Προηγμένα Θέματα Οργάνωσης Υπολογιστών
9ο εξάμηνο ΣΗΜΜΥ

ακ. έτος: 2004-2005

Νεκτάριος Κοζύρης
nkoziris@cslab.ece.ntua.gr

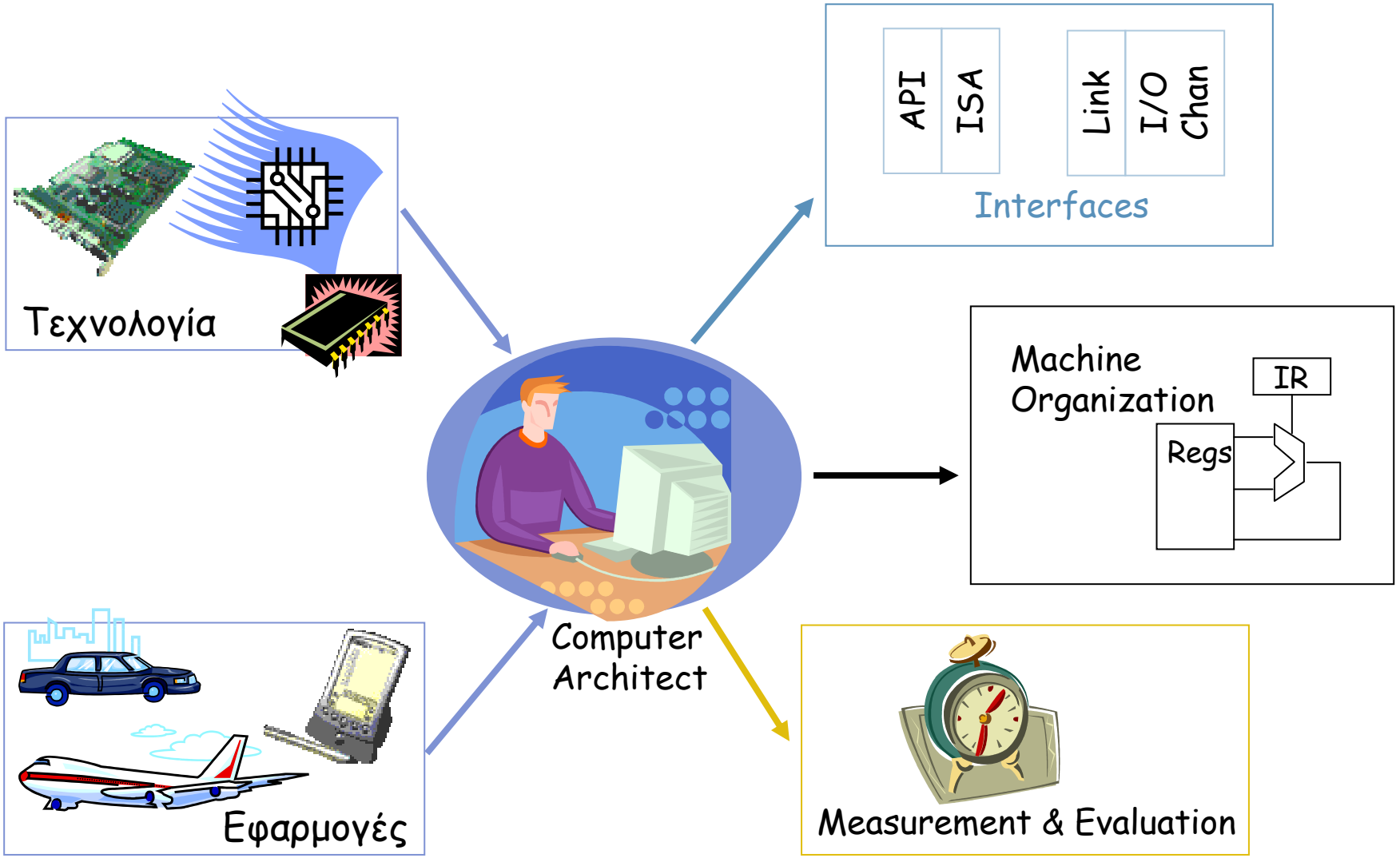
http://www.cslab.ece.ntua.gr/courses/advcomparch/

Computer Architecture Vs. Computer Organization

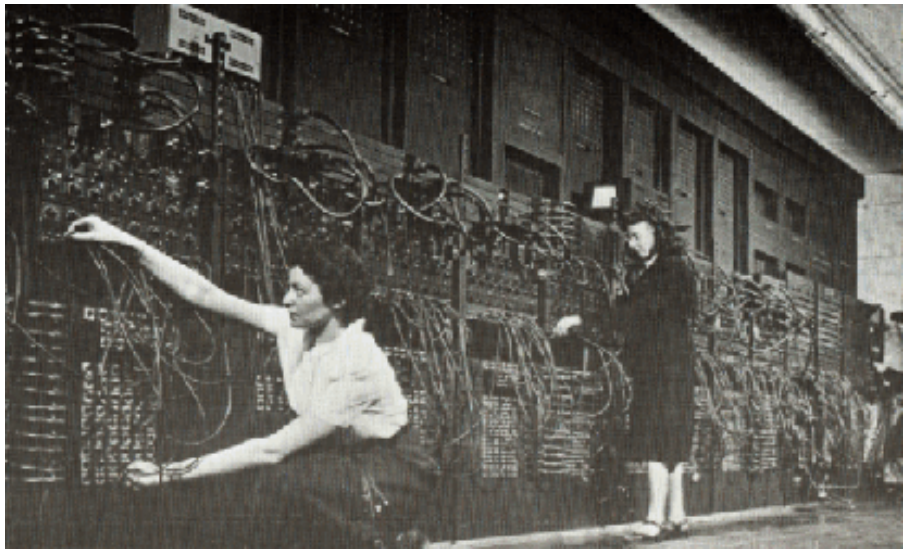
- Ο όρος Computer architecture (Αρχιτεκτονική Υπολογιστών) κακώς περιορίζεται συχνά στο σχεδιασμό και την υλοποίηση του instruction set.
- Ακριβέστεροι Ορισμοί:
  - Instruction set architecture: Εξυπηρετεί σαν όριο μεταξύ software και hardware. Είναι ορατή στον προγραμματιστή σε γλώσσα assembly ή στον κατασκευαστή του compiler (registers, data types, instruction set, instruction formats, addressing modes)
  - Η υλοποίηση ενός μηχανήματος έχει δύο συντελεστές:
    - Organization: συμπεριλαμβάνει το υψηλού επιπέδου σχεδιασμό των υπολογιστών όπως: το σύστημα μνημών, τη δομή του διαδρόμου, το εσωτερικό της CPU το οποίο συμπεριλαμβάνει υλοποίηση των αριθμητικών, λογικών εντολών, εντολών διακλάδωσης και μεταφοράς δεδομένων (datapath & control).
    - Hardware: Αναφέρεται στα ειδικά χαρακτηριστικά του μηχανήματος όπως λογικός σχεδιασμός, τεχνολογία διασύνδεσης , εξωτερικές συνδέσεις και τεχνολογία packaging.
- Γενικά, η Αρχιτεκτονική Υπολογιστών (Computer Architecture) αναφέρεται στα ακόλουθα τρία συστατικά:

1- Instruction set architecture 2- Organization. 3- Hardware.

Αρχιτεκτονική Υπολογιστών



Then …



ENIAC (1943-1946) by Mauchly and Eckert

Dimension: 3 ft × 8 ft × 100 ft

15,000 vacuum tubes + lots of switches

Memory : Twenty 10-digit decimal numbers

Speed: 800 operations/sec

10 years of service – more calculations than done by the entire human race up to 1946.

"I think there is a world market for maybe five computers."

Thomas Watson, Chairman of IBM, 1943

Now …

"Earth Simulator" (2002) by NEC

Earth Simulator Center

5120 Processors (640 Gflops at peak)

Dimension: Two basketball courts

Memory: >20 TB (≈ 60 trillion bytes)

Speed: 40 trillion operations/sec

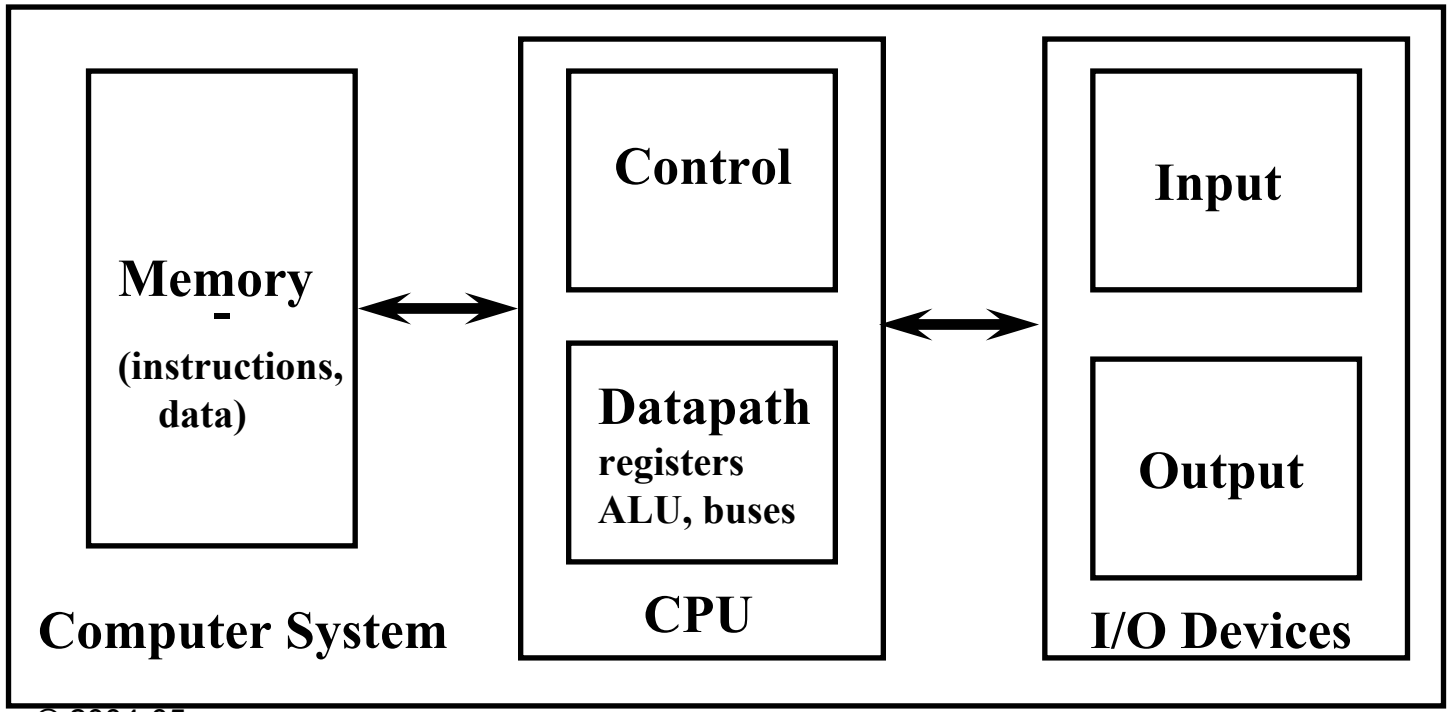
Cost: ??

Οι Γενιές του Hardware Υπολογιστών

- Πρώτη Γενιά, 1946-59: Vacuum Tubes, Relays, Mercury Delay Lines:
  - ENIAC (Electronic Numerical Integrator and Computer): Πρώτος Η/Υ, 18000 vacuum tubes, 1500 relays, 5000 additions/sec.
  - Πρώτο πρόγραμμα αποθηκευμένο σε υπολογιστή: EDSAC (Electronic Delay Storage Automatic Calculator).
- Δεύτερη Γενιά, 1959-64: Διακριτά Transistors.
- Τρίτη Γενιά, 1964-75: Μικρού και Μεσαίου μεγέθους Ολοκληρωμένα Κυκλώματα.
- Τέταρτη Γενιά, 1975-Present: Ο Μικροϋπολογιστής. Μικροεπεξεργαστές βασισμένοι σε τεχνολογία.

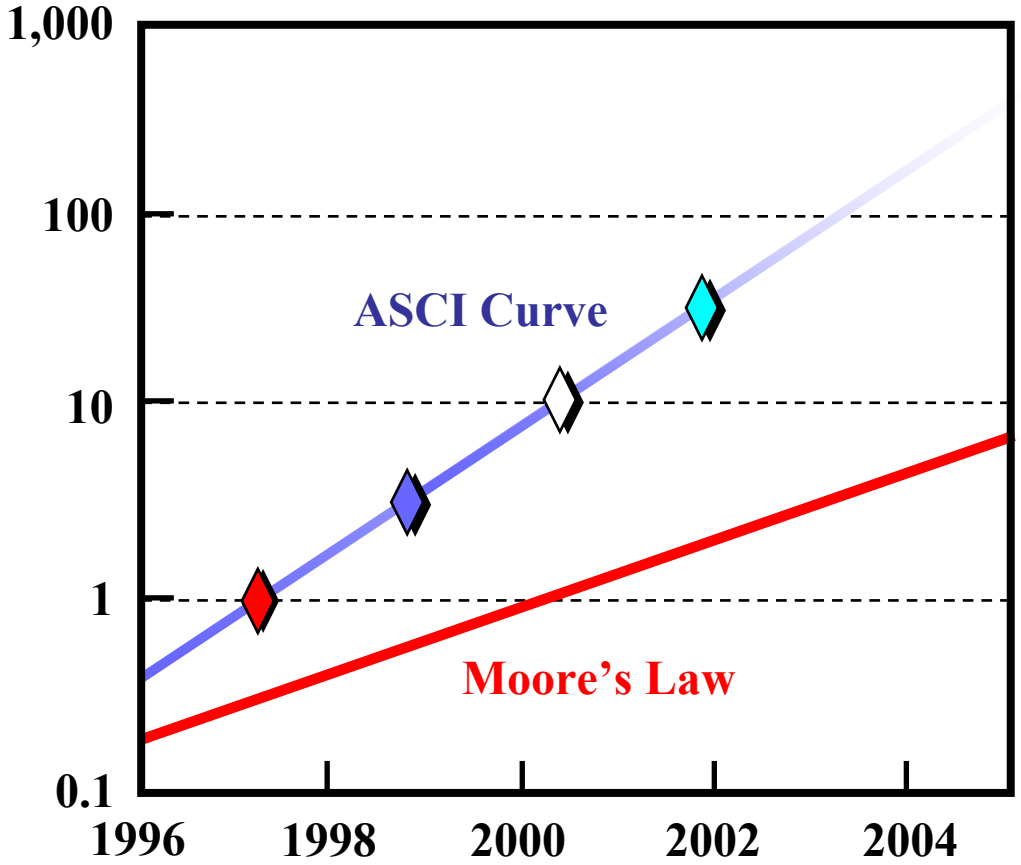
Το Υπολογιστικό Μοντέλο Von-Neumann

- Διαχωρισμός της υπολογιστικής μηχανής σε συνιστώσες:
  - Κεντρική Μονάδα Επεξεργασίας (Central Processing Unit - CPU): Control Unit (instruction decode, sequencing of operations), Datapath (registers, arithmetic and logic unit, buses).
  - Μνήμη (memory): Αποθήκευση εντολών και τελεστών.
  - Είσοδος/Έξοδος (Input/Output - I/O).
  - Η έννοια του αποθηκευμένου προγράμματος: Εντολές από ένα σύνολο εντολών εξάγονται από τη μνήμη και εκτελούνται μία-μία.



Background
Moore's Law and More

Computing Power (tflops)



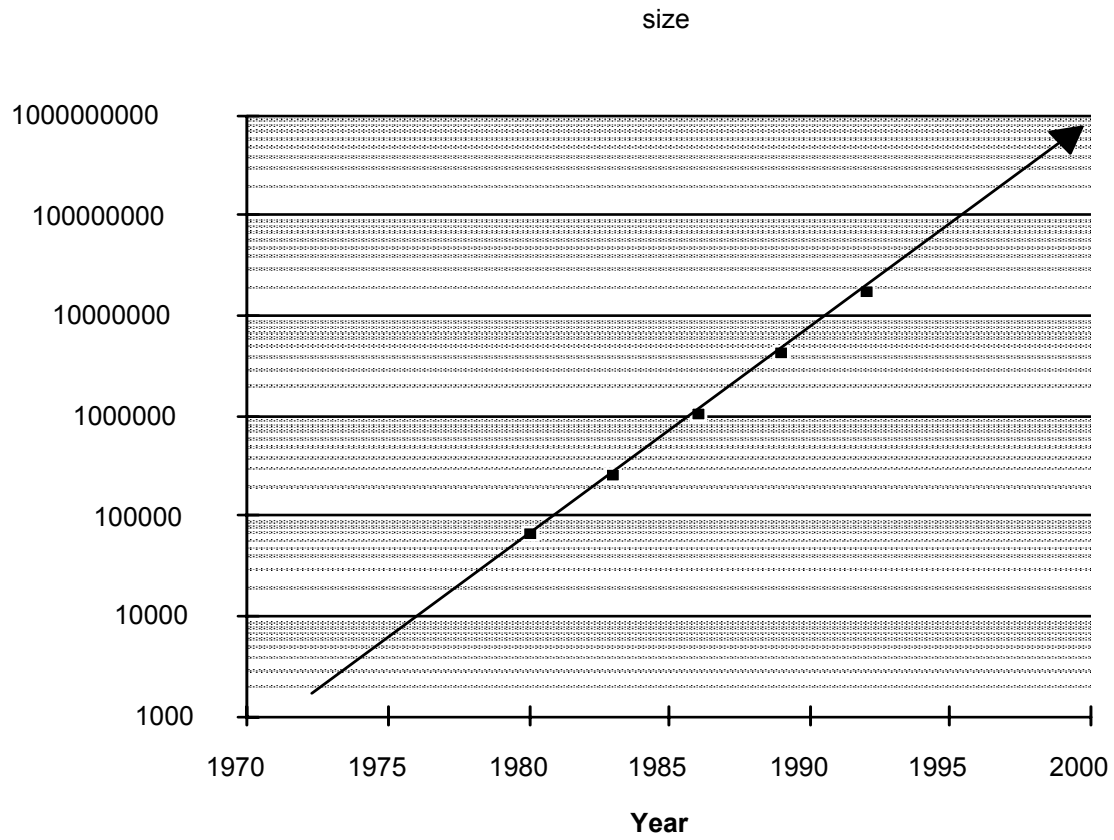
MICROPROCESSORS

2x increase in performance every 18-24 months ("Moore's Law")

INNOVATIVE DESIGNS

Specialized Computers
Cellular Architectures
Processors-in-Memory
HTMT

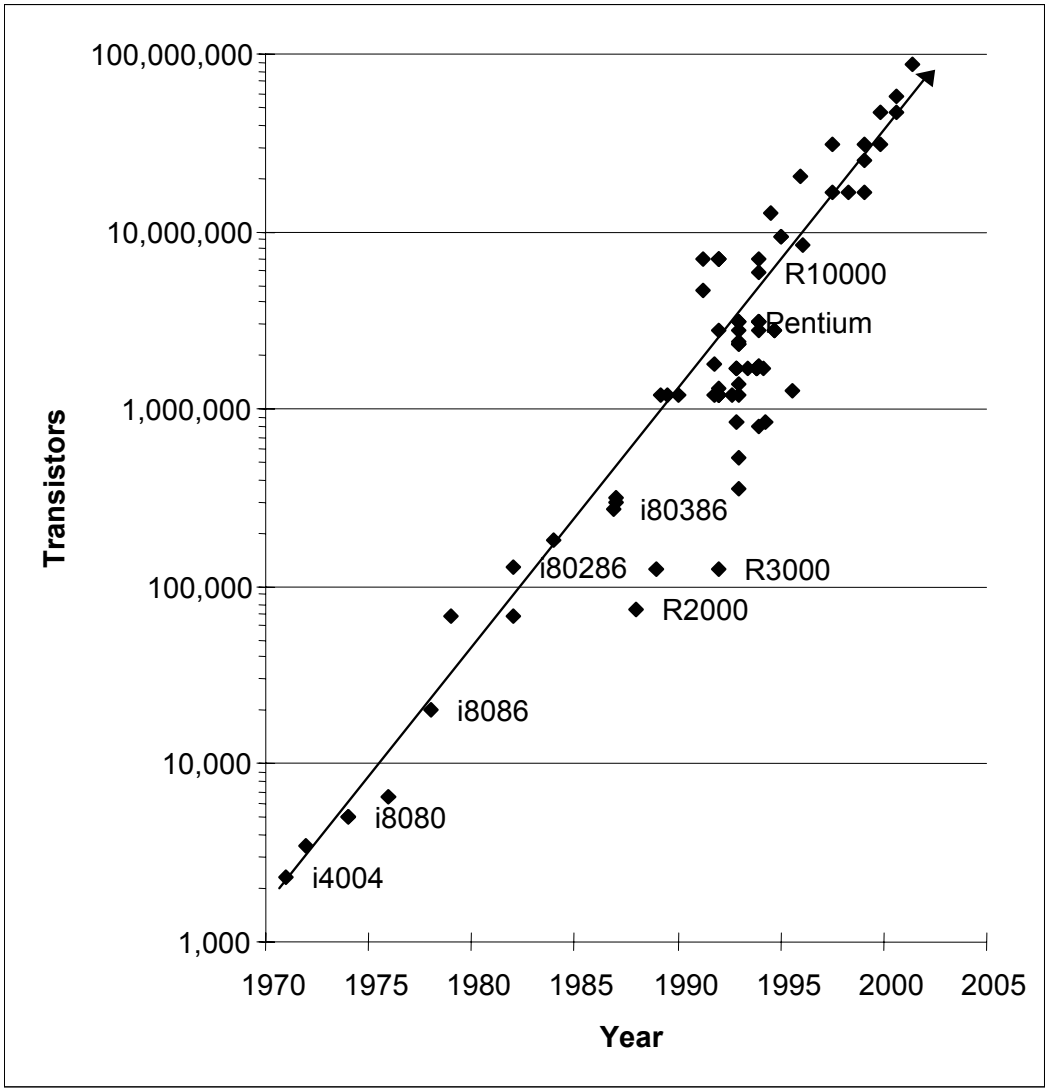
Αύξηση της χωρητικότητας των VLSI Dynamic RAM Chips



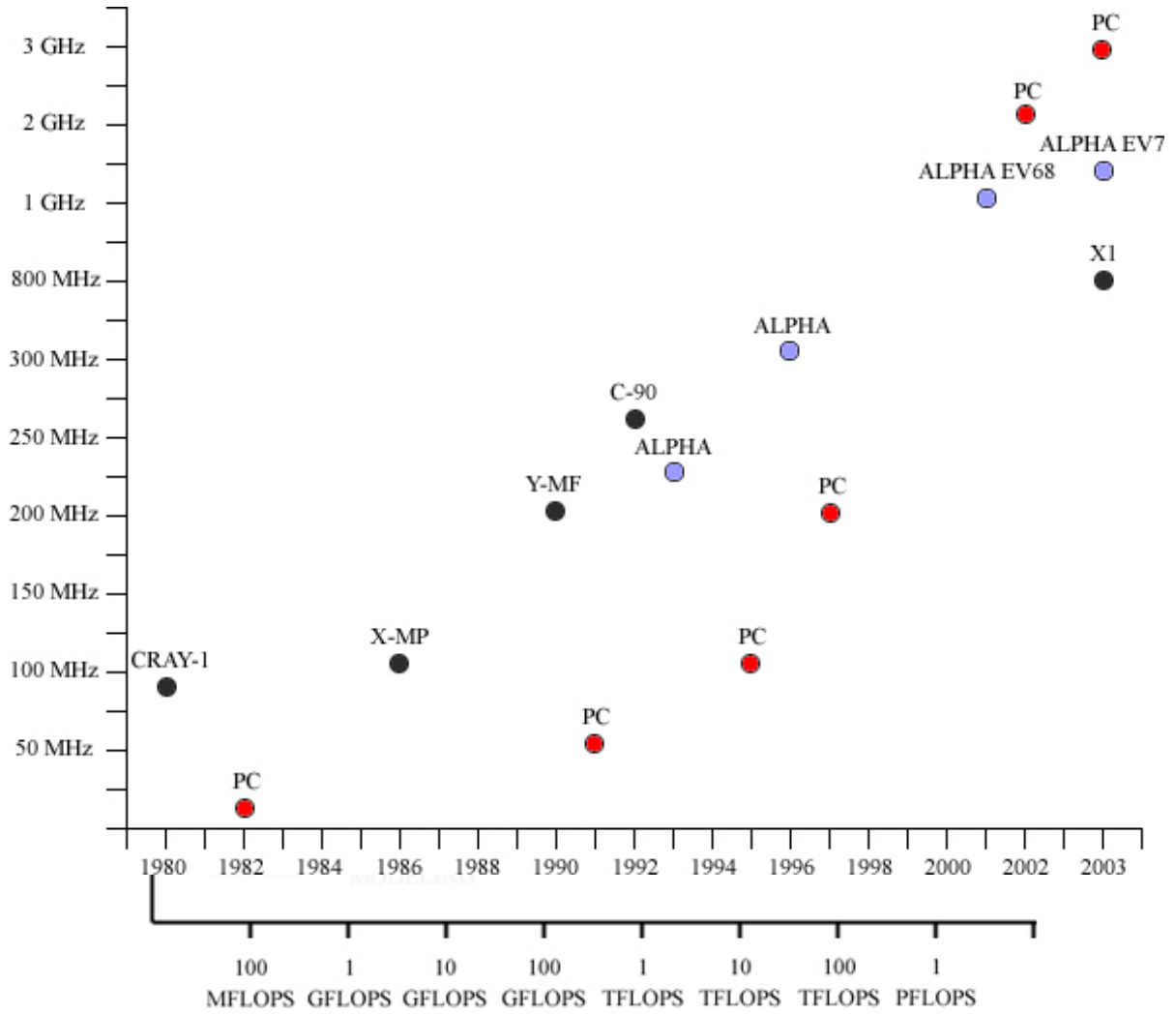
έτος	μέγεθος(Mbit)
1980	0.0625
1983	0.25
1986	1
1989	4
1992	16
1996	64
1999	256
2000	1024

1.55X/έτος, δηλαδή διπλασιάζεται κάθε 1.6 χρόνια

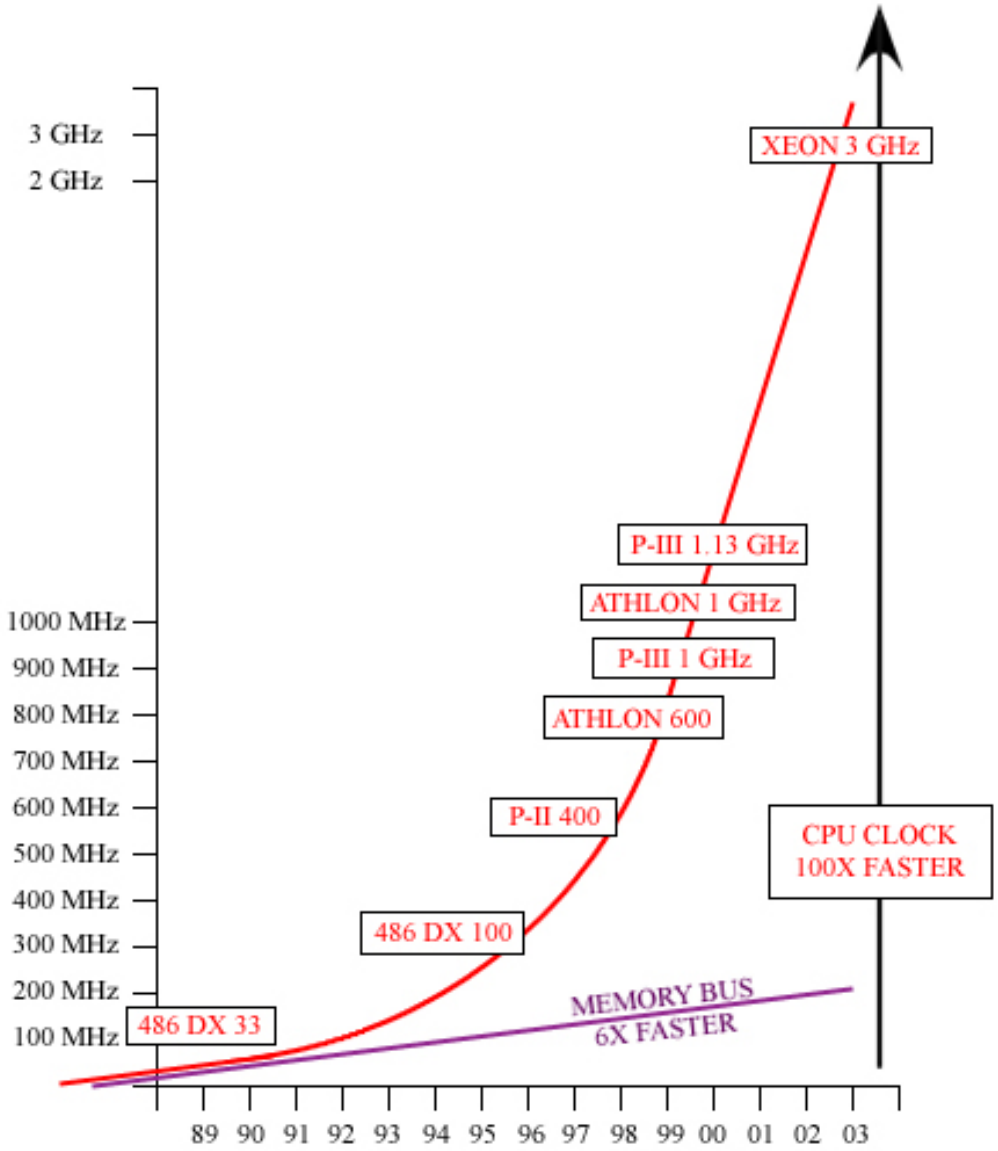
Trends in Microprocessor Transistors



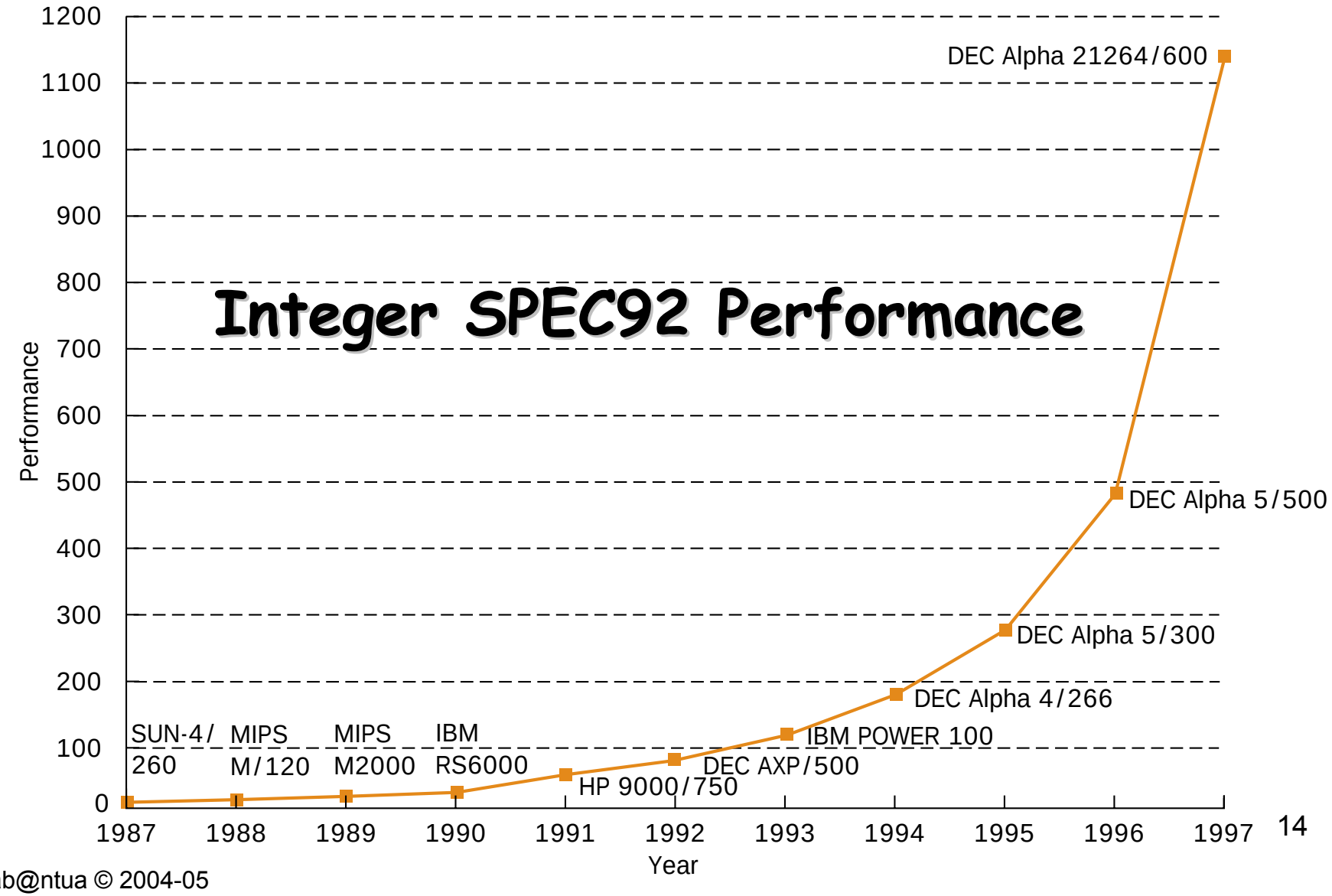
Clock Speeds



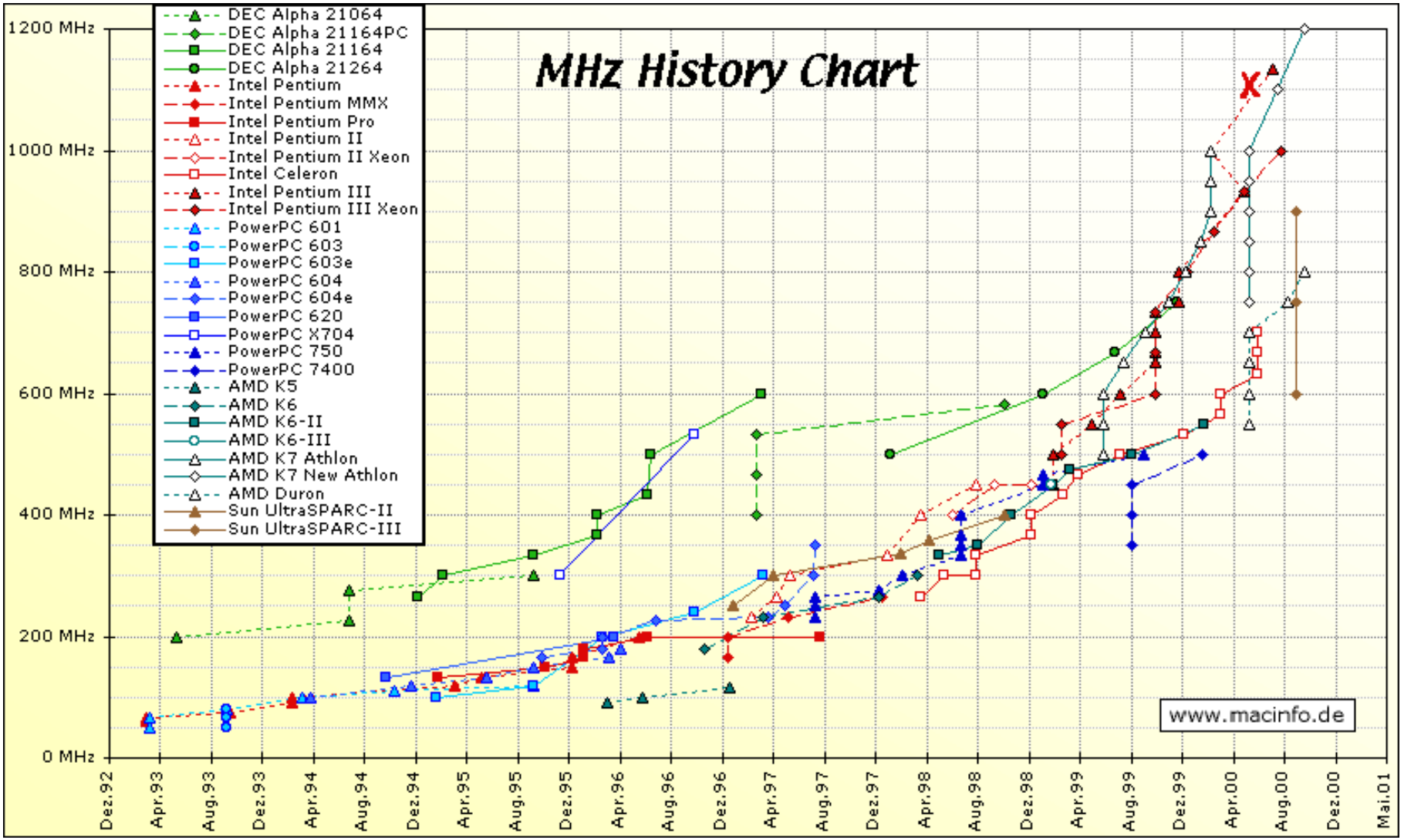
CPU Clock



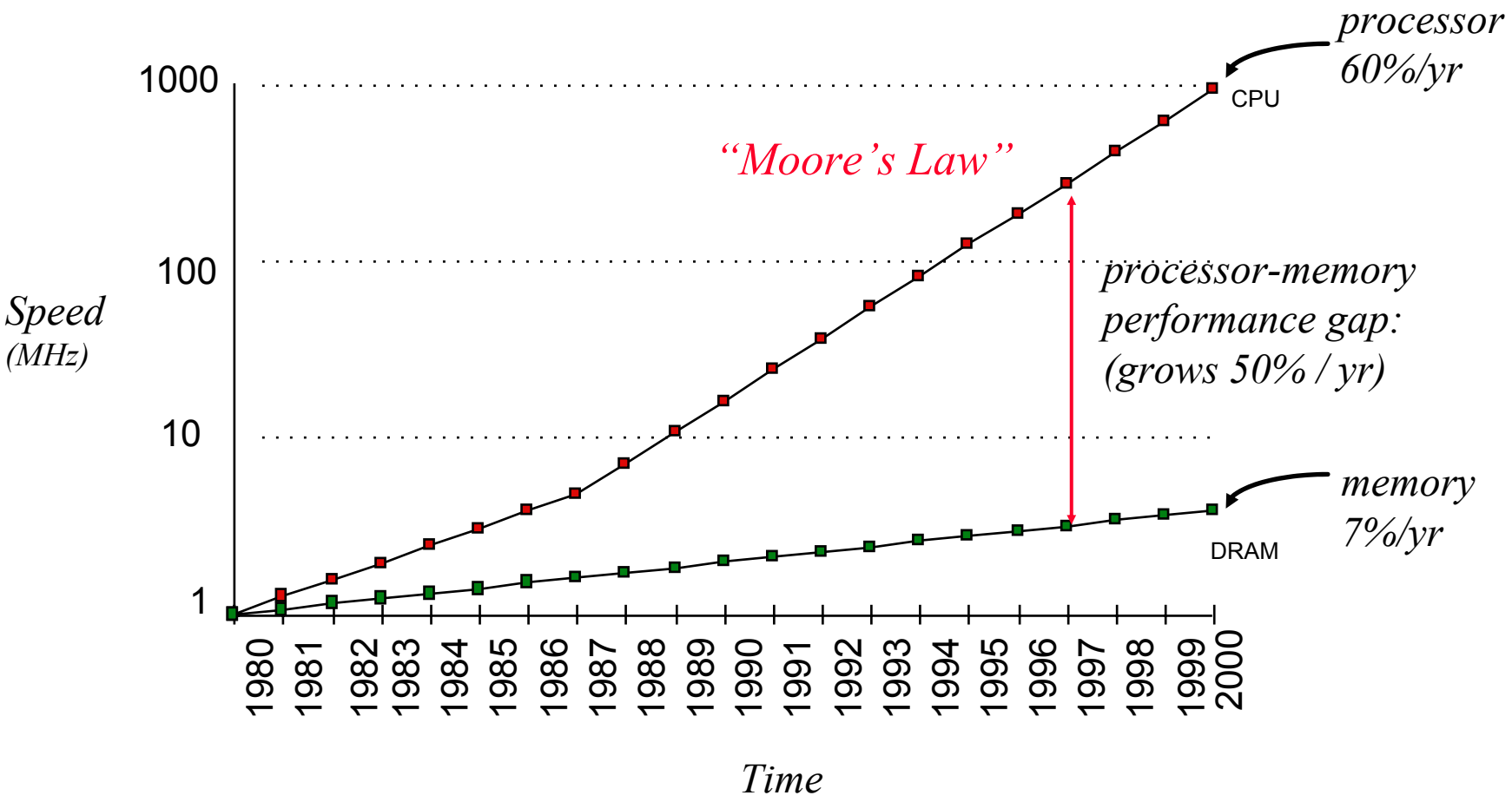
Αύξηση της Επίδοσης του Workstation-Class Microprocessors 1987-1997



Microprocessor Clock Rate



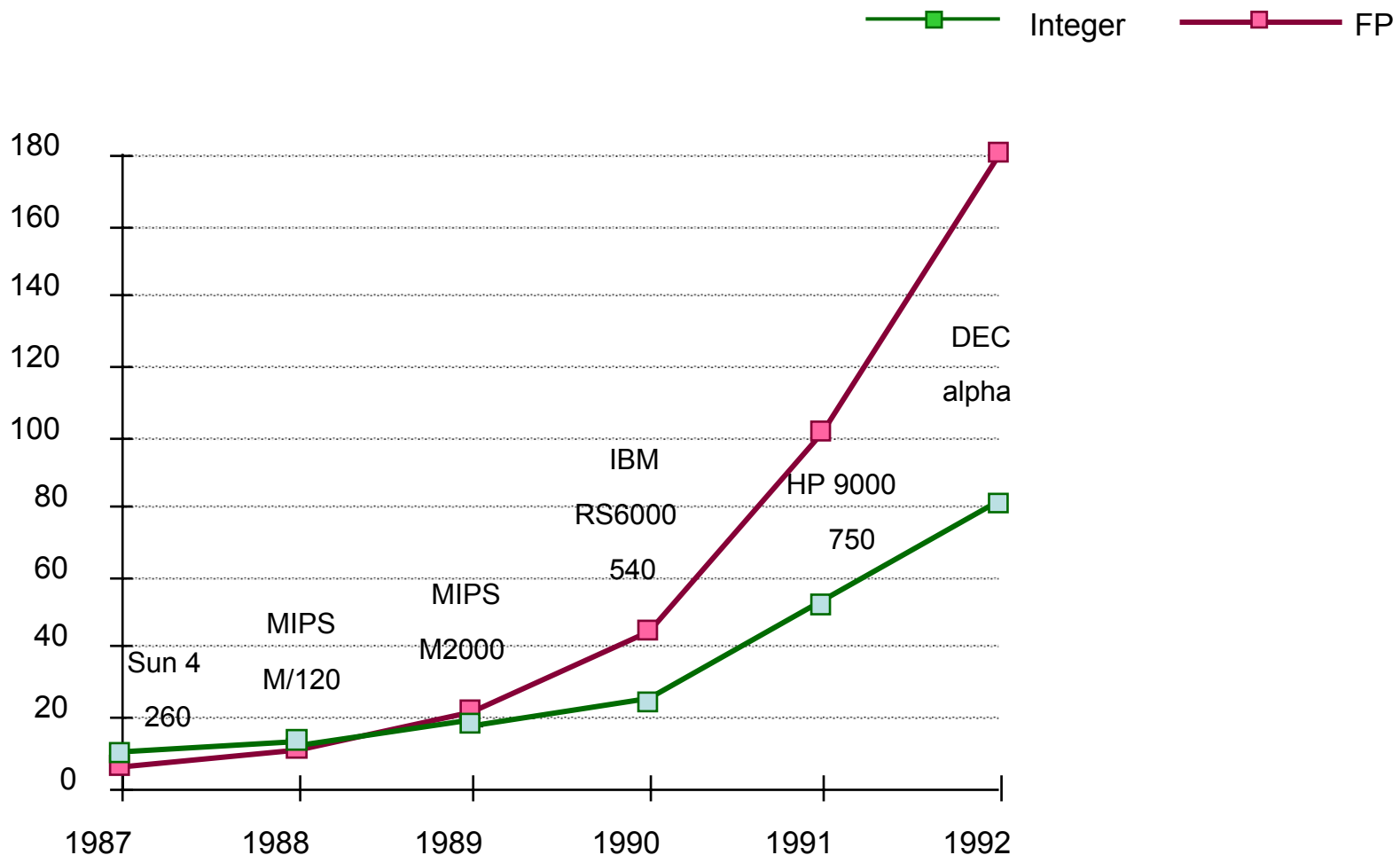
Processor-Memory Gap



Ομοίως και για το χώρο αποθήκευσης

- Απόκλιση μεταξύ χωρητικότητας μνήμης και ταχύτητας
  - Η χωρητικότητα αυξήθηκε 1000 φορές από 1980 έως 1995, η ταχύτητα μόνο 2 φορές
- Οι μεγαλύτερες μνήμες είναι πιο αργές, ενώ οι επεξεργαστές γίνονται πιο γρήγοροι
  - Ανάγκη μεταφοράς περισσότερων δεδομένων παράλληλα
  - Ανάγκη για βαθύτερες ιεραρχίες μνήμης
- Η παραλληλία αυξάνει το ενεργό μέγεθος κάθε επιπέδου ιεραρχίας της μνήμης, χωρίς να αυξηθεί ο χρόνος πρόσβασης

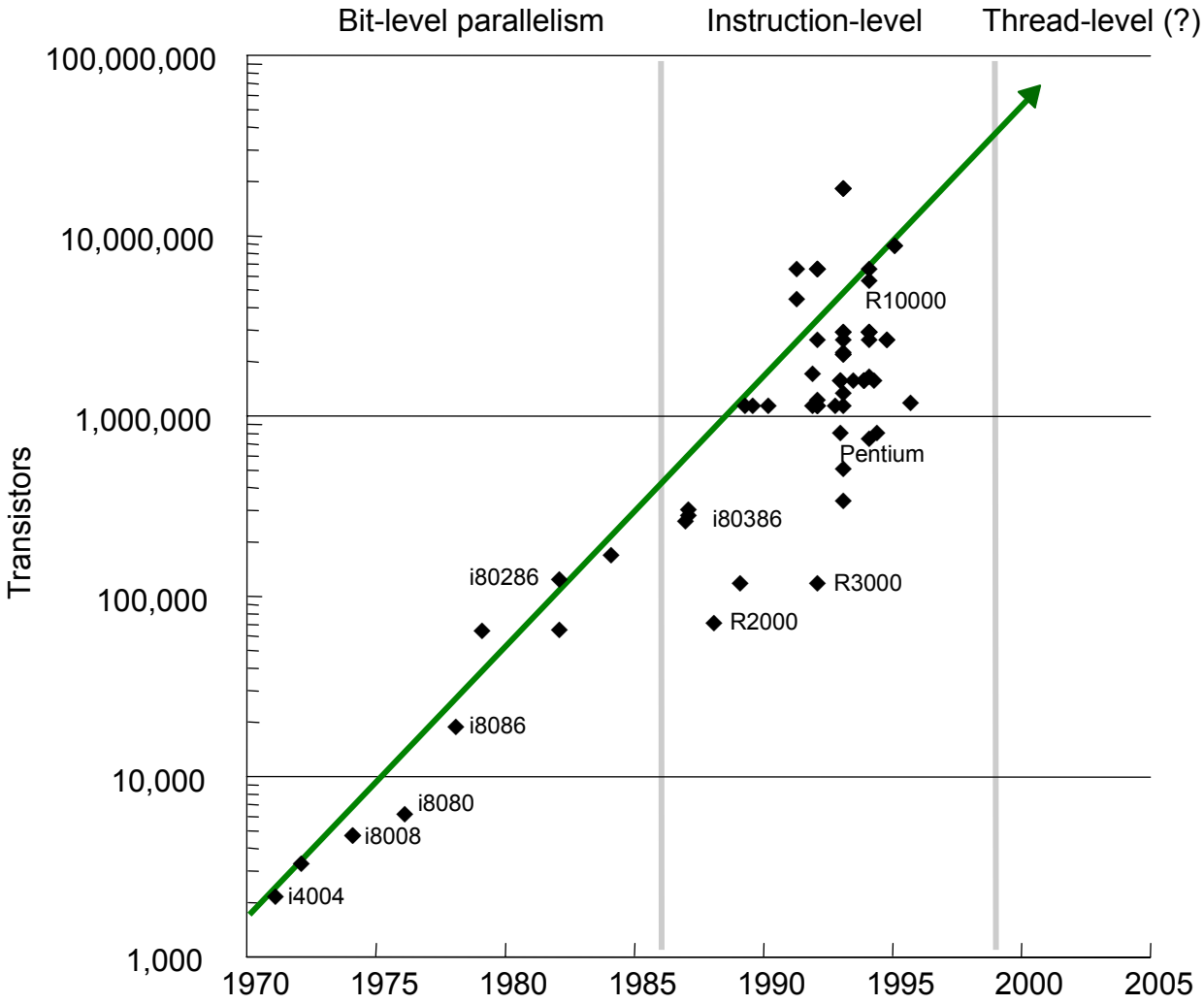
Τάσεις της Τεχνολογίας

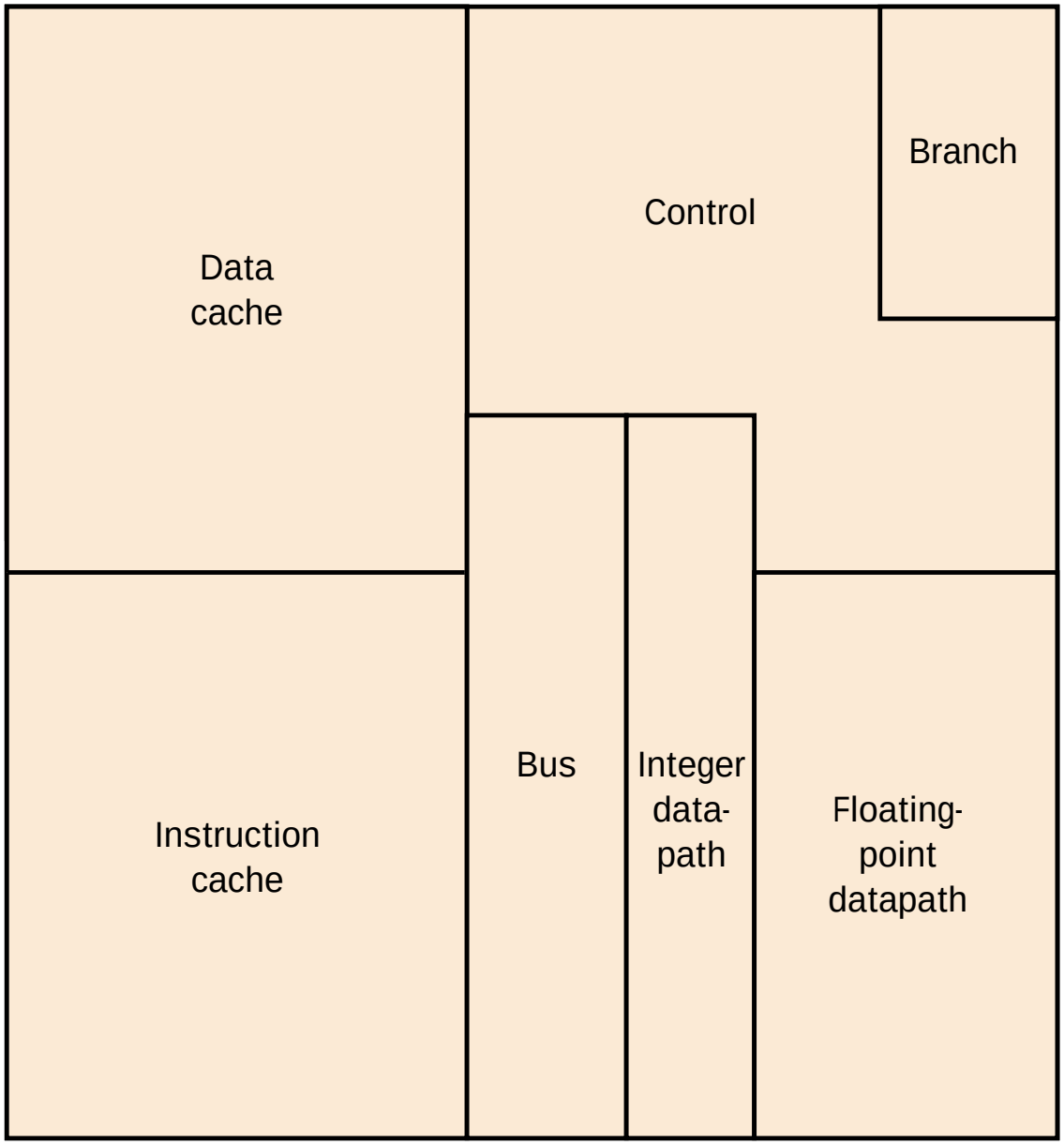


Τάσεις στην Αρχιτεκτονική Υπολογιστών

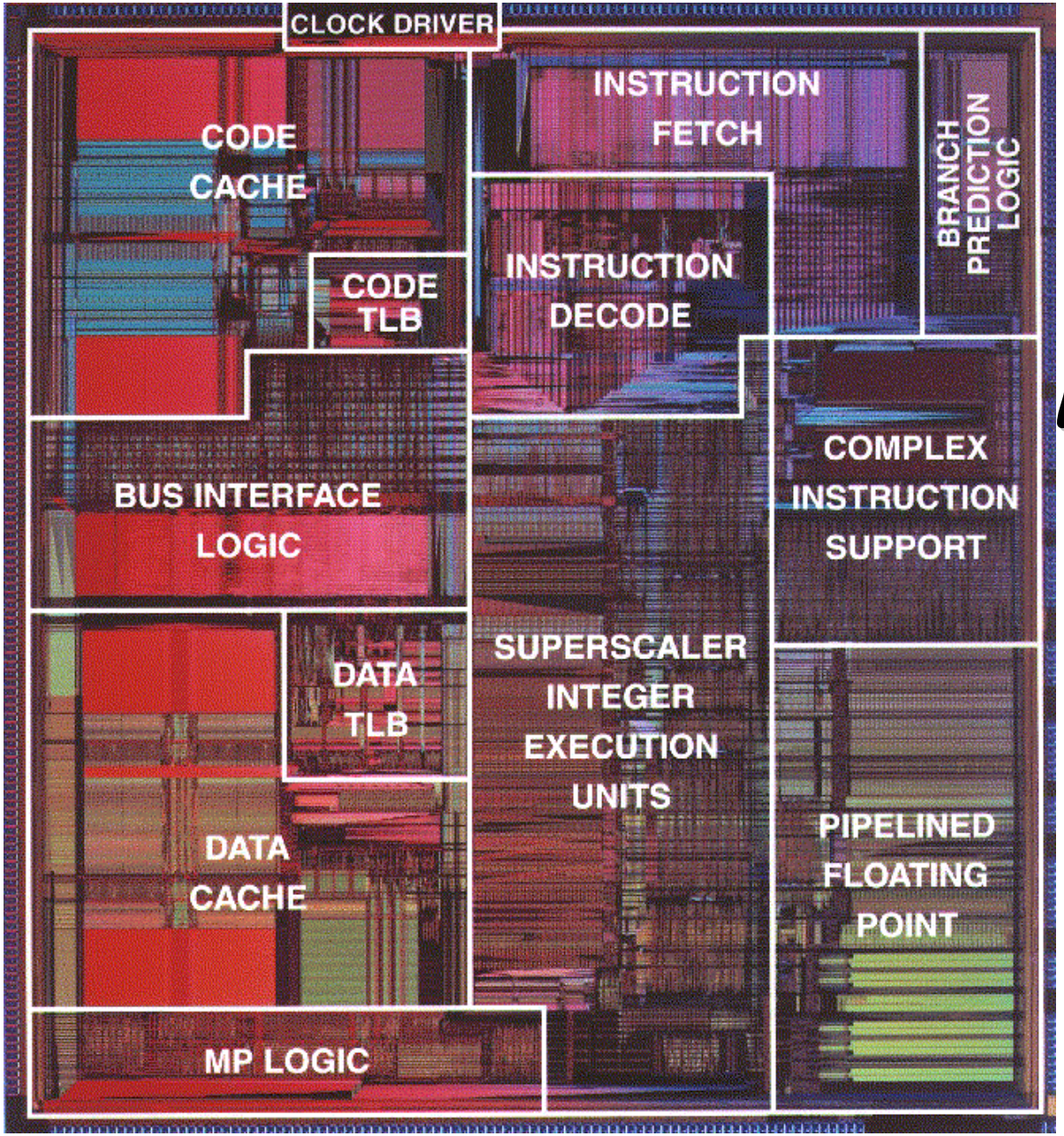
- Έως το 1985: Παραλληλία σε επίπεδο bit: 4-bit -> 8 bit -> 16-bit
- Μέσα δεκαετίας 1980s έως μέσα δεκαετίας 1990: Παραλληλία σε επίπεδο εντολής (instruction level parallelism)
- Επόμενο βήμα: Παραλληλία σε επίπεδο thread

Εξέλιξη Παραλληλίας στους επεξεργαστές

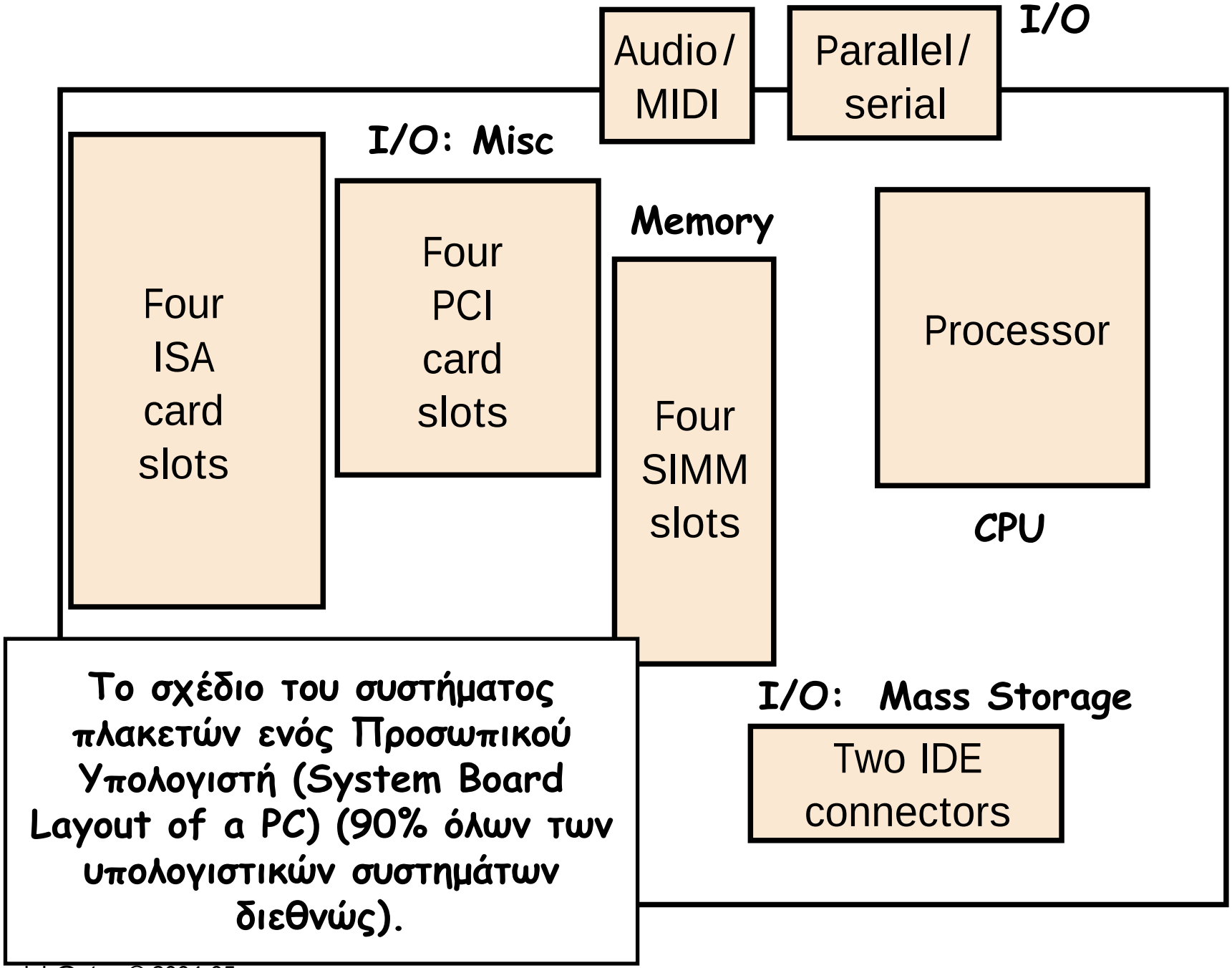




Διάταξη ενός Τυπικού Μικροεπεξεργαστή : The Intel Pentium Classic

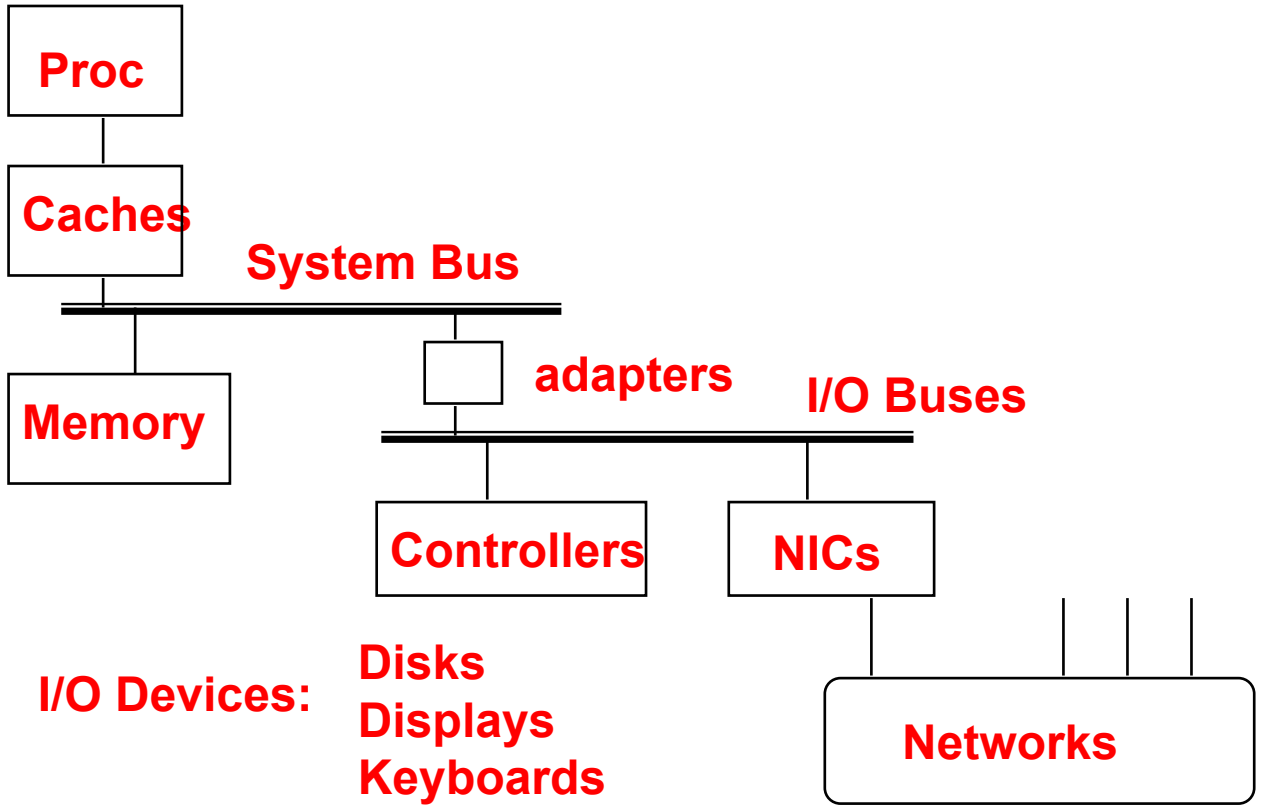


Διάταξη ενός Τυπικού Μικροεπεξεργαστή : The Intel Pentium Classic

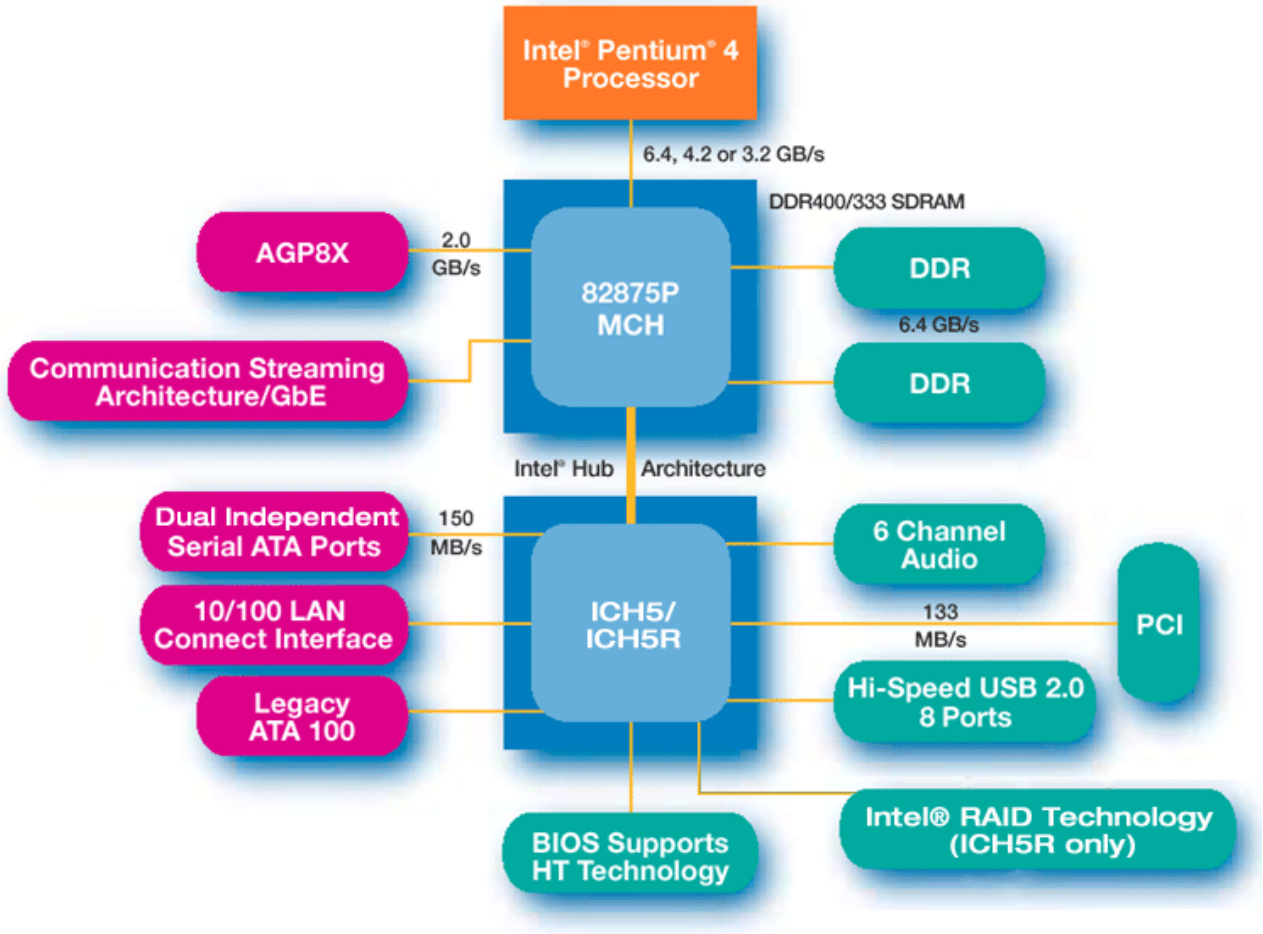


Το σχέδιο του συστήματος πλακετών ενός Προσωπικού Υπολογιστή (System Board Layout of a PC) (90% όλων των υπολογιστικών συστημάτων διεθνώς).

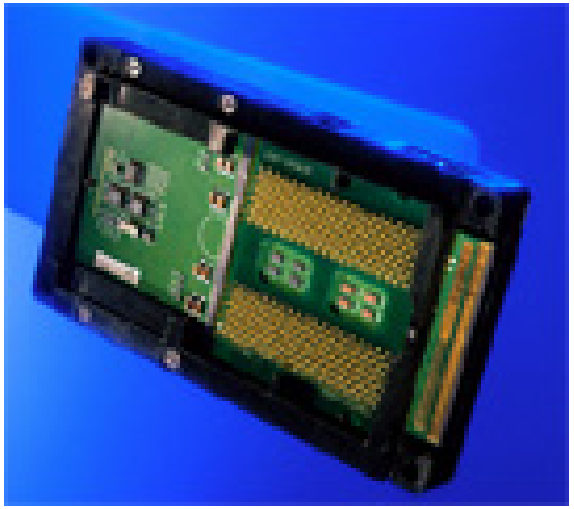
Computer System Components



Intel® 875P Chipset



How Fast Can You Go?



Intel Pentium 4

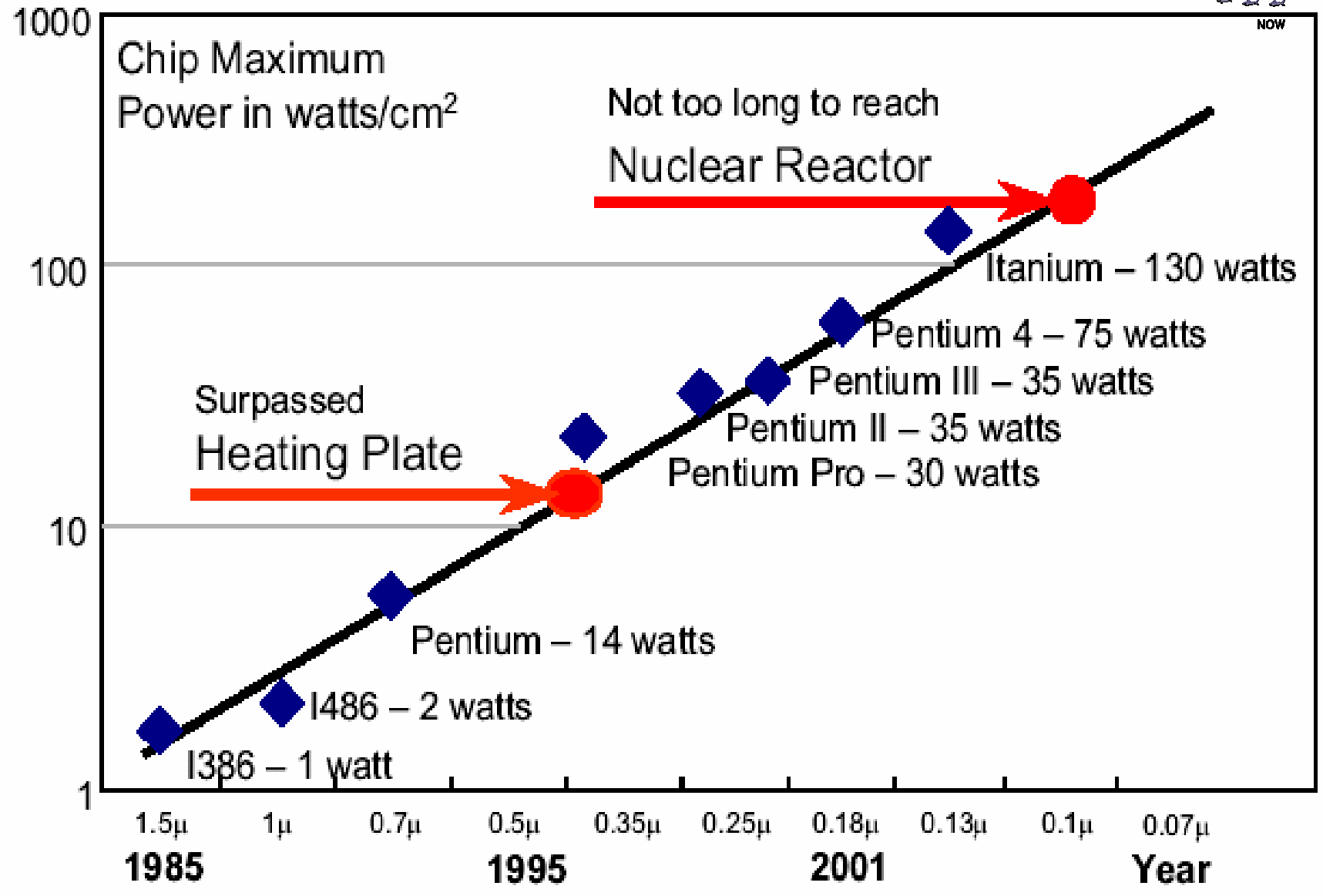
Max Speed: >3 GHz

System Bus: >800 MHz

>100 M transistors

Power dissipation???

Low Power Cluster Architectures sensitivity to power consumption

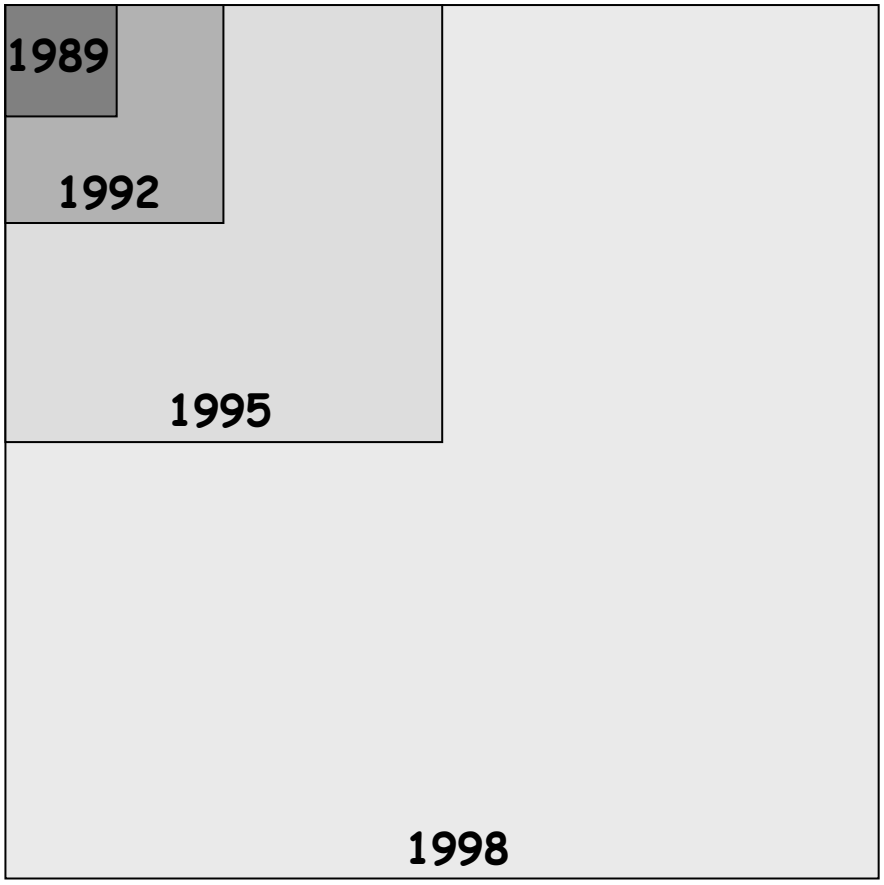


Ο ρόλος του Σχεδιαστή Υπολογιστών

- Καθορίζει ποια χαρακτηριστικά είναι σημαντικά για ένα νέο μηχάνημα. Στη συνέχεια σχεδιάζει ένα μηχάνημα που να μεγιστοποιεί την επίδοση και παράλληλη να μην υπερβαίνει τους περιορισμούς κόστους
- Επιμέρους χαρακτηριστικά
  - Σχεδιασμός του instruction set
  - Οργάνωση των λειτουργιών
  - Λογικός σχεδιασμός και υλοποίηση (IC design, packaging, power, cooling …)

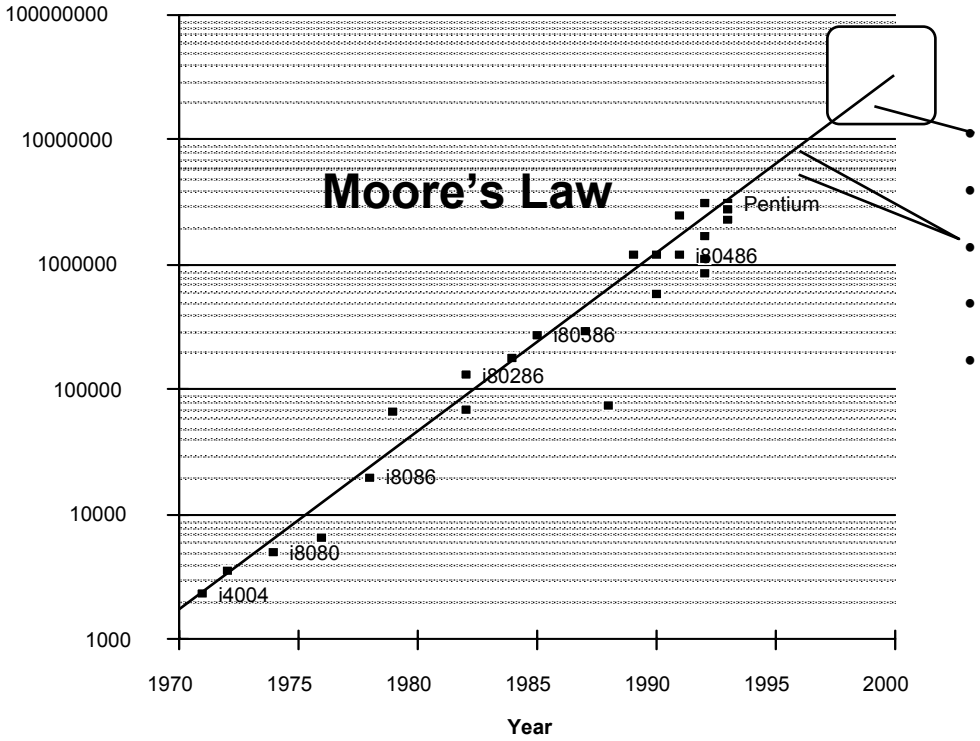
Περιορισμοί από την Τεχνολογία

- Ετήσια πρόοδος
  - Τεχνολογία ημιαγωγών
    - 60% περισσότερα στοιχεία/chip
    - 15% ταχύτερα στοιχεία
    - Βραδύτερα καλώδια
  - Μνήμη
    - 60% αύξηση χωρητικότητας
    - 3,3% μείωση του χρόνου πρόσβασης
  - Μαγνητικοί δίσκοι
    - 60% αύξηση χωρητικότητας
    - 3,3% μείωση του χρόνου πρόσβασης
  - Πλακέτες κυκλωμάτων
    - 5% αύξηση στην πυκνότητα καλωδίων
  - Καλώδια
    - καμία αλλαγή



64x περισσότερα στοιχεία από το 1989
4x γρηγορότερα στοιχεία

Χωρητικότητα μικροεπεξεργαστών και μνημών



- Alpha 21264 15εκατομμύρια
- Pentium Pro 5,5εκατομμύρια
- PowerPC 620 6,9εκατομμύρια
- Alpha 21164 9,3εκατομμύρια
- Sparc Ultra 5,2εκατομμύρια

Moore's Law -> 2x transistors/chip κάθε 1,5 χρόνο

Reuters, Δευτέρα 11/6/2001 :

Οι μηχανικοί της Intel σχεδίασαν και κατασκεύασαν το μικρότερο και ταχύτερο transistor στον κόσμο με μέγεθος 0,02 microns. Αυτό ανοίγει το δρόμο για μικροεπεξεργαστές 1 δισεκατομμυρίου transistors, με συχνότητα στα 20GHz το 2007.

DRAM	Chip Capacity	Cycle time
Έτος	Μέγεθος	ταχύτητα
1980	64Kb	250ns
1983	256Kb	220ns
1986	1Mb	190ns
1989	4Mb	165ns
1992	16Mb	145ns
1996	64Mb	120ns
2000	256Mb	100ns
2002	1Gb	??ns

Τάσεις της τεχνολογίας Υπολογιστών: *Ραγδαίες Αλλαγές*

- Processor:
  - 2X in speed every 1.5 years; 1000X performance in last decade.
- Memory:
  - DRAM capacity: > 2x every 1.5 years; 1000X size in last decade.
  - Cost per bit: Improves about 25% per year.
- Disk:
  - Capacity: > 2X in size every 1.5 years.
  - Cost per bit: Improves about 60% per year.
  - 200X size in last decade.
- Expected State-of-the-art PC by end of year 2000 :
  - Processor clock speed: 1500 MegaHertz (1.5 GigaHertz)
  - Memory capacity: 500 MegaByte (0.5 GigaBytes)
  - Disk capacity: 100 GigaBytes (0.1 TeraBytes)

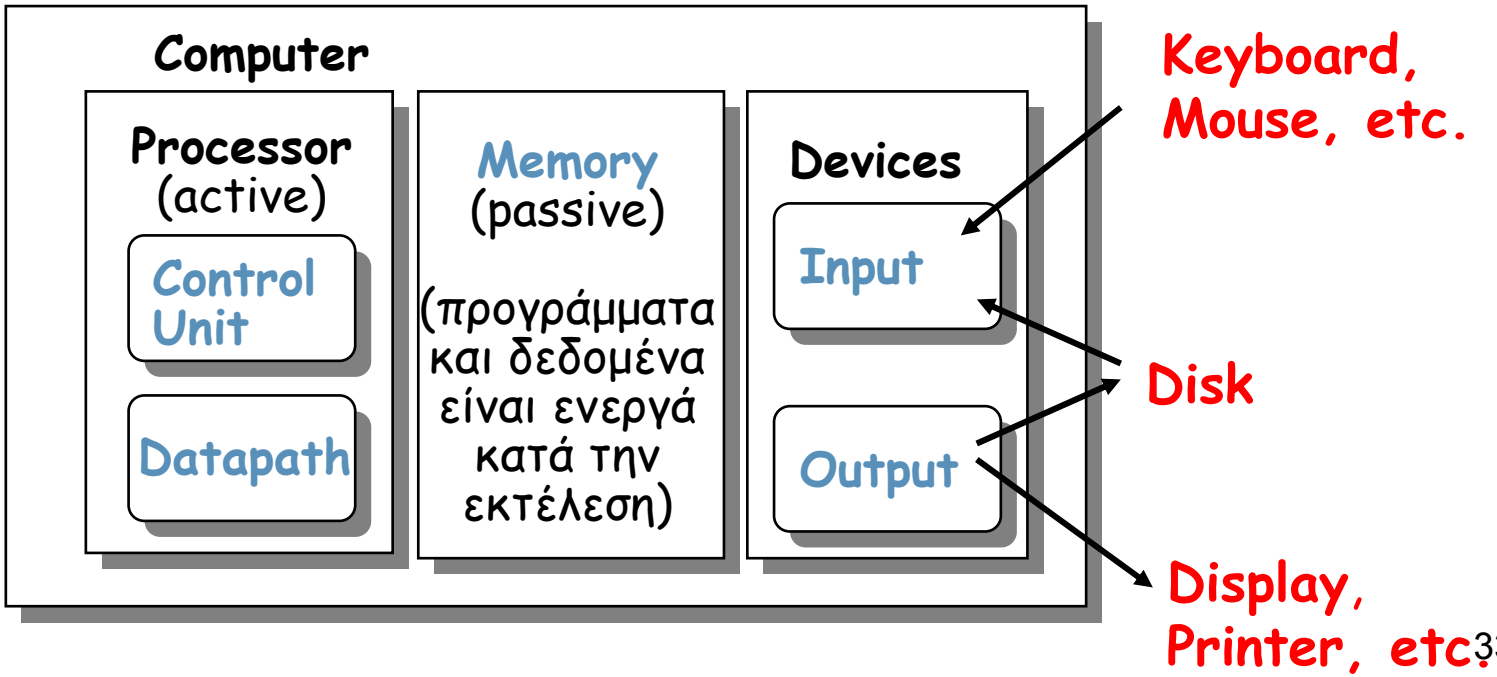
Περιορισμοί από τις Εφαρμογές

- Οι εφαρμογές οδηγούν την «ισορροπία» των μηχανημάτων
  - Αριθμητικοί προσομοιωτές
    - Επίδοση αριθμών κινητής υποδιαστολής
    - Bandwidth κύριας μνήμης
  - Διεκπεραίωση λειτουργιών
    - I/O ανά δευτερόλεπτο
    - Επίδοση της CPU με ακεραίους
  - Έλεγχος ολοκληρωμένων
    - I/O χρονισμός
  - Επεξεργασία μέσων
    - Χαμηλή ακρίβεια στην αριθμητική των pixel

Συστατικά του Hardware

Πέντε είναι τα κλασσικά συστατικά στοιχεία όλων των υπολογιστών:
1. Control Unit; 2. Datapath; 3. Memory; 4. Input; 5. Output

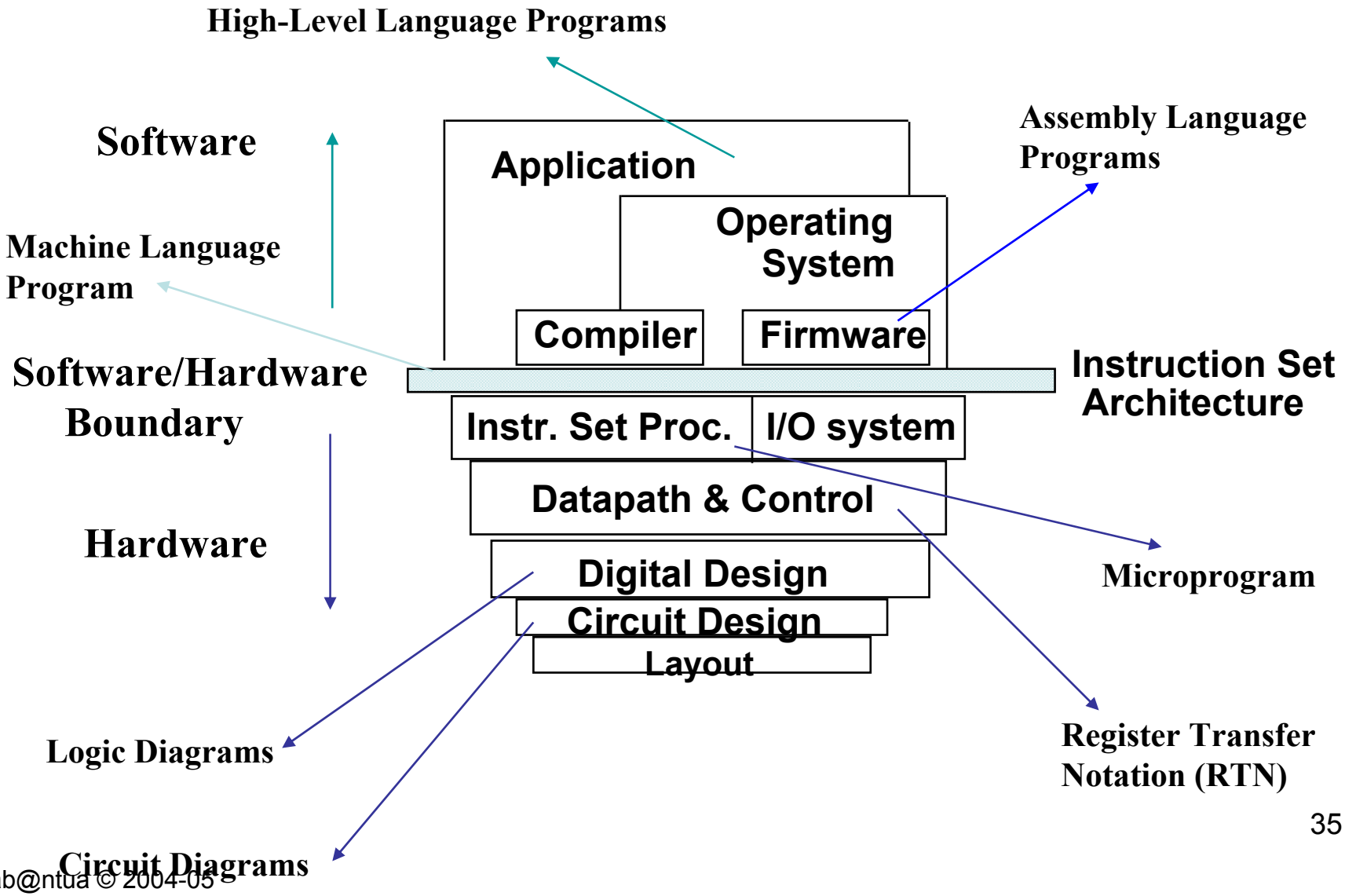
Processor



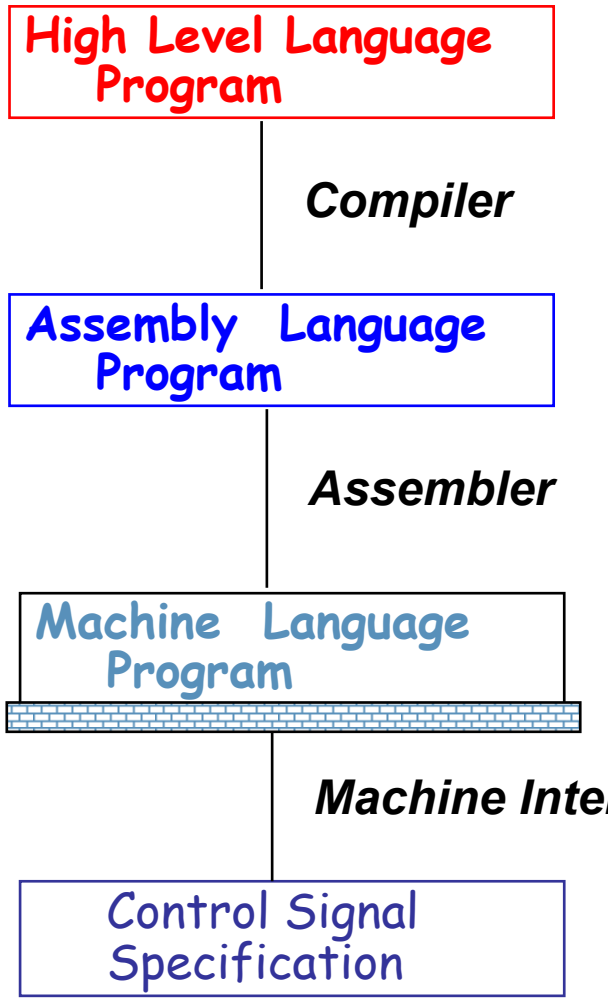
Οργάνωση της CPU

- Σχεδιασμός του Datapath:
  - Δυνατότητες & Επίδοση των χαρακτηριστικών των λειτουργικών μονάδων (FUs):
  - (e.g., Registers, ALU, Shifters, Logic Units, ...)
  - Τρόποι διασύνδεσης των στοιχείων (σύνδεση διαδρόμων, multiplexors, etc.).
  - Πώς ρέει η πληροφορία μεταξύ των στοιχείων του Η/Υ.
- Σχεδιασμός της Μονάδας Ελέγχου (Control Unit):
  - Λογική και μέσα ελέγχου της ροής πληροφορίας.
  - Έλεγχος και συντονισμός της λειτουργίας των λειτουργικών μονάδων (FUs) για την κατανόηση της Αρχιτεκτονικής τουInstruction Set Architecture που σκοπεύουμε να υλοποιήσουμε (υλοποιείται είτε με ένα μηχάνημα πεπερασμένων καταστάσεων (finite state) ή με μικροπρόγραμμα).
- Περιγραφή του Hardware description με μία κατάλληλη γλώσσα, πιθανώς χρησιμοποιώντας (RTN).

Ιεραρχία της Αρχιτεκτονικής Υπολογιστών



Μορφή προγράμματος σε κάθε επίπεδο




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

ALUOP[0:3] <= InstReg[9:11] & MASK

Register Transfer Notation (RTN)

Ιεραρχία του Σχεδιασμού Υπολογιστών

Level	Name	Modules	Primitives	Descriptive Media
1	Electronics	Gates, FF's	Transistors, Resistors, etc.	Circuit Diagrams
2	Logic	Registers, ALU's ...	Gates, FF's	Logic Diagrams
3	Organization	Processors, Memories	Registers, ALU's ...	Register Transfer Notation (RTN)
4	Microprogramming	Assembly Language	Microinstructions	Microprogram
5	Assembly language programming	OS Routines	Assembly language Instructions	Assembly Language Programs
6	Procedural Programming	Applications Drivers ..	OS Routines High-level Languages	High-level Language Programs
7	Application	Systems	Procedural Construct	Problem-Oriented Programs

Low Level - Hardware

Firmware

High Level - Software

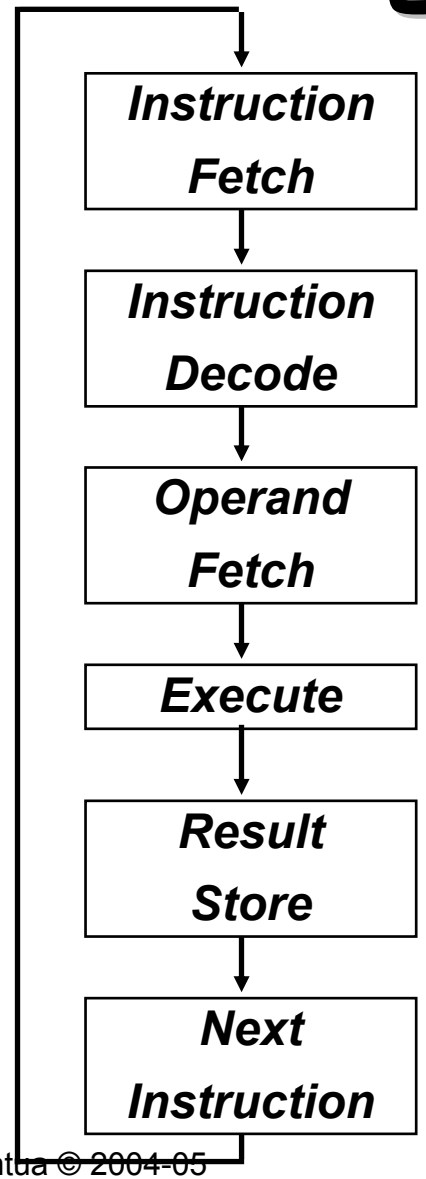
Επεξεργασία του Instruction Set

- **Αρχιτεκτονική (ISA)** – από την πλευρά του προγραμματιστή/μεταγλωτιστή
  - Λειτουργική εμφάνιση προς μέσο χρήστη / προγραμματιστή συστήματος
  - Opcodes, addressing modes, architected registers, IEEE floating point

- **Υλοποίηση (Μarchitecture)** – από την πλευρά του σχεδιαστή επεξεργαστών
  - Λογική δομή και οργάνωση της αρχιτεκτονικής
  - Pipelining, functional units, caches, physical registers

- **Πραγματοποίηση (Chip)** – από την πλευρά του σχεδιαστή chip / συστημάτων
  - Φυσική δομή της υλοποίησης
  - Gates, cells, transistors, wires

CPU Machine Instruction Execution Steps



Πάρε την εντολή από τη θέση αποθήκευσης του προγράμματος

Καθόρισε τις απαιτούμενες ενέργειες και το μέγεθος της εντολής

Εντόπισε και πάρε τα δεδομένα-τελεστές

Υπολόγισε την τιμή του αποτελέσματος ή της κατάστασης

Αποθήκευσε τα αποτελέσματα για μεταγενέστερη χρήση

Καθόρισε την επόμενη εντολή

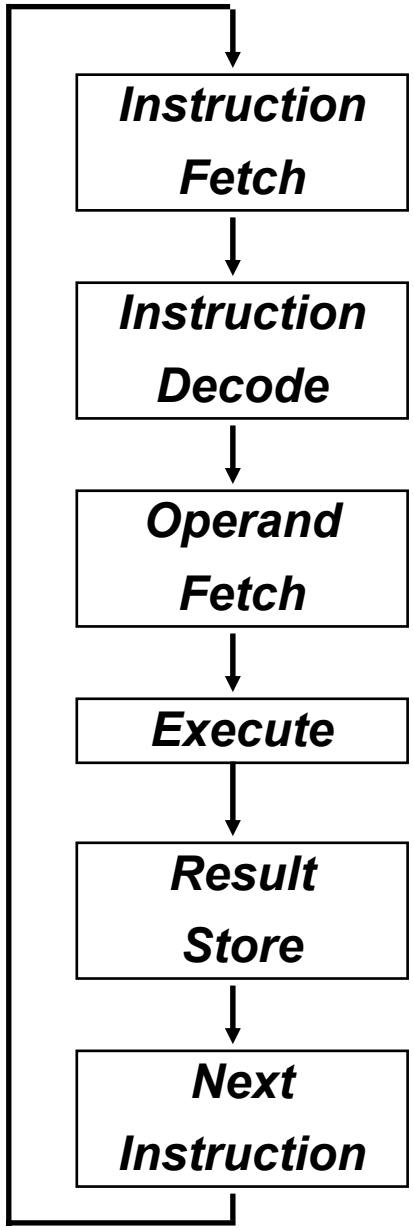
Instruction Set Architecture (ISA)

"... τα χαρακτηριστικά ενός [υπολογιστικού] συστήματος όπως φαίνεται από την πλευρά του προγραμματιστή, *π.χ.* η ιδεατή δομή και η λειτουργική συμπεριφορά, διαχωρισμένα από την οργάνωση της ροής δεδομένων και τους ελέγχους του λογικού σχεδιασμού και της φυσικής υλοποίησης (as distinct from the organization of the data flows and controls the logic design, and the physical implementation)."
– Amdahl, Blaaw, and Brooks, 1964.

- **Η αρχιτεκτονική του συνόλου των εντολών (instruction set architecture) ασχολείται με:**
  - **Οργάνωση της προγραμματιζόμενης αποθήκευσης (memory & registers):**
    - **Συμπεριλαμβάνει το ποσό της διευθυνσιοδοτημένης μνήμης (addressable memory) και τον αριθμό των διαθέσιμων καταχωρητών (registers).**
  - **Τύποι & Δομές Δεδομένων: Κωδικοποιήσεις & παρουσίαση (representations).**
  - **Σύνολο Εντολών (Instruction Set): Ποιες λειτουργίες προσδιορίζονται.**
  - **Μορφοποίηση και κωδικοποίηση Εντολών.**
  - **Τρόποι διευθυνσιοδότησης και προσπέλασης δεδομένων και εντολών**
  - **Χειρισμός Εξαιρέσεων.**

Computer Instruction Sets

- Ανεξάρτητα από τον τύπο του υπολογιστή, τη δομή της CPU, ή την οργάνωση του hardware, κάθε εντολή μηχανής πρέπει να προσδιορίζει τα ακόλουθα:
  - Opcode: Ποια εντολή εκτελείται. Παράδειγμα: add, load και branch.
  - Πού βρίσκονται οι τελεστές, αν υπάρχουν: Οι τελεστές μπορεί να είναι αποθηκευμένοι σε καταχωρητές της CPU, στην κύρια μνήμη, ή σε θύρες εισόδου/εξόδου.
  - Πού τοποθετείται το αποτέλεσμα, αν υπάρχει: Μπορεί να αναφέρεται ρητά ή να υπονοείται από τον κωδικό της εντολής (opcode).
  - Πού βρίσκεται η επόμενη εντολή: Αν δεν υπάρχουν ρητές διακλαδώσεις (branches), η προς εκτέλεση εντολή είναι η επόμενη στην ακολουθία εντολών του προγράμματος. Σε περίπτωση εντολών jump ή branch η διεύθυνση προσδιορίζεται από αυτές.



Instruction Set Architecture (ISA)

Προδιαγραφή Απαιτήσεων (Specification Requirements)

- Μορφοποίηση ή Κωδικοποίηση Εντολών:
  - Πώς κωδικοποιείται;
- Θέση τελεστών και αποτελέσματος (addressing modes):
  - Πού αλλού εκτός μνήμης;
  - Πόσοι ρητοί τελεστές;
  - Πώς αντιστοιχίζονται (located) οι τελεστές μνήμης;
  - Ποιοι μπορούν να βρίσκονται στη μνήμη και ποιοι όχι;
- Τύποι και μέγεθος δεδομένων.
- Πράξεις
  - Ποιες υποστηρίζονται
- Διαδοχή εντολών:
  - Jumps, conditions, branches.
- Fetch-decode-execute υπονοούνται.

Τύποι Εντολών στο Instruction Set

Operator Type	Παραδείγματα
Arithmetic and logical	Integer arithmetic & logical operations: add, or
Data transfer	Loads-stores (move on machines with memory addressing)
Control	Branch, jump, procedure call, & return, traps.
System	Operating system call, virtual memory management instructions
Floating point	Floating point operations: add, multiply.
Decimal	Decimal add, decimal multiply, decimal to character conversion
String	String move, string compare, string search
Graphics	Pixel operations, compression/ decompression operations

Παραδείγματα Εντολών μετακίνησης δεδομένων

Instruction	Meaning	Machine
MOV A,B	Move 16-bit data from memory loc. A to loc. B	VAX11
lwz R3,A	Move 32-bit data from memory loc. A to register R3	PPC601
li $3,455	Load the 32-bit integer 455 into register $3	MIPS R3000
MOV AX,BX	Move 16-bit data from register BX into register AX	Intel X86
LEA.L (A0),A2	Load the address pointed to by A0 into A2	MC68000

Παραδείγματα Εντολών της ALU

Instruction	Meaning	Machine
MULF A,B,C	Multiply the 32-bit floating point values at mem. locations A and B, and store result in loc. C	VAX11
nabs r3,r1	Store the negative absolute value of register r1 in r2	PPC601
ori $2,$1,255	Store the logical OR of register $1 with 255 into $2	MIPS R3000
SHL AX,4	Shift the 16-bit value in register AX left by 4 bits	Intel X86
ADD.L D0,D1	Add the 32-bit values in registers D0, D1 and store the result in register D0	MC68000

Παραδείγματα Εντολών Διακλάδωσης

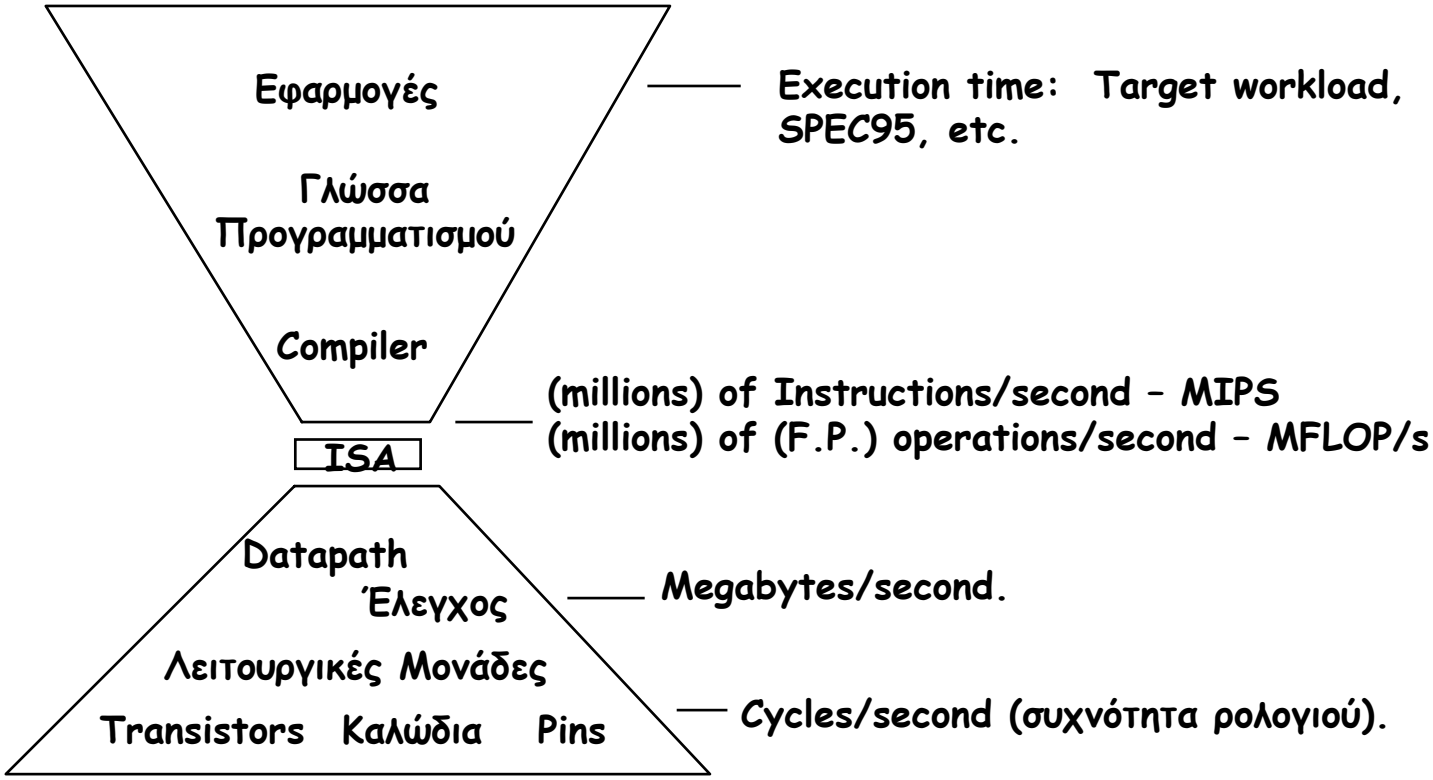
Instruction	Meaning	Machine
BLBS A, Tgt	Branch to address Tgt if the least significant bit at location A is set.	VAX11
bun r2	Branch to location in r2 if the previous comparison signaled that one or more values was not a number.	PPC601
Beq $2,$1,32	Branch to location PC+4+32 if contents of $1 and $2 are equal.	MIPS R3000
JCXZ Addr	Jump to Addr if contents of register CX = 0.	Intel X86
BVS next	Branch to next if overflow flag in CC is set.	MC68000

Παράδειγμα Χρήσης Εντολών: Top 10 Intel X86 Instructions

Κατηγορία	Εντολή	Μέσο ποσοστό συνολικής εκτέλεσης
1	load	22%
2	conditional branch	20%
3	compare	16%
4	store	12%
5	add	8%
6	and	6%
7	sub	5%
8	move register-register	4%
9	call	1%
10	return	1%
	Total	96%

Παρατήρηση: Οι απλές εντολές έχουν τις μεγαλύτερες συχνότητες χρησιμοποίησης.

Μετρικές της επίδοσης Υπολογιστών



Κάθε μέτρο έχει έναν σκοπό και καθένα μπορεί να χρησιμοποιηθεί λανθασμένα
Each metric has a purpose, and each can be misused.

Αξιολόγηση της Επίδοσης των Υπολογιστών: Cycles Per Instruction (CPI)

- Οι περισσότεροι υπολογιστές «τρέχουν» κατά τρόπο σύγχρονο, δηλαδή ένας κύκλο της μονάδας επεξεργασίας (CPU clock) εκτελείται σε συγκεκριμένη (σταθερή) συχνότητα ρολογιού (clock rate):

  όπου: Clock rate = 1 / clock cycle

- Μία εντολή μηχανής αποτελείται από έναν αριθμό μικρολειτουργιών οι οποίες ποικίλουν σε αριθμό και πολυπλοκότητα ανάλογα με την εντολή και την ακριβή οργάνωση και υλοποίηση της μονάδας επεξεργασίας (CPU).
  - Μία μικρολειτουργία είναι μία στοιχειώδης λειτουργία του hardware η οποία μπορεί να εκτελεστεί σε έναν κύκλο ρολογιού.
  - Αυτό αντιπροσωπεύει μία μικρο-εντολή σε μικροπρογραμματιζόμενες CPUs.
  - Παραδείγματα: register operations: shift, load, clear, increment, ALU operations: add , subtract, etc.
- Επομένως μία απλή εντολή μηχανής μπορεί να πάρει έναν ή περισσότερους κύκλους για να ολοκληρωθεί. Ο αριθμός των απαιτούμενων κύκλων ονομάζεται Cycles Per Instruction (CPI).

Μετρήσεις της Επίδοσης Υπολογιστών: Χρόνος εκτέλεσης προγράμματος

- Για ένα συγκεκριμένο πρόγραμμα που έχει μεταγλωττιστεί για να εκτελείται σε ένα συγκεκριμένο μηχάνημα "Α", δίνονται οι ακόλουθες παράμετροι:
  - Ο συνολικός αριθμός εντολών του προγράμματος.
  - Ο μέσος αριθμός των κύκλων ανά εντολή (average CPI).
  - Ο κύκλος ρολογιού του μηχανήματος "Α"
- Πώς μπορεί κανείς να μετρήσει την επίδοση του μηχανήματος κατά την εκτέλεση ου προγράμματος αυτού;
  - Διαισθητικά το μηχάνημα φαίνεται να έχει καλύτερη επίδοση όσο μικρότερος είναι ο συνολικός χρόνος εκτέλεσης.
  - Επομένως το αντίστροφο του συνολικού μετρούμενου χρόνου εκτέλεσης (execution time) είναι ένα πιθανό μέτρο της επίδοσης (performance):

$$\text{Performance}_A = 1 / \text{Execution Time}_A$$

Πώς συγκρίνονται διαφορετικά μηχανήματα;

Ποιοι παράγοντες επηρεάζουν την επίδοση; Πώς μπορεί να βελτιωθεί η επίδοση;

Comparing Computer Performance Using Execution Time

- Για να συγκρίνουμε την επίδοση 2 μηχανημάτων "A", "B" τα οποία εκτελούν ένα δεδομένο πρόγραμμα:

  $\text{Performance}_A = 1 / \text{Execution Time}_A$

  $\text{Performance}_B = 1 / \text{Execution Time}_B$

- Αν το μηχάνημα A είναι n φορές γρηγορότερο από το μηχάνημα B σημαίνει ότι:

  $n = \text{Performance}_A / \text{Performance}_B = \text{Execution Time}_B / \text{Execution Time}_A$

- Παράδειγμα:

  Για ένα δεδομένο πρόγραμμα :

  Execution time στο A: $\text{Execution}_A = 1$ second

  Execution time στο B: $\text{Execution}_B = 10$ seconds

  $\text{Performance}_A / \text{Performance}_B = \text{Execution Time}_B / \text{Execution Time}_A$

  $= 10 / 1 = 10$

  Η επίδοση του A είναι 10 φορές καλύτερη από την επίδοση του B όταν τρέχει το συγκεκριμένο πρόγραμμα

CPU Execution Time

- Ένα πρόγραμμα αποτελείται από έναν αριθμό εντολών
  - Μετρούμενες σε: instructions/program
- Για την ολοκλήρωση μιας μέσης εντολής απαιτείται ένας αριθμός κύκλων ανά εντολή (cycles per instruction - CPI).
  - Μετρούμενος σε: cycles/instruction
- Η CPU έχει σταθερό χρόνο κύκλου ρολογιού = 1/clock rate
  - Μετρούμενο σε: seconds/cycle
- Ο χρόνος εκτέλεσης από τη CPU (CPU execution time) είναι το γινόμενο των 3 παραπάνω παραμέτρων:

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

CPU Execution Time: Παράδειγμα

- Ένα πρόγραμμα εκτελείται σε ένα συγκεκριμένο μηχάνημα που έχει τις ακόλουθες παραμέτρους:
  - Συνολικός αριθμός εντολών(instruction count): 10,000,000 instructions
  - Μέσο CPI του προγράμματος: 2.5 cycles/instruction.
  - Συχνότητα ρολογιού της CPU: 200 MHz.
- Ποιος είναι ο χρόνος εκτέλεσης (execution time) του προγράμματος:

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

CPU time = Instruction count x CPI x Clock cycle
= 10,000,000 x 2.5 x 1 / clock rate
= 10,000,000 x 2.5 x 5×10^{-9}
= 0.125 seconds

Παράγοντες που επηρεάζουν την επίδοση της CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

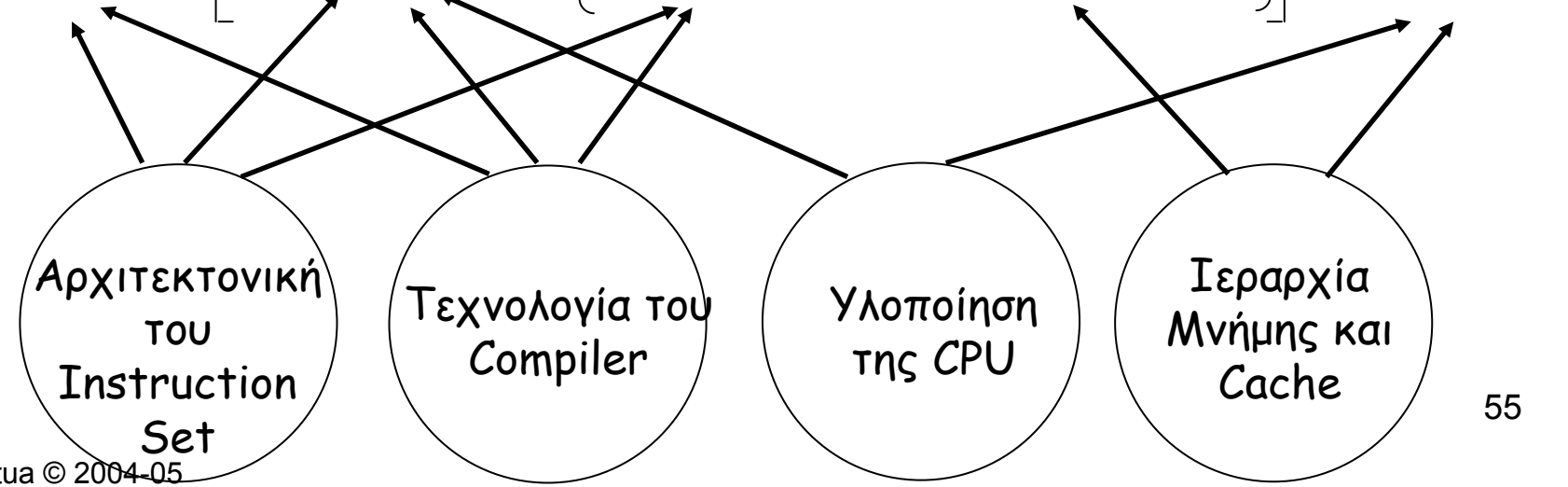
	Instruction Count	CPI	Clock Rate
Program	x		
Compiler	x	x	
Instruction Set Architecture (ISA)	x	x	x
Οργάνωση		x	x
Τεχνολογία			x

Χρόνος εκτέλεσης

$$\text{χρόνος εκτέλεσης} = \text{αριθμός εντολών} \times \frac{\text{κύκλοι}}{\text{εντολή}} \times \text{χρόνος 1 κύκλου}$$

$$= \text{αριθμός εντολών} \times \left[\frac{\text{κύκλοι CPU}}{\text{εντολή}} + \frac{\text{κύκλοι μνήμης}}{\text{εντολή}} \right] \times \text{χρόνος 1 κύκλου}$$

$$= \text{αριθμός εντολών} \times \left[\frac{\text{κύκλοι CPU}}{\text{εντολή}} + \left[\frac{\text{αναφορές}}{\text{εντολή}} \times \frac{\text{κύκλοι μνήμης}}{\text{αναφορά}} \right] \right] \times \text{χρόνος 1 κύκλου}$$



Μέτρα της Επίδοσης Υπολογιστών : MIPS (Million Instructions Per Second)

- Για ένα συγκεκριμένο πρόγραμμα που εκτελείται σε ένα συγκεκριμένο υπολογιστή, MIPS (millions of instructions per second) είναι ένα μέτρο του πόσα εκατομμύρια εντολές εκτελούνται το δευτερόλεπτο:

$$\begin{aligned} MIPS &= \text{Instruction count} \;/\; (\text{Execution Time} \times 10^6) \\ &= \text{Instruction count} \;/\; (\text{CPU clocks} \times \text{Cycle time} \times 10^6) \\ &= (\text{Instruction count} \times \text{Clock rate}) \;/\; (\text{Instruction count} \times \text{CPI} \times 10^6) \\ &= \text{Clock rate} \;/\; (\text{CPI} \times 10^6) \end{aligned}$$

- Γρηγορότερος χρόνος εκτέλεσης συνήθως σημαίνει γρηγορότερο MIPS:
  - Δε χρησιμοποιείται ο αριθμός των εντολών (No account for the instruction set used).
  - Εξάρτηση από το πρόγραμμα: Κάθε μηχάνημα δεν έχει μόνο ένα MIPS γιατί η εκτίμηση του MIPS εξαρτάται από το χρησιμοποιούμενο πρόγραμμα.
  - Easy to abuse: Program used to get the MIPS rating is often omitted.
  - Cannot be used to compare computers with different instruction sets.
  - A higher MIPS rating in some cases may not mean higher performance or better execution time. i.e. due to compiler design variations.

Μέτρα της Επίδοσης Υπολογιστών : MFOLPS (Million FLOating-Point Operations Per Second)

- Μία πράξη floating-point είναι πρόσθεση, αφαίρεση, πολ/σμός ή διαίρεση μεταξύ αριθμών με παράσταση απλής ή διπλής ακρίβειας floating-point.
- MFLOPS, για δεδομένο πρόγραμμα που εκτελείται σε δεδομένο υπολογιστή, είναι ένα μέτρο των εκατομμυρίων των πράξεων με floating point (megaflops) ανά δευτερόλεπτο:

MFLOPS = αριθμός των πράξεων floating-point / (Execution time x 10^6)

- MFLOPS είναι ένα καλύτερο μέτρο σύγκρισης μεταξύ διαφορετικών μηχανημάτων από ότι το MIPS.
- Εξάρτηση από το πρόγραμμα: Διαφορετικά προγράμματα έχουν διαφορετικά ποσοστά εντολών με floating-point πράξεις. π.χ. Οι compilers δεν περιέχουν πράξεις με floating- point και δίνουν μηδενικό MFLOPS.
- Εξαρτάται από το είδος της πράξης με floating-point που πραγματοποιείται στο πρόγραμμα.

Choosing Programs To Evaluate Performance

Levels of programs or benchmarks that could be used to evaluate performance:

- Actual Target Workload: Full applications that run on the target machine.
- Real Full Program-based Benchmarks:
  - Select a specific mix or suite of programs that are typical of targeted applications or workload (e.g SPEC95).
- Small "Kernel" Benchmarks:
  - Key computationally-intensive pieces extracted from real programs.
    - Examples: Matrix factorization, FFT, tree search, etc.
  - Best used to test specific aspects of the machine.
- Microbenchmarks:
  - Small, specially written programs to isolate a specific aspect of performance characteristics: Processing: integer, floating point, local memory, input/output, etc.

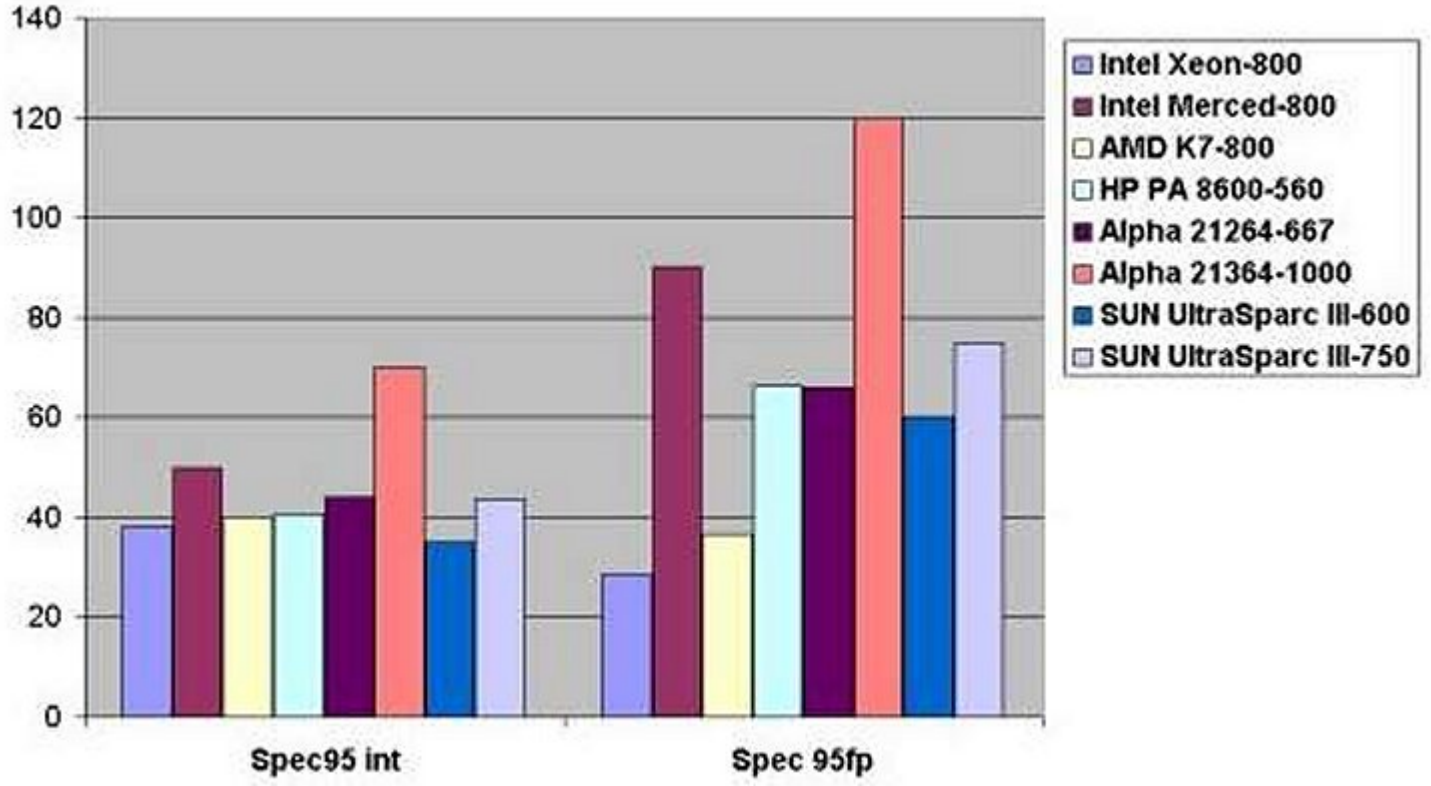
Types of Benchmarks

Pros		Cons
• Representative	Actual Target Workload	• Very specific. • Non-portable. • Complex: Difficult to run, or measure.
• Portable. • Widely used. • Measurements useful in reality.	Full Application Benchmarks	• Less representative than actual workload.
• Easy to run, early in the design cycle.	Small "Kernel" Benchmarks	• Easy to "fool" by designing hardware to run them well.
• Identify peak performance and potential bottlenecks.	Microbenchmarks	• Peak performance results may be a long way from real application performance

SPEC: System Performance Evaluation Cooperative

- The most popular and industry-standard set of CPU benchmarks.
- SPECmarks, 1989:
  - 10 programs yielding a single number ("SPECmarks").
- SPEC92, 1992:
  - SPECInt92 (6 integer programs) and SPECfp92 (14 floating point programs).
- SPEC95, 1995:
  - Eighteen new application benchmarks selected (with given inputs) reflecting a technical computing workload.
  - SPECint95 (8 integer programs):
    - go, m88ksim, gcc, compress, li, ijpeg, perl, vortex
  - SPECfp95 (10 floating-point intensive programs):
    - tomcatv, swim, su2cor, hydro2d, mgrid, applu, turb3d, apsi, fppp, wave5
  - Source code must be compiled with standard compiler flags.

SPEC95 For High-End CPUs First Quarter 2000



Σύγκριση Επίδοσης: Example

- Από το προηγούμενο παράδειγμα: Ένα πρόγραμμα εκτελείται σε συγκεκριμένο μηχάνημα με τις ακόλουθες παραμέτρους:
  - Συνολικός αριθμός εντολών: 10,000,000 instructions
  - Μέσο CPI του προγράμματος: 2.5 cycles/instruction.
  - Συχνότητα ρολογιού της CPU: 200 MHz.
- Χρησιμοποιούμε το ίδιο πρόγραμμα με τις εξής αλλαγές:
  - Νέος compiler: Νέος αριθμός εντολών 9,500,000
    Νέο CPI: 3.0
  - Γρηγορότερη υλοποίηση της CPU: Νέα συχνότητα ρολογιού = 300 MHZ
- Ποια είναι η επιτάχυνση (Speedup) με τις αλλαγές αυτές;

$$\text{Speedup} = \frac{\text{Παλιό Execution Time}}{\text{Νέο Execution Time}} = \frac{I_{\text{παλιό}} \times \text{CPI}_{\text{παλιό}} \times \text{Clock cycle}_{\text{old}}}{I_{\text{νέο}} \times \text{CPI}_{\text{νέο}} \times \text{Clock Cycle}_{\text{νέο}}}$$

Speedup = $(10{,}000{,}000 \times 2.5 \times 5\times10^{-9}) / (9{,}500{,}000 \times 3 \times 3.33\times10^{-9})$
= 0.125 / 0.095 = 1.32
ή 32 % γρηγορότερη μετά τις αλλαγές.

Τύποι Εντολών & CPI

- Δίνεται ένα προγράμματος με *n* τύπους ή κατηγορίες εντολών με τα ακόλουθα χαρακτηριστικά:

  C_i = Αριθμός εντολών τύπου$_i$

  CPI_i = Μέσος αριθμός κύκλων/εντολή τύπου$_i$

  Τότε:

$$CPU\ \ clock\ \ cycles\ \ = \sum_{i=1}^{n} (CPI\ \ _i \times C\ _i)$$

Τύποι εντολών & CPI: Παράδειγμα

- Σε Instruction Set με 3 κατηγορίες εντολών:

Κατηγορία εντολής	CPI
A	1
B	2
C	3

- 2 ακολουθίες προγραμμάτων με τον ακόλουθο αριθμό εντολών:

	Αριθμός εντολών για κάθε κατηγορία εντολής		
Code Sequence	A	B	C
1	2	1	2
2	4	1	1

- CPU cycles στο πρόγραμμα 1 = 2 x 1 + 1 x 2 + 2 x 3 = 10 cycles
  CPI στο πρόγραμμα 1 = clock cycles / instruction count
  = 10 /5 = 2
- CPU cycles στο πρόγραμμα 2 = 4 x 1 + 1 x 2 + 1 x 3 = 9 cycles
  CPI στο πρόγραμμα 2 = 9 / 6 = 1.5

Συχνότητα Εντολών & CPI

- Δίνεται ένα προγράμματος με *n* τύπους ή κατηγορίες εντολών με τα ακόλουθα χαρακτηριστικά :

  C_i = Αριθμός εντολών τύπου$_i$

  CPI_i = Μέσος αριθμός κύκλων/εντολή τύπου$_i$

  F_i = Συχνότητα της εντολής τύπου$_i$

  = C_i/ συνολικός αριθμός εντολών

  Τότε:

$$CPI = \sum_{i=1}^{n} (CPI_i \times F_i)$$

Συχνότητα Εντολών & CPI : Παράδειγμα σε RISC

Base Machine (Reg / Reg)

Op	Freq	Cycles	CPI(i)	% Time
ALU	50%	1	0.5	23%
Load	20%	5	1.0	45%
Store	10%	3	0.3	14%
Branch	20%	2	0.4	18%

Typical Mix

$$CPI = \sum_{i=1}^{n} (CPI_i \times F_i)$$

CPI = 0.5 x 1 + 0.2 x 5 + 0.1 x 3 + 0.2 x 2 = 2.2

Παραλλαγές Compiler, MIPS & Επίδοση: Παράδειγμα

- Σε Instruction Set με 3 κατηγορίες εντολών :

Κατηγορία εντολής	CPI
A	1
B	2
C	3

- Για δεδομένο πρόγραμμα, 2 compilers παράγουν τους ακόλουθους αριθμούς εντολών:

	Αριθμός εντολών (σε εκατομμύρια) για κάθε κατηγορία εντολών		
Κώδικας από:	A	B	C
Compiler 1	5	1	1
Compiler 2	10	1	1

- Το μηχάνημα έχει συχνότητα 100 MHz.

Παραλλαγές Compiler, MIPS & Επίδοση: Παράδειγμα (Συνεχίζεται)

MIPS = Clock rate / (CPI x 10^6) = 100 MHz / (CPI x 10^6)

CPI = CPU execution cycles / Instructions count

$$CPU\ clock\ cycles = \sum_{i=1}^{n} \left(CPI_i \times C_i \right)$$

CPU time = Instruction count x CPI / Clock rate

- Για τον compiler 1:
  - CPI_1 = (5 x 1 + 1 x 2 + 1 x 3) / (5 + 1 + 1) = 10 / 7 = 1.43
  - MIP_1 = 100 / (1.428 x 10^6) = 70.0
  - CPU $time_1$ = ((5 + 1 + 1) x 10^6 x 1.43) / (100 x 10^6) = 0.10 seconds
- Για τον compiler 2:
  - CPI_2 = (10 x 1 + 1 x 2 + 1 x 3) / (10 + 1 + 1) = 15 / 12 = 1.25
  - MIP_2 = 100 / (1.25 x 10^6) = 80.0
  - CPU $time_2$ = ((10 + 1 + 1) x 10^6 x 1.25) / (100 x 10^6) = 0.15 seconds