

KIẾN TRÚC MÁY TÍNH



Ths Đinh Xuân Trường
Khoa Công nghệ thông tin
Khoa học máy tính

Email: truongdx@ptit.edu.vn

Kiến trúc



Kiến trúc ...

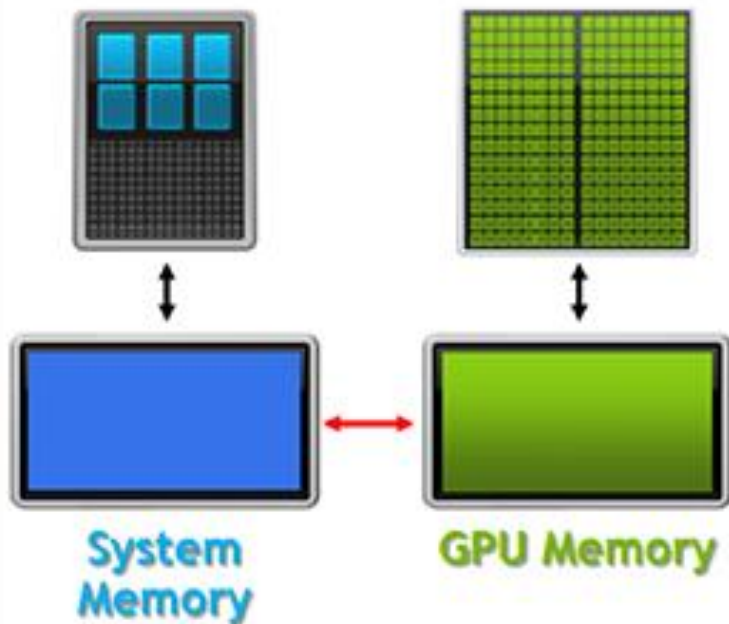


Bài học

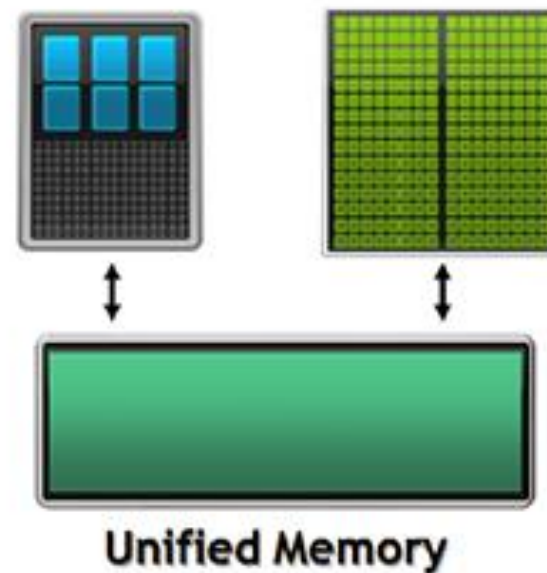
- So sánh 2 kiến trúc trên
- Giống nhau
 - Cùng chức năng một ngôi nhà (Ăn, ngủ, sinh hoạt)
- Khác nhau
 - Vật liệu khác nhau
 - Cách bố trí
 - Giá thành
- Bài học
 - Sức sáng tạo
 - Bắt đầu từ một nhu cầu đơn giản -> cải tiến và hoàn thiện

Kiến trúc

Traditional Developer View



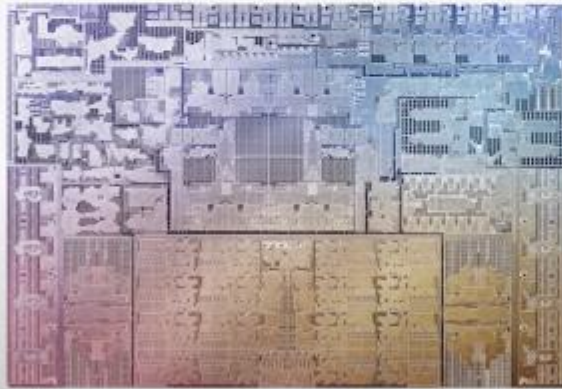
Developer View With Unified Memory



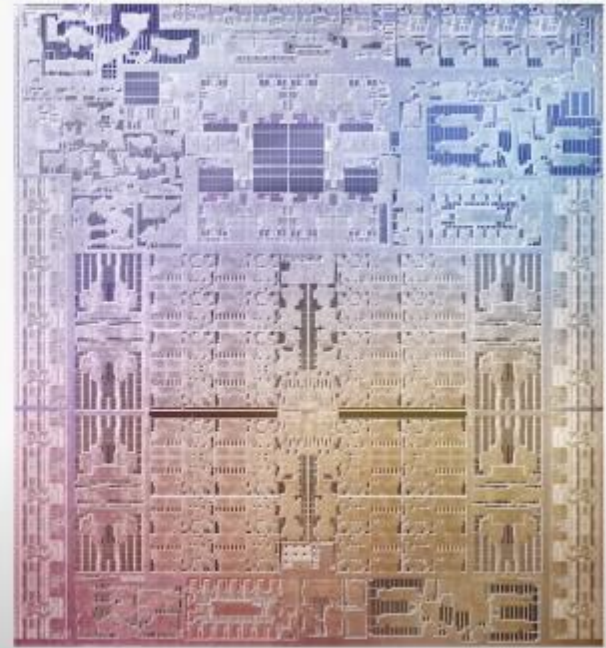
Apple Silicon



🍏 M1

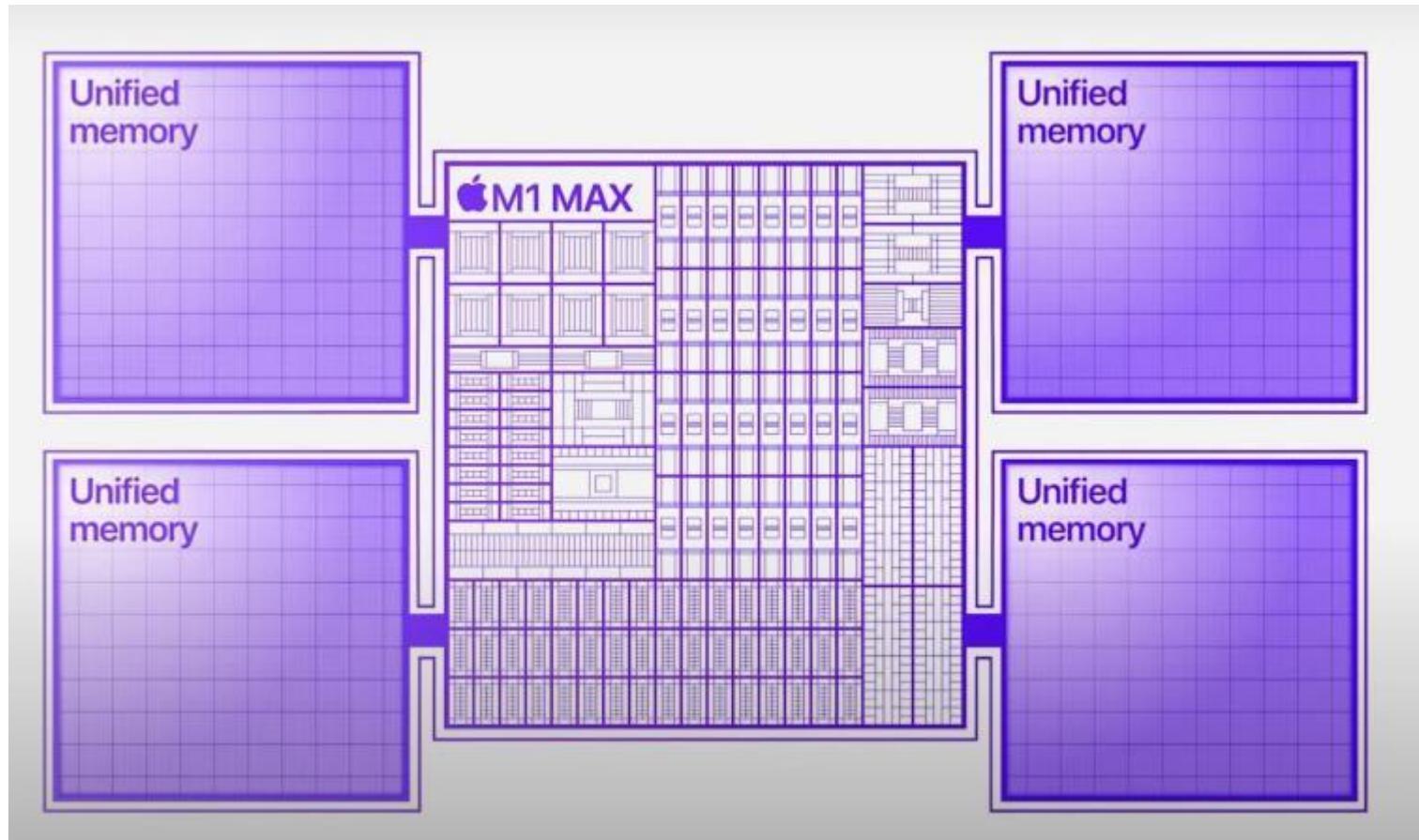


🍏 M1 Pro



🍏 M1 Max

Apple Silicon



Giới thiệu về kiến trúc máy tính

- ❑ Là khoa học cơ sở của ngành CNTT
- ❑ Khoa học về chọn lựa các thành phần phần cứng → hiệu năng cao, giá thành rẻ → chính là cơ sở của các công ty công nghệ mới (Apple, Intel, IoT Misfit, ...)
- ❑ Rất nhiều sản phẩm từ IoT (giá vài đô), đến các hệ thống lớn (nhiều triệu đô) -> Đều gọi là máy tính
- ❑ Học về các khối chức năng cơ bản cho một hệ thống máy tính hoàn chỉnh.

Tài liệu tham khảo

- Bài giảng “Kiến trúc máy tính” – Hoàng Xuân Dậu
- Stallings W., *Computer Organization and Architecture*, Prentice – Hall 2013.
- Hennesy J.L. and Patterson D.A., *Computer Architecture. A Quantitative Approach*, Morgan Kaufmann, 2003.
- Trần Quang Vinh, *Cấu trúc máy vi tính*, Nhà xuất bản Giáo dục, 1999.
- Bài giảng Computer Architecture online (youTube, ...)

Tài liệu tham khảo



[1] William Stallings - *Computer Organization and Architecture – Designing for Performance – 2009 (8th edition)*



[2] David A. Patterson & John L. Hennessy - *Computer Organization and Design: The Hardware/Software Interface – 2009 (4th edition)*



[3] David Money Harris and Sarah L. Harris, *Digital Design and Computer Architecture*, 2007



[4] Behrooz Parhami - *Computer Architecture: From Microprocessors to Supercomputers - 2005*

Điểm thành phần

- Điểm chuyên cần: 10%
- Bài tập theo từng chương : 20%
- Kiểm tra trắc nghiệm : 10%
- Thi cuối kỳ : 60%



Các nội dung chính

- Chương 1: Giới thiệu chung
- Chương 2: Khối xử lý trung tâm – CPU
- Chương 3: Tập lệnh máy tính
- Chương 4: Bộ nhớ trong
- Chương 5: Bộ nhớ ngoài
- Chương 6: Hệ thống bus và thiết bị ngoại vi



Chương 1: Giới thiệu chung

Chương 1: Nội dung chính

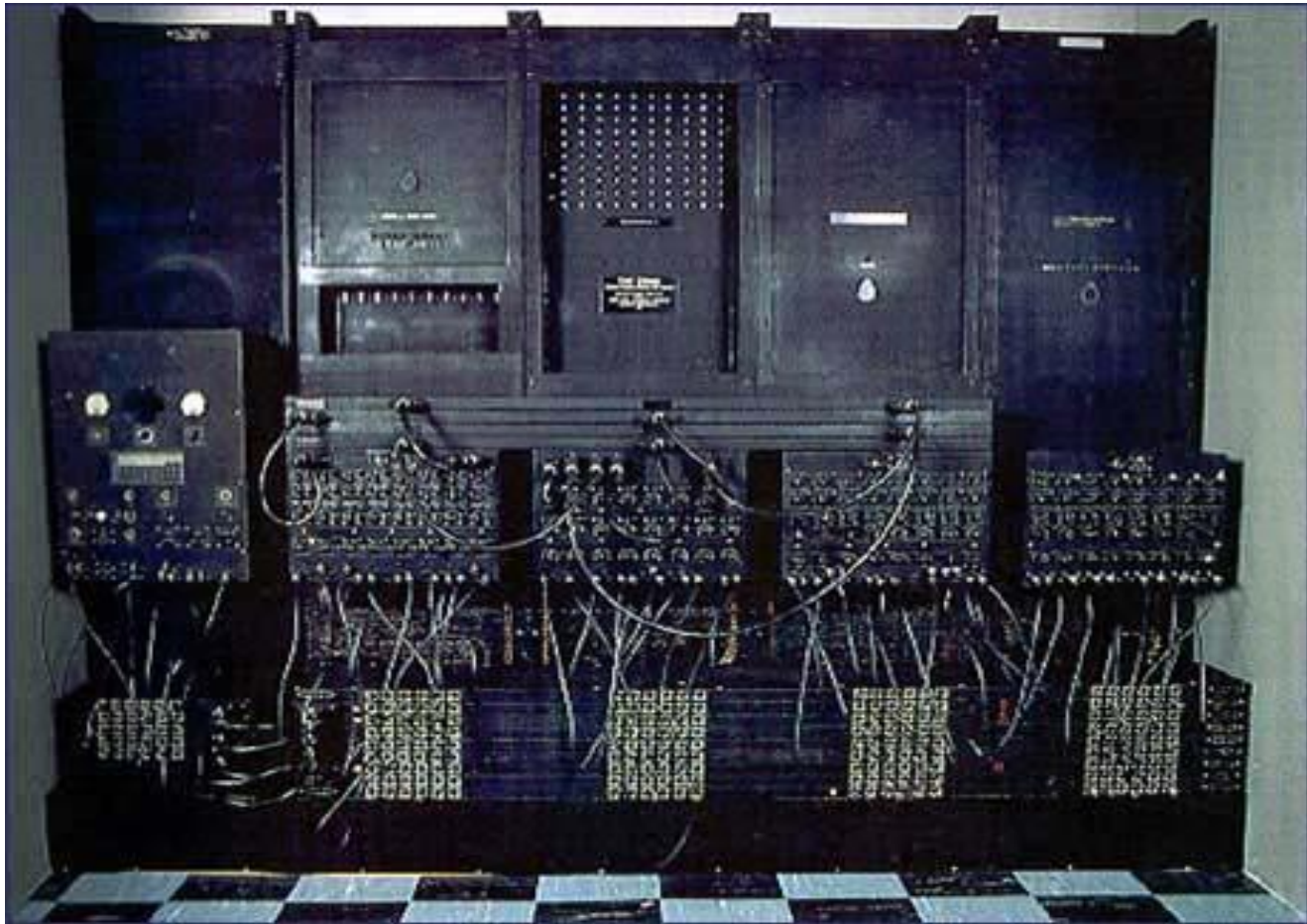
1. Giới thiệu kiến trúc máy tính
2. Lịch sử phát triển máy tính
3. Khái niệm kiến trúc và tổ chức máy tính
4. Cấu trúc và chức năng của máy tính
5. Kiến trúc Von Neumann
6. Kiến trúc Harvard
7. Biểu diễn dữ liệu trong máy tính

Lịch sử phát triển máy tính

Lịch sử phát triển máy tính

- Chia thành 5 thế hệ dựa trên sự phát triển mạch điện tử
- Thế hệ 1 (1944-1959):
 - Sử dụng bóng đèn điện tử
 - Dùng băng từ làm các thiết bị đầu vào/ ra
 - Mật độ tích hợp linh kiện: 1000 linh kiện/ foot³ (1 foot= 30.48 cm)
 - Ví dụ: ENIAC - Electronic Numerical Integrator and Computer, 1946, giá 500,000 USD.

Lịch sử phát triển máy tính - ENIAC



Lịch sử phát triển máy tính

- Thế hệ thứ 2(1960-1964):
 - Sử dụng transistors
 - $\sim 100,000$ linh kiện/ foot³
 - Ví dụ: UNIVAC 1107, UNIVAC III, IBM 7070, 7080, 7090, 1400 series, 1600 series. (1951, đầu tiên giá \$159K, sau đó UNIVAC 1 giá hơn 1 triệu \$)

Lịch sử phát triển máy tính– UNIVAC



Lịch sử phát triển máy tính

- Thế hệ thứ 3 (1964-1975):
 - Sử dụng mạch tích hợp (IC)
 - ~ 10 triệu linh kiện/ foot³
 - Ví dụ: UNIVAC 9000 series, IBM System/360, System 3, System 7

Lịch sử phát triển máy tính– UNIVAC 9400



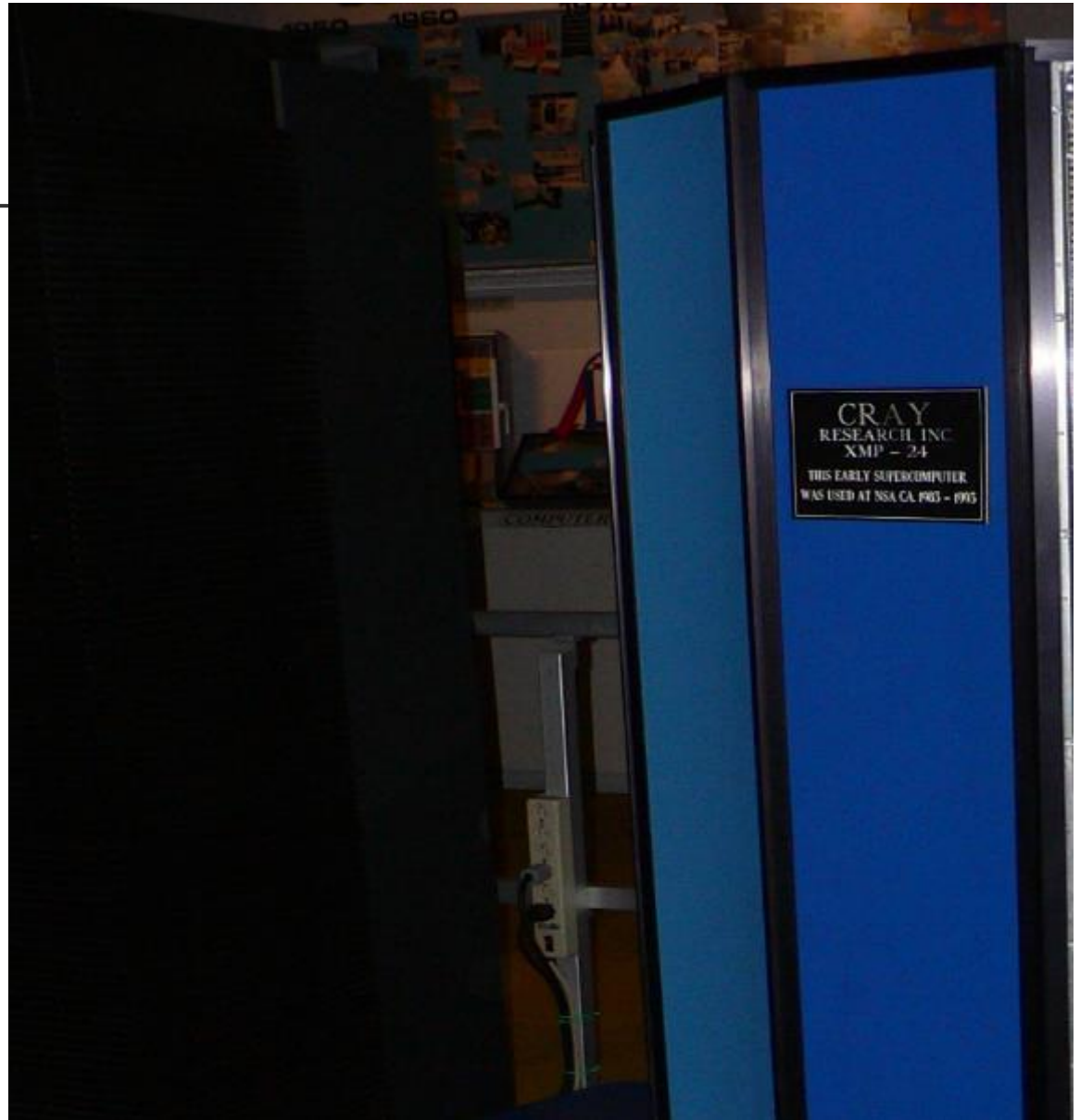
Lịch sử phát triển máy tính

□ Thế hệ 4(1975-1989):

- Sử dụng LSI – Large Scale Integrated Circuit
- ~ 1 tỷ linh kiện / foot³
- Ví dụ: IBM System 3090, IBM RISC 6000, IBM RT, Cray 2 XMP

History of computers

Cray 2 XMP



Lịch sử phát triển máy tính

- Thế hệ thứ 5 (1990- nay):
 - Sử dụng VLSI – Very Large Scale Integrated Circuit
 - $<10\text{nm} - 0.045\mu\text{m}$
 - Ví dụ: Pentium II, III, IV, M, D, Core Duo, Core 2 Duo, Core Quad,...
 - Hỗ trợ xử lý song song
 - Hiệu năng rất cao
 - Kết hợp xử lý giọng nói và hình ảnh

1. Kiến trúc và tổ chức máy tính

- ❑ Kiến trúc máy tính (computer architecture): là khoa học về lựa chọn và kết nối các thành phần phần cứng của máy tính nhằm đạt yêu cầu:
 - Hiệu năng: càng nhanh càng tốt
 - Chức năng: nhiều chức năng
 - Giá thành: càng rẻ càng tốt
- ❑ Tổ chức máy tính (computer organization): là khoa học nghiên cứu các thành phần của máy tính và phương thức làm việc của chúng dựa trên kiến trúc cho trước

Ví dụ

- ❑ Computer Architecture: IBM Thinkpad, Iphone, Android Phones → Thiết kế hệ thống dòng sản phẩm
- ❑ Computer Organization → Các thế hệ máy khác nhau với cùng kiến trúc phần cứng
 - Các đời Thinkpad khác nhau: T40, T50, ..
 - Iphone 4, 4s, 5, 5s, 6, 6s, 7, 7s, 8, X, 11, 12, 13
 - Samsung S2, S3, S4, S5, S6, S7, S8, S10, Z...

1. Kiến trúc và tổ chức máy tính

□ 3 thành phần cơ bản của kiến trúc máy tính

1. Kiến trúc tập lệnh (ISA): là hình ảnh trừu tượng của máy tính ở mức ngôn ngữ máy (hoặc hợp ngữ), bao gồm:
 - Tập lệnh
 - Các chế độ địa chỉ bộ nhớ
 - Các thanh ghi
 - Khuôn dạng địa chỉ và dữ liệu

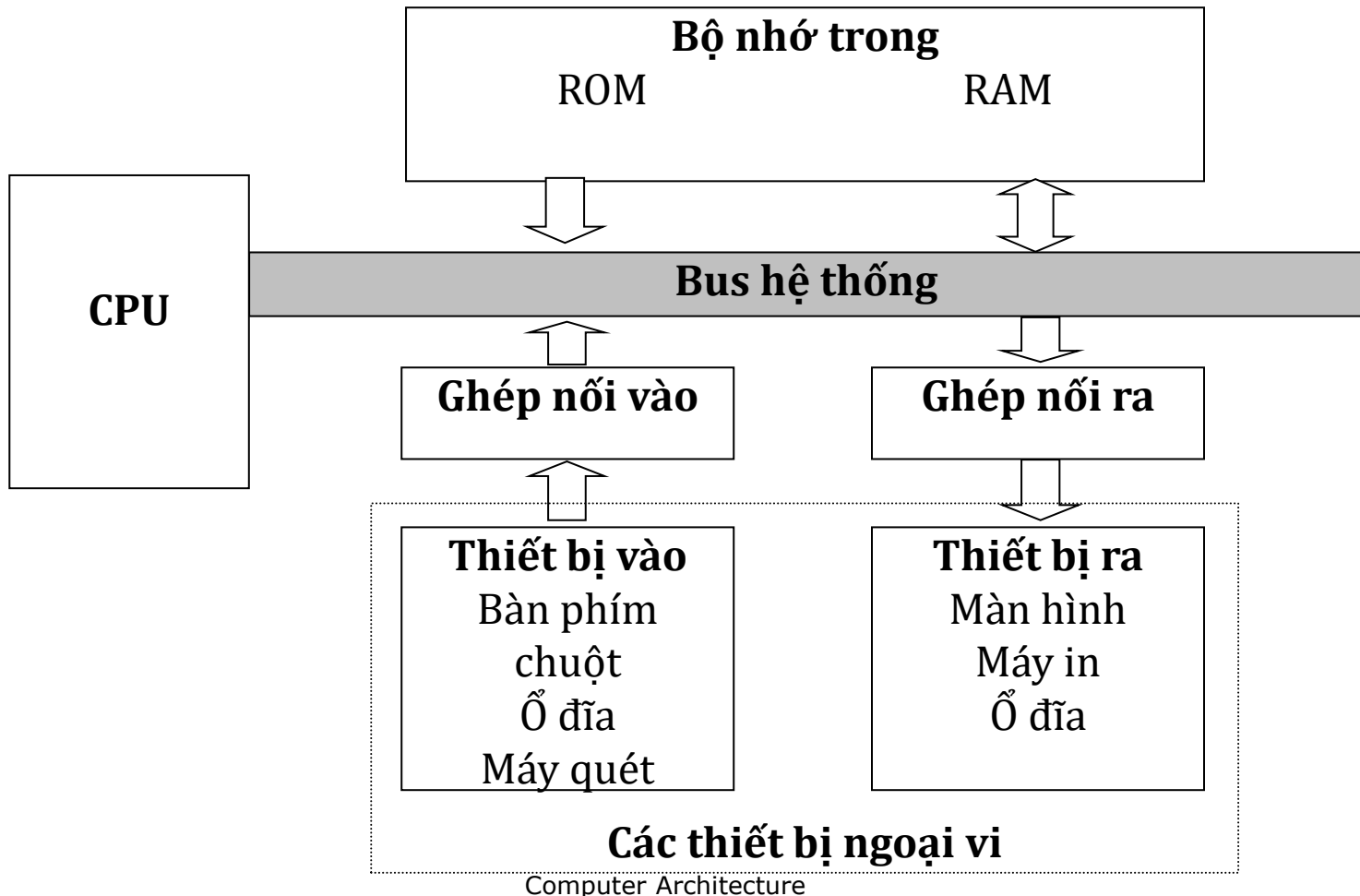
1. Kiến trúc và tổ chức máy tính

2. Vi kiến trúc (microarchitecture): còn được gọi là tổ chức máy tính, mô tả về hệ thống ở mức thấp, liên quan tới:
- ▣ Các thành phần phần cứng kết nối với nhau như thế nào
 - ▣ Các thành phần phần cứng phối hợp, tương tác với nhau như thế nào để thực hiện tập lệnh

1. Kiến trúc và tổ chức máy tính

3. Thiết kế hệ thống, bao gồm tất cả các thành phần phần cứng khác trong hệ thống máy tính, ví dụ:
- ❑ Các hệ thống kết nối như bus và chuyển mạch
 - ❑ Mạch điều khiển bộ nhớ, cấu trúc phân cấp bộ nhớ
 - ❑ Các kỹ thuật giảm tải cho CPU như truy cập trực tiếp bộ nhớ
 - ❑ Các vấn đề như đa xử lý

2. Cấu trúc và các thành phần chức năng



2. Cấu trúc và các thành phần chức năng

□ Bộ xử lý trung tâm (CPU):

■ Chức năng:

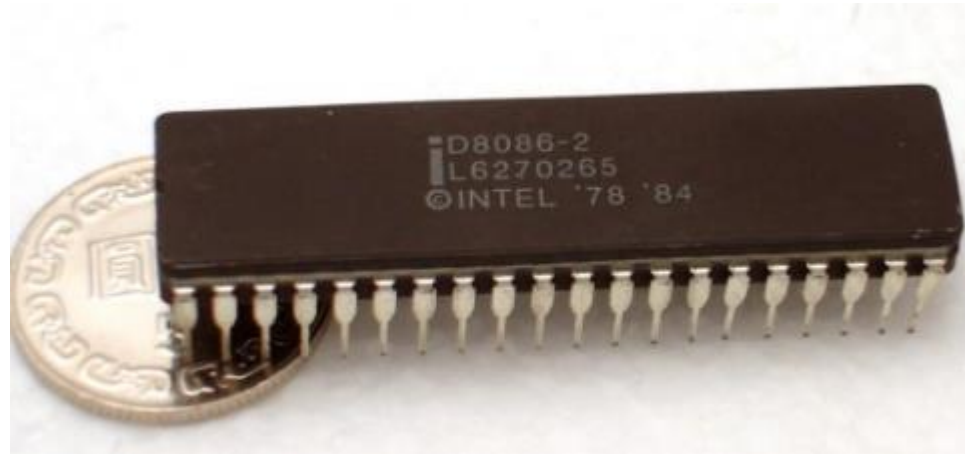
- Đọc lệnh từ bộ nhớ
- Giải mã và thực hiện lệnh

■ Bao gồm:

- Khối điều khiển (CU: Control Unit)
- Khối tính toán số học và logic (ALU: Arithmetic and Logic Unit)
- Các thanh ghi (Registers)
- Bus trong CPU

CPU

**Vi xử lý Intel
8086 (1978)**



**Vi xử lý Intel
Core 2 Duo
(2006)**



2. Cấu trúc và các thành phần chức năng

□ Bộ nhớ trong:

- Lưu trữ lệnh và dữ liệu để CPU xử lý

- Bao gồm:

- ROM – Read Only Memory:

- Lưu trữ lệnh và dữ liệu của hệ thống
 - Thông tin trong ROM vẫn tồn tại khi mất nguồn nuôi

- RAM – Random Access Memory:

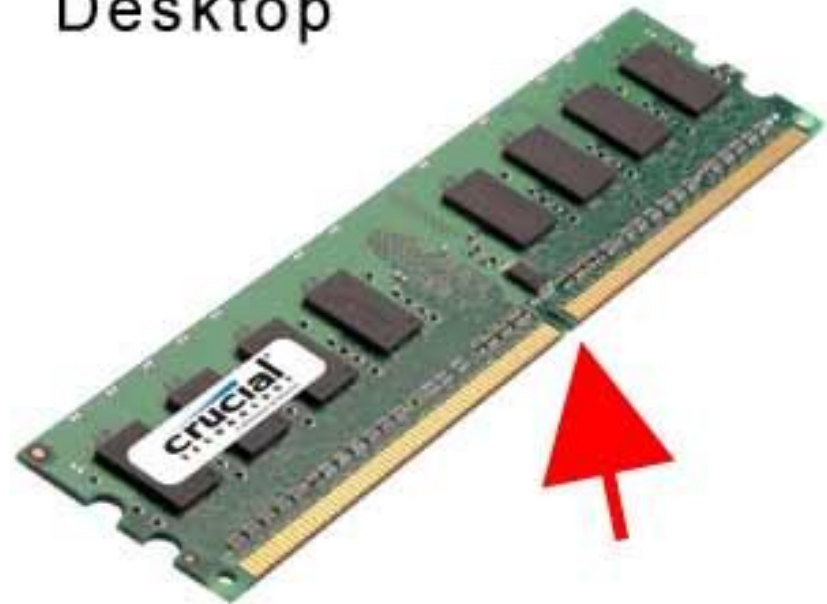
- Lưu trữ lệnh và dữ liệu của hệ thống và người dùng
 - Thông tin trong RAM sẽ mất khi mất nguồn nuôi

Các thành phần chính – bộ nhớ trong

ROM-BIOS



Crucial 240 pin DIMM
Desktop



2. Cấu trúc và các thành phần chức năng

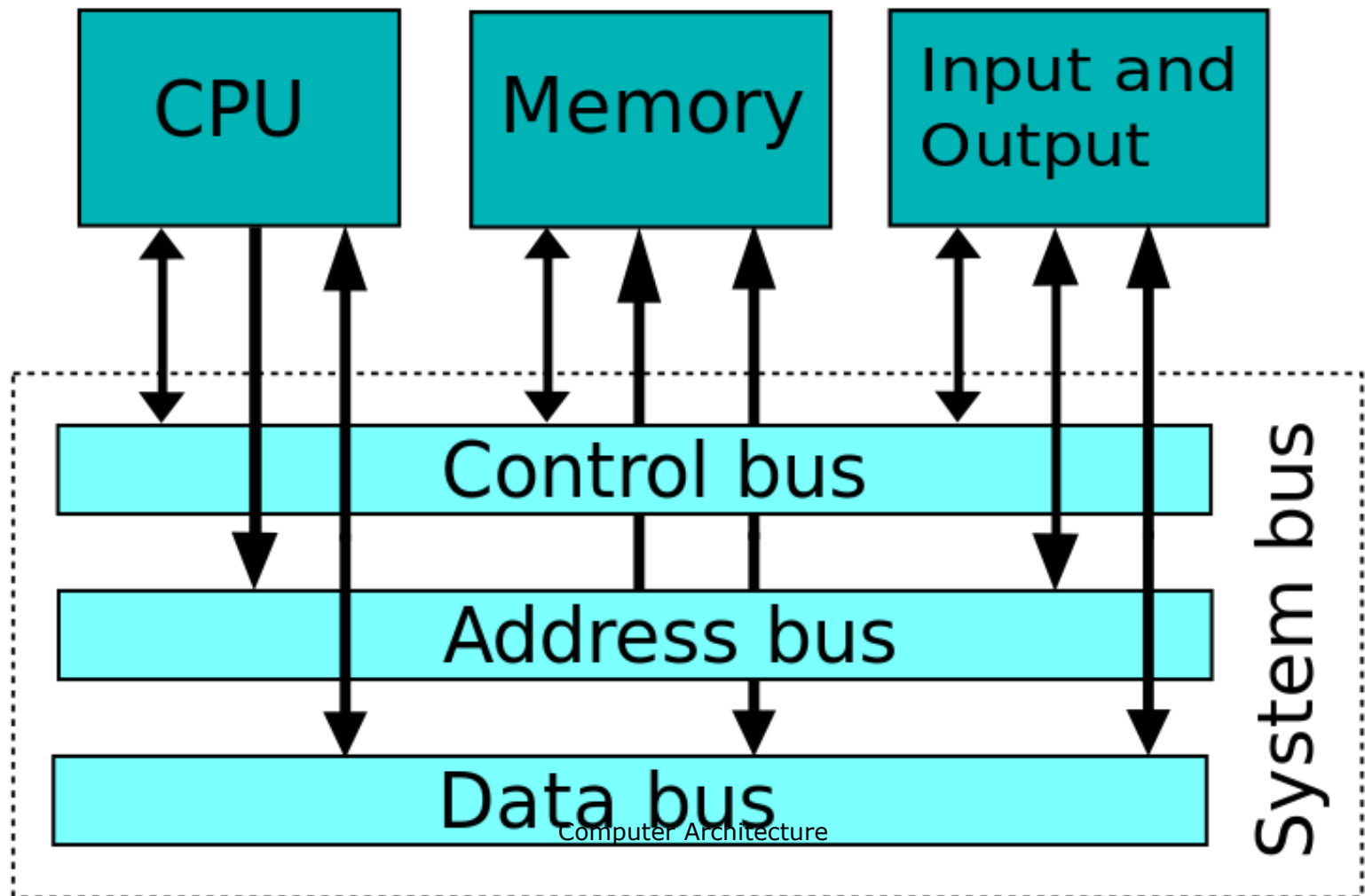
□ Các thiết bị vào ra:

- Thiết bị vào (input devices): nhập dữ liệu và điều khiển
 - Bàn phím
 - Chuột
 - Ổ đĩa
 - Máy quét
- Thiết bị ra: kết xuất dữ liệu
 - Màn hình
 - Máy in
 - Ổ đĩa

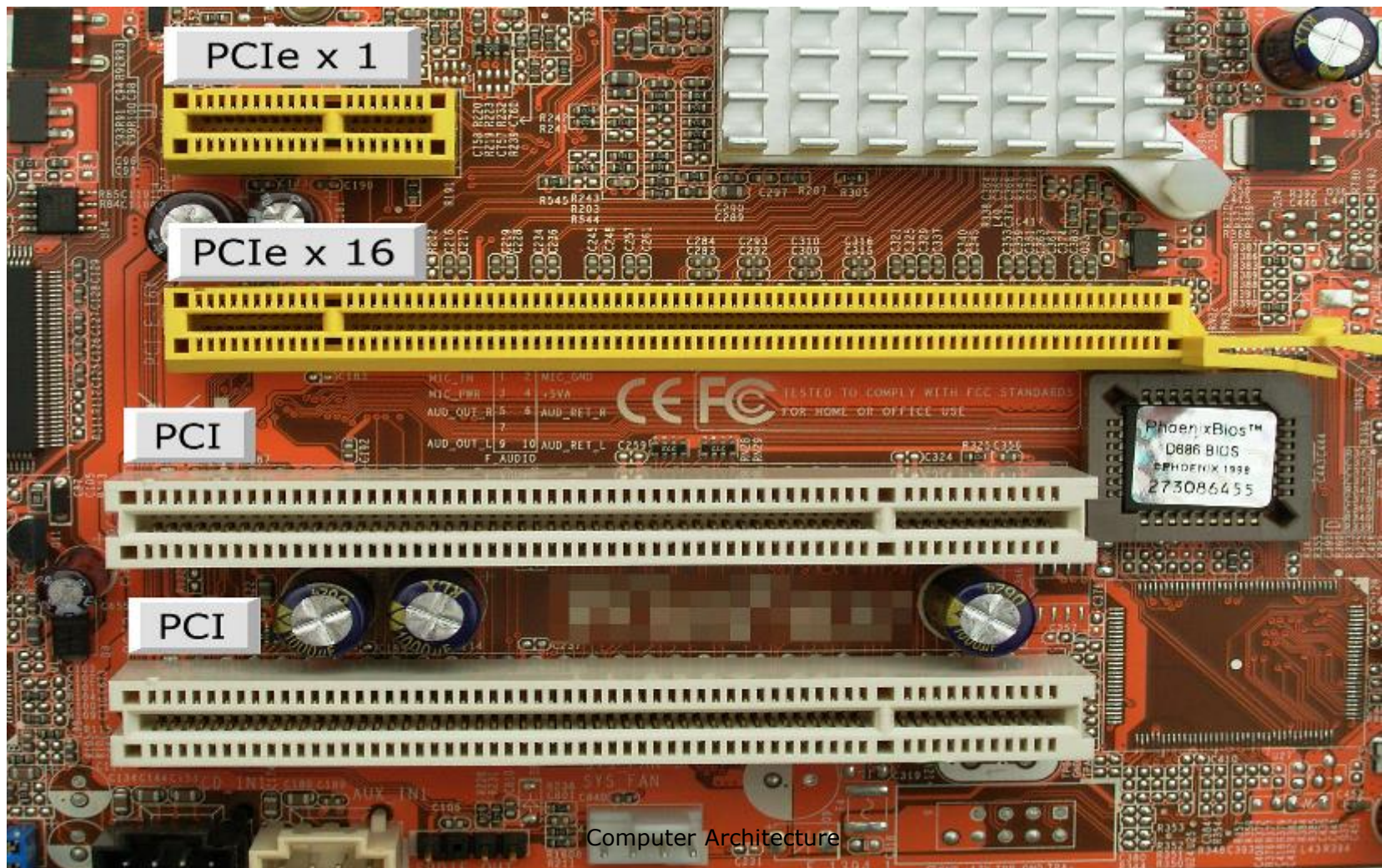
Các thành phần chính – bus hệ thống

- Tập các đường dây kết nối CPU với các thành phần khác của máy tính
- Bao gồm 3 loại:
 - ▣ Bus địa chỉ (gọi là bus A)
 - ▣ Bus dữ liệu (gọi là bus D)
 - ▣ Bus điều khiển (bus C)

Bus hệ thống



PCI bus



5. Phân loại máy tính

Phân loại máy tính kỹ nguyên PC

- ❑ Máy tính cá nhân (Personal Computers)
 - Desktop computers, Laptop computers
 - Máy tính đa dụng
- ❑ Máy chủ (Servers) – máy phục vụ
 - Dùng trong mạng để quản lý và cung cấp các dịch vụ
 - Hiệu năng và độ tin cậy cao
 - Hàng nghìn đến hàng triệu USD
- ❑ Siêu máy tính (Supercomputers)
 - Dùng cho tính toán cao cấp trong khoa học và kỹ thuật
 - Hàng triệu đến hàng trăm triệu USD
- ❑ Máy tính nhúng (Embedded Computers)
 - Đặt ẩn trong thiết bị khác
 - Được thiết kế chuyên dụng

Phân loại máy tính – kỷ nguyên sau PC

▣ Thiết bị di động cá nhân (PMD - Personal Mobile Devices)

- Smartphones, Tablet
- Kết nối Internet

▣ Điện toán đám mây (Cloud Computing)

- Sử dụng máy tính qui mô lớn (Warehouse Scale Computers), gồm rất nhiều servers kết nối với nhau
- Cho các công ty thuê một phần để cung cấp dịch vụ phần mềm
- Software as a Service (SaaS): một phần của phần mềm chạy trên PMD, một phần chạy trên Cloud Ví dụ: Amazon, Google

What Will We Learn in This Course?

- Recap – Tóm tắt lại

Answer

- How Computers Work?
- Why We Care?
- Why Do We Have Computer?
- Why Do We Do Computing?
- How Does a Computer Solve Problems?

Làm thế nào máy tính điều giải quyết một vấn đề?

□ Điều phối Electrons (Orchestrating Electrons)

Điều phối sắp xếp electrons trong thống trị công nghệ máy tính hiện nay.

Kích hoạt electrons làm những điều đó dẫn đến các giải pháp cho vấn đề.

Kiến trúc máy tính

Problem
Algorithm
Program/Language
Runtime System (VM, OS, MM)
ISA (Architecture)
Microarchitecture
Logic
Circuits
Electrons

- ✦ Từ vấn đề thực tế
- ✦ Thuật toán
- ✦ Phát triển chương trình
- ✦ Môi trường chạy chương trình
- ✦ Thực hiện trong phần cứng (CPU Architecture)

Tám ý tưởng lớn (Computer Architecture)

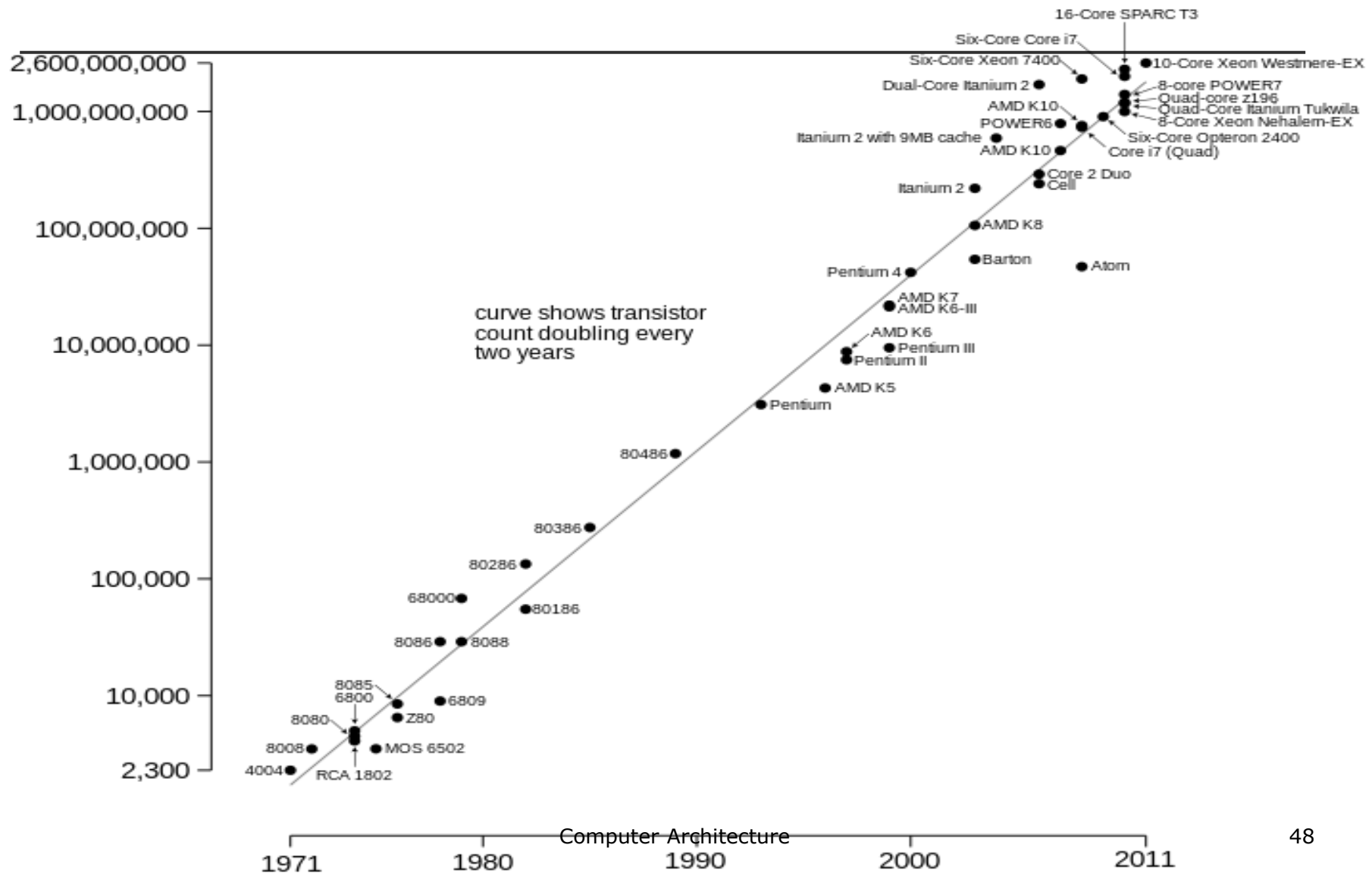
- Tìm ra luật Moore
- Trừu tượng hoá (abstraction layers) đơn giản thiết kế
- Tập trung tối ưu hệ thống chung (common case)
- Tăng hiệu năng qua
 - Thiết kế song song
 - Thiết kế pipeline
 - Thiết kế dự đoán (prediction)
 - Thiết kế bộ nhớ có cấu trúc
 - Thiết kế dự phòng (dependability via redundancy)

Luật Moore

- ❑ Quan sát bởi Gordon Moore, CEO của Intel, 1965
- ❑ Số transistors trên chip sẽ tăng gấp đôi trong $\sim 18m$
- ❑ Giá thành của chip hầu như không thay đổi
- ❑ Mật độ cao hơn, do vậy đường dẫn ngắn hơn
- ❑ Kích thước IC nhỏ hơn dẫn tới độ phức tạp tăng lên
- ❑ Điện năng tiêu thụ ít hơn
- ❑ Thiết kế tối ưu hơn

Luật Moore

Microprocessor Transistor Counts 1971-2011 & Moore's Law



6. Thiết kế kiến trúc máy tính

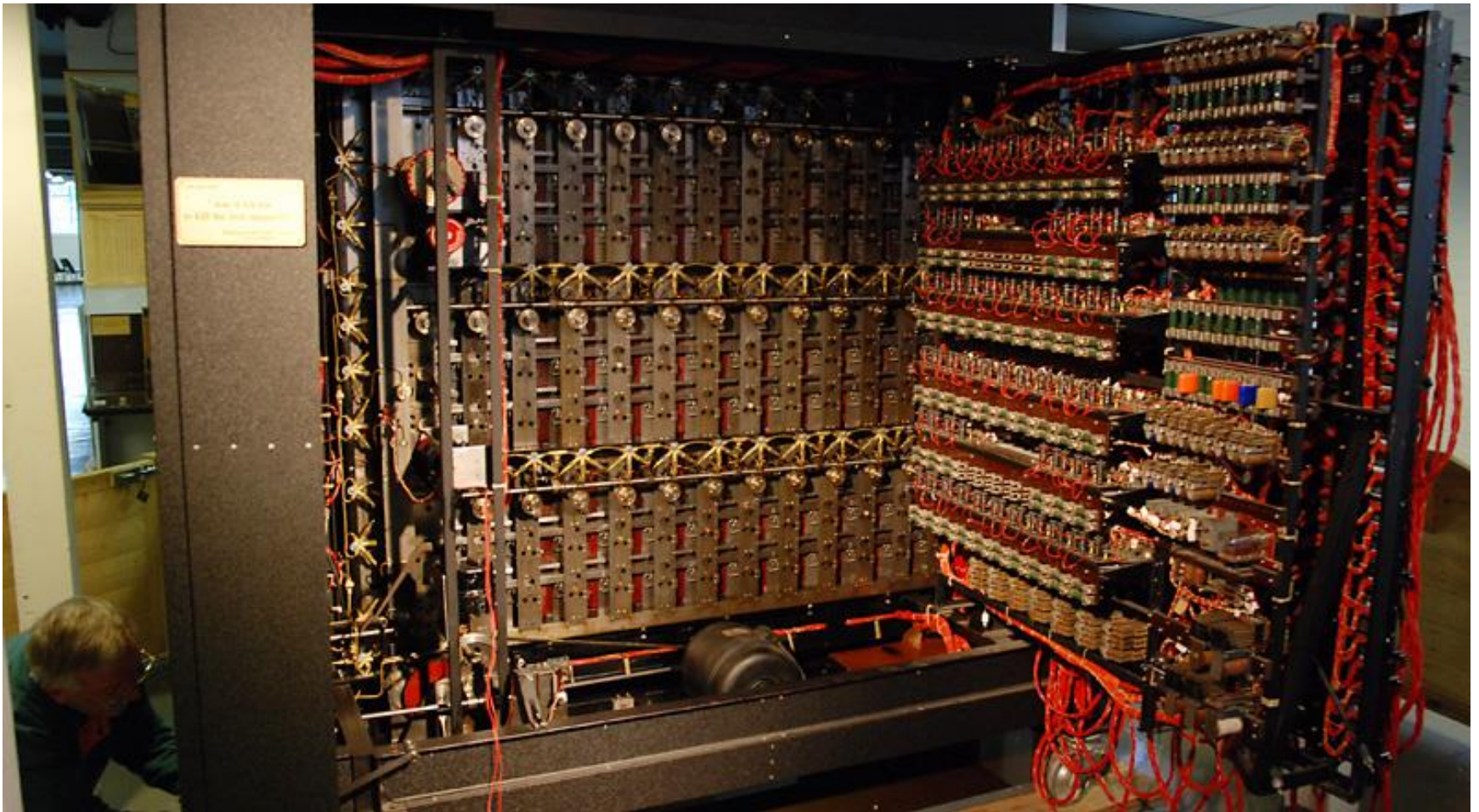
6. Thiết kế kiến trúc máy tính

- Theo sự phát triển, có nhiều loại kiến trúc ra đời -> nghiên cứu 2 kiến trúc chính sau
- **Kiến trúc Von-Neuman**
- **Kiến trúc Harvard**
- ... (>20 kiến trúc máy tính khác nhau)
- Von Neumann Architecture - Computer Science GCSE GURU

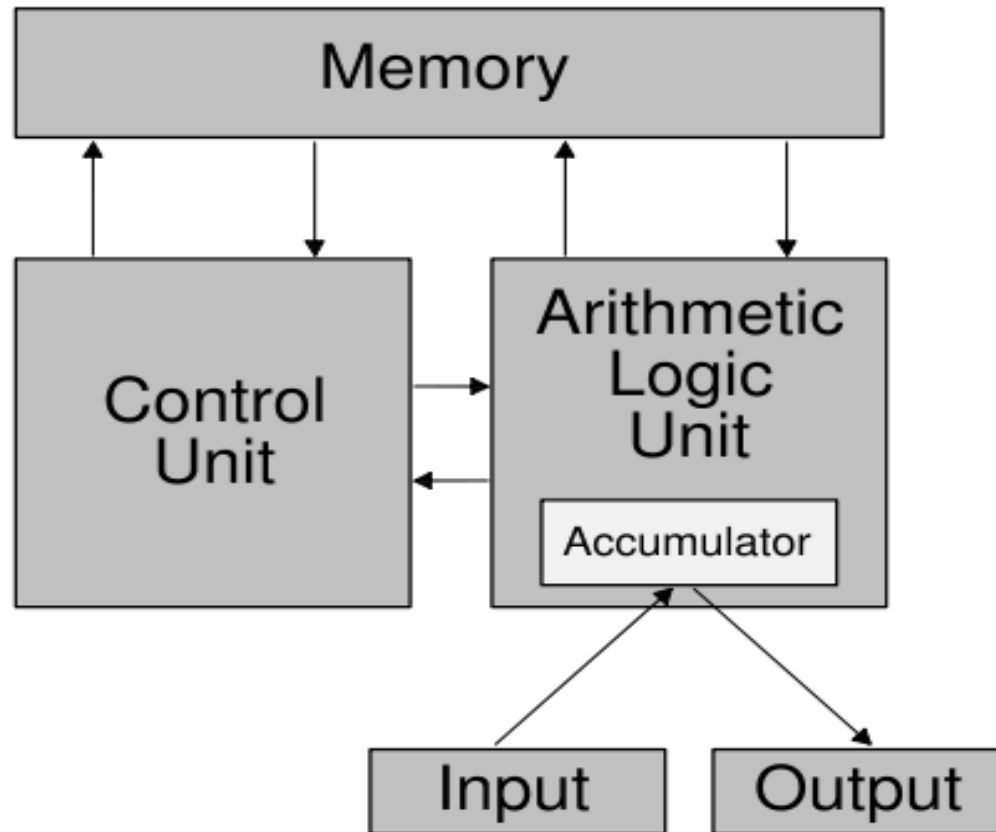
Kiến trúc Von-Neumann

- ❑ Kiến trúc Von-Neumann được giới thiệu bởi John von-Neumann vào năm 1945.
- ❑ Các máy tính kiến trúc Von-Neumann dựa trên 3 khái niệm cơ bản:
 - Dữ liệu và lệnh được lưu trong một bộ nhớ đọc/viết chia sẻ
 - Bộ nhớ được đánh địa chỉ dựa trên đoạn và không phụ thuộc vào việc nó lưu trữ gì
 - Các lệnh của chương trình được chạy lần lượt, lệnh nọ tiếp sau lệnh kia
 - → Stored-program digital computer

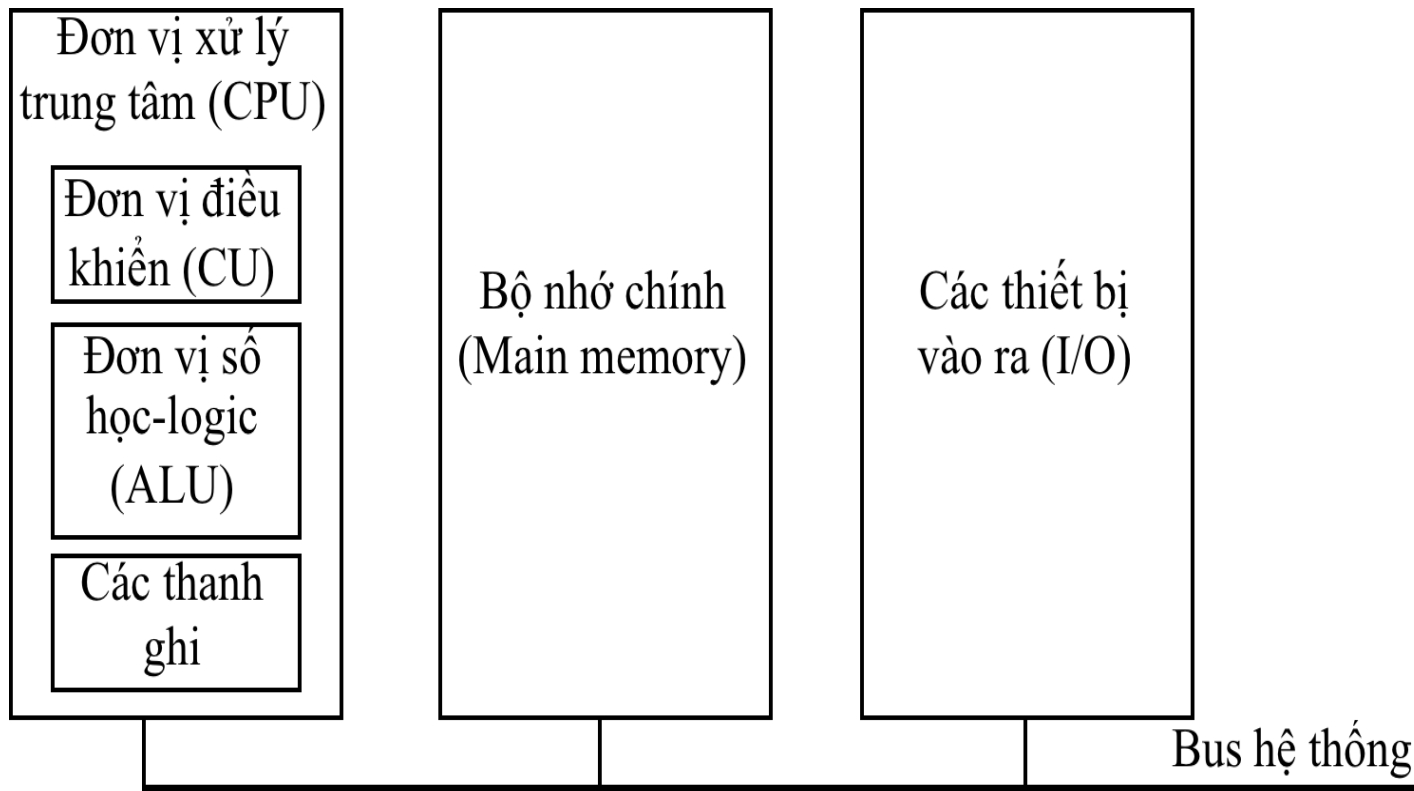
Máy tính trước đó (circus-based)



Kiến trúc Von-Neumann cũ



Kiến trúc Von-Neumann hiện đại



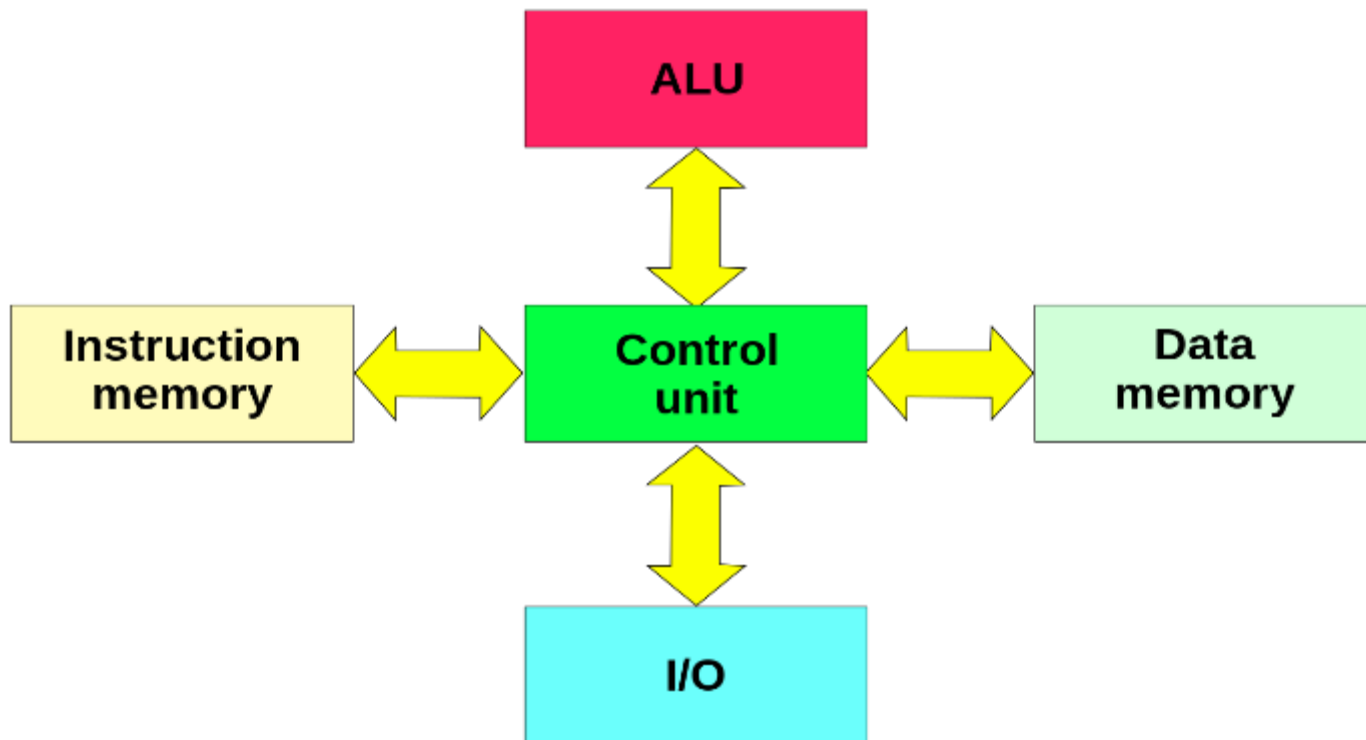
Kiến trúc Von-Neumann - Các đặc điểm

- ❑ Quá trình thực hiện lệnh được chia thành 3 giai đoạn chính :
 - CPU lấy lệnh (fetch) từ bộ nhớ
 - CPU giải mã lệnh và chạy lệnh; nếu lệnh cần dữ liệu thì đọc dữ liệu từ bộ nhớ
 - CPU viết kết quả vào bộ nhớ nếu có
- ❑ Hạn chế: bộ nhớ lệnh và dữ liệu (cổ chai) ko đc truy cập cùng lúc nên thông lượng (throughput) nhỏ hơn rất nhiều so với tốc độ CPU có thể làm việc
- ❑ Khắc phục: Dùng bộ nhớ cache giữa CPU và main memory

Kiến trúc Harvard

- Khắc phục được khuyết điểm của kiến trúc Von-Neumann
- Bộ nhớ được chia thành 2 phần:
 - Bộ nhớ chương trình
 - Bộ nhớ dữ liệu
- CPU sử dụng 2 bus hệ thống để liên hệ với bộ nhớ:
 - CPU có thể đọc lệnh và truy cập dữ liệu bộ nhớ cùng một lúc.
 - Một bus A,D cho bộ nhớ chương trình và 1 bus A,D cho bộ nhớ dữ liệu (khác nhau về định dạng)

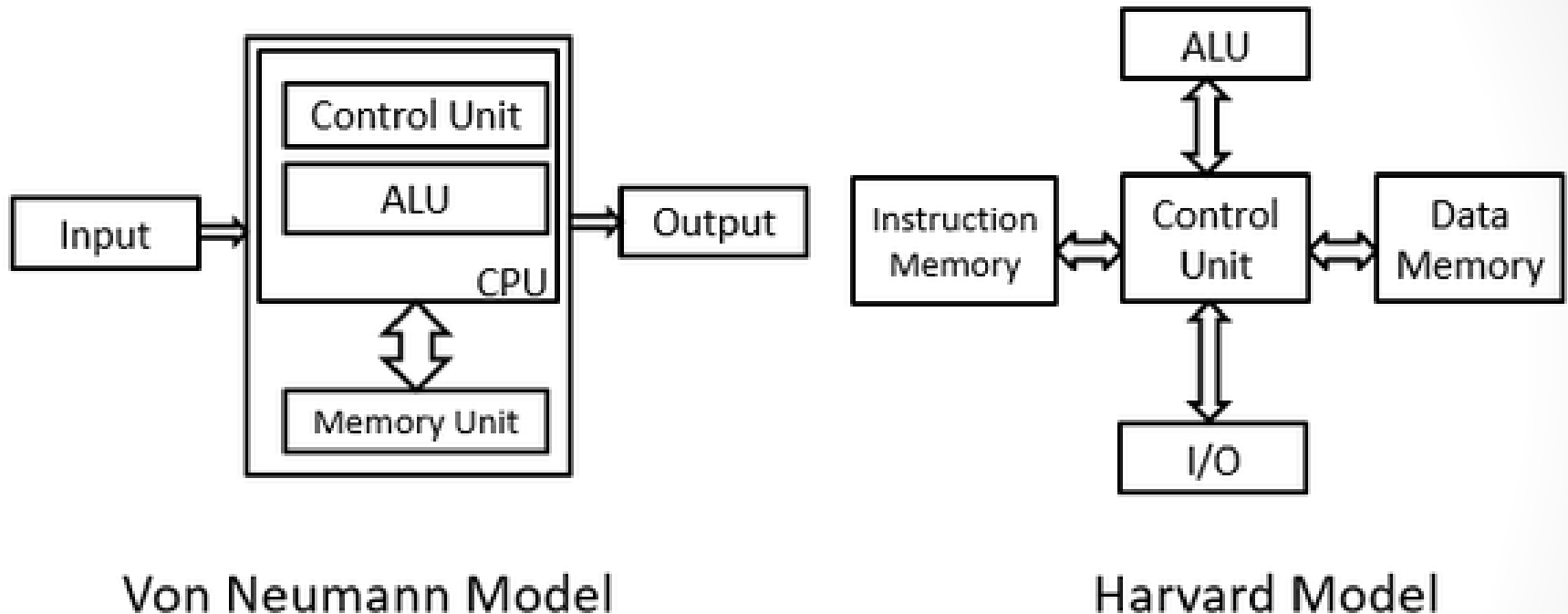
Kiến trúc Harvard



Kiến trúc Harvard

- Nhanh hơn vì băng thông bus rộng vì quá trình đọc lệnh không tranh chấp với quá trình truy xuất dữ liệu
- Hỗ trợ nhiều truy cập đọc/viết bộ nhớ cùng lúc → giảm xung đột truy cập bộ nhớ
- Ngày nay kiến trúc Harvard cải tiến được ứng dụng cho các kiến trúc máy tính hiện đại: ARM, intel x86
- Kiến trúc Harvard cũng được ứng dụng ở các hệ thống nhúng embedded system, chip chuyên xử lý tín hiệu (DSP)

So sánh Von Neumann và Harvard



Von Neumann: địa chỉ và bộ nhớ chia sẻ dữ liệu và lệnh từ CPU

Harvard: 2 địa chỉ và bộ nhớ cho dữ liệu và CPU → xử lý song song

So Sánh

Havard Architecture	Von Neumann Architecture
Harvard architecture has physically separate pathways for instructions and data.	Von Neumann architecture uses same physical pathways for instructions and data .
It has one dedicated set of address and data bus for reading data from and writing data to memory, and another set of address and data buses for fetching instructions.	It has same set of data and address buses for memory read/write and fetching instructions.
Under harvard architecture the CPU can both read an instruction and perform a data memory access at the same time .	Under pure von Neumann architecture the CPU can be either reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time since the instructions and data use the same bus system.
Harvard architecture machine has distinct code and data address spaces: instruction address zero is not the same as data address zero.	Von Neumann architecture has same data address and instruction address .

7. Biểu diễn dữ liệu trong máy tính

Biểu diễn dữ liệu trong máy tính

- Sử dụng hệ nhị phân để biểu diễn dữ liệu
- Hệ nhị phân sử dụng 2 kí tự 0 và 1; 0 biểu diễn giá trị logic False; 1 biểu diễn giá trị logic True.
- Hệ hexa cũng được sử dụng; gồm 16 kí tự: 0-9, A, B, C, D, E, F.

Các hệ đếm cơ bản

- Hệ thập phân (Decimal System)
→ con người sử dụng
- Hệ nhị phân (Binary System)
→ máy tính sử dụng
- Hệ mười sáu (Hexadecimal System)
→ dùng để viết gọn cho số nhị phân

1. Hệ thập phân

- Cơ số 10
- 10 chữ số: 0,1,2,3,4,5,6,7,8,9
- Dùng n chữ số thập phân có thể biểu diễn được 10^n giá trị khác nhau:
 - $00\dots000 = 0$
 - $99\dots999 = 10^n - 1$

Dạng tổng quát của số thập phân

$$A = a_n a_{n-1} \dots a_1 a_0 , a_{-1} \dots a_{-m}$$

Giá trị của A được hiểu như sau:


$$A = a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0 + a_{-1} 10^{-1} + \dots + a_{-m} 10^{-m}$$

$$A = \sum_{i=-m}^n a_i 10^i$$


Ví dụ số thập phân

$$472.38 = 4 \times 10^2 + 7 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2}$$

■ Các chữ số của phần nguyên:

- $472 : 10 = 47$ dư 2
 - $47 : 10 = 4$ dư 7
 - $4 : 10 = 0$ dư 4
- 

■ Các chữ số của phần lẻ:

- $0.38 \times 10 = 3.8$ phần nguyên = 3
 - $0.8 \times 10 = 8.0$ phần nguyên = 8
- 

2. Hệ nhị phân

- Cơ số 2
- 2 chữ số nhị phân: 0 và 1 (binary digit)
- chữ số nhị phân gọi là **bit**
- Bit là đơn vị thông tin nhỏ nhất
- Dùng n bit có thể biểu diễn được 2^n giá trị khác nhau:
 - $00\dots000 = 0$
 - $11\dots111 = 2^n - 1$

Bits, Bytes, Nibbles...

■ Bits

10010110

most significant bit least significant bit

■ Bytes & Nibbles

byte

10010110

nibble

■ Bytes

CEBF9AD7

most significant byte least significant byte

Lũy thừa hai

- $2^{10} = 1 \text{ Ki}$ $\approx 1000 \text{ (1024)}$
- $2^{20} = 1 \text{ Mi}$ $\approx 1 \text{ triệu (1,048,576)}$
- $2^{30} = 1 \text{ Gi}$ $\approx 1 \text{ tỷ}$
- $2^{40} = 1 \text{ Ti}$ $(1,073,741,824)$
- $2^{50} = 1 \text{ Pi}$ $\approx 1000 \text{ tỷ}$
 $\approx 1 \text{ triệu tỷ}$

Dạng tổng quát của số nhị phân

Có một số nhị phân A như sau:

$$A = a_n a_{n-1} \dots a_1 a_0, a_{-1} \dots a_{-m}$$

Giá trị của A được tính như sau:

$$A = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + \dots + a_{-m} 2^{-m}$$

$$A = \sum_{i=-m}^n a_i 2^i$$

Ví dụ số nhị phân

$$\underset{\substack{6\ 5\ 4\ 3\ 2\ 1\ 0\ -1\ -2\ -3\ -4}}{1101001.1011}_{(2)} =$$

$$= 1x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 1x2^{-4}$$

$$= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4}$$

$$= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625$$

$$= 105.6875_{(10)}$$

Chuyển đổi số nguyên thập phân sang nhị phân

- Phương pháp 1: chia dần cho 2 rồi lấy phần dư
- Phương pháp 2: Phân tích thành tổng của các số 2^i → nhanh hơn

Phương pháp chia dần cho 2

Ví dụ: chuyển đổi

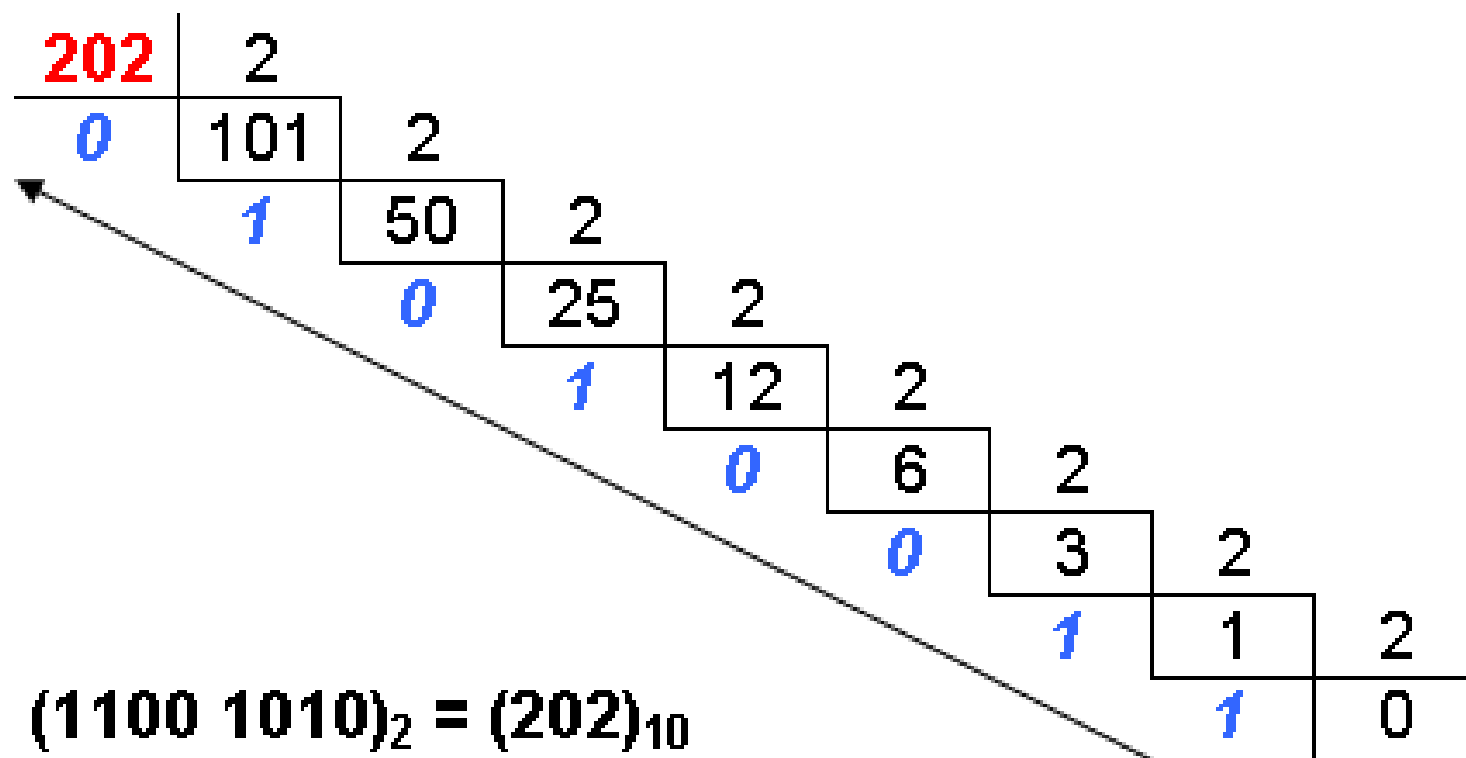
$105,6875_{(10)}$

$105 : 2 = 52$	dư 1	$0.6875 \times 2 = 1.375$	phần nguyên = 1
$52 : 2 = 26$	dư 0	$0.375 \times 2 = 0.75$	phần nguyên = 0
$26 : 2 = 13$	dư 0	$0.75 \times 2 = 1.5$	phần nguyên = 1
$13 : 2 = 6$	dư 1	$0.5 \times 2 = 1.0$	phần nguyên = 1
$6 : 2 = 3$	dư 0		
$3 : 2 = 1$	dư 1		
$1 : 2 = 0$	dư 1		

Kết quả:

$105_{(10)} = 1101001, 1011_{(2)}$

Chuyển đổi số thập phân sang nhị phân



Phương pháp phân tích thành tổng của các 2^i

- Ví dụ 1: chuyển đổi $105,6875_{(10)}$
 - $105,6875 = 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625$
 $= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-2} + 2^{-4}$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
128	64	32	16	8	4	2	1	0.5	0.25	0.125	0.0375
0	1	1	0	1	0	0	1	1	0	1	1

- Kết quả: $105_{(10)} = 01101001,1011_{(2)}$

Chuyển đổi số thập phân sang nhị phân

- Chuyển $(0.6875)_{10} \rightarrow (.)_2?$
- Chuyển $(0.81)_{10} \rightarrow (.)_2?$
- Chuyển $(0.2)_{10} ?$

3. Hệ mười sáu (Hexa)

- Cơ số 16
- 16 chữ số: 0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
- Dùng để viết gọn cho số nhị phân: cứ một nhóm 4-bit sẽ được thay bằng một chữ số Hexa

Hệ Hexa

- Mỗi kí hiệu trong hệ hexa được biểu diễn bởi 4 kí hiệu trong hệ nhị phân

Hexa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Quan hệ giữa số nhị phân và số Hexa

4-bit	Chữ số Hexa
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Ví dụ chuyển đổi số nhị phân \rightarrow số Hexa:

- $1011\ 0011_2 = B3_{16}$

- $0000\ 0000_2 = 00_{16}$

- $0010\ 1101\ 1001\ 1010_2 = 2D9A_{16}$

- $1111\ 1111\ 1111\ 1111_2 = FFFF_{16}$

Tổ chức dữ liệu

□ Bits:

- Là đơn vị dữ liệu nhỏ nhất
- Một bit chỉ có thể lưu trữ 2 giá trị: 0 hoặc 1, true hoặc false.

□ Nibbles:

- Nhóm 4 bits
- Có thể lưu trữ tới 16 giá trị từ $(0000)_2$ tới $(1111)_2$, hoặc 1 số hệ hexa.

Data Organization

□ Bytes:

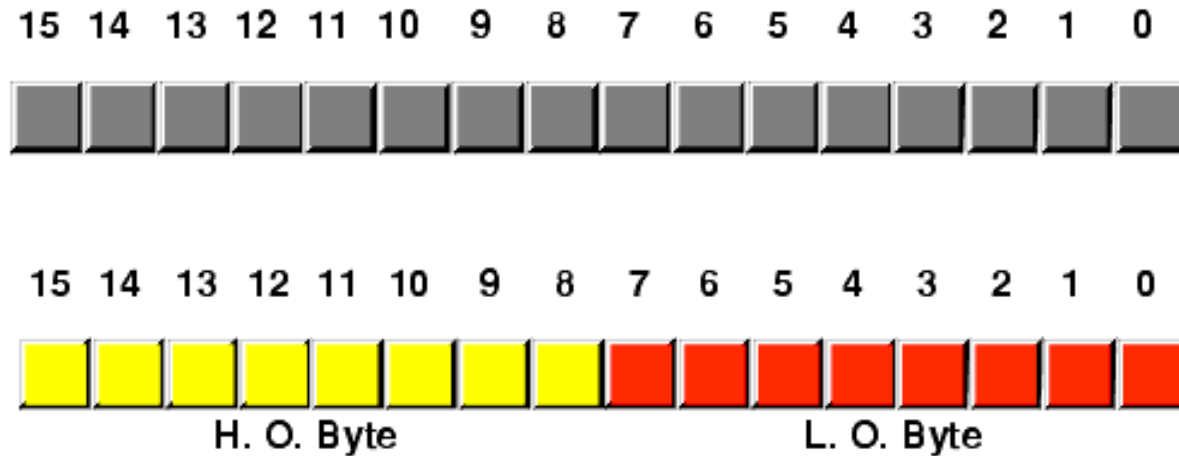
- Nhóm 8 bit hoặc 2 nibbles
- Có thể lưu tới 256 giá trị, từ $(0000\ 0000)_2$ tới $(1111\ 1111)_2$, hoặc từ $(00)_{16}$ tới $(FF)_{16}$.



Data Organization

□ Words (từ):

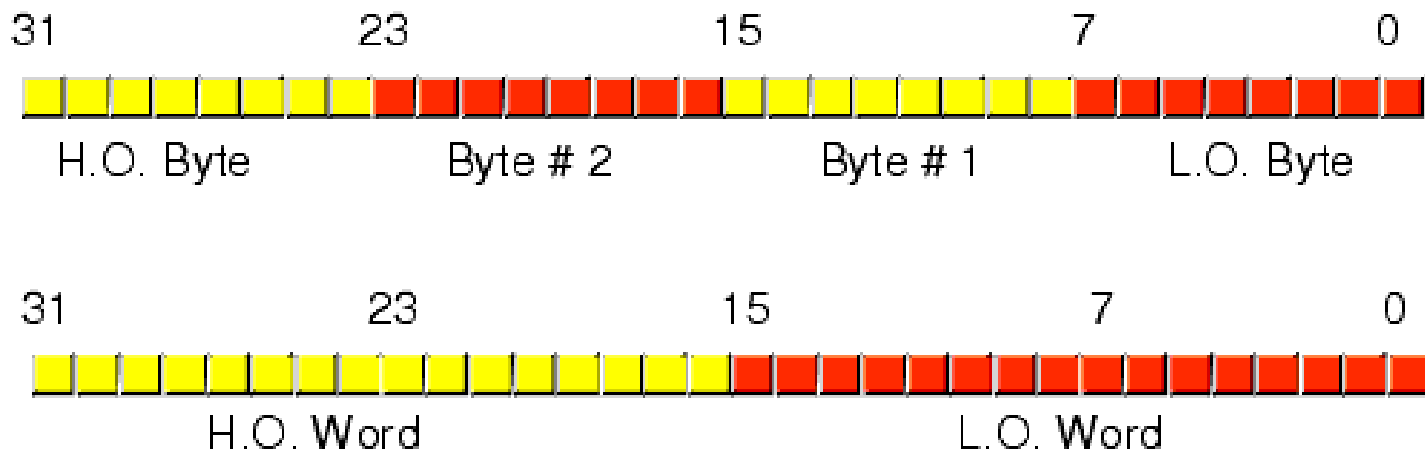
- Nhóm 16 bits, hay 2 bytes
- Có thể lưu tới 2^{16} (65536) values, từ $(0000)_{16}$ tới $(FFFF)_{16}$.



Data Organization

□ Double words:

- A double word nhóm 32 bits, hoặc 4 bytes, hoặc 2 words
- Có thể lưu tới 2^{32} values, từ $(0000\ 0000)_{16}$ tới $(FFFF\ FFFF)_{16}$.



Số có dấu và không dấu

- Trong hệ nhị phân, bit trái nhất được dùng để biểu diễn dấu của số có dấu
 - Bit trái nhất là 0 \rightarrow số dương
 - Bit trái nhất là 1 \rightarrow số âm
- Ví dụ: sử dụng 4 bit để biểu diễn các số
 - 0011, 0111, 0101 là các số dương
 - 1011, 1111, 1101 là các số âm
- Đối với các số không dấu, tất cả các bit đều lưu giá trị

Số có dấu và không dấu

- Phạm vi biểu diễn: n bits có thể biểu diễn:
 - Số có dấu: từ -2^{n-1} tới $+2^{n-1}-1$
 - 8 bits: từ -128 tới +127
 - 16 bits: từ -32768 tới +32767
 - 32 bits: từ -2,147,483,648 tới +2,147,483,647
 - Số không dấu: từ 0 tới 2^n
 - 8 bits: từ 0 tới 256
 - 16 bits: từ 0 tới 65536
 - 32 bits: từ 0 tới 4,294,967,296

Bảng mã ASCII

- ❑ ASCII (American Standard Code for Information Interchange) là bảng mã các kí tự chuẩn tiếng Anh dùng cho trao đổi dữ liệu trong các hệ thống tính toán.
- ❑ Sử dụng 8 bit để biểu diễn 1 kí tự.
- ❑ Mã ASCII gồm định nghĩa cho 128 kí tự :
 - 33 kí tự điều khiển
 - 94 kí tự
- ❑ Các giá trị còn lại (129-255) dự trữ

ASCII Table – Control chars

Binary	Oct	Dec	Hex	Abbr	PR ^[t 1]	CS ^[t 2]	CEC ^[t 3]	Description
000 0000	000	0	00	NUL	NUL	^@	\0	Null character
000 0001	001	1	01	SOH	SOH	^A		Start of Header
000 0010	002	2	02	STX	STX	^B		Start of Text
000 0011	003	3	03	ETX	ETX	^C		End of Text
000 0100	004	4	04	EOT	EOT	^D		End of Transmission
000 0101	005	5	05	ENQ	ENQ	^E		Enquiry
000 0110	006	6	06	ACK	ACK	^F		Acknowledgment
000 0111	007	7	07	BEL	BEL	^G	\a	Bell
000 1000	010	8	08	BS	BS	^H	\b	Backspace ^{[t 4][t 5]}
000 1001	011	9	09	HT	HT	^I	\t	Horizontal Tab
000 1010	012	10	0A	LF	LF	^J	\n	Line feed

ASCII Table – Printable chars

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	SP
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	\$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l

Ví dụ

- Ví dụ 1. Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:

$$A = 41 ; B = 150$$

Giải:

Ví dụ

- Ví dụ 1. Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:

$$A = 41 ; B = 150$$

Giải:

$$A = 41 = 32 + 8 + 1$$

$$41 = 0010\ 1001$$

$$B = 150 = 128 + 16 + 4 + 2$$

$$150 = 1001\ 0110$$

Bài Tập Chương 1

- Với $n=8, 16, 32$ bit,
 1. Thiết kế bộ cộng nhị phân (có dấu, không dấu) n -bit
 2. Thiết kế bộ trừ nhị phân (có dấu, không dấu) n bit
 3. Thiết kế bộ nhân nhị phân (có dấu, không dấu) n -bit
 4. Thiết kế bộ trừ nhị phân (có dấu, không dấu) n -bit

Hiệu năng máy tính

- Định nghĩa hiệu năng P(Performance):

$$P = \frac{1}{T} \quad \text{trong đó: } t \text{ là thời gian thực hiện}$$

- “Máy tính A nhanh hơn máy B n lần”

$$\text{Hiệu năng}_A / \text{Hiệu năng}_B = P_A / P_B$$

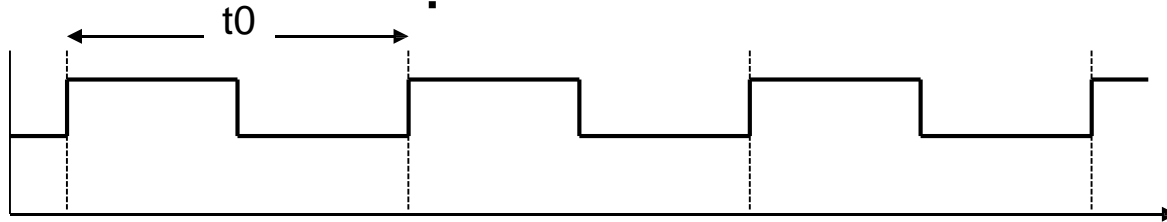
$$= \text{Thời gian thực hiện}_B / \text{Thời gian thực hiện}_A = T_B / T_A = n$$

- Ví dụ: Thời gian chạy chương trình:

- 10s trên máy A, 15s trên máy B
- $t_B / t_A = 15s / 10s = 1.5$
- Vậy máy A nhanh hơn máy B 1.5 lần

Xung nhịp của CPU

- Hoạt động của CPU được điều khiển bởi xung nhịp có tần số xác định



- Chu kỳ xung nhịp T_0 (Clock period): thời gian của một chu kỳ
- Tần số xung nhịp f_0 (Clock rate): số chu kỳ trong 1 giây.
 - $f_0 = 1/t_0$
- VD: Bộ xử lý có $f_0 = 4\text{GHz} = 4000\text{MHz} = 4 \times 10^9\text{Hz}$
 $t_0 = 1/(4 \times 10^9) = 0.25 \times 10^{-9}\text{s} = 0.25\text{ns}$

Thời gian CPU (t_{CPU})

- Thời gian thực hiện của CPU
= < số chu kì xung nhịp > * < Chu kì xung nhịp >

$$T_{\text{CPU}} = n * t_0 = n/f_0$$

- trong đó: n là số chu kì xung nhịp
- Hiệu năng được tăng lên bằng cách:
 - Giảm số chu kì xung nhịp n
 - Tăng tần số xung nhịp f_0

Ví dụ

- Máy tính A:

- Tần số xung nhịp: $f_A = 2\text{GHz}$
- Thời gian của CPU: $t_A = 10\text{s}$

- Máy tính B

- Thời gian của CPU: $t_B = 6\text{s}$
- Số chu kỳ xung nhịp của B = 1.2 x Số chu kỳ xung nhịp của A

- Xác định tần số xung nhịp của máy B (f_B)?

- Giải:
$$T_{\text{CPU (A)}} = \frac{n(A)}{f_0(A)} = 10\text{s} = \frac{n(A)}{2\text{GHz}}$$
$$T_{\text{CPU (B)}} = \frac{n(B)}{f_0(B)} = 6\text{s} = \frac{1.2 * n(A)}{f_0(B)}$$

$$\frac{T_{\text{CPU (A)}}}{T_{\text{CPU (B)}}} = \frac{10\text{s}}{6\text{s}} = \frac{\frac{n(A)}{2}}{\frac{1.2 * n(A)}{f_0(B)}} \rightarrow f_0(B) = 4\text{GHz}$$

Bài tập

- Máy tính A:
 - Tần số xung nhịp: $f_A = 4\text{GHz}$
 - Thời gian của CPU: $t_A = 9\text{s}$
- Máy tính B
 - Thời gian của CPU: $t_B = 5\text{s}$
 - Số chu kỳ xung nhịp của B = 1.5 x Số chu kỳ xung nhịp của A
- Xác định tần số xung nhịp của máy B (f_B)?

Số lệnh và số chu kỳ trên một lệnh

- Số chu kỳ = Số lệnh x Số chu kỳ trên một lệnh

$$n = IC \times CPI$$

n - số chu kỳ, IC - số lệnh (Instruction Count), CPI - số chu kỳ trên một lệnh (Clock Cycles per Instruction)

- Thời gian thực hiện của CPU:

$$\langle \text{Thời gian thực hiện} \rangle = \langle \text{Số lệnh} \rangle * \langle \text{CPI}_{\text{trung bình}} \rangle * \langle \text{Chu kì xung nhịp} \rangle$$

$$T_{\text{CPU}} = n * T_0 = IC * CPI * T_0 = \frac{IC * CPI}{f_0}$$

- Trong trường hợp các lệnh khác nhau có CPI khác nhau, cần tính CPI trung bình

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 ?

$$\begin{aligned}T_A &= \text{IC} \times \text{CPI}_A \times t_A \\ &= \text{IC} \times 2.0 \times 250\text{ps} = \text{IC} \times 500\text{ps}\end{aligned}$$

$$\begin{aligned}T_B &= \text{IC} \times \text{CPI}_B \times t_B \\ &= \text{IC} \times 1.2 \times 500\text{ps} = \text{IC} \times 600\text{ps}\end{aligned}$$

$$\frac{T_B}{T_A} = \frac{\text{IC} \times 600\text{ps}}{\text{IC} \times 500\text{ps}} = 1.2$$

Vậy:
A nhanh hơn B 1.2 lần

Bài tập

- Máy tính A: $t_A = 400\text{ps}$, $\text{CPI}_A = 3.0$
- Máy tính B: $t_B = 600\text{ps}$, $\text{CPI}_B = 1.5$
- Cùng kiến trúc tập lệnh (ISA)
- Máy nào nhanh hơn và nhanh hơn bao nhiêu ?

Chi tiết hơn về CPI

- Nếu loại lệnh khác nhau có số chu kỳ khác nhau, ta có tổng số chu kỳ:

$$n = \sum_{i=1}^K (\text{CPI}_i \times \text{IC}_i)$$

- CPI trung bình:

$$\text{CPI}_{\text{TB}} = \frac{\langle \text{Tong so chu ki} \rangle}{\langle \text{Tong so lenh} \rangle} = \frac{n}{\text{IC}} = \sum_{i=1}^n \left(\text{CPI}_i \times \frac{\text{IC}_i}{\text{IC}} \right)$$

Ví dụ

- Cho bảng chỉ ra các dãy lệnh sử dụng các lệnh thuộc các loại A, B, C. Tính CPI trung bình?

Loại lệnh	A	B	C
CPI theo loại lệnh	1	2	3
IC trong dãy lệnh 1	2	1	2
IC trong dãy lệnh 2	4	1	1

- Dãy lệnh 1: IC = 5

- Số chu kỳ
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$

- $CPI_{TB} = 10/5 = 2.0$

- Dãy lệnh 2: IC = 6

- Số chu kỳ
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$

- $CPI_{TB} = 9/6 = 1.5$

Tóm tắt về Hiệu năng

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

$$t_{CPU} = IC \times CPI \times T_0 = \frac{IC \times CPI}{f_0}$$

- Hiệu năng phụ thuộc vào:
 - Thuật toán
 - Ngôn ngữ lập trình
 - Chương trình dịch
 - Kiến trúc tập lệnh

Vtune Amplifier: đo CPI trên chip Intel

