



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

GIÁO TRÌNH

KIẾN TRÚC MÁY TÍNH

Biên soạn: ThS VÕ ĐỨC KHÁNH



NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

Chương 1

Giới thiệu chung

Môn học kiến trúc máy tính là môn học khảo sát cấu trúc và chức năng của máy tính. Môn học này giúp học viên hiểu một cách rõ ràng, đầy đủ về bản chất cũng như những đặc trưng của các hệ thống máy tính hiện đại. Đây là một nhiệm vụ có tính thách đố do:

- Tính đa dạng của máy tính thể hiện trong giá cả, kích thước, khả năng vận hành & ứng dụng.
- Sự thay đổi nhanh chóng về công nghệ máy tính, từ kỹ thuật mạch tích hợp dùng để xây dựng nên các thành phần máy tính cho đến việc gia tăng sử dụng những khái niệm về tổ chức song song trong việc kết hợp các thành phần đó.

Mặc dù có sự hiện diện của tính đa dạng và tốc độ thay đổi công nghệ trong lĩnh vực máy tính, nhiều khái niệm cơ bản vẫn được áp dụng rộng khắp. Trong giáo trình này, các yếu tố cơ bản về kiến trúc và tổ chức máy tính, mối quan hệ giữa chúng cũng như nhiều bài toán gặp phải trong thiết kế máy tính hiện nay sẽ được thảo luận chi tiết.

1.1 TỔ CHỨC & KIẾN TRÚC MÁY TÍNH

Hai thuật ngữ **tổ chức máy tính** và **kiến trúc máy tính** là hai thuật ngữ cần được phân biệt khi mô tả một hệ thống máy tính.

- **Kiến trúc máy tính** đề cập đến những thuộc tính hệ thống mà lập trình viên có thể quan sát được. Nói cách khác, đó là các thuộc tính có ảnh hưởng trực tiếp đến việc thực thi một chương trình, ví dụ như tập chỉ thị của máy tính, số bit được sử dụng để biểu diễn dữ liệu, cơ chế nhập/xuất, kỹ thuật định địa chỉ bộ nhớ, v.v...
- **Tổ chức máy tính** quan tâm đến các đơn vị vận hành và sự kết nối giữa chúng nhằm hiện thực hóa những đặc tả về kiến trúc, chẳng hạn như về tín hiệu điều khiển, giao diện giữa máy tính với các thiết bị ngoại vi, kỹ thuật bộ nhớ được sử dụng, v.v...

Để minh họa rõ hơn về hai khái niệm này, chúng ta hãy xét đến phép toán nhân. Việc máy tính có trang bị phép toán này hay không là vấn đề thuộc về kiến trúc máy tính. Trong khi đó, việc cài đặt phép toán thông qua một đơn vị nhân đặc biệt hay là qua cơ chế sử dụng lập đi lập lại đơn vị cộng của hệ thống lại là vấn đề của tổ chức máy tính. Ở đây sự chọn lựa sử dụng cơ chế nào phụ thuộc vào các yếu tố như tần số sử dụng phép toán, tốc độ tương đối của cả hai cách tiếp cận, giá cả và kích thước vật lý của một đơn vị nhân đặc biệt.

1.2 CẤU TRÚC & CHỨC NĂNG CỦA MÁY TÍNH

Máy tính là một hệ thống phức tạp với hàng triệu thành phần điện tử cơ sở. Chìa khóa chính để có thể mô tả máy tính một cách rõ ràng là sự nhận thức về bản chất phân cấp của hầu hết các hệ

thống phức tạp. Một hệ thống phân cấp là một tập hợp gồm các hệ thống con có liên quan với nhau, trong đó mỗi hệ thống con lại có tính phân cấp về cấu trúc, cứ thế tiếp tục cho đến cấp thấp nhất chứa những hệ thống con cơ sở.

Bản chất phân cấp của một hệ thống phức tạp giữ vai trò chính trong việc thiết kế và mô tả hệ thống. Tại mỗi cấp, hệ thống bao gồm một tập hợp các thành phần con cùng với những mối liên hệ giữa chúng. Ở đây có hai yếu tố được quan tâm đến là cấu trúc và chức năng:

- **Cấu trúc:** cách thức các thành phần hệ thống liên hệ với nhau.
- **Chức năng:** hoạt động của mỗi thành phần riêng lẻ với tư cách là một phần của cấu trúc.

1.2.1 CHỨC NĂNG

Một cách tổng quát, một máy tính có thể thực hiện bốn chức năng cơ bản sau:

- **Xử lý dữ liệu:** máy tính phải có khả năng xử lý dữ liệu. Dữ liệu có thể có rất nhiều dạng và phạm vi yêu cầu xử lý cũng rất rộng. Tuy nhiên, chúng ta sẽ thấy rằng chỉ có một số phương pháp cơ bản trong xử lý dữ liệu.
- **Lưu trữ dữ liệu:** máy tính cũng cần phải có khả năng lưu trữ dữ liệu. Ngay cả khi máy tính đang xử lý dữ liệu, nó vẫn phải lưu trữ tạm thời tại mỗi thời điểm phần dữ liệu nó đang làm việc. Do vậy, ít nhất chúng ta cần có chức năng lưu trữ ngắn hạn. Tuy nhiên, chức năng lưu trữ dài hạn cũng có tầm quan trọng tương

đương, vì dữ liệu cần được lưu trữ trên máy cho những lần cập nhật và tìm kiếm kế tiếp.

- **Di chuyển dữ liệu:** máy tính phải có khả năng di chuyển dữ liệu giữa nó và thế giới bên ngoài. Khả năng này được thể hiện thông qua việc di chuyển dữ liệu giữa máy tính với các thiết bị nối kết trực tiếp hay từ xa đến nó. Tùy thuộc vào kiểu kết nối và cự ly di chuyển dữ liệu, chúng ta có tiến trình nhập xuất dữ liệu hay truyền dữ liệu:

Tiến trình nhập xuất dữ liệu: thực hiện di chuyển dữ liệu trong cự ly ngắn giữa máy tính và thiết bị nối kết trực tiếp.

Tiến trình truyền dữ liệu: thực hiện di chuyển dữ liệu trong cự ly xa giữa máy tính và thiết bị nối kết từ xa.

- **Điều khiển:** bên trong hệ thống máy tính, đơn vị điều khiển có nhiệm vụ quản lý các tài nguyên máy tính và điều phối sự vận hành của các thành phần chức năng phù hợp với yêu cầu nhận được từ người sử dụng.

Tương ứng với các chức năng tổng quát nói trên, có bốn loại hoạt động có thể xảy ra gồm:

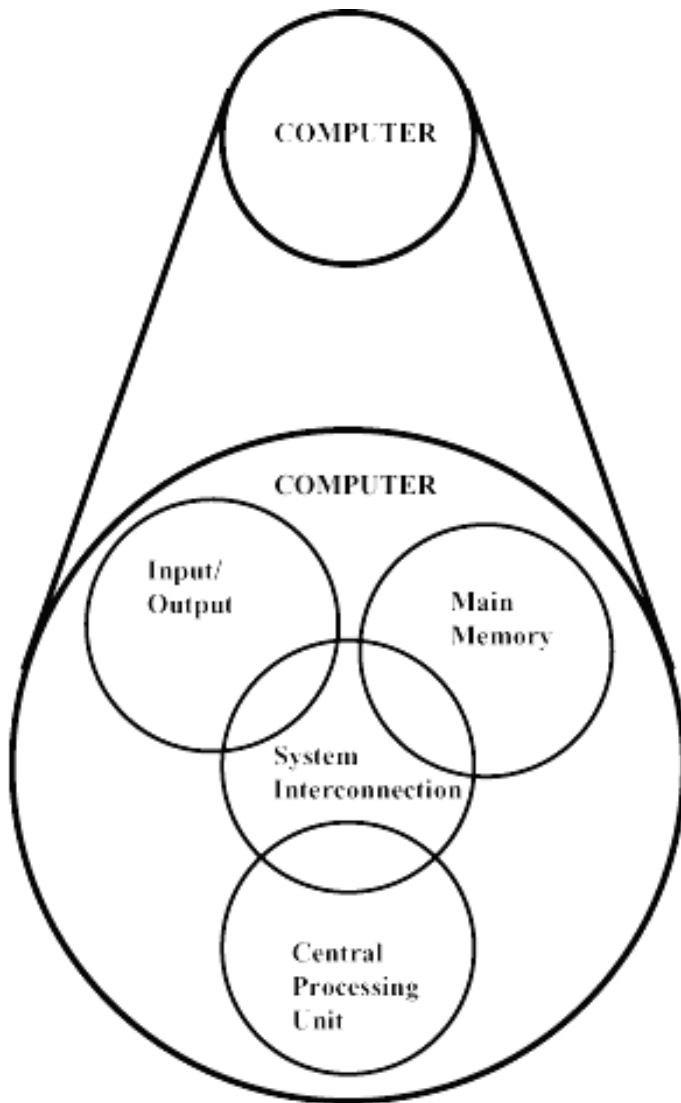
- Máy tính được dùng như một thiết bị di chuyển dữ liệu, có nhiệm vụ đơn giản là chuyển dữ liệu từ bộ phận ngoại vi hay đường liên lạc này sang bộ phận ngoại vi hay đường liên lạc khác.
- Máy tính được dùng để lưu trữ dữ liệu, với dữ liệu được chuyển từ môi trường ngoài vào lưu trữ trong máy (quá trình đọc dữ liệu) và ngược lại (quá trình ghi dữ liệu).

- Máy tính được dùng để xử lý dữ liệu thông qua các thao tác trên dữ liệu lưu trữ hoặc kết hợp giữa việc lưu trữ và liên lạc với môi trường bên ngoài.

1.2.2 CẤU TRÚC

Cấu trúc nội tại của máy tính ở cấp tổng quát nhất được thể hiện trong hình 1.1, bao gồm bốn cấu trúc chính:

- **Đơn vị xử lý trung tâm (CPU):** điều khiển hoạt động của máy tính và thực hiện các chức năng xử lý dữ liệu. CPU thường được đề cập đến với tên gọi bộ xử lý.
- **Bộ nhớ chính:** dùng để lưu trữ dữ liệu.
- **Các thành phần nhập xuất:** dùng để di chuyển dữ liệu giữa máy tính và môi trường bên ngoài.
- **Các thành phần nối kết hệ thống:** cung cấp cơ chế liên lạc giữa CPU, bộ nhớ chính và các thành phần nhập xuất.



Hình 1.1 Cấu trúc tổng quát của máy tính

1.3 CÁCH TIẾP CẬN CỦA GIÁO TRÌNH

Giáo trình được tổ chức thành hai phần chính như sau:

- Phần 1: Tổng quan về kiến trúc máy tính
- Phần 2: Hệ thống máy tính

Nội dung chi tiết của từng phần được liệt kê tiếp sau đây.

Phần 1: Tổng quan về kiến trúc máy tính

Phần 1 gồm có hai chương.

- Chương 1 giới thiệu chung về môn học và tổ chức giáo trình.
- Chương 2 trình bày lịch sử công nghệ máy tính, qua đó giới thiệu những khái niệm cơ bản về tổ chức và kiến trúc máy tính.

Phần 2: Hệ thống máy tính

Phần 2 gồm có ba chương.

- Chương 3 khảo sát kỹ thuật đường truyền hệ thống, một trong những cách tiếp cận phổ biến nhất đối với bài toán liên kết các thành phần bên trong máy tính.
- Chương 4 giới thiệu về tính phân cấp bộ nhớ, sau đó tập trung vào những vấn đề thiết kế liên quan đến bộ nhớ trong. Các chủ đề được thảo luận bao gồm bản chất và tổ chức của bộ nhớ chính bán dẫn, thiết kế cache.
- Chương 5 tìm hiểu về những tham số hiệu suất và thiết kế khác nhau có liên quan đến bộ nhớ đĩa. Ngoài ra, các lược đồ RAID, vốn đang trở nên phổ biến trên thị trường cũng được trình bày ở mức chi tiết.

Chương 2

Lịch sử máy tính

Máy tính thường được phân loại dựa trên công nghệ phần cứng cơ sở được sử dụng trong quá trình chế tạo. Lịch sử phát triển máy tính có thể chia làm bốn giai đoạn như sau:

- **Giai đoạn 1:** từ 1945 đến 1958, với máy tính thế hệ thứ nhất sử dụng công nghệ đèn chân không.
- **Giai đoạn 2:** từ 1958 đến 1964, với máy tính thế hệ thứ hai sử dụng công nghệ chất bán dẫn.
- **Giai đoạn 3:** từ 1964 đến 1974, với máy tính thế hệ thứ ba sử dụng công nghệ mạch tích hợp.
- **Giai đoạn 4:** từ 1974 đến nay, với máy tính thế hệ thứ tư sử dụng công nghệ mạch tích hợp vô cùng lớn/siêu lớn (VLSI/ULSI).

Các mục tiếp theo sẽ trình bày chi tiết về từng thế hệ máy tính cùng với công nghệ sử dụng và đại diện tiêu biểu của thế hệ đó.

2.1 MÁY TÍNH THẾ HỆ THỨ NHẤT (1945 – 1958)

❑ Máy tính ENIAC

Máy ENIAC (Electronic Numerical Integrator And Computer), do John Mauchly và John Presper Eckert (đại học Pennsylvania, Mỹ) thiết kế và chế tạo, là chiếc máy số hoá điện tử đa năng đầu tiên trên thế giới.

Nguồn gốc

Dự án chế tạo máy ENIAC được bắt đầu vào năm 1943. Đây là một nỗ lực nhằm đáp ứng yêu cầu thời chiến của BRL (Ballistics Research Laboratory – Phòng nghiên cứu đạn đạo quân đội Mỹ) trong việc tính toán chính xác và nhanh chóng các bảng số liệu đạn đạo cho từng loại vũ khí mới.

Số liệu kỹ thuật

ENIAC là một chiếc máy khổng lồ với hơn 18000 bóng đèn chân không, nặng hơn 30 tấn, tiêu thụ một lượng điện năng vào khoảng 140kW và chiếm một diện tích xấp xỉ 1393 m². Mặc dù vậy, nó làm việc nhanh hơn nhiều so với các loại máy tính điện cơ cùng thời với khả năng thực hiện 5000 phép cộng trong một giây đồng hồ.

Điểm khác biệt giữa ENIAC & các máy tính khác

ENIAC sử dụng hệ đếm thập phân chứ không phải nhị phân như ở tất cả các máy tính khác. Với ENIAC, các con số được biểu diễn dưới dạng thập phân và việc tính toán cũng được thực hiện trên hệ thập phân. Bộ nhớ của máy gồm 20 “bộ tích lũy”, mỗi bộ có khả năng lưu giữ một số thập phân có 10 chữ số. Mỗi chữ số được thể hiện bằng một vòng gồm 10 đèn chân không, trong đó tại mỗi thời điểm, chỉ có một đèn ở trạng thái bật để thể hiện một trong mười chữ số từ 0 đến 9 của hệ thập phân. Việc lập trình trên ENIAC là một công việc vất vả vì phải thực hiện nối dây bằng tay qua việc đóng/mở các công tắc cũng như cắm vào hoặc rút ra các dây cáp điện.

Hoạt động thực tế

Máy ENIAC bắt đầu hoạt động vào tháng 11/1945 với nhiệm vụ đầu tiên không phải là tính toán đạn đạo (vì chiến tranh thế giới lần

thứ hai đã kết thúc) mà để thực hiện các tính toán phức tạp dùng trong việc xác định tính khả thi của bom H. Việc có thể sử dụng máy vào mục đích khác với mục đích chế tạo ban đầu cho thấy tính đa năng của ENIAC. Máy tiếp tục hoạt động dưới sự quản lý của BRL cho đến khi được tháo rời ra vào năm 1955.

Với sự ra đời và thành công của máy ENIAC, năm 1946 được xem như năm mở đầu cho kỷ nguyên máy tính điện tử, kết thúc sự nỗ lực nghiên cứu của các nhà khoa học đã kéo dài trong nhiều năm liền trước đó.

❑ Máy tính von Neumann

Khái niệm chương trình được lưu trữ

Như đã đề cập ở trên, việc lập trình trên máy ENIAC là một công việc rất tẻ nhạt và tốn kém nhiều thời gian. Công việc này có lẽ 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 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 “khái niệm chương trình được lưu trữ”, do nhà toán học John von Neumann, một cố vấn của dự án ENIAC, đưa ra ngày 8/11/1945, trong một bản đề xuất về một loại máy tính mới có tên gọi 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.

❑ Máy IAS

Tiếp tục với ý tưởng của mình, vào năm 1946, von Neuman cùng các đồng nghiệp bắt tay vào thiết kế một máy tính mới có chương trình được lưu trữ với tên gọi IAS (Institute for Advanced Studies) tại học viện nghiên cứu cao cấp Princeton, Mỹ. Mặc dù mãi đến năm 1952 máy IAS mới được hoàn tất, nó vẫn là mô hình cho tất cả các máy tính đa năng sau này.

Cấu trúc tổng quát của máy IAS gồm có:

- **Một bộ nhớ chính** để lưu trữ dữ liệu và chương trình.
- **Một đơn vị số học – luận lý** (ALU – Arithmetic and Logic Unit) có khả năng thao tác trên dữ liệu nhị phân.
- **Một đơn vị điều khiển** có nhiệm vụ thông dịch các chỉ thị trong bộ nhớ và làm cho chúng được thực thi.
- **Thiết bị nhập/xuất** được vận hành bởi đơn vị điều khiển.

Hầu hết các máy tính hiện nay đều có chung cấu trúc và chức năng tổng quát như trên. Do vậy chúng còn có tên gọi chung là các máy von Neumann.

2.2 MÁY TÍNH THẾ HỆ THỨ HAI (1958 – 1964)

Sự thay đổi đầu tiên trong lĩnh vực máy tính điện tử xuất hiện khi có sự thay thế đèn chân không bằng đèn bán dẫn. Đèn bán dẫn nhỏ hơn, rẻ hơn, tỏa nhiệt ít hơn trong khi vẫn có thể được sử dụng theo cùng cách thức của đèn chân không để tạo nên máy tính. Không như đèn chân không vốn đòi hỏi phải có dây, có bảng kim loại, có bao thủy tinh và chân không, đèn bán dẫn là một thiết bị ở trạng thái rắn được chế tạo từ silicon có nhiều trong cát có trong tự nhiên.

Đèn bán dẫn là phát minh lớn của phòng thí nghiệm Bell Labs trong năm 1947. Nó đã tạo ra một cuộc cách mạng điện tử trong những năm 50 của thế kỷ 20. Dù vậy, mãi đến cuối những năm 50, các máy tính bán dẫn hóa hoàn toàn mới bắt đầu xuất hiện trên thị trường máy tính. Việc sử dụng đèn bán dẫn trong chế tạo máy tính đã xác định thế hệ máy tính thứ hai, với đại diện tiêu biểu là máy PDP-1 của công ty DEC (Digital Equipment Corporation) và IBM 7094 của IBM. DEC được thành lập vào năm 1957 và cũng trong năm đó cho ra đời sản phẩm đầu tiên của mình là máy PDP-1 như đã đề cập ở trên. Đây là chiếc máy mở đầu cho dòng máy tính mini của DEC, vốn rất phổ biến trong các máy tính thế hệ thứ ba.

2.3 MÁY TÍNH THẾ HỆ THỨ BA (1964 – 1974)

Một đèn bán dẫn tự chứa, đơn lẻ thường được gọi là một *thành phần rời rạc*. Trong suốt những năm 50 và đầu những năm 60 của thế kỷ 20, các thiết bị điện tử phần lớn được kết hợp từ những thành phần rời rạc – đèn bán dẫn, điện trở, tụ điện, v.v... Các thành phần rời rạc được sản xuất riêng biệt, đóng gói trong các bộ chứa riêng, sau đó được dùng để nối lại với nhau trên những bảng mạch. Các bảng này lại được gắn vào trong máy tính, máy kiểm tra dao động, và các thiết bị điện tử khác nữa. Bất cứ khi nào một thiết bị điện tử cần đến một đèn bán dẫn, một ống kim loại nhỏ chứa một mẫu silicon cỡ đầu pin sẽ phải được hàn vào một bảng mạch. Toàn bộ quá trình sản xuất, đi từ đèn bán dẫn đến bảng mạch, là một quá trình tốn kém và không hiệu quả. Những vấn đề như vậy đã làm nền tảng cho việc dẫn đến các bài toán mới trong công nghiệp máy tính. Các máy tính thế hệ thứ hai ban đầu chứa khoảng 10000 đèn bán dẫn. Con số này sau đó đã tăng lên nhanh chóng đến hàng trăm

ngàn, làm cho việc sản xuất các máy mạnh hơn, mới hơn gặp rất nhiều khó khăn.

Sự phát minh ra mạch tích hợp vào năm 1958 đã cách mạng hóa điện tử và bắt đầu cho kỷ nguyên vi điện tử với nhiều thành tựu rực rỡ. Mạch tích hợp chính là yếu tố xác định thế hệ thứ ba của máy tính. Trong mục tiếp sau đây chúng ta sẽ tìm hiểu một cách ngắn gọn về công nghệ mạch tích hợp. Sau đó, hai thành viên quan trọng nhất trong các máy tính thế hệ thứ ba, máy IBM System/360 và máy DEC PDP-8, sẽ được giới thiệu cùng với các tính năng nổi bật của chúng.

□ Vi điện tử

Kể từ buổi ban đầu của điện tử số và công nghiệp máy tính, mọi người đã có một khuynh hướng nhất quán và vững chắc trong việc thu nhỏ kích thước các mạch điện tử số. Trước khi xem xét những lợi ích do khuynh hướng này mang lại, chúng ta cần tìm hiểu đôi chút về bản chất của điện tử số.

Các thành phần cơ bản của một máy tính số, như chúng ta đã biết, phải thực hiện các chức năng lưu trữ, di chuyển, xử lý, và điều khiển. Chỉ có hai kiểu thành phần cơ sở là cần thiết: cổng và ô nhớ.

- **Cổng** là một thiết bị cài đặt một hàm luận lý hay Boolean đơn giản, chẳng hạn như **NẾU A VÀ B LÀ ĐÚNG THÌ C LÀ ĐÚNG** (cổng AND). Những thiết bị như thế được gọi là cổng vì chúng điều khiển luồng dữ liệu gần giống với cách hoạt động của những cổng tại các kênh đào.
- **Ô nhớ** là một thiết bị có thể lưu trữ một bit dữ liệu; tức là nó có thể ở một trong hai trạng thái tại một thời điểm bất kỳ.

Bằng cách liên kết một lượng lớn những thiết bị cơ sở này, chúng ta có thể xây dựng được một máy tính. Chúng ta có thể liên hệ điều này với bốn chức năng cơ bản của máy tính như sau:

- **Lưu trữ dữ liệu:** do ô nhớ cung cấp.
- **Xử lý dữ liệu:** do cổng cung cấp.
- **Di chuyển dữ liệu:** đường đi giữa các thành phần được sử dụng để di chuyển dữ liệu từ ô nhớ này sang ô nhớ khác và từ ô nhớ qua cổng đến ô nhớ khác.
- **Điều khiển:** đường đi giữa các thành phần có thể được sử dụng để mang chuyển tín hiệu điều khiển. Lấy ví dụ, một cổng sẽ có một hoặc hai bộ nhập dữ liệu cộng với một tín hiệu điều khiển cho phép kích hoạt cổng. Khi tín hiệu điều khiển là BẬT, cổng sẽ thực hiện chức năng của nó trên dữ liệu nhập và cho ra dữ liệu xuất. Một cách tương tự, ô nhớ sẽ lưu bit được nhập vào khi tín hiệu điều khiển ghi WRITE là BẬT và sẽ đặt bit đó trên bộ xuất khi tín hiệu đọc READ là BẬT.

Do đó, một máy tính sẽ bao gồm các cổng, các ô nhớ, cũng như các thành phần liên kết chúng. Cổng và ô nhớ lại được tạo nên từ những thành phần điện tử số đơn giản.

Mặc dù công nghệ bán dẫn đã được giới thiệu trong các máy tính thế hệ thứ hai, nhiều bài toán vẫn còn tồn tại. Các đèn bán dẫn được đặt riêng lẻ trong các gói và được liên kết lại trên những bảng mạch in thông qua các dây rời rạc. Đây là một quá trình phức tạp, tốn thời gian và dễ có lỗi.

Công nghệ mạch tích hợp khai thác sự kiện là những thành phần như thế (đèn bán dẫn, điện trở, và chất dẫn điện) có thể làm hàng

loạt từ một chất bán dẫn như silicon. Hàng trăm, thậm chí hàng ngàn đèn bán dẫn có thể được tạo ra cùng một lúc trên một vi silicon. Ngoài ra, những đèn bán dẫn này có thể kết nối với một quá trình kim loại hóa để tạo thành các mạch khác nhau.

Vào lúc ban đầu, chỉ có một số ít cổng hay ô nhớ có thể được sản xuất và đóng gói lại với nhau một cách đáng tin cậy. Những mạch tích hợp ban đầu này được đề cập đến với tên gọi *tích hợp mức nhỏ*. Dần dần người ta đã có thể đặt nhiều thành phần hơn trên cùng một chip. Bắt đầu ở mức đơn vị vào năm 1959, số thiết bị trên mỗi chip đã gia tăng gấp đôi hàng năm trong những năm 1960. Đến những năm 70, tốc độ này có giảm xuống, nhưng vẫn còn ở mức đáng lưu ý là tăng gấp 4 lần trong khoảng ba năm một. Mức phát triển này tồn tại cho đến đầu những năm 1990, khi tác động của những giới hạn về vật lý một lần nữa làm chậm mức độ tăng trưởng của các thành phần trên một chip. Tuy nhiên, theo các dự đoán lạc quan hơn, tích hợp ở mức giga (GSI) – một tỉ thành phần trên một chip – sẽ đạt được trong vòng một vài năm sắp đến.

Đối với nhà sản xuất máy tính, việc sử dụng nhiều IC được đóng gói mang lại nhiều điểm có ích như sau:

- Giá chip gần như không thay đổi trong quá trình phát triển nhanh chóng về độ trù mật của các thành phần trên chip. Điều này có nghĩa là giá cả cho các mạch nhớ và luận lý giảm một cách đáng kể.
- Vì những thành phần luận lý và ô nhớ được đặt gần nhau hơn trên các chip được đóng gói dày đặc, đường đi điện tử sẽ ngắn hơn dẫn đến việc gia tăng tốc độ vận hành.
- Máy tính sẽ trở nên nhỏ hơn, tiện lợi hơn để bố trí vào các loại môi trường khác nhau.

- Có sự giảm thiểu trong những yêu cầu về bộ nguồn và thiết bị làm mát hệ thống.
- Sự liên kết trên mạch tích hợp đáng tin cậy hơn trên các nối kết hàn. Với nhiều mạch trên mỗi chip, sẽ có ít sự nối kết liên chip hơn.

❑ Máy IBM System/360

Máy IBM System/360 được IBM đưa ra vào năm 1964 là họ máy tính công nghiệp đầu tiên được sản xuất một cách có kế hoạch. Khái niệm họ máy tính bao gồm các máy tính tương thích nhau là một khái niệm mới và hết sức thành công. Các đặc điểm của một họ máy như vậy gồm:

- **Tập chỉ thị đồng nhất hay tương tự:** Trong nhiều trường hợp, một tập chỉ thị máy chung được sử dụng cho toàn bộ các thành viên của họ máy. Do vậy, một chương trình nếu có thể thực thi được trên một máy thì cũng sẽ thực thi được trên những máy khác cùng họ với nó. Trong một số trường hợp, thành viên ở mức thấp nhất của họ máy có tập chỉ thị là tập con của tập chỉ thị có trong thành viên ở mức cao nhất, và do vậy chương trình có thể tương thích lên chứ không tương thích xuống.
- **Hệ điều hành đồng nhất hay tương tự:** Một hệ điều hành chung sẽ được sử dụng cho tất cả các thành viên của họ máy. Trong một số trường hợp, một số chức năng phụ sẽ được đưa vào các thành viên mức cao.
- **Gia tăng tốc độ:** Tốc độ thực thi chỉ thị gia tăng từ thành viên mức thấp đến thành viên mức cao trong cùng một họ.
- **Gia tăng số cổng nhập/xuất:** Đi từ thành viên mức thấp đến thành viên mức cao trong cùng một họ.

- **Gia tăng kích thước bộ nhớ:** Đi từ thành viên mức thấp đến thành viên mức cao trong cùng một họ.
- **Gia tăng chi phí:** Đi từ thành viên mức thấp đến thành viên mức cao trong cùng một họ.

Họ máy IBM System/360 không những đã quyết định tương lai về sau của IBM mà còn có một ảnh hưởng sâu sắc đến toàn bộ ngành công nghiệp máy tính. Nhiều đặc trưng của họ máy này đã trở thành tiêu chuẩn cho các máy tính lớn khác.

❑ Máy DEC PDP-8

Trong lúc IBM giới thiệu máy System/360 thì DEC cho ra đời một hiện tượng khác trong ngành công nghiệp máy tính. Đó là máy PDP-8. Vào lúc một máy tính cỡ trung cũng đòi hỏi một phòng có điều hòa không khí, máy PDP-8 đủ nhỏ để có thể đặt trên một chiếc ghế dài vốn thường gặp trong phòng thí nghiệm hoặc để kết hợp vào trong các thiết bị khác. Nó có thể thực hiện mọi công việc của một máy tính lớn với giá chỉ có 16000 đô la Mỹ, so với số tiền lên đến hàng trăm ngàn đô la để mua được một chiếc máy System/360 của IBM.

2.4 MÁY TÍNH THẾ HỆ THỨ TƯ (1974 – HIỆN NAY)

Với tốc độ phát triển nhanh chóng của công nghệ, mức độ cho ra đời các sản phẩm mới ở mức cao, cũng như tầm quan trọng của phần mềm, của truyền thông và phần cứng, việc phân loại máy tính theo thế hệ trở nên kém rõ ràng và ít có ý nghĩa như trước đây. Trong phần tiếp theo, hai thành tựu tiêu biểu về công nghệ của máy tính thế hệ thứ tư sẽ được giới thiệu một cách tóm lược.

❑ Bộ nhớ bán dẫn

Vào khoảng những năm 50 đến 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 (1 inch = 2.54 cm). Các vòng này được treo trên các lưỡi 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. Nhưng 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 bộ nhớ bán dẫn có dung lượng tương đối 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 tám thế hệ: 1K, 4K, 16K, 64K, 256K, 1M, 4M, và giờ đây 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.

❑ 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 lặp lại phép cộng. Theo tiêu chuẩn

ngày nay, chip 4004 rõ ràng quá đơn giản, nhưng nó đã đánh dấu sự bắt đầu của một quá trình tiến hóa liên tục về dung lượng và sức mạnh của các 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 khi đó, Intel giới thiệu bộ vi xử lý 32 bit của riêng mình là chip 80386 vào năm 1985.

CHƯƠNG 3

Đường truyền hệ thống

3.1 CÁC CẤU TRÚC LIÊN KẾT

Một máy tính bao gồm một tập các thành phần hay module thuộc ba kiểu cơ bản (CPU, bộ nhớ, thiết bị nhập xuất) liên lạc với nhau. Trong thực tế, một máy tính có thể được xem như một mạng gồm những thành phần cơ bản. Do đó phải có các đường dẫn nối các module lại với nhau.

Tập hợp các đường dẫn nối kết vô số module được gọi là *cấu trúc liên kết*. Sự thiết kế cấu trúc này sẽ phụ thuộc vào những trao đổi cần được thực hiện giữa các module.

Hình 3.1 đề nghị các kiểu trao đổi cần phải có thông qua việc chỉ ra những dạng nhập xuất chính cho mỗi loại module:

- **Bộ nhớ:** Một cách tiêu biểu, một module bộ nhớ sẽ bao gồm N word có độ dài bằng nhau. Mỗi word được gán cho một địa chỉ dạng số duy nhất $(0, 1, \dots, N-1)$. Một word dữ liệu có thể được đọc từ hay ghi vào bộ nhớ. Bản chất của thao tác sẽ được chỉ ra bởi các tín hiệu điều khiển Đọc và Ghi. Vị trí của thao tác được đặc tả thông qua địa chỉ.
- **Module nhập/xuất:** Nếu nhìn từ quan điểm của một hệ thống máy tính, thành phần nhập xuất giống với bộ nhớ về mặt chức năng. Ở đây có hai thao tác là đọc và ghi. Hơn nữa, một module nhập/xuất có thể điều khiển nhiều hơn một thiết bị ngoại vi. Chúng ta có thể đề cập đến từng giao diện

của một thiết bị ngoại vi như một *cổng* và cho nó một địa chỉ duy nhất (ví dụ 0, 1,..., M-1). Ngoài ra, còn có các đường dữ liệu ngoài cho việc nhập xuất dữ liệu với một thiết bị ngoại vi. Cuối cùng, một module nhập/xuất có thể gửi tín hiệu ngắt đến CPU.

- **CPU:** CPU đọc vào các chỉ thị và dữ liệu, ghi ra dữ liệu sau khi xử lý, và sử dụng các tín hiệu điều khiển để điều phối hoạt động của toàn thể hệ thống. Nó cũng nhận các tín hiệu ngắt.

Danh sách đề cập đến ở trên xác định dữ liệu được trao đổi. Cấu trúc liên kết phải hỗ trợ các kiểu truyền dữ liệu sau đây:

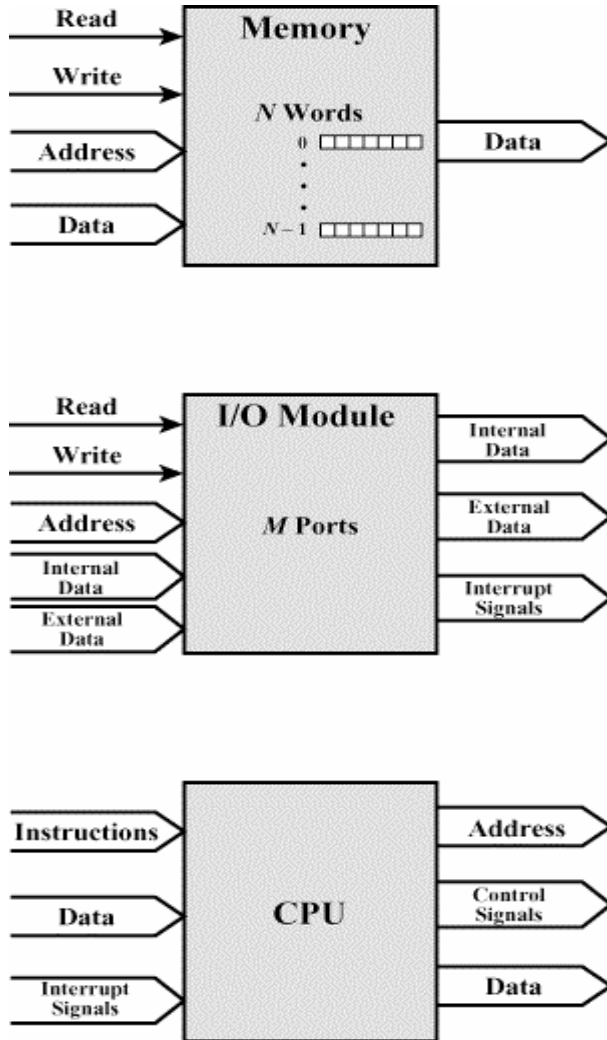
- **Bộ nhớ đến CPU:** CPU đọc một chỉ thị hay một đơn vị dữ liệu từ bộ nhớ.
- **CPU đến bộ nhớ:** CPU ghi một đơn vị dữ liệu vào bộ nhớ.
- **Thành phần nhập/xuất đến CPU:** CPU đọc dữ liệu từ một thiết bị nhập/xuất thông qua một module nhập/xuất.
- **CPU đến thành phần nhập/xuất:** CPU gửi dữ liệu đến thiết bị nhập/xuất.
- **Thành phần nhập/xuất đến hay từ bộ nhớ:** Đối với hai trường hợp này, một module nhập/xuất được cho phép trao đổi dữ liệu trực tiếp với bộ nhớ mà không qua CPU bằng cách sử dụng cơ chế truy cập bộ nhớ trực tiếp (DMA).

Trải qua nhiều năm, một số cấu trúc liên kết đã được thử nghiệm. Cho đến nay phổ biến nhất vẫn là cấu trúc đường truyền (bus) và các cấu trúc đa đường truyền khác nhau.

3.2 LIÊN KẾT ĐƯỜNG TRUYỀN

Một đường truyền là một hành lang liên lạc nối hai hay nhiều thiết bị. Đặc trưng chính của một đường truyền là sự truyền dữ liệu thông qua một phương tiện dùng chung. Nhiều thiết bị nối kết với đường truyền và có thể nhận một tín hiệu được truyền đi từ bất kỳ một thiết bị nào trong hệ thống. Nếu hai thiết bị truyền trong cùng một khoảng thời gian, tín hiệu của chúng sẽ chồng lấp lên nhau và bị làm sai lệch. Do đó chỉ có một thiết bị tại một thời điểm thì mới truyền thành công dữ liệu.

Trong nhiều trường hợp, một đường truyền thật sự bao gồm nhiều hành lang liên lạc, hay còn gọi là các đường. Mỗi đường có khả năng truyền tín hiệu nhị phân 0 và 1. Một dãy các chữ số nhị phân có thể được truyền theo thời gian trên cùng một đường. Khi hợp lại với nhau, nhiều đường của một đường truyền có thể được sử dụng để truyền các chữ số nhị phân cùng một lúc một cách song song. Lấy ví dụ, một đơn vị dữ liệu 8 bit có thể được truyền qua 8 đường của một đường truyền.



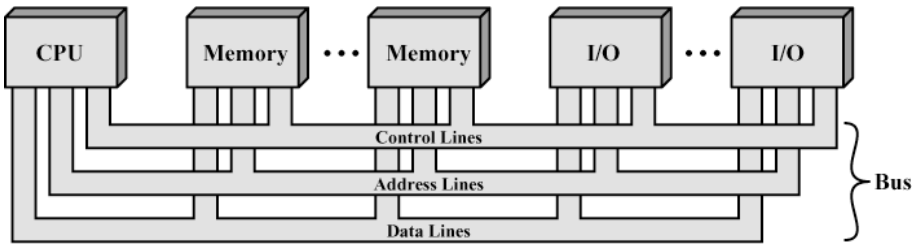
Hình 3.1 Các module máy tính

Các hệ thống máy tính chứa nhiều loại đường truyền khác nhau cung cấp các hành lang giữa các thành phần tại nhiều mức trong sự phân cấp hệ thống máy tính. Một đường truyền kết nối các thành

phần chính của máy tính như CPU, bộ nhớ, thành phần nhập/xuất thì được gọi là một *đường truyền hệ thống*. Các cấu trúc liên kết máy tính phổ biến nhất đều dựa trên việc sử dụng một hay nhiều đường truyền hệ thống.

❑ Cấu trúc đường truyền

Một đường truyền hệ thống thường có từ 50 đến 100 đường riêng biệt. Mỗi đường được gán một ý nghĩa hay chức năng cụ thể. Mặc dù có nhiều kiểu thiết kế đường truyền, trên một đường truyền bất kỳ, các đường có thể được phân thành ba nhóm chức năng như được thể hiện trong hình 3.2, bao gồm dữ liệu, địa chỉ và đường điều khiển. Ngoài ra, có thể có các đường phân phối nguồn điện cho các module nối vào đường truyền.



Hình 3.2 Sơ đồ liên kết đường truyền

Các *đường dữ liệu* cung cấp một đường dẫn cho việc di chuyển dữ liệu giữa các module hệ thống. Những đường này được gọi là *đường truyền dữ liệu*. Đường truyền dữ liệu thường có 8, 16, hay 32 đường riêng biệt, số các đường được đề cập đến với tên gọi *độ rộng* của đường truyền dữ liệu. Vì mỗi đường chỉ có thể mang 1 bit tại một thời điểm, số các đường sẽ xác định số bit có thể truyền đi tại mỗi thời điểm. Độ rộng của đường truyền dữ liệu là một nhân tố quan trọng trong việc xác định hiệu suất hệ thống toàn phần. Chẳng

hạn, nếu đường truyền dữ liệu có độ rộng 8 bit và mỗi chỉ thị có độ dài 16 bit, CPU phải truy cập module bộ nhớ hai lần trong mỗi chu kỳ chỉ thị.

Các *đường địa chỉ* được sử dụng để chỉ định nguồn hay đích của dữ liệu có trên đường truyền dữ liệu. Lấy ví dụ, nếu CPU muốn đọc một word (8, 16, hay 32 bit) dữ liệu từ bộ nhớ, nó sẽ đặt địa chỉ của word cần đọc trên các đường địa chỉ. Rõ ràng là độ rộng của đường truyền địa chỉ sẽ xác định dung lượng bộ nhớ tối đa có thể có của hệ thống. Hơn nữa, các đường địa chỉ nói chung được dùng để định địa chỉ các cổng nhập/xuất. Một cách tiêu biểu, các bit có bậc cao hơn được sử dụng để chọn một module cụ thể trên đường truyền, còn các bit có bậc thấp hơn được dùng để chọn một vị trí nhớ hay cổng nhập/xuất bên trong module đó. Chẳng hạn trên một đường truyền 8 bit, địa chỉ 01111111 và các địa chỉ dưới nó có thể tham chiếu đến các vị trí ở trong một module bộ nhớ (module 0) với 128 word nhớ, còn địa chỉ 10000000 và các địa chỉ trên nó tham chiếu đến các thiết bị gắn vào một module nhập/xuất (module 1).

Các *đường điều khiển* được sử dụng để điều khiển việc truy cập đến và sử dụng các đường dữ liệu cũng như địa chỉ. Vì các đường này được dùng chung bởi tất cả các thành phần, phải có một phương tiện điều khiển việc sử dụng chúng. Các tín hiệu điều khiển truyền cả lệnh lẫn thông tin định thời giữa những module hệ thống. Tín hiệu định thời chỉ ra sự đúng đắn của dữ liệu và thông tin địa chỉ. Tín hiệu lệnh đặc tả thao tác cần được thực hiện. Các đường điều khiển chính gồm:

- **Ghi bộ nhớ:** Làm cho dữ liệu trên đường truyền được ghi vào vị trí đã định địa chỉ.

- **Đọc bộ nhớ:** Làm cho dữ liệu từ vị trí đã định địa chỉ được đặt lên đường truyền.
- **Ghi nhập/xuất:** Làm cho dữ liệu trên đường truyền được xuất ra cổng nhập/xuất đã định địa chỉ.
- **Đọc nhập/xuất:** Làm cho dữ liệu từ cổng nhập/xuất đã định địa chỉ được đặt lên đường truyền.
- **Truyền ACK:** Chỉ ra rằng dữ liệu được chấp nhận từ hay đặt trên đường truyền.
- **Yêu cầu đường truyền:** Chỉ ra rằng một module cần quyền điều khiển đường truyền.
- **Ủy nhiệm đường truyền:** Chỉ ra rằng module đòi hỏi đã được ủy nhiệm quyền điều khiển đường truyền.
- **Yêu cầu ngắt:** Chỉ ra rằng có một ngắt đang chờ xử lý.
- **Ngắt ACK:** Cho biết ngắt treo đã được nhận biết.
- **Đồng hồ:** Được sử dụng để đồng bộ các thao tác.
- **Lấy lại giá trị ban đầu:** Khởi động tất cả module.

Thao tác trên đường truyền như sau:

Nếu có một module cần gửi dữ liệu đến một module khác, nó phải thực hiện hai việc:

- 1) Lấy quyền sử dụng đường truyền
- 2) Truyền dữ liệu qua đường truyền.

Nếu có một module cần yêu cầu dữ liệu từ một module khác, nó phải:

- 1) Lấy quyền sử dụng đường truyền
- 2) Truyền yêu cầu đến module khác thông qua các các đường địa chỉ và điều khiển thích hợp. Sau đó nó phải đợi module thứ hai gửi dữ liệu đến.

Các phân cấp đa đường truyền

Nếu có một lượng lớn các thiết bị được nối vào đường truyền, sự vận hành sẽ bị giảm sút. Có hai lý do chính:

- 1) Một cách tổng quát, khi càng có nhiều thiết bị gắn vào đường truyền thì sự chậm trễ trong lan truyền càng lớn. Khi sự điều khiển của đường truyền đi từ thiết bị này sang thiết bị khác một cách thường xuyên, những sự chậm trễ về lan truyền này có thể ảnh hưởng đáng kể đến sự vận hành.
- 2) Đường truyền có thể trở thành một nút cổ chai gây tắc nghẽn thông tin khi yêu cầu truyền dữ liệu tổng hợp đạt đến mức dung lượng của đường truyền. Bài toán này có thể giải quyết bằng cách gia tăng tốc độ truyền dữ liệu trên đường truyền và sử dụng các đường truyền rộng hơn (ví dụ như tăng đường truyền dữ liệu lên từ 32 đến 64 bit). Tuy nhiên, vì tốc độ truyền dữ liệu sinh bởi các thiết bị đã được nối kết (ví dụ các bộ điều khiển video và đồ họa, các giao tiếp mạng) gia tăng một cách nhanh chóng, đây là một cuộc chạy đua mà người thua cuộc cuối cùng chắc chắn là đường truyền đơn lẻ.

Vì lý do nói trên, hầu hết các hệ thống máy tính đều sử dụng nhiều đường truyền tạo thành một sơ đồ phân cấp. Một cấu trúc truyền thống tiêu biểu được chỉ ra trên hình 3.3 (a). Có một đường truyền cục bộ nối từ bộ xử lý đến một bộ nhớ cache và có thể hỗ trợ nhiều thiết bị cục bộ. Bộ điều khiển bộ nhớ cache không chỉ nối nó với đường truyền cục bộ này, mà còn với cả đường truyền hệ thống, nơi tất cả các module bộ nhớ chính đều được nối vào. Như sẽ được thảo luận trong bài 4, việc sử dụng cấu trúc cache cách ly bộ xử lý ra khỏi yêu cầu truy cập bộ nhớ chính thường xuyên. Do vậy, bộ nhớ chính có thể được di chuyển ra khỏi đường truyền cục bộ

vào trong một đường truyền hệ thống. Bằng cách này, các thành phần nhập/xuất truyền dữ liệu đến và đi khỏi bộ nhớ chính thông qua đường truyền hệ thống sẽ không làm ảnh hưởng đến hoạt động của bộ xử lý.

Chúng ta cũng có thể kết nối trực tiếp các bộ điều khiển nhập/xuất vào đường truyền hệ thống. Một giải pháp hiệu quả hơn là sử dụng các đường truyền mở rộng cho mục đích này. Một giao tiếp đường truyền mở rộng sẽ lưu vào vùng đệm dữ liệu truyền tải giữa đường truyền hệ thống và các bộ điều khiển có trên đường truyền mở rộng. Sự sắp xếp này cho phép hệ thống hỗ trợ một diện rộng các thiết bị nhập/xuất, đồng thời cô lập sự lưu thông bộ nhớ – bộ xử lý khỏi sự lưu thông về nhập/xuất.

Hình 3.3 (a) chỉ ra một số thiết bị nhập/xuất kiểu mẫu có thể được nối vào đường truyền mở rộng. Các nối kết mạng bao gồm các mạng cục bộ (LAN) như mạng 10-Mbps Ethernet và các nối kết sang mạng diện rộng như mạng chuyển mạch gói. SCSI (small computer system interface) bản thân nó đã là một kiểu đường truyền được dùng để hỗ trợ những ổ đĩa cục bộ cùng với các thiết bị ngoại vi khác. Một cổng tuần tự có thể được sử dụng để hỗ trợ máy in hay máy quét.

Cấu trúc đường truyền truyền thống này hoạt động rất hiệu quả, nhưng nó bắt đầu bị phá vỡ khi ngày càng có nhiều thiết bị nhập/xuất xuất hiện. Để đáp ứng những đòi hỏi ngày một tăng này, một cách tiếp cận chung đã được ngành công nghiệp máy tính áp dụng là xây dựng một đường truyền tốc độ cao cho phép tích hợp chặt chẽ với phần còn lại của hệ thống, chỉ đòi hỏi một cầu nối

giữa nó và đường truyền của bộ vi xử lý. Sự sắp xếp này đôi khi được biết đến với tên gọi kiến trúc trung gian.

Hình 3.3 (b) thể hiện một hiện thực hóa tiêu biểu của cách tiếp cận vừa đề cập đến. Một lần nữa, sẽ có một đường truyền cục bộ nối bộ xử lý đến bộ điều khiển cache. Bộ điều khiển này lại được nối vào đường truyền hệ thống có hỗ trợ bộ nhớ chính. Nó sẽ được tích hợp vào trong một cầu nối hay một thiết bị đệm và thiết bị này sẽ được nối vào đường truyền tốc độ cao. Đường truyền này hỗ trợ các kết nối đến các mạng LAN tốc độ cao, chẳng hạn như Giao diện Dữ liệu Phân phối Quang dẫn (FDDI) ở mức 100 Mbps, các bộ điều khiển trạm làm việc video và đồ họa, cũng như những bộ điều khiển giao diện đối với các đường truyền ngoại vi cục bộ bao gồm SCSI và P1394. P1394 là một tổ chức đường truyền tốc độ cao được thiết kế đặc biệt cho các thiết bị nhập/xuất dung lượng cao. Các thiết bị có tốc độ thấp vẫn được hỗ trợ thông qua một đường truyền mở rộng, với một giao diện cho phép tạo vùng đệm cho sự lưu thông giữa đường truyền mở rộng và đường truyền hệ thống.

Lợi ích của cách tiếp cận này là đường truyền tốc độ cao mang các thiết bị có yêu cầu cao lại gần hơn trong mối tích hợp với bộ xử lý và đồng thời độc lập với bộ xử lý. Do đó, các khác biệt trong bộ xử lý, trong đường truyền tốc độ cao, trong các định nghĩa về đường tín hiệu sẽ có khả năng chịu lỗi. Những thay đổi về kiến trúc bộ xử lý không làm ảnh hưởng đến đường truyền tốc độ cao và ngược lại.

Các yếu tố trong thiết kế đường truyền

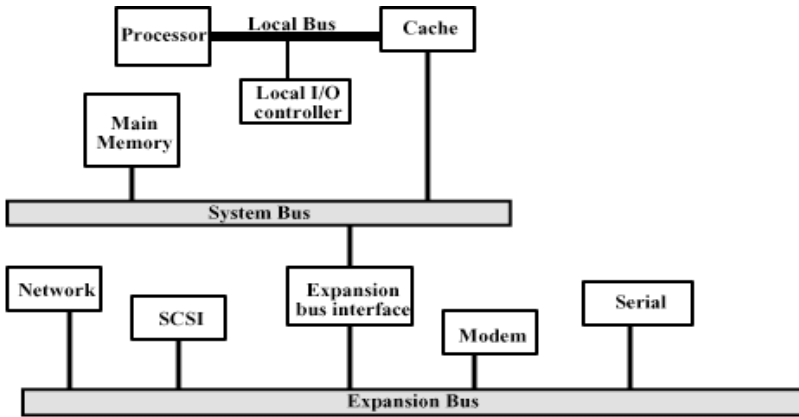
Mặc dù đã có một lượng lớn các cài đặt đường truyền tồn tại, trong thực tế chỉ có một số tham số cơ bản hay yếu tố thiết kế giữ

vai trò phân loại các đường truyền. Cụ thể chúng ta có các yếu tố chính sau:

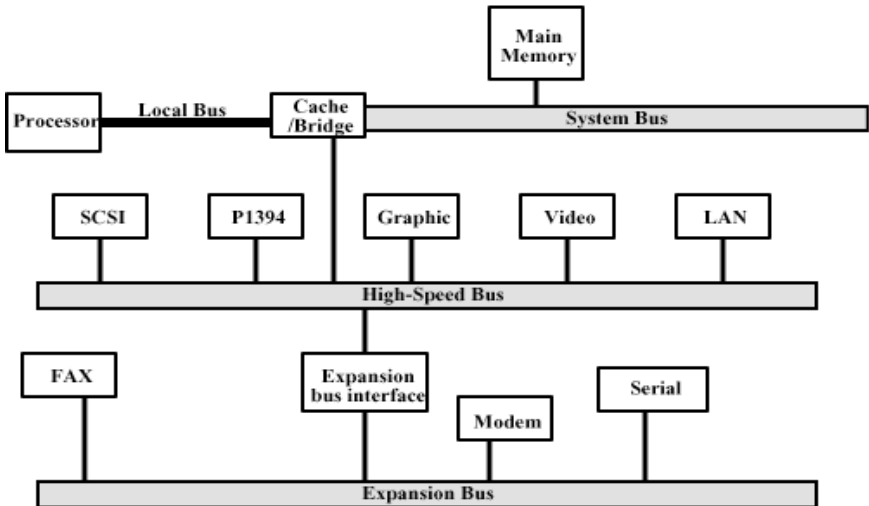
- Kiểu đường truyền
- Phương pháp phân xử
- Định thời
- Độ rộng đường truyền
- Kiểu truyền dữ liệu

Kiểu đường truyền

Có hai kiểu đường truyền tổng quát là đường truyền chuyên dụng và đường truyền đa công. Đường truyền chuyên dụng được gán vĩnh viễn cho một chức năng hay một tập con vật lý của các thành phần máy tính.



(a) Traditional Bus Architecture



(b) High-Performance Architecture

Hình 3.3 Các cấu hình đường truyền tiêu biểu

Một ví dụ cho sự chuyên biệt hóa về chức năng là việc sử dụng các đường dữ liệu và địa chỉ chuyên dụng tách biệt, vốn phổ biến

đối với nhiều đường truyền. Tuy nhiên, đây không phải là điểm cần thiết. Lấy ví dụ, thông tin dữ liệu và địa chỉ có thể được truyền đi trên cùng một tập các đường bằng cách sử dụng đường điều khiển Địa chỉ Đúng. Vào lúc bắt đầu truyền dữ liệu, địa chỉ được đặt lên đường truyền và đường điều khiển Địa chỉ Đúng được kích hoạt. Tại điểm này, mỗi module có một khoảng thời gian đã được đặc tả để sao chép địa chỉ và xác định xem đó có phải là một module đã định địa chỉ hay không. Sau đó địa chỉ sẽ được xóa khỏi đường truyền để các kết nối trên cùng đường truyền đó có thể dùng được cho các thao tác truyền dữ liệu đọc hay ghi. Phương pháp sử dụng cùng một tập các đường cho các mục đích khác nhau này được biết đến với tên gọi đa công theo thời gian.

Lợi ích của tiếp cận đa công theo thời gian là việc sử dụng ít hơn các đường, giúp tiết kiệm không gian và phí tổn. Điểm bất lợi của nó là cần nhiều mạch điện tử phức tạp hơn cho từng module. Ngoài ra sẽ có sự sụt giảm tiềm tàng về hiệu suất vì có một số sự kiện dùng chung các đường như vậy không thể diễn ra một cách song song.

Sự chuyên dụng vật lý đề cập đến việc sử dụng nhiều đường truyền, mỗi đường truyền trong chúng chỉ kết nối một tập con các thành phần. Một ví dụ cụ thể là việc sử dụng đường truyền nhập/xuất để liên kết tất cả các module nhập/xuất. Đường truyền này sau đó được kết nối với đường truyền chính thông qua một loại module bảng mạch nhập/xuất nào đó. Lợi ích tiềm ẩn của sự chuyên dụng vật lý là băng thông cao, vì sẽ có ít tranh chấp đường truyền hơn. Điểm bất lợi ở đây là sự gia tăng kích thước và phí tổn hệ thống.

Phương pháp phân xử

Trong tất cả các hệ thống, ngoại trừ các hệ thống đơn giản nhất, có thể có nhiều hơn một module cần quyền điều khiển đường truyền. Lấy ví dụ, một module nhập/xuất có thể đọc hay ghi trực tiếp vào bộ nhớ mà không cần gửi dữ liệu đến CPU. Vì chỉ có một đơn vị có thể truyền dữ liệu thành công tại mỗi thời điểm trên đường truyền, cần phải có một số các phương pháp phân xử quyền sử dụng. Các phương pháp đã được đưa ra có thể xếp vào một trong hai loại tập trung hoặc phân tán. Trong một sơ đồ tập trung, một thiết bị phần cứng đơn lẻ, được gọi là *bộ điều khiển đường truyền* hay *bộ phân xử*, có nhiệm vụ cấp phát thời gian trên đường truyền. Thiết bị đó có thể là một module riêng biệt hay một phần của CPU. Trong một sơ đồ phân tán, chúng ta sẽ không có bất kỳ bộ điều khiển tập trung nào. Mỗi module chứa luận lý điều khiển truy cập và các module sẽ hành động phối hợp với nhau nhằm sử dụng chung đường truyền. Với cả hai phương pháp nói trên về phân xử, mục tiêu đạt đến là chỉ định một thiết bị, hoặc CPU, hoặc module nhập/xuất, là thiết bị chủ. Thiết bị chủ này sau đó có thể khởi động việc truyền dữ liệu (đọc hay ghi) với một thiết bị khác, hoạt động như thiết bị phụ thuộc trong sự trao đổi dữ liệu đặc biệt này.

Định thời

Định thời đề cập đến cách thức các sự kiện được phối hợp với nhau trên đường truyền. Với sự định thời đồng bộ, việc xuất hiện các sự kiện trên đường truyền được xác định bởi một đồng hồ. Đường truyền có một đường đồng hồ trên đó đồng hồ sẽ truyền một dãy các giá trị luân phiên 0 và 1 trong những thời khoảng bằng nhau. Một phiên truyền 0 – 1 đơn được gọi là *chu kỳ đồng hồ* hay *chu kỳ đường truyền* và xác định một khe thời gian. Tất cả các thiết bị khác trên đường truyền có thể đọc đường đồng hồ, và mọi sự

kiện đều được bắt đầu tại lúc bắt đầu của một chu kỳ đồng hồ. Hình 3.4 (a) thể hiện sơ đồ định thời của một thao tác đọc đồng bộ. Các tín hiệu đường truyền khác có thể thay đổi tại cạnh dẫn của tín hiệu đồng hồ. Hầu hết các sự kiện chiếm chỉ một chu kỳ đồng hồ. Trong ví dụ này, CPU tạo ra một tín hiệu đọc và đặt địa chỉ bộ nhớ lên đường truyền địa chỉ. Nó cũng tạo ra một tín hiệu bắt đầu nhằm đánh dấu sự có mặt của thông tin địa chỉ và điều khiển trên đường truyền. Một module bộ nhớ nhận biết địa chỉ đó và sau khi chờ khoảng 1 chu kỳ, nó sẽ đặt dữ liệu cùng tín hiệu thông báo lên đường truyền.

Với sự định thời dị bộ, việc xuất hiện một sự kiện trên đường truyền theo sau và phụ thuộc vào sự xuất hiện của sự kiện đi trước. Như trong hình 3.4 (b), CPU đặt địa chỉ và các tín hiệu đọc lên đường truyền. Sau khi tạm dừng lại để các tín hiệu này ổn định, nó sẽ sinh ra một tín hiệu MSYN (master sync), chỉ ra sự hiện hữu của các tín hiệu điều khiển và địa chỉ đúng. Module bộ nhớ đáp ứng bằng tín hiệu SSYN (slave sync) và dữ liệu, chỉ ra sự trả lời.

Định thời đồng bộ là phương pháp dễ cài đặt và kiểm tra hơn. Tuy nhiên, nó kém linh hoạt so với định thời dị bộ. Do tất cả các thiết bị trên một đường truyền đồng bộ được kết buộc vào một tốc độ đồng hồ cố định, hệ thống không thu được lợi ích trong việc vận hành thiết bị. Với định thời dị bộ, sự trộn lẫn giữa các thiết bị chậm và nhanh bằng cách sử dụng các công nghệ cũ và mới có thể sử dụng chung một đường truyền.

Độ rộng đường truyền

Độ rộng của đường truyền dữ liệu có ảnh hưởng đến sự vận hành của hệ thống: đường truyền dữ liệu càng rộng, số bit được truyền đi tại một thời điểm càng lớn. Độ rộng của đường truyền địa chỉ lại có

ảnh hưởng đến dung lượng hệ thống: độ rộng của đường truyền địa chỉ càng lớn, lượng vị trí nhớ có thể tham chiếu đến càng nhiều.

Kiểu truyền dữ liệu

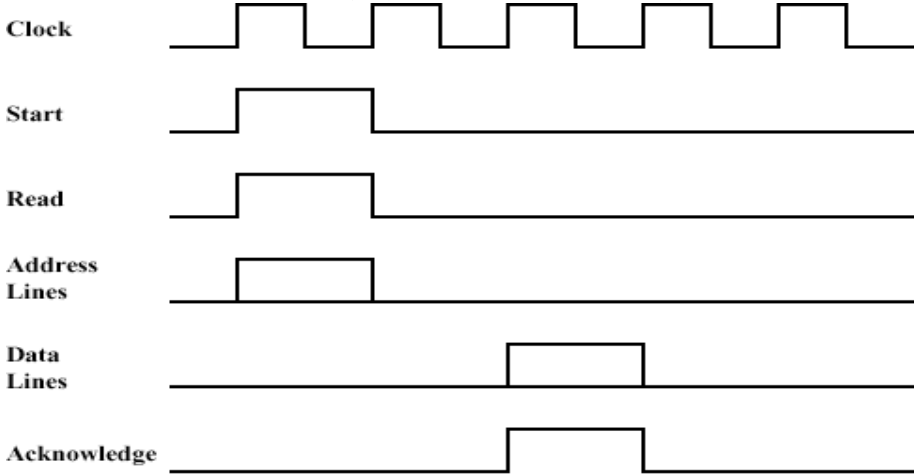
Một đường truyền có thể hỗ trợ nhiều kiểu truyền dữ liệu khác nhau. Tất cả các đường truyền đều hỗ trợ việc truyền dữ liệu dạng ghi (thiết bị chủ đến thiết bị phụ thuộc) và đọc (thiết bị phụ thuộc đến thiết bị chủ). Trong trường hợp đường truyền dữ liệu/địa chỉ đa công, đường truyền được dùng trước hết cho việc đặc tả địa chỉ và sau đó cho việc chuyển dữ liệu. Với thao tác đọc, thường có khoảng thời gian chờ khi dữ liệu được lấy ra từ thiết bị phụ thuộc và đặt lên đường truyền. Với cả đọc lẫn ghi, cũng có thể có độ trễ nếu như cần phải làm việc với bộ phân xử để giành quyền điều khiển đường truyền cho phần còn lại của thao tác.

Trong trường hợp đường truyền dữ liệu và địa chỉ chuyên dụng, địa chỉ được đặt lên đường truyền địa chỉ và tồn tại ở đó trong khi dữ liệu được đặt lên đường truyền dữ liệu. Với thao tác ghi, thiết bị chủ đặt dữ liệu lên đường truyền địa chỉ ngay khi địa chỉ ổn định và thiết bị phụ thuộc khi đó sẽ có cơ hội để nhận biết địa chỉ. Với thao tác đọc, thiết bị phụ thuộc đặt dữ liệu lên trên đường truyền dữ liệu ngay khi nó nhận ra địa chỉ và lấy được dữ liệu.

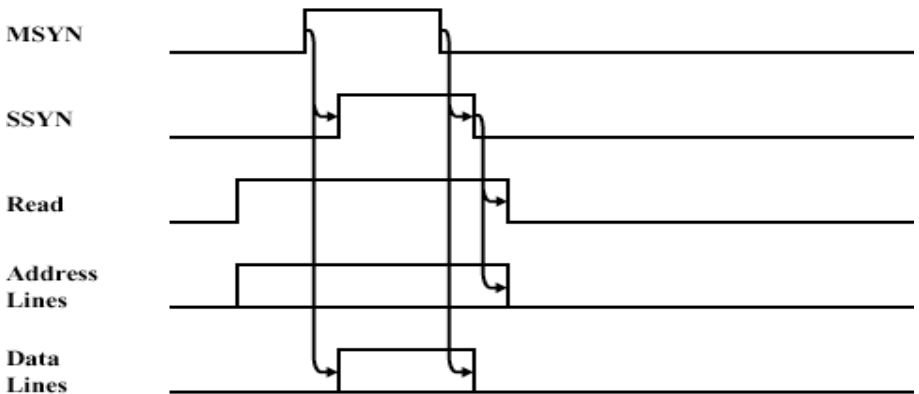
Ngoài các kiểu thao tác nói trên, một số đường truyền còn hỗ trợ các thao tác phối hợp. Một thao tác đọc – chỉnh sửa – ghi đơn giản là một phép đọc tiếp theo bằng một phép ghi trên cùng một địa chỉ. Địa chỉ chỉ được phát đi một lần tại lúc bắt đầu thao tác. Toàn bộ thao tác không thể chia nhỏ nhằm tránh mọi sự truy cập đến phần tử dữ liệu đang được chỉnh sửa.

Một số hệ thống đường truyền cũng hỗ trợ việc truyền dữ liệu dạng khối. Trong trường hợp này, một chu kỳ địa chỉ được theo sau bởi n chu kỳ dữ liệu. Mục dữ liệu đầu tiên được truyền đến hoặc từ một địa chỉ đã được đặc tả, phần dữ liệu còn lại được truyền đến/từ các địa chỉ tiếp sau.

Hình 3.4 Sự định thời của thao tác đọc



(a) Synchronous timing



(b) Asynchronous timing

Chương II

TỔ CHỨC BỘ NHỚ TRONG MÁY VI TÍNH

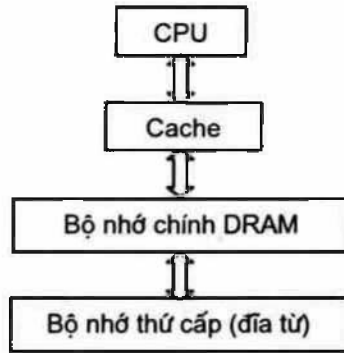
1. TỔ CHỨC BỘ NHỚ THEO PHÂN CẤP

Một trong các chức năng của CPU là tuân tự nhập lệnh từ bộ nhớ và thực hiện lệnh. Tốc độ truy nhập bộ nhớ chính (bộ nhớ thao tác) tương đối chậm (với DRAM là khoảng $100\text{ns} = 1.10^{-7}\text{s}$) so với khả năng của CPU. Nếu CPU nhập các lệnh và dữ liệu trực tiếp từ bộ nhớ chính thì tốc độ xử lý thực của CPU phụ thuộc vào tốc độ truy nhập của bộ nhớ chính. Mặt khác dung lượng nhớ của bộ nhớ chính cũng khá hạn chế, do vậy nếu hệ thống lưu trữ chỉ gồm bộ nhớ chính cũng sẽ hạn chế khả năng của CPU.

Để tăng tốc độ xử lý của CPU đồng thời đảm bảo khả năng lưu trữ lớn của bộ nhớ máy tính 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 khoảng thì các lệnh và dữ liệu được sử dụng thường đều nằm ở một khu vực tương đối nhỏ trong bộ nhớ chính. Các vùng này luôn chuyển dịch khi chạy chương trình. Cơ sở của việc quản lý hệ thống bộ nhớ có phân cấp, mà 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 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 thông tin cần có trong cache. Giải pháp cho vấn đề kích thước hạn chế của bộ nhớ chính là sử dụng bộ nhớ ngoài (thiết bị đĩa từ) như là một thành phần của hệ thống bộ nhớ (h.42).



Hình 42

2. TỔ CHỨC CACHE

Thủ tục quản lý bộ nhớ phân cấp có cache là nạp từng đoạn chương trình và dữ liệu từ bộ nhớ chính vào cache. Phương pháp đơn giản nhất được dùng để quy chiếu bộ nhớ có cache là phương pháp ánh xạ trực tiếp.

Việc quy chiếu đến cache được gọi là “trúng” (hit) nếu truy nhập được thông tin (nằm trong ô nhớ có địa chỉ CPU đòi hỏi) ở trong cache, và gọi là “trượt” (miss) nếu không truy nhập được thông tin ở trong cache và phải đọc từ bộ nhớ chính.

Bộ nhớ chính được chia thành nhiều khối, mỗi khối bao gồm nhiều từ hoặc nhiều byte (h.43).

| | | Địa chỉ thấp |
|--------|--------------|--------------|
| Khối 0 | 1 từ (1byte) | 00 |
| Khối 1 | 1 từ (1byte) | 01 |
| Khối 2 | 1 từ (1byte) | 10 |
| Khối 3 | 1 từ (1byte) | 11 |
| Khối 4 | 1 từ (1byte) | 00 |
| Khối 5 | 1 từ (1byte) | 01 |
| Khối 6 | 1 từ (1byte) | 10 |
| ⋮ | ⋮ | ⋮ |

Hình 43

Bộ nhớ cache có thể chứa nhiều khối, gọi là khối cache. Mỗi khối cache nằm ở một vị trí xác định trong cache. Mỗi khối cache chứa các thông tin như : số hiệu thẻ, bit cờ và bản thân khối dữ liệu (khối dữ liệu trong cache là bản sao của khối dữ liệu trong bộ nhớ chính).

Cấu trúc một khối cache như sau :

| | | |
|-------------|---|--------------|
| Số hiệu thẻ | F | Khối dữ liệu |
|-------------|---|--------------|

Giả thiết bộ nhớ có 2^{24} ô nhớ. Nếu cache có 2^2 vị trí chứa (khối cache) thì số lượng thẻ sẽ là $2^{24}/2^2 = 2^{22}$ (\approx 4triệu thẻ).

2.1 Trường hợp mỗi khối chứa một từ (hoặc 1 byte) dữ liệu

a) Thao tác đọc bộ nhớ

Giả sử có bộ nhớ chính có BUS địa chỉ 24 bit và cache có 4 khối.

Khi một từ (1byte) dữ liệu được đọc thì CPU cung cấp địa chỉ cho bộ điều khiển bộ nhớ. Bộ điều khiển bộ nhớ tách địa chỉ 24 bit làm hai phần (h.44) :

2 bit địa chỉ thấp nhất được đặt vào thanh ghi địa chỉ MAR của cache. Các bit này xác định vị trí (thứ tự) khối cần tìm trong cache.

22 bit địa chỉ cao mô tả số hiệu thẻ. Số hiệu thẻ là con số xác định vị trí của khối nhớ trong bộ nhớ.

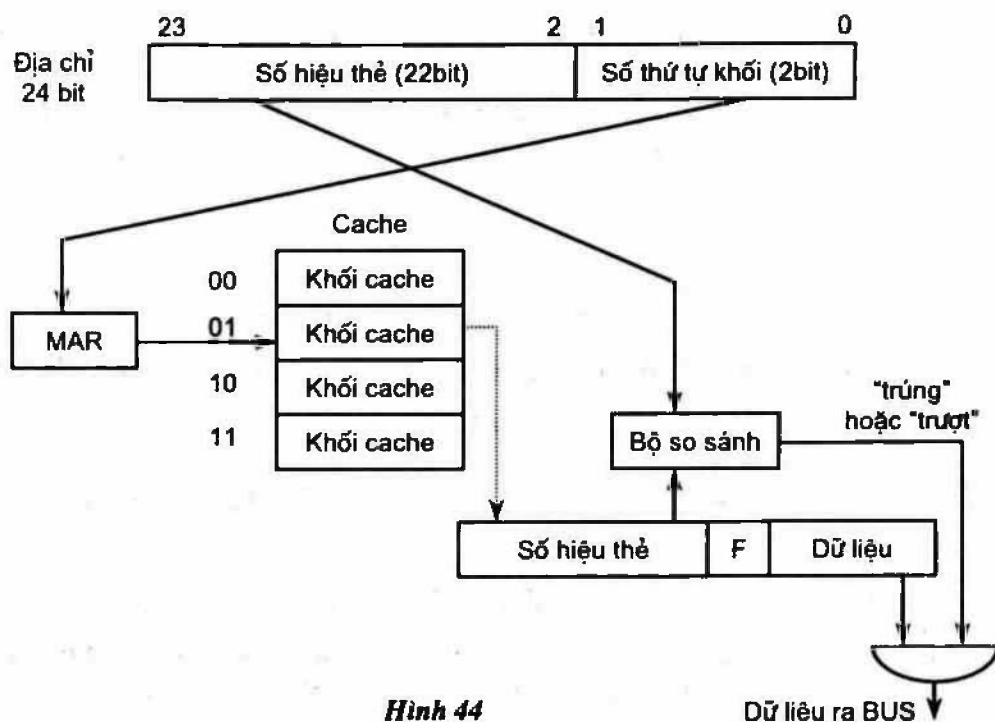
Thao tác đọc dữ liệu từ bộ nhớ được tiến hành như sau :

Bước 1 : bộ điều khiển cache đọc khối dữ liệu trong cache tại vị trí có số thứ tự trùng với phần số thứ tự khối trong địa chỉ trên BUS.

Bước 2 : bộ điều khiển cache xác định xem số hiệu thẻ của khối trong cache này có trùng với số hiệu thẻ trong địa chỉ BUS hay không.

Bước 3 : Nếu trùng thì việc quy chiếu là “trúng” và một từ dữ liệu được đọc từ cache vào CPU. Nếu không trùng (trường hợp “trượt”) thì từ dữ liệu phải được lấy từ bộ nhớ chính. Trong trường hợp “trượt” cần phải sao lưu dữ liệu trong cache vào bộ nhớ chính theo địa chỉ của nó, sau đó mới nạp dữ liệu mới cùng với thẻ của nó vào cache và đặt bit cờ $F = 0$ (ghi nhận nội dung cache trùng với nội dung bộ nhớ chính) và dữ liệu được cung cấp cho CPU.

Số thẻ cache trong các khối cache có thể trùng nhau (khi khối cache chưa bị thay thế) và có thể khác nhau (khi nội dung khối cache bị thay).



Hình 44

b) Thao tác ghi bộ nhớ

Có một vài kỹ thuật được dùng để ghi dữ liệu vào bộ nhớ chính khi thực hiện các lệnh ghi bộ nhớ : kỹ thuật ghi xuyên (write through) và kỹ thuật sao lưu (copy back).

+ Ở loại cache ghi xuyên (write through) dữ liệu được ghi lên cả cache lẫn bộ nhớ chính cùng một lúc, không dùng đến bit cờ F. Kỹ thuật này làm cho thời gian ghi bộ nhớ tăng lên.

+ Ở loại cache sao lưu (copy back) thì dữ liệu chỉ được ghi vào cache và bit cờ F được lập (F = 1), ghi nhận nội dung cache khác với nội dung bộ nhớ chính. Sau đó nếu khối dữ liệu cần được thay thế bằng khối dữ liệu khác từ bộ nhớ chính (trường hợp "trượt") thì bit cờ F được kiểm tra để xác định xem có cần thực hiện thao tác sao lưu này không, nếu F = 1 thì cần thực hiện sao lưu, nếu F = 0 thì không cần sao lưu. Kỹ thuật này làm tăng tốc độ thao tác với bộ nhớ và được gọi là kỹ thuật sao lưu có dựng cờ.

2.2 Trường hợp mỗi khối chứa nhiều từ dữ liệu

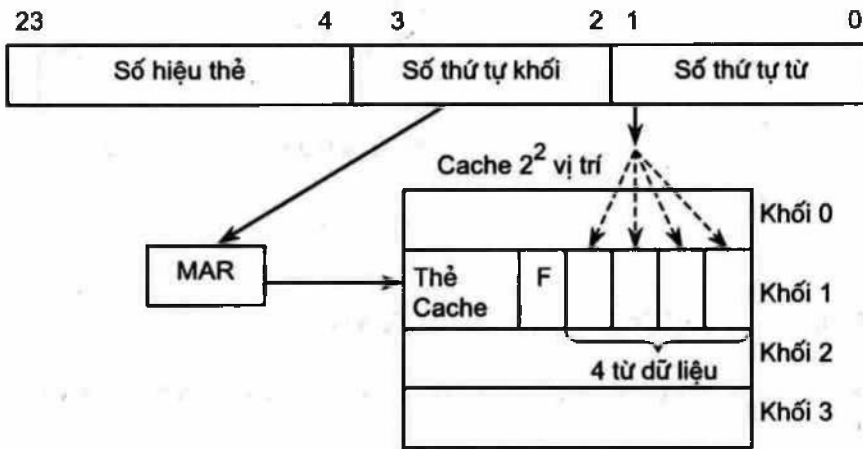
Khi có nhiều từ dữ liệu trong một khối thì kỹ thuật ánh xạ trực tiếp phức tạp hơn. Giả thiết địa chỉ là 24 bit và một khối gồm 4 từ (hoặc byte) dữ liệu. Địa chỉ trên BUS sẽ được tách thành 3 phần (h.45) :

Phần xác định số thứ tự từ (hoặc byte) trong một khối, ví dụ là 2 bit. Các bit này xác định vị trí từ dữ liệu cần truy nhập nằm trong khối cache.

Phần xác định số thứ tự khối, ví dụ là 2.

Phần số hiệu thẻ, ví dụ là 20 bit.

Các thao tác khác được thực hiện tương tự như đã trình bày ở trên.



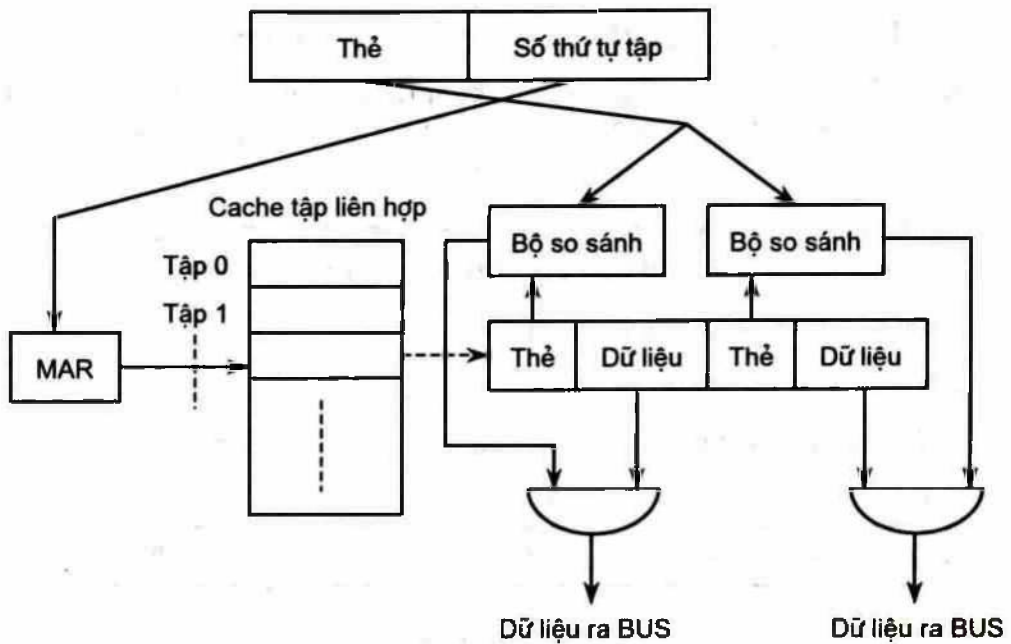
Hình 45

2.3 Kỹ thuật tập liên hợp

Vấn đề tốc độ lại được đặt ra khi CPU liên tục nhập 2 (hoặc 4) lệnh có cùng địa chỉ khối nhớ (các bit địa chỉ thấp của các khối nhớ trùng nhau).

Cách giải quyết là chứa 2 (hay 4) khối nhớ nói trên tại cùng một vị trí (một khối) trong cache, nhưng mỗi khối có một thẻ riêng. Khi vị trí này trong cache được truy nhập thì cả 2 (hay 4) thẻ trong 2 khối con (hoặc 4 khối con) cùng được kiểm tra. Các khối con này tạo thành một tập liên hợp (h.46).

Kỹ thuật đặt nhiều (2 hoặc 4) khối vào cùng một vị trí trong cache được gọi là kỹ thuật tập liên hợp



Hình 46

3. TỔ CHỨC BỘ NHỚ MÁY VI TÍNH PC

Có hai cách tổ chức quản lý bộ nhớ : tổ chức theo phân đoạn và tổ chức tuyến tính.

Ở cách *tiếp cận theo phân đoạn* chương trình và dữ liệu được người lập trình chia thành các modul. Mỗi modul có một không gian địa chỉ riêng. Mỗi modul được gọi là một đoạn và được gán một con số, gọi là địa chỉ đoạn. Mỗi ô nhớ trong đoạn được xác định bởi địa chỉ logic gồm hai con số : địa chỉ đoạn và địa chỉ offset (địa chỉ lệch). Các modul được nạp vào bộ nhớ và được quản lý theo cơ chế phân đoạn.

Ở cách *tiếp cận tuyến tính*, một bộ nhớ với không gian địa chỉ rất lớn (ảo) được cung cấp cho chương trình. Địa chỉ này là địa chỉ tuyến tính và bắt đầu từ 0. Chương trình được chia thành các khúc nhỏ có kích thước cố định, mỗi khúc này được gọi là một trang, và được nạp vào bộ nhớ vật lý. Mỗi chương trình được hệ điều hành quản lý bằng một bảng trang. Các khúc chương trình chứa trong bộ nhớ được quản lý theo cơ chế phân trang. Nhiệm vụ đang chạy tại một thời điểm chỉ cần một số ít trang chương trình và dữ liệu trong bộ nhớ

vật lý, nên kỹ thuật quản lý bộ nhớ theo trang là một kỹ thuật thích hợp cho việc tổ chức bộ nhớ ảo.

Người ta có thể kết hợp cả hai cách nói trên để quản lý bộ nhớ.

3.1 Cơ chế quản lý bộ nhớ theo phân trang

Các CPU x86 32 bit có phần cứng hỗ trợ quản lý bộ nhớ trong chế độ bảo vệ theo cả hai cơ chế phân đoạn và phân trang. Cơ chế quản lý bộ nhớ theo phân đoạn đã được trình bày ở chương 1. Trong phần này ta chỉ xét cơ chế quản lý bộ nhớ theo phân trang.

Đơn vị quản lý bộ nhớ chuyển địa chỉ logic gồm địa chỉ nền đoạn 32 bit và địa chỉ offset 32 bit thành địa chỉ tuyến tính.

Nếu chọn cách quản lý theo phân đoạn thì địa chỉ tuyến tính sẽ là địa chỉ vật lý.

Nếu chọn cách quản lý theo phân trang thì đơn vị quản lý bộ nhớ sẽ chuyển địa chỉ tuyến tính thành địa chỉ vật lý theo cơ chế quản lý theo phân trang.

Trong cơ chế quản lý theo trang, chương trình được chia thành các khúc nhỏ có kích thước xác định, được gọi là trang. Mỗi trang chương trình được gắn vào từng đoạn nhỏ của bộ nhớ, mỗi đoạn nhỏ chứa trang được gọi là một khung trang.

Hệ điều hành quản lý quá trình thực hiện chương trình bằng bảng trang. Bảng trang chỉ ra vị trí khung trang (địa chỉ nền) của mỗi trang chương trình. CPU dựa vào vị trí khung trang và địa chỉ offset của ô nhớ trong trang để tính ra địa chỉ vật lý của ô nhớ.

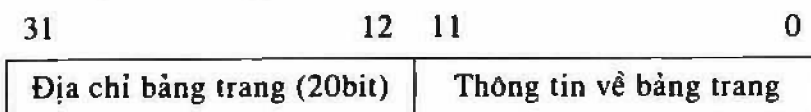
Cấu trúc của hệ thống quản lý trang gồm 3 phần : thư mục trang (Page Directory), bảng trang (Page Table) và trang (Page). Thư mục trang, bảng trang và trang đều có cùng một kích thước là 4Kbyte.

Thư mục trang :

Thư mục trang có kích thước 4Kbyte. Thư mục trang chứa tới 1024 Lối vào thư mục trang PDE (Page Directory Entry).

PDE có kích thước 32 bit (4 byte). Mỗi PDE chứa 20 bit địa chỉ nền bảng trang (trỏ đến nền bảng trang). Mỗi một thư mục trang có thể quản lý được tới 1024 bảng trang.

Cấu trúc một PDE trong thư mục trang :



Phần Địa chỉ bảng trang chứa 20 bit cao của địa chỉ nền bảng trang (địa chỉ bắt đầu một bảng trang), 12 bit thấp của địa chỉ nền bảng trang luôn bằng 0.

Phần Thông tin về bảng trang chứa các thông tin về quyền truy nhập bảng trang và một số thuộc tính khác của trang. Các bit 1 và 2 trong mỗi PDE cung cấp mức đặc quyền và quyền ghi/đọc v.v. của tất cả các trang trở bởi PDE này.

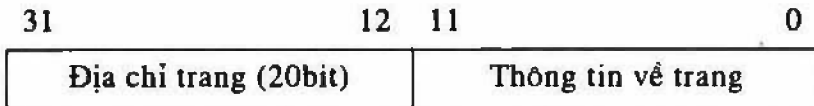
Trong cơ chế quản lý bộ nhớ theo trang, 10 bit cao (từ bit 22 đến bit 31) của địa chỉ tuyến tính được dùng để xác định vị trí của PDE trong thư mục trang (vị trí so với nền thư mục trang).

+ Bảng trang :

Bảng trang có kích thước 4 Kbyte. Bảng trang chứa 1024 Lối vào bảng trang PTE (Page Table Entry).

PTE có kích thước 32 bit (4 byte). Mỗi PTE chứa địa chỉ nền của một trang 4Kbyte, do vậy một bảng trang có thể quản lý được tới 1024 trang.

Cấu trúc một PTE :



Phần Địa chỉ trang chứa 20 bit cao của địa chỉ vật lý nền trang (địa chỉ vật lý bắt đầu một trang), 12 bit thấp của địa chỉ nền trang luôn bằng 0.

Phần Thông tin về trang chứa các thông tin về quyền truy nhập trang và một số thuộc tính khác của trang. Các bit 1 và 2 trong mỗi PTE cung cấp mức đặc quyền và quyền ghi/đọc v.v. của tất cả các trang trở bởi PTE này.

Trong cơ chế quản lý bộ nhớ theo trang, 10 bit của địa chỉ tuyến tính (từ bit 12 đến bit 21) được dùng để xác định vị trí của PTE trong bảng trang (vị trí so với nền bảng trang).

Có hai mức đặc quyền được dùng để bảo vệ trang : mức chương trình ứng dụng (tương đương mức 3 trong quản lý theo đoạn) và mức chương trình giám sát (tương đương mức 0, 1, 2 trong quản lý theo đoạn).

Mức thấp nhất trong hai mức của PDE và PTE được lấy làm mức đặc quyền và quyền ghi/đọc của từng trang.

Địa chỉ vật lý của ô nhớ trong trang được tính bằng cách kết hợp 20 bit cao (phần Địa chỉ trang) trong PTE với 12 bit thấp nhất trong địa chỉ tuyến tính.

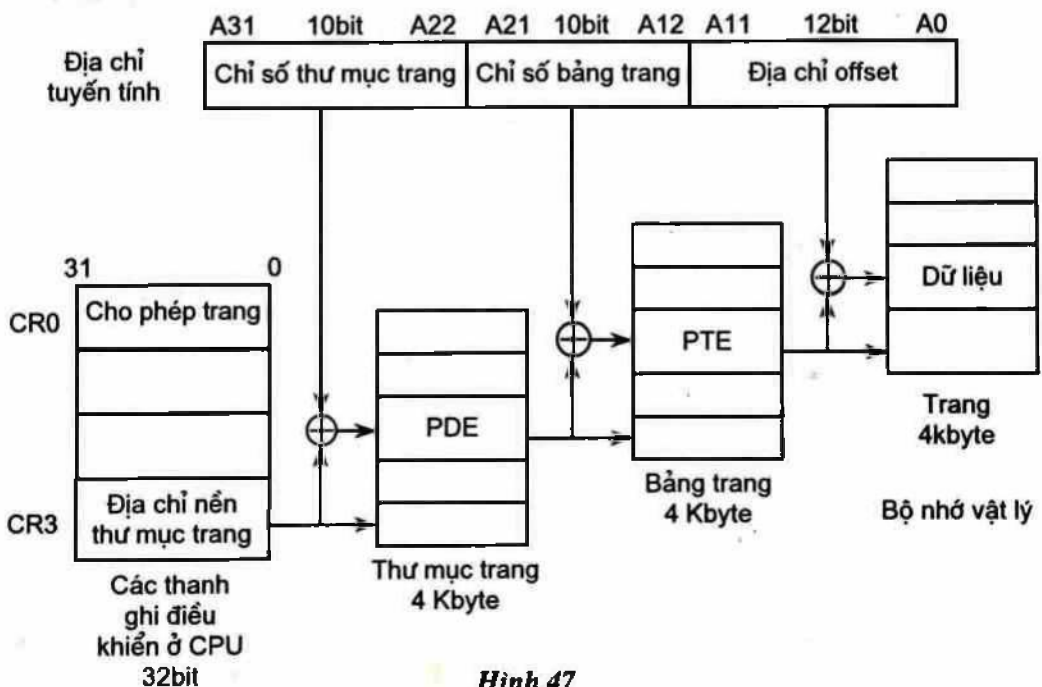
3.2 Cơ chế truy nhập bộ nhớ theo trang

Các máy tính PC loại 32 bit quản lý bộ nhớ theo cách kết hợp hai cơ chế phân đoạn và phân trang. Để truy nhập một ô nhớ CPU cần có một địa chỉ logic, gồm địa chỉ nền đoạn (nhận được từ bộ mô tả đoạn theo cơ chế đã được trình bày ở mục 2.5.2 của Chương 1) và địa chỉ offset. Địa chỉ logic này được chuyển thành địa chỉ tuyến tính. Địa chỉ tuyến tính này được dùng để tính địa chỉ vật lý và truy nhập ô nhớ theo cơ chế phân trang (h.47).

Khi truy nhập theo cơ chế phân trang địa chỉ tuyến tính được tách thành 3 phần :

- + A31 ÷ A22 : 10 bit chỉ số PDE, xác định vị trí PDE trong thư mục trang.
- + A21 ÷ A12 : 10 bit chỉ số PTE, xác định vị trí PTE trong bảng trang.
- + A11 ÷ A0 : là 12 bit địa chỉ offset của ô nhớ trong trang.

Địa chỉ vật lý của ô nhớ trong trang được tính bằng cách kết hợp 20 bit cao (phần địa chỉ trang) trong PTE với 12 bit thấp (A11 ÷ A0) nằm trong địa chỉ tuyến tính này.



Hình 47

Chương III

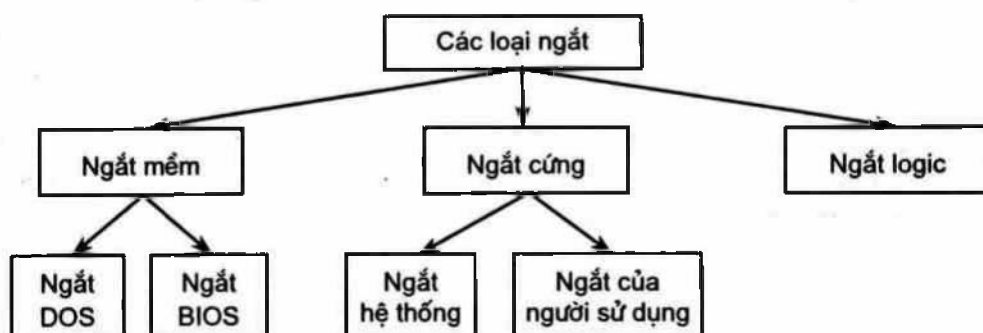
CÁC THIẾT BỊ ĐIỀU KHIỂN VÀ GIAO DIỆN VÀO – RA DỮ LIỆU

1. NGẮT VÀ BỘ ĐIỀU KHIỂN NGẮT PIC 8259

1.1 Một số khái niệm

Ngắt là sự kiện CPU bị tạm dừng việc thực hiện quá trình chính và chuyển sang thực hiện quá trình phục vụ ngắt. Ngắt cứng là phương pháp vào/ra dữ liệu, trong đó thiết bị vào/ra (thiết bị vật lý : bàn phím, máy in, đồng hồ nhịp thời gian v.v.) chủ động khởi động quá trình vào/ra. Quá trình phục vụ ngắt cứng được kích hoạt bằng một tín hiệu vật lý từ bên ngoài.

1.2 Phân loại ngắt



Thuật ngữ “ ngắt” xuất phát từ kỹ thuật ngắt cứng. Khi nói đến ngắt cứng, ngắt mềm hoặc ngắt logic (ngoại lệ) là *hàm ý nói đến các chương trình con phục vụ hoạt động của hệ thống máy tính và nói đến cách kích hoạt các chương trình con này*. Tất cả các chương trình phục vụ ngắt đều có chung đặc điểm : thứ nhất là hầu hết đã được viết sẵn (là các chương trình của hệ điều hành) và được phép sử dụng ; thứ hai là địa chỉ của các chương trình con này

phải được đặt ở một vùng xác định là *Bảng véc tơ ngắt*, nằm trong bộ nhớ chính. Các chương trình con phục vụ ngắt cứng thường được dùng để điều khiển quá trình vào/ra với các thiết bị vào-ra chuẩn ở mức vật lý. Các chương trình con phục vụ ngắt cứng được kích hoạt bởi các *tín hiệu vật lý IRQ* đến từ thiết bị vào-ra. Các chương trình con phục vụ ngắt mềm là các chương trình hệ thống thực hiện các thao tác vào-ra cơ bản ở mức logic và các hoạt động khác của hệ thống. Các chương trình con phục vụ ngắt mềm được kích hoạt bởi *lệnh INT* trong hệ lệnh của CPU. Các chương trình con phục vụ ngắt logic cũng phục vụ cho hoạt động của hệ thống, nhưng chúng chỉ được kích hoạt khi CPU thực hiện lệnh và phát sinh một *ngoại lệ* nào đó.

- Bảng véc tơ ngắt

Bảng véc tơ ngắt là bảng chứa địa chỉ của các chương trình phục vụ ngắt. Bảng này có 256 ô, các ô được đánh số thứ tự lần lượt từ 00h, 01h, ... , 08h, ... , 0Fh, 10h, ... , FFh. Số thứ tự của từng ô trong bảng được gọi là *số ngắt*. Mỗi ô chứa địa chỉ logic của một chương trình phục vụ ngắt xác định, các địa chỉ này còn được gọi là *véc tơ ngắt*.

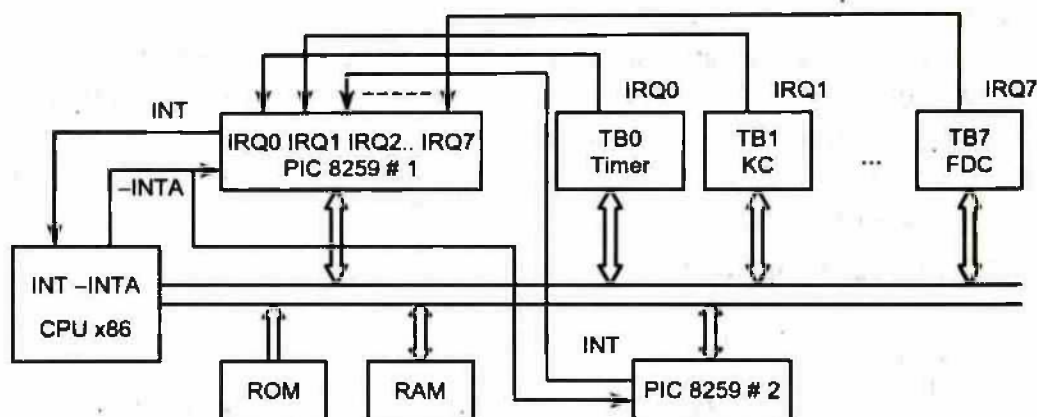
Cấu trúc của Bảng véc tơ ngắt ở chế độ thực :

| | Số t.t. | Số ngắt | Chức năng của chương trình |
|--------------------------------|---------|---------|--|
| Địa chỉ đoạn Địa chỉ offset | 00h | 00h | Xử lý chia cho 0 |
| Địa chỉ đoạn Địa chỉ offset | 01h | 01h | Thực hiện gỡ rối (debug) |
| ⋮ | | | |
| Địa chỉ đoạn Địa chỉ offset | 08h | 08h | Đồng hồ hệ thống |
| Địa chỉ đoạn Địa chỉ offset | 09h | 09h | Phục vụ bàn phím |
| ⋮ | ⋮ | ⋮ | ⋮ |
| Địa chỉ đoạn Địa chỉ offset | 21h | 21h | Thực hiện các dịch vụ của hệ điều hành |
| ⋮ | ⋮ | ⋮ | ⋮ |
| Địa chỉ đoạn Địa chỉ offset | FFh | FFh | Dự phòng |

1.3 Hệ thống ngắt cứng

CPU được thiết kế để đáp ứng được với các quá trình ngắt cứng. CPU có một đầu vào nhận tín hiệu ngắt INT, khi nhận được tín hiệu này CPU sẽ phản ứng theo cơ chế ngắt cứng. Trong thực tế có nhiều thiết bị ngoại vi yêu cầu được phục vụ theo phương pháp ngắt cứng (bàn phím, đồng hồ hệ thống, máy in, v.v.) và sinh ra nhiều yêu cầu ngắt, do vậy cần có một bộ điều khiển giúp CPU quản lý và phục vụ các yêu cầu ngắt, đó là bộ điều khiển ngắt PIC (Programmable Interrupt Controller).

Cấu trúc hệ thống ngắt cứng (h.48) :



Hình 48

Hệ thống ngắt cứng được xây dựng trên cơ sở 2 bộ điều khiển ngắt PIC 8259, mỗi PIC 8259 có thể nhận 8 tín hiệu yêu cầu ngắt IRQ từ thiết bị vào/ra. Hai PIC này được kết nối với nhau theo kiểu ghép tầng, kết hợp hoạt động để có thể phục vụ được 16 yêu cầu ngắt IRQ.

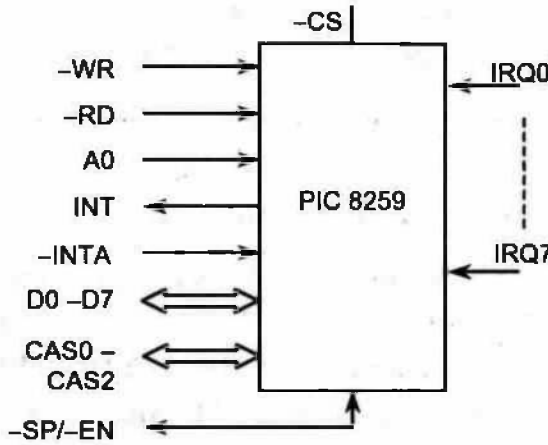
Chức năng cơ bản của PIC 8259 : PIC 8259 là một vi mạch điện tử khả trình được thiết kế để giúp CPU thực hiện quá trình ngắt cứng. PIC 8259 thực hiện các chức năng sau :

- + Ghi nhận được 8 yêu cầu ngắt IRQ_i , $i = 0, 1, \dots, 7$.
- + Cho phép chọn và phục vụ các yêu cầu ngắt theo mức ưu tiên.
- + Cung cấp cho CPU số ngắt tương ứng với yêu cầu ngắt IRQ_i . Số ngắt này đại diện cho địa chỉ của chương trình con phục vụ thiết bị yêu cầu ngắt IRQ_i .

+ Cho phép hoặc không cho phép các yêu cầu ngắt IRQ_i kích hoạt hệ thống ngắt.

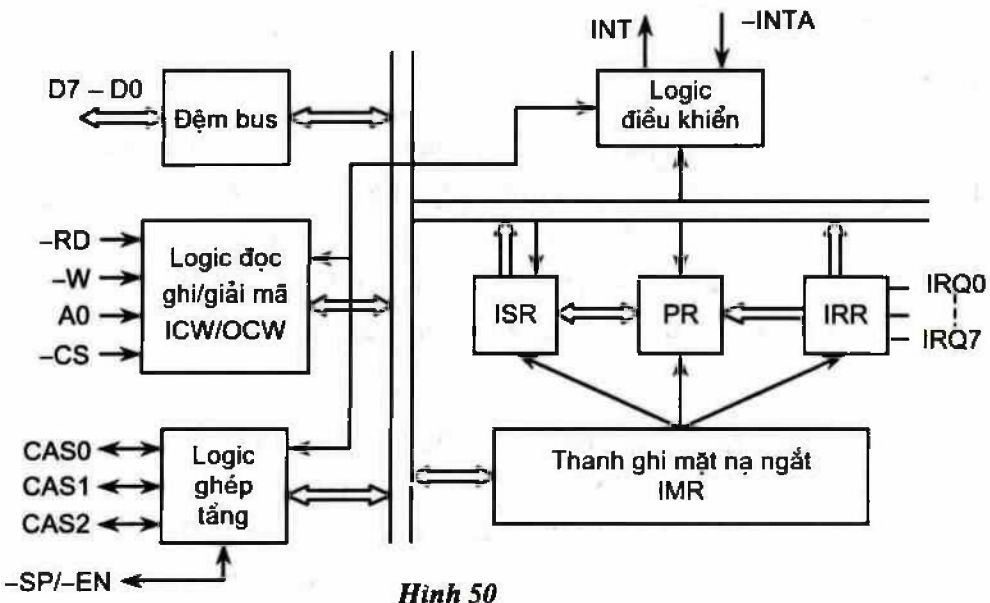
1.4 Thiết bị điều khiển ngắt PIC 8259 và cơ chế hoạt động của hệ thống ngắt cứng

Cấu trúc bên ngoài của PIC 8259 (h. 49) :



Hình 49

Cấu trúc bên trong của PIC 8259 (h.50) :



Hình 50

Các khối chức năng :

+ Thanh ghi yêu cầu ngắt IRR (Interrupt Request Register) : là thanh ghi 8 bit. IRR chứa (ghi nhận) tất cả các yêu cầu ngắt IRQ_i đòi phục vụ. Nếu tín hiệu $IRQ_i = "1"$ thì bit IRR_i tương ứng được đặt bằng "1".

+ Bộ giải quyết ưu tiên PR (Priority Resolver) : là thanh ghi 8 bit. PR xác định mức ưu tiên của các yêu cầu ngắt. Ngắt có ưu tiên cao nhất được chọn và đặt vào bit tương ứng trong ISR trong chu kỳ INTA.

+ Thanh ghi ngắt đang được phục vụ ISR (In Service Register) : là thanh ghi 8 bit. ISR ghi nhận các ngắt đang được phục vụ. Yêu cầu ngắt IRQ_i nào đang được phục vụ thì bit ISR_i tương ứng được đặt bằng "1".

+ Khối logic điều khiển : khối logic điều khiển đưa ra tín hiệu INT, được nối thẳng với chân INT của CPU. Khi INT có mức cao là đòi CPU phục vụ ngắt. Khối logic điều khiển nhận tín hiệu INTA từ CPU. Khi nhận được tín hiệu INTA, PIC 8259 sẽ cung cấp số ngắt ra BUS dữ liệu cho CPU.

+ Khối đệm Bus : là loại 8 bit, 2 hướng, 3 trạng thái. Các từ điều khiển ICW, OCW được đưa vào PIC 8259 qua khối này để xác lập chế độ hoạt động của 8259. Số ngắt và trạng thái hoạt động của PIC cũng được đưa ra BUS dữ liệu qua khối này.

+ Khối ghép tầng

PIC 8259 có cơ cấu cho phép nối ghép tầng các PIC 8259 với nhau và phối hợp hoạt động của các PIC này. Tầng thứ nhất có đầu ra INT nối trực tiếp với CPU, gọi là PIC 8259-chủ. Đầu vào IRQ_i của PIC chủ được nối với đầu ra INT của PIC 8259 thứ hai. PIC này được gọi là PIC 8259-thợ. Cơ chế ghép tầng cho phép xây dựng một hệ thống ngắt cứng quản lý được đến 64 yêu cầu ngắt IRQ.

+ Khối logic ghi/đọc và giải mã : thực hiện giải mã các từ điều khiển ICW (Initialization Command Word – Từ điều khiển khởi động) và OCW (Operation Command Word – Từ điều khiển hoạt động). Qua hai loại từ điều khiển này người sử dụng có thể lập trình xác lập chế độ hoạt động cho PIC.

+ Thanh ghi IMR : là thanh ghi 8 bit, cho phép đặt/xóa mặt nạ ngắt.

+ Bảng các tín hiệu CS, A0, RD, WR, và cách ghi/đọc PIC 8259 :

| CS | A0 | RD | WR | D4 | D3 | Hướng thông tin |
|----|----|----|----|----|----|-------------------------------|
| 0 | 0 | 0 | 1 | X | X | IRR, ISR => BUS |
| 0 | 1 | 0 | 1 | X | X | (IMR) = OCW1 => BUS |
| 0 | 0 | 1 | 0 | 0 | 0 | BUS => OCW2 |
| 0 | 0 | 1 | 0 | 0 | 1 | BUS => OCW3 |
| 0 | 0 | 1 | 0 | 1 | X | BUS => ICW1 |
| 0 | 1 | 1 | 0 | X | X | BUS => ICW2, ICW3, ICW4, OCW1 |

• Cơ chế hoạt động của hệ thống ngắt cứng :

Điều kiện ban đầu : PIC 8259 cần được lập trình khởi động qua các từ điều khiển ICW. Sau khi các từ điều khiển ICW được nạp thì PIC 8259 sẵn sàng hoạt động.

- Một hoặc nhiều thiết bị vào-ra có yêu cầu được phục vụ phát tín hiệu $IRQ_i = "1"$ (mức tích cực) cho PIC. PIC ghi nhận các yêu cầu ngắt IRQ_i này bằng cách đặt các bit IRR_i tương ứng lên "1".

- PIC 8259 chọn IRQ_i có mức ưu tiên cao nhất để phục vụ. PIC gửi tín hiệu INT cho CPU, đòi CPU phục vụ.

- CPU thực hiện các thao tác sau :

+ thực hiện nốt lệnh của quá trình hiện hành.

+ lưu địa chỉ trở về (nội dung của các thanh ghi CS, IP) và thanh ghi cờ FLAGS vào ngăn xếp.

+ gửi hai tín hiệu trả lời ngắt INTA cho PIC.

- Khi PIC 8259 nhận được tín hiệu INTA thứ 1 : bit ISR_i ứng với IRQ_i có mức ưu tiên cao nhất được thiết lập ($ISR_i = 1$) và bit IRR_i tương ứng bị xóa

($IRR_i = 0$). Trong chu kỳ INTA thứ nhất này PIC 8259 không gửi gì cho CPU qua BUS dữ liệu.

– Khi PIC 8259 nhận được tín hiệu INTA thứ 2 : PIC 8259 gửi số ngắt tương ứng với IRQ_i đang được phục vụ qua BUS dữ liệu cho CPU.

– CPU nhận số ngắt và trên cơ sở số ngắt này vào vị trí tương ứng trong Bảng véctơ ngắt để xác định địa chỉ của chương trình phục vụ ngắt. CPU nạp địa chỉ chương trình phục vụ ngắt vào các thanh ghi CS và IP và bắt đầu thực hiện chương trình phục vụ ngắt này.

– Khi thực hiện xong chương trình phục vụ ngắt (khi CPU thực hiện lệnh IRET) thì quá trình phục vụ ngắt của CPU cũng kết thúc. CPU khôi phục địa chỉ trở về vào các thanh ghi CS, IP, khôi phục nội dung thanh ghi FLAGS và tiếp tục thực hiện quá trình vừa bị ngắt.

Hệ thống ngắt cứng có thể kết thúc phục vụ ngắt hiện thời theo hai chế độ :

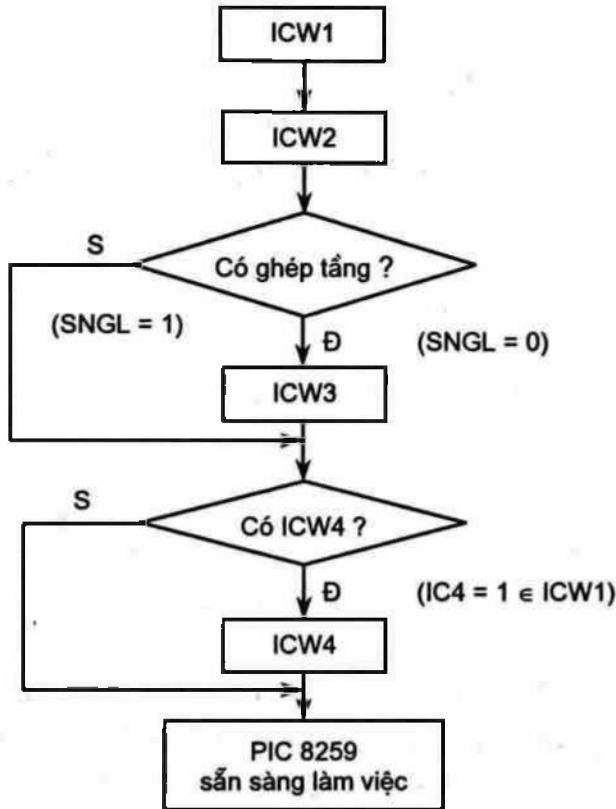
+ Kết thúc ngắt bình thường EOI (End Of Interrupt) : khi PIC được đặt chế độ kết thúc ngắt bình thường EOI thì CPU phải phát lệnh báo kết thúc ngắt EOI (qua từ điều khiển OCW2) cho PIC trước khi rời khỏi chương trình con phục vụ ngắt. Khi đó bit ISR_i của ngắt đang được phục vụ sẽ được đặt xuống 0.

+ Kết thúc ngắt tự động AEOI (Automatic EOI) : khi PIC được đặt chế độ kết thúc ngắt tự động AEOI thì tại chu kỳ INTA thứ 2 bit ISR_i của ngắt đang được phục vụ sẽ được đặt xuống 0.

Bằng các cách nói trên hệ thống ngắt cứng có thể tiếp tục phục vụ yêu cầu ngắt này ở những lần tiếp theo.

1.5 Lập trình khởi động PIC 8259 và các từ điều khiển khởi động ICW

Cần xác lập chế độ làm việc của PIC 8259 trước khi sử dụng. Quá trình này được gọi là lập trình khởi động thiết bị. Việc lập trình khởi động PIC 8259 được thực hiện qua các từ điều khiển ICW và theo lưu đồ sau :



a) ICW1

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|------|-----|------|-----|
| X | X | X | 1 | LTIM | ADI | SNGL | IC4 |

Các bit D5 – D7 không dùng dùng cho CPU x86.

+ IC4 (bit D0) : Cho biết có cần ICW4 ?

IC4 = 0 : không cần ICW4.

IC4 = 1 : có ICW4.

+ SNGL (bit D1) : cho biết hệ thống ngắt chỉ có một PIC hay có nhiều PIC ghép tầng.

SNGL = 0 có ghép tầng

SNGL = 1 chỉ có một PIC 8259

+ ADI (bit D2) : không dùng cho hệ CPU x86

+ LTIM : xác định dạng tín hiệu IRQ

LTIM = 1 IRQ phải là tín hiệu mức TTL

LTIM = 0 IRQ phải là tín hiệu dạng sườn xung.

+ D4 = 1

+ D5 = D6 = D7 = 0

b) ICW2 :

ICW2 định nghĩa số ngắt nền cho 7 số ngắt còn lại.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| T7 | T6 | T5 | T4 | T3 | x | x | x |

Các bit T7 – T3 là 5 bit cao của số ngắt, 3 bit còn lại liên quan đến các đầu vào IRQ_i

Năm bit cao T7 – T3 (do người sử dụng tùy chọn) cùng với 3 bit thấp nhất bằng 0 xác định số ngắt nền. Dựa trên số ngắt nền ứng với IRQ₀ này, PIC 8259 tự tạo ra các số ngắt tiếp theo tương ứng với các IRQ₁ đến IRQ₇.

Ví dụ : ở hệ thống ngắt cứng của máy vi tính PC, các số ngắt do PIC 8259 – chủ cung cấp như sau :

| | | | | | | | | |
|-------|---|---|---|---|---|---|---|---------------|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ứng với IRQ 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | ứng với IRQ 1 |
| | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | ứng với IRQ 7 |

c) ICW3 : liên quan đến ghép tầng.

Mạch phần cứng có chân SP/EN xác định chủ/thợ ở chế độ ghép tầng : nếu SP = 1 thì PIC là chủ, nếu SP = 0 thì PIC là thợ.

Có hai loại ICW3

– ICW3 cho PIC chủ : xác định đầu vào IRQ_i nhận tín hiệu INT từ PIC thợ thứ i.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |

Nếu $S_i = 1$ báo có PIC thợ nối vào chân IRQ_i của chủ

– ICW3 cho PIC thợ : xác định địa chỉ (chỉ thị nhận dạng) của PIC thợ.

| | | | | | | | |
|---|---|---|---|---|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | ID2 | ID1 | ID0 |
|---|---|---|---|---|-----|-----|-----|

Các bit ID2, ID1, ID0 xác định địa chỉ riêng của các PIC 8259–thợ. Khi nhận được tín hiệu INTA2, PIC 8259–thợ so sánh các tín hiệu CAS0 – CAS2 (phát ra từ PIC 8259–chủ) với ID2 – ID0, nếu chúng giống nhau thì PIC 8259–thợ gửi số ngắt lên BUS dữ liệu cho CPU, ngược lại thì không gửi.

d) ICW4 :

| | | | | | | | |
|----|----|----|------|------|-----|------|---------|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0 | 0 | 0 | SFNM | BUFF | M/S | AEOI | μP |

+ bit μP : báo cho PIC 8259 biết phải làm việc với họ vi xử lý nào.

$\mu P = 1$: làm việc với họ x86

$\mu P = 0$: làm việc với họ 8085

+bit AEOI : xác lập chế độ kết thúc ngắt.

AEOI = 0 : kết thúc bình thường EOI

AEOI = 1 : kết thúc tự động AEOI

+ bit BUFF : báo chế độ có bộ đệm BUS

BUFF = 1 : PIC làm việc ở chế độ đệm bus, lúc này tín hiệu SP/EN ở chế độ ra và việc định nghĩa chủ/thợ được xác định bằng bit M/S.

+ bit M/S : xác định chủ/thợ

M/S = 1 : PIC là chủ

M/S = 0 : PIC là thợ.

Nếu BUFF = 0 thì M/S không có ý nghĩa.

+ bit SFNM : bit này được đặt bằng 0 ngay khi khởi động hệ thống. Kiểu ưu tiên cố định là mặc định, trong đó IRQ_0 có mức ưu tiên cao nhất, IRQ_7 có mức ưu tiên thấp nhất. Có thể thay đổi kiểu ưu tiên bằng từ điều khiển OCW2. Trong kiểu ưu tiên cố định, khi SFNM = 0, khi bit $ISR_i = 1$ thì tất cả các IRQ_i có mức ưu tiên thấp hơn đều bị cấm. Chỉ có các IRQ_i có mức ưu tiên cao hơn được phép gây ngắt chương trình phục vụ ngắt hiện thời.

1.6 Các từ điều khiển hoạt động OCW

Các từ điều khiển OCW được dùng để xác lập các chế độ làm việc cụ thể trong quá trình hoạt động của PIC 8259. Có thể gửi các từ OCW này cho PIC8259 vào bất kỳ lúc nào sau khi khởi động hệ thống ngắt.

a) OCW1 : cho phép hoặc cấm nhận một yêu cầu ngắt IRQ_i nào đó bằng mặt nạ ngắt.

Với PIC chủ : địa chỉ thanh ghi chứa OCW1 là 21H

Với PIC thợ : địa chỉ thanh ghi chứa OCW1 là A1H

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|
| M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |

Mỗi bit M_i tương ứng với IRQ_i

Khi $M_i = 1$ mặt nạ ngắt được đặt, cấm PIC nhận IRQ_i (cấm IRQ_i gây ngắt)

Khi $M_i = 0$ mặt nạ ngắt được xoá, cho phép PIC nhận IRQ_i (cho phép IRQ_i gây ngắt)

Hệ điều hành đặt mặt nạ che chắn các IRQ mà hệ thống chưa dùng đến.

b) OCW2 : dùng để đổi kiểu ưu tiên và báo kết thúc ngắt EOI.

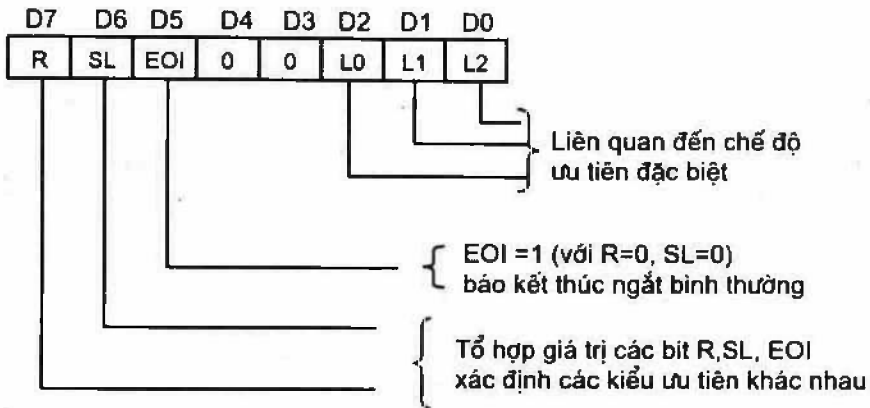
Địa chỉ thanh ghi chứa OCW2 : 20h (PIC chủ), A0h (PIC thợ)

PIC cho phép chọn một trong ba chế độ ưu tiên :

Ưu tiên cố định : IRQ_0 có mức ưu tiên cao nhất, IRQ_7 có mức ưu tiên thấp nhất. Trong chế độ này IRQ mức cao có quyền ngắt chương trình phục vụ ngắt có mức ưu tiên thấp hơn.

Ưu tiên quay vòng : IRQ_i nào vừa được phục vụ thì bit ISR_i sẽ bị xoá xuống 0 và tự động có mức ưu tiên thấp nhất. Điều này thực tế đã tạo ra các mức ưu tiên bằng nhau.

Ưu tiên đặc biệt : người lập trình có thể thay đổi mức ưu tiên bằng chương trình. Nếu các bit trong OCW2 $R = 1, SL = 1$ thì các bit $L2-L0$ sẽ đặt IRQ_n xuống mức thấp nhất và IRQ_{n+1} lên mức cao nhất.



c) OCW3 : cho phép đặt/đọc ISR và các thanh ghi khác của PIC 8259.

| | | | | | | | |
|---|------|-----|---|---|---|----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | ESMM | SMM | 0 | 1 | P | RR | RIS |

- ESMM = 1 và SMM cho phép đặt/xoá chế độ mặt nạ đặc biệt. Chế độ mặt nạ đặc biệt này chỉ cấm một IRQ và cho phép tất cả các IRQ còn lại được yêu cầu ngắt.

- $D4 = 0, D3 = 1$

- bit P : cho phép PIC 8259 làm việc với CPU ở chế độ hỏi đáp, không cần qua các tín hiệu INT, INTA. Nếu $P = 1$ thì PIC coi tín hiệu điều khiển đọc RD như là tín hiệu INTA.

- các bit RR và RIS :

RR = 1 & RIS = 0 : báo sẽ đọc IRR ở lệnh đọc tiếp sau

RR = 1 & RIS = 1 : báo sẽ đọc ISR ở lệnh đọc tiếp sau.

1.7 Phân bố chức năng các yêu cầu ngắt và số ngắt trong máy PC

PIC 8259–chủ :

PIC 8259–chủ chiếm hai địa chỉ cổng : 20h, 21h

| IRQ _i | Số ngắt | Thiết bị yêu cầu ngắt |
|------------------|---------|--|
| IRQ ₀ | 08h | Bộ tạo xung nhịp đồng hồ hệ thống |
| IRQ ₁ | 09h | Thiết bị giao diện bàn phím |
| IRQ ₂ | 0Ah | PIC 8259–thợ |
| IRQ ₃ | 0Bh | Thiết bị giao diện vào/ra nối tiếp 2 (COM 2) |
| IRQ ₄ | 0Ch | Thiết bị giao diện vào/ra nối tiếp 1 (COM 1) |
| IRQ ₅ | 0Dh | Dự phòng |
| IRQ ₆ | 0Eh | Thiết bị giao diện ổ đĩa mềm FDC |
| IRQ ₇ | 0Fh | Thiết bị giao diện vào/ra song song (LPT1) |

PIC 8259–thợ :

PIC 8259–thợ chiếm hai địa chỉ cổng : A0h, A1h

| Dãy IRQ | Số ngắt | Thiết bị yêu cầu ngắt |
|-------------------|---------|-------------------------------------|
| IRQ ₈ | 70h | Đồng hồ thời gian thực |
| IRQ ₉ | 71h | Dự phòng |
| IRQ ₁₀ | 72h | Card âm thanh |
| IRQ ₁₁ | 73h | Thiết bị giao diện vào/ra USB |
| IRQ ₁₂ | 74h | Thiết bị giao diện chuột PS/2 |
| IRQ ₁₃ | 75h | Bộ đồng xử lý x87 |
| IRQ ₁₄ | 76h | Bộ điều khiển bus IDE 1(primary) |
| IRQ ₁₅ | 77h | Bộ điều khiển bus IDE 2 (secondary) |

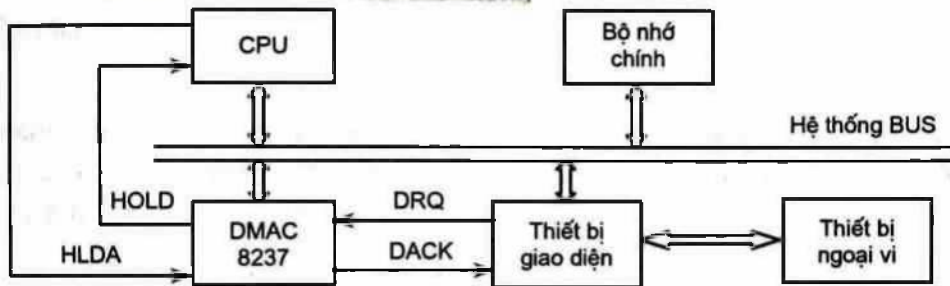
2. TRUY NHẬP TRỰC TIẾP BỘ NHỚ VÀ BỘ ĐIỀU KHIỂN DMAC 8237

2.1 Khái niệm cơ bản

Quá trình chuyển dữ liệu trực tiếp giữa thiết bị ngoại vi và bộ nhớ chính không qua CPU được gọi là quá trình DMA (Direct Memory Access).

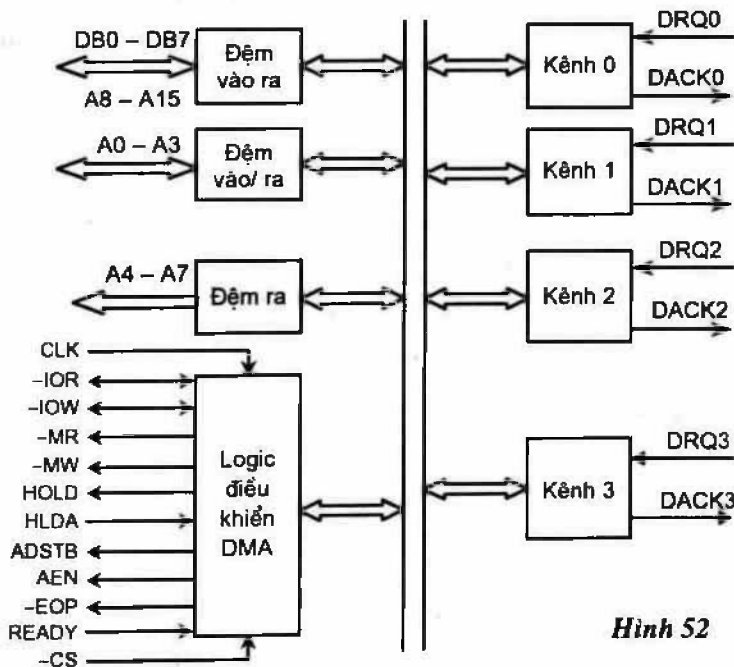
Trong quá trình DMA, việc chuyển dữ liệu không được điều khiển bằng chương trình mà được điều khiển bằng một thiết bị đặc biệt gọi là bộ điều khiển DMAC (DMA Controller).

Cấu trúc hệ thống vào-ra dữ liệu kiểu DMA như sau (h.51) :



Hình 51

2.2 Sơ đồ khối mạch DMAC 8237 (h.52)



Hình 52

– DRQ0–DRQ3 (tín hiệu vào) : tín hiệu yêu cầu DMA. Bốn dây DRQ nối với 4 thiết bị cần phục vụ kiểu DMA. Khi thiết bị ngoại vi đưa tín hiệu DRQ lên mức tích cực thì quá trình DMA bắt đầu.

DRQ0 có mức ưu tiên cao nhất, DRQ3 có mức ưu tiên thấp nhất.

– DACK0–DACK3 (tín hiệu ra) : tín hiệu trả lời DMA. DACK được nối vào từng thiết bị ngoài để báo cho thiết bị ngoài biết là DMAC đang phục vụ nó.

– HOLD (tín hiệu ra) : dùng để báo cho CPU là DMAC đòi quyền điều khiển BUS.

– HLDA(tín hiệu vào) : tín hiệu HLDA được gửi từ CPU báo cho DMAC biết là quyền điều khiển bus thuộc về DMAC.

– A0–A3(tín hiệu vào/ra) : 4 dây địa chỉ thấp. Trong chu kỳ nghỉ được dùng để xác định địa chỉ các thanh ghi trong DMAC khi lập trình hoạt động cho DMAC. Trong chu kỳ tích cực được dùng để cung cấp 4 bit địa chỉ A0–A3 cho BUS địa chỉ.

– A4– A7(tín hiệu ra) : cung cấp 4 bit địa chỉ cao cho bus địa chỉ trong quá trình DMA.

– DB0 – DB7(tín hiệu vào/ra) : là BUS 8 bit đa năng.

Trong chu kỳ nghỉ : là BUS 8 bit dữ liệu vào truyền dữ liệu xác lập chế độ làm việc cho DMAC và cũng là BUS 8 bit dữ liệu ra trong quá trình CPU đọc trạng thái DMAC.

Trong chu kỳ tích cực (chu kỳ DMA) : truyền hai loại thông tin trong 2 thời khoảng :

+ Thời khoảng đầu : với sự có mặt của tín hiệu ADSTB thì DB0–DB7 là BUS 8 bit địa chỉ cao nhất (A15–A8), 8 bit địa chỉ này cần được chốt lại để kết hợp với 8 bit địa chỉ thấp (A7–A0) thành địa chỉ 16 bit.

+ Thời khoảng sau : là BUS 8 bit dữ liệu, dành cho truyền dữ liệu trong chế độ DMA.

– MR (tín hiệu ra) : là tín hiệu điều khiển đọc bộ nhớ trong chế độ DMA.

– MW(tín hiệu ra) : là tín hiệu điều khiển ghi bộ nhớ trong chế độ DMA.

– ADSTB (tín hiệu ra) : dùng để điều khiển việc chốt (ghi lại) 8 bit địa chỉ cao (A8 – A15).

– AEN : (tín hiệu ra) cho phép/cấm thiết bị chốt địa chỉ hoạt động. AEN cũng được dùng để cấm các thiết bị điều khiển BUS khác trong chế độ DMA.

– EOP (tín hiệu vào/ra) : khi là tín hiệu vào, tín hiệu EOP buộc DMAC kết thúc quá trình DMA. Là tín hiệu ra trong chế độ hoạt động, tín hiệu EOP do DMAC phát ra báo đã truyền đủ số byte xác định bởi TC.

– IOW (tín hiệu vào/ra) : trong chế độ nghỉ là tín hiệu vào, tín hiệu này do CPU cung cấp để điều khiển ghi dữ liệu vào DMAC. Trong chu kỳ tích cực là tín hiệu ra, DMAC tạo tín hiệu điều khiển ghi thiết bị ngoại vi.

2.3 Các chế độ hoạt động của DMAC 8237

DMAC 8237 hoạt động trong 2 chu kỳ chính là : chu kỳ nghỉ và chu kỳ hoạt động.

+ Chu kỳ nghỉ : khi không có kênh nào yêu cầu thì DMAC đi vào chu kỳ nghỉ. Trong chu kỳ này DMAC “quan sát” các dây DRQ ở từng nhịp đồng hồ, đồng thời làm việc với CPU. Trong chu kỳ nghỉ của DMAC thì CPU có thể xác lập, thay đổi hoặc kiểm tra nội dung các thanh ghi chế độ làm việc của DMAC. Khi có tín hiệu DRQ tích cực, DMAC gửi ra tín hiệu HOLD và đi vào chu kỳ hoạt động.

+ Chu kỳ hoạt động : Trong chu kỳ hoạt động DMAC 8237 có thể hoạt động theo 1 trong 4 kiểu, tùy theo lập trình :

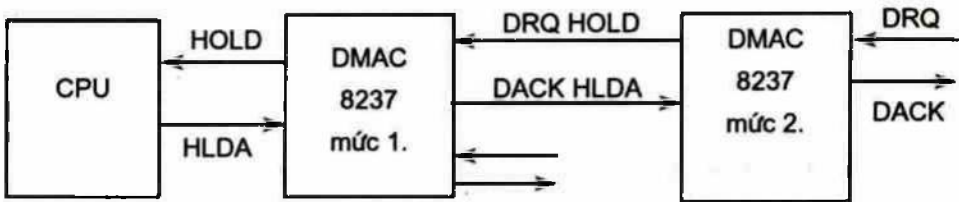
– Kiểu truyền đơn lẻ (single transfer mode) : trong kiểu truyền này, DMAC được lập trình để thực hiện chỉ một lần truyền. Thanh đếm từ TC tự động giảm, địa chỉ tăng hoặc giảm cho đến khi TC = 0. Trong kiểu truyền này DRQ phải được giữ ở mức tích cực chừng nào DACK tích cực còn được ghi nhận. Nếu DRQ còn giữ ở mức tích cực quá một lần truyền thì HOLD bị đặt xuống 0 và DMAC trả quyền điều khiển bus cho CPU.

– Kiểu truyền khối (block transfer mode) : trong kiểu truyền khối DMAC bắt đầu thực hiện việc truyền cả khối dữ liệu khi DRQ tích cực và tiếp tục truyền cho đến khi thanh đếm TC đếm xuống 0 hoặc cho đến khi nhận được tín hiệu EOP từ bên ngoài. Tín hiệu DRQ chỉ cần được giữ tích cực cho đến khi DACK trở thành tích cực.

– Kiểu truyền theo yêu cầu (demand transfer mode) : trong kiểu truyền này, dữ liệu được truyền cho đến khi DRQ không tích cực hoặc TC = 0 hoặc

nhận được tín hiệu EOP. Việc truyền dữ liệu trực tiếp cho đến khi thiết bị bên ngoài chuyển được hết dữ liệu của nó. Nếu thiết bị chuẩn bị dữ liệu chậm hơn, nó đặt DRQ thành không tích cực. Khi thiết bị chuẩn bị xong thì nó đặt DRQ lên thành tích cực và quá trình DMA lại tiếp tục.

- Kiểu ghép tầng(Cascade mode)(h.53).



Hình 53

Trong kiểu ghép tầng : dây HOLD của DMAC tầng 2 nối với dây DRQ của DMAC tầng 1. Dây DACK của DMAC tầng 1 nối với các dây HLDA của DMAC tầng 2.

Yêu cầu DRQ của DMAC tầng 2 được truyền qua mạch phân định ưu tiên của DMAC tầng 1. Trong trường hợp kênh có nối tầng, DMAC tầng 1 chỉ thực hiện chức năng phân định ưu tiên và không đưa ra các tín hiệu điều khiển BUS của riêng nó.

2.4 Các thanh ghi của DMAC

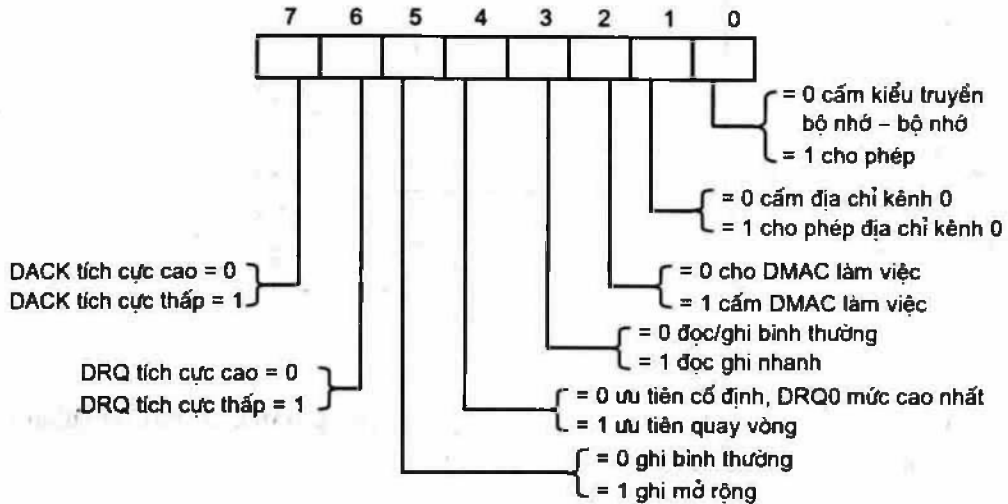
Mỗi kênh DMA có 4 thanh ghi chức năng :

- Thanh ghi địa chỉ nền (16 bit) : lưu giữ địa chỉ nền (địa chỉ đầu) của vùng nhớ cần truy cập.
- Thanh ghi đếm nền (16 bit) : lưu giữ con số xác định lượng từ dữ liệu gốc cần truyền.
- Thanh ghi địa chỉ hiện thời (16 bit) : Thanh ghi này chứa giá trị địa chỉ của ô nhớ đang được truy nhập trong quá trình DMA. Địa chỉ tự động tăng hoặc giảm sau một lần truyền và giá trị tức thời này lập tức được nạp vào đây. Thanh ghi này được ghi/đọc bởi CPU.
- Thanh ghi đếm từ hiện thời (16 bit) : thanh ghi này giữ số lần truyền được thực hiện. Số đếm này giảm sau mỗi lần truyền. Khi giá trị của thanh ghi đếm từ hiện thời giảm xuống 0 thì tín hiệu EOP được tạo ra. Thanh ghi này được ghi/đọc bởi CPU.

Các thanh ghi điều khiển và trạng thái :

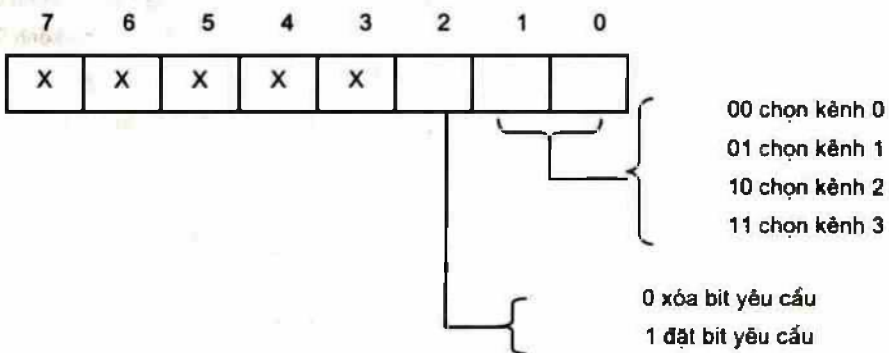
+ Thanh ghi lệnh (Command Register)

Thanh ghi lệnh (8 bit) này được nạp để xác lập chế độ làm việc cho DMAC : cho phép hoạt động, kiểu ưu tiên, kiểu tốc độ truy nhập, dạng tín hiệu DRQ và DACK.

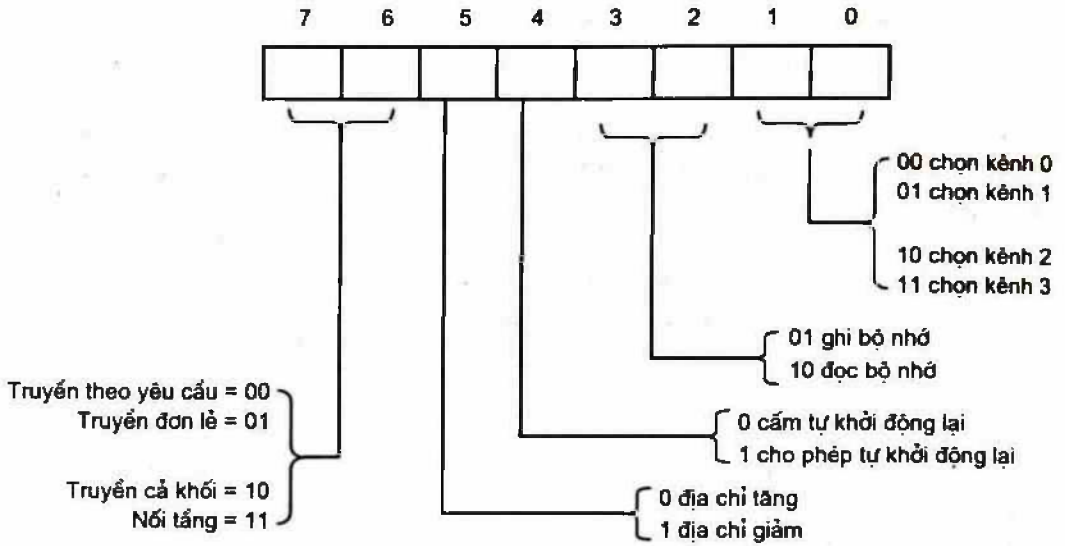


+ Thanh ghi yêu cầu (request register)

Thanh ghi yêu cầu (8 bit) : cho phép DMAC có thể trả lời DACK cho từng thiết bị yêu cầu và đặt/xóa yêu cầu DRQ.

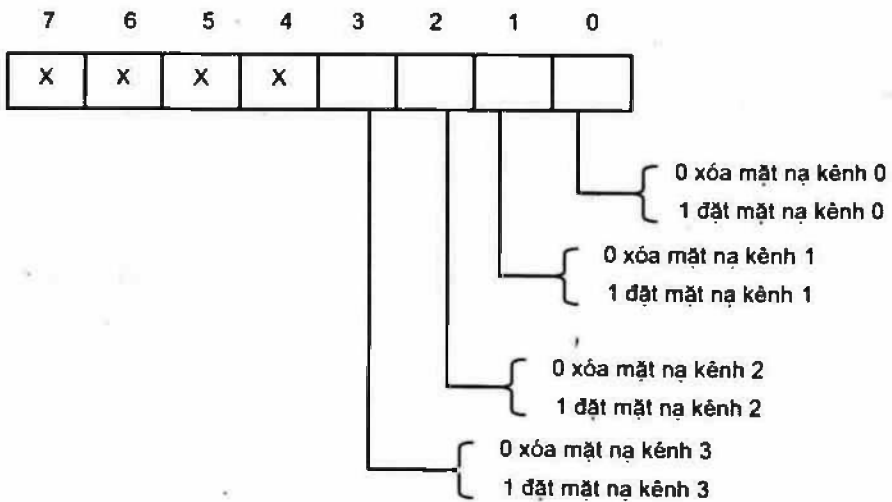


+ Thanh ghi kiểu làm việc (8 bit) : xác lập chế độ và kiểu làm việc cho từng kênh DMA, cho phép chọn : kênh, kiểu truyền, thực hiện DMA để ghi hay đọc, kiểu tăng/giảm địa chỉ , cấm/cho phép tự khởi đầu lại.



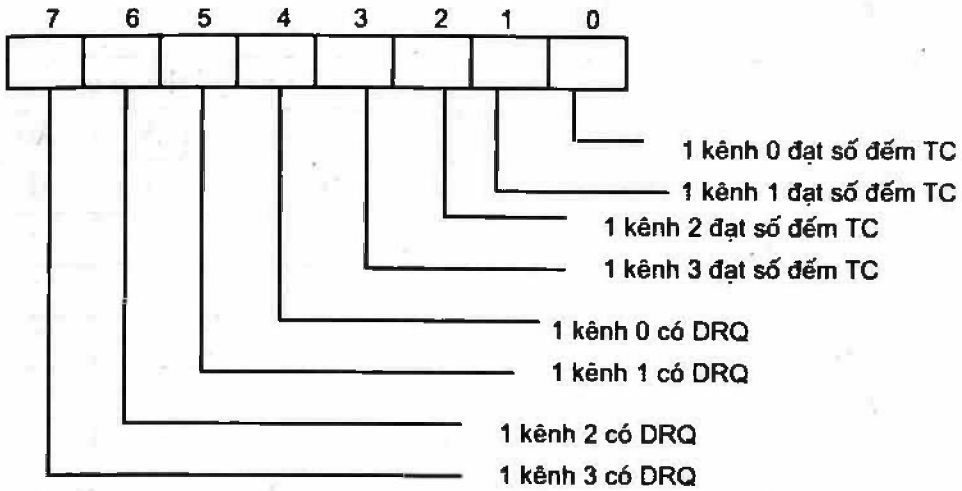
+ Thanh ghi mật nạ

Thanh ghi mật nạ (8 bit) : thanh ghi này cho phép đặt/xóa mật nạ (nhận/ không nhận DRQ) đến mức từng kênh.



+ Thanh ghi trạng thái :

Thanh ghi trạng thái(8 bit) : cho biết trạng thái DRQ và TC của từng kênh :



2.5 Lập trình chế độ làm việc DMAC 8237

Địa chỉ nền của DMAC 8237 #1 : 000h

Địa chỉ nền của DMAC 8237 #2 : 0C0h

+ Lập trình các thanh ghi điều khiển

| Thanh ghi | Thao tác | Địa chỉ thanh ghi | | | |
|-----------------|----------|-------------------|----|----|----|
| | | A3 | A2 | A1 | A0 |
| Lệnh | Ghi | 1 | 0 | 0 | 0 |
| Kiểu làm việc | Ghi | 1 | 0 | 1 | 1 |
| Trạng thái | Đọc | 1 | 0 | 0 | 0 |
| Đặt /xoá mặt nạ | Ghi | 1 | 0 | 1 | 0 |
| Yêu cầu | Ghi | 1 | 0 | 0 | 1 |

+ Lập trình các thanh ghi địa chỉ và đếm

| Kênh | Thanh ghi | Thao tác | Địa chỉ thanh ghi | | | |
|------|-------------------------|----------|-------------------|----|----|----|
| | | | A3 | A2 | A1 | A0 |
| 0 | Byte thấp địa chỉ nền | Ghi | 0 | 0 | 0 | 0 |
| | Byte cao địa chỉ nền | Ghi | 0 | 0 | 0 | 0 |
| | Byte thấp thanh đếm nền | Ghi | 0 | 0 | 0 | 1 |
| | Byte cao thanh đếm nền | Ghi | 0 | 0 | 0 | 1 |
| 1 | Byte thấp địa chỉ nền | Ghi | 0 | 0 | 1 | 0 |
| | Byte cao địa chỉ nền | Ghi | 0 | 0 | 1 | 0 |
| | Byte thấp thanh đếm nền | Ghi | 0 | 0 | 1 | 1 |
| | Byte cao thanh đếm nền | Ghi | 0 | 0 | 1 | 1 |
| 2 | Byte thấp địa chỉ nền | Ghi | 0 | 1 | 0 | 0 |
| | Byte cao địa chỉ nền | Ghi | 0 | 1 | 0 | 0 |
| | Byte thấp thanh đếm nền | Ghi | 0 | 1 | 0 | 1 |
| | Byte cao thanh đếm nền | Ghi | 0 | 1 | 0 | 1 |
| 3 | Byte thấp địa chỉ nền | Ghi | 0 | 1 | 1 | 0 |
| | Byte cao địa chỉ nền | Ghi | 0 | 1 | 1 | 0 |
| | Byte thấp thanh đếm nền | Ghi | 0 | 1 | 1 | 1 |
| | Byte cao thanh đếm nền | Ghi | 0 | 1 | 1 | 1 |

Trình tự lập trình khởi động DMAC

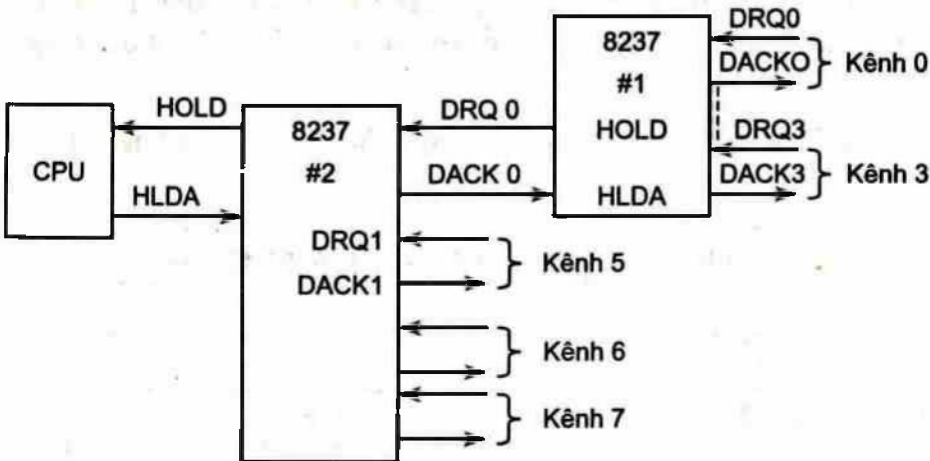
- + Nạp byte thấp cho thanh ghi địa chỉ nền
- + Nạp byte cao cho thanh ghi địa chỉ nền.
- + Nạp byte thấp cho thanh ghi đếm.
- + Nạp byte cao cho thanh ghi đếm
- + Nạp thanh ghi lệnh, thanh ghi yêu cầu và thanh ghi kiểu làm việc.
- + Nạp thanh ghi địa chỉ trang
- + Xóa mặt nạ kênh cần dùng.
- + Cho phép kênh hoạt động

2.6 Quá trình DMA

DMAC được xác lập chế độ làm việc : các thanh ghi lệnh, thanh ghi kiểu làm việc, thanh ghi địa chỉ nền và thanh đếm kích thước khối dữ liệu được nạp, cho phép kênh làm việc.

- Thiết bị vào-ra phát tín hiệu DRQ cho DMAC.
- DMAC phát tín hiệu HOLD = 1 cho CPU, đòi CPU đi vào chế độ DMA.
- CPU thực hiện nốt chu kỳ máy.
- CPU phát tín hiệu HLDA cho DMAC và tách ra khỏi hệ thống BUS. Quyền điều khiển BUS thuộc về DMAC.
- DMAC làm chủ các BUS địa chỉ, BUS số liệu và BUS điều khiển. DMAC tạo ra tín hiệu DACK, phát ra địa chỉ bộ nhớ (16 bit), tạo tín hiệu điều khiển ghi/ đọc thiết bị vào/ra, tín hiệu điều khiển ghi/đọc bộ nhớ và một byte dữ liệu được truyền trực tiếp giữa bộ nhớ và thiết bị vào-ra. Địa chỉ ô nhớ tiếp theo được phát ra và quá trình trên lặp lại cho đến khi thực hiện xong quá trình DMA theo kiểu đã được lập trình (TC = 0).
- Khi quá trình DMA kết thúc DMAC phát tín hiệu HOLD = 0 cho CPU và trả quyền điều khiển hệ thống BUS cho CPU. Quá trình DMA cũng có thể được kết thúc từ bên ngoài bằng tín hiệu EOP.
- CPU nắm lại quyền điều khiển hệ thống BUS và tiếp tục làm việc bình thường.

2.7 Mạch DMA trong máy vi tính PC/AT (h.54)



Hình 54

Có 8 kênh DMA.

- Địa chỉ cổng DMAC 8237 # 1 : 000h–01Fh
Địa chỉ nền của các thanh ghi DMAC 8237 # 1 là 000h
DMAC #1 có 4 kênh DMA
 - Kênh 0 : DRQ0, làm tươi bộ nhớ
 - Kênh 1 : DRQ1, tùy chọn
 - Kênh 2 : DRQ2, dành cho đĩa mềm
 - Kênh 3 : DRQ3, tùy chọn
- Địa chỉ cổng DMAC 8237 # 2 : 0C0h – 0DFh
Địa chỉ nền của các thanh ghi DMAC 8237 #2 là : 0C0h
DMAC #2 có 4 kênh DMA
 - Kênh 0 : DRQ4, nối tăng với DMAC 8237 #1
 - Kênh 1 : DRQ5, tùy chọn
 - Kênh 2 : DRQ6, tùy chọn
 - Kênh 3 : DRQ7, tùy chọn

2.8 Thanh ghi trang và địa chỉ trang

DMAC chỉ tạo được địa chỉ 16 bit nên chỉ cho phép truy nhập được vùng nhớ 64 Kbyte. Để thực hiện quá trình DMA với loại bộ nhớ có bus địa chỉ 24 bit ở máy PC người ta tổ chức quản lý bộ nhớ theo trang.

Người ta thiết kế các thanh ghi trang chứa 8 bit cao của địa chỉ 24 bit, kết hợp với một mạch điện tử (74LS612) để tạo các địa chỉ vật lý 24 bit trong quá trình DMA.

Hệ thống DMA trong máy vi tính PC có 4 thanh ghi trang ứng với 4 kênh DMAC.

| Thanh ghi trang | Địa chỉ cổng thanh ghi trang |
|-----------------|------------------------------|
| Kênh 0 | 87h |
| Kênh 1 | 83h |
| Kênh 2 | 81h |
| Kênh 3 | 82h |

Các thanh ghi trang chứa địa chỉ đầu của các vùng nhớ, mỗi một vùng nhớ được dùng làm vùng nhớ đệm phục vụ quá trình DMA :

| | Thanh ghi trang (chứa 8 bit cao của địa chỉ 24 bit) | Thanh ghi địa chỉ nền trong DMAC 8237 |
|----------------|--|--|
| Địa chỉ bộ nhớ | A23.....A16 | A15.....A0 |

3. VÀO-RA NỐI TIẾP VÀ THIẾT BỊ GIAO DIỆN VÀO-RA NỐI TIẾP UART 8250/16450

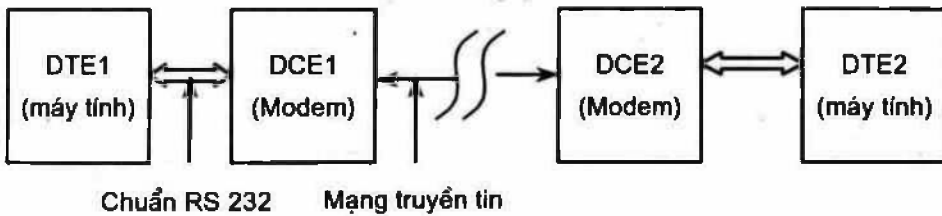
Vào/ra nối tiếp là phương pháp kết nối và truyền dữ liệu giữa máy tính và thiết bị ngoại vi trong đó mã chứa thông tin được truyền theo từng bit, bit nọ sau bit kia (truyền tin nối tiếp).

3.1 Chuẩn truyền tin RS-232

Chuẩn truyền tin RS-232 quy định về phương pháp kết nối và giao diện giữa DTE và DCE (h.55).

DTE (Data Terminal Equipment) thiết bị đầu cuối dữ liệu : là thiết bị phát hoặc nhận dữ liệu.

DCE (Data Circuit-terminating Equipment) : thiết bị truyền dữ liệu.



Hình 55

Chuẩn RS232 quy định về :

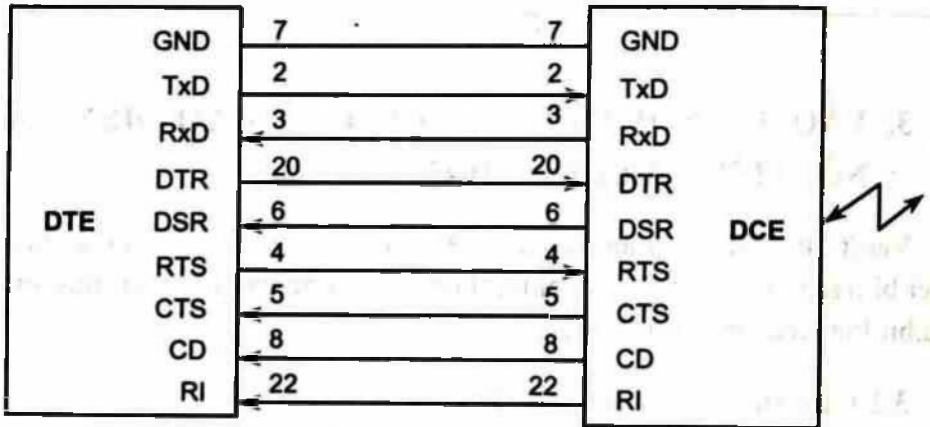
- Kết nối vật lý : loại đầu nối, số lượng, vị trí và chức năng truyền thông tin của mỗi chân trong đầu nối.

- Mức điện áp tín hiệu.

- Tốc độ truyền.

+ Kết nối vật lý : sử dụng 2 loại đầu nối 9 chân hoặc 25 chân : DB9 và DB25.

- Kết nối DTE & DCE qua đầu nối DB25 (h.56) :

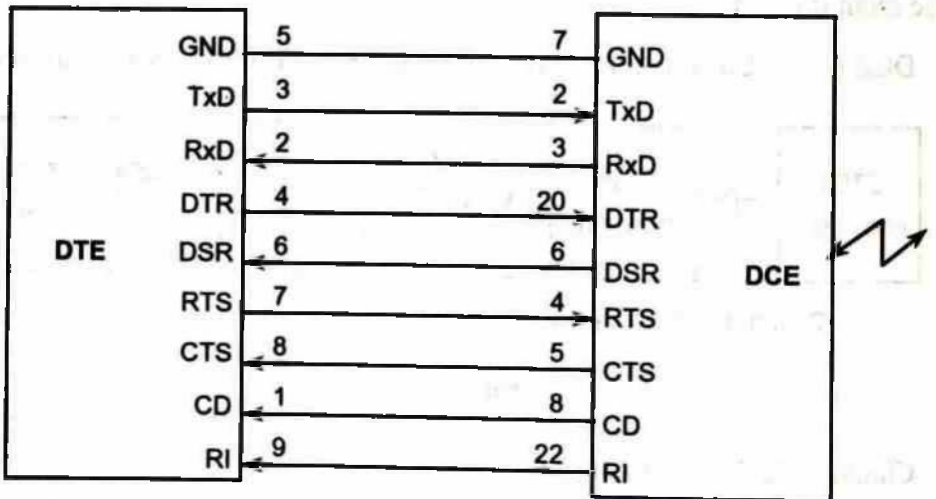


(Cổng COM của máy vi tính)

Hình 56

(cổng MODEM)

Kết nối DTE & DCE qua đầu nối DB9 (h.57)



(Cổng COM của máy vi tính)

Hình 57

(cổng MODEM)

Ý nghĩa của các tín hiệu , nhìn từ phía DTE

- Các tín hiệu mang dữ liệu
 - TxD : tín hiệu dữ liệu phát
 - RxD : tín hiệu dữ liệu nhận
- Các tín hiệu *bắt tay* : (khi các tín hiệu có mức tích cực)

Từ DTE (từ cổng COM máy vi tính)

- DTR : DTE báo sẵn sàng làm việc
- RTS : DTE báo có dữ liệu muốn gửi

Đến DTE (vào máy tính)

- DSR : DCE báo sẵn sàng làm việc
- CTS : DCE sẵn sàng nhận dữ liệu để gửi đi
- CD : DCE báo phát hiện được sóng mang của phía bên kia.
- RI : DCE báo nhận được tín hiệu rung chuông.

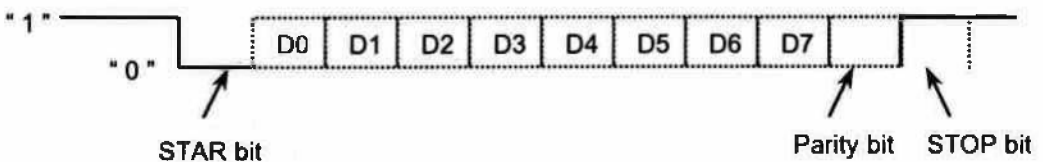
- + Mức tín hiệu: +/- 25V
- “0” : +3V đến +25V
- “1” : -3V đến -25V

+ Kích thước cáp nối không quá 20 m.

+ Tốc độ truyền : hai bên phát và nhận phải có tốc độ bằng nhau : từ 50 bit/s đến 20000bit/s.

Phương pháp truyền

DTE truyền dữ liệu không đồng bộ về pha nhưng đồng bộ trên từng byte dữ liệu bằng bit “ START”. Dữ liệu khi được truyền có khuôn dạng sau :



Khuôn dạng dữ liệu giữa bên phát và bên nhận phải được chọn giống nhau.

3.2 Modem (DCE)

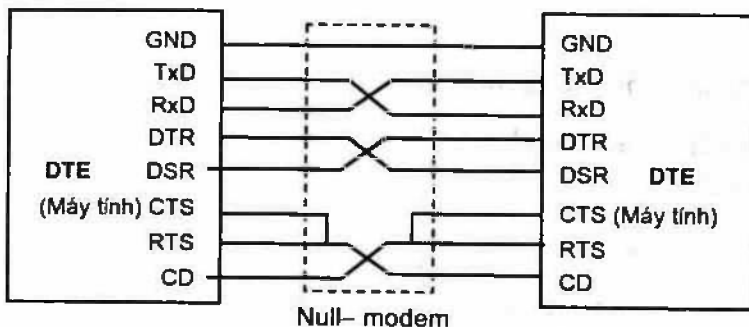
Khi cần thực hiện truyền dữ liệu giữa hai máy tính khoảng cách lớn hơn 20 mét, phương pháp có hiệu quả nhất là truyền qua hệ thống điện thoại công cộng. Thiết bị kết nối máy tính số với hệ thống điện thoại là Modem. Modem thực hiện chuyển tín hiệu số nhị phân thành tín hiệu tương tự để hệ thống điện thoại có thể truyền đi được. Modem cũng thực hiện chuyển tín hiệu tương tự nhận được từ hệ thống điện thoại thành tín hiệu số nhị phân cho máy vi tính. Ngoài chức năng điều chế (MODulate) và giải điều chế (DEMODulate), modem còn thực hiện các chức năng giao diện với hệ thống điện thoại và với máy vi tính.

3.3 Kết nối kiểu Null-modem

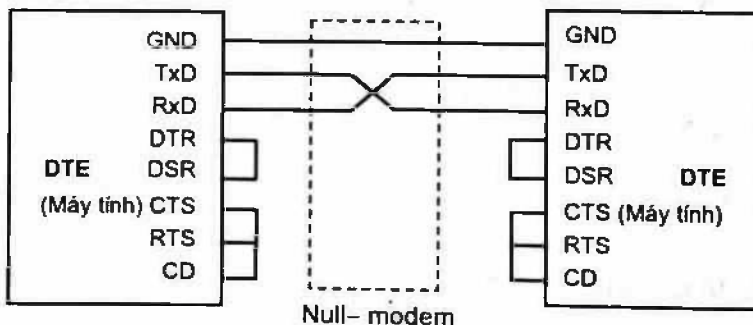
Có thể thực hiện kết nối và truyền tin trực tiếp giữa máy tính với máy tính hoặc giữa máy tính với thiết bị ngoại vi khác qua cổng nối tiếp, không qua modem. Phương pháp kết nối trực tiếp DTE-DTE (máy tính với máy tính) không qua Modem được gọi là kết nối kiểu Null-Modem. Khi thực hiện kết nối và truyền tin nối tiếp không qua Modem vẫn cần tuân theo chuẩn RS 232.

Có hai kiểu kết nối Null-modem : kiểu 7 dây và kiểu 3 dây.

Kết nối Null-modem kiểu 7 dây (h.58) và Null-modem kiểu 3 dây (h.59) :

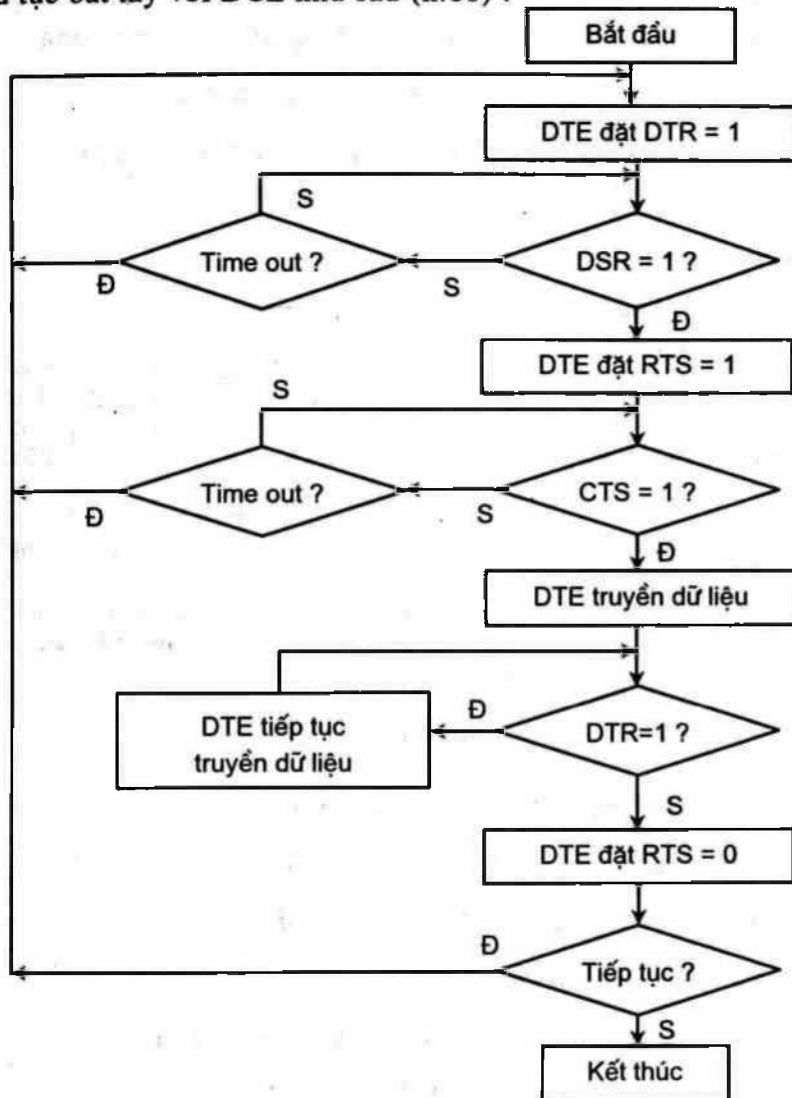


Hình 58



Hình 59

Thủ tục bắt tay với DCE như sau (h.60) :



Hình 60

3.4 Thiết bị giao diện vào/ra nối tiếp UART 8250/16450

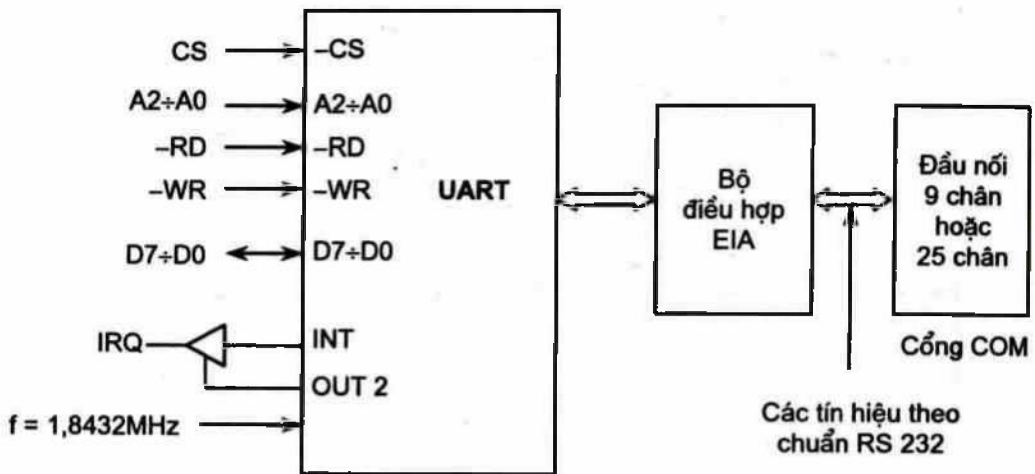
Thiết bị giao diện vào/ra nối tiếp UART 8250/16450 thực hiện hai chức năng chính :

+ Chuyển 1 byte dữ liệu (nhận từ CPU) dạng 8 bits song song thành dạng nối tiếp , tạo khung dữ liệu dạng nối tiếp và phát đi tuần tự từng bit cho đến

hết byte dữ liệu. Nhận 1 khung dữ liệu dạng nối tiếp, loại bỏ các bit tạo khung (bit START, PARITY, STOP), chuyển thành dạng dữ liệu song song.

+ Tạo và nhận các tín hiệu bắt tay theo chuẩn RS 232.

Sơ đồ khối cổng vào–ra nối tiếp trong máy vi tính PC (h.61) :



Hình 61

Địa chỉ nền của 2 thiết bị giao diện vào–ra nối tiếp :

Thiết bị UART #1 (COM1) : 3F8h

Thiết bị UART #2 (COM2) : 2F8h

a) Thanh ghi dữ liệu phát THR :

THR là nơi chứa dữ liệu (hoặc ký tự) cần phát đi.

Địa chỉ : Địa chỉ nền +0, ghi, DLAB = 0



Bit D0 là bit thấp nhất, được phát đầu tiên.

b) Thanh ghi dữ liệu nhận RBR :

RBR là nơi chứa dữ liệu (hoặc ký tự) nhận được

Địa chỉ : Địa chỉ nền +0, đọc, DLAB = 0



DLAB là bit D7 của thanh ghi LCR.

c) Thanh ghi điều khiển đường truyền LCR

Thanh ghi LCR xác định khuôn dạng dữ liệu phát/nhận và cho phép truy nhập vào các thanh ghi THR, RBR, IER hoặc BRG.

Địa chỉ : Địa chỉ nền +3, ghi



D1 và D0 : xác định kích thức dữ liệu truyền.

| D1 | D0 | |
|----|----|-------|
| 0 | 0 | 5 bit |
| 0 | 1 | 6 bit |
| 1 | 0 | 7 bit |
| 1 | 1 | 8 bit |

D2- xác định số lượng bit stop được tạo và kiểm tra

D2 = 0 1 Bit Stop

D2 = 1 2 Bit Stop

D3- cho phép tạo hoặc kiểm tra parity

D3 = 0 không cho phép

D3 = 1 cho phép

D4- chọn kiểu parity

D4 = 0 Số lượng lẻ bit "1" được báo hoặc kiểm tra

D4 = 1 Số lượng chẵn bit "1" được báo hoặc kiểm tra

D5- chọn mức tích cực của bit parity

Nếu D5 = 1 và D4 = 1 thì mức tích cực của bit parity là 0 (Parity chẵn)

Nếu D5 = 1 và D4 = 0 thì mức tích cực của bit parity là 1 (Parity lẻ)

D6- đặt điều khiển nghỉ (BREAK)

Khi D6 = 1 thì SOUT = 0. Chú ý D6 = 0 ! (thường đặt)

D7- DLAB bit : cho phép truy nhập các thanh ghi THR,RBR, IER hoặc BRG

D7 = 1 cho phép truy nhập cặp thanh ghi tốc độ truyền BRG

D7 = 0 cho phép truy nhập các thanh ghi THR, RBR và IER

d) Thanh ghi xác lập tốc độ truyền 16 bit BRG

BRG gồm 2 thanh ghi byte cao và byte thấp, xác định hệ số chia của tốc độ truyền.

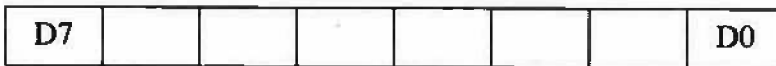
Byte thấp :

Địa chỉ : Địa chỉ nền +0, ghi , DLAB = 1



Byte cao

Địa chỉ : Địa chỉ nền +1, ghi , DLAB = 1



Nội dung thanh ghi BRG :

| Nội dung thanh ghi (Hexa) | Tốc độ truyền (baud) |
|---------------------------|----------------------|
| 0900 | 50 |
| 0180 | 300 |
| 00C0 | 600 |
| 0060 | 1200 |
| ⋮ | ⋮ |
| 0020 | 3600 |
| ⋮ | ⋮ |
| 000C | 9600 |

e) Thanh ghi điều khiển giao diện với Modem MCR

MCR điều khiển giao diện với Modem và cho phép phát yêu cầu ngắt IRQ.

Địa chỉ : Địa chỉ nền +4, ghi

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D0- DTR bit :

D0 = 1 đặt DTR tích cực

D0 = 0 đặt DTR không tích cực

D1-RTS bit :

D1 = 1 đặt RTS tích cực

D1 = 0 đặt RTS không tích cực

D2- OUT1 bit : ảnh hưởng đến đầu ra OUT1

D3 - OUT2 bit : ảnh hưởng đến đầu ra OUT2

D3 = 1 đặt OUT 2 = 0, cho phép UART phát tín hiệu IRQ

D3 = 0 không cho phép UART phát tín hiệu IRQ

D4 : cho phép vào chế độ tự kiểm tra UART 8250 bằng cách nối tắt nội bộ các đầu Tx& Rx ; DTR & DSR ; RTS & CTS....

D5 = D6 = D7 = 0

f) Thanh ghi trạng thái đường truyền LSR

LSR cung cấp thông tin về trạng thái đường truyền tin.

Địa chỉ : Địa chỉ nền +5, đọc

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D0- RBR bit :

D0 = 1 báo UART nhận được 1 byte dữ liệu và đã đặt vào thanh ghi RBR, (báo RBR đầy).

D1- Lỗi đề

D1 = 1 báo RBR có dữ liệu nhận nhưng chưa được CPU đọc và bị 1 byte mới nhận tiếp theo đề lên.

D2 – Lỗi Parity

D2 = 1 báo có lỗi Parity (lỗi truyền tin).

D3 – Lỗi khuôn dạng

D3 = 1 báo khuôn dạng dữ liệu nhận bị sai.

D4 – Lỗi break

D4 = 1 báo đầu nhận dữ liệu có trạng thái logic 0 dài hơn thời gian truyền 1 byte dữ liệu.

D5 – THR bit :

D5 = 1 báo UART sẵn sàng nhận 1 byte dữ liệu mới từ CPU để phát đi (thanh ghi THR rỗng).

D6 – TSR bit :

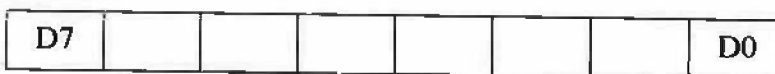
D6 = 1 báo thanh ghi TSR rỗng.

D7 – không dùng.

g) Thanh ghi trạng thái Modem MSR

MSR cung cấp thông tin về trạng thái các tín hiệu bắt tay từ Modem.

Địa chỉ : Địa chỉ nền +6, đọc



D0–D3 : báo có sự thay đổi trạng thái của các tín hiệu bắt tay từ modem

D4 – CTS bit :

D4 = 1 báo tín hiệu CTS có mức tích cực

D5 – DSR bit

D5 = 1 báo tín hiệu DSR có mức tích cực

D6 –RI bit :

D6 = 1 báo tín hiệu RI có mức tích cực

D7–CD–bit :

D7 = 1 báo tín hiệu CD có mức tích cực

h) Thanh ghi chọn nguồn ngắt IER

IER cho phép chọn các nguồn báo ngắt.

Địa chỉ : Địa chỉ nền +1, ghi , DLAB = 0

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D0 – liên quan đến báo ngắt của thanh ghi dữ liệu nhận RBR.

D0 = 1 cho phép UART báo ngắt khi nó nhận được dữ liệu (khi RBR đầy)

D0 = 0 không cho phép báo ngắt

D1– liên quan đến báo ngắt của thanh ghi phát THR

D1 = 1 cho phép UART báo ngắt khi nó sẵn sàng nhận 1 byte dữ liệu mới từ CPU để phát đi (khi THR rỗng)

D1 = 0 không cho phép

D2– liên quan đến báo ngắt của thanh ghi LSR

D2 = 1 cho phép UART báo ngắt khi có thay đổi thông tin về trạng thái truyền tin trong LSR

D2 = 0 không cho phép

D3 – liên quan đến báo ngắt của thanh ghi MSR

D3 = 1 cho phép UART báo ngắt khi có thay đổi thông tin về trạng thái modem trong MSR

D3 = 0 không cho phép báo ngắt

D4 – D7 = 0

i) Thanh ghi nhận dạng nguồn ngắt IIR

Thanh ghi IIR cho phép nhận dạng nguồn gây ngắt và kiểm tra trạng thái báo ngắt.

Địa chỉ : Địa chỉ nền +2, đọc

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D3÷D7 = 0

D0– trạng thái báo ngắt

D0 = 0 còn nguồn báo ngắt

D0 = 1 hết báo ngắt

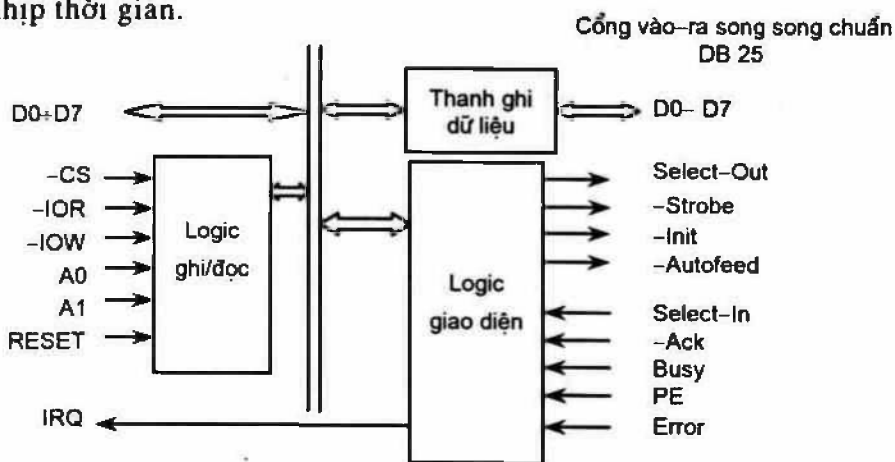
D1, D2 : tổ hợp giá trị hai bit này chỉ thị nguồn đang báo ngắt

| D2 | D1 | D0 | Nguồn báo ngắt | Mức ưu tiên | Phương pháp xoá báo ngắt |
|----|----|----|-------------------|-------------|--------------------------|
| 0 | 0 | 1 | Không có báo ngắt | | |
| 1 | 1 | 0 | LSR | Cao nhất | đọc LSR |
| 1 | 0 | 0 | RBR | nhì | đọc RBR |
| 0 | 1 | 0 | THR | ba | ghi THR |
| 0 | 0 | 0 | MSR | tư | đọc MSR |

4. THIẾT BỊ GIAO DIỆN SONG SONG

4.1 Thiết bị giao diện song song chuẩn

Thiết bị giao diện song song chuẩn (h.62) thực hiện chức năng giao diện giữa CPU và các thiết bị ngoại vi (trong đó có máy in) theo kiểu song song, trong đó các bit của 1 byte dữ liệu được đưa ra hoặc nhận vào CPU trong cùng một nhịp thời gian.



Hình 62

Các tín hiệu trên đầu nối song song chuẩn DB 25 :

| Chân số | Tín hiệu |
|---------|-------------|
| 1 | -Strobe |
| 2 + 9 | D0 + D7 |
| 10 | -Ack |
| 11 | Busy |
| 12 | Paper Error |
| 13 | - Select-In |
| 14 | -Autofeed |
| 15 | -Error |
| 16 | -Init |
| 17 | -Select-Out |
| 18 +25 | GND |

4.2 Các thanh ghi

Địa chỉ nền : 378h

a) Thanh ghi dữ liệu

Thanh ghi dữ liệu chứa dữ liệu xuất ra hoặc dữ liệu nhập vào.

Địa chỉ : Địa chỉ nền +0, đọc /ghi

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D7-D0 : các bit của byte dữ liệu.

b) Thanh ghi điều khiển

Thanh ghi điều khiển chứa các bit tạo tín hiệu bắt tay với thiết bị ngoại vi. Nếu thiết bị ngoại vi là máy in thì đây là các bit tạo tín hiệu bắt tay và điều khiển máy in.

Địa chỉ : Địa chỉ nền + 2, ghi

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D0 : STROBE-bit

D0 = 1 tạo tín hiệu STROBE với mức tích cực (thấp). Tín hiệu này được dùng để chốt dữ liệu vào thiết bị ngoại vi hoặc máy in.

D1 : AUTOFEED-bit

D1 = 1 tạo tín hiệu AUTOFEED tích cực (thấp). Với máy in : tín hiệu này điều khiển dịch một dòng in.

D2 : INIT – bit

D2 = 0 tạo tín hiệu xung INIT mức tích cực (thấp) với độ rộng 50 micro giây. Với máy in : tín hiệu này khởi động máy in.

D3 : SELECT-OUT-bit

D3 = 1 tạo tín hiệu SELECT-OUT tích cực (thấp). Tín hiệu SELECT-OUT được dùng để “bắt tay” với thiết bị ngoại vi hoặc chọn máy in.

D4 : IRQ-bit

D4 = 1 cho phép thiết bị giao diện song song chuẩn phát tín hiệu yêu cầu ngắt IRQ khi nhận được tín hiệu ACK chuyển từ mức “1” sang mức “0”.

D5-D7 : không dùng.

c) Thanh ghi trạng thái

Thanh ghi trạng thái chứa các bit thông tin phản ảnh trạng thái của các tín hiệu vào (trạng thái của thiết bị ngoại vi hoặc của máy in).

Địa chỉ : Địa chỉ nền + 1, đọc

| | | | | | | | |
|----|--|--|--|--|--|--|----|
| D7 | | | | | | | D0 |
|----|--|--|--|--|--|--|----|

D0-D2 : không dùng

D3 : ERROR-bit

D3 = 0 báo tín hiệu ERROR có mức tích cực thấp. Với máy in : báo máy in đang trong trạng thái có lỗi.

D4 : SELECT-IN- bit

D4 = 1 báo tín hiệu SELECT-IN tích cực (thấp). Với máy in : báo đã sẵn sàng làm việc.

D5 : PE– bit

D5 = 1 báo máy in gặp lỗi hết giấy, mức tích cực cao.

D6 : ACK–bit.

D6 = 0 báo tín hiệu xung ACK có mức tích cực thấp. Với máy in : tín hiệu xung ACK tích cực báo máy in sẵn sàng nhận một ký tự mới.

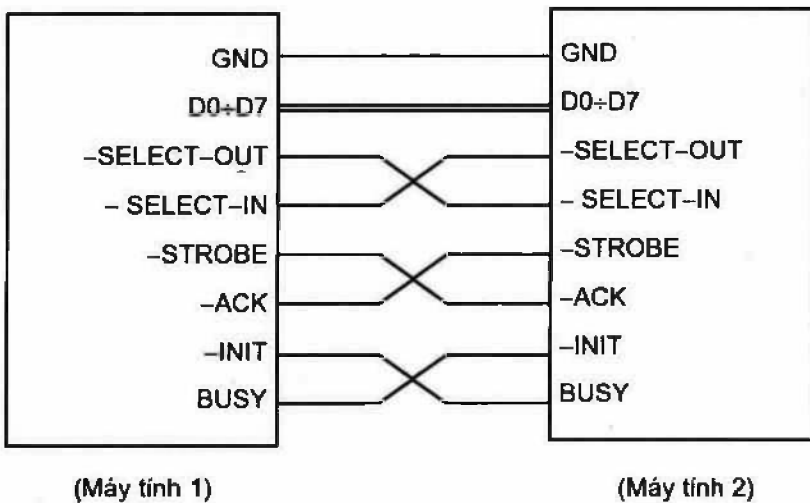
D7 : BUSY–bit

D7 = 0 báo tín hiệu BUSY có mức tích cực cao. Với máy in : báo máy in đang bận, không thể nhận dữ liệu.

D7 = "1" báo máy in không bận.

4.3 Kết nối và truyền dữ liệu trực tiếp giữa hai máy tính qua cổng song song

Có thể thực hiện kết nối trực tiếp và truyền dữ liệu giữa hai máy tính qua cổng song song chuẩn theo cách sau (h.63) :



Hình 63

Chương IV

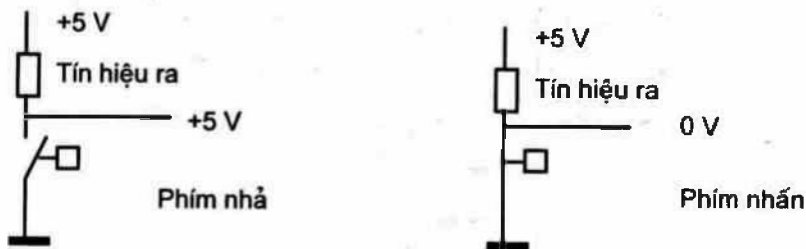
CÁC THIẾT BỊ NGOẠI VI

1. BÀN PHÍM

1.1 Công tác phím và phương pháp tạo mã quét

Bàn phím là thiết bị ngoại vi cho phép đưa thông tin vào máy tính dưới dạng mã ký tự. Bàn phím thực hiện chức năng chuyển thông tin dạng lực nhấn phím và vị trí của phím được nhấn thành mã phím và chuyển cho máy tính. Bàn phím gồm hai bộ phận chính là ma trận phím và mạch điện tử quét phím. Ma trận phím là tổ hợp các phím nhấn được sắp xếp theo các hàng và cột.

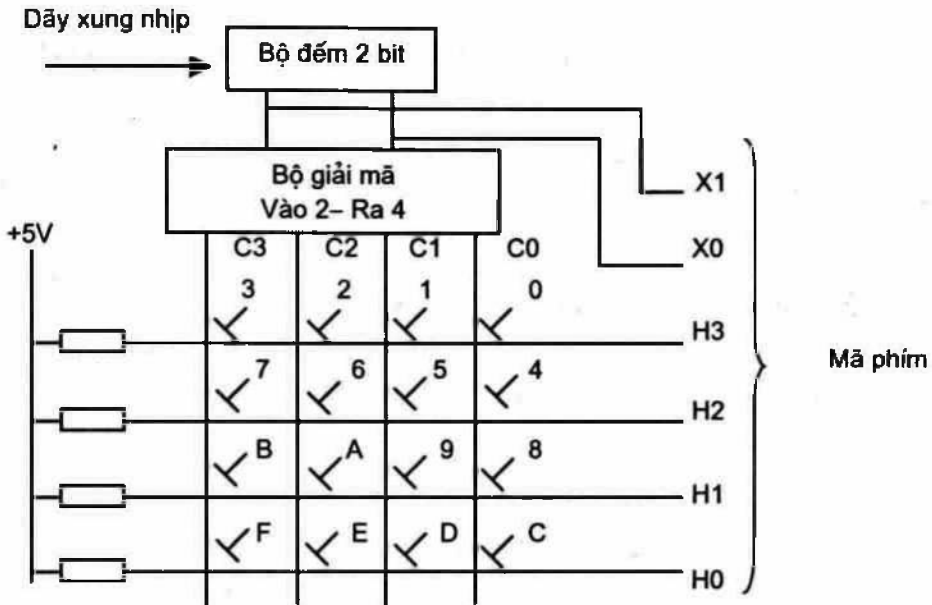
Phím nhấn có cấu trúc cơ điện như sau (h.64) :



Hình 64

Bình thường phím luôn ở trạng thái nhả, khi phím nhả thì hai tiếp điểm không được nối với nhau, đầu ra có mức điện áp dương tương ứng với mức logic 1. Khi phím được nhấn thì hai tiếp điểm được nối với nhau qua công tắc phím và đầu ra có mức điện áp bằng 0V tương ứng mức logic 0.

Để mỗi lần nhấn phím có một mã phím tương ứng được sinh ra, cần sắp xếp hệ thống phím dưới dạng ma trận phím. Cấu trúc của ma trận 16 phím cùng với mạch điện tử quét phím có dạng sau (h.65) :



Hình 65

Ma trận phím gồm các dây hàng và các dây cột giao nhau nhưng không tiếp xúc với nhau. Các công tắc phím được đặt ở chỗ giao của hàng và cột. Hai tiếp điểm của công tắc nằm ở trên hàng và cột tại chỗ giao nhau đó. Mỗi khi phím được nhấn thì hai dây hàng và cột được nối với nhau qua hai tiếp điểm và công tắc tại chỗ giao nhau.

Nguyên tắc quét phím và tạo mã như sau :

Đầu ra X0X1 của bộ đếm nhị phân 2 bit lần lượt cho ra các mã 00,01,10,11, ... mã này được đưa vào bộ giải mã 2 đầu vào 4 đầu ra. Ở đầu ra C3C2C1C0 của bộ giải mã sẽ lần lượt xuất hiện các giá trị 1110, 1101,1011,0111, ... Các dây cột Ci của ma trận phím lần lượt có mức logic 0, thao tác này được gọi là thao tác “quét” bàn phím. Khi không phím nào được nhấn thì tất cả các dây hàng H3H2H1H0 đều có mức logic 1. Khi có một phím nào đó được nhấn thì hai dây hàng và cột được nối với nhau tại chỗ phím đang nhấn, một mã tương ứng phím đang nhấn được tạo ra ở đầu ra X1X0H3H2H1H0.

Ví dụ :

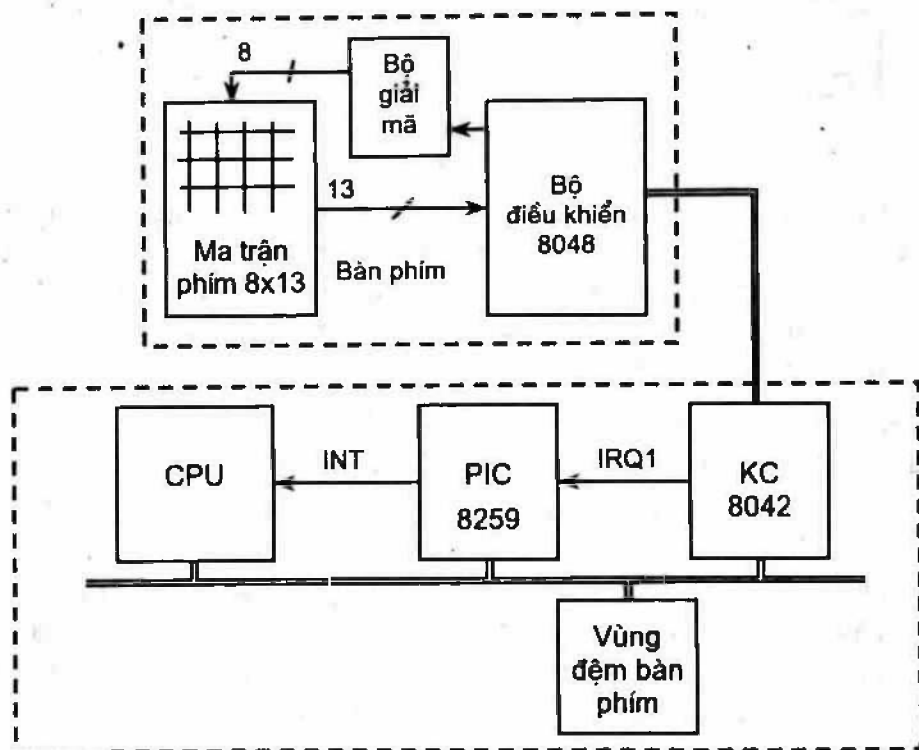
Nếu phím 6 được nhấn, tại thời điểm dây cột chứa phím 6 (dây C2) có mức logic 0 thì dây hàng chứa phím 6 (dây H2), do công tắc 6 đóng, cũng có mức logic 0. Mã đầu ra X1X0H3H2H1H0 = 101011.

Nếu phím A được nhấn, tại thời điểm dây cột chứa phím A (dây C2) có mức logic 0 thì dây hàng chứa phím A (dây H1), do công tắc A đóng, cũng có mức logic 0. Mã đầu ra $X1X0H3H2H1H0 = 101101$.

Nếu phím 7 được nhấn, tại thời điểm dây cột chứa phím 7 (dây C3) có mức logic 0 thì dây hàng chứa phím số 7 (dây H2), do công tắc 7 đóng, cũng có mức logic 0. Mã đầu ra $X1X0H3H2H1H0 = 111011$ v.v.

1.2 Hệ thống bàn phím của máy vi tính

Hệ thống bàn phím của máy vi tính gồm hai phần bàn phím và thiết bị giao diện bàn phím, được kết nối và trao đổi thông tin theo kiểu “chủ” “thợ”(h.66).



Hình 66

Bàn phím là tổ hợp của ma trận 8x13 phím và mạch điều khiển 8048. Mạch 8048 là một hệ vi xử lý nhỏ được tích hợp trên một đơn chip. Mạch 8048 bao gồm CPU, bộ nhớ ROM chứa chương trình điều khiển quét và tạo

mã phím, RAM chứa dữ liệu của chương trình điều khiển, hai cổng vào/ra P1 và P2, một cổng dữ liệu 8 bit. Mạch 8048 tuần tự đưa mã nhị phân 3 bit ra tại cổng P2, qua bộ giải mã Vào 3 Ra 8 tạo ra tín hiệu quét bàn phím. Tại thời điểm mã 3 bit được đưa ra, mạch 8048 thực hiện đọc tín hiệu 13 bit từ ma trận phím vào cổng P1, từ đây tạo ra mã phím (mã quét) của phím được nhấn. Khi phím được nhả một mã phím (mã quét) cũng được tạo ra bằng cách cộng mã phím nhấn với 80h.

Mạch 8048, được nuôi bằng nguồn từ máy tính, thực hiện trao đổi thông tin với thiết bị giao diện bàn phím KC 8042 theo kiểu nối tiếp đồng bộ. KC 8042 có cấu trúc tương tự mạch 8048. KC 8042 đóng vai trò "chủ", 8048 đóng vai trò "thợ" trong các quá trình truyền tin thông qua hai dây tín hiệu: dây "DATA" và dây "CLOCK".

Dây "DATA" truyền tín hiệu dữ liệu nối tiếp giữa 8048 và KC 8042. Tín hiệu nối tiếp bao gồm: bit START, 8 bit dữ liệu, 1 bit PARITY, 1 bit STOP. Quá trình trao đổi thông tin giữa 8048 và KC 8042 được đồng bộ bởi tín hiệu trên dây "CLOCK".

Quá trình truyền dữ liệu được thực hiện như sau:

Mạch 8048 luôn phải kiểm tra trạng thái truyền tin qua hai dây "DATA" và "CLOCK" trước khi phát đi mã phím. Khi KC 8042 đặt "DATA" = 0 và "CLOCK" = 1 thì 8048 phải nhận các chỉ lệnh từ KC 8042. Khi KC 8042 đặt "DATA" = 1 và "CLOCK" = 1 thì 8048 được quyền truyền mã phím cho máy tính. Quá trình truyền dữ liệu được đồng bộ bằng dây xung đồng bộ do 8048 phát ra trên dây "CLOCK".

Khi KC 8042 nhận được mã phím dạng nối tiếp, nó loại bỏ các bit tạo khung dữ liệu truyền, chuyển mã phím vào thanh ghi tạm và phát ra yêu cầu ngắt IRQ1 cho hệ thống ngắt cứng. Hệ thống ngắt cứng sẽ kích hoạt chương trình phục vụ bàn phím 09h (chương trình phục vụ ngắt 09h) nằm ở BIOS. Chương trình phục vụ bàn phím 09h có chức năng dịch mã phím thành mã hai byte và chứa vào vùng đệm bàn phím.

Chương trình phục vụ bàn phím 09h trước hết kiểm tra (mã) các phím trượt (Shift, Alt, Ctrl) và các phím đặc biệt (ScrollLock, NumLock, CapsLock, Insert) trước khi dịch mã phím sang mã hai byte.

Mã hai byte được chương trình phục vụ bàn phím 09h tạo ra có cấu trúc tùy thuộc mã phím hoặc tổ hợp mã phím nhận được. Nếu nhận được mã của

phím ký tự thì byte thấp của mã hai byte chứa mã ASCII của ký tự tương ứng, byte cao chứa mã phím (mã quét phím). Khi chương trình phục vụ bàn phím 09h nhận được mã các phím không phải là ký tự thì byte thấp của mã hai byte có giá trị 0, byte cao chứa mã phím mở rộng.

Vùng đệm bàn phím có kích thước 32 byte nằm trên bộ nhớ chính tại địa chỉ 0000h : 041Eh. Trạng thái của các phím trượt và các phím đặc biệt được chứa ở hai ô nhớ 0000h : 0417h và 0000h : 0418h. Có thể truy nhập vùng đệm bàn phím để đọc thông tin về bàn phím nhờ chương trình phục vụ ngắt 16h của BIOS.

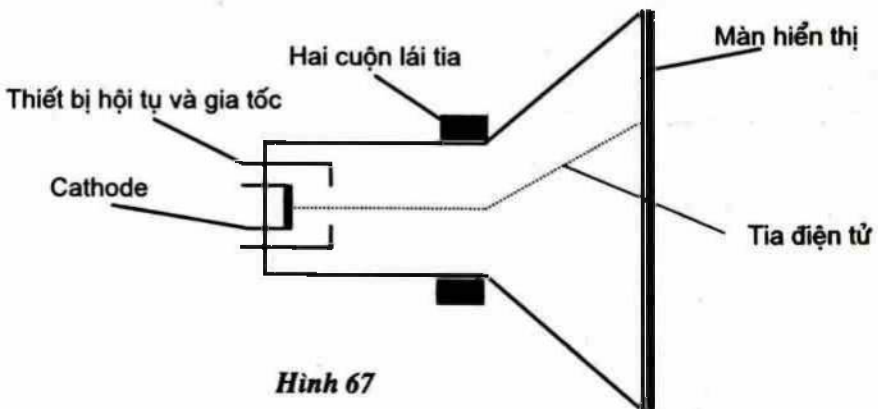
Chương trình phục vụ bàn phím 09h cũng xử lý các trường hợp đặc biệt như :

- khi phím được nhấn quá lâu (ví dụ quá 0.5 giây) và KC 8042 không nhận được mã phím nhà, nó sẽ gửi ra cho đơn vị xử lý trung tâm mã của phím được nhấn.
- khi nhận được tổ hợp các phím Ctrl+Alt+Del nó sẽ khởi động lại máy tính hoặc kết thúc một nhiệm vụ.
- khi nhận được mã phím PrintScreen nó sẽ kích hoạt ngắt 05h của BIOS.
- khi nhận được mã phím Ctrl+Break nó sẽ kích hoạt ngắt 1Bh của BIOS.

2. MÀN HÌNH

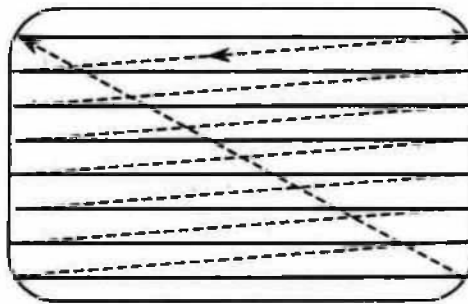
2.1 Màn hình ống tia âm cực CRT (Cathode Ray Tube)

Màn hình ống tia âm cực CRT là thiết bị hiển thị thông dụng nhất hiện nay. Màn hình CRT có cấu tạo như sau (h.67) :



Màn hình CRT là một ống thủy tinh chân không với các bộ phận : cathode phát xạ điện tử, ống phóng tia điện tử, cuộn lái tia và màn hiển thị. Cathode bằng kim loại được nối với điện áp âm , được đốt nóng và tạo ra các điện tử tự do. Màn hiển thị được phủ một lớp chất liệu phát quang và dẫn điện, được nối với điện áp dương và đóng vai trò một anode. Dưới tác dụng của điện trường cường độ cao trong ống phóng, điện tử rời khỏi cathode, được hội tụ thành chùm tia hướng về phía màn hiển thị. Cuộn lái tia có tác dụng lái chùm tia điện tử dịch chuyển theo hai chiều dọc và ngang màn hình. Khi chùm tia điện tử đập vào màn hiển thị sẽ tạo nên một điểm phát sáng. Cường độ điểm sáng phụ thuộc vào cường độ chùm tia và chất liệu phát sáng. Khi chùm tia mất đi hoặc chuyển hướng thì điểm vẫn còn lưu sáng một khoảng thời gian ngắn sau đó, thời gian lưu sáng phụ thuộc vào chất liệu phát sáng và cường độ chùm tia.

Ảnh trên màn hình CRT được tạo từ các điểm ảnh. Điểm ảnh được tạo ra khi cường độ chùm tia điện tử được tăng lên, điểm ảnh không xuất hiện khi chùm tia bị tắt đi. Các điểm ảnh được tạo theo từng dòng, từ trên xuống dưới. Một ảnh hoàn chỉnh được tạo ra trên màn hiển thị bởi các dòng chứa các điểm ảnh. Các điểm ảnh chỉ tồn tại trong một thời gian rất ngắn. Để có thể quan sát được ảnh cần làm tươi các điểm ảnh theo một chu kỳ xác định. Các điểm ảnh được làm tươi theo từng dòng, bắt đầu từ dòng thứ nhất. Các dòng được làm tươi tuần tự từ trên xuống dưới. Khi dòng cuối cùng được quét xong, quá trình làm tươi được bắt đầu lại từ dòng đầu tiên (h.68).

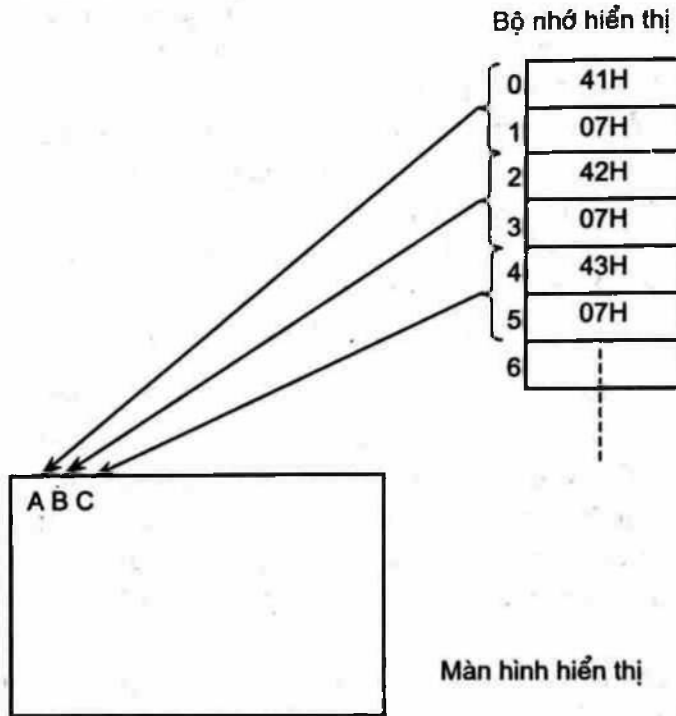


Hình 68

2.2 Hiển thị ở chế độ văn bản

Các thiết bị hiển thị được sử dụng ở máy vi tính PC đều là loại ánh xạ bộ nhớ. Bộ nhớ này được cả đơn vị xử lý trung tâm và thiết bị điều khiển màn hình cùng truy nhập và được gọi là bộ nhớ hiển thị. Thông tin cần hiển thị

được đưa ra bộ nhớ hiển thị, thiết bị điều khiển màn hình CRTC liên tục đọc bộ nhớ này để đưa ra màn hình. Hình 69 minh họa nguyên tắc ánh xạ từ bộ nhớ hiển thị ra màn hình trong chế độ văn bản :

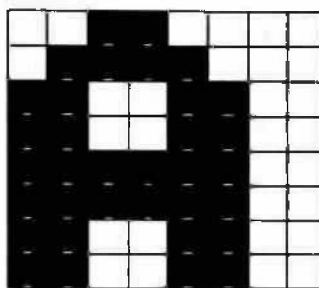


Hình 69

Mỗi một ký tự trên màn hình là một ánh xạ của một ô nhớ hai byte trong bộ nhớ hiển thị. Byte đầu chứa mã ASCII của ký tự, byte thứ hai chứa thuộc tính (màu nền, màu chữ, có/không nhấp nháy) của ký tự. Vị trí của mã ký tự trong bộ nhớ xác định vị trí ký tự trên màn hình. Mã ký tự đầu tiên trong bộ nhớ hiển thị (ví dụ : mã 41H) được ánh xạ thành ký tự (ký tự A) lên góc trái trên của màn hiển thị, mã ký tự tiếp theo được ánh xạ thành ký tự tiếp theo v.v.

Phương pháp ánh xạ bộ nhớ cho phép chương trình máy tính có thể dễ dàng thay đổi nội dung màn hiển thị bằng cách thay đổi nội dung của bộ nhớ hiển thị.

Mỗi ký tự được hiển thị trên màn hình dưới dạng một ma trận 8x8 điểm ảnh sáng/tối như trên hình vẽ (h.70) :



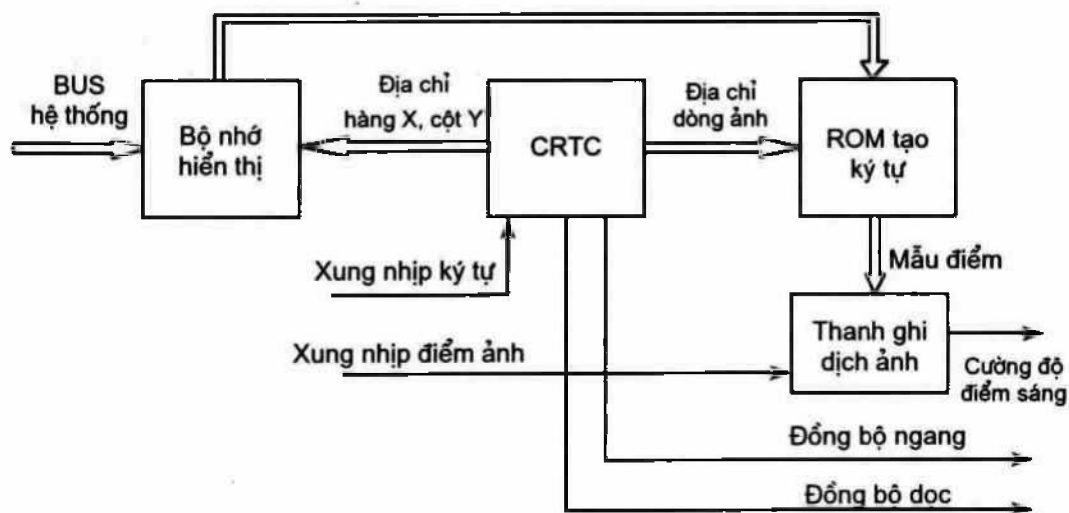
Hình 70

Phương pháp hiển thị ánh xạ bộ nhớ không hoàn toàn phù hợp với việc hiển thị các đối tượng có hình dạng không bình thường và chuyển động nhanh, đáp ứng thời gian thực bị chậm vì cần phải thao tác nhiều điểm ảnh để dịch chuyển đối tượng.

2.3 Bộ điều khiển màn hình CRTC

Thiết bị giao diện màn hình (bộ điều khiển màn hình) CRTC thực hiện việc chuyển mã ký tự trong bộ nhớ hiển thị thành ký tự hiện trên màn hình. Ở chế độ văn bản các mẫu ký tự chỉ được hiển thị ở các vị trí hàng và cột cố định (25 hàng x 80 cột).

Sơ đồ nguyên lý của thiết bị giao diện màn hình ở chế độ văn bản như sau (h.71) :



Hình 71

Mỗi một ký tự trên màn hình chứa nhiều hàng điểm ảnh. CRTIC có nhiệm vụ chuyển mỗi mã ASCII trong bộ nhớ hiển thị thành chuỗi các mẫu điểm ảnh, đưa mỗi mẫu nằm lên một dòng màn hình. Điều này được thực hiện nhờ bộ ROM tạo ký tự. ROM tạo ký tự chứa các hộp mẫu ký tự, mỗi hộp mẫu ký tự có kích thước 8 byte mang thông tin về ma trận điểm ảnh của một ký tự. Ví dụ hộp mẫu ký tự A có dạng sau :

```

00110000
01111000
11001100
11001100
11111100
11111100
11001100
11001100

```

Nếu cần hiển thị 256 ký tự ASCII cần một ROM 2Kbyte, đủ chứa 256 hộp mẫu ký tự, mỗi hộp mẫu chiếm 8 ô nhớ liền nhau. Các hộp mẫu ký tự trong bộ ROM tạo ký tự được định vị bằng địa chỉ 11 bit, trong đó 8 bit địa chỉ cao xác định vị trí của hộp trong ROM, 3 bit địa chỉ thấp xác định vị trí của từng byte mẫu điểm ảnh trong hộp đó. Các mẫu ký tự được đặt trong ROM theo trật tự của bảng mã ASCII.

Nguyên lý hoạt động của thiết bị giao diện màn hình trong chế độ văn bản như sau : Giả sử cần hiển thị hai ký tự A và B tại các vị trí hàng 0 – cột 0 và hàng 0 – cột 1 trên màn hình. Mã ASCII của hai ký tự được đặt tại hai vị trí tương ứng trong bộ nhớ hiển thị (xem hình vẽ ở mục 2.2).

CRTIC gửi địa chỉ hàng và cột màn hình cho bộ nhớ hiển thị (hàng = 0, cột = 0). Bộ nhớ hiển thị gửi mã ASCII của ký tự (ký tự A) cho ROM, mã ASCII của ký tự mang thông tin về địa chỉ của hộp mẫu ký tự trong ROM (8 bit địa chỉ cao). Tại cùng thời điểm này CRTIC gửi địa chỉ của dòng mẫu điểm ảnh (dòng mẫu điểm 0) cho ROM (3 bit địa chỉ thấp). Hai địa chỉ này được kết hợp lại tạo thành địa chỉ (11 bit) cho phép truy nhập vào dòng mẫu điểm ảnh đầu tiên của ký tự (ký tự A) trong ROM và xuất nó ra thanh ghi dịch ảnh. Từ thanh ghi dịch ảnh, từng bit mẫu ảnh tuần tự được đưa ra màn hình.

Khi tất cả các bit mẫu ảnh từ thanh ghi dịch được đẩy ra màn hình, CRTC tiếp tục gửi địa chỉ hàng–cột (hàng = 0, cột = 1) cho bộ nhớ hiển thị và gửi địa chỉ dòng mẫu điểm ảnh (dòng mẫu điểm 0) cho ROM, bộ nhớ hiển thị gửi mã ASCII của ký tự (ký tự B) cho ROM. Dòng mẫu điểm ảnh đầu tiên của ký tự (ký tự B) được xuất ra thanh ghi dịch ảnh. Tương tự như thế các dòng mẫu điểm đầu tiên của tất cả các ký tự trên cùng một hàng màn hình được hiển thị, cho đến ký tự cuối cùng trên hàng.

CRTC tiếp tục gửi địa chỉ hàng–cột (hàng = 0, cột = 0) đến bộ nhớ hiển thị, nhưng địa chỉ dòng mẫu điểm ảnh bây giờ là 1 (dòng mẫu điểm 1) cho ROM. Bộ nhớ hiển thị gửi mã ASCII của ký tự A cho ROM, ROM xuất ra dòng mẫu điểm ảnh 1 của ký tự A. Dòng 1 của ký tự B được xuất ra theo cách tương tự. Các dòng điểm ảnh tiếp theo của ký tự lần lượt được hiển thị lên màn hình cho đến khi tất cả các dòng điểm ảnh của hàng văn bản đầu tiên (hàng 0) được hiển thị trên màn hình.

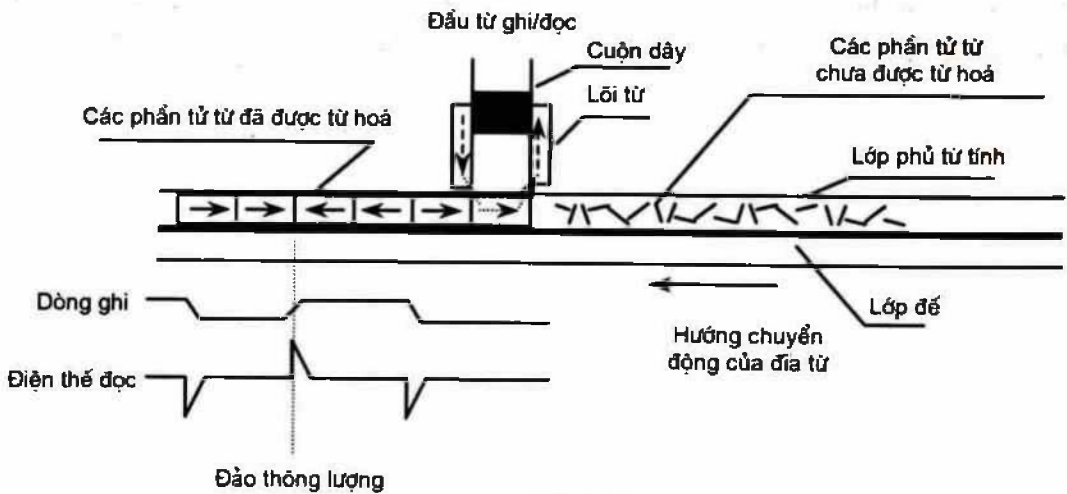
Các hàng văn bản tiếp theo cũng được hiển thị theo phương pháp nói trên.

Trên thực tế hoạt động của CRTC phức tạp hơn. CRTC phải có khả năng hiển thị ở chế độ đồ họa. CRTC phải theo dõi thông tin về thuộc tính của ký tự hiển thị, phải tạo ra điểm nháy. CRTC cũng phải tạo ra hai tín hiệu đồng bộ ảnh ngang – dọc và làm tươi màn hình. Tần số làm tươi tối thiểu là 50 Hz.

Chương V THIẾT BỊ ĐĨA TỪ

1. NGUYÊN LÝ LƯU TRỮ THÔNG TIN TRÊN ĐĨA TỪ

Các thiết bị lưu trữ thông tin như ổ đĩa mềm và ổ đĩa cứng đều thực hiện ghi/đọc thông tin trên cơ sở hiện tượng cảm ứng điện từ. Khi có một dòng điện chạy qua dây dẫn thì xuất hiện từ trường quanh dây dẫn này. Khi chiều dòng điện trong dây thay đổi thì cực của từ trường cũng thay đổi theo. Hiện tượng điện từ có tính hai chiều, khi dây dẫn nằm trong một từ trường biến thiên sẽ có dòng điện cảm ứng xuất hiện trong dây dẫn này. Từ trường sinh ra bởi dây dẫn có dòng điện chạy qua có thể tác động một lực từ lên các chất từ tính nằm trong từ trường này (h.72).



Hình 72

Thiết bị lưu trữ kiểu đĩa từ gồm hai bộ phận chính là đầu từ ghi/đọc và đĩa từ. Đầu từ ghi/đọc là một cuộn dây có lõi hình chữ U bằng chất liệu từ. Đĩa từ, vật lưu trữ thông tin, là một đĩa được làm từ chất dẻo hoặc nhôm và được phủ

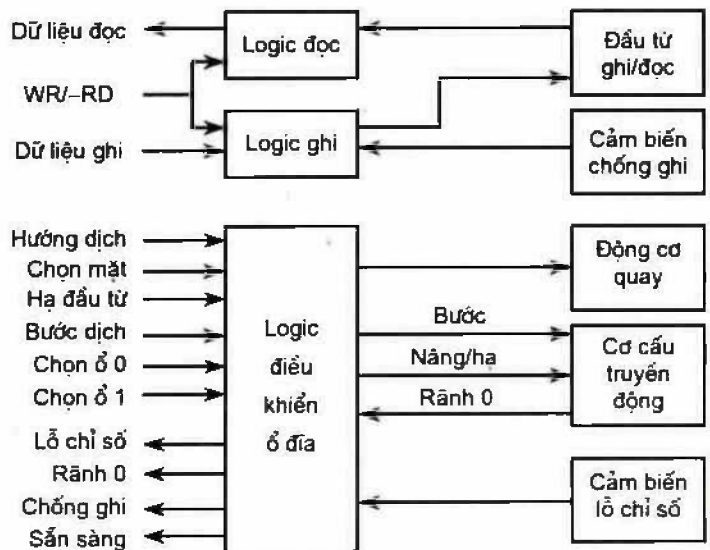
một chất liệu từ tính. Khi mới được chế tạo mỗi phần tử từ trên đĩa có một từ trường riêng, cực của các phần tử từ có hướng ngẫu nhiên, do vậy từ trường của chúng triệt tiêu lẫn nhau và không quan sát được.

Trong quá trình ghi thông tin một dòng điện được đưa vào cuộn dây, sinh ra một từ trường trong lõi dây. Đường cảm ứng từ đi qua một cực của lõi chữ U, đi qua lớp phủ từ tính trên đĩa, trở lại lõi dây qua cực thứ hai. Khi từ trường đi qua một vùng các phần tử từ tính trên đĩa sẽ làm định hướng lại cực từ của chúng. Tổng từ trường của các phần tử từ trong một vùng đã được từ hóa theo cùng hướng lúc này trở nên quan sát được (mang tải được thông tin). Khi dòng điện trong cuộn dây đảo chiều thì cũng sinh ra đảo chiều phân cực của các vùng từ tính trên mặt đĩa, hiện tượng này gọi là đảo từ thông. Các đảo từ thông tạo nên ranh giới giữa các vùng từ hóa. Việc tạo ra các đảo từ thông được bộ điều khiển ổ đĩa dùng để mã hóa việc ghi dữ liệu lên đĩa. Các phương pháp mã hóa được dùng là MF, MFM, RLL. Khi đọc, đầu từ chuyển động song song với từ trường đã được tạo ra trên mặt đĩa, vì thế dòng điện cảm ứng chỉ xuất hiện trong cuộn dây khi đầu từ đi ngang qua một đảo từ thông. Trong thao tác đọc, nhờ phát hiện được các đảo từ thông mà bộ điều khiển giải mã và khôi phục lại được thông tin đã ghi.

2. THIẾT BỊ ĐĨA MỀM

2.1 Ổ đĩa mềm

Ổ đĩa mềm có cấu trúc như sau (h.73) :



Hình 73

Ổ đĩa mềm có hai đầu từ ghi/đọc, mỗi đầu từ có hai khe ghi/đọc và xóa nằm so le với nhau. Khi ghi dữ liệu thì đầu xoá đồng thời sẽ xoá hết vệt nằm bên ngoài và tạo nên một vệt cách ly giữa các rãnh. Đầu ghi/đọc được dịch chuyển trên bề mặt đĩa mềm bởi một cơ cấu truyền động đầu từ.

Cơ cấu truyền động đầu từ sử dụng động cơ bước. Loại động cơ này mỗi lần dịch chuyển chỉ quay một góc nhất định, được gọi là một bước. Lượng bước được xác định bởi lượng xung điện cấp cho động cơ. Logic điều khiển ổ đĩa có thể dễ dàng điều khiển động cơ dịch chuyển đầu từ đến các rãnh theo lượng bước xác định trước.

Động cơ quay có chức năng quay đĩa mềm với một vận tốc cố định, thường là 300 vòng/phút. Nhờ đĩa từ được quay liên tục nên khi ghi dữ liệu được phân bố đều trên rãnh và khi đọc sẽ tạo nên một từ trường biến thiên ở đầu đọc.

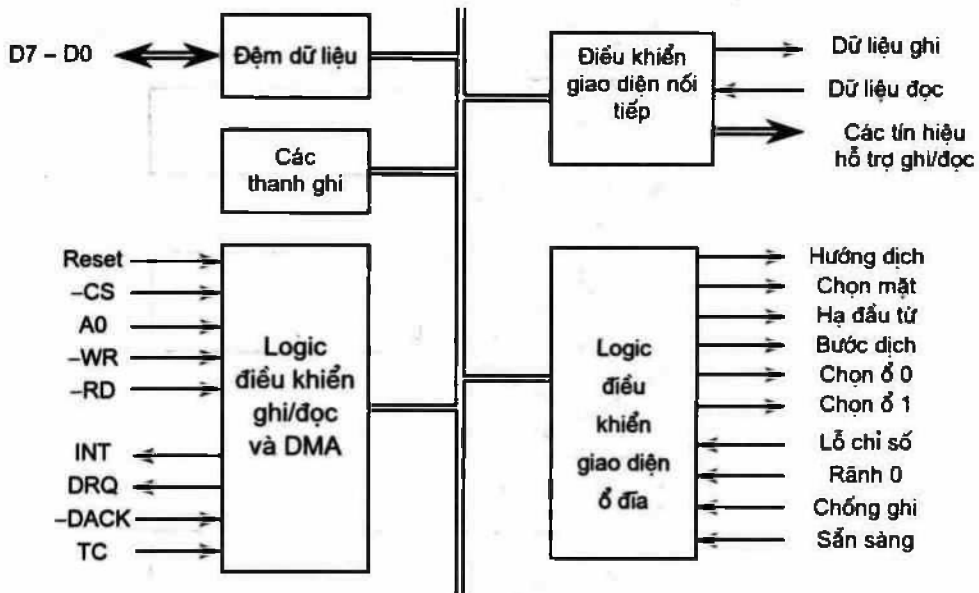
Logic ghi/đọc tạo tín hiệu ghi, nhận và khuếch đại tín hiệu đọc.

Logic điều khiển một mặt tạo giao diện giữa ổ đĩa và thiết bị giao diện đĩa mềm FDC, mặt kia tạo các tín hiệu điều khiển động cơ quay, cơ cấu truyền động đầu từ, nhận các tín hiệu cảm biến lỗ chỉ số và cảm biến chống ghi.

2.2 Thiết bị giao diện đĩa mềm FDC (Floppy Disk Controller)

Giao diện chuẩn của các ổ đĩa mềm là giao diện SA 400. Thiết bị giao diện đĩa mềm được xây dựng trên cơ sở vi mạch điều khiển chuyên dụng μ PD 765 của hãng NEC. Thiết bị giao diện đĩa mềm FDC thực hiện chức năng giao diện giữa đơn vị xử lý trung tâm và ổ đĩa mềm. FDC thực hiện giao diện với đơn vị xử lý trung tâm qua các tín hiệu điều khiển chuẩn của đơn vị xử lý trung tâm và qua một hệ 15 lệnh. FDC có khả năng làm việc ở cả hai chế độ DMA và không DMA. Ở chế độ không DMA FDC tạo ra tín hiệu ngắt mỗi khi byte dữ liệu được truyền. Ở chế độ DMA đơn vị xử lý trung tâm chỉ cần nạp lệnh cho FDC và toàn bộ việc truyền dữ liệu được thực hiện dưới sự điều khiển của FDC và DMAC.

Sơ đồ khối chức năng của vi mạch điều khiển chuyên dụng μ PD 765 như sau (h.74) :



Hình 74

FDC thực thi được 15 lệnh : ghi dữ liệu, đọc dữ liệu, đọc một rãnh, đọc trường nhận dạng trong một cung, tạo khuôn dạng một rãnh, xác lập kiểu mã hoá dữ liệu ghi/đọc, dịch chuyển đầu từ v.v.

Ví dụ, lệnh ghi dữ liệu chứa các thông số như : rãnh, mặt, vị trí cung và độ dài một cung chứa dữ liệu cần ghi. Khi lệnh ghi được nạp cho FDC, FDC bắt đầu đọc các trường địa chỉ trên rãnh, đến khi gặp trường địa chỉ phù hợp với địa chỉ yêu cầu FDC bắt đầu lấy từng byte dữ liệu đơn vị xử lý trung tâm và đưa ra ổ đĩa.

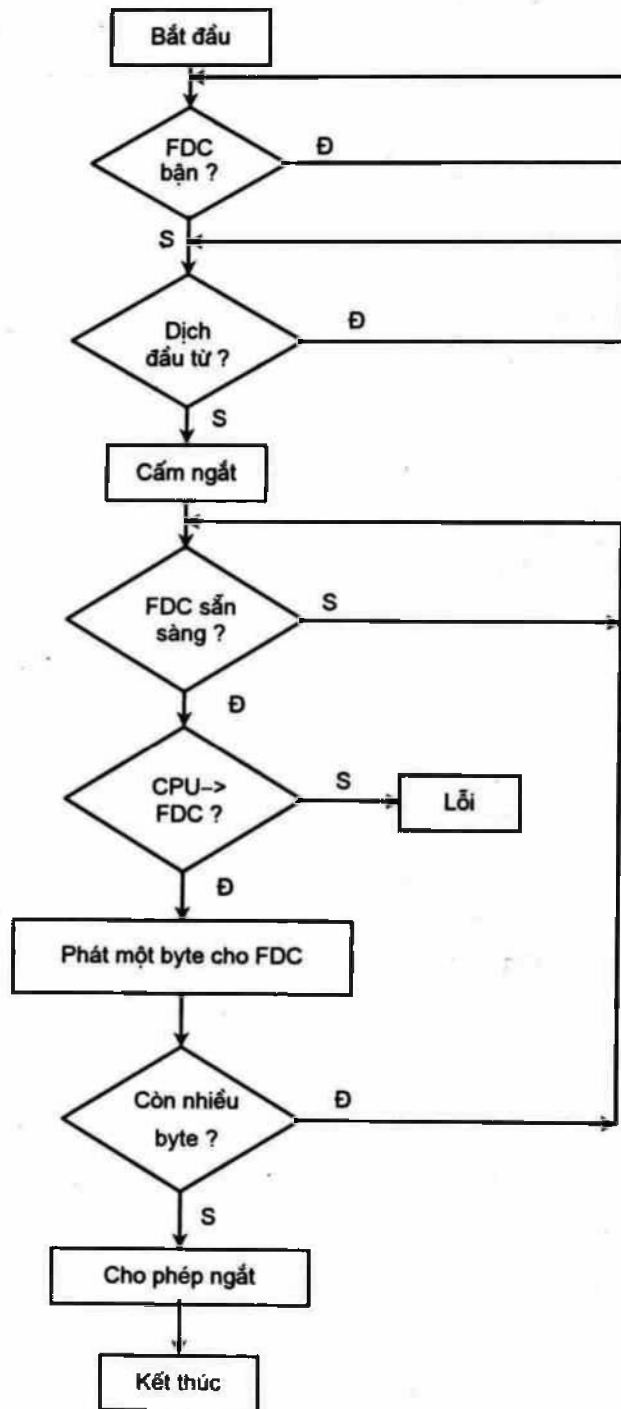
Quá trình thực hiện một lệnh trải qua 3 giai đoạn :

Giai đoạn nạp lệnh : FDC nhận lệnh từ Bộ xử lý trung tâm, giải mã để nhận được các thông tin về các thao tác cần thiết khi thực thi lệnh này.

Giai đoạn thực hiện : FDC thực hiện các thao tác thực thi lệnh.

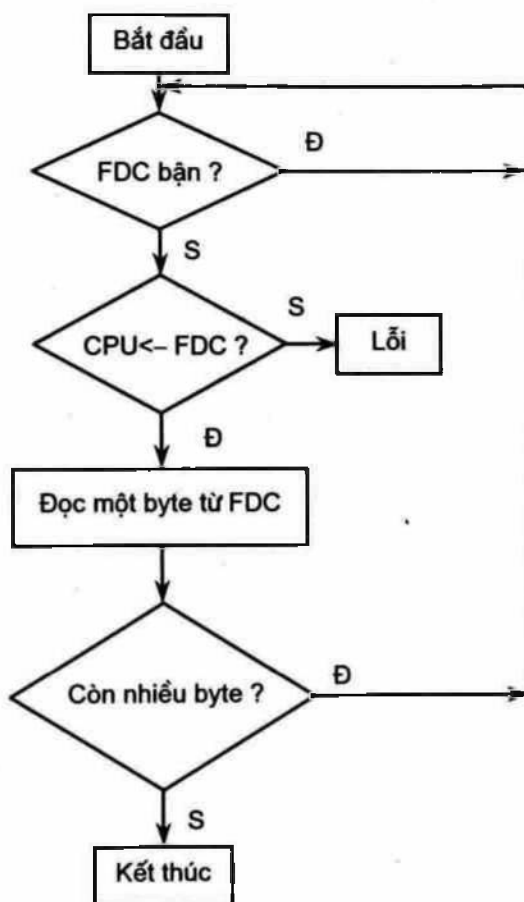
Giai đoạn kết quả : Sau khi lệnh được thực thi, trạng thái thực hiện lệnh được chuyển về cho Bộ xử lý trung tâm.

Giai đoạn nạp lệnh cho FDC được thực hiện theo lưu đồ sau (h.75) :



Hình 75

Lưu đồ thực hiện giai đoạn kết quả (h.76) :



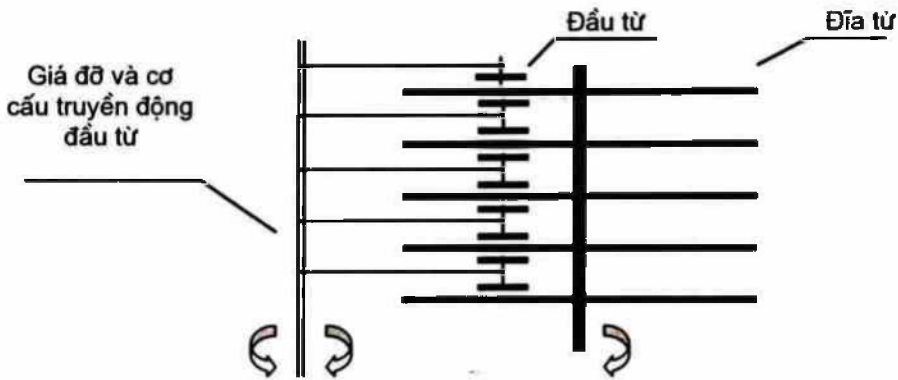
Hình 76

3. THIẾT BỊ ĐĨA CỨNG VÀ GIAO DIỆN IDE

3.1 Ổ đĩa cứng

Đĩa cứng là một thiết bị lưu trữ kiểu từ tính. Ổ đĩa cứng gồm các lá đĩa bằng hợp kim nhôm hoặc thủy tinh (h.77). Mặt lá đĩa được phủ một lớp màng mỏng ô xít kim loại có khả năng từ hóa. Các lá đĩa cứng được gắn kết đồng tâm trên một trục và được quay bởi động cơ một chiều với tốc độ 3600

vòng/phút, một số ổ đĩa có tốc độ quay tới 5400, 6400 và 7200 vòng/phút. Mỗi ổ đĩa có từ 2 đến 5 lá đĩa, một số ổ đĩa có tới 11 lá đĩa. Mỗi lá đĩa có hai mặt, các mặt đĩa được đánh số thứ tự từ 0, 1, 2, v.v Mỗi mặt đĩa có một đầu từ ghi/đọc, các đầu từ ghi và đọc dữ liệu trên đĩa theo những vòng tròn đồng tâm, được gọi là rãnh (track). Do các lá đĩa cứng được gắn chặt trên một trục, do cơ cấu điều khiển dịch chuyển đầu từ có độ chính xác cao, nên mật độ rãnh thực có thể lên tới 30000 rãnh/inch.



Hình 77

Các đầu từ ghi/đọc được gắn trên một cơ cấu dịch chuyển duy nhất, nên chúng được dịch chuyển đồng thời trên các lá đĩa. Mỗi một đầu từ nằm trên một cánh tay của cơ cấu truyền động và được ép sát mặt đĩa khi đĩa không quay. Khi ổ đĩa quay với tốc độ cao thì áp lực không khí hình thành giữa mặt đĩa và đầu từ sẽ nâng đầu từ lên cách mặt đĩa khoảng $3 \div 10 \mu\text{inch}$. Các loại đầu từ kiểu ferit có kích thước lớn và nặng nề nên không thích hợp cho việc ghi/đọc dữ liệu với mật độ và tốc độ cao ở ổ cứng. Trong các ổ cứng thường dùng đầu từ phim mỏng TF (Thin Film). Đầu từ TF được sản xuất theo công nghệ sản xuất mạch vi điện tử nên có chất lượng cao và rất nhỏ. Do nhỏ và nhẹ mà chúng có thể nổi trên bề mặt lá đĩa ở độ cao chưa tới $3 \mu\text{inch}$, điều này tạo khả năng ghi dữ liệu với mật độ cao hơn, đồng thời lại cảm nhận tín hiệu khi đọc mạnh hơn.

Cơ cấu dịch chuyển đầu từ để tìm kiếm rãnh ở ổ đĩa cứng khác hẳn so với ổ đĩa mềm. Động cơ dịch chuyển đầu từ không theo bước mà hoạt động theo cơ chế truyền động có phản hồi. Thông tin phản hồi nhận được trong quá trình

dịch chuyển đầu từ là nhờ có một thông tin đặc biệt được ghi sẵn trên đĩa cứng khi sản xuất đĩa. Thông tin về vị trí rãnh được ghi (nhúng) trên từng cung của đĩa dưới dạng mã Gray. Cơ cấu truyền động có dùng thông tin định vị nhúng giữa các cung được gọi là cơ cấu truyền động nhúng (Embedded Servo). Nhờ nhận được thông tin phản hồi liên tục về vị trí rãnh trong quá trình dịch chuyển mà việc định vị đầu từ được thực hiện chính xác và nhanh chóng.

3.2 Giao diện IDE

Giao diện IDE (Integrated Drive Electronic – mạch điện tử tích hợp trên thiết bị) là giao diện được dùng để kết nối thiết bị đĩa cứng với BUS hệ thống của máy tính. IDE là thuật ngữ dùng để mô tả việc mạch điện tử điều khiển giao diện HDC được gắn ngay cùng với ổ đĩa HDD trong thiết bị đĩa cứng.

Trước đây đã xuất hiện nhiều loại giao diện đĩa cứng. Chức năng của thiết bị giao diện đĩa cứng HDC là thực hiện truyền dữ liệu từ CPU đến ổ đĩa và nhận dữ liệu từ ổ đĩa cung cấp cho CPU. Tốc độ truyền dữ liệu với ổ cứng là một thông số quan trọng và ảnh hưởng đến tốc độ hoạt động của máy tính. Tốc độ truyền dữ liệu phụ thuộc nhiều vào cách giao diện giữa ổ cứng và CPU. Loại thiết bị đĩa cứng có bộ điều khiển giao diện tích hợp ngay trong thiết bị được gọi là ổ đĩa IDE hoặc ổ đĩa ATA IDE (AT attachment IDE). Việc đặt thiết bị giao diện, bao gồm các thiết bị tạo các tín hiệu điều khiển ổ đĩa và thiết bị mã hoá/giải mã dạng số/tương tự, ngay trong ổ đĩa làm cho ổ ATA IDE có độ tin cậy cao hơn so với kiểu thiết bị giao diện nằm độc lập và được nối với ổ đĩa bằng cáp (như ở thiết bị đĩa mềm). Thông tin dạng tương tự vốn dễ bị nhiễu loạn, đặc biệt khi đường truyền dài. Giao diện IDE làm cho đường dẫn tín hiệu ngắn nên tránh được nhiễu trên đường truyền. Cấu hình tích hợp cũng làm cho việc mã hóa dữ liệu từ số sang tương tự và ngược lại được làm trực tiếp nên tăng được tốc độ đồng hồ của bộ mã hóa cũng như mật độ lưu trữ dữ liệu. Việc tích hợp thiết bị giao diện trên ổ đĩa và thực hiện giao diện với hệ thống BUS của máy tính theo chuẩn IDE còn cho phép các nhà sản xuất ổ đĩa có thể độc lập thiết kế phát triển thiết bị đĩa cứng có các tính năng ngày càng mạnh hơn.

Phân bố tín hiệu trên đầu nối ATA IDE và BUS IDE như sau :

| Chân | Tín hiệu | Chân | Tín hiệu |
|------|--------------|------|------------|
| 1 | -RESET | 2 | GND |
| 3 | D7 | 4 | D8 |
| 5 | D6 | 6 | D9 |
| 7 | D5 | 8 | D10 |
| 9 | D4 | 10 | D11 |
| 11 | D3 | 12 | D12 |
| 13 | D2 | 14 | D13 |
| 15 | D1 | 16 | D14 |
| 17 | D0 | 18 | D15 |
| 19 | GND | 20 | để trống |
| 21 | DRQ3 | 22 | GND |
| 23 | -IOW | 24 | GND |
| 25 | -IOR | 26 | GND |
| 27 | I/O CH READY | 28 | Dự phòng |
| 29 | -DACK3 | 30 | GND |
| 31 | IRQ14 | 32 | -BUSIOCS16 |
| 33 | A1 | 34 | DRV READY |
| 35 | A0 | 36 | A2 |
| 37 | -CS1 | 38 | -CS3 |
| 39 | -DA/SP | 40 | GND |

Tín hiệu RESET được dùng để khởi động lại ổ đĩa. Dữ liệu được truyền qua BUS D0 ÷ D15. Quá trình truyền dữ liệu được thực hiện theo phương pháp truy nhập trực tiếp bộ nhớ (phương pháp DMA) thông qua các tín hiệu DRQ

và DACK. Hướng truyền dữ liệu phụ thuộc tín hiệu IOW và IOR. Tín hiệu DRV READY báo ổ đĩa sẵn sàng hoạt động. Tín hiệu DA/SP ở thời điểm khởi động máy tính cho biết ổ đĩa thợ có mặt hay không. Khi ổ đĩa hoạt động tín hiệu DA/SP được dùng để báo trạng thái hoạt động của ổ đĩa. Ổ đĩa IDE có các thanh ghi, địa chỉ của các thanh ghi này được xác định bởi các tín hiệu A0, A1, A2.

Thiết bị giao diện IDE có một tập các thanh ghi cho phép điều khiển và giao diện với ổ đĩa.

Thanh ghi dữ liệu :

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|----|

Thanh ghi dữ liệu được dùng để gửi lệnh cho thiết bị giao diện, còn các dữ liệu ghi/đọc được chuyển theo chế độ DMA.

Thanh ghi điều khiển chọn điều khiển ổ đĩa và đầu từ :

| | | | | | | | |
|---|---|---|-----|----|----|----|----|
| 1 | 0 | 1 | DRV | H3 | H2 | H1 | H0 |
|---|---|---|-----|----|----|----|----|

DRV chọn ổ đĩa

H3 – H0 chọn đầu từ

Thanh ghi điều khiển phụ :

| | | | | | | | |
|---|---|---|---|---|-----|-----|---|
| X | X | X | X | X | RST | INT | X |
|---|---|---|---|---|-----|-----|---|

RST khởi động mềm

INT cho phép báo ngắt. INT = 1 cấm thiết bị giao diện báo ngắt.

Ba thanh ghi số cung, địa chỉ cung và địa chỉ rãnh cần ghi/đọc.

Thanh ghi trạng thái :

| | | | | | | | |
|-----|-----|-----|----|------|-----|-----|-----|
| BSY | RDY | WFT | SK | DTRQ | COR | IDX | ERR |
|-----|-----|-----|----|------|-----|-----|-----|

BSY ổ đĩa bận

RDY ổ đĩa sẵn sàng

- WFT lỗi ghi
- SK trạng thái dịch chuyển đầu từ
- DTRQ yêu cầu truyền dữ liệu
- COR lỗi dữ liệu chưa được
- IDX index
- ERR có lỗi. Mã lỗi nằm ở thanh ghi trạng thái lỗi.

Thanh ghi trạng thái lỗi :

| | | | | | | | |
|-----|-----|-----|---|-----|---|----|----|
| NDM | NTR | ABT | X | NID | X | UC | BK |
|-----|-----|-----|---|-----|---|----|----|

- NMD không tìm được vùng DM
- NTR không tìm được rãnh
- ABT lệnh bị ngắt
- NID không tìm được vùng ID
- UC lỗi dữ liệu không khôi phục
- BK cung có lỗi

Quá trình thực hiện một lệnh trải qua 3 giai đoạn :

Giai đoạn nạp lệnh : HDC nhận lệnh và các thông số điều khiển từ đơn vị xử lý trung tâm, giải mã để có các thông tin về những thao tác cần thiết khi thực thi lệnh này.

Giai đoạn thực hiện : HDC thực hiện các thao tác thực thi lệnh.

Giai đoạn kết quả : sau khi lệnh được thực thi, trạng thái thực hiện lệnh được chuyển về cho Bộ xử lý trung tâm.

4. TỔ CHỨC LƯU TRỮ THÔNG TIN TRÊN ĐĨA TỪ Ở MỨC VẬT LÝ

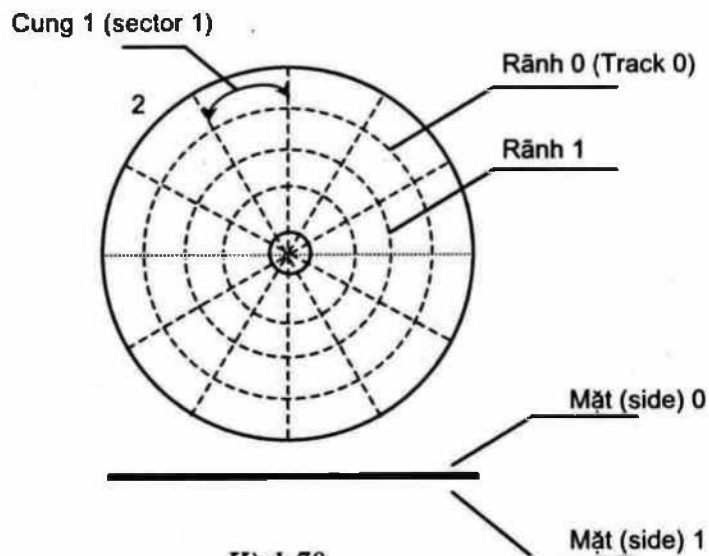
Để có thể sử dụng đĩa từ làm bộ nhớ ngoài cho máy tính cần phân chia không gian lưu trữ của đĩa từ ra thành nhiều đơn vị lưu trữ thông tin và gán địa

chỉ cho các đơn vị lưu trữ này. Quá trình phân chia và gán địa chỉ này được gọi là quá trình định dạng cấp thấp (Low Level Formatting) đĩa từ.

Mỗi ổ đĩa (mềm hoặc cứng) đều có một hay nhiều lá đĩa, mỗi lá đĩa có hai mặt (side) và được đánh số tuần tự 0, 1, 2, v.v. từ trên xuống dưới.

Mỗi mặt đĩa được chia thành các vòng tròn đồng tâm cách đều nhau, mỗi vòng tròn được gọi là một rãnh (track). Thông tin được lưu trữ trên các rãnh này. Với đĩa mềm mỗi mặt gồm 80 rãnh. Với đĩa cứng số rãnh quy chuẩn trên một mặt có thể lên tới 1024. Các rãnh được đánh số thứ tự từ ngoài vào trong. Rãnh ngoài cùng (đường kính lớn nhất) có số thứ tự là 0, rãnh tiếp theo là 1, v.v.

Mỗi rãnh được chia thành các cung (sector). Cung là đơn vị lưu trữ thông tin nhỏ nhất trên đĩa. Mỗi một cung tiêu chuẩn chứa được 512 byte dữ liệu. Số cung trên một rãnh tùy thuộc vào loại đĩa từ. Với đĩa mềm một rãnh chứa được 18 cung. Với đĩa cứng số cung quy chuẩn trên một rãnh là 63. Các cung trên một rãnh được đánh số thứ tự từ 1 đến tăng dần, theo hướng ngược chiều quay kim đồng hồ (h.78).



Hình 78

Các rãnh có cùng đường kính tạo thành một trụ (cylinder). Số thứ tự của rãnh cũng là số thứ tự của trụ.

Việc phân chia rãnh thành các cung với số lượng đều như nhau trên tất các rãnh từ ngoài vào trong (như hình vẽ) chỉ thích hợp cho các đĩa từ có mật độ

lưu trữ thấp như đĩa mềm. Kiểu ghi dữ liệu chuẩn này gây lãng phí không gian lưu trữ, đặc biệt là đối với đĩa cứng, vì các rãnh bên ngoài có chu vi lớn hơn và cũng có nghĩa là không gian lưu trữ lớn hơn các rãnh ở trong. Đối với đĩa cứng, để tăng dung lượng lưu trữ thông tin trên đĩa, người ta áp dụng kỹ thuật ghi theo vùng, mỗi vùng có nhiều rãnh. Các vùng kế tiếp nhau từ tâm đĩa ra ngoài sẽ có số cung/rãnh tăng dần lên từ trong ra ngoài. Số lượng cung trên mỗi rãnh trong cùng một vùng là giống nhau. Các ổ đĩa cứng thường có 10 vùng rãnh trở lên.

Cấu trúc điển hình của một rãnh và của một cung trên đĩa cứng như sau :

| Tên trường | Kích thước (byte) | Nội dung |
|-----------------------|-------------------|----------|
| Pre Index Gap | 56 | 00h |
| ⋮ | ⋮ | ⋮ |
| Index | 7 | 4Eh |
| Sync | 10 | 00h |
| Address Mark | 2 | A1FEh |
| Cylinder number | 2 | |
| Head number | 1 | |
| Sector number | 1 | |
| Error Correction Code | 2 | |
| Write Turn-On Gap | 5 | 00h |
| Sync | 10 | 00h |
| Data Mark | 2 | A1F8h |
| DATA | 512 | |
| Error Correction Code | 2 | |
| Write Turn-Off Gap | 15 | 00h |
| ⋮ | ⋮ | ⋮ |

- Pre Index Gap :** cung cấp khoảng thời gian tìm đầu rãnh.
- Index :** báo bắt đầu một cung.
- Sync :** đồng bộ hoá hoạt động của bộ điều khiển với tốc độ quay của đĩa.
- Address Mark :** báo bắt đầu của vùng địa chỉ.
- Cylinder number :** số thứ tự của trụ (rãnh).
- Head number :** số thứ tự của đầu từ (mặt).
- Sector number :** số thứ tự của cung vật lý.
- Error Correction Code :** mã sửa sai cho vùng địa chỉ và vùng dữ liệu.
- Write Turn-On Gap :** cung cấp vùng đệm cách ly vùng địa chỉ và vùng dữ liệu, đồng thời tạo thời gian kiểm tra lỗi vùng địa chỉ.
- Data Mark :** báo bắt đầu của vùng dữ liệu.
- DATA :** vùng chứa dữ liệu 512 byte.
- Write Turn-Off Gap :** cung cấp vùng đệm cách ly giữa các cung, đồng thời tạo thời gian kiểm tra lỗi vùng dữ liệu.

Cấu trúc một rãnh của đĩa mềm cũng tương tự như ở đĩa cứng, tuy nhiên kích thước của các trường thông tin có khác và mã ECC được thay bằng mã CRC (Cyclic Redundancy Check).

Thao tác định dạng cấp thấp ghi các byte thông tin định dạng cung và rãnh lên từng rãnh của đĩa từ. Mỗi cung có một địa chỉ xác định vị trí mặt, vị trí rãnh và vị trí của cung trên rãnh đó. Bộ ba con số này là duy nhất đối với mỗi một cung trên đĩa từ, và là địa chỉ vật lý của cung. Vùng DATA được điền bởi các byte dữ liệu giả khi thao tác định dạng cấp thấp được thực hiện.

Số lượng cung trên một rãnh đối với đĩa mềm phụ thuộc vào kích thước đĩa. Đối với đĩa cứng số lượng cung trên một rãnh phụ thuộc vào ổ đĩa và thiết bị điều khiển giao diện trên ổ. Điều đáng lưu ý là để tương thích với BIOS trong máy vi tính, các ổ đĩa cứng phiên dịch thông tin về mật độ thật của rãnh (track) trên mặt đĩa và mật độ thật của số cung (sector) trên một rãnh thành dạng tiêu chuẩn 63 cung/rãnh và cung cấp cho BIOS.

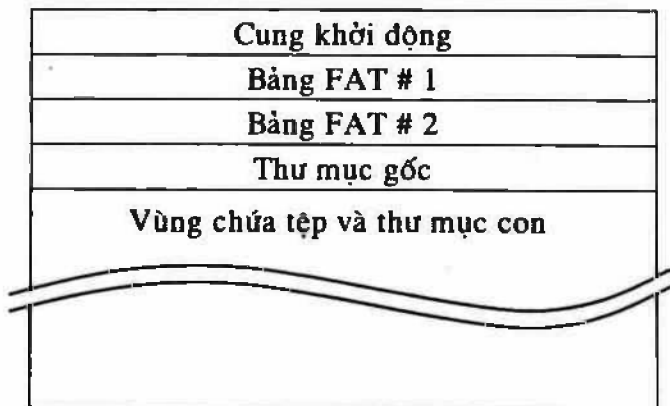
5. TỔ CHỨC LƯU TRỮ THÔNG TIN TRÊN ĐĨA TỪ Ở MỨC LOGIC

Địa chỉ vật lý (mặt, rãnh, cung) của một cung là kiểu địa chỉ không tuyến tính và không thích hợp cho việc quản lý và sử dụng cung với tư cách là một đơn vị lưu trữ thông tin của bộ nhớ máy tính. Việc dùng đĩa từ như là một hệ thống lưu trữ tập tin (tệp) của máy tính đòi hỏi phải tổ chức lại cách quản lý các cung nói riêng và quản lý việc lưu trữ thông tin trên đĩa từ nói chung.

Các hệ điều hành khác nhau có các cách tổ chức lưu trữ và quản lý tệp trên đĩa từ khác nhau. Hệ điều hành DOS và Windows 9X sử dụng cách tổ chức lưu trữ và quản lý tệp (tổ chức hệ thống tệp) kiểu FAT (File Allocation Table).

Theo cách tổ chức này bộ nhớ ngoài được coi là tập hợp của các volume, một volume có thể là toàn bộ một đơn vị vật lý (như đĩa mềm) hoặc là một phần của ổ cứng. Mỗi volume được xem là một tập hợp liên tục các cung logic. Các cung logic được đánh số thứ tự, bắt đầu là cung logic 0, tiếp đó là cung logic 1 v.v. lần lượt tăng dần cho đến hết volume. Việc tổ chức lại các thiết bị đĩa từ theo volume làm cho việc quản lý các thiết bị này trở nên thống nhất, không phụ thuộc vào cấu trúc vật lý cụ thể của các ổ đĩa. Quá trình thao tác với tệp lúc này phải thực hiện qua hai bước : trước hết yêu cầu ghi/đọc tệp từ ứng dụng được chuyển thành ghi/đọc các cung logic (có địa chỉ tuyến tính), sau đó các địa chỉ tuyến tính của các cung được chuyển thành địa chỉ vật lý (mặt, rãnh, cung). Quá trình ghi/đọc tệp thật sự được thực hiện theo địa chỉ vật lý này.

Mỗi volume được chia ra thành các vùng, mỗi vùng chứa một loại thông tin về cách tổ chức lưu trữ trên volume. Cấu trúc của một volume như sau (h.79) :



Hình 79

5.1 Cung khởi động volume (Volume Boot Sector)

Cung khởi động volume (gọi tắt là cung khởi động) nằm ở cung logic 0 của một volume. Đối với đĩa mềm cung logic 0 có địa chỉ vật lý là : mặt 0, rãnh 0, cung 1. Đối với đĩa cứng cung logic 0 của volume đầu tiên có địa chỉ vật lý là : mặt 1, rãnh 0, cung 1.

Cung khởi động chứa thông tin về cách phân vùng trên volume và chứa chương trình khởi động hệ điều hành :

| Địa chỉ offset (hexadecimal) | Nội dung | Kích thước (byte) |
|------------------------------|---|-------------------|
| 00 | Lệnh nhảy đến chương trình khởi động | 3 |
| 03 | Tên nhà sản xuất và thế hệ của HĐH | 8 |
| 0B | Số lượng byte / một cung | 2 |
| 0D | Số lượng cung / một cluster | 1 |
| 0E | Số lượng cung trước bảng FAT | 2 |
| 10 | Số lượng bảng FAT | 1 |
| 11 | Số lượng lối vào thư mục trong thư mục gốc | 2 |
| 13 | Số lượng cung trong volume | 2 |
| 15 | Byte mô tả loại đĩa | 1 |
| 16 | Số lượng cung / một bảng FAT | 2 |
| 18 | Số lượng cung / một rãnh | 2 |
| 1A | Số lượng đầu từ | 2 |
| 1C | Số lượng cung ẩn | 2 |
| 1E | Số lượng cung trong volume, nếu kích thước volume lớn hơn 32 Mb | 4 |
| 22 | Số thứ tự ổ đĩa | 1 |
| 23 – 33 | Dự phòng cho các thông tin phụ | 17 |
| 34 | Nhận dạng hệ FAT | 8 |
| 3C – 1FF | Chương trình khởi động HĐH | 472 |

Cung khởi động được kết thúc bằng 2 byte chữ ký AA55h. Khi máy tính, cung khởi động được nạp vào địa chỉ 0000h : 7C00h. Phải kiểm tra chữ ký, nếu không tìm thấy hai byte chữ ký thì sẽ

thấy chữ ký BIOS sẽ cho thực hiện chương trình từ địa chỉ 0000h :7C00h. Đó cũng là lý do vì sao byte đầu tiên của cung khởi động luôn chứa mã lệnh nhảy không điều kiện.

Cung khởi động chứa các dữ liệu mà từ đó có thể tính ra được vị trí bắt đầu và kích thước (theo đơn vị cung) của các vùng thông tin trên volume.

5.2 Bảng FAT (File Allocation Table)

Hệ điều hành lưu trữ tệp trong vùng chứa tệp. Vùng chứa tệp nằm sau thư mục gốc và chiếm toàn bộ không gian còn lại của volume. Vùng chứa tệp được coi là tập hợp của các cluster. Cluster là tập hợp của một hoặc nhiều cung liên nhau. Cluster là đơn vị lưu trữ thông tin nhỏ nhất mà hệ điều hành dùng để lưu trữ tệp. Các cluster được đánh số thứ tự bắt đầu từ 2 và tăng dần đến hết volume.

Bảng FAT nằm ngay sau cung khởi động. Mỗi volume thường có hai bảng FAT #1 và #2, nội dung hai bảng này giống hệt nhau.

Bảng FAT được dùng để quản lý các cluster. Bảng FAT chứa các lối vào (entry). Kích thước của một lối vào có thể là 12, 16 hoặc 32 bit, tùy thuộc vào kích thước của volume và cách tổ chức hệ tệp. Hai lối vào đầu tiên trong bảng FAT được dùng để chỉ ra dạng tổ chức của đĩa. Số lượng lối vào còn lại đúng bằng số lượng các cluster. Các lối vào này cũng được đánh số thứ tự bắt đầu từ 2 và tăng dần cho đến tận lối vào cuối cùng trong bảng FAT. Mỗi một lối vào, bắt đầu từ lối vào số 2, chứa một thông tin về trạng thái của cluster có số thứ tự tương ứng.

Thông tin về trạng thái của cluster được thể hiện dưới dạng mã nhị phân như sau :

| Nội dung của lối vào (hexadecimal) | Trạng thái cluster tương ứng |
|------------------------------------|---|
| (0)000 | Cluster rỗng |
| (F)FF0 – (F)FF6 | Cluster dự phòng |
| (F)FF7 | Cluster hỏng |
| (F)FF8 – (F)FFF | Cluster cuối cùng của một tệp nào đó |
| (X)XXX | Cluster tương ứng với lối vào này đang chứa tệp. Con số (X)XXX xác định địa chỉ cluster tiếp theo của tệp. |

Bảng FAT đóng vai trò một bản đồ về trạng thái các cluster.

Từ bảng FAT có thể tìm ra được một chuỗi danh sách các cluster thuộc một tệp nào đó, nhưng còn thiếu thông tin về cluster đầu tiên của tệp. Thông tin này gắn với một tệp cụ thể và nằm ở thư mục.

5.3 Thư mục gốc (Root Directory)

Mỗi volume có một thư mục gốc. Thư mục gốc nằm sau bảng FAT #2. Thư mục gốc chứa các lối vào thư mục (Directory Entry), mỗi lối vào thư mục có kích thước 32 byte. Thư mục gốc có kích thước hạn chế xác định trước.

Lối vào thư mục chứa các thông tin (tên tệp, thuộc tính, kích thước v.v.) của một tệp cụ thể đang được lưu trữ trên volume và số thứ tự cluster đầu tiên của tệp đó.

Lối vào thư mục có cấu trúc như sau :

| Địa chỉ offset (hexadecimal) | Nội dung | Kích thước (byte) |
|---------------------------------|-------------------------------------|----------------------|
| 00 | Tên tệp hoặc tên thư mục con | 8 |
| 08 | Phần tên mở rộng | 3 |
| 0B | Thuộc tính tệp | 1 |
| 0C – 15 | Dự phòng. | 10 |
| 16 | Thời gian tạo hoặc cập nhật tệp | 2 |
| 18 | Ngày tạo hoặc cập nhật tệp | 2 |
| 1A | Số thứ tự cluster đầu tiên của tệp. | 2 |
| 1C | Kích thước tệp | 4 |

– Byte đầu tiên của phần tên tệp có thể chứa các thông tin đặc biệt sau :

00h : báo thư mục kết thúc ở đây.

e5h : tệp bị xoá

2eh : lối vào thư mục của thư mục hiện thời. Lối vào này chứa thông tin về vị trí và kích thước của chính thư mục đang chứa lối vào này.

2eh 2eh : lối vào thư mục mẹ của thư mục hiện thời. Lối vào này chú thông tin về vị trí và kích thước của thư mục mẹ của thư mục đang chứa lối vào này. Nhờ thông tin này mà từ thư mục hiện thời ta có thể chuyển về thư mục mẹ.

Trường thời gian được mã hoá như sau :

| Vị trí bit | Nội dung |
|------------|--|
| 0 – 4 | Số nhị phân của giây chẵn : 0 – 29 tương ứng giây thứ 0 – 58 |
| 5 – 10 | Số nhị phân của phút : 0– 59 |
| 11 – 15 | Số nhị phân của giờ : 0– 23 |

Trường ngày được mã hoá như sau :

| Vị trí bit | Nội dung |
|------------|-------------------------------|
| 0 – 4 | Số nhị phân của ngày : 1 – 31 |
| 5 – 8 | Số nhị phân của tháng : 1– 12 |
| 9 – 15 | Số nhị phân của năm |

– Trường thuộc tính tệp, khi bit được đặt bằng 1 thì mang thông tin như sau :

| Vị trí bit | Nội dung |
|------------|--|
| 0 | Chỉ đọc |
| 1 | Ẩn |
| 2 | Hệ thống |
| 3 | Nhãn Volume |
| 4 | Thư mục con |
| 5 | Lưu trữ. Bit này bằng 1 khi tệp bị thay đổi nội dung |
| 6 | Dự phòng |
| 7 | Dự phòng |

Cấu trúc của thư mục con giống như thư mục gốc. Thư mục con nằm ở vùng chứa tệp và thư mục con. Thư mục con được quản lý như một tệp thông thường và có kích thước không hạn chế.

5.4 Vùng chứa tệp và thư mục con

Vùng chứa tệp và thư mục con là toàn bộ vùng còn lại nằm sau thư mục gốc. Vùng này được coi là tập hợp của các cluster. Các cluster được đánh số thứ tự bắt đầu từ 2 tăng dần đến cluster cuối cùng trong volume. Trạng thái của các cluster được phản ánh bởi lối vào tương ứng trong bảng FAT. Các thư mục con nằm ở vùng này. Thư mục con được quản lý như một tệp thông thường và có kích thước không hạn chế.

Một tệp được lưu trữ trên một hoặc nhiều cluster khác nhau, tùy thuộc kích thước tệp. Các cluster chứa dữ liệu của một tệp không nhất thiết phải liên kế nhau.

Nếu volume chứa hệ thống khởi động hệ điều hành thì các cluster đầu tiên của vùng này được dành để chứa các tệp hệ thống của hệ điều hành.

5.5 Mối quan hệ giữa thư mục, bảng FAT và cluster

Mối quan hệ giữa bảng FAT và cluster đã được trình bày ở mục về bảng FAT.

Mối quan hệ giữa thư mục, bảng FAT và cluster trong việc quản lý một tệp được minh họa qua ví dụ cụ thể sau (h.80) :

Giả sử có tệp tên MYFILE.DAT, tên tệp nằm tại thư mục gốc. Trong thư mục gốc ta có thể tìm thấy và đọc được một lối vào thư mục có nội dung sau :

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 4Dh | 59h | 46h | 49h | 4Ch | 45h | 20h | 20h | 44h | 41h | 54h | 20h | 00h | 00h | 00h | 00h |
| 00h | 00h | 00h | 00h | 00h | 00h | 98h | 9Ah | 9bh | 0Ah | 03h | 00h | 42h | 05h | 00h | 00h |

Các byte 00h – 0Ah : mã ASCII của tên tệp
MYFILE.DAT

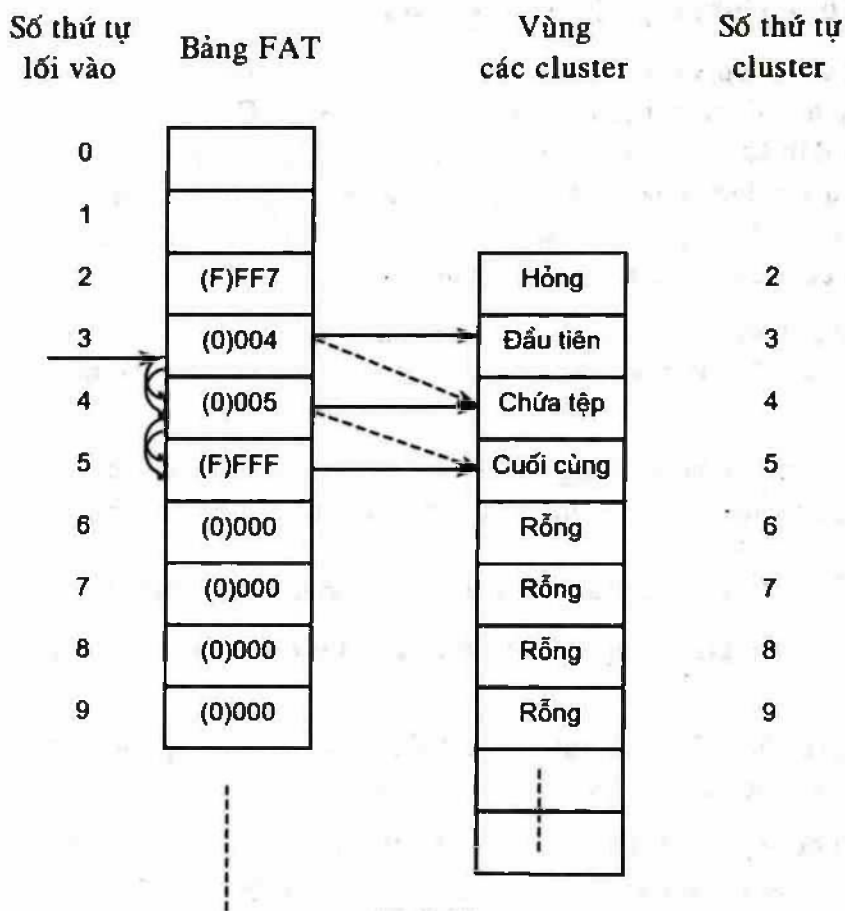
Byte 0Bh : 20h – tệp lưu trữ

Số thứ tự cluster đầu tiên của tệp

Kích thước tệp

Byte 1Ah–1Bh : (0)003h – Số thứ tự của cluster đầu tiên của tệp. Lối vào đầu tiên của tệp trên bảng FAT cũng là (0)003h.

Byte 1Ch – 1Fh : kích thước tệp 0542h \equiv 1346 byte.



Hình 80

Quá trình truy cập tệp ở mức thấp bắt đầu từ việc tìm kiếm thư mục chứa tên tệp. Trong thư mục sẽ tìm được lối vào thư mục có chứa tên tệp cần truy cập, ví dụ lối vào thư mục có chứa tên tệp MYFILE.DAT. Từ lối vào thư mục này tìm được địa chỉ (số thứ tự) cluster đầu tiên của tệp MYFILE.DAT, ví dụ (0)003h. Con số (0)003h cũng là số thứ tự của lối vào đầu tiên của tệp này trong bảng FAT. Thao tác tiếp theo là tìm đến bảng FAT. Lối vào đầu tiên cần đọc là (0)003h. Lối vào (0)003h chứa con số (0)004h, con số này phản ánh hai thông tin : một là cluster số (0)003h đang chứa một phần của tệp, hai là cluster

tiếp theo của tệp là (0)004h. Từ đây tiếp tục truy nhập và đọc lối vào số (0)004h v.v. cho đến khi gặp lối vào chứa giá trị (F)FFFh thì dừng tìm kiếm. Ví dụ ở đây ta có danh sách các cluster chứa tệp là (0)003h, (0)004h, (0)005h.

Tổng quát mà nói, từ lối vào tại vị trí I ta đọc được nội dung J, nếu J thuộc khoảng giá trị (0)002h – (F)FEFh thì nó phản ánh hai thông tin : cluster thứ I đang chứa tệp và cluster thứ J là cluster tiếp theo của tệp. Quá trình tìm kiếm danh sách các cluster chứa tệp dừng khi gặp lối vào chứa giá trị (F)FFFh. Cluster tương ứng lối vào này là cluster cuối cùng của tệp. Trên thực tế không phải bao giờ cũng nhận được chuỗi danh sách các cluster liên kế nhau.

Để truy nhập tệp trên đĩa cần phải chuyển danh sách các cluster chứa tệp thành danh sách các cung logic. Công thức chuyển đổi địa chỉ (số thứ tự) cluster thành địa chỉ (số thứ tự) cung logic như sau :

$$\text{Địa chỉ cung logic} = (\text{Địa chỉ cluster} - 2) * (\text{Số lượng cung trên một cluster}) + \text{Địa chỉ cung logic đầu tiên của vùng dữ liệu.}$$

Quá trình truy nhập cung khởi động, bảng FAT và tệp có thể thực hiện nhờ các ngắt 25h và 26h. Khi truy nhập tệp và đọc vào bộ nhớ cần lưu ý dữ liệu của tệp chỉ chiếm một phần của cluster cuối cùng chứ không phải toàn bộ cluster này. Thông tin về kích thước tệp giúp xác định chính xác lượng dữ liệu thật của tệp trên cluster cuối cùng.

6. BẢNG PHÂN KHU (PARTITION TABLE)

Một ổ cứng vật lý có thể được phân chia thành nhiều phân khu (partition) và từ mỗi phân khu tạo ra được một volume tách biệt. Việc tạo ra các phân khu cho phép ta có thể cài đặt nhiều hệ điều hành và hệ thống tệp riêng biệt trong cùng một ổ cứng vật lý. Các hệ thống tệp khác nhau (ví dụ như hệ tệp FAT, FAT32, NTFS v.v.) sẽ sử dụng các phương pháp riêng để phân chia và quản lý không gian lưu trữ tệp.

Mỗi ổ cứng phải có ít nhất 1 phân khu và có thể có tối đa 4 phân khu chính. Với hệ thống tệp FAT quá trình tạo phân khu có thể thực hiện bằng lệnh FDISK của hệ điều hành.

Thông tin về cách phân chia ổ đĩa cứng thành các phân khu được chứa trong Bảng phân khu. Hệ điều hành sử dụng một vùng riêng và cố định trên ổ đĩa cứng để chứa Bảng phân khu, đó là cung vật lý 1 trên rãnh 0 của mặt 0. Cung vật lý đầu tiên của ổ đĩa cứng (mặt 0, rãnh 0, cung 1) không chứa cung khởi động volume (Volume Boot Record) mà chứa cung khởi động chính (Master Boot Record- MBR).

- Cung khởi động chính có cấu trúc như sau :

| Địa chỉ offset (hexadecimal) | Nội dung |
|---------------------------------|---------------------------------|
| 000 | Chương trình đọc cung khởi động |
| 1BE | Bảng phân khu chính |
| 1FE | Chữ ký AA55h |

Cung khởi động chính chứa chương trình đọc Bảng phân khu chính và cung khởi động. Chức năng của chương trình đọc cung khởi động là chuyển điều khiển hệ thống đến chương trình khởi động hệ điều hành, chương trình này nằm trong cung khởi động của phân khu khởi động.

Bảng phân khu chính chứa các lối vào phân khu (Partition Entry). Bảng phân khu chính có ít nhất 1 lối vào phân khu và nhiều nhất 4 lối vào phân khu. Lối vào phân khu chứa thông tin về vị trí và kích thước của phân khu trong ổ đĩa cứng.

- Bảng phân khu chính có cấu trúc như sau :

| Địa chỉ offset (hexadecimal) | Nội dung | Kích thước (byte) |
|---------------------------------|--------------------|----------------------|
| 1BE | Lối vào phân khu 1 | 16 |
| 1CE | Lối vào phân khu 2 | 16 |
| 1DE | Lối vào phân khu 3 | 16 |
| 1EE | Lối vào phân khu 4 | 16 |

Mỗi một lối vào phân khu mang thông tin về phân khu tương ứng : vị trí (mặt, rãnh, cung) bắt đầu phân khu, vị trí (mặt, rãnh, cung) kết thúc phân khu, kích thước phân khu (tính theo đơn vị cung) v.v.

- Cấu trúc của một lối vào phân khu :

| Địa chỉ offset (hexadecimal) | Nội dung | Kích thước (byte) |
|---------------------------------|------------------------------|----------------------|
| 00 | Chỉ thị khởi động | 1 |
| 01 | Địa chỉ đầu phân khu | 3 |
| 04 | Chỉ thị hệ thống | 1 |
| 05 | Địa chỉ cuối phân khu | 3 |
| 08 | Số lượng cung trước phân khu | 4 |
| 0c | Số lượng cung trong phân khu | 4 |

- Chỉ thị khởi động :

80h Phân khu khởi động (Khởi động hệ điều hành từ phân khu này).

00h Phân khu không tích cực.

- Chỉ thị hệ thống : cho biết loại hệ thống tệp được sử dụng trong phân khu.

01h FAT 12

06h, 0Eh FAT 16

07h NTFS

0Bh FAT 32

- Khuôn dạng của trường địa chỉ đầu/địa chỉ cuối phân khu :

| Mặt | | | | | | | | Cung | | | | | Rãnh | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|------|---|---|---|---|------|---|---|---|---|---|---|---|---|---|---|
| H | H | H | H | H | H | H | H | C | C | S | S | S | S | S | S | C | C | C | C | C | C | C | C |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Địa chỉ mặt là một số nhị phân 8 bit : H7 – H0. Số mặt tối đa có thể là 256

Địa chỉ cung là một số nhị phân 6 bit : S5 – S0. Số cung tối đa có thể là 63

Địa chỉ rãnh là một số nhị phân 10 bit : C9 – C0. Số rãnh tối đa có thể là 1024

Nội dung một lối vào phân khu có thể như sau :

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 00h | 01h | 02h | 03h | 04h | 05h | 06h | 07h | 08h | 09h | 0Ah | 0Bh | 0Ch | 0Dh | 0Eh | 0Fh |
| 80h | 01h | 01h | 00h | 06h | 04h | E2h | C4h | 00h | 00h | 00h | 22h | 90h | 32h | 02h | 00h |

Phân khu này là phân khu khởi động.

Địa chỉ đầu phân khu : Mặt 1, rãnh 0, cung vật lý 1.

Địa chỉ cuối phân khu : Mặt 4, rãnh 964, cung vật lý 34

Kích thước phân khu : 023290h = 164016 cung

Tại mỗi phân khu có thể thực hiện định dạng (mức cao) để tạo volume và cài đặt hệ điều hành trên các volume đó.

*

* *

TÀI LIỆU THAM KHẢO

1. Thomas C. Barteo, *Computer Architecture and Logic Design*. McGraw-Hill, Inc. 1991.
2. Ivan Tomek, *The Foundations of Computer Architecture and Organization*. Computer Science Press. 1991.
3. Daniel Tabak, *Advanced Microprocessors*. McGraw-Hill, Inc. 1995.
4. William Stalling, *Computer Organization and Architecture*, Prentice-Hall, Inc. 1996
5. Jeffrey P. Royer, *Handbook of Software and Hardware Interfacing for IBM PCs*. Prentice-Hall, Inc. 1987.
6. IBM, *IBM-PC Technical Reference*. 1984.
7. IBM, *AT Personal Computer Hardware Reference Library*. 1986.
8. Scott Mueller, *Upgrading and Repairing PCs*. (Bản dịch tiếng Việt : Cẩm nang sửa chữa và nâng cấp máy tính cá nhân – Nhà Xuất bản Đà Nẵng, 2002).
9. Larry Jordan, *Communication and Networking for IBM-PC and Compatibles*. Brandy Publishing. 1992.
10. Henri Nussbaumer, *Computer Communication Systems*. John Wiley & Son. 1990.
11. Intel, *Component Data Catalog*. 1991.
12. Intel, *Intel Architecture : Basic Architecture*. 1997.
13. Intel, *Pentium Processor Family : Developer's Manuel*. 1997.
14. Nguyễn Văn Tam, *Các bộ vi xử lý thông dụng 16/32 bit*. Nhà xuất bản Thống kê. 1990.
15. Nguyễn Nam Trung, *Cấu trúc máy vi tính & Thiết bị ngoại vi*. Nhà xuất bản Khoa học kỹ thuật. 2000.
16. *Kỹ thuật vi xử lý*. Nhà xuất bản Thống kê. 1983.

MỤC LỤC

| | Trang |
|--|-------|
| Lời nói đầu | 3 |
| PHẦN I | |
| CƠ SỞ KIẾN TRÚC MÁY TÍNH | |
| Chương I. KIẾN TRÚC CƠ BẢN CỦA MÁY TÍNH SỐ | |
| 1. Những thành phần cơ bản của máy tính số | 5 |
| 2. Hệ đếm nhị phân và phương pháp biểu diễn thông tin trong máy tính số | 7 |
| 3. Kiến trúc một máy tính đơn giản | 16 |
| Chương II. BUS VÀ VẤN ĐỀ TRUYỀN THÔNG TIN TRONG MÁY TÍNH | |
| 1. Hệ thống BUS | 23 |
| 2. Thiết bị ba trạng thái | 25 |
| 3. Chu kỳ BUS | 26 |
| Chương III. BỘ NHỚ | |
| 1. Một vài khái niệm | 28 |
| 2. Xây dựng bộ nhớ từ các chip nhớ | 30 |
| Chương IV. CÁC PHƯƠNG PHÁP VÀO-RA DỮ LIỆU | |
| 1. Cấu trúc phần cứng của các hệ thống vào-ra dữ liệu trong máy tính | 34 |
| 2. Các phương pháp vào-ra dữ liệu | 35 |
| CHƯƠNG V. KIẾN TRÚC CỦA ĐƠN VỊ XỬ LÝ TRUNG TÂM 8 BIT VÀ CƠ CHẾ GỌI CHƯƠNG TRÌNH CON | |
| 1. Kiến trúc đơn vị xử lý trung tâm Intel 8085 | 42 |
| 2. Ngăn xếp, con trỏ ngăn xếp và cơ chế gọi chương trình con | 45 |

PHẦN II

KIẾN TRÚC MÁY VI TÍNH

| | |
|---|------------|
| Chương I. KIẾN TRÚC MÁY VI TÍNH PC VÀ ĐƠN VỊ XỬ LÝ TRUNG TÂM | |
| 1. Kiến trúc cơ bản máy vi tính PC | 51 |
| 2. Đơn vị xử lý trung tâm..... | 58 |
| 3. Pentium | 86 |
| Chương II. TỔ CHỨC BỘ NHỚ TRONG MÁY VI TÍNH | |
| 1. Tổ chức bộ nhớ theo phân cấp..... | 91 |
| 2. Tổ chức cache | 92 |
| 3. Tổ chức bộ nhớ máy vi tính PC | 96 |
| Chương III. CÁC THIẾT BỊ ĐIỀU KHIỂN VÀ GIAO DIỆN VÀO-RA DỮ LIỆU | |
| 1. Ngắt và bộ điều khiển ngắt PIC 8259 | 100 |
| 2. Truy nhập trực tiếp bộ nhớ và bộ điều khiển DMAC 8237 | 113 |
| 3. Vào-ra nối tiếp và thiết bị giao diện vào-ra nối tiếp UART 8250/16450 | 123 |
| 4. Thiết bị giao diện song song | 134 |
| Chương IV. CÁC THIẾT BỊ NGOẠI VI | |
| 1. Bàn phím..... | 138 |
| 2. Màn hình | 142 |
| Chương V. THIẾT BỊ ĐĨA TỪ | |
| 1. Nguyên lý lưu trữ thông tin trên đĩa từ | 148 |
| 2. Thiết bị đĩa mềm | 149 |
| 3. Thiết bị đĩa cứng và giao diện IDE..... | 153 |
| 4. Tổ chức lưu trữ thông tin trên đĩa từ ở mức vật lý | 158 |
| 5. Tổ chức lưu trữ thông tin trên đĩa từ ở mức logic | 162 |
| 6. Bảng phân khu | 169 |
| Tài liệu tham khảo | 173 |