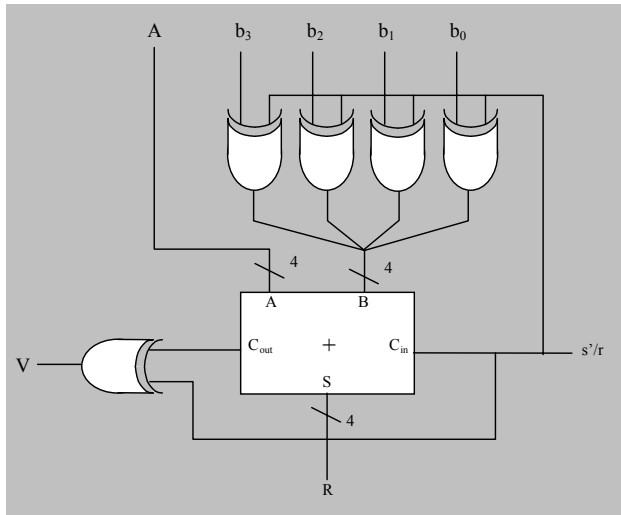
Si $s'/r = 0$, llavors $R = A + B$

Si $s'/r = 1$, llavors $R = A - B = A + B' + 1$

Podem utilitzar un sumador de números de 4 bits per fer aquest circuit.

- A una entrada hi connectarem el número A
- A l'altra entrada hi connectarem el número B o el número B', en funció de si el senyal s'/r val 0 o 1 respectivament. Recordem aquesta propietat de la funció XOR:
 $0 \oplus x = x, 1 \oplus x = x'$
 Per tant, per obtenir B o B' en funció de s'/r podem fer una XOR de s'/r amb cadascun dels bits de B
- A l'entrada de transport hi connectarem un 0 si s'/r = 0, o bé un 1 si s'/r = 1. Per tant, hi podem connectar directament el senyal s'/r.

El circuit sumador/restador que es descriu als paràgrafs anteriors es mostra a continuació.



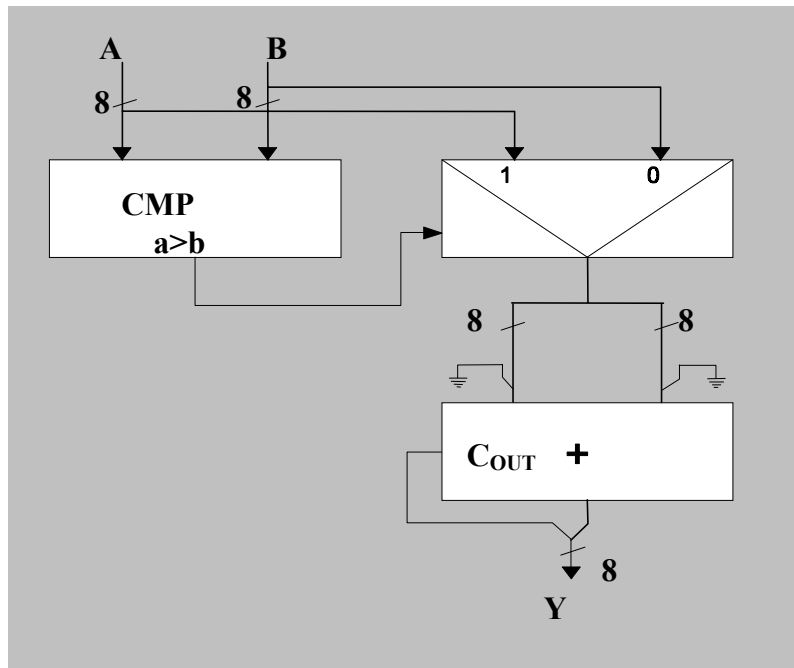
Quan es fa una suma, la sortida V coincideix amb l'últim bit de transport.
 Quan es fa la resta mitjançant la suma del complementari més 1, el transport de l'últim bit és sempre el contrari del que obtindríem si s'hagués fet directament la resta.
 Per tant, quan s'/r=0 es compleix que $V = C_{out}$, i quan s'/r=1 es compleix que $V = C_{out}'$. Així doncs, utilitzant la mateixa propietat de la funció XOR que hem usat abans, obtenim que

$$V = s'/r \oplus C_{out}$$

64.
a)

$$0'75 x = \frac{1+2}{4} x = \frac{x+2x}{4}$$

2x s'aconsegueix desplaçant un bit a l'esquerra, mentre que dividir entre quatre s'aconsegueix amb un desplaçament de dos bits a la dreta.

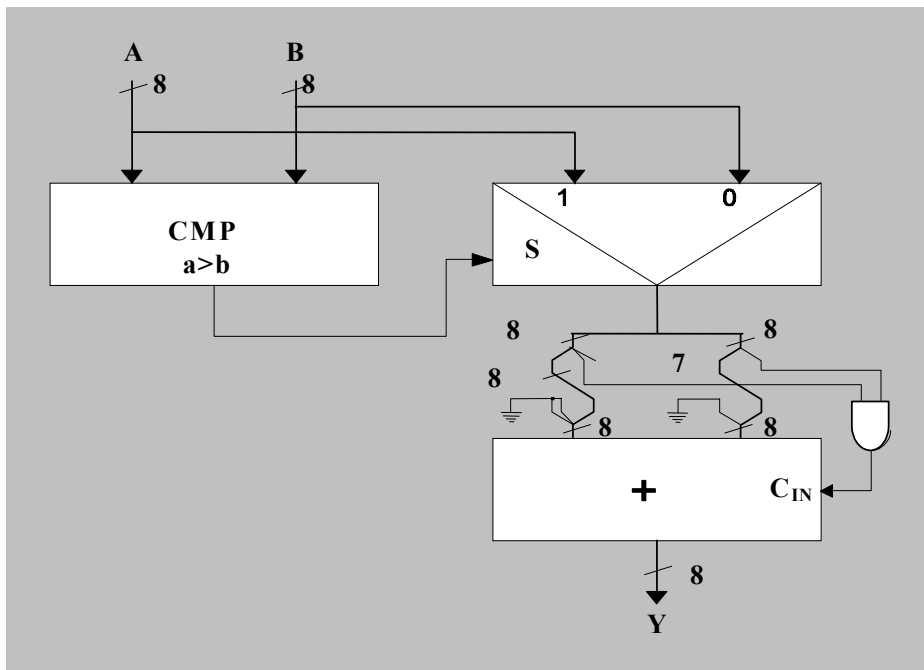


El sumador ha d'ésser de 9 bits, ja que aquests són els bits que necessitem per a representar $2x$. Els dos bits de menys pes de la suma no els tenim en compte (són els decimals), i completem el número agafant el bit de transport com a bit de més pes del resultat.

b)

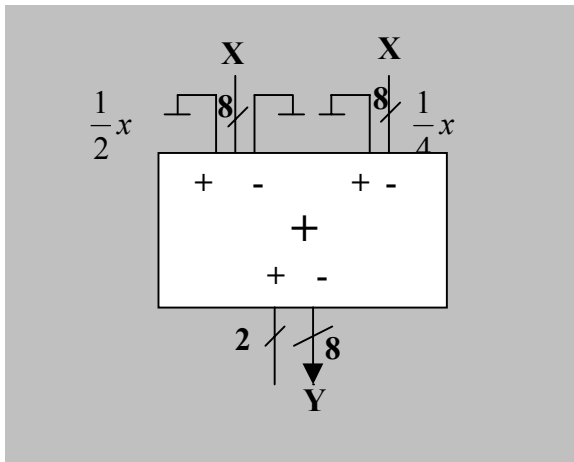
$$0'75 x = \frac{1}{2} x + \frac{1}{4} x$$

$1/2$ s'aconsegueix desplaçant un bit a la dreta, mentre que $1/4$ s'aconsegueix amb un desplaçament de dos bits a la dreta.



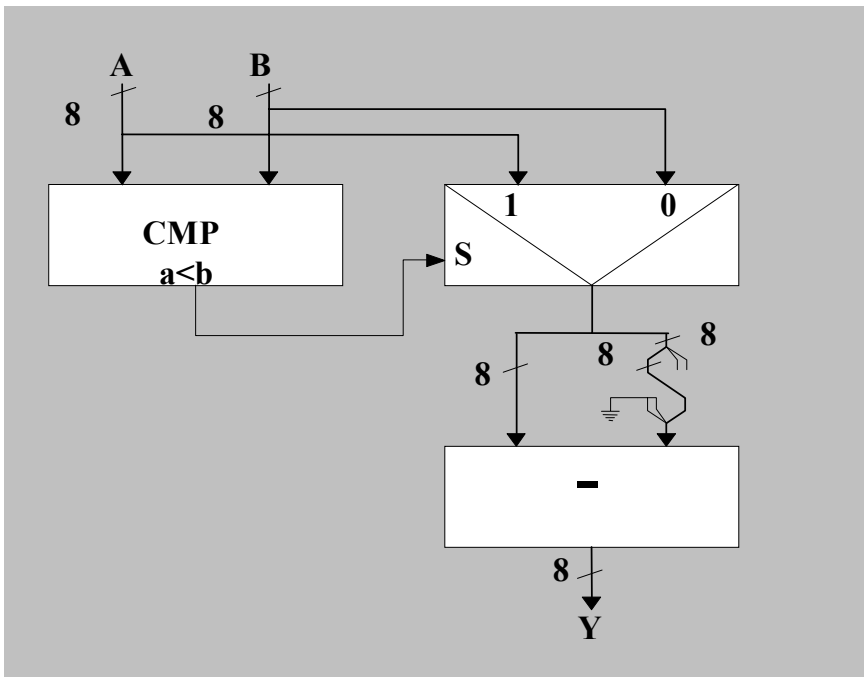
La porta *AND* es posa a l'entrada *Cin* per què, si la suma dels decimals que no s'han tingut en compte és més gran que 1 ($0'75+0'5$ en el cas que els bits desplaçats siguin 1), hem de sumar 1 al resultat; d'altra forma el resultat no seria correcte.

Un altre manera de fer-ho, per tal de tenir en compte tots els bits, és utilitzar un sumador de més bits, tal com es mostra a la següent figura:



$$c) 0'75 x = x - \frac{1}{4} x$$

Com en el primer cas, $1/4$ s'aconsegueix amb un desplaçament de dos bits a la dreta.



65.

- a) La taula de veritat és la següent (hi escrivim també la interpretació dels números en complement a 2 per fer-la més entenedora):

A	B	a ₁	a ₀	b ₁	b ₀	m ₃	m ₂	m ₁	m ₀	M=A*B
0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	-2	0	0	1	0	0	0	0	0	0
0	-1	0	0	1	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	1	1
1	-2	0	1	1	0	1	1	1	0	-2
1	-1	0	1	1	1	1	1	1	1	-1
-2	0	1	0	0	0	0	0	0	0	0
-2	1	1	0	0	1	1	1	1	0	-2
-2	-2	1	0	1	0	0	1	0	0	4
-2	-1	1	0	1	1	0	0	1	0	2
-1	0	1	1	0	0	0	0	0	0	0
-1	1	1	1	0	1	1	1	1	1	-1
-1	-2	1	1	1	0	0	0	1	0	2
-1	-1	1	1	1	1	0	0	0	1	1

b) Per obtenir la solució que es descriu a l'enunciat, amb un sumador i dos multiplicadors, s'ha d'utilitzar la següent igualtat:

$$A^4 + A^3 + A^2 = A^2 (A^2 + A + 1)$$

A continuació farem un estudi del rang que poden tenir cadascuna de les expressions per tal de decidir el número de bits de cada bloc.

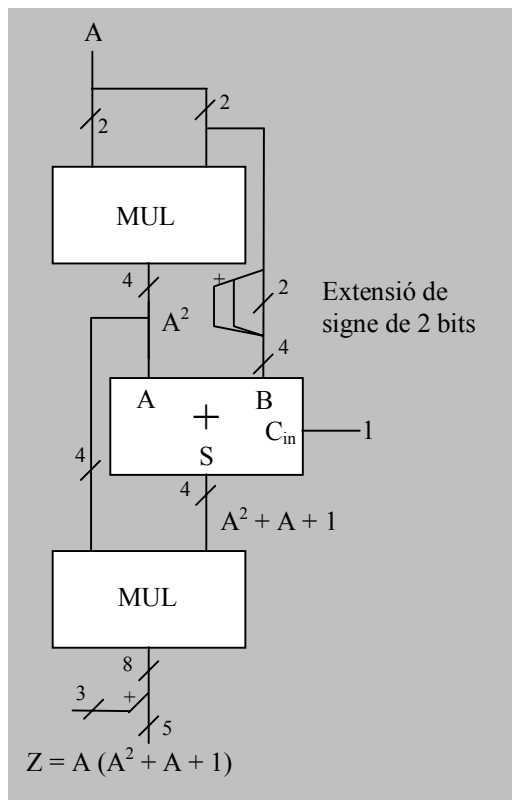
A	A	A ²	A ² +A+1	A ² ·(A ² +A+1) = A ⁴ +A ³ +A ²
00	0	0	1	0
01	1	1	3	3
10	-2	4	3	12
11	-1	1	1	1

calen: 4 bits 3 bits 5 bits

A² s'obté mitjançant un bloc MUL de 2 bits, connectant A a les dues entrades. A², doncs, tindrà 4 bits (el doble de bits que les entrades).

A² + A + 1 s'obté mitjançant un sumador, connectant l'1 a l'entrada C_{in}. Tot i que si ens fixem en el rang del resultat d'aquesta suma amb 3 bits n'hi hauria prou, el sumador ha de ser de 4 bits perquè A² té 4 bits. Per tant, caldrà ampliar a 4 bits l'entrada A (fent una extensió de signe) per connectar-la a l'altra entrada del sumador, tal com es mostra a la figura.

Un altre bloc MUL de 4 bits calcularà A² (A² + A + 1). La sortida del multiplicador serà doncs de 8 bits, tot i que el rang del resultat només en necessita 5. El resultat està format, per tant, pels 5 bits de menor pes de la sortida d'aquest segon bloc MUL.

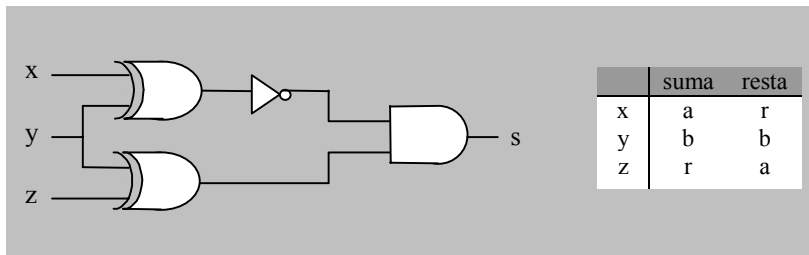


66.

a) Hem de dissenyar una UAL que pugui realitzar 4 operacions. Algunes d'elles es poden fer directament usant blocs que ja hem estudiat a la teoria d'aquest mòdul, però d'altres requereixen la utilització de circuits que combinin blocs amb portes. Analitzarem una per una les operacions, per tal de decidir com farem la seva implementació. Tal com hem vist a la teoria, una vegada implementats els circuits que fan totes les operacions, seleccionarem l'operació que s'ha de fer mitjançant un multiplexor de busos de 4 bits de 4 entrades.

- $R = A+B$ Aquesta suma es pot fer directament amb un sumador de 4 bits.
- $R = A-B$ Tal com s'ha vist al problema 61, $A-B = A+B'+1$. Per tant, per implementar aquesta resta farà falta un sumador i un bloc NOT de 4 bits.
- $R = A$ L'entrada A , de 4 bits, estarà connectada directament a una de les entrades del multiplexor.
- $R = -A$ Igual com hem fet en el cas de la resta, podem usar la igualtat $-A = A'+1$, i per tant per implementar aquesta operació necessitarem un sumador i un bloc NOT de 4 bits.

A la figura següent es pot veure la implementació descrita.



- operació $R = -A$: es produeix sobreiximent només en el cas $A = -8$, i per tant

$$V = a_3 a_2' a_1' a_0'$$

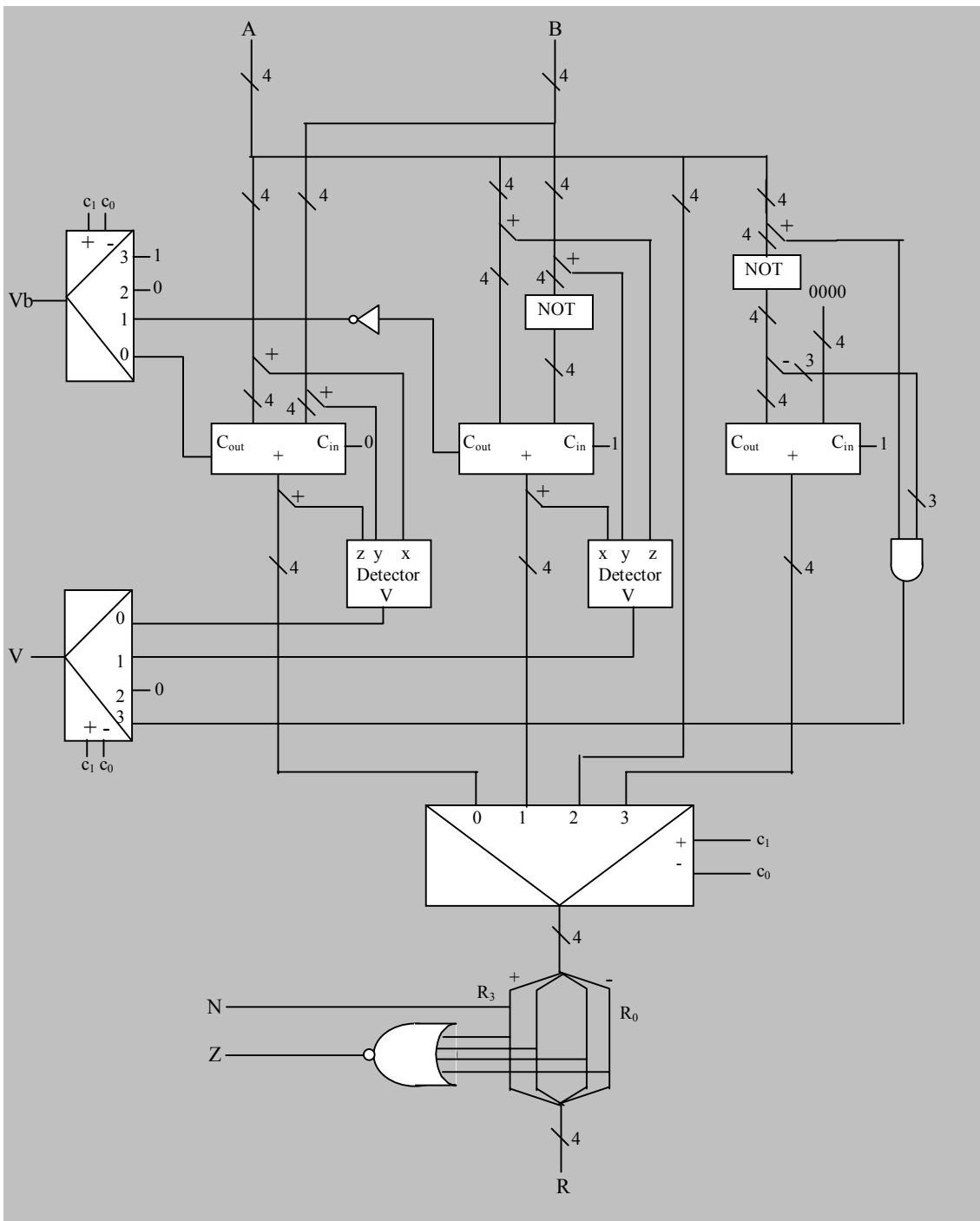
La taula següent resumeix el valor del bit V:

c_1	c_0	V
0	0	$(a_3 \oplus b_3)'(b_3 \oplus r_3)$
0	1	$(r_3 \oplus b_3)'(b_3 \oplus a_3)$
1	0	0
1	1	$a_3 a_2' a_1' a_0'$

Per a implementar els dos bits de sobreiximent usarem dos multiplexors, tots dos controlats pels bit c_1 i c_0 . El sobreiximent de cadascuna de les operacions es calcularà per separat, i els multiplexors seleccionaran en cada moment el valor correcte de V i Vb segons l'operació que s'estigui realitzant.

- N: Tal com s'ha vist a la teoria, el signe del resultat és el bit de més pes.
- Z: Tal com s'ha vist a la teoria, per calcular si el resultat és 0 es fa amb una porta NOR de 4 entrades.

A continuació es mostra el circuit.



67.

El circuit d'aquesta UAL es mostra a la figura següent. Observem que l'entrada B es connecta a les 4 entrades de menys pes del multiplexor.

