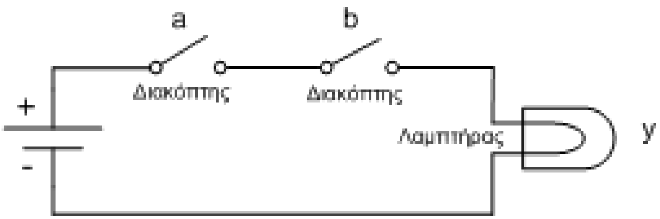
Ψηφιακά Συστήματα - Εισαγωγή

Κυκλώματα Διακοπτών και Λογικές Πύλες



Κύκλωμα

a	b	y
OPEN	OPEN	OFF
OPEN	CLOSED	OFF
CLOSED	OPEN	OFF
CLOSED	CLOSED	ON

Πίνακας Αλήθειας

(OPEN → FALSE - F)
(CLOSED → TRUE - T)

(OFF → FALSE - F)
(ON → TRUE - T)

a	b	y
F	F	F
F	T	F
T	F	F
T	T	T

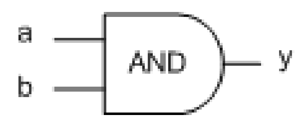
Πίνακας Αλήθειας

TRUE - T → 1
FALSE - F → 0

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας Αλήθειας

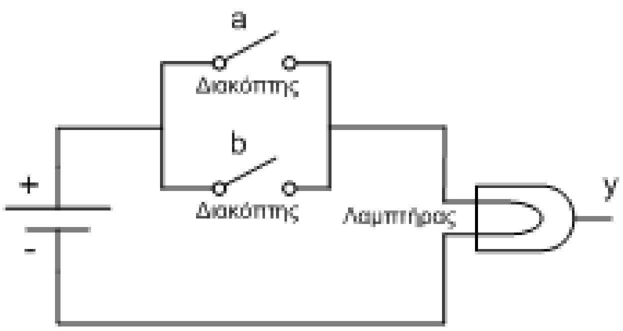
Πύλη AND



Λογική Πράξη AND

$y = a \text{ AND } b = a \cdot b = ab$

Κυκλώματα Διακοπτών και Λογικές Πύλες



Κύκλωμα

a	b	y
OPEN	OPEN	OFF
OPEN	CLOSED	ON
CLOSED	OPEN	ON
CLOSED	CLOSED	ON

Πίνακας Αλήθειας

(OPEN → FALSE - F)
(CLOSED → TRUE - T)

(OFF → FALSE - F)
(ON → TRUE - T)

a	b	y
F	F	F
F	T	T
T	F	T
T	T	T

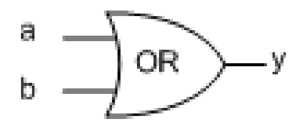
Πίνακας Αλήθειας

TRUE - T → 1
FALSE - F → 0

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

Πίνακας Αλήθειας

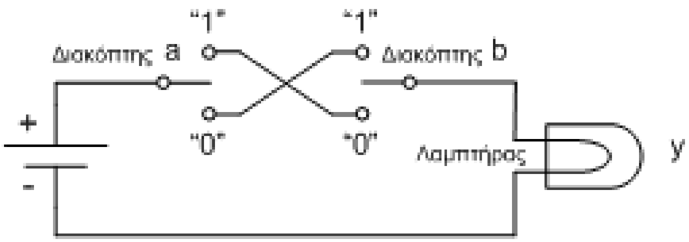
Πύλη OR



Λογική Πράξη OR

y = a OR b = a+b

Κυκλώματα Διακοπτών και Λογικές Πύλες



Κύκλωμα

a	b	y
OPEN	OPEN	OFF
OPEN	CLOSED	ON
CLOSED	OPEN	ON
CLOSED	CLOSED	OFF

Πίνακας Αλήθειας

(OPEN → FALSE - F)
(CLOSED → TRUE - T)

(OFF → FALSE - F)
(ON → TRUE - T)

a	b	y
F	F	F
F	T	T
T	F	T
T	T	F

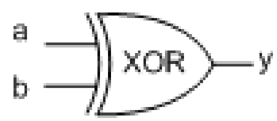
Πίνακας Αλήθειας

TRUE - T → 1
FALSE - F → 0

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας Αλήθειας

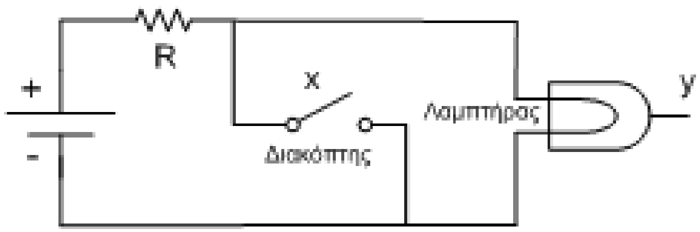
Πύλη XOR



Λογική Πράξη XOR

$y = a \text{ XOR } b = a \oplus b$

Κυκλώματα Διακοπτών και Λογικές Πύλες



Κύκλωμα

x	y
OPEN	ON
CLOSED	OFF

Πίνακας Αλήθειας

(OPEN → FALSE - F)
(CLOSED → TRUE - T)

(OFF → FALSE - F)
(ON → TRUE - T)

b	y
F	T
T	F

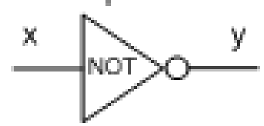
Πίνακας Αλήθειας

TRUE - T → 1
FALSE - F → 0

x	y
0	1
1	0

Πίνακας Αλήθειας

Πύλη NOT



Λογική Πράξη NOT

$y = \text{NOT}(x) = x' = \bar{x}$

Λογικές Πύλες – Πίνακες Αλήθειας – Λογικές Πράξεις

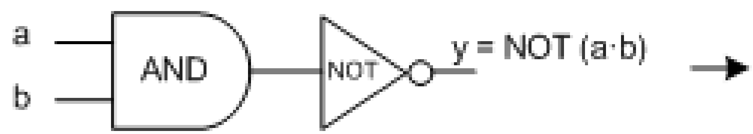
Λογική Πύλη AND

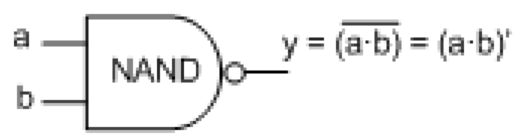
a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

Πίνακας Αλήθειας

Λογική Πράξη AND

$y = a \text{ AND } b = a \cdot b = ab$





Λογική Πύλη NAND

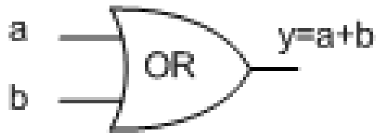
a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

Πίνακας Αλήθειας

Λογική Πράξη NAND

$y = \text{NOT}(a \text{ AND } b) = (a\cdot b)' = (ab)' = \overline{(ab)}$

Λογικές Πύλες – Πίνακες Αλήθειας – Λογικές Πράξεις



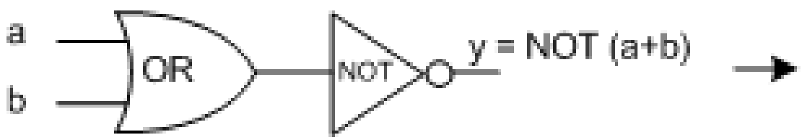
Λογική Πύλη OR

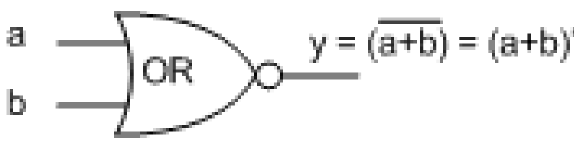
a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

Πίνακας Αλήθειας

Λογική Πράξη OR

$y = a \text{ OR } b = a+b$





Λογική Πύλη NOR

a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

Πίνακας Αλήθειας

Λογική Πράξη NOR

$y = \text{NOT}(a \text{ OR } b) = (a+b)' = \overline{(a+b)}$

Λογικές Πύλες – Πίνακες Αλήθειας – Λογικές Πράξεις

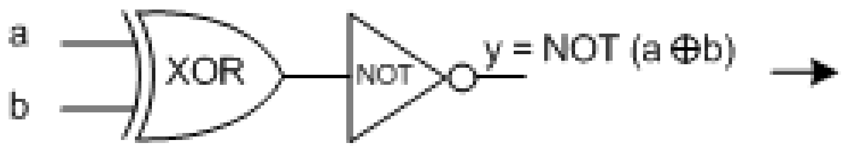
Λογική Πύλη NOR

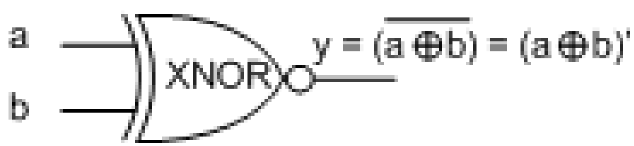
a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας Αλήθειας

Λογική Πράξη XOR

$y = a \text{ XOR } b = a \oplus b$





Λογική Πύλη XNOR

a	b	y
0	0	1
0	1	0
1	0	0
1	1	1

Πίνακας Αλήθειας

Λογική Πράξη XNOR

$y = \text{NOT}(a \text{ XOR } b) = (a \oplus b)' = \overline{(a \oplus b)}$

Συγκεντρωτικοί Πίνακες Αλήθειας

ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ ΔΥΟ ΜΕΤΑΒΛΗΤΩΝ

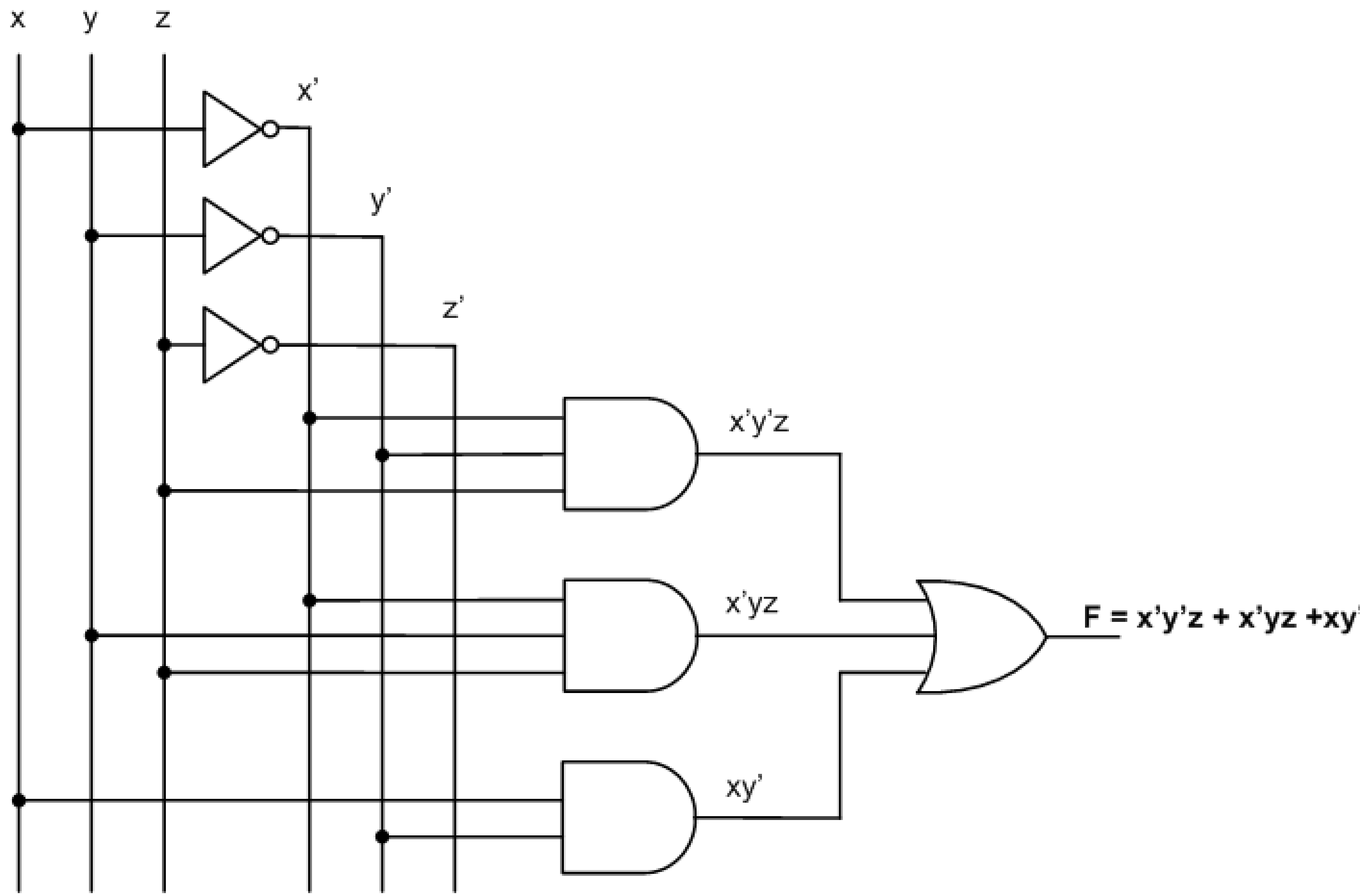
X	Y	X•Y	(X•Y)’	X+Y	(X+Y)’	X ⊕ Y	(X ⊕ Y)’
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ ΤΡΙΩΝ ΜΕΤΑΒΛΗΤΩΝ

X	Y	Z	X•Y•Z	(X•Y•Z)’	X+Y+Z	(X+Y+Z)’	X ⊕ Y ⊕ Z	(X ⊕ Y ⊕ Z)’
0	0	0	0	1	0	1	0	1
0	0	1	0	1	1	0	1	0
0	1	0	0	1	1	0	1	0
0	1	1	0	1	1	0	0	1
1	0	0	0	1	1	0	1	0
1	0	1	0	1	1	0	0	1
1	1	0	0	1	1	0	0	1
1	1	1	1	0	1	0	1	0

Λογική Συνάρτηση και Λογικό Κύκλωμα

F = x'y'z + x'yz +xy'



Άλγεβρα Boole

Αξιώματα, Θεωρήματα και Ιδιότητες της Άλγεβρας Boole	
$0 \cdot 0 = 0$	$1 + 1 = 1$
$1 \cdot 1 = 1$	$0 + 0 = 0$
$0 \cdot 1 = 0$	$1 + 0 = 1$
Εάν $x = 0$, τότε $x' = 1$	Εάν $x = 1$, τότε $x' = 0$
$x \cdot 1 = x$	$x + 0 = x$
$x \cdot x' = 0$	$x + x' = 1$
$x \cdot y = y \cdot x$	$x + y = y + x$
$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + (y \cdot z) = (x + y)\cdot(x + z)$
$x \cdot x = x$	$x + x = x$
$x \cdot 0 = 0$	$x + 1 = 1$
$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
$(x')' = x$	
$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$
$x + x \cdot y = x$	$x \cdot (x + y) = x$
$x \cdot y + x \cdot y' = x$	$(x + y) \cdot (x + y') = x$
$x \cdot (x' + y) = x \cdot y$	$x + x' \cdot y = x + y$
Αρχή του δυϊσμού: Κάθε αλγεβρική σχέση που μπορεί να προκύψει από τα αξιώματα της άλγεβρας Boole παραμένει αληθής, αν οι τελεστές (AND, OR) και τα ουδέτερα στοιχεία (‘1’ και ‘0’ αντίστοιχα) εναλλαγούν (δηλ. τα AND να γίνουν OR, τα OR να γίνουν AND, τα ‘1’ να γίνουν ‘0’ και τα ‘0’ να γίνουν ‘1’), όπως φαίνεται στις δυο στήλες του πίνακα.	

Ελαχιστόροι (minterms) και Μεγιστόροι (Maxterms)

Σε μια συνάρτηση *n* μεταβλητών, ένα λογικό γινόμενο (πράξη AND) που περιλαμβάνει όλες τις μεταβλητές στην κανονική ή την συμπληρωματική τους μορφή, ονομάζεται **ελαχιστόρος** (**minterm**). Έτσι, σε κάθε γραμμή ενός πίνακα αλήθειας, δηλ. για κάθε δυνατό συνδυασμό τιμών των ανεξάρτητων μεταβλητών (εισόδων του συστήματος), αντιστοιχεί ένας ελαχιστόρος που σχηματίζεται από το λογικό γινόμενο των ανεξάρτητων μεταβλητών, θέτοντας τη μεταβλητή σε κανονική μορφή αν σε εκείνη τη γραμμή έχει τιμή 1 και σε συμπληρωματική μορφή αν έχει τιμή 0.

		Ελαχιστόροι	
x	**y**	***Όρος***	***Ονομασία***
0	0	x’y’	m_0
0	1	x’y	m_1
1	0	xy’	m_2
1	1	xy	m_3

			Ελαχιστόροι	
x	**y**	**z**	***Όρος***	***Ονομασία***
0	0	0	x’y’z’	m_0
0	0	1	x’y’z	m_1
0	1	0	x’yz’	m_2
0	1	1	x’yz	m_3
1	0	0	xy’z’	m_4
1	0	1	xy’z	m_5
1	1	0	xyz’	m_6
1	1	1	xyz	m_7

Ελαχιστόροι (minterms) και Μεγιστόροι (Maxterms)

Σε μια συνάρτηση *n* μεταβλητών, ένα λογικό άθροισμα (πράξη OR) που περιλαμβάνει όλες τις μεταβλητές στην κανονική ή την συμπληρωματική τους μορφή, ονομάζεται **μεγιστόρος** (**Maxterm**). Έτσι, σε κάθε γραμμή ενός πίνακα αλήθειας, δηλ. για κάθε δυνατό συνδυασμό τιμών των ανεξάρτητων μεταβλητών (εισόδων του συστήματος), αντιστοιχεί ένας μεγιστόρος που σχηματίζεται από το λογικό άθροισμα των ανεξάρτητων μεταβλητών, θέτοντας τη μεταβλητή σε συμπληρωματική μορφή αν σε εκείνη τη γραμμή έχει τιμή 1 και σε κανονική μορφή αν έχει τιμή 0.

		Μεγιστόροι	
x	**y**	***Όρος***	***Ονομασία***
0	0	x+y	M_0
0	1	x+y’	M_1
1	0	x’+y	M_2
1	1	x’+y’	M_3

			Μεγιστόροι	
x	**y**	**z**	***Όρος***	***Ονομασία***
0	0	0	x+y+z	M_0
0	0	1	x+y+z’	M_1
0	1	0	x+y’+z	M_2
0	1	1	x+y’+z’	M_3
1	0	0	x’+y+z	M_4
1	0	1	x’+y+z’	M_5
1	1	0	x’+y’+z	M_6
1	1	1	x’+y’+z’	M_7

Ελαχιστόροι (minterms) και Μεγιστόροι (Maxterms)

		Ελαχιστόροι		**Μεγιστόροι**	
x	**y**	***Όρος***	***Ονομασία***	***Όρος***	***Ονομασία***
0	0	x'y'	m_0	x+y	M_0
0	1	x'y	m_1	x+y'	M_1
1	0	xy'	m_2	x'+y	M_2
1	1	xy	m_3	x'+y'	M_3

			Ελαχιστόροι		**Μεγιστόροι**	
x	**y**	**z**	***Όρος***	***Ονομασία***	***Όρος***	***Ονομασία***
0	0	0	x'y'z'	m_0	x+y+z	M_0
0	0	1	x'y'z	m_1	x+y+z'	M_1
0	1	0	x'yz'	m_2	x+y'+z	M_2
0	1	1	x'yz	m_3	x+y'+z'	M_3
1	0	0	xy'z'	m_4	x'+y+z	M_4
1	0	1	xy'z	m_5	x'+y+z'	M_5
1	1	0	xyz'	m_6	x'+y'+z	M_6
1	1	1	xyz	m_7	x'+y'+z'	M_7

Ελαχιστόροι (minterms) και Μεγιστόροι (Maxterms)

Αν έχουμε τον πίνακα αλήθειας μιας λογικής συνάρτησης F μπορούμε να γράψουμε τη λογική συνάρτηση:

- Ως λογικό **άθροισμα** (πράξη OR) εκείνων των **ελαχιστόρων** για τους οποίους στον πίνακα αλήθειας η F έχει τιμή 1.
- Ως λογικό **γινόμενο** (πράξη AND) εκείνων των **μεγιστόρων** για τους οποίους στον πίνακα αλήθειας η F έχει τιμή 0.

Παράδειγμα:

Ο πίνακας αλήθειας της συνάρτησης XOR δύο μεταβλητών είναι ο ακόλουθος:

X	Y	$F = X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

Η συνάρτηση είναι: $F = x \oplus y = x'y + xy' = m_1 + m_2 = \Sigma(1, 2)$

Επίσης: $F' = x'y' + xy$

Αλλά: $F = (F')' = (x'y' + xy)' = (x'' + y'')\bullet(x' + y') = (x + y)\bullet(x' + y') = M_0\bullet M_4 = \Pi(0, 4)$

Κανονικές Μορφές Λογικών Συναρτήσεων
(“Άθροισμα” Ελαχιστόρων και “Γινόμενο” Μεγιστόρων)

Δίνεται ο πίνακας αλήθειας της συνάρτησης F.

				Ελαχιστόροι		Μεγιστόροι	
x	**y**	**z**	**F**	***Όρος***	***Ονομασία***	***Όρος***	***Ονομασία***
0	0	0	**0**	x’y’z’	m_0	**x+y+z**	$\mathbf{M_0}$
0	0	1	**0**	x’y’z	m_1	**x+y+z’**	$\mathbf{M_1}$
0	1	0	**0**	x’yz’	m_2	**x+y’+z**	$\mathbf{M_2}$
0	1	1	**1**	**x’yz**	$\mathbf{m_3}$	x+y’+z’	M_3
1	0	0	**0**	xy’z’	m_4	**x’+y+z**	$\mathbf{M_4}$
1	0	1	**1**	**xy’z**	$\mathbf{m_5}$	x’+y+z’	M_5
1	1	0	**1**	**xyz’**	$\mathbf{m_6}$	x’+y’+z	M_6
1	1	1	1	**xyz**	$\mathbf{m_7}$	x’+y’+z’	M_7

Η F μπορεί να εκφραστεί σε ΚΑΝΟΝΙΚΗ μορφή ως **Άθροισμα Ελαχιστόρων**:

F = $x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7 = \Sigma(3, 5, 6, 7)$

ή ως **Γινόμενο Μεγιστόρων:**

F = $(x+y+z)\cdot(x+y+z')\cdot(x+y'+z)\cdot(x'+y+z) = M_0\cdot M_1\cdot M_2\cdot M_4 = \Pi(0, 1, 2, 4)$

Έκφραση Λογικής Συνάρτησης σε Κανονική Μορφή (ως άθροισμα ελαχιστόρων)

Δίνεται η συνάρτηση:

$$F = x \cdot y + x \cdot z + y \cdot z$$

Να εκφραστεί η F σε κανονική μορφή ως άθροισμα ελαχιστόρων και να συμπληρωθεί ο πίνακας αλήθειας.

$$F = x \cdot y \cdot 1 + x \cdot 1 \cdot z + 1 \cdot y \cdot z = x \cdot y \cdot (z + z') + x \cdot (y + y') \cdot z + (x + x') \cdot y \cdot z =$$

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

$$= \mathbf{m_7 + m_6 + m_5 + m_3}$$

x	y	z	F		
0	0	0	0		
0	0	1	0		
0	1	0	0		
0	1	1	**1**	**x’yz**	**m_3**
1	0	0	0		
1	0	1	**1**	**xy’z**	**m_5**
1	1	0	**1**	**xyz’**	**m_6**
1	1	1	**1**	**xyz**	**m_7**

Απλοποίηση Λογικών Συναρτήσεων

Απλοποίηση με άλγεβρα Boole:

1. $x + x'y = (x+x')(x + y) = 1\cdot(x+y) = x + y$

2. $x'y'z + x'yz + xy' = x'z(y' + y) + xy' = x'z\cdot 1 + xy' = x'z + xy'$

3. $xy + x'z + yz = xy + x'z + 1\cdot yz = xy + x'z + (x + x')yz$

$= xy + x'z + xyz + x'yz = xy(1 + z) + x'z(1 + y) = xy + x'z$

4. $(x + y)(x' + z)(y + z) = (x + y)(x' + z)$ [λόγω δυϊσμού από τη συνάρτηση 3]

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

Για να κάνουμε απλοποίηση μιας λογικής συνάρτησης με πίνακα (ή χάρτη) Karnaugh ακολουθούμε τα παρακάτω βήματα:

1. Η λογική συνάρτηση θα πρέπει να είναι σε πλήρη μορφή, ως λογικό άθροισμα ελαχιστόρων ή λογικό γινόμενο μεγιστόρων. Συνήθως χρησιμοποιούμε την μορφή του αθροίσματος ελαχιστόρων. Αν η συνάρτηση δεν είναι σε πλήρη μορφή θα πρέπει να τη φέρουμε σε τέτοια μορφή, χρησιμοποιώντας τις ιδιότητες της άλγεβρας Boole. Ειδικότερα για συνάρτηση σε μορφή λογικού αθροίσματος, που συνήθως αντιμετωπίζουμε, για τους όρους που δεν είναι πλήρεις (ελαχιστόροι) τους μετασχηματίζουμε εφαρμόζοντας τις ιδιότητες: A·1=A και B+B'=1.

Παράδειγμα: Έστω ότι έχουμε τη συνάρτηση F(A,B,C) = AB + ABC' + A'B'C'. O πρώτος όρος, AB, δεν είναι πλήρης (ελαχιστόρος). Όμως μπορούμε να τον μετασχηματίσουμε ως εξής: AB = AB·1 = AB(C+C') = ABC + ABC'. Έτσι η συνάρτηση F σε πλήρη μορφή γίνεται: F=ABC+ABC'+ABC'+A'B'C'.

Και επειδή ισχύει: A+A+…+A=A, η τελική μορφή της συνάρτησης είναι: F=ABC + ABC' + A'B'C'.

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

2. Ο πίνακας Karnaugh έχει διαστάσεις ανάλογες με το πλήθος των ανεξάρτητων μεταβλητών. Έτσι, για δυο ανεξάρτητες μεταβλητές έχουμε πίνακα δυο διαστάσεων που περιέχει $2^2 = 4$ κυψέλες, όσοι δηλαδή και οι ελαχιστόροι. Αντίστοιχα, για τρεις ανεξάρτητες μεταβλητές έχουμε πίνακα τριών διαστάσεων, που περιέχουν $2^3 = 8$ κυψέλες, αφού θα έχουμε 8 ελαχιστόρους και για τέσσερις μεταβλητές θα έχουμε πίνακα τεσσάρων διαστάσεων με 16 ελαχιστόρους και επομένως $2^4 = 16$ κυψέλες. Κάθε μια κυψέλη αντιστοιχεί σε ένα συγκεκριμένο ελαχιστόρο, που κι αυτός αντιστοιχεί σε ένα συγκεκριμένο συνδυασμό τιμών των ανεξάρτητων μεταβλητών.

3. Η αρίθμηση των γραμμών και των στηλών ακολουθεί τον κώδικα GRAY: 00, 01, 11, 10

	y	
x	0	1
0	m0	m1
1	m2	m3

F(x,y)

	yz			
x	00	01	11	10
0	m0	m1	m3	m2
1	m4	m5	m7	m6

G(x,y,z)

	zw			
xy	00	01	11	10
00	m0	m1	m3	m2
01	m4	m5	m7	m6
11	m12	m13	m15	m14
10	m8	m9	m11	m10

R(x,y,z,w)

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

4. Η συμπλήρωση του πίνακα γίνεται με βάση είτε τη λογική συνάρτηση (σε πλήρη μορφή), είτε του πίνακα αλήθειας της συνάρτησης. Και τα δυο είναι ισοδύναμα: αν έχουμε τη συνάρτηση σε μορφή αθροίσματος ελαχιστόρων, κάθε ένας ελαχιστόρος αντιστοιχεί σε ένα “1” στον πίνακα αλήθειας και αντίστροφα, κάθε ένα “1” του πίνακα αλήθειας, αντιστοιχεί σε έναν ελαχιστόρο στη λογική συνάρτηση.

   Επομένως αφού κάθε κυψέλη του πίνακα αντιστοιχεί σε ένα συγκεκριμένο ελαχιστόρο, μεταφέρουμε από τον πίνακα αλήθειας τις τιμές (“0” ή “1”) που έχει η συνάρτηση για κάθε ένα ελαχιστόρο, στην αντίστοιχη κυψέλη.

   Από τη συνάρτηση μπορούμε να συμπληρώσουμε κατευθείαν τον πίνακα βάζοντας ένα “1” για κάθε ελαχιστόρο που υπάρχει στη συνάρτηση, στην αντίστοιχη κυψέλη του πίνακα. Οι υπόλοιπες κυψέλες του πίνακα παίρνουν τιμή “0”. Μας συμφέρει να συμπληρώσουμε τον πίνακα με βάση την κατάσταση που εμφανίζεται λιγότερο συχνά. Αν π.χ. στον πίνακα αλήθειας εμφανίζονται μόνο δυο “0”, τοποθετούμε πρώτα αυτά στις κατάλληλες θέσεις του πίνακα και μετά συμπληρώνουμε τις υπόλοιπες θέσεις με “1”, επιταχύνοντας έτσι τη διαδικασία.

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

Παράδειγμα: Έστω ότι έχουμε τη συνάρτηση:

F(A,B,C) = Σ(3, 5, 6, 7) = m3 + m5 + m6 + m7 = A'BC + AB'C + ABC' + ABC.

Ο πίνακας αλήθειας και ο πίνακας Karnaugh της συνάρτησης είναι:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

F(A,B,C)

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

5. Το επόμενο στάδιο είναι **να ομαδοποιήσουμε τα “1”**. Αυτό σημαίνει να φτιάξουμε ομάδες από γειτονικά “1”. Για να είναι δυο “1” γειτονικά, πρέπει να είναι διπλανά μεταξύ τους, οριζόντια ή κάθετα, όχι διαγώνια. (Σημείωση: συχνά διαγώνιο “1” ή ομάδες από διαγώνια “1”, υποκρύπτουν πύλες XOR ή XNOR). Το σχήμα που προκύπτει από την ομαδοποίηση πρέπει να είναι είτε τετράγωνο, είτε ορθογώνιο. Ούτε “**Τ**”, ούτε “**Γ**”. Μαθηματικά αυτό σημαίνει ότι δυο “1” είναι γειτονικά όταν, και μόνον όταν, μόνο μια μεταβλητή των δυο υπό εξέταση ελαχιστόρων έχει διαφορετική τιμή. Η’ με άλλα λόγια, όταν μόνο μια μεταβλητή των δυο υπό εξέταση ελαχιστόρων είναι σε κανονική μορφή στον έναν ελαχιστόρο και σε συμπληρωματική μορφή στον άλλο ελαχιστόρο.

6. Στόχος μας είναι να επιλέξουμε λίγες και μεγάλες ομάδες (με το μεγαλύτερο δυνατό πλήθος από “1”).. Οσο πιο μεγάλη είναι μια ομάδα, τόσο πιο πολλές μεταβλητές απλοποιούνται. Οι ομάδες μπορούν να περιλαμβάνουν πλήθος “1” που μπορεί να εκφραστεί ως μια δύναμη του 2 μόνο, δηλαδή $2^0=1$ (ένα “1” μόνο του), $2^1=2$ (ζευγάρι), $2^2=4$ (τετράδα), $2^3=8$ (οκτάδα), $2^4=16$ (δεκαεξάδα). Δεν επιτρέπονται ομάδες που αποτελούνται από 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15 από “1”.

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

7. Αντίστοιχα, σε κάθε περίπτωση απλοποιείται πλήθος μεταβλητών ίσο με το εκθέτη του 2 κάθε φορά. Δηλαδή, όταν έχουμε ένα "1" μόνο του (2^0), δεν έχουμε καμιά απλοποίηση, όταν έχουμε ζευγάρι (2^1) απλοποιείται μια μεταβλητή, όταν έχουμε τετράδα (2^2) απλοποιούνται δυο μεταβλητές κλπ.

8. Σε κάθε ομάδα απλοποιούνται εκείνες οι μεταβλητές οι οποίες αλλάζουν τιμή μέσα στην συγκεκριμένη ομάδα (δηλ. που στη μια κυψέλη είναι σε κανονική μορφή και σε άλλη είναι σε συμπληρωματική μορφή).

9. Η τελική απλοποιημένη συνάρτηση είναι στη μορφή λογικού αθροίσματος λογικών γινομένων ή/και ανεξάρτητων μεταβλητών και κάθε ομάδα αντιστοιχεί σε έναν όρο στην τελική απλοποιημένη συνάρτηση.

10. Κάθε ομάδα μπορεί να περιλαμβάνει "1" που περιλαμβάνονται και σε άλλη ομάδα (κοινά "1"), όμως κάθε ομάδα πρέπει να περιλαμβάνει οπωσδήποτε τουλάχιστον ένα μη κοινό "1". Δεν πρέπει δηλαδή να φτιάξουμε ομάδα με "1" που όλα ανήκουν και σε άλλες ομάδες. Αυτό συμφέρει όταν η χρησιμοποίηση κοινών "1" σε μια ομάδα απλοποιεί επιπλέον μεταβλητές, διαφορετικά είναι είναι διαδικασία χωρίς όφελος.

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

<u>Παράδειγμα:</u>

Για την προηγούμενη συνάρτηση, F(A,B,C) = Σ(3, 5, 6, 7) = A'BC + AB'C + ABC' + ABC,

σχηματίζονται τρία ζευγάρια.

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

και η απλοποιημένη μορφή της συνάρτησης είναι: F(A, B, C) = BC + AB + AC

Στο παράδειγμα αυτό βλέπουμε ότι, αν δεν είχε χρησιμοποιηθεί το κοινό "1", θα είχαμε ένα λογικό γινόμενο δυο μεταβλητών και δυο λογικά γινόμενα τριών μεταβλητών (δηλ. στην ομαδοποίηση θα είχαμε ένα ζευγάρι από "1" και δυο "1" μόνα τους).

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

11. Στην περίπτωση που έχουμε **συνθήκες αδιαφορίας** (αδιάφορους όρους, που συμβολίζονται με X στον πίνακα αλήθειας), αυτές αντιμετωπίζονται με τέτοιο τρόπο (δηλαδή τις θεωρούμε σαν “1” ή “0”), ώστε να πετυχαίνουμε το καλύτερο δυνατό αποτέλεσμα ομαδοποίησης, δηλαδή τις λιγότερες σε πλήθος ομάδες με τα περισσότερα σε πλήθος “1” και “X” μαζί. (Δεν είναι υποχρεωτικό να πάρουμε όλα τα X σαν “1” ή όλα σαν “0”.)

Παράδειγμα 11.1:

A	B	C	F
0	0	0	X
0	0	1	X
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A \ BC	00	01	11	10
0	X	X	1	0
1	0	1	1	1

F(A,B,C) = C + AB

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

Παράδειγμα 11.2:

A	B	C	D	F
0	0	0	0	1
0	0	0	1	X
0	0	1	0	1
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	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	X
1	1	0	1	X
1	1	1	0	0
1	1	1	1	0

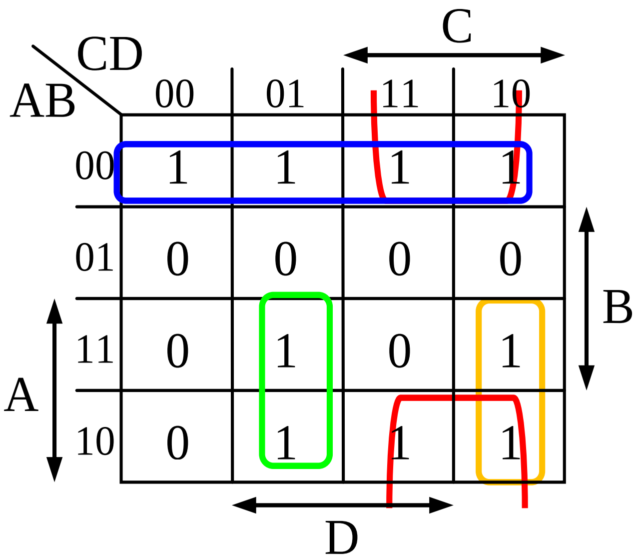
AB \ CD	00	01	11	10
00	1	X	1	1
01	0	1	1	1
11	X	X	0	0
10	1	0	1	1

F(A,B,C,D) = A'C + A'D + B'C + B'D'

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

ΠΑΡΑΔΕΙΓΜΑΤΑ

F(A,B,C,D) = Σ(m0, m1, m2, m3, m9, m10, m11, m12, m13) = **A’B’** + **B’C** + **AC’D** + **ACD’**



Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

F(A,B,C,D) = Σ(m0, m2, m4, m6, m10, m13) + X(7, 11, 15) = **A’D’** + **ABD** + **AB’C**

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	X	1
11	0	1	X	0
10	0	0	X	1

Απλοποίηση Λογικών Συναρτήσεων με πίνακες Karnaugh

F(A,B,C,D) = Σ(m0, m2, m4, m6, m12, m13, m14) + X(7, 11, 15) = **A'D'** + **AB**