

# Εισαγωγή στην πληροφορική και τις εφαρμογές της

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΖΟΓΛΟΥ

Website: [microdev.gr](http://microdev.gr)



Επιστημονικές Εκδόσεις  
**ΤΖΙΟΛΑ**

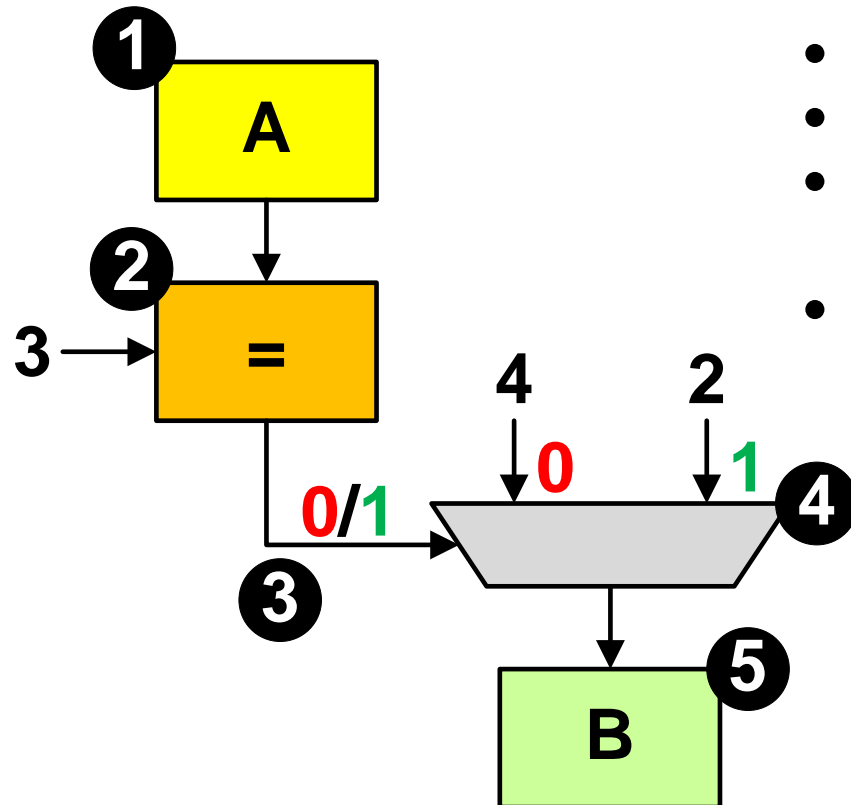
# ΚΕΦΑΛΑΙΟ 5

## Λογικά ψηφιακά κυκλώματα



# Η βασική ιδέα που κρύβεται πίσω από τη «λογική» της υλοποίησης Ψηφιακό κύκλωμα που «εκτελεί» μια συγκεκριμένη εντολή

IF A=3 THEN B=2 ELSE B=4



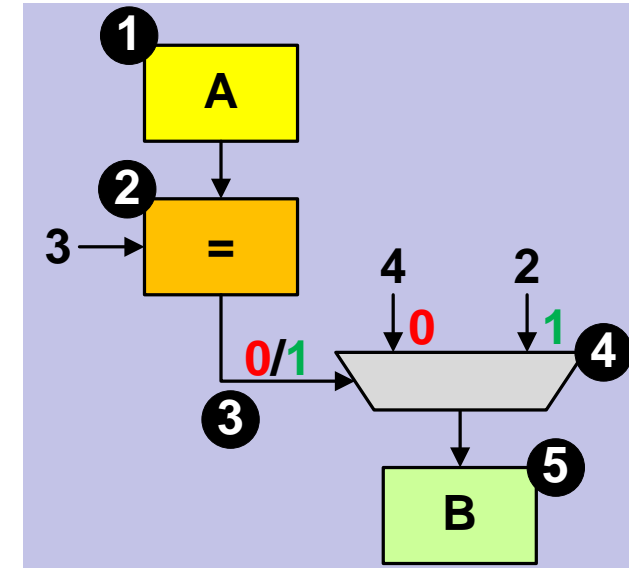
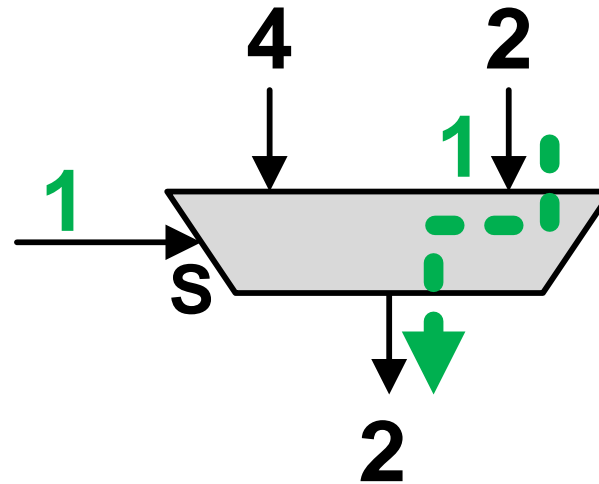
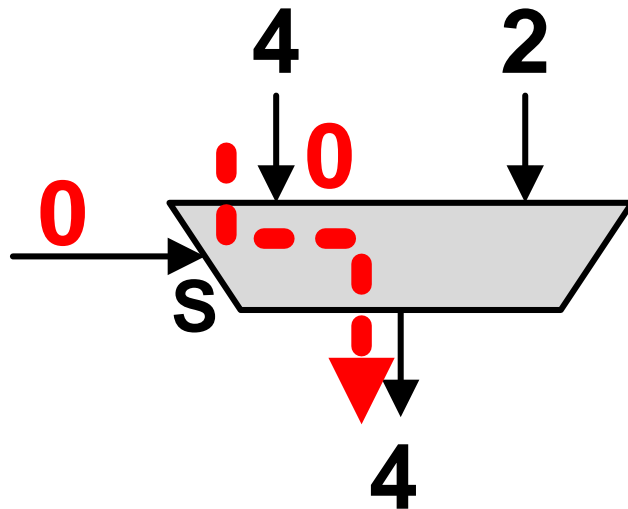
- αποθηκευμένο περιεχόμενο μεταβλητής A (συστατικό 1)
- μονάδα ελέγχου ισότητας (συστατικό 2)
- μονάδα ελέγχου της πηγής-τιμή (συστατικό 4) για ενημέρωση περιεχομένου της μεταβλητής B
- αποθηκευμένο περιεχόμενο μεταβλητής B (συστατικό 5)

# Λειτουργία μονάδας επιλογής (πολυπλέκτης)

Ψηφιακό κύκλωμα που «εκτελεί» μια συγκεκριμένη εντολή

$A \neq 3 \Rightarrow S=0, B=4$

$A = 3 \Rightarrow S=1, B=2$



# Λογικές πράξεις και εκφράσεις (1)

Αν  $P$  = λογική έκφραση

τότε  $P$  = αληθής (True) ή Ψευδής (False)

Δεχόμενου ότι το 0 αντιστοιχεί στο False και το 1 στο True, τότε  $P=0$  ή  $P=1$ .



## Λογικές πράξεις και εκφράσεις (2)

### Παράδειγμα λογικής έκφρασης

**P = έχω χρόνο ΚΑΙ καταστήματα ανοιχτά**

Αν αντιστοιχίσουμε τον όρο «έχω χρόνο» με το A και τον όρο «καταστήματα ανοιχτά» με το B, τότε το P θα γίνει

**P = A ΚΑΙ B**

Τα A και B με τη σειρά τους, θα είναι και αυτά True ή False, δηλαδή

**A=0 ή 1 και B=0 ή 1**

Όταν **A=0**, τότε ο όρος «έχω χρόνο» είναι False, πρακτικά δηλαδή, ΔΕΝ ΕΧΩ ΧΡΟΝΟ

Όταν **A=1**, τότε ο όρος «έχω χρόνο» είναι True, πρακτικά δηλαδή, ΕΧΩ ΧΡΟΝΟ

και

Όταν **B=0**, τότε ο όρος «καταστήματα ανοιχτά» είναι False, πρακτικά δηλαδή, ΤΑ ΚΑΤΑΣΤΗΜΑΤΑ ΔΕΝ ΕΙΝΑΙ ΑΝΟΙΧΤΑ

Όταν **B=1**, τότε ο όρος «καταστήματα ανοιχτά» είναι True, πρακτικά δηλαδή, ΤΑ ΚΑΤΑΣΤΗΜΑΤΑ ΕΙΝΑΙ ΑΝΟΙΧΤΑ

## Λογικές πράξεις και εκφράσεις (3)

Ποιοι είναι οι δυνατοί συνδυασμοί των A, B και ποια η αντίστοιχη τιμή του P;

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

### Λογικό ΚΑΙ

( $P = A \text{ ΚΑΙ } B$ )

είναι True μόνο αν και οι δύο όροι είναι True

# Λογικές πράξεις και εκφράσεις (4)

## Στοιχειώδεις λογικές πράξεις

- Λογικό NOT
- Λογικό AND
- Λογικό OR

### Παράδειγμα λογικού NOT

NOT A  $\rightarrow$  αντίστροφη κατάσταση του A  
Αν A=0, τότε NOT A=1 και αν A=1, τότε NOT A=0.

$P = \text{NOT } A \rightarrow$  [άλγεβρα Boole]  $\rightarrow P = \bar{A}$  ή  $P = \bar{A}$

### Παράδειγμα λογικού AND

A AND B  $\rightarrow$  [άλγεβρα Boole]  $\rightarrow P = A \cdot B$

### Παράδειγμα λογικού OR

$P = A \text{ OR } B \rightarrow$  [άλγεβρα Boole]  $\rightarrow$  γράφεται  $P = A + B$

A	B	P=A+B
0	0	0
0	1	1
1	0	1
1	1	1





# Εφαρμογή (1)

## Απόδοση υπολογιστή

Λογική πρόταση 1 (Π1): η απόδοση του υπολογιστή είναι ικανοποιητική μόνο αν ο υπολογιστής **λειτουργεί 5 ώρες** **ΚΑΙ** η θερμοκρασία του μικροεπεξεργαστή **είναι 43 βαθμοί κελσίου**

**Ψευδής** (περίπτωση 1): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί κελσίου

**Ψευδής** περίπτωση 2): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί κελσίου

**Ψευδής** (περίπτωση 3): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί κελσίου

**Αληθής** (περίπτωση 4): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί κελσίου



## Εφαρμογή (2)

### Απόδοση υπολογιστή

Λογική πρόταση 1 (Π1): η απόδοση του υπολογιστή είναι ικανοποιητική μόνο αν ο υπολογιστής **λειτουργεί 5 ώρες** **ΚΑΙ** η θερμοκρασία του μικροεπεξεργαστή **είναι 43 βαθμοί** κελσίου

Λογικές προτάσεις και στάθμες	
Λογική πρόταση	Λογική στάθμη (κατάσταση)
Ο υπολογιστής ΔΕΝ λειτουργεί 5 ώρες	0
Ο υπολογιστής λειτουργεί 5 ώρες	1
Η θερμοκρασία ΔΕΝ είναι 43 βαθμοί	0
Η θερμοκρασία είναι 43 βαθμοί	1

Λογικές καταστάσεις για την πρόταση 1 (Π1)			
A	B	Πρόταση Π1	Χαρακτηρισμός
0	0	0	Ψευδής
0	1	0	Ψευδής
1	0	0	Ψευδής
1	1	1	Αληθής



## Εφαρμογή (3)

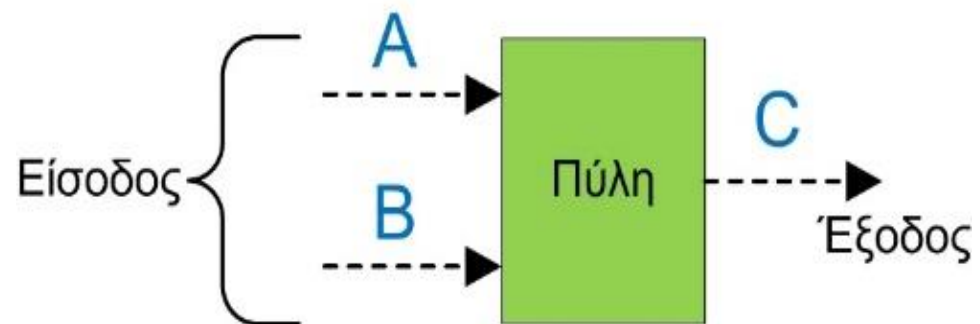
Μελετήστε την εφαρμογή 5.2 του βιβλίου

(Ροή αέρα στον υπολογιστή)



# Λογικές πύλες (1)

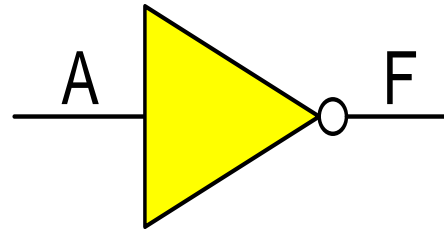
## Υλοποίηση λογικών πράξεων



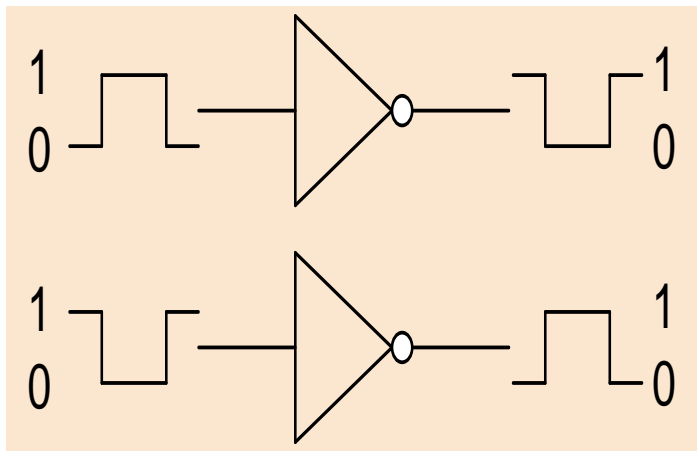
Γενικό διάγραμμα πύλης

## Λογικές πύλες (2)

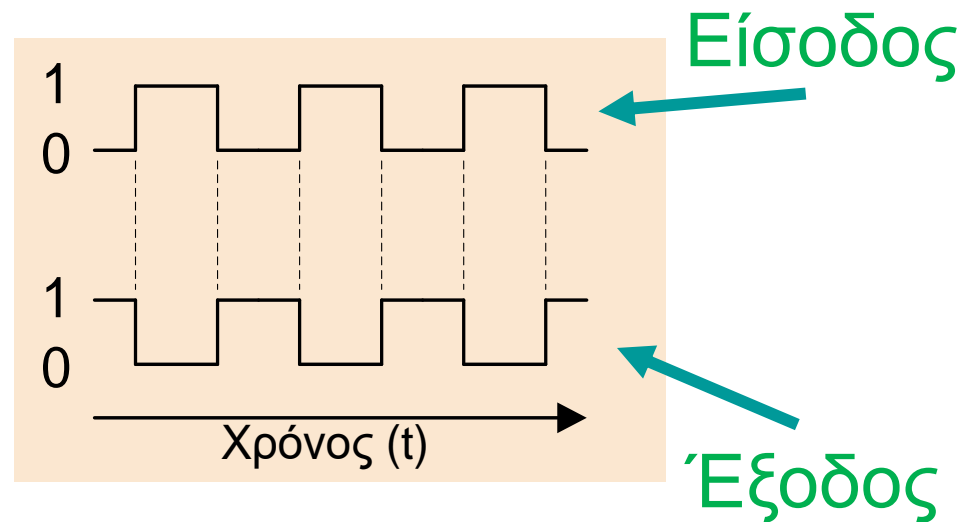
### Η πύλη NOT – Αντιστροφέας



Πίνακας αληθείας της πύλης NOT	
Είσοδος (A)	Έξοδος (F)
0	1
1	0

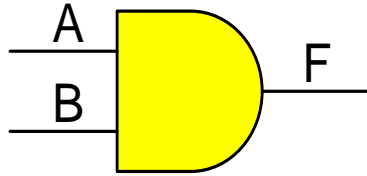


Λειτουργία

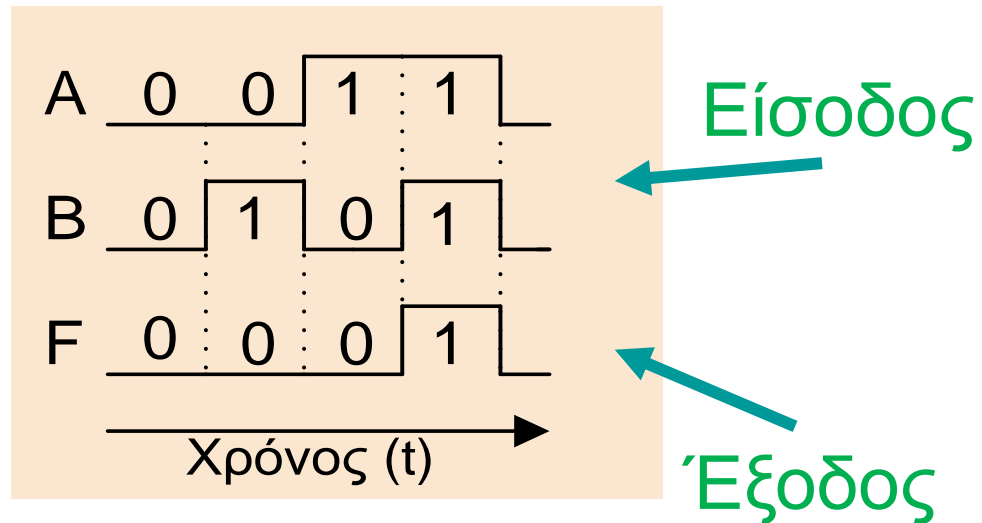


# Λογικές πύλες (3)

## Η πύλη AND

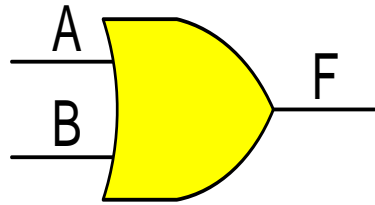


Πίνακας αληθείας πύλης AND		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

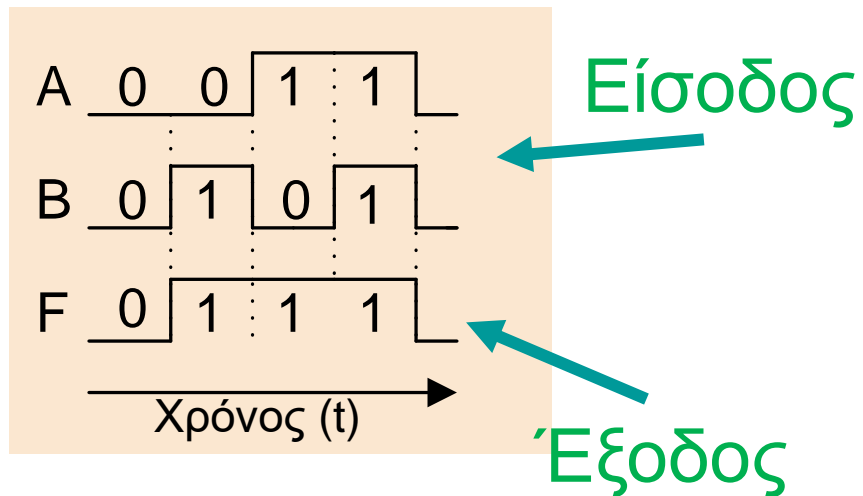


# Λογικές πύλες (4)

## Η πύλη OR

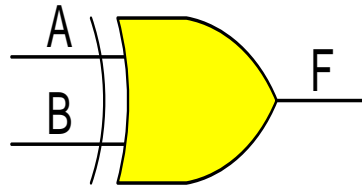


Πίνακας αληθείας πύλης OR		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

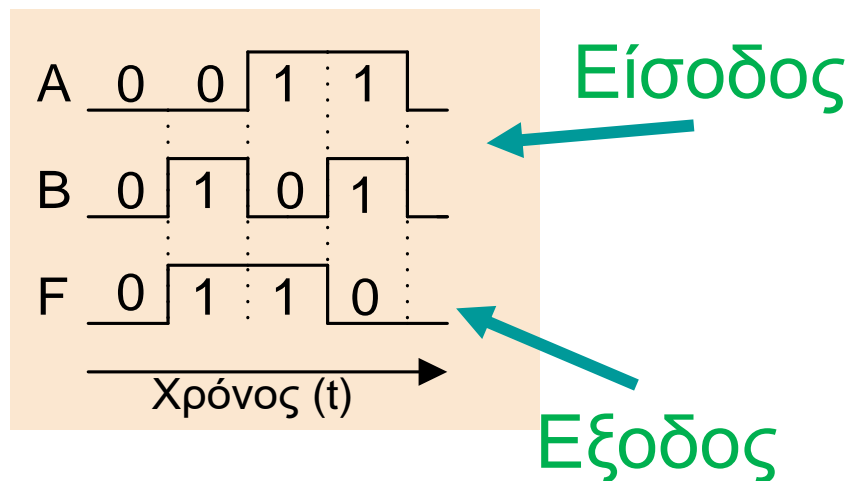


# Σύνθεση νέων πυλών

## Η πύλη XOR



Πίνακας αληθείας πύλης XOR		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

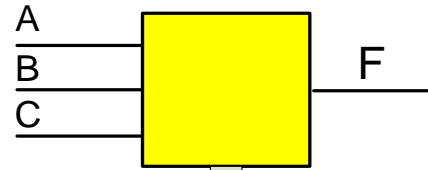




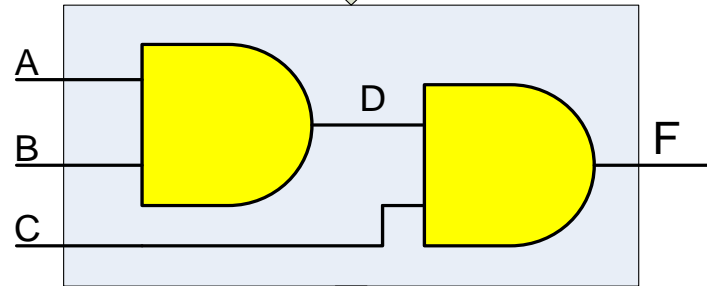
# Πύλες με περισσότερες εισόδους

## Σύνθεση μιας πύλης AND τριών εισόδων

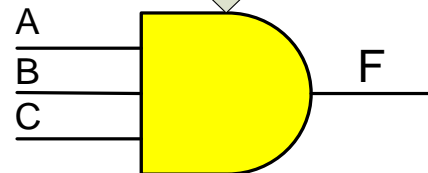
Βήμα 1



Βήμα 2



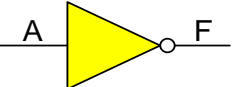
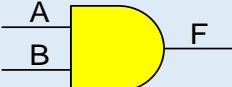
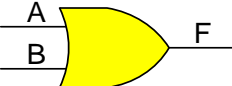
Βήμα 3





# Άλγεβρα Boole

## Συμβολισμός με την άλγεβρα Boole

Συμβολισμός με την άλγεβρα Boole		
Λογική πύλη	Περιγραφή	Έκφραση άλγεβρας Boole
	$F = \text{NOT } A$	$F = \bar{A}$
	$F = A \text{ AND } B$	$F = A \cdot B$ ή $F = AB$
	$F = A \text{ OR } B$	$F = A + B$



# Άλγεβρα Boole

## Κανόνες και ιδιότητες της άλγεβρας Boole

- **Αντιμεταθετικότητα.** AND  $\rightarrow AB = BA$ , OR  $\rightarrow A+B = B+A$
- **Προσεταιριστικότητα.** AND  $\rightarrow (AB)C = A(BC)$ , OR  $\rightarrow (A+B)+C=A+(B+C)$
- **Επιμεριστικότητα.**  $A(B+C) = (AB)+(AC)$  και  $A+(BC) = (A+B) (A+C)$
- **Απορροφητικό στοιχείο.**  $(A+B)A = A$  και  $(AB)+A=A$
- **Συμπλήρωμα.**  $AA' = 0$  και  $A+A' = 1$
- **Ουδέτερο στοιχείο.**  $A+0=A$ ,  $A1=A$  (ή  $A \cdot 1=A$ ).



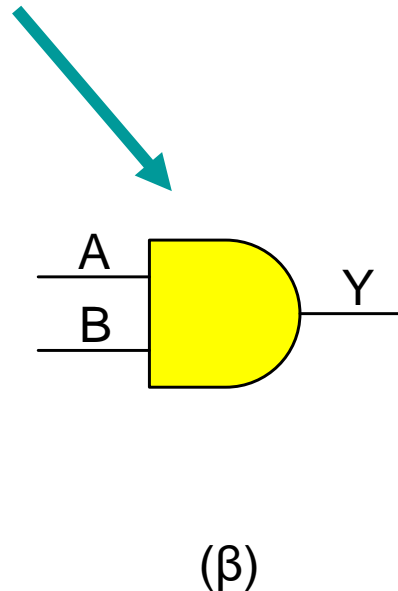
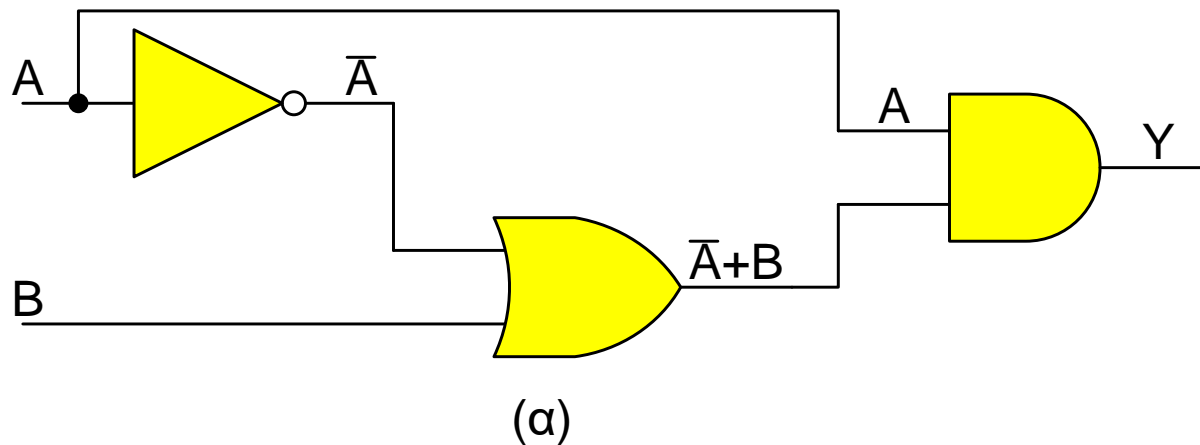
# Άλγεβρα Boole

## Απλοποίηση με την άλγεβρα Boole

$$Y = A(\bar{A} + B)$$

### Απλοποίηση

$$Y = A(\bar{A} + B) = A\bar{A} + AB = 0 + AB = AB$$

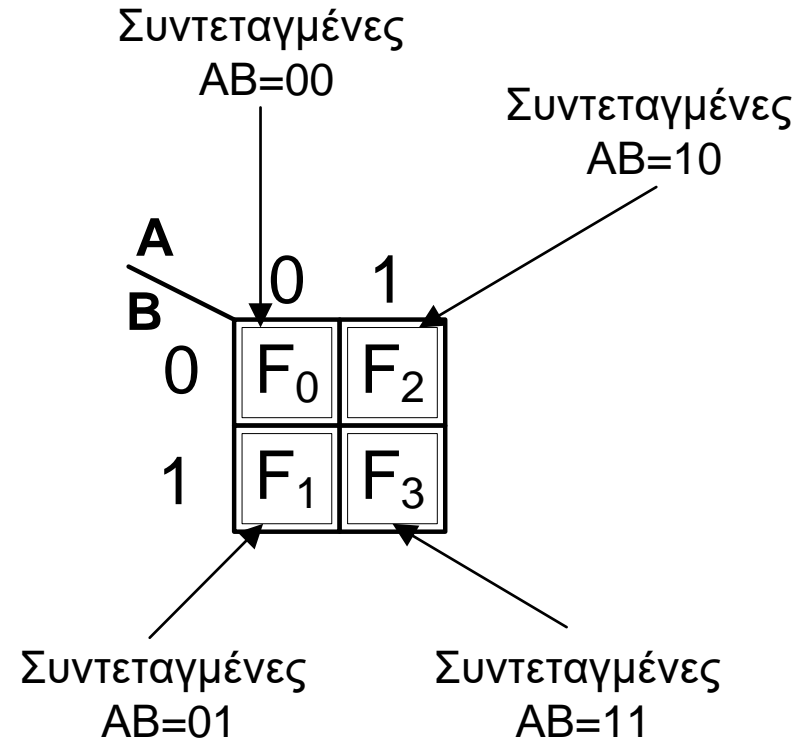
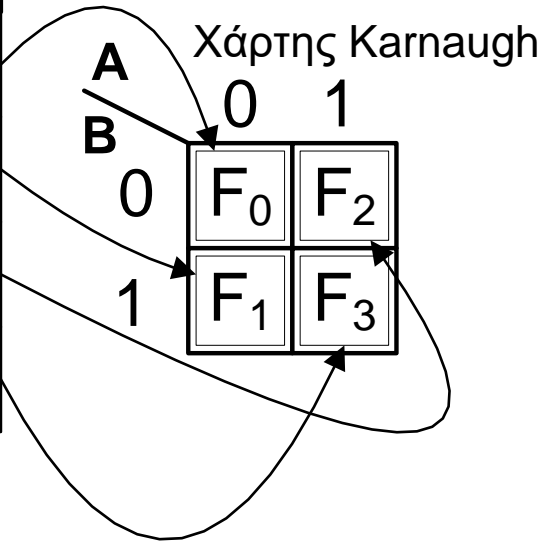


# Άλγεβρα Boole

## Απλοποίηση με χαρτογράφηση (1)

Πίνακας αληθείας

A	B	F
0	0	F <sub>0</sub>
0	1	F <sub>1</sub>
1	0	F <sub>2</sub>
1	1	F <sub>3</sub>



# Άλγεβρα Boole

## Απλοποίηση με χαρτογράφηση (2)

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

$$F = \bar{A}B\bar{C}D$$

$$F = \bar{A}BCD$$

$$F = AB\bar{C}D$$

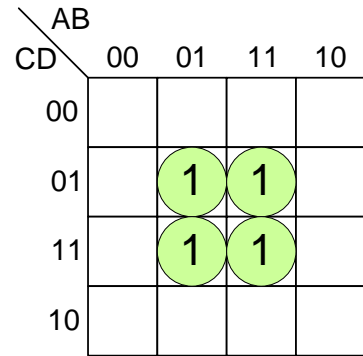
$$F = ABCD$$

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

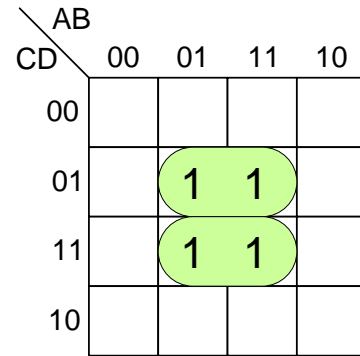


# Άλγεβρα Boole

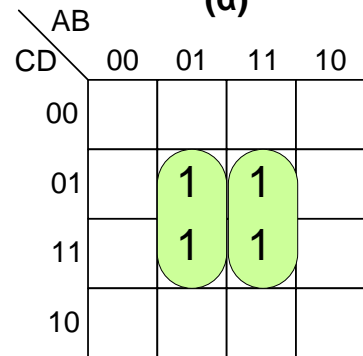
## Απλοποίηση με χαρτογράφηση (3) – τρόποι ομαδοποίησης



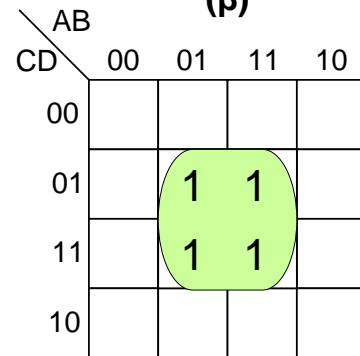
(α)



(β)



(γ)



(δ)

### Αποτελεσματικότητα ομαδοποίησης στο χάρτη Karnaugh

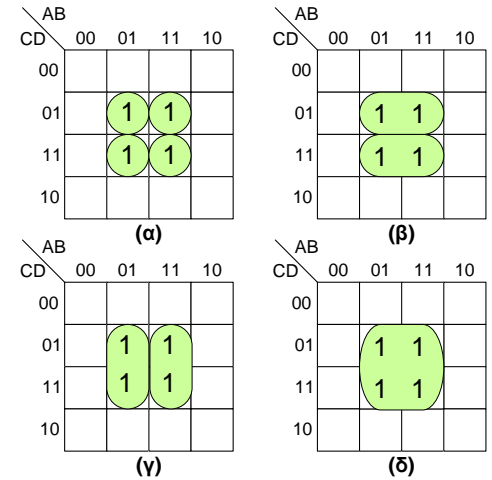
Αρχική έκφραση  $F = \bar{A}\bar{B}\bar{C}D + \bar{A}BCD + A\bar{B}\bar{C}D + ABCD$

Περίπτωση	Πλήθος ομάδων	Απλοποιημένη έκφραση
(α)	4	$F = \bar{A}\bar{B}\bar{C}D + \bar{A}BCD + A\bar{B}\bar{C}D + ABCD$
(β)	2	$F = B\bar{C}D + BCD$
(γ)	2	$F = \bar{A}BD + ABD$
(δ)	1	$F = BD$

# Άλγεβρα Boole

## Απλοποίηση με χαρτογράφηση (4) – μερικοί κανόνες

- Ομαδοποιούμε όσους περισσότερους άσσους μπορούμε
- Όσο μεγαλύτερες και λιγότερες ομάδες έχουμε, τόσο πιο απλή θα είναι η τελική έκφραση
- Στις συντεταγμένες από το 01 πάμε στο 11 και μετά στο 10 (μόνο ένα ψηφίο επιτρέπεται να αλλάζει)
- Το πλήθος των άσπων στην ομάδα πρέπει να είναι δύναμη του 2
- Από κάθε ομάδα προκύπτει μια λογική έκφραση
- Ο συνδυασμός των εκφράσεων γίνεται με την πράξη OR
- Σε κάθε ομάδα κρατάμε μόνο τις τιμές που η αντίστοιχη συντεταγμένη παραμένει σταθερή





# Άλγεβρα Boole

## Πλήρες παράδειγμα απλοποίησης

### Βήμα 1 - Σχεδίαση πίνακα αληθείας

Είσοδοι			Έξοδος (F)
A	B	C	
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

### Βήμα 4 - Ομαδοποίηση

	AB			
C	00	01	11	10
0	1	1	1	1
1			1	

### Βήμα 2 – Λογική έκφραση

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

### Βήμα 3 – Πίνακας Karnaugh

	AB			
C	00	01	11	10
0	1	1	1	1
1			1	

### Βήμα 5 – Εξαγωγή λογικής έκφρασης

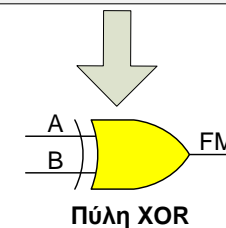
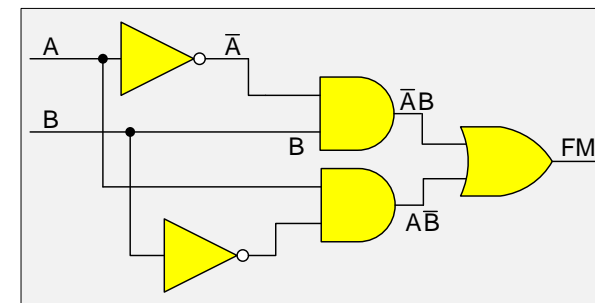
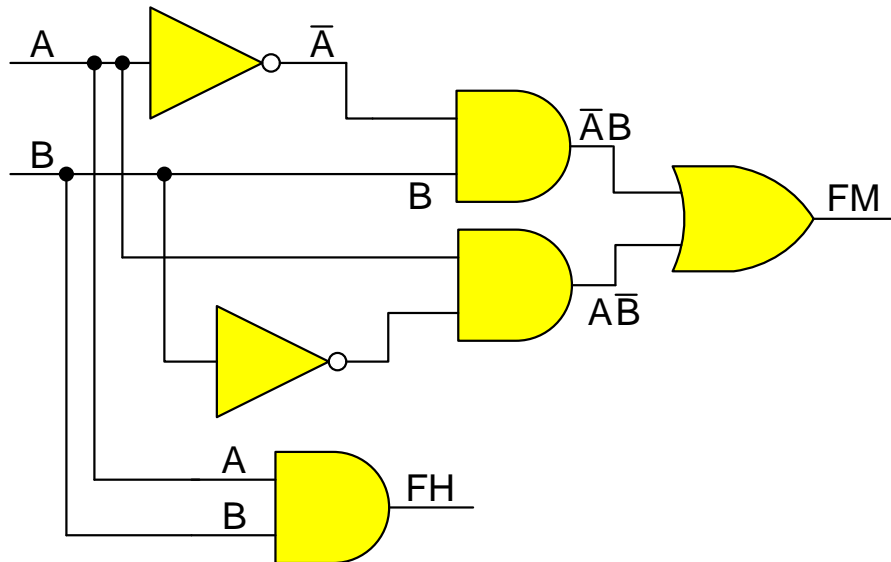
$$F = \bar{C} + ABC$$



# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

## Κύκλωμα ελέγχου ροής αέρα στον υπολογιστή (1)

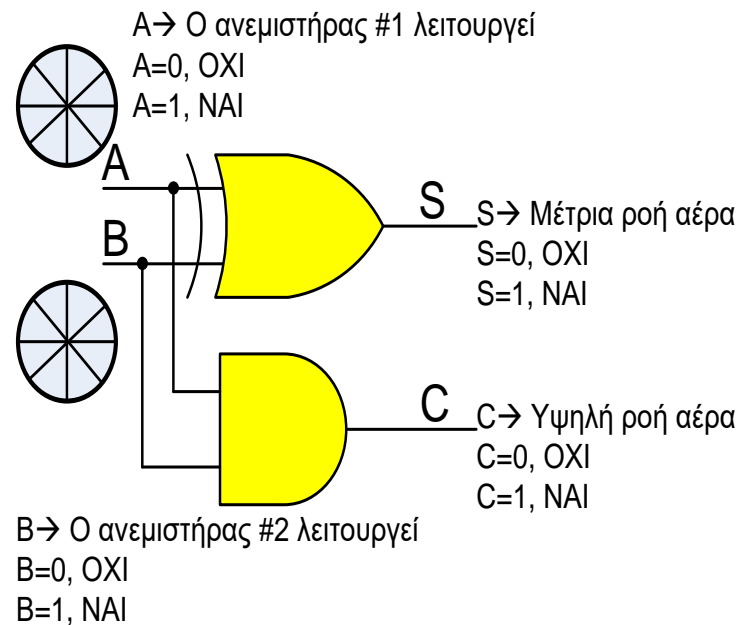
Κατάσταση ροής αέρα			
Είσοδος		Έξοδος	
Κατάσταση ανεμιστήρα #1 (A)	Κατάσταση ανεμιστήρα #2 (B)	Μέτρια ροή αέρα (FM)	Υψηλή ροή αέρα (FH)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

## Κύκλωμα ελέγχου ροής αέρα στον υπολογιστή (2)

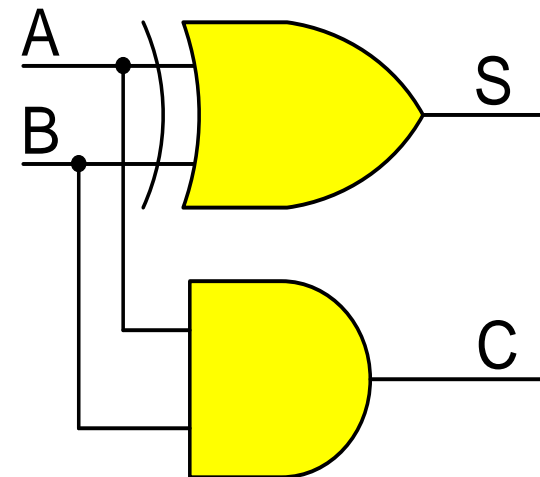
Κατάσταση ροής αέρα			
Είσοδος		Έξοδος	
Κατάσταση ανεμιστήρα #1 (A)	Κατάσταση ανεμιστήρα #2 (B)	Μέτρια ροή αέρα (FM)	Υψηλή ροή αέρα (FH)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

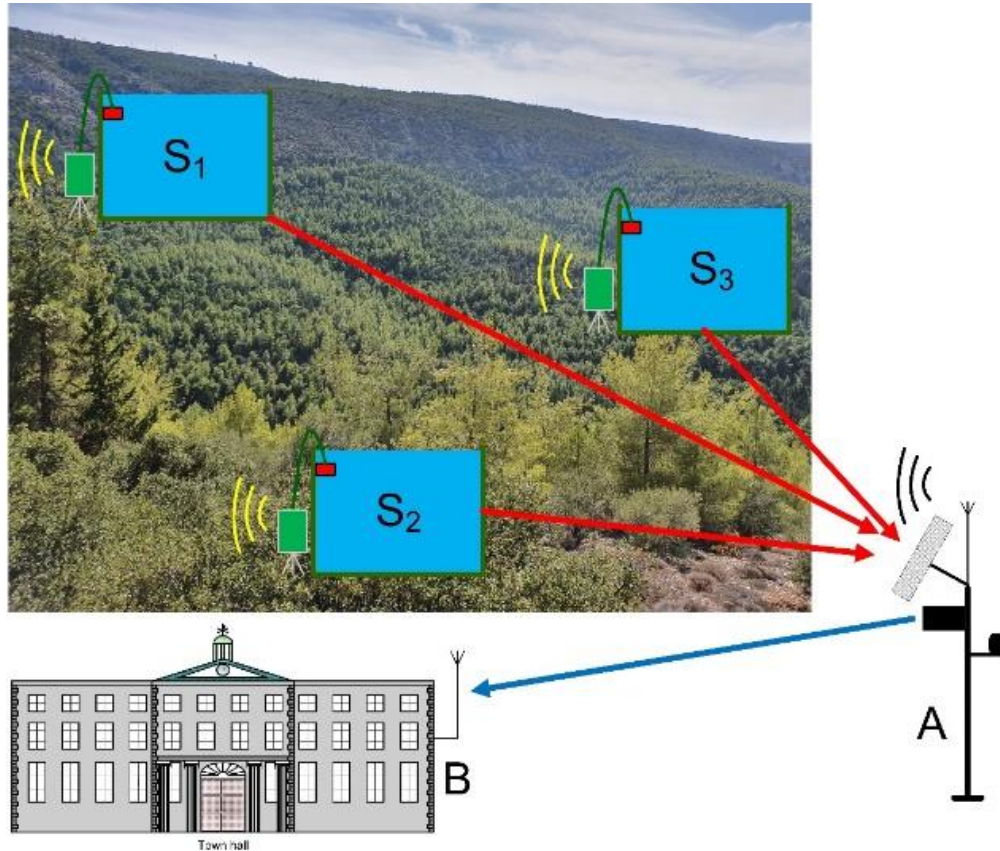
## Κύκλωμα αριθμητικής πρόσθεσης (ημιαθροιστής)

Μας ενδιαφέρει πότε η έξοδος είναι 1			
Είσοδος		Έξοδος	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

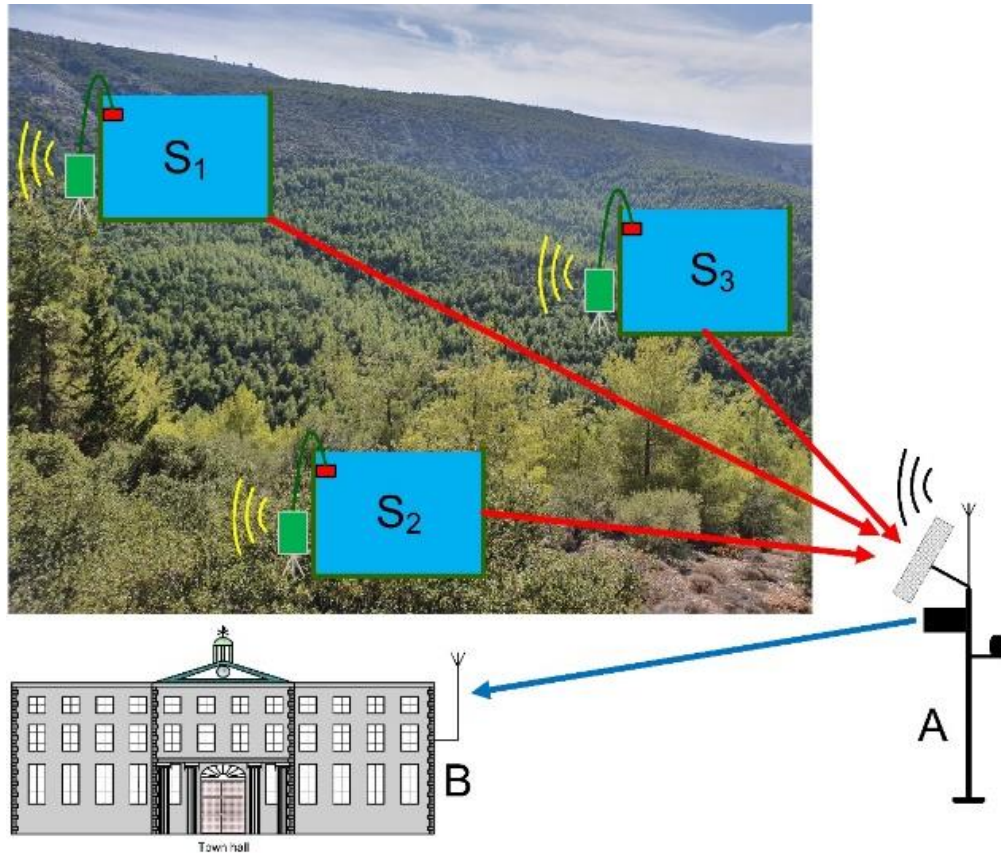
## Έλεγχος στάθμης σε δεξαμενές νερού (1)



- Δάσος του Υμηττού → τρεις δεξαμενές νερού
- Κάθε δεξαμενή → διαθέτει ασύρματο αισθητήρα στάθμης νερού
- Ενδείξεις (ενεργοποίηση)  
LOW1 : έχει αδειάσει μόνο μια δεξαμενή  
LOW2 : έχουν αδειάσει δύο δεξαμενές  
EMPTY : έχουν αδειάσει όλες οι δεξαμενές
- Μετρήσεις δεξαμενών  
Αισθητήρες στάθμης νερού (S1, S2, S3)

# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

## Έλεγχος στάθμης σε δεξαμενές νερού (2)



Έλεγχος στάθμης νερού					
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	LOW1	LOW2	EMPTY
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	0



# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

## Έλεγχος στάθμης σε δεξαμενές νερού (3)

Έλεγχος στάθμης νερού					
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	LOW1	LOW2	EMPTY
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	0

$$LOW1 = \bar{S}_1 S_2 S_3 + S_1 \bar{S}_2 S_3 + S_1 S_2 \bar{S}_3$$

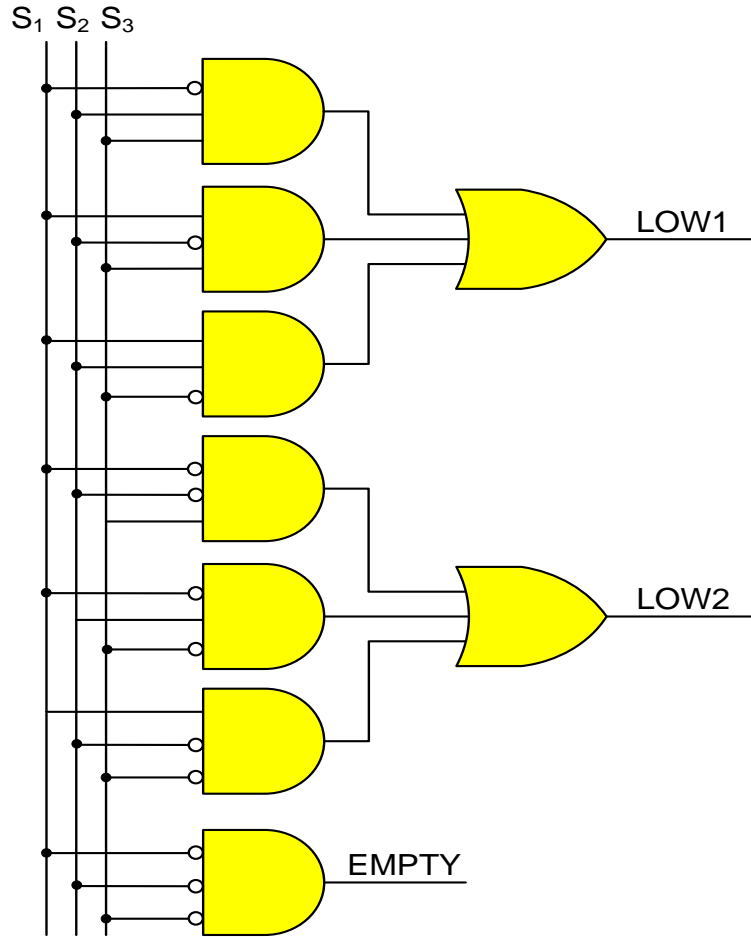
$$LOW2 = \bar{S}_1 \bar{S}_2 S_3 + \bar{S}_1 S_2 \bar{S}_3 + S_1 \bar{S}_2 \bar{S}_3$$

$$EMPTY = \bar{S}_1 \bar{S}_2 \bar{S}_3$$

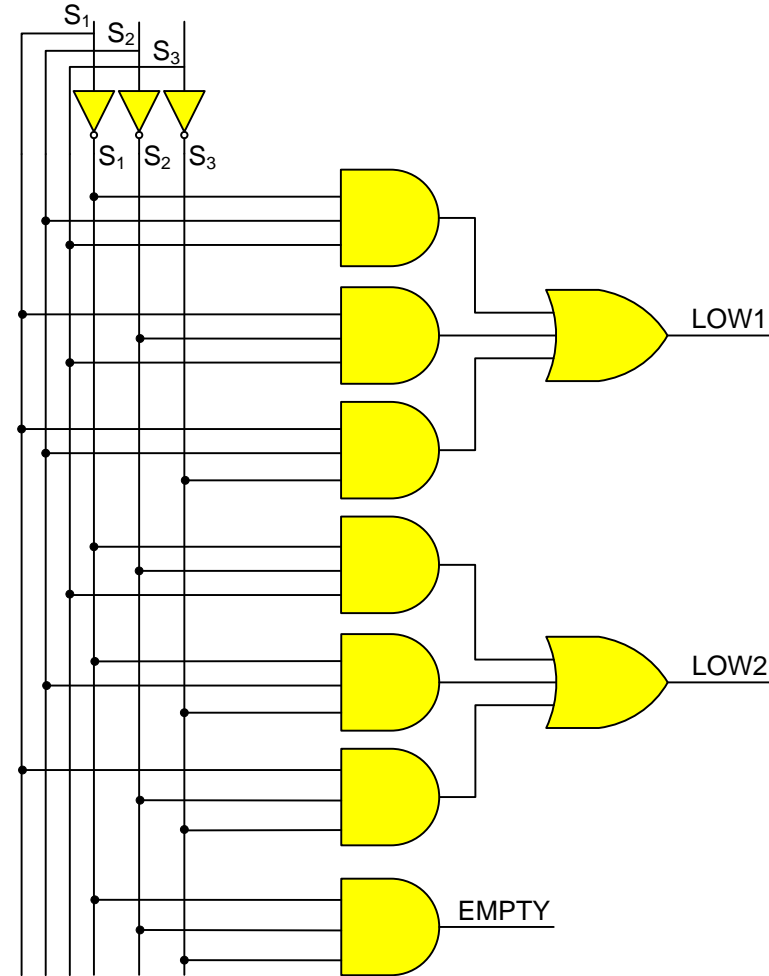


# Εφαρμογή – Σχεδίαση ψηφιακού κυκλώματος

## Έλεγχος στάθμης σε δεξαμενές νερού (4)



Αρχικό κύκλωμα



Βελτιωμένο κύκλωμα

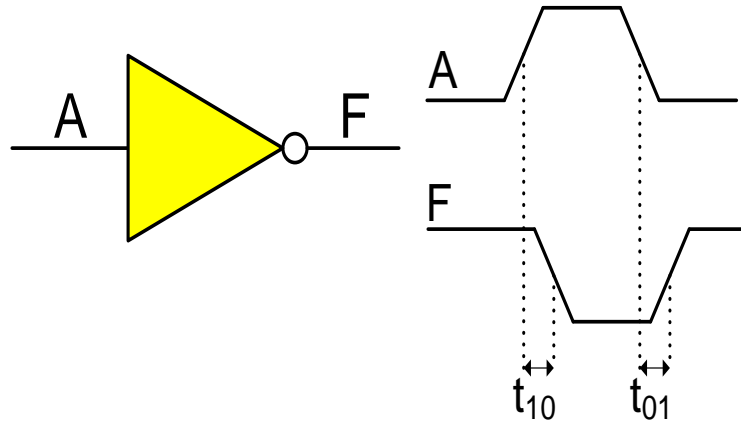




# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Γενικά χαρακτηριστικά

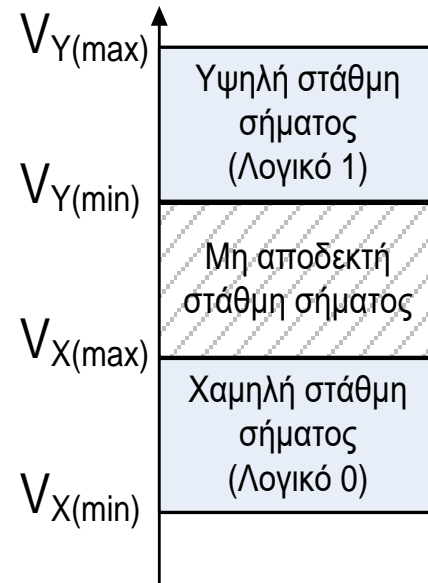
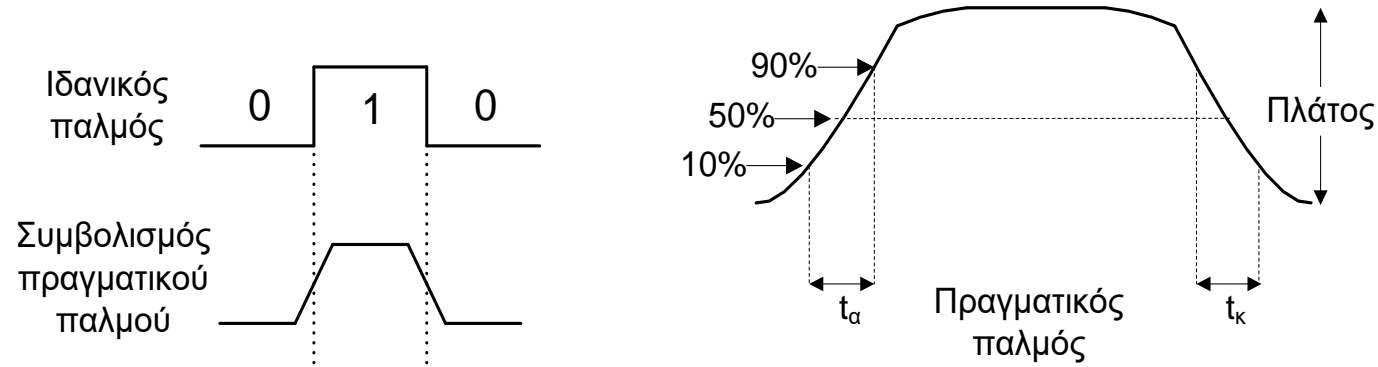
(α) χρονική απόκριση & καθυστέρηση διάδοσης



(β) κατανάλωση ενέργειας και δυνατότητα παροχής ρεύματος σε άλλες συνδεδεμένες διατάξεις

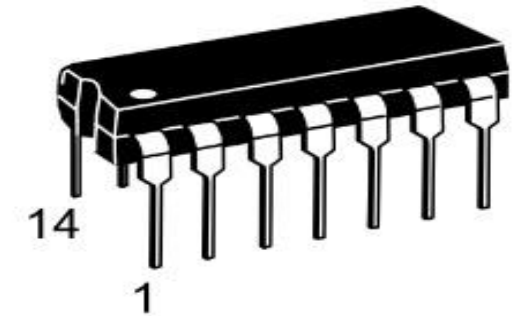
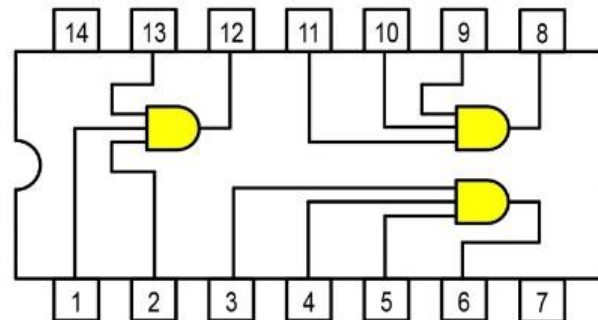
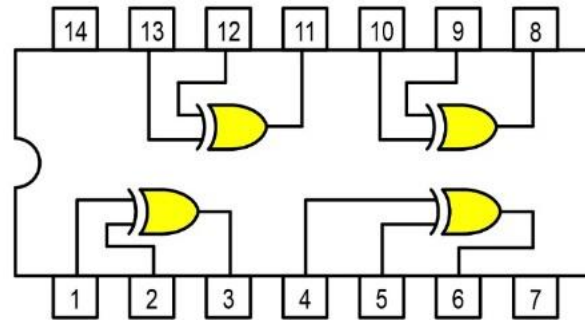
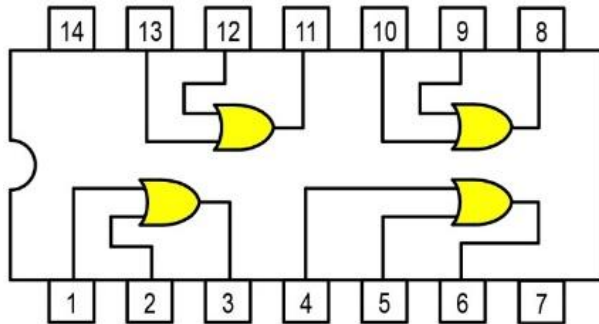
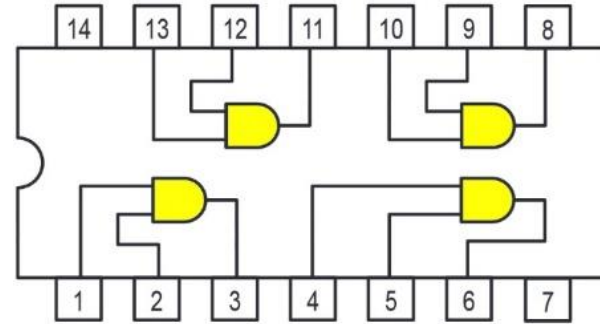
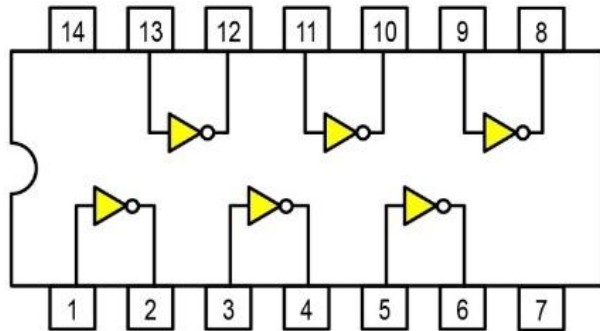
# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Τα ψηφιακά σήματα στην πράξη



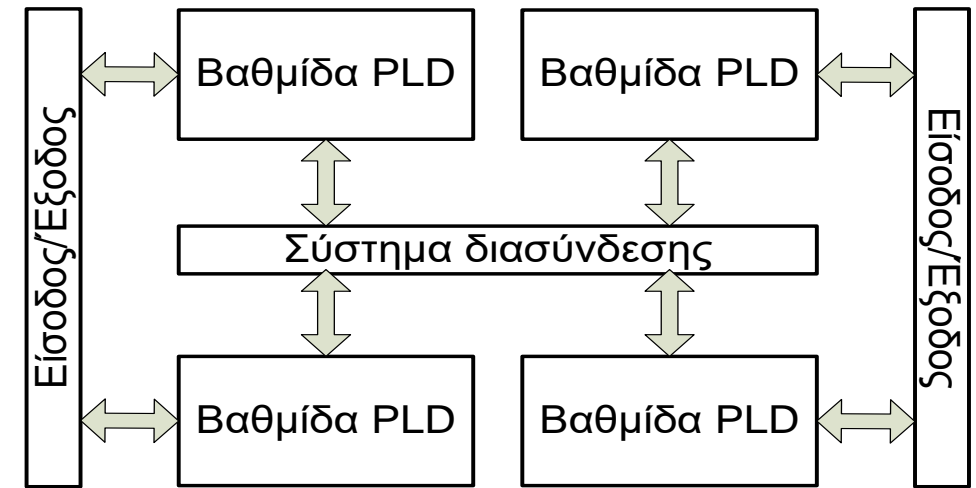
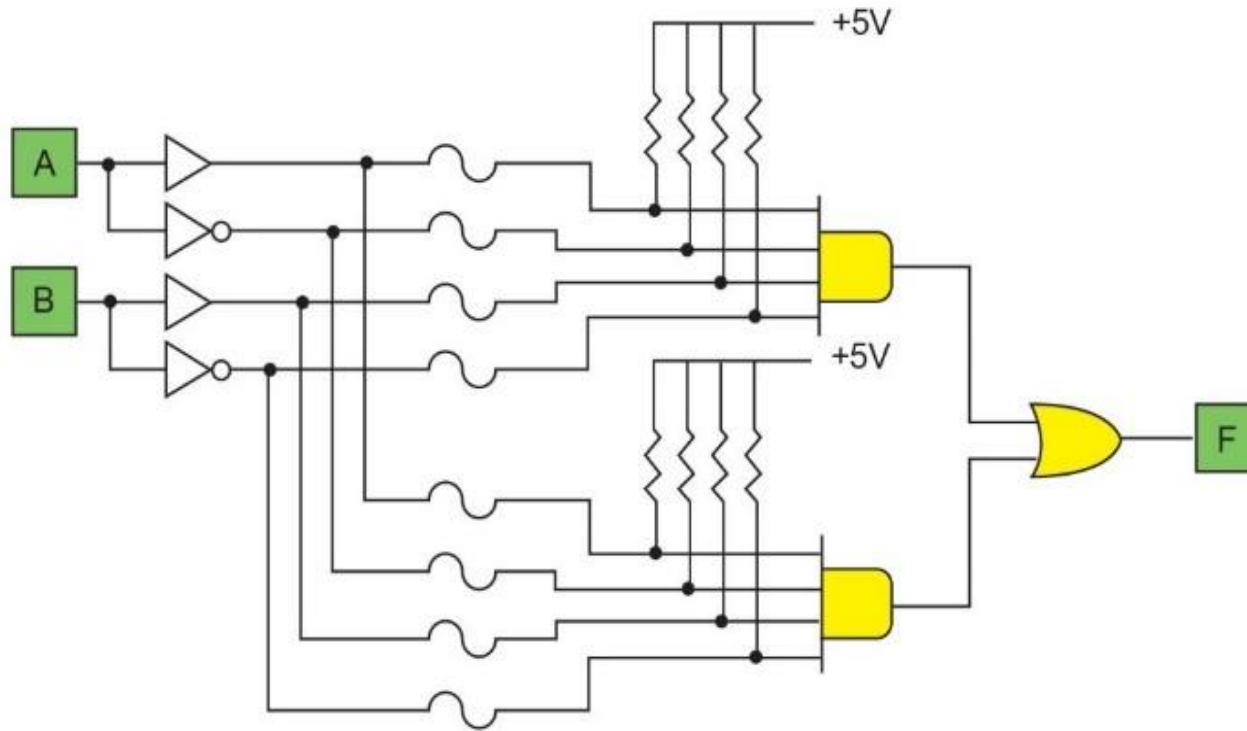
# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Ψηφιακά ολοκληρωμένα κυκλώματα (TTL)



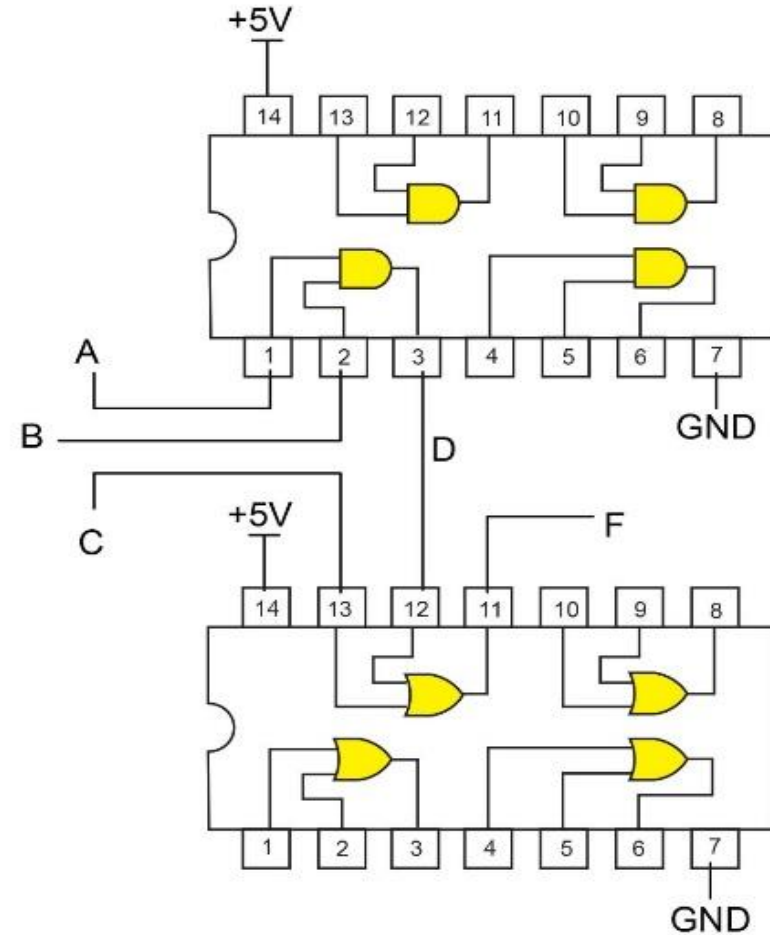
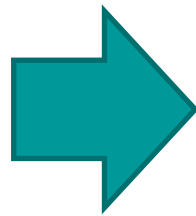
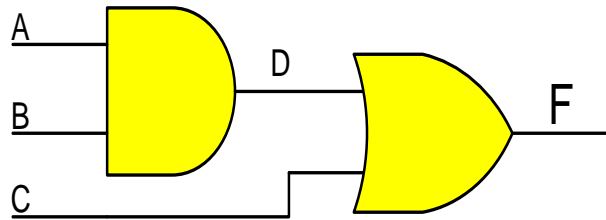
# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Ψηφιακά ολοκληρωμένα κυκλώματα (PLD-CPLD)



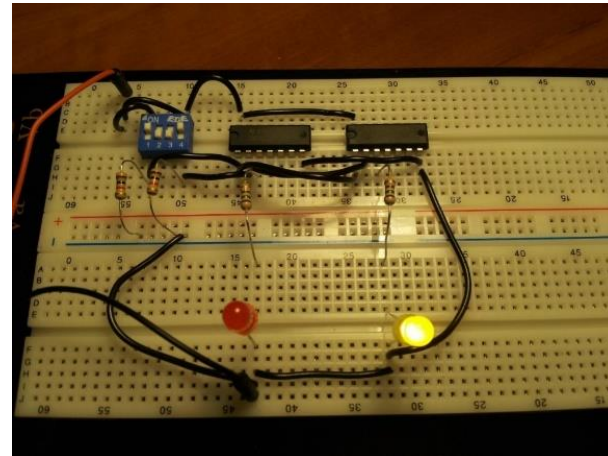
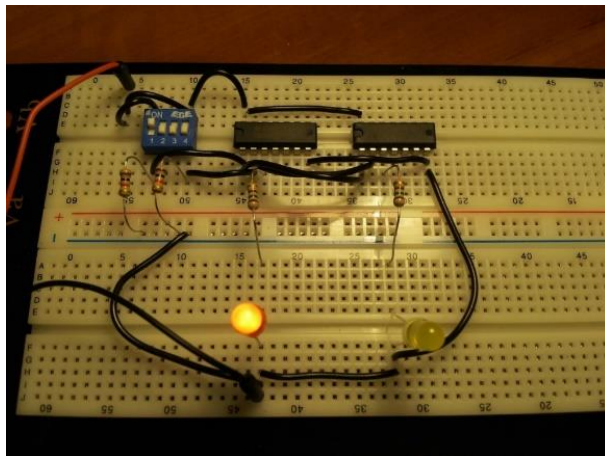
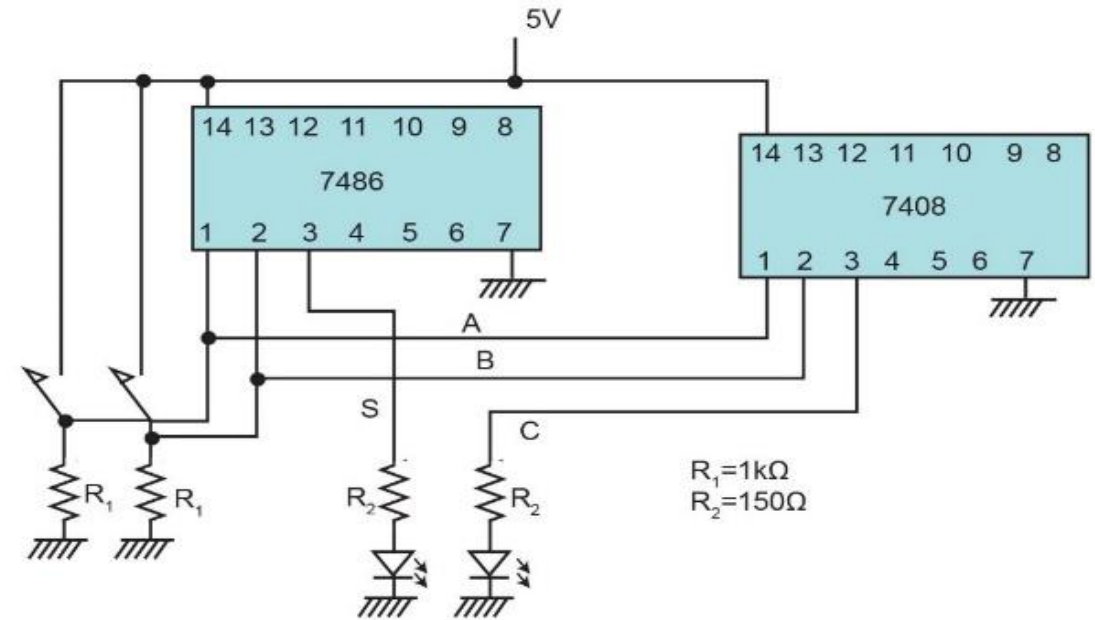
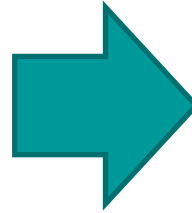
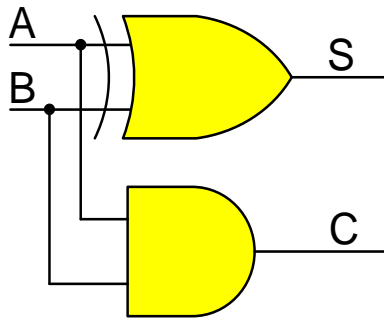
# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Υλοποίηση ψηφιακών κυκλωμάτων με ολοκληρωμένα τύπου TTL (1)



# Υλοποίηση πραγματικών ψηφιακών κυκλωμάτων

## Υλοποίηση ψηφιακών κυκλωμάτων με ολοκληρωμένα τύπου TTL (2)



# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα;

## Προηγμένη σχεδίαση ψηφιακών κυκλωμάτων

### Γλώσσες περιγραφής υλικού

Γνωστές γλώσσες τύπου HDL	
Γλώσσα τύπου HDL	Περιγραφή
<b>AHDL</b>	Γλώσσα κατασκευασμένη από την Altera
<b>JHDL</b> (Just-Another Hardware Description Language)	Γλώσσα για δημιουργία κυκλωμάτων με αντικειμενοστραφή προσέγγιση και συνεργασία με τη Java
<b>MyHDL</b>	Γλώσσα ανοιχτού κώδικα για τη μετάβαση από τη γνωστή Python σε επίπεδο υλικού
<b>Verilog HDL</b>	Χρησιμοποιεί προσέγγιση από διαφορετικό επίπεδο αφαίρεσης σε σχέση με τη VHDL. Επίσης, υποστηρίζει και αναλογικά κυκλώματα
<b>VHDL</b> (Very high speed integrated circuits-Hardware Description Language)	Χρησιμοποιείται στον προγραμματισμό ψηφιακών κυκλωμάτων και έχει τυποποιηθεί από τον οργανισμό IEEE



# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα;

## Προηγμένη σχεδίαση ψηφιακών κυκλωμάτων

### Λογισμικό ανάπτυξης

- Γραφικός σχεδιασμός
- Σχεδιασμός με προγραμματισμό
- Σχεδιασμός υψηλού επιπέδου
- Σχεδιασμός κυματομορφών
- Παρεμβάσεις σε επίπεδο υλοποίησης
- Προσομοιωτής
- Αναλυτής χρόνου





# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα;

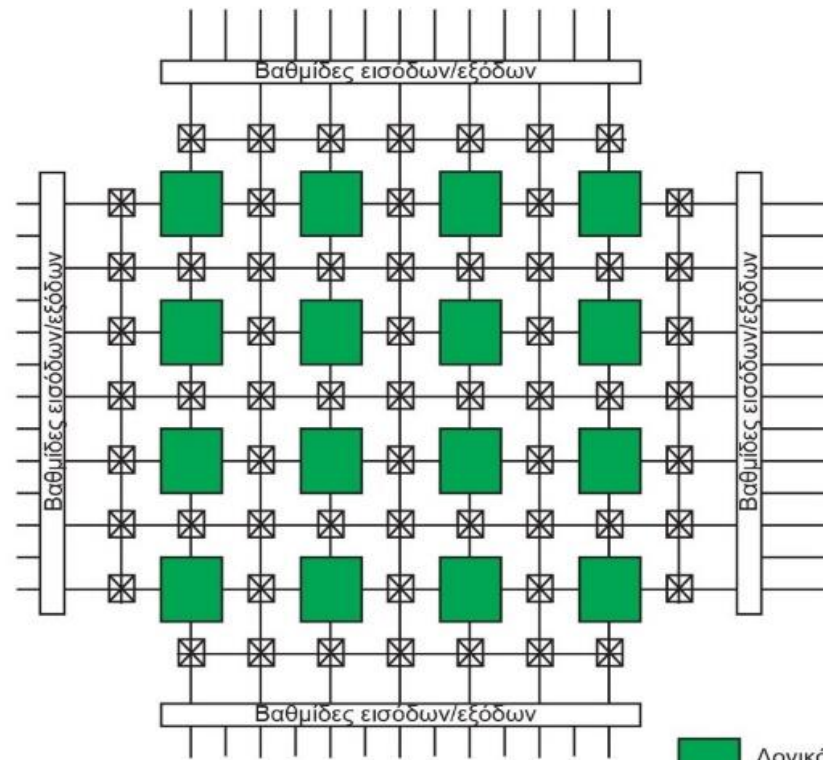
## Επαναπρογραμματιζόμενες ψηφιακές διατάξεις (FPGA) (1)



### Field Programmable Gate Arrays

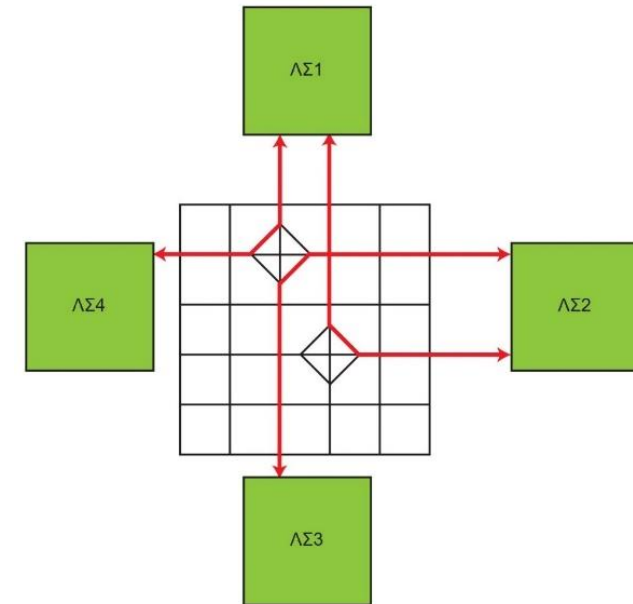
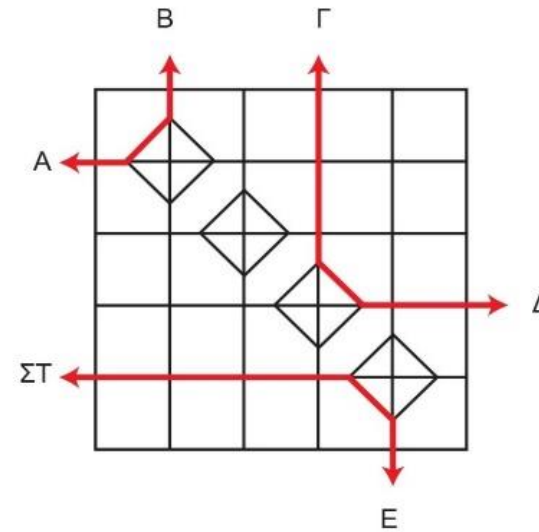
- Παρουσιάζουν **υψηλές επιδόσεις** όταν χρησιμοποιούνται για την υλοποίηση συγκεκριμένης εφαρμογής (βέλτιστο κύκλωμα)
- Προσφέρουν δυνατότητα **παράλληλης υλοποίησης** αλγορίθμων και κυκλωμάτων απευθείας στο υλικό
- Παρουσιάζουν **προσαρμοστικότητα σε πραγματικό χρόνο**, με τα στοιχεία τους να μπορούν να επαναπρογραμματιστούν κατά τη διάρκεια της λειτουργίας του κυκλώματος (εκτέλεση προσαρμοσμένων λειτουργιών)
- Συνδυάζουν σχετικά **χαμηλό κόστος** με τον **μικρό όγκο** (της διάταξης)



# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα; Επαναπρογραμματιζόμενες ψηφιακές διατάξεις (FPGA) (2) Field Programmable Gate Arrays

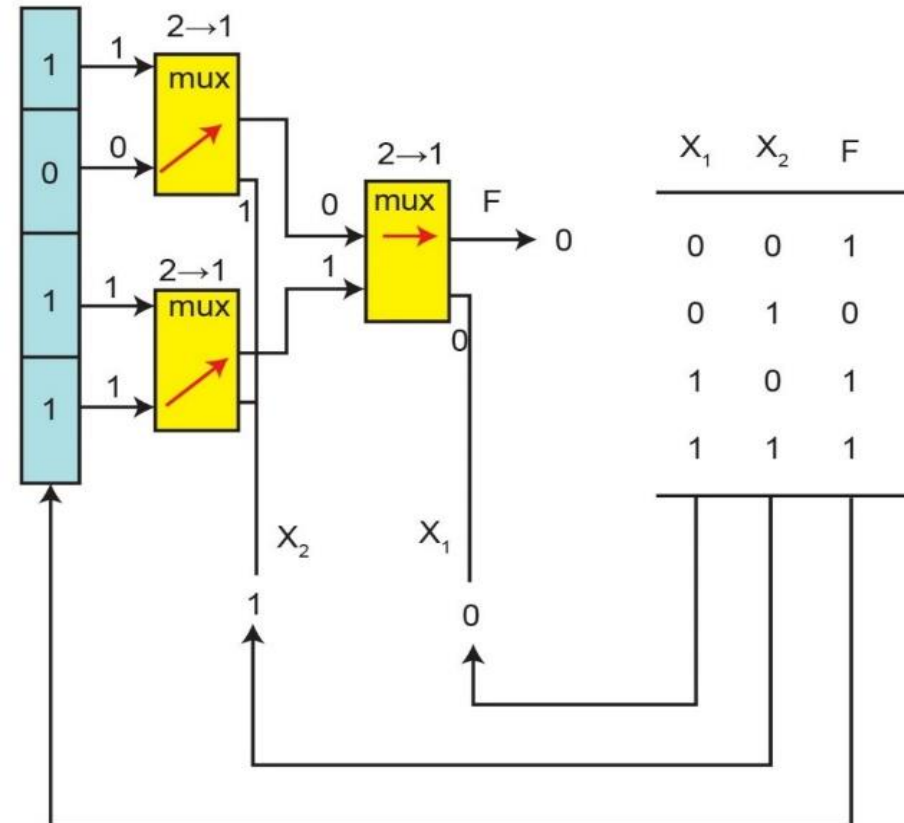
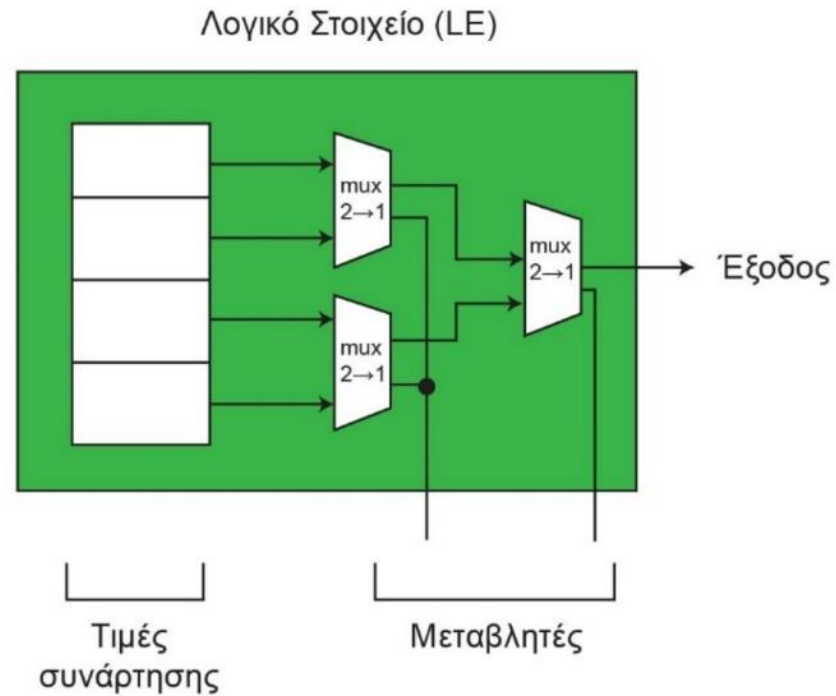


-  Λογικό Στοιχείο (Logic Element)
-  Προγραμματιζόμενος Διακόπτης (Programmable Switch)

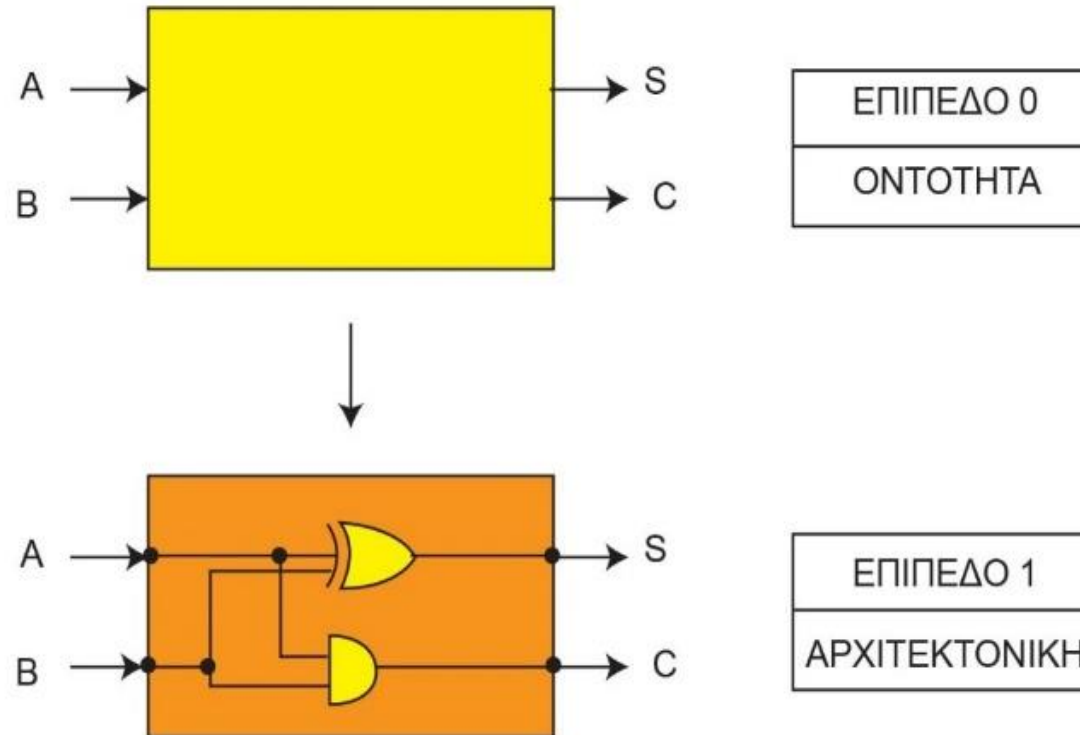


# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα; Επαναπρογραμματιζόμενες ψηφιακές διατάξεις (FPGA) (3)

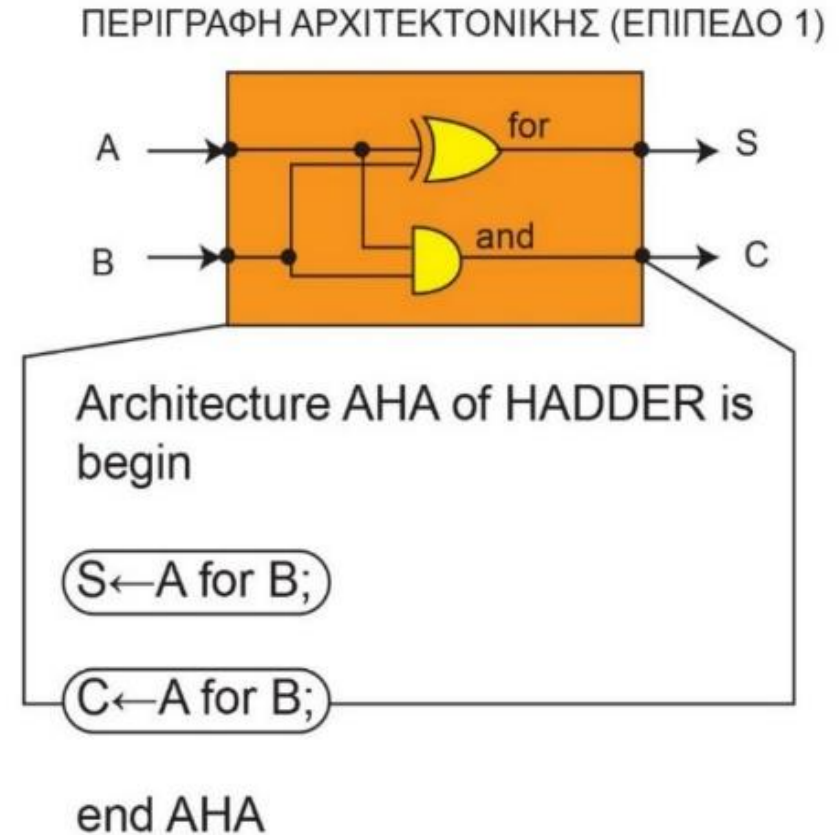
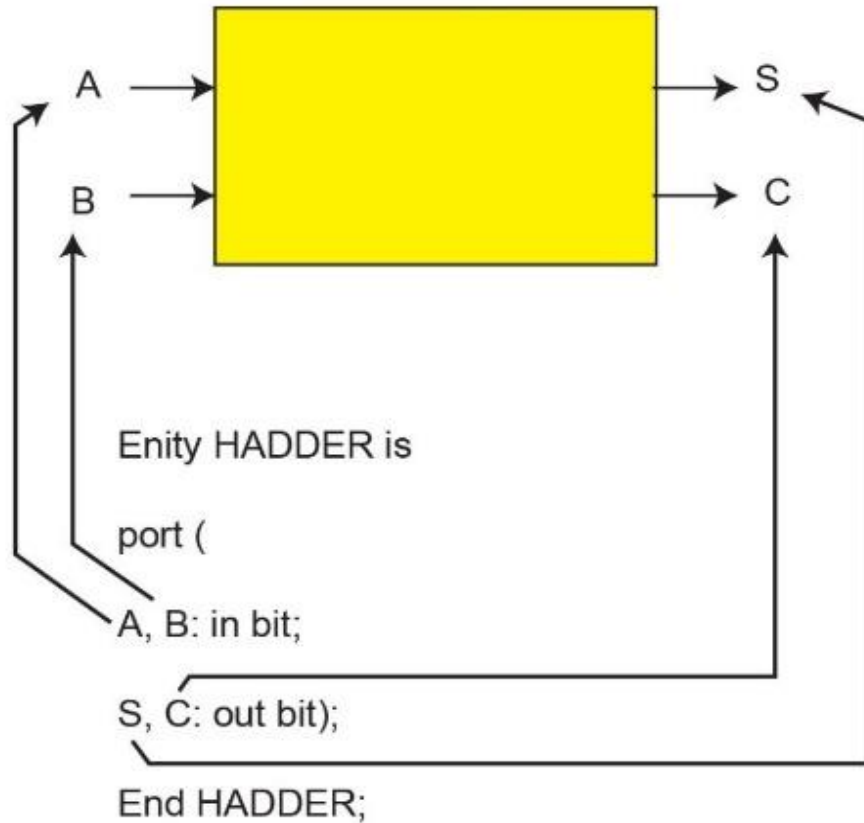
## Το Λογικό Στοιχείο (ΛΣ)



# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα; Επαναπρογραμματιζόμενες ψηφιακές διατάξεις (FPGA) (4) Η γλώσσα VHDL



# Πώς σχεδιάζονται σήμερα τα προηγμένα ψηφιακά κυκλώματα; Επαναπρογραμματιζόμενες ψηφιακές διατάξεις (FPGA) (5) Η γλώσσα VHDL



**Ολοκλήρωση κεφαλαίου**  
**Δείτε τις ασκήσεις από το βιβλίο**

