

Chương 1: Tổng quan về kiến trúc máy tính

Kiến trúc máy tính

ThS. Đinh Xuân Trường
truongdx@ptit.edu.vn



Posts and Telecommunications
Institute of Technology
Faculty of Information Technology 1



CNTT1
Học viện Công nghệ Bưu chính Viễn thông

January 15, 2023

Mục tiêu Buổi 1

Khái niệm kiến trúc và tổ chức máy tính

Lịch sử phát triển máy tính

Cấu trúc và chức năng của máy tính

Bộ xử lý trung tâm - CPU

Bộ xử nhớ trong - Memory

Các thiết bị vào ra

Bus hệ thống

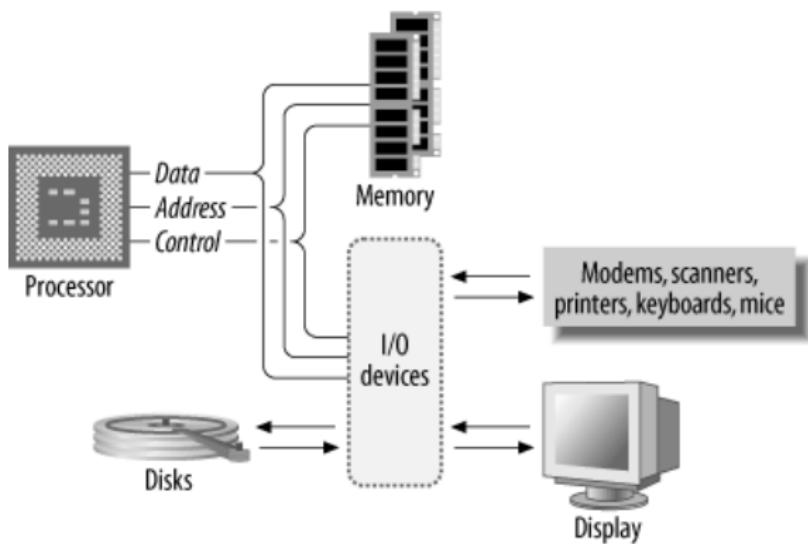
Phân loại máy tính

Thiết kế kiến trúc máy tính

Một số các hệ đếm

Nội dung chương 1

1. Khái niệm kiến trúc và tổ chức máy tính
2. Lịch sử phát triển máy tính
3. Cấu trúc và chức năng của máy tính
4. Một số các hệ đếm



Kiến trúc là gì???

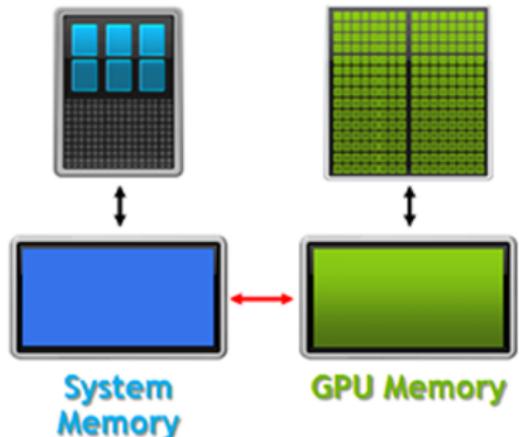


So sánh hai kiến trúc trên:

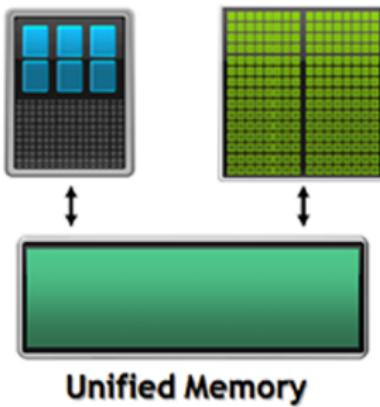
- ▶ **Giống nhau giữa hai kiến trúc:** Cùng chức năng một ngôi nhà (Ăn, ngủ, sinh hoạt)
- ▶ **Khác nhau là gì?** Vật liệu, Cách bố trí các thành phần, Giá thành.
- ▶ **Bài học rút ra:** Bắt đầu từ một nhu cầu đơn giản sau đó, cải tiến và hoàn thiện.

Kiến trúc là gì???

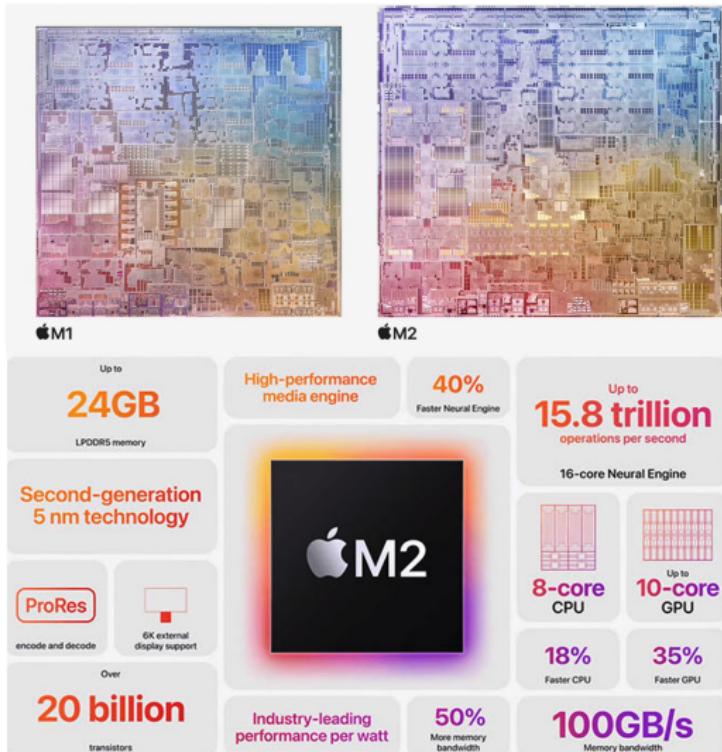
Traditional Developer View



Developer View With Unified Memory



Kiến trúc Apple:



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: IBM Thinkpad, Iphone, Android Phones là việc thiết kế hệ thống dòng sản phẩm

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: đáp ứng nhiều chức năng
- ▶ Giá thành: càng rẻ càng tốt

Ví dụ: Các thế hệ máy khác nhau với cùng kiến trúc phần cứng như Iphone 12, Iphone 13, Iphone 14...

Thành phần cơ bản của kiến trúc máy tính gồm 3 thành phần:

► **Kiến trúc tập lệnh** (ISA - Instruction Set Architecture): Mô hình trừu tượng của máy tính ở mức ngôn ngữ máy (hợp ngữ) xác định những gì bộ xử lý thực hiện và cách thực hiện:

- Tập lệnh
- Các chế độ địa chỉ bộ nhớ
- Các thanh ghi
- Khuôn dạng địa chỉ và dữ liệu

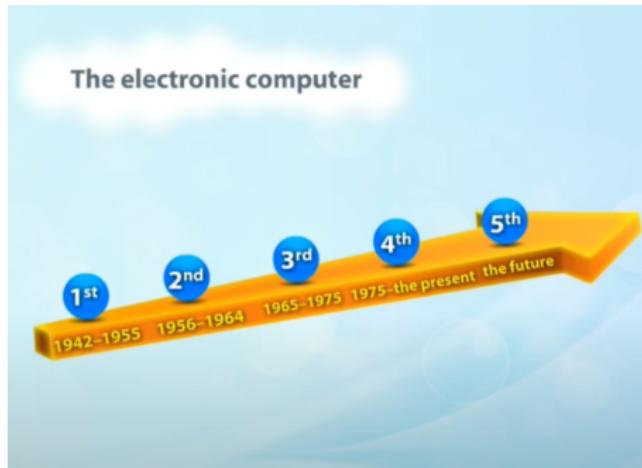
► **Vi kiến trúc** (Microarchitecture): là tổ chức máy tính, mô tả về hệ thống ở mức thấp, liên quan tới:

- 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

- ▶ **Thiết kế hệ thống:** bao gồm tất cả các thành phần phần cứng khác trong hệ thống máy tính:
 - 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

Lịch sử phát triển máy tính

Lịch sử phát triển của máy tính chia thành 5 thế hệ dựa trên sự phát triển mạch điện tử:



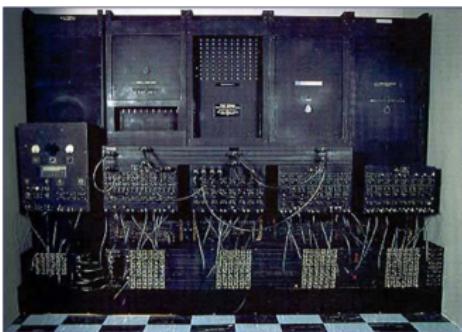
Generations of Computer

Lịch sử phát triển máy tính (cont.)

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/ *foot*³ (1 foot= 30.48 cm)
- ▶ Ví dụ: ENIAC - Electronic Numerical Integrator and Computer, 1946, giá 500,000 USD.

ENIAC



Lịch sử phát triển máy tính (cont.)

Thế hệ thứ 2 (1960-1964):

- ▶ Sử dụng transistors
- ▶ ~100,000 linh kiện / foot³
- ▶ 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 \$)

UNIVAC



Lịch sử phát triển máy tính (cont.)

Thế hệ thứ 3 (1964-1975):

- ▶ Sử dụng mạch tích hợp (IC)
- ▶ ~10 triệu linh kiện/ *foot*³
- ▶ Ví dụ: UNIVAC 9000 series, IBM System/360, System 3, System 7

UNIVAC 9400



Lịch sử phát triển máy tính (cont.)

Thế hệ thứ 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

Cray 2 XMP



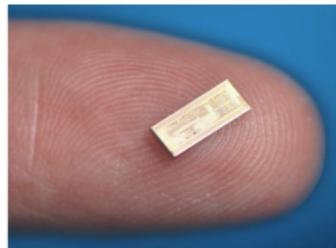
Thế hệ thứ 5 (1990- nay):

- ▶ Sử dụng VLSI – Very Large Scale Integrated Circuit
- ▶ 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

Intel



Intel Core i9-13980HX



Intel Atom

Lịch sử phát triển máy tính (cont.)

Luật Moore

- ▶ Quan sát bởi Gordon Moore, CEO của Intel, 1965
- ▶ Số transistors trên chip sẽ tăng gấp đôi trong 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

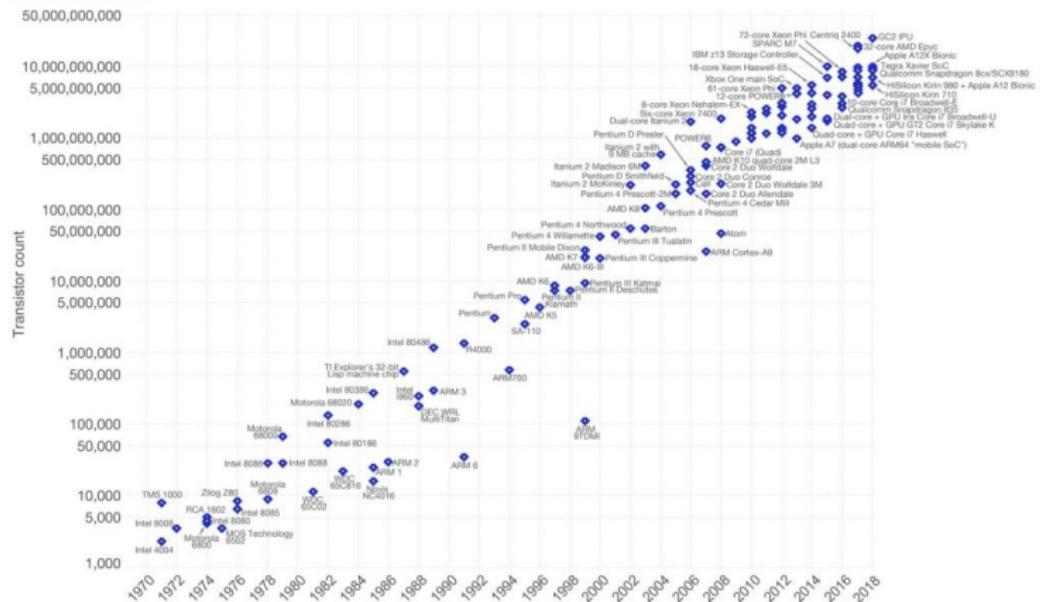
Lịch sử phát triển máy tính (cont.)



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.

Our World
in Data

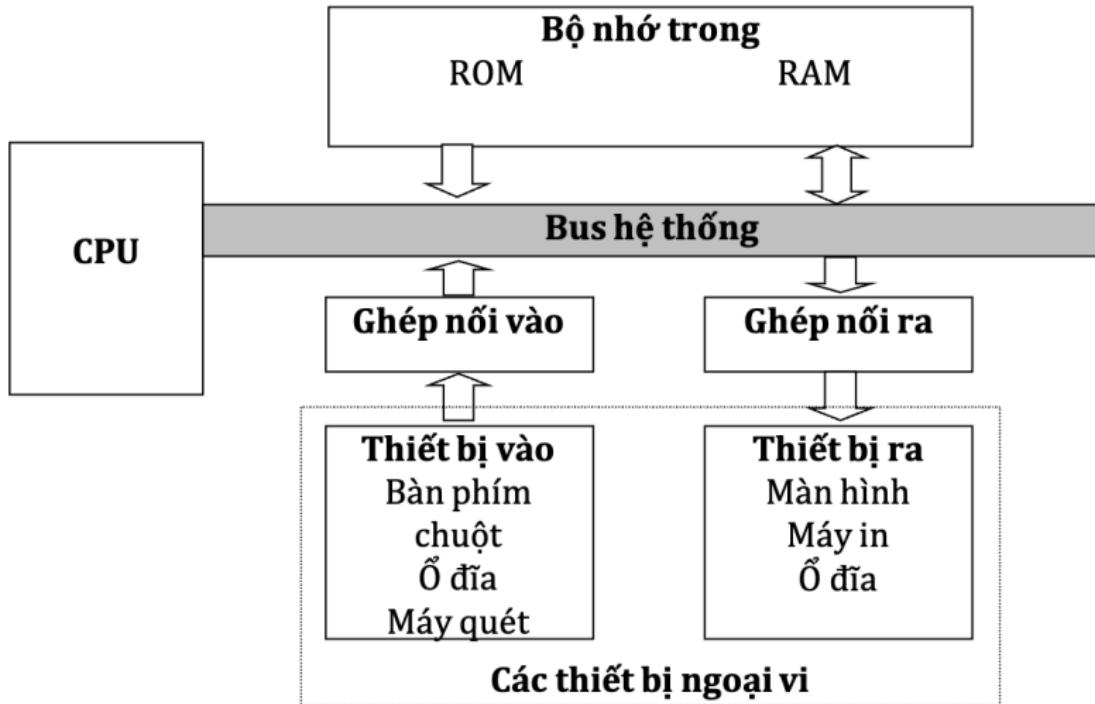


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

Data source: wikipedia (https://en.wikipedia.org/w/index.php?title=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 Böser

Microprocessor chronology



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

**Vi xử lý Intel
8086 (1978)**



**Vi xử lý Intel
Core 2 Duo
(2006)**

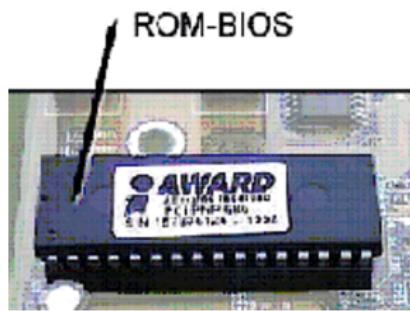


Bộ nhớ trong:

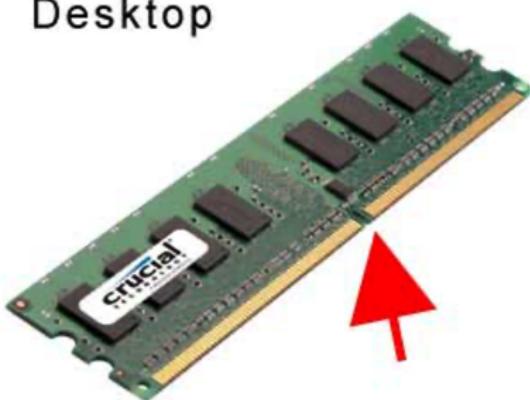
- ▶ Chức năng: Lưu trữ lệnh và dữ liệu để CPU xử lý
- ▶ Bao gồm:
 - ROM – Read Only Memory:
 - ▶ Lưu trữ lệnh và dữ liệu của hệ thống
 - ▶ Thông tin trong ROM vẫn tồn tại khi mất nguồn nuôi
 - RAM – Random Access Memory:
 - ▶ Lưu trữ lệnh và dữ liệu của hệ thống và người dùng
 - ▶ Thông tin trong RAM sẽ mất khi mất nguồn nuôi

Cấu trúc và chức năng của máy tính (cont.)

Bộ xử lý nhớ trong - Memory



Crucial 240 pin DIMM
Desktop



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

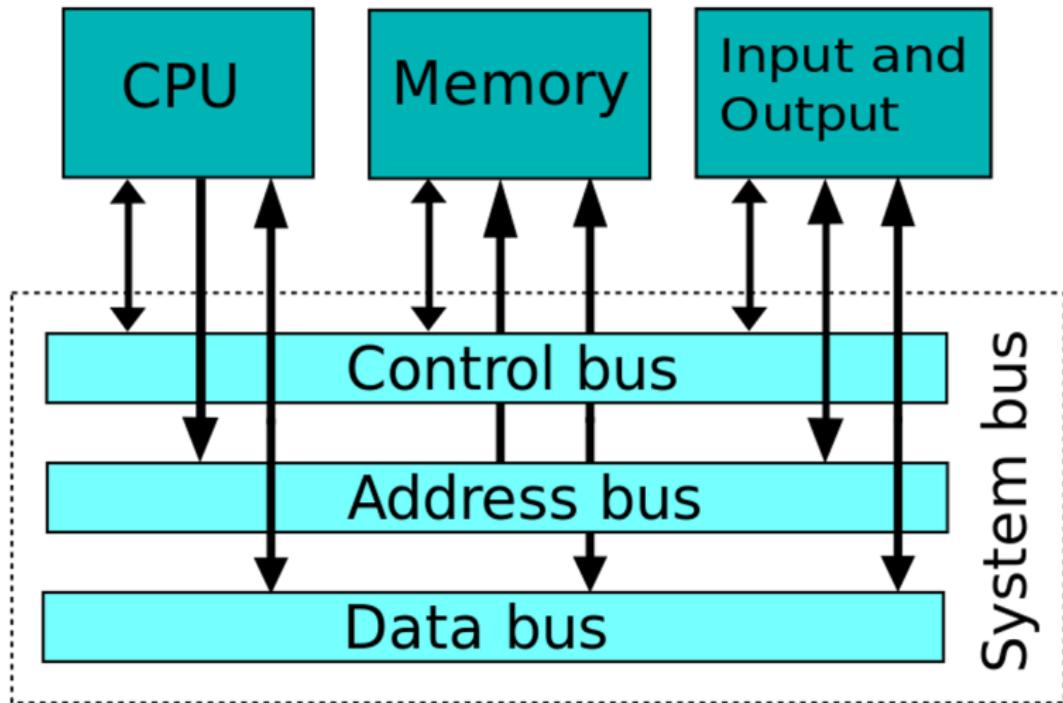
Bus hệ thống:

- ▶ 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)

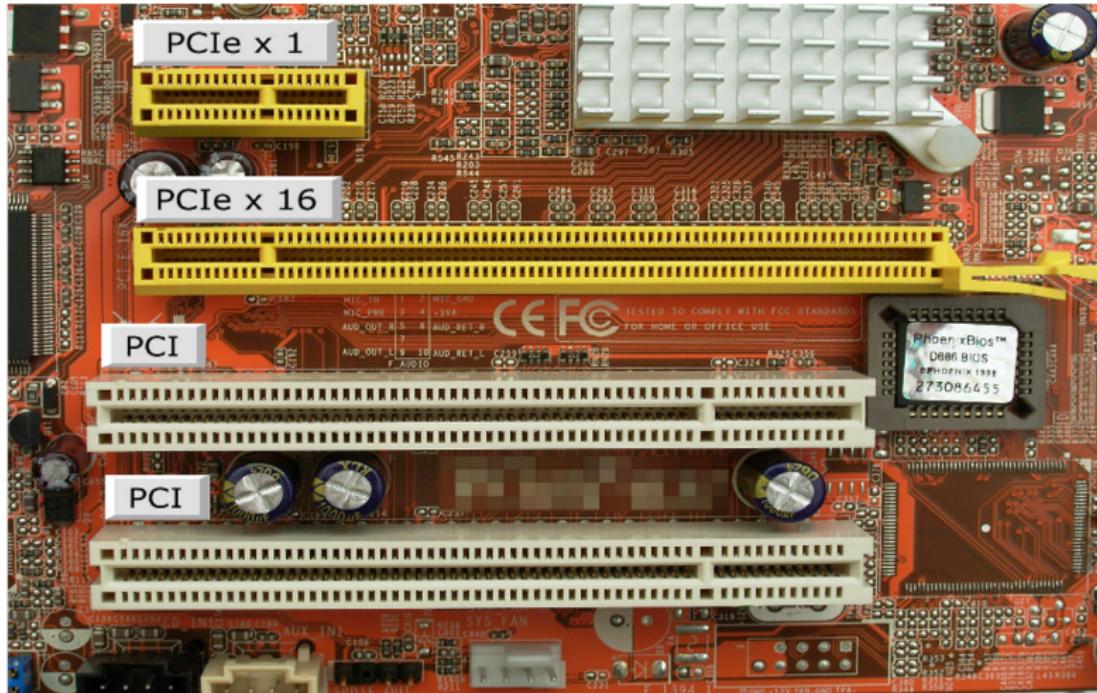


Cấu trúc và chức năng của máy tính (cont.)

Bus hệ thống



PCI bus:



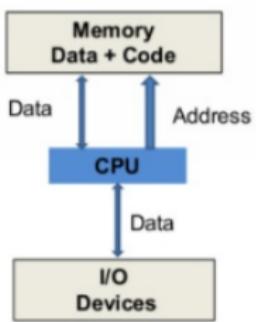
- ▶ Máy tính cá nhân (Personal Computers)
 - Desktop computers, Laptop computers
 - Máy tính đa dụng
- ▶ Máy chủ (Servers) – máy phục vụ
 - Đùng trong mạng để quản lý và cung cấp các dịch vụ
 - Hiệu năng và độ tin cậy cao
 - Hàng nghìn đến hàng triệu USD
- ▶ Siêu máy tính (Supercomputers)
 - Dùng cho 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

- ▶ 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

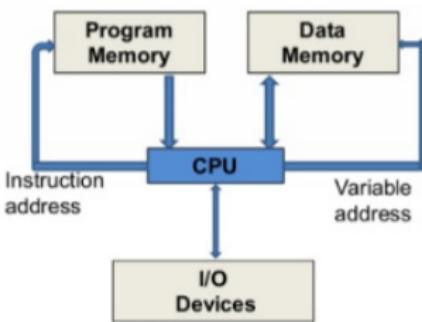
Theo sự phát triển, có nhiều loại kiến trúc ra đời (>20 kiến trúc máy tính khác nhau).

Nghiên cứu 2 kiến trúc chính sau:

- ▶ Kiến trúc Von-Neuman
- ▶ Kiến trúc Harvard



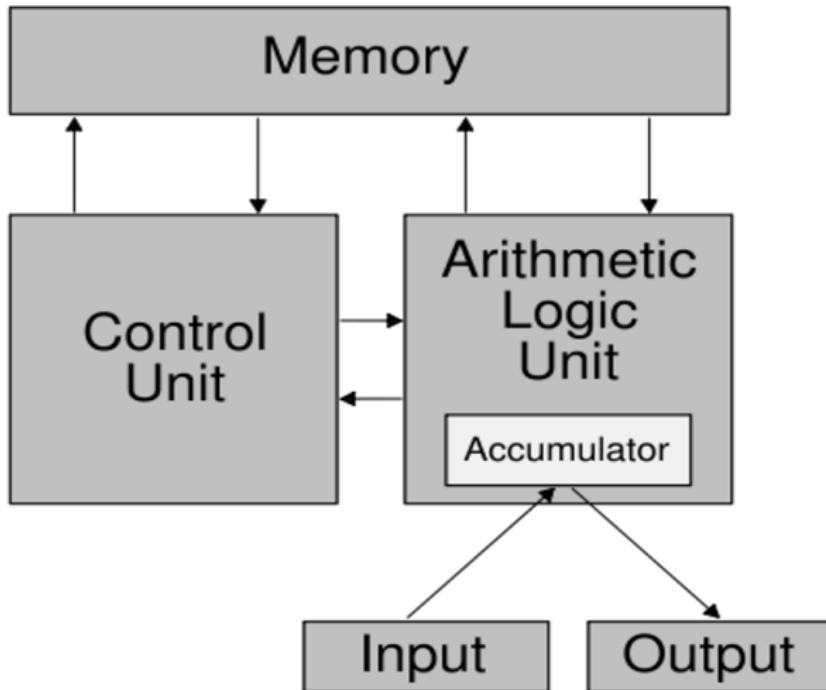
Von Neumann Machine



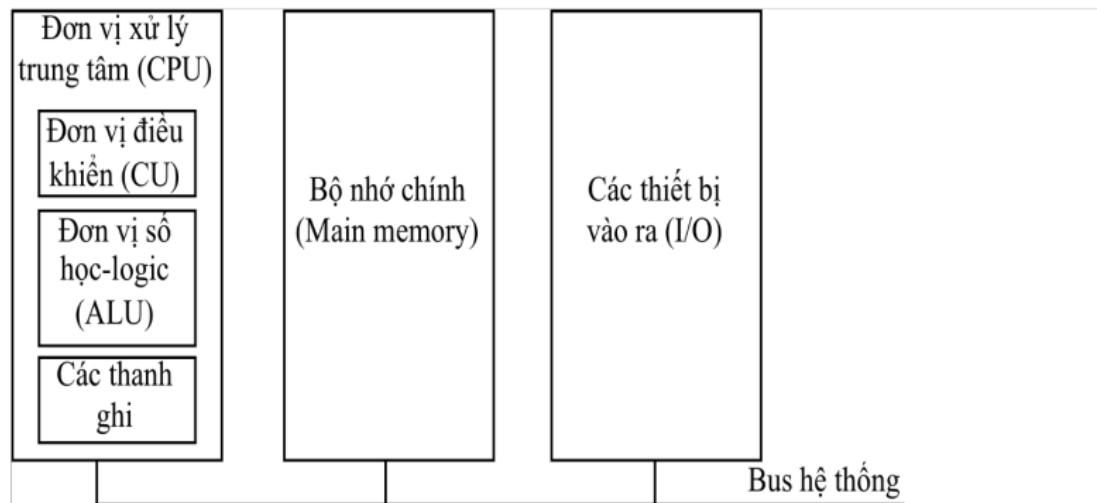
Harvard Machine

- ▶ 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*

Kiến trúc Von-Neumann cũ:



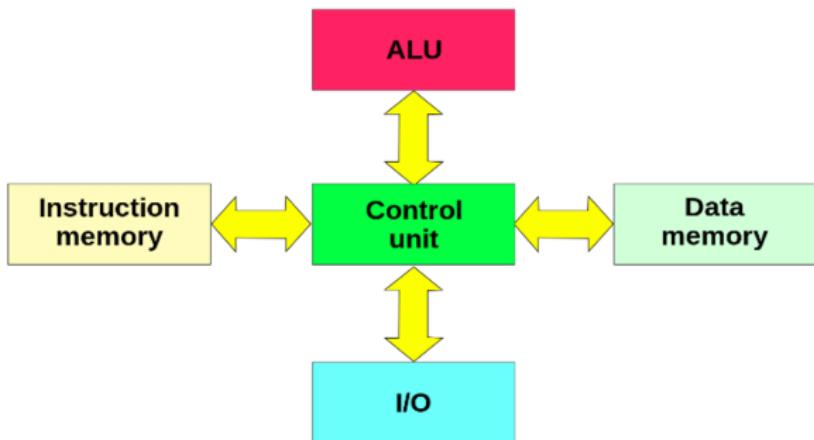
Kiến trúc Von-Neumann hiện đại:



- ▶ 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 (cố chai) không được 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

- ▶ 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



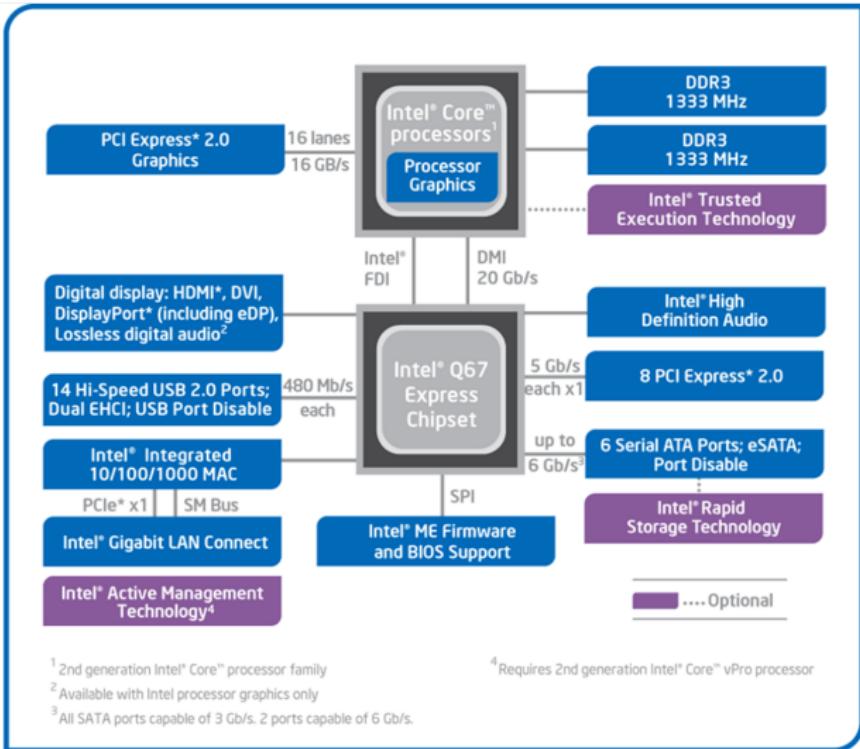
- ▶ Nhanh hơn vì băng thông bus rộng vì quá trình đọc lệnh không tranh chấp với quá trình truy xuất dữ liệu
- ▶ Hỗ trợ nhiều truy cập đọc/viết bộ nhớ cùng lúc nên giảm xung đột truy cập bộ nhớ
- ▶ Ngày nay kiến trúc Harvard cải tiến được ứng dụng cho các kiến trúc máy tính hiện đại: ARM, intel x86
-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

Câu hỏi và bài tập

- 1. Câu hỏi 1:** Vẽ sơ đồ và nêu các đặc điểm của kiến trúc máy tính von-Neumann cổ điển. Vẽ sơ đồ và phân biệt kiến trúc máy tính von-Neumann hiện đại với kiến trúc máy tính von-Neumann cổ điển?

- 2. Câu hỏi 2:** Vẽ sơ đồ và nêu các đặc điểm của kiến trúc máy tính Harvard. Kiến trúc máy tính Harvard có những ưu điểm gì so với kiến trúc máy tính von-Neumann. Các máy tính để bàn, xách tay hiện đại ngày nay sử dụng kiến trúc nào ?

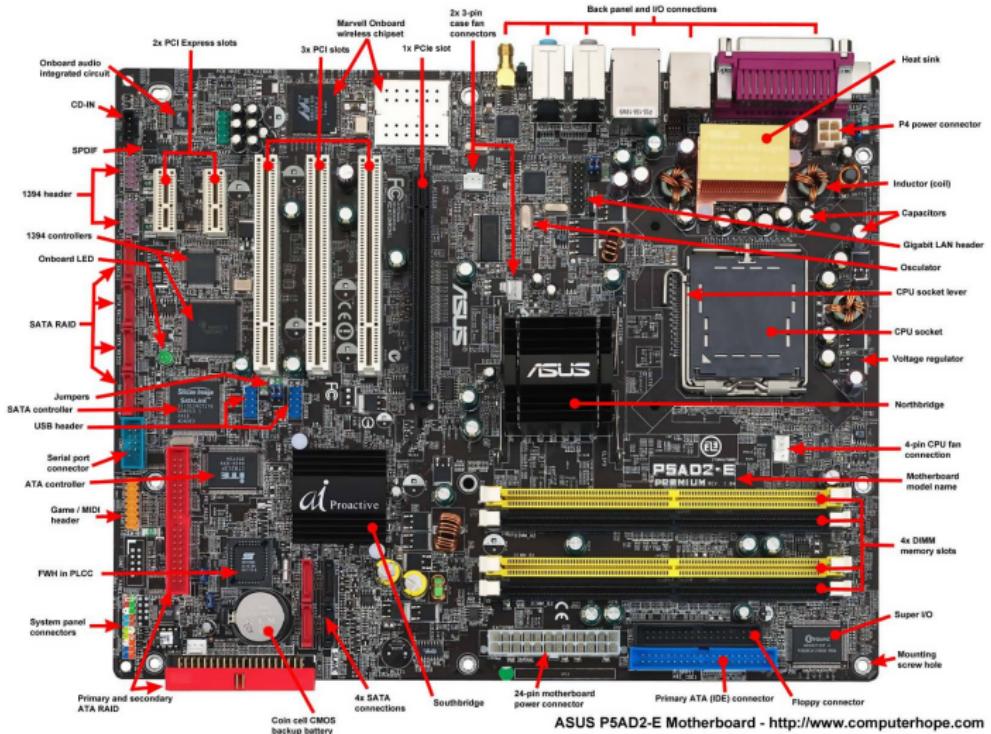
Một số tổ chức máy tính



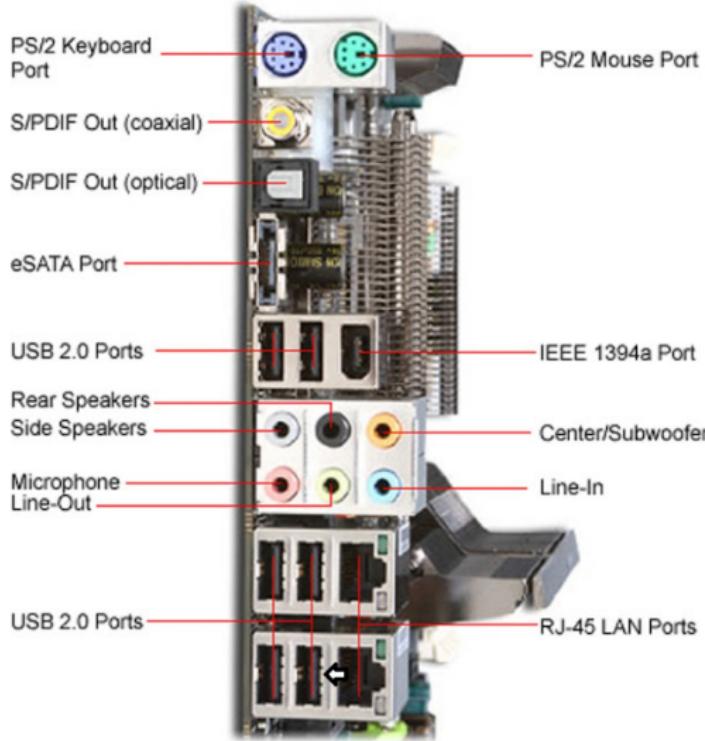
Intel® Q67 Express Chipset Platform Block Diagram

Một số tổ chức máy tính (cont.)

Bo mạch chủ:



Motherboard port:



► Hệ thập phân (Decimal System)

- Hệ đếm con người sử dụng
- Sử dụng 10 ký tự 0,1,2,..,9

► Hệ nhị phân (Binary System)

- Hệ đếm máy tính sử dụng
- Sử dụng 2 ký tự 0,1

► Hệ mười sáu (Hexadecimal System)

- dùng để viết gọn cho số nhị phân
- Sử dụng 16 ký tự 0, 1, 2,.., 9, A, B, C, D, E, F

Dùng n chữ số thập phân có thể biểu diễn được 10^n giá trị khác nhau:

- ▶ $00..000 = 0$
- ▶ $99..999 = 10^n - 1$

Một số thập phân được trình bày dưới dạng đa thức như sau:

$$\begin{aligned}a_n a_{n-1} \dots a_0, a_{-1} \dots a_{-m} &= \sum_{i=-m}^n a_i * 10^i \\&= a_n * 10^n + a_{n-1} * 10^{n-1} + \dots + a_0 * 10^0 + a_{-1} * 10^{-1} + \dots + a_{-m} * 10^{-m}\end{aligned}$$

Ví dụ:

$$\begin{aligned}123.456 &= 1 * 10^2 + 2 * 10^1 + 3 * 10^0 + 4 * 10^{-1} + 5 * 10^{-2} + 6 * 10^{-3} \\&= 100 + 20 + 3 + 0.4 + 0.05 + 0.006\end{aligned}$$

Dùng 2 chữ số nhị phân: 0 và 1 (**binary digit**) : *bit*

n bit có thể biểu diễn được 2^n giá trị khác nhau:

- ▶ $00..000 = 0$
- ▶ $11..111 = 2^n - 1$

Một số thập phân được trình bày dưới dạng đa thức như sau:

$$(a_n a_{n-1} \dots a_0, a_{-1} \dots a_{-m})_{(2)} = \sum_{i=-m}^n a_i * 2^i$$
$$= a_n * 2^n + a_{n-1} * 2^{n-1} + \dots + a_0 * 2^0 + a_{-1} * 2^{-1} + \dots + a_{-m} * 2^{-m}$$

Ví dụ:

$$1101001.1011_{(2)} = 1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} + 1 * 2^{-4}$$
$$= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625$$
$$= 105.6875_{(10)}$$

Chuyển đổi số thập phân sang nhị phân - Phương pháp 1:

- ▶ Phần nguyên: chia dần cho 2 rồi lấy phần dư
- ▶ Phần thập phân: nhân với 2 lấy phần nguyên

Ví dụ:

$$105,6875_{(10)}$$

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

Kết quả:

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

Một số các hệ đếm (cont.)

Chuyển đổi Thập phân sang Nhị phân



Chuyển đổi số thập phân sang nhị phân - Phương pháp 2:

- Phân tích thành tổng của các số 2^i nhanh hơn với số nhỏ

Ví dụ:

$$\begin{aligned}105,6875 &= 64+32+8+1+0.5+0.125+0.0625 \\&= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4}\end{aligned}$$

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	0.0625
0	1	1	0	1	0	0	1	1	0	1	1

Kết quả: $105,6875_{(10)} = 0\textcolor{red}{1}10\textcolor{yellow}{1}001,\textcolor{blue}{1}011_{(2)}$

Một số các hệ đếm

Hệ mười sáu (Hexa)



Hệ mười sáu - Hexa:

- ▶ Sử dụng 16 ký tự: 0,1,2,3,...,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 **1 chữ số Hexa**.

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

Ví dụ: Chuyển đổi số Nhị phân sang số Hexa:

1011 0011 → B3

0010 1101 1001 1010 → 2D9A16

2023A → 0010 0000 0010 0011 1010

Tổ chức dữ liệu

► Bit:

- Bit 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 16 giá trị từ $(0000)_2$ tới $(1111)_2$, hoặc 1 số hệ Hexa.

► Byte:

- Nhóm 8 bits hoặc 2 Nibbles
- Có thể lưu trữ 256 giá trị từ $(00000000)_2$ tới $(11111111)_2$, hoặc 2 số hệ Hexa 00_{16} – FF_{16} .



► Từ nhớ (word): là một nhóm bit biểu diễn dữ liệu thuộc loại nào đó

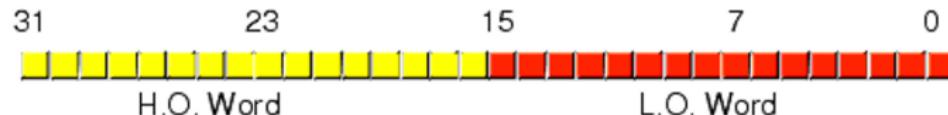
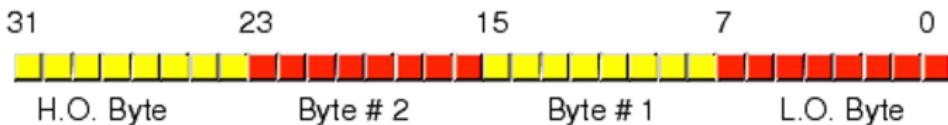
- Có kích thước trong khoảng 4 đến 64 bit.
- Thường là 8, 16, 24, 32 hoặc 64 bit phụ thuộc vào bộ VXL
- Ví dụ: Vi xử lý 16bit \rightarrow word = 16 bits = 2 bytes.



Tổ chức dữ liệu (cont.)

- ▶ Double word: là một nhóm bit gồm 2 từ nhớ (word)

- Ví dụ: Vì xử lý 16bit \rightarrow double word = 2 words = 32 bits = 8 bytes.



Số có dấu và không dấu

- ▶ Số có dấu trong hệ nhị phân, bit trái nhất được dùng để biểu diễn dấu của số (sign-magnitude):
 - Bit trái nhất là 0 ->số dương
 - Bit trái nhất là 1 ->số âm
- ▶ **Ví dụ:** sử dụng 4 bit để biểu diễn các số bằng bit dấu và trị tuyệt đối (sign-magnitude)
 - 0011, 0111, 0101 là các số dương +3, +7, +5
 - 1011, 1111, 1101 là các số âm +3, +7, +5
- ▶ Đối với các số không dấu, tất cả các bit đều lưu giá trị.
- ▶ Một phương pháp để biểu diễn số có dấu là sử dụng mã bù 2 (**2's complement**).

Số có dấu và không dấu (cont.)

► Phạm vi biểu diễn: n bits có thể biểu diễn: tối đa 2^n số:

- Số không dấu: từ 0 tới $2^n - 1$
 - ▶ 8 bits: từ 0 tới 255
 - ▶ 16 bits: từ 0 tới 65535
 - ▶ 32 bits: từ 0 tới 4,294,967,295
- Số không dấu - biểu diễn bằng dấu và giá trị tuyệt đối (sign - magnitude) biểu diễn được số $-2^{n-1} + 1$ tới $+2^{n-1} - 1$
 - ▶ 8 bits: -127 tới +127
 - ▶ 16 bits: -32767 tới +32767
 - ▶ 32 bits: 2,147,483,647 tới +2,147,483,647
- Số không dấu - biểu diễn bằng mã bù 2 (2's complement) biểu diễn được số -2^{n-1} tới $+2^{n-1} - 1$
 - ▶ 8 bits: -128 tới +127
 - ▶ 16 bits: -32768 tới +32767
 - ▶ 32 bits: 2,147,483,648 tới +2,147,483,647

Số có dấu và không dấu (cont.)

Bảng mã ASCII

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

Số có dấu và không dấu (cont.)

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	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20		100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	\$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	*	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l

Chương 1

- ▶ Khái niệm kiến trúc và tổ chức máy tính
- ▶ Lịch sử phát triển máy tính
- ▶ Cấu trúc và chức năng của máy tính
- ▶ Một số các hệ đếm

Tiếp theo Chương 2

- ▶ Sơ đồ khái tóm tắt của CPU
- ▶ Chu kỳ xử lý lệnh CPU
- ▶ Các thành phần chức năng của CPU
 - Các thanh ghi
 - Khối điều khiển CU
 - Khối số học và logic ALU
 - Bus trong CPU

Câu hỏi và bài tập