4η Θεματική Ενότητα : Συνδυαστική Λογική

Ύλη Λογικού Σχεδιασμού Ι

- Κεφ 2
- Κεφ 3
- Κεφ 4
- Κεφ 6

Εισαγωγή

Λογικά Κυκλώματα
- Συνδυαστικά: Οι έξοδοι είναι συνάρτηση των εισόδων
- Ακολουθιακά: Οι έξοδοι είναι συνάρτηση των εισόδων και της κατάστασης των στοιχείων μνήμης(προηγούμενες είσοδοι)

n είσοδοι → Συνδυαστικό Λογικό Κύκλωμα → m έξοδοι } m συναρτήσεις Boole n μεταβλητών

Διαδικασία Σχεδιασμού

1.Καθορισμός Προβλήματος
2.Καθορισμός εισόδων/εξόδων
3.Ονομασία εισόδων/εξόδων
4.Πίνακας Αλήθειας
5.Απλοποίηση συναρτήσεων
6.Σχεδιασμός Λογικού Διαγράμματος

Επιλογή Απλοποιημένης Έκφρασης

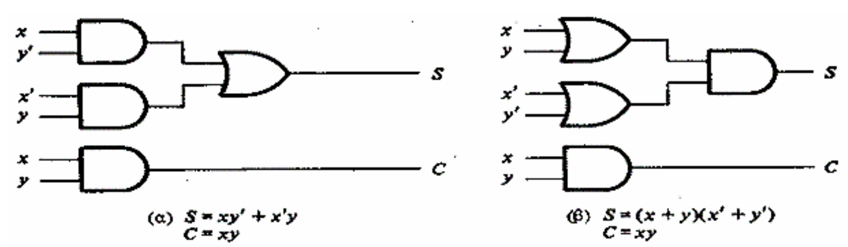
1.Ελάχιστος αριθμός Πυλών
2.Ελάχιστος αριθμός εισόδων Πύλης
3.Ελάχιστο χρόνο διάδοσης σήματος
4.Ελάχιστος αριθμός διασυνδέσεων
5.Περιορισμοί οδήγησης

Ημι-Αθροιστής

Άθροιση 2 bits ⇒ Κρατούμενο (Carry), Άθροισμα (Sum). Το κρατούμενο τροφοδοτεί την επόμενη σημαντικότερη βαθμίδα

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(α)	(β)	(γ)	(δ)
S=x΄y+xy΄	S=(x+y)(x΄+y΄)	S=(C+x΄y΄)΄	S=C΄(x+y)
C=xy	C=xy	C=xy	C=(x΄+y΄)΄



Ημι-Αθροιστής

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(γ)

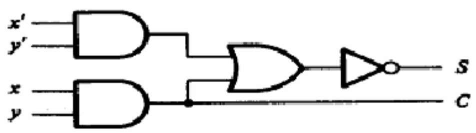
S=(C+x΄y΄)΄

C=xy

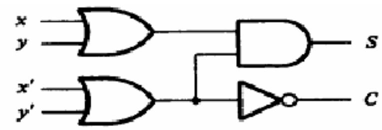
(δ)

S=C΄(x+y)

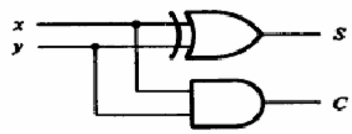
C=(x΄+y΄)΄



(γ) $S = (C + x'y')'$
$C = xy$



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

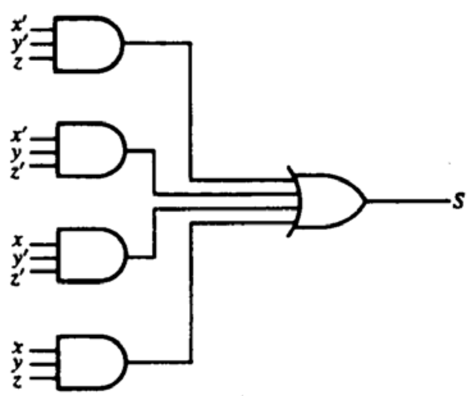


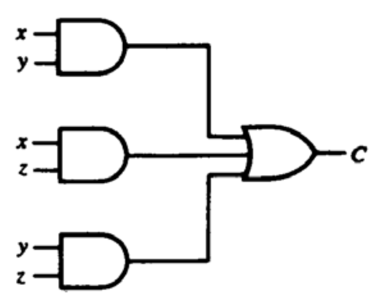
(ε) $S = x \oplus y$
$C = xy$

Υλοποιήσεις Ημιαθροιστή

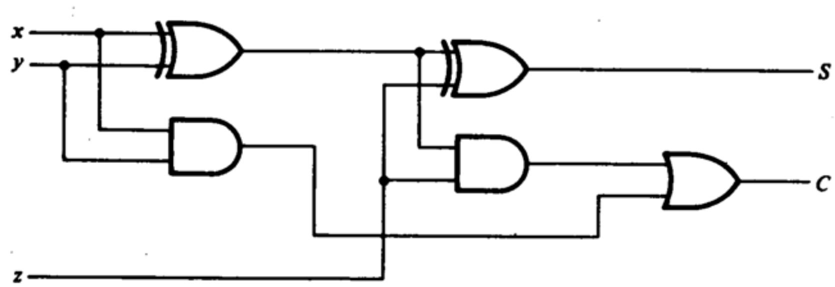
Πλήρης-Αθροιστής

Άθροιση 3 bits ⇒ Κρατούμενο (Carry), Άθροισμα (Sum). Το κρατούμενο τροφοδοτεί την επόμενη σημαντικότερη βαθμίδα





x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Αφαιρέτες

Η αφαίρεση γίνεται με πρόσθεση στον μειωτέο του συμπληρώματος του αφαιρετέου.

Ημιαφαιρέτης

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

$D = x'y + xy'$

$B = x'y$

Η D είναι ίδια με την συνάρτηση S του Αθροιστή

Πλήρης Αφαιρέτης

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

x: Μειωτέος

y: Αφαιρετέος

z: Προηγούμ. Κρατούμενο

Μετατροπή Κωδίκων

Η ύπαρξη πολλών κωδίκων οδηγεί στην ανάγκη μετατροπών ανάλογα με την λειτουργία του κάθε συστήματος. Γίνεται κυρίως για λόγους επικοινωνίας μεταξύ τους.

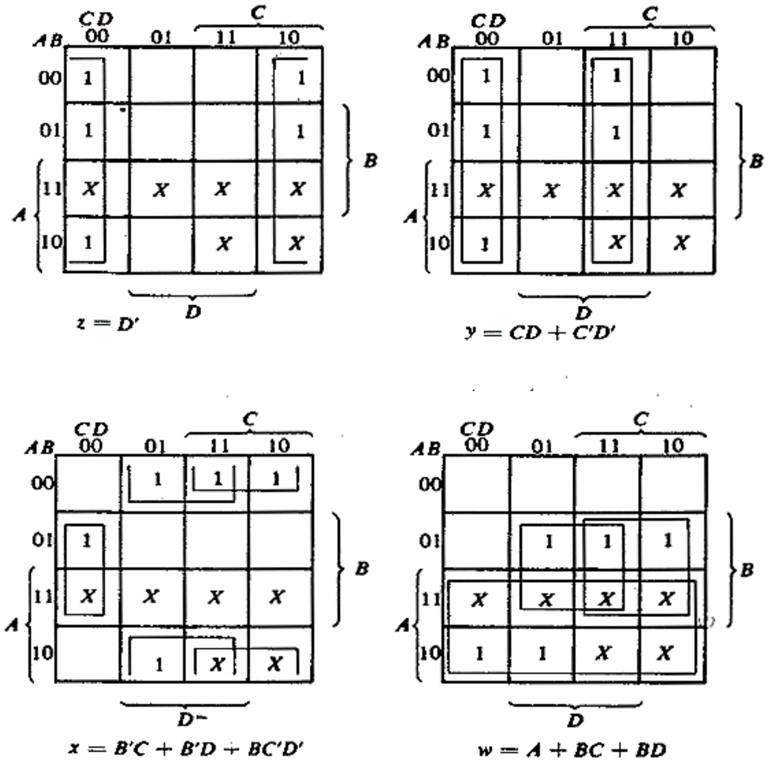
ΠΙΝΑΚΑΣ 4-1
Πίνακας Αληθείας για το Παράδειγμα Μετατροπής Κώδικα

Είσοδος Κώδικας BCD				Έξοδος Κώδικας excess-3			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Οι αχρησιμοποίητες καταστάσεις μπορούν να αποτελέσουν αδιάφορους όρους.

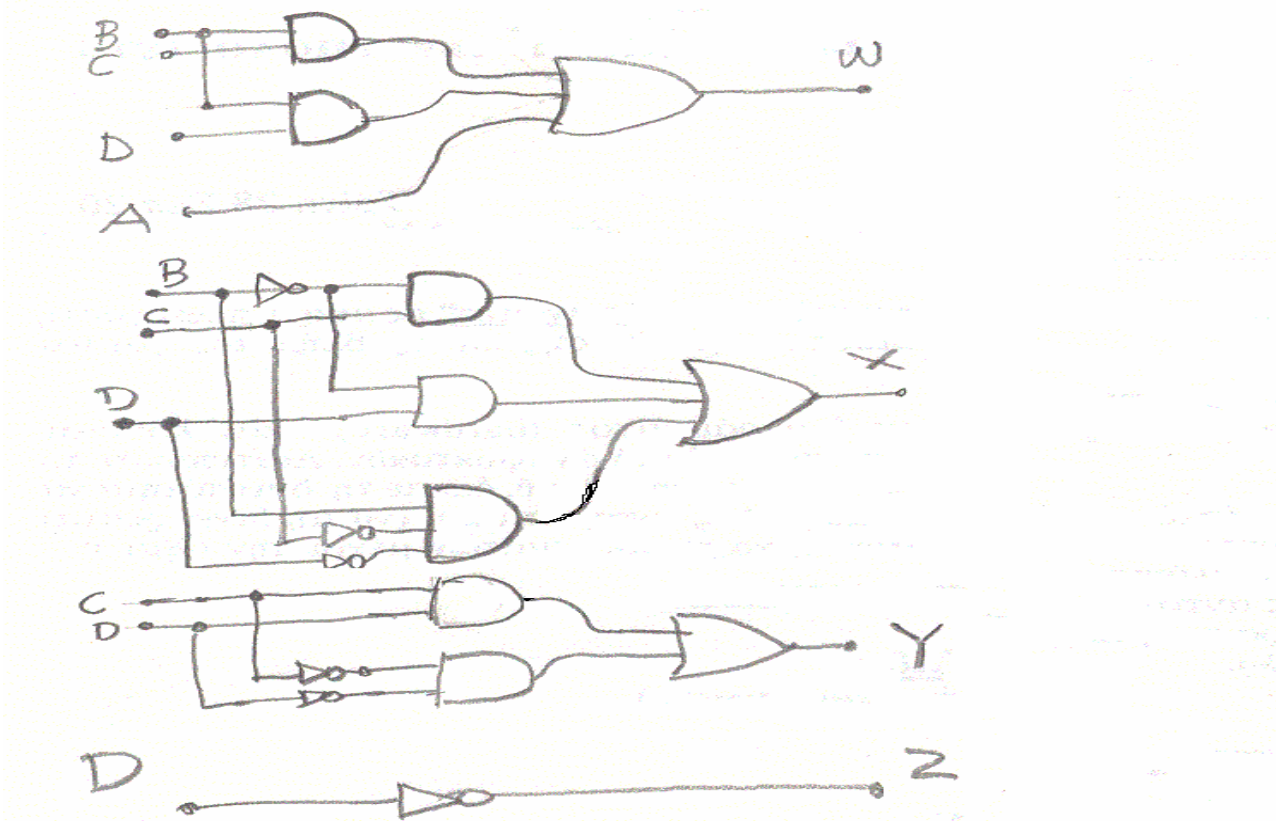
Μετατροπή Κωδίκων

Χάρτες μετατροπέα κώδικα BCD σε excess-3



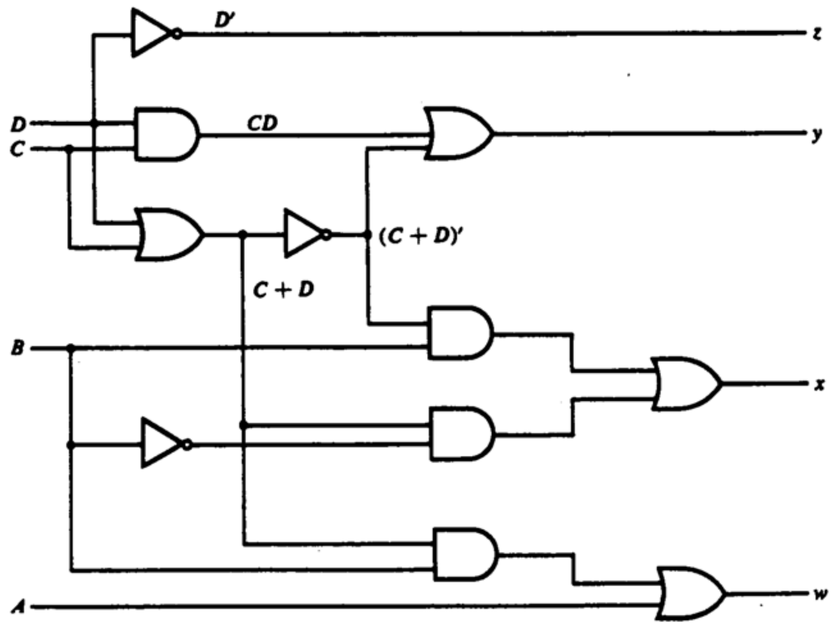
$W = A + BC + BD$ $X = B'C + B'D + BC'D'$

$Y = CD + C'D'$ $Z = D'$



Μετατροπή Κωδίκων

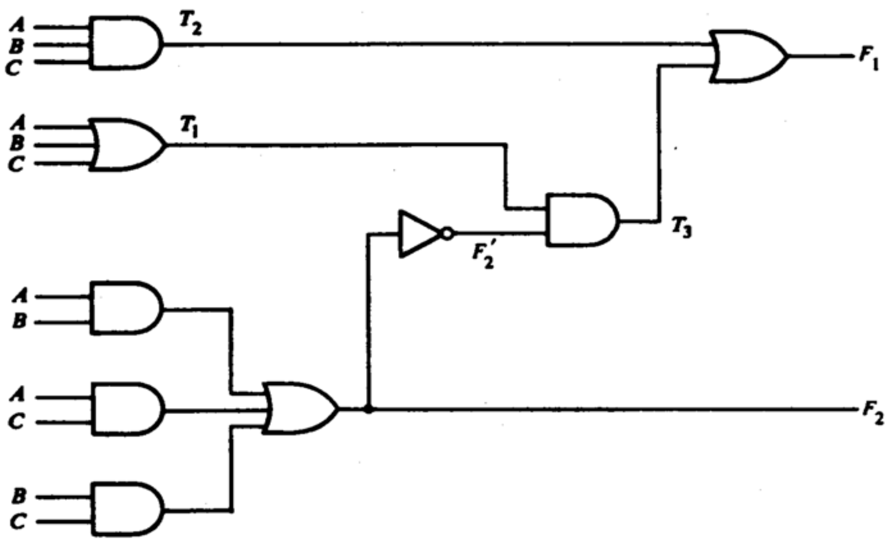
Z=D´
y=CD+C´D´=CD+(C+D)´
x=B´C+B´D+BC´D´=
=B´(C+D)+BC´D´=
=B´(C+D)+B(C+D)´
w=A+BC+BD=A+B(C+D)



ΣΧΗΜΑ 4-8
Λογικό διάγραμμα του μετατροπέα κώδικα BCD σε excess-3

Ανάλυση Κυκλώματος

Από το κύκλωμα βρίσκουμε τις συναρτήσεις Boole:
1. Ονομάζουμε τις εισόδους του κυκλώματος
2. Βρίσκουμε τις συναρτήσεις σε κάθε επίπεδο μέχρι το τελευταίο.



ΣΧΗΜΑ 4-9
Λογικό διάγραμμα για το παράδειγμα ανάλυσης

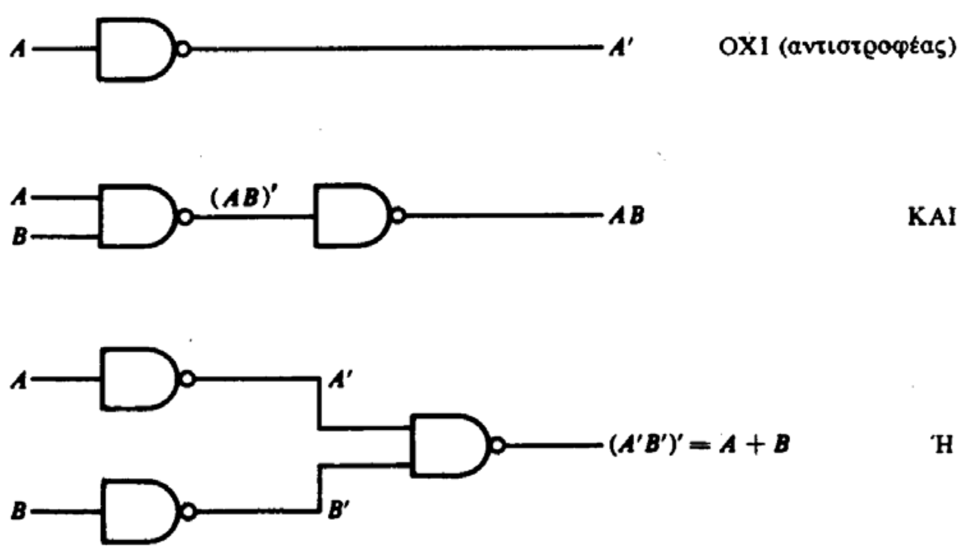
Ανάλυση Κυκλώματος

ΠΙΝΑΚΑΣ 4-2
Πίνακας Αληθείας του Λογικού Διαγράμματος του Σχήματος 4-9

A	B	C	F_2	F_2'	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

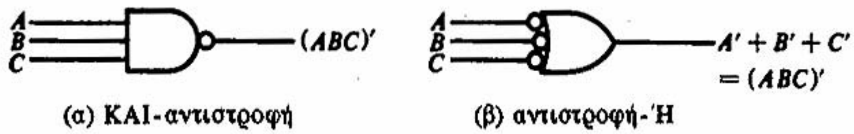
Οικουμενικότητα Πύλης Όχι-Και

Οικουμενική Πύλη: Κάθε ψηφιακό σύστημα μπορεί να υλοποιηθεί με αυτήν.



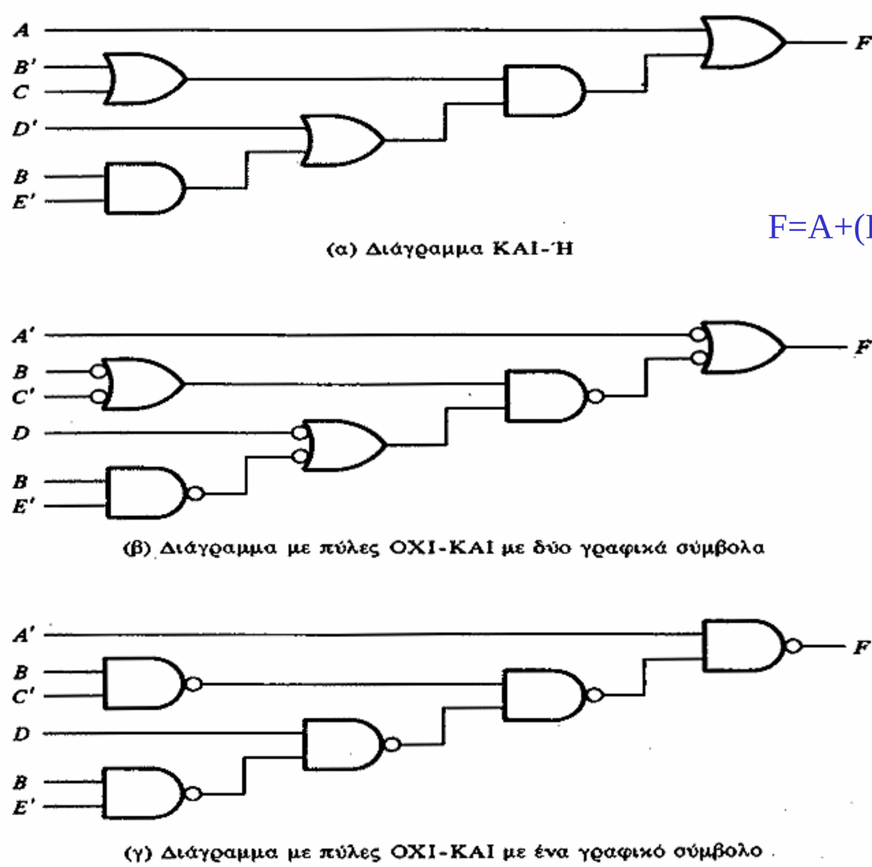
ΣΧΗΜΑ 4-10
Υλοποίηση των ΟΧΙ, ΚΑΙ, και Ή με πύλες ΟΧΙ-ΚΑΙ

Οικουμενικότητα Πύλης Όχι-Και



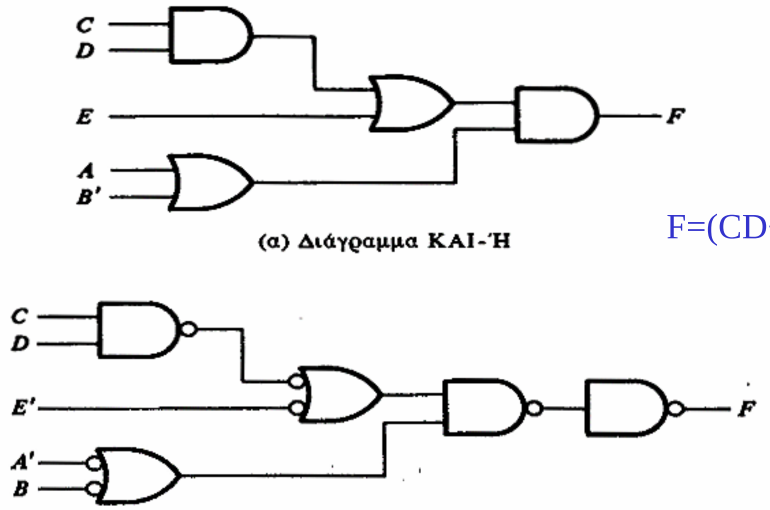
ΣΧΗΜΑ 4-11
Δύο συμβολισμοί για μια πύλη ΟΧΙ-ΚΑΙ

1. Σχεδιάζουμε το λογικό διάγραμμα με πύλες ΚΑΙ, Η και ΌΧΙ
2. Μετατρέπουμε όλες τις πύλες ΚΑΙ σε ΌΧΙ-ΚΑΙ με σύμβολα ΚΑΙ-αντιστροφής
3. Μετατρέπουμε όλες τις πύλες Η σε ΌΧΙ-ΚΑΙ με σύμβολα αντιστροφής-Η
4. Για κάθε κύκλο που δεν αναιρείται βάζουμε έναν αντιστροφέα



$F=A+(B'+C)(D'+BE')$

ΣΧΗΜΑ 4-12
Υλοποίηση της $F = A + (B' + C)(D' + E)$ με πύλες ΟΧΙ-ΚΑΙ



(α) Διάγραμμα ΚΑΙ-Ή

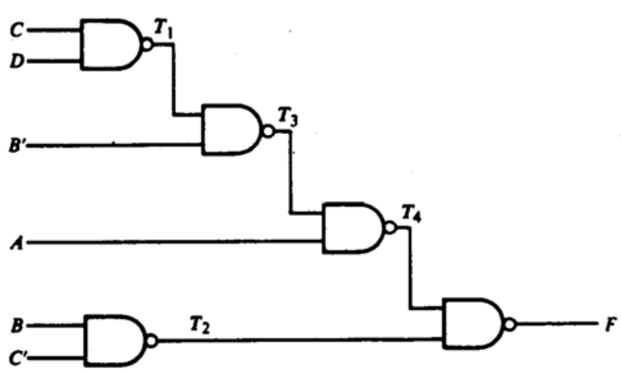
F=(CD+E)(A+B′)

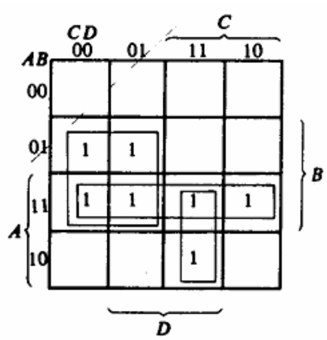
(β) Διάγραμμα ΟΧΙ-ΚΑΙ

(γ) Εναλλακτικό διάγραμμα με πύλες ΟΧΙ-ΚΑΙ

ΣΧΗΜΑ 4-13
Υλοποίηση της $F = (CD + E)(A + B')$ με πύλες ΟΧΙ-ΚΑΙ

Εξαγωγή Συνάρτησης & Πίνακα Αλήθειας





$F = AB + BC' + ACD$

$T_1=(CD)'=C'+D'$

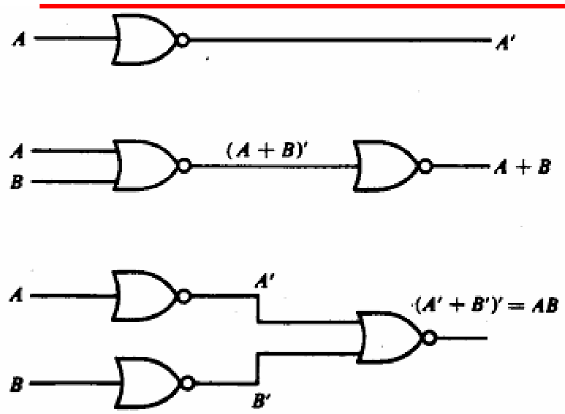
$T_2=(BC')'=B'+C$

$T_3=(B'T_1)'=\ldots=B+CD$

$T_4=(AT_3)'=[A(B+CD)]'$

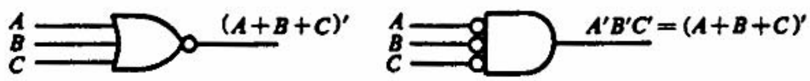
$F=(T_2T_4)=\ldots=BC'+A(B+CD)$

Οικουμενικότητα Πύλης Ούτε



ΣΧΗΜΑ 4-17
Υλοποίηση των πυλών ΟΧΙ, Ή, ΚΑΙ με πύλες-ΟΥΤΕ

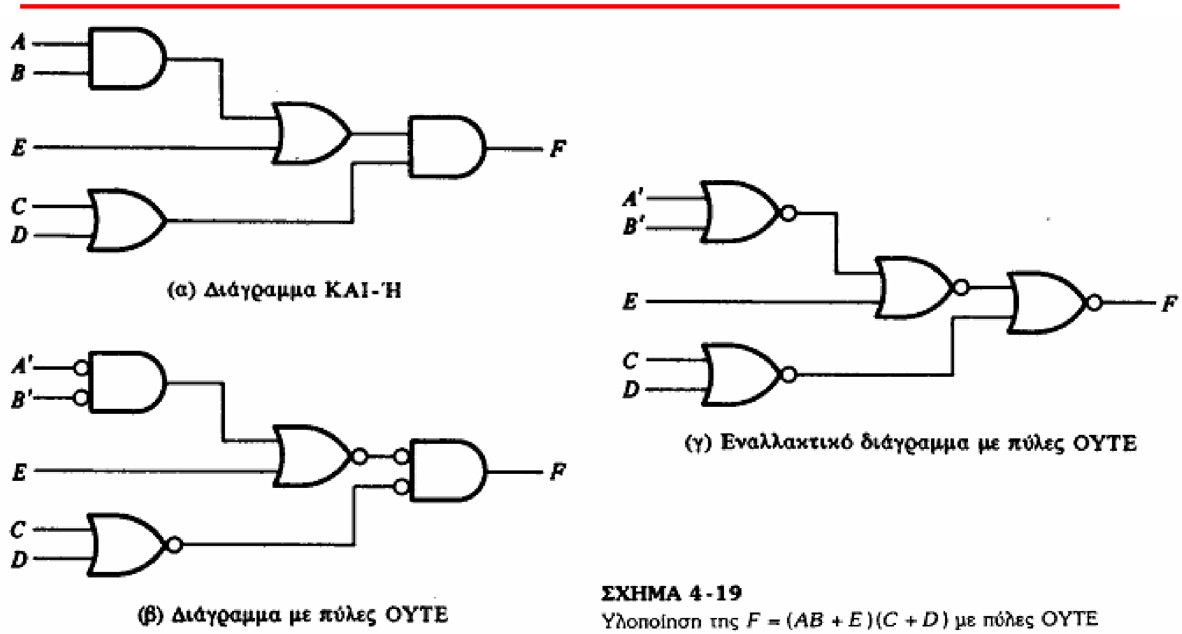
1. Σχεδιάζουμε το λογικό διάγραμμα με πύλες ΚΑΙ, Η και ΌΧΙ
2. Μετατρέπουμε όλες τις πύλες Η σε ΟΥΤΕ με σύμβολα αντιστροφής-Η
3. Μετατρέπουμε όλες τις πύλες ΚΑΙ σε ΌΥΤΕ με σύμβολα αντιστροφής-ΚΑΙ
4. Για κάθε κύκλο που δεν αναιρείται βάζουμε έναν αντιστροφέα



(α) Ή-αντιστροφή (β) Αντιστροφή-ΚΑΙ

ΣΧΗΜΑ 4-18
Δύο σύμβολα για μία πύλη ΟΥΤΕ

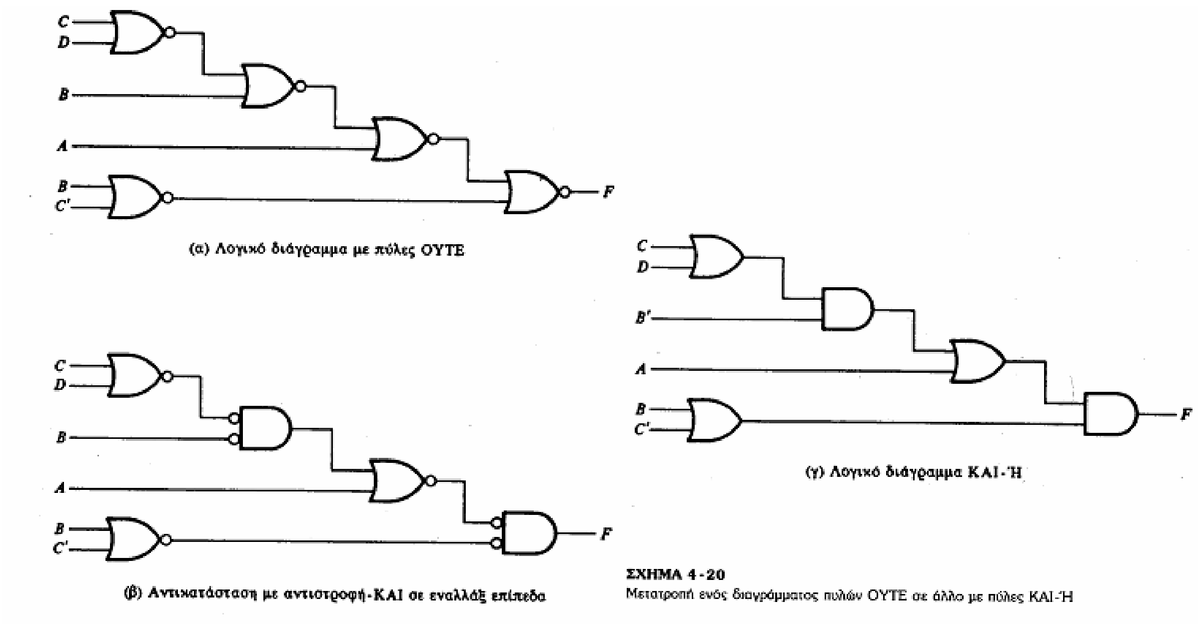
Οικουμενικότητα Πύλης Ούτε



ΣΧΗΜΑ 4-19
Υλοποίηση της $F = (AB + E)(C + D)$ με πύλες ΟΥΤΕ

F=(AB+E)(C+D)

Οικουμενικότητα Πύλης Ούτε



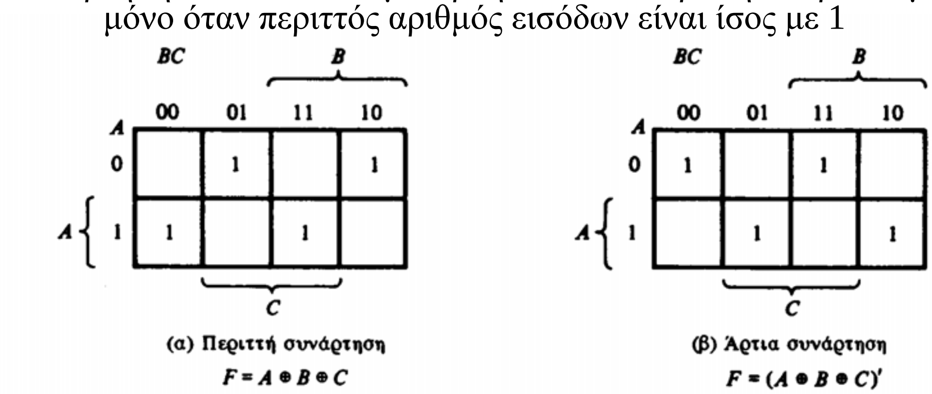
$$F=[(C+D)B'+A](B+C')=(A+C+D)(A+B')(B+C')$$

Η Συνάρτηση Αποκλειστικό Ή

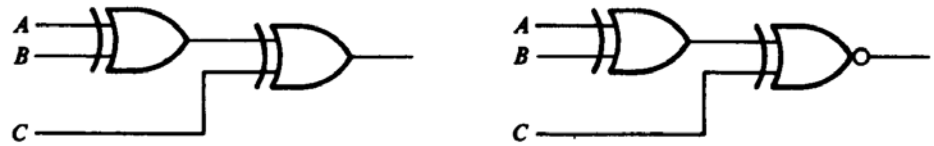
Αποκλειστικό Ή (XOR) $x\oplus y=x'y+xy'$ *Σχ. Αντιστρ.* ⇔ Αποκλειστικό ΟΥΤΕ (XNOR) $(x\oplus y)'=xy+x'y'$

- Είναι Αντιμεταθετική & Προσεταιριστική
- Δεν φτιάχνονται συχνά πύλες XOR > 2 εισόδους

Η Συνάρτηση XOR πολλών μεταβλητών είναι περιττή: παίρνει τιμή 1 μόνο όταν περιττός αριθμός εισόδων είναι ίσος με 1



Η Συνάρτηση Αποκλειστικό Ή



(α) Περιττή συνάρτηση τριών εισόδων

(β) Άρτια συνάρτηση τριών εισόδων

ΣΧΗΜΑ 4-23
Λογικά διαγράμματα περιττών και άρτιων συναρτήσεων

(α) Περιττή συνάρτηση

$F = A \oplus B \oplus C \oplus D$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

(β) Άρτια συνάρτηση

$F = (A \oplus B \oplus C \oplus D)'$

AB \ CD	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

Γεννήτρια & Ελεγκτής Ισοτιμίας

ΠΙΝΑΚΑΣ 4-4
Πίνακας Αληθείας για τη Γεννήτρια Άρτιας Ισοτιμίας

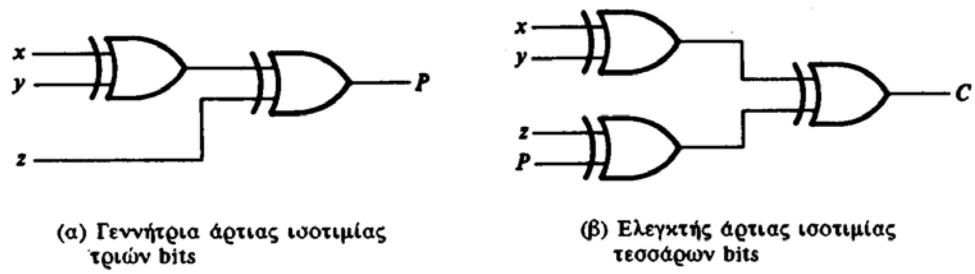
Μήνυμα τριών bits			Bit Ισοτιμίας
x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

ΠΙΝΑΚΑΣ 4-5
Ελεγκτής Άρτιας Ισοτιμίας

Τέσσερα Bits Δέκτη				Έλεγχος Λάθους Ισοτιμίας
x	y	z	P	C
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
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	1
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	0

Γεννήτρια & Ελεγκτής Ισοτιμίας

Τα κυκλώματα αυτά χρησιμοποιούνται στην ανίχνευση λαθών κατά την μετάδοση ή λειτουργία των κυκλωμάτων



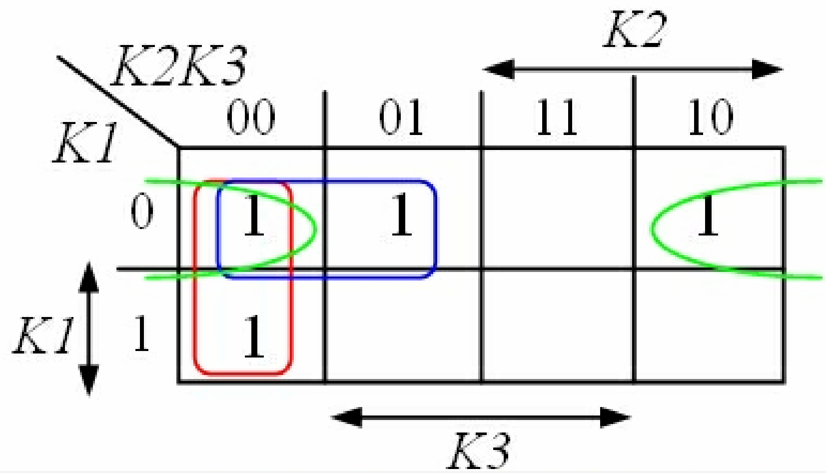
ΣΧΗΜΑ 4-25
Λογικό διάγραμμα μιας γεννήτριας και ενός ελεγκτή ισοτιμίας

Το bit ισοτιμίας είναι περιττή πληροφορία η οποία όμως μπορεί να χρησιμοποιηθεί για την ανίχνευση μονού αριθμού λαθών.

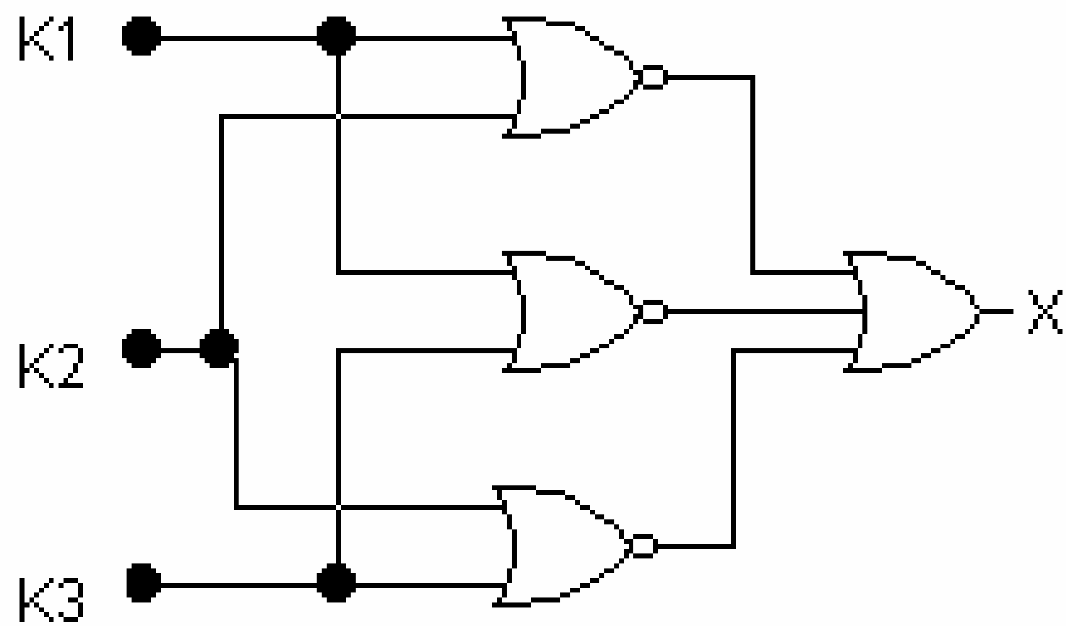
ΑΣΚΗΣΗ

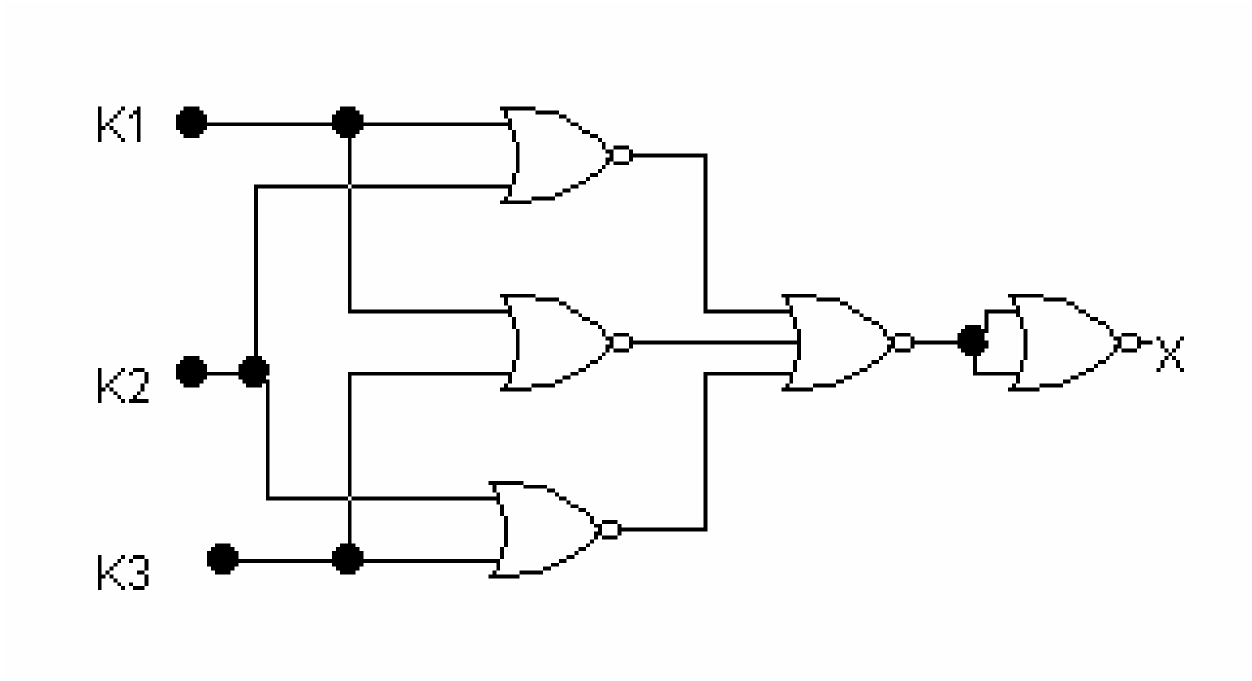
Στην άρση βαρών μια προσπάθεια ενός αθλητή θεωρείται έγκυρη όταν τη δέχονται τουλάχιστον δύο από τους τρεις κριτές (K1,K2,K3) και άκυρη διαφορετικά. Σχεδιάστε συνδυαστικό κύκλωμα, χρησιμοποιώντας μόνο πύλες NOR, το οποίο να δίνει έξοδο (X) ίση με λογικό ‘1’ όταν μια προσπάθεια θεωρείται <u>άκυρη</u>.
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'





Άσκηση

- 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.

- **Α**. Αν το σύστημα ασφαλείας υλοποιείται με ένα συνδυαστικό κύκλωμα, καθορίστε τις εισόδους και τις εξόδους του και κατασκευάστε τον πίνακα αλήθειας.
- **Β**. Απλοποιείστε το κύκλωμα χρησιμοποιώντας χάρτες Karnaugh και σχεδιάστε το με λογικές πύλες.

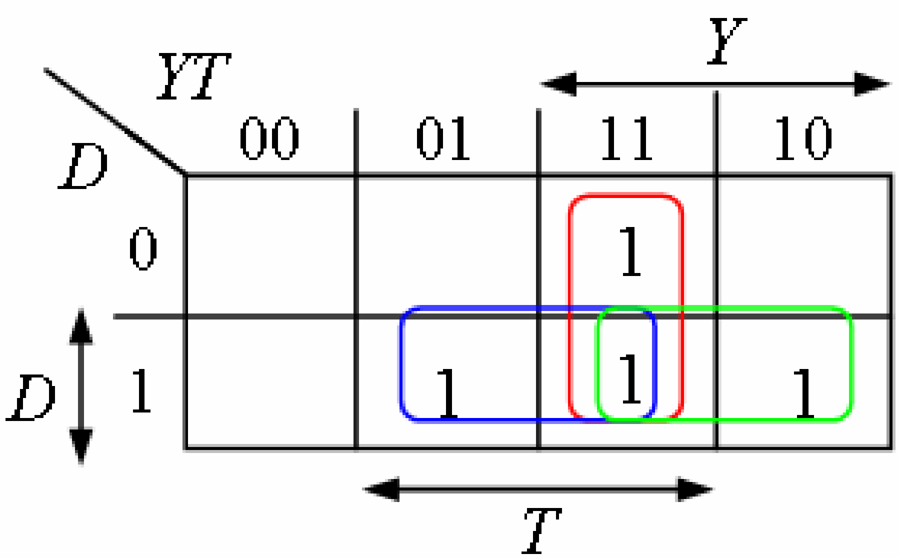
Λύση

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:

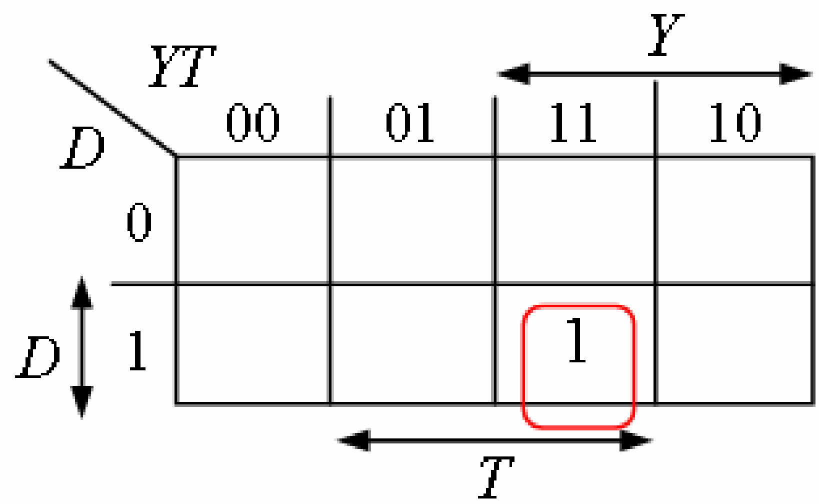
D	Y	T	X	S
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$X = \Sigma(3, 5, 6, 7)$$

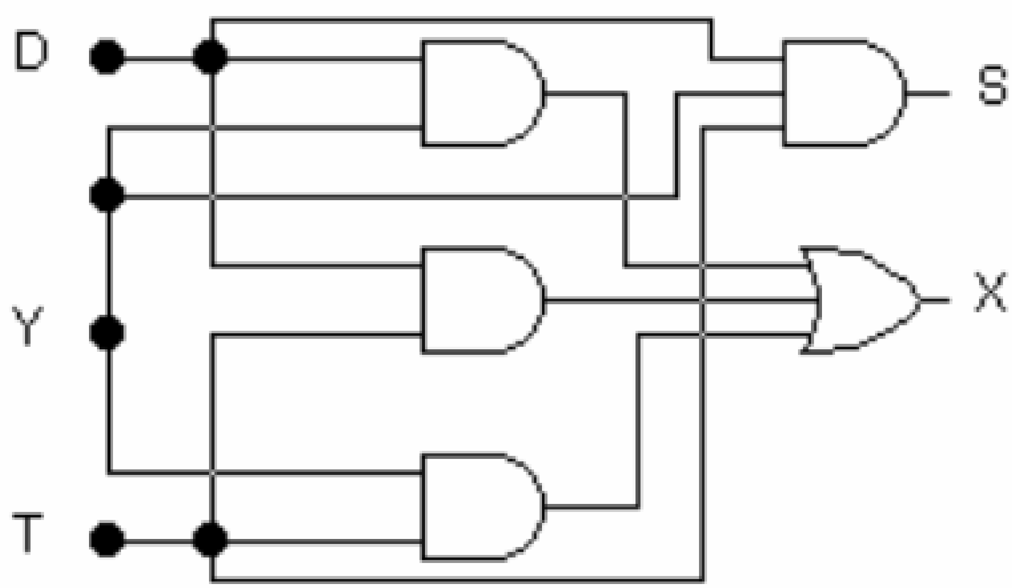
$$S = \Sigma(7)$$



$$X = YT + DT + DY$$

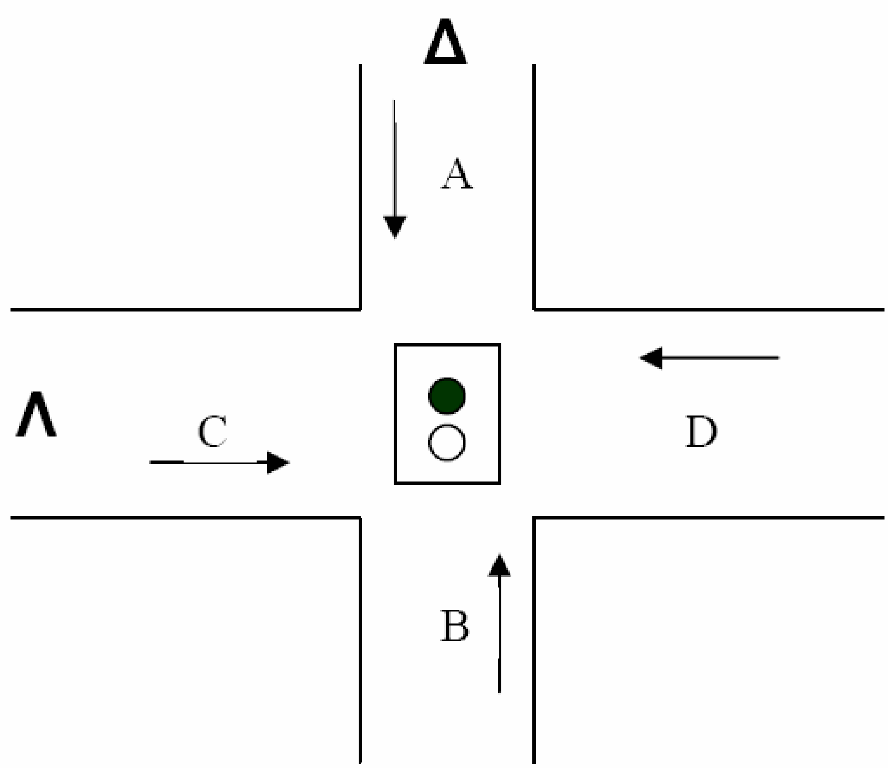


$$S = DYT$$



Άσκηση

Το παρακάτω σχήμα δείχνει τη διασταύρωση μιας λεωφόρου Λ με έναν δρόμο Δ. Στις γραμμές A, B, C, D έχουν τοποθετηθεί ανιχνευτές οχημάτων που δίνουν έξοδο '0', όταν δεν είναι κατειλημμένες για 50 μέτρα πριν τη διασταύρωση και '1', όταν συμβαίνει το αντίθετο.



Το σύστημα ελέγχου της κυκλοφορίας λειτουργεί με τον ακόλουθο τρόπο:
Το φανάρι που επιτρέπει την κυκλοφορία στη λεωφόρο Λ γίνεται πράσινο και το φανάρι στο δρόμο Δ γίνεται κόκκινο, όταν συμβαίνει τουλάχιστον μία από τις παρακάτω περιπτώσεις:

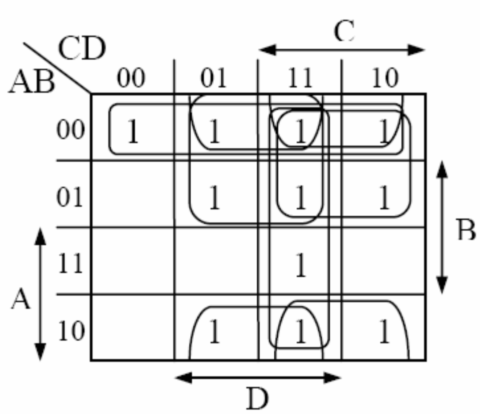
- Και οι δύο γραμμές C και D είναι κατειλημμένες.
- Δεν υπάρχουν οχήματα στις γραμμές A,B,C και D.
- Μία τουλάχιστον από τις γραμμές C,D είναι κατειλημμένη, αλλά οι γραμμές A,B δεν είναι και οι δύο κατειλημμένες.
- Σε οποιαδήποτε άλλη περίπτωση το φανάρι που επιτρέπει την κυκλοφορία στη λεωφόρο Λ είναι κόκκινο και το φανάρι που επιτρέπει την κυκλοφορία στο δρόμο Δ είναι πράσινο.

Με βάση την παραπάνω λογική να καταστρωθεί ο πίνακας αλήθειας και ο αντίστοιχος χάρτης Karnaugh.
Να σχεδιαστεί λογικό κύκλωμα με την ελάχιστη υλοποίηση του προβλήματος.

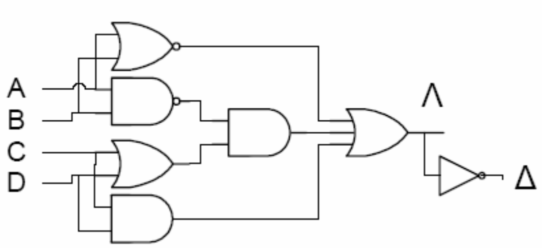
Για τις μεταβλητές A, B, C, D ορίζουμε το λογικό 1 να αντιπροσωπεύει την κατάσταση «γραμμή κατειλημμένη». Επίσης, θεωρούμε ότι Λ=1 συμβολίζει το πράσινο φανάρι και Λ=0 το κόκκινο. Η συνάρτηση Δ είναι η συμπληρωματική της Λ, δηλ. Δ=Λ’.

A	B	C	D	Λ	Δ
0	0	0	0	**1**	**0**
0	0	0	1	**1**	**0**
0	0	1	0	**1**	**0**
0	0	1	1	**1**	**0**
0	1	0	0	**0**	**1**
0	1	0	1	**1**	**0**
0	1	1	0	**1**	**0**
0	1	1	1	**1**	**0**
1	0	0	0	**0**	**1**
1	0	0	1	**1**	**0**
1	0	1	0	**1**	**0**
1	0	1	1	**1**	**0**
1	1	0	0	**0**	**1**
1	1	0	1	**0**	**1**
1	1	1	0	**0**	**1**
1	1	1	1	**1**	**0**

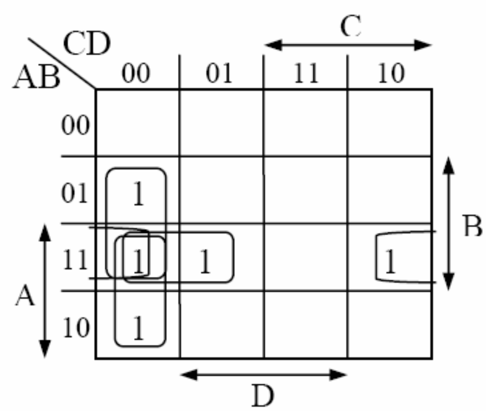
1ος ΤΡΟΠΟΣ



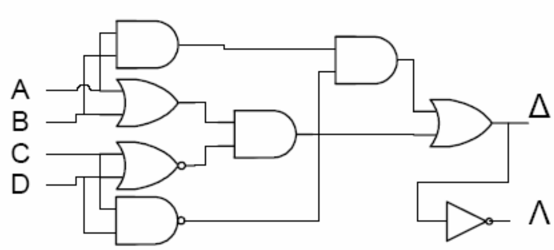
$$\mathbf{\Lambda} = A'B'+CD+A'C+A'D+B'C+B'D =$$
$$= (A+B)' + CD + (A'+B')(C+D) =$$
$$= (A+B)' + CD + (AB)'(C+D)$$



2ος ΤΡΟΠΟΣ



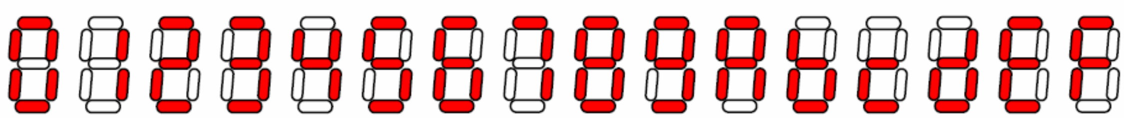
$$\mathbf{\Delta} = BC'D'+AC'D'+ABC'+ABD' =$$
$$= (A+B)(C'D') + (AB)(C'+D') =$$
$$= (A+B)(C+D)' + (AB)(CD)'$$

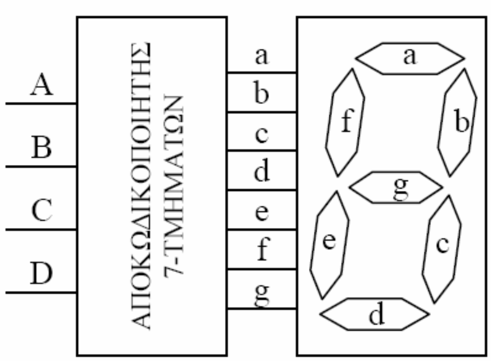


Άσκηση

Να σχεδιάσετε έναν αποκωδικοποιητή «HEX σε απεικόνιση 7-τμημάτων» (7-segment decoder). Χρησιμοποιήστε χάρτες Karnaugh για την απλοποίηση των 7 λογικών συναρτήσεων *a, b, c, d, e, f, g*.

Η απεικόνιση των αριθμών του δεκαεξαδικού συστήματος (0-F) στο display θα γίνεται ως εξής:



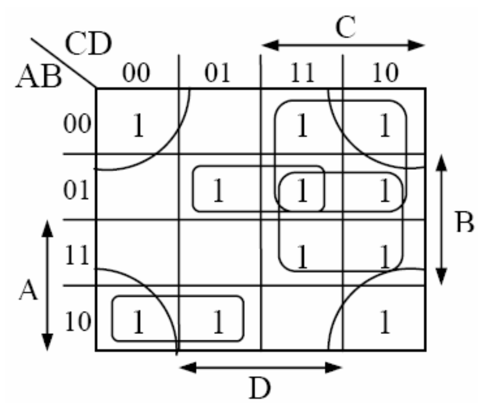


[0]={a,b,c,d,e,f}	[8]={a,b,c,d,e,f,g}
[1]={b,c}	[9]={a,b,c,d,f,g}
[2]={a,b,d,e,g}	[A]={a,b,c,e,f,g}
[3]={a,b,c,d,g}	[B]={c,d,e,f,g}
[4]={b,c,f,g}	[C]={d,e,g}
[5]={a,c,d,f,g}	[D]={b,c,d,e,g}
[6]={a,c,d,e,f,g}	[E]={a,d,e,f,g}
[7]={a,b,c}	[F]={a,e,f,g}

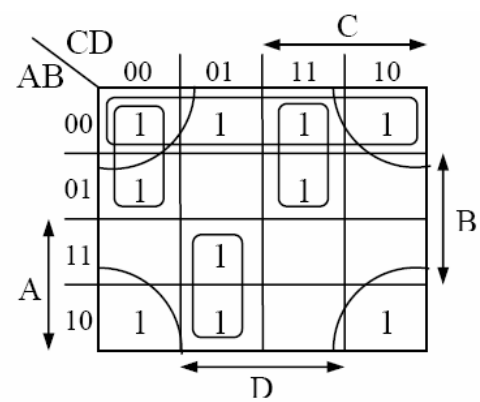
Να γράψετε όλες τις απλοποιημένες συναρτήσεις και να σχεδιάσετε τα λογικά κυκλώματα μόνο για τις συναρτήσεις *f* και *g*. Θεωρείστε ότι οι 4 είσοδοι του αποκωδικοποιητή είναι A, B, C, D (όπου A είναι το ΠΣΨ). Καθεμιά από τις διόδους led *a, b, c, d, e, f, g,* του display φωτοβολεί όταν η αντίστοιχη είσοδός της βρίσκεται σε λογικό «1».

Θεωρώντας ότι η δεκαεξαδική τιμή σχηματίζεται από τις μεταβλητές A, B, C, D, (όπου A είναι το ΠΣΨ), συμπληρώνουμε τον πίνακα αλήθειας των 7 συναρτήσεων a, b, c, d, e, f, g:

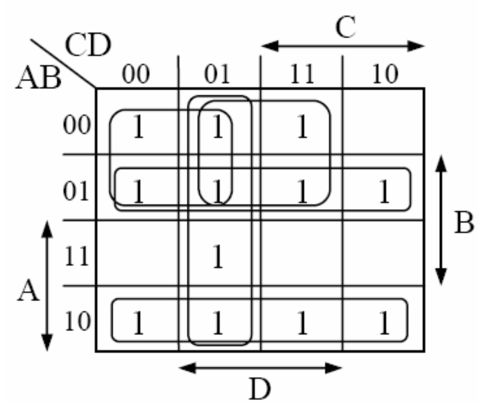
A	B	C	D	Αριθμός	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	0	1	0	A	1	1	1	0	1	1	1
1	0	1	1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



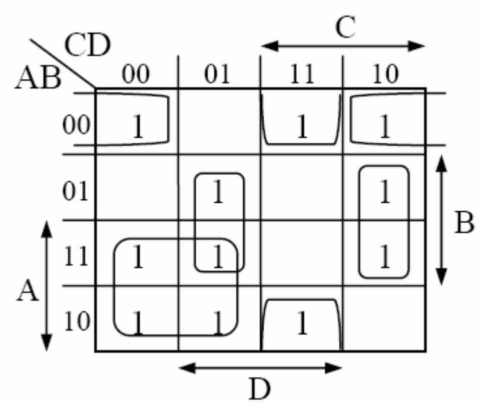
a = A’C+BC+B’D’+A’BD+AB’C’



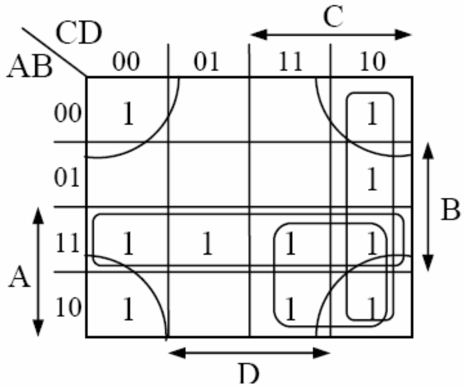
b = A’B’+B’D’+A’C’D’+A’CD+AC’D



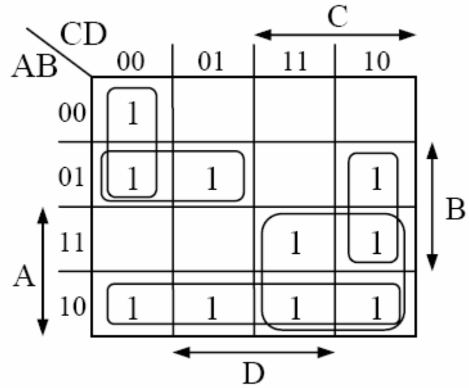
c = A’C’+A’D+A’B+AB’+C’D



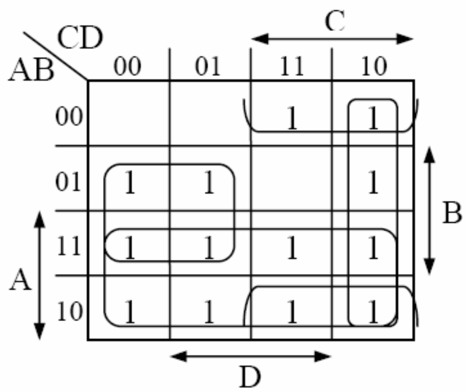
d = AC’+BC’D+BCD’+A’B’D’+B’CD



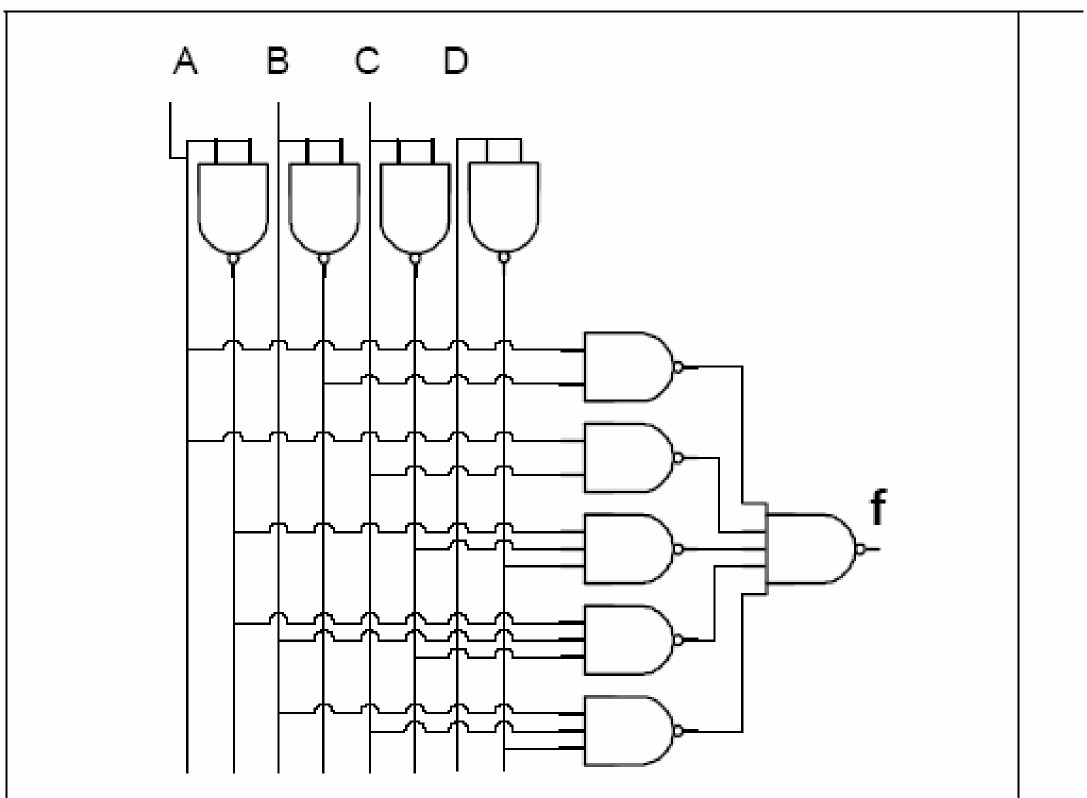
e = AB+AC+B'D'+CD'



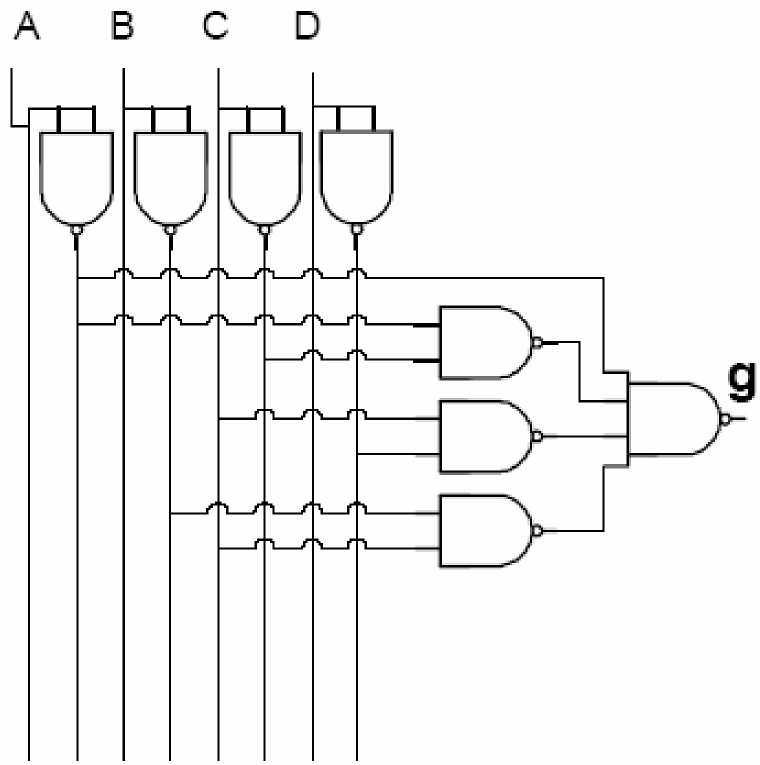
f = AB'+AC+A'C'D'+A'BC'+BCD'



g = A+BC'+CD'+B'C



f = AB'+AC+A'C'D'+A'BC'+BCD'



$$g = A+BC'+CD'+B'C$$