

Ngôn ngữ VẰM TÍNH và BỘ XỬ LÝ - Các thể hệ:
+ **0 (1642-1940)** - máy tính cơ học - Non-digital computers
+ **1 (1940 -1956)** - bóng đèn chân không Vacuum Tubes (xem như các bit), máy v tính đầu tiên IBM 700 (dòng thinkpad hiện nay), hệ thống ENIAC (Electronic Numerical Integrator and Computer), tốc độ 40.000 ops/s, (Ex: UNIVAC, EDVAC, EDSAC)
+ **2 (1956-1963)** - transistor, IBM 7094, tốc độ x, là bước đột phá , ra đời ngôn ngữ Fortran, ngôn ngữ Assembly (FORTRAN, COBOL).

+3**(1964 - 1971)** - Integrated Circuits(Ex: PDP8, IBM 360, ICL 2900) (mach tích hợp/ bản đãn, 3000 devices/chip, tốc độ x5.
+ **4 (1971 - nay)** mach tích hợp nhỏ nhất để siêu cao LSI/ vi xử lý, vi xử lý đầu tiên ra đời(intel 4004), GUI cũng đ phát triển trong gđ này(Ex: IBM, STAR 1000), hơn 100tr devices/chip, tốc độ x10.+
5 **(Trương lại - 2010 - nay)**: sự ra đời của AI, Parallel Processing, **Kiến trúc Von-Neumann**: đt liên lạc và chương trình chia trong bộ đồ ghi, bộ nhớ được phân chia chỉ cho các ngăn nhớ không phụ thuộc vào nội dung của chúng, các lệnh được thực hiện tuần tự theo 5 thành phần: CPU, RAM, ROM, Input, Output. **Quy luật Moore**: số lượng transistor (tích hợp trong một IC/chip hoặc trên mỗi đơn vị inch²) tăng gấp đôi mỗi 18 hoặc 24 tháng.
Thành phần cơ bản của máy tính (đề hàn): main hình, bo mạch chủ (mainboard), CPU, chân cắm dây nối HDD, RAM, chân cắm mô rộng PCI/PCI Express, nguồn điện, ổ quang CD/ DVD, ổ đĩa cứng, bàn phím, chuột.

Bộ xử lý vi xử lý các lệnh máy, gồm khối điều khiển (điều khiển xử lý ALU và data tren register), ALU, register, (internal bus - kết nối CU, ALU và register).

2 thành phần bên trong vi xử lý: Control Unit, Data Path
Cấu trúc chính (S): bộ xử lý (CPU), hệ thống nhớ (main memory/trong - ngoài), hệ thống kết nối (bus), I, O.

Wafer (đé chip): tấm silicon đé cấy các vật liệu > tạo vi mach, kích thước 1inch (25.4mm) - 7.9inch (200mm).

Chip: là mach tích hợp gắn trên wafer dùng đé xử lý các công việc, có thể chứa hàng chz triệu transistor, gồm 4/8/16/32/64 bit.
Chipset là tập hợp nhiều chip trên wafer, thông dụng là **CPU, GPU** (đơn vi xử lý đồ họa), **RAM** (bộ nhớ truy cập tức thời phục vụ cho CPU), **bản cầu bắc** (tích hợp trên mainboard hỗ trợ truyền thông tin cho CPU/RAM, nằm kế CPU), **bản cầu nam** (quản lý thiết bị ngoại vi như HDD, mouse, keyboard, năm cầu mainboard).
Chức năng cơ bản của máy tính: lưu trữ + xử lý + trao đổi dữ liệu + điều khiển. Thực hiện theo trình tự: nhận -> xử lý -> xuất thông tin. **Các hoạt động máy tính gồm**: thực hiện chương trình, ngắt, vào/ra.

Ngắt: khả năng tạm ngừng chương trình đé thực thi chương trình khác, xảy ra khi thiết bị phần cứng hay chương trình cần sự giúp đỡ cho CPU nó sẽ gửi đi tín hiệu hoặc lệnh để bộ vi xử lý.

Các loại ngăn trong máy tính: ngắt cứng do các tín hiệu INTR là ngắt chân được hoạt động không cần được NMI đòi hỏi CPU thực hiện ngay khi có yêu cầu (sự cố điện/ lỗi bộ nhớ), mức độ ưu tiên cao nhất; **ngắt mềm** do lệnh INT (ROM-BIOS); **ngắt ngoại** lệ do lệnh của CPU như chia 0, flag.

Các thành phần của hệ thống máy tính:

Đơn vi xử lý trung tâm (Central Processor Unit - CPU)
Bộ nhớ chính: RAM & ROM Hệ thống vào ra (Input-Output System)

Liên kết hệ thống (Buses): bus địa chỉ, bus dữ liệu hoặc bus cục bộ.

Các loại máy tính:

Super Computer: xử lý lượng dữ liệu lớn
Work Station:hệ thống máy người sử dụng
Microcomputer: cá nhân sử dụng

◻**Cấu tạo**: Processing devices: mother board, processor, RAM, ROM, switched mode power supply.

Thuật toán nhân không dấu: C (1bit) = 0 và A = 0. Khởi tạo day CAQ, Với Q là số nhân, M là số bị nhân. **Lập**: xét bit cuối của Q nếu 1 thì CA=A+M, sau đó dịch phải của CAQ. Res = AQ.
Thuật toán Booth - nhân số bù 2: A= 0, Q= 1 (1bit) = 0. Khởi tạo day AQQ1M. **Lập**: xét Qo=1 nếu = 01 thì A=A+M, nếu = 10 thì A=A-M, sau đó dịch phải day AQQ-1. Kết quả = AQ. **Thuật toán chia số kó dấu:** A= 0, Q là số bị chia, M là số chia. Khởi tạo day AQ. **Lập**: dịch trái AQ, A = A-M nếu A>0 thì Qo=1, ngược lại Qo=0 và A = A+M. **Lập**: C= Q, dur=A.

M = 7, Q = -3, n = 4

	A	Q	Q _i	M
Khởi đầu	0000	1101	0	0111
Bước 0: A=A-M	1001	1101	0	0111
shift	1100	1110	1	0111
Bước 1: A=A+M	0011	1110	1	0111
shift	0001	1111	0	0111
Bước 2: A=A-M	1010	1111	0	0111
shift	1101	0111	1	0111
Bước 3: shift	1110	1011	1	0111

Kết quả 11101011 = -21

Thuật toán chia số bù 2

thực hiện như phép chia không dấu, hệ thống kết nối (bus), I, O.

Chipset là tập hợp nhiều chip trên wafer, dùng để xử lý các công việc, có thể chứa hàng chz triệu transistor, gồm 4/8/16/32/64 bit.

Chipset là tập hợp nhiều chip trên wafer, thông dụng là **CPU, GPU** (đơn vi xử lý đồ họa), **RAM** (bộ nhớ truy cập tức thời phục vụ cho CPU), **bản cầu bắc** (tích hợp trên mainboard hỗ trợ truyền thông tin cho CPU/RAM, nằm kế CPU), **bản cầu nam** (quản lý thiết bị ngoại vi như HDD, mouse, keyboard, năm cầu mainboard).
Chức năng cơ bản của máy tính: lưu trữ + xử lý + trao đổi dữ liệu + điều khiển. Thực hiện theo trình tự: nhận -> xử lý -> xuất thông tin. **Các hoạt động máy tính gồm**: thực hiện chương trình, ngắt, vào/ra.

Ngắt: khả năng tạm ngừng chương trình đé thực thi chương trình khác, xảy ra khi thiết bị phần cứng hay chương trình cần sự giúp đỡ cho CPU nó sẽ gửi đi tín hiệu hoặc lệnh để bộ vi xử lý.

Các loại ngăn trong máy tính: ngắt cứng do các tín hiệu INTR là ngắt chân được hoạt động không cần được NMI đòi hỏi CPU thực hiện ngay khi có yêu cầu (sự cố điện/ lỗi bộ nhớ), mức độ ưu tiên cao nhất; **ngắt mềm** do lệnh INT (ROM-BIOS); **ngắt ngoại** lệ do lệnh của CPU như chia 0, flag.

Các thành phần của hệ thống máy tính:

Đơn vi xử lý trung tâm (Central Processor Unit - CPU)
Bộ nhớ chính: RAM & ROM Hệ thống vào ra (Input-Output System)

Liên kết hệ thống (Buses): bus địa chỉ, bus dữ liệu hoặc bus cục bộ.

Các loại máy tính:

Super Computer: xử lý lượng dữ liệu lớn
Work Station:hệ thống máy người sử dụng
Microcomputer: cá nhân sử dụng

◻**Cấu tạo**: Processing devices: mother board, processor, RAM, ROM, switched mode power supply.

Name	Abbr	Factor
kibi	Ki	2 ¹⁰ = 1,024
mebi	Mi	2 ²⁰ = 1,048,576
gibi	Gi	2 ³⁰ = 1,073,741,824
tebi	Ti	2 ⁴⁰ = 1,099,511,627,776
pebi	Pi	2 ⁵⁰ = 1,125,890,360,562,880
exbi	Ei	2 ⁶⁰ = 1,152,921,504,606,846,976
zobi	Zi	2 ⁷⁰ = 1,180,591,520,717,411,303,424
yobi	Yi	2 ⁸⁰ = 1,208,925,819,614,629,174,706,176

International System of Units (SI)

Name	Abbr	Factor	Bit size
kibi	K	2 ¹⁰ = 1,024	10 ³ = 1,000
mebi	M	2 ²⁰ = 1,048,576	10 ⁶ = 1,000,000
gibi	G	2 ³⁰ = 1,073,741,824	10 ⁹ = 1,000,000,000
tebi	T	2 ⁴⁰ = 1,099,511,627,776	10 ¹² = 1,000,000,000,000
pebi	P	2 ⁵⁰ = 1,125,890,360,562,880	10 ¹⁵ = 1,000,000,000,000,000
exbi	E	2 ⁶⁰ = 1,152,921,504,606,846,976	10 ¹⁸ = 1,000,000,000,000,000,000
zobi	Z	2 ⁷⁰ = 1,180,591,520,717,411,303,424	10 ²¹ = 1,000,000,000,000,000,000,000
yobi	Y	2 ⁸⁰ = 1,208,925,819,614,629,174,706,176	10 ²⁴ = 1,000,000,000,000,000,000,000,000

Chú ý: khi nói "kibibyte" chúng ta nghĩ là 1024 byte nhưng thực ra nó là 1000 bytes theo chuẩn SI, 1024 bytes là kibibyte (IEC)

Hiện nay chỉ có các nhà sản xuất đĩa cứng và viễn thông mới dùng chuẩn SI.

30 GB > 30 * 10⁶ = 28 * 2²⁸ bytes
1 MiBibits > 10⁶ bits

CHƯƠNG 3: BIỂU DIỄN SỐ THỰC

Giới hạn số chấm thập: n bit thì phần thập phân nhỏ nhất có thể biểu diễn là 2⁻ⁿ.

Theo chuẩn IEEE 754/85: có 3 dạng biểu diễn số thực + Single: độ dài 32bit với 8 chữ trong SEM là 1+8+23.

X = (-1)^S * 1.M. R*(E-127)
+ Double: độ dài 64bit với 8 chữ trong 1+11+52.

X = (-1)^S * 1.M. R*(E-1023)

+ Double extended: 80bit với 8 chữ trong 1+15+64.

X = (-1)^S * 1.M. R*(E-16383)

Số chấm động: 1 bit dấu (sign), 8 bit mũ (E) và 23 bit định trị (S) -> giá trị S = 2^{2E}.

Các số đặc biệt số 0 (mũ - Exponent = 0, trị - Significand = 0). Số không thể chuẩn hóa (denormalized) (mũ = 0, trị #0). Số vô cùng (infinity) (mũ = 111...1 (toàn bit 1), trị #0).

Số báo lỗi (NaN - Not a Number) (111...1 (toàn bit 1), trị #0).

Phạm vi biểu diễn:

+ Dạng chuẩn lớn nhất: 1,[23 số 1]*2¹²⁷.

x 0 | 1111 1110 | 1111 1111 = (2-2⁻²³)*2¹²⁷

(tương tự cho số âm nhỏ nhất - (2-2⁻²³)*2¹²⁷)

+ Dạng chuẩn nhỏ nhất: 1,[23 số 0]*2⁻¹²⁶.

x 0000 0001 0000 0000 = 2⁻¹²⁶

+ Số dạng không chuẩn lớn nhất: 0,[23 số 1]*2⁻¹²⁷. Tuy nhiên IEEE754 quy định là 0,[23 số 1]*2⁻¹²⁶ vì muốn tính gần hơn với số đương dạng chuẩn nhỏ nhất.

x 0000 0000 1111 1111

+ Số dạng không chuẩn nhỏ nhất: 1,[22 số 0]*2⁻¹²⁷. Tuy nhiên IEEE754 quy định là 0,[22 số 0]1*2⁻¹²⁶.

x 0000 0000 0000...001 = 2⁻¹⁴⁹.

Trong C: phạm vi kiểu float từ 10⁻³⁸ đến 10³⁸, kiểu double từ 10⁻³⁰⁰ đến 10³⁰⁰.

Precision: số bit được sử dụng để biểu diễn 1 giá trị. **Accuracy**: độ chính xác mà một kiểu biểu diễn có thể biểu diễn 1 giá trị.

Roundng: phần cứng hỗ trợ 2 bit nhớ cho phần định trị giúp làm tròn kết quả.

Chuẩn IEEE làm tròn số chấm động:

+ tròn lên: 1.01 10 -> 1.10, 1.01 10 -> 1.01

+ tròn xuống: ...>1.01, ...>1.10

+ tròn về 0: bỏ giá trị 2 bit nhớ.

+ tròn về giá trị gần nhất.

Mã Unicode: bộ mã 2byte, đa ngôn ngữ có hỗ trợ TV.

ASCII: bộ mã 1byte do ANSI thiết kế có chức các ký tự điều khiển truyền tin (mấy in mã nhĩn), có chứa các ký tự đé khùng. Biểu diễn số thực sau theo dạng số chấm động chính xác đơn (32 bit): X = -3050

♦ Bước 1: Đổi X sang hệ nhị phân

X = -305010 = -1011 1110 1010

♦ Bước 2: Chuẩn hóa theo dạng ±F * 2^E

X = -305010 = - 1011 1110 1010 = -1.0111101010 * 2¹¹

♦ Bước 3: Biểu diễn Floating Point

/Số âm: bit dấu Sign = 1

/Số mũ: E = 11 -> Phần mũ exponent với số thừa K=127 được biểu diễn:

→ Exponent = E + 127 = 11 + 127 = 13810 = 1000 1010 / Phần định trị = 0111 1101 0100 0000 0000 (Thêm 12 số 0 cho đn 23 bit)

→ Kết quả nhận được: 1 1000 1010 0111 1101 0100 0000 0000 000

Chương 4: Kiến trúc bộ lệnh

Các thành phần của bộ vi xử lý: 3

-**Control unit** - Dùng đé ra các tín hiệu điều khiển cho ALU, giải mã các câu lệnh.

-**ALU (Đơn vi toán học luận lý)** : Chịu trách nhiệm trong các nhiệm vụ thực hiện các câu lệnh, phép tính. Đọc các toán hạng từ Tập thanh ghi. Sau đó hoặc là lưu lại vào Tập thanh ghi, hoặc là ghi vào trong bộ nhớ.

-**Registers (Tập thanh ghi)**

Cách hoạt động của bộ vi xử lý

High-level language What is R? Assembly language What is R? Machine Language

temp = v[i];

v[i] = v[k+1];

v[k+1] = temp;

lw \$t0, 0(\$k2)

sw \$t1, 4(\$k2)

sw \$t1, 4(\$k2)

0000 1001 1100

0100 1111 0010

.....

***Instructions(lệnh/chi thị / mã máy)** Là 1 cái chuỗi bit chứa yêu cầu gửi đến CPU (ALU) thực hiện

Gồm 2 thành phần chính:

Opcode (Mã lệnh) : Cho ALU biết thao tác cần thực hiện

Operand (Toán hạng) : Cho biết ALU biết toán hạng các đối tượng bị tác động bởi thao tác các lệnh trong mã lệnh

***Có 2 trường phái thiết kế bộ lệnh**: Complete Instruction Set Computer(CISC): bộ lệnh gồm rất nhiều lệnh, từ đơn giản đến phức tạp Reduced Instruction Set Computer(RISC): bộ lệnh chỉ gồm các lệnh đơn giản.

CISC sẽ nhận phần trên Hardware hơn, Risc thường sử dụng cho các hệ thống những như MIPS

Đề chay những tập tin mã máy cần Linker& Loader.

KL: Code -> Compiler -> Hợp ngữ

Assembler -> Object file, Linker liên kết những cái Object file để thành file thực thi. exe; File. exe có đé được Loader đưa vào Bộ nhớ để cho CPU đé thực Instruction để thực hiện.

Phần biệt 3 khái niệm:

Ngôn ngữ lập trình (High-level language): Loại ngôn ngữ nhân tạo, được cấu thành bởi từ vựng và ngữ pháp. Giúp người lập trình diễn đạt hướng dẫn cho máy tính hoạt động. Có độ trừu tượng cao, con người đé hiểu nhưng máy tính khó hiểu (cần phiên dịch).

Ngôn ngữ máy (Machine language): Là dãy bit (0/1) chứa yêu cầu mà bộ vi xử lý (CPU) phải thực hiện. **Ngôn ngữ riêng của CPU**, khó hiểu với con người nhưng máy tính hiểu thực tiếp.

Hợp ngữ (Assembly language): Ngôn ngữ cấp thấp, rất gần với ngôn ngữ máy. Dùng ký hiệu mã (mnemonic) thay cho dãy bit, dùng tên (label, tên biến, tên chương trình) thay cho địa chỉ. Để lập trình hơn ngôn ngữ máy, dù đé con người hoặc là sử dụng tốt hơn. **Phụ thuộc vào kiến trúc bộ lệnh (ISA)**.

Compiler và Assembler:

Compiler: Trình biên dịch ngôn ngữ cấp cao sang hợp ngữ. **Phụ thuộc vào ngôn ngữ cấp cao được biên dịch và kiến trúc hệ thống phần cứng bên dưới** mà nó đang chạy.

Assembler: Trình biên dịch hợp ngữ sang ngôn ngữ máy. **Phụ thuộc vào kiến trúc bộ lệnh (ISA) và hệ điều hành (OS)**. Một kiến trúc ISA có thể có nhiều Assembler khác nhau.

Cấu tạo của 1 lệnh máy (Instruction): Là dãy bit chứa yêu cầu mà bộ xử lý trong CPU (ALU) phải thực hiện. Gồm 2 thành phần chính:

Mã lệnh (opcode): thao tác cần thực hiện.

Thông tin về toán hạng (operand): các đối tượng bị tác động bởi thao tác các lệnh trong mã lệnh.

Kiến trúc bộ lệnh (ISA):

Tập lệnh dành cho những bộ vi xử lý có kiến trúc tương tự nhau. **Trong cùng một dòng vi xử lý**, bộ lệnh gần giống nhau. Một số kiến trúc bộ lệnh thông dụng: 80x86 (gọi tắt x86) của Intel (IA-16, IA-32, IA-64), MIPS (dùng nhiều trong hệ thống nhúng), PowerPC của IBM.

Trường phái thiết kế bộ lệnh CISC và RISC:

Hai trường phái thiết kế bộ lệnh:

Complete Instruction Set Computer (CISC): Bộ lệnh gồm rất nhiều lệnh, từ đơn giản đến phức tạp. **Nhấn mạnh trên Hardware**.

Reduced Instruction Set Computer (RISC): Bộ lệnh chỉ gồm các lệnh đơn giản. Thường sử dụng cho các hệ thống những như MIPS

Bộ lệnh đại diện: x86 đại diện cho CISC, MIPS và PowerPC đại diện cho RISC.

Ưu nhược điểm (còn bẻn): CISC có lệnh phức tạp hơn, có thể thực hiện tác vụ trong ít hơn hơn nhưng việc giải mã và thực thi có thể chậm hơn và phức tạp hơn; RISC có lệnh đơn giản, dễ dàng thiết kế mạch xử lý pipeline tốc độ cao, nhưng cần nhiều lệnh hơn đé thực hiện cùng tác vụ.

Linker và Loader:

Linker: Chương trình dùng để liên kết các file object (đã biên dịch từ mã nguồn) và các thư viện (Program library) có sẵn lại với nhau để tạo thành tập tin thực thi (.exe, .bat, .sh,...).

Loader: Chương trình dùng để tính toán và tải tập tin thực thi (từ đĩa cứng) vào bộ nhớ chính (RAM) đé CPU có thể xử lý.

Quá trình tạo file thực thi (.exe):

Source file(s) (ngôn ngữ cấp cao) -> **Compiler** -> **Assembly language program** -> **Assembler** -> **Object file(s)** (mã máy).

Object file và **Program library** -> **Linker** -> **Executable file**.

Quá trình xử lý lệnh của CPU (Instruction Cycle): CPU xử lý lệnh qua một vòng lặp gồm 2 bước chính:

Nạp lệnh (Fetch cycle): Di chuyển lệnh từ bộ nhớ vào thanh ghi (IR) trong CPU. Các bước cơ bản: **MAR** ← **PC**, **MBR** ← **Memory[MAR]**, **IR** ← **MBR**, và cập nhật **PC** ← **PC** + **Chiều dài lệnh**.

Thực thi lệnh (Execute cycle): Giải mã lệnh (trong IR), và thực hiện tác vụ yêu cầu (ví dụ: tính toán hoặc ALU, truy cập bộ nhớ, điều khiển luồng chương trình). Gồm các bước như tính địa chỉ toán hạng, nạp toán hạng, thực hiện phép toán, ghi kết quả.

Một số thanh ghi cơ bản trong CPU:

PC (Program Counter): Lưu địa chỉ của lệnh kế tiếp trong bộ nhớ sẽ được nạp.

IR (Instruction Register): Lưu lệnh hiện tại đang được CPU giải mã và thực thi.

MAR (Memory Address Register): Lưu địa chỉ của ô nhớ hoặc thiết bị I/O mà CPU đang truy cập.

MBR (Memory Buffer Register): Lưu dữ liệu được đọc từ bộ nhớ hoặc vào bộ nhớ/thiết bị I/O.

Thanh ghi đa dụng (General Purpose Registers): Dùng để lưu trữ tạm thời dữ liệu (thông tin chung, kết quả trung gian) cho các phép tính của ALU. (Ví dụ: \$ registers cho biến lưu, \$t registers cho biến tạm trong MIPS; AX, BX, CX, DX... trong x86).

Thanh ghi có (Flag Register): Chứa các bit trạng thái về kết quả của phép tính hoặc hoạt động của CPU (ví dụ: cờ Zero, cờ Dấu, cờ Trán).

Thanh ghi ngăn xếp (Stack Pointer - SP): Lưu địa chỉ đỉnh của ngăn xếp (stack).

Thanh ghi địa chỉ nền (Base Pointer - BP): Lưu địa chỉ nền của khung ngăn xếp hiện tại.

Thanh ghi địa chỉ trả về (Return Address - RA): Lưu địa chỉ của lệnh kế tiếp sau khi một lời gọi thủ tục (hàm) kết thúc.

CHƯƠNG 5: BỘ LỆNH LEGV8

LEGV8 là tập hợp con của kiến trúc tập lệnh ARMv8, phục vụ giáo dục.

ARMv8 là kiến trúc tập lệnh 64-bit cho thiết bị hiện đại. LEGV8 đơn giản hóa ARMv8, giữ lại phần quan trọng để đé học. Mã nguồn mở.

* **Đặc điểm**:

- Dạng triển khai thực RISC, tập lệnh nhỏ gọn, đé hiểu.

- Kiến trúc 64-bit (hỗ trợ cả 32-bit nhưng tập trung 64-bit).

- Bộ lệnh gồm: Tính toán số học, Lệnh Logic, Lệnh di chuyển dữ liệu, Lệnh điều khiển rẽ nhánh.

* **Tập thanh ghi (Register File)**:

- Thanh ghi tổng quát:

+ 32 thanh ghi kích thước 64 bit (DoubleWord): X0-X31.

+ 32 thanh ghi có kích thước 32 bit (Word): W0-W31.

- Thanh ghi có (Flag Register - NZCV):

• Bộ flags gồm Negative (N), Zero (Z), Carry (C) và Overflow (V).

- Thanh ghi số thực (Floating Point Registers):

32 bit: D0-S31

64 bit: D0-D31

- Chức năng các thanh ghi tổng quát cụ thể:

X0-X7: Làm đối số / kết quả trả về của hàm.

X8: Chứa (ví trí) địa chỉ kết quả trả về.

X9 - X15: Tập thanh ghi tạm (Temporary registers).

