Kiến trúc Máy tính

Khoa học & Kỹ thuật Máy tính

Chương 1

Các khái niệm & Công nghệ

BK
TP.HCM

Cuộc cách mạng Máy tính

- Tiến bộ trong Công nghệ: theo cấp số
  - Dựa trên định luật Moore
- Biến các ứng dụng mơ ước trở thành hiện thực
  - Lĩnh vực xe hơi
  - Phone cầm tay
  - Các dự án về Gen
  - World Wide Web
  - Search Engines
- Ngày nay, máy tính hiện hữu khắp nơi

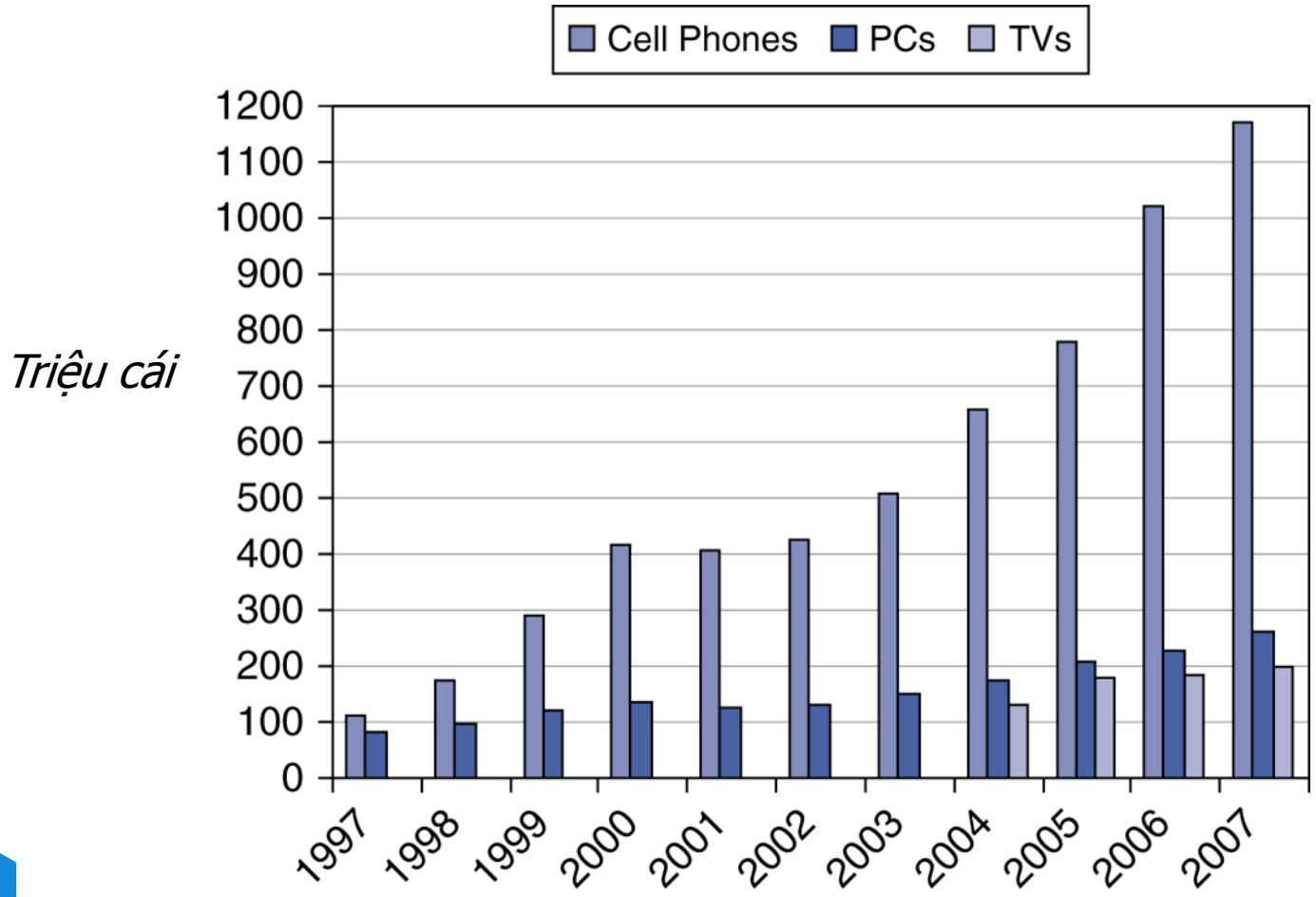
Lịch sử phát triển

- Thế hệ thứ I: 1945 - 1955
  - Đèn chân không, Board mạch
- Thế hệ thứ II: 1955 - 1965
  - transistors, hệ thống bó (IBM máy tính lớn)
- Thế hệ thứ III: 1965 – 1980
  - Mạch tổ hợps & Đa lập trình (Mini, Main Frame)
- Thế hệ thứ IV: 1980 – đến nay
  - personal computers
  - Siêu máy tính, Data Center, Tính toán lưới
  - Máy tính bảng với Điện toán đám mây

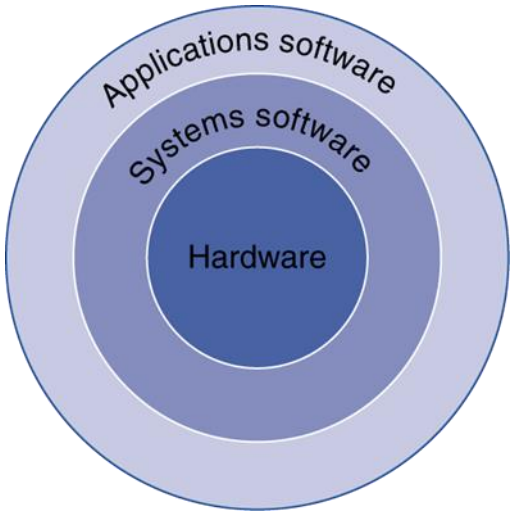
Phân loại Máy tính hiện nay

- **Máy tính để bàn (Desktop Computers)**
  - Đa năng, Đa dạng phần mềm
  - Cân đối theo giá thành/Hiệu suất
- **Máy tính Server (Server Computers)**
  - Môi trường mạng
  - Dung lượng lớn, Hiệu suất cao, Độ tin cậy tốt
  - Đủ loại cấp độ (từ nhỏ đến lớn theo yêu cầu lắp đặt)
- **Máy tính nhúng (Embedded computers)**
  - Tích hợp như là một bộ phận trong các hệ thống
  - Yêu cầu những ràng buộc chặt chẽ về Công suất/Hiệu suất/Giá thành

Thị trường tiêu thụ



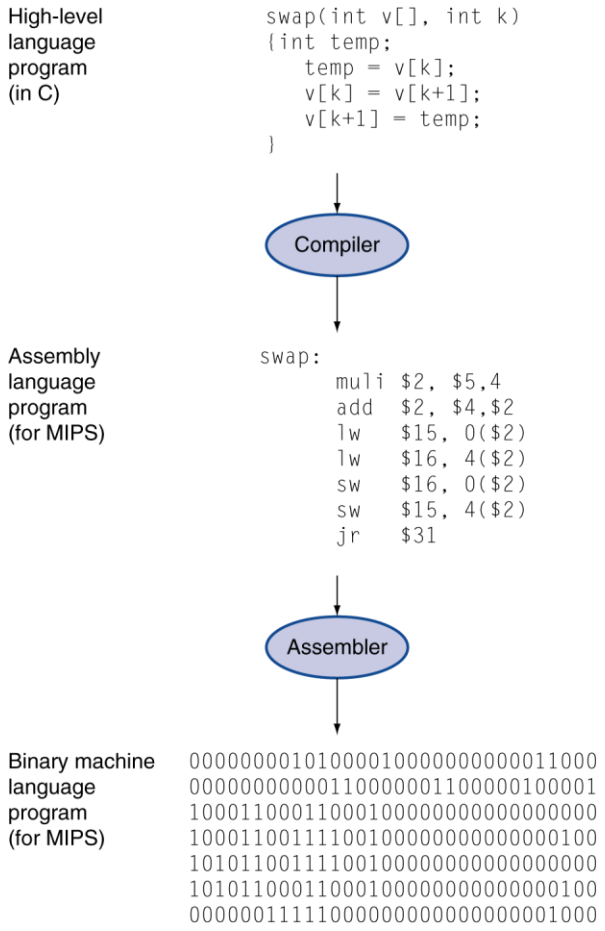
Thực thi chương trình



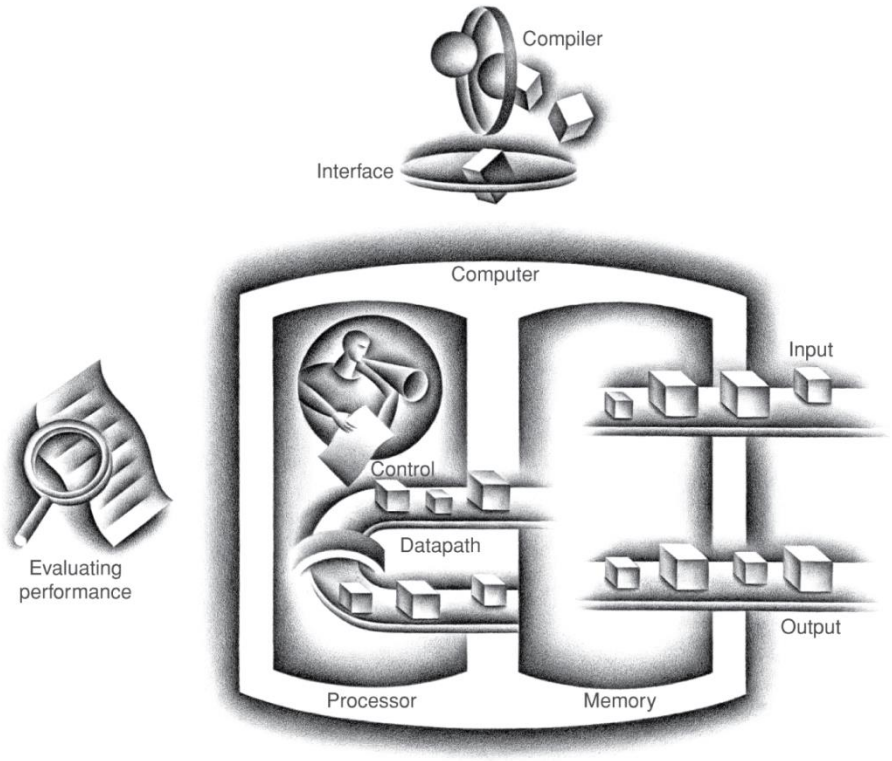
- Phần mềm ứng dụng
  - Ngôn ngữ cấp cao
- Phần mềm hệ thống
  - Biên dịch: Ngôn ngữ cấp cao → Mã máy
  - Hệ điều hành: thực thi dịch vụ
    - Xử lý Xuất/Nhập
    - Quản trị bộ nhớ chính & lưu trữ
    - Định thời công việc & tài nguyên chung
- Phần cứng
  - Bộ Xử lý, Bộ nhớ, Điều khiển Nhập/Xuất

Lộ trình thực hiện lệnh

- Ngôn ngữ cấp cao
  - Cấp độ trìu tượng sát thực với vấn đề
  - Hiệu quả (productivity) & Uyển chuyển (portability)
- Hợp ngữ (Assembly lang.)
  - Các lệnh mã máy trình bày dạng text gợi nhớ
- Biểu diễn bằng phần cứng
  - Số nhị phân (bits)
  - Mã máy lệnh & Dữ liệu



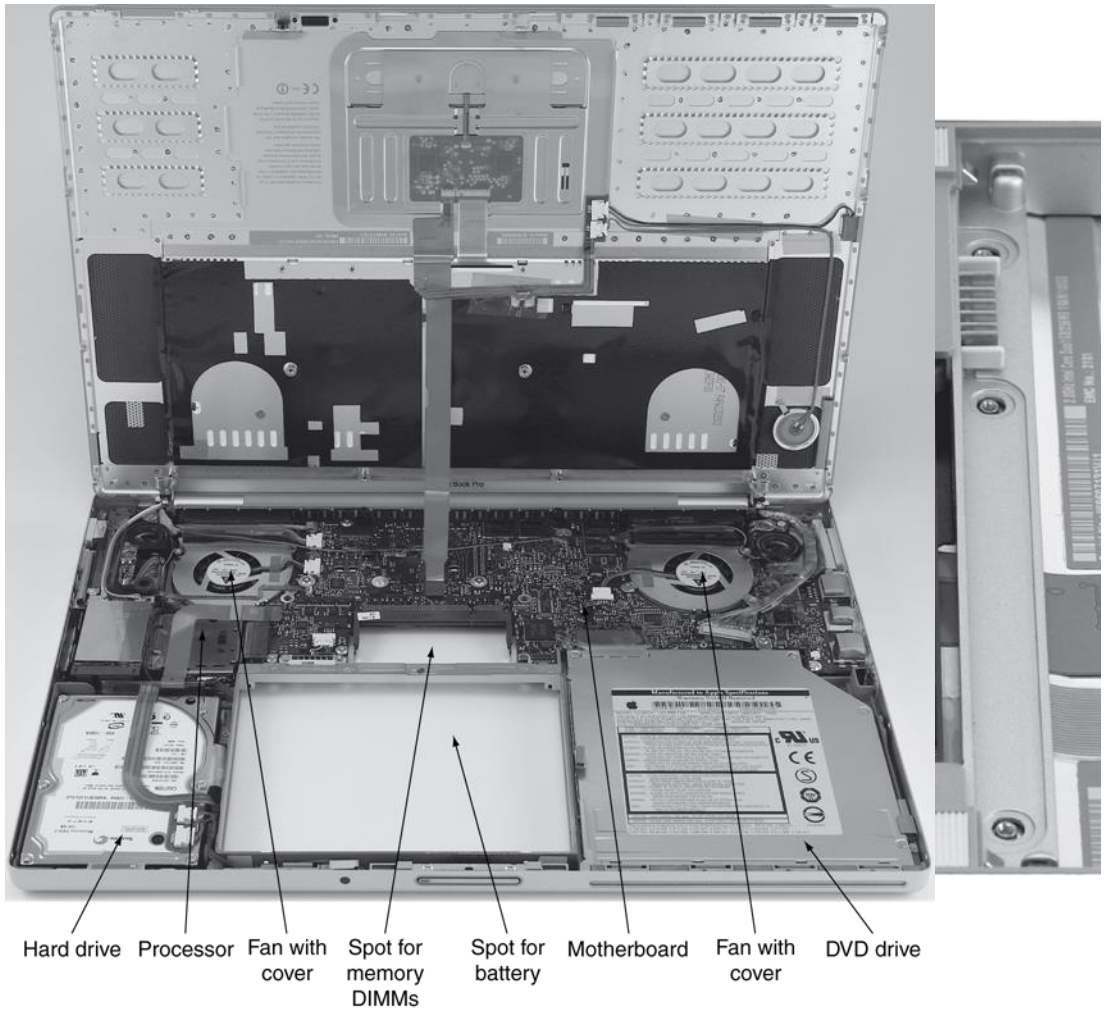
Thành phần chính của máy tính



- Giống nhau cho các loại, bao gồm (5 thành phần):
  - Để bàn, server, nhúng
- Nhập/Xuất bao gồm:
  - Giao tiếp với người dùng
    - Màn hình, bàn phím, chuột
  - Thiết bị lưu trữ
    - Đĩa cứng, CD/DVD, flash
  - Giao tiếp mạng
    - Liên lạc với các máy tính khác

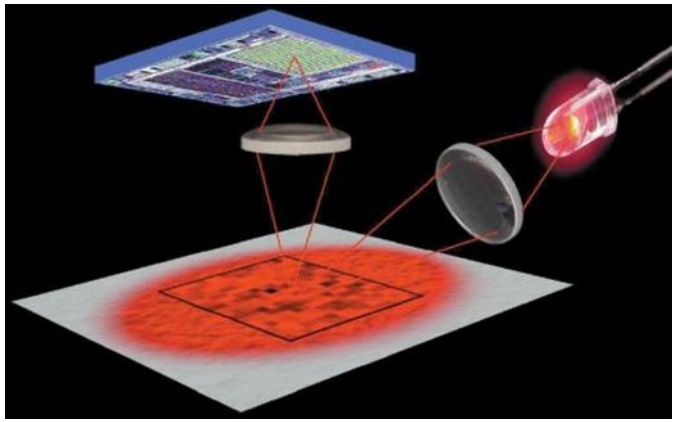
Mổ xẻ bên trong một máy tính

Ví dụ: Laptop


Hard drive
Processor
Fan with cover
Spot for memory DIMMs
Spot for battery
Motherboard
Fan with cover
DVD drive

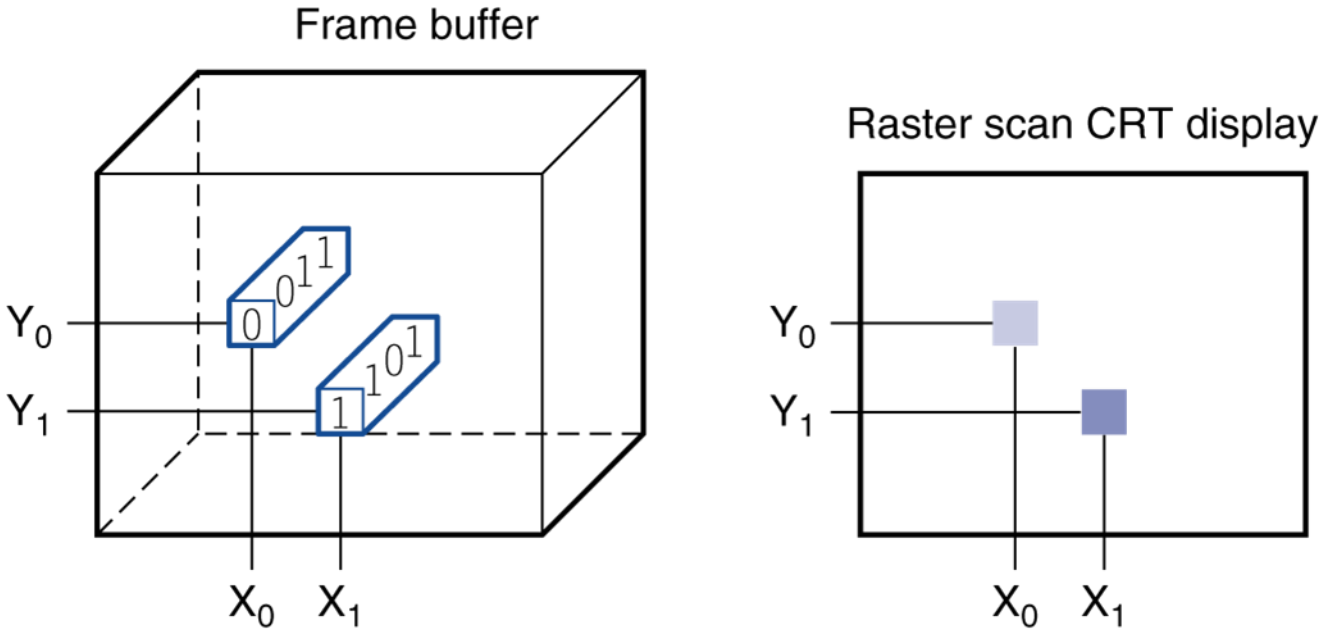
Cơ chế hoạt động của chuột

- Chuột quang
  - Bộ phận phát quang (LED)
  - Camera nhỏ thu hình
  - Bộ xử lý ảnh đơn giản
    - Thu nhận mỗi chuyển động theo trục x, y
  - Nút nhấn & đĩa lỗ phân dải
- Chuột cơ (Supersedes roller-ball)



Thể hiện thông tin trên màn hình

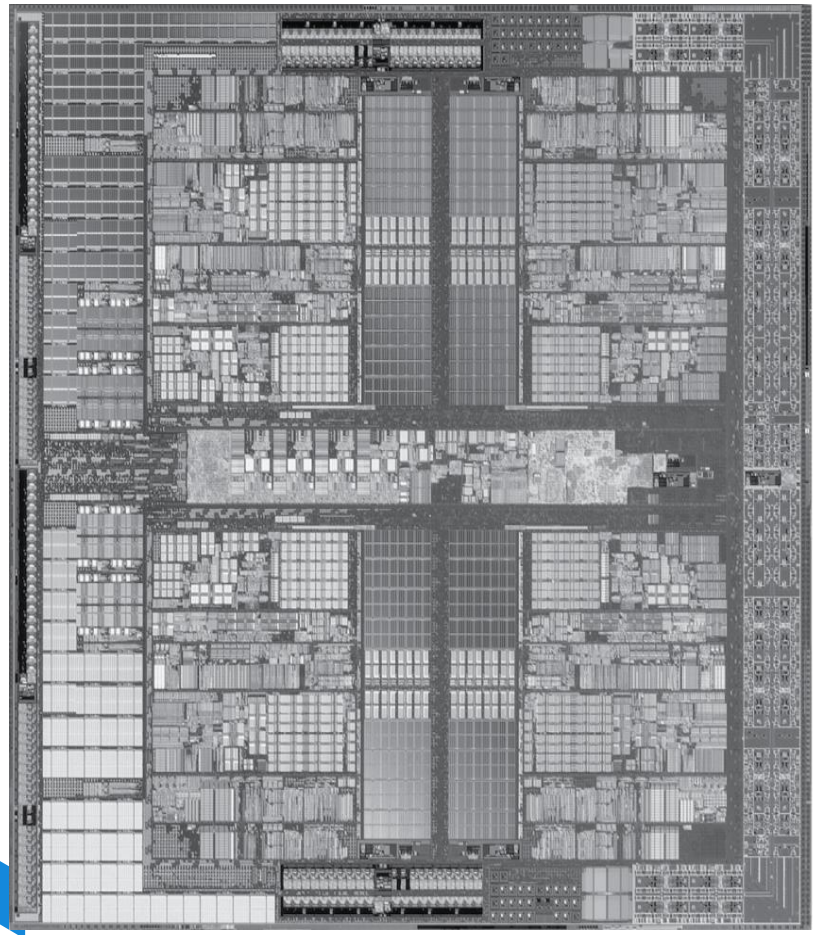
- Màn hình tinh thể lỏng(LCD): nhiều điểm (pixels)
  - Hiển thị 1 khung ảnh chứa trong bộ nhớ

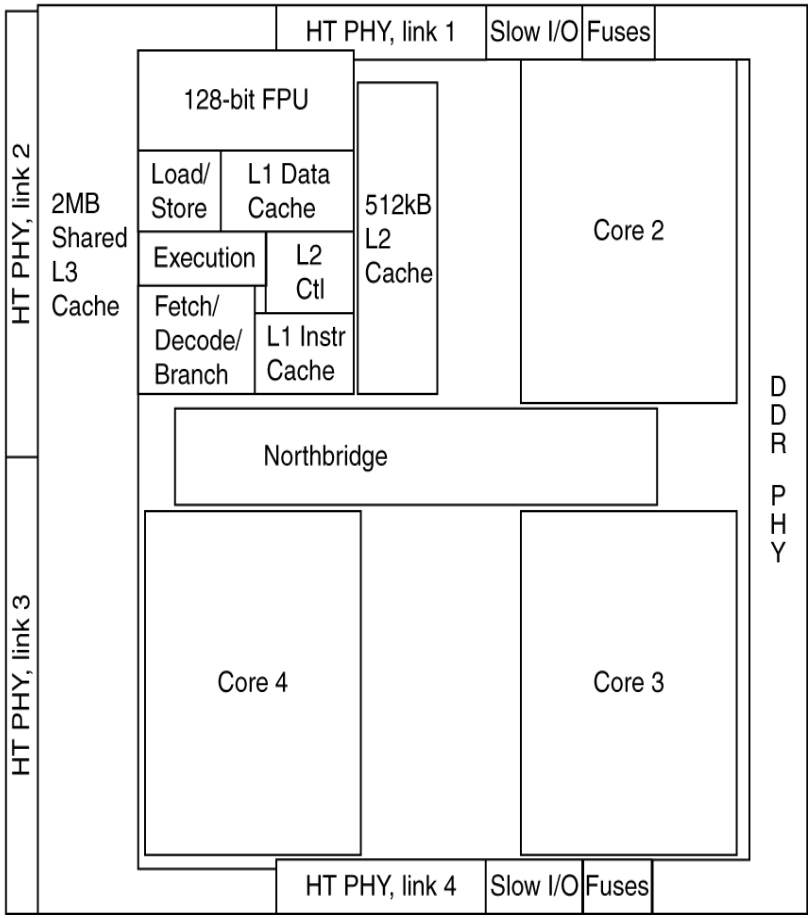


Cấu trúc bên trong Bộ xử lý (CPU)

- Datapath: lộ trình thực hiện các tác vụ với dữ liệu
- Điều khiển: lộ trình thực hiện, bộ nhớ, v.v ...
- Bộ nhớ Cache
  - Một bộ phận bộ nhớ nhỏ nhưng có tốc độ truy xuất nhanh (SRAM), dùng lưu trữ trung gian các dữ liệu trước khi được truy xuất.

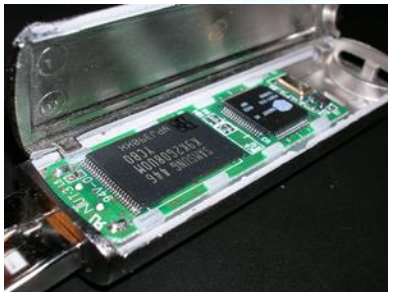
AMD Barcelona: 4 lõi (cores)





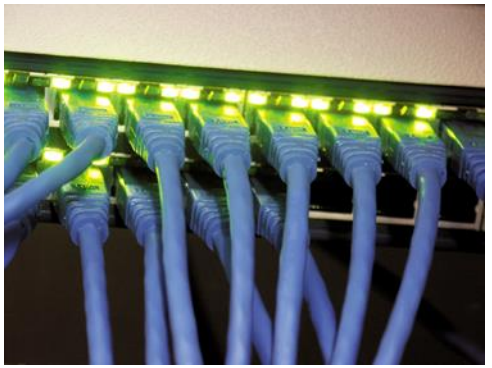
Lưu trữ dữ liệu

- Bộ nhớ chính (volatile)
  - Lưu trữ lệnh và dữ liệu. Thông tin sẽ mất khi tắt nguồn
- Bộ nhớ thứ cấp (Non-volatile)
  - Đĩa cứng (từ)
  - Bộ nhớ flash



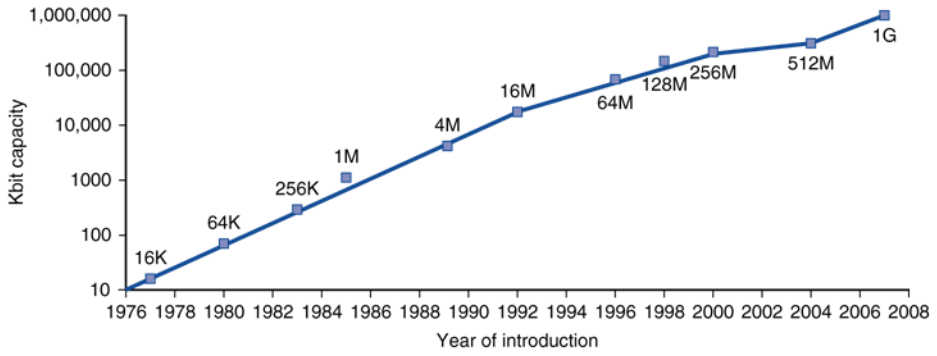
Mạng

- Môi trường liên lạc và chia sẻ tài nguyên
- Mạng cục bộ (LAN): Ethernet
  - Trong cùng văn phòng, tòa nhà, v.v.
- Mạng diện rộng (WAN: the Internet)
- Mạng không dây: WiFi, Bluetooth



Xu hướng theo công nghệ

- Công nghệ điện tử không ngừng phát triển:
  - Tăng dung lượng & Hiệu suất
  - Giảm giá thành



DRAM capacity

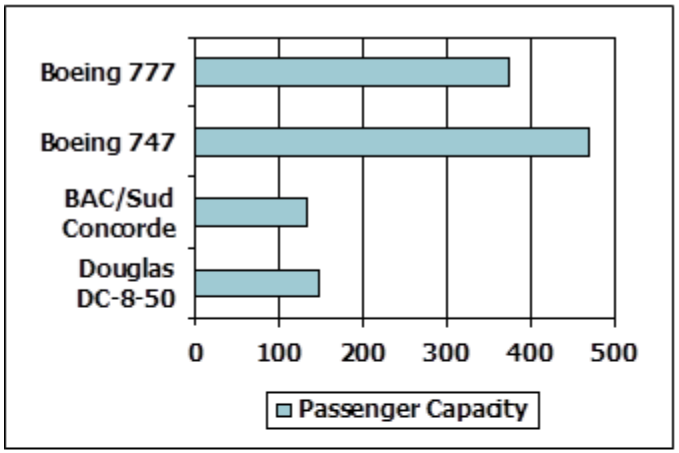
Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

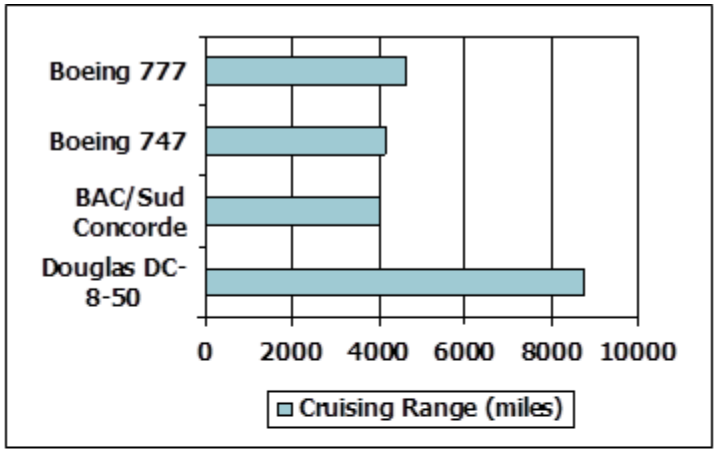
Các khái niệm trìu tượng

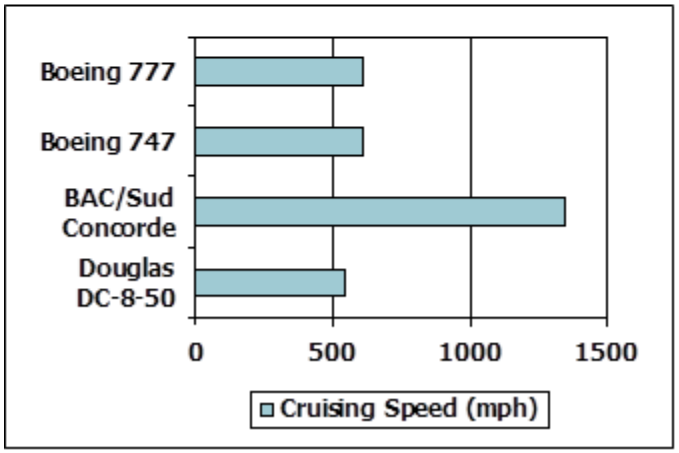
- Abstractions
- Giúp hạn chế độ phức tạp
  - Ẩn những vấn đề chi tiết cấp thấp
- Kiến trúc tập lệnh (ISA = Instruction set architecture)
  - Phần giao giữa Cứng/Mềm
- Giao tiếp ứng dụng
  - (ISA) + Phần mềm hệ thống
- Thực hiện
  - Cụ thể lớp dưới và phần giao tiếp

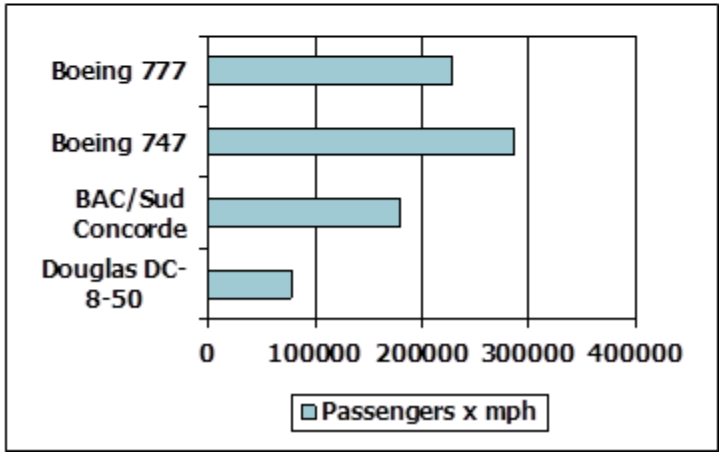
Định nghĩa về Hiệu suất

- Hàng không: loại máy bay nào có hiệu suất tốt nhất?









Hiệu suất hệ thống

- Giải thuật
  - Xác định số tác vụ thực thi (number of operations)
- Ngôn ngữ lập trình, Trình biên dịch, Kiến trúc
  - Xác định số lệnh máy thực thi cho mỗi tác vụ (operation)
- Bộ Xử lý và Hệ thống bộ nhớ
  - Xác định tốc độ xử lý mỗi lệnh máy
- Hệ thống Nhập/Xuất (bao gồm Hệ điều hành)
  - Xác định tốc độ thực thi của mỗi tác vụ I/O

Thời gian đáp ứng & hiệu suất đầu ra

- Thời gian đáp ứng (Response time)
  - Ví dụ: thời gian thực hiện 1 công việc (c.trình)
- Hiệu suất đầu ra (Throughput)
  - Có bao nhiêu tác vụ được thực hiện hoàn tất trong 1 đơn vị thời gianTotal work done per unit time
    - Ví dụ: tasks/transactions/… per hour
- Các thông số trên sẽ bị ảnh hưởng như thế nào? Khi:
  - Thay bộ xử lý có tốc độ nhanh hơn?
  - Thêm bộ xử lý vào hệ thống
- Tập trung vào Thời gian đáp ứng

Hiệu suất: Đại lượng so sánh

- ĐN: Hiệu suất = 1/Thời gian thực thi (Performance = 1/Execution Time)
- "Máy X nhanh hơn máy Y n lần", có nghĩa:

$$\text{Performance}_X / \text{Performance}_Y = \text{Execution time}_Y / \text{Execution time}_X = n$$

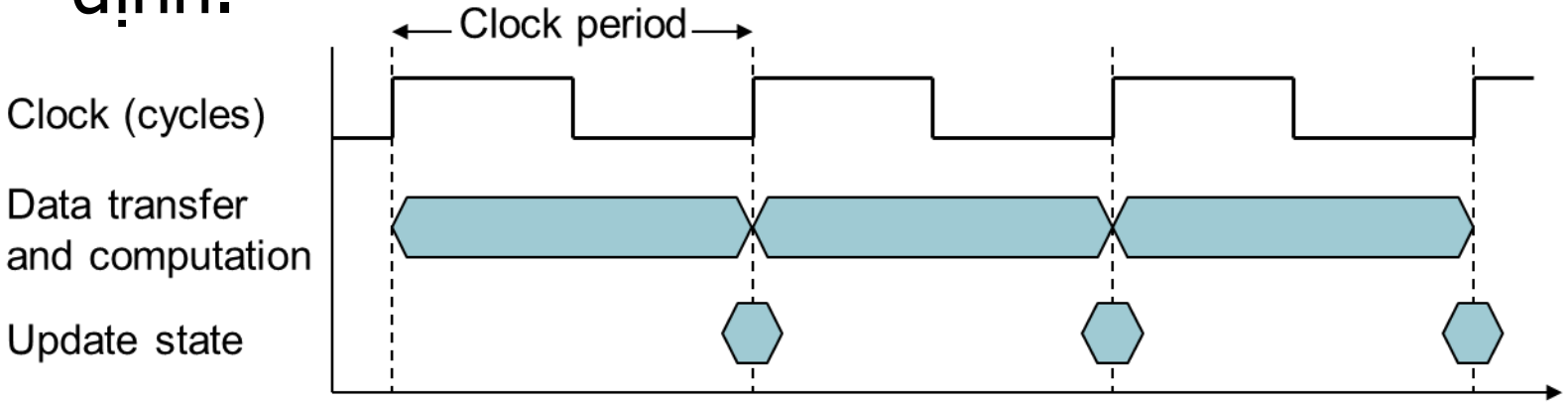
- Ví dụ: thời gian thực thi 1 chương trình
  - Mất 10s trên máy A, 15s trên máy B
  - Execution Time$_B$ / Execution Time$_A$ = 15s / 10s = 1.5
  - Có nghĩa máy A nhanh hơn máy B 1.5 lần

Đo thời gian thực thi

- Thời gian tổng thể (Elapsed time)
  - Thời gian thực thi chương trình, bao gồm:Thời gian xử lý (CPU), Xuất/Nhập, phí tổn HĐH, thời gian chết
  - Thông số xác định hiệu xuất hệ thống
- Thời gian Bộ xử lý (CPU time)
  - Thời gian của CPU xử lý chương trình
    - Không kể thời gian I/O, thời gian do chia sẻ ...
  - Bao gồm thời gian CPU dành cho chương trình người dùng + chương trình hệ thống
  - Các chương trình khác nhau sẽ bị ảnh hưởng khác nhau bởi hiệu suất CPU và hệ thống

Xung đồng hồ Bộ xử lý

- Các tác vụ mạch số (phần cứng) được thực hiện dưới tác dụng của xung đồng hồ có tần số cố định.



  - Chu kỳ đồng hồ: Khoảng thời gian cho 1 chu kỳ, ví dụ: 250ps = 0.25ns = 250×10^{-12}s
  - Tần số (rate): số chu kỳ/mỗi giây,

    Ví dụ: 4.0GHz = 4000MHz = 4.0×10^{9}Hz

Thời gian Bộ Xử lý (CPU Time)

$$\text{CPU Time} = \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$$

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

- Hiệu suất sẽ được cải thiện bằng cách
  - Giảm số chu kỳ CPU
  - Tăng tần số đồng hồ
  - Người thiết kế phần cứng luôn phải hài hòa giữa tần số đồng hồ với số chu kỳ thực hiện

Ví dụ: Thời gian Bộ xử lý

- Máy tính A: 2GHz clock, thực thi mất 10s CPU time
- Thiết kế máy tính B sao cho:
  - Thời gian thực thi chỉ mất 6s CPU time
  - Với đồng hồ nhanh hơn, nhưng mất 1.2 lần chu kỳ đồng hồ để thực thi
- Vậy đồng hồ máy B phải là bao nhiêu?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

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

Số lệnh (inst. Count) và CPI

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- Số lệnh của 1 chương trình được xác định bởi: Bản thân chương trình, ISA & Biên dịch
- Số chu kỳ trung bình cho 1 lệnh:
  - Xác định bởi phần cứng CPU
  - Nếu lệnh có giá trị CPI khác nhau: CPI trung bình tổng thể

Ví dụ: Chu kỳ/lệnh (CPI)

- Máy A: T.gian/ck = 250ps, CPI = 2.0
- Máy B: T.gian/ck = 500ps, CPI = 1.2
- A & B có cùng kiến trúc tập lệnh
- Máy nào nhanh hơn, hơn bao nhiêu?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster…

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

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

…by this much

Cách tính CPI tổng quan

- Nếu các loại lệnh khác nhau thực hiện với số chu kỳ khác nhau trên mỗi lệnh

$$\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)$$

- CPI trung bình trọng số

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n} \left(\text{CPI}_i \times \frac{\text{Instruction Count}_i}{\text{Instruction Count}} \right)$$

Relative frequency

Ví dụ: CPI trung bình

- Sau khi biên dịch 1 chương trình với 3 loại lệnh A, B, C cho kết quả:

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- Kết quả biên dịch 1: IC = 5
  - Clock Cycles
    = 2×1 + 1×2 + 2×3
    = 10
  - Avg. CPI = 10/5 = 2.0
- Kết quả biên dịch 2: IC = 6
  - Clock Cycles
    = 4×1 + 1×2 + 1×3
    = 9
  - Avg. CPI = 9/6 = 1.5

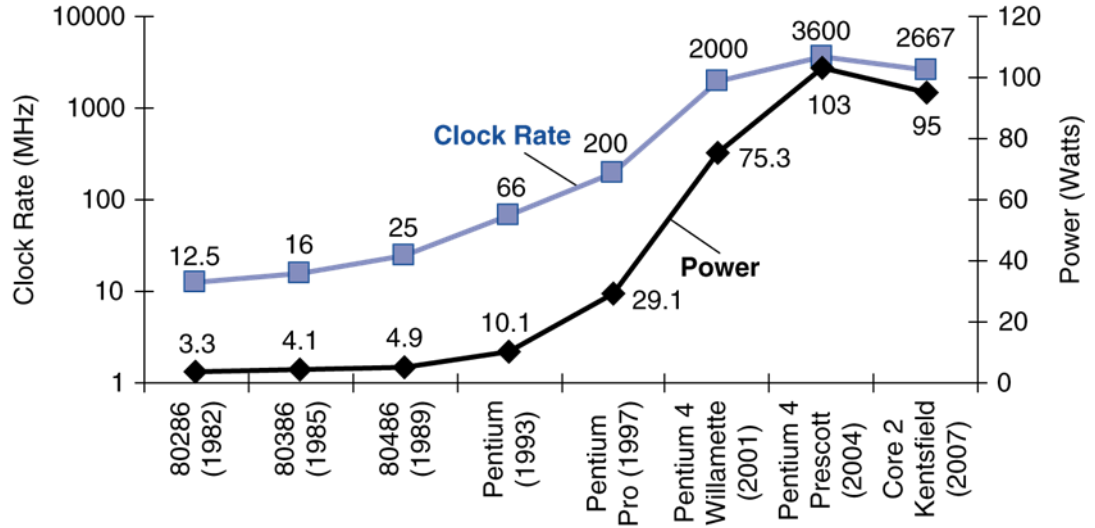
Rút ra những gì về Hiệu suất

- Công thức tổng quan

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- Phụ thuộc vào các yếu tố:
  - Giải thuật: IC, có thể cả CPI
  - Ngôn ngữ lập trình: IC, CPI
  - Biên dịch: IC, CPI
  - Kiến trúc tập lệnh: IC, CPI, T_c

Năng lượng tiêu thụ



- Trong công nghệ chế tạo CMOS IC

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

×30 | 5V → 1V | ×1000

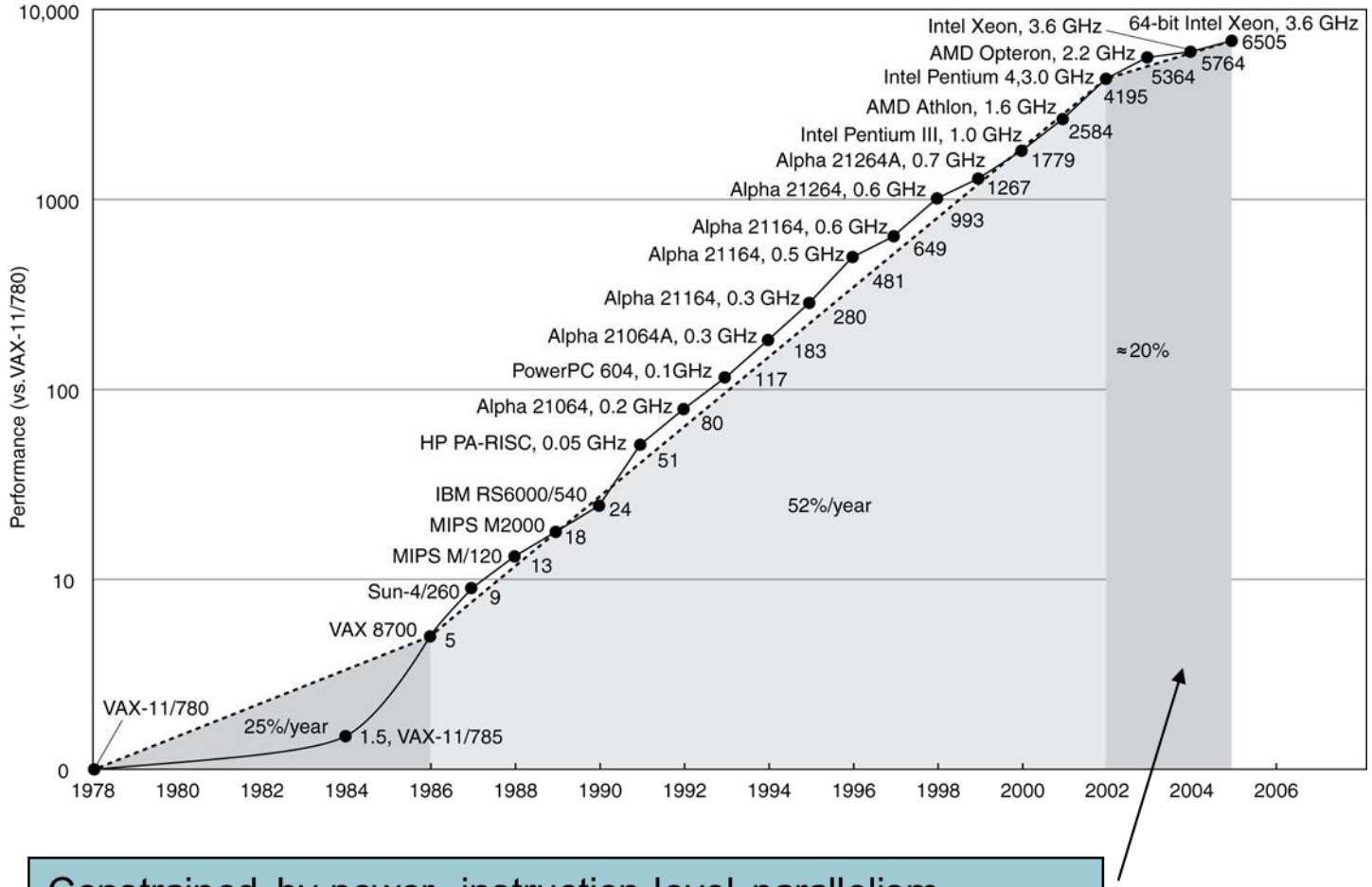
Giảm năng lượng tiêu thụ

- Giả sử 1 CPU mới so với 1 CPU cũ
  - 85% tải
  - Giảm 15% nguồn (V) và 15% tần số

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

- Ngưỡng về năng lượng tiêu thụ
  - Không thể tiếp tục giảm nguồn (v)
  - Không thể làm hạn chế nhiệt sinh ra càng tăng
- Vậy cải thiện hiệu suất bằng cách nào?

Hiệu suất đơn xử lý



Constrained by power, instruction-level parallelism, memory latency

Nhiều bộ xử lý kết hợp

- Bộ xử lý đa lõi
  - Nhiều bộ xử lý trên cùng 1 chip
- Yêu cầu lập trình song song tường minh
  - Compare with instruction level parallelism
    - Nhiều lệnh phần cứng thực hiện đồng thời
    - Hidden from the programmer
  - Khó khăn
    - Làm sao lập trình với hiệu suất cao
    - Cân bằng tải
    - Tối ưu trao đổi dữ liệu và đồng bộ

SPEC CPU Benchmark

- Tập các chương trình để đo hiệu suất
  - Có tải đặc thù sát với thực tế
- Standard Performance Evaluation Corp (SPEC)
  - Phát triển các bộ đánh giá (benchmarks) cho CPU, I/O, Web, …
- SPEC CPU2006
  - Tổng thời gian thực thi 1 nhóm chương trình được chọn ra để đánh giá
    - Không tính t.gian I/O, chỉ tập trung vào CPU
  - Normalize relative to reference machine
  - Summarize as geometric mean of performance ratios
    - CINT2006 (integer) and CFP2006 (floating-point)

$$\sqrt[n]{\prod_{i=1}^{n} \text{Execution time ratio}_i}$$

CINT2006 for Opteron X4 2356

Name	Description	IC×10^9	CPI	Tc (ns)	Exec time	Ref time	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

High cache miss rates

SPECpower_ssj2008 for X4

Target Load %	Performance (ssj_ops/sec)	Average Power (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
∑ssj_ops/ ∑power		493

MIPS đại lượng đo hiệu suất

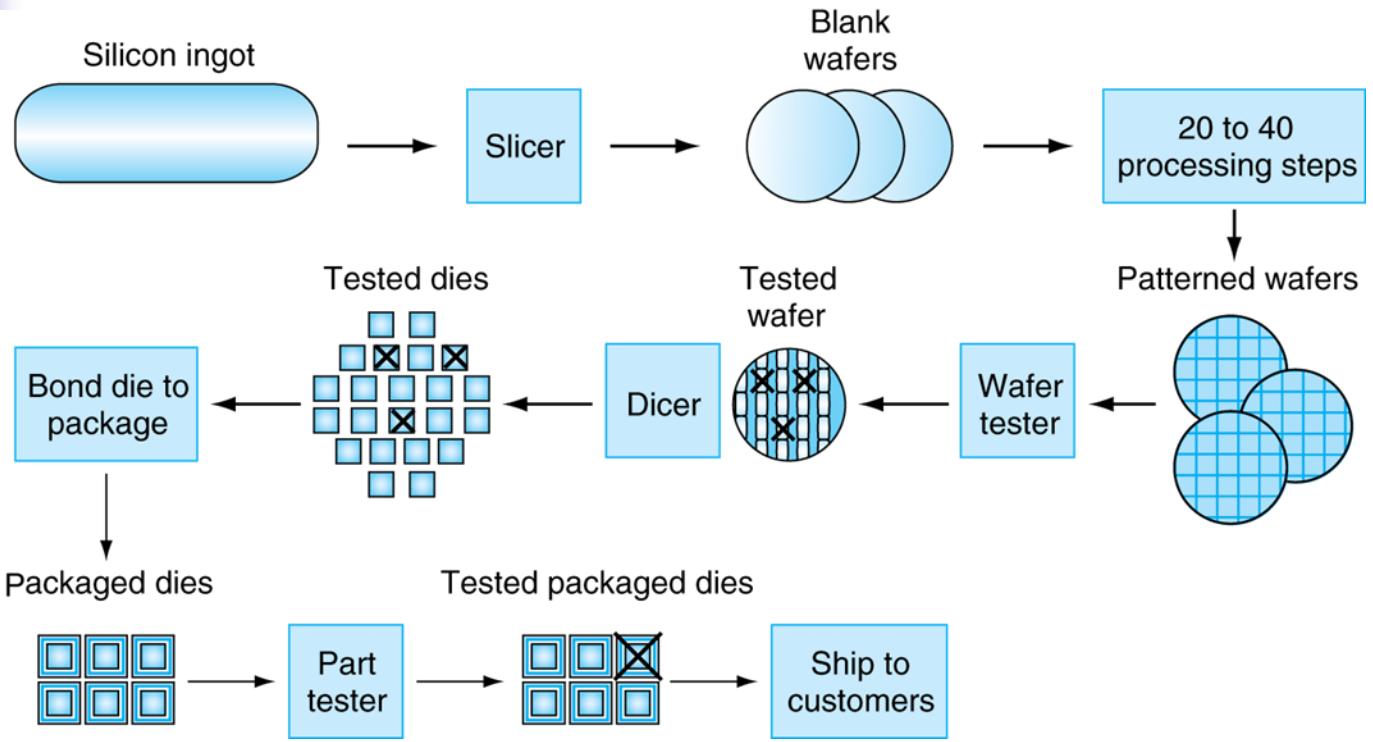
- MIPS = Millions of Instructions Per Second
  - Không dùng vào mục đích so sánh
    - Sự khác nhau về Kiến trúc tập lệnh của máy tính
    - Sự khác nhau vế độ phức tạp của lệnh

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$$= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

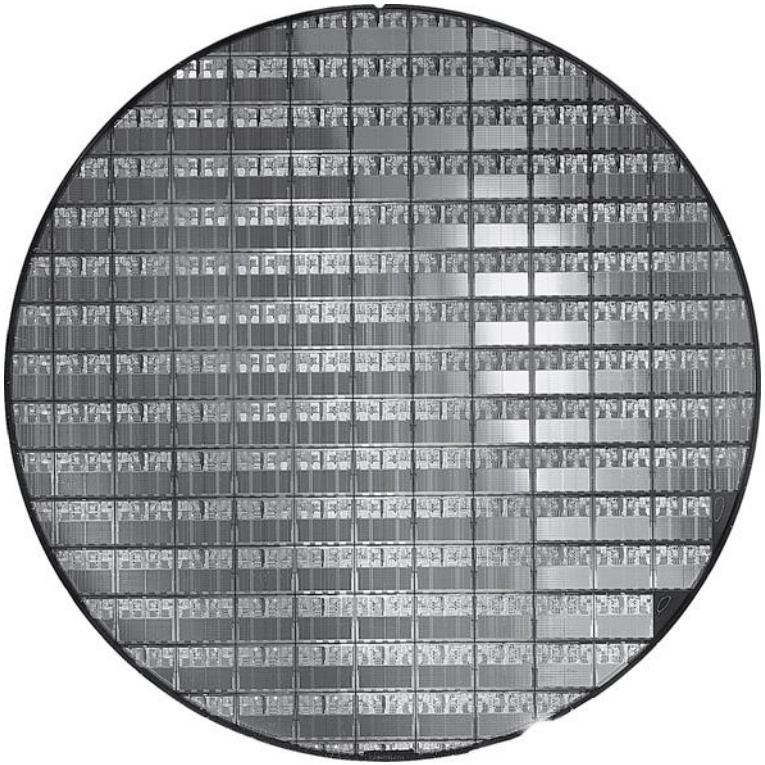
  - Các chương trình cùng thực hiện trên 1 CPU có thể có CPI khác nhau

Quy trình chế tạo mạch



- Độ lợi (Yield): số chip đạt yêu cầu/mỗi wafer

AMD Opteron X2 Wafer



- X2: 300mm wafer, 117 chips, 90nm technology
- X4: 45nm technology

Giá thành mạch tích hợp

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area} / \text{Die area}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

- Quan hệ phi tuyến với thiết diện Wafe & tỷ lệ lỗi
  - Giá thành Wafer & thiết diện cố định
  - Tỷ lệ lỗi phụ thuộc vào quy trình sản xuất
  - Thiết diện chip phụ thuộc vào kiến trúc & thiết kế mạch

Kết luận

- Giá thành/Hiệu suất ngày càng cải thiện
  - Công nghệ phát triển
- Cấu trúc tổ chức phân tầng ý niệm
  - Cả phần cứng lẫn mềm
- Kiến trúc tập lệnh
  - Phần giao Phần cứng/Mềm
- Thời gian thực thi: cách tốt nhất đo hiệu suất
- Năng lượng (Power): yếu tố cản trở nhất → Song song hóa cải thiện hiệu suất