Τμήμα Πληροφορικής
Πανεπιστήμιο Πειραιώς

Λογική Σχεδίαση Ψηφιακών Συστημάτων

Άλγεβρα Boole και Λογικές Πύλες

Μιχάλης Ψαράκης

Δυαδική λογική

- Μεταβλητές (variables) με δύο μόνο τιμές:
  - 0 ή 1
  - ναι (yes) ή όχι (no)
  - αληθές (true) ή ψευδές (false)
  - ανοικτό (open) ή κλειστό (close)
- Σε αυτό το μάθημα θα μελετήσουμε σύντομα τη δυαδική λογική και την άλγεβρα Boole ώστε:
  - να συνδυάσουμε γρήγορα την άλγεβρα των δυαδικών αριθμών με τα ψηφιακά κυκλώματα που την υλοποιούν στο υλικό (hardware): λογικές πύλες

Δυαδική λογική (συν.)

- Δυαδικές μεταβλητές
  - x, y, z, w, A, B, C, D, ….
  - 0 ή 1
- Λογικές πράξεις
  - Λογικό ΚΑΙ (AND) ή σύζευξη:
    - Συμβολίζεται με τελεία · δηλαδή x · y
    - Δίνει αποτέλεσμα 1 μόνο όταν και το x και το y είναι ίσα με 1. Διαφορετικά το αποτέλεσμα του λογικού ΚΑΙ είναι ίσο με 0.
  - Λογικό Ή (OR) ή διάζευξη:
    - Συμβολίζεται με + δηλαδή x + y
    - Δίνει αποτέλεσμα 1 όταν τουλάχιστον ένα από τα x,y είναι ίσο με 1. Διαφορετικά το αποτέλεσμα του λογικού Ή είναι ίσο με 0.
  - Λογικό ΟΧΙ (NOT) ή αντιστροφή:
    - Συμβολίζεται με τόνο (x’) ή με πάνω παύλα $\overline{x}$
    - Όταν το x είναι ίσο με 1 τότε το x’ είναι ίσο με 0 και αντίστροφα

Αριθμητικές και λογικές πράξεις

- Προσοχή
  - Αν κάνουμε αριθμητική πρόσθεση, τότε η πράξη 1 + 1 δίνει αποτέλεσμα 10_2 δηλαδή 2 στο δυαδικό
  - Αν όμως κάνουμε τη λογική πράξη OR που και πάλι συμβολίζεται με +, τότε η (λογική) πράξη 1 + 1 δίνει αποτέλεσμα 1
- Συχνά το λογικό AND ονομάζεται «πολλαπλασιασμός» και το λογικό OR ονομάζεται «πρόσθεση»

Πίνακες αληθείας (truth tables)

- Πίνακας αληθείας : τρόπος αναπαράστασης και ορισμού των λογικών συναρτήσεων
- Για κάθε συνδυασμό δυαδικών τιμών των μεταβλητών ποια είναι η τιμή (0 ή 1) της λογικής συνάρτησης
- Πίνακες αληθείας των 3 συναρτήσεων AND, OR και NOT

ΚΑΙ (AND)

x	y	x · y
0	0	0
0	1	0
1	0	0
1	1	1

Ή (OR)

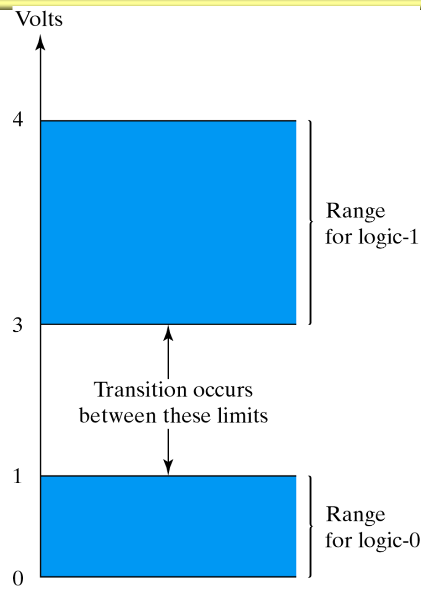
x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

ΌΧΙ (NOT)

x	x’
0	1
1	0

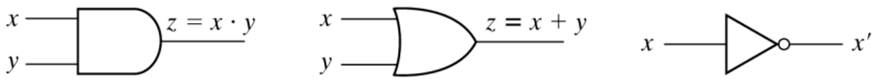
Λογικές πύλες

- Κυκλώματα που κατασκευάζονται από τρανζίστορ (transistors) που λειτουργούν ως διακόπτες
  - Ανοικτός και Κλειστός
- «Ερμηνεύουν» την ηλεκτρική τάση (voltage) στις εισόδους τους σαν λογικό-0 ή λογικό-1 ανάλογα με την τιμή της τάσης
  - Στο σχήμα δίπλα τάσεις μεταξύ 3 και 4 volts ερμηνεύονται σαν λογικό-1, και τάσεις μεταξύ 0 και 1 volts σαν λογικό-0.



Σύμβολα λογικών πυλών

- Οι λογικές πύλες που υλοποιούν στο hardware τις τρεις βασικές λογικές πράξεις (AND, OR, NOT) συμβολίζονται με τα ακόλουθα σύμβολα

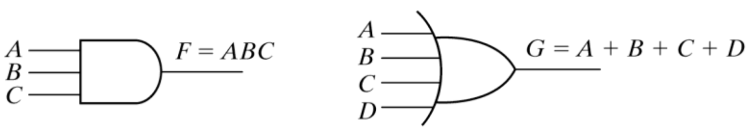


(a) Two-input AND gate　(b) Two-input OR gate　(c) NOT gate or inverter

- Όταν δεχθούν στις εισόδους ηλεκτρικά σήματα με τάση στα επίπεδα του λογικού 0 και του λογικού 1, δίνουν στην έξοδο την αντίστοιχη λογική τιμή ανάλογα με τη συνάρτηση.
  - Όπως θα δούμε σε επόμενα κεφάλαια
    - Υπάρχουν και άλλοι τύποι λογικών πυλών που υλοποιούν άλλες λογικές συναρτήσεις
    - Γενικά οι λογικές πύλες μπορούν να έχουν $k \geq 2$ εισόδους και συνεπώς να υλοποιούν λογικές συναρτήσεις με $k \geq 2$ μεταβλητές.

Πύλες πολλαπλών εισόδων

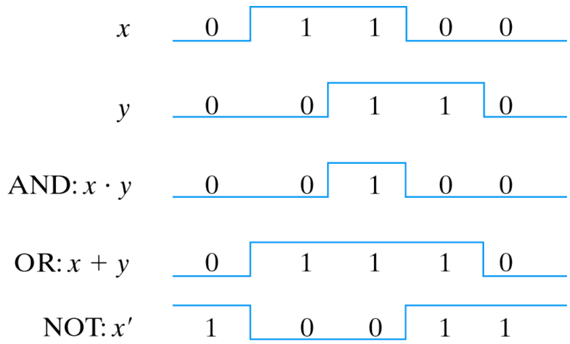
- Πύλες AND και OR τριών και τεσσάρων εισόδων (μεταβλητών) αντίστοιχα.



(a) Three-input AND gate　(b) Four-input OR gate

Σήματα εισόδου/εξόδου

- Κυματομορφές εισόδου/εξόδου
  - Άλλος τρόπος αναπαράστασης/περιγραφής λογικών κυκλωμάτων
- Μια «επάνω» γραμμή δείχνει λογικό-1
- Μια «κάτω» γραμμή δείχνει λογικό-0



Άλγεβρα Boole

- George Boole (1815-1864)
  - Άγγλος μαθηματικός και φιλόσοφος
  - Το 1854 ανέπτυξε ένα αλγεβρικό σύστημα δύο τιμών που ονομάζουμε άλγεβρα Boole
- Claude E. Shannon (1916-2001)
  - Αμερικανός μαθηματικός
  - Θεμελίωσε τη θεωρία της πληροφορίας (Information theory)
  - Το 1938 προσάρμοσε την άλγεβρα Boole ώστε να περιγράψει πλήρως τις ιδιότητες και τη λειτουργία των δισταθών ηλεκτρικών κυκλωμάτων διακοπτών (switching algebra)
    - Στους ηλεκτρικούς (ή ηλεκτρονικούς) διακόπτες (που σήμερα κατασκευάζονται από τρανζίστορ) βασίζεται (και πάντα βασίζονταν) η λειτουργία των ψηφιακών υπολογιστών !
    - "I always loved this word: Boolean" – Claude E. Shannon, σε άρθρο του στο περιοδικό Spectrum, 1992.

Άλγεβρα Boole

- Αλγεβρικό σύστημα δυο τιμών : 0, 1
- Δυαδικοί τελεστές OR “**+**” και AND “**•**”
- Μοναδιαίος τελεστής συμπληρώματος NOT “ ‘ “
- Ορισμός τελεστέων

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

Άλγεβρα Boole: αξιώματα

- Βασίζεται σε έναν αριθμό αξιωμάτων (αξιώματα Huntington):
- Κλειστότητα (closure):
  - Κλειστότητα ως προς τον τελεστή +
  - Κλειστότητα ως προς τον τελεστή •
- Κανόνας ουδέτερου στοιχείου (identity law):
  - Υπάρχει ουδέτερο στοιχείο ως προς τον + που συμβολίζεται με 0: $x + 0 = 0 + x = x$
  - Υπάρχει ουδέτερο στοιχείο ως προς τον • που συμβολίζεται με 1: $x \bullet 1 = 1 \bullet x = x$
- Αντιμεταθετικός κανόνας (commutative law):
  - Η πράξη + είναι αντιμεταθετική $x + y = y + x$
  - Η πράξη • είναι αντιμεταθετική $x \bullet y = y \bullet x$

Άλγεβρα Boole: αξιώματα (συν.)

- Προσεταιριστικός κανόνας (associative law):
  - Η πράξη + είναι προσεταιριστική: (x+y)+z = x+(y+z)
  - Η πράξη • είναι προσεταιριστική: (x•y)•z = x•(y•z)
- Επιμεριστικός κανόνας (distributive law):
  - Η πράξη • είναι επιμεριστική ως προς την + : x•(y+z) = x•y + x•z
  - Η πράξη + είναι επιμεριστική ως προς την • : x+(y•z) = (x+y) • (x+z)
- Κανόνας συμπληρώματος (complement law):
  - Για κάθε στοιχείο x, υπάρχει x' που ονομάζεται συμπλήρωμα και ισχύει x + x' = 1 και x • x' = 0

Άλγεβρα Boole: θεωρήματα

- Αρχή του δυϊσμού (duality principle)
  - Αν σε μία αλγεβρική παράσταση της άλγεβρας Boole, αλλάξουμε όλα τα 0 με 1 και αντίστροφα και όλες τις πράξεις + τις αντικαταστήσουμε με • και αντίστροφα, τότε λαμβάνουμε μια ισοδύναμη παράσταση.
- Θεώρημα 1:
  x + x = x και x • x = x
- Θεώρημα 2 :
  x + 1 = 1 και x • 0 = 0
- Θεώρημα 3 :
  (x')' = x
- Θεώρημα 4:
  x + xy = x και x(x + y) = x

Αποδείξεις θεωρημάτων

- Θεώρημα 1: $x + x = x$ και $x \bullet x = x$
  - $x + x = (x + x) \bullet 1 = (x + x) \bullet (x + x') = x + (x \bullet x') = x + 0 = x$
  - $x \bullet x = (x \bullet x) + 0 = (x \bullet x) + (x \bullet x') = x \bullet (x + x') = x \bullet 1 = x$
- Θεώρημα 2 : $x + 1 = 1$ και $x \bullet 0 = 0$
  - $x + 1 = (x + 1) \bullet 1 = (x + 1) \bullet (x + x') = x + (1 \bullet x') = x + x' = 1$
  - $x \bullet 0 = (x \bullet x) + 0 = (x \bullet x) + (x \bullet x') = x \bullet (x + x') = x \bullet 1 = x$
- Θεώρημα 4: $x + xy = x$ και $x(x + y) = x$
  - $x + x \bullet y = x \bullet 1 + x \bullet y = x \bullet (1 + y) = x \bullet 1 = x$
  - $x \bullet (x + y) = (x + 0) \bullet (x + y) = x + 0 \bullet y = x + 0 = x$

Θεώρημα DeMorgan

- Θεώρημα DeMorgan 2 μεταβλητών

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

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

- Γενικευμένο θεώρημα DeMorgan

  $(x_1 + x_2 + \ldots +x_n)' = x_1' x_2' \ldots x_n'$

  $(x_1 x_2 \ldots x_n)' = x_1' + x_2' + \ldots x_n'$

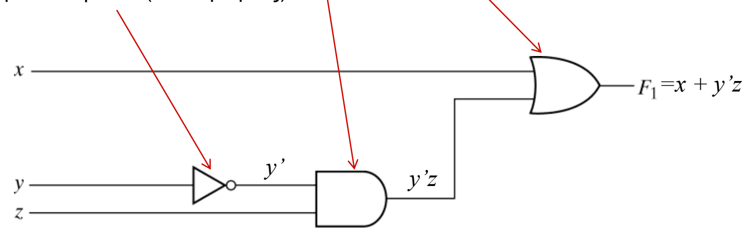
  - Έχει πολλή σημαντική εφαρμογή στη βελτιστοποίηση των λογικών συναρτήσεων

Προτεραιότητα τελεστών

- Για τον υπολογισμό της τιμής μιας αλγεβρικής παράστασης Boole ακολουθούμε την εξής σειρά προτεραιότητας:
  - Παρενθέσεις
  - Συμπλήρωμα (πράξη NOT)
  - Πράξη AND
  - Πράξη OR
- Παράδειγμα: x+yz’
  - Σειρά των πράξεων:
    - Συμπλήρωμα του z (δηλαδή z’)
    - Λογικό ΚΑΙ του y και του z’ (δηλαδή yz’)
    - Λογικό Ή του x και του yz’ (δηλαδή x+yz’)
  - Ποια είναι η τιμή της παράστασης αν x=0, y=1 και z=0;
    - z=0 → z’ = 1
    - y=1 και z’=1 → yz’ = 1
    - x=0 και yz’=1 → x+yz’=1

Λογικές συναρτήσεις (συναρτήσεις Boole)

- Περιγράφονται από αλγεβρικές εκφράσεις που περιέχουν:
  - Δυαδικές μεταβλητές (που παίρνουν τιμές 0 ή 1)
  - Λογικές πράξεις AND (·), OR (+), NOT (’)
- Υπολογισμός της τιμής μιας λογικής συνάρτησης:
  - Παράδειγμα $F_1 = x + y'z$
  - Η F1 είναι ίση με 1 εάν το x είναι ίσο με 1 ή εάν το y’ είναι ίσο με 1 και το z είναι ίσο με 1
- Από την αλγεβρική έκφραση προκύπτει μονοσήμαντα μια υλοποίηση της συνάρτησης με λογικές πύλες (που είδαμε ή θα δούμε στη συνέχεια)
  - Υλοποίηση της F1 με μία πύλη AND, μία πύλη OR και μία πύλη NOT (αντιστροφέας)



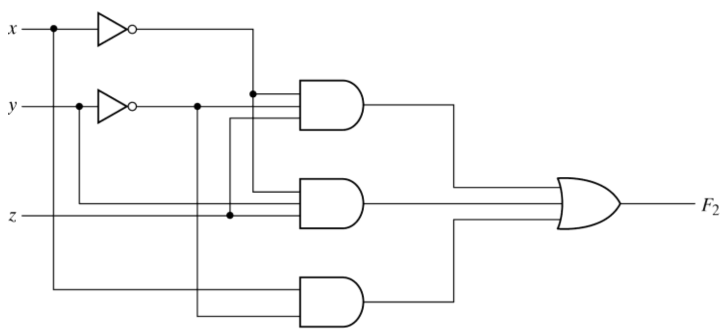
Πίνακες αληθείας (truth tables)

- Ο πίνακας αληθείας μπορεί να εξαχθεί από την αλγεβρική έκφραση μιας συνάρτησης
  - Και αντίστροφα, από τον πίνακα αληθείας μπορεί να εξαχθεί η αλγεβρική παράσταση
- Πίνακες αληθείας για τις συναρτήσεις F1 και F2

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

Η συνάρτηση F_2

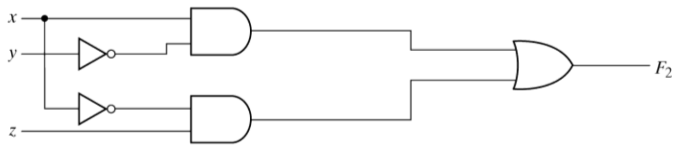
- Από τον πίνακα αληθείας της F_2 παράγεται η αλγεβρική έκφραση:
  - $F_2 = x' y' z + x' y z + x y'$
- Η υλοποίησή της με λογικές πύλες είναι η ακόλουθη:



(a) $F_2 = x'y'z + x'yz + xy'$

Αλγεβρική απλοποίηση της F_2

- Η F_2 μπορεί να απλοποιηθεί
  - $F_2 = x' y' z + x' y z + x y' = x' z (y' + y) + x y' = x' z + x y'$
  - Και η νέα υλοποίηση με λογικές πύλες είναι η ακόλουθη:



(b) $F_2 = xy' + x'z$

  - Η απλοποιημένη αλγεβρική έκφραση οδήγησε σε υλοποίηση με λιγότερες και απλούστερες πύλες (λιγότερων εισόδων)
    - Από 4 πύλες και 2 αντιστροφείς πήγαμε στις 3 πύλες και 2 αντιστροφείς
    - Από 13 εισόδους πυλών πήγαμε στις 8 εισόδους πυλών
  - Η απλοποίηση λογικών κυκλωμάτων οδηγεί σε κυκλώματα:
    - Μικρότερα, ταχύτερα, φθηνότερα και με χαμηλότερη κατανάλωση ενέργειας

Αλγεβρικοί μετασχηματισμοί

- Με αλγεβρικούς μετασχηματισμούς μπορούμε να απλοποιήσουμε μια συνάρτηση
  - Αλλά όταν οι συναρτήσεις έχουν πολλές μεταβλητές μπορεί να οδηγηθούμε σε λάθη
  - Πρέπει να χρησιμοποιήσουμε συστηματικούς τρόπους ελαχιστοποίησης λογικών συναρτήσεων
    - Που να μπορούν να προγραμματιστούν σε εργαλεία λογισμικού που να τις εκτελούν αυτόματα
  - Ένας συστηματικός τρόπος είναι η μέθοδος Χάρτη Karnaugh
    - Θα τη μελετήσουμε αργότερα

Συμπλήρωμα συνάρτησης

- Μια λογική συνάρτηση F έχει μια συμπληρωματική συνάρτηση F’ ή συμπλήρωμα της F η οποία δίνει τιμή 0 εκεί που η F δίνει 1 και αντίστροφα
- Αλγεβρικά το συμπλήρωμα παράγεται από την F με χρήση του θεωρήματος του DeMorgan
- Γενίκευση θεωρήματος DeMorgan
  - (A + B + C + D + … + F)’ = A’ B’ C’ D’ … F’
  - (A B C D … F)’ = A’ + B’ + C’ + D’ + … + F’
- Παράδειγμα: Βρείτε το συμπλήρωμα της F_1 = x’yz’+x’y’z
  - F_1’ = (x’yz’+x’y’z)’ = (x’yz’)’(x’y’z)’ = (x+y’+z)(x+y+z’)

Ελαχιστόροι

- Για μία συνάρτηση με *n* μεταβλητές κάθε γινόμενο (πράξη AND) που περιέχει όλες τις *n* μεταβλητές είτε στην κανονική είτε στην συμπληρωμένη μορφή τους, ονομάζεται πρότυπο γινόμενο ή ελαχιστόρος (minterm)
  - Συνολικά υπάρχουν 2^n ελαχιστόροι
- Παραδείγματα:
  - Για n=2 μεταβλητές x και y, οι ελαχιστόροι είναι x’y’, x’y, xy’ και xy
  - Για n=3 μεταβλητές x, y και z, οι ελαχιστόροι είναι x’y’z’, x’y’z, x’yz’, x’yz, xy’z’, xy’z, xyz’, και xyz
  - Οι ελαχιστόροι συμβολίζονται με $\mathbf{m_j}$ όπου το j προκύπτει από τις μεταβλητές βάζοντας 0 όταν έχει τόνο η μεταβλητή και 1 όταν δεν έχει

Ελαχιστόροι 3 μεταβλητών

x	y	z	Ελαχιστόροι Όρος	Ελαχιστόροι 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

Κανονικές μορφές

- Δίνονται οι συναρτήσεις f_1 και f_2

x	y	z	f_1	f_2
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

- Κάθε συνάρτηση εκφράζεται σαν άθροισμα ελαχιστόρων
- Κανονικές Μορφές (Canonical Forms)
  - f_1 = x’y’z+xy’z’+xyz = $m_1+m_4+m_7$
  - f_2 = x’yz+xy’z+xyz’+xyz = $m_3+m_5+m_6+m_7$

Άθροισμα ελαχιστόρων

- Να εκφραστεί η συνάρτηση F = A + B'C σαν άθροισμα ελαχιστόρων

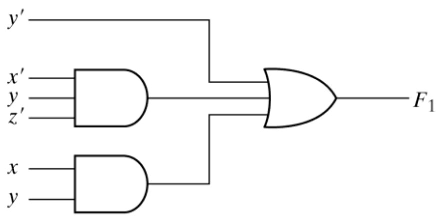
$$\begin{aligned} A + B'C &= AB + AB' + B'C \\ &= ABC' + ABC + AB'C' + AB'C + B'C \\ &= ABC' + ABC + AB'C' + AB'C + A'B'C + AB'C = \\ &= m_6+m_7+m_4+m_5+m_1 \\ &= m_1+m_4+m_5+m_6+m_7 \end{aligned}$$

- Συχνά (για συντομία), εκφράζουμε μια συνάρτηση σαν άθροισμα ελαχιστόρων ως:

  F(A,B,C) = Σ(1,4,5,6,7)

Πρότυπες μορφές (standard forms)

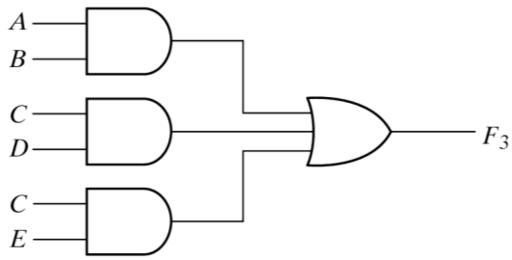
- Άθροισμα γινομένων – sum of products (όχι κατ' ανάγκη ελαχιστόρων) ή



- Ονομάζονται και διεπίπεδες υλοποιήσεις (two-level implementations)

Υλοποίηση με διαφορετικά επίπεδα

- Υλοποίηση συνάρτησης: *AB + CD + CE*

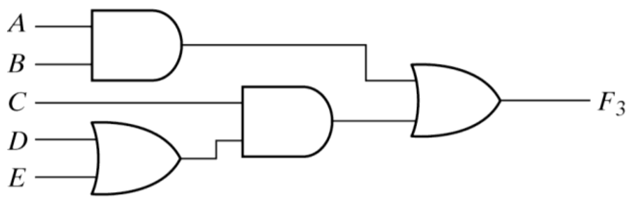


Με δύο επίπεδα:

AB + CD + CE

Με τρία επίπεδα:

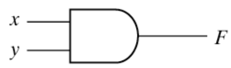
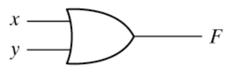
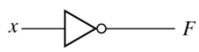
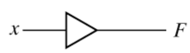
AB + C(D + E)



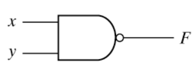
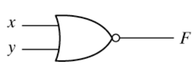
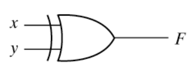
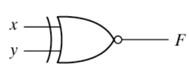
Άλλες λογικές πράξεις

- Πόσες λογικές συναρτήσεις των 2 μεταβλητών υπάρχουν;
  - Γενικά, για n μεταβλητές υπάρχουν 2^{2^n} λογικές συναρτήσεις
- Κάποιες λογικές συναρτήσεις αντιστοιχούν σε καινούργιες λογικές πύλες:
  - xy’+x’y = x ⊕ y -- Αποκλειστικό Ή (XOR)
  - (x+y)’ = x ↓y -- ΟΥΤΕ = ΌΧΙ ‘Η (NOR)
  - (xy)’ = x↑y -- ΟΧΙ ΚΑΙ (NAND)

Ψηφιακές λογικές πύλες (1)

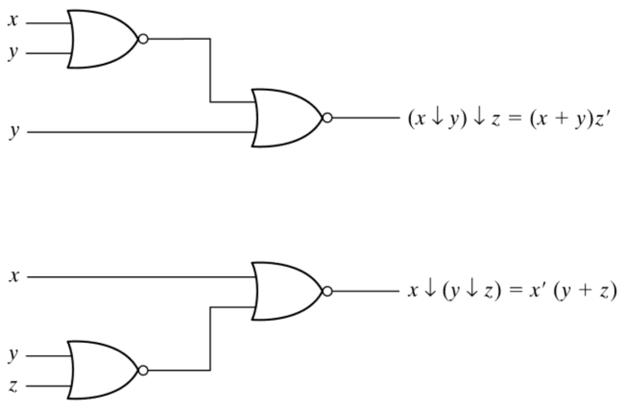
Name	Graphic symbol	Algebraic function	Truth table
AND		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

Ψηφιακές λογικές πύλες (2)

Name	Graphic symbol	Algebraic function	Truth table
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

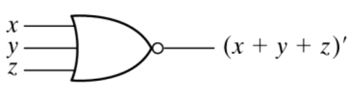
Πύλες πολλαπλών εισόδων

- Η πράξη NOR δεν είναι προσεταιριστική

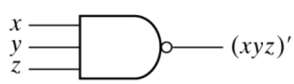


Πύλες ΟΥΤΕ (NOR) και ΟΧΙ-ΚΑΙ (NAND)

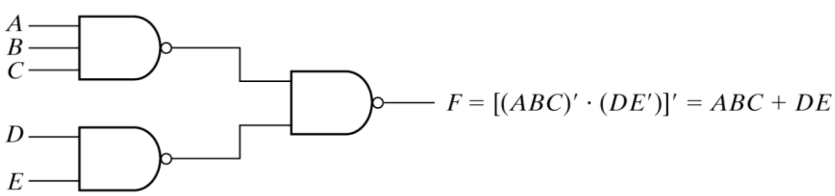
- Δύο και τριών εισόδων



(a) 3-input NOR gate



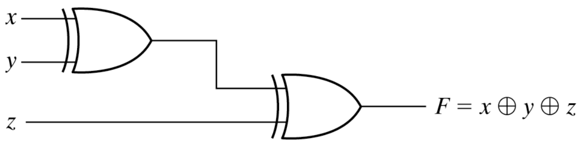
(b) 3-input NAND gate



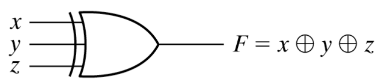
(c) Cascaded NAND gates

Αποκλειστικό Ή (Exclusive OR – XOR)

- Δύο και τριών εισόδων
- Γενικά για n εισόδους: δίνει έξοδο 1 αν περιττό πλήθος εισόδων είναι ίσες με 1, αλλιώς δίνει 0.



(a) Using 2-input gates



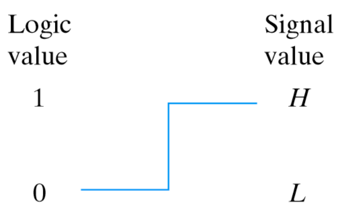
(b) 3-input gate

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

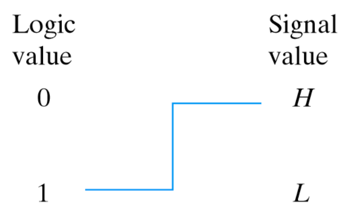
(c) Truth table

Θετική και αρνητική λογική

- Είναι ισοδύναμες
  - Είναι θέμα μετάφρασης (αντιστοίχισης) του επιπέδου τάσης (υψηλό και χαμηλό) σε λογικά σήματα 0 και 1.



(a) Positive logic

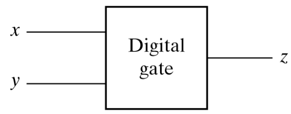


(b) Negative logic

Εξήγηση θετικής και αρνητικής λογικής

x	y	F
L	L	L
L	H	L
H	L	L
H	H	H

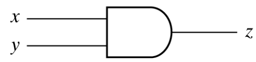
(a) Truth table with H and L



(b) Gate block diagram

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

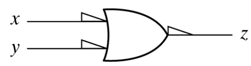
(c) Truth table for positive logic



(d) Positive logic AND gate

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

(e) Truth table for negative logic



(f) Negative logic OR gate