
IT3030 - Kiến trúc máy tính

(Dựa trên bài giảng của TS. Nguyễn Kim Khánh, bộ môn KTMT)

Ngô Lam Trung
Bộ môn Kỹ thuật Máy tính
Viện Công nghệ thông tin và Truyền thông
Đại học Bách khoa Hà Nội
E-mail: trungnl@soict.hust.edu.vn

Tại sao cần học Kiến trúc máy tính?

- ❑ Kiến trúc máy tính rất quan trọng với kỹ sư CNTT
 - | Software developer: để viết chương trình tốt hơn
 - | Hardware designer: để thiết kế máy tính tốt hơn
- ❑ Sau khi học xong: kiến thức cơ sở về kiến trúc tập lệnh và tổ chức của máy tính, các vấn đề cơ bản trong thiết kế máy tính
 - | Tìm hiểu kiến trúc tập lệnh của một bộ xử lý cụ thể
 - | Lập trình hợp ngữ trên một số kiến trúc
 - | Đánh giá hiệu năng của các họ máy tính
 - | Khai thác và quản trị hiệu quả các hệ thống máy tính
 - | Phân tích và thiết kế máy tính

Thông tin học phần

- ❑ Giảng viên: Ngo Lam Trung
505 B1, SoICT, HUST
- ❑ Tài liệu: [Tham khảo] Computer Organization and Design, 4th edition revised printing, Patterson & Hennessy 2012.

[Cần có] Computer Organization and Architecture, 8th Edition, William Stalling
- ❑ Slides: pdf
- ❑ MARS (MIPS Assembler and Runtime Simulator)
<http://courses.missouristate.edu/KenVollmar/MARS/>

Nội dung

- ❑ Chương 1. Giới thiệu chung
- ❑ Chương 2. Cơ bản về logic số
- ❑ Chương 3. Hệ thống máy tính
- ❑ Chương 4. Số học máy tính
- ❑ Chương 5. Kiến trúc tập lệnh
- ❑ Chương 6. Bộ xử lý
- ❑ Chương 7. Bộ nhớ máy tính
- ❑ Chương 8. Hệ thống vào-ra
- ❑ Chương 9. Các kiến trúc song song

Cần biết trước khi học Kiến trúc máy tính

- ❑ Đã sử dụng thành thạo máy tính
 - | VD: RAM, ROM, HDD, KB là gì?
- ❑ Kiến thức về thiết kế logic
- ❑ Khả năng lập trình (C/C++, assembly)

Chương 1: Giới thiệu chung

1.1 Máy tính và phân loại

1.2. Khái niệm kiến trúc máy tính

1.3. Sơ lược lịch sử máy tính

1.4. Hiệu năng máy tính

1.1 Máy tính và phân loại

❑ Máy tính (Computer) là thiết bị điện tử thực hiện các công việc sau:

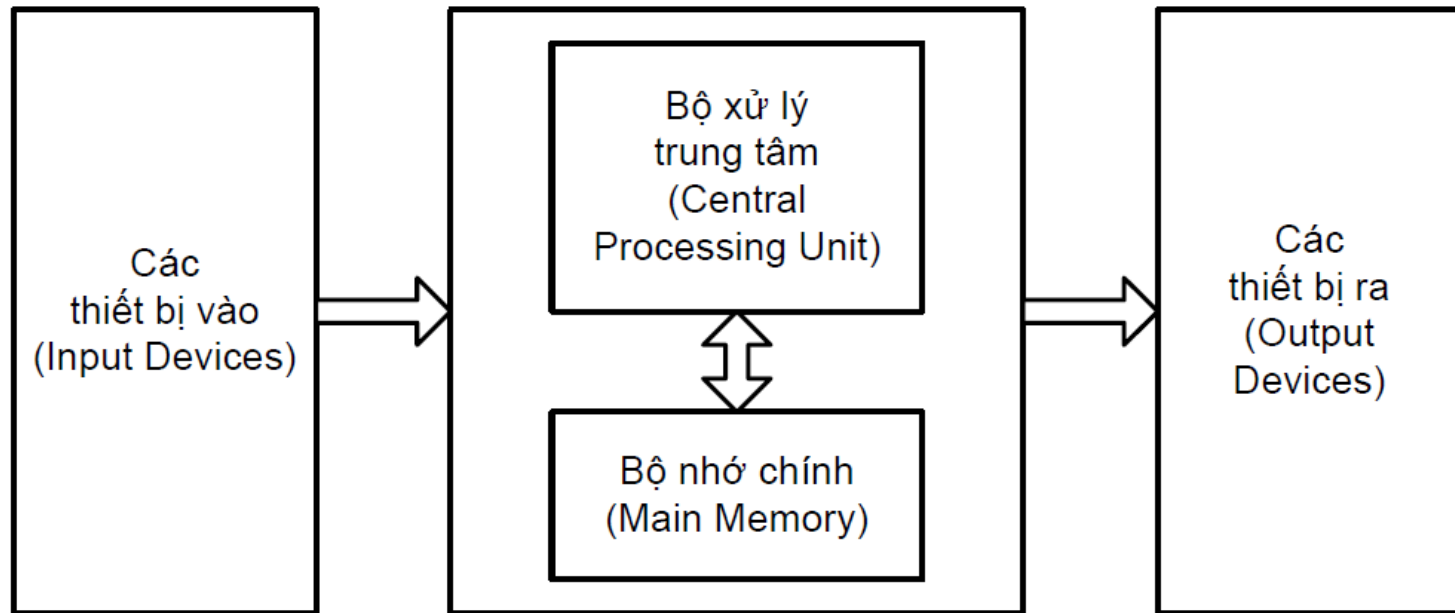
- | Nhận thông tin vào,
- | Xử lý thông tin theo dãy các lệnh được nhớ sẵn bên trong,
- | Đưa thông tin ra.

❑ Dãy các lệnh nằm trong bộ nhớ để yêu cầu máy tính thực hiện công việc cụ thể gọi là chương trình (program)

→ Máy tính hoạt động theo chương trình được lưu trữ (stored program), chương trình có thể thay đổi được.

Mô hình cơ bản của máy tính

❑ Các thành phần cơ bản



Phân loại máy tính

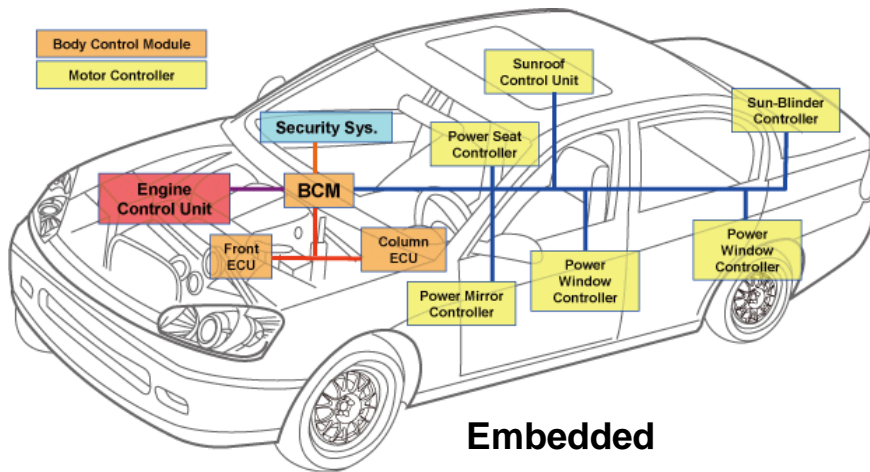
□ Phân loại truyền thống

- | Máy vi tính (Microcomputers)
- | Máy tính nhỏ (Minicomputers)
- | Máy tính lớn (Mainframe Computers)
- | Siêu máy tính (Supercomputers)

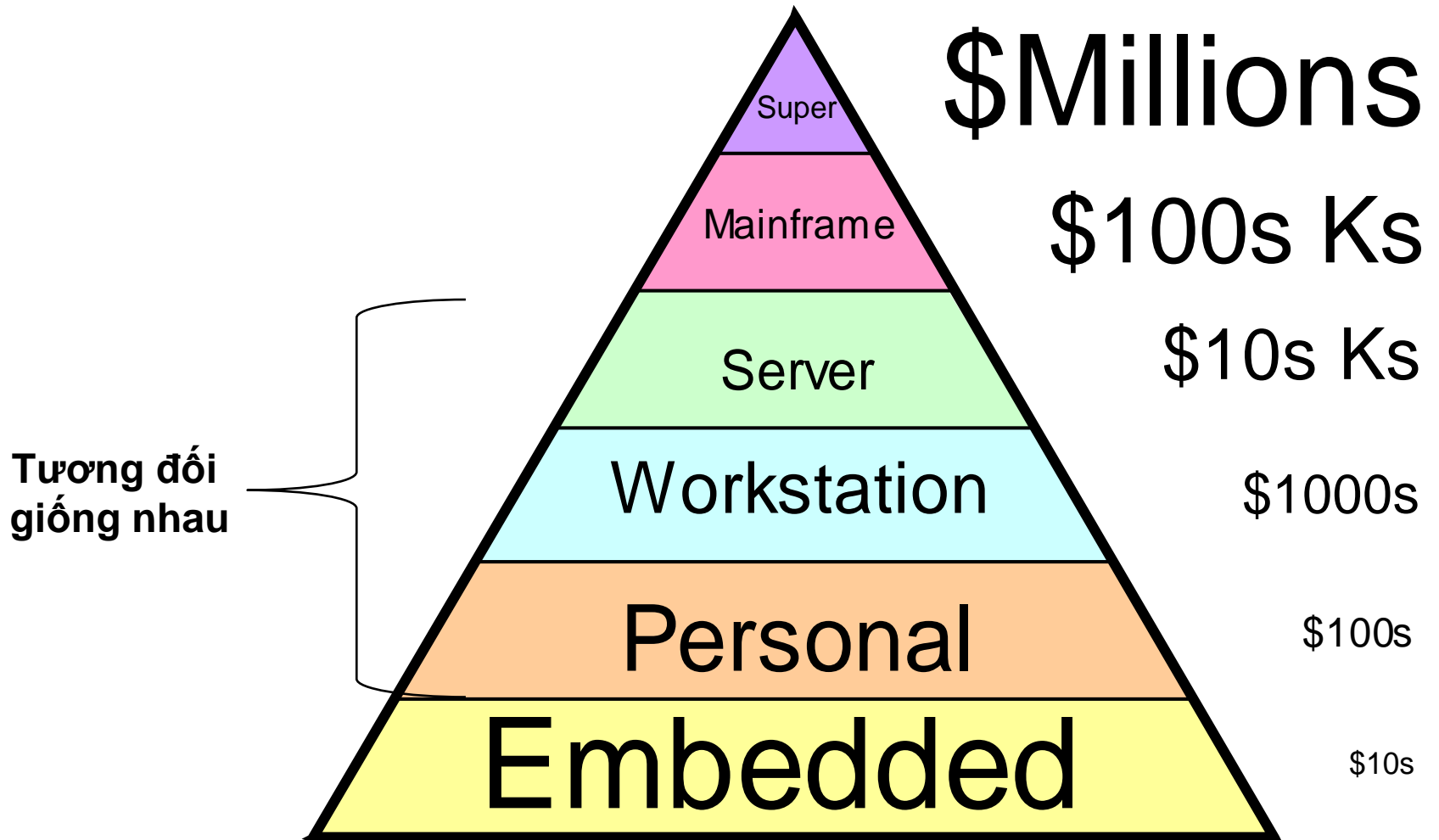
Phân loại máy tính hiện đại

- ❑ Thiết bị di động cá nhân (Personal Mobile Devices):
 - | Smartphones, Tablet
- ❑ Máy tính cá nhân, máy trạm (PC, Workstation)
 - | Desktop computers, Laptop computers
- ❑ Máy chủ (Servers)
 - | Thực chất là Máy phục vụ
 - | Dùng trong mạng theo mô hình Client/Server
- ❑ Máy tính cụm/máy tính qui mô lớn
 - | Sử dụng tại các trung tâm tính toán, trung tâm dữ liệu
- ❑ Supercomputers
- ❑ Máy tính nhúng (Embedded Computers)
 - | Đặt ẩn trong thiết bị khác
 - | Được thiết kế chuyên dụng

Look and feel



Giá thành/hiệu năng của các loại máy tính



Kỷ nguyên hậu PC (Post-PC era)

- ❑ Xu hướng công nghệ khi khối lượng tính toán không còn chỉ được thực hiện trên PC truyền thống
- ❑ Được đưa ra bởi Bill Gates (PC Plus, 1999) và Steve Jobs (post-PC device, 2007).
- ❑ Thiết bị di động: tablet, mobile phones, handheld, smart watch, wearable devices...
- ❑ Cloud Services: Azure, EC2...



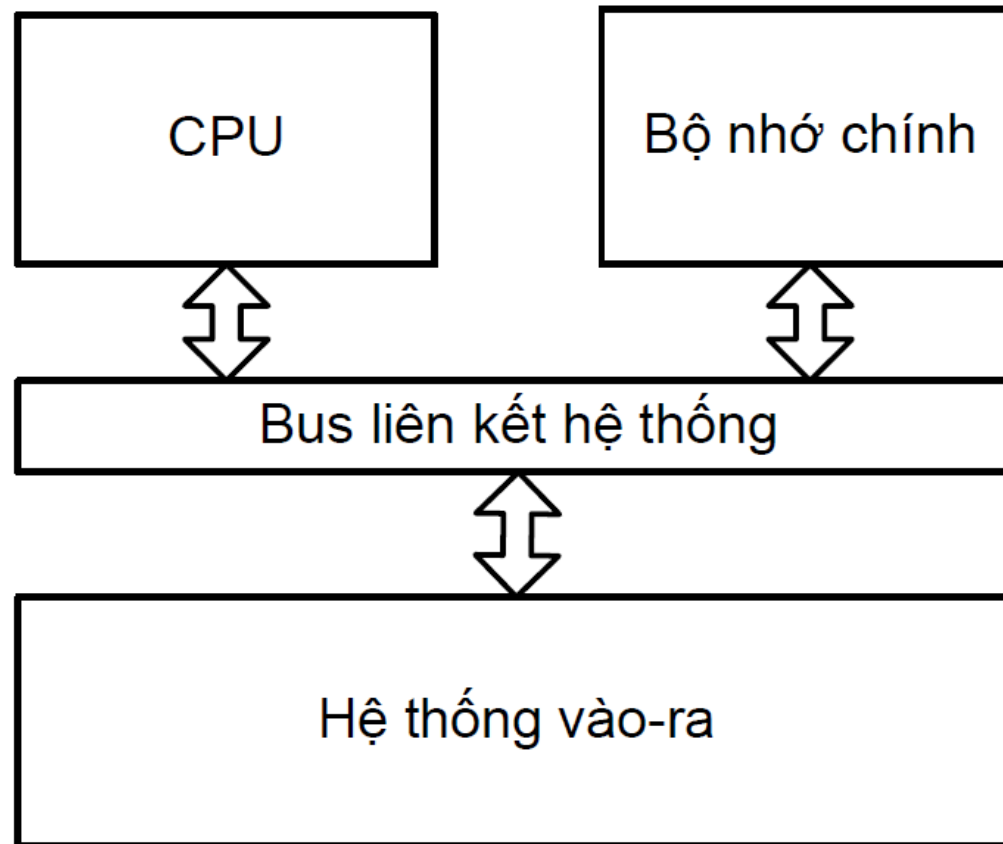
1.2 Khái niệm kiến trúc máy tính

- ❑ Kiến trúc máy tính bao gồm:
 - | **Kiến trúc tập lệnh (Instruction Set Architecture):** nghiên cứu máy tính theo cách nhìn của người lập trình (hardware/software interface).
 - | **Tổ chức máy tính (Computer Organization) hay Vi kiến trúc (Microarchitecture):** nghiên cứu thiết kế máy tính ở mức cao, chẳng hạn như hệ thống nhớ, cấu trúc bus, thiết kế bên trong CPU.
 - | **Phần cứng (Hardware):** nghiên cứu thiết kế logic chi tiết và công nghệ đóng gói của máy tính.
- ❑ Nhận xét: cùng một kiến trúc tập lệnh có thể có nhiều sản phẩm (tương ứng tổ chức, phần cứng) khác nhau.

Kiến trúc tập lệnh

- ❑ Kiến trúc tập lệnh của máy tính bao gồm:
 - | Tập lệnh: tập hợp các chuỗi số nhị phân mã hoá cho các thao tác mà máy tính có thể thực hiện
 - | Các kiểu dữ liệu: các kiểu dữ liệu mà máy tính có thể xử lý

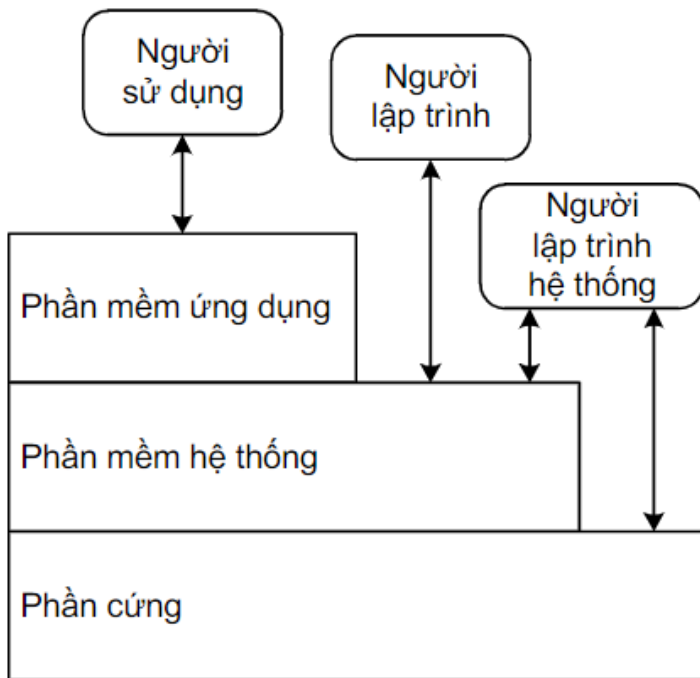
Cấu trúc cơ bản của máy tính



Các thành phần cơ bản

- ❑ Bộ xử lý trung tâm (Central Processing Unit):
 - | Điều khiển hoạt động của máy tính và xử lý dữ liệu.
- ❑ Bộ nhớ chính (Main Memory):
 - | Chứa các chương trình và dữ liệu đang được sử dụng.
- ❑ Hệ thống vào-ra (Input/Output System):
 - | Trao đổi thông tin giữa máy tính với bên ngoài.
- ❑ Bus liên kết hệ thống (System Interconnection Bus):
 - | Kết nối và vận chuyển thông tin giữa các thành phần với nhau.

Mô hình phân lớp của máy tính

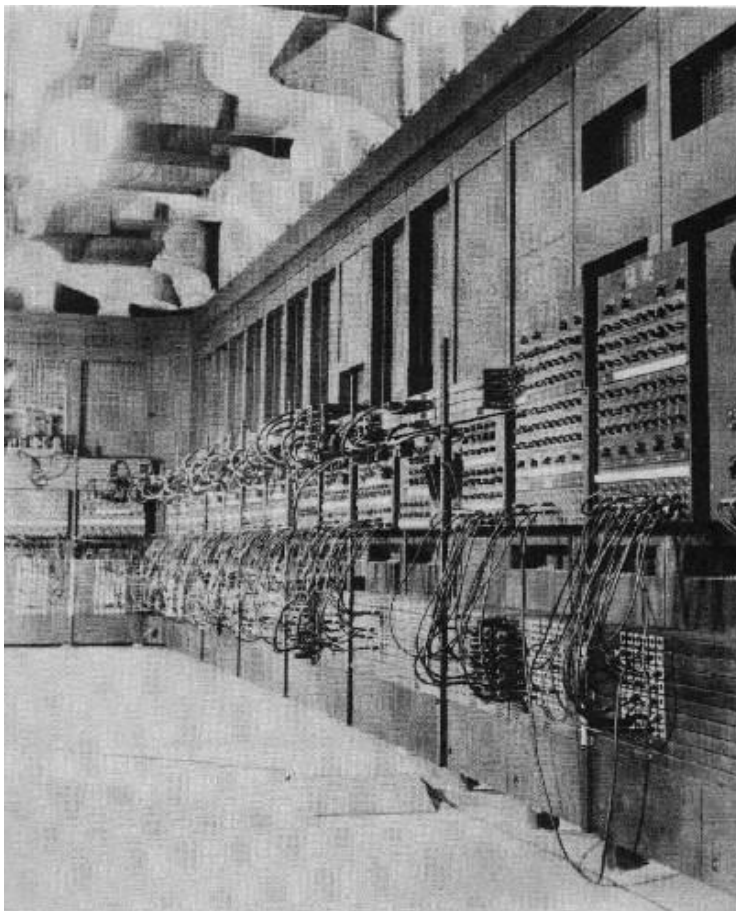


- Phần mềm ứng dụng
 - Được viết theo ngôn ngữ bậc cao
- Phần mềm hệ thống
 - Chương trình dịch (Compiler): dịch mã ngôn ngữ bậc cao thành ngôn ngữ máy
 - Hệ điều hành (Operating System)
 - Lập lịch cho các nhiệm vụ và chia sẻ tài nguyên
 - Quản lý bộ nhớ và lưu trữ
 - Điều khiển vào-ra
- Phần cứng
 - Bộ xử lý, bộ nhớ, mô-đun vào-ra

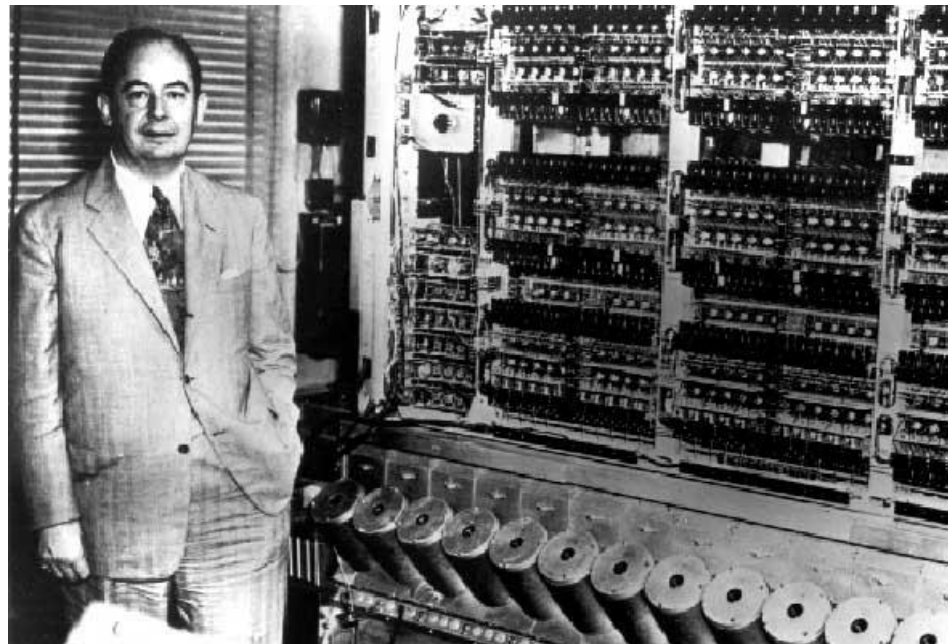
1.3 Sự tiến hóa của máy tính

Thế hệ	Công nghệ chế tạo	Bộ nhớ	Thiết bị vào ra	look & feel
0 (1600s)	(Điện-)Cơ khí	Wheel, card	Cần gạt, quay số	Factory equipment
1 (1950s)	Đèn điện tử chân không	Magnetic drum	Băng từ	Hall-size cabinet
2 (1960s)	Transistor	Magnetic core	Trống từ, máy in, đầu cuối	Room-size mainframe
3 (1970s)	Vi mạch SSI/MSI	RAM/ROM chip	Đĩa từ, màn hình, bàn phím	Desk-size mini
4 (1980s)	Vi mạch LSI/VLSI	SRAM/DRAM	Nối mạng, âm thanh, chuột, CD, đồ họa	Desktop/laptop micro
5 (1990s)	ULSI/GSI/WSI, SOC	SDRAM, flash	Cảm biến, giao diện thông minh	Invisible, embedded

Các máy tính đầu tiên



ENIAC, 1947

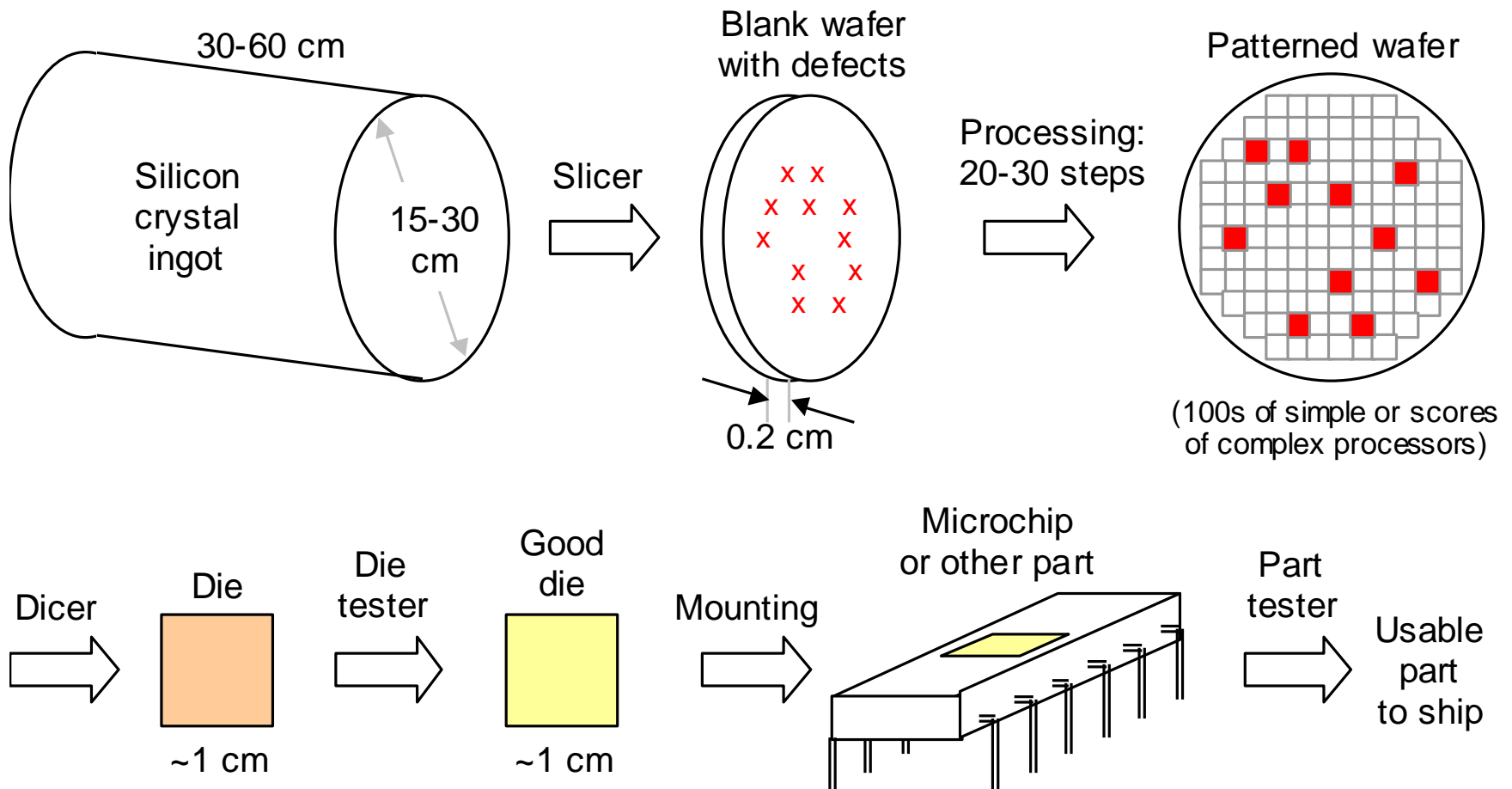


**IAS, Von Neumann, 1952
(stored program concept)
→ Mô hình cơ bản của máy
tính hiện đại**

Vi mạch (IC)

- ❑ Vi mạch hay là mạch tích hợp (Integrated Circuit - IC): mạch điện tử gồm nhiều transistors và các linh kiện khác được tích hợp trên một chip bán dẫn.
- ❑ Phân loại vi mạch theo qui mô tích hợp:
 - | SSI - Small Scale Integration
 - | MSI - Medium Scale Integration
 - | LSI - Large Scale Integration
 - | VLSI - Very Large Scale Integration
 - | ULSI - Ultra Large Scale Integration

Quy trình chế tạo IC



The manufacturing process for an IC part

Video: How an IC is made

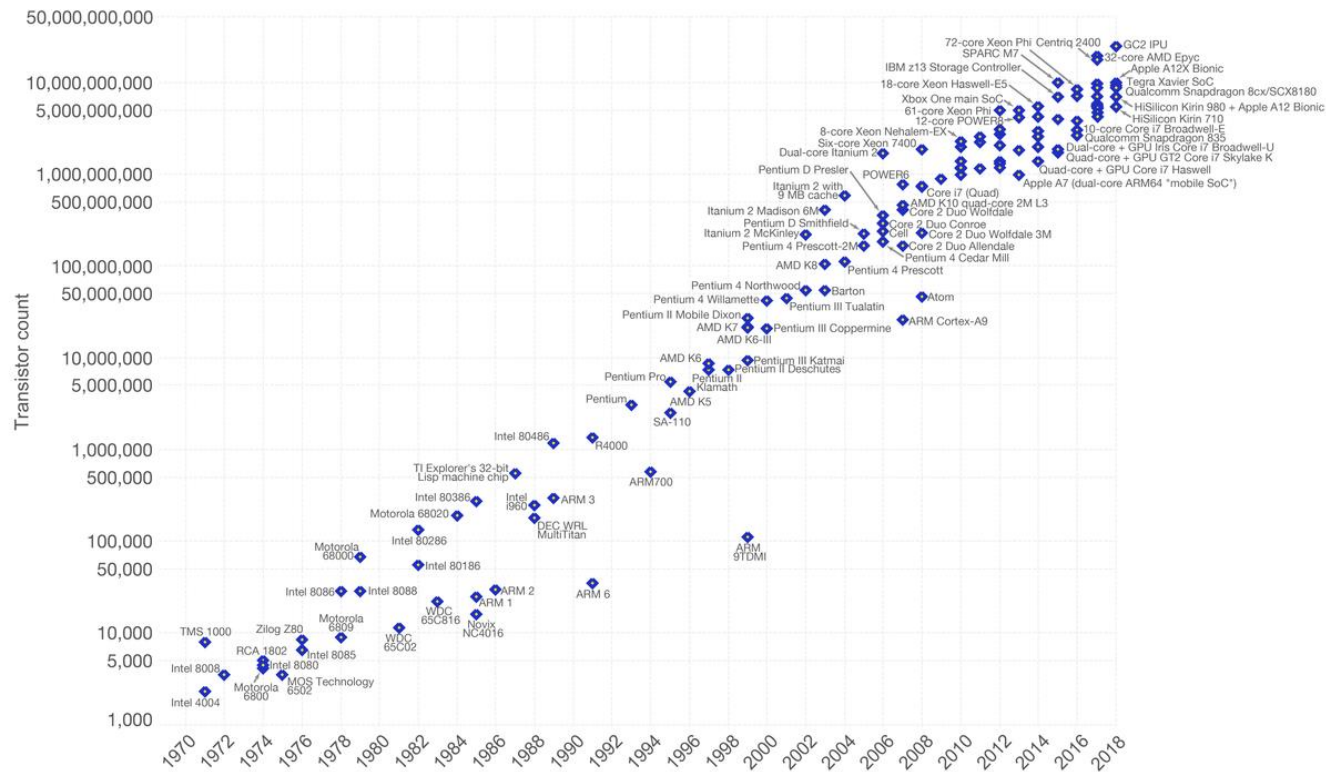
Luật Moore

❑ Số transistor trên một chip tăng gấp đôi sau 18 tháng

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.

OurWorld
in Data



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

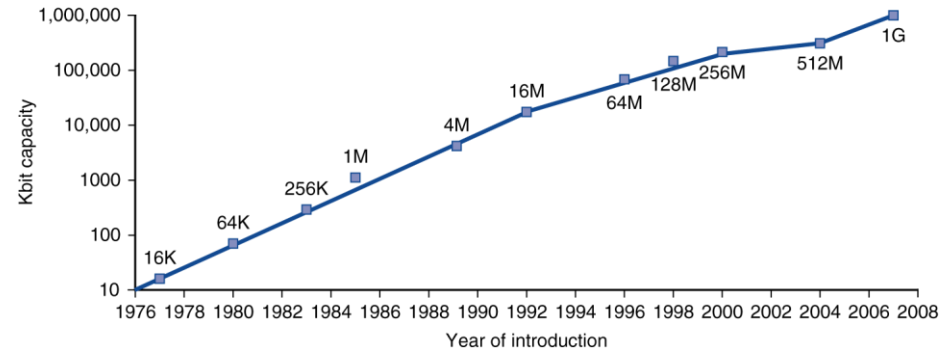
The data visualization is available at OurWorldInData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

Xu hướng công nghệ

❑ Công nghệ chế tạo vi mạch phát triển từng giờ

- | Tăng hiệu năng
- | Giảm giá thành



Dung lượng DRAM

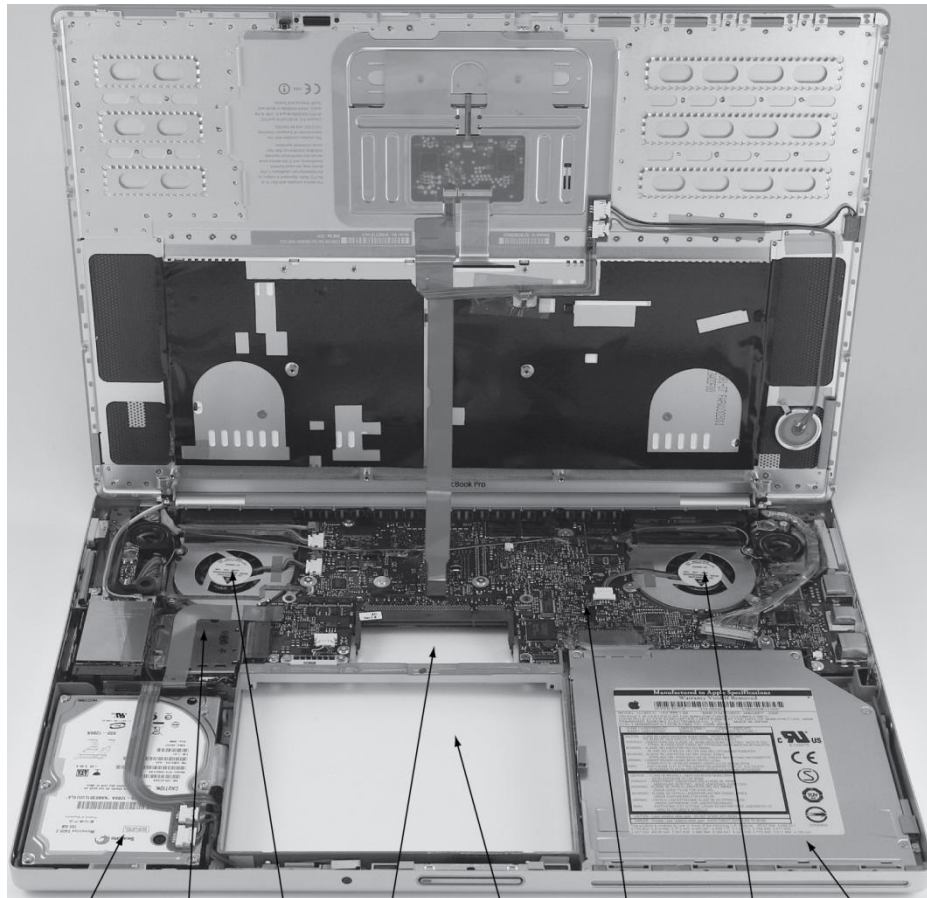
Thời gian	Công nghệ	Hiệu năng/giá thành (tương đối)
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

[Textbook]

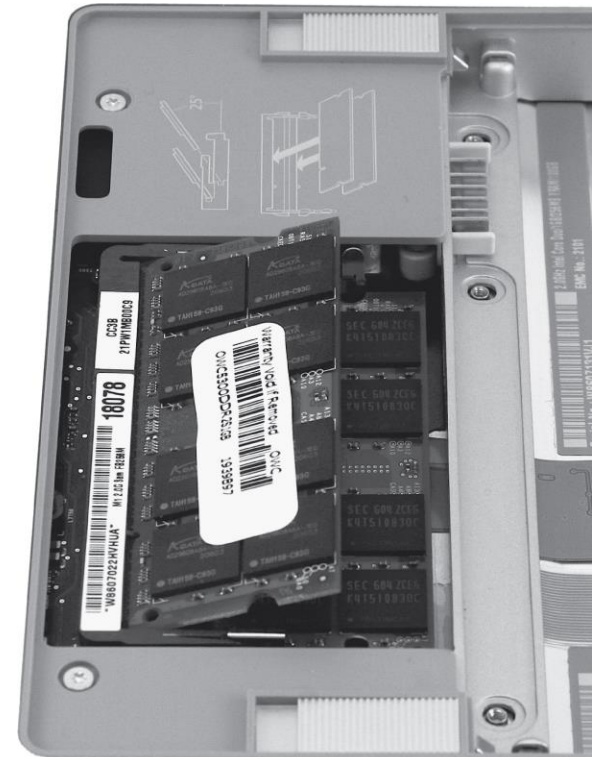
Sự phát triển của bộ xử lý

- ❑ 1971: bộ vi xử lý 4-bit Intel 4004
- ❑ 1972: các bộ xử lý 8-bit
- ❑ 1978: các bộ xử lý 16-bit
 - | IBM PC: 1981
- ❑ 1985: các bộ xử lý 32-bit
- ❑ 2001: các bộ xử lý 64-bit
- ❑ 2006: các bộ xử lý đa lõi (multicores)

Bên trong máy tính cá nhân



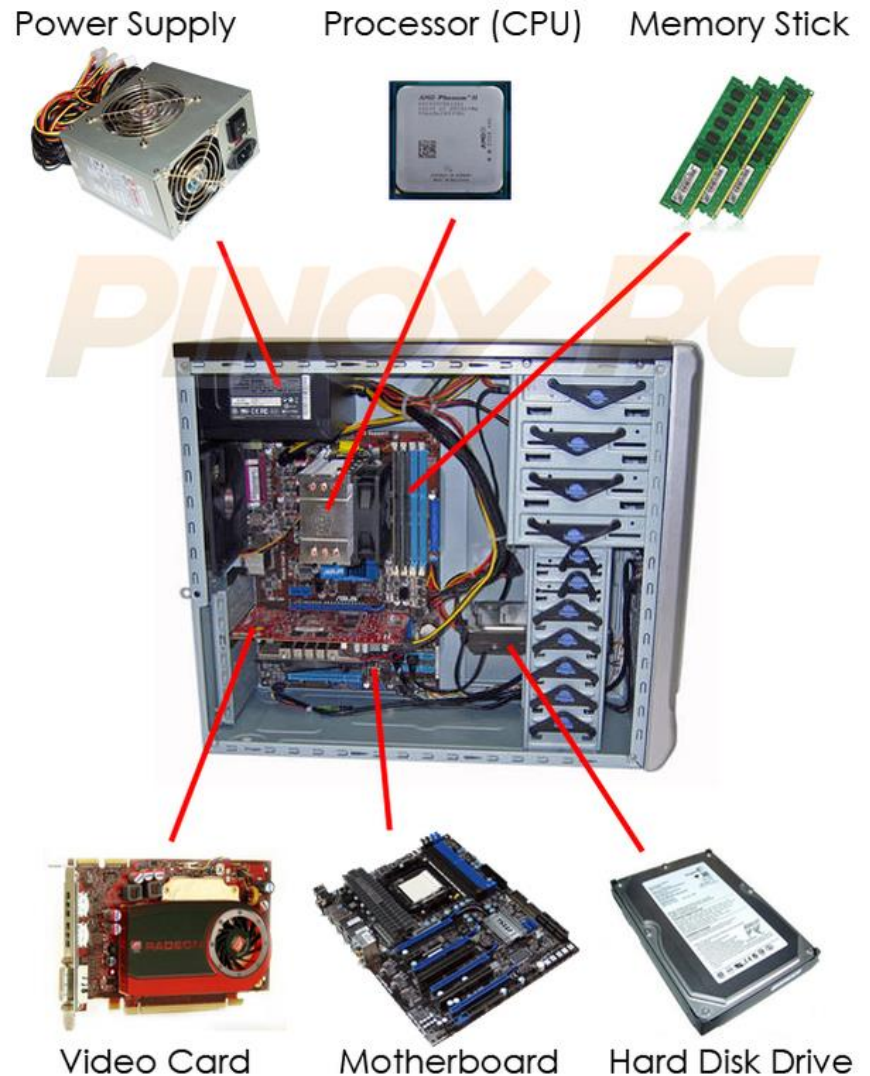
Hard drive Processor Fan with cover Spot for memory DIMMs Spot for battery Motherboard Fan with cover DVD drive cover



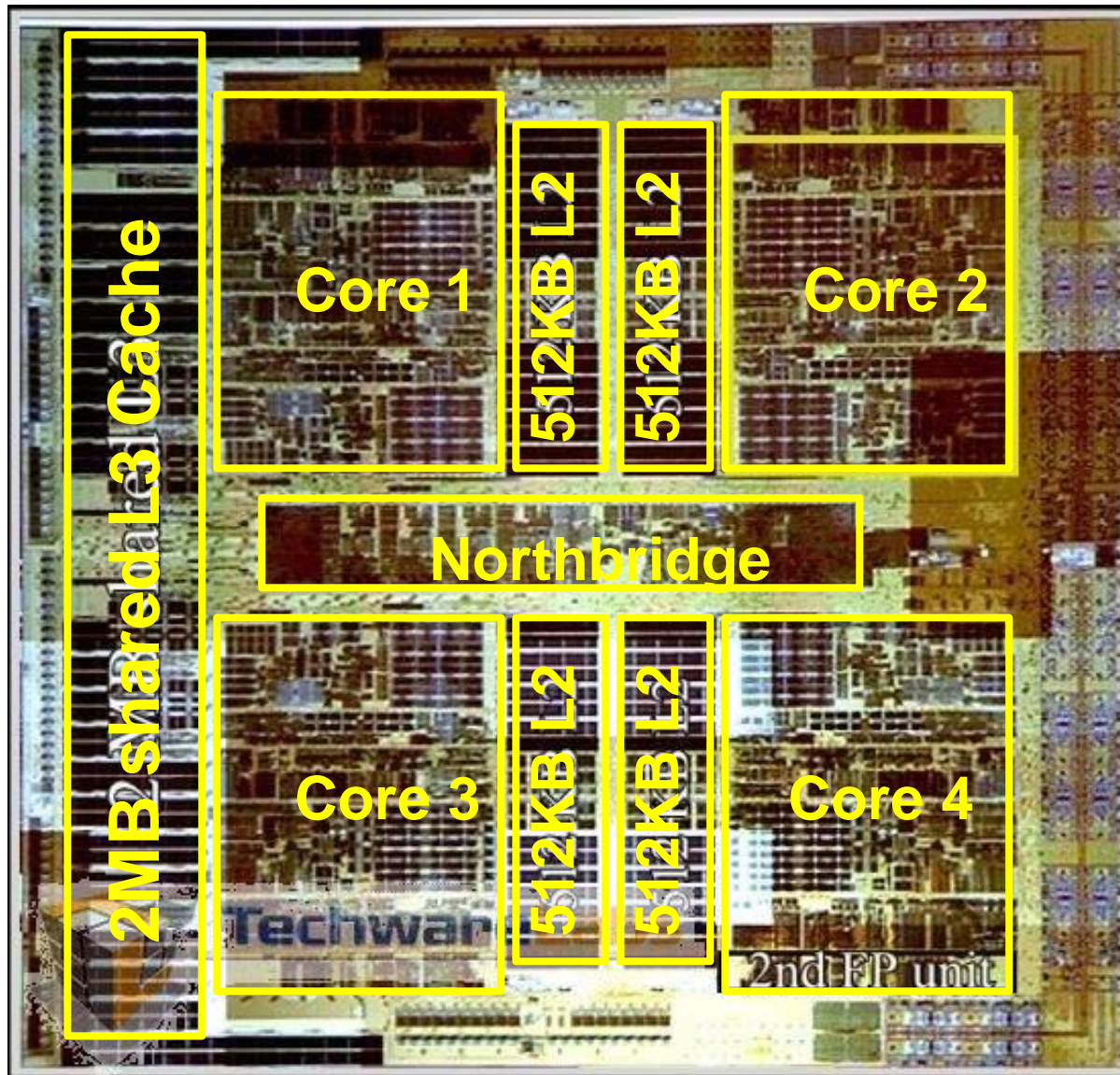
Bên trong máy tính cá nhân



**The story of each component
worth a separate course!**



Bên trong CPU (AMD Barcelona)



- ❑ Bốn lõi
- ❑ 1.9 GHz clock rate
- ❑ 65nm technology
- ❑ Ba mức caches (L1, L2, L3) on chip
- ❑ Tích hợp Northbridge

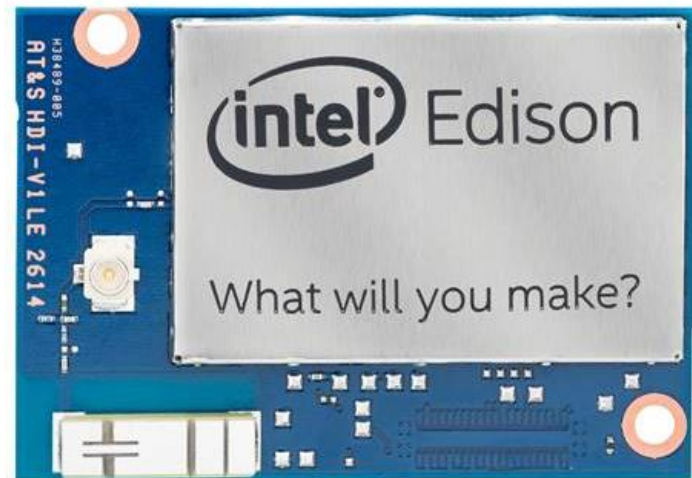
Case study

- ❑ Máy tính «nhỏ nhất» thế giới: Raspberry Pi



Case study

- ❑ Module tính toán nhỏ nhất: Intel Edison
- ❑ Kích thước 25mm x 35.5mm (thẻ SD)
- ❑ Dual core Atom 500 Mhz
- ❑ Single core Intel Quark to run RTOS
- ❑ 4GB Flash ROM
- ❑ 1GB DDR RAM



1.4 Hiệu năng của máy tính

- ❑ Thước đo hiệu năng quan trọng nhất là **thời gian thực thi**

$$\text{performance}_x = 1 / \text{execution_time}_x$$

execution_time_x : thời gian thực thi chương trình trên máy tính X

Nếu máy tính X chạy nhanh hơn máy tính Y n lần

$$\frac{\text{performance}_x}{\text{performance}_y} = \frac{\text{execution_time}_y}{\text{execution_time}_x} = n$$

Ví dụ

- ❑ Nếu máy tính A chạy một chương trình hết 10 giây, máy B chạy cùng chương trình đó hết 15 giây. So sánh hiệu năng của A và B

Ta có

$$\frac{\text{performance}_A}{\text{performance}_B} = \frac{\text{execution_time}_B}{\text{execution_time}_A} = n$$

Tỷ số hiệu năng

$$\frac{15}{10} = 1.5$$

Vậy A nhanh hơn B 1.5 lần

Ước lượng hiệu năng

- ❑ Thời gian CPU là thời gian mà CPU thực sự thực thi chương trình
 - ❑ Không kể thời gian CPU chờ hệ thống vào ra, hoặc chờ các chương trình khác

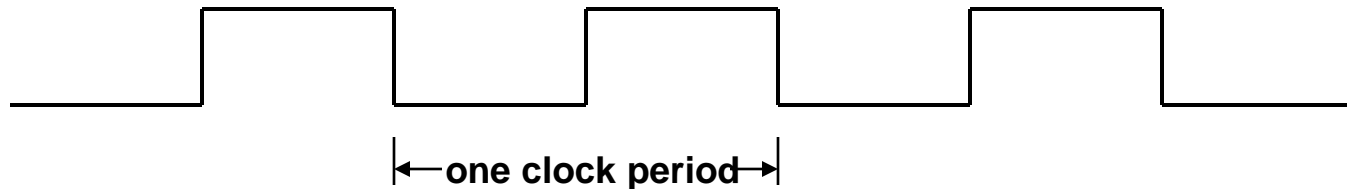
$$\begin{aligned}\text{Thời gian CPU} &= \text{số chu kỳ clock của chương trình} \times \text{thời gian của một xung clock} \\ &= \frac{\text{số chu kỳ clock của chương trình}}{\text{tần số clock}}\end{aligned}$$

- ❑ Để tăng hiệu năng:
 - ❑ Giảm số chu kỳ
 - ❑ Tăng tần số clock

Xung nhịp của CPU

- ❑ Hoạt động của CPU được điều khiển bởi xung nhịp, gọi là xung clock

$$CC = 1 / CR$$



CC: Clock Cycle,
chu kỳ xung clock

CR: Clock Rate,
tần số xung clock

10 nsec clock cycle => 100 MHz clock rate

5 nsec clock cycle => 200 MHz clock rate

2 nsec clock cycle => 500 MHz clock rate

1 nsec (10^{-9}) clock cycle => 1 GHz (10^9) clock rate

500 psec clock cycle => 2 GHz clock rate

250 psec clock cycle => 4 GHz clock rate

200 psec clock cycle => 5 GHz clock rate

Ví dụ

- ❑ Một chương trình chạy trên máy tính A có tần số xung 2 GHz mất 10 giây. Hỏi tần số xung của máy tính B cần là bao nhiêu để chạy chương trình đó trong 6 giây? Biết rằng để chạy chương trình đó, máy tính B sẽ cần số chu kỳ xung nhiều hơn máy A 1.2 lần.

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{clock rate}_A}$$

$$\begin{aligned}\text{CPU clock cycles}_A &= 10 \text{ sec} \times 2 \times 10^9 \text{ cycles/sec} \\ &= 20 \times 10^9 \text{ cycles}\end{aligned}$$

$$\text{CPU time}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{clock rate}_B}$$

$$\text{clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = 4 \text{ GHz}$$

Lệnh và chu kỳ clock

- ❑ Mỗi lệnh được thực thi trong một vài chu kỳ clock
- ❑ Các loại lệnh khác nhau cần số chu kỳ clock khác nhau
- ❑ Nếu biết số lệnh mỗi loại và số chu kỳ clock tương ứng
→ tổng số lệnh chu kỳ cần thiết

$$n = \sum CPI_i * IC_i$$

IC_i : số lệnh loại i

CPI_i : số chu kỳ cho mỗi lệnh loại i

- ❑ số chu kỳ trung bình cho mỗi lệnh

$$CPI = \frac{n}{\sum IC_i} = \sum CPI_i * \frac{IC_i}{\sum IC_i}$$

Ví dụ

- ❑ Một chương trình chứa 3 loại lệnh
 - | Loại lệnh A, tần số xuất hiện 30%, CPI = 5
 - | Loại lệnh B, tần số xuất hiện 20%, CPI = 6
 - | Loại lệnh C, tần số xuất hiện 50%, CPI = 2
- ❑ Tính CPI trung bình?

Ước lượng số chu kỳ clock

- ❑ Số chu kỳ clock = số lệnh x số chu kỳ trên một lệnh

$$CC = IC * CPI$$

- ❑ IC: instruction count, số lệnh trong chương trình
- ❑ CPI: số chu kỳ trung bình cần để thực thi mỗi lệnh

→ Thời gian thực thi của chương trình

$$\text{CPU time} = IC \times CPI \times \text{chu kỳ clock}$$

$$= \frac{IC \times CPI}{\text{tần số clock}}$$

Ví dụ

- ❑ Máy tính A: $T_A = 250\text{ps}$, $\text{CPI}_A = 2.0$
- ❑ Máy tính B: $T_B = 500\text{ps}$, $\text{CPI}_B = 1.2$
- ❑ Cùng kiến trúc tập lệnh (ISA)
- ❑ Máy nào nhanh hơn và nhanh hơn bao nhiêu ?

Cùng ISA nên số lệnh cần thực thi n là như nhau

$$\text{CPU time}_A = n \times 2.0 \times 250 \text{ ps} = 500 \times n \text{ ps}$$

$$\text{CPU time}_B = n \times 1.2 \times 500 \text{ ps} = 600 \times n \text{ ps}$$

Rõ ràng A nhanh hơn B

$$\frac{\text{performance}_A}{\text{performance}_B} = \frac{\text{execution_time}_B}{\text{execution_time}_A} = \frac{600 \times n / \text{ps}}{500 \times n / \text{ps}} = 1.2$$

Số lệnh tĩnh và động

Có bao nhiêu lệnh được thực thi trong chương trình này?

Coi rằng mỗi lệnh for ứng với 2 lệnh máy (tăng biến đếm và kiểm tra điều kiện lặp)

250 instructions

for i = 1, 100 do

20 instructions

for j = 1, 100 do

40 instructions

for k = 1, 100 do

10 instructions

endfor

endfor

endfor

Static count = 326

12,422,450 Instructions

2 + 20 + 124,200 instructions

100 iterations

12,422,200 instructions in all

2 + 40 + 1200 instructions

100 iterations

124,200 instructions in all

2 + 10 instructions

100 iterations

1200 instructions in all

Số lệnh tĩnh và động

- ❑ Ý nghĩa của số lệnh tĩnh: kích thước chương trình
- ❑ Ý nghĩa của số lệnh động: thời gian thực thi

```
1: for (i=0; i<1000000;i++)  
2:     A[i]++;
```

```
1: A[0]++;  
2: A[1]++;  
3: .....  
4: A[100]++;
```

Ước lượng hiệu năng

- ❑ Thời gian thực thi

$$\text{CPU Time} = \frac{\text{Số lệnh}}{\text{Ch.trình}} \times \frac{\text{Số chu kỳ}}{\text{Lệnh}} \times \frac{\text{Số giây}}{\text{Chu kỳ}}$$

- ❑ Rút ngắn chu kỳ clock → tăng tần số clock

→ phụ thuộc công nghệ chế tạo CPU

- ❑ Giảm CPI

→ tối ưu hóa kiến trúc tập lệnh

- ❑ Smaller instruction count

→ tối ưu hóa thuật toán và chương trình dịch

Ví dụ: tăng hiệu năng máy tính

Op	Freq	CPI _i	Freq x CPI _i			
ALU	50%	1	.5	.5	.5	.25
Load	20%	5	1.0	.4	1.0	1.0
Store	10%	3	.3	.3	.3	.3
Branch	20%	2	.4	.4	.2	.4
$Avg\ CPI = \sum freq_i * CPI_i$			= 2.2	1.6	2.0	1.95

- ❑ Hiệu năng của máy tăng bao nhiêu nếu máy được tăng cường bộ nhớ đệm giúp giảm CPI của lệnh load còn 2 chu kỳ?

CPU time mới = 1.6 x IC x CC → 2.2/1.6 ~ tăng 37.5% hiệu năng

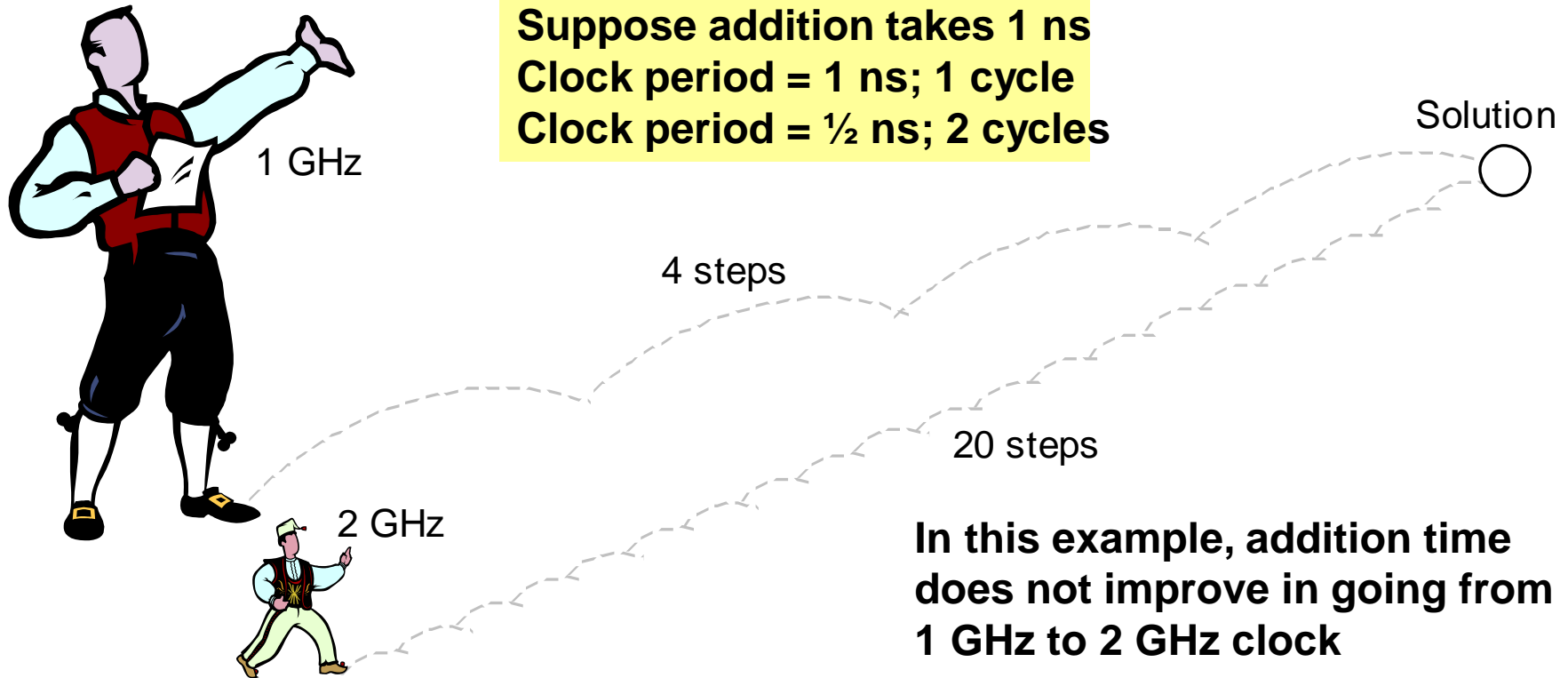
- ❑ Nếu lệnh rẽ nhánh chỉ còn một chu kỳ?

CPU time mới = 2.0 x IC x CC → 2.2/2.0 ~ tăng 10% hiệu năng

- ❑ Nếu 2 lệnh ALU có thể cùng thực thi?

CPU time mới = 1.95 x IC x CC → 2.2/1.95 ~ tăng 12.8% hiệu năng

Faster Clock \neq Shorter Running Time



Faster steps do not necessarily mean shorter travel time.