HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

HOÀNG XUÂN DẬU

BÀI GIẢNG
KIẾN TRÚC MÁY TÍNH

HÀ NỘI 2010

LỜI NÓI ĐẦU

Kiến trúc máy tính là một trong các lĩnh vực khoa học cơ sở của ngành Khoa học máy tính nói riêng và Công nghệ thông tin nói chung. Kiến trúc máy tính là khoa học về lựa chọn và ghép nối các thành phần phần cứng của máy tính nhằm đạt được các mục tiêu về hiệu năng cao, tính năng đa dạng và giá thành thấp.

Môn học Kiến trúc máy tính là môn học cơ sở chuyên ngành trong chương trình đào tạo công nghệ thông tin hệ đại học và cao đẳng. Mục tiêu của môn học là cung cấp cho sinh viên các kiến thức cơ sở của kiến trúc máy tính, bao gồm bao gồm kiến trúc máy tính tổng quát, kiến trúc bộ xử lý trung tâm và các thành phần của bộ xử lý trung tâm, kiến trúc tập lệnh máy tính, cơ chế ống lệnh; hệ thống phân cấp của bộ nhớ, bộ nhớ trong, bộ nhớ cache và các loại bộ nhớ ngoài; hệ thống bus và các thiết bị vào ra.

Kiến trúc máy tính là một lĩnh vực đã được phát triển trong một thời gian tương đối dài với lượng kiến thức đồ sộ, nhưng do khuôn khổ của tài liệu có tính chất là bài giảng môn học, tác giả cố gắng trình bày những vấn đề cơ sở nhất phục vụ mục tiêu môn học. Nội dung của tài liệu được biên soạn thành sáu chương:

Chương 1 là phần giới thiệu các khái niệm cơ sở của kiến trúc máy tính, như khái kiệm kiến trúc và tổ chức máy tính; cấu trúc và chức năng các thành phần của máy tính; các kiến trúc máy tính von-Neumann và kiến trúc Harvard. Khái niệm về các hệ đếm và cách tổ chức dữ liệu trên máy tính cũng được trình bày trong chương này.

Chương 2 giới thiệu về khối xử lý trung tâm, nguyên tắc hoạt động và các thành phần của nó. Khối xử lý trung tâm là thành phần quan trọng và phức tạp nhất trong máy tính, đóng vai trò là bộ não của máy tính. Thông qua việc thực hiện các lệnh của chương trình bởi khối xử lý trung tâm, máy tính có thể thực thi các yêu cầu của người sử dụng.

Chương 3 giới thiệu về tập lệnh của máy tính, bao gồm các khái niệm về lệnh, dạng lệnh, các thành phần của lệnh; các dạng địa chỉ và các chế độ địa chỉ. Chương cũng giới thiệu một số dạng lệnh thông dụng kèm ví dụ minh hoạ. Ngoài ra, cơ chế ống lệnh – xử lý xen kẽ các lệnh cũng được đề cập.

Chương 4 trình bày về bộ nhớ trong: khái quát về hệ thống bộ nhớ và cấu trúc phân cấp của hệ thống nhớ; giới thiệu các loại bộ nhớ ROM và RAM. Một phần rất quan trọng của chương là phần giới thiệu về bộ nhớ cache - một bộ nhớ đặc biệt có khả năng giúp tăng tốc hệ thống nhớ nói riêng và cả hệ thống máy tính nói chung.

Chương 5 giới thiệu về bộ nhớ ngoài, bao gồm các loại đĩa từ, đĩa quang, các hệ thống RAID, NAS và SAN. Bộ nhớ ngoài là dạng bộ nhớ thường có dung lượng lớn và dùng để lưu trữ thông tin ổn định, không phụ thuộc nguồn điện nuôi.

Chương 6 trình bày về hệ thống bus và các thiết bị ngoại vi. Phần trình bày về hệ thống bus đề cập đến các loại bus như ISA, EISA, PCI, AGP và PCI-Express. Phần giới thiệu các thiết bị vào ra đề cập đến nguyên lý hoạt động của một số thiết bị vào ra thông dụng, như bàn phím, chuột, màn hình và máy in.

Tài liệu được biên soạn dựa trên kinh nghiệm giảng dạy môn học Kiến trúc máy tính trong nhiều năm của tác giả tại Học viện Công nghệ Bưu chính – Viễn thông, kết hợp tiếp thu các đóng góp của đồng nghiệp và phản hồi từ sinh viên. Tài liệu có thể được sử dụng làm tài liệu học tập cho sinh viên hệ đại học và cao đẳng các ngành công nghệ thông tin và điện tử viễn thông. Trong quá trình biên soạn, mặc dù tác giả đã rất cố gắng song không thể tránh khỏi có những thiếu sót. Tác giả rất mong muốn nhận được ý kiến phản hồi và các góp ý cho các thiếu sót, cũng như ý kiến về việc cập nhật, hoàn thiện nội dung của tài liệu.

Hà nội, tháng 8 năm 2010

Tác giả

TS. Hoàng Xuân Dậu

Email: dauhx@ptit.edu.vn

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ TIẾNG ANH VÀ VIẾT TẮT

Thuật ngữ tiếng Anh	Từ viết tắt	Thuật ngữ tiếng Việt/Giải thích
Central Processing Unit	CPU	Bộ/Đơn vị xử lý trung tâm
Control Unit	CU	Bộ/Đơn vị điều khiển
Arithmethic and Logic Unit	ALU	Bộ/Đơn vị tính toán số học và logic
Program Counter	PC	Bộ đếm chương trình
System Bus		Buýt hệ thống
Memory		Bộ nhớ
Cache		Bộ nhớ đệm / bộ nhớ kết
Random Access Memory	RAM	Bộ nhớ truy cập ngẫu nhiên
Read Only Memory	ROM	Bộ nhớ chỉ đọc
Basic Input Output System	BIOS	Hệ thống vào ra cơ sở
Pipeline		Cơ chế ống lệnh hay cơ chế xử lý xen kẽ các lệnh
Hit		Đoán trúng – là sự kiện CPU truy tìm một mục tin và tìm thấy trong cache.
Miss		Đoán trượt – là sự kiện CPU truy tìm một mục tin và không tìm thấy trong cache.
Advanced Technology Attachments	ATA	Chuẩn ghép nối đĩa cứng ATA
Parallel Advanced Technology Attachments	PATA	Chuẩn ghép nối đĩa cứng PATA – hay ATA song song
Integrated Drive Electronics	IDE	Chuẩn ghép nối đĩa cứng IDE
Serial ATA	SATA	Chuẩn ghép nối đĩa cứng SATA – hay ATA nối tiếp
Small Computer System Interface	SCSI	Chuẩn ghép nối đĩa cứng SCSI
Redundant Array of Independent Disks	RAID	Công nghệ lưu trữ RAID – tạo thành từ một mảng liên kết các đĩa cứng vật lý
Network Attached Storage	NAS	Hệ thống lưu trữ gắn vào mạng
Storage Area Network	SAN	Mạng lưu trữ
Industrial Standard Architecture	ISA	Buýt theo chuẩn công nghiệp ISA
Extended ISA	EISA	Buýt theo chuẩn công nghiệp mở rộng EISA
Peripheral Component Interconnect	PCI	Bus PCI
Accelerated Graphic Port	AGP	Cổng tăng tốc đồ hoạ AGP
PCI Express	PCIe	Buýt PCIe
Cathode Ray Tube	CRT	Màn hình ống điện tử âm cực
Liquid Crystal Display	LCD	Mình hình tinh thể lỏng

CHƯƠNG 1 GIỚI THIỆU CHUNG

1.1 KHÁI NIỆM VỀ KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH

Kiến trúc máy tính (Computer Architecture) và *Tổ chức máy tính* (Computer Organization) là hai trong số các khái niệm cơ bản của ngành *Công nghệ máy tính* (Computer Engineering). Có thể nói kiến trúc máy tính là bức tranh toàn cảnh về hệ thống máy tính, còn tổ chức máy tính là bức tranh cụ thể về các thành phần phần cứng của hệ thống máy tính.

Kiến trúc máy tính là khoa học về việc lựa chọn và kết nối các thành phần phần cứng để tạo ra các máy tính đạt được các yêu cầu về chức năng (functionality), hiệu năng (performance) vàgiá thành (cost). Yêu cầu chức năng đòi hỏi máy tính phải có thêm nhiều tính năng phong phú và hữu ích; yêu cầu hiệu năng đòi hỏi máy tính phải đạt tốc độ xử lý cao hơn và yêu cầu giá thành đòi hỏi máy tính phải càng ngày càng rẻ hơn. Để đạt được cả ba yêu cầu về chức năng, hiệu năng và giá thành là rất khó khăn. Tuy nhiên, nhờ có sự phát triển rất mạnh mẽ của công nghệ vi xử lý, các máy tính ngày nay có tính năng phong phú, nhanh hơn và rẻ hơn so với máy tính các thế hệ trước.

Kiến trúc máy tính được cấu thành từ 3 thành phần con: (i) *Kiến trúc tập lệnh* (Instruction Set Architecture), (ii) *Vi kiến trúc* (Microarchitecture) và *Thiết kế hệ thống* (System Design).

- Kiến trúc tập lệnh là hình ảnh của một hệ thống máy tính ở mức ngôn ngữ máy. Kiến trúc tập lệnh bao gồm các thành phần: tập lệnh, các chế độ địa chỉ, các thanh ghi, khuôn dạng địa chỉ và dữ liệu.
- Vi kiến trúc là mô tả mức thấp về các thành phần của hệ thống máy tính, phối ghép và việc trao đổi thông tin giữa chúng. Vi kiến trúc giúp trả lời hai câu hỏi (1) Các thành phần phần cứng của máy tính kết nối với nhau như thế nào? và (2) Các thành phần phần cứng của máy tính tương tác với nhau như thế nào để thực thi tập lệnh?
- Thiết kế hệ thống: bao gồm tất cả các thành phần phần cứng của hệ thống máy tính, bao gồm: Hệ thống phối ghép (các bus và các chuyển mạch), Hệ thống bộ nhớ, Các cơ chế giảm tải cho CPU (như truy nhập trực tiếp bộ nhớ) và Các vấn đề khác (như đa xử lý vàxử lý song song).

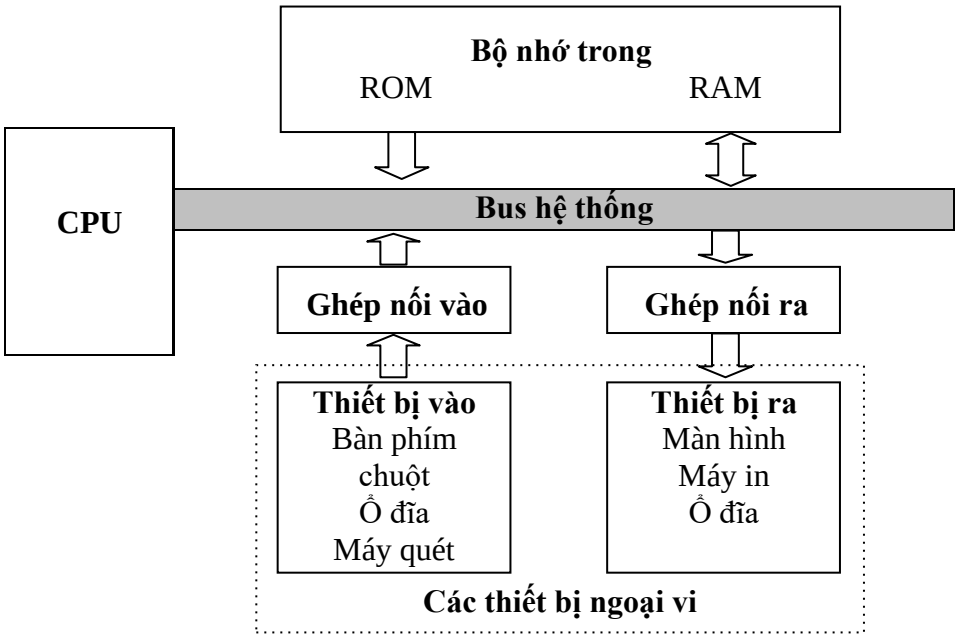
Tổ chức máy tính hay cấu trúc máy tính là khoa học nghiên cứu về các bộ phận của máy tính và phương thức làm việc của chúng. Với định nghĩa như vậy, tổ chức máy tính khá gần gũivới vi kiến trúc – một thành phần của kiến trúc máy tính. Như vậy, có thể thấy rằng, kiến trúc máy tính và khái niệm rộng hơn, nó bao hàm cả tổ chức hay cấu trúc máy tính.

1.2 CẤU TRÚC VÀ CHỨC NĂNG CÁC THÀNH PHẦN CỦA MÁY TÍNH

1.2.1 Sơ đồ khối chức năng

Hình 1 minh hoạ sơ đồ khối chức năng của một hệ thống máy tính. Theo đó, hệ thống máy tính gồm bốn thành phần chính: (1) CPU – Khối xử lý trung tâm, (2) Bộ nhớ trong, gồm bộ nhớ ROM và bộ nhớ RAM, (3) Các thiết bị ngoại vi, gồm các thiết bị vào và các thiết bị ra và (4) Bus hệ thống, là hệ thống kênh dẫn tín hiệu ghép nối các thành phần kể trên. Ngoài ra, còn

có các giao diện ghép nối vào và ghép nối ra dùng để ghép nối các thiết bị ngoại vi vào bus hệ thống. Mục 1.2.2 tiếp theo sẽ mô tả chi tiết chức năng của từng khối.



Hình 1. Sơ đồ khối chức năng của hệ thống máy tính

1.2.2 Các thành phần của máy tính

1.2.2.1 Khối xử lý trung tâm

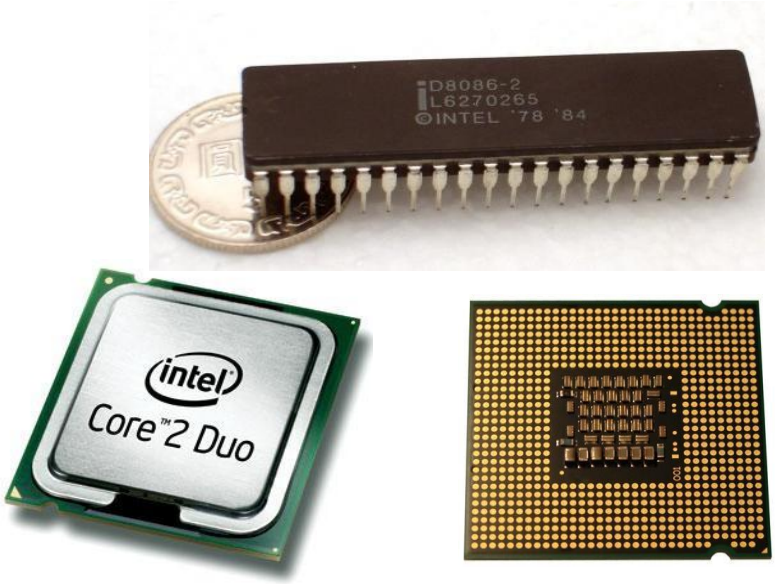
Khối xử lý trung tâm (Central Processing Unit - CPU) là thành phần quan trọng nhất - được xem là bộ não của máy tính. Các yêu cầu của hệ thống và của người sử dụng thường được biểu diễn thành các chương trình máy tính, trong đó mỗi chương trình thường được tạo thành từ nhiều lệnh của CPU. CPU đảm nhiệm việc đọc các lệnh của chương trình từ bộ nhớ, giải mã và thực hiện lệnh. Thông qua việc CPU thực hiện các lệnh của chương trình, máy tính có khả năng cung cấp các tính năng hữu ích cho người sử dụng.

CPU là vi mạch tích hợp với mật độ rất cao, được cấu thành từ bốn thành phần con: (1) Bộ điều khiển (Control Unit - CU), (2) Bộ tính toán số học và logic (Arithmetic and Logic Unit - ALU), (3) Các thanh ghi (Registers) và bus trong CPU (Internal Bus). Bộ điều khiển có nhiệm vụ đọc, giải mã và điều khiển quá trình thực hiện lệnh. Bộ tính toán số học và logic chuyên thực hiện các phép toán số học như cộng trừ, nhân, chia, và các phép toán lôgic như và, hoặc, phủ định và các phép dịch, quay. Các thanh ghi là kho chứa lệnh và dữ liệu tạm thời cho CPU xử lý. Bus trong CPU có nhiệm vụ truyền dẫn các tín hiệu giữa các bộ phận trong CPU và kết nối với hệ thống bus ngoài. Hình 2 minh hoạ hai CPU của hãng Intel là 8086 ra đời năm 1978 và Core 2 Duo ra đời năm 2006.

1.2.2.2 Bộ nhớ trong

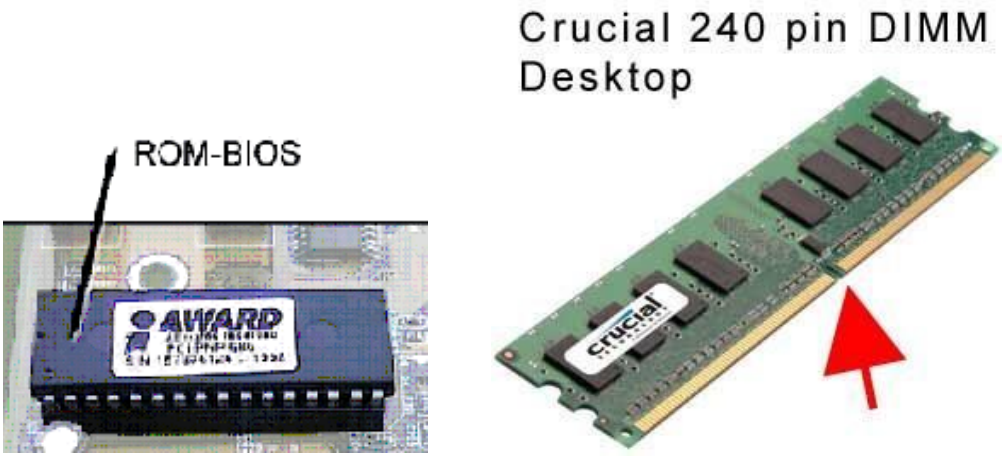
Bộ nhớ trong, còn gọi là bộ nhớ chính (Internal Memory hay Main Memory) là kho chứa lệnh và dữ liệu của hệ thống và của người dùng phục vụ CPU xử lý. Bộ nhớ trong thường là bộ nhớ bán dẫn, bao gồm hai loại: (1) Bộ nhớ chỉ đọc (Read Only Memory – ROM) và (2) Bộ nhớ truy cập ngẫu nhiên (Random Access Memory – RAM). ROM thường được sử dụng để lưu lệnh và dữ liệu của hệ thống. Thông tin trong ROM được nạp từ khi sản xuất và thường

chỉ có thể đọc ra trong quá trình sử dụng. Hơn nữa thông tin trong ROM luôn tồn tại kể cả khi không có nguồn điện nuôi.



Hình 2. CPU của hãng Intel: 8086 và Core 2 Duo

Khác với bộ nhớ ROM, bộ nhớ RAM thường được sử dụng để lưu lệnh và dữ liệu của cả hệ thống và của người dùng. RAM thường có dung lượng lớn hơn nhiều so với ROM. Tuy nhiên, thông tin trong RAM chỉ tồn tại khi có nguồn điện nuôi. Hình 3 minh hoạ vi mạch bộ nhớ ROM và các vi mạch nhớ RAM gắn trên một thanh nhớ RAM.



Hình 3 Bộ nhớ ROM và RAM

1.2.2.3 Các thiết bị vào ra

Các thiết bị vào ra (Input – Output devices), hay còn gọi là các thiết bị ngoại vi (Peripheral devices) đảm nhiệm việc nhập dữ liệu vào, điều khiển hệ thống và kết xuất dữ liệu ra. Có hai nhóm thiết bị ngoại vi: (1) Các thiết bị vào (Input devices) và (2) Các thiết bị ra (Output devices). Các thiết bị vào dùng để nhập dữ liệu vào và điều khiển hệ thống, gồm: bàn phím (keyboard), chuột (mouse), ổ đĩa (Disk Drives), máy quét ảnh (Scanners),... Các thiết bị ra dùng để xuất dữ liệu ra, gồm: màn hình (Screen), máy in (Printers), ổ đĩa (Disk Drives), máy vẽ (Plotters),...

1.2.2.4 Bus hệ thống

Bus hệ thống (System Bus) là một tập các đường dây kết nối CPU với các thành phần khác của máy tính. Bus hệ thống thường gồm ba bus con: Bus địa chỉ – Bus A (Address bus), Bus dữ liệu – Bus D (Data bus), Bus điều khiển - Bus C (Control bus). Bus địa chỉ có nhiệm vụ truyền tín hiệu địa chỉ từ CPU đến bộ nhớ và các thiết bị ngoại vi; Bus dữ liệu vận chuyển các tín hiệu dữ liệu theo hai chiều đi và đến CPU; Bus điều khiển truyền tín hiệu điều khiển từ CPU đến các thành phần khác, đồng thời truyền tín hiệu trạng thái của các thành phần khác đến CPU.

1.3 LỊCH SỬ PHÁT TRIỂN MÁY TÍNH

Lịch sử phát triển máy tính có thể được chia thành 5 thế hệ chính phụ thuộc vào sự phát triển của mạch điện tử.

1.3.1 Thế hệ 1 (1944-1959)

Máy tính thế hệ 1 sử dụng đèn điện tử làm linh kiện chính và băng từ làm thiết bị vào ra. Mật độ tích hợp linh kiện vào khoảng 1000 linh kiện / foot^3 (1 foot = 30.48 cm). Đại diện tiêu biểu của thế hệ máy tính này là siêu máy tính ENIAC (Electronic Numerical Integrator and Computer), trị giá 500.000 USD.

1.3.2 Thế hệ 2 (1960-1964)

Máy tính thế hệ 2 sử dụng bóng bán dẫn (transistor) làm linh kiện chính. Mật độ tích hợp linh kiện vào khoảng 100.000 linh kiện / foot^3. Các đại diện tiêu biểu của thế hệ máy tính này là UNIVAC 1107, UNIVAC III, IBM 7070, 7080, 7090, 1400 series, 1600 series. Máy tính UNIVAC đầu tiên ra đời vào năm 1951, có giá khởi điểm là 159.000 USD. Một số phiên bản kết tiếp của UNIVAC có giá nằm trong khoảng 1.250.000 – 1.500.000 USD.

1.3.3 Thế hệ 3 (1964-1975)

Máy tính thế hệ 3 sử dụng mạch tích hợp (IC – Integrated Circuit) làm linh kiện chính. Mật độ tích hợp linh kiện vào khoảng 10.000.000 linh kiện / foot^3. Các đại diện tiêu biểu của thế hệ máy tính này là UNIVAC 9000 series, IBM System/360, System 3, System 7.

1.3.4 Thế hệ 4 (1975-1989)

Máy tính thế hệ 4 sử dụng mạch tích hợp loại lớn (LSI – Large Scale Integrated Circuit) làm linh kiện chính. Mật độ tích hợp linh kiện vào khoảng 1 tỷ linh kiện / foot^3. Các đại diện tiêu biểu của thế hệ máy tính này là IBM System 3090, IBM RISC 6000, IBM RT, Cray 2 XMP.

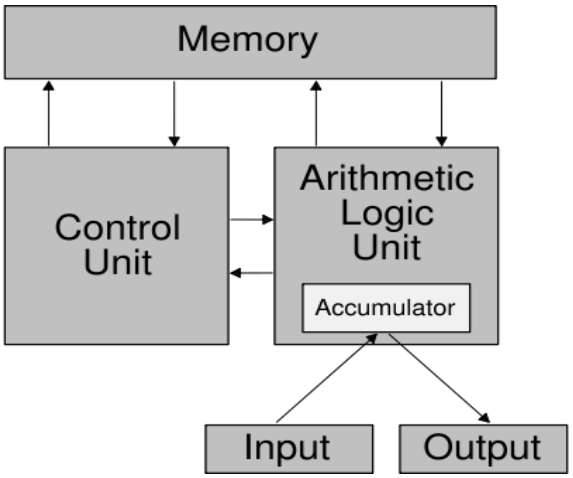
1.3.5 Thế hệ 5 (1990 - nay)

Máy tính thế hệ 5 sử dụng mạch tích hợp loại siêu lớn (VLSI – Very Large Scale Integrated Circuit) làm linh kiện chính. Mật độ tích hợp linh kiện rất cao với các công nghệ 0.180µm – 0.045µm (kích thước transistor giảm xuống còn 180 – 45 nano mét). Các đại diện tiêu biểu của thế hệ máy tính này là máy tính sử dụng CPU Intel Pentium II, III, IV, M, D, Core Duo, Core 2 Duo, Core Quad,... Máy tính thế hệ 5 đạt hiệu năng xử lý rất cao, cung cấp nhiều tính năng tiến tiến, như hỗ trợ xử lý song song, tích hợp khả năng xử lý âm thanh và hình ảnh.

1.4 KIẾN TRÚC MÁY TÍNH VON-NEUMANN

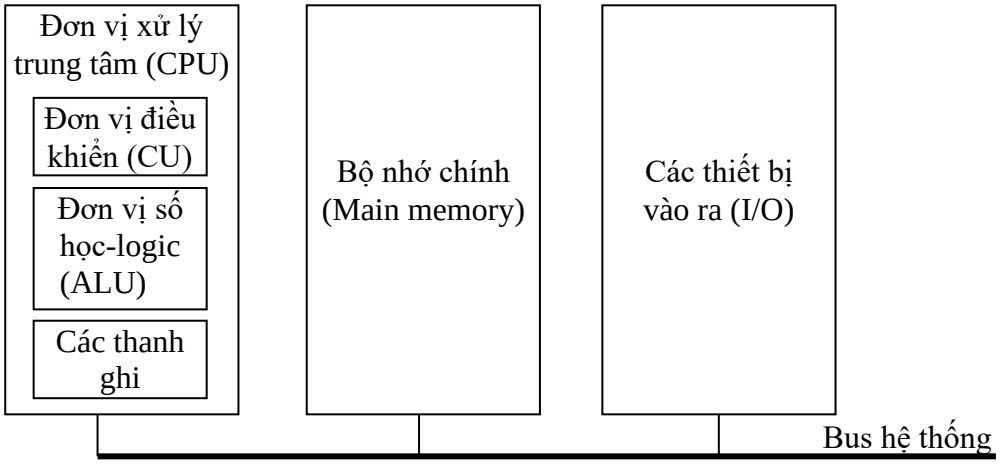
1.4.1 Sơ đồ kiến trúc máy tính von-Neumann

Kiến trúc máy tính von-Neumann được nhà toán học John von-Neumann đưa ra vào năm 1945 trong một báo cáo về máy tính EDVAC như minh hoạ trên Hình 4 - Kiến trúc máy tính von-Neumann nguyên thuỷ.



Hình 4 Kiến trúc máy tính von-Neumann nguyên thuỷ

Các máy tính hiện đại ngày nay sử dụng kiến trúc máy tính von-Neumann cải tiến – còn gọi là kiến trúc máy tính von-Neumann hiện đại, như minh hoạ trên Hình 5.



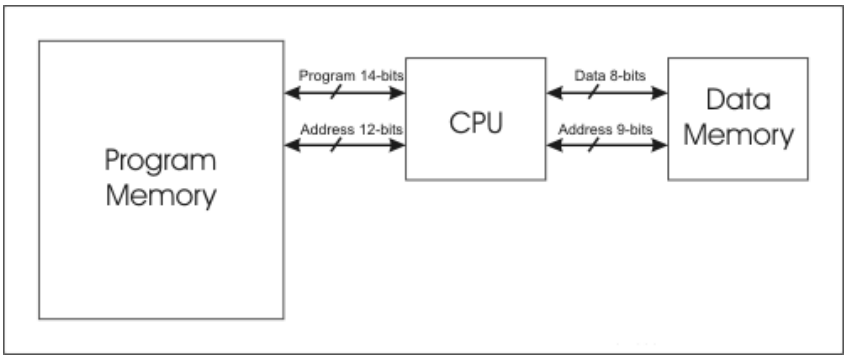
Hình 5 Kiến trúc máy tính von-Neumann hiện đại

1.4.2 Các đặc điểm của kiến trúc von-Neumann

Kiến trúc von-Neumann dựa trên 3 khái niệm cơ sở: (1) Lệnh và dữ liệu được lưu trữ trong bộ nhớ đọc ghi chia sẻ - một bộ nhớ duy nhất được sử dụng để lưu trữ cả lệnh và dữ liệu, (2) Bộ nhớ được đánh địa chỉ theo vùng, không phụ thuộc vào nội dung nó lưu trữ và (3) Các lệnh của một chương trình được thực hiện tuần tự. Quá trình thực hiện lệnh được chia thành 3 giai đoạn (stages) chính: (1) CPU đọc (fetch) lệnh từ bộ nhớ, (2) CPU giải mã và thực hiện lệnh; nếu lệnh yêu cầu dữ liệu, CPU đọc dữ liệu từ bộ nhớ; và (3) CPU ghi kết quả thực hiện lệnh vào bộ nhớ (nếu có).

1.5 KIẾN TRÚC MÁY TÍNH HARVARD

Kiến trúc máy tính Harvard là một kiến trúc tiên tiến như minh hoạ trên Hình 6.



Hình 6 Kiến trúc máy tính Harvard

Kiến trúc máy tính Harvard chia bộ nhớ trong thành hai phần riêng rẽ: Bộ nhớ lưu chương trình (Program Memory) và Bộ nhớ lưu dữ liệu (Data Memory). Hai hệ thống bus riêng được sử dụng để kết nối CPU với bộ nhớ lưu chương trình và bộ nhớ lưu dữ liệu. Mỗi hệ thống bus đều có đầy đủ ba thành phần để truyền dẫn các tín hiệu địa chỉ, dữ liệu và điều khiển.

Máy tính dựa trên kiến trúc Harvard có khả năng đạt được tốc độ xử lý cao hơn máy tính dựa trên kiến trúc von-Neumann do kiến trúc Harvard hỗ trợ hai hệ thống bus độc lập với băng thông lớn hơn. Ngoài ra, nhờ có hai hệ thống bus độc lập, hệ thống nhớ trong kiến trúc Harvard hỗ trợ nhiều lệnh truy nhập bộ nhớ tại một thời điểm, giúp giảm xung đột truy nhập bộ nhớ, đặc biệt khi CPU sử dụng kỹ thuật đường ống (pipeline).

1.6 CÁC HỆ SỐ ĐẾM VÀ TỔ CHỨC DỮ LIỆU TRÊN MÁY TÍNH

1.6.1 Các hệ số đếm

Trong đời sống hàng ngày, hệ đếm thập phân (Decimal Numbering System) là hệ đếm thông dụng nhất. Tuy nhiên, trong hầu hết các hệ thống tính toán hệ đếm nhị phân (Binary Numbering System) lại được sử dụng để biểu diễn dữ liệu. Trong hệ đếm nhị phân, chỉ 2 chữ số 0 và 1 được sử dụng: 0 biểu diễn giá trị Sai (False) và 1 biểu diễn giá trị Đúng (True). Ngoài ra, hệ đếm thập lục phân (Hexadecimal Numbering System) cũng được sử dụng. Hệ thập lục phân sử dụng 16 chữ số: 0-9, A, B, C, D, E, F.

1.6.1.1 Hệ đếm thập phân

Hệ đếm thập phân là hệ đếm cơ số 10, sử dụng 10 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Mỗi số trong hệ 10 có thể được biểu diễn thành một đa thức:

$$a_n a_{n-1} ... a_1 = a_n * 10^{n-1} a_{n-1} * 10^{n-2} * ... * a_1 * 10^0$$

Ví dụ:

$$123 = 1*10^2 + 2 * 10^1 + 3*10^0 = 100 + 20 + 3$$

$$123{,}456 = 1*10^2 + 2 * 10^1 + 3*10^0 + 4*10^{-1} + 5*10^{-2} + 6*10^{-3}$$

$$= 100 + 20 + 3 + 0.4 + 0.05 + 0.006$$

1.6.1.2 Hệ đếm nhị phân

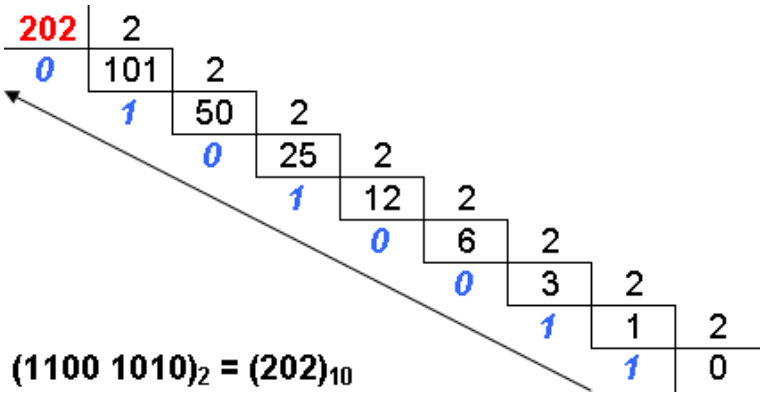
Hệ đếm nhị phân là hệ đếm cơ số 2, chỉ sử dụng 2 chữ số: 0 và 1. Mỗi số trong hệ 2 cũng có thể được biểu diễn thành 1 đa thức:

$$(a_n a_{n-1}...a_1)_2 = a_n*2^{n-1} a_{n-1}*2^{n-2}*...*a_1*2^0$$

Ví dụ:

$$(11001010)_2 = 1*2^7 + 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0$$
$$= 128 + 64 + 8 + 2 = (202)_{10}$$

Việc chuyển đổi số hệ thập phân sang số hệ nhị phân có thể được thực hiện theo thuật toán đơn giản như minh hoạ trên Hình 7.



Hình 7 Chuyển đổi số hệ thập phân sang số hệ nhị phân

1.6.1.3 Hệ đếm thập lục phân

Hệ đếm thập lục phân là hệ đếm cơ số 16, sử dụng 16 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Mỗi số trong hệ 16 được biểu diễn bởi 4 chữ số trong hệ nhị phân như minh hoạ trên Hình 8. Ưu điểm của hệ thập lục phân là số thập lục phân có thể chuyển đổi sang số hệ nhị phân và ngược lại một cách dễ dàng và cần ít chữ số hơn hệ nhị phân để biểu diễn cùng một đơn vị dữ liệu.

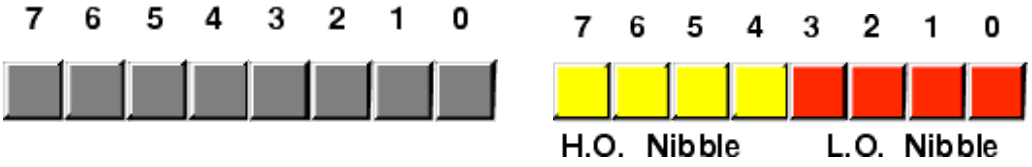
Hexa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Hình 8 Giá trị các số thập lục phân theo hệ thập phân và nhị phân

1.6.2 Tổ chức dữ liệu trên máy tính

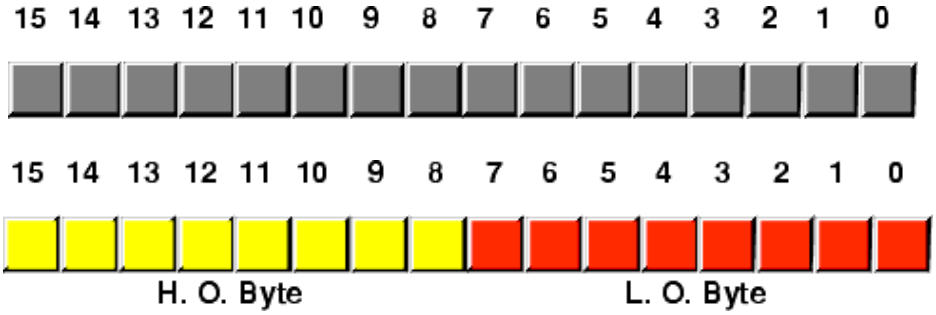
Dữ liệu trên máy tính được biểu diễn theo các đơn vị (unit). Các đơn vị biểu diễn dữ liệu cơ sở gồm: bit, nibble, byte, word và double-word. Bit là đơn vị dữ liệu nhỏ nhất: mỗi bit chỉ lưu được tối đa 2 giá trị: 0 hoặc 1, hay đúng hoặc sai. Nibble là đơn vị kế tiếp bit. Mỗi nibble là một nhóm 4 bit. Một nibble có thể lưu tối đa 16 giá trị, từ $(0000)_2$ đến $(1111)_2$, hoặc một chữ số thập lục phân.

Byte là đơn vị dữ liệu kế tiếp nibble. Một byte là một nhóm của 8 bits hoặc 2 nibbles. Một byte có thể lưu đến 256 giá trị, từ $(0000\ 0000)_2$ đến $(1111\ 1111)_2$, hoặc từ $(00)_{16}$ đến $(FF)_{16}$. Hình 9 minh hoạ đơn vị biểu diễn dữ liệu Byte.



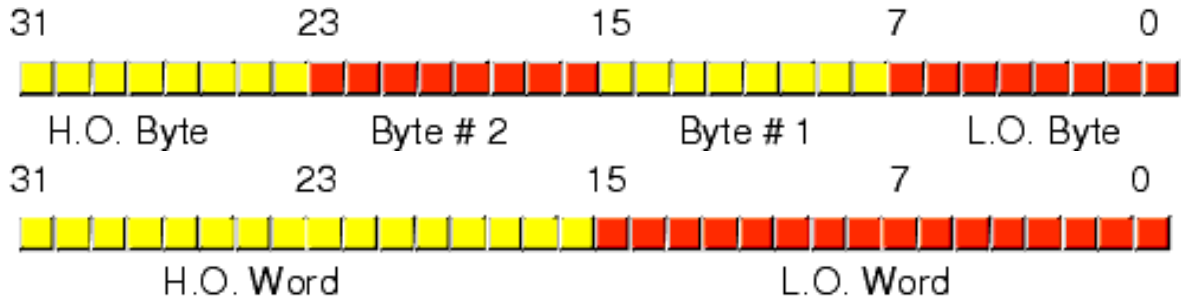
Hình 9 Đơn vị biểu diễn dữ liệu Byte

Word (từ) là đơn vị dữ liệu kế tiếp byte. Một word là một nhóm của 16 bits, hoặc 2 bytes. Một word có thể lưu đến 2^{16} (65536) giá trị, từ $(0000)_{16}$ đến $(FFFF)_{16}$. Hình 10 minh hoạ đơn vị biểu diễn dữ liệu word.



Hình 10 Đơn vị biểu diễn dữ liệu Word

Double words (từ kép) là đơn vị biểu diễn dữ liệu cơ sở lớn nhất. Một double word là một nhóm 32 bits, hoặc 4 bytes, hoặc 2 words. Một double word có thể lưu đến 2^{32} giá trị, từ $(0000\ 0000)_{16}$ đến $(FFFF\ FFFF)_{16}$. Hình 11 minh hoạ đơn vị biểu diễn dữ liệu double word.



Hình 11 Đơn vị biểu diễn dữ liệu Double word

1.6.3 Số có dấu và số không dấu

Trong các hệ thống tính toán, với cùng một số bit có thể biểu diễn các giá trị khác nhau nếu số được biểu diễn là có dấu hoặc không dấu. Để biểu diễn số có dấu, người ta sử dụng bit cao nhất (bên trái nhất) để biểu diễn dấu của số - gọi là bit dấu, chẳng hạn bit dấu có giá trị 0 là số dương và bit dấu có giá trị 1 là số âm. Với số không dấu, tất cả các bit được sử dụng để biểu diễn giá trị của số. Như vậy, miền giá trị có thể biểu diễn của một số gồm n bit như sau:

- Số có dấu: miền biểu diễn từ từ -2^{n-1} đến $+ 2^{n-1}$

- 8 bits: từ -128 đến +128
- 16 bits: từ -32768 đến +32768
- 32 bits: từ -2.147.483.648 đến +2.147.483.648

- Số không dấu: từ 0 đến 2^n
  - 8 bits: từ 0 đến 256
  - 16 bits: từ 0 đến 65536
  - 32 bits: từ 0 đến 4.294.967.296

1.6.4 Bảng mã ASCII

Bảng mã ASCII (American Standard Code for Information Interchange) là bảng mã các ký tự chuẩn tiếng Anh dùng cho trao đổi dữ liệu giữa các hệ thống tính toán. Bảng mã ASCII sử dụng 8 bit để biểu diễn 1 ký tự, cho phép định nghĩa tổng số 256 ký tự, đánh số từ 0 đến 255. 32 ký tự đầu tiên và ký tự số 127 là các ký tự điều khiển (không in ra được). Các ký tự từ số 32 đến 126 là các ký tự có thể in được (gồm cả dấu trắng). Các vị trí còn lại trong bảng (128-255) để dành cho sử dụng trong tương lai. Hình 12 và Hình 13 lần lượt là minh hoạ các ký tự điều khiển và các ký tự in được của bảng mã ASCII.

Binary	Oct	Dec	Hex	Abbr	PR[t 1]	CS[t 2]	CEC[t 3]	Description
000 0000	000	0	00	NUL	NUL	^@	\0	Null character
000 0001	001	1	01	SOH	SOH	^A		Start of Header
000 0010	002	2	02	STX	STX	^B		Start of Text
000 0011	003	3	03	ETX	ETX	^C		End of Text
000 0100	004	4	04	EOT	EOT	^D		End of Transmission
000 0101	005	5	05	ENQ	ENQ	^E		Enquiry
000 0110	006	6	06	ACK	ACK	^F		Acknowledgment
000 0111	007	7	07	BEL	BEL	^G	\a	Bell
000 1000	010	8	08	BS	BS	^H	\b	Backspace[t 4][t 5]
000 1001	011	9	09	HT	HT	^I	\t	Horizontal Tab
000 1010	012	10	0A	LF	LF	^J	\n	Line feed

Hình 12 Bảng mã ASCII - Một số ký tự điều khiển

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sp
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l

Hình 13 Bảng mã ASCII - Các ký tự in được

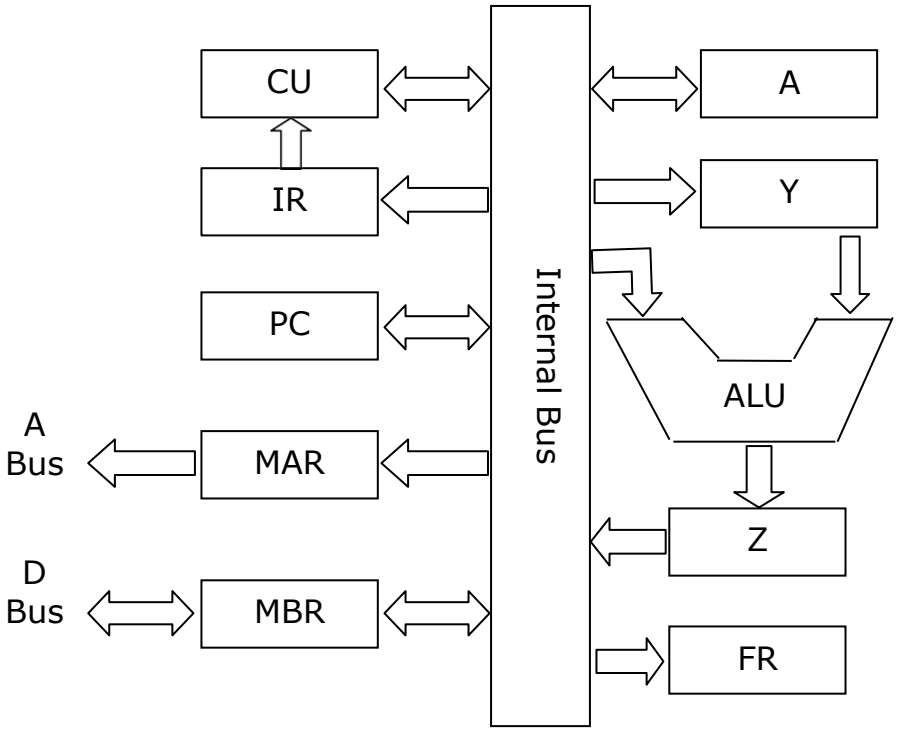
1.7 CÂU HỎI ÔN TẬP

1. Phân biệt khái niệm kiến trúc & tổ chức máy tính.
2. Nêu sơ đồ khối và mô tả chức năng từng khối của máy tính?
3. So sánh hai kiến trúc von-Neumann và Harvard.
4. Các hệ đếm 2, 10 và 16.
5. Các đơn vị lưu trữ dữ liệu trên máy tính.

CHƯƠNG 2 KHỐI XỬ LÝ TRUNG TÂM

2.1 SƠ ĐỒ KHỐI TỔNG QUÁT VÀ CHU TRÌNH XỬ LÝ LỆNH

2.1.1 Sơ đồ khối tổng quát của CPU



Hình 14 Sơ đồ khối tổng quát của CPU

Hình 14 trình bày sơ đồ khối nguyên lý tổng quát của CPU. Các thành phần của CPU theo sơ đồ này gồm:

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

2.1.2 Chu trình xử lý lệnh

Như đã trình bày trong chương 1, nhiệm vụ chủ yếu của CPU là đọc lệnh từ bộ nhớ, giải mã và thực hiện lệnh của chương trình. Khoảng thời gian để CPU thực hiện xong một lệnh kể từ khi CPU cấp phát tín hiệu địa chỉ ô nhớ chứa lệnh đến khi nó hoàn tất việc thực hiện lệnh được gọi là *chu kỳ lệnh* (Instruction Cycle). Mỗi chu kỳ lệnh của CPU được mô tả theo các bước sau:

1. Khi một chương trình được kích hoạt, hệ điều hành (OS - Operating System) nạp mã chương trình vào bộ nhớ trong;
2. Địa chỉ của ô nhớ chứa lệnh đầu tiên của chương trình được nạp vào bộ đếm chương trình PC;
3. Địa chỉ ô nhớ chứa lệnh từ PC được chuyển đến bus địa chỉ thông qua thanh ghi MAR;
4. Bus địa chỉ chuyển địa chỉ ô nhớ đến đơn vị quản lý bộ nhớ (MMU - Memory Management Unit);
5. MMU chọn ra ô nhớ và thực hiện lệnh đọc nội dung ô nhớ;
6. Lệnh (chứa trong ô nhớ) được chuyển ra bus dữ liệu và tiếp theo được chuyển tiếp đến thanh ghi MBR;
7. MBR chuyển lệnh đến thanh ghi lệnh IR; IR chuyển lệnh vào bộ điều khiển CU;
8. CU giải mã lệnh và sinh các tín hiệu điều khiển cần thiết, yêu cầu các bộ phận chức năng của CPU, như ALU thực hiện lệnh;
9. Giá trị địa chỉ trong bộ đếm PC được tăng lên 1 đơn vị lệnh và nó trỏ đến địa chỉ của ô nhớ chứa lệnh tiếp theo;
10. Các bước từ 3-9 được lặp lại với tất cả các lệnh của chương trình.

2.2 CÁC THANH GHI

2.2.1 Giới thiệu về thanh ghi

Thanh ghi (registers) là các ô nhớ bên trong CPU, có nhiệm vụ lưu trữ tạm thời lệnh và dữ liệu cho CPU xử lý. Thanh ghi thường có kích thước nhỏ, nhưng tốc độ làm việc rất cao - bằng tốc độ CPU. Các CPU cũ (80x86) có khoảng 16-32 thanh ghi. Các CPU hiện đại (như Pentium 4 và Core Duo) có thể có đến hàng trăm thanh ghi. Kích thước thanh ghi phụ thuộc vào thiết kế CPU. Các kích thước thông dụng của thanh ghi là 8, 16, 32, 64, 128 và 256 bit. CPU Intel 8086 và 80286 có các thanh ghi 8 bit và 16 bit. CPU Intel 80386 và Pentium II có các thanh ghi 16 bit và 32 bit. Các CPU Pentium 4 và Core Duo có các thanh ghi 32 bit, 64 bit và 128 bit.

2.2.1.1 Thanh tích luỹ A

Thanh tích luỹ A (Accumulator) là một trong các thanh ghi quan trọng nhất của CPU. Thanh ghi A không những được sử dụng để lưu toán hạng vào mà còn dùng để chứa kết quả ra. Ngoài ra, thanh ghi A còn thường được dùng trong các lệnh trao đổi dữ liệu với các thiết bị vào ra. Kích thước của thanh ghi A bằng kích thước từ xử lý của CPU: 8 bit, 16 bit, 32 bit hoặc 64 bit.

Ví dụ về việc sử dụng thanh ghi A trong phép toán: x + y → s

- Nạp toán hạng x vào thanh ghi A
- Nạp toán hạng y vào thanh ghi tạm thời Y
- ALU thực hiện phép cộng A + Y và lưu kết quả vào thanh ghi Z
- Kết quả phép tính từ Z được chuyển về thanh ghi A.
- Kết quả trong thanh ghi A được lưu vào ô nhớ s.

2.2.1.2 Bộ đếm chương trình PC

Bộ đếm chương trình PC (Program Counter) hoặc con trỏ lệnh (IP – Instruction pointer) luôn chứa địa chỉ của ô nhớ chứa lệnh kế tiếp được thực hiện. Đặc biệt, PC chứa địa chỉ của ô nhớ chứa lệnh đầu tiên của chương trình khi chương trình được kích hoạt và được hệ điều hành nạp vào bộ nhớ. Khi CPU thực hiện xong một lệnh, địa chỉ của ô nhớ chứa lệnh tiếp theo được nạp vào PC. Kích thước của PC phụ thuộc vào thiết kế CPU. Các kích thước thông dụng của PC là 8 bit, 16 bit, 32 bit và 64 bit.

2.2.1.3 Thanh ghi lệnh IR

Thanh ghi lệnh IR (Instruction register) lưu lệnh đang thực hiện. IR nhận lệnh từ MBR và chuyển tiếp lệnh đến CU giải mã và thực hiện.

2.2.1.4 Các thanh ghi MAR và MBR

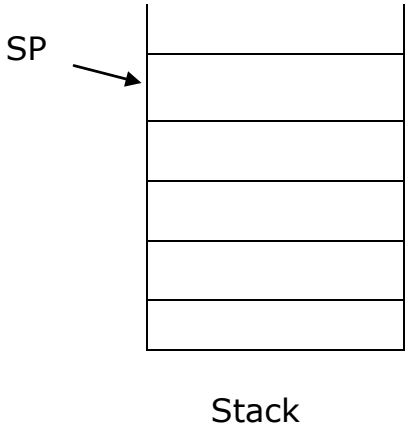
MAR là thanh ghi địa chỉ bộ nhớ (Memory address register) - giao diện giữa CPU và bus địa chỉ. MAR nhận địa chỉ ô nhớ chứa lệnh tiếp theo từ PC và chuyển tiếp ra bus địa chỉ.

MBR là thanh ghi đệm dữ liệu (Memory buffer register) - giao diện giữa CPU và bus địa chỉ. MBR nhận lệnh từ bus địa chỉ và chuyển tiếp lệnh đến IR thông qua bus trong CPU.

2.2.1.5 Các thanh ghi tạm thời

CPU thường sử dụng một số thanh ghi tạm thời để chứa toán hạng đầu vào và kết quả đầu ra, như các thanh ghi tạm thời X, Y và Z. Ngoài ra, các thanh ghi tạm thời còn tham gia trong việc hỗ trợ xử lý song song (thực hiện nhiều lệnh cùng một thời điểm) và hỗ trợ thực hiện lệnh theo cơ chế thực hiện tiên tiến kiểu không theo trật tự (OOO – Out Of Order execution).

2.2.1.6 Con trỏ ngăn xếp SP



Hình 15 Con trỏ ngăn xếp SP

Ngăn xếp (Stack) là bộ nhớ đặc biệt hoạt động theo nguyên lý vào sau ra trước (LIFO). Con trỏ ngăn xếp SP (Stack Pointer) là một thanh ghi luôn chứa địa chỉ đỉnh ngăn xếp. Có hai thao tác chính với ngăn xếp:

- Push - đẩy dữ liệu vào ngăn xếp:

  SP ← SP + 1 ; tăng địa chỉ đỉnh ngăn xếp

  {SP} ← Dữ liệu ; nạp dữ liệu vào ngăn xếp
- Pop - lấy dữ liệu ra khỏi ngăn xếp

  Thanh ghi ← {SP} ; chuyển dữ liệu từ đỉnh ngăn xếp vào thanh ghi

  SP ← SP – 1 ; giảm địa chỉ đỉnh ngăn xếp

2.2.1.7 Các thanh ghi tổng quát

Các thanh ghi tổng quát (General Purpose Registers) là các thanh ghi đa năng, có thể được sử dụng cho nhiều mục đích: để chứa toán hạng đầu vào hoặc chứa kết quả đầu ra. Chẳng hạn, CPU Intel 8086 có 4 thanh ghi tổng quát: AX - Thanh tích luỹ, BX - thanh ghi cơ sở, CX - thanh đếm và DX - thanh ghi dữ liệu.

2.2.1.8 Thanh ghi trạng thái FR

Thanh ghi trạng thái (SR - Status Register) hoặc thanh ghi cờ (FR – Flag Register) là một thanh ghi đặc biệt của CPU: mỗi bit của thanh ghi cờ lưu trạng thái của kết quả của phép tính ALU thực hiện. Có hai loại bit cờ: cờ trạng thái (CF, OF, AF, ZF, PF, SF) và cờ điều khiển (IF, TF, DF). Các bit cờ thường được sử dụng như là các điều kiện trong các lệnh rẽ nhánh để tạo logic chương trình. Kích thước của thanh ghi FR phụ thuộc thiết kế CPU.

Flag	ZF	SF	CF	AF	IF	OF	PF	1
Bit No	7	6	5	4	3	2	1	0

Hình 16 Các bit của thanh ghi cờ FR 8 bit

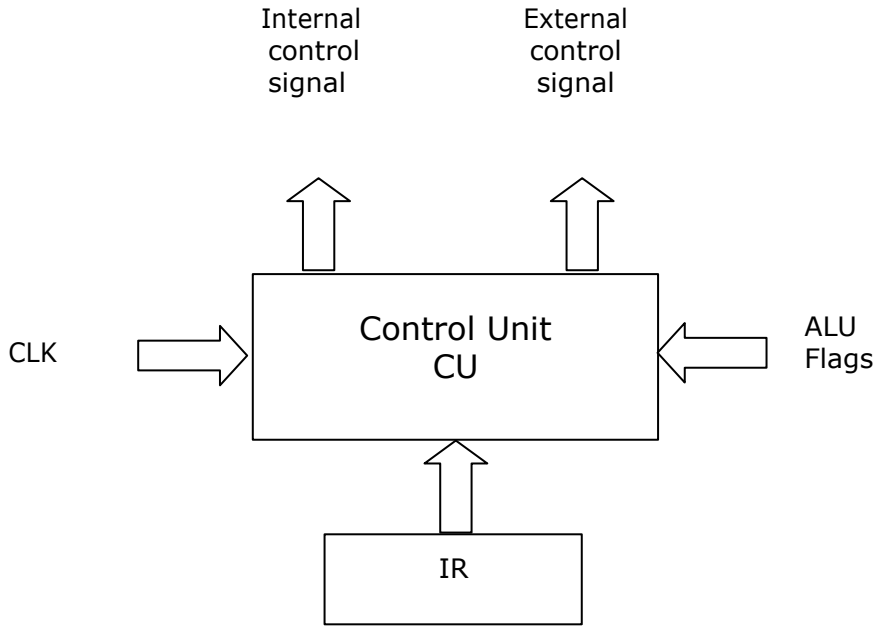
Hình 16 biểu diễn các bit của thanh ghi cờ FR. Ý nghĩa cụ thể của các bit như sau:

- ZF: Cờ Zero, ZF=1 nếu kết quả=0 và ZF=0 nếu kết quả<>0.
- SF: Cờ dấu, SF=1 nếu kết quả âm và SF=0 nếu kết quả dương.
- CF: Cờ nhớ, CF=1 nếu có nhớ/mượn, CF=0 trong trường hợp khác.
- AF: Cờ nhớ phụ, AF=1 nếu có nhớ/mượn ở nửa thấp của toán hạng.
- OF: Cờ tràn, OF=1 nếu xảy ra tràn, OF=0 trong trường hợp khác.
- PF: Cờ chẵn lẻ, PF=1 nếu tổng số bit 1 trong kết quả là lẻ và PF=0 nếu tổng số bit 1 trong kết quả là chẵn.
- IF: Cờ ngắt, IF=1: cho phép ngắt, IF=0: cấm ngắt.

2.3 KHỐI ĐIỀU KHIỂN

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

bên ngoài. Hình 17 minh hoạ phương thức làm việc của khối điều khiển CU. Khối điều khiển CU nhận ba tín hiệu đầu vào: (1) Lệnh từ thanh ghi lệnh IR, (2) Giá trị các cờ trạng thái của ALU và (3) Xung nhịp đồng hồ CLK và CU sản sinh hai nhóm tín hiệu đầu ra: (1) Nhóm tín hiệu điều khiển các bộ phận bên trong CPU (Internal control signal) và (2) Nhóm tín hiệu điều khiển các bộ phận bên ngoài CPU (External control signal).



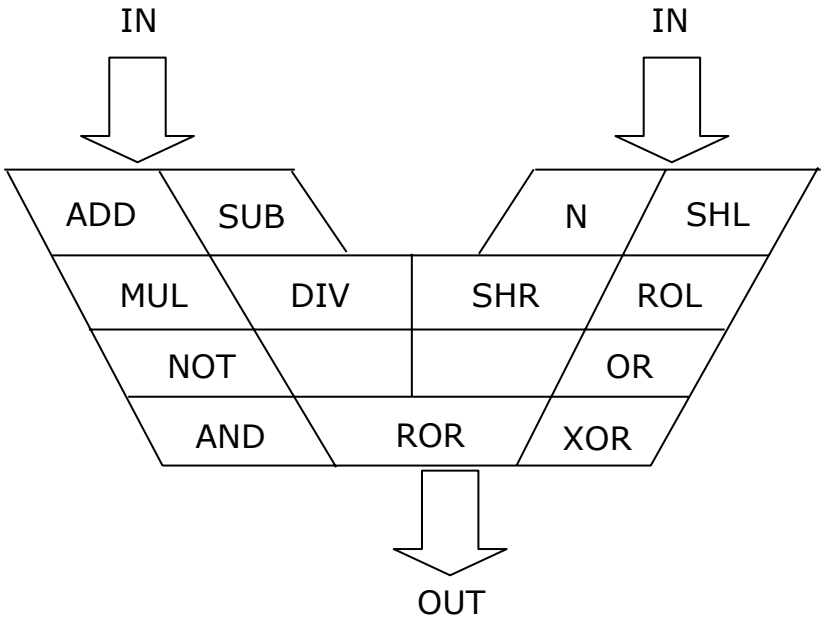
Hình 17 Khối điều khiển CU và các tín hiệu

2.4 KHỐI SỐ HỌC VÀ LOGIC

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

- Bộ cộng (ADD), bộ trừ (SUB), bộ nhân (MUL), bộ chia (DIV), ...
- Các bộ dịch (SHIFT) và quay (ROTATE)
- Bộ phủ định (NOT), bộ và (AND), bộ hoặc (OR) và bộ hoặc loại trừ (XOR)

Hình 18 minh hoạ các khối con của ALU cũng như các cổng vào và cổng ra của ALU. Hai cổng vào IN nhận các toán hạng đầu vào từ các thanh ghi và một cổng OUT kết nối với bus trong để chuyển kết quả tính toán đến thanh ghi.



Hình 18 Bộ tính toán ALU

2.5 BUS TRONG CPU

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

2.6 CÂU HỎI ÔN TẬP

1. Nêu sơ đồ khối tổng quát và các thành phần chính của CPU?
2. Nêu chu trình xử lý lệnh của CPU?
3. Nêu vai trò và chức năng của các thanh ghi của CPU?
4. Nêu sơ đồ và chức năng của CU và ALU?

CHƯƠNG 3 TẬP LỆNH MÁY TÍNH

3.1 GIỚI THIỆU VỀ TẬP LỆNH MÁY TÍNH

3.1.1 Lệnh máy tính là gì?

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

Vậy lệnh máy tính là gì? Có thể định nghĩa lệnh máy tính một cách đơn giản: Lệnh máy tính (Computer Instruction) là một từ nhị phân (binary word) được gán một nhiệm vụ cụ thể. Các lệnh của chương trình được lưu trong bộ nhớ và chúng lần lượt được CPU đọc, giải mã và thực hiện. Tập lệnh máy tính thường gồm nhiều lệnh có thể được chia thành một số nhóm theo chức năng: nhóm các lệnh vận chuyển dữ liệu (data movement), nhóm các lệnh tính toán (computational), nhóm các lệnh điều kiện và rẽ nhánh conditional and branching) và một số lệnh khác.

Việc thực hiện lệnh có thể được chia thành các pha (phase) hay giai đoạn (stage). Mỗi lệnh có thể được thực hiện theo 4 giai đoạn: (1) Đọc lệnh (Instruction fetch - IF): lệnh được đọc từ bộ nhớ về CPU; (2) Giải mã (Instruction decode - ID): CPU giải mã lệnh; (3) Thực hiện lệnh (Instruction execution – EX): CPU thực hiện lệnh; và (4) Lưu kết quả (Write back - WB): kết quả thực hiện lệnh (nếu có) được lưu vào bộ nhớ.

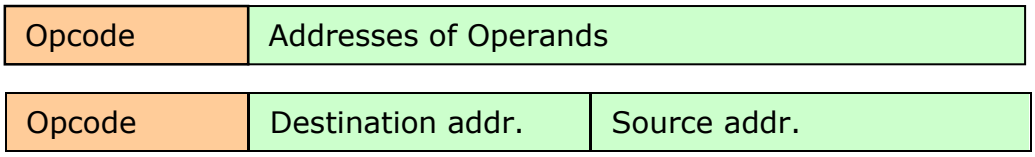
3.1.2 Chu kỳ thực hiện lệnh

Chu kỳ thực hiện lệnh (Instruction execution cycle) được định nghĩa là khoảng thời gian mà CPU thực hiện xong một lệnh. Một chu kỳ thực hiện lệnh có thể gồm một số giai đoạn thực hiện lệnh và một giai đoạn thực hiện lệnh có thể gồm một số chu kỳ máy. Một chu kỳ máy có thể gồm một số chu kỳ đồng hồ. Cụ thể hơn, chu kỳ thực hiện lệnh có thể gồm các thành phần sau:

- Chu kỳ đọc lệnh
- Chu kỳ đọc bộ nhớ (dữ liệu)
- Chu kỳ ghi bộ nhớ (dữ liệu)
- Chu kỳ đọc thiết bị ngoại vi
- Chu kỳ ghi thiết bị ngoại vi
- Chu kỳ bus rỗi.

3.2 DẠNG VÀ CÁC THÀNH PHẦN CỦA LỆNH

Dạng tổng quát của lệnh máy tính như minh hoạ trên Hình 19, gồm có 2 phần chính: (1) mã lệnh (opcode – operation code) và (2) địa chỉ của các toán hạng (Addresses of Operands). Mỗi lệnh có một mã lệnh riêng và được biểu diễn bằng một số bit. Chẳng hạn, mã lệnh của CPU Intel 8086 được biểu diễn bởi 6 bit. Mỗi lệnh có thể có một hoặc nhiều toán hạng và mỗi toán hạng là một địa chỉ. Tựu chung, có 5 dạng toán hạng của lệnh: 3 địa chỉ, 2 địa chỉ, 1 địa chỉ, 1,5 địa chỉ và 0 địa chỉ. Chi tiết về từng dạng toán hạng được trình bày trong mục 3.3.



Hình 19 Dạng và các thành phần của lệnh

3.3 CÁC DẠNG ĐỊA CHỈ / TOÁN HẠNG

3.3.1 Toán hạng dạng 3 địa chỉ

Dạng:

opcode addr1, addr2, addr3

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

Ví dụ:

ADD R_1, R_2, R_3; $R_1 \leftarrow R_2 + R_3$; R_2 cộng với R_3, kết quả lưu vào R_1.

R_i là thanh ghi của CPU.

ADD A, B, C; M[A] ← M[B] + M[C];

Lấy nội dung của ô nhớ B cộng với nội dung của ô nhớ C, kết qua lưu vào ô nhớ A

A, B, C là địa chỉ các ô nhớ. M[..] quy ước là phép tham chiếu nội dung ô nhớ.

3.3.2 Toán hạng dạng 2 địa chỉ

Dạng:

opcode addr1, addr2

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

Ví dụ:

ADD R_1, R_2; $R_1 \leftarrow R_1 + R_2$; R_1 cộng với R_2, kết quả lưu vào R_1.

R_i là thanh ghi của CPU.

ADD A, B; M[A] ← M[A] + M[B];

Lấy nội dung của ô nhớ A cộng với nội dung của ô nhớ B, kết qua lưu vào ô nhớ A

A, B là địa chỉ các ô nhớ.

3.3.3 Toán hạng dạng 1 địa chỉ

Dạng:

opcode addr2

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

Ví dụ:

ADD R_2; $R_{acc} \leftarrow R_{acc} + R_2$; R_{acc} cộng với R_2, kết quả lưu vào R_{acc}.

R_2 là thanh ghi của CPU.

ADD B; $R_{acc} \leftarrow R_{acc} + M[B]$;

Lấy nội dung của thanh ghi R_{acc} cộng với nội dung của ô nhớ B, kết qua lưu vào R_{acc}. A là địa chỉ một ô nhớ.

3.3.4 Toán hạng dạng 1,5 địa chỉ

Dạng:

opcode addr1, addr2

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

Ví dụ:

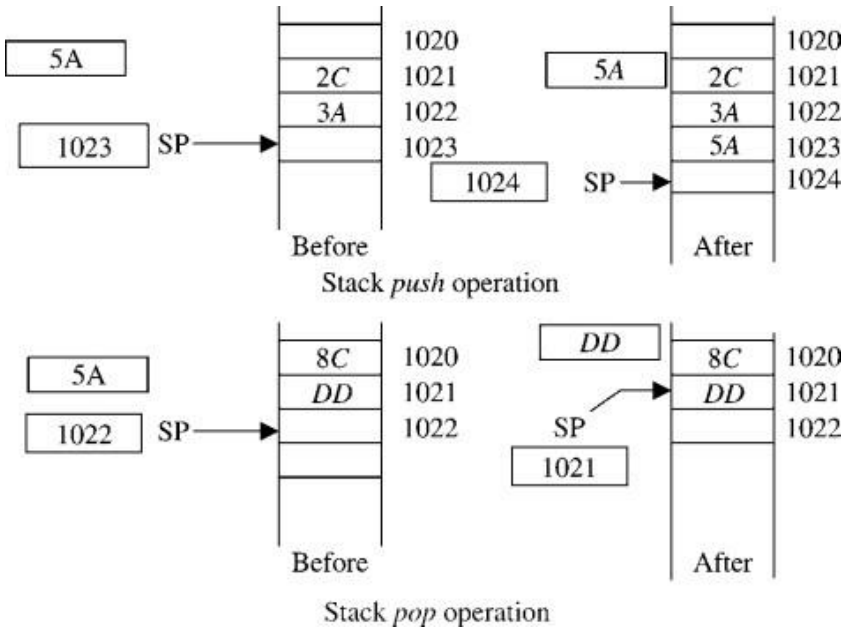
ADD R_1, A; $R_1 \leftarrow R_1 + M[A]$;

Lấy nội dung của R_1 cộng nội dung của ô nhớ A, kết quả lưu vào R_1.

R_1 là thanh ghi của CPU và A là địa chỉ ô nhớ.

3.3.5 Toán hạng dạng 0 địa chỉ

Toán hạng 0 địa chỉ thường được sử dụng trong các lệnh thao tác với ngăn xếp: PUSH và POP như minh hoạ trên Hình 20.



Hình 20 Thao tác PUSH và POP với ngăn xếp

3.4 CÁC CHẾ ĐỘ ĐỊA CHỈ

3.4.1 Giới thiệu về chế độ địa chỉ

Chế độ địa chỉ (Addressing modes) là phương thức hoặc cách thức CPU tổ chức các toán hạng của lệnh. Chế độ địa chỉ cho phép CPU kiểm tra dạng lệnh và tìm các toán hạng của lệnh. Số lượng các chế độ địa chỉ phụ thuộc vào thiết kế của CPU. Sau đây là một số chế độđịa chỉ thông dụng:

1. Tức thì (Immediate)

2. Trực tiếp (Direct)
3. Gián tiếp (indirect)
4. Chỉ số (Indexed)
5. Tương đối (Relative)

Mô tả chi tiết từng chế độ địa chỉ được thực hiện trong mục 3.4.2. Các ví dụ minh hoạ các chế độ địa chỉ sử dụng lệnh LOAD (nạp) với dạng sau:

LOAD <toán hạng đích> <toán hạng gốc>

Ý nghĩa: Nạp giá trị của <toán hạng gốc> vào <toán hạng đích>

Hay: <toán hạng đích> ← <toán hạng gốc>

3.4.2 Các chế độ địa chỉ

3.4.2.1 Chế độ địa chỉ tức thì (Immediate)

Trong chế độ địa chỉ tức thì, giá trị hằng của toán hạng nguồn (source operand) được đặt nằm ngay sau mã lệnh, còn toán hạng đích có thể là 1 thanh ghi hoặc 1 địa chỉ ô nhớ.

Ví dụ:

LOAD R1, #1000; R1 ← 1000 ; Nạp giá trị 1000 vào thanh ghi R1.

LOAD B, #100; M[B] ← 100 ; Nạp giá trị 100 vào ô nhớ B.

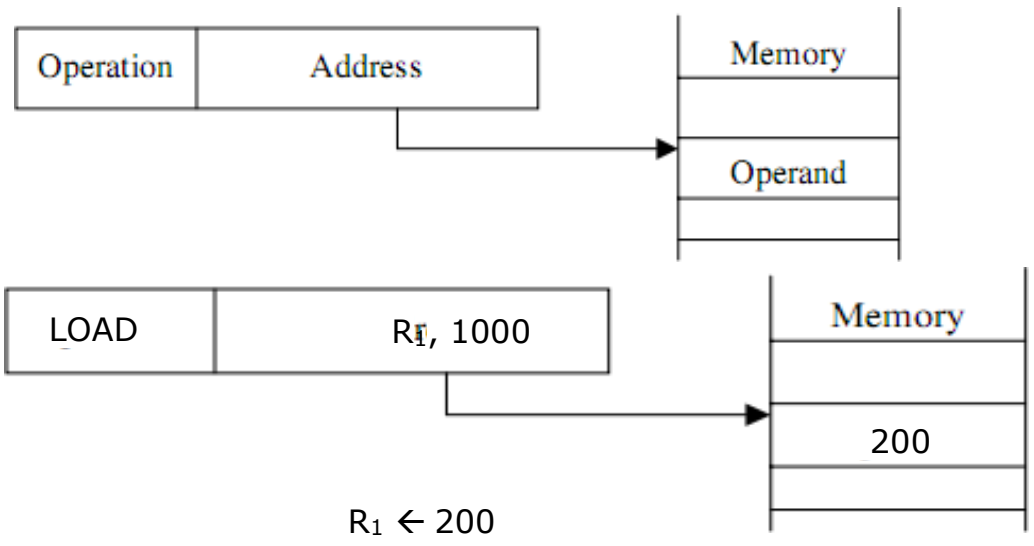
3.4.2.2 Chế độ địa chỉ trực tiếp (Direct)

Khác với chế độ địa chỉ tức thì, chế độ địa chỉ trực tiếp sử dụng một hằng để biểu diễn địa chỉ một ô nhớ làm một toán hạng. Toán hạng còn lại có thể là 1 thanh ghi hoặc 1 địa chỉ ô nhớ.

Ví dụ:

LOAD R1, 1000; R1 ← M[1000]

Nạp nội dung ô nhớ có địa chỉ 1000 vào thanh ghi R1. Hình 21 minh hoạ việc tham chiếu trong chế độ địa chỉ trực tiếp ở ví dụ trên. Địa chỉ 1000 trỏ đến ô nhớ chứa giá trị 200 và giá trị này được nạp vào thanh ghi R_1.



Hình 21 Tham chiếu với chế độ địa chỉ trực tiếp

3.4.2.3 Chế độ địa chỉ gián tiếp (Indirect)

Trong chế độ địa chỉ gián tiếp, một thanh ghi hoặc một ô nhớ được sử dụng để lưu địa chỉ một ô nhớ làm một toán hạng. Toán hạng còn lại có thể là một hằng, một thanh ghi hoặc địa chỉ một ô nhớ. Nếu thanh ghi được sử dụng để lưu địa chỉ ô nhớ ta có chế độ địa chỉ gián tiếp qua thanh ghi (register indirect); ngược lại nếu ô nhớ được dụng để lưu địa chỉ ô nhớ khác ta có chế độ địa chỉ gián tiếp qua ô nhớ (memory indirect).

Ví dụ:

Gián tiếp qua thanh ghi:

LOAD R_j, (R_i); $R_j \leftarrow M[R_i]$

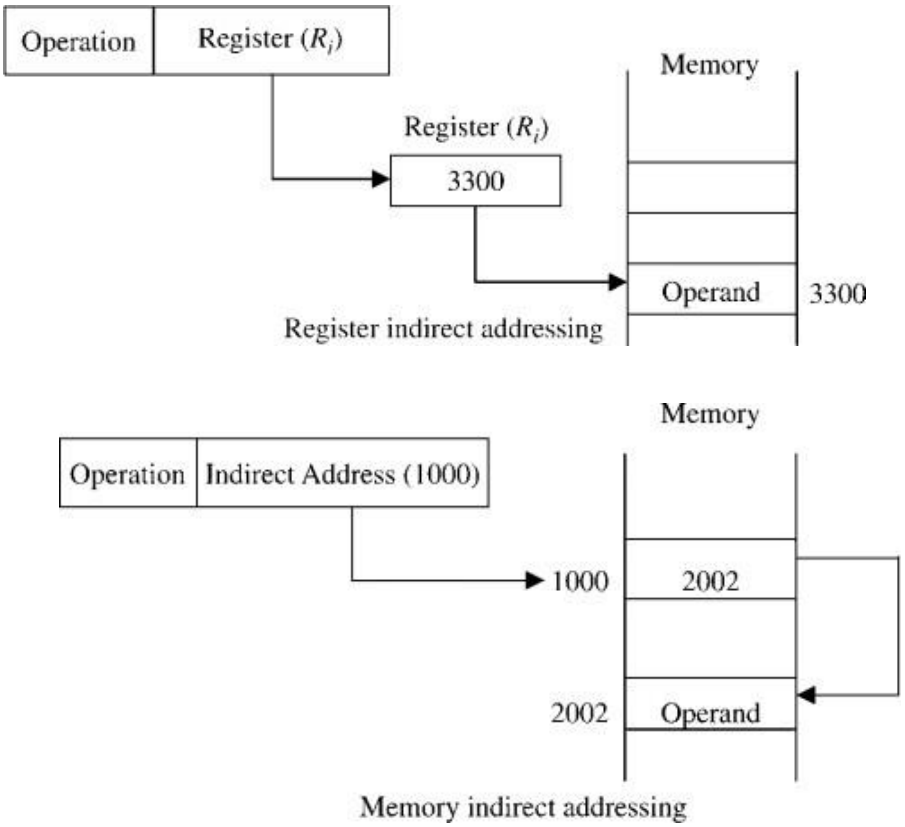
Nạp nội dung ô nhớ có địa chỉ lưu trong thanh ghi R_i vào thanh ghi R_j.

Gián tiếp qua ô nhớ:

LOAD R_i, (1000); $R_i \leftarrow M[M[1000]]$

Nạp nội dung ô nhớ có địa chỉ lưu trong ô nhớ 1000 vào thanh ghi R_i.

Hình 22 minh hoạ việc tham chiếu trong chế độ địa chỉ gián tiếp qua thanh ghi và gián tiếp qua ô nhớ. Có thể thấy rằng, chế độ địa chỉ gián tiếp qua thanh ghi chỉ yêu cầu một tham chiếu bộ nhớ cho một truy nhập, còn chế độ địa chỉ gián tiếp qua ô nhớ phải cần tới hai tham chiếu bộ nhớ cho một truy nhập.



Hình 22 Tham chiếu trong chế độ địa chỉ gián tiếp

3.4.2.4 Chế độ địa chỉ chỉ số (Indexed)

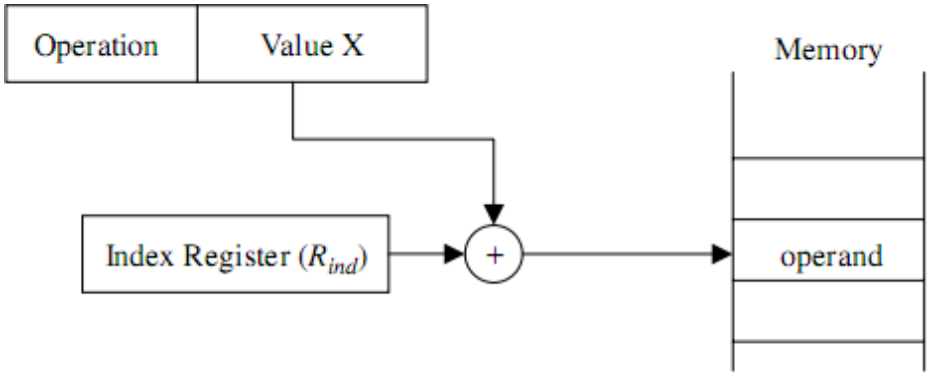
Trong chế độ địa chỉ chỉ số, địa chỉ của 1 toán hạng được tạo thành bởi phép cộng giữa 1 hằng và thanh ghi chỉ số (index register). Toán hạng còn lại có thể là một hằng, một thanh ghi hoặc địa chỉ một ô nhớ.

Ví dụ:

LOAD R_i, X(R_{ind}); R_i ← M[X+R_{ind}]

X là một hằng và R_{ind} là thanh ghi chỉ số.

Hình 23 minh hoạ phép tham chiếu trong chế độ địa chỉ chỉ số.



Hình 23 Tham chiếu trong chế độ địa chỉ chỉ số

3.4.2.5 Chế độ địa chỉ tương đối (Relative)

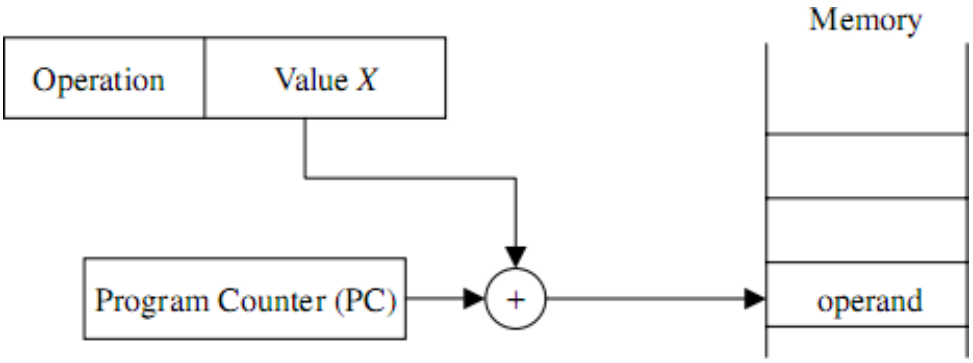
Trong chế độ địa chỉ tương đối, địa chỉ của 1 toán hạng được tạo thành bởi phép cộng giữa 1 hằng và bộ đếm chương trình PC (program counter). Toán hạng còn lại có thể là một hằng, một thanh ghi hoặc địa chỉ một ô nhớ.

Ví dụ:

LOAD R_i, X(PC); R_i ← M[X+PC]

X là một hằng và PC là bộ đếm chương trình.

Hình 24 minh hoạ phép tham chiếu trong chế độ địa chỉ chỉ số.



Hình 24 Tham chiếu trong chế độ địa chỉ tương đối

3.5 MỘT SỐ DẠNG LỆNH THÔNG DỤNG

Phụ thuộc thiết kế CPU, tập lệnh của CPU có thể có số lượng các lệnh rất khác nhau, từ vài chục lệnh đến vài trăm lệnh. Tuy nhiên, một hầu hết các tập lệnh máy tính thường bao gồm các nhóm lệnh cơ sở sau: (1) Các lệnh vận chuyển dữ liệu (Data Movement Instructions), (2) Các lệnh toán học và logic (Arithmetic and Logical Instructions), (3) Các lệnh điều khiển chương trình (Control/Sequencing Instructions) và (4) Các lệnh vào ra (Input/Output Instructions). Phần tiếp theo của mục này trình bày một số lệnh thông dụng thuộc các nhóm lệnh kể trên.

3.5.1 Các lệnh vận chuyển dữ liệu

Các lệnh vận chuyển dữ liệu vận chuyển dữ liệu giữa các bộ phận của máy tính. Cụ thể, vận chuyển dữ liệu giữa các thanh ghi của CPU, nạp dữ liệu từ các ô nhớ về các thanh ghi của CPU và ngược lại ghi dữ liệu từ các thanh ghi ra các ô nhớ. Ngoài ra, dữ liệu cũng có thể được vận chuyển giữa các ô nhớ trong bộ nhớ trong.

Ví dụ:

Vận chuyển dữ liệu giữa các thanh ghi của CPU:

MOVE R_i, R_j; $R_i \leftarrow R_j$

Chuyển (sao chép) nội dung của thanh ghi R_j sang thanh ghi R_i.

Vận chuyển dữ liệu giữa 1 thanh ghi của CPU và một ô nhớ:

MOVE 1000, R_j; M[1000] $\leftarrow R_j$

Lưu nội dung của thanh ghi R_j vào ô nhớ có địa chỉ 1000.

Vận chuyển dữ liệu giữa các ô nhớ:

MOVE 1000, (R_j); M[1000] $\leftarrow$ M[R_j]

Chuyển (sao chép) nội dung của ô nhớ có địa chỉ chứa trong thanh ghi R_j sang số nhớ có địa chỉ 1000.

Một số lệnh vận chuyển dữ liệu thông dụng

Tên lệnh	**Ý nghĩa**
MOVE	Chuyển dữ liệu giữa thanh ghi – thanh ghi, ô nhớ - thanh ghi và ô nhớ - ô nhớ.
LOAD	Nạp nội dung 1 ô nhớ vào 1 thanh ghi.
STORE	Lưu nội dung 1 thanh ghi ra 1 ô nhớ.
PUSH	Đẩy dữ liệu vào ngăn xếp.
POP	Lấy dữ liệu ra khỏi ngăn xếp.

3.5.2 Các lệnh toán học và logic

Các lệnh tính toán số học và logic được sử dụng để thực hiện các thao tác tính toán trên nội dung các thanh ghi và / hoặc nội dung các ô nhớ. Các lệnh tính toán hỗ trợ hầu hết các phép toán số học thông dụng như cộng, trừ, nhân, chia các số nguyên và các phép toán logic, như phủ định, và, hoặc, hoặc loại trừ.

Ví dụ:

Lệnh cộng:

ADD R_1, R_2, R_3; $R_1 \leftarrow R_2 + R_3$

Cộng nội dung 2 thanh ghi R_2 và R_3, kết quả lưu vào thanh ghi R_1.

ADD A, B, C; M[A] ← M[B] + M[C]

Cộng nội dung 2 ô nhớ B và C, kết quả lưu vào ô nhớ A.

Lệnh trừ:

SUBSTRACT R_1, R_2, R_3; $R_1 \leftarrow R_2 - R_3$

Lấy nội dung thanh ghi R_2 trừ đi nội dung thanh ghi R_3, kết quả lưu vào thanh ghi R_1.

Lệnh logic:

NOT R_1; $R_1 \leftarrow !(R_1)$

Lấy giá trị đảo (phủ định) của nội dung thanh ghi R_1.

AND R_1, R_2; $R_1 \leftarrow R_1 \otimes R_2$

Nhân bit nội dung 2 thanh ghi R_1 và R_2, kết quả lưu vào R_1.

Một số lệnh tính toán và logic thông dụng

Tên lệnh	**Ý nghĩa**
ADD	Cộng các toán hạng
SUBTRACT	Trừ các toán hạng
MULTIPLY	Nhân các toán hạng
DIVIDE	Chia các toán hạng
INCREMENT	Tăng một đơn vị
DECREMENT	Giảm một đơn vị
NOT	Phủ định bit
AND	Phép và (nhân) bit
OR	Phép hoặc (cộng) bit
XOR	Phép hoặc loại trừ bit
COMPARE	So sánh 2 toán hạng
SHIFT	Phép dịch bit (dịch trái, dịch phải)
ROTATE	Phép quay bit (quay trái, quay phải)

3.5.3 Các lệnh điều khiển chương trình

Các lệnh điều khiển chương trình được sử dụng để thay đổi trật tự thực hiện các lệnh khác trong chương trình hay làm thay đổi logic chương trình. Đây là nhóm lệnh gây ra các *rẽ nhánh* (branching), hoặc *nhảy* (jumping) làm cho quá trình thực hiện chương trình phức tạp hơn. Một trong các đặc tính của các lệnh này là chúng làm thay đổi nội dung của bộ đếm chương trình PC – nơi chứa địa chỉ ô nhớ chứa lệnh tiếp theo được thực hiện, có nghĩa là yêu

cầu CPU thực hiện chương trình từ một vị trí mới thay vì thực hiện lệnh kế tiếp lệnh đang thực hiện. Các lệnh điều khiển chương trình sử dụng các cờ của ALU (lưu trong thanh ghi cờ FR) để xác định điều kiện rẽ nhánh hoặc nhảy. Có thể chia các lệnh điều khiển chương trình thành 3 loại chính sau:

- Các lệnh nhảy / rẽ nhánh có điều kiện (CONDITIONAL BRANCHING/ CONDITIONAL JUMP);
- Các lệnh nhảy/ rẽ nhánh không điều kiện (UNCONDITIONAL BRANCHING / JUMP);
- Các lệnh gọi thực hiện (CALL) và trở về (RETURN) từ chương trình con.

Ví dụ: Cộng nội dung 100 ô nhớ cạnh nhau, bắt đầu từ địa chỉ 1000. Kết quả lưu vào R_0.

```
        LOAD R1, #100;      R1 ← 100
        LOAD R2, #1000;     R2 ← 1000
        LOAD R0, #0;        R0 ← 0
Loop:   ADD R0, (R2);       R0 ← R0 + M[R2]
        INCREMENT R2;       R2 ← R2 + 1
        DECREMENT R1;       R1 ← R1 – 1
        BRANCH-IF-GREATER-THAN Loop;
        ; Quay lại thực hiện lệnh sau nhãn Loop nếu R1 còn lớn hơn 0.
```

Một số lệnh điều khiển chương trình thông dụng

Tên lệnh	**Ý nghĩa**
BRANCH-IF-CONDITION	Chuyển đến thực hiện lệnh ở địa chỉ mới nếu điều kiện là đúng.
JUMP	Chuyển đến thực hiện lệnh ở địa chỉ mới.
CALL	Chuyển đến thực hiện chương trình con.
RETURN	Trở về (từ chương trình con) thực hiện tiếp chương trình gọi.

3.5.4 Các lệnh vào ra

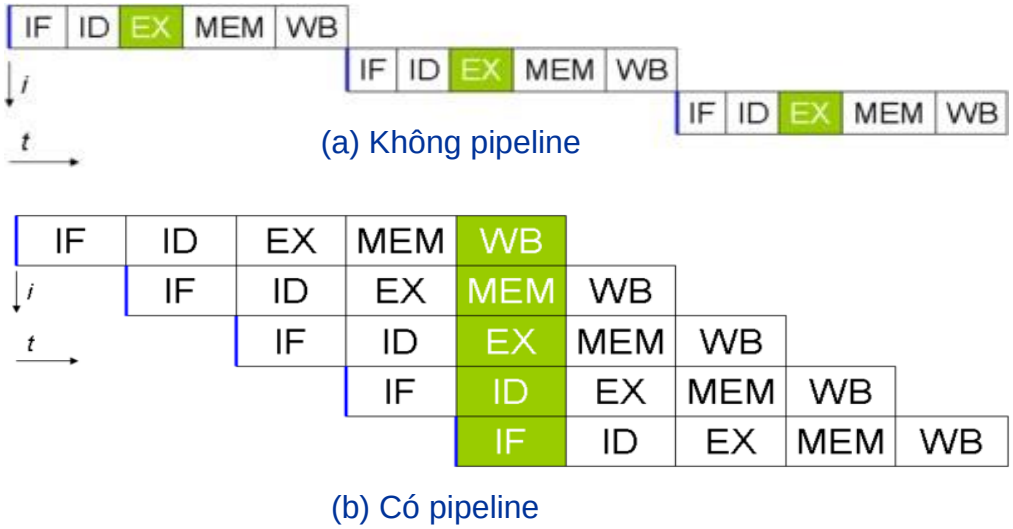
Các lệnh vào ra (I/O instructions) được sử dụng để vận chuyển dữ liệu giữa máy tính và các thiết bị ngoại vi. Các thiết bị ngoại vi giao tiếp với máy tính thông qua các cổng vào ra chuyên dụng (IO dedicated ports). Mỗi cổng vào ra được gán một địa chỉ riêng biệt. Có hai lệnh vào ra cơ bản:

- INPUT: sử dụng để chuyển dữ liệu từ thiết bị vào (input devices) đến CPU;
- OUTPUT: sử dụng để chuyển dữ liệu từ CPU đến thiết bị ra (output devices).

3.6 GIỚI THIỆU CƠ CHẾ ỐNG LỆNH (PIPELINE)

3.6.1 Giới thiệu cơ chế ống lệnh

Cơ chế ống lệnh (pipeline) hay còn gọi là cơ chế thực hiện xen kẽ các lệnh của chương trình là một phương pháp thực hiện lệnh tiên tiến, cho phép đồng thời thực hiện nhiều lệnh, giảm thời gian trung bình thực hiện mỗi lệnh và như vậy tăng được hiệu năng xử lý lệnh của CPU. Việc thực hiện lệnh được chia thành một số giai đoạn và mỗi giai đoạn được thực thi bởi một đơn vị chức năng khác nhau của CPU. Nhờ vậy CPU có thể tận dụng tối đa năng lực xử lý của các đơn vị chức năng của mình, giảm thời gian chờ cho từng đơn vị chức năng.



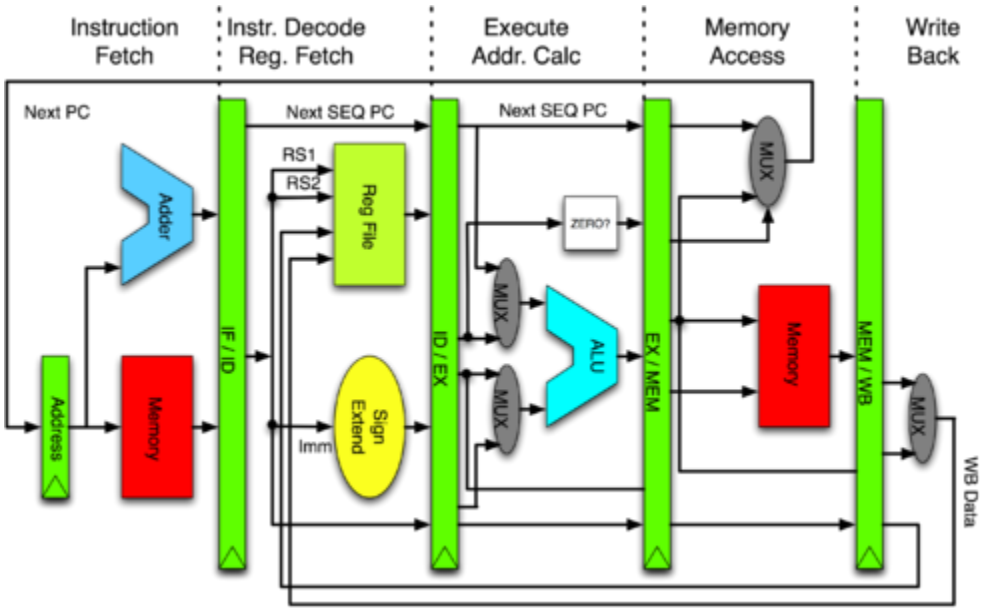
Hình 25 Thực hiện lệnh (a) không pipeline và (b) có pipeline

Hình 25 minh hoạ cơ chế thực hiện lệnh (a) không pipeline và (b) có pipeline. Trong đó, việc thực hiện lệnh được chia thành 5 giai đoạn:

- Instruction Fetch - IF: Đọc lệnh từ bộ nhớ (hoặc cache);
- Instruction Decode - ID: giải mã lệnh và đọc các toán hạng;
- Execute - EX: thực hiện lệnh; nếu là lệnh truy nhập bộ nhớ: tính toán địa chỉ bộ nhớ;
- Memory Access - MEM: Đọc/ghi bộ nhớ; no-op nếu không truy nhập bộ nhớ; no-op là giai đoạn chờ, tiêu tốn thời gian CPU, nhưng không thực hiện thao tác có nghĩa;
- Write Back - WB: Ghi kết quả vào các thanh ghi.

Có thể thấy, với cơ chế thực hiện không pipeline, tại mỗi thời điểm chỉ có một lệnh được thực hiện và chỉ có một đơn vị chức năng của CPU làm việc, các đơn vị chức năng khác trong trạng thái chờ. Ngược lại, với cơ chế thực hiện có pipeline, có nhiều lệnh đồng thời được thực hiện gối nhau trong CPU và hầu hết các đơn vị chức năng của CPU liên tục tham gia vào quá trình xử lý lệnh. Số lượng lệnh được xử lý đồng thời đúng bằng số giai đoạn thực hiện lệnh. Với 5 giai đoạn thực hiện lệnh, để xử lý 5 lệnh, CPU cần 9 nhịp đồng hồ với cơ chế thực hiện có pipeline, trong khi CPU cần đến 25 nhịp đồng hồ để thực hiện 5 lệnh với cơ chế thực hiện không pipeline. Hình 26 minh hoạ việc các đơn vị chức năng của CPU phối hợp thực hiện lệnh trong cơ chế pipeline.

Việc lựa chọn số giai đoạn thực hiện lệnh sao cho phù hợp là một trong các vấn đề quan trọng của cơ chế ống lệnh. Về mặt lý thuyết, thời gian thực hiện lệnh trung bình sẽ giảm khi tăng số giai đoạn thực hiện lệnh. Cho đến hiện nay, không có câu trả lời chính xác về số giai đoạn thực hiện lệnh tối ưu mà nó phụ thuộc nhiều vào thiết kế của CPU. Với các CPU cũ (họ Intel 80x86 và tương đương) số giai đoạn là 3 đến 5. Với các CPU Intel Pentium III và Pentium M, Core Duo, Core 2 Duo số giai đoạn là khoảng 10 đến 15. Riêng họ Intel Pentium IV có số giai đoạn vào khoảng 20 và cá biệt phiên bản Intel Pentium IV Prescott chia việc thực hiện lệnh thành 31 giai đoạn.



Hình 26 Thực hiện lệnh theo cơ chế pipeline với các đơn vị chức năng của CPU

3.6.2 Các vấn đề của cơ chế ống lệnh và hướng giải quyết

Như đã trình bày, cơ chế ống lệnh giúp giảm thời gian trung bình thực hiện từng lệnh và tăng đáng kể hiệu suất xử lý lệnh của CPU. Tuy nhiên, cơ chế ống lệnh cũng gặp phải một số vấn đề làm giảm hiệu suất thực hiện lệnh. Tựu chung, có ba vấn đề thường gặp với cơ chế ống lệnh: (1) Vấn đề xung đột tài nguyên (resource conflicts), (2) Vấn đề tranh chấp dữ liệu (Data hazards) và (3) Vấn đề nảy sinh do các lệnh rẽ nhánh (Branch instructions). Trong phạm vi của bài giảng này, hướng giải quyết các vấn đề của cơ chế ống lệnh chỉ dừng ở mức giới thiệu phương pháp.

3.6.2.1 Vấn đề xung đột tài nguyên

Vấn đề xung đột tài nguyên xảy ra khi hệ thống không cung cấp đủ tài nguyên phần cứng phục vụ CPU thực hiện đồng thời nhiều lệnh trong cơ chế ống lệnh. Hai xung đột tài nguyên thường gặp nhất là xung đột truy cập bộ nhớ và xung đột truy cập các thanh ghi. Giả sử bộ nhớ chỉ hỗ trợ một truy cập tại mỗi thời điểm và nếu tại cùng một thời điểm, có hai yêu cầu truy cập bộ nhớ đồng thời từ 2 lệnh được thực hiện trong ống lệnh (đọc lệnh – tại giai đoạn IF và đọc dữ liệu – tại giai đoạn ID) sẽ nảy sinh xung đột. Điều tương tự cũng có thể xảy ra với các thanh ghi khi có 2 hay nhiều lệnh đang thực hiện đồng yêu cầu đọc/ghi cùng một thanh ghi.

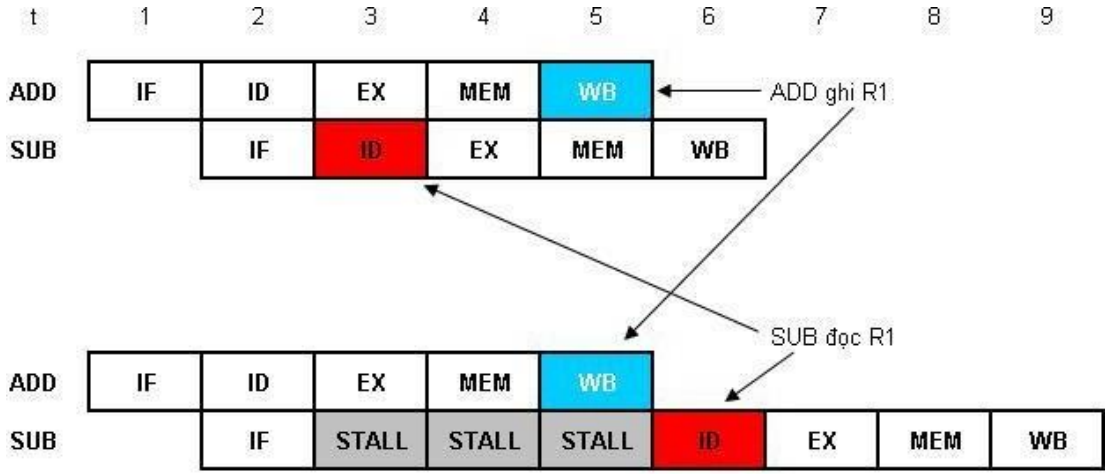
Giải pháp tối ưu cho vấn đề xung đột tài nguyên là nâng cao năng lực phục vụ của các tài nguyên phần cứng. Với xung đột truy cập bộ nhớ có thể sử dụng hệ thống nhớ hỗ trợ nhiều lệnh đọc ghi đồng thời, hoặc sử dụng các bộ nhớ tiên tiến như bộ nhớ cache. Với xung đột truy cập các thanh ghi, giải pháp là tăng số lượng thanh ghi vật lý và có cơ chế cấp phát thanh ghi linh hoạt khi thực hiện các lệnh.

3.6.2.2 Vấn đề tranh chấp dữ liệu

Tranh chấp dữ liệu cũng là một trong các vấn đề lớn của cơ chế ống lệnh và tranh chấp dữ liệu kiểu *đọc sau khi ghi* (RAW – Read After Write) là dạng xung đột dữ liệu hay gặp nhất. Để hiểu rõ tranh chấp dữ liệu kiểu RAW, ta xem xét hai lệnh sau:

ADD R_1, R_2, R_3; $R_1 \leftarrow R_2 + R_3$ (1)

SUB R_4, R_1, R_2; $R_4 \leftarrow R_1 + R_2$ (2)



Hình 27 Tranh chấp dữ liệu kiểu RAW

Hình 27 minh hoạ tranh chấp dữ liệu kiểu RAW giữa hai lệnh ADD và SUB được thực hiện kề nhau trong cơ chế ống lệnh. Có thể thấy lệnh SUB sử dụng kết quả của lệnh ADD (thanh ghi R_1 là kết quả của ADD và là đầu vào cho SUB) và như vậy hai lệnh có sự phụ thuộc dữ liệu. Tuy nhiên, lệnh SUB đọc thanh ghi R_1 tại giai đoạn giải mã (ID), trước khi lệnh ADD ghi kết quả vào thanh ghi R_1 ở giai đoạn lưu kết quả (WB). Như vậy, giá trị SUB đọc được từ thanh ghi R_1 là giá trị cũ, không phải là kết quả tạo ra bởi ADD. Để SUB đọc được giá trị mới nhất của R_1, giai đoạn ID của SUB phải lùi 3 nhịp, đến vị trí giai đoạn WB của ADD kết thúc.

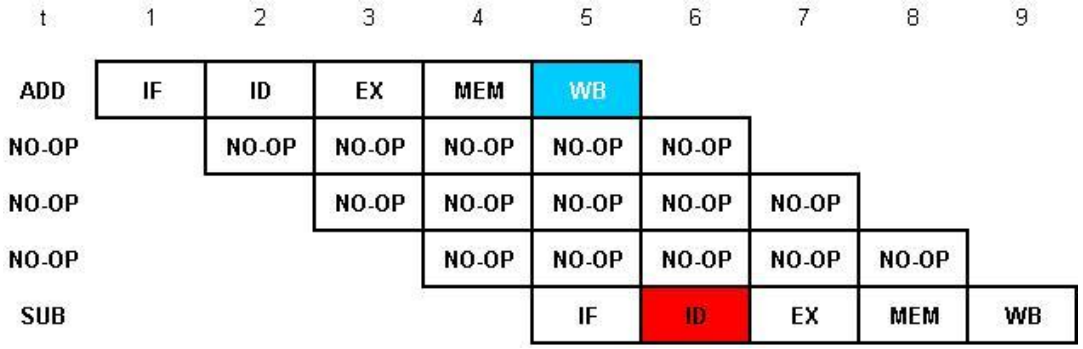
Có một số giải pháp cho vấn đề tranh chấp dữ liệu kiểu RAW. Cụ thể:

1. Nhận dạng tranh chấp RAW khi nó diễn ra;
2. Khi tranh chấp RAW xảy ra, tạm dừng (stall) ống lệnh cho đến khi lệnh phía trước hoàn tất giai đoạn WB;
3. Có thể sử dụng trình biên dịch (compiler) để nhận dạng tranh chấp RAW và thực hiện:
   - Chèn thêm các lệnh NO-OP vào giữa các lệnh có thể gây ra tranh chấp RAW; NO-OP là lệnh rỗng, không thực hiện tác vụ hữu ích mà chỉ tiêu tốn thời gian CPU.
   - Thay đổi trật tự các lệnh trong chương trình và chèn các lệnh độc lập vào giữa các lệnh có thể gây ra tranh chấp RAW;

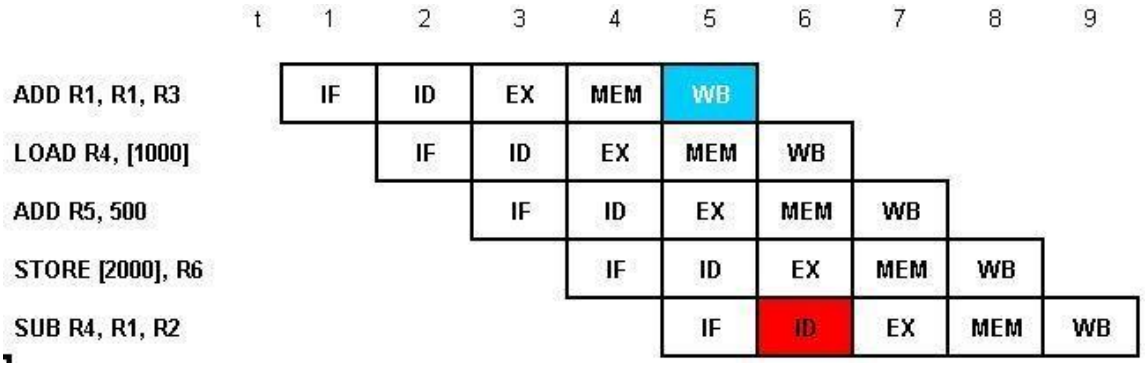
Mục đích của cả hai phương pháp kể trên là lùi việc thực hiện lệnh gây tranh chấp dữ liệu cho đến khi lệnh trước nó hoàn tất việc lưu kết quả.

4. Sử dụng phần cứng để nhận dạng tranh chấp RAW và dự đoán trước giá trị dữ liệu phụ thuộc.

Hình 28 minh hoạ giải pháp khắc phục tranh chấp RAW bằng cách chèn thêm các lệnh NO-OP. Hình 29 minh hoạ giải pháp khắc phục tranh chấp RAW bằng cách chèn thêm các lệnh độc lập với hai lệnh có tranh chấp. Các lệnh độc lập có thể có được bằng cách thay đổi trật tự thực hiện các lệnh của chương trình mà không thay đổi kết quả thực hiện nó. Cũng có thể sử dụng giải pháp kết hợp chèn NO-OP và lệnh độc lập.



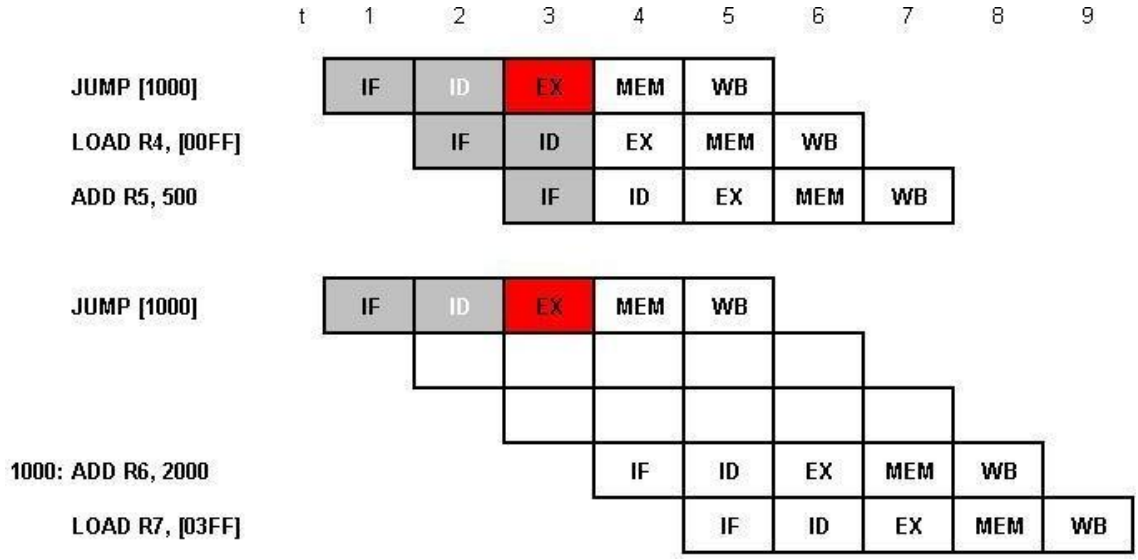
Hình 28 Khắc phục tranh chấp RAW bằng chèn thêm NO-OP



Hình 29 Khắc phục tranh chấp RAW bằng chèn các lệnh độc lập

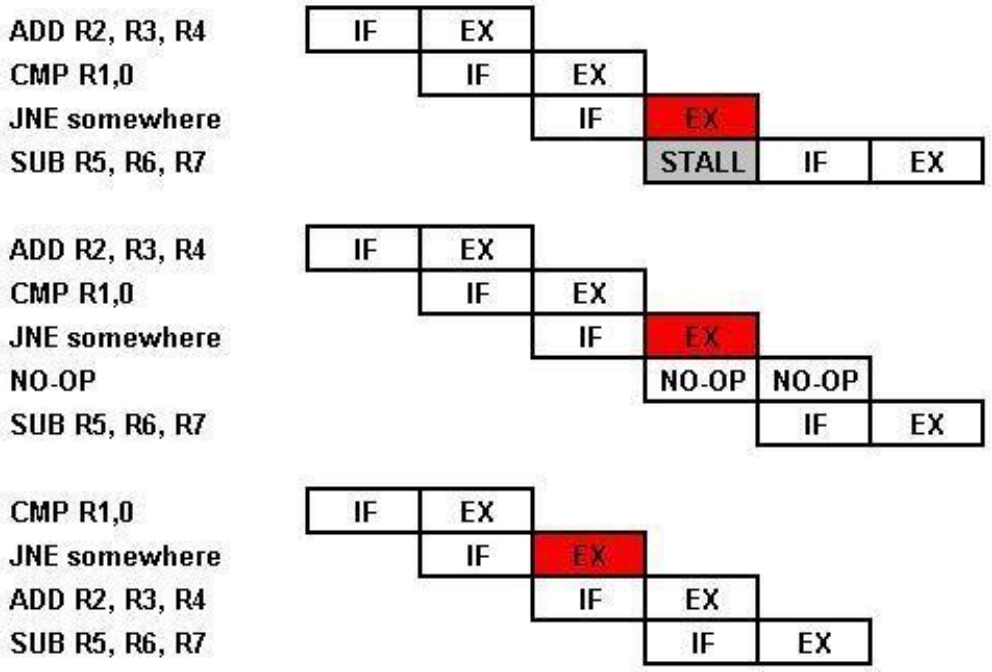
3.6.2.3 Vấn đề nảy sinh do các lệnh rẽ nhánh

Theo thống kê, tỷ lệ các lệnh rẽ nhánh trong chương trình khoảng 10-30%. Do lệnh rẽ nhánh thay đổi nội dung của bộ đếm chương trình, chúng có thể phá vỡ tiến trình thực hiện tuần tự các lệnh trong ống lệnh vì lệnh được thực hiện sau lệnh rẽ nhánh có thể không phải là lệnh liền sau nó mà là một lệnh ở vị trí khác. Như vậy, do kiểu thực hiện gối đầu, các lệnh liền sau lệnh rẽ nhánh đã được nạp và thực hiện dở dang trong trong ống lệnh sẽ bị đẩy ra làm cho ống lệnh bị trống rỗng và hệ thống phải bắt đầu nạp mới các lệnh từ địa chỉ đích rẽ nhánh. Hình 30 minh họa vấn đề nảy sinh trong ống lệnh do lệnh rẽ nhánh. Các lệnh sau lệnh rẽ nhánh LOAD và ADD bị đẩy ra và hệ thống nạp mới các lệnh từ địa chỉ đích rẽ nhánh 1000.



Hình 30 Vấn đề nảy sinh do lệnh rẽ nhánh

Có nhiều giải pháp khắc phục các vấn đề nảy sinh do các lệnh rẽ nhánh, như sử dụng đích rẽ nhánh (branch targets), làm chậm rẽ nhánh (delayed branching) và dự đoán rẽ nhánh (branch prediction). Tài liệu này chỉ giới thiệu phương pháp làm chậm rẽ nhánh. Ý tưởng chính của phương pháp làm chậm rẽ nhánh là lệnh rẽ nhánh sẽ không gây ra sự rẽ nhánh tức thì mà được làm "trễ" một số chu kỳ, phụ thuộc vào chiều dài của ống lệnh. Phương pháp này cho hiệu quả khá tốt với các ống lệnh ngắn, thường là 2 giai đoạn và với ràng buộc lệnh ngay sau lệnh rẽ nhánh luôn được thực hiện, không phụ thuộc vào kết quả của lệnh rẽ nhánh. Cách thực hiện của phương pháp chậm rẽ nhánh là chèn thêm một lệnh NO-OP hoặc một lệnh độc lập vào ngay sau lệnh rẽ nhánh. Hình 31 minh hoạ vấn đề nảy sinh do lệnh rẽ nhánh có điều kiện JNE (nhảy nếu R1 không bằng 0), giải pháp chèn một lệnh NO-OP hoặc một lệnh độc lập vào sau lệnh nhảy để khắc phục.



Hình 31 Khắc phục vấn đề lệnh rẽ nhánh bằng cách chèn NO-OP hoặc lệnh độc lập

3.7 CÂU HỎI ÔN TẬP

1. Khái niệm lệnh và tập lệnh? Chu kỳ lệnh và các giai đoạn thực hiện lệnh.
2. Dạng lệnh và các dạng địa chỉ toán hạng.
3. Khái niệm chế độ địa chỉ và các chế độ địa chỉ.
4. Nêu một số dạng lệnh thông dụng.
5. Nguyên lý hoạt động của cơ chế ống lệnh của CPU?
6. Các vấn đề của cơ chế ống lệnh của CPU và hướng khắc phục.

CHƯƠNG 4 BỘ NHỚ TRONG

4.1 PHÂN LOẠI BỘ NHỚ MÁY TÍNH

4.1.1 Phân loại bộ nhớ

Bộ nhớ máy tính gồm nhiều thành phần với tốc độ truy cập và và dung lượng khác nhau được kết hợp với nhau tạo thành hệ thống nhớ. Có nhiều cách phân loại bộ nhớ máy tính. Tựu chung, có thể chia bộ nhớ máy tính dựa trên ba tiêu chí: (1) kiểu truy cập, (2) khả năng duy trì dữ liệu và (3) công nghệ chế tạo.

Dựa trên kiểu truy cập, có thể chia bộ nhớ máy tính thành ba loại: Bộ nhớ truy cập tuần tự (Serial Access Memory - SAM), bộ nhớ truy nhập ngẫu nhiên (Random Access Memory - RAM), và bộ nhớ chỉ đọc (Read Only Memory - ROM). Trong bộ nhớ truy cập tuần tự, các ô nhớ được truy cập một cách tuần tự, có nghĩa là muốn truy cập đến ô nhớ sau phải duyệt qua ô nhớ trước nó. Tốc độ truy cập các ô nhớ có vị trí khác nhau là không giống nhau. Ngược lại, trong bộ nhớ truy nhập ngẫu nhiên, các ô nhớ có thể được truy cập ngẫu nhiên, không theo một trật tự định trước. Với bộ nhớ chỉ đọc, thông tin được ghi vào bộ nhớ một lần nhờ một thiết bị đặc biệt và sau đó chỉ có thể đọc ra.

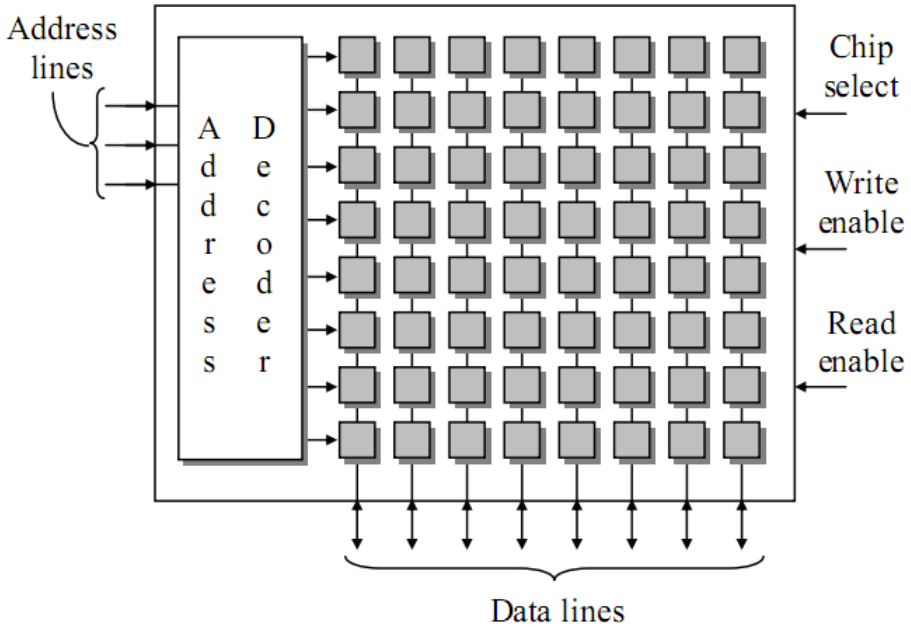
Dựa trên khả năng duy trì dữ liệu, có hai loại bộ nhớ: bộ nhớ ổn định (Non-volatile memory) và bộ nhớ không ổn định (Volatile memory). Bộ nhớ ổn định có khả năng duy trì dữ liệu kể cả khi không có nguồn nuôi. Đại diện tiêu biểu cho bộ nhớ ổn định là bộ nhớ ROM. Ngược lại, thông tin trong bộ nhớ không ổn định chỉ tồn tại khi có nguồn nuôi và sẽ mất khi mất nguồn nuôi. Đại diện tiêu biểu cho bộ nhớ không ổn định là bộ nhớ RAM.

Dựa trên công nghệ chế tạo, có ba loại bộ nhớ: bộ nhớ bán dẫn (Semiconductor memory), bộ nhớ từ tính (Magnetic memory), bộ nhớ quang học (Optical memory). Bộ nhớ bán dẫn được chế tạo bằng vật liệu bán dẫn, thường có tốc độ truy cập rất cao, nhưng giá thành đắt. Đại diện cho bộ nhớ bán dẫn là bộ nhớ ROM và RAM. Bộ nhớ từ tính là bộ nhớ dựa trên từ tính của các vật liệu có khả năng nhiễm từ để lưu trữ và đọc / ghi thông tin. Đại diện cho bộ nhớ từ tính là các loại đĩa từ (đĩa mềm, đĩa cứng) và băng từ. Bộ nhớ quang học là bộ nhớ hoạt động dựa trên các nguyên lý quang – điện. Đại diện cho bộ nhớ quang học là các loại đĩa quang, như đĩa CD, DVD,...

4.1.2 Tổ chức mạch nhớ

Một mạch nhớ (memory chip) thường gồm nhiều ô nhớ (memory cells) được tổ chức thành một ma trận nhớ gồm một số hàng và một số cột. Hình 32 minh hoạ tổ chức một mạch nhớ RAM. Ngoài ma trận nhớ gồm các ô nhớ, mạch nhớ còn gồm các đường địa chỉ (Address lines), bộ giải mã địa chỉ (Address decoder), các đường dữ liệu (Data lines) và các tín hiệu điều khiển như tín hiệu chọn mạch (Chip select - CS), tín hiệu cho phép đọc (Read enable - RE) và tín hiệu cho phép ghi (Write enable - WE).

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



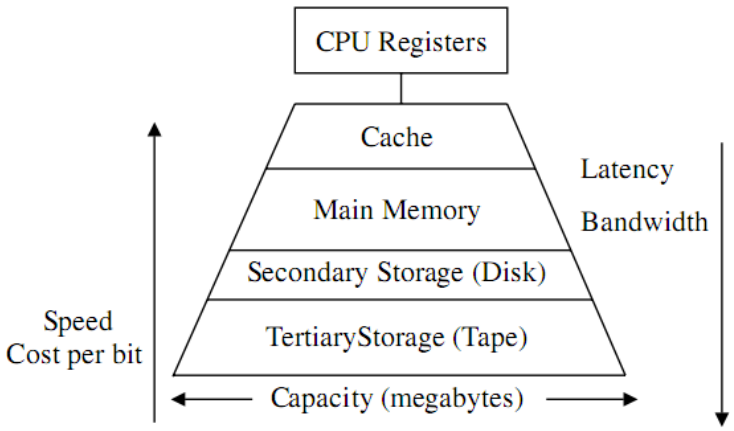
Hình 32 Tổ chức mạch nhớ

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

4.2 CẤU TRÚC PHÂN CẤP BỘ NHỚ MÁY TÍNH

4.2.1 Giới thiệu cấu trúc phân cấp hệ thống nhớ

Hầu hết hệ thống nhớ trong các thiết bị tính toán hiện đại được tổ chức theo cấu trúc phân cấp (hierarchical structure). Cấu trúc phân cấp không chỉ được sử dụng trong các hệ thống nhớ mà nó còn sử dụng rộng rãi trong đời sống xã hội, như cấu trúc tổ chức các cơ quan nhà nước, doanh nghiệp và cả các trường học. Hình 33 minh hoạ cấu trúc phân cấp hệ thống nhớ, gồm các phần chính: các thanh ghi của CPU (CPU Registers), bộ nhớ cache (Cache), bộ nhớ chính (Main Memory) và bộ nhớ ngoài (Secondary / Tertiary Storage).



Hình 33 Cấu trúc phân cấp hệ thống nhớ

	Access type	Capacity	Latency	Bandwidth	Cost/MB
CPU registers	Random	64–1024 bytes	1–10 ns	System clock rate	High
Cache memory	Random	8–512 KB	15–20 ns	10–20 MB/s	$500
Main memory	Random	16–512 MB	30–50 ns	1–2 MB/s	$20–50
Disk memory	Direct	1–20 GB	10–30 ms	1–2 MB/s	$0.25
Tape memory	Sequential	1–20 TB	30–10,000 ms	1–2 MB/s	$0.025

Hình 34 Dung lượng, thời gian truy cập và giá thành các loại bộ nhớ

Trong cấu trúc phân cấp hệ thống nhớ, dung lượng các thành phần tăng theo chiều từ các thanh ghi của CPU đến bộ nhớ ngoài. Ngược lại, tốc độ truy nhập hay băng thông và giá thành một đơn vị nhớ tăng theo chiều từ bộ nhớ ngoài đến các thanh ghi của CPU. Như vậy, các thanh ghi của CPU có dung lượng nhỏ nhất nhưng có tốc độ truy cập nhanh nhất và cũng có giá thành cao nhất. Bộ nhớ ngoài có dung lượng lớn nhất, nhưng tốc độ truy cập thấp nhất. Bù lại, bộ nhớ ngoài có giá thành rẻ nên có thể được sử dụng với dung lượng lớn.

Các thanh ghi được tích hợp trong CPU và thường hoạt động theo tần số làm việc của CPU, nên đạt tốc độ truy cập rất cao. Tuy nhiên, do không gian trong CPU rất hạn chế nên tổng dung lượng của các thanh ghi là khá nhỏ, chỉ khoảng vài chục byte đến vài kilobyte. Các thanh ghi thường được sử dụng để lưu toán hạng đầu vào và kết quả đầu ra của các lệnh phục vụ CPU làm việc.

Bộ nhớ cache có dung lượng tương đối nhỏ, khoảng từ vài chục kilobyte đến vài chục megabyte (khoảng 64KB đến 32MB với các máy tính hiện nay). Tốc độ truy cập cache cao, nhưng giá thành còn khá đắt. Cache được coi là bộ nhớ "thông minh" do có khả năng đoán trước được nhu cầu lệnh và dữ liệu của CPU. Cache "đoán" và tải trước các lệnh và dữ liệu CPU cần sử dụng từ bộ nhớ chính, nhờ vậy giúp CPU giảm thời gian truy cập hệ thống nhớ, tăng tốc độ xử lý.

Bộ nhớ chính gồm có bộ nhớ ROM và bộ nhớ RAM, có dung lượng khá lớn (khoảng từ 256MB đến 4GB với các hệ thống 32 bit), nhưng tốc độ truy cập tương đối chậm so với cache. Giá thành bộ nhớ chính tương đối thấp nên có thể sử dụng với dung lượng lớn. Bộ nhớ chính được sử dụng để lưu lệnh và dữ liệu của hệ thống và của người dùng.

Bộ nhớ ngoài hay bộ nhớ thứ cấp, gồm các loại đĩa từ, đĩa quang và băng từ. Bộ nhớ ngoài thường có dung lượng rất lớn, khoảng 20GB đến 1000GB, nhưng tốc độ truy cập rất chậm. Bộ nhớ ngoài có ưu điểm là giá thành rẻ và thường được sử dụng để lưu trữ dữ liệu lâu dài dưới dạng các tệp (files).

4.2.2 Vai trò của cấu trúc phân cấp hệ thống nhớ

Không hoàn toàn giống với vai trò của cấu trúc phân cấp trong các cơ quan và doanh nghiệp là "chia để trị", cấu trúc phân cấp trong hệ thống nhớ có hai vai trò chính: (1) tăng hiệu năng hệ thống thông qua việc giảm thời gian truy cập các ô nhớ và (2) giảm giá thành sản xuất.

Sở dĩ cấu trúc phân cấp trong hệ thống nhớ có thể giúp tăng hiệu năng hệ thống là do nó giúp dung hoà được CPU có tốc độ cao và phần bộ nhớ chính và bộ nhớ ngoài có tốc độ thấp. CPU sẽ chủ yếu trực tiếp truy cập bộ nhớ cache có tốc độ cao, và cache sẽ có nhiệm vụ chuyển

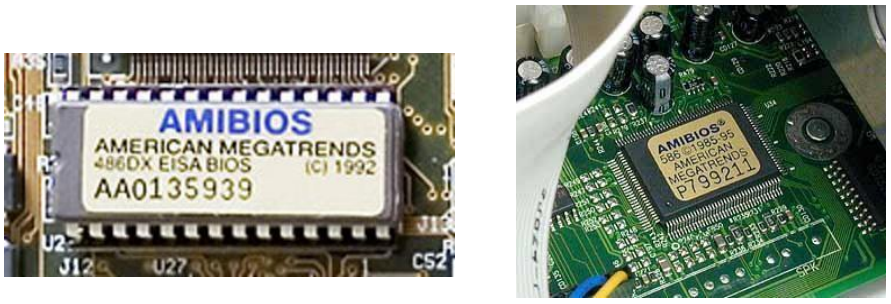
trước các dữ liệu cần thiết về từ bộ nhớ chính. Nhờ vậy, CPU sẽ không phải thường xuyên truy cập trực tiếp bộ nhớ chính và bộ nhớ ngoài để tìm dữ liệu – các thao tác tốn nhiều thời gian do các bộ nhớ này có tốc độ chậm. Như vậy, có thể nói rằng, thời gian trung bình CPU truy nhập dữ liệu từ hệ thống nhớ tiệm cận thời gian truy nhập bộ nhớ cache.

Cùng với việc có thể giúp cải thiện hiệu năng, cấu trúc phân cấp trong hệ thống nhớ có thể giúp giảm giá thành chế tạo hệ thống. Cơ sở chính là trong hệ thống nhớ phân cấp, các thành phần có tốc độ cao và đắt tiền được sử dụng với dung lượng rất nhỏ, còn các thành phần có tốc độ thấp và rẻ tiền được sử dụng với dung lượng lớn hơn. Nhờ vậy có thể giảm được giá thành chế tạo hệ thống nhớ mà vẫn đảm bảo được tốc độ cao cho cả hệ thống. Nếu ta có hai hệ thống nhớ hoạt động với cùng tốc độ thì hệ thống nhớ phân cấp sẽ có giá thành thấp hơn.

4.3 BỘ NHỚ ROM VÀ RAM

4.3.1 Bộ nhớ ROM

ROM (Read Only Memory) là bộ nhớ chỉ đọc, có nghĩa là thông tin lưu trữ trong ROM chỉ có thể đọc ra mà không được ghi vào. Trên thực tế, việc ghi thông tin vào ROM chỉ có thể được thực hiện bằng các thiết bị chuyên dùng hoặc phương pháp đặc biệt. Thông tin trong ROM thường được các nhà sản xuất ghi sẵn, gồm các thông tin về hệ thống như thông tin về cấu hình máy và hệ thống các mô đun phần mềm phục vụ việc vào ra cơ sở (BIOS - Basic Input Output System). ROM thuộc loại bộ nhớ bán dẫn và là bộ nhớ ổ định - thông tin trong ROM vẫn được duy trì kể cả khi không có nguồn điện nuôi. Hình 35 minh hoạ vi mạch nhớ ROM- BIOS được gắn trên bảng mạch chính.



Hình 35 Vi mạch nhớ ROM-BIOS

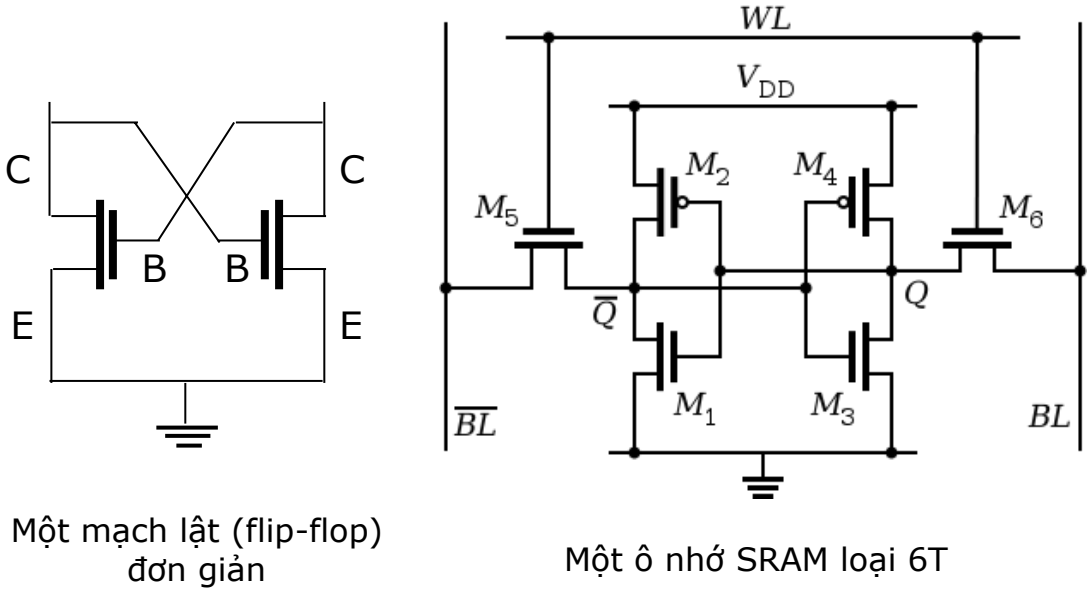
Quá trình phát triển ROM đã trải qua nhiều thế hệ. ROM các thế hệ đầu tiên hay còn gọi là ROM nguyên thuỷ (Ordinary ROM) sử dụng tia cực tím để ghi thông tin. Trong thế hệ tiếp theo - ROM có thể lập trình được (PROM - Programmable ROM), thông tin có thể được ghi vào PROM nhờ một thiết bị đặc biệt gọi là bộ lập trình PROM. Tiến thêm một bước, với ROM có thể lập trình và xoá được (EPROM - Erasable programmable read-only memory), thông tin trong EPROM có thể xoá được sử dụng tia cực tím có cường độ cao. Kế tiếp EPROM, EEPROM (Electrically Erasable PROM) là loại ROM tiến tiến nhất hiện nay. EEPROM có thể xoá được bằng điện và có thể ghi được thông tin sử dụng phần mềm chuyên dụng. Bộ nhớ Flash là một dạng bộ nhớ EEPROM được dùng phổ biến làm thiết bị lưu trữ trong các thiết bị cầm tay. Flash có tốc độ đọc ghi thông tin nhanh hơn EEPROM và thông tin được đọc ghi theo từng khối.

4.3.2 Bộ nhớ RAM

Bộ nhớ RAM được chế tạo theo công nghệ bán dẫn và thuộc loại bộ nhớ không ổn định, tức là, thông tin trong RAM chỉ tồn tại khi có nguồn điện nuôi và mất khi không còn nguồn điện nuôi. RAM là bộ nhớ cho phép truy cập ngẫu nhiên – các ô nhớ của RAM có thể được truy cập một cách ngẫu nhiên không theo trật tự nào và tốc truy cập các ô nhớ là tương đương nhau. RAM thường có dung lượng lớn hơn nhiều so với ROM và thường được sử dụng để lưu trữ các thông tin của hệ thống và của người dùng.

Có hai loại RAM cơ bản: RAM tĩnh (Static RAM hay SRAM) và RAM động (Dynamic RAM hay DRAM). Mỗi bit RAM tĩnh cấu tạo dựa trên một *mạch lật* (flip flop) – còn gọi là *mạch trigơ lưỡng ổn* (bistable latching circuit). Thông tin trong SRAM luôn ổn định và không phải "làm tươi" định kỳ. Tốc độ truy nhập SRAM cũng nhanh hơn nhiều so với DRAM. Ngược lại, mỗi bit DRAM cấu tạo dựa trên một tụ điện. Do bản chất của tụ điện luôn có khuynh hướng tự phóng điện tích, thông tin trong bit DRAM sẽ dần bị mất. Vì vậy, DRAM cần được làm tươi (refresh) định kỳ để bảo toàn thông tin. DRAM thường có tốc độ truy cập thấp hơn so với SRAM, nhưng bù lại, DRAM có cấu trúc gọn nhẹ nên có thể tăng mật độ cấy linh kiện dẫn đến giá thành một đơn vị nhớ DRAM thấp hơn SRAM.

4.3.2.1 Bộ nhớ SRAM



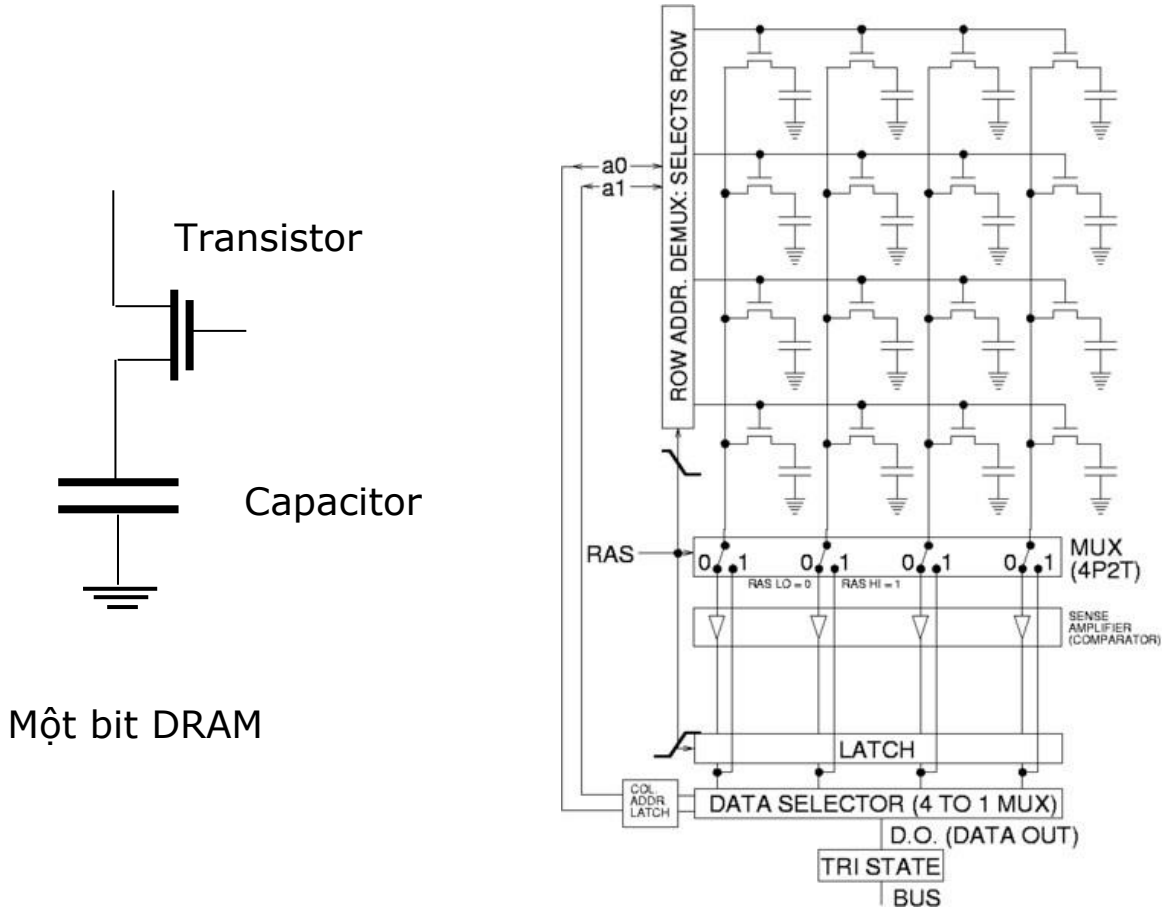
Hình 36 Cấu tạo một mạch lật trong bộ nhớ SRAM

Hình 36 minh hoạ cấu tạo một mạch lật đơn giản gồm 2 transistor và một mạch lật phức tạp hơn với 6 transistor (6T) – hình thành một bit nhớ của SRAM. Thông thường, mỗi bit nhớ của SRAM được cấu tạo từ một mạch lật 6T, 8T hoặc 10T. SRAM có tốc độ truy cập cao là do các bit SRAM có cấu trúc đối xứng và thông tin trong bit SRAM ổn định nên không cần quá trình làm tươi. Tuy nhiên, do mỗi bit SRAM cần nhiều transistor và có cấu trúc khá phức tạp nên mật độ cấy linh kiện thường thấp và giá thành SRAM khá cao.

4.3.2.2 Bộ nhớ DRAM

Khác với SRAM, các bit DRAM được hình thành dựa trên tụ điện. Hình 37 minh hoạ một bit DRAM và mạch nhớ DRAM tổ chức thành ma trận nhớ gồm các hàng và cột. Mỗi bit DRAM

có cấu tạo khá đơn giản, gồm 1 tụ điện và 1 transistor cấp nguồn. Mức điện tích trong tụ điện được sử dụng để biểu diễn các giá trị 0 và 1, chẳng hạn mức đầy điện tích ứng với mức 1, không tích điện ứng với mức 0.



Hình 37 Một bit DRAM và mạch nhớ DRAM

Do bản chất tụ thường tự phóng điện nên điện tích trong tụ có xu hướng giảm dần dẫn đếnthông tin trong tụ cũng bị mất theo. Để tránh bị mất thông tin, điện tích trong tụ cần được nạp lại thường xuyên – quá trình này được gọi là quá trình làm tươi các bit DRAM. DRAMthường có tốc độ truy cập chậm hơn so với SRAM là do: (1) có trễ khi nạp điện vào tụ, (2) cần quá trình làm tươi cho tụ và (3) các mạch DRAM thường dùng kỹ thuật dồn kênh (địa chỉcột/hàng) để tiết kiệm đường địa chỉ. Tuy nhiên, do mỗi bit DRAM có cấu trúc đơn giản, sử dụng ít transistor nên mật độ cấy linh kiện thường cao và giá thành rẻ hơn nhiều so với SRAM.

Trong các loại DRAM, SDRAM (Synchronous DRAM) được sử dụng phổ biến nhất. SDRAM là DRAM hoạt động đồng bộ với nhịp đồng hồ của bus. SDRAM được chia thành 2loại theo khả năng truyền dữ liệu: (1) SDR SDRAM (Single Data Rate SDRAM) – SDRAM có tỷ suất dữ liệu đơn, chấp nhận một thao tác đọc/ghi và chuyển 1 từ dữ liệu trong 1 chu kỳ đồng hồ với các tần số làm việc 100MHz và 133MHz và (2) DDR SDRAM (Double Data Rate SDRAM) - SDRAM có tỷ suất dữ liệu kép, chấp nhận hai thao tác đọc/ghi và chuyển 2 từ dữ liệu trong 1 chu kỳ đồng hồ. DDR SDRAM có 3 loại cho đến hiện nay:

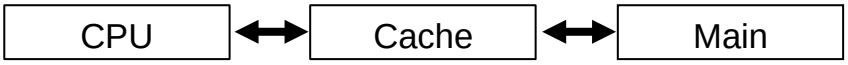
- DDR1 SDRAM: tần số làm việc 266, 333, 400 MHz: có khả năng chuyển 2 từ dữ liệu trong 1 chu kỳ đồng hồ;

- DDR2 SDRAM: tần số làm việc 400, 533, 800 MHz: có khả năng chuyển 4 từ dữ liệu trong 1 chu kỳ đồng hồ;
- DDR3 SDRAM: tần số làm việc 800, 1066, 1333, 1600 MHz: có khả năng chuyển 8 từ dữ liệu trong 1 chu kỳ đồng hồ.

4.4 BỘ NHỚ CACHE

4.4.1 Cache là gì?

Cache hay còn gọi là bộ nhớ đệm, bộ nhớ khay là một thành phần của cấu trúc phân cấp của hệ thống bộ nhớ như trình bày trong mục 4.2. Cache đóng vai trong trung gian, trung chuyển dữ liệu từ bộ nhớ chính về CPU và ngược lại. Hình 38 minh hoạ vị trí của bộ nhớ cache trong hệ thống nhớ. Với các hệ thống CPU cũ sử dụng công nghệ tích hợp thấp, bộ nhớ cache thường nằm ngoài CPU; với các CPU mới sử dụng công nghệ tích hợp cao, bộ nhớ cache thường được tích hợp vào trong CPU nhằm nâng cao tốc độ và băng thông trao đổi dữ liệu giữa CPU và cache.



Hình 38 Vị trí của bộ nhớ cache trong hệ thống nhớ

Dung lượng của bộ nhớ cache thường nhỏ so với dung lượng của bộ nhớ chính và bộ nhớ ngoài. Với các hệ thống máy tính cũ, dung lượng cache là khoảng 16KB, 32KB,..., 128KB; với các hệ thống máy tính gần đây, dung lượng cache lớn hơn, khoảng 256KB, 512KB, 1MB, 2MB, 4MB, 8MB và 16MB. Cache có tốc độ truy cập nhanh hơn nhiều so với bộ nhớ chính, đặc biệt với cache được tích hợp vào CPU. Tuy nhiên, giá thành bộ nhớ cache (tính theo bit) thường đắt hơn nhiều so với bộ nhớ chính. Với các hệ thống CPU mới, cache thường được chia thành hai hay nhiều mức (levels): mức 1 có dung lượng khoảng 16-32KB có tốc độ truy cập rất cao và mức 2 có dung lượng khoảng 1-16MB có tốc độ truy cập thấp hơn.

4.4.2 Vai trò và nguyên lý hoạt động

4.4.2.1 Vai trò của cache

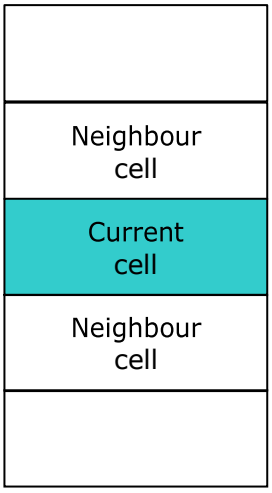
Do nhớ cache là một thành phần của hệ thống nhớ phân cấp, nên vai trò của cache tương tự như vai trò của cấu trúc phân cấp hệ thống nhớ: tăng hiệu năng hệ thống vào giảm giá thành sản xuất. Sở dĩ cache có thể giúp tăng hiệu năng hệ thống là nhờ cache có khả năng dung hoà được CPU có tốc độ cao và bộ nhớ chính có tốc độ thấp làm cho thời gian trung bình CPU truy nhập dữ liệu từ bộ nhớ chính tiệm cận thời gian truy nhập cache. Ngoài ra, do cache là một loại bộ nhớ "thông minh" có khả năng đoán và chuẩn bị trước các dữ liệu cần thiết cho CPU xử lý nên xác suất CPU phải trực tiếp truy nhập dữ liệu từ bộ nhớ chính là khá thấp và điều này cũng giúp làm giảm thời gian trung bình CPU truy nhập dữ liệu từ bộ nhớ chính.

Tuy cache có giá thành trên một đơn vị nhớ cao hơn bộ nhớ chính, nhưng do tổng dung lượng cache thường khá nhỏ nên cache không làm tăng giá thành hệ thống nhớ quá mức. Nhờ vậy, cache hoàn toàn phù hợp với cấu trúc phân cấp và có thể giúp làm giảm giá thành sản xuất trong tương quan với tốc độ của cả hệ thống nhớ. Có thể kết luận rằng, nếu hai hệ thống nhớ

có cùng giá thành, hệ thống nhớ có cache có tốc độ truy cập nhanh hơn; và nếu hai hệ thống nhớ có cùng tốc độ, hệ thống nhớ có cache sẽ có giá thành rẻ hơn.

4.4.2.2 Nguyên lý hoạt động của cache

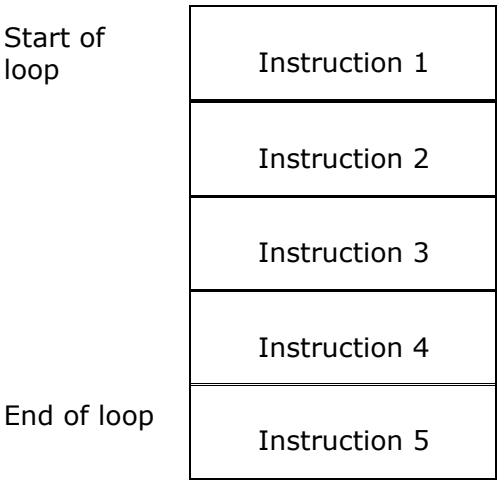
Cache sở dĩ được coi là bộ nhớ "thông minh" là do nó có khả năng đoán trước yêu cầu về dữ liệu và lệnh của CPU. Dữ liệu và lệnh cần thiết được chuyển trước từ bộ nhớ chính về cache và CPU chỉ cần truy nhập cache, giúp giảm thời gian truy nhập hệ thống nhớ. Để có được sự thông minh, cache hoạt động dựa trên hai nguyên lý cơ bản: nguyên lý *lân cận về không gian* (Spatial locality) và nguyên lý *lân cận về thời gian* (Temporal locality).



Hình 39 Lân cận về không gian trong không gian chương trình

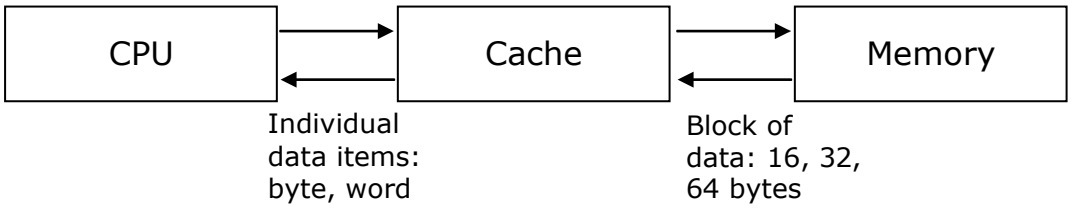
Nguyên lý lân cận về không gian có thể phát biểu như sau: "*Nếu một ô nhớ đang được truy nhập thì xác suất các ô nhớ liền kề với nó được truy nhập trong tương lai gần là rất cao*". Lân cận về không gian thường được áp dụng cho nhóm lệnh hoặc dữ liệu có tính tuần tự cao trong không gian chương trình, như minh hoạ trên Hình 39. Do các lệnh trong một chương trình thường tuần tự, cache có thể đọc cả khối lệnh từ bộ nhớ chính và khối lệnh đọc được bao phủ cả các ô nhớ lân cận (neighbour cell) của ô nhớ đang được truy nhập (current cell).

Khác với nguyên lý lân cận về không gian, nguyên lý lân cận về thời gian chú trọng hơn đến tính *lặp lại* của việc truy nhập các mẩu thông tin trong một khoảng thời gian tương đối ngắn. Có thể phát biểu nguyên lý này như sau: "*Nếu một ô nhớ đang được truy nhập thì xác suất nó được truy nhập lại trong tương lai gần là rất cao*". Lân cận về thời gian được áp dụng cho dữ liệu và nhóm các lệnh trong vòng lặp như minh hoạ trên Hình 40. Với các phần tử dữ liệu, chúng được CPU cập nhập thường xuyên trong quá trình thực hiện chương trình nên có tính lân cận cao về thời gian. Với các lệnh trong vòng lặp, chúng thường được CPU thực hiện lặp đi lặp lại nhiều lần nên cũng có tính lân cận cao về thời gian; nếu cache nạp sẵn khối lệnh chứa cả vòng lặp sẽ phủ được tính lân cận về thời gian.



Hình 40 Lân cận về thời gian với việc thực hiện vòng lặp

4.4.2.3 Trao đổi dữ liệu giữa CPU – cache – bộ nhớ chính



Hình 41 Trao đổi dữ liệu giữa CPU với cache và bộ nhớ chính

Hình 41 minh hoạ việc trao đổi dữ liệu giữa CPU với cache và bộ nhớ chính: CPU trao đổi dữ liệu với cache theo các đơn vị cơ sở như byte, từ và từ kép. Còn cache trao đổi dữ liệu với bộ nhớ chính theo các khối, với kích thước 16, 32 hoặc 64 bytes. Sở dĩ CPU trao đổi dữ liệu với cache theo các đơn vị cơ sở mà không theo khối do dữ liệu được lưu trong các thanh ghi của CPU – vốn có dung lượng rất hạn chế. Vì vậy, CPU chỉ trao đổi các phần tử dữ liệu cần thiết theo yêu cầu của các lệnh. Ngược lại, cache trao đổi dữ liệu với bộ nhớ chính theo các khối, mỗi khối gồm nhiều byte kề nhau với mục đích bao phủ các mẩu dữ liệu lân cận theo không gian và thời gian. Ngoài ra, trao đổi dữ liệu theo khối (hay mẻ) với bộ nhớ chính giúp cache tận dụng tốt hơn băng thông đường truyền và nhờ vậy có thể tăng tốc độ truyền dữ liệu.

4.4.2.4 Các hệ số Hit và Miss

Hit (đoán trúng) là một sự kiện mà CPU truy nhập một mục tin và mục tin ấy có ở trong cache. Xác suất để có một hit gọi là hệ số hit, hoặc H. Dễ thấy hệ số hit H thuộc khoảng (0, 1). Hệ số hit càng cao thì hiệu quả của cache càng cao. Ngược lại, *Miss* (đoán trượt) là một sự kiện mà CPU truy nhập một mục tin và mục tin ấy không có ở trong cache. Xác suất của một miss gọi là hệ số miss, hoặc 1-H. Cũng có thể thấy hệ số miss 1-H thuộc khoảng (0, 1). Hệ số miss càng thấp thì hiệu quả của cache càng cao.

4.4.3 Các dạng kiến trúc cache

Kiến trúc cache đề cập đến việc cache được bố trí vào vị trí nào trong quan hệ với CPU và bộ nhớ chính. Có hai loại kiến trúc cache chính: kiến trúc Look Aside (cache được đặt ngang hàng với bộ nhớ chính) và kiến trúc Look Through (cache được đặt giữa CPU và bộ nhớ chính). Mỗi kiến trúc cache kể trên có ưu điểm và nhược điểm riêng.

4.4.3.1 Kiến trúc Look Aside

Trong kiến trúc Look Aside, cache và bộ nhớ chính cùng được kết nối vào bus hệ thống. Như vậy, cả cache và bộ nhớ chính đều "thấy" chu kỳ bus của CPU tại cùng một thời điểm. Hình 42 minh hoạ kiến trúc cache kiểu Look Aside. Kiến trúc Look Aside có thiết kế đơn giản, dễ thực hiện. Tuy nhiên, các sự kiện hit của kiến trúc này thường chậm do cache kết nối với CPU sử dụng bus hệ thống – thường có tần số làm việc không cao và băng thông hẹp. Bù lại, các sự kiện miss của kiến trúc Look Aside thường nhanh hơn do khi CPU không tìm thấy mục tin trong cache, nó đồng thời tìm mục tin trong bộ nhớ chính tại cùng một chu kỳ xung nhịp.

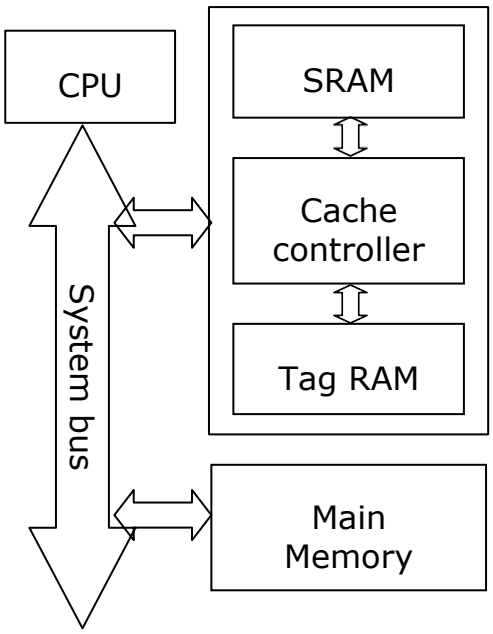
SRAM: RAM lưu dữ liệu cache

Tag RAM: RAM lưu địa chỉ dòng nhớ trong bộ nhớ

Cache controller: bộ điều khiển cache

Main memory: bộ nhớ chính

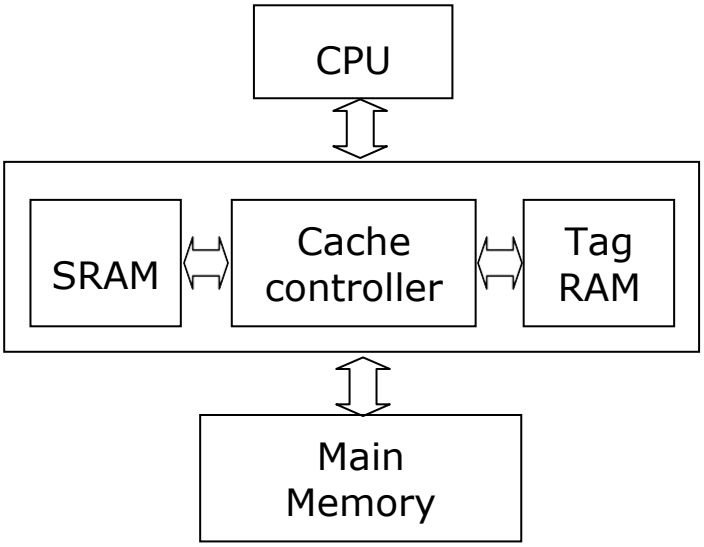
System bus: bus hệ thống



Hình 42 Kiến trúc cache kiểu Look Aside

4.4.3.2 Kiến trúc Look Through

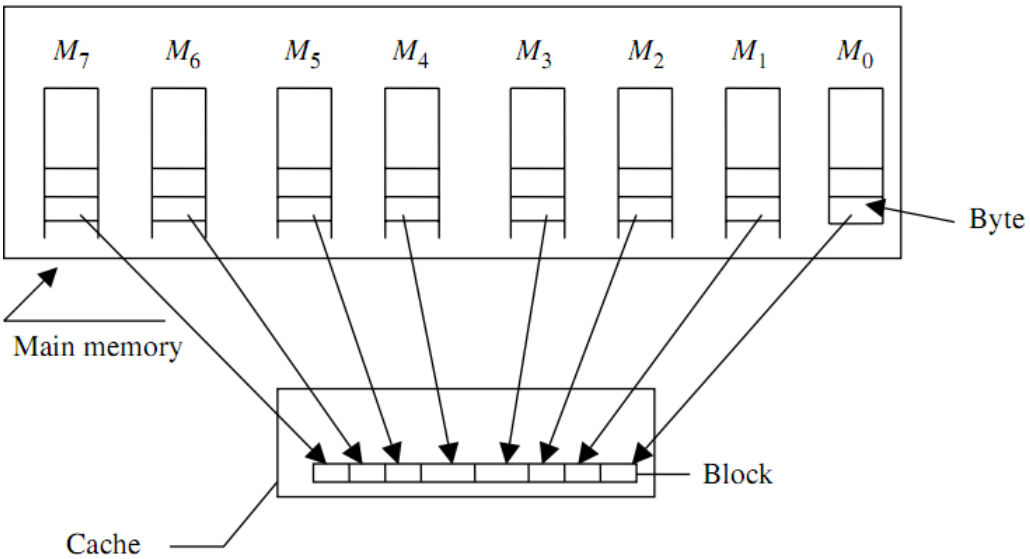
Trong kiến trúc kiểu Look Through, cache được đặt nằm giữa CPU và bộ nhớ chính như minh hoạ trên Hình 43. Như vậy cache có thể "thấy" chu kỳ bus của CPU trước, rồi nó mới "truyền" lại cho bộ nhớ chính. Cache kết nối với CPU bằng hệ thống bus riêng tốc độ cao và băng thông lớn, thường là bus mặt sau (BSB – Back Side Bus). Cache kết nối với bộ nhớ chính thông qua bus hệ thống hay bus mặt trước (FSB – Front Side Bus). FSB thường có tần số làm việc và băng thông thấp hơn nhiều so với BSB. Kiến trúc Look Through phức tạp hơn kiến trúc Look Aside. Ưu điểm chính của kiến trúc này là các sự kiện hit của kiến trúc này thường rất nhanh do CPU kết nối với cache bằng kênh riêng có tốc độ cao. Tuy nhiên, các sự kiện miss của kiến trúc Look Through thường chậm hơn do khi CPU không tìm thấy mục tin trong cache, nó cần tìm mục tin đó trong bộ nhớ chính tại một chu kỳ xung nhịp tiếp theo.



Hình 43 Kiến trúc cache kiểu Look Through

4.4.4 Các dạng tổ chức/ánh xạ cache

4.4.4.1 Giới thiệu tổ chức/ánh xạ cache



Hình 44 Quan hệ giữa các khối của bộ nhớ chính và dòng của cache

Như đã trình bày trong mục 4.2, kích thước của cache thường rất nhỏ so với kích thước bộ nhớ chính. Do vậy, tại mỗi thời điểm, chỉ có một phần nhỏ thông tin của bộ nhớ chính được chuyển vào cache. Câu hỏi đặt ra là, phải xây dựng mô hình tổ chức / ánh xạ trao đổi dữ liệu giữa các phần tử nhớ bộ nhớ chính và các phần tử nhớ của cache như thế nào để hệ thống nhớ đạt được tốc độ truy cập tối ưu.

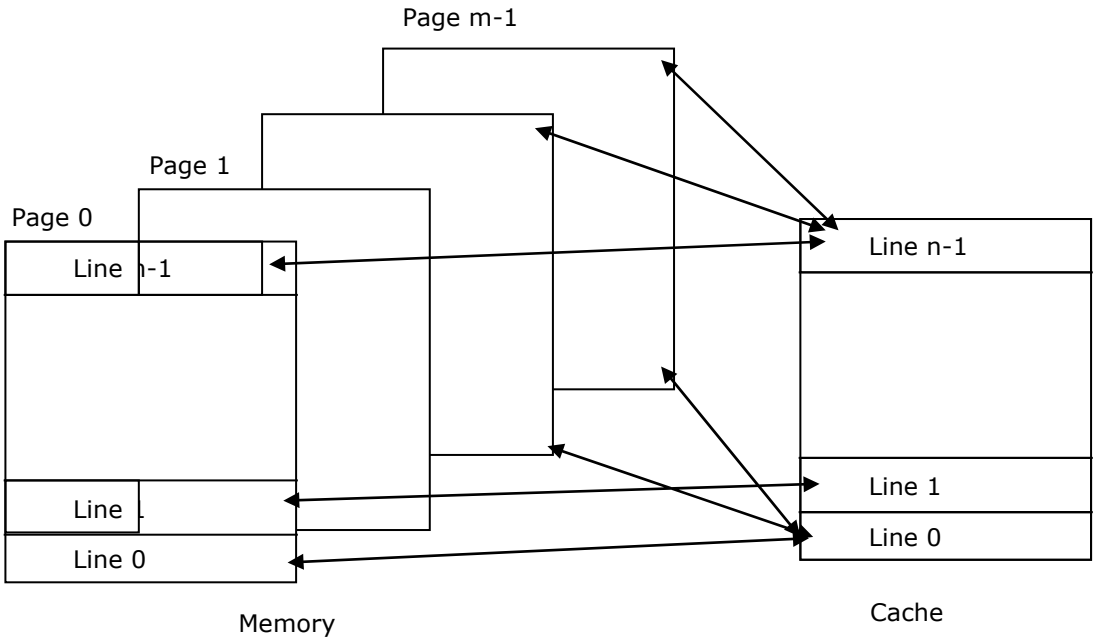
Cho đến hiện nay, có ba phương pháp tổ chức / ánh xạ cache đã được sử dụng, bao gồm: Ánh xạ trực tiếp (Direct mapping), Ánh xạ kết hợp đầy đủ (Fully associative mapping) và Ánh xạ tập kết hợp (Set associative mapping). Phương pháp ánh xạ trực tiếp có ưu điểm là thiết kế đơn giản và nhanh. Tuy nhiên, đây là dạng ánh xạ cố định dễ gây xung đột dẫn đến hiệu quả cache không cao. Phương pháp ánh xạ kết hợp đầy đủ sử dụng ánh xạ mềm, ít gây xung đột và có thể đạt hệ số hit rất cao. Tuy nhiên, phương pháp này có thiết kế phức tạp và có tốc độ

chậm. Phương pháp ánh xạ tập kết hợp là sự kết hợp của hai phương pháp ánh xạ trực tiếp và ánh xạ kết hợp đầy đủ, tận dụng được ưu điểm của cả hai phương pháp: nhanh, ít xung đột và không quá phức tạp, cho hiệu quả cao.

4.4.4.2 Ánh xạ trực tiếp

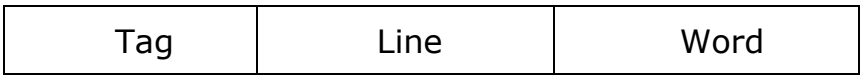
Hình 45 minh hoạ phương pháp ánh xạ trực tiếp bộ nhớ - cache. Cache được chia thành n dòng (line) đánh số từ 0 đến n-1. Bộ nhớ chính được chia thành m trang (page), đánh số từ 0 đến m-1. Mỗi trang nhớ lại được chia thành n dòng (line) đánh số từ 0 đến n-1. Kích thước mỗi trang của bộ nhớ chính bằng kích thước cache và kích thước một dòng trong trang bộ nhớ cũng bằng kích thước một dòng cache. Ánh xạ từ bộ nhớ chính vào cache được thực hiện theo quy tắc sau:

- $Line_0$ của các trang ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_0$ của cache;
- $Line_1$ của các trang ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_1$ của cache;
- ....
- $Line_{n-1}$ của các trang ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_{n-1}$ của cache.



Hình 45 Phương pháp ánh xạ trực tiếp bộ nhớ - cache

Có thể thấy với phương pháp ánh xạ trực tiếp, tại mọi thời điểm luôn có cố định m dòng bộ nhớ cùng cạnh tranh một dòng cache. Khi biết được địa chỉ của dòng trong bộ nhớ, ta biết vị trí của nó trong cache – vì thế phương pháp ánh xạ trực tiếp còn gọi là ánh xạ cứng hay ánh xạ cố định. Để có thể quản lý được các ô nhớ được nạp, cache sử dụng địa chỉ ánh xạ trực tiếp gồm 3 thành phần: *Tag*, *Line* và *Word* như minh hoạ trên Hình 46. *Tag* (bit) là địa chỉ trang trong bộ nhớ chứa dòng được nạp vào cache, *Line* (bit) là địa chỉ dòng trong cache và *Word* (bit) là địa chỉ của từ trong dòng.



Hình 46 Địa chỉ ô nhớ trong ánh xạ trực tiếp

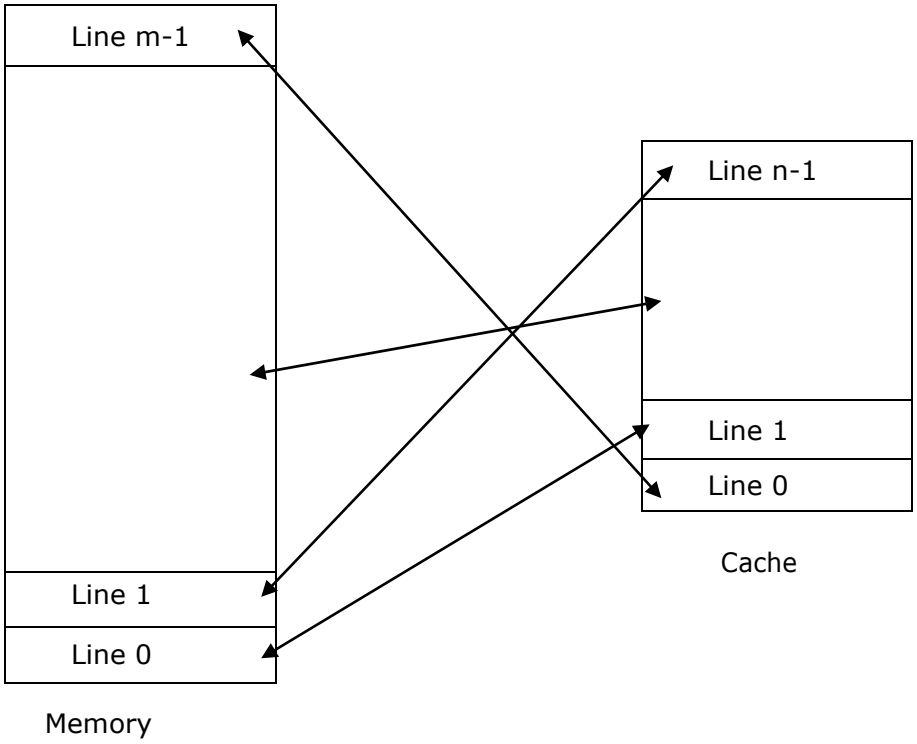
Ví dụ tính các thành phần địa chỉ ô nhớ trong ánh xạ trực tiếp:

- Vào:
  - Dung lượng bộ nhớ = 4GB
  - Dung lượng cache = 1MB
  - Kích thước dòng = 32 byte
- Ra:
  - Kích thước dòng Line = 32 byte = 2^5 , vậy Word = 5 bit
  - Dung lượng Cache = 1MB = 2^{10} $\rightarrow$ có 2^{10} / 2^5 = 2^5 dòng, vậy Line = 5 bit
  - Địa trang Tag: 4GB = 2^{32}, cần 32 bit địa chỉ tổng cộng để địa chỉ hoá các ô nhớ:

    Tag = 32 bit địa chỉ – Line – Word = 32 – 5 – 5 = 22 bit.

Phương pháp ánh xạ trực tiếp có thiết kế đơn giản và rất nhanh do không tốn nhiều thời gian truy tìm địa chỉ ô nhớ trong cache. Do các ánh xạ là cố định, nên khi biết địa chỉ ô nhớ có thể tìm được vị trí của nó trong cache rất nhanh chóng. Tuy nhiên, cũng do ánh xạ cố định nên phương pháp này dễ gây xung đột vì có thể tạo ra nhiều dòng cache bị nút cổ chai trong quá trình hoạt động của cache. Có thể có nhiều dòng cache rảnh rỗi hay ít được sử dụng, nhưng cũng có nhiều dòng cache quá tải do bị nhiều dòng bộ nhớ cùng cạnh tranh. Cũng vì lý do dễ gây xung đột nên hiệu quả tận dụng không gian cache của phương pháp ánh xạ trực tiếp không cao và hệ số hit thấp.

4.4.4.3 Ánh xạ kết hợp đầy đủ

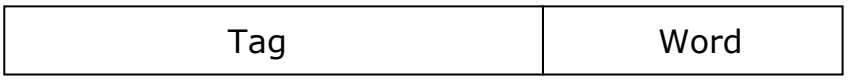


Hình 47 Phương pháp ánh xạ kết hợp đầy đủ bộ nhớ - cache

Phương pháp ánh xạ kết hợp đầy đủ hay còn gọi là ánh xạ liên kết đầy đủ được minh hoạ trên Hình 47. Cache được chia thành n dòng (line) đánh số từ 0 đến n-1. Bộ nhớ chính được chia thành m dòng (line), đánh số từ 0 đến m-1. Kích thước một dòng bộ nhớ bằng kích thước một dòng cache. Do bộ nhớ chính có kích thước lớn hơn nhiều kích thước cache, nên m >> n. Ánh xạ từ bộ nhớ chính vào cache được thực hiện theo quy tắc sau:

- Một dòng trong bộ nhớ chính có thể ánh xạ đến một dòng bất kỳ trong cache, hay
- $Line_i$ (i = 0 ÷ m-1) của bộ nhớ chính ánh xạ đến $Line_j$ (j = 0 ÷ n-1) của cache;

Có thể thấy với phương pháp ánh xạ kết hợp đầy đủ, có n dòng cache để lựa chọn ánh xạ – vì thế phương pháp ánh xạ kết hợp đầy đủ còn gọi là ánh xạ mềm hay ánh xạ không cố định. Ngược lại với phương pháp ánh xạ trực tiếp, khi biết được địa chỉ của dòng trong bộ nhớ, ta chưa biết vị trí của nó trong cache. Để có thể quản lý được các ô nhớ được nạp, cache sử dụng địa chỉ ánh xạ kết hợp đầy đủ chỉ gồm 2 thành phần: *Tag* và *Word* như minh hoạ trên Hình 46. *Tag* (bit) là địa chỉ dòng trong bộ nhớ được nạp vào cache và *Word* (bit) là địa chỉ của từ trong dòng. Phần địa chỉ *Line* như trong địa chỉ ánh xạ trực tiếp bị bỏ do bộ nhớ chính chỉ còn là một trang duy nhất với m dòng.



Hình 48 Địa chỉ ô nhớ trong ánh xạ kết hợp đầy đủ

Ví dụ tính các thành phần địa chỉ ô nhớ trong ánh xạ kết hợp đầy đủ:

- Vào:
  - Dung lượng bộ nhớ = 4GB

- Dung lượng cache = 1MB
- Kích thước dòng = 32 byte

- Ra:
  - Kích thước dòng Line = 32 byte = 2^5 , vậy Word = 5 bit
  - Địa trang Tag: 4GB = 2^{32}, cần 32 bit địa chỉ tổng cộng để địa chỉ hoá các ô nhớ:

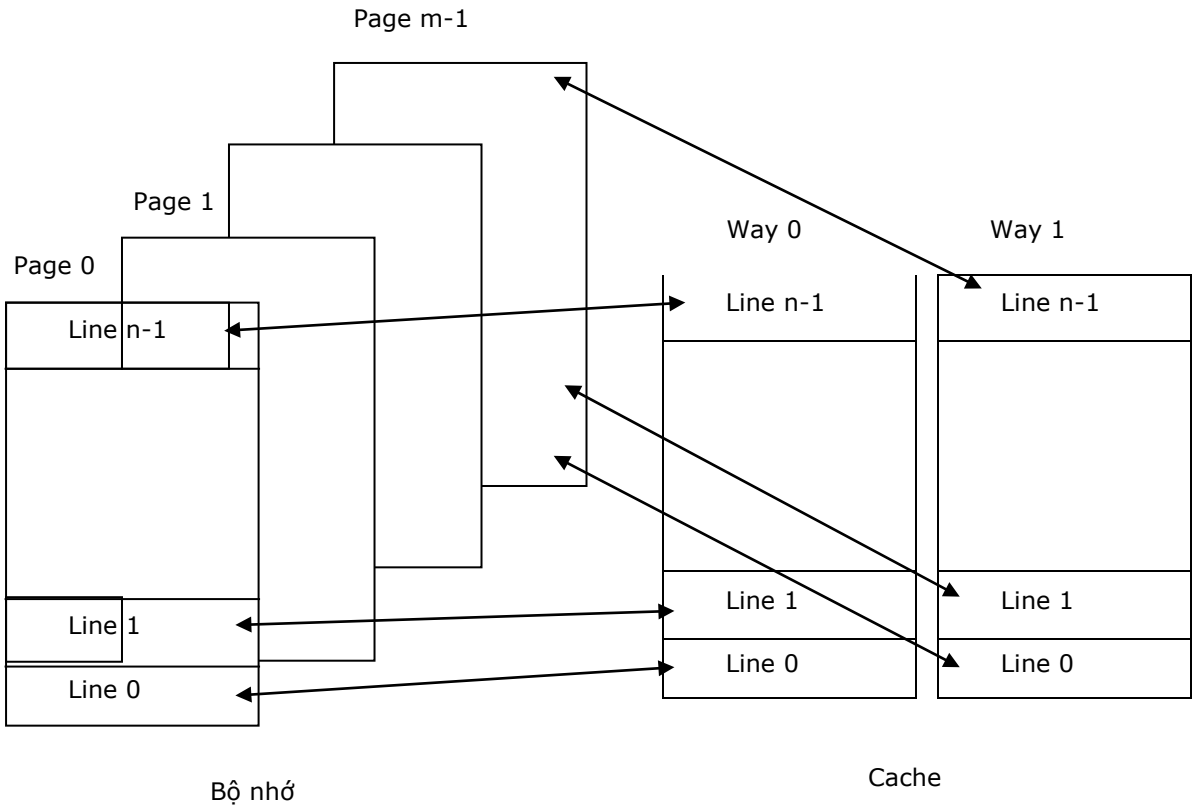
    Tag = 32 bit địa chỉ – Word = 32 – 5 = 27 bit.

Phương pháp ánh xạ kết hợp đầy đủ sử dụng ánh xạ không cố định nên có ưu điểm là mềm dẻo, giảm được xung đột sử dụng dòng cache. Việc sử dụng các dòng cache có thể được điều phối hướng đến phân bố hợp lý hơn, giảm hiện tượng tạo các dòng bị nút cổ chai với mức độ cạnh tranh lớn. Nhờ vậy, phương pháp này có hiệu suất sử dụng không gian cache cao hơn và có khả năng cho hệ số hit cao. Tuy nhiên, cũng do việc sử dụng ánh xạ không cố định, nên việc truy tìm địa dòng nhớ trong cache tốn nhiều thời gian, gây chậm trễ, đặc biệt với các cache có kích thước lớn. Ngoài ra, phương pháp này cũng có thiết kế phức tạp hơn so với phương pháp ánh xạ trực tiếp do cần bổ sung thêm các bộ so sánh địa chỉ dòng cache nhằm tăng tốc cho quá trình truy tìm địa dòng nhớ trong cache. Do vậy, phương pháp ánh xạ kết hợp đầy đủ chỉ thích hợp với các cache có dung lượng nhỏ.

4.4.4.4 Ánh xạ tập kết hợp

Phương pháp ánh xạ tập kết hợp hay còn gọi là ánh xạ liên kết nhóm được minh hoạ trên Hình 49. Cache được chia thành k đường (way) đánh số từ 0 đến k-1. Mỗi đường cache lại được chia thành n dòng (line) đánh số từ 0 đến n-1. Bộ nhớ chính được chia thành m trang (page), đánh số từ 0 đến m-1. Mỗi trang lại được chia thành n dòng (line) đánh số từ 0 đến n-1. Kích thước mỗi trang của bộ nhớ chính bằng kích thước một đường của cache và kích thước một dòng trong trang bộ nhớ cũng bằng kích thước một dòng của đường cache. Ánh xạ từ bộ nhớ chính vào cache được thực hiện theo quy tắc sau:

- Ánh xạ trang bộ nhớ đến đường cache (ánh xạ không cố định):
  - Một trang của bộ nhớ có thể ánh xạ đến một đường bất kỳ của cache.
- Ánh xạ dòng của trang đến dòng của đường (ánh xạ cố định):
  - $Line_0$ của $page_i$ của bộ nhớ ánh xạ đến $Line_0$ của way_j của cache;
  - $Line_1$ của $page_i$ của bộ nhớ ánh xạ đến $Line_1$ của way_j của cache;
  - ....
  - $Line_{n-1}$ của $page_i$ của bộ nhớ ánh xạ đến $Line_{n-1}$ của way_j của cache.



Hình 49 Phương pháp ánh xạ tập kết hợp bộ nhớ - cache

Có thể thấy phương pháp ánh xạ tập kết hợp đảm bảo được sự kết hợp hài hoà giữa ánh xạ mềm từ trang nhớ đến đường cache và ánh xạ cố định từ dòng của trang nhớ đến dòng của đường cache. Do số đường cache không lớn (thường chỉ khoảng 4, 8, 16, 32 hoặc 64 đường) nên việc tìm kiếm địa chỉ trang nhớ trong các đường cache không ảnh hưởng nhiều đến tốc độ truy cập cache. Hơn nữa, do ánh xạ từ dòng của trang nhớ đến dòng của đường cache là cố định, có thể nhanh chóng xác định được vị trí của dòng nhớ trong đường cache khi biết địa chỉ của nó. Để có thể quản lý được các ô nhớ được nạp, cache sử dụng địa chỉ ánh xạ trực tiếp gồm 3 thành phần: *Tag*, *Set* và *Word* như minh hoạ trên Hình 50. *Tag* (bit) là địa chỉ trang trong bộ nhớ chứa dòng được nạp vào cache, *Set* (bit) là địa chỉ dòng trong đường cache và *Word* (bit) là địa chỉ của từ trong dòng.

Tag	Set	Word

Hình 50 Địa chỉ ô nhớ trong ánh xạ tập kết hợp kết hợp

Ví dụ tính các thành phần địa chỉ ô nhớ trong ánh xạ tập kết hợp:

- Vào:
  - Dung lượng bộ nhớ = 4GB
  - Dung lượng cache = 1MB, 2 đường
  - Kích thước dòng = 32 byte
- Ra:

- Kích thước dòng Line = 32 byte = 2^5 , vậy Word = 5 bit
- Dung lượng Cache = 1MB = 2^{10} → có 2^{10} / 2 đường / 2^5 = 2^4 dòng / đường, vậy Set = 4 bit
- Địa trang Tag: 4GB = 2^{32}, cần 32 bit địa chỉ tổng cộng để địa chỉ hoá các ô nhớ:

  Tag = 32 bit địa chỉ – Set – Word = 32 – 4 – 5 = 23 bit.

Phương pháp ánh xạ tập kết hợp tận dụng được ưu điểm của cả hai phương pháp ánh xạ trực tiếp và ánh xạ kết hợp đầy đủ: nhanh do ánh xạ trực tiếp được sử dụng cho ánh xạ dòng - chiếm số lớn ánh xạ và mềm dẻo, ít xung đột do ánh xạ từ các trang bộ nhớ đến các đường cache là không cố định. Nhờ vậy, phân bố sử dụng không gian cache đồng đều hơn và đạt hệ số hit cao hơn. Nhược điểm lớn nhất của phương pháp này là có độ phức tạp thiết kế và điều khiển cao do cache được chia thành một số đường, thay vì chỉ một đường duy nhất.

4.4.5 Các phương pháp đọc ghi và các chính sách thay thế

4.4.5.1 Các phương pháp đọc ghi cache

Việc trao đổi thông tin giữa CPU – cache và giữa cache – bộ nhớ chính là một trong các vấn đề có ảnh hưởng lớn đến hiệu năng cache. Câu hỏi đặt ra là cần có chính sách trao đổi hay đọc ghi thông tin giữa các thành phần này như thế nào để đạt được hệ số hit cao nhất và giảm thiểu miss.

Xét trường hợp đọc thông tin và nếu đó là trường hợp hit (mẩu tin cần đọc có trong cache): mẩu tin được đọc từ cache vào CPU và bộ nhớ chính không tham gia. Như vậy thời gian CPU truy nhập mẩu tin bằng thời gian CPU truy nhập cache. Ngược lại, nếu đọc thông tin và đó là trường hợp miss (mẩu tin cần đọc không có trong cache): mẩu tin trước hết được chuyển từ bộ nhớ chính vào cache, sau đó nó được đọc từ cache vào CPU. Đây là trường hợp xấu nhất: thời gian CPU truy nhập mẩu tin bằng thời gian truy nhập cache cộng với thời gian cache truy nhập bộ nhớ chính – còn gọi là *miss penalty* (gấp đôi thời gian truy cập khi đoán trượt).

Với trường hợp ghi thông tin và nếu đó là trường hợp hit, có thể áp dụng một trong 2 chính sách ghi: *ghi thẳng* (write through) và *ghi trễ* (write back). Với phương pháp ghi thẳng, mẩu tin cần ghi được lưu đồng thời ra cache và bộ nhớ chính. Phương pháp ghi này luôn đảm bảo tính nhất quán dữ liệu giữa cache và bộ nhớ chính, nhưng có thể gây chậm trễ và tốn nhiều băng thông khi tần suất ghi lớn với nhiều mẩu tin có kích thước nhỏ. Ngược lại, với phương pháp ghi trễ, mẩu tin trước hết được ghi ra cache và dòng cache chứa mẩu tin sẽ được ghi ra bộ nhớ chính khi nó bị thay thế. Như vậy, mẩu tin có thể được ghi ra cache nhiều lần, nhưng chỉ được ghi ra bộ nhớ chính một lần duy nhất, giúp tăng tốc độ và giảm băng thông sử dụng. Phương pháp ghi trễ đang được ứng rộng rãi trong các hệ thống cache hiện nay.

Với trường hợp ghi thông tin và nếu đó là trường hợp miss, cũng có thể áp dụng một trong hai chính sách ghi: *ghi có đọc lại* (write allocate / fetch on write) và *ghi không đọc lại* (write non-allocate). Với phương pháp ghi có đọc lại, mẩu tin trước hết được ghi ra bộ nhớ chính, và sau đó dòng nhớ chứa mẩu tin vừa ghi được đọc vào cache. Việc đọc lại mẩu tin vừa ghi từ bộ nhớ chính vào cache có thể giúp giảm miss đọc kế tiếp áp dụng nguyên lý lân cận theo thời gian: mẩu tin vừa được truy nhập có thể được truy nhập lại trong tương lai gần. Với phương pháp ghi không đọc lại, mẩu tin chỉ được ghi ra bộ nhớ chính. Không có thao tác đọc dòng nhớ chứa mẩu tin vừa ghi vào cache.

4.4.5.2 Các chính sách thay thế dòng cache

Như đã đề cập, với cả ba phương pháp ánh xạ bộ nhớ chính – cache, luôn có nhiều dòng nhớ cùng ánh xạ đến một dòng cache. Do có nhiều dòng bộ nhớ chia sẻ một dòng cache, các dòng bộ nhớ được nạp vào cache sử dụng một thời gian và được thay thế bởi dòng nhớ khác theo yêu cầu thông tin phục vụ CPU. Các chính sách thay thế (replacement policies) xác định các dòng cache nào được chọn để thay thế bởi các dòng khác từ bộ nhớ nhằm đạt hệ số hit cao nhất. Có ba chính sách thay thế được sử dụng hiện nay: *thay thế ngẫu nhiên* (Random Replacement), *thay thế kiểu vào trước ra trước* (FIFO – First In First Out) và *thay thế các dòng ít được sử dụng gần đây nhất* (LRU – Least Recently Used).

Thay thế ngẫu nhiên là phương pháp đầu tiên được sử dụng do có thiết kế đơn giản và dễ cài đặt. Các dòng cache được lựa chọn để thay thế một cách ngẫu nhiên, không theo một quy luật nào. Do vậy, phương pháp thay thế ngẫu nhiên thường có hệ số miss cao do phương pháp này không xem xét đến các dòng cache đang thực sự được sử dụng. Nếu một dòng cache đang được sử dụng và bị thay thế sẽ xảy ra miss và nó lại cần được đọc từ bộ nhớ chính vào cache.

Trong phương pháp thay thế kiểu vào trước ra trước, các dòng nhớ được nạp vào cache trước sẽ được chọn để thay thế trước. Phương pháp này luôn có khuynh hướng loại bỏ các dòng cache có thời gian sử dụng lâu nhất, hay "già nhất". Nó có khả năng cho hệ số miss thấp hơn so với thay thế ngẫu nhiên do phương pháp này có xem xét đến yếu tố lân cận theo thời gian – các dòng nhớ có thời gian tồn tại trong cache lâu nhất có thể có xác suất được sử dụng thấp hơn. Tuy nhiên, phương pháp này vẫn chưa thực sự xem xét đến các dòng cache đang thực sự được sử dụng - một dòng cache "già" vẫn có thể đang được sử dụng. Một nhược điểm khác của thay thế kiểu vào trước ra trước là thiết kế và cài đặt phức tạp hơn, do cần phải có mạch mạch điện tử chuyên dụng để theo dõi trật tự nạp các dòng bộ nhớ vào cache.

Phương pháp thay thế các dòng ít được sử dụng gần đây nhất hoạt động theo nguyên tắc: các dòng cache được lựa chọn để thay thế là các dòng ít được sử dụng gần đây nhất. Phương pháp này cho hệ số miss thấp nhất so với thay thế ngẫu nhiên và thay thế FIFO, do thay thế LRU có xem xét đến các dòng đang thực sự được sử dụng – tuân theo yếu tố lân cận theo thời gian một cách chặt chẽ. Nhược điểm duy nhất của phương pháp này là thiết kế và cài đặt phức tạp hơn, do cần phải có mạch điện tử chuyên dụng để theo dõi tần suất sử dụng các dòng cache.

4.4.6 Hiệu năng cache và các yếu tố ảnh hưởng

4.4.6.1 Hiệu năng cache

Hiệu năng của cache (Cache Performance) có thể được đánh giá tổng thể theo thời gian truy nhập trung bình của CPU đến hệ thống nhớ. Thời gian truy nhập trung bình của một hệ thống nhớ có cache được tính như sau:

t_{access} = (Hit cost) + (miss rate) * (miss penalty)

$t_{access} = t_{cache} + (1 - H) * (t_{memory})$

trong đó, t_{access} là thời gian truy nhập trung bình hệ thống nhớ, t_{memory} là thời gian truy nhập bộ nhớ chính, t_{cache} là thời gian ruy nhập cache và H là hệ số hit.

Nếu t_{cache} = 5ns, t_{memory} = 60ns và H=80%, ta có:

$t_{access} = 5 + (1 – 0.8) * (60) = 5+12 = 17ns$

Nếu t_{cache} = 5ns, t_{memory} = 60ns và H=95%, ta có:

t_{access} = 5 + (1 – 0.95) * (60) = 5+3 = 8ns

Như vậy, thời gian truy nhập trung bình hệ thống nhớ tiệm cận thời gian truy nhập cache với trường hợp cache đạt hệ số hit cao.

4.4.6.2 Các yếu tố ảnh hưởng

Có nhiều yếu tố ảnh hưởng đến hiệu năng cache, trong đó ba vấn đề (1) kích thước cache, (2) chia tách cache và (3) tạo cache nhiều mức có ảnh hưởng lớn nhất. Chúng ta lần lượt xem xét ảnh hưởng của từng yếu tố đến hiệu năng cache.

Vấn đề kích thước cache

Vấn đề kích thước cache liên quan đến việc trả lời câu hỏi: nên lựa chọn kích thước cache lớn hay nhỏ? Nhiều số liệu thống kê cho thấy, kích thước cache không ảnh hưởng nhiều đến hệ số miss và hệ số miss của cache lệnh thấp hơn nhiều so với cache dữ liệu:

8KB cache lệnh có hệ số miss nhỏ hơn 1%

256KB cache lệnh có hệ số miss nhỏ hơn 0.002%

như vậy, tăng kích thước cache lệnh không giảm miss hiệu quả.

8KB cache dữ liệu có hệ số miss nhỏ hơn 4%

256KB cache dữ liệu có hệ số miss nhỏ hơn 3%

như vậy, tăng kích thước cache dữ liệu lên 32 lần, hệ số miss giảm 25%.

Trên thực tế, xu hướng chung mong muốn kích thước cache càng lớn trong giới hạn cho phép của giá thành. Với kích thước lớn, có thể tăng được số dòng bộ nhớ lưu trong cache và nhờ vậy giảm tần suất tráo đổi các dòng cache của các chương trình khác nhau với bộ nhớ chính. Đồng thời, cache lớn hỗ trợ đa nhiệm, xử lý song song và các hệ thống CPU nhiều nhân tốt hơn do không gian cache lớn có khả năng chứa đồng thời thông tin của nhiều chương trình. Nhược điểm của cache lớn là chậm, do có không gian tìm kiếm lớn hơn cache nhỏ.

Vấn đề chia tách cache

Cache có thể được tách thành cache lệnh (I-Cache) và cache dữ liệu (D-Cache) để cải thiện hiệu năng, do:

- Dữ liệu và lệnh có tính lân cận khác nhau;
- Dữ liệu thường có tính lân cận về thời gian cao hơn lân cận về không gian; lệnh có tính lân cận về không gian cao hơn lân cận về thời gian;
- Cache lệnh chỉ cần hỗ trợ thao tác đọc; cache dữ liệu cần hỗ trợ cả 2 thao tác đọc và ghi và tách cache giúp tối ưu hoá dễ dàng hơn;
- Tách cache hỗ trợ nhiều lệnh truy nhập đồng thời hệ thống nhớ, nhờ vậy giảm xung đột tài nguyên cho CPU pipeline.

Vấn đề tạo cache nhiều mức

Khi cache được chia thành nhiều mức với kích thước tăng dần và tốc độ truy nhập giảm dần sẽ giúp cải thiện được hiệu năng hệ thống do hệ thống cache nhiều mức có khả năng dung hoà tốt hơn tốc độ của CPU với tốc độ của bộ nhớ chính.

Ví dụ: xem xét 2 hệ thống nhớ có số mức cache khác nhau: hệ thống 3 mức cache (L1, L2 và L3) và hệ thống 1 mức cache (L1). Giả thiết CPU có thời gian truy nhập là 1ns, các mức cache L1, L2, L3 có thời gian truy nhập lần lượt là 5ns, 15ns và 30ns. Bộ nhớ chính có thời gian truy nhập là 60ns.

	CPU	L1	L2	L3	Bộ nhớ chính
Cache 3 mức:	1ns	5ns	15ns	30ns	60ns
Cache 1 mức:	1ns	5ns			60ns

Có thể thấy hệ thống nhớ với nhiều mức cache có khả năng dung hoà tốc độ giữa các thành phần tốt hơn và có thời gian truy nhập trung bình hệ thống nhớ thấp hơn. Trên thực tế, đa số cache được tổ chức thành 2 mức: L1 và L2. Một số cache có 3 mức: L1, L2 và L3. Ngoài ra, nhiều mức cache có thể giúp giảm giá thành hệ thống nhớ.

4.4.7 Các phương pháp giảm miss cho cache

4.4.7.1 Các loại miss của cache

Một hệ thống nhớ với cache tốt cần đạt được các yếu tố: (1) hệ số hit cao, (2) hệ số miss thấp và (3) nếu xảy ra miss thì không quá chậm. Để có thể có giải pháp giảm miss hiệu quả, ta cần phân biệt rõ các loại miss. Cụ thể, tồn tại ba loại miss chính: *miss bắt buộc* (Compulsory misses), *miss do dung lượng* (Capacity misses) và *miss do xung đột* (Conflict misses). Miss bắt buộc thường xảy ra tại thời điểm chương trình được kích hoạt, khi mã chương trình đang được tải vào bộ nhớ và chưa được nạp vao cache. Miss do dung lượng lại thường xảy ra do kích thước của cache hạn chế, đặc biệt trong môi trường đa nhiệm. Do kích thước cache nhỏ nên mã của các chương trình thường xuyên bị tráo đổi giữa bộ nhớ và cache. Theo một khía cạnh khác, miss do xung đột xảy ra khi có nhiều dòng bộ nhớ cùng cạnh tranh một dòng cache.

4.4.7.2 Các phương pháp giảm miss cho cache

Trên cơ sở các loại miss đã được đề cập, hai phương pháp giảm miss có thể phối hợp áp dụng nhằm đạt hiệu quả giảm miss tối đa, gồm: *tăng kích thước dòng cache* và *tăng mức độ liên kết cache*. Biện pháp tăng kích thước dòng cache có thể giúp giảm miss bắt buộc do dòng có kích thước lớn sẽ có khả năng bao phủ các mục tin lân cận tốt hơn. Tuy nhiên, biện pháp này sẽ làm tăng miss xung đột, do dòng kích thước lớn sẽ làm giảm số dòng cache, dẫn đến tăng mức độ cạnh tranh của các dòng nhớ đến một dòng cache. Ngoài ra, dòng kích thước lớn có thể gây lãng phí dung lượng cache do có thể có nhiều phần của dòng cache lớn không bao giờ được sử dụng. Hiện nay, kích thước dòng cache thường dùng hiện nay là 64 bytes.

Biện pháp tăng mức độ liên kết cache hay tăng số đường cache có thể giúp giảm miss xung đột, do tăng số đường cache làm tăng tính mềm dẻo của ánh xạ trang bộ nhớ đến đường cache do có nhiều lựa chọn hơn. Tuy nhiên, nếu tăng số đường cache quá lớn, có thể làm cache chậm do tăng không gian tìm kiếm các đường cache. Hiện nay, số đường cache hợp lý cho miss tối ưu thường dùng là khoảng 8 đường.

4.5 CÂU HỎI ÔN TẬP

1. Hệ thống bộ nhớ phân cấp: đặc điểm, vai trò.
2. ROM là gì? các loại ROM.
3. RAM, SRAM, DRAM là gì? Cấu tạo của SRAM và DRAM.
4. Bộ nhớ cache:
   - Cache là gì? vai trò và nguyên lý hoạt động.
   - Kiến trúc cache
   - Tổ chức/ánh xạ cache
   - Đọc ghi thông tin trong cache
   - Các chính sách thay thế dòng cache
   - Hiệu năng cache và các yếu tố ảnh hưởng
   - Các biện pháp giam miss cho cache.

CHƯƠNG 5 BỘ NHỚ NGOÀI

5.1 ĐĨA TỪ

5.1.1 Giới thiệu

Đĩa từ (Magnetic Disks) là một trong các loại thiết bị lưu trữ được sử dụng rộng rãi nhất trong các thiết bị tính toán nói chung và các máy tính cá nhân nói riêng. Đĩa từ thuộc loại bộ nhớ ổn định – thông tin lưu trên đĩa từ luôn được duy trì, không phụ thuộc vào nguồn điện nuôi bên ngoài. Đĩa từ cũng là bộ nhớ kiểu khối có dung lượng lớn, đặc biệt là các đĩa cứng, dùng để lưu trữ thông tin lâu dài dưới dạng các tệp (files). Để lưu được thông tin, đĩa từ sử dụng các đĩa nhựa hoặc đĩa kim loại có phủ lớp bột từ trên bề mặt. Bột từ được sử dụng thường là oxit sắt hoặc các hợp kim của sắt.

Có hai dạng đĩa từ chủ yếu là đĩa từ mềm (gọi tắt là đĩa mềm – Floppy Disks) và đĩa từ cứng (gọi tắt là đĩa cứng – Hard Disks). Đĩa mềm làm bằng plastic, có dung lượng nhỏ, tốc độ chậm và dễ bị hư hỏng. Người ta sử dụng ổ đĩa mềm (FDD – Floppy Disk Drive) để đọc ghi đĩa mềm. Hình 51 minh hoạ đĩa mềm và ổ đĩa mềm dung lượng 1,44MB với kích thước đĩa 3,5 inches. Ngày nay, do sự phát triển mạnh mẽ của các loại đĩa quang và đặc biệt là các thẻ nhớ flash kết nối qua cổng USB, đĩa mềm ngày càng ít được sử dụng. Nhiều hệ thống máy tính lắp mới không đi kèm ổ đĩa mềm.

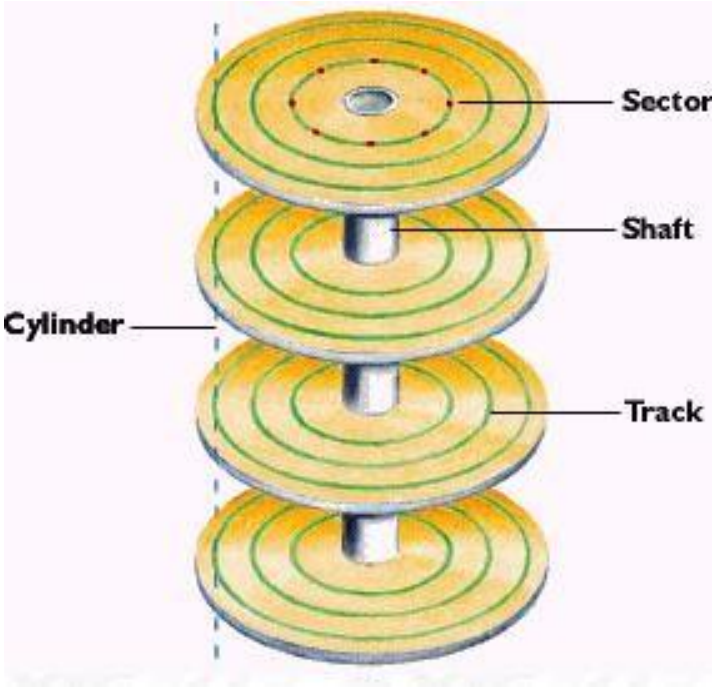
Hình 51 Đĩa mềm và ổ đĩa mềm kích thước 3,5 inches

Khác với đĩa mềm, đĩa cứng thường được gắn cố định trong ổ đĩa và được bọc trong một hộp kim loại bảo vệ như minh hoạ trên hình Hình 52. Đĩa cứng được làm bằng kim loại hoặc bằng thuỷ tinh, có dung lượng lớn và tốc độ cao hơn nhiều lần so với đĩa mềm. Hiện nay, các ổ đĩa cứng thường có dung lượng rất lớn, từ vài chục gigabyte đến hàng ngàn gigabyte và là thiết bị lưu trữ chủ yếu của các hệ thống máy tính. Do đĩa từ mềm ngày càng ít được sử dụng, phần tiếp theo của chương này chỉ đề cập đến đĩa từ cứng và ổ đĩa cứng.

Hình 52 Ổ đĩa cứng kích thước 3,5 inches

5.1.2 Đĩa cứng

5.1.2.1 Cấu tạo đĩa cứng



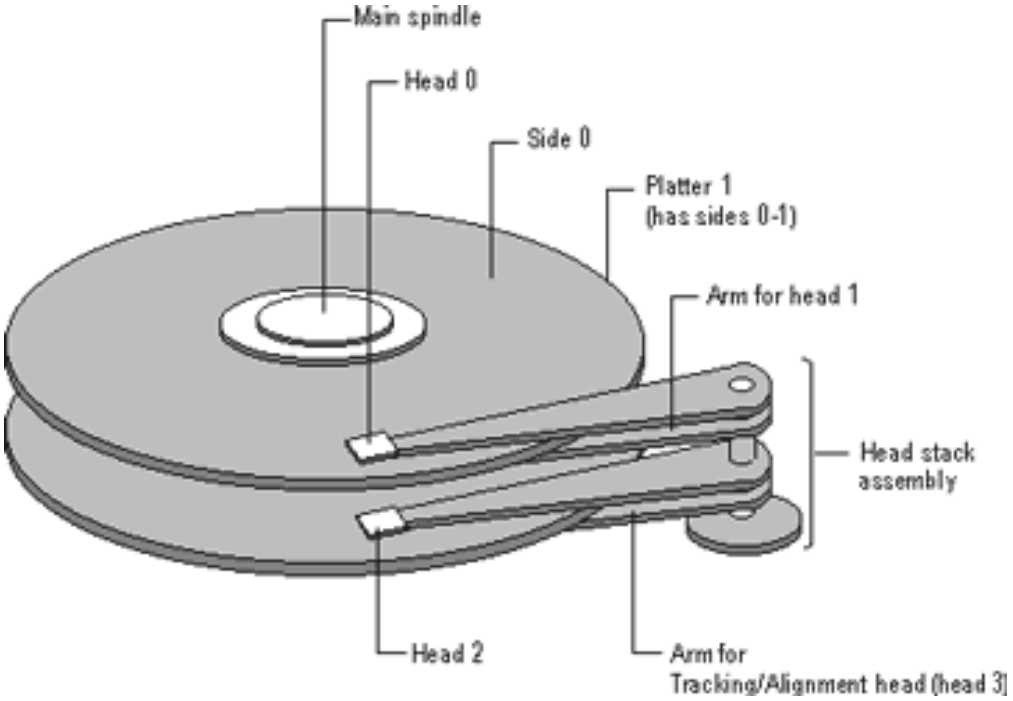
Hình 53 Các thành phần của đĩa cứng

Hình 53 minh hoạ cấu trúc của đĩa cứng và Hình 54 minh hoạ hệ thống đĩa và đầu từ đọc/ghi đĩa cứng. Đĩa cứng thường gồm các thành phần chính: các đĩa từ (Disks), các đầu từ đọc/ghi (Heads), các rãnh (Tracks), các mặt trụ (Cylinders) và các cung (Sectors).

Một ổ đĩa cứng có thể gồm một hoặc nhiều đĩa được lắp đồng trục. Các đĩa thường phẳng và được chế tạo bằng nhôm hoặc thuỷ tinh với lớp bột từ rất mỏng (khoảng 10-20nm) phủ trên bề mặt đĩa để lưu thông tin. Vật liệu từ thường dùng là oxit sắt ba (Fe_2O_3) với các ổ đĩa cứng cũ. Hiện nay vật liệu từ thường dùng là hợp kim của côban. Đĩa có thể lưu thông tin trên cả hai mặt (side), được đánh số mặt 0 và mặt 1.

Đầu từ hay đầu đọc ghi cũng là một trong các bộ phận chủ chốt của ổ đĩa cứng. Mỗi đầu từ đĩa cứng thường có kích thước rất nhỏ, được sử dụng để đọc và ghi thông tin lên đĩa. Khoảng

ách giữa đầu từ và bề mặt đĩa là rất nhỏ, nhưng không tiếp xúc mà "bay" trên mặt đĩa. Mỗi ổ đĩa cứng thường có nhiều đầu từ kết hợp thành một hệ thống đầu từ trên cùng một giá đỡ, như minh hoạ trên Hình 54. Số lượng đầu từ của mỗi ổ đĩa phụ thuộc vào thiết kế và dung lượng đĩa và thường rất khác nhau: 4, 8, 12, 16, 24, 32, 64...



Hình 54 Hệ thống đĩa và đầu từ đọc/ghi đĩa cứng

Rãnh có dạng là một đường tròn đồng tâm trên mặt đĩa để lưu thông tin. Các rãnh được đánh số từ 0 theo trật từ từ phía ngoài đĩa vào trong tâm và mỗi mặt đĩa có thể chứa hàng ngàn rãnh. Tiếp theo rãnh, mặt trụ là tập hợp của các rãnh ở các mặt đĩa khác nhau nằm trên cùng một vị trí đầu từ. Trên thực tế, mặt trụ là tham số được sử dụng nhiều hơn rãnh trong các hệ thống đĩa cứng.

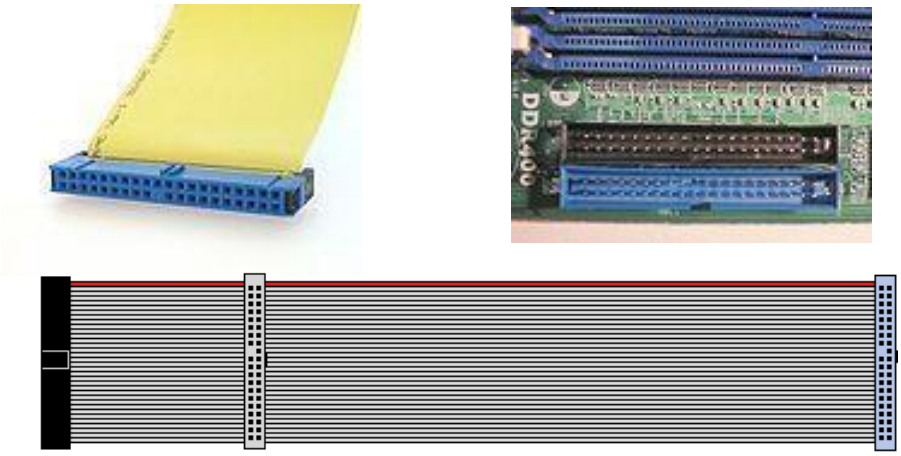
Cung là một phần của rãnh trên bề mặt đĩa và là đơn vị lưu trữ nhỏ nhất có thể quản lý của đĩa. Kích thước thông dụng của mỗi cung là 512 bytes. Với ổ đĩa cứng, ba tham số được sử dụng để tính dung lượng đĩa là: Số lượng mặt trụ (C), số lượng đầu từ (H) và số lượng cung trong một rãnh (S). Như vậy, dung lượng của đĩa cứng tính theo các tham số trên là:

Dung lượng của đĩa cứng = C x H x S x 512 bytes

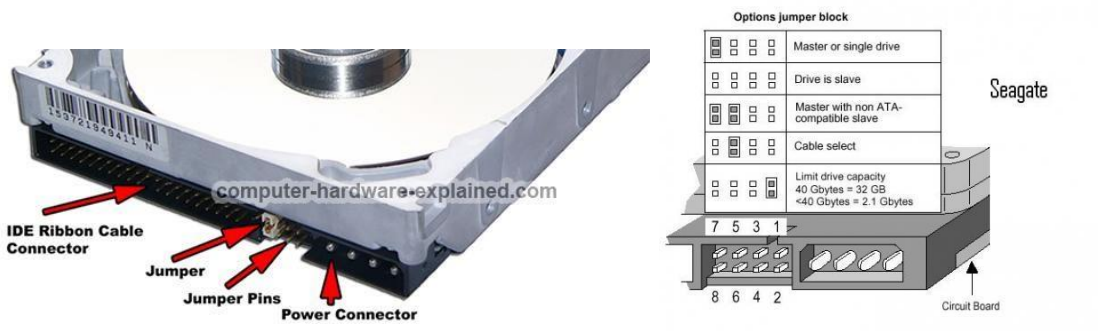
5.1.2.2 Các chuẩn ghép nối đĩa cứng

Các chuẩn hay giao diện ghép nối ổ đĩa cứng giải quyết vấn đề các ổ đĩa cứng được ghép nối và trao đổi dữ liệu với CPU như thế nào. Cho đến hiện nay, các giao diện thông dụng ghép nối ổ đĩa cứng với máy tính gồm: (1) Parallel ATA (PATA - Parallel Advanced Technology Attachments), còn gọi là ATA/IDE/EIDE (Integrated Drive Electronics), (2) Serial ATA (SATA), (3) SCSI – Small Computer System Interface (phát âm là scuzzy /skʌzi/), (4) Serial Attached SCSI (SAS) và (5) iSCSI – Internet SCSI. Trong tài liệu này, ta đề cập chi tiết ba chuẩn ghép nối thông dụng nhất cho máy tính là PATA/ATA/IDE, SATA và SCSI.

Chuẩn ghép nối ATA/IDE/PATA



Hình 55 Giao diện ghép nối và cáp ATA/IDE/PATA



Hình 56 IDE HDD jumpers & cài đặt jumpers

Chuẩn ghép nối ATA/IDE/PATA sử dụng cáp dẹt 40 hoặc 80 sợi để ghép nối ổ cứng với bảng mạch chính của máy tính. Mỗi cáp thường hỗ trợ ghép nối với 2 ổ đĩa: một ổ đĩa chủ (master) và một ổ đĩa tớ (slave). Băng thông đường truyền là 16 bit, đạt các mức thông lượng theo tần số làm việc: 16, 33, 66, 100 và 133MB/s. Hình 55 và Hình 56 minh hoạ khe cắm, cáp ghép nối và các chuyển mạch chế độ làm việc (jumpers) của ổ đĩa chuẩn ATA/IDE/PATA.

Chuẩn ghép nối SATA

Khe cắm dữ liệu SATA | Đầu cắm dữ liệu SATA | Đầu cắm nguồn SATA

Hình 57 Khe cắm và cáp ghép nối SATA

Hình 57 minh hoạ các thành phần ghép nối ổ đĩa cứng với bảng mạch chính theo chuẩn SATA. Chuẩn SATA sử dụng cùng tập lệnh mức thấp như chuẩn ATA nhưng SATA sử dụng đường truyền tin nối tiếp tốc độ cao qua 2 đôi dây với bộ điều khiển SATA sử dụng chuẩn AHCI (Advanced Host Controller Interface). SATA hỗ trợ nhiều tính năng tiên tiến vượt trội so với ATA, như truyền dữ liệu nhanh và hiệu quả hơn và đặc biệt là tính năng cắm nóng (hot plug). SATA cung cấp tốc độ truyền dữ liệu cao hơn nhiều so với ATA. Với SATA thế hệ 1, tốc độ đạt 1,5 Gb/s và lần lượt đạt 3,0 Gb/s và 6,0 Gb/s với các thế hệ 2 và thế hệ 3.

Chuẩn ghép nối SCSI

SCSI là một tập các chuẩn về kết nối vật lý và truyền dữ liệu giữa máy tính và thiết bị ngoại vi, thường được sử dụng trong các máy chủ. Tất cả các thiết bị SCSI đều kết nối đến bus SCSI theo cùng một kiểu và mỗi bus SCSI có thể kết nối 8-16 thiết bị SCSI. Tương tự SATA, chuẩn SCSI cũng cung cấp nhiều tính năng tiên tiến như tốc độ truyền dữ liệu và tính ổn định rất cao và tính năng cắm nóng. Tính năng cắm nóng rất hữu dụng trong các máy chủ do SCSI cho phép thêm, bớt các ổ cứng mà không phải tắt máy, giảm thời gian ngừng cung cấp dịch vụ. SCSI đạt được tốc độ truyền dữ liệu: 5, 10, 20, 40MB/s với các ổ SCSI cũ và 160, 320, 640 MB/s với các ổ SCSI mới. Các ổ cứng SCSI thường rất đắt tiền và được thường được sử dụng cho các máy chủ và các hệ thống lưu trữ tiên tiến như RAID, NAS và SAN.

5.1.2.3 Quản lý đĩa cứng

Các đĩa cứng được quản lý theo hai mức: mức thấp (lower level) và mức cao (high level). Quản lý đĩa ở mức thấp được thực hiện bởi các chức năng của ROM-BIOS, đĩa được quản lý ở mức cao bởi hệ điều hành. Các vấn đề liên quan đến quản lý đĩa cứng gồm: định dạng đĩa cứng, phân khu và bảng phân khu đĩa cứng, cung khởi động, hệ thống file và thư mục gốc.

Định dạng đĩa cứng

Đĩa cứng cần được định dạng (format) trước khi sử dụng. Có hai mức định dạng đĩa cứng: định dạng mức thấp (lower level format) và định dạng mức cao (high level format). Định dạng mức thấp là quá trình gán địa chỉ cho các cung vật lý trên đĩa và có thể được thực hiện bởi các chức năng của BIOS. Hiện nay, hầu hết các ổ đĩa cứng đều đã được định dạng mức thấp khi xuất xưởng. Sau khi được định dạng mức thấp, ổ đĩa cần được định dạng ở mức cao bởi hệ điều hành trước khi có thể lưu thông tin. Định dạng mức cao là quá trình gán địa chỉ cho các cung logic và khởi tạo hệ thống file.

Phân khu và bảng phân khu đĩa cứng

Một đĩa cứng vật lý có thể được chia thành nhiều phần để thuận tiện cho quản lý và lưu trữ. Mỗi phần được gọi là một phân đoạn hay một phân khu (partition). Có hai loại phân khu: phân khu chính (primary partition) và phân khu mở rộng (extended partition). Thông thường, mỗi ổ đĩa chỉ có thể có một phân khu chính và một hoặc một số phân khu mở rộng. Một phân khu lại có thể được chia thành một hoặc một số ổ đĩa logic. Phân khu chính chỉ có thể chứa duy nhất một ổ đĩa logic, nhưng phân khu mở rộng có thể được chia thành một hoặc một số ổ đĩa logic.

Bảng phân khu (partition table) là một bảng gồm các bản ghi lưu thông tin quản lý các phân khu đĩa cứng. Các thông tin cụ thể về mỗi phân khu như sau:

- Phân khu có thuộc loại tích cực (active) ?
- Số mặt trụ (C), đầu từ (H) và cung (S) điểm bắt đầu phân khu;
- Số mặt trụ (C), đầu từ (H) và cung (S) điểm kết thúc phân khu;
- Kiểu định dạng phân khu (FAT, NTFS, EXT);
- Kích thước của phân khu tính theo số cung.

Cung khởi động

Cung khởi động (boot sector) là một cung đặc biệt, luôn nằm ở vị trí cung số 1 của ổ đĩa logic. Cung khởi động chứa chương trình mồi khởi động (Bootstrap loader) có nhiệm vụ kích hoạt việc nạp các thành phần của hệ điều hành từ đĩa vào bộ nhớ.

Hệ thống file

Hệ thống file (file system) là một dạng bảng danh mục (directory) để quản lý việc lưu trữ các files trên đĩa. Các files thường được lưu trữ trong các thư mục (folders) và các thư mục được tổ chức theo mô hình cây. Hệ thống file là một thành phần của hệ điều hành và có thiết kế khác khác nhau. Sau đây là một số hệ thống file thông dụng kèm theo hệ điều hành:

- FAT (DOS, Windows 3.x, Windows 95, 98, ME)
- NTFS (Windows NT, 2000, XP, 2003, Vista, 7)
- Ext2, Ext3 (Unix, Linux)
- MFS (Macintosh FS)/HFS (Hierarchical FS) (Mac OS)

Thư mục gốc

Thư mục gốc (Root directory) là thư mục ở mức thấp nhất trong hệ thống cây thư mục của ổ đĩa logic. Thư mục gốc là điểm bắt đầu khi hệ thống tìm kiếm và truy nhập file. Cũng như các thư mục khác, thư mục gốc có thể chứa các thư mục con và các file. Điểm khác biệt của thư mục gốc với các thư mục khác là nó không có thư mục mẹ.

5.2 ĐĨA QUANG

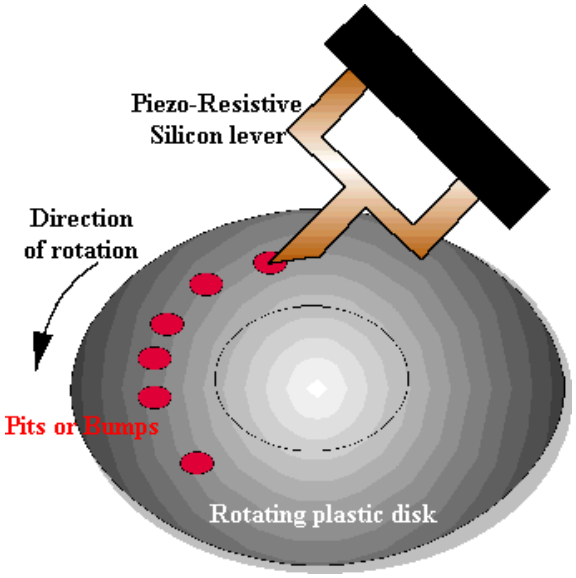
5.2.1 Giới thiệu và nguyên lý

Đĩa quang (Optical Disks) hoạt động dựa trên nguyên lý quang học: sử dụng ánh sáng để đọc và ghi thông tin trên đĩa. Các đĩa quang thường được chế tạo bằng plastic với một mặt được tráng một lớp nhôm mỏng để phản xạ tia laser. Mặt đĩa quang được "khắc" rãnh và mức lõm của rãnh được sử dụng để biểu diễn các bit thông tin, như minh hoạ trên Hình 58. Trên thực tế, các đĩa quang âm nhạc và phim được chế tạo hàng loạt theo kiểu chế bản in gồm 2 khâu: Trước hết, tạo bản đĩa chủ chứa thông tin ở dạng "âm bản" bằng thiết bị chuyên dụng, sau đó sử dụng bản đĩa chủ để "in" thông tin lên các đĩa quang trắng.

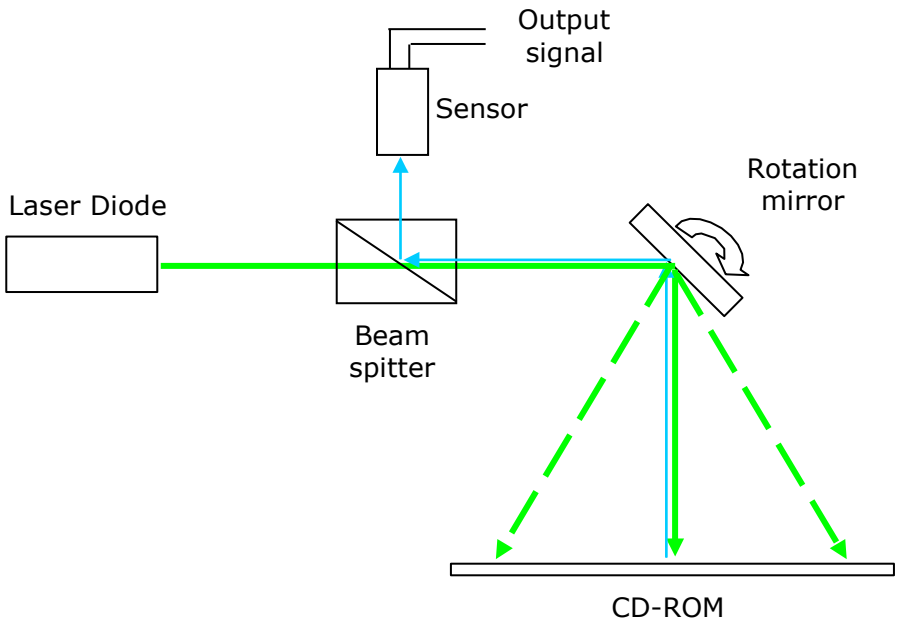
Việc đọc thông tin trên đĩa quang được thực hiện trong ổ đĩa quang (Optical Disk Drive), như minh hoạ trên Hình 59 theo các bước:

1. Tia laser từ điốt phát laser đi qua bộ tách tia đến gương quay;
2. Gương quay được điều khiển bởi tín hiệu đọc, lái tia laser đến vị trí cần đọc trên mặt đĩa;
3. Tia phản xạ từ mặt đĩa phản ánh mức lồi lõm trên mặt đĩa quay trở lại gương quay;

4. Gương quay chuyển tia phản xạ về bộ tách tia và sau đó đến bộ cảm biến quang điện;
5. Bộ cảm biến quangđiện chuyển đổi tia laser phản xạ thành tín hiệu điện đầu ra. Cường độ của tia laser được biểu diễn thành mức tín hiệu ra.



Hình 58 Lưu thông tin trên đĩa quang



Hình 59 Nguyên lý đọc thông tin trên đĩa CD-ROM

5.2.2 Các loại đĩa quang

Có hai họ đĩa quang chính: đĩa CD (Compact Disk) và đĩa DVD (Digital Video Disk). Đĩa CD ra đời trước có dung lượng nhỏ, tốc độ chậm, thường được sử dụng để lưu dữ liệu, âm thanh và phim ảnh có chất lượng thấp. Đĩa DVD ra đời sau, có dung lượng lớn, tốc độ truy nhập caovà cho phép lưu dữ liệu, âm thanh và phim ảnh có chất lượng caohơn.

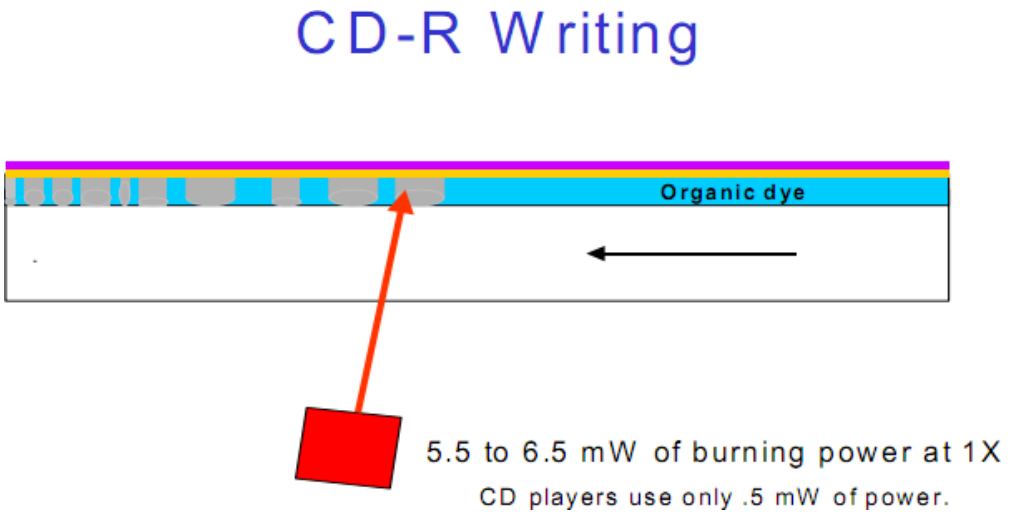
Họ đĩa CD gồm 3 loại chính: đĩa CD chỉ đọc (CD-ROM - Read Only CD), đĩa CD có thể ghi 1 lần (CD-R - Recordable CD) và đĩa CD có thể ghi lại (CD-RW - Rewritable CD). Đĩa CD-ROM được ghi sẵn nội dung từ khi sản xuất và chỉ có thể đọc ra trong quá trình sử dụng. CD-ROM thường được sử dụng để lưu âm nhạc và các phần mềm. Đĩa CD-R là đĩa có thể ghi một lần duy nhất bởi người sử dụng. Sau khi thông tin được ghi, đĩa trở thành loại chỉ đọc. Ngược lại, đĩa CD-RW cho phép xoá thông tin đã ghi và ghi lại nhiều lần. Đĩa CD-RW thường có giá thành cao và có thể ghi lại khoảng 1000 lần.

Tương tự họ CD, họ DVD cũng gồm nhiều loại: đĩa DVD chỉ đọc (DVD-ROM - Read Only DVD), đĩa có thể ghi 1 lần (DVD-R - Recordable DVD), đĩa có thể ghi lại (DVD-RW - Rewritable DVD), đĩa DVD mật độ cao (HD-DVD - High-density DVD) và đĩa DVD mật độ siêu cao (Blu-ray DVD - Ultra-high density DVD). DVD-ROM thường được sử dụng để lưu phim ảnh và các phần mềm có dung lượng lớn. Đĩa DVD-R là đĩa có thể ghi một lần duy nhất bởi người sử dụng. Sau khi thông tin được ghi, đĩa trở thành loại chỉ đọc. Ngược lại, đĩa DVD-RW cho phép xoá thông tin đã ghi và ghi lại nhiều lần. Đĩa HD-DVD và Blu-ray DVD là các loại đĩa DVD có dung lượng siêu cao với dung lượng tương ứng vào khoảng 15GB và 25GB với đĩa một lớp.

5.2.3 Giới thiệu cấu tạo một số đĩa quang thông dụng

5.2.3.1 Đĩa CD-ROM, CD-R và CD-RW

Dung lượng tối đa của đĩa CD là 700MB hoặc 80 phút nếu lưu âm thanh. Ổ đĩa sử dụng tia laser hồng ngoại với bước sóng 780 nm để đọc thông tin. Tốc độ truyền thông tin của đĩa CD được tính theo tốc độ cơ sở (150KB/s) nhân với hệ số nhân. Ví dụ, đĩa có tốc độ đọc 4x thì tốc độ tối đa có thể đọc là 4 x 150KB/s = 600 KB/s; nếu đĩa có tốc độ đọc 50x thì tốc độ tối đa có thể đọc là 50 x 150KB/s = 7500 KB/s.



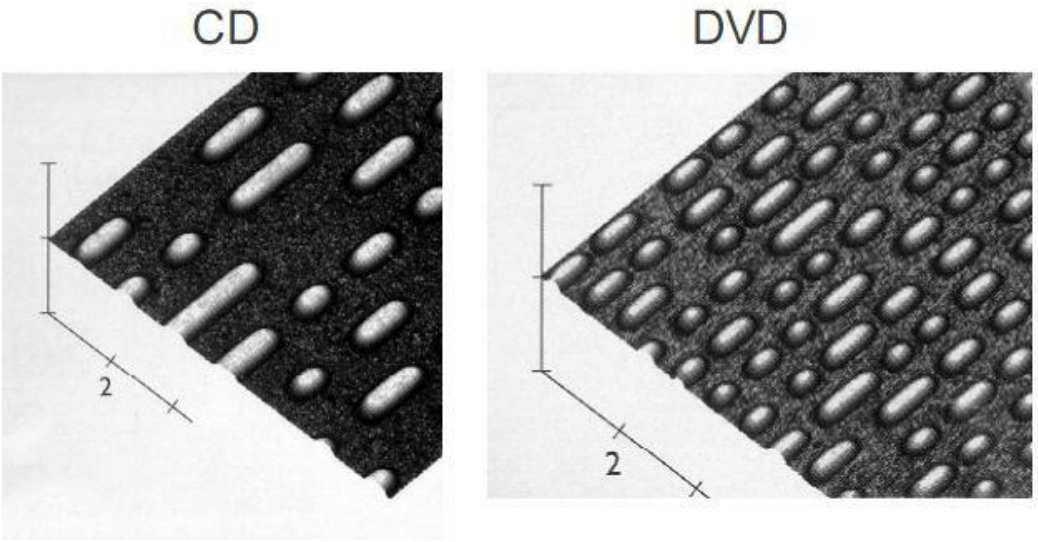
Hình 60 Cấu tạo đĩa CD-R

Đĩa CD-R về mặt hình thức và cấu tạo tương tự đĩa CD-ROM. Tuy nhiên, đĩa CD-R có thêm một lớp gọi là "organic dye", tạm dịch là *lớp hữu cơ* nằm giữa lớp plastic và lớp phản xạ bằng kim loại. Tia laser đã được điều chế bởi tín hiệu ghi được sử dụng để "đốt" lớp hữu cơ tạo thành các mức lồi lõm khác nhau trên lớp này để lưu thông tin. Sau khi đốt lớp hữu cơ bị cố

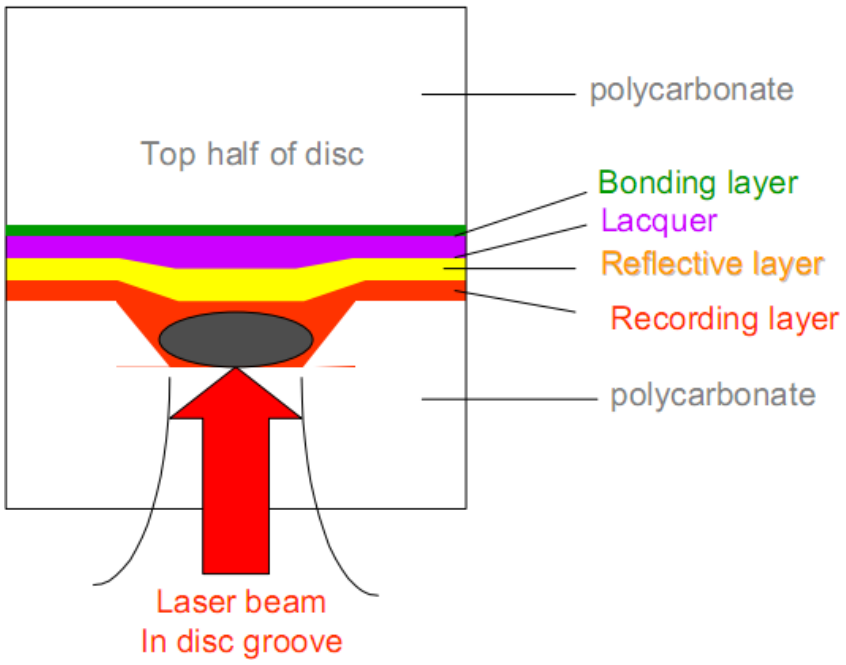
định và do vậy đĩa CD-R chỉ ghi được 1 lần. Trong đĩa CD-RW, lớp hữu cơ được thay bằng một lớp bán kim loại. Nhờ vậy, đĩa CD-RW có thể ghi được nhiều lần. Đa số các đĩa CD-RW cho phép ghi lại đến khoảng 1000 lần.

5.2.3.2 Đĩa DVD-ROM, DVD-R và DVD-RW

Dung lượng tối đa của đĩa DVD là 4,7GB với đĩa một mặt và 8,5GB với đĩa 2 mặt. Ổ đĩa DVD sử dụng tia laser hồng ngoại có bước sóng 650nm, ngắn hơn nhiều so với bước sóng tia laser dùng trong ổ đĩa CD. Nhờ sử dụng bước sóng laser ngắn hơn, đĩa DVD có mật độ ghi cao hơn nhiều so với CD, nhưng minh hoạ trên Hình 61. Tốc độ truyền thông tin của đĩa DVD được tính theo tốc độ cơ sở (1350KB/s) nhân với hệ số nhân. Ví dụ, đĩa có tốc độ đọc 4x thì tốc độ tối đa có thể đọc là 4 x 1350KB/s = 5400 KB/s; nếu đĩa có tốc độ đọc 16x thì tốc độ tối đa có thể đọc là 16 x 1350KB/s = 21600 KB/s.

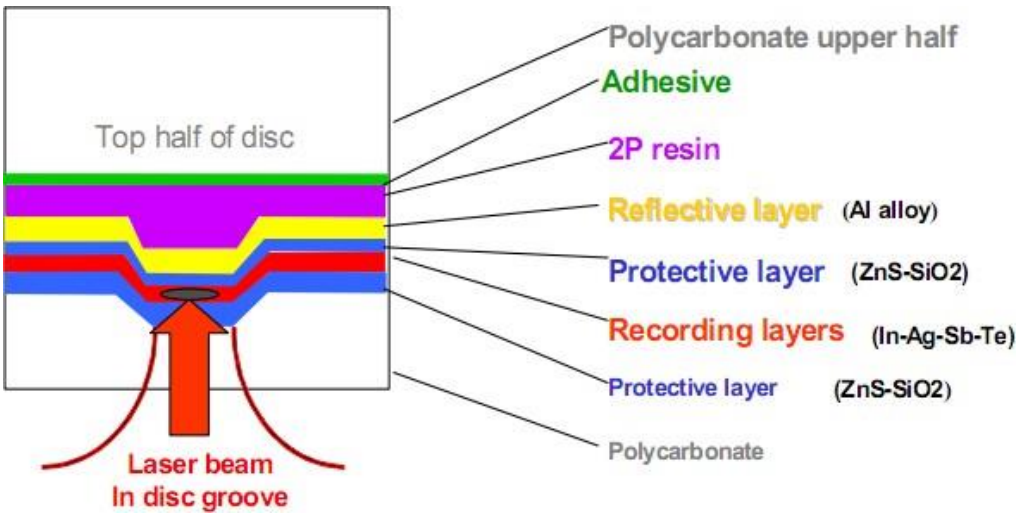


Hình 61 Mật độ ghi thông tin trên đĩa CD và DVD



Hình 62 Cấu tạo đĩa DVD-R

Đĩa DVD-R có cấu tạo tương tự đĩa CD-R, nhưng sử dụng tia laser có bước sóng ngắn hơn, là 650nm, như minh hoạ trên Hình 62. Hình 63 minh hoạ mặt cắt các lớp trong đĩa DVD-RW. Lớp bán kim loại để ghi thông tin được đặt trong hai lớp bảo vệ.



Hình 63 Cấu tạo đĩa DVD-RW

5.2.3.3 Đĩa HD-DVD và Blu-ray DVD

Đĩa HD-DVD và Blu-ray DVD là các "siêu" đĩa quang với dung lượng rất lớn và tốc độ truy nhập cao. Đĩa HD-DVD do Toshiba phát minh, sử dụng tia laser xanh với bước sóng rất ngắn. Đĩa HD-DVD đạt dung lượng 15GB cho một lớp và 30GB cho hai lớp. Do thất bại trong cạnh tranh với đĩa Blu-ray DVD, nên đĩa HD-DVD đã phải ngừng sản xuất từ tháng 2 năm 2008. Đĩa Blu-ray DVD do Sony phát minh, sử dụng tia laser với bước sóng 405nm. Đĩa Blu-ray DVD đạt dung lượng 30GB cho một lớp và 50GB cho hai lớp.

5.3 RAID

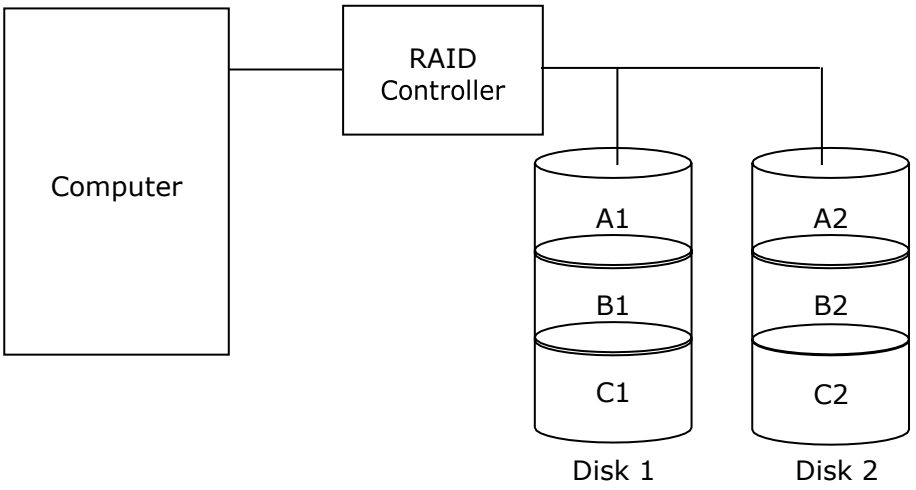
5.3.1 Giới thiệu RAID

RAID (Redundant Array of Independent Disks) là một công nghệ tạo các thiết bị lưu trữ tiên tiến trên cơ sở các ổ đĩa cứng, nhằm đạt các yêu cầu về tốc độ cao (high performance / speed), tính tin cậy cao (high reliability) và dung lượng lớn (large volume). Mặc dù RAID là một mảng của các ổ đĩa cứng, nhưng không phải tất cả các loại ổ cứng đều có thể sử dụng để tạo RAID. Trên thực tế, chỉ có các ổ cứng theo chuẩn SATA, SCSI và tương đương mới hỗ trợ tạo RAID.

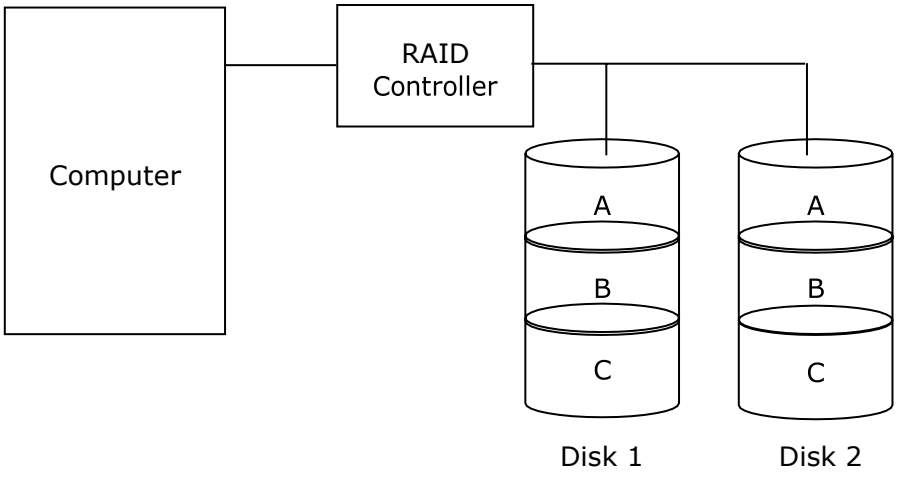
5.3.2 Các kỹ thuật tạo RAID

Có hai kỹ thuật chính được sử dụng để tạo RAID: kỹ thuật tạo lát đĩa (Disk Stripping) và kỹ thuật soi gương đĩa (Disk Mirroring). Hình 64 minh hoạ kỹ thuật tạo lát đĩa. Điểm mấu chốt của kỹ thuật này là điều khiển RAID cung cấp khả năng ghi và đọc song song các khối của cùng một đơn vị dữ liệu. Nhờ vậy tăng được tốc độ đọc ghi. Theo đó, các dữ liệu cần ghi được chia thành các khối cùng kích thước và được ghi đồng thời vào các ổ đĩa vật lý độc lập. Tương tự, trong quá trình đọc, các khối của dữ liệu cần đọc được đọc đồng thời từ các đĩa cứng độc lập, giúp giảm thời gian đọc.

Trong khi kỹ thuật tạo lát đĩa hướng đến tốc độ cao, kỹ thuật soi gương đĩa nhằm đạt độ tin cậy cao cho hệ thống lưu trữ. Hình 65 minh hoạ kỹ thuật soi gương đĩa. Theo đó, dữ liệu cũng được chia thành các khối và mỗi khối được ghi đồng thời lên hai hay nhiều ổ đĩa độc lập. Như vậy, tại mọi thời điểm ta đều có nhiều bản sao dữ liệu trên các đĩa cứng độc lập, đảm bảo tính an toàn cao.



Hình 64 RAID - Kỹ thuật tạo lát đĩa



Hình 65 RAID – Kỹ thuật soi gương đĩa

5.3.3 Giới thiệu một số loại RAID thông dụng

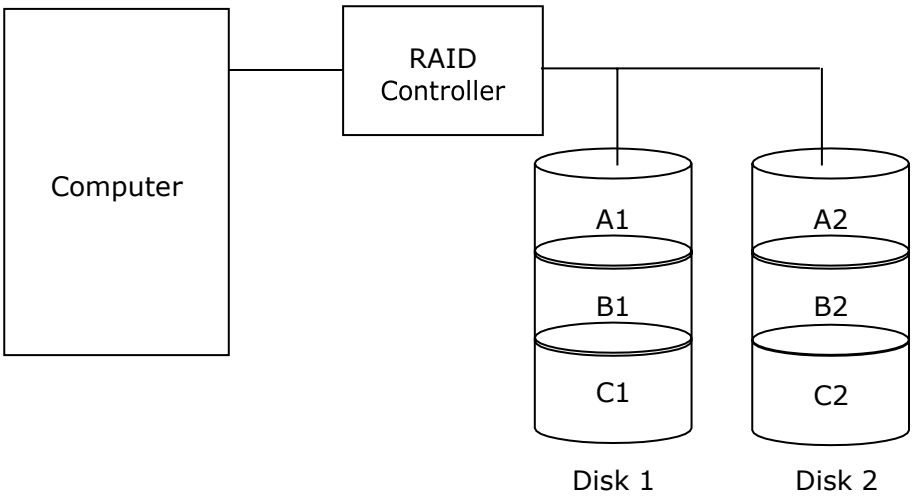
Trên cơ sở các kỹ thuật RAID được đề cập tại mục 5.3.2, trong mục này giới thiệu ba dạng RAID được sử dụng phổ biến nhất: RAID 0, RAID 1 và RAID 10. Các dạng RAID khác chẳng hạn RAID 2, 3, 4, 5, 6, 01, độc giả có thể tham khảo ở các tài liệu khác.

5.3.3.1 RAID 0

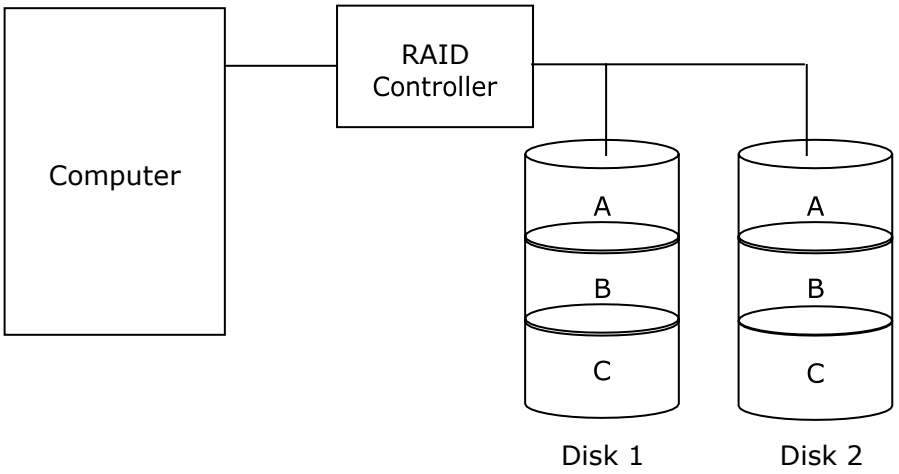
Cấu hình RAID 0 dựa trên kỹ thuật tạo lát đĩa và cần tối thiểu hai ổ đĩa vật lý, như minh hoạ trên Hình 66. Ưu điểm chính của RAID 0 là đạt tốc độ cao – tốc độ truy nhập RAID tỷ lệ thuận với số lượng đĩa độc lập của RAID. Ngoài ra, RAID 0 có thể giúp tăng dung lượng: dung lượng RAID 0 bằng tổng dung lượng của các đĩa độc lập tham gia. Hạn chế lớn nhất của RAID 0 là tính tin cậy – tính tin cậy của RAID 0 chỉ tương đương tính tin cậy của một ổ đĩa đơn.

5.3.3.2 RAID 1

Khác với RAID 0, cấu hình RAID 1 dựa trên kỹ thuật soi gương đĩa và cũng cần tối thiểu hai ổ đĩa vật lý, như minh hoạ trên Hình 67. Ưu điểm chính của RAID 1 là đạt độ tin cậy cao, dotại mỗi thời điểm luôn có nhiều bản sao lưu dữ liệu trên các đĩa độc lập. Tốc độ truy nhập vàdung lượng của RAID 1 đều tương đương với một ổ đĩa đơn.



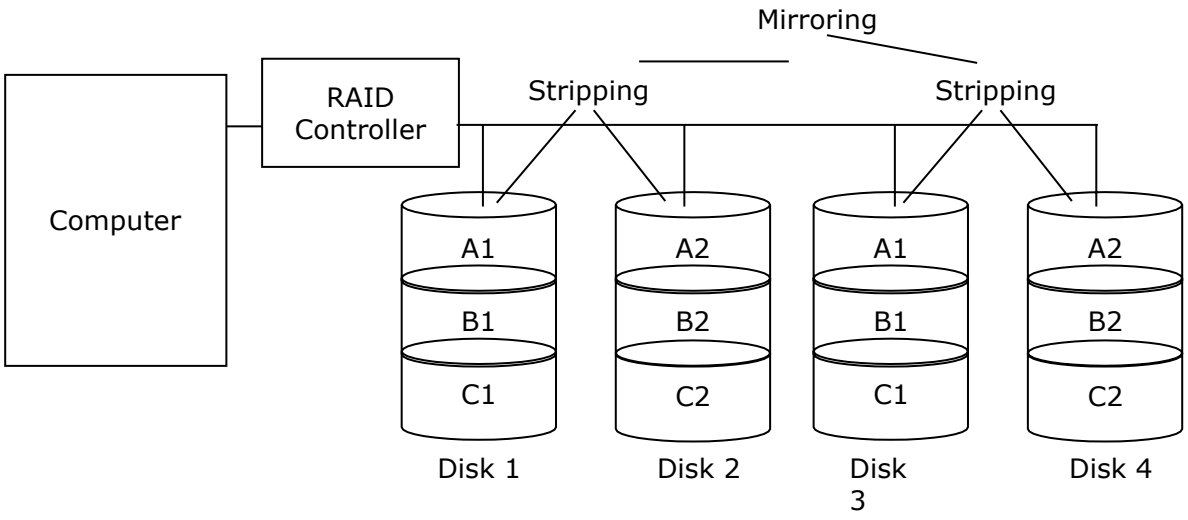
Hình 66 Cấu hình RAID 0



Hình 67 Cấu hình RAID 1

5.3.3.3 RAID 10

Cấu hình RAID 10 là sự kết hợp của RAID 1 và RAID 0, dựa trên cả hai kỹ thuật tạo lát đĩa và soi gương đĩa. RAID 10 cần tối thiểu 4 ổ đĩa độc lập như minh hoạ trên Hình 68. Ưu điểmcủa RAID 10 là đạt được cả tốc độ cao và tính tin cậy cao, nên rất phù hợp với các hệ thống máy chủ đòi hỏi tính an toàn cao, hiệu năng lớn như máy chủ cơ sở dữ liệu. Dung lượng RAID 10 bằng một nửa tổng dung lượng các đĩa độc lập tham gia tạo RAID. Nhược điểm duynhất của RAID 10 là giá thành cao.



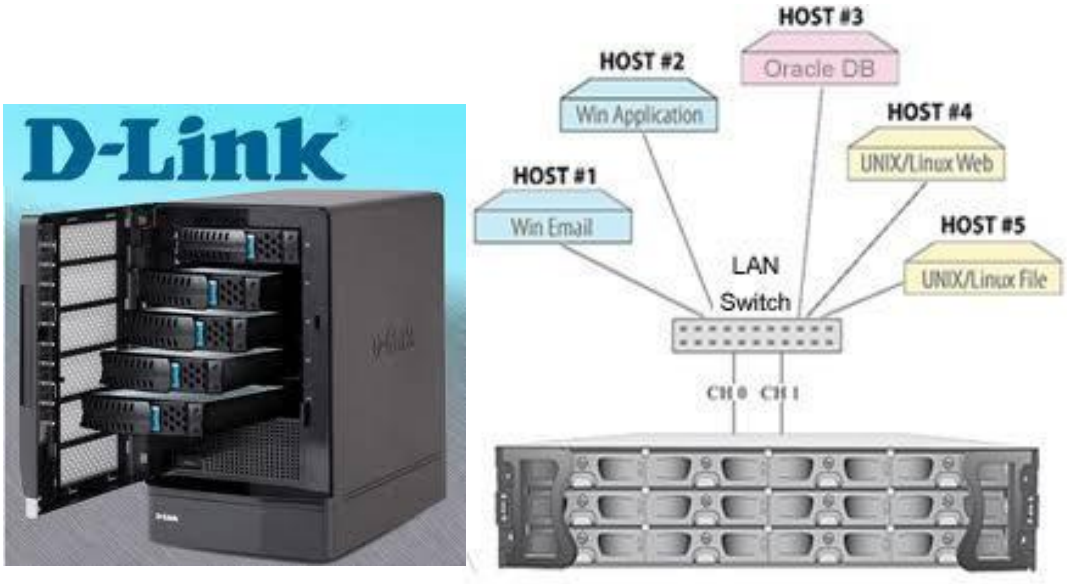
Hình 68 Cấu hình RAID 10

5.4 NAS

NAS (Network Attached Storage) là một dạng thiết bị lưu trữ được gắn trực tiếp vào mạng, thường là mạng cục bộ LAN. Hình 69 minh hoạ một thiết bị NAS USR8700 được gắn vào mạng LAN và cung cấp thiết bị lưu trữ cho cả mạng.

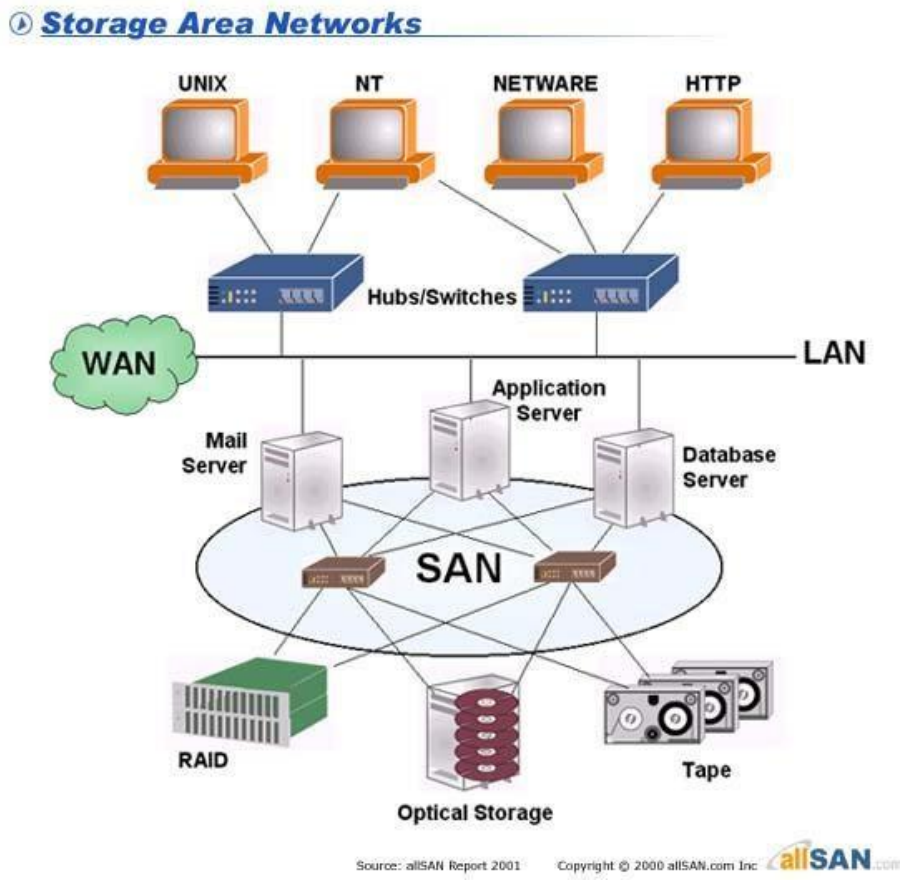
Hình 69 NAS trong một mạng LAN

Trên thực tế, NAS thường là một máy chủ chuyên dùng làm thiết bị lưu trữ, được kết nối vào mạng (thường là LAN tốc độ cao) và cung cấp các dịch vụ lưu trữ thông qua mạng. NASthường dựa trên nền tảng là một RAID có tốc độ cao, dung lượng lớn và độ tin cậy rất cao. NAS có thể cung cấp dịch vụ lưu trữ cho hầu hết các loại máy chủ có cấu hình phần cứng khác nhau và chạy các hệ điều hành khác nhau, cũng như các phần mềm ứng dụng khác nhau, như minh hoạ trên Hình 70.



Hình 70 NAS của hãng D-Link và mô hình sử dụng trong mạng LAN

5.5 SAN



Hình 71 Mô hình mạng lưu trữ SAN và ứng dụng

SAN (Storage Area Network) là một mạng của các máy chủ chuyên dụng cung cấp dịch vụ lưu trữ, với các đặc điểm:

- Tốc độ truy nhập rất cao;
- Dung lượng cực lớn;
- Độ an toàn rất cao
- An toàn dữ liệu cục bộ
- An toàn dữ liệu với các bản copy được đồng bộ ở khoảng cách xa về địa lý.

Thông thường, các SAN thường được tổ chức dưới dạng các hệ thống file phân tán (Distributed File System) phục vụ tổ chức lưu trữ lượng dữ liệu khổng lồ cho các tổ chức và công ty lớn.

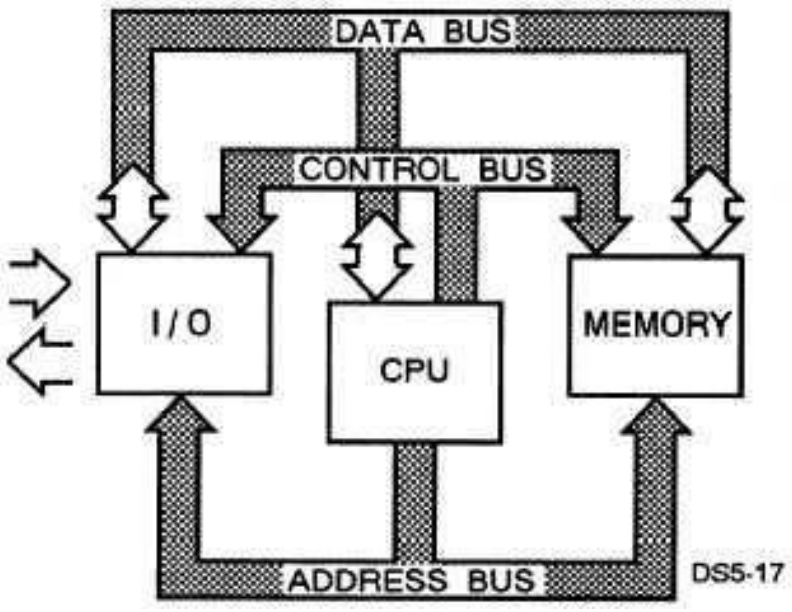
5.6 CÂU HỎI ÔN TẬP

1. Đĩa cứng: cấu tạo, các chuẩn ghép nối, bảng phân khu, thư mục gốc và hệ thống file.
2. Đĩa quang: cấu tạo, nguyên lý đọc CD và các loại đĩa quang.
3. RAID: RAID là gì? các kỹ thuật chính tạo RAID; các cấu hình RAID 0, 1 và 10.
4. Khái niệm về NAS.
5. Khái niệm về SAN.

CHƯƠNG 6 HỆ THỐNG BUS VÀ CÁC THIẾT BỊ NGOẠI VI

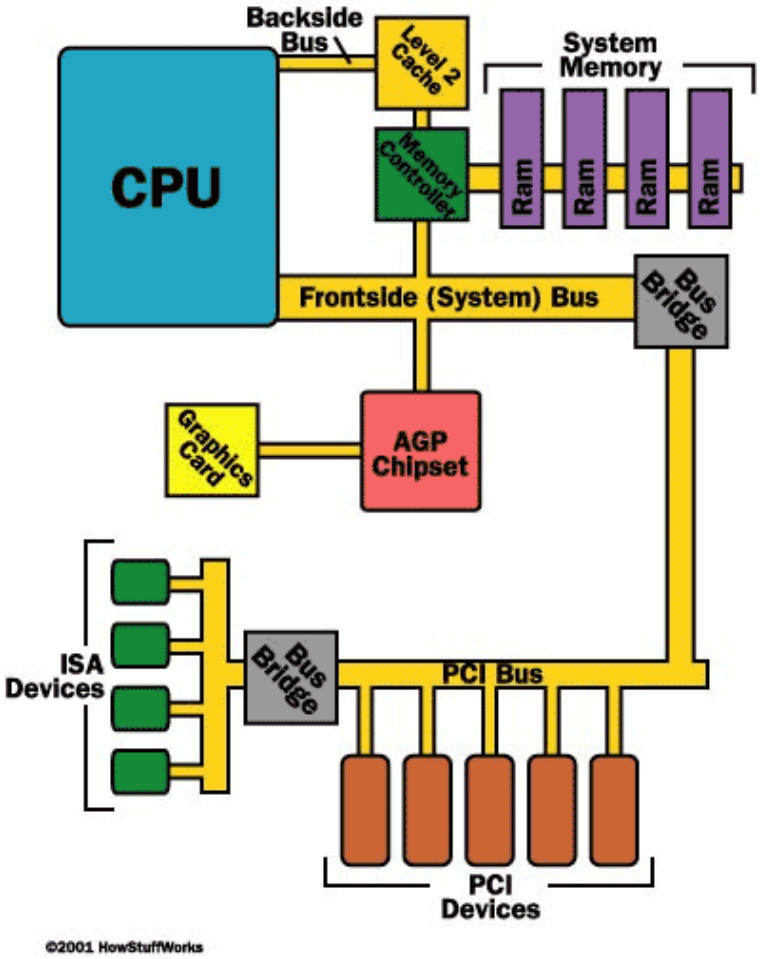
6.1 GIỚI THIỆU CHUNG VỀ HỆ THỐNG BUS

Bus là một hệ thống con (subsystem) có nhiệm vụ truyền dữ liệu giữa các bộ phận trong máy tính. Một hệ thống bus thường gồm ba thành phần: bus địa chỉ, bus dữ liệu và bus điều khiển. Bus địa chỉ (Address Bus – A Bus) là bus một chiều có nhiệm vụ truyền các tín hiệu địa chỉ phát hành bởi CPU đến bộ nhớ hoặc các thiết bị vào ra. Các tín hiệu địa chỉ giúp CPU chọn được ô nhớ cần đọc/ghi hoặc thiết bị vào ra cần trao đổi dữ liệu. Bus dữ liệu (Data Bus – DBus) là bus hai chiều có nhiệm vụ truyền các tín hiệu dữ liệu đi và đến CPU. Dữ liệu được bus dữ liệu chuyển từ CPU đến bộ nhớ hoặc thiết bị vào ra và ngược lại. Bus điều khiển (Control Bus – C Bus) là bus một chiều theo một hướng, có nhiệm vụ truyền các tín hiệu điều khiển từ CPU đến bộ nhớ hoặc thiết bị vào ra, và truyền các tín hiệu trạng thái từ bộ nhớ hoặc thiết bị vào ra về CPU. Các bus địa chỉ, dữ liệu và điều khiển thường phối hợp cùng tham gia truyền dẫn các tín hiệu địa chỉ, dữ liệu và điều khiển trong quá trình CPU trao đổi thông tin với bộ nhớ hoặc các thiết bị vào ra.



Hình 72 Hệ thống bus nguyên lý

Hình 72 minh hoạ hệ thống bus nguyên lý – một hệ thống bus duy nhất kết nối ba thành phần quan trọng nhất của máy tính là CPU, bộ nhớ (memory) và các thiết bị vào ra (I/O). Trên thực tế, hệ thống bus thường được chia thành một số hệ thống bus con theo tần số làm việc và băng thông, nhằm làm cho hệ thống bus làm việc nhịp nhàng hơn với các thành phần có liên quan. Hình 73 minh hoạ hệ thống bus của các máy vi tính được sử dụng gần đây. Theo đó, hệ thống bus gồm các bus: Backside Bus (BSB), Frontside Bus (FSB), AGP Bus, PCI Bus và ISA Bus. BSB là bus riêng kết nối CPU với bộ nhớ cache, còn FSB kết nối CPU với bộ nhớ chính. AGP là bus dành riêng phục vụ card giao tiếp đồ hoạ và PCI bus thường được sử dụng để kết nối với các thiết bị ngoại vi. Bus ISA được sử dụng để kết nối với các thiết bị ngoại vi cũ. Các hệ thống bus con được kết nối với nhau thông qua các cầu bus (Bus Bridge).



Hình 73 Hệ thống bus thực tế

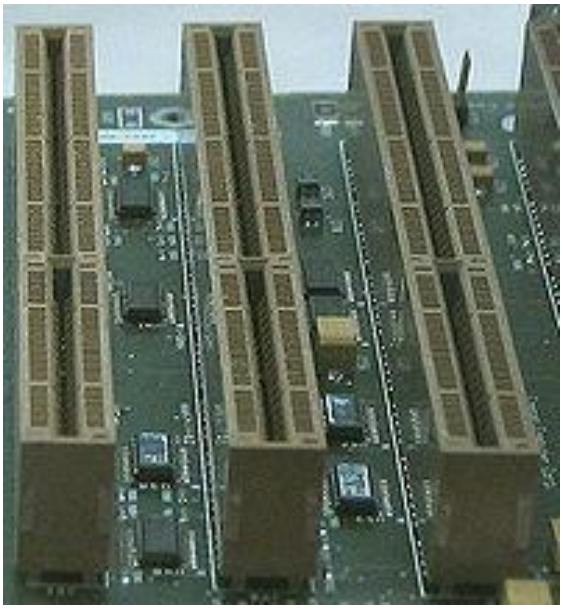
6.2 GIỚI THIỆU MỘT SỐ LOẠI BUS THÔNG DỤNG

6.2.1 Bus ISA và EISA

Bus ISA (Industrial Standard Architecture) là một trong các bus được phát triển sớm nhất. Bus ISA do IBM phát triển năm 1981 với băng thông 8 bit trên máy XT, hoặc 16 bit trên máy AT. ISA hỗ trợ tối đa 6 thiết bị kết nối đồng thời và hoạt động ở các xung nhịp 4, 6 và 8MHz. Hình 74 minh hoạ các khe cắm mở rộng của bus ISA được dùng để kết nối với các card mở rộng ISA.

Hình 74 Khe cắm mở rộng ISA

Bus EISA là một mở rộng của bus ISA ra đời vào năm 1988. EISA hỗ trợ băng thông 32 bits, nhưng nó vẫn tương thích với các thiết bị theo chuẩn ISA 8 và 16 bit. EISA hoạt động với xung nhịp 8.33MHz và đạt tốc độ truyền dữ liệu 33MB/s. Hình 75 minh hoạ các khe cắm mở rộng của bus EISA được dùng để kết nối với các card mở rộng ISA và EISA. Hiện nay, bus ISA và EISA đã lạc hậu và không còn được sử dụng.

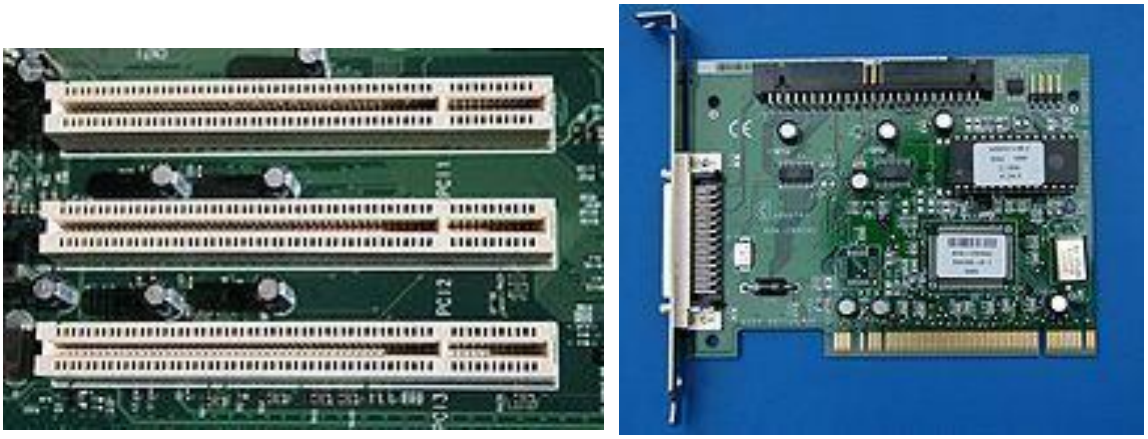


Hình 75 Khe cắm mở rộng EISA

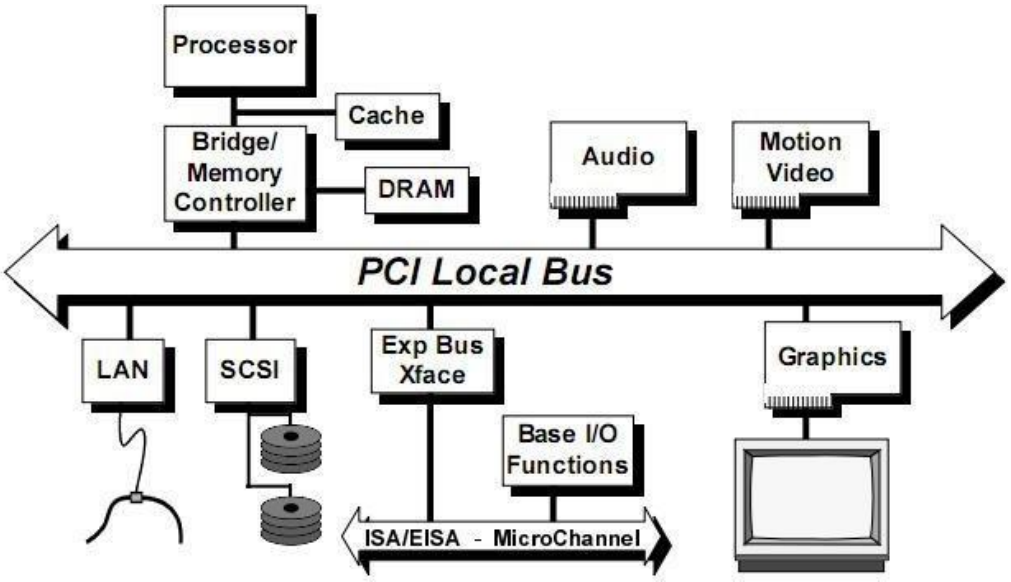
6.2.2 Bus PCI

6.2.2.1 Giới thiệu bus PCI

Bus PCI (Peripheral Component Interconnect) do Intel phát triển năm 1993 và được phát triển thành một trong các bus được sử dụng rộng rãi nhất cho đến ngày nay. PCI hỗ trợ băng thông 32 bit hoặc 64 bit và đạt tốc độ truyền dữ liệu khá cao theo tần số làm việc và băng thông. Với băng thông 32 bit, tốc độ truyền dữ liệu đạt 133 MB/s tại tần số 33MHz và 266 MB/s tại tần số 66MHz. Với băng thông 64 bit, tốc độ truyền dữ liệu đạt 266 MB/s tại tần số 33MHz và 533 MB/s tại tần số 66MHz. Hình 76 minh hoạ khe cắm PCI và card mở rộng thiết bị PCI và Hình 77 minh hoạ bus cục bộ PCI – các thành phần tham gia vào “gia đình” PCI.

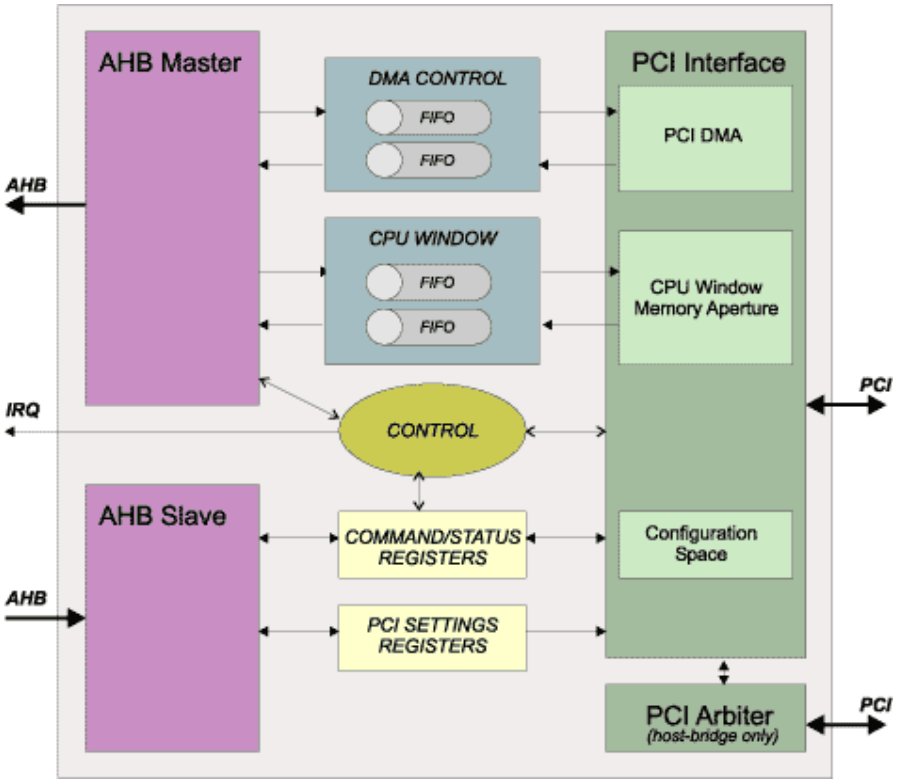


Hình 76 Khe cắm và card thiết bị PCI



Hình 77 Bus cục bộ PCI

6.2.2.2 Nguyên lý hoạt động của bus PCI



Hình 78 Sơ đồ khối nguyên lý hoạt động của bus PCI

Hình 78 nêu sơ đồ khối nguyên lý hoạt động của bus PCI. Theo đó PCI là một bus dùng chung hay bus chia sẻ (shared bus). PCI hỗ trợ nhiều thiết bị kết nối đồng thời, nhưng tại mỗi thời điểm, chỉ có một cặp thiết bị được sử dụng bus để trao đổi dữ liệu. Việc trao đổi dữ liệu trên bus PCI được thực hiện thông qua các *giao dịch* (transaction). Thiết bị khởi tạo (Initiator) quá trình truyền dữ liệu được gọi là *thiết bị chủ* (ABH Master) và thiết bị nhận dữ liệu hay

thiết bị đích (Target) là *thiết bị thợ* (ABH Slave). Một trọng tài có nhiệm vụ điều độ các giao dịch trên bus PCI được gọi là bộ tuỳ chọn (PCI Arbiter).

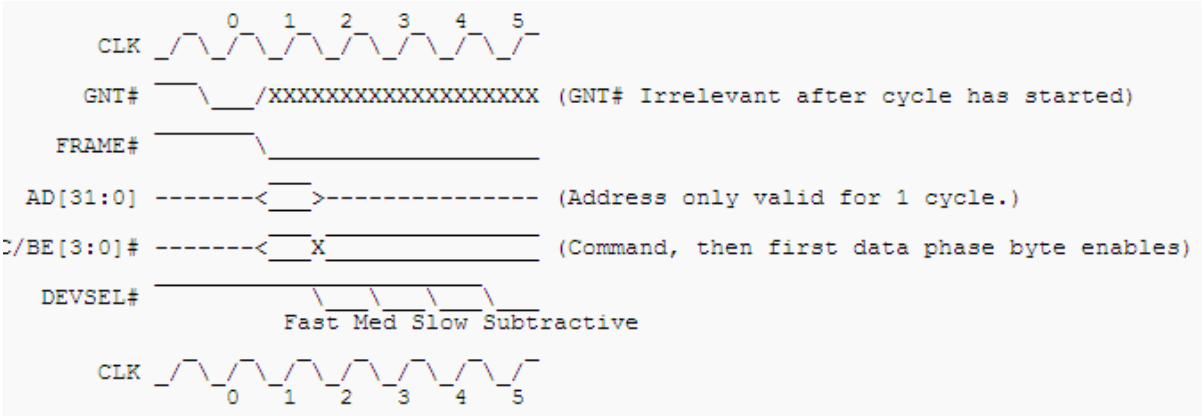
Việc thực hiện các giao dịch trên bus PCI được điều khiển bởi các tín hiệu. Hai nhóm tín hiệu chính được sử dụng, gồm: tín hiệu khởi tạo giao dịch và tín hiệu điều khiển giao dịch. Các tín hiệu khởi tạo một giao dịch, gồm tín hiệu REQ# do thiết bị khởi tạo giao dịch gửi tín hiệu yêu cầu sử dụng bus và tín hiệu GNT# do bộ tuỳ chọn gửi tín hiệu cho phép sử dụng bus. Các tín hiệu điều khiển một giao dịch, gồm tín hiệu FRAME# - bắt đầu chu kỳ bus, tín hiệu IRDY# - thiết bị khởi tạo đã sẵn sàng, tín hiệu DEVSEL# - thiết bị đích xác nhận bắt đầu giao dịch, tín hiệu TRDY# - thiết bị đích đã sẵn sàng và tín hiệu STOP# - dừng giao dịch.

Một giao dịch PCI được thực hiện theo 3 pha: pha tuỳ chọn (Arbitration), pha địa chỉ (Address) và pha dữ liệu (Data). Pha tuỳ chọn có nhiệm vụ khởi tạo giao dịch, pha địa chỉ xác định địa chỉ bên tham gia giao dịch và pha dữ liệu truyền dữ liệu giữa các bên. Pha tuỳ chọn được thực hiện thông qua các bước sau:

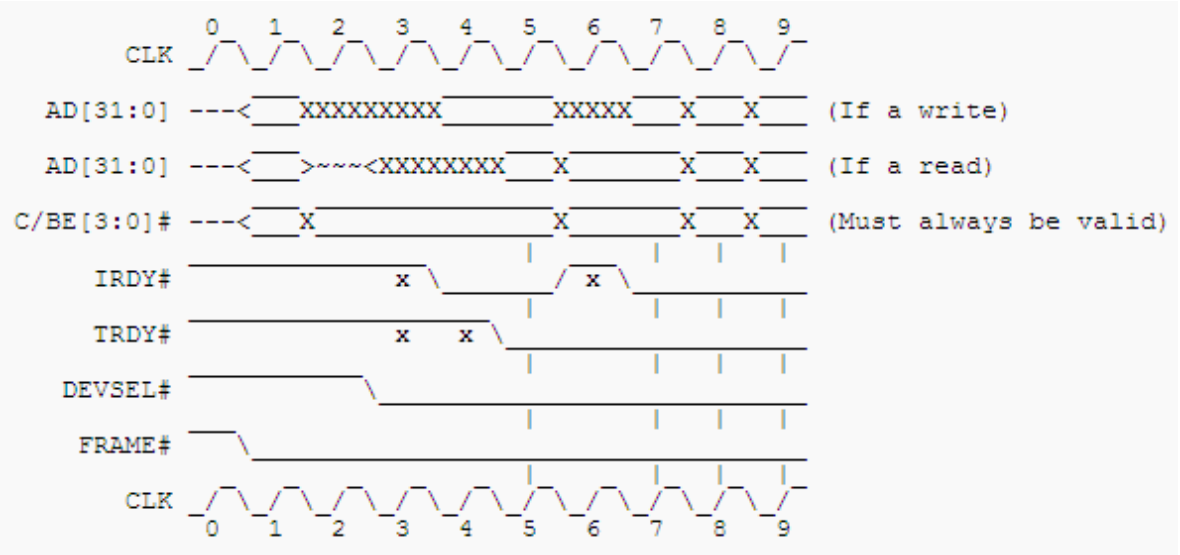
- Thiết bị PCI (Initiator) gửi tín hiệu REQ# đến Arbiter yêu cầu sử dụng bus;
- Nếu bus rỗi, Arbiter gửi tín hiệu cho phép sử dụng bus GNT# đến Initiator;
- Nếu bus bận, yêu cầu sử dụng bus được đưa vào hàng đợi;
- Tín hiệu cho phép sử dụng bus GNT# có thể bị Arbiter huỷ tại bất kỳ thời điểm nào;
- Thiết bị PCI được cấp tín hiệu cho phép sử dụng bus GNT# có thể bắt đầu phiên truyền dữ liệu nếu bus rỗi.

Pha địa chỉ của giao dịch như minh hoạ trên Hình 79, có thể gồm các bước:

- Thiết bị PCI (Initiator) có tín hiệu cho phép sử dụng bus GNT# có thể bắt đầu một giao dịch PCI bằng việc gửi tín hiệu FRAME# và gửi địa chỉ thiết bị đích cùng các lệnh liên quan (Read/Write);
- Mỗi thiết bị PCI sẽ kiểm tra địa chỉ và lệnh kèm theo để xác định mình có phải là thiết bị đích hay không. Thiết bị đích (có địa chỉ trùng với địa chỉ gửi bởi Initiator) sẽ gửi tín hiệu trả lời DEVSEL# đến Initiator;
- Thiết bị đích phải gửi tín hiệu trả lời DEVSEL# trong thời gian 3 chu kỳ đồng hồ.



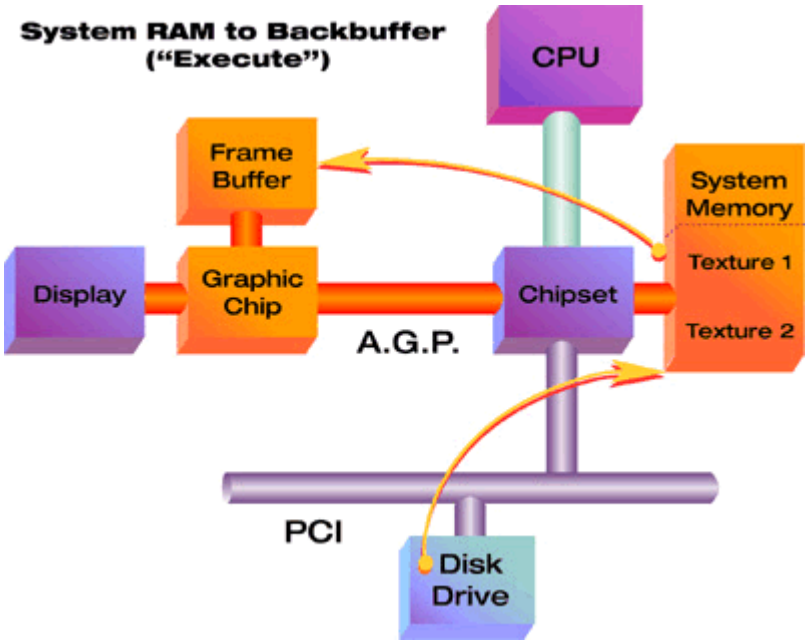
Hình 79 Pha địa chỉ giao dịch PCI



Hình 80 Pha dữ liệu giao dịch PCI

Hình 80 minh hoạ các tín hiệu trong pha dữ liệu của giao dịch PCI. Sau pha địa chỉ, khi tín hiệu DEVSEL# ở mức thấp là một hoặc một số pha dữ liệu. Kết thúc pha dữ liệu, thiết bị đích gửi tín hiệu STOP#.

6.2.3 Bus AGP

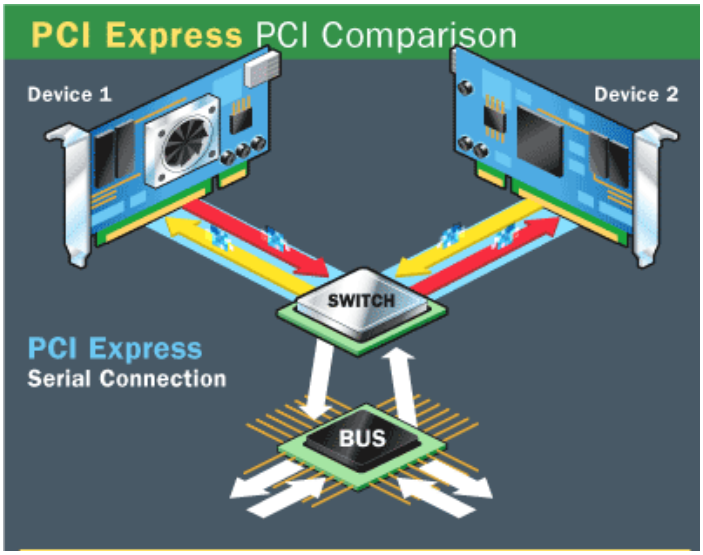


Hình 81 Sơ đồ nguyên lý hoạt động của AGP

Bus AGP (Accelerated Graphic Port) do Intel phát triển năm 1993 với mục đích chính sử dụng cho kết nối với các mạch xử lý đồ hoạ tốc độ cao. AGP đã hoàn toàn thay thế PCI trong lĩnh vực giao tiếp đồ hoạ trong các năm sau đó. AGP hỗ trợ băng thông 32 bit với tốc độ truyền dữ liệu nhanh gấp nhiều lần so với bus PCI. Cụ thể, AGP hỗ trợ 4 cấp tốc độ truyền dữ liệu là 1x, 2x, 4x và 8x, với tốc độ lần lượt là 266MB/s, 533MB/s, 1066MB/s và 2133MB/s tại các tần số tương ứng 66MHz, 133MHz, 266MHz và 533MHz.

6.2.4 Bus PCI Express

Bus PCI Express (còn gọi là PCIe) do Intel phát triển năm 2004, là một dạng bus truyền dữ liệu nối tiếp, kiểu điểm đến điểm (point to point) với tốc độ cao. Độ rộng bus là từ 1-32 bit tuỳ theo cấu hình. PCI Express được cấu trúc từ các liên kết nối tiếp điểm đến điểm và một cặp liên kết nối tiếp (theo 2 chiều ngược nhau) tạo thành một luồng (lane). Các luồng được định tuyến đồng thời qua một bộ chuyển mạch (crossbar switch). Tối đa, bus PCI Express có thể hỗ trợ đến 32 luồng. Tốc độ truyền dữ liệu của bus PCI Express phụ thuộc số luồng sử dụng và phiên bản của chuẩn. Với một luồng, tốc độ truyền đạt 250MB/s, 500MB/s và 1GB/s tương ứng với các phiên bản 1.x, 2.0 và 3.0.



Hình 82 Truyền dữ liệu qua bộ Switch trong PCI Express

Khác với PCI là bus chia sẻ, bus PCI Express có khả năng cung cấp đường truyền riêng cho các cặp thiết bị tham gia sử dụng bus. Đồng thời PCI Express cũng hỗ trợ nhiều cặp thiết bị cùng tham gia truyền dữ liệu sử dụng các luồng truyền khác nhau. Hình 82 minh hoạ việc truyền dữ liệu qua bộ chuyển mạch (Switch) trong PCI Express.

6.3 GIỚI THIỆU CHUNG VỀ CÁC THIẾT BỊ NGOẠI VI

6.3.1 Giới thiệu chung

Các thiết bị ngoại vi (peripheral devices) hay còn gọi là thiết bị vào ra, là các bộ phận của hệ thống máy tính có nhiệm vụ: (1) tiếp nhận các thông tin từ thế giới bên ngoài đi vào máy tính và (2) kết xuất các thông tin từ máy tính ra thế giới bên ngoài. Nhiệm vụ (1) được đảm bảo bởi nhóm các thiết bị vào (input devices) và nhiệm vụ (2) được đảm bảo bởi nhóm các thiết bị ra (output devices). Các thiết bị vào gồm có: bàn phím, chuột, ổ đĩa (đọc thông tin), máy quét ảnh và máy đọc mã vạch. Hình 83 minh hoạ thiết bị vào chuẩn là bàn phím và chuột. Các thiết bị ra gồm có: màn hình, máy in, ổ đĩa (ghi thông tin) và máy vẽ. Hình 84 minh hoạ hai loại màn hình thông dụng: màn hình CRT và LCD. Hình 85 minh hoạ các máy in laser và máy in phun mực.

Hình 83 Bàn phím và chuột

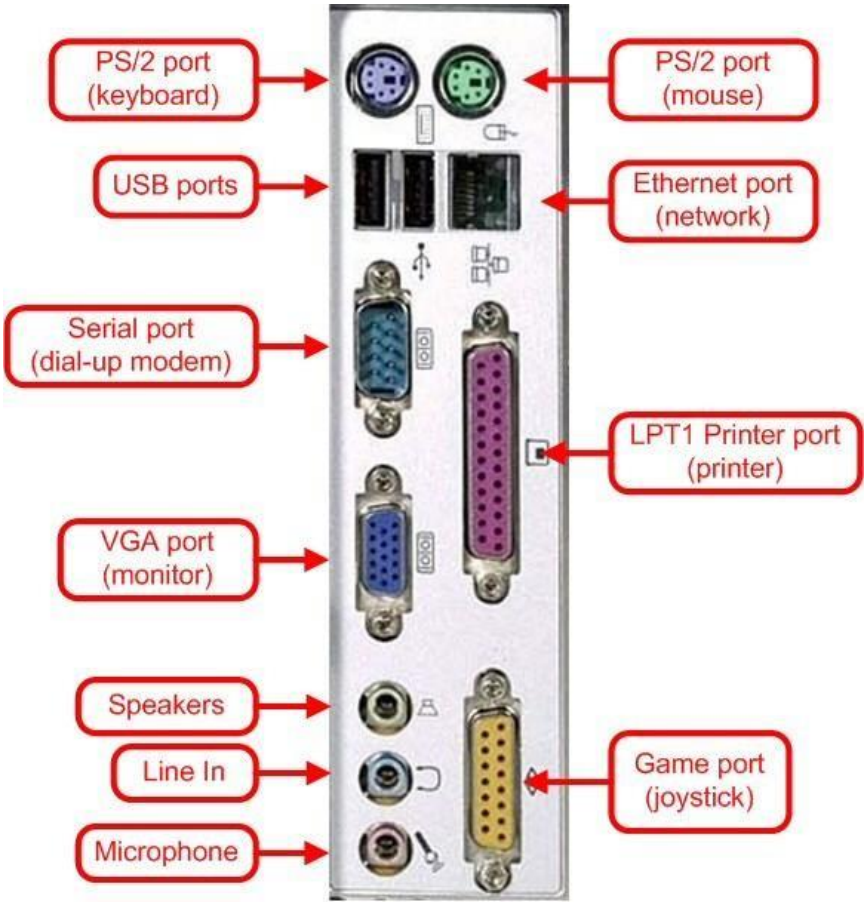
Hình 84 Màn hình CRT và LCD

Hình 85 Máy in Laser và máy in phun mực

6.3.2 Các cổng giao tiếp

Các thiết bị vào ra thường kết nối với máy tính thông qua các cổng giao tiếp (communication ports). Mỗi cổng giao tiếp được gán một địa chỉ và có tập tham số làm việc riêng. Hình 86 minh hoạ các cổng giao tiếp ở phía sau máy tính. Các cổng giao tiếp thông dụng:

1. PS/2: kết nối chuột và bàn phím.
2. Cổng COM và LPT.
3. Cổng USB: cổng giao tiếp đa năng theo chuẩn USB
   - USB 1.0: 12Mb/s
   - USB 2.0: 480Mb/s (hiện tại)
   - USB 3.0: 1.5Gb/s (tương lai).
4. Cổng IDE, SATA và E-SATA: ghép nối các loại ổ đĩa.
5. Cổng LAN: ghép nối mạng.
6. Cổng Audio: ghép nối âm thanh.
7. Cổng đọc các thẻ nhớ.
8. Cổng Firewire /IEEE 1394: ghép nối các loại ổ đĩa ngoài.
9. Cổng VGA/Video: ghép nối với màn hình.
10. Cổng DVI: ghép nối với màn hình số.

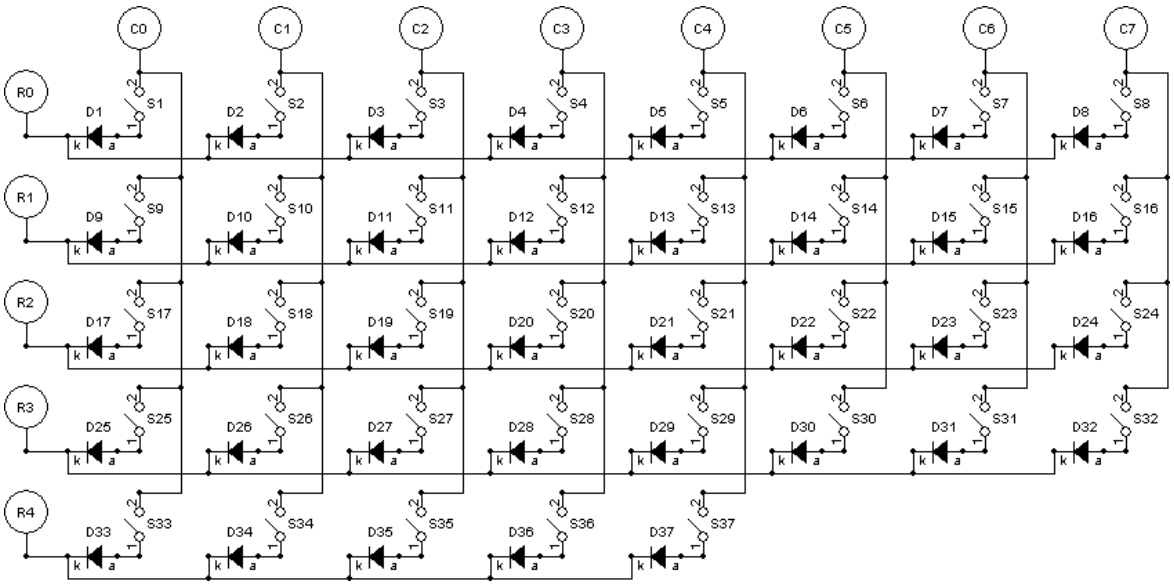


Hình 86 Một số cổng giao tiếp với máy tính

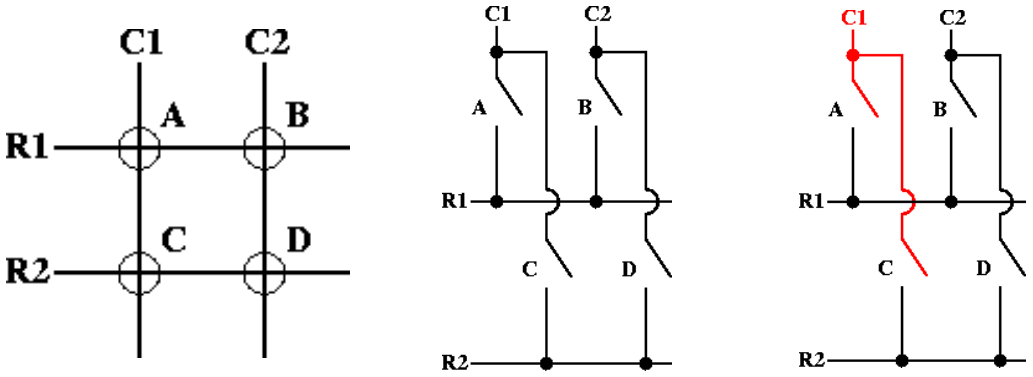
6.4 GIỚI THIỆU MỘT SỐ THIẾT BỊ VÀO RA THÔNG DỤNG

6.4.1 Bàn phím

Bàn phím (keyboard) là thiết bị vào chuẩn của máy tính do bàn phìm có thể đảm nhiệm cả chức năng nhập dữ liệu và điều khiển máy tính. Bàn phím tiêu chuẩn có 101 phím: các phím ký tự (a-z), các phím số (0-9), các phím phép toán (+, - *, /), các phím chức năng (F1-F12), các phím điều khiển (Ctrl, Alt, Shift, ..) và các phím di chuyển: Home, End, Page Up, Page Down, Up, Down, Left, Right, ...



Hình 87 Mạch tạo phím



Hình 88 Ma trận phím và phát hiện các phím được nhấn

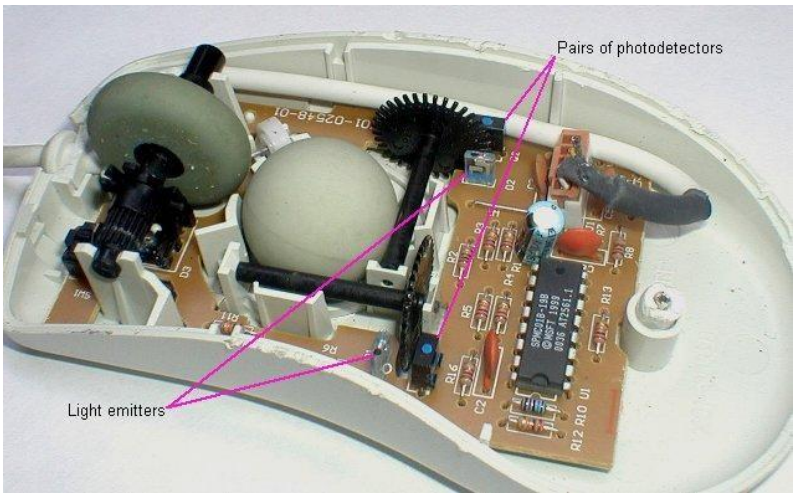
Bàn phím sử dụng một ma trận hình thành bởi các dòng và cột dây dẫn, như minh hoạ trên hình Hình 87 và Hình 88. Mỗi phím hoạt động như một công tắc điện. Khi phím được ấn, dây dẫn cột được nối với dây dẫn dòng tạo thành một mạch kín. Bộ điều khiển bàn phím liên tục quét ma trận phím để phát hiện mạch kín và ghi nhận phím được ấn. Quá trình xử lý phím ấn và tạo tín hiệu gửi CPU xử lý trong bàn phím có thể được tóm tắt như sau:

- Khi một phím được ấn, bộ điều khiển bàn phím phát hiện và sinh ra một mã quét tương ứng (scan code);
- Một ngắt (interrupt) bàn phím được gửi đến máy tính;

- Khi nhận được tín hiệu ngắt bàn phím:
  - Máy tính thực hiện chương trình điều khiển ngắt bàn phím:
    - Đọc mã quét phím
    - Chuyển mã quét phím thành mã ký tự tương ứng (thông thường là mã ASCII).
  - Một ký tự có thể được hiển thị theo nhiều hình thức khác nhau theo các bộ font.

6.4.2 Chuột

Chuột (mouse) là một trong các thiết bị vào của máy tính được sử dụng rộng rãi nhất. Chức năng chính của chuột là điều khiển. Thông qua các phần mềm, hình thức hiển thị của chuột được thể hiện rất đa dạng, từ hình mũi tên đơn giản, đến bàn tay, đồng hồ cát, ... theo các trạng thái làm việc của chương trình. Hiện nay, có rất nhiều loại chuột đang được sử dụng. Ngoài chuột bi (còn gọi là chuột cơ khí), còn có chuột quang, chuột laser, chuột cảm ứng và chuột không dây. Các phím bấm chuột cũng rất đa dạng: thông thường là loại 3 phím (trái, phải và cuộn); một số chuột có thể có thêm cả phím tiến (forward) và phím lùi (backward).

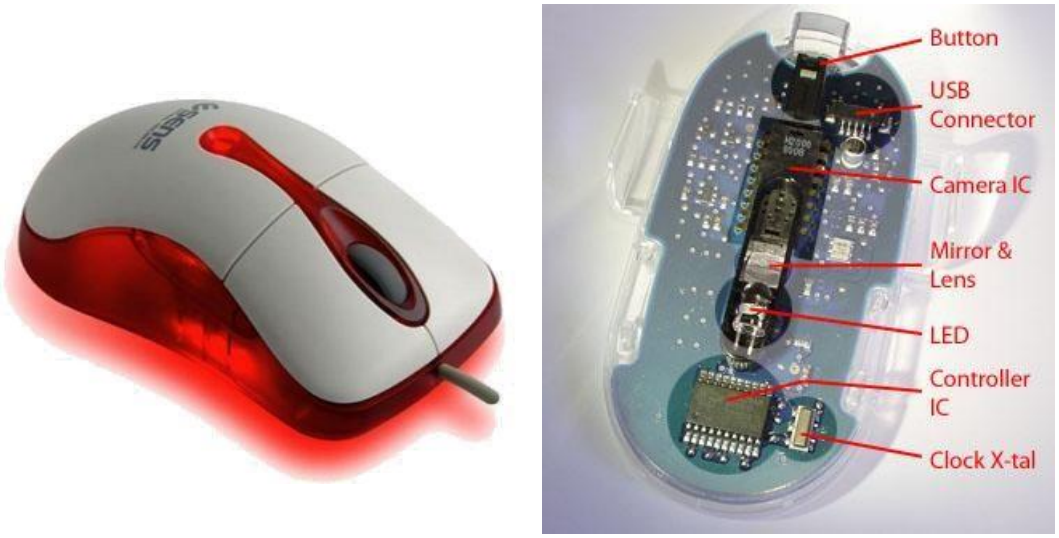


Hình 89 Chuột bi hay chuột cơ khí

Chuột bi hay chuột cơ khí là loại chuột có cấu tạo đơn giản và được sử dụng sớm nhất. Hình 89 cho thấy các thành phần bên trong của chuột bi. Chuột bi hoạt động theo nguyên tắc cơ khí – quang – điện: biến chuyển động của viên bi khi rê chuột thành các tín hiệu điện biểu diễn các chuyển động theo phương ngang và phương đứng của chuột. Cụ thể, nguyên tắc hoạt động của chuột bi có thể tóm tắt như sau:

- Khi chuột di chuyển, viên bi chuột quay;
- Khi bi quay nó kéo theo 2 trục áp vào quay theo. Hai trục được gắn bánh xe răng cưa ở 2 đầu:
  - Một trục dùng để phát hiện chuyển động theo phương đứng
  - Một trục dùng để phát hiện chuyển động theo phương ngang
- Hai đi-ốt sinh tia hồng ngoại chiếu qua phần bánh răng cưa gắn trên các trục kể trên:
  - Khi bánh răng cưa quay, ánh sáng hồng ngoại chiếu qua sẽ bị ngắt quãng;

- Ở phía đối diện có 2 bộ cảm biến chuyển ánh sáng hồng ngoại sau bánh răng cưa thành tín hiệu điện;
- Tín hiệu điện thu được phản ánh chuyển động của chuột được chuyển cho máy tính xử lý.



Hình 90 Chuột quang và cấu tạo

Khác với chuột bi, chuột quang (optical mouse) không có bi nên thường nhẹ và đạt độ chính xác cao hơn. Hiện nay, chuột quang đã thay thế hầu hết các chuột bi. Hình 90 minh hoạ chuột quang và cấu tạo của nó. Chuột quang sử dụng nguyên tắc liên tục chụp và phân tích ảnh bề mặt chuột di chuyển để phát hiện chuyển động của chuột. Cụ thể, nguyên tắc hoạt động của chuột quang có thể tóm tắt như sau:

- Một đi-ốt phát ánh sáng đỏ qua ống kính chiếu xuống mặt phẳng di chuột; ánh sáng phản xạ từ mặt phẳng di chuột quay ngược trở lại phía dưới chuột;
- Một camera đặt phía dưới chuột liên tục chụp ảnh của bề mặt di chuột nhờ ánh sáng phản xạ. Tốc độ chụp là khoảng 1500 ảnh/giây;
- IC điều khiển chuột sẽ phân tích và so sánh các ảnh kề nhau và qua đó phát hiện ra chuyển động chuột;
- Tín hiệu biểu diễn chuyển động chuột do IC điều khiển chuột sinh ra được chuyển cho máy tính xử lý.

Tương tự như chuột quang, chuột laser cũng sử dụng phương pháp chụp và phân tích ảnh bề mặt kề nhau để phát hiện chuyển động. Tuy nhiên, chuột laser sử dụng ánh sáng laser với tốc độ chụp ảnh lên đến 6000 ảnh/giây. Nhờ vậy, chuột laser thường có độ chính xác và độ nhạy cao hơn so với chuột quang.

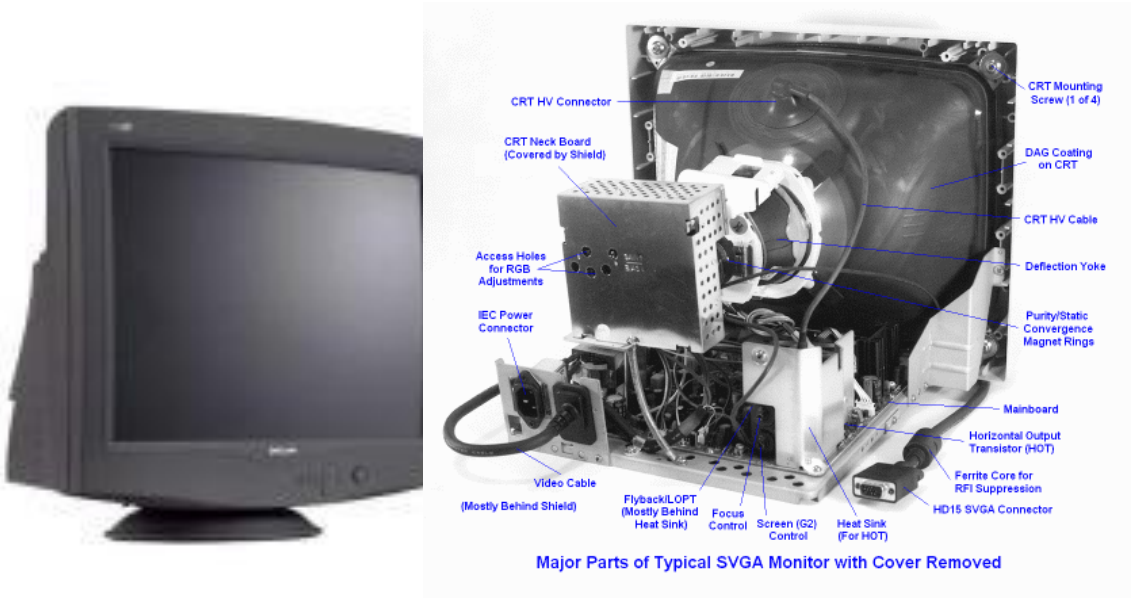
6.4.3 Màn hình

Màn hình (monitor / screen) là thiết bị ra chuẩn có thể hiển thị thông tin dưới dạng văn bản hoặc hình ảnh. Cùng với bàn phím và chuột, màn hình là thiết bị không thể thiếu đối với máy tính. Có ba dạng màn hình được sử dụng thông dụng: màn hình ống điện tử CRT, màn hình

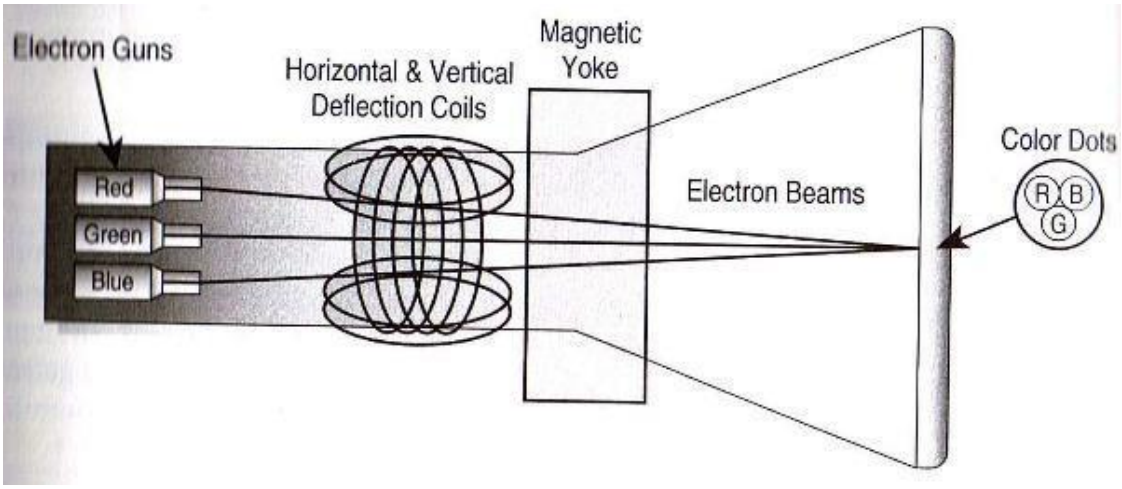
tinh thể lỏng LCD và màn hình plasma. Tài liệu này chỉ đề cập đến hai loại màn hình được sử dụng phổ biến cho máy tính là màn hình CRT và LCD.

6.4.3.1 Màn hình CRT

Màn hình CRT (Cathode Ray Tube) sử dụng tia điện tử phát ra từ cực Cathode bắn lên mặt huỳnh quang phốt pho để tạo ảnh. Tia điện tử được điều khiển bởi 2 cuộn lái tia (dòng vàmành) để quét hết cả màn hình, đảm bảo tốc độ quét tối thiểu là 24 màn hình/giây. Tín hiệu hình ảnh (video) được sử dụng để điều khiển mật độ tia điện tử bắn lên màn huỳnh quang tạo các mức sáng/tối khác nhau. Màn hình đen trắng sử dụng 1 súng điện tử, còn màn hình mầu sử dụng 3 súng điện tử ứng với 3 mầu cơ bản Đỏ (Red), Xanh lá cây (Green) và Xanh da trời (Blue). Ba mầu này được trộn với nhau theo tỷ lệ khác nhau tạo thành tất cả các mầu có trong tự nhiên cho điểm ảnh. Hình 91 minh hoạ bên ngoài và các bộ phận bên trong của màn hình CRT, còn Hình 92 minh hoạ nguyên lý tạo điểm ảnh của màn hình CRT màu.



Hình 91 Bên ngoài và bên trong màn hình CRT

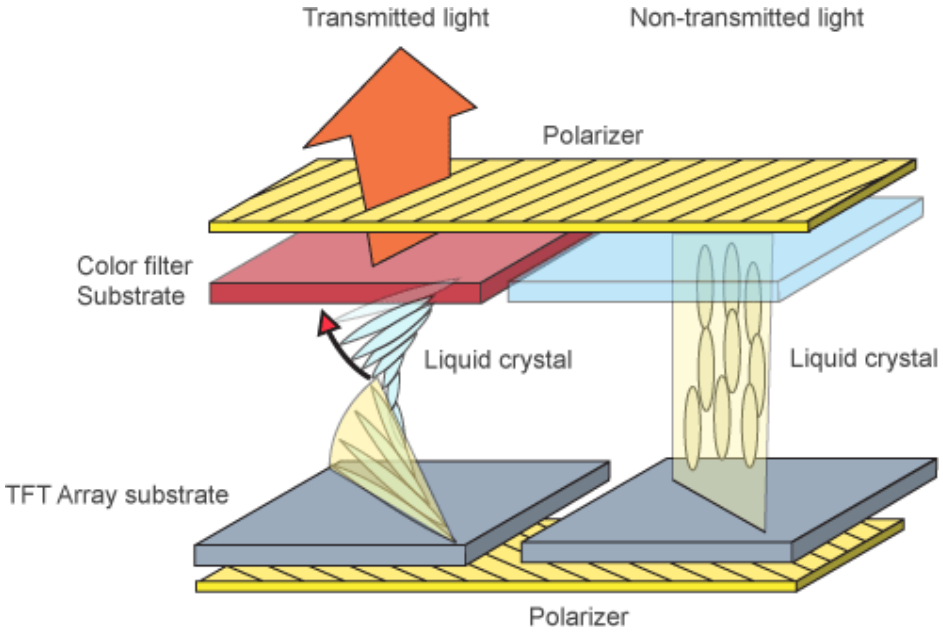


Hình 92 Nguyên lý điểm ảnh của màn hình CRT màu

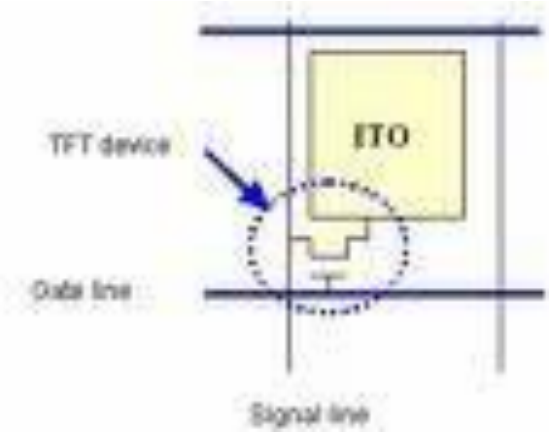
6.4.3.2 Màn hình LCD

Màn hình LCD (Liquid Crystal Display) là màn hình tạo ảnh dựa trên sự linh động của các "*tinh thể lỏng*" (Liquid Crystals). Tinh thể lỏng là các chất bán rắn lỏng rất nhạy cảm với nhiệt độ và dòng điện. So với màn hình CRT, màn hình LCD mỏng hơn, nhẹ hơn và tiêu thụ ít điện năng hơn. Ngoài ra, phần diện tích màn hình thực để hiển thị ảnh (viewable size) của LCD cũng lớn hơn. Chẳng hạn màn hình LCD 15" có phần màn hình thực tương đương màn hình CRT 17". Nhược điểm của LCD so với CRT là không hỗ trợ nhiều độ phân giải, chất lượng ảnh không cao, thời gian đáp ứng (response time) lớn và góc nhìn (view angle) nhỏ.

Có thể phân loại màn hình LCD thành 2 loại theo nguồn pháp sáng: LCD chiếu sau (backlit) và LCD phản xạ (reflective). LCD chiếu sau sử dụng nguồn sáng riêng đặt ở phía sau, thường dùng trong các LCD có công suất lớn, như màn hình máy tính và màn hình tivi. LCD phản xạ sử dụng ánh sáng phản xạ của nguồn sáng từ bên ngoài. LCD phản xạ có thiết kế đơn giản, rẻ tiền, thường thích hợp với các màn hình có công suất nhỏ, như màn hình đồng hồ, màn hình máy tính tay.

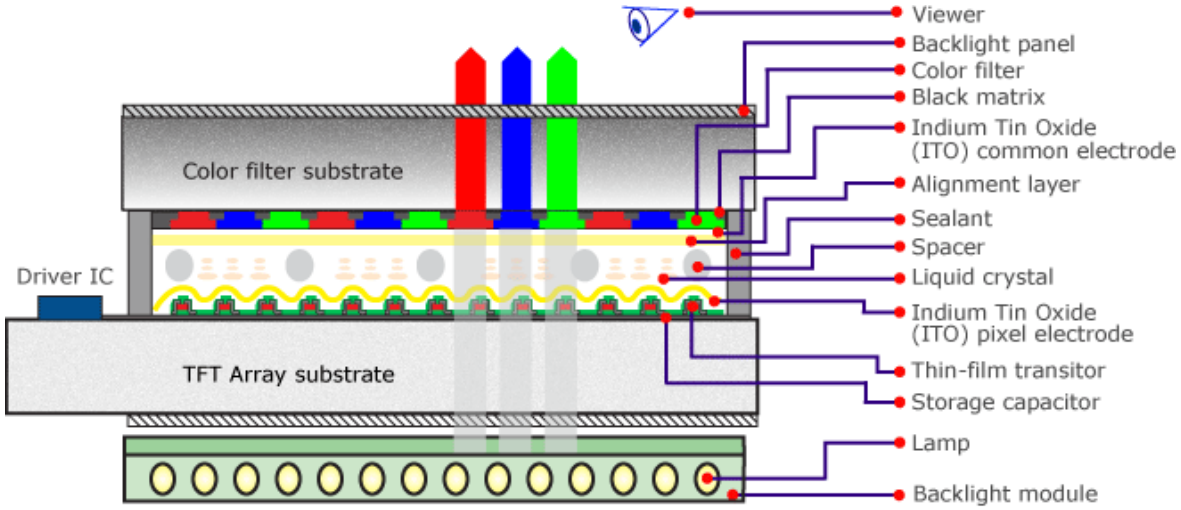


Hình 93 Mô hình lọc ánh sáng của tinh thể lỏng điều khiển bằng điện



Hình 94 Một TFT - Thin Film Transistor

Bản thân tinh thể lỏng không có khả năng phát sáng, nhưng chúng có khả năng "lọc" hay thay đổi cường độ ánh sáng đi qua theo điện áp dòng điện đặt vào. Hình 93 minh hoạ mô hình lọc ánh sáng của tinh thể lỏng được điều khiển bằng điện. Dựa trên phương pháp điều khiển các tinh thể lỏng, ta có 2 loại LCD: LCD ma trận thụ động (Passive matrix) và LCD ma trận chủ động (Active matrix). LCD ma trận thụ động sử dụng lưới hoặc ma trận để định nghĩa từng điểm ảnh (pixel) bởi hàng và cột của nó. Một điểm ảnh (giao giữa 1 hàng và 1 cột) được kích hoạt khi điện áp được đặt vào cột và dòng tương ứng được nối đất. Ngược lại, LCD ma trận chủ động sử dụng một TFT (Thin Film Transistor) để điều khiển một phần tử tinh thể lỏng. Các TFT hoạt động tương tự như các bộ chuyển mạch, như minh hoạ trên Hình 94.



Hình 95 Cấu trúc của màn hình TFT-LCD

Hình 95 minh hoạ cấu trúc của màn hình TFT-LCD. TFT-LCD hoạt động theo nguyên lý tóm tắt như sau:

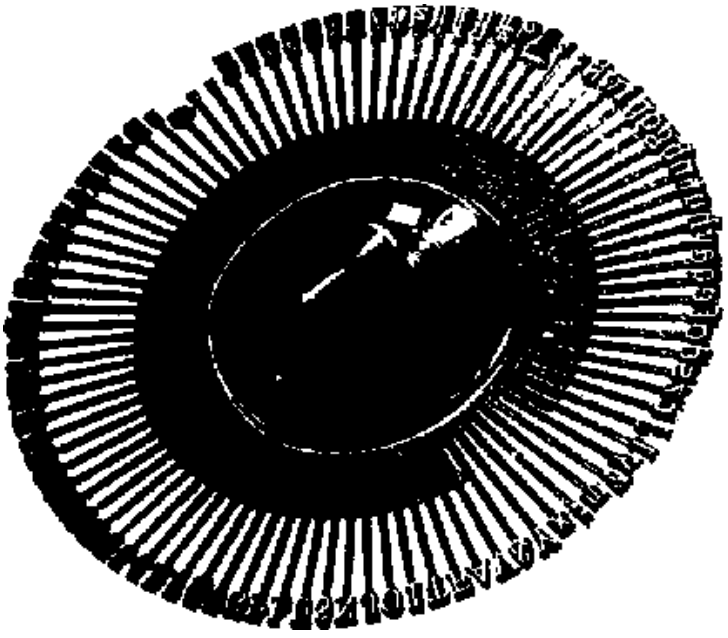
- TFT LCD là thiết bị được điều khiển bằng các tín hiệu điện;
- Lớp tinh thể lỏng nằm giữa 2 lớp trong suốt chứa các điện cực ITO (Indium Tin Oxide);
- Các phần tử tinh thể lỏng được sắp đặt theo các hướng khác nhau theo sự thay đổi điện áp đặt vào các điện cực ITO;
- Hướng của các phần tử tinh thể lỏng trực tiếp ảnh hưởng đến cường độ ánh sáng đi qua và nó gián tiếp điều khiển mức sáng / tối (còn gọi là mức xám) của ảnh hiển thị;
- Màu của hình ảnh được tạo bởi một lớp lọc màu;
- Mức xám của các điểm ảnh được thiết lập theo mức điện áp của tín hiệu video đưa vào điện cực điều khiển.

6.4.4 Máy in

Máy in (printer) là thiết bị ra phổ biến dùng để kết xuất thông tin ra giấy. Qua quá trình phát triển, có nhiều loại máy in được sử dụng như máy in búa (Typewriter-derived printers), máy in kim (Dot-matrix printers), máy in laser (Laser printers), máy in phun mực (Inkjet printers), máy in màu (Colour printers) và các máy in đa chức năng (Multi-function printers).

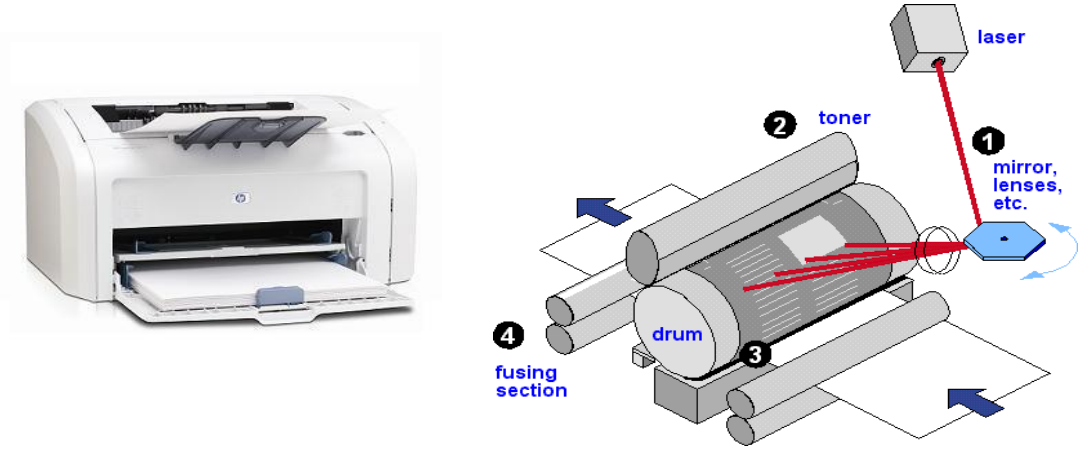
Hình 96 minh hoạ máy in búa. Máy in búa sử dụng các con chữ có kích thước cố định như máy đánh chữ. Ngược lại, máy in kim sử dụng bộ kim để tạo ma trận các chấm để tạo khuôn chữ như minh hoạ trên Hình 97. Hình 98 và Hình 99 minh hoạ nguyên lý hoạt động của máy in laser. Khác với các dòng máy in đi trước, máy in laser sử dụng phương pháp chụp ảnh điện tích bằng tia laser để tạo chữ. Cụ thể như sau:

- Trống cảm quang được nạp một lớp điện tích nhờ 1 điện cực;
- Tia laser từ nguồn sáng laser đi qua một gương quay và bộ điều chế tia được điều khiển bởi tín hiệu cần in đến mặt trống;
- Ánh sáng laser làm thay đổi mật độ điện tích trên mặt trống; Như vậy, mật độ điện tích trên mặt trống thay đổi theo tín hiệu cần in;
- Khi trống cảm quang quay đến hộp mực thì điện tích trên trống hút các hạt mực được tích điện trái dấu. Các hạt mực dính trên trống biểu diễn âm bản của văn bản/thông tin cần in;
- Giấy từ khay được kéo lên cũng được điện cực nạp điện tích trái dấu với điện tích của mực nên hút các hạt mực khỏi trống cảm quang.
- Giấy tiếp tục đi qua trống sấy nóng làm các hạt mực chảy ra và bị ép chặt vào giấy.

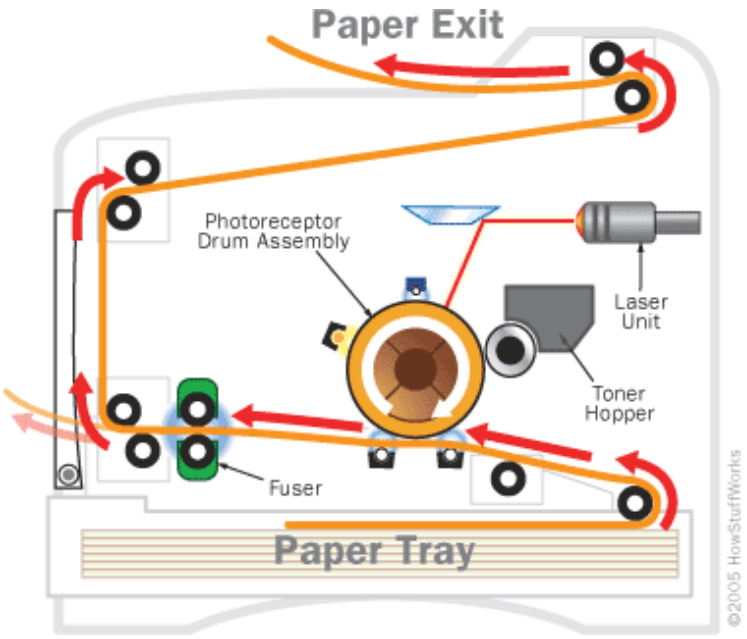


Hình 96 Máy in búa

Hình 97 Máy in kim



Hình 98 Máy in Laser



Hình 99 Nguyên lý in Laser

6.5 CÂU HỎI ÔN TẬP

1. Các thành phần của hệ thống bus và các loại bus.
2. Nguyên lý làm việc của bus PCI.
3. Nguyên lý làm việc của bus PCI Express.
4. Giới thiệu các thiết bị vào ra và các cổng vào ra.
5. Nguyên lý hoạt động của bàn phím.
6. Nguyên lý hoạt động của chuột quang.
7. Nguyên lý hoạt động của màn hình CRT.
8. Nguyên lý hoạt động của màn hình TFT LCD.
9. Nguyên lý hoạt động của máy in laser.

TÀI LIỆU THAM KHẢO

1. Stallings W., *Computer Organization and Architecture: Designing for Performance*, 8th Edition, Prentice – Hall 2009.
2. Mostafa Abd-El-Barr and Hesham El-Rewini, *Fundamentals of Computer Organization and Architecture*, John Wiley & Sons, Inc, 2005.
3. Hennesy J.L. and Patterson D.A., *Computer Architecture. A Quantitative Approach*, Morgan Kaufmann, 4th Edition, 2006.
4. Hồ Khánh Lâm, *Kỹ thuật vi xử lý*, Nhà xuất bản Bưu điện, 2005
5. Trần Quang Vinh, *Cấu trúc máy vi tính*, Nhà xuất bản Giáo dục, 1999.
6. Trang Wikipedia.org, tham khảo năm 2009 và 2010.
7. Trang Howstuffworks.com, tham khảo năm 2009 và 2010.
8. Trang PCGuide.com, tham khảo năm 2009 và 2010.