

**Οργάνωση και Σχεδίαση Υπολογιστών**  
**Η Διασύνδεση Υλικού και Λογισμικού, 4<sup>η</sup> έκδοση**

# **Κεφάλαιο 1**

**Αφηρημένες έννοιες και  
τεχνολογία υπολογιστών**

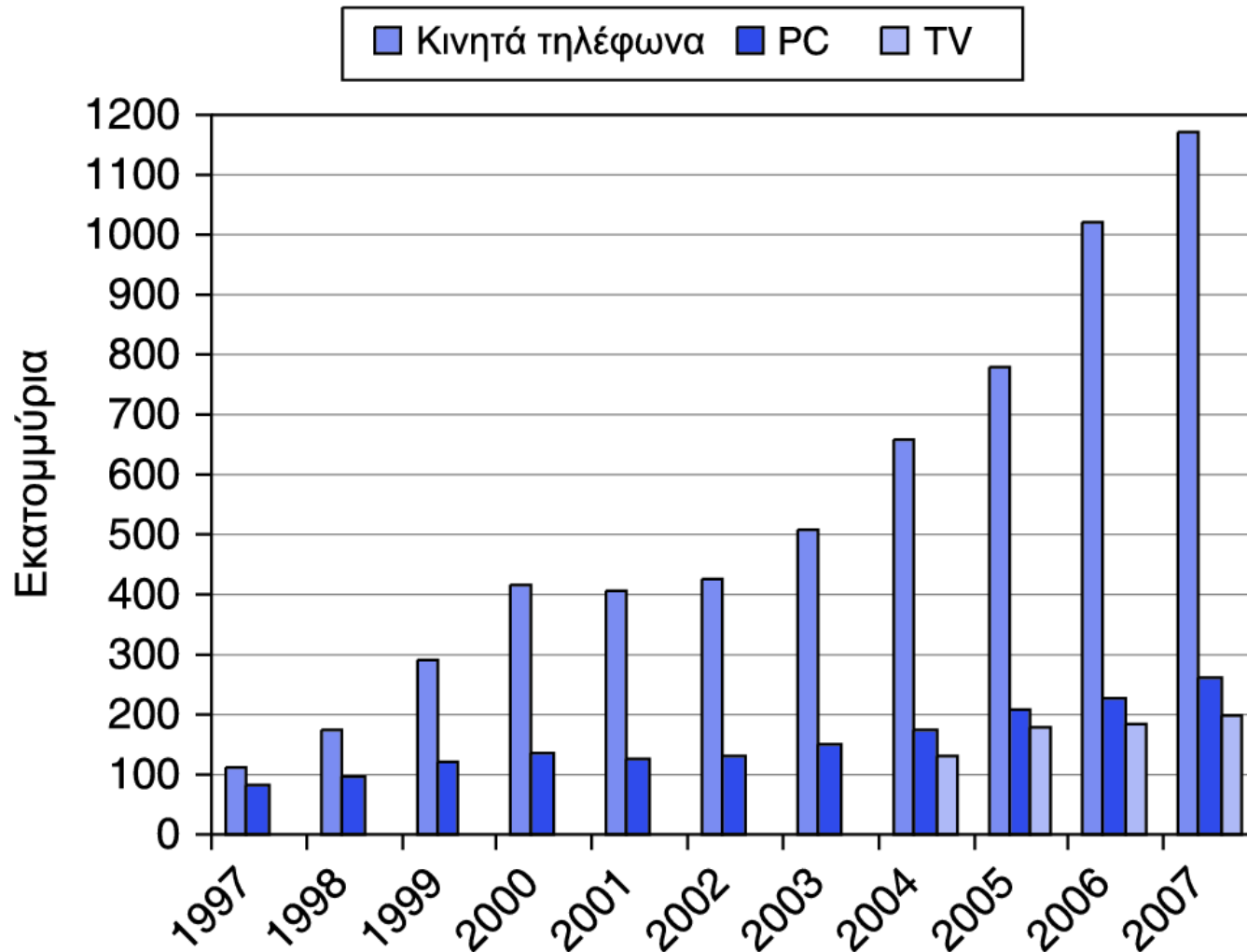
# Επανάσταση υπολογιστών

- Πρόοδος της τεχνολογίας υπολογιστών
  - Θεμελιώνεται από το νόμο του Moore
- Καινοτόμες εφαρμογές γίνονται εφικτές
  - Υπολογιστές σε αυτοκίνητα
  - Κινητά τηλέφωνα
  - Ερευνητικό πρόγραμμα ανθρώπινου γονιδιώματος (human genome)
  - Παγκόσμιος Ιστός (World Wide Web)
  - Μηχανές αναζήτησης
- Οι υπολογιστές βρίσκονται παντού

# Κατηγορίες υπολογιστών

- Επιτραπέζιοι (desktop) υπολογιστές
  - Γενικού σκοπού, ποικιλία λογισμικού
  - Συμβιβασμοί (tradeoffs) κόστους/απόδοσης
- Διακομιστές (servers)
  - Βασίζονται σε δίκτυο
  - Μεγάλη χωρητικότητα, απόδοση, αξιοπιστία
  - Εύρος: μικροί διακομιστές μέχρι μέγεθος κτηρίου
- Ενσωματωμένοι (embedded) υπολογιστές
  - «Κρυμμένοι» ως συστατικά άλλων συστημάτων
  - Αυστηροί περιορισμοί ισχύος/απόδοσης/κόστους

# Η αγορά των επεξεργαστών



# Τι θα μάθετε

- Πώς μεταφράζονται τα προγράμματα στη γλώσσα της μηχανής
  - Και πώς τα εκτελεί το υλικό
- Τη διασύνδεση υλικού και λογισμικού
- Τι καθορίζει την απόδοση των προγραμμάτων
  - Και πώς αυτή μπορεί να βελτιωθεί
- Πώς βελτιώνουν την απόδοση οι σχεδιαστές του υλικού
- Τι είναι η παράλληλη επεξεργασία

# Κατανόηση της απόδοσης

- Αλγόριθμος
  - Καθορίζει τον αριθμό λειτουργιών που εκτελούνται
- Γλώσσα προγραμματισμού, μεταγλωττιστής, αρχιτεκτονική
  - Καθορίζουν τον αριθμό των εντολών μηχανής που εκτελούνται ανά λειτουργία
- Επεξεργαστής και σύστημα μνήμης
  - Καθορίζουν την ταχύτητα εκτέλεσης των εντολών
- Σύστημα Εισόδου/Εξόδου (περιλαμβάνει και το Λειτουργικό Σύστημα)
  - Καθορίζει την ταχύτητα εκτέλεσης των λειτουργιών εισόδου/εξόδου

# Κάτω από το πρόγραμμά σας



- Λογισμικό εφαρμογών
  - Γραμμένο σε γλώσσα υψηλού επιπέδου (high-level language)
- Λογισμικό συστημάτων
  - Μεταγλωττιστής (compiler): μεταφράζει κώδικα HLL σε κώδικα μηχανής
  - Λειτουργικό σύστημα: κώδικας υπηρεσιών
    - Χειρισμός εισόδου/εξόδου
    - Διαχείριση μνήμης και αποθήκευσης
    - Χρονοπρογραμματισμός εργασιών & κοινή χρήση πόρων

## Υλικό

- Επεξεργαστής, μνήμη, ελεγκτές εισόδου/εξόδου

# Επίπεδα κώδικα προγράμ/τος

- Γλώσσα υψηλού επιπέδου
  - Επίπεδο αφαίρεσης πιο κοντά στο πρόβλημα
  - Παρέχει παραγωγικότητα και φορητότητα
- Συμβολική γλώσσα (assembly language)
  - Αναπαράσταση εντολών μηχανής με κείμενο
- Αναπαράσταση υλικού
  - Δυαδικά ψηφία (bit)
  - Κωδικοποιημένες εντολές και δεδομένα

Πρόγραμμα  
γλώσσας  
υψηλού  
επιπέδου  
(σε γλώσσα C)

```
swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Μεταγλωττιστής

Πρόγραμμα  
συμβολικής  
γλώσσας  
(για επεξεργαστή MIPS)

```
swap:
    multi $2, $5, 4
    add    $2, $4, $2
    lw     $15, 0($2)
    lw     $16, 4($2)
    sw     $16, 0($2)
    sw     $15, 4($2)
    jr     $31
```

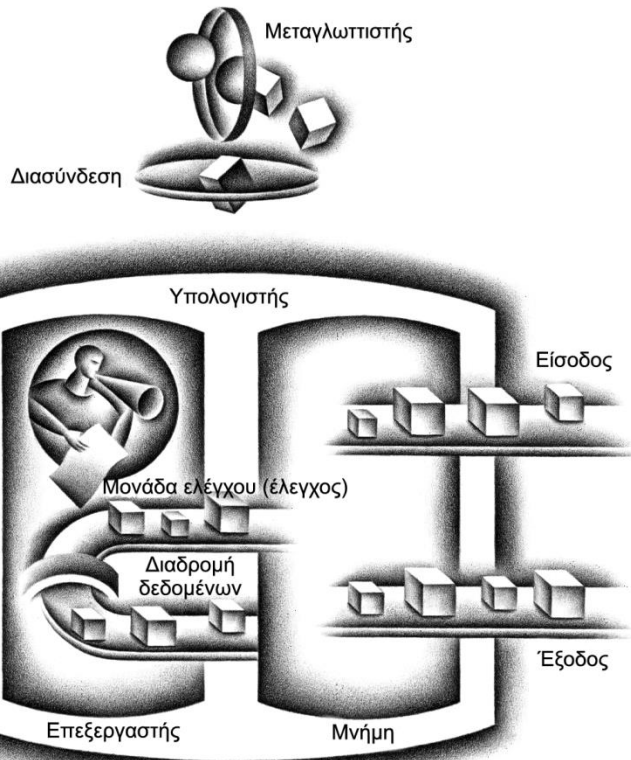
Συμβολομεταφραστής

Πρόγραμμα  
δυαδικής  
γλώσσας  
μηχανής  
(για επεξεργαστή  
MIPS)

```
000000001010001000000000100011000
00000000100000100001000000100000
10001101111000100000000000000000
100011100001001000000000000000100
10101110000100100000000000000000
101011011110001000000000000000100
00000011111000000000000000001000
```

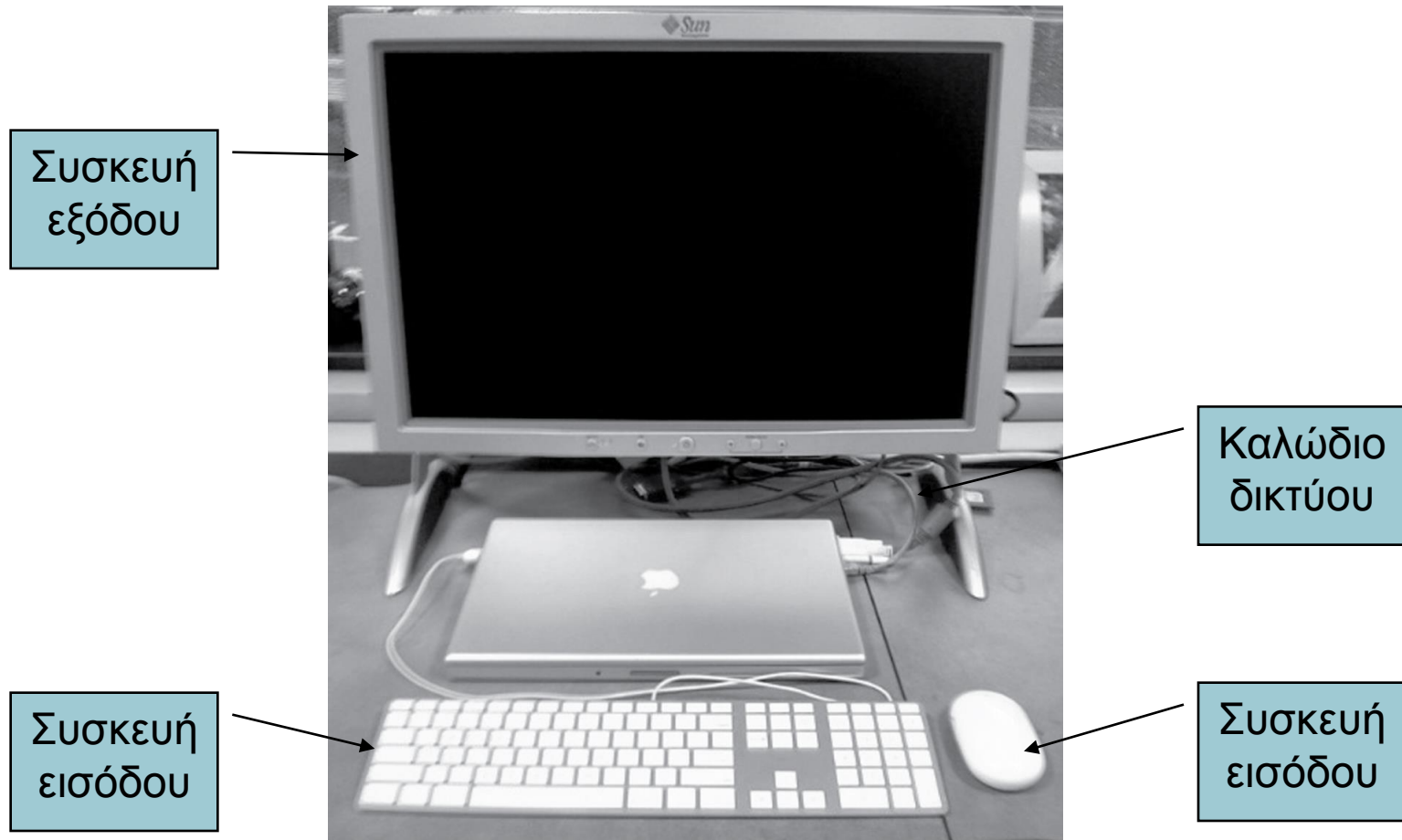
# Συστατικά ενός υπολογιστή

## ΓΕΝΙΚΗ εικόνα



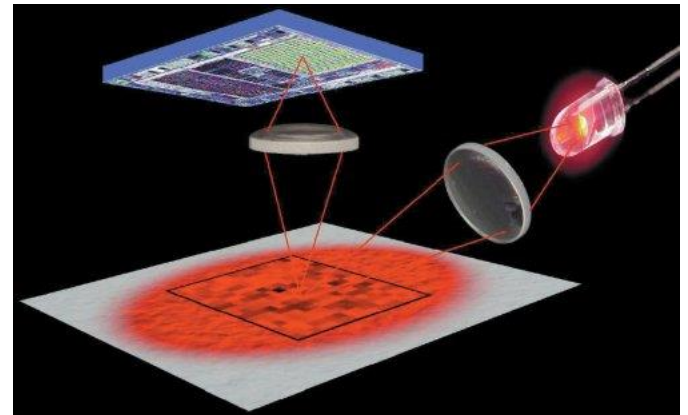
- Τα ίδια συστατικά για όλα τα είδη υπολογιστών
  - Επιτραπέζιοι, διακομιστές, ενσωματωμένοι
- Η είσοδος/έξοδος περιλαμβάνει
  - Συσκευές διασύνδεσης χρήστη (user-interface)
    - Οθόνη, πληκτρολόγιο, ποντίκι
  - Συσκευές αποθήκευσης
    - Σκληρός δίσκος, CD/DVD, flash
  - Προσαρμογείς δικτύου (network adapters)
    - Επικοινωνία με άλλους υπολογιστές

# Ανατομία ενός υπολογιστή



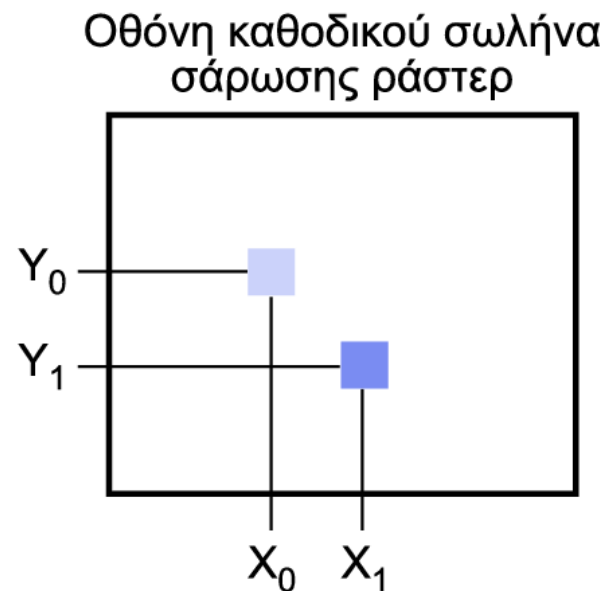
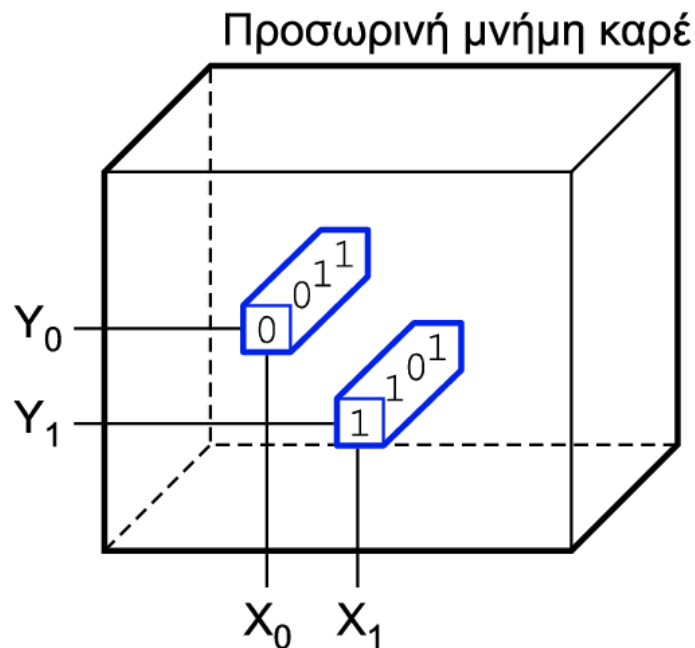
# Ανατομία ενός ποντικιού

- Οπτικό ποντίκι
  - Μια LED φωτίζει την επιφάνεια γραφείου
  - Μικρή κάμερα χαμηλής ανάλυσης
  - Βασικός οπτικός επεξεργαστής
    - Παρατηρεί μετακινήσεις στους άξονες x, y
  - Πλήκτρα & τροχός
- Αντικαθιστά το μηχανικό ποντίκι με την μπίλια

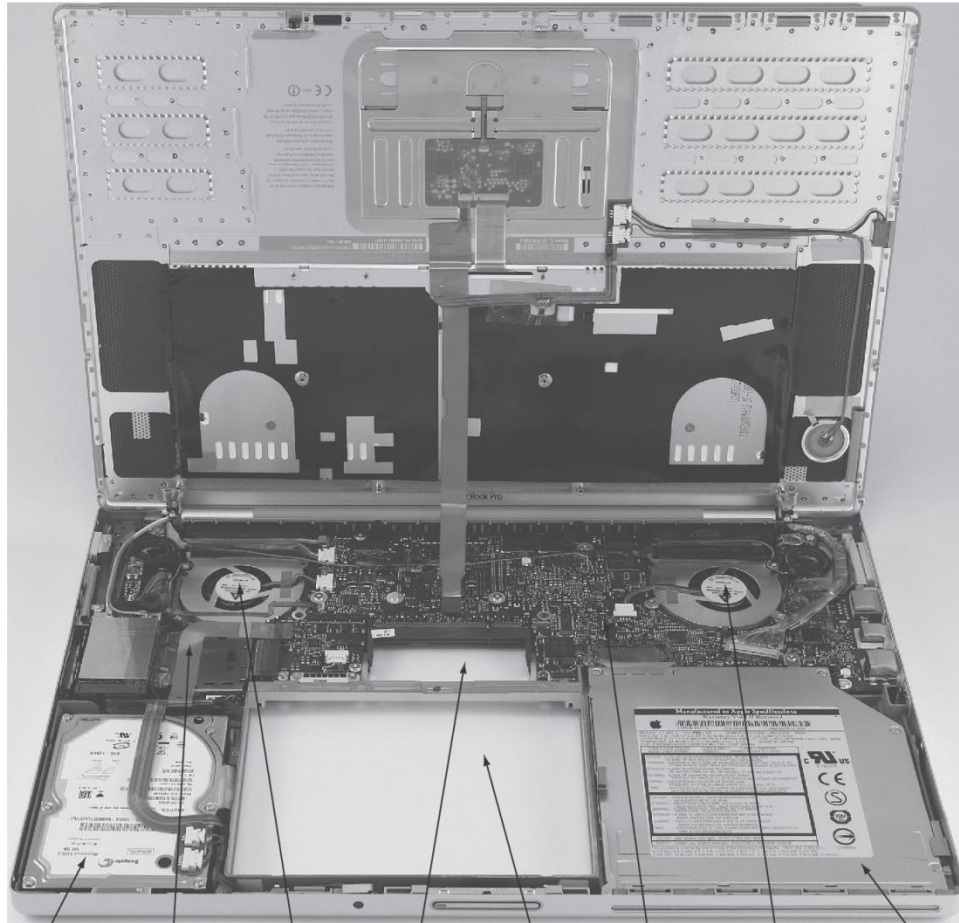


# Μέσα από τον καθρέφτη

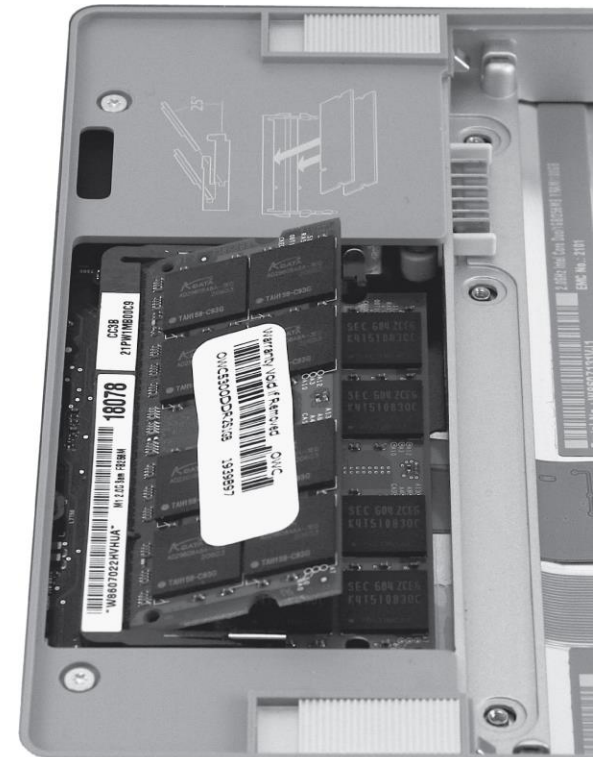
- Οθόνη LCD (liquid crystal display): πίξελ (pixels), ή εικονοστοιχεία
  - Αντικατοπτρίζει τα περιεχόμενα της προσωρινής μνήμης καρέ (frame buffer)



# Το άνοιγμα του κουτιού



Σκληρός δίσκος    Επεξεργαστής    Ανεμιστήρας με κάλυμμα    Θέση DIMM μνήμης    Θέση μπαταρίας    Μητρική κάρτα    Ανεμιστήρας με κάλυμμα    Μονάδα DVD

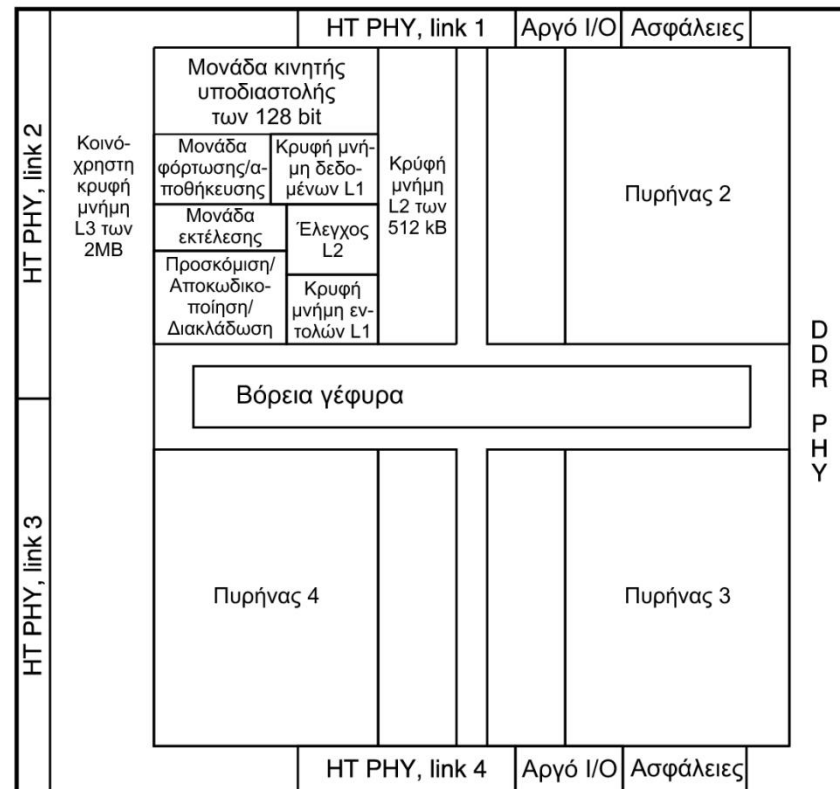
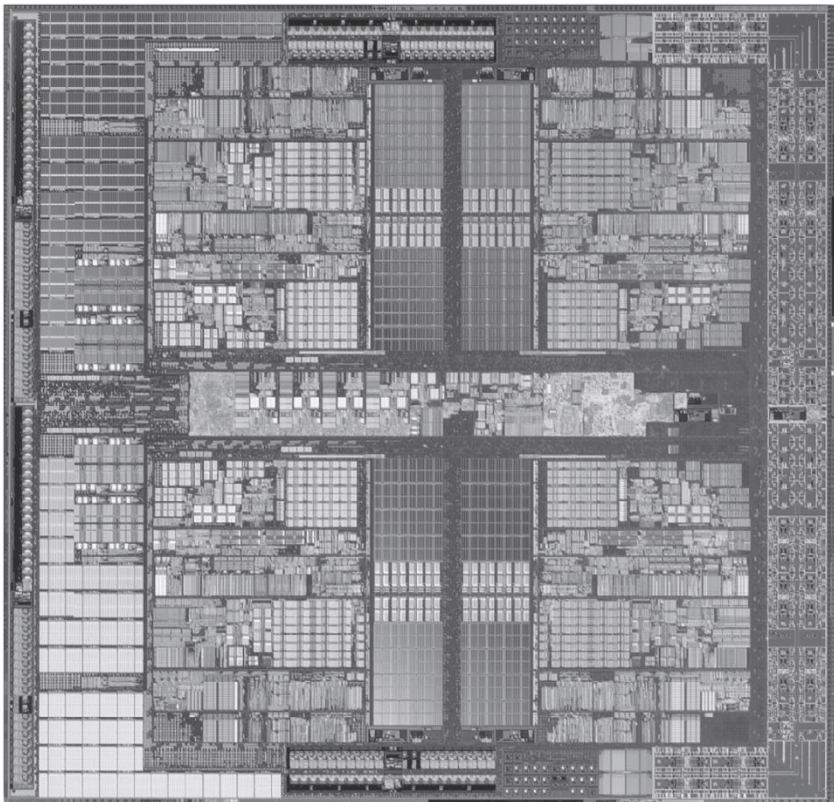


# Μέσα στον επεξεργαστή, CPU

- Διαδρομή δεδομένων (datapath): εκτελεί λειτουργίες στα δεδομένα
- Έλεγχος: ορίζει την ακολουθία λειτουργιών στη διαδρομή δεδομένων, τη μνήμη, ...
- Κρυφή μνήμη (cache memory)
  - Μικρή γρήγορη μνήμη SRAM για άμεση προσπέλαση δεδομένων

# Μέσα στον Επεξεργαστή

- AMD Barcelona: 4 πυρήνες (cores) επεξεργαστή



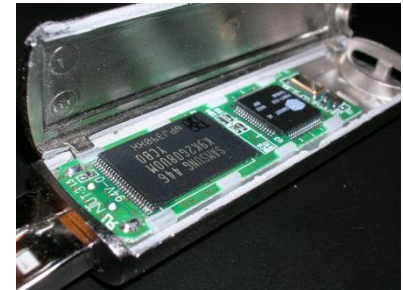
# Αφαιρέσεις (Abstractions)

## ΓΕΝΙΚΗ εικόνα

- Οι αφαιρέσεις μάς βοηθούν να χειριστούμε την πολυπλοκότητα
  - Κρύβουν λεπτομέρειες χαμηλού επιπέδου
- Αρχιτεκτονική συνόλου εντολών (Instruction set architecture – ISA)
  - Η διασύνδεση υλικού και λογισμικού
- Δυαδική διασύνδεση εφαρμογής (application binary interface)
  - Η ISA και επιπλέον η διασύνδεση του λογισμικού συστημάτων
- Υλοποίηση
  - Οι λεπτομέρειες κάτω από μια διασύνδεση

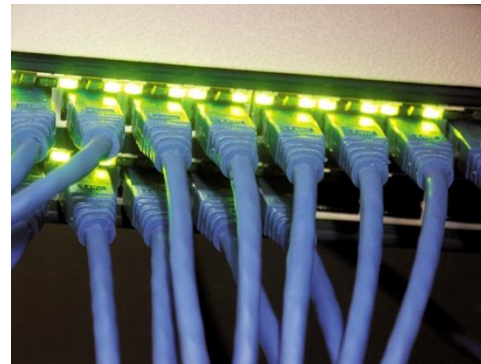
# Ασφαλές μέρος για δεδομένα

- Πτητική (volatile) κύρια μνήμη
  - Χάνει τις εντολές και δεδομένα όταν δεν έχει ρεύμα
- Μη πτητική (non-volatile) δευτερεύουσα μνήμη
  - Μαγνητικός δίσκος
  - Μνήμη φλας (flash)
  - Οπτικός δίσκος (CDROM, DVD)



# Δίκτυα

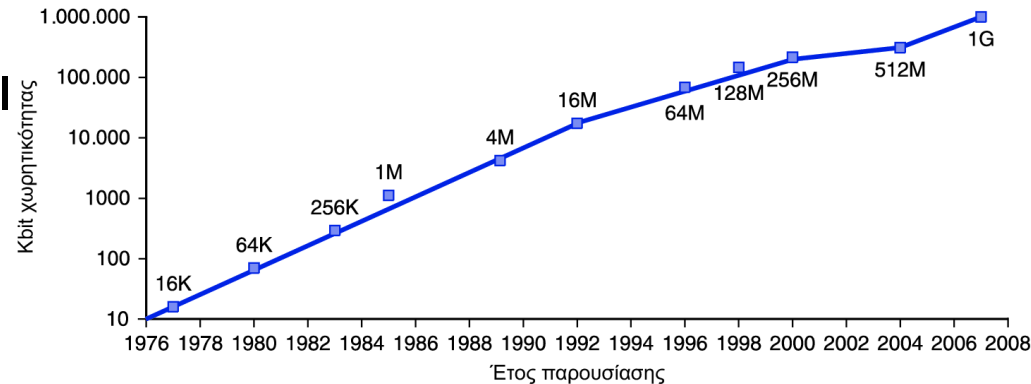
- Επικοινωνία και κοινή χρήση πόρων
- Τοπικό δίκτυο (local area network, LAN): Ethernet
  - Μέσα σε ένα κτήριο
- Δίκτυο ευρείας περιοχής (wide area network, WAN): το Internet
- Ασύρματο δίκτυο: WiFi, Bluetooth



# Τάσεις της τεχνολογίας

- Τεχνολογία ηλεκτρονικών συνεχίζει να εξελίσσεται

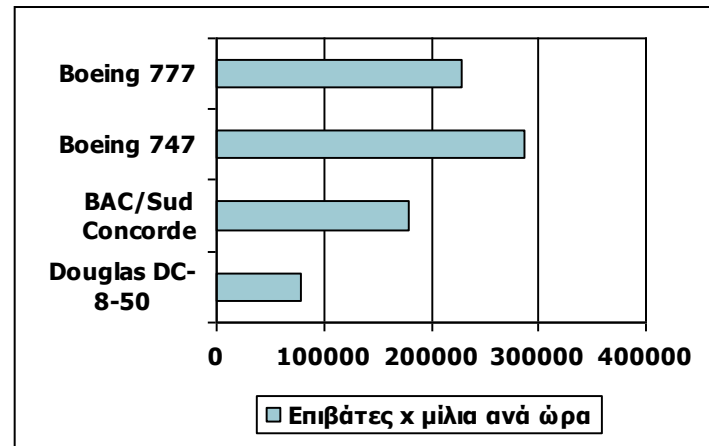
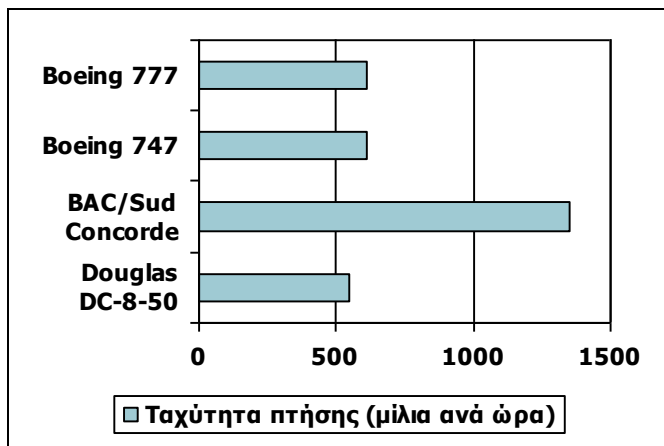
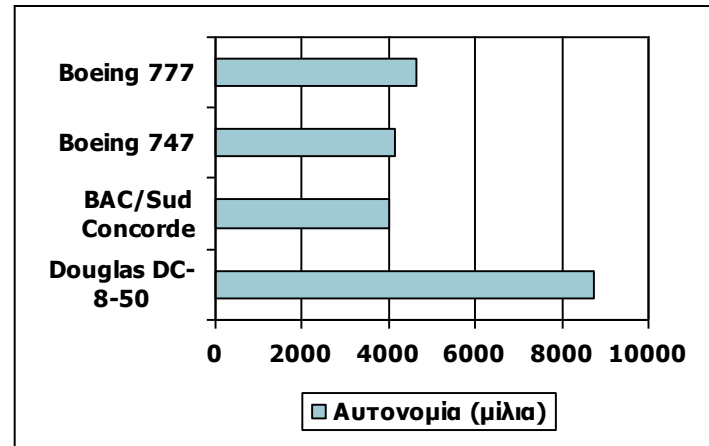
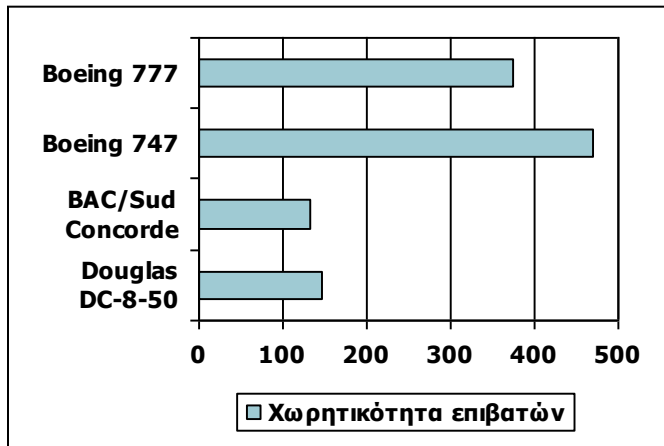
- Μεγαλύτερη χωρητικότητα και απόδοση
- Μειωμένο κόστος



| Έτος | Τεχνολογία                                        | Σχετική απόδοση/κόστος |
|------|---------------------------------------------------|------------------------|
| 1951 | Λυχνία κενού                                      | 1                      |
| 1965 | Τρανζίστορ                                        | 35                     |
| 1975 | Ολοκληρωμένο κύκλωμα                              | 900                    |
| 1995 | Πολύ μεγάλης κλίμακας ολοκληρωμένο κύκλωμα (VLSI) | 2.400.000              |
| 2005 | Υπερυψηλής κλίμακας ολοκληρωμένο κύκλωμα (ULSI)   | 6.200.000.000          |

# Ορισμός της απόδοσης

- Ποιο αεροπλάνο έχει την καλύτερη απόδοση;



# Response time – Throughput

- Χρόνος απόκρισης (response time)
  - Πόσο διαρκεί η εκτέλεση μιας εργασίας
- Διεκπεραιωτική ικανότητα (throughput)
  - Συνολική δουλειά που γίνεται ανά μονάδα χρόνου
    - π.χ. εργασίες/συναλλαγές/... ανά ώρα
- Πώς επηρεάζονται ο χρόνος απόκρισης και η διεκπεραιωτική ικανότητα από
  - Αντικατάσταση του επεξεργαστή με ταχύτερη έκδοση;
  - Προσθήκη νέων επεξεργαστών;
- Εστιάζουμε στο χρόνο απόκρισης προς το παρόν...

# Σχετική απόδοση

- Ορισμός Απόδοση =  $1/\text{Χρόνος εκτέλεσης}$
- “ο X είναι  $n$  φορές ταχύτερος από τον Y”

$$\text{Απόδοση}_X / \text{Απόδοση}_Y$$

$$= \text{Χρονος εκτέλεσης}_Y / \text{Χρόνος εκτέλεσης}_X = n$$

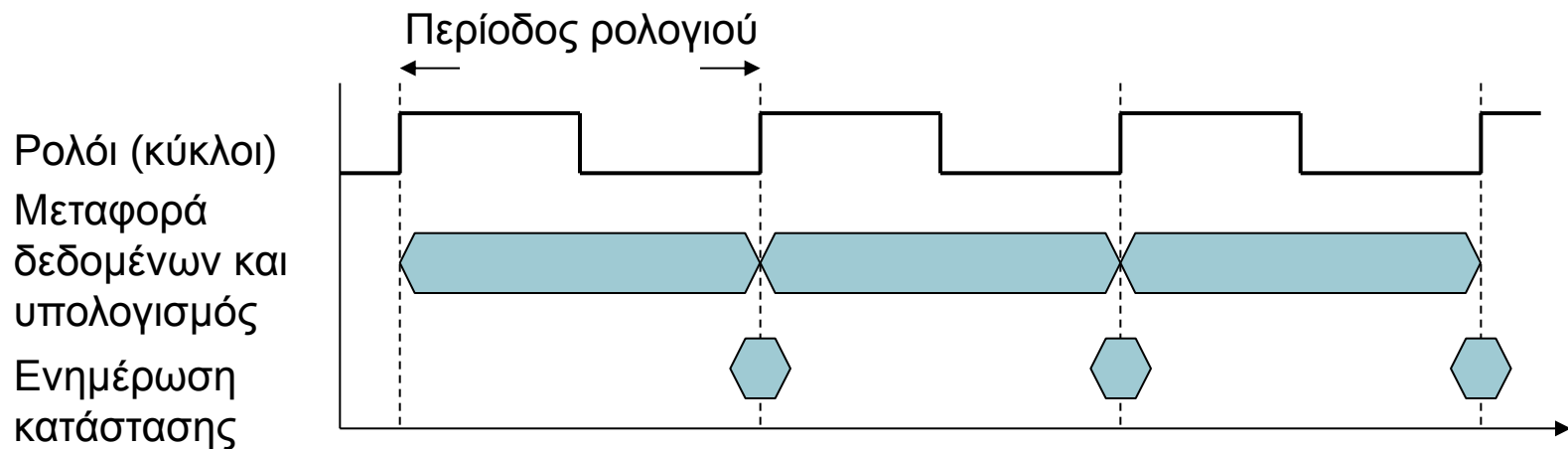
- παράδειγμα: χρόνος εκτέλεσης προγρ/τος
  - 10s στον A, 15s στον B
  - $\text{Χρόνος εκτέλεσης}_B / \text{Χρόνος εκτέλεσης}_A$   
 $= 15s / 10s = 1.5$
  - Άρα ο A είναι 1.5 φορές ταχύτερος του B

# Μέτρηση χρόνου εκτέλεσης

- «Παρελθών» χρόνος (elapsed time)
  - Συνολικός χρόνος απόκρισης, περιλαμβάνει τα πάντα
    - Επεξεργασία, είσοδο/έξοδο, επιβάρυνση ΛΣ, αδρανής χρόνος
  - Καθορίζει την απόδοση του συστήματος
- Χρόνος CPU (CPU time)
  - Ο χρόνος επεξεργασίας για μια συγκεκριμένη εργασία
    - Χωρίς το χρόνο εισόδου/εξόδου και την κοινή χρήση από άλλες εργασίες
  - Αποτελείται από το χρόνο CPU χρήστη (user CPU time) και το χρόνο CPU συστήματος (system CPU time)
  - Διαφορετικά προγράμματα επηρεάζονται διαφορετικά από την απόδοση της CPU και του συστήματος

# Χρονισμός CPU (clocking)

- Η λειτουργία του ψηφιακού υλικού ρυθμίζεται από ένα ρολόι σταθερού ρυθμού



- Περίοδος ρολογιού: η διάρκεια ενός κύκλου
  - π.χ.,  $250\text{ps} = 0.25\text{ns} = 250 \times 10^{-12}\text{s}$
- Συχνότητα (ρυθμός) ρολογιού: κύκλοι/second
  - π.χ.,  $4.0\text{GHz} = 4000\text{MHz} = 4.0 \times 10^9\text{Hz}$

# Χρόνος CPU (CPU time)

$$\begin{aligned}\text{Χρόνος CPU} &= \text{Κύκλοι ρολογιού CPU} \times \text{Χρόνος κύκλου ρολογιού} \\ &= \frac{\text{Κύκλοι ρολογιού CPU}}{\text{Ρυθμός ρολογιού}}\end{aligned}$$

- Η απόδοση βελτιώνεται με
  - Μείωση του αριθμού των κύκλων ρολογιού
  - Αύξηση του ρυθμού του ρολογιού
  - Ο σχεδιαστής του υλικού πρέπει να κάνει συχνά συμβιβασμούς μεταξύ του ρυθμού ρολογιού και του πλήθους των κύκλων ρολογιού

# Παράδειγμα χρόνου CPU

- Υπολογιστής A: ρολόι 2GHz, χρόνος CPU 10s
- Σχεδίαση του υπολογιστή B
  - Στόχος είναι χρόνος CPU 6s
  - Μπορεί το ρολόι να είναι ταχύτερο, αλλά προκαλεί αύξηση των κύκλων  $1.2 \times$  κύκλοι ρολογιού
- Πόσο ταχύτερο μπορεί να είναι το ρολόι του B;

$$\text{Ρυθμός ρολογιού}_B = \frac{\text{Κύκλοι ρολογιού}_B}{\text{Χρόνος CPU}_B} = \frac{1.2 \times \text{Κύκλοι ρολογιού}_A}{6s}$$

$$\begin{aligned}\text{Κύκλοι ρολογιού}_A &= \text{Χρόνος CPU}_A \times \text{Ρυθμός ρολογιού}_A \\ &= 10s \times 2\text{GHz} = 20 \times 10^9\end{aligned}$$

$$\text{Ρυθμός ρολογιού}_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4\text{GHz}$$

# Πλήθος εντολών και CPI

Κύκλοι ρολογιού = Πλήθος εντολών × Κύκλοι ανά εντολή

$$\begin{aligned}\text{Χρόνος CPU} &= \text{Πλήθος εντολών} \times \text{CPI} \times \text{Χρόνος κύκλου ρολογιού} \\ &= \frac{\text{Πλήθος εντολών} \times \text{CPI}}{\text{Ρυθμός ρολογιού}}\end{aligned}$$

- Πλήθος εντολών (Instruction Count) προγράμματος
  - Καθορίζεται από το πρόγραμμα, την αρχιτεκτονική συνόλου εντολών (ISA), και το μεταγλωττιστή
- Μέσος αριθμός κύκλων ανά εντολή (CPI – clocks per instruction)
  - Καθορίζεται από το υλικό της CPU
  - Αν οι διαφορετικές εντολές έχουν διαφορετικό CPI
    - Το μέσο CPI επηρεάζεται από το μίγμα των εντολών

# Παράδειγμα CPI

- Υπολογιστής A: Cycle Time = 250ps, CPI = 2.0
- Υπολογιστής B: Cycle Time = 500ps, CPI = 1.2
- Ίδια αρχιτεκτονική συνόλου εντολών (ISA)
- Ποιος είναι ταχύτερος, και κατά πόσο;

$$\begin{aligned}\text{Χρόνος CPU}_A &= \text{Πλήθος εντολών} \times \text{CPI}_A \times \text{Χρόνος κύκλου}_A \\ &= 1 \times 2.0 \times 250\text{ps} = 1 \times 500\text{ps} \quad \leftarrow \text{A ταχύτερος...}\end{aligned}$$

$$\begin{aligned}\text{Χρόνος CPU}_B &= \text{Πλήθος εντολών} \times \text{CPI}_B \times \text{Χρόνος κύκλου}_B \\ &= 1 \times 1.2 \times 500\text{ps} = 1 \times 600\text{ps}\end{aligned}$$

$$\frac{\text{Χρόνος CPU}_B}{\text{Χρόνος CPU}_A} = \frac{1 \times 600\text{ps}}{1 \times 500\text{ps}} = 1.2 \quad \leftarrow \text{...κατά τόσο}$$

# Το CPI με λεπτομέρεια

- Αν διαφορετικές κατηγορίες εντολών διαρκούν διαφορετικό αριθμό κύκλων

$$\text{Κύκλοι ρολογιού} = \sum_{i=1}^n (\text{CPI}_i \times \text{Πλήθος εντολών}_i)$$

- Σταθμισμένο (weighted) μέσο CPI

$$\text{CPI} = \frac{\text{Κύκλοι ρολογιού}}{\text{Πλήθος εντολών}} = \sum_{i=1}^n \left( \text{CPI}_i \times \frac{\text{Πλήθος εντολών}_i}{\text{Πλήθος εντολών}} \right)$$

Σχετική συχνότητα εντολών

# Παράδειγμα CPI

- Εναλλακτικές ακολουθίες μεταγλωττισμένου κώδικα με εντολές τριών κατηγοριών A, B, C

| Κατηγορία                        | A | B | C |
|----------------------------------|---|---|---|
| CPI της κατηγορίας               | 1 | 2 | 3 |
| Πλήθος εντολών (IC) ακολουθίας 1 | 2 | 1 | 2 |
| Πλήθος εντολών (IC) ακολουθίας 2 | 4 | 1 | 1 |

- Ακολουθία 1: IC = 5
  - Κύκλοι ρολογιού  
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$   
 $= 10$
  - Μέσο CPI =  $10/5 = 2.0$
- Ακολουθία 2: IC = 6
  - Κύκλοι ρολογιού  
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$   
 $= 9$
  - Μέσο CPI =  $9/6 = 1.5$

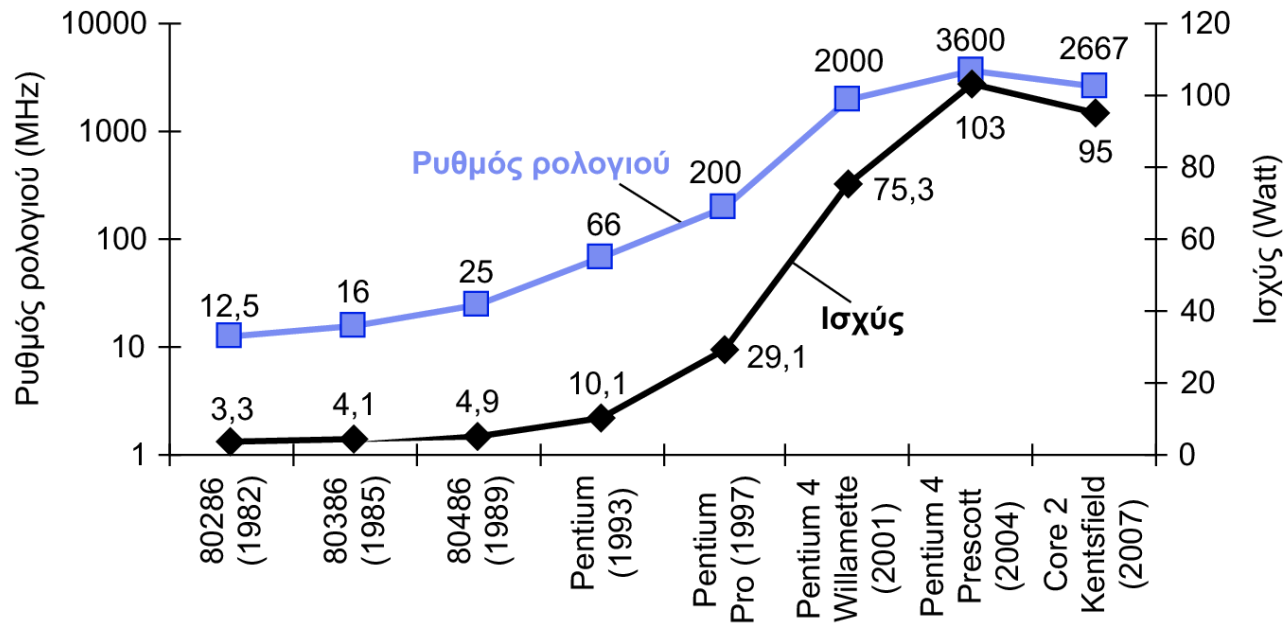
# Σύνοψη της απόδοσης

## ΓΕΝΙΚΗ εικόνα

$$\text{Χρόνος CPU} = \frac{\text{Εντολές}}{\text{Πρόγραμμα}} \times \frac{\text{Κύκλοι ρολογιού}}{\text{Εντολή}} \times \frac{\text{Δευτερόλεπτα}}{\text{Κύκλος ρολογιού}}$$

- Η απόδοση εξαρτάται από
  - Αλγόριθμο: επηρεάζει το πλήθος εντολών, πιθανόν και το CPI
  - Γλώσσα προγραμματισμού: επηρεάζει το πλήθος εντολών και το CPI
  - Μεταγλωττιστής: επηρεάζει το πλήθος εντολών και το CPI
  - Αρχιτεκτονική συνόλου εντολών (ISA): επηρεάζει το πλήθος εντολών, το CPI, και την περίοδο (συχνότητα) του ρολογιού

# Οι τάσεις στην ηλεκτρική ισχύ



- Στη τεχνολογία ολοκληρωμένων κυκλωμάτων CMOS

$$\text{Ισχύς} = \text{Φορτίο χωρητικότητα} \times \text{Τάση}^2 \times \text{Συχνότητα}$$

×30

5V → 1V

×1000

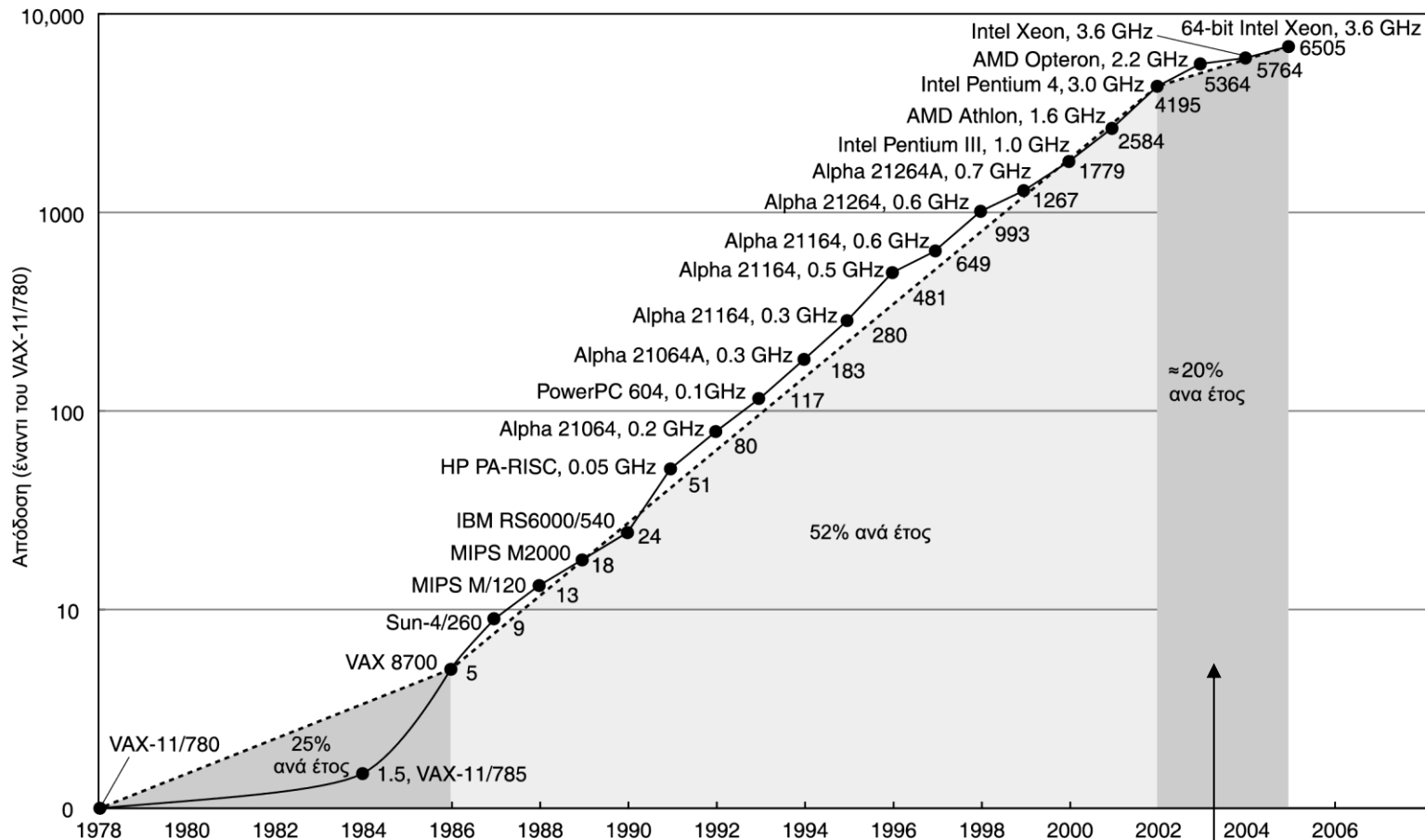
# Μείωση της ισχύος

- Υποθέστε ότι μια νέα CPU έχει
  - 85% του φορτίου χωρητικότητας (capacitive load) της παλιάς CPU
  - μειωμένη τάση κατά 15% και συχνότητα κατά 15%

$$\frac{P_{\text{new}}}{P_{\text{old}}} = \frac{C_{\text{old}} \times 0.85 \times (V_{\text{old}} \times 0.85)^2 \times F_{\text{old}} \times 0.85}{C_{\text{old}} \times V_{\text{old}}^2 \times F_{\text{old}}} = 0.85^4 = 0.52$$

- Το τείχος της ισχύος (power wall)
  - Δεν μπορούμε να μειώσουμε άλλο την τάση
  - Δεν μπορούμε να απαγάγουμε τη θερμότητα
- Πώς αλλιώς μπορούμε να βελτιώσουμε την απόδοση;

# Απόδοση μονοεπεξεργαστών

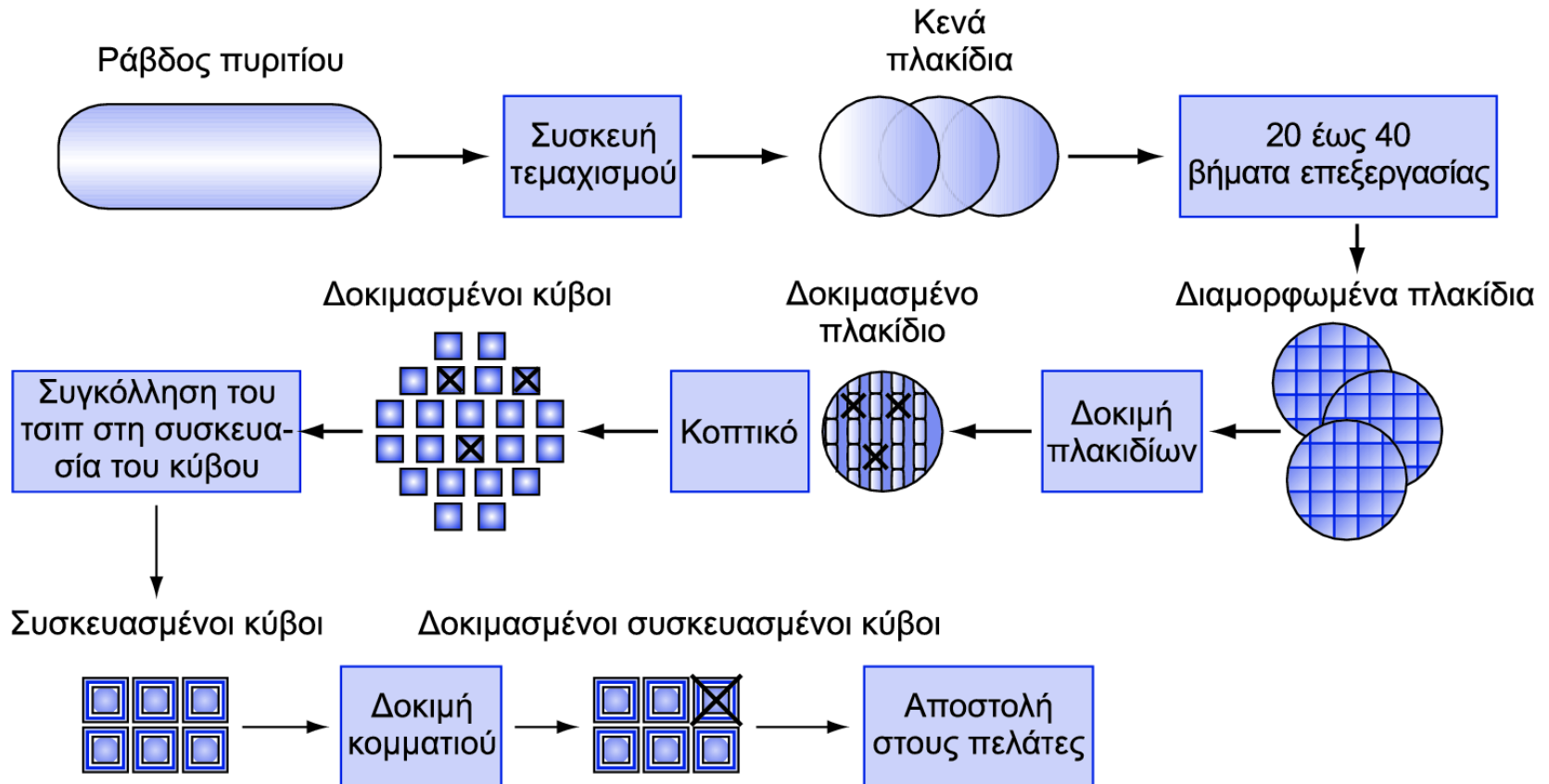


Περιορίζεται από την ισχύ, την παραλληλία επιπέδου εντολής (instruction-level parallelism), το λανθάνοντα χρόνο μνήμης (memory latency)

# Πολυεπεξεργαστές

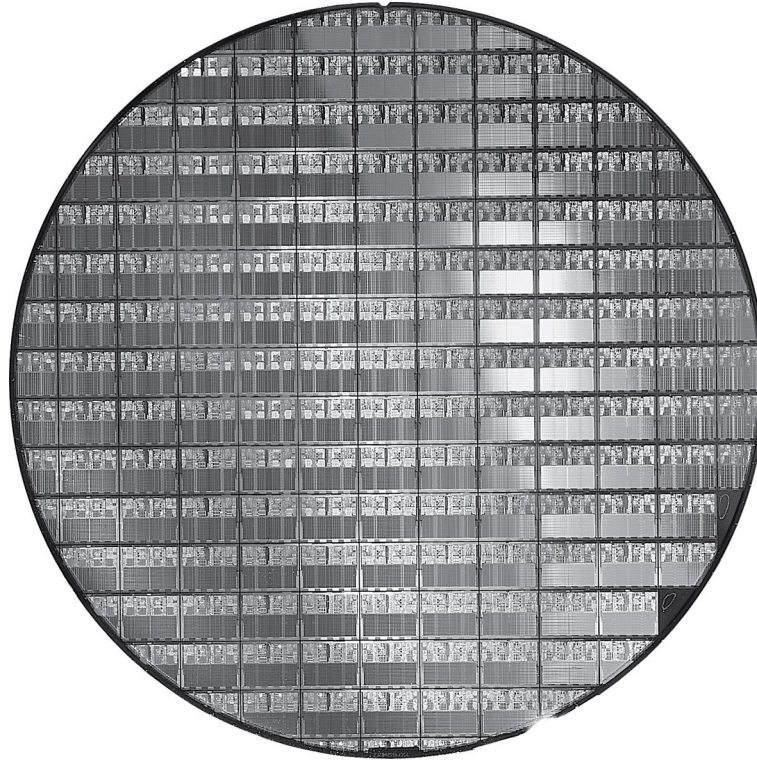
- Πολυπύρρηνοι μικροεπεξεργαστές (multicore microprocessors)
  - Περισσότεροι του ενός επεξεργαστές ανά τσιπ
- Απαιτεί ρητό παράλληλο προγραμματισμό (explicitly parallel programming)
  - Σύγκριση με παραλληλία επιπέδου εντολής (instruction level parallelism – ILP)
    - Το υλικό εκτελεί πολλές εντολές ταυτόχρονα
    - Ο προγραμματιστής δεν το γνωρίζει
  - Μεγάλες δυσκολίες
    - Προγραμματισμός για μέγιστη απόδοση
    - Εξισορρόπηση φορτίου (load balancing)
    - Βελτιστοποίηση επικοινωνίας (communication) και συγχρονισμού (synchronization)

# Κατασκευή ολοκληρωμένων



- **Εσοδεία (yield):** ποσοστό τσιπ ανά πλακίδιο (wafer) που λειτουργούν σωστά

# AMD Opteron X2 Wafer



- X2: 300mm πλακίδιο, 117 τσιπ, τεχνολογία 90nm
- X4: τεχνολογία 45nm

# Κόστος Ολοκληρωμένων

$$\text{Κόστος ανά κύβο} = \frac{\text{Κόστος ανά πλακίδιο}}{\text{Κύβοι ανά πλακίδιο} \times \text{Εσοδεία}}$$

$$\text{Κύβοι ανά πλακίδιο} \approx \text{Επιφάνεια πλακιδίου} / \text{Επιφάνεια κύβου}$$

$$\text{Εσοδεία} = \frac{1}{(1 + \text{Ατέλειες ανά μονάδα επιφανείας} \times \text{Επιφάνεια κύβου} / 2))^2}$$

- Μη γραμμική εξάρτηση από την επιφάνεια (area) και το ρυθμό ατελειών (defect rate)
  - Κόστος και επιφάνεια πλακιδίου (wafer): σταθερά
  - Ρυθμός ατελειών (defect rate) εξαρτάται από τη διαδικασία κατασκευής
  - Επιφάνεια τσιπ (die/chip area): εξαρτάται από την αρχιτεκτονική και τη σχεδίαση του κυκλώματος

# Μετροπρογράμματα SPEC CPU

- Χρησιμοποιούνται προγράμματα για τη μέτρηση της απόδοσης
  - Υποτίθεται τυπικά για ένα πραγματικό φορτίο εργασίας (workload)
- Standard Performance Evaluation Corp (SPEC)
  - Αναπτύσσει μετροπρογράμματα (benchmarks) για CPU, είσοδο/έξοδο, Ιστό, ...
- SPEC CPU2006
  - «Παρελθών» (elapsed) χρόνος για την εκτέλεση μιας συλλογής προγραμμάτων
    - Αμελητέα είσοδος/έξοδος, άρα εστιάζουν στην απόδοση της CPU
  - Κανονικοποίηση σε σχέση με μια μηχανή αναφοράς (reference machine)
  - Σύνοψη ως γεωμετρικός μέσος (geometric mean) των λόγων απόδοσης (performance ratios)
    - CINT2006 (integer) and CFP2006 (floating-point)

$$\sqrt[n]{\prod_{i=1}^n \text{Λόγος χρόνου εκτέλεσης}_i}$$

# CINT2006 για Opteron X4 2356

| Όνομα          | Περιγραφή                     | IC×10 <sup>9</sup> | CPI   | Tc (ns) | Χρόν. εκτ. | Χρον. ανφ. | SPECratio |
|----------------|-------------------------------|--------------------|-------|---------|------------|------------|-----------|
| perl           | Interpreted string processing | 2,118              | 0.75  | 0.40    | 637        | 9,777      | 15.3      |
| bzip2          | Block-sorting compression     | 2,389              | 0.85  | 0.40    | 817        | 9,650      | 11.8      |
| gcc            | GNU C Compiler                | 1,050              | 1.72  | 0.47    | 24         | 8,050      | 11.1      |
| mcf            | Combinatorial optimization    | 336                | 10.00 | 0.40    | 1,345      | 9,120      | 6.8       |
| go             | Go game (AI)                  | 1,658              | 1.09  | 0.40    | 721        | 10,490     | 14.6      |
| hmmer          | Search gene sequence          | 2,783              | 0.80  | 0.40    | 890        | 9,330      | 10.5      |
| sjeng          | Chess game (AI)               | 2,176              | 0.96  | 0.48    | 37         | 12,100     | 14.5      |
| libquantum     | Quantum computer simulation   | 1,623              | 1.61  | 0.40    | 1,047      | 20,720     | 19.8      |
| h264avc        | Video compression             | 3,102              | 0.80  | 0.40    | 993        | 22,130     | 22.3      |
| omnetpp        | Discrete event simulation     | 587                | 2.94  | 0.40    | 690        | 6,250      | 9.1       |
| astar          | Games/path finding            | 1,082              | 1.79  | 0.40    | 773        | 7,020      | 9.1       |
| xalancbmk      | XML parsing                   | 1,058              | 2.70  | 0.40    | 1,143      | 6,900      | 6.0       |
| Geometric mean |                               |                    |       |         |            |            | 11.7      |

Υψηλοί ρυθμοί αστοχίας κρυφής  
μνήμης (cache misses)

# Μετροπρογράμματα SPEC Power

- Κατανάλωση ισχύος διακομιστή (server) σε διαφορετικά επίπεδα φορτίου εργασίας
  - Απόδοση: ssj\_ops/sec
  - Ισχύς: Watts (Joules/sec)

$$\text{Overall ssj\_ops per Watt} = \left( \sum_{i=0}^{10} \text{ssj\_ops}_i \right) / \left( \sum_{i=0}^{10} \text{power}_i \right)$$

# SPECpower\_ssj2008 για X4

| Φορτίο στόχου %  | Απόδοση (ssj_ops/sec) | Μέση ισχύς (Watts) |
|------------------|-----------------------|--------------------|
| 100%             | 231,867               | 295                |
| 90%              | 211,282               | 286                |
| 80%              | 185,803               | 275                |
| 70%              | 163,427               | 265                |
| 60%              | 140,160               | 256                |
| 50%              | 118,324               | 246                |
| 40%              | 920,35                | 233                |
| 30%              | 70,500                | 222                |
| 20%              | 47,126                | 206                |
| 10%              | 23,066                | 180                |
| 0%               | 0                     | 141                |
| Overall sum      | 1,283,590             | 2,605              |
| Σssj_ops/ Σpower |                       | 493                |

# Παγίδα: νόμος του Amdahl

- Η βελτίωση μιας πλευράς ενός υπολογιστή και η αναμονή ανάλογης βελτίωσης της συνολικής απόδοσης

$$T_{\text{μετά τη βελτίωση}} = \frac{T_{\text{που επηρεάζεται}}}{\text{συντελεστής βελτίωσης}} + T_{\text{που δεν επηρεάζεται}}$$

- Παράδειγμα: ο πολ/σμός είναι τα 80s/100s
  - Πόση βελτίωση της απόδοσης του πολ/σμού ώστε η συνολική απόδοση να 5-πλασιαστεί;

$$20 = \frac{80}{n} + 20 \quad \text{■ Δεν γίνεται!}$$

- Πόρισμα: κάνε τη συνηθισμένη περίπτωση γρήγορη

# Πλάνη: Χαμηλή ισχύς αδράνειας

- Δείτε το μετροπρόγραμμα ισχύος στον X4
  - Στο 100% του φορτίου: 295W
  - Στο 50% του φορτίου: 246W (83%)
  - Στο 10% του φορτίου: 180W (61%)
- Κέντρο δεδομένων Google
  - Κυρίως λειτουργεί στο 10% – 50% του φορτίου
  - Με φορτίο 100% σε λιγότερο από 1% του χρόνου
- Σκεφθείτε τη σχεδίαση επεξεργαστών με κατανάλωση ισχύος ανάλογη με το φορτίο

# Παγίδα: μέτρο απόδοσης MIPS

- MIPS: Millions of Instructions Per Second
  - Δε λαμβάνει υπόψη:
    - Διαφορές ISA μεταξύ υπολογιστών
    - Διαφορές πολυπλοκότητας μεταξύ εντολών

$$\begin{aligned} \text{MIPS} &= \frac{\text{Πλήθος εντολών}}{\text{Χρόνος εκτέλεσης} \times 10^6} \\ &= \frac{\text{Πλήθος εντολών}}{\frac{\text{Πλήθος εντολών} \times \text{CPI}}{\text{Ρυθμός ρολογιού}} \times 10^6} = \frac{\text{Ρυθμός ρολογιού}}{\text{CPI} \times 10^6} \end{aligned}$$

- Το CPI ποικίλει μεταξύ προγραμμάτων σε μια δεδομένη CPU

# Συμπερασματικές παρατηρήσεις

- Ο λόγος κόστος/απόδοση βελτιώνεται
  - Λόγω της εξέλιξης της τεχνολογίας
- Ιεραρχικά επίπεδα αφαίρεσης
  - Στο υλικό και στο λογισμικό
- Αρχιτεκτονική συνόλου εντολών (Instruction set architecture – ISA)
  - Η διασύνδεση υλικού και λογισμικού
- Χρόνος εκτέλεσης: το καλύτερο μέτρο απόδοσης
- Η ισχύς είναι περιοριστικός παράγοντας
  - Χρήση παραλληλίας για βελτίωση της απόδοσης