Ηλεκτρονική ΙΙ

Τμήμα Φυσικής Πανεπιστημίου Θεσσαλίας

Μέρος 2

Περιεχόμενα

- Συνδυαστικά Κυκλώματα

Συνδυαστικά Κυκλώματα

Συνδυαστικό Κύκλωμα

- Ένα Συνδυαστικό Κύκλωμα (ΣΚ) n εισόδων και m εξόδων περιγράφεται από m λογικές συναρτήσεις n μεταβλητών.
- Η κάθε μία από τις n μεταβλητές εισόδου μπορεί να πάρει δύο μόνο τιμές, το λογικό “1” και το λογικό “0”.
- Επομένως, οι δυνατοί συνδυασμοί των μεταβλητών εισόδου είναι 2^n.
- Για κάθε συνδυασμό των μεταβλητών εισόδου, η κάθε μία μεταβλητή εξόδου παίρνει μία μόνο τιμή: το λογικό “1” ή το λογικό “0”.
- Ο πίνακας αληθείας της λογικής συνάρτησης περιγράφει αυτή τη σχέση εισόδων-εξόδου.

Σχεδίαση Συνδυαστικών Κυκλωμάτων

- Για να σχεδιάσουμε ένα Συνδυαστικό Κύκλωμα ακολουθούμε τα εξής βήματα:
  – Κατασκευάζουμε τον πίνακα αληθείας του Συνδυαστικού Κυκλώματος
  – Γράφουμε τις λογικές συναρτήσεις των εξόδων συναρτήσει των εισόδων
  – Απλοποιούμε τις συναρτήσεις χρησιμοποιώντας χάρτες Karnaugh
  – Σχεδιάζουμε το κύκλωμα τηρώντας την προτεραιότητα των πράξεων

.

Παράδειγμα 1

- Να σχεδιαστεί ένα Συνδυαστικό Κύκλωμα (ΣΚ) που αναγνωρίζει αν ένας 3-bit αριθμός είναι μικρότερος από 3, χρησιμοποιώντας μόνο πύλες: A) NOT και πύλες AND και OR δύο εισόδων, B) NAND/NOR
- Το ΣΚ έχει τρεις εισόδους A, B και C, που αποτελούν τη δυαδική αναπαράσταση ενός δεκαδικού αριθμού από το 0 έως και το 7 (με 3 bit μπορούμε να μετρήσουμε $2^3=8$ αριθμούς) και μία έξοδο Y. Η έξοδος του ΣΚ είναι "1" όταν το δεκαδικό ισοδύναμο του 3-bit δυαδικού αριθμού των εισόδων του ΣΚ είναι μικρότερο από 3.

Παράδειγμα 1

Από την περιγραφή της λειτουργίας του ΣΚ κατασκευάζεται ο παρακάτω πίνακας αληθείας του ΣΚ:

δεκαδικός	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

< 3

Επομένως, η συνάρτηση εξόδου του ΣΚ ευρίσκεται ως συνάρτηση των εισόδων του: $Y=A'B'C'+A'B'C+A'BC'$

Παράδειγμα 1

Ο χάρτης Karnaugh της συνάρτησης εξόδου του ΣΚ είναι:

B

		00	01	11	10
	0	1	1		1
A	1				

C

Η απλοποιημένη συνάρτηση είναι: Y=A’B’+A’C’

Παράδειγμα 1

Η συνάρτηση γράφεται:

$$\mathbf{Y=A'B'+A'C'=A'(B'+C')=A'(BC)'=(A+BC)'}$$

Για τη σχεδίαση του κυκλώματος, ξεκινώντας από την έξοδο προς τις εισόδους του κυκλώματος, σχεδιάζονται οι πύλες του κυκλώματος λαμβάνοντας υπόψη τις λογικές πράξεις της συνάρτησης εξόδου του ΣΚ.

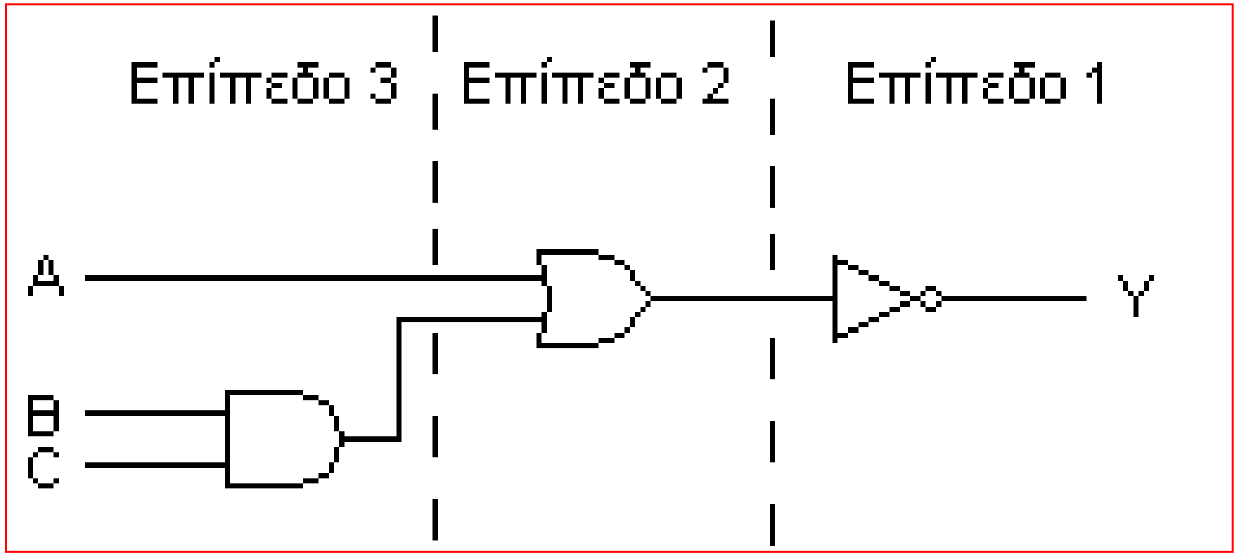
- Το κύκλωμα χωρίζεται σε επίπεδα που περιέχουν τις πύλες, <u>με βάση την προτεραιότητα των πράξεων</u>. Ξεκινώντας από την έξοδο του ΣΚ προς τις εισόδους του ΣΚ, το κύκλωμα χωρίζεται σε τρία επίπεδα πυλών.

Παράδειγμα 1

Επίπεδο 1. Μία πύλη **NOT** που χρησιμοποιείται για την εύρεση της εξόδου Y=(A+BC)’ του ΣΚ, αποτελεί το τελευταίο επίπεδο πυλών.

Επίπεδο 2. Μία πύλη **OR** δύο εισόδων που χρησιμοποιείται για τον υπολογισμό A+BC, αποτελεί το δεύτερο επίπεδο πυλών.

Επίπεδο 3. Μία πύλη **AND** δύο εισόδων, που χρησιμοποιείται για τον υπολογισμό BC, αποτελεί το πρώτο επίπεδο πυλών.



Παράδειγμα 2

Το σύστημα ασφάλειας του χρηματοκιβωτίου μιας τράπεζας έχει δυο πόρτες, την εξωτερική (X) και την εσωτερική (S) που οδηγεί στο χώρο που φυλάγεται ο χρυσός της τράπεζας. Οι δυο αυτές πόρτες έχουν ηλεκτρονικές κλειδαριές με θέσεις για τρία κλειδιά τα οποία έχουν:

Ο Διευθυντής (D) - Ο Υποδιευθυντής (Y) - Ο Ταμίας (T).

Η εξωτερική πόρτα ανοίγει με τα κλειδιά οποιονδήποτε δύο από τους παραπάνω υπαλλήλους.

Η εσωτερική πόρτα ανοίγει μόνο με τα κλειδιά και των τριών υπαλλήλων.

Συμβολίστε με λογικό ‘1’ την παρουσία ενός υπαλλήλου με το κλειδί και με λογικό ‘0’ το αντίθετο.

Συμβολίστε με λογικό ‘1’ την κατάσταση μία πόρτα να είναι ανοιχτή και με λογικό ‘0’ την κατάσταση να είναι κλειστή.

A. Αν το σύστημα ασφαλείας υλοποιείται με ένα συνδυαστικό κύκλωμα, καθορίστε τις εισόδους και τις εξόδους του και κατασκευάστε τον πίνακα αλήθειας.

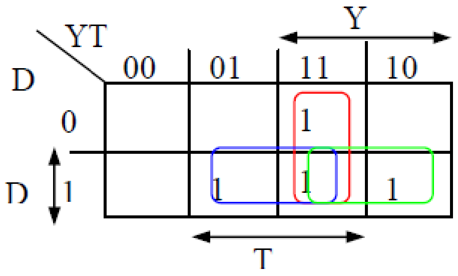
B. Απλοποιείστε το κύκλωμα χρησιμοποιώντας χάρτες Karnaugh και σχεδιάστε το με λογικές πύλες.

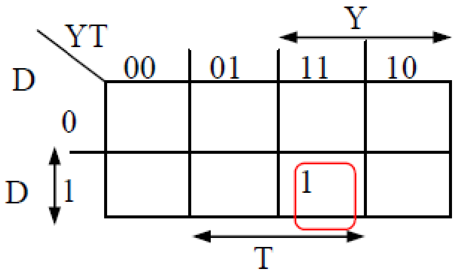
Παράδειγμα 2

A. Είναι προφανές ότι στο συνδυαστικό κύκλωμα οι είσοδοι αντιστοιχούν στα τρία κλειδιά που έχουν οι υπάλληλοι (D,Y,T) ενώ οι έξοδοι του είναι οι δύο πόρτες (X,S). Λαμβάνοντας υπόψη τα δεδομένα της εκφώνησης ο πίνακας αλήθειας συμπληρώνεται όπως φαίνεται παρακάτω:

D	Y	T	X	S
0	0	0	0	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	1	1

B. Χρειαζόμαστε 2 χάρτες Karnaugh των τριών μεταβλητών, έναν για κάθε έξοδο.

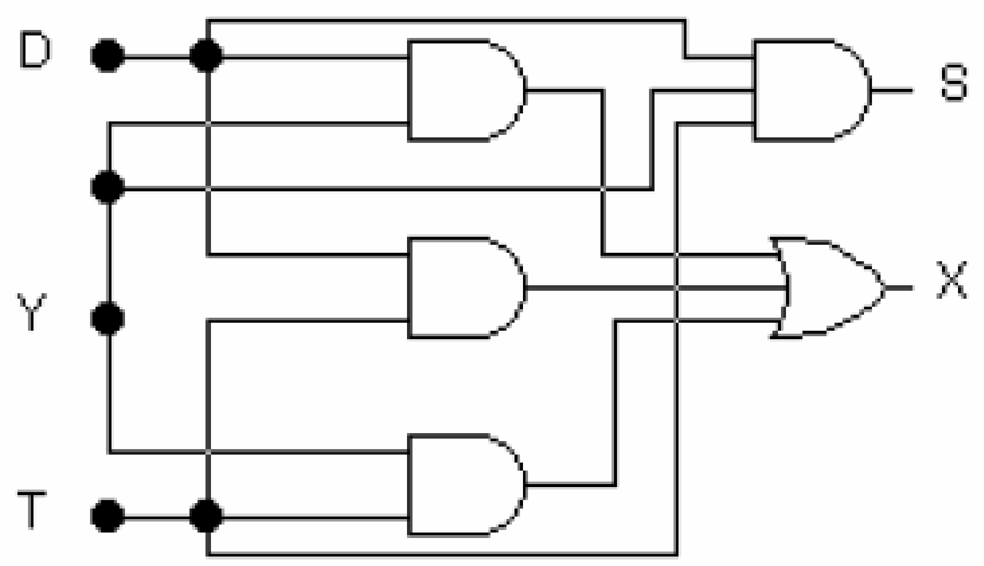




$X = YT + DT + DY$

$S = DYT$

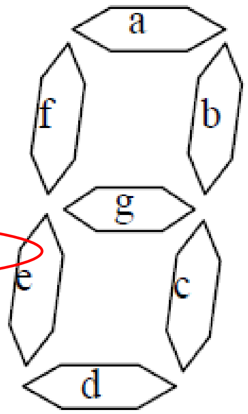
Παράδειγμα 2



Παράδειγμα 3

Να σχεδιάσετε έναν αποκωδικοποιητή «BCD σε απεικόνιση-7-τμημάτων» (BCD to 7-segment display). Χρησιμοποιήστε απλοποίηση με χάρτες Karnaugh για την υλοποίηση των 7 συναρτήσεων: *a, b, c, d, e, f, g*.

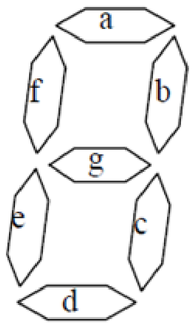
Τα 10 BCD ψηφία απεικονίζονται στο display ως εξής:
[0] = {a,b,c,d,e,f}
[1] = {b,c}
[2] = {a,b,d,e,g}
[3] = {a,b,c,d,g}
[4] = {b,c,f,g}
[5] = {a,c,d,f,g}
[6] = {a,c,d,e,f,g}
[7] = {a,b,c}
[8] = {a,b,c,d,e,f,g}
[9] = {a,b,c,d,f,g}
(Οι τιμές 10 έως 15 δεν ενδιαφέρει πώς θα απεικονίζονται.)



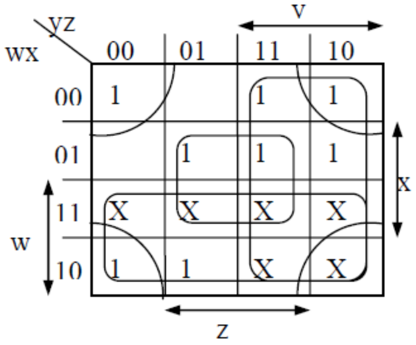
Παράδειγμα 3

Θεωρώντας ότι η BCD τιμή σχηματίζεται από τις μεταβλητές w, x, y, z (όπου w είναι το ΠΣΨ), συμπληρώνουμε τον πίνακα αλήθειας των 7 συναρτήσεων a, b, c, d, e, f, g:

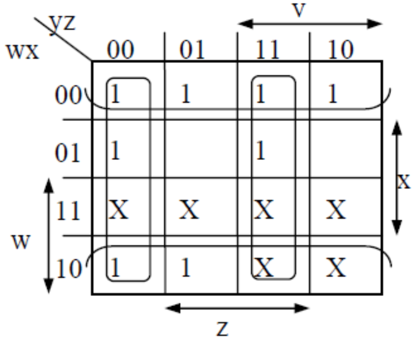
w	x	y	z	Αριθμός	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	X	1	X	A,B,E,F	X	X	X	X	X	X	X
1	1	X	X	C,D	X	X	X	X	X	X	X



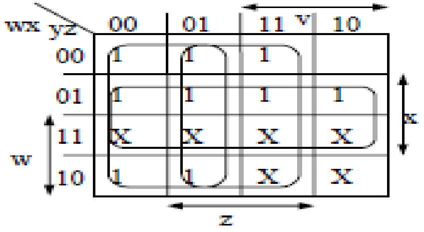
Παράδειγμα 3



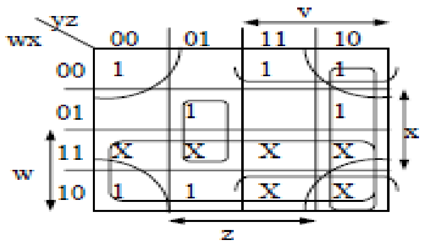
a = w + y + x·z + x’·z’



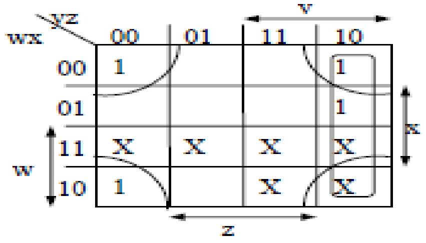
b = x’ + y·z + y’·z’



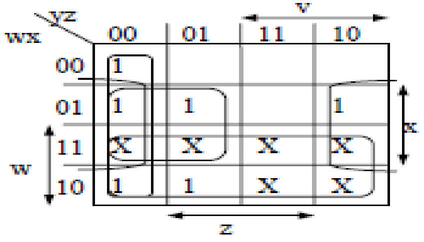
c = x + y’ + z



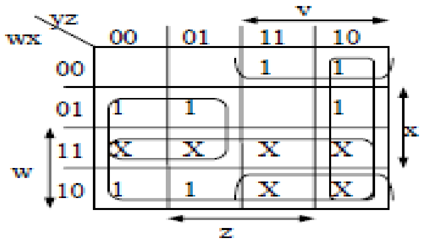
d = w + x·y’·z + x’·y + x’·z’ + y·z’



e = x’·z’+ y·z’



f = w + x·y’ + x·z’ + y’·z’

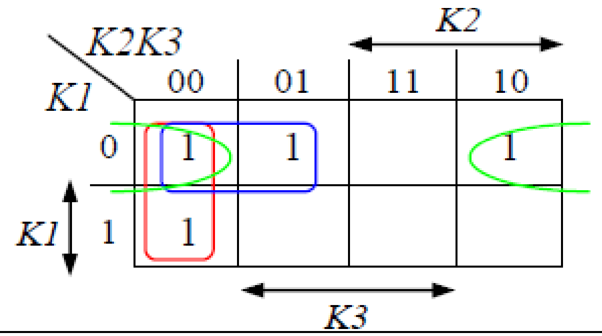


g = w + x·y’ + x’·y + y·z’

Παράδειγμα 4

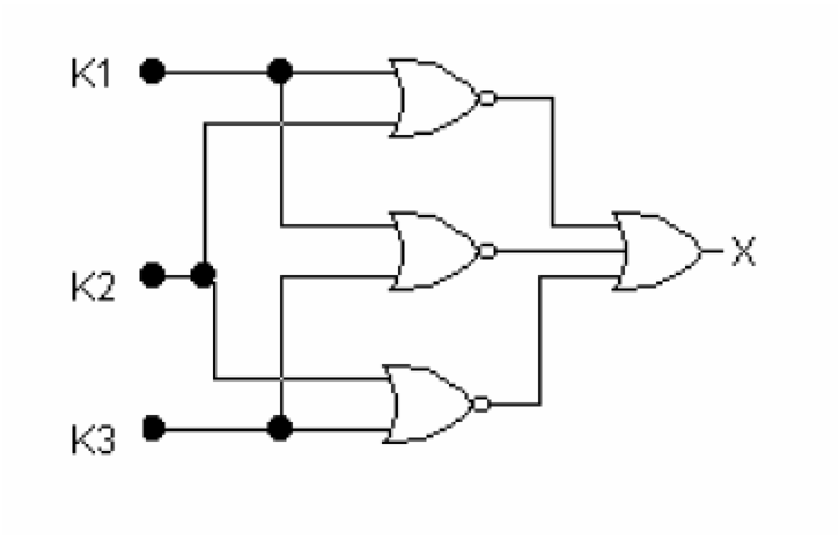
Στην άρση βαρών μια προσπάθεια ενός αθλητή θεωρείται έγκυρη όταν τη δέχονται τουλάχιστον δύο από τους τρεις κριτές (Κ1,Κ2,Κ3) και άκυρη διαφορετικά. Σχεδιάστε συνδυαστικό κύκλωμα, χρησιμοποιώντας μόνο πύλες NOR, το οποίο να δίνει έξοδο (X) ίση με λογικό '1' όταν μια προσπάθεια θεωρείται άκυρη.
Συμβολίστε με λογικό '1' την αποδοχή της προσπάθειας από έναν κριτή και με λογικό '0' το αντίθετο.

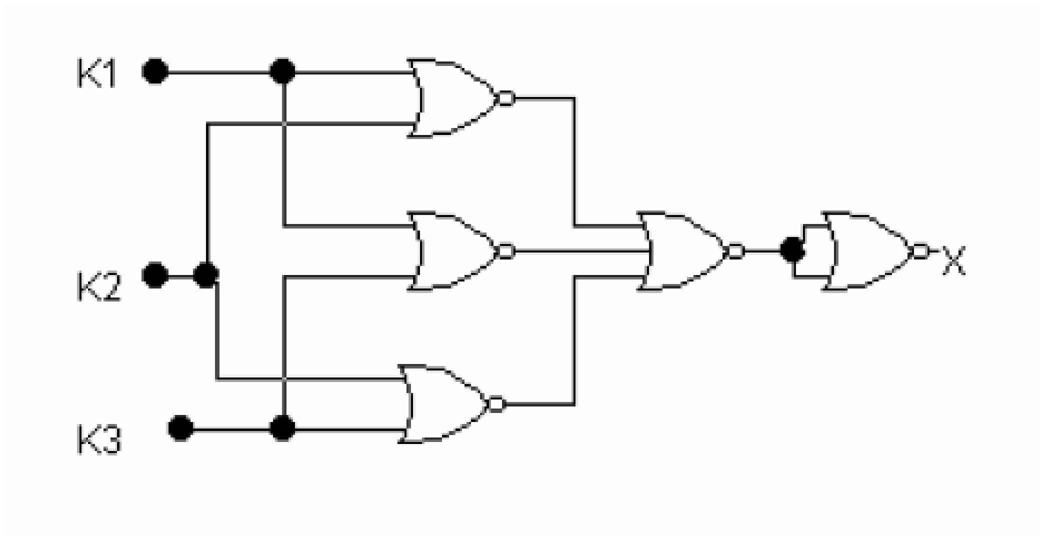
K1	K2	K3	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



X = K2'K3' + K1'K2' + K1'K3'

Παράδειγμα 4





- **Άσκηση: Να σχεδιάσετε το κύκλωμα χρησιμοποιώντας μόνο πύλες $(NAND)_2$**

Παράδειγμα 5

Ένα συνδυαστικό κύκλωμα δέχεται στην είσοδο ένα τετραψήφιο δυαδικό αριθμό ($X=X_3X_2X_1X_0$) και δίνει στην έξοδο έναν δυαδικό αριθμό (Y) ίσο με Y=X-10. Π.χ. για τον αριθμό $X=12_{10}=1100_2$, η έξοδος πρέπει να δίνει $Y=X-10=12-10=2_{10}$.

Όταν η έξοδος έχει αρνητική τιμή δεν μας ενδιαφέρει να εμφανίζεται η σωστή ένδειξη.

Α. Καθορίστε τις εισόδους και τις εξόδους του κυκλώματος και κατασκευάστε τον πίνακα αλήθειας.

Β. Απλοποιείστε τις συναρτήσεις εξόδου του κυκλώματος χρησιμοποιώντας χάρτες Karnaugh.

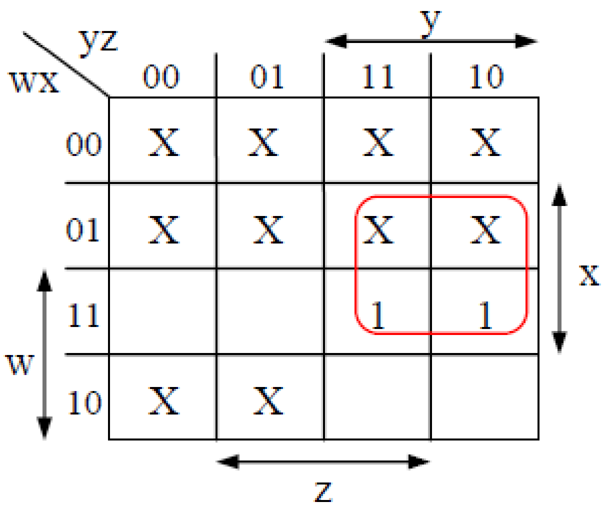
Γ. Υλοποιήστε το κύκλωμα χρησιμοποιώντας μόνο πύλες NAND.

Δ. Βρείτε τις απλοποιημένες συναρτήσεις εξόδου που προκύπτουν χωρίς την εκμετάλλευση των αδιάφορων όρων.

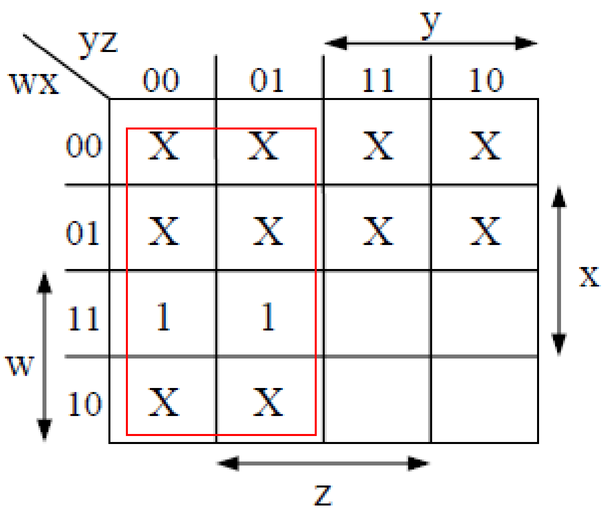
Παράδειγμα 5

Δυαδική Είσοδος				Δεκαδικοί Αριθμοί		Δυαδική Έξοδος		
X_3	X_2	X_1	X_0	X	Y	Y_2	Y_1	Y_0
0	0	0	0	0	-10	X	X	X
0	0	0	1	1	-9	X	X	X
0	0	1	0	2	-8	X	X	X
0	0	1	1	3	-7	X	X	X
0	1	0	0	4	-6	X	X	X
0	1	0	1	5	-5	X	X	X
0	1	1	0	6	-4	X	X	X
0	1	1	1	7	-3	X	X	X
1	0	0	0	8	-2	X	X	X
1	0	0	1	9	-1	X	X	X
1	0	1	0	10	0	0	0	0
1	0	1	1	11	1	0	0	1
1	1	0	0	12	2	0	1	0
1	1	0	1	13	3	0	1	1
1	1	1	0	14	4	1	0	0
1	1	1	1	15	5	1	0	1

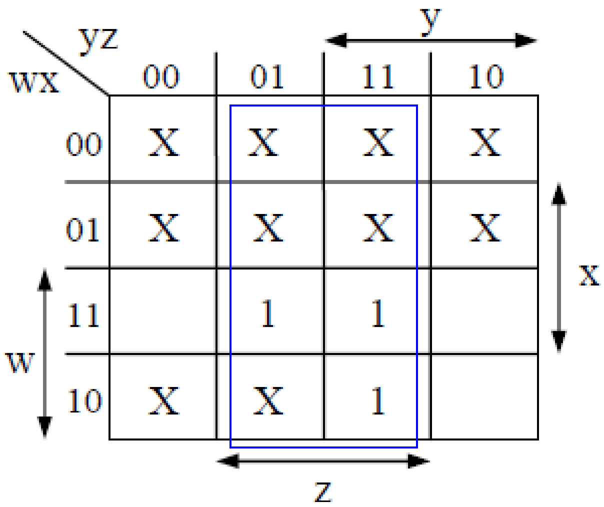
Παράδειγμα 5



$Y_2(w,x,y,z) = x\,y$



$Y_1(w,x,y,z) = y'$



$Y_0(w,x,y,z) = z$

Παράδειγμα 5

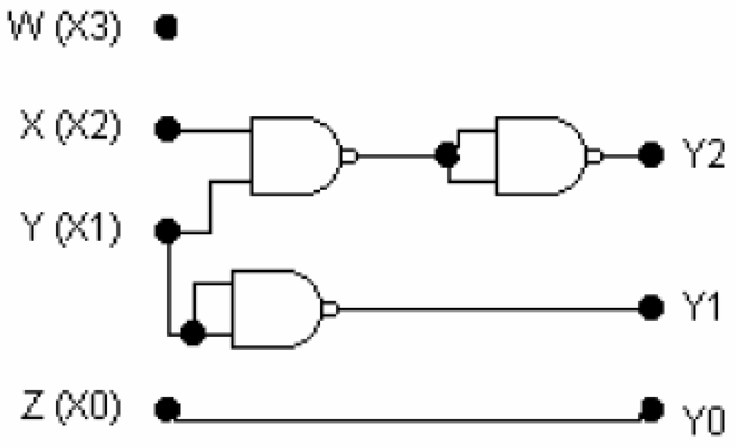
Γ. Για την υλοποίηση του κυκλώματος μόνο με πύλες NAND οι συναρτήσεις των εξόδων πρέπει να τροποποιηθούν ως εξής:

$Y_2 = xy = [(xy)']' = [(xy)'\,(xy)']'$

$Y_1 = y' = (yy)'$

$Y_0 = z$

Η υλοποίηση φαίνεται στο παρακάτω σχήμα:



Παράδειγμα 6

Κύκλωμα ‘πλειοψηφίας’ τεσσάρων εισόδων λέγεται το κύκλωμα το οποίο δέχεται στην είσοδο τέσσερα δυαδικά ψηφία (w,x,y,z) και δίνει στην έξοδο E λογικό ‘1’ όταν στις εισόδους το πλήθος των λογικών ‘1’ είναι μεγαλύτερο από το πλήθος των λογικών ‘0’.

A. Κατασκευάστε τον πίνακα αλήθειας του κυκλώματος.

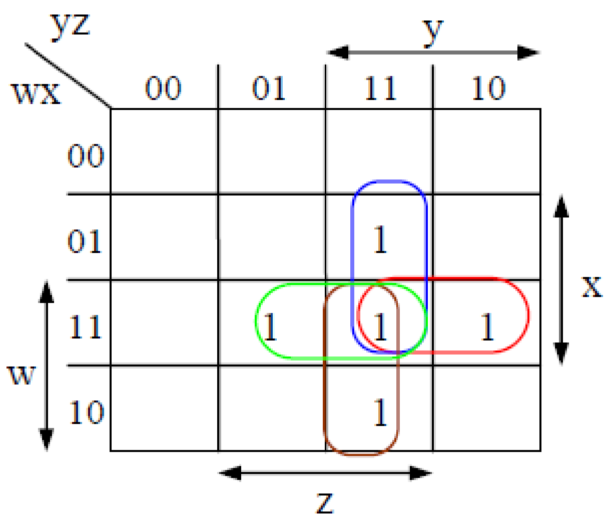
B. Απλοποιείστε τη συνάρτηση εξόδου του κυκλώματος χρησιμοποιώντας χάρτη Karnaugh.

Γ. Στην περίπτωση που δεν μας ενδιαφέρει το αποτέλεσμα της εξόδου όταν το πλήθος των λογικών ‘1’ είναι ίσο με το πλήθος των λογικών ‘0’, υλοποιήστε το κύκλωμα χρησιμοποιώντας μόνο πύλες NAND.

Παράδειγμα 6

Α. Το κύκλωμα αποτελείται από τέσσερις εισόδους (w,x,y,z) και μία έξοδο Ε. Ο πίνακας αλήθειας έχει 16 γραμμές και είναι ο ακόλουθος:

w	x	y	z	E
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
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	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



$$E = wxy + xyz + wxz + wyz$$

Παράδειγμα 6

Γ. Αφού δεν μας ενδιαφέρει το αποτέλεσμα της εξόδου όταν το πλήθος των λογικών ‘1’ είναι ίσο με το πλήθος των λογικών ‘0’ σε αυτές τις περιπτώσεις προκύπτουν αδιάφοροι όροι. Ο χάρτης Karnaugh των τεσσάρων μεταβλητών με τους αδιάφορους όρους φαίνεται παρακάτω. Οι γειτονιές ελαχιστόρων με την εκμετάλλευση των αδιάφορων όρων φαίνονται σε διάφορα χρώματα. Η συνάρτηση εξόδου και τα γινόμενα που δίνουν φαίνονται με αντίστοιχα χρώματα στο πλαίσιο.

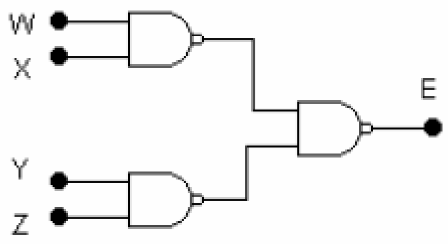
wx \ yz	00	01	11	10
00			X	
01		X	1	X
11	X	1	1	1
10		X	1	X

$$\mathbf{E} = wx + yz$$

Παράδειγμα 6

$E = wx + yz = [(wx + yz)']' = [(wx)'(yz)']'$

Η υλοποίηση φαίνεται στο παρακάτω σχήμα:



Παράδειγμα 7

Έστω ψηφιακό σύστημα τεσσάρων εισόδων x,y,z,w το οποίο δίνει στην μια και μοναδική έξοδο του "1" εαν ο αριθμός που σχηματίζεται είναι "πρώτος", διαφορετικά δίνει "0". Να σημειωθεί ότι "πρώτος" ονομάζεται ένας αριθμός που έχει δύο αποκλειστικά διαιρέτες: τον εαυτό του και τη μονάδα. (Σημείωση: οι αριθμοί 0 και 1 **δεν** είναι πρώτοι)

Α. Να συμπληρωθεί ο σχετικός πίνακας αληθείας

Β. Να συμπληρωθεί πίνακας Karnaugh και να γίνει εξαγωγή της συνάρτησης

Γ. Να σχεδιασθεί κύκλωμα χρησιμοποιώντας μόνο πύλες 2 εισόδων και αντιστροφείς.

Παράδειγμα 7

x	y	z	w	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
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

B.

xy \ zw	00	01	11	10
00			1	1
01		1	1	
11		1		
10			1	

F=yz'w+x'y'z+x'zw+y'zw

F=yz'w+x'y'z+x'zw+y'zw=w(y xor z)+x'z(y'+w)

Παράδειγμα 8

Θεωρείστε ότι σε έναν δυαδικό κώδικα είναι επιτρεπτές μόνο οι λέξεις οι οποίες δεν έχουν σε συνεχόμενες θέσεις ψηφία με τιμή ‘1’.

A) Κατασκευάστε τον πίνακα αληθείας λογικής συνάρτησης G η οποία λαμβάνει την τιμή 1 αν μια λέξη μήκους τεσσάρων δυαδικών ψηφίων είναι επιτρεπτή σύμφωνα με τον κώδικα αυτό και την τιμή 0 αν δεν είναι.

B) Απλοποιήστε τη συνάρτηση G και σχεδιάστε το αντίστοιχο λογικό κύκλωμα. Εξηγήστε.

Γ) Σχεδιάστε το λογικό κύκλωμα της συνάρτησης G χρησιμοποιώντας τον ελάχιστο δυνατό αριθμό πυλών. Εξηγήστε.

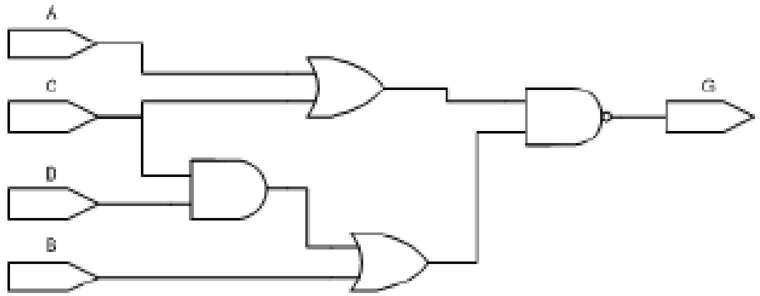
Παράδειγμα 8

	ABCD	G
0	0000	1
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	1
6	0110	0
7	0111	0
8	1000	1
9	1001	1
10	1010	1
11	1011	0
12	1100	0
13	1101	0
14	1110	0
15	1111	0

Γ) Η σχέση $G = A'C' + B'D' + B'C'$ χρησιμοποιώντας DeMorgan γράφεται ως

$$\begin{aligned} G &= A'C' + B'(D' + C') \\ &= A'C' + B'(DC)' \\ &= (A+C)' + (B+DC)' \\ &= ((A+C)(B+DC))' \end{aligned}$$

Η υλοποίηση αυτής της έκφρασης με πύλες, έχει ως εξής:



Παράδειγμα 9

Έστω κύκλωμα το οποίο δέχεται στην είσοδο τέσσερα δυαδικά ψηφία (w,x,y,z) και δίνει στην έξοδο E λογικό '1' όταν το πλήθος των **διαδοχικών** λογικών '0' των τεσσάρων δυαδικών ψηφίων είναι μεγαλύτερο ή ίσο από 2 ενώ δίνει στην έξοδο E λογικό '0' διαφορετικά.

A. Κατασκευάστε τον πίνακα αλήθειας του κυκλώματος.

B. Βρείτε την απλοποιημένη συνάρτηση εξόδου χρησιμοποιώντας χάρτη Karnaugh.

Γ. Τροποποιείστε την απλοποιημένη συνάρτηση εξόδου ώστε το κύκλωμα να υλοποιείται χρησιμοποιώντας το πολύ **4 πύλες** των δύο εισόδων η καθεμία (Δεν υπάρχει η δυνατότητα χρήσης πυλών NOT).

Δ. Στην περίπτωση που δεν μας ενδιαφέρει το αποτέλεσμα της εξόδου όταν το πλήθος των διαδοχικών λογικών '0' είναι ίσο με 2, προσπαθήστε να υλοποιήσετε το κύκλωμα χρησιμοποιώντας μόνο μία πύλη δύο εισόδων (Δεν υπάρχει η δυνατότητα χρήσης πυλών NOT).

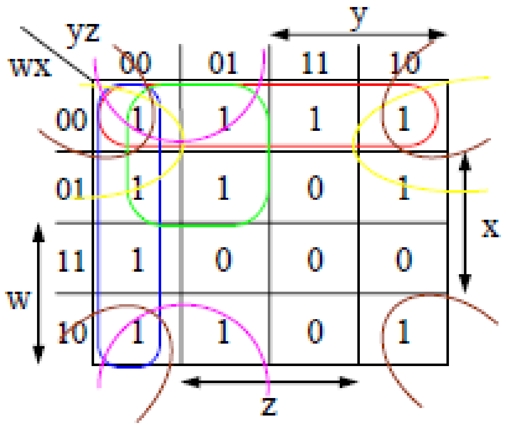
Παράδειγμα 9

Α. Το κύκλωμα αποτελείται από τέσσερις εισόδους (w,x,y,z) και μία έξοδο Ε. Ο πίνακας αλήθειας έχει 16 γραμμές και είναι ο ακόλουθος:

w	x	y	z	E
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Παράδειγμα 9

Β. Ο χάρτης Karnaugh των τεσσάρων μεταβλητών φαίνεται παρακάτω. Οι γειτονιές ελαχιστόρων φαίνονται σε διάφορα χρώματα. Η συνάρτηση εξόδου και τα γινόμενα που δίνουν φαίνονται με αντίστοιχα χρώματα στο πλαίσιο.

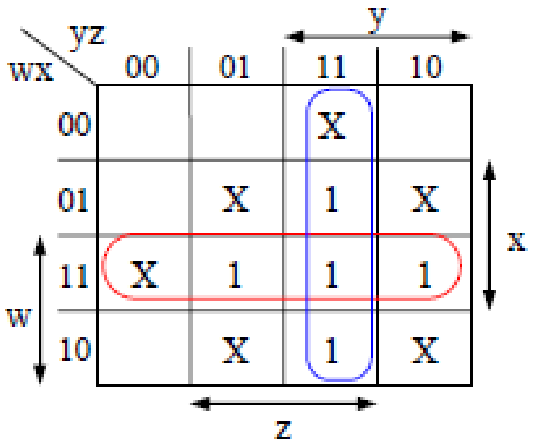


$$\mathbf{E} = w'x' + w'y' + w'z' + x'y' + x'z' + y'z'$$

Γ. Για την υλοποίηση του κυκλώματος με τον μικρότερο δυνατό αριθμό πυλών δύο εισόδων η συνάρτηση εξόδου μπορεί να τροποποιηθεί ως εξής:

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

Παράδειγμα 9

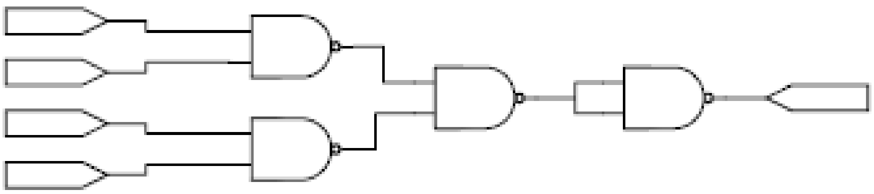


$$E' = wx + yz$$

Για την υλοποίηση του κυκλώματος μόνο με πύλες NAND η συνάρτηση εξόδου πρέπει να τροποποιηθεί ως εξής:

$$E = (E')' = (wx + yz)' = (wx)'(yz)' = [(wx)'(yz)']'' = [\,[(wx)'(yz)']' \cdot [(wx)'(yz)']'\,]'$$

Η υλοποίηση με τέσσερις πύλες NAND φαίνεται στο παρακάτω σχήμα:



Παράδειγμα 10

Έστω λογικό κύκλωμα στο οποίο είναι δυνατόν να εμφανιστούν ως είσοδοι **μόνο οι ακέραιοι από 1 έως και 9**, κωδικοποιημένοι κατά 8421. Το κύκλωμα αυτό παράγει μια έξοδο F η οποία είναι 1 αν ο ακέραιος αριθμός της εισόδου διαιρείται ακριβώς με το 3, και μηδέν διαφορετικά. Να υλοποιηθεί το κύκλωμα με 4 πύλες των 2 εισόδων η καθεμία.

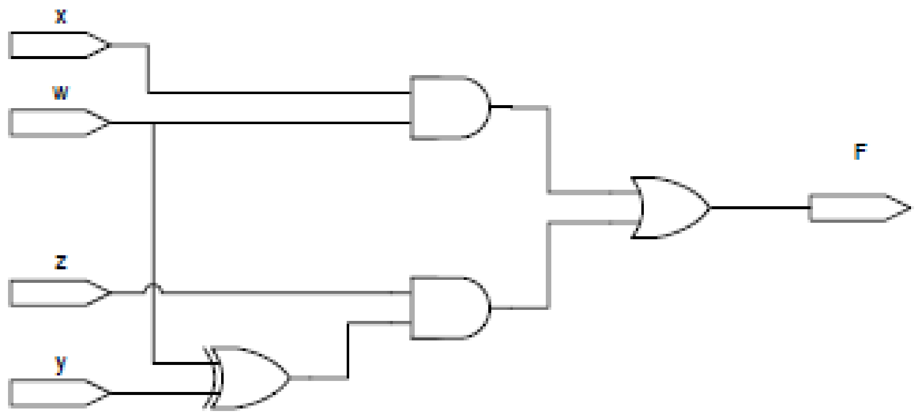
xyzw	F
0000	X
0001	0
0010	0
0011	1
0100	0
0101	0
0110	1
0111	0
1000	0
1001	1
1010	X
1011	X
1100	X
1101	X
1110	X
1111	X

Στη συνέχεια με βάση τον πίνακα αληθείας κατασκευάζουμε το χάρτη Karnaugh

xy \ zw	00	01	11	10
00	X		1	
01				1
11	X	X	X	X
10		1	X	X

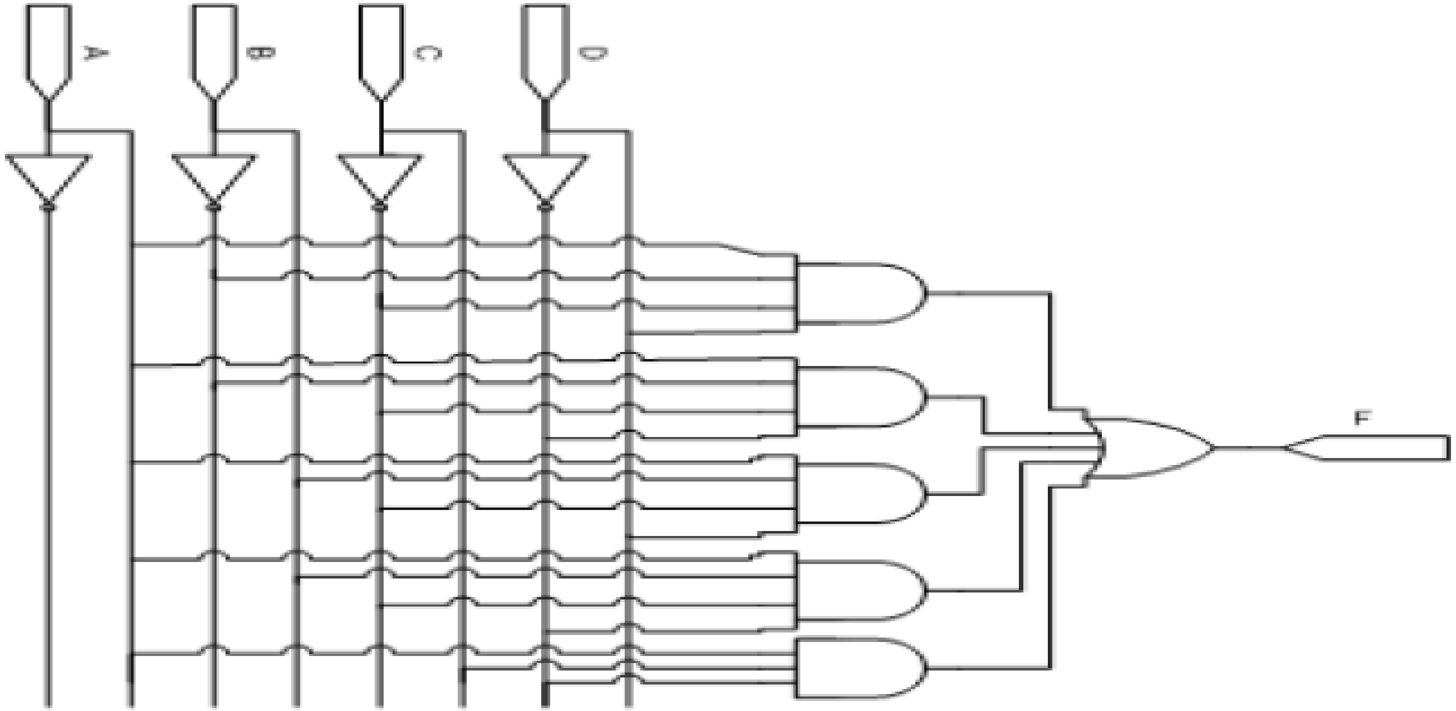
Παράδειγμα 10

από τον οποίο προκύπτει ότι η συνάρτηση F απλοποιείται ως $F = xw + yzw' + y'zw = xw + z(yw'+y'w) = x\,w + z\,(y \oplus w)$. Η σχετική υλοποίηση έχει ως εξής:



Παράδειγμα 11

Να βρεθεί η απλοποιημένη λογική συνάρτηση του παρακάτω κυκλώματος και να υλοποιηθεί με τη χρήση **ΜΟΝΟ τριών (3)** πυλών NAND 2 εισόδων.



Παράδειγμα 11

$$F = AB'C'D + AB'C'D' + ABC'D + ABC'D' + ACD'$$

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

Από τη παραπάνω ομαδοποίηση προκύπτει η λογική εξίσωση:

$$F = AC' + AD'$$

Βγάζοντας κοινό παράγοντα το A προκύπτει η παρακάτω μορφή:

$$F = (AC' + AD') = A\,(C' + D') = A\,(C\,D)' = (A\,(C\,D)')''$$

και τελικά προκύπτει το παρακάτω κύκλωμα με χρήση μόνο 3 πυλών NAND 2 εισόδων:

Παράδειγμα 12

Α. Να συμπληρώσετε πίνακα, η αριστερή στήλη του οποίου να περιέχει όλους τους συνδυασμούς ($C_3C_2C_1C_0$, όπου C_3 το ΠΣΨ) σε μορφή «συμπληρώματος προς 2 (2's complement)», ενώ η δεξιά στήλη τους αντίστοιχους αριθμούς του δεκαδικού συστήματος.

Β. Χρησιμοποιώντας τον ελάχιστο δυνατό αριθμό λογικών πυλών, να σχεδιάσετε κύκλωμα το οποίο να μετατρέπει προσημασμένους αριθμούς των 4 bits ($C_3C_2C_1C_0$, όπου C_3 το ΠΣΨ) από μορφή «συμπληρώματος ως προς 2 (2's complement)» σε μορφή «προσημασμένου μέτρου (signed magnitude)».

Παράδειγμα 12

A

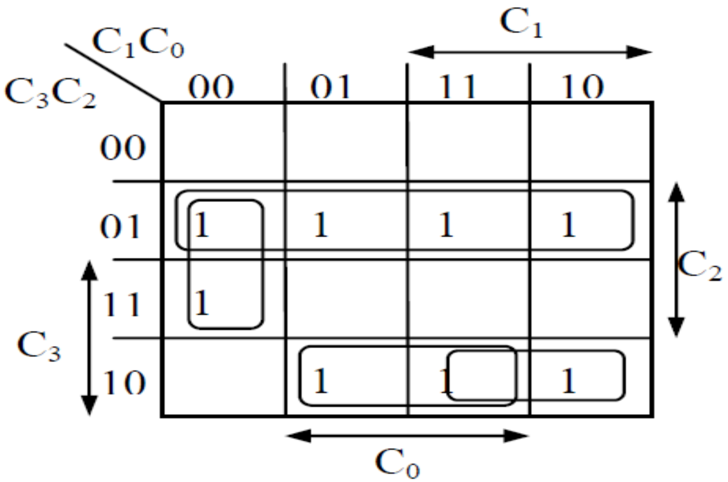
Συμπλήρωμα ως προς 2				Αριθμός Δεκαδικού Συστήματος
C_3	C_2	C_1	C_0	
0	0	0	0	0
0	0	0	1	+1
0	0	1	0	+2
0	0	1	1	+3
0	1	0	0	+4
0	1	0	1	+5
0	1	1	0	+6
0	1	1	1	+7
1	0	0	0	-8
1	0	0	1	-7
1	0	1	0	-6
1	0	1	1	-5
1	1	0	0	-4
1	1	0	1	-3
1	1	1	0	-2
1	1	1	1	-1

B

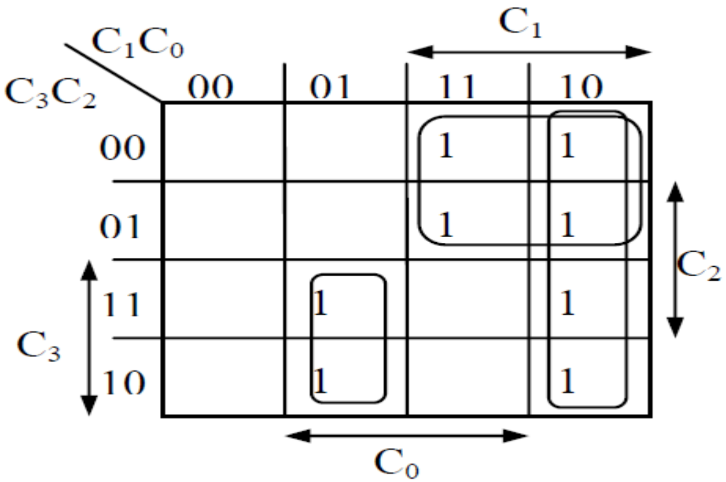
Συμπλήρωμα ως προς 2				Αριθμός Δεκαδικού Συστήματος	Προσημασμένο Μέτρο				
C_3	C_2	C_1	C_0		M_4	M_3	M_2	M_1	M_0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	+1	0	0	0	0	1
0	0	1	0	+2	0	0	0	1	0
0	0	1	1	+3	0	0	0	1	1
0	1	0	0	+4	0	0	1	0	0
0	1	0	1	+5	0	0	1	0	1
0	1	1	0	+6	0	0	1	1	0
0	1	1	1	+7	0	0	1	1	1
1	0	0	0	-8	1	1	0	0	0
1	0	0	1	-7	1	0	1	1	1
1	0	1	0	-6	1	0	1	1	0
1	0	1	1	-5	1	0	1	0	1
1	1	0	0	-4	1	0	1	0	0
1	1	0	1	-3	1	0	0	1	1
1	1	1	0	-2	1	0	0	1	0
1	1	1	1	-1	1	0	0	0	1

Παράδειγμα 12

$M_4=C_3$, $M_3=C_3C_2'C_1'C_0'= C_3(C_0+C_1+C_2)'$ και $M_0=C_0$.



M_2 $=C_2C_3'+C_0'C_1'C_2+C_0C_2'C_3+C_1C_2'$

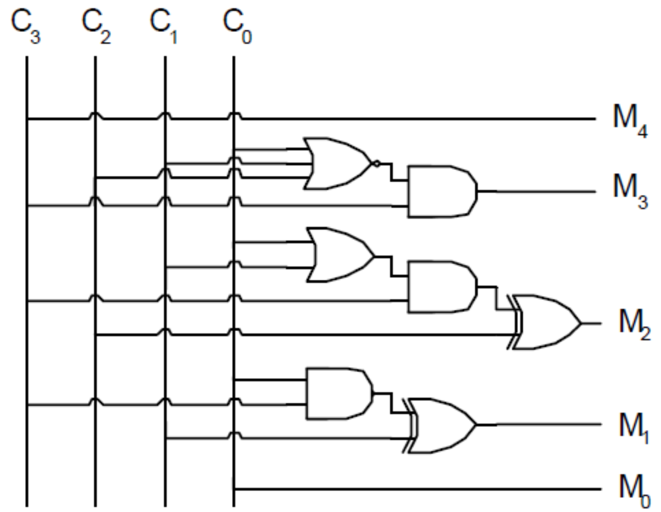


M_1 $= C_0'C_1+C_1C_3'+C_0C_1'C_3$

Παράδειγμα 12

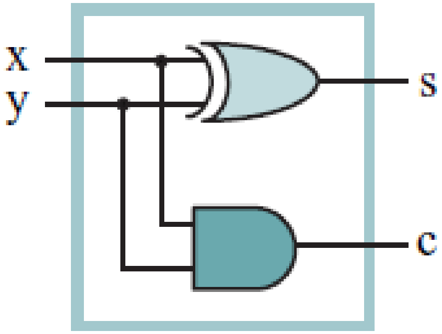
$$M_2 = C_2C_3'+C_0'C_1'C_2+C_0C_2'C_3+C_1C_2'C_3 = C_2(C_3'+C_0'C_1') + C_2'C_3(C_0+C_1) =$$
$$= C_2[C_3(C_0+C_1)]' + C_2'[(C_0+C_1)C_3] = C_2 \oplus [C_3(C_0+C_1)].$$

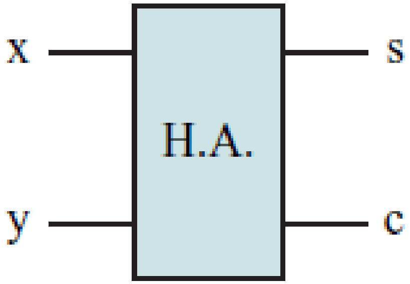
$$M_1 = C_0'C_1+C_1C_3'+C_0C_1'C_3 = C_1(C_0'+C_3') + C_1'C_0C_3 = C_1(C_0C_3)' + C_1'(C_0C_3) = C_1 \oplus (C_0C_3).$$



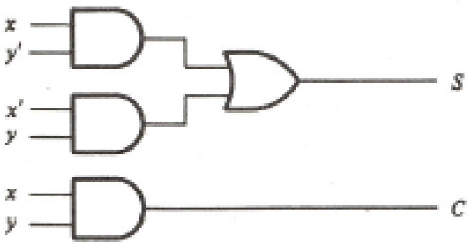
Ημιαθροιστής

x y	s	c
0 0	0	0
0 1	1	0
1 0	1	0
1 1	0	1

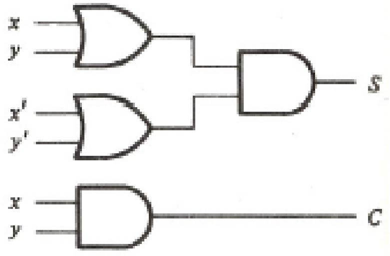




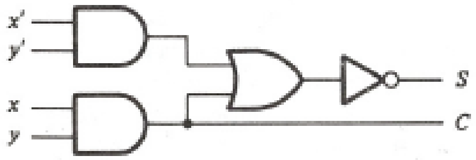
Υλοποιήσεις του ημιαθροιστή



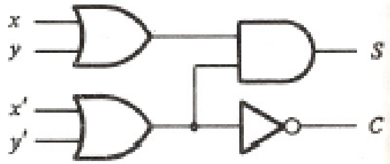
(α) $S = xy' + x'y$
$C = xy$



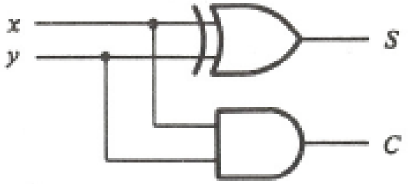
(β) $S = (x + y)(x' + y')$
$C = xy$



(γ) $S = (C + x'y')'$
$C = xy$



(δ) $S = (x + y)(x' + y')$
$C = (x' + y')'$



(ε) $S = x \oplus y$
$C = xy$

- ✓ Η (α) είναι η απευθείας υλοποίηση των συναρτήσεων $S=xy'+x'y$ και $C=xy$ με πύλες OR και AND.
- ✓ Η (β) προκύπτει από την έκφραση του S ως γινόμενο αθροισμάτων.
- ✓ Η (γ) προκύπτει από το γεγονός πως $(C+x'y')'=(xy+x'y')'=(x'+y')(x+y)=xy'+x'y$ $=S$
- ✓ Η (δ) προκύπτει από το ότι $C=(xy)''=(x'+y')'$
- ✓ Η (ε) προκύπτει από τον πίνακα αληθείας της πύλης XOR, δηλαδή το ότι $x\oplus y=xy'+x'y$

Αθροιστής

x_i	y_i	c_{i-1}	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

c_{i-1} \ $x_i\,y_i$	00	01	11	10
0		1		1
1	1		1	

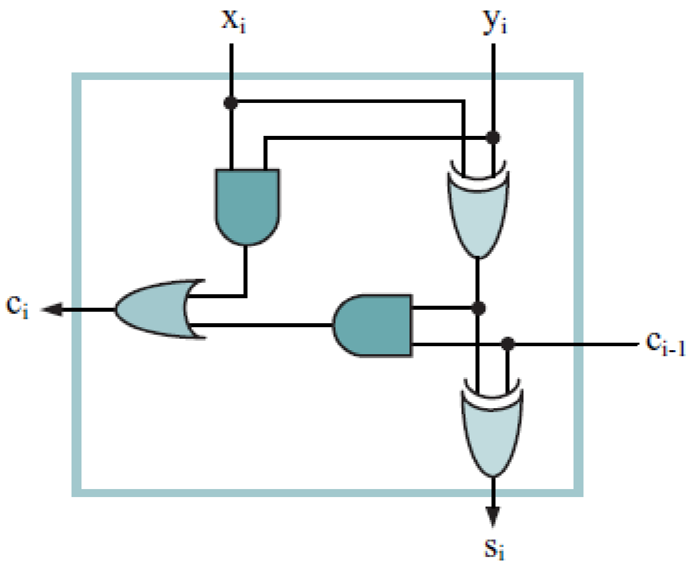
$s_i = x_i \oplus y_i \oplus c_{i-1}$

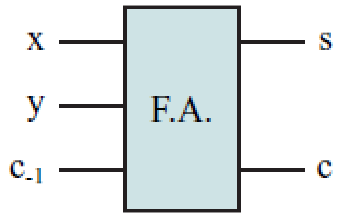
Άθροισμα

c_{i-1} \ $x_i\,y_i$	00	01	11	10
0			1	
1		1	1	1

$c_i = x_i \cdot y_i + c_{i-1} \cdot (x_i \oplus y_i)$

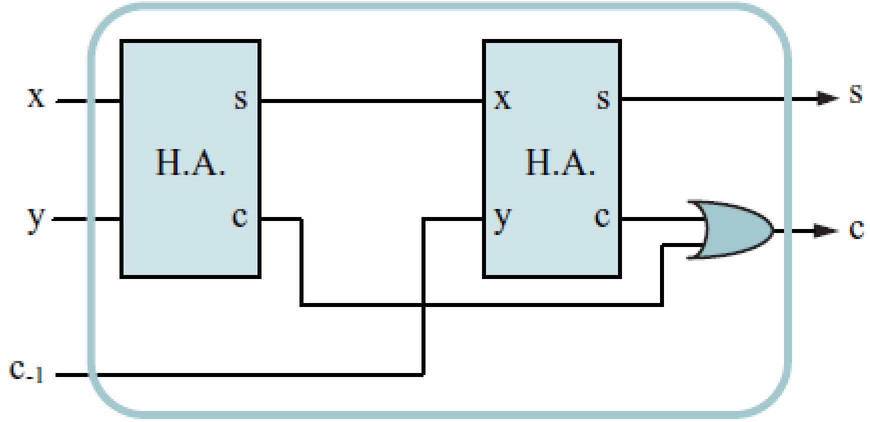
Κρατούμενο



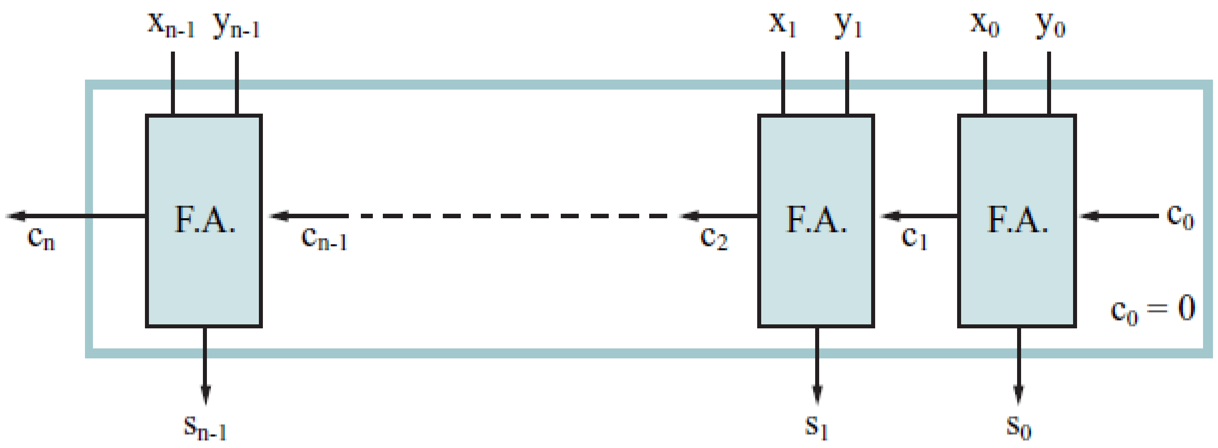


Αθροιστής

Ένας πλήρης αθροιστής κατασκευάζεται από δύο ημιαθροιστές (H.A.) και μία πύλη OR ως εξής:



Παράλληλος Αθροιστής



Παράλληλος αθροιστής n bit

- Ο πλήρης αθροιστής προσθέτει ταυτόχρονα τρία bits. Ένας απλός τρόπος κατασκευής ενός κυκλώματος για την ταυτόχρονη (παράλληλη) πρόσθεση δυαδικών αριθμών των n bits (n>1) είναι να συνδέσουμε διαδοχικά n μονάδες πλήρων αθροιστών

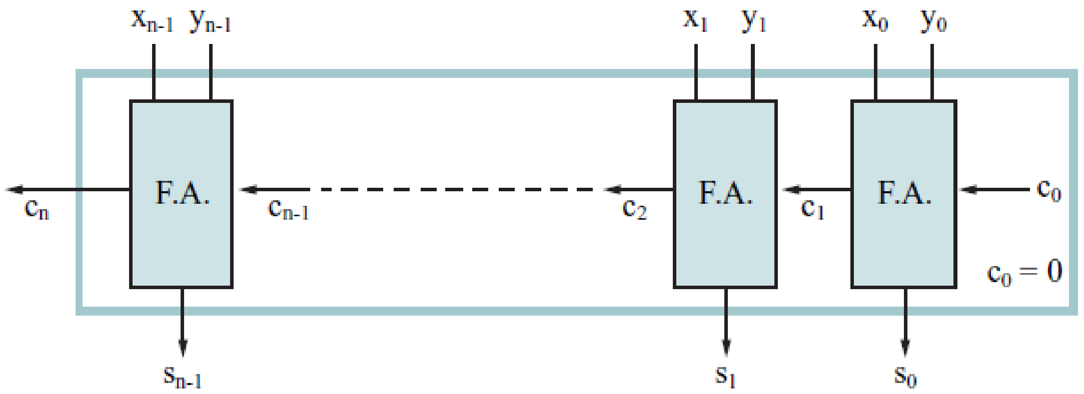
Πρόσθεση Αριθμών με n bits

- ✓ Για να προσθέσουμε αριθμούς με n bits μπορούμε να χρησιμοποιήσουμε πολλά τέτοια κυκλώματα.
- ✓ Για παράδειγμα αν έχουμε να προσθέσουμε τον A=1011 και τον B=0011 μπορούμε να πραγματοποιήσουμε την πρόσθεση σε τέσσερα στάδια ως εξής:

Δείκτης *i*	4	3	2	1
Κρατούμενο Εισόδου	0	1	1	0
Προσθετέος 1	1	0	1	1
Προσθετέος 2	0	0	1	1
Άθροισμα	1	1	1	0
Κρατούμενο Εξόδου	0	0	1	1

- ✓ Ο **σειριακός** αθροιστής χρησιμοποιεί μόνο ένα κύκλωμα πλήρους αθροιστή και χρησιμοποιεί κυκλώματα μνήμης για να αποθηκεύει τα ενδιάμεσα αποτελέσματα.
- ✓ Ο **παράλληλος αθροιστής** χρησιμοποιεί n κυκλώματα πλήρους αθροιστή για να παράγει το αποτέλεσμα.

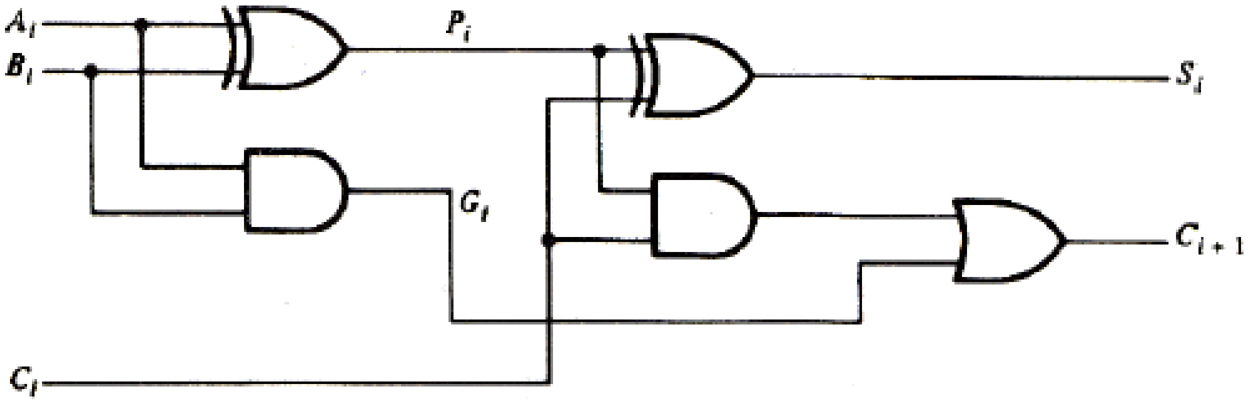
Κυματικός Αθροιστής



Εάν στον αθροιστή του Σχήματος τα bits x_i και y_i δημιουργήσουν ένα κρατούμενο c_i, τότε αυτό ενδέχεται να επηρεάσει τα επόμενα αθροίσματα μέχρι τις τελικές εξόδους s_{n-1} και c_n. Επομένως πρέπει να περιμένουμε λίγο χρόνο, μέχρι να διαδοθεί το κρατούμενο, για να πάρουμε το σωστό τελικό άθροισμα. Επειδή δεν γνωρίζουμε πότε και πού δημιουργείται κρατούμενο, είμαστε υποχρεωμένοι να περιμένουμε τη χειρότερη καθυστέρηση, δηλ. όταν το κρατούμενο πηγάζει από τα bits x_0, y_0, η οποία είναι n–πλάσια της καθυστέρησης ενός πλήρη αθροιστή. Η κυματομορφή του κρατουμένου, καθώς διαδίδεται διαδοχικά από τη μία μονάδα στην επόμενη, μοιάζει με ένα κυματάκι που εξαπλώνεται και γι' αυτό ο τύπος αυτός του παράλληλου αθροιστή ονομάζεται *κυματικός*.

Διάδοση Κρατουμένου

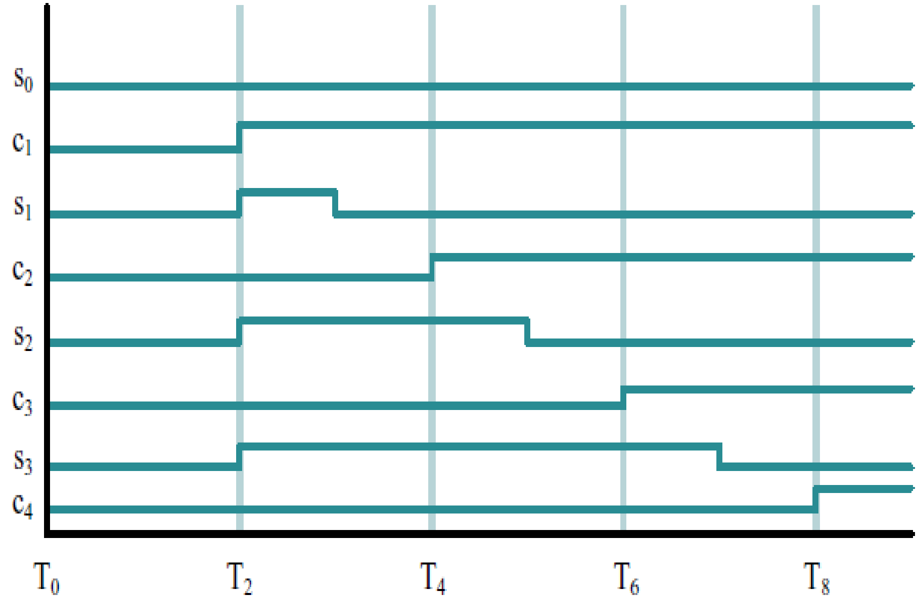
- ✓ Στο κύκλωμα του παράλληλου αθροιστή, χρειάζεται κάποιος χρόνος ώστε το κρατούμενο να διαδοθεί από τον έναν αθροιστή στον άλλο.
- ✓ Στην τελευταία βαθμίδα για παράδειγμα του προηγούμενου σχήματος, το κρατούμενο C_4 λαμβάνει την σωστή του τιμή, μόνο όταν ο 3ος πλήρης αθροιστής παράγει το σωστό αποτέλεσμα.
- ✓ Από το κύκλωμα του πλήρους αθροιστή καταλαβαίνουμε πως η κάθε βαθμίδα εισάγει καθυστέρηση δύο πυλών



- Χρόνος Διάδοσης (Καθυστέρηση): Επίπεδα Πυλών × Καθυστέρηση Πύλης
- Παράλληλος Αθροιστής με διάδοση κρατουμένου : Η μεγαλύτερη καθυστέρηση οφείλεται στη διάδοση του κρατουμένου.

Διάδοση Κρατουμένου

- Έστω ότι προσθέτουμε τους δυαδικούς αριθμούς 1111 και 0001 σε έναν παράλληλο αθροιστή n bits.
- Εάν θεωρήσουμε ότι κάθε λογική πύλη καθυστερεί το αποτέλεσμά της κατά μία μονάδα χρόνου, τότε τα αποτελέσματα του αθροίσματος s και του κρατούμενου c σε ένα πλήρη αθροιστή 1 bit καθυστερούν δύο μονάδες χρόνου.
- Στο διπλανό σχήμα φαίνονται οι κυματομορφές των αντίστοιχων bit του αθροίσματος και του κρατούμενου που σχηματίζονται στον παράλληλο αθροιστή στις διάφορες χρονικές στιγμές.



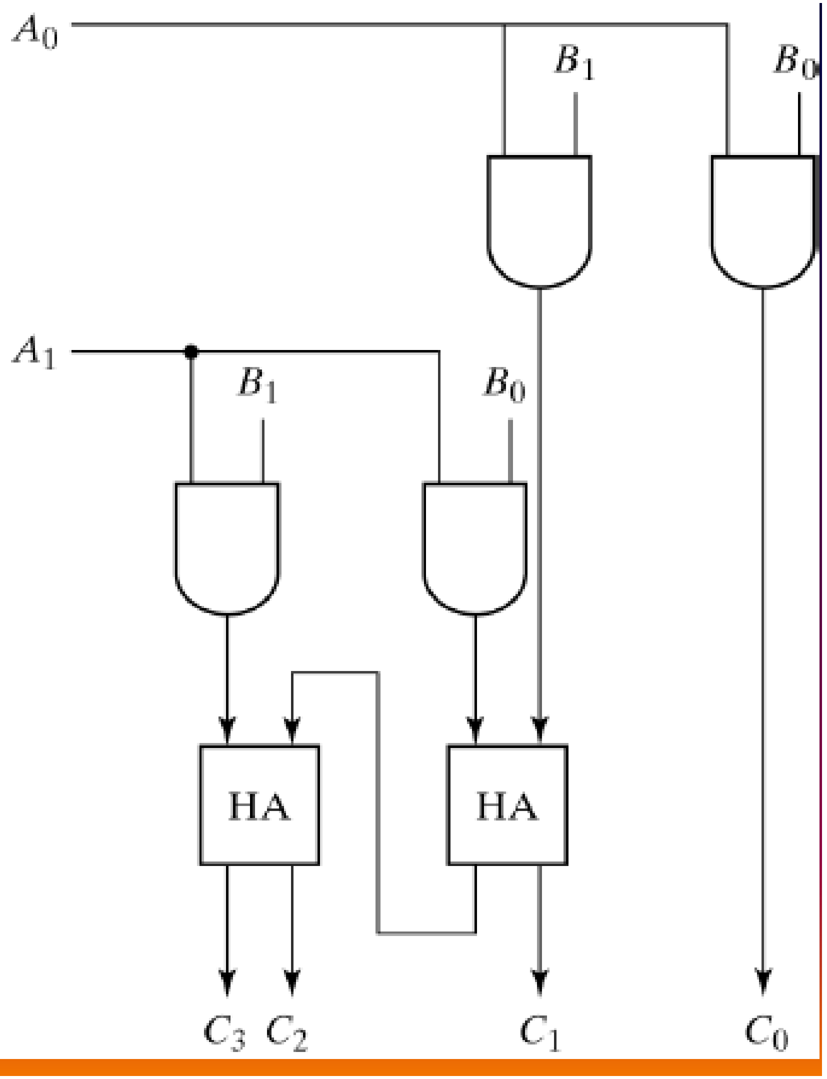
Συμπέρασμα

- *Ο ημιαθροιστής προσθέτει ένα ζεύγος ψηφίων x και y και παράγει το άθροισμα s και το κρατούμενο c.*
- *Ο πλήρης αθροιστής προσθέτει τρία bit, τα x, y και c_{-1}, και παράγει το άθροισμα s και το κρατούμενο c.*
- *Για την πρόσθεση αριθμών με n bits χρησιμοποιείται ο παράλληλος αθροιστής. Ο απλός παράλληλος αθροιστής (κυματικός) έχει το μειονέκτημα ότι καθυστερεί κατά χρόνο n–πλάσιο της πρόσθεσης ενός ζεύγους bit.*
- *Ο παράλληλος αθροιστής προσθέτει n bits ταυτόχρονα και αποτελείται από μια αλυσίδα πλήρων αθροιστών.*

Ο κυματικός αθροιστής είναι ένας παράλληλος αθροιστής που, λόγω της εσωτερικής σύνδεσής του, παρουσιάζει καθυστέρηση ανάλογη του πλήθους των bit.

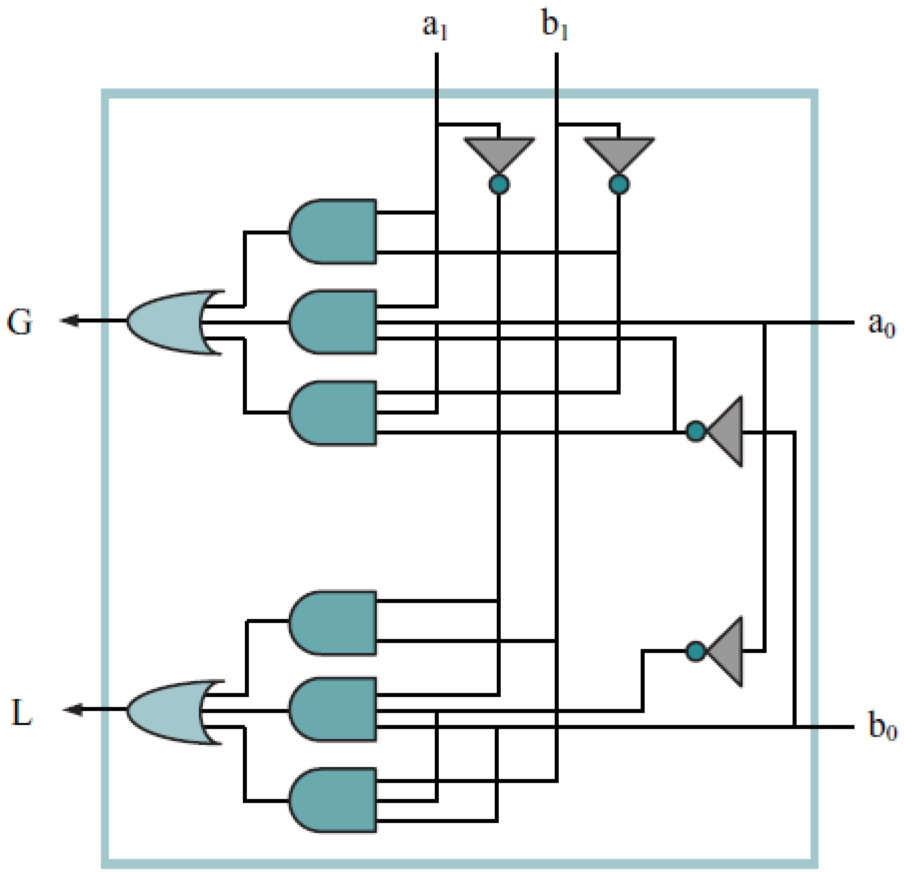
Δυαδικός Πολλαπλασιαστής

		B_1	B_0
		A_1	A_0
		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	
C_3	C_2	C_1	C_0



Συγκριτές

a_1	b_1	a_0	b_0	G	L
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	0



G L	Σχέση X,Y
0 0	X=Y
0 1	X<Y
1 0	X>Y

Συγκριτές

Συγκριτής Μεγέθους: συγκρίνει δύο αριθμούς και βρίσκει την σχέση τους (<,>,=).

Για δύο αριθμούς των n bits έχουμε 2^{2n} συνδυασμούς (πολύ μεγάλο).

Το κύκλωμα του συγκριτή έχει αρκετή κανονικότητα (αλγοριθμικός σχεδιασμός)

<u>Ισότητα</u>

$A=A_3A_2A_1A_0$
$B=B_3B_2B_1B_0$

$x_i=A_iB_i+A_i'B_i'$

$(A=B)=x_3x_2x_1x_0$

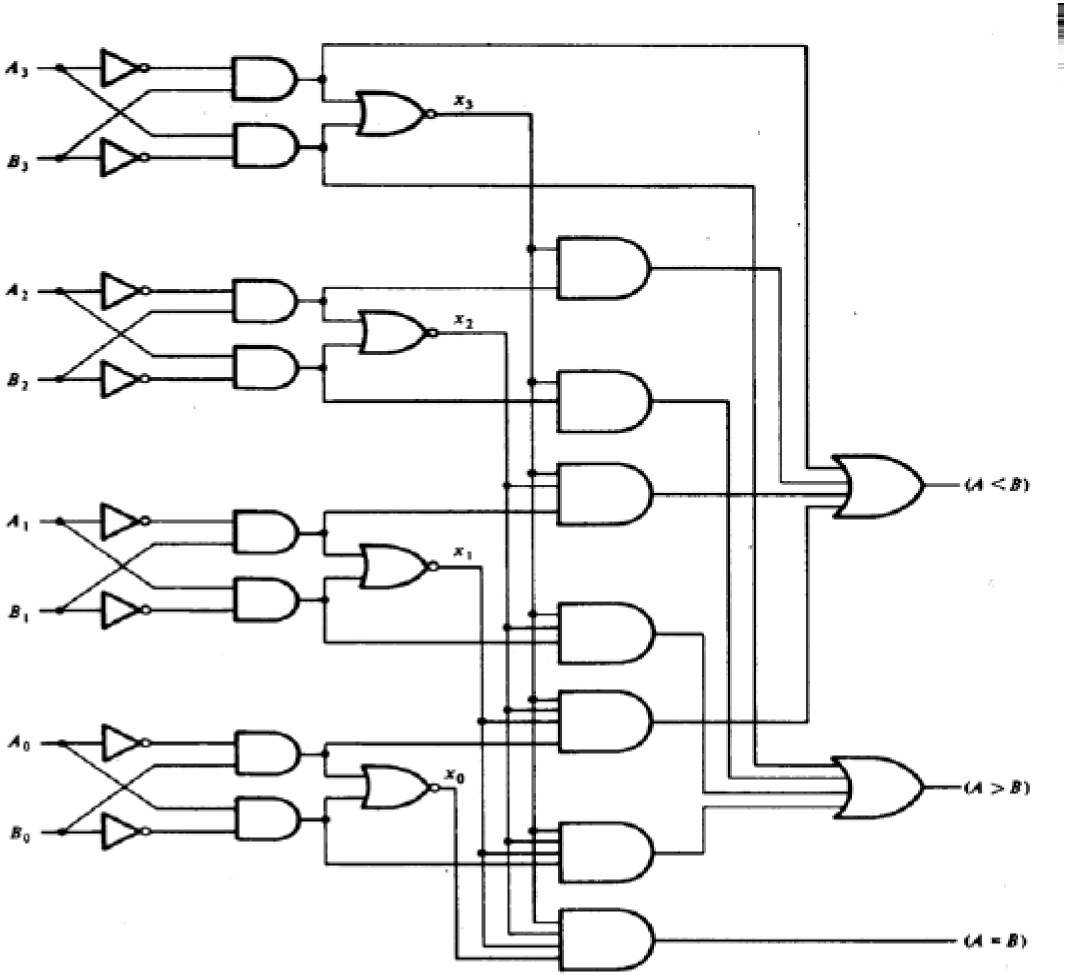
<u>Ανισότητα</u>

$(A>B)=A_3B_3'+x_3A_2B_2'+x_3x_2A_1B_1'+x_3x_2x_1A_0B_0'$: δίνει 1 μόνο όταν $A_i=1$ και $B_i=0$ με $A_j=B_j$ για $j>I$ δηλαδή ελέγχει από αριστερά προς δεξιά ένα ένα τα bits.

$(A<B)=B_3A_3'+x_3B_2A_2'+x_3x_2B_1A_1'+x_3x_2x_1B_0A_0'$

Πλεονέκτημα : Η κανονικότητα του κυκλώματος

Συγκριτές



Ημιαφαιρέτης

Το κύκλωμα που πραγματοποιεί την αφαίρεση των ψηφίων χωρίς να υπολογίζει τυχόν προηγούμενο δανεικό ονομάζεται **Ημιαφαιρέτης**. Ο Ημιαφαιρέτης έχει δυο εισόδους x και y (τα bit που αφαιρούνται) και δυο εξόδους B (δανεικό) και D (διαφορά).

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$D=x\oplus y$

$B=x'y$

Πλήρης Αφαιρέτης

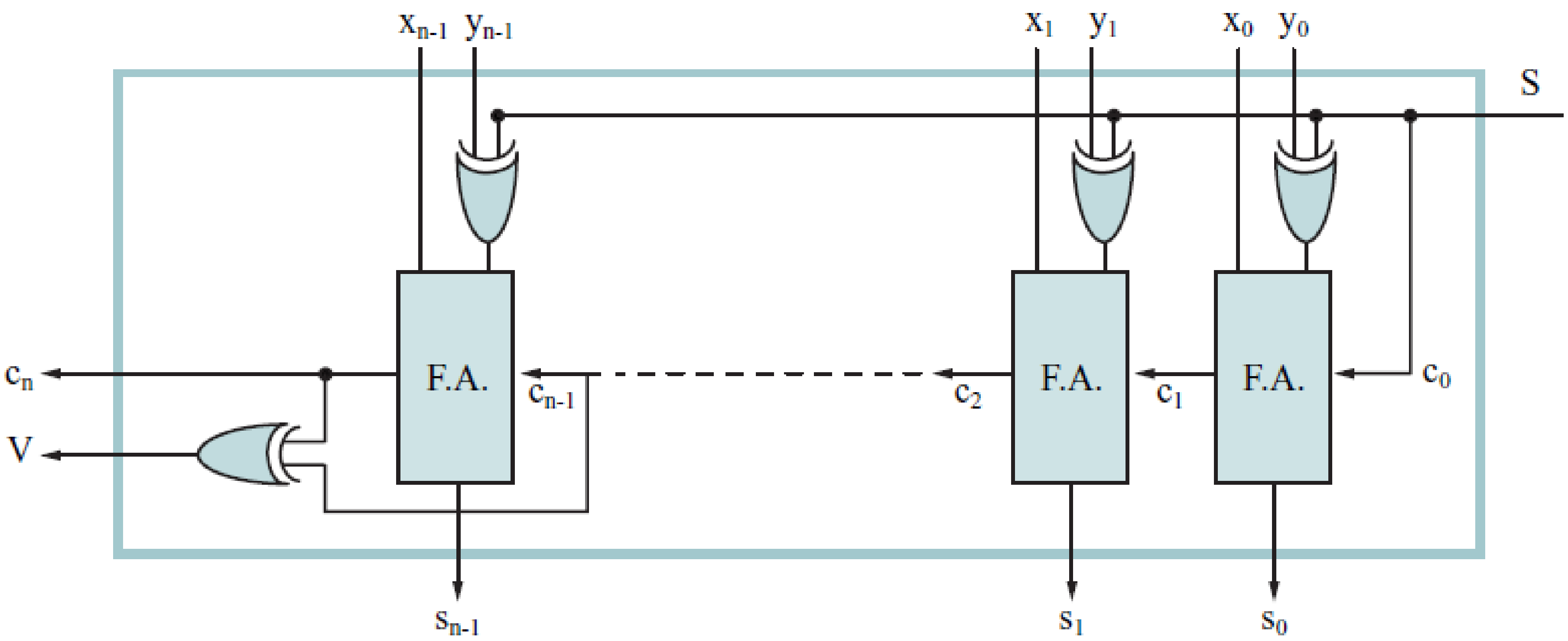
Το κύκλωμα που πραγματοποιεί την αφαίρεση δυο ψηφίων λαμβάνοντας υπόψη τυχόν προηγούμενο δανεικό ονομάζεται **Πλήρης Αφαιρέτης**. Ο Πλήρης Αφαιρέτης έχει τρεις εισόδους x, y (τα bit που προστίθενται) και z (δανεικό εισόδου) και δυο εξόδους B (δανεικό εξόδου) και D (διαφορά).

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\mathbf{D=(x \oplus y) \oplus z}$$

$$\mathbf{B=x'y+(x \oplus y)'z}$$

Αθροιστής/Αφαιρέτης



$$
\begin{array}{lcccc}
 & x_{n-1} & \cdots & x_1 & x_0 \\
 & y'_{n-1} & \cdots & y'_1 & y'_0 \\
+ & & & & 1 \\
\hline
 & z_{n-1} & \cdots & z_1 & z_0
\end{array}
$$

ΜΕΣΩ
XOR

Αθροιστής/Αφαιρέτης

- Ο αντίθετος ενός αριθμού Y = $y_{n-1} \ldots y_0$ στο Συμπλήρωμα ως προς 2 υπολογίζεται εάν (α) αντιστρέψουμε όλα τα bit του Y και (β) στο αποτέλεσμα προσθέσουμε το 1. Συνολικά, η αφαίρεση Z = X–Y, όπου X = $x_{n-1} \ldots x_0$, συντομεύεται εάν γίνει η πρόσθεση:

$$
\begin{array}{rcccc}
 & x_{n-1} & \cdots & x_1 & x_0 \\
 & y'_{n-1} & \cdots & y'_1 & y'_0 \\
+ & & & & 1 \\
\hline
 & z_{n-1} & \cdots & z_1 & z_0
\end{array}
$$

- Η αντιστροφή των bits γίνεται πολύ εύκολα μέσω μιας λογικής πύλης XOR.
- Η έξοδος V σηματοδοτεί την υπερχείλιση (V=1) των προσημασμένων αριθμών.

ΚΩΔΙΚΟΠΟΙΗΤΗΣ

Οι κωδικοποιητές και τα συμπληρωματικά τους κυκλώματα, οι αποκωδικοποιητές, χρησιμοποιούνται σε υπολογιστές και σε πολλές άλλες εφαρμογές είτε για να ενεργοποιήσουν μία μονάδα από ένα σύνολο n μονάδων (αποκωδικοποιητές) είτε, αντίθετα, για να ενημερώσουν μια κεντρική μονάδα ποια από ένα σύνολο n περιφεριακών μονάδων έχει ενεργοποιηθεί (κωδικοποιητής)

Ο **Κωδικοποιητής** (Encoder) 2^nxn είναι ένα συνδυαστικό κύκλωμα που έχει είσοδο από 2^n γραμμές και δίνει έξοδο από n γραμμές..

Ο Κωδικοποιητής παράγει στην έξοδό του το δυαδικό κώδικα που αντιστοιχεί στις εισόδους του.

Ο **Αποκωδικοποιητής (Decoder)** n x 2^n είναι ένα συνδυαστικό κύκλωμα που μετατρέπει τη δυαδική πληροφορία n γραμμών εισόδου σε 2^n γραμμές εξόδου που αποτελούν τους ελάχιστους όρους των μεταβλητών εισόδου.

Ο αποκωδικοποιητής, για παράδειγμα, αποτελεί τμήμα της Κεντρικής Μνήμης των υπολογιστών, στην οποία στέλνοντας τη διεύθυνση που θέλουμε να έχουμε πρόσβαση μέσα στη Μνήμη φροντίζει για την ενεργοποίηση των αντίστοιχων κυττάρων της Μνήμης. Μπορεί να χρησιμοποιηθεί επίσης, λόγω του χαμηλού κόστους, για την εύκολη υλοποίηση λογικών συναρτήσεων.

Ο αποκωδικοποιητής είναι ένα κύκλωμα που έχει:

m εισόδους και n εξόδους,

όπου n ≤ 2m, και ονομάζεται *αποκωδικοποιητής m σε n*.

Συνήθως υπάρχει και μία ακόμη είσοδος E, που ονομάζεται *είσοδος ενεργοποίησης*.

ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ 2x4

Ο Αποκωδικοποιητής 2x4 έχει δυο εισόδους Α και Β και τέσσερις εξόδους D0, D1, D2 και D3.

A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

D0=A’B’

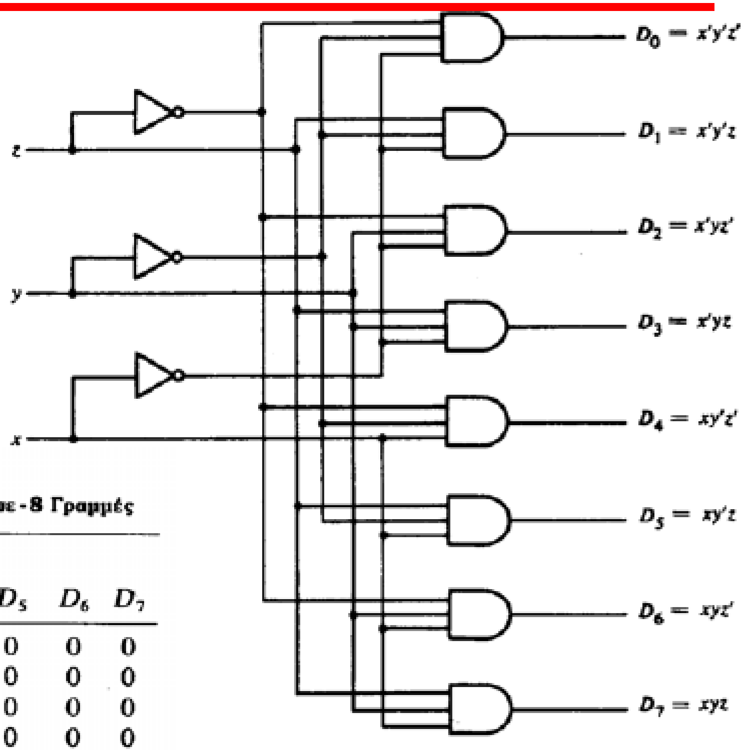
D1=A’B

D2=AB’

D3=AB

Αποκωδικοποιητές - Κωδικοποιητές

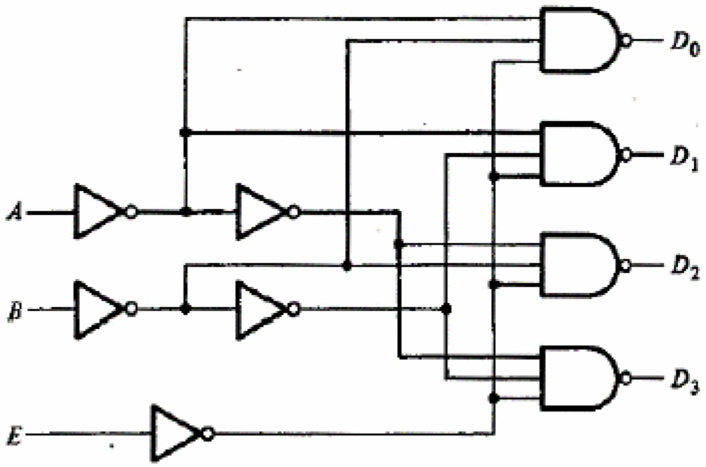
Αποκωδικοποιητής: κύκλωμα που μετατρέπει την δυαδική πληροφορία των n γραμμών εισόδου σε 2^n μοναδικές γραμμές εξόδου (ελαχιστόροι n μεταβλητών)



ΠΙΝΑΚΑΣ 5-2
Πίνακας Αληθείας ενός Αποκωδικοποιητή από 3-σε-8 Γραμμές

Είσοδοι			Έξοδοι							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Αποκωδικοποιητές



(α) Λογικό διάγραμμα

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

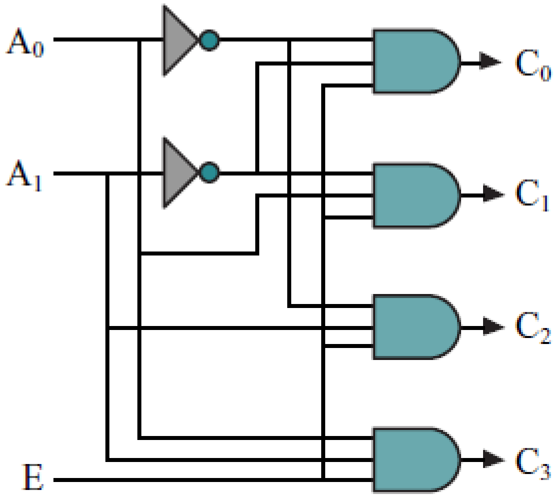
(β) Πίνακας αληθείας

Όταν E=0, τότε **όλες** οι έξοδοι του C_k=0 (k=0, …, n–1).

Όταν E=1, τότε η έξοδος C_A=1, όπου ο δείκτης A είναι ο δυαδικός αριθμός που σχηματίζεται από τις εισόδους, και οι υπόλοιπες έξοδοι C_k=0, όπου k=(0, …, n–1) και k≠A.

E	A_1	A_0	C_3	C_2	C_1	C_0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	X	X	0	0	0	0

(α)



(β)

ΥΛΟΠΟΙΗΣΗ ΣΥΝΔΥΑΣΤΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ ΜΕ ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗ

Κάθε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί με έναν **Αποκωδικοποιητή** $n \times 2^n$ και μία πύλη **OR**.

Η διαδικασία που ακολουθείται είναι η εξής:

- γράφεται η λογική συνάρτηση σε μορφή αθροίσματος ελαχίστων όρων
- σχεδιάζουμε το κύκλωμα με έναν Αποκωδικοποιητή $n \times 2^n$ και μία πύλη OR με εισόδους τους ελάχιστους όρους που αντιστοιχούν σε “1”.

Κάθε συνδυαστικό κύκλωμα n εισόδων και m εξόδων μπορεί να υλοποιηθεί με έναν Αποκωδικοποιητή $n \times 2^n$ και m πύλες OR.

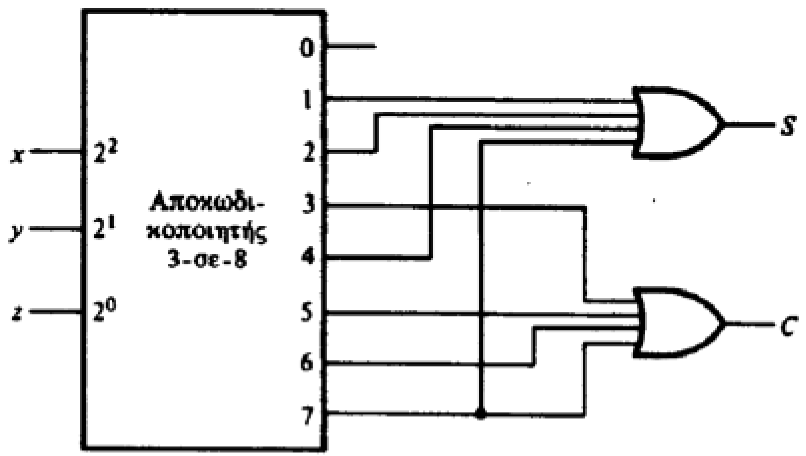
Υλοποίηση Συνδυαστικής Λογικής

Αφού ο αποκωδικοποιητής παράγει τους 2^n ελαχιστόρους μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση με προσθήκη πυλών OR

Παράδειγμα: Υλοποίηση πλήρους αθροιστή

$S(x,y,z)=\Sigma(1,2,4,7)$

$C(x,y,z)=\Sigma(3,5,6,7)$

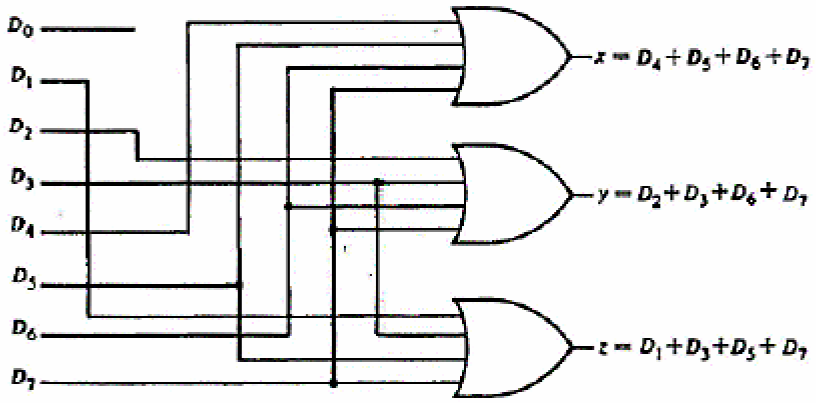


Κωδικοποιητές

Ο Κωδικοποιητής εκτελεί την αντίστροφη λειτουργία από τον Αποκωδικοποιητή: Έχει 2^n γραμμές εισόδου και n γραμμές εξόδου και δίνει τον δυαδικό κώδικα που αντιστοιχεί στις γραμμές εισόδου.

Είσοδοι								Έξοδοι		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Κωδικοποιητές



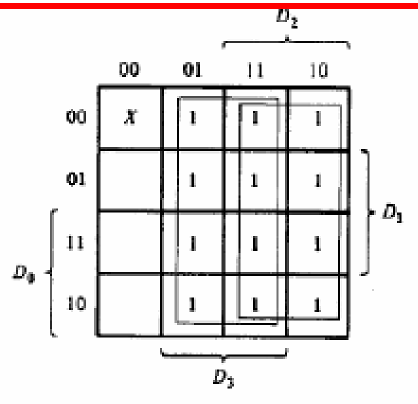
ΣΧΗΜΑ 5-13
Κωδικοποιητής από οκταδικό σε δυαδικό

Προβλήματα:

- Όταν περισσότερες της μίας είσοδοι είναι στον άσσο τότε η έξοδος είναι απροσδιόριστη (Λύση: προτεραιότητα)
- Όταν όλες οι είσοδοι είναι στο μηδέν τότε η έξοδος είναι 000 που δεν είναι σωστό αφού η $D_0 \neq 1$ (Λύση: διάκριση της κατάστασης)

Κωδικοποιητές Προτεραιότητας

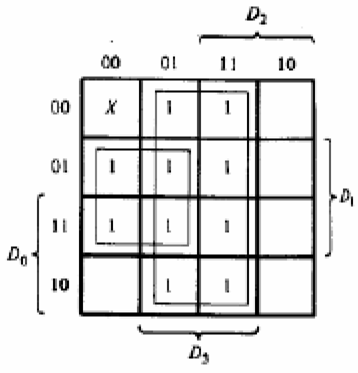
Ο Κωδικοποιητής εκτελεί την αντίστροφη λειτουργία από τον Αποκωδικοποιητή: Έχει 2^n γραμμές εισόδου και n γραμμές εξόδου και δίνει τον δυαδικό κώδικα που αντιστοιχεί στις γραμμές εισόδου.



$x = D_2 + D_3$

Πίνακας Αληθείας για έναν Κωδικοποιητή Προτεραιότητας

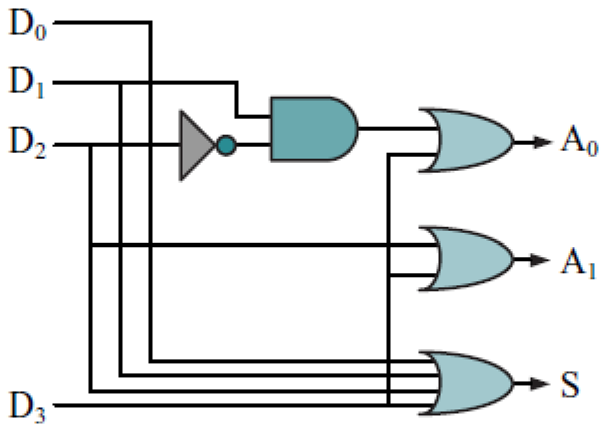
Είσοδοι				Έξοδοι		
D_0	D_1	D_2	D_3	X	Y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1



$y = D_3 + D_1 D_2'$

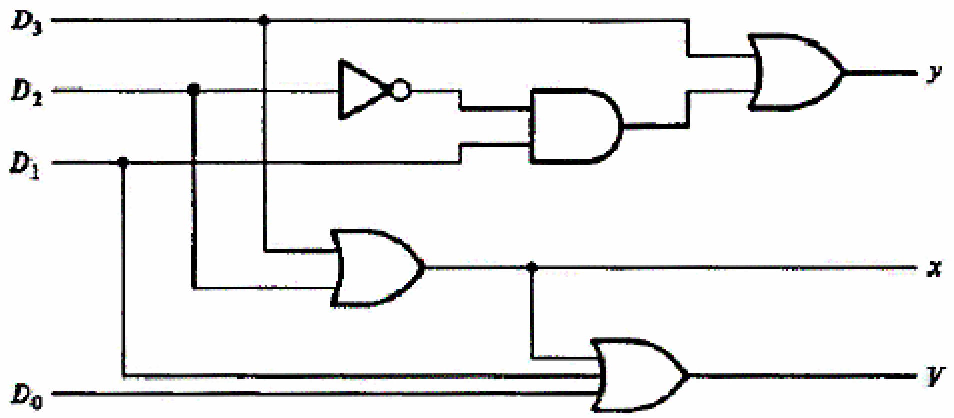
D_3	D_2	D_1	D_0	A_0	A_1	S
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

(α)



(β)

Κωδικοποιητές Προτεραιότητας

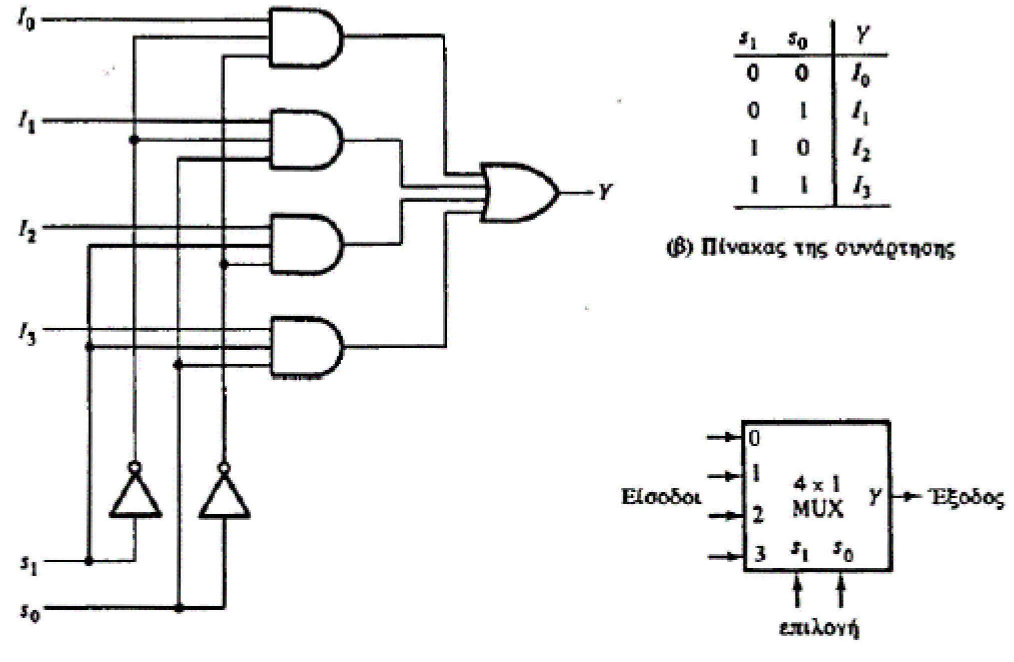


$x = D_2 + D_3$
$y = D_3 + D_1 D_2'$
$V = D_0 + D_1 + D_2 + D_3$

Λύνει το πρόβλημα της επιλογής όταν περισσότερες της μίας εισόδων είναι στον άσσο: επιλέγει αυτήν με την μεγαλύτερη προτεραιότητα

Πολυπλέκτες

Ο Πολυπλέκτης είναι ένα συνδυαστικό κύκλωμα που επιλέγει δυαδικές πληροφορίες ανάμεσα σε πολλές γραμμές εισόδου και τις κατευθύνει σε μία γραμμή εξόδου



s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(α) Λογικό διάγραμμα

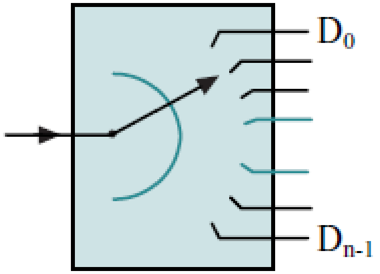
(β) Πίνακας της συνάρτησης

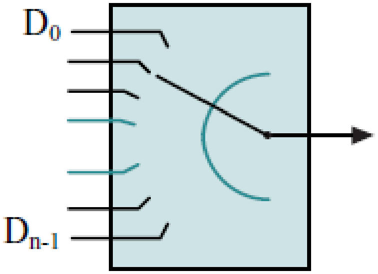
(γ) Σχηματικό διάγραμμα

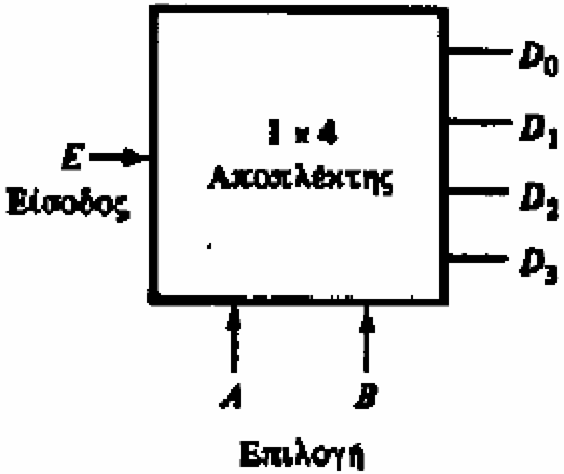
Σε πολλές εφαρμογές υπάρχει η ανάγκη να επιλέγονται δεδομένα από διαφορετικές πηγές, με τη δυνατότητα η επιλογή να καθορίζεται από άλλα δεδομένα. Εχουμε δύο περιπτώσεις: να επιλέγουμε δεδομένα από διαφορετικές πηγές ή να διανέμουμε δεδομένα σε πολλούς παραλήπτες.

Στην Ενότητα αυτή θα μελετήσουμε τα κυκλώματα του *πολυπλέκτη* και του *αποπλέκτη* και θα μάθουμε να τα σχεδιάζουμε και να τα χρησιμοποιούμε.

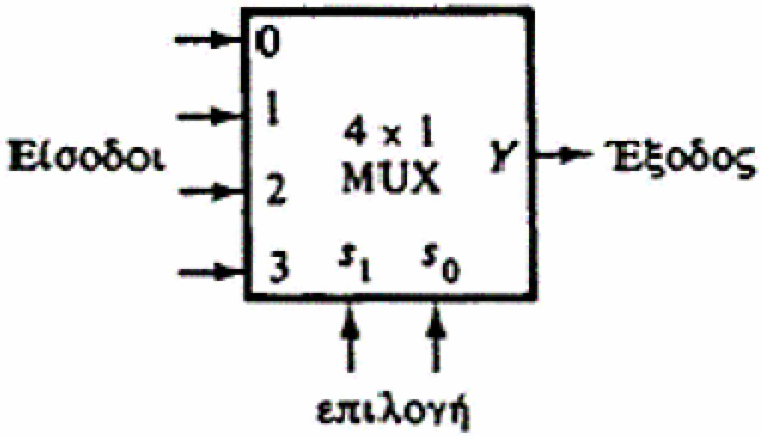
Με τη γενική ονομασία *πολυπλέκτης* συνήθως εννοούμε δύο κατηγορίες κυκλωμάτων, τον κανονικό *πολυπλέκτη* και τον *αποπλέκτη*, από τα οποία το καθένα εκτελεί την αντίθετη λειτουργία από το άλλο.







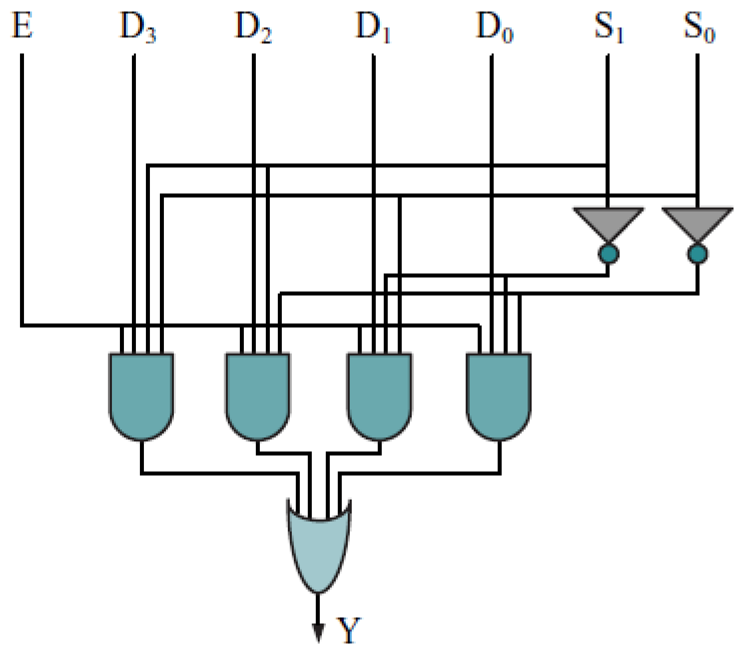
Αποπλέκτης



πολυπλέκτης

E	S_1	S_0	Y
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3
0	X	X	0

(α)



(β)

$$Y = E\,S_1\,S_0\,D_3 + E\,S_1\,S'_0\,D_2 + E\,S'_1\,S_0\,D_1 + E\,S'_1\,S'_0\,D_0$$

ΑΠΟΠΛΕΚΤΗΣ

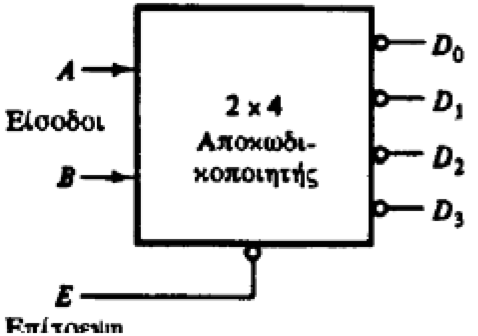
Ο αποπλέκτης είναι ένα κύκλωμα που εκτελεί ακριβώς την αντίστροφη εργασία από τον πολυπλέκτη,

Εχει μία είσοδο δεδομένων, n *εξόδους δεδομένων* D_i (i=0, …, n–1) και m *εισόδους επιλογής* S_j (j=0, …, m–1), όπου $n=2^m$. Ο δυαδικός αριθμός k, από 0 έως 2^m-1, που σχηματίζεται από τα σήματα στις εισόδους επιλογής, οδηγεί τον πολυπλέκτη να διοχετεύσει τα δεδομένα από την είσοδό του στην έξοδο D_k.

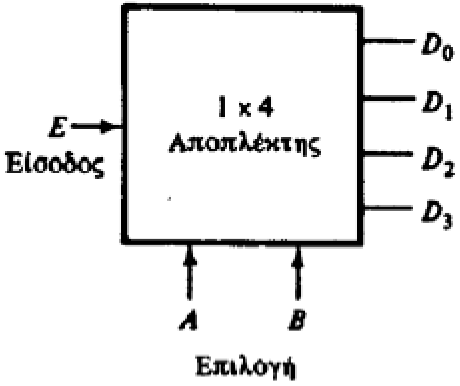
Οι εφαρμογές του πολυπλέκτη και του αποπλέκτη είναι πολλές τόσο σε κυκλώματα υπολογιστών (Κεντρική Μνήμη, επιλογή διαύλων, ολίσθηση των bits ενός αριθμού για την εκτέλεση των πράξεων του πολλαπλασιασμού και της διαίρεσης κ.λπ.), όσο και σε άλλα ψηφιακά συστήματα. Ενας συνδυασμός πολυπλέκτη–αποπλέκτη, χρησιμοποιείται για τη σειριακή μεταφορά - μιας δέσμης παράλληλων σημάτων $\Sigma_1 \ldots \Sigma_N$.

Αποπλέκτης

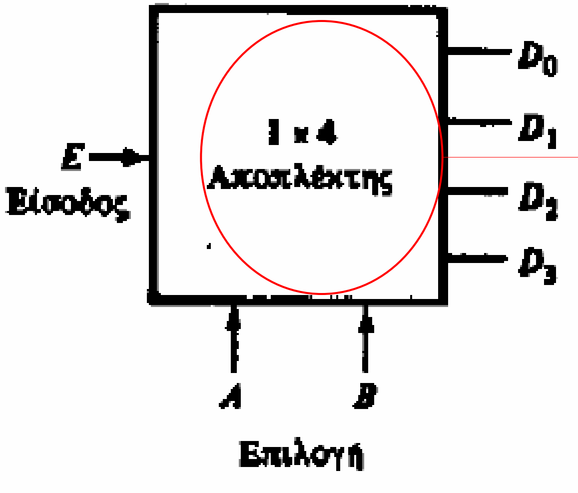
Είναι ουσιαστικά ένας αποκωδικοποιητής με συμπληρωματικές εξόδους και είσοδο επίτρεψης. Μεταβιβάζει τις πληροφορίες από την είσοδο επίτρεψης σε οποιαδήποτε έξοδο επιλέγουν οι υπόλοιπες είσοδοι



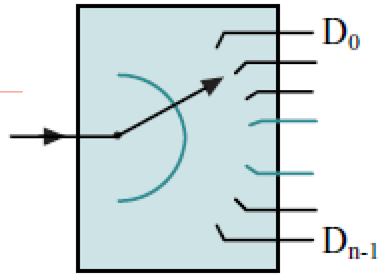
ψη



(β) Αποπλέκτης



Αποπλέκτης



ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗΣ ΑΠΟΠΛΕΚΤΗΣ 2x4

Ο Αποκωδικοποιητής/Αποπλέκτης 2x4 έχει δυο εισόδους A και B και τέσσερις εξόδους D0, D1, D2 και D3.

A	B	D0'	D1'	D2'	D3'
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

D0=A+B

D1=A+B’

D2=A’+B

D3=A’+B’

* Στην έξοδο D_i μεταφέρεται η τιμή της εισόδου, όπως αυτή επιλέγεται από το συνδυασμό της επιλογής.

ΥΛΟΠΟΙΗΣΗ ΣΥΝΔΥΑΣΤΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ ΜΕ ΑΠΟΚΩΔΙΚΟΠΟΙΗΤΗ/ΑΠΟΠΛΕΚΤΗ

Κάθε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί με έναν **Αποκωδικοποιητή/Αποπλέκτη** n$\times 2^n$ και μία πύλη **NAND**.

Κάθε συνδυαστικό κύκλωμα n εισόδων και m εξόδων μπορεί να υλοποιηθεί με έναν Αποκωδικοποιητή/Αποπλέκτη n$\times 2^n$ και m πύλες NAND.

Υλοποίηση Συναρτήσεων Boole

Αλγόριθμος υλοποίησης συνάρτησης με χρήση πολυπλέκτη:

1. Εκφράζουμε την συνάρτηση σε άθροισμα ελαχιστόρων.
2. Συνδέουμε τις n-1 μεταβλητές στις γραμμές επιλογής και κρατάμε την αριστερότερη (πιο σημαντική) έστω Α.
3. Καταγράφουμε τις εισόδους του πολυπλέκτη και κάτω από αυτές όλους τους ελαχιστόρους σε δύο σειρές (αντίστοιχα για Α=0 και Α=1).
4. Σημειώνουμε τους ελαχιστόρους που έχει η συνάρτηση.
5. Σε κάθε στήλη βάζουμε 0 αν δεν έχει σημειωθεί ελαχιστόρος, 1 αν έχουν σημειωθεί και οι δύο, Α΄ αν έχει σημειωθεί ο πάνω και Α αν έχει σημειωθεί ο κάτω ελαχιστόρος.

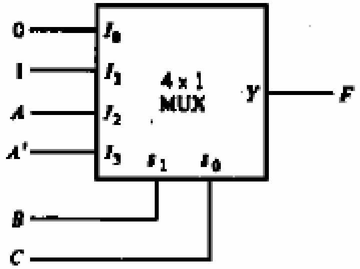
Τα κυκλώματα με λίγες εξόδους υλοποιούνται καλύτερα με πολυπλέκτες, ενώ αυτά με πολλές εξόδους υλοποιούνται καλύτερα με αποκωδικοποιητές

Υλοποίηση Συναρτήσεων Boole

Κάθε πολυπλέκτης 2^n σε 1 μπορεί να υλοποιήσει οποιαδήποτε συνάρτηση n μεταβλητών ως εξής:

1. Βάζουμε τις n-1 μεταβλητές στις εισόδους επίτρεψης.

2. Χρησιμοποιούμε την τελευταία μεταβλητή για τις εισόδους.



(α) Υλοποίηση με πολυπλέκτη

Ελαχιστόρος	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

(β) Πίνακας αληθείας

	I_0	I_1	I_2	I_3
A'	0	1	2	3
A	4	5	6	7
	0	1	A	A'

(γ) Πίνακας υλοποίησης

Υλοποίηση της $F(A,B,C) = \Sigma(1,3,5,6)$ με πολυπλέκτη

ΥΛΟΠΟΙΗΣΗ ΣΥΝΔΥΑΣΤΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ ΜΕ ΠΟΛΥΠΛΕΚΤΕΣ

Κάθε λογική συνάρτηση n μεταβλητών μπορεί να υλοποιηθεί χρησιμοποιώντας έναν Πολυπλέκτη 2^{n-1}x1. Οι n-1 μεταβλητές εισόδου της συνάρτησης αποτελούν τις γραμμές επιλογής του Πολυπλέκτη. Κάθε είσοδος του Πολυπλέκτη είναι η n-οστή μεταβλητή ή το συμπλήρωμά της ή το “0” ή το “1”, όπως προκύπτει από τον **πίνακα υλοποίησης** του Πολυπλέκτη. Η συνάρτηση αποτελεί την έξοδο του Πολυπλέκτη.

Κάθε συνδυαστικό κύκλωμα n εισόδων και m εξόδων μπορεί να υλοποιηθεί με m Πολυπλέκτες 2^{n-1}x1.

ΛΟΓΙΚΗ ΣΥΝΑΡΤΗΣΗ ΜΕ ΠΟΛΥΠΛΕΚΤΗ

Η συνάρτηση τριών μεταβλητών **F(A,B,C)=Σ(1,3,5,6)**

μπορεί να υλοποιηθεί με έναν Πολυπλέκτη 8x1

που έχει οκτώ εισόδους I0, I1, I2, I3, I4, I5, I6, I7,

τρεις επιλογές S2, S1, S0

και μία έξοδο Y.

Οι μεταβλητές εισόδου της συνάρτησης αποτελούν τις γραμμές επιλογής του Πολυπλέκτη: S2=A, S1=B, S0=C

Οι είσοδοι του Πολυπλέκτη επιλέγονται κατάλληλα από τον πίνακα αληθείας της συνάρτησης: I0=0, I1=1, I2=0, I3=1, I4=0, I5=1 I6=1, I7=0

Η συνάρτηση αποτελεί την έξοδο του Πολυπλέκτη: Y=F

$F = A'B'C' I_0 + A'B'CI_1 + A'BC'I_2 + \ldots + ABCI$

ΣΥΝΔΥΑΣΤΙΚΟ ΚΥΚΛΩΜΑ ΜΕ ΠΟΛΥΠΛΕΚΤΕΣ

ΠΑΡΑΔΕΙΓΜΑ

Η συνάρτηση τριών μεταβλητών

$$\mathbf{F(A,B,C)=\Sigma(1,3,5,6)}$$

μπορεί να υλοποιηθεί με έναν Πολυπλέκτη 4x1

που έχει τέσσερις εισόδους I0, I1, I2, I3,

δυο επιλογές S1, S0 και μία έξοδο Y.

Αν τις μεταβλητές B και C χρησιμοποιηθούν ως επιλογές του Πολυπλέκτη: S1=B, S0=C
τότε, ο πίνακας υλοποίησης του Πολυπλέκτη είναι:

	I0	I1	I2	I3
A’	0	1	2	3
A	4	5	6	7
	0	1	A	A’

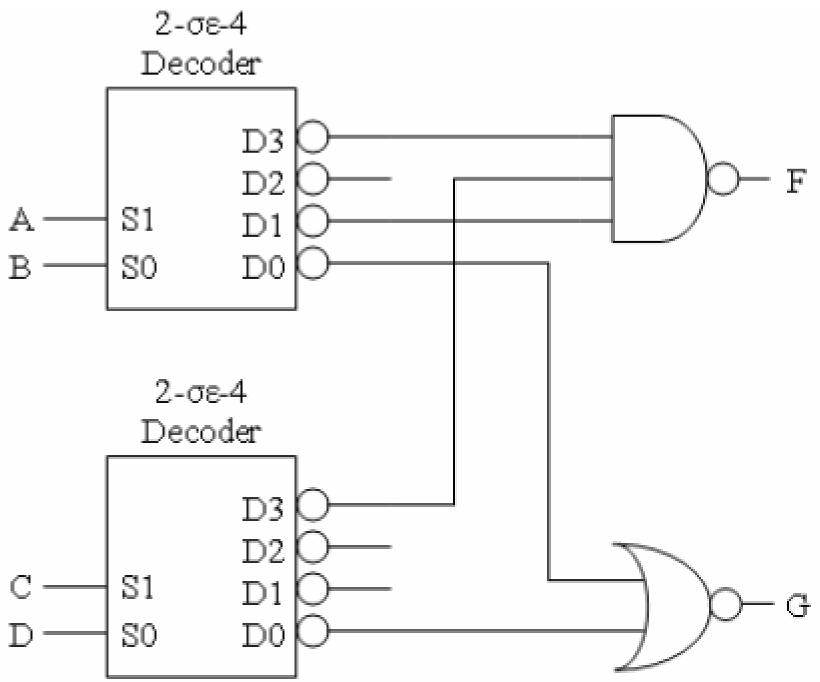
Οι είσοδοι του Πολυπλέκτη επιλέγονται κατάλληλα από τον πίνακα υλοποίησης του Πολυπλέκτη:
I0=0, I1=1, I2=A, I3=A’
Η συνάρτηση αποτελεί την έξοδο του Πολυπλέκτη: Y=F

Αν τις μεταβλητές A και B χρησιμοποιηθούν ως επιλογές του Πολυπλέκτη: S1=A, S0=B
τότε, ο πίνακας υλοποίησης του Πολυπλέκτη είναι:

	I0	I1	I2	I3
C’	0	2	4	6
C	1	3	5	7
	C	C	C	C’

Οι είσοδοι του Πολυπλέκτη επιλέγονται κατάλληλα από τον πίνακα υλοποίησης του Πολυπλέκτη:
I0=C, I1=C, I2=C, I3=C’
Η συνάρτηση αποτελεί την έξοδο του Πολυπλέκτη: Y=F

Να γράψετε τις συναρτήσεις F(A,B,C,D) και G(A,B,C,D), που υλοποιεί το παρακάτω κύκλωμα σε ελάχιστη μορφή αθροίσματος γινομένων καθώς και σε κανονική μορφή αθροίσματος ελάχιστων όρων (sum of minterms). (Το ψηφίο A είναι το ΠΣΨ).



- Από τη λειτουργία του αποκωδικοποιητή με ανεστραμμένες (active low) εξόδους έχουμε:
- Για τον επάνω decoder: $D0 = (A'\cdot B')'$, $D1 = (A'\cdot B)'$, $D2 = (A\cdot B')'$, $D3 = (A\cdot B)'$.
- Για τον κάτω decoder: $D0 = (C'\cdot D')'$, $D1 = (C'\cdot D)'$, $D2 = (C\cdot D')'$, $D3 = (C\cdot D)'$.

Οι **F** και **G** υπολογίζονται σαν (ελάχιστα) αθροίσματα ως εξής:

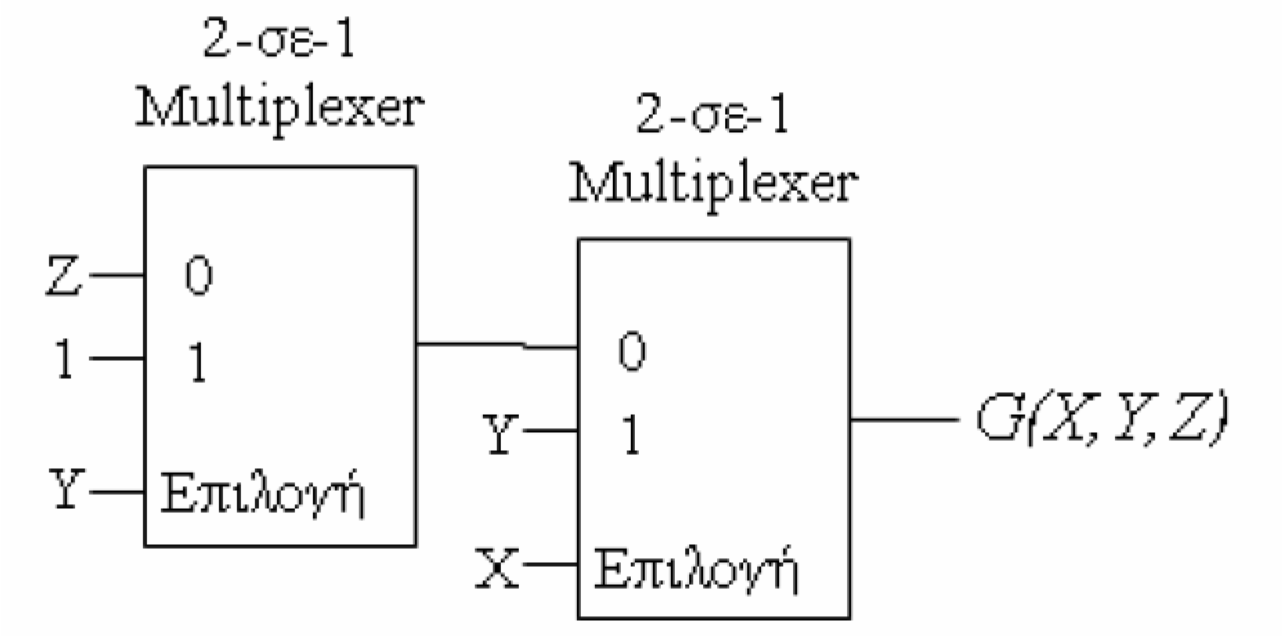
$$\mathbf{F} = ((A\cdot B)' \cdot (A'\cdot B)' \cdot (C\cdot D)')' =$$
$$(A\cdot B)'' + (A'\cdot B)'' + (C\cdot D)'' =$$
$$A\cdot B + A'\cdot B + C\cdot D =$$
$$(A+A')\cdot B + C\cdot D =$$
$$1\cdot B + C\cdot D = B + C\cdot D.$$

$$\mathbf{G} = ((A'\cdot B')' + (C'\cdot D'))' = (A'\cdot B')'' \cdot (C'\cdot D')'' = A'\cdot B'\cdot C'\cdot D'.$$

οι συναρτήσεις F και G γράφονται σε κανονική μορφή ελαχίστων όρων (sum of minterms) ως εξής:

- $F = \Sigma\ \{3,4,5,6,7,11,12,13,14,15\}$
- $G = \Sigma\ \{0\}$.

Να βρείτε τη δυαδική συνάρτηση *G(X,Y,Z)* που υλοποιεί το παρακάτω λογικό κύκλωμα.



- Έστω $F(Y,Z)$ η συνάρτηση εξόδου του πρώτου πολυπλέκτη.

Τότε ισχύει: $F = Y \cdot 1 + Y' \cdot Z = Y \cdot (Z+1) + Y' \cdot Z = Y+Z$.

- Αντίστοιχα, για τη συνάρτηση εξόδου του δεύτερου πολυπλέκτη ισχύει:

$G = X \cdot Y + X' \cdot F = X \cdot Y + X' \cdot (Y+Z) = X \cdot Y + X' \cdot Y + X' \cdot Z = Y + X' \cdot Z$

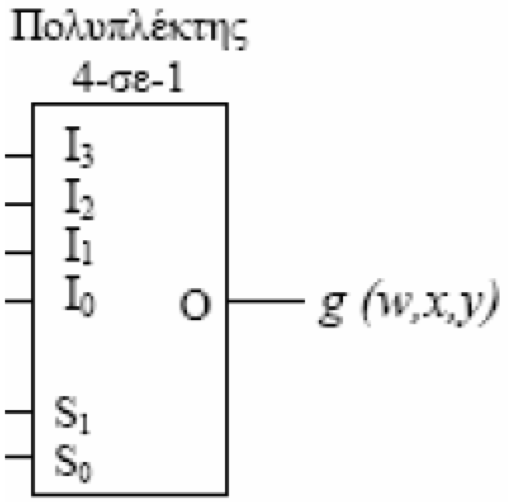
- $\boldsymbol{G = Y + X' \cdot Z}$

Άσκηση

Να υλοποιήσετε τη συνάρτηση

$$g(w,x,y) = w \cdot y + x' \cdot y + w' \cdot x \cdot y'$$

χρησιμοποιώντας μόνο έναν πολυπλέκτη 4-σε-1. Να υποθέσετε ότι οι είσοδοι ελέγχου είναι: $S_0=x$, $S_1=w$ και ότι τα συμπληρώματα των μεταβλητών w,x,y είναι διαθέσιμα.



Η λογική συνάρτηση την οποία υλοποιεί ο πολυπλέκτης είναι η ακόλουθη:

$$O = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$
$$= w'x'I_0 + w'xI_1 + wx'I_2 + wxI_3$$

το οποίο σημαίνει ότι:

$O = I_0$, μόνο όταν $w = 0$ και $x = 0$, δηλ. $w'x' = 1$, άρα $I_0 = g(0,0,y)$

$O = I_1$, μόνο όταν $w = 0$ και $x = 1$, δηλ. $w'x = 1$, άρα $I_1 = g(0,1,y)$

$O = I_2$, μόνο όταν $w = 1$ και $x = 0$, δηλ. $wx' = 1$, άρα $I_2 = g(1,0,y)$

$O = I_3$, μόνο όταν $w = 1$ και $x = 1$, δηλ. $wx = 1$, άρα $I_3 = g(1,1,y)$.

Έτσι,

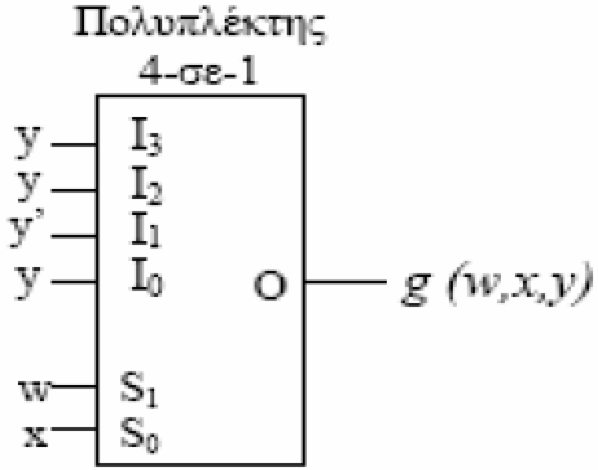
Έτσι,

$$I_0 = g(0,0,y) = 0 \cdot y + 1 \cdot y + 0 \cdot y' = y$$

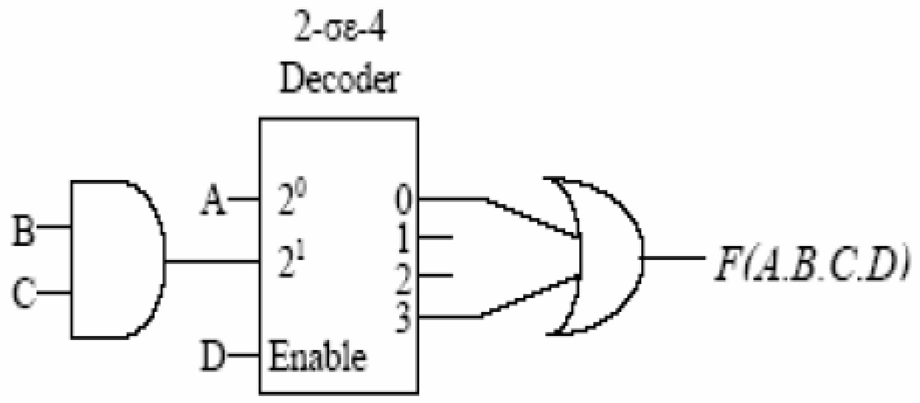
$$I_1 = g(0,1,y) = 0 \cdot y + 0 \cdot y + 1 \cdot 1 \cdot y' = y'$$

$$I_2 = g(1,0,y) = 1 \cdot y + 1 \cdot y + 0 \cdot 0 \cdot y' = y$$

$$I_3 = g(1,1,y) = 1 \cdot y + 0 \cdot y + 0 \cdot 1 \cdot y' = y$$



Να βρείτε τη δυαδική συνάρτηση *F* που υλοποιεί το παρακάτω λογικό κύκλωμα.



Η λύση να απλοποιηθεί και να εκφραστεί σε μορφή αθροίσματος γινομένων.

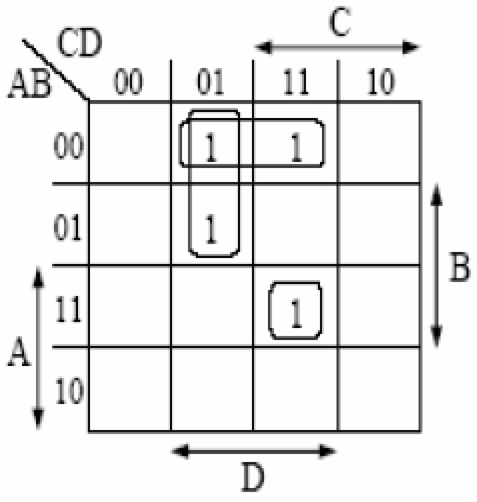
Έστω F_0 και F_3 οι συναρτήσεις των εξόδων 0 και 3 του αποκωδικοποιητή, αντίστοιχα.

Τότε: $F_0 = D\cdot[(B\cdot C)' \cdot A'] = D\cdot[(B'+C') \cdot A'] = A'\cdot B'\cdot D + A'\cdot C'\cdot D$

και: $F_3 = D\cdot[(B\cdot C) \cdot A] = A\cdot B\cdot C\cdot D$.

Συνεπώς: $F = F_0 + F_3 = A'\cdot B'\cdot D + A'\cdot C'\cdot D + A\cdot B\cdot C\cdot D$

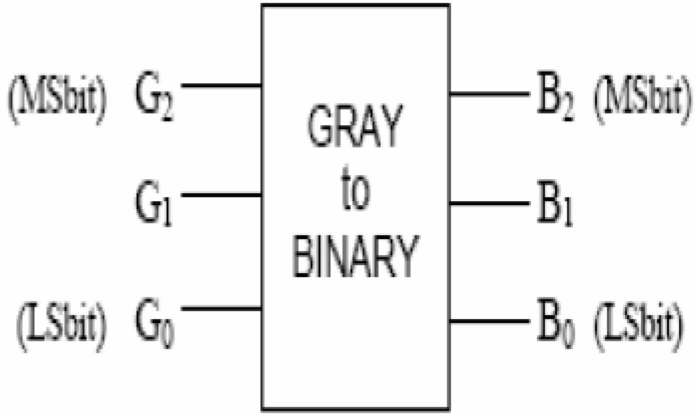
Με χάρτη Karnaugh εύκολα επιβεβαιώνουμε ότι η παραπάνω είναι και η απλοποιημένη



μορφή της συνάρτησης F.

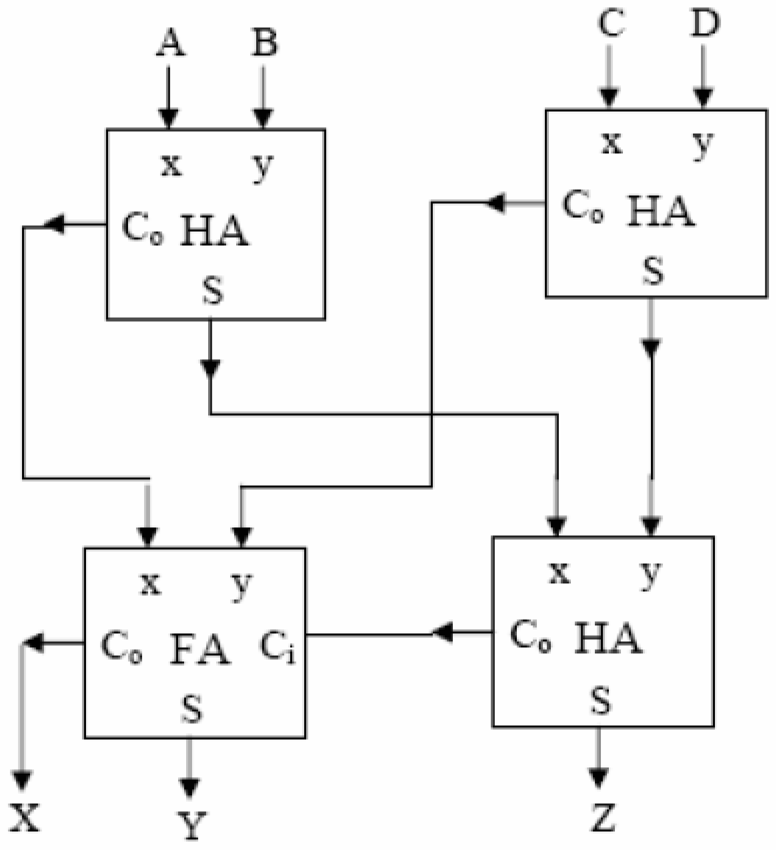
Άσκηση

Χρησιμοποιώντας τον ελάχιστο δυνατό αριθμό λογικών πυλών, να σχεδιάσετε κύκλωμα το οποίο να μετατρέπει αριθμούς των 3 bits από τον ανακλαστικό κώδικα Gray στο δυαδικό κώδικα με βάρη 4-2-1.



Υλοποιήστε τις εξόδους χρησιμοποιώντας: a) πολυπλέκτες 4->1, b) 3 πολυπλέκτες 3->8

Δίνεται το παρακάτω κύκλωμα, το οποίο μετατρέπει έναν τετραψήφιο κώδικα (ABCD) σε τριψήφιο (XYZ). Αναλύστε τη λειτουργία του κυκλώματος και συμπληρώστε τον πίνακα αλήθειας τc



Gray			Binary 4-2-1		
G_2	G_1	G_0	B_2	B_1	B_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	1	0	1	0
0	1	0	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1
1	0	1	1	1	0
1	0	0	1	1	1