

## **Chương 1: TỔNG QUAN VỀ KIẾN TRÚC MÁY TÍNH**

Kiến trúc máy tính đề cập đến các thuộc tính mà lập trình viên có thể quan sát được. Nói cách khác, đó là các thuộc tính có ảnh hưởng trực tiếp đến việc thực thi một chương trình, ví dụ tập chỉ thị (lệnh) của máy tính, số bit được sử dụng để biểu diễn dữ liệu, cơ chế nhập/ xuất, kỹ thuật định địa chỉ bộ nhớ .v.v...

Môn học kiến trúc máy tính là môn học khảo sát cấu trúc và chức năng của máy tính.

### **1.1. CHỨC NĂNG VÀ CẤU TRÚC CỦA MÁY TÍNH:**

#### **1.1.1. Chức năng của máy tính:**

Máy tính điện tử: là một loại thiết bị đặc biệt có thể được dùng để giải quyết một công việc do con người đặt ra thông qua việc thực hiện lần lượt các câu lệnh của một chương trình mô tả công việc đó.

Để thực hiện một công việc như vậy, máy tính cần phải:

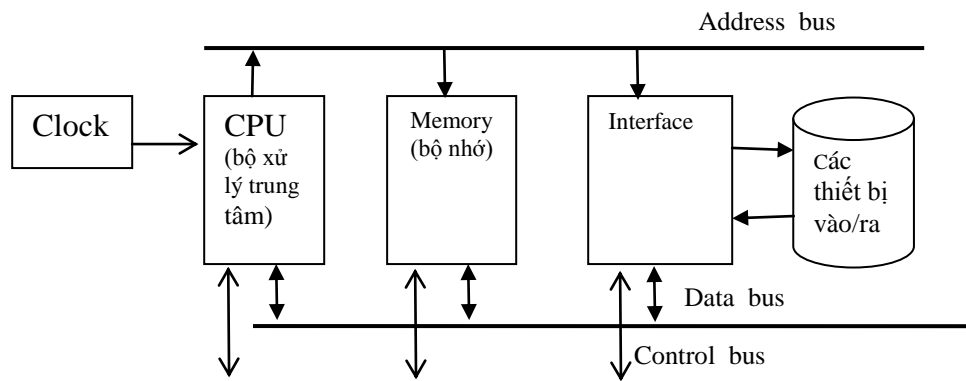
- Tiếp nhận các số liệu ban đầu được đưa vào từ bên ngoài.
- Thực hiện các phép tính cần thiết để xử lý các số liệu đó.
- Lưu trữ các kết quả thực hiện theo một trật tự mong muốn.
- Đưa ra thông tin về kết quả thực hiện chương trình ở dạng thích hợp để trao đổi với bên ngoài (con người hoặc các thiết bị khác)

Do vậy, máy tính có chức năng xử lý thông tin, chức năng trao đổi vào/ra và các chức năng nhớ.

#### **1.1.2. Cấu trúc của máy tính**

##### *1.1.2.1. Sơ đồ cấu trúc phần cứng*

Phần cứng được hiểu là các bộ phận cấu thành có của máy tính như các vi mạch, bảng mạch in, dây nối nguồn cung cấp, máy đọc, máy in, và các thiết bị vào ra khác. Mặc dù trải qua nhiều giai đoạn phát triển, hầu hết các máy tính hiện nay đều theo kiến trúc Von Neumann và có sơ đồ cấu trúc điển hình như sau:



Hình 1-1 Sơ đồ cấu trúc phần cứng của máy tính

### 1.1.2.2. Khối xử lý trung tâm (CPU)

Khối xử lý trung tâm được coi như là bộ não của máy tính, chức năng của khối này là lần lượt đọc các lệnh của chương trình từ bộ nhớ và thực hiện chúng. Trong các máy tính hiện nay, CPU được chế tạo gọn trên một chip vi mạch được gọi là bộ vi xử lý (Micro Processor – MP). CPU gồm các bộ phận sau:

- Bộ điều khiển (Control Unit –CU) tìm đọc các lệnh từ bộ nhớ và giải mã xác định các bước thực hiện lệnh.
- Bộ số học logic (Arithmetic Logical Unit – ALU) thực hiện các phép toán (cộng , trừ ...) và logic (and, or..) lên các toán hạng của lệnh.
- Các thanh ghi (Registers) để chứa các số liệu tạm thời, các thông tin điều khiển cần thiết của CPU.
- Bộ tạo nhịp (Clock) tạo các xung để điều khiển hoạt động của CPU theo trình tự và đồng bộ sự hoạt động của các khối trong toàn hệ thống.

### 1.1.2.3. Khối bộ nhớ (memory)

Trong máy tính, Bộ nhớ, dùng để chứa các lệnh và dữ liệu phục vụ chương trình. Bộ nhớ được chia thành các đơn vị cơ bản gọi là ô nhớ. Đơn vị thông tin cơ bản trong máy tính hiện nay là byte (8bit), do vậy bộ nhớ cũng được tổ chức theo đơn vị byte.

Các ô nhớ được đánh thứ tự theo mã nhị phân.

Để truy nhập (đọc/viết) một ô nhớ, bộ vi xử lý đưa ra địa chỉ của nó và kích hoạt tín hiệu điều khiển (Read hoặc Write) trước khi nhận hoặc gửi số liệu. Có hai dạng cơ bản là:

ROM (Read Only Memory) bộ nhớ cố định chỉ có thể đọc không viết được, không bị mất nội dung khi mất nguồn cung cấp. Bộ nhớ ROM dùng để chứa các chương trình như chương trình khởi động máy tính (ROMBIOS) hoặc các chương trình điều khiển hệ thống trong các máy tính điều khiển.

RAM (Random Access Memory)- bộ nhớ có thể đọc và viết nội dung vào nó, khi mất nguồn cung cấp, nội dung của bộ nhớ Ram cũng mất. Ram được dùng để nạp các chương trình của máy tính khi hoạt động.

#### 1.1..2.4. *Khối các thiết bị vào ra (Input/Output)*

Gồm các thiết bị ngoài như bàn phím, màn hình, máy in, ổ đĩa, các hệ thống truyền tin... gọi chung là các thiết bị ngoại vi. Các thiết bị ngoại vi thường có tốc độ và mức tín hiệu rất khác so với hệ thống trung tâm của máy tính (CPU, các bộ nhớ) để ghép nối các thiết bị ngoại vi với hệ thống trung tâm của máy tính cần có một số mạch hỗ trợ gọi là Interface hoặc cổng vào ra. Mỗi cổng vào ra có một địa chỉ xác định. Về cơ bản việc đọc viết một cổng vào cũng tương tự như đọc/ viết một ô nhớ.

#### 1.1..2.5. *BUS*

Là phương tiện chuyên chở thông tin giữa các bộ phận trong hệ thống máy tính. Về thiết bị nối với bus cần có bộ phận gọi là bus Interface. Có 3 nhóm bus:

- Address bus (địa chỉ) gồm nhóm bus mang thông tin về địa chỉ bộ nhớ. Địa chỉ là do bộ vi xử lý đưa ra để thâm nhập bộ nhớ hoặc các cổng vào ra , nó có chiều từ bộ vi xử lý.
- Data bus (số liệu) là nhóm bus mang thông tin về dữ liệu cần trao đổi giữa bộ vi xử lý với bộ nhớ hoặc các cổng vào/ ra. Trong quá trình làm việc của máy tính, dữ liệu có thể được viết ra từ bộ vi xử lý hoặc được đọc về bộ vi xử lý. Như vậy data bus là bus 2 chiều.
- Control bus (điều khiển và trạng thái) là nhóm bus mang thông tin điều khiển hệ thống (ví dụ đọc, hay viết, các yêu cầu ngắt... ) và trạng thái của các bộ phận (ví dụ ready, busy..) Control bus có thể truyền tín hiệu 2 chiều (tới hoặc từ bộ vi xử lý). Mỗi tín hiệu trong Control bus thường chỉ có một chiều nhất định.

## 1.2. CÁC THỂ HỆ MÁY TÍNH

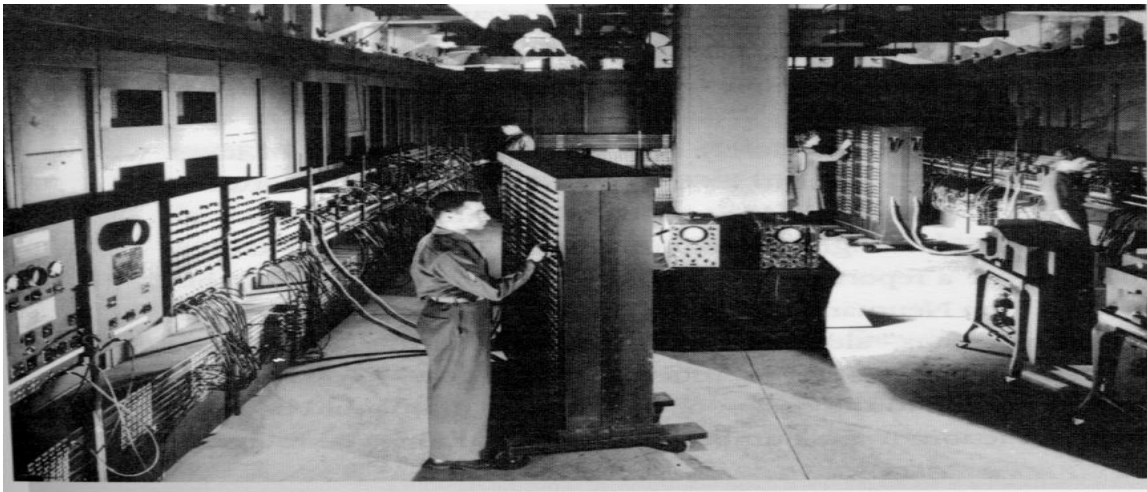
Sự phát triển của máy tính được mô tả dựa trên sự tiến bộ của các công nghệ chế tạo các linh kiện cơ bản của máy tính như: bộ xử lý, bộ nhớ, các thiết bị ngoại vi,... Ta có thể nói máy tính điện tử số trải qua bốn thế hệ liên tiếp. Việc chuyển từ thế hệ trước sang thế hệ sau được đặc trưng bằng một sự thay đổi cơ bản về công nghệ.

### **1.2.1. Thế hệ đầu tiên (1946-1957)**

ENIAC (Electronic Numerical Integrator and Computer) là máy tính điện tử số đầu tiên do Giáo sư Mauchly và người học trò Eckert tại Đại học Pennsylvania thiết kế vào năm 1943 và được hoàn thành vào năm 1946. Đây là một máy tính khổng lồ cân nặng 30 tấn chiếm một diện tích xấp xỉ 1393m<sup>2</sup>. ENIAC bao gồm: 18.000 đèn điện tử, 1.500 công tắc tự động và tiêu thụ 140KW giờ. Nó có 20 thanh ghi 10 bit (tính toán trên số thập phân). Có khả năng thực hiện 5.000 phép toán cộng trong một giây. Công việc lập trình được thực hiện bằng tay bằng cách đấu nối các đầu cắm điện và dùng các ngắt điện.

Giáo sư toán học John Von Neumann đã đưa ra ý tưởng thiết kế máy tính IAS (Princeton Institute for Advanced Studies): chương trình được lưu trong bộ nhớ, bộ điều khiển sẽ lấy lệnh và biến đổi giá trị của dữ liệu trong phần bộ nhớ, bộ làm toán và luận lý (ALU: Arithmetic And Logic Unit) được điều khiển để tính toán trên dữ liệu nhị phân, điều khiển hoạt động của các thiết bị vào ra. Đây là một ý tưởng nền tảng cho các máy tính hiện đại ngày nay. Máy tính này còn được gọi là *máy tính Von Neumann*.

Vào những năm đầu của thập niên 50, những máy tính thương mại đầu tiên được đưa ra thị trường.



### Hình 1.2: Máy tính ENIAC

#### **1.2.2. Thế hệ thứ hai (1958-1964)**

Công ty Bell đã phát minh ra *transistor* vào năm 1947 và do đó thế hệ thứ hai của máy tính được đặc trưng bằng sự thay thế các đèn điện tử bằng các transistor lưỡng cực. Tuy nhiên, đến cuối thập niên 50, máy tính thương mại dùng transistor mới xuất hiện trên thị trường. Kích thước máy tính giảm, rẻ tiền hơn, tiêu tốn năng lượng ít hơn. Vào thời điểm này, mạch in và bộ nhớ bằng xuyên từ được dùng. Ngôn ngữ cấp cao xuất hiện (như FORTRAN năm 1956, COBOL năm 1959, ALGOL năm 1960) và hệ điều hành kiểu tuần tự (Batch Processing) được dùng. Trong hệ điều hành này, chương trình của người dùng thứ nhất được chạy, xong đến chương trình của người dùng thứ hai và cứ thế tiếp tục.

#### **1.2.3. Thế hệ thứ ba (1965-1971)**

Thế hệ thứ ba được đánh dấu bằng sự xuất hiện của các *mạch kết* (mạch tích hợp - IC: Integrated Circuit). Các mạch kết mật độ tích hợp thấp (SSI: Small Scale Integration) có thể chứa vài chục linh kiện và mạch kết mật độ tích hợp trung bình (MSI: Medium Scale Integration) chứa hàng trăm linh kiện trên mạch tích hợp.

Mạch in nhiều lớp xuất hiện, bộ nhớ bán dẫn bắt đầu thay thế bộ nhớ bằng xuyên từ. Máy tính đa chương trình và hệ điều hành chia thời gian được dùng.

#### **1.2.4. Thế hệ thứ tư (1972-????)**

Thế hệ thứ tư được đánh dấu bằng các *IC có mật độ tích hợp cao* (LSI: Large Scale Integration) có thể chứa hàng ngàn linh kiện. Các IC mật độ tích hợp rất cao (VLSI: Very Large Scale Integration) có thể chứa hơn 10 ngàn linh kiện trên mạch. Hiện nay, các chip VLSI chứa hàng triệu linh kiện.

Với sự xuất hiện của bộ vi xử lý (microprocessor) chứa cả phần thực hiện và phần điều khiển của một bộ xử lý, sự phát triển của công nghệ bán dẫn các máy vi tính đã được chế tạo và khởi đầu cho các thế hệ máy tính cá nhân.

Các bộ nhớ bán dẫn, bộ nhớ cache, bộ nhớ ảo được dùng rộng rãi.

Các kỹ thuật cải tiến tốc độ xử lý của máy tính không ngừng được phát triển: kỹ thuật ống dẫn, kỹ thuật vô hướng, xử lý song song mức độ cao,...

Năm 1975 công ty MITS ( Mỹ ) giới thiệu chiếc máy tính cá nhân Altair đầu tiên trên thế giới, chiếc máy này sử dụng bộ vi xử lý 8080 của Intel, chiếc máy tính đầu tiên không có màn hình mà chỉ hiện kết quả thông qua các đèn Led (Hình 1.3)



Hình 1.3



Năm 1977 công ty Apple đưa ra thị trường máy tính AppleII có màn hình và bàn phím.  
Hình 1.4

Năm 1981 công ty IBM sản xuất máy tính PC có hệ thống mở, tức là máy có nhiều khe cắm mở rộng để có thể cắm thêm các thứ khác vào đó, về sau thiết kế này đã phát triển thành tiêu chuẩn của máy tính ngày nay.

Công ty IBM (một công ty khổng lồ lúc đó) đã tìm đến một công ty nhỏ có tên là Microsoft để thuê viết phần mềm cho máy tính PC của mình, đó là cơ hội ngàn năm có một để cho Microsoft trở thành công ty phần mềm lớn nhất thế giới hiện nay. Bill Gate làm việc suốt ngày để hoàn thành hệ điều hành MS DOS cho công ty IBM .



Hình 1.5

Sau khi phát minh ra chuẩn PC mở rộng, IBM đã cho phép các nhà sản xuất PC trên thế giới nhái theo chuẩn của IBM và chuẩn máy tính IBM PC đã nhanh chóng phát triển thành hệ thống sản xuất máy PC khổng lồ trên toàn thế giới.

### **1.2.5. Khuynh hướng hiện tại**

Việc chuyển từ thế hệ thứ tư sang thế hệ thứ 5 còn chưa rõ ràng. Người Nhật đã và đang đi tiên phong trong các chương trình nghiên cứu để cho ra đời thế hệ thứ 5 của máy tính, thế hệ của những máy tính thông minh, dựa trên các ngôn ngữ trí tuệ nhân tạo như LISP và PROLOG,... và những giao diện người - máy thông minh. Đến thời điểm này, các nghiên cứu đã cho ra các sản phẩm bước đầu và gần đây nhất (2004) là sự ra mắt sản

phẩm người máy thông minh gần giống với con người nhất: ASIMO (*Advanced Step Innovative Mobility: Bước chân tiên tiến của đổi mới và chuyển động*). Với hàng trăm nghìn máy móc điện tử tối tân đặt trong cơ thể, ASIMO có thể lên/xuống cầu thang một cách uyển chuyển, nhận diện người, các cử chỉ hành động, giọng nói và đáp ứng một số mệnh lệnh của con người. Thậm chí, nó có thể bắt chước cử động, gọi tên người và cung cấp thông tin ngay sau khi bạn hỏi, rất gần gũi và thân thiện. Hiện nay có nhiều công ty, viện nghiên cứu của Nhật thuê Asimo tiếp khách và hướng dẫn khách tham quan như: Viện Bảo tàng Khoa học năng lượng và Đổi mới quốc gia, hãng IBM Nhật Bản, Công ty điện lực Tokyo. Hãng Honda bắt đầu nghiên cứu ASIMO từ năm 1986 dựa vào nguyên lý chuyển động bằng hai chân. Cho tới nay, hãng đã chế tạo được 50 robot ASIMO.

Các tiến bộ liên tục về mật độ tích hợp trong VLSI đã cho phép thực hiện các mạch vi xử lý ngày càng mạnh (8 bit, 16 bit, 32 bit và 64 bit với việc xuất hiện các bộ xử lý RISC năm 1986 và các bộ xử lý siêu vô hướng năm 1990). Chính các bộ xử lý này giúp thực hiện các máy tính song song với từ vài bộ xử lý đến vài ngàn bộ xử lý. Điều này làm các chuyên gia về kiến trúc máy tính tiên đoán thế hệ thứ 5 là thế hệ các máy tính xử lý song song.

<b>Thế hệ</b>	<b>Năm</b>	<b>Kỹ thuật</b>	<b>Sản phẩm mới</b>	<b>Hãng sản xuất và máy tính</b>
1	1946-1957	Đèn điện tử	Máy tính điện tử tung ra thị trường	IBM 701, UNIVAC
2	1958-1964	Transistors	Máy tính rẻ tiền	Burroughs 6500, NCR, CDC 6600, Honeywell
3	1965-1971	Mach IC	Máy tính mini	50 hãng mới: DEC PDP-11, Data general, Nova
4	1972-????	LSI - VLSI	Máy tính cá nhân và trạm làm việc	Apple II, IBM-PC, Appolo DN 300, Sun 2
5 ??	????-????	Xử lý song song	Máy tính đa xử lý. Đa máy	Sequent ? Thinking

			tính	Machine Inc.? Honda, Casio
--	--	--	------	-------------------------------

**Bảng 1.1: Các thế hệ máy tính**

### 1.3. PHÂN LOẠI MÁY TÍNH

Thông thường máy tính được phân loại theo tính năng kỹ thuật và giá tiền.

#### 1.3.1. Các siêu máy tính (Super Computer):

Là các máy tính đắt tiền nhất và tính năng kỹ thuật cao nhất. Giá bán một siêu máy tính từ vài triệu USD. Các siêu máy tính thường là các máy tính vectơ hay các máy tính dùng kỹ thuật vô hướng và được thiết kế để tính toán khoa học, mô phỏng các hiện tượng. Các siêu máy tính được thiết kế với kỹ thuật xử lý song song với rất nhiều bộ xử lý (hàng ngàn đến hàng trăm ngàn bộ xử lý trong một siêu máy tính).

#### 1.3.2. Các máy tính lớn (Mainframe)

Là loại máy tính đa dụng. Nó có thể dùng cho các ứng dụng quản lý cũng như các tính toán khoa học. Dùng kỹ thuật xử lý song song và có hệ thống vào ra mạnh. Giá một máy tính lớn có thể từ vài trăm ngàn USD đến hàng triệu USD.

#### 1.3.3. Máy tính mini (Minicomputer)

Là loại máy cỡ trung, giá một máy tính mini có thể từ vài chục ngàn USD đến vài trăm ngàn USD.

#### 1.3.4. Máy vi tính (Microcomputer)

Là loại máy tính dùng bộ vi xử lý, giá một máy vi tính có thể từ vài trăm USD đến vài ngàn USD.

### 1.4. THÀNH QUẢ CỦA MÁY TÍNH

#### 2.1.1. Lịch sử phát triển của máy tính

Mạch điện tử quan trọng nhất của máy tính là đơn vị xử lý trung tâm, gọi tắt là CPU (Central Processing Unit). Từ khi xuất hiện công nghệ vi điện tử CPU được gọi là mạch vi xử lý, gọi tắt là MP hoặc  $\mu$ P (Micro Processor). Sau đây ta sẽ tìm hiểu lịch sử phát triển kiến trúc bộ vi xử lý (BXL) của Intel.

##### ➤ BXL 4 bit

4004 là BXL đầu tiên được Intel giới thiệu vào tháng 11 năm 1971, sử dụng trong máy tính (calculator) của Busicom. 4004 có tốc độ 0.1MHz, khả năng xử lý 0,06 triệu lệnh mỗi



giây (million instructions per second - MIPS); được sản xuất trên công nghệ 10 $\mu$ m, có 2.300 transistor (bóng bán dẫn), bộ nhớ mở rộng đến 640 byte.

4040, phiên bản cải tiến của 4004 được giới thiệu vào năm 1974, có 3.000 transistor, tốc độ từ 500 KHz đến 740KHz.

➤ **BXL 8bit**

8008 (năm 1972) được sử dụng trong thiết bị đầu cuối Datapoint 2200 của Computer Terminal Corporation (CTC). 8008 có tốc độ 200kHz, sản xuất trên công nghệ 10 $\mu$ m, với 3.500 transistor, bộ nhớ mở rộng đến 16KB.

8080 (năm 1974) sử dụng trong máy tính Altair 8080, có tốc độ gấp 10 lần 8008 (2MHz), sản xuất trên công nghệ 6 $\mu$ m, khả năng xử lý 0,64 MIPS với 6.000 transistor, có 8 bit bus dữ liệu và 16 bit bus địa chỉ, bộ nhớ mở rộng tới 64KB.

8085 (năm 1976) sử dụng trong Toledo scale và những thiết bị điều khiển ngoại vi. 8085 có tốc độ 2MHz, sản xuất trên công nghệ 3  $\mu$ m, với 6.500 transistor, có 8 bit bus dữ liệu và 16 bit bus địa chỉ, bộ nhớ mở rộng 64KB.

➤ **BXL 16bit**

8086 xuất hiện tháng 6 năm 1978, sử dụng trong những thiết bị tính toán di động. 8086 được sản xuất trên công nghệ 3 $\mu$ m, với 29.000 transistor, có 16 bit bus dữ liệu và 20 bit bus địa chỉ, bộ nhớ mở rộng 1MB. Các phiên bản của 8086 gồm 5, 8 và 10 MHz.

8088 trình làng vào tháng 6 năm 1979, là BXL được IBM chọn đưa vào chiếc máy tính (PC) đầu tiên của mình; điều này cũng giúp Intel trở thành nhà sản xuất BXL máy tính lớn nhất trên thế giới. 8088 giống hệt 8086 nhưng có khả năng quản lý địa chỉ dòng lệnh. 8088 cũng sử dụng công nghệ 3 $\mu$ m, 29.000 transistor, kiến trúc 16 bit bên trong và 8 bit bus dữ liệu ngoài, 20 bit bus địa chỉ, bộ nhớ mở rộng tới 1MB. Các phiên bản của 8088 gồm 5 MHz và 8 MHz.

80186 (năm 1982) còn gọi là IAPX 186. Sử dụng chủ yếu trong những ứng dụng nhúng, bộ điều khiển thiết bị đầu cuối. Các phiên bản của 80186 gồm 10 và 12 MHz.

80286 (năm 1982) được biết đến với tên gọi 286, là BXL đầu tiên của Intel có thể chạy được tất cả ứng dụng viết cho các BXL trước đó, được dùng trong PC của IBM và các PC tương thích. 286 có 2 chế độ hoạt động: chế độ thực (real mode) với chương trình

DOS theo chế độ mô phỏng 8086 và không thể sử dụng quá 1 MB RAM; chế độ bảo vệ (protect mode) gia tăng tính năng của bộ vi xử lý, có thể truy xuất đến 16 MB bộ nhớ.

286 sử dụng công nghệ 1,5 $\mu$ m, 134.000 transistor, bộ nhớ mở rộng tới 16 MB. Các phiên bản của 286 gồm 6, 8, 10, 12,5, 16, 20 và 25MHz.

➤ **BXL 32bit** vi kiến trúc NetBurst (NetBurst MICRO-ARCHITECTURE)

Intel386 gồm các họ 386DX, 386SX và 386SL. Intel386DX là BXL 32 bit đầu tiên Intel giới thiệu vào năm 1985, được dùng trong các PC của IBM và PC tương thích. Intel386 là một bước nhảy vọt so với các BXL trước đó. Đây là BXL 32 bit có khả năng xử lý đa nhiệm, nó có thể chạy nhiều chương trình khác nhau cùng một thời điểm. 386 sử dụng các thanh ghi 32 bit, có thể truyền 32 bit dữ liệu cùng lúc trên bus dữ liệu và dùng 32 bit để xác định địa chỉ. Cũng như BXL 80286, 80386 hoạt động ở 2 chế độ: real mode và protect mode.

386DX sử dụng công nghệ 1,5 $\mu$ m, 275.000 transistor, bộ nhớ mở rộng tới 4GB. Các phiên bản của 386DX gồm 16, 20, 25 và 33MHz (công nghệ 1  $\mu$ m). 386SX (năm1988) sử dụng công nghệ 1,5  $\mu$ m, 275.000 transistor, kiến trúc 32 bit bên trong, 16 bit bus dữ liệu ngoài, 24 bit bus địa chỉ, bộ nhớ mở rộng 16MB; gồm các phiên bản 16, 20, 25 và 33 MHz.

386SL (năm1990) được thiết kế cho thiết bị di động, sử dụng công nghệ 1  $\mu$ m, 855.000 transistor, bộ nhớ mở rộng 4GB; gồm các phiên bản 16, 20, 25MHz. 486DX ra đời năm 1989 với cấu trúc bus dữ liệu 32 bit.

486DX có bộ nhớ sơ cấp (L1 cache) 8 KB để giảm thời gian chờ dữ liệu từ bộ nhớ đưa đến, bộ đồng xử lý toán học được tích hợp bên trong. Ngoài ra, 486DX được thiết kế hàng lệnh (pipeline), có thể xử lý một chỉ lệnh trong một xung nhịp.

486DX sử dụng công nghệ 1 $\mu$ m, 1,2 triệu transistor, bộ nhớ mở rộng 4GB; gồm các phiên bản 25MHz, 33MHz và 50 MHz (0,8 $\mu$ m). 486SX (năm 1991) dùng trong dòng máy tính cấp thấp, có thiết kế giống hệ 486DX nhưng không tích hợp bộ đồng xử lý toán học. 486DX sử dụng công nghệ 1  $\mu$ m (1,2 triệu transistor) và 0,8 $\mu$ m (0,9 triệu transistor), bộ nhớ mở rộng 4GB; gồm các phiên bản 16, 20, 25, 33MHz.

486SL (năm 1992) là *BXL đầu tiên dành cho máy tính xách tay* (MTXT), sử dụng công nghệ 0,8  $\mu\text{m}$ , 1,4 triệu transistor, bộ nhớ mở rộng 4GB; gồm các phiên bản 20, 25 và 33 MHz.

➤ BXL Intel Pentium, thế hệ kế tiếp 486 ra đời năm 1993. Cải tiến lớn nhất của Pentium là thiết kế hai hàng lệnh (pipeline), dữ liệu bên trong có khả năng thực hiện hai chỉ lệnh trong một chu kỳ, do đó Pentium có thể xử lý chỉ lệnh nhiều gấp đôi so với 80486 DX trong cùng thời gian. Bộ nhớ sơ cấp 16KB gồm 8 KB chứa dữ liệu và 8 KB khác để chứa lệnh. Bộ đồng xử lý toán học được cải tiến giúp tăng khả năng tính toán đối với các trình ứng dụng.

Pentium sử dụng công nghệ 0,8  $\mu\text{m}$  chứa 3,1 triệu transistor, có các tốc độ 60, 66 MHz (socket 4 273 chân, PGA). Các phiên bản 75, 90, 100, 120 MHz sử dụng công nghệ 0,6  $\mu\text{m}$  chứa 3,3 triệu transistor (socket 7, PGA). Phiên bản 133, 150, 166, 200 sử dụng công nghệ 0,35  $\mu\text{m}$  chứa 3,3 triệu transistor (socket 7, PGA)

Pentium MMX (năm 1996), phiên bản cải tiến của Pentium với công nghệ MMX được Intel phát triển để đáp ứng nhu cầu về ứng dụng đa phương tiện và truyền thông. MMX kết hợp với SIMD (Single Instruction Multiple Data) cho phép xử lý nhiều dữ liệu trong cùng chỉ lệnh, làm tăng khả năng xử lý trong các tác vụ đồ họa, đa phương tiện.

Pentium MMX sử dụng công nghệ 0,35  $\mu\text{m}$  chứa 4,5 triệu transistor, có các tốc độ 166, 200, 233 MHz (Socket 7, PGA).

Pentium Pro. Nối tiếp sự thành công của dòng Pentium, Pentium Pro được Intel giới thiệu vào tháng 9 năm 1995, sử dụng công nghệ 0,6 và 0,35  $\mu\text{m}$  chứa 5,5 triệu transistor, socket 8 387 chân, Dual SPGA, hỗ trợ bộ nhớ RAM tối đa 4GB. Điểm nổi bật của Pentium Pro là bus hệ thống 60 hoặc 66MHz, bộ nhớ đệm L2 (cache L2) 256KB hoặc 512KB (trong một số phiên bản). Pentium Pro có các tốc độ 150, 166, 180, 200 MHz.

Pentium II (năm 1997), phiên bản cải tiến từ Pentium Pro được sử dụng trong những dòng máy tính cao cấp, máy trạm (workstation) hoặc máy chủ (server). Pentium II có bộ nhớ đệm L1 32KB, L2 512KB, tích hợp công nghệ MMX được cải tiến giúp việc xử lý dữ liệu video, audio và đồ họa hiệu quả hơn. Pentium II có đế cắm dạng khe - Single-Edge contact (SEC) 242 chân, còn gọi là Slot1.

➤ BXL Pentium II

BXL Pentium II đầu tiên, tên mã Klamath, sản xuất trên công nghệ 0,35  $\mu\text{m}$ , có 7,5 triệu transistor, bus hệ thống 66 MHz, gồm các phiên bản 233, 266, 300MHz.

Pentium II, tên mã Deschutes, sử dụng công nghệ 0,25  $\mu\text{m}$ , 7,5 triệu transistor, gồm các phiên bản 333MHz (bus hệ thống 66MHz), 350, 400, 450 MHz (bus hệ thống 100MHz). Celeron (năm 1998) được “rút gọn” từ kiến trúc BXL Pentium II, dành cho dòng máy cấp thấp. Phiên bản đầu tiên, tên mã Covington không có bộ nhớ đệm L2 nên tốc độ xử lý khá chậm, không gây được ấn tượng với người dùng. Phiên bản sau, tên mã Mendocino, đã khắc phục khuyết điểm này với bộ nhớ đệm L2 128KB.

Covington sử dụng công nghệ 0,25  $\mu\text{m}$ , 7,5 triệu transistor, bộ nhớ đệm L1 32KB, bus hệ thống 66MHz, để cắm 242 chân Slot 1 SEPP (Single Edge Processor Package), tốc độ 266, 300 MHz.

Mendocino cũng sử dụng công nghệ 0,25  $\mu\text{m}$  có đến 19 triệu transistor, bộ nhớ đệm L1 32KB, L2 128KB, bus hệ thống 66 MHz, để cắm Slot 1 SEPP hoặc socket 370 PGA, tốc độ 300, 333, 366, 400, 433, 466, 500, 533 MHz.

### ➤ BXL Pentium III

Pentium III (năm 1999) bổ sung 70 lệnh mới (Streaming SIMD Extensions - SSE) giúp tăng hiệu suất hoạt động của BXL trong các tác vụ xử lý hình ảnh, audio, video và nhận dạng giọng nói. Pentium III gồm các tên mã Katmai, Coppermine và Tualatin.

Katmai sử dụng công nghệ 0,25  $\mu\text{m}$ , 9,5 triệu transistor, bộ nhớ đệm L1 32KB, L2 512KB, để cắm Slot 1 SECC2 (Single Edge Contact cartridge 2), tốc độ 450, 500, 550, 533 và 600 MHz (bus 100 MHz), 533, 600 MHz (bus 133 MHz).

Coppermine sử dụng công nghệ 0,18  $\mu\text{m}$ , 28,1 triệu transistor, bộ nhớ đệm L2 256 KB được tích hợp bên trong nhằm tăng tốc độ xử lý. Để cắm Slot 1 SECC2 hoặc socket 370 FC-PGA (Flip-chip pin grid array), có các tốc độ như 500, 550, 600, 650, 700, 750, 800, 850 MHz (bus 100MHz), 533, 600, 667, 733, 800, 866, 933, 1000, 1100 và 1133 MHz (bus 133MHz).

Tualatin áp dụng công nghệ 0,13  $\mu\text{m}$  có 28,1 triệu transistor, bộ nhớ đệm L1 32KB, L2 256 KB hoặc 512 KB tích hợp bên trong BXL, socket 370 FC-PGA (Flip-chip pin grid array), bus hệ thống 133 MHz. Có các tốc độ như 1133, 1200, 1266, 1333, 2900 MHz.

Celeron Coppermine (năm 2000) được “rút gọn” từ kiến trúc BXL Pentium III Coppermine, còn gọi là Celeron II, được bổ sung 70 lệnh SSE. Sử dụng công nghệ 0,18  $\mu\text{m}$  có 28,1 triệu transistor, bộ nhớ đệm L1 32KB, L2 256 KB tích hợp bên trong BXL, socket 370 FC-PGA, Có các tốc độ như 533, 566, 600, 633, 667, 700, 733, 766, 800 MHz (bus 66 MHz), 850, 900, 950, 1000, 1100, 1200, 1300 MHz (bus 1000 MHz).

Tualatin Celeron (Celeron S) (năm 2000) được “rút gọn” từ kiến trúc BXL Pentium III Tualatin, áp dụng công nghệ 0,13  $\mu\text{m}$ , bộ nhớ đệm L1 32KB, L2 256 KB tích hợp, socket 370 FC-PGA, bus hệ thống 100 MHz, gồm các tốc độ 1,0, 1,1, 1,2, 1,3 và 2,9 GHz.

#### ➤ BXL Pentium IV

Intel Pentium 4 (P4) là BXL thế hệ thứ 7 dòng x86 phổ thông, được giới thiệu vào tháng 11 năm 2000. P4 sử dụng vi kiến trúc NetBurst có thiết kế hoàn toàn mới so với các BXL cũ (PII, PIII và Celeron sử dụng vi kiến trúc P6). Một số công nghệ nổi bật được áp dụng trong vi kiến trúc NetBurst như Hyper Pipelined Technology mở rộng số hàng lệnh xử lý, Execution Trace Cache tránh tình trạng lệnh bị chậm trễ khi chuyển từ bộ nhớ đến CPU, Rapid Execution Engine tăng tốc bộ đồng xử lý toán học, bus hệ thống (system bus) 400 MHz và 533 MHz; các công nghệ Advanced Transfer Cache, Advanced Dynamic Execution, Enhanced Floating point và Multimedia Unit, Streaming SIMD Extensions 2 (SSE2) cũng được cải tiến nhằm tạo ra những BXL tốc độ cao hơn, khả năng tính toán mạnh hơn, xử lý đa phương tiện tốt hơn

Pentium 4 đầu tiên (tên mã Willamette) xuất hiện cuối năm 2000 đặt dấu chấm hết cho "triều đại" Pentium III. Willamette sản xuất trên công nghệ 0,18  $\mu\text{m}$ , có 42 triệu transistor (nhiều hơn gần 50% so với Pentium III), bus hệ thống (system bus) 400 MHz, bộ nhớ đệm tích hợp L2 256 KB, socket 423 và 478. P4 Willamette có một số tốc độ như 1,3, 1,4, 1,5, 1,6, 1,7, 1,8, 1,9, 2,0 GHz.

Socket 423 chỉ xuất hiện trong khoảng thời gian rất ngắn, từ tháng 11 năm 2000 đến tháng 8 năm 2001 và bị thay thế bởi socket 478.

Xung thực (FSB) của Pentium 4 là 100 MHz nhưng với công nghệ Quad Data Rate cho phép BXL truyền 4 bit dữ liệu trong 1 chu kỳ, nên bus hệ thống của BXL là 400 MHz.

P4 Northwood. Xuất hiện vào tháng 1 năm 2002, được sản xuất trên công nghệ 0,13  $\mu\text{m}$ , có khoảng 55 triệu transistor, bộ nhớ đệm tích hợp L2 512 KB, socket 478. Northwood có 3 dòng gồm Northwood A (system bus 400 MHz), tốc độ 1,6, 1,8, 2,0, 2,2, 2,4, 2,5, 2,6 và 2,8 GHz. Northwood B (system bus 533 MHz), tốc độ 2,26, 2,4, 2,53, 2,66, 2,8 và 3,06 GHz (riêng 3,06 GHz có hỗ trợ công nghệ siêu phân luồng Hyper Threading - HT). Northwood C (system bus 800 MHz, tất cả hỗ trợ HT), gồm 2,4, 2,6, 2,8, 3,0, 3,2, 3,4 GHz.

P4 Prescott (năm 2004). Là BXL đầu tiên Intel sản xuất theo công nghệ 90 nm, kích thước vi mạch giảm 50% so với P4 Willamette. Điều này cho phép tích hợp nhiều transistor hơn trên cùng kích thước (125 triệu transistor so với 55 triệu transistor của P4 Northwood), tốc độ chuyển đổi của transistor nhanh hơn, tăng khả năng xử lý, tính toán. Dung lượng bộ nhớ đệm tích hợp L2 của P4 Prescott gấp đôi so với P4 Northwood (1MB so với 512 KB). Ngoài tập lệnh MMX, SSE, SSE2, Prescott được bổ sung tập lệnh SSE3 giúp các ứng dụng xử lý video và game chạy nhanh hơn. Đây là giai đoạn "giai thời" giữa socket 478 - 775LGA, system bus 533 MHz - 800 MHz và mỗi sản phẩm được đặt tên riêng khiến người dùng càng bối rối khi chọn mua.

Prescott A (FSB 533 MHz) có các tốc độ 2,26, 2,4, 2,66, 2,8 (socket 478), Prescott 505 (2,66 GHz), 505J (2,66 GHz), 506 (2,66 GHz), 511 (2,8 GHz), 515 (2,93 GHz), 515J (2,93 GHz), 516 (2,93 GHz), 519J (3,06 GHz), 519K (3,06 GHz) sử dụng socket 775LGA.

Prescott E, F (năm 2004) có bộ nhớ đệm L2 1 MB (các phiên bản sau được mở rộng 2 MB), bus hệ thống 800 MHz. Ngoài tập lệnh MMX, SSE, SSE2, SSE3 tích hợp, Prescott E, F còn hỗ trợ công nghệ siêu phân luồng, một số phiên bản sau có hỗ trợ tính toán 64 bit.

Dòng sử dụng socket 478 gồm Pentium 4 HT 2.8E (2,8 GHz), 3.0E (3,0 GHz), 3.2E (3,2 GHz), 3.4E (3,4 GHz). Dòng sử dụng socket 775LGA gồm Pentium 4 HT 3.2F, 3.4F,

3.6F, 3.8F với các tốc độ tương ứng từ 3,2 GHz đến 3,8 GHz, Pentium 4 HT 517, 520, 520J, 521, 524, 530, 530J, 531, 540, 540J, 541, 550, 550J, 551, 560, 560J, 561, 570J, 571 với các tốc độ từ 2,8 GHz đến 3,8 GHz.

➤ BXL Celeron

BXL Celeron được thiết kế với mục tiêu dung hòa giữa công nghệ và giá cả, đáp ứng các yêu cầu phổ thông như truy cập Internet, Email, chat, xử lý các ứng dụng văn phòng. Điểm khác biệt giữa Celeron và Pentium là về công nghệ chế tạo và số lượng Transistor trên một đơn vị.

Celeron Willamette 128 (2002), bản "rút gọn" từ P4 Willamette, sản xuất trên công nghệ 0,18  $\mu\text{m}$ , bộ nhớ đệm L2 128 KB, bus hệ thống 400 MHz, socket 478. Celeron Willamette 128 hỗ trợ tập lệnh MMX, SSE, SSE2. Một số BXL thuộc dòng này như Celeron1.7 (1,7GHz) và Celeron 1.8 (1,8GHz).

Celeron NorthWood 128, "rút gọn" từ P4 Northwood, công nghệ 0,13  $\mu\text{m}$ , bộ nhớ đệm tích hợp L2 128 KB, bus hệ thống 400 MHz, socket 478. Celeron NorthWood 128 cũng hỗ trợ các tập lệnh MMX, SSE, SSE2, gồm Celeron 1.8A, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 tương ứng với các tốc độ từ 1,8 GHz đến 2,8 GHz.

Celeron D (Prescott 256), được xây dựng từ nền tảng P4 Prescott, sản xuất trên công nghệ 90nm, bộ nhớ đệm tích hợp L2 256 KB (gấp đôi dòng Celeron NorthWood), bus hệ thống 533 MHz, socket 478 và 775LGA. Ngoài các tập lệnh MMX, SSE, SSE2, Celeron D hỗ trợ tập lệnh SSE3, một số phiên bản sau có hỗ trợ tính toán 64 bit. Celeron D gồm 310, 315, 320, 325, 325J, 326, 330, 330J, 331, 335, 335J, 336, 340, 340J, 341, 345, 345J, 346, 350, 351, 355 với các tốc độ tương ứng từ 2,13 GHz đến 3,33 GHz.

➤ BXL Pentium 4 Extreme Edition

Pentium 4 Extreme Edition (P4EE) xuất hiện vào tháng 9 năm 2003, là BXL được Intel "ưu ái" dành cho game thủ và người dùng cao cấp. P4EE được xây dựng từ BXL Xeon dành cho máy chủ và trạm làm việc. Ngoài công nghệ HT "đình đám" thời bấy giờ, điểm nổi bật của P4EE là bổ sung bộ nhớ đệm L3 2 MB. Phiên bản đầu tiên của P4 EE (nhân Gallatin) sản xuất trên công nghệ 0,13  $\mu\text{m}$ , bộ nhớ đệm L2 512 KB, L3 2 MB, bus

hệ thống 800 MHz, sử dụng socket 478 và 775LGA, gồm P4 EE 3.2 (3,2 GHz), P4 EE 3.4 (3,4 GHz).

➤ BXL 64 BIT, Vi kiến trúc NETBURST

P4 Prescott (năm 2004)

Vi kiến trúc NetBurst 64 bit (Extended Memory 64 Technology - EM64T) đầu tiên được Intel sử dụng trong BXL P4 Prescott (tên mã Prescott 2M). Prescott 2M cũng sử dụng công nghệ 90 nm, bộ nhớ đệm L2 2 MB, bus hệ thống 800 MHz, socket 775LGA. Ngoài các tập lệnh MMX, SSE, SSE2, SSE3, công nghệ HT và khả năng tính toán 64 bit, Prescott 2M (trừ BXL 620) có hỗ trợ công nghệ Enhanced SpeedStep để tối ưu tốc độ làm việc nhằm tiết kiệm điện năng. Các BXL 6x2 có thêm công nghệ ảo hóa (Virtualization Technology). Prescott 2M có một số tốc độ như P4 HT 620 (2,8 GHz), 630 (3,0 GHz), 640 (3,2 GHz), 650 (3,4 GHz), 660, 662 (3,6 GHz) và 670, 672 (3,8 GHz).

Prescott Cedar Mill (năm 2006) hỗ trợ các tập lệnh và tính năng tương tự Prescott 2M nhưng không tích hợp Virtualization Technology. Cedar Mill được sản xuất trên công nghệ 65nm nên tiêu thụ điện năng thấp hơn, tỏa nhiệt ít hơn các dòng trước, gồm 631 (3,0 GHz), 641 (3,2 GHz), 651 (3,4 GHz) và 661 (3,6 GHz).

Pentium D (năm 2005)

Pentium D (tên mã Smithfield, 8xx) là BXL lõi kép (dual core) đầu tiên của Intel, được cải tiến từ P4 Prescott nên cũng gặp một số hạn chế như hiện tượng thắt cổ chai do băng thông BXL ở mức 800 MHz (400 MHz cho mỗi lõi), điện năng tiêu thụ cao, tỏa nhiều nhiệt. Smithfield được sản xuất trên công nghệ 90nm, có 230 triệu transistor, bộ nhớ đệm L2 2 MB (2x1 MB, không chia sẻ), bus hệ thống 533 MHz (805) hoặc 800 MHz, socket 775LGA. Ngoài các tập lệnh MMX, SSE, SSE2, SSE3, Smithfield được trang bị tập lệnh mở rộng EMT64 hỗ trợ đánh địa chỉ nhớ 64 bit, công nghệ Enhanced SpeedStep (830, 840). Một số BXL thuộc dòng này như Pentium D 805 (2,66 GHz), 820 (2,8 GHz), 830 (3,0 GHz), 840 (3,2 GHz).

Cùng sử dụng vi kiến trúc NetBurst, Pentium D (mã Presler, 9xx) được Intel thiết kế mới trên công nghệ 65nm, 376 triệu transistor, bộ nhớ đệm L2 4 MB (2x2 MB), hiệu năng cao hơn, nhiều tính năng mới và ít tổn điện năng hơn Smithfield. Pentium D 915 và 920



tốc độ 2,8 GHz, 925 và 930 (3,0GHz), 935 và 940 (3,2 GHz), 945 và 950 (3,4 GHz), 960 (3,6GHz). Presler dòng 9x0 có hỗ trợ Virtualization Technology.

#### Pentium Extreme Edition (năm 2005)

BXL lõi kép dành cho game thủ và người dùng cao cấp. Pentium EE sử dụng nhân Smithfield, Presler của Pentium D trong đó Smithfield sử dụng công nghệ 90nm, bộ nhớ đệm L2 được mở rộng đến 2 MB (2x1 MB), hỗ trợ tập lệnh MMX, SSE, SSE2, SSE3, công nghệ HT, Enhanced Intel SpeedStep Technology (EIST) và EM64T. Pentium 840 EE (3,20 GHz, bus hệ thống 800 MHz, socket 775LGA) là một trong những BXL thuộc dòng này.

Pentium EE Presler sử dụng công nghệ 65 nm, bộ nhớ đệm L2 được mở rộng đến 4 MB (2x2 MB), hỗ trợ tập lệnh MMX, SSE, SSE2, SSE3, công nghệ HT, Enhanced Intel SpeedStep Technology (EIST), EM64T và Virtualization Technology. Một số BXL thuộc dòng này là Pentium EE 955 (3,46GHz) và Pentium EE 965 (3,73GHz) có bus hệ thống 1066 MHz, socket 775.

#### ➤ BXL 64bit, kiến trúc Core

Tại diễn đàn IDF đầu năm 2006, Intel đã giới thiệu kiến trúc Intel Core với năm cải tiến quan trọng là khả năng mở rộng thực thi động (Wide Dynamic Execution), tính năng quản lý điện năng thông minh (Intelligent Power Capability), chia sẻ bộ nhớ đệm linh hoạt (Advanced Smart Cache), truy xuất bộ nhớ thông minh (Smart Memory Access) và tăng tốc phương tiện số tiên tiến (Advanced Digital Media Boost). Những cải tiến này sẽ tạo ra những BXL mạnh hơn, khả năng tính toán nhanh hơn và giảm mức tiêu thụ điện năng, tỏa nhiệt ít hơn so với kiến trúc NetBurst. Tham khảo chi tiết kiến trúc Core trong bài viết "Intel Core vi kiến trúc hai nhân chung đệm", ID: A0605\_124.

#### Intel Core 2 Duo

BXL lõi kép sản xuất trên công nghệ 65 nm, hỗ trợ SIMD instructions, công nghệ Virtualization Technology cho phép chạy cùng lúc nhiều HĐH, tăng cường bảo vệ hệ thống trước sự tấn công của virus (Execute Disable Bit), tối ưu tốc độ BXL nhằm tiết kiệm điện năng (Enhanced Intel SpeedStep Technology), quản lý máy tính từ xa (Intel

Active Management Technology). Ngoài ra, còn hỗ trợ các tập lệnh MMX, SSE, SSE2, SSE3, SSE3.

Core 2 Duo (tên mã Conroe) có 291 triệu transistor, bộ nhớ đệm L2 4 MB, bus hệ thống 1066 MHz, socket 775LGA. Một số BXL thuộc dòng này: E6600 (2,4 GHz), E6700 (2,66 GHz). Core 2 Duo (tên mã Allendale) E6300 (1,86 GHz), E6400 (2,13 GHz) có 167 triệu transistor, bộ nhớ đệm L2 2MB, bus hệ thống 1066 MHz, socket 775LGA. E4300 (1,8 GHz) xuất hiện năm 2007 có bộ nhớ đệm L2 2 MB, bus 800 MHz, không hỗ trợ Virtualization Technology.

### Core 2 Extreme

BXL lõi kép dành cho game thủ sử dụng kiến trúc Core, có nhiều đặc điểm giống với BXL Core 2 như công nghệ sản xuất 65 nm, hỗ trợ các công nghệ mới Enhanced Intel SpeedStep Technology, Intel x86-64, Execute Disable Bit, Intel Active Management, Virtualization Technology, Intel Trusted Execution Technology... các tập lệnh MMX, SSE, SSE2, SSE3, SSE3.

Core 2 Extreme (tên mã Conroe XE) (tháng 7 năm 2006) với đại diện X6800 2,93 Ghz, bộ nhớ đệm L2 đến 4 MB, bus hệ thống 1066 MHz, socket 775LGA. Cuối năm 2006, con đường phía trước của BXL tiếp tục rộng mở khi Intel giới thiệu BXL 4 nhân (Quad Core) như Core 2 Extreme QX6700, Core 2 Quad Q6300, Q6400, Q6600 và BXL 8 nhân trong vài năm tới. Chắc chắn những BXL này sẽ thỏa mãn nhu cầu người dùng đam mê công nghệ và tốc độ.

**Bảng 1.2 Tóm tắt lịch sử phát triển kiến trúc bộ vi xử lý của hãng Intel :**

Năm	Số bit	Bộ xử lý	Tốc độ xử lý	Độ rộng đường truyền	Dung lượng bộ nhớ	Số transistor
1971	4	4004	0.1MHz		640 byte	2.300
1974		4040	500->740KHz		640 byte	3.000
1972	8	8008	200 KHz	14 bit	16 KB	3.500
1974		8080	2MHz	16 bit	64KB	6.000
1976		8085	2MHz			6.500
1978	16	8086	5, 8 và 10MHz	20 bit	1MB	29.000

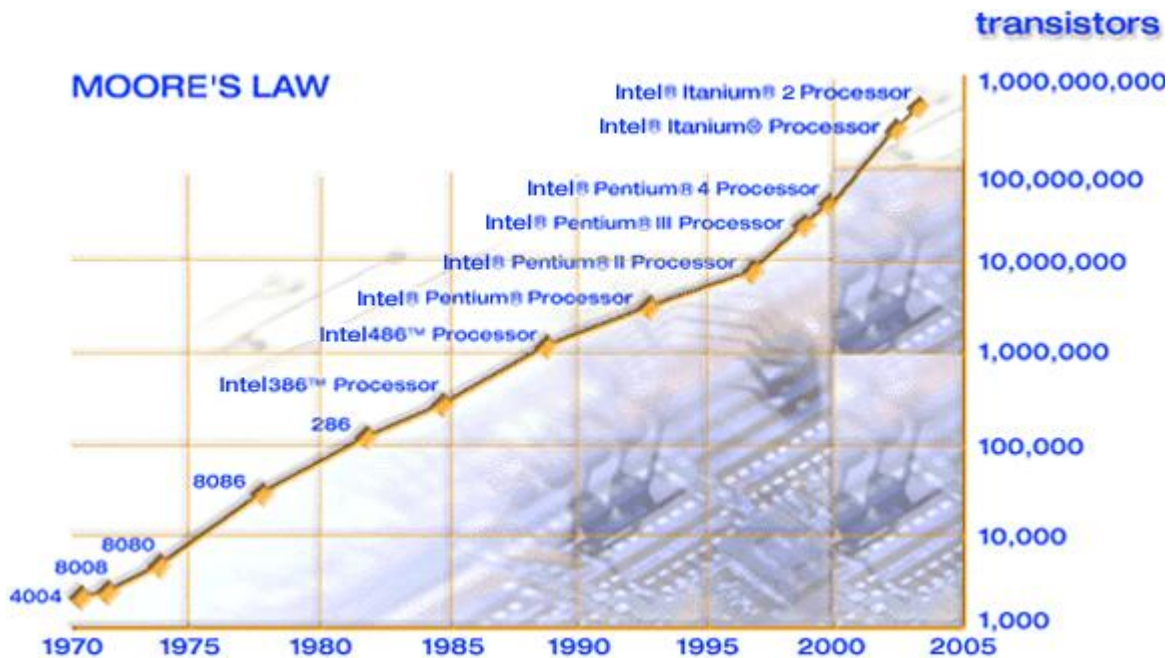
1979		8088	5, 8MHZ	20 bit		29.000
1982		80186(IAPX186)	10,12MHZ		1MB	
		80286(286)	5, 6, 8, 10, 12, 16, 20, 25MHZ	24 bit	16MB	134.000
1985	32	386DX	16, 20, 25,33MHZ	32 bit	4GB	275.000
1988		386SX	16, 20, 25,33MHZ	24 bit	16MB	275.000
1990		386SL	16, 20, 25MHZ	32 bit	4GB	855.000
1989		486DX	25, 35, 50MHZ	32 bit	4GB	1,2 triệu
1991		486SX	16, 20, 25, 3MHZ	32 bit	4GB	1,2 triệu
1992		486SL	20, 25, 33MHZ	32 bit	4GB	1,4 triệu
1993		Pentium	75, 90, 100, 120MHZ	32 bit	4GB	3,1 triệu
1995		Pentium MMX	166, 200, 233MHZ			4,5 triệu
1996		Pentium Pro	150, 166,180, 200MHZ		4GB	5,5 triệu
1997		Pentium II-Klamath	233, 266, 300MHZ			7,5 triệu
		Pentium II-Deschutes	333, 350, 400, 450MHZ			7,5 triệu
1998		Celeron-Covington	266, 300MHZ			7,5 triệu
		Celeron-Mendocino	300, 333, 366, 400, 433, 466, 500, 533MHZ			19 triệu
1999		Pentium III-Katmai	450, 500, 550, 533, 600MHZ			19,5 triệu
		Pentium III-Coppermine	500,550,600,650,70 0,750,800, 850MHZ(Bus 100MHZ); 533,600,667,733,80 0,866,933,1000,110 0,1133MHZ,(133M)			28,1 triệu
		Pentium III-Tualatin	1133, 1200, 1266, 1333, 2900MHZ			28,1 triệu
2000		Celeron-Coppermine	533, 566, 600, 633, 667, 700, 733, 766, 800MHZ (66MHZ); 850, 900, 950,			28,1 triệu

			1000, 1200, 1300MHz (1000MHz)			
		Celeron-Tualatin	1.0, 1.1, 1.2, 1.3, 2.9GHz			28,1 triệu
		Pentium IV-Willamette	1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0GHz			42 triệu
		511	2.8GHz			
		515	2.93GHz			
		515J	2.93GHz			
		516	2.93GHz			
		519J	3.06GHz			
		519K	3.06GHz			
2002		Celeron Willamette 128	1.7, 1.8GHz			Hơn 42 triệu
		Pentium IV- Northwood	A 1.6, 1.8, 2.0, 2.2, 2.4, 2.5, 2.6, 2.8GHz			55 triệu
	B 2.26,2.4,2.53,2.66,2 .8,3.06GHz					
	C 2.4, 2.6, 2.8, 3.0, 3.2, 3.4GHz					
		505	2.66GHz			
		505J	2.66GHz			
		506	2.66GHz			
		Celeron Northwood 128(1.8A>2.8)	1.8, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8GHz			Hơn 55 triệu
		Celeron D	2.13->3.33GHz			Hơn 125 triệu
2003		P4 Extreme Edition	3.2, 3.4GHz			
2004	64	NetBurst 64bit	800MHz			
		P4HT 620	2.8GHz			
		P4HT 630	3.0GHz			

		P4HT 640	3.2GHz			
		P4HT 650	3.4GHz			Hơn 125 triệu
		P4HT 641	3.2GHz			
		Pentium D	800MHz			
	2005	Pentium D805, D820, D830, D840	2.66, 2.8, 3.0, 3.2GHz			230 triệu
		Pentium EE Em64T	3.2GHz			Hơn 230 triệu
	2006	Intel Core 2 Duo				291 triệu
		E6600	2.4GHz			167 triệu
		E6700	2.66GHz			
	2006	Core 2 Extreme				Hơn 167 triệu
		X6800, QX6700, Q6300, Q6400, Q6600	2.93GHz			

### 2.1.2. Qui luật MOORE (MOORE'S LAW) về sự phát triển của máy tính

Sự tăng trưởng theo hàm số mũ của công nghệ chế tạo transistor MOS là nguồn gốc của thành quả các máy tính.



Hình 1.6 Sự phát triển của bộ xử lý Intel dựa vào số lượng transistor trong một mạch tích hợp theo qui luật Moore

Từ năm 1965, Gordon Moore (đồng sáng lập công ty Intel) quan sát và nhận thấy số transistor trong mỗi mạch tích hợp có thể tăng gấp đôi sau mỗi năm, G. Moore đã đưa ra dự đoán: ***Khả năng của máy tính sẽ tăng lên gấp đôi sau 18 tháng với giá thành là như nhau.***

Kết quả của quy luật Moore là:

- ⌚ Chi phí cho máy tính sẽ giảm.
- ⌚ Giảm kích thước các linh kiện, máy tính sẽ giảm kích thước
- ⌚ Hệ thống kết nối bên trong mạch ngắn: tăng độ tin cậy, tăng tốc độ .
- ⌚ Tiết kiệm năng lượng cung cấp, toả nhiệt thấp.
- ⌚ Các IC thay thế cho các linh kiện rời.

## CÂU HỎI ÔN TẬP CHƯƠNG I

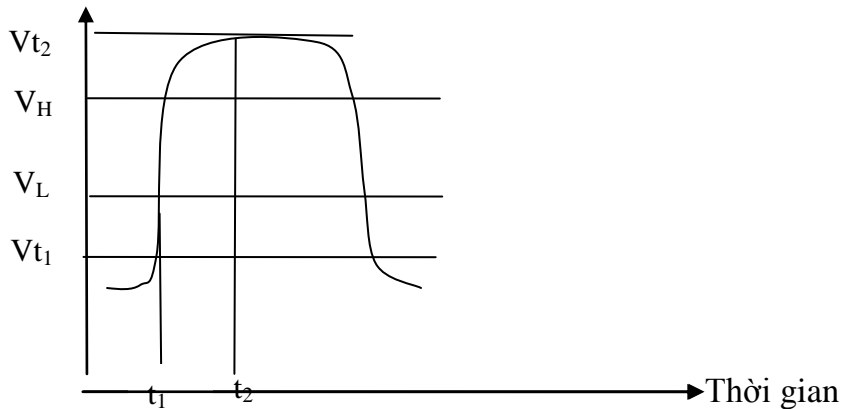
1. Trình bày cấu trúc và chức năng của máy tính
2. Dựa vào tiêu chuẩn nào người ta phân chia máy tính thành các thế hệ?
3. Đặc trưng cơ bản của mỗi thế hệ máy tính
4. Khuynh hướng phát triển của máy tính điện tử ngày nay là gì?
5. Việc phân loại máy tính dựa vào tiêu chuẩn nào?
6. Trình bày tóm tắt lịch sử phát triển kiến trúc bộ vi xử lý của hãng Intel.

## Chương 2: BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH

### 2.1 THÔNG TIN VÀ SỰ MÃ HOÁ THÔNG TIN

#### 2.1.1. Khái niệm thông tin

Hiệu điện thế



Hình 2.1: Thông tin về 2 trạng thái có ý nghĩa của hiệu điện thế

Khái niệm về thông tin gắn liền với sự hiểu biết một trạng thái cho sẵn trong nhiều trạng thái có thể có vào một thời điểm cho trước.

Trong hình này, chúng ta quy ước có hai trạng thái có ý nghĩa: trạng thái thấp khi hiệu điện thế thấp hơn  $V_L$  và trạng thái cao khi hiệu điện thế lớn hơn  $V_H$ . Để có thông tin, ta phải xác định thời điểm ta nhìn trạng thái của tín hiệu. Thí dụ, tại thời điểm  $t_1$  thì tín hiệu ở trạng thái thấp và tại thời điểm  $t_2$  thì tín hiệu ở trạng thái cao.

#### 2.1.2. Lượng thông tin và sự mã hoá thông tin

Thông tin được đo lường bằng đơn vị thông tin mà ta gọi là *bit*. Lượng thông tin được định nghĩa bởi công thức:

$$I = \text{Log}_2(N)$$

Trong đó: I: là lượng thông tin tính bằng bit

N: là số trạng thái có thể có

Vậy một bit ứng với sự hiểu biết của một trạng thái trong hai trạng thái có thể có.

Thí dụ, sự hiểu biết của một trạng thái trong 8 trạng thái có thể ứng với một lượng thông tin là:  $I = \text{Log}_2(8) = 3$  bit

Tám trạng thái được ghi nhận nhờ 3 số nhị phân (mỗi số nhị phân có thể có giá trị 0 hoặc 1).

Như vậy lượng thông tin là số con số nhị phân cần thiết để biểu diễn số trạng thái có thể có. Do vậy, một con số nhị phân được gọi là một bit. Một từ n bit có thể tượng trưng một trạng thái trong tổng số  $2^n$  trạng thái mà từ đó có thể tượng trưng. Vậy một từ n bit tương ứng với một lượng thông tin n bit.

## 2.2 HỆ ĐẾM

Hệ đếm là tập hợp các ký hiệu và một số qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị số.

Mỗi hệ đếm có một số ký số (digits) hữu hạn. Tổng số ký số của mỗi hệ đếm được gọi là cơ số (base hay radix), ký hiệu là b.

Hệ đếm cơ số b (với là số nguyên dương và  $b \geq 2$ ) mang tính chất sau:

- Có b ký số để thể hiện giá trị số. Ký số nhỏ nhất là 0 và ký số lớn nhất là b-1.
- Giá trị vị trí thứ n trong một số bằng cơ số b lũy thừa n:  $b^n$

- Một số N trong hệ đếm cơ số b được thể hiện:  $N_{(b)} = a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$

Trong đó, số  $N_{(b)}$  có n+1 ký số ở phần nguyên và m ký số ở phần lẻ sẽ có giá trị là:

$$N_{(b)} = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m} \cdot b^{-m} \quad \text{hay} \quad N_{(b)} = \sum_{i=-m}^n a_i \cdot b^i$$

### 2.2.1. Hệ đếm thập phân (decimal, b=10) và Hệ nhị phân (binary, b=2).

Trong hệ thập phân người ta sử dụng 10 ký hiệu cơ bản 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 để biểu diễn các giá trị số  $a_i$  khác nhau. Một số lớn hơn 9 được biểu diễn thông qua tổ hợp của các ký hiệu cơ bản đó.

Ví dụ: Một số tự nhiên  $1975 = 1 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 5 \times 10^0$

Hệ nhị phân sử dụng hai ký hiệu 0 và 1 để biểu diễn các giá trị số  $a_i$ ,

Ví dụ: Số nhị phân 10111 được tính bằng  $1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 (= 23$  trong hệ thập phân)

### 2.2.2. Chuyển đổi giá trị giữa hệ nhị phân và hệ thập phân.

+ *Hệ thập phân sang nhị phân:*

Để biến đổi một số tự nhiên từ hệ thập phân sang hệ nhị phân, ta lấy số thập phân đã cho chia 2, số dư của phép chia này được dùng làm bit bé nhất (LSB) của số nhị phân. Thương của phép chia này lại được chia tiếp cho 2 và số dư của phép chia lại được dùng



làm bit tiếp theo của số nhị phân. Quá trình cứ tiếp tục cho đến khi thương của phép chia bằng 0 thì thôi

Ví dụ: Biến đổi số thập phân 41 sang hệ nhị phân,

41 →

$41/2=20$ dư 1	↑ Các số dư được sắp xếp theo trật tự này để tạo nên số nhị phân
$20/2=10$ dư 0	
$10/2=5$ dư 0	
$5/2 = 2$ dư 1	
$2/2 = 1$ dư 0	
$1/2 = 0$ dư 1	
Kết thúc	
$(41)_{10} = (1\ 0\ 1\ 0\ 0\ 1)_2$	

+ *Chuyển đổi số thập phân lẻ sang nhị phân:* Trước tiên chuyển phần lẻ. Để chuyển đổi ta nhân phần lẻ của số thập phân với 2, phần nguyên nhận được mà chỉ có thể là 1 hoặc 0 là bit MSB của phần lẻ của số nhị phân mong muốn. Tiếp theo nhân phần lẻ mới của số thập phân để tìm bit nhị phân kế và tiếp tục như vậy cho đến khi phần lẻ thập phân hết (trở thành 000).

Ví dụ số thập phân 0.6875:

$$0.6875 \times 2 = 1.3750 \text{ bit nhị phân là } 1 \text{ (MSB)}$$

$$0.3750 \times 2 = 0.7500 \text{ bit nhị phân là } 0$$

$$0.7500 \times 2 = 1.5000 \text{ bit nhị phân là } 1$$

$$0.5000 \times 2 = 1.0000 \text{ bit nhị phân là } 1 \text{ (LSB)}$$

Kết quả là  $0.6875_{10} = 0.1011$

Vì bit đầu tiên là MSB nên các bit càng về sau có nghĩa càng thấp, tức càng có giá trị nhỏ nên trong trường hợp vài phép nhân 2 đầu không dẫn đến phần lẻ thập phân là 0 thì ta vẫn có thể dừng hay tiếp tục cho đến khi đủ số lẻ nhị phân cho sự chính xác cần thiết.

Khi số thập phân gồm phần nguyên và phần thập phân, ta chuyển đổi hai phần riêng biệt rồi kết hợp lại.

Ví dụ:

Đã biết  $41_{10} = 101001$

Và  $0.6875_{10} = 0.1011$

Nên  $41.6875_{10} = 101001.1011$

+ *Hệ nhị phân sang thập phân:*

Để biến đổi một số từ hệ nhị phân sang hệ thập phân ta dùng ngay công thức tính đã dẫn ở trên, tức là:

$$N = b_n b_{n-1} \dots b_0 = b_n \times 2^n + b_{n-1} \times 2^{n-1} + \dots + b_0 \times 2^0$$

Ví dụ: số nhị phân  $101001 = 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^0 (=41 \text{ ở hệ thập phân})$

+ *Số nhị phân biểu thị số lẻ*

Số lẻ (số thực) là số có phần nguyên (integer) và phần phân số (fraction) mà được viết gần cách nhau bởi dấu chấm (dấu chấm được dùng thay dấu phẩy) gọi dấu chấm thập phân ở hệ thập phân. Ví dụ 41.6875. tương tự, ở hệ nhị phân người ta dùng dấu chấm nhị phân để ngăn cách phần nguyên và phần phân số. Ví dụ 1101.101.

Cách viết số lẻ thập phân như ví dụ sau:

$$\begin{aligned} 725.475 &= 7 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2} + 5 \times 10^{-3} \\ &= 7 \times 100 + 2 \times 10 + 5 \times 1 + 4 \times 0.1 + 7 \times 0.01 + 5 \times 0.001 \\ &= 700 + 20 + 5 + 0.4 + 0.07 + 0.005 \end{aligned}$$

Tương tự, cách viết số lẻ ở hệ nhị phân có nghĩa như qua ví dụ:

$$\begin{aligned} 1101.101 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 4 + 1 + 0.5 + 0.125 \\ &= 13.625 \end{aligned}$$

### 2.2.3. Các hệ đếm bát phân (b=8) và thập lục phân (b=16)

Việc chuyển đổi giá trị giữa hệ nhị phân và thập phân dẫu sao cũng có những nhược điểm nhất định. Do vậy, để khắc phục những nhược điểm đó người ta sử dụng các hệ đếm trung gian là hệ 8 (Octal) và hệ 16 (Hexadecimal) là những hệ đếm rất thuận tiện khi chuyển đổi giá trị với hệ 2 (nhị phân).

1. *Hệ bát phân (b=8):* Một số tự nhiên N trong hệ 8 có thể được viết và tính như sau:

$$N = o_n o_{n-1} \dots o_0 = o_n \times 8^{n-1} + o_{n-1} \times 8^{n-2} + \dots + o_0 \times 8^0$$

Trong đó, mỗi chữ số  $o_i$  có thể nhận 1 trong 8 giá trị ( $o_i = \{0, 1, 2, \dots, 7\}$ ).

Với quy ước như vậy, việc chuyển đổi giá trị giữa hệ 2 và hệ 8 là rất đơn giản để chuyển đổi một giá trị từ hệ 2 sang hệ 8 ta chỉ việc thay một nhóm 3 bit bằng một chữ số hệ 8, ngược lại để chuyển một giá trị của số từ hệ 8 sang hệ 2 ta chỉ việc thay một chữ số hệ 8 bằng một nhóm 3 bit.

Ví dụ:  $10101011_2$  (hệ 2) =  $253_8$  (hệ 8)

2. *Hệ thập lục phân* gọi tắt là hệ 16 hoặc hex ( hoặc HEX) hay hexa (hoặc HEXA): một số tự nhiên N trong hệ 16, có thể được viết và tính như sau:

$$N = h_n h_{n-1} \dots h_0 = h_n \times 16^n + h_{n-1} \times 16^{n-1} + \dots + h_0 \times 16^0$$

Trong đó, mỗi chữ số  $h_i$  có thể nhận 1 trong 16 giá trị ( $h_i = \{0, 1, \dots, 9, A, B, C, D, E, F\}$ ).

Cũng như đối với hệ 8, để chuyển đổi qua lại một giá trị giữa hệ 2 và hệ 16 ta chỉ cần thay thế qua lại giữa một chữ số thuộc hệ 16 bằng 1 nhóm 4 bit .

Ví dụ:  $10101011_B$  (hệ 2) =  $253_O$  (hệ 8) =  $AB_H$  (hệ 16)

#### 2.2.4. Mã BCD (Binary Coded Decimal)

Nhiều khi để chuyển giá trị thập phân sang dạng nhị phân một cách nhanh chóng ta sử dụng một cách biểu diễn nhị phân khác gọi là mã BCD. Mã BCD thay thế một chữ số thập phân bằng một nhóm 4 bit nhị phân. Với việc sử dụng mã BCD, phép chuyển giá trị của số từ hệ thập phân sang hệ nhị phân trở nên đơn giản hơn nhiều.

Ví dụ: số thập phân 41, chuyển sang mã BCD sẽ là:  $\underbrace{0100}_4 \underbrace{0001}_1$

Cần lưu ý rằng, một nhóm 4 bit có thể biểu diễn được 16 tổ hợp khác nhau, mã BCD chỉ sử dụng 10 tổ hợp đầu ứng với các giá trị thập phân từ 0 ( $0000_2$ ) đến 9 ( $1001_2$ ), 6 tổ hợp đầu ứng với các giá trị từ 10 ( $1010_2$ ) đến 15 ( $1111_2$ ) không dùng. Điều này có nghĩa là phép ánh xạ giữa hệ đếm thập phân và mã BCD không phải là 1-1, do vậy cần phải có sự chú ý để điều chỉnh khi thực hiện các phép tính số học đối với số được biểu diễn theo mã BCD.

Đơn vị thông tin cơ bản dùng trong máy tính là byte (8 bit). Do đó, khi biểu diễn mã BCD trong máy tính có thể có hai khả năng:

- Sử dụng một byte để biểu diễn một chữ số thập phân (thực chất là chỉ dùng 4 bit thấp, 4 bit cao nhận giá trị 0), cách biểu diễn này gọi là unpacked BCD.

- Một byte biểu diễn được hai chữ số thập phân (4 bit thấp biểu diễn một chữ số thập phân, và 4 bit cao biểu diễn một chữ số thập phân khác), cách biểu diễn này gọi là Packed BCD.

Bảng 2.1 liệt kê 16 giá trị tự nhiên đầu tiên được biểu diễn theo các hệ đếm khác nhau

Hệ 10	Hệ 2	Mã BCD	Hệ 16	Hệ 8
0	0000	0000	0	0
1	0001	0001	1	1
2	0010	0010	2	2
3	0011	0011	3	3
4	0100	0100	4	4
5	0101	0101	5	5
6	0110	0110	6	6
7	0111	0111	7	7
8	1000	1000	8	10
9	1001	1001	9	11
10	1010	0001 0000	A	12
11	1011	0001 0001	B	13
12	1100	0001 0010	C	14
13	1101	0001 0011	D	15
14	1110	0001 0100	E	16
15	1111	0001 0101	F	17

### 2.2.5. Chuyển đổi giữa hệ thập phân và hệ Hex.

+ Hệ thập phân sang hệ hex

Để chuyển đổi từ thập phân sang hệ hex, ta phải thuộc lòng các số hex tương ứng với số thập phân từ 0 đến 15. Với số thập phân lớn hơn 15 cách cơ bản là thực hiện chia số thập phân cho 16 để tìm thương và dư, dư là bit  $b_0$  (LSB) của số hex và thương là bit  $b_1$  của số hex.

Ví dụ chuyển đổi  $16_{10}$ ,  $20_{10}$  và  $254_{10}$  sang hex

$$16 : 16 \quad \text{vậy} \quad 16_{10} = 10_H$$

0            1  
 (b<sub>0</sub>)        (b<sub>1</sub>)

20    :    16    vậy    20<sub>10</sub> = 14<sub>H</sub>  
 4            1  
 (b<sub>0</sub>)        (b<sub>1</sub>)

254            :    16    vậy    254<sub>10</sub> = FE<sub>H</sub>  
 94            15  
 14            (F)  
 (E)            (b<sub>1</sub>)  
 (b<sub>0</sub>)

Trường hợp thương lớn hơn 15 ta chia cho 16 để tìm thương thứ hai mà là bit b<sub>2</sub> và dư thứ hai mà là bit b<sub>1</sub>, còn dư thứ nhất là bit b<sub>0</sub>.

Ví dụ chuyển đổi 1023<sub>10</sub> sang hex:

1023            :    16    vậy    1023<sub>10</sub> = 3FF<sub>H</sub>  
 063            63        :    16  
   15            15            3  
 (F)            (F)            (b<sub>2</sub>)  
 (b<sub>0</sub>)            (b<sub>1</sub>)

+ Hệ Hex sang hệ thập phân

Để tìm giá trị thập phân của một số hex thì hoặc là ta đổi hex thành số nhị phân (từng nhóm 4 bit) rồi tìm trị giá số nhị phân này. Cách thứ hai là chuyển đổi trực tiếp từ hex sang thập phân theo quy ước về giá trị tương tự như ở hệ thập phân, nhị phân.

Ví dụ:

$$\begin{aligned}
 8B5D_{\text{HEX}} &= 8 \times 16^3 + B \times 16^2 + 5 \times 16^1 + D \times 16^0 \\
 &= 8 \times 16^3 + 11 \times 16^2 + 5 \times 16^1 + 13 \times 16^0 \\
 &= 8 \times 4096 + 11 \times 256 + 5 \times 16 + 14 \times 1 \\
 &= 35677_{10}
 \end{aligned}$$

## 2.3 BIỂU DIỄN SỐ NGUYÊN VÀ SỐ THỰC TRONG HỆ NHỊ PHÂN

### 2.3.1. Số nguyên có dấu:

Trong các máy tính điện tử, một số nguyên có dấu có thể được biểu diễn bằng một trong các cách sau:

+ *Dấu và trị tuyệt đối:*

Trong số nhị phân  $n$  bit, MSB được dùng làm bit dấu (với quy ước 0 là số dương và 1 là số âm),  $n-1$  bit còn lại biểu diễn giá trị tuyệt đối.

Ví dụ: 0000 1111 và 1000 1111 là biểu diễn nhị phân là 8 bit của 15 và  $-15$ .

Số 0 có thể biểu diễn bằng 0000 0000 (+0) hoặc 1000 0000 (-0). Giá trị nguyên có thể biểu diễn được theo nguyên tắc này của một số nhị phân  $n$  bit là từ  $-2^{n-1}-1$  đến  $2^{n-1}-1$ , của một số 8 bit là từ  $-2^7-1$  đến  $2^7-1$ .

+ *Mã bù 1:*

Số đối của  $A$  có được bằng cách đảo giá trị tất cả các bit của  $A$ .

Ví dụ: 0000 1111 và 1111 0000 là hai số nhị phân 8 bit đối nhau (15 và  $-15$ ).

Có hai cách biểu diễn số 0 là 0000 0000 (+0) và 1111 1111 (-0). Dãy giá trị nguyên có thể biểu diễn được theo quy tắc bù 1 của một số nhị phân  $n$  bit là từ  $-2^{n-1}$  đến  $2^{n-1}-1$  của một số 8 bit là từ  $-2^7$  đến  $2^7-1$ .

+ *Mã bù 2:*

Là cách biểu diễn số thường hay được dùng nhất trong các phép tính số học đối với các số nguyên nhị phân. Mã bù 2 của một số được xác định bằng cách lấy mã bù 1 của nó cộng với 1.

Ví dụ: 0000 1111 và 1111 0001 là các số 15 và  $-15$  biểu diễn theo mã bù 2.

Biểu diễn bù 2 (8 bit) của số 0 vẫn là 0000 0000. Dãy giá trị nguyên có thể biểu diễn được theo quy tắc bù 2 của một số nhị phân  $n$  bit là từ  $-2^{n-1}$  đến  $2^{n-1}-1$ , của một số nhị phân 8 bit là từ  $-2^7$  đến  $2^7-1$ .

Số thập phân	Mã nhị phân		
	Dấu và trị tuyệt đối	Bù 1	Bù 2
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101

+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1000	1111	(0000)
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8			1000

**Bảng 2.2.** Các cách biểu diễn số nhị phân có dấu (4 bit)

### 2.3.2. Biểu diễn số thực

-Một số thực R có thể biểu diễn bằng một từ mã nhị phân trong máy tính như sau:

S	N	L
---	---	---

Trong đó :

S là phần thể hiện dấu gồm chỉ L bit;

N là phần nguyên gồm n bit ( $b_{n-1}b_{n-2}...b_0$ );

L là phần lẻ gồm L bit ( $b_{-1}b_{-2}...b_{-L}$ ).

Cách tính dấu thường hay dùng là *dấu và trị tuyệt đối hoặc bù 2*. Ví dụ, nếu sử dụng quy tắc *dấu và trị tuyệt đối* thì số thực R sẽ được tính theo quy tắc:

$$R = S N \cdot L = sb_{n-1}b_{n-2}...b_0b_{-1}b_{-2}...b_{-L}$$

$$= (-1)^s \times (b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + \dots + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots + b_{-L} \cdot 2^{-L})$$

Ở đây ta tưởng tượng có một dấu · Ngăn cách phần nguyên và phần lẻ của số thực R; mặc dù trên thực tế, không tồn tại dấu · Như vậy trong các máy tính điện tử. Cách biểu diễn số thực như vậy được gọi là *fixed-point* (dấu chấm động)

Ví dụ: từ mã nhị phân 8 bit 10110101, nếu hiểu là số thực *fixed-point* theo quy tắc và trị tuyệt đối gồm 5 bit nguyên và 2 bit lẻ, sẽ được tính thành:

$$R = (-1)^1 \times (1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}) = -13,25$$

-Số thực R cũng có thể được biểu diễn dưới một dạng khác, như sau:

S	E	M
---	---	---

Trong đó : s là phần thể hiện dấu gồm chỉ 1 bit;

E là phần bậc (mũ) gồm e bit ( $b_{-1}b_{-2}...b_{-e}$ );

M là phần định trị gồm m bit ( $b_{-1}b_{-2}...b_{-m}$ ).

Cách biểu diễn số thực theo kiểu này gọi là floating-point. Giá trị của số thực R dạng floating-point được tính như sau:  $R = (-1)^s \times M \times 2^E$

Có nhiều dạng biểu diễn floating-point khác nhau (IBM, VAX, IEEE...). Dạng biểu diễn nhị phân floating-point thường dùng nhất là chuẩn floating-point IEEE 754. Một số floating-point theo chuẩn IEEE754 được biểu diễn bằng một mã nhị phân 32 bit (single precision) hoặc 64 bit (double precision). Số thực floating-point single precision có độ dài 32 bit, đầu tiên là một bit dấu tiếp theo là 8 bit phần mũ và cuối cùng là 23 bit phần định trị. Số thực floating-point double precision có độ dài 64 bit, gồm một bit dấu tiếp theo là 11 bit phần mũ và cuối cùng là 52 bit phần định trị.

Single precision:

(32bit)

S	B <sub>7</sub> b <sub>6</sub> b <sub>5</sub> b <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>	B <sub>-1</sub> b <sub>-2</sub> b <sub>-3</sub> ... b <sub>-23</sub>
1 bit (dấu)	8 bit (mũ)	23 bit (định trị)

Double precision:

(64 bit)

S	B <sub>10</sub> b <sub>9</sub> b <sub>8</sub> .. B <sub>4</sub> b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>	B <sub>-1</sub> b <sub>-2</sub> b <sub>-3</sub> ... b <sub>-52</sub>
1 bit (dấu)	11 bit (mũ)	52 bit (định trị)

Ví dụ: Tính giá trị của số thực

floating-point 32 bit sau:

1	10000110	011001000100000000000000
---	----------	--------------------------

$$R = (-1)^1 \times (1+2^{-2}+2^{-3}+2^{-6}+2^{-10}) \times 2^{134-127}$$

$$= -1 \times (2^7+2^5+2^4+2^1+2^{-3})$$

$$= 128 + 32 + 16 + 2 + 0,125 = 178,125$$

## 2.4 CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN VÀ THẬP LỤC PHÂN

### 2.4.1. Cộng và trừ nhị phân

- *Phép cộng số học các số nhị phân được thực hiện theo quy tắc sau:*

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

$$1 + 1 = 10,$$

Trong kết quả này số 0 được gọi là tổng (sum) số 1 được gọi là nhớ (carry). Phép cộng số học các số nhị phân nhiều bit được thực hiện theo quy tắc đó, áp dụng từ phải sang trái, carry từ phép cộng bit thấp được cộng thêm vào phép cộng ở bit cao hơn, carry của bit cao nhất là carry của cả phép cộng.

Ví dụ:



$$\begin{array}{r}
 \text{a) } 1001\ 0110 \\
 +0101\ 1011 \\
 \hline
 1111\ 0001
 \end{array}
 \qquad
 \begin{array}{r}
 \text{b) } 1100\ 1001 \\
 +\ 0110\ 1101 \\
 \hline
 1\ 0011\ 0110
 \end{array}$$

→ Số 1 này là carry của phép cộng

Người ta cần phải căn cứ vào quy tắc mã hoá số để so cách thức xử lý kết quả thích hợp. Trong máy tính, bộ phận chứa kết quả cũng có độ dài (số bit) như là bộ phận chứa các toán hạng. Khi phép cộng cho kết quả có số bit vượt quá số bit của bộ phận thì kết quả của phép tính có thể không còn đúng nữa. Tuy nhiên, điều này lại còn phụ thuộc vào quy tắc mã hoá số.

Nếu các số hạng thực hiện phép cộng được mã hoá theo luật nhị phân tự nhiên thì số nhớ ra khỏi MSB (gọi là carry) báo hiệu có vấn đề ở kết quả tính. Xét hai ví dụ trên, ta thấy ở ví dụ a) phép cộng không có carry cho thấy kết quả của phép tính là bình thường ( $118 + 91 = 209$ ), còn ở ví dụ b) carry của phép cộng cho thấy kết quả phép tính là không bình thường ( $201 + 109 = 54$  ?).

Nếu các số hạng thực hiện phép cộng được mã hoá theo luật của số nguyên (ví dụ mã bù 2) thì ngay cả khi không có carry, kết quả của phép tính cũng là không đáng tin nếu như có nhớ ở bit cao nhất của phần giá trị. Lại xét hai ví dụ trên, ta thấy ở ví dụ a) mặc dù không có carry nhưng kết quả của phép tính là không đúng ( $118 + 91 = -47$ ?). Còn ở ví dụ b) mặc dù phép cộng có carry nhưng kết quả của phép tính lại đúng ( $-55 + 109 = 54$ ). Việc nhận xét kết quả phép cộng số nguyên phải dựa vào một dấu hiệu gọi là overflow, tức là trạng thái có một và chỉ một carry trong số hai bit tận cùng bên trái của kết quả (bit dấu và bit cao nhất của phần giá trị).

Nếu các số hạng được mã hóa theo BCD thì khi có nhớ ra khỏi một nhóm của một nhóm (một digit) hoặc giá trị một nhóm bốn bit lớn hơn 9 ta cần phải có sự hiệu chỉnh thích hợp, cụ thể là thêm 6 cho mỗi nhóm bốn bit đó.

- *Phép trừ các số nhị phân được thực hiện theo qui tắc sau:*

$$1 - 0 = 1 \qquad 0 - 0 = 1 - 1 = 0 \qquad 0 - 1 = 11,$$

trong kết quả (11) này ,số 1 bên phải được gọi là hiệu ,số 1 bên trái là nhớ vay của phép trừ (gọi là carry hoặc borrow).

Phép trừ các số nhị phân nhiều bit được thực hiện như sau :

$$\begin{array}{r}
 1001\ 0110 \\
 -0101\ 1011 \\
 \hline
 0011\ 1011
 \end{array}
 \qquad
 \begin{array}{r}
 0111\ 1001 \\
 -\ 1100\ 1101 \\
 \hline
 1\ 1010\ 1100
 \end{array}$$

→ Số 1 này là carry/borrow của phép trừ

Phép trừ số nhị phân cũng có thể được thực hiện thông qua phép cộng :  $A - B = A + (-B)$

Những lưu ý về tràn số và hiệu chỉnh kết quả phép tính đã trình bày ở phép cộng cũng được áp dụng đối với phép trừ.

### 2.4.2. Cộng và trừ số hex:

- *Cộng :*

Khi cộng hai con số thập phân thì nếu tổng lớn hơn 9 ta viết con số đơn vị và nhớ con số hàng chục lên hàng cao kế. Tương tự khi cộng hai con số thập lục phân, nếu tổng lớn hơn F (tức  $15_{10}$ ) ta viết con số đơn vị và nhớ con số hàng thập lục lên hàng cao kế.

Ví dụ:

$$\begin{array}{r}
 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad F \\
 + \quad + \quad + \quad + \quad + \quad + \quad + \\
 \hline
 1 \quad 2 \quad 7 \quad 8 \quad A \quad F \quad F \\
 \hline
 9 \quad A \quad F \quad 10 \quad 12 \quad 17 \quad 1E
 \end{array}$$

Trường hợp  $8 + 8 = 16_{10}$  ta viết  $16-16 = 0$  và nhớ 1

Trường hợp  $8 + A = 18_{10}$  ta viết  $18-16 = 2$  và nhớ 1

Trường hợp  $8 + F = 23_{10}$  ta viết  $23-16 = 7$  và nhớ 1

Cùng qui luật trên áp dụng khi cộng hai số hex có nhiều con số, dĩ nhiên số nhớ cho hàng nào thì phải được cộng thêm vào cho hàng đó:

Ví dụ:

$$\begin{array}{r}
 18 \quad 88 \quad AA \quad 3CA \quad FFFF \\
 + \quad + \quad + \quad + \quad + \\
 \hline
 8 \quad 88 \quad EE \quad 5F7 \quad FFF \\
 \hline
 20 \quad 110 \quad 198 \quad 9C1 \quad 10FFE
 \end{array}$$

- *Trừ*

Khi trừ hai số hex, nếu số trừ lớn hơn số bị trừ, ta mượn  $16_{10}$  để thêm vào số bị trừ và trả  $1_{16}$  cho số trừ của hàng cao kế.

Ví dụ:

$$\begin{array}{r}
 8 \quad F \quad 5A \quad FE \quad 10FFE \\
 - \quad - \quad - \quad - \quad + \\
 \hline
 2 \quad A \quad F \quad C2 \quad FFF \\
 \hline
 6 \quad 5 \quad 4B \quad 3C \quad FFFF
 \end{array}$$

### 2.4.3. Cộng và trừ các số BCD

- Cộng hai số BCD có điểm khác so với cộng hai số nhị phân bình thường. Khi tổng số ở mỗi số hạng của số BCD bằng 9 (=1001) hay nhỏ hơn thì tổng số đó là kết quả cuối cùng.

Ví dụ

$$\begin{array}{r}
 0101 \text{ (5)} \\
 + \quad 0010 \text{ (2)} \\
 \hline
 0111 \text{ (7)}
 \end{array}
 \qquad
 \begin{array}{r}
 0101 \text{ (5)} \\
 + \quad 0100 \text{ (4)} \\
 \hline
 1001 \text{ (9)}
 \end{array}
 \qquad
 \begin{array}{r}
 0101 \ 0101 \text{ (55)} \\
 + \quad 0010 \ 0100 \text{ (24)} \\
 \hline
 0111 \ 1001 \text{ (79)}
 \end{array}$$

Khi tổng số hai số nhị phân lớn hơn 9 tức từ 1001 trở lên thì tổng số không phải là số BCD nên phải cộng tổng số với 6(0110) để có tổng số là 9 hoặc nhỏ hơn và tạo ra số nhớ 1 lên hàng BCD có nghĩa cao hơn.

Ví dụ.

$$\begin{array}{r}
 1000 \text{ (8)} \\
 + \\
 0111 \text{ (7)} \\
 \hline
 1111 \text{ không phải là số BCD} \\
 + \\
 0110 \text{ thêm 6} \\
 \hline
 \end{array}$$

0001 ← 0101 đúng BCD .

Lý do cộng thêm 6 là vì mã BCD không dùng 6 mã cao nhất của số nhị phân 4 bit, có thể cộng nhiều hàng

Ví dụ:

$$\begin{array}{r}
 0110 \ 0111 \text{ (67)} \\
 + \\
 0101 \ 1000 \text{ (58)} \\
 \hline
 1011 \ 1111 \\
 + \quad 0110 \quad \text{thêm 6 ở hàng đơn vị} \\
 \hline
 1100 \ 0101 \\
 + \quad 0110 \quad \text{thêm 6 ở hàng chục} \\
 \hline
 \end{array}$$

0001 ← 0010 0101 đúng BCD (125)

- Trừ BCD: cũng theo quy luật trừ nhị phân. Nếu số bị trừ nhỏ hơn số trừ thì phải mượn 1 ở hàng có nghĩa trên mà có giá trị là 10 nhị phân ở hàng đang bị trừ

(giống số mượn ở phép trừ thập phân). Để tiện việc sắp xếp ta chuyển 1 ở hàng có nghĩa kể trên xuống hàng đang bị trừ thành 1010 (=10 thập phân) rồi cộng vào với số bị trừ trước khi thực hiện phép trừ

Ví dụ:

$$\begin{array}{r}
 \text{(a) } 1001 \text{ (9)} \\
 - \quad 0100 \text{ (4)} \\
 \hline
 0101 \text{ (5)}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{(b) } 0111 \ 1001 \text{ (79)} \\
 - \quad 0101 \ 0101 \text{ (55)} \\
 \hline
 0010 \ 0100 \text{ (24)}
 \end{array}$$

$$\begin{array}{r}
 \text{(c) } 0011 \ 0101 \text{ (35)} \\
 - \quad 0001 \ 0111 \text{ (17)} \\
 \hline
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 0010 \ 1010 \text{ (2) (10)} \\
 + \quad 0101 \text{ (5)} \\
 \hline
 0010 \ 1111 \text{ (2) (15)} \\
 - \quad 0001 \ 0111 \text{ (1) (7)} \\
 \hline
 0001 \ 1000 \text{ (1) (8)}
 \end{array}$$

#### 2.4.4. Phép tính đối với số thực floating-point.

Khi thực hiện các phép toán số học với số thực floating-point, ta cần phải lưu ý phần mũ của chúng. Cụ thể, trước khi thực hiện các phép cộng/ trừ cần phải đưa về các toán hạng cùng bậc (có giá trị phần mũ như nhau) rồi mới thực hiện phép tính với phần định trị và giữ nguyên giá trị của phần mũ ở kết quả. Đối với các phép nhân (chia), ta thực hiện phép tính nhân (chia) đối với phần định trị và phép tính cộng(trừ) đối với mũ.

#### 2.4.5. Biểu diễn các ký tự

Tùy theo các hệ thống khác nhau, có thể sử dụng các bảng mã khác nhau: ASCII, EBCDIC, UNICODE.

##### + Mã ASCII

Mã ASCII (American standard code for information interchange) do Viện tiêu chuẩn quốc gia hoa kỳ (ANSI) giới thiệu năm 1986 . Đó là một loại mã 7 bit được sử dụng để mã hóa tất cả các chữ cái, chữ số, các ký tự thông dụng thường gặp trên hầu hết các ký hiệu đặc biệt (không in ra được ) dùng trong việc điều khiển truyền tin. Mã ASCII được chấp nhận sử dụng rộng rãi để biểu diễn thông tin trong hầu hết các hệ thống máy tính trên khắp thế giới. Mã ASCII chỉ sử dụng 7 bit (00 ÷ 7F<sub>H</sub>, biểu biểu diễn được 128 ký hiệu khác nhau) nhưng thường được viết dưới dạng 1 byte (đủ 8 bit ), với bit cao nhất là b<sub>7</sub> = 0. Bit b<sub>7</sub> đôi khi còn được dùng để tạo bit parity (phát hiện lỗi), nó cũng có thể được dùng để mã ASCII (cụ thể là biểu diễn các ký tự không có trong bảng chữ cái la-tinh).

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	^	3C	<	4C	L	5C	\	6C	l	7C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

NUL	Null	FF	Form feed	CAN	Cancel
SOH	Start of heading	CR	Carriage return	EM	End of medium
STX	Start of text	SO	Shift out	SUB	Substitute
ETX	End of text	SI	Shift in	ESC	Escape
EOT	End of transmission	DLE	Data link escape	FS	File separator
ENQ	Enquiry	DC1	Device control 1	GS	Group separator
ACK	Acknowledge	DC2	Device control 2	RS	Record separator
BEL	Bell	DC3	Device control 3	US	Unit separator
BS	Backspace	DC4	Device control 4	SP	Space
HT	Horizontal tab	NAK	Negative acknowledge	DEL	Delete
LF	Line feed	SYN	Synchronous idle		
VT	Vertical tab	ETB	End of transmission block		

### + Unicode

Với sự phổ biến ngày càng rộng rãi của máy tính, nhu cầu biểu diễn được tất cả các hệ thống ký tự của các quốc gia khác nhau là một đòi hỏi tất yếu. Để đáp ứng yêu cầu đó người ta đưa ra chuẩn Unicode, chuẩn này đề nghị sử dụng 2 byte (16 bit) để mã hoá một ký tự, với 16 bit, ta có  $2^{16} = 65536$  tổ hợp mã khác nhau để biểu diễn các ký tự cần thiết, gồm các chữ cái của tất cả các quốc gia dân tộc trên thế giới, các ký hiệu khoa học kỹ thuật và các chữ tượng hình trong hệ thống chữ Hán...

Như vậy, trong bộ nhớ của máy tính, một từ mã nhị phân có thể biểu diễn một mã lệnh của bộ vi xử lý, mã nhị phân của một ký tự, hoặc một toán hạng số với các kiểu (type) khác nhau. Bộ vi xử lý phân biệt đối tượng thể hiện của mỗi từ mã nhị phân nhờ vào logic của chương trình (phần mềm).

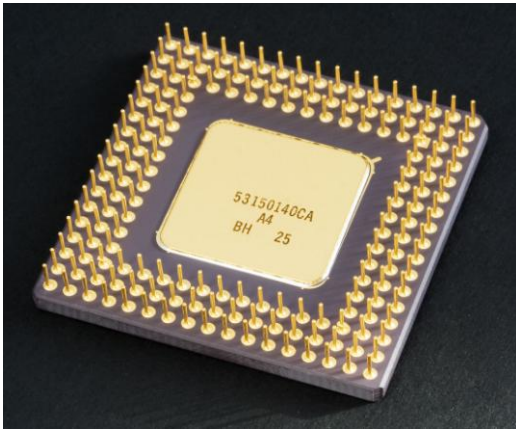


## Chương 3: BỘ XỬ LÝ TRUNG TÂM (CPU)

### 3.1. MÔ TẢ:

CPU là từ viết tắt của chữ Central Processing Unit (đơn vị xử lý trung tâm). CPU có thể được xem như não bộ của máy tính. CPU đảm nhận thực hiện chức năng cơ bản của máy tính là thực thi chương trình.

Chương trình được thực thi gồm một dãy các chỉ thị được lưu trữ trong bộ nhớ. Quá trình thực thi chương trình gồm hai bước: CPU đọc chỉ thị từ bộ nhớ và thực thi chỉ thị đó. Việc thực thi chương trình là sự lặp đi lặp lại quá trình lấy chỉ thị và thực thi chỉ thị. CPU là một mạch tích hợp phức tạp gồm hàng triệu transistor trên một bảng mạch nhỏ.



CPU có nhiều kiểu dáng khác nhau. Ở hình thức đơn giản nhất, CPU là một con chip với vài chục chân. Phức tạp hơn, CPU được ráp sẵn trong các bộ mạch với hàng trăm con chip khác. Tốc độ xử lý của máy tính phụ thuộc vào tần số đồng hồ làm việc của CPU (tính bằng MHz, GHz, ...) nhưng nó cũng phụ thuộc vào các phần khác như bộ nhớ đệm, RAM hay bo mạch đồ họa.

Hình 3.1 Bộ vi xử lý Intel 80486DX2 trong **PGA** bằng **đồ gốm**



Hai nhà sản xuất CPU lớn hiện nay là Intel và AMD. CPU của AMD thường có giá rẻ hơn Intel để thu hút khách hàng nhưng thị phần của AMD vẫn thấp hơn nhiều so với Intel.

#### Các thông số của CPU

- *Kiểu CPU*: CPU đời sau luôn có công nghệ và hiệu năng cao hơn CPU đời trước.
- *Tần số đồng hồ làm việc (tốc độ)*: Đối với các CPU cùng loại, tần số này càng cao thì tốc độ xử lý càng tăng. Đối với CPU khác loại, thì điều này chưa chắc đã đúng, ví dụ

CPU 486 tần số 20MHz có thể xử lý dữ liệu nhanh hơn CPU 386 33MHz, cũng như không thể so sánh tần số của CPU một nhân với CPU hai nhân.

- *Bộ nhớ đệm (cache)*: Bộ nhớ đệm dùng để lưu các lệnh hay dùng, giúp cho việc nhập dữ liệu xử lý nhanh hơn. Dung lượng bộ nhớ đệm càng nhiều càng giúp CPU làm việc nhanh hơn. Bộ nhớ đệm tích hợp vào CPU có hiệu quả cao hơn bộ nhớ đệm nằm rời bên ngoài.

- *Socket*: Chỉ loại khe cắm của CPU. Đây là đặc điểm để xét sự tương thích giữa CPU và bo mạch chủ.

- *Tốc độ FSB (Front Side Bus)*: Là kênh truyền dữ liệu giữa CPU và bộ nhớ trên Mainboard. Nó còn được gọi là System Bus (kênh truyền hệ thống). Tốc độ này càng cao hệ thống chạy càng nhanh.

Bộ xử lý trung tâm bao gồm:

### 3.1.1. Bộ điều khiển (CU-Control Unit)

Bộ điều khiển tạo các tín hiệu điều khiển di chuyển số liệu (tín hiệu di chuyển số liệu từ các thanh ghi đến bus hoặc tín hiệu viết vào các thanh ghi), điều khiển các tác vụ mà các bộ phận chức năng phải làm (điều khiển ALU, điều khiển đọc và viết vào bộ nhớ trong...).

Bộ điều khiển cũng tạo các tín hiệu giúp các lệnh được thực hiện một cách tuần tự.

Việc cài đặt bộ điều khiển có thể dùng một trong hai cách sau: dùng mạch điện tử hoặc dùng vi chương trình (microprogram).

Các thành phần của bộ điều khiển là :

- *Bộ đếm chương trình (Program Counter : PC)* : Đây là một số ghi chứa địa chỉ ô nhớ. Ô nhớ này chứa lệnh cần tìm để thi hành.
- *Thanh ghi lệnh (Instruction Register :IR)* : CPU tìm lệnh có địa chỉ nằm trong bộ đếm chương trình và đặt lệnh này vào thanh ghi lệnh.
- *Bộ giải mã lệnh (Instruction Decoder)* : Xác định các tác vụ của lệnh nằm trong thanh ghi lệnh,
- *Bộ tạo tín hiệu điều khiển (Squencer)* : Tạo ra các tín hiệu điều khiển cần thiết để thi hành các tác vụ mà bộ giải mã vừa xác định.
- *Tín hiệu đồng hồ (Clock)* : Là tín hiệu điện tử tuần hoàn làm đồng bộ các hoạt động của CPU.



### 3.1.2. Các thanh ghi (Registers)

Thanh ghi là phần tử nhớ nhanh tạm thời các dữ liệu và địa chỉ phục vụ cho hoạt động ở thời điểm hiện tại của bộ vi xử lý. Số lượng các thanh ghi quyết định đến hiệu năng xử lý của bộ vi xử lý.

Các thanh ghi được phân loại theo chức năng của chúng:

- Thanh ghi đa năng (General Register): Chứa các dữ liệu hoặc địa chỉ.

- Thanh ghi đoạn (Segment Registers): Trong không gian bộ nhớ được chia thành nhiều vùng khác nhau để: chứa mã chương trình, chứa dữ liệu, kết quả trung gian của chương trình và tạo ra một vùng nhớ đặc biệt được gọi là ngăn xếp (Stack) dùng vào việc quản lý các thông số của bộ vi xử lý khi gọi chương trình con hoặc trở về từ chương trình con. Các vùng này được gọi là thanh ghi đoạn gồm:

- Thanh ghi con trỏ:

- + Thanh ghi con trỏ lệnh IP (Instruction Pointer): Dùng để ghi địa chỉ của lệnh tiếp theo sẽ được nhận vào. Sau khi một lệnh được nhận vào thì nội dung trong mạch đếm chương trình PC (Program Counter) tự động tăng để trỏ sang lệnh tiếp theo.

- + Thanh ghi con trỏ cơ sở BP (Base Pointer): luôn trỏ vào dữ liệu nằm trong đoạn ngăn xếp.

- + Thanh ghi con trỏ trong ngăn xếp SP (Stack Pointer)

Ngăn xếp (Stack) là vùng nhớ, thường sử dụng cho chương trình con. Đáy ngăn xếp là một ngăn nhớ xác định, đỉnh ngăn xếp chứa thông tin (đỉnh ngăn xếp có thể bị thay đổi).

Như vậy, con trỏ ngăn xếp SP chứa địa chỉ của ngăn nhớ đỉnh ngăn xếp, khi cất thêm thông tin vào ngăn xếp thì nội của con trỏ ngăn xếp tự động giảm, con trỏ ngăn xếp trỏ vào đáy.

- Thanh ghi chỉ số: chứa độ lệch địa chỉ giữa ngăn nhớ mà bộ vi xử lý cần truy cập so với ngăn nhớ cơ sở.

- Thanh ghi cờ FR (Flag Register): dùng để ghi trạng thái kết quả của bộ vi xử lý, gồm có:

- + Cờ trạng thái: phản ánh kết quả của một lệnh mà bộ xử lý thực hiện.

+ Cờ điều khiển (Control Flag): cho phép hoặc không cho phép một thao tác nào đó của bộ vi xử lí.

### 3.1.3. Bộ số học-logic (ALU-Arithmetic Logic Unit)

Có chức năng thực hiện các lệnh của đơn vị điều khiển và xử lý tín hiệu. Theo tên gọi, đơn vị này dùng để thực hiện các phép tính số học (+,-,\*,/) hay các phép tính logic (so sánh lớn hơn, nhỏ hơn...)

## 3.2. DIỄN TIẾN THI HÀNH LỆNH MÃ MÁY

Việc thi hành một lệnh mã máy có thể chia thành 5 giai đoạn:

- *Đọc lệnh* (IF: Instruction Fetch)
- *Giải mã lệnh*(ID: Instruction Decode)
- *Thi hành lệnh*(EX: Execute)
- *Thâm nhập bộ nhớ trong hoặc nhảy*(MEM: Memmory)
- *Lưu trữ kết quả*(RS: Result Storing).

Mỗi giai đoạn được thi hành trong một hoặc nhiều chu kỳ xung nhịp.

### 3.2.1. Đọc lệnh:

$MAR \leftarrow PC$

$IR \leftarrow M[MAR]$

Bộ đếm chương trình PC được đưa vào thanh ghi địa chỉ bộ nhớ MAR (Memmmory Address Register). Lệnh được đọc từ bộ nhớ trong, tại các ô nhớ có địa chỉ nằm trong MAR và được đưa vào thanh ghi lệnh IR.

### 3.2.2. 2. Giải mã lệnh và đọc các thanh ghi nguồn:

$A \leftarrow Rs1$

$B \leftarrow Rs2$

$PC \leftarrow PC + 1$

Lệnh được giải mã. Kế đó các thanh ghi Rs1 và Rs2 được đưa vào A và B. Thanh ghi PC được tăng lên để chỉ tới lệnh kế đó.

### 3.2.3. Thi hành lệnh:

Tùy theo loại lệnh mà một trong ba nhiệm vụ sau đây được thực hiện:

+ Liên hệ tới bộ nhớ

MAR  $\leftarrow$  Địa chỉ do ALU tính tùy theo kiểu định vị (Rs2).

MBR  $\leftarrow$  Rs1

Địa chỉ hiệu dụng do ALU tính được đưa vào MAR và thanh ghi nguồn Rs1 được đưa vào MBR (Memory Buffer Register) để được lưu vào bộ nhớ trong.

+ Một lệnh của ALU

Ngã ra ALU  $\leftarrow$  Kết quả của phép tính

ALU thực hiện phép tính xác định trong mã lệnh, đưa kết quả ra ngã ra.

+ Một phép nhảy (thay đổi thứ tự thực thi của dãy lệnh)

Ngã ra ALU  $\leftarrow$  Địa chỉ lệnh tiếp theo do ALU tính.

ALU cộng địa chỉ của PC với độ dời để làm thành địa chỉ đích và đưa địa chỉ này ra ngã ra. Nếu là một phép nhảy có điều kiện thì thanh ghi trạng thái được đọc quyết định có cộng độ dời vào PC hay không.

#### 3.2.4. *Tham khảo bộ nhớ trong hoặc nhảy*

Giai đoạn này thường chỉ được dùng cho các lệnh nạp dữ liệu, lưu giữ dữ liệu và lệnh nhảy.

+ Tham khảo đến bộ nhớ:

MBR  $\leftarrow$  M[MAR] hoặc M[MAR]  $\leftarrow$  MBR

Số liệu được nạp vào MBR hoặc lưu vào địa chỉ mà MAR trỏ đến.

+ Nhảy:

If (điều kiện), PC  $\leftarrow$  ngã ra ALU

Nếu điều kiện đúng, ngã ra ALU được nạp vào PC. Đối với lệnh nhảy không điều kiện, ngã ra ALU luôn được nạp vào thanh ghi PC.

#### 3.2.5. *Lưu trữ kết quả:*

Rd  $\leftarrow$  Ngã ra ALU hoặc Rd  $\leftarrow$  MBR

Lưu trữ kết quả trong thanh ghi đích.

\* **Quá trình thực thi chương trình:** Xét trên quan điểm xử lý lệnh, 5 giai đoạn trên có thể mô tả đơn giản gồm 2 bước: CPU đọc lệnh từ bộ nhớ và thực thi lệnh đó. Việc thực thi chương trình là sự lặp đi lặp lại quá trình lấy lệnh và thực thi lệnh.

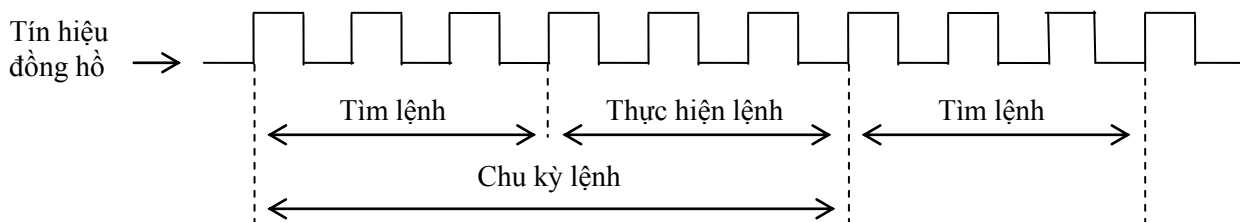
Sự thực thi chương trình bị kết thúc nếu máy bị tắt, có lỗi không thể phục hồi xuất hiện hay chương trình đi đến một lệnh kết thúc nó.

**\* Sự đồng bộ các tác vụ :**

Tín hiệu đồng hồ có nhiệm vụ làm đồng bộ đồng hồ và điều khiển các hoạt động của máy tính.

Tín hiệu này là tín hiệu tuần hoàn. Chu kỳ của nó được gọi là chu kỳ căn bản hay chu kỳ máy (clock cycle).

Một lệnh có thể được thực hiện trong một hay nhiều chu kỳ máy. Việc tìm lệnh cũng mất nhiều chu kỳ máy. Thời gian tìm lệnh và thực hiện lệnh được gọi là chu kỳ lệnh.



Chu kỳ lệnh ngắn nhất của một máy tính được gọi là chu kỳ CPU.

Ví dụ:

Một máy tính giả thiết có các đặc trưng sau:

Dạng lệnh 16 bit gồm  $2^4=16$  mã thao tác và  $2^{12}= 4096(4KB)$  word nhớ định vị địa chỉ trực tiếp

Dạng dữ liệu: 16 bit gồm 1 bit dấu và 15 bits dữ liệu.

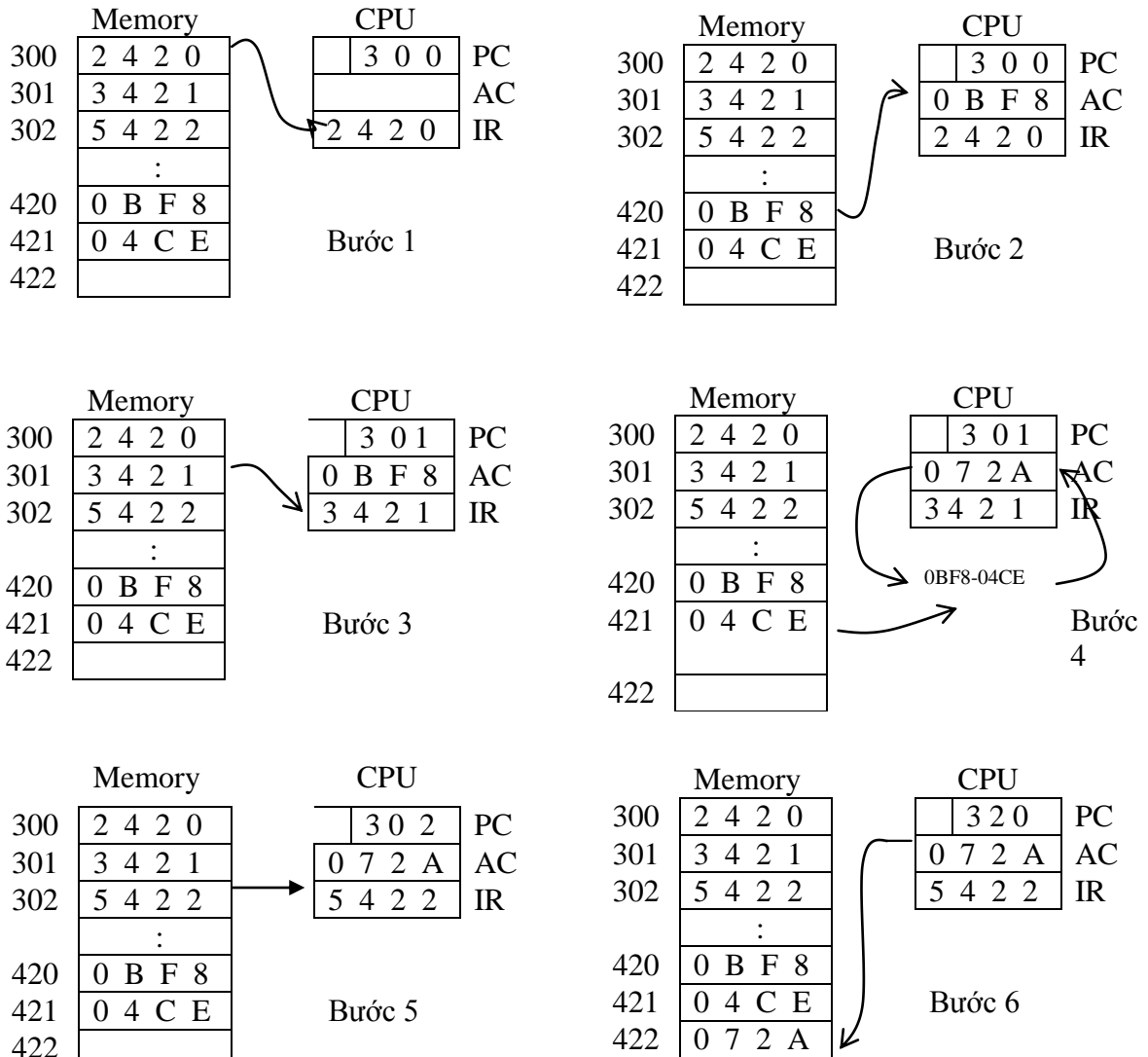
Các thanh ghi của CPU gồm: PC ( Program Counter) 12 bits, IR (Intruction Register) 16 bits và AC (Accumalator) 16 bits.

Mã lệnh được cho như sau:

- 0010: Load AC from memory
- 0101: Store AC to memory
- 0011: Sub to AC from memory

Mô tả sự thực thi một phần chương trình ở địa chỉ bắt đầu  $300_H$  để trừ giá trị tại ô nhớ  $420_H$  (có giá trị  $BF8_H$ ) cho nội dung tại ô nhớ  $421_H$  (có giá trị  $4CE_H$ ) và lưu vào ô nhớ  $422_H$  như sau. Ký hiệu được sử dụng ở dạng thập lục phân.

Sự thực thi đoạn chương trình đã cho được mô tả như sau:



Trong hình trên 3 lệnh được yêu cầu, được mô tả thành ba chu kỳ lấy và thực thi, gồm 6 bước như sau:

1. Bộ đếm chương trình (PC) chứa 300 là địa chỉ của lệnh đầu tiên. Địa chỉ này được tải vào thanh ghi lệnh, (IR). Chú ý rằng quá trình này sẽ bao gồm luôn việc dùng thanh ghi địa chỉ bộ nhớ và thanh ghi vùng nhớ đệm bộ nhớ (MBR), để đơn giản các thanh ghi trung gian này được bỏ qua.
2. 4 bit đầu tiên trong IR chỉ ra rằng bộ tích lũy (AC) tải vào, 12 bit còn lại đặc tả địa chỉ, ở đây là 420<sub>H</sub>.
3. PC được tăng lên một đơn vị và lệnh kế tiếp được lấy ra
4. Nội dung trước đó của AC được trừ cho nội dung tại vị trí 421<sub>H</sub>, kết quả được lưu trong AC.
5. PC được tăng lên một đơn vị và lệnh kế tiếp được lấy ra.

6. Nội dung AC được lưu ở vị trí 422<sub>H</sub>.

Trong ví dụ này, ba chu kỳ lệnh, mỗi chu kỳ gồm một chu kỳ lấy và thực thi được yêu cầu để trừ nội dung ở 420<sub>H</sub> cho nội dung tại 421<sub>H</sub>.

### 3.3. NGẮT QUÃNG (INTERRUPT)

Ngắt quãng là một sự kiện xảy ra một cách ngẫu nhiên trong máy tính và làm ngưng tính tuần tự của chương trình (nghĩa là tạo ra một lệnh nhảy). Phần lớn các nhà sản xuất máy tính (ví dụ như IBM, INTEL) dùng từ ngắt quãng để ám chỉ sự kiện này, tuy nhiên một số nhà sản xuất khác dùng từ “ngoại lệ”, “lỗi”, “bẫy” để chỉ định hiện tượng này.

Bộ điều khiển của CPU là bộ phận khó thực hiện nhất và ngắt quãng là phần khó thực hiện nhất trong bộ điều khiển. Để nhận biết được một ngắt quãng lúc đang thi hành một lệnh, ta phải biết điều chỉnh chu kỳ xung nhịp và điều này có thể ảnh hưởng đến hiệu quả của máy tính.

Người ta đã nghĩ ra “ngắt quãng” là để nhận biết các sai sót trong tính toán số học, và để ứng dụng cho những hiện tượng thời gian thực. Bây giờ, ngắt quãng được dùng cho các công việc sau đây:

- Ngoại vi đòi hỏi nhập hoặc xuất số liệu.
- Người lập trình muốn dùng dịch vụ của hệ điều hành.
- Cho một chương trình chạy từng lệnh.
- Làm điểm dừng của một chương trình.
- Báo tràn số liệu trong tính toán số học.
- Trang bộ nhớ thực sự không có trong bộ nhớ.
- Báo vi phạm vùng cấm của bộ nhớ.
- Báo dùng một lệnh không có trong tập lệnh.
- Báo phần cứng máy tính bị hư.
- Báo điện bị cắt.

Dù rằng ngắt quãng không xảy ra thường xuyên nhưng bộ xử lý phải được thiết kế sao cho có thể lưu giữ trạng thái của nó trước khi nhảy đi phục vụ ngắt quãng. Sau khi thực hiện xong chương trình phục vụ ngắt, bộ xử lý phải khôi phục trạng thái của nó để có thể tiếp tục công việc. Để đơn giản việc thiết kế, một vài bộ xử lý chỉ chấp nhận ngắt sau khi thực hiện xong lệnh đang chạy. Khi một ngắt xảy ra, bộ xử lý thi hành các bước sau đây:

- Thực hiện xong lệnh đang làm.
- Lưu trữ trạng thái hiện tại.
- Nhảy đến chương trình phục vụ ngắt
- Khi chương trình phục vụ chấm dứt, bộ xử lý khôi phục lại trạng thái cũ của nó và tiếp tục thực hiện chương trình mà nó đang thực hiện khi bị ngắt.

### 3.4. CÁC KIẾN TRÚC XỬ LÝ TỐC ĐỘ CỦA BỘ VI XỬ LÝ

#### 3.4.1. Kiến trúc tuần tự (Xử lý dạng đường ống)

Kiến trúc kiểu tuần tự: Đó là quá trình đọc, giải mã và thực hiện lệnh được thực hiện tuần tự, tức là lệnh này kết thúc thì mới đến lệnh khác được truyền vào bộ xử lý.

#### 3.4.2. Kiến trúc song song

Kiến trúc tuần tự chỉ thực hiện được một lệnh trong một chu kỳ nếu ghép nhiều bộ xử lý trong một bộ vi xử lý lớn, hoặc trong bộ xử lý có nhiều bộ xử lý số học và logic (ALU), thì có thể thực hiện được nhiều lệnh trong một chu kỳ. Các lệnh có thể thực hiện song song với nhau được tách ra từ các lệnh tổng hợp.

Có thể chia máy song song thành ba loại dựa vào số lượng lệnh và dòng dữ liệu.

SISD (Single Instruction Single Data): đơn dòng lệnh đơn dòng dữ liệu.

SIMD (Single Instruction Multiple Data): đơn dòng lệnh đa dòng dữ liệu.

MIMD (Complex Instruction Set Computer) là một máy tính được thiết kế với một tập lệnh đầy đủ.

#### 3.4.3. Kiến trúc CISC và RISC

+ CISC (Complex Instruction set computer): là một máy tính được thiết kế với một tập lệnh đầy đủ. Có từ đầu những năm 60, sử dụng vi chương trình (cài đặt lệnh của máy bằng cách viết một vi chương trình) tập lệnh phức tạp, dạng lệnh có chiều dài khác nhau và dạng thức phức tạp.

Các bộ xử lý của Intel từ 8088 đến 80486 đều dựa trên thiết kế này.

+ RISC (reduced Instruction Set Computer): là một máy tính được thiết kế với một tập lệnh rút gọn và thời gian thực hiện cố định, các lệnh có chiều dài cố định, có dạng đơn giản để giải mã thực hiện thanh ghi – thanh ghi, chỉ lệnh đọc hoặc ghi ô nhớ mới cho phép thâm nhập ô nhớ.

Thiết kế RISC có thể thấy trong các bộ xử lý mới như DEC Alpha, IBM và Intel Pentium.

### 3.5. CÁC KIỂU THI HÀNH MỘT LỆNH

Như đã mô tả, một lệnh mã máy bao gồm một mã tác vụ và các toán hạng.

Ví dụ: lệnh mã máy 01101001010101010000001101100101

Việc chọn số toán hạng cho một lệnh mã máy là một vấn đề then chốt vì phải có một sự cân đối giữa tốc độ tính toán và số các mạch tính toán phải dùng. Tùy theo tần số sử dụng các phép như trên mà các nhà thiết kế máy tính quyết định số lượng các mạch chức năng cần thiết cho việc tính toán. Thông thường số toán hạng thay đổi từ 0 tới 3.

Ví dụ: lệnh  $Y := A + B + C + D$  có thể được hiện bằng một lệnh mã máy nếu ta có 3 mạch cộng, hoặc được thực hiện bằng 3 lệnh mã máy nếu chúng ta chỉ có một mạch cộng, nếu việc tính toán trên xảy ra ít, người ta chỉ cần thiết kế một mạch cộng thay vì phải tốn chi phí lắp đặt 3 mạch cộng. Tuy nhiên, với một mạch cộng thì thời gian tính toán của hệ thống sẽ chậm hơn với hệ thống có ba mạch cộng. Vị trí của toán hạng cũng được xem xét. Chọn một vài nhà sản xuất máy tính và 3 kiểu cơ bản của vị trí các toán hạng đối với những lệnh tính toán trong ALU là: ở ngăn xếp, trên thanh ghi tích lũy, và trên các thanh ghi đa dụng. Những kiến trúc phần mềm này được gọi là kiến trúc ngăn xếp, kiến trúc thanh ghi tích lũy và kiến trúc thanh ghi đa dụng **Bảng 3.1**

Vị trí các toán hạng	Thí dụ	Toán hạng cho lệnh tính toán trong ALU	Vị trí đặt kết quả	Cách thức thâm nhập vào toán hạng
Ngăn xếp	B 5500 HP 3000/70	0	Ngăn xếp	Lệnh Push, Pop
Thanh ghi tích lũy	PDP 8 Motorola 6809	1	Thanh ghi tích lũy	Lệnh nạp vào hoặc lấy ra từ thanh ghi tích lũy (load, store)
Thanh ghi đa dụng	IBM 360 DEC, VAX	2 hoặc 3	Thanh ghi hoặc bộ nhớ	Lệnh nạp vào hoặc lấy ra từ thanh ghi hoặc bộ nhớ



Một vài nhà sản xuất máy tính tuân thủ chặt chẽ các kiểu chọn vị trí toán hạng nêu trên, nhưng phần nhiều các bộ xử lý dùng kiểu hỗn tạp. Ví dụ, mạch xử lý 8086 của Intel dùng cùng một lúc kiểu "thanh ghi đa dụng" và kiểu "thanh ghi tích lũy"

Ví dụ minh họa chuỗi lệnh phải dùng để thực hiện phép tính  $C := A + B$  trong 3 kiểu kiến trúc phần mềm.

Kiến trúc ngăn xếp	Kiến trúc thanh ghi tích lũy	Kiến trúc thanh ghi đa dụng
Push A Push B ADD Pop C	Load A <b>ADD B</b> Store C	Load R1, A ADD B,R1 Store R1,C

**Bảng 3.2:** Chuỗi lệnh dùng thực hiện phép tính  $C := A + B$

(giả sử  $A, B, C$  đều nằm trong bộ nhớ trong)

Hiện tại các nhà sản xuất máy tính có khuynh hướng dùng kiến trúc phần mềm thanh ghi đa dụng vì việc thâm nhập các thanh ghi đa dụng nhanh hơn thâm nhập bộ nhớ trong, và vì các chương trình dịch dùng các thanh ghi đa dụng có hiệu quả hơn.

### 3.6. TẬP LỆNH

Mỗi bộ vi xử lý có một tập lệnh xác định, tập lệnh thường có hàng chục đến hàng trăm lệnh và mỗi lệnh là một chuỗi nhị phân mà bộ xử lý hiểu được để thực hiện một thao tác xác định. Các lệnh được mô tả bằng các kí hiệu gọi nhớ, đó chính là các lệnh của lập trình hợp ngữ.

#### + Các thành phần của lệnh máy:

- **Mã thao tác:** mã hoá cho thao tác mà bộ xử lý phải xác thực hiện.

- **Địa chỉ toán hạng:** dữ liệu vào ra của các thao tác.

#### + Một số lệnh thường dùng của bộ vi xử lý 8086

**Dạng lệnh:** mã thao tác, địa chỉ toán hạng đích, địa chỉ toán hạng nguồn

Thực hiện mã thao tác trên toán hạng đích với toán hạng nguồn, kết quả ghi vào toán hạng đích.

Hoặc mã thao tác, toán hạng

Các thao tác cho bộ vi xử lý như các lệnh di chuyển dữ liệu, các lệnh số học, các lệnh logic và các lệnh điều khiển chương trình.

##### 1. Các lệnh di chuyển dữ liệu:

- + Lệnh MOV: Chuyển nội dung từ nguồn đến đích.
- + Lệnh LOAD: Chuyển nội dung từ bộ nhớ đến bộ xử lí.
- + Lệnh STORE: Chuyển dữ liệu từ bộ xử lí đến bộ nhớ.
- + Lệnh PUSH: đưa số liệu vào ngăn xếp (làm giảm con trỏ ngăn xếp xuống 2 đơn vị)
- + Lệnh POP: lấy số liệu ra từ ngăn xếp (làm tăng con trỏ ngăn xếp lên 2 đơn vị)

## 2. Các lệnh số học:

- + Lệnh ADD: cộng hai toán hạng,
- + Lệnh SUB: trừ hai toán hạng
- + Lệnh MUL: nhân hai toán hạng
- + Lệnh DIV: chia hai toán hạng
- + Lệnh INC: tăng nội dung lên 1 đơn vị
- + Lệnh DEC: giảm nội dung xuống 1 đơn vị
- Các lệnh logic: thực hiện các phép toán về logic.

- + Lệnh AND: thực hiện phép toán logic ‘và’.
- + Lệnh OR: thực hiện phép toán logic ‘hoặc’
- + Lệnh XOR: hoặc LOẠI
- + Lệnh NOT: đảo ngược

## 3. Các lệnh điều khiển chương trình

- + Lệnh JMP: lệnh nhảy sẽ điều khiển bộ xử lý lấy lệnh ở chỗ khác chứ không ở vị trí theo đúng trình tự.
- + Lệnh IN: chuyển dữ liệu từ một cổng xác định đến đích.
- + Lệnh OUT: chuyển dữ liệu từ nguồn đến một cổng xác định.

### 3.7. KIỂU ĐỊNH VỊ

Kiểu định vị định nghĩa cách thức thâm nhập các toán hạng. Một vài kiểu xác định cách thâm nhập toán hạng bộ nhớ, nghĩa là cách tính địa chỉ của toán hạng, các kiểu khác xác định các toán hạng nằm trong các thanh ghi.

#### Các kiểu định vị:

##### 1. Định vị tức thì :

Trong cách định vị này số liệu nằm ngay trong lệnh.

##### 2. Định vị trực tiếp :

Trong định vị trực tiếp, địa chỉ thực sự của toán hạng nằm ngay trong lệnh.

### 3. Định vị gián tiếp :

Trong phép tính định vị này, lệnh chứa địa chỉ của địa chỉ của toán hạng.

Định vị gián tiếp làm cho lệnh và số liệu cần xử lý độc lập với nhau và giúp mở rộng vùng nhớ mà lệnh có thể liên hệ được.

Tuy nhiên định vị gián tiếp thì chậm so với định vị trực tiếp (2 lần thâm nhập ô nhớ) và mạch điều khiển phức tạp hơn.

### 4. Định vị thanh ghi :

Ta có 2 kiểu định vị thanh ghi :

- Định vị trực tiếp thanh ghi: Thanh ghi chứa trực tiếp toán hạng.
- Định vị gián tiếp thanh ghi: Thanh ghi chứa địa chỉ của toán hạng.

Kiểu định vị thanh ghi nhanh hơn các kiểu định vị trực tiếp và gián tiếp ô nhớ.

### 5. Định vị tương đối :

Trong định vị tương đối, địa chỉ của toán hạng được xác định bằng độ dời (offset) của nó so với một địa chỉ nào đó mà ta gọi là địa chỉ tĩnh tiến. Vậy địa chỉ của toán hạng là độ dời + địa chỉ tĩnh tiến

Trong một số máy tính, độ dời có thể là một số đại số.

Định vị tương đối rút ngắn chiều dài của lệnh, làm cho lệnh và số liệu độc lập nhau và chương trình có thể di dời (relocatable) đến bất kỳ nơi nào trong bộ nhớ trong nghĩa là có thể nạp chương trình vào bất kỳ vùng ô nhớ nào của bộ nhớ trong mà không cần phải thay đổi nó.

Hơn nữa định vị tương đối giới hạn vùng ô nhớ mà lệnh có thể liên hệ nên bảo vệ an toàn cho các vùng ô nhớ khác.

#### - Định vị nền :

Trong kiểu định vị này, địa chỉ của toán hạng là tổng độ dời với nội dung một thanh ghi mà ta gọi là thanh ghi nền. Ta có thể thấy đây là trường hợp đặc biệt của định vị tương đối.

#### - Định vị chỉ số :

Cũng giống như định vị nền, trong định vị chỉ số, địa chỉ của toán hạng là tổng độ dời với nội dung một thanh ghi mà ta gọi là thanh ghi chỉ số.

Tuy nhiên trong định vị nền, thanh ghi nền chứa một địa chỉ mà trong định vị chỉ số thì thanh ghi chỉ số chứa độ dời và được dùng để đánh dấu một bảng, một cấu trúc.

Thanh ghi nền ít khi thay đổi nội dung còn thanh ghi chỉ số thì luôn thay đổi nội dung.

Thông thường thanh ghi chỉ số tự động tăng hoặc giảm sau khi thi hành lệnh.

#### - Định vị nền + chỉ số :

Trong kiểu định vị này địa chỉ của toán hạng bằng tổng của độ dời + nội dung thanh ghi nền + nội dung thanh ghi chỉ số.

Kiểu định vị này được dùng rộng rãi.

### 6. Định vị ngăn xếp :

Nội dung con trỏ ngăn xếp là địa chỉ của dữ liệu.

## 1 Ví dụ Kiểu định vị của một bộ xử lý có kiến trúc phần mềm kiểu thanh ghi đa dụng.

Kiểu định vị	Ví dụ về lệnh	Giải thích
<i>Tức thì</i>	Add R4, #3 ; Move R1, #5	$R4 \leftarrow R4 + 3$ ; $R1 \leftarrow 5$
<i>Trực tiếp (thanh ghi)</i>	Add R3, R4 ; Move R1, R2	$R3 \leftarrow R3 + R4$ ; $R1 \leftarrow R2$
<i>Trực tiếp (bộ nhớ)</i>	Add R1, (1001)	$R1 \leftarrow R1 + M[1001]$
<i>Gián tiếp (thanh ghi)</i>	ADD R4, (R1)	$R4 \leftarrow R4 + M[R1]$
<i>Gián tiếp (bộ nhớ)</i>	Add R1, @ (R3)	$R1 \leftarrow R1 + M[M[R3]]$
<i>Gián tiếp (thanh ghi + Độ dời)</i>	Add R4, 100(R1)	$R4 \leftarrow R4 + M[R1 + 100]$
<i>Gián tiếp (thanh ghi + thanh ghi)</i>	Add R3, (R1 + R2)	$R3 \leftarrow R3 + M[R1 + R2]$
<i>Gián tiếp ( t/g nền + t/g chỉ số + độ dời )</i>	Add R1, 100(R2)[R3]	$R1 \leftarrow R1 + M[100 + R2 + d * R3]$
<i>Ngăn xếp</i>	Add R1, (SP)	$R1 \leftarrow R1 + M(SP)$

## 3.8. NGÔN NGỮ CẤP CAO VÀ NGÔN NGỮ MÁY

Trong chi phí cho một hệ thống tin học, bao gồm giá tiền của máy tính, giá tiền các phần mềm hệ thống và các phần mềm ứng dụng, thì chi phí cho triển khai phần mềm luôn lớn hơn chi phí mua phần cứng. Vì thế các nhà tin học đã triển khai từ lâu các ngôn ngữ gọi là ngôn ngữ cấp cao. Ngôn ngữ cấp cao dùng các lệnh có cấu trúc gần với ngôn ngữ thông thường hơn ngôn ngữ máy. Các ngôn ngữ cấp cao nổi tiếng là: FORTRAN cho tính toán khoa học, COBOL cho quản lý, LISP và PROLOG dùng trong trí tuệ nhân tạo, PASCAL, C, ... Điểm chính của các ngôn ngữ này là sự cô đọng và sự độc lập đối với mọi bộ xử lý. Sự độc lập đối với mọi máy tính có nghĩa là có thể được thi hành trên mọi kiến trúc phần mềm của bộ xử lý, với điều kiện là phải có chương trình dịch để dịch chương trình viết bằng ngôn ngữ cấp cao thành chương trình mã máy của máy tính đang sử dụng.

Ở đây, chúng ta không quan tâm đến các đặc tính của ngôn ngữ cấp cao mà chỉ quan tâm đến quan hệ của nó đối với ngôn ngữ máy. Thậ vậy, muốn cho một chương trình ngôn ngữ máy được thực hiện một cách hữu hiệu thì chương trình dịch phải dịch hữu hiệu các lệnh của ngôn ngữ cấp cao thành lệnh mã máy. Muốn thế thì kiến trúc phần mềm của bộ xử lý rất quan trọng đối với chương trình dịch.

*Quá trình chuyển đổi từ ngôn ngữ cấp cao sang ngôn ngữ máy:* một bộ biên dịch (Compiler) chuyển đổi ngôn ngữ cấp cao (độc lập với kiến trúc phần mềm) sang dạng hợp ngữ (phụ thuộc kiến trúc phần mềm). Một chương trình dịch hợp ngữ (Assembler) chuyển đổi một chương trình viết bằng hợp ngữ (Assembly Language) sang ngôn ngữ máy để máy tính có thể thực hiện được chương trình đó .

### CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG III

1. Chức năng của CPU là gì? Mô tả CPU và các thành phần của nó
2. Trình bày diễn tiến thi hành lệnh mã máy? Mô tả quá trình thực thi lệnh của một máy giả thiết với các đặc trưng cho trước.
3. Ngắt quãng là gì? Nó được dùng cho những công việc nào, các giai đoạn mà BXL phải thi hành khi có ngắt quãng xảy ra?
4. Tập lệnh
5. Trình bày Các kiểu định vị

**Bài tập:** Một máy tính có từ nhớ 16 bit, các thanh ghi 16 bit. Với SP =300H, hãy cho biết kết quả của AX, BX, CX và SP sau khi thực hiện chuỗi lệnh:

```
MOV AX, FE2
MOV BX, AFB
SUB AX, BX
MOV CX, 9AC
PUSH AX
PUSH BX
MOV AX,CX
POP CX
PUSH AX
POP BX
POP AX
```

## Chương 4: BỘ NHỚ BÁN DẪN (BỘ NHỚ TRONG)

### 4.1. ĐẶC ĐIỂM VÀ PHÂN CẤP HỆ THỐNG NHỚ TRONG MÁY TÍNH

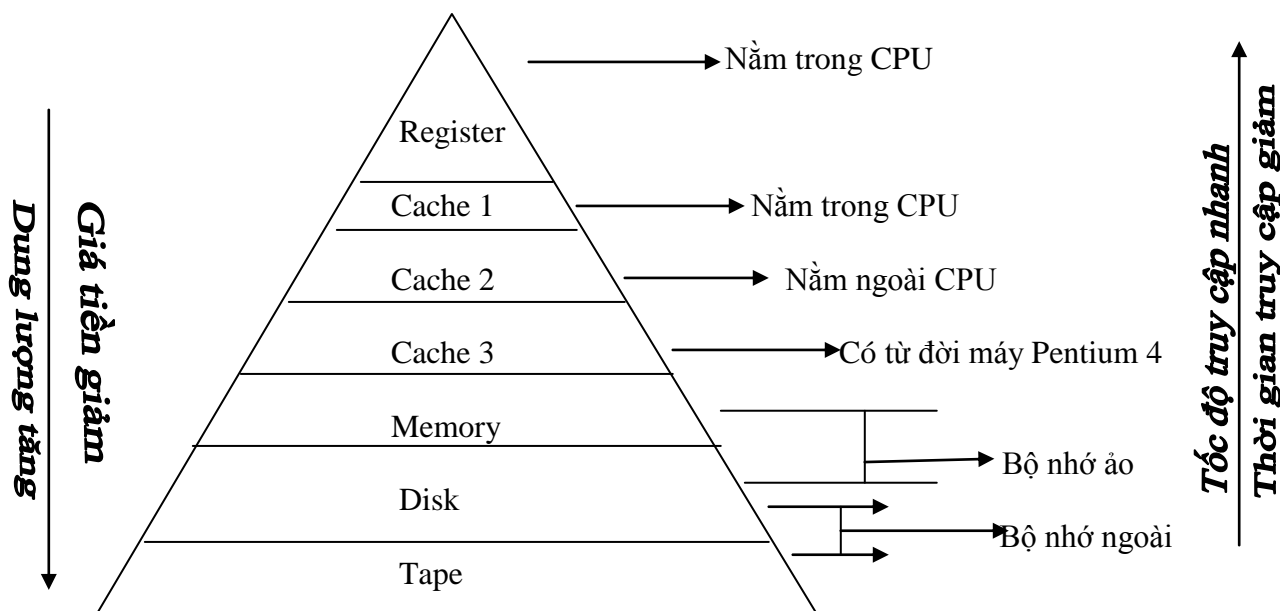
#### 4.2.1. ĐẶC ĐIỂM:

Bộ nhớ bán dẫn được xây dựng từ các mạch bán dẫn, được dùng để lưu trữ thông tin, gồm các bộ nhớ sau:

- Bộ nhớ bên trong bộ vi xử lý: Tập các thanh ghi, bộ nhớ vi chương trình (đơn vị điều khiển)
- Bộ nhớ trong gồm có bộ nhớ chính và bộ nhớ Cache. Phương pháp truy cập của bộ nhớ trong là truy cập ngẫu nhiên hay truy cập liên kết (Cache)

#### 4.2.2. PHÂN CẤP BỘ NHỚ

Các đặc tính như lượng thông tin lưu trữ, thời gian thâm nhập bộ nhớ, chu kỳ bộ nhớ, giá tiền mỗi bit nhớ khiến ta phải phân biệt các cấp bộ nhớ: các bộ nhớ nhanh với dung lượng ít đến các bộ nhớ chậm với dung lượng lớn



Hình 4.1 Sơ đồ phân cấp bộ nhớ

Các đặc tính chính của các cấp bộ nhớ dẫn đến hai mức chính là: *mức cache - bộ nhớ trong* và *mức bộ nhớ ảo* (bao gồm bộ nhớ trong và không gian cấp phát trên đĩa cứng).

Cách tổ chức này trong suốt đối với người sử dụng. Người sử dụng chỉ thấy duy nhất một không gian định vị ô nhớ, độc lập với vị trí thực tế của các lệnh và dữ liệu cần thâm nhập.

## 4.2. CÁC LOẠI BỘ NHỚ TRONG

### 4.2.1. Bộ nhớ truy cập ngẫu nhiên (RAM: Random Access Memory)

RAM chứa các chương trình và dữ liệu đang thực hiện, nhưng khi mất điện thì các thông tin trên đó sẽ bị mất, RAM có đặc tính là các ô nhớ có thể được đọc hoặc viết vào trong khoảng thời gian bằng nhau cho dù chúng ở bất kỳ vị trí nào trong bộ nhớ. Mỗi ô nhớ có một địa chỉ, được định bởi bộ vi xử lý, thông thường, mỗi ô nhớ là một byte (8 bit), nhưng hệ thống có thể đọc ra hay viết vào nhiều byte (2,4 hay 8 byte). Bộ nhớ RAM được đặc trưng bằng dung lượng và tổ chức của nó (số ô nhớ và số bit cho mỗi ô nhớ), thời gian thâm nhập (thời gian từ lúc đưa ra địa chỉ ô nhớ đến lúc đọc được nội dung ô nhớ đó) và chu kỳ bộ nhớ (thời gian giữa hai lần liên tiếp thâm nhập bộ nhớ). Dung lượng của bộ nhớ bao giờ cũng nhỏ hơn không gian địa chỉ mà bộ vi xử lý quản lý.

Tuỳ theo công nghệ chế tạo, người ta phân biệt RAM tĩnh (SRAM: Static RAM) và RAM động (DRAM: Dynamic RAM).

**4.2.1.1. RAM tĩnh:** được chế tạo theo công nghệ ECL (CMOS và BiCMOS). Mỗi bit nhớ gồm có các cổng logic với độ 6 transistor MOS, việc nhớ một dữ liệu là tồn tại nếu bộ nhớ được cung cấp điện. SRAM là bộ nhớ nhanh, việc đọc không làm huỷ nội dung của ô nhớ và thời gian thâm nhập bằng chu kỳ bộ nhớ.

**4.2.1.2. RAM động** dùng công nghệ MOS. Mỗi bit nhớ gồm có một transistor và một tụ điện. Cũng như SRAM, việc nhớ một dữ liệu là tồn tại nếu bộ nhớ được cung cấp điện. Việc ghi nhớ dựa vào việc duy trì điện tích nạp vào tụ điện và như vậy việc đọc một bit nhớ làm nội dung bit này bị huỷ. Vậy sau mỗi lần đọc một ô nhớ, bộ phận điều khiển bộ nhớ phải viết lại ô nhớ đó nội dung vừa đọc và do đó chu kỳ bộ nhớ động ít nhất là gấp đôi thời gian thâm nhập ô nhớ. Việc lưu giữ thông tin trong bit nhớ chỉ là tạm thời vì tụ điện sẽ phóng hết điện tích đã nạp vào và như vậy phải làm tươi bộ nhớ sau mỗi 2 $\mu$ s. Làm tươi bộ nhớ là đọc ô nhớ và viết lại nội dung đó vào lại ô nhớ. Việc làm tươi được thực hiện với tất cả các ô nhớ trong bộ nhớ. Việc làm tươi bộ nhớ được thực hiện tự động bởi một vi mạch bộ nhớ. Bộ nhớ DRAM chậm nhưng rẻ tiền hơn SRAM.

- **SDRAM** (Synchronous DRAM – DRAM đồng bộ), một dạng DRAM đồng bộ bus bộ nhớ. Tốc độ SDRAM đạt từ 66-133MHz (thời gian thâm nhập bộ nhớ từ 75ns-150ns).
- **DDR SDRAM** (Double Data Rate SDRAM) là cải tiến của bộ nhớ SDRAM với tốc độ truyền tải gấp đôi SDRAM nhờ vào việc truyền tải hai lần trong một chu kỳ bộ nhớ. Tốc độ DDR SDRAM đạt từ 200-400MHz
- **RDRAM** (Rambus RAM) là một loại DRAM được thiết kế với kỹ thuật hoàn toàn mới so với kỹ thuật SDRAM. RDRAM hoạt động đồng bộ theo một hệ thống lặp và truyền dữ liệu theo một hướng. Một kênh bộ nhớ RDRAM có thể hỗ trợ đến 32 chip DRAM. Mỗi chip được ghép nối tuần tự trên một module gọi là RIMM (Rambus Inline Memory Module) nhưng việc truyền dữ liệu giữa các mạch điều khiển và từng chip riêng biệt chứ không truyền giữa các chip với nhau. Bus bộ nhớ RDRAM là đường dẫn liên tục đi qua các chip và module trên bus, mỗi module có các chân vào và ra trên các đầu đối diện. Do đó, nếu các khe cắm không chứa RIMM sẽ phải gắn một module liên tục để đảm bảo đường truyền được nối liền. Tốc độ RDRAM đạt từ 400-800MHz

**4.2.2. Bộ nhớ chỉ đọc (ROM: Read Only memory)** cũng được chế tạo bằng công nghệ bán dẫn. Chương trình trong ROM được viết vào lúc chế tạo nó. Thông thường, ROM chứa chương trình khởi động máy tính, chương trình điều khiển trong các thiết bị điều khiển tự động,...

- **PROM** (Programable ROM): Chế tạo bằng các mối nối (cầu chì - có thể làm đứt bằng điện). Chương trình nằm trong PROM có thể được viết vào bởi người sử dụng bằng thiết bị đặc biệt và không thể xóa được.
- **EPROM** (Erasable Programable ROM): Chế tạo bằng nguyên tắt phân cực tĩnh điện. Chương trình nằm trong ROM có thể được viết vào (bằng điện) và có thể xóa (bằng tia cực tím - trung hòa tĩnh điện) để viết lại bởi người sử dụng.
- **EEPROM** (Electrically Erasable Programable ROM): Chế tạo bằng công nghệ bán dẫn. Chương trình nằm trong ROM có thể được viết vào và có thể xóa (bằng điện) để viết lại bởi người sử dụng.

Ngoài ra, người ta còn sản xuất ROM – BIOS (Basic Input/Output System) được sử dụng để chứa các chương trình điều khiển cơ sở vào/ra trong máy tính. Ngoài ra nó còn thực hiện chức năng sau:



- **POST** (Power On Selt Test): Khi bật máy, tiến hành kiểm tra bộ vi xử lí, bộ nhớ, chipset, card màn hình và các cấu kiện lắp vào máy (dữ liệu được lấy từ ROM và CMOS).

- **CMOS Setup**: chương trình cài đặt cấu hình hệ thống. Đây là chương trình được điều khiển bằng chương trình đơn, được kích hoạt nhờ một phím đặc biệt (phím này sẽ được hiển thị trên màn hình trong quá trình POST). Cho phép định cấu hình bo mạch chính, đặt ngày, tháng, năm, mật khẩu, ổ đĩa ... (mục này được gọi là BIOS Setup).

- Bootstrap Loader (Bootling): chương trình khởi động có chức năng tìm đọc Boot Sector (Sector 0) của ổ đĩa để bắt đầu đọc hệ điều hành.

### 4.2.3. CACHE VÀ VẬN HÀNH CỦA CACHE

**4.2.3.1. Cache:** là bộ nhớ nhanh, nó chứa lệnh và dữ liệu thường xuyên dùng đến.

- Khi thiết kế bộ vi xử lí, người ta thấy rằng tốc độ xử lý dữ liệu của nó rất cao, trong khi đó tốc độ đọc-ghi dữ liệu của bộ nhớ chính lại chậm. Do đó, bộ vi xử lí khi xử lí dữ liệu xong phải chờ một thời gian để chuyển dữ liệu từ bộ nhớ chính vào, vì vậy người ta phải thiết kế một bộ đệm (cache) là nơi trung chuyển giữa bộ vi xử lí và bộ nhớ chính, để có thể nhanh bằng tốc độ của bộ vi xử lí, người ta thiết kế hệ thống sao cho bộ điều khiển cache có thể đoán trước nhu cầu của bộ vi xử lí và nạp trước các dữ liệu cần thiết vào bộ nhớ cache.

Gọi là cache mức 1 (L1: Level 1), nằm bên trong CPU, cache mức 2 (L2: Level 2), là cache ngoài (nằm bên ngoài CPU) và một số bộ vi xử lí Pentium 4 có thêm mức cache 3 (L3: Level 3) (cũng nằm bên ngoài CPU)..

Cache có hai nguyên tắc:

□□ **Nguyên tắc về thời gian:** cho biết các ô nhớ được hệ thống xử lý thâm nhập có khả năng sẽ được thâm nhập trong tương lai gần. Thật vậy, các chương trình được cấu tạo với phần chính là phần được thi hành nhiều nhất và các phần phụ dùng để xử lý các trường hợp ngoại lệ. Còn số liệu luôn có cấu trúc và thông thường chỉ có một phần số liệu được thâm nhập nhiều nhất mà thôi.

□□ **Nguyên tắc về không gian:** cho biết, bộ xử lý thâm nhập vào một ô nhớ thì có nhiều khả năng thâm nhập vào ô nhớ có địa chỉ kế tiếp do các lệnh được sắp xếp thành chuỗi có thứ tự.

Tổ chức các cấp bộ nhớ sao cho các lệnh và dữ liệu thường dùng được nằm trong bộ nhớ cache, điều này làm tăng hiệu quả của máy tính một cách đáng kể.

**4.2.3.2. Vận hành:**

Mức cache -bộ nhớ trong trong bảng các cấp bộ nhớ có cơ cấu vận hành trong suốt đối với bộ xử lý. Với thao tác đọc bộ nhớ, bộ xử lý gửi một địa chỉ và nhận một dữ liệu từ bộ nhớ trong. Với thao tác ghi bộ nhớ, bộ xử lý viết một dữ liệu vào một ô nhớ với một địa chỉ được chỉ ra trong bộ nhớ. Để cho chương trình vận hành bình thường thì cache phải chứa một phần con của bộ nhớ trong để bộ xử lý có thể thâm nhập vào các lệnh hoặc dữ liệu thường dùng từ bộ nhớ cache. Do dung lượng của bộ nhớ cache nhỏ nên nó chỉ chứa một phần chương trình nằm trong bộ nhớ trong. Để đảm bảo sự đồng nhất giữa nội dung của cache và bộ nhớ trong thì cache và bộ nhớ trong phải có cùng cấu trúc. Việc chuyển dữ liệu giữa cache và bộ nhớ trong là việc tải lên hay ghi xuống các *khối dữ liệu*. Mỗi khối chứa nhiều từ bộ nhớ tùy thuộc vào cấu trúc bộ nhớ cache. Sự lựa chọn kích thước của khối rất quan trọng cho vận hành của cache có hiệu quả.

### 4.3. BỘ NHỚ ẢO

Bộ nhớ ảo là kỹ thuật sử dụng bộ nhớ thứ hai là ổ đĩa, để mở rộng thêm kích thước của bộ nhớ vật lý. Nó cho phép mỗi quá trình sử dụng bộ nhớ chính, và nếu chỉ có một người sử dụng, thì nó mở rộng kích thước của bộ nhớ chính vượt quá kích thước thực tế. Khái niệm này không mới, nó được mô tả lần đầu tiên bởi Kiburn và một số người vào năm 1962. Bộ nhớ ảo được thực hiện bằng cách sử dụng đơn vị quản lý bộ nhớ (Memory Management Unit – MMU) để dịch mỗi tham khảo địa chỉ logic thành tham khảo địa chỉ vật lý được đặt vào giữa CPU và bộ nhớ vật lý để thực hiện quá trình dịch dưới sự điều khiển của hệ điều hành. Mỗi tham khảo bộ nhớ của CPU sẽ được dịch từ không gian địa chỉ logic sang không gian địa chỉ vật lý. Các bảng ánh xạ hướng dẫn quá trình dịch được đặt dưới sự điều khiển của hệ điều hành. Quá trình dịch tham khảo bộ nhớ phải được thực hiện bằng phần cứng, bởi vì đòi hỏi tốc độ cao, hệ điều hành cập nhật mới nội dung các bảng ánh xạ.

Bộ nhớ ảo luôn sử dụng định dạng trang, nghĩa là trang được chuyển từ ổ đĩa vào bộ nhớ chính chỉ khi bộ xử lý truy cập một từ trên trang. Các trang bộ nhớ ảo luôn luôn có một vị trí trên đĩa, mỗi khi chúng được tạo ra, và chỉ được sao lên bộ nhớ chính khi xuất hiện *miss* hay *lỗi trang* còn gọi là *miss bộ nhớ ảo*.

Hầu hết các CPU hiện nay địa chỉ logic rất lớn so với địa chỉ vật lý. Khi dùng khái niệm bộ nhớ ảo, mỗi chương trình có thể sử dụng toàn bộ không gian địa chỉ logic của CPU, nếu chỉ có một người dùng thì bao gồm toàn bộ dung lượng của bộ nhớ thứ hai (ổ đĩa). Hệ điều hành chịu trách nhiệm luân chuyển các phần tử của chương trình hay dữ liệu cần thiết để làm việc mà

không cần người lập trình hay người dùng quan tâm tới. Trước đây MMU được thực hiện trên các chip hay bản mạch riêng lẻ, nhưng các bộ xử lý thế hệ hiện nay cài đặt bộ quản lý bộ nhớ ngay trong bộ xử lý.

**Câu hỏi ôn tập**

- 1/ Trình bày các cấp bộ nhớ. Mục tiêu của phân cấp bộ nhớ
- 2/ SRAM và DRAM khác nhau thế nào? Trong máy tính chúng được dùng ở đâu?
- 3/ Trình bày các loại bộ nhớ bán dẫn?
- 4/ Cache và sự vận hành của nó?
- 5/ Bộ nhớ ảo

## Chương 5: BỘ NHỚ NGOÀI

Bộ nhớ ngoài gồm: các đĩa từ, băng từ, đĩa quang, các loại thẻ nhớ là những bộ phận lưu trữ thông tin trữ lượng lớn và thông tin được lưu trữ trong thời gian dài. Dựa trên nguyên tắc lưu trữ từ, quang, hoặc quang từ.

### 5.1. ĐĨA TỪ

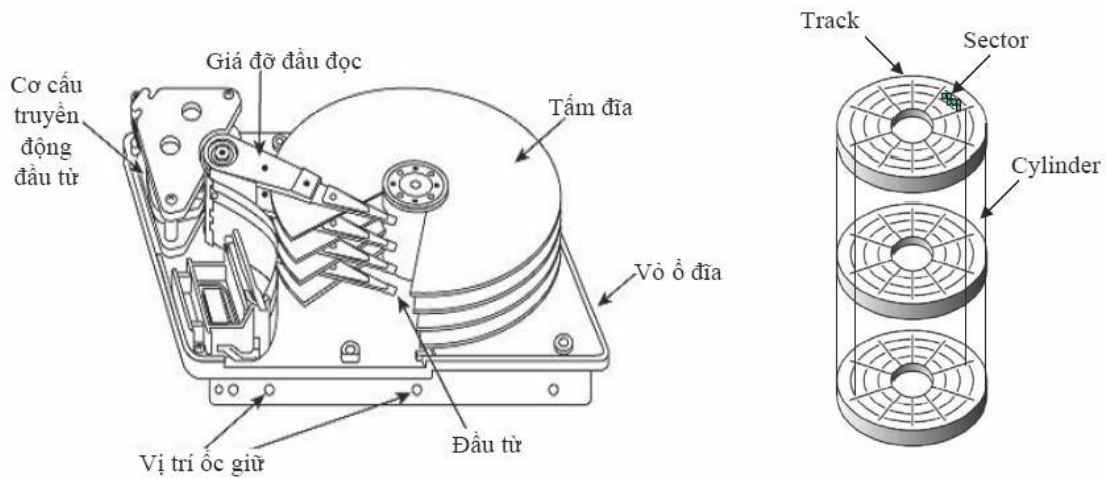
Dù rằng công nghệ mới không ngừng phát minh nhiều loại bộ phận lưu trữ một lượng thông tin lớn nhưng đĩa từ vẫn giữ vị trí quan trọng từ năm 1965. Đĩa từ có hai nhiệm vụ trong máy tính.

- Lưu trữ dài hạn các tập tin.
- Thiết lập một cấp bộ nhớ bên dưới bộ nhớ trong để làm bộ nhớ ảo lúc chạy chương trình.

Do đĩa mềm dần được các thiết bị lưu trữ khác có các tính năng ưu việt hơn nên chúng ta không xét đến thiết bị này trong chương trình mà chỉ nói đến đĩa cứng. Trong tài liệu này mô tả một cách khái quát cấu tạo, cách vận hành cũng như đề cập đến các tính chất quan trọng của đĩa cứng.

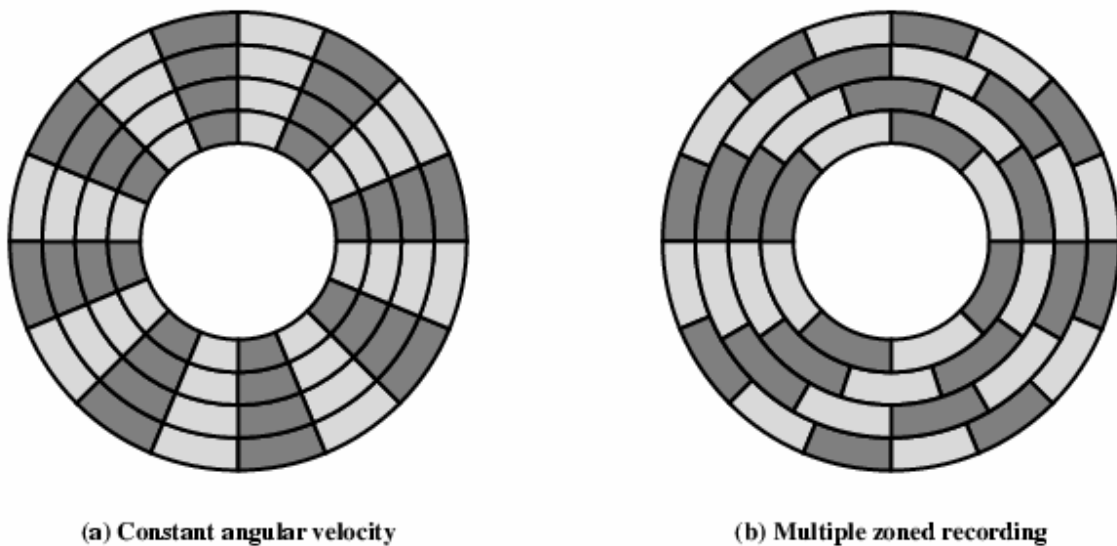
Một đĩa cứng chứa nhiều lớp đĩa (từ 1 đến 4) quay quanh một trục khoảng 3.600-15.000 vòng mỗi phút. Các lớp đĩa này được làm bằng kim loại với hai mặt được phủ một chất từ tính. Đường kính của đĩa thay đổi từ 1,3 inch đến 8 inch. Mỗi mặt của một lớp đĩa được chia thành nhiều đường tròn đồng trục gọi là *rãnh*. Thông thường mỗi mặt của một lớp đĩa có từ 10.000 đến gần 30.000 rãnh. Mỗi rãnh được chia thành nhiều *cung* (sector) dùng chứa thông tin. Một rãnh có thể chứa từ 64 đến 800 cung. Cung là đơn vị nhỏ nhất mà máy tính có thể đọc hoặc viết (thông thường khoảng 512 bytes). Chuỗi thông tin ghi trên mỗi cung gồm có: số thứ tự của cung, một khoảng trống, số liệu của cung đó bao gồm cả các mã sửa lỗi, một khoảng trống, số thứ tự của cung tiếp theo.

Với kỹ thuật ghi mật độ không đều, tất cả các rãnh đều có cùng một số cung, điều này làm cho các cung dài hơn ở các rãnh xa trục quay có mật độ ghi thông tin thấp hơn mật độ ghi trên các cung nằm gần trục quay.



**Hình 5.1: Cấu tạo của một đĩa cứng**

Với công nghệ ghi với mật độ đều, người ta cho ghi nhiều thông tin hơn ở các rãnh xa trục quay. Công nghệ ghi này ngày càng được dùng nhiều với sự ra đời của các chuẩn giao diện thông minh như chuẩn SCSI.



**Mật độ ghi không đều**

**Mật độ ghi đều**

**Hình 5.2: Mật độ ghi dữ liệu trên các loại đĩa cứng**

Để đọc hoặc ghi thông tin vào một cung, ta dùng một đầu đọc ghi di động áp vào mỗi mặt của mỗi lớp đĩa. Các đầu đọc/ghi này được gắn chặt vào một thanh làm cho chúng cùng di chuyển trên một đường bán kính của mỗi lớp đĩa và như thế tất cả các đầu này đều ở trên

những rãnh có cùng bán kính của các lớp đĩa. Từ “trụ” (cylinder) được dùng để gọi tất cả các rãnh của các lớp đĩa có cùng bán kính và nằm trên một hình trụ.

Người ta luôn muốn đọc nhanh đĩa từ nên thông thường ổ đĩa đọc nhiều hơn số dữ liệu cần đọc; người ta nói đây là cách đọc trước. Để quản lý các phức tạp khi kết nối (hoặc ngưng kết nối) lúc đọc (hoặc ghi) thông tin, và việc đọc trước, ổ đĩa cần có bộ điều khiển đĩa.

Công nghiệp chế tạo đĩa từ tập trung vào việc nâng cao dung lượng của đĩa mà đơn vị đo lường là mật độ trên một đơn vị bề mặt.

<b>Bảng thông số kỹ thuật đĩa cứng</b>	
Dung lượng tối đa	có thể đạt 500 GB
Số lượng đầu đọc	1 – 8
Số tấm ghi (đĩa)	1 - 4
Cache (bộ đệm)	2 – 16 MB
Số cung (Sectors - 512 bytes/sector)	xxx,xxx,xxx
Tốc độ quay đĩa (RPM)	3600 - 15000
Mật độ	có thể đạt 95 Gb/in <sup>2</sup>
Mật độ rãnh (TPI - Max Tracks/Inch)	có thể đạt 120,000
Mật độ ghi BPI (Max Bits/Inch)	có thể đạt 702,000
Tốc độ dữ liệu tối đa (Internal)	có thể đạt 900 Mb/s
Tốc độ truyền dữ liệu với ngoại vi	có thể đạt 320 MB/s
Thời gian chuyển track R/W	có thể đạt 15 ms
Thời gian quay nửa vòng	có thể đạt 6 ms

**Bảng 5.1: Thông số kỹ thuật của đĩa cứng**

## 5.2. ĐĨA QUANG

Các thiết bị lưu trữ quang rất thích hợp cho việc phát hành các sản phẩm văn hoá, sao lưu dữ liệu trên các hệ thống máy tính hiện nay. Ra đời vào năm 1978, đây là sản phẩm của sự hợp tác nghiên cứu giữa hai công ty Sony và Philips trong công nghiệp giải trí. Từ năm 1980 đến nay, công nghiệp đĩa quang phát triển mạnh trong cả hai lĩnh vực giải trí và lưu trữ dữ liệu máy tính. Quá trình đọc thông tin dựa trên sự phản chiếu của các tia laser năng lượng thấp từ lớp lưu trữ dữ liệu. Bộ phận tiếp nhận ánh sáng sẽ nhận biết được những điểm mà tại đó tia laser bị phản xạ mạnh hay biến mất do các vết khắc (pit) trên bề mặt

đĩa. Các tia phản xạ mạnh chỉ ra rằng tại điểm đó không có lỗ khắc và điểm này được gọi là điểm nền (land). Bộ nhận ánh sáng trong ổ đĩa thu nhận các tia phản xạ và khuếch tán được khúc xạ từ bề mặt đĩa. Khi các nguồn sáng được thu nhận, bộ vi xử lý sẽ dịch các mẫu sáng thành các bit dữ liệu hay âm thanh. Các lỗ trên CD sâu 0,12 micron và rộng 0,6 micron (1 micron bằng một phần ngàn mm). Các lỗ này được khắc theo một track hình xoắn ốc với khoảng cách 1,6 micron giữa các vòng, khoảng 16.000 track/inch. Các lỗ (pit) và nền (land) kéo dài khoảng 0,9 đến 3,3 micron. Track bắt đầu từ phía trong và kết thúc ở phía ngoài theo một đường khép kín các rìa đĩa 5mm. Dữ liệu lưu trên CD thành từng khối, mỗi khối chứa 2.352 byte. Trong đó, 304 byte chứa các thông tin về bit đồng bộ, bit nhận dạng (ID), mã sửa lỗi (ECC), mã phát hiện lỗi (EDC). Còn lại 2.048 byte chứa dữ liệu. Tốc độ đọc chuẩn của CD-ROM là 75 khối/s hay 153.600 byte/s hay 150KB/s (1X). Dưới đây là một số loại đĩa quang thông dụng.

**CD (Compact Disk):** Đĩa quang không thể xoá được, dùng trong công nghiệp giải trí (các đĩa âm thanh được số hoá). Chuẩn đĩa có đường kính 12 cm, âm thanh phát từ đĩa (không dùng) khoảng 60 phút.

**CD-ROM (Compact Disk Read Only Memory):** Đĩa không xoá dùng để chứa các dữ liệu máy tính. Chuẩn đĩa có đường kính 12 cm, lưu trữ dữ liệu hơn 650 MB. Khi phát hành, đĩa CD-ROM đã có chứa nội dung. Thông thường, đĩa CD-ROM được dùng để chứa các phần mềm và các chương trình điều khiển thiết bị.

**CD-R (CD-Recordable):** Giống như đĩa CD, đĩa mới chưa có thông tin, người dùng có thể ghi dữ liệu lên đĩa một lần và đọc được nhiều lần. Dữ liệu trên đĩa CD-R không thể bị xoá.

**CD-RW (CD-Rewritable):** Giống như đĩa CD, đĩa mới chưa có thông tin, người dùng có thể ghi dữ liệu lên đĩa, xoá và ghi lại dữ liệu trên đĩa nhiều lần.

**DVD (Digital Video Disk - Digital Versatile Disk):** Ra đời phục vụ cho công nghiệp giải trí, đĩa chứa các hình ảnh video được số hoá. Ngày nay, DVD được sử dụng rộng rãi trong các ứng dụng công nghệ thông tin. Kích thước đĩa có hai loại: 8cm và 12 cm. Đĩa DVD có thể chứa dữ liệu trên cả hai mặt đĩa, dung lượng tối đa lên đến 17GB. Các thông số kỹ

thuật của đĩa DVD-ROM (loại đĩa chỉ đọc) so với CD-ROM. Tốc độ đọc chuẩn (1X) của DVD là 1.3MB/s (1X của DVD tương đương khoảng 9X của CDROM).

**DVD-R (DVD-Recordable):** Giống như đĩa DVD-ROM, người dùng có thể ghi dữ liệu lên đĩa một lần và đọc được nhiều lần. Đĩa này chỉ có thể ghi được trên một mặt đĩa, dung lượng ghi trên mỗi mặt tối đa là 4.7 GB.

**DVD-RW (DVD-Rewritable):** Giống như đĩa DVD-ROM, người dùng có thể ghi, xoá và ghi lại dữ liệu lên đĩa nhiều lần.. Đĩa này cũng có thể ghi được trên một mặt đĩa, dung lượng ghi trên mỗi mặt tối đa là 4.7 GB.

Với các đặc tính của đĩa quang, giá thành ngày càng thấp, được xem như một phương tiện thích hợp để phân phối các phần mềm cho máy vi tính. Ngoài ra, đĩa quang còn được dùng để lưu trữ lâu dài các dữ liệu thay thế cho băng từ.

### **5.3. CÁC LOẠI THẺ NHỚ**

Hiện nay, thẻ nhớ là một trong những công nghệ mới nhất được dùng làm thiết bị lưu trữ. Thẻ nhớ flash là một dạng bộ nhớ bán dẫn EEPROM(công nghệ dùng để chế tạo các chip BIOS trên các vi mạch chính), được lắp qua cổng USB hoặc qua khe card riêng.

### **5.4. BĂNG TỪ**

Băng từ có cùng công nghệ với các đĩa từ nhưng khác đĩa từ hai điểm:

- Việc thâm nhập vào đĩa từ là ngẫu nhiên còn việc thâm nhập vào băng từ là tuần tự. Như vậy việc tìm thông tin trên băng từ mất nhiều thời gian hơn việc tìm thông tin trên đĩa từ.
- Đĩa từ có dung lượng hạn chế còn băng từ gồm có nhiều cuộn băng có thể lấy ra khỏi máy đọc băng nên dung lượng của băng từ là rất lớn (hàng trăm GB). Với chi phí thấp, băng từ vẫn còn được dùng rộng rãi trong việc lưu trữ dữ liệu dự phòng.

Các băng từ có chiều rộng thay đổi từ 0,38cm đến 1,27 cm được đóng thành cuộn và được chứa trong một hộp bảo vệ. Dữ liệu ghi trên băng từ có cấu trúc gồm một số các rãnh song song theo chiều dọc của băng.

Có hai cách ghi dữ liệu lên băng từ:

Ghi nối tiếp: với kỹ thuật ghi xoắn ốc, dữ liệu ghi nối tiếp trên một rãnh của băng từ, khi kết thúc một rãnh, băng từ sẽ quay ngược lại, đầu từ sẽ ghi dữ liệu trên rãnh mới tiếp theo nhưng với hướng ngược lại. Quá trình ghi cứ tiếp diễn cho đến khi đầy băng từ.



Ghi song song: để tăng tốc độ đọc-ghi dữ liệu trên băng từ, đầu đọc - ghi có thể đọc-ghi một số rãnh kề nhau đồng thời. Dữ liệu vẫn được ghi theo chiều dọc băng từ nhưng các khối dữ liệu được xem như ghi trên các rãnh kề nhau. Số rãnh ghi đồng thời trên băng từ thông thường là 9 rãnh (8 rãnh dữ liệu - 1 byte và một rãnh kiểm tra lỗi).

## 5.5. HỆ THỐNG DĨA DỰ PHÒNG (RAID)

Người ta thường chú trọng đến sự an toàn trong lưu giữ thông tin ở đĩa từ hơn là sự an toàn của thông tin trong bộ xử lý. Bộ xử lý có thể hư mà không làm tổn hại đến thông tin. Ổ đĩa của máy tính bị hư có thể gây ra các thiệt hại rất to lớn.

Một phương pháp giúp tăng cường độ an toàn của thông tin trên đĩa từ là dùng một mảng đĩa từ. Mảng đĩa từ này được gọi là *Hệ thống đĩa dự phòng (RAID - Redundant Array of Independent Disks)*. Cách lưu trữ dư thông tin làm tăng giá tiền và sự an toàn (ngoại trừ RAID 0). Cơ chế RAID có các đặc tính sau:

1. RAID là một tập hợp các ổ đĩa cứng (vật lý) được thiết lập theo một kỹ thuật mà hệ điều hành chỉ “nhìn thấy” chỉ là một ổ đĩa (logic) duy nhất.
2. Với cơ chế đọc/ghi thông tin diễn ra trên nhiều đĩa (ghi đan chéo hay soi gương).
3. Trong mảng đĩa có lưu các thông tin kiểm tra lỗi dữ liệu; do đó, dữ liệu có thể được phục hồi nếu có một đĩa trong mảng đĩa bị hư hỏng .

Tuỳ theo kỹ thuật thiết lập, RAID có thể có các mức sau:

**5.5.1. RAID 0:** Thực ra, kỹ thuật này không nằm trong số các kỹ thuật có cơ chế an toàn dữ liệu. Khi mảng được thiết lập theo RAID 0, ổ đĩa logic có được (mà hệ điều hành nhận biết) có dung lượng bằng tổng dung lượng của các ổ đĩa thành viên. Điều này giúp cho người dùng có thể có một ổ đĩa logic có dung lượng lớn hơn rất nhiều so với dung lượng thật của ổ đĩa vật lý cùng thời điểm. Dữ liệu được ghi phân tán trên tất cả các đĩa trong mảng. Đây chính là sự khác biệt so với việc ghi dữ liệu trên các đĩa riêng lẻ bình thường bởi vì thời gian đọc-ghi dữ liệu trên đĩa tỉ lệ nghịch với số đĩa có trong tập hợp (số đĩa trong tập hợp càng nhiều, thời gian đọc – ghi dữ liệu càng nhanh). Tính chất này của RAID 0 thật sự hữu ích trong các ứng dụng yêu cầu nhiều thâm nhập đĩa với dung lượng lớn, tốc độ cao (đa phương tiện, đồ hoạ,...). Tuy nhiên, như đã nói ở trên, kỹ thuật

này không có cơ chế an toàn dữ liệu, nên khi có bất kỳ một hư hỏng nào trên một đĩa thành viên trong mảng cũng sẽ dẫn đến việc mất dữ liệu toàn bộ trong mảng đĩa. Xác suất hư hỏng đĩa tỉ lệ thuận với số lượng đĩa được thiết lập trong RAID 0. RAID 0 có thể được thiết lập bằng phần cứng (RAID controller) hay phần mềm (Striped Applications)

**5.5.2. . RAID 1 (Mirror - Đĩa gương):** Phương cách thông thường tránh mất thông tin khi ổ đĩa bị hư là dùng đĩa gương, tức là dùng 2 đĩa. Khi thông tin được viết vào một đĩa, thì nó cũng được viết vào đĩa gương và như vậy luôn có một bản sao của thông tin. Trong cơ chế này, nếu một trong hai đĩa bị hư thì đĩa còn lại được dùng bình thường. Việc thay thế một đĩa mới (cung thông số kỹ thuật với đĩa hư hỏng) và phục hồi dữ liệu trên đĩa đơn giản. Căn cứ vào dữ liệu trên đĩa còn lại, sau một khoảng thời gian, dữ liệu sẽ được tái tạo. trên đĩa mới (rebuild). RAID 1 cũng có thể được thiết lập bằng phần cứng (RAID controller) hay phần mềm (Mirror Applications) với chi phí khá lớn, hiệu suất sử dụng đĩa không cao (50%).

**5.5.3. . RAID 3:** Dùng kỹ thuật ghi song song, trong kỹ thuật này, mảng được thiết lập với yêu cầu tối thiểu là 3 đĩa có các thông số kỹ thuật giống nhau, chỉ một đĩa trong mảng được dùng để lưu các thông tin kiểm tra lỗi (parity bit). Như vậy, khi thiết lập RAID 3, hệ điều hành nhận biết được một đĩa logic có dung lượng  $n-1/n$  ( $n$ : số đĩa trong mảng). Dữ liệu được chia nhỏ và ghi đồng thời trên  $n-1$  đĩa và bit kiểm tra chẵn lẻ được ghi trên đĩa dùng làm đĩa chứa bit parity – chẵn lẻ đan chéo ở mức độ bit. Bit chẵn lẻ là một bit mà người ta thêm vào một tập hợp các bit làm cho số bit có trị số 1 (hoặc 0) là chẵn (hay lẻ). Thay vì có một bản sao hoàn chỉnh của thông tin gốc trên mỗi đĩa, người ta chỉ cần có đủ thông tin để phục hồi thông tin đã mất trong trường hợp có hỏng ổ đĩa. Khi một đĩa bất kỳ trong mảng bị hư, hệ thống vẫn hoạt động bình thường. Khi thay thế một đĩa mới vào mảng, căn cứ vào dữ liệu trên các đĩa còn lại, hệ thống tái tạo thông tin. Hiệu suất sử dụng đĩa cho cách thiết lập này là  $n-1/n$ . RAID 3 chỉ có thể được thiết lập bằng phần cứng (RAID controller).

**5.5.4. . RAID 2:** Dùng kỹ thuật truy cập đĩa song song, tất cả các đĩa thành viên trong RAID đều được đọc khi có một yêu cầu từ ngoại vi. Một mã sửa lỗi (ECC) được tính toán dựa vào các dữ liệu được ghi trên đĩa lưu dữ liệu, các bit được mã hoá được lưu trong các

đĩa dùng làm đĩa kiểm tra. Khi có một yêu cầu dữ liệu, tất cả các đĩa được truy cập đồng thời. Khi phát hiện có lỗi, bộ điều khiển nhận dạng và sửa lỗi ngay mà không làm giảm thời gian truy cập đĩa. Với một thao tác ghi dữ liệu lên một đĩa, tất cả các đĩa dữ liệu và đĩa sửa lỗi đều được truy cập để tiến hành thao tác ghi. Thông thường, RAID 2 dùng mã Hamming để thiết lập cơ chế mã hoá, theo đó, để mã hoá dữ liệu được ghi, người ta dùng một bit sửa lỗi và hai bit phát hiện lỗi. RAID 2 thích hợp cho hệ thống yêu cầu giảm thiểu được khả năng xảy ra nhiều đĩa hư hỏng cùng lúc

**5.5.5. . RAID 4:** từ RAID 4 đến RAID 6 dùng kỹ thuật truy cập các đĩa trong mảng độc lập. Trong một mảng truy cập độc lập, mỗi đĩa thành viên được truy xuất độc lập, do đó mảng có thể đáp ứng được các yêu cầu song song của ngoại vi. Kỹ thuật này thích hợp với các ứng dụng yêu cầu nhiều ngoại vi là các ứng dụng yêu cầu tốc độ truyền dữ liệu cao. Trong RAID 4, một đĩa dùng để chứa các bit kiểm tra được tính toán từ dữ liệu được lưu trên các đĩa dữ liệu. Khuyết điểm lớn nhất của RAID 4 là bị nghẽn cổ chai tại đĩa kiểm tra khi có nhiều yêu cầu đồng thời từ các ngoại vi.

**5.5.6. RAID 5:** yêu cầu thiết lập giống như RAID 4, dữ liệu được ghi từng khối trên các đĩa thành viên, các bit chẵn lẻ được tính toán mức độ khối được ghi trải đều lên trên tất cả các ổ đĩa trong mảng. Tương tự RAID 4, khi một đĩa bất kỳ trong mảng bị hư hỏng, hệ thống vẫn hoạt động bình thường. Khi thay thế một đĩa mới vào mảng, căn cứ vào dữ liệu trên các đĩa còn lại, hệ thống tái tạo thông tin. Hiệu suất sử dụng đĩa cho cách thiết lập này là  $n-1/n$ . RAID 5 chỉ có thể được thiết lập bằng phần cứng (RAID controller). Cơ chế này khắc phục được khuyết điểm đã nêu trong cơ chế RAID 4.

**5.5.7. RAID 6:** Trong kỹ thuật này, cần có  $n+2$  đĩa trong mảng. Trong đó,  $n$  đĩa dữ liệu và 2 đĩa riêng biệt để lưu các khối kiểm tra. Một trong hai đĩa kiểm tra dùng cơ chế kiểm tra như trong RAID 4&5, đĩa còn lại kiểm tra độc lập theo một giải thuật kiểm tra. Qua đó, nó có thể phục hồi được dữ liệu ngay cả khi có hai đĩa dữ liệu trong mảng bị hư hỏng. Hiện nay, RAID 0,1,5 được dùng nhiều trong các hệ thống. Các giải pháp RAID trên đây (trừ RAID 6) chỉ đảm bảo an toàn dữ liệu khi có một đĩa trong mảng bị hư hỏng. Ngoài ra, các hư hỏng dữ liệu do phần mềm hay chủ quan của con người không được đề cập

trong chương trình. Người dùng cần phải có kiến thức đầy đủ về hệ thống để các hệ thống thông tin hoạt động hiệu quả và an toàn.

## **CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG V**

1. Các bộ phận lưu trữ thông tin Các giao diện với các bộ phận vào ra.
2. Mô tả vận hành của ổ đĩa cứng. Cách lưu trữ thông tin trong ổ đĩa cứng
3. Mô tả các biện pháp an toàn trong việc lưu trữ thông tin trong đĩa cứng.
4. Nguyên tắc vận hành của đĩa quang. Ưu khuyết điểm của các loại đĩa quang.  
Các loại đĩa quang.

## Chương 6: NHẬP XUẤT

Bộ xử lý của máy tính điện tử liên hệ với bên ngoài nhờ các bộ phận xuất nhập (I/O) mà ta còn gọi là ngoại vi.

Các ngoại vi thông dụng là: Màn hình, bàn phím, chuột, máy in, thẻ mạng... là những bộ phận giúp con người sử dụng máy tính dễ dàng và. bộ nhớ ngoài.

Tất cả các ngoại vi đều được kết nối vào bộ xử lý và bộ nhớ trong.

### 6.1. BUS NỐI NGOẠI VI VÀO BỘ XỬ LÝ VÀ BỘ NHỚ TRONG

Trong máy tính, bộ xử lý và bộ nhớ trong liên lạc với các ngoại vi bằng bus. Bus là một hệ thống các dây cáp nối (khoảng 50 đến 100 sợi cáp riêng biệt) trong đó một nhóm các cáp được định nghĩa chức năng khác nhau bao gồm: các đường dữ liệu, các đường địa chỉ, các dây điều khiển, cung cấp nguồn. Dùng bus có 2 ưu điểm là giá tiền thấp và dễ thay đổi ngoại vi. Người ta có thể gỡ bỏ một ngoại vi hoặc thêm vào ngoại vi mới cho các máy tính dùng cùng một hệ thống bus.

Giá tiền thiết kế và thực hiện một hệ thống bus là rẻ, vì nhiều ngã vào/ra cùng chia sẻ một số đường dây đơn giản. Tuy nhiên, điểm bất lợi chính của bus là tạo ra nghẽn cổ chai, điều này làm giới hạn lưu lượng vào/ra tối đa. Các hệ thống máy tính dùng cho quản lý phải dùng thường xuyên các ngoại vi, nên khó khăn chính là phải có một hệ thống bus đủ khả năng phục vụ bộ xử lý trong việc liên hệ với các ngoại vi.

Một trong những lý do khiến cho việc thiết kế một hệ thống bus khó khăn là tốc độ tối đa của bus bị giới hạn bởi các yếu tố vật lý như chiều dài của bus và số bộ phận được mắc vào bus.

Các bus thường có hai loại: bus hệ thống nối bộ xử lý với bộ nhớ (system bus, Front Side Bus-FSB) và bus nối ngoại vi (bus vào/ra – I/O bus). Bus vào/ra có thể có chiều dài lớn và có khả năng nối kết với nhiều loại ngoại vi, các ngoại vi này có thể có lưu lượng thông tin khác nhau, định dạng dữ liệu khác nhau. Bus kết nối bộ xử lý với bộ nhớ thì ngắn và thường thì rất nhanh. Trong giai đoạn thiết kế bus kết nối bộ xử lý với bộ nhớ, nhà thiết kế biết trước các linh kiện và bộ phận mà ông ta cần kết nối lại, còn nhà thiết kế bus vào/ra phải thiết kế bus thoả mãn nhiều ngoại vi có mức trì hoãn và lưu lượng rất khác nhau

Hiện nay, trong một số hệ thống máy tính, bus nối ngoại vi được phân cấp thành hai hệ thống bus con. Trong đó, bus tốc độ cao (high-speed bus) hỗ trợ kết nối các thiết bị tốc độ cao như SCSI, LAN, Graphic, Video,... và hệ thống bus mở rộng (expansion bus) được thiết kế để kết nối với các ngoại vi yêu cầu tốc độ thấp như: modem, cổng nối tiếp, cổng song song,... Giữa hai hệ thống bus nối ngoại vi trong tổ chức hệ thống bus phân cấp là một giao diện đệm.

#### **6.1.1. Cổng nối tiếp (Serial Port)**

Cổng nối tiếp (còn gọi là cổng COM) được sử dụng cho những thiết bị phải giao tiếp hai chiều với hệ thống: như modem, chuột và những thiết bị truyền, nhận thông tin từ các máy khác.

Cổng nối tiếp dữ liệu phải truyền lần lượt từng bit, vì vậy cần có bộ chuyển đổi dữ liệu song song trong máy tính thành dạng nối tiếp và sau đó chuyển dữ liệu nối tiếp trở lại dạng song song. Đó chính là chip điều khiển quá trình phân chia dữ liệu UART (Universal Asynchronous Receiver Transmitter).

Do truyền lần lượt nên tốc độ chậm và cần ít đường truyền.

#### **6.1.2. Cổng song song (Parallel Port)**

Cổng song song (còn gọi là cổng LPT) ban đầu được thiết kế làm giao diện kết nối với máy in, tuy nhiên trở nên hữu ích trong vai trò một giao diện tốc độ cao.

Cổng song song dữ liệu truyền được nhiều bit song song, do đó tốc độ nhanh và cần nhiều đường truyền. Tuy nhiên, hạn chế của cổng song song đó là chiều dài cáp.

#### **6.1.3. Giao diện USB và IEEE 1394**

Hai kiến trúc bus nối tiếp USB và IEEE 1394. Đây là cổng giao diện có tốc độ vượt xa so với các cổng nối tiếp và song song chuẩn.

**6.1.3.1. Giao diện USB (Universal Serial Bus) :** là chuẩn bus cho các thiết bị ngoại vi gắn ngoài, nhằm đem lại khả năng Plug – and – Play cho việc kết nối các thiết bị bên ngoài muốn kết nối vào máy tính. Kiến trúc USB 1.0 được đưa vào sử dụng từ năm 1996, đến nay kiến trúc này luôn được phát triển để đạt được khả năng truyền dữ liệu cao.

**6.1.3.2. Giao diện IEEE 1394 :** là công nghệ khá mới. Công nghệ này ra đời xuất phát từ nhu cầu truyền dữ liệu rất lớn của các thiết bị đa phương tiện.

Chuẩn 1394 có các tốc độ truyền khác nhau: 100, 200, 400 Mbit/s và Gigabit/s.

Ta có thể có nhiều lựa chọn trong việc thiết kế một bus, như trong bảng 6.1

Đặc tính của bus	Bus hệ thống	Bus nối ngoại vi
Độ rộng của bus	Đường dây địa chỉ và số liệu khác nhau	Đường địa chỉ và số liệu được đa hợp
Độ rộng bus số liệu	Càng rộng càng nhanh (ví dụ 64 bit)	Càng hẹp càng ít tốn kém (ví dụ 8 bit)
Số từ được chuyển	Chuyển nhiều từ	Chuyển đơn giản mỗi lần một từ
Chủ nhân của bus	Nhiều	Một
Chuyển từng gói	Có. Cần nhiều chủ nhân bus	Không. Kết nối một lần và chuyển hết thông tin
Xung nhịp	Đồng bộ	Bất đồng bộ

**Bảng 6.1: Các lựa chọn chính yếu cho một bus**

Trong bảng 6.1 có khái niệm sau đây liên quan đến các *chủ nhân của bus* - các bộ phận có thể khởi động một tác vụ đọc hoặc viết trên bus. Ví dụ bộ xử lý luôn là một chủ nhân của bus. Một bus có nhiều chủ nhân khi nó có nhiều bộ xử lý, hoặc khi các ngoại vi có thể khởi động một tác vụ có dùng bus. Nếu có nhiều chủ nhân của bus thì phải có một cơ chế trọng tài để quyết định chủ nhân nào được quyền chiếm lĩnh bus. Một bus có nhiều chủ, có thể cấp một dải thông rộng (bandwidth) bằng cách sử dụng các gói tin thay vì dùng bus cho từng tác vụ riêng lẻ. Kỹ thuật sử dụng gói tin được gọi là phân chia nhỏ tác vụ (dùng bus chuyển gói). Một tác vụ đọc được phân tích thành một tác vụ yêu cầu đọc (tác vụ này chứa địa chỉ cần đọc), và một tác vụ trả lời của bộ nhớ (chứa thông tin cần đọc). Mỗi tác vụ đều có một nhãn cho biết loại của tác vụ. Trong kỹ thuật phân chia nhỏ tác vụ, trong khi bộ nhớ đọc các thông tin ở địa chỉ đã xác định thì bus được dành cho các chủ khác.

Bus hệ thống là một bus đồng bộ, nó gồm có một xung nhịp trong các đường dây điều khiển, và một nghi thức cho các địa chỉ và các số liệu đối với xung nhịp. Do có rất ít hoặc không có mạch logic nào dùng để quyết định hành động kế tiếp nào cần thực hiện, nên các bus đồng bộ vừa nhanh, vừa rẻ tiền. Trên bus này, tất cả đều phải vận hành với cùng một xung nhịp.

Ngược lại, các bus vào/ra thuộc loại bus bất đồng bộ, các bus này không có xung nhịp đồng bộ trong hệ thống bus. Thay vào đó có các nghi thức bắt tay với các quy định riêng về thời gian, được dùng giữa các bộ phận phát và bộ phận thu của bus. Bus bất đồng bộ rất dễ thích ứng với nhiều ngoại vi và cho phép nối dài bus mà không phải lo ngại gì đến vấn đề đồng bộ. Bus bất đồng bộ cũng dễ thích ứng với những thay đổi công nghệ.

## 6.2. CÁC CHUẨN VỀ BUS

Số lượng và chủng loại các bộ phận vào/ra không cần định trước trong các hệ thống xử lý thông tin. Điều này giúp cho người sử dụng máy tính dùng bộ phận vào/ra nào đáp ứng được các yêu cầu của họ. Vào/ra là giao diện trên đó các bộ phận (thiết bị) được kết nối vào hệ thống. Nó có thể xem như một bus nối rộng dùng để kết nối thêm ngoại vi vào máy tính. Các chuẩn làm cho việc nối kết các ngoại vi vào máy tính được dễ dàng; bởi vì, trong khi các nhà thiết kế-sản xuất máy tính và các nhà thiết kế-sản xuất ngoại vi có thể thuộc các công ty khác nhau. Sự tồn tại các chuẩn về bus là rất cần thiết. Như vậy, nếu nhà thiết kế máy tính và nhà thiết kế ngoại vi tôn trọng các chuẩn về bus này thì các ngoại vi có thể kết nối dễ dàng vào máy tính. Chuẩn của bus vào/ra là tài liệu quy định cách kết nối ngoại vi vào máy tính.

Các máy tính quá thông dụng thì các chuẩn về bus vào/ra của chúng có thể được xem là chuẩn cho các hãng khác (ví dụ: trước đây, UNIBUS của máy PDP 11, các chuẩn về bus của máy IBM PC, AT và hiện nay là các chuẩn của hãng Intel liên quan đến các máy vi tính). Các chuẩn về bus phải được các cơ quan về chuẩn như ISO, ANSI và IEEE công nhận.

## 6.3. GIAO DIỆN GIỮA BỘ XỬ LÝ VỚI CÁC BỘ PHẬN VÀO RA

Bộ xử lý dùng 2 cách để liên lạc với các bộ phận vào ra:

\* *Cách thứ nhất*, thường được dùng: là cách dùng một vùng địa chỉ của bộ nhớ làm vùng địa chỉ của các ngoại vi. Khi đọc hay viết vào vùng địa chỉ này của bộ nhớ là liên hệ đến các ngoại vi.

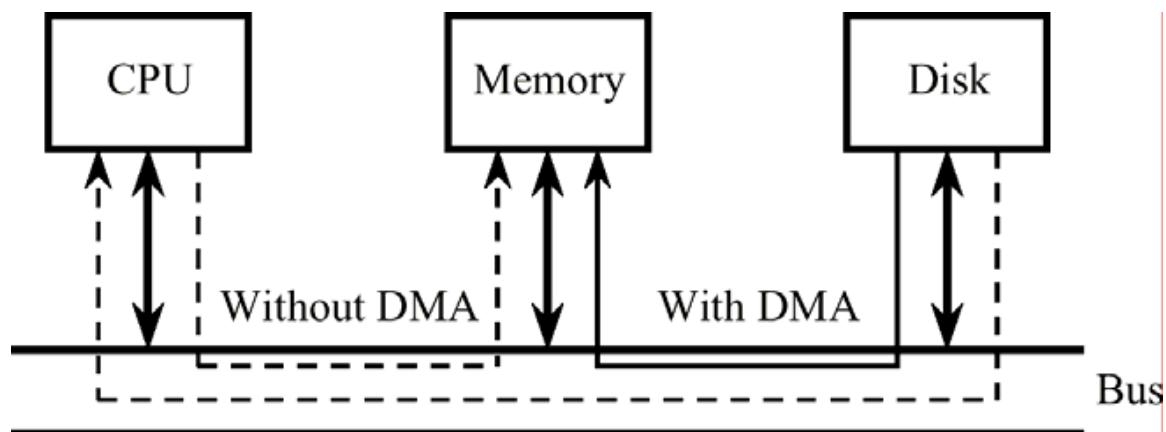
\* *Cách thứ hai*, dùng mã lệnh riêng biệt cho vào/ra (tức là có các lệnh vào/ra riêng, không trùng với lệnh đọc hay viết vào ô nhớ). Trong trường hợp này, bộ xử lý gửi một tín hiệu điều khiển cho biết địa chỉ đang dùng là của một ngoại vi. Vi mạch Intel 8086 và máy IBM 370 là các ví dụ về bộ xử lý dùng lệnh vào/ra riêng biệt.

Dù dùng cách nào để định vị vào/ra thì mỗi bộ phận vào/ra đều có các thanh ghi để cung cấp thông tin về trạng thái và về điều khiển. Bộ phận vào/ra dùng bit trạng thái “*sẵn sàng*” để báo cho bộ xử lý nó sẵn sàng nhận số liệu. Định kỳ bộ xử lý xem xét bit này để biết bộ phận vào ra có sẵn sàng hay không. Phương pháp này là *phương pháp thăm dò*



(polling). Và nhược điểm của phương pháp này là làm mất thời gian của bộ xử lý vì định kỳ phải thăm dò tính sẵn sàng của các thiết bị ngoại vi. Điều này đã được nhận thấy từ lâu và đã dẫn đến phát minh ra ngắt quãng (interrupt) để báo cho bộ xử lý biết lúc có một bộ phận vào/ra cần được phục vụ.

Việc dùng ngắt quãng làm cho bộ xử lý không mất thời gian thăm dò xem các ngoại vi có yêu cầu phục vụ hay không, nhưng bộ xử lý phải mất thời gian chuyển dữ liệu. Thông thường việc trao đổi số liệu giữa ngoại vi và CPU là theo khối số liệu, nên vi mạch thâm nhập trực tiếp bộ nhớ trong (DMA: Direct Memory Access) được dùng trong nhiều máy tính để chuyển một khối nhiều từ mà không có sự can thiệp của CPU.



**Hình 6.1 Sơ đồ hoạt động của hệ thống bus có vi mạch DMA**

DMA là một vi mạch chức năng đặc biệt. Nó chuyển số liệu giữa ngoại vi và bộ nhớ trong, trong lúc đó CPU rảnh rỗi để làm công việc khác. Vậy DMA nằm ngoài CPU và tác động như là một chủ nhân của bus. Bộ xử lý khởi động các thanh ghi của DMA, các thanh ghi này chứa địa chỉ ô nhớ và số byte cần chuyển. DMA chủ động chuyển số liệu và khi chấm dứt thì trả quyền điều khiển cho bộ xử lý.

Vi mạch DMA càng thông minh thì công việc của CPU càng nhẹ đi. Nhiều vi mạch được gọi là bộ xử lý vào/ra (hay bộ điều khiển vào/ra) thực hiện công việc mình theo một chương trình cố định (chứa trong ROM), hay theo một chương trình mà hệ điều hành nạp vào bộ nhớ trong. Hệ điều hành thiết lập một hàng chờ đợi gồm các khối điều khiển các

bộ phận vào/ ra. Các khối chứa các thông tin như là vị trí của số liệu (nguồn và đích) và số số liệu. Các bộ xử lý vào/ra lấy các thông tin này trong hàng chờ đợi, thực hiện các việc cần phải làm và gửi về CPU tín hiệu ngắt khi đã thực hiện xong công việc.

Một máy tính có bộ xử lý vào/ra được xem như một máy tính đa xử lý vì DMA giúp cho máy tính thực hiện cùng lúc nhiều quá trình. Tuy nhiên bộ xử lý vào/ra không tổng quát bằng các bộ xử lý vì chúng chỉ làm được một số việc nhất định. Hơn nữa bộ xử lý vào/ra không chế biến số liệu như các bộ xử lý thường làm. Nó chỉ di chuyển số liệu từ nơi này sang nơi khác.

## **CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG VI**

**\*\*\*\*\***

1. Thông thường có bao nhiêu loại bus? Tại sao phải có các chuẩn cho các bus vào ra?
2. Thế nào là chủ nhân của bus? Khi bus có nhiều chủ nhân thì làm thế nào để giải quyết tranh chấp bus?
3. Sự khác biệt giữa bộ xử lý vào ra và bộ xử lý trung tâm của máy tính.

MỤC LỤC

<b>Chương I: TỔNG QUAN VỀ KIẾN TRÚC MÁY TÍNH</b>	<b>1</b>
I. CẤU TRÚC VÀ CHỨC NĂNG CỦA MÁY TÍNH:	1
A. CHỨC NĂNG CỦA MÁY TÍNH:	1
B CẤU TRÚC PHẦN CỨNG CỦA MÁY TÍNH.....	<b>Error! Bookmark not defined.</b>
1 Sơ đồ cấu trúc phần cứng.....	1
2. Khối xử lý trung tâm (CPU) .....	2
3. Khối bộ nhớ (RAM/ROM) .....	2
4. Khối các thiết bị vào ra (Input/Output).....	3
5 BUS.....	3
II. CÁC THỂ HỆ MÁY TÍNH .....	3
A. Thế hệ đầu tiên (1946-1957) .....	4
B. Thế hệ thứ hai (1958-1964) .....	5
C Thế hệ thứ ba (1965-1971) .....	5
D Thế hệ thứ tư (1972-????).....	5
E. Khuynh hướng hiện tại.....	6
III. PHÂN LOẠI MÁY TÍNH .....	8
A Các siêu máy tính (Super Computer): .....	8
B Các máy tính lớn (Mainframe).....	8
C Máy tính mini (Minicomputer).....	8
D.Máy vi tính (Microcomputer).....	8
IV. THÀNH QUẢ CỦA MÁY TÍNH.....	8
<b>CÂU HỎI ÔN TẬP CHƯƠNG I</b> .....	<b>22</b>
<b>Chương II: BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH</b> .....	<b>23</b>
I. THÔNG TIN VÀ SỰ MÃ HOÁ THÔNG TIN .....	23
A. Khái niệm thông tin .....	23
B Lượng thông tin và sự mã hoá thông tin.....	23
II. HỆ ĐẾM .....	24
A. Hệ đếm nhị phân và thập phân .....	<b>Error! Bookmark not defined.</b>
B Chuyển đổi giá trị giữa hệ nhị phân và hệ thập phân.....	24
1 Hệ thập phân sang nhị phân: .....	24
2 Chuyển đổi số thập phân lẻ sang nhị phân: .....	25
3. Hệ nhị phân sang thập phân: .....	26
4. Số nhị phân biểu thị số lẻ.....	26
C Các hệ đếm bát phân (8) và thập lục phân (16) .....	26
D Mã BCD (Binary Coded Decimal) .....	27
E Chuyển đổi giữa hệ thập phân và hệ Hex. ....	28
1 Hệ thập phân sang hệ hex .....	28
2. Hệ Hex sang hệ thập phân.....	29
III. BIỂU DIỄN SỐ NGUYÊN VÀ SỐ THỰC TRONG HỆ NHỊ PHÂN.....	30
A Số nguyên có dấu:.....	30
1 Dấu & trị tuyệt đối:.....	30
2 Mã bù 1:.....	30
3 Mã bù 2:.....	30
B Biểu diễn số thực .....	31
IV. CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN VÀ THẬP LỤC PHÂN .....	32
A.Cộng và trừ nhị phân .....	32

B.Cộng và trừ số hex: .....	34
C.Cộng và trừ các số BCD.....	34
D.Phép nhân và chia số nhị phân.....	<b>Error! Bookmark not defined.</b>
E.Phép tính đối với số thực floating-point.....	36
V. Biểu diễn các ký tự.....	36
A.Mã ASCII.....	36
B.Unicode .....	37
<b>Chương 3 BỘ XỬ LÝ TRUNG TÂM (CPU)</b> .....	39
I. MÔ TẢ:.....	39
A.Bộ điều khiển (CU-Control Unit) .....	40
B Các thanh ghi (Registers) .....	41
C Bộ số học-logic (ALU-Arithmetic Logic Unit).....	42
II. DIỄN TIẾN THI HÀNH LỆNH MÃ MÁY .....	42
III. NGẮT QUÃNG (INTERRUPT).....	46
IV. CÁC KIẾN TRÚC XỬ LÝ TỐC ĐỘ CỦA BỘ VI XỬ LÝ .....	47
V. CÁC KIỂU THI HÀNH MỘT LỆNH.....	48
VI. TẬP LỆNH.....	49
A.Các thành phần của lệnh máy: .....	49
B Một số lệnh thường dùng của bộ vi xử lý 8086.....	49
VII. KIỂU ĐỊNH VỊ.....	50
A.Các kiểu định vị: .....	50
B.Ví dụ Kiểu định vị của một bộ xử lý có kiến trúc phần mềm kiểu thanh ghi đa dụng.....	52
VIII. NGÔN NGỮ CẤP CAO VÀ NGÔN NGỮ MÁY .....	52
<b>CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG III</b> .....	53
<b>Chương IV: BỘ NHỚ BÁN DẪN (BỘ NHỚ TRONG)</b> .....	54
I. ĐẶC ĐIỂM VÀ PHÂN CẤP CỦA HỆ THỐNG NHỚ TRONG MÁY TÍNH.....	54
A. ĐẶC ĐIỂM: .....	54
B. PHÂN CẤP BỘ NHỚ.....	54
II. CÁC LOẠI BỘ NHỚ TRONG .....	55
A. BỘ NHỚ TRUY CẬP NGẪU NHIÊN .....	52
B BỘ NHỚ CHỈ ĐỌC .....	53
C. CACHE VÀ VẬN HÀNH CỦA CACHE .....	57
III. BỘ NHỚ ẢO .....	58
<b>Chương V: BỘ NHỚ NGOÀI</b> .....	60
I. ĐĨA TỪ .....	60
II. ĐĨA QUANG.....	62
III. CÁC LOẠI THẺ NHỚ .....	64
IV. BĂNG TỪ .....	64
V. HỆ THỐNG ĐĨA DỰ PHÒNG (RAID) .....	65
<b>CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG V</b> .....	68
<b>Chương VI: NHẬP XUẤT</b> .....	69
I. BUS NỐI NGOẠI VI VÀO BỘ XỬ LÝ VÀ BỘ NHỚ TRONG.....	69
II. CÁC CHUẨN VỀ BUS.....	70
III. GIAO DIỆN GIỮA BỘ XỬ LÝ VỚI CÁC BỘ PHẬN VÀO RA .....	70
<b>CÂU HỎI ÔN TẬP VÀ BÀI TẬP CHƯƠNG VI</b> .....	74