



# ΟΡΓΑΝΩΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

## Η διασύνδεση υλικού και λογισμικού



### Κεφάλαιο 1

## Αφηρημένες έννοιες και τεχνολογία υπολογιστών

# Η επανάσταση των υπολογιστών

- Πρόοδος στην τεχνολογία των Η/Υ
  - Στηριγμένη σε επιταχυντές συγκεκριμένου τομέα
- Καθιστά εφικτή τη δημιουργία καινοτόμων εφαρμογών
  - Υπολογιστές σε αυτοκίνητα
  - Κινητά τηλέφωνα
  - Ερευνητικό πρόγραμμα του ανθρωπίνου γονιδιώματος
  - Παγκόσμιος Ιστός
  - Μηχανές αναζήτησης
- Η χρήση των υπολογιστών εξαπλώνεται παντού

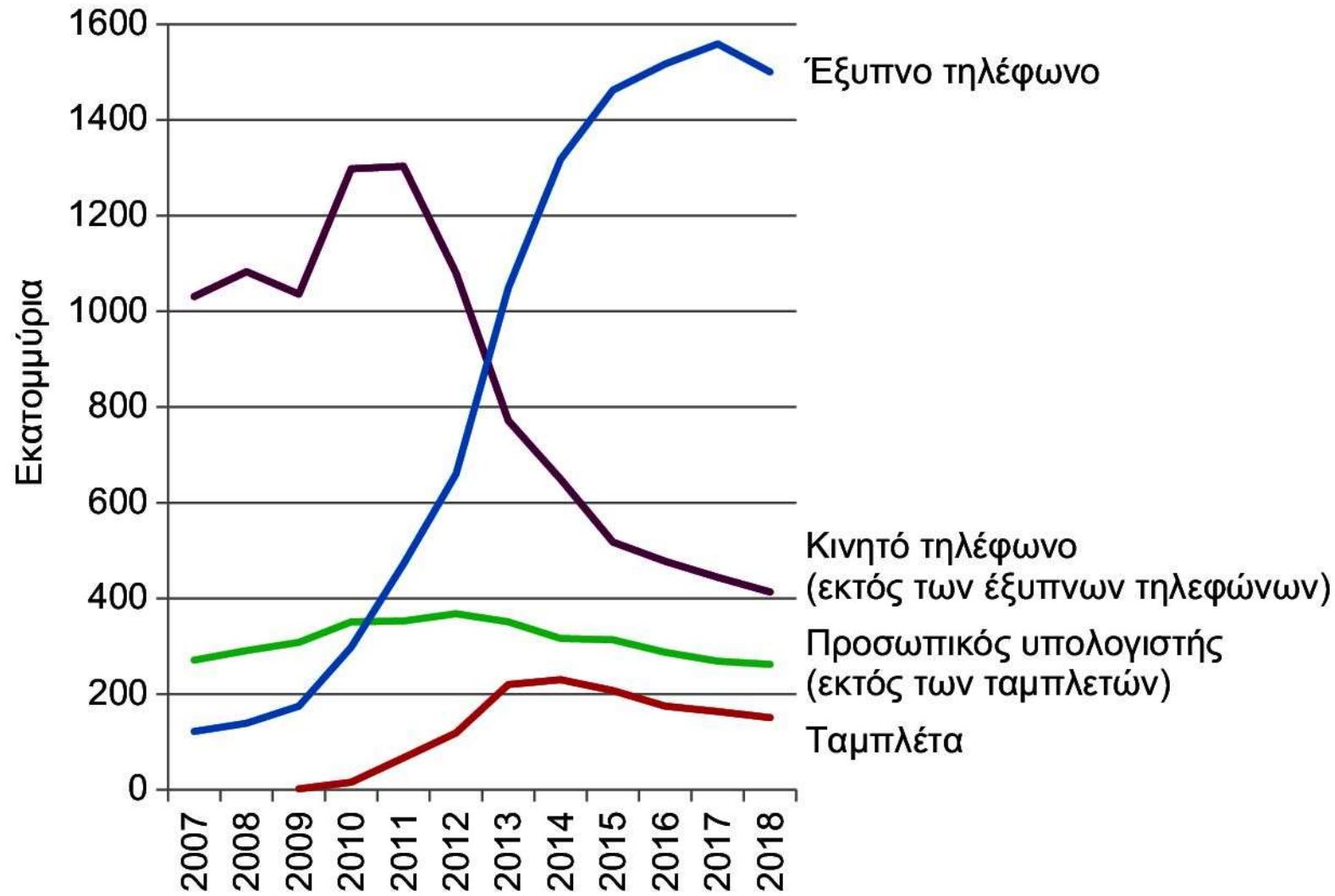
# Κατηγορίες υπολογιστών

- Προσωπικοί υπολογιστές
  - Γενικού σκοπού, ποικιλία λογισμικού
  - Υπόκεινται σε συμβιβασμούς μεταξύ κόστους και απόδοσης
- Διακομιστές
  - Δικτυακοί
  - Υψηλή παραγωγική δυναμικότητα, χωρητικότητα, απόδοση, αξιοπιστία
  - Το μέγεθός τους κυμαίνεται από μικρό έως τεράστιο (έως και ολόκληρα κτίρια)

# Κατηγορίες υπολογιστών

- Υπερυπολογιστές
  - Τύπος διακομιστή
  - Υψηλών απαιτήσεων επιστημονικοί και τεχνικοί υπολογισμοί
  - Κορυφαίοι σε υπολογιστικές δυνατότητες, αλλά αποτελούν ένα σχετικά μικρό κλάσμα της συνολικής αγοράς υπολογιστών
- Ενσωματωμένοι υπολογιστές
  - «Κρυμμένοι» ως στοιχεία συστημάτων
  - Αυστηροί περιορισμοί ισχύος/απόδοσης/κόστους

# Η «εποχή μετά τον PC»



# Η «εποχή μετά τον PC»

- Προσωπική φορητή συσκευή (personal mobile device, PMD)
  - Λειτουργούν με μπαταρία
  - Συνδέονται στο Διαδίκτυο
  - Κοστίζουν εκατοντάδες ευρώ
  - «Έξυπνα τηλέφωνα», υπολογιστές-ταμπλέτες, ηλεκτρονικά γυαλιά
- Υπολογιστική νέφους
  - Υπολογιστές κλίμακας αποθήκης (WSC)
  - Λογισμικό ως υπηρεσία (software as a service, SaaS)
  - Μέρος του λογισμικού εκτελείται στην προσωπική φορητή συσκευή και ένα άλλο μέρος εκτελείται στο Νέφος
  - Amazon και Google

# Τι θα μάθετε

- Πώς τα προγράμματα μεταφράζονται σε γλώσσα μηχανής
  - Πώς εκτελούνται από το υλικό
- Τη διασύνδεση υλικού/λογισμικού
- Τι καθορίζει την απόδοση ενός προγράμματος
  - Και πώς μπορεί να βελτιωθεί
- Πώς οι σχεδιαστές υλικού βελτιώνουν την απόδοση
- Τι είναι η παράλληλη επεξεργασία

# Κατανόηση της απόδοσης

- Αλγόριθμος
  - Καθορίζει τον αριθμό των εκτελούμενων λειτουργιών
- Γλώσσα προγραμματισμού, μεταγλωττιστής, αρχιτεκτονική
  - Καθορίζει τον αριθμό των εντολών γλώσσας μηχανής που εκτελούνται ανά λειτουργία
- Επεξεργαστής και σύστημα μνήμης
  - Καθορίζουν την ταχύτητα εκτέλεσης των εντολών
- Σύστημα Ε/Ε (μαζί με το λειτουργικό σύστημα)
  - Καθορίζει την ταχύτητα εκτέλεσης των λειτουργιών εισόδου/εξόδου

# Επτά σπουδαίες ιδέες

- Χρήση **αφαιρέσεων** για την απλοποίηση της σχεδίασης
- Επιτάχυνση της **πιο κοινής (συνηθισμένης) περίπτωσης**
- Απόδοση μέσω **παραλληλίας**
- Απόδοση μέσω **διοχέτευσης**
- Απόδοση μέσω **πρόβλεψης**
- **Ιεραρχία** μνημών
- **Φερεγγυότητα** μέσω πλεονασμού



# Κάτω από το πρόγραμμά σας



- Λογισμικό εφαρμογών
  - Γραμμένο σε γλώσσα υψηλού επιπέδου
- Λογισμικό συστημάτων
  - Μεταγλωττιστής: μεταφράζει τον κώδικα της γλώσσας υψηλού επιπέδου σε κώδικα γλώσσας μηχανής
  - Λειτουργικό σύστημα: ο κώδικας των υπηρεσιών συστήματος
    - Διαχείριση λειτουργιών εισόδου/εξόδου
    - Διαχείριση μνήμης και αποθήκευσης
    - Χρονοπρογραμματισμός εργασιών και κοινή χρήση πόρων
- ΥΛΙΚΟ
  - Επεξεργαστής, μνήμη, ελεγκτές Ε/Ε

# Επίπεδα κώδικα προγραμμάτων

- Γλώσσα υψηλού επιπέδου
  - Επίπεδο αφαίρεσης πιο κοντά στον τομέα του προβλήματος
  - Εξασφαλίζει παραγωγικότητα και φορητότητα
- Συμβολική γλώσσα (assembly language)
  - Αναπαράσταση εντολών σε μορφή κειμένου
- Αναπαράσταση υλικού
  - Δυαδικά ψηφία (binary digits, bits)
  - Κωδικοποιημένες εντολές και δεδομένα

High-level  
language  
program  
(in C)

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

Assembly  
language  
program  
(for RISC-V)

```
slli x6, x11, 3
add x6, x10, x6
lw x5, 0(x6)
lw x7, 4(x6)
sw x7, 0(x6)
sw x5, 4(x6)
jalr x0, 0(x1)
```

Binary machine  
language  
program  
(for RISC-V)

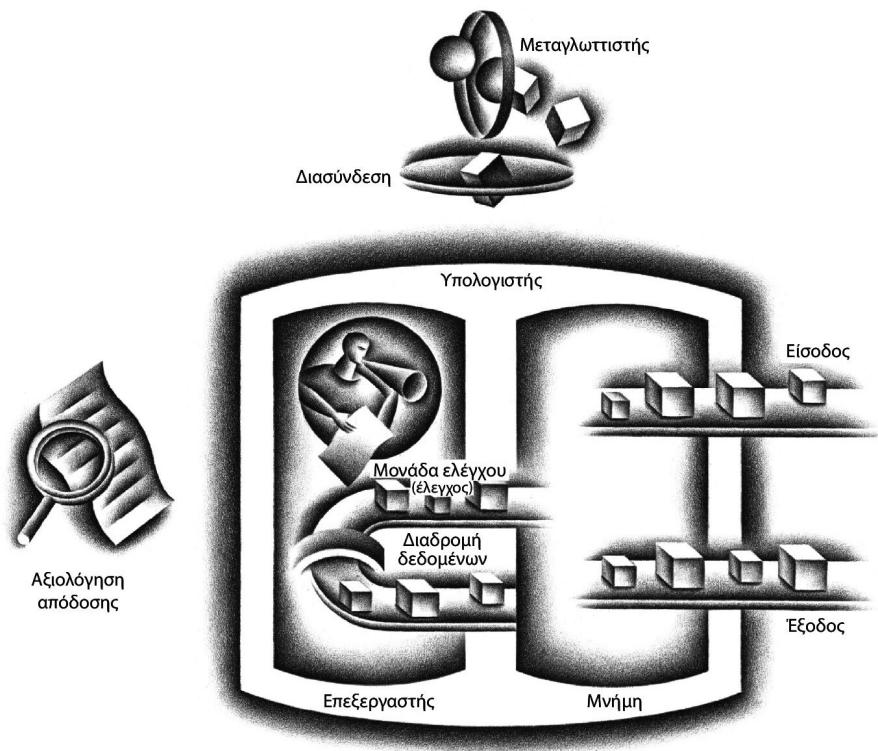
```
00000000001101011001001100010011
00000000011001010000001100110011
0000000000000000110011001010000011
000000001000001100110011001110000011
00000000011100110011000000100011
00000000010100110011010000100011
000000000000000000001000000001100111
```

Μεταγλωτιστής

Συμβολομεταφραστής

# Τα συστατικά στοιχεία του υπολογιστή

## η ΓΕΝΙΚΗ εικόνα

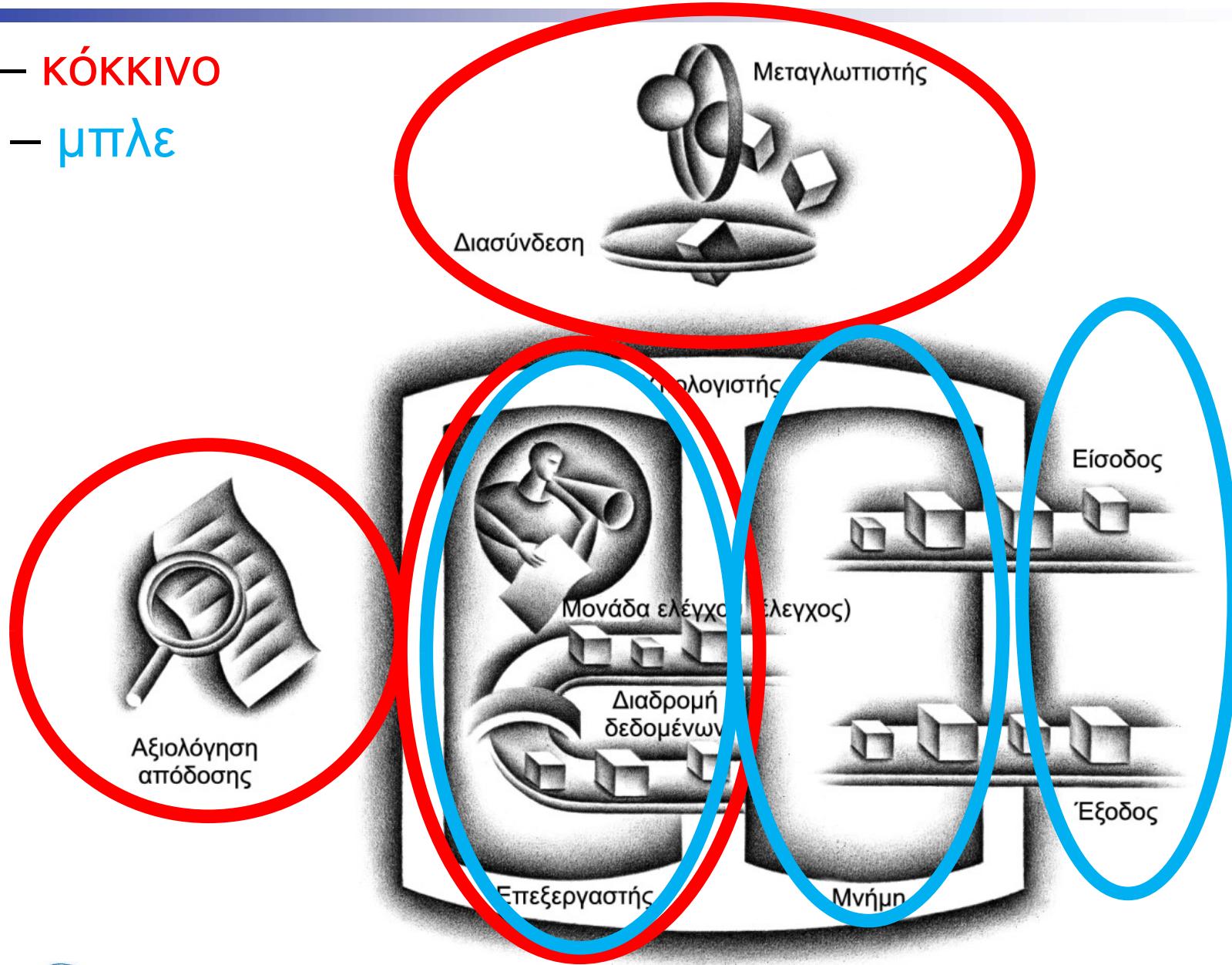


- Συστατικά στοιχεία ίδια για όλα τα είδη υπολογιστών
  - Επιτραπέζιους, διακομιστές, ενσωματωμένους υπολογιστές
- Η είσοδος/έξοδος περιλαμβάνει
  - Συσκευές για τη διασύνδεση με τον χρήστη
    - Οθόνη, πληκτρολόγιο, ποντίκι
  - Συσκευές αποθήκευσης
    - Σκληροί δίσκοι, CD/DVD, μονάδες μνήμης/δίσκων flash
  - Προσαρμογείς δικτύου
    - Για επικοινωνία με άλλους υπολογιστές

**Κεφάλαιο 1 – Αφηρημένες έννοιες και τεχνολογία υπολογιστών – 12**

# Αρχιτεκτονική I και II

- I – Κόκκινο
- II – μπλε



Κεφάλαιο 1 – Αφηρημένες έννοιες και τεχνολογία υπολογιστών – 13

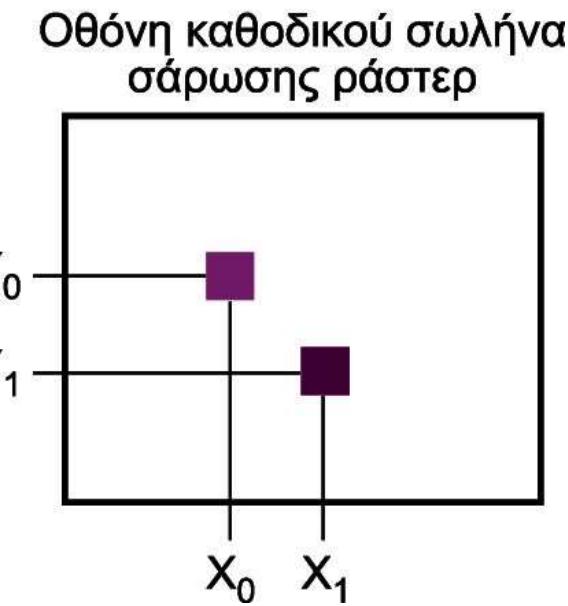
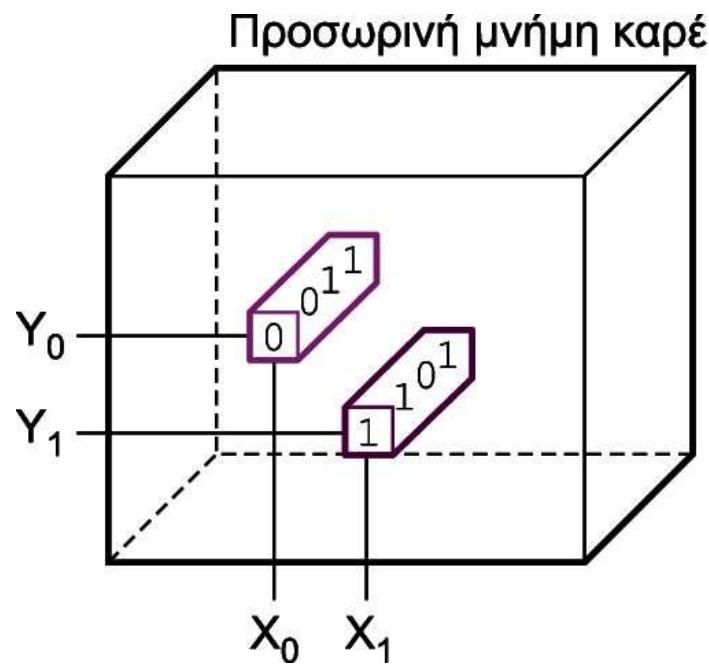
# Οθόνη αφής

- Συσκευή της «εποχής μετά τον PC»
- Υποκαθιστά πληκτρολόγιο και ποντίκι
- Δύο τύπων: τεχνολογίας πίεσης και τεχνολογίας αίσθησης χωρητικότητας
  - Τεχνολογία αίσθησης χωρητικότητας στα περισσότερα έξυπνα τηλέφωνα και υπολογιστές-ταμπλέτες
  - Η τεχνολογία αίσθησης χωρητικότητας επιτρέπει πολλά αγγίγματα της οθόνης ταυτόχρονα

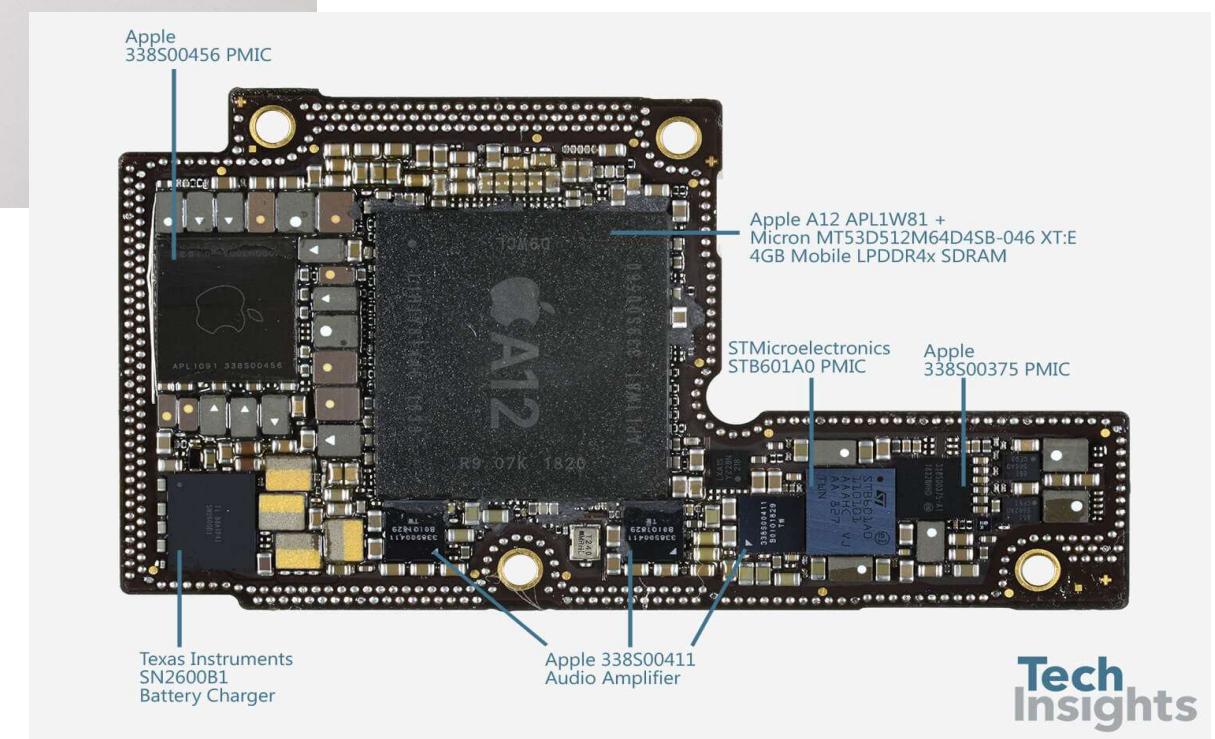
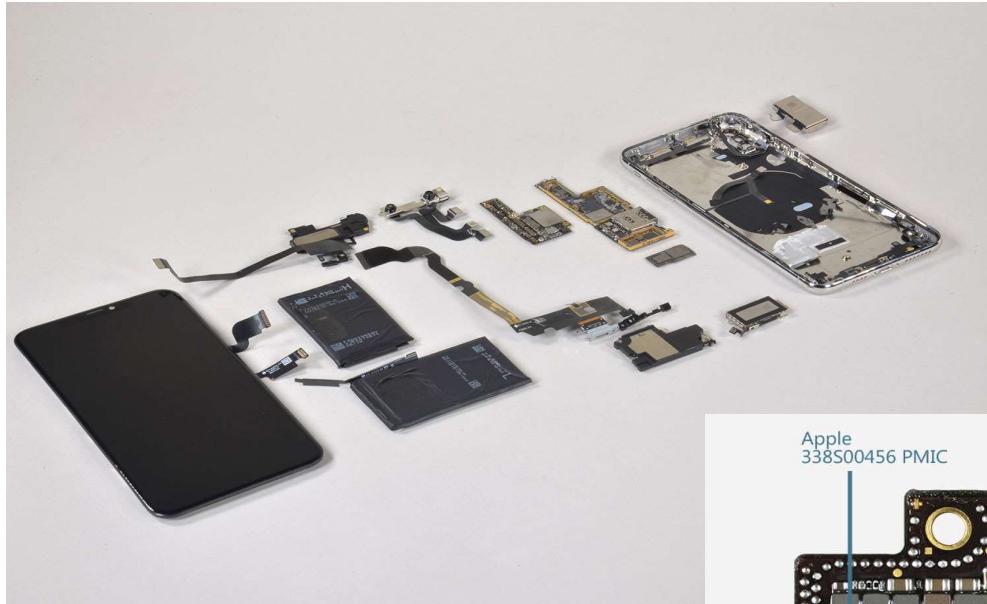


# Μέσα από τον καθρέφτη

- Οθόνη LCD: εικονοστοιχεία (picture elements, pixels)
  - Αντικατοπτρίζει το περιεχόμενο της προσωρινής μνήμης καρέ



# Το άνοιγμα του κουτιού



# Μέσα στον επεξεργαστή (CPU)

- Διαδρομή δεδομένων (datapath): εκτελεί τις λειτουργίες στα δεδομένα
- Μονάδα ελέγχου (control): Ορίζει την αλληλουχία λειτουργίας της διαδρομής δεδομένων, της μνήμης, ...
- Κρυφή μνήμη (cache memory)
  - Μικρή, γρήγορη μνήμη SRAM για άμεση προσπέλαση δεδομένων

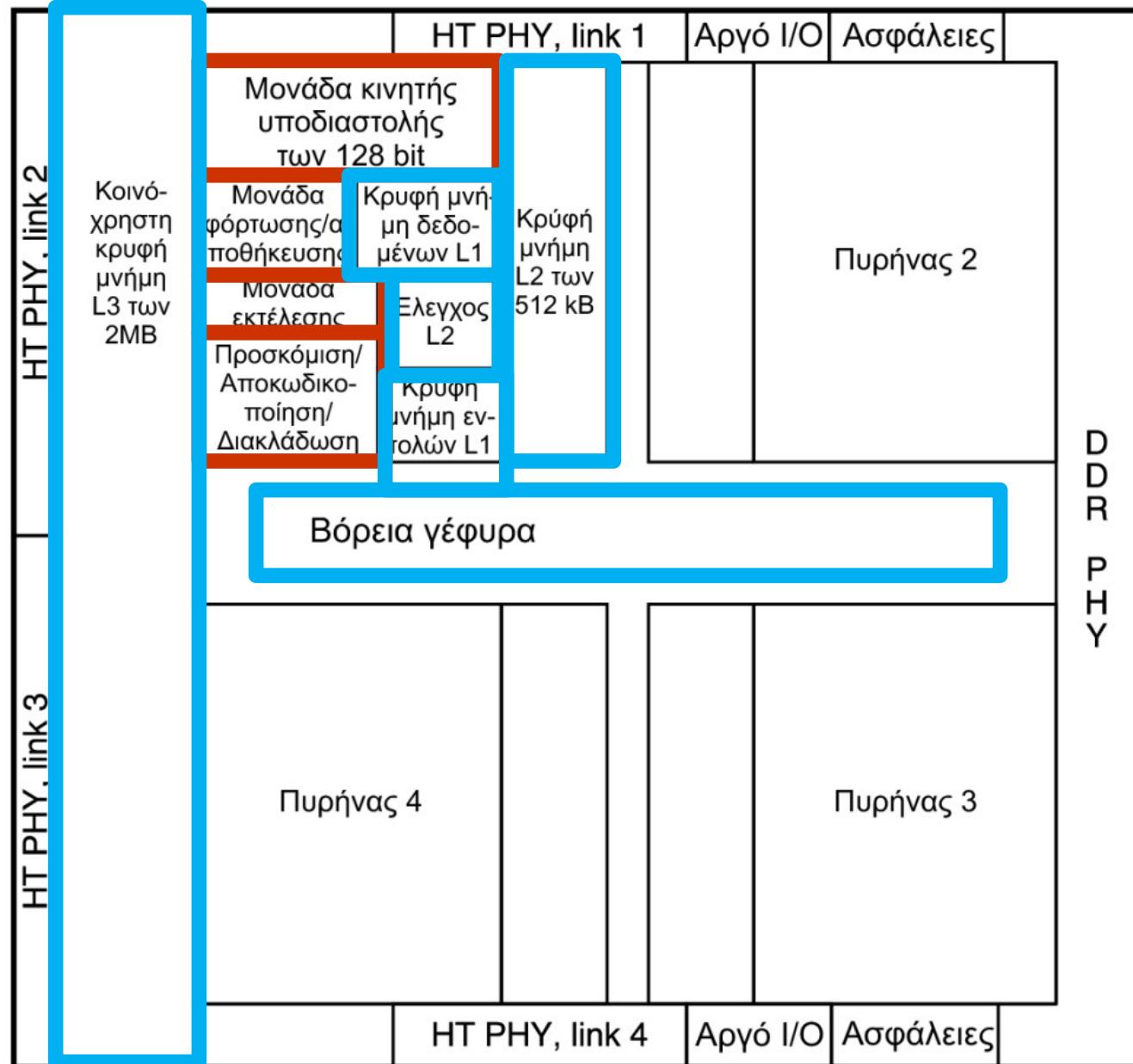
# Μέσα στον επεξεργαστή

## ■ Επεξεργαστής A12



# Αρχιτεκτονική I και II

- **Αρχιτεκτονική I**
- **σύνολο εντολών, επεξεργαστής**
- **Αρχιτεκτονική II**
- **προηγμένες τεχνικές επεξεργαστή, μνήμες και είσοδος/έξοδος**



Κεφάλαιο 1 – Αφηρημένες έννοιες και τεχνολογία υπολογιστών – 19

# Αφαίρεση (abstraction)

## η ΓΕΝΙΚΗ εικόνα

- Η αφαίρεση μας βοηθά να αντιμετωπίζουμε την πολυπλοκότητα
  - Αποκρύπτονται οι χαμηλού επιπέδου λεπτομέρειες
- Αρχιτεκτονική συνόλου εντολών (ISA)
  - Διασύνδεση υλικού/λογισμικού
- Δυαδική διασύνδεση εφαρμογής (ABI)
  - Η ISA συν η διασύνδεση λογισμικού του συστήματος
- Υλοποίηση
  - Οι υποκείμενες λεπτομέρειες και διασυνδέσεις

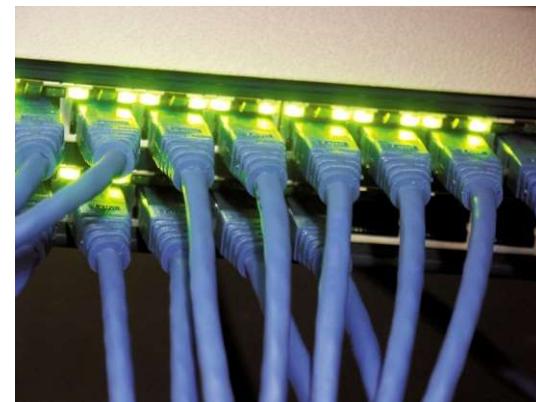
# Ένα ασφαλές μέρος για δεδομένα

- Πτητική (volatile) κύρια μνήμη
  - Οι εντολές και τα δεδομένα χάνονται όταν διακόπτεται η τροφοδοσία
- Μη πτητική (non-volatile)  
δευτερεύουσα μνήμη
  - Μαγνητικοί δίσκοι
  - Μνήμη φλας
  - Οπτικοί δίσκοι (CDROM, DVD)



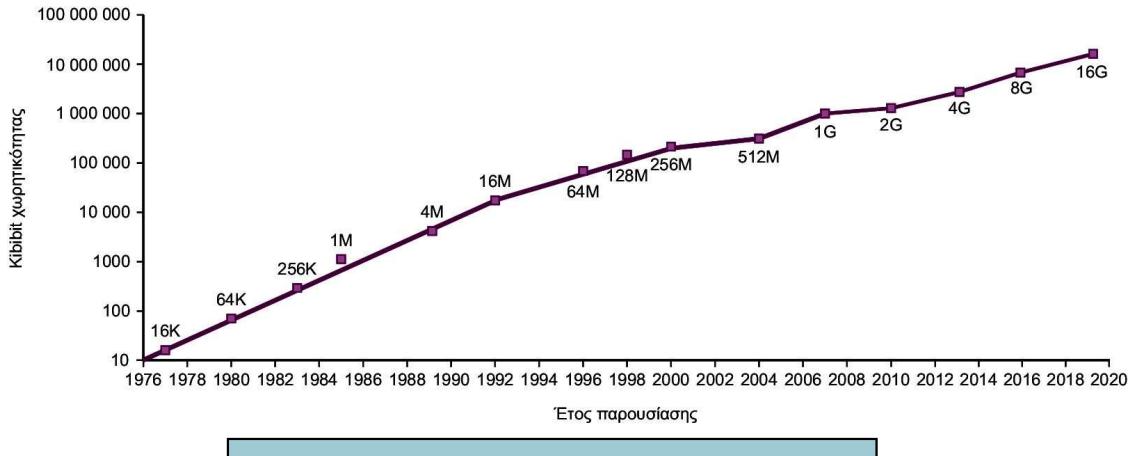
# Δίκτυα

- Επικοινωνία, κοινή χρήση πόρων, μη τοπική πρόσβαση (non-local access)
- Τοπικό δίκτυο (local area network, LAN): Ethernet
- Δίκτυο ευρείας περιοχής (wide area network, WAN): το Διαδίκτυο
- Ασύρματα δίκτυα: WiFi, Bluetooth



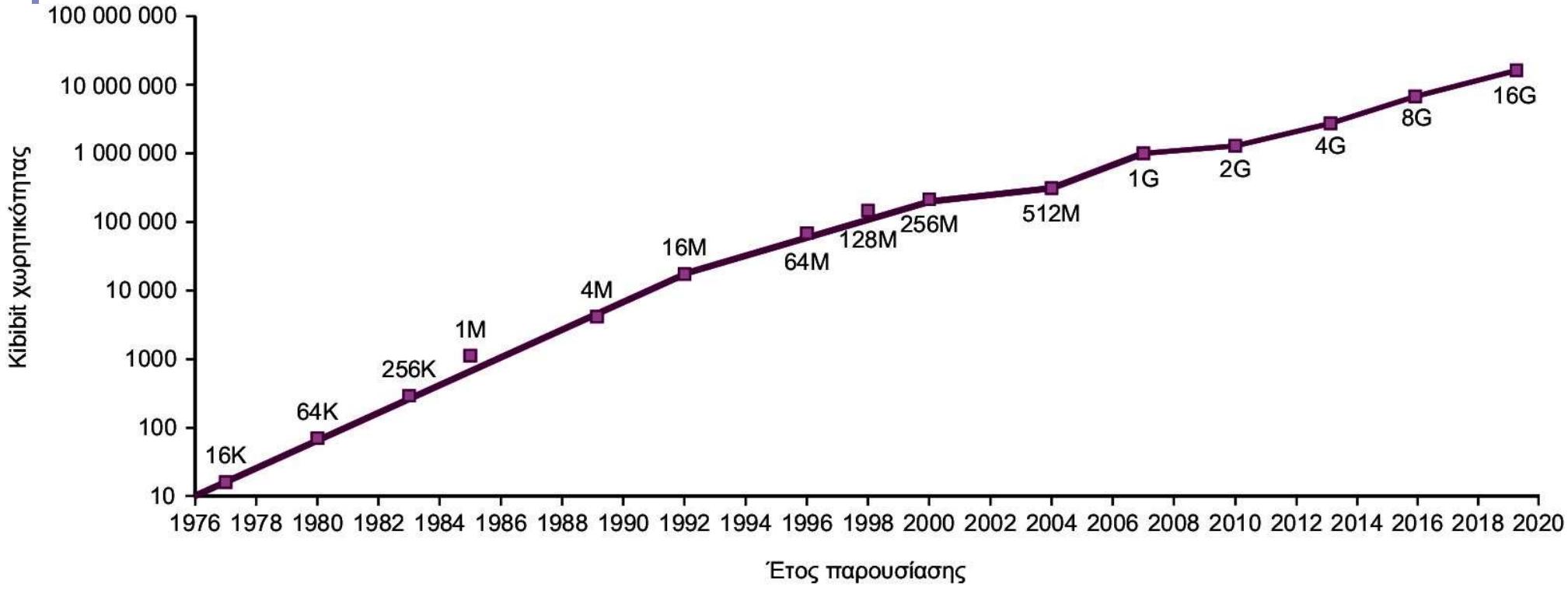
# Τάσεις της τεχνολογίας

- Η τεχνολογία των ηλεκτρονικών συνεχίζει να εξελίσσεται
  - Αύξηση της χωρητικότητας και της απόδοσης
  - Μείωση κόστους



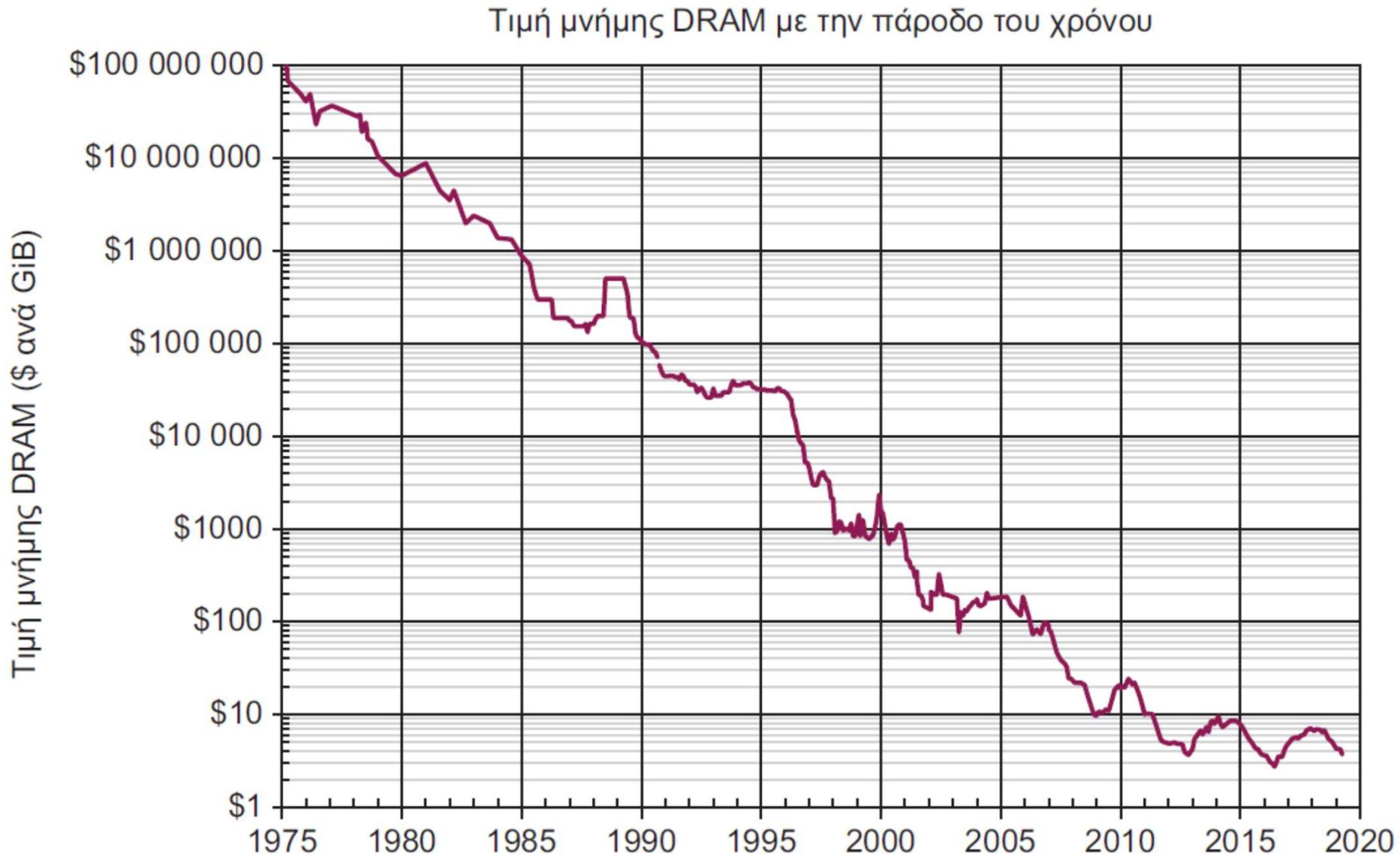
Έτος	Τεχνολογία	Σχέση απόδοσης-κόστους
1951	Λυχνία κενού	1
1965	Τρανζίστορ	35
1975	Ολοκληρωμένο κύκλωμα	900
1995	Πολύ μεγάλης κλίμακας ολοκληρωμένο κύκλωμα (VLSI)	2 400 000
2013	Υπερυψηλής κλίμακας ολοκληρωμένο κύκλωμα (ULSI)	250 000 000 000

# Χωρητικότητα DRAM



- Επί 20 χρόνια: 4x η χωρητικότητα κάθε 3 χρόνια (+60% κάθε χρόνο) – συνολικά > 16000x
- Τα τελευταία χρόνια: περίπου 2x κάθε 2-3 χρόνια

# Κόστος DRAM

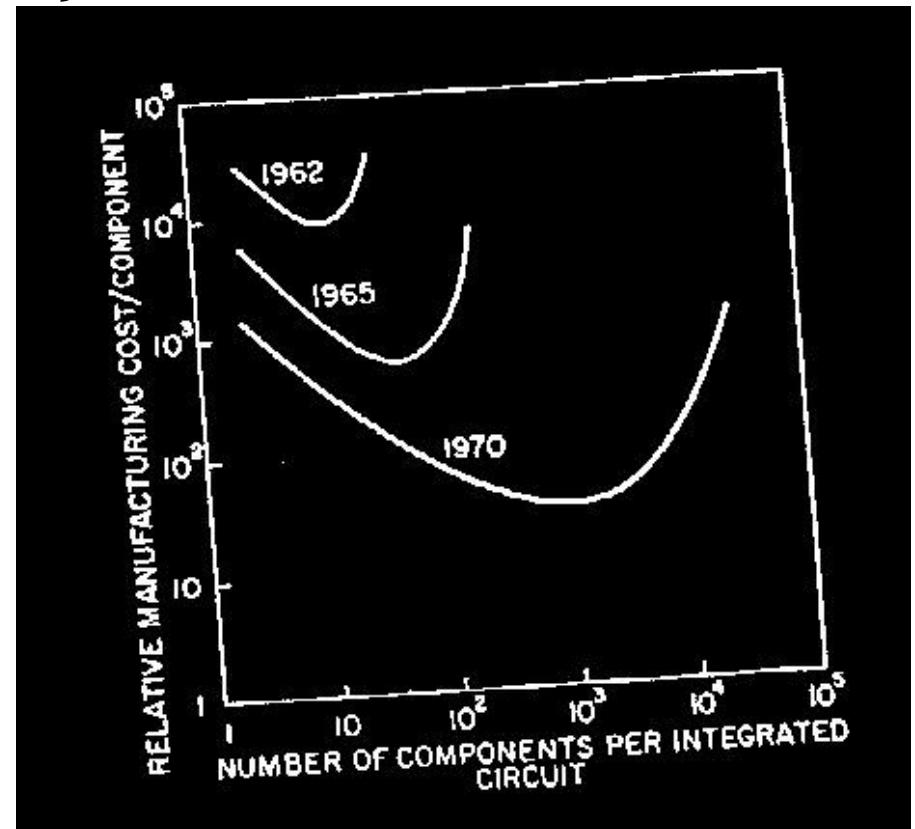


# Τεχνολογία ημιαγωγών

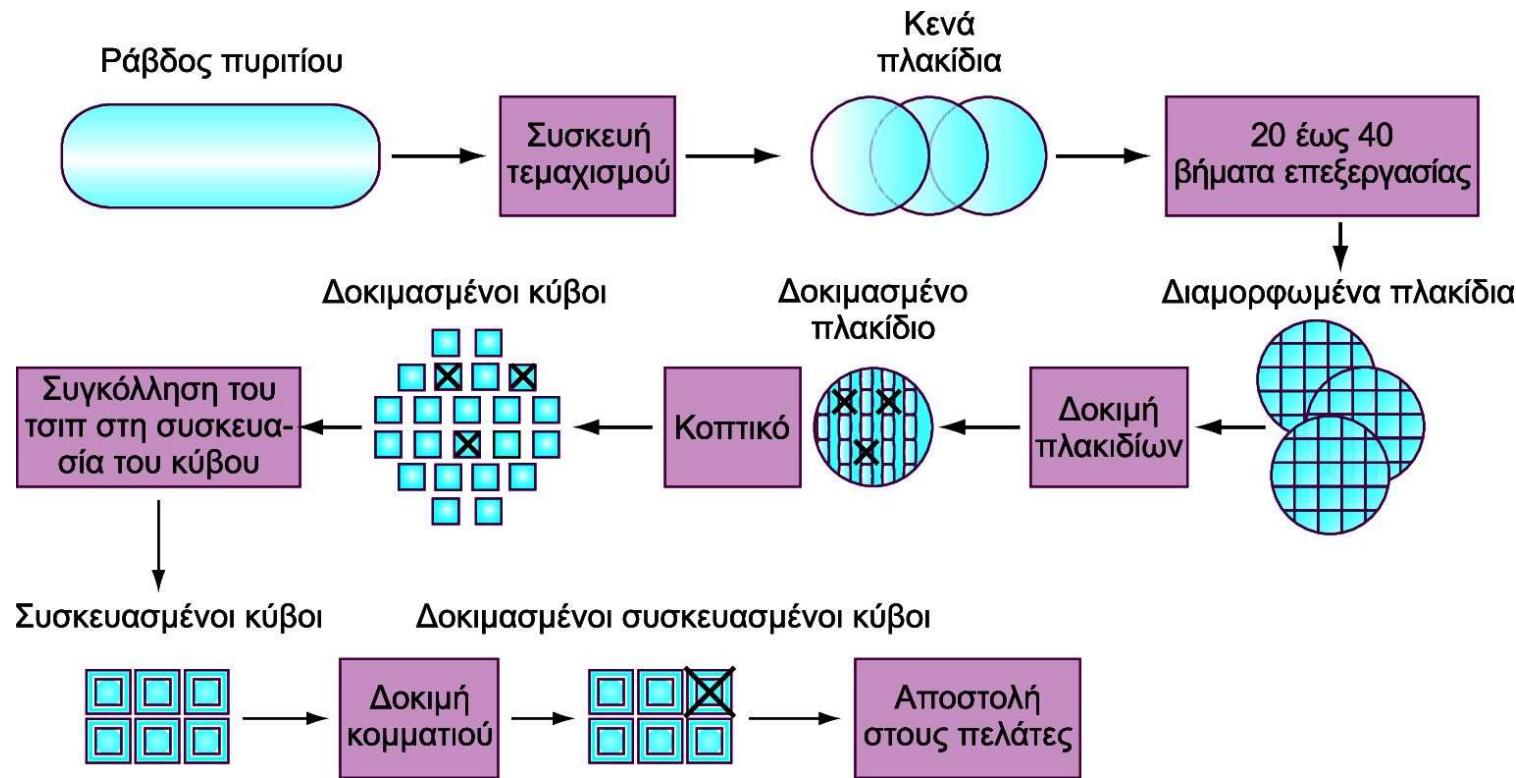
- Πυρίτιο: ημιαγωγός
  - Στοιχεία στη Γη:
    - (1) Οξυγόνο – 49.5%
    - **(2) Πυρίτιο – 28%**
    - (3) Αλουμίνιο – 8%
- Μετασχηματισμός ιδιοτήτων με προσθήκη υλικών:
  - Αγωγοί του ηλεκτρισμού
  - Μονωτές του ηλεκτρισμού
  - Μεταγωγείς (διακόπτες)

# Νόμος του Moore

- Η χωρητικότητα των μικροεπεξεργαστών σε τρανζίστορ θα διπλασιάζεται κάθε 18-24 μήνες.
  - Gordon Moore, συνιδρυτής της Intel, σε δημοσίευσή του τον Απρίλιο 1965 (*Electronics*, vol. 38, number 8)
- Η διατήρηση αυτού του ρυθμού προόδου για σχεδόν 40 χρόνια χρειάστηκε απίστευτες καινοτομίες στις τεχνικές κατασκευής.

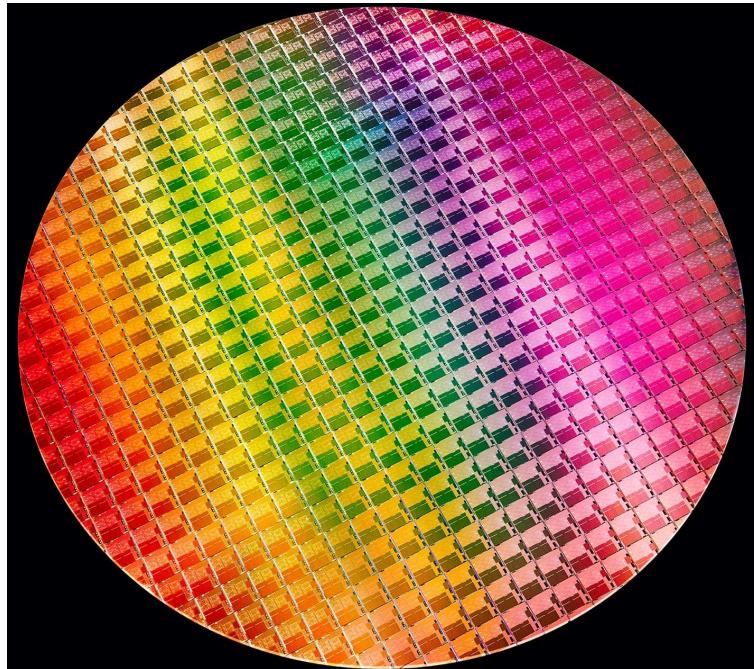


# Παραγωγή ολοκληρωμένων κυκλωμάτων



- Εσοδεία: το ποσοστό των καλών τσιπ στο σύνολο των τσιπ ενός πλακιδίου

# Intel® Core 10<sup>th</sup> Gen



- Πλακίδιο 300mm, 506 τσιπ, τεχνολογία 10 nm
- Κάθε κύβος (τσιπ) έχει διαστάσεις  
 $11.4 \times 10.7$  mm

# Κόστος ολοκληρωμένων κυκλωμάτων

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

$$\text{Κύβοι ανά πλακίδιο} \approx \frac{\text{Επιφάνεια πλακιδίων}}{\text{Επιφάνεια κύβων}}$$

$$\text{Εσοδεία} = \frac{1}{(1 + (\text{Ατέλειες ανά περιοχή} \times \text{Επιφάνεια κύβων}))^n}$$

- Η σχέση με την επιφάνεια και το ποσοστό ατελειών δεν είναι γραμμική
  - Το κόστος και η επιφάνεια των πλακιδίων είναι σταθερά
  - Το ποσοστό ατελειών καθορίζεται από την κατασκευαστική διαδικασία
  - Η επιφάνεια των κύβων καθορίζεται από την αρχιτεκτονική και τη σχεδίαση των κυκλωμάτων

# Ορισμός της απόδοσης

- Ποιο από τα αεροπλάνα έχει την καλύτερη απόδοση;

Αεροπλάνο	Χωρητικότητα επιβατών	Αυτονομία (μίλια)	Ταχύτητα πτήσης (μίλια ανά ώρα)	Διεκπεραιωτική ικανότητα επιβατών (επιβάτες × μίλια ανά ώρα)
Boeing 737	240	3000	564	135 360
BAC/Sud Concorde	132	4000	1350	178 200
Boeing 777-200LR	301	9395	554	166 761
Airbus A380-800	853	8477	587	500 711

# Χρόνος απόκρισης και διεκπεραιωτική ικανότητα

- Χρόνος απόκρισης (response time)
  - Πόσος χρόνος χρειάζεται για να εκτελεστεί μια εργασία
- Διεκπεραιωτική ικανότητα
  - Η συνολική ποσότητα εργασίας που ολοκληρώνεται σε δεδομένο χρόνο
    - π.χ. εργασίες/συναλλαγές/... ανά ώρα
- Πώς επηρεάζονται ο χρόνος απόκρισης και η διεκπεραιωτική ικανότητα από
  - Την αντικατάσταση του επεξεργαστή με μια ταχύτερη έκδοση;
  - Την προσθήκη επιπλέον επεξεργαστών;
- Προς το παρόν θα επικεντρωθούμε στον χρόνο απόκρισης...

# Σχετική απόδοση

- Απόδοση =  $1/\text{χρόνος εκτέλεσης}$
- «ο X είναι  $n$  φορές ταχύτερος από τον Y»

$$\frac{\text{Απόδοση}_X}{\text{Απόδοση}_Y} = \frac{\text{Χρόνος εκτέλεσης}_Y}{\text{Χρόνος εκτέλεσης}_X} = n$$

- Παράδειγμα: χρόνος που χρειάστηκε για την εκτέλεση ενός προγράμματος
  - 10 sec στον A, 15 sec στον B
  - $\text{Χρόνος εκτέλεσης}_B / \text{Χρόνος εκτέλεσης}_A = 15 \text{ sec} / 10 \text{ sec} = 1.5$
  - Άρα ο A είναι 1.5 φορές ταχύτερος από τον B

# Μέτρηση του χρόνου εκτέλεσης

## ■ Παρελθών χρόνος

- Συνολικός χρόνος απόκρισης όλων των πτυχών
  - Επεξεργασία, Ε/Ε, επιβάρυνση του ΛΣ, χρόνος αδράνειας
- Καθορίζει την απόδοση του συστήματος

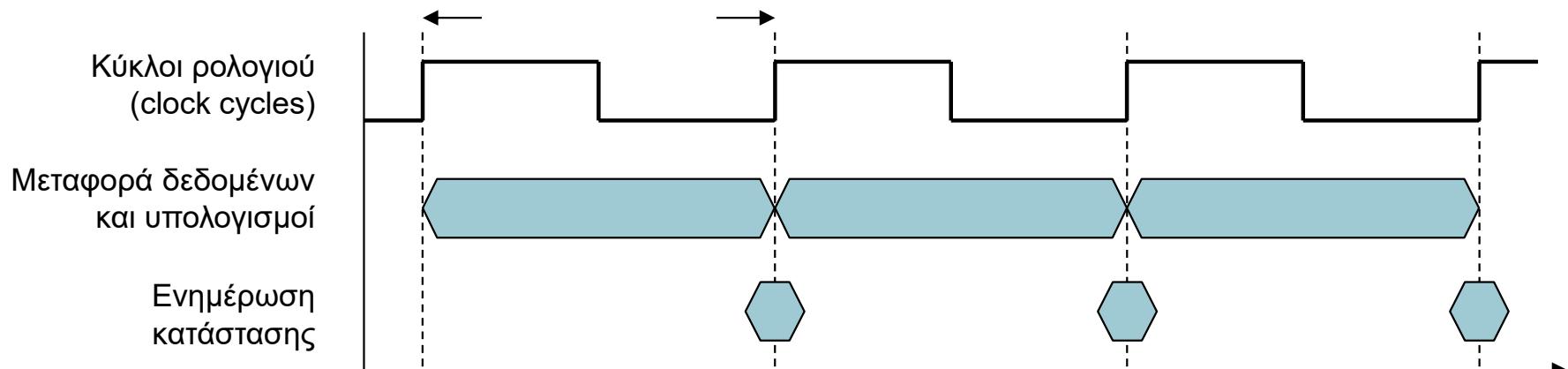
## ■ Χρόνος CPU (CPU time)

- Ο χρόνος που αναλώνεται στην επεξεργασία μιας ορισμένης εργασίας
  - Δεν περιλαμβάνεται ούτε ο χρόνος Ε/Ε ούτε τα χρονομερίδια άλλων εργασιών
- Αποτελείται από τον χρόνο CPU χρήστη (user CPU time) και τον χρόνο CPU συστήματος (system CPU time)
- Η απόδοση της CPU και η απόδοση του συστήματος επηρεάζουν κάθε πρόγραμμα διαφορετικά

# Χρονισμός CPU

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

Περίοδος ρολογιού (clock period)



- Περίοδος ρολογιού: η διάρκεια ενός κύκλου του ρολογιού
  - π.χ.  $250 \text{ ps} = 0.25 \text{ ns} = 250 \times 10^{-12} \text{ sec}$
- Συχνότητα ρολογιού: κύκλοι ανά δευτερόλεπτο
  - π.χ.  $4.0 \text{ GHz} = 4000 \text{ MHz} = 4.0 \times 10^9 \text{ Hz}$

# Χρόνος CPU (CPU time)

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

## ■ Η απόδοση βελτιώνεται

- Με τη μείωση των κύκλων του ρολογιού
- Με την αύξηση της συχνότητας του ρολογιού
- Συχνά ο σχεδιαστής πρέπει να αντισταθμίσει τη συχνότητα του ρολογιού και τον αριθμό κύκλων

# Ένα παράδειγμα για τον χρόνο CPU

- Υπολογιστής A: Ρολόι 2 GHz, χρόνος CPU 10 sec
- Κατά τη σχεδίαση του υπολογιστή B
  - Στόχος είναι ο χρόνος CPU να είναι ίσος με 6 sec
  - Είναι εφικτό το ταχύτερο ρολόι, αλλά οι κύκλοι ρολογιού θα είναι 1.2 φορές περισσότεροι
- Πόσο γρήγορο πρέπει να είναι το ρολόι του υπολογιστή B;

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

$$6 \text{ δευτερόλεπτα} = \frac{1.2 \times 20 \times 10^9 \text{ κύκλοι}}{\text{Ρυθμός ρολογιού}_B}$$

$$\text{Κύκλοι ρολογιού CPU}_A = 10 \text{ δευτερόλεπτα} \times 2 \times 10^9 \frac{\text{κύκλοι}}{\text{δευτερόλεπτο}} = 20 \times 10^9 \text{ κύκλοι}$$

# Ένα παράδειγμα για τον χρόνο CPU

- Πόσο γρήγορο πρέπει να είναι το ρολόι του υπολογιστή B; (συνέχεια)

$$\begin{aligned} \text{Ρυθμός ρολογιού}_B &= \frac{1.2 \times 20 \times 10^9 \text{ κύκλοι}}{6 \text{ δευτερόλεπτα}} = \frac{0.2 \times 20 \times 10^9 \text{ κύκλοι}}{\text{δευτερόλεπτο}} = \\ &= \frac{4 \times 10^9 \text{ κύκλοι}}{\text{δευτερόλεπτο}} = 4 \text{ GHz} \end{aligned}$$

# Πλήθος εντολών και CPI

$$\text{Κύκλοι ρολογιού CPU} = \frac{\text{Εντολές για ένα πρόγραμμα}}{\text{Μέσος αριθμός κύκλων ανά εντολή}}$$

$$\text{Χρόνος CPU} = \text{Πλήθος εντολών} \times \text{CPI} \times \text{Χρόνος κύκλου ρολογιού}$$

$$\text{Χρόνος CPU} = \frac{\text{Πλήθος εντολών} \times \text{CPI}}{\text{Ρυθμός ρολογιού}}$$

- Ο αριθμός των εντολών που απαιτούνται από ένα πρόγραμμα
  - Καθορίζεται από το πρόγραμμα, την ISA, και τον μεταγλωττιστή
- Μέσος αριθμός κύκλων ρολογιού ανά εντολή (clock cycles per instruction, CPI)
  - Καθορίζεται από το υλικό της CPU
  - Αν κάθε εντολή έχει διαφορετικό CPI
    - Η μέση τιμή CPI επηρεάζεται από το μίγμα εντολών

# Ένα παράδειγμα για την τιμή CPI

- Υπολογιστής A: Χρόνος κύκλου = 250 ps, CPI = 2.0
- Υπολογιστής B: Χρόνος κύκλου = 500 ps, CPI = 1.2
- ίδια ISA
- Ποιος είναι ταχύτερος και πόσο;

$$\begin{aligned}\text{Χρόνος } \text{CPU}_A &= \text{Κύκλοι ρολογιού } \text{CPU}_A \times \text{Χρόνος κύκλου ρολογιού} \\ &= I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}\end{aligned}$$

$$\text{Χρόνος } \text{CPU}_B = I \times 1.2 \times 500 \text{ ps} = 600 \times I \text{ ps}$$

Ο Α είναι  
ταχύτερος...

$$\frac{\text{Απόδοση } \text{CPU}_A}{\text{Απόδοση } \text{CPU}_B} = \frac{\text{Χρόνος εκτέλεσης}_B}{\text{Χρόνος εκτέλεσης}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}} = 1.2$$

...κατά τόσο

# Η τιμή CPI πιο αναλυτικά

- Αν για κάθε κατηγορία εντολών χρειάζονται διαφορετικοί αριθμοί κύκλων ρολογιού

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

- Σταθμισμένη μέση τιμή CPI

$$\text{CPI} = \frac{\text{Κύκλοι ρολογιού}}{\text{Πλήθος εντολών}} = \sum_{i=0}^n \left( \text{CPI}_i \times \underbrace{\frac{\text{Πλήθος εντολών}_i}{\text{Πλήθος εντολών}}} \right)$$

Σχετική συχνότητα

# Ένα παράδειγμα για την τιμή CPI

- Εναλλακτικές ακολουθίες μεταγλωτισμένου κώδικα με εντολές των κατηγοριών Α, Β, Γ

Κατηγορία	A	B	Γ
CPI για κάθε κατηγορία	1	2	3
Πλήθος εντολών για την ακολουθία 1	2	1	2
Πλήθος εντολών για την ακολουθία 2	4	1	1

- Ακολουθία 1: Αριθμός εντολών (IC) = 5

- Κύκλοι ρολογιού  
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$   
 $= 10$
- Μέσος CPI =  $10/5 = 2.0$

- Ακολουθία 2: Αριθμός εντολών (IC) = 6

- Κύκλοι ρολογιού  
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$   
 $= 9$
- Μέσος CPI =  $9/6 = 1.5$

# Απόδοση: Σύνοψη

## η ΓΕΝΙΚΗ εικόνα

$$\text{Χρόνος} = \frac{\Delta\text{ευτερόλεπτα}}{\text{Πρόγραμμα}} = \frac{\text{Εντολές}}{\text{Πρόγραμμα}} \times \frac{\text{Κύκλοι}}{\text{Εντολή}} \times \frac{\rho\text{ολογιού}}{\text{Κύκλος}} \times \frac{\Delta\text{ευτερόλεπτα}}{\rho\text{ολογιού}}$$

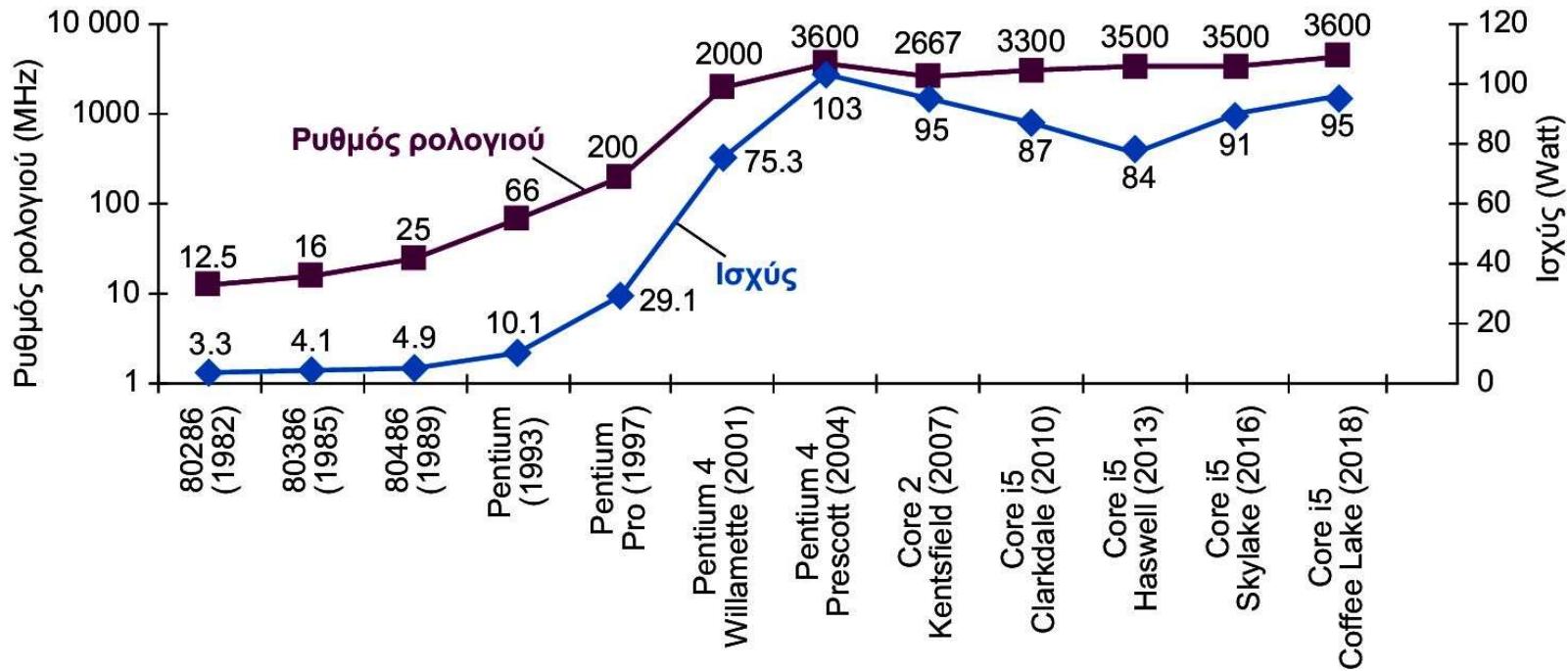
### ■ Η απόδοση εξαρτάται από:

- Τον αλγόριθμο: επηρεάζει τον αριθμό εντολών, πιθανώς το CPI
- Τη γλώσσα προγραμματισμού: επηρεάζει τον αριθμό εντολών, το CPI
- Τον μεταγλωττιστή: επηρεάζει τον αριθμό εντολών, το CPI
- Την αρχιτεκτονική συνόλου εντολών: επηρεάζει τον αριθμό εντολών, το CPI, το  $T_c$

# CPI και IPC

- Κάποιοι επεξεργαστές προσκομίζουν και εκτελούν πολλές εντολές ανά κύκλο ρολογιού
- Τότε το CPI γίνεται μικρότερο από το 1
- Χρησιμοποιείται και το IPC (instructions per cycle)
  - π.χ. αν εκτελεί 4 εντολές ανά κύκλο έχει  $IPC=4$  και  $CPI=0.25$

# Τάσεις της ισχύος



- Στην τεχνολογία ολοκληρωμένων κυκλωμάτων CMOS

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

$\times 30$

$5V \rightarrow 1V$

$\times 1000$

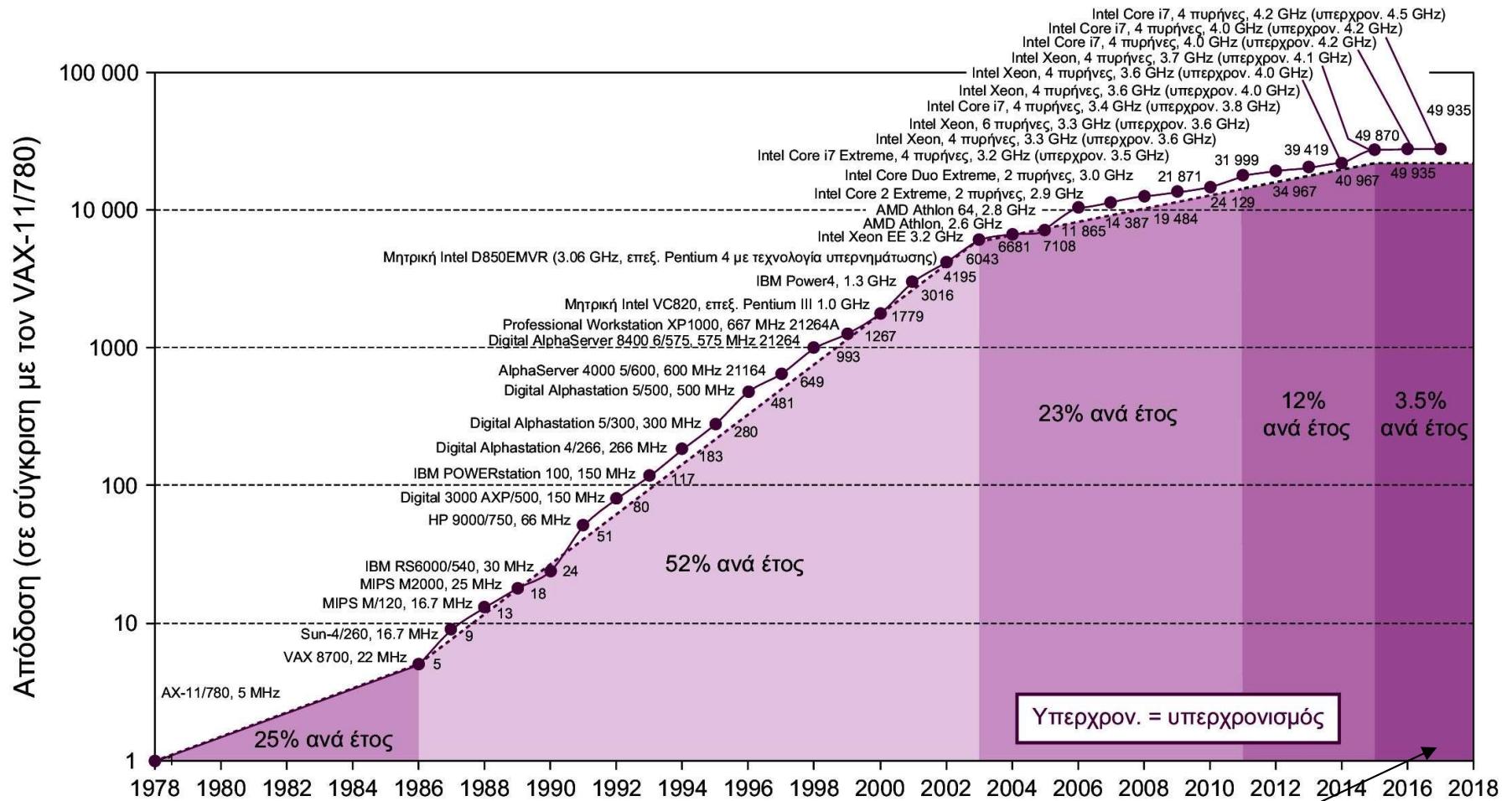
# Μείωση της ισχύος

- Έστω ότι μια νέα CPU έχει
  - το 85% του φορτίου χωρητικότητας του παλαιότερου επεξεργαστή
  - μείωση 15% στην τάση και 15% στη συχνότητα

$$\frac{\text{Ισχύς}_{\text{νέου}}}{\text{Ισχύς}_{\text{παλιού}}} = \frac{\left( \frac{\text{Φορτίο}}{\text{χωρητικότητας}} \times 0.85 \right) \times \left( \text{Tάση} \times 0.85 \right)^2 \times \left( \frac{\text{Συχνότητα}}{\text{μεταγωγής}} \times 0.85 \right)}{\text{Φορτίο χωρητικότητας} \times \text{Tάση}^2 \times \text{Συχνότητα μεταγωγής}}$$
$$= 0.85^4 = 0.52$$

- Το «τείχος της ισχύος»
  - Δεν μπορούμε να μειώσουμε περαιτέρω την τάση
  - Δεν μπορούμε να απομακρύνουμε περισσότερη θερμότητα
- Με ποιον άλλο τρόπο μπορούμε να βελτιώσουμε την απόδοση;

# Απόδοση μονοεπεξεργαστών



Περιορισμός από την ισχύ, την παραλληλία επιπέδου εντολών, και τον λανθάνοντα χρόνο της μνήμης

# Πολυεπεξεργαστές

- Πολυπύρηνοι (multicore) μικροεπεξεργαστές
  - Περισσότεροι του ενός επεξεργαστές ανά τσιπ
- Απαιτείται ρητά παράλληλος προγραμματισμός
  - Συγκριτικά με την παραλληλία επιπέδου εντολών
    - Το υλικό εκτελεί περισσότερες από μία εντολές ταυτόχρονα
    - Χωρίς να είναι ορατό από τον προγραμματιστή
  - Δύσκολη η εφαρμογή
    - Δημιουργία κώδικα με γνώμονα την απόδοση
    - Εξισορρόπηση φορτίου
    - Βελτιστοποίηση επικοινωνίας και συγχρονισμού

# Μορφές παραλληλίας

- Παραλληλία επιπέδου εντολής (Instruction Level Parallelism – ILP)
  - Παράλληλη εκτέλεση εντολών του ίδιου προγράμματος
- Παραλληλία επιπέδου δεδομένων (Data Level Parallelism – DLP)
  - Παράλληλη εκτέλεση της ίδιας λειτουργίας σε πολλά διαφορετικά δεδομένα
- Παραλληλία επιπέδου νήματος (Thread Level Parallelism – TLP)
  - Παράλληλη εκτέλεση πολλών νημάτων εκτέλεσης (execution threads)
  - Στον ίδιο πυρήνα ή σε διαφορετικούς πυρήνες

# Μετροπρογράμματα SPEC CPU

- Προγράμματα που χρησιμοποιούνται για τη μέτρηση της απόδοσης
  - Θεωρητικά αντιπροσωπευτική μέτρηση του πραγματικού φόρτου εργασίας
- Οργανισμός Αξιολόγησης Απόδοσης Συστημάτων (Standard Performance Evaluation Corp, SPEC)
  - Καταρτίζει δοκιμές και προγράμματα για τη μέτρηση της απόδοσης της CPU, της Ε/Ε, του Ιστού, ...
- SPEC CPU2006
  - Ο χρόνος που έχει παρέλθει για την εκτέλεση ορισμένων επιλεγμένων προγραμμάτων
    - Αμελητέα Ε/Ε, έμφαση στην απόδοση της CPU
  - Κανονικοποίηση σε σχέση με τον υπολογιστή αναφοράς
  - Συνοψίζεται στον γεωμετρικό μέσο των δεικτών απόδοσης
    - CINT2006 (μετροπρογράμματα ακεραίων) και CFP2006 (μετροπρογράμματα κινητής υποδιαστολής)

<http://www.spec.org>

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

# Τα μετροπρογράμματα SPECspeed 2017 Integer όταν εκτελούνται σε έναν Intel Xeon E5-2650L στα 1.8 GHz

Περιγραφή	Όνομα	Πλήθος εντολών $\times 10^9$	CPI	Χρόνος κύκλου ρολογιού ( $sec \times 10^{-9}$ )	Χρόνος εκτέλεσης (sec)	Χρόνος αναφοράς (sec)	SPECratio
Διερμηνευτής Perl	perlbench	2684	0.42	0.556	627	1774	2.83
Μεταγλωττιστής C της GNU	gcc	2322	0.67	0.556	863	3976	4.61
Σχεδίαση διαδρομής (route planning)	mcf	1786	1.22	0.556	1215	4721	3.89
Προσομοίωση διακριτών συμβάντων – δίκτυο υπολογιστών	omnetpp	1107	0.82	0.556	507	1630	3.21
Μετατροπή XML σε HTML μέσω XSLT	xalancbmk	1314	0.75	0.556	549	1417	2.58
Συμπίεση βίντεο	x264	4488	0.32	0.556	813	1763	2.17
Τεχνητή νοημοσύνη: Αναζήτηση δέντρου alpha-beta (παιχνίδι σκακιού)	deepsjeng	2216	0.57	0.556	698	1432	2.05
Τεχνητή νοημοσύνη: Αναζήτηση δέντρου MonteCarlo (παιχνίδι Go)	leela	2236	0.79	0.556	987	1703	1.73
Τεχνητή νοημοσύνη: Γεννήτρια αναδρομικής λύσης (παιχνίδι Sudoku)	exchange2	6683	0.46	0.556	1718	2939	1.71
Γενική συμπίεση δεδομένων	xz	8533	1.32	0.556	6290	6182	0.98
Γεωμετρικός μέσος							2.36

# Εξήγηση πίνακα

Περιγραφή	Όνομα	Πλήθος εντολών $\times 10^9$	CPI	Χρόνος κύκλου ρολογιού (sec $\times 10^{-9}$ )	Χρόνος εκτέλεσης (sec)	Χρόνος αναφοράς (sec)	SPECratio
Διερμηνευτής Perl	perlbench	2684	0.42	0.556	627	1774	2.83
Μεταγλωττιστής C της GNU	gcc	2322	0.67	0.556	863	3976	4.61
Τεχνητή νοημοσύνη: Γεννήτρια αναδρομικής λύσης (παιχνίδι Sudoku)	exchange2	6683	0.46	0.556	1718	2939	1.71
Γενική συμπίεση δεδομένων	xz	8533	1.32	0.556	6290	6182	0.98
Γεωμετρικός μέσος							2.36

Χρόνος εκτέλεσης =  
 $IC \times 10^9 \times CPI \times T_c$

Χρ. εκτέλεσης στον υπολογιστή  
αναφοράς

Χρ.αναφοράς / Χρ. εκτέλεσης

# Μετροπρόγραμμα SPEC Power

- Κατανάλωση ισχύος διακομιστών σε διαφορετικά επίπεδα φορτίου εργασίας
  - Απόδοση: ssj\_ops/sec
  - Ισχύς: Watt (Joule/sec)

$$\text{ολικό ssj\_ops ανά watt} = \left( \sum_{i=0}^{10} \text{ssj\_ops}_i \right) \Bigg/ \left( \sum_{i=0}^{10} \text{power}_i \right)$$

# Το μετροπρόγραμμα SPECpower\_ssj2008 στον Χεον Ε5-2650L

Στόχος φορτίου %	Απόδοση (ssj_ops)	Μέση ισχύς (Watt)
100%	4 864 136	347
90%	4 389 196	312
80%	3 905 724	278
70%	3 418 737	241
60%	2 925 811	212
50%	2 439 017	183
40%	1 951 394	160
30%	1 461 411	141
20%	974 045	128
10%	485 973	115
0%	0	48
Σύνολο	26 815 444	2 165
$\sum \text{ssj\_ops} / \sum \text{power} =$		12 385

# Πλάνες και παγίδες

- Το βιβλίο περιέχει ειδικές ενότητες
- Πλάνες: συνηθισμένες παρανοήσεις (νομίζουμε ότι ισχύει κάτι αλλά δεν ισχύει)
  - Δίνονται αντιπαραδείγματα
- Παγίδες: λάθη που γίνονται εύκολα
  - Λανθασμένα γενικεύουμε κάποια αρχή η οποία όμως είναι αληθής μόνο σε περιορισμένο πλαίσιο

# Παγίδα: Ο νόμος του Amdahl

- Η γνώμη ότι η βελτίωση μιας πτυχής ενός υπολογιστή θα αυξήσει την απόδοση κατά ποσότητα ανάλογη του μεγέθους της βελτίωσης

Χρόνος εκτέλεσης μετά τη βελτίωση =

$$\frac{\text{Χρόνος εκτέλεσης που επηρεάζεται από τη βελτίωση}}{\text{Ποσότητα βελτίωσης}} + \frac{\text{Χρόνος εκτέλεσης που δεν επηρεάζεται}}$$

- Παράδειγμα: στον πολλαπλασιασμό αναλώνονται 80 από τα 100 sec

- Πόσο πρέπει να βελτιωθεί η απόδοση στον πολλαπλασιασμό ώστε το πρόγραμμα να εκτελείται 5× πιο γρήγορα;

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

- Άρα: Επιτάχυνση της πιο κοινής (συνηθισμένης) περίπτωσης

# Πλάνη: Χαμηλή κατανάλωση ισχύος όταν ο υπολογιστής είναι αδρανής

- Ας κοιτάξουμε πάλι το αποτέλεσμα του μετροπρογράμματος για την ισχύ του i7
  - Σε φορτίο 100%: 258W
  - Σε φορτίο 50%: 170W (66%)
  - Σε φορτίο 10%: 121W (47%)
- Τα κέντρο δεδομένων της Google
  - Λειτουργούν ως επί το πλείστον στο 10% – 50%
  - Λειτουργούν στο 100% για λιγότερο από το 1% του χρόνου
- Σχεδίαση των επεξεργαστών ώστε η ισχύς που χρησιμοποιούν να είναι ανάλογη του φορτίου

# Παγίδα: Η παράμετρος MIPS ως μέτρο της απόδοσης

- MIPS: εκατομμύρια εντολές ανά δευτερόλεπτο (million instructions per second)
  - Δεν λαμβάνονται υπόψη
    - Διαφορές στην ISA των υπολογιστών
    - Διαφορές στην πολυπλοκότητα των εντολών

$$\text{MIPS} = \frac{\text{Πλήθος εντολών}}{\frac{\text{Πλήθος εντολών} \times \text{CPI}}{\text{Ρυθμός ρολογιού}} \times 10^6} = \frac{\text{Ρυθμός ρολογιού}}{\text{CPI} \times 10^6}$$

- Η τιμή CPI διαφέρει για κάθε πρόγραμμα σε μια ορισμένη CPU

# Συμπερασματικές παρατηρήσεις

- Η σχέση κόστους-απόδοσης βελτιώνεται
  - Λόγω της ανάπτυξης της υποκείμενης τεχνολογίας
- Ιεραρχία στρωμάτων αφαίρεσης
  - Τόσο στο υλικό όσο και στο λογισμικό
- Αρχιτεκτονική συνόλου εντολών
  - Διασύνδεση υλικού/λογισμικού
- Χρόνος εκτέλεσης: το καλύτερο μέτρο για την απόδοση
- Η ισχύς είναι περιοριστικός παράγοντας
  - Βελτίωση της απόδοσης με χρήση παραλληλίας