ΕΠΛ221: Οργάνωση Υπολογιστών

Γιάννος Σαζεϊδης

Κοινή λογική 1: Ίδια δουλειά τελειώνει στον ίδιο χρόνο

(A)

```
for(i=0;i<SIZE;i++)
    for(j=0;j<SIZE;j++)
      s += a[i][j];
```

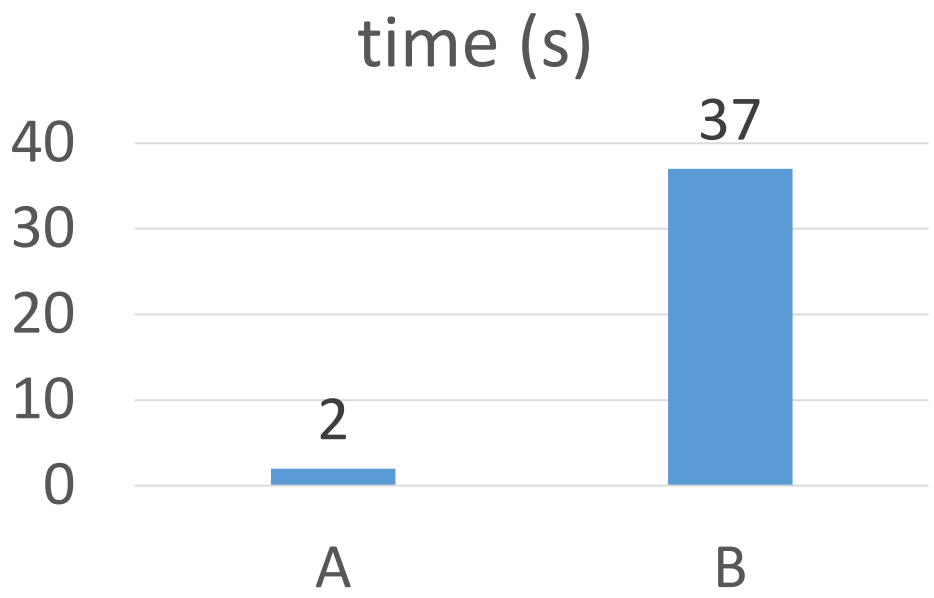
(B)

```
for(j=0;j<SIZE;j++)
    for(i=0;i<SIZE;i++)
      s += a[i][j];
```

Ποιο είναι πιο γρήγορο;

a[SIZE][SIZE] δεδομένα που επεξεργάζεται το πρόγραμμα

SIZE=32000



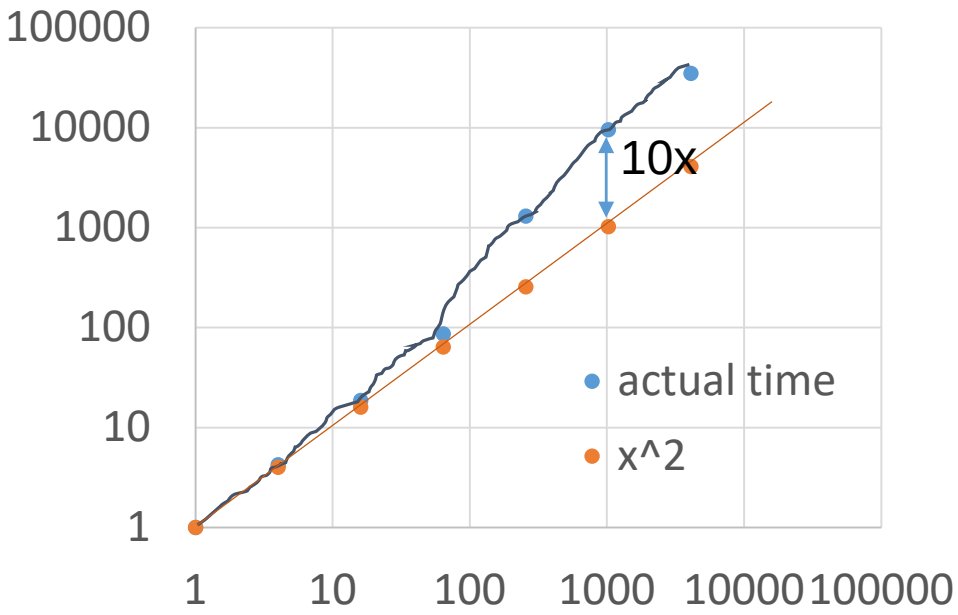
Κοινή λογική 2: x φορές περισσότερη δουλειά χρειάζεται x φορές περισσότερο χρόνο

(B)

```
for(j=0;j<SIZE;j++)
    for(i=0;i<SIZE;i++)
      s += a[i][j];
```

Εάν το SIZE αυξηθεί κατά x ο αριθμός των πράξεων αυξάνεται x^2. Ο χρόνος;

Πως μεγαλώνει ο χρόνος με μεγαλύτερο πρόβλημα



Κοινή λογική 3: Πιο λίγη δουλειά τελειώνει πιο γρήγορα

(A)

```
for(i=0;i<SIZE;i++)
   if (a[i]>1000000000)
     s++;
   else if (a[i] > 500000000)
     n++;
   else
     p++;
 }
printf("%d %d %d\n",s,n,p);
```

a[SIZE] πίνακας με μέγεθός 16000000 ακεραίων (16εκ.)

(A)

```
for(i=0;i<SIZE;i++)
    if (a[i]>1000000000)
      s++;
    else if (a[i] > 500000000)
      n++;
    else
      p++;
 }
printf("%d %d %d\n",s,n,p);
```

8.6e9 3.4e9 4.0e9

1 x 8.6 + 2 x 3.4 + 3 x 4.0 = 27.4

(A)

```
for(i=0;i<SIZE;i++)
    if (a[i]>1000000000)
      s++;
    else if (a[i] > 500000000)
      n++;
    else
      p++;
 }
printf("%d %d %d\n",s,n,p);
```

8.6e9 3.4e9 4.0e9

1 x 8.6 + 2 x 3.4 + 3 x 4.0 = 27.4

(B)

```
for(i=0;i<SIZE;i++)
    if (a[i]==0)
      s++;
    else if (a[i]==1)
      n++;
    else
      p++;
 }
 printf("%d %d %d\n",s,n,p);
```

0 0 16.0e9

3 x 16.0 = 48

(A)

```
for(i=0;i<SIZE;i++)
    if (a[i]>1000000000)
      s++;
    else if (a[i] > 500000000)
      n++;
    else
      p++;
 }
printf("%d %d %d\n",s,n,p);
```

8.6e9 3.4e9 4.0e9

1 x 8.6 + 2 x 3.4 + 3 x 4.0 = 27.4

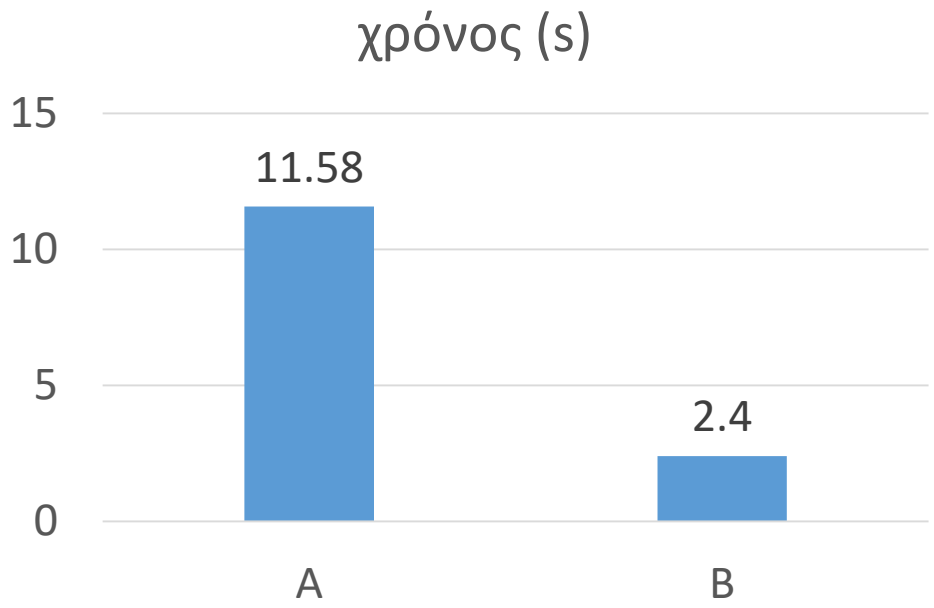
(B)

```
for(i=0;i<SIZE;i++)
    if (a[i]==0)
      s++;
    else if (a[i]==1)
      n++;
    else
      p++;
 }
 printf("%d %d %d\n",s,n,p);
```

0 0 16.0e9

16.0 * 3 = 48

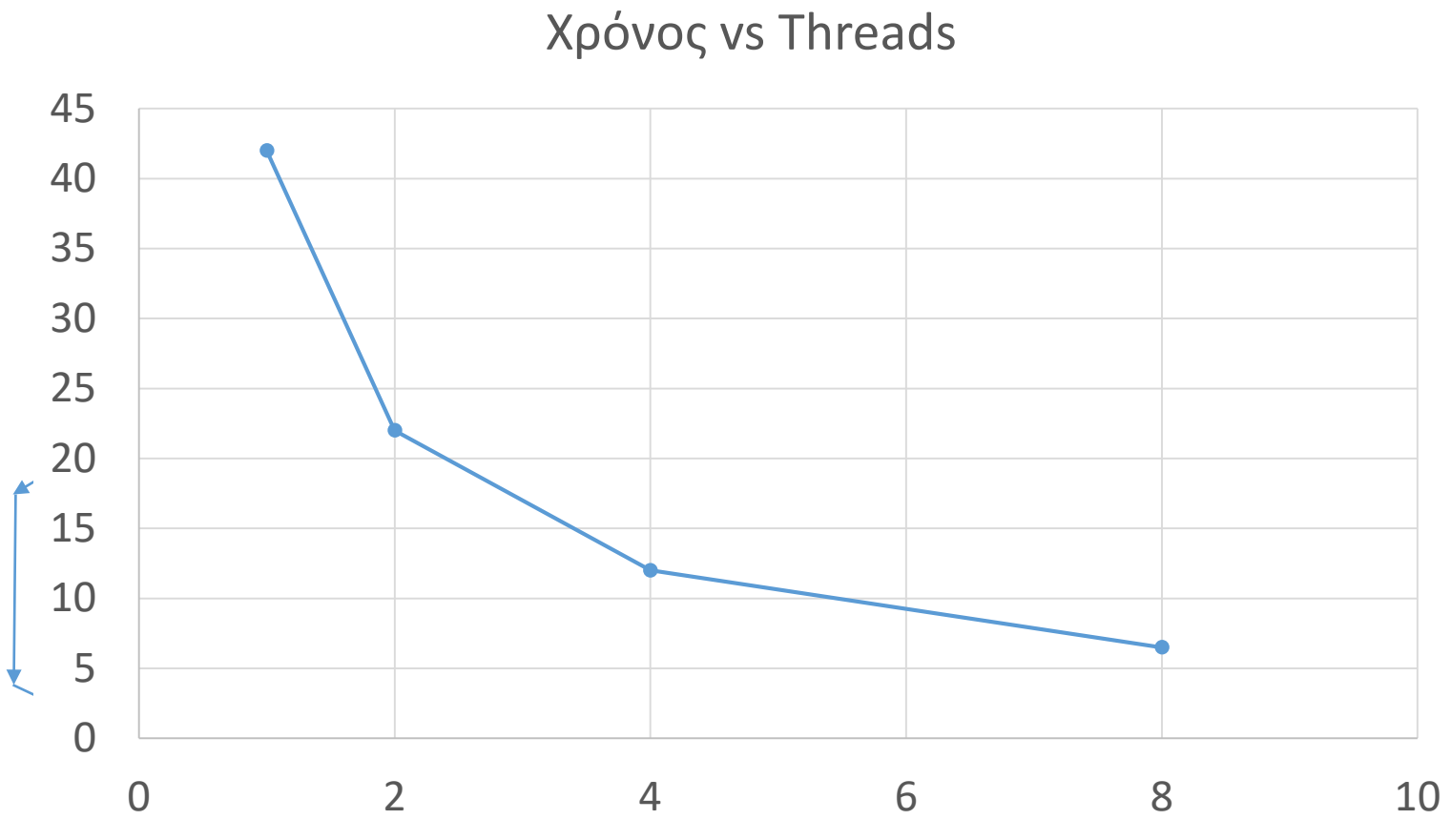
48/27.4 = 1.75 περισσότερες συγκρίσεις το B


χρόνος (s)
15
10
5
0
11.58
2.4
A
B

Παράλληλη vs Σειριακή Επεξεργασία

```
void countf(int *a, int key, int begin, int end,int *count){
        int i;
        for(i=begin;i<end;i++)
                if (a[i]==key)
                        ++(*count);
}
```

```
void countf(int *a, int key, int begin, int end,int *count){
        int i;
```




```
}
```

Στόχοι Οργάνωσης Υπολογιστών (221), Αρχιτεκτονικής Υπολογιστών (420) και Παράλληλης Επεξεργασία (325)

- Να γράφετε προγράμματα με καλύτερη απόδοση
- Να αναλύετε που πάει ο χρόνος κατά την διάρκεια της εκτέλεσης ενός προγράμματος
- Να μάθετε βασικές και προχωρημένες τεχνικές που χρησιμοποιούνται για βελτίωση της επίδοσης των μοντέρνων υπολογιστικών συστημάτων και πως αλληλοεπιδρούν με το λογισμικό και τα δεδομένα
  - Pipelining, prediction, caches, speculation, hyperthreading, multi-cores, prefetching, multi channel DRAM, Flash/SSD, turbo-mode, DVFS, accelerators (SIMD, GPUs, Deep-Neural-Nets)
- Να γνωρίσετε την αρχιτεκτονική διαφόρων μοντέρνων υπολογιστικών συστημάτων
  - Sensors, tablets, smartphones, laptops, servers, data centers
- Παράλληλη Επεξεργασία και Προγραμματισμός
- Τις τάσεις τεχνολογίας υπολογιστών
  - Περιορισμοί και δυνατότητες

Τι θα μάθετε στο 221:

- Προγραμματισμός σε συμβολικό επίπεδο
- Βασικές Τεχνικές Υλικού για βελτίωση της επίδοσης ενός προγράμματος
  - Διασωλήνωση και Ιεραρχία Μνήμης (Pipelining and Memory Hierarchy)
- Σχεδιασμός απλού επεξεργαστή με διασωλήνωση
- Μέτρηση και σύγκριση της επίδοσης υπολογιστών
- Βασικές Έννοιες Παραλληλισμού
  - SIMD, ILP, DLP, TLP, RLP
  - Superscalar processors, simd, multicores, accelerators (GPU)

Πληροφορίες για το ΕΠΛ221

- Διδάσκων: Γιάννος Σαζεϊδης
- Διαλέξεις: ΧΩΔ02-9 4.30-6 Δευτέρα και Πέμπτη, και
  ΧΩΔ02-14 12-1 Τετάρτη
- Βοηθός Διδασκαλίας: Πέτρος Παναγή
- www.cs.ucy.ac.cy/courses/E_PL221

- Προαπαιτούμενα
  - Ψηφιακά Συστήματα (ΕΠΛ121)

- Βιβλιογραφία: Computer Organization and Design, Henessy & Patterson 4th edition

Πληροφορίες για ΕΠΛ221

- Εργαστήριο:
  - εισαγωγή εργαλείων
  - εμπέδωση εννοιών
  - αξιολόγηση εργασιών, Quiz
- Αξιολόγηση
  - Εργασίες, Quiz – 25-30 %
    - 6-8
  - Παρουσίαση (ομάδα 2 ατόμων) - 5%
  - Ενδιάμεση – 20%
  - Τελική – 45-50%
  - Για επιτυχία στο μάθημα

    Συνολικός Βαθμός $\geq$ 50%

    (Ενδιάμεση * 0.2 + Τελική * 0.5) / 0.7 $\geq$ 50%

Με την επιτυχή ολοκλήρωση του 221:

Καλύτερη Κατανόηση

- Τάσεων Τεχνολογίας Υπολογιστών
- Πως γίνεται η διασύνδεση μεταξύ λογισμικού και υλικού
- Τι επηρεάζει την επίδοση ενός προγράμματος
  - Λογισμικό και Υλικό

ΕΠΛ221: Οργάνωση Υπολογιστών

Γιάννος Σαζεϊδης

Κεφ. 1: Computer Abstractions and Technology

Υπολογιστής: Αφαιρετικότητα και Τεχνολογία

Ολο το κεφ. Εκτός από 1.4,1.5

1η Εργασία

Θα ανακοινωθεί αύριο 4/9.

Ημερομηνία παράδοσης 10/9 μεσημέρι.

The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore's Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
  - IoT (Internet of Things)
- Computers are pervasive

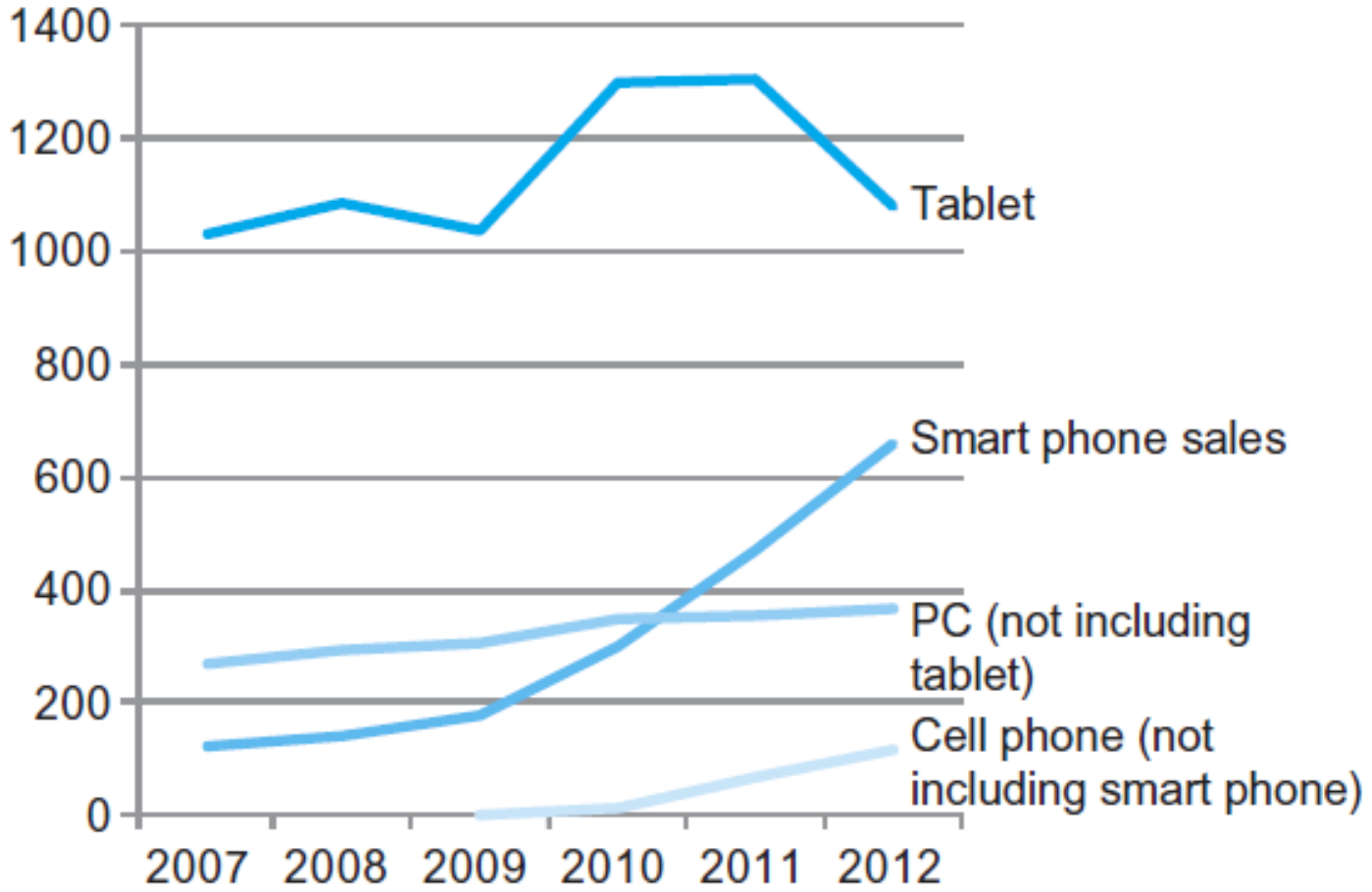
Classes of Computers

- Personal computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized

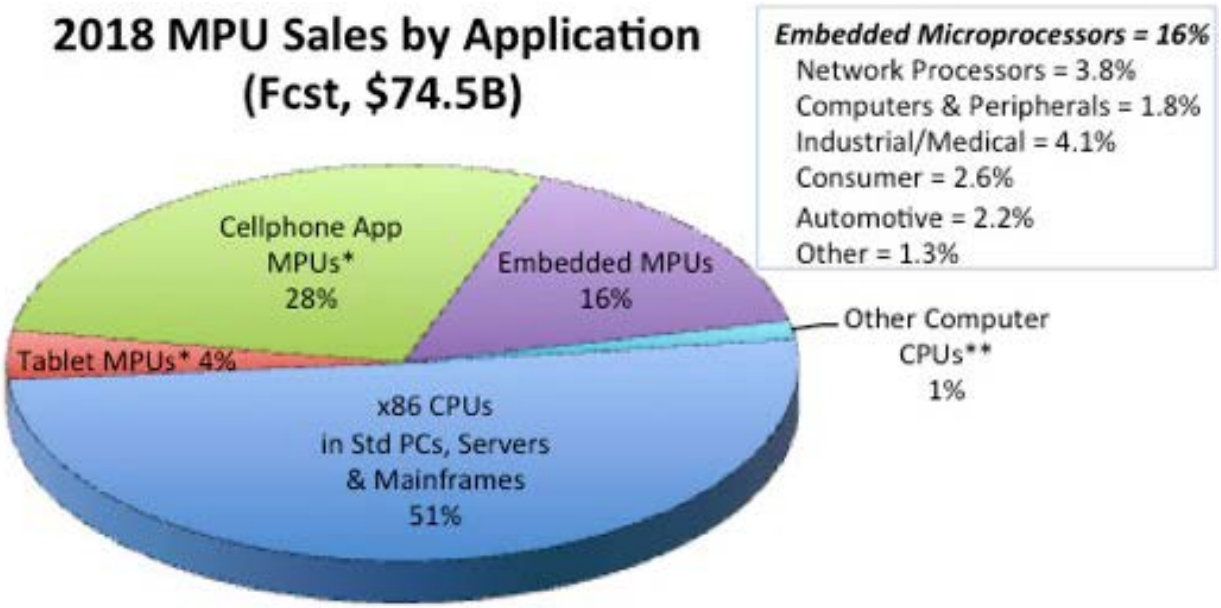
Classes of Computers

- Supercomputers
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

The PostPC Era



Market Landscape



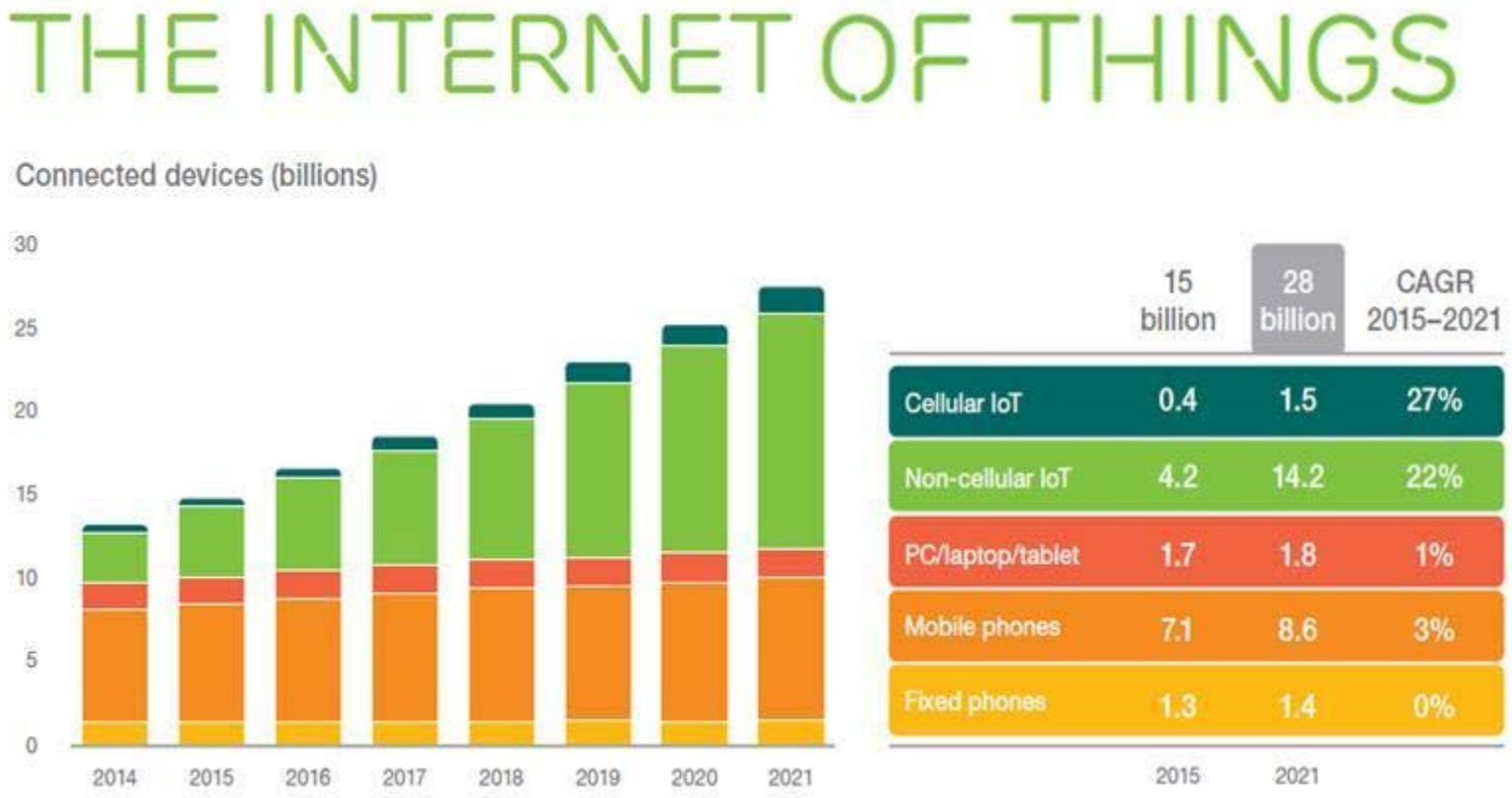
*Includes ARM-based and x86 processors. **Includes ARM-based and other RISC processors.

Source: IC Insights

The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- Cloud computing
  - Warehouse Scale Computers (WSC)
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion run in the Cloud
  - Amazon, Google, Microsoft

The IoT Era



Many sensory devices that collect (pre-process) and transmit data (directly or through other devices) to data centers
Data centers servers analyze (data analytics) and take decisions
E.g. Autonomous Cars

What You Will Learn in 221

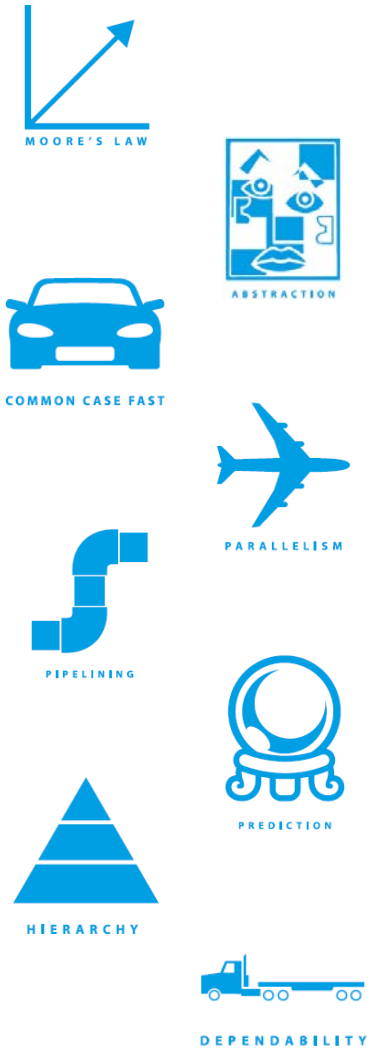
- The hardware/software interface
- What determines program performance
  - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

Understanding Performance

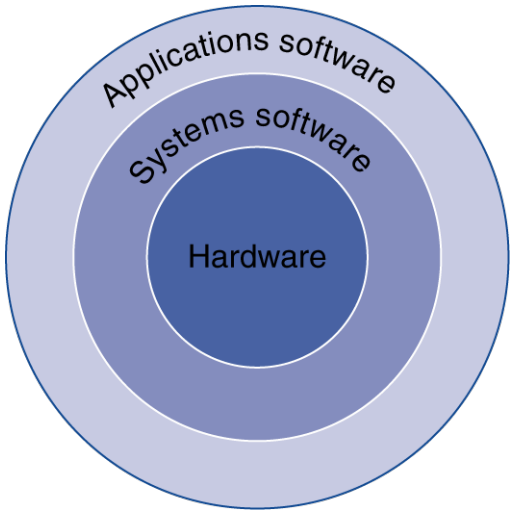
- Algorithm
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation
- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

Eight Great Ideas

- Design for ***Moore's Law***
- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** *via* redundancy



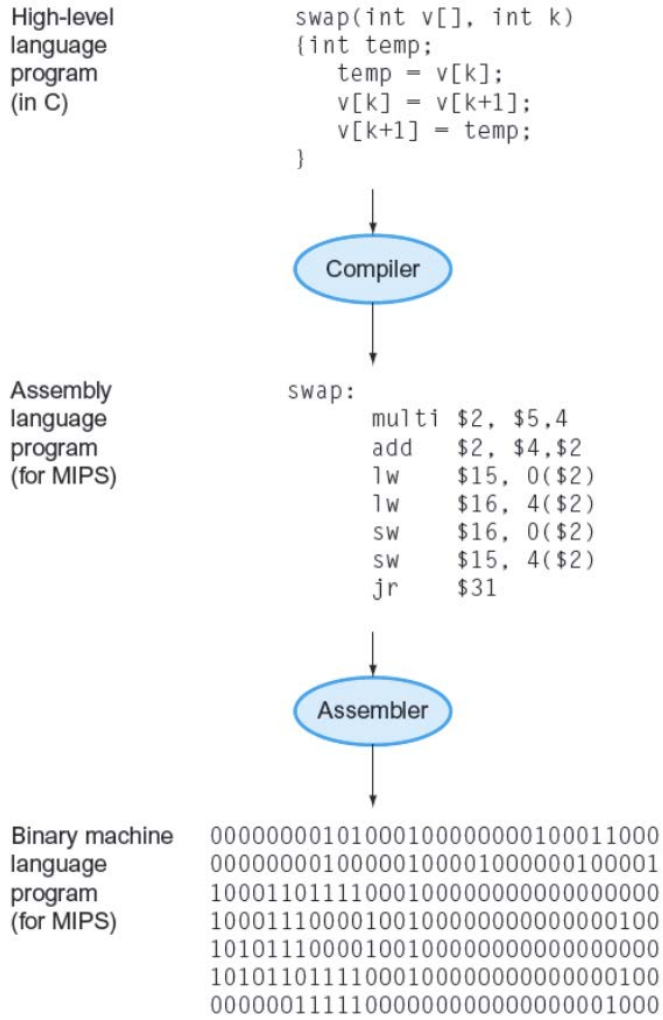
Below Your Program



- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

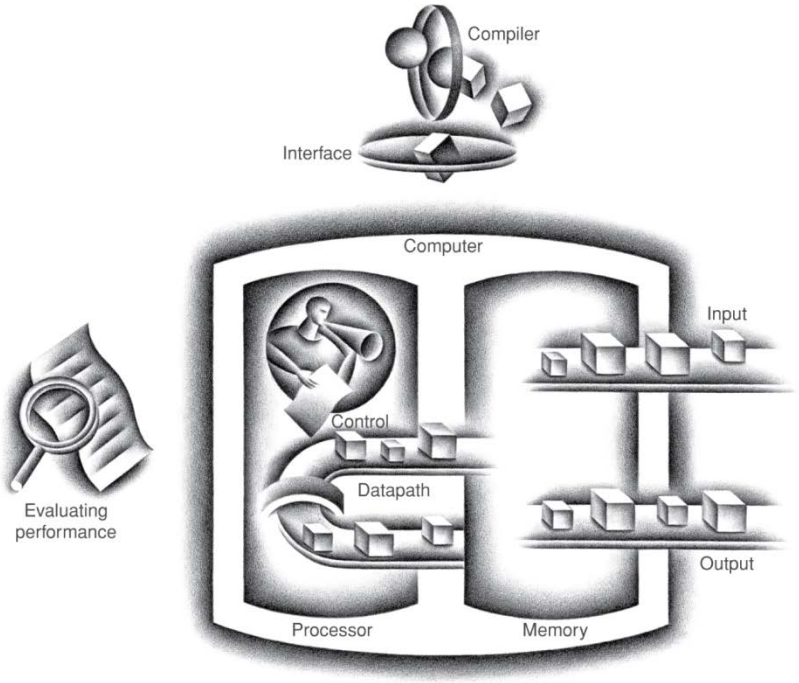
Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



Components of a Computer

The BIG Picture



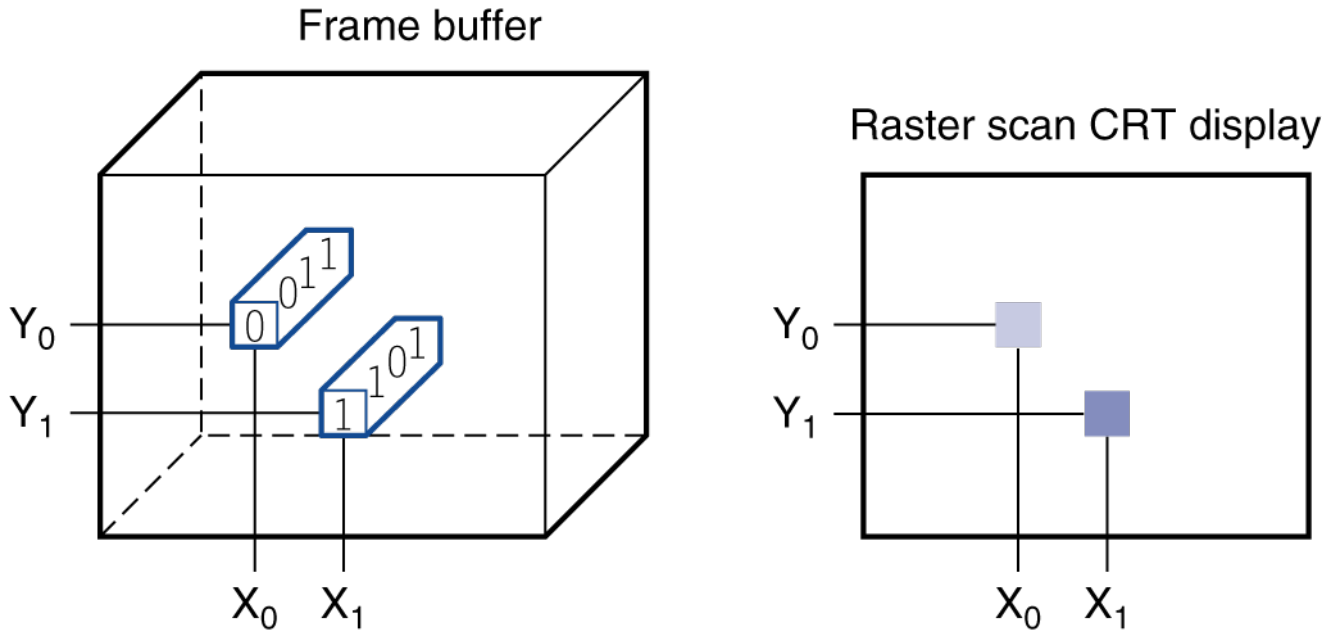
- Same components for all kinds of computer
  - Desktop, server, embedded, PMD
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

Touchscreen

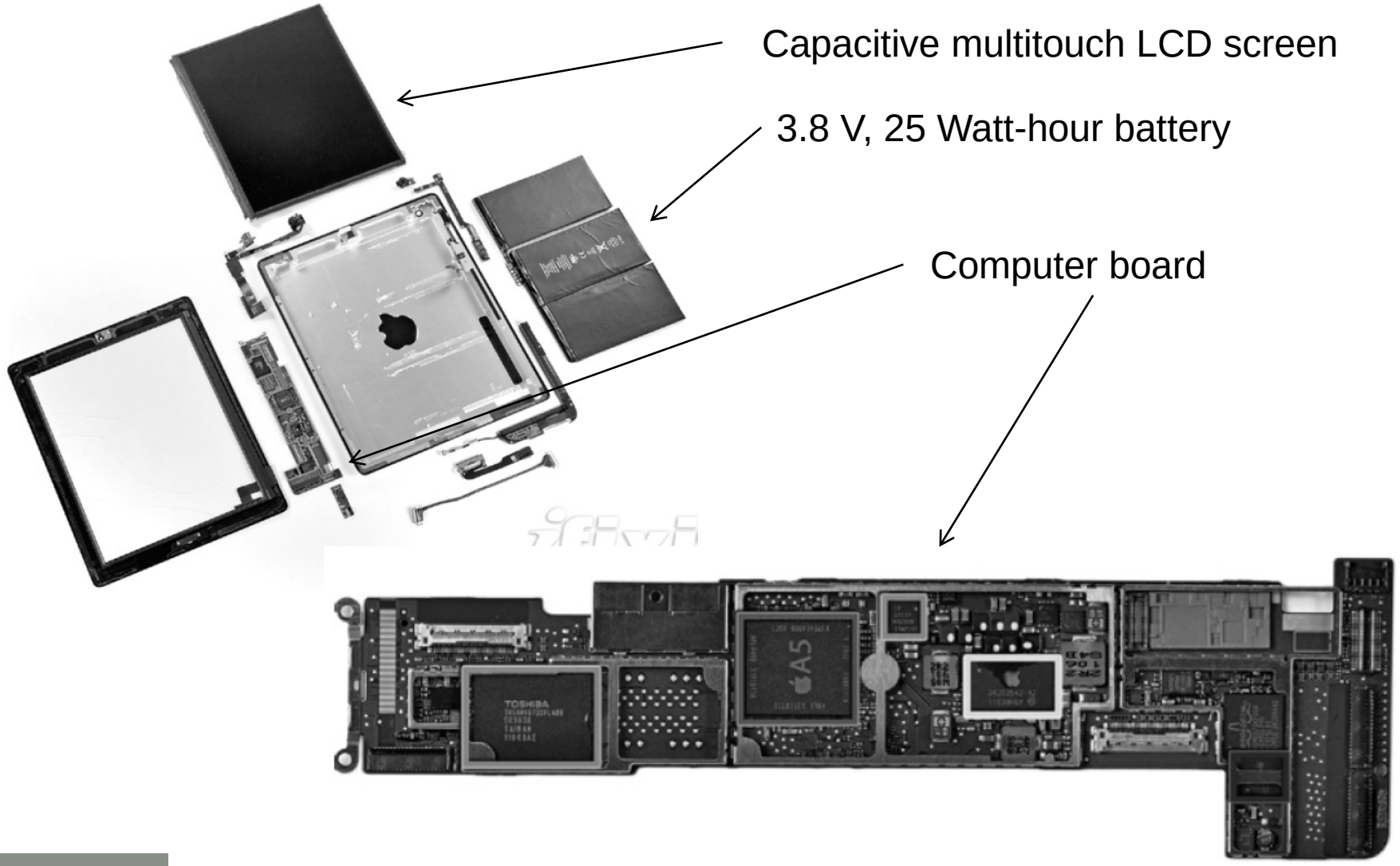
- PostPC device
- Supersedes keyboard and mouse
- Resistive and Capacitive types
  - Most tablets, smart phones use capacitive
  - Capacitive allows multiple touches simultaneously

Through the Looking Glass

- LCD screen: picture elements (pixels)
  - Mirrors content of frame buffer memory



Opening the Box

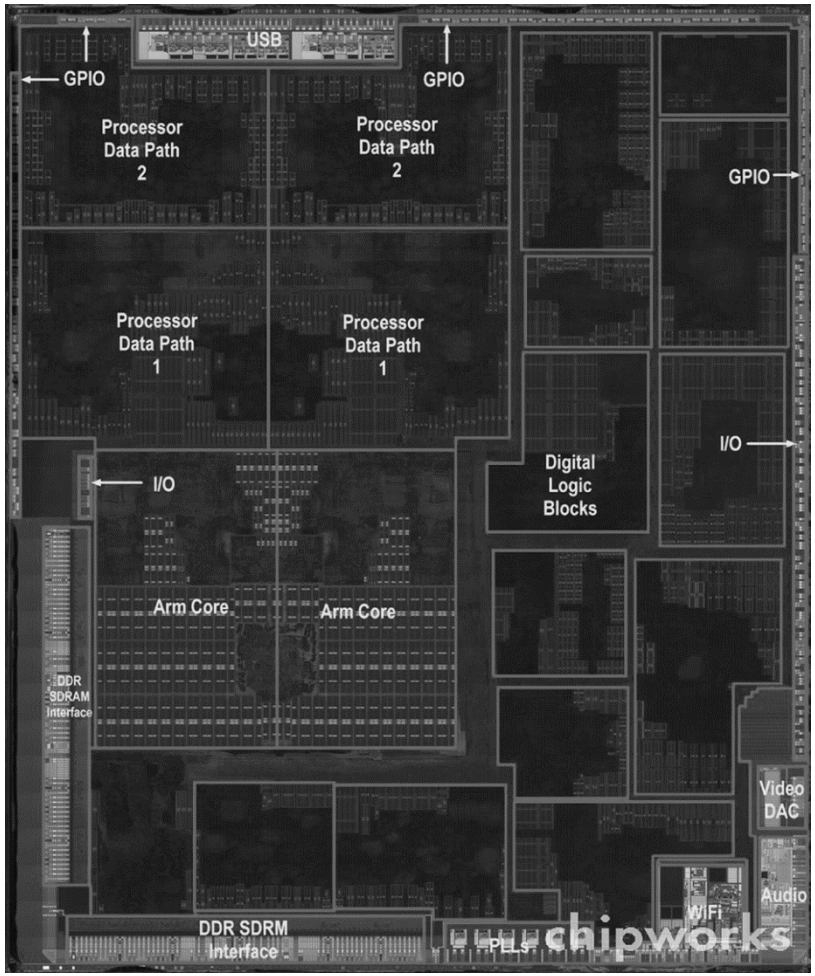


Inside the Processor (CPU)

- Datapath: performs operations on data
- Control: sequences datapath, memory, ...
- Cache memory
  - Small fast SRAM memory for immediate access to data

Inside the Processor

- Apple A5



Abstractions

The BIG Picture

- Abstraction helps us deal with complexity
  - Hide lower-level detail
- Instruction set architecture (ISA)
  - The hardware/software interface
- Application binary interface (ABI)
  - The ISA plus system software interface
- Implementation
  - The details underlying and interface

Αρχιτεκτονική Συνόλου Εντολών
ISA Instruction Set Architecture

Γλώσσα Ψηλού Επιπέδου

Μεταγλωττιστής

Συμβολική Γλώσσα

Συμβολομεταφραστής

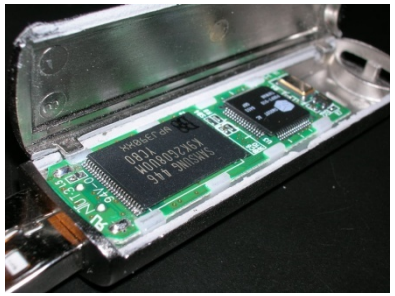
Γλώσσα Μηχανής

Αρχιτεκτονική Συνόλου Εντολών(ISA)

Υλικό (Υπολογιστής)

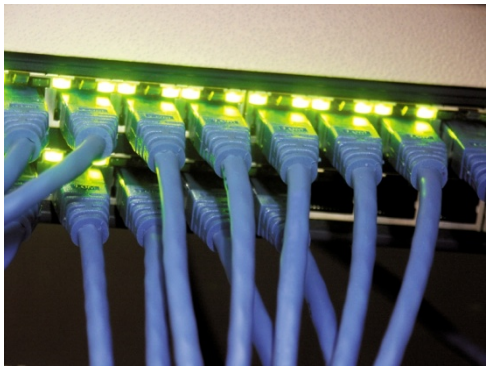
A Safe Place for Data

- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)



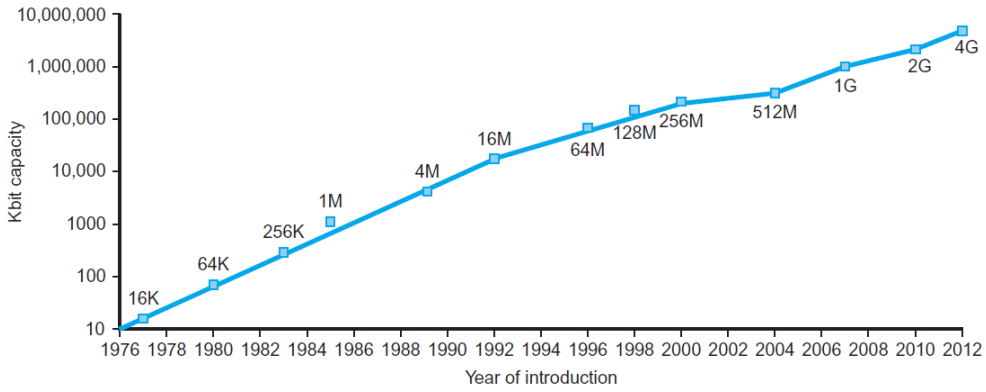
Networks

- Communication, resource sharing, nonlocal access
- Local area network (LAN): Ethernet
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth



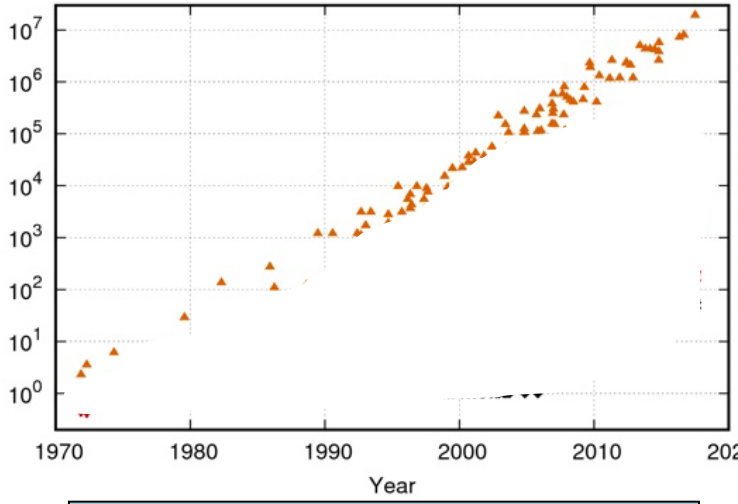
Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



DRAM capacity (there exist 8,16 and 32 GB products)

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

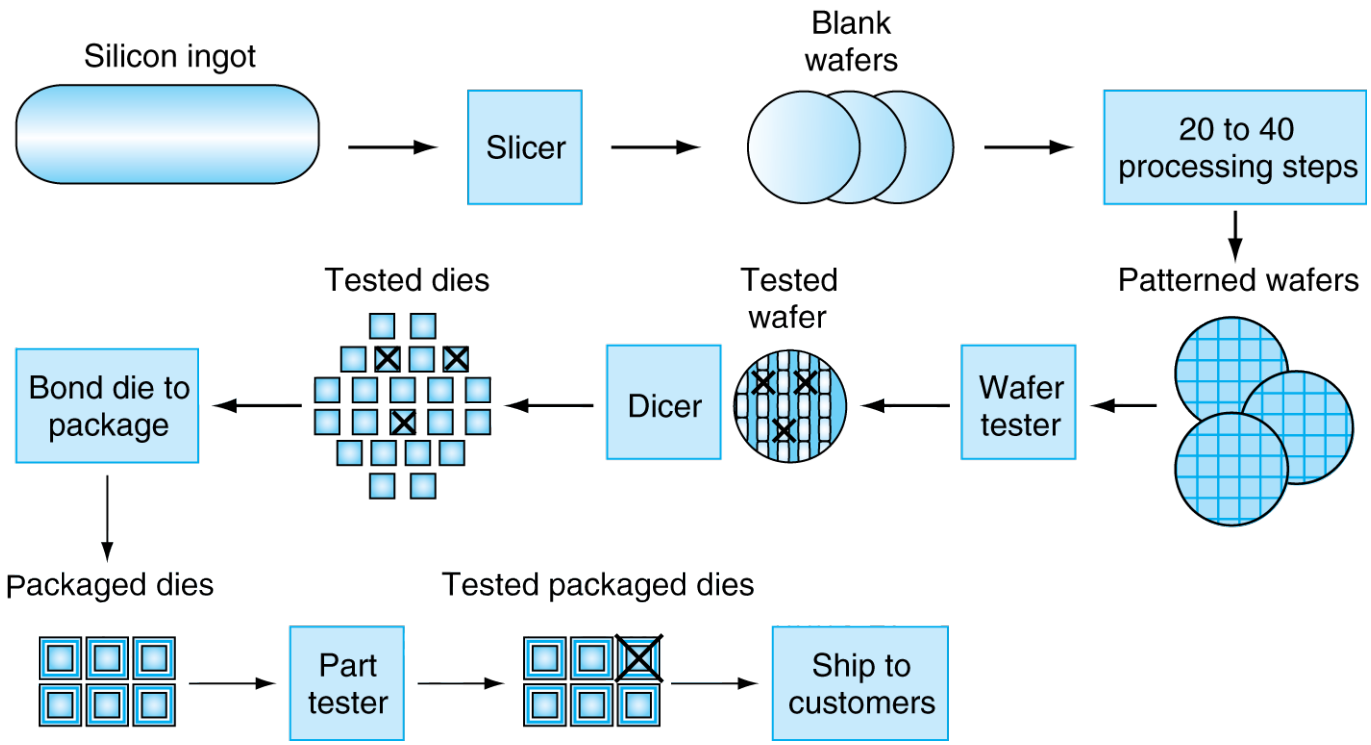


x1K Transistors per Processor Chip

Semiconductor Technology

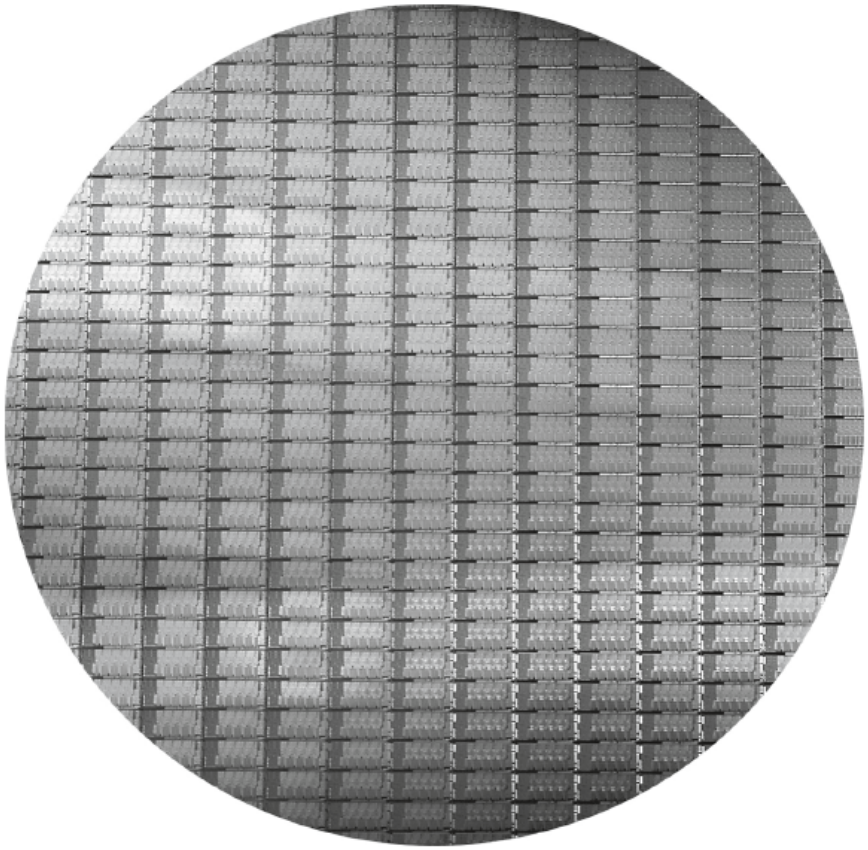
- Silicon: semiconductor
- Add materials to transform properties:
  - Conductors
  - Insulators
  - Switch

Manufacturing ICs



- Yield: proportion of working dies per wafer

Intel Core i7 Wafer

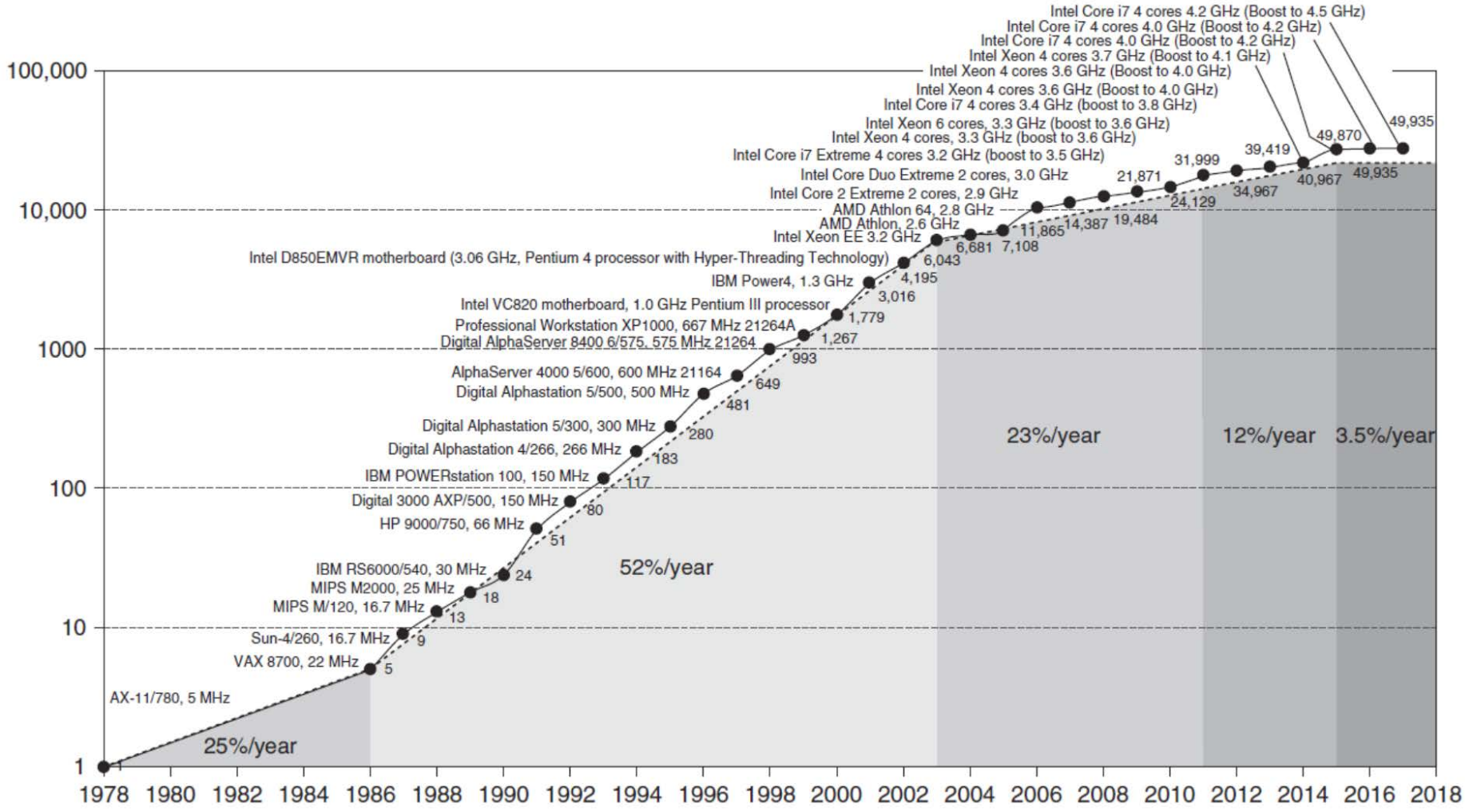


- 300mm wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm

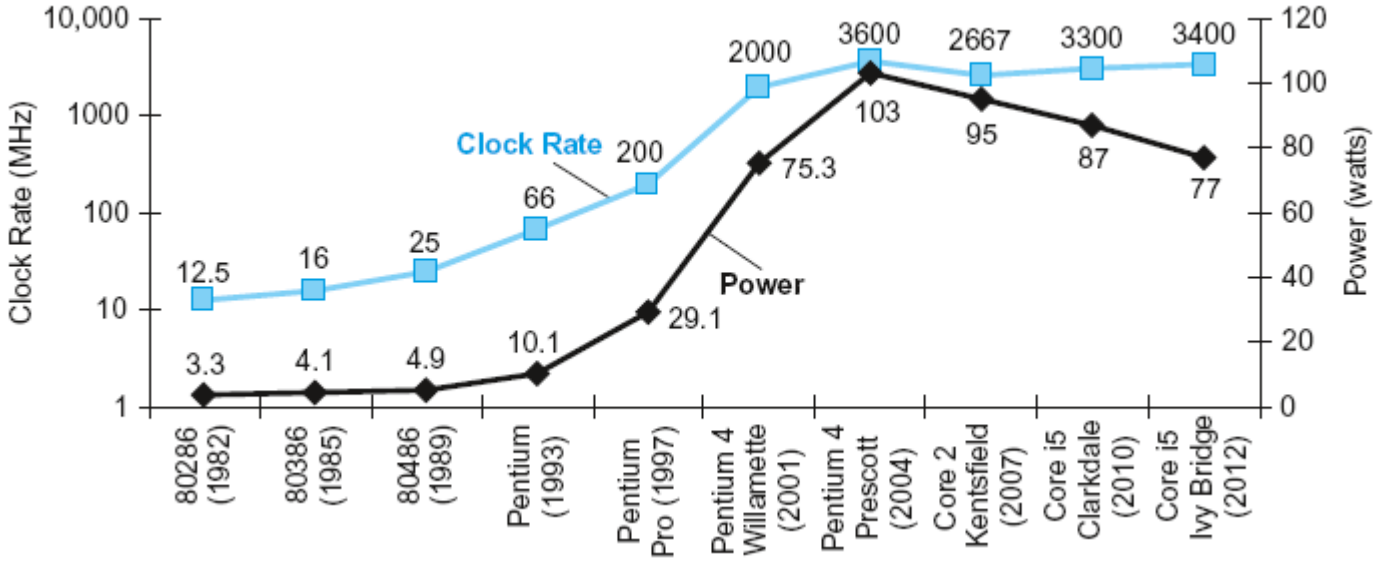
Πρόοδος της Τεχνολογίας

- Οι υπολογιστές είναι πιο γρήγοροι
  - 1971: 1 million instructions/sec
  - 2018: 10 billion instructions/sec
- Έχουν πιο πολύ μνήμη
  - 1971: 0.125 megabytes
  - 2018: 16.0 gigabytes
- Κοστίζουν πιο λίγο
  - 1971: $4,000,000
  - 2018: $1,000
- Speed/size/cost improvement factor:~100s billion

Single Processor Performance



Power Trends



- In CMOS IC technology

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

×30

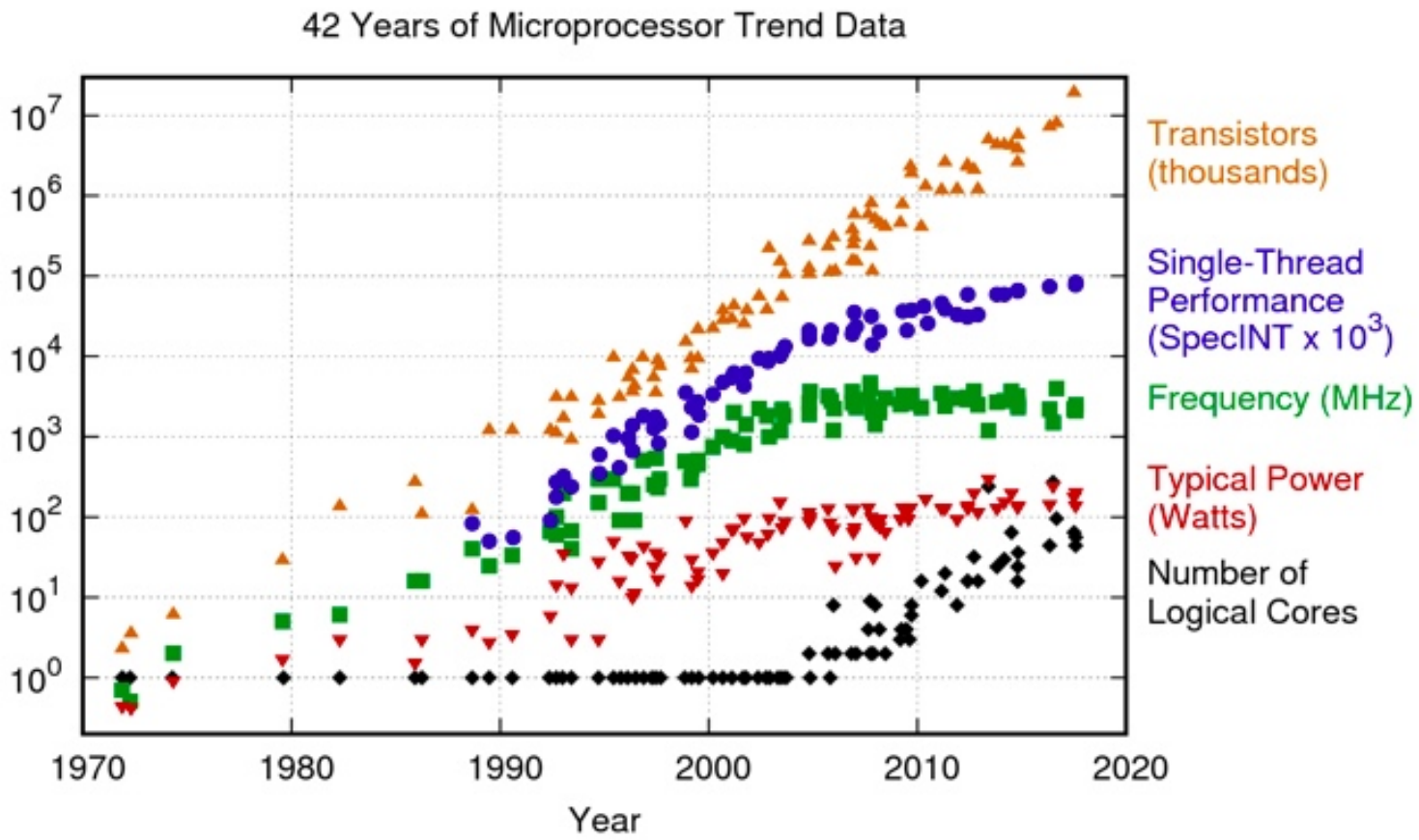
5V → 1V

×1000

Multiprocessors

- Multicore microprocessors
  - More than one processor (core) per chip
- Requires explicitly parallel programming
  - Compare with instruction level parallelism
    - Hardware executes multiple instructions at once
    - Hidden from the programmer
  - Hard to do
    - Programming for performance
    - Load balancing
    - Optimizing communication and synchronization

Microprocessor Trends



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/

Concluding Remarks

- Hardware properties challenge common sense
  - For better performance (analysis) need to understand computer organization/architecture
- Cost/performance is improving
  - Due to underlying technology development
  - Though rate of reduction is slowing down
- Hierarchical layers of abstraction
  - In both hardware and software
- Instruction set architecture
  - The hardware/software interface
- Power is a limiting factor
  - Use parallelism to improve performance