

KIẾN TRÚC MÁY TÍNH

CHƯƠNG 1

MÁY TÍNH – CÁC KHÁI NIỆM VÀ CÔNG NGHỆ

1. Giới thiệu
2. Bên dưới chương trình ứng dụng
3. Bên trong máy tính
4. Hiệu suất (Performance)

1. Giới thiệu

2. Bên dưới chương trình ứng dụng

3. Bên trong máy tính

4. Hiệu suất (Performance)

Giới thiệu

- ✓ Máy tính – cuộc cách mạng thứ ba của nền văn minh
(cùng với cuộc cách mạng về nông nghiệp và công nghiệp)
- ✓ Xu hướng nghiên cứu khoa học mới:
 - Các nhà khoa học tính toán, lý thuyết và thực nghiệm cùng hợp tác nhau trong việc khám phá ra những thành tựu mới trong thiên văn học, sinh học, hóa học, vật lý, v.v...
- ✓ Những ứng dụng được xem là “khoa học viễn tưởng” trước đây:
 - Máy tính trong ô tô
 - Điện thoại
 - Dự án di truyền học người
 - World Wide Web
 - Công cụ tìm kiếm

Giới thiệu

- ✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:
 - Máy tính để bàn (Desktop computers)
 - Máy chủ (Servers)
 - Máy tính nhúng (Embedded computers)

Giới thiệu

- ✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:
 - **Máy tính để bàn**
 - Sử dụng bởi cá nhân, thường tích hợp màn hình hiển thị (graphic display), chuột (mouse) và bàn phím (keyboard).
 - Hiệu năng tốt đối với người dùng đơn lẻ, mức chi phí thấp, và thường được dùng để thực thi các phần mềm của hãng thứ ba, hay còn gọi là shrink-wrap software.
 - Là máy tính phổ biến nhất, cũng được biết đến với tên gọi máy tính cá nhân (personal computer).
 - Ví dụ: Intel Core i7-2600 (3.4Ghz), Intel Core i7-4770K Processor (3.5Ghz)
 - Máy chủ
 - Máy tính nhúng

Giới thiệu

- ✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:
 - Máy tính để bàn
 - **Máy chủ**
 - Dùng để chạy các chương trình lớn có nhiều người dùng đồng thời và thường được truy cập qua hình thức mạng.
 - Máy chủ có thể chạy ứng dụng đơn có tính phức tạp cao (như ứng dụng kỹ thuật và khoa học), hoặc điều khiển nhiều công việc nhỏ (như khi xây dựng một máy chủ Web lớn)
 - Những ứng dụng này thường dựa trên các phần mềm phát triển từ một nguồn khác (như hệ thống cơ sở dữ liệu hoặc mô phỏng), và thường được hiệu chỉnh để phù hợp với một chức năng cụ thể.
 - Máy chủ được xây dựng theo cùng công nghệ như máy tính để bàn, nhưng cung cấp khả năng mở rộng lớn về mặt tính toán và số lượng các ngõ nhập xuất (hiệu năng của máy chủ được đo bằng nhiều cách, tùy thuộc vào ứng dụng được dùng).
 - Máy tính nhúng

Giới thiệu

✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:

- Máy tính để bàn

- **Máy chủ**

Có nhiều loại khác nhau về chi phí và công suất:

- **Low-end servers:** được sử dụng trong các ứng dụng lưu trữ, ứng dụng cho doanh nghiệp nhỏ, dịch vụ web, có thể không kèm màn hình và bàn phím, chi phí khoảng 1000\$.
- **Supercomputers:**
 - thường dùng cho các công việc tính toán kỹ thuật và khoa học phức tạp và cao cấp, ví dụ như dự báo thời tiết, khai phá dầu mỏ, tìm ra cấu trúc của protein v.v... với hiệu năng cao nhất.
 - bao gồm hàng trăm đến hàng ngàn bộ xử lý, cùng với bộ nhớ kích cỡ gigabytes đến terabytes và khả năng lưu trữ dữ liệu terabytes đến petabytes, chi phí hàng triệu đến hàng trăm triệu đôla.
- **Datacenter:** mặc dù không được gọi với tên supercomputers, các Internet datacenters được sử dụng bởi những công ty như eBay, Google cũng chứa hàng ngàn bộ xử lý, với bộ nhớ hàng terabytes, và khả năng lưu trữ hàng petabytes. Datacenter thường được xem như là các cụm máy tính lớn.

- Máy tính nhúng

Giới thiệu

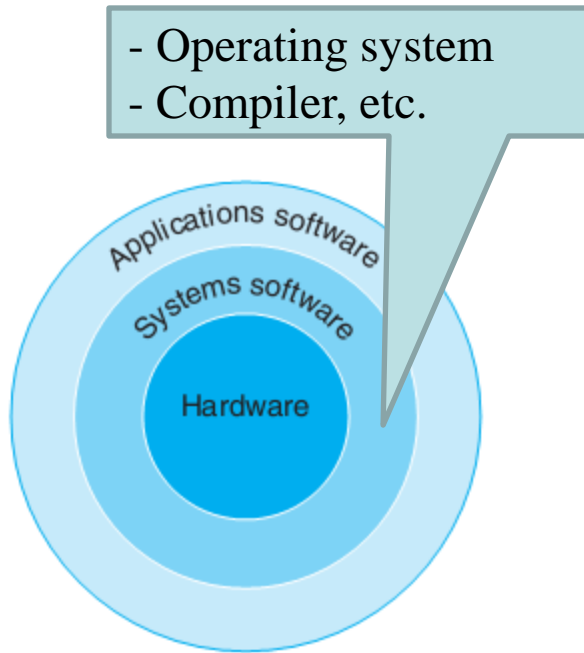
- ✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:
 - Máy tính để bàn
 - Máy chủ
 - **Máy tính nhúng**
 - Là máy tính bên trong một thiết bị nào đó, được dùng để chạy một ứng dụng hay một tập hợp các phần mềm cho trước; là lớp máy tính phổ biến nhất và trải rộng nhất về mặt ứng dụng và hiệu năng
 - **Máy tính nhúng:** bao gồm các vi xử lý được tìm thấy trong máy giặt, xe hơi, điện thoại, ti vi kỹ thuật số,...
 - **Hệ thống tính toán nhúng:** được thiết kế để chạy một ứng dụng hoặc một tập các ứng dụng có liên quan, thường được tích hợp với phần cứng và phân phối như một hệ thống duy nhất; theo đó, mặc dù các máy tính nhúng rất phổ biến, đa số người dùng không bao giờ thật sự nhận ra họ đang dùng một máy tính.
 - Yêu cầu quan trọng nhất của ứng dụng nhúng là đạt được hiệu năng hoạt động cần thiết tối thiểu với chi phí và năng lượng tiêu thụ thấp nhất
 - Trong nhiều năm vừa qua, tốc độ phát triển máy tính nhúng là nhanh hơn nhiều so với máy tính để bàn và máy chủ.

Giới thiệu

- ✓ Máy tính được sử dụng trong 3 lớp ứng dụng chính:
 - Máy tính để bàn
 - Máy chủ
 - Máy tính nhúng
- ➔ Nội dung sách chủ yếu trình bày về máy tính đa dụng (general-purpose computer), tuy nhiên đa số các khái niệm đều có thể áp dụng trực tiếp (hoặc với một số hiệu chỉnh nhỏ) cho các máy tính nhúng.

1. Giới thiệu
- 2. Bên dưới chương trình ứng dụng**
3. Bên trong máy tính
4. Hiệu suất (Performance)

Bên dưới chương trình ứng dụng



Hình 1 Các lớp phân cấp phần cứng và phần mềm của máy tính

Là các lớp phần cứng và phần mềm sau:

- **Ứng dụng** (Application)
- **Phần mềm hệ thống** (System software)
- **Phần cứng** (Hardware)

Phần mềm hệ thống (System Software): Phần mềm cung cấp các dịch vụ hữu ích và phổ biến.

Có nhiều phần mềm hệ thống, nhưng hai loại điển hình nhất cho hầu hết mọi hệ thống máy tính ngày nay là:

- Hệ điều hành
- Trình biên dịch

Hệ điều hành (Operating System): Điều hành chương trình dùng để quản lý các nguồn tài nguyên của máy tính, nhằm hỗ trợ các chương trình chạy trên máy tính đó.

Trình biên dịch (Compiler): Chương trình dịch các câu lệnh ở ngôn ngữ cấp cao sang ngôn ngữ assembly.

❖ Hệ điều hành

Hệ điều hành đóng vai trò giao tiếp giữa chương trình của người dùng và phần cứng, đồng thời cung cấp nhiều dịch vụ khác nhau và các chức năng quản lý. Một số chức năng quan trọng như:

- Điều khiển các hoạt động nhập xuất cơ bản
- Cấp phát bộ nhớ và vùng lưu trữ
- Quản lý chia sẻ tài nguyên máy tính khi có nhiều ứng dụng cùng chạy đồng thời

Một số hệ điều hành được sử dụng hiện nay: Windows, Linux, and MacOS.

❖ Trình biên dịch

Trình biên dịch thực hiện một chức năng quan trọng khác: dịch chương trình được viết bằng ngôn ngữ cấp cao (C, Java) thành tập các lệnh mà phần cứng máy tính có thể thực thi. Với sự phức tạp của các ngôn ngữ lập trình hiện đại và tính đơn giản của các lệnh thực thi bởi phần cứng, việc biên dịch từ chương trình ngôn ngữ cấp cao thành các lệnh phần cứng là khá phức tạp.

Bên dưới chương trình ứng dụng

❖ Từ ngôn ngữ cấp cao đến ngôn ngữ phần cứng

Bảng chữ cái cho máy tính: 0 và 1

Để giao tiếp với một máy điện tử, ta cần gửi đi các tín hiệu điện. Các tín hiệu dễ dàng nhất cho máy hiểu là tín hiệu **on (0)** và **off (1)** (mở và tắt).

- Bảng chữ cái tiếng Anh có 26 kí tự
- Bảng chữ cái cho máy tính có 2 kí tự → số nhị phân (*binary number*); mỗi kí tự là một kí số nhị phân (*binary digit*) hay còn gọi là *bit*

Ngôn ngữ máy tính

Lệnh (Instruction): Một yêu cầu được đưa ra mà phần cứng máy tính có thể hiểu và đáp ứng:

Ví dụ: 1000110010100000 – yêu cầu máy tính cộng hai số

Cách thức nhà lập trình giao tiếp với máy tính

Những nhà lập trình đầu tiên giao tiếp với máy tính thông qua các số nhị phân, một công việc khá buồn tẻ, và họ nhanh chóng tìm ra những cách viết mới gần gũi hơn với cách thức suy nghĩ của con người.

Bên dưới chương trình ứng dụng

❖ Từ ngôn ngữ cấp cao đến ngôn ngữ phần cứng

Cách thức nhà lập trình giao tiếp với máy tính

Ngôn ngữ Assembly (Hợp ngữ): ngôn ngữ mô tả lệnh của máy tính thông qua kí hiệu biểu diễn (symbol)

Assembler: chương trình dịch lệnh hợp ngữ sang lệnh nhị phân.

Ngôn ngữ lập trình cấp cao: Các ngôn ngữ có tính linh động (portable) như C, Fortran, Java; bao gồm các từ và kí hiệu số học, có thể được dịch sang ngôn ngữ Assembly bởi một trình biên dịch

Chú ý: Việc dịch từ ngôn ngữ cấp cao sang ngôn ngữ máy nhị phân gồm 2 bước (Hình 2), tuy nhiên một số trình biên dịch cắt giảm bước trung gian và dịch trực tiếp sang ngôn ngữ nhị phân.

High-level
language
program
(in C)

```
swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add $2, $4, $2
    lw $15, 0($2)
    lw $16, 4($2)
    sw $16, 0($2)
    sw $15, 4($2)
    jr $31
```

Assembler

Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000001000
00000011111000000000000000001000
```

Hình 2 Một chương trình C được dịch sang ngôn ngữ Assembly và sau đó là ngôn ngữ máy nhị phân

1. Giới thiệu
2. Bên dưới chương trình ứng dụng
- 3. Bên trong máy tính**
4. Hiệu suất (Performance)

Bên trong máy tính

Phần cứng của một máy tính bất kỳ thực hiện những chức năng cơ bản sau:

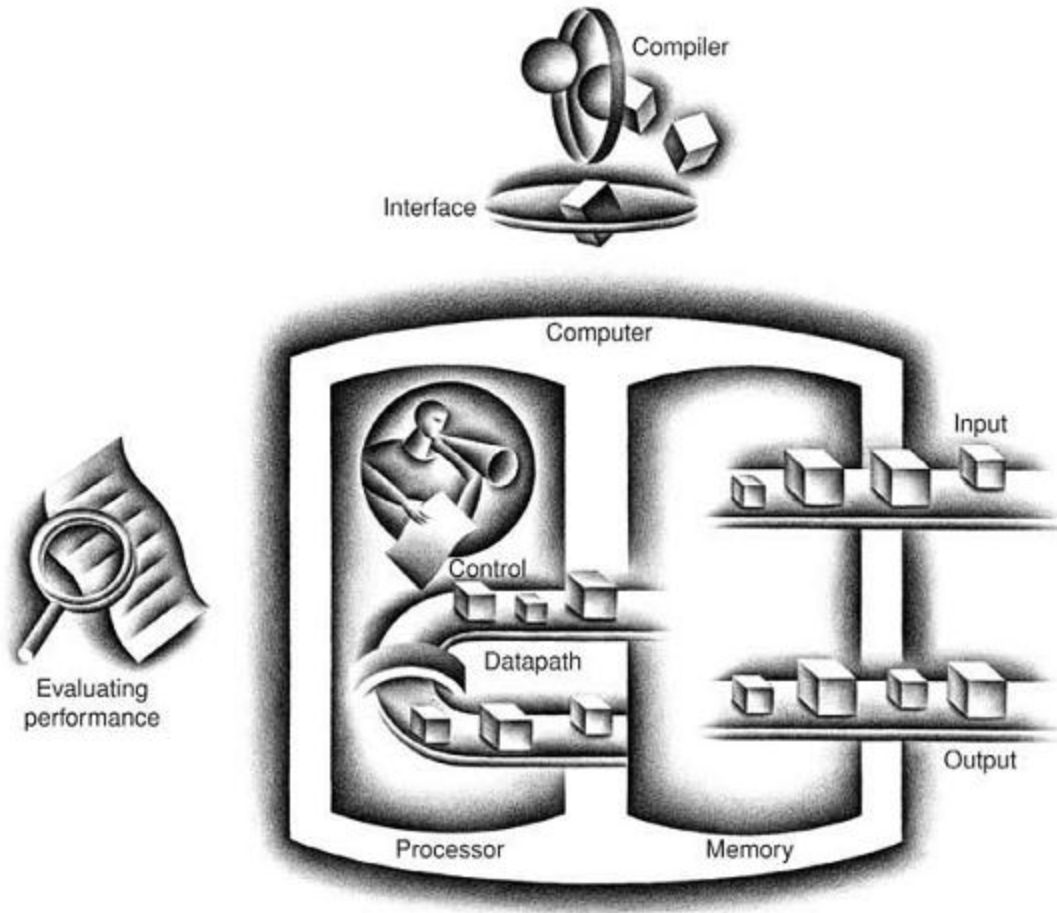
- ✓ **Nhập dữ liệu**
- ✓ **Xuất dữ liệu**
- ✓ **Xử lý dữ liệu**
- ✓ **Lưu trữ dữ liệu**

Năm thành phần căn bản của máy tính bao gồm:

- ✓ **Ngõ nhập**
- ✓ **Ngõ xuất**
- ✓ **Bộ nhớ**
- ✓ **Đường dữ liệu
(Data path)**
- ✓ **Khối điều khiển
(Control)**

(Data path và Control thường được kết hợp lại với tên gọi Processor)

Bên trong máy tính



Bộ xử lý (The processor): nhận lệnh và dữ liệu từ bộ nhớ. Ngõ nhập viết dữ liệu vào bộ nhớ, và ngõ xuất đọc dữ liệu từ bộ nhớ.

Khối điều khiển (Control) gửi các tín hiệu điều khiển hoạt động của đường dữ liệu, bộ nhớ, ngõ nhập và ngõ xuất.

Hình 3. Tổ chức của một máy tính, bao gồm 5 thành phần căn bản.

Bên trong máy tính



Hình 4. Máy tính để bàn
(Desktop computer)

- ✓ Màn hình (Screen) là thiết bị xuất chuẩn
- ✓ Bàn phím (Keyboard) và chuột (Mouse): thiết bị nhập chuẩn
- ✓ Thùng máy chứa bộ xử lý và các thiết bị I/O khác

(Một số thiết bị vừa là ngõ nhập vừa là ngõ xuất của máy tính như mạng và ổ đĩa)

- Chuột cơ điện (Electromechanical mouse, original mouse)
- Chuột quang (Optical mouse)

LCD – Liquid Crystal Displays
CRT - Cathode Ray Tube

Bên trong máy tính

❖ Ảnh màu

- Mỗi bức ảnh là một ma trận các phần tử ảnh, hay **pixels**, được biểu diễn thành một ma trận bit, gọi là *bit map*.

Pixel: Phần tử ảnh nhỏ nhất. Màn hình bao gồm hàng trăm ngàn triệu pixel được tổ chức thành một ma trận.

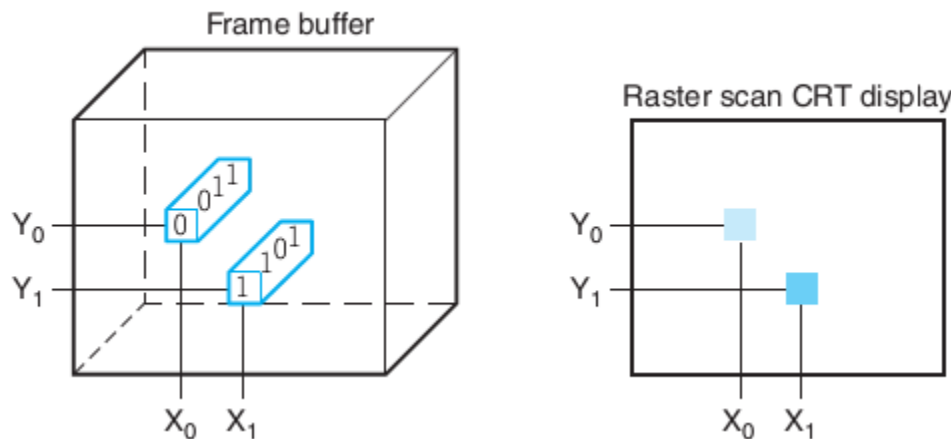
- Tùy vào kích cỡ màn hình và độ phân giải, ma trận hiển thị có thể có độ lớn từ 640 x 480 đến 2560 x 1600 pixels trong năm 2008
- Đối với ảnh màu, dùng 8 bit cho mỗi màu sắc (red, blue, green), tương đương 24 bits cho một pixel, cho phép hàng triệu màu sắc khác nhau được hiển thị.

Bên trong máy tính

❖ Ảnh màu

Phần cứng máy tính hỗ trợ đồ họa có thành phần chủ yếu là raster refresh buffer, hay còn gọi là frame buffer, để lưu nội dung bitmap.

Hình ảnh hiển thị lên màn hình được lưu trong frame buffer, và thông tin bit của mỗi pixel sẽ được đọc để hiển thị lên màn hình mỗi lần refresh.



Hình 5. Thiết kế đơn giản của Frame buffer với chỉ 4 bits cho một pixel.

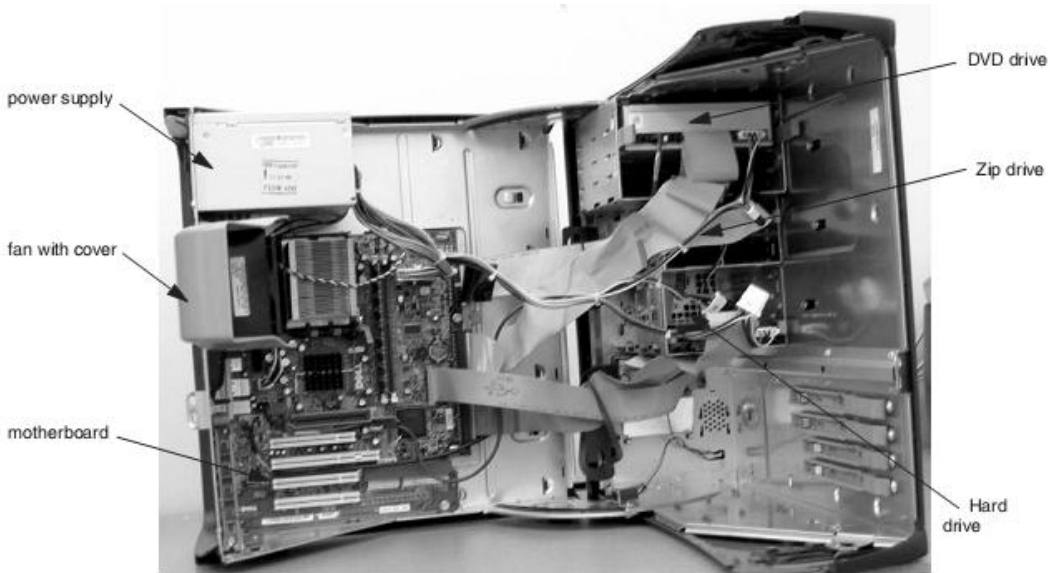
Mỗi tọa độ trong frame buffer bên trái xác định bóng của tọa độ tương ứng trong mỗi lần quét raster màn hình CRT bên phải. Pixel (X_0, Y_0) chứa thông tin bit 0011, tương ứng với màu xám nhạt trên màn hình, khác với thông tin bit 1101 của pixel (X_1, Y_1).

Ví dụ

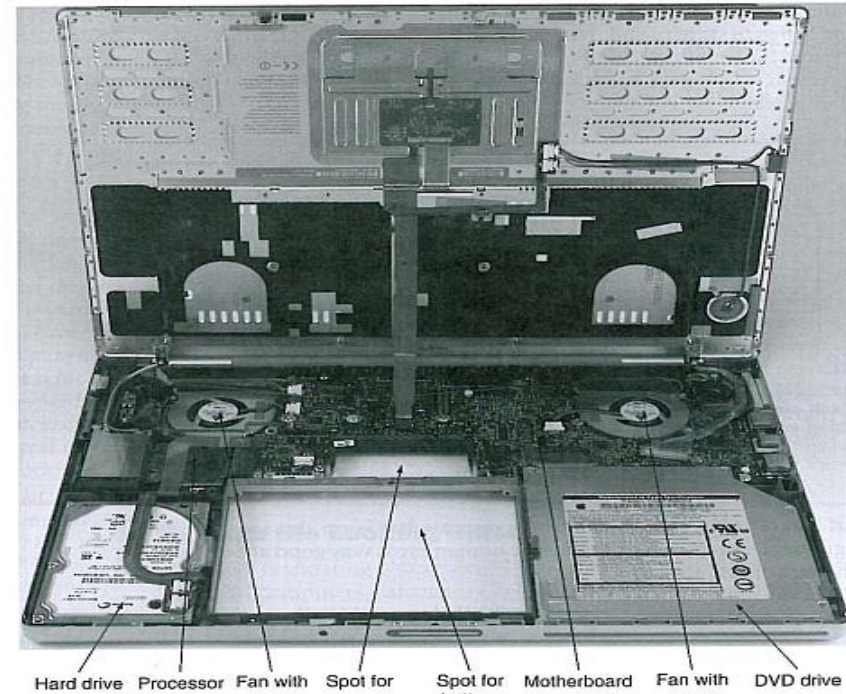
	Configuration	Resolution	Main Memory	Ethernet Network
a.	1	640 × 480	2 Gbytes	100 Mbit
	2	1280 × 1024	4 Gbytes	1 Gbit
b.	1	1024 × 768	2 Gbytes	100 Mbit
	2	2560 × 1600	4 Gbytes	1Gbit

- ☐ For a color display using 8 bits for each of the primary colors (red, green, blue) per pixel, what should be the minimum size in bytes of the frame buffer to store a frame?
- ☐ How many frames could it store, assuming the memory contains no other information

Bên trong máy tính



Hình 6. Bên trong máy tính để bàn



Hình 7. Bên trong máy tính xách tay

Bên trong máy tính

❖ Bên trong thùng máy

- **Board mạch chủ (Motherboard):** Là một bảng mạch bằng plastic, chứa các khối mạch tích hợp hay chips, gồm có bộ xử lý, cache, bộ nhớ, và kết nối cho các thiết bị I/O.
- **Mạch tích hợp:** Cũng gọi là chip, chứa đựng hàng chục đến hàng triệu transistors
- **Bộ nhớ:** là vùng lưu trữ chứa đựng chương trình đang chạy và chứa dữ liệu mà chương trình chạy cần dùng
 - ✓ DRAM (Dynamic random access memory): bộ nhớ cho phép truy cập ngẫu nhiên đến vị trí bất kỳ trong bộ nhớ
 - ✓ RAM (Random access memory): khác với các bộ nhớ truy cập tuần tự, như đĩa từ (magnetic tapes), thời gian truy cập vào bất kì vị trí nào trong bộ nhớ RAM cơ bản là như nhau.
 - ✓ DIMM (dual inline memory module): Một board nhỏ chứa chip DRAM trên cả hai mặt của board. SIMMs có DRAMs chỉ trên một mặt.

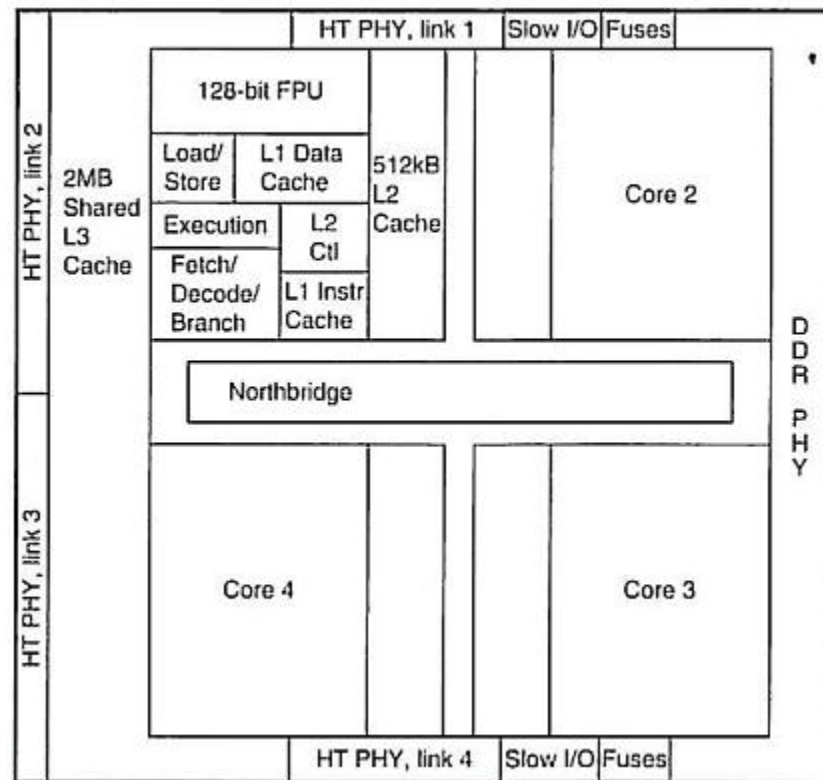
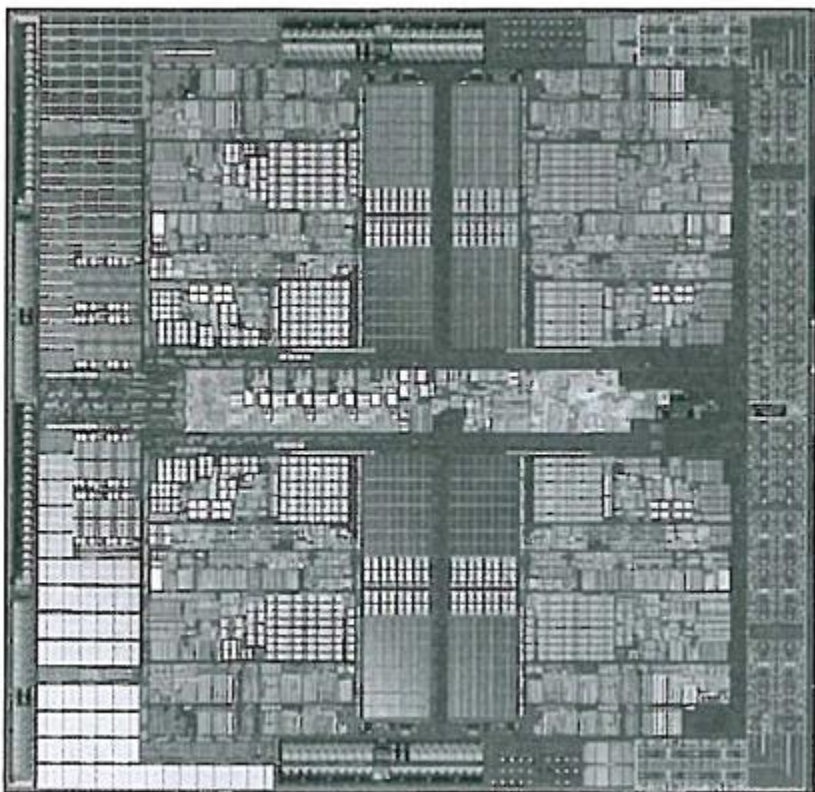
Bên trong máy tính

❖ Bên trong thùng máy

- **Đơn vị xử lý trung tâm (Central processor unit - CPU):** cũng gọi là bộ xử lý (Processor), bộ phận hoạt động tích cực của máy tính, chứa dữ liệu và khối điều khiển, thực hiện việc cộng số, kiểm tra số, kích hoạt các thiết bị I/O, v.v...
- **Datapath:** Thành phần của bộ xử lý, thực hiện các tính toán toán học
- **Control:** Thành phần của bộ xử lý, điều khiển đường dữ liệu, bộ nhớ, và các thiết bị I/O tùy theo lệnh nào đang thực thi của chương trình.
 - ➔ Đường dữ liệu thực hiện các tính toán toán học, và khối điều khiển sẽ hướng dẫn đường dữ liệu, bộ nhớ, và các thiết bị I/O những việc cần làm dựa trên yêu cầu của lệnh chương trình.
 - ➔ Datapath and control, giống như cơ bắp và bộ não của bộ xử lý.

Bên trong máy tính

❖ Chi tiết một bộ vi xử lý



Hình 8. Bên trong bộ vi xử lý AMD Barcelona. Hình bên trái là ảnh vi mô của chip xử lý AMD Barcelona, hình bên phải thể hiện các khối chính trong bộ xử lý. Chip này có 4 nhân xử lý, hay còn gọi là 4 “core”.

Bên trong máy tính

❖ Chi tiết một bộ vi xử lý

Cache:

- ✓ Bên trong bộ xử lý còn có một dạng bộ nhớ, gọi là bộ nhớ đệm (cache memory)
- ✓ Bộ nhớ Cache là một bộ nhớ nhỏ, nhanh, hoạt động như một bộ đệm cho bộ nhớ DRAM.
- ✓ Cache được xây dựng trên một công nghệ thiết kế bộ nhớ khác biệt, dựa trên static random access memory (SRAM). SRAM có tốc độ truy cập nhanh hơn và ít dày đặc hơn, do đó đắt hơn DRAM.

Bên trong máy tính

❖ Nơi lưu dữ liệu an toàn

- **Bộ nhớ khả biến (Volatile memory):** chỉ lưu dữ liệu khi có nguồn điện (vd: DRAM)
- **Bộ nhớ bất biến (Nonvolatile memory):** có thể lưu dữ liệu ngay cả khi không được cấp nguồn điện, dùng để lưu trữ chương trình giữa các lần chạy. Đĩa từ (Magnetic disk) là một dạng bộ nhớ bất biến.
 - **Bộ nhớ chính (Main/Primary memory):** là bộ nhớ khả biến, dùng để lưu chương trình đang chạy, điển hình là DRAM trong các máy tính ngày nay.
 - **Bộ nhớ thứ cấp (Secondary memory):** là bộ nhớ bất biến, dùng để lưu chương trình và dữ liệu giữa các lần chạy, điển hình là đĩa từ trong các máy tính ngày nay
 - ✓ **Đĩa từ (Magnetic disk, cũng gọi là hard disk):** bộ nhớ thứ cấp bất biến, bao gồm các đĩa quay được phủ bởi vật liệu ghi bằng từ.
 - Gigabyte: theo truyền thống, tương đương 1.073.741.824 (2^{30}) bytes, tuy nhiên một số hệ thống lưu trữ và giao tiếp hiện nay định nghĩa bằng 1.000.000.000 (10^9) bytes. Tùy thuộc ngữ cảnh, megabyte hoặc là 10^9 hoặc là 2^{30} bytes.
 - Đa số ổ đĩa cứng (hard drive) nằm bên trong máy tính, ngoài ra cũng có các dạng được kết nối vào máy tính như USB (universal serial bus).
 - ✓ **Đĩa quang Optical disks:** CDs (Compact disks) và DVDs (Digital video disks)

Bên trong máy tính

❖ Nơi lưu dữ liệu an toàn

Bộ nhớ Flash:

- Bộ nhớ bán dẫn không thay đổi, dùng để thay thế cho bộ nhớ đĩa (disk) trong các thiết bị di động như điện thoại, máy chơi nhạc, và hướng đến cho laptop.
- Rẻ hơn và chậm hơn DRAM nhưng mất và nhanh hơn đĩa từ

Ví dụ

For problems below, use the information about access time for every type of memory in the following table.

	Cache	DRAM	Flash Memory	Magnetic Disk
a.	5 ns	50 ns	5 μ s	5 ms
b.	7 ns	70 ns	15 μ s	20 ms

- ☐ Find how long it takes to read a file from a DRAM if it takes 2 microseconds from the cache memory
- ☐ Find how long it takes to read a file from a disk if it takes 2 microseconds from the cache memory.
- ☐ Find how long it takes to read a file from a flash memory if it takes 2 microseconds from the cache memory

Bên trong máy tính

❖ Giao tiếp với các máy tính khác

Mạng máy tính: kết nối tất cả máy tính, cho phép người dùng máy tính mở rộng năng lực tính toán thông qua giao tiếp giữa các máy tính. Mạng máy tính ngày càng trở nên phổ biến và là xương sống cho các hệ thống máy tính hiện nay.

Máy tính được kết nối mạng có nhiều thuận lợi:

- **Giao tiếp:** Thông tin được trao đổi giữa các máy tính với tốc độ cao.
- **Chia sẻ tài nguyên:** Khác với máy tính riêng lẻ xài riêng các thiết bị I/O, máy tính bên trong một mạng kết nối có thể chia sẻ các thiết bị này cho nhau .
- **Truy cập bất cục bộ:** Bằng cách kết nối máy tính từ xa, người dùng không cần ở gần máy tính mà họ đang sử dụng.

Bên trong máy tính

❖ Giao tiếp với các máy tính khác

Mạng máy tính khác nhau về độ dài và hiệu năng, theo đó chi phí sẽ tăng tương ứng với tốc độ truyền tải và khoảng cách mà thông tin có thể được truyền tới.

- **Ethernet** (hình thức mạng phổ biến nhất): có thể dài 1 km và tốc độ truyền dữ liệu có thể lên tới 10 gigabits trên giây
 - ➔ hữu ích khi kết nối các máy tính trong cùng một tầng của tòa nhà, là một ví dụ của dạng mạng cục bộ (**local area network**.)
 - Local area network (LAN)**: mạng được thiết kế để truyền dữ liệu trong một vùng địa lý giới hạn, điển hình là mạng trong một tòa nhà.
- **Wide area networks** (mạng xuyên lục địa, là xương sống của mạng Internet, hỗ trợ World Wide Web): có thể dài tới hàng trăm km và tốc độ truyền lên tới hàng gigabits trên giây.
 - ➔ thường được xây dựng trên công nghệ sợi quang (optical fibers) và được cung cấp bởi các công ty viễn thông.
- **Wireless technology** (mạng không dây, được phát triển rộng rãi, đa số laptop sử dụng công nghệ mạng này): mạng không dây phổ biến hiện nay theo chuẩn IEEE 802.11, cho phép tốc độ truyền dữ liệu trong khoảng 1 – 100 triệu bit trên giây.

Ví dụ

	Configuration	Resolution	Main Memory	Ethernet Network
a.	1	640 × 480	2 Gbytes	100 Mbit
	2	1280 × 1024	4 Gbytes	1 Gbit
b.	1	1024 × 768	2 Gbytes	100 Mbit
	2	2560 × 1600	4 Gbytes	1Gbit

❑ If a 256 Kbytes file is sent through the Ethernet connection, how long it would take?

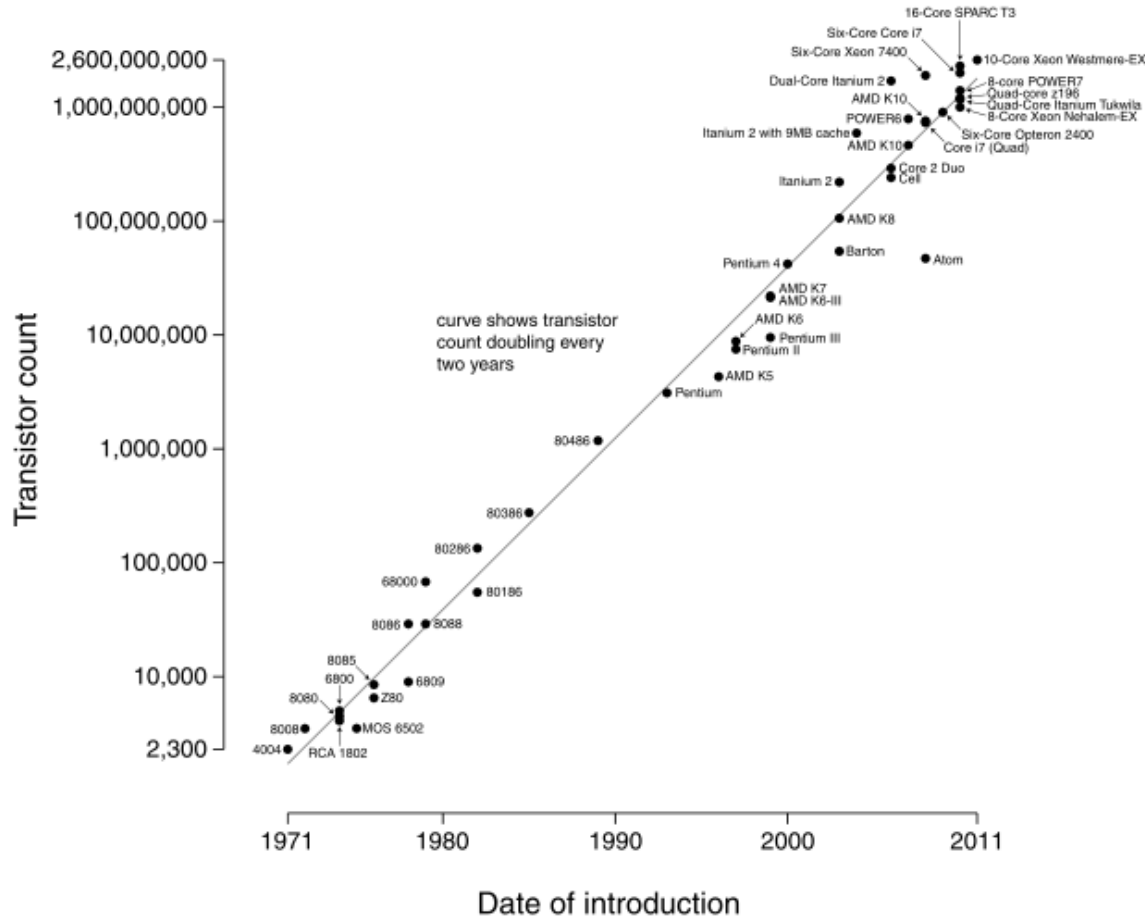
Bên trong máy tính

❖ Công nghệ xây dựng Bộ xử lý và Bộ nhớ

- **Transistor:** Công tắc đóng/mở được điều khiển bằng điện.
- **Very large scale integrated circuit (VLSI):** Mạch tích hợp chứa hàng trăm ngàn đến hàng triệu transistor.
- **Moore's law:** số lượng transistor của mạch tích hợp sẽ tăng gấp đôi trong khoảng thời gian mỗi 18–24 tháng
(Gordon Moore, một trong những nhà sáng lập Intel vào những năm 1960s.)

Bên trong máy tính

❖ Công nghệ xây dựng Bộ xử lý và Bộ nhớ



Hình 9. Định luật Moore. Nguồn: http://en.wikipedia.org/wiki/Moore's_law

1. Introduction
2. Below your program
3. Under the Covers
- 4. Hiệu suất (Performance)**

Hiệu suất

Thời gian đáp ứng (Response time): Cũng gọi là thời gian thực thi (**execution time**), là tổng thời gian để máy tính hoàn thành một tác vụ, bao gồm thao tác truy cập ổ đĩa, truy cập bộ nhớ, hoạt động I/O, thời gian thực thi của hệ điều hành (operating system overhead), v.v...

Thông năng (Throughput): Cũng gọi là **bandwidth**, là số lượng tác vụ hoàn thành trong một đơn vị thời gian:

$$\text{Performance}_x = \frac{1}{\text{Execution time}_x}$$

$$\text{Performance}_x > \text{Performance}_y$$

$$\frac{1}{\text{Execution time}_x} > \frac{1}{\text{Execution time}_y}$$

$$\text{Execution time}_y > \text{Execution time}_x$$

Hiệu suất

Chu kỳ xung đồng hồ (Clock cycle): Cũng gọi là **tick, clock tick, clock period, clock, cycle**. Thời gian hoàn thành một chu kỳ clock, thường là clock của bộ xử lý chạy ở tốc độ cố định.

Clock period (clock cycle time): thời gian của một chu kỳ clock. Ta có:

$$\text{CPU execution time for a program} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

Hoặc (clock rate và clock period là nghịch đảo của nhau):

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

Hiệu suất

CPI (clock cycle per instruction): Số lượng chu kỳ clock cần để thực thi một lệnh của chương trình.

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \text{Average clock cycles per instruction}$$

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

Hoặc:

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

Hiệu suất

MIPS (Million instructions per second): Một cách đo tốc độ thực thi của chương trình, dựa trên số lượng triệu lệnh trên giây. MIPS được tính bằng số lượng lệnh chia cho tích của thời gian thực thi và giá trị 10^6 .

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

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

Ví dụ 1.

Một chương trình máy tính chạy trong 10 giây trên máy tính A có tần số xung clock 2GHz. Một nhà thiết kế sẽ xây dựng máy tính B chạy chương trình này chỉ trong 6 giây. Nhà thiết kế quyết định tăng tần số xung clock, nhưng việc tăng giá trị này ảnh hưởng đến những phần thiết kế khác của CPU, khiến máy tính B yêu cầu nhiều chu kỳ clock hơn máy tính A 1.2 lần để chạy chương trình. Hỏi tần số xung clock nhà thiết kế đã dùng là bao nhiêu?

Hiệu suất

Ví dụ 1.

Let's first find the number of clock cycles required for the program on A:

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

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{2 \times 10^9 \frac{\text{cycles}}{\text{second}}}$$

$$\cdot \quad \text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

CPU time for B can be found using this equation:

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

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

$$\text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4 \text{ GHz}$$

To run the program in 6 seconds, B must have twice the clock rate of A.

Ví dụ 2.

Giả sử có hai cách hiện thực cho cùng một kiến trúc tập lệnh. Máy tính A có chu kỳ clock 250ps và CPI là 2.0 khi chạy một chương trình, và máy tính B có chu kỳ clock 500ps và CPI 1.2 khi chạy cùng chương trình trên. Máy tính nào chạy chương trình trên nhanh hơn và nhanh hơn bao nhiêu?

Hiệu suất

Ví dụ 2.

We know that each computer executes the same number of instructions for the program; let's call this number I . First, find the number of processor clock cycles for each computer:

$$\text{CPU clock cycles}_A = I \times 2.0$$

$$\text{CPU clock cycles}_B = I \times 1.2$$

Now we can compute the CPU time for each computer:

$$\begin{aligned}\text{CPU time}_A &= \text{CPU clock cycles}_A \times \text{Clock cycle time} \\ &= I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}\end{aligned}$$

Likewise, for B:

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

Clearly, computer A is faster. The amount faster is given by the ratio of the execution times:

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

We can conclude that computer A is 1.2 times as fast as computer B for this program.

Ví dụ 3.

So sánh hiệu suất thực thi của đoạn mã chương trình:

Một người viết trình biên dịch (compiler) đang chọn lựa giữa hai chuỗi thực thi code (code sequence) cho một máy tính. Nhà thiết kế phần cứng của máy tính này cung cấp thông tin như sau:

	CPI for each instruction class		
	A	B	C
CPI	1	2	3

Đối với một ngôn ngữ lập trình cấp cao cụ thể, hai chuỗi code này có số lượng lệnh tương ứng như sau:

Code sequence	Instruction counts for each instruction class		
	A	B	C
1	2	1	2
2	4	1	1

Chuỗi code nào tốn nhiều lệnh hơn? Chuỗi code nào thực thi nhanh hơn? CPI của mỗi chuỗi?

Ví dụ 3.

Sequence 1 executes $2 + 1 + 2 = 5$ instructions. Sequence 2 executes $4 + 1 + 1 = 6$ instructions. Therefore, sequence 1 executes fewer instructions.

We can use the equation for CPU clock cycles based on instruction count and CPI to find the total number of clock cycles for each sequence:

$$\text{CPU clock cycles} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

This yields

$$\text{CPU clock cycles}_1 = (2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10 \text{ cycles}$$

$$\text{CPU clock cycles}_2 = (4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9 \text{ cycles}$$

So code sequence 2 is faster, even though it executes one extra instruction. Since code sequence 2 takes fewer overall clock cycles but has more instructions, it must have a lower CPI. The CPI values can be computed by

Hiệu suất

Tóm lại, những yếu tố cơ bản để quyết định hiệu suất máy tính gồm:

Yếu tố	Đơn vị đo
Thời gian CPU thực thi một chương trình (CPU execution time)	Giây (Seconds)
Số lượng lệnh (Instruction count)	Số lượng lệnh được thực thi cho một chương trình
CPI (Clock cycles per instruction)	Số lượng chu kỳ clock để thực thi một lệnh
Thời gian một chu kỳ clock (Clock cycle time)	Giây

Hiệu suất

Hiệu suất của chương trình phụ thuộc vào thuật toán, ngôn ngữ, trình biên dịch, kiến trúc và phần cứng máy tính. Bảng bên dưới tóm tắt sự ảnh hưởng của những yếu tố này lên hiệu suất của CPU.

Yếu tố phần cứng/phần mềm	Tác động vào gì?	Như thế nào?
Thuật toán	Số lượng lệnh, và có thể cả CPI	Thuật toán sẽ quyết định có bao nhiêu lệnh trong chương trình nguồn và theo đó là số lượng lệnh mà CPU phải thực thi. Thuật toán cũng có thể ảnh hưởng đến CPI về khía cạnh lệnh chạy nhanh hay chậm, ví dụ, nếu thuật toán có nhiều phép tính trên số floating-point, khả năng sẽ có CPI cao hơn.
Ngôn ngữ lập trình	Số lượng lệnh, CPI	Ngôn ngữ lập trình chắc chắn ảnh hưởng đến số lượng lệnh, vì các chương trình viết bằng ngôn ngữ lập trình sẽ được chuyển thành lệnh cho bộ xử lý. Ngoài ra, CPI cũng có thể bị ảnh hưởng, ví dụ, một ngôn ngữ có tính năng hỗ trợ mạnh về trừu tượng hóa dữ liệu (như Java) sẽ có nhiều lời gọi lệnh không trực tiếp, do đó sẽ sử dụng nhiều lệnh có CPI cao.
Trình biên dịch	Số lượng lệnh, CPI	Vì trình biên dịch thực hiện việc chuyển các lệnh từ ngôn ngữ cấp cao sang ngôn ngữ máy nên chắc chắn tác động đến số lượng lệnh và CPI. Vai trò của trình biên dịch rất phức tạp và ảnh hưởng đến CPI theo một cách phức tạp.
Kiến trúc tập lệnh	Số lượng lệnh, tần số xung clock, CPI	Kiến trúc tập lệnh tác động đến cả 3 yếu tố của hiệu năng CPU, vì nó quyết định các lệnh cần để thực hiện một chức năng, số lượng chu kỳ cho một lệnh, và tần số clock tổng quan của bộ xử lý.