

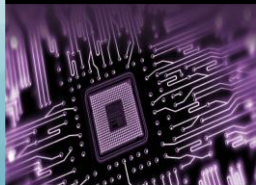
# Μικροεπεξεργαστές Αρχές & Εφαρμογές

## Υποστηρικτικό υλικό για τη διδασκαλία

Δρ. Παναγιώτης Παπάζογλου

Εκδόσεις Τζιόλα

Μικροεπεξεργαστές  
Αρχές & Εφαρμογές

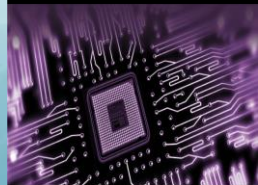


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Κεφάλαιο 1 - Εισαγωγή

**Μικροεξεργαστές**

Αρχές & Εφαρμογές

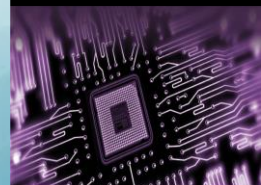
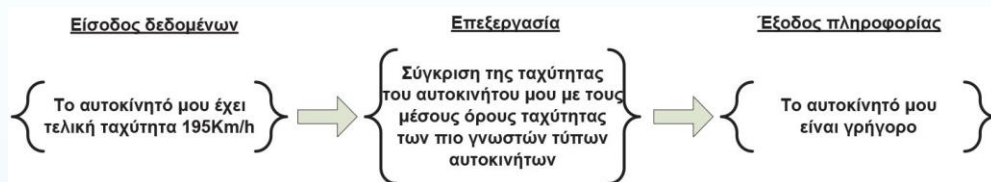


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

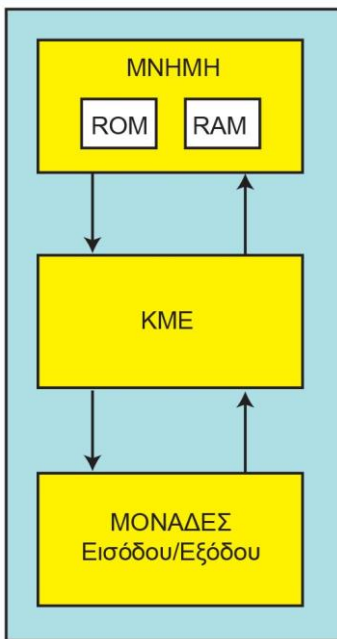
# Φιλοσοφία λειτουργίας υπολογιστικού συστήματος



## Παράδειγμα επεξεργασίας



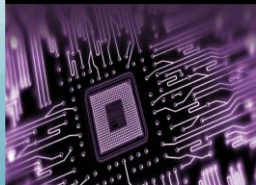
# Δομή Υπολογιστή



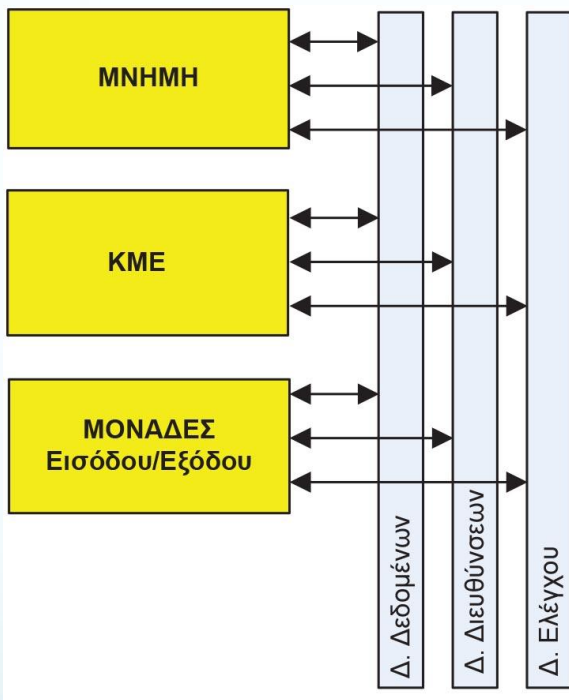
**Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ).** Η ΚΜΕ υλοποιείται σε ένα ολοκληρωμένο ψηφιακό κύκλωμα (IC) που ονομάζεται **μικροεπεξεργαστής**. Η αποστολή του μικροεπεξεργαστή είναι η εκτέλεση των εντολών που απαρτίζουν το πρόγραμμα.

**Μνήμη.** Η μνήμη φιλοξενεί τα προγράμματα (εντολές) καθώς και τα δεδομένα που αυτά διαχειρίζονται.

**Μονάδες Εισόδου/Εξόδου.** Οι μονάδες αυτές εξασφαλίζουν τόσο την αλληλεπίδραση του υπολογιστή με τον «έξω κόσμο» (π.χ. πληκτρολόγιο, οθόνη, εκτυπωτής), όσο και με αλλά βασικά συστατικά όπως τα μέσα αποθήκευσης (π.χ. σκληρός δίσκος), κλπ.



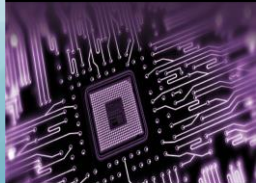
# Σύστημα διαύλων



**Δίαυλος Δεδομένων (Data Bus)**  
μεταφορά των δεδομένων (π.χ. μεταξύ μικροεπεξεργαστή και μνήμης)

**Δίαυλος Διευθύνσεων (Address Bus)**  
ενεργοποίηση της τοποθεσίας που θα μεταφερθούν τα δεδομένα

**Δίαυλος Ελέγχου (Control bus)**  
συντονισμός μονάδων συστήματος



# Βασικές υπομονάδες ΚΜΕ

## ΚΜΕ

Αριθμητική και Λογική  
μονάδα (ΑΛΜ-ALU)

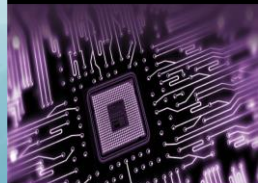
Μονάδα Ελέγχου (CU)

Σύστημα  
καταχωρητών

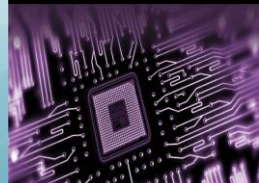
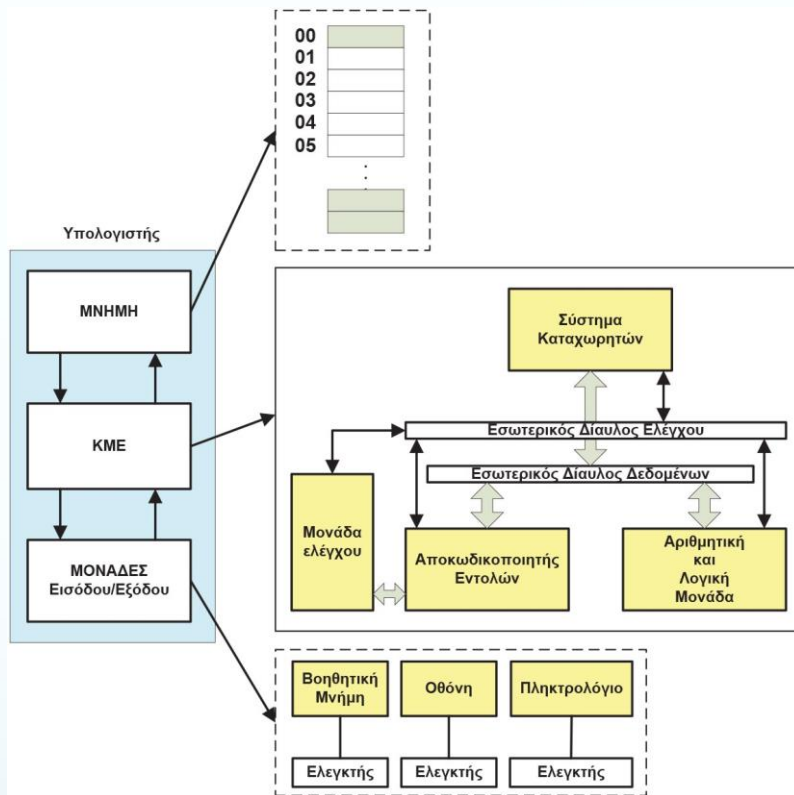
**Αριθμητική και Λογική Μονάδα** (ΑΛΜ-ALU-Arithmetic & Logic Unit) = εκτέλεση αριθμητικών και λογικών πράξεων

**Μονάδα ελέγχου** = «συντονισμός» εσωτερικών λειτουργιών μικροεπεξεργαστή

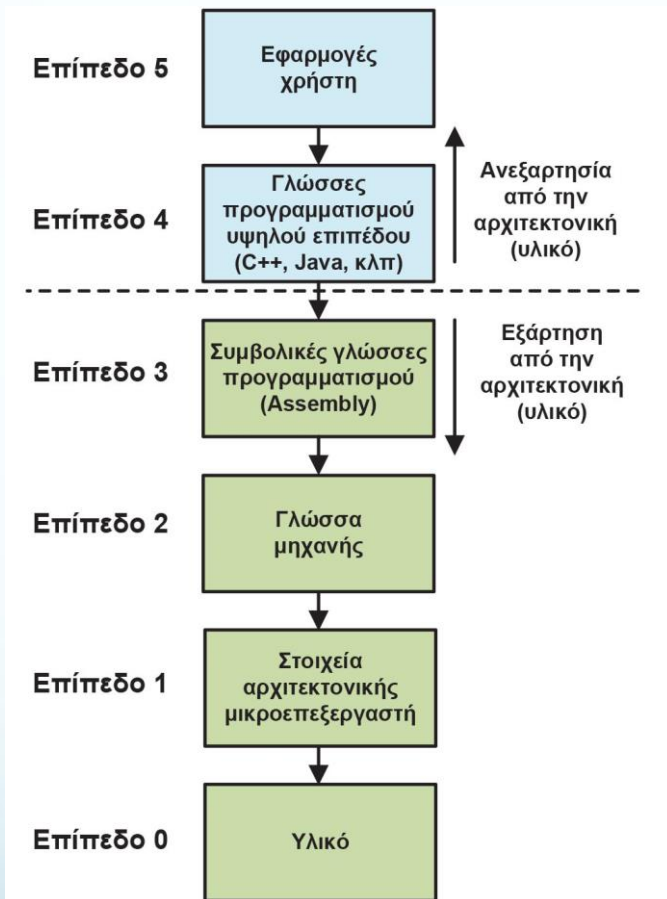
**Καταχωρητές** = συγκεκριμένες περιοχές αποθήκευσης που χρησιμοποιούνται για τη διαχείριση των δεδομένων που λαμβάνει χώρα στο εσωτερικό της ΚΜΕ



# Βασικά χαρακτηριστικά υπολογιστικού συστήματος



# Επίπεδα συστήματος

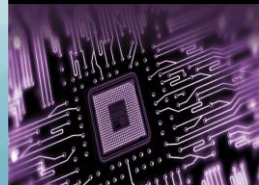


## Υλικό (Hardware)

- ❑ Περιλαμβάνει ολοκληρωμένα κυκλώματα και ηλεκτρονικές διατάξεις
- ❑ Καθιστά το υπολογιστικό σύστημα λειτουργικό

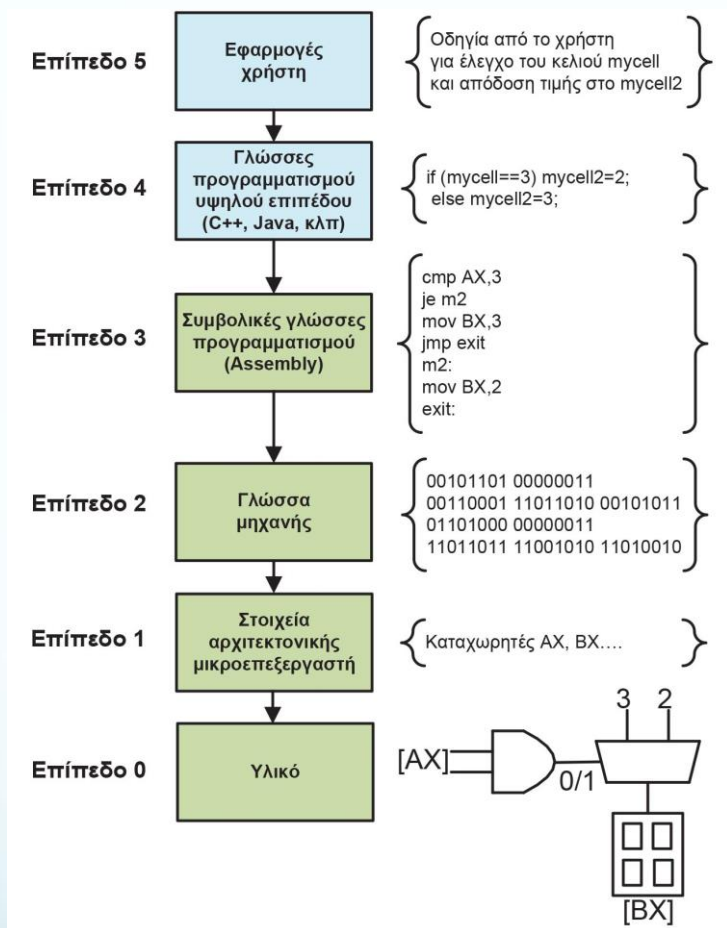
## Λογισμικό (Software)

- ❑ Δίνει «ζωή» στο υλικό
- ❑ Υποστηρίζει τη λειτουργία του συστήματος και τις εφαρμογές των χρηστών
- ❑ Αξιοποιεί το υλικό



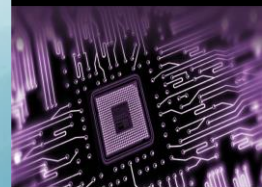


# Στάδια υλοποίησης οδηγίας-εντολής

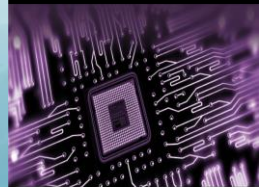
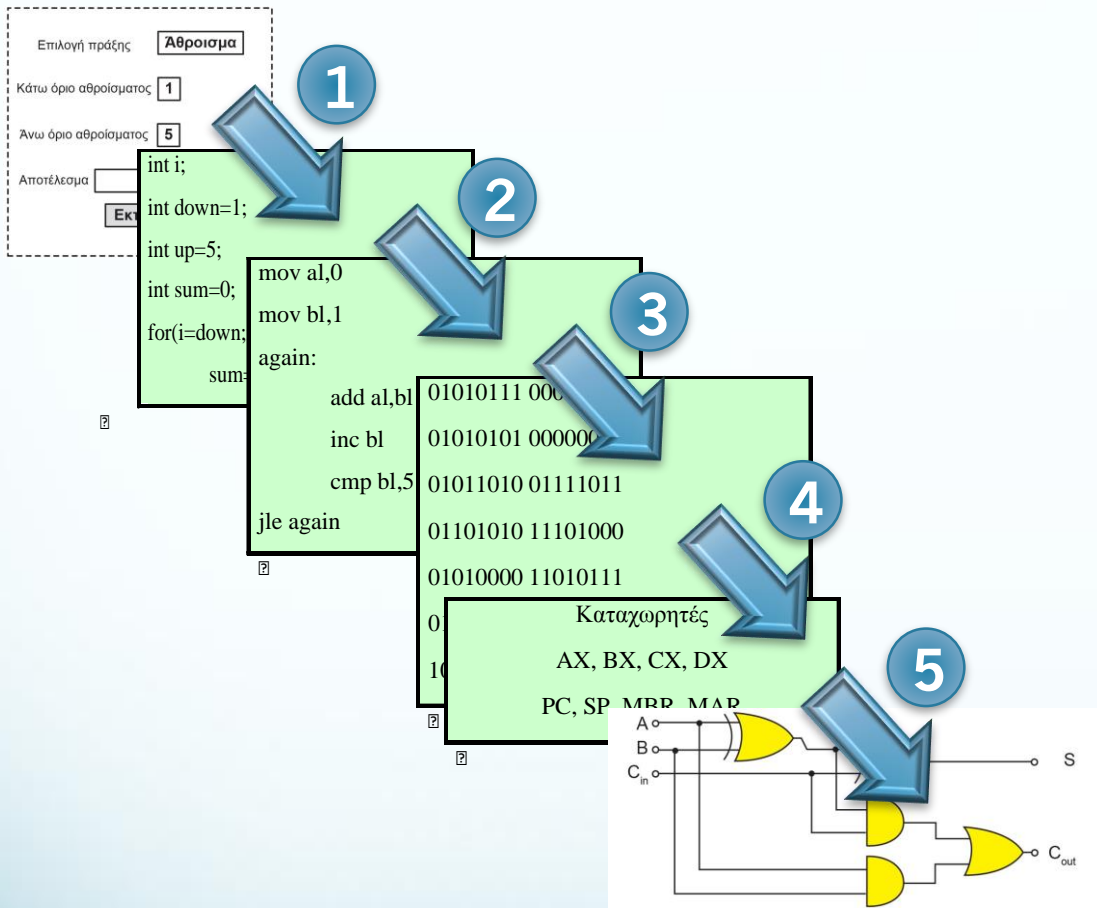


## Ένα ενδεικτικό παράδειγμα

το τελικό ψηφιακό κύκλωμα είναι υποθετικό και δεν αντανακλά τα κυκλώματα γενικού σκοπού που χρησιμοποιούνται στα σημερινά υπολογιστικά συστήματα



# «Ροή» υλοποίησης οδηγίας-εντολής



# Συστήματα γενικού και ειδικού σκοπού

**Συστήματα γενικού σκοπού** = ανάπτυξη μεγάλου εύρους εφαρμογών για κάθε χρήστη. Τα συστήματα αυτά βασίζονται στους σημερινούς σύγχρονους **μικροεπεξεργαστές** που έχουν οι υπολογιστές που χρησιμοποιούμε.

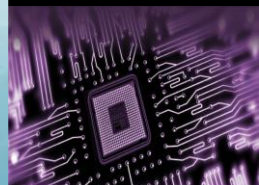
**Συστήματα ειδικού σκοπού** = καλύπτουν ένα συγκεκριμένο εύρος εφαρμογών. Εκτελούν συνήθως το ίδιο πρόγραμμα και επεξεργάζονται συγκεκριμένα δεδομένα. Στα σημερινά συστήματα ειδικού σκοπού χρησιμοποιούνται ειδικού τύπου ολοκληρωμένα κυκλώματα που καλούνται **μικροελεγκτές**.

**Μικροεπεξεργαστής** = απαιτεί μεγάλο πλήθος εξωτερικών κυκλωμάτων για τη λειτουργία του (μνήμη, κυκλώματα ελεγκτών, περιφερειακές μονάδες, κλπ).  
**Έμφαση** στις επιδόσεις.

**Μικροελεγκτής** = λειτουργεί αυτόνομα (περιλαμβάνει μνήμη, ελεγκτές εισόδου/εξόδου στο ίδιο ολοκληρωμένο κύκλωμα).  
**Έμφαση** στην αυτονομία λειτουργίας

Μικροεπεξεργαστές

Αρχές & Εφαρμογές

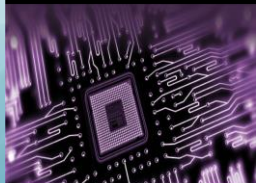
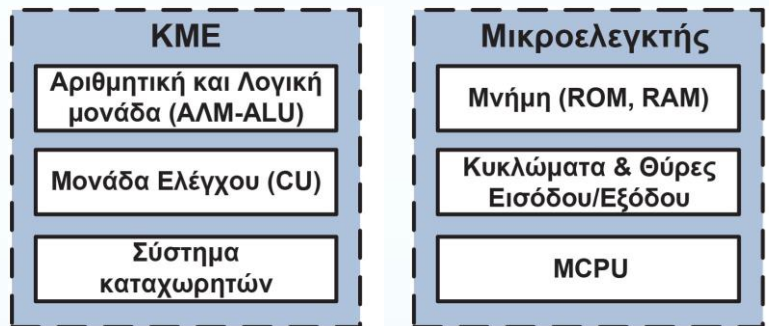


Παναγιώτης Παπάζογλου

Εκδόσεις Τζόλα

# Διαφορές μικροεπεξεργαστή - μικροελεγκτή

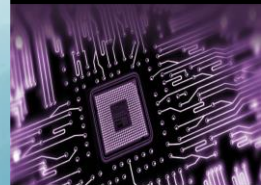
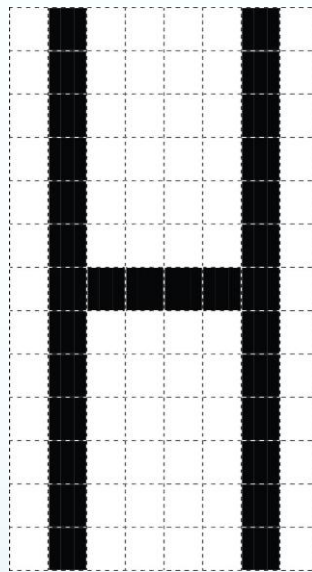
- ❑ Υπολογιστική ισχύς (ταχύτητα υπολογισμών)
- ❑ Είδος υποστηριζόμενων εφαρμογών (γενικού σκοπού ή υποστήριξη συγκεκριμένων εφαρμογών)
- ❑ Κλίμακα ολοκλήρωσης (πλήθος και πυκνότητα ψηφιακών κυκλωμάτων)
- ❑ Κόστος
- ❑ Μέγεθος κατασκευής
- ❑ Τεχνολογία κατασκευής
- ❑ Αυτονομία λειτουργίας
- ❑ Φορητότητα



# Bit & Byte

- ❑ **Bit** = Ελάχιστη μονάδα πληροφορίας = 1 δυαδικό ψηφίο = τιμή 0 ή 1
- ❑ **1 Byte** = 8 bit
- ❑ Παράδειγμα αναπαράστασης με bit (σύμβολο στην οθόνη)

0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0

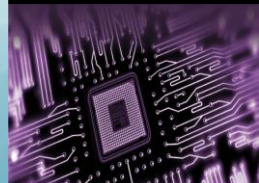


# Δυαδικές ετικέτες σε προϊόντα

- ❑ 1 προϊόν = 1 μοναδική ετικέτα
- ❑ Πόσα δυαδικά ψηφία απαιτούνται για κάθε ετικέτα (αποτύπωση 10 διαφορετικών προϊόντων)
- ❑  $2^N \geq 10$ , δηλαδή απαιτούνται  $N=4$  δυαδικά ψηφία
- ❑ Γενικά, για  $K$  αντικείμενα, απαιτούνται  $N$  δυαδικά ψηφία, έτσι ώστε  $2^N \geq K$

Ετικέτες προϊόντων	
Ετικέτα	Προϊόν
0000	Προϊόν Α
0001	Προϊόν Β
0010	Προϊόν Γ
0011	Προϊόν Δ
0100	Προϊόν Ε
0101	Προϊόν Ζ
0110	Προϊόν Η
0111	Προϊόν Θ
1000	Προϊόν Ι
1001	Προϊόν Κ

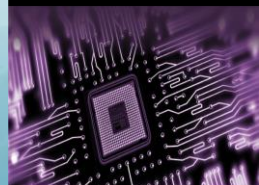
Τι γνωρίζετε για τον κώδικα ASCII; Πως γίνεται η αναπαράσταση κάθε συμβόλου; Αναζητήστε απαντήσεις στο Ύστερνετ



# Εκφράζοντας μεγαλύτερο όγκο πληροφορίας

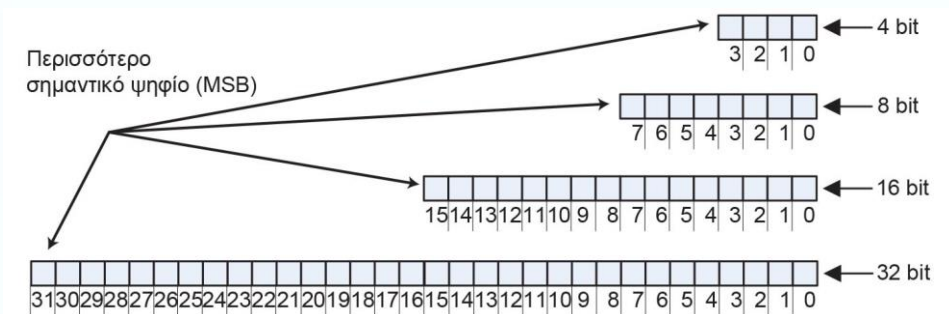
Προθέματα		
Σύμβολο	Πρόθεμα	Αξία
P	Peta	$10^{15}$
T	Tera	$10^{12}$
G	Giga	$10^9$
M	Mega	$10^6$
K	Kilo	$10^3$

- ❑ 1Kbyte=1024byte (και όχι 1000byte)
- ❑ N Kbyte = N πολλαπλάσια των 1024byte
- ❑ 64Kbyte = 64 x 1024 byte

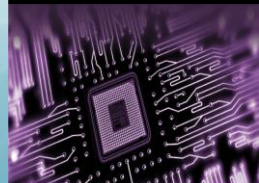


# Βασικά χαρακτηριστικά μικροεπεξεργαστών

## ❑ Μήκος λέξης

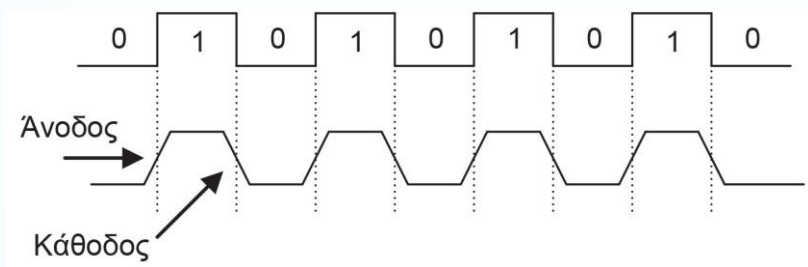


- ❑ Μήκος εντολής
- ❑ Τρόποι διευθυνσιοδότησης και πλήθος καταχωρητών
- ❑ Χρονισμός και χρονική απόκριση (επιδόσεις σε επίπεδο εντολών)





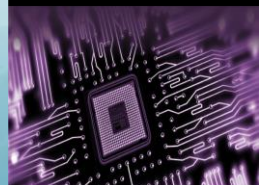
# Σήμα ρολογιού



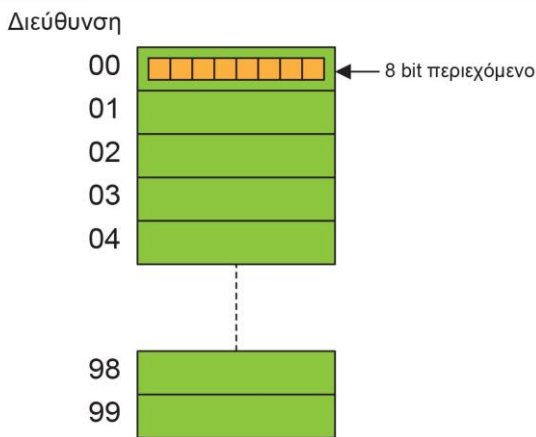
❑ Ιδανικό σήμα

❑ Προσεγγίζοντας περισσότερο το πραγματικό

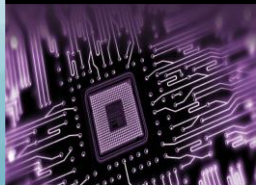
Ένα από τα βασικά κριτήρια που καθορίζουν τις επιδόσεις του συστήματος



# Γενικό μοντέλο μνήμης



- ❑ Κάθε θέση = ίδια χωρητικότητα
- ❑ Μοναδικό αναγνωριστικό κάθε θέσης = μοναδική διεύθυνση
- ❑ Μνήμη  $K$  θέσεων απαιτεί  $N$  bit διεύθυνσης για κάθε θέση ( $2^N \geq K$ )
- ❑ Συνηθισμένο μοντέλο = Μια θέση μνήμης έχει χωρητικότητα 8bit (1 byte)



# Προγραμματισμός χαμηλού επιπέδου

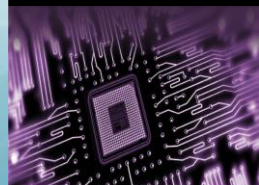
**Συμβολική γλώσσα (Assembly)** = γλώσσα βασισμένη απευθείας στην αρχιτεκτονική του μικροεπεξεργαστή

## ❑ Πλεονεκτήματα

- Ταχύτερα και μικρότερα προγράμματα
- Άμεση πρόσβαση στο υλικό και τις δυνατότητές του
- Κατανόηση της αρχιτεκτονικής και της λειτουργίας του υπολογιστικού συστήματος

## ❑ Μειονεκτήματα

- Υψηλότερη πολυπλοκότητα και μεγαλύτερος χρόνος ανάπτυξης
- Μη μεταφίσιμος κώδικας



# Το χθες και το σήμερα (1)



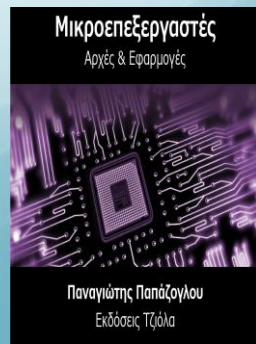
1981

Από το 1981 στο 2015		
Στοιχείο	ZX81	Σημερινός υπολογιστής (ενδεικτικά)
Έτος	1981	2015
Μικροεπεξεργαστής	Z80	Intel® Core™ i7-4790K (4 πυρήνες)
Χρονισμός (ρολόι συστήματος)	3.25MHz	3.00-4.40GHz
Μνήμη RAM	1K (επεκτάσιμη στα 16K)	4-32GB
Μήκος λέξης	8bit	64bit
Σύστημα απεικόνισης	Σε μονόχρωμη τηλεόραση (UHF)	Σε οθόνη (ο μικροεπεξεργαστής περιέχει και υποεπεξεργαστή απεικόνισης)
Ανάλυση κειμένου	24 γραμμές των 32 χαρακτήρων	Μεγαλύτερο από 80X100
Ανάλυση γραφικών	64 x 48 pixels	Μεγαλύτερο από 1920x1200
Τροφοδοσία	9V	Πολυαπλή +/-
Εξωτερική αποθήκευση	Όχι	Σκληρός δίσκος 1-5TB



2015

\* Εικόνες με πνευματικά δικαιώματα. Δείτε το βιβλίο για περισσότερες πληροφορίες.



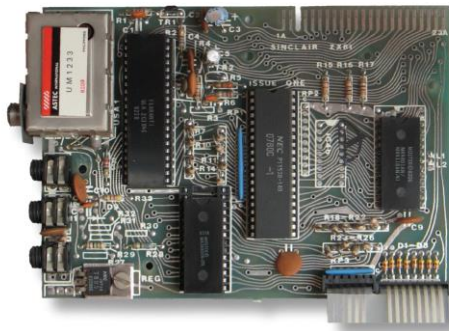
# Το χθες και το σήμερα (2)



**Z80**



**INTEL i7**



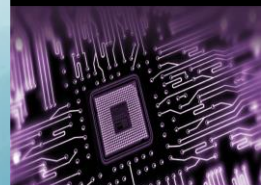
**Board για Z80**



**Board για INTEL**

\* Εικόνες με πνευματικά δικαιώματα. Δείτε το βιβλίο για περισσότερες πληροφορίες.

Μικροεξεργαστές  
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάλα

# Εξέλιξη της κλίμακας ολοκλήρωσης

**Περισσότερες δυνατότητες και υπολογιστική ισχύς** = περισσότερα και πολλές φορές πολυπλοκότερα εσωτερικά ψηφιακά κυκλώματα

**Υλοποίηση κυκλωμάτων** = ημιαγωγοί

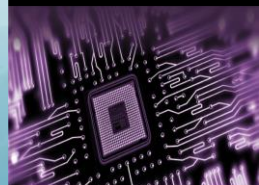
**Μικρότερο φυσικό μέγεθος και αποστάσεις τοποθέτησης των ημιαγωγών** = περισσότερα κυκλώματα, περισσότερες δυνατότητες

**Κλίμακα ολοκλήρωσης** = Πλήθος ημιαγωγών (τρανζίστορ) που περιέχονται στο ολοκληρωμένο κύκλωμα

Εξέλιξη της κλίμακας ολοκλήρωσης		
Έτος	Ονομασία	Πλήθος στοιχείων
Αρχή δεκαετίας '60	<b>SSI</b> (Small Scale Integration)	Μερικές δεκάδες
Τέλη δεκαετίας '60	<b>MSI</b> (Medium Scale Integration)	Μερικές εκατοντάδες
Μέσα δεκαετίας '70	<b>LSI</b> (Large Scale Integration)	Δεκάδες χιλιάδες
Δεκαετία '80	<b>VLSI</b> (Very Large Scale Integration)	Ξεκίνησαν με εκατοντάδες χιλιάδες και έχουν φτάσει σήμερα σε επίπεδο δισεκατομμυρίου

Μικροεξεργαστές

Αρχές & Εφαρμογές



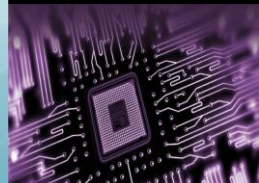
Παναγιώτης Παπάζογλου

Εκδόσεις Τζόλα

# Κεφάλαιο 1

## Τα κύρια σημεία σε τίτλους

- ❑ Αποστολή του υπολογιστή
- ❑ Επεξεργασία-Πρόγραμμα-ΚΜΕ
- ❑ Συστατικά του υπολογιστή
- ❑ Σύστημα διαύλων
- ❑ Γλώσσες προγραμματισμού υψηλού και χαμηλού επιπέδου
- ❑ Υλικό (hardware) και λογισμικό (software)
- ❑ Συστήματα γενικού και ειδικού σκοπού
- ❑ Μικροεπεξεργαστές και Μικροελεγκτές
- ❑ BIT και BYTE
- ❑ Πλήθος ψηφίων-αναπαράσταση καταστάσεων
- ❑ Προθέματα-όγκος πληροφορίας
- ❑ Βασικά χαρακτηριστικά μικροεπεξεργαστών
- ❑ Ιδανικό και πιο πραγματικό σήμα ρολογιού
- ❑ Γενικό μοντέλο μνήμης
- ❑ Πλεονεκτήματα και μειονεκτήματα Assembly

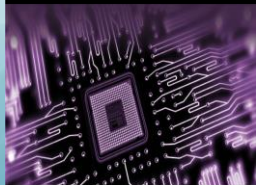


# Κεφάλαιο 2

## Αριθμητικά συστήματα και μετατροπές

Μικροεξεργαστές

Αρχές & Εφαρμογές



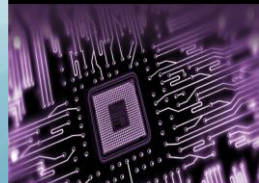
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα



# Βασικά χαρακτηριστικά αριθμητικών συστημάτων

Βασικά χαρακτηριστικά αριθμητικών συστημάτων			
	Βάση συστήματος (b)	Πλήθος συμβόλων	Σύμβολα (0 έως b-1)
Δεκαδικό	10	10	0,1,2,3,4,5,6,7,8,9
Δυαδικό	2	2	0,1
Δεκαεξαδικό	16	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

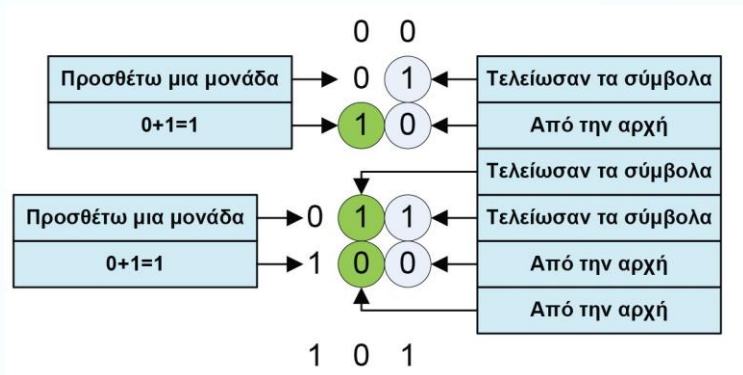
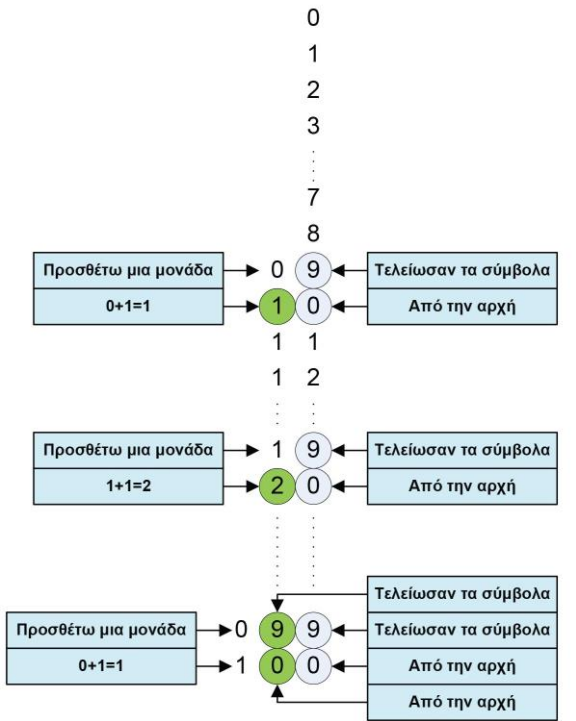
2



# Μέτρηση στα αριθμητικά συστήματα

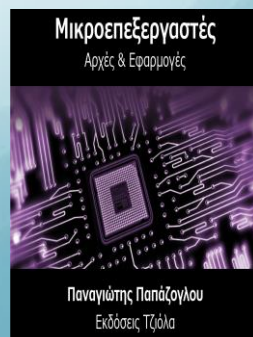
Δεκαδικό

Δυαδικό

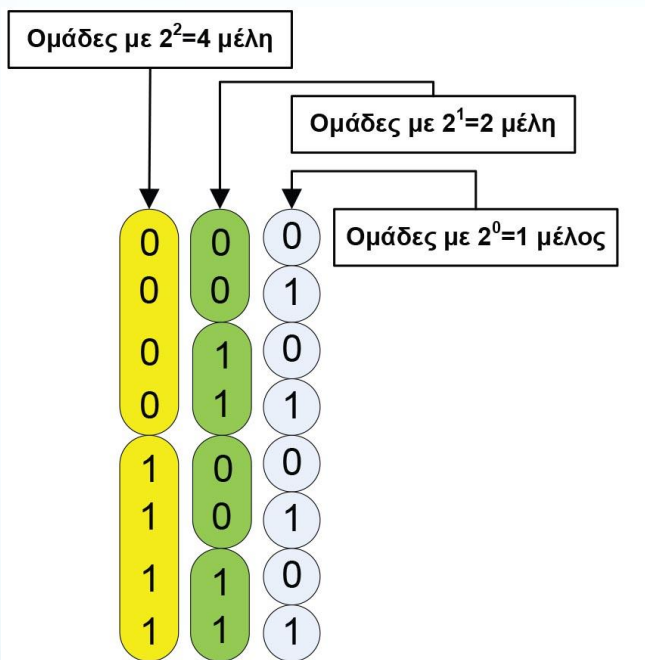


Δεκαεξαδικό

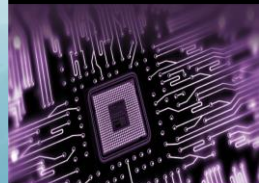
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20  
...



# Γρήγορη μέτρηση με εναλλαγή ομαδοποιημένων συμβόλων



Αναδίκιο



# Υπολογισμός αριθμητικής αξίας στο δεκαδικό σύστημα

$$N = \sum_{i=-n}^{m-1} a_i b^i = a_{m-1}b^{m-1} + a_{m-2}b^{m-2} + \dots + a_1b^1 + a_0b^0 + \dots \\ + a_{-1}b^{-1} + a_{-2}b^{-2} + a_{-n}b^{-n}$$

**Θετικοί εκθέτες** = για τα ψηφία αριστερά από την υποδιαστολή

**Αρνητικοί εκθέτες** = για τα ψηφία δεξιά από την υποδιαστολή

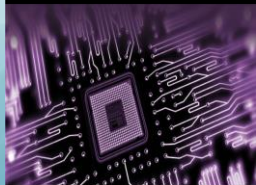
**$\alpha$**  = τιμή ψηφίου (θέση  $i$ )

**$b$**  = βάση αριθμητικού συστήματος

**$N$**  = τελική αριθμητική αξία (στο δεκαδικό σύστημα)

Μικροεξεργαστές

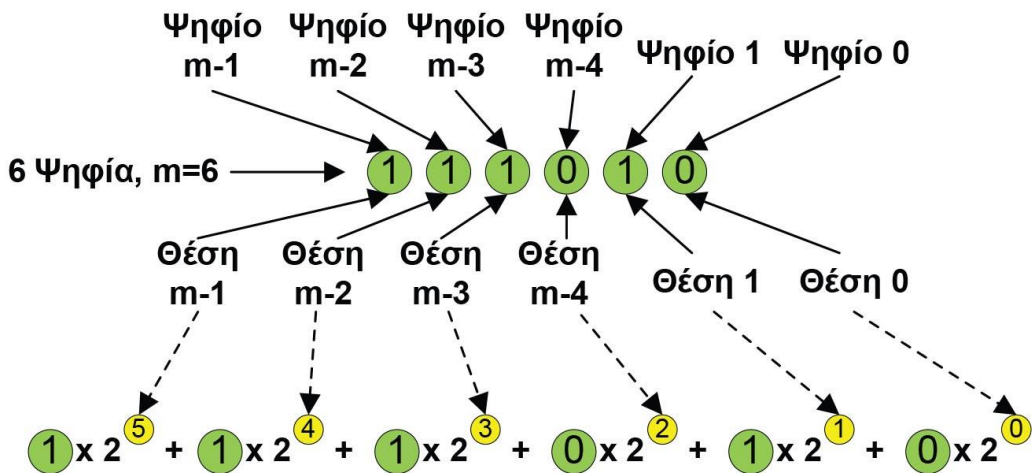
Αρχές & Εφαρμογές



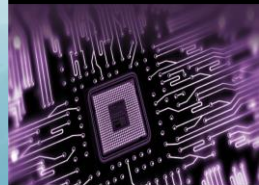
Παναγιώτης Παπάζογλου

Εκδόσεις Τζόλα

# Παράδειγμα υπολογισμού αριθμητικής αξίας

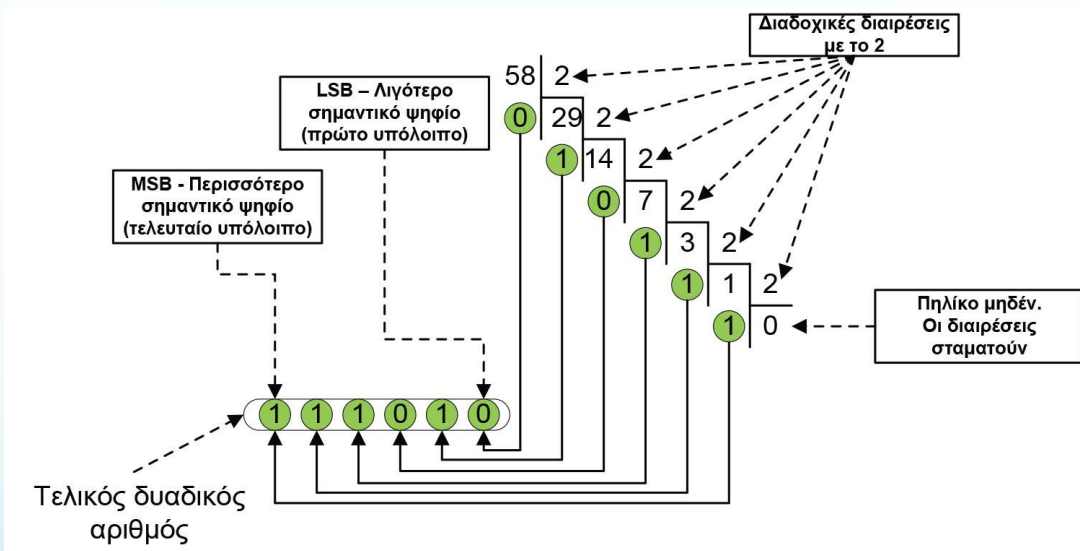


$$\begin{aligned} & 111010_{(2)} \\ & = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ & = 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 \\ & = 32 + 16 + 8 + 2 \\ & = 58_{(10)} \end{aligned}$$



# Μετατροπή αριθμού σε άλλο σύστημα με διαδοχικές διαιρέσεις

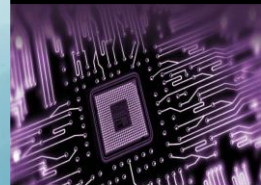
Μετατροπή του αριθμού 58 στο δυαδικό σύστημα



# Γρήγορη μετατροπή με πίνακα βαρών

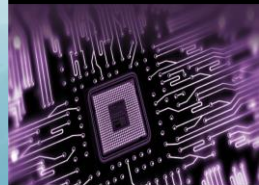
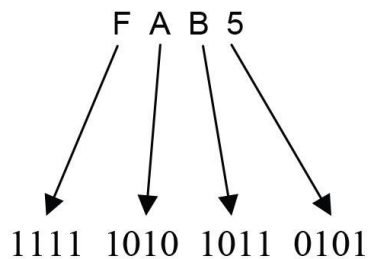
Πίνακας βαρών						
$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
64	32	16	8	4	2	1
Δυαδική μορφή αριθμού $58_{(10)}$						
0	1	1	1	0	1	0
Βάρος ψηφίων που χρησιμοποιούνται						
0	32	16	8	0	2	0
Άθροισμα βαρών: $32+16+8+2 = 58$						

Ο αριθμός 58  
στο δυαδικό



# Γρήγορες μετατροπές μεταξύ δυναδικού και δεκαεξαδικού συστήματος

Δεκαεξαδικό	Δυναδικό
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

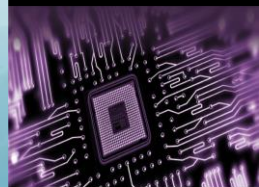




# Κεφάλαιο 2

## Τα κύρια σημεία σε τίτλους

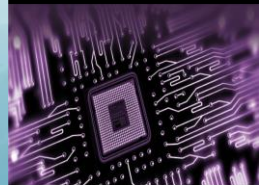
- ❑ Χαρακτηριστικά αριθμητικών συστημάτων
- ❑ Μέτρηση στο δυαδικό σύστημα
- ❑ Αξία στο δεκαδικό σύστημα
- ❑ Μετατροπές με πίνακα βαρών
- ❑ Γρήγορες μετατροπές μεταξύ δυαδικού και δεκαεξαδικού συστήματος



# Κεφάλαιο 3

## Αριθμητική μικροεπεξεργαστή

**Μικροεπεξεργαστές**  
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Πρόσθεση δύο και τριών αριθμών του ενός bit

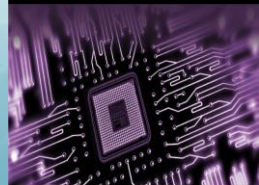
Δύο αριθμοί

A	B	A+B	Κρατούμενο
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} \text{Κρατούμενο } 1 \\ \phantom{0} \phantom{0} \phantom{0} \\ + \phantom{0} \phantom{0} \phantom{0} \\ \hline 1 \phantom{0} \phantom{0} \end{array}$$

Τρεις αριθμοί

A	B	C	A+B+C	Κρατούμενο
0	0	0	0	0
0	0	1	1	0
0	1	0	1	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

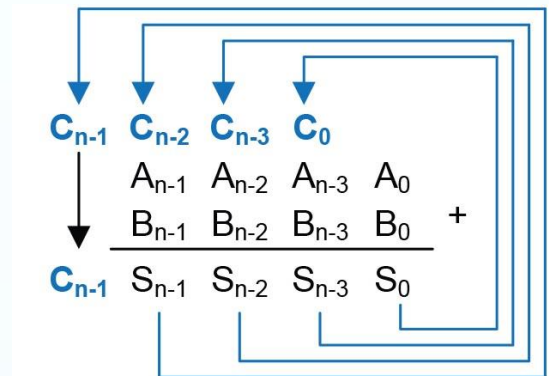


# Πρόσθεση αριθμών n bit

Πρόσθεση ψηφίων ανά στήλη

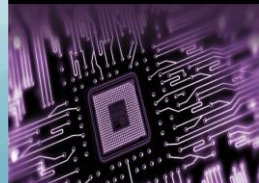
$$\begin{array}{rcccc} C_{n-1} & C_{n-2} & C_{n-3} & C_0 \\ \downarrow & & & & \\ A_{n-1} & A_{n-2} & A_{n-3} & A_0 & \\ \downarrow & & & & \\ B_{n-1} & B_{n-2} & B_{n-3} & B_0 & + \\ \hline C_{n-1} & S_{n-1} & S_{n-2} & S_{n-3} & S_0 \end{array}$$

Διάδοση κρατούμενου



Μικροεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Παράδειγμα πρόσθεσης δύο αριθμών των τεσσάρων bit

					Πράξη	Άθροισμα	Κρατούμενο (πιθανό)
$C_3$	$C_2$	$C_1$	$C_0$		$A_0+B_0$	$S_0$	$C_0$
	$A_3$	$A_2$	$A_1$	$A_0$	$C_0+A_1+B_1$	$S_1$	$C_1$
	$B_3$	$B_2$	$B_1$	$B_0$	$C_1+A_2+B_2$	$S_2$	$C_2$
					$C_2+A_3+B_3$	$S_3$	$C_3$
$C_3$	$S_3$	$S_2$	$S_1$	$S_0$			

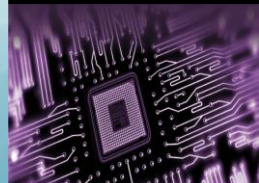
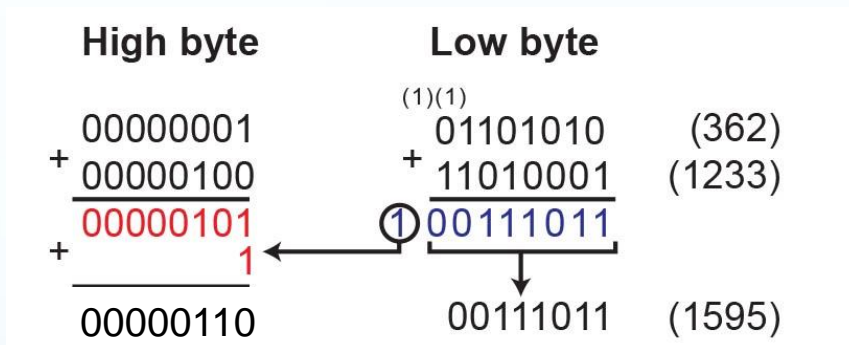
## Αριθμητικό παράδειγμα

$$\begin{array}{rcccccc}
 \textcircled{1} & \textcircled{1} & \textcircled{1} & \textcircled{1} & \leftarrow \text{Κρατούμενα} & & \\
 \downarrow & 1 & 1 & 0 & 1 & & \\
 \downarrow & 0 & 1 & 1 & 1 & + & \\
 \hline
 1 & 0 & 1 & 0 & 0 & & 
 \end{array}$$

# Πρόσθεση πολλαπλής ακρίβειας

- ❑ Πρόσθεση δύο αριθμών που εκφράζονται με 2 byte
- ❑ Κρατούμενο από την πρόσθεση των λιγότερων σημαντικών byte θα συνυπολογιστεί στην πρόσθεση των περισσότερο σημαντικών byte

Παράδειγμα (362+1233)



# Αφαίρεση δύο και τριών αριθμών του ενός bit

Δύο αριθμοί

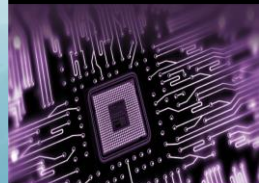
A	B	A-B	Δανεικό
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Παράδειγμα αφαίρεσης με τη χρήση δανεικού

$$\begin{array}{r}
 \rightarrow 1 \\
 100101 \\
 010001 - \\
 \hline
 010100
 \end{array}$$

Τρεις αριθμοί

A	B	C	A-B-C	Δανεικό
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



# Πολλαπλασιασμός

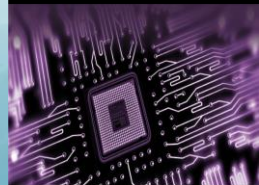
Δύο αριθμοί του  
ενός bit

A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1

Αριθμητικό παράδειγμα

$$\begin{array}{r} \phantom{000}1011 \\ \text{Κρατούμενο} \phantom{00}1101 \times \\ \hline \phantom{000}11011 \\ \phantom{000}10000 \\ \phantom{000}11011 \phantom{0} + \\ \phantom{000}1011 \\ \hline 10001111 \end{array}$$

Μικροεξεργαστές  
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα



# Διαίρεση με διαδοχικές αφαιρέσεις

- ❑ Ξεκινάμε αρχικά αφαιρώντας από τον διαιρετέο τον διαιρέτη
- ❑ Οι αφαιρέσεις συνεχίζονται αφαιρώντας εκ νέου τον διαιρέτη από το προηγούμενο αποτέλεσμα
- ❑ Η διαδικασία ολοκληρώνεται όταν προκύψει αποτέλεσμα μηδέν ή αρνητικό
- ❑ Το πλήθος των αφαιρέσεων (εκτός αυτής που έδωσε αρνητικό αποτέλεσμα) αντιστοιχεί στο πηλίκο
- ❑ Το υπόλοιπο προκύπτει από την τελευταία διαίρεση που δεν έδωσε αρνητικό αποτέλεσμα

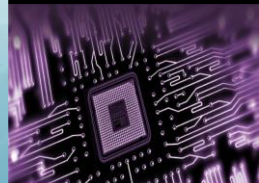
Παράδειγμα  
11/2

Βήμα 1	11-
2=9	
Βήμα 2	9-
2=7	
Βήμα 3	7-
2=5	
Βήμα 4	5-
2=3	
Βήμα 5	3-
2=1	
Βήμα 6	1-
2=-1	

Πηλίκο 5  
(5 αποδεκτές  
αφαιρέσεις)

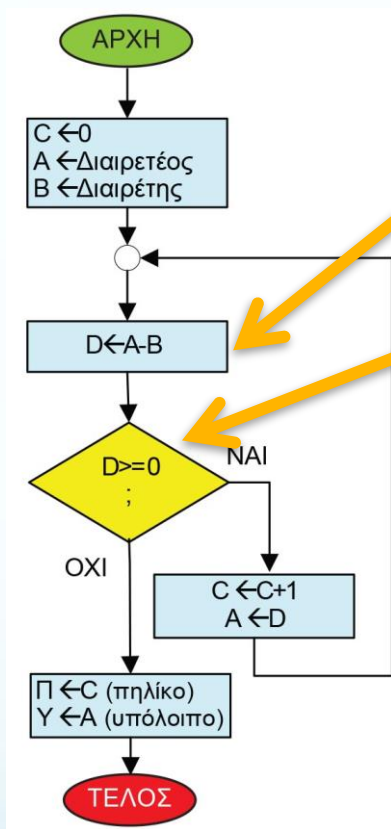
Υπόλοιπο 1

Μικροεπεξεργαστές  
Αρχές & Εφαρμογές



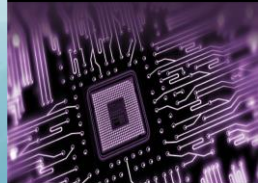
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

# Αλγόριθμος διαίρεσης με διαδοχικές αφαιρέσεις



Αφαίρεση  
(διαιρετέος – διαιρέτης)

Συνθήκη για νέα  
αφαίρεση ή όχι  
Αν  $D \geq 0$  ενημερώνουμε  
το μετρητή των  
αφαιρέσεων (πηλίκo)  
Αν  $D < 0$  η διαδικασία  
σταματά αλλά χωρίς να  
αλλάξει το πηλίκo και το  
υπόλοιπο



# Προσημασμένοι αριθμοί <sup>(1)</sup>

## Αναπαράσταση «πρόσημο και μέτρο»

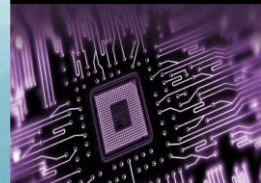
- ❑ **Αναπαράσταση πρόσημου** = ένα bit του αριθμού (Συνήθως το αριστερό)
- ❑ **Bit πρόσημου** = 1 αρνητικό, 0 θετικό
- ❑ Αν το σύνολο των bit ήταν τρία = εύρος θετικών 0)00 έως 0)11, εύρος αρνητικών 1)00 έως 1)11

Μέτρηση με προσημασμένους αριθμούς			
Δεκαδικό σύστημα	Μη προσημασμένοι	Δεκαδικό σύστημα	Προσημασμένοι «πρόσημο και μέτρο»
0	000	0	000
1	001	1	001
2	010	2	010
3	011	3	011
4	100	-0	100
5	101	-1	101
6	110	-2	110
7	111	-3	111
<b>Εύρος τιμών</b>	0 έως 7	<b>Εύρος τιμών</b>	-3 έως 3

Μέτρο του αριθμού

$$\sum_{i=0}^{n-2} 2^i \alpha_i \quad \text{εφόσον } \alpha_{n-1}=0$$

$$- \sum_{i=0}^{n-2} 2^i \alpha_i \quad \text{εφόσον } \alpha_{n-1}=1$$



# Προσημασμένοι αριθμοί <sup>(2)</sup>

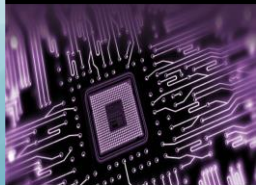
Αναπαράσταση με συμπλήρωμα ως προς 2

- ❑ Λύνει το πρόβλημα της διπλής αναπαράστασης του μηδενός
- ❑ Αυξάνει το εύρος των αριθμών που μπορούν να αναπαρασταθούν κατά 1
- ❑ Κυκλική διαδικασία εναλλαγής προσήμου
- ❑ Προκύπτει αντιστρέφοντας τα ψηφία του αριθμού και προσθέτοντας μια μονάδα

Κυκλική λειτουργία του συμπληρώματος

$$\Sigma_2(1100) = 0011 + 1 = 0100 = 4_{(10)}$$

$$\Sigma_2(0100) = \Sigma_2(4) = 1011 + 1 = 1100 = -4_{(10)}$$



# Αφαίρεση με συμπλήρωμα ως προς 2

□  $A-B=A+(-B)$ , Η αφαίρεση υλοποιείται τελικά με πρόσθεση

□  $-B$ =Συμπλήρωμα ως προς 2 ( $\Sigma_2$ ) του B

Παράδειγμα 1 ( $81-54=81+(-54)$ )

$$81_{(10)} = 1010001_{(2)}$$

$$54_{(10)} = 0110110_{(2)}$$

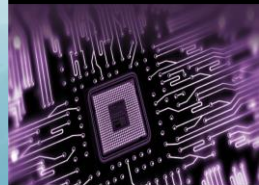
$$-54_{(10)} = \Sigma_2(54) = 1001001_{(2)} + 1 = 1001010_{(2)}$$

1 κρ.

$$\begin{array}{r} 1010001 \\ + 1001010 \\ \hline \end{array}$$

~~1~~0011011

Αν προκύψει κρατούμενο αριστερά από το περισσότερο σημαντικό ψηφίο (MSB), τότε το διαγράφουμε και το αποτέλεσμα είναι θετικό, ενώ το μέτρο του αποτελείται από τα υπόλοιπα bit



# Αφαίρεση με συμπλήρωμα ως προς 2

Παράδειγμα 2 ( $54-81=54+(-81)$ )

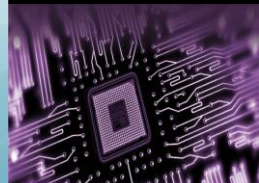
$$81_{(10)} = 1010001_{(2)}$$

$$54_{(10)} = 0110110_{(2)}$$

$$-81_{(10)} = \Sigma_2(81) = 0101110_{(2)} + 1 = 0101111_{(2)}$$

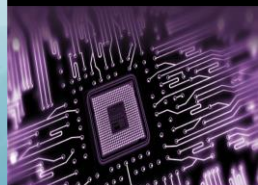
$$\begin{array}{r} 1111 \text{ κρ.} \\ 0110110 \\ + 0101111 \\ \hline 1100101 \end{array}$$

Αν δεν προκύψει κρατούμενο, τότε το αποτέλεσμα είναι αρνητικό με μέτρο το συμπλήρωμα ως προς 2 των bit του αποτελέσματος



# Διαίρεση με συμπλήρωμα ως προς 2

- ❑ Σύμφωνα με τον προηγούμενο αλγόριθμο, οι διαιρέσεις γίνονται με διαδοχικές αφαιρέσεις
- ❑ Οι αφαιρέσεις γίνονται με πρόσθεση (χρήση συμπληρώματος ως προς 2)
- ❑ Η διαίρεση υλοποιείται τελικά με πρόσθεση



# Αναπαράσταση αριθμών κινητής υποδιαστολής <sup>(1)</sup>

## Το πρότυπο IEEE 754

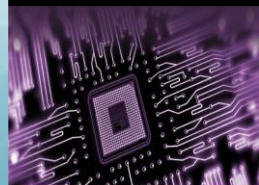
### Συστατικά αριθμού κινητής υποδιαστολής

- πρόσημο
- εκθέτης
- κλασματικό μέρος (γνωστό και στη βιβλιογραφία ως Mantissa)

**Βάση συστήματος** = θεωρείται πάντα ότι είναι το 2

### Τύποι αναπαράστασης

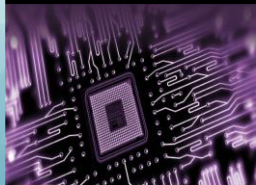
- απλής ακρίβειας, 32bit
- διπλής ακρίβειας, 64bit





# Αναπαράσταση αριθμών κινητής υποδιαστολής (2)

Πρότυπο απλής και διπλής ακρίβειας							
	Πρόσημο		Εκθέτης		Κλασματικό μέρος		Πόλωση
	Πλήθος bit	Θέση bit	Πλήθος bit	Θέση bit	Πλήθος bit	Θέση bit	
Απλή ακρίβεια	1	31	8	30-23	23	22-00	127
Διπλή ακρίβεια	1	63	11	62-52	52	51-00	1023



# Αναπαράσταση αριθμών κινητής υποδιαστολής (3)

$$X = (-1)^S \times (1+KM) \times 2^{(E-\Pi)}$$

X = τελικός αριθμός

S = πρόσημο

KM = κλασματικό μέρος

E = εκθέτης

Π = πόλωση

Ποιο αριθμό του δεκαδικού συστήματος

Αντιπροσωπεύει ο δυαδικός αριθμός

**11000000101000000000000000000000**

Με βάση το πρότυπο IEEE 754;

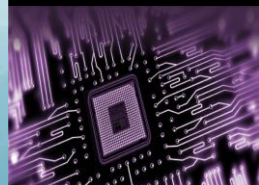
$$X = (-1)^1 \times (1+01_2) \times 2^{(129-127)}$$

$$= (-1) \times 1.25 \times 2^2$$

$$= -5.0$$

Μικροεξεργαστές

Αρχές & Εφαρμογές



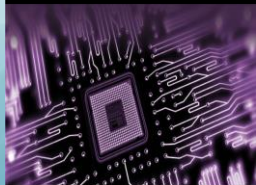
Παναγιώτης Παπάζογλου

Εκδόσεις Τζιόλα

# Κεφάλαιο 3

## Τα κύρια σημεία σε τίτλους

- ❑ Πρόσθεση δύο και τριών αριθμών του ενός bit
- ❑ Πρόσθεση πολλαπλής ακρίβειας
- ❑ Αφαίρεση, Πολλαπλασιασμός, Διαίρεση
- ❑ Προσημασμένοι ακέραιοι αριθμοί (π.χ. συμπλήρωμα ως προς 2)
- ❑ Πράξεις με συμπλήρωμα ως προς 2
- ❑ Αναπαράσταση αριθμών κινητής υποδιαστολής

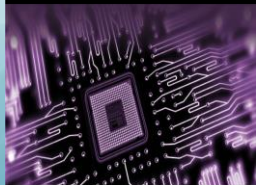


# Κεφάλαιο 4

## Βασικά ψηφιακά κυκλώματα

Μικροεξεργαστές

Αρχές & Εφαρμογές



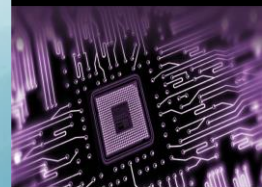
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Αναπτύσσοντας λογικές προτάσεις <sup>(1)</sup>

## Παράδειγμα 1

**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. Δηλαδή, η πρόταση Π1 θα είναι **αληθής** εφόσον 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  
**Ψευδής** (περίπτωση 2): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί **Ψευδής**  
(περίπτωση 3): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί **Αληθής**  
(περίπτωση 4): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί

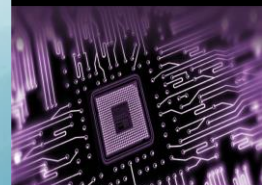


# Αναπτύσσοντας λογικές προτάσεις <sup>(2)</sup>

## Παράδειγμα 1

**Ψευδής** (περίπτωση 1): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί  
**Ψευδής** (περίπτωση 2): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί **Ψευδής**  
(περίπτωση 3): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί **Αληθής**  
(περίπτωση 4): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί

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



# Αναπτύσσοντας λογικές προτάσεις <sup>(3)</sup>

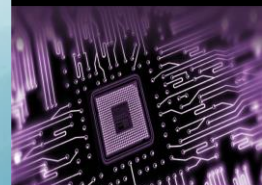
## Παράδειγμα 1

**Ψευδής** (περίπτωση 1): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί  
**Ψευδής** (περίπτωση 2): Αν ο υπολογιστής **ΔΕΝ λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί **Ψευδής**  
(περίπτωση 3): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **ΔΕΝ είναι** 43 βαθμοί **Αληθής**  
(περίπτωση 4): Αν ο υπολογιστής **λειτουργεί** 5 ώρες **ΚΑΙ** η θερμοκρασία **είναι** 43 βαθμοί

**A** = κατάσταση για τον υπολογιστή

**B** = κατάσταση για τη θερμοκρασία

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



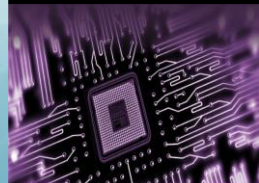
# Αναπτύσσοντας λογικές προτάσεις <sup>(4)</sup>

## Παράδειγμα 2

### Λογική πρόταση 2 (Π2)

- ❑ Π2α) Η ροή αέρα είναι υψηλή μόνο αν λειτουργούν ΚΑΙ οι δύο ανεμιστήρες
- ❑ Π2β) Η ροή αέρα είναι μέτρια μόνο αν λειτουργεί ΕΝΑΣ ΑΠΟ ΤΟΥΣ ΔΥΟ ανεμιστήρες

Π2α **Ψευδής** αν δεν λειτουργεί κανένας ανεμιστήρας ή λειτουργεί μόνο ένας από τους δύο  
Π2α **Αληθής** αν λειτουργούν και οι δύο ανεμιστήρες  
Π2β **Ψευδής** αν δεν λειτουργεί κανένας ανεμιστήρας ή λειτουργούν και οι δύο  
Π2β **Αληθής** αν δουλεύει μόνο ένας από τους δύο ανεμιστήρες





# Αναπτύσσοντας λογικές προτάσεις <sup>(5)</sup>

## Παράδειγμα 2

Π2α **Ψευδής** αν δεν λειτουργεί κανένας ανεμιστήρας ή λειτουργεί μόνο ένας από τους δύο

Π2α **Αληθής** αν λειτουργούν και οι δύο ανεμιστήρες

Π2β **Ψευδής** αν δεν λειτουργεί κανένας ανεμιστήρας ή λειτουργούν και οι δύο

Π2β **Αληθής** αν δουλεύει μόνο ένας από τους δύο ανεμιστήρες

### Λογικές προτάσεις και στάθμες

#### Λογική πρόταση

#### Λογική στάθμη (κατάσταση)

Ο ανεμιστήρας 1 **δεν** λειτουργεί

0

Ο ανεμιστήρας 1 **λειτουργεί**

1

Ο ανεμιστήρας 2 **δεν** λειτουργεί

0

Ο ανεμιστήρας 2 **λειτουργεί**

1

**A** = κατάσταση ανεμιστήρα 1

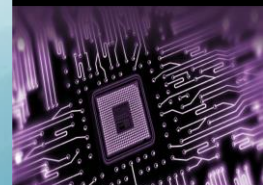
**B** = κατάσταση ανεμιστήρα 2

### Λογικές καταστάσεις για την πρόταση 2

A	B	Πρόταση Π2α	Πρόταση Π2β
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Μικροεπεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου

Εκδόσεις Τζόλα

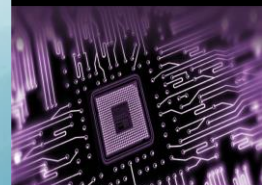
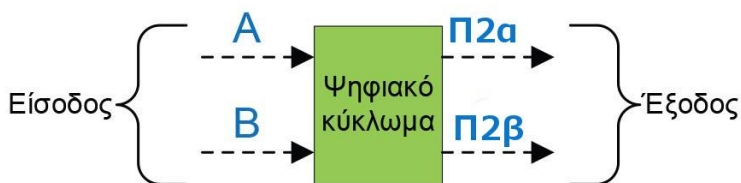
# Αναπτύσσοντας λογικές προτάσεις <sup>(6)</sup>

## Παράδειγμα 2

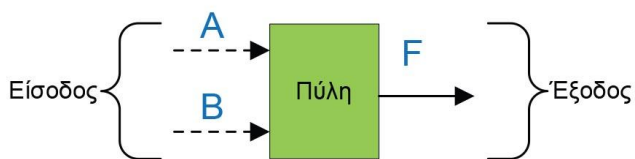
Λογικές καταστάσεις για την πρόταση 2

A	B	Πρόταση Π2α	Πρόταση Π2β
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- Πρέπει να σχεδιάσουμε ένα ψηφιακό κύκλωμα που να υλοποιεί τη λογική της πρότασης Π2
- Τι θα περιλαμβάνει το ψηφιακό κύκλωμα ;
- Θα πρέπει να χρησιμοποιήσουμε λογικές πύλες



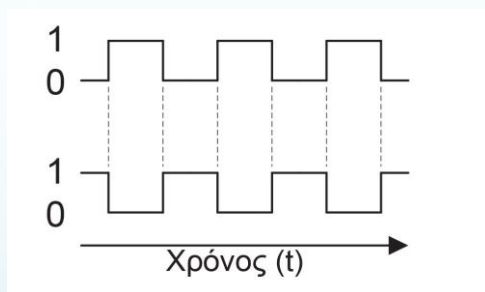
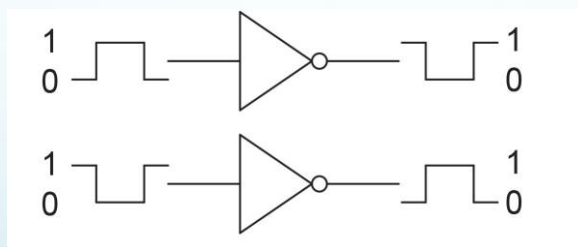
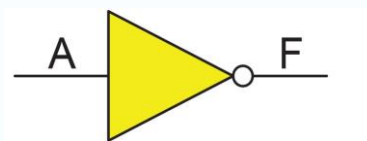
# Βασικές λογικές πύλες <sup>(1)</sup>



Κάθε πύλη υλοποιεί μια λογική πράξη

## Πύλη NOT

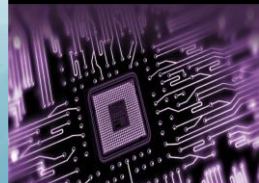
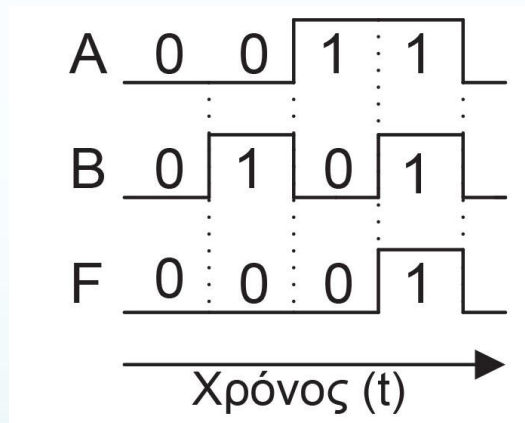
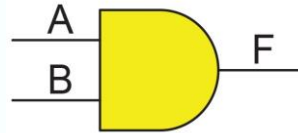
Πίνακας αληθείας της πύλης NOT	
Είσοδος (A)	Έξοδος (F)
0	1
1	0



# Βασικές λογικές πύλες (2)

## πύλη AND

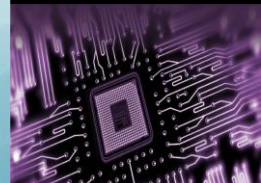
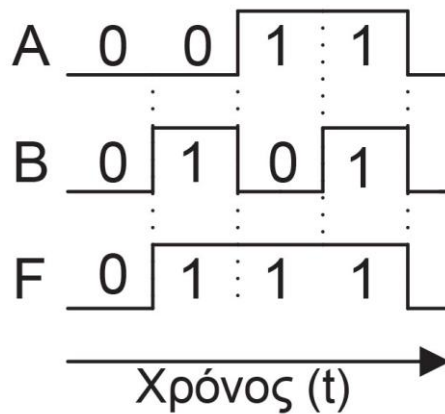
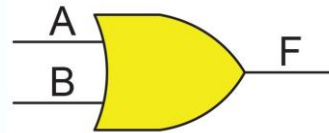
Πίνακας αληθείας πύλης AND		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



# Βασικές λογικές πύλες <sup>(3)</sup>

## πύλη OR

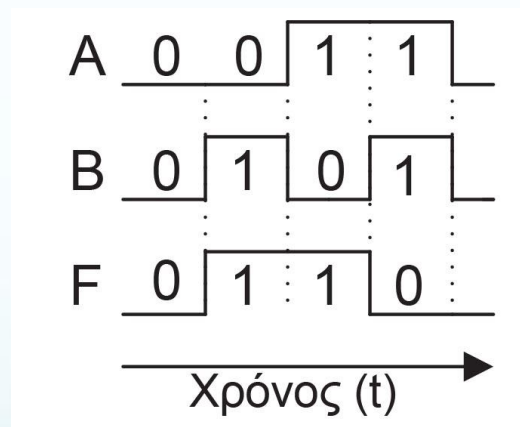
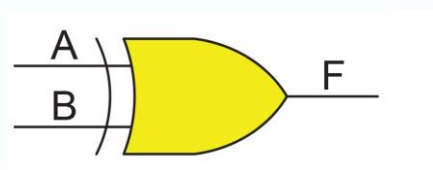
Πίνακας αληθείας πύλης OR		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



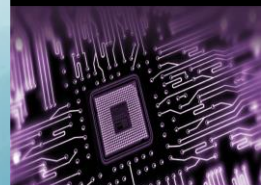
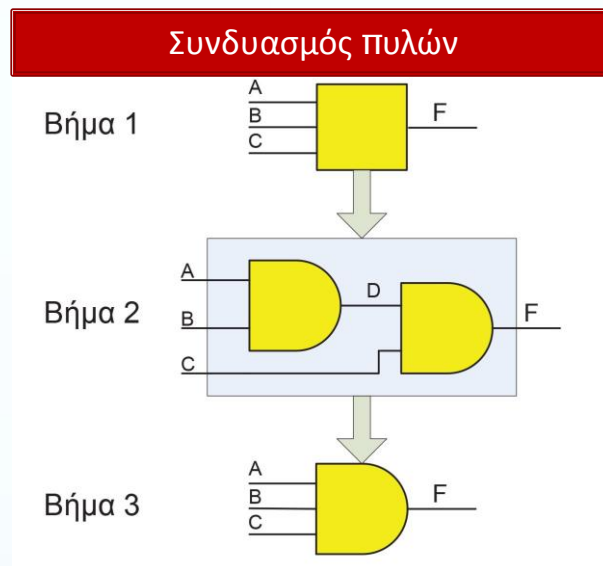
# Βασικές λογικές πύλες <sup>(4)</sup>

## Πύλη XOR

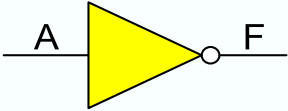
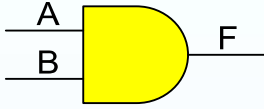
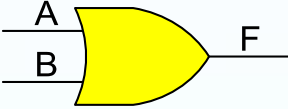
Πίνακας αληθείας πύλης XOR		
Είσοδος		Έξοδος
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

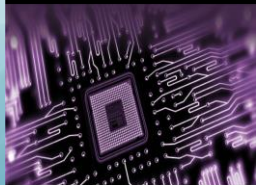


# Πύλες με Περισσότερες εισόδους



# 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 $F = AB$
	$F = A \text{ OR } B$	$F = A + B$





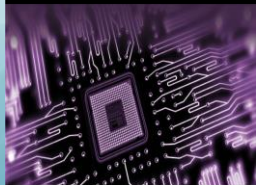
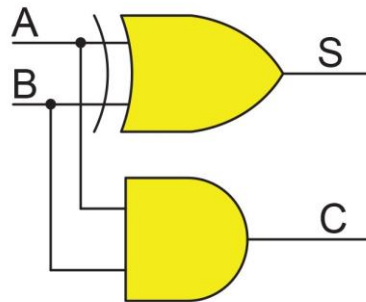
# Σχεδίαση απλών ψηφιακών κυκλωμάτων

## Παράδειγμα 1 - Ημιαθροιστής

Είσοδος		Έξοδος	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**S** = άθροισμα  
**C** = κρατούμενο

## Κύκλωμα

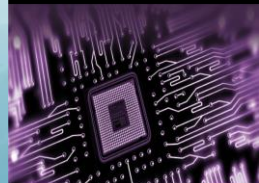
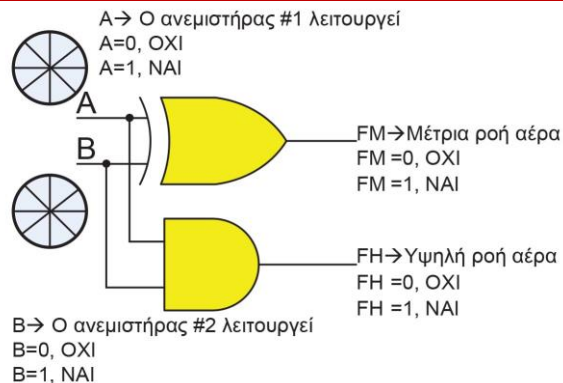


# Σχεδίαση απλών ψηφιακών κυκλωμάτων

Παράδειγμα 2 – Σύστημα ένδειξης ροής αέρα στον υπολογιστή

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

## Κύκλωμα

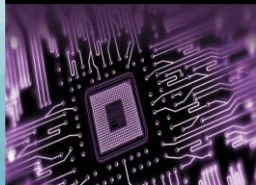
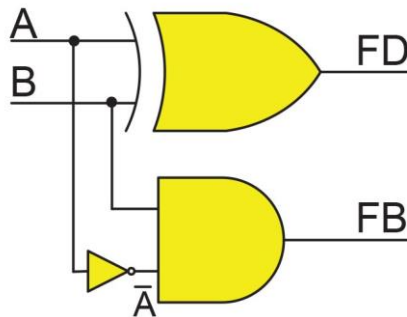


# Σχεδίαση απλών ψηφιακών κυκλωμάτων

## Παράδειγμα 3 – Ημιαφαιρέτης

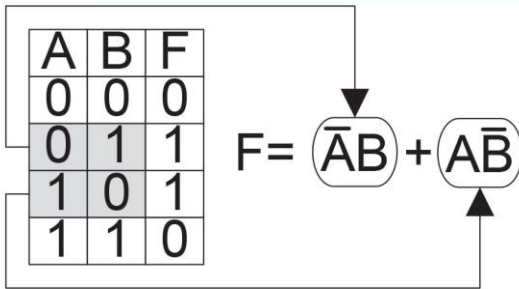
Είσοδος		Έξοδος	
A	B	A-B (FD)	Δανεικό (FB)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

## Κύκλωμα

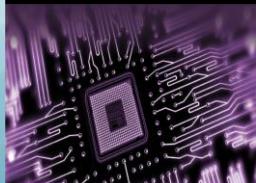
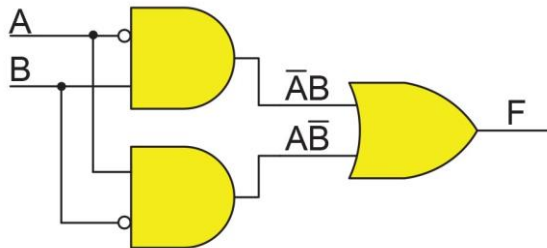


# Σχεδίαση απλών ψηφιακών κυκλωμάτων

## Παράδειγμα 4 – Υλοποίηση πύλης XOR

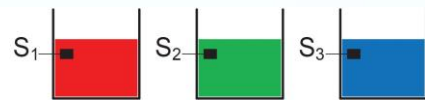


## Κύκλωμα

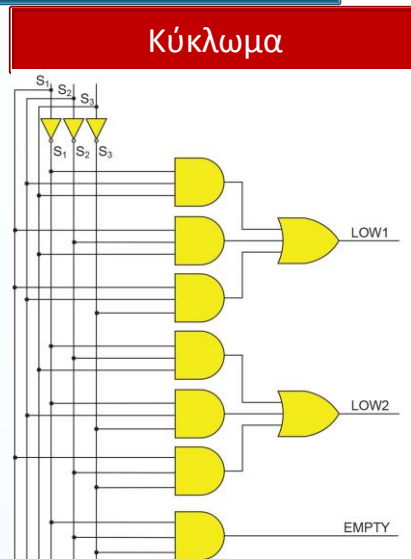


# Σχεδίαση απλών ψηφιακών κυκλωμάτων

Παράδειγμα 5 – Έλεγχος στάθμης σε δοχεία μελανιών



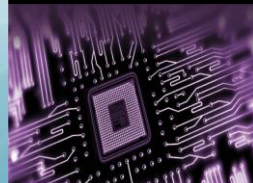
S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	LOW1	LOW2	EMPTY
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	0



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

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

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

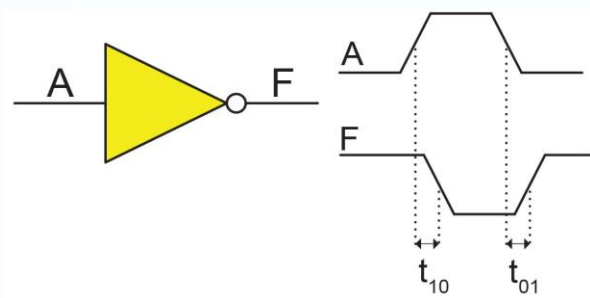


# Υλοποίηση ψηφιακών κυκλωμάτων

(1)

## Βασικά χαρακτηριστικά

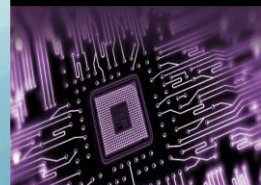
❑ χρονική απόκριση & καθυστέρηση διάδοσης



❑ κατανάλωση ενέργειας και δυνατότητα παροχής ρεύματος σε άλλες συνδεδεμένες διατάξεις

Μικροεξεργαστές

Αρχές & Εφαρμογές

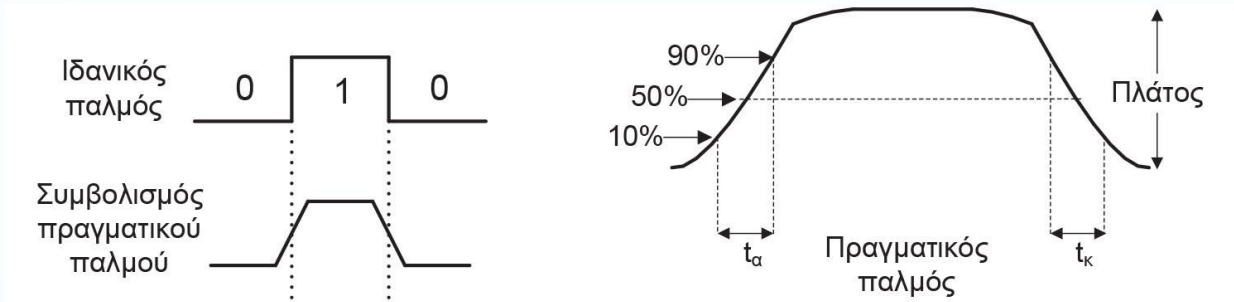


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

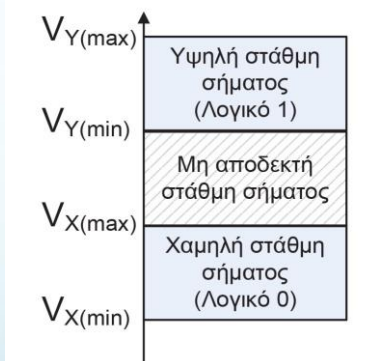
# Υλοποίηση ψηφιακών κυκλωμάτων

(2)

## Τα ψηφιακά σήματα στην πράξη

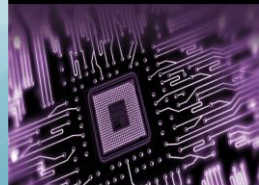


## Εύρος τιμών σήματος



Μικροεξεργαστές

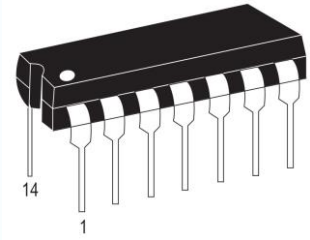
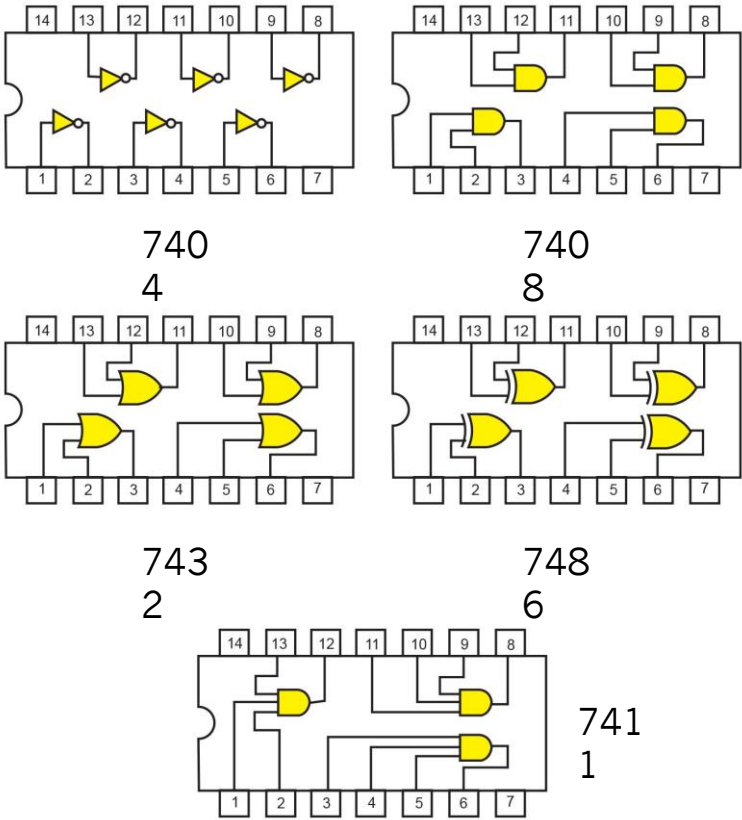
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

# Ψηφιακά ολοκληρωμένα κυκλώματα

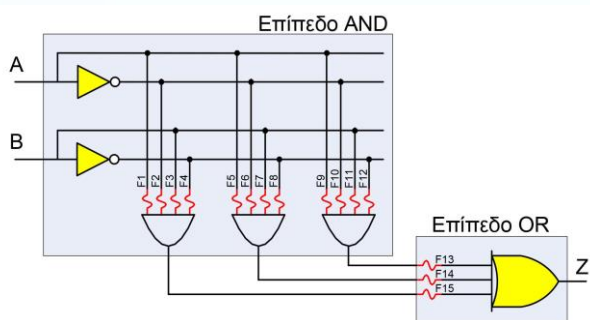
TTL



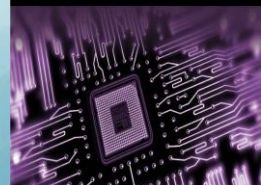
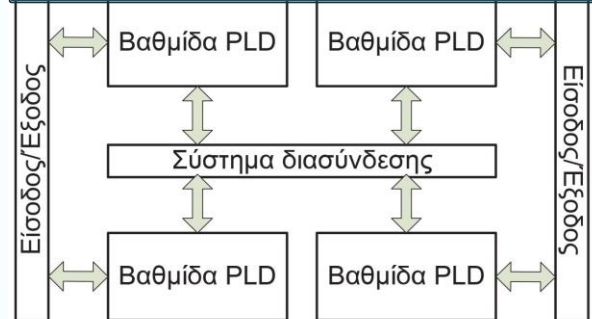


# Απλά προγραμματιζόμενα Ψηφιακά ολοκληρωμένα κυκλώματα (1)

## PLD - Programmable Logic Device

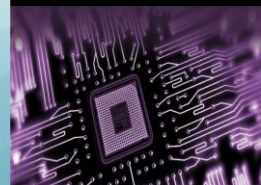
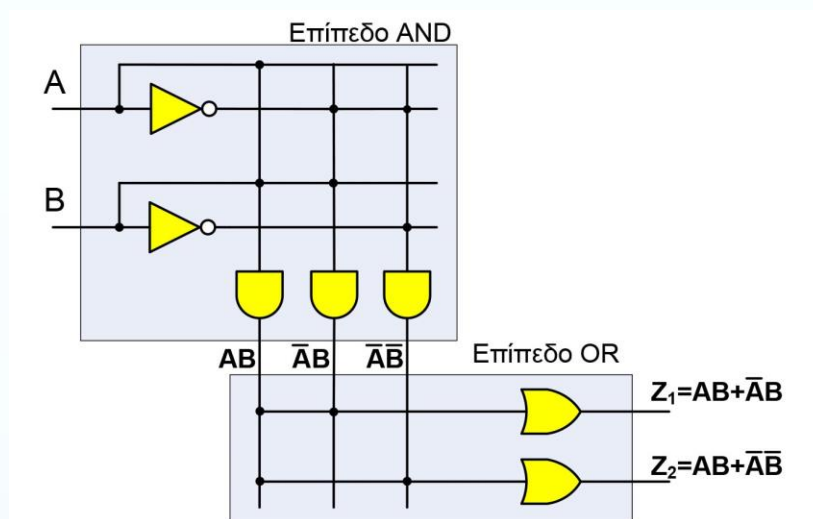


## CPLD - Complex PLD

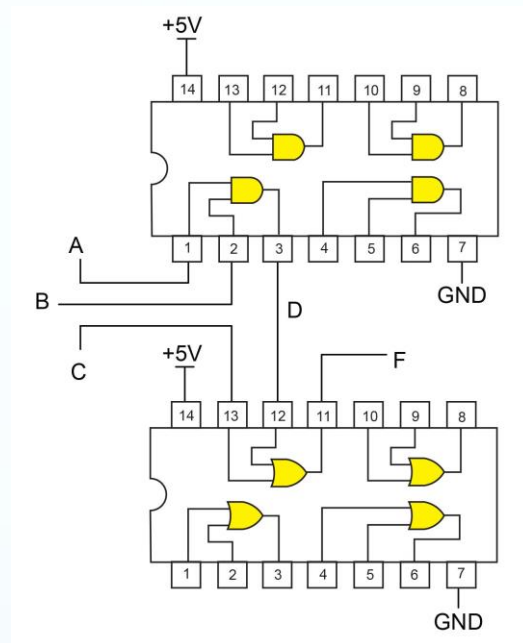
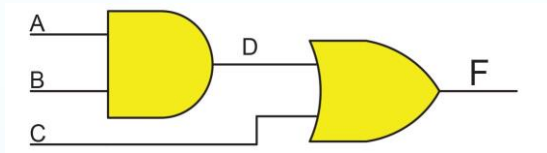


# Απλά προγραμματιζόμενα Ψηφιακά ολοκληρωμένα κυκλώματα (2)

Παράδειγμα υλοποίησης λογικών συναρτήσεων με PLD



# Υλοποίηση λογικών κυκλωμάτων με την τεχνολογία TTL

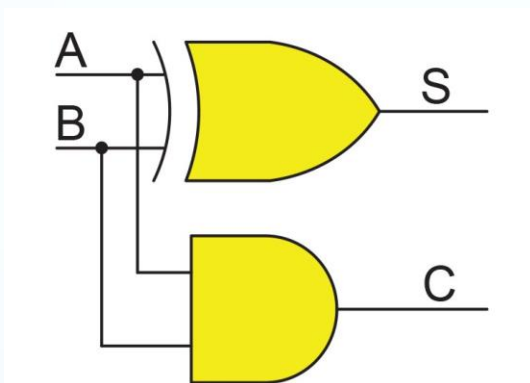


7408

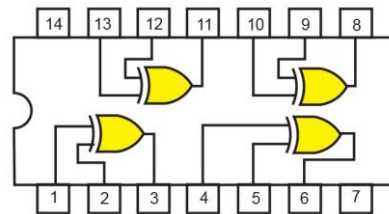
7408

# Πραγματικό πείραμα στο εργαστήριο <sup>(1)</sup>

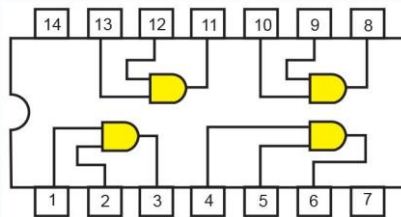
Ημιαθροιστής



Κύκλωμα που θα υλοποιήσουμε



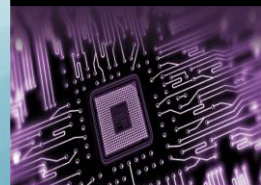
7486



7408

Ολοκληρωμένα  
κυκλώματα TTL που θα  
χρησιμοποιήσουμε

Μικροεξεργαστές  
Αρχές & Εφαρμογές

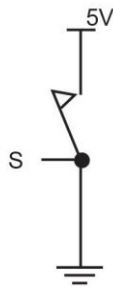


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζόλα

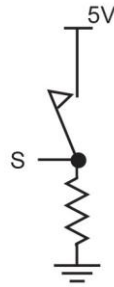
# Πραγματικό πείραμα στο εργαστήριο (2)

Ημιαθροιστής

Παραγωγή σημάτων εισόδου



ΛΑΘΟΣ

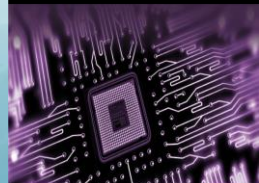


ΣΩΣΤΟ

Διακόπτης ανοιχτός, **S=0V**, λογικό 0  
Διακόπτης κλειστός, **S=5V**, λογικό 1

Μικροεξεργαστές

Αρχές & Εφαρμογές

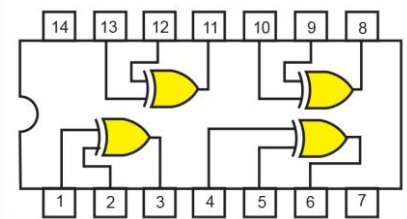


Παναγιώτης Παπάζογλου

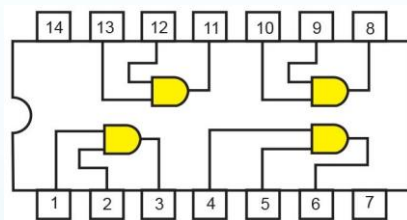
Εκδόσεις Τζάβλα

# Πραγματικό πείραμα στο εργαστήριο (3)

Ημιαθροιστής

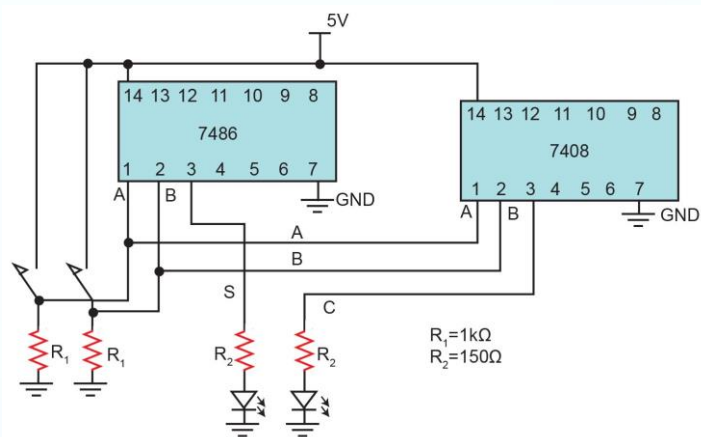


7486



7408

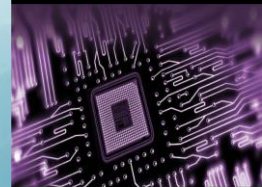
Τελικό κύκλωμα δοκιμής



Τα LED στην έξοδο  
δείχνουν τη στάθμη  
σήματος στα S (άθροισμα)  
και C (κρατούμενο)

Μικροεξεργαστές

Αρχές & Εφαρμογές

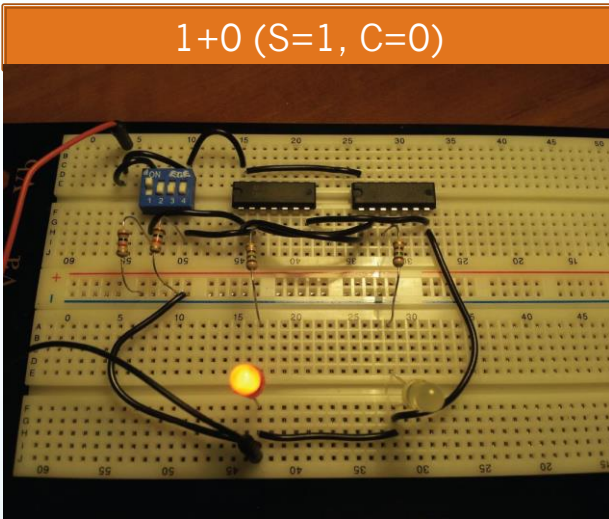


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάβλα

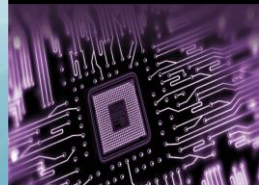
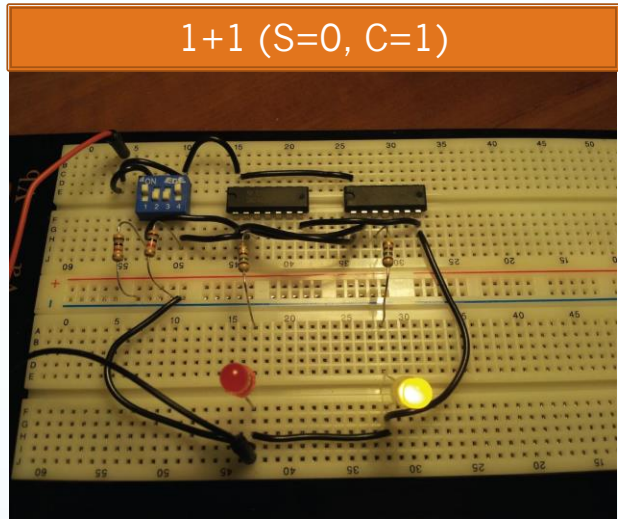
# Πραγματικό πείραμα στο εργαστήριο (4)

Ημιαθροιστής

1+0 (S=1, C=0)



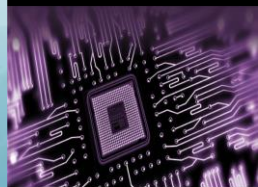
1+1 (S=0, C=1)



# Κεφάλαιο 4

## Τα κύρια σημεία σε τίτλους

- Λογικές καταστάσεις
- Λογικές πράξεις
- Βασικές Πύλες
- Συμβολισμός άλγεβρας Boole
- Ημιαθροιστής, ημιαφαιρέτης και άλλα κυκλώματα
- Υλοποίηση ψηφιακών κυκλωμάτων
- Ψηφιακά ολοκληρωμένα κυκλώματα TTL
- Κυκλώματα PLD-CPLD
- Πραγματικό Πείραμα στο εργαστήριο



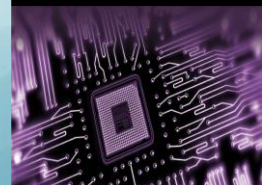


# Κεφάλαιο 5

## Εσωτερική οργάνωση του μικροεπεξεργαστή

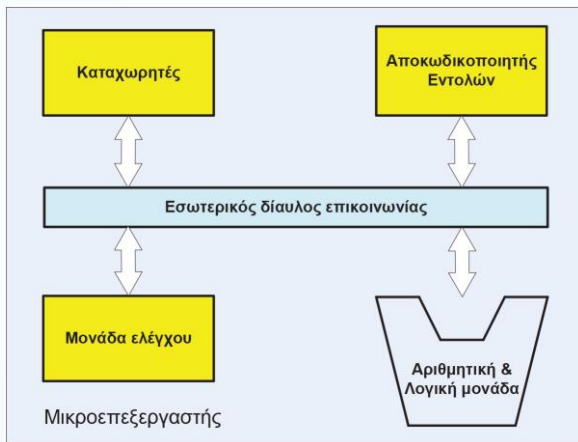
**Μικροεπεξεργαστές**

Αρχές & Εφαρμογές



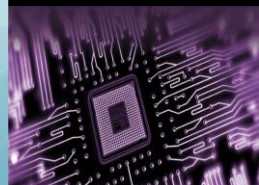
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Γενική δομή μικροεπεξεργαστή<sup>(1)</sup>

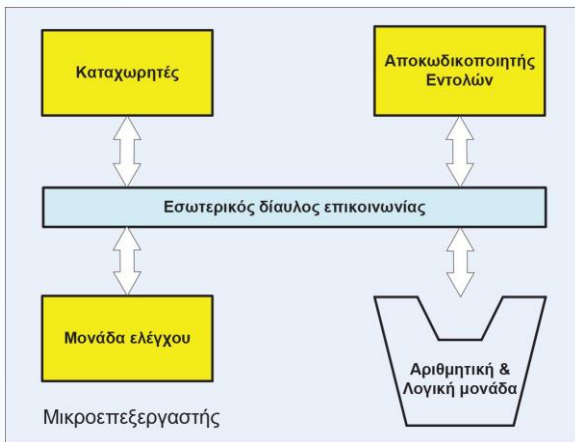


## Καταχωρητές

- ❑ Συγκεκριμένες περιοχές προσωρινής αποθήκευσης
- ❑ Συγκεκριμένο όνομα και χωρητικότητα
- ❑ Πλήθος και χωρητικότητα ανάλογα με την αρχιτεκτονική του μικροεπεξεργαστή
- ❑ Τυπική χωρητικότητα 8/16/32/64/128 bit
- ❑ Φιλοξενούν δεδομένα που μεταφέρονται από τη μνήμη στον μικροεπεξεργαστή και το αντίστροφο
- ❑ Χρησιμοποιούνται υποχρεωτικά για τη μεταφορά δεδομένων
- ❑ Γενικής και ειδικής χρήσης



# Γενική δομή μικροεπεξεργαστή<sup>(2)</sup>



## Μονάδα ελέγχου

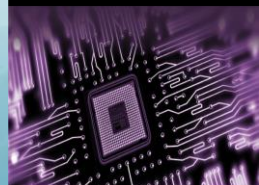
- ❑ Συντονισμός ενεργειών των μονάδων του μικροεπεξεργαστή προκειμένου να είναι δυνατή η εκτέλεση των εντολών
- ❑ Παραγωγή κατάλληλων σημάτων «ενεργοποίησης» (μέσω συστήματος εσωτερικών διαύλων)

## Αποκωδικοποιητής εντολών

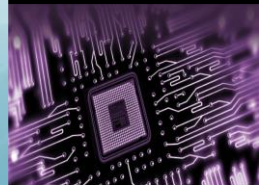
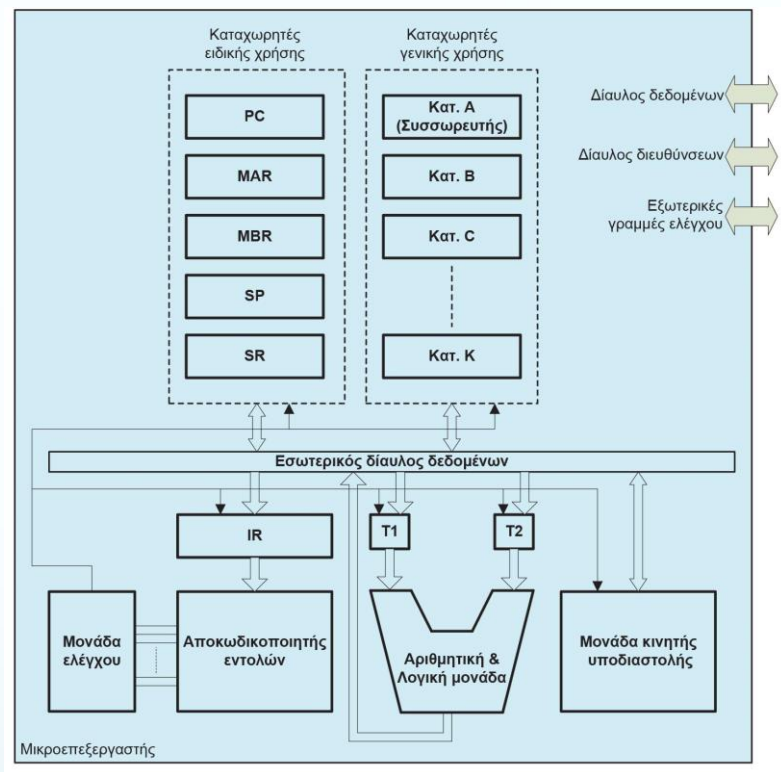
- ❑ «Υποδέχεται» μια εντολή όταν αυτή «έρθει» από τη μνήμη, ώστε στη συνέχεια να γνωρίζει ο μικροεπεξεργαστής τι ακριβώς θα πρέπει να κάνει αναφορικά με την εκτέλεσή της

## Αριθμητική & Λογική μονάδα

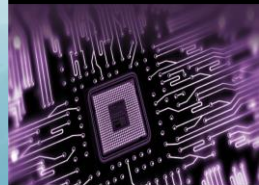
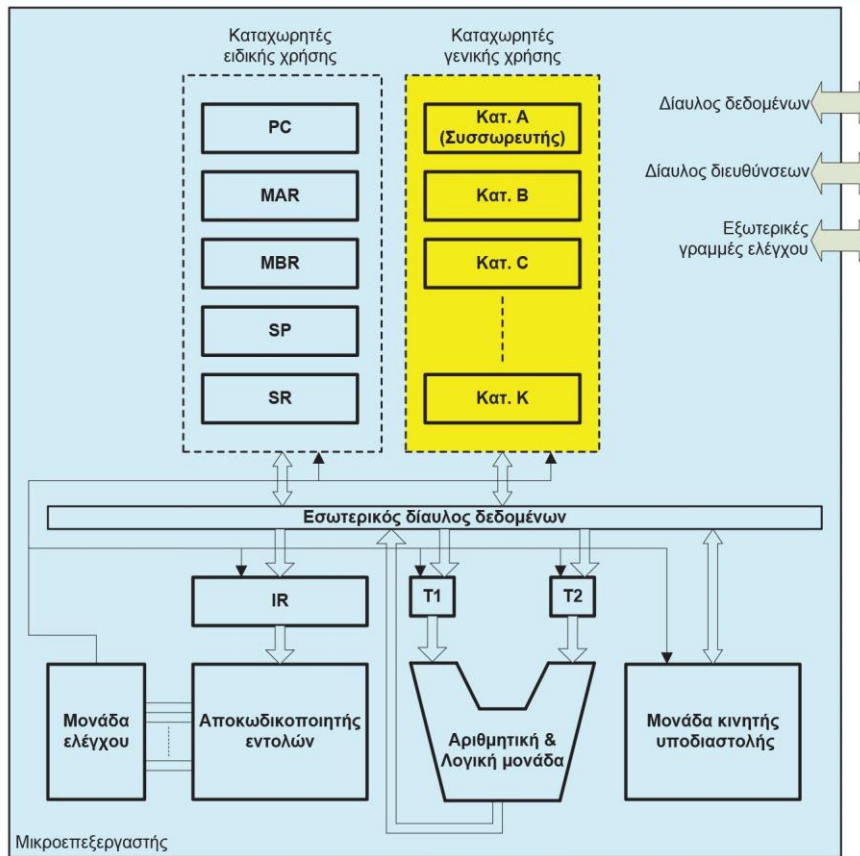
- ❑ Εκτέλεση αριθμητικών και λογικών πράξεων για τις εντολές του προγράμματος ή τις εσωτερικές λειτουργίες του μικροεπεξεργαστή



# Περισσότερα για το εσωτερικό του μικροεπεξεργαστή

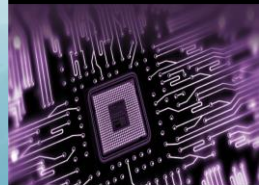
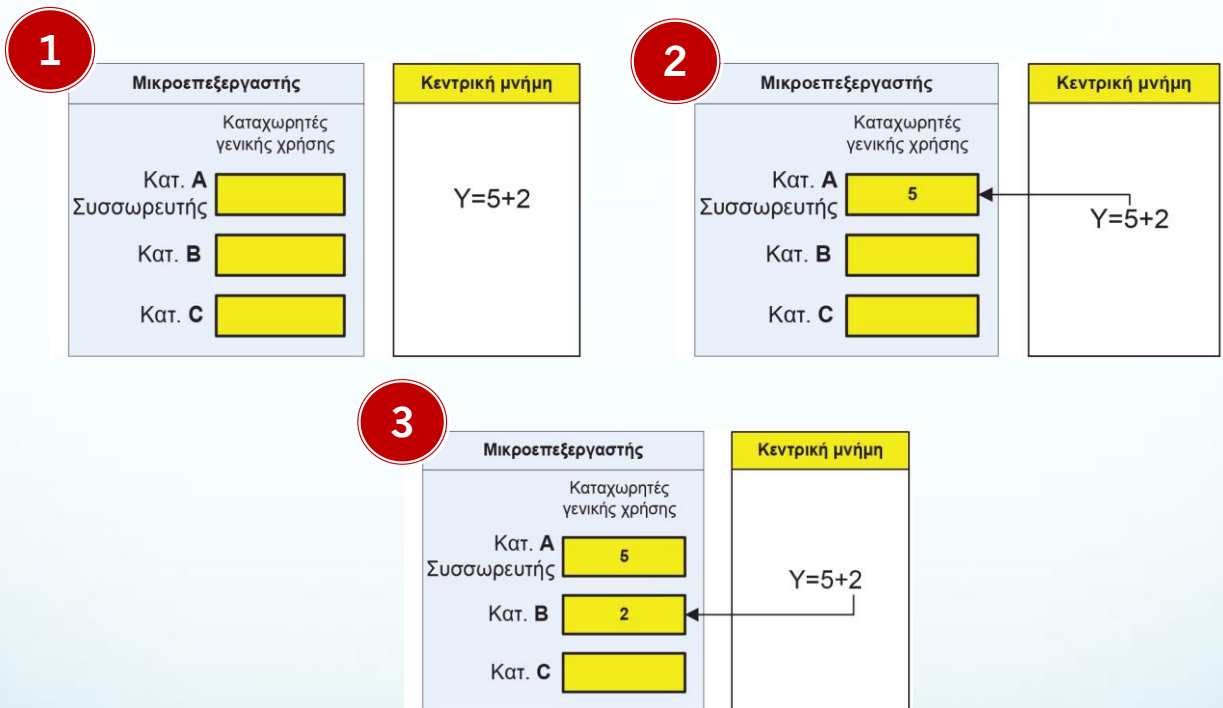


# Καταχωρητές γενικής χρήσης

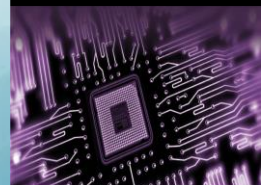


# Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών<sup>(1)</sup>

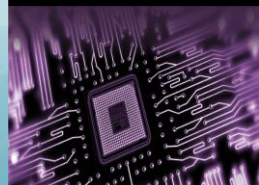
Ο προγραμματιστής έχει δημιουργήσει την εντολή  $Y=5+2$  (πρόσθεση των αριθμών 5 και 2)



# Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών<sup>(2)</sup>



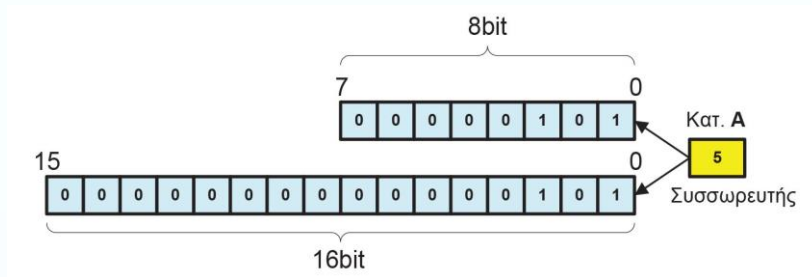
# Ένα γενικό παράδειγμα αριθμητικής πράξης με την εμπλοκή καταχωρητών<sup>(3)</sup>



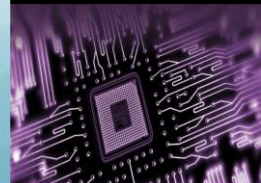
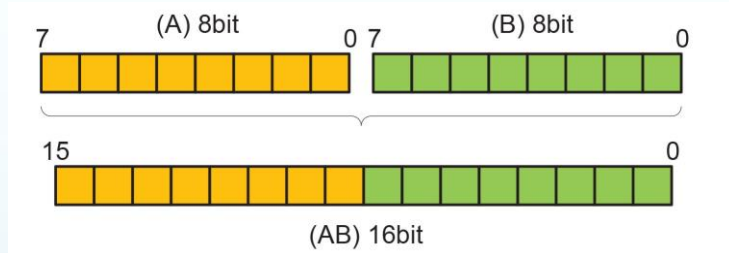


# Στο εσωτερικό των καταχωρητών

Ο καταχωρητής A ( $A=5$ ) του παραδείγματος (8 ή 16bit)

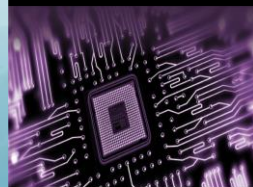


Συνδυασμός καταχωρητών για σύνθεση νέου μεγαλύτερης χωρητικότητας



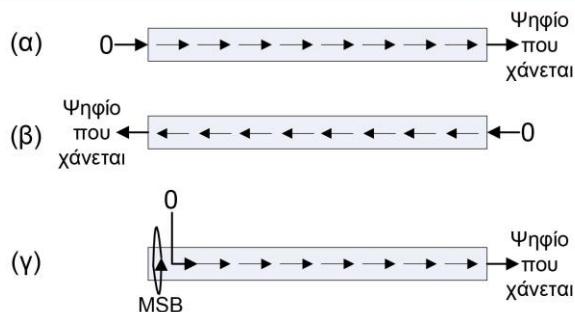
# Βασικές λειτουργίες καταχωρητών <sup>(1)</sup>

- Φόρτωση περιεχομένου (παράλληλα ή σειριακά)
- Ολίσθηση περιεχομένου

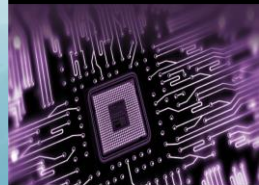
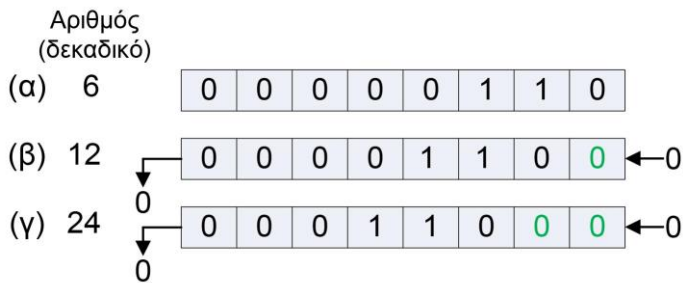


# Βασικές λειτουργίες καταχωρητών (2)

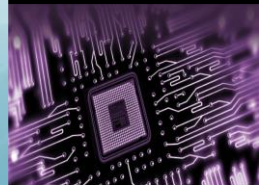
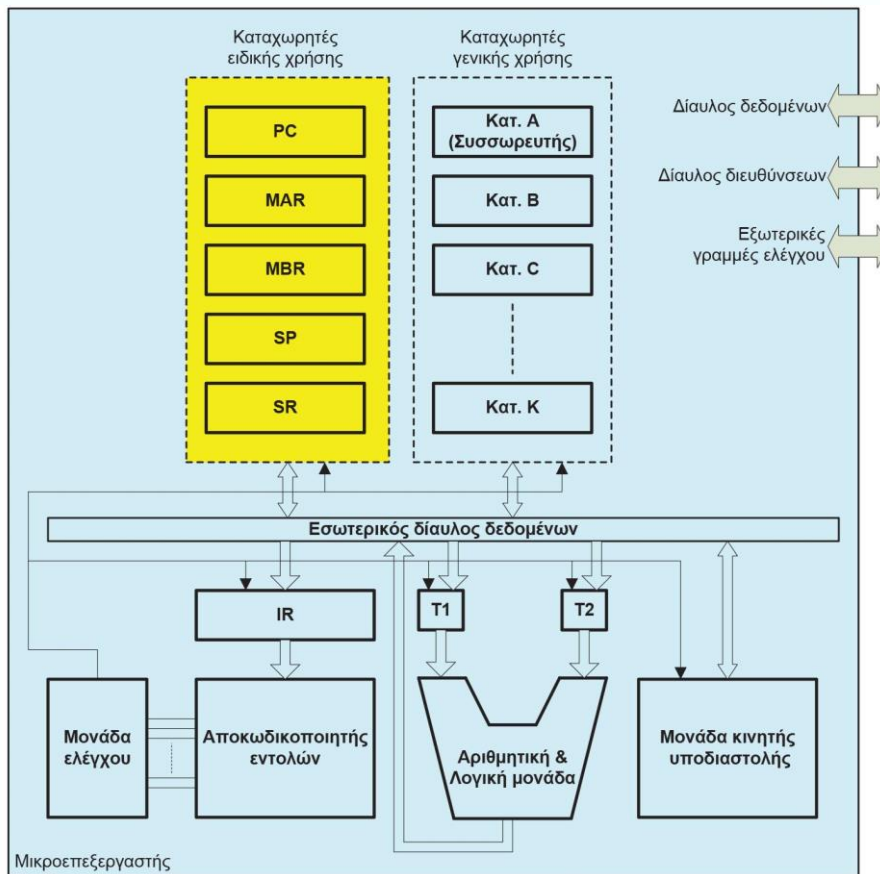
## Ολίσθηση περιεχομένου



## Πολλαπλασιασμός με ολίσθηση

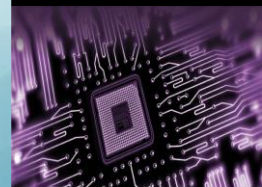


# Καταχωρητές ειδικής χρήσης



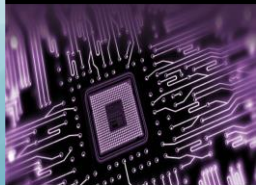
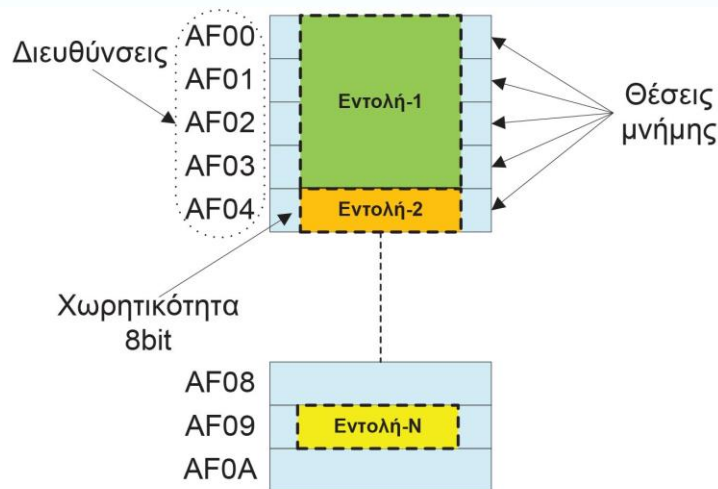
# PC – Program Counter (ΜΠ – Μετρητής Προγράμματος)

- ❑ Δείχνει τη διεύθυνση της επόμενης προς εκτέλεση εντολής
- ❑ Οι εντολές του προγράμματος βρίσκονται αποθηκευμένες στη μνήμη
- ❑ Εκτέλεση εντολής = προϋποθέτει την ανάκλησή της από τη μνήμη
- ❑ Ανάκληση = ο μικροεπεξεργαστής πρέπει να γνωρίζει σε ποια θέση μνήμης (διεύθυνση) ξεκινά η εντολή (μια εντολή μπορεί να καταλαμβάνει περισσότερες από μια θέσεις μνήμης)

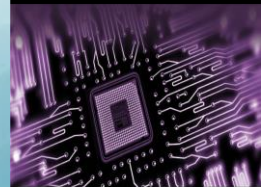
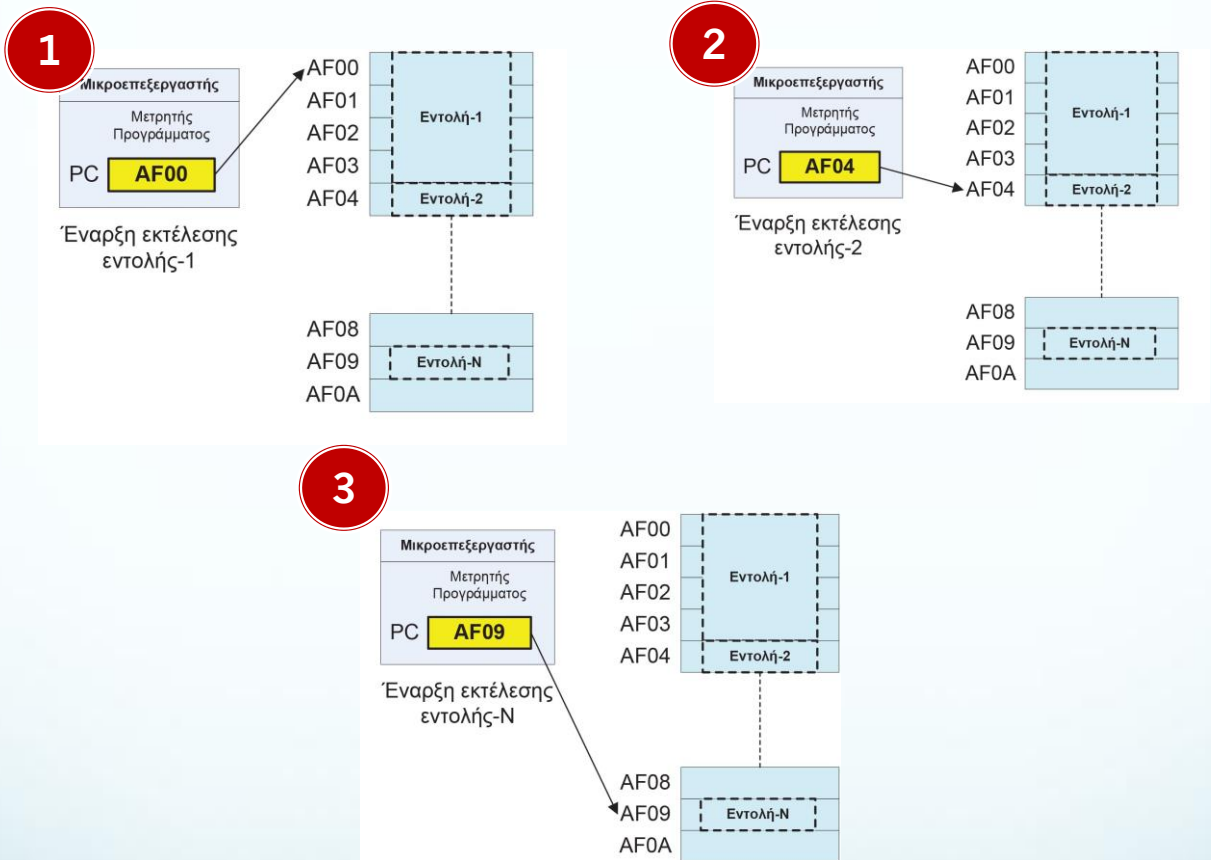


# Γενικό παράδειγμα εκτέλεσης εντολών (χρήση του ΜΠ)<sup>(1)</sup>

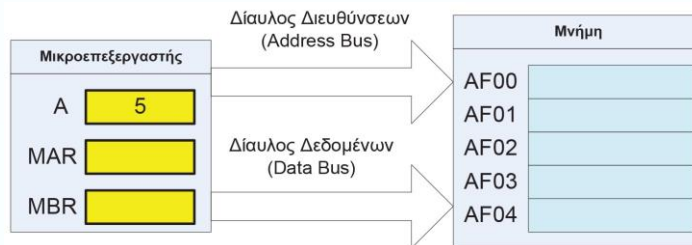
Υποθέτουμε ότι βρίσκονται N εντολές αποθηκευμένες στη μνήμη. Στο σχήμα φαίνονται οι διευθύνσεις στις οποίες ξεκινούν αλλά και οι θέσεις που καταλαμβάνουν.



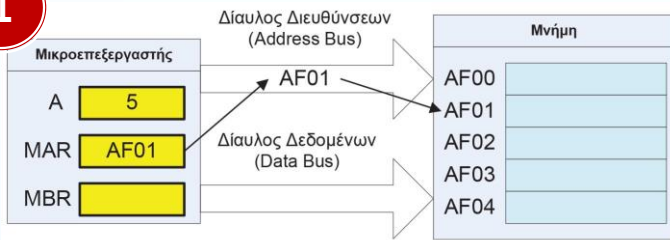
# Γενικό παράδειγμα εκτέλεσης εντολών (χρήση του ΜΠ)<sup>(2)</sup>



# MAR – Memory Access Register (ΚΔΜ – Καταχωρητής Διευθύνσεων Μνήμης) και MBR – Memory Buffer Register (ΠΚΔ – Προσωρινός Καταχωρητής Δεδομένων)

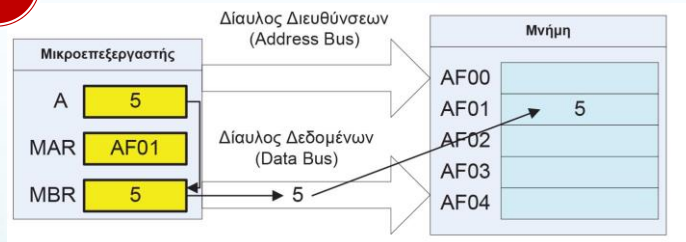


1



Αποθήκευση του αριθμού 5 στη θέση μνήμης AF01

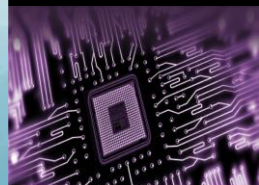
2



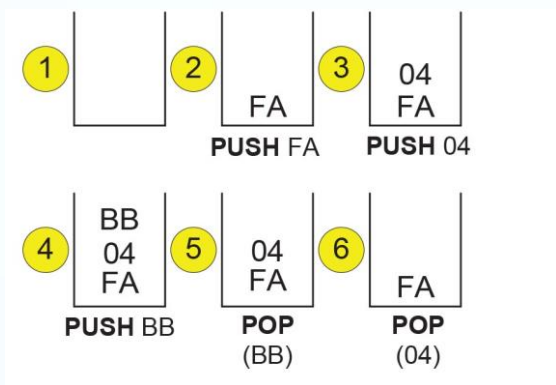


# SP – Stack Pointer (ΔΣ – Δείκτης Σωρού)

- ❑ Δείχνει πάντα την κορυφή του σωρού
- ❑ Σωρός = τρόπος οργάνωσης δεδομένων
- ❑ Υλοποίηση = στη μνήμη
- ❑ Πρόσβαση στο σωρό = στην κορυφή του
- ❑ Προώθηση – Ανάκτηση = Αυτό που προωθείται τελευταίο ανακτάται πρώτο
- ❑ Απλός τρόπος διαχείρισης



## Παράδειγμα χρήσης σωρού



**Βήμα 1:** Ο σωρός είναι κενός

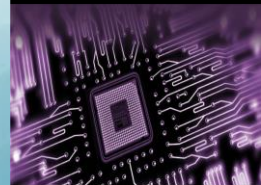
**Βήμα 2:** Αποθήκευση (ώθηση) του αριθμού FA στην κορυφή του σωρού (διαδικασία PUSH)

**Βήμα 3:** Αποθήκευση (ώθηση) του αριθμού 04 στην κορυφή του σωρού (διαδικασία PUSH)

**Βήμα 4:** Αποθήκευση (ώθηση) του αριθμού BB στην κορυφή του σωρού (διαδικασία PUSH)

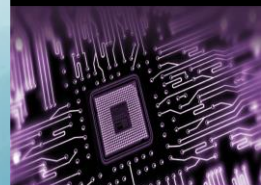
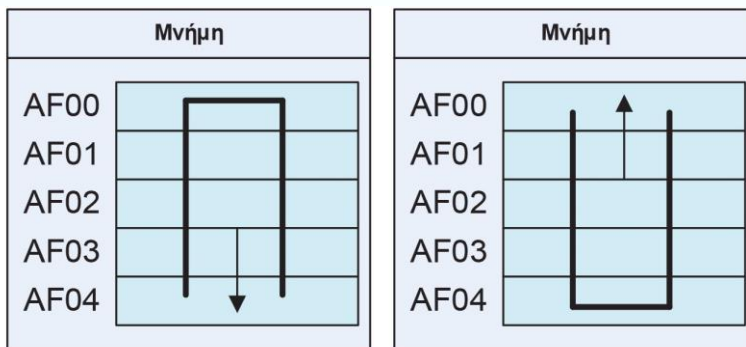
**Βήμα 5:** Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός BB, διαδικασία POP)

**Βήμα 6:** Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός 04, διαδικασία POP)

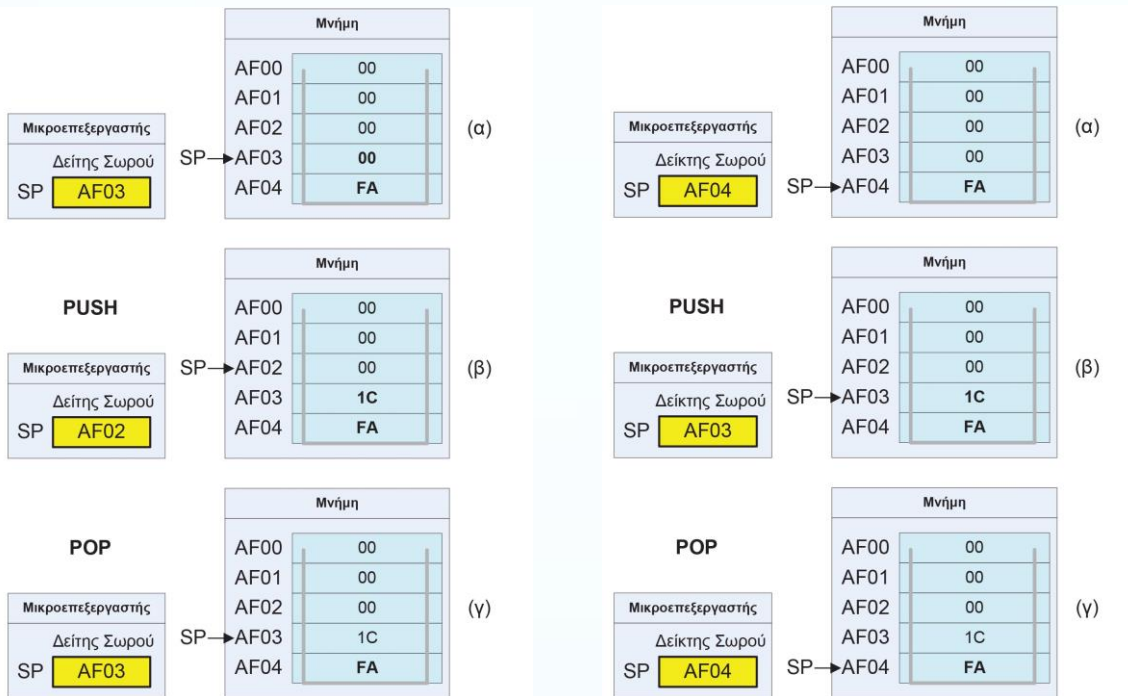


# Υλοποίηση του σωρού

Προς τις υψηλές ή χαμηλές διευθύνσεις

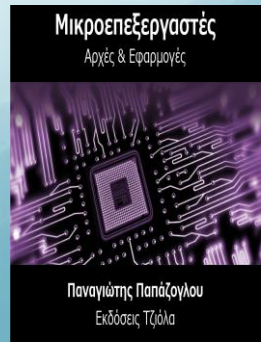


# Τρόποι διαχείρισης του σωρού



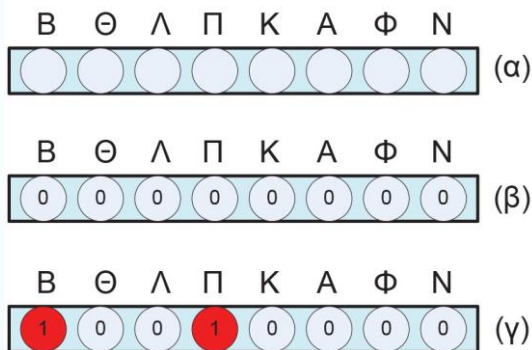
**Δείκτης σωρού σε κενή θέση**  
 Όθηση: αποθήκευση,  $\Delta\Sigma = \Delta\Sigma - 1$   
 Ανάκτηση:  $\Delta\Sigma = \Delta\Sigma + 1$ , ανάκτηση

**Δείκτης σωρού σε δεδομένα**  
 Όθηση:  $\Delta\Sigma = \Delta\Sigma - 1$ , αποθήκευση  
 Ανάκτηση: ανάκτηση,  $\Delta\Sigma = \Delta\Sigma + 1$



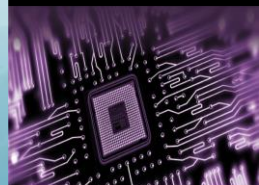
# SR – Status Register (ΚΚ – Καταχωρητής Κατάστασης)<sup>(1)</sup>

- ❑ Αποτελείται από bit που λειτουργούν ανεξάρτητα μεταξύ τους
- ❑ Τα bit δείχνουν ανά πάσα στιγμή την «κατάσταση» του μικροεπεξεργαστή
- ❑ Πληροφορίες για κρατούμενο, υπερχείλιση, πρόσημο, κλπ



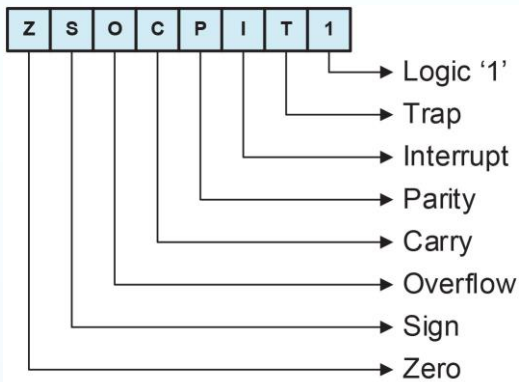
- (α) Ενδεικτικά λαμπάκια
- (β) Δεν υπάρχει κάποια ειδοποίηση
- (γ) Υπάρχουν δύο ειδοποιήσεις (B, Π)

- ❑ Ας φανταστούμε τα ενδεικτικά λαμπάκια του αυτοκινήτου
- ❑ Κάθε ένα από αυτά λειτουργεί ανεξάρτητα και μας δίνει διαφορετική πληροφορία
- ❑ Στο διπλανό σχήμα, οι ενδείξεις είναι:  
**B**: Βενζίνη (χαμηλή στάθμη), **Θ**: Θερμοκρασία (υψηλή), **Λ**: Λάδια (χαμηλή στάθμη), **Π**: Πόρτες (ανοιχτές), **Κ**: Καυσαέρια (κακή ποιότητα), **Α**: Αερόσακος (δυσλειτουργία), **Φ**: Φρένα (χαμηλή πίεση), **Ν**: Νερό (χαμηλή στάθμη)



# SR – Status Register (ΚΚ – Καταχωρητής Κατάστασης)<sup>(2)</sup>

Συνηθισμένη δομή καταχωρητή κατάστασης στους μικροεπεξεργαστές



**Zero:** έχει προκύψει μηδέν από μια αριθμητική πράξη (π.χ. μείωση του Περιεχομένου ενός καταχωρητή)

**Sign:** όταν το πρώτο bit ενός αριθμού γίνει 1 (σε προσημασμένους αριθμούς)

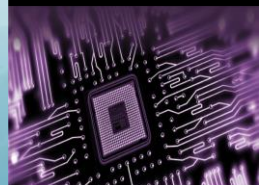
**Overflow:** όταν ένας αριθμός δεν χωρά να αποθηκευτεί στη θέση προορισμού (π.χ. και σε υπερχειλίση αριθμητικής με συμπλήρωμα ως προς 2)

**Carry:** έχει προκύψει κρατούμενο από πράξη

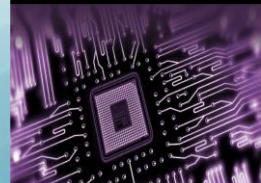
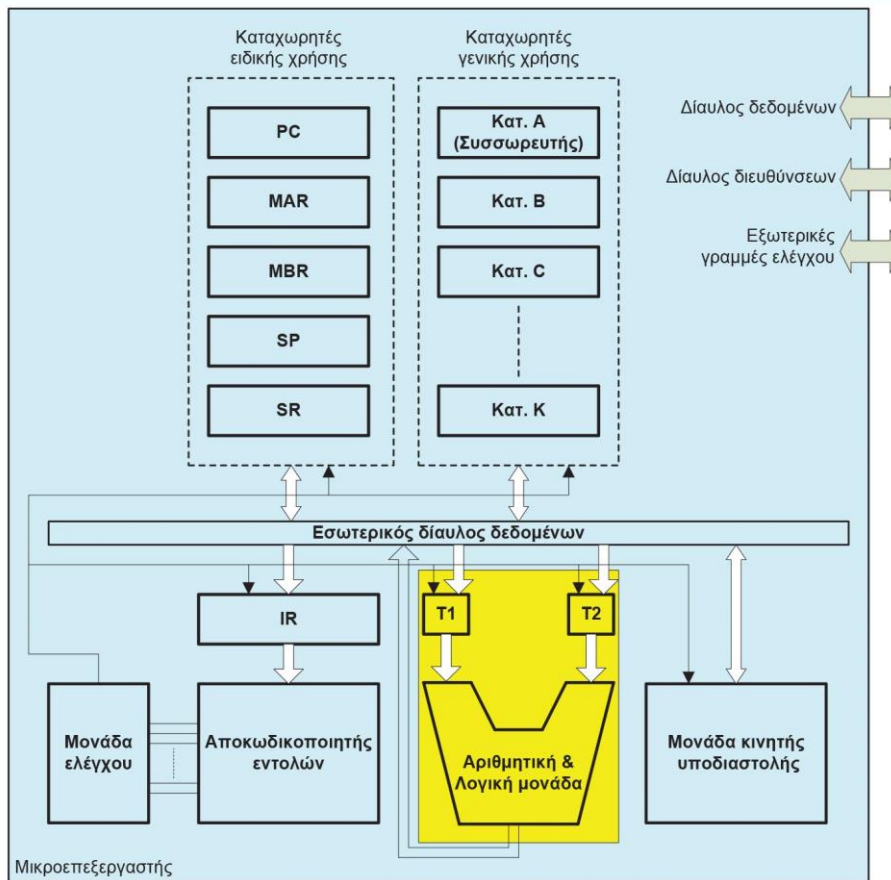
**Parity:** ισοτιμία των bit ενός αριθμού (π.χ. περιττό πλήθος ψηφίων με την τιμή 1)

**Interrupt:** επιτρέπονται ή όχι οι διακοπές

**Trap:** χρήση για βήμα προς βήμα αποσφαλμάτωση **1:** λογικό '1' ως τιμή αναφοράς



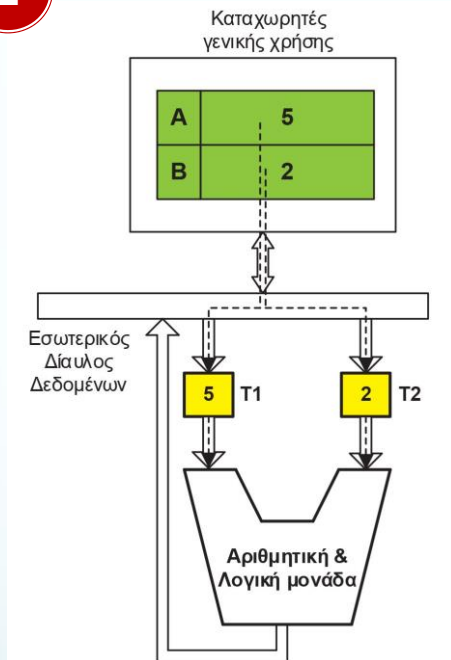
# Αριθμητική και λογική μονάδα



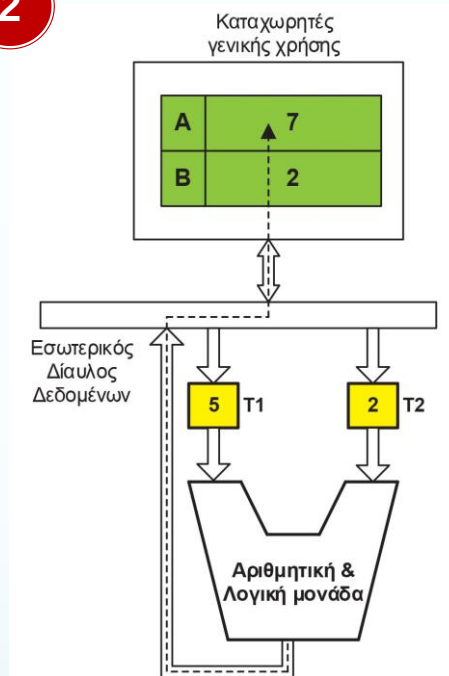
# Αριθμητική και λογική μονάδα

Φιλοσοφία λειτουργίας για την πρόσθεση 5+2

1

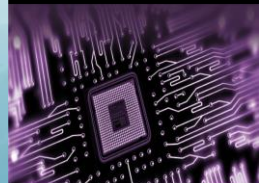


2



Μικροεξεργαστές

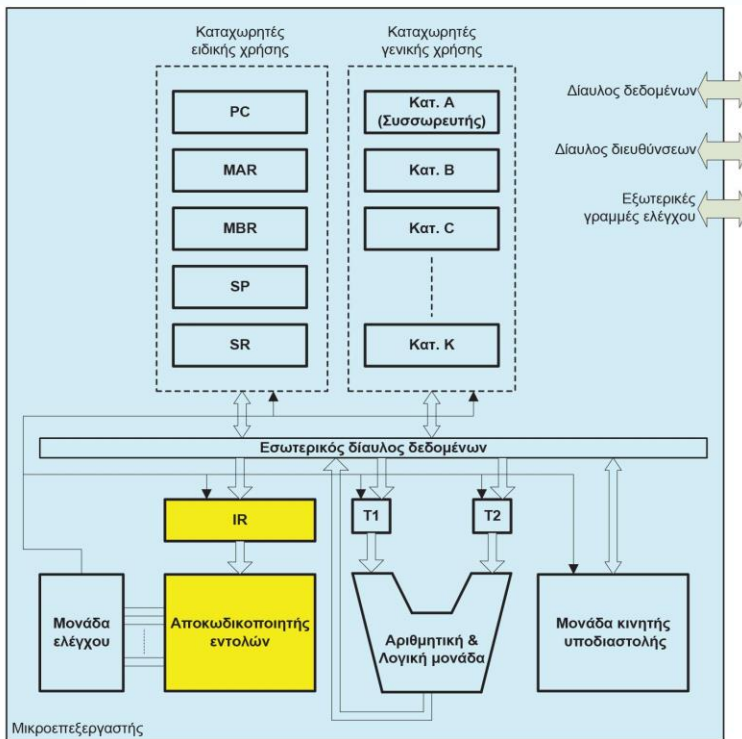
Αρχές & Εφαρμογές



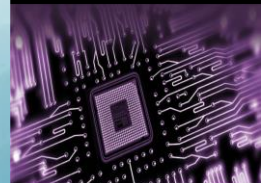
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα



# Αποκωδικοποιητής εντολών



- ❑ Όταν πρόκειται να εκτελεστεί μια εντολή, μεταφέρεται ο αντίστοιχος κώδικας (σε ορισμένες περιπτώσεις μαζί με τα ορίσματα από την πρώτη «μεταφορά») από τη μνήμη στον ΚΕ (IR-Instruction Register)
- ❑ Ο αποκωδικοποιητής δέχεται ως είσοδο το περιεχόμενο του ΚΕ και «δίνει οδηγίες» για την εκτέλεση της εντολής



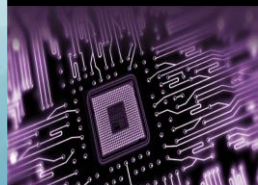
# Άλλες μονάδες

## Μονάδα ελέγχου

- ❑ Συντονίζει με τα κατάλληλα σήματα τη λειτουργία όλων των μονάδων του μικροεπεξεργαστή ώστε να είναι δυνατή η εκτέλεση των εντολών.
- ❑ Ο συντονισμός αυτός εστιάζεται κυρίως στον έλεγχο της ροής δεδομένων μεταξύ των μονάδων του μικροεπεξεργαστή μέσω του εσωτερικού διαύλου δεδομένων
- ❑ Ελέγχει το χρονισμό των μονάδων με την αξιοποίηση του ρολογιού
- ❑ Δέχεται πληροφορίες από τον αποκωδικοποιητή

## Μονάδα κινητής υποδιαστολής

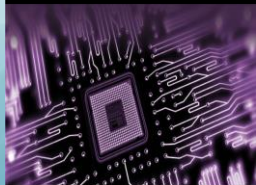
- ❑ Παλαιοί μικροεπεξεργαστές = αριθμητικές πράξεις μέσω λογισμικού, αργότερα από ξεχωριστό ολοκληρωμένο κύκλωμα (συνεπεξεργαστής)
- ❑ Σύγχρονοι μικροεπεξεργαστές = την περιλαμβάνουν στο εσωτερικό τους, διαθέτει δικές της εντολές και καταχωρητές, ενώ απαιτεί πιο «ειδικό» προγραμματισμό



# Κεφάλαιο 5

## Τα κύρια σημεία σε τίτλους

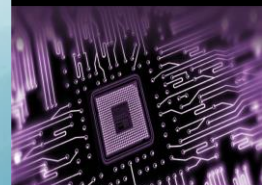
- Ο μικροεπεξεργαστής
- Καταχωρητές
- Καταχωρητές γενικής χρήσης
- Καταχωρητές ειδικής χρήσης
- Ο σωρός
- Καταχωρητής Κατάστασης
- Αριθμητική και Λογική Μονάδα
- Αποκωδικοποιητής εντολών
- Άλλες μονάδες



# Κεφάλαιο 6

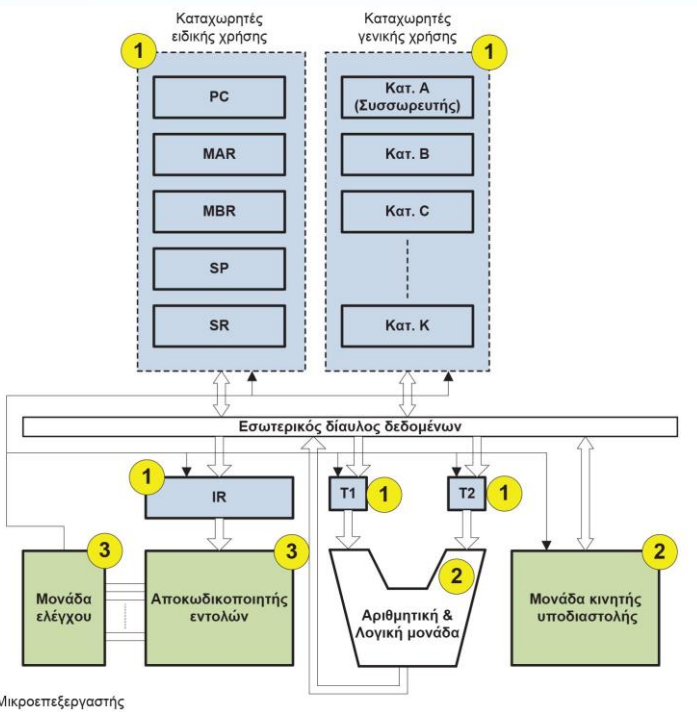
## Βασικά ψηφιακά κυκλώματα μικροεπεξεργαστή

Μικροεπεξεργαστές  
Αρχές & Εφαρμογές

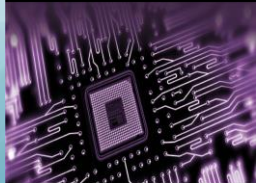


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

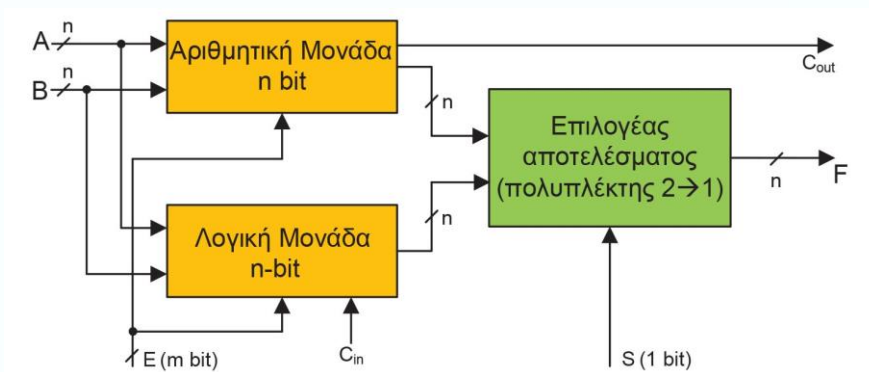
# Κατηγορίες ψηφιακών κυκλωμάτων μικροεπεξεργαστή



- 1 Στοιχείο μνήμης (κύκλωμα μνήμης)
- 2 Στοιχείο υπολογισμού (αριθμητικά και λογικά κυκλώματα)
- 3 Ειδικά στοιχεία (εξειδικευμένα και προσαρμοσμένα κυκλώματα)



# Δομή αριθμητικής και λογικής μονάδας



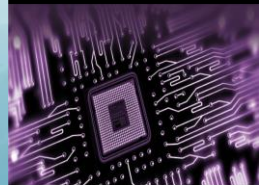
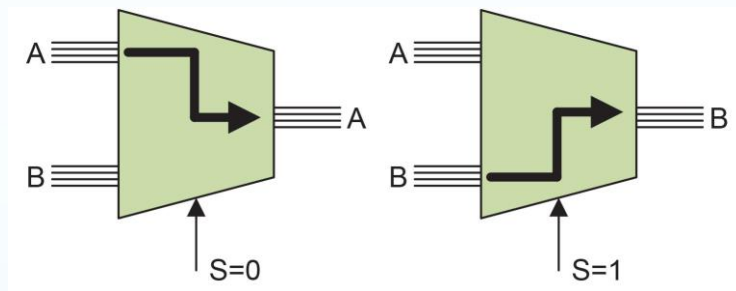
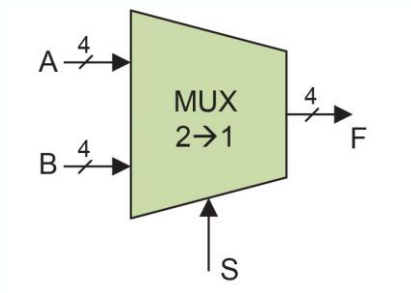
**Έλεγχος λειτουργίας**

Επιλογή πράξης		Εκτέλεση πράξης	
E1	E0	Αριθμητική	Λογική
0	0	A+B	A OR B
0	1	A-B	A AND B
1	0	A+1	NOT A
1	1	B+1	NOT B

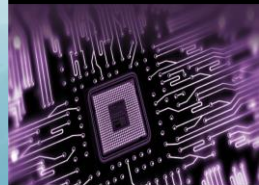
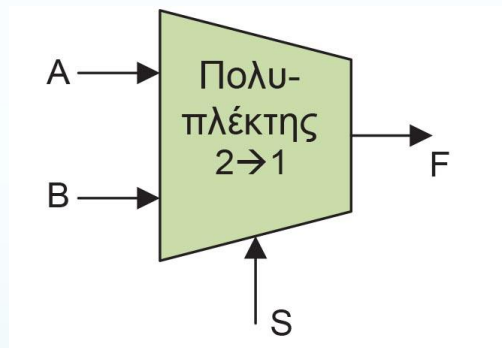
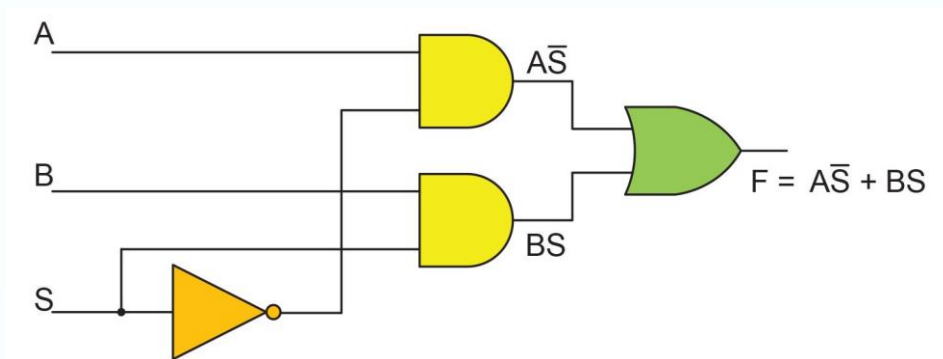
**Επιλογή αποτελέσματος**

S	F (είδος αποτελέσματος)
0	Αριθμητικό
1	Λογικό

# Λειτουργία πολυπλέκτη

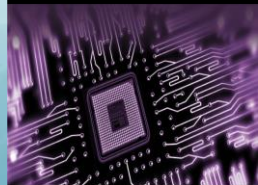
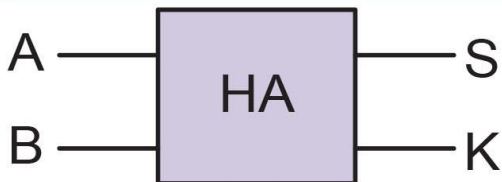


## Υλοποίηση πολυπλέκτη





## Ημιαθροιστής και πλήρης αθροιστής

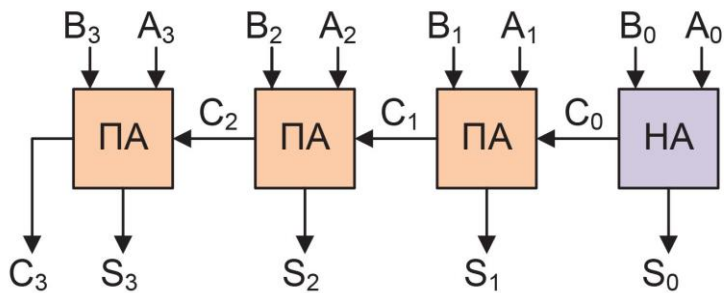


# Άθροισμα δύο αριθμών των 4bit <sup>(1)</sup>

Συνδυασμός ψηφίων για πρόσθεση

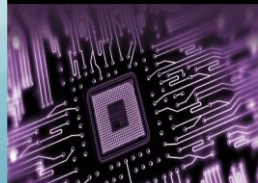
$$\begin{array}{rcccc} C_3 & C_2 & C_1 & C_0 \\ \downarrow & & & & \\ & A_3 & A_2 & A_1 & A_0 \\ & B_3 & B_2 & B_1 & B_0 & + \\ \hline C_3 & S_3 & S_2 & S_1 & S_0 \end{array}$$

Παράλληλος αθροιστής



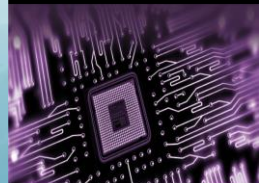
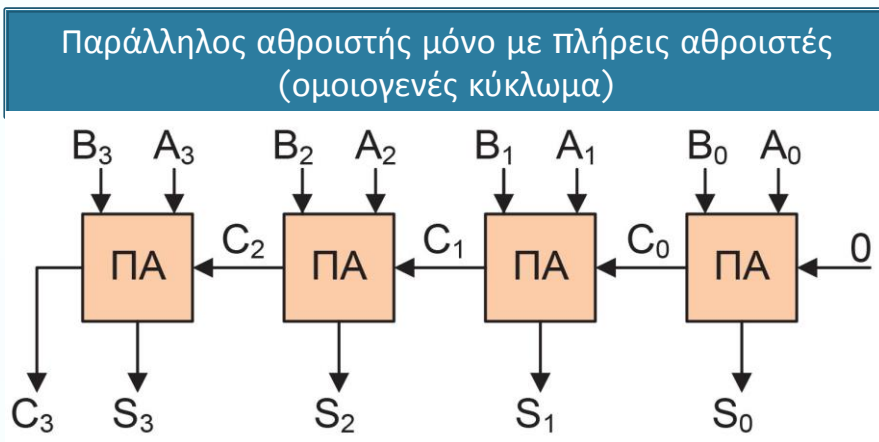
Μικροεξεργαστές

Αρχές & Εφαρμογές

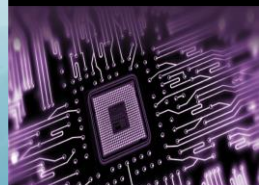
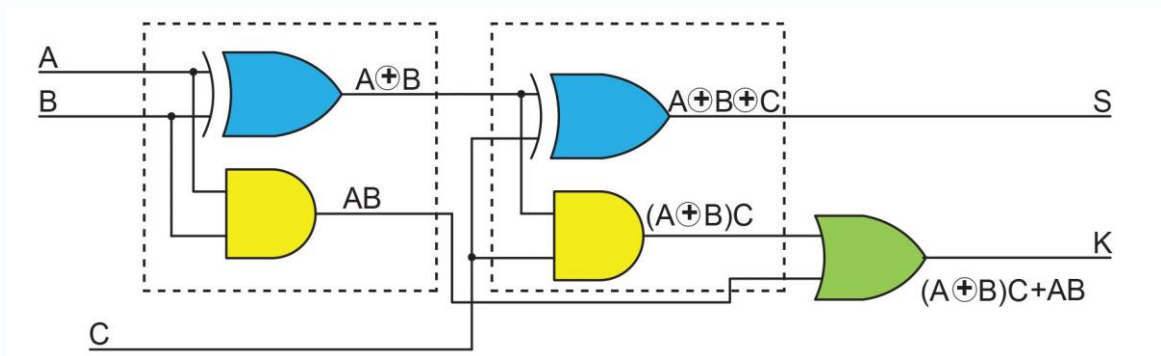


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

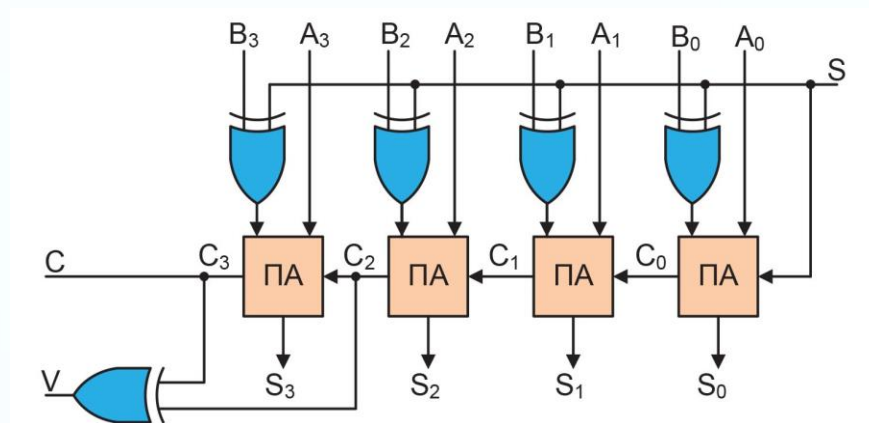
## Άθροισμα δύο αριθμών των 4bit (2)



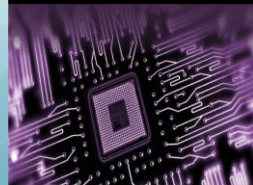
## Υλοποίηση στοιχείου πλήρη αθροιστή



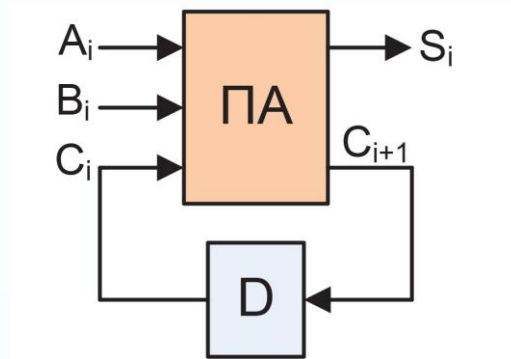
# Κύκλωμα πρόσθεσης-αφαίρεσης



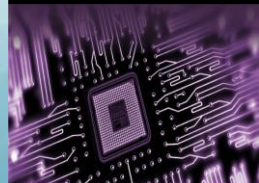
- ❑ **S = 0**, Πρόσθεση,  $A+B$
- ❑ **S = 1**, Αφαίρεση,  $A-B = A+(-B) = A+\Sigma_2(B)$
- ❑ Για  $S=1$  αντιστρέφονται τα ψηφία του  $B$  και προστίθεται και μία μονάδα στον πρώτο αθροιστή (υπολογισμός συμπληρώματος ως προς 2)



## Σειριακός αθροιστής



Το κρατούμενο  $C_i$  συνδυάζεται με τα ψηφία  $A_{i+1}$  και  $B_{i+1}$  ώστε να υπολογιστεί το άθροισμα της επόμενης στήλης, κ.ο.κ.

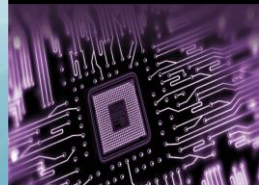
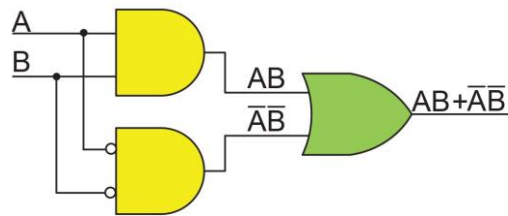


# Λογικό κύκλωμα συγκριτή (1)

Έλεγχος ισότητας ( $A=B$ )

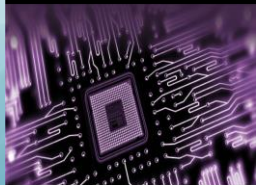
$$X_i = A_i B_i + \bar{A}_i \bar{B}_i$$

Κύκλωμα υλοποίησης



## Λογικό κύκλωμα συγκριτή (2)

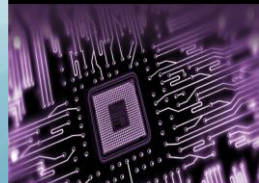
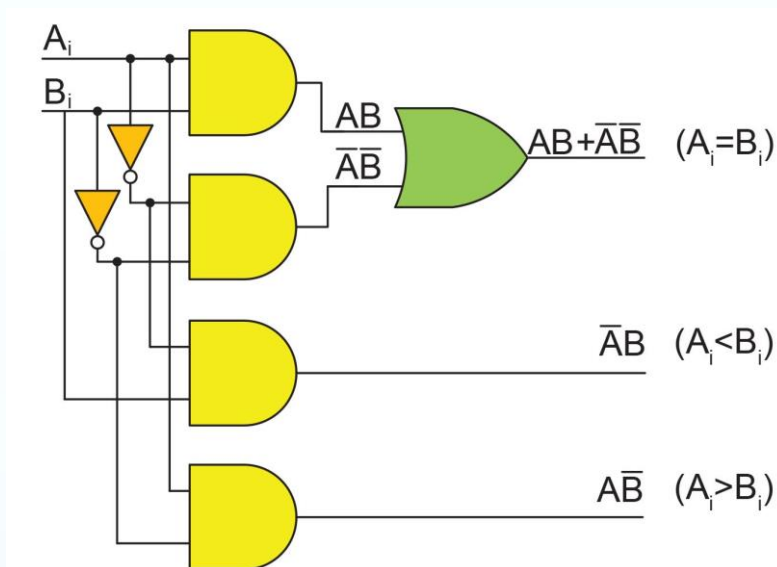
Έλεγχος αν $A < B$			
Αν $A_3 < B_3$ , τότε $A < B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 < B_2$ , τότε $A < B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 = B_2$ ΚΑΙ $A_1 < B_1$ , τότε $A < B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 = B_2$ ΚΑΙ $A_1 = B_1$ ΚΑΙ $A_0 < B_0$ , τότε $A < B$
Έλεγχος αν $A > B$			
Αν $A_3 > B_3$ , τότε $A > B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 > B_2$ , τότε $A > B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 = B_2$ ΚΑΙ $A_1 > B_1$ , τότε $A > B$	Αν $A_3 = B_3$ ΚΑΙ $A_2 = B_2$ ΚΑΙ $A_1 = B_1$ ΚΑΙ $A_0 > B_0$ , τότε $A > B$
(α)	(β)	(γ)	(δ)





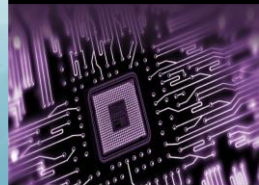
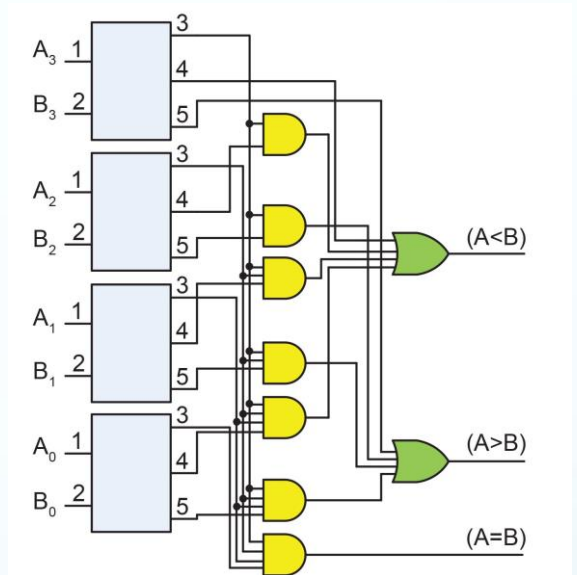
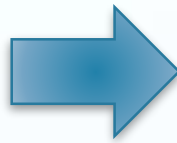
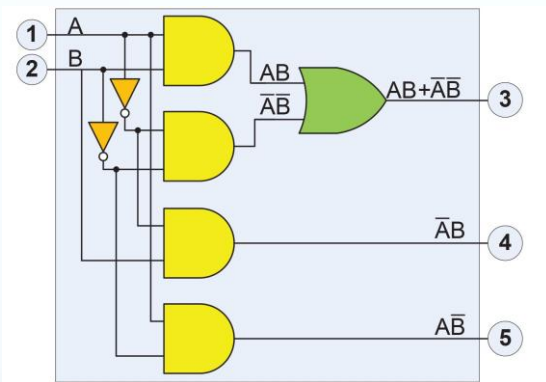
## Λογικό κύκλωμα συγκριτή (3)

Υλοποίηση τριών ελέγχων ( $A=B$ ,  $A<B$ ,  $A>B$ ) για δύο αριθμούς του ενός bit



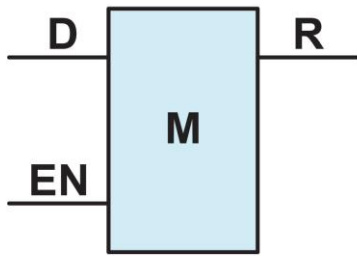
## Λογικό κύκλωμα συγκριτή (4)

Επαναχρησιμοποίηση βασικού κυκλώματος για σύγκριση δύο αριθμών των 4 bit

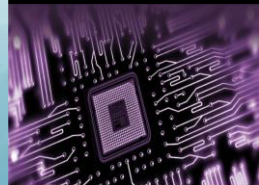
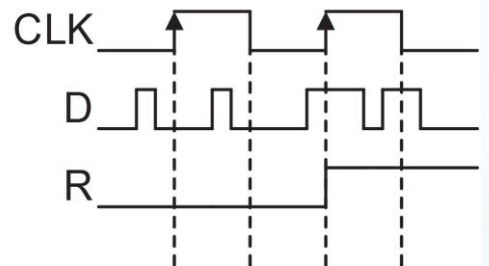
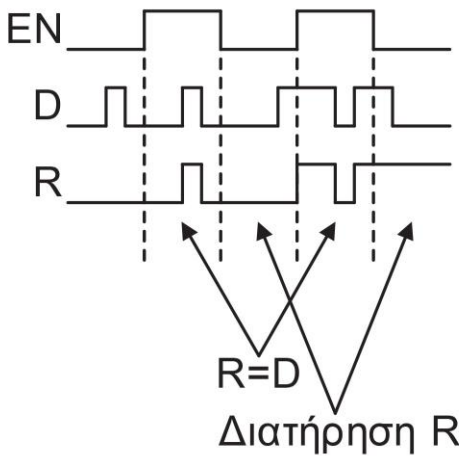
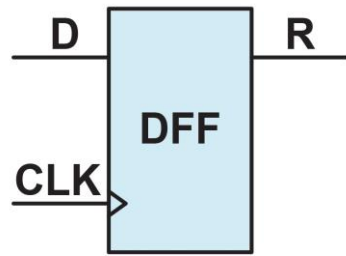


# Βασικό στοιχείο μνήμης (1)

Μανδαλωτής

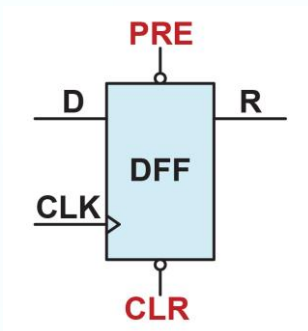


Flip-Flop

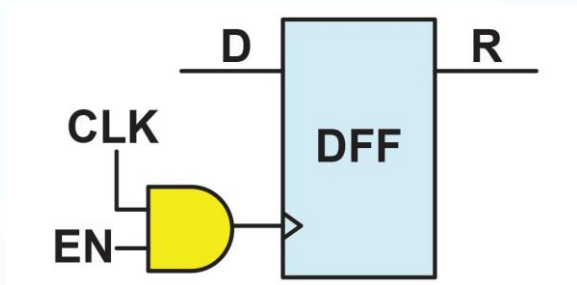


## Βασικό στοιχείο μνήμης (2)

πλήρες DFF

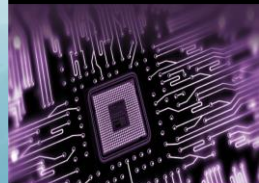
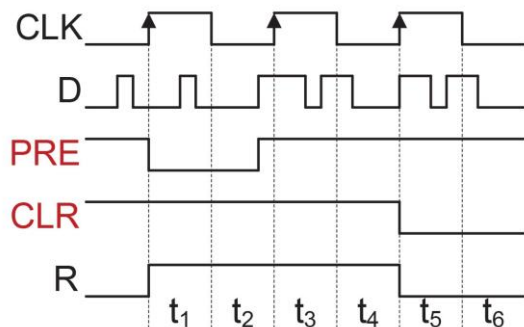


DFF με έλεγχο (EN)

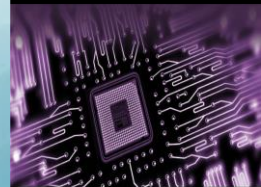
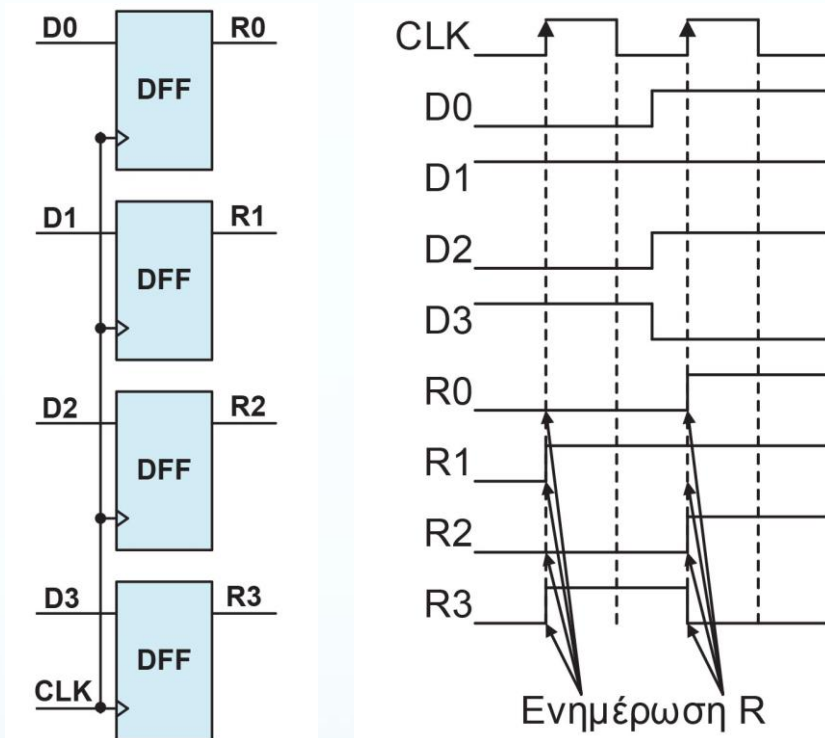


**PRE:** R=1 Ανεξάρτητα από τα D και CLK

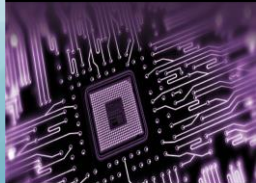
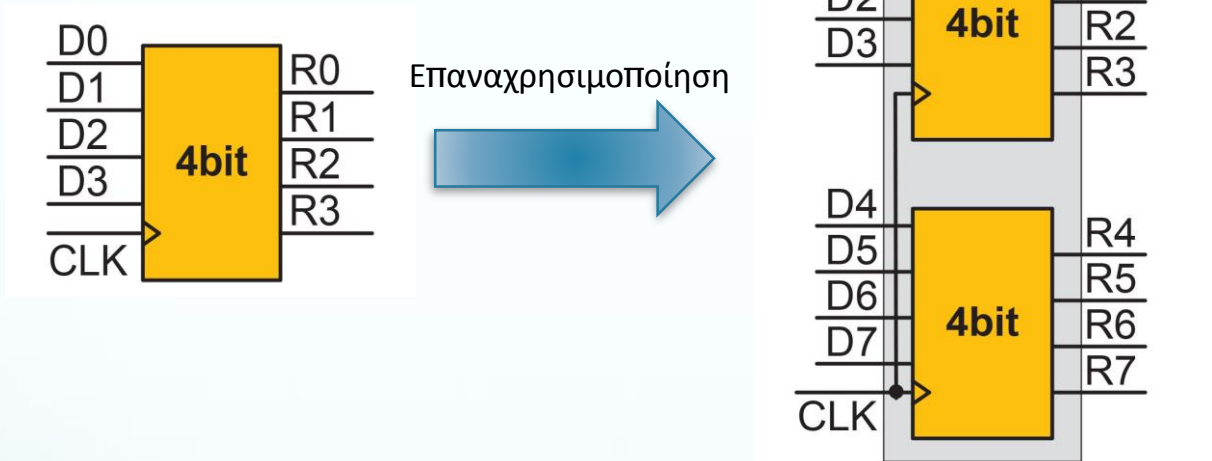
**CLR:** R=0 Ανεξάρτητα από τα D και CLK



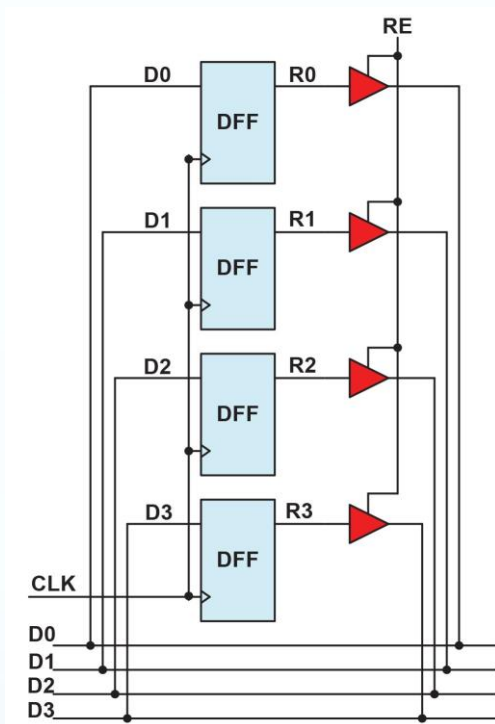
# Καταχωρητής 4bit



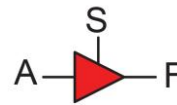
# Καταχωρητής 8bit



## Καταχωρητής 4bit με κοινό δίαυλο δεδομένων



Απομονωτής

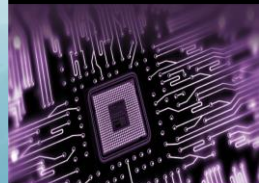


S	A	F
0	0	Z
0	1	Z
1	0	0
1	1	1

Με το σήμα RE ελέγχουμε τη ροή των δεδομένων (R0 έως R3) στον κοινό δίαυλο δεδομένων

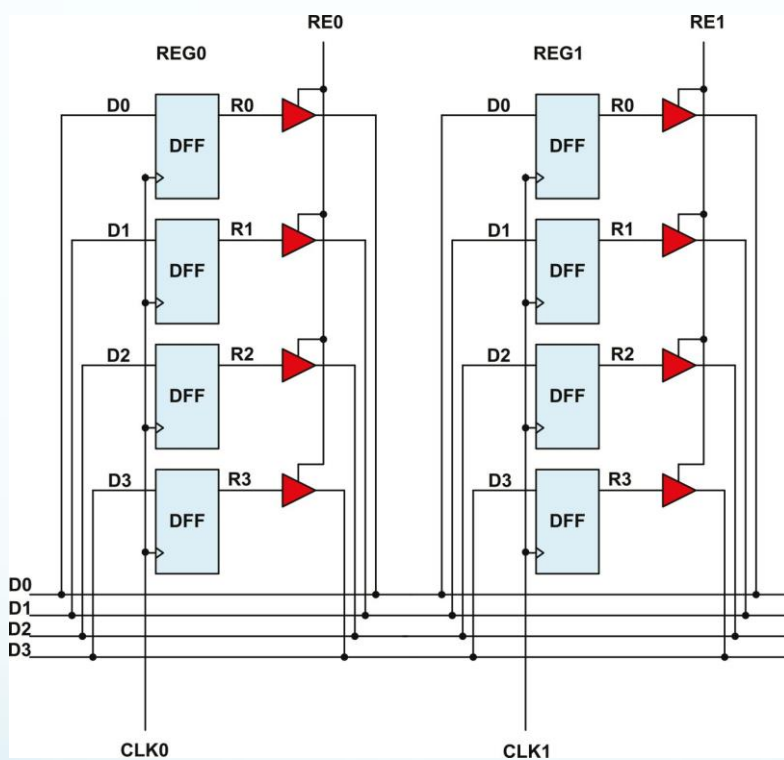
Μικροεπεξεργαστές

Αρχές & Εφαρμογές



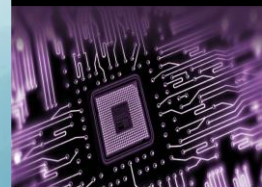
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (1)



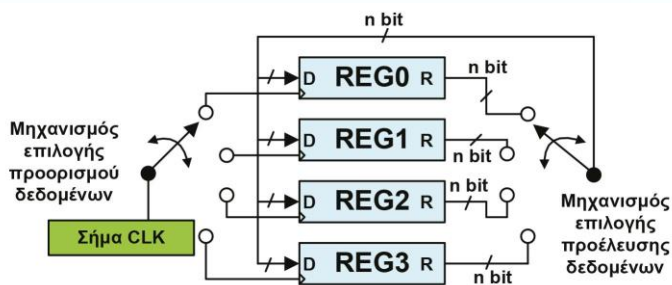
### Υλοποίηση λειτουργίας **REG1=REG0**

- (α)  $RE1=0$  και  $RE0=1$  (Απενεργοποίηση ανάγνωσης από τον REG1 και ενεργοποίηση ανάγνωσης από τον REG0)
- (β) Εφαρμογή παλμού στη γραμμή CLK1 (ενεργοποίηση εγγραφής στον REG1)
- (γ)  $RE0=0$ . Απενεργοποίηση ανάγνωσης από τον πρώτο καταχωρητή (επαναφορά στην αρχική κατάσταση)

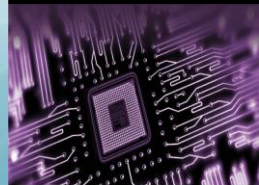




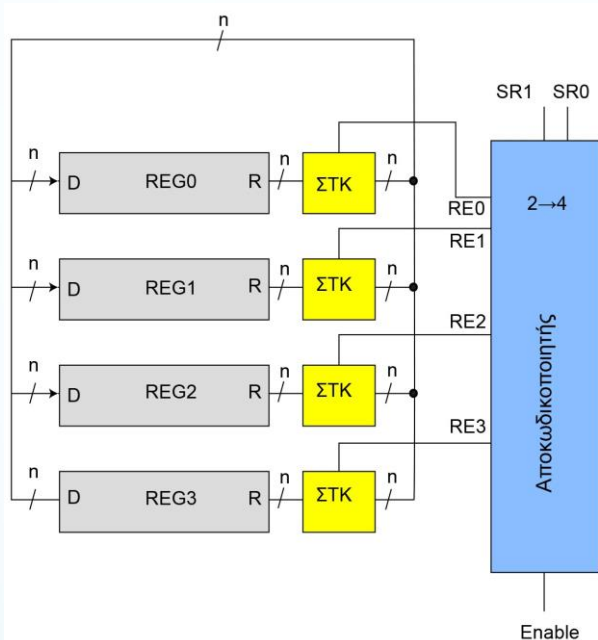
## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (2)



- ❑ Πρέπει να δημιουργήσουμε ένα κύκλωμα που να υλοποιεί τη λειτουργία της φόρτωσης με «αυτοματοποιημένο» τρόπο
- ❑ Ας φανταστούμε αρχικά ένα μηχανισμό επιλογής καταχωρητή προέλευσης και προορισμού
- ❑ Θα πρέπει να αντικατασταθεί αυτός ο μηχανισμός με ένα κύκλωμα που θα ελέγχεται από συγκεκριμένα σήματα



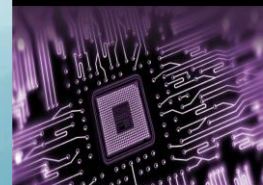
## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (3)



### Κύκλωμα εξόδου

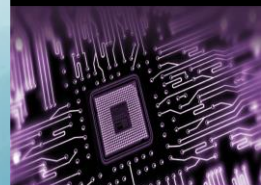
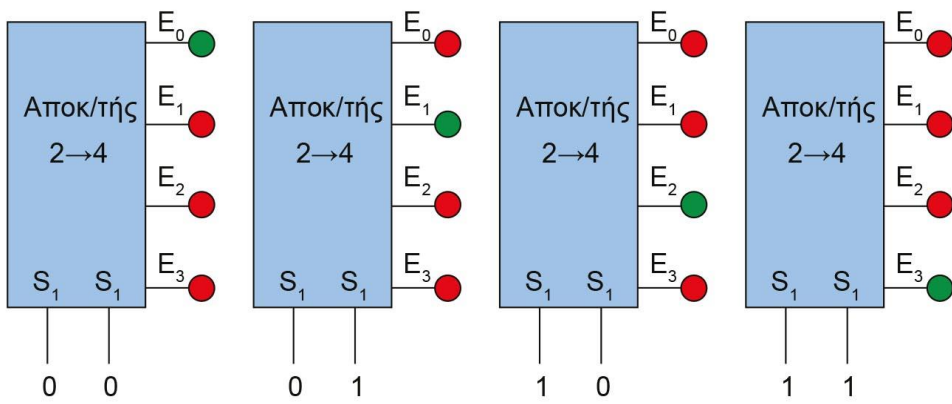
- ❑ Σήματα SR1 και SR0 = ενεργοποίηση μιας από τις εξόδους του αποκωδικοποιητή (RE0 έως RE3)
- ❑ Σήματα (RE0 έως RE3) = Ενεργοποίηση ενός στοιχείου τριών καταστάσεων (ΣΤΚ) για διοχέτευση του περιεχομένου καταχωρητή στο δίαυλο δεδομένων (καταχωρητής προέλευσης)

SR1	SR0	Καταχωρητής προέλευσης	Δεδομένα στο δίαυλο (D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> )	RE0	RE1	RE2	RE3
0	0	REG0	REG0	1	0	0	0
0	1	REG1	REG1	0	1	0	0
1	0	REG2	REG2	0	0	1	0
1	1	REG3	REG3	0	0	0	1

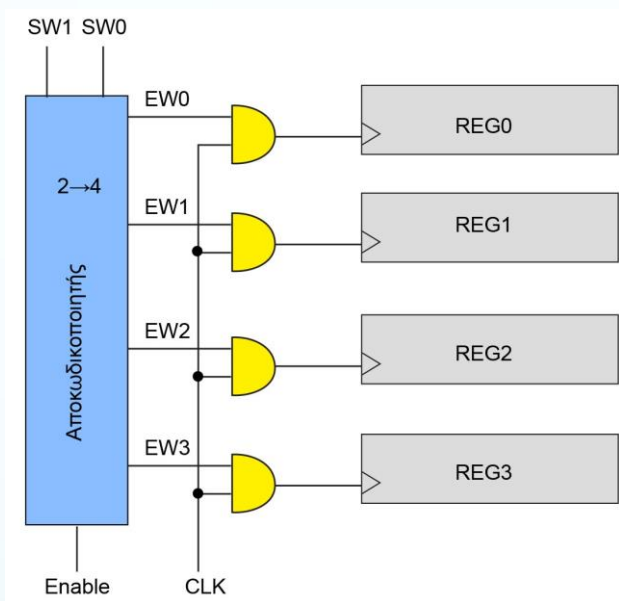


## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (4)

### Λειτουργία αποκωδικοποιητή



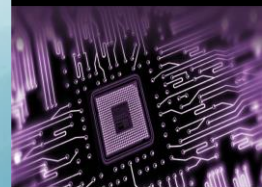
## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (5)



### Κύκλωμα εισόδου

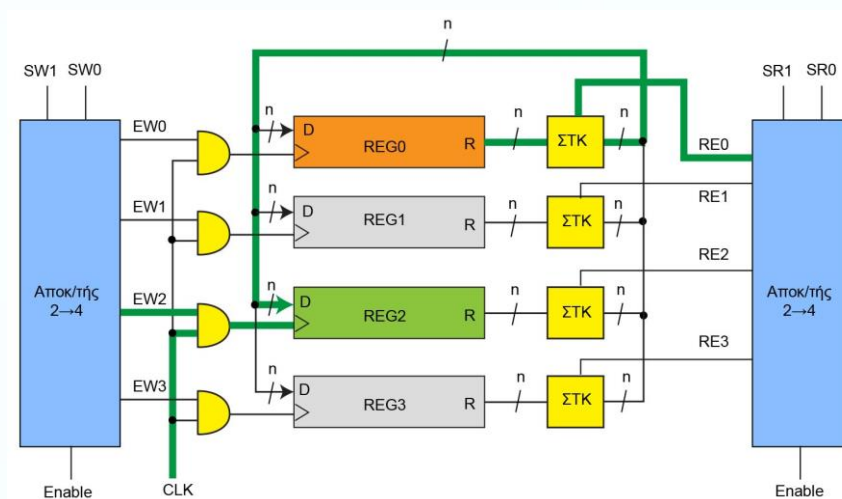
- ❑ Σήματα SW1 και SW0 = ενεργοποίηση μιας από τις εξόδους του αποκωδικοποιητή (EW0 έως EW3)
- ❑ Σήματα (EW0 έως EW3) = Έλεγχος ενεργοποίησης καταχωρητή για εγγραφή (καταχωρητής προορισμού)
- ❑ Εγγραφή = Ενεργό σήμα EW και παλμός CLK

SW1	SW0	Καταχωρητής προορισμού	EW0	EW1	EW2	EW3
0	0	REG0	1	0	0	0
0	1	REG1	0	1	0	0
1	0	REG2	0	0	1	0
1	1	REG3	0	0	0	1



## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (6)

### Ολοκληρωμένο κύκλωμα

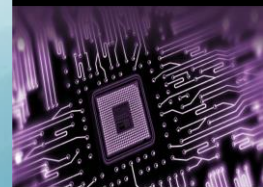


#### Λειτουργία **REG2=REG0**

(α) Ενεργοποίηση ανάγνωσης από τον καταχωρητή REG0 ( $SR1=0$  και  $SR0=0$ ).  
Αν  $SR1=0$  και  $SR0=0$ , τότε  $RE0=1$ ,  $RE1=0$ ,  $RE2=0$  και  $RE3=0$

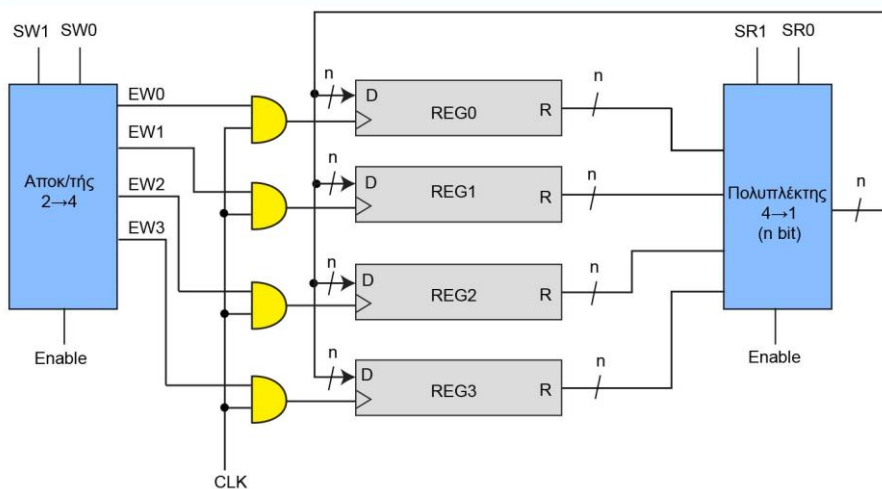
(β) Ενεργοποίηση εγγραφής μόνο στον καταχωρητή REG2 ( $SW1=1$  και  $SW0=0$ ).  
Αν  $SW1=1$  και  $SW0=0$ , τότε  $EW2=1$

(γ) Απενεργοποίηση εγγραφής και ανάγνωσης ( $Enable=0$ )

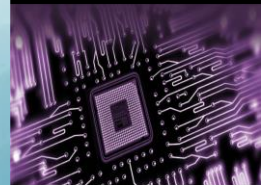


## Φόρτωση και ανάγνωση καταχωρητών με επιλογή (7)

### Υλοποίηση με πολυπλέκτη στο κύκλωμα εξόδου

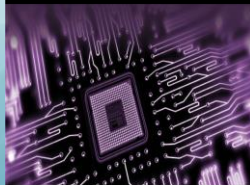


- Όλα τα περιεχόμενα των καταχωρητών διοχετεύονται στον πολυπλέκτη
- Με τα σήματα SR1 και SR0 επιλέγουμε από ποιον καταχωρητή τα περιεχόμενα θα προωθηθούν στο δίαυλο δεδομένων



## Μονάδα ελέγχου (1)

- ❑ Παραγωγή σημάτων για την εκτέλεση εντολής (ενεργοποίηση μονάδων)
- ❑ Κατηγορίες λειτουργίας («Μετάφραση» και εκτέλεση εντολής, προετοιμασία επόμενης εντολής)



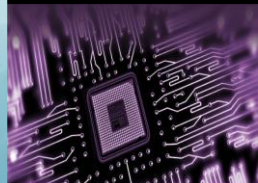
## Μονάδα ελέγχου (2)

### Σχεδίαση με καλωδιωμένη (hardwired) λογική

- ❑ Υλοποίηση λειτουργιών με αφιερωμένα ψηφιακά κυκλώματα
- ❑ Πολυπλοκότητα κυκλωμάτων, δυσκολία αλλαγής
- ❑ Μεγάλες ταχύτητες λόγω υλοποίησης στο υλικό
- ❑ Κατάλληλη για μικροεπεξεργαστές που διαθέτουν απλές εντολές
- ❑ Αλλαγή των χαρακτηριστικών μπορεί να σημαίνει επανασχεδιασμό ολόκληρης της μονάδας
- ❑ Συνηθίζεται στους μικροεπεξεργαστές τύπου RISC

### Σχεδίαση με μικροπρογραμματιζόμενη (microprogrammed) λογική

- ❑ Περιγραφή λειτουργιών με ειδικές λέξεις (μικροεντολές)
- ❑ Η μικροεντολή περιέχει στον κώδικά της τα σήματα ενεργοποίησης των μονάδων
- ❑ Προσφέρει ευελιξία σε τυχόν αλλαγές
- ❑ Προϋποθέτει ειδική μονάδα εκτέλεσης των μικροεντολών
- ❑ Προϋποθέτει την ύπαρξη μικρομνήμης
- ❑ Συνηθίζεται στους μικροεπεξεργαστές τύπου CISC

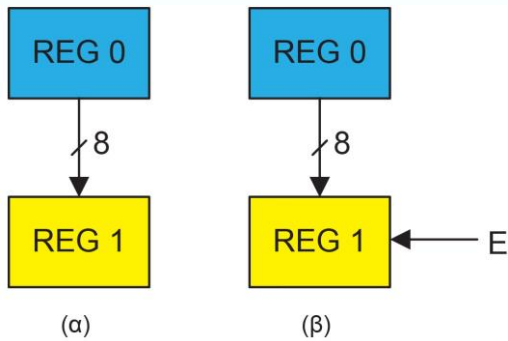




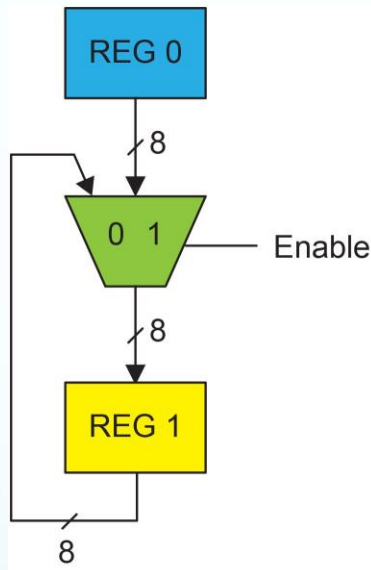
## Μονάδα ελέγχου (3)

Παραδείγματα λογικής λειτουργίας και ελέγχου κυκλωμάτων  
(γενική φιλοσοφία υλοποίησης)

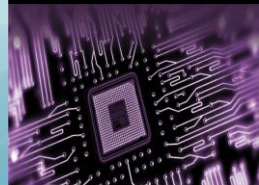
Παράδειγμα 1 : **REG1  $\leftarrow$  REG0**



**(α)** Άμεση φόρτωση  
**(β)** Φόρτωση με έλεγχο  
(E:REG1  $\leftarrow$  REG0)



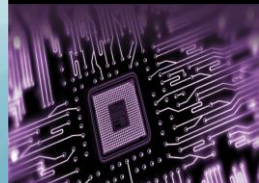
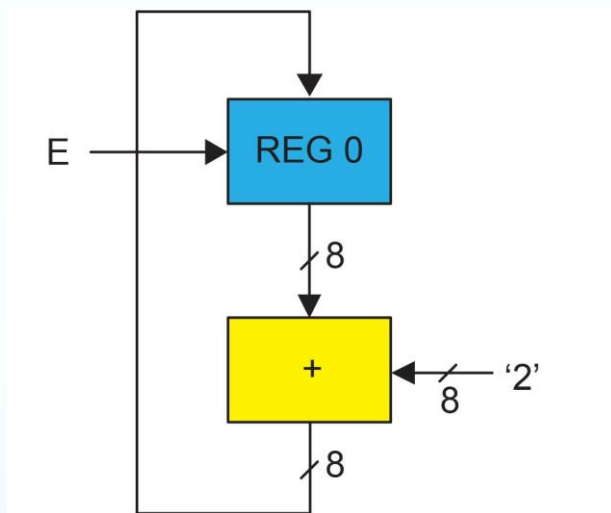
Λειτουργία σήματος Enable



## Μονάδα ελέγχου (4)

Παραδείγματα λογικής λειτουργίας και ελέγχου κυκλωμάτων  
(γενική φιλοσοφία υλοποίησης)

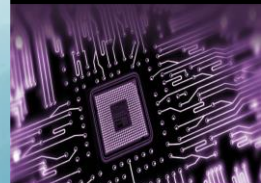
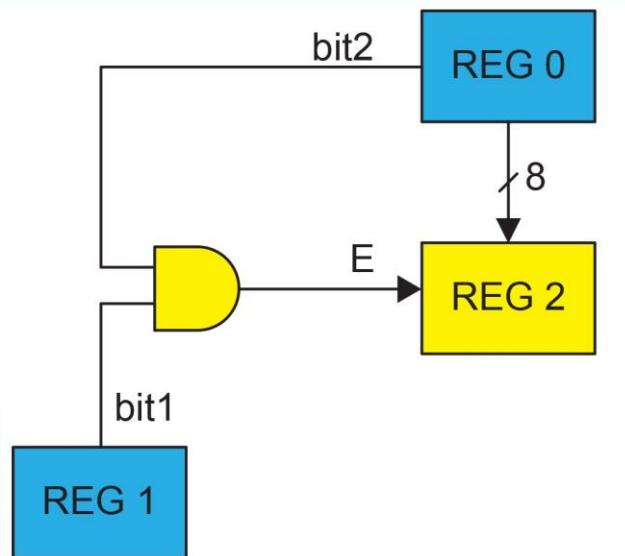
Παράδειγμα 2 : **REG0**  $\leftarrow$  **REG0** +  
**2**



## Μονάδα ελέγχου (5)

Παραδείγματα λογικής λειτουργίας και ελέγχου κυκλωμάτων  
(γενική φιλοσοφία υλοποίησης)

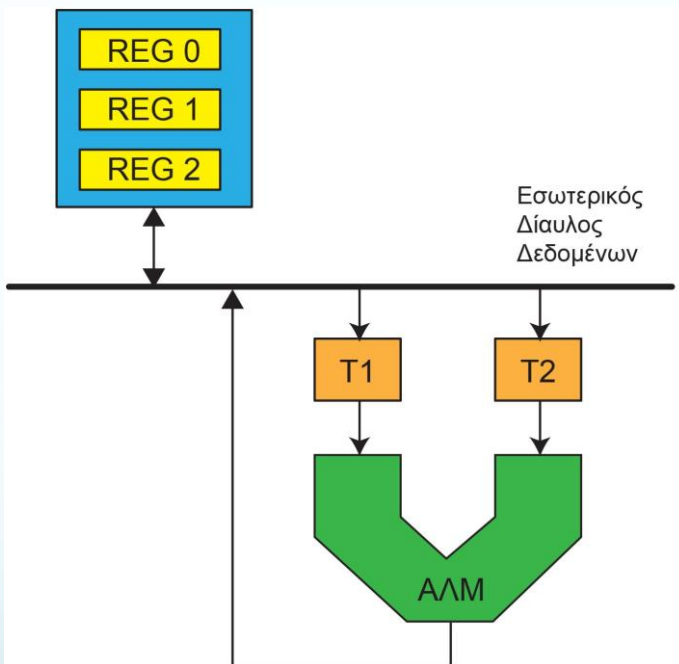
Παράδειγμα 3 : **IF REG0[2]=1 AND REG1[1]=1 THEN  
REG2←REG0**



## Μονάδα ελέγχου (6)

Μονοί και πολλαπλοί δίαυλοι  
(γενική φιλοσοφία λειτουργίας)

Ένας εσωτερικός δίαυλος



**REG0 ← REG1 + REG2**

**Κύκλος 1**

T1 ← REG1

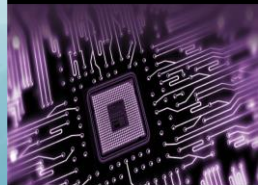
**Κύκλος 2**

T2 ← REG2

**Κύκλος 3**

Εκτέλεση

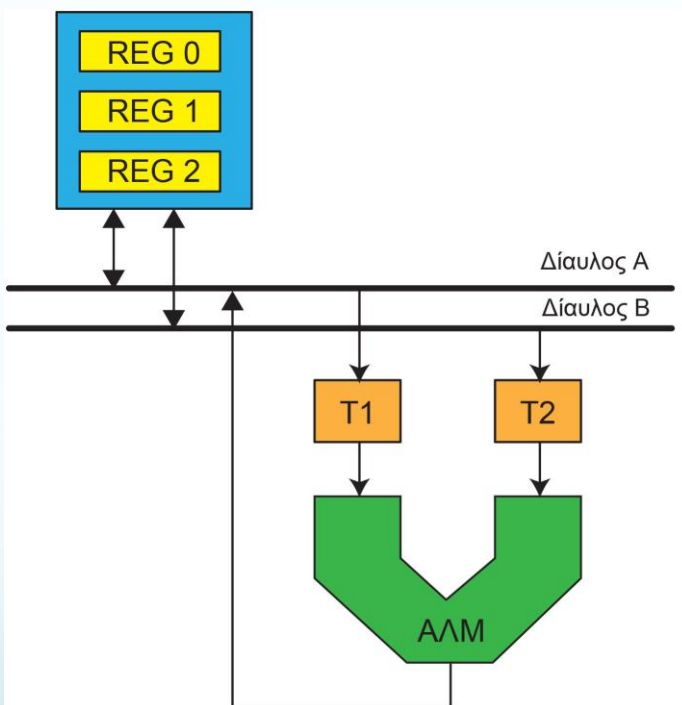
REG0 ← αποτέλεσμα



## Μονάδα ελέγχου (7)

Μονοί και πολλαπλοί δίαυλοι  
(γενική φιλοσοφία λειτουργίας)

Δύο εσωτερικοί δίαυλοι



**REG0 ← REG1 + REG2**

**Κύκλος 1**

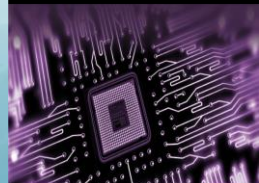
T1 ← REG1

T2 ← REG2

**Κύκλος 2**

Εκτέλεση

REG0 ← αποτέλεσμα

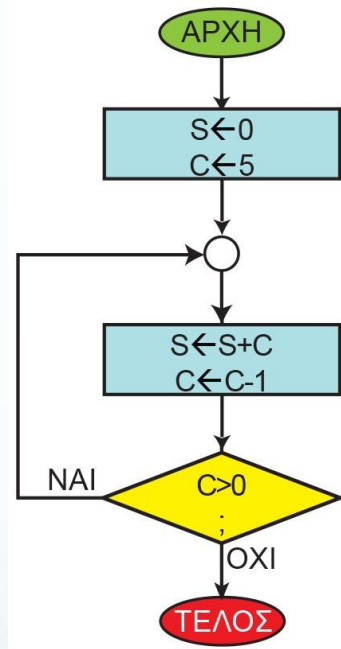


## Μονάδα ελέγχου (8)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1, 5]$ )

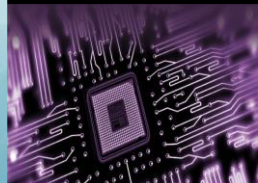
Βήμα 1

Διάγραμμα ροής



Μικροεπεξεργαστές

Αρχές & Εφαρμογές



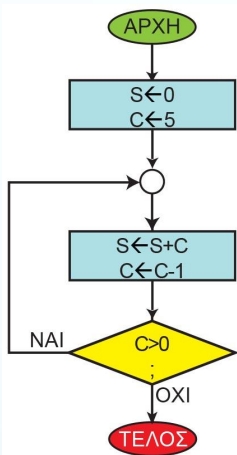
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάλα

## Μονάδα ελέγχου (9)

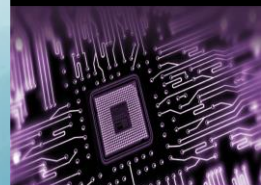
Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1, 5]$ )

Βήμα 2

Λειτουργία αλγόριθμου



Αριθμός επανάληψης	Καταχωρητής S	Καταχωρητής C
1	$0+5=5$	4
2	$5+4=9$	3
3	$9+3=12$	2
4	$12+2=14$	1
5	$14+1=15$	0



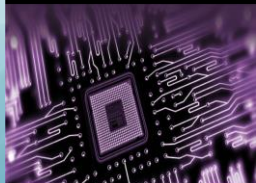
## Μονάδα ελέγχου (10)

**Σχεδίαση με καλωδιωμένη λογική**  
(υλοποίηση υπολογισμού αθροίσματος των ακέραιων αριθμών στο διάστημα [1,5])

**Βήμα 3**

**Τυπική περιγραφή αλγόριθμου**

Περιγραφή	Σχόλια
Begin: $S \leftarrow 0, C \leftarrow$ Δίαυλος εισόδου	$S$ =καταχωρητής αθροίσματος $C=5$ =μετρητής βρόχου
Again: $S \leftarrow S+C, C \leftarrow C-1$ Αν $C>0$ , πήγαινε στο Again	Ενημέρωση αθροίσματος και μετρητή Έλεγχος μετρητή και επαναφορά
Δίαυλος εξόδου $\leftarrow S$	Μεταφορά αποτελέσματος στο δίαυλο εξόδου
End: Πήγαινε στο End (goto End)	Τερματισμός με επαναφορά στο ίδιο σημείο



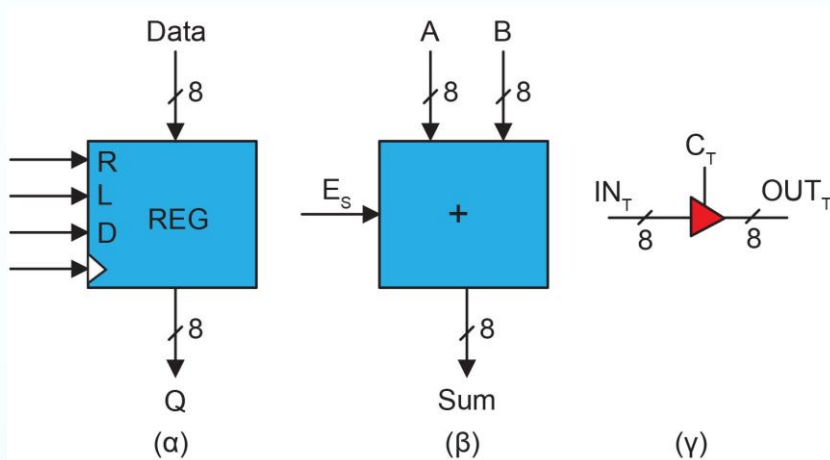


## Μονάδα ελέγχου (11)

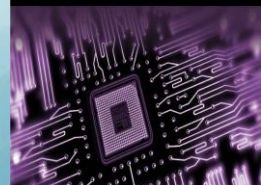
**Σχεδίαση με καλωδιωμένη λογική**  
(υλοποίηση υπολογισμού αθροίσματος των ακέραιων αριθμών στο διάστημα [1,5])

**Βήμα 4**

**Στοιχεία διαχείρισης δεδομένων**



(α) Καταχωρητής  
(β) Αθροιστής  
(γ) Απομονωτής



## Μονάδα ελέγχου (12)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 5

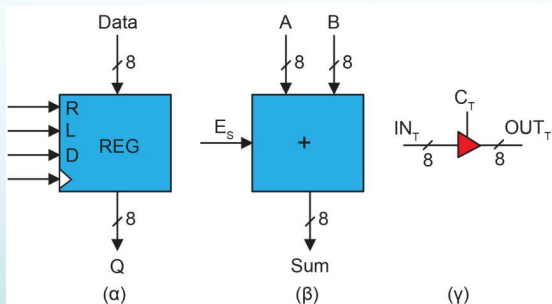
Λειτουργία στοιχείων διαχείρισης

Καταχωρητής				
R	L	D	CLK	Λειτουργία
0	0	0	↑	- (καμία)
1	0	0	↑	Μηδενισμός περιεχομένου (Reset)
0	1	0	↑	Φόρτωση
0	0	1	↑	Μείωση κατά 1

Αθροιστής	
$E_S$	Λειτουργία
0	- (καμία)
1	A+B (πρόσθεση)

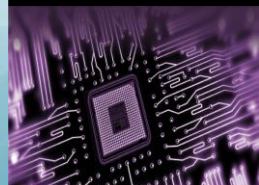
  

Απομονωτής	
$C_T$	Λειτουργία
0	$OUT_T = Z$
1	$OUT_T = IN_T$



Μικροεξεργαστές

Αρχές & Εφαρμογές



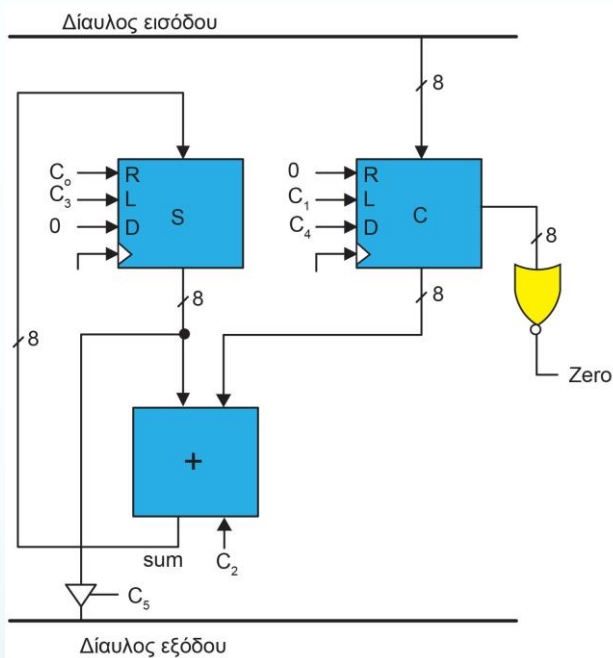
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

## Μονάδα ελέγχου (13)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

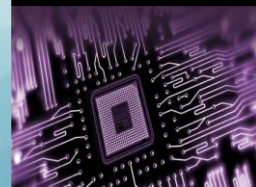
Βήμα 6α

Διασυνδέσεις στοιχείων – ροή δεδομένων



Μικροεξεργαστές

Αρχές & Εφαρμογές



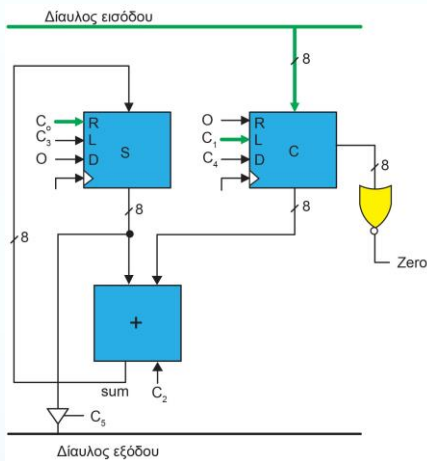
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

## Μονάδα ελέγχου (14)

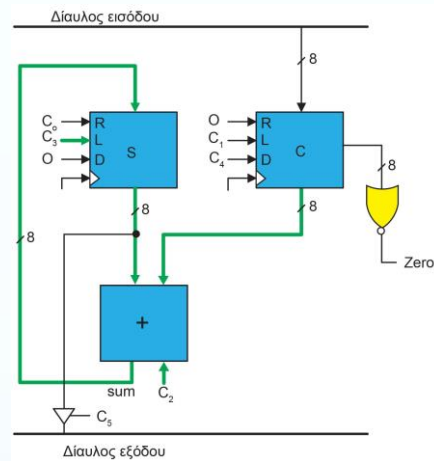
Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1,5]$ )

Βήμα 6β

Διασυνδέσεις στοιχείων – ροή δεδομένων



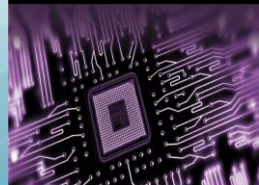
(α) Μηδενισμός S και  
φόρτωση C



(β) Υπολογισμός και  
αποθήκευση  
αθροίσματος

Μικροεξεργαστές

Αρχές & Εφαρμογές



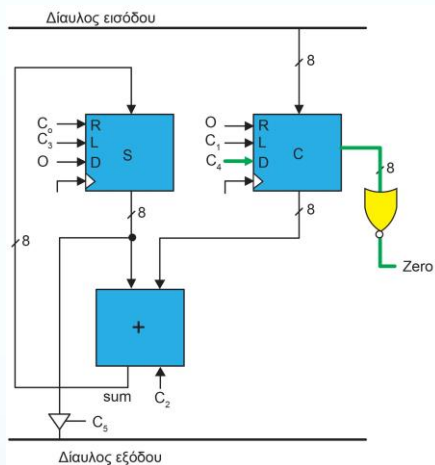
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάφλα

## Μονάδα ελέγχου (15)

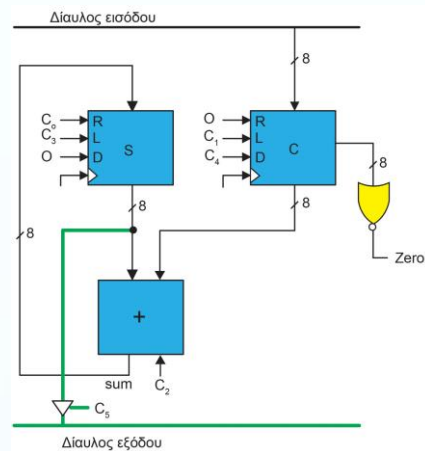
Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1,5]$ )

Βήμα 6γ

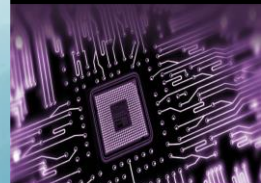
Διασυνδέσεις στοιχείων – ροή δεδομένων



(γ) Ενημέρωση μετρητή



(δ) Διοχέτευση  
αποτελέσματος στο  
δίαυλο εξόδου

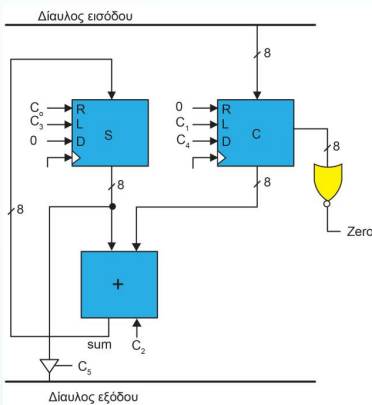


## Μονάδα ελέγχου (16)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 7

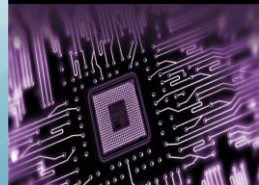
Λίστα σημάτων ελέγχου



Σήμα ελέγχου	Λειτουργία	Μονάδα υλικού
$C_0$	$S \leftarrow 0$	Καταχωρητής S
$C_1$	$C \leftarrow$ διάυλος εισόδου	Καταχωρητής C
$C_2$	$Sum \leftarrow S+C$	Αθροιστής
$C_3$	$S \leftarrow Sum$	Καταχωρητής S
$C_4$	$C \leftarrow C-1$	Καταχωρητής C
$C_5$	Διάυλος εξόδου $\leftarrow S$	Απομονωτής

Μικροεπεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

## Μονάδα ελέγχου (17)

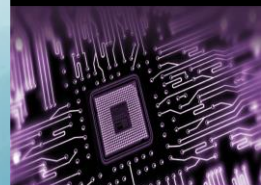
**Σχεδίαση με καλωδιωμένη λογική**  
(υλοποίηση υπολογισμού αθροίσματος των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 8

Παραγωγή σημάτων ελέγχου



- Ενεργοποιεί τα σήματα  $C_0$  έως  $C_5$
- Είναι συγχρονισμένος με το σήμα CLK
- Ανιχνεύει το μηδενισμό του μετρητή (Zero)
- Μπορεί να αρχικοποιηθεί (Reset)

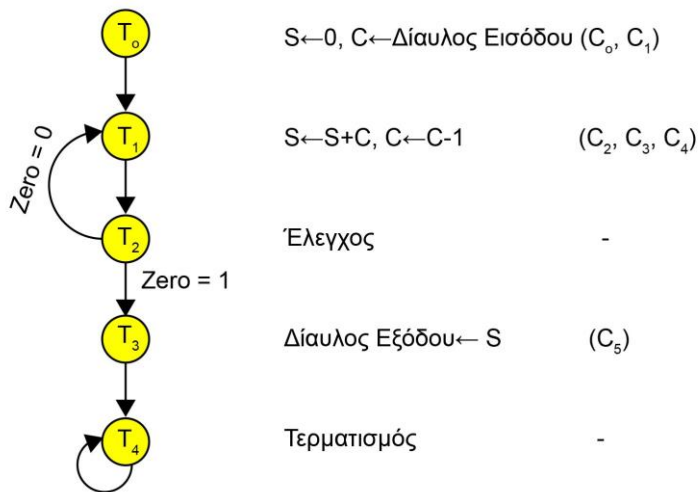


## Μονάδα ελέγχου (18)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1, 5]$ )

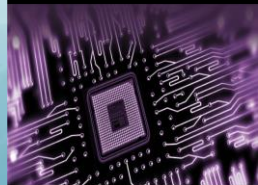
Βήμα 9

Διάγραμμα καταστάσεων



Μικροεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιζόλα

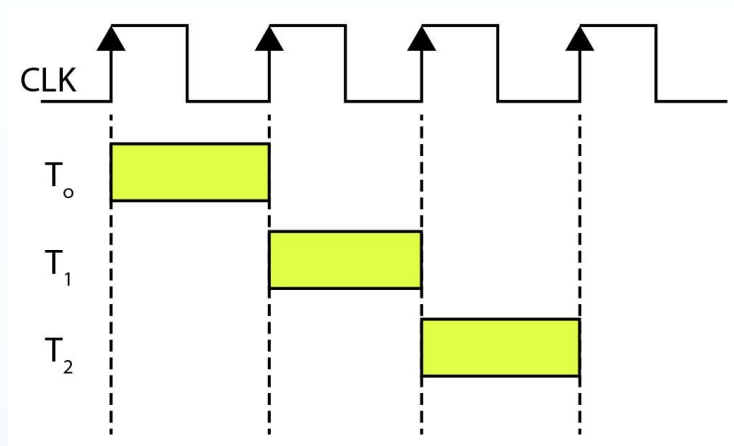
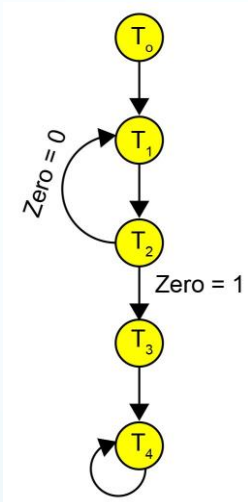


## Μονάδα ελέγχου (19)

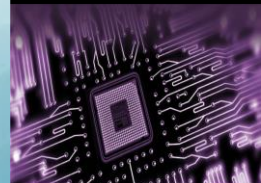
Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1,5]$ )

Βήμα 10

Μετάβαση καταστάσεων



(Ενδεικτικά  
)

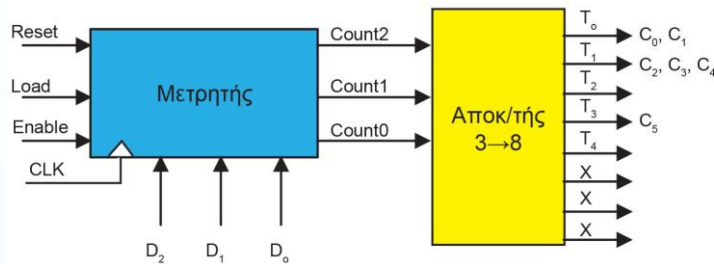


## Μονάδα ελέγχου (20)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 11α

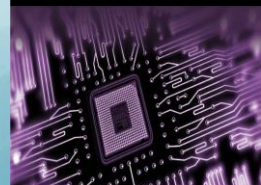
Σχεδίαση ελεγκτή



Count2	Count1	Count0	T <sub>0</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Μικροεπεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου

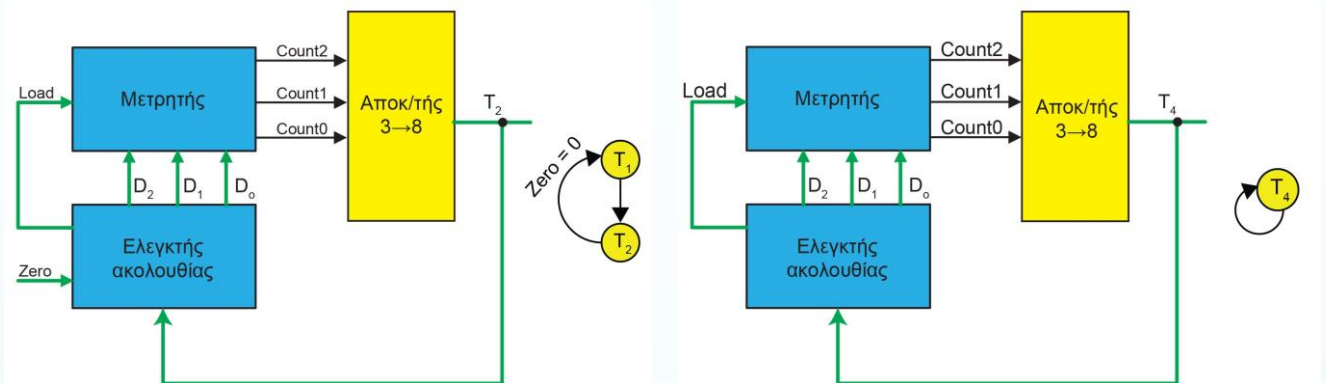
Εκδόσεις Τζάβλα

## Μονάδα ελέγχου (21)

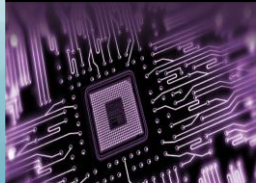
Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 11β

Σχεδίαση ελεγκτή



Ο ελεγκτής ακολουθίας «επιβάλλει» νέα τιμή μέτρησης για τις μεταβάσεις  $T_2 \rightarrow T_1$ ,  $T_4 \rightarrow T_4$

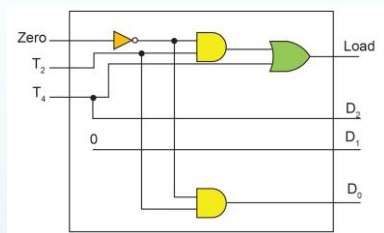
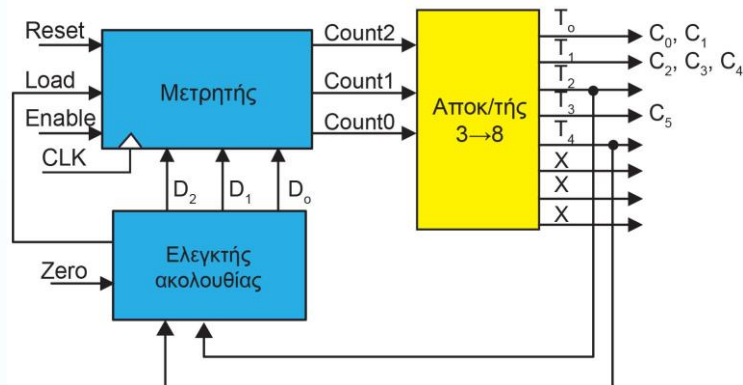


## Μονάδα ελέγχου (22)

Σχεδίαση με καλωδιωμένη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 11γ

Σχεδίαση ελεγκτή



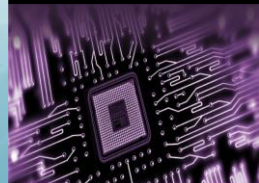
$$\text{Load} = (\text{Zero})'$$

$$T_2 + T_4$$

$$D_2 = T_4$$

$$D_1 = 0$$

$$D_0 = (\text{Zero})' T_2$$

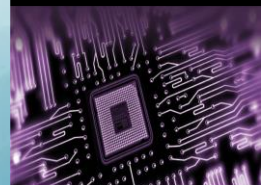


## Μονάδα ελέγχου (23)

Σχεδίαση με μικροπρογραμματιζόμενη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Σήμα Ελέγχου	Λειτουργία
$C_0$	$S \leftarrow 0$
$C_1$	$C \leftarrow$ δίαυλος εισόδου
$C_2$	$Sum \leftarrow S+C$
$C_3$	$S \leftarrow Sum$
$C_4$	$C \leftarrow C-1$
$C_5$	Δίαυλος εξόδου $\leftarrow S$

Μικροεντολή	$C_5C_4C_3C_2C_1C_0$	Ενεργοποίηση
1 <sup>η</sup>	000011	$C_1, C_0$
2 <sup>η</sup>	001100	$C_3, C_2$
3 <sup>η</sup>	010000	$C_4$
4 <sup>η</sup>	100000	$C_5$



## Μονάδα ελέγχου (24)

Σχεδίαση με μικροπρογραμματιζόμενη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 1

Μικροπρόγραμμα – υλοποίηση αλγορίθμου

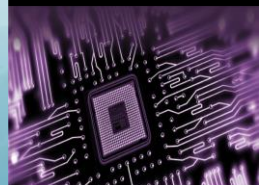
Διεύθυνση στη μικρομνήμη	Αριθμός μικροεντολής	
0	1	Begin: $S \leftarrow 0, C \leftarrow$ Δίαυλος εισόδου
1	2	Again: $S \leftarrow S+C$
2	3	$C \leftarrow C-1$
3	4	Αν $C > 0$ , πήγαινε στο Again
4	5	Δίαυλος εξόδου $\leftarrow S$
5	6	End: Πήγαινε στο End (goto End)

Δομή μικροεντολής

Επιλογή (είδος συνθήκης (ΕΣ))	Διεύθυνση Διακλάδωσης (ΔΔ)	Σήματα Ελέγχου (ΣΕ)
-------------------------------	----------------------------	---------------------

Μικροεξεργαστές

Αρχές & Εφαρμογές



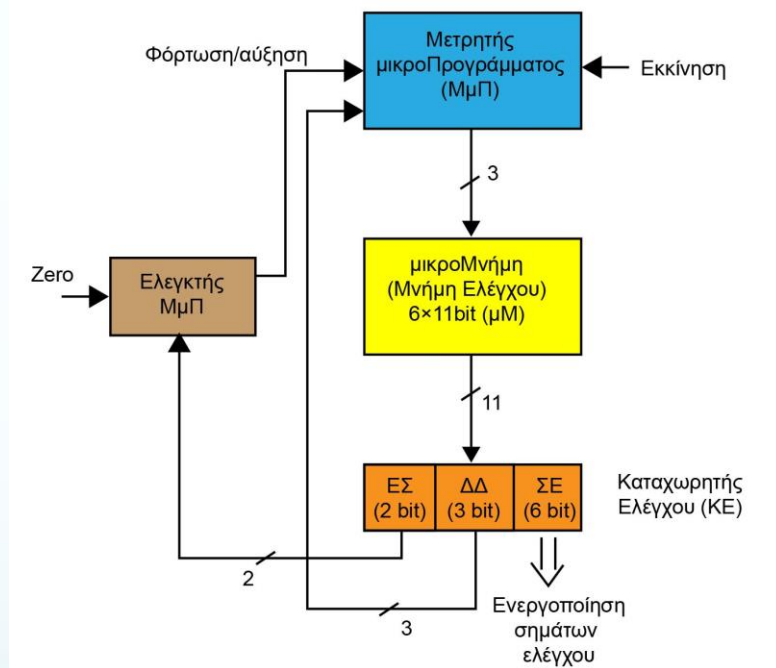
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάλα

## Μονάδα ελέγχου (25)

Σχεδίαση με μικροπρογραμματιζόμενη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1,5]$ )

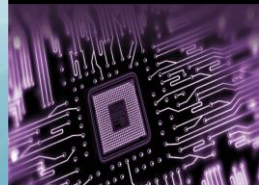
Βήμα 2

Δομή μονάδας ελέγχου



Μικροεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου

Εκδόσεις Τζάλα

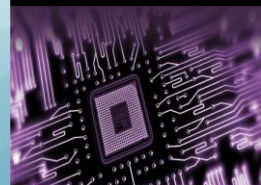
## Μονάδα ελέγχου (26)

Σχεδίαση με μικροπρογραμματιζόμενη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 3

Το μικροπρόγραμμα στη μικρομνήμη

Μικρομνήμη (μνήμη ελέγχου)		Πεδία μικροεντολής			
Αριθμός θέσης	Διεύθυνση	ΕΣ (2 bit)	ΔΔ (3 bit)	ΣΕ (C5..C0) (6 bit)	Λειτουργία
0	000	00	000	000011	$S \leftarrow 0, C \leftarrow$ Δίαυλος εισόδου
1	001	00	000	001100	Again: $S \leftarrow S+C$
2	010	00	000	010000	$C \leftarrow C-1$
3	011	01	001	000000	Αν $C>0$ , πήγαινε στο Again
4	100	00	000	100000	Δίαυλος εξόδου $\leftarrow S$
5	101	10	101	000000	End: Πήγαινε στο End (goto End)





## Μονάδα ελέγχου (27)

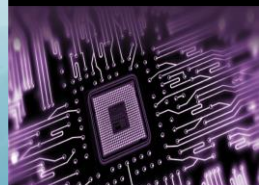
**Σχεδίαση με μικροπρογραμματιζόμενη λογική**  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα  $[1, 5]$ )

**Βήμα 4**

**Βελτιστοποίηση μικροπρογράμματος**

- Όταν πρόκειται για άλμα, το πεδίο ΣΕ περιέχει μηδενικά
- Ορθή διαχείριση πεδίου ΣΕ
- Το πεδίο ΣΕ μπορεί να φιλοξενεί διεύθυνση και σήματα ελέγχου κατά περίπτωση
- Απαραίτητο πεδίο = Ταυτότητα εντολής (αν το ΣΕ θα περιέχει διεύθυνση ή σήματα ελέγχου)

Πεδίο ΕΣ	Σημασία
00	ΣΕ = σήματα ελέγχου
01	ΣΕ = διεύθυνση
10	ΣΕ = διεύθυνση (goto)



## Μονάδα ελέγχου (28)

Σχεδίαση με μικροπρογραμματιζόμενη λογική  
(υλοποίηση υπολογισμού αθροίσματος  
των ακέραιων αριθμών στο διάστημα [1,5])

Βήμα 5

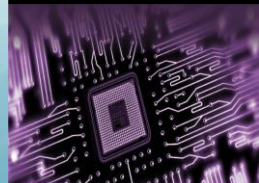
Νέα μορφή μικροεντολής

<b>Επιλογή (είδος) συνθήκης (ΕΣ) 2 bit</b>	<b>Διεύθυνση Διακλάδωσης (ΔΔ) 3 bit</b>	<b>Σήματα Ελέγχου (ΣΕ) 6 bit</b>
--	---	--

Δομή μικροεντολής πριν τη βελτιστοποίηση

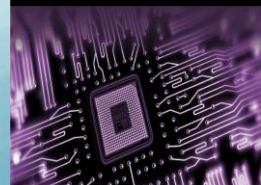
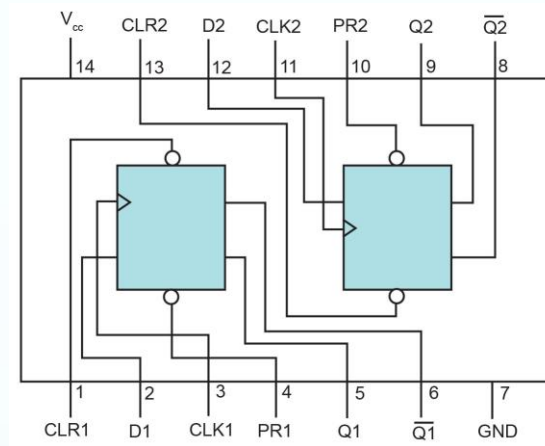
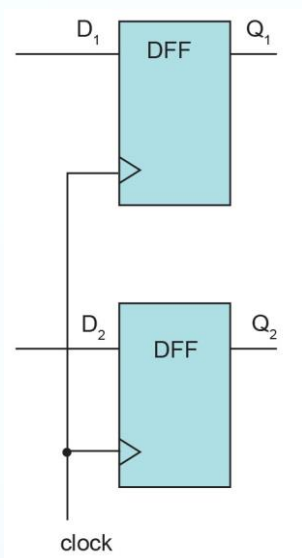
<b>Ταυτότητα εντολής (ΕΣ) 2 bit</b>	<b>Σήματα Ελέγχου/Διεύθυνση Διακλάδωσης (ΣΕ) 6 bit</b>
---	--

Δομή μικροεντολής μετά τη βελτιστοποίηση



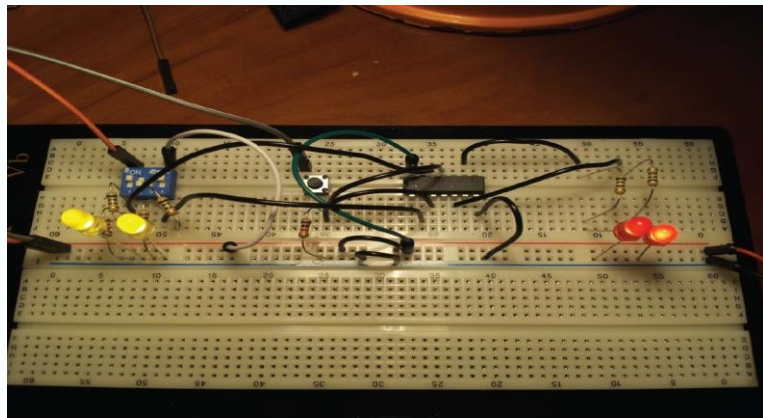
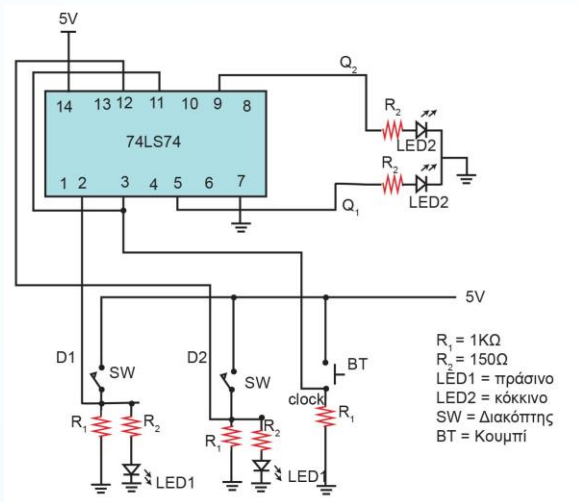
# Πραγματικό Πείραμα στο εργαστήριο (1)

Κύκλωμα μνήμης 2bit



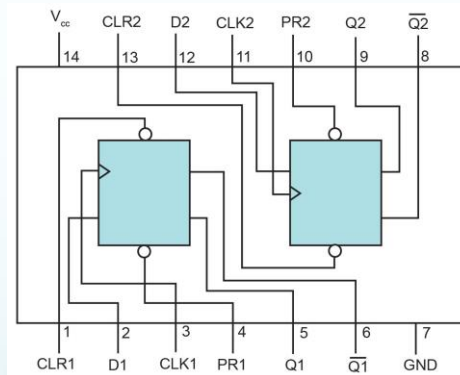
## Πραγματικό Πείραμα στο εργαστήριο (2)

### Κύκλωμα μνήμης 2bit



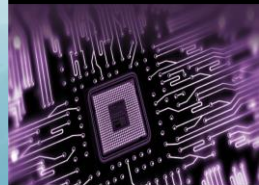
**SW** = σήμα δεδομένων εισόδου

**BT** = κουμπί παλμού clock



Μικροεξεργαστές

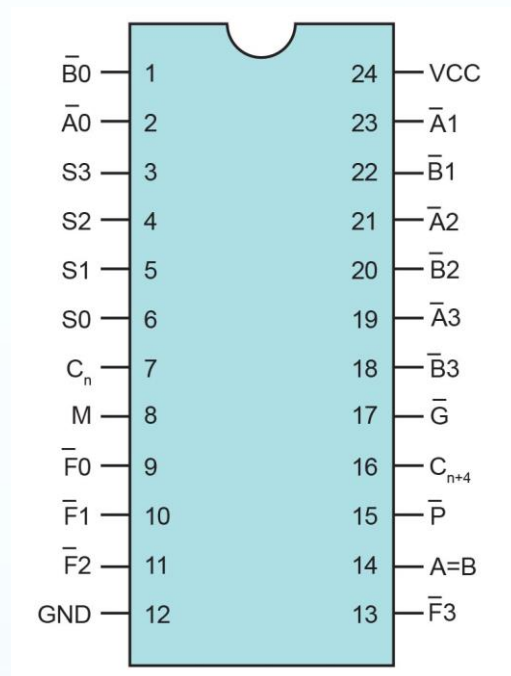
Αρχές & Εφαρμογές



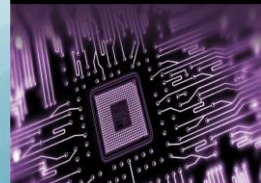
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάβλα

## Πραγματικό Πείραμα στο εργαστήριο (3)

Αριθμητική και λογική μονάδα 4bit



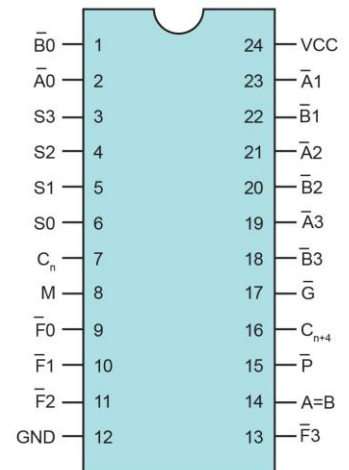
74181



## Πραγματικό Πείραμα στο εργαστήριο (4)

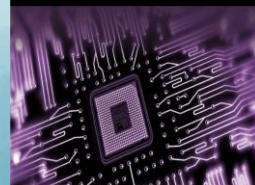
### Αριθμητική και λογική μονάδα 4bit

Είσοδοι				Ενεργοί σε LOW τελεστές & έξοδοι $F_n$		Ενεργοί σε HIGH τελεστές & έξοδοι $F_n$	
S3	S2	S1	S0	Λογική (M=H)	Αριθμητική <sup>(2)</sup> (M=L)(C <sub>n</sub> =L)	Λογική (M=H)	Αριθμητική <sup>(2)</sup> (M=L)(C <sub>n</sub> =H)
L	L	L	L	$\bar{A}$	A μείον 1	$\bar{A}$	A
L	L	L	H	$\bar{A}\bar{B}$	AB μείον 1	$\bar{A}+\bar{B}$	A+B
L	L	H	L	$\bar{A}+\bar{B}$	$A\bar{B}$ μείον 1	$\bar{A}B$	$A+\bar{B}$
L	L	H	H	λογικό 1	μείον 1	λογικό 0	μείον 1
L	H	L	L	$\bar{A}+\bar{B}$	A συν ( $A+\bar{B}$ )	$\bar{A}\bar{B}$	A συν $A\bar{B}$
L	H	L	H	$\bar{B}$	AB συν ( $A+\bar{B}$ )	$\bar{B}$	(A+B) συν $A\bar{B}$
L	H	H	L	$\bar{A}\oplus\bar{B}$	A μείον B μείον 1	$A\oplus B$	A μείον B μείον 1
L	H	H	H	$A+\bar{B}$	$A+\bar{B}$	$A\bar{B}$	AB μείον 1
H	L	L	L	$\bar{A}B$	A συν (A+B)	$\bar{A}+B$	A συν AB
H	L	L	H	$A\oplus B$	A συν B	$\bar{A}\oplus\bar{B}$	A συν B
H	L	H	L	B	$A\bar{B}$ συν (A+B)	B	( $A+\bar{B}$ ) συν AB
H	L	H	H	A+B	A+B	AB	AB μείον 1
H	H	L	L	λογικό 0	A συν A <sup>(1)</sup>	λογικό 1	A συν A <sup>(1)</sup>
H	H	L	H	$A\bar{B}$	AB συν A	$A+\bar{B}$	(A+B) συν A
H	H	H	L	$AB$	$A\bar{B}$ μείον A	A+B	( $A+\bar{B}$ ) συν A
H	H	H	H	A	A	A	A μείον 1



Μικροεπεξεργαστές

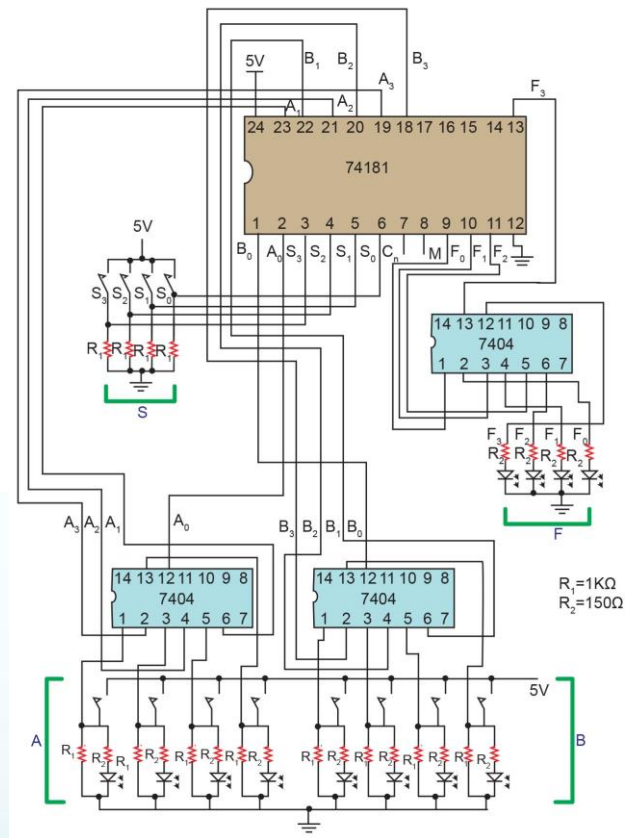
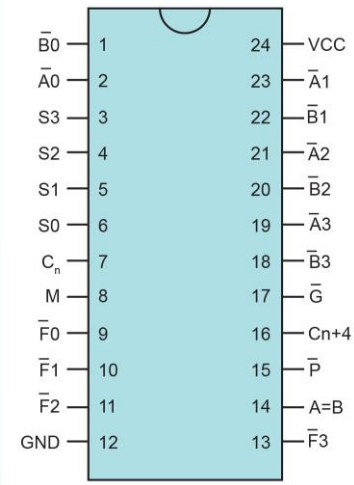
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Πραγματικό Πείραμα στο εργαστήριο (5)

## Αριθμητική και λογική μονάδα 4bit



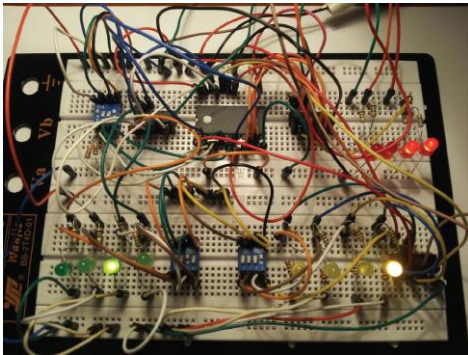
Βάσει του αντίστοιχου πίνακα αληθείας (Περιέχεται στο Datasheet) επιβεβαιώνουμε τη λειτουργία του

**Μικροεξεργαστές**  
Αρχές & Εφαρμογές

Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

## Πραγματικό Πείραμα στο εργαστήριο (6)

Αριθμητική και λογική μονάδα 4bit



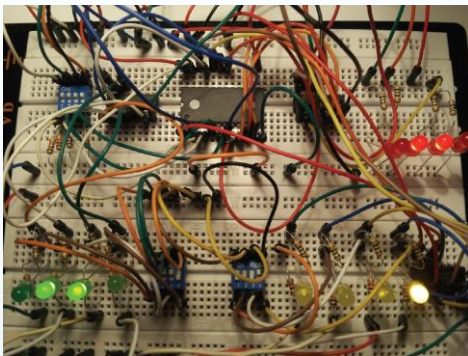
**Ακροδέκτες ελέγχου** HLLH ( $S_3S_2S_1S_0=1001$ )

**Πράξη**  $A+B$

0010=πράσινα LED

0001=κίτρινα LED

0011=κόκκινα LED

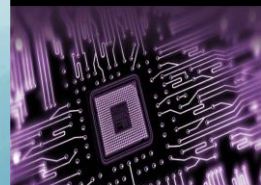


**Ακροδέκτες ελέγχου** HHHH ( $S_3S_2S_1S_0=1111$ )

**F=A**

A=πράσινα LED

F=κόκκινα LED

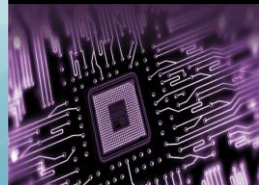




# Κεφάλαιο 6

## Τα κύρια σημεία σε τίτλους

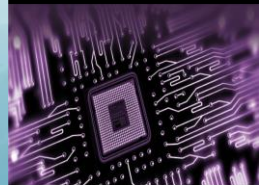
- ❑ Κυκλώματα μικροεπεξεργαστή
- ❑ Αριθμητική και λογική μονάδα
- ❑ Πολυπλέκτης
- ❑ Αθροιστές-ημιαθροιστές, κυκλώματα άθροισης
- ❑ Κύκλωμα συγκριτή
- ❑ Βασικό στοιχείο μνήμης
- ❑ Καταχωρητές (κύκλωμα, φόρτωση, ανάγνωση)
- ❑ Σχεδίαση μονάδας ελέγχου
- ❑ Πραγματικά πειράματα στο εργαστήριο



# Κεφάλαιο 7

## Συμβολική γλώσσα (Assembly) και εκτέλεση προγράμματος

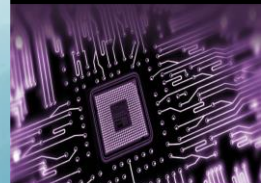
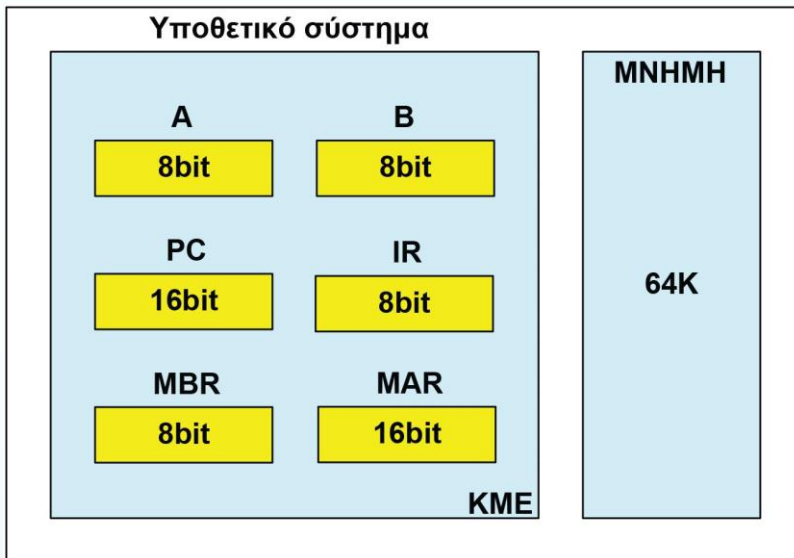
Μικροεξεργαστές  
Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Υποθετικό σύστημα

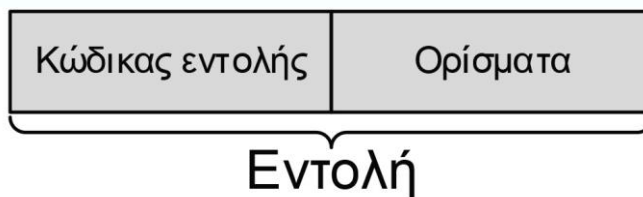
Για την εισαγωγή στη φιλοσοφία του προγραμματισμού σε συμβολική γλώσσα, θα χρησιμοποιήσουμε ένα απλό υποθετικό μικροεπεξεργαστή με τα αρχιτεκτονικά στοιχεία που φαίνονται στο σχήμα.



# Γενικά για τις εντολές συμβολικής γλώσσας

Το σύστημα υποστηρίζει ένα σύνολο από υποθετικές εντολές. Η κατανόηση τους προϋποθέτει την ανάλυση της δομής και της λειτουργίας τους

Γενική μορφή εντολής

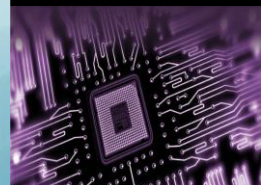


## Κώδικας εντολής

- ❑ Μοναδικό αναγνωριστικό (δυναμικός αριθμός) για να γνωρίζει ο μικροεπεξεργαστής την αντίστοιχη λειτουργία
- ❑ Για τον προγραμματιστή, υπάρχει το μνημονικό (σύντομο και κατανοητό όνομα που μαρτυρά τη λειτουργία της)

## Ορίσματα

- ❑ Ολοκληρώνουν την εντολή
- ❑ Αποτελούν π.χ. αριθμητικά δεδομένα με τα οποία θα γίνει κάποια πράξη, κλπ



# Οι πρώτες εντολές

## **MOV A,x**

$A=x$ , Φόρτωση του αριθμού  $x$  στον καταχωρητή  $A$

## **DEC A**

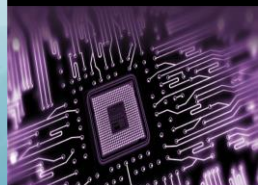
$A=A-1$ , Μείωση του περιεχομένου του  $A$  κατά  $1$

## **JNZ m**

Μετάβαση στη διεύθυνση  $m$  αν από το αποτέλεσμα κάποιας πράξης δεν προκύψει αποτέλεσμα μηδέν (Jump Non Zero).

Εντολή	Κώδικας εντολής
<b>MOV A,x</b>	00000
<b>DEC A</b>	00001
<b>JNZ m</b>	00010

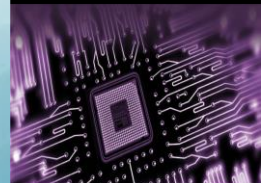
- ❑ Κώδικας εντολής 5bit
- ❑  $2^5=32$  διαφορετικές εντολές μπορεί να υποστηρίξει το υποθετικό σύστημα



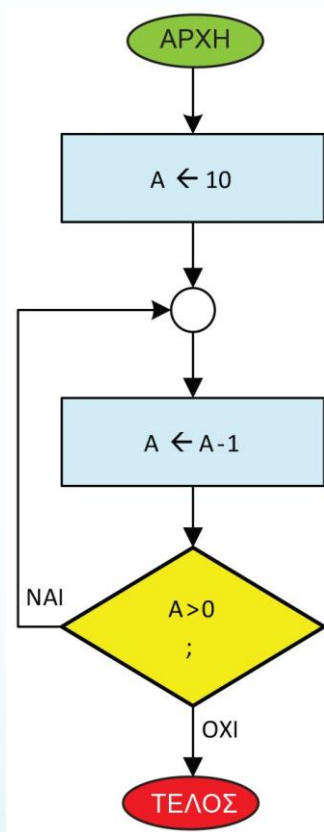
# Μήκος εντολών

Εντολή	Bit κώδικα	Bit ορίσματος	Σύνολο bit
MOV A,x	5	8	13
DEC A	5	-	5
JNZ m	5	16	21

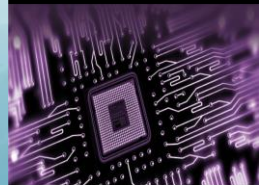
- ❑ Η εντολή **MOV A,x** έχει ως όρισμα το x που είναι ακέραιος αριθμός. Εφόσον το A είναι 8bit, τότε το x θα είναι επίσης 8 bit
- ❑ Η εντολή **DEC A** δεν έχει ορίσματα αφού εκτελεί πάντα την ίδια λειτουργία
- ❑ Η εντολή **JNZ m** έχει ως όρισμα μια διεύθυνση. Δεδομένου ότι το σύστημα διαθέτει μνήμη 64Kbyte, το m θα είναι 16bit



# Το πρώτο μας πρόγραμμα

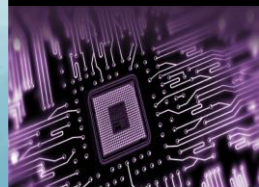


Κώδικας C	Συμβολική γλώσσα
<pre>int i=10; do {     i=i-1; } while (i&gt;0)</pre>	<pre>MOV A,10 START:     DEC A     JNZ START</pre>



# Το πρώτο μας πρόγραμμα στη μνήμη

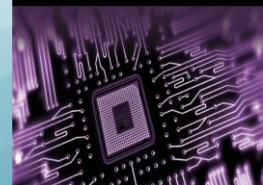
Διεύθυνση	Περιεχόμενο (Δεκαεξαδικό)	Περιεχόμενο (Δυαδικό)	Τύπος	Εντολή	Αριθμός εντολής
BF00	00	(000)00000	Κώδικας εντολής 5bit	MOV A,10	1η
BF01	0A	00001010	Όρισμα 8bit		
BF02	01	(000)00001	Κώδικας εντολής 5bit	DEC A	2η
BF03	02	(000)00010	Κώδικας εντολής 5bit	JNZ START	3η
BF03	BF	10111111	Όρισμα 2x8bit		
BF04	02	00000010			





# Όλες οι εντολές του υποθετικού συστήματος

Εντολή	Κώδικας εντολής (δυαδικό)	Κώδικας εντολής (16δικό)	Ορίσματα	Bit κώδικα	Bit ορισμάτων	Σύνολο Bit
MOV A, x	00000	00	x	5	8	5+8=13
DEC A	00001	01	-	5	-	5
JNZ m	00010	02	m	5	16	5+16=21
RD	00011	03	-	5	-	5
WR	00100	04	-	5	-	5
MOV A, [m]	00101	05	m	5	16	5+16=21
MOV [m], A	00110	06	m	5	16	5+16=21
MOV B, [m]	00111	07	m	5	16	5+16=21
ADD B	01000	08	-	5	-	5
ADD n	01001	09	n	5	8	5+8=13
ADD x, y	01010	0A	x,y	5	8,8	5+8+8=21
ADD [m]	01011	0B	m	5	16	5+16=21
MUL [m]	01100	0C	m	5	16	5+16=21
SUB [m]	01101	0D	m	5	16	5+16=21
INC A	01110	0E	-	5	-	5
JP +/-n	01111	0F	n	5	8	5+8=13
JZ +/-n	10000	10	n	5	8	5+8=13
JN +/-n	10001	11	n	5	8	5+8=13
STOP	10010	12	-	5	-	5

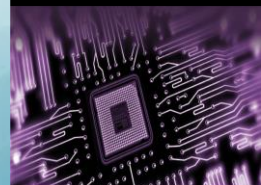


# Ανάπτυξη προγράμματος

Βάσει των υποθετικών εντολών θα αναπτύξουμε ένα πρόγραμμα που θα διαβάζει ένα ακέραιο αριθμό ( $X$ ) από το πληκτρολόγιο και στη συνέχεια θα υπολογίζει το άθροισμα  $3X+5X$

**RD** Διάβασμα ακέραιου αριθμού ( $X$ ) από το πληκτρολόγιο ( $A=X$ )  
**MOV [AF00],A** Αποθήκευση αριθμού στη θέση AF00 ( $[AF00]=A$ )  
**MOV A,3** Φόρτωση του αριθμού 3 στον καταχωρητή A ( $A=3$ )  
**MUL [AF00]** Πολλαπλασιασμός A με το περιεχόμενο της θέσης AF00 ( $A=A*[AF00]$ )  
**MOV [AF01],A** Αποθήκευση A (αποτέλεσμα πολλαπλασιασμού  $3X$ ) στη θέση AF01  
**MOV A,5** Φόρτωση του αριθμού 5 στον καταχωρητή A ( $A=5$ )  
**MUL [AF00]** Πολλαπλασιασμός A με το περιεχόμενο της θέσης AF00 ( $A=A*[AF00]$ )  
**MOV [AF00],A** Αποθήκευση A (αποτέλεσμα πολλαπλασιασμού  $5X$ ) στη θέση AF00  
**MOV A,[AF01]** Φόρτωση του αριθμού  $3X$  στον καταχωρητή A (θέση AF01)  
**MOV B,[AF00]** Φόρτωση του αριθμού  $5X$  στον καταχωρητή A (θέση AF01)  
**ADD B** Υπολογισμός  $3X+5X$  ( $A=A+B$ )  
**WR** Εμφάνιση αποτελέσματος στην οθόνη  
**STOP** Τερματισμός προγράμματος

- Πως αλλάζει το περιεχόμενο καταχωρητών και μνήμης καθώς εκτελούνται οι εντολές;
- Ποια η μορφή του προγράμματος στη μνήμη;



# Διαδικασία εκτέλεσης προγράμματος <sup>(1)</sup>

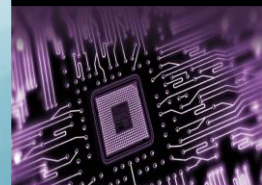
## **Μεταβλητό μήκος εντολών** (συστήματα τύπου CISC)

- (α) Ανάκληση κώδικα εντολής (διάβασμα κώδικα εντολής και μεταφορά στον μικροεπεξεργαστή)
- (β) Αποκωδικοποίηση (υπόδειξη τύπου εντολής και ορισμάτων)
- (γ) Ανάκληση ορισμάτων (διάβασμα ορισμάτων και μεταφορά στον μικροεπεξεργαστή)
- (δ) Εκτέλεση εντολής (ολοκλήρωση διαδικασίας εκτέλεσης)

## **Σταθερό μήκος εντολών** (συστήματα τύπου RISC)

- (α) Ανάκληση εντολής (κώδικας και ορίσματα)
- (β) Αποκωδικοποίηση και εκτέλεση εντολής

Το υποθετικό σύστημα που εξετάζουμε σε αυτό το κεφάλαιο υποστηρίζει εντολές μεταβλητού μήκους



# Διαδικασία εκτέλεσης προγράμματος (2)

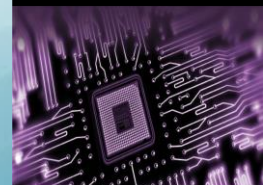
Εντολή υπό εκτέλεση: **ADD 5,2**

Βήμα 1

## Έναρξη εκτέλεσης προγράμματος

Καταχωρητής	Περιεχόμενο (δυαδικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	<b>1010111100000000</b>	<b>AF00</b>
IR		
MAR		
MBR		

Το υποθετικό μας πρόγραμμα ξεκινά στη διεύθυνση AF00



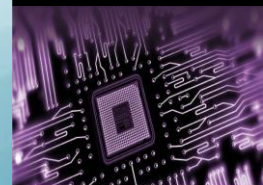
# Διαδικασία εκτέλεσης προγράμματος <sup>(3)</sup>

Εντολή υπό εκτέλεση: **ADD 5,2**

Βήμα 2

Έναρξη διαδικασίας μεταφοράς κώδικα εντολής		
Καταχωρητής	Περιεχόμενο (δυαδικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	1010111100000000	AF00
IR		
MAR	<b>1010111100000000</b>	<b>AF00</b>
MBR		

Ενεργοποίηση διεύθυνσης AF00 για μεταφορά του κώδικα εντολής



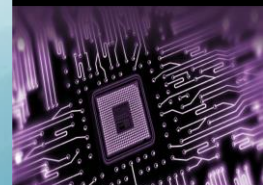
# Διαδικασία εκτέλεσης προγράμματος (4)

Εντολή υπό εκτέλεση: **ADD 5,2**

Βήμα 3

Μεταφορά του κώδικα εντολής		
Καταχωρητής	Περιεχόμενο (δυαδικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	1010111100000000	AF00
IR		
MAR	1010111100000000	AF00
MBR	<b>00001010</b>	<b>0A</b>

Μεταφορά του κώδικα εντολής (0A) μέσω του MBR



# Διαδικασία εκτέλεσης προγράμματος (5)

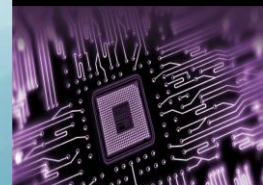
Εντολή υπό εκτέλεση: **ADD 5,2**

## Βήμα 4

### Προετοιμασία διαδικασίας αποκωδικοποίησης

Καταχωρητής	Περιεχόμενο (δυαδικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	1010111100000000	AF00
IR	<b>00001010</b>	<b>0A</b>
MAR	1010111100000000	AF00
MBR	00001010	<b>0A</b>

Φόρτωση του κώδικα εντολής στον καταχωρητή IR για αποκωδικοποίηση



# Διαδικασία εκτέλεσης προγράμματος <sup>(6)</sup>

Εντολή υπό εκτέλεση: **ADD 5,2**

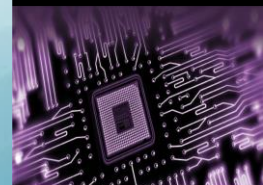
Βήμα 5

Αποκωδικοποίηση εντολής

Βήμα 6

Ενημέρωση του Μετρητή Προγράμματος		
Καταχωρητής	Περιεχόμενο (δυναμικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	<b>1010111100000011</b>	<b>AF03</b>
IR	00001010	0A
MAR	<b>1010111100000001</b>	<b>AF01</b>
MBR	00001010	0A

Η ενημέρωση του ΜΠ γίνεται βάσει του μήκους της τρέχουσας εντολής





# Διαδικασία εκτέλεσης προγράμματος (7)

Εντολή υπό εκτέλεση: **ADD 5,2**

Βήμα 7

## Μεταφορά πρώτου ορίσματος από τη μνήμη

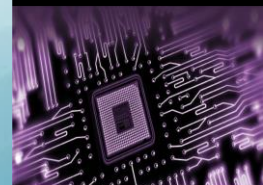
Καταχωρητής	Περιεχόμενο (δυναδικό)	Περιεχόμενο (δεκαεξαδικό)
A		
B		
PC	1010111100000011	AF03
IR	00001010	0A
MAR	1010111100000001	AF01
MBR	<b>00000101</b>	<b>05</b>

## Αποθήκευση του ορίσματος στον καταχωρητή A

Καταχωρητής	Περιεχόμενο (δυναδικό)	Περιεχόμενο (δεκαεξαδικό)
A	<b>00000101</b>	<b>05</b>
B		
PC	1010111100000011	AF03
IR	00001010	0A
MAR	1010111100000001	AF01
MBR	00000101	05

Μικροεξεργαστές

Αρχές & Εφαρμογές



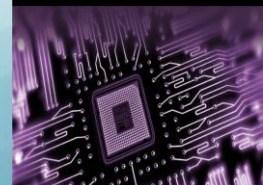
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζάβλα

# Διαδικασία εκτέλεσης προγράμματος <sup>(8)</sup>

Εντολή υπό εκτέλεση: **ADD 5,2**

Βήμα 8

Μεταφορά και αποθήκευση δεύτερου ορίσματος στον καταχωρητή B		
Καταχωρητής	Περιεχόμενο (δυναμικό)	Περιεχόμενο (δεκαεξαδικό)
A	00000101	05
B	<b>00000010</b>	<b>02</b>
PC	1010111100000011	AF03
IR	00001010	0A
MAR	1010111100000010	AF02
MBR	00000010	02

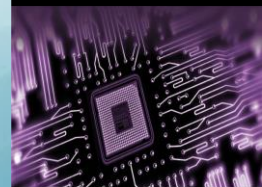


# Διαδικασία εκτέλεσης προγράμματος <sup>(9)</sup>

Εντολή υπό εκτέλεση: **ADD 5,2**

## Βήμα 9

Μέσω του εσωτερικού διαύλου δεδομένων, αντίγραφα των A και B μεταφέρονται στις εισόδους της ALU και εκτελείται η πρόσθεση



# Διαδικασία εκτέλεσης προγράμματος <sup>(10)</sup>

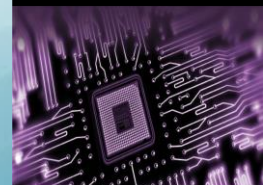
Εντολή υπό εκτέλεση: **ADD 5,2**

## Βήμα 10

Αποθήκευση αποτελέσματος στον καταχωρητή A		
Καταχωρητής	Περιεχόμενο (δυαδικό)	Περιεχόμενο (δεκαεξαδικό)
A	00000111	<b>07</b>
B	00000010	02
PC	1010111100000011	AF03
IR	00001010	0A
MAR	1010111100000010	AF02
MBR	00000010	02

## Βήμα 11

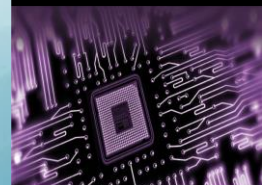
Η διαδικασία τώρα συνεχίζεται με τον ίδιο τρόπο από το βήμα 2



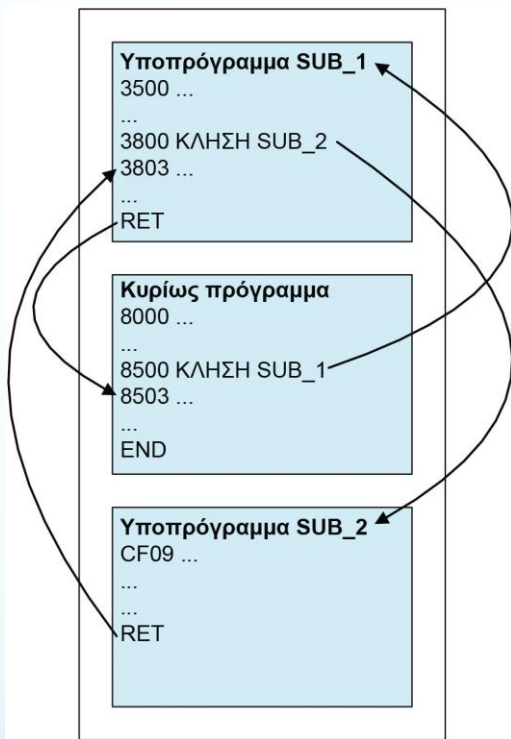
# Διαδικασία εκτέλεσης προγράμματος (11)

**Εντολή υπό εκτέλεση: ADD 5,2 – Η διαδικασία συνολικά**

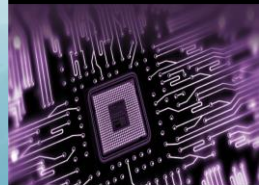
Βήμα	1	2	3	4	5	6	7	8	9	10	
<b>Κατ</b>											
<b>A</b>					Αποκωδικοποίηση		05		Εκτέλεση Εντολής	07	
<b>B</b>								02			
<b>PC</b>	AF00						AF03				
<b>IR</b>				0A							
<b>MAR</b>		AF00					AF01			AF02	
<b>MBR</b>			0A					05		02	
	Προσκόμιση κώδικα εντολής						Προσκόμιση ορισμάτων				



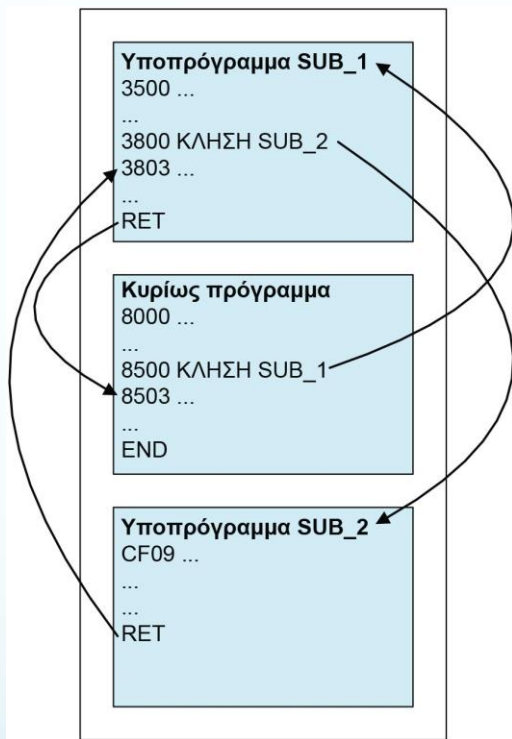
# Ο ρόλος του σωρού στην εκτέλεση προγράμματος



- ❑ Θα μελετήσουμε τη χρήση του σωρού στην εκτέλεση υποπρογραμμάτων
- ❑ Για λόγους απλότητας δεν εξετάζουμε πιθανά ορίσματα
- ❑ Θα δούμε πώς εξασφαλίζεται η ομαλή επιστροφή από υποπρογράμματα μέσω του σωρού



# Χρήση του σωρού κατά την κλήση και επιστροφή από υποπρογράμματα <sup>(1)</sup>



## Βήμα 1

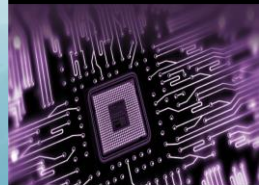
### Η εκτέλεση του Προγράμματος ξεκινά

```
F000 [           ]  
F002 [           ]  
F004 [  XXXX    ] <== ΔΣ
```

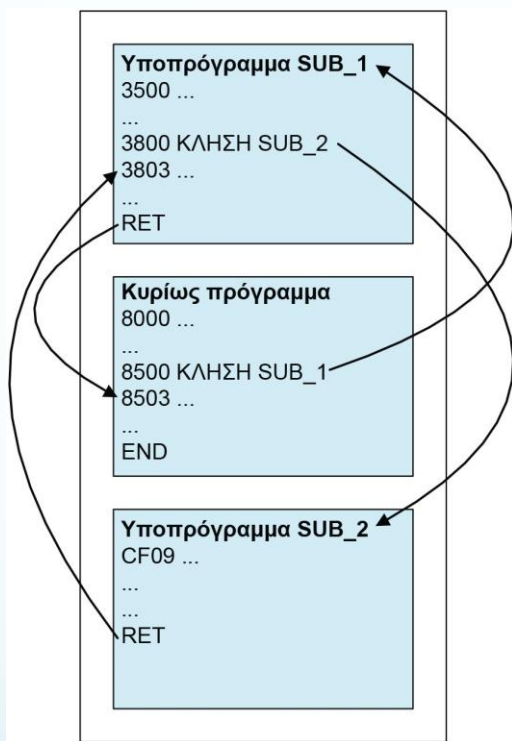
## Βήμα 2

### Προετοιμασία κλήσης υποπρογράμματος SUB\_1

```
F000 [           ]  
F002 [  8503    ] <== ΔΣ  
F004 [  XXXX    ]
```



## Χρήση του σωρού κατά την κλήση και επιστροφή από υποπρογράμματα (2)



### Βήμα 3

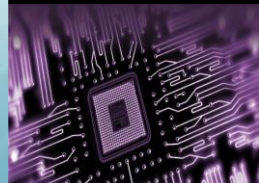
#### Προετοιμασία κλήσης υποπρογράμματος SUB\_2

F000	[	3803	]	<==	ΔΣ
F002	[	8503	]		
F004	[	XXXX	]		

### Βήμα 4

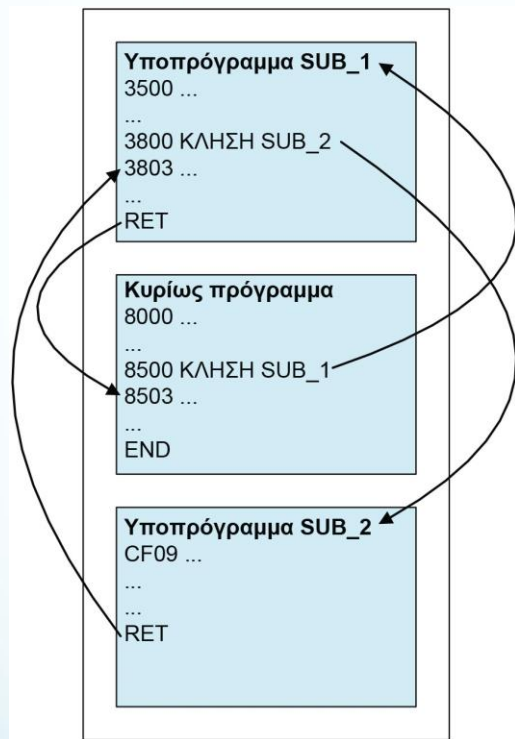
#### Εκτέλεση εντολής RET (SUB\_2)

F000	[		]		
F002	[	8503	]	<==	ΔΣ
F004	[	XXXX	]		





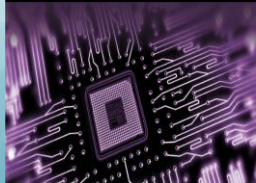
# Χρήση του σωρού κατά την κλήση και επιστροφή από υποπρογράμματα <sup>(3)</sup>



## Βήμα 5

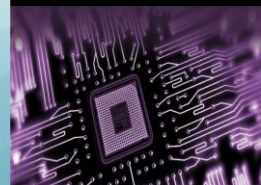
### Εκτέλεση εντολής RET (SUB\_1)

```
F000 [           ]  
F002 [           ]  
F004 [  XXXX    ] <== ΔΣ
```



# Μέθοδοι διευθυνσιοδότησης στη συμβολική γλώσσα

Μορφή εντολής	Διευθυνσιοδότηση	Λειτουργία
MOV A,n	Απευθείας	$A=n$
MOV A,[m]	Άμεση	$A=[m]$ Φόρτωση στο A του περιεχομένου της διεύθυνσης m
MOV A,[B]	Έμμεση με καταχωρητή	$A=[B]$ Φόρτωση στο A του περιεχομένου της διεύθυνσης που περιέχεται στον καταχωρητή B
MOV A,[B+x]	Έμμεση με καταχωρητή και απόκλιση	$A=[B+x]$ Φόρτωση στο A του περιεχομένου της διεύθυνσης που προσδιορίζεται από το περιεχόμενο του καταχωρητή B συν την τιμή του x
MOV A,Ετικέτα [x]	Άμεση με Ετικέτα και απόκλιση	$A=Ετικέτα+x$ Φόρτωση στο A του περιεχομένου της διεύθυνσης που προσδιορίζεται από την Ετικέτα με απόκλιση x



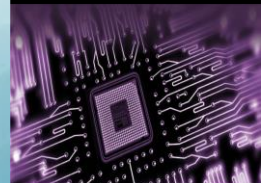
# Βελτίωση της εκτέλεσης προγράμματος (1)

## Διαδικασία εκτέλεσης εντολών

- Προσκόμιση κώδικα εντολής
- Αποκωδικοποίηση
- Προσκόμιση ορισμάτων
- Εκτέλεση εντολής

Κλασικός τρόπος εκτέλεσης εντολών

4 βήματα για την εκτέλεση εντολής												
1	2	3	4	1	2	3	4		1	2	3	4
Εντολή-1				Εντολή-2				...	Εντολή-n			
Χρόνος ==> +												



## Βελτίωση της εκτέλεσης Προγράμματος (2)

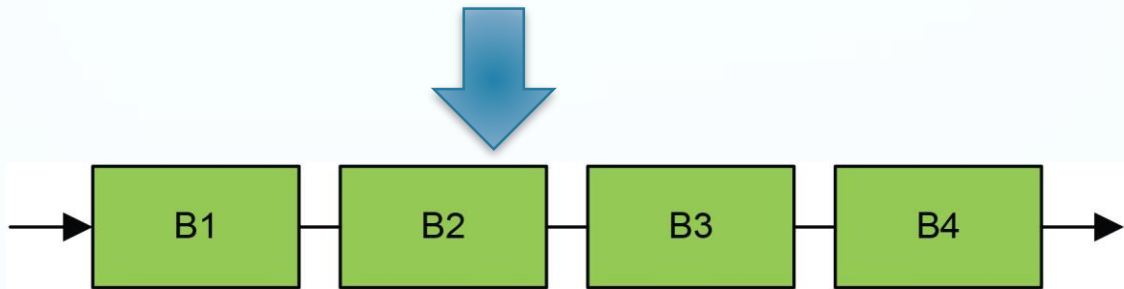
### Η τεχνική του αγωγού

**B1:** Προσκόμιση κώδικα εντολής

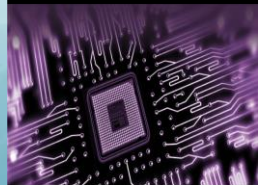
**B2:** Αποκωδικοποίηση

**B3:** Προσκόμιση ορισμάτων

**B4:** Εκτέλεση εντολής



Υλοποίηση λειτουργιών από ξεχωριστές βαθμίδες



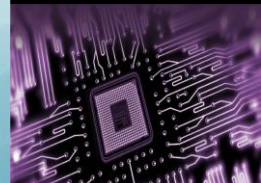
# Βελτίωση της εκτέλεσης Προγράμματος (3)

Εκτέλεση εντολής με την τεχνική του αγωγού

## Βήμα 1

Βαθμίδα								
B1	ΠΚΕ1							
B2								
B3								
B4								
Χρόνος	$t_0$	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Η εντολή 1 εισέρχεται στον αγωγό  
(Προσκόμιση Κώδικα Εντολής 1)



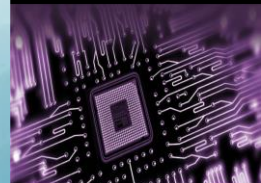
# Βελτίωση της εκτέλεσης Προγράμματος (4)

## Εκτέλεση εντολής με την τεχνική του αγωγού

### Βήμα 2

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2						
B2		ΑΕ1						
B3								
B4								
Χρόνος	$t_0$	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Η εντολή 1 περνάει στη φάση της αποκωδικοποίησης (Αποκωδικοποίηση Εντολής 1), ενώ απελευθερώνει τη βαθμίδα B1. Στη βαθμίδα B1 εισέρχεται τώρα η εντολή 2 (Προσκόμιση Κώδικα Εντολής 2)



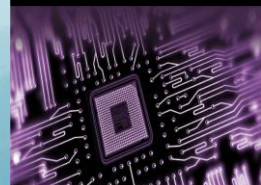
# Βελτίωση της εκτέλεσης Προγράμματος (5)

## Εκτέλεση εντολής με την τεχνική του αγωγού

### Βήμα 3

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3					
B2		ΑΕ1	ΑΕ2					
B3			ΠΟΕ1					
B4								
Χρόνος	$t_0$	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Η εντολή 1 περνάει στη φάση της προσκόμισης ορισμάτων (**ΠΟΕ1**), ενώ απελευθερώνει τη βαθμίδα B2 στην οποία εισέρχεται η εντολή 2 (**Αποκωδικοποίηση Εντολής 2**) η οποία απελευθερώνει τη βαθμίδα B1. Στη βαθμίδα B1 εισέρχεται τώρα η εντολή 3 (**Προσκόμιση Κώδικα Εντολής 3**)



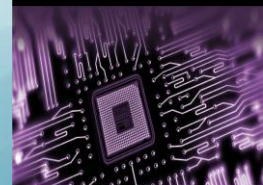
## Βελτίωση της εκτέλεσης Προγράμματος (6)

### Εκτέλεση εντολής με την τεχνική του αγωγού

#### Βήμα 4

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3	ΠΚΕ4				
B2		ΑΕ1	ΑΕ2	ΑΕ3				
B3			ΠΟΕ1	ΠΟΕ2				
B4				ΕΚΕ1				
Χρόνος	$t_0$	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
B1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3	ΠΚΕ4				

- Κάθε εντολή που περνά στο επόμενο στάδιο απελευθερώνει την προηγούμενη βαθμίδα. Έτσι, όσες ακολουθούν πηγαίνουν στην επόμενη βαθμίδα. Σε αυτή τη φάση, θα εισέλθει στη βαθμίδα B1 και η εντολή 4.
- Αυτή τη φορά θα έχουμε και το πρώτο αποτέλεσμα (ΕΚτέλεση Εντολής 1)



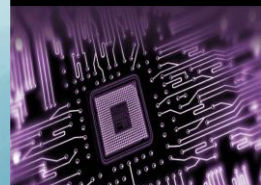


# Βελτίωση της εκτέλεσης Προγράμματος (7)

## Εκτέλεση εντολής με την τεχνική του αγωγού

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3	ΠΚΕ4	ΠΚΕ5			
B2		ΑΕ1	ΑΕ2	ΑΕ3	ΑΕ4			
B3			ΠΟΕ1	ΠΟΕ2	ΠΟΕ3			
B4				ΕΚΕ1	ΕΚΕ2			
<b>Χρόνος</b>	$t_0$	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα				1 <sub>ο</sub>	2 <sub>ο</sub>			

Στον επόμενο χρόνο θα έχουμε αποτέλεσμα και από τη δεύτερη εντολή

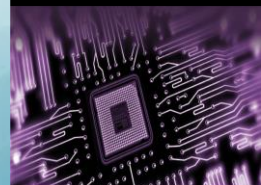


## Βελτίωση της εκτέλεσης προγράμματος (8)

### Σύγκριση συνολικών χρόνων εκτέλεσης

Ακολουθιακή εκτέλεση	$T_{ακ}(ε,ν) = ν * ρ(ε) * t_0$
Τεχνική αγωγού	$T_{αν}(ε,ν) = (α(ε) - 1 + ρ(ε) + ν) t_0$
Υπολογιστικές μονάδες	$T_{κ}(ε,ν) = (ν/k) * ρ(ε) * t_0$

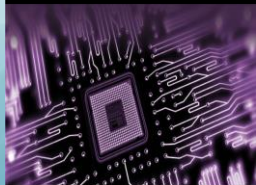
Στην τρίτη περίπτωση θεωρούμε ότι έχουμε  $k$  υπολογιστικά συστήματα ή  $k$  ΚΜΕ



# Κεφάλαιο 7

## Τα κύρια σημεία σε τίτλους

- ❑ Κάθε εντολή σε συμβολική γλώσσα αποτελείται από τον κώδικα και τα ορίσματα
- ❑ Το πρόγραμμα σε συμβολική γλώσσα στην κεντρική μνήμη
- ❑ Εκτέλεση Προγράμματος
- ❑ Ο ρόλος του σωρού κατά την κλήση υποπρογραμμάτων
- ❑ Τεχνικές διευθυνσιοδότησης
- ❑ Η τεχνική του αγωγού

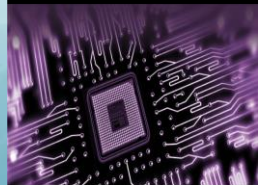


# Κεφάλαιο 8

## Μνήμη

**Μικροεξεργαστές**

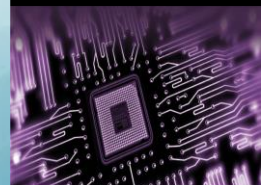
Αρχές & Εφαρμογές



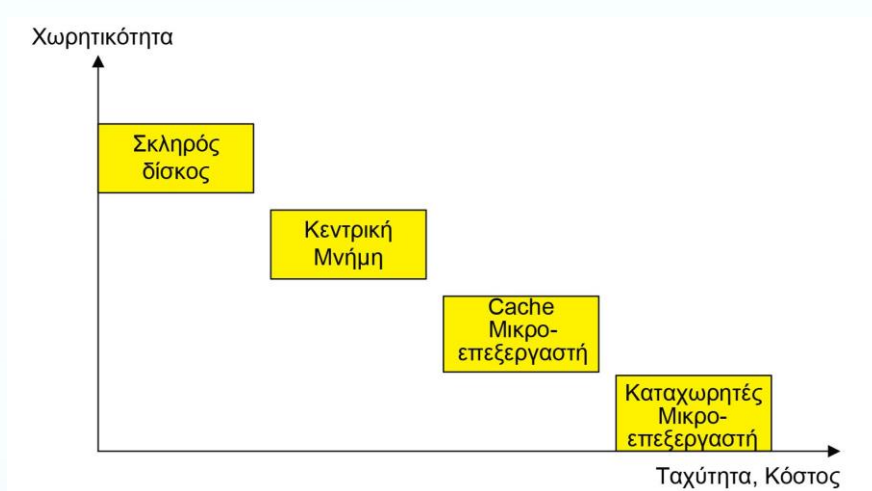
Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

# Συστήματα και στοιχεία μνήμης

Στοιχείο μνήμης	Επίπεδο	Ταχύτητα	Χωρητικότητα	Κόστος
Καταχωρητές μικροεπεξεργαστή	0	Υψηλή	Μικρή	Υψηλό
Cache μικροεπεξεργαστή	1	Υψηλή	Μικρή	Υψηλό
Κεντρική μνήμη	2	Χαμηλή	Μεγάλη	Χαμηλό
Περιφερειακή μνήμη (π.χ. σκληρός δίσκος)	3	Χαμηλή	Μεγάλη	Χαμηλό

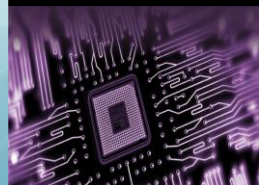


# Σχέση χωρητικότητας, ταχύτητας και κόστους



Μικροεπεξεργαστές

Αρχές & Εφαρμογές



Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

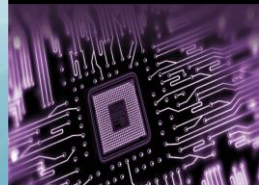
# Μνήμη Cache <sup>(1)</sup>

- ❑ Συνήθως εντός του μικροεπεξεργαστή
- ❑ Χαμηλή χωρητικότητα, υψηλές επιδόσεις
- ❑ Χρήσιμη = όταν περιέχει σημαντικές πληροφορίες
- ❑ Βελτίωση χρόνου εκτέλεσης = Ανάκληση εντολών από την cache και όχι από την κεντρική μνήμη
- ❑ Κρίσιμη απόφαση = ποιες εντολές του προγράμματος θα πρέπει να περιέχει

## **Ενημέρωση μνήμης cache βάσει της αρχής της τοπικότητας**

**Χωρική τοπικότητα** : όταν γίνεται αναφορά σε μια συγκεκριμένη διεύθυνση (για ανάκτηση περιεχομένου), είναι πολύ πιθανό να ζητηθούν και οι κοντινές διευθύνσεις στην αμέσως επόμενη χρονική περίοδο

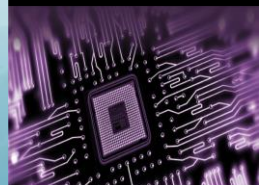
**Προσωρινή τοπικότητα** : η ίδια διεύθυνση (ανάκτηση του περιεχομένου) μπορεί να ζητηθεί ξανά (π.χ. σε μια περίπτωση βρόχου επανάληψης)



# Μνήμη Cache (2)

## Η λειτουργία στην πράξη

- ❑ Ένα τμήμα από την κεντρική μνήμη αντιγράφεται στην cache
- ❑ Το αίτημα του μικροεπεξεργαστή για περιεχόμενο θέσης μνήμης, μεταφέρεται αρχικά στην cache
- ❑ Επιτυχής αναφορά στην cache = hit, διαφορετικά miss
- ❑ Στην περίπτωση miss, το αίτημα μεταφέρεται στην κεντρική μνήμη
- ❑ Ένα τμήμα της κεντρικής μνήμης (μαζί με τη ζητούμενη πληροφορία) μεταφέρεται στην cache





# Μνήμη Cache <sup>(3)</sup>

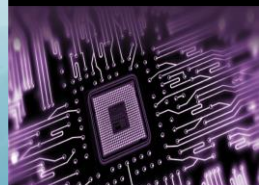
## Οργάνωση

### Κριτήρια

- ❑ Για την τοποθέτηση ενός μπλοκ στην cache από την κεντρική μνήμη
- ❑ Για την αντικατάσταση ενός μπλοκ εντός της cache από ένα νέο μπλοκ της κεντρικής μνήμης

### Μέθοδοι

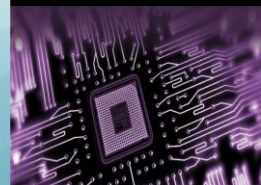
- ❑ Απευθείας χαρτογράφηση (σταθερή συσχέτιση μπλοκ cache/κεντρικής μνήμης)
- ❑ Πλήρως συσχετιζόμενη χαρτογράφηση (τοποθέτηση μπλοκ σε κάποιο διαθέσιμο της cache)
- ❑ Συσχετιζόμενη χαρτογράφηση συνόλου (διαχωρισμός cache σε επιμέρους σύνολα)



# Μνήμη Cache (4)

Στην απευθείας χαρτογράφηση ένα μπλοκ  $i$  της κεντρικής μνήμης που εισέρχεται στην Cache θα συσχετιστεί με το μπλοκ  $j$  της Cache σύμφωνα με τον τύπο  $j = i \bmod N$ , όπου  $N$  το πλήθος των μπλοκ στην Cache.

Παραδείγματα συχέτισης
$384 \bmod 128 = 0$
$129 \bmod 128 = 1$
$4095 \bmod 128 = 127$



# Κεντρική μνήμη

## Θέσεις και διευθύνσεις

Διεύθυνση	Περιεχόμενο
00	
01	
02	
03	
04	
05	
06	
07	
08	
09	

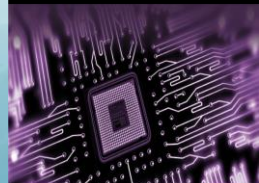
$N$  το πλήθος bit για την αποτύπωση κάθε διεύθυνσης ( $M$ =πλήθος θέσεων μνήμης)

$$2^N \geq M$$

Για μεγάλο  $M$  χρησιμοποιούμε τον επόμενο τύπο (Το  $N$  στρογγυλοποιημένο προς τα πάνω)

$$N = \lceil \log_2 M \rceil$$

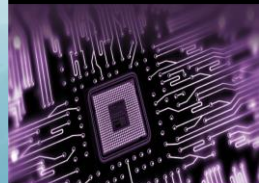
Υποθετικό μοντέλο 10 θέσεων



# Γενικά χαρακτηριστικά της κεντρικής μνήμης (1)



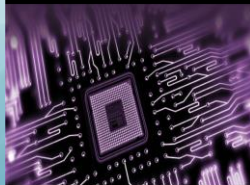
**Ανάγνωση του περιεχομένου της θέσης AF00**



# Γενικά χαρακτηριστικά της κεντρικής μνήμης (2)

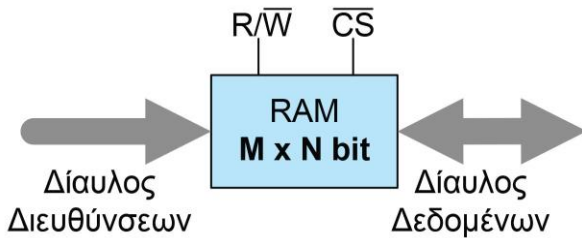
## Λειτουργίες κυκλώματος μνήμης (μέσω σημάτων)

- ❑ Επιλογή διεύθυνσης μνήμης για εγγραφή / ανάγνωση
- ❑ Επιλογή λειτουργίας (ανάγνωση / εγγραφή)
- ❑ Τροφοδοσία με δεδομένα προς αποθήκευση
- ❑ Δέσμευση ή απελευθέρωση μνήμης

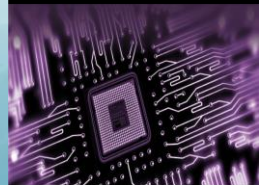
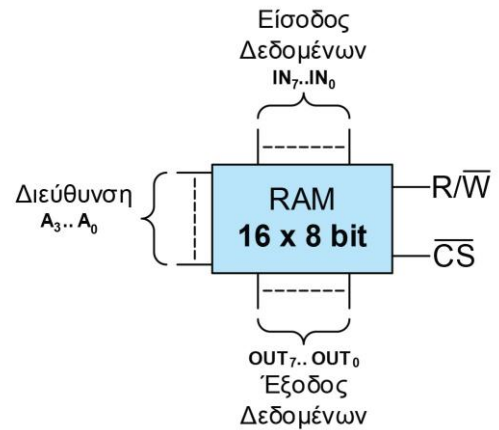


# Γενικά χαρακτηριστικά της κεντρικής μνήμης (3)

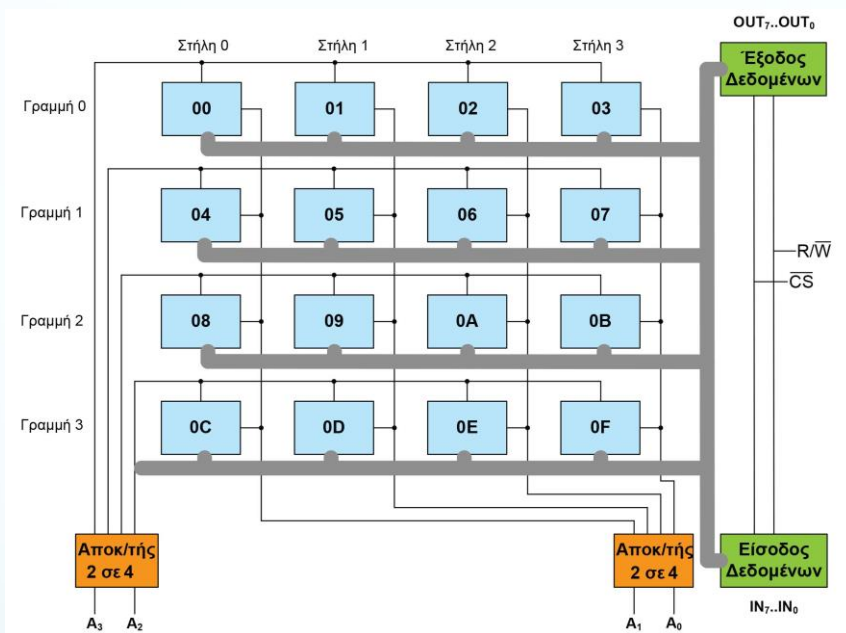
Γενική μορφή στοιχείου μνήμης RAM



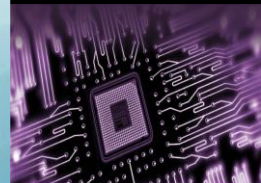
Στοιχείο μνήμης RAM 16x8 bit



# Εσωτερική αρχιτεκτονική στοιχείου μνήμης RAM 16x8bit



- ❑ Κάθε θέση έχει συγκεκριμένες συντεταγμένες (γραμμή, στήλη)
- ❑ Για να ενεργοποιηθεί μια θέση πρέπει να υπάρχει αντίστοιχο σήμα στη γραμμή και τη στήλη στην οποία ανήκει
- ❑ Για την ενεργοποίηση της θέσης 0B, θα ενεργοποιηθεί η γραμμή 2 και η στήλη 3
- ❑ Αυτό θα γίνει εφαρμόζοντας τα σήματα  $A_3=1$ ,  $A_2=0$ ,  $A_1=1$ ,  $A_0=1$  στις εισόδους των αποκωδικοποιητών
- ❑ Τα σήματα  $A_3, A_2, A_1, A_0$  προέρχονται από το δίαυλο διευθύνσεων

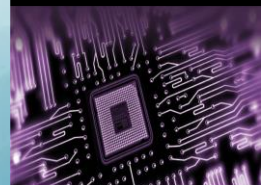


# Χαρτογράφηση μνήμης

Έστω ότι ένα σύστημα διαθέτει μνήμη που αποτελείται από δύο περιοχές RAM μεγέθους 8Kbyte η κάθε μία. Γνωρίζοντας ότι 1Kbyte=1024byte μπορούμε να βρούμε τις οριακές διευθύνσεις των αντίστοιχων περιοχών.

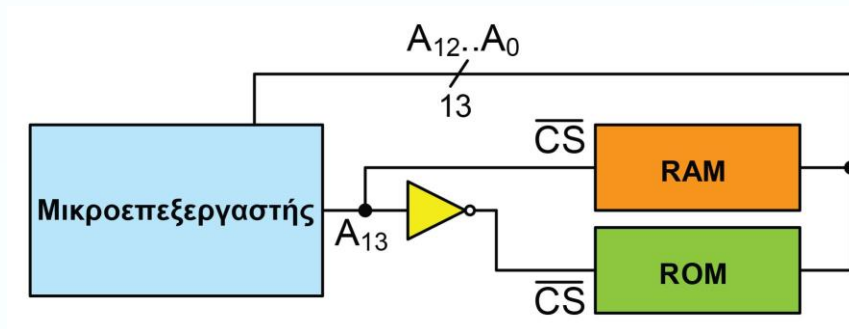
Οριακές διευθύνσεις	Περιοχή
0	8Kbyte
$(8 \cdot 1024) - 1 = 8191$	
8192	8Kbyte
$(16 \cdot 1024) - 1 = 16383$	

	Περιοχή
0 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
0 1 1 1 1 1 1 1 1 1 1 1 1 1	
1 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
1 1 1 1 1 1 1 1 1 1 1 1 1 1	



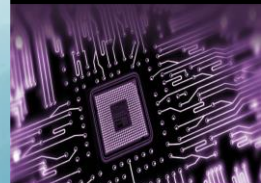


## Κύκλωμα διευθυνσιοδότησης



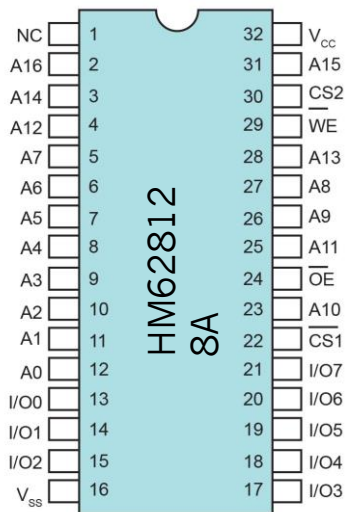
Το περισσότερο σημαντικό bit ( $A_{13}$ ) θα χρησιμοποιηθεί για την επιλογή περιοχής μνήμης

	Περιοχή
0 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
0 1 1 1 1 1 1 1 1 1 1 1 1 1	
1 0 0 0 0 0 0 0 0 0 0 0 0 0	8Kbyte
1 1 1 1 1 1 1 1 1 1 1 1 1 1	



# Πραγματικό Πείραμα στο εργαστήριο (1)

Ολοκληρωμένο κύκλωμα μνήμης 128K x 8bit

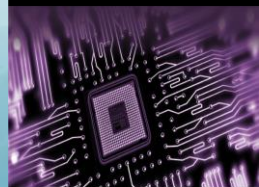


## Ακροδέκτες ολοκληρωμένου

Ακροδέκτης	Περιγραφή
A0-A16	Διεύθυνση
I/O0-I/O7	Είσοδος/Εξοδος δεδομένων
CS1'	Chip Select1 (ενεργό στο 0)
CS2	Chip Select2 (ενεργό στο 1)
WE'	Write Enable (ενεργοποίηση εγγραφής)
OE'	Output Enable (ενεργοποίηση ανάγνωσης)
NC	Δεν χρησιμοποιείται
V <sub>cc</sub>	Τροφοδοσία (5V)
V <sub>ss</sub>	Γείωση

Μικροεξεργαστές

Αρχές & Εφαρμογές



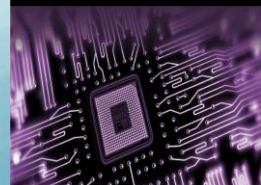
Παναγιώτης Παπάζογλου

Εκδόσεις Τζόλα

## Πραγματικό Πείραμα στο εργαστήριο (2)

Λειτουργία κυκλώματος μνήμης 128K x 8bit

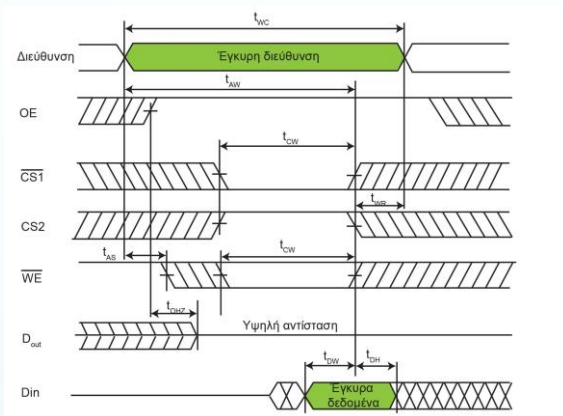
Συμπεριφορά κυκλώματος (X=αδιάφορο)					
CS1'	CS2	OE'	WE'	Τρόπος λειτουργίας	Ακροδέκτες I/O
H	X	X	X	Σε αναμονή	Υψηλή αντίσταση
X	L	X	X	Σε αναμονή	Υψηλή αντίσταση
L	H	H	H	Έξοδος απενεργοποιημένη	Υψηλή αντίσταση
L	H	L	H	Ανάγνωση	Δεδομένα εξόδου
L	H	H	L	Εγγραφή	Δεδομένα εισόδου
L	H	L	L	Εγγραφή	Δεδομένα εισόδου



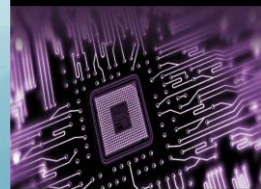
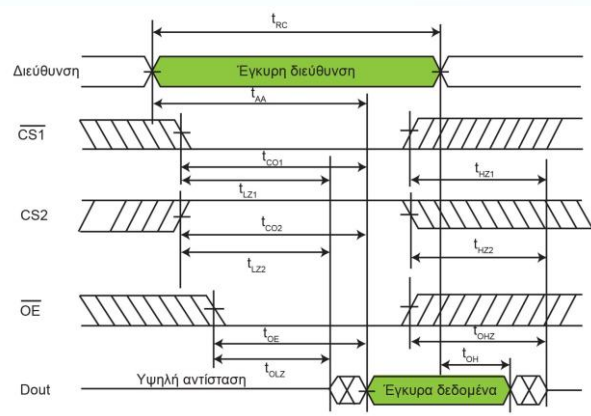
# Πραγματικό πείραμα στο εργαστήριο (3)

## Χρονισμός

### Εγγραφή

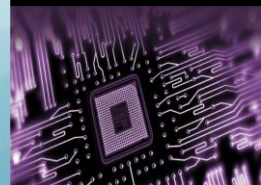
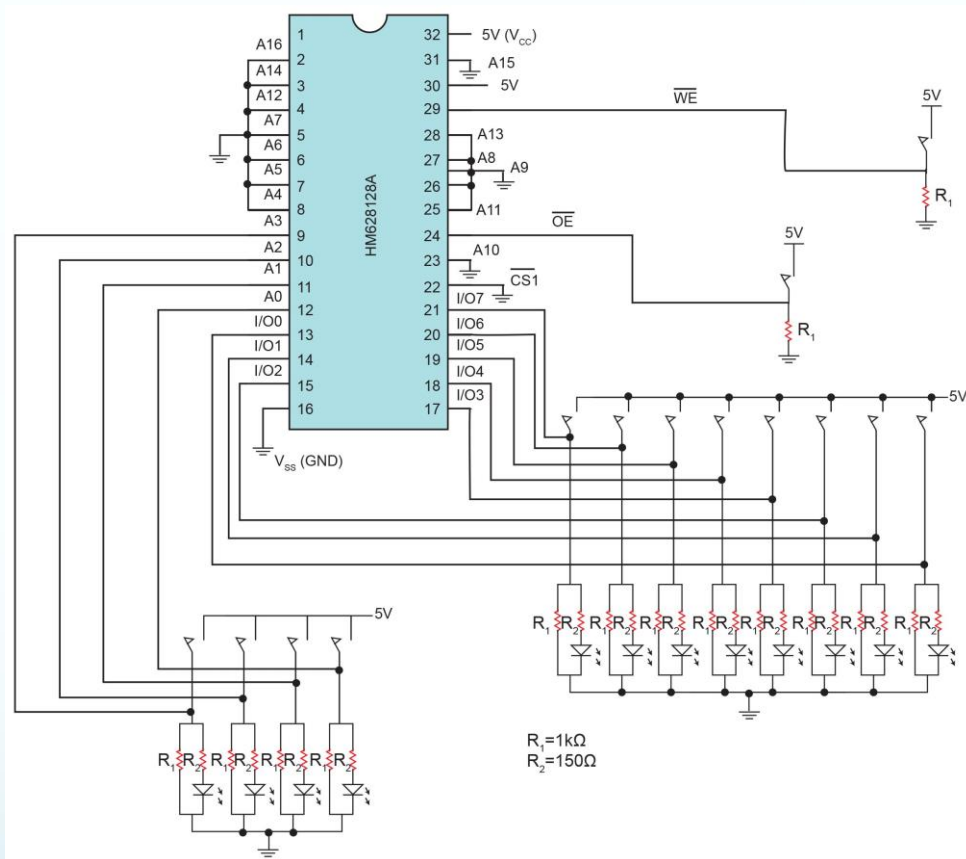


### Ανάγνωση



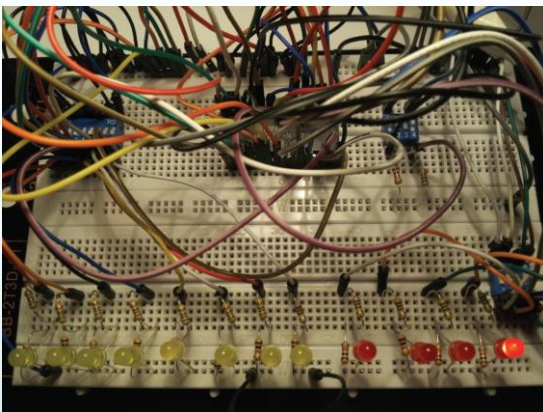
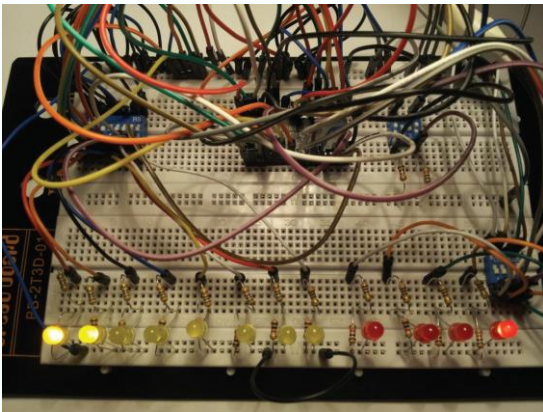
# Πραγματικό Πείραμα στο εργαστήριο (4)

## Δοκιμαστικό κύκλωμα

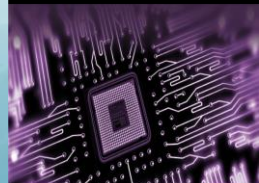


## Πραγματικό Πείραμα στο εργαστήριο (5)

Πραγματική μορφή δοκιμαστικού κυκλώματος

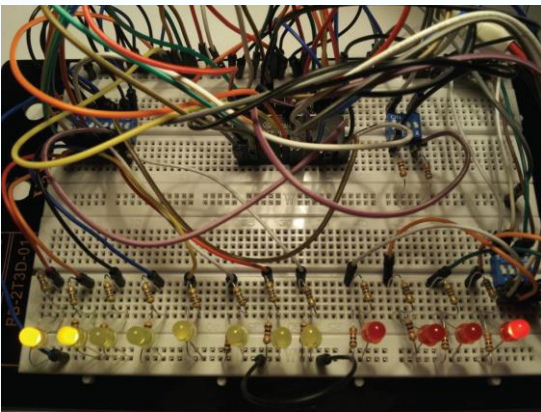


- ❑ **Εγγραφή** πληροφορίας 11000000 (κίτρινα LED, αριστερά) στη διεύθυνση 0001 (κόκκινα LED, δεξιά)
- ❑ Στη δοκιμή χρησιμοποιούμε μόνο 4bit από το δίαυλο διευθύνσεων
- ❑ Κατά την εγγραφή ενεργοποιούμε το σήμα WE' (διακόπτης 1 → 0 και επαναφορά)
- ❑ **Πριν την ανάγνωση** : όλοι οι διακόπτες απενεργοποιημένοι
- ❑ Εφαρμογή της διεύθυνσης 0001 (κόκκινα LED, δεξιά)
- ❑ Καμία πληροφορία στο δίαυλο δεδομένων (κίτρινα LED αριστερά)

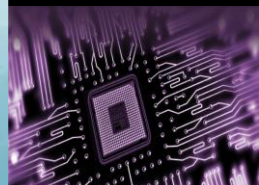


## Πραγματικό Πείραμα στο εργαστήριο (6)

Πραγματική μορφή δοκιμαστικού κυκλώματος



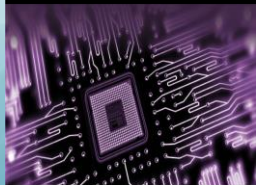
- ❑ **Ανάγνωση** πληροφορίας από τη διεύθυνση 0001 (κόκκινα LED, δεξιά)
- ❑ Για την ανάγνωση ενεργοποιούμε το σήμα OE' (διακόπτης 1 → 0)
- ❑ Στα κίτρινα LED αριστερά εμφανίζεται η αποθηκευμένη πληροφορία



# Κεφάλαιο 8

## Τα κύρια σημεία σε τίτλους

- ❑ Η μνήμη Cache
- ❑ Θέσεις και διευθύνσεις μνήμης
- ❑ Χαρακτηριστικά και λειτουργία μνήμης
- ❑ Χαρτογράφηση και διευθυνσιοδότηση
- ❑ Πραγματικό Πείραμα στο εργαστήριο





# Κεφάλαιο 9

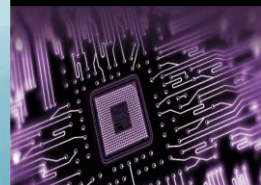
## ΕΠΙΚΟΙΝΩΝΙΑ

### ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗ ΜΕ

### ΕΞΩΤΕΡΙΚΕΣ ΣΥΣΚΕΥΕΣ

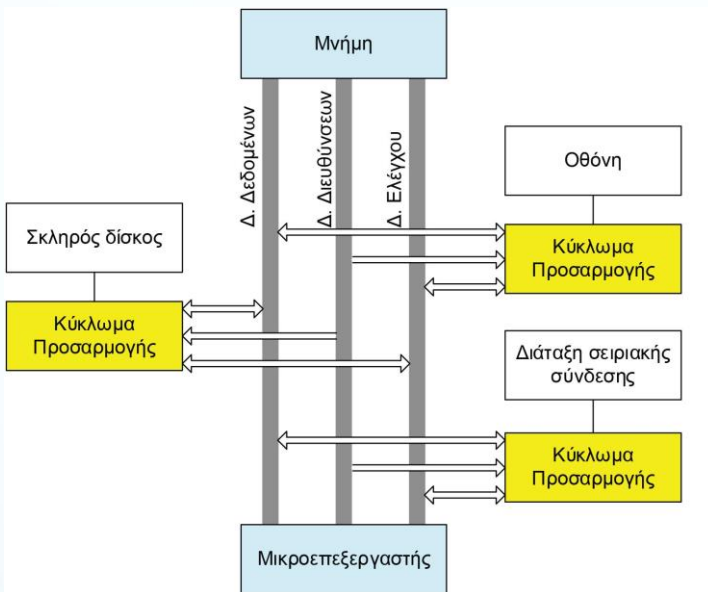
**Μικροεπεξεργαστές**

Αρχές & Εφαρμογές

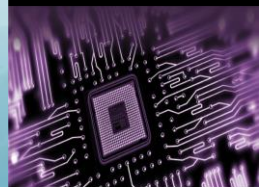


Παναγιώτης Παπάζογλου  
Εκδόσεις Τζιόλα

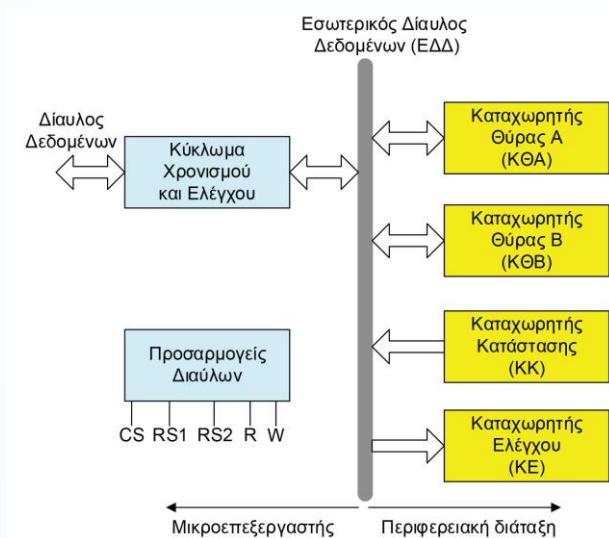
# Κυκλώματα προσαρμογής



- ❑ Απαιτείται μετατροπή σημάτων αφού οι περιφερειακές συσκευές είναι πολλές φορές ηλεκτρομηχανικά ή ηλεκτρομαγνητικά συστήματα και δεν μπορούν άμεσα να συνδεθούν με την κεντρική μνήμη ή τον μικροεπεξεργαστή
- ❑ Η κωδικοποίηση των δεδομένων από αυτές τις συσκευές διαφέρει από αυτή που χρησιμοποιεί ο μικροεπεξεργαστής
- ❑ Ο υπολογιστικός χρόνος του μικροεπεξεργαστή πρέπει να αξιοποιείται σωστά και η περιφερειακή συσκευή να εξυπηρετείται όταν πρέπει
- ❑ Πρέπει να επιτυγχάνεται ειδικός συγχρονισμός στην επικοινωνία επειδή οι συσκευές λειτουργούν σε χαμηλότερες ταχύτητες από τον μικροεπεξεργαστή



# Υποδειγματικό κύκλωμα προσαρμογής



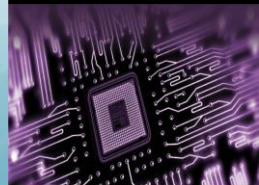
## Περιλαμβάνει

- ❑ καταχωρητές (που ονομάζονται θύρες)
- ❑ καταχωρητή ελέγχου
- ❑ καταχωρητή κατάστασης
- ❑ προσαρμογείς διαύλων
- ❑ κυκλώματα χρονισμού και ελέγχου

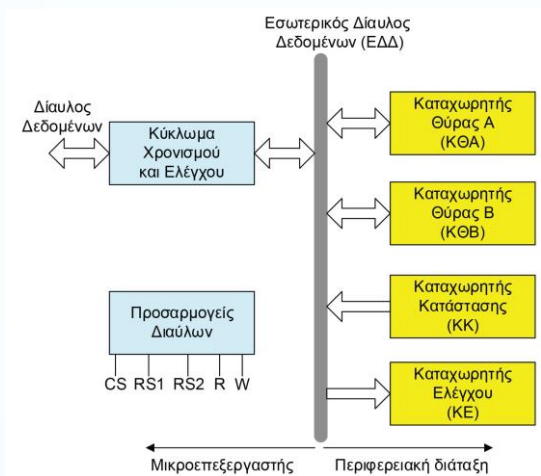
❑ ΚΠ = συνδεδεμένο με το διάυλο δεδομένων που εξασφαλίζει την ανταλλαγή δεδομένων (επικοινωνία) με τον μικροεπεξεργαστή του υπολογιστικού συστήματος

❑ CS (Chip Select – Ενεργοποίηση ολοκληρωμένου), RS (Register Select – ενεργοποίηση καταχωρητή) = καθορίζουν τη διεύθυνση που θα αποδοθεί στο ΚΠ

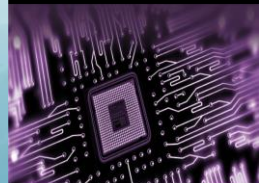
- ❑ W (εγγραφής), R (ανάγνωσης) = καθορίζουν αν θα πραγματοποιηθεί είσοδος ή έξοδος
- ❑ Οι τέσσερις καταχωρητές έχουν άμεση επικοινωνία με το κύκλωμα εισόδου/εξόδου του ΚΠ



# Υποδειγματικό κύκλωμα προσαρμογής

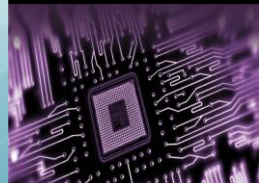
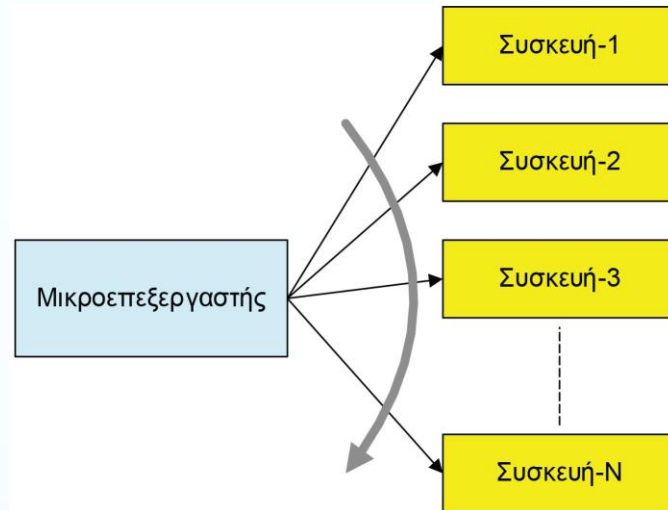


Επιλογή καταχωρητή ΚΠ			
RS1	RS0	CS	Επιλεγμένος καταχωρητής
0	0	1	ΚΘΑ
0	1	1	ΚΘΒ
1	0	1	ΚΕ
1	1	1	ΚΚ
X	X	0	Καμία επιλογή



# Τεχνικές επικοινωνίας <sup>(1)</sup>

## Τεχνική Polling



# Τεχνικές επικοινωνίας (2)

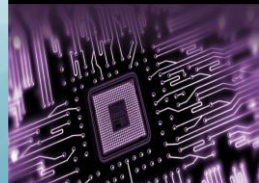
## Τεχνική Polling

### Πλεονεκτήματα

- ❑ Εύκολη σχεδίαση του υπολογιστικού συστήματος (σχεδίαση αντίστοιχου υλικού)
- ❑ Εύκολη σχεδίαση και υλοποίηση του προγράμματος Polling

### Μειονεκτήματα

- ❑ Χρειάζεται αλλαγή το πρόγραμμα Polling όταν προστεθεί μια νέα συσκευή
- ❑ Χρονική καθυστέρηση αφού ο μικροεπεξεργαστής πρέπει να σταματά το πρόγραμμα του χρήστη προκειμένου να ελέγξει μια συσκευή
- ❑ Όταν για μεγάλο χρονικό διάστημα δεν υπάρχουν καινούργια δεδομένα από τις συσκευές, τότε αυτός ο χρόνος είναι χαμένος και κυρίως έχει αφαιρεθεί από το διαθέσιμο χρόνο εκτέλεσης προγραμμάτων του χρήστη
- ❑ Η τεχνική αυτή βρίσκει εφαρμογή μόνο σε επικοινωνία με αργές συσκευές

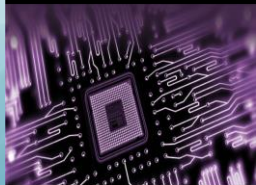


# Τεχνικές επικοινωνίας <sup>(3)</sup>

## Τεχνική των διακοπών (interrupts)

### Χαρακτηριστικά

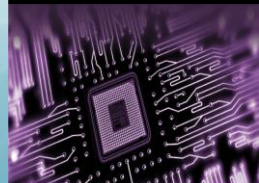
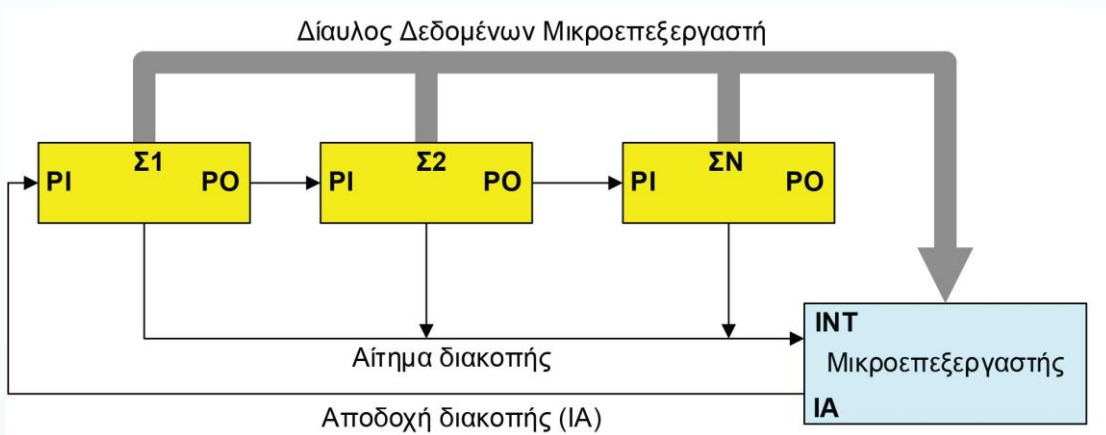
- ❑ Επικρατέστερη τεχνική στα σημερινά υπολογιστικά συστήματα
- ❑ Στόχος = αποφόρτιση μικροεπεξεργαστή από το συνεχή έλεγχο των εξωτερικών συσκευών
- ❑ Κάθε εξωτερική συσκευή ειδοποιεί η ίδια όταν χρειάζεται εξυπηρέτηση
- ❑ Ενεργοποιείται ειδικός ακροδέκτης του μικροεπεξεργαστή όταν κάποια συσκευή ζητήσει εξυπηρέτηση
- ❑ Υποστηρίζονται γρήγορες συσκευές
- ❑ Αξιοποιείται ο υπολογιστικός χρόνος για τα προγράμματα του χρήστη
- ❑ Πολυπλοκότητα κυκλωμάτων διασύνδεσης των εξωτερικών συσκευών που υποστηρίζουν διακοπές
- ❑ Προγραμματιζόμενος ελεγκτής διακοπών



# Προτεραιότητα διακοπών (1)

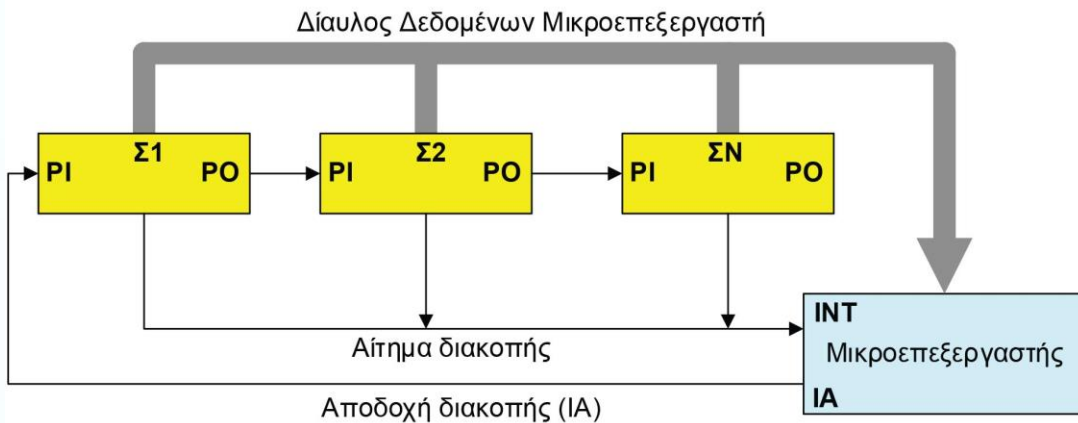
Σε ένα σύστημα με πολλές εξωτερικές συσκευές, πρέπει να καθορίζονται:

- ❑ ο αριθμός της συσκευής που ζήτησε εξυπηρέτηση
- ❑ η προτεραιότητα της εξυπηρέτησης
- ❑ ποιο πρόγραμμα θα ενεργοποιείται για την εξυπηρέτηση της συσκευής

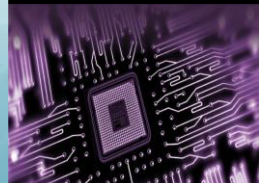




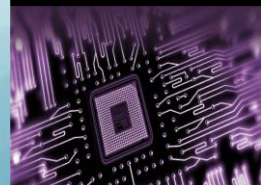
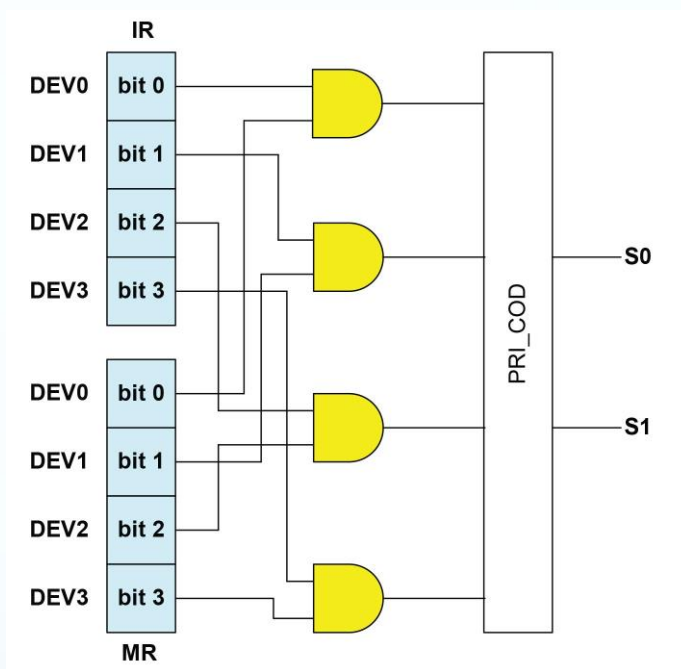
## Προτεραιότητα διακοπών (2)



- ❑ Η αποδοχή διακοπής (IA) μεταφέρεται από τον μικροεπεξεργαστή αποκλειστικά στην πρώτη συσκευή (Σ1)
- ❑ Με τη λήψη του σήματος IA, εξυπηρετείται η Σ1. Αν η Σ1 δεν έχει κάνει αίτημα διακοπής, τότε το σήμα IA μεταφέρεται στην επόμενη συσκευή (PO → PI)
- ❑ Εξυπηρετείται πάντα η πρώτη συσκευή στη σειρά που έχει κάνει αίτημα διακοπής

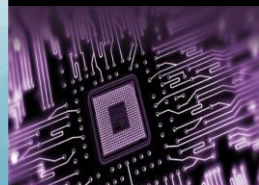
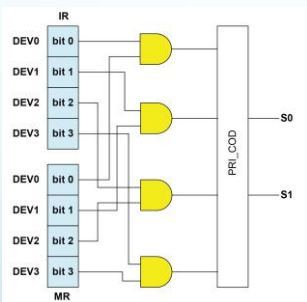


# Κύκλωμα ελέγχου διακοπών με μάσκα (1)

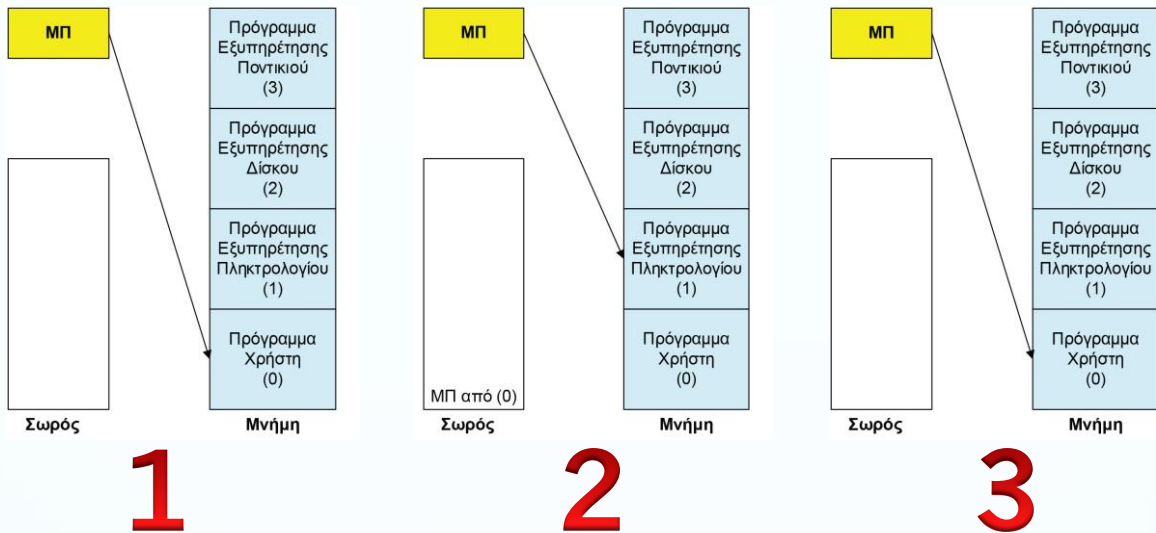


## Κύκλωμα ελέγχου διακοπών με μάσκα (2)

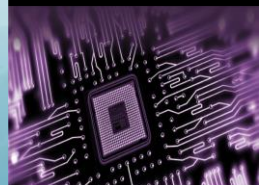
- ❑ Κωδικοποιητής προτεραιότητας (PRI\_COD) = παράγει ως έξοδο μια πληροφορία των 2 bit που αντιστοιχεί στην είσοδο που ενεργοποιήθηκε
- ❑ Ενεργοποίηση εισόδου = κάποια από τις τέσσερις συσκευές (DEV0 έως DEV3) έκανε αίτημα διακοπής
- ❑ Περισσότερα αιτήματα διακοπών = μεταφορά στην έξοδο του κωδικοποιητή των 2 bit που αντιστοιχούν στην πρώτη είσοδό του ή στην είσοδο που βρίσκεται πιο κοντά στον πρώτο ακροδέκτη
- ❑ Μεταφορά σήματος διακοπής στον κωδικοποιητή = η αντίστοιχη πύλη AND να τροφοδοτηθεί με δύο ψηφία που να έχουν την τιμή 1
- ❑ Τα bit του καταχωρητή μάσκας (MR) προσδιορίζονται ξεχωριστά και καθορίζουν αν επιτρέπονται διακοπές από κάποια συσκευή



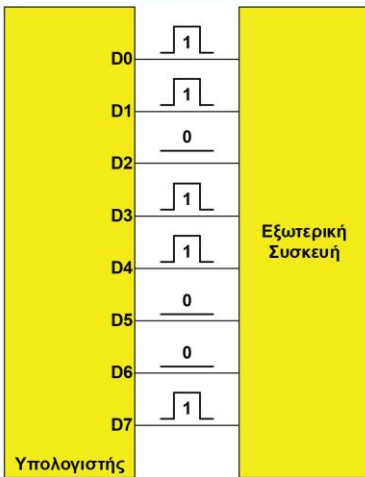
# Προγράμματα εξυπηρέτησης διακοπών



- ❑ Το κυρίως πρόγραμμα διακόπτεται στιγμιαία για να εξυπηρετηθεί μια εξωτερική συσκευή
- ❑ Ο σωρός «φιλοξενεί» τις απαραίτητες διευθύνσεις ώστε να εξασφαλίζεται η επιστροφή από κάθε υποπρόγραμμα που έχει ενεργοποιηθεί



# Ανταλλαγή δεδομένων

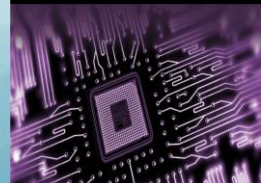
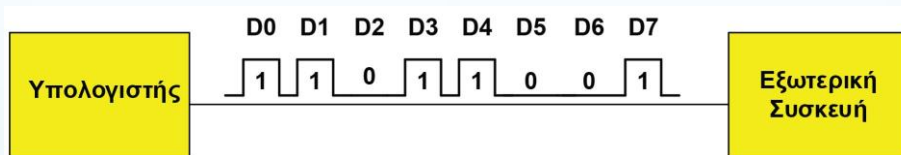


## Παράλληλη επικοινωνία

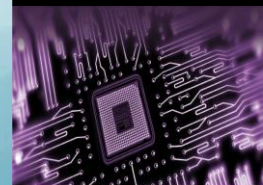
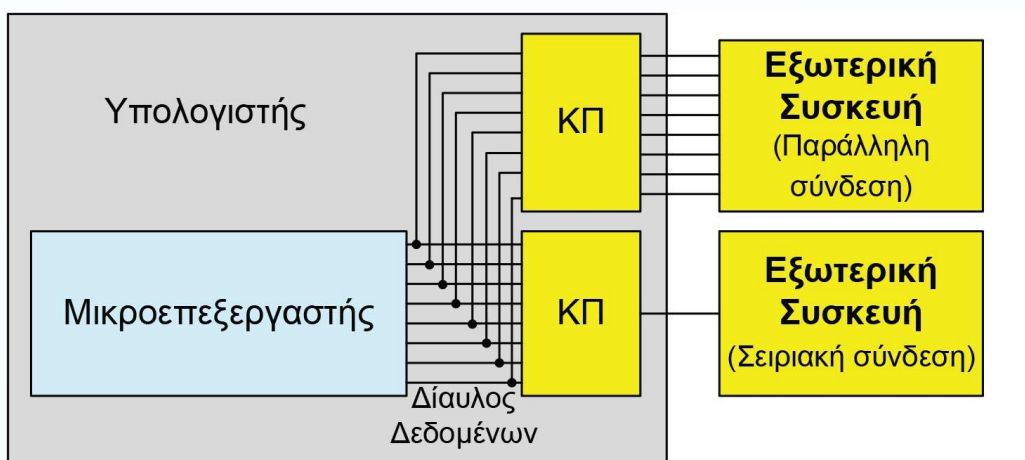
- Ένα κανάλι ανά bit
- Υψηλό κόστος μεταφοράς

## Σειριακή επικοινωνία

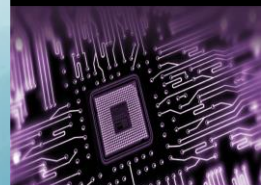
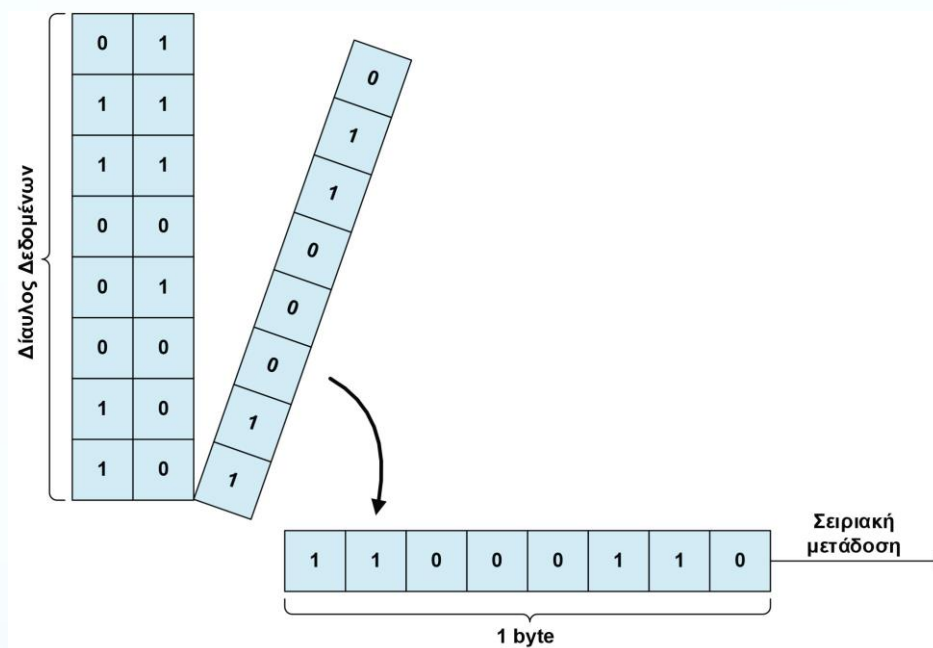
- Ένα κανάλι για όλα τα δεδομένα
- Σύγχρονη ή ασύγχρονη
- Μικρό κόστος μεταφοράς



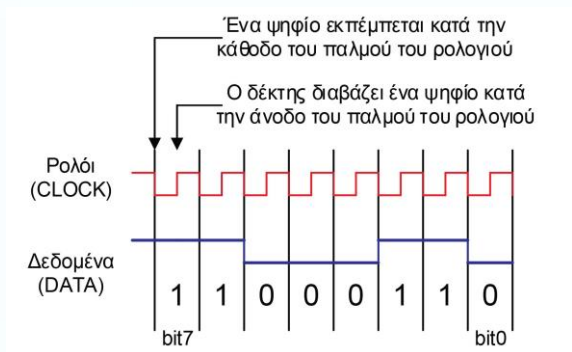
# Κυκλώματα προσαρμογής για επικοινωνία



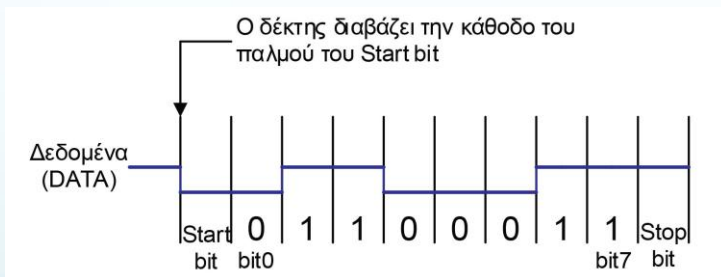
# Σειριακή επικοινωνία (1)



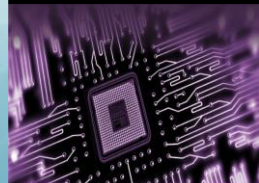
## Σειριακή επικοινωνία (2)



**Σύγχρονη επικοινωνία**



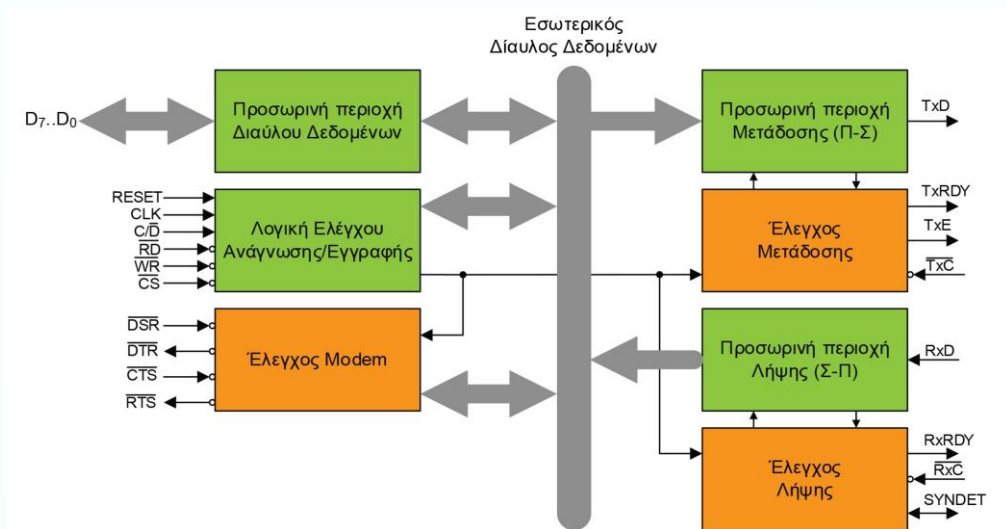
**Ασύγχρονη επικοινωνία**



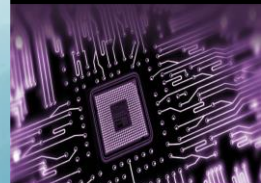


# Το ολοκληρωμένο κύκλωμα UART/USART

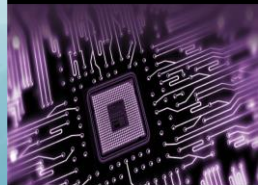
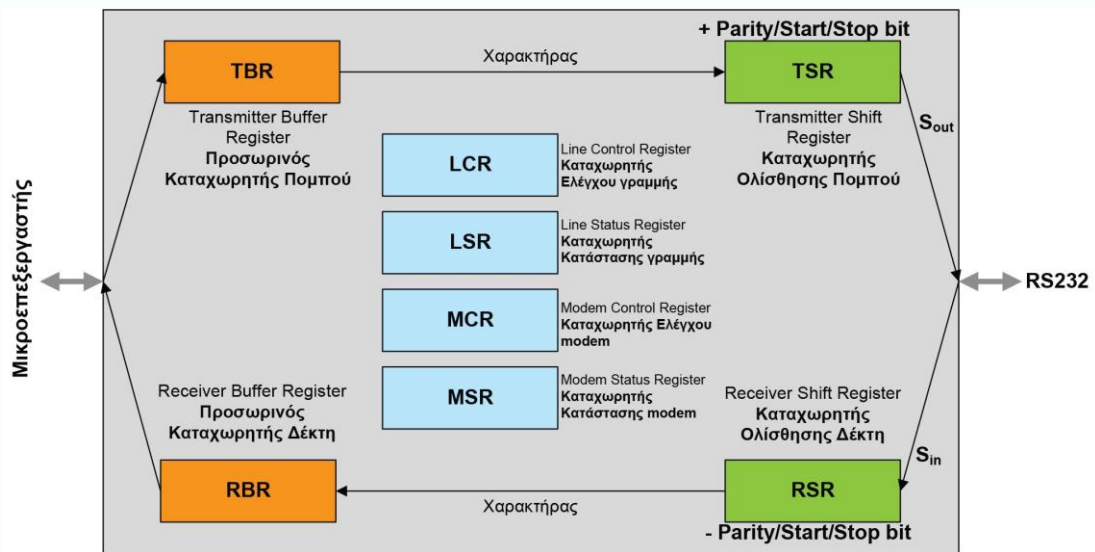
## Το ολοκληρωμένο κύκλωμα 8251



- ❑ UART/USART (Universal Asynchronous Receiver Transmitter/Universal Synchronous Asynchronous Receiver Transmitter)
- ❑ Μετατροπή παράλληλων δεδομένων σε σειριακά και το αντίθετο



# Καταχωρητές UART



# Κεφάλαιο 9

## Τα κύρια σημεία σε τίτλους

- ❑ Κυκλώματα προσαρμογής
- ❑ Τεχνικές επικοινωνίας
- ❑ Προγράμματα εξυπηρέτησης διακοπών
- ❑ Σειριακή και παράλληλη επικοινωνία
- ❑ Έλεγχος σειριακής επικοινωνίας με ολοκληρωμένα κυκλώματα
- ❑ Σύγχρονη και ασύγχρονη επικοινωνία
- ❑ Ολοκληρωμένο κύκλωμα UART/USART

