

Chương 5

THIẾT BỊ NHẬP XUẤT

5.1. DẪN NHẬP

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

Các ngoại vi thông dụng là:

- Màn hình, bàn phím, chuột, máy in, thẻ mạng... là những bộ phận giúp con người sử dụng máy tính dễ dàng.
- Các đĩa từ, băng từ, đĩa quang, các loại thẻ nhớ là những bộ phận lưu trữ thông tin trữ lượng lớn.

Tất cả các ngoại vi đều được nối vào bộ xử lý và bộ nhớ trong bằng một hệ thống dây nối phức tạp vì tính đa dạng của các ngoại vi.

Trong chương này chúng ta tập trung nói đến các bộ phận lưu trữ số liệu có trữ lượng cao (đĩa từ, đĩa quang, băng từ) và sự kết nối các bộ phận này vào máy tính.

5.2. ĐĨA TỪ

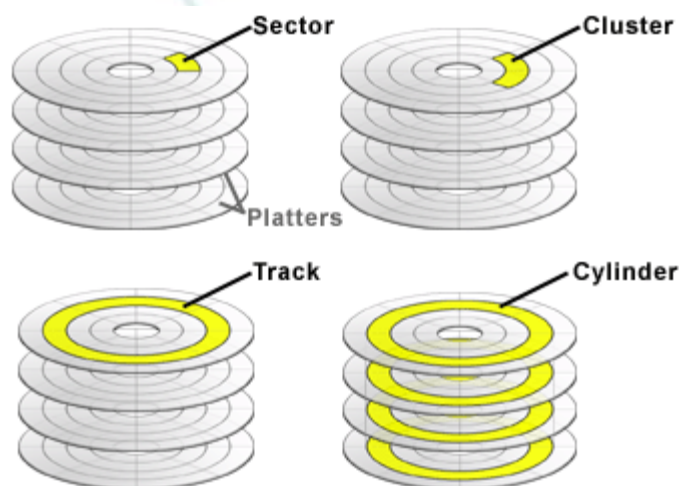
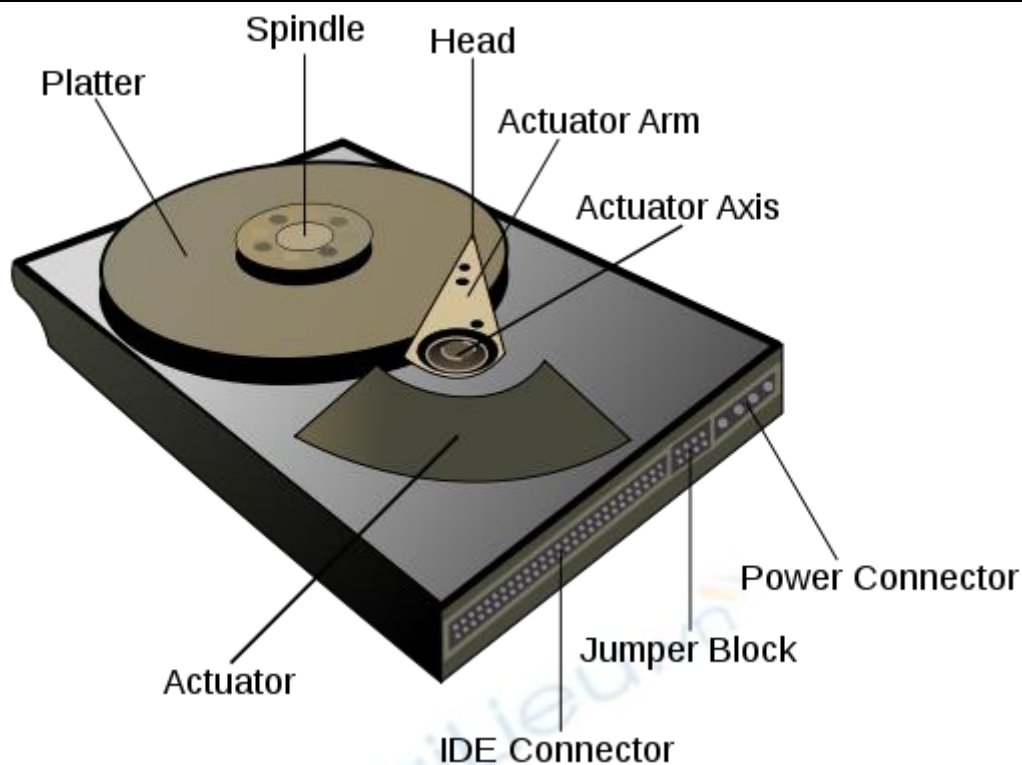
Dù rằng công nghệ mới không ngừng phát minh nhiều loại bộ phận lưu trữ một lượng thông tin lớn nhưng đĩa từ vẫn giữ vị trí quan trọng từ năm 1965. Đĩa từ có hai nhiệm vụ trong máy tính.

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

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

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

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



Hình 5.1. Cấu tạo của một đĩa cứng

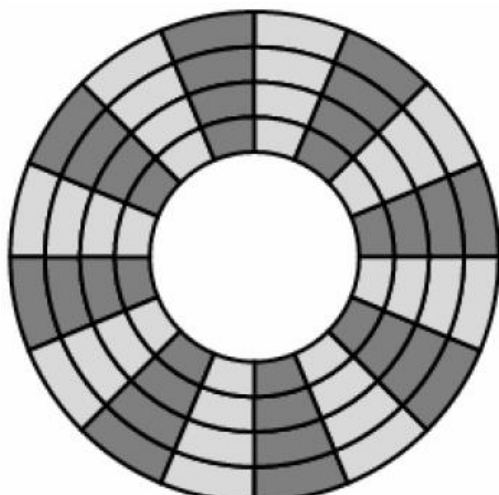
Với công nghệ ghi với mật độ đều, người ta cho ghi nhiều thông tin hơn ở các rãnh xa trục quay. Công nghệ ghi này ngày càng được dùng nhiều với sự ra đời của các chuẩn giao diện thông minh như chuẩn SCSI.

Để đọc hoặc ghi thông tin vào một cung, ta dùng một đầu đọc ghi di động áp vào mỗi mặt của mỗi lớp đĩa. Các đầu đọc/ghi này được gắn chặt vào một thanh làm cho chúng cùng di chuyển trên một đường bán kính của mỗi lớp đĩa và như thế tất cả các đầu này đều ở trên những rãnh có cùng bán kính của các lớp đĩa. Từ “trụ” (cylinder) được dùng để gọi tất cả các rãnh của các lớp đĩa có cùng bán kính và nằm trên một hình trụ.

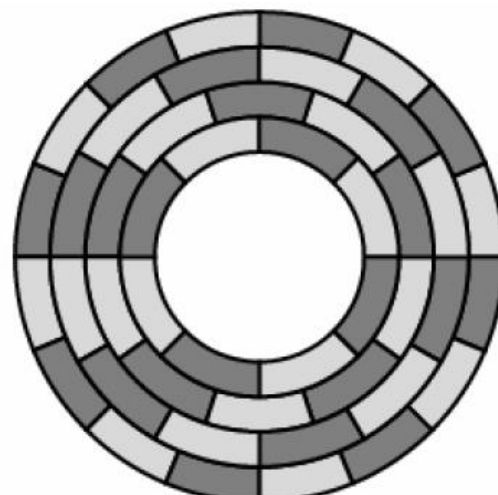
Người ta luôn muốn đọc nhanh đĩa từ nên thông thường ổ đĩa đọc nhiều hơn số dữ liệu cần đọc; người ta nói đây là cách đọc trước. Để quản lý các phức tạp khi kết nối

(hoặc ngưng kết nối) lúc đọc (hoặc ghi) thông tin, và việc đọc trước, ổ đĩa cần có bộ điều khiển đĩa.

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

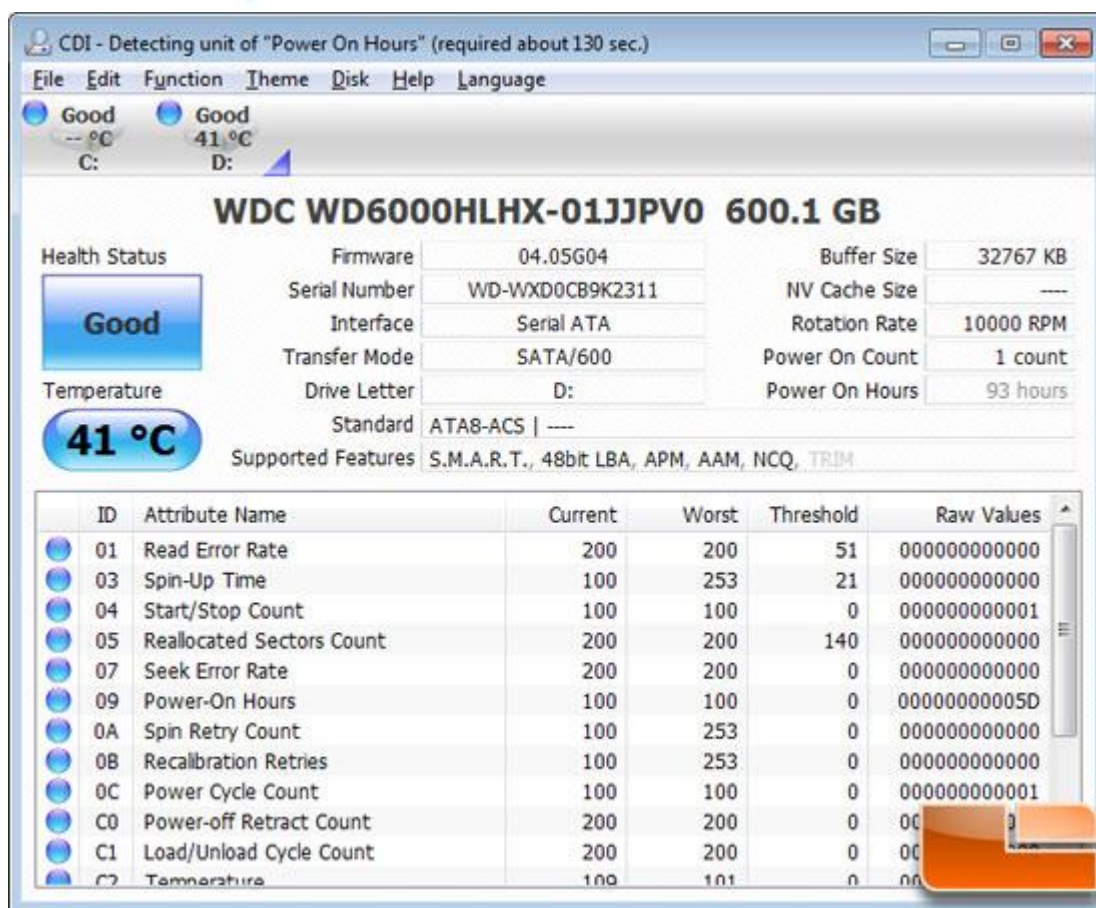


Mật độ ghi không đều



Mật độ ghi đều

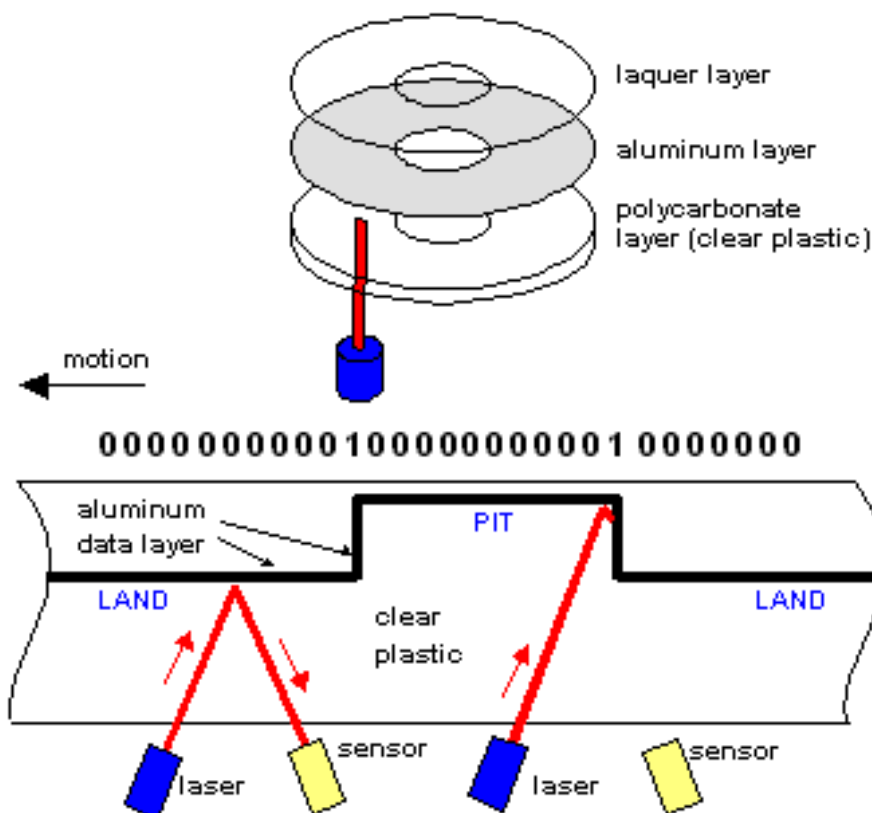
Hình 5.2. Mật độ ghi dữ liệu trên các loại đĩa cứng



Hình 5.3. Thông số kỹ thuật của một đĩa cứng

5.3. ĐĨA QUANG

Các thiết bị lưu trữ quang rất thích hợp cho việc phát hành các sản phẩm văn hoá, sao lưu dữ liệu trên các hệ thống máy tính hiện nay. Ra đời vào năm 1978, đây là sản phẩm của sự hợp tác nghiên cứu giữa hai công ty Sony và Philips trong công nghiệp giải trí. Từ năm 1980 đến nay, công nghiệp đĩa quang phát triển mạnh trong cả hai lĩnh vực giải trí và lưu trữ dữ liệu máy tính. Quá trình đọc thông tin dựa trên sự phản chiếu của các tia laser năng lượng thấp từ lớp lưu trữ dữ liệu. Bộ phận tiếp nhận ánh sáng sẽ nhận biết được những điểm mà tại đó tia laser bị phản xạ mạnh hay biến mất do các vết khắc (pit) trên bề mặt đĩa. Các tia phản xạ mạnh chỉ ra rằng tại điểm đó không có lỗ khắc và điểm này được gọi là điểm nền (land). Bộ nhận ánh sáng trong ổ đĩa thu nhận các tia phản xạ và khuếch tán được khúc xạ từ bề mặt đĩa. Khi các nguồn sáng được thu nhận, bộ vi xử lý sẽ dịch các mẫu sáng thành các bit dữ liệu hay âm thanh. Các lỗ trên CD sâu 0,12 micron và rộng 0,6 micron (1 micron bằng một phần ngàn mm). Các lỗ này được khắc theo một track hình xoắn ốc với khoảng cách 1,6 micron giữa các vòng, khoảng 16.000 track/inch. Các lỗ (pit) và nền (land) kéo dài khoản 0,9 đến 3,3 micron. Track bắt đầu từ phía trong và kết thúc ở phía ngoài theo một đường khép kín các rìa đĩa 5mm. Dữ liệu lưu trên CD thành từng khối, mỗi khối chứa 2.352 byte. Trong đó, 304 byte chứa các thông tin về bit đồng bộ, bit nhận dạng (ID), mã sửa lỗi (ECC), mã phát hiện lỗi (EDC). Còn lại 2.048 byte chứa dữ liệu. Tốc độ đọc chuẩn của CD-ROM là 75 khối/s hay 153.600 byte/s hay 150KB/s (1X).



Hình 5.4. Cơ chế lưu trữ thông tin trên đĩa quang

Dưới đây là một số loại đĩa quang thông dụng.

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

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

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

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

DVD (Digital Video Disk - Digital Versatile Disk): Ra đời phục vụ cho công nghiệp giải trí, đĩa chứa các hình ảnh video được số hoá. Ngày nay, DVD được sử dụng rộng rãi trong các ứng dụng công nghệ thông tin. Kích thước đĩa có hai loại: 8cm và 12 cm. Đĩa DVD có thể chứa dữ liệu trên cả hai mặt đĩa, dung lượng tối đa lên đến 17GB. Các thông số kỹ thuật của đĩa DVD-ROM (loại đĩa chỉ đọc) so với CD-ROM. Tốc độ đọc chuẩn (1X) của DVD là 1.3MB/s (1X của DVD tương đương khoảng 9X của CDROM).

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

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

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

5.4. CÁC LOẠI THẺ NHỚ

Hiện nay, thẻ nhớ là một trong những công nghệ mới nhất được dùng làm thiết bị lưu trữ. Thẻ nhớ flash là một dạng bộ nhớ bán dẫn EEPROM (công nghệ dùng để chế tạo các chip BIOS trên các vi mạch chính), được cấu tạo bởi các hàng và các cột. Mỗi vị trí giao nhau là một ô nhớ gồm có hai transistor, hai transistor này cách nhau bởi một lớp ô-xít mỏng. Một transistor được gọi là floating gate và transistor còn lại được gọi là control gate. Floating gate chỉ có thể nối kết với hàng (word line) thông qua control gate. Khi đường kết nối được thiết lập, bit có giá trị 1. Để chuyển sang giá trị 0 theo một qui trình có tên Fowler-Nordheim tunneling. Tốc độ, yêu cầu về dòng điện cung cấp thấp và đặc biệt với kích thước nhỏ gọn của các loại thẻ nhớ làm cho kiểu bộ nhớ này được dùng rộng rãi trong công nghệ lưu trữ và giải trí hiện nay.

Tên	Ký hiệu	Kích thước
PC Card	PCMCIA	85.6 × 54 × 3.3 mm
CompactFlash I	CF-I	43 × 36 × 3.3 mm
CompactFlash II	CF-II	43 × 36 × 5.5 mm
SmartMedia	SM / SMC	45 × 37 × 0.76 mm
Memory Stick	MS	50.0 × 21.5 × 2.8 mm

Memory Stick Duo	MSD	31.0 × 20.0 × 1.6 mm
Memory Stick PRO Duo	MSPD	31.0 × 20.0 × 1.6 mm
Memory Stick PRO-HG Duo	MSPDX	31.0 × 20.0 × 1.6 mm
Memory Stick Micro M2	M2	15.0 × 12.5 × 1.2 mm
Miniature Card		37 × 45 × 3.5 mm
Multimedia Card	MMC	32 × 24 × 1.5 mm
Reduced Size Multimedia Card	RS-MMC	16 × 24 × 1.5 mm
MMCmicro Card	MMCMicro	12 × 14 × 1.1 mm
P2 card	P2	
Secure Digital card	SD	32 × 24 × 2.1 mm
SxS	SxS	
Universal Flash Storage	UFS	
miniSD card	miniSD	21.5 × 20 × 1.4 mm
microSD card	microSD	15 × 11 × 0.7 mm
xD-Picture Card	xD	20 × 25 × 1.7 mm
Intelligent Stick	iStick	24 × 18 × 2.8 mm
Serial Flash Module	SFM	45 × 15 mm
μ card	μcard	32 × 24 × 1 mm
NT Card	NT NT+	44 × 24 × 2.5 mm
XQD card	XQD	38.5 × 29.8 × 3.8 mm

Bảng 5.1. Các loại thẻ nhớ

5.5. BĂNG TỪ

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

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

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

Có hai cách ghi dữ liệu lên băng từ:

- Ghi nối tiếp: với kỹ thuật ghi xoắn ốc, dữ liệu ghi nối tiếp trên một rãnh của băng từ, khi kết thúc một rãnh, băng từ sẽ quay ngược lại, đầu từ sẽ ghi dữ liệu trên rãnh mới tiếp theo nhưng với hướng ngược lại. Quá trình ghi cứ tiếp diễn cho đến khi đầy băng từ.
- Ghi song song: để tăng tốc độ đọc-ghi dữ liệu trên băng từ, đầu đọc - ghi có thể đọc-ghi một số rãnh kề nhau đồng thời. Dữ liệu vẫn được ghi theo chiều dọc băng từ nhưng các khối dữ liệu được xem như ghi trên các rãnh kề nhau. Số rãnh ghi đồng thời trên băng từ thông thường là 9 rãnh (8 rãnh dữ liệu - 1 byte và một rãnh kiểm tra lỗi).

5.6. CÁC CHUẨN BUS

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

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

5.7. MỘT SỐ BIỆN PHÁP AN TOÀN DỮ LIỆU TRONG VIỆC LƯU TRỮ THÔNG TIN

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

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

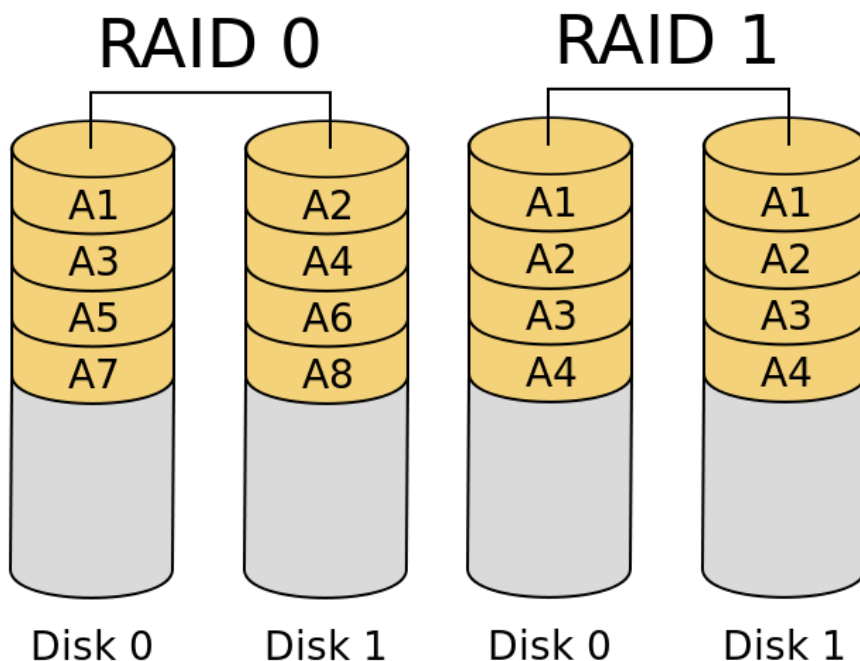
1. RAID là một tập hợp các ổ đĩa cứng (vật lý) được thiết lập theo một kỹ thuật mà hệ điều hành chỉ “nhìn thấy” chỉ là một ổ đĩa (logic) duy nhất.
2. Với cơ chế đọc/ghi thông tin diễn ra trên nhiều đĩa (ghi đan chéo hay soi gương).
3. Trong mảng đĩa có lưu các thông tin kiểm tra lỗi dữ liệu; do đó, dữ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

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

RAID 0: Thực ra, kỹ thuật này không nằm trong số các kỹ thuật có cơ chế an toàn dữ liệu. Khi mảng được thiết lập theo RAID 0, ổ đĩa logic có được (mà hệ điều hành

nhận biết) có dung lượng bằng tổng dung lượng của các ổ đĩa thành viên. Điều này giúp cho người dùng có thể có một ổ đĩa logic có dung lượng lớn hơn rất nhiều so với dung lượng thật của ổ đĩa vật lý cùng thời điểm. Dữ liệu được ghi phân tán trên tất cả các đĩa trong mảng. Đây chính là sự khác biệt so với việc ghi dữ liệu trên các đĩa riêng lẻ bình thường bởi vì thời gian đọc-ghi dữ liệu trên đĩa tỉ lệ nghịch với số đĩa có trong tập hợp (số đĩa trong tập hợp càng nhiều, thời gian đọc – ghi dữ liệu càng nhanh). Tính chất này của RAID 0 thật sự hữu ích trong các ứng dụng yêu cầu nhiều thâm nhập đĩa với dung lượng lớn, tốc độ cao (đa phương tiện, đồ họa,...). Tuy nhiên, như đã nói ở trên, kỹ thuật này không có cơ chế an toàn dữ liệu, nên khi có bất kỳ một hư hỏng nào trên một đĩa thành viên trong mảng cũng sẽ dẫn đến việc mất dữ liệu toàn bộ trong mảng đĩa. Xác suất hư hỏng đĩa tỉ lệ thuận với số lượng đĩa được thiết lập trong RAID 0. RAID 0 có thể được thiết lập bằng phần cứng (RAID controller) hay phần mềm (Stripped Applications).

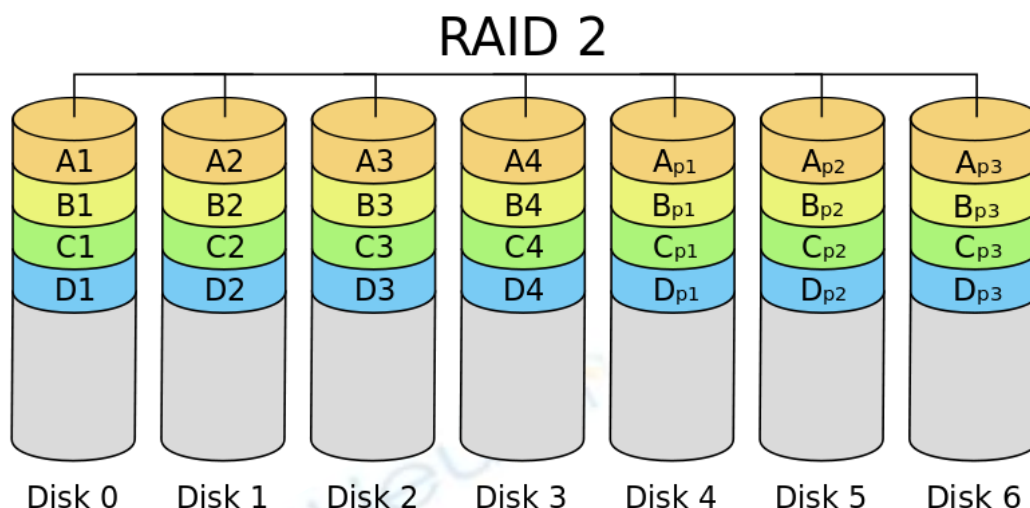
RAID 1 (Mirror - Đĩa gương): Phương cách thông thường tránh mất thông tin khi ổ đĩa bị hư là dùng đĩa gương, tức là dùng 2 đĩa. Khi thông tin được viết vào một đĩa, thì nó cũng được viết vào đĩa gương và như vậy luôn có một bản sao của thông tin. Trong cơ chế này, nếu một trong hai đĩa bị hư thì đĩa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



Hình 5.5. RAID 0 và RAID 1

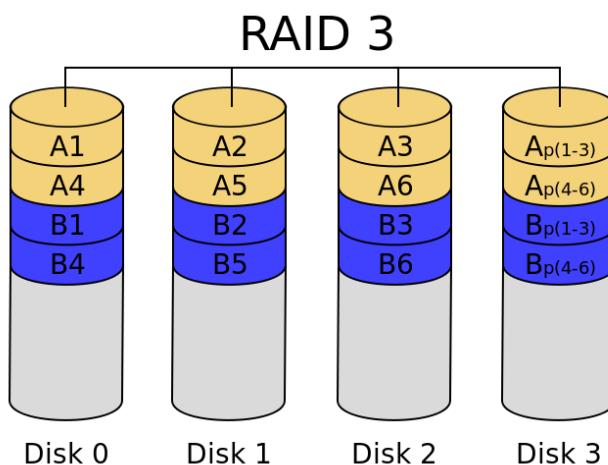
RAID 2: Dùng kỹ thuật truy cập đĩa song song, tất cả các đĩa thành viên trong RAID đều được đọc khi có một yêu cầu từ ngoại vi. Một mã sửa lỗi (ECC) được tính toán dựa vào các dữ liệu được ghi trên đĩa lưu dữ liệu, các bit được mã hóa được lưu trong các đĩa dùng làm đĩa kiểm tra. Khi có một yêu cầu dữ liệu, tất cả các đĩa được truy cập đồng thời. Khi phát hiện có lỗi, bộ điều khiển nhận dạng và sửa lỗi ngay mà không làm giảm thời gian truy cập đĩa. Với một thao tác ghi dữ liệu lên một đĩa, tất cả các đĩa dữ liệu và

đĩa sửa lỗi đều được truy cập để tiến hành thao tác ghi. Thông thường, RAID 2 dùng mã Hamming để thiết lập cơ chế mã hoá, theo đó, dữ liệu được ghi, người ta dùng một bit sửa lỗi và hai bit phát hiện lỗi. RAID 2 thích hợp cho hệ thống yêu cầu giảm thiểu được khả năng xảy ra nhiều đĩa hư hỏng cùng lúc.



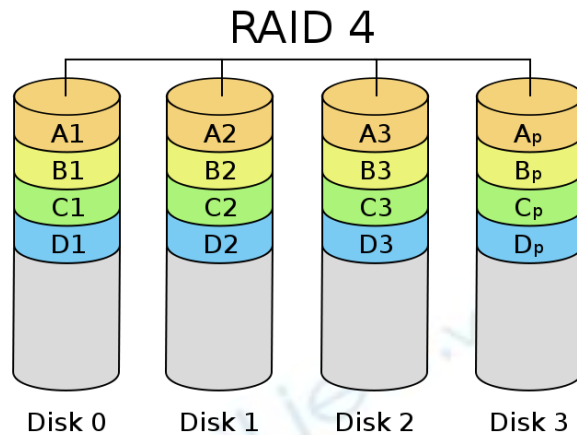
Hình 5.6. RAID 2

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



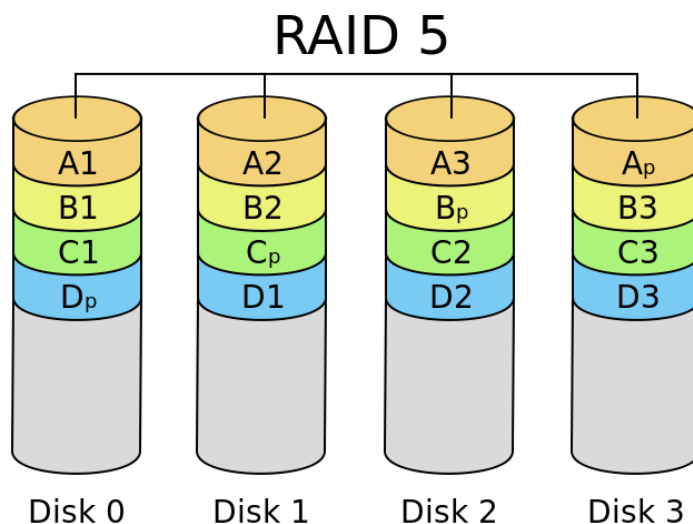
Hình 5.7. RAID 3

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



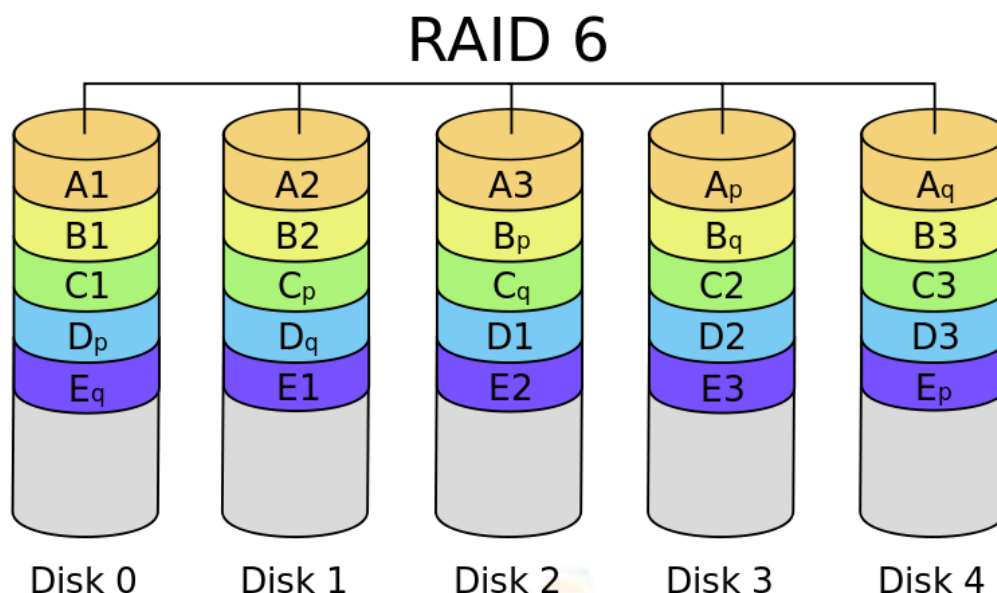
Hình 5.8. RAID 4

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



Hình 5.9. RAID 5

RAID 6: Trong kỹ thuật này, cần có $n+2$ đĩa trong mảng. Trong đó, n đĩa dữ liệu và 2 đĩa riêng biệt để lưu các khối kiểm tra. Một trong hai đĩa kiểm tra dùng cơ chế kiểm tra như trong RAID 4&5, đĩa còn lại kiểm tra độc lập theo một giải thuật kiểm tra. Qua đó, nó có thể phục hồi được dữ liệu ngay cả khi có hai đĩa dữ liệu trong mảng bị hư hỏng.



Hình 5.10. RAID 6

Hiện nay, RAID 0,1,5 được dùng nhiều trong các hệ thống. Các giải pháp RAID trên đây (trừ RAID 6) chỉ đảm bảo an toàn dữ liệu khi có một đĩa trong mảng bị hư hỏng. Ngoài ra, các hư hỏng dữ liệu do phần mềm hay chủ quan của con người không được đề cập trong chương trình. Người dùng cần phải có kiến thức đầy đủ về hệ thống để các hệ thống thông tin hoạt động hiệu quả và an toàn.

CÂU HỎI ÔN TẬP VÀ BÀI TẬP

- Mô tả vận hành của ổ đĩa cứng. Cách lưu trữ thông tin trong ổ đĩa cứng
- Mô tả các biện pháp an toàn trong việc lưu trữ thông tin trong đĩa cứng.
- Nguyên tắc vận hành của đĩa quang. Ưu khuyết điểm của các loại đĩa quang.
- Thông thường có bao nhiêu loại bus? Tại sao phải có các chuẩn cho các bus vào ra?
- Thế nào là chủ nhân của bus? Khi bus có nhiều chủ nhân thì làm thế nào để giải quyết tranh chấp bus?
- Giải thích việc nói rộng dải thông bằng cách sử dụng các gói tin.
- Sự khác biệt giữa bộ xử lý vào ra và bộ xử lý trung tâm của máy tính.
- Tìm hiểu các thông tin liên quan đến ổ cứng
 - Giao tiếp: loại giao tiếp (PATA, SATA, SCSI), các phiên bản, tốc độ
 - Công nghệ S.M.A.R.T
 - Khái niệm Partition, cách tổ chức phân vùng trên đĩa
 - Format: khái niệm, các loại định dạng (FAT, NTFS, ...)
 - Tìm hiểu về ổ cứng SSD
 - Tìm hiểu về ổ cứng lai
- Tìm hiểu các thông tin liên quan đến đĩa quang
 - Cách đọc, ghi thông tin

- b. Các chuẩn CD
- c. Các chuẩn DVD

10. Tìm hiểu các thông tin liên quan đến thẻ nhớ

- a. Các loại thẻ nhớ: CompactFlash Type I/II (CF), Microdrive, Secure Digital (SD), miniSD, Micro SD, MultiMediaCard (MMC), RS-MMC, Micro MMC, Memory Stick, Memory Stick PRO, Memory Stick Duo, Memory Stick PRO Duo, xD
- b. Các tiêu chuẩn tốc độ của thẻ SD (SD Class)

11. Tìm hiểu thông tin các chuẩn bus

- a. PCI
- b. PCI-Express
- c. USB
- d. eSATA
- e. IEEE 1394
- f. Thunderbolt

12. Tìm hiểu thông tin về các mức RAID

Chương 6

TỔNG QUAN VỀ LẬP TRÌNH HỢP NGỮ

6.1. NGÔN NGỮ MÁY

Chương trình là tập các lệnh được đưa vào bộ nhớ cho máy thực hiện, chương trình có nhiều dạng ngôn ngữ khác nhau. Dạng cơ bản nhất mà máy (CPU) có thể hiểu và thực thi được gọi là ngôn ngữ máy (Machine Language). Tùy theo CPU mà ngôn ngữ máy có một dạng nhất định, có nghĩa là mỗi loại CPU có một loại ngôn ngữ riêng.

Ví dụ: một đoạn chương trình ngôn ngữ máy thuộc họ CPU Intel 8006/8088 được biểu diễn dưới dạng số nhị phân là một dãy các byte như sau:

```
10110100 00000010 10000000 11000010 00110010 01010000
```

Đoạn chương trình trên gồm có 3 lệnh có chiều dài lần lượt là 2, 3, 1 byte. Chiều dài của lệnh là do CPU qui định.

Vì là ngôn ngữ riêng của máy nên chương trình viết bằng ngôn ngữ thực hiện rất nhanh và chiếm ít chỗ trong bộ nhớ, tuy nhiên chương trình rất khó viết và rất khó nhớ các lệnh.

6.2. HỢP NGỮ (ASSEMBLY LANGUAGE)

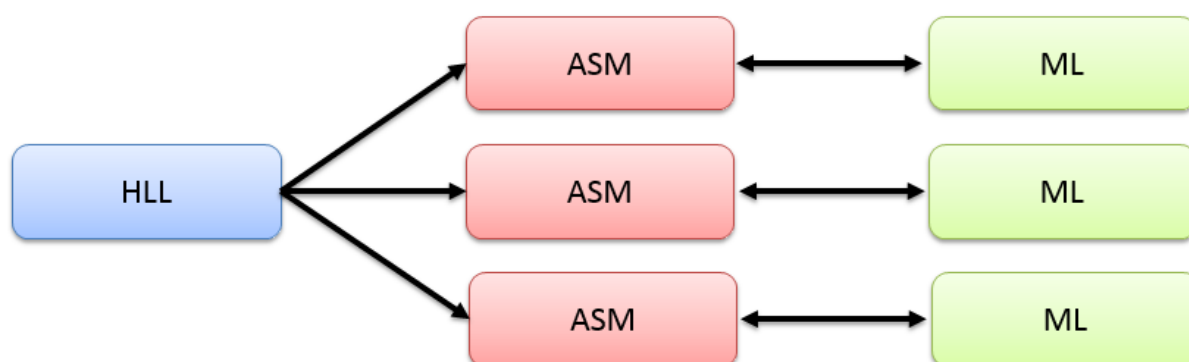
6.2.1. Khái niệm

Hợp ngữ là loại ngôn ngữ giúp lập trình viên viết chương trình dễ dàng hơn thay cho ngôn ngữ máy. Hợp ngữ có dạng như ngôn ngữ máy tức là một lệnh hợp ngữ tương đương với một lệnh ngôn ngữ máy (có thể có một lệnh hợp ngữ tương đương với nhiều lệnh ngôn ngữ máy gọi là lệnh vĩ mô), nhưng khác với ngôn ngữ máy ở chỗ thay vì viết chương trình dưới mã máy (thường là dạng nhị phân), người ta dùng một số ký hiệu tượng trưng cho dễ nhớ.

Ví dụ: Lệnh hợp ngữ: MOV AH, 2

Tương đương với lệnh sau của CPU Intel 8086/8088 dưới dạng mã máy: 10110100

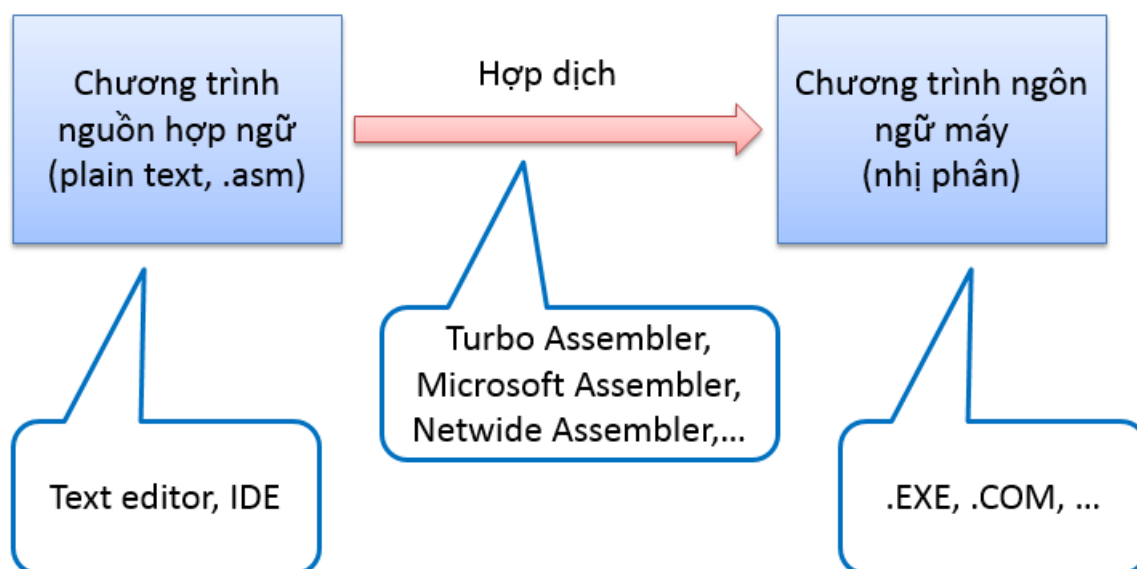
Hợp ngữ được gọi là ngôn ngữ cấp thấp bởi vì nó sát với ngôn ngữ máy theo cấu trúc và chức năng. Mỗi một lệnh hợp ngữ tương ứng với một lệnh ngôn ngữ máy (tương quan một – một). Ngược lại, mỗi một lệnh của ngôn ngữ cấp cao – High Level Language (Pascal, Basic, C, C++, ...) tương ứng với nhiều lệnh mã máy.



Hình 6.1. Phát sinh ngôn ngữ máy bởi các chương trình hợp ngữ và ngôn ngữ cấp cao

6.2.2. Trình hợp dịch (Assembler)

Do máy không thể hiểu được chương trình viết bằng hợp ngữ nên phải qua giai đoạn để dịch chương trình từ hợp ngữ ra ngôn ngữ máy. Chương trình làm nhiệm vụ dịch các chương trình viết bằng hợp ngữ ra ngôn ngữ máy gọi là trình hợp dịch (Assembler). Chương trình viết bằng hợp ngữ gọi là chương trình nguồn và chương trình dưới dạng ngôn ngữ máy được dịch từ chương trình nguồn gọi là chương trình đích.



6.3. ỨNG DỤNG CỦA HỢP NGỮ

Hợp ngữ có sự tương ứng trực tiếp với ngôn ngữ máy nên viết chương trình bằng hợp ngữ có những ưu điểm sau:

- Tốc độ: chương trình viết bằng hợp ngữ luôn đạt tốc độ cao nhất do được viết rất sát phần cứng máy mà nó thực hiện.
- Tiết kiệm bộ nhớ: chương trình hợp ngữ rất nhỏ gọn do rất ít có các lệnh dư thừa và người lập trình kiểm soát trực tiếp dữ liệu trong bộ nhớ.
- Khả năng: lập trình bằng hợp ngữ cho phép ta kiểm soát trực tiếp phần cứng máy tính do đó không gặp phải những giới hạn thường gặp trong ngôn ngữ cấp cao khi sử dụng phần cứng.
- Ngoài ra lập trình bằng hợp ngữ giúp ta tìm hiểu sâu thêm về kiến trúc máy tính, hệ điều hành, các thiết bị phần cứng. Điều này giúp ích cho ta rất nhiều khi viết chương trình, ngay cả khi viết bằng ngôn ngữ cấp cao.

Tuy nhiên, ngày nay rất khó thấy các chương trình được viết hoàn toàn bằng hợp ngữ. Điều đó xuất phát từ những nguyên nhân sau:

- Hợp ngữ rất khó học, rất khó để đọc và hiểu một chương trình hợp ngữ.
- Hợp ngữ đòi hỏi phải tập trung chú ý vào từng chi tiết nhỏ. Do đó phải mất quá nhiều thời gian để viết và bảo trì chương trình.

- Hợp ngữ phụ thuộc vào nền phần cứng mà nó thực hiện do đó một chương trình hợp ngữ viết trên hệ phần cứng này không thể đem sang thực hiện trên một hệ phần cứng khác.
- Tốc độ máy tính và dung lượng bộ nhớ hiện nay rất cao cùng với sự phát triển của các chương trình dịch (compiler), cải tiến các giải thuật cho phép các chương trình có thể chạy nhanh và sinh mã máy hiệu quả mà không cần thiết phải sử dụng hợp ngữ.

Thay vào đó, hợp ngữ thường được dùng để tối ưu hóa một số đoạn nhất định của các trình ứng dụng để tăng tốc và để truy cập phần cứng máy tính (dưới dạng các chương trình con). Hợp ngữ cũng được dùng khi viết các chương trình trong hệ thống nhúng (embedded system) – là những chương trình lưu trữ trong các chip ROM của các thiết bị phần cứng, hoặc được dùng để viết các chương trình điều khiển thiết bị (device driver) của hệ điều hành.

6.4. CÁC KIỂU DỮ LIỆU

Mục đích chính của máy tính là lưu trữ, truy tìm và xử lý dữ liệu. Do đó ta sẽ khảo sát cách thức máy tính lưu trữ dữ liệu trong bộ nhớ.

6.4.1. Số học

Hầu hết các hệ thống máy tính hiện nay đều không sử dụng hệ thống số thập phân để biểu diễn các con số. Thay vào đó là hệ thống số nhị phân (binary), trong hệ thống này mỗi ký số chỉ có thể là 0 hay 1, tương ứng với trạng thái on/off của mạch điện tử.

Ghi chú:

- Trong hợp ngữ, để biểu diễn hằng số ở hệ 2, 16 ta ghi giá trị kèm theo ký tự b/B, h/H). Ví dụ: 11001010b, 3Fh
- Nếu hằng hệ 16 bắt đầu bằng ký tự thì phải thêm 0 vào trước. Ví dụ: 0F8H

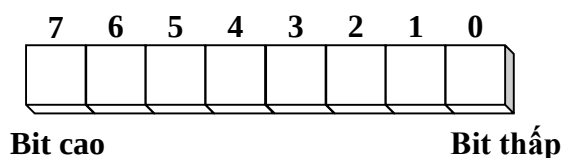
Bit

Biểu diễn một ký số nhị phân 0 hoặc 1.

Byte

Có chiều dài là 8 bit, các bit được đánh số từ 0 đến 7 từ phải sang trái, bit 7 là bit cao, bit 0 là bit thấp, dùng để biểu diễn:

- Một số nguyên không dấu có giá trị từ 0 đến 255 (nhị phân không dấu).
- Một số nguyên có dấu có giá trị từ -128 đến 127 (nhị phân có dấu theo kiểu bù hai).

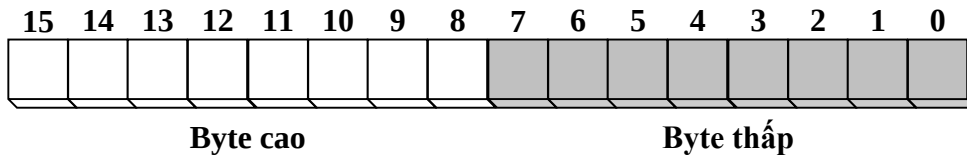


Word

Có chiều dài 16 bit, các bit được đánh số từ 0 đến 15, bit 15 là bit cao, bit 0 là bit thấp, byte bên phải gọi là byte thấp, byte bên trái gọi là byte cao dùng để biểu diễn:

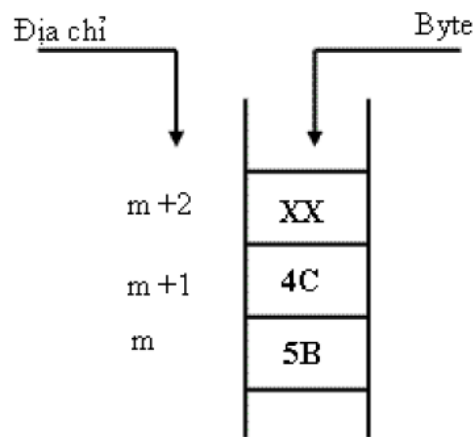
- Một số nguyên không dấu có giá trị từ 0 đến 65535.

- Một số nguyên có dấu có giá trị từ -32768 đến +32767.



Khi lưu trữ trong bộ nhớ, byte thấp sẽ lưu ở địa chỉ thấp, byte cao được lưu ở địa chỉ cao.

Ví dụ: word 4C5B được lưu trong bộ nhớ tại địa chỉ bắt đầu là m như sau: byte thấp 5B được lưu tại ô nhớ m, byte cao 4C được lưu tại ô nhớ m+1.



Double Word

Có chiều dài 32 bit, các bit được đánh số từ 0 đến 31, bao gồm 2 word 16 bit. Word thấp chiếm các bit từ 0 đến 15, word cao chiếm các bit từ 16 đến 31. Double word có thể chứa một số nguyên có dấu từ -2147483648 đến 2147483647 hoặc một số nguyên không dấu từ 0 đến 4294967295.

Số BCD không nén (Binary – Coded Decimal unpacked)

Gồm nhiều byte, mỗi byte biểu diễn một ký số thập phân.

Ví dụ: số 1234 thập phân lưu trong bộ nhớ theo dạng BCD không nén sẽ chiếm 4 byte liên tiếp nhau có giá trị lần lượt là 1, 2, 3, 4.

Số BCD nén (Binary – Coded Decimal packed)

Gồm nhiều byte, mỗi byte biểu diễn hai ký số thập phân, mỗi 4 bit (nibble) biểu diễn một ký số thập phân.

Ví dụ: số 1234 thập phân lưu trong bộ nhớ theo dạng BCD nén sẽ chiếm 2 byte liên tiếp nhau có giá trị như sau:

0001 0010 0011 0100

6.4.2. Ký tự

Các máy tính chỉ lưu trữ các số nhị phân, do đó để lưu trữ các ký tự máy tính phải sử dụng một lược đồ mã hóa ký tự phiên dịch các ký tự thành các con số và ngược lại. Bảng mã được sử dụng phổ biến nhất của các máy tính tên là ASCII (American Standard Code for Information Interchange). Trong bảng mã ASCII, mỗi ký tự chiếm 1 byte và

được gán một trị số duy nhất, kể cả các ký tự điều khiển dùng khi in hoặc truyền dữ liệu giữa các máy tính.

Ví dụ: các ký tự A, B, a trong bảng mã ASCII có các trị số lần lượt là 65, 66, 97

Một chuỗi ký tự là một dãy các byte liên tiếp, mỗi byte chứa mã ASCII biểu diễn cho một ký tự trong chuỗi. Chuỗi được lưu trong bộ nhớ từ địa chỉ thấp đến địa chỉ cao.

Hằng chuỗi nằm giữa hai dấu nháy đơn hay nháy kép.

Ví dụ: chuỗi ký tự “Aba” lưu trong bộ nhớ gồm 3 byte liên tiếp nhau như sau:

65	66	97
----	----	----

6.4.3. Chú ý

Trong lĩnh vực lập trình ngôn ngữ cấp cao, ta có khuynh hướng xem dữ liệu và mã lệnh là khác nhau. Tuy nhiên, về mặt vật lý, cả hai đều là những chuỗi bit nhị phân trong bộ nhớ. Ví dụ: chuỗi bit 010000110101110010001001 có thể là 3 byte biểu diễn cho một chuỗi, 3 byte nhớ biểu diễn cho 3 số nguyên hoặc cũng có thể là 3 byte mã của một lệnh máy.

Mặt khác, trong hợp ngữ sự phân biệt giữa các kiểu dữ liệu cũng chỉ có tính tương đối. Ví dụ: một chuỗi bit 1010001100010010 có thể xem là byte thấp và byte cao của 1 word, 2 byte số nguyên riêng lẻ, hoặc là một chuỗi 2 ký tự.

Máy tính không thể tự phân biệt được những trường hợp trên. Do đó khi lập trình hợp ngữ, trách nhiệm của lập trình viên là phải giữ tách biệt được mã và dữ liệu đồng thời kiểm soát được dữ liệu trong bộ nhớ - đây là phần việc đã được các ngôn ngữ lập trình cấp cao tự thực hiện.

CÂU HỎI ÔN TẬP VÀ BÀI TẬP

1. Nêu hai kiểu chương trình hoặc ứng dụng thích hợp với hợp ngữ hơn so với ngôn ngữ cấp cao.
2. Cho biết ý nghĩa của môi quan hệ một – nhiều khi so sánh ngôn ngữ cấp cao với ngôn ngữ máy.
3. Tại sao việc tìm hiểu hệ điều hành lại quan trọng khi ta nghiên cứu về hợp ngữ?
4. Tính năng kiểm tra dữ liệu tỏ ra mạnh hơn trong hợp ngữ hay trong ngôn ngữ cấp cao?
5. Có phải chương trình viết trên bộ xử lý Intel 8086/8088 đều chạy được trên các bộ xử lý 386, 486, Pentium và ngược lại? Tại sao?
6. Tại sao hợp ngữ không được dùng khi viết các ứng dụng lớn?
7. Thực hiện chuyển đổi số nguyên giữa các cơ hệ thập phân, nhị phân, thập lục phân.
8. Sử dụng ngôn ngữ cấp cao (Pascal, C, ...) viết các chương trình chuyển đổi số giữa các cơ hệ thập phân, nhị phân, thập lục phân.
9. Thực hiện các phép tính cơ bản (cộng, trừ) ở hệ nhị phân, thập lục phân.
10. Tính số lượng bit nhị phân cần thiết để biểu diễn một số nguyên.

TaiLieu.vn

Chương 7

TỔ CHỨC BỘ XỬ LÝ INTEL 8086/8088

7.1. TỔ CHỨC BỘ NHỚ

Bộ nhớ chính của 8086/8088 là một dãy các byte liên tiếp nhau được đánh số từ 0 đến $20^{20}-1$ (00000h đến FFFFh). Số thứ tự đó gọi là địa chỉ (dài 20 bit) và tổng cộng bộ nhớ có 1MB.

Để tận dụng bộ nhớ và thuận lợi trong xử lý, người ta chia bộ nhớ thành từng đoạn (segment), mỗi đoạn bắt đầu từ địa chỉ chia hết cho 16 và có chiều dài tối đa 64KB. Lúc đó vị trí một byte trong bộ nhớ được xác định bằng một segment:offset (gọi là địa chỉ tương đối, vị trí thực trong bộ nhớ gọi là địa chỉ tuyệt đối). Segment và offset là một số nhị phân 16 bit có trị từ 0 đến FFFFh.

Như vậy địa chỉ của một ô nhớ bất kỳ trong bộ nhớ 1MB được xác định bằng các tổ hợp địa chỉ 16 bit địa chỉ segment và 16 bit địa chỉ offset.

Địa chỉ vật lý

Là đại lượng có chiều dài 20 bit xác định vị trí của một ô nhớ trong 1MB bộ nhớ.

Địa chỉ logic

Gồm 2 phần, mỗi phần có chiều dài 16 bit lần lượt biểu diễn địa chỉ segment và offset của một ô nhớ. Ký hiệu:

Segment : Offset

Ví dụ: với địa chỉ vật lý FFFFEh ta có thể định vị theo địa chỉ logic là F000:FFFEh hoặc FFFF:000Eh

Người lập trình chỉ có thể truy xuất các ô nhớ thông qua địa chỉ logic, còn địa chỉ vật lý chỉ dành riêng cho CPU sử dụng.

Cách chuyển từ địa chỉ logic sang địa chỉ vật lý

Bằng cách dịch địa chỉ segment sang trái 4 bit rồi cộng với địa chỉ offset.

Ví dụ: ta có địa chỉ logic 1134:1032h thì địa chỉ vật lý là:

1134h dịch trái 4 bit $\rightarrow$ 11340h

11340h + 1032h $\rightarrow$ 12372h

Một ô nhớ chỉ có duy nhất một địa chỉ vật lý, nhưng ta có thể dùng nhiều địa chỉ logic khác nhau để định vị cùng một ô nhớ.

Ví dụ: hai địa chỉ logic sau:

35B7:1000h

36B7:0000h

$\rightarrow$ Biểu thị cùng một địa chỉ vật lý 36B70h

Sự phân bố các đoạn trong bộ nhớ

Các đoạn bộ nhớ có thể chồng lên nhau, có thể nối tiếp nhau, có thể tách rời nhau.

Việc phân bộ nhớ thành từng đoạn làm cho việc tổ chức chương trình và dữ liệu khá mềm dẻo. Chương trình không cần phải viết như một dãy liên tục các chỉ thị (instruction) và dữ liệu mà có thể là các đơn thể (module) gồm mã và dữ liệu.

Dữ liệu cũng có thể tổ chức thành các cấu trúc dữ liệu khác nhau: dữ liệu dùng riêng, dữ liệu dùng chung với các chương trình khác trong hệ thống. Mỗi đơn thể mã và dữ liệu này có thể có kích thước khác nhau.

7.2. TỔ CHỨC THANH GHI

Thanh ghi (register) là nơi dữ liệu bên trong CPU tùy theo độ dài 8 bit hay 16 bit và tùy theo chức năng khi đó thanh ghi được dùng để chứa dữ liệu sẽ thao tác hoặc kết quả các phép tính hoặc các địa chỉ dùng để định vị ô nhớ khi cần thiết. Có tất cả 14 thanh ghi, mỗi thanh ghi dài 16 bit chia thành 5 nhóm:

7.2.1. Nhóm thanh ghi đoạn (segment register)

Gồm 4 thanh ghi: đoạn mã CS, đoạn dữ liệu DS, đoạn bổ sung ES và đoạn stack SS. Đó là những thanh ghi chứa địa chỉ segment của các ô nhớ khi cần truy xuất.

	15	0
CS		
DS		
SS		
ES		

Thanh ghi đoạn mã CS (Code Segment): Lưu địa chỉ segment chứa chương trình ngôn ngữ máy đang thực thi.

Thanh ghi đoạn dữ liệu DS (Data Segment): Lưu địa chỉ segment của đoạn dữ liệu trong chương trình.

Thanh ghi đoạn bổ sung ES (Extra Segment): Lưu địa chỉ segment của đoạn dữ liệu bổ sung.

Thanh ghi đoạn Stack SS (Stack Segment): Lưu địa chỉ segment của đoạn stack.

Bốn thanh ghi này có thể truy xuất dữ liệu trên bốn đoạn khác nhau và một chương trình chỉ có thể sử dụng cùng một lúc tối đa bốn đoạn

Trên CPU 80386 còn có hai thanh ghi chức năng tương tự như ES là FS và GS.

7.2.2. Nhóm thanh ghi đa dụng (general register)

Gồm bốn thanh ghi AX, BX, CX, DX. Các thanh ghi này có thể xem như một thanh ghi 16 bit hoặc hai thanh ghi mỗi thanh ghi 8 bit

Đối với CPU 80386 các thanh ghi đa dụng có thể kéo dài đến 32 bit tạo thành thanh ghi EAX, EBX, ECX, EDX.