



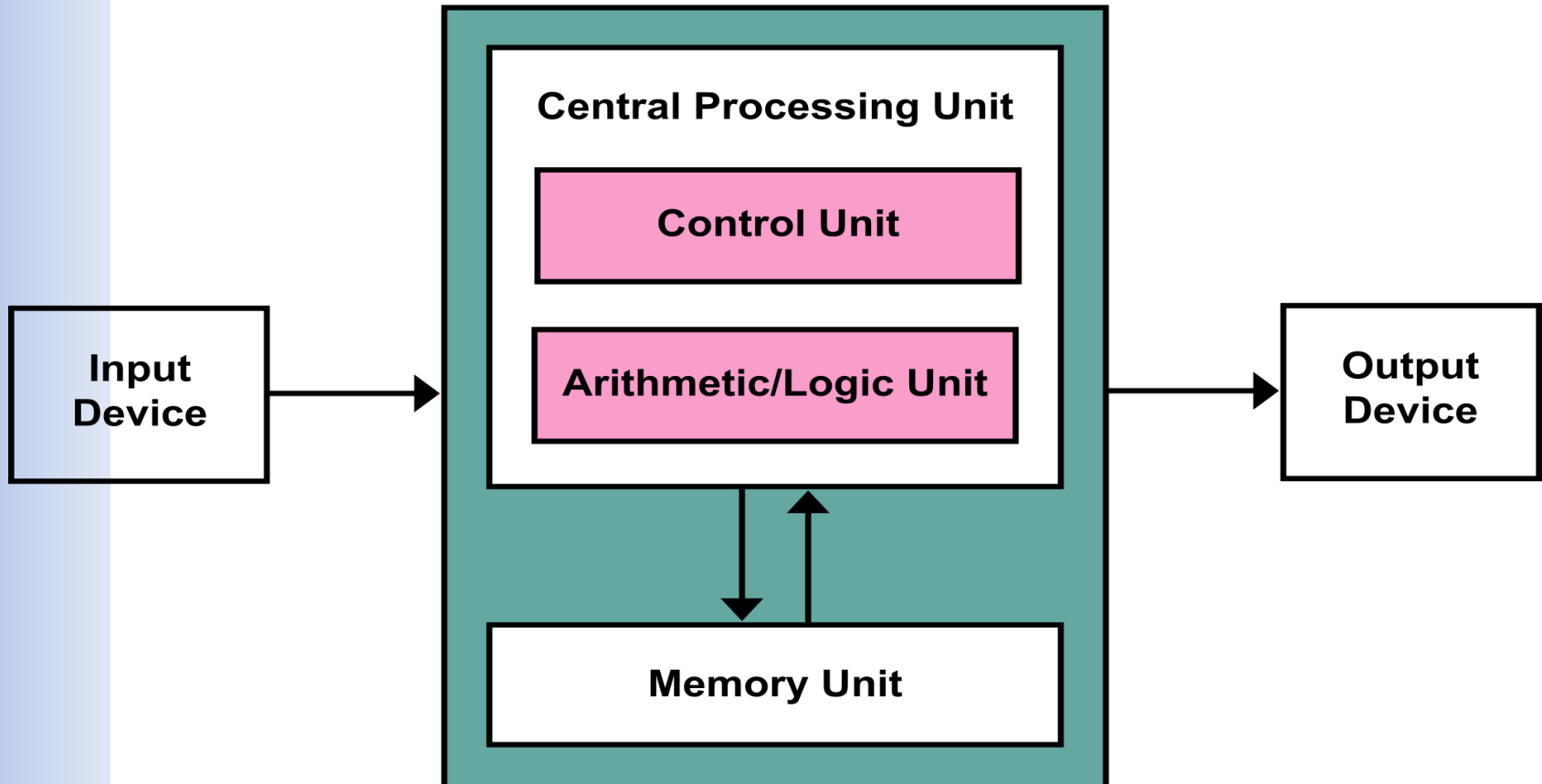
ΤΕΙ Ιονίων Νήσων  
Τμήμα Τεχνολογίας Τροφίμων  
Πληροφορική  
Σημειώσεις  
Τεύχος 3

Μάκης Σταματελάτος  
makiss@teiion.gr

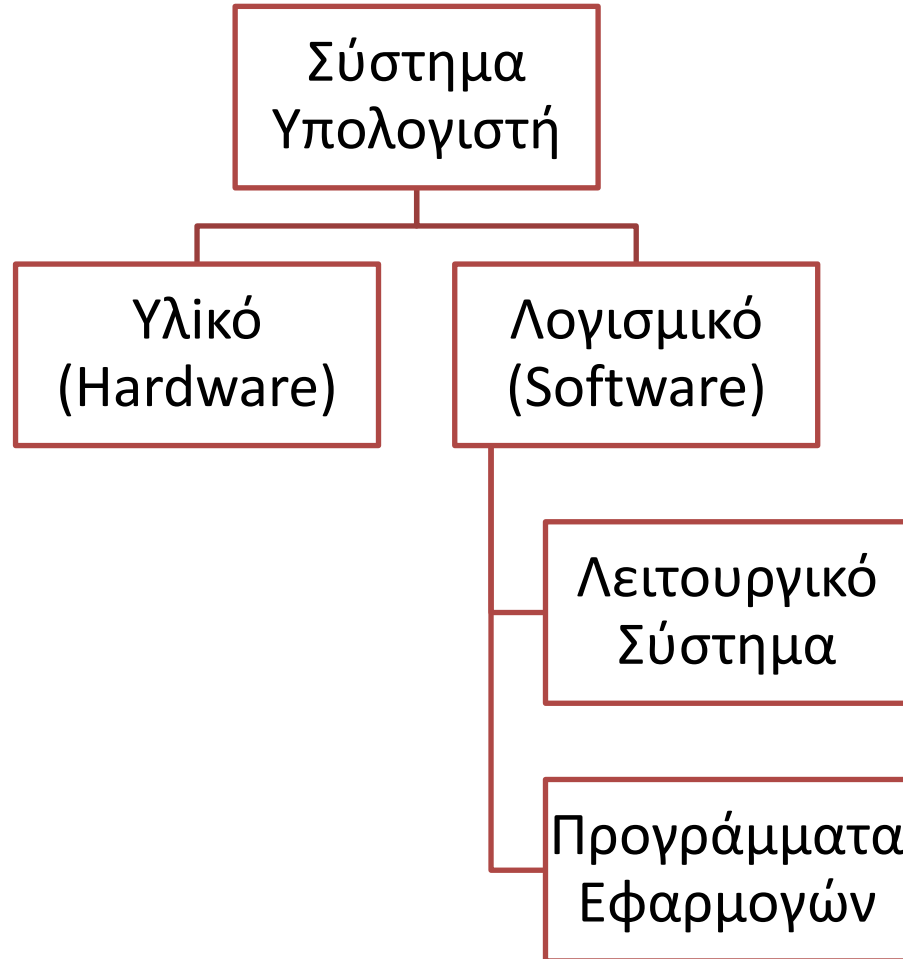


# Αρχιτεκτονική Η/Υ (1)

Κλασική προσέγγιση (von Neumann)



# Σύνθεση - Επίπεδα



# Κεντρική Μονάδα Επεξεργασίας

Η Κεντρική Μονάδα Επεξεργασίας που είναι γνωστή ως ο **Επεξεργαστής** του Υπολογιστή είναι η «καρδιά» του υπολογιστή και είναι υπεύθυνη για:

- τη **μεταφορά των εντολών των προγραμμάτων** που βρίσκονται στη μνήμη
- την **προετοιμασία και τελικά την εκτέλεση των εντολών των προγραμμάτων**

Η ΚΜΕ αποτελείται από δύο βασικά τμήματα:

- μονάδα ελέγχου, ΜΕ (Control Unit, CU)
- αριθμητική και λογική μονάδα, ΑΛΜ (Arithmetic and Logic Unit, ALU)



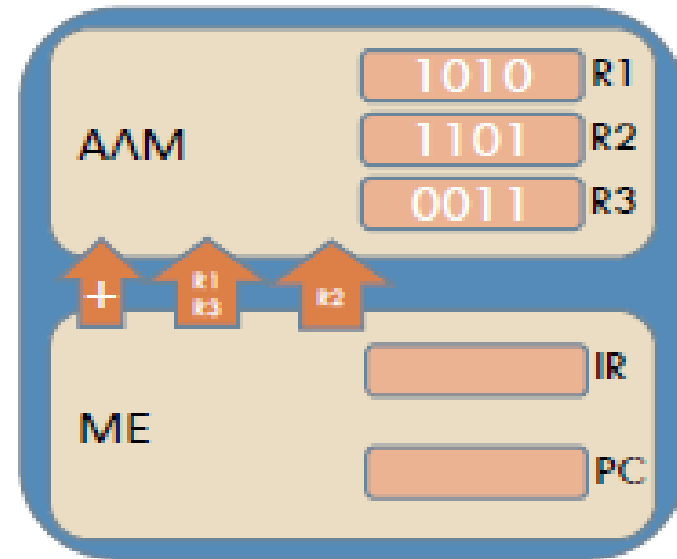
Καταχωρητές (registers) : θέσεις μνήμης για αποθήκευση δεδομένων



ΤΕΙ ΙΟΝΙΩΝ  
ΝΗΣΩΝ

# Η Αριθμητική και Λογική Μονάδα

- Η ΑΛΜ εκτελεί κάθε αριθμητική πράξη (+, -, /, \*) ή λογική (AND, OR, NOT, ...) που απαιτείται
- Η ΜΕ την καθοδηγεί στις εξής λειτουργίες:
  - ποια πράξη να εκτελέσει
  - από πού να πάρει δεδομένα
  - που να τοποθετήσει το αποτέλεσμα



# Αριθμητική και Λογική Μονάδα (ΑΛΜ – ALU)

Τα κυκλώματα, τα οποία συνήθως περιέχονται σε μια ALU:

- Ένας αθροιστής (adder) για την εκτέλεση των βασικών αριθμητικών πράξεων
- Ένας συγκριτής (comparator), για την σύγκριση των αριθμών που τοποθετούνται στους καταχωρητές
- Ένας ολισθητής (shifter), για την ολίσθηση ή την περιστροφή ενός αριθμού
- Μία μονάδα λογικής επεξεργασίας, για την εκτέλεση λογικών πράξεων πάνω στους αριθμούς, που τοποθετούνται στους καταχωρητές
- Κάποιοι καταχωρητές για την αποθήκευση των αριθμών και τα αποτελέσματα των πράξεων

# Μονάδα Ελέγχου (ΜΕ)

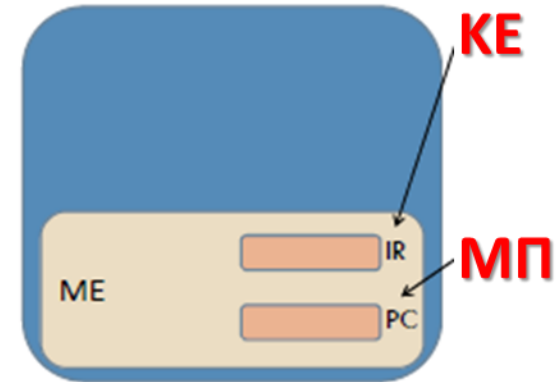
Η **Μονάδα Ελέγχου** του επεξεργαστή συντονίζει τη λειτουργία του κάνοντας τις εξής εργασίες:

- Μοιράζει το χρόνο του επεξεργαστή στα προγράμματα που εκτελούνται
- Ξεκινά και διακόπτει την εκτέλεση ενός προγράμματος
- Μεταφέρει τις εντολές των προγραμμάτων από τη μνήμη και τις αποκωδικοποιεί
- Συντονίζει τις διαδικασίες καταχώρισης και ανάκλησης δεδομένων στη μνήμη
- Ελέγχει για εξωτερικά γεγονότα (διακοπές) που μπορούν να τροποποιήσουν τη ροή εκτέλεσης των εργασιών

# Μονάδα Ελέγχου (ΜΕ)

Η ΜΕ είναι το οργανωτικό κέντρο του Η/Υ, που περιέχει δύο καταχωρητές:

- Καταχωρητή Εντολής, ΚΕ (Instruction Register, IR), που περιέχει την εντολή που εκτελείται αυτή τη στιγμή
- Μετρητή Προγράμματος, ΜΠ (program counter, PC), που περιέχει τη διεύθυνση (στη μνήμη) της εντολής που θα εκτελεστεί αμέσως μετά





# Η Μνήμη

## Μνήμη

### Κύρια Μνήμη

Η Κύρια Μνήμη περιέχει τις εντολές των προγραμμάτων που εκτελούνται κάθε στιγμή από τον υπολογιστή καθώς και τα δεδομένα που χρησιμοποιούν αυτά

**RAM**  
**ROM**  
**Cache**  
**Virtual Memory**

### Βοηθητική Μνήμη

Η Βοηθητική Μνήμη αποθηκεύει μόνιμα προγράμματα και δεδομένα καθώς αυτά χάνονται από την κύρια μνήμη όταν σταματήσει η τροφοδοσία με ρεύμα

**Σκληρός Δίσκος**  
**Οπτικός Δίσκος**

# Η Κύρια Μνήμη - Οργάνωση

Η Κύρια μνήμη του Ηλεκτρονικού Υπολογιστή είναι οργανωμένη σαν ένας πίνακας αποθήκευσης όπου κάθε θέση έχει μια διεύθυνση (μέσω τις οποίες είναι δυνατή η προσπέλασή της) και ένα περιεχόμενο που αποθηκεύει.

Τόσο η διεύθυνση όσο και το περιεχόμενο της μνήμης είναι δυαδικοί αριθμοί (ακόμη και οι εντολές των προγραμμάτων είναι δυαδικοί αριθμοί και η ΚΜΕ τους αποκωδικοποιεί προκειμένου να εντοπίσει σε ποια εντολή αντιστοιχεί ο εκάστοτε αριθμός.

Διεύθυνση 1	Περιεχόμενο 1 (εντολή προγράμματος ή δεδομένα)
Διεύθυνση 2	Περιεχόμενο 2 (εντολή προγράμματος ή δεδομένα)
...	...
Διεύθυνση N	Περιεχόμενο N (εντολή προγράμματος ή δεδομένα)

# Ανάγνωση -Εγγραφή

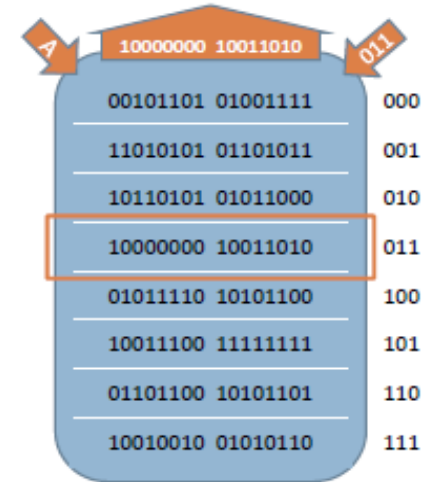
## ΑΝΑΓΝΩΣΗ

Η ΚΜΕ αποστέλλει:

- τον κωδικό της πράξης της ανάγνωσης
- τη διεύθυνση της λέξης που θέλει να διαβάσει

Η μνήμη επιστρέφει

- τα bit της λέξης

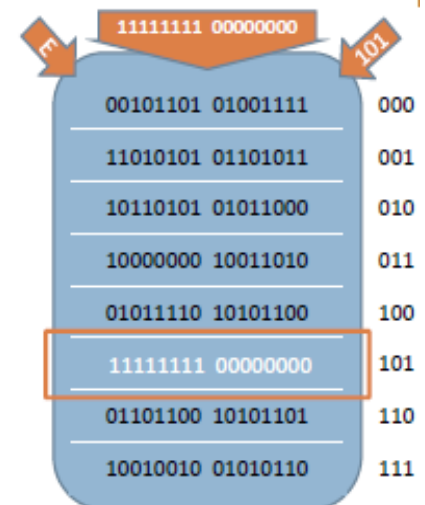


## ΕΓΓΡΑΦΗ

Η ΚΜΕ αποστέλλει:

- τον κωδικό της πράξης της εγγραφής
- τη διεύθυνση της λέξης που θέλει να γράψει
- τα bit της λέξης

Η μνήμη εκτελεί την αλλαγή



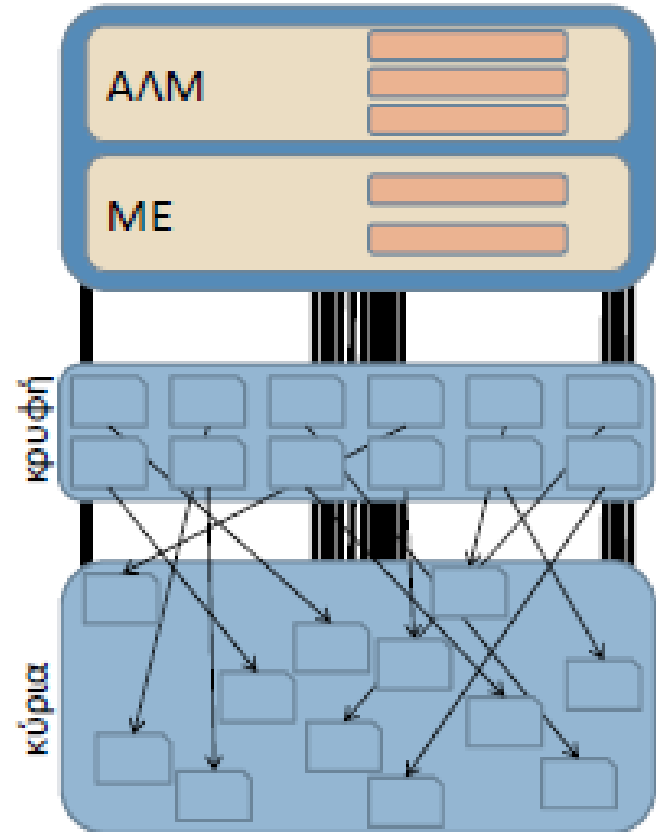
# Κρυφή Μνήμη (Cache Memory)

- Η λανθάνουσα ή κρυφή μνήμη (cache memory) βοηθά στην επιτάχυνση της λειτουργίας ενός επεξεργαστή κρατώντας το περιεχόμενο των πιο πρόσφατα χρησιμοποιημένων θέσεων μνήμης
- οι πιο πολλοί προσωπικοί υπολογιστές έχουν δύο διαφορετικούς τύπους λανθάνουσας μνήμης: «επιπέδου 1» (Level 1 ή L1 cache) & «επιπέδου 2» (Level 2 ή L2 cache)



# Κρυφή Μνήμη (Cache Memory)

- **L1:** βρίσκεται στο εσωτερικό του επεξεργαστή και κρατά διευθύνσεις μνήμης που αντιστοιχούν σε δεδομένα αλλά και σε εντολές μηχανής (~ 2MB)
- **L2:** είναι μεγαλύτερου μεγέθους, πιο αργή και βρίσκεται πάνω στη μητρική κάρτα του υπολογιστή (~ 4MB)

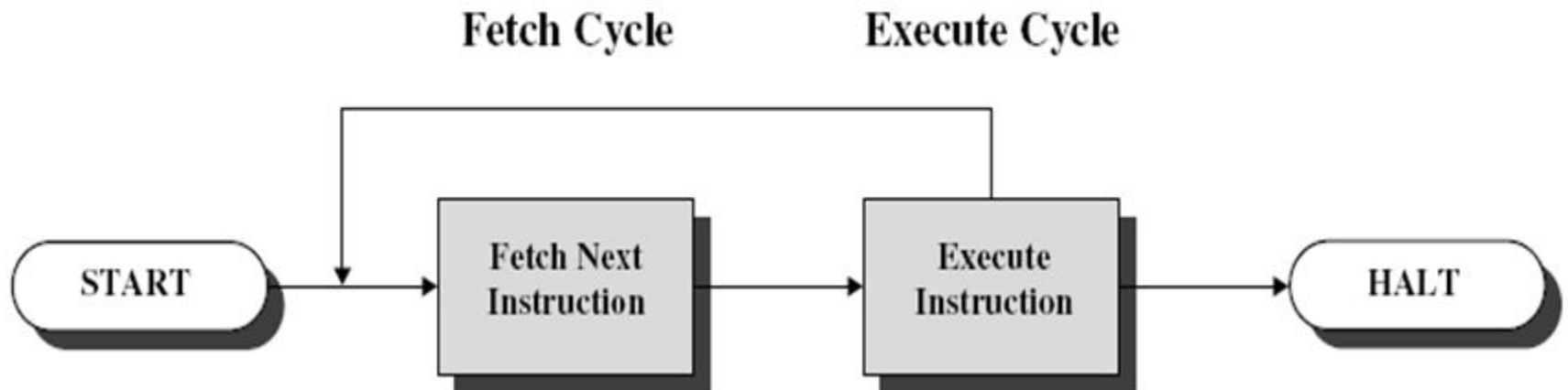


# Εικονικής Μνήμης (Virtual Memory)

- Αύξηση αναγκών σε μνήμη
- κάνει το χρήστη να βλέπει την κύρια μνήμη και ένα μέρος της μνήμης στο δίσκο σαν μία ενιαία, πολύ μεγάλη άμεσα προσπελάσιμη από την ΚΜΕ μνήμη
- η ιδεατή μνήμη γίνεται αντιληπτή από το λογισμικό του συστήματος, αλλά όχι και από το λογισμικό των εφαρμογών

# Κύκλος Μηχανής

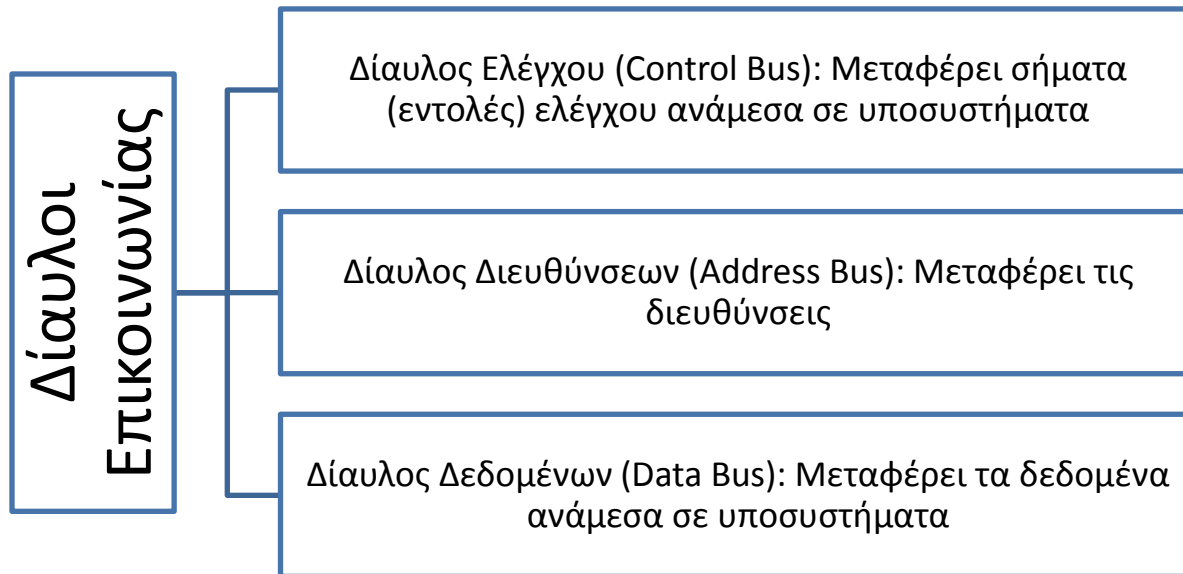
- Κάθε εντολή εκτελείται/αποτελείται σε δύο φάσεις (μέσα στον επεξεργαστή) :
  - Προσκόμιση εντολής (Fetch)
  - Εκτέλεση εντολής (Execute)
- Η διεργασία που απαιτείται για την εκτέλεση μιας εντολής ενός προγράμματος ονομάζεται κύκλος μηχανής (εντολής)



# Δίαυλοι Επικοινωνίας

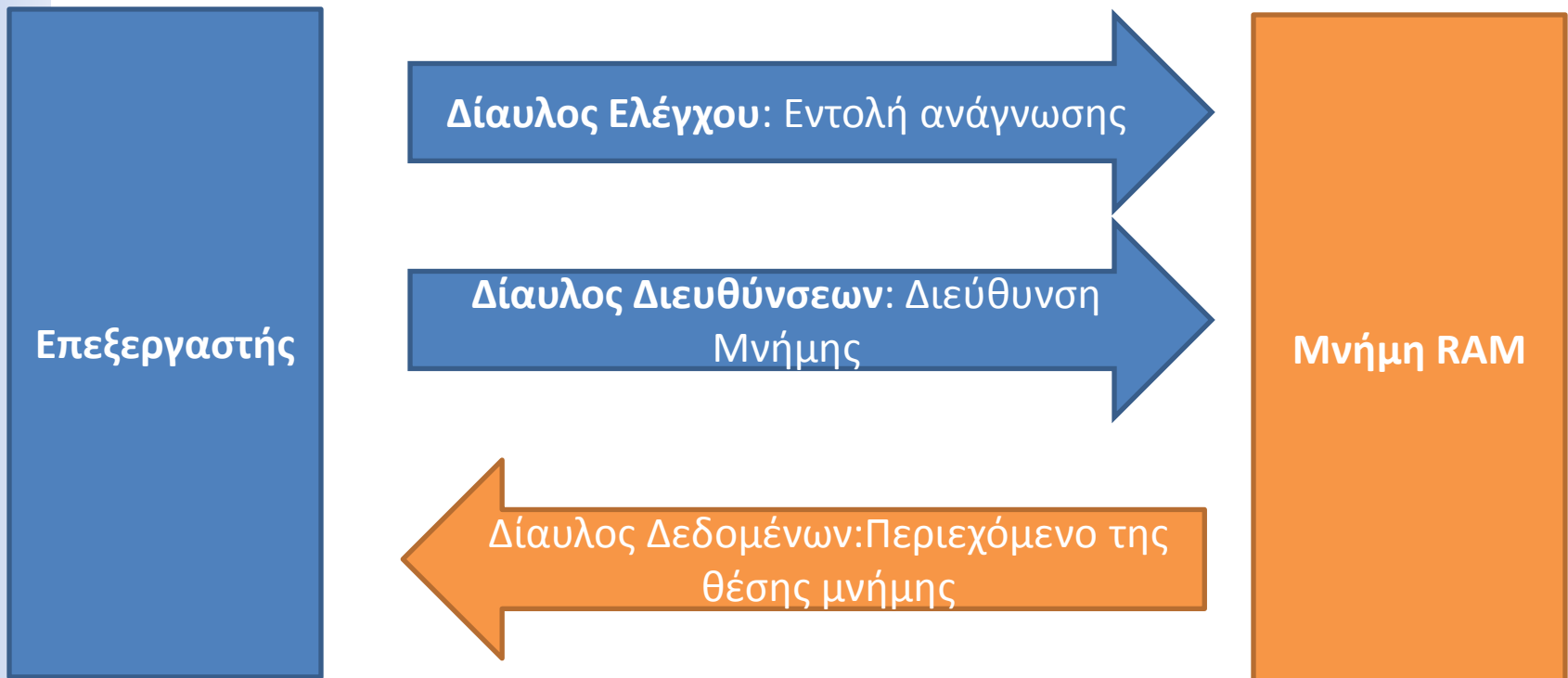
Οι δίαυλοι επικοινωνίας είναι τα «κανάλια» μέσω των οποίων τα υποσυστήματα του υπολογιστή όπως ο επεξεργαστής, η κύρια μνήμη, οι μονάδες εισόδου και εξόδου, οι κάρτες επέκτασης επικοινωνούν μεταξύ τους.

Οι δίαυλοι επικοινωνίας είναι «χαραγμένοι» στη μητρική πλακέτα καθώς σε αυτή τοποθετούνται ή συνδέονται όλα τα υποσυστήματα του υπολογιστή.





# Δίαυλοι Επικοινωνίας



# Μηχανές - Εντολές

- Ένα πρόβλημα μπορεί να λυθεί από μία μηχανή → είναι σε
- θέση να τίθεται σε διάφορες «καταστάσεις» και να εκτελεί
- συγκεκριμένες εργασίες για κάθε μία από αυτές
- Εργασία (μεταφορά περιεχομένου, πρόσθεση, κ.λπ.) →
- ενσωματωμένη στο σύστημα της μηχανής
- Εντολές → συστατικά μέρη μιας γλώσσας προγραμματισμού
- Προγραμματισμός → διατύπωση διαδικασίας επίλυσης
- προβλήματος
- Πρόγραμμα → σύνολο εντολών

# Κατηγορίες Εντολών

- Αριθμητικές Εντολές (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση, αύξηση ή μείωση του καταχωρητή κατά 1, αντιστροφή και αύξηση αληθές συμπλήρωμα) και ολίσθηση)
- Λογικές εντολές
- Εντολές μεταφοράς ελέγχου ή διακλάδωσης
- Εντολές μεταφοράς και καταχώρησης
- Εντολές εισόδου/εξόδου
- Ειδικές εντολές (stop, end, κ.τ.λ.)

# Γλώσσα μηχανής

- Οι εντολές της γλώσσας μηχανής διατυπώνονται κωδικοποιημένες στο δυαδικό σύστημα
- Αποτελούνται από:
  - Τον κωδικό εντολής: προσδιορίζει το είδος της ενέργειας
  - Τη διεύθυνση του παράγοντα: προσδιορίζει τη διεύθυνση της Κ.Μ.Ε. στην οποία ο υπολογιστής θα βρει τον παράγοντα ή τους παράγοντες
  - Το μήκος της εντολής (bits) ποικίλει ανάλογα με τον τύπο του υπολογιστή

# Προγραμματισμός

- Προγραμματισμός: Μέθοδος με την οποία ο χρήστης καθοδηγεί τον υπολογιστή στην επίλυση προβλημάτων
- Πρόγραμμα: λίστα από οδηγίες αναγκαίες για τον υπολογιστή προκειμένου να περατώσει μια εργασία

# Διαδικασία Προγραμματισμού

Αποτελείται από επτά βήματα:

1. Μαθηματική διατύπωση και ανάλυση του προβλήματος
2. Δημιουργία αλγορίθμου
3. Σχηματική παρουσία αλγορίθμου
4. Μεταφορά της διαδικασίας του αλγορίθμου σε γλώσσα προγραμματισμού
5. Εκτέλεση του προγράμματος στον υπολογιστή
6. Έλεγχος αποτελεσμάτων
7. Διόρθωση σφαλμάτων

# Αλγόριθμος

- Ο όρος αλγόριθμος χρησιμοποιείται για να δηλώσει μεθόδους που εφαρμόζονται για την επίλυση προβλημάτων
- Είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, για την επίλυση ενός προβλήματος



ΤΕΙ ΙΟΝΙΩΝ  
ΝΗΣΩΝ

# Αλγόριθμοι - Κριτήρια

- **Είσοδος (input).** Καμία, μία ή και περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια των συναρτήσεων παραγωγής τυχαίων αριθμών, ή με την βοήθεια άλλων απλών εντολών.
- **Έξοδος (output).** Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
- **Καθοριστικότητα (definiteness).** Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσης της. Για παράδειγμα, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης λαμβάνει τη μηδενική τιμή.
- **Περατότητα (finiteness).** Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).
- **Αποτελεσματικότητα (effectiveness).** Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μια εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.



# Αλγόριθμοι – μελέτη

- **Υλικό (hardware).** Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή.
- **Γλώσσες Προγραμματισμού (programming languages).** Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου.
- **Θεωρητική (theoretical).** Το ερώτημα που συχνά τίθεται είναι, αν πραγματικά υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος. Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική γιατί προσδιορίζει τα όρια της λύσης που θα βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα.
- **Αναλυτική (analytical).** Μελετώνται οι υπολογιστικοί πόροι (computer resources) που απαιτούνται από ένα αλγόριθμο όπως για παράδειγμα το μέγεθος της κύριας μνήμης και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες της CPU και για τις λειτουργίες εισόδου/εξόδου.

# Αλγόριθμοι – Περιγραφή – Αναπαράσταση

- **με ελεύθερο κείμενο (free text)**, που αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγορίθμων, δηλαδή την αποτελεσματικότητα.
- **με διαγραμματικές τεχνικές (diagramming techniques)**, που συνιστούν ένα γραφικό τρόπο παρουσίασης αλγορίθμου. Η πιο γνωστή από αυτές είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση γι' αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και πράξη.
- **με φυσική γλώσσα (natural language)** κατά βήματα. Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το τρίτο βασικό χαρακτηριστικό ενός αλγορίθμου, όπως προσδιορίστηκε προηγουμένως δηλαδή το κριτήριο του καθορισμού.
- **με κωδικοποίηση (coding)**, δηλαδή με ένα πρόγραμμα που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

# Γλώσσες προγραμματισμού

- Χαμηλού επιπέδου: συγγενικές με τη γλώσσα μηχανής. Συνδυασμοί δυαδικών ακολουθιών → «μνημονικοί» συμβολισμοί
- Υψηλού επιπέδου: ανεξάρτητες από τα διάφορα συστήματα υπολογιστών
- Ενδιάμεση χρήση συμβολομεταφραστή
  - Μεταγλωττιστές (compilers): δέχονται ένα ολόκληρο αρχικό πρόγραμμα και παράγουν ένα ισοδύναμο τελικό πρόγραμμα σε κώδικα μηχανής
  - Διερμηνευτές (interpreters): καλούνται κάθε φορά που επιθυμούμε να εκτελέσουμε ένα πρόγραμμα. Διαβάζει μία μία τις εντολές και εκτελεί για κάθε μία την ισοδύναμη ακολουθία εντολών μηχανής

# Γλώσσες Προγραμματισμού

- Διαδικαστικές γλώσσες (FORTRAN, COBOL, PASCAL, C...)
- Αντικειμενοστραφείς γλώσσες (Java, Visual C++, ...)
- Συναρτησιακές γλώσσες (Lisp, Hope)
- Μη διαδικαστικές γλώσσες (Prolog)

# Γλώσσες Προγραμματισμού

- Μεγάλη ποικιλία γλωσσών προγραμματισμού
- ΔΕΝ υπάρχει μία και μόνο γλώσσα κατάλληλη για όλες τις δουλειές (one-fits-all)

# Γλώσσες Προγραμματισμού

- Γενικής Χρήσης: C, Pascal, Fortran, C++, C#, Java, Python, Javascript, Ruby, Perl, PHP, SML, O'CamI, Haskell, Lisp, Erlang, Prolog, Assembly
- Domain specific: SQL, VHDL
- Scripting: Shell, AWK, sed
- Γλώσσες markup: SGML, HTML, XML, TeX

# Γλώσσες Προγραμματισμού

- Η C εξακολουθεί να είναι από τις πιο διαδεδομένες γλώσσες προγραμματισμού γενικής χρήσης, και φαίνεται σαν ένας κοινός παρονομαστής ανάμεσα στους περισσότερους προγραμματιστές· η καλή γνώση της κρίνεται απαραίτητη. Αν μη τι άλλο είναι χρήσιμη σε περίπτωση που χρειάζεται να μελετήσουμε τον κώδικα κάποιας εφαρμογής γραμμένη σε αυτή. Επιπλέον, η *σύνταξη* εξίσου διαδεδομένων γλωσσών όπως η C++, Java και C# είναι αρκετά όμοια.
- Από τις τελευταίες η Java κατέχει σημαντικό μερίδιο στην αγορά εργασίας (κυρίως σε ότι αφορά δικτυακές και enterprise εφαρμογές), ενώ χάνει σιγά σιγά έδαφος από την C#. Η C++ είναι ακόμα πολύ δυνατή σε ορισμένους τομείς, ειδικά στο Game Development.
- Πολύ ενδιαφέρουσες είναι και οι γλώσσες συναρτησιακού προγραμματισμού. Από αυτές, αρκετά μέλη της κοινότητας φαίνεται να ξεχωρίζουν την Haskell και την O'CamI. Οι γλώσσες αυτές τα τελευταία χρόνια έχουν αρχίσει να ξεφεύγουν από τον ακαδημαϊκό τομέα και να γίνονται περισσότερο mainstream.
- Scripting γλώσσας υψηλού επιπέδου, σαν την Python, την Ruby ή την Perl. Είναι εξαιρετικά χρήσιμες για prototyping, εφαρμογές web αλλά και rapid application development. Επιπλέον μπορούν να αντικαταστήσουν το shell scripting σε ορισμένα administrative tasks, όπου απαιτείται μια περισσότερο δομημένη προσέγγιση.

# Ανοιχτός Κώδικας

- Πρόγραμμα υπολογιστή στον οποίο ο πηγαίος κώδικας είναι διαθέσιμος στο ευρύ κοινό για χρήση ή / και την τροποποίηση από τον αρχικό σχεδιασμό του.
- Ο Ανοιχτός πηγαίος κώδικας δημιουργείται τυπικά ως μια συλλογική προσπάθεια στην οποία προγραμματιστές βελτιώνουν τον κώδικα και μοιράζονται τις αλλαγές εντός της κοινότητας.
- Ο Ανοιχτός κώδικας βλάστησε στην τεχνολογική κοινότητα ως μια απάντηση στο ιδιόκτητο λογισμικό που ανήκει σε εταιρείες.



# Ελεύθερο Λογισμικό

- Διαφορά μεταξύ Ελεύθερου Λογισμικού και Λογισμικού Ανοικτού Κώδικα.
- Οι δύο όροι αναφέρονται σε παρόμοιες αντιλήψεις: και οι δύο προσπαθούν να προωθήσουν τη δημιουργία καλύτερου λογισμικού και γενικά να προωθήσουν την “κοινή γνώση”.
- Η κύρια διαφορά τους
  - το Ελεύθερο Λογισμικό δίνει έμφαση στην ελευθερία (για μάθηση, προσαρμογή, βελτίωση),
  - το Λογισμικό Ανοικτού Κώδικα δίνει έμφαση στη δημιουργία καλύτερου λογισμικού μέσω της συνεργασίας των προγραμματιστών και των μηχανικών όλου του κόσμου.
- Συνεπώς, το Λογισμικό Ανοικτού Κώδικα είναι λιγότερο αυστηρό και πιο “φιλικό” προς τις εταιρίες που επιθυμούν να αξιοποιήσουν αλγορίθμους που υπάρχουν σε έργα τύπου Λογισμικού Ανοικτού Κώδικα.