Ψηφιακή Σχεδίαση

Άλγεβρα Boole & Λογικές Πύλες

ΕΛΕΥΘΕΡΙΟΣ ΚΟΣΜΑΣ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2019-2020 | ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Περίληψη

στην παρούσα διάλεξη...

- θα συζητήσουμε για τη δυαδική λογική και τις λογικές πύλες

- θα κάνουμε μία εισαγωγή στην άλγεβρα Boole, παρουσιάζοντας το παράδειγμα της δύτιμης άλγεβρας Boole
- θα συζητήσουμε τα αξιώματα και τις ιδιότητες της άλγεβρας Boole

- θα μελετήσουμε τις συναρτήσεις Boole και τον τρόπο απλοποίησής τους
- θα παρουσιάσουμε τις κανονικές και πρότυπες μορφές συναρτήσεων Boole

- θα συζητήσουμε το σύνολο των λογικών πράξεων και των λογικών πυλών

Εισαγωγή

Δυαδική λογική & λογικές πύλες

Δυαδική λογική

αφορά δυαδικές μεταβλητές και λογικές πράξεις

- δυαδικές μεταβλητές: παίρνουν δύο διακριτές τιμές
  - ναι/όχι, σωστό/λάθος
  - δυαδικό ψηφίο (bit): 0 ή 1 (→ Άλγεβρα Boole)
- λογικές πράξεις
  1. **AND** (**ΚΑΙ**): x · y = z (ή x y = z)
     - z = 1 αν και μόνο αν x = 1 και y = 1
     - αλλιώς z = 0
  2. **OR** (**Ή**): x + y = z
     - z = 1 αν είτε x = 1 ή y = 1
     - αλλιώς αν x = 0 και y = 0 → z = 0
  3. **NOT** (**ΟΧΙ**): x' = z
     - αν x = 0 → z = 1, αλλιώς εάν x = 0 → z = 1
     - ✍ αναφέρεται και ως **πράξη συμπληρώματος**

πίνακες αληθείας λογικών πράξεων

x	y	x · y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

Δυαδική λογική
σε σχέση με τη δυαδική αριθμητική

- η δυαδική λογική έχει ομοιότητες με τη δυαδική αριθμητική
- οι πράξεις AND και OR, έχουν ομοιότητες με τις πράξεις πρόσθεση και πολλαπλασιασμό
  - τα σύμβολα είναι δανεισμένα από τη δυαδική αριθμητική

όμως, η δυαδική λογική και δυαδική αριθμητική έχουν **διαφορές**

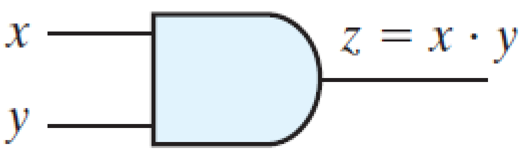
- μια αριθμητική μεταβλητή μπορεί να πάρει την τιμή ενός αριθμού
  - ✍ ο αριθμός μπορεί να αποτελείται από πολλά ψηφία

  π.χ. $1_2 + 1_2 = 10_2$ *(το οποίο σημαίνει "**ένα συν ένα ισον δύο**")*
- μία λογική μεταβλητή έχει πάντα τιμή 1 ή 0

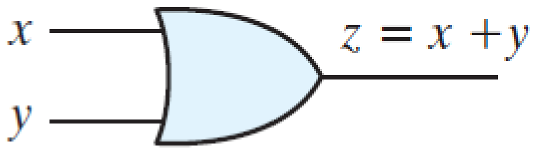
  π.χ. $1 + 1 = 1$ *(το οποίο διαβάζεται "**ένα OR ένα ισον με ένα**")*

Λογικές πύλες

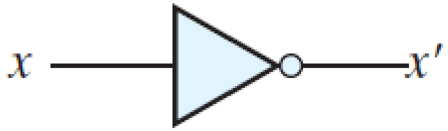
- μαθηματικές οντότητες που αντιστοιχούν σε ηλεκτρονικά κυκλώματα
  - δέχονται ένα ή περισσότερα σήματα εισόδου → παράγουν ένα σήμα εξόδου



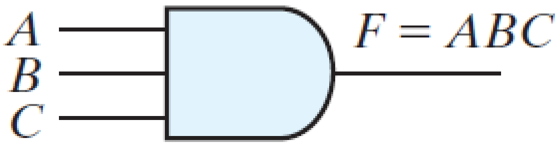
πύλη **AND**
δύο εισόδων



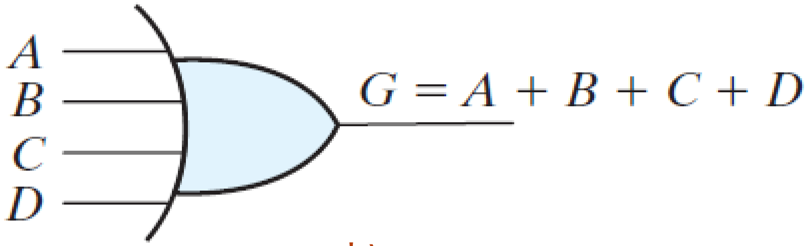
πύλη **OR**
δύο εισόδων



πύλη **NOT**
μίας εισόδου
ή αντιστροφέας



πύλη **AND**
τριών εισόδων

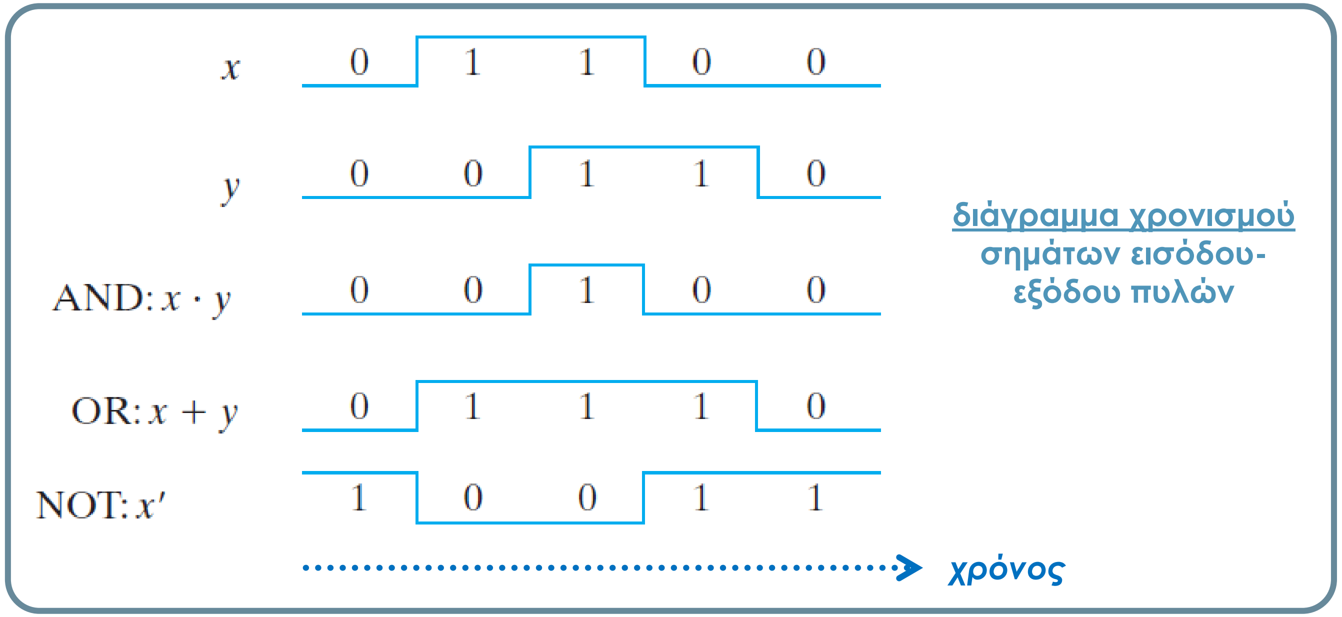


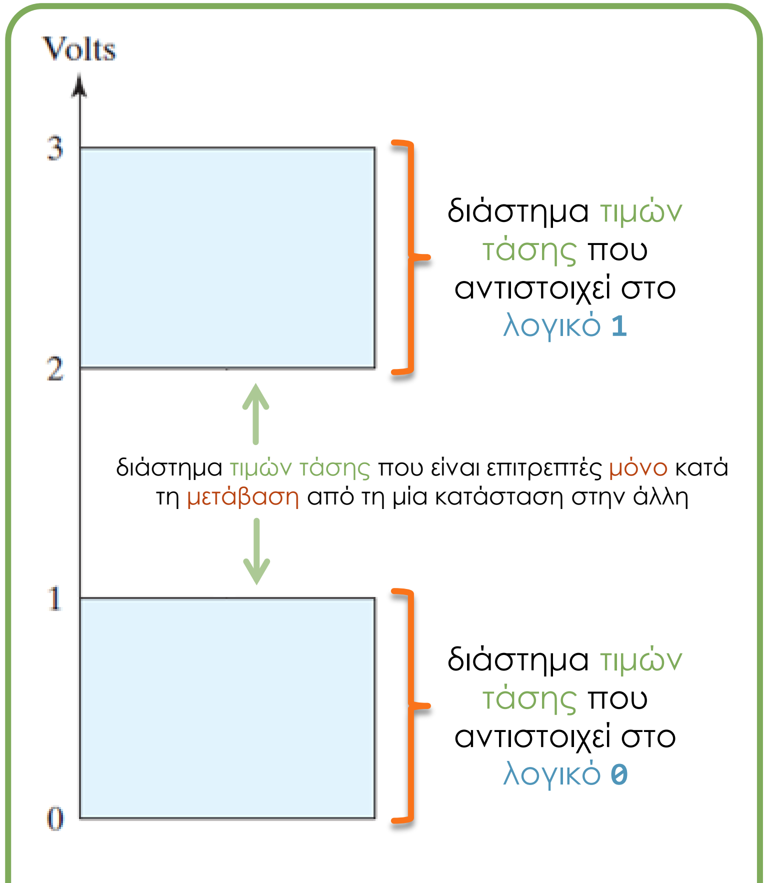
πύλη **OR**
τεσσάρων εισόδων

Λογικές πύλες

Ηλεκτρικά σήματα & Διακριτές λογικές τιμές

- στο ψηφιακό σύστημα, όλες οι τιμές των σημάτων είναι διακριτές (`0` ή `1`)





παράδειγμα δυαδικών σημάτων

- τα ηλεκτρικά σήματα (τάση, ρεύμα) είναι αναλογικά (ή συνεχή)
  - τιμές από 0 έως 3V

Δυαδική λογική & λογικές πύλες

Σύνοψη

- δυαδικές μεταβλητές: παίρνουν δυαδικές τιμές (**0** ή **1**)
- οι λογικοί τελεστές λειτουργούν σε δυαδικές τιμές και δυαδικές μεταβλητές
  - οι βασικοί λογικοί τελεστές είναι οι λογικές συναρτήσεις **AND**, **OR** και **NOT**
  - πίνακας αληθείας
    - η αναπαράσταση των τιμών μιας συνάρτησης σε πίνακα για όλους τους δυνατούς συνδυασμούς των μεταβλητών (ορισμάτων) της συνάρτησης
- οι λογικές πύλες υλοποιούν λογικές συναρτήσεις

Άλγεβρα Boole

Άλγεβρα Boole

- ένα χρήσιμο μαθηματικό σύστημα για περιγραφή και μετασχηματισμούς λογικών συναρτήσεων
- στόχος: εξεύρεση
  - λειτουργικά ισοδύναμων,
  - αλλά απλούστερων και
  - οικονομικότερων

  υλοποιήσεων ενός ψηφιακού κυκλώματος

✍ μελετούμε την Άλγεβρα Boole διότι είναι η βάση για σχεδιασμό και ανάλυση Ψηφιακών Συστημάτων!

Μαθηματικά συστήματα
Βασικοί ορισμοί

1. κλειστότητα (closure): ένα σύνολο **S** θεωρείται κλειστό ως προς ένα δυαδικό τελεστή ***** αν
   - για κάθε ζεύγος στοιχείων του **S**
   - ο τελεστής ***** καθορίζει έναν κανόνα για την εύρεση ενός μοναδικού στοιχείου, το οποίο ανήκει στο **S**

π.χ.

a) το σύνολο **N** των φυσικών αριθμών (**N={1,2,3,…}**) είναι κλειστό ως προς το δυαδικό τελεστή **+** (πρόσθεση)
   - επειδή για κάθε **a**, **b** στο **N** υπάρχει ένα μοναδικό c τέτοιο ώστε: **a + b = c**

b) ωστόσο, το σύνολο των φυσικών αριθμών δεν είναι κλειστό ως προς το δυαδικό τελεστή **-** (αφαίρεση)
   - επειδή **2 - 3 = -1**
   - **2**, **3** είναι φυσικοί αριθμοί ανήκουν στο **N**,
   - το **-1** δεν είναι φυσικός αριθμός

Μαθηματικά συστήματα
Βασικοί ορισμοί (II)

έστω, ένας δυαδικός τελεστής *, ορισμένος σε ένα σύνολο **S**

2. προσεταιριστικός κανόνας (associative law):
   ο τελεστής * είναι προσεταιριστικός όταν $(x * y) * z = x * (y * z)$,
   για κάθε στοιχείο **x**, **y**, **z** του **S**

3. αντιμεταθετικός κανόνας (commutative law):
   ο τελεστής * είναι αντιμεταθετικός όταν: $x * y = y * x$,
   για κάθε στοιχείο **x**, **y** του **S**

4. ουδέτερο στοιχείο (identity element):
   το σύνολο **S** έχει το ουδέτερο στοιχείο **e** ως προς τον τελεστή *, εάν υπάρχει **e** στο **S** με την εξής ιδιότητα: $e * x = x * e = x$,
   για κάθε στοιχείο **x** του **S**
   - π.χ. το **0** είναι ουδέτερο στοιχείο ως προς το δυαδικό τελεστή + στο σύνολο των ακεραίων **I** = **{...,-2,-1,0,1,2,...}**, επειδή $0 + x = x + 0 = x$,
     για κάθε στοιχείο **x** του **S**
   - π.χ. το **1** είναι ουδέτερο στοιχείο ως προς τον τελεστή · στο **I**, επειδή $1 \cdot x = x \cdot 1 = x$,
     για κάθε στοιχείο **x** του **S**

Μαθηματικά συστήματα

Βασικοί ορισμοί (III)

έστω, ένας δυαδικός τελεστής *, ορισμένος σε ένα σύνολο **S**

5. αντίστροφο (inverse):
   για το ένα σύνολο **S** που έχει ουδέτερο στοιχείο **e** ως προς τον τελεστή *, λέμε ότι εάν υπάρχει αντίστροφο, όταν για κάθε **x** στο **S** υπάρχει ένα στοιχείο **y** στο **S** ώστε: `x * y = e`
   - π.χ. στο σύνολο των ακεραίων **I** με τον τελεστή **+** και το ουδέτερο στοιχείο `0`, το αντίστροφο ενός στοιχείου **x** είναι το `(-x)`, επειδή: `x + (-x) = 0`
   - π.χ. στο σύνολο των πραγματικών αριθμών με τον τελεστή · και το ουδέτερο στοιχείο `1`, το αντίστροφο ενός στοιχείου **x** είναι το `(1/x)`, επειδή: `x · (1/x) = 1`

έστω, δύο δυαδικοί τελεστές * και · ορισμένοι σε ένα σύνολο **S**

6. επιμεριστικός κανόνας (distributive law):
   λέμε ότι ο τελεστής * είναι επιπεριστικός σε σχέση με τον τελεστή · όταν:

   `x * (y · z) = (x * y) · (x * z)`

Μαθηματικά συστήματα

Παράδειγμα - Πεδίο πραγματικών αριθμών

- πεδίο: σύνολο από στοιχεία, στο οποίο έχουν οριστεί δύο δυαδικοί τελεστές, οι οποίοι ικανοποιούν τις ιδιότητες 1 έως 6
  - π.χ. για το πεδίο πραγματικών αριθμών, με τους δυαδικούς τελεστές **+** (πρόσθεση) και **·** (πολλαπλασιασμός), ισχύουν οι ιδιότητες 1 έως 3, καθώς επίσης:

4. το ουδέτερο στοιχείο
   a) της πρόσθεσης είναι το **0**
   b) του πολλαπλασιασμού είναι το **1**
5. η αντίστροφη πράξη
   a) της πρόσθεσης είναι η αφαίρεση
   b) του πολλαπλασιασμού είναι η διαίρεση
6. ο μόνος επιμεριστικός κανόνας που ισχύει είναι αυτός του **·** ως προς το **+** :

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

Μαθηματικά συστήματα

Παράδειγμα - Άλγεβρα Boole

μία αλγεβρική δομή που ορίζεται σε ένα σύνολο στοιχείων **B**

- έχει δύο δυαδικούς τελεστές: **+** και **·**
- ισχύουν οι ιδιότητες 1 έως 3, καθώς επίσης:

4. το ουδέτερο στοιχείο
   a) ως προς **+** είναι το **0**
   b) ως προς **·** είναι το **1**
5. για κάθε στοιχείο **x** υπάρχει ένα στοιχείο **x'** (συμπλήρωμα του **x**), ώστε
   a) $x + x' = 1$
   b) $x \cdot x' = 0$
6. ισχύουν:
   a) η πράξη **·** είναι επιμεριστική ως προς την **+** : $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
   b) η πράξη **+** είναι επιμεριστική ως προς την **·** : $x + (y \cdot z) = (x + y) \cdot (x + z)$

Άλγεβρα Boole
Δύο τιμών ή άλγεβρα διακοπτών ή δυαδική λογική

- ορίζεται σε ένα σύνολο δύο στοιχείων, δηλαδή **`B = {0,1}`**
- κανόνες για τους δυαδικούς τελεστές δυαδικούς τελεστές: **+** (OR) και **·** (AND) που ορίζονται ως εξής:

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

x	x'
0	1
1	0

- οι κανόνες αυτοί είναι ακριβώς ίδιοι με εκείνους που χρησιμοποιήθηκαν για τον ορισμό των πράξεων AND, OR και NOT

Άλγεβρα Boole

Ιδιότητα δυϊσμού

- η άλγεβρα Boole ικανοποιεί την αρχή του δυϊσμού:

  κάθε αλγεβρική έκφραση που συνάγεται με βάση τα αξιώματα της άλγεβρας Boole παραμένει σε ισχύ αν οι τελεστές και τα ουδέτερα στοιχεία εναλλαχθούν

Άλγεβρα Boole
Αξιώματα και Θεωρήματα

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Θεώρημα 1	$x + x = x$	$x \cdot x = x$
Θεώρημα 2	$x + 1 = 1$	$x \cdot 0 = 0$
Θεώρημα 3, Διπλή άρνηση	$(x')' = x$	
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	$x + y = y + x$	$xy = yx$
Θεώρημα 4 (προσεταιριστική ιδιότητα)	$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
Αξίωμα 4 (επιμεριστική ιδιότητα)	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Θεώρημα 5, DeMorgan	$(x + y)' = x'y'$	$(xy)' = x' + y'$
Θεώρημα 6, Απορρόφηση	$x + xy = x$	$x(x+y) = x$

- τα **Αξιώματα** αποτελούν μέρος του ορισμού της Άλγεβρας Boole (→ δε χρειάζονται απόδειξη)
- τα **Θεωρήματα** αποδεικνύονται

Άλγεβρα Boole
Θεώρημα 1 - Απόδειξη

- τα **Θεωρήματα** που εμπλέκουν μία μόνο μεταβλητή αποδεικνύονται βάσει των **Αξιωμάτων**

	(α)	(β)
Αξίωμα 2	x + 0 = x	x · 1 = x
Αξίωμα 5	x + x' = 1	x · x' = 0
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	x + y = y + x	xy = yx
Αξίωμα 4 (επιμεριστική ιδιότητα)	x(y + z) = xy + xz	x + yz = (x + y)(x + z)

Θεώρημα 1 (α): x + x = x

πρόταση	**αιτιολογία**
x + x = (x + x) · 1	Αξίωμα 2(β)
= (x + x)(x + x')	Αξίωμα 5(α)
= x + xx'	Αξίωμα 4(β)
= x + 0	Αξίωμα 5(β)
= x	Αξίωμα 2(α)

Θεώρημα 1 (β): x · x = x

πρόταση	**αιτιολογία**
x · x = xx + 0	Αξίωμα 2(α)
= xx + xx'	Αξίωμα 5(β)
= x(x + x')	Αξίωμα 4(α)
= x · 1	Αξίωμα 5(α)
= x	Αξίωμα 2(β)

Άλγεβρα Boole

Θεώρημα 1 - Απόδειξη

- τα **Θεωρήματα** που εμπλέκουν μία μόνο μεταβλητή αποδεικνύονται βάσει των **Αξιωμάτων**

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	$x + y = y + x$	$xy = yx$
Αξίωμα 4 (επιμεριστική ιδιότητα)	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$

Θεώρημα 1 (α): $x + x = x$

πρόταση	αιτιολογία
$x + x = (x + x) \cdot 1$	Αξίωμα 2(β)
$= (x + x)(x + x')$	Αξίωμα 5(α)
$= x + xx'$	Αξίωμα 4(β)
$= x + 0$	Αξίωμα 5(β)
$= x$	Αξίωμα 2(α)

Θεώρημα 1 (β): $x \cdot x = x$

- το Θέωρημα 1(β) είναι η **δυϊκή μορφή** του Θεωρήματος 1(α)
- κάθε **βήμα** της απόδειξης στο τμήμα (β) είναι το **δυϊκό** του αντίστοιχου βήματος του τμήματος (α)

→ **οποιοδήποτε Θεώρημα μπορεί να αποδειχθεί από το δυϊκό του με ανάλογο τρόπο**

Άλγεβρα Boole

Θεώρημα 2 - Απόδειξη

- τα **Θεωρήματα** που εμπλέκουν μία μόνο μεταβλητή αποδεικνύονται βάσει των **Αξιωμάτων**

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	$x + y = y + x$	$xy = yx$
Αξίωμα 4 (επιμεριστική ιδιότητα)	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$

Θεώρημα 2 (α): $x + 1 = 1$

πρόταση	αιτιολογία
$x + 1 = 1 \cdot (x + 1)$	Αξίωμα 2(β)
$= (x + x')(x + 1)$	Αξίωμα 5(α)
$= x + x' \cdot 1$	Αξίωμα 4(β)
$= x + x'$	Αξίωμα 2(β)
$= 1$	Αξίωμα 5(α)

Θεώρημα 2 (β): $x \cdot 0 = 0$

ισχύει λόγω δυϊσμού

Άλγεβρα Boole
Θεώρημα 3 - Απόδειξη

- τα **Θεωρήματα** που εμπλέκουν μία μόνο μεταβλητή αποδεικνύονται βάσει των **Αξιωμάτων**

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	$x + y = y + x$	$xy = yx$
Αξίωμα 4 (επιμεριστική ιδιότητα)	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$

Θεώρημα 3: $(x')' = x$

πρόταση	αιτιολογία
$(x + x') = 1$ και $(x \cdot x') = 0$ → οι σχέσεις ορισμού του συμπληρώματος του x	Αξίωμα 5

- $(x + x') = 1$ και $(x \cdot x') = 0$ → οι σχέσεις ορισμού του συμπληρώματος του **x**
- το συμπλήρωμα του x' είναι
  - **x** και
  - $(x')'$
- επειδή το συμπλήρωμα είναι μοναδικό → $(x')' = x$

Άλγεβρα Boole
Θεώρημα 6 - Απόδειξη

- τα **Θεωρήματα** που εμπλέκουν δύο ή τρεις μεταβλητές αποδεικνύονται βάσει των **Αξιωμάτων** και των Θεωρημάτων που έχουν ήδη αποδειχθεί

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Αξίωμα 3 (αντιμεταθετική ιδιότητα)	$x + y = y + x$	$xy = yx$
Αξίωμα 4 (επιμεριστική ιδιότητα)	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Θεώρημα 2	$x + 1 = 1$	$x \cdot 0 = 0$

Θεώρημα 6 (α): $x + xy = x$

πρόταση	**αιτιολογία**
$x + xy = x \cdot 1 + xy$	Αξίωμα 2(β)
$= x(1 + y)$	Αξίωμα 4(α)
$= x(y + 1)$	Αξίωμα 3(α)
$= x \cdot 1$	Θεώρημα 2(α)
$= x$	Αξίωμα 2(β)

Θεώρημα 6 (β): $x(x + y) = x$

ισχύει λόγω δυϊσμού

Άλγεβρα Boole
Απόδειξη Θεωρημάτων με πίνακα αληθείας

- τα Θεωρήματα της άλγεβρας Boole μπορούν να αποδειχθούν
  - με αλγεβρικό τρόπο (όπως στις προηγούμενες διαφάνειες)
  - με χρήση πινάκων αληθείας

  π.χ.
  - απόδειξη **Θεωρήματος 6(a)** (απορρόφηση): `x + xy = x`

οι πίνακες αληθείας των x και x+xy είναι <u>ίδιοι</u>→ επομένως το Θεώρημα 6 <u>ισχύει</u>

x	y	xy	x + xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

  - απόδειξη **Θεωρήματος 5(a)** (**DeMorgan**): `(x + y)' = x'y'`

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

x'	y'	x'y'
1	1	1
1	0	0
0	1	0
0	0	0

οι πίνακες αληθείας των (x + y)' και x'y' είναι <u>ίδιοι</u> → επομένως το Θεώρημα 5α <u>ισχύει</u>

Άλγεβρα Boole

Προτεραιότητα τελεστών

παρενθέσεις > NOT > AND > OR

επομένως,

- πριν από όλες τις άλλες πράξεις πρέπει να υπολογίζονται οι **εκφράσεις** που είναι μέσα σε παρενθέσεις
- έπειτα, τα **συμπληρώματα**
- έπειτα η πράξη **AND**
- και τέλος, η πράξη **OR**

π.χ.

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

Συναρτήσεις Boole

Συνάρτηση Boole

- εκφράζει τη λογική σχέση ανάμεσα
  - σε μία εξαρτημένη δυαδική μεταβλητή και
  - έναν αριθμό από ανεξάρτητες δυαδικές μεταβλητές

  π.χ. $F = x + y'z$

- περιγράφεται από μία αλγεβρική έκφραση Boole που μπορεί να έχει:
  - ανεξάρτητες δυαδικές μεταβλητές
  - σταθερές (**0** ή **1**)
  - σύμβολα λογικών πράξεων

- οι τιμές που μπορεί να πάρει η συνάρτηση (δηλαδή η εξαρτημένη μεταβλητή) βρίσκονται εάν υπολογίσουμε την τιμή της έκφρασης Boole
  - για οποιοδήποτε συνδυασμό τιμών των ανεξάρτητων μεταβλητών, η συνάρτηση θα πάρει τιμή **0** ή **1**

Συνάρτηση Boole
Πίνακας αληθείας

- μία συνάρτηση Boole μπορεί να παρασταθεί με έναν πίνακα αληθείας
  - έστω **n** το πλήθος των ανεξάρτητων μεταβλητών
  - τότε, το πλήθος των γραμμών είναι: 2^n

π.χ. η συνάρτηση

`F = x + y'z`

έχει το διπλανό πίνακα αληθείας

x	y	z	F = x + y'z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

✍ παρατηρήστε ότι η **F** παίρνει τιμή **1**

- όταν `x = 1` και
- `yz = 01`

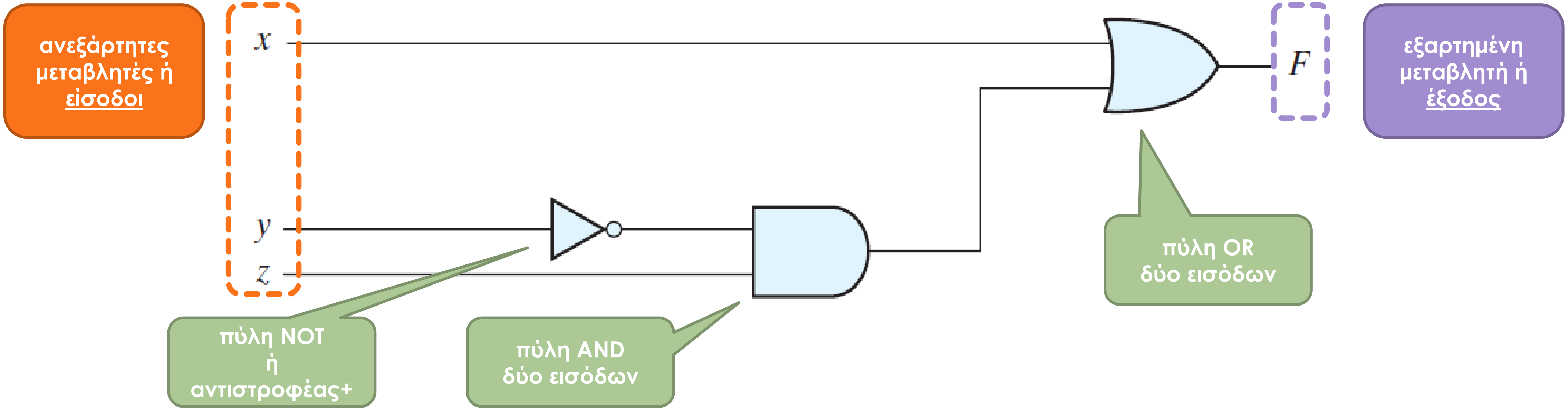
ενώ σε οποιαδήποτε άλλη περίπτωση παίρνει τιμή **0**

Συνάρτηση Boole

Λογικό κύκλωμα

- μία συνάρτηση Boole σε αλγεβρική μορφή μπορεί να μετασχηματιστεί σε ένα λογικό κύκλωμα
  - διασυνδεδεμένες λογικές πύλες που σχηματίζουν μια συγκεκριμένη δομή

π.χ. ακολουθεί η υλοποίηση με λογικές πύλες της συνάρτησης $F = x + y'z$



Συνάρτηση Boole
Τρόποι αναπαράστασης

- υπάρχει ένας και μοναδικός τρόπος αναπαράστασης με πίνακα αληθείας

- ωστόσο, εάν η συνάρτηση Boole είναι σε αλγεβρική μορφή
  - μπορεί να εκφραστεί με πολλούς τρόπους
  - καθένας από τους οποίους έχει ένα ισοδύναμο λογικό κύκλωμα

- αλγεβρική έκφραση που χρησιμοποιείται → υποδεικνύει τον τρόπο διασύνδεσης των πυλών στο αντίστοιχο λογικό κύκλωμα
  - απλούστερη αλγεβρική έκφραση → λιγότερες πύλες, με λιγότερες εισόδους ανά πύλη → οικονομικότερο ψηφιακό κύκλωμα
  - ✍ είναι ο βασικός λόγος που μελετάμε την άλγεβρα Boole

Συνάρτηση Boole

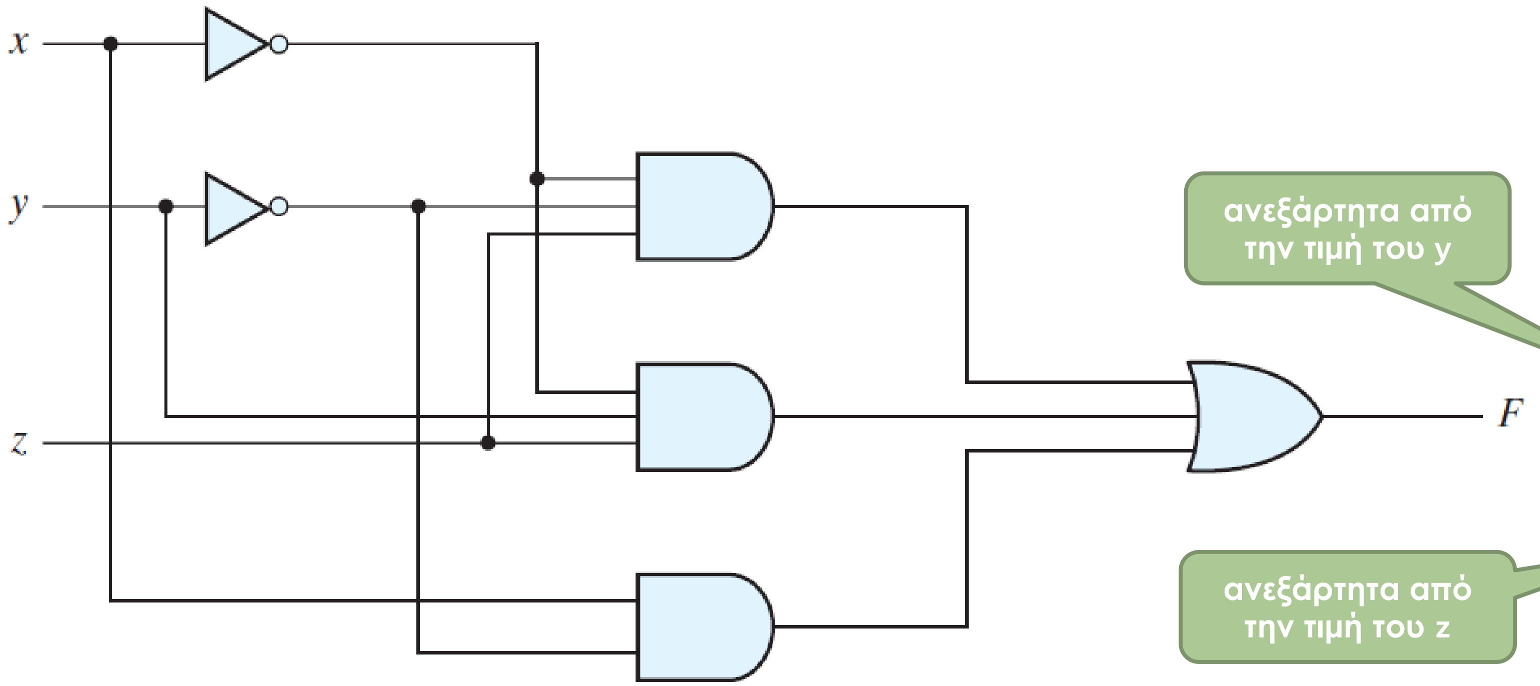
Διαφορετικά λογικά κυκλώματα - Παράδειγμα

- η συνάρτηση: **F = x'y'z + x'yz + xy'**

  έχει το διπλανό πίνακα αληθείας

  και αντιστοιχεί στο ακόλουθο λογικό κύκλωμα

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



ανεξάρτητα από την τιμή του y

ανεξάρτητα από την τιμή του z

✍ παρατηρήστε ότι η **F** παίρνει τιμή **1**

- όταν **xyz = 001** ή **011** → άρα όταν **xz = 01**, και
- όταν **xy = 10**

ενώ σε οποιαδήποτε άλλη περίπτωση παίρνει τιμή **0**

Συνάρτηση Boole

Διαφορετικά λογικά κυκλώματα - Παράδειγμα (II)

- μπορούμε να απλοποιήσουμε τη συνάρτηση:

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

χρησιμοποιώντας την άλγεβρα Boole

π.χ.

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Θεώρημα 1	$x + x = x$	$x \cdot x = x$
Θεώρημα 2	$x + 1 = 1$	$x \cdot 0 = 0$
Θεώρημα 3	$(x')' = x$	
Αξίωμα 3	$x + y = y + x$	$xy = yx$
Θεώρημα 4	$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
Αξίωμα 4	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Θεώρημα 5	$(x + y)' = x'y'$	$(xy)' = x' + y'$
Θεώρημα 6	$x + xy = x$	$x(x+y) = x$

πρόταση	**αιτιολογία**
$x'y'z + x'yz + xy' = x'z(y' + y) + xy'$	Αξίωμα 4(α)
$= x'z \cdot 1 + xy'$	Αξίωμα 5(α)
$= x'z + xy'$	Αξίωμα 2(β)

απλοποιημένη συνάρτηση

$$\boxed{F = x'z + xy'}$$

Συνάρτηση Boole

Διαφορετικά λογικά κυκλώματα - Παράδειγμα (III)

μπορούμε να επιβεβαιώσουμε ότι η αρχική F και η απλοποιημένη είναι ισοδύναμες, εξετάζοντας εάν οι πίνακες αληθείας τους είναι ίδιοι

- μία απλοποίηση της συνάρτησης:

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

είναι: $$F = x'z + xy'$$

η οποία έχει το διπλανό πίνακα αληθείας

και αντιστοιχεί στο ακόλουθο λογικό κύκλωμα

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

το λογικό κύκλωμα της απλοποιημένης F είναι απλούστερο από το λογικό κύκλωμα της αρχικής F

✍ παρατηρήστε ότι η F παίρνει τιμή **1**

- όταν xz = **01** και
- όταν xy = **10**

ενώ σε οποιαδήποτε άλλη περίπτωση παίρνει τιμή **0**

Συνάρτηση Boole
Σύνοψη

- κάθε συνάρτηση αποτελείται από όρους και μεταβλητές
- κατά τη μετατροπή της συνάρτησης σε λογικό κύκλωμα
  - οι όροι μετατρέπονται σε πύλες και
  - κάθε μεταβλητή του όρου, αποτελεί είσοδο της αντίστοιχης πύλης

- ονομάζουμε παράγοντα μία μεταβλητή (ως συμπλήρωμα ή μη) που εμφανίζεται σε κάποιο όρο

  π.χ. η συνάρτηση $F = x'y'z + x'yz + xy'$
  - έχει τρεις όρους με οκτώ παράγοντες

  ενώ η συνάρτηση $F = x'z + xy'$
  - έχει δύο όρους με τέσσερις παράγοντες

- κάθε μεταβλητή μπορεί να είναι είτε στην κανονική της μορφή είτε ως συμπλήρωμα,
- εάν εμφανίζεται και στις δύο μορφές → μετρά ως δύο παράγοντες

- αν σε μία συνάρτηση, μειώσουμε είτε τον αριθμό των όρων είτε τον αριθμό των παραγόντων → επιτυγχάνουμε απλούστερο κύκλωμα

Συναρτήσεις Boole

Απλοποίηση

Συνάρτηση Boole

Απλοποίηση - 1ο & 2ο παράδειγμα

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	x + 0 = x	x · 1 = x
Αξίωμα 5	x + x' = 1	x · x' = 0
Θεώρημα 1	x + x = x	x · x = x
Θεώρημα 2	x + 1 = 1	x · 0 = 0
Θεώρημα 3	(x')' = x	
Αξίωμα 3	x + y = y + x	xy = yx
Θεώρημα 4	x + (y + z) = (x + y) + z	x(yz) = (xy)z
Αξίωμα 4	x(y + z) = xy + xz	x + yz = (x + y)(x + z)
Θεώρημα 5	(x + y)' = x'y'	(xy)' = x' + y'
Θεώρημα 6	x + xy = x	x(x+y) = x

1ο παράδειγμα: x(x' + y)

πρόταση	αιτιολογία
x(x' + y) = xx' + xy	Αξίωμα 4(α)
= 0 + xy	Αξίωμα 5(β)
= xy	Αξίωμα 2(α)

2ο παράδειγμα: x + x'y

πρόταση	αιτιολογία
x + x'y = (x + x')(x + y)	Αξίωμα 4(α)
= 1 · (x + y)	Αξίωμα 5(α)
= x + y	Αξίωμα 2(β)

Συνάρτηση Boole

Απλοποίηση - 1ο & 2ο παράδειγμα (II)

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	x + 0 = x	x · 1 = x
Αξίωμα 5	x + x' = 1	x · x' = 0
Θεώρημα 1	x + x = x	x · x = x
Θεώρημα 2	x + 1 = 1	x · 0 = 0
Θεώρημα 3	(x')' = x	
Αξίωμα 3	x + y = y + x	xy = yx
Θεώρημα 4	x + (y + z) = (x + y) + z	x(yz) = (xy)z
Αξίωμα 4	x(y + z) = xy + xz	x + yz = (x + y)(x + z)
Θεώρημα 5	(x + y)' = x'y'	(xy)' = x' + y'
Θεώρημα 6	x + xy = x	x(x+y) = x

1ο παράδειγμα: $\boxed{x(x' + y)}$

πρόταση	αιτιολογία
$x(x' + y) = xx' + xy$	Αξίωμα 4(α)
$= 0 + xy$	Αξίωμα 5(β)
$= xy$	Αξίωμα 2(α)

2ο παράδειγμα: $\boxed{x + x'y}$

πρόταση αιτιολογία

x +

η 2η συνάρτηση είναι η δυϊκή της 1ης
→ άρα θα μπορούσαμε να την απλοποιήσουμε εφαρμόζοντας την ιδιότητα δυϊσμού

Συνάρτηση Boole

Απλοποίηση - 1ο & 2ο παράδειγμα (ΙΙΙ)

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	x + 0 = x	x · 1 = x
Αξίωμα 5	x + x' = 1	x · x' = 0
Θεώρημα 1	x + x = x	x · x = x
Θεώρημα 2	x + 1 = 1	x · 0 = 0
Θεώρημα 3	(x')' = x	
Αξίωμα 3	x + y = y + x	xy = yx
Θεώρημα 4	x + (y + z) = (x + y) + z	x(yz) = (xy)z
Αξίωμα 4	x(y + z) = xy + xz	x + yz = (x + y)(x + z)
Θεώρημα 5	(x + y)' = x'y'	(xy)' = x' + y'
Θεώρημα 6	x + xy = x	x(x+y) = x

1ο παράδειγμα: $\boxed{x(x' + y)} = \boxed{xy}$

2ο παράδειγμα: $\boxed{x + x'y}$

η 2η συνάρτηση είναι η <u>δυϊκή</u> της 1ης → άρα μπορούμε να την απλοποιήσουμε <u>εφαρμόζοντας</u> την <u>ιδιότητα δυϊσμού</u>

<u>πρόταση</u>

$x + x'y = \boxed{x + y}$

<u>αιτιολογία</u>

εφαρμόζοντας την ιδιότητα δυισμού στην απλοποιημένη μορφή της 1ης συνάρτησης

Συνάρτηση Boole

Απλοποίηση - 3ο Παράδειγμα

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	x + 0 = x	x · 1 = x
Αξίωμα 5	x + x' = 1	x · x' = 0
Θεώρημα 1	x + x = x	x · x = x
Θεώρημα 2	x + 1 = 1	x · 0 = 0
Θεώρημα 3	(x')' = x	
Αξίωμα 3	x + y = y + x	xy = yx
Θεώρημα 4	x + (y + z) = (x + y) + z	x(yz) = (xy)z
Αξίωμα 4	x(y + z) = xy + xz	x + yz = (x + y)(x + z)
Θεώρημα 5	(x + y)' = x'y'	(xy)' = x' + y'
Θεώρημα 6	x + xy = x	x(x+y) = x

3ο παράδειγμα: (x + y)(x + y')

πρόταση	**αιτιολογία**
(x + y)(x + y') = (x + y)x + (x + y)y'	Αξίωμα 4(α)
= xx + xy + xy' + yy'	Αξίωμα 4(α)
= x + xy + xy' + yy'	Θεώρημα 1(β)
= x + xy + xy' + 0	Αξίωμα 5(β)
= x + xy + xy'	Αξίωμα 2(α)
= x(1 + y + y') = x(1 + 1) = x · 1	Αξίωμα 4(α), Αξίωμα 5(α)
= x	Αξίωμα 2(β)

Συνάρτηση Boole
Απλοποίηση - 4ο παράδειγμα

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Θεώρημα 1	$x + x = x$	$x \cdot x = x$
Θεώρημα 2	$x + 1 = 1$	$x \cdot 0 = 0$
Θεώρημα 3	$(x')' = x$	
Αξίωμα 3	$x + y = y + x$	$xy = yx$
Θεώρημα 4	$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
Αξίωμα 4	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Θεώρημα 5	$(x + y)' = x'y'$	$(xy)' = x' + y'$
Θεώρημα 6	$x + xy = x$	$x(x+y) = x$

4ο παράδειγμα: $xy + x'z + yz$

πρόταση	αιτιολογία
$xy + x'z + yz = xy + x'z + yz \cdot 1$	Αξίωμα 2(β)
$= xy + x'z + yz(x + x')$	Αξίωμα 5(α)
$= xy + x'z + xyz + x'yz$	Αξίωμα 4(α)
$= xy(1 + z) + x'z(1 + y)$	Αξίωμα 4(α)
$= xy \cdot 1 + x'z \cdot 1 = xy + x'z$	Θεώρημα 2(α), Αξίωμα 2(β)

Συνάρτηση Boole
Απλοποίηση - 5ο παράδειγμα

- στόχος: η μείωση
  - είτε του αριθμού των όρων
  - είτε του αριθμού των παραγόντων μίας συνάρτησης
- το επιτυγχάνουμε μέσω της αλγεβρικής επεξεργασίας της άλγεβρας Boole

	(α)	(β)
Αξίωμα 2	$x + 0 = x$	$x \cdot 1 = x$
Αξίωμα 5	$x + x' = 1$	$x \cdot x' = 0$
Θεώρημα 1	$x + x = x$	$x \cdot x = x$
Θεώρημα 2	$x + 1 = 1$	$x \cdot 0 = 0$
Θεώρημα 3	$(x')' = x$	
Αξίωμα 3	$x + y = y + x$	$xy = yx$
Θεώρημα 4	$x + (y + z) = (x + y) + z$	$x(yz) = (xy)z$
Αξίωμα 4	$x(y + z) = xy + xz$	$x + yz = (x + y)(x + z)$
Θεώρημα 5	$(x + y)' = x'y'$	$(xy)' = x' + y'$
Θεώρημα 6	$x + xy = x$	$x(x+y) = x$

4ο παράδειγμα: $\boxed{xy + x'z + yz} = \boxed{xy + x'z}$

5ο παράδειγμα: $\boxed{(x + y)(x' + z)(y + z)}$

η 5η συνάρτηση είναι η δυϊκή της 4ης
→ άρα μπορούμε να την απλοποιήσουμε εφαρμόζοντας την ιδιότητα δυϊσμού

πρόταση	αιτιολογία
$(x + y)(x' + z)(y + z) = \boxed{(x + y)(x' + z)}$	εφαρμόζοντας την ιδιότητα δυισμού στην απλοποιημένη μορφή της 4ης συνάρτησης

Συναρτήσεις Boole

Συμπλήρωμα

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - Πίνακας αληθείας

το συμπλήρωμα **F'** μιας συνάρτησης **F**

- προκύπτει από τον πίνακα αληθείας
  εάν εναλλάξουμε τα bits (**0**→**1** και **1**→**0**)

x	y	z	F	F'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - 1ος αλγεβρικός τρόπος

το συμπλήρωμα **F'** μιας συνάρτησης **F**

- παράγεται με χρήση του Θεωρήματος DeMorgan (Θεώρημα 5)
  - για δύο μεταβλητές:

(α)	(β)
(x + y)' = x'y'	(xy)' = x' + y'

  - για τρεις μεταβλητές:
    - <u>απόδειξη</u>:

(α)	(β)
(x + y + z)' = x'y'z'	(xyz)' = x' + y' + z'

πρόταση	**αιτιολογία**
(x + y + z)' = (x + A)'	έστω ότι y + z = A
= x'A' = x'(y + z)'	Θεώρημα 5(α), αντικατάσταση του A με το y + z
= x'(y'z') = x'y'z'	Θεώρημα 5(α), Θεώρημα 4(β)

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - 1ος αλγεβρικός τρόπος (II)

το συμπλήρωμα **F'** μιας συνάρτησης **F**

- παράγεται με χρήση του Θεωρήματος DeMorgan (Θεώρημα 5)
  - για δύο μεταβλητές:

(α)	(β)
(x + y)' = x'y'	(xy)' = x' + y'

  - για τρεις μεταβλητές:
    - <u>απόδειξη</u>:

(α)	(β)
(x + y + z)' = x'y'z'	(xyz)' = x' + y' + z'

  - γενική μορφή:

(α)	(β)
(x + y + z + … + w)' = x'y'z'…w'	(xyz…w)' = x' + y' + z' + … + w'

✍ το **συμπλήρωμα** μίας συνάρτησης παράγεται:

1. αν **αλλάξουμε** τους **τελεστές AND** και **OR**, και
2. **συμπληρώσουμε** κάθε παράγοντα
   - δηλαδή εκτελέσουμε την πράξη **NOT** σε κάθε παράγοντα

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - 1ος αλγεβρικός τρόπος - 1ο παράδειγμα

βρείτε το συμπλήρωμα F' της συνάρτησης: $F = x'yz' + x'y'z$

- $F' = (x'yz' + x'y'z)'$

πρόταση	αιτιολογία
$(x'yz' + x'y'z)' = (x'yz')'(x'y'z)'$	Θεώρημα 5(α) (για τρεις μεταβλητές)
$= ((x')' + y' + (z')')((x')' + (y')' + z')$	Θεώρημα 5(β) (για τρεις μεταβλητές)
$= (x + y' + z)(x + y + z')$	Θεώρημα 3

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - 1ος αλγεβρικός τρόπος - 2ο παράδειγμα

βρείτε το συμπλήρωμα F' της συνάρτησης: $F = x(y'z' + yz)$

- $F' = (x(y'z' + yz))'$

πρόταση	**αιτιολογία**
$(x(y'z' + yz))' = x' + (y'z' + yz)'$	Θεώρημα 5(β) (για δύο μεταβλητές)
$= x' + (y'z')'(yz)'$	Θεώρημα 5(α) (για δύο μεταβλητές)
$= x' + ((y')' + (z')')(y' + z')$	Θεώρημα 5(β) (για δύο μεταβλητές)
$= x' + (y + z)(y' + z')$	Θεώρημα 3
$= x' + (y + z)y' + (y + z)z'$	Αξίωμα 4(α)
$= x' + yy' + y'z + yz' + zz'$	Αξίωμα 4(α)
$= x' + 0 + y'z + yz' + 0 = x' + y'z + yz'$	Αξίωμα 5(β), Αξίωμα 2(α)

Συνάρτηση Boole

Συμπλήρωμα συνάρτησης - 2ος αλγεβρικός τρόπος

το συμπλήρωμα **F'** μιας συνάρτησης **F** μπορεί να παραχθεί ως εξής:

1. να πάρουμε τη **δυϊκή** μορφή της **F** και
2. να **συμπληρώσουμε** κάθε παράγοντα

- π.χ. βρείτε το συμπλήρωμα **F'** της συνάρτησης: $F = x'yz' + x'y'z$
  1. η δυική μορφή της **F** είναι: $(x' + y + z')(x' + y' + z)$
  2. η συμπλήρωση κάθε παράγοντα δίνει: $(x + y' + z)(x + y + z') = F'$

- π.χ. βρείτε το συμπλήρωμα **F'** της συνάρτησης: $F = x(y'z' + yz)$
  1. η δυική μορφή της **F** είναι: $x + (y' + z')(y + z)$
  2. η συμπλήρωση κάθε παράγοντα δίνει: $x' + (y + z)(y' + z') = F'$

Συναρτήσεις Boole

Κανονική μορφή: άθροισμα ελαχιστόρων ή γινόμενο μεγιστόρων

Συνάρτηση Boole
Ελαχιστόροι

- μία δυαδική μεταβλητή **x** μπορεί να εμφανιστεί σε μία αλγεβρική έκφραση
  - είτε στην κανονική της μορφή: **x**
  - είτε συμπληρωμένη: **x'**
- για το γινόμενο (πράξη AND) δύο δυαδικών μεταβλητών **x** και **y** υπάρχουν **2^2** διαφορετικές δυνατές μορφές

  x'y', x'y , xy', xy

  ✍ κάθε ένας από τους όρους ονομάζεται **ελαχιστόρος**
- γενικά, **n** μεταβλητές μπορούν να σχηματίσουν **2^n** **ελαχιστόρους**

Συνάρτηση Boole
Ελαχιστόροι (II)

γενικά, **n** μεταβλητές μπορούν να σχηματίσουν **2^n** **ελαχιστόρους** οι οποίοι προκύπτουν ως εξής:

1. οι δυαδικοί αριθμοί των **n** bits τοποθετούνται κατά αύξουσα σειρά, κάτω από τις **n** μεταβλητές
2. κάθε **ελαχιστόρος** είναι ένα γινόμενο **n** μεταβλητών, με κάθε μεταβλητή:
   - σε συμπληρωμένη μορφή, αν το αντίστοιχο bit του δυαδικού αριθμού είναι **0**
   - σε κανονική μορφή, αν το αντίστοιχο bit του δυαδικού αριθμού είναι **1**

✍ κάθε **ελαχιστόρος** ονομάζεται m_j όπου **j** είναι ο δεκαδικός αριθμός που έχει την ίδια τιμή με τον δυαδικό αριθμό που αντιστοιχεί στον ελαχιστόρο

			ελαχιστόροι	
x	y	z	όρος	όνομα
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

παράδειγμα για $n = 3$

Συνάρτηση Boole

Μεγιστόροι

- μεγιστόρος ονομάζεται το άθροισμα (πράξη OR) n δυαδικών μεταβλητών, όπου κάθε μεταβλητή είναι
  - σε συμπληρωμένη μορφή, αν το αντίστοιχο bit του δυαδικού αριθμού είναι **1**
  - σε κανονική μορφή, αν το αντίστοιχο bit του δυαδικού αριθμού είναι **0**
- υπάρχουν 2^n μεγιστόροι των n μεταβλητών

✍ κάθε **μεγιστόρος** ονομάζεται M_j όπου j είναι ο δεκαδικός αριθμός που έχει την ίδια τιμή με τον δυαδικό αριθμό που αντιστοιχεί στο **μεγιστόρο**

			μεγιστόροι	
x	y	z	όρος	όνομα
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

παράδειγμα για n = 3

Συνάρτηση Boole

Ελαχιστόροι & μεγιστόροι

- ✍ παρατηρείστε ότι κάθε **ελαχιστόρος** είναι το συμπλήρωμα του αντίστοιχου **μεγιστόρου**
  - και αντίστροφα
  - δηλαδή ισχύει ότι:
    - $m'_j = M_j$ και
    - $M'_j = m_j$

			ελαχιστόροι		μεγιστόροι	
x	y	z	όρος	όνομα	όρος	όνομα
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

παράδειγμα για $n = 3$

Συνάρτηση Boole
Κανονική μορφή - Άθροισμα ελαχιστόρων

- χρησιμοποιώντας τον πίνακα αληθείας μίας συνάρτησης, μπορούμε να την εκφράσουμε ως άθροισμα **ελαχιστόρων**
  - επιλέγουμε όσους δίνουν τιμή **1** στη συνάρτηση

x	y	z	F	ελαχιστόροι	
				όρος	όνομα
0	0	0	0	x'y'z'	m_0
0	0	1	1	x'y'z	m_1
0	1	0	0	x'yz'	m_2
0	1	1	0	x'yz	m_3
1	0	0	1	xy'z'	m_4
1	0	1	0	xy'z	m_5
1	1	0	0	xyz'	m_6
1	1	1	1	xyz	m_7

$$F = x'y'z + xy'z' + xyz$$
$$= m_1 + m_4 + m_7$$

Συνάρτηση Boole

Κανονική μορφή - Γινόμενο μεγιστόρων

- χρησιμοποιώντας τον πίνακα αληθείας μίας συνάρτησης, μπορούμε να την εκφράσουμε ως γινόμενο **μεγιστόρων**
  - επιλέγουμε όσους δίνουν τιμή 0 στη συνάρτηση

x	y	z	F	μεγιστόροι: όρος	μεγιστόροι: όνομα
0	0	0	0	x + y + z	M_0
0	0	1	1	x + y + z'	M_1
0	1	0	0	x + y' + z	M_2
0	1	1	0	x + y' + z'	M_3
1	0	0	1	x' + y + z	M_4
1	0	1	0	x' + y + z'	M_5
1	1	0	0	x' + y' + z	M_6
1	1	1	1	x' + y' + z'	M_7

$$F = (x + y + z)\cdot(x + y' + z)\cdot(x + y' + z')\cdot(x' + y + z')\cdot(x' + y' + z) = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

Συνάρτηση Boole

Κανονική μορφή

- χρησιμοποιώντας τον πίνακα αληθείας μίας συνάρτησης, μπορούμε να την εκφράσουμε ως:
  1. άθροισμα ελαχιστόρων
     - επιλέγουμε όσους δίνουν τιμή **1** στη συνάρτηση
  2. γινόμενο μεγιστόρων
     - επιλέγουμε όσους δίνουν τιμή **0** στη συνάρτηση

- μία συναρτήση Boole είναι σε **κανονική μορφή**, όταν έχει εκφραστεί ως **άθροισμα ελαχιστόρων** ή ως **γινόμενο μεγιστόρων**

Συνάρτηση Boole
Κανονική μορφή - Παράδειγμα

				ελαχιστόροι		μεγιστόροι	
x	y	z	F	όρος	όνομα	όρος	όνομα
0	0	0	0	x'y'z'	m_0	x + y + z	M_0
0	0	1	0	x'y'z	m_1	x + y + z'	M_1
0	1	0	0	x'yz'	m_2	x + y' + z	M_2
0	1	1	1	x'yz	m_3	x + y' + z'	M_3
1	0	0	0	xy'z'	m_4	x' + y + z	M_4
1	0	1	1	xy'z	m_5	x' + y + z'	M_5
1	1	0	1	xyz'	m_6	x' + y' + z	M_6
1	1	1	1	xyz	m_7	x' + y' + z'	M_7

$$F = (x + y + z)\cdot(x + y + z')\cdot(x + y' + z)\cdot(x' + y + z) = M_0 \cdot M_1 \cdot M_2 \cdot M_4$$

$$F = x'yz + xy'z' + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

Συνάρτηση Boole

Μετατροπή σε άθροισμα ελαχιστόρων - 1η μέθοδος

ακολουθούμε την εξής διαδικασία:

1. αναπτύσουμε την αλγεβρική έκφραση σε μορφή αθροίσματος γινομένων (**όρων AND**)
2. εάν κάποιος **όρος** δεν περιέχει όλες τις μεταβλητές της συνάρτησης
   - για κάθε μεταβλητή **A** που λείπει → πολλαπλασιάζουμε τον **όρο** με το $(A + A')$

- π.χ. γράψτε τη συνάρτηση Boole ως **άθροισμα ελαχιστόρων** $F = x + y'z$

η F είναι ήδη σε μορφή αθροίσματος γινομένων (1ο βήμα) και αποτελείται από δύο όρους
- από τον πρώτο όρο λείπουν οι y και z
- από τον δεύτερο όρο λείπει η x

2. 1ος όρος: $x = x(y + y')$
$= xy + xy' = (xy + xy')(z + z')$
$= xyz + xyz' + xy'z + xy'z'$

2ος όρος: $y'z = y'z(x + x') = xy'z + x'y'z$

$F = xyz + xyz' + \boxed{xy'z} + xy'z' + \cancel{\boxed{xy'z}} + x'y'z$
$= xyz + xyz' + xy'z + xy'z' + x'y'z$

καθώς (από το Θεώρημα 1α) ισχύει ότι: x + x = x

$\boxed{F = m_1 + m_4 + m_5 + m_6 + m_7}$

Συνάρτηση Boole

Μετατροπή σε άθροισμα ελαχιστόρων - 2η μέθοδος

ακολουθούμε την εξής διαδικασία:

1. κατασκευάζουμε τον πίνακα αληθείας της συνάρτησης, χρησιμοποιώντας την αλγεβρική έκφραση
2. προσδιορίζουμε τους **ελαχιστόρους** από τον πίνακα αληθείας

❖ π.χ. γράψτε τη συνάρτηση Boole $F = x + y'z$ ως **άθροισμα ελαχιστόρων**

1. κατασκευάζουμε τον πίνακα αληθείας βάζοντας την τιμή **1** στην **F**, στις γραμμές όπου:
   - $x = 1$
   - $yz = 01$
2. προσδιορίζουμε τους **ελαχιστόρους**
   - $F = xyz + xyz' + xy'z + xy'z' + x'y'z$

$$\boxed{F = m_1 + m_4 + m_5 + m_6 + m_7}$$

				ελαχιστόροι	
x	y	z	F	όρος	όνομα
0	0	0	0	x'y'z'	m_0
0	0	1	1	x'y'z	m_1
0	1	0	0	x'yz'	m_2
0	1	1	0	x'yz	m_3
1	0	0	1	xy'z'	m_4
1	0	1	1	xy'z	m_5
1	1	0	1	xyz'	m_6
1	1	1	1	xyz	m_7

Συνάρτηση Boole

Μετατροπή σε γινόμενο μεγιστόρων - 1η μέθοδος

ακολουθούμε την εξής διαδικασία:

1. αναπτύσουμε την αλγεβρική έκφραση σε μορφή γινομένου αθροισμάτων (**όρων OR**)
   - χρησιμοποιούμε τον επιμεριστικό κανόνα: $A + BC = (A + B)(A + C)$
2. εάν κάποιος **όρος** δεν περιέχει όλες τις μεταβλητές της συνάρτησης
   - για κάθε μεταβλητή **A** που λείπει → προσθέτουμε στον **όρο** το AA'

- π.χ. γράψτε τη συνάρτηση Boole $F = xy + x'z$ ως **γινόμενο μεγιστόρων**

στο 1ο βήμα εφαρμόζουμε επαναληπτικά τον επιμεριστικό κανόνα

x + x' = 1 (Αξίωμα 5a)

1. $F = xy + x'z = (xy + x')(xy + z) = (x + x')(y + x')(x + z)(y + z)$
   $= (x' + y)(x + z)(y + z)$

2. 1ος όρος: $x' + y = x' + y + zz' = (x' + y + z)(x' + y + z')$
   2ος όρος: $x + z = x + z + yy' = (x + y + z)(x + y' + z)$
   3ος όρος: $y + z = y + z + xx' = (x + y + z)(x' + y + z)$

χρησιμοποιούμε το Θεώρημα 1β (xx = x) ώστε να διώξουμε ίδιους όρους

$F = (x' + y + z)(x' + y + z')(x + y + z)(x + y' + z)\cancel{(x + y + z)}\cancel{(x' + y + z)}$
$= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$

$\boxed{F = M_0 \cdot M_2 \cdot M_4 \cdot M_5}$

Συνάρτηση Boole

Μετατροπή σε γινόμενο μεγιστόρων - $2^η$ μέθοδος

ακολουθούμε την εξής διαδικασία:

1. κατασκευάζουμε τον πίνακα αληθείας της συνάρτησης, χρησιμοποιώντας την αλγεβρική έκφραση
2. προσδιορίζουμε τους **μεγιστόρους** από τον πίνακα αληθείας

- π.χ. γράψτε τη συνάρτηση Boole ως **γινόμενο μεγιστόρων** $F = xy + x'z$
  1. κατασκευάζουμε τον πίνακα αληθείας βάζοντας την τιμή **1** στην **F**, στις γραμμές όπου:
     - $xy = 11$
     - $xz = 01$
  2. προσδιορίζουμε τους **μεγιστόρους**
     - $F = (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z')$
     - $\boxed{F = M_0 \cdot M_2 \cdot M_4 \cdot M_5}$

				μεγιστόροι	
x	y	z	F	όρος	όνομα
0	0	0	0	$x + y + z$	M_0
0	0	1	1	$x + y + z'$	M_1
0	1	0	0	$x + y' + z$	M_2
0	1	1	1	$x + y' + z'$	M_3
1	0	0	0	$x' + y + z$	M_4
1	0	1	0	$x' + y + z'$	M_5
1	1	0	1	$x' + y' + z$	M_6
1	1	1	1	$x' + y' + z'$	M_7

Συνάρτηση Boole
Κανονική μορφή - Συντομεύσεις

- έστω η συνάρτηση Boole: $F = xyz + xyz' + xy'z + xy'z' + x'y'z$
  $= m_1 + m_4 + m_5 + m_6 + m_7$
  - ✍ η **συντόμευση** της είναι: $\boxed{F(x,y,z) = \Sigma(1,4,5,6,7)}$

    οι μεταβλητές της συνάρτησης, <u>διατεταγμένες</u> όπως εμφανίζονται στους <u>ελαχιστόρους</u>

    οι <u>δείκτες</u> των ελαχιστόρων

- έστω η συνάρτηση Boole: $F = (x + y + z)(x + y' + z)$
  $(x' + y + z)(x' + y + z')$
  $= M_0 \cdot M_2 \cdot M_4 \cdot M_5$
  - ✍ η **συντόμευση** της είναι: $\boxed{F(x,y,z) = \Pi(0,2,4,5)}$

    οι μεταβλητές της συνάρτησης, <u>διατεταγμένες</u> όπως εμφανίζονται στους <u>μεγιστόρους</u>

    οι <u>δείκτες</u> των μεγιστόρων

Συνάρτηση Boole

Μετατροπή από άθροισμα ελαχιστόρων σε γινόμενο μεγιστόρων

- το συμπλήρωμα μιας συνάρτησης που εκφράζεται ως άθροισμα ελαχιστόρων ισούται με το άθροισμα των ελαχιστόρων που λείπουν από την αρχική παράσταση

π.χ. έστω: $F(x,y,z) = \Sigma(1,4,5,6,7)$ ← οι δείκτες των ελαχιστόρων που δίνουν τιμή 1 στην F

- τότε: $F'(x,y,z) = \Sigma(0,2,3)$ ← οι δείκτες των ελαχιστόρων που δίνουν τιμή 0 στην F

- επίσης ισχύει:

$$\begin{aligned} F(x,y,z) &= (F')'(x,y,z) \\ &= (m_0 + m_2 + m_3)' \\ &= m_0' \cdot m_2' \cdot m_3' && \leftarrow \text{Θεώρημα 5a (DeMorgan)} \\ &= M_0 \cdot M_2 \cdot M_3 && \leftarrow \text{καθώς ισχύει ότι: } m'_j = M_j \\ &= \Pi(0,2,3) \end{aligned}$$

- επομένως: $F(x,y,z) = \Sigma(1,4,5,6,7) = \Pi(0,2,3)$

Συνάρτηση Boole

Μετατροπή μεταξύ κανονικών μορφών

- στην προηγούμενη διαφάνεια παρουσιάστηκε η μετατροπή μίας συνάρτησης από άθροισμα ελαχιστόρων σε γινόμενο μεγιστόρων
  - η αντίστροφη μετατροπή είναι αντίστοιχη

✍ γενικά, για να μετατρέψουμε τη μία κανονική μορφή στην άλλη

1. εναλλάσουμε τα σύμβολα **Σ** και **Π**
2. στη μορφή που προκύπτει, αντικαθιστούμε τους δείκτες με όσους έλειπαν από την αρχική μορφή
   - για να εντοπίσουμε σωστά τους δείκτες που λείπουν πρέπει να γνωρίζουμε το πλήθος των μεταβλητών (n) → καθώς υπάρχουν 2^n δείκτες

π.χ.

$$F_1(x,y,z) = \Sigma(1,4,5,6,7) = \Pi(0,2,3)$$

$$F_2(x,y) = \Pi(0,1,2) = \Sigma(3)$$

$$F_3(x,y,z,w) = \Sigma(0,4,7) = \Pi(1,2,3,5,6,8,9,10,11,12,13,14,15)$$

Συναρτήσεις Boole

Κανονικές μορφές - Συμπέρασμα

- ✍ οι κανονικές μορφές σπάνια έχουν τον ελάχιστο αριθμό παραγόντων
  - ? *γιατί;* → εξαιτίας του ορισμού του ελαχιστόρου/μεγιστόρου
    - περιέχει όλες τις ανεξάρτητες μεταβλητές στην κανονική τους μορφή ή ως συμπλήρωμα

- ☞ εκτός από τις κανονικές μορφές, υπάρχουν και οι πρότυπες μορφές
  - κάθε όρος μπορεί να αποτελείται απο οποιονδήποτε αριθμό παραγόντων
  - υπάρχουν δύο τύποι
    1. άθροισμα γινομένων
    2. γινόμενο αθροισμάτων

Συναρτήσεις Boole

Πρότυπες μορφές: άθροισμα γινομένων ή γινόμενο αθροισμάτων

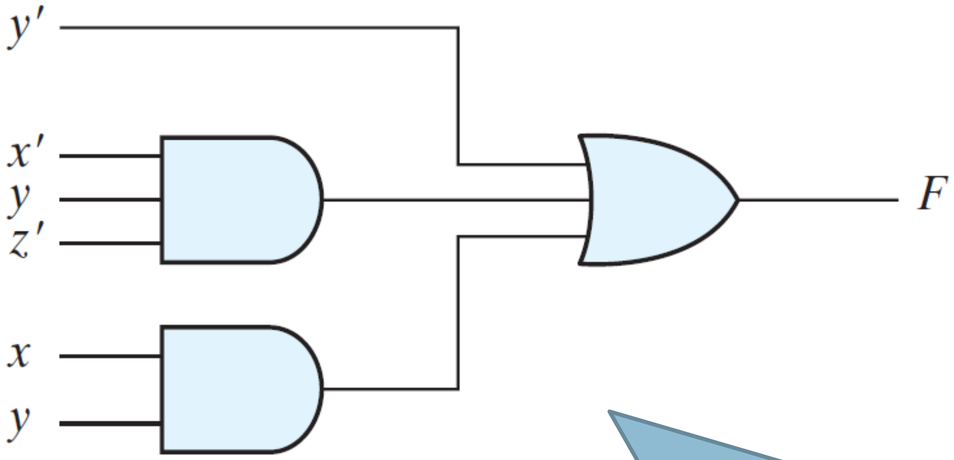
Συναρτήσεις Boole

Πρότυπες μορφές: Άθροισμα γινομένων

- μία συνάρτηση Boole που περιέχει όρους γινομένου (πράξη **AND**), για τους οποίους η λέξη άθροισμα αναφέρεται στην πράξη **OR**
  - κάθε όρος μπορεί να περιέχει οποιοδήποτε αριθμό παραγόντων
  - το αντίστοιχο λογικό διάγραμμα αποτελείται από μία ομάδα πυλών **AND** που τροφοδοτούν μία πύλη **OR**
    - ✍ **υλοποίηση δύο επιπέδων**

π.χ. $F = y' + xy + x'yz'$

η F αποτελείται από τρεις όρους, με έναν, δύο και τρεις παράγοντες, αντίστοιχα



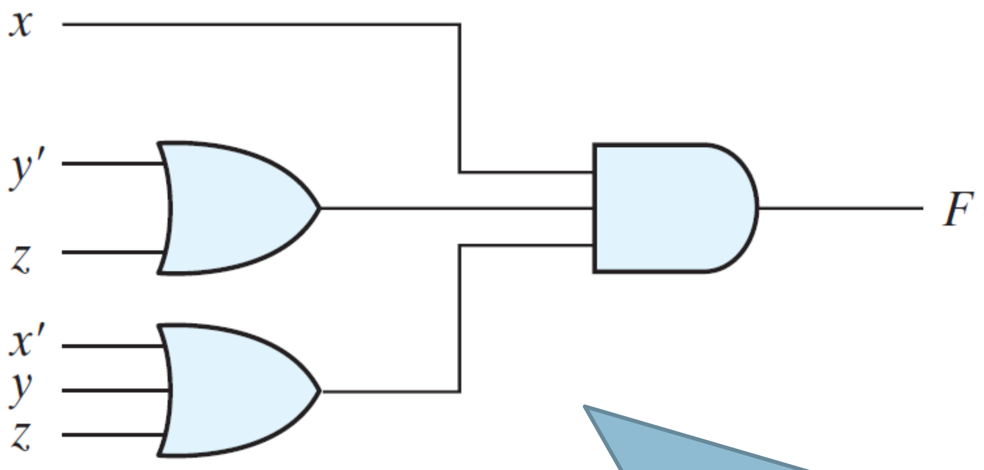
θεωρούμε ότι έχουμε διαθέσιμα τα συμπληρώματα των εισόδων → δε χρησιμοποιούμε αντιστροφείς

Συναρτήσεις Boole
Πρότυπες μορφές: Γινόμενο αθροισμάτων

- μία συνάρτηση Boole που περιέχει όρους αθροισμάτων (πράξη **OR**), για τους οποίους η λέξη γινόμενο αναφέρεται στην πράξη **AND**
  - κάθε όρος μπορεί να περιέχει οποιοδήποτε αριθμό παραγόντων
  - το αντίστοιχο λογικό διάγραμμα αποτελείται από μία ομάδα πυλών **OR** που τροφοδοτούν μία πύλη **AND**
    - ✍ **υλοποίηση δύο επιπέδων**

π.χ. $F = x(y' + z)(x' + y + z')$

η F αποτελείται από τρεις όρους, με έναν, δύο και τρεις παράγοντες, αντίστοιχα



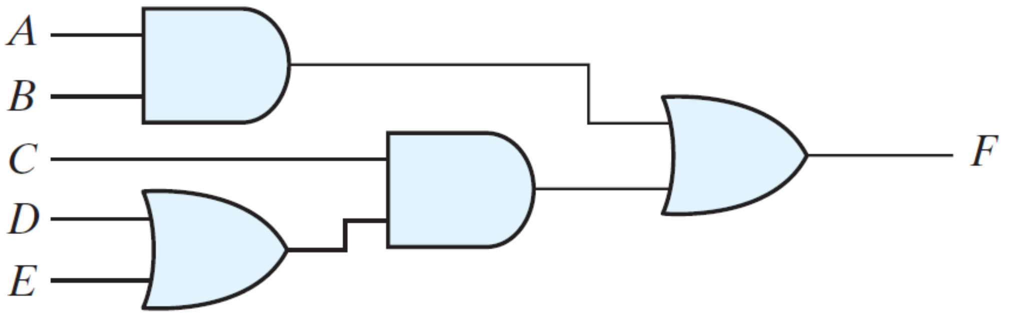
θεωρούμε ότι έχουμε διαθέσιμα τα συμπληρώματα των εισόδων → δε χρησιμοποιούμε αντιστροφείς

Συναρτήσεις Boole

Πρότυπη μορφή - Μετατροπή - Παράδειγμα

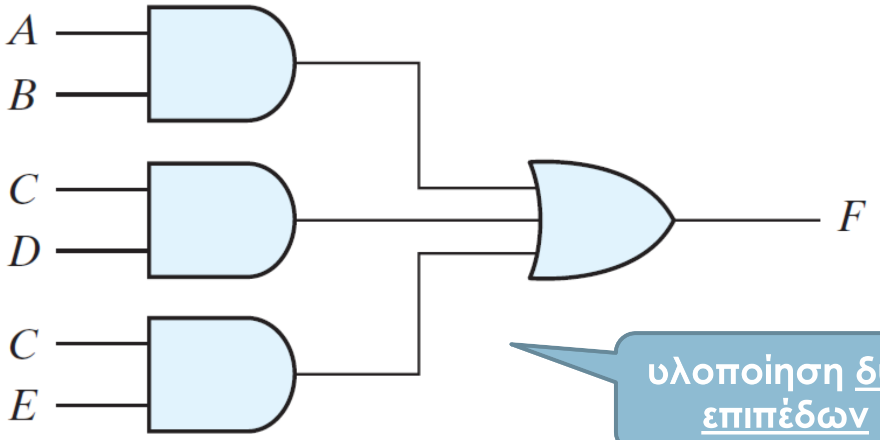
- μία συνάρτηση Boole μπορεί να εκφραστεί σε μία μη πρότυπη μορφή
  - δεν είναι ούτε άθροισμα γινομένων, ούτε γινόμενο αθροισμάτων

π.χ. **F = AB + C(D + E)**



✍ η συνάρτηση μπορεί να μετατραπεί σε πρότυπη μορφή, χρησιμοποιώντας την επιμεριστική ιδιότητα (Αξίωμα 4)

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



υλοποίηση δύο επιπέδων

Λογικές πράξεις

Λογικές πράξεις

- το πλήθος όλων των δυνατών συναρτήσεων **n** μεταβλητών είναι 2^{2^n} , επειδή:
  1. **n** μεταβλητές μπορούν να σχηματίσουν 2^n διαφορετικούς συνδυασμούς ανεξάρτητων μεταβλητών (→ δηλαδή, **ελαχιστόρους**)
  2. **κάθε** συνάρτηση μπορεί να πάρει τιμή **0** ή **1** σε κάθε **ελαχιστόρο**

- έτσι, για **n=2** μεταβλητές το πλήθος των συναρτήσεων Boole ή λογικών πράξεων είναι **16**
  - οι συναρτήσεις **AND** και **OR** είναι δύο από αυτές

x	y	F_0	F_1 (AND)	F_2	F_3	F_4	F_5	F_6	F_7 (OR)	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Πίνακας αληθείας για τις 16 συναρτήσεις δύο δυαδικών μεταβλητών

Λογικές πράξεις
Εκφράσεις Boole

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Συναρτήσεις Boole	Σύμβολο τελεστή	Όνομα	Σχόλια
F_0 = 0		Null	δυαδική σταθερά 0
F_1 = xy	$x \cdot y$	AND	x και y
F_2 = xy'	x/y	αποτροπή	x αλλά όχι y
F_3 = x		μεταφορά	x
F_4 = x'y	y/x	αποτροπή	y αλλά όχι x
F_5 = y		μεταφορά	y
F_6 = xy' + x'y	$x \oplus y$	αποκλειστικό-OR	x ή y, αλλά όχι και τα δύο
F_7 = x + y	$x + y$	OR	x ή y

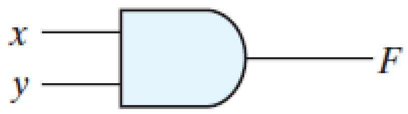
Συναρτήσεις Boole	Σύμβολο τελεστή	Όνομα	Σχόλια
F_8 = (x+y)'	$x \downarrow y$	NOR	Όχι-OR
F_9 = xy + x'y'	$(x \oplus y)'$	ισοδυναμία	x ίσον y
F_{10} = y'	y'	συμπλήρωμα	Όχι y
F_{11} = x + y'	$x \subset y$	συνεπαγωγή	Αν y, τότε x
F_{12} = x'	x'	συμπλήρωμα	Όχι x
F_{13} = x' + y	$x \supset y$	συνεπαγωγή	Αν x, τότε y
F_{14} = (xy)'	$x \uparrow y$	NAND	Όχι-AND
F_{15} = 1		ταυτότητα	δυαδική σταθερά 1

Λογικές πύλες

Λογικές πύλες
Δύο εισόδων - AND, OR, NOT

- οι συναρτήσεις Boole εκφράζονται με χρήση των τελεστών **AND**, **OR** και **NOT**

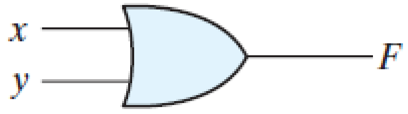
AND



$F = x \cdot y$

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

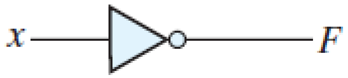
OR



$F = x + y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

Αντιστροφέας



$F = x'$

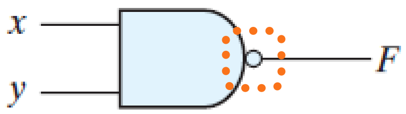
x	F
0	1
1	0

Λογικές πύλες

Δύο εισόδων - NAND, NOR

- οι συναρτήσεις **NAND** και **NOR** είναι το συμπλήρωμα των συναρτήσεων AND και OR, αντίστοιχα
- έτσι, η διαφορά των πυλών **NAND** και **NOR** από τις πύλες AND και OR, αντίστοιχα, είναι ο μικρός κύκλος

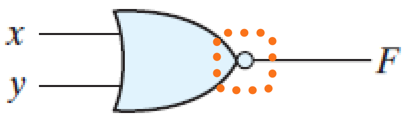
NAND



$F = (xy)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

NOR



$F = (x + y)'$

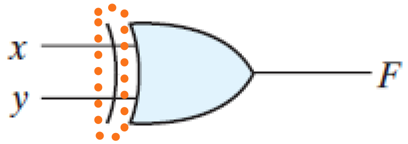
x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

- οι πύλες **NAND** και **NOR**
  - χρησιμοποιούνται πιο συχνά από τις πύλες AND και OR
  - κατασκευάζονται εύκολα
- τα ψηφιακά κυκλώματα μπορούν να υλοποιηθούν πολύ εύκολα με πύλες **NAND** και **NOR**

Λογικές πύλες
Δύο εισόδων - XOR, XNOR

- η διαφορά των πυλών **XOR** και **XNOR** από τις πύλες OR και NOR, αντίστοιχα, είναι η επιπλέον καμπύλη γραμμή στην είσοδο

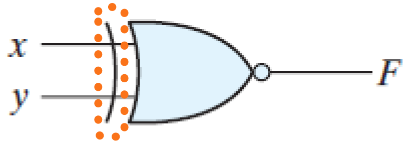
Αποκλειστικό-OR
(XOR)



$$F = xy' + x'y = x \oplus y$$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Αποκλειστικό-NOR
(XNOR)
ή ισοδυναμία



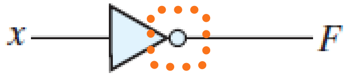
$$F = xy + x'y' = (x \oplus y)'$$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Λογικές πύλες
Μίας εισόδου - NOT, Απομονωτής

- η μεταξύ τους διαφορά είναι ο μικρός κύκλος

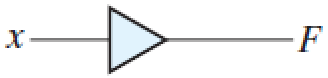
Αντιστροφέας



$F = x'$

x	F
0	1
1	0

Απομονωτής

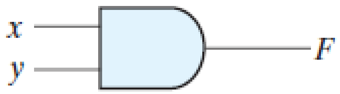


$F = x$

x	F
0	0
1	1

Λογικές πύλες
Σύνοψη

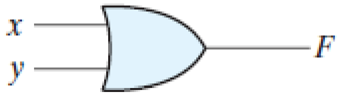
AND



$F = x \cdot y$

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

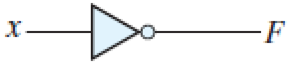
OR



$F = x + y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

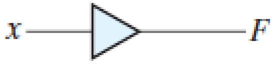
Inverter



$F = x'$

x	F
0	1
1	0

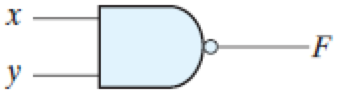
Buffer



$F = x$

x	F
0	0
1	1

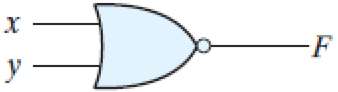
NAND



$F = (xy)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

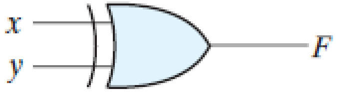
NOR



$F = (x + y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

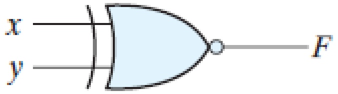
Exclusive-OR (XOR)



$F = xy' + x'y = x \oplus y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR or equivalence



$F = xy + x'y' = (x \oplus y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Λογικές πύλες

Πολλών εισόδων - AND & OR

- κάθε πύλη (δύο εισόδων) μπορεί να επεκταθεί ώστε να αποκτήσει περισσότερες εισόδους
  - αρκεί η δυαδική πράξη που παριστάνει να είναι αντιμεταθετική και προσεταιριστική

- οι πράξεις **AND** και **OR** έχουν αυτές τις ιδιότητες

  π.χ. για την **OR** ισχύουν:

$x + y = y + x$ (αντιμεταθετική ιδιότητα)

$(x + y) + z = x + (y + z)$ (προσεταιριστική ιδιότητα)

Λογικές πύλες
Πολλών εισόδων - NAND & NOR

- κάθε πύλη (δύο εισόδων) μπορεί να επεκταθεί ώστε να αποκτήσει περισσότερες εισόδους
  - αρκεί η δυαδική πράξη που παριστάνει να είναι αντιμεταθετική και προσεταιριστική

- οι πράξεις **NAND** και **NOR**
  - ✓ έχουν την αντιμεταθετική ιδιότητα
  - ✗ δεν έχουν την προσεταιριστική ιδιότητα

π.χ. για την **NOR**:

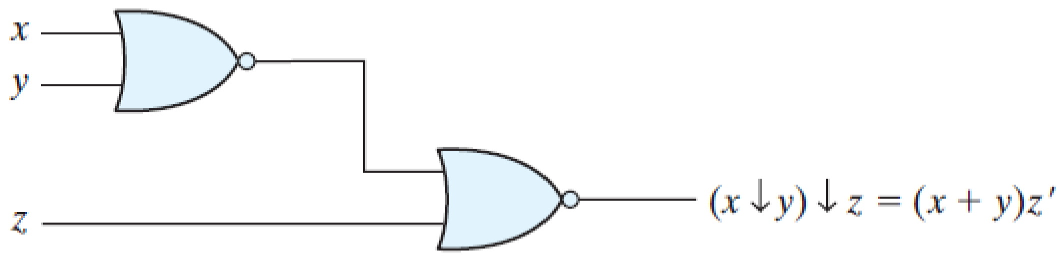
$$\left.\begin{array}{l} (x \downarrow y) \downarrow z = ((x + y)' + z)' = (x + y)z' = xz' + yz' \\ x \downarrow (y \downarrow z) = (x + (y + z)')' = x'(y + z) = x'y + x'z \end{array}\right\} \Rightarrow (x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$$

Λογικές πύλες

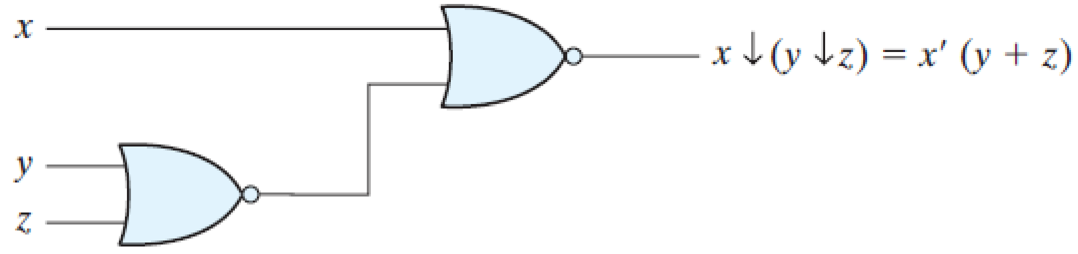
Πολλών εισόδων - NAND & NOR (II)

- οι πράξεις **NAND** και **NOR** δεν έχουν την προσεταιριστική ιδιότητα
  - γραφική απεικόνιση για την **NOR**

πύλη **NAND** τριών εισόδων χρησιμοποιώντας πύλες **NAND** δύο εισόδων

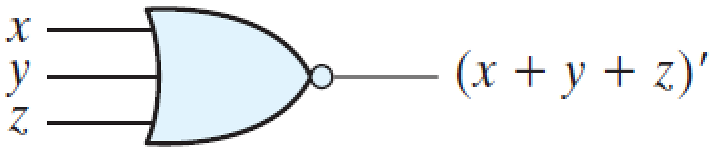


πύλη **NAND** τριών εισόδων χρησιμοποιώντας πύλες **NAND** δύο εισόδων



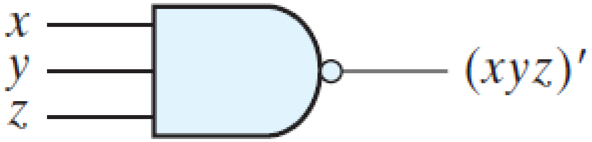
Λογικές πύλες
Πολλών εισόδων - NAND & NOR (III)

- οι πράξεις **NAND** και **NOR** δεν έχουν την προσεταιριστική ιδιότητα
- για να ξεπεράσουμε αυτή τη δυσκολία → ορίζουμε την πύλη **NOR** (ή **NAND**) πολλών εισόδων ως το συμπλήρωμα της αντίστοιχης πύλης **OR** (ή **AND**)
  - οπότε, εξ ορισμού ισχύει:
    - $x \downarrow y \downarrow z = (x + y + z)'$



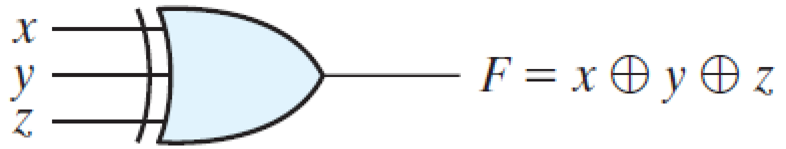
πύλη **NOR**
τριών εισόδων

    - $x \uparrow y \uparrow z = (xyz)'$



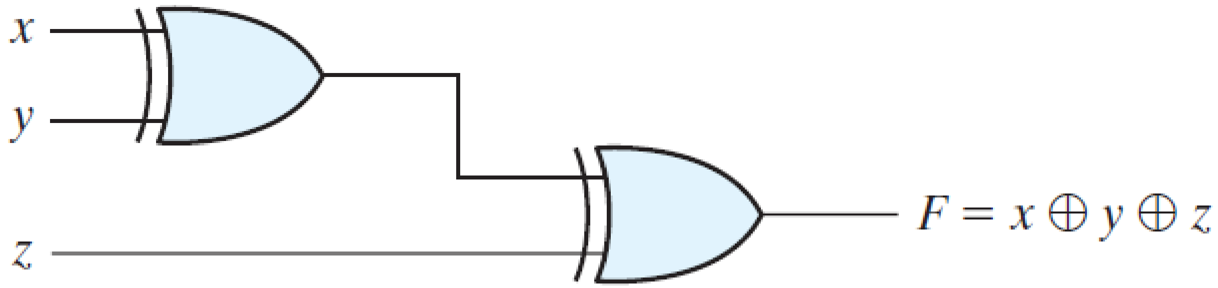
πύλη **NAND**
τριών εισόδων

Λογικές πύλες
Πολλών εισόδων - XOR



πύλη **XOR** τριών εισόδων

- κάθε πύλη (δύο εισόδων) μπορεί να επεκταθεί ώστε να αποκτήσει περισσότερες εισόδους
  - αρκεί η δυαδική πράξη που παριστάνει να είναι αντιμεταθετική και προσεταιριστική
- η πράξη **XOR** ικανοποιεί και τις δύο αυτές ιδιότητες
  - είναι μία περιττή συνάρτηση → παίρνει τιμή **1** όταν και μόνο όταν στις μεταβλητές εισόδου εμφανιστεί περιττός αριθμών άσων (**1**)

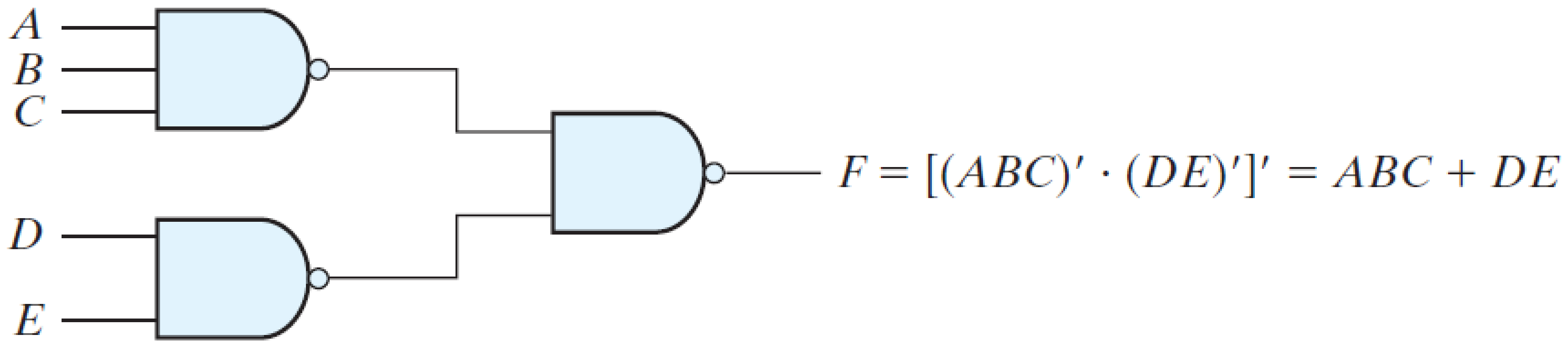


πύλη **XOR** τριών εισόδων χρησιμοποιώντας πύλες **XOR** δύο εισόδων

x	y	z	F
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

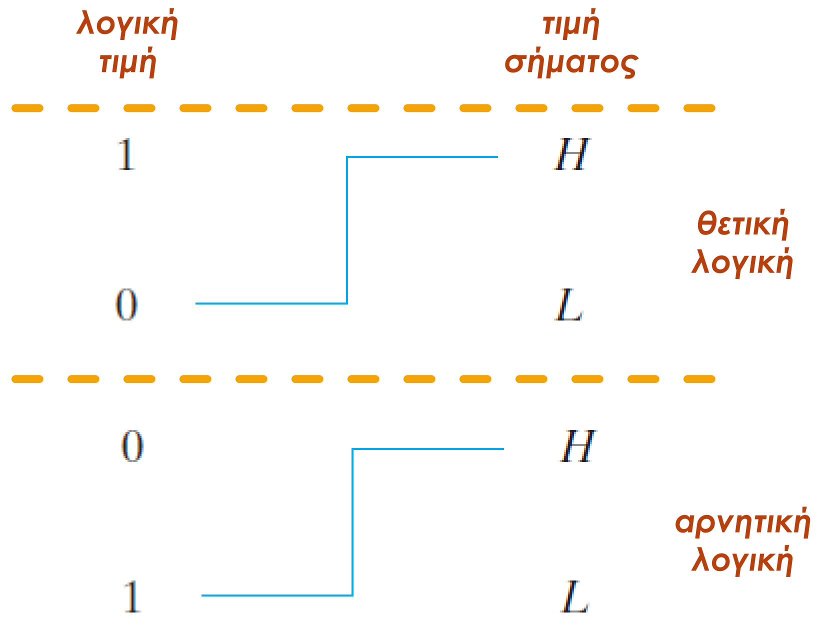
Λογικές πύλες
NAND

- ✍ τα ψηφιακά κυκλώματα μπορούν να υλοποιηθούν πολύ εύκολα με πύλες **NAND**
  - ▶ π.χ. μία έκφραση σε πρότυπη μορφή αθροίσματος γινομένων: **F = ABC + DE** μπορεί να υλοποιηθεί μόνο με πύλες **NAND**



Λογικές πύλες
Θετική και αρνητική λογική

- στο ψηφιακό σύστημα, όλες οι τιμές των σημάτων είναι διακριτές (`0` ή `1`)
- τα ηλεκτρικά σήματα (τάση, ρεύμα) είναι αναλογικά (ή συνεχή)
  - π.χ. τιμές από `0` έως 3V
- σε οποιαδήποτε χρονική στιγμή, το δυαδικό σήμα στις εισόδους ή εξόδους οποιασδήποτε πύλης μπορεί να έχει μόνο μία από τις δύο τιμές
  - εξαιρουμένων των περιόδων μετάβασης
- υπάρχουν δύο διαφορετικοί τρόποι αντιστοίχισης επιπέδου σήματος (τιμή τάσης) στη λογική τιμή
  1. **Θετική λογική**:
     επιλογή του υψηλότερου επιπέδου (**H**) για τη λογική τιμή **1**
  2. **αρνητική λογική**:
     επιλογή του χαμηλότερου επιπέδου (**L**) για τη λογική τιμή **1**



Λογικές πύλες
Θετική και αρνητική λογική (ΙΙ)

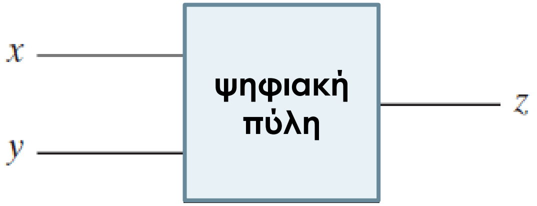
- η λειτουργία των ηλεκτρονικών ψηφιακών πυλών ορίζεται με χρήση των τιμών σήματος **H** και **L**
- ο χρήστης αποφασίζει εάν θα χρησιμοποιήσει θετική ή αρνητική λογική

π.χ.

πίνακας αληθείας

x	*y*	*z*
L	*L*	*L*
L	*H*	*L*
H	*L*	*L*
H	*H*	*H*

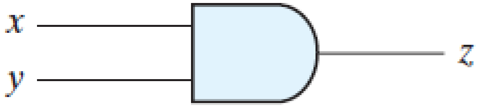
διάγραμμα ψηφιακής πύλης



τα μικρά τρίγωνα αποτελούν ένδειξη αρνητικής πολικότητας

θετική λογική

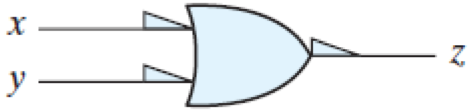
x	*y*	*z*
0	0	0
0	1	0
1	0	0
1	1	1



πύλη **AND**
θετικής λογικής

αρνητική λογική

x	*y*	*z*
1	1	1
1	0	1
0	1	1
0	0	0



πύλη **OR**
αρνητικής λογικής

Λογικές πύλες
Θετική και αρνητική λογική - Σύνοψη

✍ από την προηγούμενη διαφάνεια → η ίδια φυσική πύλη μπορεί να λειτουργήσει ως πύλη **AND** θετικής λογικής ή ως πύλη **OR** αρνητικής λογικής

✍ η αλλαγή* όλων των ακροδεκτών από μία πολικότητα στην άλλη,

*δηλαδή η μετατροπή από τη θετική στην αρνητική λογική (και αντίστροφα)

αλλάζει τα **1** σε **0** και τα **0** σε **1**

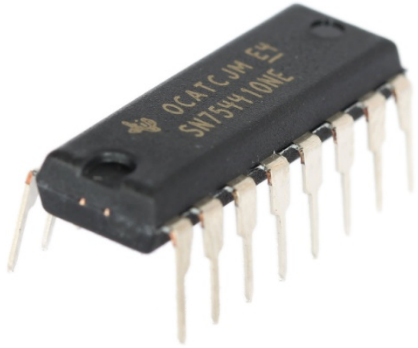
→ οπότε, παράγεται η δυϊκή μορφή μιας συνάρτησης

→ άρα, όλες οι πράξεις **AND** μετατρέπονται σε πράξεις **OR** , και αντίστροφα

Ολοκληρωμένα κυκλώματα

Ολοκληρωμένο κύκλωμα

- αποτελείται από έναν κρύσταλλο ημιαγωγού πυριτίου → τσιπ (chip)
  - περιέχει διασυνδεδεμένα ηλεκτρονικά στοιχεία που υλοποιούν τις ψηφιακές πύλες
  - οι πύλες (του τσιπ) σχηματίζουν τα επιθυμητά κυκλώματα
  - το τσιπ τοποθετείται σε κεραμική ή πλαστική θήκη → συσκευασία
- οι ακροδέκτες του ενώνονται σε εξωτερικούς της θήκης ακροδέκτες → σχηματισμός ολοκληρωμένου κυκλώματος
  - μπορεί να υπάρχουν (από 14) έως και χιλιάδες ακροδέκτες
- στην επιφάνειά του χαράζεται ένας αριθμός → ταυτότητα



✍ ταξινομούνται βάσει:

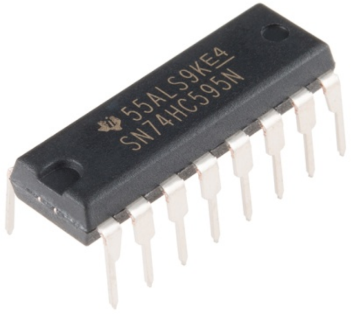
1. του επιπέδου ολοκλήρωσης
2. της οικογένειας ψηφιακής λογικής

Ολοκληρωμένα κυκλώματα
Επίπεδα ολοκλήρωσης

ταξινόμηση σύμφωνα με την πολυπλοκότητα των κυκλωμάτων τους (→ δηλαδή τον αριθμό των πυλών σε κάθε συσκευασία)

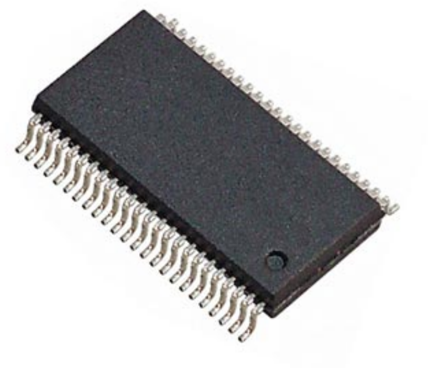
1. **μικρής κλίμακας ολοκλήρωσης** (small-scale integration, **SSI**)
   - αρκετές ανεξάρτητες πύλες σε μία συσκευασία
     - συνήθως, **λιγότερες από 10**
   - είσοδοι/έξοδοι πυλών συνδέονται απευθείας στους ακροδέκτες της συσκευασίας

2. **μέτριας κλίμακας ολοκλήρωσης** (medium-scale integration, **MSI**)
   - **10 έως 1000 πύλες** σε μία συσκευασία
   - εκτελούν στοιχειώδεις πράξεις
   - π.χ. αποκωδικοποιητές, αθροιστές, πολυπλέκτες, καταχωρητές και μετρητές

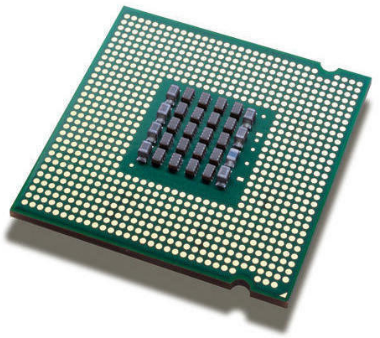


Ολοκληρωμένα κυκλώματα

Επίπεδα ολοκλήρωσης (ΙΙ)



3. **μεγάλης κλίμακας ολοκλήρωσης** (large-scale integrations, **LSI**)
   - αρκετές χιλιάδες πύλες, σε μία συσκευασία
   - π.χ. επεξεργαστές, τσιπ μνήμης και προγραμματίσιμες λογικές διατάξεις
4. **πολύ μεγάλης κλίμακας ολοκλήρωσης** (very large-scale integrations, **VLSI**)
   - εκατοντάδες χιλιάδες πύλες, σε μία συσκευασία
   - π.χ. μεγάλες διατάξεις μνήμης και σύνθετα τσιπ μικροϋπολογιστών
   - ✍ εξαιτίας του μικρού μεγέθους και του χαμηλού κόστους → έχουν αλλάξει ριζικά την τεχνολογία σχεδίασης υπολογιστικών συστημάτων
     - ✓ δυνατότητα σχεδιασμού υπολογιστικών δομών που παλιότερα ήταν οικονομικά ασύμφορες



Ολοκληρωμένα κυκλώματα
Οικογένειες ψηφιακής λογικής

ταξινόμηση σύμφωνα με την τεχνολογία που κατασκευάζονται

- κάθε οικογένεια έχει το δικό της βασικό ηλεκτρονικό κύκλωμα, βάσει του οποίου αναπτύσσονται σύνθετες ψηφιακές διατάξεις και κυκλώματα
  - μία πύλη **NAND** ή μία πύλη **NOR**
- ονομασίες ανάλογα με τα ηλεκτρονικά στοιχεία που χρησιμοποιούνται για την κατασκευή του βασικού κυκλώματος

1. TTL transistor-transistor logic (λογική τρανζίστορ-τρανζίστορ)
2. ECL emitter-coupled logic (λογική σύζευξης εκπομπού)
3. MOS metal-oxide-semiconductor (τεχνολογία μετάλλου-οξειδίου-ημιαγωγού)
4. CMOS complementary metal-oxide-semiconductor (τεχνολογία συμπληρωματικού μετάλλου-οξειδίου-ημιαγωγού)

υψηλή ταχύτητα

υψηλή πυκνότητα

χαμηλή κατανάλωση ενέργειας → κυρίαρχη οικογένεια (π.χ. ψηφιακές φωτογραφικές μηχανές και φορητές συσκευές)

Σύνοψη

- Δυαδική λογική
  - πίνακες αληθείας
- Άλγεβρα Boole
  - βασικοί ορισμοί
  - άλγεβρα Boole δύο τιμών
  - ιδιότητα δϋισμού, αξιώματα και θεωρήματα (αποδείξεις)
  - προτεραιότητα τελεστών
- Συναρτήσεις Boole
  - αναπαράσταση με πίνακα
  - πολλαπλές αλγεβρικές εκφράσεις → πολλαπλά λογικά κυκλώματα
  - απλοποίηση
  - συμπλήρωμα
  - κανονική & πρότυπη μορφή
- Λογικές πράξεις
- Λογικές πύλες
  - πύλες δύο εισόδων
  - πύλες πολλαπλών εισόδων
  - θετική και αρνητική λογική