

GIỚI THIỆU CHUNG

1.1. Máy tính và các thành phần cơ bản

1.1.1. Một số khái niệm

1.1.1.1. Máy tính

Máy tính còn được gọi là máy vi tính hay máy điện toán, là những thiết bị hay hệ thống dùng để tính toán hay kiểm soát các hoạt động mà có thể biểu diễn dưới dạng số hay quy luật logic.

Máy tính được lắp ghép bởi các thành phần có thể thực hiện các chức năng đơn giản đã định nghĩa trước. Quá trình tác động tương hỗ phức tạp của các thành phần này tạo cho máy tính một khả năng xử lý thông tin. Nếu được thiết lập chính xác (thông thường bởi các chương trình máy tính) máy tính có thể mô phỏng lại một số khía cạnh của một vấn đề hay của một hệ thống. Trong trường hợp này, khi được cung cấp một bộ dữ liệu thích hợp nó có thể tự động giải quyết vấn đề hay dự đoán trước sự thay đổi của hệ thống.

Từ “máy tính” (*Computers*), đầu tiên, được dùng cho những người tính toán số học, có hoặc không có sự trợ giúp của máy móc, nhưng hiện nay nó hoàn toàn có nghĩa là một loại máy móc. Đầu tiên máy tính chỉ giải các bài toán số học, nhưng máy tính hiện đại làm được nhiều hơn thế. Máy tính có thể mua ở Anh đầu tiên là máy Ferranti Mark 1 Star sản xuất năm 1951.

Đến những năm 1990, khái niệm máy tính đã thực sự tách rời khỏi khái niệm điện toán và trở thành một ngành khoa học riêng biệt với nhiều lĩnh vực đa dạng và khái niệm hơn hẳn ngành điện toán thông thường và được gọi là công nghệ thông tin. Tuy vậy đến ngày nay, một số người vẫn còn nhầm lẫn giữa hai khái niệm điện toán và công nghệ thông tin.

Máy tính có thể làm việc thông qua sự chuyển động của các bộ phận cơ khí, điện tử (*Electron*), photon, hạt lượng tử hay các hiện tượng vật lý khác đã biết. Mặc dù máy tính được xây dựng từ nhiều công nghệ khác nhau song gần như tất cả các máy tính hiện nay đều là máy tính điện tử.

Máy tính có thể trực tiếp mô hình hóa các vấn đề cần được giải quyết, trong khả năng của nó các vấn đề cần được giải quyết sẽ được mô phỏng gần giống nhất với những hiện tượng vật lý đang khai thác. Ví dụ, dòng chuyển động của các điện tử có thể được sử dụng để mô hình hóa sự chuyển động của nước trong đập. Những chiếc máy tính tương tự (*Analog computer*) giống như thế đã rất phổ biến trong thập niên 1960 nhưng hiện nay còn rất ít.

Trong phần lớn các máy tính ngày nay, trước hết, mọi vấn đề sẽ được chuyển thành các yếu tố toán học bằng cách diễn tả mọi thông tin liên quan thành các số theo hệ nhị phân (hệ thống đếm dựa trên các số 0 và 1 hay còn gọi là hệ đếm cơ số 2). Sau đó, mọi tính toán trên các thông tin này được tính toán bằng đại số Bool (*Boolean algebra*).

Các mạch điện tử được sử dụng để mô tả các phép tính Bool. Vì phần lớn các phép tính toán học có thể chuyển thành các phép tính Bool nên máy tính điện tử đủ nhanh để xử lý phần lớn các vấn đề toán học (phần lớn thông tin của vấn đề cần giải quyết đã được chuyển thành các vấn đề toán học). Ý tưởng cơ bản này, được nhận biết và nghiên cứu bởi Claude E. Shannon - người đã làm cho máy tính kỹ thuật số (*Digital computer*) hiện đại trở thành hiện thực.

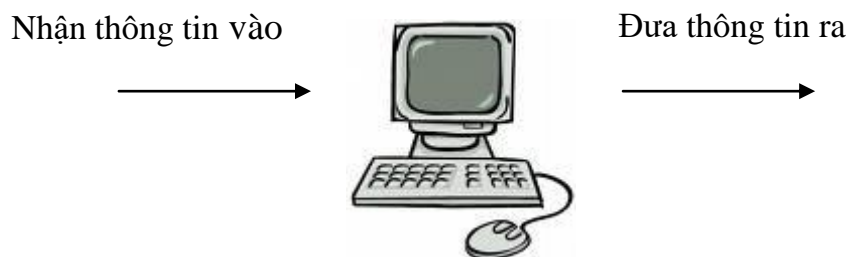
Máy tính không thể giải quyết tất cả mọi vấn đề của toán học. Ông Alan Turing đã sáng tạo ra khoa học lý thuyết máy tính trong đó đề cập tới những vấn đề mà máy tính có thể hay không thể giải quyết. Khi máy tính kết thúc tính toán một vấn đề, kết

quả của nó được hiển thị cho người sử dụng thấy thông qua thiết bị xuất như: bóng đèn, màn hình, máy in...

Những người mới sử dụng máy tính, đặc biệt là trẻ em, thường cảm thấy khó hiểu về ý tưởng cơ bản là máy tính chỉ là một cái máy, nó không thể “suy nghĩ” hay “hiểu” những gì nó hiển thị. Máy tính chỉ đơn giản thi hành các tìm kiếm cơ khí trên các bảng màu và đường thẳng đã lập trình trước, rồi sau đó thông qua các thiết bị đầu ra (màn hình, máy in,...) chuyển đổi chúng thành những ký hiệu mà con người có thể cảm nhận được thông qua các giác quan (hình ảnh trên màn hình, chữ trên văn bản được in ra). Chỉ có bộ não của con người mới nhận thức được những ký hiệu này tạo thành các chữ hay số và gán ý nghĩa cho chúng. Trong quan điểm của máy tính thì mọi thứ mà nó “nhận thấy” (kể cả khi máy tính được coi là có khả năng tự nhận biết) chỉ là các hạt electron tương đương với các số 0 và 1.

Như vậy, máy tính là một thiết bị có khả năng thao tác (lưu trữ, xử lý) trên dữ liệu (thông tin) theo một cách phức tạp và lập trình được. Việc tính toán của nó thực hiện theo một chương trình. Dữ liệu có thể được biểu diễn dưới rất nhiều hình thức của thông tin như: số, ký tự, hình ảnh, âm thanh,...

Xử lý thông tin theo chương
trình được lưu trong bộ nhớ



Hình 1-1. Mô tả chức năng của máy tính

1.1.1.2. Chương trình (Program)

Chương trình là dãy các câu lệnh trong bộ nhớ, nhằm mục đích hướng dẫn máy tính thực hiện một công việc cụ thể nào đây.

1.1.1.3. Phần mềm (Software)

Bao gồm các thuật toán và các biểu diễn cho máy tính của chúng ta, đó chính là các chương trình. Chương trình có thể được biểu diễn (lưu trữ) trên bìa đục lỗ, băng từ, đĩa từ,... hay các môi trường khác, tuy nhiên cái cơ bản nhất của phần mềm chính là tập hợp các câu lệnh (chỉ thị) tạo nên chương trình chứ không phải là môi trường vật lý được sử dụng để ghi (lưu trữ) chương trình.

1.1.1.4. Phần cứng (Hardware)

Chương trình được viết bằng ngôn ngữ máy ở mức 1 có thể được thi hành trực tiếp bởi các mạch điện mà không cần một trình thông dịch hoặc trình biên dịch trung gian nào cả. Các mạch điện như vậy cùng với bộ nhớ và các thiết bị ngoại vi (vào/ra) tạo thành phần cứng máy của tính (hardware). Phần cứng bao gồm các đối tượng hữu hình như các vi mạch (IC), các bảng (board) mạch in, cáp nối, nguồn điện, bộ nhớ, máy đọc bìa, máy in, terminal,..., chứ không phải là các ý tưởng, các thuật toán hay các câu lệnh (chỉ thị).

1.1.1.5. Phần sụn (Firmware):

Phần sụn (hay còn gọi là phần dẻo) là dạng trung gian giữa phần cứng và phần mềm, nó là phần mềm được nhúng vào các mạch điện tử trong quá trình chế tạo ra các

mạch điện tử này. Firmware được sử dụng khi các chương trình hiếm khi hoặc không bao giờ cần thay đổi.

Một ví dụ trực quan cho phần sụn chính là ROM BIOS chứa các chương trình khởi động, các dịch vụ vào/ra cơ sở, dữ liệu về cấu hình của hệ thống mà chúng đã được tối ưu, hoàn chỉnh không cần phải thay đổi nữa (ít thay đổi). Hay các phần mềm trong đồ chơi hoặc trong các dụng cụ máy móc, điện thoại di động, ...

Nói chung một thao tác được thực hiện bằng phần mềm thì cũng có thể xây dựng phần cứng để thực hiện trực tiếp thao tác đó, ngược lại mọi thao tác (các lệnh, chỉ thị) được thực hiện bằng phần cứng thì cũng có thể mô phỏng bằng phần mềm.

Việc quyết định đưa những chức năng nhất định nào vào phần cứng và các chức năng nào vào phần mềm được dựa trên các yếu tố như giá cả, tốc độ, độ tin cậy và tần suất của sự thay đổi có thể xảy ra. Không có những quy tắc bắt buộc quy định một cách rõ ràng rằng phải đưa thao tác X này vào trong phần cứng, còn thao tác Y kia phải được thực hiện bằng phần mềm (được lập trình). Những người thiết kế máy tính khác nhau, với những mục tiêu khác nhau có thể thường quyết định khác nhau về vấn đề này.

1.1.2. Các lĩnh vực ứng dụng của máy tính

Trong ngành khoa học và trong đời sống, có thể thấy máy vi tính nói chung được ứng dụng trong 3 lĩnh vực:

1.1.2.1. Máy vi tính dùng để xử lý dữ liệu

Thuộc lớp các bài toán quản lý, tin học văn phòng, xử lý đồ họa. Mục đích của các chương trình này là sử dụng dữ liệu là các con số, ký tự, điểm ảnh nhập vào máy vi tính, xử lý chúng và sử dụng kết quả tính toán theo cách mong muốn. Các kết quả đó có thể được thể hiện lên màn hình, in ra giấy hoặc được lưu trữ dưới dạng file trong bộ xử lý dữ liệu tùy người chạy chương trình.

Các máy tính dùng trong bộ xử lý dữ liệu thường được đánh giá trên cơ sở bộ nhớ ngoài lớn bao nhiêu và các máy in cũng như màn hình được tăng cường chất lượng như thế nào. Lĩnh vực xử lý dữ liệu hiện đang được rất nhiều người quan tâm và sử dụng.

1.1.2.2. Máy vi tính dùng để xử lý số

Trong chức năng xử lý dữ liệu, máy tính dành nhiều thời gian cho việc xử lý các dữ liệu dưới dạng các ký hiệu biểu diễn cho các thông tin. Ví dụ, thông tin về các điểm ảnh trong một bức ảnh. Lúc này, thời gian cần để tính toán các phép tính số là không quan trọng so với thời gian xử lý dữ liệu. Ngược lại, có khi máy tính lại được sử dụng trong các ứng dụng liên quan tới một lượng rất lớn các phép toán số học, đôi khi lên tới hàng tỉ phép toán cho một công việc.

Chương trình của những bài toán thuộc loại này đòi hỏi phải được chạy trong thời gian rất dài trên những máy vi tính lớn hay siêu máy tính. Gần đây, đã có một xu hướng kết nối nhiều máy vi tính để tạo thành một mạng lưới tính toán song song để tận dụng hiệu suất các máy vi tính trên diện rộng nhằm giải quyết các bài toán xử lý số lớn mà không cần có các máy vi tính lớn.

1.1.2.3. Máy vi tính dùng trong điều khiển và đo lường tự động

Do giá thành ngày càng rẻ, máy vi tính còn được đặt tại các trung tâm của nhiều hệ thống điều khiển tự động. Có khi máy tính còn được kết nối để điều khiển với các thiết bị bên ngoài thông qua các cổng vào ra (I/O), một số lại được gắn vào bên trong một hệ thống lớn để điều khiển nó.

Ví dụ

Một máy vi tính on-chip được gắn trong một máy giặt tự động để tính toán thời gian cần thiết cho các quá trình làm sạch cũng như điều khiển tốc độ quay của động cơ quay vắt nước quần áo hiện nay đã thay thế cho các cơ cấu định thời bằng cơ khí của những năm trước đây. Các hệ thống điều khiển như vậy có mặt ở khắp nơi, từ hệ thống lớn như hệ thống điều khiển hạ cánh tự động trên máy bay đến các hệ thống điều khiển trên điện thoại di động, các thiết bị thu hình, các thiết bị điện tử gia dụng.

1.1.3. Lịch sử phát triển máy tính

Khi nói về nguồn gốc xa xưa của máy tính điện tử, người ta vẫn thường nhắc tới chiếc bàn tính gậy tay được phát minh vào khoảng 3000 năm trước công nguyên tại thành phố Babylon, nằm bên bờ sông Euphrat phía nam thành phố Batda - Irak. Sau này, cùng với sự phát triển của khoa học con người đã phát minh ra hàng loạt các phương tiện và công cụ tính toán khác nhau.

Tuy vậy, mãi cho đến thế kỷ thứ 17, chiếc máy tính cơ học đầu tiên thực hiện các phép tính cộng và trừ mới được chế tạo thành công. Người có công chế tạo chiếc máy tính đó là nhà bác học người Pháp Blaise Pascal (1623 – 1662). Chiếc máy tính này được ông chế tạo lúc mới 19 tuổi nhằm giúp đỡ công việc của bố mình. Tên của ông sau này được đặt cho một ngôn ngữ lập trình có cấu trúc nổi tiếng – Ngôn ngữ Pascal.

Nền móng lý thuyết quan trọng đặt tiền đề cho việc xây dựng máy tính điện tử sau này chính là công trình của nhà toán học Geogre Boole “Phân tích toán học của phép logic-số” được phát minh vào năm 1847. Điều thú vị là phát minh này hầu như không được quan tâm trong suốt thế kỷ thứ 19 song nó lại là cơ sở lý thuyết hết sức quan trọng cho hàng loạt các ứng dụng gần 100 năm sau. Phương pháp logic toán học của ông sau này đã trở thành một ngành khoa học có tên gọi là Đại số bool hay logic bool.

Năm 1930 chiếc máy tính tương tự đầu tiên được nhà bác học người Mỹ Vanevar Bush xây dựng và đến năm 1944 giáo sư Howad Aiken của hãng IBM hãng hàng đầu thế giới về sản xuất và hoàn thiện máy tính.

Song năm khai sinh ra chiếc máy tính điện tử hiện đại đầu tiên trên thế giới được ghi nhận là vào năm 1946 tại trung tâm nghiên cứu đường đạn thuộc trường Đại học Pennsylvania của Mỹ. Người có ý tưởng xây dựng chiếc máy tính điện tử đầu tiên này là John Mauchley và học trò của ông là J.Presper Eckert. Năm 1943 ông đã thuyết phục được quân đội Mỹ cấp tiền cho việc thiết kế chế tạo chiếc máy tính điện tử đầu tiên này. Máy tính này có tên gọi là thiết bị tổng hợp phân tích và tính toán bằng kỹ thuật điện tử số **ENIAC – Electronic Numerical Integrate Analizer and Computer**.

Máy tính ENIAC có 18.000 bóng đèn điện tử, 1.500 rơ le, nặng 30 tấn, tiêu thụ 174 kw điện và chiếm trọn một gian phòng rộng 200 m². Về cấu trúc, máy có 20 thanh ghi, mỗi thanh ghi chứa 1 số thập phân 10 chữ số.

Sau này, thầy trò John Mauchley và J.Presper Eckert đã thành lập một công ty máy tính riêng mang tên hai ông, công ty Eckert-Mauchly, công ty này chính là tiền thân của công ty máy tính nổi tiếng Unisys ngày nay.

Sau thế hệ máy tính điện tử ENIAC, đã có rất nhiều nhà khoa học tập trung nghiên cứu, thiết kế và xây dựng các máy tính điện tử theo ý tưởng của mình, trong đó có John Von Neumann (1903 – 1957) một thiên tài người Đức. Năm 23 tuổi, ông tốt nghiệp trường Đại học Berlin về hóa học đồng thời ông cũng nhận được bằng tiến sĩ toán và sau này ông trở thành một nhà vật lý học, toán học nổi tiếng. Ông nói thông

thạo nhiều ngôn ngữ khác nhau trên thế giới, có khả năng trích dẫn đúng nguyên văn từ trong trí nhớ những đoạn sách mà ông đã đọc từ nhiều năm trước.

Khi xây dựng máy tính của mình ông là người đầu tiên đề xuất ý tưởng chương trình máy tính có thể được biểu diễn dưới dạng số và đặt vào trong bộ nhớ của máy tính bên cạnh dữ liệu. Cũng chính ông đề nghị sử dụng số nhị phân thay cho số thập phân đã được sử dụng trong máy tính ENIAC. Máy tính được xây dựng trên những nguyên lý của ông đề xuất được gọi là máy Von Neumann và đó cũng là cơ sở quan trọng cho các thế hệ máy tính điện tử số sau này.

1.2. Phân loại máy tính

Các thiết bị tính toán tăng gấp đôi năng lực (được định nghĩa là số phép tính thực hiện trong một giây cho mỗi 1.000 USD chi phí) sau mỗi 18 đến 24 tháng kể từ năm 1900. Gordon E. Moore, người đồng sáng lập ra Intel, lần đầu tiên đã miêu tả tính chất này của sự phát triển vào năm 1965. Cùng với việc tăng khả năng tính toán trên một đơn vị chi phí thì tốc độ của sự thu nhỏ kích thước cũng tương tự. Những chiếc máy tính điện tử đầu tiên như ENIAC (ra đời năm 1946) là một thiết bị khổng lồ nặng hàng tấn, tiêu thụ nhiều điện năng, chiếm một diện tích lớn, thực hiện được ít phép tính và đòi hỏi nhiều người điều khiển để có thể hoạt động được. Những cỗ máy này đắt đến mức chỉ có các chính phủ hay các viện nghiên cứu lớn mới có đủ điều kiện để duy trì hoạt động của chúng. Ngược lại, các máy tính ngày nay có nhiều sức mạnh hơn, rẻ tiền hơn, có kích thước nhỏ hơn, tiêu thụ ít điện năng hơn và phổ biến ở mọi nơi. Hiện nay máy tính được phân loại theo một số tính năng cơ bản sau:

1.2.1. Phân loại theo công nghệ

Cho đến nay, nếu xét theo sự phát triển của công nghệ thì máy tính điện tử có thể được phân thành 5 thế hệ khác nhau và mỗi thế hệ được đặc trưng bởi trình độ công nghệ:

1.2.1.1. Thế hệ thứ nhất (1940 – 1950)

Các máy tính ở thế hệ này được xây dựng từ các bóng đèn điện tử chân không (Vacuum tube, còn được gọi tắt là tube hay valve). Đặc trưng của các máy tính ở thế hệ này là:

- Sử dụng thiết bị phóng electron.
- Dữ liệu và chương trình được lưu trữ trong bộ nhớ đọc/ghi đơn giản.
- Truy cập bộ nhớ theo địa chỉ.
- Ngôn ngữ máy hoặc ngôn ngữ Assemble.
- Thực thi lệnh tuần tự.
- Năng lực tính toán chậm, tiêu thụ điện năng lớn và chiếm chiều diện tích.

1.2.1.2. Thế hệ thứ hai (1950 – 1964)

Đặc trưng cơ bản của thế hệ này là sử dụng Transistor để làm giảm kích thước của máy tính, xây dựng bộ xử lý trung tâm và các mạch điện của máy tính, nhờ vậy tăng đáng kể tốc độ thực hiện công việc và độ tin cậy của máy tính cũng được cải thiện.

- Hệ điều hành ra đời. Nạp và thi hành lần lượt từng chương trình.
- Thực hiện chuyển mạch bật tắt bằng điều khiển.
- Sử dụng ngôn ngữ lập trình bậc cao.
- Thực hiện phép toán với dấu chấm động.

1.2.1.3. Thế hệ thứ ba (1964 – 1974)

Thế hệ này gắn liền với sự xuất hiện và ứng dụng các vi mạch điện tử IC (Integrated Circuit – IC). Các chip vi xử lý tổ hợp bởi hàng ngàn transistor, toàn bộ

mạch được thiết kế nằm trên một bản mạch máy tính, máy tính đã có kích thước nhỏ hơn, khả năng tính toán nhanh hơn.

- Sử dụng bộ nhớ bán dẫn.
- Nhiều mô hình máy tính với các đặc trưng thực thi khác nhau.
- Máy tính nhỏ hơn và không cần các phòng chứa đặc biệt.

1.2.1.4. Thế hệ thứ tư (1974 đến nay)

Thế hệ thứ tư được đánh dấu bằng các IC có mật độ tích hợp cao (Large Scale Integration - LSI) có thể chứa hàng ngàn linh kiện. Các IC mật độ tích hợp rất cao (Very Large Scale Integration - VLSI) 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.

Hai thành tựu tiêu biểu về công nghệ của máy tính ở thế hệ thứ 4 là bộ nhớ bán dẫn và bộ vi xử lý:

a. Bộ nhớ bán dẫn

Vào khoảng những năm 50 đến năm 60 của thế kỷ này, hầu hết bộ nhớ máy tính đều được chế tạo từ những vòng nhỏ làm bằng vật liệu sắt từ, mỗi vòng có đường kính khoảng 1/16 inch. Các vòng này được treo trên các lưới ở trên những màn nhỏ bên trong máy tính. Khi được từ hóa theo một chiều, một vòng (gọi là một lõi) biểu thị giá trị 1, còn khi được từ hóa theo chiều ngược lại, lõi sẽ đại diện cho giá trị 0. Bộ nhớ lõi từ kiểu này làm việc khá nhanh. Nó chỉ cần một phần triệu giây để đọc một bit lưu trong bộ nhớ. Những nó rất đắt tiền, cồng kềnh và sử dụng cơ chế hoạt động loại trừ. Một thao tác đơn giản như đọc một lõi sẽ xóa dữ liệu lưu trong lõi đó. Do vậy cần phải cài đặt các mạch phục hồi dữ liệu ngay khi nó được lấy ra ngoài.

Năm 1970, Fairchild chế tạo ra một bộ nhớ bán dẫn có dung lượng tương đương đầu tiên. Chip này có kích thước bằng một lõi đơn, có thể lưu 256 bit nhớ, hoạt động không theo cơ chế loại trừ và nhanh hơn bộ nhớ lõi từ. Nó chỉ cần 70 phần tỉ giây để đọc ra một bit dữ liệu trong bộ nhớ. Tuy nhiên giá thành cho mỗi bit cao hơn so với lõi từ.

Kể từ năm 1970, bộ nhớ bán dẫn đã đi qua 8 thế hệ: 1K, 4K, 16K, 64K, 156K, 1M, 4M và bây giờ là 16M bit trên một chip đơn ($1K = 2^{10}$, $1M = 2^{20}$). Mỗi thế hệ cung cấp khả năng lưu trữ nhiều gấp bốn lần so với thế hệ trước, cùng với sự giảm thiểu giá thành trên mỗi bit và thời gian truy cập.

b. Bộ vi xử lý

Vào năm 1971, hãng Intel cho ra đời chip 4004, chip đầu tiên có chứa tất cả mọi thành phần của một CPU trên một chip đơn. Kỷ nguyên bộ vi xử lý đã được khai sinh từ đó. Chip 4004 có thể cộng hai số 4 bit và nhân bằng cách lập lại phép cộng. Theo tiêu chuẩn ngày nay, chip 4004 rõ ràng là quá đơn giản, nhưng nó đã đánh dấu sự bắt đầu một quá trình tiến hóa liên tục về dung lượng và sức mạnh của một bộ vi xử lý. Bước chuyển biến kế tiếp trong quá trình tiến hóa nói trên là sự giới thiệu chip Intel 8008 vào năm 1972. Đây là bộ vi xử lý 8 bit đầu tiên và có độ phức tạp gấp đôi chip 4004.

Đến năm 1974, Intel đưa ra chip 8080, bộ vi xử lý đa dụng đầu tiên được thiết kế để trở thành CPU của một máy vi tính đa dụng. So với chip 8008, chip 8080 nhanh

hơn, có tập chỉ thị phong phú hơn và có khả năng định địa chỉ lớn hơn. Cũng trong cùng thời gian đó, các bộ vi xử lý 16 bit đã bắt đầu được phát triển. Mặc dù vậy, mãi đến cuối những năm 70, các bộ vi xử lý 16 bit đa dụng mới xuất hiện trên thị trường. Sau đó đến năm 1981, cả Bell lab và Hewlett-Packard đều đã phát triển các bộ vi xử lý đơn chip 32 bit. Trong đó, Intel giới thiệu bộ vi xử lý 32 bit của riêng mình là chip 80386 vào năm 1985.

1.2.1.5. Thế hệ thứ năm – Khuyňh hướng hiện đại

Việc chuyển từ thế hệ thứ tư sang thế hệ thứ năm 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 năm của máy tính, thế hệ của những máy tính thông minh, dựa trên các ngôn ngữ trí tuệ nhân tạo như LISP, 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 năm 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.

1.2.2. Phân loại theo tính năng

1.2.2.1. Siêu máy tính (Super Computer)

Là loại máy tính có giá thành rất cao được thiết kế để thực hiện các phép tính phức tạp và với tốc độ tính toán cực đại mà công nghệ hiện đại cho phép.

1.2.2.2. Máy tính lớn (Computer Mainframe)

Là loại máy tính nhiều người dùng, được thiết kế để đáp ứng nhu cầu của một tổ chức lớn.

1.2.2.3. Máy tính Mini (Mini Computer)

Một loại máy tính nhiều người sử dụng, được thiết kế nhằm đáp ứng các nhu cầu cho một công ty nhỏ (khoảng từ 4 đến 100 người cùng sử dụng một lúc).

1.2.2.4. Trạm công tác hay trạm làm việc (Workstation)

Trong mạng cục bộ, đây là một máy tính loại để bàn chạy các chương ứng dụng và đóng vai trò là một điểm để thâm nhập vào mạng.

1.2.2.5. Máy tính cá nhân hay còn gọi là máy PC (Personal Computer)

Là loại máy tính được sử dụng bởi riêng một người, đây là máy tính độc lập được trang bị đầy đủ các phần mềm hệ thống, tiện ích, ứng dụng cũng như các thiết bị vào ra và các thiết bị khác.

1.2.2.6. Máy vi tính (Micro Computer)

Đề chỉ những máy tính có đơn vị xử lý trung tâm CPU (Central Processing Unit) là bộ vi xử lý (Micro Processor). Máy vi tính xuất hiện lần đầu tiên vào cuối những năm 1970.

Hiện nay sự phân biệt giữa máy tính Mini (nhiều người dùng) với máy tính PC (một người dùng) không rõ ràng.

1.2.3. Phân loại theo kích thước

Tùy thuộc vào kích thước các máy tính hiện nay có thể phân loại như sau:

- + Máy tính lớn (Mainframe).
- + Máy tính để bàn (Desktop).
- + Máy tính xách tay (Laptop).
- + Máy tính kiểu sổ tay (Notebook).

1.2.4. Phân loại theo bộ xử lý

+ Máy tính sử dụng bộ vi xử lý của hãng Intel như: 8088, 80286, 80386, 80486, Pentium II, III, IV, Dual Core, Core 2... Các máy sử dụng bộ vi xử lý của Intel điển hình như: IBM, Dell, Gateway, DNA, Compaq,...

+ Các máy tính sử dụng bộ vi xử lý Motorola như: Admiga, Machintosh...

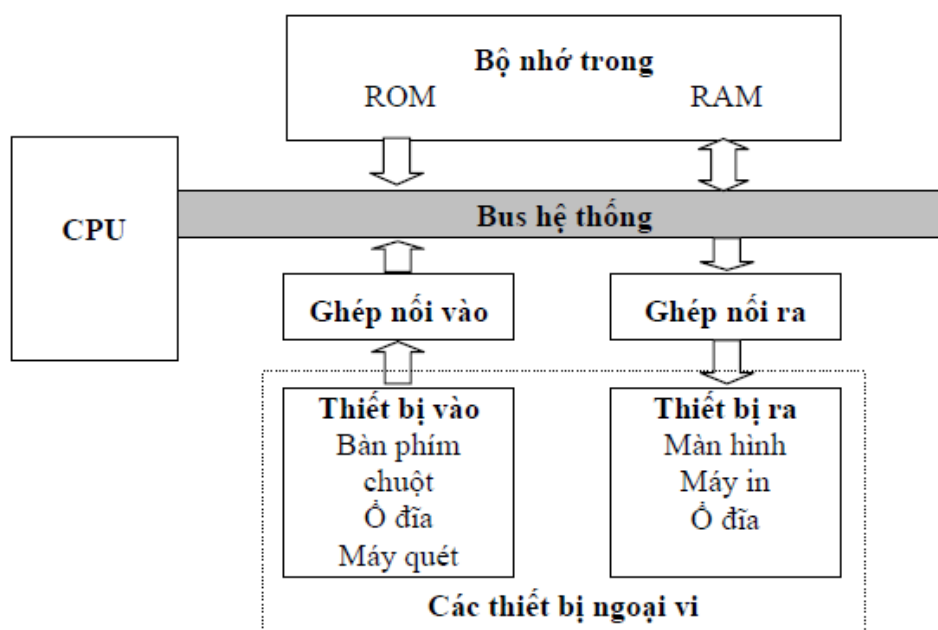
+ Các máy tính xây dựng dựa trên bộ vi xử lý của hãng Zilog như: Z40, Z80, Bondwell, NEC, Sony...

1.3. Cấu trúc chung của hệ thống máy tính IBM-PC

1.3.1. Sơ đồ chức năng

Hình 1-2 minh họa sơ đồ khối chức năng của một hệ thống máy tính. Theo đó, hệ thống máy tính gồm 4 thành phần chính:

1. Đơn vị xử lý trung tâm (Central Processing Unit – CPU)
2. Bộ nhớ trong (Internal Memory)
3. Hệ thống vào ra (Input/Output System – I/O System)
4. Bus liên kết hệ thống (System Bus)



Hình 1-2. Sơ đồ khối cấu trúc chung của hệ thống máy tính

1.3.2. Các thành phần của máy tính

1.3.2.1. Đơn vị xử lý trung tâm

Đơn vị xử lý trung tâm (Central Processing Unit - CPU) là thành phần quan trọng nhất được xem là bộ não của máy tính. Nó có chức năng xử lý dữ liệu và điều khiển hoạt động của hệ thống.

CPU hoạt động theo xung nhịp, nó nạp lần lượt từng lệnh từ bộ nhớ, giải mã lệnh để phát tín hiệu điều khiển thực hiện lệnh. Việc thực hiện chương trình thực chất là sự lặp lại quá trình nạp lệnh, giải mã lệnh và thực thi lệnh được nạp.

Mỗi lệnh được thực hiện trong một chu kỳ lệnh. Chu kỳ lệnh không nhất thiết phải bằng nhau, mỗi chu kỳ lệnh xảy ra trên một hoặc nhiều xung nhịp đồng hồ hệ thống.

Mỗi lệnh được thực hiện theo nhiều giai đoạn, phụ thuộc vào loại lệnh. Một lệnh thường trải qua 5 giai đoạn.

Các yêu cầu của hệ thống và của người sử dụng thường được biểu diễn thành các chương trình máy tính, trong đó mỗi chương trình thường được tạo thành từ nhiều lệnh của CPU. CPU đảm nhiệm việc đọc các lệnh của chương trình từ bộ nhớ, giải mã và thực hiện lệnh. Thông qua việc CPU thực hiện các lệnh của chương trình, máy tính có khả năng cung cấp các tính năng hữu ích cho người sử dụng.

CPU là vi mạch tích hợp với mật độ rất cao, được cấu thành từ 4 thành phần con:

1. Đơn vị điều khiển (Control Unit - CU): Điều khiển hoạt động của CPU và các thành phần khác của máy tính.

2. Đơn vị số học và logic (Arithmetic and Logic Unit - ALU): Thực hiện các chức năng xử lý dữ liệu.

3. Các thanh ghi (Registers): Là các ngăn nhớ đặc biệt nằm trong CPU để chứa các thông tin tạm thời phục vụ cho quá trình thực hiện chương trình.

4. Bus bên trong CPU (Internal Bus): Dùng để kết nối và trao đổi thông tin giữa các thành phần với nhau.

Ngoài 4 thành phần nói trên, trong CPU còn có bộ nhớ cache. Đây là bộ nhớ đệm có tốc độ truy nhập cao, giúp vào ra dữ liệu nhanh hơn, tăng hiệu quả làm việc của CPU.

1.3.2.2. Bộ nhớ trong

Bộ nhớ trong được chế tạo từ các chip nhớ bán dẫn, bao gồm 2 loại:

- + Bộ nhớ BIOS ROM (Basic Input Output System Read Only Memory): Là bộ nhớ ROM, chứa chương trình khởi động đầu tiên của máy tính và chương trình điều khiển vào ra cơ bản trong một hệ thống máy tính và nó chứa thông tin cố định trong hệ thống. Đặc trưng cơ bản của bộ nhớ ROM ngày nay là mất nguồn điện nuôi thì không mất thông tin. Do đó nó thường được dùng để chứa thông tin cố định trong hệ thống hoặc làm bộ nhớ ngoài lưu trữ thông tin ổn định lâu dài.

- + Bộ nhớ chính (Main Memory): Là bộ nhớ RAM (Random Access Memory), đây là thành phần được nối trực tiếp với CPU và được điều khiển bởi CPU. Các chương trình đang thực hiện phải nằm trong bộ nhớ chính. Bộ nhớ chính bao gồm các ngăn nhớ và mỗi ngăn nhớ có một địa chỉ xác định, các ngăn nhớ được tổ chức theo byte. Bộ nhớ chính thường có tốc độ cao, dung lượng nhỏ và chứa các chương trình đang thực thi, các dữ liệu đang cần được xử lý trong máy tính. Đặc trưng cơ bản của bộ nhớ RAM là truy cập ngẫu nhiên, tốc độ truy cập cao, có dung lượng lớn hơn nhiều so với ROM. Tuy nhiên, thông tin trong RAM chỉ tồn tại khi có nguồn điện nuôi.

1.3.2.3. Hệ thống vào ra

Các thiết bị vào ra (Input – Output devices – I/O), hay còn gọi là các thiết bị ngoại vi (Peripheral devices) đảm nhiệm việc nhập dữ liệu vào, điều khiển hệ thống và kết xuất dữ liệu ra. Các thiết bị ngoại vi làm nhiệm vụ chuyển thông tin ở dạng vật lý

nào đó về dạng dữ liệu phù hợp với máy tính và ngược lại. Do các thiết bị ngoại vi không được ghép nối trực tiếp tới CPU mà phải thông qua các mạch ghép nối vào ra. Trong mạch ghép nối vào ra có các thanh ghi được đánh địa chỉ xác định và nối trực tiếp tới các cổng vào ra. Các thiết bị vào ra được ghép nối thông qua các cổng, địa chỉ của các cổng chính là địa chỉ của các thanh ghi dữ liệu. Có 2 nhóm thiết bị ngoại vi:

a. Các thiết bị vào (Input devices)

Thực hiện biến đổi tín hiệu tự nhiên hoặc cho bởi con người sang dạng mã máy để bộ vi xử lý thực hiện. Các thiết bị vào thông dụng nhất hiện nay có thể kể như bàn phím (Keyboard), chuột (Mouse), máy quét ảnh (Scanners), thiết bị phân tích nhận dạng tiếng nói, CD-ROM, máy đọc mã vạch

b. Các thiết bị ra (Output devices)

Biến đổi các mã bên trong máy tính sau khi xử lý để con người có thể hiểu được hoặc điều khiển các thiết bị khác. Các thiết bị ra gồm: màn hình (Screen), máy in (Printers), máy vẽ (Plotters), loa, faxmodem hay các thiết bị điều khiển khác.

1.3.2.4. Bus liên kết hệ thống

a. Định nghĩa Bus

Bus là tập hợp các đường dây dẫn điện để vận chuyển thông tin, tín hiệu điện (các bit) từ phần mạch này đến các phần mạch khác trong phạm vi của máy tính. Trong các máy tính hiện đại ngày nay, có hai nhóm Bus đó là Bus hệ thống, nối giữa CPU với bộ nhớ chính và các Bus vào ra nối ghép giữa các thiết bị ngoại vi tới CPU thông qua các “cầu nối” hay còn gọi là các “cầu”. Tham gia điều khiển bus còn có các vi mạch điều khiển bus. Các vi mạch trước đây thường được nằm riêng lẻ còn hiện nay chúng được tích hợp vào bên trong các vi mạch tổng hợp chipset. Bít là từ viết tắt của “Binary digiT”. Về bản chất vật lý của bit:

- Không có điện áp → truyền 0.
- Có điện áp → truyền 1.

Tập các đường dây vận chuyển thông tin đồng thời tại một thời điểm được gọi là độ rộng của Bus. Ví dụ, có 8 đường dây vận chuyển thông tin đồng thời thì độ rộng là 8 bít.

b. Phân loại Bus

Bus được chia thành 3 loại: Bus địa chỉ; Bus dữ liệu; Bus điều khiển.

- Bus địa chỉ:

CPU muốn trao đổi dữ liệu với ngăn nhớ nào, với cổng vào ra nào thì nó cần phải xác định được địa chỉ của đối tượng cần trao đổi dữ liệu đó.

Bus địa chỉ vận chuyển tín hiệu từ CPU đến bộ nhớ hay cổng vào ra để xác định xem ngăn nhớ nào hay cổng vào ra nào cần trao đổi thông tin.

Bus địa chỉ, nói tổng quát hơn, nó gồm n đường dây từ $A_0 \div A_{n-1}$ thì gọi độ rộng Bus là n bít và n bít này được dùng để đánh địa chỉ. Do đó nó có khả năng quản lý tối đa 2^n giá trị địa chỉ ngăn nhớ hay 2^n byte nhớ (vì bộ nhớ chính quản lý theo byte).

Ví dụ

8088/8086: Bộ vi xử lý có $n = 20 \rightarrow$ quản lý tối đa 2^{20} byte = 1 MB

80286 $n = 24 \rightarrow$ quản lý tối đa 2^{24} byte = $2^4 \times 2^{20} = 16$ MB

80386 $n = 32 \rightarrow$ quản lý tối đa 2^{32} byte = $2^2 \times 2^{30} = 4$ GB

Pentium II $n = 36 \rightarrow$ quản lý tối đa 2^{36} byte = $2^6 \times 2^{30} = 64$ GB

Như vậy, bus địa chỉ có độ rộng và chỉ vận chuyển tín hiệu theo một chiều, đó là chiều từ CPU ra ngoài.

- Bus dữ liệu:

- Được dùng để vận chuyển dữ liệu giữa các thành phần khác nhau trong phạm vi máy tính như CPU, bộ nhớ, hệ thống vào ra.

- Bus dữ liệu gồm m đường dây từ $D_0 \div D_{m-1}$ thì độ rộng của bus là m bit. Trong đó m thường có giá trị là 8, 16, 20, 24, 32, 64 đường dây tùy thuộc vào từng CPU cụ thể.

Ví dụ

8088: $m = 8$, tức là vận chuyển một lúc 1 byte

8086/80286: $m = 16$, tức là vận chuyển một lúc 2 byte

80386/486: $m = 32$, tức là vận chuyển một lúc 4 byte

Pentium II: $m = 64$, tức là vận chuyển một lúc 8 byte

Bus dữ liệu là loại 2 chiều. Các phần tử có đầu ra nối thẳng với bus dữ liệu đều phải được trang bị đầu ra 3 trạng thái để đảm bảo cho bus hoạt động bình thường.

- Bus điều khiển

Là tập hợp các tín hiệu điều khiển riêng lẻ, hoặc là phát ra từ CPU để điều khiển việc đọc ghi bộ nhớ hay hệ thống vào ra, hoặc là phát ra từ bộ nhớ hay hệ thống vào ra đến yêu cầu CPU. Bus điều khiển thường gồm hàng chục dây tín hiệu khác nhau và nó là loại Bus 2 chiều.

Một số tín hiệu điển hình của Bus điều khiển:

- Các tín hiệu phát ra từ CPU điều khiển ghi, đọc bộ nhớ hay cổng vào ra

Có 4 tín hiệu cơ bản:

+ *Memory Read (MEMR)*:

Là tín hiệu điều khiển đọc dữ liệu từ một ngăn nhớ có địa chỉ xác định đưa lên Bus dữ liệu. Khi Bus địa chỉ xác định ngăn nhớ cần đọc thì MEMR sẽ điều khiển mở ngăn nhớ để đưa dữ liệu lên Bus dữ liệu.

+ *Memory Write (MEMW)*:

Điều khiển việc ghi dữ liệu có sẵn trên Bus dữ liệu đến ngăn nhớ có địa chỉ xác định. Khi Bus địa chỉ xác định ngăn nhớ cần ghi và dữ liệu trên Bus dữ liệu đã ổn định thì MEMW sẽ điều khiển mở ngăn nhớ để đưa dữ liệu từ Bus dữ liệu vào ngăn nhớ.

+ *Input Output Read (IOR)*

Là tín hiệu điều khiển đọc dữ liệu từ cổng.

+ *Input Output Write (IOW)*

Là tín hiệu điều khiển ghi dữ liệu ra cổng.

- Các tín hiệu điều khiển ngắt

Là các tín hiệu yêu cầu CPU dừng công việc hiện tại để chuyển sang thực hiện công việc khác, còn gọi là yêu cầu CPU ngắt. Nó có các dạng sau:

+ *Non Maskable Interrupt (NMI)*:

Đây là tín hiệu ngắt không che được. Nó là tín hiệu từ mạch bên ngoài gửi đến chân NMI của CPU và CPU phải ngắt ngay sau khi hoàn thành các lệnh đang thực hiện.

+ *Interrupt Reques (INTR)*:

Đây là tín hiệu ngắt che được. Nó là tín hiệu được gửi từ các mạch điều khiển ưu tiên ngắt gửi đến các chân INTR của CPU và yêu cầu CPU ngắt, CPU có thể ngắt hoặc không ngắt.

+ *Interrupt Acknowledge (INTA)*:

Là tín hiệu phát ra từ chân INTA của CPU, báo cho mạch bên ngoài biết CPU có chấp nhận ngắt hay không.

- Các tín hiệu điều khiển việc chuyển nhượng Bus

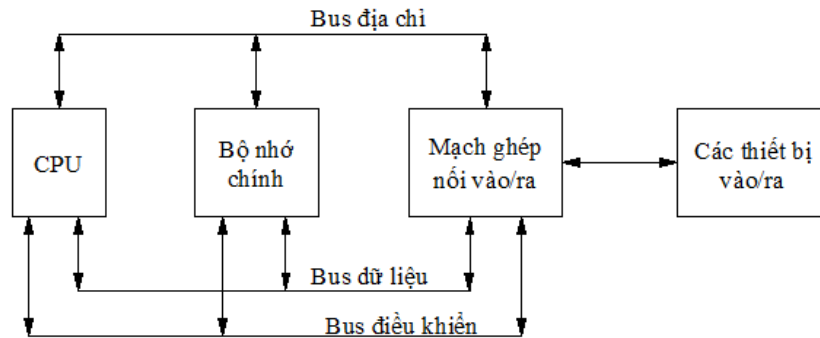
Ở trạng thái bình thường CPU có toàn quyền điều khiển hệ thống, nó sử dụng các Bus địa chỉ, Bus dữ liệu, Bus điều khiển. Khi có một tín hiệu thực hiện xin quyền điều khiển của hệ thống thì CPU có thể chuyển nhượng quyền sử dụng Bus. Có một số dạng sau:

+ *Bus Reques (BRQ)*:

HOLD là tín hiệu điều khiển từ mạch bên ngoài gửi đến chân HOLD của CPU, yêu cầu CPU chuyển nhượng các quyền sử dụng Bus.

+ *Bus Grant (BGT)*:

HLDA là tín hiệu phát ra từ CPU trả lời có chấp nhận việc chuyển nhượng Bus hay không.



Hình 1-3. Sơ đồ phối ghép Bus

1.4. Các kiến trúc của máy tính

1.4.1. Kiến trúc tuần tự

1.4.1.1. Kiến trúc Von-Neumann

Như đã đề cập ở trên, việc lập trình trên máy ENIAC là một công việc tẻ nhạt và mất rất nhiều thời gian. Công việc này sẽ đơn giản hơn nếu chương trình có thể được biểu diễn dưới dạng thích hợp cho việc lưu trữ trong bộ nhớ cùng với dữ liệu cần xử lý. Khi đó, máy tính chỉ cần lấy chỉ thị bằng cách đọc từ bộ nhớ, ngoài ra chương trình còn có thể được thiết lập hay thay đổi thông qua sự chỉnh sửa các giá trị lưu trong một phần nào đó của bộ nhớ.

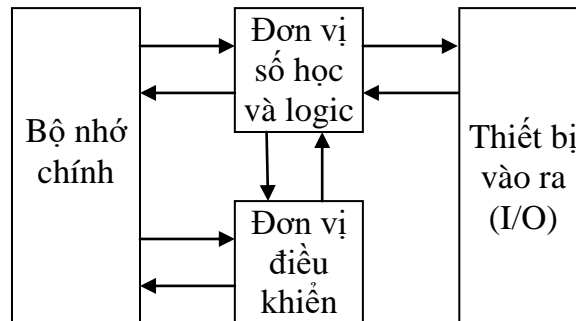
Ý tưởng này được biết đến với tên gọi là “*Khái niệm chương trình được lưu trữ*” do nhà toán học Jonh Von-Neumann, một cố vấn của dự án ENIAC, đưa ra ngày 08/11/1945, trong một bản đề xuất về một loại máy tính mới có tên gọi là EDVAC (Electronic Discrete Variable Computer). Máy tính này cho phép nhiều thuật toán khác nhau có thể được tiến hành trong máy tính mà không cần phải nối dây lại như máy ENIAC. Tiếp tục với ý tưởng của mình, vào năm 1946, Von-Neumann cùng với các đồng nghiệp đã bắt tay vào thiết kế một máy tính mới có chương trình được lưu trữ với tên gọi là ISA (Institute for Advanced Studies) tại Học viện nghiên cứu cao cấp Princeton của Mỹ.

Đây là kiến trúc Von-Neuman cổ điển. Kiến trúc này có một bộ xử lý thực hiện một lệnh ở một thời điểm, cho nên nó có kiểu là SISD (Single Instruction Stream-Single Data Stream – Đơn dòng lệnh, đơn dòng dữ liệu) có nghĩa là thực hiện một lệnh ở một thời điểm, nhận và cất dữ liệu tại một thời điểm.

Mặc dù mãi đến năm 1952 máy IAS mới được hoàn tất, nó vẫn là mô hình máy tính cho tất cả các máy tính đa năng sau này. Cấu trúc tổng quát của máy IAS bao gồm:

- + Đơn vị điều khiển, đơn vị số học và logic (ALU), bộ nhớ chính và các thiết bị vào – ra (I/O).
- + Bộ nhớ chính chứa chương trình và dữ liệu.

- + Bộ nhớ chính được đánh địa chỉ theo từng ngăn nhớ, không phụ thuộc vào nội dung của nó.
- + ALU thực hiện các phép toán với số nhị phân.
- + Đơn vị điều khiển nhận lệnh từ bộ nhớ, giải mã và thực hiện lệnh một cách tuần tự.
- + Đơn vị điều khiển điều khiển hoạt động của các thiết bị vào – ra (I/O).
- + Tốc độ chậm.



Hình 1-4. Cấu trúc của máy IAS

1.4.1.2. Các đặc điểm của kiến trúc von-Neumann

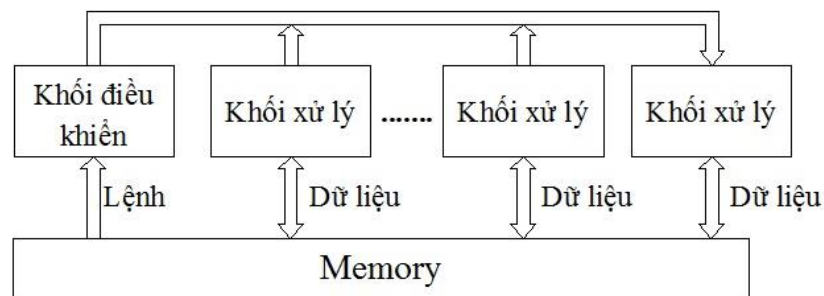
Kiến trúc von-Neumann dựa trên 3 khái niệm cơ sở:

- + Lệnh và dữ liệu được lưu trữ trong bộ nhớ đọc ghi chia sẻ - một bộ nhớ duy nhất được sử dụng để lưu trữ cả lệnh và dữ liệu
 - + Bộ nhớ được đánh địa chỉ theo vùng, không phụ thuộc vào nội dung nó lưu trữ.
 - + Các lệnh của một chương trình được thực hiện tuần tự.
- Quá trình thực hiện lệnh được chia thành 3 giai đoạn (Stages) chính:
- CPU đọc (Fetch) lệnh từ bộ nhớ.
 - CPU giải mã và thực hiện lệnh, nếu lệnh yêu cầu dữ liệu, CPU đọc dữ liệu từ bộ nhớ.
 - CPU ghi kết quả thực hiện lệnh vào bộ nhớ (nếu có).

1.4.2. Kiến trúc song song

1.4.2.1. Kiến trúc SIMD (Single Instruction stream Multiple Data stream)

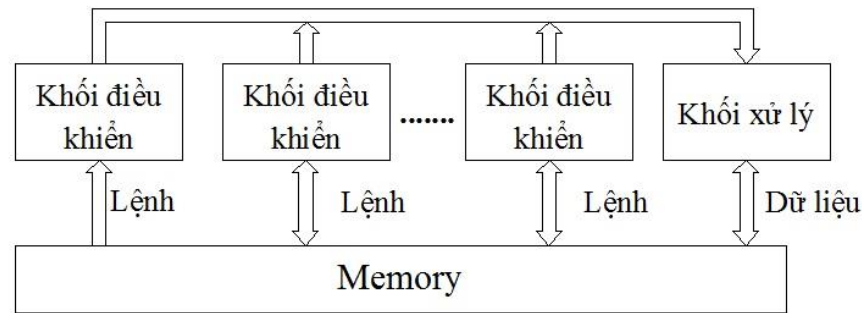
Máy tính SIMD có một đơn vị điều khiển hoạt động như máy tính kiểu Von Neumann, nghĩa là có thực hiện đơn dòng lệnh, nhưng nó có nhiều đơn vị xử lý thực hiện cùng một thao tác trên các mục dữ liệu khác.



Hình 1-5. Kiến trúc SIMD

1.4.2.2. Kiến trúc MISD (Multiple Instruction stream Single Data stream)

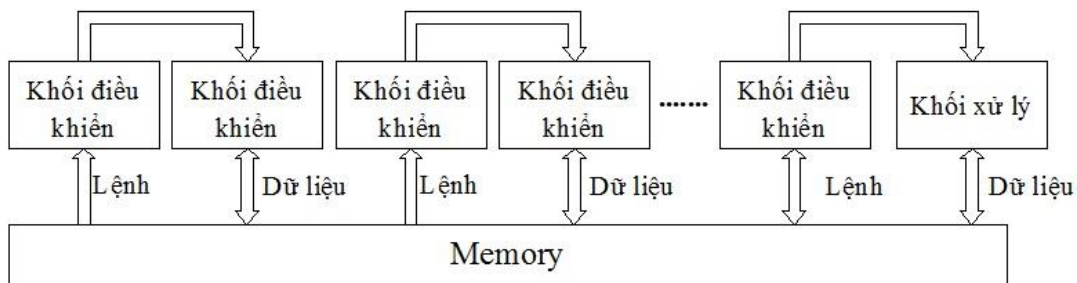
Về nguyên tắc, Trong một thời điểm có một vài lệnh cùng thao tác trên cùng một dữ liệu. Trên thực tế thì không có máy tính nào thuộc loại này vì có rất ít bài toán yêu cầu như vậy.



Hình 1-6. Kiến trúc MISD

1.4.2.3. Kiến trúc MIMD (Multiple Instruction stream Multiple Data stream)

Các máy tính loại này có thể có nhiều bộ xử lý độc lập, mọi bộ xử lý có thể thực hiện một chương trình khác nhau và xử lý trên các dữ liệu khác nhau.



Hình 1-7. Kiến trúc MIMD

Kiểu phân loại này đơn giản, dễ hiểu, vẫn còn hiệu lực đến hôm nay, mặc dù có những máy tính dùng kiến trúc hỗn hợp.

- Các máy tính SISD tương ứng với các máy một bộ xử lý mà chúng ta đã nghiên cứu.
- Các máy MISD kiểu máy tính này không sản xuất thương mại.
- Các máy SIMD có một số lớn các bộ xử lý giống nhau, cùng thực hiện một lệnh giống nhau để xử lý nhiều dòng dữ liệu khác nhau. Mỗi bộ xử lý có bộ nhớ dữ liệu riêng, nhưng chỉ có một bộ nhớ lệnh và một bộ xử lý điều khiển, bộ này đọc và thi hành các lệnh.

Các máy MIMD có kiến trúc song song, những năm gần đây, các máy MIMD nổi lên và được xem như một kiến trúc đương nhiên phải chọn cho các máy nhiều bộ xử lý dùng trong các ứng dụng thông thường, một tập hợp các bộ xử lý thực hiện một chuỗi các lệnh khác nhau trên các tập hợp dữ liệu khác nhau.

Các máy MIMD hiện tại có thể được xếp vào ba loại hệ thống là: *SMP* (Symmetric Multiprocessors), *Cluster* và *NUMA* (Nonuniform Memory Access)

a. Hệ thống SMP (Symmetric Multiprocessors)

Bao gồm nhiều bộ xử lý giống nhau được lắp đặt bên trong một máy tính, các bộ xử lý này kết nối với nhau bởi một hệ thống bus bên trong hay một vài sự sắp xếp chuyển mạch thích hợp. Vấn đề lớn nhất trong hệ thống SMP là sự kết hợp các hệ thống cache riêng lẻ. Vì mỗi bộ xử lý trong SMP có một cache riêng của nó, do đó, một khối dữ liệu trong bộ nhớ có thể tồn tại trong một hay nhiều cache khác nhau. Nếu một khối dữ liệu trong một cache của một bộ xử lý nào đó bị thay đổi sẽ

dẫn đến dữ liệu trong cache của các bộ xử lý còn lại và trong bộ nhớ trong không đồng nhất. Các giao thức cache kết hợp được thiết kế để giải quyết vấn đề này.

b. Hệ thống Cluster

Các máy tính độc lập được kết nối với nhau thông qua một hệ thống kết nối tốc độ cao (mạng tốc độ cao Fast Ethernet hay Gigabit) và hoạt động như một máy tính thống nhất. Mỗi máy trong hệ thống được xem như là một phần của cluster, được gọi là một nút (node). Hệ thống cluster có các ưu điểm:

- Tốc độ cao: Có thể tạo ra một hệ thống cluster có khả năng xử lý mạnh hơn bất cứ một máy tính đơn lẻ nào. Mỗi cluster có thể bao gồm hàng tá máy tính, mỗi máy có nhiều bộ xử lý.

- Khả năng mở rộng cao: có thể nâng cấp, mở rộng một cluster đã được cấu hình và hoạt động ổn định.

- Độ tin cậy cao: Hệ thống vẫn hoạt động ổn định khi có một nút trong hệ thống bị hư hỏng. Trong nhiều hệ thống, khả năng chịu lỗi (Fault tolerance) được xử lý tự động bằng phần mềm.

- Chi phí đầu tư thấp: Hệ thống cluster có khả năng mạnh hơn một máy tính đơn lẻ mạnh nhất với chi phí thấp hơn.

c. Hệ thống NUMA (Nonuniform Memory Access)

Là hệ thống đa xử lý được giới thiệu trong thời gian gần đây, đây là hệ thống với bộ nhớ chia sẻ, thời gian truy cập các vùng nhớ dành riêng cho các bộ xử lý thì khác nhau. Điều này khác với kiểu quản lý bộ nhớ trong hệ thống SMP (bộ nhớ dùng chung, thời gian truy cập các vùng nhớ khác nhau trong hệ thống cho các bộ xử lý là như nhau). Hệ thống này có những ưu điểm và nhược điểm như sau:

Ưu điểm:

- + Thực hiện hiệu quả hơn so với hệ thống SMP trong các xử lý song song.
- + Không thay đổi phần mềm chính.
- + Bộ nhớ có khả năng bị nghẽn nếu có nhiều truy cập đồng thời, nhưng điều này có thể được khắc phục bằng cách:

- Cache L1 và L2 được thiết kế để giảm tối thiểu tất cả các thâm nhập bộ nhớ.

- Cần các phần mềm cục bộ được quản lý tốt để việc các ứng dụng hoạt động hiệu quả.

- Quản trị bộ nhớ ảo sẽ chuyển các trang tới các nút cần dùng.

Nhược điểm:

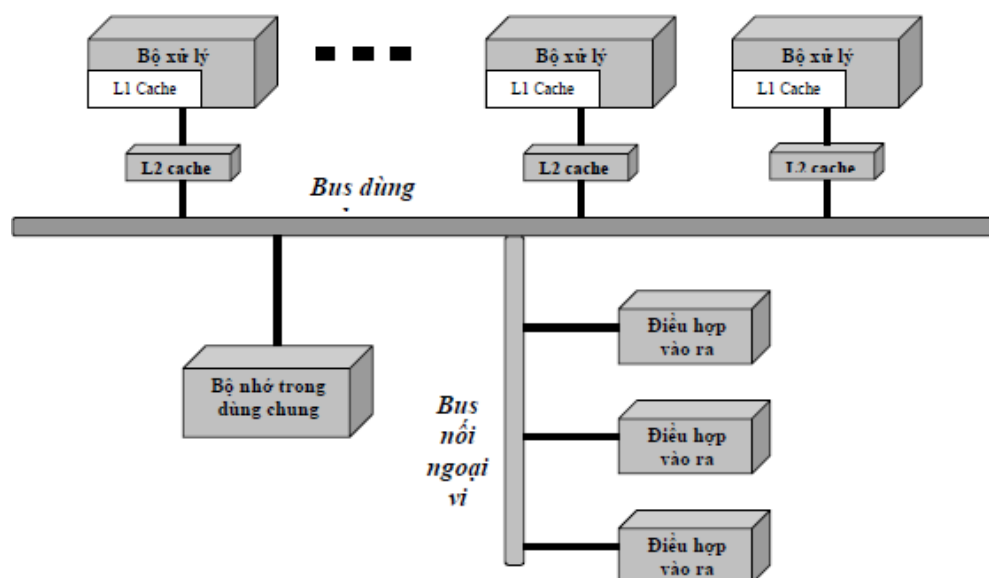
- + Hệ thống hoạt động không thông suốt như SMP, việc cấp phát các trang, các quá trình có thể được thay đổi bởi các phần mềm hệ thống nếu cần.

- + Hệ thống phức tạp.

Liên quan đến bộ nhớ trong các máy tính song song, chúng ta có thể chia thành hai nhóm máy:

- Nhóm máy thứ nhất

Ta gọi là máy có kiến trúc bộ nhớ chia sẻ, có một bộ nhớ trung tâm duy nhất được phân chia cho các bộ xử lý và một hệ thống bus chia sẻ để nối các bộ xử lý và bộ nhớ. Vì chỉ có một bộ nhớ trong nên hệ thống bộ nhớ không đủ khả năng đáp ứng nhu cầu thâm nhập bộ nhớ của một số lớn các bộ xử lý. Kiểu kiến trúc bộ nhớ chia sẻ được dùng trong hệ thống SMP.



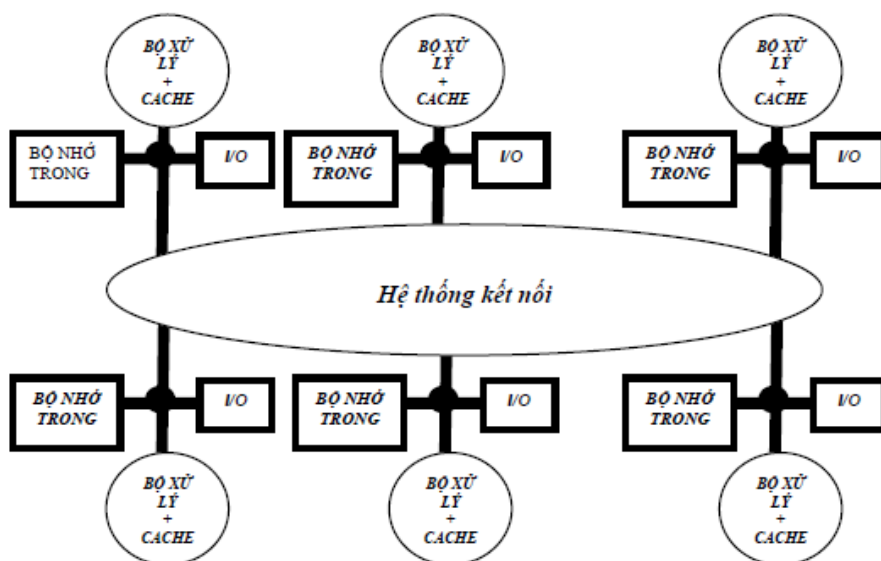
Hình 1-8. Máy tính song song với bộ nhớ dùng chung và hệ thống Bus dùng chung

- Nhóm máy thứ hai

Bao gồm các máy có bộ nhớ phân tán vật lý. Mỗi máy của nhóm này gồm có các nút, mỗi nút chứa một bộ xử lý, bộ nhớ, một vài cổng vào ra và một giao diện với hệ thống kết nối giữa các nút. Việc phân tán bộ nhớ cho các nút có hai thuận lợi:

- Thứ nhất, đây là một cách phân tán việc thâm nhập bộ nhớ.
- Thứ hai, cách này làm giảm thời gian chờ đợi lúc thâm nhập bộ nhớ cục bộ.

Các thuận lợi trên làm cho kiến trúc có bộ nhớ phân tán được dùng cho các máy đa xử lý có một số ít bộ xử lý. Điểm bất lợi chính của kiến trúc máy tính này là việc trao đổi dữ liệu giữa các bộ xử lý trở nên phức tạp hơn và mất nhiều thời gian hơn vì các bộ xử lý không cùng chia sẻ một bộ nhớ trong chung. Cách thực hiện việc trao đổi thông tin giữa bộ xử lý và bộ nhớ trong, và kiến trúc logic của bộ nhớ phân tán là một tính chất đặc thù của các máy tính với bộ nhớ phân tán.



Hình 1-9. Cấu trúc nền của một bộ nhớ phân tán

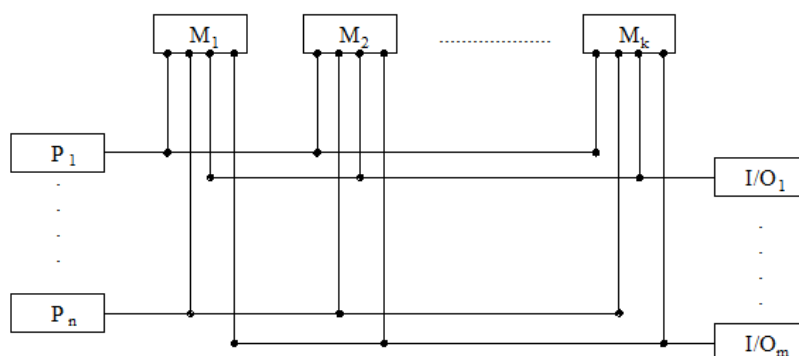
Có 2 phương pháp được dùng để truyền dữ liệu giữa các bộ xử lý.

+ Phương pháp thứ nhất

Là các bộ nhớ được phân chia một cách vật lý có thể được thâm nhập với một định vị chia sẻ một cách logic, nghĩa là nếu một bộ xử lý bất kỳ có quyền truy xuất, thì nó có thể truy xuất bất kỳ ô nhớ nào. Trong phương pháp này các máy được gọi có kiến trúc bộ nhớ chia sẻ phân tán (DSM: Distributed Sharing Memory). Từ bộ nhớ chia sẻ cho biết không gian định vị bị chia sẻ. Nghĩa là cùng một địa chỉ vật lý cho 2 bộ xử lý tương ứng với cùng một ô nhớ.

+ *Phương pháp thứ hai*

Không gian định vị bao gồm nhiều không gian định vị nhỏ không giao nhau và có thể được một bộ xử lý thâm nhập. Trong phương pháp này, một địa chỉ vật lý gắn với 2 máy khác nhau thì tương ứng với 2 ô nhớ khác nhau trong 2 bộ nhớ khác nhau. Mỗi môđun bộ xử lý-bộ nhớ thì cơ bản là một máy tính riêng biệt và các máy này được gọi là đa máy tính. Các máy này có thể gồm nhiều máy tính hoàn toàn riêng biệt và được nối vào nhau thành một mạng cục bộ.



Hình 1-10. Tổ chức kết nối của máy tính song song có bộ nhớ phân tán

Kiến trúc song song phát triển mạnh trong thời gian gần đây do các lý do:

- Việc dùng xử lý song song đặc biệt trong lãnh vực tính toán khoa học và công nghệ. Trong các lãnh vực này người ta luôn cần đến máy tính có tính năng cao hơn.
- Người ta đã chấp nhận rằng một trong những cách hiệu quả nhất để chế tạo máy tính có tính năng cao hơn các máy đơn xử lý là chế tạo các máy tính đa xử lý.
- Máy tính đa xử lý rất hiệu quả khi dùng cho đa chương trình. Đa chương trình được dùng chủ yếu cho các máy tính lớn và cho các máy phục vụ lớn.

Một số siêu máy tính dùng kỹ thuật xử lý song song:

- Máy điện toán Blue Gene/L của IBM đang được đặt tại Phòng thí nghiệm Lawrence Livermore, và đứng đầu trong số 500 siêu máy tính mạnh nhất thế giới. Siêu máy tính Blue Gene/L sẽ được sử dụng cho các công việc "phi truyền thống", chủ yếu là giả lập và mô phỏng các quá trình sinh học và nguyên tử. Máy điện toán Blue Gene/L đã đạt tốc độ hơn 70 teraflop (nghìn tỷ phép tính/giây). Kết quả này có thể sẽ đưa cỗ máy lên vị trí dẫn đầu trong danh sách các siêu máy tính nhanh nhất thế giới, được công bố ngày 8/11/2004. Theo đó, siêu máy tính do IBM lắp ráp đã đạt tốc độ 70,72 teraflop trong các cuộc thử nghiệm hồi tháng 10/2004. IBM nghiên cứu và phát triển Blue Gene với mục đích thử nghiệm nhằm tạo ra các hệ thống cực mạnh nhưng chiếm ít không gian và tiêu thụ ít năng lượng. IBM dự kiến, sẽ lắp đặt cho phòng thí nghiệm quốc gia Lawrence Livermore một siêu máy tính có tốc độ nhanh gấp 4 lần so với kỷ lục vừa đạt được. Khi đó, thiết bị sẽ được ứng dụng vào nhiều nghiên cứu khoa học. Hệ thống mới bao gồm 16,384 giao điểm điện toán kết nối 32.768 bộ xử lý.

- Siêu máy tính IBM Blue Gene/L vừa thiết lập kỷ lục mới đó là có khả năng xử lý 135,5 nghìn tỷ phép tính/giây (135,3 teraflop), vượt xa kỷ lục 70,72 teraflop do chính siêu máy tính này lập nên. Số bộ xử lý (BXL) của Blue Gene/L vừa được các

nhà khoa học tăng lên gấp đôi (64.000 BXL) nhằm tăng cường khả năng tính toán cho siêu máy tính này. Cũng cần phải nhắc lại rằng thiết kế hoàn thiện của siêu máy tính Blue Gene/L, dự kiến sẽ hoàn tất vào khoảng tháng 6 tới, sẽ bao gồm 130.000 BXL với tốc độ tính toán được kỳ vọng vào khoảng 360 teraflop.

- Hãng điện tử khổng lồ NEC phát hành một Supercomputer dạng vector, máy SX-8 mới ra đời có tốc độ xử lý cực đại lên tới 65 teraflop (65 nghìn tỷ phép tính dấu phẩy động/giây) và khả năng hoạt động ổn định ở mức xấp xỉ 90% của tốc độ 58,5% teraflop. Máy SX-8 có kiến trúc khác hẳn Blue Gene/L của IBM. Nó dùng kiến trúc vector nên đem đến độ ổn định khi hoạt động cao hơn nhiều so với dạng máy tính vô hướng (scalar) như của IBM.

- Một hệ thống tại trung tâm nghiên cứu của Cơ quan hàng không vũ trụ Mỹ (NASA) tại California cũng đạt được tốc độ 42,7 teraflop. Với tên gọi Columbia, siêu máy tính này sẽ được sử dụng để nghiên cứu khí tượng và thiết kế máy bay. Hệ thống trị giá 50 triệu USD (thời điểm tháng 10/2004) này sử dụng phần mềm Linux và đã được SGI ký hợp đồng bán cho Cơ quan hàng không vũ trụ Mỹ NASA. Nó có thể thực hiện 42,7 nghìn tỷ phép tính/giây (42,7 teraflop).

Tuy nhiên, tốc độ đó chưa phải là tất cả những gì nổi bật của siêu máy tính này: hệ thống mới chỉ khai thác có 4/5 công suất của *10.240 bộ xử lý Intel Itanium 2* trong toàn bộ cỗ máy đặt ở trung tâm nghiên cứu của NASA ở California (Mỹ). Siêu máy tính này không giống với hầu hết các siêu máy tính hiện nay thường được tạo nên theo kiểu cluster, với sự tham gia của nhiều cỗ máy giá rẻ. Columbia được thiết lập từ 20 máy tính mà mỗi chiếc có 512 bộ xử lý, kết nối bằng công nghệ mạng cao tốc và đều chạy một hệ điều hành độc lập. Cách xây dựng này rất hữu ích cho những công việc như giả lập các yếu tố khí động lực cho tàu không gian. Một ứng dụng khác của siêu máy tính Columbia là việc dự báo bão. Phần mềm cho tác vụ này đang được thiết kế và hứa hẹn khả năng dự báo chính xác đường đi của bão sớm 5 ngày. Toàn bộ máy Columbia chiếm dụng một diện tích bằng khoảng 3 sân bóng rổ.

Chương 2

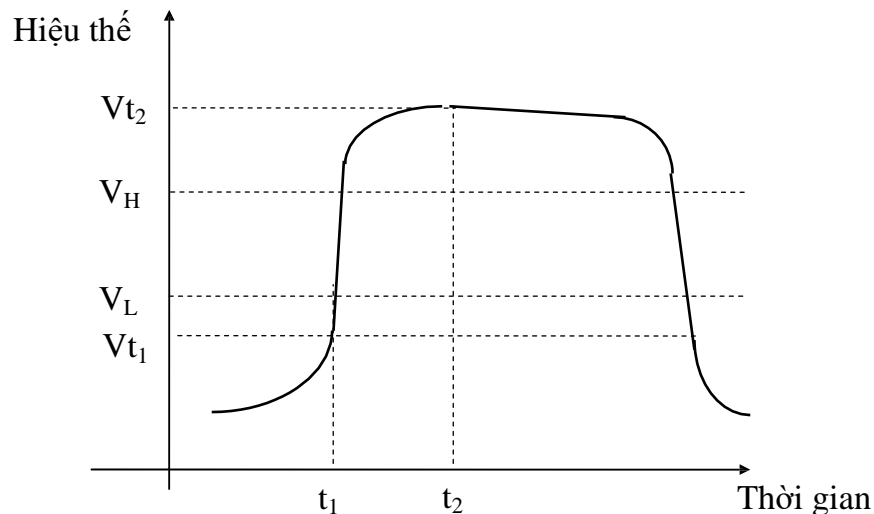
BIỂU DIỄN SỐ HỌC TRÊN MÁY TÍNH

2.1. Thông tin – biểu diễn và xử lý

2.1.1. Thông tin

Thông tin là sự hiểu biết, nhận thức của con người về thế giới khách quan. Nói cách khác, thông tin là một đại lượng phi vật chất mà con người chỉ có thể cảm nhận thông qua thế giới hiện thực khách quan. Muốn lưu trữ và truyền tải thông tin thì ta phải vật chất hóa thông tin thông qua một đại lượng vật lý nào đó.

Tuy là đại lượng phi vật chất, nhưng bản thân thông tin lại có giá trị nhất định nào đó. Việc lượng giá thông tin phụ thuộc vào xác suất xuất hiện của thông tin. Nếu xác suất xuất hiện càng cao thì thông tin có giá trị càng thấp, ngược lại xác suất xuất hiện càng thấp thì thông tin có giá trị càng cao.



Hình 2-1. Thông tin về hai trạng thái có ý nghĩa của hiệu điện thế

Trong hình 2-1, 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. Dữ liệu

Dữ liệu chính là thông tin đã được vật chất hóa thông qua một đại lượng vật lý nào đó, hay nói cách khác, dữ liệu là một đại lượng mang tin. Muốn có thông tin thì ta phải tập hợp và xử lý dữ liệu.

Trong thực tế, dữ liệu có thể là các loại tín hiệu vật lý (sóng điện từ, sóng âm thanh, tín hiệu điện, tín hiệu ánh sáng) hay có thể là số liệu, hình ảnh, chữ viết,...

Ví dụ

Bảng tính lương được lưu trữ trên máy tính, muốn lấy thông tin về những người có mức lương thấp hơn tối thiểu để có chính sách trợ cấp khó khăn thì phải qua một quá trình xử lý trên máy tính. Quá trình xử lý thông tin có thể là tìm kiếm lấy ra hay lọc theo một điều kiện nhất định nào đó hoặc có thể là giải mã dữ liệu đã được mã hóa

Cùng một thông tin, con người có thể biểu diễn thông tin này qua các đại lượng mang tin khác nhau, thể hiện qua các dạng dữ liệu khác nhau, ví dụ như văn bản, âm thanh, hình ảnh,... Nói cách khác, thông tin nằm trong dữ liệu. Xử lý thông tin bao gồm nhiều quá trình xử lý dữ liệu (truyền tin, lọc nhiễu, loại bỏ thông tin

du thừa, lưu trữ, tìm kiếm, lấy ra, sao chép, mã hóa, giải mã,...) để đưa ra thông tin hữu ích nhằm phục vụ nhu cầu của người sử dụng.

2.1.3. Biểu diễn thông tin và xử lý dữ liệu trong máy tính

Khi nói biểu diễn thông tin trong máy tính, chính là biểu diễn dữ liệu trong máy tính. Trong máy tính, thông tin được vật chất hóa thông qua tín hiệu điện, nói cách khác, dữ liệu trong máy tính được biểu diễn, lưu trữ và truyền tải thông qua tín hiệu điện.

Trong máy tính có rất nhiều bóng đèn, mỗi bóng đèn ở một trong hai trạng thái là sáng hay tắt.

- Trạng thái tắt (không có điện hay mức điện áp thấp) cho ta tín hiệu 0.

- Trạng thái sáng (có điện hay mức điện áp cao) cho ta tín hiệu 1.

Tập các tín hiệu 0/1 cho ta một số hệ cơ số 2 (hệ nhị phân) biểu diễn một giá trị dữ liệu nào đó.

Ví dụ

Có 8 bóng đèn được đánh số từ 0 đến 7, mỗi một bóng đèn lại ở một trạng thái tương ứng như: tắt, sáng, tắt, tắt, tắt, tắt, tắt, sáng, tức là ta có một tập các tín hiệu 0/1 tương ứng là: 0,1,0,0,0,0,0,1 và biểu diễn một số nhị phân là 01000001_B.

- Nếu dữ liệu lưu trữ là số thì tập tín hiệu này biểu diễn số 65.

- Nếu dữ liệu lưu trữ là ký tự thì tập tín hiệu này biểu diễn ký tự "A"

Như vậy, quá trình xử lý dữ liệu trong máy tính có thể được tóm tắt như sau:

- + Nhận dữ liệu đầu vào đã được số hóa.

- + Xử lý dữ liệu thông qua dãy các câu lệnh.

- + Đưa ra kết quả sau khi đã xử lý và lưu trữ dữ liệu đầu ra.

2.1.4. Đơn vị thông tin

Chúng ta đã biết, thông tin có thể tạo ra, phát sinh, truyền đi, lưu trữ và chọn lọc. Thông tin cũng có thể bị méo mó, sai lệch do nhiễu tác động hay do tác động của con người.

Người ta có thể định lượng tin tức bằng cách đo độ bất định của hành vi, trạng thái, Xác suất xuất hiện một tin càng thấp thì lượng tin càng cao vì độ bất ngờ của nó càng lớn.

Ví dụ

Nếu bản thân được thông báo cho biết điểm thi trong khi ta đã biết chính xác nó rồi, lúc đó ta nói lượng tin bằng 0 (tức là giá trị tin tức đem lại bằng không). Còn nếu tin tức đó được đưa đến trong tâm trạng ta đang chờ đợi, nhất là khi ta đang nghĩ mình có thể bị điểm kém, mà lại nhận được tin đúng đắn là được điểm cao thì thông tin này có lượng tin càng cao, càng có giá trị.

Để định lượng tin tức, năm 1948 Shannon đã đưa ra công thức tính lượng tin được gọi là Entropi:

$$H = -\sum_{i=1}^n P_i \log_a (P_i)$$

Trong đó:

P_i : là xác suất xuất hiện sự kiện i của hệ và hệ có a khả năng khác nhau.

Để chỉ ra được đơn vị đo, cần chỉ rõ ra được cơ số hóa hàm logarit:

- Nếu $a = 2$ thì đơn vị đo được gọi là đơn vị nhị phân (Binary unit – Bit).

- Nếu $a = e$ thì đơn vị đo được gọi là đơn vị tự nhiên (Natural unit - Nat).
 - Nếu $a = 10$ thì đơn vị đo được gọi là đơn vị thập phân (Hartley) ít được dùng.
- Trong trường hợp hệ có hai khả năng ($a = 2$).

Ví dụ

Lượng tin của việc xuất hiện mặt sấp hoặc ngửa khi chúng ta tung một đồng xu thì lượng tin của việc xuất hiện mặt sấp hoặc ngửa là:

$$H = -\sum_{i=1}^2 \frac{1}{2} \log_2 \left(\frac{1}{2} \right) = -\log_2 \left(\frac{1}{2} \right) = \log_2 2 = 1 \text{ (bit)}$$

Như vậy, trong trường hợp hệ có hai trạng thái ($a = 2$), hệ cơ số 2 sẽ cho ta đơn vị đo thông tin 1 bit.

Trong máy tính, để biểu diễn một giá trị số, chúng ta dùng hệ cơ số 2 hoặc nói ngắn gọn hơn là hệ 2 (Binary number system, viết tắt là B). Trong đó chỉ tồn tại hai chữ số 0 và 1 để biểu diễn các giá trị số (ứng với hai trạng thái là không có điện và có điện), 0 và 1 cũng là các giá trị có thể có của một chữ số hệ 2.

Mỗi một bit sẽ cho ta biết được trạng thái của một tín hiệu điện trên một đường dây tại một thời điểm: như điện áp ở mức cao (có điện) là 1, điện áp ở mức thấp (không có điện) là 0.

Bit là đơn vị cơ sở để xác định dung lượng của bộ nhớ, bộ nhớ được tổ chức theo byte.

Như vậy:

1 Byte = 8 bit.

1024 Byte = 1 KB.

1024 KB = 1 MB.

1024 MB = 1 GB.

1024 GB = 1 TB.

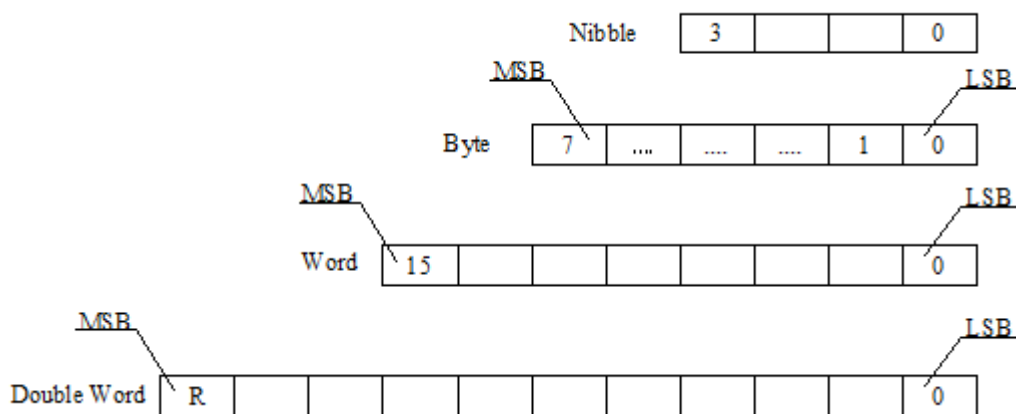
2.2. Các hệ đếm cơ bản

2.2.1. Hệ đếm nhị phân (Binary System)

Hệ đếm cơ số 2 hay hệ đếm nhị phân (Binary system), được viết tắt là Bin trên các máy tính khoa học và máy tính *Calculator* được cài đặt trên *Window*). Khi máy tính điện tử xuất hiện, người ta sử dụng hệ đếm nhị phân. Đó là hệ đếm chỉ sử dụng hai ký tự 1 và 0. Mỗi ký tự đứng bên trái bằng hai lần ký tự đứng bên phải nó nếu các ký tự đó là như nhau. Việc sử dụng hệ đếm nhị phân với hai ký tự 0 và 1 rất gần với logic vì mệnh đề chỉ có thể nhận một trong hai giá trị đúng hoặc sai tương ứng với giá trị 1 hoặc 0. Nó cũng tương ứng với việc một mạch điện chỉ có thể ở một trong hai trạng thái đóng hoặc mở. Phép đếm nhị phân cùng với phép toán logic là cơ sở hoạt động của máy tính.

Một số ở hệ 2 gồm các bit được đánh dấu bằng ký tự B hoặc b đi kèm ở cuối để phân biệt với các hệ khác khi làm việc cùng một lúc với nhiều hệ đếm khác nhau. Một cụm 4 bit sẽ tạo thành 1 nibble, một cụm 8 bit tạo thành 1 byte, một cụm 16 bit tạo thành 1 word (một từ), một cụm 32 bit tạo thành 1 double word (từ kép). Chữ số đầu tiên bên trái trong dãy các số hệ 2 loại 8, 16, 32 bit được gọi là bit có ý nghĩa lớn nhất (Most Significant Bit – MSB), còn bit cuối cùng bên phải (bit 0) trong dãy được gọi là bit có ý nghĩa bé nhất (Least Significant Bit – LSB)

Do chỉ có hai ký tự nên việc biểu diễn của một số trong hệ đếm cơ số 2 rất dài, vì vậy trong máy tính còn sử dụng hệ đếm cơ số 8 và hệ đếm cơ số 16, rất thuận tiện trong biểu diễn các số vì 2 là ước của 8 và 16.



Hình 2-2. Các đơn vị đo độ dài của số hệ 2 dẫn xuất từ bit

2.2.2. Hệ đếm thập phân (Decimal System)

Trong cuộc sống hàng ngày chúng ta thường sử dụng các số trong hệ đếm thập phân. Tất cả các số của hệ thập phân được tạo nên từ các chữ số từ 0 đến 9. Hệ đếm thập phân, hay còn gọi là hệ đếm cơ số 10 (Decimal system, được viết tắt là Dec trên các máy tính điện tử khoa học – *Scientific Calculator*, thường được dịch là máy tính cầm tay hoặc máy tính bỏ túi và máy tính *Calculator* được cài đặt trên *Window*).

Hệ đếm thập phân xuất hiện đầu tiên ở Ấn độ vào thế kỷ 5 sau công nguyên. Đến năm 1202 nhờ tác phẩm *Liber Abaci* của L. Fibonacci, một nhà toán học và thương gia người Ý, thì khoa học Ả rập và hệ đếm cơ số 10 mới được truyền bá vào châu Âu. Với sự phát minh ra nghề in vào thế kỷ 15 thì 10 chữ số mới có hình dạng cố định như hiện nay.

Các số viết trong hệ thập phân gồm 2 phần: Phần nguyên và phần thập phân được ngăn cách bởi dấu phẩy hoặc dấu chấm. Máy tính điện tử và các nước trên thế giới sử dụng dấu chấm, nhưng ở Việt nam thì sử dụng dấu phẩy (,).

Hệ đếm thập phân chỉ sử dụng 10 ký tự lần lượt là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Hệ đếm thập phân là hệ đếm theo quy tắc *vị trí*. Giá trị các ký tự giống nhau hoàn toàn khác nhau nếu nó đứng ở những vị trí khác nhau: gấp 10 thì thêm một nấc (đủ 10 thì thêm 1 đơn vị vào hàng bên trái nó), hay còn gọi là *hệ thập tiến*.

Do tính thập tiến người ta biết rằng mỗi chữ số đứng bên trái bằng 10 lần chữ số đứng bên phải nó nếu hai chữ số đó là như nhau. Điều này khác với hệ La Mã. Người ta cũng cố lý giải tại sao hệ đếm thập phân lại được đa số các nước trên thế giới sử dụng đến như vậy.

Có nhiều cách lý giải đưa ra như do hai bàn tay có 10 ngón, do đó ta dễ dàng đếm trên 10 ngón tay. Và khi đưa trẻ đầu tiên tập đếm thì chúng thường đếm trên đầu các ngón tay.

2.2.3. Hệ đếm thập lục phân (Hexadecimal System)

Hệ đếm cơ số 16 (Hexadecimal system, được viết tắt là Hex trên các máy tính khoa học và *Caculator*). Nếu chỉ sử dụng 10 ký tự từ 0 đến 9 như ở hệ đếm thập phân thì chưa đủ để biểu diễn các số trong hệ đếm cơ số 16. Vì vậy người ta đưa thêm vào các ký tự: A, B, C, D, E, F tương ứng với 10, 11, 12, 13, 14, 15.

Như vậy ở hệ đếm này ta sử dụng 16 ký tự: 0, 1, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Mỗi ký tự đứng bên trái bằng 16 lần ký tự đứng bên phải nó nếu hai ký tự đó giống nhau. Thực ra thì hệ đếm cơ số 16 cũng đã có ở Trung Quốc từ xưa, vì thời trước 1 cân của Trung Quốc có tới 16 lạng (do đó dân gian vẫn có câu “bên tám lạng bên nửa cân, bằng nhau”).

2.3. Mã hóa và lưu trữ dữ liệu trong máy tính

2.3.1. Nguyên tắc chung về mã hóa dữ liệu

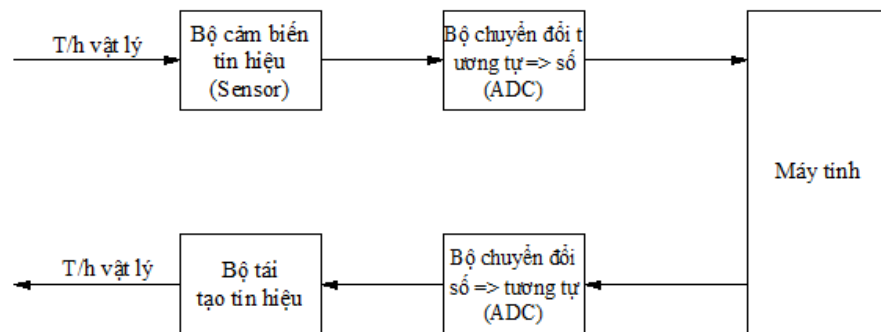
Nguyên tắc chung về mã hoá dữ liệu là mọi dữ liệu được đưa vào máy tính được mã hoá thành số nhị phân.

+ Các loại dữ liệu:

- Dữ liệu nhân tạo: do con người quy ước
- Dữ liệu tự nhiên: tồn tại khách quan với con người

+ Mã hoá dữ liệu nhân tạo

- Dữ liệu số nguyên: mã hoá theo một số chuẩn đã quy ước
- Dữ liệu số thực: mã hoá bằng số dấu chấm động
- Dữ liệu phi số (ký tự): mã hoá theo các bộ mã ký tự hiện hành như: ASCII, Unicode



Hình 2-3. Cách thức một tín hiệu vật lý qua máy tính

2.3.2. Lưu trữ dữ liệu trong máy tính

Bộ nhớ chính tổ chức lưu trữ dữ liệu theo đơn vị byte. Độ dài từ dữ liệu có thể chiếm từ 1 đến 4 byte. Vì vậy cần phải biết thứ tự chúng lưu trữ trong bộ nhớ chính đối với các dữ liệu nhiều byte.

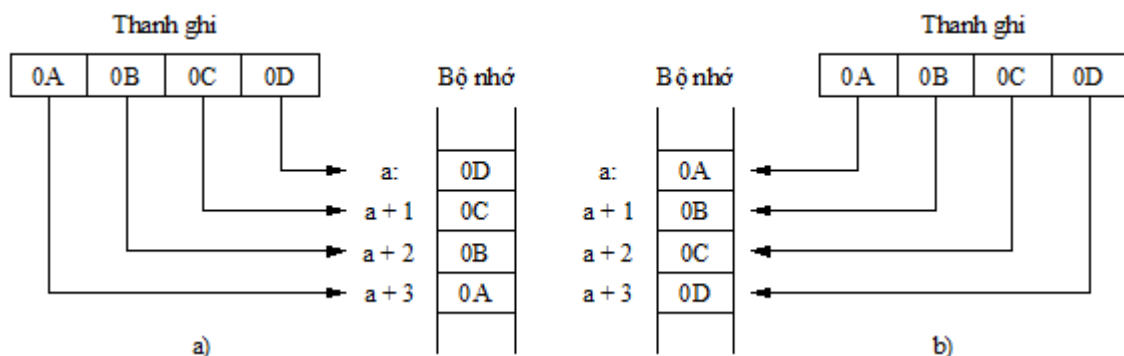
Có hai cách lưu trữ được đưa ra:

2.3.2.1. Lưu trữ đầu nhỏ (Little Endian)

Byte có ý nghĩa thấp hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ nhỏ hơn.

2.3.2.2. Lưu trữ đầu to (Big Endian)

Byte có ý nghĩa thấp hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ lớn hơn.



Hình 2-4a. Lưu trữ đầu nhỏ và hình 2-4b. Lưu trữ đầu to

Ví dụ

Mô tả vị trí các địa chỉ được lưu trữ theo hai trường hợp đầu to và đầu nhỏ

0001 1010	0010 1011	0011 1100	0100 1101
1A	2B	3C	4D

4D	300	1A	410
3C	301	2B	411
2B	302	3C	412
1A	303	4D	413

Little-endian Big-endian

Hình 2-5. Minh họa việc lưu trữ đầu to và lưu trữ đầu nhỏ

2.4. Biểu diễn các số

Khái niệm hệ thống số: Cơ sở của một hệ thống số định nghĩa phạm vi các giá trị có thể có của một chữ số. Ví dụ, trong hệ thập phân, một chữ số có giá trị từ 0 - 9, trong hệ nhị phân, một chữ số (một bit) chỉ có hai giá trị là 0 hoặc 1.

2.4.1. Chuyển đổi giữa các hệ số

2.4.1.1. Đổi một số hệ 2 sang hệ 10

Dạng tổng quát để biểu diễn giá trị của một số:

$$V_k = \sum_{i=-m}^{n-1} b_i \times k^i$$

Trong đó:

V_k : Số cần biểu diễn giá trị

m : số thứ tự của chữ số phần lẻ (phần lẻ của số có m chữ số được đánh số thứ tự từ -1 đến - m).

$n-1$: số thứ tự của chữ số phần nguyên (phần nguyên của số có n chữ số được đánh số thứ tự từ 0 đến $n-1$).

b_i : giá trị của chữ số thứ i .

k : hệ số ($k = 10$: hệ thập phân; $k = 2$: hệ nhị phân;...).

Hoặc

$$N_{(b)} = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots + a_{-m} b^{-m}$$

Như vậy muốn đổi một số từ hệ 2 sang hệ 10 tương ứng, ta áp dụng các công thức trên với cơ số là 2, có nghĩa là ta chỉ cần tính các giá trị 2^i tương ứng với các chữ số khác 0 thứ i của hệ 2 rồi sau đó cộng chúng lại với nhau.

Ví dụ

Số 11101.11₍₂₎ sẽ là:

Số nhị phân	: 1	1	1	0	1	.	1	1
Số vị trí	: 4	3	2	1	0		-1	-2
Trị vị trí	: 2 ⁴	2 ³	2 ²	2 ¹	2 ⁰		2 ⁻¹	2 ⁻²
Hệ 10 là	: 16	8	4	2	1		0.5	0.25

Như vậy

$$11101.11_{(2)} = 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25 = 29.75_{(10)}$$

2.4.1.2. Đổi một số hệ 10 sang hệ 2

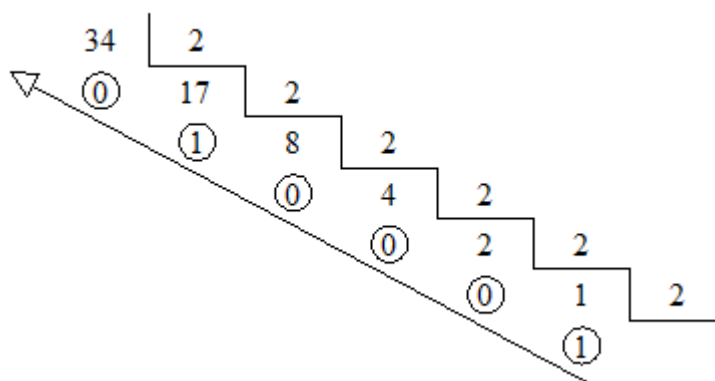
+ Trường hợp b là một số nguyên (quy tắc chia liên tiếp)

- Muốn đổi một số nguyên thập phân S thành số nguyên nhị phân x . Ta chia S cho 2 rồi chia thương nhận được cho 2, sau đó lại chia thương mới nhận được cho 2,...

Cứ chia mãi như vậy cho tới khi nào thương nhận được bé hơn 2 thì dừng. Thương nhận được cuối cùng sẽ là chữ số đầu tiên của x. Phần dư nhận được cuối cùng sẽ là chữ số thứ hai của x. Phần dư nhận được trước đó sẽ là chữ số thứ ba của x...

Ví dụ

Đổi số 34 từ hệ 10 sang hệ 2. Ta thực hiện như sau:



Hình 2-6. Trường hợp b là một số nguyên

Các số dư trong khung hình tròn sẽ được sắp xếp theo chiều mũi tên (lấy ngược lại với thứ tự thực hiện phép chia), ta nhận được kết quả là 100010_B

+ Trường hợp b là một số thập phân (nhân liên tiếp)

- Muốn đổi một số nguyên thập phân S thành số nguyên nhị phân x. Ta nhân S với 2 rồi nhân phần phân của tích nhận được với 2, sau đó lại nhân phần phân của tích mới nhận được với 2. Cứ nhân mãi như vậy cho tới khi nào nhận được tích với phần phân bằng 0 hoặc (đạt được độ chính xác cần thiết) thì dừng. Phần nguyên của tích nhận được đầu tiên sẽ là chữ số đầu tiên sau dấu phẩy của x. Phần nguyên của tích thứ hai sẽ là chữ số thứ hai của x,...

Ví dụ

Đổi số 0.125 từ hệ 10 sang hệ 2. Ta thực hiện như sau:

$$\begin{array}{r}
 0.125 \\
 \times \quad 2 \\
 \hline
 0.250 \\
 \times \quad 2 \\
 \hline
 0.500 \\
 \times \quad 2 \\
 \hline
 1.000
 \end{array}$$

Như vậy, số 0.125 ở hệ 10 sau khi đổi sang hệ 2 là $0,001_B$

Ví dụ

Đổi số 34,125 từ hệ 10 sang hệ 2. Sau khi thực hiện, ta thu được kết quả như sau:

$$34.125_D = 100010.001_B$$

2.4.1.3. Đổi một số hệ 10 sang hệ 16

Trước hết ta tách phần nguyên và phần lẻ rồi tiến hành biến đổi chúng riêng biệt. Sau đó chúng ta sẽ ghép lại để có kết quả cần tìm. Biến đổi số nguyên, ta làm như sau:

Cho N là số tự nhiên. Ta viết N dưới dạng đa thức:

$$N = d_n b_n + d_{n-1} b^{n-1} + \dots + d_0$$

Nhận xét rằng, $0 \leq d_0 < b$. Do vậy, khi chia N cho b thì phần dư của phép chia đó là d_0 còn thương số N1 sẽ là:

$$N1 = d_n b^{n-1} + d_{n-1} b^{n-2} + \dots + d_1$$

Tương tự, d_1 chính là phần dư của phép chia N_1 cho b . Gọi N_2 là thương của phép chia này. Quá trình chia như vậy được thực hiện liên tiếp và ta sẽ lần lượt nhận được giá trị các d_i . Quá trình sẽ dừng lại khi nhận được thương số bằng 0. Để có biểu diễn cần tìm, các phần dư thu được cần sắp xếp theo thứ tự ngược lại.

Ví dụ

Đổi số 125_D sang hệ 16

- Ta lấy 125 chia cho 16 được 7 dư 13 = C, nhớ C.
- Lấy 7 chia cho 16 được 0 dư 7
- Kết quả ta được $7C_H$

2.4.1.4. Đổi một số từ hệ 16 sang hệ 10

Để biến đổi một số thập lục phân sang số thập phân tương đương, chúng ta nhân mỗi giá trị thập phân của từng con số trong số thập lục phân với số mũ của 16, rồi tìm tổng của các giá trị.

Ví dụ

Đổi số $9E4B$ sang hệ thập phân.

Đánh số thứ tự các kí tự từ phải sang trái, bắt đầu từ 0.

Số hex cần chuyển	9	E	4	B
Số thứ tự	3	2	1	0
Giá trị tương ứng hệ 10	9	14	4	11

$$9E4B_H = (9 \times 16^3) + (14 \times 16^2) + (4 \times 16^1) + (11 \times 16^0) \\ = (36864) + (3584) + (64) + (11) = 40523_D$$

2.4.2. Biểu diễn số nguyên

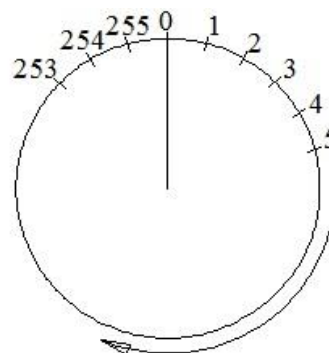
2.4.2.1. Số nguyên không dấu (Unsigned Integer)

- + Bước 1. Chuyển M ở hệ đếm bất kỳ sang M ở hệ nhị phân ($M \rightarrow M_B$).
- + Bước 2. Mỗi bit của M_B được đặt vào một cột nhị phân theo quy tắc: bit có trọng số thấp nhất để ở bên phải, cao nhất để ở bên trái.

- + Bước 3. Xác định khoảng giá trị các số không dấu biểu diễn trên n cột nhị phân (có nghĩa là ta đi tìm $M = M_{\min} \div M_{\max}$).

- M_{\min} : Khi các cột nhị phân nhận giá trị 0.

- M_{\max} : Khi các cột nhị phân nhận giá trị 1.



Hình 2-7. Biểu diễn số nguyên không dấu

Ví dụ

Biểu diễn số 7 trên 8 cột nhị phân

- + Bước 1. Chuyển số 7 ở hệ 10 sang hệ 2 ta được: $7_D = 0111_B$.
- + Bước 2. Đặt các bit 0111 vào 8 cột nhị phân.
- + Bước 3. Xác định khoảng các số không dấu.

Dạng tổng quát

$$M = 0 \div 2^n - 1 \Rightarrow M = \text{Min} \div \text{Max}$$

Ở đây, nếu $n = 8$ bit thì ta

có:

$$M = 0 \div 2^8 - 1 = 0 \div 255.$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	1	0	1

Chú ý:

- Với $n = 16$ bit ta có: $M = 0 \div 2^{16} - 1$.
- Với $n = 32$ bit ta có $M = 0 \div 2^{32} - 1$.

- Với $n = 64$ bit ta có $M = 0 \div 2^{64} - 1$.

Như vậy. Dải miền trị của số nguyên không dấu được biểu diễn bằng hình tròn có:

- Giá trị nhỏ nhất bằng 0.
- Giá trị lớn nhất bằng $2^n - 1$

2.4.2.2. Số nguyên có dấu (Signed Integer)

Có nhiều cách để biểu diễn số nguyên n bit có dấu, sau đây ta sẽ làm quen một vài cách biểu diễn số nguyên có dấu cơ bản sau:

a. Biểu diễn bằng dấu và trị tuyệt đối

Sử dụng bit cao nhất (s) biểu thị dấu của số, các bit còn lại biểu diễn giá trị tuyệt đối.

- Nếu là số âm (-) thì bit cao nhất (s) điền vào số 1.

- Nếu là số dương (+) thì bit cao nhất (s) điền vào số 0.

+ Một byte Với $n = 8$ cho

phép biểu diễn tất cả các số có dấu từ -127 (1000 0000) đến +127 (0111 1111).

+ Với $n = 16$ có thể biểu diễn các số từ -32767 đến +32767.

Có hai cách biểu diễn số 0 là: 0000 0000 (+0) và 1000 0000 (-0).

+ Dải giá trị biểu diễn: $-(2^{n-1} - 1)$ đến $+(2^{n-1} - 1)$.

Ví dụ

$$+25_{10} = 0001\ 1001_2$$

$$-25_{10} = 1001\ 1001_2$$

b. Biểu diễn bằng số bù 1

Dùng n bit biểu diễn số nguyên có dấu A có dạng:

$$a_{n-1}, a_{n-2}, \dots, a_2, a_1, a_0$$

- Với A là số dương, bit $a_{n-1} = 0$, các bit còn lại biểu diễn độ lớn như số không dấu

- Với A là số âm: được biểu diễn bằng số bù 1, thực hiện bằng cách ta lấy đảo các bit tương ứng (kể cả bit dấu), có nghĩa là nếu bit 0 \rightarrow 1 và bit 1 \rightarrow 0.

Ví dụ

$$\text{Cho } A = +25_{10} = 00011001_2$$

$$\text{Số bù 1 của } A = -25_{10} = 11100110_2$$

- Một Byte cho phép biểu diễn tất cả các số có dấu từ -127 (1000 0000₂) đến 127 (0111 1111₂)

- Có hai cách biểu diễn cho số 0 là 0000 0000 (+0) và 1111 1111 (-0).

Ví dụ

Biểu diễn các số nguyên có dấu sau đây theo mã bù một:

$$A = +58; B = -80$$

Ta có:

$$+A = +58 = 0011\ 1010$$

$$+B = -80$$

$$\text{Ta có: } +80 = 0101\ 0000$$

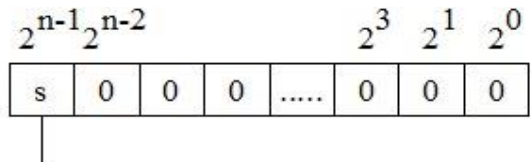
$$\text{Số bù một} = 1010\ 1111$$

$$\text{Vậy mã bù một của } -80 \text{ là: } 1010\ 1111$$

Ví dụ

Hãy xác định giá trị của các số nguyên có dấu được biểu diễn theo mã bù một dưới đây:

$$P = 0110\ 0010$$



Hình 2-8. Biểu diễn số nguyên có dấu

$$Q = 1101 \ 1011$$

Ta có

$$P = 0110 \ 0010 = 26 + 25 + 21 = 64 + 32 + 2 = +98$$

$$Q = 1101 \ 1011 = -(27 - 1) + 26 + 24 + 23 + 21 + 20 = -127 + 64 + 16 + 8 + 2 + 1 = -36$$

c. Biểu diễn bằng số bù 2

Để có số bù 2 của A, người ta lấy số bù 1 của A rồi cộng thêm 1.

$$\begin{array}{rcl} \text{Cho số A} & = & 0010 \ 0101 \\ \text{Số bù một} & = & 1101 \ 1010 \\ & + & \underline{ 1} \\ \text{Số bù hai} & = & 1101 \ 1011 \end{array}$$

+ Chỉ có một cách biểu diễn số 0 là: 0000 0000

+ Một từ n bit có thể biểu diễn các số có dấu từ $-(2^{n-1})$ đến $+(2^{n-1} - 1)$.

- Dùng 1 Byte (8 bit) để biểu diễn một số có dấu lớn nhất là +127 và số nhỏ nhất là -128.

+ Với n = 8 bit

Biểu diễn được các giá trị từ -128 đến +127

$$0000 \ 0000 = 0$$

$$0000 \ 0001 = +1$$

$$0000 \ 0010 = +2$$

$$0000 \ 0011 = +3$$

....

$$0111 \ 1111 = +127$$

$$1000 \ 0000 = -128$$

$$1000 \ 0001 = -127$$

....

$$1111 \ 1110 = -2$$

$$1111 \ 1111 = -1$$

Chú ý:

$$+127 + 1 = -128$$

$$-128 - 1 = +127 \rightarrow \text{Do tràn xảy ra}$$

+ Với n = 16 bit, biểu diễn được các giá trị từ -32768 đến +32767

$$0000 \ 0000 \ 0000 \ 0000 = 0$$

$$0000 \ 0000 \ 0000 \ 0001 = 1$$

.....

$$0111 \ 1111 \ 1111 \ 1111 = +32767$$

$$1000 \ 0000 \ 0000 \ 0000 = -32768$$

....

$$1111 \ 1111 \ 1111 \ 1111 = -1$$

+ Với n = 32 bit, biểu diễn được các giá trị từ -2^{31} đến $2^{31} - 1$

+ Với n = 64 bit, biểu diễn được các giá trị từ -2^{63} đến $2^{63} - 1$

Ví dụ

Biểu diễn các số nguyên có dấu sau đây theo mã bù hai: A = +58; B = -80

Ta có

$$A = +58 = 0011 \ 1010$$

$$B = -80$$

$$\text{Ta có: } +80 = 0101 \ 0000$$

$Số\ bù\ một = 1010\ 1111$
 $\quad\quad\quad + \quad\quad\quad 1$
 $Số\ bù\ hai = 1011\ 0000$
 Vậy mã bù hai của -80 là: 1011 0000

2.4.3. Biểu diễn số nguyên theo mã BCD (Binary Coded Decimal Code)

+ Sử dụng 4 bit để mã hóa 10 chữ số thập phân từ 0 đến 9.

0 -> 0000 5 -> 0101
 1 -> 0001 6 -> 0110
 2 -> 0010 7 -> 0111
 3 -> 0011 8 -> 1000
 4 -> 0100 9 -> 1001

+ Có 6 tổ hợp không dùng là:

1010, 1011, 1100, 1101, 1110, 1111

Ví dụ

+ Số 35 -> 0011 0101 BCD

+ Số 61 -> 0110 0001 BCD

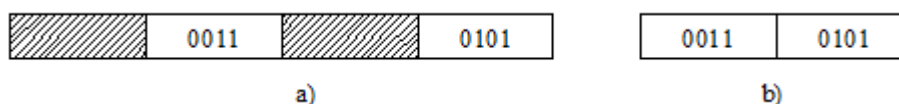
+ Số 1087 -> 0001 0000 1000 0111 BCD

2.4.3.1. Cách lưu trữ mã BCD

Có hai cách lưu trữ mã BCD là:

+ BCD dạng không gói (Unpacked BCD): Mỗi BCD 4 bit được lưu trữ trong 4 bit thấp của mỗi byte

+ BCD dạng gói (Packed BCD): Hai số BCD được lưu trữ trong 1 byte



Hình 2-9a. Lưu trữ dạng không gói và hình 2-9b. Lưu trữ dạng gói

2.4.3.2. Thực hiện các phép toán số học với hệ 2

a. Phép cộng

+ **Cộng số nguyên không dấu**

Khi ta cộng hai số nguyên không dấu n bit, kết quả nhận được là n-bit.

- Nếu không có nhớ ra khỏi bit cao nhất thì kết quả nhận được luôn luôn đúng $C_{out} = 0$.

- Nếu có nhớ ra khỏi bit cao nhất thì kết quả nhận được là sai. Hay nói chính xác hơn là phép cộng bị tràn $C_{out} = 1$.

- Tràn còn nhớ xảy ra (Carry Out) khi kết quả của phép toán $> n^n - 1$.

Quy tắc phép cộng số hệ 2 được mô tả trong bảng 2.1

Bảng 2-1. Mô tả qui tắc $y = a + b$

A	B	Tổng (Y)	Nhớ (Carry)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Ví dụ

Thực hiện phép cộng hai số nguyên không dấu sau:

(Ta đổi 71 ở hệ 10 sang hệ 2 (chia liên tiếp cho 2), và thêm 0 vào phía trước cho đủ $n = 8$ bit

71 = 01000111 Vì 71 là số dương nên ta thêm số 0 vào phía trước cho đủ 8 bit

+ 25 = 00011001 Vì 25 là số dương nên ta thêm số 0 vào phía trước cho đủ 8 bit
 = 96 = 01100000 = $1 \times 2^6 + 1 \times 2^5 = 64 + 32 = 96$ (Đúng)

251 = 11111011
 + 16 = 00010000 Vì 25 là số dương nên ta thêm số 0 vào phía trước cho đủ 8 bit
 = 267 1 = $1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 = 8 + 2 + 1 = 11$ (Sai)
 0001011 (Trường hợp này có tràn nhớ ra ngoài $C_{out} = 1$)

209 = 11010001
 + 73 = 01001001
 = 282 = 1 = $16 + 8 + 2 = 26$ (Sai)
 00011010

Nếu muốn có kết quả đúng, ta phải thực hiện cộng theo $n = 16$ bit

209 = 0000 0000 1101 0001
 73 = 0000 0000 0100 1001
 282 = 0000 0001 0001 1010 = $256 + 16 + 8 + 2 = 282$ (Đúng)

+ Cộng số nguyên có dấu

Khi cộng hai số nguyên có dấu n bit, ta không cần quan tâm đến bit nhớ ra ngoài (C_{out}), kết quả nhận được là n bit.

- Khi cộng 2 số khác dấu, kết quả luôn đúng.

- Khi cộng 2 số cùng dấu:

* Nếu dấu kết quả cùng dấu với các số hạng thì kết quả là đúng.

* Nếu kết quả có dấu ngược lại, khi đó có tràn xảy ra (Overflow) và kết quả sai.

- Tràn xảy ra khi tổng nằm ngoài dải biểu diễn của nó $[-(2^{n-1})$ đến $+(2^{n-1}-1)$]

Ví dụ

-71 = đổi ra nhị phân +71 = 01000111, bù 1 = 10111000, bù 2 = 10111001

+25 = 00011001

=> -71 + (+25) = -45 = 10111001 + 00011001 = 11010010

Đảo của 11010010 = 00101101, bù 2 = 00101110 = $32+8+4+2 = -46$ (đúng)

b. Phép trừ

Trong máy tính không có bộ trừ, chỉ có bộ cộng để thực hiện phép cộng. Do vậy, với phép toán $C = A - B$ thì máy tính sẽ thực hiện là $C = A + (-B)$. Có nghĩa là lấy bù 2 của B để được $-B$ rồi đem cộng với A .

Số bù 2

Trong máy tính tận dụng các bộ cộng đã có sẵn, với phép trừ được chuyển đổi thành phép cộng số bị trừ với số đảo dấu của số trừ, số đảo dấu của số trừ chính là một số âm. Khi thực hiện phép trừ trong máy tính tất yếu cũng sẽ có kết quả là một số âm. Vấn đề đặt ra là khi đổi số ở hệ 2 là phải biểu diễn số âm nhị phân như thế nào cho thích hợp để đáp ứng được việc sử dụng các bộ cộng trong máy tính. Việc sử dụng số bù 2 chính là cách biểu diễn số có dấu trong máy tính ngày nay.

Như vậy, số bù 2 của số B là số đảo dấu của của chính số B đó.

Qui tắc phép trừ số hệ 2 được mô tả trong bảng 2.2

Bảng 2-2. Mô tả qui tắc $y = a - b$

A	B	Tổng (Y)	Mượn
0	0	0	0
0	1	1	1

1	0	1	0
1	1	0	0

Để thuận tiện trong việc tính các phép trừ, ta có bảng biểu diễn các số theo số hệ 2, số hệ 2 có dấu và số bù 2.

Bảng 2-3. Biểu diễn các số theo hệ 2, số hệ 2 có dấu và số bù 2

Số hệ 2 (8 bit)	Số hệ 10 tương ứng (số không dấu)	Số hệ 10 tính theo số bù 2
0000 0000	0	0
0000 0001	1	+1
0000 0010	2	+2
0000 0011	3	+3
....
0111 1101	125	+125
0111 1110	126	+126
0111 1111	127	+127
1000 0000	128	-128
...
1111 1101	253	-3
1111 1110	254	-2
1111 1111	255	-1

c. Phép nhân

Phép nhân số hệ 2 thực hiện giống phép nhân số hệ 10. Quy tắc nhân số hệ 2 được mô tả trong bảng 2.4.

Bảng 2-4. Mô tả qui tắc nhân

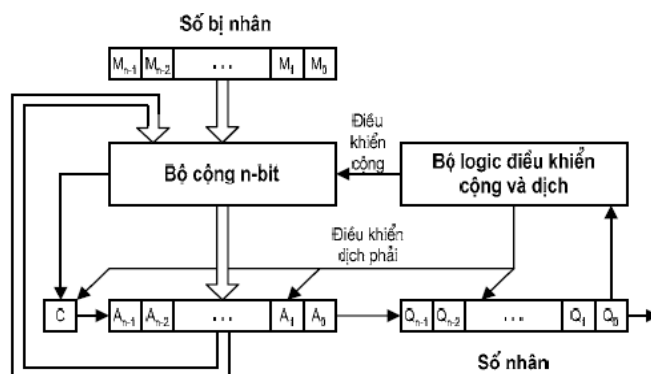
A	B	Tổng (Y)
0	0	0
0	1	0
1	0	0
1	1	1

Nhân số nguyên không dấu

Các tích riêng phần được xác định như sau:

- + Nếu bit của số nhân bằng 0 → tích riêng phần bằng 0.
- + Nếu bit của số nhân bằng 1 → tích riêng phần bằng số bị nhân.
- + Tích riêng phần tiếp theo được dịch trái một bit so với tích riêng phần trước đó.

- + Tích bằng tổng các tích riêng phần.
- + Nhân hai số nguyên n bit, tích có độ dài 2n bit (không bao giờ tràn).



Hình 2-10. Bộ nhân số nguyên

Độ dài cực đại trong ví dụ 2.31 là 8 bit. Nếu ta nhân các số 8 bit hoặc 16 bit thì độ dài cực đại của kết quả là 16 bit hoặc 32 bit. Mỗi lần nhân một bit khác 0 của số nhân với số bị nhân, ta thu được chính số bị nhân đã dịch trái nó một số lần (một số bit) tương ứng với vị trí các bit khác 0, tạo ra một thành phần của tổng tích lũy. Tổng của các thành phần như ví dụ 2.31 là kết quả của phép nhân.

Phân tích ví dụ 2.31 ta thấy, phép nhân có thể thực hiện theo thuật toán cộng và dịch như sau:

- + Thành phần đầu tiên của tổng tích lũy thu được là tích của LSB trong số nhân với số bị nhân. Nếu $LSB = 0$ thì thành phần này bằng 0; nếu $LSB = 1$ thì thành phần này bằng chính số bị nhân.

- + Mỗi thành phần thứ i tiếp theo của tổng tích lũy sẽ được tính bằng cách tương tự, nhưng phải dịch trái đi i bit (có thể bỏ qua các thành phần $= 0$).

- + Tổng của các thành phần là tích cần tìm.

Nhân số nguyên có dấu

Phương pháp 1:

- + Bước 1. Chuyển đổi số bị nhân và số nhân thành số dương tương ứng.

- + Bước 2. Nhân hai số dương bằng thuật giải nhân số nguyên không dấu, được tích của hai số dương.

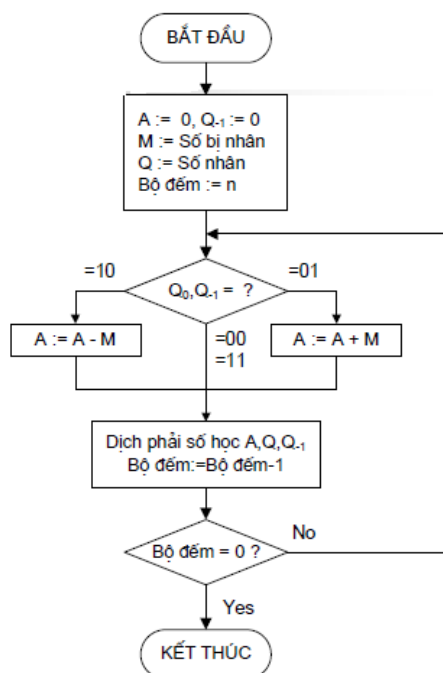
- + Bước 3. Hiệu chỉnh dấu của tích:

- Nếu hai thừa số ban đầu cùng dấu thì giữ nguyên kết quả ở bước 2.

- Nếu hai thừa số ban đầu là khác dấu thì đảo dấu kết quả của bước 2.

Phương pháp 2:

Sử dụng thuật toán nhân nhanh Booth



Hình 2-12. Lưu đồ giải thuật Booth

Trong máy tính, phép nhân được thực hiện bởi bộ cộng và bộ dịch trái theo như thuật toán cộng và dịch vừa trình bày.

d. Phép chia

Phép chia là phép tính ngược của phép nhân. Từ đó ta suy ra phép chia có thể thực hiện được bằng các phép trừ và phép dịch phải liên tiếp cho tới khi không thể trừ được nữa (do không còn gì để trừ hoặc số bị trừ nhỏ hơn số chia).

Ví dụ 2.34

1	0	0	1	1	1	1	1	1	1	1	0	0
1	1	0	0								1	1
1	1	1	1	1							1	0
1	1	0	0								1	1
				1	1	1	1					
				1	1	0	0					
								1	1			

Trong phép tính ở trên, ta liên tục cần phải dự đoán và kiểm tra để tìm ra kết quả đúng. Công việc này rất khó khăn với các mạch điện tử của khối tính toán so học (vốn là các phần tử để thực hiện phép cộng và dịch trong máy tính).

2.5. Biểu diễn số thực (dấu phẩy động)**2.5.1. Nguyên tắc chung**

Trong máy tính, số thực được biểu diễn và được viết dưới dạng dấu phẩy động (dạng số mũ). Còn trong thực tế, con người biểu diễn số thực ở hai dạng:

+ Dạng bình thường, chẳng hạn như: 3.14; 3.0; -24.123456; -0.001. Việc sử dụng dấu chấm thay vì dấu phẩy mục đích là để tách giữa phần nguyên và phần thực của số thực. Đây là cách viết của các nước Anh, Mỹ.

+ Dạng viết dưới dạng số mũ gọi là số thực dấu chấm động.

Để hiển thị ra màn hình một số thực dạng bình thường, hay dạng biết dấu chấm động ta phải thông qua giả lập bằng phần mềm.

2.5.2. Số chấm động

Một số thực R được biểu diễn dưới dạng số dấu chấm động chuẩn hóa có dạng như sau:

$$R = (-1)^S \times M \times R^E$$

Trong đó:

S: Đặc trưng cho dấu của số (nếu S = 0 thì R dương; ngược lại S = 1 thì R âm).

M: Phần định trị được chuẩn hóa.

R: Là cơ số cho biết số ở hệ cơ số nào.

E: Số mũ.

Ví dụ

- Số thực là 84537.8765275 được chuẩn hóa ở dạng dấu chấm động là:

$$84537.8765275 = (-1)^0 \times 8.45378765275 \times 10^4$$

- Số thực là 0.0045378765275 được chuẩn hóa ở dạng dấu chấm động là:

$$0.0045378765275 = (-1)^0 \times 4.5378765275 \times 10^{-3}$$

- Số thực là 1011.001_B được chuẩn hóa ở dạng dấu chấm động là:

$$1011.001_B = (-1)^0 \times 1.011001_B \times 2^3$$

- Số thực là -537.8765275 được chuẩn hóa ở dạng dấu chấm động là:

$$-537.8765275 = (-1)^1 \times 5.378765275 \times 10^2$$

- Số thực là -0.00045378765275 được chuẩn hóa ở dạng dấu chấm động là:

$$-0.00045378765275 = (-1)^1 \times 4.5378765275 \times 10^{-4}$$

- Số thực là 0.00011011011001_B được chuẩn hóa ở dạng dấu chấm động là:

$$0.00011011011001_B = (-1)^0 \times 1.1011011001_B \times 2^{-4}$$

Có hai chuẩn định dạng dấu chấm động quan trọng là: chuẩn MSBIN của Microsoft và chuẩn IEEE. Cả hai chuẩn này đều dùng hệ đếm nhị phân.

Hầu hết các máy tính ngày nay thường dùng chuẩn IEEE 754 (Institute of Electric & Electronic Engineers) để biểu diễn số thực. Đây là chuẩn được mọi hãng chấp nhận và được dùng trong bộ xử lý toán học của Intel. Bit dấu nằm tại vị trí cao nhất; kích thước phần mũ và khuôn dạng phần định trị thay đổi theo từng loại số thực.

Dạng tổng quát của dấu chấm động hệ 2 theo nguyên lý che số 1 như sau:

$$\mathbf{R = (-1)^S \times 1.M \times 2^{E - Bias}} \quad (2.1)$$

Số thực (dấu chấm phẩy động) được biểu diễn bằng hai phần chính và một bit dấu theo khuôn dạng sau:

Trong đó:

M: Là phần định trị (Mantissa) chỉ lưu phần lẻ nhị phân của phần đã được chuẩn hóa (chứa các bit của số sau dấu chấm nhị phân). Phần nguyên luôn luôn bằng 1 không lưu.

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

Hình 2-13. Biểu diễn dấu phẩy động

E: Là phần me (Exponent) dùng để điều chỉnh lại vị trí của dấu chấm (phẩy) nhị phân về đúng vị trí thật trong số đó. Vì giá trị E lưu trữ phải luôn không âm, mà trong thực tế số mũ thật sự lại có thể âm, nên sử dụng thủ thuật là dùng một số Bias lấy lại giá trị thực của số mũ

Tùy vào từng loại số thực khác nhau trong máy tính và độ dài bit của trường E mà Bias được xác định giá trị cụ thể khác nhau và Bias luôn là một số dương. Nếu ta chọn 8 bit cho phần mũ thì:

$$E = 2^8/2 - 1 = 2^7 - 1 = 127$$

Như vậy, giá trị Bias là 127.

S: Là bit dấu, nó là bit cao nhất, nếu bit dấu = 1 thì số thực là số âm, ngược lại bit dấu = 0 thì số thực là số dương.

Ví dụ

Số thập phân -4.5 biểu diễn dưới dạng nhị phân là -100.1_B và được chuẩn hóa như sau:

$$-100.1_B = (-1)^1 \times 1.001_B \times 2^2$$

Giá trị lưu trữ theo khuôn dạng S E M sẽ là:

+ Phần biểu diễn dấu chấm động của nó có phần định trị là 1.00_B . Như vậy, giá trị M = 001_B (không được bỏ hai chữ số 0 phía trước, mặc dù khi viết độc lập nó không có nghĩa).

+ Phần mũ là 2, tức là $E - Bias = 2 \rightarrow E = Bias + 2 = 127 + 2 = 129$

+ Bit dấu có S = 1

Ví dụ

Số thập phân 0.0625 biểu diễn dưới dạng nhị phân là 0.0001_B và được chuẩn hóa như sau:

$$0.0001_B = (-1)^0 \times 1.000_B \times 2^{-4}$$

Giá trị lưu trữ theo khuôn dạng S E M sẽ là:

+ Phần biểu diễn dấu chấm động của nó có phần định trị là 1.000_B . Như vậy, giá trị M = 000_B (không được bỏ hai chữ số 0 phía trước, mặc dù khi viết độc lập nó không có nghĩa).

+ Phần mũ là -4, tức là $E - Bias = -4 \rightarrow E = Bias - 4 = 127 - 4 = 123$.

+ Bit dấu có S = 0.

Nếu dùng 8 bit để lưu trữ E thì Bias = 127 và E = 123; tức là phải đảm bảo E có giá trị luôn luôn là dương (không âm).

Như vậy, với công thức biểu diễn số chấm động ở dạng nhị phân trên máy tính như công thức (2.1) ở trên thì phần nguyên luôn luôn bằng 1, phần lẻ sau dấu chấm

(phẩy) động là M; phần mũ là E – Bias, trong đó Bias là giá trị khử dấu cho phần mũ và việc lưu trữ số dấu chấm động trên máy chỉ lưu các giá trị S, E, M.

2.5.3. Kiểu dữ liệu và độ chính xác dữ liệu

2.5.3.1. Kiểu dữ liệu

Trong máy tính, dữ liệu được biểu diễn thông qua số nhị phân (mức điện áp cao/thấp). CPU không thể tự phân biệt được một giá trị nhị phân là kiểu dữ liệu gì (ký tự, số nguyên có dấu, số nguyên không dấu hay là số thực,...) để thực hiện áp dụng các phép tính cho phù hợp. Khi thực hiện một chương trình, để chương trình dịch chỉ ra đúng kiểu dữ liệu và lệnh áp dụng phép tính phù hợp thì trong chương trình phải khai báo kiểu dữ liệu.

Giả sử trong máy tính, một biến $X = CCh$, $Y = 05h$, thì X có thể là một số nguyên không dấu 8 bit (qui đổi ra hệ 10 thì bằng +204). Khi thực hiện phép nhân CPU sẽ yêu cầu thực hiện lệnh MUL (với CPU 8088/8086) và X có thể là một số nguyên có dấu 8 bit (qui đổi ra hệ 10 thì bằng -52), khi thực hiện phép nhân CPU sẽ được yêu cầu thực hiện lệnh IMUL (với CPU 8088/8086). Vậy, để cho chương trình dịch ánh xạ được đến được giá trị thực của dữ liệu thì ta phải khai báo dữ liệu cho biến X .

2.5.3.2. Độ chính xác dữ liệu

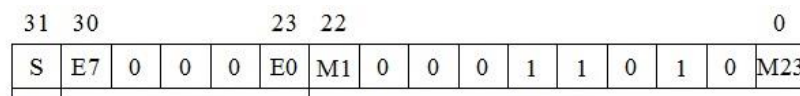
Trong máy tính, số thực cũng được lưu trữ dưới dạng một số nhị phân theo cấu trúc cho số thực chấm động đã được chuẩn hóa theo công thức (2.1).

Có hai kiểu dữ liệu số thực cơ bản thường dùng là kiểu Short real (32 bit) và kiểu Long real (64 bit).

a. Kiểu Short real (dạng số thực ngắn 32 bit)

Kiểu này sử dụng 4 byte để lưu trữ gồm 1 bit dấu, 8 bit phần mũ từ và 23 bit cho phần định trị (không chứa phần nguyên của phần định trị), Bias = 127.

Kiểu Short real được lưu trữ trong máy tính theo cấu trúc sau:



Trong đó, 1 bit 8 bit 23 bit

Hình 2-14. Biểu diễn số dấu chấm (phẩy) động 32 bit trong máy tính

S: Là bit dấu

E: Là phần mũ (Exponent) phần này chiếm 8 bit từ $E_0 \div E_7$ và luôn ≥ 0

M: Là phần định trị (Mantissa) phần này chỉ lưu phần lẻ sau dấu chấm nhị phân, nó chiếm 32 bit từ $M_1 \div M_{23}$.

Độ chính xác dữ liệu là 2^{-127} vì $E \geq 0$ nên $E - \text{Bias} \geq -127$, do đó giá trị dữ liệu biểu diễn nhỏ nhất với kiểu Short real là 2^{-127}

Quy tắc chuyển

+ Bước 1: Đổi số thực đó sang dạng nhị phân

+ Bước 2: Dịch trái hoặc dịch phải để đưa về dạng chuẩn tắc sao cho phần nguyên hay phần trước dấu phẩy của nó = 1 thì được một số M_{B1} và ký hiệu là $M_B \rightarrow M_{B1}$

- Số chữ số dịch N_d

Nếu dịch trái thì $+N_d$

Nếu dịch phải thì $-N_d$

+ Bước 3: Tính $N_d = 127 + N_d$

+ Bước 4: Chuyển số N_d sang N_b và làm tương ứng với các giá trị $E_j \dots E_0$ với các bit của N_b

+ Bước 5: Chọn dạng số thực để biểu diễn

+ Bước 6: Chuyển sang hệ 16

Ví dụ

Biểu diễn số sau -12.652 sang chuẩn IEEE 754/1985

+ Bước 1: chuyển số $M_D = -12.652$ sang nhị phân ta được $M_B = -1100.101$

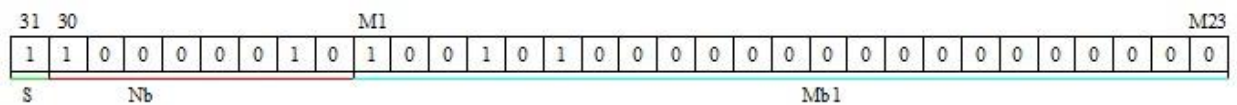
+ Bước 2: Dịch dấu phẩy để đưa về dạng chuẩn tắc ta được $M_B = -1100.101 \Rightarrow M_{B1} = (-1)^1 \times 1.100101 \times 2^3$ (dịch trái 3 bit) $\Rightarrow N_d = +3$

+ Bước 3: Tính

$$N_d = 127 + 3 = 130$$

+ Bước 4: Chuyển số $N_d = 130$ sang nhị phân ta được $N_b = 10000010$

+ Bước 5: Chọn dạng số thực để biểu diễn. Chọn số có độ chính xác đơn dài 32 bit (single)

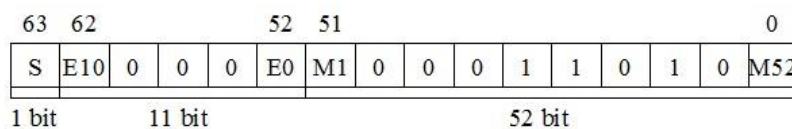


+ Bước 6: Chuyển sang hệ 16: $C14A0000_H$

b. Kiểu Long real (dạng số thực dài 64 bit)

Kiểu này dùng 8 byte để lưu trữ gồm 1 bit dấu, 11 bit phần mũ và 52 bit phần định trị (không chứa phần nguyên của phần định trị), Bias = 1023.

Kiểu Long real được lưu trữ trong máy tính theo cấu trúc sau:



Hình 2-15. Biểu diễn số dấu chấm (phẩy) động 64 bit trong máy tính

Trong đó:

S: Là bit dấu

E: Là phần mũ (Exponent) phần này chiếm 11 bit từ $E_0 \div E_{10}$ và luôn ≥ 0

M: Là phần định trị (Mantissa) phần này chỉ lưu phần lẻ sau dấu chấm nhị phân, nó chiếm 52 bit từ $M_1 \div M_{52}$.

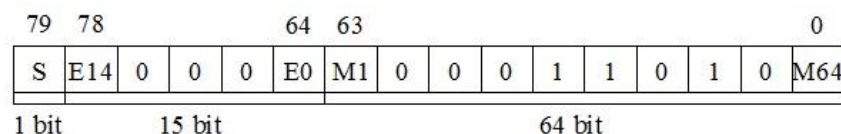
Độ chính xác dữ liệu là 2^{-1023} vì $E \geq 0$ nên $E - \text{Bias} \geq -1023$, do đó giá trị dữ liệu biểu diễn nhỏ nhất với kiểu Long real là 2^{-1023}

c. Kiểu dữ liệu có độ chính xác thay đổi (dạng mở rộng 80 bit)

Để dự phòng cho biểu diễn dữ liệu có độ chính xác cao hơn, người ta đưa ra số thực kiểu Temporary real được lưu trữ trong cấu trúc 80 bit.

Kiểu Temporary real sử dụng 10 byte để lưu trữ gồm 1 bit dấu, 15 bit phần mũ và 64 bit phần định trị (không chứa phần nguyên của phần định trị), Bias = 16383.

Kiểu Temporary real được lưu trữ trong máy tính theo cấu trúc sau:



Hình 2-16. Biểu diễn số dấu chấm (phẩy) động 80 bit trong máy tính

Trong đó:

S: Là bit dấu

E: Là phần mũ (Exponent) phần này chiếm 15 bit từ $E_0 \div E_{14}$ và luôn ≥ 0

M: Là phần định trị (Mantissa) phần này chỉ lưu phần lẻ sau dấu chấm nhị phân, nó chiếm 64 bit từ $M_1 \div M_{64}$.

Độ chính xác dữ liệu là 2^{-16383} vì $E \geq 0$ nên $E - \text{Bias} \geq -16383$, do đó giá trị dữ liệu biểu diễn nhỏ nhất với kiểu Long real là 2^{-16383}

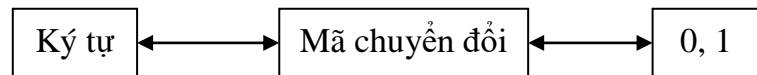
2.6. Biểu diễn ký tự

Không phải mọi số liệu mà máy tính xử lý đều là các con số. Các thiết bị ngoại vi như màn hình hay máy in đều có xu hướng làm việc với các ký tự. Ngoài ra, các chương trình như chương trình xử lý văn bản chuyên làm việc với các dữ liệu kiểu ký tự.

Cũng như tất cả các dữ liệu khác, các ký tự cũng cần phải được mã hóa thành dạng nhị phân để máy tính có thể xử lý chúng. Một kiểu mã hóa

Con người

Máy



Hình 2-17. Sơ đồ mã hóa các ký tự trong máy tính

thông dụng nhất cho các ký tự đó là mã ASCII (American Standard Code for Information Interchange – Mã chuẩn của Mỹ dùng để trao đổi thông tin).

Trước đây mã ASCII được sử dụng trong thông tin với các thiết bị teletype, ngày nay mã ASCII được sử dụng trên tất cả các máy tính cá nhân.

Chúng ta biết rằng máy tính không thể lưu trữ được các ký tự, mà chỉ là lưu trữ các mã hoá của các ký tự đó. Ta có sơ đồ như hình 2-17.

2.6.1. Bộ mã 1 byte

Hệ thống mã ASCII chuẩn sử dụng 7 bit để mã hóa cho một ký tự, do đó có tổng cộng $2^7 = 128$ mã ASCII như vậy Bảng mã ASCII chuẩn mã hoá được 128 ký tự, ký hiệu từ 00h ÷ 7Fh (0 ÷ 127), bao gồm:

- 33 ký tự điều khiển: 00h ÷ 1Fh, 7Fh (0 ÷ 31, 127) dùng để:

- + Điều khiển định dạng (màn hình, máy in)

- + Điều khiển truyền tin

- + Điều khiển phân cách thông tin

- + Một số điều khiển khác

Các mã còn lại, từ 20h ÷ 7E h (32 ÷ 126): mã hoá các ký tự hiển thị, gồm:

- + Các chữ cái in hoa, “A” – “Z”: 41h ÷ 5Ah. Khi bấm “A” thì máy hiểu và mã hoá 0100 0001 và hiển thị lên màn hình “A”.

- + Các chữ cái thường, “a” – “z”: 61h ÷ 7Ah. Khi bấm “a” thì máy mã hoá 0110 0001 và hiển thị “a” lên màn hình.

- + Các chữ số, “0” – “9”: 30h ÷ 39h.

- + Các dấu câu, các phép toán.

- + Một số ký tự đặc biệt khác (\$, #, @, ...).

Hầu hết các máy vi tính chỉ sử dụng các ký tự in được và một số ký tự điều khiển như:

- CR (về đầu dòng)

- LF (xuống dòng)

- HT (tab)

- BS (xóa lùi)

- BEL (đưa ra loa tiếng bít)

Vì mỗi ký tự mã ASCII được mã hóa bằng 7 bit nên mã của một ký tự chứa vừa vặn trong 1 byte với MSB = 0. Các ký tự in được có thể hiển thị ra màn hình hoặc ra máy in, trong khi đó các ký tự điều khiển lại được dùng để điều khiển các thiết bị này.

Để hiển thị ký tự A lên màn hình, chương trình sẽ gửi mã ASCII 41h đến màn hình; còn để lùi con trỏ trở lại đầu dòng thì chương trình sẽ gửi mã ASCII có giá trị là 13 (0Dh – mã ASCII của ký tự điều khiển CR) đến màn hình.

Ngoài ra, máy tính cũng hiển thị một số ký tự đặc biệt ứng với một số mã ASCII không in được.

Hiện nay, ở Việt Nam cũng đã đưa ra và sử dụng một số mã 8 bit (1 byte) để mã hóa các ký tự có dấu giúp cho việc soạn thảo tiếng Việt như bộ mã TCVN3-ABC, VNI Win,...

Mã ASCII mở rộng. Dùng 8 bit để mã hoá, gồm 256 ký tự, từ 00h ÷ FFh, gồm hai phần:

+ 00h ÷ 7Fh: cố định, không thể thay đổi.

+ 80h ÷ FFh: có thể mở rộng (do các nhà chế tạo máy tính và các nhà phân tích phần mềm quy định).

Bảng 2-5. Bảng mã ASCII với 128 ký tự đầu (Bảng mã ASCII chuẩn)

Hexa-decimal	0	1	2	3	4	5	6	7
0	<NUL> 0	<DLE> 16		0 48	@ 64	P 80	' 96	P 112
1	1 1	<DC1> 17	! 33	1 49	A 65	Q 81	a 97	q 113
2	2 2	<DC2> 18	" 34	2 50	B 66	R 82	b 98	r 114
3	♥ 3	<DC3> 19	# 35	3 51	C 67	S 83	c 99	s 115
4	♦ 4	<DC4> 20	\$ 36	4 52	D 68	T 84	d 100	t 116
5	♣ 5		% 37	5 53	E 69	U 85	e 101	u 117
6	♠ 6	<SYN> 22	& 38	6 54	F 70	V 86	f 102	v 118
7	<BEL> 7		' 39	7 55	G 71	W 87	g 103	w 119
8	<BS> 8	<CAN> 24	(40	8 56	H 72	X 88	h 104	x 120
9	<HT> 9	 25) 41	9 57	I 73	Y 89	i 105	y 121
A	<LF> 10		★ 42	: 58	J 74	Z 90	j 106	z 122
B	<VT> 11	<ESC> 27	+ 43	; 59	K 75	[91	k 107	{ 123
C	<FF> 12	<FS> 28	, 44	< 60	L 76	\ 92	l 108	 124
D	<CR> 13		- 45	= 61	M 77] 93	m 109	} 125
E	<SO> 14		. 46	> 62	N 78	^ 94	n 110	~ 126
F	<SI> 15		/ 47	? 63	O 79	_ 95	o 111	 127

Bảng 2-6. Bảng mã ASCII với 128 ký tự nửa sau (Bảng mã ASCII mở rộng)

Hexa-decimal	8	9	A	B	C	D	E	F
0	à 128	ã 144	á 160				α 224	≡ 112
1	ÿ 129	ô 145	ó 161				õ 225	± 241
2	É 130	ă 146	ú 162				γ 226	≥ 242
3	Ö 131	Õ 147	Ű 163				π 227	≤ 243
4	ø 132	ű 148	ę 164				Σ 228	∫ 244
5	À 133	ũ 149	ẹ 165				σ 229	Ј 245
6	ô 134	Ū 150	ê 166				μ 230	÷ 246
7	õ 135	ұ 151	° 167				τ 231	≈ 247
8	ò 136	ÿ 152	ı 168					• 248
9	ớ 137	ệ 153					θ 233	· 249
A	ó 138	ĩ 154					Ω 234	– 250
B	Ū 139	Ă 155	ă 171				δ 235	
C	ợ 140	Ê 156	ă 172				∞ 126	^ 252
D	õ 141	Ő 157	 173				φ 127	2 253
E	Ä 142		ô 174				∈ 128	■ 254
F	Ă 143	f 159	ă 175				∩ 129	

2.6.2. Bộ mã 2 byte

Unicode (Universal Code) là bộ mã tiêu chuẩn quốc tế đa ngôn ngữ (tiêu chuẩn mã hóa ký tự 16 bit do Unicode Consortium phát triển trong thời gian từ năm 1988 đến năm 1991). Do nó sử dụng 2 byte để thể hiện từng ký tự

Unicode cho phép thể hiện hầu hết mọi ngôn ngữ viết trên thế giới (trong đó có tiếng Việt) bằng một tập ký tự đơn nhất.

Song song với tổ chức Unicode Consortium còn có tổ chức ISO (tổ chức tiêu chuẩn quốc tế) cũng nghiên cứu một bộ mã đa ngôn ngữ được sử dụng trong ngành Công nghệ thông tin là ISO/IEC 10646. Unicode và ISO từ năm 1993 đã thống nhất cùng nhau phát triển và đồng nhất hai bộ mã ở miền 16 bit. Unicode là bộ mã 16 bit có 65.536 ô mã.

2.6.3. Cấu trúc biểu diễn dữ liệu

Trong thực tế có nhiều kiểu dữ liệu cần biểu diễn trong máy tính, mỗi kiểu dữ liệu lại được lưu trữ biểu diễn theo các phương pháp khác nhau với cấu trúc khác nhau. Thông thường có hai loại dữ liệu là dữ liệu cơ bản và dữ liệu do người sử dụng

định nghĩa. Như vậy, cấu trúc biểu diễn dữ liệu rất đa dạng và hệ thống kiến trúc máy tính phải đáp ứng được điều này. Cho nên, ngày nay các hệ thống máy tính có thể biểu diễn, xử lý mọi bài toán trong cuộc sống, giải quyết các tính toán trong toán học, xử lý đồ họa, quản lý mô phỏng, xử lý âm thanh,...

Ví dụ

Biểu diễn số thực trong máy tính phải biểu diễn thông qua số thực có dấu chấm động và máy tính lưu trữ giá trị của R theo cấu trúc bao gồm:

- Giá trị M (chỉ lấy phần lẻ nhị phân)
- Giá trị của bit dấu S và số mũ E , còn cơ số 2 là mặc định (trong máy tính dữ liệu luôn luôn được biểu diễn ở dạng cơ số 2)

Trong một đơn vị, để quản lý một người thì phải lưu trữ các thông tin về người đó. Thông tin về bản thân của mỗi người cần được lưu trữ trong một vùng nhớ và vùng nhớ chứa nội dung của một dữ liệu mang đặc thù riêng thường được gọi là cấu trúc dữ liệu kiểu bản ghi. Tức là, dữ liệu kiểu bản ghi này có cấu trúc và bao gồm nhiều nhiều kiểu dữ liệu khác nhau, mỗi kiểu dữ liệu con trong nó không nhất thiết có độ dài phải có độ dài bằng nhau và mỗi kiểu dữ liệu con trong nó có thể có cấu trúc lưu trữ khác nhau.

Chương 3

BỘ XỬ LÝ TRUNG TÂM

3.1. Kiến trúc cơ bản của một máy tính điện tử số và chu trình xử lý lệnh

3.1.1. Kiến trúc cơ bản của một máy tính điện tử số

Các thành phần của CPU gồm:

- + Khối điều khiển (Control Unit – CU).
- + Khối tính toán số học và logic (Arithmetic and Logic Unit).
- + Bus trong CPU (CPU Internal Bus).
- + Các thanh ghi của CPU bao gồm:
 - Thanh ghi tích lũy A (Accumulator).
 - Bộ đếm chương trình PC (Program Counter).
 - Thanh ghi lệnh IR (Instruction Register).
 - Thanh ghi địa chỉ bộ nhớ MAR (Memory Address Register).
 - Thanh ghi đệm dữ liệu MBR (Memory Buffer Register).
 - Các thanh ghi tạm thời Y và Z.
 - Thanh ghi cờ FR (Flag Register).

3.1.1.1. Khối điều khiển (Control Unit – CU)

Khối điều khiển (Control Unit – CU) là một trong các khối quan trọng nhất của CPU. CU đảm nhiệm việc điều khiển toàn bộ các hoạt động của CPU theo xung nhịp đồng hồ. CU sử dụng xung nhịp đồng hồ để đồng bộ các đơn vị chức năng trong CPU và giữa CPU với các bộ phận bên ngoài.

(1) Lệnh từ thanh ghi lệnh IR.

(2) Giá trị các cờ trạng thái của ALU.

(3) Xung nhịp đồng hồ CLK và CU sản sinh hai nhóm tín hiệu đầu ra.

- Nhóm tín hiệu điều khiển các bộ phận bên trong CPU (Internal control signal) và Nhóm tín hiệu điều khiển các bộ phận bên ngoài CPU (External control signal).

3.1.1.2. Khối tính toán số học và logic (Arithmetic and Logic Unit)

Khối số học và logic (Arithmetic and Logic Unit – ALU) đảm nhiệm chức năng tính toán trong CPU. ALU bao gồm một loạt các đơn vị chức năng con để thực hiện các phép toán số học trên số nguyên và logic:

+ Bộ cộng (ADD), bộ trừ (SUB), bộ nhân (MUL), bộ chia (DIV)

+ Các bộ dịch (SHIFT) và quay (ROTATE)

+ Bộ phủ định (NOT), bộ và (AND), bộ hoặc (OR) và bộ hoặc loại trừ (XOR)

3.1.1.3. Các Bus trong CPU

Bus trong CPU (Internal bus) là kênh giao tiếp giữa các bộ phận bên trong CPU, cụ thể giữa bộ điều khiển CU với các thanh ghi và bộ tính toán ALU. Bus trong hỗ trợ kênh giao tiếp song công (full duplex) và cung cấp giao diện để kết nối với bus ngoài (bus hệ thống). So với bus ngoài, bus trong thường có băng thông lớn hơn và có tốc độ nhanh hơn.

3.1.2. Chu trình xử lý lệnh

3.1.2.1. Nguyên tắc chung

Việc thực hiện chương trình thực chất là sự lặp lại quá trình nạp lệnh, giải mã lệnh và thực thi lệnh. Thời gian tính từ khi nạp lệnh đến khi thi hành lệnh hoàn thành được gọi là *một chu kỳ lệnh*. Sau mỗi một chu kỳ lệnh, CPU kiểm tra xem có yêu cầu ngắt hợp lệ hay không (Check for Interrupt?); nếu có, CPU nhận lệnh từ thủ tục ngắt (Process Interrupt); sau khi hoàn thành thủ tục phục vụ ngắt, CPU bắt đầu thực hiện

chu kỳ lệnh mới từ nơi nó đã bị ngắt (thực hiện chu kỳ lệnh đứng ngay sau lệnh gọi ngắt).

3.1.2.2. Các thao tác cơ bản khi thực hiện lệnh thao tác dữ liệu bộ nhớ

Giai đoạn nhận lệnh: Là giai đoạn chung cho tất cả mọi lệnh thao tác dữ liệu. Nếu chương trình chưa kết thúc thì nội dung của bộ đếm chương trình PC được chuyển vào thanh ghi địa chỉ bộ nhớ MAR ($MAR \leftarrow PC$) và một yêu cầu đọc vị trí nhớ có trong địa chỉ MAR (Read M(MAR)) được gửi tới bộ nhớ. Dữ liệu yêu cầu (lệnh cần thực hiện) tại vị trí nhớ M(MAR) được chuyển tới thanh ghi đệm bộ nhớ MBR ($MBR \leftarrow M(MAR)$). Lúc này nội dung của PC được tăng lên một giá trị bằng độ dài lệnh vừa nạp là I ($PC \leftarrow PC + I$) để chuẩn bị cho việc nạp lệnh tiếp theo được thực hiện. Phần opcode của lệnh vừa nạp trong MBR được gửi vào thanh ghi lệnh (IR) và phần địa chỉ (Addr) trong lệnh vừa nạp được chuyển vào thanh ghi địa chỉ bộ nhớ (MAR).

Giai đoạn giải mã lệnh: Nội dung của thanh ghi lệnh được đưa vào đơn vị điều khiển ($CU \leftarrow IR$). Tại đây, tùy theo mã lệnh mà CPU biết cần thực hiện thao tác gì (nạp toán hạng – Load, lưu toán hạng ra bộ nhớ – Store, cộng – Add, and logic – And, lệnh nhảy – Jump, lệnh dịch phải – Shift,...).

Giai đoạn thi hành lệnh: trong giai đoạn này, các vi thao tác (Micro – operations) tùy thuộc vào lệnh đã nạp. Ta thấy rằng, mỗi một lệnh được thực thi bởi một chuỗi thao tác như sau:

- + Lệnh Load yêu cầu hai thao tác: nạp toán hạng từ vị trí nhớ có địa chỉ chứa trong thanh ghi địa chỉ bộ nhớ MAR vào MBR ($MBR \leftarrow M(MAR)$) và sau đó nội dung của MBR được đưa vào thanh ghi Acc ($Acc \leftarrow MBR$);

- + Lệnh Store yêu cầu hai thao tác: chuyển nội dung của Acc vào MBR ($MBR \leftarrow Acc$) và nội dung của MBR được đưa ghi ra vị trí nhớ có địa chỉ chứa trong MAR ($M(MAR) \leftarrow MBR$);

- + Lệnh Add có thể yêu cầu ba thao tác: đầu tiên, nạp toán hạng thứ hai từ vị trí nhớ có địa chỉ chứa trong MAR đưa vào thanh ghi dữ liệu MBR ($MBR \leftarrow M(MAR)$), tiếp theo là chuyển nội dung của MBR vào thanh ghi tạm TMP ($TMP \leftarrow MBR$), cuối cùng là thực hiện cộng nội dung của Acc với TMP và kết quả cất trong Acc ($Acc \leftarrow Acc + TMP$).

3.1.2.3. Các thao tác cơ bản được thực hiện lệnh vào/ ra dữ liệu trực tiếp với thiết bị vào/ra

Giai đoạn nhận lệnh: Là giai đoạn chung cho các lệnh vào/ra dữ liệu, giai đoạn này chỉ khác với giai đoạn nhận lệnh của lệnh xử lý dữ liệu là phần địa chỉ (Addr) của lệnh vừa nạp trong MBR được chuyển vào thanh ghi địa chỉ vào/ra (IOAR).

Giai đoạn giải mã lệnh: Nội dung của thanh ghi lệnh được đưa vào đơn vị điều khiển ($CU \leftarrow IR$). Tại đây, tùy theo lệnh mà CPU biết cần thực hiện thao tác gì? (nhập dữ liệu – IN, xuất dữ liệu – Out).

Giai đoạn thi hành lệnh: Với lệnh vào/ra dữ liệu trực tiếp, CPU thường dùng lệnh IN/OUT. Lệnh IN đọc nội dung của cổng vào có địa chỉ chứa trong IOAR đưa vào thanh ghi IOBR ($IOBR \leftarrow IO(IOAR)$), sau đó dữ liệu trong IOBR được chuyển vào Acc ($Acc \leftarrow IOBR$). Còn lệnh OUT, đưa nội dung của Acc vào thanh ghi IOBR ($IOBR \leftarrow Acc$), sau đó dữ liệu từ thanh ghi IOBR được chuyển ra cổng ra có địa chỉ chứa trong IOAR ($IO(IOAR) \leftarrow IOBR$).

Trong trường hợp chu kỳ lệnh với ngắt:

Tại thời điểm hoàn thành giai đoạn thi hành lệnh, một thao tác kiểm tra được thực hiện để quyết định yêu cầu ngắt có được chấp nhận hay không. Nếu yêu cầu ngắt được chấp nhận, các thao tác sau được thực hiện:

```
t1:   MBR ← PC
t2:   MAR ← Save_Address
      PC ← Routine_Address
t3:   M(MAR) ← MBR
```

3.2. Các thanh ghi của CPU

Các thanh ghi trong CPU là loại bộ nhớ tốt nhất trong hệ thống nhớ và là cách giúp CPU truy cập dữ liệu nhanh nhất. Toán hạng được sử dụng thường nằm trong một thanh ghi và được chỉ ra trong một phần của câu lệnh đã được giải mã. Câu lệnh đã được giải mã thuộc tập lệnh của CPU, nội dung của các thanh ghi thường được đưa vào từ bộ nhớ và có thể nói việc sử dụng các thanh ghi làm cho hiệu năng thực thi chương trình là cao nhất. Việc làm này được thực hiện bởi một chương trình dịch trong giai đoạn biên dịch chương trình (gọi là giai đoạn sinh mã lệnh).

Số lượng thanh ghi tùy thuộc vào bộ vi xử lý cụ thể. Như vậy, số lượng thanh ghi càng nhiều thì càng làm tăng hiệu năng của CPU.

Dung lượng của các thanh ghi thường được xác định bởi số bit mà chúng có, ví dụ thanh ghi 8 bit, thanh ghi 32 bit. Ngày nay, các thanh ghi trong CPU được coi như là tập các thanh ghi, chúng được xây dựng từ các mạch flip – flops với các phương pháp khác nhau và tùy theo các loại máy tính khác nhau.

Trong giáo trình này chúng tôi đề cập đến một số thanh ghi theo chức năng của chúng. Hiện nay, trong máy tính tồn tại một số loại thanh ghi sau:

3.2.1. Thanh ghi tích lũy (Accumulator – Acc)

Thanh ghi tích lũy A tham gia vào phần lớn các phép tính. Nó cất giữ toán hạng (operand) hoặc các kết quả phép tính của ALU. Các phép tính số học và logic phải sử dụng ALU và thanh ghi tích lũy A. Vì vậy, thanh ghi tích lũy A là một thanh ghi chính của vi xử lý, và là một thành phần trong mô hình lập trình của vi xử lý. Độ dài từ xử lý của vi xử lý phụ thuộc vào độ dài của thanh ghi tích lũy. Cho nên, thanh ghi tích lũy của các loại vi xử lý 8-bit có độ dài 8-bit (1 byte). Tuy nhiên, ở một số loại vi xử lý có thể có những thanh ghi tích lũy có ghi tích lũy độ dài kép. Thậm chí một số loại vi xử lý còn có một số thanh ghi tích lũy, ví dụ một thanh ghi tích lũy A, thanh ghi tích lũy khác gọi là B. Trong trường hợp này phải có lệnh chỉ cho bộ vi xử lý chuyển kết quả tính của ALU vào thanh ghi tích lũy A, và một lệnh chuyển dữ liệu vào thanh ghi tích lũy B. Tương tự, có lệnh xóa thanh ghi tích lũy A và lệnh xóa thanh ghi tích lũy B.

3.2.2. Thanh ghi lệnh (Instruction Register – IR)

Thanh IR chứa lệnh đang thực hiện. Trong chu trình đọc lệnh (Instruction cycle) từ bộ nhớ, lệnh được nạp vào IR thông qua bus dữ liệu trong. IR như là bộ đệm duy trì nội dung lệnh và đầu ra của thanh ghi lệnh đưa tới bộ giải mã lệnh (Instruction decoder) để tạo ra chuỗi các tín hiệu điều khiển thực hiện lệnh. Độ dài của IR phụ thuộc vào từng bộ vi xử lý, trong một số loại vi xử lý, độ dài của nó bằng độ dài của từ xử lý. Cũng có thể dựa vào số lượng lệnh của tập lệnh đoán được độ dài thanh ghi lệnh, vì 2^N = số lượng tối đa các lệnh (Maximal number of instruction), trong đó N = số bit của thanh ghi lệnh. Đôi khi độ dài của IR chỉ ngắn bằng 3 đến 4 bit.

3.2.3. Thanh ghi địa chỉ

3.2.3.1. Bộ đếm chương trình (Program Counter - PC)

Bộ đếm chương trình là một trong những thanh ghi quan trọng nhất trong vi xử lý. Chương trình là tập hợp liên tục các chỉ dẫn (Instructions) lưu giữ trong bộ nhớ bên

ngoài vi xử lý. Mỗi một chỉ dẫn (lệnh máy) trong chương trình phải được vi xử lý thực hiện theo một thứ tự mong muốn của người lập trình. Bộ đếm chương trình có nhiệm vụ giữ tiến trình thực hiện chương trình theo mong muốn đó. Trong tiến trình thực hiện chương trình, bộ đếm chương trình luôn chứa địa chỉ ngăn nhớ lưu giữ chỉ dẫn tiếp theo mà vi xử lý phải thực hiện.

Bộ đếm chương trình thường có độ dài lớn hơn độ dài từ mã vi xử lý thực hiện. Đối với vi xử lý 8 bit, bộ đếm chương trình có độ dài 16 bit. Vì nó dùng để chứa địa chỉ ngăn nhớ nên không gian mà có thể địa chỉ được là 64 Kbyte từ 8 bit, tức là một bộ nhớ với dung lượng $64 \text{ Kbyte} = 2^{16}$ (hay 65.536). Chỉ có một số ít chỉ dẫn làm việc với bộ đếm chương trình.

Trước khi vi xử lý có thể thực hiện một chương trình, bộ đếm chương trình phải được nạp một giá trị. Giá trị này là địa chỉ của ngăn nhớ chứa lệnh đầu tiên phải thực hiện trong chương trình. Thường thì giá trị này là toàn các bit 0, hay toàn các bit 1.

- Toàn các bit 0 nghĩa là địa chỉ ngăn nhớ đầu tiên của bộ nhớ (0000h). Toàn các bit 1 nghĩa là địa chỉ của ngăn nhớ cuối cùng của bộ nhớ (FFFFh). Trong ngăn nhớ này phải chứa nội dung của lệnh đầu tiên của chương trình. Tất cả những ngăn nhớ còn lại của chương trình có thể chứa lệnh hay dữ liệu là tùy thuộc vào cấu trúc chương trình.

3.2.3.2. Thanh ghi con trỏ dữ liệu (DP – Data Pointer)

Chứa địa chỉ của ngăn nhớ dữ liệu mà BXL cần truy cập.

3.2.3.3. Con trỏ ngăn xếp (SP – Stack Pointer)

Ngăn xếp là bộ nhớ có cơ chế truy cập theo kiểu LIFO (Last in first out), nghĩa là byte dữ liệu nào ghi vào ngăn xếp cuối cùng thì sẽ được đọc ra đầu tiên. Ngăn xếp có thể là một vùng nào đó của bộ nhớ chính, hay là một mảng thanh ghi riêng biệt. Nó làm nhiệm vụ lưu giữ những thông tin phải dùng đi dùng lại nhiều lần trong quá trình thực hiện chương trình, đặc biệt là khi có nhiều chương trình con.

Giống như bộ đếm chương trình, con trỏ ngăn xếp phải có độ dài đủ để chứa giá trị địa chỉ của một ngăn nhớ trong bộ nhớ. Trong vi xử lý 16 bit nó có độ dài 16 bit. Nội dung của con trỏ ngăn xếp là địa chỉ trỏ đến vùng nhớ tiếp theo. Ngăn xếp luôn được truy cập từ đỉnh (TOP). Trong hầu hết các loại vi xử lý, con trỏ ngăn xếp giảm địa chỉ tiếp theo thấp hơn sau khi nó được sử dụng. Điều này cho phép người lập trình xây dựng ngăn xếp từ đỉnh xuống đáy trong bộ nhớ. Các lệnh thao tác với ngăn xếp là các lệnh hai byte.

3.2.3.4. Thanh ghi cơ sở và thanh ghi chỉ số (Base Register & Index Register)

+ Thanh ghi cơ sở (BR – Base Register)

Thanh ghi cơ sở chứa địa chỉ của ngăn nhớ cơ sở (ngăn nhớ gốc tương đối) còn được gọi là địa chỉ đoạn (Segment).

+ Thanh ghi chỉ số (IR – Index Register)

Chứa độ lệch của địa chỉ giữa ngăn nhớ mà BXL cần truy cập so với ngăn nhớ cơ sở, còn được gọi là địa chỉ Offset.

+ Địa chỉ của ngăn nhớ cần truy cập = địa chỉ cơ sở + chỉ số.

3.2.4. Các thanh ghi dữ liệu tạm thời (Temporary data Register – TR)

Trong ALU không có bộ đệm, do đó khi tính toán, các dữ liệu phải được lấy từ hoặc đưa trở lại bus dữ liệu trong theo thời điểm nào đó. Những việc này, nếu thiếu thanh ghi dữ liệu tạm thời thì không thể thực hiện. Ngoài ra trong quá trình xử lý cần sự ổn định dữ liệu trong thời gian đủ ngắn đủ để cho ALU thực hiện được xong phép tính. Có hai thanh ghi dữ liệu tạm thời ở hai cổng IN của ALU cho phép ALU thực hiện các phép tính với hai từ dữ liệu 8 bit.

3.2.5. Thanh ghi địa chỉ bộ nhớ và logic

Thanh ghi địa chỉ bộ nhớ (MAR – Memory Address Register) có nội dung là địa chỉ ngăn nhớ mà trong ngăn nhớ đó chứa lệnh hay dữ liệu mà vi xử lý cần đến hoặc phải ghi dữ liệu vào đó. Vì là địa chỉ bộ nhớ nên thanh ghi địa chỉ bộ nhớ phải có độ dài 16 bit. Các đường ra của thanh ghi địa chỉ bộ nhớ được điều khiển nối với bus địa chỉ 16 bit của hệ thống, qua đó giá trị nội dung của thanh ghi được dẫn đến các mạch logic giải mã chọn địa chỉ ngăn nhớ của bộ nhớ hay các cổng vào/ra kết nối với các thiết bị ngoại vi.

3.2.6. Các thanh ghi trạng thái (Status Register)

Thanh ghi trạng thái của vi xử lý còn được gọi là thanh ghi cờ (Flag register), và dùng để ghi kết quả của các lệnh kiểm tra, so sánh khi thực hiện chương trình. Một số phép tính thực hiện với ALU và với các thanh ghi có thể thiết lập hoặc xóa một số bit trong thanh ghi trạng thái. Các bit trong thanh ghi trạng thái còn gọi là các bit cờ. Mỗi một bit của nó có một ý nghĩa và bị tác động tùy theo lệnh máy thực hiện.

Sử dụng các bit của thanh ghi trạng thái có thể thực hiện rẽ nhánh chương trình bằng các lệnh nhảy có điều kiện (nhảy theo giá trị các bit cờ), và bộ đếm chương trình phải được nạp địa chỉ của đích nhảy tới. Sự rẽ nhánh được thực hiện nếu giá trị cờ thỏa mãn.

Trong tập lệnh của vi xử lý, các lệnh được mô tả và tác động đến giá trị của các thanh ghi trạng thái, nhờ đó người lập chương trình có thể thực hiện kiểm tra giá trị các bit trạng thái và rẽ nhánh của chương trình. Nội dung của thanh ghi trạng thái có thể được đọc ra nhờ chương trình thông qua bus dữ liệu bên trong, nhưng nó không thể nhận dữ liệu từ bus dữ liệu bên trong. Do đó, thanh ghi trạng thái chỉ có thể đọc được mà không ghi được bởi lập trình.

3.2.7. Các thanh ghi con trỏ, chỉ số

+ SP (Stack pointer) con trỏ ngăn xếp: địa chỉ đỉnh ngăn xếp. SP cho phép truy xuất dễ dàng các địa chỉ trong đoạn ngăn xếp SS (Stack segment). Giá trị trong SP mô tả phải là Offset của địa chỉ ngăn xếp kế tiếp so với địa chỉ hiện tại đang được lưu trong SS.

+ BP (Base pointer) con trỏ cơ sở: mô tả Offset tính từ SS nhưng còn được sử dụng truy nhập DL trong SS.

+ I (Index) thanh ghi chỉ số: lưu địa chỉ Offset đối với những lệnh truy nhập DL cất trong đoạn DL.

3.2.8. Thanh ghi đoạn

Bộ nhớ được chia thành các đoạn logic (Segment) dài 64kb. CPU có thể truy nhập 1 lần tới 4 đoạn. Địa chỉ đoạn chứa trong thanh ghi đoạn.

+ Thanh ghi đoạn mã CS (Code Segment) nhận diện địa chỉ bắt đầu của đoạn chương trình hiện hành trong bộ nhớ.

+ DS (Data Segment) đoạn DL : địa chỉ bắt đầu đoạn số liệu.

+ SS (Stack Segment) đoạn ngăn xếp: địa chỉ logic đoạn ngăn xếp.

+ EX (Extra Segment) đoạn mở rộng: Đ/c DL các chuỗi.

3.2.9. Thanh ghi đa năng

Vi xử lý 8 bit có 6 thanh ghi đa năng (B, C, D, E, H, L). Cũng giống như thanh ghi tích lũy, đây là những thanh ghi 8 bit. Các loại vi xử lý khác nhau có thể có nhiều thanh ghi đa năng hơn, nhiều ứng dụng hơn. Các thanh ghi đa năng có thể được sử dụng cho nhiều mục đích khác nhau và có nhiều lệnh sử dụng các thanh ghi này: chứa các toán hạng để thực hiện các phép tính số học, logic, dịch.... Có thể thực hiện các lệnh xử lý các bit trong từng thanh ghi đa năng.

Các thanh ghi đa năng có thể ghép lại thành từng đôi. Ví dụ, các cặp thanh ghi BC, DE và HL có chức năng thống nhất: chúng có thể được thao tác như các thanh ghi 16 bit riêng biệt. Điều này cho phép thực hiện các lệnh với các dữ liệu 16 bit.

Cặp thanh ghi HL có thể được dùng để chứa địa chỉ ngăn nhớ, và nội dung của HL có thể được tăng lên 1 (Increment) hoặc giảm đi 1 (Decrement) để trở tới ngăn nhớ tiếp theo.

3.2.10. Thanh ghi cờ (Flag Register - FR)

Dùng để chứa các cờ, mỗi cờ là một bit và phản ánh thông tin về trạng thái làm việc của CPU hoặc trạng thái kết quả sau khi thực hiện lệnh. Thanh ghi cờ tạo ra mối quan hệ logic giữa các lệnh được thực hiện trong chương trình.

Có 9 trong số 16 bit của thanh ghi này được sử dụng, mỗi bit có thể được thiết lập hay xóa để chỉ thị kết quả của mỗi thao tác trước đó hoặc trạng thái hiện thời của bộ xử lý.

- + CF Carry: nhớ
- + PF Parity: chẵn lẻ
- + ZF Zero: kết quả phép toán = 0
- + SF Sign: 0 dương, 1 âm.
- + OF Overflow: tràn

3.3. Đơn vị ghép nối Bus

Kết nối và trao đổi thông tin giữa bus bên trong (Internal bus) và bus bên ngoài (External bus). Một máy tính bao gồm các bộ phận hay các đơn vị của ba thành phần chính: CPU, hệ thống nhớ, thiết bị vào ra, được liên lạc với nhau. Về thực chất, máy tính được coi là một mạng của các đơn vị cơ bản. Hơn nữa cần phải có các đường để kết nối các đơn vị với nhau.

Mặc dù có rất nhiều cách thiết kế Bus khác nhau, nhưng trên bất kỳ cách nào các đường Bus cũng phân loại thành ba nhóm chính: Bus dữ liệu, Bus địa chỉ và Bus điều khiển, ngoài ra có thể có một số đường cung cấp năng lượng cho các module tham gia Bus.

3.4. Tập lệnh máy

3.4.1. Lệnh máy tính là gì?

Có thể nói, nếu coi phần mạch điện tử của CPU là “phần xác” thì tập lệnh (Instruction Set) chính là “phần hồn” của bộ não máy tính. Nhờ có tập lệnh, CPU có khả năng lập trình được để thực hiện các công việc hữu ích cho người dùng.

Việc thực hiện lệnh có thể được chia thành các pha (Phase) hay giai đoạn (Stage). Một lệnh có thể được thực hiện theo 5 giai đoạn sau:

1. Đọc lệnh (Instruction fetch - IF): lệnh được đọc từ bộ nhớ về CPU.
2. Giải mã (Instruction decode - ID): CPU giải mã lệnh.
3. Thực hiện lệnh (Instruction execution - EX): CPU thực hiện lệnh.
4. Tham nhập bộ nhớ trong hoặc nhảy (MEM: Memory access).
5. Lưu kết quả (Write back - WB): kết quả thực hiện lệnh (nếu có) được lưu vào bộ nhớ.

3.4.2. Ngắt (Interrupt)

Ngắt 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 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 cũng là phần khó thực hiện nhất trong bộ điều khiển. Để nhận biết được một ngắt 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.

Trên thực tế, ngắt được chia thành hai loại: ngắt cứng và ngắt mềm:

3.4.2.1. Ngắt cứng

Còn gọi là ngắt ngoài vì do nguyên nhân bên ngoài. VXL có các lối vào dành cho ngắt ngoài. Khi có tín hiệu vào lối vào này, chương trình VXL đang thực hiện sẽ bị dừng.

3.4.2.2. Ngắt mềm

Hay ngắt bên trong do lệnh của chương trình, do VXL gặp các lệnh gây ra ngắt hoặc tình huống đặc biệt khi thực hiện lệnh (ngắt logic) và ngắt của hệ điều hành.

3.4.3. Dạng lệnh và các dạng địa chỉ toán hạng

3.4.3.1. Dạng lệnh

Dạng tổng quát lệnh của máy tính bao gồm 2 phần chính:

+ Mã thao tác (opcode – operation code).

+ Địa chỉ của các toán hạng (Addresses of Operands).

Mã thao tác	Địa chỉ toán hạng
-------------	-------------------

- Mã thao tác (Operation Code: Opcode): Mã chỉ ra thao tác mà bộ vi xử lý cần phải thực hiện.

- Địa chỉ toán hạng (Operand Address): Chỉ ra nơi chứa các toán hạng mà mã thao tác sẽ tác động.

- Toán hạng nguồn: dữ liệu vào của thao tác.

- Toán hạng đích: dữ liệu ra của thao tác.

Mỗi lệnh có một mã lệnh riêng và được biểu diễn bằng một số bit. Chẳng hạn, mã lệnh của CPU Intel 8086 được biểu diễn bởi 6 bit. Mỗi lệnh có thể có một hoặc nhiều toán hạng và mỗi toán hạng là một địa chỉ. Có 5 dạng toán hạng của lệnh: 3 địa chỉ, 2 địa chỉ, 1 địa chỉ, 1,5 địa chỉ và 0 địa chỉ.

3.4.3.2. Các dạng địa chỉ toán hạng

a. Toán hạng dạng 3 địa chỉ

Dạng:

Opcode addr1, addr2, addr3

Mỗi địa chỉ addr1, addr2, addr3 tham chiếu đến một ô nhớ hoặc một thanh ghi.

b. Toán hạng dạng 2 địa chỉ

Dạng:

Opcode addr1, addr2

Mỗi địa chỉ addr1, addr2 tham chiếu đến một ô nhớ hoặc một thanh ghi.

c. Toán hạng dạng 1 địa chỉ

Dạng:

Opcode addr2

Địa chỉ addr2 tham chiếu đến một ô nhớ hoặc một thanh ghi. Ngoài ra, thanh ghi tích lũy Racc được sử dụng và có vai trò như addr1 trong toán hạng dạng 2 địa chỉ.

d. Toán hạng dạng 1,5 địa chỉ

Dạng:

Opcode addr1, addr2

Một địa chỉ tham chiếu đến một ô nhớ và địa chỉ còn lại tham chiếu đến một thanh ghi. Dạng 1,5 địa chỉ là dạng toán hạng hỗn hợp giữa ô nhớ và thanh ghi.

e. Toán hạng dạng 0 địa chỉ

Toán hạng 0 địa chỉ thường được sử dụng trong các lệnh thao tác với ngăn xếp: PUSH và POP.

3.4.4. Các nhóm lệnh của CPU

3.4.4.1. Nhóm lệnh chuyển dữ liệu

a. Nhóm lệnh chuyển dữ liệu đa dụng

+ Lệnh MOV dst, src: chuyển nội dung toán hạng src vào toán hạng dst.

Toán hạng nguồn src có thể là thanh ghi (reg), bộ nhớ (mem) hay giá trị tức thời (immed); toán hạng đích dst có thể là reg hay mem.

Lệnh MOV có thể có các trường hợp sau:

Reg8	← reg8 MOV AL, AH
Reg16	← reg16 MOV AX, BX
Mem8	← reg8 MOV [BX], AL
Reg8	← mem8 MOV AL, [BX]
Mem16	← reg16 MOV [BX], AX
Reg16	← mem16 MOV AX, [BX]
Reg8	← immed8 MOV AL, 04h
Mem8	← immed8 MOV mem [BX], 01h
Reg16	← immed16 MOV AL, 0F104h
Mem16	← immed16 MOV mem [BX], 0101h
SegReg	← reg16 MOV DS, AX
SegReg	← mem16 MOV DS, mem
Reg16	← segreg MOV AX, DS
Mem16	← segreg MOV [BX], DS

- Lệnh MOV không ảnh hưởng đến các cờ.

- Không thể chuyển trực tiếp dữ liệu giữa hai ô nhớ mà phải thông qua một thanh ghi.

MOV AX, mem1

MOV mem2, AX

- Không thể chuyển giá trị trực tiếp vào thanh ghi đoạn.

MOV AX, 1010h

MOV DS, AX

- Không thể chuyển trực tiếp giữa 2 thanh ghi đoạn.

- Không thể dùng thanh ghi CS làm toán hạng đích.

+ Lệnh XCHG dst, src: (Exchange) hoán chuyển nội dung 2 toán hạng.

Toán hạng chỉ có thể là reg hay mem.

- Lệnh XCHG không ảnh hưởng đến các cờ.

- Không thể dùng cho các thanh ghi đoạn.

+ Lệnh PUSH src: cất nội dung một thanh ghi vào stack. Toán hạng chỉ có thể là reg16.

+ Lệnh POP dst: lấy dữ liệu 16 bit từ stack đưa vào toán hạng dst.

Ta có thể dùng nhiều lệnh PUSH để cất dữ liệu vào stack nhưng khi dùng lệnh POP để lấy dữ liệu ra thì phải dùng theo thứ tự ngược lại:

PUSH AX

PUSH BX

PUSH CX

POP CX

POP BX

POP AX

+ Lệnh XLAT [src]: chuyển nội dung của ô nhớ 8 bit vào thanh ghi AL.
Địa chỉ ô nhớ xác định bằng cặp thanh ghi DS:BX (nếu không chỉ ra src) hay src, địa chỉ offset chứa trong thanh ghi AL. Lệnh XLAT tương đương với các lệnh:

```
MOV    AH, 0
MOV    SI, AX
MOV    AL, [BX+SI]
```

b. Nhóm lệnh chuyển địa chỉ

+ Lệnh LEA reg16, mem16: (Load Effective Address) chuyển địa chỉ offset của toán hạng bộ nhớ vào thanh ghi reg16.

Lệnh này sẽ tương đương với MOV reg16, OFFSET mem16

+ Lệnh LDS reg16, mem32: (Load pointer using DS) chuyển nội dung bộ nhớ toán hạng mem32 vào cặp thanh ghi DS:reg16.

Lệnh LDS AX,mem tương đương với:

```
MOV    AX, mem
MOV    BX, mem + 2
MOV    DS, BX
```

+ Lệnh LES reg16, mem32: (Load pointer using ES) giống như lệnh LDS nhưng dùng cho thanh ghi ES.

c. Nhóm lệnh chuyển cờ hiệu

+ Lệnh LAHF: (Load AH from flag) chuyển các cờ SF, ZF, AF, PF và CF vào các bit 7, 6, 4, 2 và 0 của thanh ghi AH (3 bit còn lại không đổi).

+ Lệnh SAHF: (Store AH into flag) chuyển các bit 7, 6, 4, 2 và 0 của thanh ghi AH vào các cờ SF, ZF, AF, PF và CF.

+ Lệnh PUSHF: chuyển thanh ghi cờ vào stack.

+ Lệnh POPF: lấy dữ liệu từ stack chuyển vào thanh ghi cờ.

d. Nhóm lệnh chuyển dữ liệu qua cổng

Mỗi I/O port giao tiếp với CPU sẽ có một địa chỉ 16 bit cho nó. CPU gửi hay nhận dữ liệu từ cổng bằng cách chỉ đến địa chỉ cổng đó. Tùy theo chức năng mà cổng có thể: chỉ đọc dữ liệu (Input port), chỉ ghi dữ liệu (Output port) hay có thể đọc và ghi dữ liệu (Input/Output port).

+ Lệnh IN: đọc dữ liệu từ cổng và đưa vào thanh ghi AL:

```
IN     AL, port8
IN     AL, DX
```

- Nếu địa chỉ port chỉ có 8 bit thì có thể đưa giá trị trực tiếp vào, nếu là 16 bit thì phải thông qua thanh ghi AX.

+ Lệnh OUT: ghi dữ liệu trong thanh ghi AL ra cổng:

```
OUT    port8, AL
OUT    DX, AL
```

3.4.4.2. Nhóm lệnh chuyển điều khiển chương trình

a. Lệnh nhảy không điều kiện JMP

```
JMP label
```

```
JMP reg/mem
```

Lệnh JMP dùng để chuyển điều khiển chương trình từ vị trí này sang vị trí khác (thay đổi nội dung cặp thanh ghi CS:IP).

b. Lệnh nhảy có điều kiện

Lệnh nhảy có điều kiện chỉ sử dụng cho các nhãn nằm trong khoảng từ -127 đến 128 byte so với vị trí của lệnh.

+ Lệnh JA label: (Jump if Above). Nếu CF = 0 và ZF = 0 thì JMP label.

- + Lệnh JAE label: (Jump if Above or Equal). Nếu $CF = 0$ thì JMP label.
- + Lệnh JB label: (Jump if Below). Nếu $CF = 1$ thì JMP label.
- + Lệnh JBE label: (Jump if Below or Equal). Nếu $CF = 1$ hoặc $ZF = 1$ thì JMP label.
- + Lệnh JNA label: (Jump if Not Above). Giống lệnh JBE.
- + Lệnh JNAE label: (Jump if Not Above or Equal). Giống lệnh JB.
- + Lệnh JNB label: (Jump if Not Below). Giống lệnh JAE.
- + Lệnh JNBE label: (Jump if Not Below or Equal). Giống lệnh JA.
- + Lệnh JG label: (Jump if Greater). Nếu $SF = OF$ và $ZF = 0$ thì JMP label.
- + Lệnh JGE label: (Jump if Greater or Equal). Nếu $SF = OF$ thì JMP label.
- + Lệnh JL label: (Jump if Less). Nếu $SF \neq OF$ thì JMP label.
- + Lệnh JLE label: (Jump if Less or Equal). Nếu $CF \neq OF$ hoặc $ZF = 1$ thì JMP label.
- + Lệnh JNG label: (Jump if Not Greater). Giống lệnh JLE.
- + Lệnh JNGE label: (Jump if Not Greater or Equal). Giống lệnh JL.
- + Lệnh JNL label: (Jump if Not Less). Giống lệnh JGE.
- + Lệnh JNLE label: (Jump if Not Less or Equal). Giống lệnh JG.
- + Lệnh JC label: (Jump if Carry). Giống lệnh JB.
- + Lệnh JNC label: (Jump if Not Carry). Giống lệnh JNB.
- + Lệnh JZ label: (Jump if Zero). Nếu $ZF = 1$ thì JMP label.
- + Lệnh JE label: (Jump if Equal). Giống lệnh JZ.
- + Lệnh JNZ label: (Jump if Not Zero). Nếu $ZF = 0$ thì JMP label.
- + Lệnh JNE label: (Jump if Equal). Giống lệnh JNZ.
- + Lệnh JS label: (Jump on Sign). Nếu $SF = 1$ thì JMP label.
- + Lệnh JNS label: (Jump if No Sign). Nếu $SF = 0$ thì JMP label.
- + Lệnh JO label: (Jump on Overflow). Nếu $OF = 1$ thì JMP label.
- + Lệnh JNO label: (Jump if No Overflow). Nếu $OF = 0$ thì JMP label.
- + Lệnh JP label: (Jump on Parity). Nếu $PF = 1$ thì JMP label.
- + Lệnh JNP label: (Jump if No Parity). Nếu $PF = 0$ thì JMP label.
- + Lệnh JCXZ label: (Jump if CX Zero). Nếu $CX = 1$ thì JMP label.

c. Lệnh so sánh

CMP left(reg/mem), right(reg/mem/immed). Lệnh CMP dùng để so sánh nội dung 2 toán hạng, kết quả chứa vào thanh ghi cờ và không làm thay đổi nội dung các toán hạng.

d. Các lệnh vòng lặp

+ Lệnh LOOP:

LOOP label

Mô tả:

$CX = CX - 1$

Nếu $CX \neq 0$ thì JMP label.

+ Lệnh LOOPE:

LOOPE label

Mô tả:

$CX = CX - 1$

Nếu $(ZF = 1)$ và $(CX \neq 0)$ thì JMP label.

+ Lệnh LOOPZ:

Giống lệnh LOOPE

+ Lệnh LOOPNE:

LOOPNE label

Mô tả:

$CX = CX - 1$

Nếu $(ZF = 0)$ và $(CX \neq 0)$ thì JMP label.

+ Lệnh **LOOPNZ**:

Giống lệnh LOOPNE

e. Lệnh liên quan đến chương trình con

+ Lệnh **CALL**:

Lệnh CALL dùng để gọi một chương trình con, có thể là near hay far.

CALL label ; Gọi chương trình con tại vị trí xác định bởi nhãn label.

CALL reg/mem ; Gọi chương trình con tại vị trí xác định; trong reg/mem

+ Lệnh **RET**: (return)

RET [n]

RETN [n]

RETF [n]

Lệnh RET dùng để kết thúc chương trình con, điều khiển sẽ được đưa về địa chỉ trước khi gọi chương trình con. RETN để kết thúc chương trình con dạng near và RETF dùng để kết thúc chương trình con dạng far.

Trong trường hợp lệnh RET có hằng số n theo sau thì sẽ cộng với thanh ghi SP giá trị n (n phải là số chẵn). Lệnh này dùng để loại bỏ một số tham số chương trình con sử dụng ra khỏi stack.

3.4.4.3. Nhóm lệnh xử lý số học

a. Xử lý phép cộng

+ Lệnh **ADD dst, src**:

$dst \leftarrow dst + src$

Toán hạng src có thể là reg, mem hay immed còn toán hạng dst là reg hay mem.

- Không thể cộng trực tiếp 2 thanh ghi đoạn.

- Lệnh ADD ảnh hưởng đến các cờ sau:

+ Cờ CF: = 1 khi kết quả phép cộng có nhớ hay có mượn.

+ Cờ AF: = 1 khi kết quả phép cộng có nhớ hay có mượn đối với 4 bit thấp.

+ Cờ PF: = 1 khi kết quả phép cộng có tổng 8 bit thấp là một số chẵn.

+ Cờ ZF: = 1 khi kết quả phép cộng là 0.

+ Cờ SF: = 1 nếu kết quả phép cộng là một số âm.

+ Cờ OF: = 1 nếu kết quả phép cộng bị sai dấu, nghĩa là vượt ra ngoài phạm vi lớn nhất hay nhỏ nhất mà số có dấu có thể chứa trong toán hạng dst.

+ Lệnh **ADC dst, src**: (Add with Carry)

$dst \leftarrow dst + src + CF$

Lệnh ADC thường dùng để cộng các số lớn hơn 16 bit.

+ Lệnh **INC dst**: (Increment)

$dst \leftarrow dst + 1$

Dst có thể là reg hay mem.

+ Lệnh **AAA**: (ASCII Adjust for Addition)

Hiệu chỉnh kết quả phép cộng 2 số BCD dạng không nén (mỗi chữ số BCD lưu bằng 1 byte).

b. Xử lý phép trừ

+ Lệnh **SUB dst, src**:

$dst \leftarrow dst - src$

Toán hạng src có thể là reg, mem hay immed còn toán hạng dst chỉ có thể là reg hay mem.

- Không thể trừ trực tiếp thành ghi đoạn.
- Ảnh hưởng đến các cờ AF, CF, OF, PF, SF và ZF.

+ Lệnh **SBB dst,src**:

$dst \leftarrow dst - src - CF$

Lệnh ADC thường dùng để trừ các số lớn hơn 16 bit.

+ Lệnh **DEC dst**: (decrement)

$dst \leftarrow dst - 1$

dst là reg hay mem. Lệnh DEC ảnh hưởng đến các cờ AF, OF, PF, SF, ZF.

+ Lệnh **NEG dst**:

$dst \leftarrow -dst$

dst là reg hay mem.

Lệnh NEG ảnh hưởng đến các cờ:

CF = 1 nếu nội dung kết quả là số khác 0.

SF = 1 nếu nội dung kết quả là số âm khác 0.

PF = 1 nếu tổng 8 bit thấp là một số chẵn.

ZF = 1 nếu nội dung kết quả là 0.

OF = 1 nếu nội dung toán hạng dst là 80h (dạng byte) hay 8000h (dạng word).

c. Xử lý phép nhân

+ Lệnh **MUL src**:

- Nếu src là reg hay mem 8 bit: $AX \leftarrow AL * src$

- Nếu src là reg hay mem 16 bit: $DX:AX \leftarrow AX * src$

Lệnh MUL chỉ ảnh hưởng đến cờ CF và OF.

+ Lệnh **IMUL src**:

Giống như lệnh MUL nhưng kết quả là số có dấu.

+ Lệnh **AAM**: (Ascii Adjust for Multiple)

Hiệu chỉnh kết quả phép nhân 2 số BCD dạng không nén, lệnh AAM thực hiện chia AL cho 10, lưu phần thương vào AL và phần dư vào AH. Lệnh AAM ảnh hưởng đến các cờ PF, SF và ZF.

d. Xử lý phép chia

+ Lệnh **DIV src**:

Nếu src là reg/mem 8 bit: $AL \leftarrow AX \text{ DIV } src$ và $AH \leftarrow AX \text{ MOD } src$.

Nếu src là reg/mem 16 bit: $AX \leftarrow DX:AX \text{ DIV } src$ và $DX \leftarrow DX:AX \text{ MOD } src$.

Src.

Lệnh DIV không ảnh hưởng đến các cờ nhưng xảy ra tràn trong các trường hợp sau:

- Chia cho 0.
- Thương lớn hơn 256 đối với dạng 8 bit.
- Thương lớn hơn 65536 đối với dạng 16 bit.

+ Lệnh **IDIV src**:

Giống như lệnh DIV nhưng kết quả là số có dấu. Các trường hợp tràn:

- Chia cho 0
- Thương nằm ngoài khoảng (-128,127) đối với dạng 8 bit.
- Thương nằm ngoài khoảng (-32767,32768) đối với dạng 16 bit.

+ Lệnh **AAD**: (Ascii Adjust for Division)

Hiệu chỉnh kết quả phép chia 2 số BCD dạng không nén. Lưu ý rằng lệnh AAD phải được thực hiện trước lệnh chia. Sau khi thực hiện chia thì phải hiệu chỉnh lại dạng BCD bằng cách dùng lệnh AAM.

+ Lệnh **CBW**: (Convert Byte to Word)

Nếu $AL < 80h$ thì $AH = 0$, ngược lại $AH = 0FFh$.

Lệnh CBW dùng để chuyển số nhị phân có dấu 8 bit thành số nhị phân có dấu 16 bit.

+ Lệnh **CWD**: (Convert Word to Double word).

- Nếu $AX < 8000h$ thì $DX = 0$, ngược lại $DX = 0FFFFh$

Lệnh CWD dùng để chuyển số nhị phân có dấu 16 bit thành số nhị phân có dấu 32 bit chứa trong $DX:AX$.

e. Dịch chuyển và quay

+ Lệnh **SHL**: (Shift Logical Left)

SHL dst,1

SHL dst,CL

Dịch trái 1 bit hay CL bit.

$CF \leftarrow dst7 \leftarrow dst6 \dots \leftarrow dst0 \leftarrow 0$

+ Lệnh **SHR**: (Shift Logical Right)

SHR dst,1

SHR dst,CL

Dịch phải 1 bit hay CL bit.

$0 \rightarrow dst7 \rightarrow dst6 \dots \rightarrow dst0 \rightarrow CF$

+ Lệnh **SAL**: giống SHL

+ Lệnh **SAR**:

Giống như lệnh SHR nhưng giá trị bit $dst7$ không thay đổi, nghĩa là $dst7 \rightarrow dst7 \rightarrow dst6 \dots \rightarrow dst0 \rightarrow CF$

+ Lệnh **ROL**: (Rotate Left)

ROL dst,1

ROL dst,CL

Quay trái 1 bit hay CL bit.

$CF \leftarrow dst7 \leftarrow dst6 \dots \leftarrow dst0 \leftarrow dst7$

+ Lệnh **ROR**: (Rotate Right)

ROR dst,1

ROR dst,CL

Quay phải 1 bit hay CL bit.

$dst0 \rightarrow dst7 \rightarrow dst6 \dots \rightarrow dst0 \rightarrow CF$

+ Lệnh **RCL**: (Rotate though Carry Left)

RCL dst,1

RCL dst,CL

Quay trái 1 bit hay CL bit.

$CF \leftarrow dst7 \leftarrow dst6 \dots \leftarrow dst0 \leftarrow CF$

+ Lệnh **RCR**: (Rotate though Carry Right)

RCR dst,1

RCR dst,CL

Quay phải 1 bit hay CL bit.

$CF \rightarrow dst7 \rightarrow dst6 \dots \rightarrow dst0 \rightarrow CF$

f. Các lệnh logic

+ Lệnh **AND**:

AND dst,src
dst \leftarrow dst AND src
CF \leftarrow 0, OF \leftarrow 0

Src là reg, mem hay immed còn dst là reg, mem.

+ Lệnh **OR**:

OR dst,src
dst \leftarrow dst OR src
CF \leftarrow 0, OF \leftarrow 0

+ Lệnh **XOR**:

XOR dst,src
dst \leftarrow dst XOR src
CF \leftarrow 0, OF \leftarrow 0

+ Lệnh **NOT**:

NOT dst
dst \leftarrow NOT dst
Lệnh NOT không ảnh hưởng đến các cờ.

+ Lệnh **TEST**:

TEST dst,src

Lệnh TEST thực hiện phép toán AND 2 toán hạng nhưng chỉ ảnh hưởng đến các cờ và không ảnh hưởng đến toán tử.

3.4.4.4. Nhóm lệnh xử lý chuỗi

Bao gồm các lệnh sau:

+ Lệnh MOVS: chuyển dữ liệu từ vùng nhớ này sang vùng nhớ khác.

+ MOVSB: chuyển 1 byte từ vị trí chỉ đến bởi SI đến vị trí chỉ bởi DI. Nếu DF = 0 thì SI \leftarrow SI + 1, DI \leftarrow DI + 1 còn nếu DF = 1 thì SI \leftarrow SI - 1, DI \leftarrow DI - 1.

+ MOVSW: chuyển 1 word từ vị trí chỉ đến bởi SI đến vị trí chỉ bởi DI. Nếu DF = 0 thì SI \leftarrow SI + 2, DI \leftarrow DI + 2 còn nếu DF = 1 thì SI \leftarrow SI - 2, DI \leftarrow DI - 2.

+ Lệnh CMPS: so sánh nội dung 2 vùng nhớ

+ CMPSB: so sánh 1 byte tại vị trí chỉ đến bởi SI và tại vị trí chỉ bởi DI. Nếu DF = 0 thì SI \leftarrow SI + 1, DI \leftarrow DI + 1 còn nếu DF = 1 thì SI \leftarrow SI - 1, DI \leftarrow DI - 1.

+ CMPSW: so sánh 1 word tại vị trí chỉ đến bởi SI và tại vị trí chỉ bởi DI. Nếu DF = 0 thì SI \leftarrow SI + 2, DI \leftarrow DI + 2 còn nếu DF = 1 thì SI \leftarrow SI - 2, DI \leftarrow DI - 2.

+ Lệnh SCAS: tìm một phần tử trong vùng nhớ, địa chỉ vùng nhớ xác định bằng cặp thanh ghi ES:DI, giá trị cần tìm đặt trong thanh ghi AL, nếu tìm thấy thì ZF = 1. Giá trị của DI và SI thay đổi giống như trên.

+ Lệnh LODS: đưa một byte hay word có địa chỉ xác định bởi cặp thanh ghi DS:SI vào thanh ghi AL hay AX. Giá trị của DI và SI thay đổi giống như trên.

+ Lệnh STOS: chuyển nội dung của AL hay AX vào vùng nhớ xác định bởi cặp thanh ghi ES:DI. Giá trị của DI và SI thay đổi giống như trên.

3.5. Các phương pháp định địa chỉ

Trong việc xác định các định dạng của lệnh, ta cần phải chú ý đến một số yêu cầu như:

+ Số lệnh đã được xác định trước cho tập lệnh.

+ Khả năng địa chỉ và phương thức xác định địa chỉ.

+ Sự thuận lợi cho giải mã lệnh

+ Kiểu của trường bit trong lệnh (cố định hay thay đổi)

+ Giá phải trả cho yêu cầu phần cứng để giải mã lệnh và thi hành lệnh

Một lệnh khi đã được giải mã ở dạng mã máy thường có định dạng sau:

OP-CODE	ADDRESS (ES)
---------	--------------

Trong đó:

OP-CODE: Là phần mã thi hành (ngôn ngữ máy).

ADDRESS (ES): Có thể có 0, 1, 2 hoặc 3 trường địa chỉ để xác định địa chỉ của các toán hạng có trong lệnh.

Có 3 nhóm định vị toán hạng:

- Định vị tức thời.
- Định vị thanh ghi.
- Định vị bộ nhớ.

3.7. Giới thiệu một số bộ vi xử lý điển hình

3.7.1. Bộ vi xử lý 8088/8086

Bộ vi xử lý 8088/8086 thực thi chương một chương trình bằng cách lặp đi lặp lại 3 thao tác chính là lấy lệnh (Fetch), giải mã lệnh (Decode) và thực hiện (Execute). Sơ đồ khối của bộ vi xử lý 8088/8086 gồm 2 đơn vị chính (Hình 3.26):

- EU (Execution Unit): Đơn vị (khối) thực hiện lệnh.
- BIU (Bus Interface Unit): Đơn vị giao tiếp bus (khối tương thích bus) để điều khiển bus hiệu quả hơn.

3.7.1.1. EU

Đơn vị EU của 8088 và 8086 giống nhau, bao gồm ALU, thanh ghi cờ, các thanh ghi đệm và các thanh ghi đa năng. Các bus dữ liệu bên trong của EU đều là 16 bit. EU không kết nối trực tiếp với bên ngoài, nó nhận lệnh từ hàng đợi lệnh bên trong BUI. Nếu là lệnh cần truy nhập bộ nhớ hoặc cổng vào/ra (I/O port - thiết bị ngoại vi) thì EU yêu cầu BIU lấy hoặc gửi dữ liệu. Tất cả các địa chỉ mà EU thao tác đều là 16 bit, khi gửi sang BIU thì BIU sẽ thực hiện tính toán để tạo địa chỉ vật lý 20 bit và phát ra các chân địa chỉ của chip.

3.7.1.2. Các thanh ghi của EU

Có 8 thanh ghi, là thành phần nhớ có tốc độ truy nhập rất cao. Bao gồm các thanh ghi đa năng Ax, Bx, Cx, Dx. Mỗi thanh ghi 16 bit có thể được phân chia thành 2 thanh ghi 8 bit làm việc độc lập nhau.

+ AX (Accumulator, Acc): Thanh chứa, chứa tạm thời dữ liệu (toán hạng, kết quả phép toán như nhân, chia và được coi là Acc). AX có thể được phân chia thành AH (Phần cao) và AL (Phần thấp).

+ BX (Basic): Thanh ghi cơ sở, thường được dùng để chứa địa chỉ cơ sở cần truy nhập trong lệnh XLAT. BX có thể được phân chia thành BH (Phần cao) và BL (Phần thấp).

+ CX (Count): Bộ đếm, thường được dùng để đếm số lần lặp của công việc (số lần lặp trong các vòng lặp). CX có thể được phân chia thành CH (Phần cao) và CL (Phần thấp). CX thường được dùng để chứa số lần lặp trong các lệnh LOOP, còn CL thường chứa số lần dịch hoặc quay trong các lệnh dịch quay thanh ghi.

+ DX (Data): Thanh ghi dữ liệu, sử dụng để chứa toán hạng, kết quả. DX cùng AX dùng để chứa toán hạng hoặc kết quả của các phép nhân, chia số 16 bit. DX còn chứa địa chỉ cổng trong các lệnh vào ra trực tiếp (IN, OUT). DX có thể được phân chia thành DH (Phần cao) và DL (Phần thấp).

3.7.1.3. Các thanh con trỏ

Các thanh ghi SP, BP là các thanh ghi con trỏ không tách rời.

+ SP (Stack Pointer): Thanh ghi con trỏ ngăn xếp, trỏ vào đỉnh hiện thời của ngăn xếp nằm trong đoạn ngăn xếp SS (Nó luôn kết hợp với thanh ghi SS). Ta có địa

chỉ logic SS: SP. Sau mỗi thao tác cất một word vào stack (thao tác Push) thì SP tự động giảm 2 đơn vị, còn sau thao tác lấy một word ra khỏi stack (thao tác Pop), SP được tự động tăng 2 đơn vị. BP (Base stack Pointer): Con trỏ cơ sở, luôn trỏ vào một dữ liệu cụ thể nằm trong đoạn ngăn xếp SS. Ta có địa chỉ logic SS: BP.

+ SI (Source Index): Thanh ghi chỉ số nguồn (hay nguồn), chỉ vào dữ liệu nằm trong đoạn DS. Ta có địa chỉ logic DS: SI.

+ DI (Destination Index): Thanh ghi chỉ số đích, chỉ dữ liệu trong đoạn DS. Ta có địa chỉ logic DS: DI.

3.7.1.4. Khối ALU

Làm nhiệm vụ thực hiện các lệnh số học và logic.

+ Số học: +, -, *, /, so sánh, đảo dấu.

+ Logic: NOT, AND, OR, XOR.

3.7.1.5. Thanh ghi cờ:

Đây là thanh ghi 16 bit, mỗi bit được sử dụng để thể hiện một trạng thái của bộ vi xử lý tại một thời điểm nhất định trong quá trình thực hiện chương trình (dãy các câu lệnh), nhưng chỉ dùng 9 bit đối với bộ vi xử lý 8088/8086. Mỗi bit đó được gọi là một cờ (flag). Giá trị của mỗi cờ được biểu diễn bằng các ký hiệu gọi nhớ như cách biểu diễn của chương trình Debug của DOS.

OF DF IF TF SF ZF AF PF CF

CF (Carry Flag): Cờ nhớ (cờ tràn).

CF = 1 (CY - Carry) khi kết quả phép toán vượt (tràn) khuôn khổ biểu diễn (có nhớ) mượn lên trên (từ MSB), CF = 0 (NC - No Carry) trong trường hợp ngược lại. Cờ này thường được sử dụng khi thực hiện các phép cộng, trừ các số nhiều byte.

PF (Parity Flag): Cờ chẵn lẻ.

PF = 1 (PE - Parity Even) khi số bit 1 trong kết quả phép toán (hay các phép vận chuyển dữ liệu) là chẵn, PF = 0 (PO - Parity Odd) trong trường hợp ngược lại.

AF (Carry Flag): Cờ nhớ phụ (trần phụ - cờ hỗ trợ).

Cờ này có ý nghĩa khi ta làm việc với số BCD. AF = 1 (AC - Auxiliary Carry) khi có nhớ hoặc mượn từ một số BCD thấp (4 bit thấp) sang một số BCD cao (4 bit cao) và AF = 0 (NA - No Auxiliary carry) trong trường hợp ngược lại.

SF (Sign Flag): Cờ dấu. Trong bộ vi xử lý 8088/8086 các số âm được biểu diễn dưới dạng số bù 2, nên phải dùng cờ SF để chỉ thị dấu của kết quả.

SF = 1 (NG - NeGative), khi kết quả phép toán là một số âm, SF = 0 (PL- PPlus) trong trường hợp ngược lại.

ZF (Zero Flag): Cờ rỗng.

ZF = 1 (ZR - ZeRo) khi kết quả phép toán = 0, ZF = 0 (NZ-Non Zero) trong trường hợp ngược lại.

OF (Overflow Flag): Cờ tràn.

OF = 1 (OV-OVerflow) khi kết quả là số bù 2 vượt khuôn khổ biểu diễn (trần số học, hay nói cách khác: khi cộng hai số cùng dấu mà kết quả là một số trái dấu thì OF = 1), OF = 0 (NV-Non oVerflow) trong trường hợp ngược lại (cờ này làm việc với số có dấu).

Cờ điều khiển

Cờ trạng thái phụ thuộc kết quả phép toán, còn với cờ điều khiển ta có thể thiết lập nhờ lệnh.

IF (Interrupt Flag): Cờ ngắt.

IF = 1 (EI-Enable Interrup), CPU cho phép ngắt, IF = 0 (DI-Disable Interrup) CPU không cho phép ngắt (cấm) các loại ngắt che được (Maskable)..

TF (Trap Flag): Cờ bẫy.

TF = 1 CPU làm việc trong chế độ chạy từng lệnh, thường dùng để gỡ rối chương trình (debug). Sau khi thực hiện xong mỗi lệnh, bộ vi xử lý sẽ phát ra một lệnh ngắt (INT) để có kiểm tra chương trình.

DF (Direction Flag): Cờ hướng.

Điều khiển hướng xử lý đối với thao tác chuỗi. DF = 1 (DN-DowN) thì các lệnh vận chuyển dữ liệu hay xử lý chuỗi sẽ thao tác lùi từ phải đến trái (địa chỉ cao đến địa chỉ thấp). DF=0 (UP) trong trường hợp ngược lại (thao tác các phần tử từ địa chỉ thấp đến địa chỉ cao).

3.7.1.6. BIU

Hàng đợi: Là tập thanh ghi.

Với 8086 hàng đợi lệnh là 6 byte, 8088 hàng đợi lệnh là 4 byte nên chứa được tối đa là 4 lệnh. CPU chứa tập thanh ghi theo kiểu LIFO.

Cơ chế:

Đọc lệnh (lấy lệnh).

Giải mã lệnh.

Thực hiện.

Thanh ghi IP: Thanh ghi con trỏ lệnh, trỏ vào lệnh tiếp theo chuẩn bị được thực hiện nằm trong đoạn CS (CS: IP). Sau khi đọc 1 byte, IP tự động tăng thêm 1. Như vậy thực tế thì cặp CS:IP mới là con trỏ lệnh vì nó chứa địa chỉ đầy đủ của một lệnh trong bộ nhớ.

Thanh ghi đoạn:

CS (Code Segment): Thanh ghi đoạn mã, là thanh ghi địa chỉ đoạn mã lệnh, chứa địa chỉ cơ sở (địa chỉ đoạn) của chương trình đang thực hiện.

DS (Data Segment): Thanh ghi đoạn dữ liệu, là thanh ghi địa chỉ đoạn dữ liệu, chứa địa chỉ đoạn của vùng dữ liệu mà chương trình đang thực hiện sử dụng. Vùng này thường chứa các biến của chương trình.

SS (Stack Segment): Thanh ghi đoạn ngăn xếp, là thanh ghi địa chỉ đoạn bộ nhớ ngăn xếp (stack) của chương trình đang chạy.

ES (Extra Segment): Thanh ghi đoạn dữ liệu phụ, là thanh ghi địa chỉ đoạn dữ liệu bổ sung mà chương trình đang thực hiện sử dụng. Vùng này cũng thường chứa các biến của chương trình.

Không gian địa chỉ

8088 có 20 bit địa chỉ, trong khi đó các thanh ghi quản lý bộ nhớ là 16 bit. Để xác định một ngăn nhớ có địa chỉ 20 bit, nó dùng 2 thanh ghi 16 bit để xác định địa chỉ.

Một thanh ghi lưu địa chỉ đoạn.

Một thanh ghi lưu địa chỉ lệnh (Offset).

Một địa chỉ yêu cầu trong một lệnh nào đó là địa chỉ logic có dạng:

DS:SI

DS:DI

DS:XX

Địa vật lý (phải là số 20 bit) được xác định như sau:

Địa chỉ vật lý = Segment * 16 + Offset

3.7.2. Bộ vi xử lý 80286

3.7.2.1. Đặc điểm chung và các chân tín hiệu.

Vi xử lý 80286 được dùng đầu tiên trong các máy IBM PC/AT và ở một số loại IBM PS/2. 80286 cũng là loại vi xử lý 16 bit như 8086 nghĩa là có thể nhập xuất 2 byte

dữ liệu một lần trên bus số liệu rộng 16 bit. Nhưng 80286 lại có bus địa chỉ rộng lên tới 24 bit nên có thể quản lý được tới 16 Mbyte nhớ vật lý. Cấu trúc bên trong cơ bản giống như 8086 nhưng 80286 có tới 134.000 transistor trên chip và hoạt động ở tần số xung đồng hồ hệ thống cao hơn là 8 MHz đến 25 MHz, vì xử lý 80286 cho tốc độ truyền dữ liệu lên tới 2 triệu lệnh trên mỗi giây (2 MIPS – 2 milion instructions per second).

80286 tương thích hoàn toàn với 8086, điều đó có nghĩa là tất cả chương trình nào được viết trong 8086 sẽ chạy tốt hơn trong 80286. Thêm nữa, 80286 chạy chương trình nhanh hơn và linh hoạt hơn 8086. Sự cải tiến quan trọng nhất đối với 80286 là nó hỗ trợ cho hoạt động đa nhiệm và có hai chế độ hoạt động:

- Chế độ thực (Real mode).
- Chế độ bảo vệ (Protected mode).

Trong chế độ thực, 80286 được lập trình giống như 8086 và có thể thâm nhập một dải địa chỉ lên tới 1 Mbyte giống 8086. Trong chế độ bảo vệ, 80286 có thể địa chỉ hóa tới 16 Mbyte nhớ và có thể dữ trữ một lượng bộ nhớ định mức cho mỗi chương trình khả thi nhằm đề phòng vùng nhớ này có thể được dùng bất kỳ một chương trình khác. Chế độ bảo vệ hỗ trợ cho hoạt động đa nhiệm. Điều này có nghĩa rằng một vài chương trình có thể chạy đồng thời mà không xảy ra sự rủi ro về một chương trình tình cờ thay đổi các nội dung của các vùng nhớ của chương trình khác. Một hệ điều hành được khi sử dụng chế độ bảo vệ 80286 có thể định vị bộ nhớ trong số một vài nhiệm vụ thì hiệu quả hơn hệ điều hành dùng 8086.

Vì xử lý 80286 có 68 chân và có thể chia thành các nhóm chân tín hiệu như:

- $D_0 \div D_{15}$: bus dữ liệu 16 bit
- $A_0 \div A_{23}$: bus địa chỉ 24 bit
- \overline{BHE} : Có chức năng giống như trong 8086
- M/\overline{IO} : nếu = 1, truy xuất bộ nhớ, nếu = 0, truy xuất vào/ra
- $\overline{COD}/\overline{INTA}$: mã lệnh trả lời ngắt
- HOLD, HLDA: yêu cầu và trả lời treo bus. HOLD là tín hiệu vào, do ngoại vi muốn là chủ bus sinh ra. Nếu chấp nhận thì vi xử lý đưa tín hiệu HLDA về mức tích cực 1.

- $\overline{PERE}/\overline{PEACK}$: yêu cầu và trả lời đồng bộ xử lý.

- \overline{BUSY} , \overline{ERROR} , \overline{BUSY} : báo cho vi xử lý biết tiến trình đang bận, vi xử lý chỉ thực hiện các lệnh ESC và WAIT để chờ.

- RESET: đưa mạch logic bên trong vi xử lý về trạng thái ban đầu, tín hiệu có tác dụng khi chuyển từ mức 0 lên 1 và giữ ở đó ít nhất là 16 nhịp đồng hồ.

- CLK2: tần số nhịp vào chân này sẽ được chia đôi trong vi xử lý thành tín hiệu PCLK, một chu kỳ PCLK được tạo bởi 2 chu kỳ CLK2, được gọi là pha 1 và pha 2.

Các chân HOLD, HLDA, INTR, \overline{INTA} , NMI, \overline{READY} và \overline{LOCK} cũng như \overline{RESET} về cơ bản có chức năng hoạt động giống như trong 8086. Chip 82284 được dùng để phát ra các xung đồng hồ cho 80286 và đồng bộ các tín hiệu RESET và \overline{READY} . Bốn tín hiệu PEREQ, \overline{PEACK} , \overline{BUSY} và $\overline{TEST1}$ được dùng để ghép nối 80286 với các phân tử rộng của vi xử lý như bộ đồng xử lý toán 80287. Dạng sóng trong chu kỳ máy của 80286 cũng giống trong 8086.

3.7.2.2. Chế độ hoạt động bảo vệ

Sau khi 80286 được khởi động nó sẽ khởi phát chạy trong chế độ thực giống ở 8086. Trong chế độ này 80286 có thể định địa chỉ tới 1 Mbyte nhớ vật lý. Tuy nhiên,

do được tăng cường cải tiến về các kiểu định địa chỉ đường ống và các phần cứng khác nên nó sẽ chạy nhanh hơn 8086 ở cùng một tốc độ xung nhịp. Muốn chuyển sang chế độ bảo vệ phải đặt bit cho phép bảo vệ trong một thanh ghi mới có ở 80286 và thanh ghi từ trạng thái máy MSW (Machine Status Word) lên mức logic cao. Khác với chế độ thực, trong chế độ bảo vệ việc quản lý bộ nhớ có phức tạp và chặt chẽ hơn. Sang đến vi xử lý thế hệ sau từ Intel 80386 trở đi chế độ bảo vệ càng được phát huy rõ nét trong các vi xử lý 32 bit. Vì vậy, phần này chỉ trình bày vắn tắt về chế độ bảo vệ và các vấn đề liên quan trong 80286.

3.7.3. Bộ vi xử lý 80386

3.7.3.1. Đặc điểm chung và các thanh ghi.

Bắt đầu từ 80386 các vi xử lý của Intel chuyển sang thế giới 32 bit. Tức là chúng có cả bus số liệu và bus địa chỉ đều có độ rộng 32 bit. Các vi xử lý này hỗ trợ rất mạnh cho các hệ điều hành đa nhiệm. 80386 được dùng trong các máy IBM PS/2 80 đầu tiên và nhiều máy nhái khác. Có thể liệt kê vài đặc điểm chính của 80386 như sau:

- Bus dữ liệu rộng 32 bit nên có thể truy xuất 4 byte số liệu một lúc.
- Bus địa chỉ rộng 32 bit nên có thể quản lý được $2^{32} = 4$ Gbyte nhớ vật lý.
- Với trên 275000 transistor và tốc độ xung đồng hồ 33 MHz, 80386 có tốc độ 11 MIPS, tức là 11 triệu lệnh trong một giây (Million instructions per second), cho phép hỗ trợ các chức năng cơ bản như 80286 bao gồm quản lý bộ và bảo vệ bộ nhớ, dùng tốt cho các hệ điều hành đa nhiệm. Vì có mạch tính số học và logic ALU 32 bit nên nó có thể làm trực tiếp với từ dữ liệu dài 32 bit. Bộ nhớ của 80386 có thể lên tới 4 Gbyte và một chương trình có thể có tới 16.384 đoạn nên không gian địa chỉ ảo là 16.384×4 Gbyte = 6 Tbyte nhớ.

- $D_0 \div D_{31}$ là 32 bit dữ liệu. Tùy theo mức tín hiệu \overline{NA} trên bus điều khiển, thông tin truyền qua bus số liệu có thể 32 bit hoặc 16 bit.

- $A_0 \div A_{31}$ là 32 bit địa chỉ. Các tín hiệu $\overline{BE}_0 - \overline{BE}_3$ cho biết byte số liệu nào được truyền. Thông thường: \overline{BE}_0 chỉ $D_0 \div D_7$ được truyền, \overline{BE}_1 chỉ $D_8 \div D_{15}$, \overline{BE}_2 chỉ $D_{16} \div D_{23}$, \overline{BE}_3 chỉ

- $D_{24} \div D_{31}$ được truyền. Để truyền số liệu dài 32 bit thì cả 4 tín hiệu này phải ở mức tích cực thấp.

- CLK2 là lối vào xung nhịp cấp cho vi xử lý, chân này sẽ được chia đôi trong vi xử lý thành tín hiệu PCLK để đồng bộ quá trình thực hiện lệnh.

+ **Các tín hiệu ngắt: INTR và NMI.**

+ RESET: Đưa vi xử lý vào trạng thái khởi động lại. Chân này phải ở mức tích cực ít nhất 15 chu kỳ CLK2, nó nhạy với mức xung và phải đồng bộ với CLK2.

+ **Các tín hiệu xác định chu kỳ bus:** Đây là các tín hiệu ra 3 trạng thái, xác định các loại chu kỳ bus. $\overline{W}/\overline{R}$ phân biệt giữa chu kỳ nhớ và công vào ra. Bảng 3-1 chỉ ra các chu kỳ bus tương ứng với các tín hiệu trạng thái.

Bảng 3-1. Chu kỳ bus với các tín hiệu ngắt

$\overline{M}/\overline{IO}$	$\overline{D}/\overline{C}$	$\overline{W}/\overline{R}$	Chu kỳ bus	LOCK?
0	0	0	Trả lời ngắt	Có
0	0	1	Không xảy ra	-
0	1	0	Đọc vào ra	Không
0	1	1	Viết vào ra	Không
1	0	0	Đọc mã lệnh	Không

1	0	1	Dừng	Không
1	1	0	Đọc bộ nhớ	Có một số chu kỳ
1	1	1	Viết bộ nhớ	Có một số chu kỳ

+ Các tín hiệu điều khiển bus:

- \overline{ADS} là tín hiệu ra, ba trạng thái, cho biết chu kỳ bus bắt đầu và các tín hiệu $\overline{W/R}$, $\overline{C/R}$, $\overline{M/IO}$, $\overline{BE_0} - \overline{BE_3}$, $A_2 \div A_{31}$ được đưa ra các chân của vi xử lý.

- \overline{REDY} là tín hiệu vào, cho biết chu kỳ bus đã kết thúc.

- \overline{NA} (Next address) là tín hiệu vào dùng cho phương pháp định địa chỉ đường ống, mức tích cực thấp báo cho CPU biết là hệ thống đã sẵn sàng, nhận giá trị mới của tín hiệu $\overline{W/R}$, $\overline{C/R}$, $\overline{M/IO}$, $\overline{BE_0} - \overline{BE_3}$, $A_2 \div A_{31}$, mặc dù chu kỳ bus hiện hành chưa được trả lời bằng tín hiệu $\overline{REDY} = 0$.

- $\overline{BS16}$ là tín hiệu vào, cho phép vi xử lý nối với bus số liệu 32 hoặc 16 bit. Khi $\overline{BS16} = 0$ làm cho chu kỳ bus hiện hành chỉ sử dụng 16 bit thấp hơn bus số liệu $D_0 \div D_{15}$ (tương ứng $\overline{BE_0}$ và $\overline{BE_1}$). Nếu trong chu kỳ bus hiện tại tại $\overline{BE_0} = 0$ và $\overline{BE_1} = 0$ thì đưa tín hiệu $\overline{BS16} = 0$ không ảnh hưởng. Nhưng nếu $\overline{BE_2} = 0$ hoặc $\overline{BE_3} = 0$ mà đưa tín hiệu $\overline{BS16} = 0$ thì CPU sẽ tự động thích nghi để chuyển các byte này sang $D_0 \div D_{15}$. Tín hiệu $\overline{BS16}$ không được ở mức tích cực trong chu kỳ làm việc với bộ đồng xử lý.

+ Các tín hiệu trọng tài bus:

- \overline{HOLD} là tín hiệu vào do một thiết bị chủ khác đưa đến yêu cầu làm chủ bus. Nếu đồng thời có cả tín hiệu \overline{RESET} thì tín hiệu đó sẽ có độ ưu tiên lớn hơn.

HLDA: 80386 đưa ra tín hiệu này để trả lời tín hiệu \overline{HOLD} và chuyển vào trạng thái “trả lời \overline{HOLD} ”. Ở trạng thái này vi xử lý chỉ điều khiển tín hiệu HLDA còn tất cả các tín hiệu ra 2 chiều khác đều ở trạng thái trở kháng cao.

80386 cũng có 2 chế độ hoạt động, chế độ thực và chế độ bảo vệ. Ngoài ra còn có một chế độ hoạt động kiểu 8086 ảo cho phép dễ dàng chuyển qua lại giữa các nhiệm vụ trong chế độ bảo vệ và chế độ thực. Các thanh ghi trong 80386 vì dài 32 bit được ký hiệu thêm chữ E (Extended) ở đầu, ví dụ EAX, EBX. Một đặc điểm mới là 80386 có các thanh ghi gỡ rối. Một bộ gỡ rối mềm có thể nạp các địa chỉ điểm dừng trong chúng để hỗ trợ cho việc gỡ rối. 80386 có thể được ra lệnh “dừng” khi đơn vị địa chỉ trong bộ xử lý tính toán một địa chỉ tuyến tính trùng hợp với một trong các địa chỉ ở các thanh ghi gỡ rối. Dưới đây liệt kê các thanh ghi của 80386.

3.7.3.2. Tổ chức bộ nhớ của 80386

Bộ nhớ trong 80386 được tổ chức theo byte (8 bit), theo từ (16 bit) hay từ kép (32 bit). Ngoài chế độ thực, 80386 có những chế độ bảo vệ như sau:

Tổ chức *phân đoạn*, chia bộ nhớ thành từng đoạn riêng biệt dành cho mã lệnh, số liệu và ngăn xếp. Điều này cho phép nhiều chương trình có thể chạy cùng một lúc mà không ảnh hưởng lẫn nhau. Tổ chức theo đoạn rất tiện lợi cho việc lập trình theo môđun, là công cụ tốt cho những người lập trình ứng dụng.

Tổ chức *phân trang*, sử dụng bộ nhớ ảo gồm nhiều trang có kích thước nhỏ và cố định. Do đó chỉ một phần chương trình đang chạy được đặt vào bộ nhớ khi cần thiết. Điều này cho phép tách biệt được các nhiệm vụ đang chạy cùng một lúc. Tổ chức theo trang thì thuận lợi với người lập trình hệ thống khi cần quản lý bộ nhớ vật lý.

Quản lý bộ nhớ trong 80386 còn cho phép tạo một chế độ *8086 ảo*, là một nhiệm vụ đặc biệt chạy trong chế độ bảo vệ. Lúc này bộ vi xử lý mô phỏng hoàn toàn

vì xử lý 8086 và như vậy cho phép nhiều chương trình 8086 ảo có thể chạy song song đồng thời.

3.7.3.3. Chế độ bảo vệ phân đoạn của 80386

Chế độ bảo vệ phân đoạn trong 80386 cũng giống như trong 80286. Một trong những xuất phát điểm của việc phân đoạn bộ nhớ là do yêu cầu bảo vệ các nhiệm vụ khác nhau trong các chương trình. Giải pháp tổng quát cho bộ nhớ lúc này là định nghĩa nhiều không gian địa chỉ hoàn toàn độc lập gọi là các *đoạn nhớ* cho mỗi nhiệm vụ. Mỗi đoạn gồm các địa chỉ tuyến tính kế tiếp nhau từ 0 đến một giá trị tối đa. Như vậy mỗi đoạn có một độ dài xác định, chiều dài này có thể thay đổi trong thời gian thực thi chương trình.

Giống như trong 80286, mỗi đoạn nhớ cũng được gán cho một trong 4 mức đặc quyền đánh số từ 0 đến 3 với mức đặc quyền giảm dần như chỉ ra trên hình 3-30. Mức đặc quyền gán cho một chương trình cho biết chương trình có thẩm quyền làm nhiệm vụ gì khi nó được thực hiện bởi một nhiệm vụ. Mỗi đoạn gồm các địa chỉ tuyến tính kế tiếp nhau từ 0 đến một giá trị tối đa. Như vậy mỗi đoạn có một độ dài xác định, chiều dài này có thể thay đổi trong thời gian thực thi chương trình.

3.7.3.4. Chế độ bảo vệ phân trang của 80386

Trong phương pháp này, không gian địa chỉ ảo được tách ra thành nhiều trang có kích thước bằng nhau. Thường kích thước trang bằng lũy thừa của 2, ví dụ bằng $2^{12} = 4$ Kbyte hay $2^{22} = 4$ Mbyte. Không gian địa chỉ vật lý được tách thành nhiều mảng theo cách tương tự, mỗi mảng có kích thước bằng kích thước trang sao cho mỗi mảng của bộ nhớ chính có khả năng lưu giữ đúng một trang. Khi một nhiệm vụ yêu cầu truy cập một địa chỉ lôgic, bộ xử lý chuyển địa chỉ lôgic này thành địa chỉ tuyến tính và trong chế độ phân đoạn địa chỉ này cũng chính là địa chỉ vật lý. Nếu trang chứa địa chỉ tuyến tính không tồn tại trong bộ nhớ vật lý thì một ngoại lệ sẽ xảy ra. Nếu chế độ trang được phép, bộ xử lý sẽ chuyển địa chỉ tuyến tính sang địa chỉ vật lý tương ứng với các trang. Thông tin mà bộ xử lý dùng để ánh xạ địa chỉ tuyến tính vào không gian địa chỉ vật lý được lưu trữ vào các vùng nhớ gọi là *thư mục trang và bảng các trang*.

Thư mục trang (Page directory) có $2^{10} = 1024$ điểm vào 32 bit (4 byte). Mỗi điểm vào trỏ tới một trong 1024 bảng trang. Như vậy, thư mục trang dài 4 Kbyte. Có một thanh ghi điều khiển đặc biệt 32 bit CR3 chứa địa chỉ cơ sở của thư mục trang. Chỉ có 20 bit cao của địa chỉ cơ sở được ghi trong CR3 còn 12 bit thấp được mặc định bằng không vì địa chỉ thư mục trang phải được tuân theo đơn vị nhỏ nhất của trang là 4 Kbyte. Nội dung của mỗi điểm vào thư mục trang như sau:

31	12	11	10	9	8	7	6	5	4	3	2	1	0
20 bit địa chỉ cơ sở bảng trang				OS	G	PS	0	A	PCD	PWT	U/S	R/W	P

+ Bảng các trang (Page table) cũng có $2^{10} = 1024$ điểm vào 32 bit (4 byte). Mỗi điểm vào trỏ tới một trong 1024 bảng trang (còn gọi là khung trang). Như vậy bảng các trang cũng dài 4 kbyte. Nội dung của mỗi điểm vào bảng trang như sau:

31	12	11	10	9	8	7	6	5	4	3	2	1	0
20 bit địa chỉ cơ sở trang				OS	G	0	D	A	PCD	PWT	U/S	R/W	P

20 bit cao của các điểm vào xác định vị trí byte đầu tiên của trang trong bộ nhớ vật lý và tự động định vị các trang theo khoảng cách 4 Kbyte. Nếu dùng trang dài 4 Mbyte thì chỉ có các bit từ $22 \div 31$ có giá trị, còn các bit từ $12 \div 21$ bằng 0.

Chương 4

BỘ NHỚ MÁY TÍNH

4.1. Khái quát về hệ thống nhớ của MTĐT

Bộ nhớ máy tính gồm nhiều thành phần với tốc độ truy cập và dung lượng khác nhau được kết hợp với nhau tạo thành hệ thống nhớ. Có nhiều cách phân loại bộ nhớ máy tính, có thể chia bộ nhớ máy tính dựa trên ba tiêu chí:

1. Kiểu truy cập.
2. Khả năng duy trì dữ liệu.
3. Công nghệ chế tạo.

Dựa trên kiểu truy cập, có thể chia bộ nhớ máy tính thành ba loại: Bộ nhớ truy cập tuần tự (Serial Access Memory - SAM), bộ nhớ truy cập ngẫu nhiên (Random Access Memory - RAM), và bộ nhớ chỉ đọc (Read Only Memory - ROM). Trong bộ

nhớ truy cập tuần tự, các ô nhớ được truy cập một cách tuần tự, có nghĩa là muốn truy cập đến ô nhớ sau phải duyệt qua ô nhớ trước nó. Tốc độ truy cập các ô nhớ có vị trí khác nhau là không giống nhau. Ngược lại, trong bộ nhớ truy nhập ngẫu nhiên, các ô nhớ có thể được truy cập ngẫu nhiên, không theo một trật tự định trước. Với bộ nhớ chỉ đọc, thông tin được ghi vào bộ nhớ một lần nhờ một thiết bị đặc biệt và sau đó chỉ có thể đọc ra.

4.1.1. Tổ chức mạch nhớ

Một mạch nhớ (Memory chip) thường gồm nhiều ô nhớ (Memory cells) được tổ chức thành một ma trận nhớ gồm một số hàng và một số cột.

Các đường địa chỉ là một tập các chân tín hiệu kết nối với bus địa chỉ nhận các tín hiệu địa chỉ ô nhớ từ CPU. Bộ giải mã địa chỉ giải mã các tín hiệu địa chỉ ô nhớ thành các địa chỉ hàng và cột để có thể chọn ra được ô nhớ. Các đường dữ liệu là một tập các chân tín hiệu kết nối với bus dữ liệu để nhận tín hiệu dữ liệu từ CPU và gửi tín hiệu dữ liệu đọc được từ ô nhớ về CPU.

Các tín hiệu điều khiển có nhiệm vụ điều khiển hoạt động của mạch nhớ theo các tín hiệu lệnh gửi đến từ CPU. Tín hiệu chọn mạch CS cho phép kích hoạt mạch nhớ làm việc với CPU khi CS = 0. Thông thường, tại mỗi thời điểm chỉ có một mạch nhớ được chọn kích hoạt làm việc với CPU, còn các mạch khác ở trạng thái không được kích hoạt. Tín hiệu cho phép ghi WE = 0 sẽ cho phép ghi thông tin vào các ô nhớ trong một dòng. Tương tự, tín hiệu cho phép đọc RD = 0 sẽ cho phép đọc dữ liệu từ các ô nhớ trong một dòng.

4.1.2. Các đặc trưng cơ bản của hệ thống nhớ

4.1.2.1. Vị trí

Bộ nhớ máy tính bao gồm cả hai loại bộ nhớ trong và ngoài. Bộ nhớ trong của máy tính thường được đề cập đến như bộ nhớ chính. Bộ nhớ ngoài của máy tính gồm các thiết bị lưu trữ ngoại vi, như đĩa và băng từ, vốn có thể truy cập được đối với CPU thông qua các bộ điều khiển nhập/xuất.

4.1.2.2. Dung lượng

Với bộ nhớ trong, dung lượng thường được biểu diễn dưới dạng byte. (1 byte = 8 bit) hay word. Các độ dài word phổ biến là 8, 16, và 32 bit. Bộ nhớ ngoài có dung lượng được biểu thị theo byte.

4.1.2.3. Đơn vị truyền

Với bộ nhớ trong, đơn vị truyền bằng với số đường dữ liệu vào/ra khỏi module bộ nhớ. Giá trị này thường bằng với độ dài của một word, nhưng cũng có thể không. Để hiểu rõ hơn khái niệm này, chúng ta hãy xem xét ba khái niệm có liên quan đến bộ nhớ trong:

a. Word:

Đơn vị tự nhiên của tổ chức máy tính. Kích thước của một word thường bằng với số bit được sử dụng để biểu diễn một số hay độ dài của chỉ thị. Tuy nhiên có rất nhiều ngoại lệ. Ví dụ, máy CRAY-1 có độ dài word 64 bit trong khi có biểu diễn số nguyên 24 bit. Máy VAX có nhiều loại độ dài chỉ thị dưới dạng bội số của byte, và có kích thước word 32 bit.

b. Các đơn vị khả định địa chỉ:

Trong nhiều hệ thống, đơn vị khả định địa chỉ là word. Mặc dù vậy, có một số hệ thống cho phép định địa chỉ ở mức byte. Trong mọi trường hợp, mối quan hệ giữa độ dài A của một địa chỉ và số N các đơn vị khả định địa chỉ là $2^A = N$.

c. Đơn vị truyền:

Đối với bộ nhớ chính, đây là số bit đọc/ghi vào bộ nhớ tại một thời điểm. Đơn vị truyền không nhất thiết bằng một word hay một đơn vị khả định địa chỉ. Với bộ nhớ ngoài, dữ liệu thường được truyền theo những đơn vị lớn hơn nhiều so với word và được gọi là khối.

4.1.2.4. Phương thức truy nhập

Đây là một trong những yếu tố rõ nhất giúp phân biệt các kiểu bộ nhớ. Có bốn loại phương thức truy cập:

a. Truy nhập tuần tự

Bộ nhớ được tổ chức thành các đơn vị dữ liệu gọi là bản ghi. Việc truy cập phải được thực hiện theo một dãy tuyến tính cụ thể. Thông tin địa chỉ được lưu trữ được dùng để phân tách các bản ghi và hỗ trợ quá trình tìm kiếm lấy thông tin. Một bộ phận đọc/ghi dùng chung được sử dụng. Bộ phận này phải được di chuyển từ vị trí hiện thời của nó đến vị trí được yêu cầu, quét qua và từ chối các bản ghi trung gian. Do đó, thời gian để truy cập một bản ghi tùy ý biến đổi khá cao.

b. Truy nhập trực tiếp

Cũng như với truy cập tuần tự, truy cập trực tiếp bao gồm việc dùng chung một bộ phận đọc/ghi. Tuy nhiên, các khối hay bản ghi riêng lẻ có một địa chỉ duy nhất dựa trên vị trí vật lý. Việc truy cập được thực hiện thông qua truy cập trực tiếp cộng với tìm kiếm tuần tự, đếm, hay chờ để đến được vị trí cuối cùng. Một lần nữa, thời gian truy cập là biến đổi.

c. Truy nhập ngẫu nhiên

Mỗi vị trí khả định địa chỉ trong bộ nhớ có mô tả cơ chế định địa chỉ vật lý duy nhất. Thời gian truy cập một vị trí cho trước độc lập với dãy các truy cập trước đó và không thay đổi. Do đó, bất kỳ một vị trí nào cũng có thể được chọn ngẫu nhiên và được định địa chỉ cũng như truy cập trực tiếp. Các hệ thống bộ nhớ chính được truy cập ngẫu nhiên.

d. Truy nhập liên kết

Đây là kiểu truy cập ngẫu nhiên bộ nhớ cho phép thực hiện việc so sánh các vị trí bit có yêu cầu trong một word phục vụ cho việc đối sánh đặc biệt nào đó, và có thể thực hiện thao tác này cùng một lúc cho tất cả các word. Do đó một word được trích ra dựa trên một phần nội dung của nó chứ không phải dựa trên địa chỉ. Tương tự như với phương thức truy cập ngẫu nhiên thông thường, mỗi vị trí nhớ có cơ chế định địa chỉ riêng, và thời gian lấy thông tin không đổi, độc lập với vị trí hoặc khuôn dạng truy cập trước đó.

4.1.2.5. Hiệu suất

Đứng trên quan điểm người sử dụng, hai đặc trưng quan trọng nhất của bộ nhớ là dung lượng và hiệu suất vận hành. Có ba tham số hiệu suất được sử dụng:

a. Thời gian truy nhập:

Đối với bộ nhớ truy cập ngẫu nhiên, đây là thời gian cần thiết để thực hiện một thao tác đọc hay ghi, tức là thời gian từ lúc một địa chỉ có mặt trong bộ nhớ cho đến lúc dữ liệu được lưu trữ xong hoặc đã sẵn sàng để sử dụng. Với bộ nhớ truy cập không ngẫu nhiên, thời gian truy cập là thời gian cần để định vị bộ phận đọc/ghi tại vị trí được yêu cầu.

b. Thời gian chu kỳ bộ nhớ:

Khái niệm này chủ yếu được áp dụng cho bộ nhớ truy cập ngẫu nhiên và bao gồm thời gian truy cập cộng với bất kỳ thời gian phụ thêm nào được yêu cầu trước khi truy cập thứ hai có thể được thực hiện. Phần thời gian phụ thêm này có thể được yêu cầu nhằm phát sinh dữ liệu nếu chúng được đọc một cách không loại trừ.

c. Tốc độ truyền:

Đây là tốc độ truyền dữ liệu vào/ra một đơn vị bộ nhớ. Với bộ nhớ truy cập ngẫu nhiên, giá trị này bằng $1/(\text{Thời gian Chu kỳ})$. Với bộ nhớ truy cập không ngẫu nhiên, quan hệ sau đây được duy trì:

$$TN = TA + N/R$$

Trong đó:

TN = thời gian trung bình để đọc/ghi **N** bit

TA = thời gian truy cập trung bình

N = số các bit

R = tốc độ truyền, theo đơn vị bit/giây (bps)

4.1.2.6. Kiểu vật lý

Hiện nay có hai kiểu vật lý phổ biến nhất là bộ nhớ bán dẫn, sử dụng công nghệ LSI hay VLSI, và bộ nhớ với bề mặt từ hóa, được dùng cho đĩa và băng từ.

4.1.2.7. Đặc tính vật lý

Nhiều đặc tính vật lý của sự lưu trữ dữ liệu là rất quan trọng. Trong một bộ nhớ khả biến, thông tin phân rã một cách tự nhiên hoặc bị mất đi khi nguồn điện bị tắt. Trong một bộ nhớ bất biến, thông tin một khi đã được ghi sẽ được lưu giữ mà không bị thoái hóa. Các bộ nhớ có bề mặt từ hóa thuộc loại bất biến. Bộ nhớ bán dẫn có thể khả biến hoặc bất biến. Bộ nhớ không thể xóa thì không thể thay đổi được, ngoại trừ việc phá hủy đơn vị lưu trữ. Bộ nhớ bán dẫn kiểu này được biết đến với tên gọi bộ nhớ chỉ đọc (ROM).

4.1.2.8. Cách tổ chức

Ở đây là cách sắp xếp vật lý các bit để tạo thành các word. Cách sắp xếp hiển nhiên không phải lúc nào cũng được sử dụng.

4.2. Tổ chức bộ nhớ theo phân cấp

4.2.1. Giới thiệu cấu trúc phân cấp hệ thống nhớ

Hầu hết hệ thống nhớ trong các thiết bị tính toán hiện đại được tổ chức theo cấu trúc phân cấp (Hierarchical structure). Cấu trúc phân cấp không chỉ được sử dụng trong các hệ thống nhớ mà nó còn sử dụng rộng rãi trong đời sống xã hội, như cấu trúc tổ chức các cơ quan nhà nước, doanh nghiệp và cả các trường học.

Để tăng tốc độ xử lý của CPU, đồng thời đảm bảo khả năng lưu trữ dữ liệu với khối lượng lớn, người ta tổ chức bộ nhớ máy tính theo kiểu hệ thống có phân cấp.

Ý tưởng chính trong việc sử dụng hệ thống bộ nhớ có phân cấp là tại một thời điểm thì các lệnh và dữ liệu được sử dụng đều nằm ở một vùng tương đối nhỏ của bộ nhớ chính. Các lệnh và dữ liệu trong vùng này luôn chuyển dịch khi thực hiện chương trình. Cơ sở của việc quản lý hệ thống nhớ có phân cấp, trong đó bao gồm nhiều loại thiết bị nhớ khác nhau là dựa trên nguyên lý quy chiếu phân vùng.

Kỹ thuật được sử dụng để giảm thời gian trung bình truy nhập bộ nhớ chính, là thêm một bộ nhớ có tốc độ truy nhập cao (bộ nhớ SRAM) vào hệ thống lưu trữ này. Bộ nhớ loại này được gọi là bộ nhớ Cache (bộ nhớ ẩn). Bộ nhớ cache được sử dụng để lưu trữ các lệnh và dữ liệu thường được sử dụng nhiều trong quá trình thực hiện chương trình. Việc quy chiếu truy nhập đến bộ nhớ chính chỉ xảy ra khi không tìm thấy lệnh hoặc dữ liệu cần có trong cache.

4.2.2. Vai trò của cấu trúc phân cấp hệ thống nhớ

Không hoàn toàn giống với vai trò của cấu trúc phân cấp trong các cơ quan và doanh nghiệp là “chia để trị”, cấu trúc phân cấp trong hệ thống nhớ có hai vai trò chính:

1. Tăng hiệu năng hệ thống thông qua việc giảm thời gian truy cập các ô nhớ.

2. Giảm giá thành sản xuất.

Sở dĩ cấu trúc phân cấp trong hệ thống nhớ có thể giúp tăng hiệu năng hệ thống là do nó giúp dung hoà được CPU có tốc độ cao và phần bộ nhớ chính và bộ nhớ ngoài có tốc độ thấp. CPU sẽ chủ yếu trực tiếp truy cập bộ nhớ cache có tốc độ cao, và cache sẽ có nhiệm vụ chuyển trước các dữ liệu cần thiết về từ bộ nhớ chính. Nhờ vậy, CPU sẽ không phải thường xuyên truy cập trực tiếp bộ nhớ chính và bộ nhớ ngoài để tìm dữ liệu – các thao tác tốn nhiều thời gian do các bộ nhớ này có tốc độ chậm. Như vậy, có thể nói rằng, thời gian trung bình CPU truy nhập dữ liệu từ hệ thống nhớ tiệm cận thời gian truy nhập bộ nhớ cache.

4.3. Bộ nhớ bán dẫn

4.3.1. Đặc điểm

Tính ưu việt chủ yếu của các hệ thống số so với hệ thống tương tự là khả năng lưu trữ một lượng lớn thông tin số và dữ liệu trong những khoảng thời gian nhất định. Khả năng nhớ này là điều làm cho hệ thống số trở thành đa năng và có thể thích hợp với nhiều tình huống. Ví dụ, trong một máy tính số, bộ nhớ trong chứa những lệnh mà theo đó máy tính có thể hoàn tất công việc của mình với sự tham gia ít nhất của con người.

Bộ nhớ bán dẫn được sử dụng làm bộ nhớ chính trong các máy tính nhờ vào khả năng thỏa mãn tốc độ truy xuất dữ liệu của bộ xử lý trung tâm (CPU).

Chúng ta đã quá quen thuộc với Flip-Flop (FF), một linh kiện điện tử có tính nhớ. Chúng ta cũng đã thấy một nhóm các FF hợp thành thanh ghi để lưu trữ và dịch chuyển thông tin như thế nào. Các FF chính là các phần tử nhớ tốc độ cao được dùng rất nhiều trong việc điều hành bên trong máy tính, nơi mà dữ liệu dịch chuyển liên tục từ nơi này đến nơi khác.

4.3.1.1. Dung lượng bộ nhớ

Số lượng bit mà chip nhớ bán dẫn có thể dùng để lưu dữ liệu được gọi là dung lượng chip nhớ. Đơn vị dung lượng chip nhớ thường dùng là Kbit (Kilobit), Mbit (Megabit), cần phải phân biệt khái niệm này với dung lượng bộ nhớ của máy tính. Dung lượng chip nhớ luôn được tính bằng bit còn dung lượng bộ nhớ của máy tính được tính bằng byte

$DLN = \text{Không gian địa chỉ nhớ} \times \text{kích thước của 1 đơn vị nhớ.}$

4.3.1.2. Tổ chức của bộ nhớ

Bộ nhớ được tổ chức theo các ô nhớ trong vi mạch nhớ, mỗi ô nhớ có thể chứa 1 bit, 4 bit, 8 bit hoặc 16 bit, tùy thuộc vào thiết kế. Số lượng bit nhớ trong mỗi ô nhớ của chip nhớ luôn bằng số chân dữ liệu của chip nhớ đó, còn số ô nhớ lại phụ thuộc vào số chân địa chỉ và bằng 2 lũy thừa của số chân địa chỉ. Do đó tổng các bit mà IC nhớ có thể lưu giữ sẽ bằng số các ô nhớ nhân với số bit trong mỗi ô

Như vậy:

- Mỗi chip nhớ có 2^x ô nhớ, trong đó x là số chân địa chỉ của chip.
- Mỗi ô nhớ có m bit, trong đó m là số chân dữ liệu trên chip.
- Toàn bộ chip sẽ chứa $2^x \times m$ bit, trong đó x là số chân địa chỉ và m là số chân dữ liệu trên chip.

4.3.1.3. Tốc độ

Một trong những thông số quan trọng của vi mạch nhớ là tốc độ truy nhập dữ liệu. Để truy nhập dữ liệu, trước tiên địa chỉ cần được chuyển tới các chân địa chỉ và sau một khoảng thời gian dữ liệu mới được áp đến các chân dữ liệu. Khoảng thời gian này càng ngắn càng tốt tất nhiên chip nhớ sẽ càng đắt. Tốc độ của chip nhớ thường gọi

là thời gian truy nhập, thường từ vài ns đến hàng trăm ns phụ thuộc vào công nghệ chế tạo nó.

Như vậy, Bộ nhớ bán dẫn thường được sử dụng làm bộ nhớ trong của máy tính. Bộ nhớ trong được nối với các bộ phận khác trong máy tính thông qua Bus dữ liệu và Bus địa chỉ. Điều khiển các mạch nhớ có đường điều khiển cho phép mở (Device enable). Ngoài ra còn có thêm đường điều khiển cho phép đọc/ghi (read/write/enable).

Thông thường các ngăn nhớ trong bộ nhớ được bố trí ở dạng ma trận. Để xác định vị trí nhớ hay phần tử nhớ cần có mạch giải mã địa chỉ. Mạch này gồm 2 phần:

- Mạch chọn địa chỉ hàng RAS (Row Address Selector).
- Mạch chọn địa chỉ cột CAS (Column Address Selector).

Đường dây Device Enable cho phép mở các mạch điện lối ra của bộ nhớ theo 3 trạng thái còn đường dây Read/write enable sẽ xác định dạng thao tác là đọc hay là ghi vào bộ nhớ.

4.3.2. Phân loại bộ nhớ bán dẫn

4.3.2.1. Bộ nhớ chỉ đọc ROM (Read Only Memory)

- Bộ nhớ ROM dùng để lưu chương trình do người viết chương trình viết ra. Chương trình là tập hợp các câu lệnh thể hiện các thuật toán để giải quyết các công việc cụ thể, sau đó được đưa vào lưu trong ROM của vi điều khiển, khi hoạt động, vi điều khiển truy xuất từng câu lệnh trong ROM để thực hiện chương trình. ROM còn dùng để chứa số liệu các bảng, các tham số hệ thống, các số liệu cố định của hệ thống. Trong quá trình hoạt động nội dung ROM là cố định, không thể thay đổi, nội dung ROM chỉ thay đổi khi ROM ở chế độ xóa hoặc nạp chương trình (do các mạch điện riêng biệt thực hiện).

+ PROM (Programmable ROM)

Là bộ nhớ mà người sử dụng có thể lập trình được. Mỗi bit của PROM có 1 cầu chì. PROM được lập trình bằng cách đốt cháy các cầu chì. Nếu thông tin đốt cầu chì của PROM bị sai thì PROM đó không dùng được nữa vì khi cầu chì bị đứt thì không thể khôi phục lại được. Do đó PROM còn được gọi là bộ nhớ lập trình 1 lần (One Time Programmable - OTP). Quá trình nạp PROM còn được gọi là đốt hay nạp và cần có 1 thiết bị chuyên dùng gọi là bộ nạp.

+ EPROM (Erasable Programmable ROM)

Loại này có thể thay đổi (xóa đi và nạp lại) nội dung nhớ nhiều lần. Thời gian xóa EPROM chỉ cần khoảng $15 \div 20$ phút.

- Như vậy để lập trình cho EPROM, ta cần phải có thiết bị xóa và thiết bị nạp riêng biệt. Nhược điểm chủ yếu của EPROM là không thể lập trình trực tiếp khi chúng đang nằm trên bảng mạch chủ. Để khắc phục nhược điểm này, người ta chế tạo bộ nhớ khác gọi là EEPROM

+ ROM mặt nạ (Mask Programmed ROM, MROM)

Khi chế tạo nó được ghi luôn thông tin và nội dung thông tin đó vĩnh viễn không thay đổi được. Khi xuất xưởng toàn bộ thông tin trong ROM được nạp sẵn các thông tin hay dữ liệu lên. Khi đã được chương trình hóa như vậy thì các bit dữ liệu không thể thay đổi được nữa.

Đây là loại ROM được chế tạo để thực hiện một công việc cụ thể như các bảng tính, bảng lượng giác, bảng logarit ngay sau khi xuất xưởng. Nói cách khác, các tế bào nhớ trong ma trận nhớ đã được tạo ra theo một chương trình đã xác định trước bằng phương pháp mặt nạ: đưa vào các linh kiện điện tử nối từ đường từ qua đường bit để tạo ra một giá trị bit và để trống cho giá trị bit ngược lại.

+ EEPROM (Electrically Erasable Programmable ROM)

- So với EPROM thì EEPROM có một số ưu điểm như nhờ sử dụng phương pháp xóa điện nên việc xóa được thực hiện gần như tức thì, ngoài ra nó có thể chọn riêng từng byte để xóa chứ không phải xóa toàn bộ nội dung như ở EPROM.

- Ưu điểm chính của EEPROM là có thể lập trình và xóa nội dung ngay trên bản mạch chủ mà không cần phải gỡ ra khỏi vi mạch.

+ **Bộ nhớ Flash EPROM**

Từ đầu những năm 90 flash EPROM đã trở thành vi mạch nhớ lập trình được sử dụng khá phổ biến với một số ưu điểm. Trước hết là thời gian xóa toàn bộ nội dung chưa đến 1 giây, do đó nó còn được gọi là bộ nhớ cực nhanh, ngoài ra nó còn được xóa bằng điện nên có thể gọi là flash EEPROM. Điểm khác biệt lớn nhất giữa EEPROM và Flash là khi xóa bộ nhớ Flash thì toàn bộ nội dung sẽ bị xóa, còn đối với EEPROM thì có thể xóa từng byte hoặc từng phần.

4.3.2.2. **Bộ nhớ truy cập ngẫu nhiên RAM (Random Access Memory)**

RAM là thế hệ kế tiếp của ROM, cả RAM và ROM đều là bộ nhớ truy xuất ngẫu nhiên, tức là dữ liệu được truy xuất không cần theo thứ tự. Tuy nhiên ROM chạy chậm hơn RAM rất nhiều. Thông thường ROM cần trên 50ns để xử lý dữ liệu trong khi đó RAM cần dưới 10ns.

Bộ nhớ RAM được chế tạo theo công nghệ bán dẫn và thuộc loại bộ nhớ không ổn định, nó có nhược điểm là thuộc loại “bay hơi” (Volatite), nghĩa là mọi thông tin trong RAM chỉ tồn tại khi có nguồn điện nuôi và mất khi không còn nguồn điện nuôi.

a. Các đặc trưng cơ bản của RAM

+ **Dung lượng bộ nhớ:** Tổng số byte của bộ nhớ (nếu tính theo byte) hoặc là tổng số bit trong bộ nhớ nếu tính theo bit.

+ **Tổ chức bộ nhớ:** Số ô nhớ và số bit cho mỗi ô nhớ.

+ **Thời gian thâm nhập:** Thời gian từ lúc đưa ra địa chỉ của ô nhớ đến lúc đọc được nội dung của ô nhớ đó.

+ **Chu kỳ bộ nhớ:** Thời gian giữa hai lần liên tiếp thâm nhập bộ nhớ.

b. Phân loại RAM

Dựa vào công nghệ ta chia RAM thành hai loại:

+ **RAM tĩnh (Static RAM)**

Đối với SRAM, nội dung ô nhớ vẫn giữ nguyên khi chưa mất nguồn cung cấp mà không cần phải tốn thời gian làm tươi ô nhớ. Do điện áp chênh lệch lớn nên thời gian xử lý khuếch đại sẽ nhỏ hơn trong SRAM (thời gian truy xuất của SRAM là khoảng 1ns trong khi của DRAM khoảng 40ns). Từ đó, SRAM không thực hiện phân kênh địa chỉ hàng và cột (điều này sẽ làm giảm thời gian truy xuất). Sau khi dữ liệu ổn định, bộ giải mã cột chọn cột phù hợp và đưa dữ liệu đến bộ đệm đầu ra.

Bộ nhớ được xây dựng từ các phần tử nhớ 1 bit SRAM. Mỗi bit SRAM được xây dựng trên các mạch điện tử flip – flop.

- Các khác biệt của SRAM so với DRAM:

- Tốc độ của SRAM lớn hơn DRAM do không phải tốn thời gian refresh..

- Chế tạo SRAM tốn kém hơn DRAM nên thông thường sử dụng DRAM để hạ giá thành sản phẩm.

Trong máy tính nó được dùng làm cache L1(sơ cấp), L2 (thứ cấp).

Phần tử nhớ 1 bit của bộ nhớ SRAM là phần tử nhớ bán dẫn có tính chất là khi ta thiết lập cho phần tử nhớ 1 bit một giá trị 0 hoặc 1 thì nó nhớ mãi giá trị đó cho đến khi ta thiết lập cho nó một giá trị mới.

Các đường dây truyền tín hiệu trên sơ đồ mạch có ý nghĩa như sau:

D_{in}: tín hiệu thông tin (dữ liệu) đầu vào.

D_{out} : tín hiệu thông tin (dữ liệu) đầu ra.

X_i, Y_j : Là các dây địa chỉ, nếu ta tổ chức một mảng các phần tử nhớ hai chiều thì phần tử S_{ij} nằm ở hàng i cột j và X_i sẽ được nối với hàng i của ma trận, còn Y_j sẽ được nối với cột j của ma trận.

CS (Chip Select): tín hiệu chọn chip (tín hiệu chọn vỏ), đôi khi được ký hiệu là CE (Chip Enable). Khi có nhiều chip nhớ RAM cùng nối với đường tín hiệu chung (D_{in}, D_{out}) thì đầu vào CS có nhiệm vụ chọn xem chip RAM nào được truyền thông với Bus dữ liệu.

Với các phần tử nhớ 1 bit có cấu tạo như trên, ta rất dễ kết hợp chúng với nhau để tạo nên các bộ nhớ có ngăn nhớ với kích thước mong muốn (8 bit, 1 word).

+ Điều kiện ghi:

$CS = 1, X_i = 1, WE = 1$ khi đó Q sẽ được thiết lập bằng D_{in} đồng thời H và D_{out} ở trạng thái trở kháng cao.

+ Điều kiện đọc:

$CS = 1, X_i = 1, Y_j = 1, WE = 0$, khi đó D_{out} sẽ nhận giá trị bằng Q và Q không phụ thuộc vào D_{in} .

- WE (Write Enable): tín hiệu cho phép ghi.

+ Khi $WE = 1$ cho phép ghi thông tin D_{in} vào phần tử nhớ. Khi đó:

- $WE = 1, \overline{WE} = 0, CS = 1 \rightarrow C = 0$, lúc này, đầu ra D_{out} ở trạng thái trở kháng cao, có thể nói đường dây ra D_{out} ngắt mạch với phần tử nhớ

- $X_i = 1, Y_j = 1 \rightarrow A = 1$

- $WE = 1, CS = 1 \rightarrow B = 0 \rightarrow E = 1$

+ Nếu $D_{in} = 0 \rightarrow G = 1$

- $G = 1, E = 1 \rightarrow F = 0 \rightarrow \overline{Q} = 1$

- $\overline{G} = 1, Q = 1 \rightarrow Q = 0$ hay $Q = D_{in}$

+ Nếu $D_{in} = 1$ với $E = 1 \rightarrow G = 0$

- $G = 0, E = 1 \rightarrow F = 1$

- $G = 0 \rightarrow Q = 1$ hay $Q = D_{in}$

- $G = 1, F = 1 \rightarrow \overline{Q} = 0$

Vậy ta thấy trong trường hợp này Q luôn $= D_{in}$.

+ Khi $WE = 0$ cho phép đọc thông tin từ phần tử nhớ, khi đó:

- $WE = 0, \overline{WE} = 1, CS = 1 \rightarrow C = 1$, lúc này đầu ra D_{out} luôn bằng H

- $X_i = 1, Y_j = 1 \rightarrow A = 1$

- $WE = 0, CS = 1 \rightarrow B = 1 \rightarrow E = 0 \rightarrow G = 1, F = 1$

+ Giả sử $Q = 1 \rightarrow \overline{Q} = F \text{ NAND } Q = 1 \text{ NAND } 1 = Q \rightarrow Q = 1$ (1)

+ Giả sử $Q = 0 \rightarrow \overline{Q} = 1 \rightarrow Q = \overline{Q} \text{ NAND } G = 1 \text{ NAND } 1 = Q$ (2)

Từ (1) và (2) suy ra Q không đổi.

Vậy trong trường hợp này Q không đổi trạng thái (giá trị tín hiệu đã ghi được bảo toàn), tức là không bị phụ thuộc vào D_{in} .

+ RAM động (Dynamic RAM)

Được xây dựng trên cơ sở nhớ là các tụ điện, khi giữa hai cực của tụ điện có điện áp (tích điện) thì bit thông tin là 1, theo thời gian mức điện áp này bị giảm dần do có sự dò điện tích trên tụ. Do vậy DRAM là loại phải “làm tươi” (Refresh) theo chu kỳ, tức là phải nạp lại các dữ liệu đang được lưu giữ theo từng chu kỳ. Làm tươi có thể thực hiện bằng cách nhắc lại thao tác đọc hoặc ghi hoặc cũng có thể bằng những thao tác đặc biệt khác. Mật độ phần tử nhớ rất cao nên giá thành của DRAM được tính theo

dung lượng nên nó rẻ hơn so với SRAM. Nhược điểm của DRAM là tốc độ hoạt động chậm hơn so với SRAM. DRAM là loại thông dụng nhất trong các hệ máy vi tính hiện nay. DRAM có một số đặc điểm sau:

- Thông tin không ổn định, do đó cần phải có mạch làm tươi để ổn định (mạch bù điện áp bị sụt).

- Tốc độ chậm.

- Dung lượng IC lớn.

- Giá thành thấp.

DRAM được dùng để thiết kế ra bộ nhớ chính và có 2 loại phổ biến là:

- + EDO – DRAM (Extended Data Output DRAM) được cải tiến ở mạch ra dữ liệu để truy nhập nhanh.

- + SDRAM (Synchronous DRAM) loại này có chân để đưa tín hiệu xung clock vào nhằm tạo ra sự đồng bộ theo đầu vào của tín hiệu clock, do đó phân mức thực hiện đồng thời kiểu Pipeline.

- Địa chỉ xác định ô nhớ của RAM được chia thành 2 phần: địa chỉ hàng và cột.

- Hai địa chỉ này được đưa lần lượt vào bộ đệm. Quá trình dò kênh địa chỉ điều khiển bằng các tín hiệu RAS (Row Access Strobe) và CAS (Column Access Strobe). Bộ điều khiển nhớ của CPU phải thực hiện 3 công việc sau:

- + Chia địa chỉ từ CPU thành các địa chỉ hàng và cột.

- + Tích cực các chân RAS, CAS và WE một cách chính xác.

- + Truyền và nhận các dữ liệu đọc, ghi.

- + RAS (Row Address Select).

- + CAS (Column Address Select).

- + Nếu RAS = 0 thì các đường địa chỉ từ $A_0 \div A_{n-1}$ được đưa vào bộ giải mã hàng.

- + Nếu CAS = 0 thì đưa vào bộ giải mã cột.

Chúng ta đã biết, các phần tử SRAM được cấu tạo từ các mạch flip-flop. Một phần tử SRAM 1 bit được cấu tạo bởi 6 cổng NAND và một bộ đệm 3 trạng thái. Nếu mỗi cổng hoặc bộ đệm được tạo bởi 3 transistor thì cần đến 19 transistor để nhớ được 1 bit thông tin. Ngày nay người ta sử dụng các phần tử SRAM đơn giản hơn chứa từ 1 đến 6 transistor.

* **Phần tử DRAM 1 transistor**

Bộ nhớ DRAM được tổ chức thành một ma trận nhớ (thường là ma trận vuông), trong đó dây từ (Word line) là một trong các dây của ma trận hàng; còn dây bit (bit line) là một trong các dây của ma trận cột. Transistor T là 1 transistor trường (Field Effect Transistor) đóng vai trò một chuyển mạch điện tử. T có 3 cực là cổng G (Gate), cực máng D (Drain) và cực nguồn S (Source), trong đó G là cực điều khiển. D được nối với S khi G có mức điện áp cao (bằng 1) so với S, ngược lại thì điện trở giữa D và S sẽ rất lớn.

+ **Việc ghi (write)**

Khi dây từ có mức tích cực (= 1), T ở trạng thái mở, nối tụ C với dây bit. Nếu thao tác là ghi thì giá trị cần ghi phải đặt trên dây bit. Nếu giá trị đó là 1 thì tụ C sẽ được nạp tới điện áp ứng với giá trị 1 trên dây bit, còn nếu giá trị đó là 0 thì tụ C sẽ bị phóng hết điện tức là có giá trị bằng 0.

+ **Việc đọc (read)**

Việc đọc phức tạp hơn việc ghi do điện tích trên tụ C ứng với giá trị cần đọc rất nhỏ. Điện áp trên tụ C điều khiển mức điện áp trên C_B , tức là đưa ra dữ liệu. Giả sử trên C có mức điện áp V, thì sau khi đọc mức điện áp trên dây bit là:

$$V_B = V_x \frac{C}{C + C_B}$$

Do nhiều phần tử nối vào dây bit nên điện dung C_B tương đối lớn ($C_B \gg C$). dữ liệu đọc ra có mức độ điện bé: giá trị mức logic 0 và mức logic 1 trên dây bit không khác xa nhau. Để xác định mức điện áp thay đổi nhỏ trên dây bit cần phải dùng bộ khuếch đại đọc ra có độ nhạy cao.

+ **Làm tươi (refresh)**

Vì mọi tụ điện đều có điện trở rò và transistor mắc nối tiếp với nó dù ở trạng thái cấm cũng có một điện trở rò nhất định, cho nên sau khi được nạp, điện tích trên tụ C sẽ bị phóng điện liên tục, sau một khoảng nào đó sẽ phóng hết điện và không phản ánh chính xác bit dữ liệu nó chứa. chính vì vậy cần phải nạp điện cho tụ điện (nếu lưu thông tin mức 1) trước khi điện áp trên tụ giảm thấp hơn một ngưỡng nào đó, việc làm này được gọi là bù điện áp “làm tươi” (Refresh). Xong vấn đề là không chỉ xét đến các bit nhớ có nội dung là để làm tươi. Để làm tươi các ô nhớ DRAM trong bộ nhớ DRAM, cần phải đọc nội dung của nó rồi viết trở lại. Việc làm tươi này được tiến hành đều đặn theo một chu kì nhất định, gọi là chu kì làm tươi. Tên gọi RAM động xuất phát từ đặc tính và sự hoạt động này.

* **Phần tử nhớ động 3 transistor**

Hoạt động của phần tử nhớ động MOS 3 transistor được mô tả trên hình 4-13, dữ liệu lưu ở điện dung C chính là mức điện áp cực G của transistor T_2 .

Điện áp trên C điều khiển trạng thái của T_2 . Các dây đọc/ghi/bit và dây chọn đọc/ghi đều riêng biệt. Dây chọn đọc điều khiển bóng T_3 , dây chọn ghi điều khiển bóng T_1 . T_4 là bóng bán dẫn mạch nạp trước khi dùng chung cho cột phần tử nhớ của ma trận.

Quá trình đọc: Đầu tiên dây đọc được nạp trước đến mức cao. Tiếp theo, dây chọn đọc nâng lên mức cao làm cho T_3 thông. Giả sử C được nạp điện đến mức lớn hơn điện áp cắt của T_2 (nội dung bit nhớ là 1) làm cho dây đọc bit ở mức thấp. Giả sử C không tích điện (nội dung bit nhớ là 0), thì T_2 ngắt, C_B không có đường phóng điện, mức độ trên dây đọc bit ở mức thấp. Như vậy, tín hiệu (điện áp) trên dây đọc bit là tín hiệu đảo của bit lưu trữ trong C . Tín hiệu ra trên dây đọc bit phải qua bộ khuếch đại đọc và mạch đảo đưa đến đầu ra của bộ nhớ.

Quá trình ghi thông tin: Khi truyền đến dây ghi tín hiệu cần ghi và tín hiệu trên dây chọn ghi ở mức cao, T_1 thông nên tụ C được nạp qua T_1 . Như vậy, ta đã ghi được thông tin vào phần tử nhớ. Mức điện áp đặt ở G (đặt ở tụ C) cùng mức điện áp trên dây bit.

Làm tươi bộ nhớ: Nếu có tín hiệu chọn đồng thời đến T_1 và T_3 thì thông tin đọc được sau đó được khuếch đại qua mạch đảo đưa vào để ghi lại cho tụ C .

* **Phần tử nhớ động 4 transistor**

Dữ liệu (điện tích) lưu trữ trên C_1 và C_2 . Điện áp trên C_1 , C_2 điều khiển T_1 , T_2 thông hoặc ngắt.

Khi C_1 nạp điện tích (điện áp trên C_1 lớn hơn điện áp cắt của T_1) và C_2 không có điện tích (điện áp trên C_2 nhỏ hơn điện áp cắt của T_2) thì T_1 thông, T_2 ngắt, trường hợp này tương ứng với trạng thái 0 của phần tử nhớ.

Khi C_2 nạp điện tích (điện áp trên C_2 lớn hơn điện áp cắt của T_2) và C_1 không có điện tích (điện áp trên C_1 nhỏ hơn điện áp cắt của T_1) thì T_2 thông, T_1 ngắt, trường hợp này tương ứng với trạng thái 1 của phần tử nhớ.

T_3 và T_4 là những bóng để điều khiển sự nối thông phần tử nhớ với dây bit. T_5 và T_6 là mạch nạp trước của dây bit, dùng chung cho tất cả các phần tử nhớ cùng cột trong ma trận nhớ. Khi bắt đầu truy xuất bộ nhớ, trên cực cổng của T_5 và T_6 có xung

nạp trước nên T_5 và T_6 nối thông, do đó trên các dây bit B và \bar{B} có mức điện áp cao vì thông đến nguồn E_D . Sau khi kết thúc xung nạp trước, T_5 và T_6 ngắt, các dây bit cách ly khỏi nguồn E_D . Nhưng do tác động của điện dung phân bố C_B và $C_{\bar{B}}$, khi đó mức điện áp cao của dây bit có thể duy trì thêm một khoảng thời gian nữa.

Trong khoảng thời gian này, giả sử tiến hành đọc dữ liệu, dây X có mức cao, T_3 và T_4 thông. Giả sử phần tử nhớ có trạng thái 0 (T_1 thông, T_2 ngắt) thì G_1 có mức cao, G_2 có mức thấp. Lúc này C_B phóng điện qua T_1 và T_3 . Do đó, dây bit B biến thành dây có mức điện áp thấp. Do T_2 ngắt, dây bit \bar{B} vẫn ở mức cao. Vậy dữ liệu trong phần tử nhớ đó được đọc ra dây bit B và \bar{B} . Nếu dây Y cũng ở mức cao thì tín hiệu sẽ đưa đến đầu ra của RAM qua dây dữ liệu $D (= 0)$ và $\bar{D} (= 1)$.

Vậy, mạch điện nạp trước của dây bit có tác dụng gì? Trong khoảng thời gian T_3, T_4 thông, nếu dây bit không được nạp điện trước thì mức cao có được của dây B chỉ do C_1 phóng qua T_4 nạp vào C_B . Nếu điện tích trên C_1 bị suy giảm, điện áp trên C_B thậm chí lớn hơn trên C_1 (vì có nhiều phần tử nhớ nối vào dây bit). Vậy G_1 có thể không giữ nguyên mức cao sau một lần đọc, tức là dữ liệu bị mất. Nhờ có mạch điện nạp trước, điện thế trên dây bit B còn cao hơn điện thế G_1 một chút. Vậy khi đọc dữ liệu, điện tích trên C_1 không những không bị suy giảm mà còn được làm tươi nhờ sự nạp thêm cho C_1 qua T_4 .

Khi ghi dữ liệu, dây Y và dây X đều có mức điện áp cao, đầu vào dữ liệu của RAM sẽ làm thay đổi trạng thái phần tử nhớ thông qua dây dữ liệu và dây bit, tức là đưa dữ liệu vào trong phần tử nhớ.

Tóm lại, Ba loại phần tử nhớ động vừa trình bày ở trên đều có ưu, nhược điểm như:

- Mạch 4 bóng chiếm diện tích lớn trên chip, nhưng không cần mạch làm tươi riêng, quá trình đọc ra đồng thời là quá trình làm tươi, vì vậy mạch phụ đơn giản.

- Mạch 3 bóng giảm điện tích một chút, nhưng yêu cầu các dây chọn đọc, chọn ghi riêng và dây dữ liệu đều riêng biệt, cần mạch ngoài điều khiển phản hồi để làm tươi, vì vậy dây nối với mạch ngoài tương đối nhiều.

- Mạch 1 bóng là đơn giản nhất, nhưng yêu cầu bộ khuếch đại đọc ra phải có độ nhạy cao và yêu cầu làm tươi mỗi lần đọc dữ liệu, do vậy mạch điện phụ ngoài phức tạp. RAM tĩnh (SRAM) so với RAM động (DRAM): số lượng bóng bán dẫn của SRAM nhiều hơn, độ tích hợp thấp hơn, nhưng không phải làm tươi, mạch phụ đơn giản, tiện sử dụng.

4.4. Bộ nhớ đệm (Cache memory)

4.4.1. Đặc điểm và chức năng của bộ nhớ đệm

Cache hay còn gọi là bộ nhớ đệm, là một thành phần của cấu trúc phân cấp của hệ thống bộ nhớ. Cache đóng vai trò trung gian, trung chuyển dữ liệu từ bộ nhớ chính về CPU và ngược lại

Với các hệ thống CPU cũ sử dụng công nghệ tích hợp thấp, bộ nhớ cache thường nằm ngoài CPU; với các CPU mới sử dụng công nghệ tích hợp cao, bộ nhớ cache thường được tích hợp vào trong CPU nhằm nâng cao tốc độ và băng thông trao đổi dữ liệu giữa CPU và cache.

Dung lượng của bộ nhớ cache thường nhỏ so với dung lượng của bộ nhớ chính và bộ nhớ ngoài. Với các hệ thống máy tính cũ, dung lượng cache là khoảng 16KB, 32KB, 128KB; với các hệ thống máy tính gần đây, dung lượng cache lớn hơn, khoảng 256KB, 512KB, 1MB, 2MB, 4MB, 8MB và 16MB.

Cache có tốc độ truy cập nhanh hơn nhiều so với bộ nhớ chính, đặc biệt với cache được tích hợp vào CPU. Tuy nhiên, giá thành bộ nhớ cache (tính theo bit) thường đắt hơn nhiều so với bộ nhớ chính.

Với các hệ thống CPU mới, cache thường được chia thành hai hay nhiều mức (levels): mức 1 có dung lượng khoảng 16-32KB có tốc độ truy cập rất cao và mức 2 có dung lượng khoảng 1-16MB có tốc độ truy cập hơn hơn.

- + Cache có tốc độ truy xuất nhanh hơn rất nhiều bộ nhớ chính
- + Cache được đặt giữa CPU và bộ nhớ chính nhằm tăng tốc độ trao đổi thông tin giữa CPU và bộ nhớ chính.
- + Cache thường được đặt trong chip vi xử lý

4.4.2. Tổ chức của bộ nhớ Cache

- Bộ nhớ chính được chia làm m block bằng nhau, kí hiệu là B_i [$i = 0 \div (m - 1)$].
- Bộ nhớ cache được chia thành k line, ký hiệu L_j [$j = 0 \div (k - 1)$]. Mỗi line chứa các thông tin sau:

- + Số hiệu thẻ (tag).
- + Bít cờ (F).
- + Phần chứa dữ liệu (Data).

Dung lượng của phần chứa dữ liệu trong một line bằng dung lượng của một block (có thể nói, khối dữ liệu trong cache là bản sao của khối dữ liệu trong bộ nhớ chính và ngược lại).

- Tag dùng để ghi số hiệu block nhớ được nạp vào line (block nhớ là bản copy của line). Bít cờ F cho biết nội dung của line đó ghi ra bộ nhớ hay chưa (trong trường hợp ghi bộ nhớ). Nếu $F = 0$, tức là nội dung line đó ghi vào bộ nhớ chính; nếu $F = 1$, tức nội dung line chưa ghi ra bộ nhớ chính.

- Khi CPU truy nhập dữ liệu, dữ liệu đó đã có sẵn trong cache. Trường hợp này được gọi là trúng (cache hit). Ngược lại, thông tin không tìm thấy trong bộ nhớ cache thì gọi là trượt (cache miss), khi đó CPU phải đọc thông tin từ bộ nhớ chính và thực hiện nạp cache (nạp từ block vào line).

- Nếu xác suất xuất hiện trường hợp hit cao thì hiệu năng hệ thống cao và ngược lại, xác suất xuất hiện trường hợp miss cao thì hiệu năng hệ thống thấp.

Có hai trường hợp tổ chức cache: mỗi khối chứa 1 byte dữ liệu (hoặc một từ dữ liệu) và mỗi khối chứa nhiều byte dữ liệu (hoặc nhiều từ dữ liệu). Trong tài liệu này, chúng ta chỉ xem xét trường hợp máy tính tổ chức, quản lý bộ nhớ trong theo byte dữ liệu (mỗi khối chứa 1 byte dữ liệu và mỗi khối chứa nhiều byte dữ liệu).

4.4.3. Các phương pháp đọc ghi và chính sách thay thế Cache

4.4.3.1. Các phương pháp đọc ghi Cache

Việc trao đổi thông tin giữa CPU – cache và giữa cache – bộ nhớ chính là một trong các vấn đề có ảnh hưởng lớn đến hiệu năng cache. Câu hỏi đặt ra là cần có chính sách trao đổi hay đọc ghi thông tin giữa các thành phần này như thế nào để đạt được hệ số hit cao nhất và giảm thiểu miss.

Xét trường hợp đọc thông tin và nếu đó là trường hợp hit (mẫu tin cần đọc có trong cache): mẫu tin được đọc từ cache vào CPU và bộ nhớ chính không tham gia. Như vậy thời gian CPU truy nhập mẫu tin bằng thời gian CPU truy nhập cache. Ngược lại, nếu đọc thông tin và đó là trường hợp miss (mẫu tin cần đọc không có trong cache): mẫu tin trước hết được chuyển từ bộ nhớ chính vào cache, sau đó nó được đọc từ cache vào CPU. Đây là trường hợp xấu nhất: thời gian CPU truy nhập mẫu tin bằng thời gian truy nhập cache cộng với thời gian cache truy nhập bộ nhớ chính – còn gọi là *miss penalty* (gấp đôi thời gian truy cập khi đoán trượt).

Với trường hợp ghi thông tin và nếu đó là trường hợp hit, có thể áp dụng một trong 2 chính sách ghi: *ghi thẳng* (Write through) và *ghi trả* (Write back).

Với phương pháp ghi thẳng, mẫu tin cần ghi được lưu đồng thời ra cache và bộ nhớ chính. Phương pháp ghi này luôn đảm bảo tính nhất quán dữ liệu giữa cache và bộ nhớ chính, nhưng có thể gây chậm trễ và tốn nhiều băng thông khi tần suất ghi lớn với nhiều mẫu tin có kích thước nhỏ. Ngược lại, với phương pháp ghi trả, mẫu tin trước hết được ghi ra cache và dòng cache chứa mẫu tin sẽ được ghi ra bộ nhớ chính khi nó bị thay thế. Như vậy, mẫu tin có thể được ghi ra cache nhiều lần, nhưng chỉ được ghi ra bộ nhớ chính một lần duy nhất, giúp tăng tốc độ và giảm băng thông sử dụng. Phương pháp ghi trả đang được ứng dụng rãi trong các hệ thống cache hiện nay. Với trường hợp ghi thông tin và nếu đó là trường hợp miss, cũng có thể áp dụng một trong hai chính sách ghi: *ghi có đọc lại* (write allocate / fetch on write) và *ghi không đọc lại* (Write non-allocate). Với phương pháp ghi có đọc lại, mẫu tin trước hết được ghi ra bộ nhớ chính, và sau đó dòng nhớ chứa mẫu tin vừa ghi được đọc vào cache. Việc đọc lại mẫu tin vừa ghi từ bộ nhớ chính vào cache có thể giúp giảm miss đọc kế tiếp áp dụng nguyên lý lân cận theo thời gian: mẫu tin vừa được truy nhập có thể được truy nhập lại trong tương lai gần. Với phương pháp ghi không đọc lại, mẫu tin chỉ được ghi ra bộ nhớ chính. Không có thao tác đọc dòng nhớ chứa mẫu tin vừa ghi vào cache.

4.4.3.2. Các phương pháp thay thế Cache

Như đã đề cập, với cả ba phương pháp ánh xạ bộ nhớ chính – cache, luôn có nhiều dòng nhớ cùng ánh xạ đến một dòng cache. Do có nhiều dòng bộ nhớ chia sẻ một dòng cache, các dòng bộ nhớ được nạp vào cache sử dụng một thời gian và được thay thế bởi dòng nhớ khác theo yêu cầu thông tin phục vụ CPU. Các chính sách thay thế (Replacement policies) xác định các dòng cache nào được chọn để thay thế bởi các dòng khác từ bộ nhớ nhằm đạt hệ số hit cao nhất. Có ba chính sách thay thế cache được sử dụng hiện nay:

- + Thay thế ngẫu nhiên (Random Replacement).
- + Thay thế kiểu vào trước ra trước (FIFO – First In First Out).
- + Thay thế các dòng ít được sử dụng gần đây nhất (LRU – Least Recently Used).

Chương 5

THIẾT BỊ LƯU TRỮ

5.1. Đĩa từ mềm và ổ đĩa mềm (Floppy Disk Drive - FDD)

Đĩa từ mềm còn có tên gọi là đĩa mềm, được chế tạo gồm một đĩa nhựa hình tròn có phủ một lớp sắt từ (thường là hợp chất của sắt hoặc coban) để lưu thông tin. Để bảo vệ khỏi bụi bẩn và hư hỏng khác, đĩa này thường được nắp trong một phong bì giấy cứng

Là một thiết bị dùng để đọc ghi đĩa mềm. Ứng với kích thước của mỗi loại đĩa mềm ta sẽ có một loại ổ đĩa mềm khác nhau. Do ngày nay, người ta chỉ sản xuất loại đĩa mềm kích thước 3½" và hỗ trợ dung lượng cao 1.44MB, khi ta tham khảo bảng giá linh kiện máy tính ở bất cứ đâu ta chỉ thấy mục: FDD 1.44MB hiệu Sony hoặc Mitsumi. Đó chính là loại FDD 3½" high density.

Là phương tiện lưu trữ từ tính (bộ nhớ thứ cấp) có thể tái sử dụng nhiều lần được IBM giới thiệu vào năm 1971. Đĩa mềm là phương thức chủ yếu dùng cho việc phân phối các phần mềm và dữ liệu máy tính cho đến giữa thập kỷ 1990 trước khi đĩa CD-ROM được ưa chuộng như ngày nay. Đĩa mềm được sử dụng ngày nay là loại có vỏ nhựa "cứng" và chứa được 1.44 MB. Tên đĩa "mềm" xuất phát từ lý do là các đời đĩa đầu tiên được bọc trong các lớp vỏ mềm có thể uốn cong, gấp khúc được. Đĩa mềm trở nên "quá nhỏ bé" so với nhu cầu sử dụng ngày nay và tương lai của chúng có vẻ "kém chắc chắn".

Đĩa mềm còn được gọi là "Diskette", là một vật liệu có từ tính (Magnetic material) hình tròn mềm tương tự như băng từ (Magnetic tape), chỉ khác ở chỗ cả hai bề mặt của đĩa mềm được sử dụng để lưu thông tin. Đầu đọc/ghi tiếp xúc với bề mặt qua một khoảng mở qua vỏ nhựa phía trên đầu đĩa. Đầu đọc đĩa mềm (gọi là ổ đĩa mềm - floppy drive) "giữ chặt" vùng trung tâm của vỏ đĩa và làm quay đĩa mềm ở bên trong để truy xuất dữ liệu. Các đĩa mềm quay với tốc độ 300 vòng/phút (300 rpm), chậm hơn từ 10 đến 30 lần so với tốc độ đĩa cứng. Khi không có nhu cầu truy xuất, đĩa mềm sẽ nghỉ ngơi (không quay). Đĩa mềm có hình tròn và được bọc trong lớp vỏ hình vuông. Nó có hai loại với kích thước (đường kính) 5½" và 3½". Loại 5½" là loại cũ và chứa được 360KB (Single density) và 1.2MB (Double-sided & high density). Loại 3½" (loại đang sử dụng ngày nay) có dung lượng tương ứng là 720KB và 1.44MB.

Do có các phương tiện lưu trữ dữ liệu khác có nhiều ưu điểm hơn hẳn so với lưu trữ bằng đĩa mềm nên ổ đĩa mềm hiếm khi còn sử dụng, chúng dần biến mất khỏi các máy tính ngày nay.

5.2. Đĩa cứng

Đĩa cứng (Hard disk) hay còn gọi là đĩa cố định (Fixed disk), lần đầu tiên đưa vào sử dụng trong máy tính của IBM vào đầu năm 1970. Đĩa cứng có nhiều ưu điểm so với các thiết bị nhớ ngoài ra đời trước nó: như dung lượng nhớ lớn, tốc độ trao đổi dữ liệu cao, thời gian truy nhập tương đối ngắn; tổ chức đọc/ghi mềm dẻo; giá rẻ dễ bảo quản, sử dụng.

Dù rằng công nghệ mới không ngừng phát minh nhiều loại bộ phận lưu trữ một lượng thông tin lớn nhưng đĩa cứng vẫn giữ vị trí quan trọng từ năm 1965. Đĩa cứng 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 thay thế nên chúng ta không xét đến thiết bị này trong giáo trình mà chỉ nói đến đĩa cứng.

Một đĩa cứng chứa nhiều lớp đĩa (từ 1 đến 4) quay quanh một trục khoảng trên 3.600 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.

5.2.1. Cấu tạo đĩa từ cứng

Đĩa cứng thường gồm các thành phần chính sau:

- + Các đĩa từ (Disks), các đầu từ đọc/ghi (Heads), các rãnh (Tracks), các mặt trụ (Cylinders) và các cung (Sectors).
- + Một ổ đĩa cứng có thể gồm một hoặc nhiều đĩa được lắp đồng trục. Các đĩa thường phẳng và được chế tạo bằng nhôm hoặc thủy tinh với lớp bột từ rất mỏng (khoảng 10 ÷ 20 nm) phủ trên bề mặt đĩa để lưu thông tin. Vật liệu từ thường dùng là

oxit sắt ba (Fe_2O_3) với các ổ đĩa cứng cũ. Hiện nay vật liệu từ thường dùng là hợp kim của coban.

+ Đĩa có thể lưu thông tin trên cả hai mặt (Side), được đánh số mặt 0 và mặt 1.

+ Đầu từ hay đầu đọc ghi cũng là một trong các bộ phận chủ chốt của ổ đĩa cứng. Mỗi đầu từ đĩa cứng thường có kích thước rất nhỏ, được sử dụng để đọc và ghi thông tin lên đĩa. Khoảng cách giữa đầu từ và bề mặt đĩa là rất nhỏ, nhưng không tiếp xúc mà “bay” trên mặt đĩa. Mỗi ổ đĩa cứng thường có nhiều đầu từ kết hợp thành một hệ thống đầu từ trên cùng một giá đỡ. Số lượng đầu từ của mỗi ổ đĩa phụ thuộc vào thiết kế và dung lượng đĩa và thường rất khác nhau: 4, 8, 12, 16, 24, 32, 64.

+ Rãnh có dạng là một đường tròn đồng tâm trên mặt đĩa để lưu thông tin. Các rãnh được đánh số từ 0 theo trật tự từ phía ngoài đĩa vào trong tâm và mỗi mặt đĩa có thể chứa hàng ngàn rãnh. Tiếp theo rãnh, mặt trụ là tập hợp của các rãnh ở các mặt đĩa khác nhau nằm trên cùng một vị trí đầu từ. Trên thực tế, mặt trụ là tham số được sử dụng nhiều hơn rãnh trong các hệ thống đĩa cứng. Cung là một phần của rãnh trên bề mặt đĩa và là đơn vị lưu trữ nhỏ nhất có thể quản lý của đĩa. Kích thước thông dụng của mỗi cung là 512 byte.

Với ổ đĩa cứng, ba tham số được sử dụng để tính dung lượng đĩa là: số rãnh, số lượng cung và mật độ cung trong một rãnh. Như vậy, dung lượng của đĩa cứng tính theo các tham số trên là:

Số rãnh = số xi lanh \times số đầu từ

Dung lượng của đĩa cứng = số rãnh \times số cung \times mật độ cung.

Ổ đĩa cứng thường có 17 cho tới 36 cung/rãnh và 512 byte/cung phụ thuộc vào từng loại đĩa.

Trong khi đĩa mềm chỉ quay khi truy xuất thông tin trên đĩa thì đĩa cứng luôn quay trong quá trình máy tính hoạt động. Hiện nay có những chương trình cho phép tiết kiệm năng lượng, tức là khi gặp khoảng thời gian dài không có sự truy xuất thì chương trình sẽ tự động giảm tốc độ quay hoặc ngừng hẳn công việc quay đĩa nhưng về nguyên tắc trong một phiên làm việc, đĩa luôn được quay với tốc độ không đổi.

Dữ liệu cũng được tổ chức thành các rãnh từ, sector, liên cung như trong đĩa mềm và đặc biệt là có từ trụ. Đầu từ đọc viết cũng được chuyển động bởi cần truy xuất, nhưng các chi tiết cơ khí và mạch điều khiển ổ đĩa phải có những điểm đặc biệt để có thể làm việc được với các đĩa có dung lượng nhớ lớn (mật độ ghi lớn) cùng các chỉ tiêu kỹ thuật khác cao hơn so với ổ đĩa mềm. Các chi tiết này phải được bảo vệ cẩn thận để tránh bụi bẩn. Các đĩa từ cứng, các đầu từ, cần phải truy xuất và cơ chế quay nằm trong hộp đĩa được bọc kín, còn mạch điều khiển điện tử được lắp trên bản mạch bên ngoài ổ đĩa. Tốc độ của mô tơ quay như ta đã biết là khoảng 3600 vòng/phút được ổn định bởi một mạch phản hồi cho phép một sai số cỡ $\pm 0.3\%$ và ở những ổ đĩa chất lượng cao là khoảng $\pm 0.1\%$.

Đến nay, các đĩa cứng hiện đại đã có tốc độ lên đến 7200 vòng/phút. Điều này có nghĩa là các phần tử từ trên đĩa cứng chuyển động tương đối so với một giá trị của đầu từ nhanh gấp nhiều lần so với đĩa mềm. Các đĩa cứng đơn giản nhất cũng có từ 17 sector/rãnh đến 70 sector/rãnh, lượng dữ liệu đĩa qua đầu từ được nâng lên khoảng 100 lần so với 360 Kbyte ở ổ đĩa mềm. Do đó, đĩa cứng phải có các bộ phận điều khiển mạnh hơn.

Các ổ đĩa cứng thường có từ 2 đến 8 đĩa hai mặt, do đó phải cần từ 4 đến 16 đầu từ. Tất cả các đầu từ này đều được gắn ở đầu của mỗi cần truy xuất được nằm bên trong hộp kín và cùng được chuyển động với nhau.

Khác với cấu tạo của ổ đĩa mềm, khi đĩa cứng quay vòng, đầu từ không chạm vào mặt đĩa mà bay lơ lửng trên bề mặt đĩa. Khi đĩa quay, nó cuốn không khí trên bề mặt và tạo nên một dòng khí xoáy. Dòng khí này đủ mạnh để tạo ra một lớp đệm khí trên mặt đĩa cho phép nâng các đầu từ lên khỏi mặt đĩa với một khoảng cách rất nhỏ, khoảng cách này phụ thuộc vào loại đầu từ, hình dạng của đầu từ, bề dày của lớp đệm không khí, phụ thuộc vào áp suất không khí, áp suất càng thấp thì lớp đệm càng mỏng. Khoảng cách này được thiết kế tùy thuộc vào tốc độ quay cũng như mật độ ghi tin của đĩa cứng.

Trong ổ đĩa cứng thường có một đầu từ được kích hoạt cho việc đọc hoặc viết dữ liệu tại một thời điểm. Cần truy xuất được chuyển động bởi mô tơ bước hoặc mô tơ tuyến tính. Các mô tơ này được dùng để định vị đầu từ trên các rãnh hoặc các từ trụ cần truy xuất. Trên các ổ đĩa cứng thường có một đèn LED dùng để chỉ thị khi xảy ra quá trình đọc hoặc ghi dữ liệu.

Các đầu từ càng nhỏ thì càng tạo được các ô bit nhỏ, tức là đĩa có mật độ ghi thông tin cao hơn. Đầu từ nhỏ tức là có trọng lượng nhỏ, có quán tính nhỏ nên khi chuyển động tới một vị trí mong muốn nào đó trong đĩa sẽ nhanh hơn. Hiện nay có hai công nghệ chế tạo ra đầu từ là: đầu từ Ferrit và đầu từ màng mỏng (Thin film)

- Đầu từ Ferrit có giá thành thấp và độ tin cậy do công nghệ chế tạo ổn định, chúng phát ra từ trường bởi cuộn dây cuốn quanh lõi Ferrit.

- Đầu từ màng mỏng có kích thước và trọng lượng nhỏ hơn, được chế tạo theo công nghệ vi cơ điện tử như trong chế tạo các công cụ bán dẫn. Đó là việc phủ và ăn mòn để tạo nên các lớp bán dẫn khác nhau có những thuộc tính điện từ nào đó thay thế cho cuộn dây và lõi Ferrit, ví dụ như loại đầu từ màng mỏng dùng hiệu ứng từ khổng lồ GMR (Giant Magnetic Resistance). Các đầu từ màng mỏng này rất nhỏ và nhẹ nhưng lại phát ra một từ trường hoặc thu được một tín hiệu điện đủ mạnh trong một khoảng cách giữa đầu từ và mặt đĩa.

Khi ổ đĩa cứng bị mất nguồn điện, đầu từ bị dừng lại trên một rãnh không dùng để trữ dữ liệu gọi là *landing zone*. Điều này để tránh cho việc cọ xát vào các rãnh trữ dữ liệu khi tăng tốc cho đĩa quay lúc khởi động máy. Cần truy xuất gồm tất cả các cấu kiện cơ khí cần thiết dùng cho chuyển động của các đầu từ. Hiện có thể dùng mô tơ bước hoặc mô tơ tuyến tính để chuyển dịch cần này. Mô tơ bước trong đĩa cứng cũng được quay từng bước một bởi các xung bước (mỗi bước là một giá trị góc quay xác định nào đó). Chuyển động quay của mô tơ này qua các chi tiết cơ khí được biến thành chuyển động tịnh tiến của cần truy xuất theo phương bán kính của đĩa để định vị các đầu từ tới vị trí mong muốn. Thường một bước quay của mô tơ được thiết kế tương ứng với một chuyển dịch của đầu từ đi một rãnh từ trên đĩa. Khi dùng mô tơ bước thì không cần mạch điều khiển phản hồi. Do đó, nếu cần chuyển đầu từ từ rãnh 0 đến rãnh 219 thì bộ điều khiển chỉ đơn giản cấp 219 xung tới mô tơ bước là xong. Với đĩa mềm có bề rộng rãnh cỡ 0.1 mm, cách xử lý này là đủ độ chính xác cho phép định vị đầu từ tới giữa rãnh. Nhưng với những đĩa cứng kích thước 3.5" với hơn 1.000 rãnh thì việc sử dụng mô tơ bước như vậy không đảm bảo độ chính xác khi định vị đầu từ. Để định vị chính xác đầu từ, không chỉ độ chính xác mà cả độ lặp lại phải cao. Một vấn đề khác là sự biến dạng nhiệt liên quan tới việc sử dụng mô tơ bước. Do sự biến dạng nhiệt của các linh kiện cơ khí, ví dụ như cần truy xuất bị dài ra do nhiệt làm cho lượng dịch chuyển của nó coi như dài hơn và có thể gây ra lỗi định vị trên các rãnh từ. Tóm lại, các sai số này xảy ra do không có một sự phản hồi về vị trí thực sự của đầu từ và lượng điều khiển của mô tơ để mà hiệu chỉnh. Các ổ đĩa cứng hiện đại lại quay về sử dụng mô tơ tuyến tính thông thường thay cho mô tơ bước, chỉ có điều thêm vào đó một

mạch điện tử cho phép so sánh vị trí hiện tại của đầu từ với vị trí cần định vị để hiệu chỉnh và do đó đầu từ luôn được định vị chính xác.

Lọc khí và thông khí cũng là một vấn đề, tất cả các ổ đĩa cứng đều có một bộ lọc khí. Bộ lọc này không dùng cho lọc khí từ bên ngoài xâm nhập vào ổ đĩa mà chủ yếu dùng để lọc khí đã ở bên trong hộp ổ đĩa. Nếu áp suất không khí ở môi trường thay đổi thì chỉ xảy ra một sự trao đổi khí nhỏ mà thôi. Vì vậy, hầu như không phải thay bộ lọc khí trong suốt thời gian hoạt động của ổ đĩa. Khi đĩa chuyển động sẽ tạo nên một dòng không khí xoáy trên mặt đĩa nâng đầu từ lên. Mỗi ổ đĩa đều có một khe thông khí dẫn tới bộ lọc khí bên trong. Tùy vào áp suất khí của môi trường mà không khí có thể thấm vào hộp ổ đĩa hoặc thoát ra từ đó.

5.2.2. Các đặc trưng chính của đĩa từ

Những đặc trưng chính của đĩa bao gồm: Sự di chuyển của đầu đọc; Tính khả chuyển của đĩa; Các mặt đĩa; Các tấm đĩa; Cơ chế làm việc của đầu đọc.

5.2.2.1. Sự di chuyển của đầu đọc

Đầu đọc có thể được giữ cố định hoặc di chuyển theo hướng tâm đĩa. Với đầu đọc cố định, có một đầu đọc/ghi cho mỗi track. Tất cả các đầu đọc được gắn lên một tay đòn kéo dài qua tất cả các track trên đĩa. Với đầu đọc có thể di chuyển, chỉ có một đầu đọc/ghi duy nhất. Ở đây đầu đọc cũng được gắn lên một tay đòn có thể kéo dài hoặc thu ngắn ứng với vị trí của track cần truy cập trên đĩa.

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.

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.

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 ngừng kết nối) lúc đọc (hoặc ghi) thông tin, và việc đọc trước, ổ đĩa cần có bộ điều khiển đĩa.

5.2.2.2. Tính khả chuyển của đĩa

Bản thân đĩa thường được gắn vào trong một ổ đĩa với một tay đòn, một động cơ quay đĩa cũng như các mạch điện tử cần thiết cho việc nhập/xuất dữ liệu nhị phân. Đĩa không thể tháo rời được gắn vĩnh viễn vào trong ổ đĩa trong khi đĩa có thể tháo rời có thể lấy ra và thay thế bằng một ổ đĩa khác. Lợi ích của loại đĩa có thể tháo rời là khả năng cung cấp một lượng dữ liệu không có giới hạn trên một hệ thống giới hạn các ổ đĩa. Hơn nữa đĩa đó còn có thể di chuyển từ máy tính này sang máy tính khác.

5.2.2.3. Cơ chế làm việc của đầu đọc

Cơ chế làm việc của đầu đọc trong thao tác đọc/ghi cho phép phân loại đĩa rất rõ ràng. Chúng ta có ba cơ chế như sau:

Cơ chế 1: giữ đầu đọc ở một khoảng cách cố định với bề mặt đĩa.

Cơ chế 2: cho phép tiếp xúc về mặt vật lý giữa đầu đọc với bề mặt đĩa.

Cơ chế 3: khoảng cách giữa đầu đọc với bề mặt đĩa có thể thay đổi trong quá trình đọc/ghi.

Để hiểu rõ cơ chế 3 nói trên, chúng ta cần tìm hiểu về mối quan hệ giữa độ trữ mật dữ liệu và kích thước của khối không khí trống giữa đầu đọc và bề mặt đĩa bên dưới nó.

Đầu đọc phải sinh ra một trường điện từ đủ lớn để đọc/ghi một cách chính xác. Đầu đọc càng hẹp thì nó càng phải ở gần bề mặt đĩa hơn để hoạt động. Đầu đọc hẹp hơn dẫn đến bề rộng track nhỏ hơn và do vậy độ trữ mật dữ liệu tăng lên rất nhiều. Tuy nhiên, khi đầu đọc càng gần bề mặt đĩa, nguy cơ pha tạp và không chính xác về dữ liệu càng lớn.

Nhằm đẩy kỹ thuật đĩa lên một bước xa hơn, người ta đã chế tạo ra ổ đĩa Winchester. Các đầu đọc Winchester được sử dụng trong các ổ đĩa gần như không có tác nhân tạp chất. Chúng được thiết kế để hoạt động gần hơn với bề mặt đĩa nếu so với các đầu đọc kiểu cũ, do đó cho phép làm tăng độ trữ mật của dữ liệu trên đĩa. Đầu đọc có dạng một lá khí động học nằm trên bề mặt đĩa khi đĩa không chuyển động. áp suất không khí sinh bởi đĩa xoay tròn đủ làm cho lá rút lên khỏi bề mặt đĩa. Phần hệ thống không tiếp xúc với đĩa có thể được chế tạo để sử dụng với những đầu đọc kiểu mới này.

5.2.3. Dung lượng đĩa và tốc độ truyền dữ liệu

5.2.3.1. Dung lượng trữ tin

Dung lượng trữ tin của đĩa sau khi đã định dạng thường nhỏ hơn tới 20% trước khi định dạng vì rằng phải mất một số byte cho các trường địa chỉ, các khe phân cách. Đơn vị chỉ thị dung lượng thường là MB nhưng đôi khi có thể là *miobyte* (Million byte) nhỏ hơn MB là 5%. Mật độ lưu trữ thông tin trên đĩa đến nay được tính theo số Gbit trên một inch vuông. Các ổ đĩa sử dụng hiệu ứng từ trở khổng lồ GMR (Giant Magneto Resistive) cho mật độ lên tới 4.2 Gbit/inch.

5.2.3.2. Thời gian truy xuất dữ liệu

Thời gian truy xuất dữ liệu của ổ đĩa là một thông số quan trọng ảnh hưởng tới việc xử lý số liệu trong máy tính, thời gian này là tổng hợp của thời gian tìm kiếm (Seek time) thời gian chuyển đầu từ (Head switch time) và thời gian quay trễ (Rotational latency). Thời gian tìm kiếm là thời gian chuyển động đầu từ từ một rãnh này sang rãnh khác, thời gian tìm kiếm chậm nhất ứng với hai rãnh trong và ngoài cùng của đĩa, thời gian tìm kiếm trong bình bằng thời gian cần có để đặt đầu từ tới một vị trí được yêu cầu ngẫu nhiên. Thời gian chuyển đầu từ là một thời gian ổ đĩa phải bỏ ra để chuyển giữa hai trong số các đầu từ khi đọc hay viết dữ liệu.

Thời gian quay trễ là thời gian tính từ khi đầu từ được tạo lên một rãnh đến khi nó tới được sector mong muốn. Tất cả các thông số trên đầu được đo bằng mili giây (ms). Ổ đĩa có thời gian truy xuất trên 40 ms là loại chậm, từ 40 ms ÷ 28 ms là trung bình, từ 28 ms ÷ 18 ms là loại nhanh, dưới 18 ms là loại rất nhanh. So với thời gian thâm nhập từ 130 ms ÷ 80 ms của ổ đĩa CD và DVD, thời gian thâm nhập của các ổ đĩa cứng hiện đại là rất nhanh cỡ từ 15 ms ÷ 3 ms.

5.2.3.3. Tốc độ truyền dữ liệu của ổ đĩa

Tốc độ truyền dữ liệu của ổ đĩa phụ thuộc vào 3 nhân tố chính là sự truyền tín hiệu giữa đĩa và bộ điều khiển, xử lý trong bộ điều khiển và việc truyền tín hiệu giữa bộ điều khiển và bộ nhớ chính. Nhân tố một bị hạn chế bởi số byte chuyển qua đầu từ trong mỗi giây. Với đĩa có 17 sector/rãnh tại tốc độ quay 3600 vòng/phút, giá trị này cỡ 600 KB/giây tương đương với tốc độ danh định 5Mbit/giây của ghép nối ST412 với mã MFM. Bộ điều khiển sẽ tách tất cả các byte không thuộc trường dữ liệu và như vậy

chỉ còn lại 510 KB. Số Kbyte được đi qua đầu từ trong một giây này chỉ cho một sector riêng biệt, ta phải viết 512 byte/980 μ s vì một sector gồm 512 byte dữ liệu cùng các trường địa chỉ. Tiếp đó dữ liệu nằm trong bộ đệm sector và phải được truyền tiếp nữa vào bộ nhớ chính. Nếu bộ điều khiển và hệ thống nhớ được nhanh để thực hiện việc này trước khi sector tiếp theo xuất hiện (với hệ số đan xen của đĩa bằng 1:1), tốc độ truyền này giữ nguyên trong toàn rãnh bằng tốc độ hiệu dụng là 510 KB/giây. Thời gian xử lý trong bộ điều khiển với kiểm tra CRC và ECC thường rất ngắn nên có thể bỏ qua, nhưng tốc độ truyền giữa bộ điều khiển và bộ nhớ chính qua bus hệ thống máy tính trong khoảng thời gian ngắn đến khi cung từ tiếp theo xuất hiện là một vấn đề. Khoảng thời gian giữa hai trường dữ liệu của các sector kế tiếp cỡ 100 μ s. Vậy tốc độ truyền vào bộ nhớ phải tương đương 512byte/100 μ s hoặc 5 Mbyte/s, do đó bus hệ thống của máy tính phải có khả năng đảm bảo một tốc độ ít nhất như vậy để có hệ số đan xen 1:1. Bus ISA chỉ có tốc độ cực đại là 8.33 MHz hay tốc độ truyền dữ liệu là 16.67 Mbyte/giây. Chế độ truyền khối (Burst mode) của bus EISA hoặc chế độ DMA có tốc độ truyền là 33.3 MB/giây. Do đó dữ liệu được lập mã RLL với ghép nối ST412 và đĩa 26 sector/rãnh sẽ làm cạn kiệt dung lượng của bus ISA. Do vậy, các ổ đĩa dung lượng cao tới 70 sector/rãnh và hệ số đan xen bằng 1:1 đều có một đĩa cache (hoặc ít nhất một rãnh thay cho bộ đệm sector). Cache hoạt động nhanh hơn việc truyền qua bus ISA hoặc EISA và có thể chứa được toàn bộ nội dung của một rãnh. Sự truyền vào bộ nhớ chính qua bus máy tính có thể thực hiện một cách liên tục và dung lượng truyền 8.33 MB/giây hoặc 16.67 MB/giây có thể được sử dụng cho thời gian dài hơn 100 μ s giữa hai sector trong trường hợp đĩa được lập mã MFM với ghép nối ST412.

Đầu đọc là một thiết bị tương đối nhỏ có khả năng đọc/ghi từ/lên một phần của tấm đĩa quay bên dưới. Điều này dẫn đến việc tổ chức dữ liệu trên đĩa theo dạng một tập hợp các vòng tròn đồng tâm gọi là các track (rãnh). Mỗi track có cùng độ rộng như đầu đọc.

Các rãnh kề nhau được phân biệt bằng các khoảng trống (gap), nhờ đó ngăn chặn ở mức tối thiểu các sai sót do sự canh biên không chính xác của đầu đọc hoặc nhiễu của các trường điện từ. Để đơn giản hóa về mặt điện từ, mỗi track có số bit thông tin bằng nhau. Do đó độ trù mật tính theo số bit trên mỗi inch tuyến tính sẽ tăng lên khi chúng ta đi từ track xa nhất bên ngoài vào đến track gần nhất bên trong đĩa.

Dữ liệu được chuyển đến và đi khỏi đĩa theo từng khối một. Thông thường kích thước của một khối dữ liệu nhỏ hơn dung lượng của track. Cụ thể hơn, dữ liệu được lưu trong các vùng có kích thước khối gọi là *sector* (cung). Trên mỗi track thường có từ 10 đến 100 sector dữ liệu và chúng có thể có độ dài cố định hay thay đổi. Để tránh những sai sót về độ chính xác có thể xảy ra trong hệ thống, các sector liên nhau được tách biệt bởi các gap liên track (liên – bản ghi).

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 cách tổ chức như vậy, để xác định từng sector trong track, cần phải có một điểm bắt đầu của track cũng như cách thức nhận biết điểm bắt đầu và kết thúc của một sector. Các yêu cầu này được quản lý thông qua dữ liệu điều khiển đã được ghi lên đĩa.

Do vậy đĩa sẽ được định dạng với một số dữ liệu phụ đi kèm và chỉ được sử dụng bởi ổ đĩa. Người sử dụng không thể truy cập đến những dữ liệu này.

5.2.4. Định dạng và tổ chức dữ liệu

5.2.4.1. Định dạng cấp thấp

Đĩa cứng phải được định dạng cấp thấp (Low – level format) trước khi tạo phân khu và định dạng cấp cao bằng lệnh Format. Đó là việc tạo ra những rãnh và sector cùng các thông tin liên quan đến chúng trên đĩa. Trong quá trình định dạng cấp thấp, chương trình cũng đánh dấu các sector hỏng để hệ điều hành không sử dụng đến nữa cùng với việc phân định hệ số đan xen.

Đó là việc đánh dấu các sector logic liên tiếp không theo các tuần tự liên tiếp nhau mà cách ra n sector nếu thời gian xử lý dữ liệu của hệ thống cho một sector dài hơn thời gian phân cách giữa hai sector liên tiếp.

Khác với đĩa mềm, lệnh Format định dạng cấp cao trong đĩa cứng chỉ xác định cấu trúc của các phân khu đĩa. Nó tạo ra các sector khởi động, hai bảng FAT và thư mục gốc. Tất cả các điểm vào của FAT được điền số 00h để chỉ ra phân khu còn trống.

Khi định dạng lại một đĩa cứng thì chỉ có cấu trúc logic bị xóa (giống như lệnh DEL hay RMDIR) chứ về mặt vật lý các dữ liệu vẫn được giữ nguyên.

5.2.4.2. Kỹ thuật mã hóa dữ liệu ở ổ đĩa cứng

Dữ liệu được ghi trên đĩa dưới dạng các phần tử mã nhị phân được gọi là ô bit. Bit có giá trị cao nhất MSB của byte được ghi đầu tiên, byte có giá trị thấp nhất LSB được ghi sau cùng. Có nhiều phương pháp để biểu diễn một phần tử mã hóa trong một ô bit và do tốc độ truyền dữ liệu cũng sẽ khác nhau. Các phương pháp hiện nay thường sử dụng đó là điều biến tần số, điều biến tần số cải biến và độ dài giới hạn không đảo từ thông.

5.2.4.3. Hệ số đan xen (Interleaving)

Khi chuyển động theo các rãnh, đầu từ phải đọc từng cung và chuyển dữ liệu đọc được tới bộ điều khiển đĩa. Bộ điều khiển đĩa lại chuyển tiếp dữ liệu qua các bus tới máy tính. Việc đọc và truyền dữ liệu cần có một thời gian nhất định, trong khi đĩa cứng quay với tốc độ rất cao, do vậy xuất hiện vấn đề đồng bộ giữa đầu từ với các cung dữ liệu cần đọc. Nếu đầu từ và bộ điều khiển đĩa không theo kịp với cung dữ liệu kế tiếp thì có hai giải pháp khắc phục, đó là: hoặc giảm tốc độ quay của đĩa hoặc sử dụng giải pháp đan xen.

Nếu dùng giải pháp giảm tốc độ quay của đĩa, ví dụ từ 3600 rpm xuống còn 600 rpm thì thời gian truy nhập đĩa được tăng đến mức khó chấp nhận được. Cho nên trong thực tế, chỉ còn giải pháp thứ hai là sử dụng kỹ thuật đan xen. Chúng ta đã biết, đầu từ và bộ điều khiển không thể vừa xử lý dữ liệu ở cung 1 lại sẵn sàng kịp cho cung 2 (các cung được đánh số tuần tự). Bởi vậy, cung 2 chỉ có thể được xử lý ở vòng quay tiếp theo. Cũng tương tự, khi cung 2 được xử lý thì cung 3 chỉ có thể được xử lý ở vòng quay tiếp theo. Có nghĩa là để có thể đọc được hết 17 cung của mỗi rãnh thì phải cần tới 17 vòng quay. Hệ số đan xen như vậy có giá trị 1:1.

Trường hợp hệ số đan xen là 2:1, các cung được truy nhập xen kẽ. Nếu bộ điều khiển không đủ nhanh, người ta sử dụng hệ số đan xen là 3:1. Ở trường hợp này, cứ đến cung thứ 3 thì sẽ truy nhập dữ liệu một lần. Như vậy, với hệ số đan xen là 2:1 thì phải cần đến 2 vòng để truy nhập toàn bộ dữ liệu trên rãnh, còn với hệ số 3:1 thì cần tới 3 vòng. Hệ số đan xen 3:1 hiện được sử dụng rộng rãi ở trong các máy tính PC/XT. Tuy nhiên, một số máy tính có tốc độ cao sử dụng bộ điều khiển IDE và SCSI thì với hệ số đan xen 1:1 vẫn có thể đọc được toàn bộ rãnh của đĩa trong một vòng với lý do là bộ điều khiển có tốc độ lớn và bus dữ liệu rộng.

5.2.5. Tổ chức logic của đĩa cứng

Về mặt logic, đĩa từ được tổ chức thành các đơn vị sau:

5.2.5.1. Sector logic

BIOS dùng các *sector vật lý* như trên để quản lý số liệu trong khi DOS và nhiều hệ điều hành khác lại dùng một sơ đồ khác gọi là sector logic.

Đó là cách đánh số các sector vật lý một cách liên tục từ 0 lần lượt từ mặt đĩa này tới mặt đĩa kia, từ đĩa này tới đĩa tiếp theo.

5.2.5.2. Phân khu (Partition)

Một ổ đĩa cứng có thể được chia thành các thành phần khác nhau gọi là phân khu và DOS coi các phần tử đó như các ổ đĩa độc lập gọi là ổ đĩa logic được gán cho các tên từ C đến Z. Các phân khu trên đĩa cứng được tạo lập và kiểm tra bởi chương trình FDISK.EXE. Nội dung của chúng có thể được khảo sát bằng các chương trình như DISKEDIT.EXE trong gói chương trình Norton Utilities NU.

Có ba loại phân khu trên đĩa cứng:

- + Phân khu DOS chính (primary DOS partition).
- + Phân khu DOS mở rộng (extended DOS partition).
- + Phân khu phi DOS (non-DOS partition).

Để lưu trữ thông tin về các phân khu, hệ điều hành dùng một vùng cố định trong mọi đĩa cứng: đầu từ rãnh 0, rãnh 0, sector vật lý 1. Đây là sector vật lý đầu tiên của đĩa cứng và sector này cũng được thường gọi là *sector phân khu* (Partition sector). Thông tin về từng phân khu được lưu trữ bởi các điểm vào phân khu trong bảng phân khu (Partition table).

5.2.5.3. Boot sector (Boot record)

Vùng này dài 512 byte đặt tại sector logic 0 của ổ đĩa logic. Nó chứa một chương trình ngắn đặc biệt gọi là *chương trình khởi động* (Bootstrap loader), dùng để nạp lõi DOS lên RAM để khởi động máy tính. Bảng 5-4 cho cấu trúc của một boot sector

5.2.5.4. Bảng định vị FAT (File Allocation Table)

Hệ điều hành lưu các file lên đĩa theo các đơn vị gọi là *liên cung* (cluster). Mỗi liên cung chứa một nhóm gồm một hoặc thành nhiều sector (ví dụ, trên đĩa cứng, một cluster gồm 4 hoặc 8 sector nghĩa là dài 2KB hoặc 4KB). Để theo dõi những liên cung nào được sử dụng, liên cung nào còn trống, hệ điều hành dựa trên một cấu trúc gọi là bảng FAT sau khi tham khảo thư mục gốc.

Hai điểm vào đầu tiên để nhận dạng loại đĩa, các điểm vào sau cho biết: tình trạng của liên cung (tự do hay đã bị chiếm dụng,...). Phần này bao giờ cũng ghi số liên cung đang trữ phần dữ kiện kế tiếp của file nếu file nằm trải dài trên nhiều liên cung. Bảng sau cho biết ý nghĩa nội dung của một điểm vào FAT-16:

Nhiều khi các liên cung của một file không liền kề như trên mà cách xa nhau. Có những phần mềm tiện ích như Diskedit.exe trong Norton Utility cho ta ngay trị số của bảng FAT rất tiện dụng trong việc khảo sát, sửa chữa nó. Với cách tổ chức bảng FAT để quản lý file như trên, các file không nhất thiết phải ghi lên các liên cung liên tục kế cận nhau mà còn có thể nằm rải rác ở nhiều nơi. Do vậy tiết kiệm được chỗ lưu trữ (hiện tượng phân mảnh - fragmentation). Khi xóa file chỉ cần cho điểm vào tương ứng bằng 0000h để báo cho biết liên cung đó còn trống.

5.2.6. Ghép nối bộ điều khiển và ổ đĩa cứng

Một chương trình máy trong ROM vì mã được một vi xử lý chuyên dụng trên bộ điều khiển sử dụng. Dữ liệu ghi, đọc trên sector được gửi bởi bộ đệm sector. Bộ đệm này luôn trữ đủ nội dung toàn bộ của một sector. Mạch ghép nối ổ đĩa được thiết

kể rất khác nhau tùy thuộc chúng là loại gì ST412/506, ESDI hay điều khiển chuyên dụng. Mạch điều khiển chuyên dụng được tích hợp ngay trên ổ đĩa và chỉ dùng cho riêng ổ đĩa.

Để đọc một sector, vi xử lý truyền một khối lệnh (gồm các lệnh và các byte thông số) tới một thanh ghi trong mạch ghép nối máy tính. Vi xử lý của bộ điều khiển phát hiện các lệnh này và dịch lã lệnh. Bộ điều khiển thực hiện lệnh thích hợp (đây là việc đọc 1 sector) dưới sự điều khiển của chương trình trong ROM vi mã. Nó sẽ điều khiển ổ đĩa định vị đầu từ trên rãnh bằng các xung cấp tới mô tơ. Đầu sẽ liên tục đọc các tín hiệu phát ra từ các ô bit đi qua dưới nó, truyền các tín hiệu phù hợp qua qua ghép nối ổ đĩa tới một bộ phận tách dữ liệu. Tùy thuộc vào phương pháp ghi, bộ phận tách sẽ biến đổi dữ liệu được lập mã RLL thành các mã trung gian NRZ (non-return to zero). Vi xử lý sẽ xác định các trường địa chỉ nhận dạng và dữ liệu đã đi qua đầu từ. Nếu nó phát hiện được dấu địa chỉ mong muốn thì nó sẽ xác định xem có đúng là rãnh và sector cần thiết không. Nếu không đúng nó sẽ hướng dẫn ổ đĩa tiến hành việc định vị khác và đợi cho tới dấu địa chỉ tiếp theo sau. Nếu số sector đúng bằng số mong muốn, bộ điều khiển sẽ mở cổng đọc và chấp nhận dữ liệu đã đọc từ đĩa. Số liệu được giải mã sẽ truyền vào bộ đệm sector. Cùng lúc đó mạch sửa lỗi ECC kiểm tra các dữ liệu của sector có phù hợp với các byte ECC hay không. Nếu không thì dữ liệu được sửa bởi các byte này.

5.3. Đĩa và ổ đĩa quang

5.3.1. Đĩa quang

Đĩa laser CD (Compact Disk) ngày nay không những được sử dụng mạnh trong lĩnh vực nghe-nhìn (audi-video) mà còn được dùng làm bộ nhớ dung lượng lớn gọi là đĩa quang thay cho đĩa từ trong máy vi tính. Giống như đĩa từ, việc ghi và đọc đĩa quang được thực hiện trong ổ đĩa quang. Đĩa quang có thể tháo lắp vào ổ đĩa từ mềm được. Ngoài kĩ thuật đọc đĩa CD thông thường, ổ đĩa quang chỉ thêm một bộ ghép nối để truyền số liệu tới bus hệ thống của máy vi tính và một ít linh kiện ghép nối nhằm cho vi xử lý truy nhập các số liệu nhất định với những lệnh phần mềm, đĩa quang loại CD-ROM dùng trong máy tính có dung lượng điển hình cỡ 680 MB. Các ổ đĩa CD-ROM đầu tiên có tốc độ truyền số liệu cỡ 150 KB/s và với tốc độ quay tăng từ 530 tới 1,060 vòng/phút (tốc độ 2x), tốc độ truyền dữ liệu tăng lên cỡ 300 KB/s. Thời gian thâm nhập dữ liệu chậm từ 80 đến 150 ms.

Nguyên tắc hoạt động của ổ đĩa quang như sau: Bên trong ổ đĩa có một diốt laser bán dẫn nhỏ phát ra một tia laser rất mảnh với bước sóng chừng 850 nm ở vùng gần hồng ngoại nên không nhìn thấy được. Một gương quang học, được chuyển động một cách rất chính xác nhờ bộ điều khiển điện tử, sẽ hội tụ tia laser thành một điểm rất nhỏ lên bề mặt của đĩa quang. Ánh sáng phản xạ sẽ được thu nhận bởi một cảm biến thu. Tín hiệu lỗi ra của cảm biến chứa thông tin về các bit dữ liệu được ghi lên đĩa. Dữ liệu số được khắc vào một mặt đĩa thành các hốc và mặt được sắp xếp thành rãnh theo quỹ đạo hình xoắn tròn ốc như trong đĩa hát. Các hốc và mặt được phủ bằng nhựa bảo vệ trong suốt. Cường độ tia laser phản xạ từ đĩa đang chuyển động sẽ có sự thay đổi khi chiếu qua các mép chuyển từ mặt sang hốc và từ hốc sang mặt. Sự thay đổi đó cho thông tin bằng 1, còn lại cho các bit thông tin bằng 0 như hình 6.18 chỉ ra. Hiện nay có 4 loại đĩa quang:

5.3.1.1. Đĩa CD-ROM (compact disk ROM)

Trong đó thông tin được lưu trữ ngay khi sản xuất đĩa tên các hốc và mặt như nói trên. Công nghệ khuôn đúc cho phép sản xuất hàng loạt các CD-ROM giống nhau với giá rẻ.

5.3.1.2. Đĩa WORM (Write Once, Read Many time)

Là đĩa quang ghi được 1 lần, đọc nhiều lần. Do vậy ngoài tia laser đọc ra, ổ đĩa này còn có một tia laser thứ hai gọi là tia viết có cường độ mạnh hơn nhiều để tạo ra một xung đốt cháy bề mặt đĩa trong quá trình ghi thông tin lên đĩa. Tùy thuộc vào cấu trúc và bề mặt có hai loại xử lý.

- Loại một, lớp phủ đĩa ở vùng bị đốt nóng sẽ bay hơi và làm lộ ra bề mặt của vùng đĩa dưới không còn lớp phủ. Hai vùng này sẽ có hệ số phản xạ ánh sáng khác nhau và mang thông tin bit khác nhau.

- Loại hai, ở cuối quá trình viết bằng xung laser này, lớp phủ đĩa bị chảy ra sẽ được làm lạnh rất nhanh và kết rắn ở lại ở dạng vô định hình.

Lớp vô định hình này sẽ có hệ số phản xạ khác với lớp phủ cũ và do vậy có hai vùng vật liệu mang thông tin bit khác nhau. Hiện nay, hay dùng lớp phủ là một loại chất Polymer có kí hiệu là CD-R.

5.3.1.3. Đĩa CD-RW

Cho phép ghi và đọc nhiều lần. Đĩa này gồm lớp mỏng cỡ 50 đến 100 nm phản xạ bằng vàng hay hợp kim mà mạ bạc như nhôm, lớp cách điện, lớp kim loại khi chiếu tia laser có cường độ ánh sáng mạnh lên nó: Trạng thái tinh thể cho phép phản xạ ánh sáng. Do vậy laser trong ổ đĩa có 3 mức công suất khác nhau: công suất cao dùng khi ghi thông tin khi tạo lớp vô định hình (không phản xạ ánh sáng), công suất vừa dùng để xóa thông tin khi các lớp tinh thể (phản xạ ánh sáng) và công suất thấp dùng để đọc thông tin.

5.3.1.4. Đĩa DVD (Digital video disk hay digital versatile disk)

Có dung lượng lớn và khả năng truy cập nhanh hơn các đĩa quang kể trên. Do đó nó thường được sử dụng trong các ứng dụng lưu trữ phim ảnh và các âm thanh chất lượng cao. Nguyên tắc cấu tạo của đĩa DVD cũng giống như các đĩa quang khác nhưng chỉ tiêu kĩ thuật khác nhiều: kích thước các hố nhỏ hơn, khoảng cách giữa các rãnh nhỏ hơn, vùng dữ liệu lớn hơn và mã hóa dữ liệu tiết kiệm hơn. Do vậy, cùng một kích thước như các đĩa CD thông thường với đường kính 4,75'' (120 mm), thay vì 680 MB một đĩa DVD có thể chứa tới 4,7 GB nhớ.

5.3.2. Đĩa quang-từ

Nguyên tắc hoạt động của loại đĩa này sử dụng cả hai hiệu ứng quang và từ, đó là sử dụng ảnh hưởng của từ trường lên sự phân cực của sóng ánh sáng.

Ánh sáng của tia laser là sóng điện từ được phân cực theo một hướng xác định. Khi sóng điện từ đi qua một vật liệu đặt trong từ trường, từ trường đó làm xoay trục của sóng và làm quay mặt phẳng phân cực của nó, đó là hiệu ứng Faraday mà chúng ta đã biết. Đĩa quang-từ sử dụng hiệu ứng đó để ghi nhận thông tin. Đĩa quang từ có một lớp phủ đồng nhất làm bằng vật liệu hợp kim lantan sắt từ. Khi tia laser phân cực đốt bề mặt đĩa, hướng phân cực được dùng làm kính phân tích (analyzer) cho hướng phân cực của tia phản xạ và cảm biến sau bộ lọc nhạy với cường độ ánh sáng đi qua bộ lọc phân cực đó.

Khi ghi một bit lên đĩa, xung laser ngắn và mạnh sẽ đốt bề mặt đĩa ở vùng cần thiết lên quá nhiệt độ Curie T_c làm cho từ độ của vùng phủ sắt từ đó bằng 0. Cùng lúc đó, một nam châm phát ra từ trường với hướng phụ thuộc vào giá trị của bit cần viết ("1" sẽ có hướng ngược với "0"). Hướng của từ trường này bây giờ xác định hướng của các domain từ trong vùng bị đốt nóng khi mà chúng được làm lạnh xuống dưới nhiệt độ Curie. Khi lớp phủ đó được đông rắn lại là quá trình ghi hoàn thiện.

Khi đọc một bit, tia laser sẽ quét bề mặt đĩa và hệ phân cực sẽ nhạy với hướng phân cực của tia phản xạ. Nếu tia laser chiếu trên vùng viết bit, hướng của mặt phẳng

phân cực tia phản xạ sẽ phụ thuộc vào hướng từ độ của lớp phủ bề mặt vùng đó. Ví dụ, hướng phân cực trái sẽ đại diện cho mức logic “0” và phân cực phải cho mức “1”. Trong trường hợp thứ nhất, ánh sáng không thể đi tới bộ lọc phân cực được và cảm biến cho tín hiệu điện tương ứng với giá trị 0; trong trường hợp sau, tia sáng sẽ đi qua bộ lọc phân cực tới cảm biến với cường độ sáng toàn bộ và cảm biến cho tín hiệu điện tương ứng với giá trị 1. Bằng cách làm nóng các điểm xác định với hướng phù hợp của từ trường được phát bởi nam châm, bit được viết có thể được xóa. Việc xử lý có thể làm đi làm lại nhiều lần như đĩa từ. Tuy nhiên, đến nay do giá thành của đĩa quang từ rất cao nên ít được sử dụng trong thị trường.

Chương 6.

HỆ THỐNG VÀO RA

6.1. Giới thiệu chung

Hệ thống vào ra: trao đổi thông tin giữa máy tính và thế giới bên ngoài, bao gồm:

- + Các môđun vào ra (mạch ghép nối IO): ghép nối giữa CPU và bộ nhớ với các TBNV.
- + Các thiết bị ngoại vi: mạch ghép nối vào ra tổ chức thành các cổng vào ra sao cho mỗi cổng có một địa chỉ xác định.
- + Địa chỉ hoá tách biệt (vào ra trực tiếp): không gian địa chỉ cổng độc lập với không gian địa chỉ bộ nhớ.
- + Địa chỉ hoá theo bản đồ bộ nhớ: không gian địa chỉ cổng vào ra nằm trong không gian địa chỉ bộ nhớ.

Các thiết bị vào ra thường kết nối với máy tính thông qua các cổng giao tiếp (Communication ports). Mỗi cổng giao tiếp được gán một địa chỉ và có tập tham số làm việc riêng.

Một số cổng giao tiếp thông dụng sau:

1. PS/2: kết nối chuột và bàn phím.
2. Cổng COM và LPT.
3. Cổng USB: cổng giao tiếp đa năng theo chuẩn USB
 - + USB 1.0: 12Mb/s
 - + USB 2.0: 480Mb/s (hiện tại)
 - + USB 3.0: 1.5Gb/s (tương lai).
4. Cổng IDE, SATA và E-SATA: ghép nối các loại ổ đĩa.
5. Cổng LAN: ghép nối mạng.
6. Cổng Audio: ghép nối âm thanh.
7. Cổng đọc các thẻ nhớ.
8. Cổng Firewire /IEEE 1394: ghép nối các loại ổ đĩa ngoài.
9. Cổng VGA/Video: ghép nối với màn hình.
10. Cổng DVI: ghép nối với màn hình số.

6.2. Một số thiết bị ngoại vi cơ bản

Các thiết bị ngoại vi (Peripheral devices) hay còn gọi là thiết bị vào ra, là các bộ phận của hệ thống máy tính có nhiệm vụ:

1. Tiếp nhận các thông tin từ thế giới bên ngoài đi vào máy tính.
2. Kết xuất các thông tin từ máy tính ra thế giới bên ngoài. Nhiệm vụ (1) được đảm bảo bởi nhóm các thiết bị vào (Input devices) và nhiệm vụ (2) được đảm bảo bởi nhóm các thiết bị ra (Output devices).

Các thiết bị vào gồm có: bàn phím, chuột, ổ đĩa (đọc thông tin), máy quét ảnh và máy đọc mã vạch. Các thiết bị ra gồm có: màn hình, máy in, ổ đĩa (ghi thông tin) và máy vẽ.

6.2.1. Bàn phím

6.2.1.1. Cấu tạo và nguyên tắc hoạt động

Bàn phím (Keyboard) là thiết bị vào chuẩn của máy tính do bàn phím có thể đảm nhiệm cả chức năng nhập dữ liệu và điều khiển máy tính. Bàn phím tiêu chuẩn có 101 phím: các phím ký tự (a-z), các phím số (0 - 9), các phím phép toán (+, -, *, /), các phím chức năng (F1 - F12), các phím điều khiển (Ctrl, Alt, Shift,...) và các phím di chuyển: Home, End, Page Up, Page Down, Up, Down, Left, Right,...

Bàn phím có một vi xử lý riêng như chip 8042 cho máy PC/XT và 8048 cho máy AT. Một ma trận quét (bao gồm các đường dẫn dọc và ngang) được dẫn tới chip này. Ở các điểm vắt chéo được đặt các công tắc trên đó là các phím nhấn như hình 6.4. Nếu một phím được nhấn, công tắc sẽ đóng mạch tiếp xúc điện giữa các đường dây hàng và cột vắt chéo nhau tương ứng và mạch xử lý của bàn phím sẽ xác định phím nào được nhấn. Thông tin tọa độ của phím đó sẽ được truyền qua bộ đệm tới mạch ghép nối bàn phím trên bản mạch chính về vi xử lý.

6.2.1.2. Thâm nhập trực tiếp bàn phím qua các cổng vào/ra

Bàn phím cũng là một thiết bị ngoại vi nên về nguyên tắc có thể thâm nhập nó qua các cổng vào/ra. Kể từ máy AT trở đi với bàn phím MF-II điều này có thể được thực hiện qua các cổng có địa chỉ 60h và 64h.

Các vùng đệm dùng để truyền các mã bàn phím từ bàn phím về vi xử lý và các lệnh mà vi xử lý truyền tới bàn phím. Các lệnh được truyền qua cổng 64h còn dữ liệu phải truyền qua cổng 60h.

Thanh ghi trạng thái sẽ cho biết byte được truyền là lệnh (cổng 64h) hay dữ liệu (cổng 60h) được mô tả qua bảng sau:

Bảng 6-1. Bảng mô tả các cổng

Cổng	Thanh ghi	Đọc (R), viết (W)
60h	Đệm bồi ra	R
60h	Đệm nối vào	W
64h	Thanh ghi điều khiển	W
64h	Thanh ghi trạng thái	R

Thanh ghi trạng thái xác định trạng thái hiện tại của bộ điều khiển bàn phím. Thanh ghi này là chỉ đọc (Read only) và định dạng từ trạng thái như sau:

7	6				2	1	0
PARE	TIM	AUXB	KEYL	C/D	X	INPB	OUTB

Trong đó:

PARE: Lỗi chặn lẻ của byte cuối cùng được vào từ bàn phím; 1: có lỗi chặn lẻ; 0: không có.

TIM: Lỗi quá thời gian (time – out); 1: có lỗi; 0: không có.

AUXB: Đệm ra cho thiết bị phụ (chỉ ở máy PS/2); 1: giữ số liệu cho thiết bị; 0: giữ số liệu cho bàn phím.

KEYL: Trạng thái khóa bàn phím; 1: không khóa; 0: khóa.

C/D: Lệnh/số liệu; 1: viết qua cổng 64h; 0: viết qua cổng 60h.

INPB: Trạng thái đệm vào; 1: số liệu CPU trong bộ đệm vào; 0: đệm vào rỗng

OUTB: TRạng thái đệm ra; 1: số liệu bộ điều khiển bàn phím trong bộ đệm ra; 0: đệm ra rỗng.

Thanh ghi điều khiển ở địa chỉ 064h là loại chỉ viết, mã lệnh của từ điều khiển được mô tả ở bảng sau:

Bảng 6-2. Mô tả mã lệnh của từ điều khiển

Mã	Lệnh
A7h	Cấm thiết bị phụ
A8h	Cho phép thiết bị phụ
A9h	Kiểm tra ghép nối tới thiết bị phụ
AAh	Tự kiểm tra
ABh	Kiểm tra ghép nối bàn phím
ADh	Cấm bàn phím
A Eh	Cho phép bàn phím
C0h	Đọc cổng vào
C1h	Đọc cổng vào ra (byte thấp)
C2h	Đọc cổng vào ra (byte cao)
D0h	Đọc cổng ra
D1h	Viết cổng ra
D2h	Viết đệm ra bàn phím
D3h	Viết đệm ra thiết bị phụ
D4h	Viết thiết bị phụ
E0h	Kiểm tra đọc cổng vào
F0h	Gửi một xung tới lỗi ra
FFh	Cổng

Máy tính có thể ra các lệnh điều khiển bàn phím qua bộ đệm ra có địa chỉ 60h. Dưới đây là tóm tắt các lệnh đó. Nhớ rằng đây là địa chỉ của cổng B chip PPI-8255A trong đó 2 bit thấp nhất.

Bảng 6-3. Mô tả các lệnh điều khiển bàn phím

Mã	Lệnh	Mô tả
EDh	ON/OFF LEDs	Bật / tắt các đèn LED của bàn phím
EEh	Echo	Trả về byte EEh
F0h	Đặt/nhận diện	Đặt một trong ba mã quét và nhận diện các mã quét
F2h	Nhận diện bàn phím	Nhận diện ACK = AT, ACK + abh + 41h = MFII
F3h	Đặt tốc độ lặp lại/trễ	Đặt tốc độ lặp lại và thời gian trễ của bàn phím
F4h	Enable	Cho phép bàn phím hoạt động
F5h	Chuẩn/không cho phép	Đặt giá trị chuẩn và cấm bàn phím
F6h	Chuẩn/cho phép	Đặt giá trị chuẩn và cho phép bàn phím
FEh	Resend	Bàn phím truyền kí tự cuối cùng một lần nữa tới PC
FFh	Reset	Khởi động bên trong bàn phím

6.2.1.3. *Tham nhập bàn phím qua BIOS*

BIOS viết các kí tự do việc nhấn các phím vào bộ đệm tạm thời được gọi là bộ đệm bàn phím. Bộ đệm có địa chỉ 40:1E, gồm 32 byte và do vậy kết thúc ở địa chỉ 40:3D. Mỗi kí tự được lưu trữ ở đây bằng 2 byte, byte cao là mã quét và mã thấp là mã ASCII. Như vậy, bộ đệm có thể lưu trữ tạm thời đến 16 kí tự. Chương trình xử lý ngắt sẽ xác định mã ASCII từ mã quét bằng bảng biến đổi và viết cả hai mã vào bộ đệm bàn phím.

Bộ đệm bàn phím được tổ chức như bộ đệm vòng (Ring buffer) và được quản lý bởi 2 con trỏ. Các giá trị con trỏ được lưu trữ trong vùng số liệu của BIOS ở địa chỉ 40:1A và 40:1C.

Ngắt 16h trong BIOS cung cấp 8 hàm cho bàn phím. Thường các hàm BIOS trả về một giá trị 0 của mã ASCII nếu phím điều khiển hoặc chức năng được nhấn, ví dụ phím dịch con trỏ, phím F1 hoặc phím HOME.

Ngoài các hoạt động kể trên BIOS còn dịch các tổ hợp phím nhấn. Có 5 hoạt động tạo cho BIOS thực hiện các nhiệm vụ đặc biệt như sau:

+ Alt_Sys_Req yêu cầu BIOS phát ra ngắt 15h với hàm 85h. Người sử dụng có thể tạo một chương trình phục vụ ngắt 15 của riêng mình khi các phím này được nhấn.

+ Ctrl_Alt_Del bắt máy tính khởi động nóng.

+ Shift_PrtSc (Prin Screen) cho phép ngắt 09h trong BIOS gọi ngắt mềm 05h gửi nội dung hiện tại của màn hình tới máy in.

+ Ctrl_Num Lock (Pause) ngừng hoạt động của chương trình tới khi một phím bất kì được nhấn.

+ Ctrl_Break yêu cầu BIOS phát ra ngắt mềm 1Bh và đặt bit 7 của byte ở 0040:0071 lên 1.

Người dùng cũng có thể nhập một mã ASCII từ 01h tới FFh bằng việc trong khi nhấn phím Alt đồng thời nhập mã kí tự ASCII từ các phím số ở phần bên phải bàn phím, Ngắt 16h cho phép người sử dụng xâm nhập bàn phím, bảng 6.4 sau khi chỉ ra chức năng của ngắt này.

Bảng 6-4. Các hàm của ngắt 16h

Chức năng	Các thanh ghi vào	Các thanh ghi ra
Nhấn một phím nhấn	AH = 00h	AH = mã quét AL = mã ASCII

Kiểm tra một phím nhấn	AH = 01h	Cờ Zêro ZF = 1: không có phím nhấn Cờ Zêro ZF = 0: có phím nhấn AH = mã quét AL = mã ASCII
------------------------	----------	---

6.2.2. Chuột

6.2.2.1. Cấu tạo và nguyên tắc hoạt động

Chuột (Mouse) là một trong các thiết bị vào của máy tính được sử dụng rộng rãi nhất. Chức năng chính của chuột là điều khiển. Thông qua các phần mềm, hình thức hiển thị của chuột được thể hiện rất đa dạng, từ hình mũi tên đơn giản, đến bàn tay, đồng hồ cát, theo các trạng thái làm việc của chương trình. Hiện nay, có rất nhiều loại chuột đang được sử dụng. Ngoài chuột bi (còn gọi là chuột cơ khí), còn có chuột quang, chuột laser, chuột cảm ứng và chuột không dây. Các phím bấm chuột cũng rất đa dạng: thông thường là loại 3 phím (trái, phải và cuộn); một số chuột có thể có thêm cả phím tiến (Forward) và phím lùi (Backward). Hiện nay có hai loại chuột phổ biến là chuột cơ và chuột quang

a. Chuột cơ (hay chuột bi)

Phần trung tâm của chuột cơ là một viên bi thép xoay dễ dàng khi chuột được dịch chuyển. Viên bi được phủ keo hoặc nhựa nhằm làm tăng ma sát cho sự truyền chuyển động tới hai thanh thép tròn nhỏ được đặt vuông góc với nhau. Các thanh này sẽ biến chuyển động của viên bi, tức là của chuột theo hướng trục X và trục Y thành sự quay tương ứng của hai đĩa gắn với chúng. Trên hai đĩa có các khe (hoặc lỗ) liên tục cho qua và chặn hai chùm tia sáng tới các sensor nhạy sáng để tạo ra các xung điện đi tới vi mạch điều khiển bên trong chuột. Số các xung này tỉ lệ với lượng chuyển động của chuột theo các hướng trục X và trục Y và số xung trên một giây biểu hiện tốc độ của chuyển động chuột.

Cụ thể, nguyên tắc hoạt động của chuột bi có thể tóm tắt như sau:

- Khi chuột di chuyển, viên bi chuột quay;
- Khi bi quay nó kéo theo 2 trục áp vào quay theo. Hai trục được gắn bánh xe răng cưa ở 1 đầu:

- Một trục dùng để phát hiện chuyển động theo phương đứng.
- Một trục dùng để phát hiện chuyển động theo phương ngang.
- Hai đi-ốt sinh tia hồng ngoại chiếu qua phần bánh răng cưa gắn trên các trục

kể trên:

- Khi bánh răng cưa quay, ánh sáng hồng ngoại chiếu qua sẽ bị ngắt quãng;
- Ở phía đối diện có 2 bộ cảm biến chuyển ánh sáng hồng ngoại sau bánh răng

cưa thành tín hiệu điện;

- Tín hiệu điện thu được phản ánh chuyển động của chuột được chuyển cho máy tính xử lý.

b. Chuột quang (Optical mouse)

Sử dụng hai cặp optron, mỗi cặp gồm một diode phát quang LED chiếu lên một bề mặt di chuột (Mouse pad) và một sensor nhạy ánh sáng phản xạ từ bề mặt. Thời gian đầu, chuột quang được thiết kế chỉ để sử dụng được với một bề mặt được chế đặc biệt bằng kim loại có in một lưới các đường xanh dương (hướng Y) và xám (hướng X). Mỗi cặp optron nhạy với một loại màu, do đó khi di chuyển chuột cũng cho được các xung điện ở các lối ra giống như các trường hợp trong chuột cơ. Gần đây, một chip xử lý ảnh đặc biệt được gắn trong chuột cho phép phát hiện chuyển động tương đối của nó trên các bề mặt không cần đặc biệt nữa. Hệ thống xử lý trong chuột cho phép xử lý

một chuỗi các khung ảnh bề mặt phản quang khi chuột di chuyển. Sự sai khác giữa các khung ảnh liên tiếp nhau được xử lý và cung cấp thông tin về chuyển động của chuột theo các hướng tọa độ. Ví dụ, chuột Agilent Technologies ADNS-2610 xử lý 1.512 khung trên một giây, mỗi khung là một ảnh vuông 18×18 điểm ảnh và mỗi điểm ảnh chứa 64 mức xám.

Ưu điểm của chuột quang là hầu như không phải bảo dưỡng do bị bẩn như thương gặp trong chuột cơ (chuột bi). Năm 2004 chuột laser xuất hiện, trong đó người ta thay nguồn diode phát quang LED bằng diode laser. Điều này cho phép tăng được chi tiết của các hình ảnh dưới bề mặt chuột.

Kèm theo chuột đến nay thường có hai phím nhấn với các chức năng được quy định. Ngoài ra các chuột còn có thêm một bánh xe lăn cho phép cuộn lên (Scroll-up), cuộn xuống các trang văn bản trong windows rất thuận tiện.

Nguyên tắc hoạt động của chuột quang có thể tóm tắt như sau:

- Một đi-ốt phát ánh sáng đỏ qua ống kính chiếu xuống mặt phẳng di chuột; ánh sáng phản xạ từ mặt phẳng di chuột quay ngược trở lại phía dưới chuột;
- Một camera đặt phía dưới chuột liên tục chụp ảnh của bề mặt di chuột nhờ ánh sáng phản xạ. Tốc độ chụp là khoảng 1500 ảnh/giây;
- IC điều khiển chuột sẽ phân tích và so sánh các ảnh kế nhau và qua đó phát hiện ra chuyển động chuột;
- Tín hiệu biểu diễn chuyển động chuột do IC điều khiển chuột sinh ra được chuyển cho máy tính xử lý.

Tương tự như chuột quang, chuột laser cũng sử dụng phương pháp chụp và phân tích ảnh bề mặt kế nhau để phát hiện chuyển động. Tuy nhiên, chuột laser sử dụng ánh sáng laser với tốc độ chụp ảnh lên đến 6000 ảnh/giây. Nhờ vậy, chuột laser thường có độ chính xác và độ nhạy cao hơn so với chuột quang.

6.2.2.2. Ghép nối chuột

Tín hiệu điện ở lối ra của chuột được chuyển tới máy tính qua một cáp nối, thường được cắm vào cổng COM. Đến nay cũng đã có các loại *chuột quang không dây* trong đó việc truyền tín hiệu được thực hiện bằng ánh sáng hồng ngoại hoặc bằng sóng vô tuyến. Trong loại dùng sóng hồng ngoại cần có nguồn thu phát hồng ngoại lắp đặt trên một cổng ở máy tính. Trong loại dùng sóng vô tuyến cần có một bộ thu phát riêng cắm vào một cổng USB trên máy tính. Các nguồn thu phát sóng này dùng để thông tin với bộ phận thu phát trong chuột qua không gian gần máy. Vì vậy các loại chuột này giảm được dây nối từ chuột tới máy tính. Khi dịch chuyển hoặc nhấn một phím chuột, vi mạch điều khiển chuột sẽ phát ra gói dữ liệu tới mạch ghép nối và mạch này sẽ phát ra một ngắt tới máy vi tính. Điều khiển ngắt này là một chương trình ghép nối quản lý chuột. Nó chặn ngắt lại để đọc dữ liệu và cập nhật các giá trị bên trong liên quan đến trạng thái hiện tại của bàn phím cũng như vị trí của chuột. Hơn nữa trình quản lý cung cấp một ghép nối mềm qua ngắt 33h của BIOS để dịch các giá trị bên trong này. Chương trình ghép nối chuột thực hiện việc nối con chuột với các chương trình ứng dụng. Chúng thường được cài đặt như các trình quản lý thiết bị (Driver) khi khởi động hệ thống hoặc như là một trình nội trú TSR. Con trỏ chuột cho biết vị trí tương đối của con chuột trên màn hình. Hình dạng của nó có thể thay đổi tùy theo chương trình điều khiển và ứng dụng. Trình điều khiển có trách nhiệm dịch chuyển con trỏ chuột trên màn hình. Con trỏ chuột có thể hiện trong chế độ đồ họa với *con trỏ chuột đồ họa* và *con trỏ cứng hoặc mềm* trong chế độ văn bản (text-mode). Con trỏ chuột cứng là một con trỏ thông thường chuyển động trên màn phù hợp với chuyển động chuột. Con trỏ chuột mềm lại có dạng là một kí tự do người lập trình tự chọn. Có thể định nghĩa loại

và dạng con trỏ chuột bằng hàm 09h và 0Ah của ngắt 33h. Kích thước con trỏ cứng phụ thuộc vào ma trận kí tự hiện tại và chế độ màn hình. Trong khi đó con trỏ mềm được tạo bằng 2 giá trị 16 bit là mặt nạ màn hình (Screen mask) và mặt nạ con trỏ (Cursor mask). Các giá trị này sẽ được kết hợp với từ video cũ (gồm 2 byte mã kí tự và thuộc tính) trong RAM màn hình để tạo thành một từ nhớ video mới cho con trỏ như công thức sau:

Từ video mới = (từ video cũ AND screen mask) XOR cursor mask

Trong chế độ đồ họa, cách hiện con trỏ chuột cũng tương tự. Ở đây, một điểm ảnh được gán bởi 1 hoặc nhiều bit. Do vậy phải biết được số bit trên một điểm ảnh. Hơn nữa kích thước con trỏ chuột ở đây luôn là 16 x 16 điểm ảnh.

Chuột hoạt động trên một tập các tọa độ chuột là trục X và trục Y. Bảng 6-5 chỉ ra các tọa độ tại các chế độ màn hình khác nhau. Trong chế độ văn bản tọa độ X cực đại là 640 và Y cực đại là 200. Trong chế độ đồ họa 640 × 480 các thông số này thay đổi là 640 × 480.

Bảng 6-5. Độ phân giải chuột trong các chế độ màn hình

Mode video	Card màn hình	Chế độ	Độ phân giải màn hình	Độ phân giải chuột
00h	CGA/EGA/VGA	Văn bản	80 × 25	640 × 200
0Dh	EGA/VGA	Đồ họa	320 × 200	340 × 200
12h	VGA	Đồ họa	640 × 480	640 × 480
25h	SVGA	Đồ họa	1024 × 768	1024 × 768

Nhìn vào bảng ta thấy độ phân giải chuột trong chế độ văn bản là 640 x 480 và độ phân giải màn hình là 80 kí tự trong một dòng và 25 dòng trên toàn màn hình (80 cột, 25 dòng). Như vậy tọa độ chuột lúc này phải được biến đổi thành 1 tọa độ hàng và cột bằng cách nhân tọa độ chuột X với 80/60 và tọa độ chuột Y với 25/200.

Hàm 09h cho phép định nghĩa con trỏ chuột về hình dạng và thuộc tính trong chế độ đồ họa. Các byte Screen và Sursor mask được giữ trong một bộ đệm. Con trỏ chuột luôn có 16×16 điểm ảnh và các điểm ảnh đó phải được phủ bởi các byte Screen mask và Cursor mask. Vậy tùy thuộc vào chế độ màn hình và số maud được sử dụng mà một bộ đệm đủ lớn cần thiết cho các bit mặt nạ (Mask) này.

6.2.3. Máy in (Printer)

6.2.3.1. Cấu tạo và nguyên tắc hoạt động

Máy in là thiết bị ra phổ biến dùng để kết xuất thông tin ra giấy. Qua quá trình phát triển, có nhiều loại máy in được sử dụng như máy in búa (Typewriter-derived printers), máy in kim (Dot-matrix printers), máy in laser (Laser printers), máy in phun mực (Inkjet printers), máy in màu (Colour printers) và các máy in đa chức năng (Multi-function printers).

6.2.3.2. Ghép nối máy in qua cổng LPT

Thường các máy in được ghép nối với máy tính qua các cổng song song LPT như đã nói ở trên. Mỗi cổng LPT có 3 thanh ghi có thể thâm nhập qua 3 địa chỉ cổng vào/ra gán cho chúng. Các thanh ghi này được dùng để gửi dữ liệu tới máy in, xác định trạng thái máy in và điều khiển hoạt động của máy in. Thanh ghi dữ liệu được thâm nhập vào địa chỉ vào/ra cơ sở (ví dụ: 378h). Những PC hiện nay đều cho phép những thanh ghi này có thể được sử dụng để viết và đọc số liệu (theo 2 hướng). Thanh ghi trạng thái có địa chỉ bằng địa chỉ cơ sở cộng 1 (ví dụ: 379h). Đây là thanh ghi chỉ đọc. Có thể đọc trực tiếp nó bằng các lệnh vào/ra hoặc dùng ngắt 17h, hàm 02h trong

BIOS. Thanh ghi điều khiển được thâm nhập ở địa chỉ cơ sở cộng 2 (ví dụ 37Ah) vào có thể dùng cho cả đọc lẫn viết.

Vì tốc độ truyền ký tự của cổng song song nhanh hơn tốc độ in của máy in nên thông tin ở đây phải có móc nối như đã trình bày.

Lập trình cho máy in có thể được thực hiện bằng ngôn ngữ bậc cao, bằng các hàm trong ngắt 21h của DOS và hàm 40h. Để dùng hàm này gửi số liệu đến cổng máy in khác ta phải dùng hàm 3Dh để mở cổng mong muốn (LPT2 hoặc LPT3) như một thiết bị rồi gọi ngắt 21h hàm 40h. Có thể in một ký tự đơn tới LPT1 bằng ngắt 21h hàm 05h.

Ngắt 17h trong BIOS so với 3 hàm có thể làm việc với bất kỳ cổng song song nào. Chức năng cùng các thanh ghi vào (cần nạp trước khi gọi ngắt) và các thanh ghi ra (chứa các byte trạng thái sau khi ngắt) như thể hiện trong bảng 6-6. Trạng thái của máy in được biết qua việc đọc thanh ghi trạng thái cổng. Mức thấp nhất của việc lập trình với cổng song song được thực hiện bằng việc viết hoặc đọc trực tiếp các cổng vào/ra liên quan đến cổng. Cách này cũng được dùng khi cần truyền số liệu hai hướng giữa hai máy tính qua các cổng song song.

Bảng 6-6. Các thanh ghi vào ra của cổng LPT

Chức năng	Các thanh ghi vào	Các thanh ghi ra
Khởi động cổng máy in	AH = 01h (hàm 01h) DX = Số máy in (00h÷02h)	AH: chứa trạng thái của máy in
Viết một ký tự tới máy in	AH = 00h (hàm 00h) AL = ký tự cần in DX = số máy in	AH: chứa trạng thái của máy in
Nhận trạng thái máy in	AH = 02h (hàm 02h) DX = số máy in	AH: chứa trạng thái của máy in

6.2.4. Màn hình

Màn hình (Monitor hoặc Screen) là thiết bị ra chuẩn có thể hiện thị thông tin dưới dạng văn bản hoặc hình ảnh. Cùng với bàn phím và chuột, màn hình là thiết bị không thể thiếu đối với máy tính. Có ba dạng màn hình được sử dụng thông dụng: màn hình ống tia điện tử CRT (Cathode ray tube), màn hình tinh thể lỏng LCD (Liquid crystal display) và màn hình Plasma. Trong giáo trình này chỉ đề cập đến hai loại màn hình được sử dụng phổ biến cho máy tính là màn hình CRT và LCD.

6.2.4.1. Màn hình quét màn hình CRT

a. Cấu tạo và nguyên tắc hoạt động

Màn hình CRT (Cathode Ray Tube) sử dụng tia điện tử phát ra từ cực Cathode bắn lên mặt huỳnh quang phát pho để tạo ảnh. Tia điện tử được điều khiển bởi 2 cuộn lái tia (dòng và màn) để quét hết cả màn hình, đảm bảo tốc độ quét tối thiểu là 24 màn hình/giây. Tín hiệu hình ảnh (video) được sử dụng để điều khiển mật độ tia điện tử bắn lên màn huỳnh quang tạo các mức sáng/tối khác nhau. Màn hình đen trắng sử dụng 1 súng điện tử, còn màn hình màu sử dụng 3 súng điện tử ứng với 3 màu cơ bản Đỏ (Red), Xanh lá cây (Green) và Xanh da trời (Blue). Ba màu này được trộn với nhau theo tỷ lệ khác nhau tạo thành tất cả các màu có trong tự nhiên cho điểm ảnh.

Phương pháp phổ biến nhất để tạo ra hình ảnh trên màn hình CRT là quét đi quét lại một chùm tia điện tử ngang qua màn hình từ trái sang phải bằng từ trường biến thiên được sinh ra do các cuộn lái tia như hình 6-10 chỉ ra.

Các điện tử phát xạ từ cathode được hội tụ thành một chùm tia, sau đó được tăng tốc và được làm lệch hướng chuyển động bởi các *bộ phận lái tia*. Tia này sẽ đập vào hình có phủ chất huỳnh quang để tạo thành một điểm sáng là một điểm ảnh. Do hiện tượng lưu ảnh trong võng mạc của mắt người nên khi tia điện tử được quét rất nhanh theo chiều ngang từ trái sang phải sẽ tạo nên một vệt sáng ngang được gọi là *dòng quét*. Đến cuối một dòng, nó được quét ngược trở về bên trái để quét tiếp dòng thứ hai bên dưới,... Quá trình quét các dòng được dịch dần từ trên xuống dưới cho suốt chiều dọc của màn hình được gọi là *quét dọc* để tạo lên một khung ảnh gọi là *màn*. Các màn được tạo ra nhiều lần trong một giây. Tốc độ lặp lại này gọi là *tốc độ khung*, *tốc độ quét dọc*, hay *tốc độ làm tươi*.

Tốc độ làm tươi là một đặc tính rất quan trọng của hệ thống hiển thị màn hình. Sau khi tia điện tử đi qua điểm phát sáng, cường độ phát quang giảm dần, tốc độ giảm đi được gọi là *độ lưu ảnh*. Độ lưu ảnh cao có nghĩa là phải mất thời gian dài để cường độ phát sáng giảm hết. Nếu cường độ sáng giảm đi quá chậm, những hình ảnh chuyển động trên màn sẽ bị nhòe đi. Nếu quá nhanh tốc độ làm tươi phải đủ cao nếu không sẽ bị hiện tượng nhấp nháy. Các card màn hình này thường làm tươi với tốc độ $50 \div 60$ Hz. Hiện nay có khuynh hướng sử dụng các màn huỳnh quang có độ lưu ảnh thấp nên tốc độ làm tươi thường trên 70 Hz.

Một điểm phát sáng *màu tự nhiên* trên màn hình được thực hiện nhờ sự trộn lẫn của 3 màu đỏ (red), xanh lơ (blue), xanh lá cây (green) theo một tỷ lệ nào đó. 3 màu này được tạo ra nhờ 3 tia điện tử cùng được bắn vào 3 điểm kề cận nhau trên màn hình, mỗi điểm được phủ một chất huỳnh quang phát ra các màu tương ứng. 3 chùm tia điện tử đó được phát ra bởi 3 súng điện tử là 3 cathode được xếp đặt bên trong ống hình CRT.

Độ giãn cách điểm ảnh gọi là *dot pitch*, là khoảng cách từ tâm điểm ảnh này đến tâm điểm ảnh kề cận. Thường màn hình các máy vi tính có các dot pitch từ 0.39 mm đến 0.22 mm. Nếu biết được dot pitch và kích thước màn hình có thể tính được *độ phân giải cực đại* của màn hình tính theo pixel. Thường tỉ số chiều cao trên chiều rộng của màn hình máy tính là $3/4 = 0,75$. Hiện nay các màn hình máy tính 15 inch (bằng kích thước đường chéo) với dot pitch 0.28 mm, có thể hiển thị tốt ở độ phân giải 1024×768 điểm ảnh. Có 2 chế độ quét tia điện tử:

- + Quét xen kẽ (Interlaced), trong đó các dòng lẻ được quét trước cho đến hết màn hình theo chiều dọc tạo lên màn hình lẻ, sau đó các dòng chẵn tạo lên màn hình chẵn được quét sau. Phương pháp này có ưu điểm là giảm được tốc độ truyền dữ liệu xuống 2 lần nhưng khi quan sát những hình ảnh tĩnh thường có trong máy tính sẽ bị cảm giác nhấp nháy.

- + Quét không xen kẽ (No - interlaced), trong đó các dòng quét được thực hiện tuần tự dòng nọ sau dòng kia. Ưu điểm là hình là hình ảnh có thể được điều chỉnh chính xác và ổn định nhưng thiết kế mạch điện khó hơn vì phải giải quyết vấn đề tăng rải tần làm việc. Hiện nay tất cả các màn hình đều dùng chế độ không xen kẽ.

b. Hiển thị các điểm ảnh trên màn hình CRT

Các ký tự cũng như hình ảnh được hiển thị trên màn hình CRT theo dạng các chấm tối hoặc sáng. Các chấm này được tạo thành khi chùm tia điện tử đập hay không đập vào màn hình trong suốt quá trình quét. Hình 6-11a biểu diễn cách hiển thị các chữ P và H bởi các điểm ảnh.

Các chấm tròn đen biểu diễn các điểm ảnh mà tại đó chùm tia đập vào và làm điểm này phát sáng còn các chấm tròn trắng biểu diễn các điểm ảnh tại đó chùm tia không đập vào màn hình, không phát sáng. Trong hình vẽ, ta có thể thấy mỗi ký tự là

một ma trận điểm có kích cỡ rộng 5 điểm và cao 7 điểm. Các kích thước thường dùng khác cho các ma trận điểm biểu diễn ký tự là 7×9 , 7×12 và 9×14 .

Vị trí các điểm ảnh của mỗi ký tự theo từng dòng quét được nhớ trong một bộ nhớ ROM được gọi là ROM ký tự. Với trường hợp này, bộ nhớ ROM sử dụng ma trận 5×7 cho mỗi ký tự như hình 6-11b. các bit nhớ có giá trị 1 sẽ ứng với điều khiển tia điện tử đập vào màn hình phát sáng còn các bit nhớ có giá trị 0 thì điều khiển tia điện tử không đập vào màn hình, nghĩa là không phát sáng.

c. Bản mạch ghép nối màn hình (Card màn hình)

Bản mạch ghép nối màn hình (Card màn hình) có nhiệm vụ chuyển tải các thông tin từ vi xử lý ra màn hình. Nó thường được cắm trên một khe cắm mở rộng của máy vi tính, đầu ra được nối tới màn hình. Hiện nay cũng có nhiều bản mạch chính có các mạch điện ghép nối màn hình được tích hợp ngay trong các chipset lắp đặt trên nó mà không cần bản mạch lắp rời nữa.

Phần trung tâm của nó là *chíp điều khiển ống hình CRTC* (Cathode ray tube controller). Vi xử lý thâm nhập *RAM video qua mạch ghép nối bus* để viết thông tin xác định ký tự hoặc hình vẽ cần hiển thị. CRTC liên tục phát ra các địa chỉ để RAM video đọc các ký tự trong đó và truyền chúng tới *máy phát ký tự*.

Trong chế độ văn bản, các ký tự được xác định bởi mã ASCII trong đó có cả các thông tin về *thuộc tính* (attribute) của ký tự, thí dụ thứ tự được hiện theo cách nhập nháy hay đảo màu đen trắng. *ROM ký tự* lưu trữ các hình mẫu điểm ảnh của các ký tự tương ứng để máy phát ký tự biến đổi các mã ký tự đó thành một chuỗi các bit điểm ảnh và chuyển chúng tới một *thanh ghi dịch*. Sử dụng các bit điểm ảnh này cùng với các thông tin thuộc tính từ RAM video và các tín hiệu đồng bộ từ CRTC, máy phát tín hiệu sẽ phát ra các tín hiệu cần thiết đi tới màn hình.

Trong chế độ đồ họa, thông tin trong RAM video được sử dụng trực tiếp cho việc phát ra các ký tự. Lúc này các thông tin về thuộc tính không cần nữa. Chỉ từ các giá trị bit trong thanh ghi dịch, máy phát tín hiệu sẽ phát các tín hiệu về độ sáng và màu cho màn hình.

6.2.4.2. Các chuẩn màn hình thông dụng và bộ xử lý đồ họa

a. Các card màn hình

Có rất nhiều phần cứng và chế độ hoạt động cho màn hình và Card màn hình máy tính. Những máy vi tính đầu tiên thường dùng hai loại mạch ghép nối màn hình đó là: Card hiển thị đơn sắc MDA (Monochrome Display Adapter) và card hiển thị màu đồ họa CGA (Color Graphics Adapter). Card MDA chỉ hỗ trợ chế độ văn bản 80 cột, 25 dòng với độ phân giải 720×200 điểm ảnh và. Các ký tự lưu trữ trong ROM với ma trận 9×14 điểm ảnh. Card CGA hỗ trợ cả văn bản và đồ họa 16 màu với 160×200 điểm ảnh, 4 màu từ một bảng 16 màu với 320×200 điểm ảnh và 2 trong 16 màu với 640×200 điểm ảnh. Tất cả ký tự trong các chế độ tiêu dùng ma trận 8×8 điểm ảnh trong ROM ký tự. Mạch MDA dùng tốc độ làm tươi 50Hz quét không xen kẽ, còn CGA là 60 Hz. Tiếp đó là sự cải tiến mạch MDA với sự ra đời của Card hercules với độ phân giải 720×348 điểm ảnh đơn sắc.

Cùng với máy PC/AT, hãng IBM cho ra đời card màn hình EGA (Enhanced Graphics Adapter). Ngoài việc hỗ trợ các chế độ hiển thị cũ, mạch còn có tốc độ làm tươi cao hơn với 64 màu và độ phân giải 640×350 điểm ảnh. Các loại card màn hình trên đều có tín hiệu lỗi ra là số, như vậy chức năng chuyển đổi số/tương tự được đặt trong màn hình.

Tiếp theo IBM giới thiệu mạch ghép nối VGA (Video Graphics Array). Chuẩn này tiếp tục hỗ trợ các chuẩn trước, nhưng có tốc độ quét lên tới 32 kHz và tốc độ làm

tươi 60 Hz với 256 màu trong bảng 262.144 màu. Nó có các chế độ 640×400 và 640×480 điểm ảnh. Đặc điểm khác của card này là không giao tiếp với màn hình qua các tín hiệu số nữa mà qua các tín hiệu tương tự, đó là các tín hiệu thành phần R (đỏ), xanh lá cây (G) và xanh dương (B).

Hãng máy tính IBM còn đưa ra các mạch ghép nối đồ họa khác là MCGA (Multi-Color Graphics Adapter), XGA (Extended Graphics Array) và PGA (Professional Graphics Array) nhưng ít được sử dụng. Hiệp hội VESA cũng đưa ra chuẩn SVGA (Super VGA) hỗ trợ chế độ 800×600 điểm ảnh 16 màu với tốc độ làm tươi 56 kHz. Các chế độ hiển thị ở đây lên tới 1280×1024 điểm ảnh với độ sâu màu lên tới 24 bit (True color) và tốc độ làm tươi lên tới 75 Hz. Khi các ứng dụng bắt đầu dùng những độ phân giải cao mới và độ sâu màu gia tăng, thì hiệu năng của máy tính trong các chương trình đồ họa bị sụt giảm do các card trên bị hạn chế. Do vậy các nhà sản xuất đã giải quyết vấn đề này bằng cách thêm vào phần cứng gia tốc cho các chip SVGA. Do đó xuất hiện các card màn hình AVGA (Accelerated VGA).

Các nhà sản xuất trình điều khiển thiết bị (Driver) cho phép bản mạch giao tiếp với trình ứng dụng trong windows và tự động chuyển đổi các lệnh đồ họa của windows thành những lệnh riêng biệt dành cho phần cứng gia tốc trong chip AVGA.

Giáo trình này chỉ liệt kê hai loại card màn hình là EGA và VGA chuẩn.

Card EGA

Bản mạch này cho độ phân giải tối đa là 640×350 điểm ảnh với 16 màu. Đặc điểm như sau:

- Có thể định nghĩa và nạp ma trận ký tự của riêng người dùng.
- Có 2 tập ký tự, tức là có thể hiện 512 thay vì 256 ký tự. Như vậy, một mã 9 bit bao gồm mã ký tự 8 bit và 1 bit nhấp nháy được sử dụng ở đây (đó là bit thứ 3 trong byte thuộc tính - hàm 10h, hàm phụ 03h của INT 10h).
- Màn hình có thể chia bởi phần cứng thành 2 phần tách biệt. EGA hiện các thông tin trên chúng từ 2 vùng RAM video hoàn toàn tách biệt nhau.
- RAM video được phân thành vài lớp ảnh, các lớp này có thể được chạy hay bị khóa hay không bằng việc thâm nhập các thanh ghi.

RAM video được chia thành 4 lớp nhớ song song. Các lớp này được khởi phát ở địa chỉ A0000h và chúng chồng lên nhau. Một thâm nhập viết tới địa chỉ byte A0000h sẽ truyền đồng thời giá trị byte vào cả 4 lớp, có thể cho phép lớp nào được đọc hoặc viết.

Card VGA

Hiện nay, các máy vi tính tương thích IBM PC chủ yếu dùng các bản mạch VGA hoặc SVGA với các vi mạch tích hợp cỡ lớn được lắp trên đó. Chuẩn VGA với bộ điều khiển màn hình CRT cho phép lựa chọn tần số nhịp đồng hồ, số dòng quét ngang, tốc độ làm tươi, lượng quét quá,..., có thể lập trình cho một hệ VGA hoạt động theo các chế độ CGA, EGA hoặc một số chế độ khác. Nó có đặc điểm là có độ phân giải 640×480 điểm ảnh, độ sâu màu 18 bit với $2^{18} = 262.144$ màu, có thể hiện đồng thời 256 màu. Dĩ nhiên card không thể phát tín hiệu màu số tới màn hình vì phải cần đến 18 đường tín hiệu số. Do vậy, card VGA phát ra tín hiệu Analog để điều khiển màn hình gọi là *tín hiệu video màu tổng hợp* (Composite video color monitor) chứa mọi tín hiệu thành phần và chỉ truyền trên một đường dây cho một thành phần đỏ, xanh lá cây hay xanh dương. VGA được trang bị bộ nhớ màn hình chuẩn là 256 kb, những loại mới hơn thậm chí có tới 1Mbyte nhớ và có độ phân giải tới 1024×768 điểm ảnh. Sau đây là các thông số của card VGA chuẩn.

Các chế độ CGA, EGA cũng như MDA hoàn toàn có thể chạy trên card VGA. Tất cả các thanh ghi điều khiển cũng có thể được đọc để cho CPU có thể xác định và lưu trữ được chế độ màn hình hiện tại và các thông số liên quan. BIOS của VGA được tăng cường thêm hàm 1Ch dùng để back-up các trạng thái của video. Điều này là quan trọng trong hoạt động đa nhiệm vì chương trình có thể thường xuyên biến đổi mode video.

b. Bộ xử lý đồ họa

Trong nhiều ứng dụng các phần mềm đồ họa như CAD (Computer Added Design), CAE (Computer Added Enginnering) hoặc EDA (Electronic Design Automasion) độ phân giải cần phải có 1024×768 hoặc hơn nữa. Điều này liên quan tới quá trình làm tươi màn hình và làm tươi bộ nhớ RAM. Thêm nữa các chi tiết hình ảnh phức tạp sẽ đòi hỏi nhiều phép tính hơn trong quá trình vẽ các điểm ảnh. Trong những trường hợp này có thể dùng các bản mạch ghép nối đồ họa đặc biệt trong đó có các bộ xử lý đồ họa như TMS34020 của hãng Texas Instruments để tạo ra các hình ảnh có độ phân giải 1024×1024 điểm ảnh 8 bit.

Ở đây VRAM được dùng cho bộ đệm video vì nó có khả năng nhận số liệu ra đủ nhanh để làm tươi điểm ảnh. Các tín hiệu lối ra VRAM được nối tới các thanh ghi bảng màu và lối ra của các thanh ghi bảng màu được đưa vào các bộ biến đổi D/A để tạo ra các tín hiệu RGB Analog cấp cho màn hình. Các DRAM và ROM trong bản mạch được dùng để giữ các chương trình và dữ liệu cho bộ xử lý đồ họa. Những lệnh đồ họa cơ bản có sẵn trong chip như vẽ đường thẳng, chuyển điểm ảnh, tô màu một vùng,..., được phát triển thành các thư viện cho phép lập trình dễ dàng các chương trình thực hiện vẽ rất nhanh các hình phức tạp.

Hầu hết các máy vi tính hiện nay đều được trang bị một khối có chức năng điều khiển hiển thị AVGA. Một mạch AVGA có thể cho độ phân giải lên tới 1200×1600 điểm ảnh và độ sâu màu 24 bit (True color).

6.2.4.2. Màn hình LCD

Màn hình LCD (Liquid Crystal Display) là màn hình tạo ảnh dựa trên sự linh động của các “*tinh thể lỏng*” (Liquid Crystals). Tinh thể lỏng là các chất bán rắn lỏng rất nhạy cảm với nhiệt độ và dòng điện. So với màn hình CRT, màn hình LCD mỏng hơn, nhẹ hơn và tiêu thụ ít điện năng hơn. Ngoài ra, phần diện tích màn hình thực để hiển thị ảnh (Viewable size) của LCD cũng lớn hơn. Chẳng hạn màn hình LCD 15” có phần màn hình thực tương đương màn hình CRT 17”. Nhược điểm của LCD so với CRT là không hỗ trợ nhiều độ phân giải, chất lượng ảnh không cao, thời gian đáp ứng (response time) lớn và góc nhìn (View angle) nhỏ. Có thể phân loại màn hình LCD thành 2 loại theo nguồn phát sáng:

- + LCD chiếu sau (backlit).
- + LCD phản xạ (reflective).

LCD chiếu sau sử dụng nguồn sáng riêng đặt ở phía sau, thường dùng trong các LCD có công suất lớn, như màn hình máy tính và màn hình tivi. LCD phản xạ sử dụng ánh sáng phản xạ của nguồn sáng từ bên ngoài. LCD phản xạ có thiết kế đơn giản, rẻ tiền, thường thích hợp với các màn hình có công suất nhỏ, như màn hình đồng hồ, màn hình máy tính tay.

Dưới tác dụng của trường tĩnh điện, một số chất có những phân tử có hình thon dài cho phép định hình thành những thanh mảnh ở *trạng thái tinh thể lỏng*. Trạng thái này được tồn tại ở một nhiệt độ cao hơn nhiệt độ nóng chảy của tinh thể chất một chút. Lúc này chất có tính phân cực quang, tức là chỉ có sóng ánh sáng phân cực (vectơ điện trường chỉ dao động trong một mặt phẳng) truyền qua khi mặt phẳng phân cực

của ánh sáng trùng với mặt phẳng phân cực của chất. Phương của mặt phân cực của tinh thể lỏng có thể điều khiển được bằng điện trường áp đặt lên nó.

Để tạo ra màn hình LCD màu, người ta dùng các nguồn sáng huỳnh quang mạnh phía sau truyền qua các tinh thể lỏng và một số bộ lọc màu để tạo ra các hình ảnh có màu sắc mong muốn.

Vì màn hình LCD không có quét tia điện tử, nên thay vào các điểm ảnh riêng biệt là các phần tử phát sáng được định địa chỉ một cách tuần tự. Do vậy, trên các màn hình này, hình ảnh cũng được phát ra từng dòng một. Quá trình quét ngược cũng không còn nữa vì ở đây đơn giản chỉ việc thay đổi địa chỉ về phần tử đầu dòng tiếp theo.

Việc lập trình cho bản mạch ghép nối màn hình cũng có thể thực hiện bằng việc thâm nhập trực tiếp tới các thanh ghi điều khiển CRT, các thanh ghi bảng màu, bộ đếm khung,... Cũng có thể thâm nhập qua ngắt 10h trong BIOS nhưng sẽ đạt được tốc độ chậm hơn. Và tất nhiên có thể sử dụng các hàm đồ họa trong các ngôn ngữ bậc cao

Dựa trên phương pháp điều khiển các tinh thể lỏng, ta có 2 loại LCD:

+ LCD ma trận thụ động (Passive matrix).

+ LCD ma trận chủ động (Active matrix).

LCD ma trận thụ động sử dụng lưới hoặc ma trận để định nghĩa từng điểm ảnh (pixel) bởi hàng và cột của nó. Một điểm ảnh (giao giữa 1 hàng và 1 cột) được kích hoạt khi điện áp được đặt vào cột và dòng tương ứng được nối đất.

Ngược lại, LCD ma trận chủ động sử dụng một TFT (Thin Film Transistor) để điều khiển một phần tử tinh thể lỏng. Các TFT hoạt động tương tự như các bộ chuyển mạch.

- TFT LCD là thiết bị được điều khiển bằng các tín hiệu điện;
- Lớp tinh thể lỏng nằm giữa 2 lớp trong suốt chứa các điện cực ITO (Indium Tin Oxide);
- Các phần tử tinh thể lỏng được sắp đặt theo các hướng khác nhau theo sự thay đổi điện áp đặt vào các điện cực ITO;
- Hướng của các phần tử tinh thể lỏng trực tiếp ảnh hưởng đến cường độ ánh sáng đi qua và nó gián tiếp điều khiển mức sáng / tối (còn gọi là mức xám) của ảnh hiện thị;
- Màu của hình ảnh được tạo bởi một lớp lọc màu;
- Mức xám của các điểm ảnh được thiết lập theo mức điện áp của tín hiệu video đưa vào điện cực điều khiển.

Chương 7

GHÉP NỐI MÁY TÍNH VỚI CÁC THIẾT BỊ NGOẠI VI

7.1. Các phương pháp định địa chỉ thiết bị ngoại vi

Ta biết rằng giống như các ô nhớ, mỗi thiết bị được gán cho một địa chỉ để vi xử lý có thể định vị khi cần trao đổi thông tin với nó. Trong hệ thống ghép nối máy tính, có thể định địa chỉ cho các thiết bị ngoại vi theo hai phương pháp sau:

7.1.1. Vào ra ánh xạ bộ nhớ (Memory mapped I/O)

Thiết bị ngoại vi được định địa chỉ trong cùng một không gian địa chỉ với bộ nhớ chính. Lúc này các thiết bị ngoại vi sẽ chiếm một vùng nào đó trong không gian bộ nhớ, phần còn lại dành cho các vi mạch nhớ. Ví dụ, các bản mạch ghép nối màn

hình thường được định địa chỉ trong không gian bộ nhớ như các card EGA, VGA,... có địa chỉ trong dải từ A0000h đến C7FFFh.

7.1.2. Vào ra tách biệt (Isolated I/O)

Các thiết bị ngoại vi được định địa chỉ trong một không gian riêng được gọi là không gian vào ra rộng 64 KB. Các địa chỉ trong không gian này được gọi là cổng (port). Đây là phương pháp được dùng khá phổ biến hiện nay cho hầu hết các thiết bị ngoại vi.

7.1.3. Quá trình móc nối thông tin

Dữ liệu được máy tính nhập vào hoặc xuất ra theo chương trình. Khi máy tính muốn gửi dữ liệu ra ngoại vi, một lệnh trong chương trình sẽ viết dữ liệu tới cổng ra (Out port) nhằm thông tin với thiết bị ngoại vi. Khi máy tính muốn nhận dữ liệu từ ngoại vi, một lệnh đặt dữ liệu từ cổng vào (In port).

Thuật ngữ cổng (Port) biểu thị một đầu nối vật lý ghép từ máy tính đến thiết bị ngoại vi. Ví dụ, cổng COM phổ biến trong máy tính là một đầu nối 9 chân thường đặt ở phía sau máy vi tính. Nó còn biểu thị các khâu nối trên các bus tới các ngoại vi, lúc đó các cổng vào/ra này được hiểu là các mạch điện. Cổng thường có các *thanh ghi* dùng để chứa các từ dữ liệu tạm thời gọi là các *thanh ghi đệm số liệu* mà bộ xử lý viết tới hoặc đọc từ cổng, các *thanh ghi điều khiển* cho phép bộ xử lý viết các từ điều khiển tới đó để điều khiển thiết bị ngoại vi và các *thanh ghi trạng thái* lưu trữ từ chỉ trạng thái hiện tại của cổng.

Quá trình phát và thu các *tín hiệu điều khiển* trong trao đổi dữ liệu giữa các bên đối tác thông tin gọi là quá trình *móc nối thông tin* (Handshaking). Các tín hiệu điều khiển đó gọi là các *tín hiệu móc nối* hay các tín hiệu bắt tay. Việc trao đổi thông tin giữa các đối tác thông tin (vi xử lý, bộ nhớ và các thiết bị ngoại vi) được thực hiện theo một trong hai cách:

7.1.3.1. Thông tin không có móc nối

Các bên thông tin với nhau (đọc dữ liệu từ bên kia hoặc viết dữ liệu tới bên kia) một cách tùy ý mà không cần biết đối tác có sẵn sàng hoặc đã ghi nhận hay không. Ví dụ, trong hệ đo nhiệt độ ghép nối máy vi tính: điện áp ra trên cặp nhiệt điện được khuếch đại lên, tiếp đó được biến đổi thành tín hiệu số rồi được đọc vào máy vi tính. Rõ ràng tín hiệu này lúc nào cũng sẵn sàng cho vi xử lý đọc, vì nhiệt độ là một đại lượng biến đổi rất chậm so với tốc độ đọc dữ liệu của máy vi tính. Cũng vậy, nếu dùng máy vi tính điều khiển bộ hiển thị số của các giá trị nhiệt độ kể trên thì bất cứ lúc nào vi xử lý cũng có thể đưa tín hiệu hiển thị ra được. Đây gọi là các quá trình *vào/ra không có móc nối* hay vào/ra không bắt tay.

7.1.3.2. Thông tin có móc nối

Các bên thông tin chỉ trao đổi dữ liệu khi có tín hiệu móc nối sẵn sàng READY hoặc tín hiệu ghi nhận ACK từ bên kia. Phương thức này dùng khi tốc độ xử lý dữ liệu của các đối tác thông tin không đáp ứng được ngay lập tức khả năng của bên kia. Ví dụ như bên phát dữ liệu có tốc độ phát nhanh hơn tốc độ xử lý thông tin bên thu thì không thể phát liên tục các byte số liệu được. Đó là trường hợp như đã nói trong chu kỳ bus viết bộ nhớ, do tốc độ truy cập bộ nhớ chậm so với khả năng truyền dữ liệu của bộ xử lý thì bộ xử lý phải trải qua các trạng thái đợi tín hiệu sẵn sàng từ bộ nhớ ms truyền tiếp các byte sau. Các tín hiệu đó gọi là tín hiệu điều khiển móc nối. Đây là quá trình vào/ra có móc nối hay vào/ra có bắt tay.

7.1.4. Ghép nối thiết bị ngoại vi

7.1.4.1. Ghép nối song song

Truyền các bit dữ liệu được truyền song song trong cùng một thời điểm trên nhiều đường dây.

Tốc độ truyền cao

Cần đường truyền song song để tải các bit dữ liệu cùng đi, điều đó khiến phương pháp này tốn kém về dây dẫn.

Điện hình của phương pháp này công máy in 25 chân LPT.

7.1.4.2. Ghép nối nối tiếp

Từng Bit của dữ liệu lần lượt được gửi đi trên một đường truyền duy nhất.

Dữ liệu trong máy tính thương ở dạng 8 bit, 16 bit, vì thế cần có bộ chuyển đổi từ song song sang nối tiếp.

Tốc độ truyền của phương pháp này chậm vì truyền từng bit trên một đường dây.

Ưu điểm là kinh tế, không tốn dây, có thể dùng để truyền đi xa.

Các cấu hình ghép nối:

- Điểm - điểm (point to point): Qua một cổng vào ra chỉ có thể ghép một thiết bị ngoại vi (PS/2, COM, LPT,...)

- Điểm - đa điểm (Point to multipoint): Thông qua một cổng vào ra ghép nhiều thiết bị vào ra. Ví dụ như: SCSI (7,15), USB (127),...

7.2. Các cổng ghép nối vào/ra đa năng

Các máy vi tính được trang bị 2 loại cổng cho phép nối các thiết bị bên ngoài là *cổng song song* và *cổng nối tiếp*. Các cổng này được dùng cho nhiều mục đích. Cổng được gọi là song song vì nó cho phép truyền số liệu đồng thời từ các bit của 1 dữ liệu trong 1 nhịp truyền. Như vậy nếu từ dữ liệu dài 8 bit thì ngoài dây chung (dây đất) ra, cổng sẽ phải có 8 dây dữ liệu cho 8 bit thông tin. Trong khi đó, cổng nối tiếp chỉ cho phép truyền lần lượt mỗi nhịp 1 bit trong từ dữ liệu và như vậy cổng chỉ cần 1 dây truyền dữ liệu như lại cần ít nhất là 8 nhịp truyền cho một từ dữ liệu như thí dụ vừa nêu.

7.2.1. Cổng song song LPT

Cổng song song LPT trong máy vi tính có đầu nối loại D-25 theo chuẩn Centronics. Nó cho phép dùng cho cả phát và nhận số liệu, do đó có thể thiết kế các thiết bị ngoại vi như ổ đĩa cứng mang xách được, máy quét hình ghép nối với máy tính qua cổng này. Tuy vậy cổng song song được dùng chủ yếu cho việc ghép nối với máy in và do vậy nó cũng có tên là LPT (Line PrinTer). Hiện nay thường có 1 cổng LPT được đặt ở sau máy. Cổng có các dây tín hiệu nối tới đầu chip super I/O hoặc chipset lắp trên bản mạch chính.

Sơ đồ khối của mạch ghép nối song song như hình 7-4. có 3 thanh ghi dùng để truyền số liệu vào điều khiển ngoại vi trên cổng. Đó là các thanh ghi số liệu 2 hướng, thanh ghi trạng thái và thanh ghi điều khiển. Địa chỉ cơ sở của các thanh ghi được lưu giữ trong các vùng số liệu BIOS trong đoạn 040h. Thanh ghi số liệu có địa chỉ Offset bằng 00h, thanh ghi trạng thái là 01h và thanh ghi điều khiển là 02h.

Nói chung địa chỉ cơ sở của LPT1 là 378h và của LPT2 là 278h. Như vậy địa chỉ thanh ghi số liệu trong cổng LPT1 là 378h, địa chỉ thanh ghi trạng thái là 379h và địa chỉ thanh ghi điều khiển là 37Ah.

7.2.1.1. Chế độ cơ bản SPP (Standard Paralled Port)

Còn gọi là chế độ Centronic, trong đó truyền dữ liệu được đồng bộ qua xung STB qua các giai đoạn cho trường hợp ghép nối với máy in. Vì tốc độ truyền ký tự của cổng song song nhanh hơn tốc độ in của máy in nên thông tin ở đây phải có móc nối.

Khởi đầu, máy tính đặt các số liệu lên bus. Khi máy in xử lý xong dữ liệu, nó sẽ trả lại tín hiệu \overline{ACK} xuống mức thấp để ghi nhận. Máy tính đợi cho đến khi đường BSY từ máy in xuống thấp (máy in không bận) thì sẽ đưa tiếp số liệu trên bus. Điều khiển mức nổi mềm như trên cho tốc độ khoảng 150 KB/s. Gần đây, một số chip super I/O dùng bộ nhớ FIFO để đệm dữ liệu và dùng các tín hiệu mức nổi cứng trên tốc độ tăng khoảng 500 KB/s.

7.2.1.2. Chế độ cải tiến EPP (Enhanced Paralled Port)

Có 4 chu kỳ viết dữ liệu, viết địa chỉ, đọc địa chỉ. Ngoài 3 thanh ghi dùng chung cho cả chế độ SPP và EPP, chế độ này còn có thêm thanh ghi địa chỉ EPP có địa chỉ Offset bằng 3, thanh ghi dữ liệu EPP có địa chỉ Offset bằng 4 và 1 số thanh ghi dữ liệu có địa chỉ Offset từ 5 đến 7. Quá trình truyền dữ liệu xảy ra trong 1 chu kỳ đọc/viết của bus mở rộng ISA, do đó tốc độ truyền có thể nâng lên khoảng $500\text{KB/s} \div 2 \text{ MB/s}$.

7.2.1.3. Chế độ mở rộng ECP (Extended Capability Port)

Chế độ này cho phép truyền dữ liệu theo 2 hướng với 2 chu kỳ: chu kỳ lệnh và chu kỳ dữ liệu. Chế độ ECP sử dụng mã chiều dài chạy RLE (Run - Length) và nhiều kênh địa chỉ nên cho phép nối nhiều thiết bị ngoại vi trao đổi thông tin với máy tính qua một cổng ghép nối duy nhất. Chuẩn ECP định nghĩa nhiều chế độ làm việc và có thêm nhiều thanh ghi.

Các máy tính hiện nay có thể cài đặt mọi chế độ kể trên. Số liệu còn có thể trao đổi qua các cổng song song với nhau giữa các máy vi tính. Muốn vậy, một vài đường điều khiển và trạng thái phải được nối với nhau để phục vụ việc mức nổi cứng

Các máy tính PC được trang bị ít nhất là 1 cổng song song và 1 cổng nối tiếp. Khác với ghép nối nối tiếp có nhiều ứng dụng, ghép nối song song thường chỉ phục vụ cho máy in. Sơ đồ ghép nối song song như hình sau:

7.2.2. Cổng nối tiếp COM

Khác với cổng song song, cổng nối tiếp cho phép truyền dữ liệu dưới dạng các chuỗi bit kế tiếp nhau trong các từ dữ liệu nên chỉ cần 1 đường dây (và một dây đất). Truyền nối tiếp được phân biệt thành 2 dạng:

- Truyền đồng bộ, trong đó ngoài tín hiệu số liệu phải thêm vào tín hiệu nhịp đồng hồ làm chuẩn. Thường ngoài đường dây dữ liệu, phải đưa thêm vào một đường tín hiệu đồng bộ để chỉ thị rằng khi nào bit tiếp theo ổn định trên đường dữ liệu.

- Truyền không đồng bộ, trong đó các bit số liệu tự nó chứa các thông tin để đồng bộ. Thông tin đồng bộ (trong truyền không đồng bộ) gồm có các *bit khởi phát* (Start) chỉ thị bắt đầu của khối dữ liệu được đưa truyền và ít nhất có một *bit kết thúc* (Stop) chỉ thị kết thúc khối dữ liệu đó. Ngoài ra các chuỗi *bit chẵn lẻ* còn có thể được thêm vào, dùng cho phát hiện lỗi trên đường truyền.

Một thông số khác liên quan tới truyền số liệu nối tiếp là *tốc độ truyền* được gọi là *số baud* (theo tên nhà bác học Pháp J.M.E. Baudot) là số thay đổi trạng thái tín hiệu trên đường truyền trong một giây. Với tín hiệu máy tính, số baud chính bằng số bit được truyền trong một giây (bps).

7.2.2.1. Chuẩn ghép nối nối tiếp RS-232C

Các cổng nối tiếp đa năng trong máy vi tính đều được thiết kế hoạt động tuân theo tiêu chuẩn RS -232C (Reference Standard) của IEA (Electronic Industries Association) hoặc theo tổ chức CCITT ở Châu Âu là V.24. Chuẩn này được quy định ghép nối về cơ khí, điện và logic giữa một *thiết bị đầu cuối dữ liệu DTE* (Data Terminnal Equipment) và *thiết bị truyền số liệu DCE* (Data Communication Equipment). Ví dụ, DTE là máy tính còn DCE là modem.

Tín hiệu theo tiêu chuẩn RS-232C là lưỡng cực, trên đường truyền có logic âm như sau:

- Mức logic cao “1” có điện thế trong dải từ -3V đến -15V.
- Mức logic thấp “0” có điện thế trong dải từ +3V đến +15V.

Có các phương thức thông tin giữa DTE và DCF như sau:

- Truyền đơn công (Simplex Connection): số liệu chỉ được gửi theo một chiều.
- Truyền bán song công (Half – Duplex): số liệu chỉ được gửi theo hai chiều, nhưng mỗi thời điểm chỉ được truyền theo một chiều.
- Truyền song công (full-Duplex): số liệu được truyền đồng thời theo hai chiều.

Dữ liệu trên đường truyền chỉ ở một trong hai trạng thái: đánh dấu (Mark) hoặc trống (Space), lần lượt tương ứng với trạng thái điện thế âm hoặc dương tức là tương ứng với mức logic 1 hoặc 0. Dữ liệu được truyền lần lượt theo từng nhóm bit, mỗi nhóm gọi là một đơn vị dữ liệu nối tiếp SDU (Serial Data Unit) hay một khung truyền (Frame). Một khung truyền bao gồm:

- + 1 *bit start* luôn ở một mức logic thấp, điện thế dương.
- + 1 hoặc 1.5 hoặc 2 *bit stop* luôn ở mức logic cao, điện thế âm.
- + 1 hoặc không có 1 *bit kiểm tra chẵn lẻ*.
- + 5, 6 hoặc 7 *bit số liệu*

Chuẩn RS - 232C cho phép truyền tín hiệu với tốc độ 20,000 baud, nhưng nếu cáp truyền đủ ngắn thì tốc độ này có thể đạt tới 115.200 baud. Có được những chỉ tiêu này là do đặc điểm của vi mạch điều khiển ghép nối nối tiếp UART. Chiều dài cáp cực đại trong truyền thông tin theo chuẩn RS-232C là từ 17 ÷ 20 m.

Tất cả các máy vi tính hiện nay đều được lắp đặt 1 hoặc 2 cổng ghép nối nối tiếp gọi là COM 1 (hoặc COM3) và COM 2 (hoặc COM 4). Có hai loại đầu cắm tín hiệu cho các cổng này là D-25 (25 chân) và D-9 (9 chân) thường được gắn ở phía sau hộp máy như hình 7-7. Các đầu cắm cho các cổng nối tiếp gắn trên hộp máy vi tính bao giờ cũng là loại đầu cắm đực (Male), đầu cắm ở cáp nối ra các thiết bị ngoại vi là đầu cắm cái (Female). Việc này để tránh nhầm lẫn với đầu cắm D-25 dùng cho cổng song song LPT luôn là loại đầu cắm cái.

Ngoài dây đất GND có điện thế 0V, có thể phân thành 2 nhóm đường dây gồm nhóm các đường truyền dữ liệu T×D, R×D và nhóm các đường tín hiệu điều khiển (gọi là các tín hiệu móc nối thông tin) gồm các đường còn lại.

7.2.2.2. Thâm nhập cổng nối tiếp qua DOS và BIOS

Bằng lệnh ngoại trú MODE của DOS, ta có thể có thể đặt các thông số cho cổng nối tiếp COM.

MODE COM2: 2400, E, 8, 1

Là chọn cổng COM 2, tốc độ 2400 baud, bit chẵn lẻ, 8 bit số liệu, 1 bit Stop.

Cũng có thể dùng ngắt 21 của DOS để phát hoặc thu số liệu qua cổng COM bằng 4 hàm:

- 03h: đọc một ký tự từ cổng.
- 04h: phát một ký tự từ cổng.
- 3Fh: đọc một file qua cổng.
- 40h: viết một file qua cổng.

BIOS cho phép thâm nhập cổng COM qua ngắt số 14 bằng các hàm 00h (khởi động cổng). Khởi tạo định dạng dữ liệu, đặt tốc độ truyền,..., hàm 01h và 02h cho phép phát và thu một ký tự; hàm 03h trở về trạng thái của cổng; các hàm 04h và 05h cho phép mở rộng các điều kiện khởi động khối ghép nối cũng như cho phép thâm nhập

các thanh ghi điều khiển modem. Các hàm này sau khi được thực hiện điều trả về byte trạng thái trong thanh ghi AH.

7.3. Cấu trúc Bus của máy tính

7.3.1. Băng thông

Cấu trúc bus nói một cách đơn giản là cách mà các vi xử lý và các bộ phận của máy tính liên kết với nhau qua các bus. Ta đã biết chu kỳ máy hay chu kỳ bus là một quá trình xảy ra trên bus để truyền tải dữ liệu. Lượng dữ liệu tối đa được truyền qua bus trong một khoảng thời gian nào đó gọi là *tốc độ truyền tải tối đa* hay còn được gọi là *băng thông* (bandwidth) của bus. Tốc độ này là một nhân tố quan trọng ảnh hưởng đến hiệu năng của máy tính, nó được tính theo số mega byte được truyền tải trong 1 giây theo công thức sau:

$$\text{Băng thông [MB/s]} = \frac{\text{Tốc độ bus (MHz)} \times \text{số byte trong một lần chuyển tải}}{\text{Số chu kỳ đồng hồ cho mỗi lần chuyển tải}}$$

Một máy vi tính có thể có nhiều loại bus như sau:

7.3.1.1. Bus bộ xử lý

Còn gọi là (Back Side Bus - BSB), là các đường truyền giữa vi xử lý và các mạch đệm trung gian, thường là đường truyền giữa bộ xử lý và bộ cache ngoại L2 hoặc L3. Bus này hoạt động với tốc độ nhanh nhất so với các loại bus khác và không bị tắc nghẽn. Nó cũng bao gồm các bus thành phần dữ liệu, địa chỉ và điều khiển. Ví dụ, trong hệ thống Pentium, bus bộ xử lý có 64 đường dữ liệu, 32 đường địa chỉ và các đường điều khiển.

7.3.1.2. Bus hệ thống

Còn gọi là (Front Side Bus - FSB), được sử dụng để truyền thông tin giữa vi xử lý và bộ nhớ chính RAM cũng như tới các ổ đĩa,... Bus này hoặc là thành phần của chính bus bộ xử lý hoặc trong nhiều trường hợp được phân cách với bus bộ xử lý bằng các mạch điện là các chip chuyên dụng. Với các hệ thống chạy ở tốc độ đồng hồ bản mạch chính cao sẽ có một chip điều khiển bộ nhớ cho phép điều khiển sự ghép nối giữa bus bộ xử lý có tốc độ nhanh hơn và bộ nhớ chính có tốc độ truy xuất chậm hơn. Do đó thông tin truyền trên bus hệ thống được truyền với tốc độ chậm hơn so với thông tin trên bus bộ xử lý. Tuy nhiên độ rộng bus dữ liệu ở đây luôn được thiết kế bằng độ rộng bus dữ liệu của bus bộ xử lý.

7.3.1.3. Bus vào/ra

Còn gọi là bus mở rộng: cho phép vi xử lý thông tin được với các thiết bị ngoại vi. Nó cho phép bổ xung vào hệ thống máy tính các thiết bị để mở rộng tính năng của máy vi tính. Các khe cắm mở rộng được nối vào bus mở rộng. Các bản mạch ghép nối được cắm vào các khe cắm này. Do đó, khi nói về tiêu chuẩn cho một loại bus mở rộng nào đó cũng có nghĩa là nói cho các khe cắm mở rộng và card dùng với nó.

Trong các máy vi tính hiện nay, nhiều ngoại vi được tích hợp ngay trên bản mạch chính. Thí dụ chúng có ít nhất hai bộ điều khiển chuẩn ghép nối ổ đĩa IDE (sơ và thứ cấp). bộ điều khiển đĩa âm thanh, ghép nối mạng,... thường nằm trong vài chip riêng gọi là chip Super I/O hay Chipset. Trong một hệ thống như vậy, có thể các khe cắm mở rộng không cần thiết nhưng những bộ điều khiển cài sẵn và các cổng vẫn sử dụng bus vào/ra để thông tin với CPU như bình thường giống như chúng được lắp trên các bản mạch cắm vào khe cắm mở rộng như trước đây.

Cần phân biệt bus mở rộng và bus cục bộ. Thí dụ, các bus mở rộng như bus ISA nối từ các bộ đệm ra tới các khe cắm mở rộng nhằm liên kết vi xử lý với các bản mạch ghép nối cắm trên chúng với điều kiện bản mạch này hoạt động ở tốc độ xung nhịp

không quá 8,33 MHz. Trong khi đó *bus cục bộ* (local bus) là các hệ bus nối giữa vi xử lý và các bộ đệm hoạt động không bị hạn chế bởi giới hạn 8,33 MHz này.

Một hệ thống bus thường gồm ba thành phần: bus địa chỉ, bus dữ liệu và bus điều khiển. Bus địa chỉ (Address Bus – A Bus) là bus một chiều có nhiệm vụ truyền các tín hiệu địa chỉ phát hành bởi CPU đến bộ nhớ hoặc các thiết bị vào ra. Các tín hiệu địa chỉ giúp CPU chọn được ô nhớ cần đọc/ghi hoặc thiết bị vào ra cần trao đổi dữ liệu. Bus dữ liệu (Data Bus – D Bus) là bus hai chiều có nhiệm vụ truyền các tín hiệu dữ liệu đi và đến CPU. Dữ liệu được bus dữ liệu chuyển từ CPU đến bộ nhớ hoặc thiết bị vào ra và ngược lại. Bus điều khiển (Control Bus – C Bus) là bus một chiều theo một hướng, có nhiệm vụ truyền các tín hiệu điều khiển từ CPU đến bộ nhớ hoặc thiết bị vào ra, và truyền các tín hiệu trạng thái từ bộ nhớ hoặc thiết bị vào ra về CPU. Các bus địa chỉ, dữ liệu và điều khiển thường phối hợp cùng tham gia truyền dẫn các tín hiệu địa chỉ, dữ liệu và điều khiển trong quá trình CPU trao đổi thông tin với bộ nhớ hoặc các thiết bị vào ra.

7.3.2. Cấu trúc các loại Bus trong máy tính

7.3.2.1. Cấu trúc PC/XT và Bus PC

Bus PC (Personal Computer bus) ra đời phục vụ cho kiến trúc của vi xử lý Intel 8086 trước đây chạy ở 4,77 MHz và là cơ sở của máy vi tính PC/XT (Extended Technology). Tần số xung nhịp trên bus được chuẩn hóa là 8,33 MHz. Nó có bus số liệu rộng 8 bit = 1 byte, do vậy băng thông tối đa của bus là 8,33 MB/s. Bus địa chỉ nối ra các khe cắm rộng 20 bit, do vậy có thể quản lý được tới 1 M địa chỉ. Khe cắm mở rộng nối với nó là loại hai hàng chân và 62 tiếp điểm. Các tiếp điểm được chế tạo từ bản mạch in 2 mặt có mạ vàng hoặc thiếc. Kích thước tiêu chuẩn của một card PC là: cao 106,68 mm (4,2 inch), dài 333,5 mm (13,13 inch) và rộng 12,7 mm (0,5 inch).

7.3.2.2. Cấu trúc PC/AT và bus ISA

Năm 1984, ba năm sau khi giới thiệu máy vi tính, hãng IBM cho ra đời máy PC/AT (Advanced Technology) đầu tiên cho một hiệu năng cao cấp từ ba đến năm lần so với PC XT. Các máy AT đầu tiên dùng vi xử lý 80286 không chỉ có bus truyền dữ liệu 16 bit và định địa chỉ 24 bit mà còn có thêm 3 kênh vào DMA và 7 mức ngắt bổ sung cho bus mở rộng. Các máy AT đầu tiên có tốc độ xung nhịp trên các bus mở rộng bằng tốc độ của vi xử lý là từ 6 MHz. Khi tốc độ của vi xử lý tăng lên, tốc độ của các bus mở rộng này cũng tăng theo đến chuẩn 8,33 MHz.

Trong các máy AT dùng các vi xử lý 80386, 80486 do tốc độ vi xử lý tăng lên cao nên đã xuất hiện các chip điều khiển bus mở rộng cùng các chip hỗ trợ. Những bộ chip mới này đã phân vùng cho bản mạch chính sao cho bộ nhớ DRAM cũng như bộ nhớ cache được kết nối thẳng với bus bộ xử lý để tránh hiện tượng thất cổ chai, còn lại các ngoại vi tốc độ thấp vẫn được kết nối với vi xử lý qua bus mở rộng có tốc độ thấp.

Chuẩn bus AT tương thích đến hơn 90% với chuẩn do hiệp hội chuẩn công nghiệp ISA (Industrial Standard Association) đề ra. Do vậy đến nay bus mở rộng trên máy AT gọi là bus ISA. Vì có độ rộng bus số liệu lên tới 16 bit nên trên bus có thể truyền được 2 byte trong một nhịp đồng hồ hay tính ra với tần số xung nhịp chuẩn hóa là 8,33 MHz, băng thông tối đa có thể đạt tới là 16,66 MB/s. Do bus địa chỉ cũng được mở rộng lên tới 24 bit nên có thể quản lý được tới 16 M địa chỉ. Khe cắm ISA là một khe cắm PC cộng thêm một khe cắm 2 hàng chân nữa gồm 32 tiếp điểm, trong đó được thêm vào 8 đường số liệu và 4 đường địa chỉ cao như hình bên. Do vậy, một card PC bất kỳ hoàn toàn có thể cắm và sử dụng được khe cắm ISA.

7.3.2.3 Bus MCA

Hãng IBM đã phát triển cấu trúc bus trên kênh MCA (Microchannel Interface Architecture) cho các máy tính PS/2 của họ. Cấu trúc này thêm vào một số đặc điểm như: việc làm chủ bus được hỗ trợ tốt cho phép một bản mạch ghép nối được nắm quyền điều khiển bus từ bộ nhớ xử lý và thông tin dữ liệu với bộ nhớ chính một cách độc lập với bộ xử lý; hỗ trợ cho việc tự định cấu hình của các bản mạch ghép nối (tiền đề cho chức năng cắm là chạy Plug and play),... Bus này hoàn toàn không tương thích với bus ISA và có thể hoạt động với bề rộng bus dữ liệu 16 hoặc 32 bit. Sự khác nhau chính về mặt kỹ thuật là trong khi các bus kể trên là đồng bộ thì bus MCA là một loại bus không đồng bộ. Nghĩa là nó được làm việc ở một tần số nhịp đồng hồ cố định không phụ thuộc vào đồng hồ hệ thống của PC.

Một bus MCA có thể cho một tốc độ truyền tải tối đa lên tới 160 Mbyte/ giây. Chuẩn bus MCA cho 4 kích thước card khác nhau với 4 kiểu khe cắm 16 bit, 16 bit với những phần mở rộng video, 16 bit với những phần mở rộng bộ nhớ và 32 bit.

7.3.2.4 Bus EISA

Chuẩn bus này được ra đời nhằm mở rộng cấu trúc ISA để hỗ trợ ngoại vi 32-bit với tốc độ truyền cao hơn, do sự cạnh tranh với bus MCA.

Khe cắm EISA (Extended ISA) có 2 nấc. Nấc trên cho các tiếp điểm với tuân hoàn tương thích với chuẩn ISA, do vậy một card ISA có thể cắm vào đó. Có một cái chốt đặc biệt cho phép card EISA có thể được cắm sâu hơn nữa vào khe cắm để có thể tiếp xúc với nấc dưới có các tiếp điểm được nối với các bus số liệu và địa chỉ rộng 32 bit. Chuẩn EISA cũng có tốc độ nhịp trên bus là 8,33MHz nhưng do độ rộng bus dữ liệu là 32 bit nên tốc độ truyền tải tối đa ở đây lên tới 33,32 Mbyte/s và bus có thể địa chỉ hoá được bộ nhớ tới 4 Gbyte. Kích thước tiêu chuẩn của một card EISA là: cao 127mm (5 inch), dài 33,5 mm (1,33 inch) và rộng 12,7 mm (0,5 inch).

7.3.2.5 Bus cục bộ VL

Nhược điểm chính của các bus PC, ISA và EISA là gặp phải hiệu ứng “thắt cổ chai” làm lãng phí của các vi xử lý tốc độ cao hơn trong khi phải truyền tin trên các bus có băng thông thấp và như vậy làm giảm hiệu suất của toàn bộ hệ thống. Do đó hiệp hội VESA (Video Electronics Standards Association) đã phát triển chuẩn bus cục bộ VL (VESA local bus) cho phép các card ghép nối từ vi xử lý đến bản mạch video tốc độ nhanh có thể hoạt động được.

7.3.2.6. Bus PCI

Chuẩn kết nối các thành phần ngoại vi PCI (Peripheral Component Interconnect) là loại chuẩn bus vào/ra được dùng phổ biến nhất hiện nay. Nó cung cấp một đường dữ liệu chia sẻ giữa vi xử lý và các bộ điều khiển thiết bị ngoại vi trong tất cả các máy tính từ máy tính xách tay đến máy tính lớn.

Trong máy tính chỉ dành một số hạn chế các ngắt cứng và bus PCI được thiết kế cho phép chia sẻ giữa các ngắt này. Vì vậy trên các bản mạch chính chỉ có khe cắm PIC không bao giờ xảy ra xung đột giữa các ngắt như trường hợp hay xảy ra với các card ghép nối cắm vào khe cắm ISA như trước đây. Công nghệ PCI cho phép truy cập tốt các bộ nhớ nhanh, các ổ đĩa và card màn hình tốc độ cao. Một tập các vi mạch ghép nối như 82430 PCI được sản xuất sẵn cho các mục đích đó. Giống như bus VL, bus PCI cũng sử dụng đồng hồ hệ thống cho việc truyền dữ liệu nhưng có ưu điểm hơn là nó có thể hoạt động trên các đường dẫn 32 bit hoặc 64 bit.

Do tốc độ truyền tải tốc độ dữ liệu cao nên số ghép nối bus được hạn chế tới 2 hoặc 3 đơn vị (thường là chỉ gồm card màn hình và bộ điều khiển ổ đĩa cứng). Trong trường hợp tốc độ xung nhịp là 33 MHz với độ rộng bus dữ liệu là 64 bit (8 byte), bus này cho một tốc độ truyền tải dữ liệu lên tới 264 Mbyte/s. Những năm gần đây hầu hết

các bản mạch như card âm thanh, card mạng LAN, card Fax- Modem cũng được thiết kế và chạy với bus PCI.

7.3.2.7. Bus AGP

Bus AGP (Accelerated Graphic Port) do Intel phát triển năm 1993 với mục đích chính sử dụng cho kết nối với các mạch xử lý đồ họa tốc độ cao. AGP đã hoàn toàn thay thế PCI trong lĩnh vực giao tiếp đồ họa trong các năm sau đó.

AGP hỗ trợ băng thông 32 bit với tốc độ truyền dữ liệu nhanh gấp nhiều lần so với bus PCI. Cụ thể, AGP hỗ trợ 4 cấp tốc độ truyền dữ liệu là 1x, 2x, 4x và 8x, với tốc độ lần lượt là 266MB/s, 533MB/s, 1066MB/s và 2133MB/s tại các tần số tương ứng 66MHz, 133MHz, 266MHz và 533MHz.

7.3.2.8. Bus nối tiếp đa năng USB

Bus USB (Universal Serial Bus) là một công nghệ mới và đã được phát triển nhanh chóng trong các nhà sản xuất thiết bị đầu cuối truyền thông tin nối tiếp. Đây là một giao diện mạnh, đơn giản và dễ sử dụng nên được dùng cho phép nối với nhiều loại thiết bị như bàn phím, chuột, ổ đĩa, camera, màn hình, máy in,... Nó cho phép nối lên tới 127 thiết bị theo kiểu móc xích.

Chuẩn USB 1.0 và 1.1 có băng thông cực đại là 12 Mbps và chuẩn USB 2.0 đạt tới 480 Mbps. Với tốc độ truyền dữ liệu 1.5 Mbps rồi 12 bus này cũng đáp ứng thỏa đáng cho hầu hết các thiết bị ngoại vi. Một lợi thế của chuẩn này là các thiết bị ngoại vi tự nhận dạng tương thích với hệ thống “cắm và chạy ngay” (Plug and Play - PnP) và cung cấp tiêu chuẩn công nghệ cho các kết nối tương lai. Hơn nữa những thiết bị USB có thể cắm hoặc ngắt “nóng”, nghĩa là không phải tắt máy vì tính khi muốn cắm hoặc rút ra một thiết bị USB vào hệ thống máy tính. Thời gian gần đây hầu hết các bản mạch chính đã được thiết kế hỗ trợ chuẩn bus này.

Chuẩn USB chia ra 2 loại thiết bị: HUB gồm bộ điều khiển và bộ lặp (Repeater) có nhiệm vụ chuyển một ổ cắm USB thành nhiều ổ và các *thiết bị chức năng* chính là thiết bị ngoại vi.

7.3.2.9. Bus FireWire (IEEE – 1394)

Đây là một công nghệ bus phục vụ cho nhu cầu truyền dữ liệu lớn liên quan đến các thiết bị đa phương tiện video và âm thanh.

Bus FireWire hỗ trợ cho truyền thông tin nối tiếp. Chuẩn này được Ủy ban tiêu chuẩn IEEE công bố vào năm 1995. Tiêu chuẩn này tồn tại đồng thời 3 tốc độ truyền tín hiệu khác nhau: 100Mbps, 200Mbps và 400Mbps. Chuẩn này có thể kết nối tới 63 thiết bị ngoại vi theo kiểu móc xích. Cáp nối gồm 4 dây tín hiệu và 2 dây nguồn được nối với bản mạch chính thông qua các bản mạch ghép nối IEEE – 1394 hoặc qua các bản mạch ghép nối PCI.

Giống như chuẩn USB, việc truyền dữ liệu trong IEEE – 1394 cũng theo các gói với hai hình thức là:

- Truyền không đồng bộ: dành cho truyền địa chỉ và các ứng dụng tốc độ chậm như máy in, máy quét,...
- Truyền đồng bộ theo số kênh: trong đó một gói dữ liệu đồng bộ cần khoảng 125μs dùng để truyền các hình ảnh chuyển động. Địa chỉ cho một thiết bị là 64 bit bao gồm các trường: mã số mạng (10 bit), mã số điểm nối (6 bit) và địa chỉ (48 bit). Với cấu trúc như vậy, Bus FireWire có thể quản lý được 1023 mạng, 63 thiết bị ngoại vi và 281 Terabyte nhớ.