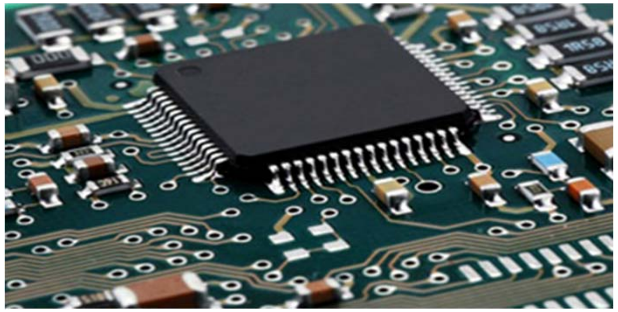
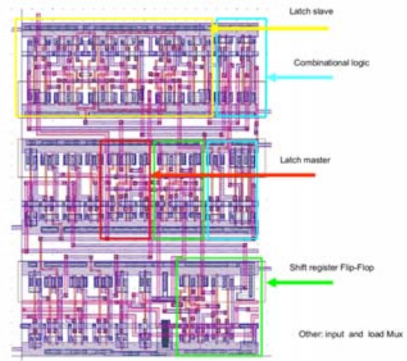




Ψηφιακή Σχεδίαση

Διάλεξη 2 – Αναπαράσταση Αριθμών, Δυαδικοί Αριθμοί, Άλγεβρα Boole

Γεώργιος Κεραμίδας, Επίκουρος Καθηγητής

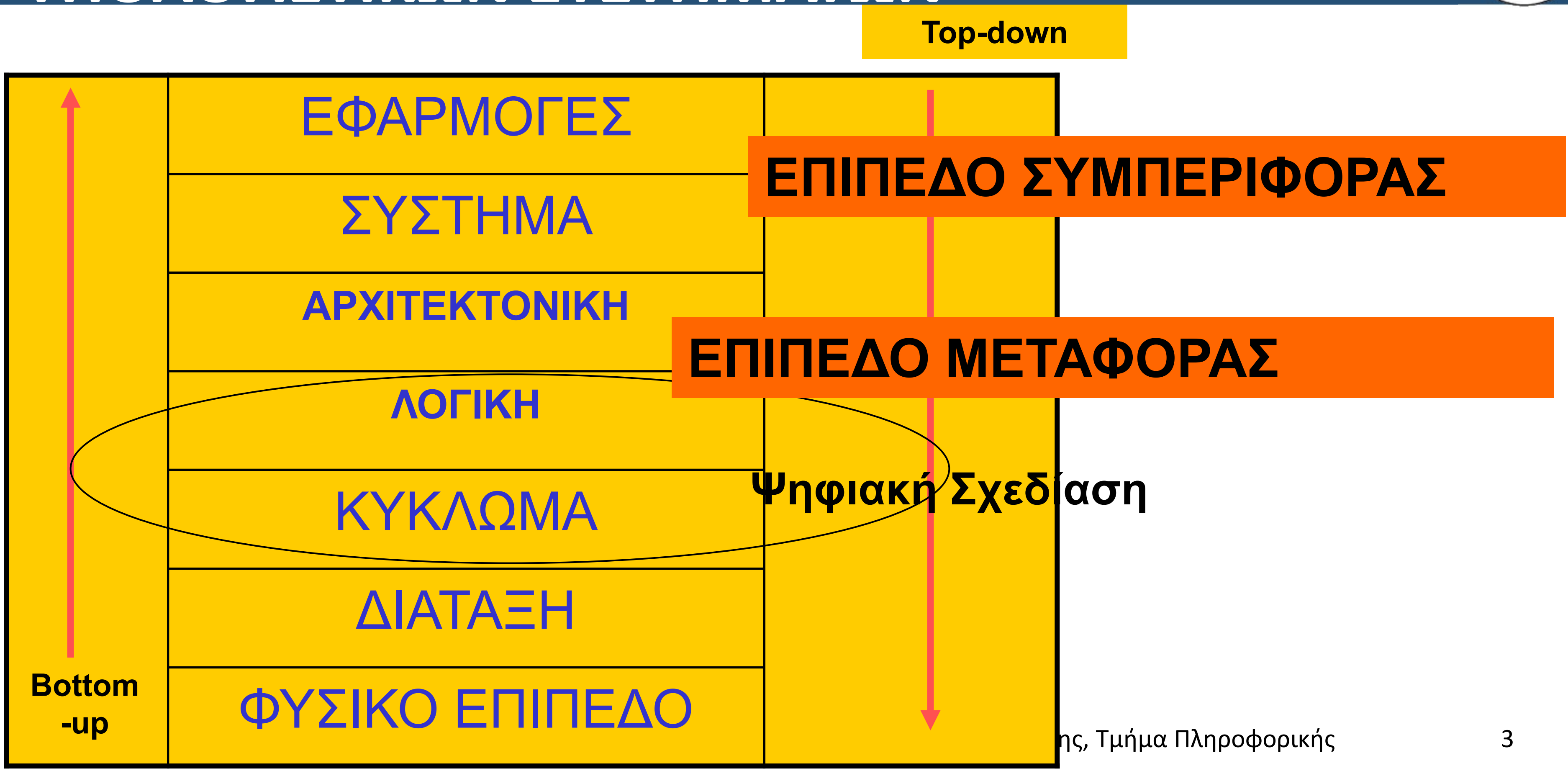
2ο Εξάμηνο, Τμήμα Πληροφορικής

Αντιστοίχιση με ύλη Βιβλίου

- **Το συγκεκριμένο σετ διαφανειών καλύπτει τα εξής κεφάλαια/ενότητες:**
  - **Κεφάλαιο 1: 1.3, 1.4, 1.5**
  - **Κεφάλαιο 2: 2.1, 2.2, 2.3**

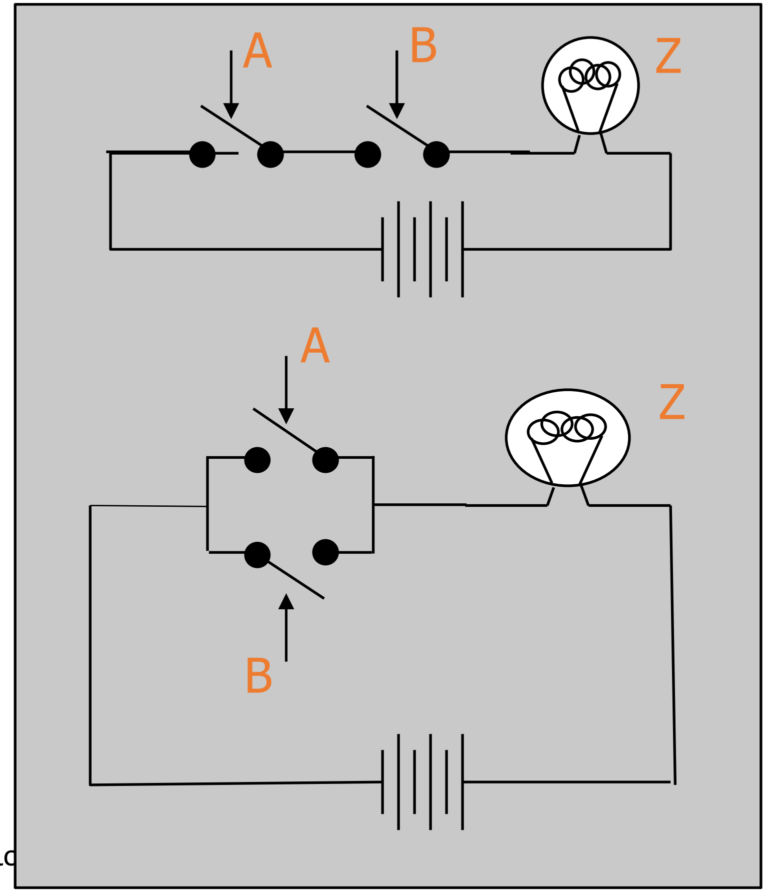
- **Βιβλίο [68406394]: Ψηφιακή Σχεδίαση, 5η Έκδοση, Mano Morris, Ciletti Michael**

ΕΠΙΠΕΔΑ ΜΕΛΕΤΗΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

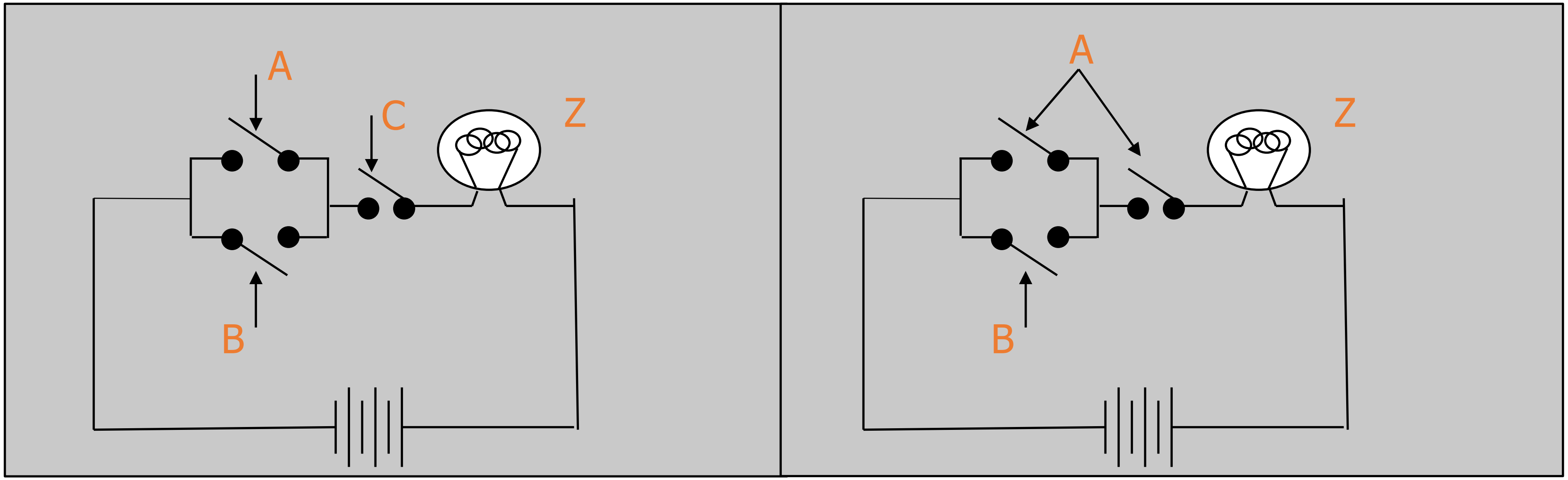


Απλές λογικές συναρτήσεις 2 δυαδικών μεταβλητών

- **Οι συναρτήσεις αυτές ονομάζονται και λογικές γιατί τις χρησιμοποιούμε στη λογική των προτάσεων που συντάσσουμε καθημερινά.**



Σύνθεση συναρτήσεων



- **Όσο πιο πολύπλοκο γίνεται το σχέδιο, τόσο πιο προφανές γίνεται ότι χρειαζόμαστε κάποιο τυπικό τρόπο, για να βρούμε πότε το Z θα γίνει 1.**
- **Χρειαζόμαστε συνεπώς μια άλγεβρα !**

Τι είναι μια άλγεβρα ?

- Μια μαθηματική δομή :
  - Σύνολο ψηφίων & αναπαραστάσεων
  - Σύνολο τελεστών & προτεραιότητες
  - Αξιώματα
  - Θεωρήματα
  - Συναρτήσεις
- Αλγεβρα των φυσικών αριθμών
  - Ψηφίο {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, Αναπαραστάσεις = άπειρο
  - Τελεστές : +, -, *, /   Προτεραιότητες : {,[,(, /, *, -, +
  - Αξιώματα : ουδετερότητα του 0 στη +, του 1 στον *, προσεταιριστικότητα του +, επιμεριστικότητα του * ως προς το +, ...
  - Θεωρήματα : x+x = 2x
  - Συναρτήσεις f(x,y) = 3x+5y

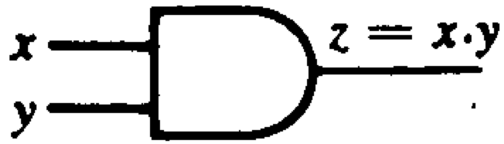
Αλγεβρα Boole

- **Το 1854 (!!!) ο Αγγλος μαθηματικός George Boole εισήγαγε μια άλγεβρα δύο τιμών : αλήθεια και ψέμα (true, false) .**
- **Προτάθηκε για το λογισμό της αλήθειας ή του ψέμματος προτάσεων**
- **Π.χ.**
  - **Δεχόμαστε τη πρόταση "Οσοι κάνουν μπάνιο κάθε μέρα έχουν πολλά λεφτά ή δε τους αρέσει η βρωμιά" ως αληθή.**
  - **Δεχόμαστε και τη πρόταση "Κανένας μηχανικός υπολογιστών δεν έχει πολλά λεφτά" ως αληθή.**
  - **Εστω η πρόταση "Αν είσαι μηχανικός υπολογιστών και σου αρέσει η βρωμιά, κάνεις μπάνιο κάθε μέρα".**
  - **Τι μπορούμε να αποφανθούμε για την αλήθεια ή το ψέμα της ?**

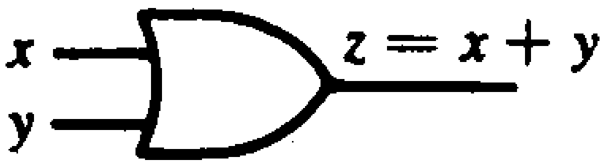
Shannon

- Σχεδόν ένα αιώνα αργότερα (1938), ο Claude Shannon διαπιστώνει ότι απλά αντιστοιχίζοντας τις "αλήθεια" και "ψέμμα" στο "κλειστός διακόπτης" και "ανοικτός διακόπτης", μπορεί να εφαρμόσει τα όσα ανέπτυξε ο Boole και σε κυκλώματα διακοπτών
- Ετσι η άλγεβρα Boole έγινε "switching algebra".
- Εμείς θα πάμε απλά ένα βήμα πιο πέρα. Αντί για "κλειστός διακόπτης" και "ανοικτός διακόπτης" θα χρησιμοποιούμε τις τιμές δυαδικών μεταβλητών "1" και "0" αντίστοιχα.
- Ετσι έχουμε τη δυαδική άλγεβρα (λογική άλγεβρα)

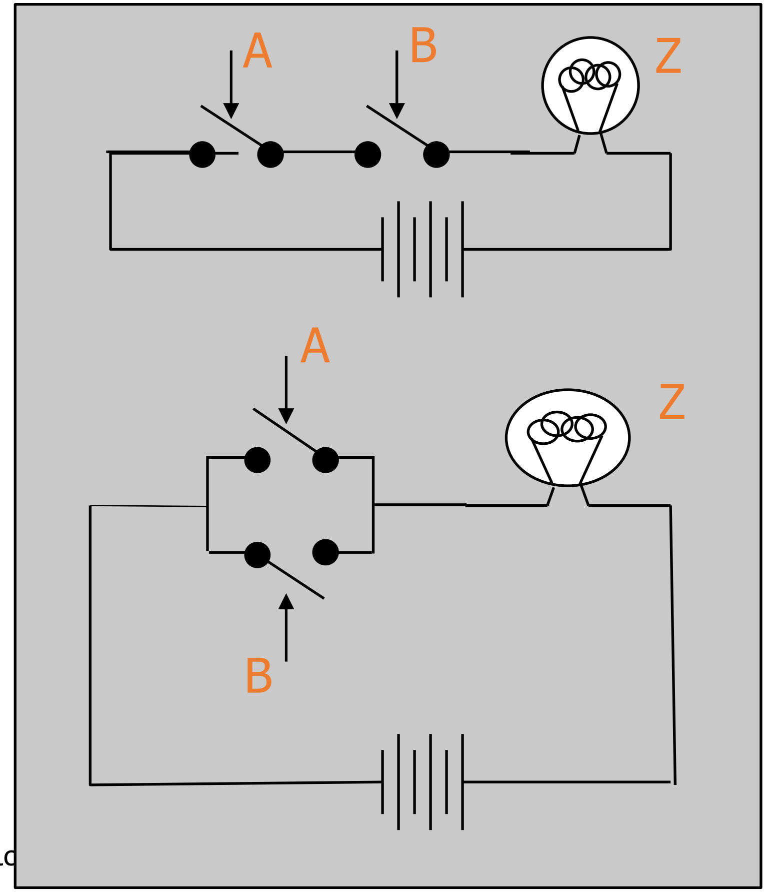
Απλές λογικές συναρτήσεις 2 δυαδικών μεταβλητών



AND



OR



Αξιώματα

- Δίτιμη άλγεβρα. Κάθε στοιχείο της X $\in$ {0,1}
- Υπαρξη συμπληρώματος (α')
  - Αν α = 0 => α' = 1, Αν α = 1 => α' = 0
- Δύο λογικές πράξεις :
  - Λογική σύζευξη (and / και) : • (το σύμβολο μπορεί να παραλείπεται σε απλή παράταξη μεταβλητών)
  - Λογική διάζευξη (or / ή) : +
- Οι λογικές πράξεις ακολουθούν τους εξής κανόνες :
  - 0 • 0 = 0, 1 + 1 = 1
  - 1 • 1 = 1, 0 + 0 = 0
  - 0 • 1 = 1 • 0 = 0, 1 + 0 = 0 + 1 = 1
- Σε μία σύνθετη συνάρτηση των δύο λογικών πράξεων, οι προτεραιότητες είναι : {, [, (, ', •, +

Θεωρήματα μιας μεταβλητής

(Απόδειξη με εξέταση όλων των δυνατών τιμών της μεταβλητής)

- **$X + 0 = X$, $X \bullet 1 = X$ (Ουδέτερα στοιχεία)**
- **$X + 1 = 1$, $X \bullet 0 = 0$ (Απορροφητικά στοιχεία)**
- **$X + X = X$, $X \bullet X = X$ (Αυτοαπορρόφησης)**
- **$(X')' = X$ (Διπλοαντιστροφής)**
- **$X + X' = 1$ $X \bullet X' = 0$ (Συμπληρωματικά στοιχεία)**

Θεωρήματα δύο και τριών μεταβλητών (1/2)

(Απόδειξη με εξέταση όλων των δυνατών τιμών των δύο μερών)

- **X + Y = Y + X, X • Y = Y • X (Αντιμεταθετική)**
- **(X + Y) + Z = X + (Y + Z), (X • Y) • Z = X • (Y • Z) (Προσεταιριστική)**
  - **Συμπέρασμα : Σε λογικά γινόμενα ή λογικά αθροίσματα, η χρήση παρενθέσεων είναι προαιρετική. Για παράδειγμα μπορώ να γράφω w + x + y + z αφού όπως κι αν υπολογιστεί αυτή η έκφραση θα πάρω το ίδιο λογικό αποτέλεσμα.**
- **X • Y + X • Z = X • (Y + Z), (X + Y) • (X + Z) = X + (Y • Z) (Επιμεριστική)**
  - **Ισχύουν και οι 2 επιμερισμοί. Στην άλγεβρα των αριθμών ισχύει μόνο του x ως προς το +.**

Θεωρήματα δύο και τριών μεταβλητών (1/2)

(Απόδειξη με εξέταση όλων των δυνατών τιμών των δύο μερών)

- **Παράδειγμα :**
- $(W + Y) \bullet (X + Z) \bullet (V + Y) \bullet (W + Z) \bullet (X + Y) \bullet (V + Z) =$

$$= [W + (Y \bullet Z)] \bullet [X + (Y \bullet Z)] \bullet [V + (Y \bullet Z)] =$$

$$= (Y \bullet Z) + (V \bullet X \bullet W)$$

Θεωρήματα δύο και τριών μεταβλητών (2/2)

(Απόδειξη με εξέταση όλων των δυνατών τιμών των δύο μερών)

- **X + X • Y = X, X • (X + Y) = X (Κάλυψης)**
- **X • Y + X • Y'= X (X + Y) • (X + Y') = X (Συνδυασμών)**
  - **Παράδειγμα : X • Y • Z' + X • Y' • Z' + X • Y • Z + X • Y' • Z = X • Z' + X • Z = X**
  - **Συμπέρασμα : Σε ένα άθροισμα γινομένων, που κάθε γινόμενο έχει κ μεταβλητές, αν υπάρχουν όλοι οι συνδυασμοί τιμών των κ-1 μεταβλητών τότε μπορούν να διαγραφούν από την έκφραση.**

Θεωρήματα γενικευμένου αριθμού μεταβλητών

- $X + X + \dots + X = X$, $X \bullet X \bullet \dots \bullet X = X$ (Αυτοαπορρόφησης)

- $(X_1 + X_2 + \dots + Xn)' = X_1' \bullet X_2' \bullet \dots \bullet Xn'$
- $(X_1 \bullet X_2 \bullet \dots \bullet Xn)' = X'_1 + X'_2 + \dots + X'n$ (Θεωρήματα De Morgan)
  - Παράδειγμα : Δίδεται η $F = (W' \bullet X) + (X \bullet Y) + [W \bullet (X' + Z')]$.
  - Τότε $F' = ((W')' + X') \bullet (X' + Y') \bullet [W' + (X \bullet Z)] =$
    $= (W + X') \bullet (X \bullet Y)' \bullet [W' + (X \bullet Z)]$

- $F(X_1, X_2, \dots, Xn) = X_1 \bullet F(1, X_2, \dots, Xn) + X_1' \bullet F(0, X_2, \dots, Xn)$
- $F(X_1, X_2, \dots, Xn) = [X_1 + F(0, X_2, \dots, Xn) \bullet [X_1' + F(1, X_2, \dots, Xn)]$ (Θεωρήματα Shannon)
  - Παράδειγμα : Δίδεται η $F(X, W, Z) = X + W \bullet Z$.
  - Τότε $F(0, W, Z) = W \bullet Z$ και $F(1, W, Z) = 1$.
  - Αρα $F = X \bullet 1 + X' \bullet W \bullet Z$ καθώς και $F = (X + W \bullet Z) \bullet (X' + 1)$

- **Αναπαράσταση μεγεθών Κωδικοποίηση**

Συστήματα αριθμών

- **Δεκαδικό σύστημα**

$$D_{10} = \left(d_n 10^n + d_{n-1} 10^{n-1} + \ldots d_1 10^1 + d_0 10^0 + d_{-1} 10^{-1} + \ldots d_{-n} 10^{-n}\right)$$

- **Παράδειγμα**

$$503{,}14 =$$

$$500 + 0 + 3 + \frac{1}{10} + \frac{4}{100} =$$

$$5 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2}$$

Δυαδικό σύστημα

- **Στο δυαδικό σύστημα, που έχει βάση το 2, υπάρχουν δύο ψηφία, το 0 και το 1:**

$$B_2 = b_n 2^n + b_{n-1} 2^{n-1} + \ldots b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + \ldots b_{-m} 2^{-m}$$

- **Παράδειγμα**

1010

$$1x2^3 + 0x2^2 + 1x2^1 + 0x2^0$$

- **Γενικά ένας δυαδικός αριθμός με n ψηφία μπορεί να παραστήσει ένα εύρος από 2^n δεκαδικούς αριθμούς**
- **2 ψηφία (0_3), 5 ψηφία (0_31), 8 ψηφία (0_255)**

Μετατροπή δεκαδικού σε δυαδικό

- **Μετατροπή ενός ακέραιου δεκαδικού σε δυαδικό → χρησιμοποιείται η διαδικασία της διαδοχικής διαίρεσης**

- **Παράδειγμα:**

Μετατροπή του 19_{10} στον αντίστοιχο δυαδικό

19/2= πηλίκο 9 και υπόλοιπο 1 άρα $b_0=1$

9/2= πηλίκο 4 και υπόλοιπο 1 άρα $b_1=1$

4/2= πηλίκο 2 και υπόλοιπο 0 άρα $b_2=0$

2/2= πηλίκο 1 και υπόλοιπο 0 άρα $b_3=0$

1/2= πηλίκο 0 και υπόλοιπο 1 άρα $b_4=1$

$$B_2=10011=19_{10}$$

Μετατροπή του κλασματικού μέρους ενός δεκαδικού αριθμού στον αντίστοιχο δυαδικό:

- Διαδοχικοί πολλαπλασιασμοί. Επαναλαμβάνεται η διαδικασία μέχρι να προκύψει κλασματικό μέρος μηδέν ή να επιτευχθεί η επιθυμητή ακρίβεια.
- Παράδειγμα:
  - Μετατροπή του 0,375 στον αντίστοιχο δυαδικό

0,375 x 2 = 0,75 ακέραιο μέρος 0, κλασματικό 0,75 $b_{-1}=0$

0,75 x 2 = 1,5 ακέραιο μέρος 1, κλασματικό 0,5 $b_{-2}=1$

0,5 x 2 = 1,0 ακέραιο μέρος 1, κλασματικό 0 $b_{-3}=1$

B_2= : $,011_2$

- Μετατροπή του 28,375 στον αντίστοιχο δυαδικό
- Απάντηση: B_2= : $11100,011_2$

Βασικές λογικές πράξεις – λογικές πύλες

- **Μία λογική πράξη μεταξύ μεταβλητών είναι μία συνάρτηση που ορίζεται από έναν πίνακα αληθείας (truth table).**
- **Το αντίστοιχο κύκλωμα ονομάζεται λογική ή ψηφιακή πύλη και παριστάνεται από ένα σύμβολο.**
- **Τα δυαδικά ψηφία 1 και 0 (αληθής (true), ψευδής (false)), στη φυσική τους υπόσταση είναι δυο διακριτά επίπεδα ηλεκτρικής τάσης (συνήθως στην ιδανική περίπτωση 5V και 0V).**

Δυνατοί πίνακες αληθείας στο δυαδικό σύστημα

- **Ένας πίνακας αληθείας παριστάνει τη συνάρτηση μεταξύ των εισόδων και της εξόδου ενός λογικού συστήματος. Για δυο εισόδους υπάρχουν τέσσερις πιθανοί συνδυασμοί πραγματικών τιμών: FF, FT, TF, TT**

Όλοι οι πίνακες αληθείας για δύο εισόδους Α. Β και μία έξοδο Ζ

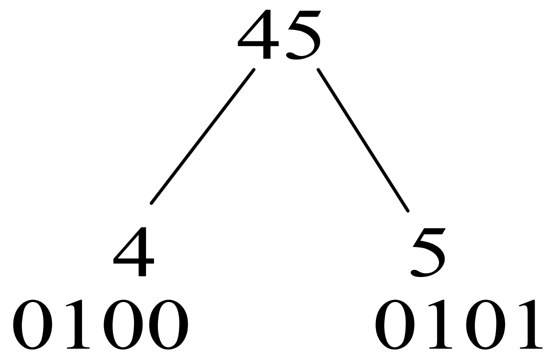
	τιμές εισόδου					
A	F	F	T	T		
B	F	T	F	T	Συνάρτηση (έξοδος Ζ)	Σύμβολο
0	F	F	F	F	πάντοτε 0	0
1	F	F	F	T	AND	$A \cdot B$
2	F	F	T	F	-	-
3	F	F	T	T	είσοδος Α	A
4	F	T	F	F	-	-
5	F	T	F	T	είσοδος Β	B
6	F	T	T	F	XOR	$A \oplus B$
7	F	T	T	T	OR	$A + B$
8	T	F	F	F	NOR	$\overline{A + B}$
9	T	F	F	T	XNOR	$\overline{A \oplus B}$
10	T	F	T	F	Not B	$\overline{B}$
11	T	F	T	T	-	-
12	T	T	F	F	Not A	$\overline{A}$
13	T	T	F	T	-	-
14	T	T	T	F	NAND	$\overline{A \cdot B}$
15	T	T	T	T	πάντοτε 1	1

Άλλοι τρόποι δυαδικής κωδικοποίησης

- **Εκτός από την κανονική δυαδική κωδικοποίηση υπάρχουν και άλλοι τρόποι δυαδικής κωδικοποίησης οι οποίοι χρησιμοποιούνται σε διάφορες περιπτώσεις:**
- **Κωδικοποίηση BCD (Binary Coded Decimal)**
- **Η κωδικοποίηση καθιστά δυνατή την απλή μετατροπή μεταξύ δυαδικού και δεκαδικού αριθμού. Κάθε ψηφίο ενός δεκαδικού αριθμού αντικαθίσταται από 4 bits του αντίστοιχου δυαδικού του**

Μετατροπή του 4510 σε BCD

45_{10}=01000101_{BCD}



Μετατροπή από BCD σε δεκαδικό

Η δυαδική λέξη χωρίζεται σε ομάδες των 4bits ξεκινώντας από το λιγότερο σημαντικό ψηφίο. Κατόπιν η κάθε ομάδα μετατρέπεται στον αντίστοιχο δεκαδικό

Μετατροπή 1010011_{BCD} σε δεκαδικό

Πρόσθεση μηδενικού . Χωρισμός σε ομάδες των 4. Μετατροπή της κάθε ομάδας στον αντίστοιχο δεκαδικό

$[0101][0011]_{BCD}=53_{10}$

Κώδικας Gray

- Λύνει κάποιο πρόβλημα της δυαδικής κωδικοποίησης
- **Σε μετρήσεις της θέσης ενός αντικειμένου, θα μπορούσε να φαίνεται ότι γειτονικές θέσεις του αντικειμένου διαφέρουν περισσότερο από ένα bit, εάν χρησιμοποιηθεί η απευθείας δυαδική κωδικοποίηση.**

Decimal	Gray Code	
0	0	
1	1	
2	11	Set bit 1. Reflect bit 0
3	10	Set bit 2. Reflect bits 1 and 0
4	110	
5	111	
6	101	
7	100	
8	1100	Set bit 3. Reflect bits 2, 1 and 0
9	1101	
10	1111	
11	1110	
12	1010	
13	1011	
14	1001	
15	1000	
16	11000	Set bit 4. Reflect bits 3, 2, 1 and 0

Κώδικας Gray

- Για να μετατρέψουμε αριθμούς σε κώδικα Gray κάνουμε χρήση της συνάρτησης XOR, η οποία έχει πίνακα αλήθειας:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

- Και χρησιμοποιούμε τον παρακάτω αλγόριθμο.

Κώδικας Gray

$$g_n = b_n$$

$$g_{n-1} = b_n \oplus b_{n-1}$$

$$g_{n-2} = b_{n-1} \oplus b_{n-2}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$g_1 = b_2 \oplus b_1$$

Κώδικας Gray

Παράδειγμα : Να μετατραπεί ο 43_{10} σε κώδικα GRAY.

$$43_{10} = 101011_2$$

1 0 1 0 1 1

1 1 1 1 1 0

δηλαδή $43_{10} = 101011_2 = 111110_{GRAY}$.

Κώδικας Gray

$$b_n = g_n$$

$$b_{n-1} = b_n \oplus g_{n-1}$$

$$b_{n-2} = b_{n-1} \oplus g_{n-2}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$b_1 = b_2 \oplus g_1$$

Κώδικας Gray

ΚΩΔΙΚΑΣ GRAY (5 bits)

0	0	0	0	0	0		1	1	0	0	0	16
0	0	0	0	1	1		1	1	0	0	1	17
0	0	0	1	1	2		1	1	0	1	1	18
0	0	0	1	0	3		1	1	0	1	0	19
0	0	1	1	0	4		1	1	1	1	0	20
0	0	1	1	1	5		1	1	1	1	1	21
0	0	1	0	1	6		1	1	1	0	1	22
0	0	1	0	0	7		1	1	1	0	0	23
0	1	1	0	0	8		1	0	1	0	0	24
0	1	1	0	1	9		1	0	1	0	1	25
0	1	1	1	1	10		1	0	1	1	1	26
0	1	1	1	0	11		1	0	1	1	0	27
0	1	0	1	0	12		1	0	0	1	0	28
0	1	0	1	1	13		1	0	0	1	1	29
0	1	0	0	1	14		1	0	0	0	1	30
0	1	0	0	0	15		1	0	0	0	0	31
						1	1	0	0	0	0	32

Κώδικες με ανίχνευση σφάλματος

- Σε όλα τα συστήματα εμφανίζονται σφάλματα. Για παράδειγμα κάποιο 1 μπορεί να μετατραπεί σε ψηφίο 0, είτε κατά τη μετάδοση, είτε γιατί το ψηφιακό σύστημα δεν λειτούργησε σωστά. Μία απλή μέθοδος, ανίχνευσης του σφάλματος, είναι η χρήση του κώδικα ανίχνευσης λάθους, η οποία χρησιμοποιεί ένα επιπλέον ψηφίο ισοτιμίας (parity bit).
- Κώδικες ισοτιμίας
- Δυο είδη { *άρτια ισοτιμία* / *περιττή ισοτιμία*

Κώδικες με ανίχνευση σφάλματος

- **Κώδικας περιττής ισοτιμίας**
  - **Το ψηφίο ισοτιμίας είναι 0 αν το σύνολο των ψηφίων, 1, είναι περιττό. Το ψηφίο ισοτιμίας είναι 1 αν το σύνολο των ψηφίων, 1, είναι άρτιο.**
  - **Για παράδειγμα η δυαδική λέξη 010001 έχει αριθμό ψηφίων '1' άρτιο, συνεπώς θα μεταδοθεί με ψηφίο ισοτιμίας '1', είτε: 1 | 010001**

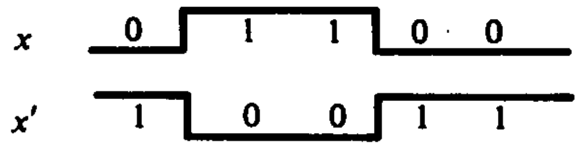
- **Κώδικας άρτιας ισοτιμίας**
  - **Αντίστροφος της περιττής ισοτιμίας. Το ψηφίο ισοτιμίας είναι 1 αν το σύνολο των '1' είναι περιττό. Το ψηφίο ισοτιμίας είναι 0 αν το σύνολο των '1' είναι άρτιο**
  - **Για παράδειγμα η δυαδική λέξη 10110 έχει αριθμό ψηφίων '1' περιττό, συνεπώς θα μεταδοθεί με ψηφίο ισοτιμίας '1', είτε: 1 | 10110**

Γιατί τόση θεωρία ?

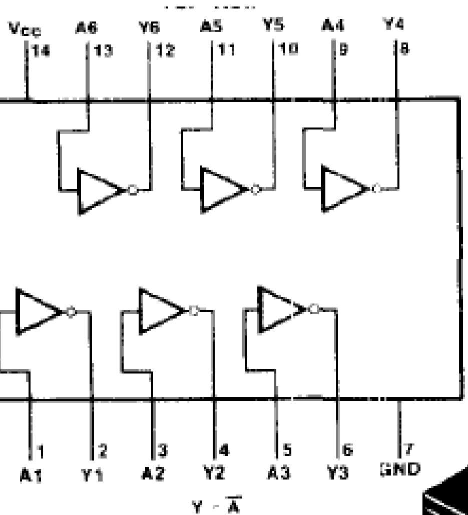
- **Όσα είδαμε έχουν άμεση εφαρμογή στο πραγματικό κόσμο.**
- **Υπάρχουν έτοιμοι ψηφιακοί σχεδιασμοί οι οποίοι υλοποιούν τις βασικές λογικές συναρτήσεις**
- **Οι σχεδιασμοί αυτοί ονομάζονται ψηφιακές πύλες.**
- **Θα εισάγουμε :**
  - **κάποια σχηματικά για την απεικόνιση αυτών.**
  - **Ενα πίνακα που δείχνει τη λογική συνάρτηση του κυκλώματος**
  - **Ο πίνακας αυτός είναι ο πίνακας αληθείας της συνάρτησης, άρα και του ψηφιακού σχεδιασμού.**

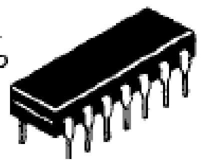
Ο αντιστροφέας

- **Παράγει το συμπλήρωμα μιας δυαδικής μεταβλητής.**
- **Μπορούμε αντί για το πλήρες σχηματικό του αντιστροφέα, να χρησιμοποιούμε μόνο το κύκλο.**
- **Είναι διαθέσιμο ως ψηφιακό κύκλωμα σε 6άδες, εντός ενός ολοκληρωμένου με κωδικό 7404.**



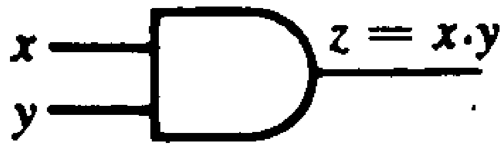
x	x'
0	1
1	0

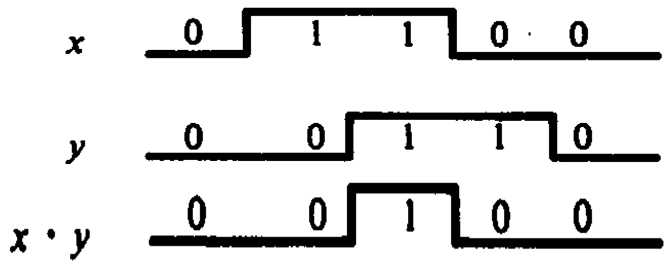




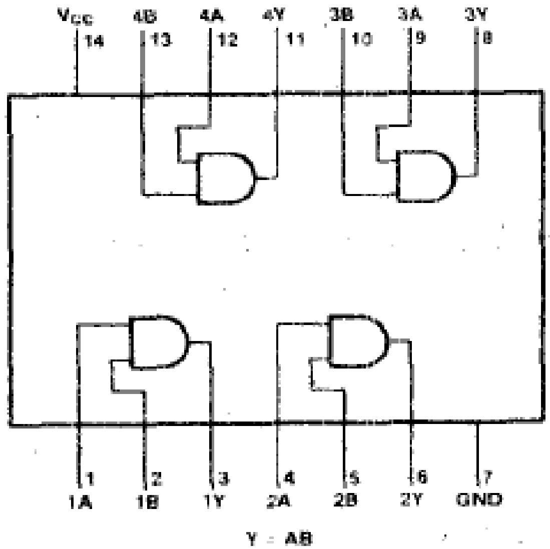
Η πύλη AND δύο μεταβλητών

- Εκτελεί το λογικό AND των δύο μεταβλητών.
- Είναι διαθέσιμο ως ψηφιακό κύκλωμα σε 4άδες, εντός ενός ολοκληρωμένου με κωδικό 7408.



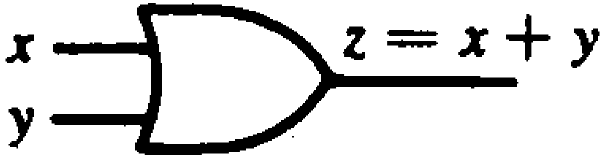


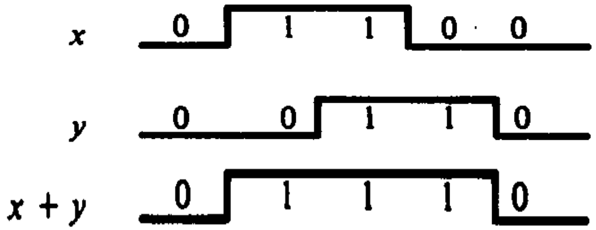
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1



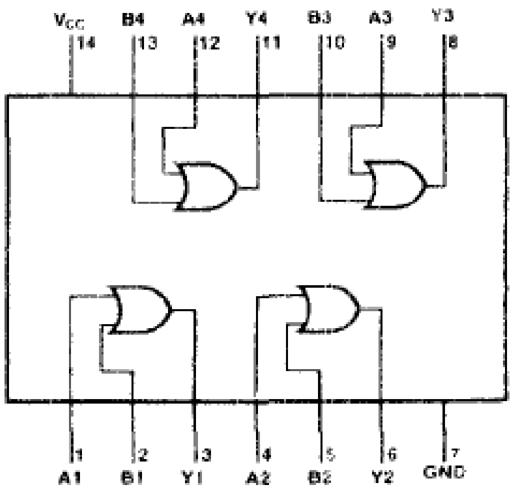
Η πύλη OR δύο μεταβλητών

- Εκτελεί το λογικό OR των δύο εισόδων.
- Είναι διαθέσιμο ως ψηφιακό κύκλωμα σε 4άδες, εντός ενός ολοκληρωμένου με κωδικό 7432.



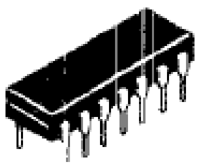


x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1



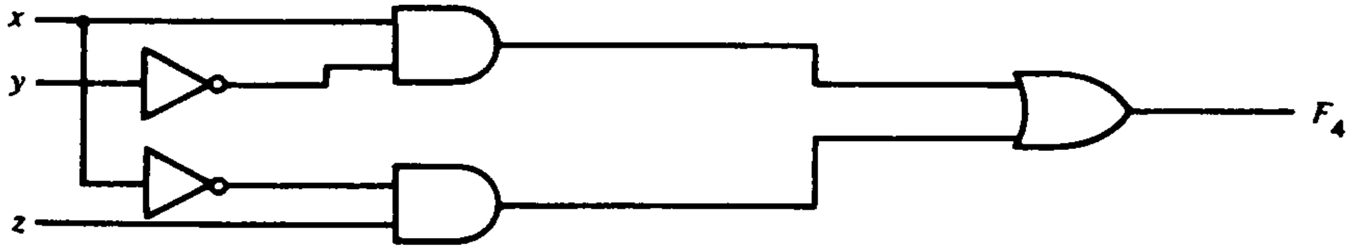
Dual-In-Line and Flat Package

Y = A + B



Λογικό διάγραμμα

- Η χρήση των σχηματικών των διαφόρων πυλών και η διασύνδεσή τους οδηγεί σε ένα λογικό διάγραμμα που περιγράφει κάποια πιο σύνθετη συνάρτηση.
- Για παράδειγμα :



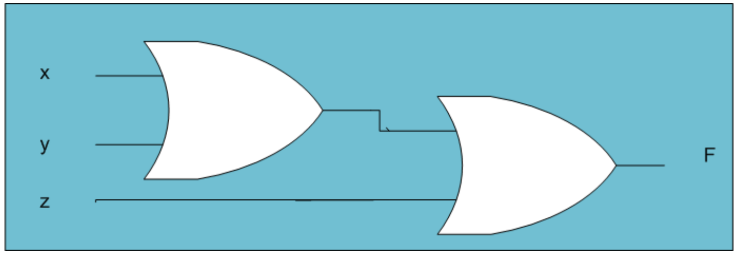
- Το παραπάνω είναι ένα διάγραμμα που περιγράφει τη συνάρτηση :

$$F_4(x, y, z) = x' \bullet z + y' \bullet x$$

- **Ποιο είναι το λογικό διάγραμμα της G(X, W, Y, Z) = [X' • Y + X] • (Z + W') ;**

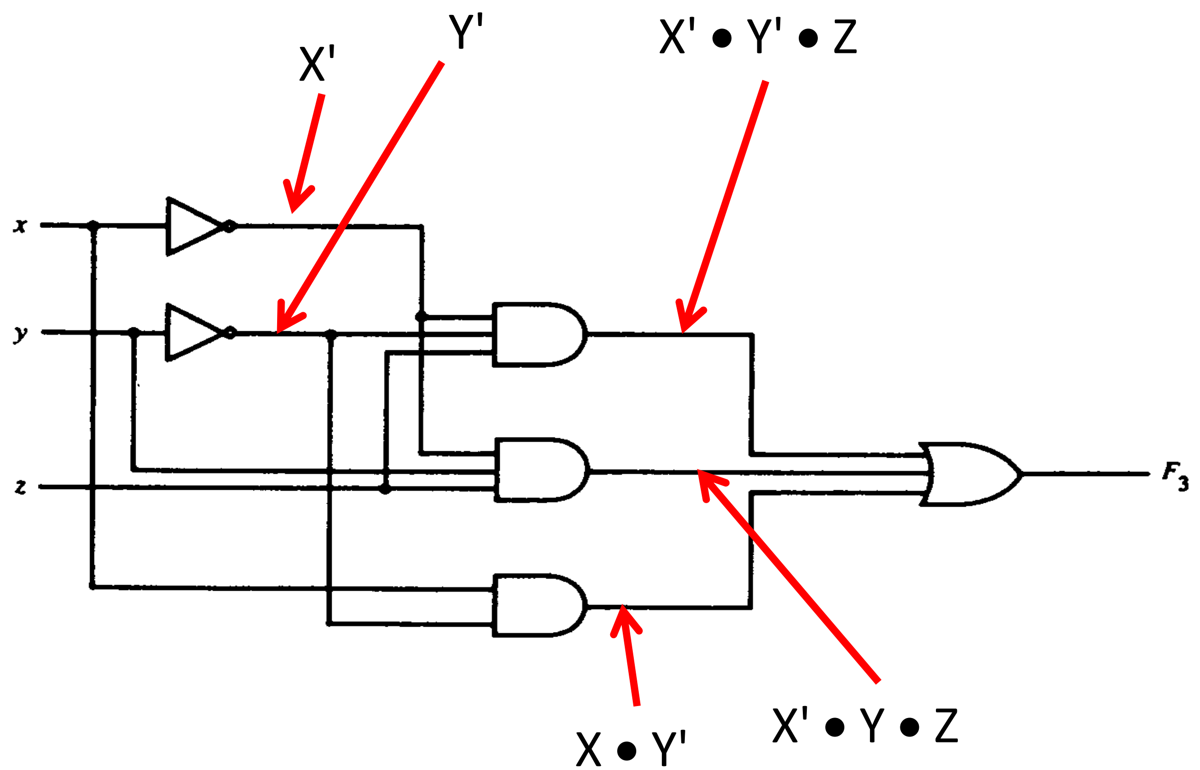
Επέκταση σε περισσότερες εισόδους

- Αν ήθελα το λογικό διάγραμμα της F(x, y, z) = x + y + z πως θα το έφτιαχνα ?
- Από τη θεωρία μου γνωρίζω ότι F(x, y, z) = x + y + z = (x + y) + z και συνεπώς θα μπορούσα να χρησιμοποιήσω κάτι τέτοιο :



- Μήπως υπάρχουν έτοιμες και πύλες περισσότερων εισόδων ?
- Φυσικά ! Αυτό όμως δε σημαίνει ότι υπάρχουν οσωνδήποτε εισόδων.
- Κι εδώ πρέπει να διαχωρίσουμε το πραγματικό από τον ιδεατό κόσμο :
  - Στον ιδεατό κόσμο μπορούμε να φτιάχνουμε λογικά διαγράμματα με πύλες όσων εισόδων θέλουμε.
  - Στη πράξη αυτά θα υλοποιηθούν με όσα πραγματικά υπάρχουν.

Ποια συνάρτηση επιτελεί αυτό το κύκλωμα ?



Ποιες άλλες συναρτήσεις και πύλες υπάρχουν ?

- **Πόσες διαφορετικές συναρτήσεις των ν μεταβλητών υπάρχουν ?**
  - **Μια συνάρτηση ν μεταβλητών, έχει 2^v πιθανές τιμές εισόδου.**
  - **Διαφορετική συνάρτηση ←→ διαφορετική έξοδος έστω και για κάποιον συνδυασμό εισόδου. Αφού η έξοδός μου για κάθε συνδυασμό εισόδου μπορεί να είναι 0 ή 1 θα υπάρχουν 2^{2^v} συναρτήσεις.**
- **Για ν =2 ποιες είναι οι συναρτήσεις που υπάρχουν και ποιες από αυτές είναι χρήσιμες ?**

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
Σύμβολο τελεστή			$\cdot$	/		/		$\oplus$	+	$\downarrow$	$\odot$	′	$\subset$	′	$\supset$	$\uparrow$	

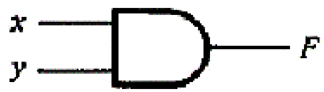
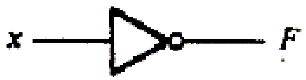
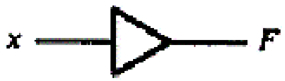
- **Δυο σταθερές 0, 1.**
- **Τέσσερις unary συμπληρώματος/ μεταφοράς.**
- **Δέκα συναρτήσεις με δυαδικούς τελεστές.**

Μας ενδιαφέρουν επίσης οι :

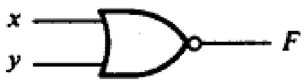
x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Σύμβολο τελεστή			$\cdot$	$/$		$/$		$\oplus$	$+$	$\downarrow$	$\odot$	$'$	$\subset$	$'$	$\supset$	$\uparrow$	

- **F_{14} που είναι συμπληρωματική της AND. Θα τη λέμε NAND (not – AND).**
- **F_8 που είναι συμπληρωματική της OR. Θα τη λέμε NOR (not – OR).**
- **F_6 που μας δίνει 1 μόνο όταν μόνο 1 είσοδος είναι στο λογικό 1 (για περισσότερες εισόδους, όταν ο αριθμός των 1 στις εισόδους είναι περιττός). Θα την ονομάζουμε αποκλειστικό-OR (eXclusive-OR) – XOR.**
- **F_9 που μας δίνει 1 μόνο όταν μόνο 0 ή 2 είσοδοι είναι στο λογικό 1 (για περισσότερες εισόδους όταν ο αριθμός των 1 στις εισόδους είναι άρτιος). Θα την ονομάζουμε συνάρτηση ισοδυναμίας (not eXclusive-OR) – XNOR.**

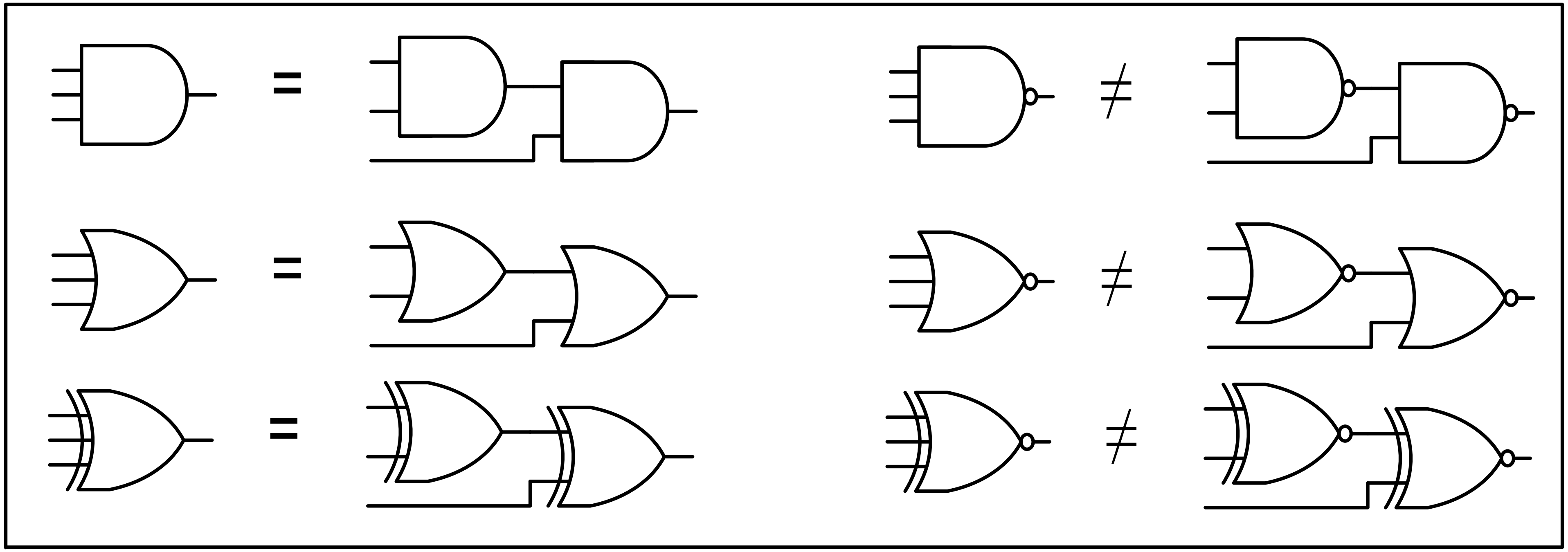
Ετσι έχουμε για τα λογικά μας διαγράμματα : (1/

Όνομα	Γραφικό Σύμβολο	Αλγεβρική Συνάρτηση	Πίνακας Αληθείας
AND ΚΑΙ		$F = xy$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR Ή		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Αντιστροφέας		$F = x'$	x | F 0 | 1 1 | 0
Απομονωτής		$F = x$	x | F 0 | 0 1 | 1

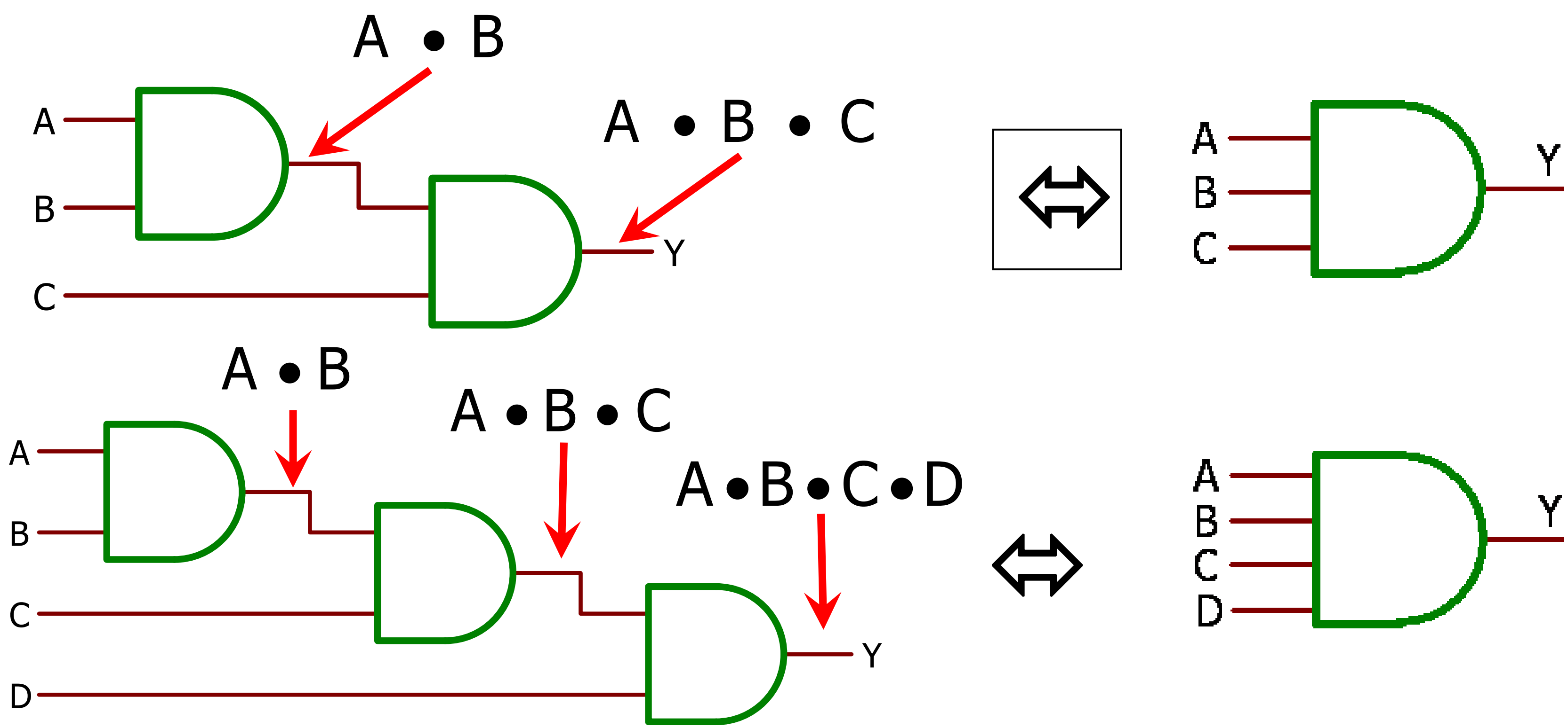
Ετσι έχουμε για τα λογικά μας διαγράμματα : (2/

Όνομα	Γραφικό Σύμβολο	Αλγεβρική Συνάρτηση	Πίνακας Αληθείας
NAND ΟΧΙ-ΚΑΙ		$F = (xy)'$	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR ΟΥΤΕ		$F = (x + y)'$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
XOR Αποκλειστό-Ή		$F = xy' + x'y$ $= x \oplus y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Ισοδυναμία ή Αποκλειστικό -ΟΥΤΕ		$F = xy + x'y'$ $= x \odot y$	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

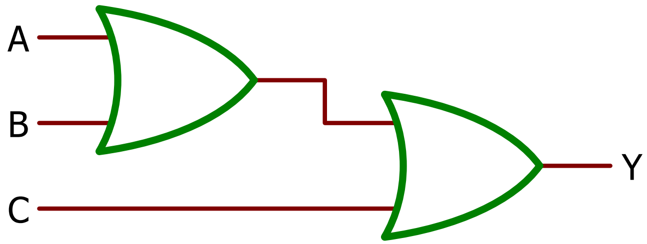
Επέκταση σε περισσότερες εισόδους ?



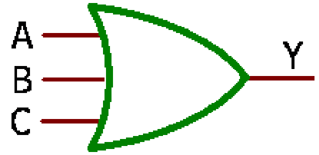
Επέκταση σε περισσότερες εισόδους

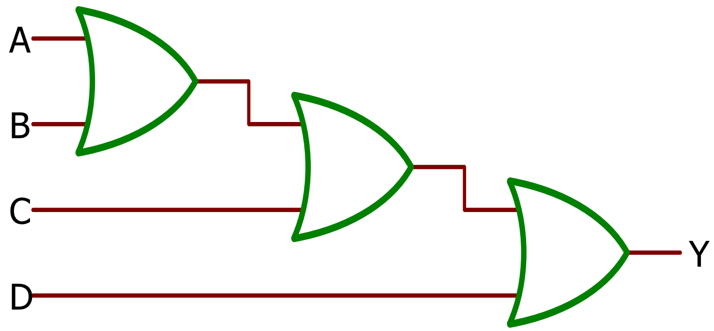


Επέκταση σε περισσότερες εισόδους

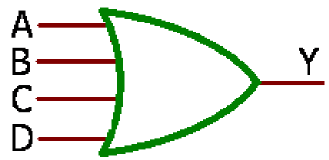


⇔

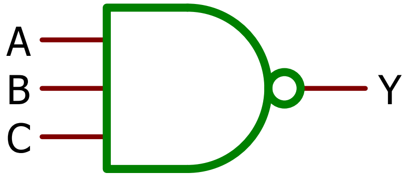




⇔

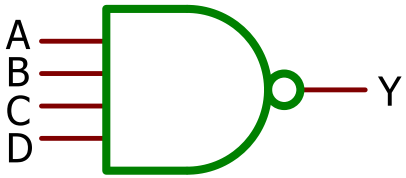


Επέκταση σε περισσότερες εισόδους



A	B	C	Y
X	X	0	1
X	0	X	1
0	X	X	1
1	1	1	0

$$Y = \overline{A \bullet B \bullet C}$$



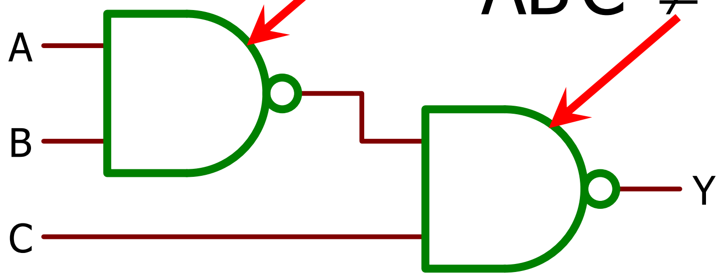
A	B	C	D	Y
X	X	X	0	1
X	X	0	X	1
X	0	X	X	1
0	X	X	X	1
1	1	1	1	0

$$Y = \overline{A \bullet B \bullet C \bullet D}$$

Επέκταση σε περισσότερες εισόδους

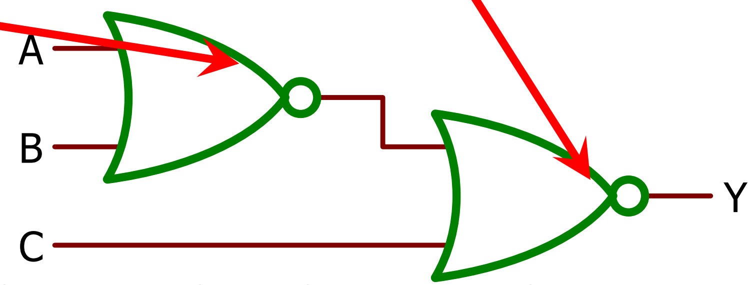
$\overline{AB}$

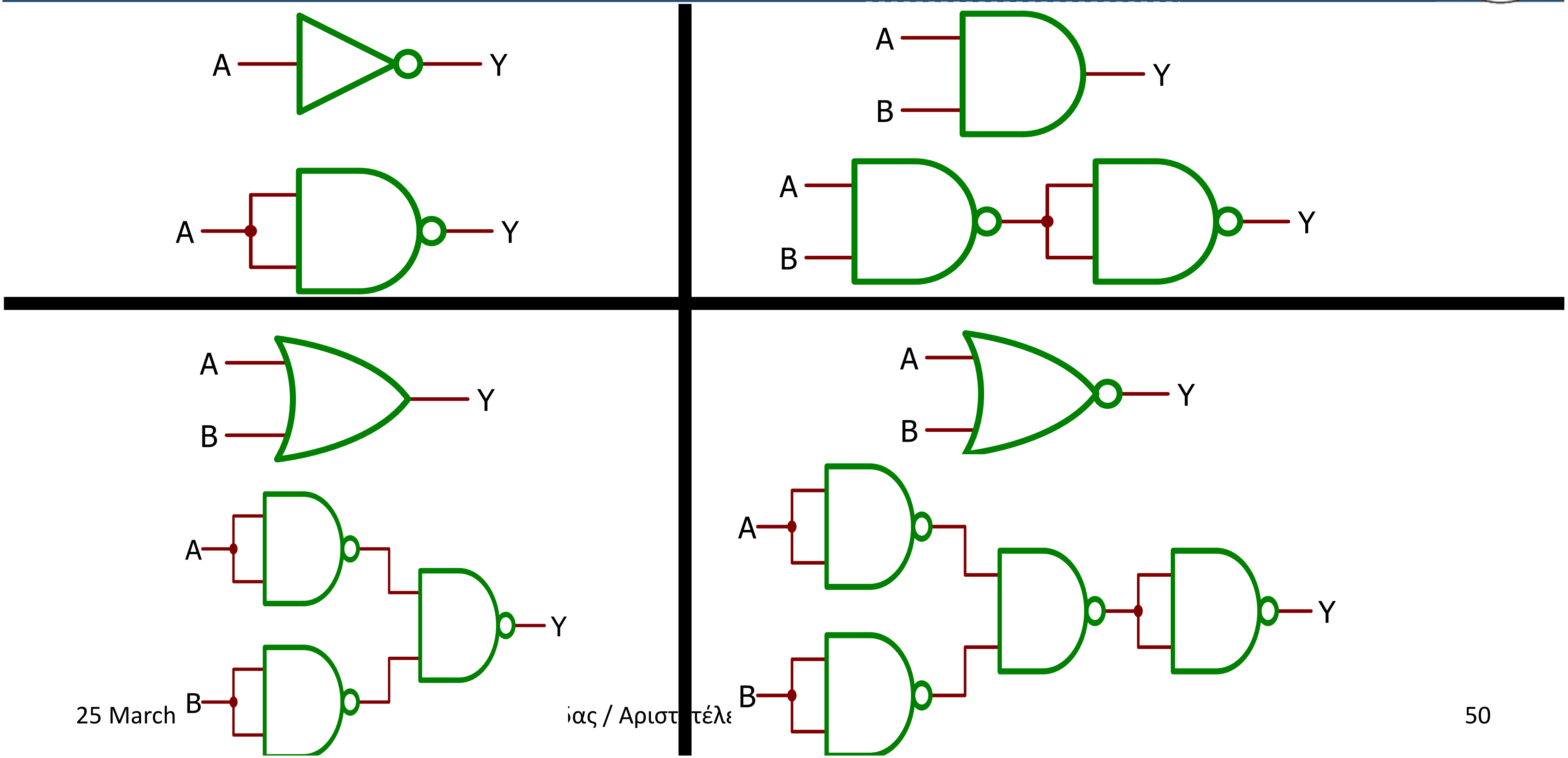
$\overline{\overline{AB}C} \neq \overline{ABC}$

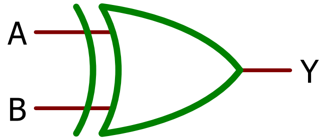


$\overline{\overline{A+B}+C} \neq \overline{A+B+C}$

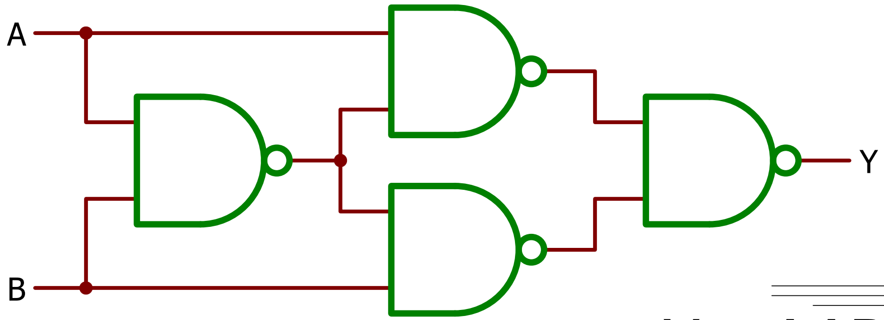
$\overline{A+B}$




A
Y
B



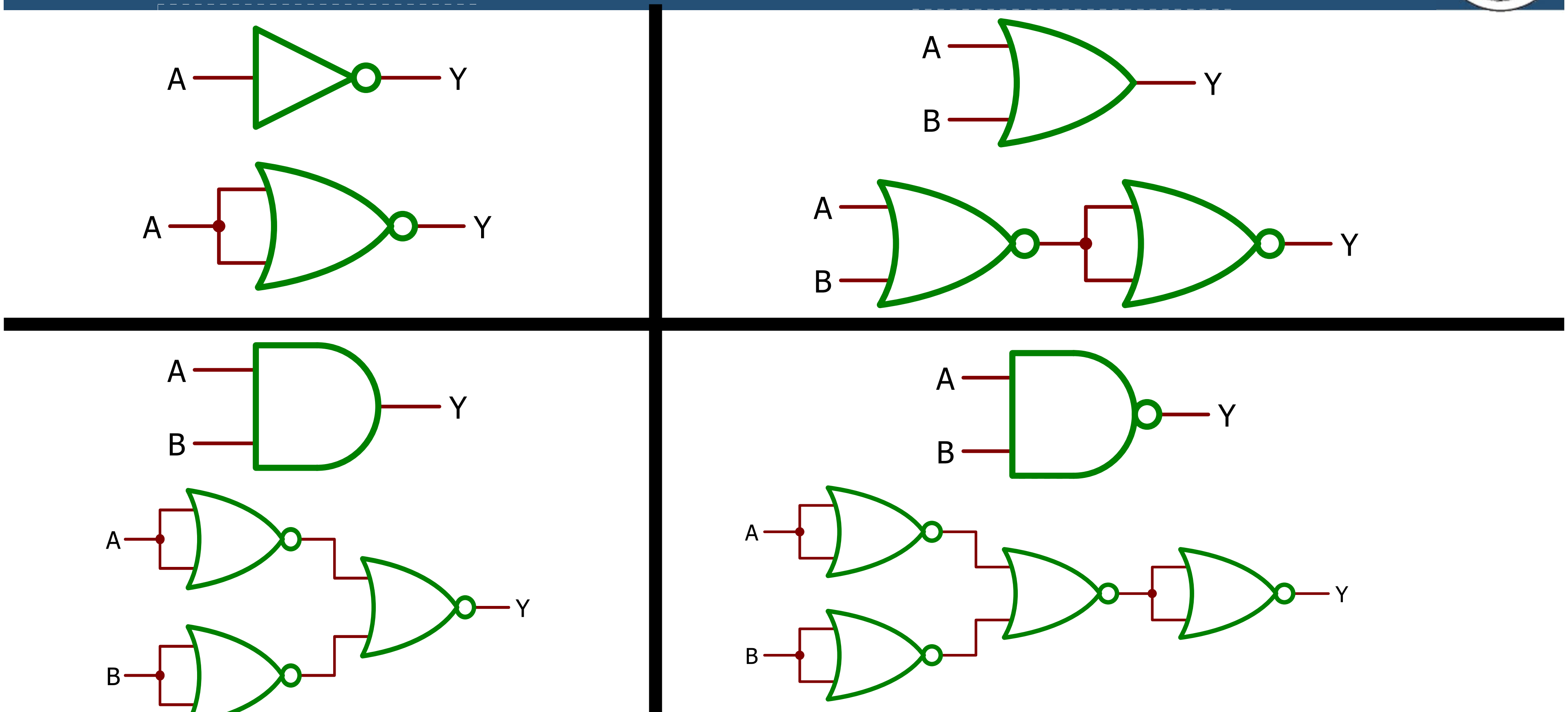
$$\boxed{Y = A \oplus B}$$



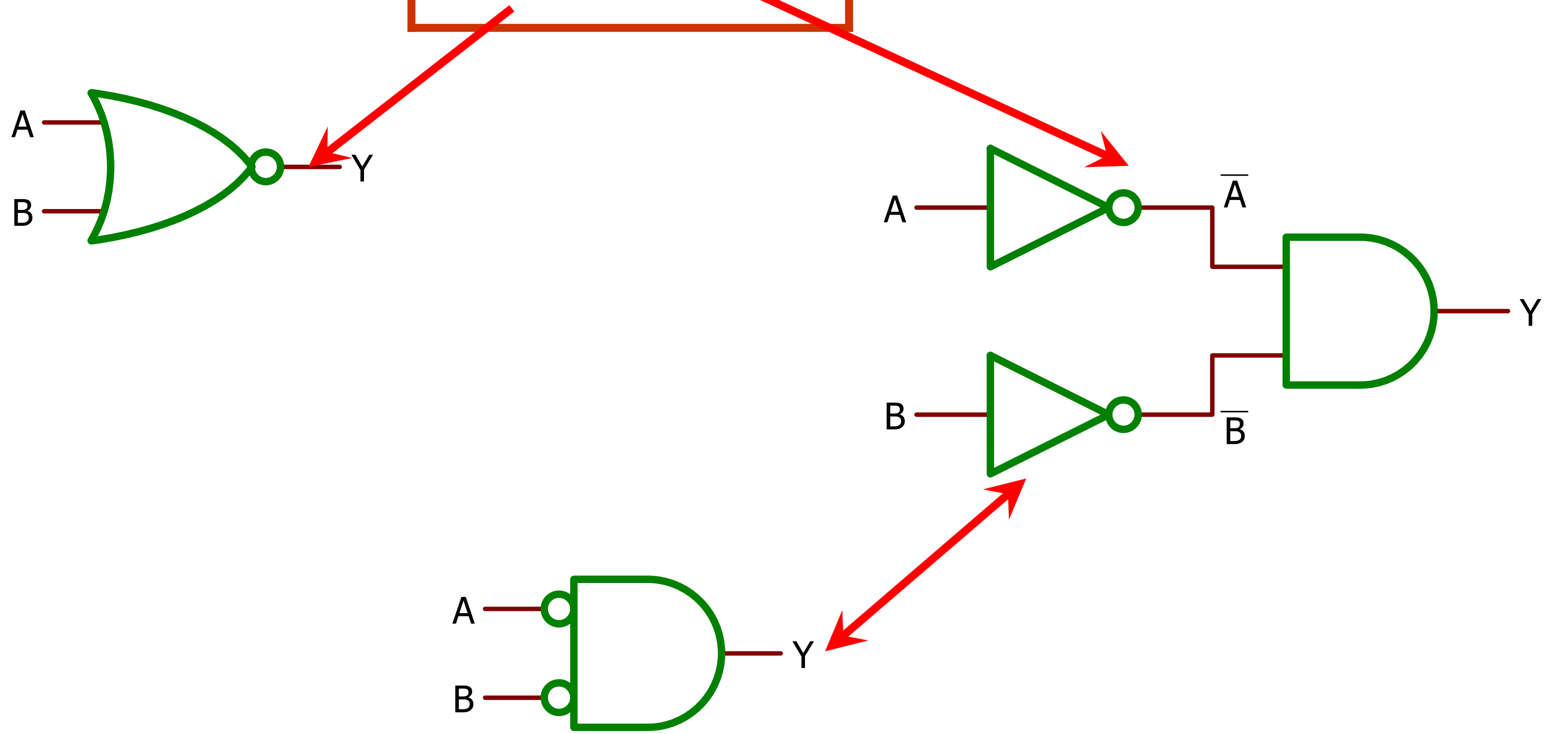
$$Y = \overline{\overline{A\overline{AB}}\;\overline{B\overline{AB}}} = A\overline{AB} + B\overline{AB} =$$

$$= A\left(\overline{A} + \overline{B}\right) + B\left(\overline{A} + \overline{B}\right) =$$

$$= \overline{A}B + A\overline{B} = A \oplus B$$


A
Y
A
Y
A
B
Y
A
B
Y
A
B
Y
A
B
Y
A
B
Y
A
B
Y

$$\boxed{\overline{A+B} = \overline{A}\cdot\overline{B}}$$



Κόστος πυλών

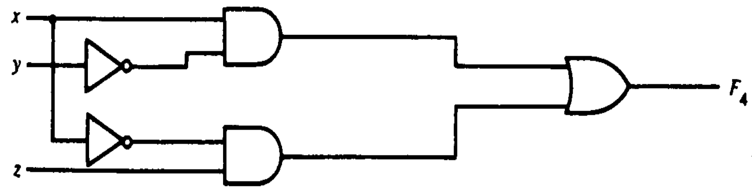
- Κοστίζουν όλες οι πύλες το ίδιο ?
- Όχι. Άλλωστε είναι χαρακτηριστικό ότι σε ένα ολοκληρωμένο κύκλωμα έχουμε 6 inverters και μόνο 4 OR, AND, …
- Το κόστος μιας πύλης εξαρτάται :
  - Από τον αριθμό των εισόδων της. Περισσότεροι είσοδοι => ↑ κόστος
  - Από τη συνάρτηση που επιτελεί :
    - NAND, NOR οι πιο απλές.
    - AND, OR, ελάχιστα πιο πολύπλοκες (σε επίπεδο transistor).
    - XOR, XNOR αρκετά πιο πολύπλοκες
    - Μεγαλύτερη πολυπλοκότητα => ↑ κόστος

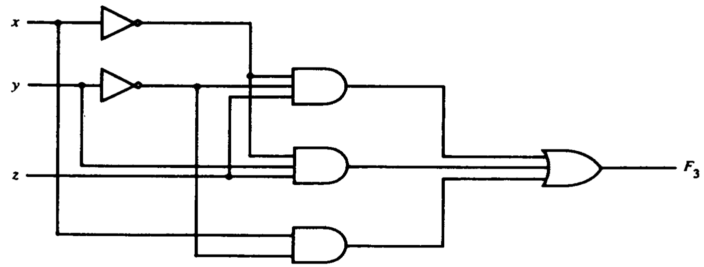
Ισοδύναμες και συμπληρωματικές συναρτήσεις

- Δύο συναρτήσεις των κ μεταβλητών είναι **ισοδύναμες**, αν για κάθε συνδυασμό τιμών των μεταβλητών εισόδου, οι συναρτήσεις οδηγούν στην ίδια έξοδο.
  - Συμπέρασμα 1 : Ισοδύναμες συναρτήσεις μπορεί να έχουν διαφορετικές αλγεβρικές εκφράσεις.
  - Συμπέρασμα 2 : Ισοδύναμες συναρτήσεις μπορεί να έχουν διαφορετικά λογικά διαγράμματα.
  - Συμπέρασμα 3 : Ισοδύναμες συναρτήσεις έχουν τον ίδιο πίνακα αληθείας.
- Δύο συναρτήσεις των κ μεταβλητών είναι **συμπληρωματικές**, αν για κάθε συνδυασμό των μεταβλητών εισόδου, οι συναρτήσεις παράγουν συμπληρωματικές τιμές.

Μεταξύ ισοδύναμων συναρτήσεων κάποιες είναι σαφώς προτιμητέες !

- Η F_3 είναι σαφώς πιο πολύπλοκη συνάρτηση από την F_4.
- Χρησιμοποιεί περισσότερες πύλες, μα ταυτόχρονα και πύλες με περισσότερες εισόδους.
- Παράλληλα χρησιμοποιεί 4 διαφορετικά ολοκληρωμένα έναντι 3 της F_4.
- Το μεγάλο συνεπώς ερώτημα που προκύπτει είναι πως θα βρω τη συνάρτηση με το ελάχιστο κόστος ανάμεσα στις ισοδύναμες ?
- Αυτό είναι το πιο ενδιαφέρον σημείο του μαθήματος : απλοποίηση συναρτήσεων





x	y	z	F_3	F_4
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Η θεωρία παρέχει τη μέθοδο της αλγεβρικής απλοποίησης

- **Θυμηθείτε ότι για την F_3 είχαμε ότι :**

$$F_3\ (X, Y, Z) = X \bullet Y' + X' \bullet Y \bullet Z + X' \bullet Y' \bullet Z$$

- **Από τη θεωρία βάσει του θεωρήματος των συνδυασμών γνωρίζω ότι :**

$$X' \bullet Y \bullet Z + X' \bullet Y' \bullet Z = X' \bullet Z$$

$$\text{Αρα } F_3 = X \bullet Y' + X' \bullet Z = F4.$$

- **Η αλγεβρική απλοποίηση δεν είναι πάντα τόσο εύκολη !**
- **Ακόμα χειρότερα, ποτέ δε γνωρίζω αν έχοντας κάνει κάποια στάδια απλοποίησης έχω βρει την απολύτως ελάχιστη ισοδύναμη συνάρτηση.**
- **Η αλγεβρική απλοποίηση πρέπει να χρησιμοποιείται για συναρτήσεις πολύ λίγων μεταβλητών από έμπειρους σχεδιαστές.**

Παραδείγματα αλγεβρικής απλοποίησης

- $F(X, Y, Z) = X \bullet Y' \bullet Z + X' \bullet Y \bullet Z + Y \bullet Z =$

$$= X \bullet Y' \bullet Z + Y \bullet Z \qquad \text{(Απορρόφηση)}$$

$$= Z \bullet (X \bullet Y' + Y) \qquad \text{(Επιμεριστική)}$$

$$= Z \bullet (X + Y) \qquad \text{(1ο αποδειχθέν θεώρημα)}$$

- Υλοποιείστε με τον ελάχιστο αριθμό πυλών τη

$F(X, Y, Z) = X \bullet Y' \bullet Z + X \bullet Y' \bullet Z + X \bullet Y \bullet Z' =$

$$= X \bullet Y' \bullet Z + X \bullet Y \bullet Z' \qquad \text{(Αυτοαπορρόφηση)}$$

$$= X \bullet (Y' \bullet Z + Y \bullet Z') \qquad \text{(Επιμεριστική)}$$

$$= X \bullet (Y \oplus Z) \qquad \text{(συνάρτηση XOR)}$$

Προβλήματα & Αλγόριθμοι

- **Η αλγεβρική απλοποίηση**
  - **Είναι δύσκολη**
  - **Μη ντετερμινιστική**
  - **Χωρίς σίγουρο αποτέλεσμα**
- **Θα ήθελα συνεπώς μια ντετερμινιστική μεθοδολογία.**
- **Στη γλώσσα των υπολογιστών μια μεθοδολογία που αποτελείται από μικρά κατανοητά βήματα και η οποία αν ακολουθηθεί παράγει τη λύση σε κάποιο πρόβλημα ονομάζεται αλγόριθμος.**
- **Για να εφαρμοστεί κάποιος αλγόριθμος όμως απαιτείται να υπάρχει μια σταθερή αρχική μορφή του προβλήματος.**
- **Πριν λοιπόν δώσουμε αλγόριθμο απλοποίησης, πρέπει πρώτα να εισάγουμε πρότυπες μορφές έκφρασης μιας συνάρτησης.**
- **Σε αυτό θα μας βοηθήσουν οι ελαχιστόροι και οι μεγιστόροι.**

Πίνακας αλήθειας

- Ισοδύναμες συναρτήσεις έχουν ποικίλες αλγεβρικές αναπαραστάσεις, ποικίλα λογικά διαγράμματα, αλλά κοινό πίνακα αλήθειας.
- Ο πίνακας αλήθειας μας δίνει τη τιμή μιας συνάρτησης για κάθε πιθανό συνδυασμό εισόδων.
- Συνήθως διατάσσουμε τους συνδυασμούς εισόδων σαν αύξοντες δυαδικούς αριθμούς.
- Ο πίνακας αλήθειας μιας λογικής συνάρτησης ν μεταβλητών έχει 2^{ν} γραμμές.

Οροι, αθροίσματα, γινόμενα

- **Μια μεταβλητή στη κανονική ή τη συμπληρωματική της μορφή είναι ένας όρος**
  - **Παραδείγματα όρων : X, X', Y, Z'**
- **Ενα γινόμενο είναι είτε ένας όρος είτε το λογικό AND δύο ή περισσότερων όρων**
  - **Παραδείγματα γινομένων : X, X • Z', Y • X', X • Z' • Y’**
- **Ενα άθροισμα είναι είτε ένας όρος είτε το λογικό OR δύο ή περισσοτέρων όρων**
  - **Παραδείγματα αθροισμάτων : X, X + Z', Y+ X', X + Z' + Y'**

Οροι, αθροίσματα, γινόμενα

- Αθροισμα γινομένων (sum of products – SOP) είναι κάθε άθροισμα του οποίου οι όροι είναι γινόμενα
    - Παράδειγμα SOP : X + X • Z' + Y • X' + X • Z' • Y'
- Γινόμενο αθροισμάτων (product of sums – POS) είναι κάθε γινόμενο του οποίου οι όροι είναι αθροίσματα.
    - Παράδειγμα POS : X • (X + Z') • (Y+ X') • (X + Z' + Y')
- Κανονικός όρος είναι ένα άθροισμα ή ένα γινόμενο, στο οποίο κάθε μεταβλητή (κανονική ή συμπληρωμένη) εμφανίζεται μόνο 1 φορά. Κάθε μη κανονικός όρος μπορεί να μετατραπεί μέσω απλοποίησης σε κανονικό.
    - Κανονικοί όροι : X • Z' • Y', X + Z' + Y'
    - Μη κανονικοί όροι : X • Z' • Y' • X, X + Z' + Y' + Z

Ελαχιστόροι - Μεγιστόροι

- Κάθε κανονικός όρος - γινόμενο μιας συνάρτησης κ μεταβλητών, είναι ελαχιστόρος αν περιέχει κ όρους. Κάθε συνάρτηση κ μεταβλητών έχει $2^κ$ ελαχιστόρους.
  - Η F(X, Y) έχει τους ελαχιστόρους : X' • Y΄, X΄ • Y, X • Y' και X • Y
  - Κάθε ελαχιστόρος αντιπροσωπεύει μία γραμμή του πίνακα αληθείας

- Κάθε κανονικός όρος – άθροισμα μιας συνάρτησης κ μεταβλητών, είναι μεγιστόρος αν περιέχει κ όρους. Κάθε συνάρτηση κ μεταβλητών έχει $2^κ$ μεγιστόρους.
  - Η F(X, Y) έχει τους μεγιστόρους : X' + Y΄, X΄ + Y, X + Y' και X + Y
  - Κάθε μεγιστόρος αντιπροσωπεύει μία γραμμή του πίνακα αληθείας

Ελαχιστόροι – Μεγιστόροι και πίνακας αλήθειας

- Μεταξύ πίνακα αλήθειας και ελαχιστόρων (μεγιστόρων) υπάρχει μια στενή σχέση. Ο ελαχιστόρος (μεγιστόρος) μπορεί να οριστεί σα το γινόμενο (άθροισμα) που μπορεί να γίνει 1 (0) μόνο για τις τιμές εισόδου μιας και μόνο γραμμής του πίνακα αλήθειας.

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

- Μπορούμε να ορίσουμε τη δυαδική τιμή που επαληθεύει τον όρο (κάνει 1 τον ελαχιστόρο ή 0 τον μεγιστόρο) σα το διακριτικό του αντίστοιχου όρου.
  - Π.χ. Ελαχιστόρος 4 = x y' z'. Μεγιστόρος 5 = x' + y + z'

Συνάρτηση σα κανονικό άθροισμα

- Αφού κάθε γραμμή του πίνακα αληθείας αντιστοιχεί σε ένα ελαχιστόρο μπορούμε κοιτώντας το πίνακα αληθείας να δούμε ποιοι ελαχιστόροι επαληθεύουν τη συνάρτηση.
- Το λογικό άθροισμα αυτών μας δίνει την έκφραση της συνάρτησης σα κανονικό άθροισμα.

x	y	z	F_3
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

- Η συνάρτηση επαληθεύεται όταν επαληθεύεται ο
  ελαχιστόρος 1 ή
  ο ελαχιστόρος 3 ή
  ο ελαχιστόρος 4 ή
  ο ελαχιστόρος 5

Ισοδύναμα είναι $F_3(x, y, z) = m_1 + m_3 + m_4 + m_5 =$
$x'y'z + x'yz + xy'z' + xy'z =$
$\Sigma(1, 3, 4, 5)$

Συνάρτηση σα κανονικό άθροισμα

- **Αφού ισοδύναμες συναρτήσεις έχουν κοινό πίνακα αλήθειας, θα έχουν και κοινά κανονικά αθροίσματα. Συνεπώς το κανονικό άθροισμα είναι μια πρότυπη μορφή πάνω στην οποία μπορώ να εφαρμόσω κάποιον αλγόριθμο.**

x	y	z	F_3
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

- **Η συνάρτηση επαληθεύεται όταν επαληθεύεται ο**

ο ελαχιστόρος 1 ή

ο ελαχιστόρος 3 ή

ο ελαχιστόρος 4 ή

ο ελαχιστόρος 5

Ισοδύναμα είναι $F_3(x, y, z) = m_1 + m_3 + m_4 + m_5 =$

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

$\Sigma(1, 3, 4, 5)$

Συνάρτηση σαν κανονικό γινόμενο

- **Αφού κάθε γραμμή του πίνακα αληθείας αντιστοιχεί σε ένα μεγιστόρο μπορούμε κοιτώντας το πίνακα αληθείας να δούμε ποιοί μεγιστόροι δεν επαληθεύουν τη συνάρτηση.**
- **Το λογικό γινόμενο αυτών μας δίνει την έκφραση της συνάρτησης σα κανονικό γινόμενο.**

x	y	z	F_3
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

- **Η συνάρτηση δεν επαληθεύεται αν επαληθεύεται (παίρνει δηλαδή τη τιμή 0)**

ο μεγιστόρος 0 ή

ο μεγιστόρος 2 ή

ο μεγιστόρος 6 ή

ο μεγιστόρος 7

Ισοδύναμα είναι $F'_3(x, y, z) = M'_0 + M'_2 + M'_6 + M'_7$ =>

$$F_3(x, y, z) = M_0 \bullet M_2 \bullet M_6 \bullet M_7 =$$

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

$$\Pi(0, 2, 6, 7)$$

Συνάρτηση σαν κανονικό γινόμενο

- **Αφού ισοδύναμες συναρτήσεις έχουν κοινό πίνακα αλήθειας, θα έχουν και κοινά κανονικά γινόμενα. Συνεπώς το κανονικό γινόμενο είναι μια πρότυπη μορφή πάνω στην οποία μπορώ να εφαρμόσω κάποιον αλγόριθμο.**

x	y	z	F_3
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

- **Η συνάρτηση δεν επαληθεύεται αν επαληθεύεται (παίρνει δηλαδή τη τιμή 0)**

μεγιστόρος 0 ή

μεγιστόρος 2 ή

μεγιστόρος 6 ή

μεγιστόρος 7

Ισοδύναμα είναι $F'_3(x, y, z) = M'_0 + M'_2 + M'_6 + M'_7 \Rightarrow$

$F_3(x, y, z) = M_0 \bullet M_2 \bullet M_6 \bullet M_7 =$

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

$\Pi(0, 2, 6, 7)$

Μετατροπές μεταξύ κανονικών μορφών

- **Αφού κάθε γραμμή του πίνακα αληθείας είναι είτε 0 είτε 1 αν γνωρίζω τη μορφή της συνάρτησης στη μία κανονική μορφή η άλλη προκύπτει από τους όρους που λείπουν.**

  Π.χ. $F(A, B, C) = \Sigma\,(1,4,6) \Rightarrow F = \Pi\,(0, 2, 3, 5, 7)$

  $G(W, X, Y, Z) = \Pi\,(1, 8, 11, 14, 15) \Rightarrow$

  $G = \Sigma\,(0, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13)$
- **Είναι επίσης πολύ εύκολο για μια συνάρτηση F να βρώ τη F'**
- **Ισχύει ότι $m'_i = M_i$ και φυσικά $M'_i = m_i$**
- **Για παράδειγμα είναι $m'0= (x'\ \ y'\ \ z')' = x + y + z = M0$**
- **Αρα αν $F(x, y, z) = \Sigma\,(1, 3,4) \Rightarrow F = m_1 + m_3 + m_4 \Rightarrow$**
- **$F' = (m_1 + m_3 + m_4)' = M_1 \bullet M_3 \bullet M_4 = \Pi(1, 3, 4) = \Sigma\,(0, 2, 5, 6, 7)$**
- **Δηλαδή η συμπληρωματική μιας συνάρτησης προκύπτει σα κανονικό άθροισμα των ελαχιστόρων που λείπουν από το κανονικό της άθροισμα**

Ελαχιστόροι από Αλγεβρική έκφραση

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

F (X, Y, Z) = X • Y' + X’ • Z • Y + X' • Y' • Z

Πρόβλημα 2→ Η σειρά είναι λάθος

Πρόβλημα 1 → Λείπει ο όρος Z

F (X, Y, Z) = X • Y’•(Z+Z’) + X’• Y• Z + X'• Y'• Z

F (X, Y, Z) = X•Y’•Z + X•Y’•Z’ + X’•Y• Z + X'• Y'• Z

101 100 011 001

5 4 3 1

x	y	z	F_3
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

Ισοδύναμα είναι F(x, y, z) = $m_1 + m_3 + m_4 + m_5$ =

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

Σ(1, 3, 4, 5)

Να θυμάστε

Πίνακας αλήθειας
Κανονικό άθροισμα
Κανονικό γινόμενο

} πρότυπες μορφές

Αλγεβρική μορφή
Λογικό διάγραμμα
Λεκτική περιγραφή

} μή πρότυπες μορφές

- **Πρέπει επίσης να μπορείτε να πηγαίνετε από μη πρότυπες μορφές σε πρότυπες.**

Απλοποίηση Συναρτήσεων

- Η πολυπλοκότητα του κυκλώματος που υλοποιεί μια συνάρτηση Boole σχετίζεται άμεσα με την πολυπλοκότητα της αλγεβρικής έκφρασης από την οποία η συνάρτηση υλοποιείται.
- Σκοποί της απλοποίησης
  - Λιγότεροι όροι
  - Απλούστεροι όροι
- Θέλουμε απλές και συστηματικές μεθόδους
- Υπάρχουν :
  - Η μέθοδος του χάρτη (μέθοδος Karnaugh / k-map) : γραφική μέθοδος για συναρτήσεις έως 5 μεταβλητών.
  - Η μέθοδος Quine-McClauskey : αλγεβρική μέθοδος
  - Η μέθοδος Espresso : αλγεβρική μέθοδος
- Οι μέθοδοι αυτοί δε μας δίνουν τις υλοποιήσεις με τις λιγότερες πύλες, αλλά τις απλούστερες υλοποιήσεις με NOT, AND & OR.