Άλγεβρα Boole και Λογικές Πύλες

Βαρτζιώτης Φώτιος

Άλγεβρα

- Τι είναι η Άλγεβρα
  - Ένα επαγωγικό μαθηματικό σύστημα που αποτελείται από:
    - Ένα σύνολο στοιχείων
    - Ένα σύνολο τελεστών
    - Έναν αριθμό αξιωμάτων (μη αποδείξιμων)
- Γιατί είναι σημαντική;
  - Καθορίζει τους κανόνες των "Πράξεων - Υπολογισμών"
- Παράδειγμα: Αριθμητική Φυσικών Αριθμών
  - Σύνολο στοιχείων: $N = \{1,2,3,4,\ldots\}$
  - Τελεστές: +, –, *
  - Αξιώματα: Κλειστότητα, Προσεταιριστικός κανόνας, Επιμεριστικός κανόνας, Αντιμεταθετικός κανόνας, Ουδέτερο στοιχείο, Αντίστροφος κ.α.
- Σημείωση: Οι τελεστές δύο εισόδων καλούνται και <u>**δυαδικοί**</u>
  - Δεν σημαίνει ότι αναφέρονται σε δυαδικούς αριθμούς!
  - Οι τελεστές μιας εισόδου καλούνται <u>**μοναδιαίοι**</u>

Βασικοί Ορισμοί

- Ένα σύνολο ορίζεται σαν μια συλλογή από οντότητες, που (συνήθως) έχουν μια κοινή ιδιότητα.
  - S: Σύνολο, x και y: οντότητες
  - Παράδειγμα: $S = \{1, 2, 3, 4\}$
    - Αν x = 2, τότε $x \in S$.
    - Αν $y = 5$, τότε $y \notin S$.
- Ένας δυαδικός τελεστής που ορίζεται σε ένα σύνολο στοιχείων S, είναι ένας κανόνας ο οποίος καθορίζει μία αντιστοίχιση κάθε ζεύγους στοιχείων του S σε ένα μοναδικό στοιχείο του S.
  - Παράδειγμα: Έστω ένα σύνολο S, θεωρείστε $a*b = c$ και το '*' είναι ένας δυαδικός τελεστής.
  - Αν το ζεύγος (a, b) μέσω του τελεστή '*' παίρνει την τιμή c και τα $a, b, c \in S$, τότε το '*' είναι ένας δυαδικός τελεστής του S.
  - Αντίθετα, το '*' δεν είναι δυαδικός τελεστής του συνόλου S εάν $a, b \in S$, και το $c \notin S$.

Βασικοί Ορισμοί

- Τα πιο συνηθισμένα αξιώματα που χρησιμοποιούνται για τη διαμόρφωση διαφόρων αλγεβρικών δομών είναι τα εξής:

1. Κλειστότητα (Closure): Ένα σύνολο ***S*** θεωρείται κλειστό ως προς ένα δυαδικό τελεστή εάν, για κάθε ζεύγος στοιχείων του ***S***, ο δυαδικός τελεστής καθορίζει έναν κανόνα για την εύρεση ενός μοναδικού στοιχείου που ανήκει επίσης στο *S*.
    - *Παράδειγμα, το σύνολο N των φυσικών αριθμών, N={1,2,3,...}, είναι κλειστό ως προς τον δυαδικό τελεστή '+' που αντιστοιχεί στον κανόνα της αριθμητικής πρόσθεσης, αφού, για κάθε a, b$\in$N, υπάρχει ένα μοναδικό στοιχείο c$\in$N τέτοιο ώστε:*
        - *a+b = c*
        - Αντίθετα, ο δυαδικός τελεστής '–' δεν είναι κλειστός ως προς το *N*, επειδή 2-3 = -1 και τα 2, 3 $\in$*N*, αλλά το (-1)$\notin$*N*.

Βασικοί Ορισμοί

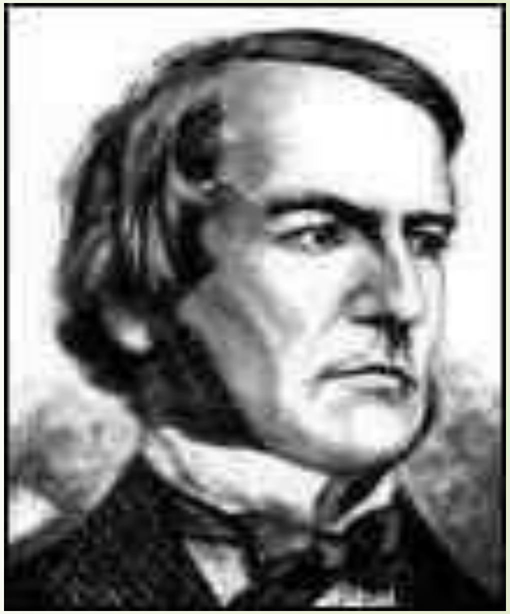
2. Προσεταιριστικός Κανόνας (Associative law): ένας δυαδικός τελεστής ‘*’ ορισμένος σε ένα σύνολο **S**, είναι προσεταιριστικός όταν:
   - $(x * y) * z = x * (y * z)$ $\forall$ (για κάθε) $x, y, z \in S$
     - $(x+y)+z = x+(y+z)$
3. Αντιμεταθετικός Κανόνας (Commutative law): ένας δυαδικός τελεστής ‘*’ ορισμένος σε ένα σύνολο ***S***, είναι αντιμεταθετικός όταν:
   - $x * y = y * x$ $\forall x, y \in S$
     - $x+y = y+x$

Βασικοί Ορισμοί

4. *Ουδέτερο Στοιχείο (Identity element)*: Λέμε ότι ένα σύνολο ***S*** έχει το ουδέτερο στοιχείο **e** ως προς μία δυαδική πράξη ‘*****’, ορισμένη στο ***S***, εάν υπάρχει στοιχείο **e**∈***S*** με την εξής ιδιότητα:
   - $e * x = x * e = x \ \forall x \in S$
     - $0+x = x+0 = x \ \forall x \in I$. $I = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$.
     - $1*x = x*1 = x \ \forall \ x \in I$. $I = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$.
5. *Αντίστροφο (Inverse)*: Λέμε ότι σε ένα σύνολο ***S***, το οποίο έχει το ουδέτερο στοιχείο **e** ως προς έναν δυαδικό τελεστή ‘*’, υπάρχει αντίστροφο, όταν για κάθε **x**∈**S** υπάρχει ένα στοιχείο **y**∈**S**, τέτοιο ώστε:
   - $x * y = e$
     - Έστω ο τελεστής ‘+’ στο I, με $e = 0$, ο αντίστροφος του στοιχείου a είναι $(-a)$, αφού $a+(-a) = 0$.
6. *Επιμεριστικός Κανόνας (Distributive law)*: Εάν τα ‘*’ και ‘•’ είναι δύο δυαδικοί τελεστές σε ένα σύνολο ***S***, λέμε ότι το ‘*’ είναι επιμεριστικό σε σχέση με το ‘•’, όταν:
   - $x * (y \cdot z) = (x * y) \cdot (x * z)$

George Boole

- Πατέρας της Boolean άλγεβρας
- Σκέφτηκε ένα είδος γλωσσικής άλγεβρας με τρεις βασικές λειτουργίες, οι οποίες ήταν (και εξακολουθούν να είναι) οι AND, OR και NOT. Οι παραπάνω τρεις λειτουργίες μπορούσαν να υποστηρίξουν από μόνες τους την εκτέλεση συγκρίσεων και άλλων βασικών μαθηματικών λειτουργιών.
- Το σύστημα του Boole (που περιγράφεται λεπτομερώς στο 'An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities', 1854) βασίστηκε σε μια δυαδική προσέγγιση, που επεξεργάζεται μόνο δύο οντότητες / αντικείμενα - τα ναι-όχι, αληθής-ψευδής, on - off, μηδέν (0) – ένα (1).
- Παραδόξως, δεδομένης της θέσης του στην ακαδημαϊκή κοινότητα, η ιδέα του Boole είτε επικρίθηκε είτε αγνοήθηκε εντελώς
- Τελικά, ένας φοιτητής, ο Claude Shannon (πατέρας της θεωρίας πληροφορίας, 1916 - 2001), πήρε την ιδέα και την ανέπτυξε.



George Boole (1815 - 1864)

Αξιωματικός Ορισμός της Boolean Άλγεβρας

- Χρειάζεται να ορίσουμε μια αλγεβρική δομή για τις δυαδικές τιμές
  - Αναπτύχθηκε από τον George Boole το 1854
- Ο Huntington διατύπωσε ένα σύνολο αξιωμάτων για την Boolean άλγεβρα (1904):
- *Έστω Β* = {0, 1} με δύο δυαδικούς τελεστές, '+' και ' · '
  - Ισχύει η Κλειστότητα ως προς τους τελεστές '+' και ' · '
  - Το 0 είναι το ουδέτερο στοιχείο ως προς τον τελεστή '+' και το 1 για τον τελεστή ' ·'
  - Ισχύει ο αντιμεταθετικός κανόνας ως προς '+' και ' ·'

    $x+y = y+x, \quad x \cdot y = y \cdot x$
  - Ισχύει ο επιμεριστικός κανόνας για τον ' · ' σε σχέση με τον '+', και τον '+' σε σχέση με τον ' ·'

    $x \cdot (y+z) = (x \cdot y)+(x \cdot z)$ *και* $x+(y \cdot z) = (x+y) \cdot (x+z)$
  - Ορίζεται για κάθε στοιχείο **x** το συμπλήρωμα **x'**, **ώστε x+x'=1**, $x \cdot x'=0$
  - Υπάρχουν τουλάχιστον δύο στοιχεία **x,y∈B**, τέτοια ώστε $x \neq y$

Boolean Άλγεβρα

- Ορολογία:
  - *Παράγοντας:* Μια μεταβλητή ή το συμπλήρωμά της
  - *Γινόμενο:* Σύμβολα συνδεδεμένα με τον τελεστή '•'
  - *Άθροισμα:* Σύμβολα συνδεδεμένα με τον τελεστή '+'

Αξιωματικός Ορισμός Άλγεβρας Boole Δύο Τιμών

- *B* = {0, 1}, δύο δυαδικοί τελεστές, ‘**+**’ και ‘ · ’ και ένας μοναδιαίος ‘’’
- Οι κανόνες που διέπουν τη λειτουργία των τελεστών:

AND

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

OR

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

NOT

x	x'
0	1
1	0

1. Κλειστότητα (‘+’ και ‘·’)
2. Ουδέτερα στοιχεία

   (1) +: 0

   (2) · : 1

Αξιωματικός Ορισμός Άλγεβρας Boole δύο τιμών

3. Αντιμεταθετικός Κανόνας
4. Επιμεριστικός Κανόνας

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

Αξιωματικός Ορισμός Άλγεβρας Boole Δύο Τιμών

5. Συμπλήρωμα
   - x+x'=1 → 0+0'=0+1=1; 1+1'=1+0=1
   - x · x'=0 → 0 · 0'=0 · 1=0; 1 · 1'=1 · 0=0
6. Υπάρχουν δύο διακριτά στοιχεία, τα 1 και 0, με 0 ≠ 1

- Σημείωση
  - Ένα σύνολο δύο στοιχείων
  - + : πράξη OR
  - · : πράξη AND
  - ‘: πράξη NOT
  - Η δυαδική λογική είναι ισοδύναμη με την άλγεβρα Bool δύο τιμών.

Δυϊσμός

- **Η αρχή του δυϊσμού** είναι μια σημαντική έννοια. Μας λέει ότι εάν μια παράσταση είναι έγκυρη στην Boolean άλγεβρα, η δυϊκή της έκφραση είναι επίσης έγκυρη.
- Για να σχηματίσετε μια δυϊκή έκφραση μιας παράστασης, αντικαταστήστε όλους τους τελεστές '+' με τελεστές '•'', και όλους τους τελεστές '•' με τελεστές '+', όλα τα 1 με 0 και όλα τα 0 με 1.
- Σχηματίστε τη δυϊκή έκφραση της παράστασης:

  a + (bc) = (a + b)(a + c)
- Ακολουθώντας του κανόνες αντικατάστασης...

  a(b + c) = ab + ac
- Προσοχή στις παρενθέσεις, εάν υπάρχουν.

Βασικά Θεωρήματα

Αξίωμα 2	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Αξίωμα 5	(a) $x + x' = 1$	(b) $x \cdot x' = 0$
Θεώρημα 1	(a) $x + x = x$	(b) $x \cdot x = x$
Θεώρημα 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Θεώρημα 3, διπλή άρνηση	$(x')' = x$	
Αξίωμα 3, αντιμετάθεση	(a) $x + y = y + x$	(b) $xy = yx$
Θεώρημα 4, προσεταιρισμός	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Αξίωμα 4, επιμερισμός	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Θεώρημα 5, DeMorgan	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Θεώρημα 6, απορρόφηση	(a) $x + xy = x$	(b) $x(x + y) = x$

Βασικά Θεωρήματα

- Τα αξιώματα του Huntington ορίζουν κάποιους κανόνες

Αξίωμα 1:	closure
Αξίωμα 2:	(a) $x+0=x$, (b) $x\cdot 1=x$
Αξίωμα 3:	(a) $x+y=y+x$, (b) $x\cdot y=y\cdot x$
Αξίωμα 4:	(a) $x(y+z) = xy+xz$,
	(b) $x+yz = (x+y)(x+z)$
Αξίωμα 5:	(a) $x+x'=1$, (b) $x\cdot x'=0$

- Χρειαζόμαστε επιπλέον κανόνες για να χειριζόμαστε πιο αποτελεσματικά τις αλγεβρικές παραστάσεις
  - Θεωρήματα που προκύπτουν από την εφαρμογή των αξιωμάτων
- Τι είναι το ***Θεώρημα***;
  - Ένας τύπος ή κανόνας που προκύπτει από την εφαρμογή των αξιωμάτων (ή άλλων αποδεδειγμένων Θεωρημάτων)
- Βασικά Θεωρήματα της άλγεβρας Bool
  - Θεώρημα 1 (a): $x + x = x$ (b): $x \cdot x = x$
  - Δείχνουν αυτονόητα, αλλά πρέπει να αποδειχθούν!

Απόδειξη του Θεωρήματος 1: $x+x=x$, $x \bullet x=x$

- Μπορούμε να χρησιμοποιήσουμε μόνο τα αξιώματα Huntington:
- Να δειχθεί ότι: $x+x=x$.

$x+x = (x+x) \cdot 1$	2(b)
$= (x+x)(x+x')$	5(a)
$= x+xx'$	4(b)
$= x+0$	5(b)
$= x$	2(a)

- Με τον ίδιο τρόπο αποδεικνύεται και η δυική της έκφραση: $x \bullet x=x$

Αξιώματα Huntington

Αξίωμα 1: closure

Αξίωμα 2: (a) $x+0=x$, (b) $x \cdot 1=x$

Αξίωμα 3: (a) $x+y=y+x$, (b) $x \cdot y=y \cdot x$

Αξίωμα 4: (a) $x(y+z) = xy+xz$,
(b) $x+yz = (x+y)(x+z)$

Αξίωμα 5: (a) $x+x'=1$, (b) $x \cdot x'=0$

Μπορούμε τώρα να χρησιμοποιήσουμε το Θεώρημα 1 σε μελλοντικές αποδείξεις.

Απόδειξη των Θεωρημάτων 2 και 3

- Θεώρημα 2(a): $x + 1 = 1$

$x + 1 = 1 \cdot (x + 1)$	2(b)
$=(x + x')(x + 1)$	5(a)
$= x + x'\ 1$	4(b)
$= x + x'$	2(b)
$= 1$	5(a)

- Θεώρημα 2(b): $x \cdot 0 = 0$ Δυισμός
- Θεώρημα 3: $(x')' = x$
  - Το αξίωμα 5 ορίζει το συμπλήρωμα του x, $x + x' = 1$ και $x \bullet x' = 0$
  - Το συμπλήρωμα του x' *είναι* x *και επίσης* $(x')'$

Αξιώματα Huntington

Αξίωμα 1: closure

Αξίωμα 2: (a) $x+0=x$, (b) $x \cdot 1=x$

Αξίωμα 3: (a) $x+y=y+x$, (b) $x \cdot y=y \cdot x$

Αξίωμα 4: (a) $x(y+z) = xy+xz$,
(b) $x+yz = (x+y)(x+z)$

Αξίωμα 5: (a) $x+x'=1$, (b) $x \cdot x'=0$

Απόδειξη του Θεωρήματος Απορρόφησης

- Θεώρημα 6(a): $x + xy = x$
  - $x + xy = x \cdot 1 + xy$ 2(b)
    $= x (1 + y)$ 4(a)
    $= x (y + 1)$ 3(a)
    $= x \cdot 1$ Θ2(a)
    $= x$ 2(b)

- Θεώρημα 6(b): $x (x + y) = x$ *Δυισμός*
- Η απορρόφηση αποδεικνύεται και μέσω του Πίνακα Αληθείας

Αξιώματα Huntington
Αξίωμα 1: closure
Αξίωμα 2: (a) $x+0=x$, (b) $x\cdot 1=x$
Αξίωμα 3: (a) $x+y=y+x$, (b) $x\cdot y=y\cdot x$
Αξίωμα 4: (a) $x(y+z) = xy+xz$,
(b) $x+yz = (x+y)(x+z)$
Αξίωμα 5: (a) $x+x'=1$, (b) $x\cdot x'=0$

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

Απόδειξη του Θεωρήματος DeMorgan

- Θεώρημα 5(a): $(x + y)' = x'y'$
- Θεώρημα 5(b): $(xy)' = x' + y'$
- Απόδειξη μέσω πίνακα αληθείας

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

Απόδειξη του Θεωρήματος 7 (Consensus)

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

- **Απόδειξη:**

$$
\begin{aligned}
xy + x'z + yz &= xy + x'z + (x+x')yz \\
&= xy + x'z + xyz + x'yz \\
&= (xy + xyz) + (x'z + x'zy) \\
&= xy + x'z
\end{aligned}
$$

(Το 2 είναι επίσης αληθές λόγω δυισμού)

Προτεραιότητα Τελεστών

- Η προτεραιότητα τελεστών για τον υπολογισμό μιας Boolean έκφρασης είναι:
  - Παρενθέσεις
  - NOT
  - AND
  - OR
- Παραδείγματα
  - $x\,y' + z$
  - $(x\,y + z)'$

Συναρτήσεις Bool

- Μια συνάρτηση Bool μπορεί να περιλαμβάνει:
  - Δυαδικές μεταβλητές
  - Τους Δυαδικούς Τελεστές OR και AND
  - Τον μοναδιαίο τελεστή NOT
  - Παρενθέσεις
- Παραδείγματα
  - $F_1 = x\,y\,z'$
  - $F_2 = x + y'z$
  - $F_3 = x'\,y'\,z + x'\,y\,z + x\,y'$
  - $F_4 = x\,y' + x'\,z$

Συναρτήσεις Bool

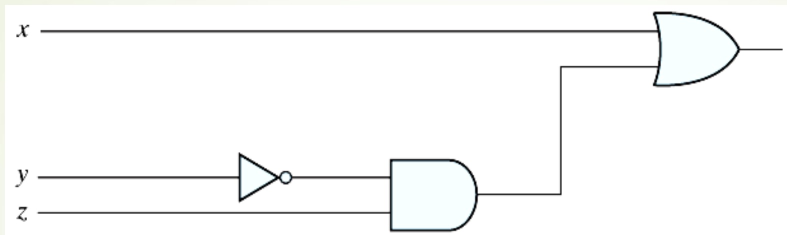
- Πίνακας αληθείας με $2^{n(=3)}$ συνδυασμούς

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

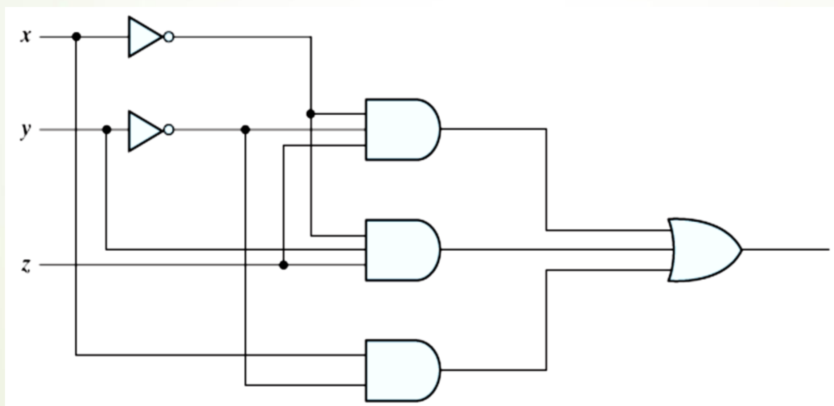
- Δύο διαφορετικές αλγεβρικές εκφράσεις Bool μπορεί να ορίζουν την ίδια συνάρτηση.
- *Π.χ* $F_3 = F_4$

Συναρτήσεις Bool

- Μπορούν να μετασχηματιστούν σε λογικά κυκλώματα
  - Π.χ.

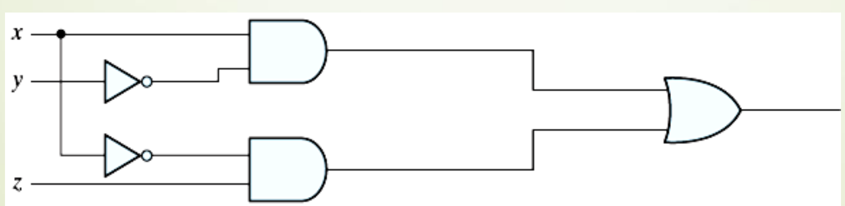


$$F_1 = x + y'z$$



$$F_3 = x'y'z + x'yz + xy'$$

  - Η πιο οικονομικά, με αλγεβρικό χειρισμό,



$$F_3 = xy' + x'z$$

Αλγεβρικός Χειρισμός

- Έχει ως στόχο την ελαχιστοποίηση παραστάσεων Bool
  - Παράγοντας: Μια ανεξάρτητη ή όχι μεταβλητή, δηλώνει μια είσοδο σε λογική πύλη
  - Όρος: Δηλώνει μια υλοποίηση λογικής πύλης
  - Εάν ελαχιστοποιήσουμε τον αριθμό των παραγόντων και των όρων, δημιουργούμε (λογικά) κυκλώματα που απαιτούν λιγότερους μηχανισμούς
  - Είναι δύσκολο πρόβλημα (χωρίς καθορισμένα βήματα και κανόνες)
- Παράδειγμα (2.1)
  1. $x(x'+y) = xx' + xy = 0+xy = xy$
  2. $x+x'y = (x+x')(x+y) = 1\cdot(x+y) = x+y$
  3. $(x+y)(x+y') = x+xy+xy'+yy' = x(1+y+y') = x$
  4. $xy + x'z + yz = xy + x'z + yz(x+x') = xy + x'z + yzx + yzx' = xy(1+z) + x'z(1+y) = xy + x'z$
  5. $(x+y)(x'+z)(y+z) = (x+y)(x'+z)$, με δυισμό από τη συνάρτηση 4. (Θεώρημα 7 με δυισμό)

Συμπλήρωμα μιας Συνάρτησης

- Με εναλλαγή των 0 σε 1 και των 1 σε 0 στην τιμή της F
  - Αλγεβρικά, από τη γενικευμένη μορφή του θεωρήματος DeMorgan
  - $(A+B+C)' = (A+X)'$ έστω $B+C = X$

    $= A'X'$ Θεώρημα 5(a) (DeMorgan)

    $= A'(B+C)'$ αντικατάσταση $B+C = X$

    $= A'(B'C')$ Θεώρημα 5(a) (DeMorgan)

    $= A'B'C'$ προσεταιριστική ιδιότητα
  - $(A+B+C+D+ \ldots +F)' = A'B'C'D' \ldots F'$
  - $(ABCD \ldots F)' = A'+ B'+C'+D' \ldots +F'$
- *Από τη γενικευμένη μορφή των θεωρημάτων DeMorgan συνεπάγεται ότι το συμπλήρωμα μιας συνάρτησης μπορεί να παραχθεί εναλλάσσοντας τους τελεστές AND και OR και συμπληρώνοντας κάθε παράγοντα (δηλαδή εκτελώντας την πράξη NOT σε κάθε παράγοντα.*

Παραδείγματα

- Παράδειγμα (2.2)
  - $F_1' = (x'yz' + x'y'z)' = (x'yz')' (x'y'z)' = (x+y'+z) (x+y+z')$
  - $F_2' = [x(y'z'+yz)]' = x' + (y'z'+yz)' = x' + (y'z')' (yz)'$

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

    $= x' + yz' + y'z$
- Παράδειγμα (2.3): μια πιο απλή διαδικασία
  - Εφαρμόστε δυισμό στη συνάρτηση και συμπληρώστε κάθε παράγοντα

1. $F_1 = x'yz' + x'y'z.$

   Με δυισμό η F_1 είναι $(x'+y+z') (x'+y'+z)$.

   Συμπληρώστε κάθε παράγοντα: $(x+y'+z)(x+y+z') = F_1'$
2. $F_2 = x(y'z' + yz).$

   Με δυισμό η F_2 είναι $x+(y'+z') (y+z)$.

   Συμπληρώστε κάθε παράγοντα : $x'+(y+z)(y'+z') = F_2'$

Κανονικές και Πρότυπες Μορφές

Ελαχιστόροι (Minterms) και Μεγιστόροι (Maxterms)

- Ελαχιστόρος (γινόμενο): Μια πράξη AND που περιλαμβάνει το σύνολο των παραγόντων στην αρχική ή τη συμπληρωματική μορφή τους.
  - Για παράδειγμα, δύο δυαδικές μεταβλητές *x και y,*
    - *xy, xy', x'y, x'y'*
  - *n* μεταβλητές μπορούν να σχηματίσουν 2^n διαφορετικούς ελαχιστόρους.
- Μεγιστόρος (άθροισμα): Μια πράξη OR που περιλαμβάνει το σύνολο των παραγόντων στην αρχική ή τη συμπληρωματική μορφή τους.
  - 2^n maxterms.

Ελαχιστόροι (Minterms) και Μεγιστόροι (Maxterms)

- Κάθε *maxterm* είναι το συμπλήρωμα του αντίστοιχου *minterm*, και αντίστροφα.

			Minterms		Maxterms	
x	**y**	**z**	**Όρος**	**Συμβολισμός**	**Όρος**	**Συμβολισμός**
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)

- Μια συνάρτηση Boole μπορεί να εκφραστεί με:
  - Πίνακα Αληθείας
  - Άθροισμα Ελαχιστόρων
  - $f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$ (Minterms)
  - $f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$ (Minterms)

x	*y*	*z*	F1	F2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Ελαχιστόροι (Minterms) και Μεγιστόροι (Maxterms)

- Το συμπλήρωμα μιας συνάρτησης Bool προκύπτει:
  - Από τα minterms που παράγουν την τιμή 0
  - $f_1' = m_0 + m_2 + m_3 + m_5 + m_6 = x'y'z'+x'yz'+x'yz+xy'z+xyz'$
  - $f_1 = (f_1')' = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y'+z) = M_0 M_2 M_3 M_5 M_6$
  - $f_2 = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z)=M_0M_1M_2M_4$
- Άρα, κάθε συνάρτηση Boole μπορεί να εκφραστεί ως
  - Άθροισμα (OR) ελαχιστόρων (sum (OR) of minterms)
  - Γινόμενο (AND) μεγιστόρων (product (AND) of maxterms)
  - Τότε, λέμε ότι και οι δύο συναρτήσεις είναι σε κανονική μορφή

Άθροισμα Ελαχιστόρων (Sum of Minterms)

- Άθροισμα ελαχιστόρων (sum of minterms): Υπάρχουν 2^n minterms και 2^{2^n} συναρτήσεις με *n* Boolean μεταβλητές.
- Παράδειγμα (2.4): Εκφράστε την συνάρτηση *F = A+BC'* *σαν άθροισμα ελαχιστόρων.*
  - $F = A+B'C = A\ (B+B') + B'C = AB + AB' + B'C = AB(C+C') + AB'(C+C') + (A+A')B'C = ABC+ABC'+AB'C+AB'C'+A'B'C$
  - $F = A'B'C + AB'C' + AB'C + ABC' + ABC = m_1 + m_4 + m_5 + m_6 + m_7$
  - $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$
  - ή, γράψτε πρώτα τον πίνακα αληθείας

A	***B***	***C***	***F***
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

Γινόμενο Μεγιστόρων (Product of Maxterms)

- Γινόμενο μεγιστόρων : με χρήση του επιμεριστικού κανόνα:
  - $x + yz = (x + y)(x + z) = (x+y+zz')(x+z+yy') = (x+y+z)(x+y+z')(x+y'+z)$
- Παράδειγμα (2.5): Εκφράστε την συνάρτηση $F = xy + x'z$ *σαν γινόμενο μεγιστόρων*
  - $F = xy + x'z = (xy + x')(xy + z) = (x+x')(y+x')(x+z)(y+z) = (x'+y)(x+z)(y+z)$
  - $x'+y = x' + y + zz' = (x'+y+z)(x'+y+z')$
  - $F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z') = M_0M_2M_4M_5$
  - $F(x, y, z) = \Pi(0, 2, 4, 5)$

Μετατροπή μεταξύ Κανονικών Μορφών

- Το συμπλήρωμα μιας συνάρτησης που εκφράζεται ως άθροισμα ελαχιστόρων ισούται με το άθροισμα των ελαχιστόρων που λείπουν από την αρχική συνάρτηση.
  - $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$
  - Άρα, $F'(A, B, C) = \Sigma(0, 2, 3)$
  - Από το θεώρημα DeMorgan

    $F(A, B, C) = \Pi(0, 2, 3)$

    $F'(A, B, C) = \Pi(1, 4, 5, 6, 7)$
  - $m_j' = M_j$
  - Άθροισμα ελαχιστόρων = γινόμενο μεγιστόρων
  - Εναλλάξτε τα σύμβολα Σ και Π και τοποθετήστε στη λίστα εκείνους τους αριθμούς που λείπουν από την αρχική μορφή
    - Σ of 1's
    - Π of 0's

Μετατροπή μεταξύ Κανονικών Μορφών

- Παράδειγμα
  - $F = xy + x'z$
  - $F(x, y, z) = \Sigma(1, 3, 6, 7)$
  - $F(x, y, z) = \Pi\ (0, 2, 4, 5)$

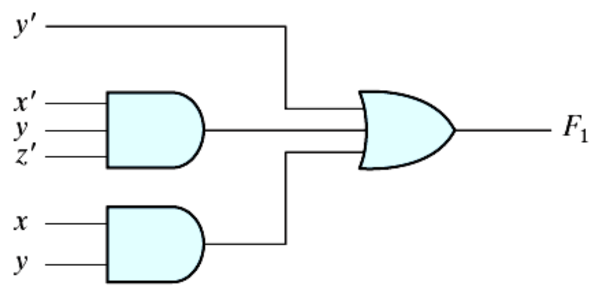
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Πρότυπες Μορφές

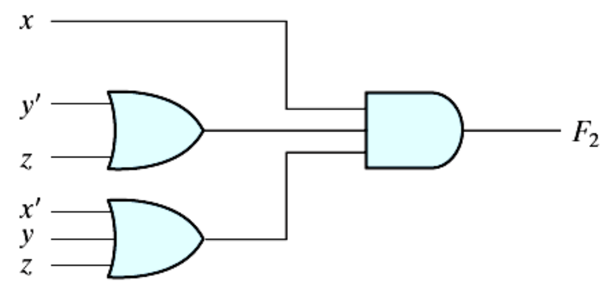
- Οι κανονικές μορφές μιας συνάρτησης σπάνια είναι αυτές με τους λιγότερους όρους και παράγοντες.
- Πρότυπες Μορφές: Οι όροι που σχηματίζουν τη συνάρτηση μπορεί να περιέχουν έναν, δύο ή και οποιονδήποτε αριθμό παραγόντων
  - Άθροισμα γινομένων: $F_1 = y' + xy + x'yz'$
  - Γινόμενο αθροισμάτων: $F_2 = x(y'+z)(x'+y+z')$
  - $F_3 = A'B'CD+ABC'D'$

Υλοποίηση

- Υλοποίηση δύο επιπέδων



(a) Sum of Products

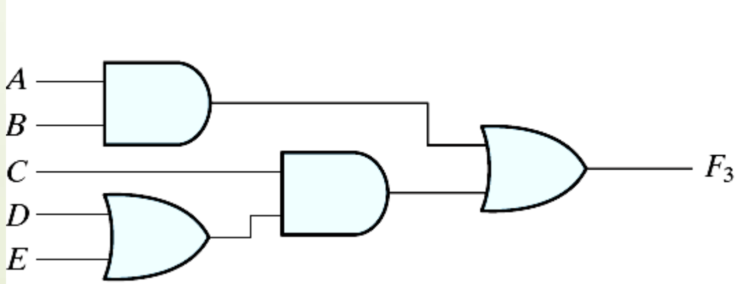


(b) Product of Sums

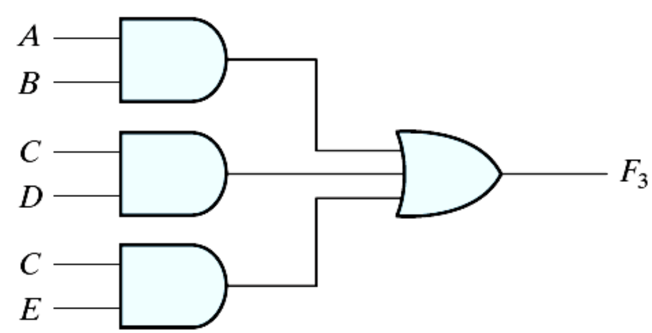
$F_1 = y' + xy + x'yz'$

$F_2 = x(y'+z)(x'+y+z')$

- Υλοποίηση πολλών επιπέδων



(a) $AB + C(D + E)$



(b) $AB + CD + CE$

Άλλες Λογικές Πράξεις

- Υπάρχουν 2^n γραμμές σε έναν πίνακα αληθείας n δυαδικών μεταβλητών.
- 2^{2^n} συναρτήσεις για n δυαδικές μεταβλητές.
  - 16 συναρτήσεις δύο δυαδικών μεταβλητών.

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

- Από τα σύμβολα που προκύπτουν από τις παραπάνω συναρτήσεις μόνο αυτό της **αποκλειστικό-OR** χρησιμοποιείται αρκετά από τους σχεδιαστές ψηφιακών κυκλωμάτων.

Εκφράσεις Bool

Λογικές Συναρτήσεις	Τελεστής	Τελεστής
$F_0 = 0$		**Null**
$F_1 = xy$	$x \cdot y$	AND
$F_2 = xy'$	x/y	Αποτροπή
$F_3 = x$		Μεταφορά
$F_4 = x'y$	y/x	Αποτροπή
$F_5 = y$		Μεταφορά
$F_6 = xy' + x'y$	$x \oplus y$	Αποκλειστικό-OR
$F_7 = x + y$	$x + y$	OR
$F_8 = (x + y)'$	$x \downarrow y$	NOR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Ισοδυναμία
$F_{10} = y'$	y'	Συμπλήρωμα
$F_{11} = x + y'$	$x \subset y$	Συνεπαγωγή
$F_{12} = x'$	x'	Συμπλήρωμα
$F_{13} = x' + y$	$x \supset y$	Συνεπαγωγή
$F_{14} = (xy)'$	$x \uparrow y$	NAND
$F_{15} = 1$		Ταυτότητα

Ψηφιακές Λογικές Πύλες

Εκφράσεις Bool:

Πράξεις AND, OR και NOT

Κατασκευή πυλών άλλων λογικών πράξεων

Πρέπει να εξεταστεί:

Η σκοπιμότητα και το κόστος.

Η δυνατότητα επέκτασης των εισόδων της πύλης.

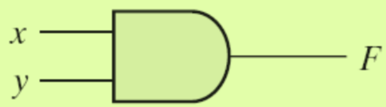
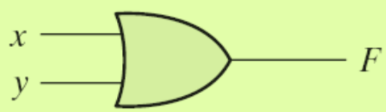
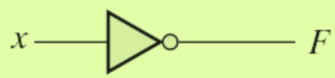
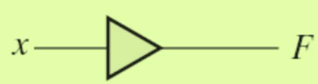
Οι βασικές ιδιότητες των δυαδικών λειτουργιών (αντιμεταθετική και προσεταιριστική ιδιότητα).

Η ικανότητα της πύλης να υλοποιεί Boolean συναρτήσεις.

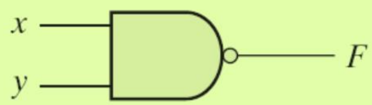
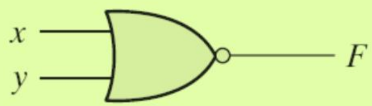
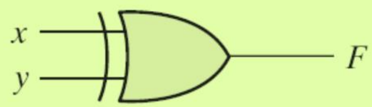
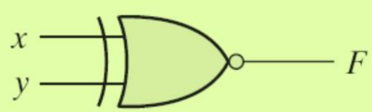
Πρότυπες Πύλες

- Θεωρήστε τις 16 συναρτήσεις των 2 δυαδικών μεταβλητών
  - Δύο είναι ίσες με μια σταθερά (F_0 and F_{15}).
  - Τέσσερις είναι επαναλήψεις άλλων συναρτήσεων (F_4, F_5, F_{10} και F_{11}).
  - Η αποτροπή (F_2) και η συνεπαγωγή (F_{13}) δεν ικανοποιούν την αντιμεταθετική ή την προσεταιριστική ιδιότητα.
  - Οι άλλες οκτώ: το συμπλήρωμα (F_{12}), η μεταφορά (F_3), η AND (F_1), η OR (F_7), η NAND (F_{14}), η NOR (F_8), η XOR (F_6), και η ισοδυναμία (XNOR) (F_9) χρησιμοποιούνται ως πρότυπες πύλες.
    - Το συμπλήρωμα αντιστοιχεί στον αντιστροφέα (inverter).
    - Η Μεταφορά αντιστοιχεί στον απομονωτή (buffer – αυξάνει την ικανότητα οδήγησης)
    - Το Αποκλειστικό-OR στην πύλη XOR
    - Ισοδυναμία: XNOR.

Πρότυπες Πύλες

<table>
<tr><th>Όνομα</th><th>Σύμβολο</th><th>Συνάρτηση</th><th>Πίνακας Αληθείας</th></tr>
<tr><td>AND</td><td>
</td><td>$F = xy$</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OR</td><td>
</td><td>$F = x + y$</td><td><table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>Inverter</td><td>
</td><td>$F = x'$</td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>Buffer</td><td>
</td><td>$F = x$</td><td><table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td></tr>
</table>

Πρότυπες Πύλες

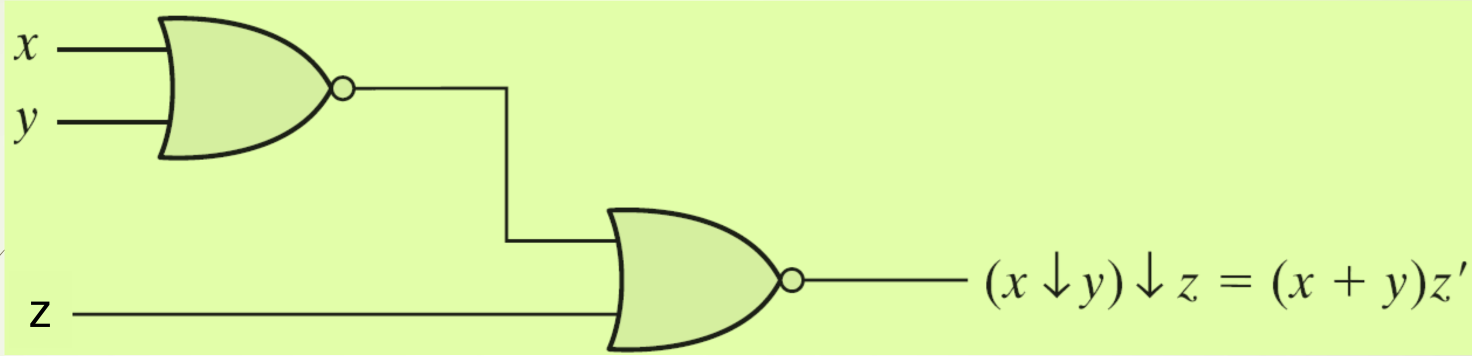
NAND		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

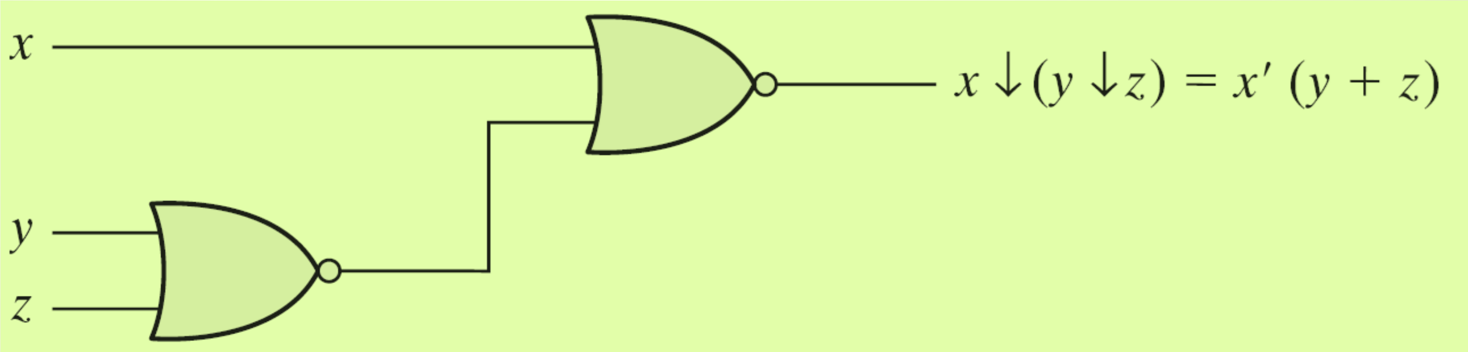
Πολλαπλές Είσοδοι

- Επέκταση σε πολλαπλές εισόδους
  - Μια πύλη μπορεί να επεκταθεί ώστε να περιλαμβάνει πολλαπλές εισόδους αν:
    - Η δυαδική πράξη που υλοποιεί υποστηρίζει την αντιμετάθεση και την προσεταιριστική ιδιότητα.
  - Οι AND και OR τις υποστηρίζουν.
    - OR
      - $x+y = y+x$
      - $(x+y)+z = x+(y+z) = x+y+z$
    - AND
      - $xy = yx$
      - $(x\ y)z = x(y\ z) = x\ y\ z$

Πολλαπλές Είσοδοι

- Οι NAND και NOR υποστηρίζουν την αντιμετάθεση αλλά όχι την προσεταιριστική ιδιότητα → Δεν είναι επεκτάσιμες.

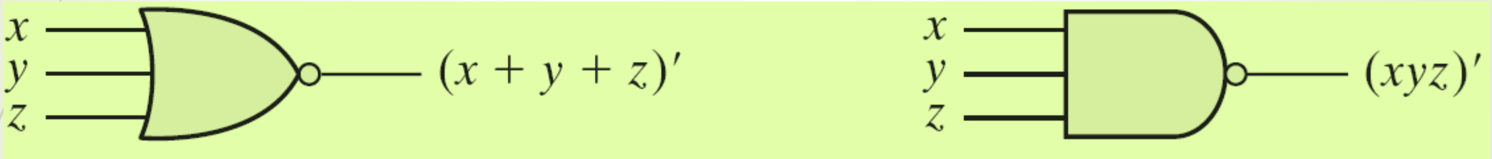




$$(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$$

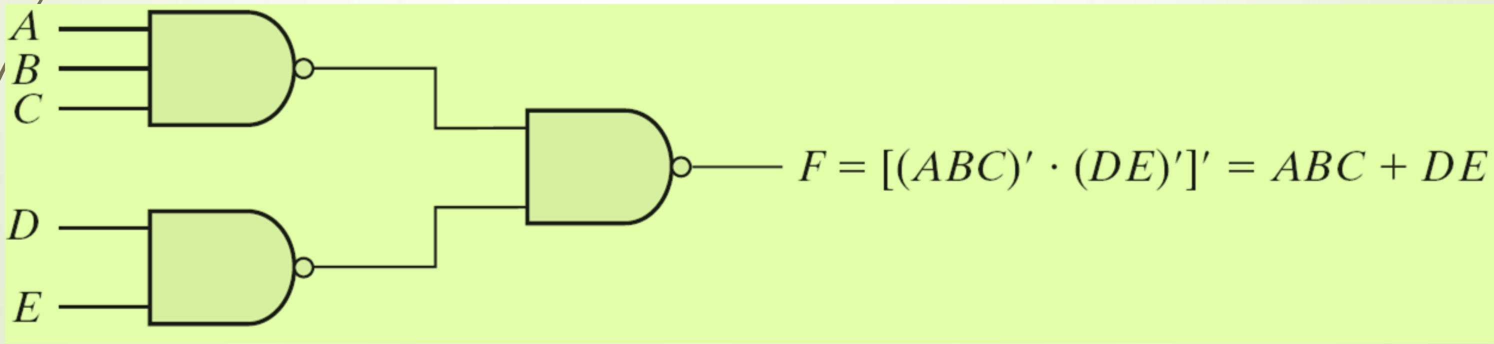
Πολλαπλές Είσοδοι

- NOR πολλαπλών εισόδων = Συμπλήρωμα της πύλης OR,
- NAND πολλαπλών εισόδων = Συμπλήρωμα της πύλης AND.



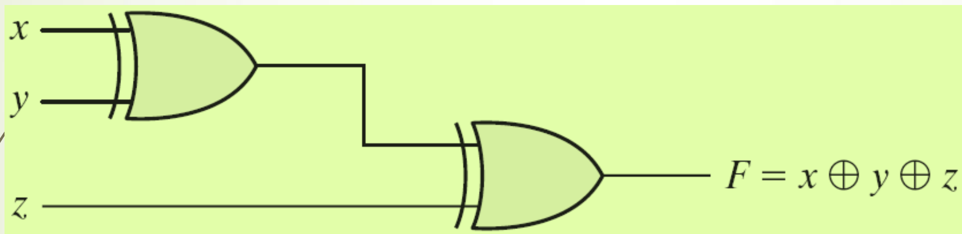
(a) 3-input NOR gate (b) 3-input NAND gate

- Οι διαδοχικές πράξεις NAND= Άθροισμα γινομένων.
- Οι διαδοχικές πράξεις NOR = Γινόμενο αθροισμάτων

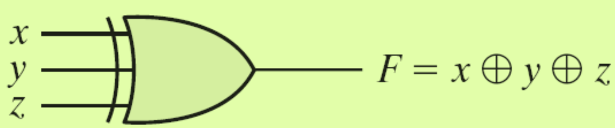


Πολλαπλές Είσοδοι

- Οι πύλες XOR και XNOR υποστηρίζουν την αντιμετάθεση και την προσεταιριστική ιδιότητα gates
- Δύσκολη στην κατασκευή
- Η XOR είναι μια περιττή συνάρτηση: είναι ίση με 1 όταν περιττός αριθμός εισόδων παίρνει την τιμή 1



(a) Χρησιμοποιώντας πύλες 2 εισόδων



(b) XOR πύλη 3 εισόδων

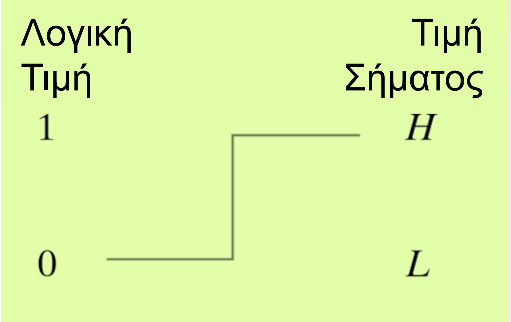
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Πίνακας Αληθείας

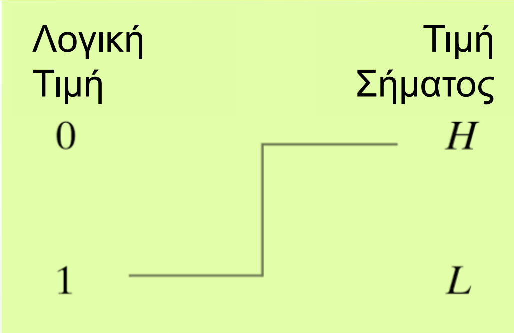
Θετική και Αρνητική Λογική

- Θετική και Αρνητική Λογική
  - Δύο τιμές σημάτων <=> Δύο λογικές τιμές
  - Θετική Λογική: H=1; L=0
  - Αρνητική Λογική: H=0; L=1
- Έστω μια λογική πύλη
  - Μια πύλη AND θετικής λογικής

  => Μια πύλη OR αρνητικής λογικής
  - Υιοθετούμε τη θετική λογική



(a) Θετική Λογική

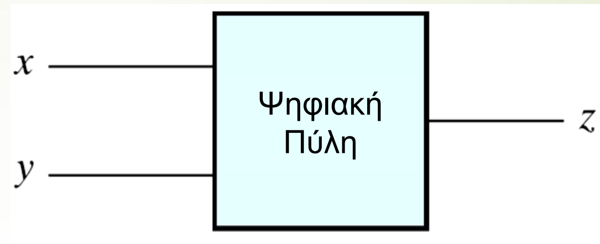


(b) Αρνητική Λογική

Θετική και Αρνητική Λογική

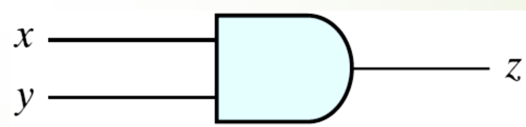
x	y	z
L	L	L
L	H	L
H	L	L
H	H	H

Γενική
Εικόνα



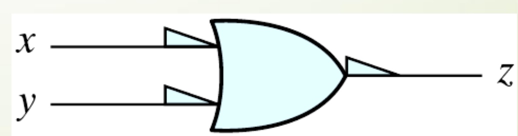
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

Θετική
Λογική



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

Αρνητική
Λογική



Ολοκληρωμένα Κυκλώματα

Επίπεδο Ολοκλήρωσης

- Ολοκληρωμένο Κύκλωμα (Integrated Circuits - IC) ή τσιπ (chip)
- Παραδείγματα:
  - Μικρής Κλίμακας Ολοκλήρωση (Small-scale Integration - SSI): < 10 πύλες
  - Μεσαίας Κλίμακας Ολοκλήρωση (Medium-scale Integration - MSI): 10 ~ 100 πύλες
  - Μεγάλης Κλίμακας Ολοκλήρωση (Large-scale Integration - LSI): 100 ~ xk πύλες
  - Πολύ Μεγάλης Κλίμακας Ολοκλήρωση (Very Large-scale Integration - VLSI): > xk gates
- VLSI
  - Μικρό Μέγεθος
  - Χαμηλό κόστος
  - Χαμηλή κατανάλωση ισχύος
  - Μεγάλη Αξιοπιστία
  - Υψηλής Ταχύτητας

Οικογένειες Ψηφιακής Λογικής

- Οικογένειες Ψηφιακής Λογικής: Τεχνολογία κυκλωμάτων
  - TTL: transistor-transistor logic (Εγκαταλείπεται;)
  - ECL: emitter-coupled logic (Υψηλή ταχύτητα, Μεγάλη κατανάλωση)
  - MOS: metal-oxide semiconductor (NMOS, Μεγάλη πυκνότητα ολοκλήρωσης)
  - CMOS: complementary MOS (Χαμηλή Κατανάλωση)

Οικογένειες Ψηφιακής Λογικής

- Κύρια Χαρακτηριστικά των Οικογενειών Ψηφιακής Λογικής
  - **Οδήγηση** εξόδου (Fan-out): Το πλήθος των τυπικών φορτίων που η έξοδος μιας λογικής πύλης μπορεί να οδηγήσει, χωρίς να αποκλίνει από την κανονική λειτουργία της. Το τυπικό φορτίο ορίζεται συνήθως ως η τιμή της έντασης του ρεύματος που απαιτείται από μία είσοδο μιας άλλης πύλης της ίδιας οικογένειας.
  - **Οδήγηση** εισόδου (Fan-in): Το πλήθος των εισόδων που είναι διαθέσιμες σε μία πύλη.

Οικογένειες Ψηφιακής Λογικής

- Κύρια Χαρακτηριστικά των Οικογενειών Ψηφιακής Λογικής
  - Κατανάλωση ισχύος (Power dissipation): Η ισχύς που καταναλώνεται από την πύλη, η οποία πρέπει να διατεθεί από την τροφοδοσία του συστήματος.
  - Καθυστέρηση διάδοσης (Propagation delay): Ο μέσος χρόνος που χρειάζεται ένα σήμα για να μεταδοθεί από την είσοδο ως την έξοδο μιας πύλης.
    - Παράδειγμα: Εάν η είσοδος ενός αντιστροφέα αλλάξει από 0 σε 1, τότε η έξοδός του θα αλλάξει από 1 σε 0 όμως όχι αμέσως, αλλά μετά από ένα χρονικό διάστημα το οποίο υπαγορεύεται από την καθυστέρηση διάδοσης του αντιστροφέα.
  - Περιθώριο θορύβου (Noise margin): Η μέγιστη τάση εξωτερικού θορύβου που μπορεί να προστεθεί σε ένα σήμα εισόδου χωρίς να προκαλέσει ανεπιθύμητη μεταβολή στην έξοδο του κυκλώματος.

CAD

- CAD – Computer-Aided Design
  - Τσιπ με εκατομμύρια τρανζίστορς
  - Μοντελοποίηση σε υπολογιστή
  - Αυτοματοποίηση της σχεδιαστικής διεργασίας
  - Εισαγωγή σχεδιασμού
    - Σχηματικό διάγραμμα
    - Γλώσσα Περιγραφής Υλικού (HDL – Hardware Description Language)
      - Verilog, VHDL
  - Προσομοίωση
  - Φυσική Υλοποίηση
    - ASIC, FPGA, PLD

Σχεδίαση Chip

- Γιατί επιθυμούμε να έχουμε πολλές πύλες σε ένα απλό chip;
  - Διευκολύνει την υλοποίηση περίπλοκων συστημάτων
  - Χαμηλότερη κατανάλωση ισχύος
  - Υψηλές ταχύτητες (Υψηλή συχνότητα ρολογιού)

- Ποια είναι τα μειονεκτήματα των μεγάλων κυκλωμάτων;
  - Δύσκολο να τα σχεδιάσεις
  - Αυξάνουν οι περιορισμοί στην σχεδίαση και κατασκευή
  - Δύσκολο να ελέγξεις την ορθή λειτουργία τους

- Η ανάπτυξη ολοκληρωμένων κυκλωμάτων απαιτεί τη χρήση εργαλείων
  - Computer Aided Design (CAD) εργαλεία
  - Αυτοματοποίηση περίπλοκων βημάτων στη σχεδίαση
  - Γλώσσα περιγραφής υλικού (HDL)