

***Chương 1:* TỔNG QUAN VỀ CẤU TRÚC MÁY TÍNH VÀ THIẾT BỊ NGOẠI VI**

TS. Phạm Văn Thành
(phamvanthanh@hus.edu.vn)

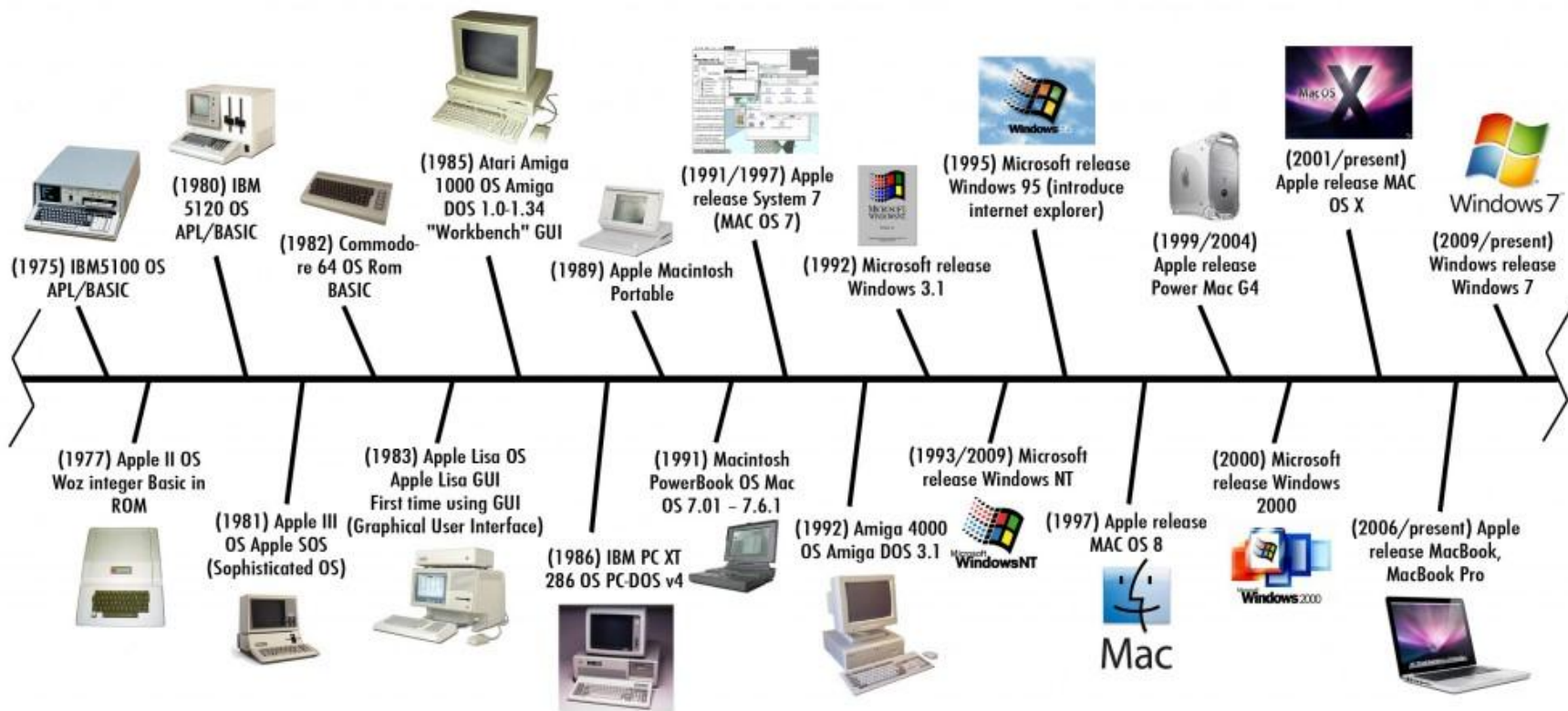
Nội dung

1. Lịch sử phát triển của máy vi tính điện tử.
2. Phân loại máy vi tính.
3. Các thành phần cơ bản của máy vi tính PC (personal computer).

http://en.wikipedia.org/wiki/Personal_computer

<http://en.wikipedia.org/wiki/Computer>

1. Lịch sử phát triển của máy vi tính điện tử.



1946-1959

1. Lịch sử.

- Chế tạo: bóng đèn điện tử chân không, tiêu thụ điện năng lớn.
- Tốc độ: vài nghìn phép tính trên một giây.
- Ngôn ngữ lập trình: các ngôn ngữ ký hiệu, ngôn ngữ máy.



ENIAC (Electronic Numerical Integrator And Computer)

17,468 vacuum tubes

7,200 crystal diodes

1,500 relays

70,000 resistors

10,000 capacitors

2.4 m × 0.9 m × 30 m

150kW

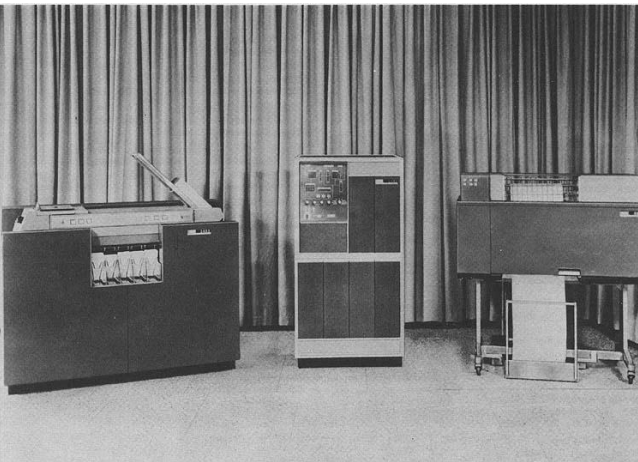
\$500,000 (~\$6,000,000 today)

5,000 simple addition or subtraction

1959-1965

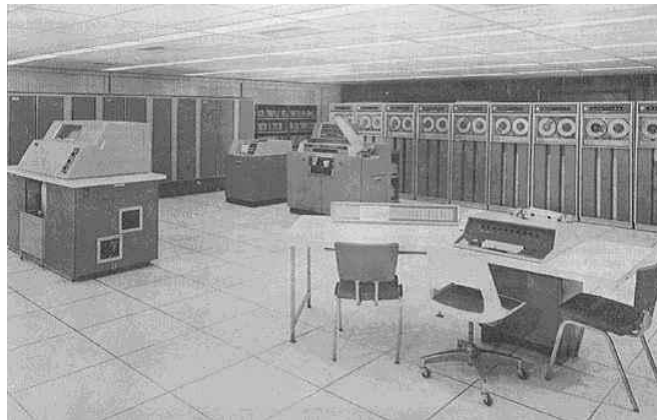
1. Lịch sử.

- Chế tạo: các chất bán dẫn, tiêu thụ điện năng ít.
- Tốc độ: hàng vạn phép tính trên một giây.
- Trang bị bộ nhớ trong lớn, các thiết bị ngoại vi bắt đầu phát triển như màn hình đen trắng, bàn phím.
- Ngôn ngữ lập trình: Fortran, Cobol,....



IBM 1401

(http://en.wikipedia.org/wiki/IBM_1401)



HoneyWell 400



Univac III

(http://en.wikipedia.org/wiki/UNIVAC_III)

1965-1975

1. Lịch sử.

- Chế tạo: Các mạch IC-Integrated circuit (Bộ vi xử lý đầu tiên của Intel là 4004)
- Tốc độ: tốc độ tính toán hàng triệu phép tính trên giây.
- Ngôn ngữ lập trình và hệ điều hành phát triển mạnh.



IBM360



UNIVAC 1108

1. Lịch sử.

1975-1980

- IC công suất lớn ra đời và là cơ sở để các vi xử lý **Intel 8080, 8085** ra đời (8 bit Bus dữ liệu và 16 bit Bus địa chỉ), **Intel 8086** (16 bit Bus dữ liệu và 20 bit Bus địa chỉ, 1st generation of x86 family).
- Tốc độ: tốc độ xử lý lên tới hàng triệu phép tính trên giây.
- Có bộ nhớ trong lớn, thiết bị ngoại vi phát triển mạnh: màn hình màu, bàn phím, máy in, ổ đĩa CD_ROM,...
- Hệ điều hành DOS, ngôn ngữ trí tuệ nhân tạo Prolog (Programming Logic)



IBM

1982-1983

1. Lịch sử.

- Máy tính cá nhân PC XT của IBM (Intel 8088 4.77MHz, có 8 bit Bus dữ liệu và 20 bit Bus địa chỉ).
- PC XT là máy vi tính đầu tiên được trang bị ổ cứng 10 MB Seagate ST-412, đĩa mềm 360KB và bộ nhớ trong 256KB, có khe cắm mở rộng ISA 8 bit tăng khả năng kết nối các thành phần ngoại vi
- Hệ điều hành: IBM BASIC / PC DOS 2.0-3.20 / SCO (Santa Cruz Operation)



IBM PC/XT (model 5160)

http://en.wikipedia.org/wiki/IBM_5160

1. Lịch sử.

1984-1987

- Máy tính cá nhân PC AT của IBM (Intel 80286, 6~8 MHz, 16 bit Bus dữ liệu và 24 bit Bus địa chỉ, quản lý bộ nhớ 16MB).
- Ổ cứng 20 MB hard disk drive, có khe cắm mở rộng ISA 16 bit
- Hệ điều hành: PC DOS 3.x/ Window 1.0



IBM AT (model 5170)

http://en.wikipedia.org/wiki/IBM_Personal_Computer/AT

1. Lịch sử.

1987-1989

- Ra đời máy tính sử dụng **Intel 80386** (33Mhz, 32 bit Bus dữ liệu, 32 bit Bus địa chỉ), khả năng quản lý bộ nhớ lên tới 4 GB, trang bị khe cắm mở rộng EISA(Extended Industry Standard Architecture) 32 bit,
- Có nhiều thiết bị ngoại vi phục vụ cho các nhu cầu giải trí, xem phim, nghe nhạc, trò chơi, đồ hoạ,....
- **Chất lượng chưa cao.**



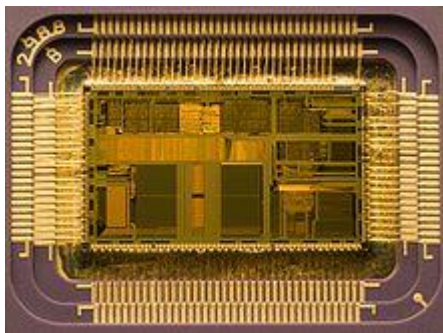
Intel 80386

http://en.wikipedia.org/wiki/Intel_80386

1. Lịch sử.

1990-1992

- Ra đời chiếc máy vi tính sử dụng **Intel 80486** (32 bit Bus dữ liệu và 32 bit Bus địa chỉ), được bổ sung thêm 8 KB bộ nhớ đệm mã lệnh vì vậy tốc độ nhanh hơn rất nhiều lên tới 66Mhz.
- Trang bị Bus cục bộ **VESA** (Video Electronics Standard Association) đáp ứng các nhu cầu về tốc độ và độ phân giải của các card video, truyền dữ liệu của các máy vi tính thông qua mạng.
- Hệ điều hành: **Windows 3.0**.



Intel 80486

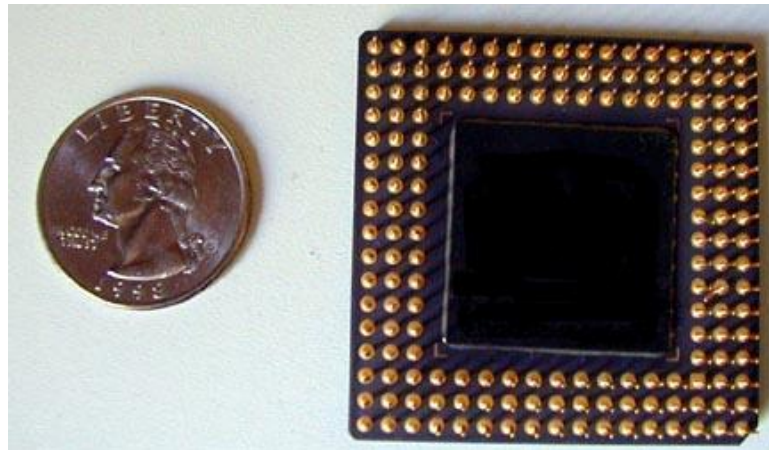
http://en.wikipedia.org/wiki/Intel_80486



1993-1994

1. Lịch sử.

- Ra đời chiếc máy vi tính sử dụng Intel 80586 hay pentium (64 bit Bus dữ liệu và 32 bit Bus địa chỉ), 8 KB bộ nhớ đệm cho mã lệnh và 8 KB bộ nhớ đệm cho dữ liệu, tốc độ lên tới 200Mhz.
- Trang bị Bus PCI (Peripheral Components Interconnect) 32 bit, có tốc truyền gấp 4 lần Bus ISA và là Bus hỗ trợ chức năng ‘cắm là chạy’ (Plug and Play).
- Hệ điều hành: Windows NT là hệ điều hành tách rời khỏi DOS.



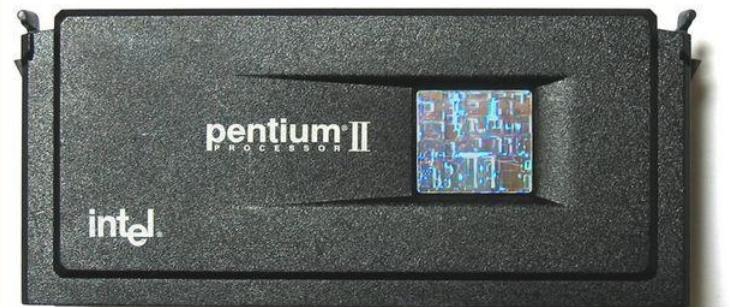
<http://en.wikipedia.org/wiki/Pentium>

<http://www.computermuseum.li/Testpage/Chip-Pentium-1993.htm>

1995-1999

1. Lịch sử.

- Các thế hệ máy vi tính sử dụng Intel Pentium MMX, Pentium Pro và Pentium II (133 Mhz đến 450 Mhz).
 - + **Pentium MMX**: mục đích về đa phương diện như đồ họa, chơi game, xem phim, nghe nhạc,...
 - + **Pentium Pro**: máy chủ và máy trạm làm việc, thích hợp cho việc xử lý tính toán có độ phức tạp cao.
 - + **Pentium II**: Kết hợp cả 2 loại trên, **có khả năng xử lý đồ họa 3D**.
- Trang bị Bus tuần tự đa năng **USB** (Universal Serial Bus) tốc độ nhanh gấp 10 lần giao diện song song, 100 lần giao diện tuần tự, có khả năng kết nối được 127 thiết bị ngoại vi thông qua hệ thống cáp USB.
- Hệ điều hành: Windows 95, Windows 98.



1. Lịch sử.

Năm 1999

- Pentium III (Từ 450Mhz tới 1,2 Ghz).
 - + Hỗ trợ mạnh về khả năng biểu diễn không gian 3 chiều,
 - + Khả năng nhận biết và tổng hợp tiếng nói
 - + Có thể làm nhiều công việc cùng một lúc



1. Lịch sử.

Năm 2000 ~

- Pentium IV (Từ 1.4 Ghz tới 4.4 Ghz).
- Pentium Dual core
- Pentium Core 2 Duo
- Core i3, i5, i7
- Hệ điều hành: Windows, Mac OS, hệ điều hành cho smash phone,...



2. Phân loại máy vi tính.

- a) Siêu máy tính (Supercomputer).
- b) Máy vi tính lớn (Mainframe computer).
- c) Máy vi tính con (Minicomputer).
- d) Máy vi tính (Microcomputer – personal computer).

2. Phân loại

a) Siêu máy tính (Supercomputer).

- Một **siêu máy tính** là một máy tính vượt trội trong khả năng và tốc độ xử lý, sử dụng hàng nghìn bộ vi xử lý.
- Có tốc độ xử lý hàng nghìn **teraflop** (một teraflop tương đương với hiệu suất một nghìn tỷ phép tính/giây).
- Giá thành rất cao.
- Sử dụng trong mô phỏng tính toán những bài toán phức tạp.

Bảng xếp hạng những máy tính nhanh nhất thế giới hàng năm (top500.org)

2002	NEC Earth Simulator	35.86 TFLOPS	Earth Simulator Center, Yokohama, Japan
2004	IBM Blue Gene/L	70.72 TFLOPS	DoE/IBM Rochester, Minnesota, USA
2005		136.8 TFLOPS	DoE/U.S. National Nuclear Security Administration, Lawrence, California, USA
		280,6 TFLOPS	
2007		478.2 TFLOPS	
2008	IBM Roadrunner	1.026 PFLOPS	DoE-Los Alamos National Laboratory, New Mexico, USA
2009	Cray Jaguar	1.759 PFLOPS	Oak Ridge National Laboratory, New Mexico, USA
		2.331 PFLOPS	
2010	Thiên Hà 1A	2.507 PFLOPS	Trung tâm siêu máy tính Thiên Tân, Thiên Tân, Trung Quốc
2011	Fujitsu K computer	10.51 PFLOPS	RIKEN, Wakō, Saitama, Nhật Bản

Name	FLOPS
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}
gigaFLOPS	10^9
megaFLOPS	10^6
kiloFLOPS	10^3

2. Phân loại

b) Máy vi tính lớn (Mainframe computer).

- Đây là loại máy vi tính có kích thước rất lớn, tốc độ tính toán rất nhanh, Bus dữ liệu lên tới 256 bit và đặc biệt có bộ nhớ cực lớn.
- Loại máy vi tính này chủ yếu được sử dụng trong hệ thống ngân hàng, hệ thống vũ trụ, hay trong quân sự tính toán những bài toán quy mô lớn.
- **Giá thành cao và công kênh.**

http://en.wikipedia.org/wiki/IBM_System_z9



IBM System z9

2. Phân loại

c) Máy vi tính con (Minicomputer).

- Là dạng thu nhỏ về kích thước cũng như tính năng của máy vi tính lớn.
- Tốc độ xử lý của nó chậm hơn máy vi tính lớn và dung lượng bộ nhớ cũng nhỏ hơn máy vi tính lớn.
- Bus dữ liệu 32 bit hoặc 64 bit.



Digital Equipment Corporation
(DEC) PDP-8

2. Phân loại

d) Máy vi tính (Microcomputer – personal computer)

- Đây là loại máy vi tính được thiết kế gọn nhẹ phù hợp với mục đích sử dụng cá nhân.
- Bus dữ liệu từ 4 bit đến 64 bit, có tốc độ xử lý không thua kém gì máy vi tính con



PC

Laptop
Tablet



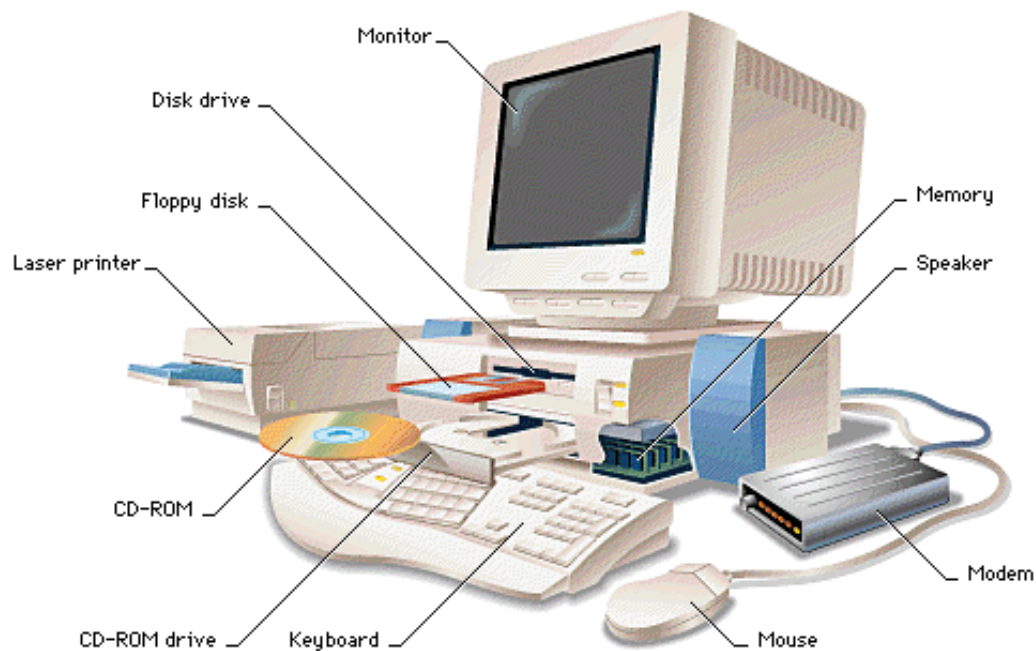
Smart phone



game consoles

3. Các thành phần cơ bản của máy vi tính PC (personal computer).

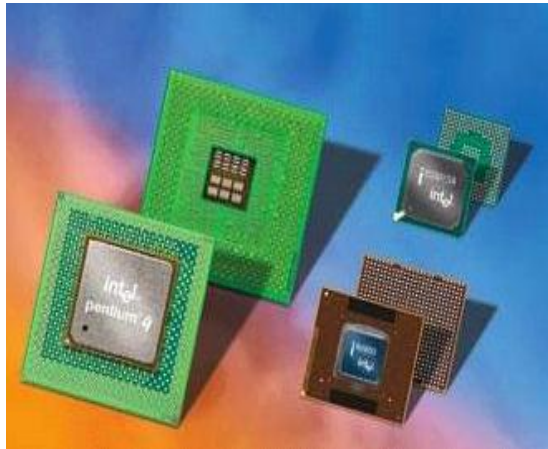
- a) **CPU (Central Processing Unit)**: Bộ vi xử lý trung tâm.
- b) **Bộ nhớ (Memory)**: dùng để lưu trữ thông tin.
- c) **Thiết bị ngoại vi**
- d) **Hệ thống Bus**: được sử dụng để kết nối các thành phần trên lại với nhau.



3. Thành phần cơ bản

a) CPU (Central Processing Unit)

- CPU là thành phần đóng vai trò quan trọng nhất và được xem như bộ não của máy vi tính.
- Thu nhận, xử lý và thực hiện một cách trực tiếp hay gián tiếp các lệnh mà chúng ta yêu cầu.



CPU dạng hình vuông



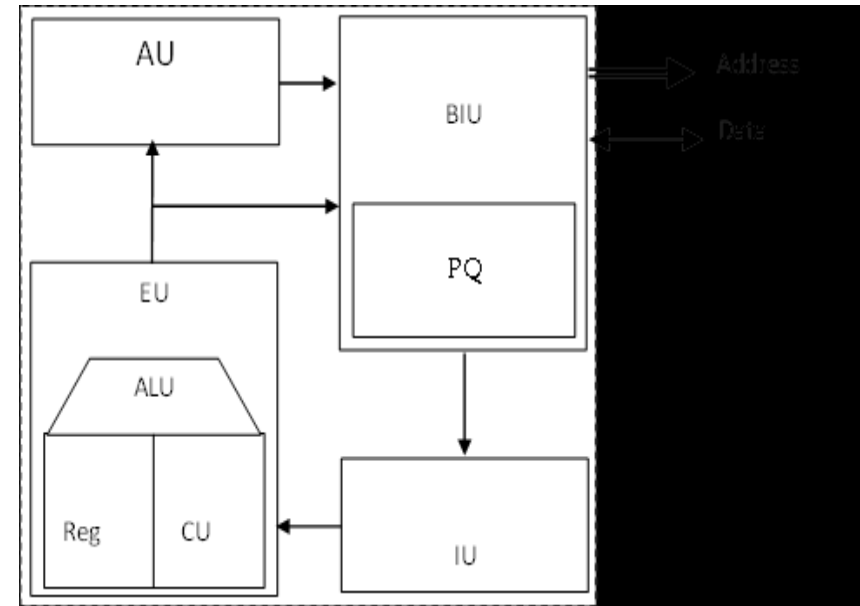
CPU dạng thẻ cắm

3. Thành phần cơ bản

a) CPU (Central Processing Unit)

- Cấu tạo.

- ✓ AU (Address Unit): Khôi định địa chỉ.
- ✓ EU (Execution Unit): Khôi thực hiện lệnh.
- ✓ ALU (Arithmetic and Logic Unit): Khôi số học và logic.
- ✓ CU (Control Unit): Khôi điều khiển.
- ✓ BIU (Bus Interface Unit): Khôi giao diện Bus.
- ✓ PQ (Prefetch Queue): Hàng nhận lệnh trước (hay còn gọi là bộ nhớ đệm sơ cấp Cache L1).
- ✓ IU (Instruction Unit): Khôi lệnh.
- ✓ Reg (Registers): Các thanh ghi.



3. Thành phần cơ bản

a) CPU (Central Processing Unit)

- Bus BIU:
 - Khôi thực hiện tất cả các chức năng về Bus cho EU, chịu trách nhiệm đưa tín hiệu địa chỉ ra ngoài Bus và trao đổi dữ liệu với Bus.
 - Bên trong BIU có hàng nhận lệnh trước PQ (hay còn gọi là bộ nhớ đệm sơ cấp cache L1), sử dụng để chứa các mã lệnh được lấy trước từ bộ nhớ chính, các mã lệnh này nằm chờ EU xử lý.
 - PQ làm tăng tốc độ xử lý của CPU, nâng cao hiệu quả hoạt động của hệ thống, cho phép bộ vi xử lý có khả năng xử lý liên tục dòng mã lệnh. PQ hoạt động theo nguyên tắc FIFO (first in first out): lệnh nào vào trước được xử lý trước, sự vào ra liên tục của dòng mã lệnh làm cho hoạt động giữa EU và BIU nhịp nhàng.

3. Thành phần cơ bản

a) CPU (Central Processing Unit)

- Khối thực hiện lệnh EU:
 - Có nhiệm vụ **giải mã lệnh** thành các xung điện áp khác nhau để điều khiển các khối khác thực hiện lệnh
 - EU có khối điều khiển CU chứa các **mạch giải mã lệnh**, mã lệnh sẽ được đưa vào từ PQ (từ bộ nhớ chính, từ ổ đĩa,...) thông qua khối lệnh IU đưa đến đầu vào của bộ giải mã. Các thông tin thu được ở đầu ra sẽ được đưa tới ALU nếu như đó là các bài toán số học(+, -, *, /) và logic (AND, OR, NAND, NOR, XOR, ...), hoặc đưa tới mạch tạo xung điều khiển để điều khiển các khối khác thực hiện (các thiết bị khác) nếu như mã lệnh là các lệnh điều khiển.

3. Thành phần cơ bản

b) Bộ nhớ (Memory)

- Thông tin được tập hợp thành các file và lưu trữ trong đĩa cứng, đĩa mềm, đĩa CD hay trong các thanh RAM, ROM, Cache L1, L2, những thông tin được xử lý nhờ CPU ở bên trong máy tính. Những thiết bị được sử dụng để lưu trữ dữ liệu đó gọi chung là bộ nhớ
- Có 2 loại bộ nhớ chính là bộ nhớ ngoài (CD, HDD,...) và bộ nhớ trong (RAM, ROM, Cache,...)

3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ ngoài

- Chia theo cấu tạo của bộ nhớ: Bộ nhớ từ, bộ nhớ quang và bộ nhớ flash.
 - Bộ nhớ từ sử dụng các vật liệu từ để nhớ bao gồm: có đĩa mềm FDD (floppy disk), đĩa cứng HDD (hard disk), băng từ.
 - Bộ nhớ quang cấu tạo bởi các vật liệu nhớ bằng quang: CD-ROM, DVD, ...
 - Bộ nhớ flash: USB, SSD, ...

3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ ngoài

Bộ nhớ từ



HDD



Ổ đĩa mềm floppy

Bộ nhớ quang



Ổ đĩa CD



Ổ đĩa DVD

3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ ngoài

Bộ nhớ



USB
(2013 1TB)

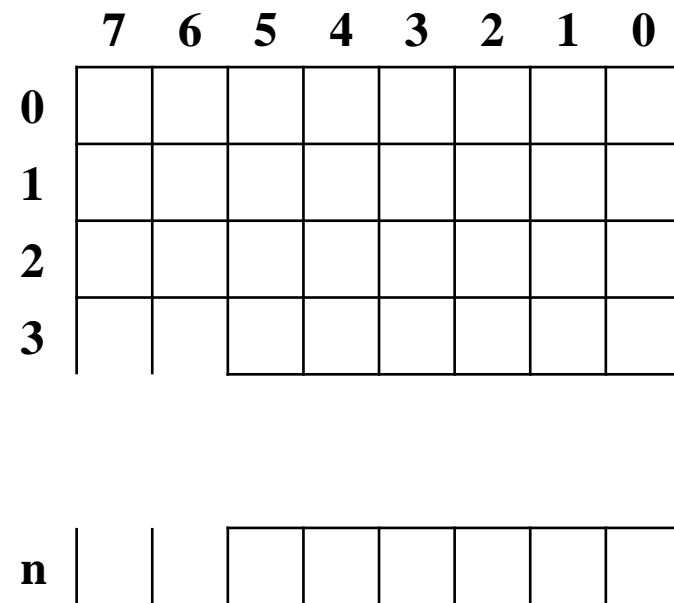


SSD
(Solid State Drive)

3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ trong

- Được cấu tạo từ các vi mạch nhớ bán dẫn.
- Sử dụng để lưu trữ các lệnh, các chương trình và dữ liệu trong quá trình xử lý.
- Được chia thành các ô nhớ, mỗi một ô nhớ được chia thành nhiều ngăn, trong đó mỗi ngăn lưu trữ 1 bit thông tin (bit 0 hoặc bit 1), thông thường một ô nhớ tương ứng với 1 byte (1 byte gồm có 8 bit).

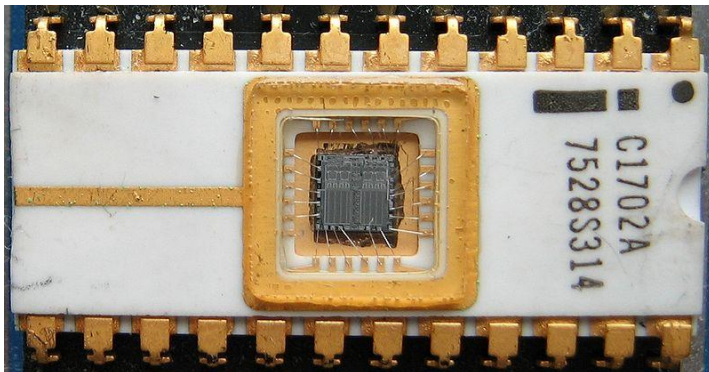


3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ trong

Bộ nhớ trong được chia làm 2 loại chính: Bộ nhớ chỉ đọc ROM và bộ nhớ truy nhập bất kỳ RAM

- Bộ nhớ chỉ đọc ROM (Read Only Memory):
 - ✓ Là bộ nhớ chỉ đọc không thể ghi hay xóa được
 - ✓ Được sử dụng để chứa các chương trình điều khiển hệ thống BIOS (Basic Input Output System - hệ thống vào ra cơ sở).



The first EPROM (Intel 1702)

3. Thành phần cơ bản

b) Bộ nhớ (Memory)- Bộ nhớ trong

- Bộ nhớ truy nhập bất kỳ RAM (Random Access Memory):
 - ✓ Là bộ nhớ có khả năng đọc và ghi dữ liệu
 - ✓ Được sử dụng để lưu trữ các mã lệnh, dữ liệu trong quá trình xử lý
 - ✓ Dữ liệu sẽ bị mất khi máy vi tính mất điện

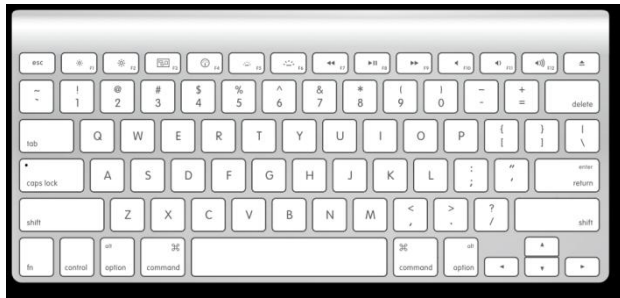


3. Thành phần cơ bản

c) Thiết bị ngoại vi

Thiết bị ngoại vi bao gồm thiết bị vào và thiết bị ra

- **Thiết bị vào:** Là những thiết bị được sử dụng để đưa thông tin vào máy vi tính.



3. Thành phần cơ bản

c) Thiết bị ngoại vi

- **Thiết bị ra:** : Là những thiết bị được sử dụng để đưa thông tin ra khỏi máy vi tính.



3. Thành phần cơ bản

d) Hệ thống Bus

- Các đường dẫn kết nối các thiết bị lại với nhau,
- Được sử dụng để truyền tín hiệu giữa CPU, bộ nhớ và các thiết bị ngoại vi với nhau.
- Các loại hệ thống Bus: Bus dữ liệu, Bus địa chỉ và Bus điều khiển.
 - ✓ Bus dữ liệu:
 - ❑ Các đường dây được sử dụng để truyền thông tin giữa CPU, bộ nhớ và các thiết bị ngoại vi với nhau
 - ❑ Bus 4 bit, bus 8 bit, bus 16 bit, bus 32 bit hoặc bus 64 bit tùy theo từng thế hệ thế hệ vi xử lý, số các bit này chính là số lượng dây dẫn cấu thành nên bus

3. Thành phần cơ bản

d) Hệ thống Bus

✓ Bus dữ liệu:

Ví dụ: Intel 4004 có 4 bit dữ liệu

Intel 8088 có 8 bit dữ liệu

Intel 8086 có 16 bit dữ liệu.

✓ Bus địa chỉ:

❑ Được sử dụng để truyền các tín hiệu địa chỉ từ bộ vi xử lý tới bộ nhớ hoặc tới các thiết bị ngoại vi.

❑ Gồm có 12, 16, 20, 24, 32, 36 bit tùy theo từng thế hệ vi xử lý

Ví dụ: Intel 8086 có 20 bit bus địa chỉ.

Intel pentium có 32 bit bus địa chỉ.

Intel pentium III có 36 bit bus địa chỉ.

3. Thành phần cơ bản

d) Hệ thống Bus

✓ Bus địa chỉ:

- ❑ Số lượng bit của bus chỉ ra **khả năng quản lý không gian bộ nhớ vật lý của CPU**

Ví dụ: CPU có bus địa chỉ là 24 bit thì như vậy nó có khả năng quản lý được $2^{24} = 16 \times 2^{10} \times 2^{10}$ bytes = 16 **Mbytes** ô nhớ khác nhau tương ứng với địa chỉ ô nhớ là từ 0 đến $2^{24}-1$

✓ Bus điều khiển:

- ❑ Được sử dụng để điều khiển bộ nhớ hoặc các thiết bị ngoại vi khác theo yêu cầu của CPU.
- ❑ Việc điều khiển này có thể là **điều khiển đọc/ghi** đối với **bộ nhớ** hoặc là **điều khiển vào ra** đối với **thiết bị ngoại vi**.



THANKS FOR YOUR ATTENTION



Chương 7

Các thiết bị I/O
(Các thiết bị ngoại vi)

Nội dung chính của chương

- Nguyên tắc cơ bản của việc cài đặt và sử dụng các thiết bị ngoại vi
- Sử dụng các cổng và các khe cắm mở rộng để cài đặt bổ sung các thiết bị ngoại vi
- Bàn phím được tổ chức và hoạt động như thế nào
- Các thiết bị trỏ (Các loại chuột)
- Hệ thống con video hoạt động như thế nào



Để hoạt động, một thiết bị mới có thể cần:

- Trình điều khiển thiết bị hoặc BIOS
- Tài nguyên hệ thống (IRQ, DMA, địa chỉ I/O , địa chỉ bộ nhớ)
- Phần mềm ứng dụng để khai thác chức năng của thiết bị



Nguyên tắc cơ bản của việc cài đặt các thiết bị ngoại vi

- Thiết bị ngoại vi là một thiết bị phần cứng được điều khiển bằng phần mềm vì thế cần phải cài đặt cả hai
- Phần mềm có thể có nhiều kiểu (cấp) khác nhau: phải cài đặt tất cả các cấp
- Có thể có nhiều thiết bị đòi hỏi cùng các tài nguyên hệ thống: cần phải giải quyết việc xung đột tài nguyên nếu nó xảy ra



Thủ tục cài đặt thiết bị ngoại vi

1. Cài đặt thiết bị (trong hoặc ngoài)
2. Cài đặt trình điều khiển thiết bị
3. Cài đặt phần mềm ứng dụng sử dụng được chức năng của thiết bị



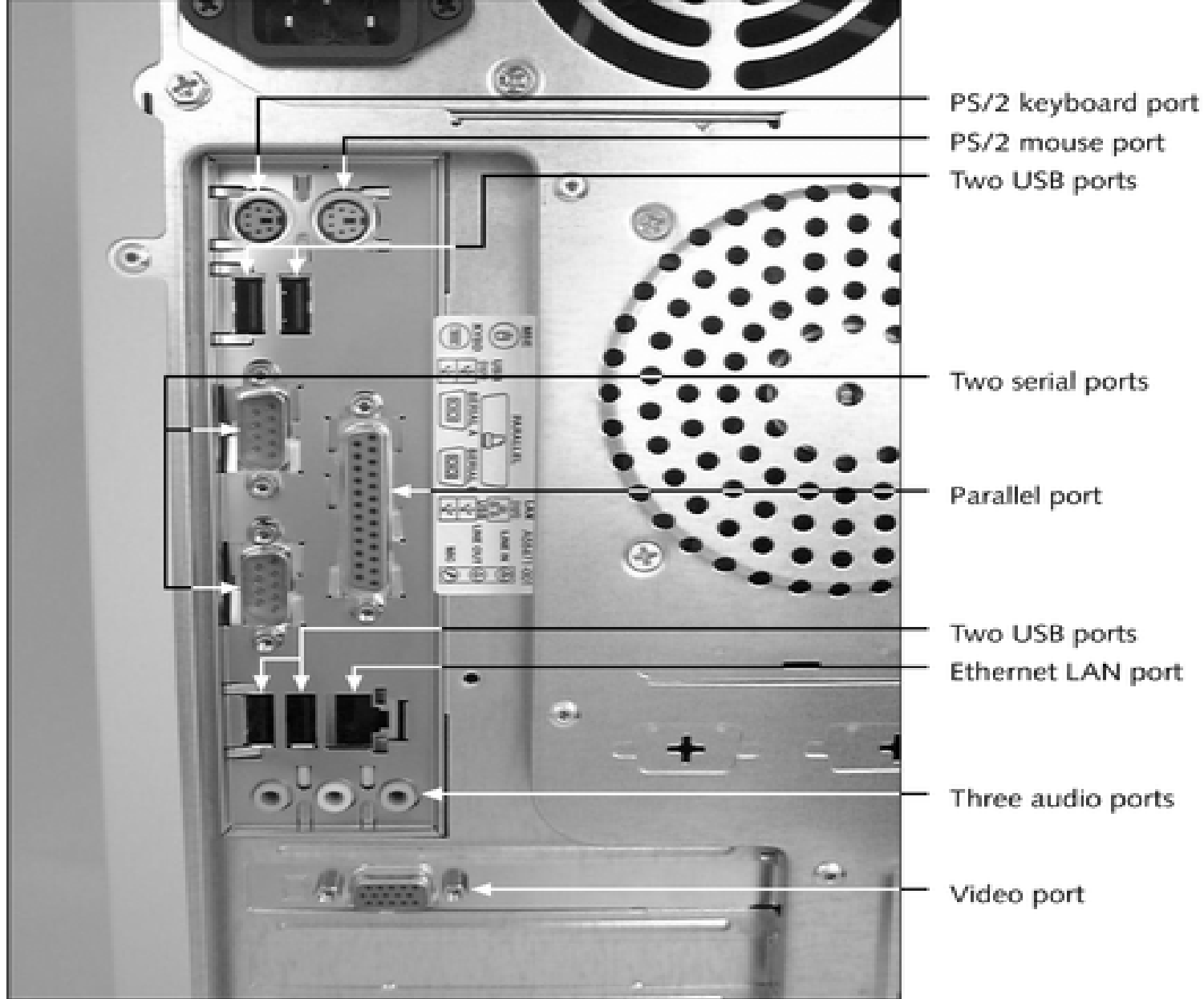


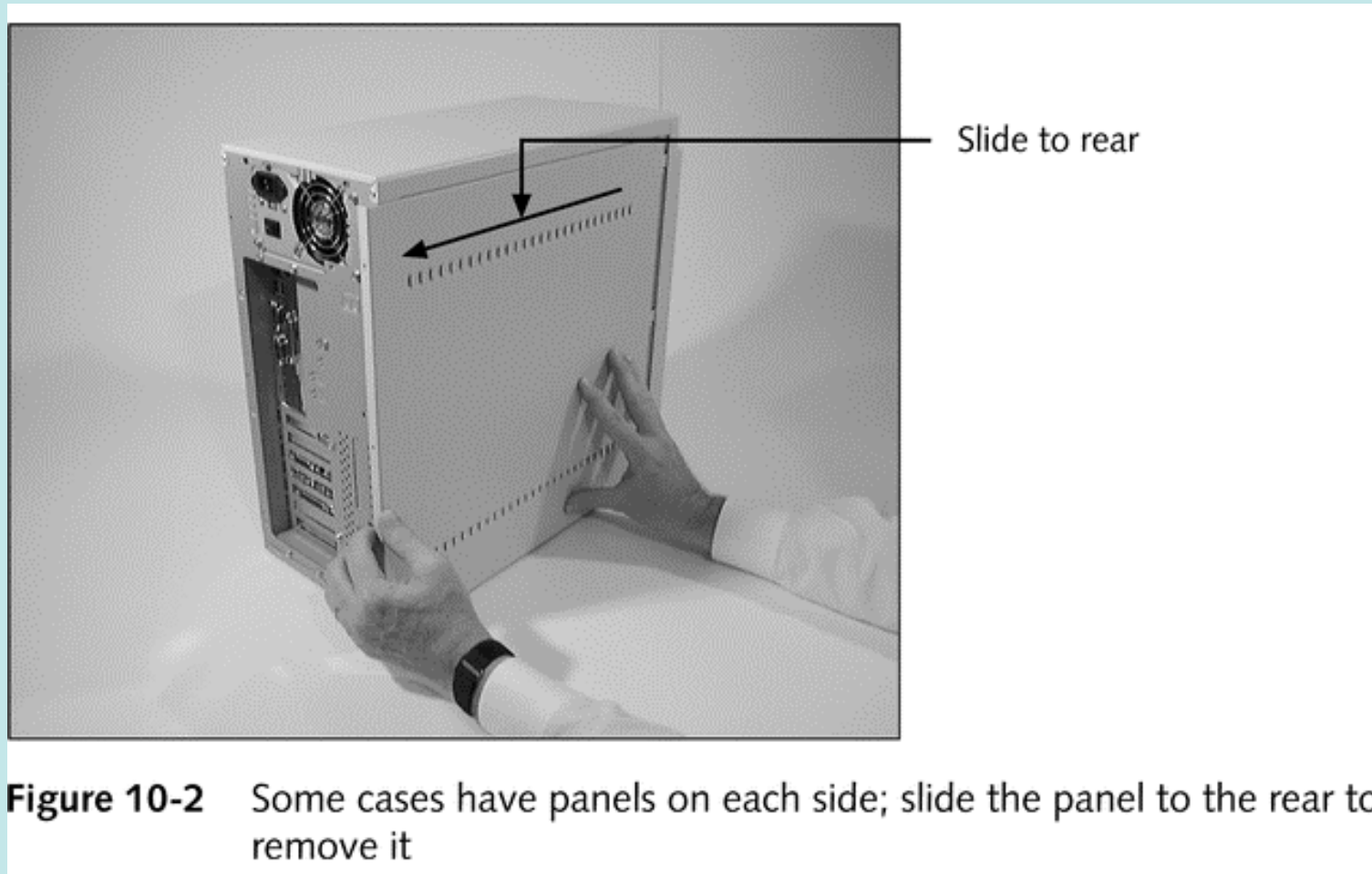
Figure 10-1 Rear of computer case showing ports; only one port (the video port) is not coming directly off the motherboard

Cài đặt thiết bị ngoài dùng các cổng

- Tắt PC, gắn thiết bị, khởi động lại PC
- Nếu thiết bị là PnP, thì **Add New Hardware Wizard** sẽ tự động chạy và lần lượt đưa ra các chỉ dẫn để cài đặt thiết bị



Cài đặt thiết bị trong



Cài đặt một card mở rộng

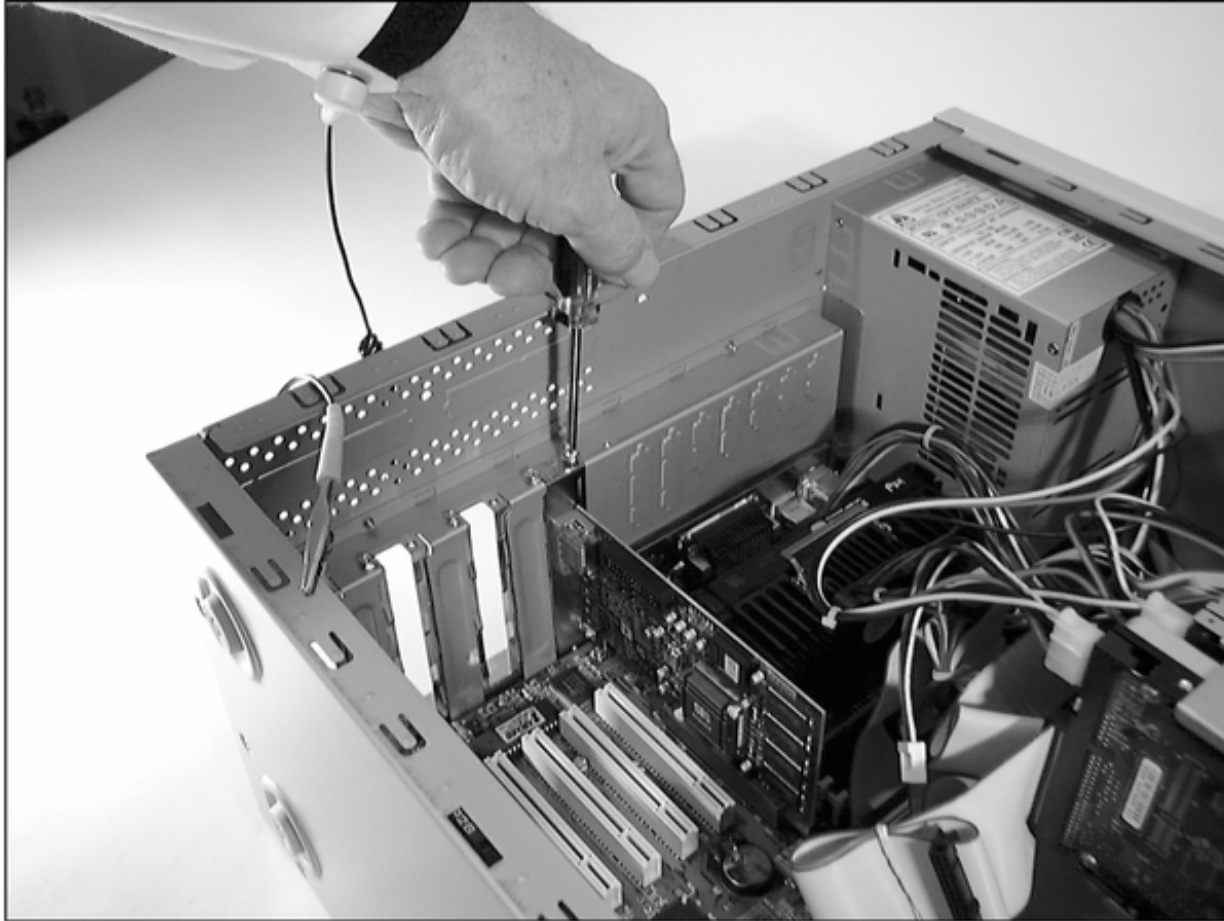


Figure 10-3 Secure an expansion card in the slot with a screw

Sử dụng các cổng và các khe cắm mở rộng để cài đặt thiết bị

- Các thiết bị có thể:
 - Cắm trực tiếp vào các cổng (nối tiếp, song song, USB, hay IEEE 1394)
 - Dùng một card mở rộng cắm vào một khe cắm mở rộng
- Các máy tính thường có:
 - 1 hoặc 2 cổng nối tiếp
 - 1 cổng song song
 - 1 hoặc nhiều cổng USB hoặc 1 cổng IEEE 1394 (trên các máy tính mới)



Tốc độ truyền dữ liệu của cổng

Table 10-1 Data transmission speeds for various port types

Port Type	Maximum Speed
IEEE 1394/FireWire	1.2 Gbps (gigabits per second)
Hi-Speed USB	480 Mbps (megabits per second)
Original USB	12 Mbps
Parallel	1.5 Mbps
Serial	115.2 Kbps (kilobits per second)



Cổng nối tiếp

- Truyền số liệu nối tiếp
- Đếm số chân của cổng để nhận dạng
- Còn được gọi là các đầu nối DB-9 và DB-25
- Luôn là male
- Có thể có COM1, COM2, COM3 và COM4
- Tuân theo chuẩn giao tiếp RS-232c



Nhận dạng một số cổng

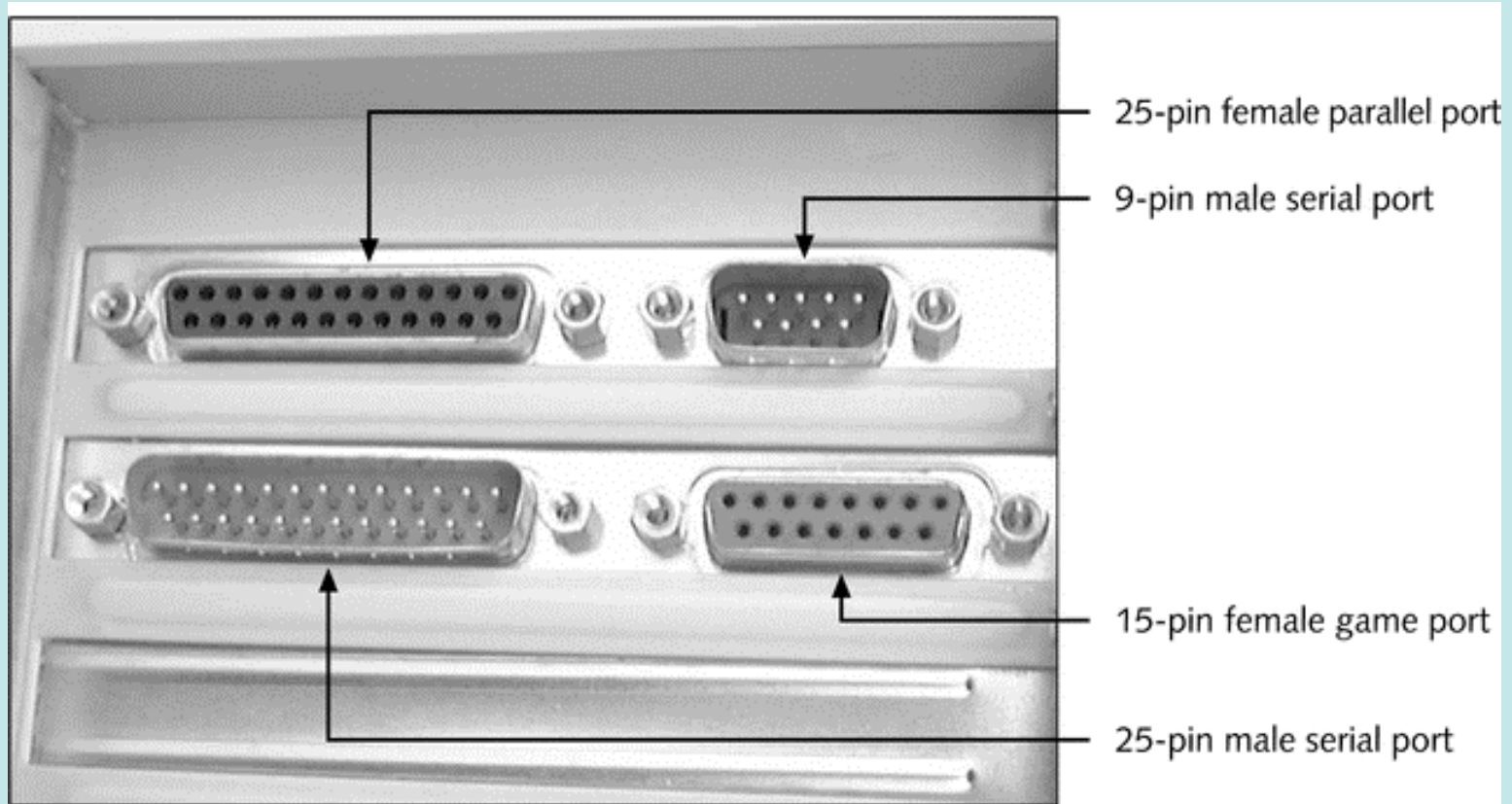


Figure 10-4 Serial, parallel, and game ports



Các cổng nối tiếp và song song

Table 10-2 Default port assignments on many computers

Port	IRQ	I/O Address (in Hex)	Type
COM1	IRQ 4	03F8 – 03FF	Serial
COM2	IRQ 3	02F8 – 02FF	Serial
COM3	IRQ 4	03E8 – 03EF	Serial
COM4	IRQ 3	02E8 – 02EF	Serial
LPT1	IRQ 7	0378 – 037F	Parallel
LPT2	IRQ 5	0278 – 027F	Parallel



Tài nguyên hệ thống cho các cổng

ROM PCI/ISA BIOS (<<P2B>>)
CHIPSET FEATURES SETUP
AWARD SOFTWARE, INC.

SDRAM CONFIGURATION	: By SPD	Onboard FDC Controller	: Enabled
SDRAM CAS Latency	: 2T	Onboard FDC Swap A & B	: No Swap
SDRAM RAS to CAS Delay	: 3T	Onboard Serial Port 1	: 3F8H/IRQ4
SDRAM RAS Precharge Time	: 3T	Onboard Serial Port 2	: 2F8H/IRQ3
DRAM Idle Timer	: 16T	Onboard Parallel Port	: 378H/IRQ7
SDRAM MA Wait State	: Normal	Parallel Port Mode	: ECP-EPP
Snoop Ahead	: Enabled	ECP DMA Select	: 3
Host Bus Fast Data Ready	: Enabled	VART2 Use Infrared	: Disabled
16-bit I/O Recovery Time	: 1 BUSCLK	Onboard PCI IDE Enable	: Both
8-bit I/O Recovery Time	: 1 BUSCLK	IDE Ultra DMA Mode	: Auto
Graphics Aperture Size	: 64MB	IDE0 Master PIO/DMA Mode	: Auto
Video Memory Cache Mode	: UC	IDE0 Slave PIO/DMA Mode	: Auto
PCI 2.1 Support	: Enabled	IDE1 Master PIO/DMA Mode	: Auto
Memory Hole At 15W-16W	: Disabled	IDE1 Slave PIO/DMA Mode	: Auto
DRAM are 64 (Not 72) bits wide			
Data Integrity Mode	: Non-ECC		

ESC : Quit ↑ ↓ → ← : Select Item
F1 : Help PU/PD/-/=: Modify
F5 : Old Values (Shift)F2 : Color
F6 : Load BIOS Defaults
F7 : Load Setup Defaults

Serial and parallel port assignments

Figure 10-5 CMOS setup screen for chipset features

Kiểm tra cấu hình của cổng

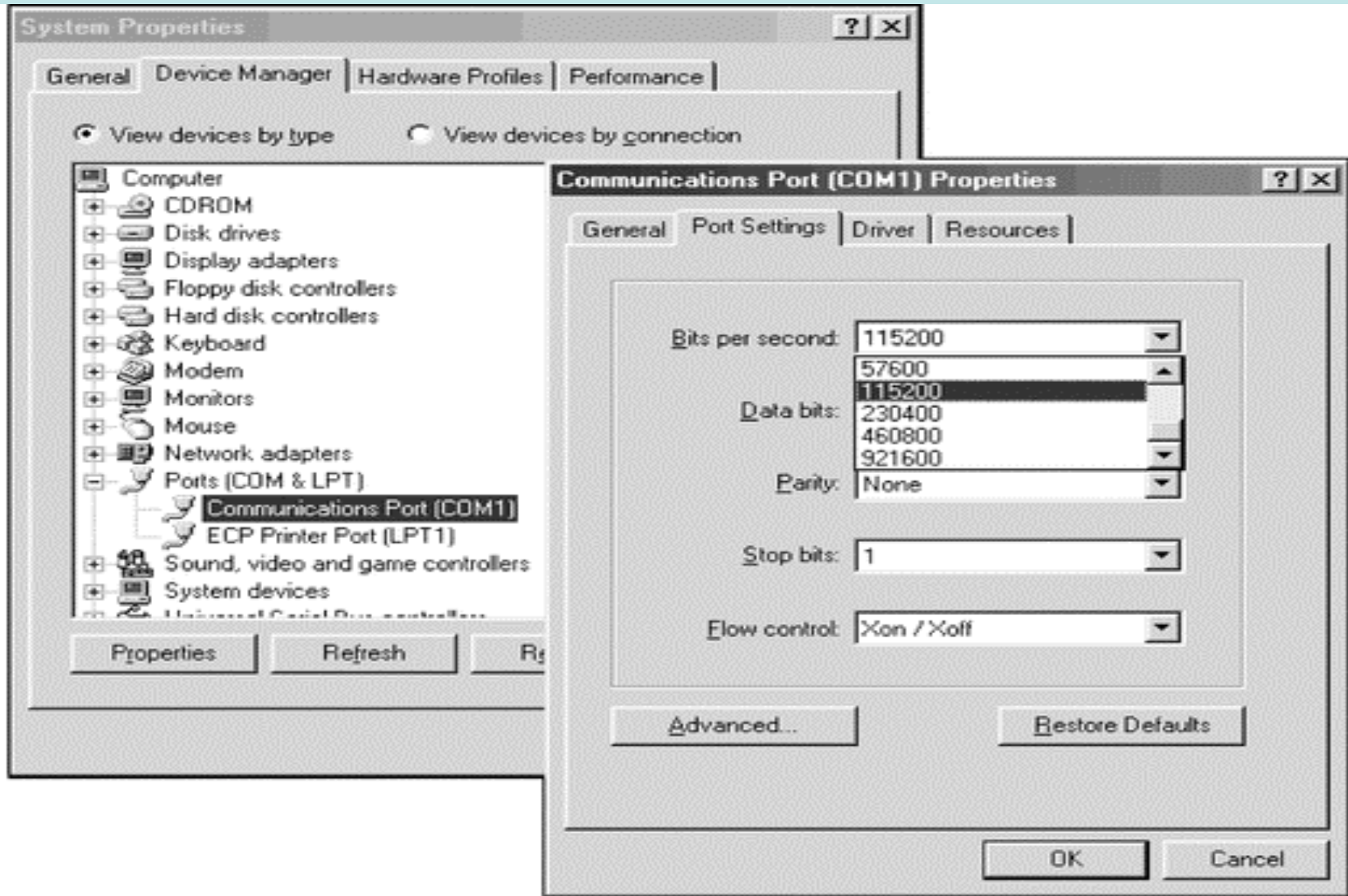


Figure 10-6 Properties of the COM1 serial port in Windows 9x

Tín hiệu của cổng nối tiếp

Table 10-3 9-pin and 25-pin serial port specifications

Pin Number for 9-Pin	Pin Number for 25-Pin	Pin Use	Description	LED Light
1	8	Carrier detect	Connection with remote is made.	CD or DCD
2	3	Receive data	Receiving data	RD or TXD
3	2	Transmit data	Sending data	SD or TXD
4	20	Data terminal ready	Modem hears its computer.	TR or DTR
5	7	Signal ground	Not used with PCs	
6	6	Data set ready	Modem is able to talk.	MR or DSR
7	4	Request to send	Computer wants to talk.	RTS
8	5	Clear to send	Modem is ready to talk.	CTS
9	22	Ring indicator	Someone is calling.	RI



Kết nối không modem

- Một cáp đặc biệt (null modem cable hay modem eliminator) cho phép truyền dữ liệu giữa hai thiết bị DTE không cần modem
- Cáp này có một vài dây nối chéo nhau để mô phỏng modem



Cách nối dây

Table 10-4 Pin connections for a 25-pin null modem cable

Pin on one end is	Connected to the pin on the other end	So that
2	3	Data sent by one computer is received by the other.
3	2	Data received by one computer is sent by the other.
6	20	One end says to the other end, "I'm able to talk."
20	6	One end hears the other end say, "I'm able to talk."
4	5	One end says to the other, "I'm ready to talk."
5	4	One end hears the other say, "I'm ready to talk."
7	7	Both ends are grounded.



Cách nối dây

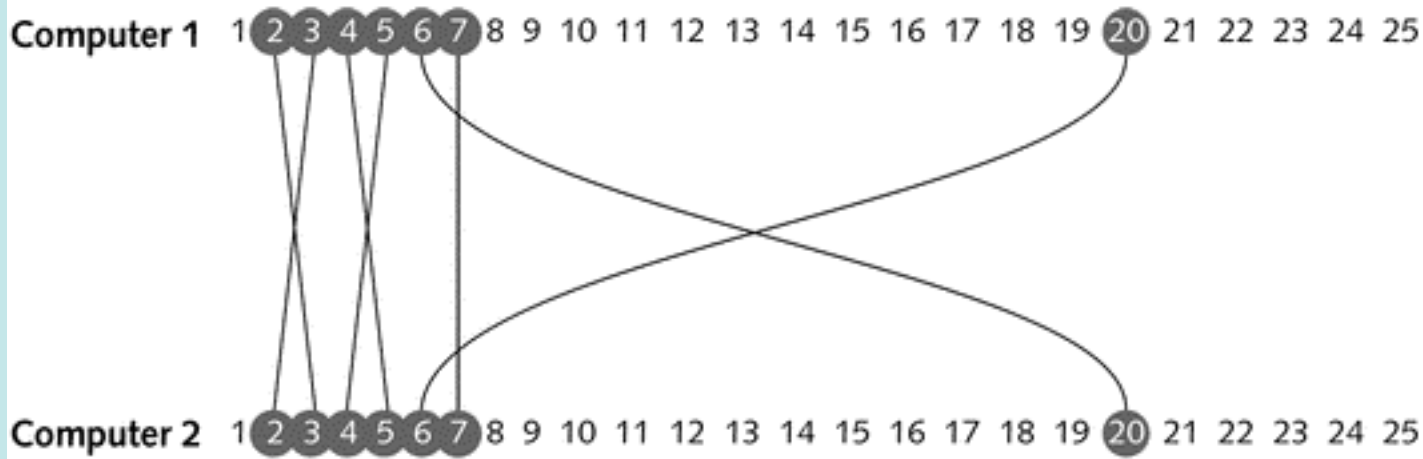


Figure 10-7 Wire connections on a 25-pin null modem cable used to transmit data

Bộ thu phát hồng ngoại

- Sử dụng tài nguyên của các cổng nối tiếp và song song để truyền thông tin
- Tạo ra các cổng hồng ngoại ảo để sử dụng các thiết bị hồng ngoại
- Đặc điểm chung: LOS (Line of Sight)
 - Công nghệ Radio (Bluetooth hay 802.11b) là phương pháp phổ biến nhất để sử dụng các thiết bị I/O không dây



Sử dụng cổng song song

- Truyền số liệu song song
- Cáp có chiều dài hạn chế (10 đến 15 feet)
- Luôn là cổng female
- Thường được dùng cho máy in song song, nhưng còn có những ứng dụng khác
- Có thể có LPT1, LPT2, LPT3



Các tín hiệu ở cổng song song SPP

Table 10-5 25-pin parallel port pin connections

Pin	Input or Output from PC	Description
1	Output	Strobe
2	Output	Data bit 0
3	Output	Data bit 1
4	Output	Data bit 2
5	Output	Data bit 3
6	Output	Data bit 4
7	Output	Data bit 5
8	Output	Data bit 6
9	Output	Data bit 7
10	Input	Acknowledge
11	Input	Busy
12	Input	Out of paper
13	Input	Select
14	Output	Auto feed
15	Input	Printer error
16	Output	Initialize paper
17	Output	Select input
18	Input	Ground for bit 0
19	Input	Ground for bit 1
20	Input	Ground for bit 2
21	Input	Ground for bit 3
22	Input	Ground for bit 4
23	Input	Ground for bit 5
24	Input	Ground for bit 6
25	Input	Ground for bit 7

3 kiểu cổng song song

- Standard parallel port (SPP)
 - Một chiều
 - Tốc độ tương đối thấp
- Enhanced Parallel Port (EPP)
 - Hai chiều
- Extended Capabilities Port (ECP)
 - Hai chiều
 - Chiếm 1 kênh DMA



Cáp máy in

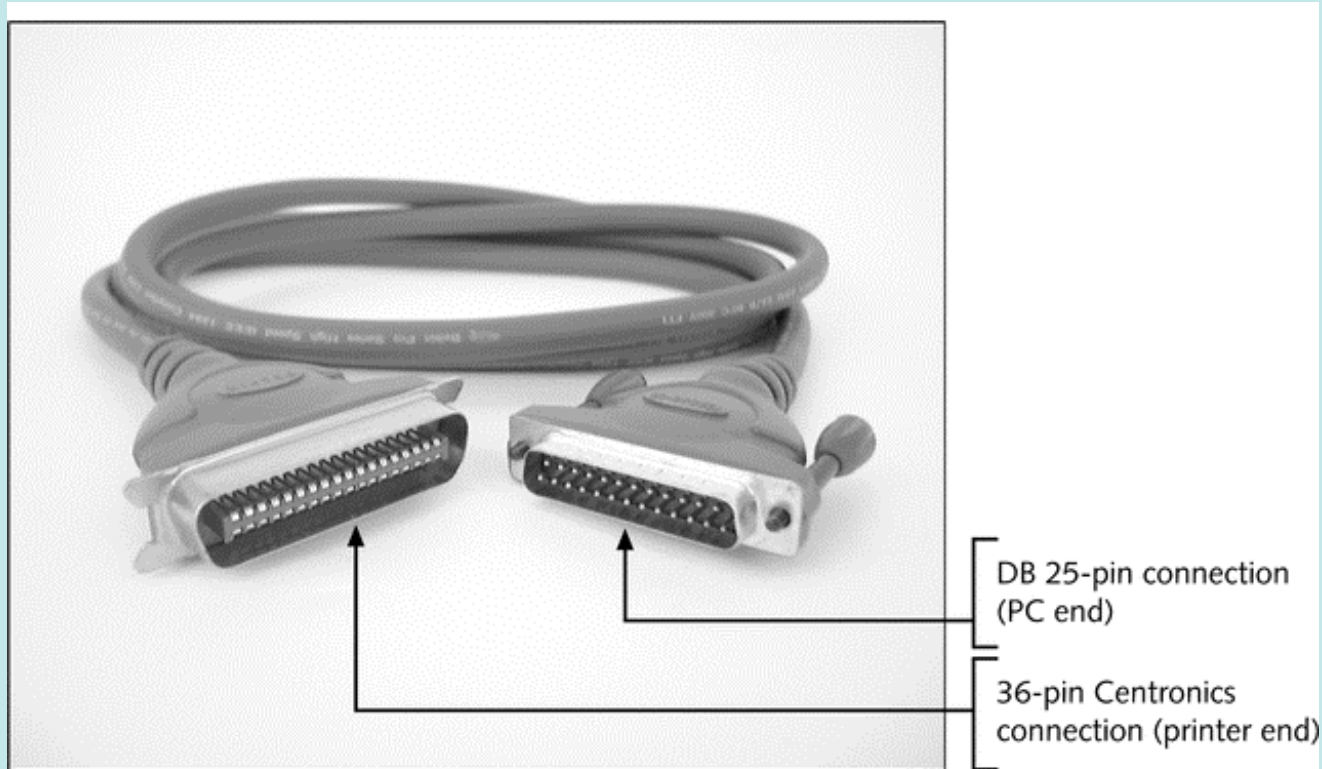


Figure 10-8 A parallel cable has a DB-25 connection at the PC end of the cable and a 36-pin Centronics connection at the printer end of the cable. Printers can use this 36-pin Centronics connection or the smaller and less common mini-Centronics connector that also has 36 pins.

Dùng cổng USB

- Tương lai sẽ thay thế hoàn toàn các cổng nối tiếp và song song, nhờ:
 - Nhanh hơn rất nhiều
 - Dễ sử dụng
- Cho phép cắm tháo nóng với các thiết bị
- Hiện đã có nhiều thiết bị sử dụng cổng USB (Chuột, Joystick, Bàn phím, Máy in, Đĩa cứng di động)



Dùng cổng USB



Figure 10-9 A motherboard with two USB ports and a USB cable; note the rectangular shape of the connection as compared to the nearby serial and parallel D-shaped ports

USB (Universal Serial Bus)



Figure 10-10 Hi-Speed and Original USB logos appear on products certified by the USB Forum

Dùng cổng USB

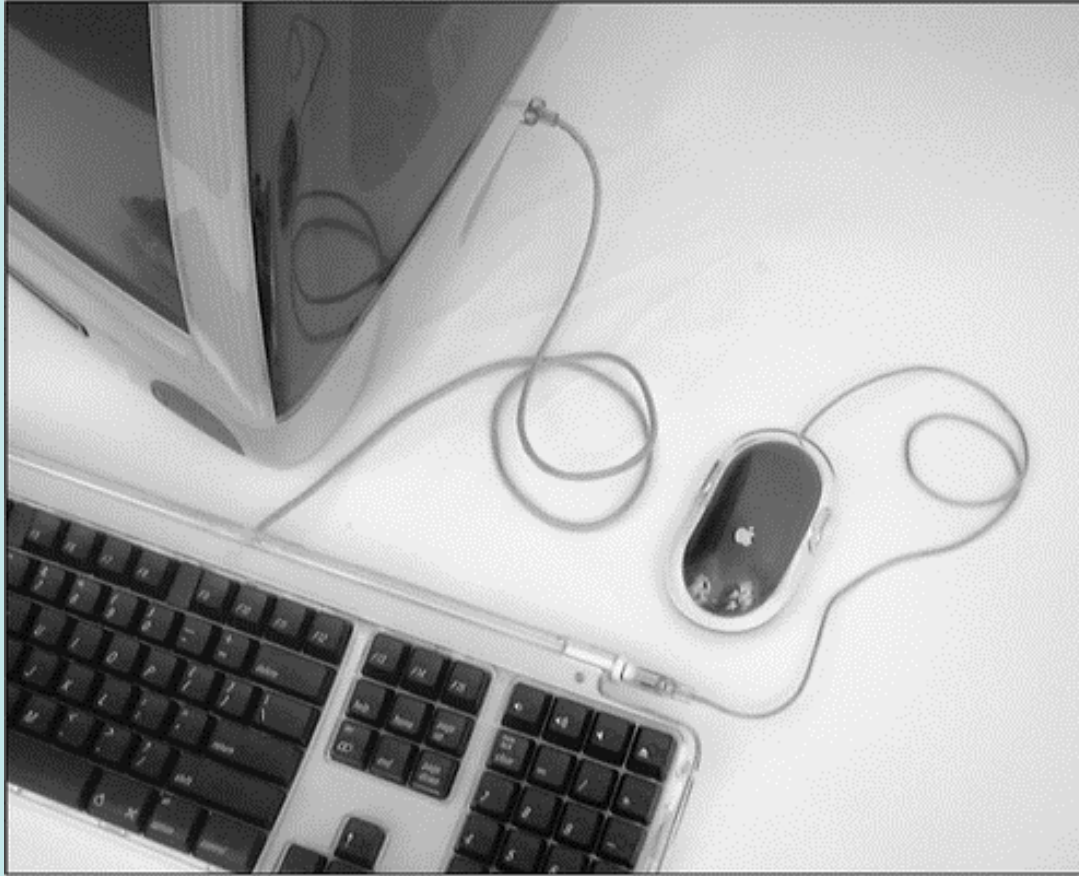


Figure 10-11 A keyboard and a mouse using a USB port daisy-chained together

Bộ điều khiển chủ USB

- Thăm dò từng thiết bị để biết nhu cầu nhập xuất dữ liệu
- Quản lý thông tin đến CPU cho tất cả các thiết bị chỉ dùng duy nhất một kênh IRQ, một dải địa chỉ I/O và một kênh DMA
- Tự động gán tài nguyên hệ thống lúc khởi động startup (cùng với OS)



Điều kiện để có một thiết bị USB sử dụng được

- Bo mạch hệ thống hoặc một card mở rộng cung cấp ít nhất 1 cổng USB
- OS có đặc tính hỗ trợ USB (Windows 98, XP, 2000)
- Thiết bị USB
- Trình điều khiển thiết bị USB đó (của hệ điều hành hoặc độc lập)



Kiểm tra USB host controller đã được cài đặt chưa?

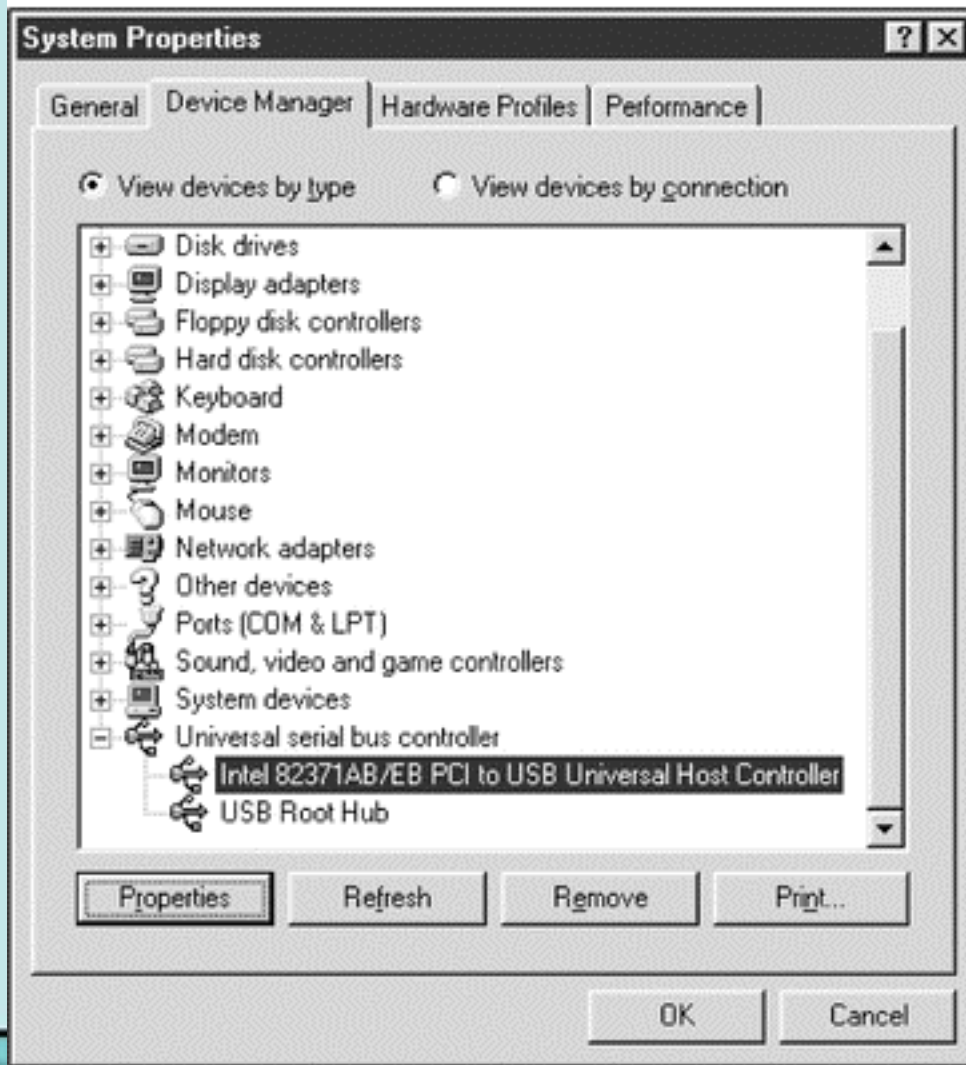


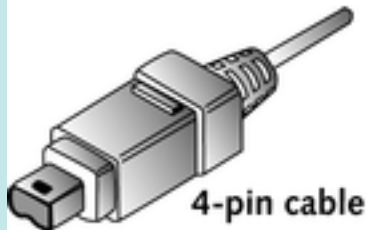
Figure 10-12 Using Device Manager, verify that the USB host controller is installed

Dùng cổng IEEE 1394

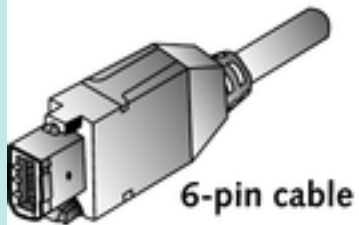
- Còn được gọi là FireWire và iLink
- Truyền số liệu nối tiếp nhưng nhanh hơn USB
- Có thể cung cấp tốc độ 1.2 Gbps
- Có thể thay thế cho SCSI đối với các thiết bị multimedia dung lượng lớn
- Các thiết bị có thể được xâu chuỗi vào nhau và được quản lý bởi bộ điều khiển chủ sử dụng chung tài nguyên hệ thống



Các kiểu cổng IEEE 1394



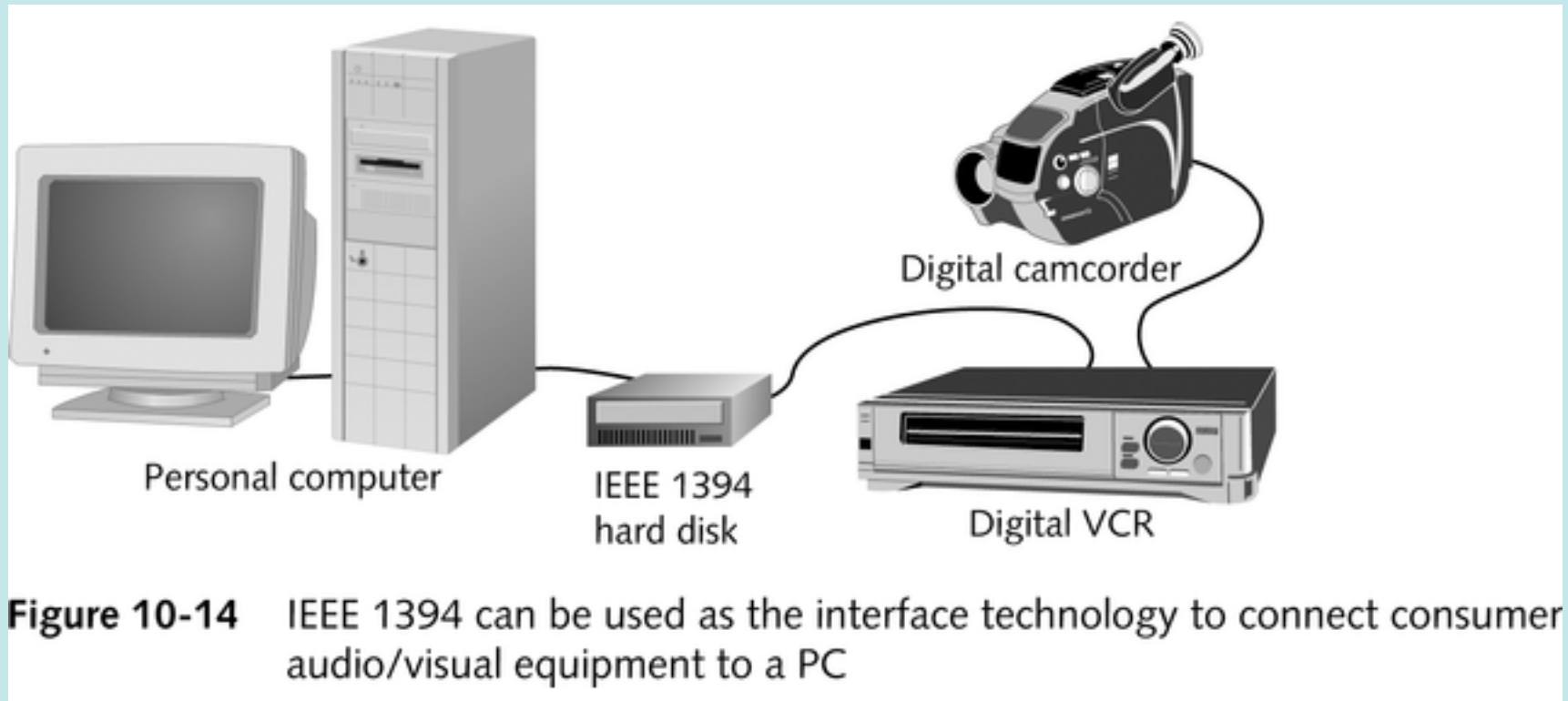
(Device requires AC adapter.)



(Two pins are used for voltage and ground.)

Figure 10-13 Two types of IEEE 1394 cable connectors; the 6-pin cable provides voltage to the device from the PC

Các thiết bị IEEE 1394



Dùng khe cắm PCI

- Bus PCI là bus I/O chuẩn hiện nay
- Các thiết bị nối với bus PCI có thể chạy với tốc độ khác với tốc độ của CPU
- Thường dùng cho các thiết bị yêu cầu tốc độ cao (Card mạng hoặc card chủ SCSI)



Các khe cắm PCI trên bo mạch hệ thống

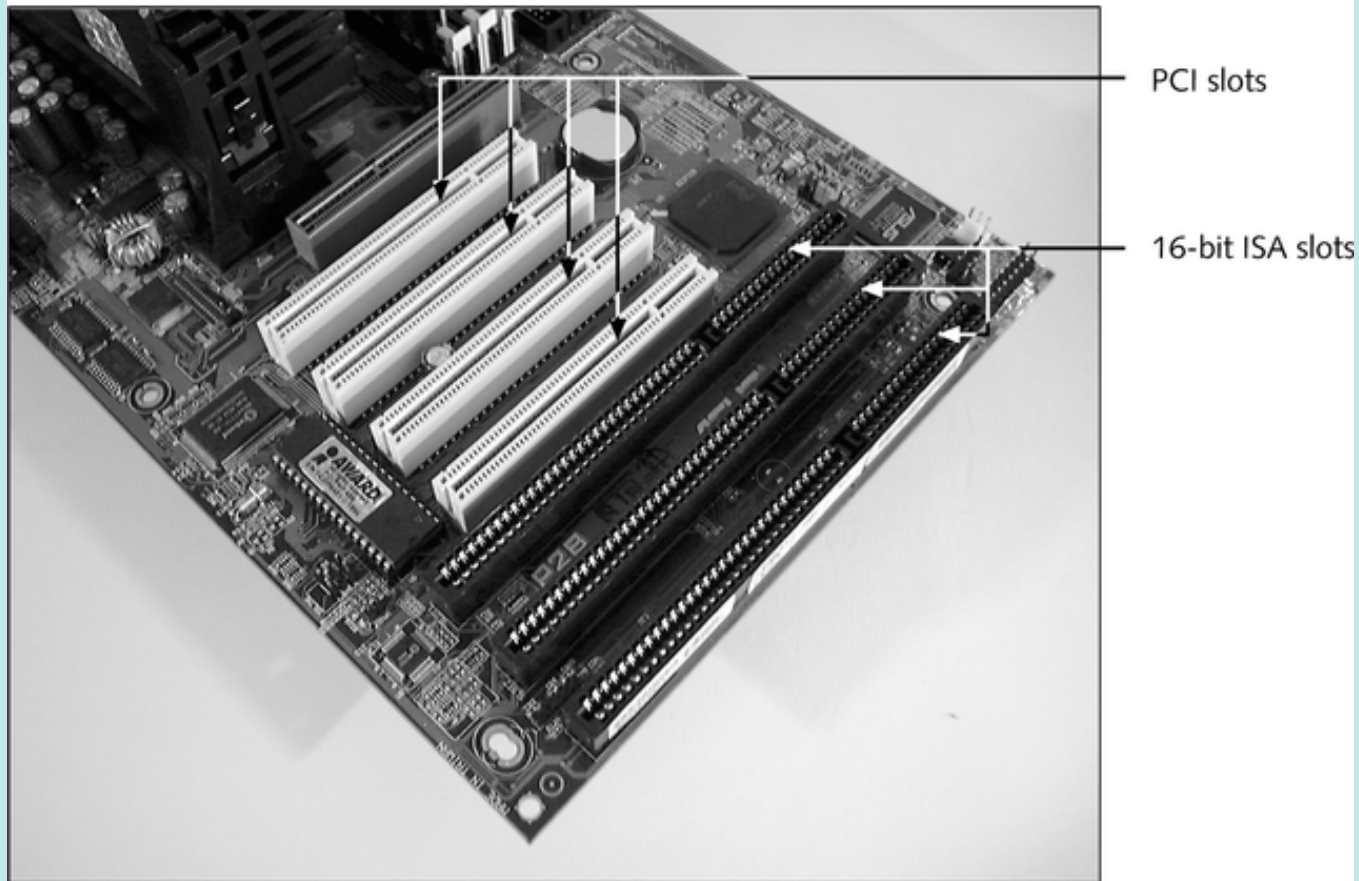


Figure 10-15 PCI bus expansion slots are shorter than ISA slots and offset farther

Chủ Bus PCI

- Quản lý bus PCI và các khe cắm mở rộng PCI
- Gán IRQ và địa chỉ I/O cho card PCI
- Bus PCI sử dụng một ngắt chuyển tiếp từ card PCI qua kênh IRQ đến CPU



Kiểm tra kênh IRQ nào được gán cho thiết bị PCI

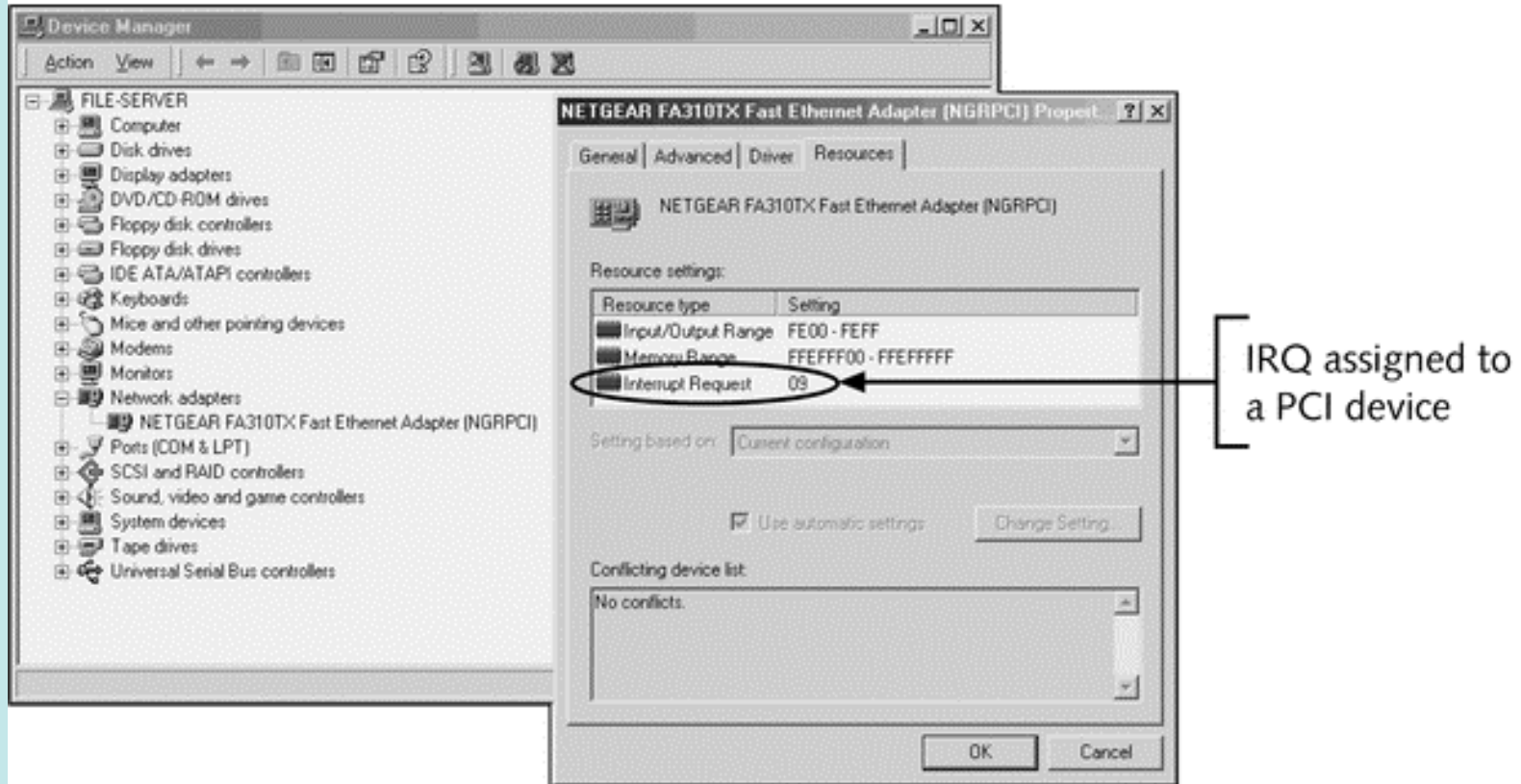


Figure 10-17 Use Device Manager to determine which IRQ has been assigned to a PCI device

Setup cho PCI

ROM PCI/ISA BIOS (<<P2B>>)
PNP AND PCI SETUP
AWARD SOFTWARE, INC.

PNP OS Installed : No	DMA 1 Used By ISA : No/ICU
Slot 1 (RIGHT) IRQ : Auto	DMA 3 Used By ISA : No/ICU
Slot 2 IRQ : Auto	DMA 5 Used By ISA : No/ICU
Slot 3 IRQ : Auto	ISA MEM Block BASE : No/ICU
Slot 4 (LEFT) IRQ : Auto	
PCI Latency Timer : 32 PCI Clock	

IRQ 3 Used By ISA : No/ICU	SVMBIOS SCSI BIOS : Auto
IRQ 4 Used By ISA : No/ICU	USB IRQ : Enabled
IRQ 5 Used By ISA : No/ICU	
IRQ 7 Used By ISA : No/ICU	
IRQ 9 Used By ISA : No/ICU	
IRQ 10 Used By ISA : No/ICU	
IRQ 11 Used By ISA : No/ICU	
IRQ 12 Used By ISA : No/ICU	
IRQ 14 Used By ISA : No/ICU	
IRQ 15 Used By ISA : No/ICU	

ESC : Quit ↑↓→← : Select Item
F1 : Help PU/PD/-/ : Modify
F5 : Old Values (Shift)F2 : Color
F6 : Load BIOS Defaults
F7 : Load Setup Defaults

IRQ assignment for PCI slots: choices are Auto, 5, 7, 9, 10, 11, 12, 14, or 15 for each slot

Reserve an IRQ for an ISA legacy device

Figure 10-18 CMOS setup screen for Plug and Play and PCI options

Dùng khe cắm ISA

- Cấu hình không tự động
- Bus ISA không quản lý tài nguyên hệ thống
- Thiết bị ISA phải yêu cầu tài nguyên hệ thống lúc khởi động
- Dùng cho các thiết bị cũ hoặc không yêu cầu tốc độ cao



Bàn phím có thể được chế tạo theo:

- Thiết kế đơn giản theo truyền thống (thông dụng)
- Thiết kế công thái học (cầu kỳ)
- Hiện nay bàn phím có rất nhiều kiểu dáng khác nhau



Bàn phím theo công thái học

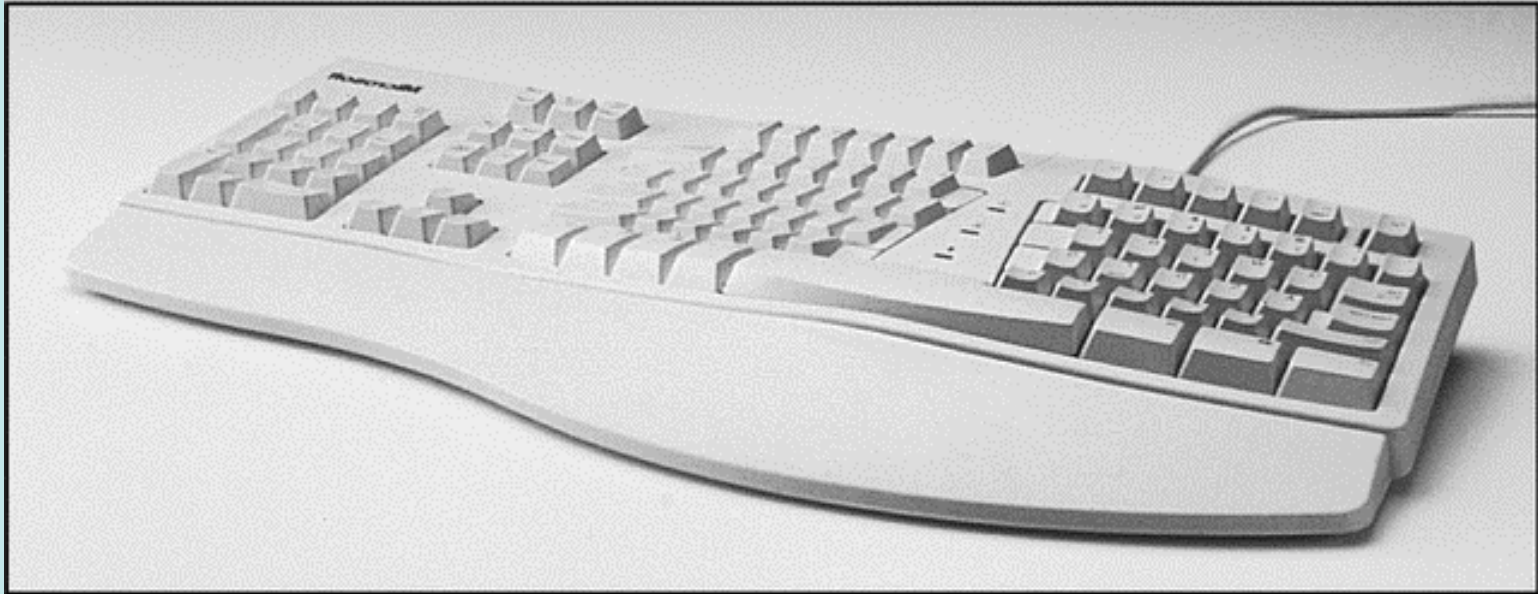


Figure 10-19 An ergonomic keyboard

Nên tập thói quen sử dụng bàn phím



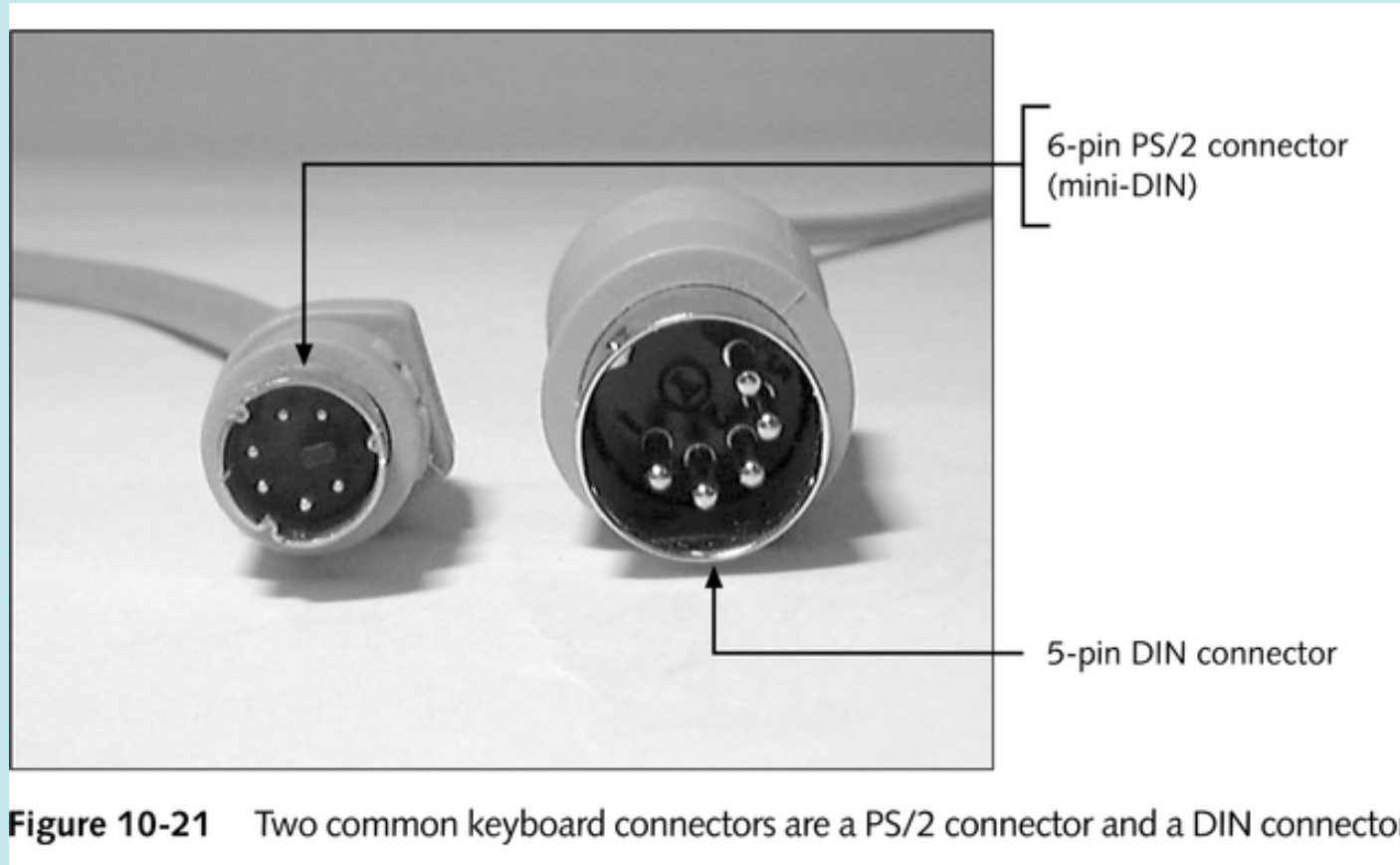
Figure 10-20 Keep wrists level, straight, and supported while at the keyboard

Đầu nối bàn phím

- PS/2 (mini-DIN)
 - Nhỏ, Tròn, 6 chân
- DIN
 - Lớn hơn, Tròn, 5 chân
- USB
- Kết nối không dây



PS/2 v   DIN



Chuyển đổi PS/2 - DIN

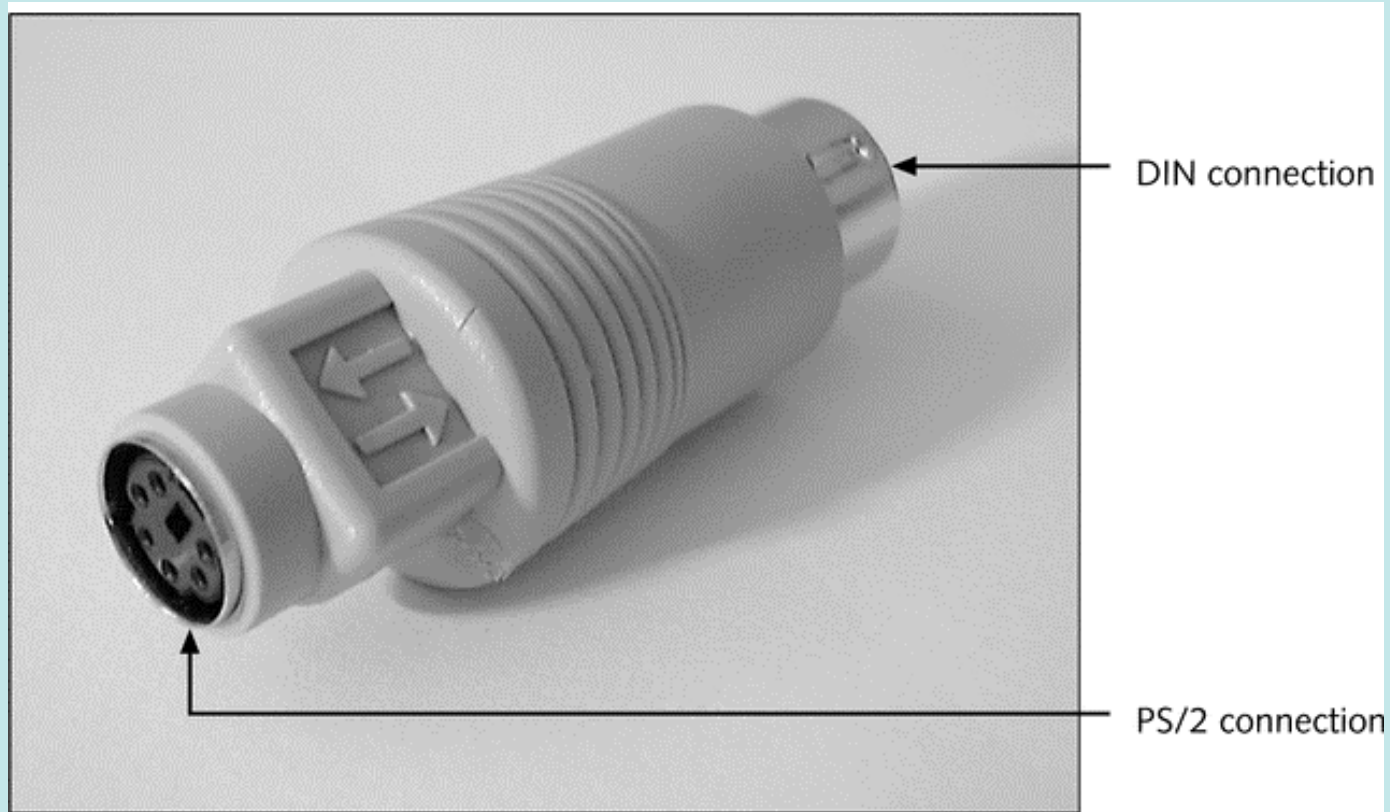


Figure 10-22 A keyboard adapter

Các tín hiệu ở đầu nối bàn phím PS/2 và DIN

Table 10-6 Pinouts for keyboard connectors

Description	6-pin Connector (PS/2)	5-Pin Connector (DIN)
Keyboard data	1	2
Not used	2	3
Ground	3	4
Current (+5 volts)	4	5
Keyboard clock	5	1
Not used	6	-



Cài đặt bàn phím

- Chỉ đơn giản là gắn bàn phím trước khi khởi động PC (đối với bàn phím dùng cổng DIN hoặc PS/2)
- BIOS quản lý bàn phím, do vậy không cần cài đặt trình điều khiển (ngoại trừ bàn phím không dây)



Các loại bàn phím và tổ chức các phím

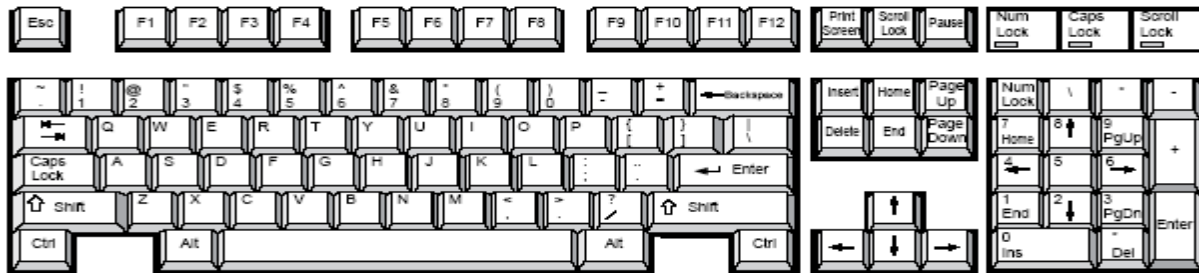
Original PC/XT - Keyboard with 83 keys



AT - Keyboard (MF-I) with 84 keys



AT - Keyboard (MF-II) with 101 or 102 keys

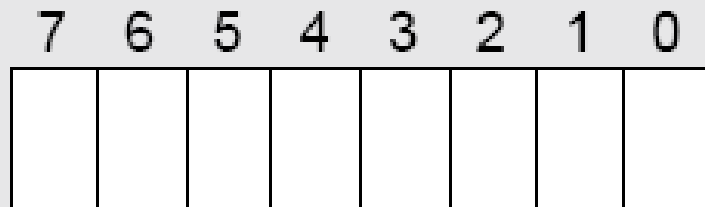


Các nhóm phím trên bàn phím hiện nay

- Các phím Trạng thái: Shift, Alt, Ctrl
- Các phím Tắt mở: CapsLock, NumLock, Scroll Lock, Insert
- Các phím đặc biệt: Print Screen, Pause
- Các phím chức năng: F1 – F12
- Các phím ASCII: A, B, C ...
- Các phím hỗ trợ Windows



Byte trạng thái bàn phím

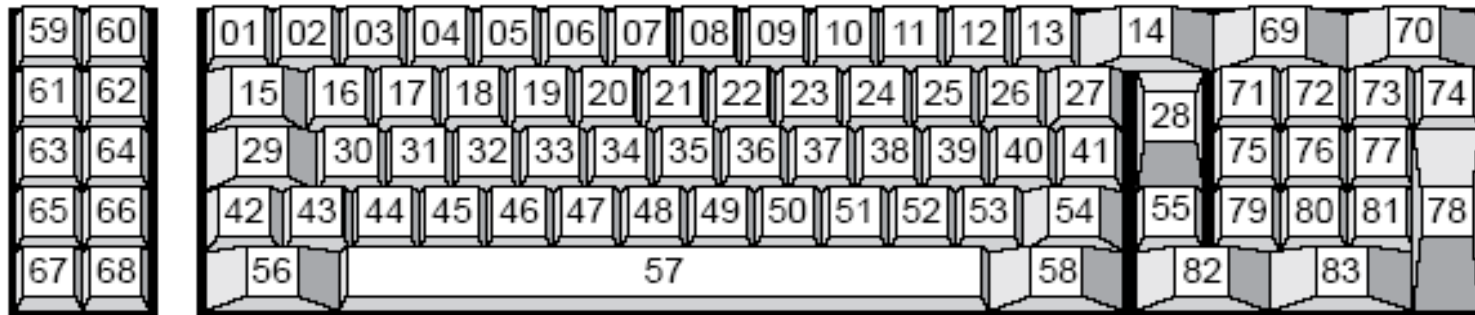


Keyboard Status Type

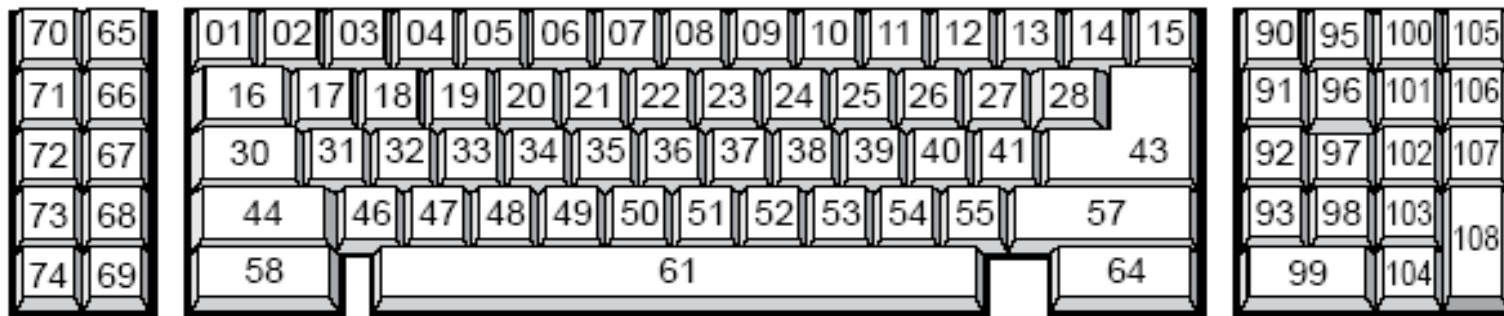
- 1 = Right **Shift** key pressed
- 1 = Left **Shift** key pressed
- 1 = **Ctrl** key pressed
- 1 = **Alt** key pressed
- 1 = SCROLL LOCK on
- 1 = NUM LOCK on
- 1 = CAPS LOCK on
- 1 = INSERT on

Mã quét của các phím

The scan codes for the PC/XT keyboard



The scan codes for the AT keyboard



Các thiết bị trỏ

- Chuột
 - Chuột bi lăn
 - Chuột quang
- Trackball (Chuột bi xoay)
- Touch pads (Chuột trên máy xách tay)

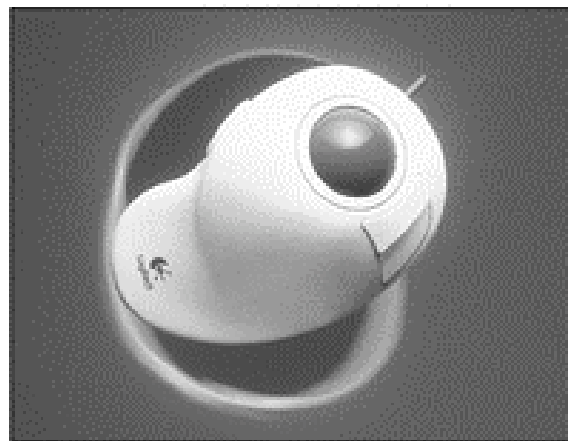
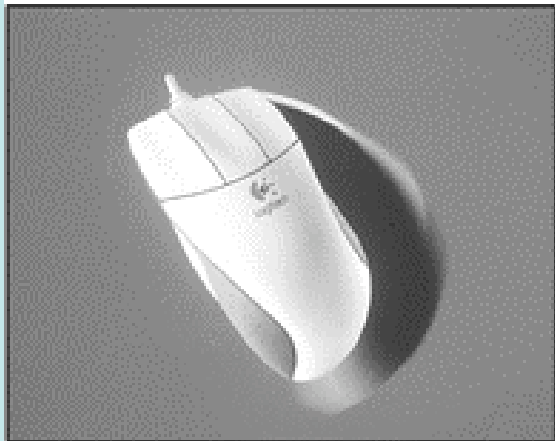


Figure 10-23 The most common pointing devices: a mouse, a trackball, and a touchpad

Chuột bi lăn

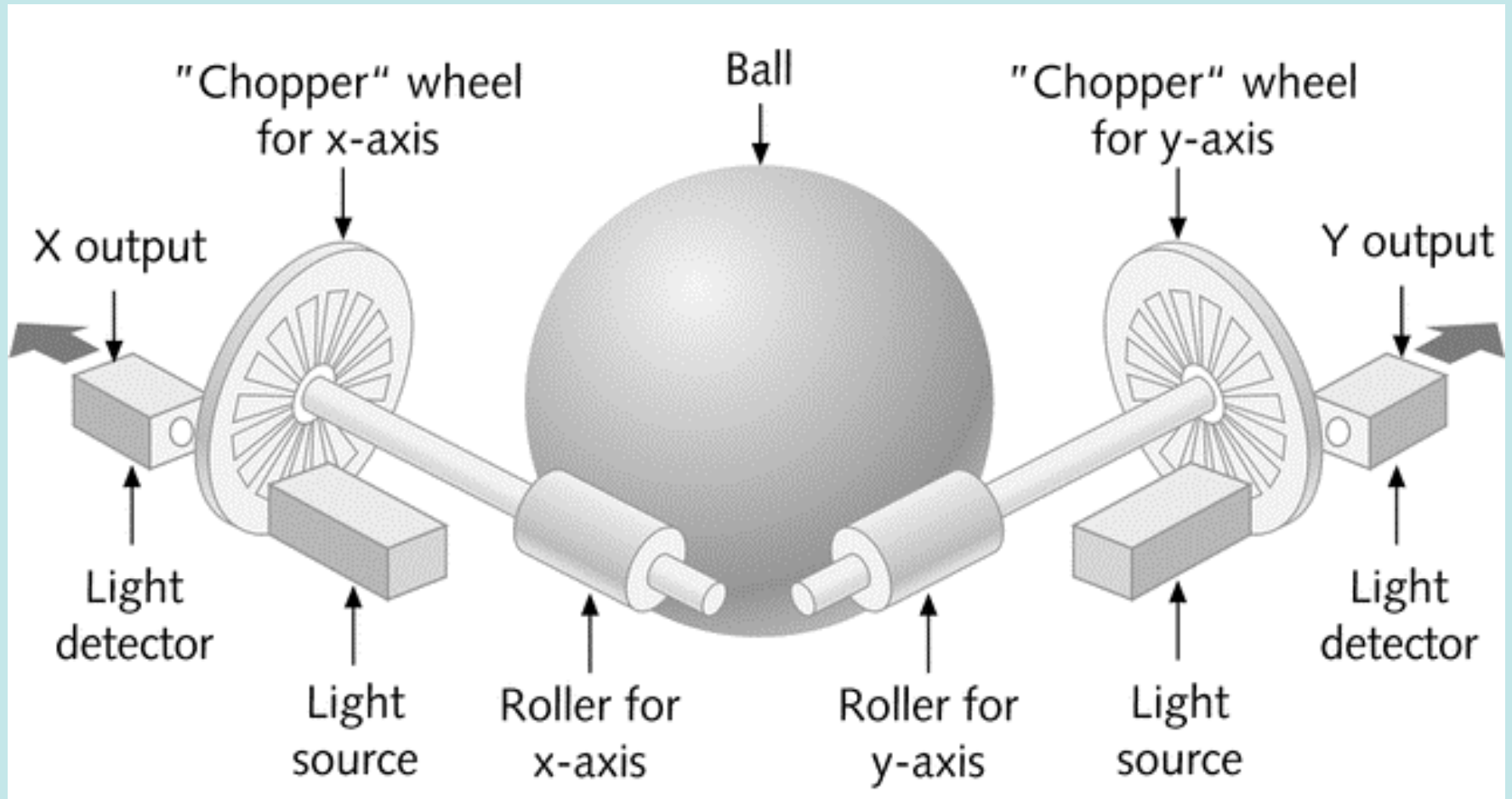


Figure 10-24 How a wheel mouse works

Kết nối chuột

- Cổng nối tiếp
- Cổng PS/2 hoặc DIN từ bo mạch hệ thống
- Cổng USB
- Dùng đầu nối chữ Y để nối chuột với bàn phím
- Kết nối không dây



Các hãng sản xuất bàn phím và các thiết bị trợ

Table 10-7 Manufacturers of keyboards and pointing devices

Manufacturer	Web Site
Mitsumi	www.mitsumi.com
Logitech	www.logitech.com
Microsoft	www.microsoft.com
Intel	www.intel.com
Belkin	www.belkin.com
Keytec, Inc	www.magictouch.com



Hệ thống con Video của PC

- Hệ thống con video = các thiết bị phần cứng chịu trách nhiệm chính trong việc hiển thị thông tin (văn bản và hình ảnh) của PC, bao gồm:
 - Video controller (card video): quyết định chất lượng hiển thị thông tin = Độ mịn và màu sắc
 - Monitor: CRT hoặc LCD



Monitor

- Đánh giá theo kích cỡ màn hình, độ phân giải, tốc độ làm tươi và đặc trưng quét xen dòng
- Hầu hết thoả mãn tiêu chuẩn Super VGA (Video Graphics Adapter)
- Sử dụng công nghệ CRT (cathode-ray tube) hoặc công nghệ LCD (liquid crystal display) technology



CRT Monitor

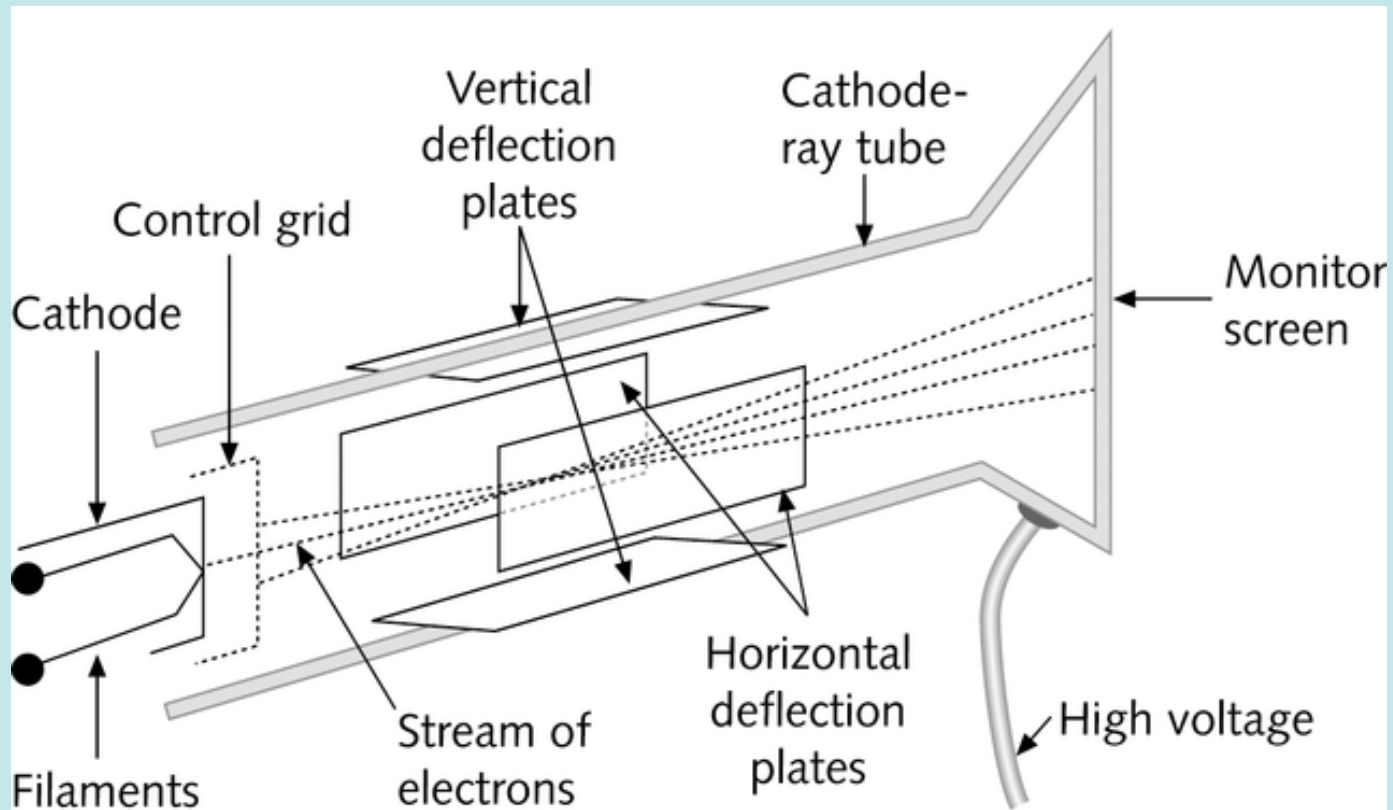


Figure 10-25 How a CRT monitor works

Chọn lựa Monitor

Table 10-8 Some features of a monitor

Monitor Characteristic	Description
Screen size	Diagonal length of the screen surface
Refresh rate	The number of times an electronic beam fills a video screen with lines from top to bottom in one second
Interlaced	The electronic beam draws every other line with each pass, which lessens the overall effect of a lower refresh rate.
Dot pitch	The distance between adjacent dots on the screen
Resolution	The number of spots, or pixels, on a screen that can be addressed by software
Multiscan	Monitors that offer a variety of refresh rates so they can support several video cards
Green monitors	Monitors that save electricity and support the EPA Energy Star program



LCD Panel

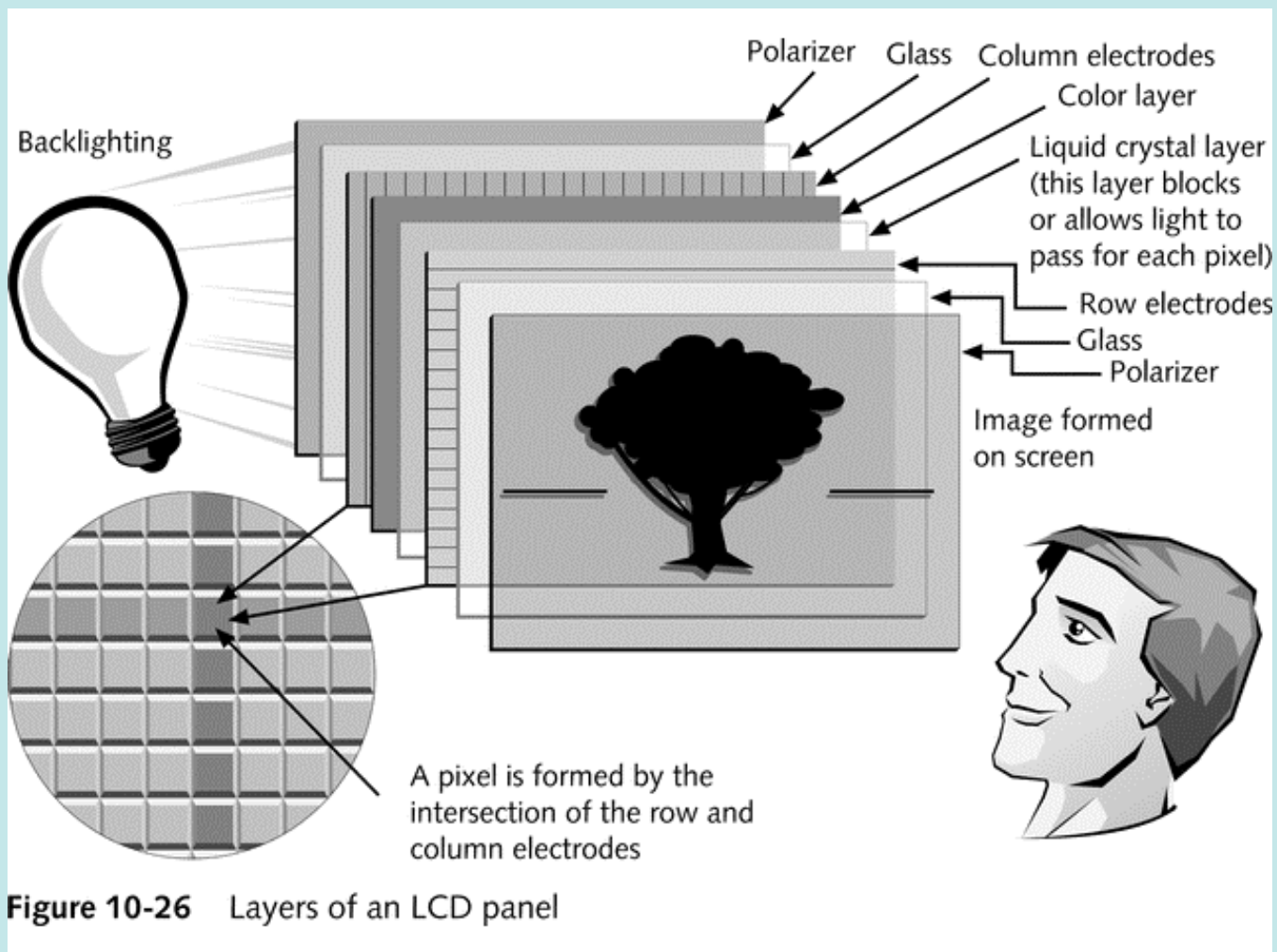


Figure 10-26 Layers of an LCD panel

Có thể dùng 2 monitor

- Tăng kích cỡ của Windows desktop

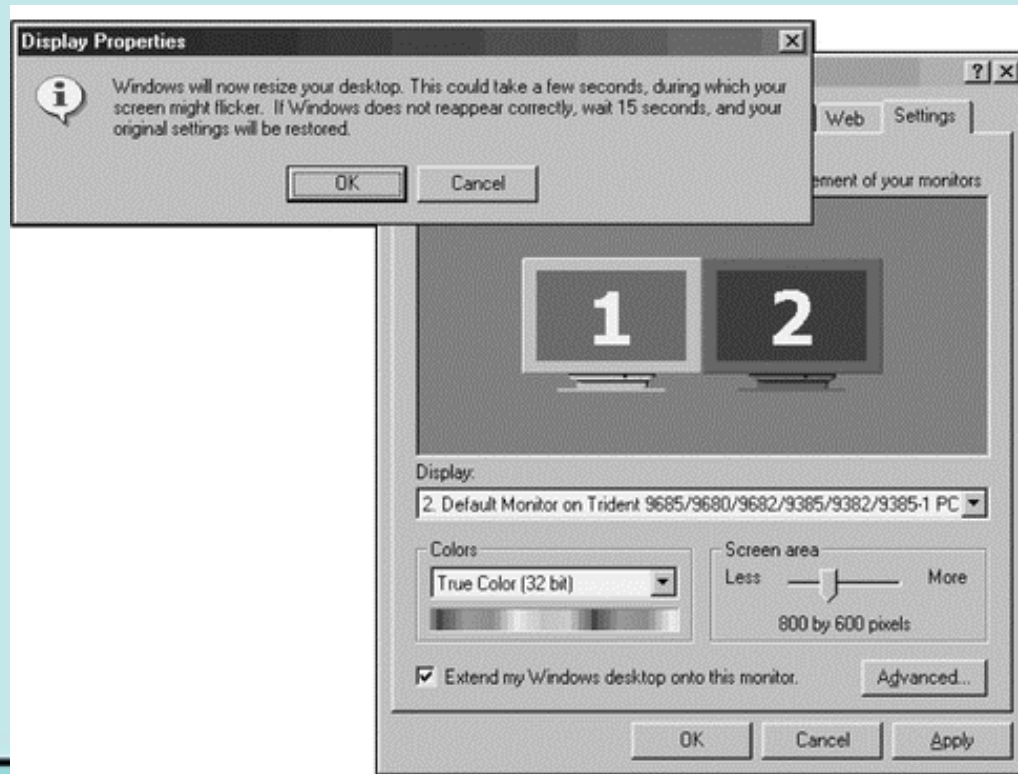


Figure 10-27 You must choose to activate a second monitor before it will be used by Windows

Card Video

- Chất lượng của hệ thống con video được đánh giá theo:
 - Tính năng của toàn bộ hệ thống PC
 - Chất lượng hình ảnh (Độ phân giải và Màu sắc)
 - Đặc trưng tiết kiệm điện năng
 - Dễ cài đặt và sử dụng
- Yếu tố quyết định nhất của 1 card video:
 - Bus dữ liệu
 - Dung lượng RAM video

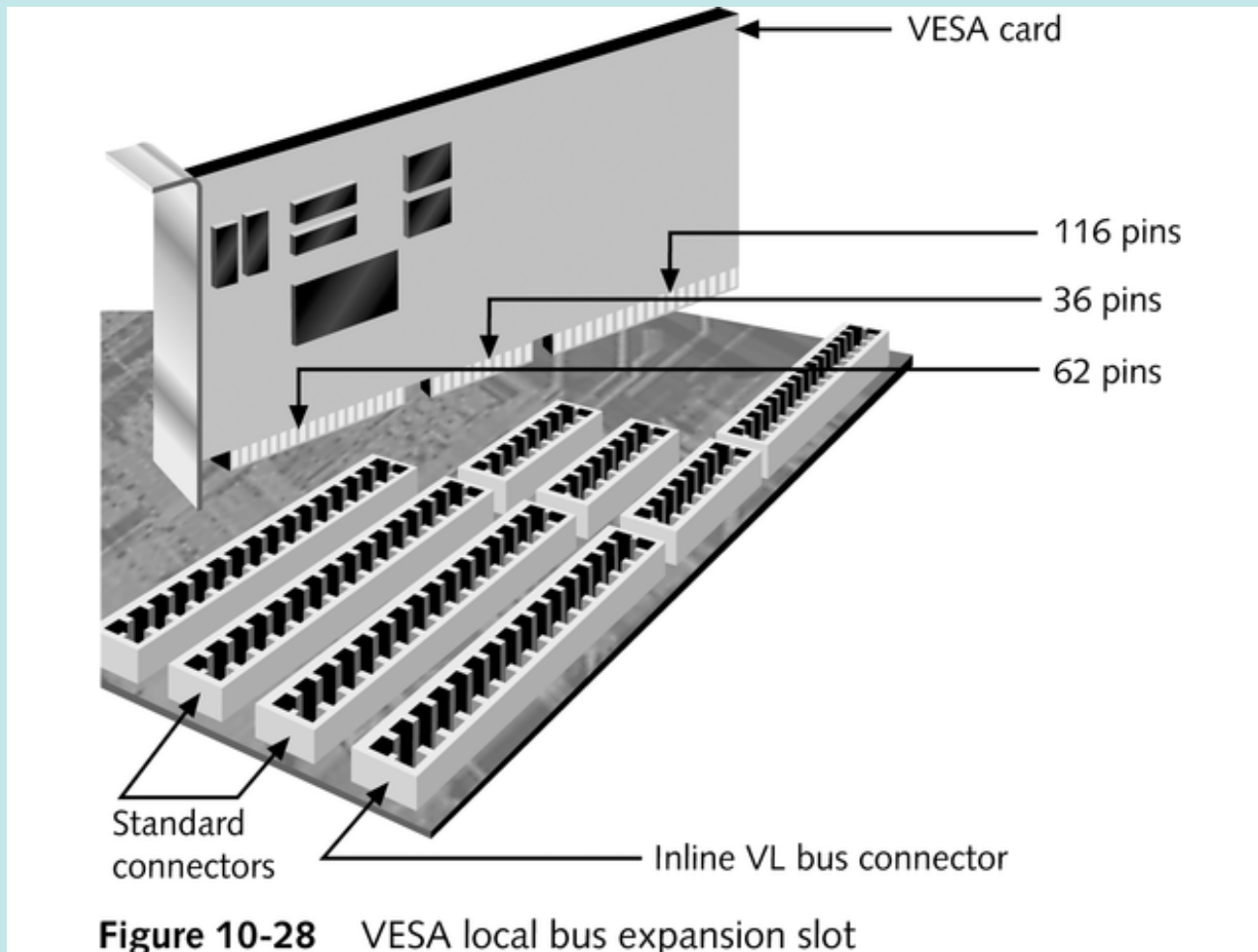


Các kiểu Bus được sử dụng cho Card video

- Bus VESA (Video Electronics Standards Association)
- Bus PCI
- Bus AGP



VESA Bus



AGP Bus

dùng với card gia tốc đồ họa

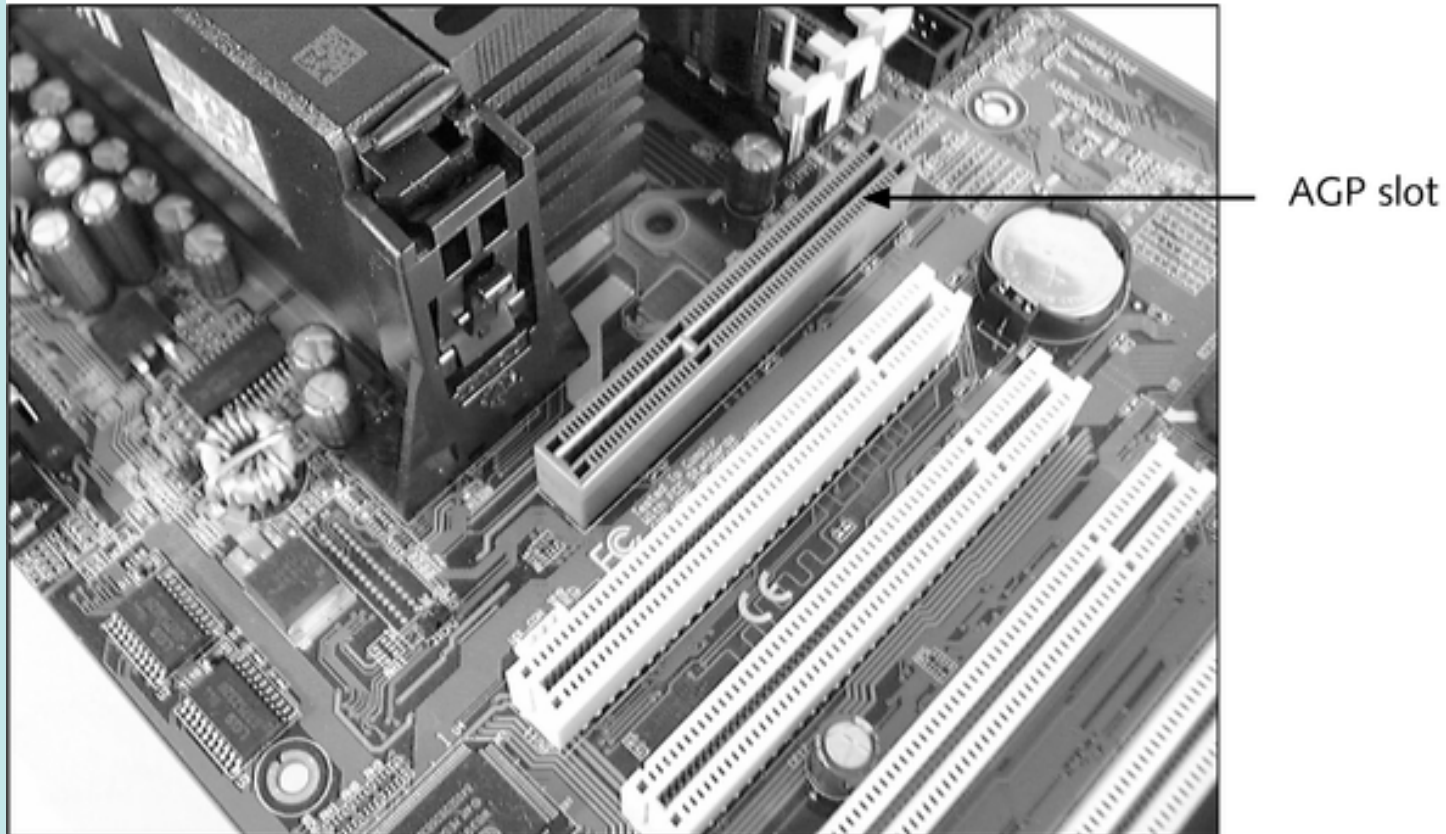


Figure 10-29 A motherboard has only one AGP slot, which is used to support a video/graphics card

Card gia tốc đồ hoạ

- Card video có bộ xử lý riêng để tăng cường tính năng hiển thị thông tin của PC
- Giảm bớt gánh nặng cho CPU trong việc:
 - Giải mã MPEG
 - Đồ hoạ 3-D
 - Cổng kép
 - Chuyển đổi không gian màu
 - Interpolated scaling
 - EPA Green PC support

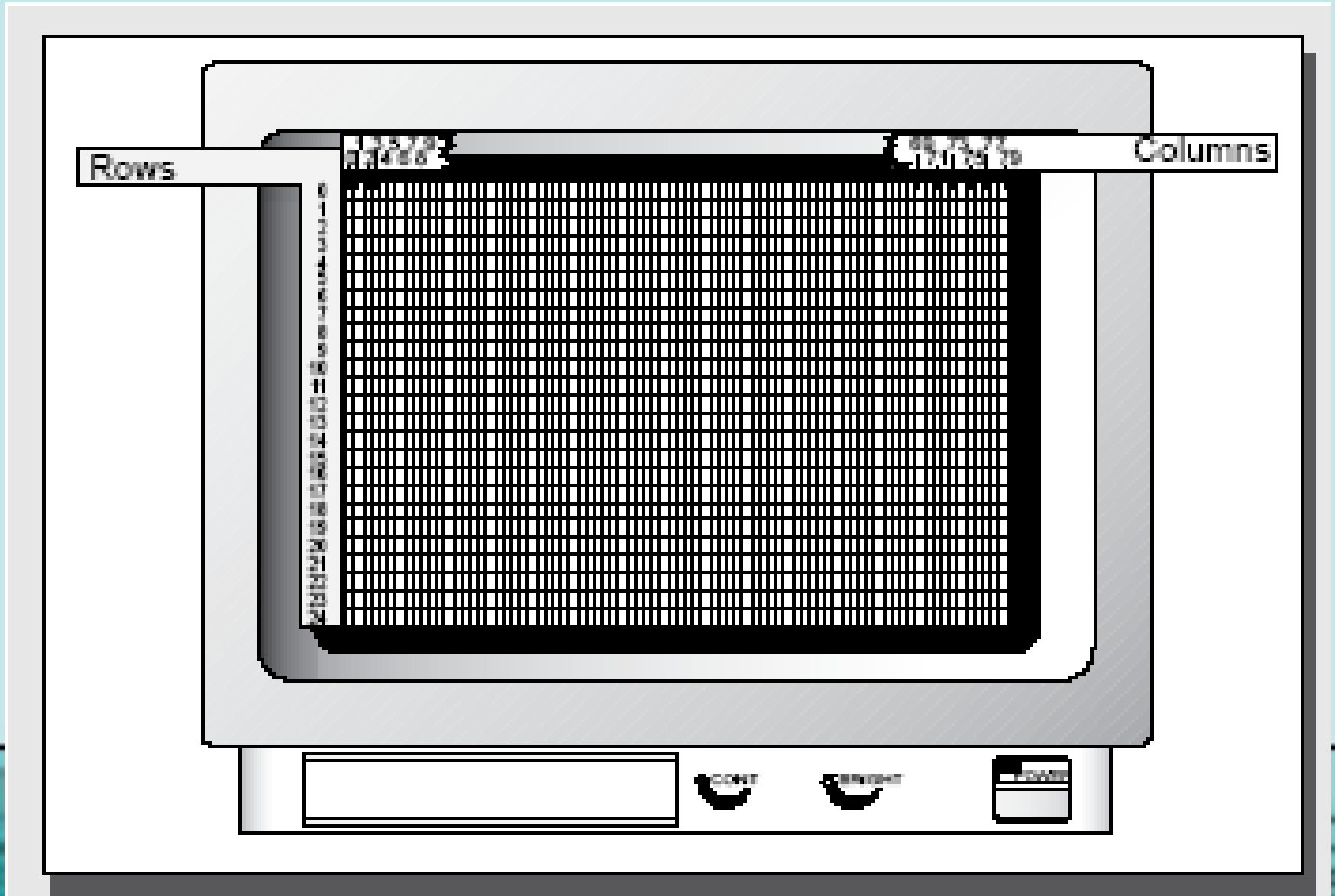


Bộ nhớ Video (RAM video)

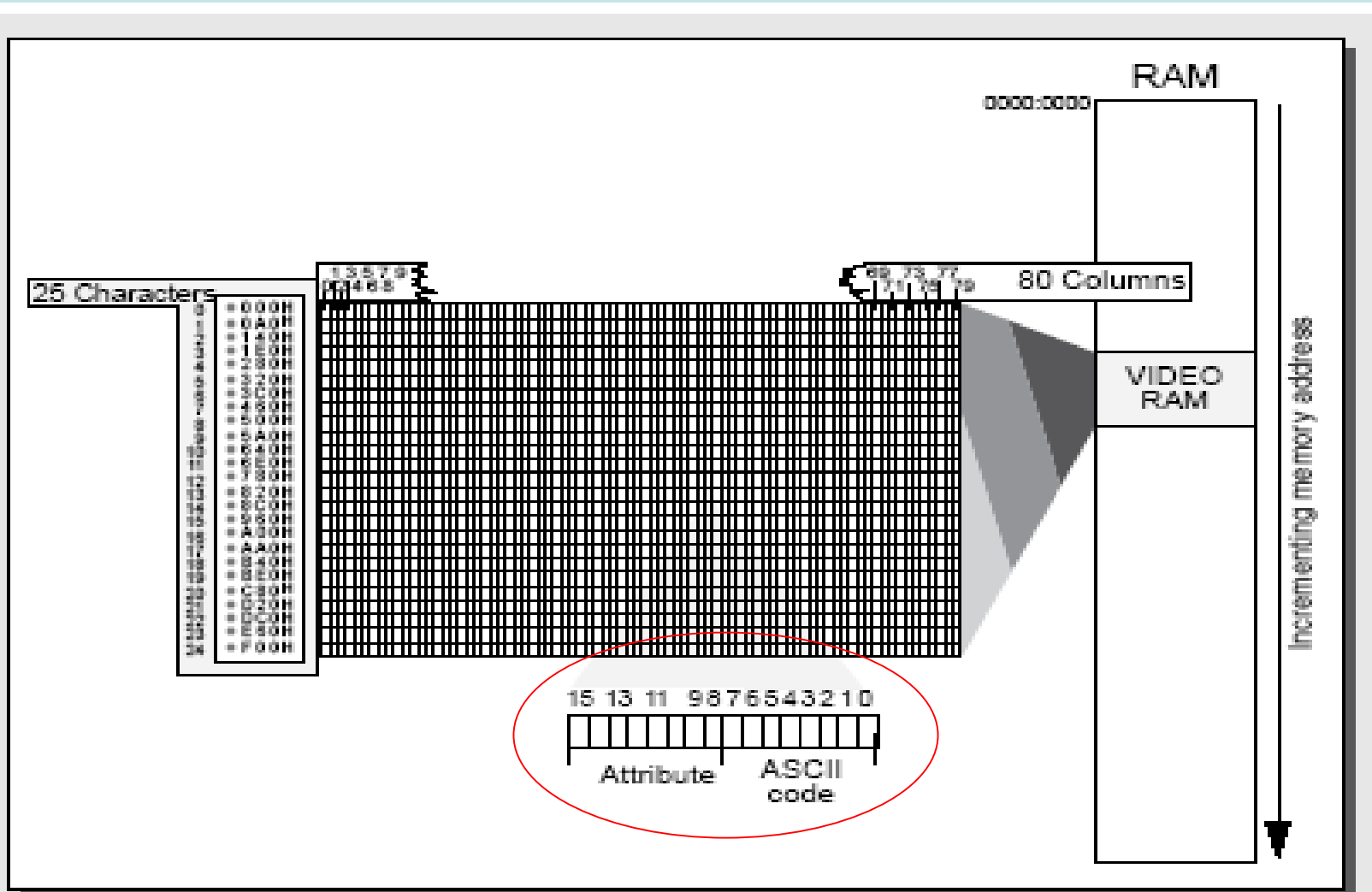
- Định vị trên card video, có nhiều loại:
- VRAM (Video RAM)
- SGRAM (synchronous graphics RAM)
- WRAM (window RAM)
- 3-D RAM



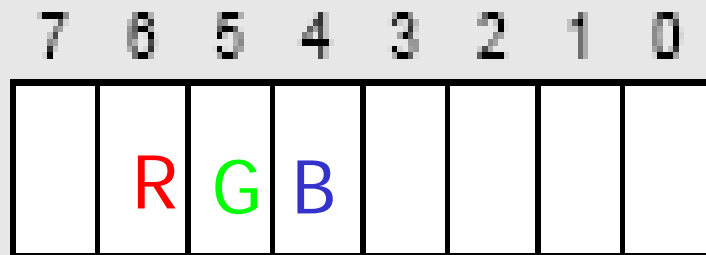
Khi hiển thị văn bản, màn hình là tập hợp các vị trí ký tự



Tổ chức của RAM video ở mode văn bản: Tốn 2 byte cho một vị trí ký tự gồm byte chứa mã ASCII và byte thuộc tính



Khuôn dạng của byte thuộc tính



Attribute byte structure
Graphics Adapter (CGA)

- ▶ Character color
- ▶ Background color
- ▶ Blinking (or background intensity)
0 = off
1 = on

Màu của ký tự

Color/Graphics Adapter color palette

Decimal	Hex	Bin	Color	Decimal	Hex	Bin	Color
0	00H	0000(b)	Black	8	08H	1000(b)	Dark gray
1	01H	0001(b)	Blue	9	09H	1001(b)	Light blue
2	02H	0010(b)	Green	10	0AH	1010(b)	Light green
3	03H	0011(b)	Cyan	11	0BH	1011(b)	Light cyan
4	04H	0100(b)	Red	12	0CH	1100(b)	Light red
5	05H	0101(b)	Purple	13	0DH	1101(b)	Light purple
6	06H	0110(b)	Brown	14	0EH	1110(b)	Yellow
7	07H	0111(b)	Light gray	15	0FH	1111(b)	White

Độ phân giải và Độ sâu màu

- Độ phân giải $m \times n$
 m = số lượng pixel theo chiều ngang
 n = số lượng pixel theo chiều đứng
- Độ phân giải càng cao thì hình ảnh càng mịn
- Độ sâu màu: D màu hoặc b bit
 D = số lượng màu mà pixel có thể hiển thị
 $b = \log_2 D$ = số bit dùng để biểu diễn màu của pixel
- Dung lượng RAM video cần thiết = $m \times n \times b$ (bit)



Outline Slides

- 70 [Thumbnail]
- 71 [Thumbnail]
- 72 [Thumbnail]
- 73 [Thumbnail]
- 74 [Thumbnail]
- 75 [Thumbnail]
- 76 [Thumbnail]
- 77 [Thumbnail]

Độ phân giải và Độ sâu màu


- Độ phân giải
 - $m = \text{số lượng}$
 - $n = \text{số lượng}$
- Độ phân
- Độ sâu n
 - $D = \text{số l}$
 - $b = \log_2 I$
- Dung lư

pixel (bit)

Colors

Standard Custom

Colors:



OK Cancel

Font: Times I

Effect

New Current

Slide Layout

Apply slide layout:

Text Layouts

Content Layouts

Text and Content Layouts

Show when inserting new slides

Một số ví dụ

Expanded graphic modes of various Super VGA cards

Resolution	Colors	Pixels	Memory	Resolution	Colors	Pixels	Memory
512x480	256	245,760	256K	720x540	256	388,800	512K
640x400	256	256,000	256K	752x410	16	308,320	256K
640x480	256	307,200	512K	800x600	16	480,000	256K
720x396	16	285,120	256K	800x600	256	480,000	512K
720x512	16	368,640	256K	960x720	16	691,200	512K
720x512	256	368,640	512K	1024x768	16	786,432	512K
720x540	16	388,800	256K	1024x768	256	786,432	1 Meg

Các nhà sản xuất card Video

Table 10-9 Video card manufacturers

Manufacturer	Web Site
ASUSTeK Computer, Inc.	www.asus.com
ATI Technologies, Inc.	www.ati.com
Creative Technology, Ltd.	www.creative.com
Gainward Co., Ltd.	www.gainward.com
Hercules Computer Technology	www.hercules.com
Matrox Graphics, Inc.	www.matrox.com
MSI Computer Corporation	www.msicomputer.com
nVidia	www.nvidia.com
VisionTek	www.visiontek.com



Tóm tắt chương

- Vấn đề cài đặt và hỗ trợ các thiết bị I/O
- Thủ tục cài đặt chung
- Cài đặt thiết bị I/O dùng cổng nối tiếp, song song, USB, IEEE 1394 và các khe cắm mở rộng
- Các thiết bị I/O cơ bản của PC: bàn phím, chuột và hệ thống con video



Thi học kỳ

- Thi trắc nghiệm trên máy tính
- 40 câu hỏi
- Mỗi câu hỏi có 5 trả lời: A,B,C,D,E
- Chỉ được phép chọn 1 trả lời
- Thời gian 60 phút

Good Luck to You!

PHẦN I

GIỚI THIỆU KHÁI QUÁT VỀ MÁY TÍNH VÀ CÁC THIẾT BỊ NGOẠI VI

§.1. TỔNG QUAN VỀ CẤU TRÚC MÁY VI TÍNH

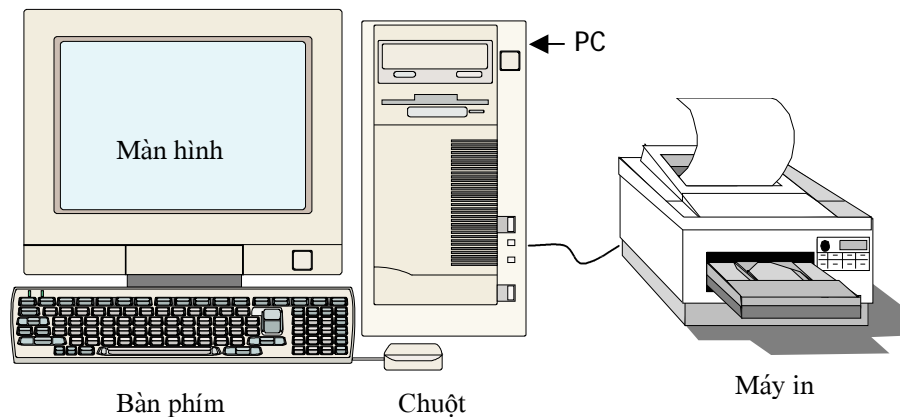
I. Cấu trúc chung của máy vi tính

Máy vi tính là một hệ thống được ghép nhiều thành phần tạo nên. Do đó, để máy tính có thể hoạt động được ta phải lắp ghép các thành phần của nó một cách hợp lý và khai báo với các thành phần khác. Ngày nay ngành tin học dựa trên các máy tính hiện đang phát triển trên cơ sở hai phần:

Phần cứng: Gồm những đối tượng vật lý hữu hình như vi mạch, bản mạch in, dây cáp nối mạch điện, bộ nhớ, màn hình, máy in, thiết bị đầu cuối, nguồn nuôi,... Phần cứng thực hiện các chức năng xử lý thông tin cơ bản ở mức thấp nhất tức là các tín hiệu nhị phân.

Phần mềm: Là các chương trình (Program) điều và phối tác các hoạt động phần cứng của máy vi tính và chỉ đạo việc xử lý số liệu. Phần mềm của máy tính có thể chia thành hai loại: Phần mềm hệ thống (System Software) và phần mềm ứng dụng (Applications software). Phần mềm hệ thống khi được đưa vào bộ nhớ chính, nó chỉ đạo máy tính thực hiện các công việc. Phần mềm ứng dụng là các chương trình được thiết kế để giải quyết một bài toán hay hay một vấn đề cụ thể để đáp ứng một nhu cầu riêng trong một số lĩnh vực.

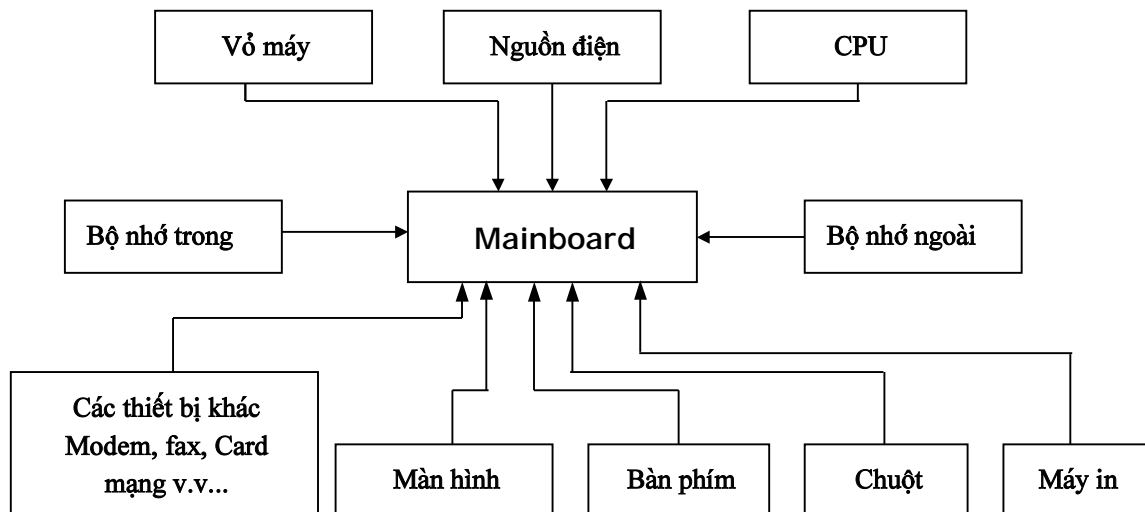
Máy tính các nhân PC (Personal Computer): Thệp đúng tên gọi của nó là máy tính có thể được sử dụng bởi riêng một người.



Hình 1

Hình 1 là một hệ thống máy vi tính thường được sử dụng. Phần trung tâm là máy PC, nó gồm có: Bộ xử lý dữ liệu, đĩa cứng (HDD), đĩa mềm (FDD), CDROM, các mạch ghép nối... Bên ngoài có bàn phím (Key board), màn hình (Monitor), chuột (Mouse), máy in (Printer).

II. Các thành phần cơ bản của máy vi tính



Sơ đồ tổng quan về cấu trúc máy tính

1. Vỏ máy: Là nơi để gắn các thành phần của máy tính thành khối như nguồn, Mainboard, card v.v... có tác dụng bảo vệ máy tính.

2. Nguồn điện: Cung cấp hầu hết hệ thống điện cho các thiết bị bên trong máy tính.

3. Mainboard: Có chức năng liên kết các thành phần tạo nên máy tính và là bảng mạch lớn nhất trên máy vi tính.

4. CPU (Central Processing Unit): Bộ vi xử lý chính của máy tính.

5. Bộ nhớ trong (ROM, RAM): Là nơi lưu trữ dữ liệu và chương trình phục vụ trực tiếp cho việc xử lý của CPU, nó giao tiếp với CPU không qua một thiết bị trung gian.

6. Bộ nhớ ngoài: Là nơi lưu trữ dữ liệu và chương trình gián tiếp phục vụ cho CPU, bao gồm các loại: đĩa mềm, đĩa cứng, CDROM, v.v... Khi giao tiếp với CPU nó phải qua một thiết bị trung gian (thường là RAM) hay gọi là ngắt.

7. Màn hình: Là thiết bị đưa thông tin ra giao diện trực tiếp với người dùng. Đây là thiết bị xuất chuẩn của máy vi tính hay còn gọi là bộ trực (Monitor).

8. Bàn phím (Keyboard): Thiết bị nhập tin vào giao diện trực tiếp với người dùng. Đây là thiết bị nhập chuẩn của máy vi tính.

9. Chuột (Mouse): Thiết bị điều khiển trong môi trường đồ họa giao diện trực tiếp với người sử dụng.

10. Máy in (Printer): Thiết bị xuất thông tin ra giấy thông dụng nhất.

11. Các thiết bị như Card mạng, Modem, máy Fax,... phục vụ cho việc lắp đặt mạng máy tính và các chức năng khác.

----- § - > -----

§.2. NGUỒN ĐIỆN CHO MÁY TÍNH

Nguồn điện máy tính có chức năng chuyển đổi nguồn điện 110V/220V thành nguồn điện một chiều ± 3 , 3V, ± 5 V và ± 12 V cung cấp cho toàn bộ hệ thống máy tính. Công suất trung bình của bộ nguồn hiện nay khoảng 200W. Công suất tiêu thụ một số thành phần như sau:

Mainboard	:	20W - 35W.
CD-ROM	:	20W - 25W
Ổ đĩa mềm	:	5W - 15W.
Ổ đĩa cứng	:	5W - 15W.
Ram	:	5W /MB.
Card	:	5W - 15W.
CPU	:	Tùy theo mức độ làm việc nhiều hay ít.

Các số liệu trên đây chỉ mang tính chất tham khảo, bởi vì hiện nay xu thế các hãng sản xuất đưa ra các thiết bị tiêu thụ điện năng nhỏ. Bên cạnh đó, tùy thuộc vào số lượng thiết bị mà máy tính sử dụng nhiều hay ít điện năng.

Hiện nay, máy vi tính cá nhân thường sử dụng hai loại bộ nguồn điện là AT và ATX. Sau đây, ta xét cho thành phần của nguồn AT còn ATX tương tự.

Có thể chia đầu ra nguồn điện máy tính thành hai loại như sau:

1. Phích dùng cho main board: Gồm 12 dây chia thành 2 phích cắm có cấu trúc như sau:

Dây	Màu	Tín hiệu
1	Gạch	Điều chỉnh
2	Đỏ	+5V

3	Vàng	+12V
4	Xanh	-12V
5	Đen	Nối đất
6	Đen	Nối đất
7	Đen	Nối đất
8	Đen	Nối đất
9	Trắng	-5V
10	Đỏ	+5V
11	Đỏ	+5V
12	Đỏ	+5V

* Quy tắc cắm vào mainboard: Một số mainboard có ghi rõ từ chân 1 đến chân 12, cứ thế ta cắm cho đúng vào khe cắm trên mainboard.

2. Phích dùng cho các thành phần khác: Là loại phích 4 dây thường dùng cho ổ đĩa cứng, ổ đĩa mềm, CDROM v.v..., cấu trúc của loại này như sau:

Chân	Màu	Tín hiệu
1	Đỏ	+5V
2	Đen	Nối đất
3	Đen	Nối đất
4	Vàng	+12V

Thông thường, ta cắm phích điện vào đúng ơg của phích cắm điện của thiết bị. Nếu phích hoặc thiết bị không có ơg thì ta phải cắm đúng số hiệu chân có ghi trên thiết bị.

Khi có nghi ngờ về bộ nguồn của máy tính như điện không ổn định ta dễ dàng kiểm tra bộ nguồn bằng cách dùng đồng hồ đo điện.

Thực tế, hiện nay có loại nguồn ATX có nhiều chức năng như có thể tự ngắt điện khi máy tính thoát khỏi Windows 95 trở về sau. Song về cấu trúc, cách cắm của chúng cơ bản là giống loại nguồn AT ở trên, chỉ khác ở phích cắm vào mainboard có 20 dây và có dây -3,3V và +3,3V. Sau đây là sơ đồ chân của phích cắm của nguồn ATX:

Dây	Màu	Tín hiệu	Dây	Màu	Tín hiệu
1	Gạch	+3,3V	11	Gạch	+3,3
2	Gạch	+3,3V	12	Xanh sẫm	-12V
3	Đen	Nối đất	13	Đen	Nối đất
4	Đỏ	+5V	14	Xanh lá	PW_ON
5	Đen	Nối đất	15	Đen	Nối đất
6	Đỏ	+5V	16	Đen	Nối đất
7	Đen	Nối đất	17	Đen	Nối đất
8	Xám	PWRGOOD	18	Trắng	-5V
9	Tím	+5VS	19	Đỏ	+5V
10	Vàng	+12V	20	Đỏ	+5V

----- § - > -----

§.3. BẢNG MẠCH CHÍNH (MAINBOARD)

I. Giới thiệu về bảng mạch chính

Đây là bảng mạch lớn nhất trong máy vi tính. Mainboard có chức năng liên kết và điều khiển các thành phần được cắm vào nó. Đây là cầu nối trung gian cho quá trình giao tiếp của các thiết bị được cắm vào mainboard.

Khi có một thiết bị yêu cầu được xử lý thì nó gửi tín hiệu qua mainboard, ngược lại, khi CPU cần đáp ứng lại cho thiết bị nó cũng phải thông qua mainboard. Hệ thống làm công việc vận chuyển trong mainboard gọi là bus, được thiết kế theo nhiều chuẩn khác nhau.

Một mainboard cho phép nhiều loại thiết bị khác nhau với nhiều thế hệ khác nhau cắm trên nó. Ví dụ: một mainboard cho phép nhiều thế hệ của CPU cắm vào nó (Xem Catalogue đi cùng mainboard để biết chi tiết nó tương thích với các loại CPU nào).

Mainboard có rất nhiều loại do nhiều hãng sản xuất khác nhau như Intel, Compact v.v..., mỗi hãng sản xuất có những đặc điểm riêng cho loại mainboard

của mình. Nhưng nhìn chung chúng có các thành phần và đặc điểm giống nhau, ta sẽ khảo sát các thành phần trên mainboard trong mục sau.

II. Các thành phần cơ bản trên Mainboard

1. Khe cắm CPU: Có hai loại cơ bản là Slot và Socket.

- Slot: Là khe cắm dài như một thanh dùng để cắm các loại CPU đời mới như Pentium II, Pentium III, Pentium Pro, loại này chỉ có trên các mainboard mới. Khi ấn CPU vào Slot còn có thêm các vít để giữ chặt CPU.

- Socket: là khe cắm hình chữ nhật có xăm lỗ để cắm CPU vào. Loại này dùng cho tất cả các loại CPU còn lại không cắm theo Slot. Hiện nay, đa số CPU dùng Socket 7, Socket 370 (có vát 1 chân). Một số ít CPU đời cũ dùng Socket 4, Socket 3 (đủ chân).

2. Khe cắm RAM: Thường có hai loại chính DIMM và SIMM. Ngoài ra, còn có các loại DIMM RAM, SIMM RAM thường được gắn sẵn đi cùng với mainboard.

- DIMM: Loại khe RAM có 168 chân dùng cho loại 16 MB trở lên.

- SIMM: Loại khe cắm 72 chân dùng cho các loại còn lại.

Hiện nay có rất nhiều loại mainboard có cả hai loại khe SIMM và DIMM trên nên rất tiện cho việc nâng cấp và sử dụng lại RAM cũ.

3. Bus: Là đường dẫn thông tin trong bảng mạch chính, nối từ vi xử lý đến bộ nhớ và các thẻ mạch, khe cắm mở rộng. Bus được thiết kế theo nhiều chuẩn khác nhau như PCI, ISA, EISA, VESA v.v...

4. Khe cắm bộ điều hợp: Dùng để cắm các bộ điều hợp như Card màn hình, Card mạng, Card âm thanh v.v... Chúng cũng gồm nhiều loại được thiết kế theo các chuẩn như ISA, EISA, PCI v.v...

+ ISA (Industry Standard Architecture): Là khe cắm card dài dùng cho các card làm việc ở chế độ 16 bit.

+ EISA (Extended Industry Standard Architecture): Là chuẩn cải tiến của ISA để tăng khả năng giao tiếp với Bus mở rộng và không qua sự điều khiển của CPU.

+ PCI (Peripheral Component Interface): là khe cắm ngắn dùng cho loại Card 32 bit.

5. Khe cắm IDE (Integrated Drive Electronics): Có hai khe cắm dùng để cắm cáp đĩa cứng và CDROM.

6. Khe cắm Floppy: Dùng để cắm cáp ổ đĩa mềm.

7. Cổng nối bàn phím.

8. Các khe cắm nối tiếp (thường là COM1 và COM2): sử dụng cho các thiết bị nối tiếp như: chuột, modem v.v... Các bộ phận này được sự hỗ trợ của các chip truyền nhận không đồng bộ vạn năng UART (Universal Asynchronous Receiver Transmitter) được cắm trực tiếp trên mainboard để điều khiển trao đổi thông tin nối tiếp giữa CPU với các thiết bị ngoài. Các chip này thường có tên Intel 8251, 8250 hay motorola 6821, 6530 v.v...

9. Các khe cắm song song (thường là LPT1 và LPT2): Dùng để cắm các thiết bị giao tiếp song song như máy in.

10. Khe cắm điện cho mainboard thường có hai khe, một dùng cho loại nguồn AT và một dùng cho loại ATX.

11. Các ROM chứa các chương trình hỗ trợ khởi động và kiểm tra thiết bị. Tiêu biểu là ROM BIOS chứa các trình điều khiển, kiểm tra thiết bị và trình khởi động máy.

12. Các chip DMA (Direct Memory Access): Đây là chip truy cập bộ nhớ trực tiếp, giúp cho thiết bị truy cập bộ nhớ không qua sự điều khiển của CPU.

13. Pin và CMOS lưu trữ các thông số thiết lập cấu hình máy tính gồm cả RTC (Real Time Clock - đồng hồ thời gian thực).

14. Các thành phần khác như thỏi dao động thạch anh, chip điều khiển ngắt, chip điều khiển thiết bị, bộ nhớ Cache v.v... cũng được gắn sẵn trên mainboard.

15. Các Jump thiết lập các chế độ điện, chế độ truy cập, đèn báo v.v... Trong một số mainboard mới, các Jump này được thiết lập tự động bằng phần mềm.

Mặc dù được thiết kế tích hợp nhiều phần nhưng được sản xuất với công nghệ cao, nên khi bị hỏng một bộ phận thường phải bỏ nguyên cả mainboard.

III. Các loại Mainboard thường được sử dụng hiện nay

Loại Mainboard	CPU được hỗ trợ	Khe cắm RAM	Các chip phụ trợ
SQ594(Có jump)	Socket7: Intel P54C(100-200MHz), P55C(166-233MHz). AMDK5(100-200MHz), AMDK6 (PR 166, PR200, PR233). Cyrix	4 SIMM, 2 DIMM	Intel 82371SB,82437VX, 82438 VX, Cache, BIOS

	6x86, 6x86 L/M2		
CE (Có jump)	Socket7: Intel(P75 - P200, P166MMX, P200MMX). AMDK5 (K5PR75 - K5PR 166), AMDK6(K6PR 166, K6PR200, K6PR233) Cyrix(M1PR 120, 6x86 L/M2	4SIMM, 1DIMM	Intel SB 82371, 82437VX, 82438 VX, Cache, BIOS
Intel i430 VX2 (Có jump)	Socket7: Intel(P75 - P200). AMDK5 (K5PR90 - K5PR 166), Cyrix(M1PR 150,166)	4SIMM,	Intel SB 82371, 82437VX, 82438 VX, Cache, BIOS
TXPRO II (Có jump)	Socket7: Intel(P54 - P200, P166MMX, P200MMX). AMDK5 (K5PR75 - K5PR 166), AMDK6(K6PR 166, K6PR200, K6PR233) Cyrix(M1PR 120, 6x86 L/M2).	4SIMM, 2DIMM	Intel SB 82371, 82437VX, 82438 VX, Cache, BIOS
ACORP-5TX29 (Có jump)	Socket7: Intel(P90 - P200, P166MMX, P200MMX). AMDK5 (K5PR75 - K5PR 166), AMDK6(K6PR 166, K6PR200, K6PR233) Cyrix(M1PR 120, 6x86 L/M2)	4SIMM, 2DIMM	Intel 82371AB, 82439TX, Cache, BIOS
SP-PIII LXB/EXB (Có jump)	Slot: Intel Pentium II(166, 333).	3DIMM	Intel FW82443LX, FX82371AB, Cache, FlashBIOS
EX-98 (Auto jump)	Slot: Intel Pentium II(166, 333).	2DIMM	Intel 440EX, 82371EX, Cache, FlashBIOS



LX-98 (Auto jump)	Socket370: Celeron.	2DIMM	Intel440LX, 82371LX, Cache, FlashBIOS
-------------------	---------------------	-------	---------------------------------------

Như vậy, một mainboard có thể hỗ trợ nhiều CPU khác nhau có tốc độ khác nhau nên ta có thể nâng cấp chúng bằng cách tra loại CPU tương thích với loại mainboard đó.

----- § - > -----

§.4. CPU (CENTRAL PROCESSING UNIT)

I. Giới thiệu về CPU

Đây là bộ não của máy tính, nó điều khiển mọi hoạt động của máy tính. CPU liên hệ với các thiết bị khác qua mainboard và hệ thống cáp của thiết bị. CPU giao tiếp trực tiếp với bộ nhớ RAM và ROM, còn các thiết bị khác được liên hệ thông qua một vùng nhớ (địa chỉ vào ra) và một ngắt thường gọi chung là cổng.

Khi một thiết bị cần giao tiếp với CPU nó sẽ gửi yêu cầu ngắt (Interrupt Request - IRQ) và CPU sẽ gọi chương trình xử lý ngắt tương ứng và giao tiếp với thiết bị thông qua vùng địa chỉ quy định trước. Chính điều này dẫn đến khi ta khai báo hai thiết bị có cùng địa chỉ vào ra và cùng ngắt giao tiếp sẽ dẫn đến lỗi hệ thống (xung đột ngắt - IRQ Conflict) có thể làm treo máy.

Ngày nay với các thế hệ CPU mới có khả năng làm việc với tốc độ cao và bus dữ liệu rộng giúp cho việc xây dựng chương trình đa năng ngày càng dễ dàng hơn.

Để đánh giá các CPU, người ta thường căn cứ vào các thông số của CPU như tốc độ, độ rộng của bus, độ lớn của Cache và tập lệnh được CPU hỗ trợ. Tuy nhiên rất khó có thể đánh giá chính xác các thông số này, do đó người ta vẫn thường dùng các chương trình thử trên cùng một hệ thống có các CPU khác nhau để đánh giá các CPU.

II. Phân loại CPU

Hiện nay, trên thị trường có rất nhiều loại CPU do nhiều hãng sản xuất khác nhau với các tốc độ và khả năng khác nhau dẫn đến giá cả của chúng cũng khác nhau. Ta có thể phân loại CPU theo 2 cách như sau:

1. Phân loại theo đời

1. Các CPU đời cũ như 8080, 8086, 8088 là các bộ vi xử lý cơ sở cho các vi xử lý sau này. Do giới hạn về khả năng quản lý bộ nhớ, số bit dữ liệu cũng như tốc độ nên loại này hiện nay không được dùng nữa mà nhường cho các thế hệ sau.

2. Các CPU 80286, 80386, 80486: Có nhiều đột phá so với thế hệ trước trong việc quản lý bộ nhớ như sử dụng bộ nhớ mở rộng, đáp ứng các chương trình đa nhiệm, hỗ trợ bộ đồng xử lý giúp cho việc xử lý các phép toán động có hiệu quả.

3. Các CPU Pentium như Pentium I, Pentim II, Celeron, AMDK5 v.v... Đây là các CPU được sử dụng rộng rãi hiện nay. Chúng có nhiều ưu điểm về tốc độ, bus dữ liệu và đáp ứng được nhiều chương trình đồ họa có tính đa nhiệm cao.

4. Các CPU đời mới: Gần đây, Intel đã cho ra đời Pentium III, IV với tốc độ lên đến 2.6 GHz, hoặc AMDK6 v.v... Có nhiều ưu điểm về công nghệ cao, tốc độ xử lý cao, song giá thành của chúng giảm đi rất nhiều do có nhiều hãng sản xuất cạnh tranh với nhau.

2. Phân loại theo hãng sản xuất:

Có rất nhiều hãng sản xuất CPU, song ta có thể phân loại theo các hãng sản xuất chính mà CPU của họ được dùng rộng rãi hiện nay như sau:

Nhà sản xuất	Các CPU tương ứng
Intel	Đời trước: 8080,8086,8088,80286,80386,80484SX,80486DX v.v... PentiumI:(PR 75- PR 166, PR 166MMX- PR 233 MMX) PentiumII:(266 - 450), Celeron v.v... Pentium III, IV.
AMD	K5 (PR75 - PR166) K6 (PR166 -PR 233)
Cyrix/IBM	M1: PR120, PR133, PR150, PR166, PR200, PR200L M2: PR166, PR200, PR233

III. Cách cắm CPU vào Mainboard và thiết lập các thông số

Như đã giới thiệu ở phần trên, hiện nay có hai tiêu chuẩn chính để gắn CPU vào Mainboard là Socket và Slot. Song riêng mỗi loại mainboard cũng chỉ cho phép với một số loại CPU nhất định nào đó (Điều này phải tham khảo trong Catalogue đi kèm với mainboard).

Khi CPU làm việc, nó tỏa lượng nhiệt tương đối lớn, do đó bị nóng lên. Chính vì vậy nó thường được lắp kèm với bộ giải nhiệt hoặc quạt để làm giảm nhiệt trong quá trình hoạt động xử lý.

Sau khi tham khảo các tham số của CPU cho phép cắm lên mainboard đạt yêu cầu, ta tiến hành đi vào thiết lập các thông số làm việc cho nó. Đây là bước quan trọng vì nếu thiết lập không đúng các thông số cho CPU sẽ là giảm khả năng làm việc, giảm tuổi thọ cũng như có thể làm cháy CPU. Một CPU thường có hai thông số chính phải thiết lập là mức điện áp tiêu thụ và hệ số ratio.

- Mức điện áp tiêu thụ: là mức điện áp cần thiết cho CPU làm việc, nó thường được ghi trực tiếp trên mặt CPU. Nếu thiết lập mức điện áp dưới mức này CPU không làm việc, nếu trên sẽ làm cháy CPU. Hiện nay mức này cho các CPU thường là 2,8V - 3,3V.

- Hệ số Ratio: Là hệ số đồng bộ giữa tốc độ CPU (tính bằng giao động đồng hồ tinh thể) và tốc độ mainboard (tính bằng giao động thạch anh) để điều khiển đồng hồ gõ nhịp đồng bộ (Ví dụ: CPU có tốc độ 200 MHz, Mainboard 66 MHz thì hệ số này là 3 vì $66 \times 3 \approx 200$). Hệ số này thường cũng được ghi trực tiếp trên CPU.

Để thiết lập 2 hệ số trên ta phải tra trên Catalogue của mainboard để tìm ra các CPU được hỗ trợ cùng với cách cắm của các Jump trên mainboard. Sau đó, truy tìm các Jump trên mainboard để cắm cho đúng. Một số mainboard mới hiện nay có chức năng Autojump sẽ tự động xác định các thông số điện áp ta chỉ chọn thông số Ratio cho phù hợp theo hình thức *Tốc độ CPU/Tốc độ mainboard* trong CMOS.

IV. Ngắt (Interrupt Request)

Như ta đã biết CPU làm việc với nhiều thiết bị khác nhau. Song tại một thời điểm thì CPU chỉ phục vụ được một công việc nhất định nào đó. Do đó, để CPU có thể kiểm soát và phục vụ tốt tất cả các thiết bị thì khi một thiết bị có yêu cầu xử lý nó sẽ gọi CPU ngừng các công việc khác để phục vụ cho mình, quá trình trên gọi là ngắt. Khi làm việc, mỗi thiết bị được đặt trưng bởi một chỉ số ngắt nào đó mà thôi. Như vậy, ngắt làm việc như thế nào và có bao nhiêu loại ngắt ?

Khi có yêu cầu ngắt thì CPU sẽ lưu giữ các thông số của chương trình đang thực hiện, sau đó gọi chương trình xử lý ngắt để thực hiện đáp ứng. Địa chỉ của chương trình xử lý ngắt được chứa trong bảng Vector ngắt. Khi thực hiện xong ngắt, CPU khôi phục lại lệnh của chương trình đang thực hiện dở để tiếp

tục. Bộ xử lý ngắt phần cứng có thể được thực hiện bởi một chip được gắn trực tiếp trên mainboard hay trên thiết bị gọi là chip xử lý ngắt.

Ngắt có nhiều cách phân loại như ngắt cứng và ngắt mềm, ngắt che được và không che được, ngắt trong và ngắt ngoài v.v... Do khi định nghĩa người ta nhìn theo các góc độ khác nhau. Ở đây ta chỉ định nghĩa theo cách ngắt cứng và ngắt mềm.

+ Ngắt cứng là ngắt được tạo ra bởi các phần cứng như ngắt bàn phím, chuột v.v... Ngắt cứng thường do một chip xử lý ngắt của thiết bị tạo ra mỗi khi có yêu cầu phục vụ.

+ Ngắt mềm là do chương trình tạo ra như các ngắt để yêu cầu dữ liệu, ngắt chia cho 0 v.v...

Ngoài ra, khi giao tiếp với các thiết bị ngoài, CPU còn quy định vùng trao đổi dữ liệu cho thiết bị gọi là vùng nhớ vào ra (I/O - Input/Output). Do đó, mỗi hệ thống có một sơ đồ bộ nhớ cơ bản vào ra (I/O Map). Sơ đồ này có thể do người sử dụng khai báo hoặc máy tính tự động cập nhật. Sự kết hợp giữa ngắt và địa chỉ bộ nhớ vào ra cho một thiết bị gọi là cổng của thiết bị đó.

----- § - > -----

§.5. BỘ NHỚ TRONG (RAM & ROM)

I. Giới thiệu về bộ nhớ trong

Xét trong giới hạn bộ nhớ gắn trên mainboard thì đây là bộ nhớ trực tiếp làm việc với CPU. Nó là nơi CPU lấy dữ liệu và chương trình để thực hiện, đồng thời cũng là nơi chứa dữ liệu để xuất ra ngoài.

Để quản lý bộ nhớ này người ta tổ chức gộp chúng lại thành nhóm 8 bits rồi cho nó một địa chỉ để CPU truy cập đến. Chính điều này khi nói đến dung lượng bộ nhớ, người ta chỉ đề cập đến đơn vị byte chứ không phải bit như ta đã biết. Bộ nhớ trong gồm 2 loại là ROM và RAM.

1. ROM (Read Only Memory): Đây là bộ nhớ mà CPU chỉ có quyền đọc và thực hiện chứ không có quyền thay đổi nội dung vùng nhớ. Loại này chỉ được ghi một lần với thiết bị ghi đặc biệt. ROM thường được sử dụng để ghi các chương trình quan trọng như chương trình khởi động, chương trình kiểm tra thiết bị v.v... Tiêu biểu trên mainboard là ROM BIOS.

Hiện nay, trên hầu hết các thiết bị đều có gắn ROM để phục vụ các chương trình cần thiết. ROM có nhiều loại với công nghệ khác nhau như EPROM, FROM, v.v...

2. RAM (Random Access Memory): Đây là bộ nhớ chính mà CPU giao tiếp trong quá trình xử lý dữ liệu của mình, bởi loại này cho phép ghi và xóa dữ liệu nhiều lần giúp cho việc trao đổi dữ liệu trong quá trình xử lý của CPU thuận lợi hơn.

RAM được tổ chức thành các byte xếp sát nhau và được đánh địa chỉ cho từng byte. Khi CPU ghi dữ liệu vào bộ nhớ, nó sẽ giữ giá trị ô nhớ đầu và độ dài ghi được để khi truy cập CPU tìm đến địa chỉ đầu của mục cần tìm và từ đó đọc tiếp các thông tin còn lại.

Khi thực hiện chương trình, CPU đọc chương trình và ghi lên bộ nhớ sau đó mới tiến hành thực hiện các lệnh. Ngày nay, các chương trình có kích thước rất lớn và yêu cầu dữ liệu càng lớn. Do đó, để máy tính thực hiện nhanh chóng yêu cầu phải có bộ nhớ RAM lớn và tốc độ truy cập RAM cao. Chính vì thế mà các hãng sản xuất mainboard và bộ nhớ không ngừng đưa ra các dạng RAM có tốc độ cao và có kích thước lớn.

II. Phân loại RAM

* Có nhiều cách để phân loại RAM. Nếu phân loại theo khe cắm trên mainboard thì RAM có các loại như sau:

- SIMM (Single Inline Module Memory): đây là loại RAM giao tiếp 72 chân được sử dụng nhiều ở các mainboard cũ, dung lượng mỗi thanh có thể là: 4MB, 8MB, 16MB, 32MB v.v...

- DIMM (Dual Inline Module Memory): Là chuẩn thanh RAM 168 chân có mặt ở các mainboard mới, các thanh này có kích thước 8 MB trở lên và được cắm vào khe DIMM trên mainboard.

- SIPRAM (Single Inline Pin Random Access Memory) và DIPRAM (Dual Inline Pin Random Access Memory): Đây là 2 loại RAM thường được cắm sẵn trên mainboard và thường có dung lượng nhỏ tính theo Kb. Các mainboard mới hiện nay không còn thấy các loại này.

- Cache: (Bộ nhớ khay) là bộ nhớ có tốc độ cực nhanh, làm việc trung gian giữa bộ nhớ và CPU nhằm để tăng tốc độ truy cập dữ liệu của CPU trong quá trình xử lý. Cache thường được phân biệt theo 2 loại là Cache nội (Internal Cache) được tích hợp trên CPU và Cache ngoại (External Cache) được gắn trên mainboard hay trên các thiết bị.

Trên các mainboard mới hiện nay hầu hết hỗ trợ cả DIMM lẫn SIMM rất thuận tiện cho quá trình nâng cấp RAM.

* Nếu phân loại theo công nghệ thì RAM có các loại như sau:

+ SRAM (Static Random Access Memory): Còn được gọi là RAM tĩnh, loại này có tốc độ cao nhưng độ linh hoạt kém, ngày nay ít được dùng riêng rẽ.

+ DRAM (Dynamic Random Access Memory): Còn gọi là RAM động, loại này làm việc linh động hơn nhưng độ ổn định không cao.

+ SDRAM (Static Dynamic Random Access Memory): Là loại kết hợp công nghệ của hai loại trên và được sử dụng rộng rãi hiện nay để chế tạo các thanh DIMM, SIMM .

+ EDORAM (Extended Data Out Dynamic Random Access Memory): Biểu thị cho việc sử dụng băng thông mở rộng dữ liệu, do vậy loại này có tốc độ nhanh hơn 25 % so với các loại tương đương cùng chuẩn.

Ngoài ra, còn có các loại RAM khác như Cache (Bộ nhớ khay) có tốc độ rất cao, làm nhiệm vụ trung gian của bộ nhớ và CPU để tăng tốc độ xử lý.

Khi cắm RAM nên cẩn thận, bởi vì nguyên nhân máy không khởi động do RAM rất hay gặp trong thực tế. Ngoài ra, tùy theo mức độ sử dụng các chương trình có kích thước lớn của chúng ta mà chọn cấu hình RAM cho phù hợp.

Ngày nay, được sự hỗ trợ của các chip DMA và cũng để tăng tốc độ của các thiết bị ngoại vi, hầu hết các Card thiết bị điều khiển gắn ROM và RAM riêng trên nó để tăng các tính năng làm việc, tốc độ giao tiếp.

III. Chip truy cập bộ nhớ trực tiếp (DMA - Direct Memory Access)

Ngày nay, kích thước của chương trình ngày càng lớn, số lượng xử lý của CPU càng nhiều. Do đó, nếu chỉ để CPU đơn phương thực hiện tất cả các công việc từ đầu đến cuối thì sẽ làm chậm hệ thống đi rất nhiều do phải chờ để truy cập cho các thành phần từ bên ngoài vào bộ nhớ trong. Để khắc phục điều này người ta đưa ra chip truy cập bộ nhớ trực tiếp, cho phép trao đổi dữ liệu giữa bộ nhớ trong với thiết bị ngoại vi mà không qua sự điều khiển của CPU. Các chip đó gọi là chip DMA.

Các chip DMA được gắn trên mainboard hay trên các thiết bị. Nó thường là các chip mang tên 8237, 82C37 v.v...

§.6. BỘ NHỚ NGOÀI (FLOPPY, HARD DISK, CDROM)

Trong phần trước ta đã khảo sát xong bộ nhớ trong của máy tính và thấy được chúng có ưu điểm về tốc độ rất lớn và làm việc trực tiếp với CPU. Tuy nhiên, chúng có giới hạn về dung lượng cũng như giá cả của nó cũng khá đắt. Hơn nữa, bộ nhớ RAM bị mất dữ liệu khi bị ngắt điện, còn ROM thì chỉ ghi được một lần. Để có thể lưu giữ dữ liệu và di chuyển chúng một cách độc lập, rõ ràng, ta phải cần một bộ nhớ khác có khả năng lưu dữ liệu khi không có điện và di chuyển được dễ dàng hơn. Bộ nhớ đó là bộ nhớ ngoài bao gồm đĩa mềm, đĩa cứng, CDROM và một số ổ đĩa khác.

I. Đĩa mềm và ổ đĩa mềm

Đĩa mềm được làm bằng nhựa, bên trong có lớp nhiễm từ bằng chất dẻo dùng để lưu trữ dữ liệu. Đĩa mềm có nhiều loại, có kích thước và dung lượng khác nhau.

Ví dụ: Thường có hai loại: 5.1/4 inch: 360 KB, 720 KB, 1.2 MB v.v...

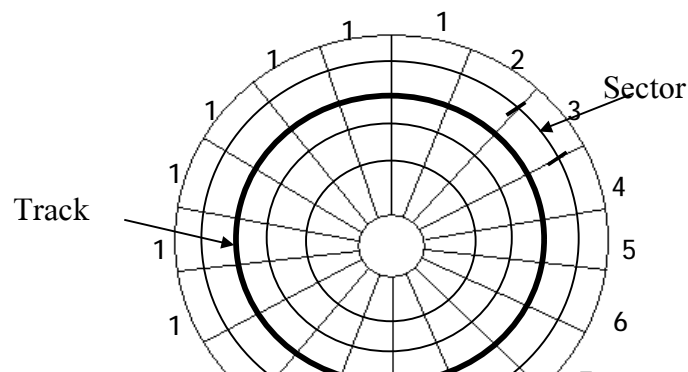
3.1/2 inch: 360KB, 720KB, 1.2MB, 1.44 MB v.v...

Hiện nay đa số dùng loại 3.1/2 inch 1.44MB.

Khi đĩa mềm làm việc, nó được đặt trong một ổ đĩa, ổ đĩa này có tác dụng làm quay đĩa và có một đầu từ sẽ làm nhiễm từ trên bề mặt đĩa ứng với các bit được ghi vào. Ổ đĩa này giao tiếp với mainboard qua một sợi cáp được cắm vào khe cắm Floppy trên mainboard.

1. Tổ chức vật lý của đĩa mềm

Để ổ đĩa mềm có thể ghi dữ liệu được và có thể đọc được dữ liệu ra từ đĩa mềm, người ta phải định dạng và đánh địa chỉ vào từng đơn vị của đĩa mềm. Phần này do hệ điều hành đảm nhiệm. Để có thể đánh địa chỉ cho đĩa người ta chia cấu trúc vật lý của đĩa mềm như sau:



Hình 6.1. Mô tả cấu trúc đĩa mềm loại 3.1/2 inch (1.44MB)

Mô tả đĩa mềm loại 3.1/2 inch (1.44MB):

Track: Track là các đường tròn đồng tâm, được chia làm 80 đường được đánh số từ ngoài vào trong, track ngoài cùng mang số 0.

Sector: Là đơn vị quản lý nhỏ nhất trên đĩa 1 Sector = 512 bytes cho mọi loại đĩa. Một track được chia làm nhiều Sector.

Cluster: Đơn vị này được tính theo Sector, 1 Cluster = 1,2,4,8,16,... Sector. Đây là đơn vị truy xuất thông tin trên đĩa, thường thì đĩa mềm 1Cluster = 1sector.

Head: Vì đĩa mềm có khả năng làm việc trên 2 mặt nên nó có 2 đầu ứng với 2 mặt đó. Để phân biệt hai đầu từ trên người ta đưa ra các giá trị 0 và 1 ứng với 2 đầu từ.

Như vậy, ta đã thấy tất cả các điểm trên đĩa mềm đều có thể chỉ định bằng tổ hợp các giá trị (Head, Track, Sector) hay (Head, Track, Cluster). Tuy vậy, trên đây chỉ là cách tổ chức vật lý, rất khó cho quá trình làm việc của máy tính. Do đó, người ta đưa ra cách đánh địa chỉ cho các đơn vị lưu trữ trên đĩa theo logic chính xác và đơn giản hơn.

2. Tổ chức logic của đĩa mềm

Để quản lý các thông tin trên đĩa mềm người ta tổ chức cấu trúc logic của nó như sau:

Boot Sector	Hidden	FAT1	FAT2	Root Directory	004	005
-------------	--------	------	------	----------------	-----	-----

Boot Sector: Là Sector đầu tiên của đĩa mềm chứa các thông tin về đĩa mềm đó và các đoạn chương trình điều khiển khởi động (Bootstrap) nếu đó là đĩa khởi động. Khi truy xuất máy tính đọc các thông số này để xác định cấu trúc của đĩa, từ đó xác định vị trí thông tin truy xuất. Nếu là đĩa khởi động thì chương trình khởi động sẽ được thực hiện để tìm ra các file khởi động trên đĩa.

Bảng FAT (File Allocation Table)

Bảng FAT là công cụ lưu giữ các thông tin liên quan đến Cluster trên đĩa. Mỗi điểm vào của bảng FAT là ứng với một giá trị của tổ hợp (Head, Track, Cluster). Toàn bộ bảng FAT là ánh xạ của toàn bộ các đơn vị trên đĩa. Các điểm vào này được đánh địa chỉ tuần tự. Các điểm vào trên FAT của các Cluster trống trên đĩa thì mang giá trị 000, các Cluster đã ghi dữ liệu sẽ được đánh dấu bằng giá trị của Cluster tiếp theo trong chuỗi, nếu là Cluster cuối cùng sẽ mang giá trị fff.

Khi ghi một file, HĐH sẽ lần trên bảng FAT tìm Cluster trống và ghi dữ liệu vào Cluster đó trên đĩa, đồng thời gán giá trị cho Cluster đó chỉ đến địa chỉ của các Cluster tiếp theo hoặc mang giá trị kết thúc fff. Trong giai đoạn này, HĐH cũng ghi địa chỉ của Cluster đầu tiên của chuỗi vào bảng thư mục.

Khi đọc dữ liệu từ một file, HĐH sẽ tìm đến Cluster đầu tiên để đọc dữ liệu. Sau đó, đọc dần đến các Cluster tiếp theo cho đến Cluster mang giá trị fff. Sau đây là đoạn trích của một đoạn FAT:

003	004	005	006	007	008	009	
004	005	009	5f7	000	000	fff	

Các Cluster 003, 004, 005, 009: cùng một file và Cluster 009 là Cluster kết thúc. Cluster 006 thuộc một file khác, Cluster 007, 008 là các Cluster còn trống.

Như vậy, bảng FAT rất quan trọng đối với đĩa, nó quyết định việc truy xuất thông tin trên đĩa và được sử dụng thường xuyên nên rất dễ bị lỗi. Do đó, trên đĩa người ta phải lưu trữ dự phòng bảng FAT thành 2 bảng là FAT1 và FAT2. Trong đó FAT1 được sử dụng và FAT2 dự trữ, khi FAT1 bị lỗi ta có thể thay FAT1 bằng FAT2 để tiếp tục làm việc.

Root Directory: Là bảng chứa thông tin về thư mục, mỗi điểm vào của bảng là những thông tin về các File hay Thư mục chứa trong thư mục gốc của đĩa. Những thông tin này giúp cho việc đọc dữ liệu trên đĩa. Cấu trúc của bảng có thể được mô tả như sau:

Tên file hay thư mục	Cluster đầu	Ngày thành lập	Ngày cập nhật gần nhất	Độ lớn tập tin, thư mục	. . .
.					
.					
Tên file hay thư mục	Cluster đầu	Ngày thành lập	Ngày cập nhật gần nhất	Độ lớn tập tin, thư mục	. . .

- Nếu là File thì mục Cluster đầu sẽ lưu giữ Cluster đầu tiên của File, được ghi vào khi File được ghi và được sử dụng khi truy cập File. Nếu là thư mục con thì trở đến địa chỉ đầu của bảng thư mục con có cấu trúc như bảng thư mục gốc trên.

- Khi truy cập thư mục thì thông tin xuất hiện trên màn hình chính là thông tin chứa trong bảng thư mục, do đó ta thấy dường như những thông tin này xuất hiện tức thời.

Để đĩa mềm có thể sử dụng được ta phải định dạng nó. Trong DOS, lệnh này là *Format a:*. Nếu muốn đĩa này thành đĩa khởi động ta thêm thông số /s vào lệnh Format như sau *Format a: /s* để HDH copy các file hệ thống vào đĩa giúp nó trở thành đĩa khởi động.

3. Ổ đĩa mềm

Có thể xem ổ đĩa mềm gồm một motor quay để quay tròn đĩa, motor bước và các đầu từ được dịch chuyển qua lại nhằm xác định vị trí cần truy cập trên đĩa. Một bảng mạch điều khiển để điều khiển hoạt động của các motor, các thành phần dẫn dữ liệu và một số thành phần phụ trợ khác.

Ngoài ra, để ổ đĩa mềm làm việc được với nhiều loại đĩa có kích thước khác nhau cần phải được điều khiển bởi phần mềm gọi là trình điều khiển ổ đĩa mềm, phần này thường được chứa trong chương trình BIOS. Do đó, khi có ổ đĩa mềm mới mà BIOS cũ (phần mềm điều khiển không phù hợp với ổ đĩa) thì có thể gây ra lỗi hoặc không sử dụng được nên phải nâng cấp BIOS cho phù hợp.

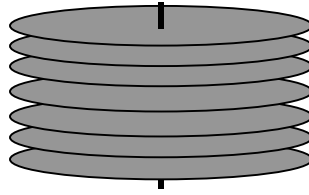
Hiện nay, đa số các BIOS đều hỗ trợ 2 ổ đĩa mềm có tên là A và B. Khi cài đặt, ta phải khai báo trong CMOS thì mới sử dụng được.

II. Ổ đĩa cứng

Đĩa cứng cũng là một loại đĩa từ có cấu trúc và cách làm việc giống như đĩa mềm, nhưng nó gồm nhiều lá đồng trục xếp lại và được đặt trong một vỏ kim loại kết hợp với bộ điều khiển thành ổ đĩa cứng. Do mỗi lá đã có dung lượng lớn hơn đĩa mềm và gồm nhiều lá nên ổ cứng có dung lượng rất lớn và có tốc độ truy cập rất cao. Hiện nay có rất nhiều loại đĩa cứng có tốc độ cao và dung lượng hàng GB như Seagate, Quantum v.v...

1 Cách tổ chức vật lý của đĩa cứng

Như đã giới thiệu, đĩa cứng rất giống đĩa mềm. Do đó, về cấu tạo và tổ chức của nó cũng giống nhau gồm Head, Track, Sector, Cluster, FAT. Tuy nhiên, chúng cũng có thêm một số khác biệt như sau:



Do có cấu trúc nhiều lá nên số đầu từ của ổ đĩa cứng cũng nhiều hơn so với ổ đĩa mềm và được đánh số từ 0 cho lớp trên cùng và cứ thế tăng dần xuống dưới. Cũng vì lý do như trên mà trong ổ đĩa cứng còn có khái niệm Cylinder là hình trụ, tập hợp các Track có cùng chỉ số.

2. Cách tổ chức logic ổ đĩa cứng

Do dung lượng đĩa cứng lớn nên để nguyên ổ đĩa như vậy sẽ gây khó khăn cho việc tổ chức cũng như tìm kiếm thông tin trên đĩa. Để khắc phục tình trạng trên, người ta cho phép chia ổ đĩa cứng thành nhiều phần có kích thước nhỏ hơn. Mỗi phần này hoạt động tương tự như một ổ đĩa cứng riêng biệt gọi là Partition. Để quản lý các Partition này, người ta dùng bảng Master Boot Record để lưu giữ các thông tin này, toàn bộ cấu trúc logic của đĩa cứng như sau:

Master Boot Record					
Boot Sector	Hidden	FAT1	FAT2	Root Directory	
.					
.					
Boot Sector	Hidden	FAT1	FAT2	Root Directory	

a. Master Boot Record

Master Boot Record là Sector đầu tiên của ổ đĩa cứng, nó chứa các thông tin về các Partition như số thứ tự, tên ổ đĩa logic, trạng thái, kích thước của Partition v.v... gọi là các điểm vào. Mỗi Master Boot Record có thể quản lý 4 điểm vào, mỗi điểm vào có kích thước 16 bytes, như vậy cần 64 bytes để lưu giữ các điểm vào này gọi là bảng Partition. Không gian còn lại của Sector này được lưu trữ chương trình Bootstrap của đĩa khởi động.

Như trên, ta thấy mỗi Master Boot Record chỉ chứa 4 điểm vào, như vậy mỗi đĩa cứng chỉ phân tối đa thành 4 phần. Để khắc phục điều này, người ta lấy

Sector đầu tiên của Partition thứ 4 để quản lý các phần chia tiếp theo như là một Master Boot Record thực thụ gọi là Master Boot Record phụ, cứ như thế mà ta có thể chia đĩa cứng thành nhiều phần khác nhau.

Master Boot Record được tạo ra bởi chương trình Fdisk của DOS, do đó, ta có thể khôi phục lại nó bằng lệnh này khi nó bị hỏng với tham số *mbr*, tức là lệnh Fdisk /mbr.

b. Partition (Phân khu):

Là phần được chia bởi ổ đĩa cứng, nó làm việc như một ổ đĩa biệt lập và có cấu trúc giống hệt như ổ đĩa mềm. Thông tin về Partition được lưu giữ trong bảng Partition trên Master Boot Record.

Đối với các hệ điều hành DOS và Windows chỉ cho phép khởi động ở Partition đầu tiên. Ngoài ra, còn có một số hệ điều hành cho phép khởi động từ các Partition khác.

Để phân đĩa cứng thành các Partition, ta dùng lệnh Fdisk của DOS, theo dõi các trình đơn của tiện ích này để chia đĩa cứng và tạo Partition khởi động.

c. Bảng FAT:

Về cơ bản, bảng FAT ở cứng giống hệt như việc tổ chức trên đĩa mềm, song chúng chỉ khác nhau về kích thước.

Đối với đĩa mềm, do kích thước đĩa hạn chế nên chỉ cần dùng 12 bits để đánh địa chỉ là đủ, thường được gọi là FAT 12. (12 bits đánh được 2^{12} địa chỉ điểm vào của FAT, nếu dùng 1 Cluster = 1 Sector ta sẽ đánh địa chỉ cho đĩa có dung lượng: $2^{12} \cdot 512 = 2^{21} = 2$ MB lớn hơn các loại đĩa mềm hiện nay).

Song đối với đĩa cứng có dung lượng lớn, nếu dùng FAT 12 để quản lý toàn bộ đĩa cứng ta phải tăng chỉ số Cluster lên rất nhiều gây lãng phí đĩa. (Ví dụ ở 500MB dùng FAT 12 thì lúc đó 1Cluster = 250 Sector ($1.024.000/4096$) = 125 KB. Song mỗi lần ghi dùng một Cluster nên nếu ghi một file có kích thước 100 byte cũng phải sử dụng 125 KB thật là lãng phí).

Để khắc phục tình trạng trên người ta đã đưa ra các bảng FAT16 (2^{16} điểm vào) và FAT32 (2^{32} điểm vào) để quản lý cho đĩa cứng. Với tốc độ tăng dung lượng của đĩa cứng như hiện nay trong tương lai chắc chắn sẽ có FAT 64 và hơn nữa. Tuy nhiên, với một ổ đĩa nhỏ mà ta dùng bảng FAT lớn sẽ gây lãng phí không gian chứa bảng FAT và ảnh hưởng đến tốc độ truy tìm.

d. Lắp ráp và khai báo sử dụng đĩa cứng:

Hiện nay, đa số đĩa cứng được thiết kế theo 2 chuẩn IDE (Integrated Drive Electronics) và SCSI (Small Computer System Interface). Song IDE được sử dụng rộng rãi hơn.

Các loại đĩa IDE giao tiếp với hệ thống thông qua Bus cắm vào hai khe cắm IDE1 và IDE2 trên Mainboard. Mỗi khe cắm cho dùng chung hai thiết bị làm việc theo chế độ khách chủ. Như vậy, trên toàn bộ máy tính sử dụng ổ đĩa IDE có thể sử dụng 4 ổ đĩa như sau:

- 1: Primary Master.
- 2: Primary Slave
- 3: Secondary Master.
- 4: Secondary Slave.

Để thiết lập chế độ Master, Slave cho ổ đĩa cứng ta cắm lại Jump thiết lập, thường được chỉ dẫn trực tiếp trên đĩa cứng hoặc Catalogue đi cùng. Tuy nhiên, một số loại đĩa cứng tự động nhận Master khi cắm cùng với các ổ đĩa khác.

Sau khi thiết lập xong phần cứng, ta phải khai báo sử dụng đĩa cứng trong mục Standard của CMOS và kiểm tra bằng mục Auto Detect Hard Disk để xem đĩa cứng có được nhận diện hay không.

Đối với loại đĩa giao diện SCSI thì cần phải có Card giao diện SCSI để điều khiển đĩa này. Card này được cắm vào bus PCI hay ISA của Mainboard. Các loại đĩa này cho phép sử dụng tối đa 7 thiết bị và không qua kiểm tra của CMOS.

3. Định dạng ổ đĩa cứng:

Để ổ đĩa cứng có thể làm việc được ta cần phải định dạng nó để tạo ra cấu trúc logic. Toàn bộ quá trình định dạng có thể chia thành các bước như sau:

* Định dạng cấp thấp: Đây là phương án định dạng về các mặt vật lý cho ổ đĩa cứng như Track, Cluster, Cylinder, hệ số đan xen. Chương trình này kiểm tra đến từng Sector của đĩa cứng và đánh dấu bỏ qua các Sector hỏng và đưa các giá trị thông tin về cùng một dạng 0,1. Do đó, đây cũng là chương trình cần để loại tận gốc dữ liệu trên đĩa cứng cũng như sửa các lỗi *Bad Sector* của đĩa cứng. Các Mainboard hiện nay đa số có hỗ trợ chương trình này trong BIOS qua mục *Hard disk Level Low Format*.

* Phân chia đĩa: Phân chia đĩa cứng thành nhiều thành phần (*Partition*) để tạo các ổ đĩa logic như đã trình bày ở trên. Chức năng này do chương trình Fdisk của hệ điều hành đảm nhiệm, chương trình tạo ra các Partition, xác định Partition cho phép khởi động và tạo ra Master Boot Record chứa bảng các thông số về

Partition. Ngoài ra, chương trình cũng cho phép xem, sửa và xóa các Partition đã có.

* Định dạng cấp cao: Đây là phần xác định các thông số logic, cấu hình các Partition đã được chia để nó làm việc như một ổ đĩa thực thụ. Phần này do chương trình Format của hệ điều hành đảm nhiệm, nhằm tạo ra Boot Sector, FAT, Root Directory v.v...

Khi muốn tạo ra đĩa khởi động ta dùng lệnh sau đối với các Partition đã được thiết kế khởi động trong phần phân đĩa ở trên:

Format Tên ổ đĩa logic : / s.

Đối với các Partition không cần khởi động ta dùng lệnh sau để tạo một ổ đĩa lưu dữ liệu bình thường:

Format Tên ổ đĩa logic.

Kết thúc các quá trình này ta đã kết thúc quá trình định dạng đĩa cứng và có thể sử dụng bình thường.

III. Quá trình khởi động máy tính trong DOS

Đây là quá trình diễn biến từ khi bật máy cho đến khi xuất hiện dấu nhắc của hệ điều hành. Quá trình này rất quan trọng trong việc xác định sự cố của máy vi tính. Toàn bộ quá trình được tả như sau:

- Khi bật máy nếu nguồn điện hoạt động tốt thì chương trình POST (Power On Self Test) trong ROM BIOS sẽ chạy tự động để kiểm tra các thiết bị cơ bản như CPU, RAM, Mainboard, Card màn hình. Nếu các thiết bị trên hoạt động tốt thì sẽ nhận được một tín hiệu Bíp của RAM và hệ thống tiếp tục làm việc. Nếu có thiết bị lỗi sẽ có một dòng tiếng Bíp thoát ra hoặc hệ thống không thực hiện gì cả.
- Chương trình POST tiếp tục kiểm tra các thiết bị được cắm vào máy tính và khai báo sử dụng trong CMOS, nếu có thiết bị nào đó bị lỗi thì xuất hiện thông báo lỗi, còn không thì chuyển qua bước sau.
- Kiểm tra các thông số cấu hình trong CMOS để hỗ trợ các thông số Plug and Play của BIOS cho thiết bị, đồng thời cũng đưa luôn bảng Vector ngắt lên vùng nhớ cơ bản và xác định thiết bị khởi động để chuyển điều khiển đến.
- Thực hiện chương trình Bootstrap trên Boot Sector của đĩa khởi động để tìm ra hai file khởi động chính là IO.SYS và MSDOS.SYS để chuyển điều khiển đến đó. Hai tệp IO.SYS và MSDOS.SYS được định vị tại hai vị trí đã xác định trước nên ta không có quyền thay đổi nó. Nếu nó

bị thay đổi hoặc bị lỗi thì đĩa đó không khởi động được nữa và thông báo lỗi *Non System Disk*.

- Kiểm tra *Command.com* để thực hiện, nếu file này bị lỗi sẽ thông báo *Command.com not Interpret*. Nếu bình thường thì hệ thống sẽ đọc *Command.com* lên RAM rồi thực hiện các lệnh trong *Config.sys* và *Autoexec.bat*, cuối cùng là dấu nhắc của hệ điều hành.

Dựa vào các thông báo lỗi và các diễn biến xảy ra để xác định các thành phần gây lỗi và tìm cách khắc phục.

IV. CDROM (Compact Disk Read Only Memory)

Khác với 2 loại đĩa trước hoạt động bằng phương thức nhiễm từ, CDROM hoạt động bằng phương pháp quang học. Nó được chế tạo bằng vật liệu cứng có tráng chất phản quang trên bề mặt.

Khi ghi đĩa CD, người ta sử dụng tia laser để chiếu lên bề mặt của đĩa tạo ra vùng dữ liệu ứng với các giá trị của bit 0 và 1. Do đó, đĩa CDROM chỉ ghi được 1 lần. Khi đọc ổ đĩa CDROM chiếu tia sáng xuống bề mặt phản quang và thu tia phản xạ, căn cứ vào cường độ tia phản xạ người ta suy ra đó là bit 0 hay bit 1.

Cách tổ chức về cấu trúc vật lý và logic của đĩa CDROM tương tự như trên đĩa mềm nên ở đây ta không nhắc lại.

CDROM có dung lượng rất lớn (khoảng 650MB), có thể di chuyển dễ dàng và giá tương đối rẻ, rất thuận tiện cho việc lưu giữ các chương trình nguồn có kích thước lớn, phim ảnh, v.v... nên hiện nay nó được dùng rất rộng rãi.

Để có thể đọc được ổ CDROM cần có một ổ đĩa CDROM được cài đặt đúng vào máy tính. Ổ đĩa CDROM hiện nay có rất nhiều loại có tốc độ khác nhau như 4x, 8x, 16x, 24x, 32x, 64x v.v...(1x=150 kbyte/s). Ổ CDROM hiện nay được thiết kế theo tiêu chuẩn SCSI nhưng nó có bảng mạch chuyển theo chuẩn IDE nên thường được cắm vào khe cắm IDE trên Mainboard hoặc gắn đi kèm với đĩa cứng.

* Cài đặt trình điều khiển cho ổ đĩa CDROM

- Trong Windows 95 về sau, chế độ Plus and Play tự động nhận và thiết lập trình điều khiển cho CDROM do đó chúng ta không cần cài đặt. Song chúng ta cũng có thể cài đặt lại chúng.

- Trong DOS, ta nhất thiết phải cài trình điều khiển cho ổ CDROM thì nó mới làm việc được. Cho chạy file Setup.exe trong đĩa cài đặt ổ đĩa CDROM để cài đặt trình điều khiển cho nó. Khi chạy chương trình này sẽ tạo ra thư mục

CDPRO chứa các file cấu hình và các file thông tin về ổ đĩa, chép file mscdex.exe vào đĩa khởi động để hỗ trợ trình điều khiển mở rộng đồng thời sửa lại hai file Config.sys và Autoexec.bat. Khởi động lại máy tính để cập nhật trình điều khiển ổ đĩa, nếu không có lỗi thì ta đã cài thành công ổ đĩa CDROM và có thể dùng nó như một ổ đĩa bình thường chỉ đọc.

Để ghi CDROM, người ta dùng một ổ ghi riêng và giá cả ổ đĩa này tương đối cao nên ít được dùng rộng rãi. Một số CDROM hiện nay được ghi thêm các file hệ thống cho phép khởi động từ ổ đĩa CDROM rất thuận tiện cho việc bảo quản hệ thống "sạch".

Một chức năng thường được sử dụng nữa của CDROM là nghe nhạc. Nếu kết hợp với Card âm thanh và loa thì có thể nghe nhạc, xem hình từ đĩa Compact Disk thông qua một số chương trình điều khiển như CD Player trong Windowsv.v.....

1 Để có thể sử dụng được các phần mềm có âm thanh, ta cần phải có Sound Card (Card âm thanh). Sound Card có tác dụng chuyển tín hiệu số (digital signal) thành tín hiệu tương tự (*analog*) phát ra loa. Để Card âm thanh làm việc được, ta cần phải cài đặt trình điều khiển cho nó.

§.7. CÁC THIẾT BỊ NGOẠI VI THÔNG DỤNG

I. Màn hình (Monitor)

Màn hình là thiết bị đưa thông tin của máy tính ra ngoài để giao tiếp với người sử dụng. Nó là bộ xuất chuẩn cho máy tính hay còn gọi là bộ trực. Hiện nay, có nhiều hãng sản xuất màn hình như Acer, IBM, Funal, Samsung, LG, Hitashi v.v... Nếu phân loại theo tính năng, màn hình bao gồm: Mono, EGA, VGA, SVGA v.v... Màn hình giao tiếp với Mainboard qua một bộ điều hợp gọi là card màn hình được cắm qua khe PCI, ISA hoặc EISA trên.

Ba vấn đề cần quan tâm trên màn hình là con trỏ màn hình, độ phân giải và màu sắc. Con trỏ màn hình chỉ định vị trí dữ liệu sẽ xuất ra trên màn hình, độ phân giải đặc trưng cho độ mịn màn hình.

+ Con trỏ: Là nơi để máy tính đưa thông tin tiếp theo ra từ đó, nó được đặt trưng bởi cặp tọa độ (x, y: Chỉ xét cho độ phân giải chứ không xét theo chế độ văn bản hay chế độ đồ họa) trên màn hình.

+ Độ phân giải: Màn hình được chia thành nhiều điểm ảnh, số điểm ảnh được tính bằng tích số dòng dọc và dòng ngang chia trên màn hình. Cặp giá trị ngang, dọc gọi là độ phân giải của màn hình như (480 x 640), (600 x 800), (1024 x 768) v.v...

+ Màu sắc: màu của các đối tượng trên màn hình do màu các điểm ảnh tạo nên.

Card màn hình điều khiển đến từng điểm ảnh trên màn hình. Nội dung của các điểm ảnh (vị trí và màu sắc) trên màn hình được lưu giữ trên một chip nhớ (RAM Card) và cứ sau một khoảng thời gian nhỏ hơn 1/24 giây nó sẽ quét toàn bộ màn hình một lần làm cho chúng ta thấy hình ảnh hiển thị liên tục trên màn hình. Khi chip này có dung lượng lớn, nó có khả năng lưu giữ số điểm ảnh nhiều hơn và màu cho mỗi điểm ảnh cũng đa dạng hơn tạo cho màn hình có độ mịn và độ nét cao. Tuy nhiên, nếu màn hình hoạt động chậm mà độ phân giải cao và chế độ màu lớn sẽ không đảm bảo mức độ chu kỳ 1/24 giây dẫn đến màn hình nhấp nháy.

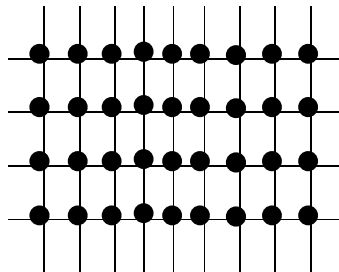
Hệ điều hành DOS và Windows đều hỗ trợ tính năng Plus and Play (cắm sử dụng) cho màn hình. Tuy nhiên, trong các chế độ đồ họa cao cấp yêu cầu phải có trình điều khiển đúng cho màn hình thì mới đạt được hiệu quả cao. Card màn hình có loại được Windows tự động cài trình điều khiển, có loại ta phải tự cài lấy. Để cài lại trình điều khiển cho Card màn hình ta thực hiện như sau:

1. Khởi động Windows, vào Control Panel/ System/ Device manager
2. Remove điều khiển card màn hình có sẵn.
3. Chọn Add New Hardware rồi chỉ đường dẫn đến trình điều khiển card màn hình.
4. Chọn Apply để áp dụng trình điều khiển mới và khởi động lại máy tính.

II. Bàn phím (Keyboard)

Bàn phím là một thiết bị đưa thông tin vào trực tiếp giao diện với người sử dụng. Nó được nối kết với Mainboard thông qua cổng bàn phím (đặc trưng bởi vùng nhớ I/O và ngắt bàn phím).

Bàn phím được tổ chức như một mạng mạch đan xen nhau mà mỗi nút mạng là một phím. Khi ấn một phím sẽ làm chập mạch điện tạo ra xung điện tương ứng với phím được ấn gọi là mã quét (Scan Code). Mã này được đưa vào bộ xử lý bàn phím (8048,8042) diễn dịch ra ký tự theo một chuẩn nào đó, thông thường là chuẩn ASCII (American Standard Code for Information Interchange). Sau đó, bộ xử lý ngắt bàn phím yêu cầu ngắt và gửi vào CPU xử lý. Vì thời gian thực hiện rất nhanh nên ta thấy các phím được xử lý tức thời.



Hiện nay, trên thị trường có nhiều loại bàn phím do nhiều hãng sản xuất khác nhau như Acer, IBM, Turbo Plus, Mitsumi v.v... Tuy nhiên, chúng có chung một số các phím cơ bản từ 101 đến 105 phím được chia làm 2 nhóm:

- Nhóm ký tự: Là nhóm các phím khi gõ lên có ký tự xuất hiện trên màn hình.
- Nhóm điều khiển: khi gõ không thấy xuất hiện ký tự trên màn hình mà thường dùng để thực hiện một tác vụ nào đó.

Tất cả các phím đều được đặc trưng bởi một mã, một số tổ hợp phím cũng có mã riêng của nó. Điều này giúp cho việc điều khiển bàn phím rất thuận lợi, nhất là trong công việc lập trình.

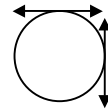
III. Chuột (Mouse)

Chuột là là thiết bị điều khiển trở trực tiếp phổ biến nhất, đặc biệt là trong lĩnh vực đồ họa. Hiện nay, có rất nhiều loại chuột do nhiều hãng sản xuất khác nhau như IBM, Acer, Mitsumi, Genius, Logitech v.v... đa số được thiết kế theo hai chuẩn cổng cắm tròn và đẹp. Tuy nhiên, chúng có cấu tạo và chức năng như nhau.

Về cấu trúc thì chuột có các loại như chuột cơ học, chuột quang học, chuột cơ quang v.v... Song chỉ có loại chuột cơ học là phổ biến còn các loại khác được dùng trong các lĩnh vực đặc biệt. Chuột cơ học học có hai bộ phận là bi di chuyển và các nút nhấn.

- Bi di chuyển: Gồm một viên bi và hai thanh quay ngang, dọc.

Khi di chuyển chuột tương ứng theo các chiều sẽ làm các thanh quay



tương ứng tạo ra xung điện di chuyển vị trí chuột tương ứng trên màn hình.

- Nút nhấn: Tạo ra xung chỉ thị sự thực hiện các lệnh điều khiển tại vị trí chuột trên màn hình. Nhấp chuột (Click) là động tác ấn phím trái của chuột, song lại thả nhanh ra ngay (thường dùng để chọn một cái gì đó trên màn hình). Nhấp kép (Double Click) có tác dụng cho chạy một chương trình. Nhấp phải (Right Click) thường sử dụng trong môi trường Windows 9x, 2000, NT... nhằm mở trình đơn phụ (pop up) của một đối tượng.

Đối với Windows 95 trở lên chuột được Plus and Play, còn đối với DOS chúng ta phải cài đặt trình điều khiển cho chuột (thường là file mouse.com, gmouse.com) thì nó mới có thể hoạt động được.

IV. Máy in (Printer)

Máy in là thiết bị chủ đạo để xuất dữ liệu máy tính lên giấy. Khi muốn in một file dữ liệu ra giấy thì CPU sẽ gửi toàn bộ dữ liệu ra hàng đợi (queue) máy in và máy in sẽ lần lượt in từ đầu cho đến hết file.

Máy in hiện nay có rất nhiều loại với nhiều cách thức làm việc khác nhau như máy in kim, máy in phun, máy in lazer v.v...Để đánh giá về chất lượng của máy in người ta căn cứ vào hai yếu tố của máy in là tốc độ (speed) và độ mịn.

- Tốc độ của máy in thường đo bằng trang/ giây (chỉ tương đối). Tốc độ này nhiều khi còn phụ thuộc vào tốc độ của máy tính và mật độ của trang in chứ không chỉ của máy in. Đối với máy in kim thì tốc độ rất hạn chế song đến máy in Laze thì tốc độ đã được cải thiện đi rất nhiều.

- Độ mịn (dots per inch): Độ mịn phụ thuộc vào nhiều yếu tố song yếu tố cơ bản phụ thuộc thông số dpi được ghi trực tiếp trên máy in.

Máy in giao tiếp với CPU thông qua các cổng song song LPT1, LPT2, LPT3, LPT4 được gắn qua khe cắm trên Mainboard.

Hầu hết các hệ điều hành đều hỗ trợ máy in. Đối với DOS thì ta phải cài đặt Driver của máy in cho hệ điều hành thì nó mới làm việc được. Song đối với các hệ điều hành từ Windows 95 trở lên chế độ Plug and Play hỗ trợ hầu hết các loại máy in hiện nay, do đó ta chỉ chọn cho đúng trình điều khiển mà thôi.

Để thiết lập máy in và in được một file ta làm như sau:

1. Cắm máy in vào máy tính và cắm điện cho máy in.
2. Bật máy tính và cài đặt máy in cho hệ điều hành đang sử dụng.
3. Bật điện máy in và cho giấy vào để chuẩn bị sẵn sàng.
4. Chọn file cần in và gõ lệnh in. Trong DOS là lệnh PRN tên file. Trong Windows mở file cần in. sau đó chọn File/Print.

V. Một số thiết bị khác

Ngoài ra, còn rất nhiều thiết bị được cắm vào máy tính để phục vụ cho nhiều mục đích khác nhau như Card mạng, Modem, Scanner, v.v... Sau đây, xin giới thiệu sơ lược về một số thiết bị:

1. Card mạng (Network Adapter): là vi mạch được nối vào máy thông qua Bus PCI hoặc ISA, đầu ra sử dụng các đầu nối để nối dây mạng. Card mạng dùng để thiết lập mạng cho mục đích giao tiếp giữa các máy tính với nhau. Để Card mạng hoạt động được, ta phải thiết lập đúng trình điều khiển của nó, địa chỉ của các máy tính trên mạng, và cài đúng giao thức (Protocol) để giao tiếp.

2. Modem: Là từ viết tắt của *Modulator - Demodulator* là thiết bị điều chế - giải điều chế. Modem là thiết bị truyền dữ liệu được dùng để nối các máy tính với nhau bằng đường dây viễn thông với cự ly bất kỳ trên thế giới như mạng Internet.

Tín hiệu xử lý trong máy tính hoặc tín hiệu bắt tay giữa hai máy tính là tín hiệu số (*Digital Signal*), trong khi đường truyền viễn thông chủ yếu phục vụ tín hiệu dạng tương tự (*Analog*). Tín hiệu truyền trên đường dây điện thoại là tín hiệu đã được điều chế biên độ *AM (Amplitude Modulation)*, vì vậy Modem có nhiệm vụ chuyển đổi tín hiệu số từ máy tính thành tín hiệu AM và gửi đi. Tại đầu nhận, Modem lại giải điều chế (*Demodulation*) tín hiệu AM lấy lại tín hiệu số

cung cấp cho máy tính. Nhờ có Modem mà hai máy tính ở khoảng cách xa có thể "nói chuyện" được với nhau.

Modem có hai loại: Loại lắp thẳng vào trong máy tính bằng một card riêng được gọi là Modem trong (*Internal Modem*), hoặc Modem ngoài (*External Modem*), Modem ngoài được nối thông qua cổng nối tiếp của máy tính như cổng COM1, COM2.

Khi nói đến Modem, người ta thường quan tâm đến tốc độ truyền. Đơn vị là Baud = bit/ giây (thường được ký hiệu là bps, KBps). Tốc độ thường từ 9600 bps đến 33600 bps. Hiện nay, đa số Modem có tốc độ là 56K bps.

3. Scanner: là thiết bị chuyên dùng để quét các hình ảnh và lưu vào máy tính dưới dạng tập tin ảnh.

VI. Truyền song song (Parallel), nối tiếp (Serial):

Trên đây, ta đã xét xong các thiết bị ngoại vi của máy tính cũng như quá trình giao tiếp của chúng với CPU. Nhưng ta chưa xét về cách truyền dữ liệu của chúng như thế nào. Tất cả các thiết bị giao tiếp với CPU đều sử dụng các dạng truyền song song hoặc nối tiếp hoặc vừa nối tiếp vừa song song.

- Truyền nối tiếp: Là hình thức dữ liệu được truyền và nhận theo dòng từng bit một. Loại này có nhược điểm là tốc độ truyền dữ liệu nhỏ, song việc kiểm soát dữ liệu được truyền thì đơn giản. Hình thức này thường được dùng trong các giao tiếp của bàn phím, chuột v.v...
- Truyền song song: Là hình thức dữ liệu được truyền theo nhiều đường cùng một lúc. Tại một thời điểm, có thể truyền được nhiều bit dữ liệu, do đó mà tốc độ tăng lên rất nhiều. Cũng chính điều này làm cho việc kiểm soát dữ liệu nơi nhận phức tạp hơn nhiều. Kiểu truyền này thường được sử dụng rộng rãi trong giao tiếp máy in.
- Hiện nay, hầu hết các thiết bị máy tính kết hợp cả hai kiểu truyền song song và nối tiếp để vừa đảm bảo tốc độ nhanh vừa đảm bảo kiểm soát dễ dàng. Khi dữ liệu song song vào đoạn nối tiếp sẽ bị ngẽn lại. Để giải quyết tình trạng này, người ta dùng bộ thu phát đa năng không đồng bộ UART (Universal Asynchronous Receiver / Transmitter) làm việc với tất cả các phần mềm truyền thông để giải quyết vấn đề trên.

PHẦN II

CÀI ĐẶT HỆ THỐNG MÁY TÍNH

§.8. RÁP MÁY

I. Các thành phần cần thiết

Sau đây là tất cả các thành phần cần thiết để chuẩn bị cho việc ráp máy:

- Hộp máy và bộ nguồn.
- Mainboard (Mainboard).
- CPU và quạt CPU.
- Ổ đĩa cứng.
- Ổ đĩa mềm.
- Ổ đĩa CDROM.
- Màn hình.
- Bộ điều hợp màn hình.
- Card âm thanh.
- Card MODEM.
- Bàn phím.
- Chuột.
- Cáp IDE.
- Cáp ổ đĩa mềm.
- Cáp audio ổ đĩa CDROM.
- Phần mềm: Ở đây ta sẽ sử dụng các hệ điều hành thông dụng của Microsoft tức là các phiên bản của Windows (*Windows 95, Windows 98, Windows 2000 hoặc Windows XP*) và những phần phần mềm cần thiết khác.
- Một đĩa mềm khởi động.

II. Dụng cụ

Trước khi bắt đầu, bạn nên tập hợp tất cả các chi tiết máy và chuẩn bị một tuốt - nơ - vít và một cái kềm mở dài. Kềm mở dài dùng để đặt cấu hình cho các cầu nối nhỏ. Nếu bạn không có kềm mở dài bạn có thể sử dụng nhíp.

III. Cẩn thận với dòng điện tĩnh

Trước khi chạm vào bất cứ linh kiện nào, bạn phải phóng tất cả các dòng điện tĩnh trong cơ thể bạn. Cơ thể người có thể chứa từ 300V dòng điện tĩnh trở lên. Nếu bạn chạm vào bất kỳ một bộ phận nhạy điện nào, dòng điện tĩnh sẽ được xả qua nó. Dòng điện tĩnh này sẽ phá huỷ hoặc gây hư hỏng nặng những thiết bị nhỏ.

Khi bạn chạm vào tay nắm cửa bằng kim loại, bạn đã có thể tự phóng dòng điện tĩnh đang tích lũy trong cơ thể bạn. Tốt hơn hết, bạn nên chạm vào những vật gì nó trực tiếp tiếp xúc với đất như ống nước hay bằng kim loại thuần của máy tính bạn. Hầu hết các board và các thiết bị đều có dán lời cảnh báo về dòng điện tĩnh trên các bao hình.

IV. Các bước thực hiện

Trước khi ráp máy bạn nên tập hợp chúng lại và để trên một cái bàn hay một khu vực nào dành riêng cho nó. Công việc tiếp theo là bật công tắc nguồn và thử nó trước khi ráp nó vào hộp máy để phòng khi có vấn đề gì xảy ra cũng dễ phát hiện hơn khi nó vẫn còn trong trạng thái mở. Phía sau mainboard và các board khác có phần nhô ra rất nhọn, vì vậy bạn nên đặt các board mạch lên trên nhiều lớp báo để tránh gây trầy xước cho mặt bàn.

Các bước lắp đặt chi tiết được liệt kê ở phần sau, tóm tắt quá trình như sau: Gắn cáp nguồn điện vào mainboard. Nếu bạn sử dụng nguồn điện kiểu cũ (nguồn AT) thì 4 dây cáp màu đen phải cắm ở giữa, nếu bạn sử dụng mainboard và bộ nguồn loại ATX thì các ổ cắm trên nó được thiết kế chỉ cho phép bạn gắn bộ nối cáp theo một cách duy nhất. Kế đến, bạn nối dây ổ đĩa mềm, ổ đĩa cứng, bàn phím, Card màn hình và màn hình. Xong bạn bật nguồn điện, cho máy khởi động thử xem nó có hoạt động được không.

1. Xác định sơ bộ một số cấu hình

Các CPU: AMD K6, Cyrix 6x86MX, IDT Centaur C6 và Intel Pentium MMX được thiết để sử dụng trên mainboard Socket 7.

Tất cả ổ cắm CPU cho các loại CPU Socket 7 là một ổ cắm ZIF (Zero Insertion Force). Bên dưới ổ cắm là một đòn bẩy khi bạn nâng nó lên, nó sẽ mở

tất cả các chỗ tiếp xúc để bạn dễ dàng gắn CPU vào. Khi bạn hạ nó xuống, các chân của CPU được kẹp chặt bên trong ổ cắm này.

CPU Intel Pentium II và III được lắp trên một board nhỏ, Intel gọi board này là bộ nối cạnh đơn SEC (Single Edge Contact). board SEC được cắm vào mainboard nhờ vào bộ nối Slot 1.

2. Cấu hình cho mainboard

Nếu mua chung mainboard và CPU, các cầu nối trên mainboard đã được cài và cấu hình sẵn CPU rồi nhưng bạn cũng nên đọc tài liệu hướng dẫn để kiểm tra lại cho chắc chắn. Nếu bạn mua mainboard và CPU rồi bạn phải sử dụng đến tài liệu hướng dẫn đi kèm để cài các cầu nối (jump) CPU cho đúng bởi vì trên một mainboard cho phép bạn sử dụng rất nhiều loại CPU khác nhau. Cầu nối trên mainboard rất nhỏ, vì vậy bạn cần phải sử dụng đến kèm mỏ dùi hoặc nhíp để cài đặt chúng. Cầu nối nhỏ thường được dùng để cấu hình cho điện áp sử dụng trên CPU, tần số, tốc độ Bus, loại bộ nhớ, và nhiều chức năng khác nữa. Bạn cần cẩn thận khi cài đặt các cầu nối này. Ví dụ, các mainboard đều cho phép sử dụng nhiều loại CPU khác nhau nên nếu bạn cài đặt mức điện áp cho CPU không đúng có thể dẫn tới cháy CPU.

3. Lắp CPU vào mainboard Socket 7

Để gắn CPU vào mainboard Socket 7 ta chỉ việc nhấc đòn bẩy ZIF lên và đặt CPU xuống. Bạn nên chú ý là ở một góc của CPU có dấu chấm và góc này bị cắt để cho biết đó là chân số 1. Bạn tìm chân số 1 và đặt CPU khớp vào socket, ta phải rất cẩn thận bởi các chân của CPU rất yếu. Khi đã đặt CPU vào, kéo đòn bẩy xuống và gắn quạt lên trên CPU và nối nguồn điện cho quạt. Nguồn điện cho quạt CPU tùy theo đầu nối điện là loại cắm thẳng lên mainboard hoặc nối trực tiếp vào nguồn điện mà ta cắm tương ứng.

Nếu CPU là loại Pentium III, trước tiên bạn phải lắp thêm một cái vòng kẹp lên mainboard để nó giữ CPU. CPU được gắn vào bộ nối Slot 1, xung quanh là 4 vít để giữ cho chặt.

4. Lắp bộ nhớ

Bước kế tiếp là lắp các chip bộ nhớ . Một khi đã gắn mainboard vào bạn sẽ rất khó đụng tới các khe cắm bộ nhớ, vì vậy tốt hơn hết bạn nên gắn các chip bộ nhớ vào mainboard trước khi lắp mainboard vào hộp máy.

Các khe để cắm chip bộ nhớ không được dán nhãn một cách rõ ràng. Vì vậy, bạn nên sử dụng tài liệu hướng dẫn đi kèm với mainboard để xác định xem cần gắn vào khe nào trước. Thông thường bạn phải gắn vào dải được đánh số nhỏ nhất, số 0 (hoặc 1) trước. Bộ nhớ rất dễ gắn vì nó được thiết kế sao cho bạn chỉ có một cách duy nhất để gắn. Đối với các module nhớ một hàng chân (SIMM) bạn chỉ việc đặt chúng hơi nghiêng một chút vào các khe và kéo chúng về phía bạn cho tới khi vòng kẹp bên ngoài kẹp chặt chúng .

Đối với Module nhớ có 2 hàng chân (DIMM), việc lắp đặt nó hơi khó hơn SIMM một chút. Thay vì đặt nghiêng và kéo từ từ như SIMM, với DIMM bạn ấn thẳng từ trên xuống cho tới khi chúng được khoá chặt lại

5. Lắp đặt ổ đĩa

Đối với một vài ổ đĩa cứng, thật khó xác định mặt nào là mặt trên. Thông thường mặt trên thường được bịt kín, mặt dưới thường có các thiết bị linh kiện điện tử được bóc trần. Trước khi lắp ổ đĩa, bạn phải cấu hình cho chúng. Nếu bạn chỉ lắp một ổ đĩa IDE, các hãng sản xuất đã cài sẵn cho bạn nó là ổ đĩa 1 hoặc ổ đĩa chính (Master). Bạn dùng đoạn cáp ruy băng 40 dây đi kèm, thường có 3 đầu nối, một ở đầu cuối cùng dùng để gắn vào cổng trên mainboard được đánh dấu là Primary. Bạn nối ổ đĩa cứng với một trong hai bộ nối còn lại.

Nếu bạn lắp hai ổ đĩa cứng IDE, thì ổ đĩa C sẽ là ổ khởi động (là ổ đĩa chính), ổ đĩa còn lại sẽ là ổ đĩa phụ. Thông thường, khi sản xuất, các hãng sản xuất đều cài sẵn ổ đĩa cứng nào cũng là ổ đĩa chính. Nếu ta không cấu hình lại cho đúng là ổ đĩa phụ, ta sẽ không truy xuất vào các ổ đĩa được.

Trên mainboard thường có 2 hàng chân để gắn các ổ đĩa IDE, được đánh dấu là "Primary" (hoặc IDE 0, IDE1) và "Secondary" (hoặc IDE 1, IDE2). Nếu ta lắp một ổ đĩa cứng thì gắn chúng trên hàng chân có đánh dấu là Primary. Bạn phải xác định mặt có màu của cáp để gắn cho đúng chân số 1. Nếu bạn lắp nhiều hơn 2 ổ đĩa IDE, bạn phải lắp chúng trên hàng chân phụ thứ hai (có dấu là Secondary) . Thông thường, trên hàng chân được gọi là Primary sẽ cho phép ta gắn 2 ổ đĩa: 1 ổ chính, và một ổ phụ; trên hàng chân được gọi là Secondary cũng cho phép bạn gắn 2 ổ đĩa như trên hàng chân Primary.

Đối với các ổ đĩa bạn nên sử dụng mỗi bên hai con vít giữ chúng nhưng bạn đừng nên siết chặt quá bởi vì các khung của ổ đĩa được làm bằng chất liệu nhôm, mềm, rất dễ bị tróc. Bạn cũng không nên sử dụng các con vít quá dài, nếu quá dài chúng sẽ chạm vào mạch điện trên ổ đĩa.

6. Lắp các board Plus-in

Sau khi gắn tất cả các cáp, bước kế tiếp là gắn các board plus-in. Giả bạn có một board điều hợp màn hình, một board âm thanh, một board FAX/MODEM. Tất cả các board này đều gắn lên trên mainboard.

Tiếp theo là bạn gắn các đèn LED ở mặt trước hộp máy để báo hiệu trạng thái và hoạt động của máy và ổ cứng. Trên đó cũng có các hàng chân để gắn loa, gắn công tắc khởi động lại.

Cuối cùng bạn gắn bàn phím và chuột.

7. Bật nguồn và khởi động máy

Trước khi gắn nắp hộp máy lại, ta nên kiểm tra lại xem tất cả cáp đã được gắn đúng hay chưa để khi gắn vào rồi bạn không phải mất thời gian kiểm tra lại.

Sau khi kiểm tra lại đã đúng toàn bộ, bật nguồn và khởi động máy. Ta phải chuẩn bị một đĩa mềm khởi động 3 $\frac{1}{2}$ inch, gồm có cả các chương trình FDISK và FORMAT để thực hiện việc định dạng cho ổ đĩa cứng.

Lưu ý: Đối với các máy tính chưa được cài đặt thông tin cấu hình hệ thống (*BIOS Setup*) lần nào thì phải được thực hiện trước tiên sau khi bật máy lần đầu. Bước này ta sẽ nghiên cứu kỹ ở bài sau.

8. Định dạng ổ đĩa cứng

Sau khi ráp xong, cần phải định dạng và nạp các phần mềm vào ổ đĩa cứng. Bạn không thể sử dụng ổ đĩa cứng cho tới khi chúng được định dạng hoàn chỉnh.

Các phần mềm Windows 95/ 98 thường được ghi trên một ổ đĩa CDROM 650MB, trên đó bao gồm hàng trăm chủ đề và tập tin giúp đỡ bạn. Bạn sẽ không thể tìm thấy bất cứ một giúp đỡ nào về việc định dạng đĩa cứng mà chỉ có tập tin giúp đỡ bạn định dạng đĩa mềm. Thật vậy, cũng có rất ít sách nói chi tiết về cách thức định dạng một ổ đĩa cứng.

a. Mục đích của việc định dạng

Công việc định dạng sẽ giúp bạn tổ chức đĩa để dữ liệu có thể được lưu trữ và truy xuất một cách dễ dàng và nhanh chóng. Nếu dữ liệu không được tổ chức, sẽ rất khó tìm nó trên một đĩa cứng có kích thước lớn. Thông thường, các tập tin được ghi trên một đĩa cứng đều được lưu trên các rãnh ghi và cung từ đã được đánh số. Bảng định vị tập tin FAT sẽ thiết lập tập tin về vị trí của từng rãnh ghi, từng cung từ ở trên đĩa.

Cách thức tổ chức đĩa cứng giống như cách thức tổ chức một mảnh đất. Một mảnh đất được chia thành nhiều lô, trên mỗi lô sẽ xây một căn nhà mỗi nhà sẽ có một địa chỉ duy nhất tức là có tên đường và có số nhà. Rãnh ghi trên đĩa tương đương với tên đường và số cung từ tương đương với số nhà.

Bảng FAT tương tự như một bảng chỉ dẫn đường hay một trang mục lục ở bất kỳ một quyển sách nào. Khi có một yêu cầu được gửi đến đầu ghi để đọc hoặc ghi tập tin, nó sẽ đi tới bảng FAT, tìm vị trí định vị của tập tin đó và di chuyển đến nó. Đầu ghi có thể tìm bất cứ tập tin nào hoặc bất cứ phần nào của tập tin một cách nhanh chóng và dễ dàng.

Công việc định dạng không phải là công việc được thực hiện mỗi ngày, nó chỉ được thực hiện lúc mới "bóc hộp", phát hiện hỏng đĩa, chia lại đĩa v.v... Sở dĩ các hãng sản xuất không thực hiện việc định dạng trước cho ổ đĩa cứng vì nó có quá nhiều cách để chọn lựa. Ví dụ: nếu ta có một ổ đĩa cứng 20 GB, ta có thể chia nó thành 2 hoặc 3 ổ đĩa logic tùy ý bạn. Vì vậy, họ không thực hiện định dạng trước mà phải do bạn tự định dạng.

b. Các bước thực hiện

Bạn phải có một đĩa mềm khởi động chứa các tập tin Command.com, IO.sys, MSDOS.sys, Fdisk.com, Format.com, Sys.com, Config.sys, Autoexec.bat và Mscdex.exe. Cho đĩa khởi động này vào ổ đĩa mềm và bật nguồn. Tại đường dẫn A: bạn gõ fdisk <Enter>. Chương trình Fdisk cho phép bạn phân ổ đĩa thành một hoặc nhiều ổ đĩa logic. Ổ đĩa đầu tiên của bạn nên là ổ đĩa hệ thống DOS chính, tức là ổ đĩa C. Trong hệ thống của chỉ có thể có một ổ đĩa chính hoạt động, còn lại tất cả các ổ đĩa logic khác đều là các ổ đĩa DOS mở rộng.

c. Các tùy chọn khi thực hiện chương trình FDISK

FDisk có nghĩa là "Fixed Disk" hay "Format Disk". Nó là một tập tin lệnh ngoại trú của DOS trên đĩa khởi động. Bạn sẽ không thể sử dụng được đĩa cứng

cho tới khi nó được phân chia bằng lệnh FDisk, sau đó nó được định dạng ở mức cao (lệnh Format).

DOS sử dụng tất cả các ký tự trong bảng chữ cái để đặt tên cho ổ đĩa. Nó dành ký tự A cho ổ đĩa mềm, C cho ổ đĩa khởi động. Nếu có một đĩa dung lượng lớn, ta có thể chia thành 23 phần logic nhỏ hơn được đặt tên là từ ổ đĩa D đến ổ đĩa Z.

Khi bạn gõ lệnh FDisk, nếu bạn sử dụng MSDOS 6.22 hoặc các phiên bản về sau. Bạn sẽ nhận được thông báo đầu tiên khi chạy tập tin Fdisk như sau:

Fdisk Options:

Current Fixed Disk Drive: 1

Choose one of the following:

- 1. Create DOS Partition or Logical DOS Drive*
- 2. Set active Partition*
- 3. Delete Partition or Logical DOS Drive*
- 4. Display Partition Information*

Ý nghĩa các mục chọn trên như sau:

- 1: Tạo mới một vùng hoặc tạo ổ đĩa logic
- 2: Chọn lựa vùng hoạt động ban đầu
- 3: Xoá vùng đã phân chia hoặc xoá ổ đĩa logic
- 4: Hiện thị thông tin các vùng đã phân chia

Nếu chọn số 1 (*Nếu là lần đầu Fdisk đĩa*), các tùy chọn tiếp như sau:

Enter choice: [1]

Press ESC to exit FDISK

Create DOS Partition or Logical DOS Drive

Current Fixed Drive: 1

Choose one of the following:

- 1. Create Primary DOS Partition*
- 2. Create Extended DOS Partition*

3. Create logical DOS drive(s) in the Extended DOS Partition

Enter choice: [1]

Press ESC to return to FDISK Options

Nếu bạn muốn khởi động từ đĩa cứng, phải chọn số 1 để tạo một phần DOS chính và làm cho nó hoạt động. Nếu chọn 1, ta sẽ nhận được một câu nhắc nhở như sau:

Do you wish to use the maximum size for a primary DOS Partition and make the Partition active (Y/N) ? [Y]

Có nghĩa là: *Bạn có muốn sử dụng hết dung lượng đĩa cho phần DOS chính và cho phép phần này hoạt động không ?*

Nếu bạn gõ *Y* để trả lời đồng ý, toàn bộ ổ đĩa của sẽ được tạo thành một ổ đĩa C: duy nhất. Nếu bạn gõ *N*, nó sẽ xuất hiện xuất hiện dung lượng đĩa tối đa của bạn và hỏi bạn muốn dành bao nhiêu phần trăm cho ổ đĩa chính. Bạn có thể chọn 50% hoặc tùy theo cách chia của bạn. Bạn có thể sử dụng một ổ đĩa là một phần duy nhất, nhưng tốt hơn hết bạn nên phân nó thành 2 phần nhỏ trở lên.

Sau khi bạn tạo phần chính xong, ấn phím ESC, ta sẽ nhận được màn hình với các tùy chọn sau:

Create DOS Partition or Logical DOS Drive

Current Fixed Drive: 1

Choose one of the following:

1. Create Primary DOS Partition

2. Create Extended DOS Partition

3. Create logical DOS drive(s) in the Extended DOS Partition

Enter choice: [1]

Press ESC to return to FDISK Options

Phần chính đã được tạo nên phần này ta chọn 2 để tạo phần DOS mở rộng. Nó sẽ cho biết ổ đĩa của còn lại bao nhiêu sau khi bạn đã chỉ định cho phần chính. Nếu ta có ổ đĩa 20 GB và đã chỉ định 5 GB cho ổ đĩa chính, Fdisk sẽ báo cho biết còn 15 GB cho các phần còn lại.

Tại đây, bạn không thể thực hiện phân ổ đĩa của bạn được. Bạn chỉ có thể chấp nhận con số được đưa ra. Nếu bạn cố gắng phân ổ đĩa tại đây thì con số bạn chỉ định là bao nhiêu nó sẽ là tất cả những gì bạn có thể sử dụng được. Theo ví dụ trên: với tùy chọn 2, nếu ổ đĩa còn lại 15 GB và bạn gõ con số 7.5, chương trình hiểu là toàn bộ phần ổ đĩa mở rộng chỉ là 7.5 GB. Như vậy, bạn sẽ không thể sử dụng 7.5 GB còn lại. Tại đây bạn phải báo là sử dụng 15 GB hiện có.

Kế tiếp, ấn Esc và trở về bảng tùy chọn, sau đó chọn số 3. Tùy chọn này cho phép chia phần này thành bao nhiêu ổ đĩa tùy ý.

Tại đây, chương trình sẽ cho biết là còn bao nhiêu không gian đĩa cho phần mở rộng. Mặc định nó sẽ hiển thị tổng số dung lượng đĩa còn lại. Nếu bạn đồng ý chỉ sử dụng một ổ đĩa chính và một ổ đĩa mở rộng, chỉ việc ấn phím Enter. Ngược lại, gõ vào số Megabyte phù hợp, cứ như thế, tạo các ổ đĩa logic cho đến hết toàn bộ đĩa.

Sau đó, ấn phím Esc và ấn số 4 để đọc thông tin các phần vừa tạo. Sau cùng, khởi động lại máy và tiến hành định dạng mức cao.

d. Định dạng mức cao

Sau khi hoàn tất công đoạn FDisk, khởi động lại máy, trở về ổ đĩa A và định dạng mức cao cho ổ đĩa C. Do việc sẽ sử dụng ổ đĩa C thành đĩa khởi động nên ta phải chuyển các tập tin hệ thống (*Command.com, MSDOS.sys, IO.sys*) từ đĩa A sang đĩa C, thao tác định dạng có chuyển các tập tin hệ thống này sang đĩa C bằng lệnh:

```
Format C:/S <Enter>
```

DOS sẽ hiển thị thông báo:

Warning! All data on non-removable disk drive C: will be lost !

Proceed with format (Y/N)

Bảng này cảnh báo: *Tất cả dữ liệu trên ổ đĩa C: sẽ bị mất. Tiếp tục định dạng (Y/N)*

Nếu bạn gõ Y, ổ đĩa sẽ bắt đầu được định dạng. Tùy theo dung lượng ổ đĩa lớn hay nhỏ mà thời gian định dạng sẽ nhanh hay chậm, xong DOS hiển thị:

Format complete

System transferred

Volume label (11 characters, ENTER for none) ?

Có nghĩa: quá trình định dạng đã hoàn tất - Các tập tin hệ thống đã được truyền tải - Bạn có thể gán tên nhãn cho ổ đĩa nếu bạn muốn.

Quá trình trên chỉ mới định dạng cho đĩa C, nếu bạn có các phần khác hay một ổ cứng thứ hai bạn cũng tiến hành định dạng tương tự như vậy nhưng không chuyển các tập tin khởi động (*/S trong câu lệnh Format*) sang các phần khác hay các ổ đĩa khác.

-----Š----->-----

§.9. GIỚI THIỆU VỀ BIOS VÀ CMOS

Một số người thường lầm lẫn giữa BIOS và CMOS trong hệ thống. Thực ra chúng là hai phần hoàn toàn tách biệt nhau.

BIOS trên mainboard được lưu trữ trên một chip ROM cố định, bởi vậy các thông tin của nó không thể thay đổi. Để có thể hoạt động, BIOS cần phải được cung cấp các tham số phù hợp khác nhau. Các thông số này được lưu trữ thường trực trong một loại chip nhớ có tên RTC/NVRAM. Chip này đóng vai trò là đồng hồ thời gian thực (Real - Time Clock), giữ xung nhịp số. Nó có vài bytes bộ nhớ phụ. Chip đầu tiên được sử dụng là chip Motorola MC 1468, có 64 bytes bộ nhớ lưu trữ, trong đó, 10 bytes dành cho chức năng clock. Khi không có nguồn điện cung cấp, dữ liệu và các cài đặt time/ date trong phần bộ nhớ phụ sẽ bị xoá hết.

Chip này được thiết kế theo công nghệ CMOS (*Complementary Metal Oxide Semiconductor*), nó hoạt động được nhờ một viên pin nhỏ và cần rất ít năng lượng, chỉ với dòng điện 1 microampe. Khoảng 5 năm, pin này sẽ hết và các dữ liệu trên RTC/NVRAM bị xoá.

Khoảng từ năm 1996 về sau, các PC thường sử dụng Flash ROM để lưu trữ BIOS. Flash ROM là chip EEPROM (Electrically Erasable Programmable Read Only Memory). Đối với Flash ROM, có thể xoá và ghi lại bằng việc lập trình mà không cần phải tháo ra.

Việc xác lập các thông tin cho chip này gọi là *BIOS Setup*. Trên thị trường hiện nay, thông thường, công việc *BIOS Setup* do nơi cung cấp máy tính thực hiện ngay sau khi ráp bộ máy tính. Tuy nhiên, bạn cũng phải biết cách *BIOS Setup* để đề phòng máy tự mất các thông tin lưu trong BIOS vì các lý do như: hết pin, nhiễu điện, virus phá hoại, v.v... Tùy mỗi loại mainboard theo từng hãng chế tạo (*Award, Ami, Pheonix...*) mà các mục trong BIOS Setup có thể khác nhau, tuy nhiên, về căn bản chúng tương tự nhau. Trong phần này, chúng ta bàn về các tính năng phổ biến, còn các tính năng riêng, mới của mỗi BIOS trên các mainboard khác nhau bạn phải tự tìm hiểu thêm nhờ vào các kiến thức căn bản này.

Màn hình BIOS Setup đa số là màn hình chạy ở chế độ Text. Gần đây, đang phát triển loại BIOS Win (Ami) có màn hình Setup thể hiện như khi chạy trong Windows và sử dụng được Mouse trong khi Setup, đương nhiên các mục vẫn không thay đổi.

Lưu ý:

- Thao tác để vào BIOS Setup tùy mỗi loại BIOS của các hãng sản xuất trên mainboard nên sẽ khác nhau, ta sẽ ấn các phím quy định trong quá trình POST để vào:

- + ấn *Delete* đối với AMI BIOS.
- + ấn *F2* đối với Phoenix BIOS.
- + ấn *Ctrl - Alt - Esc* hoặc *Delete* đối với Award BIOS.
- + ấn *Esc* đối với Microid Research BIOS.

- Trường hợp, máy thuộc dạng chính hãng:

- + ấn *F1* đối với IBM Aptiva/Valuepoint.
- + ấn *Ctrl - Alt - Esc* hoặc *Ctrl - Alt - S* đối với Older Phoenix BIOS.
- + ấn *F10* đối với máy Compaq.

- Như vậy, cái mà người ta thường gọi là CMOS thực ra là một loại chip nhỏ, còn CMOS chỉ là tên một công nghệ chế tạo ra chip nhỏ đó. Tên đúng của chip này là RTC/NVRAM còn gọi là CMOS RAM.

- Chương trình BIOS Setup thường sử dụng các phím mũi tên để chọn lựa các mục. Thay đổi giá trị của các mục đang Set bằng hai phím *Page Up* và *Page Down*, hoặc "+" và "-", ấn *Esc* để thoát khỏi mục. Ấn *F10* để thoát khỏi BIOS Setup, nếu muốn lưu các thay đổi, khi hộp thoại hiện ra, bấm *Y* để lưu, *N* để trở lại màn hình BIOS Setup.

1. Setup các thành phần căn bản (Standard CMOS Setup)

Đây là các thành phần căn bản của BIOS trên tất cả các PC gồm các thông số về ngày, giờ, ổ đĩa cứng, ổ đĩa mềm v.v... Ngoài ra, mục này còn cho biết thêm các thông tin về bộ nhớ hiện có trên máy.

Ngày, giờ (Date/Day/Time):

Bạn khai báo ngày, tháng, năm vào mục này. Khai báo này sẽ được máy tính xem là thông tin gốc và sẽ bắt đầu tính từ đây trở đi. Các thông tin về ngày giờ được sử dụng khi thao tác các tập tin, thư mục, v.v... Có nhiều loại chương trình khi chạy cũng cần thông tin này. Thí dụ để báo cho bạn cập nhật khi quá hạn, chấm dứt hoạt động khi đến ngày quy định v.v... Bình thường, phần này bị Set sai hay không Set cũng chẳng ảnh hưởng gì đến hoạt động của máy. Các hoạt động này có thể sửa chữa trực tiếp ngoài DOS bằng hai lệnh *Date* và *Time*, hay bằng Control Panel của Windows mà không cần vào BIOS Setup.

Lưu ý: Đồng hồ máy tính có khi chạy chậm khoảng vài giây/ngày, thỉnh thoảng bạn nên chỉnh lại giờ cho đúng. Nhưng nếu quá chậm là có vấn đề cần phải xem lại mainboard.

Ổ đĩa mềm (Drive A/B):

Khai báo loại ổ đĩa cho ổ A và B, bạn căn cứ vào việc nối dây cho ổ đĩa để xác định ổ đĩa nối với đầu nối ngoài cùng của dây nối là ổ A, ổ kia là ổ B, ổ có kích thước lớn là 1.2MB - 5.25 inch, ổ nhỏ là 1.44MB - 3.5 inch. Nếu không có thì chọn *Not Install*. Trường hợp, có ổ đĩa nhưng ta khai báo *Not Install* sẽ làm cho ổ đĩa mềm không hoạt động nhưng không ảnh hưởng gì, khi cần sử dụng ta khai báo lại. Trong các Mainboard sử dụng BIOS đời mới, khai báo sai loại ổ đĩa 1.2MB thành 1.44MB và ngược lại, ổ đĩa vẫn hoạt động bình thường nhưng kêu rất lớn lúc mới đầu đọc đĩa, về lâu dài có thể mau chóng làm hỏng đĩa.

Các BIOS và các Card I/O đời mới có thể cho phép bạn tráo đổi hai ổ đĩa mềm mà không cần tráo đổi dây (swap floppy drive), tức là ổ A thành ổ B và ngược lại khi sử dụng. Khi tráo đổi bằng cách Set Jumper trên Card I/O, bạn nhớ khai báo lại trong BIOS Setup (khi tráo đổi bằng lệnh Swap trong BIOS thì không cần khai báo lại), nhưng có ứng dụng không chịu cài đặt khi Swap đĩa mềm, nhất là các ứng dụng có bảo vệ chống sao chép.

Ổ đĩa cứng (Drive C/D) loại IDE:

Phần khai báo ổ đĩa cứng rắc rối hơn, bắt buộc bạn phải khai báo chi tiết các thông số, bạn khai báo sai không những ổ cứng không hoạt động mà đôi khi còn làm hư ổ cứng nếu bạn khai báo quá dung lượng thật sự của ổ cứng và cho tiến hành FDISK. FORMAT theo dung lượng sai này. May mắn là các BIOS sau này đều có phần dò tìm thông số ổ cứng IDE tự động (IDE HDD auto detection) nên các bạn khỏi mất công nhớ khi sử dụng ổ cứng loại IDE. Chúng tôi sẽ nói về phần Auto Detect này sau. Ngoài ra, các ổ cứng sau này đều có ghi thông số lên nhãn dán trên mặt. Bạn cho chạy auto detect, BIOS sẽ tự động điền các tham số này dùm bạn. Việc khai báo ổ cứng C và D đòi hỏi phải đúng với việc Set các Jumper trên hai ổ cứng. Bạn xác lập ổ cứng không phải qua đầu nối dây mà bằng các Jumper trên mạch điều khiển ổ cứng. Các ổ cứng đời mới chỉ có một Jumper 3 vị trí: Ổ duy nhất, ổ Master (ổ C), ổ Slave (ổ D) và có ghi rõ cách Set trên nhãn. Các ổ đĩa cứng đời cũ nhiều Jumper hơn nên nếu không có tài liệu hướng dẫn là rắc rối, phải mò mẫm rất lâu.

- Ổ đĩa cứng (Drive E/F) loại IDE:

Các BIOS và các Card I/O đời mới cho phép gắn với 4 ổ đĩa cứng, vì hiện nay các ổ đĩa CDROM cũng sử dụng đầu nối ổ cứng để hoạt động, gọi là CDROM Interfaces IDE (giao diện đĩa IDE) để đơn giản việc lắp đặt.

Chú ý: Khai báo là NONE trong BIOS Setup cho ổ đĩa CD-ROM.

- **Màn hình (Video) - Primary Display:**

+ EGA/VGA: Dành cho màn hình sử dụng Card màu EGA hay VGA, Supper VGA.

+ CGA 40/CGA 80: Dành cho loại màn hình sử dụng Card màu CGA 40 cột hay CGA 80 cột.

- **Treo máy nếu phát hiện lỗi khi khởi động (Error Halt):**

+ Tắt cả các lỗi (All error): Treo máy khi phát hiện bất cứ lỗi nào trong quá trình kiểm tra máy, bạn không nên chọn mục này vì BIOS sẽ treo máy khi gặp lỗi đầu tiên nên bạn không biết các lỗi khác, nếu có.

+ Bỏ qua lỗi của Keyboard (All, but Keyboard): Tắt cả các lỗi ngoại trừ lỗi của bàn phím.

+ Bỏ qua lỗi đĩa (All, but Diskette): Tắt cả các lỗi ngoại trừ lỗi của đĩa

+ Bỏ qua lỗi đĩa và bàn phím (All, but Disk/Key): Tắt cả các lỗi ngoại trừ lỗi của đĩa và bàn phím.

+ Không treo máy khi có lỗi (No error): Tiến hành quá trình kiểm tra máy cho đến khi hoàn tất dù phát hiện bất cứ lỗi gì. Bạn nên chọn mục này để biết máy bị trục trặc ở bộ phận nào mà có phương hướng giải quyết.

- **Keyboard:**

+ Install: Cho kiểm tra bàn phím trong quá trình khởi động, thông báo trên màn hình nếu bàn phím có lỗi.

+ Not Install: Không kiểm tra bàn phím khi khởi động

Chú ý: Chọn mục này không có nghĩa là vô hiệu bàn phím vì nếu vậy làm sao điều khiển máy. Nó chỉ có tác dụng cho BIOS khởi mất công kiểm tra bàn phím nhằm rút ngắn thời gian khởi động.

2. Setup các thành phần nâng cao (Advanced Setup):

Cho phép thiết lập các thông số về chống Virus, chọn Cache, thứ tự khởi động máy, các tùy chọn bảo mật v.v... Song chúng ta cần chú ý các thông số chính sau đây:

- **Virut Warning:** Nếu Enabled, BIOS sẽ báo động và treo máy khi có hành động viết vào Boot Sector hay Partition của ổ cứng. Nếu bạn cần chạy chương trình có thao tác vào 2 nơi đó như Fdisk, Format ... bạn cần phải Disable.
 - **Internal Cache:** Cho hiệu lực (Enable) hay vô hiệu hóa (Disable) cache (L1) nội trong CPU 586 trở về lên.
 - **External cache:** Cho hiệu lực (Enable) hay vô hiệu hóa (Disable) cache trên mainboard, còn gọi là Cache mức 2 (L2).
 - **Quyck Power On Seft Test:** Nếu Enable BIOS sẽ rút ngắn và bỏ qua vài mục không quan trọng trong quá trình khởi động, để giảm thời gian khởi động tối đa.
 - **About 1 MB Memory Test:** Nếu Enable BIOS sẽ kiểm tra tất cả bộ nhớ. Nếu Disable chỉ kiểm tra 1 MB bộ nhớ đầu tiên.
 - **Memory Test Tick Sound:** Cho phát âm (Enable) hay không (Disable) trong thời gian Test bộ nhớ.
 - **Etended BIOS Ram Area:** Khai báo mục này nếu muốn dùng 1 Kb trên đỉnh của bộ nhớ quy ước, tức 1Kb bắt đầu từ địa chỉ 639 K hay 0:300 của vùng BIOS hệ thống trong bộ nhớ quy ước để lưu các thông tin về đĩa cứng. Xác lập có thể 1K hay 0:300.
 - **Swap Floppy Drive:** Trao đổi tên hai ổ đĩa mềm, khi chọn mục này bạn không cần khai báo lại ổ đĩa như khi trao bằng cách Set Jumper trên Card I/O.
 - **Boot Sequence:** Chọn ổ đĩa khi BIOS tìm hệ điều hành khi khởi động. Có thể C rồi đến A hay A rồi đến C hay chỉ có C. Bạn nên chọn C, A hay chỉ có C, để đề phòng trường hợp vô tình khởi động bằng đĩa mềm có virus.
- Hiện nay trên các Mainboard Pentium. BIOS cho phép bạn chỉ định khởi động từ một trong hai ổ mềm hay trong 4 ổ cứng IDE hay bằng ổ cứng SCSI thậm chí bằng ổ CD-Rom cũng được.
- **Boot Up Floopy Seek:** Nếu Enable BIOS sẽ dò tìm kiểu của đĩa mềm là 80 track hay 40 track. Nếu disable BIOS sẽ bỏ qua. Chọn Enable làm chậm thời gian khởi động vì BIOS luôn luôn phải đọc đĩa mềm trước khi đọc đĩa cứng, mặt dù bạn đã chọn chỉ khởi động bằng ổ đĩa C.
 - **Boot Up Numlock Status:** Nếu ON là cho phím Numlock mở (đèn Numlock sáng) sau khi khởi động, nhóm phím bên tay phải bàn phím dùng để đánh số. Nếu OFF là phím Numlock tắt (đèn Numlock tối) , nhóm phím bên tay phải dùng để di chuyển con trỏ.
 - **Boot Up System Speed:** Quy định tốc độ CPU trong thời gian khởi động là High (cao) hay Low (thấp).
 - **Memory Parity Check:** Kiểm tra chẵn lẻ bộ nhớ. Chọn theo mainboard vì có loại cho phép mục này Enable có loại bắt bạn chọn Disable mới chịu chạy. Đầu

tiên bạn chọn Enable, nếu máy treo bạn chọn lại là Disable. Mục này không ảnh hưởng đến hệ thống, chỉ có tác dụng kiểm tra RAM.

- **IDE HDD Block Mode:** Nếu ổ cứng của bạn hỗ trợ kiểu vận chuyển dữ liệu theo từng khối (các ổ đĩa đời mới có dung lượng cao). Bạn cho Enable để tăng tốc cho ổ đĩa. Nếu ổ đĩa đời cũ bạn cho Disable mục này.

- **Pri.Master/Slave LBA (Logic Block Addressing) Mode:** Nếu hai ổ đĩa cứng được nối vào đầu nối Primary của card I/O có dung lượng lớn hơn 528 MB, bạn cho Enable mục này.

- **Sec.IDE Ctrl Drive Install:** Mục này để khai báo máy bạn có ổ đĩa cứng nối vào đầu nối Secondary của card I/O. các chỉ định có thể là Master, Mst/Slv và Disable.

- **Sec Master/Slave LBA Mode:** Xác lập LBA cho đầu nối thứ 2.

Chú ý: Các mục hỗ trợ cho ổ đĩa cứng có dung lượng lớn và các card I/O đời mới giúp bạn sử dụng ổ đĩa có dung lượng trên 528 MB. Trong trường hợp bạn cho Enable các mục này rồi mới tiến hành Fdisk và Format đĩa, nếu sau đó bạn lại Disable các mục này hay đem gắn qua máy khác cũng chọn Disable, bạn sẽ không thể sử dụng được ổ đĩa cứng. Khi dùng ổ CD-ROM có đầu nối IDE, bạn nên gắn vào đầu nối Secondary để khỏi ảnh hưởng đến ổ đĩa cứng (gắn vào đầu nối Pri) khi cần chạy 32 Bit DiskAccess trong Windows.

- **Typematic Rate Setting:** Nếu Enable là bạn cho 2 mục dưới đây có hiệu lực. Hai mục này thay thế lệnh Mode của DOS, quy định tốc độ và thời gian trễ của bàn phím.

- **Typematic Rate (Chars/Sec):** Bạn lựa chọn số ký tự /giây tùy theo tốc độ đánh phím nhanh hay chậm của bạn. Nếu bạn Set thấp hơn tốc độ đánh thì máy sẽ phát tiếng Bíp khi nó chạy theo không kịp.

- **Typematic Delay (Msec):** Chỉ định thời gian lặp lại ký tự khi bạn ấn và giữ luôn phím, tính bằng mili giây.

- **Security Option:** Mục này dùng để giới hạn việc sử dụng hệ thống và BIOS Setup.

Setup: Giới hạn việc thay đổi BIOS Setup, mỗi khi muốn vào BIOS Setup bạn phải đánh đúng mật khẩu đã quy định trước.

System hay Always: giới hạn việc sử dụng máy. Mỗi khi mở máy, BIOS luôn luôn hỏi mật khẩu, nếu không biết mật khẩu Bioc sẽ không cho phép sử dụng máy.

Chú ý:Trong trường hợp bạn chưa chỉ định mật khẩu, để Disable (vô hiệu hóa) mục này, bạn chọn Password Setting, bạn đừng đánh gì vào ô nhập mật khẩu mà chỉ cần bấm ENTER. Trong trường hợp bạn đã có chỉ định mật khẩu nay

lại muốn bỏ đi. Bạn chọn Password setting bạn đánh mật khẩu cũ vào ô nhập mật khẩu cũ (Old Password) còn trong ô nhập mật khẩu mới (New Password) bạn đừng đánh gì cả mà chỉ cần bấm ENTER. Còn mainboard thiết kế thêm một jumper để xóa riêng mật khẩu ngoài jumper để xóa toàn bộ thông tin trong CMOS. Tốt hơn hết là bạn đừng sử dụng mục này vì bản thân chúng tôi chứng kiến nhiều trường hợp dở khóc dở cười do mục này gây ra. Lợi ít mà hại nhiều. Chỉ những máy tính công cộng mới chỉ sử dụng mục này thôi.

- **System BIOS Shadow, Video BIOS Shadow:** Nếu Enable là cho copy các dữ liệu về System và video trong BIOS (có tốc độ chậm) vào RAM (tốc độ nhanh) để rút ngắn thời gian khi cần truy cập vào các dữ liệu này

- **Wait for <F1> if Any Error:** Cho hiện thông báo chờ ấn phím F1 khi có lỗi.

- **Numeric Processor:** Thông báo gắn CPU đồng xử lý (Present) trên máy hay không (absent). Mục này thường có trong các máy dùng CPU 286, 386, 486 SX. Từ 486 DX trở về sau đã có con đồng xử lý bên trong CPU nên trên các máy mới có thể không có mục này.

- **Turbo Switch Function:** Cho nút Turbo có hiệu lực (Enable) hay không (Disable). Mục này thường thấy ở các BIOS đời cũ, trên các máy đời mới lựa chọn này thường bằng cách Set Jumper của Mainboard. Từ Mainboard Pentium trở đi không có mục này.

3. Setup các thành phần có liên quan đến vận hành hệ thống (Chipset Setup):

Thiết lập các thông số cho các chip phụ trợ như RAM, Cache định thời khởi động từ xa, Cache cho VIDEO BIOS v.v...

- **Auto Configuration:** Nếu Enable, BIOS sẽ tự động xác lập các thành phần về DRAM, Cache ... mỗi khi khởi động tùy theo CPU type (kiểu CPU) và System clock (tốc độ hệ thống). Nếu Disable là để cho bạn tự chỉ định.

- **AT Clock Option:** Nếu Asyne (không đồng bộ) là lấy dao động chuẩn của bộ dao động thạch anh chia đôi àm tốc độ hoạt động cho AT Bus (bus-8-16 Bit). Thường là 14.318Mhz/2 tức 7159 MHz. Có BIOS còn cho chọn tốc độ của mục này là 14,318 Mhz. Nếu Syns (đồng bộ) là dùng System Clock (do bạn chỉ định bằng cách Set Jumper trên Mainboard) làm tốc độ chuẩn.

- **Synchronoun AT Clock/AT Bus Clock Selector:** Chỉ định tốc độ hoạt động cho AT Bus bằng cách lấy tốc độ chuẩn (System Clock) chia nhỏ để còn lại khoảng 8M cho phù hợp với Card 16 bit. Các lựa chọn như sau:

-CLKI/3 khi System clock là 20-25 MHz.

- CLKI/4 khi System clock là 33 MHz.
- CLKI/5 khi System clock là 40 MHz.
- CLKI/6 khi System clock là 50 MHz.

Tốc độ này càng lớn (số chia càng nhỏ), máy chạy càng nhanh do tăng tốc độ vận chuyển dữ liệu. Tuy nhiên lớn đến đâu còn phụ thuộc vào Mainboard và card cắm trên các Slot (quan trọng nhất là các I/O). Các bạn phải thí nghiệm giảm số chia từng nấc và chú ý máy có khởi động hay đọc đĩa bình thường không, nếu phát sinh trục trặc thì giảm xuống một nấc. Thường thì bạn có thể tăng được hai nấc. Thí dụ: System clock là 40 MHz, bạn chọn CLKI/3. Card ISA 8 và 16 bit có thể chạy tốt trong khoảng từ 8 MHz-14 MHz. Nếu nhanh quá, thường Card I/O gặp trục trặc trước (không đọc được đĩa cứng).

- **AT Cycle Wait States/Extra AT Cycle WS:** Để Enable hay Disable việc chèn thêm một thời gian chờ vào thời gian chuẩn của AT Bus. Nếu System clock dưới 33 MHz chọn Disable. Nếu trên 33 MHz chọn Enable.

- **Fast AT Cycle:** Khi Enable sẽ rút ngắn thời gian chuẩn của AT Bus.

- **DRAM ReadWait States/DRAM Burst Cycle:**

Dưới 33 MHz là: 3-2-2-2 hay 2-1-1-1.

Từ 33 đến 45 MHz là: 4-3-3-3 hay 2-2-2-2.

50 MHz là: 5-4-4-4 hay 3-2-2-2.

Chọn mục này ảnh hưởng lớn đến tốc độ CPU.

- **DRAM?Memory Write Wait States:**

Chọn IWS kkhi hệ thống nhanh hay DRAM chậm (tốc độ 40 MHz trở lên). Chọn OWS khi hệ thống và DRAM có thể tương thích (33 MHz trở xuống).

- **Hidden Refresh Option:** Khi Enable, CPU sẽ làm việc nhanh hơn do không phải chờ mỗi khi DRAM được làm tươi.

- **Slow Refresh Enable:** Mục này nhằm bảo đảm an toàn dữ liệu trên DRAM, thời gian làm tươi sẽ kéo dài hơn bình thường. Bạn chỉ được Enable mục này khi bộ nhớ của máy hỗ trợ việc cho phép làm tươi chậm.

- **L1 Cache Mode:** Lựa chọn giữa Write-Through và Write-Back cho Cache nội trong CPU 486 trở lên. Xác lập Write-Through máy sẽ chạy chậm hơn Write-Back. Nhưng việc lựa chọn còn tùy thuộc vào loại CPU.

- **L2 Cache Mode:** Xác lập cho Cache trên Mainboard.

- **IDE HDD Auto Detection/IDE SETUP:**

Khi chọn mục này sẽ xuất hiện một cửa sổ cho bạn chỉ định ổ đĩa cần dò tìm thông số (2 hay 4 ổ đĩa tùy theo BIOS) sau đó bấm OK hay Yes để BIOS điền vào phần Standard dùng cho bạn. Trong BIOS đời mới, auto Detect có thể

đưa ra vài loại ổ đĩa. Tùy theo cách sử dụng ổ đĩa (Normal, LBA, ...) mà bạn chọn loại thích hợp.

4. Power Management Setup:

Đối với CPU 486:

Phần này là các chỉ định cho chương trình tiết kiệm năng lượng sẵn chứa trong các BIOS đời mới. Chương trình này dùng được cho cả hai loại CPU: Loại thường và loại CPU kiểu S. CPU kiểu S hay CPU có hai ký tự cuối SL là một loại CPU được chế tạo đặc biệt, có thêm bộ phận quản lý năng lượng trong CPU. Do đó trong phần có hai loại chỉ định dành cho hai loại CPU.

Đối với Pentium:

Dùng chung cho mọi loại Pentium hay các chip của các hãng khác cùng đời với Pentium.

- *Power Management/Power Saving Mode:*

Disable: Không sử dụng chương trình này.

Enable/User Define: Cho chương trình này có hiệu lực.

Min Saving: Dùng các giá trị thời gian dài nhất cho các lựa chọn (tiết kiệm năng lượng ít nhất).

- *Pmi/Smi:* Nếu chọn Smi là máy đang gắn CPU kiểu S của hãng Intel. Nếu chọn auto là máy gắn CPU thường.

- *Doze Timer:* Mục này chỉ dùng cho CPU kiểu S. Khi đúng thời gian máy đã rảnh (không nhận được tín hiệu từ các ngắt) theo quy định CPU tự động hạ tốc độ xuống còn 8 MHz. Bạn chọn thời gian theo ý bạn (có thể từ 10 giây đến 4 giờ) hay Disable nếu không muốn sử dụng mục này.

- *Sleep timer/Standby Timer:* Mục này chỉ dùng cho CPU kiểu S. Chỉ định thời gian máy rảnh trước khi vào chế độ Sleep (ngưng hoạt động). Thời gian có thể từ 10 giây đến 4 giờ.

- *Sleep Clock:* Mục này chỉ dùng cho CPU kiểu S: Stop CPU hạ tốc độ xuống còn 0 MHz (ngưng hẳn). Slow CPU hạ tốc độ xuống còn 8 MHz.

- *HDD Standby Timer/HDD Power Down:*

Chỉ định thời gian ngừng motor của ổ đĩa cứng.

- *CRT Sleep:* Nếu chọn Enable là màn hình sẽ tắt khi máy vào chế độ Sleep.

Chỉ định: Các chỉ định cho chương trình quản lý nguồn biết cần kiểm tra bộ phận nào khi chạy.

Chú ý: Do BIOS được sản xuất để sử dụng cho nhiều loại máy khác nên các bạn luôn gặp phần này trong các BIOS. Thực ra chúng chỉ có giá trị trong các máy xách tay (laptop) vì xài Pin nên vấn đề tiết kiệm năng lượng được đặt lên

hàng đầu. Chúng tôi khuyên các bạn đang sử dụng máy để bàn (desktop) nên vô hiệu hóa tất cả các mục trong phần này, để tránh các tình huống bất ngờ như: Đang cài chương trình tự nhiên máy ngưng hoạt động, đang chạy Dafrag tự nhiên máy chậm cực kỳ...

5. Phần dành riêng cho mainboard theo chuẩn giao tiếp PCI có I/O và IDE On Board (Peripheral Setup):

- **PCI On Board IDE:** Cho hiệu lực (Enabled) hay vô hiệu (Disabled) 2 đầu nối ổ đĩa cứng IDE trên mainboard. Khi sử dụng card PCI IDE rồi, ta cần chọn Disabled.

- **PCI On Board Secondary IDE:** Cho hiệu lực (Enable) hay vô hiệu (Disable) đầu nối ổ đĩa cứng IDE thứ hai trên mainboard. Mục này bổ sung cho mục trên và chỉ có tác dụng với đầu nối thứ 2.

- **PCI On Board Speed Mode:** Chỉ định kiểu vận chuyển dữ liệu (PIO Speed Mode). Có thể là Disable, mode 1, mode 2, mode 3, mode 3, auto. Trong đó mode 4 là nhanh nhất.

- **PCI Card Present on:** Khai báo sử dụng Card PCI IDE rồi hay không và nếu có thì được cắm vào Slot nào. Các mục chọn là: Disable, auto, Slot 1, Slot 2, Slot 3, Slot 4.

- **PCI IRQ, PCI Primary IDE IRQ, PCI Secondary IDE IRQ:**

Chỉ định cách xác lập ngắt cho PCI IDE rồi.

Chú ý: Trong mục này có phần xác lập thứ tự gán ngắt cho các Card bổ sung. Thí dụ: 1=9, 2=10, 3=11, 4=12 có nghĩa là Card đầu tiên cắm vào bất kỳ Slot nào sẽ được gán ngắt 9, nếu có 2 Card thì Card cắm vào Slot có số thứ tự nhỏ sẽ được gán ngắt 9, Slot có số thứ tự lớn sẽ được gán ngắt 10 v.v...

- **IDE 32 Bit Transfers Mode:**

Xác lập nhằm tăng cường tốc độ cho ổ đĩa cứng trên 528 MB, nhưng cũng có ổ đĩa không khởi động được khi Enable mục này dù Fdisk và Format vẫn bình thường.

- **Host to PCI Post Write W/s, Host to Burst Write Host to DRAM Burst Write:**

Các xác lập này cho CPU Bus, không ảnh hưởng nhiều đến tốc độ của CPU, có thể để nguyên xác lập mặc nhiên.

- **PCIBus Park, Post Write Buffer:**

Khi Enable các mục này có thể tăng cường thêm tốc độ hệ thống

- **FDC Control:**

Cho hiệu lực hay không đầu nối cáp và xác lập địa chỉ cho ổ đĩa mềm.

- Primary Serial Port:

Cho hiệu lực hay không cổng COM 1 và xác lập địa chỉ cho cổng này.

- Secondary Serial Port:

Cho hiệu lực hay không cổng COM 2 và xác lập địa chỉ cho cổng này.

Chú ý: Nếu bạn sử dụng Card bổ sung có xác lập địa chỉ là COM 1 và COM 2, bạn phải Disable cổng tương ứng trong hai mục trên.

- Parallel Port:

Cho hiệu lực hay không Cổng LPT 1 và xác lập địa chỉ cho cổng này.

6. Hướng dẫn BIOS Setup:

Trong các tài liệu đi kèm mainboard, đều có hướng dẫn BIOS Setup. Khi mua máy hay mua Mainboard, các bạn nhớ đòi các tài liệu này vì nó rất cần cho việc sử dụng máy.

Trong các phần Setup trên, phần Standard. Advanced có ảnh hưởng đến việc cấu hình máy. Phần Chipset ảnh hưởng đến tốc độ máy. Phần PCI ảnh hưởng đến các gán ngắt, địa chỉ cho các Slot PCI, cổng; cách vận chuyển dữ liệu cho IDE On Board.

Nếu gặp các thành phần hoàn toàn mới, trước tiên bạn hãy Set các thành phần đã biết, kiểm tra việc thay đổi của máy, cuối cùng mới Set tới các thành phần chưa biết. Chúng tôi xin nhắc lại, việc BIOS Setup sai không bao giờ làm hư máy và các bạn sẽ dễ dàng Setup lại nhờ vào chính BIOS. Trên Mainboard luôn luôn có một Jumper dùng để xóa các thông tin chứa trong CMOS để bạn có thể tạo lại các thông tin này trong trường hợp không thể vào lại BIOS Setup khi khởi động máy.

Khi tiến hành tìm hiểu BIOS Setup, bạn nên theo một quy tắt sau: Chỉ Set từng mục một rồi khởi động máy lại, chạy các chương trình kiểm tra để xem tốc độ CPU, ổ đĩa có thay đổi gì không? Cách làm này giúp bạn phát hiện ảnh hưởng của từng mục vào hệ thống và bạn có thể biết chắc trực tiếp phát sinh do mục nào để sửa chữa. Khi xảy ra trục trặc mà bạn không biết cách đối phó, bạn chỉ cần vào lại BIOS Setup chọn Load BIOS Default hay bấm F6 trong phần Set mà bạn muốn phục hồi sau đó khởi động máy lại là xong.

PHẦN III**CÀI ĐẶT PHẦN MỀM MÁY TÍNH**

Trong phần trước ta đã xét về cấu trúc cũng như cách làm việc của các thành phần cấu tạo nên máy tính và đã cài đặt hệ điều hành đơn giản nhất là MSDOS. Song đối với các hệ điều hành, các chương trình lớn đòi hỏi sự tương thích về cấu hình cũng như quá trình cài đặt của nó phải chính xác thì mới có thể làm việc có hiệu quả, nhất là các phần mềm đồ họa như Windows, Microsoft Office v.v...

Cài đặt phần mềm là quá trình xác định nguồn tài nguyên mà hệ điều hành, phần mềm đó được sử dụng trên hệ thống và các thành phần của phần mềm được sử dụng. Từ đó phân bố các thông tin này vào các file chương trình khởi động hay các file cấu hình cho phù hợp. Có thể đơn cử quá trình cài đặt chung của phần mềm gồm các bước sau:

- Kiểm tra các tài nguyên hệ thống có đảm bảo không như CPU, RAM, Màn hình, Bàn phím, Chuột, không gian đĩa v.v...
- Xác định các thành phần của phần mềm cài đặt.
- Chép các file chương trình, dữ liệu lên đĩa đích.
- Kiểm tra tất cả các thành phần hệ thống và đưa thông tin vào các file *.sys hay *.ini.
- Cập nhật các thông tin đi cùng với chế độ khởi động cũng như các điều kiện làm việc. Tiêu biểu là các file Config.sys và Autoexec.bat .
- Xác định các thành phần hiện có cho phần mềm và cập nhật các logo đi cùng.
- Xây dựng các trình điều khiển thiết bị hệ thống cho phù hợp nếu cần.

Trong phần này ta xét hai quá trình cài đặt tiêu biểu và thường gặp là Microsoft Windows 95 và Microsoft Office 97. Sau đó xét thêm các thành phần phụ trợ như Vietware và các trình quản lý thiết bị.

.10. GIỚI THIỆU QUÁ TRÌNH CÀI ĐẶT WINDOWS

Windows là một hệ điều hành sử dụng giao diện đồ họa, do đó nó đòi hỏi việc cấu hình phần cứng phải chính xác. Ngoài ra Windows còn quản lý tất cả

các tài nguyên của máy tính và cập nhật các trình điều khiển của các thiết bị để làm việc cho thích hợp. Do đó để Windows làm việc tốt yêu cầu phải cài đặt Windows. Vì quá trình cài đặt cho các phiên bản Windows 9.x trở lên được sử dụng hiện nay, rất giống với quá trình cài đặt của Windows 95 nên ở đây ta chỉ xét cho quá trình cài đặt của Windows 95.

1. Chuẩn bị phần cứng:

1. CPU và Mainboard 486 trở lên (386 vẫn được nhưng rất chậm và ít được dùng hiện nay).

2. 8MB RAM bộ nhớ trong, nhưng càng nhiều càng tốt. (Đối với Windows 98 tối thiểu 16 MB RAM).

3. Ổ đĩa cứng trống từ 30MB trở lên đã có các file hệ thống để khởi động được máy. (Tùy theo phiên bản có thể khác nhau).

4. Màn hình và Card màn hình màu.

5. Bàn phím và Chuột.

6. Các thiết bị khác có thể có như CDROM, máy in, modem v.v...

2. Chuẩn bị phần mềm

Để cài Window 98 ta phải có bộ chương trình nguồn có file Setup và số xê_ri (Serial Number) của nó. Bộ chương trình này vào khoảng vài trăm MB tùy theo phiên bản và các thành phần đi kèm. Bộ chương trình nguồn này có trong đĩa cứng, CDROM hoặc mạng máy tính có sẵn.

* Các thành phần của Window 9x

Windows được thiết kế theo các module ghép lại với nhau, do đó ta có thể chọn và cài đặt các thành phần thích hợp. Có thể chia Windows 9x thành các thành phần chính như sau:

1. Phần nhân (Kernel) thành phần cốt lõi của Windows, do đó khi cài đặt không thể thiếu phần này.

2. Phần hỗ trợ các trình điều khiển thiết bị như Chuột, màn hình, bàn phím, máy in v.v... Phần này sử dụng cho chế độ Plus and Play.

3. Các công cụ hệ thống như Scandisk, Defragmenter, Speed disk v.v...

4. Các tiện ích đi cùng như Wordpad, Notepad, Paintbrush v.v...

5. Các thành phần hỗ trợ mạng như mạng ngang hàng, Windows NT, Netware, Internet v.v... như Internet Explorer, Email, Microsoft Network v.v...

6. Các module hỗ trợ lập trình và giao diện phần mềm như các hàm API, dịch vụ chuyển file kết nhúng v.v...

*** Cài thẳng sâu Setup:**

/?-Help

/c-Không chấu SMARTDrive.

/id- Không kiểm tra dung lượng áéa.

/it- Không kiểm tra các chương trình tháàng trui.

/is- Không chấu Scandisk.

/l- Sái dùng Logitech mouse trong thài gian Setup.

/n- Không dùng mouse.

3. Tiến trình cài đặt Windows

1. Chạy file Setup.exe của chương trình cài đặt. File này chứa trong đĩa thứ nhất của bộ chương trình nguồn. Khi chạy Setup.exe, Windows 9x tự động chạy trình kiểm tra đĩa bằng chương trình Scandisk, nếu không muốn chạy trình này ta dùng Setup /is.

2. Chương trình Setup yêu cầu số xê_ri (Serial number) để bảo vệ bản quyền. Nếu cho đúng số này chương trình tiếp tục và yêu cầu nhập các thông tin của bạn như tên, cơ quan v.v... để sử dụng vào một số mục đích sau này.

3. Yêu cầu chỉ định nơi cài đặt để chương trình Setup copy các file cần thiết vào đó.

4. Chọn chế độ Plug and play cho cả tình điều khiển thiết bị.

5. Chọn chế độ cài đặt, có 3 tùy chọn cho chế độ này là:

- Typical: Cài đặt đầy đủ tất cả các thành phần của Windows .

- Custom: Cho người sử dụng chọn các thành phần cài đặt.

- Portable:Chè cài các thành phần chè sái dùng trăn máy tênh xáich tay.

- Compact: Cài tối thiểu, bỏ hết các tiện ích để đỡ tốn đĩa.

Nếu chọn ở chế độ Custom ta phải chọn các thành phần cài đặt như đã trình bày ở trên.

6. Cấu hình các thành phần mạng như Card mạng, địa chỉ, giao thức v.v...

7. Tọa đĩa cứu hộ dùng khi Windows bị lỗi.

8. Chương trình Setup cập nhật cấu hình và cho khởi động lại máy tính để kết thúc quá trình cài đặt.

Theo dõi quá trình cài đặt và ghi lại và xử lý các lỗi nếu có. Nếu không có lỗi thì quá trình cài đặt đã thành công.

4. Hoàn chỉnh windows sau khi cài đặt

a. Thêm bớt các module:

Trong quá trình cài đặt nếu có module nào đó cài đặt chưa thành công hoặc muốn thêm bớt chúng ta làm như sau:

- Đưa bộ nguồn Windows vào thiết bị trên máy để chuẩn bị sẵn sàng.
- Chạy file Setup và chỉ đường dẫn đến bộ nguồn trên.
- Chọn thành phần cần sửa chữa sau đó chọn Add để thêm vào và chọn Remove để bỏ đi.

b. Xem sự xung đột về thiết bị

Như ta đã biết các thiết bị giao tiếp với CPU qua một địa chỉ vùng nhớ gọi là vùng nhớ vào ra và một ngắt, do đó khi có sự trùng lặp giữa các thông tin này hay sai trình điều khiển thiết bị sẽ xảy ra xung đột làm cho các thành phần hoạt động không bình thường.

Để kiểm tra vấn đề trên ta vào mục Setting\Control panel\System\ Device để xem các cấu hình cho các thiết bị. Khi thiết bị bị lỗi có xuất hiện một dấu bên cạnh thiết bị đó, có thể là để xử lý ta làm như sau:

- Chọn trình điều khiển đó rồi chọn Remove để gỡ bỏ.
- Chọn Add để cấu hình lại cho phù hợp.

5. Quá trình khởi động của Windows 95

Windows 95 là một hệ điều hành nên nó quản lý quá trình khởi động của máy tính. Có nghĩa là nó thay đổi đoạn chương trình khởi động Bootstrap trong Boot Sector của đĩa khởi động. Thông qua quá trình này ta có thể chẩn đoán được các lỗi trên máy tính Có thể mô tả quá trình khởi động của Windows như sau:

- Khi bật máy quá trình khởi động được thực hiện trong quá trình khởi động DOS.
- Chương trình khởi động tìm các file khởi động như IO.SYS, MSDOS.SYS và có thể có WINBOOT.SYS. Ở đây file IO.SYS vai trò chủ đạo trong quá trình khởi động của Windows 95.
- Tiếp theo thực hiện các lệnh trong CONFIG.SYS và AUTOEXEC.BAT, sau đó là các lệnh xây dựng cấu hình cho Windows đã được cấu hình sẵn trong hệ thống.
- Tải các trình điều khiển thiết bị sử dụng trong giao diện đồ họa.

- Tải các trình điều khiển thiết bị cho chương trình, các cấu hình chương trình thông qua 2 file System.ini và Win.ini.

- Cuối cùng thực hiện file tạo môi trường làm việc cho người sử dụng như các chương trình, Shortcut, vào mạng v.v...

Toàn bộ quá trình khởi động của Windows 9.x trở lên được lưu giữ trong file Bootlog.txt. Khảo sát đầy đủ quá trình khởi động của Windows có thể cho ra các chuẩn đoán chính xác về lỗi trong Windows.

*** Ý nghĩa file Setuplog.txt**

Đây nghĩa là file ASSCII, được Windows tạo ra trong thời gia Setup. Nó chứa tất cả các thông tin về các hoạt động của chương trình Setup. File này có mục đích giúp người dùng tham khảo để có hướng khắc phục khi không Setup được.

Setup dùng file này khi chạy Smart Recovery nhằm tránh việc lặp lại lỗi đã mắc trước đó. Nếu Setup xấu, bạn cần khởi động lại máy tính Setup tìm đọc file SETUPLOG.TXT, khi đến thủ tục bị lỗi nó sẽ bỏ qua thủ tục này và tiếp tục quá trình Setup.

*** Ý nghĩa file DETLOG.TXT**

File này chứa các thông tin về phần cứng.

Nếu máy tính bị treo trong thời gian dò tìm phần cứng, bạn có thể phát hiện lỗi này bằng cách xem dòng cuối cùng của DETLOG.TXT.

*** Ý nghĩa file BOOTLOG.TXT**

BOOTLOG.TXT là một file text ASCII chứa đựng những thông tin về trạng thái khởi động của Windows 95. Được tạo ra trong thời gian Setup hay mỗi lần Windows 95 khởi động (Ấn F8 khi khởi động dòng "Starting Windows 95", chọn mục tạo BOOTLOG.TXT. BOOTLOG.TXT sẽ được lưu trong thư mục của ổ đĩa khởi động).

File này có thể được dùng để chữa lỗi khi Setup hay khi Windows bị trục trặc. Bạn căn cứ vào các dòng nạp có thông báo Failed hay dòng lệnh cuối cùng trước khi treo máy để biết lỗi do driver thiết bị nào.

10. Thay đổi địa chỉ nguồn Setup sau khi cài

Có nhiều trường hợp bạn cần thay đổi địa chỉ nguồn. Thí dụ như trước đây bạn Setup Windows 95 từ ổ đĩa cứng, bây giờ bạn có Windows trên CD ROM, bạn cần thay đổi địa chỉ nguồn Setup để có thể truy cập các file trên CD ROM.

Để thay đổi bạn làm theo các bước sau:



-Bấm nút Start, bấm Run, đánh regedit trong Open box, bấm OK để khởi động Registry Editor.

-Bạn lần lượt mở các mục theo thứ tự sau:

HKEY_LOCAL_MACHINE_SOFTWARE\Microsoft\Windows\Current Version\Setup.

Bấm kép vào mục SourcePath và sửa chữa lại địa chỉ trong ô Value data. Thoát Regedit.

Chú ý: Trước khi sửa bạn nhớ Backup các file Registry (SYSTEM.DAT và USER.DAT).

§.11. CÀI ĐẶT MICROSOFT OFFICE

Microsoft Office là bộ công cụ văn phòng mạnh nhất hiện nay. Nó tích hợp hầu hết các công cụ hỗ trợ cho công việc văn phòng như văn bản, tính toán, quản lý, trình diễn v.v... Do vậy yêu cầu sử dụng tài nguyên của nó cũng rất lớn và quá trình cài đặt nó cũng tương đối phức tạp.

Về yêu cầu phần cứng và phần mềm nó cũng giống như Windows song Microsoft Office yêu cầu không gian đĩa lớn hơn như 200MB cho các thành phần của Microsoft Office97. Tuy nhiên Microsoft Office 97 gồm nhiều thành phần nên ta có thể chọn các thành phần cài đặt cho thích hợp.

1. Các thành phần của Microsoft Office 97

Microsoft Office 97 được thiết kế theo các module ghép lại với nhau. Do đó trong quá trình cài đặt ta dễ dàng loại bỏ hay cài đặt các thành phần của chúng. Toàn bộ Microsoft Office 97 được chia thành các thành phần như sau:

- Microsoft Word: Trình xử lý văn bản.
- Microsoft Exel: Trình xử lý bảng tính.
- Microsoft Access: Hệ quản trị cơ sở dữ liệu.
- Microsoft Power Point: Phần trình diễn các chữ hình ảnh v.v...
- Các thành phần khác như Microsoft Tool, Convert, Database v.v...

2. Tiến trình cài đặt

1. Cho chạy file Setup.exe đi kèm với bộ chương trình nguồn để thực hiện tiến trình cài đặt.

2. Cho số xê_ri (Serial number) và các thông tin liên quan đến người sử dụng.
3. Chọn nơi để cài đặt Microsoft Office. (Mặc định là thư mục PROGRAM FILES\MSOFFICE). Chương trình cài đặt sẽ chép các file cần thiết vào đây.
4. Chọn chế độ cài đặt Typical, Custom, Minimum.
5. Chọn Custom và chọn các thành phần cài đặt lên máy tính của bạn.
6. Chương trình Setup sẽ cập nhật hệ thống để hoàn thành quá trình cài đặt.
7. Kiểm tra các thành phần cài đặt đã hoàn thiện chưa. Bằng cách vào từng thành phần một và kiểm tra các chức năng.

3. Hoàn thiện các thành phần sau khi cài đặt

Sau khi cài đặt hoặc sau một thời gian sử dụng ta thấy phần nào còn thiếu hoặc thừa ta có thể sửa lại như sau:

1. Cho bộ nguồn Office vào thiết bị để sẵn sàng.
2. Chạy file setup cho đến mục Add, Remove.
3. Chọn Add và chọn các thành phần để thêm vào.
4. Chọn Remove và chọn các thành phần cần bỏ đi.
5. Xem lại các thành phần vừa sửa xong.

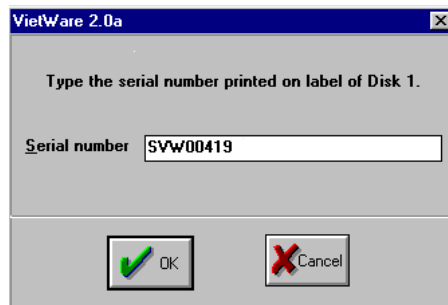
§.12. CÀI ĐẶT CÁC PHẦN MỀM KHÁC

1. Cài các Font tiếng việt

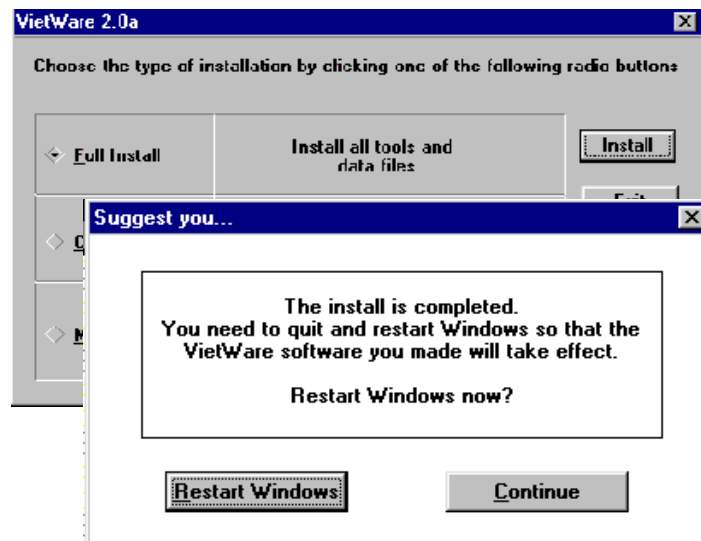
Bạn có thể cài các font tiếng việt khác nhau như Font ABC, Vietware, Vietkey,...nhưng đều có chung quy tắc là bạn chọn thư mục có chứa chương trình nguồn, sau đó bạn chọn tập tin **setup.exe**. Tiếp đến bạn gõ mã số đăng ký, chọn nơi để cài đặt, chọn chế độ cài đặt (Typical, Custom, Minimum). Chương trình Setup sẽ cập nhật hệ thống để hoàn thành quá trình cài đặt. Cuối cùng bạn chọn Yes để chấp nhận khởi động lại máy.

Ví dụ: Bạn cài font Vietware, các bước tiến hành như sau:

1. Bạn chọn tập tin **Setup.exe** trong chương trình nguồn, thì có một cửa sổ xuất hiện như sau:



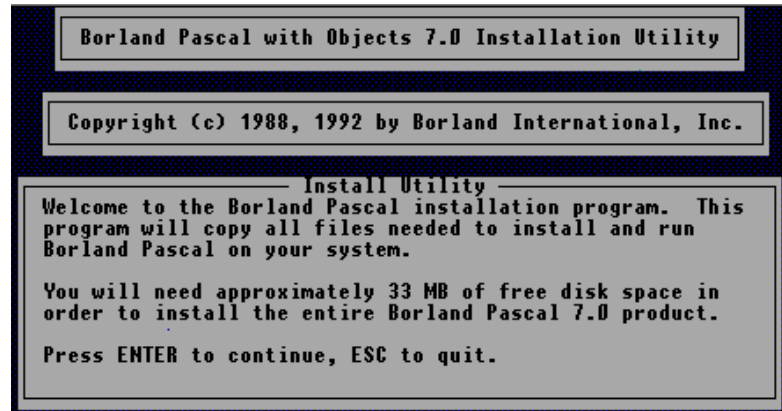
2. Bạn gõ mã số đăng ký vào hộp Serial number và sau khi bạn gõ xong bạn chọn OK xuất hiện cửa sổ



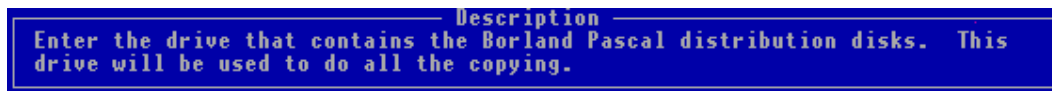
3. Tiếp theo bạn chọn chế độ cài đặt là: Full, Custom, Minimum. Sau khi bạn chọn lựa xong click vào install để tiếp tục. Và thư mục ngầm định được cài đặt là C:\VW20 nếu không muốn bạn có thể gõ lại đường dẫn. Và bạn chọn Continue, bạn chờ khoảng vài phút sẽ xuất hiện hộp thoại
4. Nếu bạn muốn khởi động lại máy thì click Restart Windows, còn nếu bạn không muốn khởi động lại thì click vào Continue để hoàn thành việc cài đặt.

2. Cài các phần mềm khác

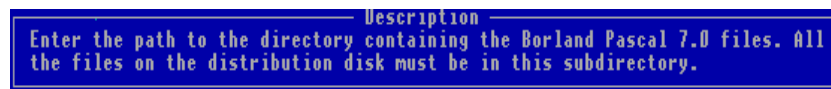
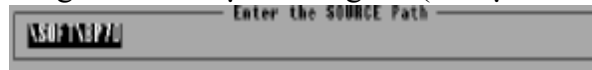
- Cài Turbo Pascal 7.0: Cách thức cài đặt cũng khá đơn giản chỉ cần có bộ nguồn của phần mềm (có thể chứa trên CDROM hay đĩa cứng). Bạn chọn tập tin **install.exe** . Sau đó bạn ấn phím ENTER để tiếp tục, bạn sẽ thấy xuất hiện màn hình dưới đây:



Bạn gõ tên ổ đĩa chứa những tập tin nguồn (là A nếu cài từ đĩa mềm hay C,D... nếu từ đĩa cứng)



- Sau đó là gõ tên thư mục chứa nguồn (Ví dụ: \SOFT\BP70).



- Dùng phím mũi tên lên xuống để chọn mục Borland Pascal directory và sau đó ấn phím Enter để đổi lại đường dẫn của thư mục nếu như bạn muốn (giả sử là C:\TURBO).



- Sau cùng là dùng phím mũi tên đưa con trỏ đến mục Start installation rồi ấn Enter để cài đặt Turbo Pascal vào máy.

- **Tóm lại:** Các phần mềm khác bạn cũng cài đặt tương tự như cách cài các phần mềm ở trên vậy.

PHỤ LỤC 1

CÁC MÃ LỖI

Sự cố bản mạch chính (101 - 109)

- **101 System interrupt Failed:** Sự cố này có thể là một vấn đề tranzito(hở mạch) không thường xuyên xảy ra hoặc board bổ sung đang xâm phạm tới chip điều khiển tín hiệu ngắt. Nếu bạn không thể vượt qua được mã 101, sẽ phải thay bản mạch chính.
- **102 System Timer Failed:** Chip bộ định thời trên bản mạch chính bị hư, phải thay bản mạch chính nếu lỗi thường xuyên xuất hiện.
- **103 System Timer interrupt Failed:** Chip bộ định thời không có chip điều khiển tín hiệu ngắt để truyền tín hiệu ngắt zero (tính ngắt định thời).
- **104 Protecd Mode Operation Failed:** Mã lỗi này chỉ áp dụng cho máy tính AT.
- **105 8042 Command Not Accepted.Keyboard Communication Failed:** Bị một chip điều khiển bàn phím 8042 hoặc bàn phím kém chất lượng.
- **106 Post logic test Problem Logic Test Failed:** Lỗi này có thể gây ra do board hệ thống bị hư hoặc các yếu tố khác như các thẻ mạch không chính xác cũng có thể gây ra lỗi.
- **107 MNI Test Failed:** Kiểm tra "ngắt không che được " của bản mạch chính bị sự cố , một NMI là tín hiệu ngắt không thể được vô hiệu hoá bằng một tín hiệu khác. Nếu lỗi này vẫn còn, phải thay bộ xử lý.
- **108 Failed System Timer Test:** Chip bộ định thời trên bản mạch chính không làm việc.
- **109 Probem With First 64K Ram, DMA Test Error:** Mã này chỉ một vấn đề trong RAM 64K đầu tiên trong các PC ban đầu hoàn toàn là khả năng của bản mạch chính. Có thể tìm thay các chip không chính xác hoặc thay bản mạch chính.

Các mã lỗi ps/2

- **110 PS/2 System Board Error, Parity chek:** Lỗi Board hệ thống, Kiểm tra chẵn lẻ.
- **111 PS/2 Memry Adapter Error:** Lỗi bộ phối hợp bộ nhớ.

- 112 PS/2 Microchannel arbitration Error ,System Board: Lỗi phân xử lý kênh, Board hệ thống.
- 113 PS/2 Microchannel arbitration Error ,System Board: Lỗi phân xử lý kênh, Board hệ thống.
- 165 PS/2 System option not test: Các tùy chọn hệ thống không được đặt.
- 166 PS/2 Microchannel adapter timeout Error: Lỗi thời gian không tính bộ phối hợp vi kênh.
- 199 PS/2 Configuration not correct. Check Setup: Cấu hình không chính xác. Kiểm tra cài đặt.

Các mã lỗi IBM

Các mã lỗi này được sử dụng trong một số máy IBM và một số máy nhái đã được đặt tương tự.

- 115 System Board ,CPU Error: Bản mạch hệ thống, lỗi CPU
- 118 System Board memory Error: Lỗi bộ nhớ Board hệ thống.
- 119 2,88MB diskette drive installed but not supported: Ổ đĩa mềm 2,88MB được cài đặt , nhưng không hỗ trợ.
- 120 System Board processor, cache (bộ nhớ truy cập nhanh) Error: Lỗi bộ nhớ truy cập nhanh ,bộ vi xử lý hệ thống.
- 121 Unexpected hardware interrupts occurred: Các tín hiệu ngắt phần cứng bất ngờ xảy ra.
- 130 POST-no operation System ,check diskettes, configuration: Hệ điều hành không có POST, kiểm tra các đĩa mềm, cấu hình.
- 131 Cassette interface test Failed , PS/2 System Board: Giao diện cassette bị sự cố Board hệ thống PS/2.
- 132 DMA (direct memory access- truy cập bộ nhớ trực tiếp) extended registers Error . Run diagnostics: Lỗi các thanh ghi bổ sung DMA. Chạy chương trình chẩn đoán.
- 133 DMA (direct memory access - truy cập bộ nhớ trực tiếp) Error . Run diagnostics:Lỗi DMA. Chạy chương trình chẩn đoán.

Các mã lỗi tổng quát

- **162 system option not set, or Possible Bad Battery:** Tùy chọn hệ thống không được cài đặt, hoặc pin có thể không chất lượng.
- **162 system option not set, or invalid Checksum, or Configuration incorrect:** Tùy chọn hệ thống không được cài đặt, hoặc tổng kiểm tra không giá trị, hay cấu hình không chính xác.
- **163 Time and Date Not Set:** Thời gian và ngày tháng không được cài đặt.
- **106 Memory Size Error:** Có vấn đề liên quan đến bộ nhớ CMOS.
- **201 Memory Error:** Lỗi Ram.
- **202 Memory Address Error Lines 0-15, 203 Memory Address Error 16-23:** Chỉ một hoặc nhiều chip bộ nhớ bị hư.
- **301 Keyboard Error:** Lỗi đối với bàn phím.
- **302 System Unit Keylock Is Locked:** Bộ chuyển mạch khoá phím bị lỗi hoặc bàn phím bị liệt.
- **303 Keyboard ở System Unit Error , 304 Keyboard ở System Unit Error , Keyboard Clockline Error:** Kiểm tra các phím bị liệt ,cáp nối bàn phím hoặc chính bàn phím bị hư.
- **601 Disk Error:** Chỉ vấn đề đĩa có thể do máy tính đó tìm một ổ đĩa mềm không có.
- **602 Disk boot Record Error:** Có thể do đĩa mềm bị hư hoặc một bộ điều khiển đĩa mềm bị hư.
- **1701 Hard Disk Failure:** Chỉ bộ điều khiển đĩa cứng không nhận được trả lời của đĩa cứng mà nó đang chờ.
- **1780 Disk 0 Failure , 1790 Disk 0 Error , 1781 Disk 1 Failure , 1791 Disk 1 Error:** Bộ điều khiển đĩa cứng không nhận được trả lời của từ đĩa cứng 0 hoặc 1.
- **1782 Disk Controller Failure:** Bộ điều khiển đĩa có thể bị hư.

PHỤ LỤC 2

THÔNG BÁO LỖI

- **128 NOT OK, Parity Disa(Industry Atandard Architect-kiến trúc tiêu chuẩn công nghệ) Bled:** 128 không được, chẵn lẻ bị vô hiệu hoá.
- **8042 Gate - A20 Error:** (cửa8042-lỗi A20) thường do bàn phím bị hư.
- **Access Denied:** truy cập bị từ chối.
- **Address Line Short!:** Điều có thể là vấn đề của chip bộ nhớ và cũng có thể do bản mạch chính và phải thay.
- **Allocation Error ,Size Adjusted:** Lỗi phân phối, kích thước bị điều chỉnh.
- **Attempted Write - Protect Violation:** thử định dạng 1 đĩa mềm chống ghi.
- **Bad DMA PORT:** Cổng truy cập bộ nhớ trực tiếp bị hư.
- **Bad Or Missing command interpreter:** Bộ dịch lệnh bị hư hoặc mất.
- **Bad Patition Table, Error Reading/Writing the Patition Table:** Bảng phân chia bị hư, lỗi đọc/viết bảng phân chia.
- **Nnnk Base Memory , Base Memory Size=nnk:** Bộ nhớ cơ sở Nnnk, kích thước bộ nhớ cơ sở = nnK.
- **Bus timeOut NMI At Slot X:** NMI không định thời gian Bus tại khe X.
- **C: Drive Error , Disk: Drive Error:** Lỗi ổ đĩa C, D. Ổ đĩa C hoặc D không được cài đặt chính xác trong CMOS.
- **C: Drive Failure ,D: Drive Failure:** Sự cố ổ đĩa C hoặc D .
- **Cache Memory bad , do Not Enable Cache:** Bộ nhớ truy cập nhanh trên bản mạch chính bị lỗi.
- **Cannot Chdir to (phatname).Tree past this point not processed:** Không thể kiểm tra thư mục tới (tên đường dẫn). Cây qua điểm này không xử lý được.Một trong các tập tin của thư mục đã bị rác (lỗi).
- **Cannot chdir to Root:** Không thể kiểm tra thư mục tới thư mục gốc. (Thư mục gốc đã bị rác).
- **Cannot Recover (.) Entry Processing Continue:** Không thể phục hồi(.) Xử lý tiếp tục.
- **Cannot Recover (..) Entry Processing Continue:** Không thể phục hồi(..) Xử lý tiếp tục.

- **Cannot Recover (..) Entry, Entry Has a bad attribute (or link or size):**
Không thể phục hồi(..) nhập, nội dung có thuộc tính (hoặc liên kết hoặc kích thước) bị hư.
- **CMOS barrety state low:** tình trạng pin cmos yếu (thay pin đồng hồ Cmos).
- **CMOS checksum Failure:** Sự cố kiểm tra tổng quát CMOS.
- **CMOS display type mismatch:** Không thích hợp loại màn hình CMOS.
- **CMOS Memory size mismatch:** Không thích hợp kích thước bộ nhớ CMOS.
- **CMOS System Options not set:** Các tùy chọn hệ thống CMOS không được cài đặt.
- **CMOS Time & Date not Set:** Thời gian và ngày tháng CMOS không được cài đặt.
- **COM port does not Exit:** Cổng COM không có.
- **Configuration Error For Slot n:** Lỗi cấu hình đối với khe n.
- **Convert Directory to file?:** Có chuyển đổi thư mục thành tệp không.
- **Convert Lost Chains to files(Y/N)?:** Chuyển đổi móc nối bị mất thành tệp (C/K) ?.
- **Error Found, F Parameter Not Specified:**Phát hiện lỗi ,Tham số F không rõ
- sửa lỗi sẽ không được ghi vào đĩa.
- **Disk Bad:** Đĩa hư.
- **Disk Boot Error , Replace and strike Key to retry:** Lỗi khởi động đĩa, thay và gõ phím để thử lại.
- **Disk configuration Error:** Lỗi cấu hình đĩa.
- **Hard Disk configuration Error:** Lỗi cấu hình đĩa cứng.
- **Disk Boot Failure:** Sự cố khởi động đĩa.
- **Disk Drive Failure:** sự cố ổ đĩa.
- **Diskette Drive X Failure:** Sự cố ổ đĩa mềm.
- **Diskette Read Failure:** sự cố đọc đĩa mềm.
- **DMA (Direct Memory Access) Error:** Lỗi truy cập bộ nhớ trực tiếp.
- **Drive not ready. Abort, Retry, Ignore, Fail ?:**ổ đĩa không sẵn sàng. Huỷ, thử lại, bỏ qua, hư ?
- **FDD controller Failure:** Sự cố bộ điều khiển ổ đĩa mềm.
- **FDD A is not installed:** ổ đĩa mềm A không được cài đặt.

- **File allocation table bad:** Bảng phân phối tệp hư.
- **Fixed disk configuration error:** Lỗi cấu hình đĩa cố định.
- **Fixed disk controller Failure:** Sự cố bộ điều khiển đĩa cố định.
- **Fixed disk Failure:** Chỉ bộ điều khiển đĩa cứng không nhận được trả lời của đĩa cứng mà nó đang chờ.
- **Hard Disk Failure:** Sự cố đĩa cứng.
- **Invalid boot diskette:** Đĩa mềm khởi động không hợp lệ.
- **Invalid configuration information. Please run setup program:** Thông tin cấu hình không hợp lệ. Chạy chương trình cài đặt.
- **Keyboard bad:** bàn phím hư
- **Keyboard data line Failure:** Sự cố đường truyền dữ liệu của bàn phím.
- **Keyboard controller Failure:** Sự cố bộ điều khiển bàn phím
- **Keyboard Error:** Lỗi bàn phím
- **Non-system disk or disk error. Replace and Strike and key When Ready:** Không có đĩa hệ thống hoặc đĩa bị lỗi. Thay và gõ phím bất kỳ khi sẵn sàng.
- **Non-system disk or disk error.Press A key to continue:** Không có đĩa hệ thống hoặc đĩa bị lỗi. Ấn một phím để tiếp tục.
- **No a boot disk-strike F1 to retry boot:** Không có đĩa khởi động, gõ phím F1 để thử khởi động lại.
- **Real time clock Failure:** Đồng hồ thực hoặc pin hỗ trợ bị sự cố.
- **Track 0 bad - disk unsuable:** Đĩa hư không sử dụng được track 0. Lỗi này có thể xảy ra khi định dạng đĩa mềm 1.44MB ,hoặc đĩa mềm đó bị hư track 0. Nếu thông báo này trên đĩa cứng thì phải thay đĩa cứng.
- **Write protect error writing Drive X:** Chốt bảo vệ ổ đĩa có thể chưa mở.

PHỤ LỤC 2

CÁC MÃ LỖI BIP

1. CÁC MÃ AMI

- **Một 'bip'**: Sự cố làm tươi của DRAM. Nếu máy tính hiển thị thông tin tiêu chuẩn trên màn hình, bạn không gặp vấn đề gì; nếu có vấn đề trở ngại, máy tính sẽ thông báo lỗi trên màn hình.
- **Hai 'bip'**: Sự cố hệ mạch chẩn lẻ / lỗi chẩn lẻ.
- **Ba 'bip'**: Sự cố bộ nhớ 64K cơ sở
- **Bốn 'bip'**: Bộ hẹn thời hệ thống không hoạt động.
- **Năm 'bip'**: Sự cố bộ vi xử lý
- **Sáu 'bip'**: Sự cố của A20 / bộ điều khiển bàn phím 8042
- **Bảy 'bip'**: Lỗi ngoại lệ chế độ thực/ lỗi ngắt ngoại lệ bộ vi xử lý
- **Tám 'bip'**: Lỗi viết đọc bộ nhớ màn hình
- **Chín 'bip'**: Lỗi kiểm tra tổng quát ROM BIOS. Cho biết ROM BIOS bị hư.
- **Mười 'bip'**: Lỗi viết / đọc của thanh ghi bị CMOS đóng.
- **Mười một 'bip'**: Bộ nhớ cache bị hư - không hữu hiệu hoá được cache.
- **Không có các 'bip'**: Nếu không nghe thấy các 'bip' và không có hình ảnh trên màn hình, kiểm tra bộ nguồn bằng đồng hồ von. Kế đến, kiểm tra bản mạch chính nghi ngờ có kết nối lỏng ra không. Chip CPU, BIOS, sẽ gây ra cho bản mạch chính có vấn đề.

2. CÁC MÃ PHOENIX

Máy tính được cài BIOS Phoenix sử dụng một nhóm ba bộ 'Bip' được tách ra và ở đây ghi những mã này theo số tiếng 'Bip' liên tiếp, ví dụ:

1-1-3 nghĩa là 'Bip', ngưng, 'Bip', ngưng, 'Bip' 'Bip' 'Bip'.

Hơn nữa, còn có các mã đặc biệt sử dụng tiếng 'Bip' ngắn và 'Bip' kéo dài.

- **Một 'bip'**: điều này thường không có vấn đề gì, 'Bip' phát ra khi việc tự kiểm tra hoàn tất trước khi DOS được tải.
- **Hai 'Bip'**: Có thể cấu hình bị lỗi.

- **Một 'Bip' dài, một 'Bip':** Chỉ sự cố video. Kiểm tra các bộ cầu nhảy và các bộ chuyển mạch DIP trên thẻ mạch video hoặc bản mạch chính.
- **Một 'Bip' dài, một 'bip' ngắn, Một 'Bip' dài, một 'bip' ngắn:** Chỉ sự cố của bộ phối hợp video đơn sắc và màu. BIOS đã thử khởi tạo, nhưng cả hai đều lỗi và không hiển thị.
- **1-1-3 CMOS Write /read Failure:** Máy tính không đọc được cấu hình được lưu trong CMOS. Nếu lỗi vẫn tiếp tục, thay bản mạch chính.
- **1-1-4 Rom BIOS checksum Error:** Rom BIOS đã bị hư và phải thay.
- **1-2-1 Programmable interval timer Failure:** Chip bộ định thời trên bản mạch chính bị hư và bản mạch chính sẽ phải thay.
- **1-2-2 DMA initialization Failure:** Chip DMA có thể bị hư.
- **1-3-1 Ram refresh verification Failure:** Có thể các bộ nhớ chip bị hư, chip DMA bị hư hoặc các chip địa chỉ bộ nhớ trên bản mạch chính bị hư.
- **1-4-2 parity Failure first 64K or Ram:** Chip bộ nhớ bị hư, hoặc một trong các chip nhạy cảm với việc kiểm tra lỗi chẵn lẻ.
- **3-2-4 Keyboard controller test Failure:** Chip điều khiển bàn phím không đáp ứng các tín hiệu lúc khởi động.

**TRƯỜNG ĐẠI HỌC HÀNG HẢI
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN KỸ THUẬT MÁY TÍNH**



**BÀI GIẢNG MÔN HỌC
KIẾN TRÚC MÁY TÍNH
VÀ THIẾT BỊ NGOẠI VI**

Hải Phòng – 2009

MỤC LỤC

Chương I: GIỚI THIỆU CHUNG	
• LỊCH SỬ PHÁT TRIỂN VÀ PHÂN LOẠI	
1. Lịch sử phát triển	
2. Phân loại máy tính	
II. BIỂU DIỄN THÔNG TIN TRÊN MÁY TÍNH.....	
Hệ đếm	
3. Các loại mã.....	
4. Biểu diễn số nguyên theo mã nhị phân	
5. Biểu diễn số thực theo mã nhị phân.	
6. Biểu diễn các dạng thông tin khác	
III. CÁC LOẠI HÌNH MÁY TÍNH CÁ NHÂN.	
Chương II: BỘ XỬ LÝ TRUNG TÂM	
1. Tổ chức bộ xử lý	
2. Tổ chức thanh ghi.....	
3. Đơn vị số học và logic ALU (Arithmetic and logic unit).....	
4. Đơn vị điều khiển CU(Control Unit).....	
5. Một số mở rộng của vi xử lý máy tính cho đến ngày nay	
6. BUS	
Chương III: HỆ THỐNG NHỚ	
I. KHÁI QUÁT VỀ HỆ THỐNG NHỚ CỦA MÁY TÍNH	
II. PHÂN CẤP BỘ NHỚ	
III. BỘ NHỚ BÁN DẪN	
1. Các loại bộ nhớ bán dẫn.....	
2. Tổ chức bộ nhớ	
IV. CACHE MEMORY.....	
1. Nguyên tắc (principle).....	
2. Kỹ thuật ánh xạ bộ nhớ cache	
IV. QUẢN LÝ BỘ NHỚ	
1. Các kỹ thuật quản lý bộ nhớ	
2. Bộ nhớ ảo	
3. Sự phân đoạn.....	
V. KỸ THUẬT GIẢI MÃ ĐỊA CHỈ.....	
1. Cấu tạo một vi mạch nhớ.....	
2. Giải mã địa chỉ cho bộ nhớ	
3. Giải mã địa chỉ bằng các mạch NAND.....	
4. Giải mã dùng mạch giải mã kiểu 74LS138	
Chương IV: TẬP LỆNH VÀ CÁC MODE ĐỊA CHỈ	
1. Tập lệnh của CPU	
2. Các nhóm lệnh của CPU	
3. Hợp ngữ(Assembly)	
4. Các Mode địa chỉ.....	
Chương V: HỆ THỐNG VÀO RA	
I. GIỚI THIỆU CHUNG	
1. Các thiết bị ngoại vi	
2. Modul vào ra.....	
II. GHÉP NỐI MÁY TÍNH VỚI THIẾT BỊ NGOẠI VI	
1. Ghép nối nối tiếp.....	
2. Ghép nối song song.....	

III. CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN VÀO RA	
1. Vào ra điều khiển bằng cách thăm dò.....	
2. Vào ra điều khiển bằng Ngắt	
3. Vào ra điều khiển bằng DMA	
4. Khối điều khiển DMAC.....	
Chương VI: MÀN HÌNH	
I. Những khái niệm cơ bản	
1. Nguyên lý của phương pháp hiển thị hình ảnh video.....	
2. Những đặc điểm chung của màn hình	
II. Màn hình màu CRT (Cathod Ray Tube)	
1. Cấu tạo	
2. Phương pháp quét dòng	
Chương VII: BÀN PHÍM.....	
1. Khái niệm	
2. Kỹ thuật dò phím	
Chương VIII: Ổ ĐĨA	
1. Đĩa từ (Magetic).....	
2. Đĩa Quang (Optical Disk).....	
Chương IX: THIẾT BỊ GHÉP NỐI VÀ TRUYỀN THÔNG	
1. Bộ chuyển đổi tín hiệu	
2. Modem (Modulation - Demodulation).....	
3. Các chuẩn giao tiếp	
TÀI LIỆU THAM KHẢO	

Chương I: GIỚI THIỆU CHUNG

• LỊCH SỬ PHÁT TRIỂN VÀ PHÂN LOẠI

1. Lịch sử phát triển

Nhiều thế hệ trôi qua con người đã thực hiện các phép toán với các con số chủ yếu bằng tay hay bằng các công cụ tính thô sơ (bảng tính, thước tính ...).

Năm 1943, John Mauchley và các học trò của ông đã chế tạo ra chiếc máy tính điện tử đầu tiên ở Mỹ - chiếc máy tính được đặt tên là ENIAC (Electronic Numerical Integrator And Calculator). Nó gồm 18.000 đèn điện tử, 1500 rơ le, nặng 30 tấn, tiêu thụ công suất điện 140KW. Chiếc máy này mục đích phục vụ quân đội trong chiến tranh thế giới lần thứ 2 nhưng đến năm 1946 nó mới hoàn thành.

Cho đến ngày nay máy tính đã có những sự phát triển vượt bậc, ứng dụng trong hầu hết các hoạt động của xã hội với rất nhiều chủng loại thế hệ tùy theo công việc. Tuy nhiên kể từ đó đến nay có thể phân máy tính ra thành các thế hệ sau:

Thế hệ 1: (1950-1959):

- Về kỹ thuật: linh kiện dùng đèn điện tử, độ tin cậy thấp, tổn hao năng lượng. Tốc độ tính toán từ vài nghìn đến vài trăm nghìn phép tính/giây.
- Về phần mềm: chủ yếu dùng ngôn ngữ máy để lập trình.
- Về ứng dụng: mục đích nghiên cứu khoa học kỹ thuật.

Thế hệ 2: (1959-1964):

- Về kỹ thuật: linh kiện bán dẫn chủ yếu là transistor. Bộ nhớ có dung lượng khá lớn.
- Về phần mềm: đã bắt đầu sử dụng một số ngôn ngữ lập trình bậc cao: Fortran, Algol, Cobol, ...
- Về ứng dụng: tham gia giải các bài toán kinh tế xã hội.

Thế hệ 3 (1964-1974)

- Về kỹ thuật: linh kiện chủ yếu sử dụng các mạch tích hợp (IC), các thiết bị ngoại vi được cải tiến, đĩa từ được sử dụng rộng rãi. Tốc độ tính toán đạt vài triệu phép toán trên giây; dung lượng bộ nhớ đạt vài MB (Megabytes).
- Về phần mềm: Xuất hiện nhiều hệ điều hành khác nhau. Xử lý song song. Phần mềm đa dạng, chất lượng cao, cho phép khai thác máy tính theo nhiều chế độ khác nhau.
- Về ứng dụng: tham gia trong nhiều lĩnh vực của xã hội.

Thế hệ thứ 4 (1974-199?):

- Về kỹ thuật: Xử dụng mạch tích hợp cỡ lớn (Very large scale integration) VLSI, thiết kế các cấu trúc đa xử lý. Tốc độ đạt tới hàng chục triệu phép tính /giây.

Ở đây chúng ta chủ yếu nói về cấu trúc máy vi tính tương thích IBM nên lịch sử của chiếc máy PC gắn liền với sự phát triển của IBM-PC. Chiếc máy tính cá nhân đã phát triển cùng với sự phát triển của các bộ vi xử lý.

Máy IBM_PC coi như được khởi đầu từ một công trình của phòng thí nghiệm tại Atlanta của IBM.

- Từ năm 1979-1980 IBM hoàn thành chiếc máy Datamaster. Máy này dùng vi xử lý 16 bit của Intel.

- Năm 1980 kế hoạch sản xuất máy PC bắt đầu được thực hiện. Chiếc máy IBM_PC đầu tiên dùng một bộ vi xử lý 8 bits của Intel, bộ VXL 8085.
 - Năm 1981-1982 IBM sản xuất máy tính PC sử dụng bộ vi xử lý 8086,8088.
 - Năm 1984 máy tính xử dụng chip 80286.
 - Năm 1987 máy tính xử dụng bộ VXL 32bits 80386.
 - Năm 1990 bộ VXL 80486 ra đời với nhiều tính năng hơn.
 - Năm 1993 Bộ VXL Pentium ra đời mở ra một thế hệ vi tính cá nhân mới với 64 bits dữ liệu, 32 bit địa chỉ.
 - 1995-1999 các thế hệ VXL mới như MMX,Pentium II,III với khả năng biểu diễn không gian 3 chiều, nhận dạng tiếng nói...
 - Từ năm 2000 cùng với Merced một thế hệ VXL 64 bit với cấu trúc hoàn toàn mới ra đời đã tạo ra một thế hệ máy vi tính mới.
- Về ứng dụng : Máy tính đã được áp dụng trong hầu hết các lĩnh vực của xã hội.

Thế hệ thứ 5: Theo đề án của người Nhật chiếc máy tính điện tử thế hệ thứ 5 có cấu trúc hoàn toàn mới, bao gồm 4 khối cơ bản. Một trong các khối cơ bản là máy tính điện tử có cấu trúc như hiện nay và liên hệ trực tiếp với người sử dụng thông qua khối giao tiếp trí thức gồm 3 khối con: bộ xử lý giao tiếp, cơ sở tri thức và khối lập trình.

2. Phân loại máy tính

Máy tính (computer) là một khái niệm tương đối rộng, tùy theo cấu trúc, chức năng, hình dáng... mà có thể phân ra nhiều loại khác nhau. Về căn bản máy tính được phân làm các loại chính sau:

a. Phân loại theo khả năng

- Máy tính lớn (mainframe computer)
- Máy tính con (mini computer)
- máy vi tính (Microcomputer).

Máy tính lớn (mainframe computer): có khả năng giải những bài toán lớn tốc độ tính toán nhanh. Chúng được thiết kế đặc biệt với chiều dài bus dữ liệu rộng 64 bit hoặc hơn. Kích thước bộ nhớ làm việc rất lớn. Giá thành cao chỉ được chuyên dùng cho các ứng dụng trong quân sự, ngân hàng, khí tượng. Máy tính lớn được dùng trong khoa học để mô phỏng nghiên cứu các hiện tượng vật lý như các vụ nổ hạt nhân. Ví dụ như máy tính lớn IBM 4381, Honeywell DSP8, hay Deepblue...

Máy tính con (mini computer) là một dạng thu nhỏ của máy tính lớn. Chiều rộng dữ liệu vào khoảng 32 bit đến 64 bit. Do giá thành thấp hơn máy tính lớn, tính năng mạnh nên máy tính con rất được ưa dùng trong nghiên cứu khoa học.

Máy vi tính (MicroComputer): Những máy dùng bộ vi xử lý (họ Intel, Motorola) làm cốt lõi, vi điều khiển (microcontroller) và máy tính trong một vi mạch (one-chip microcomputer) đều thuộc họ máy vi tính. Đặc điểm chung về công nghệ của họ này mức độ tổ hợp lớn VLSI (very large scale integration) và dùng công nghệ CMOS (complementary metal oxide silicon) để chế tạo các mạch logic. Tốc độ phát triển các vi xử lý 32 bit và 64 bit hiện đại làm khoảng cách giữa máy tính lớn và máy vi tính ngày càng thu hẹp.

Trạm làm việc (workstation) cũng là một loại máy vi tính, đặc điểm khác biệt so với máy tính cá nhân PC là có khả năng được nhiều người cùng xử dụng cùng một lúc.

Máy tính cá nhân PC (Personal Computer) chỉ được một người sử dụng. Giá thành của chúng rẻ do cấu hình đơn giản, được chuẩn hoá, và được sản xuất hàng loạt với số lượng lớn. Cùng với sự phát triển của khoa học công nghệ mà máy tính cá nhân ngày nay đã có thể làm được những công việc mà trước kia vốn chỉ là đặc quyền của máy tính lớn.

b. Phân loại theo nguyên lý

- Máy tính cơ khí.
- Máy tính tương tự

- Máy tính số

c. Phân loại theo kiến trúc

*Kiến trúc tuần tự (kiến trúc VonNewman cổ điển)

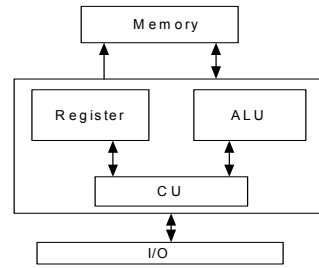
Máy tính gồm CPU, Memory, I/O.

- ✓ CPU gồm:
- ✓ thanh ghi (register)
- ✓ ALU (Arithmetic Logical Unit)
- ✓ CU (Control Unit).

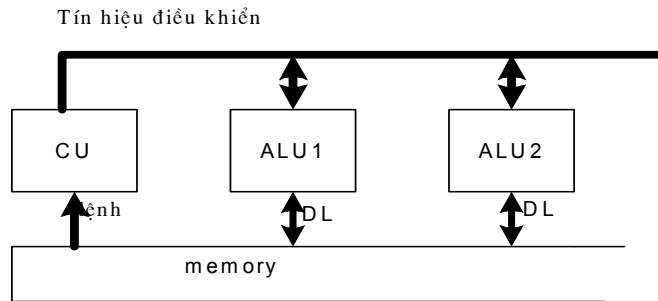
Đặc điểm :

- ✓ Thực hiện lần lượt từng lệnh một
- ✓ Tốc độ chậm

Còn được gọi là kiến trúc SISD(Single Instruction Stream-Single Data Stream)



*Kiến trúc song song

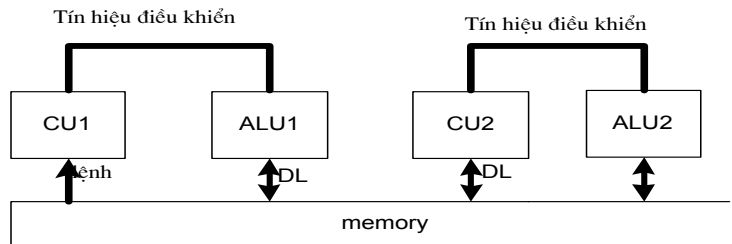


+SIMD(Single Instruction Stream-Multiple Data Stream)

Đặc điểm: Có một đơn vị điều khiển, n phần tử xử lý

Đơn vị điều khiển: điều khiển đồng thời tất cả các phần tử tại cùng một thời điểm các phần tử xử lý thực hiện cùng một thao tác trên các tập dữ liệu khác nhau.

+ MIMD (Multiple Instruction Stream-Multiple Data Stream)



Đặc điểm:

Máy gồm hai hoặc nhiều bộ vi xử lý tương tự về khả năng, có thể thực hiện những phép toán khác nhau trên các DL khác nhau.

Tất cả các CPU cùng chia sẻ một bộ nhớ chung. Một số bộ nhớ cục bộ cũng có thể được dùng.

Tất cả CPU cùng xử dụng chung các thiết bị vào ra, có thể dùng chung một số kênh hoặc dùng các kênh khác nhau dẫn đến cùng một thiết bị.

Hệ thống được điều khiển bởi hệ thống hoạt động tích hợp. Nó cung cấp sự ảnh hưởng lẫn nhau giữa CPU và các chương trình chạy trên chúng ở task, file, các thành phần dữ liệu...

+MISD(Multiple Instruction Stream-Single Data Stream)

Đặc điểm: Có nhiều đơn vị điều khiển

Thực hiện lệnh theo các công đoạn, tại một thời điểm mỗi đơn vị điều khiển xử lý một công đoạn như vậy có thể tiết kiệm được số chu kỳ máy cần để xử lý lệnh.

Đây cũng là nguyên tắc của Pipeling khi chia lệnh thành các công đoạn: nhận lệnh F(Fetch), giải mã lệnh D(Decode), thực hiện lệnh E(Execute), và ghi kết quả W(write back). Như vậy với 4 lệnh có thể tiết kiệm tới 9 chu kỳ máy(hình vẽ)

II. BIỂU DIỄN THÔNG TIN TRÊN MÁY TÍNH

Hệ đếm

a. Hệ đếm bất kỳ

Bất kỳ một hệ đếm nào đều biểu diễn một số nguyên theo nguyên tắc sau:

$$N = a_{n-1} \dots a_0 = a_0 \cdot s^0 + a_1 \cdot s^1 + \dots + a_{n-1} \cdot s^{n-1} = \sum_{i=0}^{n-1} a_i \cdot s^i \quad (1.1)$$

Trong đó N là một số nguyên có n chữ số. Chữ số a_i tại vị trí i ($i=0 \dots n-1$) được gọi là trị số (hay còn gọi là trọng số). Giá trị s là cơ số của hệ đếm. Hệ đếm được đặt tên theo giá trị cơ số s. Chẳng hạn, với $s=2$ ta có hệ đếm cơ số 2, với $s=10$ ta có hệ đếm cơ số 10 và với $s=16$ ta có hệ đếm 16. Giá trị s cũng xác định số ký tự cần dùng để biểu diễn trị số. Chẳng hạn với $s=2$ hệ đếm sẽ cần hai ký tự để biểu diễn, vì thế ta có khái niệm hệ nhị phân(chia ra làm hai). Tương tự như vậy, hệ đếm 10 và 16 còn được gọi là hệ thập phân và hệ thập lục phân.

b. Hệ đếm thập phân

Định nghĩa: là hệ đếm quen thuộc nhất của nhân loại. Có lẽ hệ đếm này bắt nguồn từ việc người tiền sử dùng mười đầu ngón tay để đếm các đồ vật xung quanh. Ngày nay toàn thế giới thống nhất sử dụng những ký tự số Ả Rập để biểu diễn hệ thập phân. Các ký tự số đó là: 0,1,2,3,4,5,6,7,8,9. Việc phát minh ra số 0 mới có khả năng biểu diễn số nguyên theo đúng nguyên tắc đã nêu trong phương trình (1.1).

Ngoài ra như chúng ta đã biết một số nền văn minh khác cũng phát minh ra hệ đếm của mình như Trung Quốc, La Mã cổ.... Tuy nhiên vì không có ký tự số 0 nên các hệ đếm này đều cần nhiều hơn 10 ký tự để biểu diễn số nguyên.

Ví dụ biểu diễn số nguyên:

$$N=1547D=1 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0.$$

c. Hệ đếm nhị phân

Được hình thành trên cơ sở đại số lô gic Boole, xuất hiện từ cuối thế kỷ 19. Hệ đếm này và các môn toán liên quan đến nó thực sự phát huy được sức mạnh khi có mạch điện hai trạng thái. Với hai con số 0,1 có thể biểu diễn một số nguyên bất kỳ. Mỗi ký tự (hay mỗi trị số) của hệ nhị phân được gọi là

một bit (binary digit). Đối với máy tính điện tử các bit được biểu diễn bằng một hiệu điện thế tương ứng: mức 0 (0V-1 V), mức 1 (2v-5v).

Để giản tiện trong việc sử dụng số nhị phân, người ta còn đặt nhiều bội số của hệ nhị phân như sau:

- 4 bit là một nibble.
- 8 bit là một byte.
- 16 bit là một từ (word).
- 32 bit là một từ kép (double word)
- 2^{10} bit là một kilobit (Kbit).
- 2^{20} bit là một Megabit (Mbit).
- 2^{30} bit là một Gigabit (Gbit).

Ví dụ biểu diễn một số nguyên:

$$N=1011B=1.2^3+0.2^2+1.2^1+1.2^0=8+0+2+1=11D$$

d. Hệ thập lục phân (hexa).

Xuất hiện như một cách biểu diễn giản tiện trong công nghệ tin học. Vì một số nhị phân quá dài và bất tiện khi viết và tính toán. 4 chữ số nhị phân được gộp thành một chữ số thập lục phân. Như vậy có số của hệ thập lục phân là $s=16$. Điều này có nghĩa là cần có 16 ký tự khác nhau để biểu diễn hệ thập lục phân. Các ký tự đó là :0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Ví dụ biểu diễn một số nguyên:

$$N=2BC1h=2.16^3+11.16^2+12.16^1+1.16^0=11201D.$$

Đổi số thập phân ra số nhị phân hoặc ngược lại

Để đổi số thập phân ra số nhị phân hay thập lục phân, ta chỉ cần chia số thập phân cho cơ số của hệ (2 hoặc 16). Số dư là trị số, thương số được chia tiếp để tính trị số tiếp theo. Trong hệ nhị phân, trị số đầu tiên (ngoài cùng bên phải) được gọi là LSB (least significant bit) và trị số cuối cùng (ngoài cùng bên trái) được gọi là MSB (most significant bit).

Ví dụ:

$$N=113D$$

Nhị phân	Thập lục phân
$113/2=56$ dư 1 LSB	$113/16=7$ dư 1
$56/2=28$ dư 0	$7/16=0$ dư 7
$28/2=14$ dư 0	
$14/2=7$ dư 0	
$7/2=3$ dư 1	
$3/2=1$ dư 1MSB	

$$\text{Kết quả}=113D=110001B=71H$$

3. Các loại mã

a. Mã BCD

Dùng 4 bit hệ 2 để biểu diễn một số hệ 10

b. Mã ASCII

Dùng 7 bit để mã hoá, bit cuối cùng là bit kiểm tra chẵn lẻ, phát hiện lỗi khi truyền

4. Biểu diễn số nguyên theo mã nhị phân

Dùng số nhị phân không dấu:

n bit biểu diễn 2^n số từ 0 đến 2^n-1

Dùng số nhị phân có dấu:

n bit biểu diễn 2^n số từ -2^{n-1} đến $+2^{n-1}-1$

Số bù 2:

Số bù 1: 1 đổi thành 0, 0 đổi thành 1

Số bù 2: số bù 1 cộng 1

5. Biểu diễn số thực theo mã nhị phân.

a. Biểu diễn dấu chấm cố định:

Cách biểu diễn dấu chấm cố định trong hệ nhị phân hoàn toàn giống cách biểu diễn số thực thông thường của hệ thập phân

$$R = a_{n-1} \dots a_0, b_0 \dots b_{m-1} = \sum_{i=0}^{n-1} a_i \cdot s^i + \sum_{i=0}^{m-1} b_i \cdot s^{-i}$$

Trong đó R số thực cần biểu diễn gồm n trị số đứng trước và m trị số đứng sau dấu chấm. Tùy thuộc vào hệ thập phân hay nhị phân mà cơ số s có giá trị là 2 hay 10.

b. Biểu diễn dấu chấm động

Chia làm 4 thành phần:

M: phần định trị

- E: phần mũ
- R: cơ số
- S: dấu

Như vậy $X = (-1)^S \cdot M \cdot R^E$

Ví dụ: $R = -750 = -0,75 \cdot 10^3 = -0,75E3$

Để định dạng dấu chấm động có thể dùng chuẩn IEEE754-1985 (Institute of Electrical and Electronic Engineering) 32 bit hoặc 64 bit:

Đây là chuẩn được mọi hãng chấp nhận và được dùng trong bộ xử lý toán học của Intel. Bit dấu nằm ở vị trí cao nhất, kích thước phần mũ và khuôn dạng phần định trị thay đổi theo từng loại số thực

Giá trị số thực IEEE754-1985 được tính như sau:

$$R = (-1)^S \cdot (1 + M_1 \cdot 2^{-1} + \dots + M_n \cdot 2^{-n}) \cdot 2^{E_7 \dots E_0 - 127}$$

S	$E_7 - E_0$	Định trị ($M_1 - M_{23}$)
---	-------------	-----------------------------

Ví dụ:

428CE9FCH=0100 0010 1000 1100 1110 1001 1111 1100

Phần dấu(bit cao nhất): 0 = số dương

Phần mũ: $2^8 + 2^2 + 2^0 - 127 = 133 - 127 = 6$

Phần định trị: $2^{-4} + 2^{-5} + 2^{-8} + 2^{-10} + 2^{-12} + 2^{-15} + 2^{-16} + 2^{-17} + 2^{-18} + 2^{-19} + 2^{-20} + 2^{-21} = 0,1008906$
như vậy giá trị ngầm định là 1,1008906

Quy tắc đổi ngược lại:

Chuyển số dấu phẩy động về dạng nhị phân

Đưa về dạng 1.xxxxEyyyy

xác định bit 31: dấu

Xác định bit từ 30-23: yyyy+7Fh

Xác định bit 22-0: xxxx00..00

6. Biểu diễn các dạng thông tin khác

Biểu diễn hình ảnh

Biểu diễn âm thanh

Các đại lượng vật lý khác

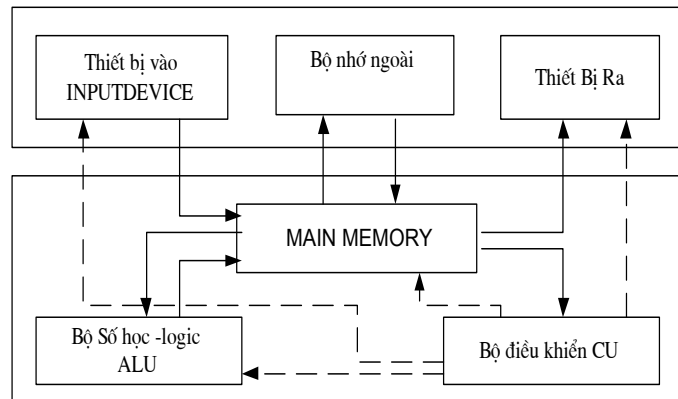
III. CÁC LOẠI HÌNH MÁY TÍNH CÁ NHÂN.

Để đảm bảo tính tương thích, cấu trúc phần cứng bên trong các máy vi tính cá nhân về cơ bản là giống nhau. Vì thế chúng chỉ được phân loại theo hình dạng vật lý.

- Loại để bàn(desktop), loại để bàn thu nhỏ (desktop slim-line)
- Loại đặt đứng (tower),mini-tower
- Loại xách tay (notebook).
- Loại bỏ túi (palmtop,palmpilot).

Kiến trúc chung của máy tính điện tử

1. Bộ nhớ trung tâm (Central Memory or Main Memory) Có nhiệm vụ chứa những chương trình và dữ liệu trước khi chương trình được thực thi
2. Bộ điều khiển (Control Unit -CU) Có nhiệm vụ điều khiển sự hoạt động của tất cả các thành phần của hệ thống máy tính theo chương trình mà nó được giao thi hành.
3. Bộ số học và logic (Arithmetic Logical Unit, thường được viết tắt là ALU) . Có nhiệm vụ thực hiện các thao tác tính toán theo sự điều khiển của CU.
4. Thiết bị vào (Input Device). Có nhiệm vụ nhận các thông tin từ thế giới bên ngoài, biến đổi sang dạng số một cách thích hợp rồi đưa vào bộ nhớ trong.
5. Thiết bị ra (Output Device) Có nhiệm vụ đưa thông tin số từ bộ nhớ trong ra ngoài dưới dạng những dạng mà con người yêu cầu.



Các đơn vị chức năng cơ bản của máy tính điện tử
(Các đường vẽ nét đứt chỉ mối quan hệ. Các đường nét liền là đường truyền dữ liệu)

Chương II: BỘ XỬ LÝ TRUNG TÂM

1. Tổ chức bộ xử lý

Để hiểu được tổ chức của CPU, chúng ta hãy xem xét những yêu cầu đặt ra trên CPU, những thứ nó phải làm:

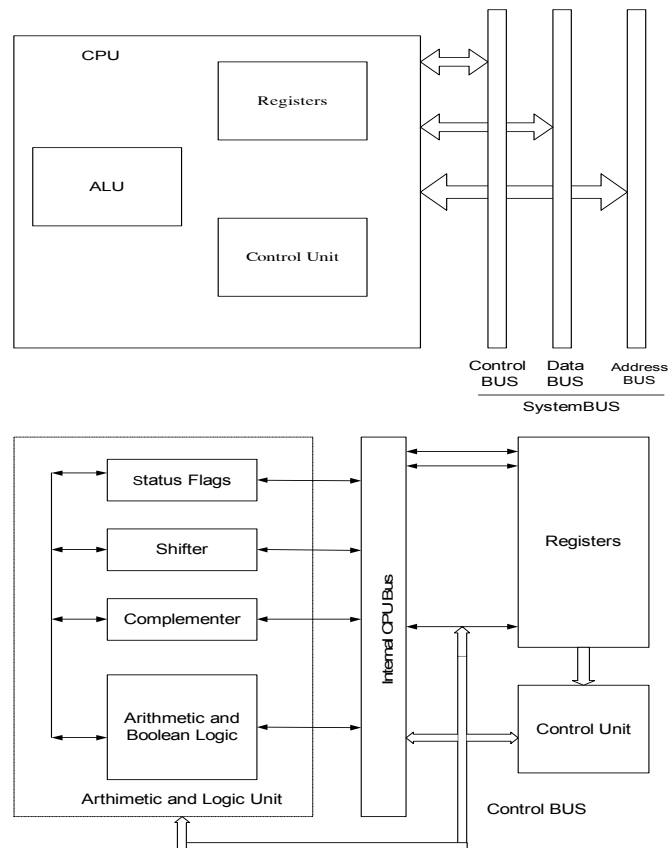
- Fetch Instructions(chỉ lệnh tìm nạp): CPU phải đọc các chỉ lệnh từ bộ nhớ.
- Interpret Instructions: chỉ lệnh phải được giải mã để xác định hành động nào được yêu cầu.
- Fetch data (dữ liệu tìm nạp): Sự thi hành một chỉ lệnh có thể yêu cầu thực hiện một vài thao tác số học hoặc logic trên dữ liệu.
- Write Data: Những kết quả của sự thi hành có thể yêu cầu viết dữ liệu vào bộ nhớ hoặc module vào ra.

Thông thường CPU phải thực hiện các công việc này, nó có thể rõ ràng CPU cần một chỗ chứa tạm thời để chứa dữ liệu. Nó phải nhớ vị trí của chỉ lệnh sau cùng đến mức nó có thể biết nơi nào chứa lệnh tiếp theo. Nó cần chứa các chỉ lệnh và dữ liệu tạm thời trong khi một lệnh được thực thi. Nói cách khác CPU cần một bộ nhớ trong nhỏ.

Hình 1.1 là 1 hình ảnh đơn giản của CPU, xác định các kết nối của nó với phần còn lại của hệ thống thông qua bus hệ thống. Một giao diện tương tự có thể cần đến cho tất cả các cấu trúc kết nối khác. Các bộ phận chính của một CPU là đơn vị số học và logic (ALU) và đơn vị điều khiển (CU). ALU làm công việc tính toán thực sự hoặc xử lý dữ liệu. Đơn vị điều khiển CU chuyển dữ liệu và các chỉ lệnh vào và ra khỏi CPU và điều khiển các thao tác của ALU. Thêm nữa, hình trên còn mô tả một bộ nhớ trong, là chỗ chứa tạm thời gọi là thanh ghi (Register)

Hình 1.2 mô tả hình ảnh chi tiết hơn cấu trúc CPU. Các đường truyền dữ liệu và các đường điều khiển logic đều được xác định, bao gồm một thành phần gắn nhãn internal CPU bus. Thành phần này được yêu cầu chuyển dữ liệu giữa các thanh ghi khác nhau và ALU, từ ALU thực tế hoạt động chỉ trên dữ liệu trong bộ nhớ trong CPU. Hình vẽ cũng mô tả các thành phần cơ

bản tiêu biểu của ALU. Chú ý sự tương tự giữa cấu trúc trong của máy tính và cấu trúc trong của CPU. Trong cả hai trường hợp, có một sự tập hợp của các thành phần chính (computer: CPU, I/O, bộ nhớ; CPU: CU,ALU, các thanh ghi) được kết nối bằng các đường dữ liệu.



2. Tổ chức thanh ghi

Một hệ thống máy tính dùng một hệ thống cấp bậc bộ nhớ. Tại các mức cao hơn trong hệ thống cấp bậc, bộ nhớ nhanh hơn, nhỏ hơn, và đắt hơn (tính theo bit). Trong CPU, có tập hợp các thanh ghi chức năng là mức nhớ trên bộ nhớ chính và bộ nhớ cache trong hệ thống cấp bậc. Các thanh ghi trong CPU phục vụ 2 chức năng chính:

- **User-Visible Registers**: Nó cho phép người lập trình ngôn ngữ máy hoặc ngôn ngữ Assembly thu nhỏ bộ nhớ chính bằng tối ưu hoá việc sử dụng các thanh ghi.
- **Control and Status Registers**: Các thanh ghi này được sử dụng bởi đơn vị điều khiển CU để điều khiển các thao tác của CPU và bằng phân quyền, các chương trình điều khiển hệ thống điều khiển sự thực thi của các chương trình khác.

Không có sự riêng biệt rõ ràng giữa các thanh ghi trong hai loại trên. Ví dụ trên một số máy chương trình đếm là thanh ghi user-visible (ví dụ VAX) nhưng trên nhiều máy khác lại không phải vậy. Cho các mục đích sẽ được thảo luận dưới đây, chúng ta sẽ sử dụng hai loại này

User-Visible Registers:

Thanh ghi User-Visible là một trong những thành phần được tham chiếu bởi cách thức của ngôn ngữ máy được CPU thi hành. Thực sự tất cả các thiết kế CPU đương thời cung cấp một số các thanh ghi User-Visible đối lập với một thanh ghi tổng đơn giản. Chúng ta có thể mô tả đặc điểm của chúng trong các loại sau:

- ✓ Mục đích chung
- ✓ Dữ liệu
- ✓ Địa chỉ
- ✓ Mã điều kiện

Các thanh ghi mục đích chung (general-purpose registers) có thể bị phân chia cho các chức năng khác nhau bởi người lập trình. Thông thường, chúng sử dụng trong tập lệnh trực giao với thao tác. Đó là, bất cứ một thanh ghi mục đích chung nào có thể chứa đựng toán hạng cho opcode. Nó cung cấp sử dụng thanh ghi mục đích chung thực sự. Thông thường, có các giới hạn ví dụ có thể có các thanh ghi cho các thao tác con trỏ động.

Trong một số trường hợp các thanh ghi mục đích chung có thể được dùng cho các chức năng đặc chỉ hoá (ví dụ thanh ghi gián tiếp, dịch chuyển). Trong các trường hợp khác, có một phần hoặc sự phân chia rõ ràng giữa thanh ghi dữ liệu và thanh ghi địa chỉ. Các thanh ghi dữ liệu có thể được sử dụng chỉ để giữ dữ liệu và không thể được dùng trong việc tính toán của một địa chỉ toán hạng. Các thanh ghi địa chỉ có thể tự bản thân là thanh ghi mục đích chung, hoặc nó có thể được dành hết cho chế độ địa chỉ riêng.

- ✓ **Con trỏ đoạn**: Trong một máy với phương pháp địa chỉ đoạn, một thanh ghi đoạn giữ địa chỉ cơ sở của đoạn. Có thể có nhiều thanh ghi: ví dụ, một cho hệ thống điều khiển và một cho tiến trình hiện tại.
- ✓ **Thanh ghi chỉ số**: Được dùng trong chế độ địa chỉ chỉ số và có thể được tự động đánh chỉ số.
- ✓ **Con trỏ ngăn xếp**: Nếu có user-visible stack addressing, sau đó ngăn xếp tiêu biểu là trong bộ nhớ và có một thanh ghi chỉ đến đầu ngăn xếp. Nó cho phép đánh địa chỉ tuyệt đối; đó là push, pop, và các chỉ lệnh ngăn xếp khác cần không chứa một toán hạng ngăn xếp rõ ràng.

Control and Status Registers:

Có rất nhiều thanh ghi CPU khác nhau được sử dụng để điều khiển thao tác của CPU. Hầu hết chúng trên đa số máy là không hữu hình với người dùng. Một vài thanh ghi có thể hữu hình với các lệnh máy thực thi trong chế độ điều khiển hoặc trong operating-system mode.

Tất nhiên, các máy khác nhau sẽ có tổ chức thanh ghi khác nhau và sử dụng thuật ngữ khác nhau. Chúng tôi liệt kê ra đây một danh sách hoàn toàn hợp lý của các loại thanh ghi với một mô tả ngắn gọn:

Bốn thanh ghi là cốt tuỷ đối với sự thi hành lệnh.

- Program Counter(PC): chứa địa chỉ của một chỉ lệnh được tìm nạp.
- Thanh ghi lệnh (Instruction Register): chứa chỉ lệnh được tìm nạp gần nhất.
- Thanh ghi địa chỉ bộ nhớ(Memory Address Register): chứa địa chỉ của các vị trí trong bộ nhớ.
- Thanh ghi bộ nhớ đệm (Memory Buffer Register): chứa một từ dữ liệu được ghi vào trong bộ nhớ hoặc từ được đọc gần đây nhất.

Ví dụ các tổ chức thanh ghi vi xử lý.

Các ví dụ cung cấp tài liệu để nghiên cứu và so sánh tổ chức thanh ghi của các hệ thống có thể so sánh được. Trong phần này, chúng ta sẽ xem xét 3 bộ vi xử lý 16 bit được thiết kế ở cùng một thời điểm: Zilog Z8000(PEUT79), Intel 8086 [MORS78,HEYW83], Và Motorola MC6800 [STR179].

Ví dụ với bộ xử lý 8086:

Bao gồm:

- 1 thanh ghi con trỏ lệnh IP (instruction Pointer): Lưu trữ địa chỉ lệnh kế tiếp sẽ được chạy trong đoạn CT hiện thời. Mỗi 1 từ lệnh được đọc từ bộ nhớ BIU sẽ thay đổi giá trị IP sao cho nó chỉ đến địa chỉ của từ lệnh kế tiếp trong bộ nhớ.
- 8 thanh ghi chung
- 4 thanh ghi dữ liệu AX,BX, CX, DX.
 - AX: (Accumulator Register) thanh ghi tích lũy các kết quả tính toán.
 - BX (Base Register) thanh ghi cơ sở: chỉ địa chỉ cơ sở của vùng nhớ thuộc bộ nhớ.
 - CX (Counter Register) thanh ghi đếm: Khai báo số lần 1 thao tác nào đó phải được thực hiện trong các vòng lặp, phép dịch, quay.
 - DX (Data Register) thanh ghi số liệu: lưu trữ si làm thông số chuyển giao CT (2 byte).

Khi cần truy nhập chỉ với 1 byte thì byte cao hay thấp được nhận diện H,L.

*Các thanh ghi con trỏ, chỉ số:

- SP (Stack pointer) con trỏ ngăn xếp: địa chỉ đỉnh ngăn xếp. SP cho phép truy xuất dễ dàng các địa chỉ trong đoạn ngăn xếp SS (stack segment). Giá trị trong SP mô tả phải offset của địa chỉ ngăn xếp kế tiếp so với địa chỉ hiện tại đang được lưu trong SS.
- BP (Base pointer) con trỏ cơ sở: mô tả offset tính từ SS nhưng còn được sử dụng truy nhập DL trong SS.
- I (index) thanh ghi chỉ số: lưu địa chỉ offset đối với những lệnh truy nhập DL cất trong đoạn DL

*Thanh ghi đoạn:

Bộ nhớ được chia thành các đoạn logic (segment) dài 64kb. CPU có thể truy nhập 1 lần tới 4 đoạn.

Địa chỉ đoạn chứa trong thanh ghi đoạn.

- Thanh ghi đoạn mã CS (code Segment) nhận diện ĐC bắt đầu của đoạn chương trình hiện hành trong bộ nhớ.
- DS (data Segment) đoạn DL : địa chỉ bắt đầu đoạn số liệu.

- SS (Stack Segment) đoạn ngăn xếp: địa chỉ logic đoạn ngăn xếp.
- EX (extra Segment) đoạn mở rộng: Đ/c DL các chuỗi.

*Thanh ghi cờ: Flag Register

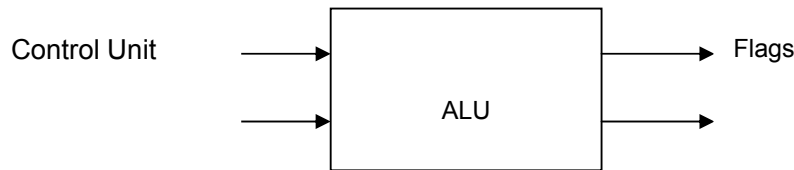
9 trong số 16 bit của thanh ghi này được sử dụng, mỗi bit có thể được thiết lập hay xóa để chỉ thị kết quả của mỗi thao tác trước đó hoặc trạng thái hiện thời bộ XL

- CF Carry : nhớ
- PF parity: chẵn lẻ
- ZF zero : kết quả phép toán =0
- SF sign : 0 dương, 1 âm.
- OF overflow : tràn

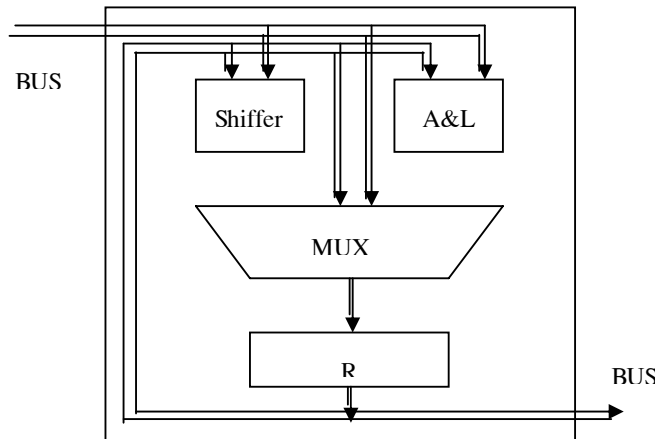
3. Đơn vị số học và logic ALU (Arithmetic and logic unit)

Đơn vị số học và logic (ALU) là một phần của máy tính thực sự thực hiện các thao tác số học và logic trên dữ liệu. Tất cả các thành phần khác của hệ thống máy tính-đơn vị điều khiển, thanh ghi, bộ nhớ, chủ yếu mang dữ liệu vào cho ALU để ALU xử lý và sau đó đưa kết quả ra ngoài.

Đơn vị số học và logic và tất cả các thành phần điện tử trong máy tính đều dựa trên việc sử dụng các thiết bị số đơn giản có thể chứa các con số nhị phân, và thực hiện các thao tác boolean logic đơn giản.



Hình trên chỉ ra trong một giới hạn chung, ALU được kết nối với phần còn lại của CPU như thế nào. Dữ liệu được sẵn sàng cho ALU trong các thanh ghi, và kết quả của một thao tác được chứa trong các thanh ghi khác. Các thanh ghi là chỗ chứa tạm thời trong CPU được kết nối bởi các đường tín hiệu tới ALU. ALU sẽ đặt cờ như là kết quả của một thao tác. Ví dụ có tràn được đặt lên 1 nếu kết quả của việc tính toán vượt quá chiều dài của thanh ghi chứa. Giá trị cờ được chứa trong các thanh ghi trong CPU. Đơn vị điều khiển cung cấp tín hiệu điều khiển thao tác của ALU, và sự di chuyển dữ liệu vào và ra khỏi ALU. (Cấu tạo của ALU được mô tả trong hình 1.2)



Các phép toán cơ bản của ALU

Bộ cộng, trừ:

4. Đơn vị điều khiển CU(Control Unit)

Như đã biết các thành phần chức năng cơ bản của CPU là:

- Đơn vị số học và Logic (ALU)
- Tập các Thanh ghi
- Các đường dữ liệu trong
- Các đường dữ liệu ngoài
- Đơn vị điều khiển(CU)

ALU là thành phần chức năng thực sự của máy tính, Các thanh ghi dùng để chứa dữ liệu trong CPU, Một vài thanh ghi chứa thông tin trạng thái cần để quản lý chỉ lệnh sắp xếp liên tục (ví dụ từ trạng thái chương trình). Những thanh ghi khác chứa dữ liệu đưa đến hoặc lấy từ ALU, bộ nhớ, module vào ra. Các đường dữ liệu trong được dùng chuyển dữ liệu giữa các thanh ghi, giữa các thanh ghi và ALU. Các đường dữ liệu ngoài liên kết các thanh ghi với bộ nhớ và module vào ra. thường bằng phương tiện của bus hệ thống. Đơn vị điều khiển tạo ra các thao tác xảy ra trong CPU.

Sự thi hành một chương trình bao gồm các thao tác liên quan đến các thành phần CPU. Như chúng ta đã thấy, các thao tác này bao gồm sự liên tục của các vi thao tác (vi điều khiển). Tất cả các vi thao tác là một trong các loại sau:

- Truyền dữ liệu từ một thanh ghi đến thanh ghi khác.
- Truyền dữ liệu từ một thanh ghi đến một giao diện ngoài (ví dụ system bus)
- Truyền dữ liệu từ một giao diện ngoài tới thanh ghi.
- Thực hiện thao tác số học và logic, sử dụng thanh ghi để nhận và ghi dữ liệu.

Tất cả các vi thao tác cần thực hiện trong một chu kỳ chỉ lệnh (bao gồm tất cả các vi thao tác để thực hiện mọi chỉ lệnh trong tập chỉ lệnh, nằm trong một trong những loại trên)

Đơn vị điều khiển thực hiện hai công tác chính:

- Sự sắp xếp chuỗi (sequencing): Đơn vị điều khiển khiến CPU sắp xếp chuỗi vi thao tác vào một chuỗi liên tục thích hợp, dựa trên chương trình đang được thực hiện
- Sự thi hành (Execution): Đơn vị điều khiển khiến mỗi vi thao tác được thực hiện.

Đơn vị điều khiển thao tác dựa vào việc sử dụng các **tín hiệu điều khiển**.

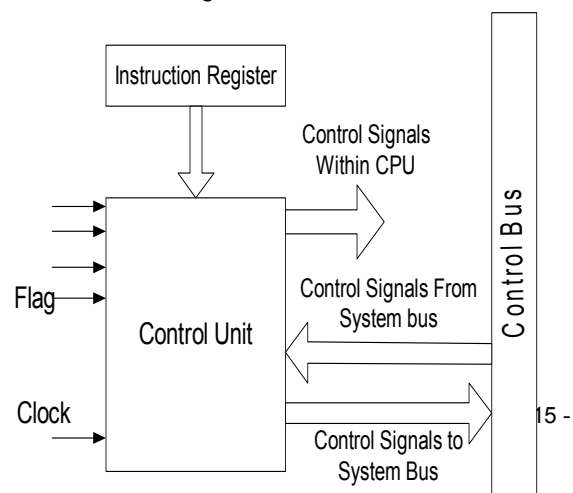
Tín hiệu điều khiển:

Chúng ta đã định nghĩa các thành phần đã tạo ra CPU (ALU, thanh ghi, đường dẫn dữ liệu) và các vi thao tác đang được thực hiện. Đối với đơn vị điều khiển để thực hiện các chức năng của nó, nó phải có dữ liệu vào cho phép nó xác định trạng thái của hệ thống và mục ra cho phép nó điều khiển tác động của hệ thống. Có các chi tiết kỹ thuật ngoài của đơn vị điều khiển. Nội tại, đơn vị điều khiển phải có logic yêu cầu thực hiện chuỗi vi thao tác và thi hành các chức năng.

Các yêu cầu của phần này là liên quan với sự tương tác giữa đơn vị điều khiển và các thành phần khác của CPU.

Hình 14.4 là mô hình chung của một đơn vị điều khiển, trình bày tất cả các tín hiệu vào và ra. Tín hiệu vào là:

- Clock: đây là cách đơn vị điều khiển “giữ thời gian” Đơn vị điều khiển tạo ra một vi thao tác (hoặc một tập các thao



tác đồng thời) được thực hiện với mỗi xung đồng hồ. Đây là một vài lần nhắc đến như là chu kỳ thời gian xử lý, hoặc chu kỳ thời gian đồng hồ.

- **Thanh ghi chỉ lệnh:** mã chỉ lệnh hiện tại được dùng để xác định vi thao tác nào được thực hiện trong chu kỳ thi hành.
- **Cờ:** Có các yêu cầu bởi đơn vị điều khiển để xác định trạng thái của CPU và kết quả của thao tác ALU trước. Ví dụ, đối với chỉ lệnh Increment and skip-if rezo (ISZ), đơn vị điều khiển sẽ lượng giá PC nếu cờ Rezo được đặt.
- **Các tín hiệu điều khiển từ bus điều khiển:** Khẩu phần bus điều khiển của bus hệ thống cung cấp tín hiệu cho đơn vị điều khiển, như là tín hiệu ngắt và sự công nhận.

Mục ra là:

- **Tín hiệu điều khiển trong CPU:** có 2 loại: Nó khiến dữ liệu bị di chuyển từ một thanh ghi tới các thanh ghi khác, và làm hoạt động các chức năng ALU cụ thể.
- **Các tín hiệu điều khiển điều khiển bus:** Cũng có 2 loại: các tín hiệu điều khiển bộ nhớ, và tín hiệu điều khiển module vào ra.

Thành phần mới đã được giới thiệu trong hình này là tín hiệu điều khiển. Ba kiểu tín hiệu được sử dụng: kích hoạt một chức năng ALU, kích hoạt các đường dữ liệu, và là các tín hiệu trên bus hệ thống ngoài hoặc giao diện ngoài. Tất cả các dạng tín hiệu này được cung cấp cuối cùng trực tiếp như các tín hiệu vào hay các cổng logic riêng biệt.

Chúng ta hãy xem xét lại chu kỳ tìm nạp để xem cách đơn vị điều khiển duy trì điều khiển. Đơn vị điều khiển giữ dấu vết nơi nó là trong chu kỳ tìm nạp. Tại điểm quy định, nó biết chu kỳ tìm nạp được thực hiện tiếp. Bước đầu tiên là di chuyển các nội dung của PC và MAR. Đơn vị điều khiển dùng nó bằng việc kích hoạt tín hiệu điều khiển mở các cổng giữa các bit của PC và các bit của MAR. Bước tiếp theo là đọc một từ trong bộ nhớ vào MBR và giá lượng PC. Đơn vị điều khiển làm việc này bằng việc gửi các tín hiệu kèm theo đồng thời.

1. Một tín hiệu điều khiển mở các cổng cho phép nội dung của MAR đưa vào bus địa chỉ.
2. Bộ nhớ đọc tín hiệu điều khiển trên bus điều khiển.
3. Một tín hiệu điều khiển các cổng cho phép nội dung bus dữ liệu được chứa trong MBR.
4. Các tín hiệu điều khiển thêm một vào nội dung của PC và chứa kết quả trở lại cho PC

Theo đó, đơn vị điều khiển gửi tín hiệu điều khiển mở các cổng giữa MBR và IR

(Memory buffer register)

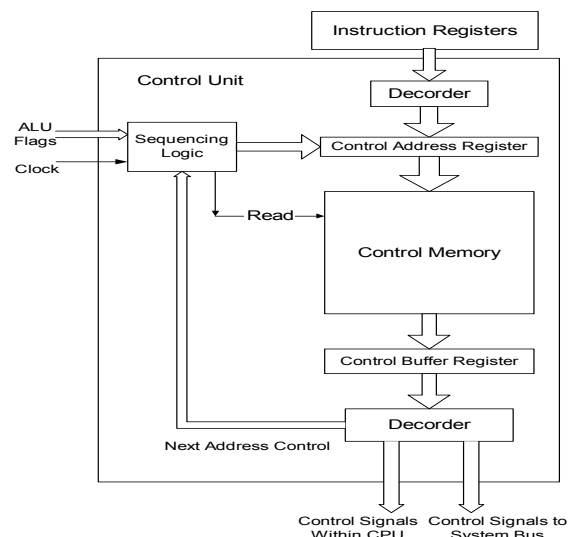
(Instruction Register)

Đơn vị điều khiển vi chương trình:

Các chức năng chính của đơn vị điều khiển này:

Để thực hiện một lệnh, đơn vị logic tuần tự đưa ra một lệnh đọc tới bộ nhớ điều khiển

1. Từ mã địa chỉ được xác định trong thanh ghi địa chỉ điều khiển được đọc vào thanh ghi bộ đệm điều khiển.
2. Nội dung của thanh ghi bộ đệm điều khiển phát ra tín hiệu điều



khiển và thông tin địa chỉ tiếp theo cho đơn vị logic tuần tự.

- Đơn vị logic tuần tự tải địa chỉ mới vào trong thanh ghi địa chỉ điều khiển dựa vào thông tin địa chỉ tiếp theo từ thanh ghi bộ đệm điều khiển và các cờ ALU.

Tất cả xảy ra trong một xung đồng hồ.

5. Một số mở rộng của vi xử lý máy tính cho đến ngày nay

Từ sự phát triển của các máy tính chứa chương trình đầu tiên những năm 1950, đã có một số sự cách tân thực sự rõ rệt trong các khu vực của tổ chức máy tính. Sau đây không phải là một danh sách hoàn chỉnh, mà chỉ là một vài tiến bộ chính kể từ ngày sinh của máy tính.

The Family Concept: được giới thiệu bởi IBM với hệ thống System/360 năm 1964, tiếp theo ngay sau đó là DEC với PDP-S. Khái niệm gia đình tách riêng kiến trúc của máy từ sự thi hành của nó. Một tập hợp các máy tính được đề nghị, với sự khác nhau giữa đặc trưng giá/tính năng đưa ra cùng một kiến trúc cho người dùng. Sự khác nhau trong giá và hiệu suất là bởi tại sự thi hành khác nhau của cùng một kiến trúc.

Đơn vị điều khiển vi chương trình (Microprogrammed Control Unit): Được đề xuất bởi Wikes năm 1951, và được giới thiệu bởi IBM trên hệ thống S/360 line trong năm 1964. Lập trình vi chương trình làm giảm bớt công tác thiết kế và thực hiện đơn vị điều khiển và hỗ trợ cho family concept.

Bộ nhớ Cache (cache Memory): Đầu tiên được giới thiệu rộng rãi trên hệ thống IBM S/360 Model 85 năm 1968. Sự thêm vào thành phần này trong hệ thống phân cấp bộ nhớ cải thiện rõ rệt hiệu suất

Pipelining: Một biện pháp đưa tính toán song song vào bản chất tuần tự của một chương trình chỉ lệnh máy. Các ví dụ là ống dẫn chỉ lệnh và xử lý vector

Instruction Pipelining

Như sự tiến hoá của các hệ thống máy tính, hiệu suất cao hơn có thể được đạt được bởi việc nắm bắt các tiến bộ của sự phát triển công nghệ. Hơn nữa, sự cải tiến tổ chức của CPU có thể làm tăng hiệu suất. Chúng ta đã có một số ví dụ ví như sử dụng các thanh ghi bội hơn là sử dụng một thanh ghi chứa đơn, và sử dụng bộ nhớ cache. Một phương pháp tổ chức khác rất thông dụng là Instruction Pipe. (Còn thiếu)

Chiến lược ống dẫn

Ống dẫn chỉ lệnh tương tự việc sử dụng một dây chuyền trong kế hoạch sản xuất. Một dây chuyền tạo ra các thuận lợi trong thực tế một sản phẩm đi qua nhiều trạng thái khác nhau của quá trình sản xuất. Bằng cách bố trí tiến trình sản xuất trong một dây chuyền, các sản phẩm ở những trạng thái khác nhau có thể được làm đồng thời. Tiến trình này cũng được quy cho là **pipelining**, Bởi vì như trong một ống dẫn, một sản phẩm vào mới được chấp nhận ở một đầu cuối trước các sản phẩm vào được chấp nhận trước đó xuất hiện như sản phẩm ra ở đầu cuối khác.

6. BUS

Một máy tính bao gồm các bộ phận hay các đơn vị của ba thành phần chính: CPU, hệ thống nhớ, thiết bị vào ra, được liên lạc với nhau. Về thực chất máy tính được coi là một mạng của các đơn vị cơ bản. Hơn nữa cần phải có các đường để kết nối các đơn vị với nhau. Tập hợp các đường kết nối các đơn vị được gọi là Interconnection Structure.

Cấu trúc kết nối thông dụng nhất được sử dụng trong máy tính là BUS

BUS là tập hợp các đường dây kết nối hai hay nhiều thiết bị với nhau. Rất nhiều thiết bị kết nối với BUS, một tín hiệu được truyền đi từ bất kì một thiết bị nào cũng có thể được gửi đến tất cả các thiết bị kết nối với BUS. Nếu có hai thiết bị cùng truyền dữ liệu đồng thời trong một thời điểm, những tín hiệu này sẽ gối lên nhau và sẽ bị sai lệch, như vậy chỉ một thiết bị có thể truyền dữ liệu thành công trong một thời điểm. Trong nhiều trường hợp, BUS thực sự gồm nhiều đường liên lạc, mỗi đường có khả

năng truyền các tín hiệu mô tả các giá trị nhị phân 0, 1. Các số nhị phân được truyền liên tục thông qua một đường, một số đường của BUS truyền các bit nhị phân đồng thời(kết nối song song).

Một hệ thống máy tính chứa đựng một số loại BUS khác nhau tùy thuộc các đường kết nối giữa các bộ phận ở các mức khác nhau của hệ thống. BUS kết nối các bộ phận chính của máy gọi là BUS hệ thống.

BUS hệ thống bao gồm từ 50 đến 100 đường truyền riêng biệt, mỗi đường được phân chia một chức năng hay một ý nghĩa riêng biệt. Mặc dù có rất nhiều cách thiết kế BUS khác nhau, nhưng trên bất kỳ cách nào các đường BUS cũng phân loại thành ba nhóm chính: BUS dữ liệu, BUS địa chỉ, BUS điều khiển, ngoài ra có thể có một số đường cung cấp năng lượng cho các module tham gia BUS.

BUS dữ liệu: truyền tải dữ liệu tới các thiết bị. Một BUS dữ liệu tiêu biểu bao

gồm 8, 16 hay 32 đường, số đường được coi là độ rộng của BUS dữ liệu. Mỗi đường chỉ có thể mang một bit dữ liệu tại một thời điểm, số lượng đường xác định số lượng bit có thể được truyền trong một thời điểm.

BUS địa chỉ: dùng chỉ định rõ nguồn gốc hay đích đến của dữ liệu trên BUS dữ liệu. Địa chỉ thường là địa chỉ các cổng vào/ra, từ nhớ trong ngăn nhớ.

BUS điều khiển: điều khiển việc truy nhập và việc sử dụng các đường địa chỉ và dữ liệu. Các đường dữ liệu và địa chỉ được chia sẻ cho tất cả các bộ phận, phải có sự điều khiển việc sử dụng các đường đó. Các tín hiệu điều khiển truyền cả lệnh và thông tin thời gian giữa các module hệ thống. Tín hiệu thời gian chỉ ra những thông tin về địa chỉ và dữ liệu hợp lệ. Các tín hiệu lệnh định rõ thao tác được thực hiện

Những đường điều khiển tiêu biểu:

- **Memory write:** điều khiển dữ liệu trên BUS được viết vào vị trí đã được xác định bằng địa chỉ
- **Memory read:** điều khiển việc đưa dữ liệu từ một vị trí xác định vào BUS
- **I/O write:** điều khiển đưa dữ liệu từ BUS ra cổng vào/ra đã xác định
- **I/O read:** điều khiển việc nhận dữ liệu từ cổng vào/ra chuyển vào BUS
- **Transfer ACK:** chỉ ra dữ liệu đã được chấp nhận
- **BUS request:** chỉ ra module cần chiếm quyền điều khiển BUS
- **BUS grant:** chỉ ra module đang yêu cầu đã được cấp quyền điều khiển BUS
- **Interrupt request:** yêu cầu ngắt từ thiết bị ngoại vi
- **Interrupt ACK:** chấp nhận ngắt từ CPU
- **Clock:** xung đồng hồ dùng trong quá trình đồng bộ
- **Reset:** khởi động lại các module

Phân loại BUS theo đường truyền

BUS đồng bộ: được điều khiển bởi nhịp đồng hồ với chu kỳ nhất định. Hoạt động của vi xử lý đòi hỏi thời gian là bội số của chu kỳ máy

BUS không đồng bộ: không hoạt động theo xung đồng hồ nhất định, khi truyền tín hiệu thiết bị truyền phát tín hiệu MSYN báo cho thiết bị nhận chạy nhanh nhất có thể, sau đó khi hoàn thành thiết bị nhận phát lại tín hiệu SSYN.

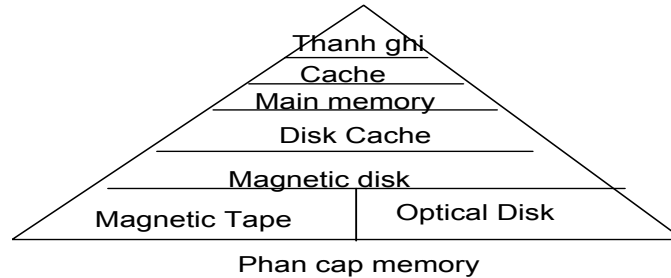
Chương III: HỆ THỐNG NHỚ

I. KHÁI QUÁT VỀ HỆ THỐNG NHỚ CỦA MÁY TÍNH

- Vị trí:
 - ✓ bên trong CPU (register)
 - ✓ Internal memory (main memory)
 - ✓ Bộ nhớ ngoài (đĩa cứng, đĩa quang)
- Dung lượng
 - ✓ Kích thước từ nhớ (word size) :thường là 8,16,32 bits
 - ✓ số lượng từ nhớ
- Đơn vị truyền
 - ✓ Word: đơn vị tự nhiên ở tổ chức bộ nhớ. Kích thước từ nhớ thường là số bit dùng để biểu diễn số hoặc độ dài lệnh .
 - ✓ Khối (block) là đơn vị truyền dữ liệu lớn hơn từ nhớ, thường được dùng truyền dữ liệu với bộ nhớ ngoài.
- Phương pháp truy nhập
 - ✓ Sequential access (truy nhập tuần tự) thường được dùng truy cập băng từ.
 - ✓ Truy nhập trực tiếp (direct memory) giống như truy nhập tuần tự, truy nhập trực tiếp bao hàm việc chia sẻ đọc viết cơ khí.Những từ nhớ của bản ghi có địa chỉ cơ sở duy nhất trên vị trí vật lý. Việc truy nhập được hoàn thành bởi truy nhập trực tiếp là đi đến vùng lân cận chung cộng với tìm kiếm tuần tự, đếm hoặc đợi để đi đến vị trí cuối cùng.Thời gian truy nhập có thể thay đổi được. Các loại đĩa sử dụng phương pháp truy nhập trực tiếp.
 - ✓ Truy nhập ngẫu nhiên (Random access) : mỗi vị trí địa chỉ trong bộ nhớ là độc nhất. Thời gian truy nhập các vị trí đã cho là độc lập với dãy truy nhập ưu tiên và là hằng số.Như vậy, vị trí nào cũng có thể được chọn ngẫu nhiên, và địa chỉ trực tiếp.Bộ nhớ chính là truy nhập ngẫu nhiên.
 - ✓ Truy nhập liên kết: đây là kiểu truy nhập ngẫu nhiên có thể làm sự so sánh vị trí bit trong từ cho một phép toán cụ thể và làm việc này cho tất cả các từ đồng thời. Vì vậy một từ được tìm lại được dựa vào chính nội dung của nó thay vì địa chỉ của nó.Với truy nhập ngẫu nhiên thông thường, mỗi vị trí có địa chỉ cơ khí của mình, và thời gian tìm là hằng số độc lập với vị trí hay mẫu hình truy nhập ưu tiên.Bộ nhớ cache dùng cách truy nhập này.
- Sự thi hành.
 - ✓ Thời gian truy nhập: (access time) : đối với truy nhập ngẫu nhiên đó là thời gian để thực hiện hoạt động đọc ghi. Đó là thời gian từ khi địa chỉ đã sẵn sàng trong bộ nhớ đến khi dữ liệu được cất trữ hoặc được làm có thể sử dụng được. Đối với truy nhập không phải là ngẫu nhiên thời gian truy nhập là thời gian đưa vị trí đọc viết cơ khí đến vị trí mong muốn.
 - ✓ Cycle time (chu kỳ thời gian):
 - ✓ Transfer rate: đó là tốc độ dữ liệu có thể được truyền vào hoặc ra khỏi đơn vị nhớ.
- Kiểu vật lý
 - ✓ Bán dẫn
 - ✓ Từ (magnetic)

- ✓ quang (optical)
- Đặc tính vật lý
 - ✓ Có thể thay đổi/ không thay đổi
 - ✓ Có thể xoá được/ không thể xoá được

II. PHÂN CẤP BỘ NHỚ



Việc phân cấp bộ nhớ theo các tiêu chuẩn:

- ✓ giảm giá/bit
- ✓ tăng dung lượng
- ✓ tăng thời gian truy nhập
- ✓ giảm tần số truy nhập của bộ nhớ bởi CPU.

Theo chiều từ trên xuống dưới:

- ✓ dung lượng tăng dần
- ✓ tốc độ truy nhập giảm dần.

III. BỘ NHỚ BÁN DẪN

1. Các loại bộ nhớ bán dẫn

Tất cả các loại bộ nhớ được trình bày sau đây là truy nhập ngẫu nhiên. Đó là những từ nhớ riêng biệt được truy nhập trực tiếp qua địa chỉ logic

- **RAM (random- access memory)** : đặc điểm phân biệt là có thể đọc dữ liệu từ bộ nhớ và dễ dàng ghi dữ liệu vào. Việc đọc và ghi dữ liệu được hoàn thành nhờ các tín hiệu điện.

Một đặc tính khác của RAM là thay đổi được. RAM được nuôi bằng một nguồn điện ổn định .Nếu nguồn nuôi bị ngắt dữ liệu trên RAM sẽ mất. Vì vậy RAM được dùng làm chỗ trữ tạm thời.

Công nghệ RAM chia làm 2 loại: RAM tĩnh và RAM động

- ✓ **RAM tĩnh**: giá trị nhị phân được cất trữ dùng các flip-flop truyền thống cấu hình cổng logic. Static RAM sẽ giữ được dữ liệu ổn định, tốc độ nhanh.
- ✓ **RAM động (Dinamic RAM)**: sử dụng các tế bào chứa dữ liệu dựa trên sự nạp điện cho các tụ điện. Vì các tụ điện có xu hướng phóng điện nên RAM động yêu cầu nạp điện làm tươi định kỳ để giữ thông tin .

- **ROM (Read only Memory)**

Tương phản với Ram là Rom. Rom chứa đựng các kiểu dữ liệu không thể bị thay đổi trong một thời gian dài. Một đặc tính của Rom là chỉ có thể đọc dữ liệu từ đó mà không thể ghi dữ liệu mới vào nó. Một ứng dụng quan trọng của Rom là chứa đựng các vi chương trình . Những ứng dụng tiềm tàng khác bao gồm:

- ✓ Thư viện thủ tục con cho các chức năng được sử dụng liên tục.
- ✓ Các chương trình hệ thống.

✓ Các bảng chức năng.

ROM được sản xuất tương tự như các mạch điện tích hợp khác., với dữ liệu được ghi vào chip trong quá trình chế tạo.

Phân loại:

- ✓ Maskable ROM: ghi khi chế tạo
- ✓ PROM (Programmable ROM) chỉ ghi một lần.
- ✓ EPROM (Erasable PROM) xoá được bằng tia cực tím.
- ✓ Flash Rom : Flash memory có thể xoá được bằng tín hiệu điện .
- ✓ Flash Rom có thể xoá và ghi lại được bằng tín hiệu điện. Thêm nữa nó có thể chỉ xoá các khối nhớ thay vì phải xoá toàn bộ chip. Flash memory sử dụng một transistor trên một bit, và do đó giành được mật độ cao.

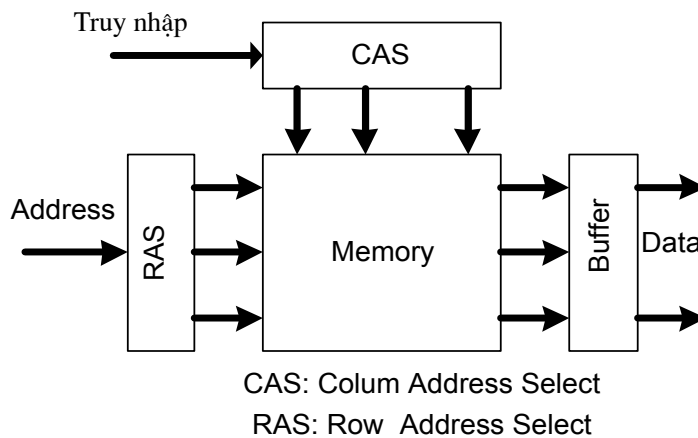
2. Tổ chức bộ nhớ

Dựa trên các mạch Flip- flop

Có 2^N ngăn nhớ -> N chân địa chỉ.

Độ dài mỗi ngăn nhớ m bits

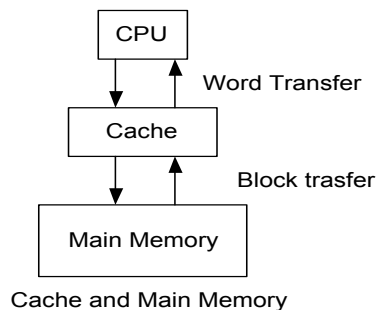
dung lượng $2^N * m$



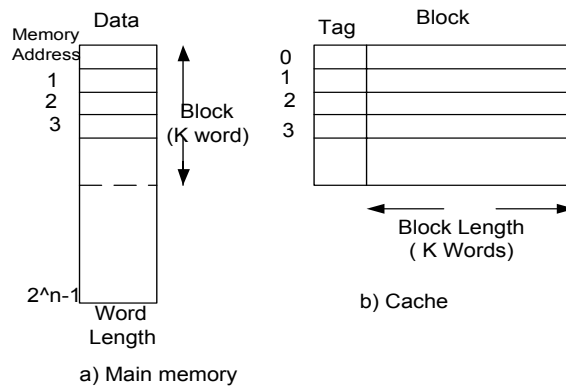
IV. CACHE MEMORY

1. Nguyên tắc (principle)

Cache memory được dùng cho tốc độ nhớ gần bằng tốc độ của các bộ nhớ nhanh nhất có sẵn, và tại cùng thời gian cung cấp một kích thước bộ nhớ rộng với giá không đắt hơn các kiểu bộ nhớ bán dẫn.



Đây là mối quan hệ giữa bộ nhớ chính lớn và chậm hơn với bộ nhớ cache nhỏ nhưng nhanh hơn. Bộ nhớ cache sao chép một phần của bộ nhớ chính. Khi CPU cố gắng đọc một từ nhớ của bộ nhớ, sự kiểm tra được làm để xác định rõ nếu từ nhớ nằm trong cache. Trong trường hợp đó, từ nhớ được cung cấp cho CPU. Nếu không khối nhớ của bộ nhớ chính, bao gồm một số từ nhớ cố định được đọc vào trong cache và sau đó từ nhớ được cung cấp cho CPU. Bởi vì hiện tượng tham vấn cục bộ, khi khối dữ liệu được đem về trong cache để thoả mãn tín hiệu tham chiếu bộ nhớ, hầu như những tham chiếu tương lai sẽ là những từ nhớ khác của khối nhớ.



Hình trên mô tả cấu trúc của hệ thống cache/Main memory. Bộ nhớ chính bao gồm tới 2^n từ nhớ có thể đánh địa chỉ, với mỗi từ nhớ có một địa chỉ n bit duy nhất. Cho mục đích ánh xạ, bộ nhớ này coi như bao gồm một số của độ dài những khối cố định của mỗi K từ nhớ. Tức là có $M=2^n/k$ khối nhớ. Cache bao gồm C khe của mỗi K từ nhớ, và số của các khe, hoặc các hàng, nó coi như ít hơn số khối nhớ của bộ nhớ chính ($C \ll M$). Tại bất kỳ thời điểm nào, một vài tập con của khối của bộ nhớ lưu trữ in các khe của cache. Nếu từ nhớ trong khối nhớ của bộ nhớ được đọc, khối nhớ đó được truyền vào 1 trong các khe của bộ nhớ cache. Bởi có nhiều khối nhớ hơn các khe, mộ khe riêng biệt không thể đọc nhất và thường xuyên dành cho một khối riêng biệt. Vì vậy, mỗi khe bao gồm 1 nhãn để nhận dạng khối riêng biệt hiện đang được trữ. Nhãn thường là một phần của địa chỉ bộ nhớ chính,.

Hoạt động đọc của cache: khi CPU phát địa chỉ, RA của từ nhớ sẽ được đọc. Nếu từ nhớ được chứa trong Cache,, nó sẽ được cung cấp cho CPU. Ngược lại, khối nhớ chứa từ nhớ đó sẽ được tải vào trong bộ nhớ và từ nhớ đó sẽ được cung cấp cho CPU.

2. Kỹ thuật ánh xạ bộ nhớ cache

Bộ nhớ chính có 2^N byte nhớ dùng N bit địa chỉ để địa chỉ hóa cho bộ nhớ.

Chia bộ nhớ chính thành các khối, mỗi khối có $K=2^{N_1}$ byte - có $M=2^N/k$ khối.

Chia Cache thành C đường, mỗi đường k byte nhớ: $C \ll M$. Việc trao đổi thông tin giữa bộ nhớ chính và cache theo đơn vị khối.

Vì có có ít đường cache hơn các khối nhớ của bộ nhớ chính, **một thuật toán** là cần thiết cho việc ánh xạ khối nhớ của bộ nhớ chính vào các đường của cache. Hơn nữa, có nghĩa là cần xác định khối bộ nhớ chính đang sử dụng cache line. Việc lựa chọn hàm ánh xạ ra lệnh cho việc tổ chức cache ntn.

Có 3 kỹ thuật ánh xạ

- ✓ ánh xạ trực tiếp: Direct Mapping
- ✓ ánh xạ liên kết hoàn toàn: Full Associative Mapping
- ✓ ánh xạ liên kết tập hợp: Set Associative Mapping

a. ánh xạ trực tiếp

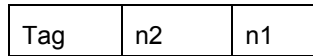
Kỹ thuật đơn giản nhất được biết đến là ánh xạ trực tiếp. ánh xạ mỗi khối nhớ của bộ nhớ chính vào một đường cache có thể.

- ✓ Block 0 -> line 0

- ✓ Block 1 -> line 1
- ✓ Block C -> line 0
- ✓ Block i -> line (i mod C)

Giả sử cache có 2^{n_2} ngăn nhớ (đường)

địa chỉ do CPU phát ra là n bit



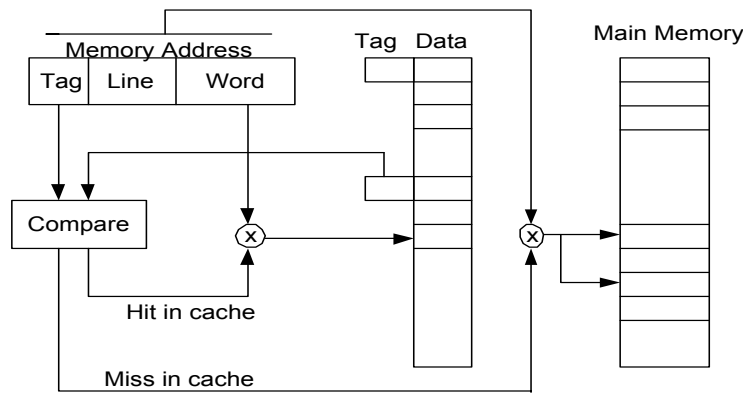
n_1 xác định số byte trong khối 2^{n_1} -> byte

$n - n_1$ bit còn lại : xác định khối nằm trong bộ nhớ chính.

n_2 bit tiếp theo xác định đường trong cache

còn lại là trường Tag

Mỗi 1 block được ghi vào cache thì cần 1 chỗ để ghi Tag (biết được đường nào nằm trong cache)



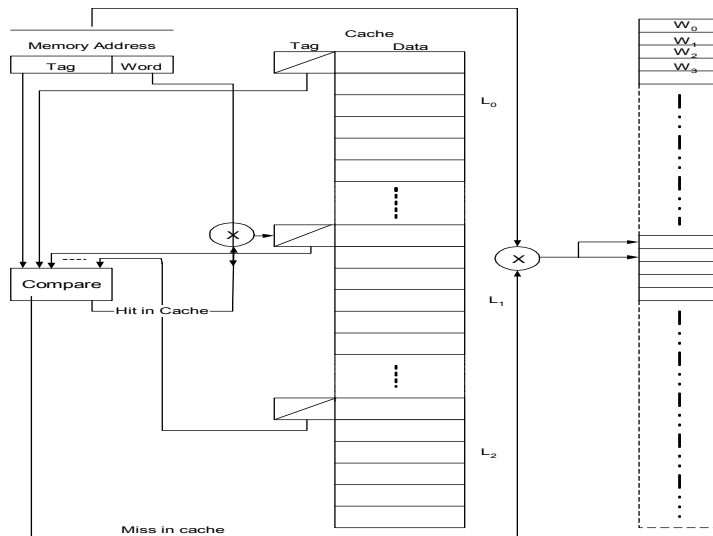
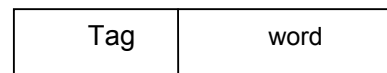
Bộ so sánh : thông dịch địa chỉ và so sánh nối vào.

Nhược điểm của phương pháp ánh xạ này là có một vị trí cố định của cache cho bất cứ khối đã cho nào. Hơn nữa nếu xảy ra chương trình muốn tham vấn lại từ nhớ từ 2 khối khác nhau được ánh xạ vào cùng một đường, khi đó các khối sẽ tiếp tục được trao đổi trong cache, và tỉ lệ thành công sẽ giảm xuống.

b. ánh xạ liên kết hoàn toàn

Ánh xạ liên kết sẽ khắc phục nhược điểm trên bằng cách cho phép mỗi khối bộ nhớ chính được nạp vào trong bất kỳ đường nào của cache.

Trong trường hợp này n bit chia ra 2 trường:



=Khi CPU phát ra địa chỉ thì nó so sánh với tất cả các Tag được ghi trong cache, nếu có 1 Tag nào trong cache trùng với Tag địa chỉ thì hit in cache

c. ánh xạ liên kết tập hợp (cache 2 đường)

Chia cache thành các tập hợp

Mỗi tập hợp có m đường

ánh xạ:

- ✓ Block 0 -> set 0
- ✓ ...
- ✓ Block i -> Set (i mod S)

Địa chỉ

Tag	Set	n1
-----	-----	----

Ví dụ :

Bộ nhớ chính 4 GB

Kích thước cache 16 kB

Block 32 byte

Số Bit địa chỉ là n=32 ($2^{32}=4\text{ G}$)

Xét trong trường hợp ánh xạ trực tiếp

Địa chỉ do CPU phát ra n=32 bit

1 khối (block) 32 byte-> n1=5 ($2^5=32$)

n2=số đường trong cache

$$\frac{16kb}{32byte} = \frac{2^{14}}{2^5} = 2^9 \rightarrow n2=9\text{ bit}$$

→ Tag=32-9-5=18 bit

Với liên kết hoàn toàn Tag=27

Với liên kết tập hợp (2 đường) Tag=19

> Trực tiếp

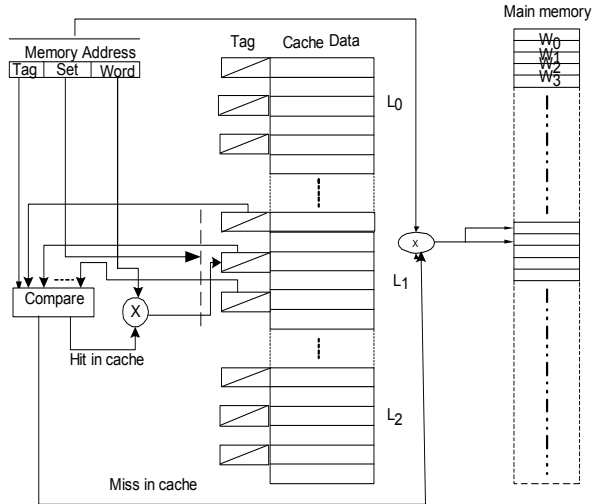
--	--	--

Hoàn toàn

--	--	--

Tập hợp

--	--	--



VI. QUẢN LÝ BỘ NHỚ

1. Các kỹ thuật quản lý bộ nhớ

Trong một hệ thống lập trình đơn nhiệm, bộ nhớ chính được chia thành hai phần: *một phần cho hệ điều hành(resident monitor)* và *một phần cho chương trình đang được thực hiện*. Trong một hệ thống lập trình đa nhiệm, “user” phần của bộ nhớ phải được chia nhỏ hơn nữa cung cấp cho các tiến trình

phức tạp. Công tác chia nhỏ được quản lý động bởi hệ điều hành và còn được biết dưới tên quản lý bộ nhớ (memory management).

quản lý bộ nhớ thực sự là quan trọng trong một hệ thống đa nhiệm. Nếu chỉ là một vài tiến trình trong bộ nhớ, trong hầu hết thời gian tất cả các tiến trình sẽ phải đợi việc truy nhập vào ra và bộ vi xử lý sẽ bị nhàn rỗi. Như vậy, bộ nhớ cần phải định vị hiệu quả để sắp đặt càng nhiều tiến trình trong bộ nhớ càng tốt.

a. Swapping (hoán đổi)

Trong hình 7.14 chúng ta đã thảo luận 3 kiểu hàng đợi (queues): hàng đợi cho các tiến trình mới, hàng đợi cho các tiến trình sẵn sàng sử dụng CPU, hàng đợi cho các tiến trình không sẵn sàng sử dụng CPU. Nhắc lại lý do của thiết bị phức tạp này là các hoạt động vào ra là chậm hơn rất nhiều so với sự tính toán. và bởi vậy cho nên CPU trong các hệ thống lập trình đơn nhiệm là nhàn rỗi trong hầu hết thời gian.

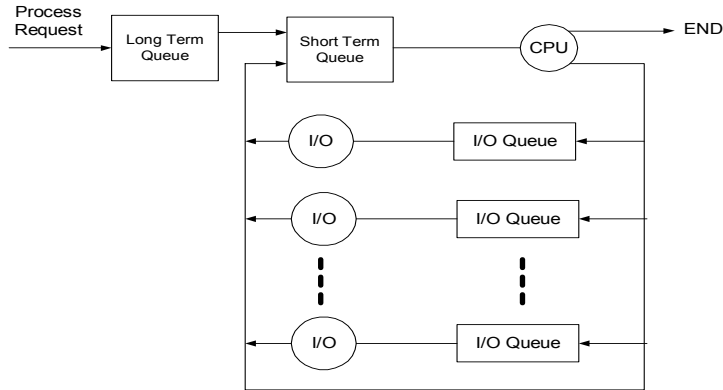
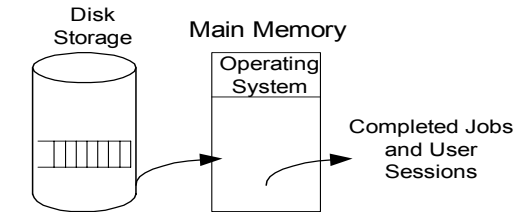


Figure 7.14 Queuing diagram representation of processor scheduling

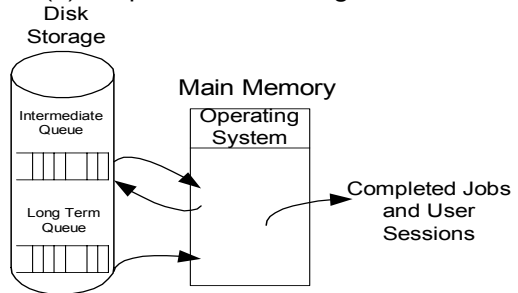
Nhưng sự sắp xếp trong hình 7.14 không giải thích toàn bộ vấn đề, Thực sự trong trường hợp này, bộ nhớ giữ các tiến trình và CPU có thể chuyển tới tiến trình khác khi một tiến trình đang đợi. Nhưng vì xử lý nhanh hơn thiết bị vào ra đối với tất cả các tiến trình trong bộ nhớ đang đợi I/O. Vì vậy thậm chí với lập trình đa nhiệm, một bộ CPU có thể nhàn rỗi trong hầu hết thời gian.

Bộ nhớ chính có thể được mở rộng, và có thể điều tiết cho nhiều tiến trình. Nhưng có hai trở ngại trong cách tiếp cận này. Đầu tiên, bộ nhớ chính là rất đắt. Thứ hai, sự yêu cầu bộ nhớ của các chương trình phát triển rất nhanh khi giá bộ nhớ giảm xuống. Kết quả các tiến trình chưa chắc tăng lên khi dung lượng bộ nhớ lớn hơn.

Một giải pháp khác là trao đổi (swapping). được mô tả trong hình 7.15. Chúng ta có một hàng đợi (long-term queue) của các yêu cầu tiến trình được trữ trên đĩa. Nó được nạp vào một lần khi không gian bộ nhớ cho phép. Khi các tiến trình hoàn thành, chúng được chuyển ra khỏi bộ nhớ chính. Bây giờ, trạng thái sẽ xuất hiện là không có bất kỳ tiến trình nào trong bộ nhớ nằm ở trạng thái sẵn sàng. Đúng hơn trong thời gian rỗi còn lại. CPU trao đổi một trong những tiến trình trở lại đĩa vào trong một hàng đợi trung gian. Đây là hàng đợi của các tiến trình đã tồn tại đã được tạm thời đưa ra khỏi bộ nhớ. Hệ điều hành sau đó nạp một tiến trình khác từ hàng đợi trung gian hoặc thực hiện một yêu cầu tiến trình mới từ hàng đợi (long-term queue).



(a) Simple Job Scheduling



(b) Swapping

Figure 7.15. The use of swapping

Kỹ thuật trao đổi, dù sao cũng là thao tác vào ra(I/O) và vì vậy có một tiềm năng tạo ra vấn đề nguy hiểm hơn, không tốt hơn. Nhưng, khi disk I/O là thiết bị vào ra nhanh nhất trên hệ thống(ví dụ so sánh với băng từ hoặc với vào ra máy in), trao đổi sẽ nâng cao sự thực thi. một giải pháp đơn giản hơn là bộ nhớ ảo sẽ cải thiện sự thi hành hơn sự trao đổi đơn giản. Bộ nhớ ảo sẽ được đề cập ngay sau đây, nhưng trước tiên, chúng ta phải chuẩn bị nền tảng bằng việc giải thích phân vùng và phân trang.

b. Phân vùng (Partitioning)

Hệ điều hành chiếm giữ một phần cố định của bộ nhớ. Phần còn lại của bộ nhớ được phân vùng cho việc sử dụng của các tiến trình. Luộc đồ đơn giản nhất cho bộ nhớ có thể phân vùng là sử dụng “các phân vùng kích thước cố định” (fixed-size partitions) như được trình bày trong hình 7.16.

Chú ý rằng, mặc dù các phân vùng có kích thước cố định, chúng không có kích thước bằng nhau. Khi một tiến trình được tải vào bộ nhớ, nó được đặt vào một phân vùng nhỏ nhất có thể.

Thậm chí với việc sử dụng những partition kích thước cố định không bằng nhau, sẽ có sự lãng phí bộ nhớ. Trong hầu hết các trường hợp, một tiến trình sẽ không yêu cầu chính xác dung lượng bộ nhớ được cung cấp bởi một phân vùng. Ví dụ, một tiến trình yêu cầu 128kbytes bộ nhớ có thể được đặt vào phân vùng dung lượng 192kbytes, lãng phí 64 kbytes không thể dùng bởi tiến trình khác.

Một cách tiếp cận hiệu quả hơn là sử dụng các phân vùng kích thước thay đổi được. Khi một tiến trình được nạp vào trong bộ nhớ, nó được cấp chính xác dung lượng bộ nhớ nó yêu cầu và không hơn. Một ví dụ trình bày trong hình 7.17. Bộ nhớ chính ban đầu rỗng, trừ phần cung cấp cho hệ điều hành (a). Nó bỏ mặc một “lỗ hổng” ở phần cuối bộ nhớ vì quá nhỏ cho tiến trình thứ 4. Khi tiến trình 2 được tráo đổi ra ngoài (b) có một chỗ trống cho tiến trình thứ 4. Tiến trình 4 nhỏ hơn tiến trình 2, một lỗ trống nhỏ được tạo ra. Như ví dụ đã trình bày, phương thức này khởi đầu tốt nhưng cuối cùng dẫn đến một trạng thái trong đó có rất nhiều “lỗ trống nhỏ” trong bộ nhớ. Càng ngày, bộ nhớ càng bị phân mảnh và không tận dụng được bộ nhớ. Một kỹ thuật khắc phục vấn đề này là “compaction”. Từ đó trở đi, hệ điều hành luân chuyển các tiến trình trong bộ nhớ để đặt tất cả “lỗ trống” lại với nhau trong một khối. Đây là một thủ tục lãng phí thời gian, lãng phí thời gian sử lý của CPU.

Operating System 128k
64K
192K
256K
384K

Figure 7.16 Example of Fixed Partitioning

Trước khi xem xét cách giải quyết với sự thiếu sót của sự phân vùng, chúng ta phải giải quyết một vấn đề. Nếu người đọc cân nhắc một chút, nó có thể trở lên rõ ràng rằng một tiến trình hầu như không được tải vào trong cùng một chỗ trong bộ nhớ mỗi lần nó được tráo đổi vào. Hơn nữa, nếu “compaction” được thực hiện một tiến trình có thể phải luân chuyển trong bộ nhớ chính. Bây giờ, tiến trình trong bộ nhớ bao gồm các chỉ lệnh và dữ liệu. Các chỉ lệnh sẽ bao gồm địa chỉ các vị trí trong bộ nhớ thuộc 2 loại:

- ✓ Địa chỉ của mục dữ liệu.
- ✓ Địa chỉ của các chỉ lệnh sử dụng cho sự phân nhánh chỉ lệnh

Nhưng bây giờ chúng ta thấy rằng những địa chỉ đó không cố định. Chúng sẽ thay đổi mỗi lần tiến trình được tráo đổi. Để giải quyết vấn đề này, một sự phân biệt được tạo ra giữa địa chỉ logic và địa chỉ vật lý. Địa chỉ logic được biểu diễn một vị trí liên quan tới khởi đầu của chương trình. Các chỉ lệnh trong chương trình bao hàm chỉ một địa chỉ logic. Địa chỉ vật lý là vị trí thực trong bộ nhớ chính. Khi CPU thực hiện một tiến trình, nó tự động chuyển đổi từ địa chỉ logic sang địa chỉ vật lý bằng việc công thêm vị trí khởi đầu hiện tại của tiến trình, được gọi là địa chỉ cơ sở, cho mỗi địa chỉ logic. Một ví dụ khác của CPU là đặc tính phần cứng được thiết kế tương thích với yêu cầu của hệ điều hành. trạng thái tự nhiên chính xác của đặc trưng phần cứng phụ thuộc vào chiến thuật quản lý bộ nhớ được sử dụng.

c. Phân trang

Cả các phân vùng kích thước cố định và thay đổi đều không hiệu quả trong việc sử dụng bộ nhớ. Giả định, bộ nhớ đã được phân vùng vào các chunks liên kết nhỏ có kích thước cố định bằng nhau, và mỗi tiến trình cũng được phân chia vào trong một số các chunks đó. Sau đó các chunks của một chương trình, được gọi là các “trang”, có thể được phân bổ vào các chunks có thể trong bộ nhớ, gọi

là các frame (khung), hoặc trang khung. Tất nhiên phần bộ nhớ lãng phí của tiến trình này là một phần nhỏ trên trang cuối cùng.

Hình trên trình bày một ví dụ về việc sử dụng các trang và các khung. Ở mỗi điểm thời gian đã cho, một vài khung trong bộ nhớ được sử dụng và một vài khung trống. Danh sách các khung trống được duy trì bởi hệ điều hành. Tiến trình A, được trữ trên đĩa, bao gồm bốn trang. Khi đến thời gian nạp tiến trình này, hệ điều hành tìm 4 khung trống và nạp bốn trang của tiến trình A vào trong bốn khung đó.

Bây giờ giả định, như trong ví dụ này, không có các khung trống kế tiếp để giữ tiến trình. Nó có ngăn cản hệ điều hành nạp tải tiến trình không? Câu trả lời là không, bởi vì chúng ta có thể sử dụng lại khái niệm địa chỉ logic. Một địa chỉ cơ sở đơn giản sẽ không đủ. Hơn nữa hệ điều hành duy trì một bảng trang cho mỗi tiến trình. Bảng trang lưu giữ vị trí các khung cho mỗi trang của tiến trình. Trong chương trình, mỗi địa chỉ logic bao gồm một số hiệu trang và một địa chỉ quan hệ trong trang. Trong trường hợp phân vùng đơn giản một địa chỉ logic là vị trí của một từ chỉ tới điểm khởi đầu của chương trình; CPU biên dịch nó thành địa chỉ vật lý. Với phân trang sự biên dịch địa chỉ logic-vật lý vẫn được làm bởi phần cứng CPU. Bây giờ CPU phải biết cách truy nhập bảng trang của tiến trình hiện tại. Trình diện với một địa chỉ logic (số hiệu trang, địa chỉ liên quan), CPU sử dụng bảng trang để đưa ra địa chỉ vật lý (số hiệu khung, địa chỉ liên quan).

Con đường này giải quyết vấn đề đơn giản hơn. Bộ nhớ chính được chia thành các khung nhỏ có kích thước bằng nhau. Mỗi tiến trình được chia vào các trang khung(frame-size page). Các tiến trình nhỏ hơn yêu cầu ít trang hơn các tiến trình lớn. Khi một tiến trình được nạp vào, các trang của nó được tải vào các khung rồi và một bảng trang được thiết lập.

2. Bộ nhớ ảo

a. Yêu cầu phân trang

*Với việc sử dụng phân trang, các hệ thống chương trình đa nhiệm trở lên thực sự hiệu quả. Việc chia nhỏ một tiến trình vào các trang dẫn đến sự phát triển của một khái niệm quan trọng khác: **bộ nhớ ảo**.*

Để hiểu bộ nhớ ảo, chúng ta phải tìm hiểu về bản đồ phân trang. Sự tìm hiểu chọn lọc này là yêu cầu phân trang, chúng chỉ đơn giản là mỗi trang của tiến trình được đưa vào trong bộ nhớ chỉ một khi chúng được cần đến, đó là yêu cầu.

Xem xét một tiến trình lớn, bao gồm một chương trình dài cộng với một số lượng lớn các mảng dữ liệu. Trong bất kỳ chu trình nào, sự thực hiện có thể bị hạn chế chỉ một section(đoạn) chương trình(ví dụ một thủ tục con).và có lẽ chỉ một hoặc hai mảng dữ liệu được dùng. Đây là nguyên tắc định hướng. Sẽ rất lãng phí khi tải vào rất nhiều trang cho tiến trình này khi chỉ một vài trang được dùng trước đó khi chương trình tạm treo. Chúng ta có thể sử dụng bộ nhớ tốt hơn bằng việc tải vào chỉ một vài trang. Sau đó, nếu chương trình phân nhánh tới một chỉ lệnh trên một trang không nằm trong bộ nhớ chính, hoặc nếu chương trình tham chiếu dữ liệu trên một trang không nằm trong bộ nhớ, một ngoại lệ lỗi trang sẽ xảy ra. Nó yêu cầu hệ điều hành nạp các trang đã yêu cầu vào bộ nhớ.

hư vậy, bất kỳ thời điểm nào, chỉ một vài trang của chương trình là nằm trong bộ nhớ và bởi vậy rất nhiều tiến trình có thể được duy trì trong bộ nhớ. . Hơn nữa, thời gian được ghi lại bởi vì những trang không được dùng không được trao đổi vào ra trong bộ nhớ. Tuy nhiên, hệ điều hành phải khéo léo quản lý bản đồ phân trang. Khi nó nạp một trang vào, nó phải đẩy một trang khác ra. Nếu nó đẩy ra một trang vừa được sử dụng, sau đó nó sẽ phải tìm lại trang đó ngay lập tức. Rất nhiều trong chúng dẫn tới một trạng thái được gọi là **Thrashing**: tiến trình tiêu tốn hầu hết thời gian trao đổi các trang hơn là thực hiện các chỉ lệnh. Việc tránh thrashing đã là một nghiên cứu chính trong thập niên 70 và dẫn đến nhiều giải thuật phức hợp đa dạng nhưng hiệu quả.

Với yêu cầu phân trang, nó không cần thiết tải toàn bộ chương trình vào trong bộ nhớ chính. Thực tế có một kết quả rõ rệt: các tiến trình có thể lớn hơn bộ nhớ chính. Một trong những giới hạn nền tảng trong lập trình đã được nâng lên. Không yêu cầu phân trang, một lập trình viên phải nhận thức sâu

sắc về dung lượng bộ nhớ cho phép. Nếu chương trình được viết quá lớn, lập trình viên phải nghĩ ra con đường cấu trúc chương trình vào trong các bộ phận nhỏ có thể được nạp vào tại một thời điểm. Với yêu cầu phân trang, công việc đó được giao phó cho hệ điều hành và phần cứng. Người lập trình sẽ được phân phát một bộ nhớ rất lớn, kích thước liên kết với thiết bị lưu trữ ngoài. Hệ điều hành sử dụng yêu cầu phân trang để nạp các phần của tiến trình vào bộ nhớ chính.

Bởi vì một tiến trình thực hiện chỉ trong bộ nhớ chính, chỉ bộ nhớ chính được tham chiếu như là bộ nhớ thực sự. Nhưng một lập trình viên, hoặc người dùng nhận thấy một bộ nhớ lớn hơn rất nhiều được định rõ trên đĩa cứng. Nó được tham chiếu đến như là bộ nhớ ảo.

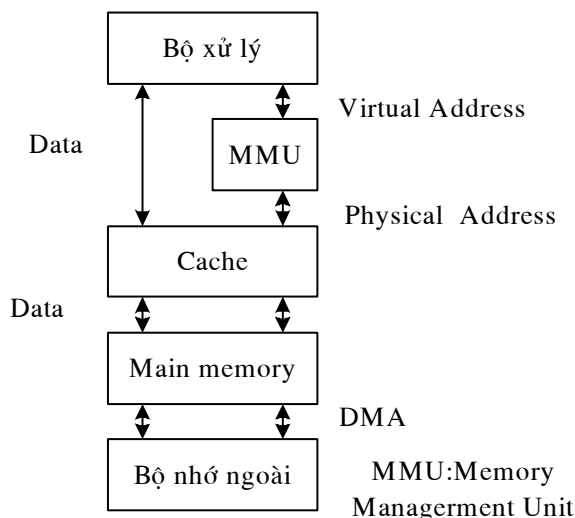
Bộ nhớ ảo cho phép lập trình đa nhiệm hiệu quả và giảm nhẹ những ép buộc không cần thiết của người dùng với bộ nhớ chính.

b. Phân trang trong quản lý bộ nhớ ảo

Chia bộ nhớ thành các trang nhớ có kích thước cố định từ vài Kb->vài chục kb

Bộ xử lý phát ra địa chỉ ảo thông qua MMU để chuyển thành địa chỉ vật lý.

Kích thước trang với các bộ xử lý 86x thường là 4kbyte hay 4Mb. Các trang này có thể ánh xạ vào bộ nhớ vật lý hay đĩa cứng. Khi một chương trình (hay một nhiệm vụ) yêu cầu truy nhập một địa chỉ logic, VXL biên dịch địa chỉ logic này sang địa chỉ tuyến tính. Sau đó, dùng phương pháp phân trang biên dịch địa chỉ tuyến tính sang địa chỉ vật lý tương ứng. Nếu trang chứa địa chỉ tuyến tính trên không tồn tại trong bộ nhớ vật lý, bộ VXL ra ngoại lệ "lỗi trang" #PF. Chương trình xử lý ngoại lệ này nạp trang cần truy nhập từ đĩa cứng về bộ nhớ vật lý (có thể nạp một trang khác từ bộ nhớ vật lý lên đĩa cứng để lấy chỗ). Sau khi trang cần truy nhập có mặt trong bộ nhớ vật lý, lệnh return từ chương trình xử lý ngoại lệ khiến bộ vi xử lý thực hiện lệnh đã gây ra ngoại lệ #PF. Thông tin mà bộ vi xử lý dùng để ánh xạ địa chỉ tuyến tính vào không gian địa chỉ vật lý (cũng như để tạo ngoại lệ "lỗi trang") được lưu trữ trong danh mục trang (PDE) và bảng trang. Danh mục trang và bảng trang đều nằm trong bộ nhớ vật lý.



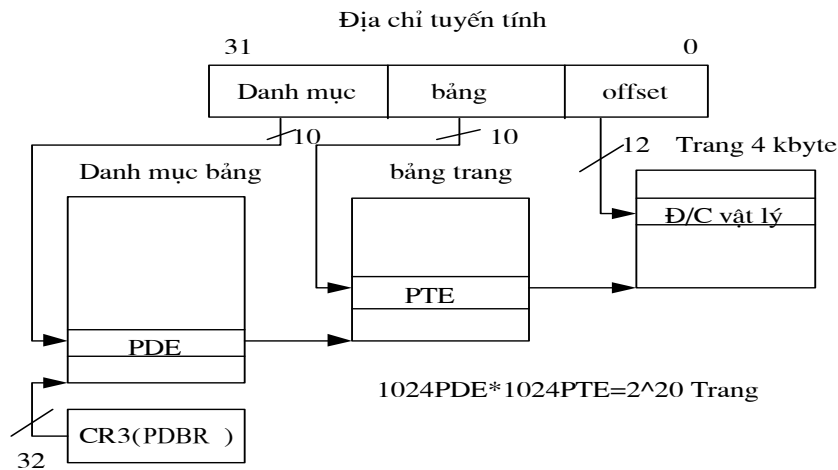
c. Cách quản lý trang và phương pháp biên dịch địa chỉ tuyến tính

Máy tính quản lý bộ nhớ phân trang dựa trên danh mục trang, bảng trang và trang.

- ✓ Danh mục trang: Tổ hợp các giá trị 32 bit, được gọi là PDE (page-directory entry). Danh mục trang có chiều dài bằng một trang (4kbyte) và chứa 1024 PDE.
- ✓ Bảng trang: tổ hợp các giá trị 32 bit, được gọi là PTE, bảng trang cũng có chiều dài bằng 1 trang(4kbyte) và chứa tất cả 1024 PTE. Nếu dùng trang kích thước 4 Mbyte hay 2 Mbyte thì bộ vi xử lý không cần đến PTE. Các trang lớn (4 hay 2 Mb) được ánh xạ trực tiếp từ danh mục trang.
- ✓ Trang (page) không gian địa chỉ phẳng có kích thước 4 kbyte, 4 Mbyte,
- ✓ Bảng con trở danh mục trang: tổ hợp 3 giá trị 64 bit, mỗi giá trị trở đến một danh mục trang. Cấu trúc này chỉ được dùng khi mở rộng không gian địa chỉ lên 36 bit.

Biên dịch địa chỉ tuyến tính (4kbyte)

Hình vẽ cho ta thấy cách dùng danh mục trang và bảng trang khi ánh xạ địa chỉ tuyến tính sang trang 4 kbyte. Giá trị trong danh mục trang trở đến một vị trí trong bảng trang, giá trị tương ứng trong bảng trang trở đến trang cần truy nhập trong bộ nhớ vật lý. Cơ chế này cho phép truy nhập 2^{20} trang hay một không gian bộ nhớ vật lý gồm (2^{32} byte = 4 Gbyte).



Biên dịch địa chỉ tuyến tính (trang 4 kbyte)

Để biên dịch địa chỉ tuyến tính sang địa chỉ vật lý, địa chỉ tuyến tính được chia làm 3 phần:

- ✓ Giá trị danh mục trang: bit 22 đến 31, là giá trị lệch của một vị trí (PDE) trong bảng danh mục (có tất cả 2^{10} PDE), PDE cho biết vị trí bảng trang cần truy nhập.
- ✓ Giá trị bảng trang: bit 12 đến bit 21 là giá trị lệch của một vị trí (PTE) trong bảng trang. PTE cho biết địa chỉ cơ sở của một trang trong bộ nhớ vật lý.
- ✓ Địa chỉ lệch trong trang: bit 0 đến bit 11 cho biết vị trí của byte cần truy nhập trong trang được tron bằng giá trị PTE.

Địa chỉ cơ sở của danh mục trang

Địa chỉ cơ sở của danh mục trang hiện tại được lưu trữ trong thanh ghi điều khiển CR3 (vì vậy thanh ghi này còn được gọi là thanh ghi cơ sở danh mục trang PDBR-page directory base register). Nếu dùng cơ chế phân trang, thanh ghi PDBR cần được nạp ngay khi khởi động máy (trước khi cho phép cơ chế phân trang). Nội dung của thanh ghi PDBR có thể được thay đổi tự động khi thay đổi nhiệm vụ bằng lệnh MOV.

3. Sự phân đoạn

Có một cách quản lý bộ nhớ khác gọi là phân đoạn bộ nhớ. Trong khi việc phân trang cho phép cung cấp cho lập trình viên một không gian địa chỉ lớn hơn, sự phân đoạn thường cung cấp cho lập trình viên một tiện ích cho việc tổ chức các chương trình và dữ liệu, và như là một phương tiện cho việc tích hợp quyền ưu tiên và các thuộc tính bảo vệ với các chỉ lệnh và dữ liệu.

Sự phân đoạn cho phép người lập trình xem xét bộ nhớ như là việc bao gồm nhiều không gian địa chỉ hay các đoạn. Các đoạn là biến đổi, động về kích thước. Đặc biệt, người lập trình hoặc hệ điều hành sẽ phân bổ các chương trình và dữ liệu vào các đoạn khác nhau. Có thể có một số lượng các đoạn chương trình cho các kiểu chương trình khác nhau giống như một số lượng các đoạn dữ liệu. Mỗi đoạn có thể được phân bổ truy nhập và quyền sử dụng. Sự tham chiếu bộ nhớ bao gồm một dạng địa chỉ. Cách tổ chức này có một số thuận lợi cho người lập trình so với không gian địa chỉ không phân đoạn.

1. nó làm đơn giản cách trình bày sự phát triển của các cấu trúc dữ liệu. Nếu lập trình viên không biết kích thước của một cấu trúc dữ liệu riêng biệt sẽ là bao nhiêu, nó sẽ không cần hướng dẫn. Cấu trúc dữ liệu có thể được phân bổ và đoạn của nó và hệ điều hành sẽ mở rộng hoặc co rút lại đoạn nếu cần thiết.
2. Nó cho phép các chương trình được biến đổi và biên dịch lại độc lập, không yêu cầu thiết đặt toàn bộ các chương trình phải được liên kết và nạp tải lại. Một lần nữa, đây đang sử dụng hoàn hảo đa đoạn .

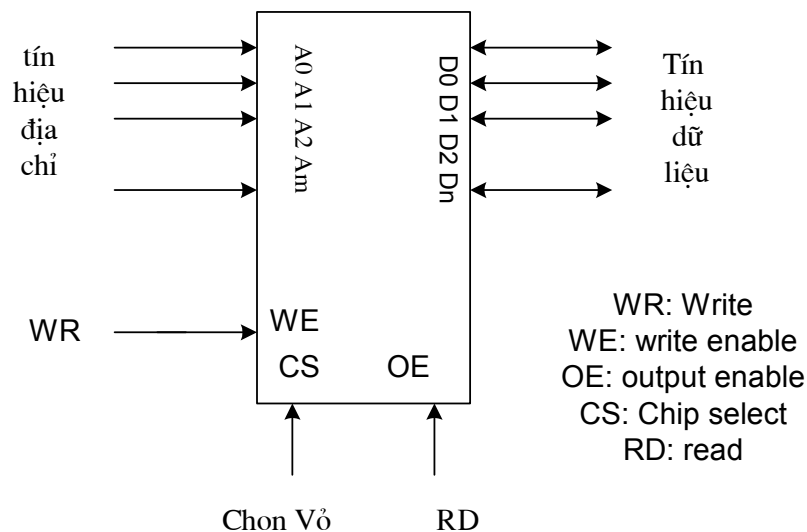
3. Nó tự giúp việc chia sẻ giữa các tiến trình. Một lập trình viên có thể đặt một chương trình tiện ích hoặc một bảng dữ liệu hữu dụng trong một đoạn có thể được đánh địa chỉ bởi một tiến trình khác.
4. Nó tự bảo vệ mình. Khi một đoạn có thể được xây dựng để chứa đựng một tập các chương trình hoặc dữ liệu đã được định nghĩa, lập trình viên hoặc nhà quản trị mạng có thể phân bổ quyền ưu tiên truy nhập trong một kiểu phù hợp.

Những thuận lợi này không có sẵn với việc phân trang, (việc phân trang là không thấy được với lập trình viên). Nói cách khác, chúng ta đã nhìn thấy kỹ thuật phân trang cung cấp một kiểu quản lý bộ nhớ hiệu quả. Để kết hợp các thuận lợi của cả hai, một số hệ thống được hỗ trợ với phần cứng và phần mềm hệ điều hành để cung cấp cả hai phương pháp quản lý.

VII. KỸ THUẬT GIẢI MÃ ĐỊA CHỈ

1. Cấu tạo một vi mạch nhớ

Một mạch nhớ được tạo nên từ nhiều vi mạch nhớ. Một vi mạch nhớ thường có cấu trúc tiêu biểu như



sau:

Nhóm tín hiệu địa chỉ : các tín hiệu địa chỉ có tác dụng chọn ra một ô nhớ (từ nhớ) cụ thể để ghi / đọc. Các ô nhớ có độ dài khác nhau tùy theo nhà sản xuất : 1,2,4,8 bit. Một vi mạch nhớ có m bit địa chỉ thì có 2^m từ nhớ.

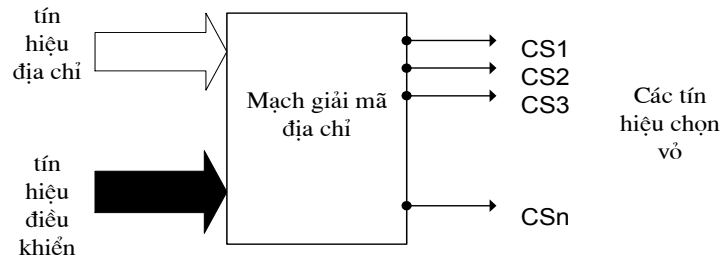
- Nhóm tín hiệu dữ liệu: Số đường dây dữ liệu quyết định độ dài từ nhớ của mạch nhớ. Thông thường người ta hay nói rõ dung lượng và độ dài từ nhớ cùng một lúc. Ví dụ mạch nhớ: 1K x 8...
- Nhóm tín hiệu chọn vi mạch (chọn vỏ): Các tín hiệu chọn vỏ là CS (chip select) hoặc CE (chip enable) thường được dùng để chọn ra vi mạch nhớ cụ thể được ghi / đọc. Tín hiệu ở mạch nhớ Ram thường là CS ở ROM là CE. Các tín hiệu chọn vỏ thường nối với đầu ra của mạch giải mã địa chỉ. Khi một mạch nhớ không được chọn thì bus dữ liệu của nó bị treo (ở trạng thái trở kháng cao).
- Nhóm tín hiệu điều khiển: Một mạch nhớ RAM thường có 1 tín hiệu điều khiển là R/W để điều khiển quá trình ghi / đọc. Nếu mạch nhớ RAM có 2 tín hiệu điều khiển thì đó là WE (write enable) để điều khiển ghi và OE để điều khiển đọc. Hai tín hiệu này phải ngược pha nhau để điều khiển việc ghi đọc mạch nhớ.

2. Giải mã địa chỉ cho bộ nhớ

Mỗi một mạch nhớ ghép nối với CPU cần phải được CPU quy chiếu tới một cách chính xác khi thực hiện thao tác ghi / đọc. Điều đó có nghĩa là mỗi mạch nhớ phải được gán cho một vùng riêng biệt có địa chỉ xác định nằm trong không gian địa chỉ tổng thể của bộ nhớ. Việc gán địa chỉ cụ thể cho từng mạch nhớ được thực hiện nhờ bộ giải mã địa chỉ.

Việc phân thành các vùng nhớ khác nhau để thực hiện các chức năng nhất định gọi là phân vùng bộ nhớ.

a. Nguyên tắc của bộ giải mã địa chỉ thường có cấu tạo như sau:



Mạch giải mã địa chỉ tổng quát

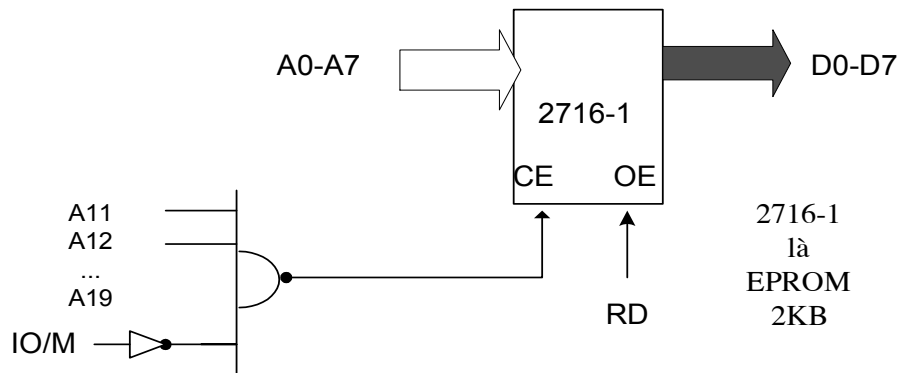
- Tín hiệu điều khiển : IO/ M dùng để phân biệt đối tượng mà CPU chọn làm việc là bộ nhớ hay thiết bị vào ra.
- Tín hiệu địa chỉ : là các bit địa chỉ có quan hệ nhất định đến việc chọn võ ở đầu ra.

Thông thường khi thiết kế mạch giải mã người ta thường tính dôi ra để dự phòng, sao cho sau này có thể tăng thêm dung lượng bộ nhớ.

3. Giải mã địa chỉ bằng các mạch NAND

Ví dụ mạch giải mã đơn giản cho EPROM 2761-1 dung lượng 2Kx8 có địa chỉ nằm trong khoảng FF800H-FFFFFH (vùng địa chỉ có chứa địa chỉ khởi động của CPU 8088.)

Sơ đồ mạch giải mã:



A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

=FF800h
=FFFFh

Số chân địa chỉ trong CPU 8088 là 20 chân đánh số từ A_0 đến A_{19} . Trong mạch giải mã này EPROM dung lượng 2KB -> sử dụng 11 bit địa chỉ thấp từ A_0 đến A_{10} để chọn từ nhớ trong EPROM. Các bit cao còn lại A_{11} đến A_{19} kết hợp với xung IO/M (đã được đảo) để tạo xung chọn vỏ cho 2kb đặt tại vùng nhớ cao nhất của CPU 8088

4. Giải mã dùng mạch giải mã kiểu 74LS138

a. Sơ đồ mạch giải mã LS138

4 Giải mã dùng mạch giải mã kiểu 74LS138 (mạch giải mã 3 ra 8).

Sơ đồ mạch giải mã LS138

Bảng chức năng:

A	B	C	G2B	G2A	G1	y0	y1	y2	y3	y4	y5	y6	y7
x	x	x	1	x	x	1	1	1	1	1	1	1	1
x	x	x	x	1	x	1	1	1	1	1	1	1	1
x	x	x	x	x	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	1	1	1	1	1
0	1	1	0	0	1	1	1	1	0	1	1	1	1
1	0	0	0	0	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	0	1	1
1	1	0	0	0	1	1	1	1	1	1	1	0	1
1	1	1	0	0	1	1	1	1	1	1	1	1	0

Ví dụ giải mã địa chỉ vùng nhớ 64 kb địa chỉ bắt đầu F0000 h đến FFFFF h bằng EPROM 8KB

Với địa chỉ vùng nhớ như trên cần 20 bit địa chỉ để mã hoá : $A_0 \div A_{19}$

dung lượng của EPROM là 8 kb -> cần 13 bit địa chỉ để xác định từ nhớ (ô nhớ)-> sử dụng 13 bit thấp để mã hoá: $A_0 \div A_{12}$.

Các bit địa chỉ $A_{13} \div A_{15}$ được đưa vào các đầu vào chọn A, B, C của bộ giải mã LS138 để chọn EPROM tương ứng.

Các chân còn lại $A_{16} \div A_{18}$ qua mạch NAND đưa vào chân G_{2a}

tín hiệu IO/M được đưa vào chân G_{2b}

A_{19} được đưa vào G_1

Nếu dùng EPROM 2Kb -> cần sử dụng 32 mạch nhớ.-> sử dụng 4 bộ giải mã LS138 để chọn chip nhớ tương ứng.

Dung lượng của EPROM là 2kb -> sử dụng 11 bit thấp $A_0 \div A_{10}$ để xác định từ nhớ trong mạch nhớ.

Các bit $A_{11} \div A_{13}$ được đưa vào chân chọn vỏ của các bộ giải mã LS138

Bit A_{14} đưa vào chân G_{2A} của bộ giải mã LS138 1 và 3; qua mạch NOT đổi thành xung âm đưa vào chân G_{2A} của bộ giải mã LS138 2 và 4.

Bit A_{15} được đưa vào chân G_{2B} của bộ giải mã LS138 1 và 2; qua mạch NOT thành xung âm đưa vào chân G_{2B} của bộ giải mã 3,4.

Các chân $A_{16}÷A_{19}$ Kết hợp với xung tín hiệu IO/M (đã được đổi dấu) qua mạch AND đưa vào chân G1.

Sơ đồ ghép nối EPROM và giải mã địa chỉ:

Bài tập : Cách ghép nối và giải mã địa chỉ vùng nhớ 8 KB có địa chỉ $F0000h÷FFFFFh$ sử dụng mạch giải mã LS138 và EPROM 1Kx4 .

Chương IV: TẬP LỆNH VÀ CÁC MODE ĐỊA CHỈ

1. Tập lệnh của CPU

Chức năng máy tính:

- ✓ Xử lý tin
- ✓ Truyền thông

Về cơ bản việc xử lý thông tin và truyền thông đều dựa trên nguyên tắc thực hiện lệnh (Instruction).

Lệnh bao gồm:

- ✓ Mã lệnh + Toán hạng
- ✓ *Toán tử*

Toán tử chứa mã lệnh dạng tượng trưng

- ✓ Mã lệnh: chức năng của thao tác
- ✓ Dẫn hướng biên dịch: toán tử chứa toán tử giả(pseudo_op), các toán tử giả này không được dịch sang mã máy mà chỉ báo cho chương trình dịch làm việc gì đó.

Toán hạng

- ✓ Toán hạng: xác định dữ liệu sẽ được thao tác
- ✓ Toán hạng: Đích, Nguồn

Các nhóm lệnh của CPU

Ngôn ngữ máy (Machine Language):

Chương trình đưa vào bộ nhớ cho máy thực hiện theo nhiều dạng, dạng cơ bản nhất mà máy có thể hiểu ngay được gọi là ngôn ngữ máy. Tùy theo CPU mà ngôn ngữ máy có dạng nhất định, chương trình viết bằng ngôn ngữ máy thực hiện rất nhanh và chiếm ít chỗ trong bộ nhớ, tuy nhiên chương trình khó viết và khó nhớ.

Hợp ngữ(Assembly)

Ngôn ngữ giúp lập trình viên viết chương trình dễ dàng hơn, thay cho ngôn ngữ máy. Một lệnh của hợp ngữ tương đương như một lệnh của ngôn ngữ máy nhưng thay viết chương trình dưới dạng nhị phân sẽ dùng kí hiệu tượng trưng.

Để biểu diễn các nhóm lệnh CPU, dùng tập lệnh Hợp ngữ dùng cho VXL 8086 Intel. Các nhóm lệnh bao gồm:

- ✓ Nhóm lệnh cơ sở
- ✓ Các lệnh vào ra
- ✓ Nhóm lệnh số học
- ✓ Nhóm lệnh logic
- ✓ Các lệnh điều khiển, rẽ nhánh.

Nhóm lệnh cơ sở

Lệnh MOV và XCHG

Lệnh MOV(move): chuyển dữ liệu giữa các thanh ghi, giữa 1 thanh ghi và 1 ô nhớ, hoặc trực tiếp 1 số vào 1 thanh ghi hay ô nhớ

Cú pháp:

MOV

đích,nguồn;

không làm thay đổi nội dung nguồn

Lệnh XCHG(exchange): hoán chuyển nội dung của 2 thanh ghi, 1 thanh ghi và 1 ô nhớ

Cú pháp:

XCHG đích,nguồn

Chú ý: 1 lệnh trên không hợp lệ khi cả đích và nguồn là các ô nhớ

Lệnh LEA (load effective address): nạp địa chỉ thực(hiệu dụng)

Hàm 9h của ngắt 21h yêu cầu địa chỉ tương đối của chuỗi kí tự chứa trong DX, thực hiện điều này dùng lệnh LEA

Cú pháp:

LEA đích,nguồn

Đích: thanh ghi công dụng chung

Nguồn: ô nhớ

Nhóm lệnh vào/ra

CPU liên lạc với các thiết bị ngoại vi qua các thanh ghi vào/ra hay các cổng vào/ra. Có 2 lệnh truy nhập trực tiếp các cổng đó là lệnh IN và OUT, tuy nhiên ít sử dụng

Lệnh INT: dùng để gọi các chương trình con ngắt của DOS và BIOS

Cú pháp:

IN số hiệu ngắt

Số hiệu ngắt là 1 số xác định 1 chương trình(ngắt mềm)

Ta xem xét các ngắt của DOS:

Ngắt 20h: kết thúc chương trình, 1 chương trình có thể dùng ngắt 20h để trả điều khiển về cho DOS(dùng trong chương trình đuôi .COM)

Ngắt 22h-26h: các phục vụ quản lý CTRL + BREAK, các lỗi nghiêm trọng và truy nhập trực tiếp đĩa

Ngắt 27h: kết thúc chương trình và ở lại thường trú

Ngắt 21h: gọi các hàm:

Hàm 0h: kết thúc chương trình

Hàm 1h: vào từ bàn phím : đọc 1 ký tự từ thiết bị vào chuẩn sau đó đưa kí tự đó tới thiết bị ra và trả về mã ASCII của dữ liệu trong DL

- AH = 01h
- AL = kí tự vào

Hàm 2h: hiển thị: đưa kí tự trong DL ra thiết bị ra chuẩn

- AH = 02h
- DL = kí tự ra

Hàm 5h: in ra: đưa dữ liệu trong DL ra thiết bị in

- AH = 5h
- DL = kí tự ra

Hàm 9h: in chuỗi: đưa chuỗi kí tự ra thiết bị chuẩn

- AH = 9h

- DS:DX con trỏ đến chuỗi kí tự kết thúc bằng \$

Chú ý: hàm 2h của ngắt 21h cũng có thể sử dụng để thực hiện 1 chức năng điều khiển nếu DL chứa mã ASCII của kí tự điều khiển, hàm này sẽ thi hành chức năng đó:

Mã ASCII	Kí hiệu	Chức năng điều khiển
7	BEL	Phát tiếng Bíp
8	BS(back space)	Lùi lại 1 kí tự
9	HT	Tab
A	LF(line feed)	Xuống dòng
D	CR(carry return)	Xuống dòng, về đầu dòng

Nhóm lệnh số học

Các chỉ thị ADD, SUB, INC, DEC

Các chỉ thị ADD(add) và SUB(subtract): được sử dụng để cộng hoặc trừ nội dung của 2 thanh ghi, 1 thanh ghi và 1 ô nhớ hoặc cộng trừ 1 số vào thanh ghi hay 1 ô nhớ

Cú pháp:

ADD **đích,nguồn**

SUB **đích,nguồn**

Ví dụ:

SUB AX,DX; AX nhận giá trị AX - DX

ADD BL,5; BL nhận giá trị BL + 5

Chú ý: phép cộng, trừ trực tiếp giữa các ô nhớ là không hợp lệ

Chỉ thị INC(Increment) và DEC(decrement): được sử dụng để cộng, trừ 1 vào nội dung 1 thanh ghi hay ô nhớ

Cú pháp:

INC **đích**

DEC **đích**

Ví dụ:

INC WORD; WORD nhận giá trị WORD + 1

Lệnh NEG (negavi): lệnh NEG dùng phủ định nội dung của toán hạng đích, thay thế nội dung bởi phần bù 2

Cú pháp:

NEG **đích**

Toán hạng đích có thể là 1 thanh ghi hay ô nhớ

Các lệnh nhân và chia

Các lệnh MUL(Multiply) và IMUL(Integer MUL)

Khi xét các số có dấu và không dấu thì kết quả thực hiện phép toán là khác nhau, với các số có dấu ta dùng lệnh IMUL, các số không dấu ta dùng lệnh MUL

Cú pháp:

MUL **toán hạng nguồn**

IMUL toán hạng nguồn

Khi nhân các byte với nhau, 1 số được chứa trong toán hạng nguồn, số còn lại được giả thiết đã chứa trong AL. Toán hạng nguồn có thể là 1 thanh ghi hay ô nhớ.

Các lệnh DIV(Device) và IDIV(Integer DIV)

Khi xét các số có dấu và không dấu thì kết quả thực hiện phép toán là khác nhau, với các số có dấu ta dùng lệnh IDIV, các số không dấu ta dùng lệnh DIV

Cú pháp:

DIV	số chia
IDIV	số chia

Dạng byte: số chia là thanh ghi hay ô nhớ 1 byte, số bị chia: 16 bit được giả định chứa trong AX thương 8 bit chứa trong AL, số dư 8 bit trong AH

Dạng word: số chia 16 bit, số bị chia giả định chứa trong DX:AX, thương 16 bit chứa trong AX, số dư 16 bit trong DX

Các lệnh logic dịch và quay

Các lệnh logic: AND, OR, NOT, XOR: thay đổi từng bit trong máy

Cú pháp:

AND	đích,nguồn
OR	đích,nguồn
XOR	đích,nguồn

Ảnh hưởng tới các cờ:

- SF, ZF, PF phản ánh kết quả lệnh
- AF không xác định
- CF, OF=0

Lệnh AND: xoá các bit nhất định của toán hạng đích trong khi giữ nguyên các bit còn lại, các bit cần xoá được AND với 0

Lệnh OR: thiết lập các bit xác định của toán hạng đích khi vẫn giữ nguyên các bit còn lại, các bit cần thiết lập được OR với 1

Lệnh XOR: đảo các bit xác định

Sử dụng:

Đổi mã ASCII của 1 số thành số tương ứng: khi đọc 1 kí tự từ bàn phím, AL sẽ chứa mã ASCII của kí tự đó vì vậy dùng lệnh AND đổi mã ASCII của kí tự ra giá trị thập phân tương ứng

Ví dụ:

Số 5 mã ASCII là 35h vì vậy khi đổi ra giá trị thập phân thực hiện phép **AND** AL với 0Fh

Đổi chữ thường thành chữ hoa: có thể dùng lệnh SUB(**SUB đích,20h**) hoặc dùng lệnh AND:
AND đích,0DFh

Xoá 1 thanh ghi:

XOR đích,nguồn; đích=nguồn

Ví dụ:

MOV AX,0

SUB AX,AX

XOR AX,AX

Kiểm tra xem 1 thanh ghi có bằng 0 hay không:

OR **đích,nguồn; đích=nguồn**

Ví dụ:

OR CX,CX;(CMP CX,0)

Lệnh NOT: lấy số bù 1 của toán hạng đích

Cú pháp:

NOT **toán hạng đích; không ảnh hưởng tới cờ**

Lệnh TEST: thực hiện phép AND giữa toán hạng đích với toán hạng nguồn nhưng không làm thay đổi toán hạng đích mà chỉ thiết lập cờ

Cú pháp:

TEST **toán hạng đích,toán hạng nguồn**

Các cờ bị tác động:

SF, ZF, PF: phản ánh kết quả

AF: không xác định

CF, OF=0

Các lệnh dịch và quay

Quay và dịch các bit trong toán hạng đích sang trái hoặc phải 1 hoặc 1 số vị trí

Dịch: các bit bị dịch ra khỏi toán hạng sẽ bị mất

Quay: các bit bị dịch ra 1 phía của toán hạng đích sẽ được đưa trở lại phía bên kia quay 1 vị trí:

Mã lệnh **Toán hạng đích,1**

quay N vị trí

Mã lệnh **Toán hạng đích,CL; CL chứa N**

Các lệnh dịch trái

Lệnh SHL(Shift Left): dịch các bit của toán hạng sang trái 1 vị trí:

SHL **toán hạng đích,1;**

giá trị 0 sẽ được đưa vào vị trí bên phải nhất của toán hạng, còn MSB của nó sẽ được đưa vào CF N vị trí

SHL **toán hạng đích,CL**

N phép dịch trái sẽ được thực hiện

Có thể dùng lệnh SHL để thực hiện phép nhân nhị phân

Tương tự như vậy có lệnh SAL(Shift Arithmetic Left)

Lệnh SHR(Shift Right): dịch các bit của toán hạng sang phải 1 vị trí:

SHR **toán hạng đích,1;**

giá trị 0 sẽ được đưa vào vị trí bên phải nhất của toán hạng, còn LSB của nó sẽ được đưa vào CF N vị trí

SHR toán hạng đích,CL

N phép dịch phải sẽ được thực hiện

Có thể dùng lệnh SHR để thực hiện phép chia nhị phân

Tương tự như vậy có lệnh SAR(Shift Arithmetic Right)

Các lệnh quay

Lệnh ROL(Rotate Left): dịch các bit của toán hạng sang trái 1 vị trí:

ROL toán hạng đích,1;

bit MSB sẽ được dịch vào vị trí bên phải nhất của toán hạng, đồng thời được đưa vào CF N vị trí

ROL toán hạng đích,CL

N phép quay trái sẽ được thực hiện

Lệnh ROR(Rotate Right): dịch các bit của toán hạng sang phải 1 vị trí:

ROR toán hạng đích,1;

bit bên phải nhất(LSB) sẽ được dịch vào vị trí bên trái (MSB) nhất của toán hạng, đồng thời được đưa vào CF N vị trí

ROR toán hạng đích,CL

N phép quay phải sẽ được thực hiện

Ví dụ: đếm số bit 1 có trong thanh ghi BX mà không làm thay đổi nội dung BX, kết quả lưu trong AX

XOR AX,AX; xoá AX

MOV CX,16; biến đếm vòng lặp

Top:

ROL BX,1

JNC Next; bit 0?

INC AX; không, tăng biến đếm kết quả

Next:

LOOP Top; quay lại

Lệnh RCL: quay trái qua cờ nhớ

Lệnh RCL(Rotate Carry Left): dịch các bit của toán hạng đích sang trái. Bit MSB được đặt vào CF, giá trị của CF được đưa vào bit phải nhất(LSB) của toán hạng đích

Cú pháp:

RCL toán hạng,1

hoặc

RCL toán hạng,CL

Lệnh RCR: quay phải qua cờ nhớ: tương tự

Ví dụ: đảo các bit trong 1byte hay 1word

MOV CX,8; số lần lặp

Everse:

SHL **AL,1**; lấy 1 bit vào CF
RCR **BL,1**; quay, đưa vào BL
LOOP Reverse

MOV **AL,BL**; đưa vào AL

Các lệnh điều khiển rẽ nhánh

Cho phép chọn lựa và lặp lại các đoạn mã lệnh

Các lệnh nhảy có điều kiện

Cú pháp:

Tên lệnh **nhãn đích**

Nếu điều kiện của lệnh nhảy thoả mãn, lệnh có nhãn đích sẽ được thực hiện. Lệnh này có thể ở trước hoặc sau lệnh nhảy. Nếu điều kiện không thoả lệnh ngay sau lệnh nhảy được thực hiện

Phạm vi của lệnh nhảy có điều kiện:

Nhãn đích phải đứng trước lệnh nhảy không quá 126byte hoặc đứng sau lệnh nhảy không quá 127byte

- ✓ CPU thực hiện 1 lệnh nhảy ntn?
- ✓ CPU dựa vào thanh ghi cờ để điều chỉnh IP chỉ đến nhãn đích

các lệnh nhảy có điều kiện

Kí hiệu	Chức năng	Điều kiện nhảy
JG/JNLE	Nhảy nếu lớn hơn, nhảy nếu không nhỏ hơn hay bằng (Jump if Greater, Jump if Not Less than or Equal)	ZF=0 và SF=0
JGE/JNL	Nhảy nếu lớn hơn hay bằng, nhảy nếu không nhỏ hơn	SF=OF
JL/JNGE	Nhảy nếu nhỏ hơn, nhảy nếu không lớn hơn hay bằng	SF<>OF
JLE/JNG	Nhảy nếu nhỏ hơn hoặc bằng, nhảy nếu không lớn hơn	ZF=1 hay SF=OF
Các lệnh nhảy không dấu		
JA/JNBE	Nhảy nếu lớn hơn, nhảy nếu không nhỏ hơn hoặc bằng (Jump if Above, Jump if Not Below or Equal)	CF=0 và ZF=0
JAЕ/JNB/JNC	Nhảy nếu không nhớ (No Carry)	CF=0
JB/JNAE/JC	Nhảy nếu có nhớ	CF=1
JBE/JNA	Nhảy nếu nhỏ hơn hay bằng	CF=1 hay ZF=1
Các lệnh nhảy điều kiện đơn		
JE/JZ	Nhảy nếu kết quả bằng nhau, nhảy nếu kết quả bằng không (Jump if Equal, Jump if Zero)	ZF=1
JNE/JNZ	Nhảy nếu không bằng nhau, nếu kết quả khác 0	ZF=0
JO	Nhảy nếu tràn (Jump if Overflow)	OF=1
JNO	Nhảy nếu không tràn	OF=0

<i>Kí hiệu</i>	<i>Chức năng</i>	<i>Điều kiện nhảy</i>
JS	Nhảy nếu dấu âm(Jump if Signed)	SF=1
JNS	Nhảy nếu dấu dương	SF=0
JP/JPE	Nhảy nếu chẵn(Jump if Parity, Jump if Parity Even)	PF=1
JNP/JPO	Nhảy nếu lẻ(Jump if Parity Odd)	PF=0

Lệnh CMP(compare)

Các điều kiện nhảy thường được cung cấp bởi lệnh CMP

Cú pháp:

CMP đích,nguồn

So sánh các toán tử đích với toán tử nguồn bằng cách lấy toán tử đích trừ đi toán tử nguồn

Toán tử đích không thể là hằng số, các toán tử không cùng là ô nhớ

Lệnh JMP(Jump)

Lệnh JMP dẫn đến việc chuyển điều khiển không điều kiện

Cú pháp:

JMP đích

Đích phải là 1 nhãn trong cùng 1 đoạn với lệnh JMP

Cấu trúc lặp

Lặp: cho phép lặp lại 1 đoạn chương trình nào đó, số lần lặp có thể biết trước hoặc không biết trước

Vòng lặp FOR

FOR **số lần lặp** DO

các dòng lệnh

END_FOR

Thực hiện: dùng lệnh LOOP

Cú pháp:

LOOP nhãn đích

Bộ đếm vòng lặp là thanh ghi CX, được khởi tạo bằng số lần lặp

Mỗi lần thực hiện LOOP thanh ghi CX tự động giảm đi 1, và nếu CX<>0 thì điều khiển được chuyển tới nhãn đích. Nếu CX=0 thì lệnh tiếp theo LOOP sẽ được thực hiện

3. Hợp ngữ(Assembly)

Cú pháp của hợp ngữ

Các chương trình hợp ngữ được dịch ra các chỉ thị máy bằng một chương trình biên dịch vì vậy khi viết phải phù hợp với các khuôn mẫu của trình biên dịch đó

Các dòng lệnh:

Chương trình là tập hợp của các dòng lệnh, bao gồm:

- Lệnh mà trình biên dịch dịch ra mã máy
- Lệnh dẫn hướng biên dịch

Cú pháp:

Tên Toán tử Toán hạng Chú thích

Các trường cách nhau ít nhất một dấu cách hay TAB

Ví dụ:

START:MOV CX,5 ; Khởi tạo CX

a. Tên trường:

Sử dụng: Nhãn lệnh, tên thủ tục, tên biến

Chương trình biên dịch chuyển các tên thành địa chỉ bộ nhớ

- ✓ Độ dài: 1 đến 31 kí tự(không chứa dấu cách)
- ✓ Không bắt đầu bởi chữ số
- ✓ Không phân biệt chữ hoa, chữ thường
- ✓ Nếu có dấu chấm(.) phải đặt ở đầu

b. Toán tử

- ✓ Toán tử chứa mã lệnh dạng tượng trưng
- ✓ Mã lệnh: chức năng của thao tác
- ✓ Dẫn hướng biên dịch: toán tử chứa toán tử giả(pseudo_op), các toán tử giả này không được dịch sang mã máy mà chỉ báo cho chương trình dịch làm việc gì đó.

c. Toán hạng

- ✓ Toán hạng: xác định dữ liệu sẽ được thao tác
- ✓ Toán hạng: Đích, Nguồn

d. Lờì giải thích

Đặt sau dấu ; và giải thích xem dòng lệnh đó làm gì

Dữ liệu chương trình

Biểu diễn dữ liệu dưới dạng số nhị phân, thập phân, hexa thậm chí kí tự

- ✓ Số nhị phân: kết thúc bằng B hoặc b
- ✓ Số thập phân: kết thúc bằng D hoặc d
- ✓ Số hexa: kết thúc bằng H hoặc h, bắt đầu bằng chữ số thập phân

Các ký tự: bao trong dấu nháy kép ""

Các toán tử giả định nghĩa số liệu

DB byte
DW word(2 byte)
DD double word(2 word)
DQ quard word(4 word)
DT 10 byte liên tiếp

Các biến

Mỗi biến có một kiểu dữ liệu và được chương trình gán cho một địa chỉ bộ nhớ

Biến kiểu Byte

Định nghĩa:

Tên DB giá trị khởi tạo

ví dụ:

ALPHA DB 4

giới hạn thập phân của các giá trị khởi tạo: -128 đến 127 hoặc 0 đến 255

nếu dùng dấu ? thì biến không được khởi tạo

Biến kiểu Word

Định nghĩa:

Tên DW giá trị khởi tạo

ví dụ:

ALPHA DW ?

Biến kiểu Mảng

Mảng: chuỗi byte nhớ hay từ nhớ

Định nghĩa:

Tên Kiểu giá trị khởi tạo

Phần tử đầu tiên của mảng chính là tên mảng

mảng byte: tên+1 là phần tử tiếp theo

mảng word: tên+1 là phần tử tiếp theo

ví dụ:

ARRAY DB 4h,5h,6h

phần tử 1: ARRAY

phần tử 2: ARRAY+1

phần tử 3: ARRAY+2

Chú ý: byte thấp và byte cao trong một từ

ví dụ:

WORD DB 1234h

byte thấp: WORD, nội dung: 34h

byte cao: WORD+1, nội dung: 12h

Chuỗi kí tự

Có thể được khởi tạo bằng bảng mã ASCII

CHAR DB 'ABC'

hoặc

CHAR DB 41h,42h,43h

cũng có thể kết hợp các kí tự và số:

MSG DB 'HELLO',0Ah,'\$'

hoặc

MSG DB 48h,45h,4Ch,4Fh,0Ah,24h

chú ý phân biệt chữ hoa và chữ thường

Các hằng có tên

Dùng các tên tượng trưng để biểu diễn các hằng số

Cú pháp:

Tên	EQU	Hằng số
------------	------------	----------------

ví dụ:

CONSTEQU		0Ah
----------	--	-----

cũng có thể dùng chuỗi:

MSG	EQU	"hello"
-----	-----	---------

chú ý: bộ nhớ không dành chỗ cho các hằng có tên

Cấu trúc chương trình

DOS thi hành được hai loại tập tin: dạng .COM và .EXE. Tập tin dạng .EXE thường dùng để xây dựng các chương trình lớn, còn các tập tin .COM tạo các chương trình nhỏ hơn. ASM cho phép tạo cả hai loại tập tin nói trên song cách viết là khác nhau:

Tập tin dạng .COM

Đặc điểm:

- Chỉ có duy nhất một đoạn, chương trình, dữ liệu và STACK đều chung đoạn này
- Kích thước tối đa của tệp là 64K
- Thực hiện nhanh hơn tệp .EXE

Cách thực hiện một tệp tin dạng .COM

- DOS khởi tạo vùng nhớ 256byte offset: 0000h gọi là vùng nhớ PSP(Program Segment Prefix)
- Định vị tệp tin vào vùng nhớ với offset 100h
- Các thanh ghi đoạn CS, DS, ES, SS trở tới PSP
- IP được gán giá trị 100h
- SP được gán giá trị FFFh

Cấu trúc chương trình

.MODEL TINY

.CODE

ORG 100h

START: JMP CONTINUE

; khai báo dữ liệu

CONTINUE:

MAIN PROC

; đoạn mã

...

INT 20h; về DOS

MAIN ENDP

; các lệnh chương trình con

END START

Tập tin dạng .EXE

Đặc điểm:

- Chương trình có thể khai báo nhiều đoạn khác nhau, mỗi chương trình có thể có nhiều đoạn chương trình, nhiều đoạn dữ liệu
- Có thể gọi chương trình con
- Kích thước của tệp tùy ý và lớn hơn 64K

Cách thực hiện một tệp tin dạng .EXE

- DOS khởi tạo vùng nhớ 256byte offset: 0000h gọi là vùng nhớ PSP(Program Segment Prefix)
- Nạp Header của tệp sau PSP
- Các thanh ghi đoạn CS, IP được xác định từ Header là địa chỉ bắt đầu của chương trình

Cấu trúc chương trình

.MODEL SMALL

.STACK

.DATA

; khai báo dữ liệu

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

; các lệnh

...

MOV AX,4Ch

INT 21h; về DOS

MAIN ENDP

; các lệnh chương trình con

END

MAIN

Các chỉ dẫn

.MODEL: xác định kiểu bộ nhớ dành cho đoạn mã và đoạn dữ liệu

Các kiểu bộ nhớ thường dùng:

- TINY: mã và dữ liệu nằm trong phạm vi 1 đoạn
- SMALL: mã nằm trong phạm vi đoạn 64K, dữ liệu nằm trong đoạn khác
- MEDIUM: mã nằm trong đoạn >64K, dữ liệu nằm trong đoạn 64K
- COMPACT: mã nằm trong phạm vi đoạn 64K, và dữ liệu nằm trong đoạn >64K
- LARGE: mã và dữ liệu nằm trong đoạn >64K, nhưng một mảng dữ liệu <64K
- HUGE: mã và dữ liệu nằm trong đoạn >64K, mảng dữ liệu >64K

.STACK: kích thước Stack khi có chương trình con

- Qui định kích thước Stack là 512byte(100h)
- Mặc định: 1K

.CODE: điểm bắt đầu đoạn mã chương trình

.DATA: điểm bắt đầu đoạn dữ liệu

.MAIN PROC

 ;thân chương trình

END MAIN

Tạo lập và chạy một chương trình

- o Dùng các phần mềm soạn thảo văn bản (SK, NC...) để tạo lập tệp văn bản chương trình gốc bằng hợp ngữ, đuôi tệp là ASM
- o Dùng chương trình dịch MASM (Microsoft Macro Assembly) hoặc TASM(Turbo...) để dịch tệp .ASM ra mã máy dưới dạng tệp .OBJ(Object)
- o Dùng chương trình LINK hoặc TLINK để kết nối các OBJ lại với nhau thành chương trình .EXE
- o Nếu chương trình viết ra để dịch ra kiểu .COM thì dùng chương trình EXE2BIN của DOS để dịch tiếp từ .EXE sang .COM
- o Chạy chương trình

Ví dụ: viết chương trình đưa ra màn hình lời chào "Hello"

4. Các Mode địa chỉ

Chế độ địa chỉ dùng để xác định toán hạng, bao gồm:

- ✓ Chế độ địa chỉ thanh ghi: toán hạng là thanh ghi
- ✓ Chế độ địa chỉ tức thì: toán hạng là hằng số
- ✓ Chế độ địa chỉ trực tiếp: toán hạng là biến nhớ

Chế độ địa chỉ gián tiếp thanh ghi

Địa chỉ offset của toán hạng được chứa trong một thanh ghi, thanh ghi đóng vai trò nh một con trỏ trỏ đến các ô nhớ

Khuông dạng toán hạng: [thanh ghi]

Thanh ghi có thể là BX, DI, SI hay BP với các thanh ghi BX, DI, SI số hiệu đoạn của toán hạng chứa trong DS, với thanh ghi BP số hiệu đoạn chứa trong SS.

Ví dụ: SI chứa địa chỉ offset 0100h và các từ nhớ tại địa chỉ 0100h có giá trị là 1234h. Khi thực hiện

```
MOV     AX,[SI] thì CPU kiểm tra SI để suy ra địa chỉ từ nhớ là DS:0100h sau đó chuyển nội dung của từ nhớ này(1234h) vào AX, nh vậy AX nhận giá trị 1234h
```

```
MOV     AX,SI thì AX nhận giá trị 100h
```

Ví dụ: tính tổng một mảng 10 phần tử, lu kết quả vào AX

```
XOR     AX,AX
```

```
LEA     SI,W
```

Tổng:

```
ADD     AX,[SI]
```

```
ADD     SI,2
```

```
LOOP   Tong
```

Chế độ địa chỉ cơ sở và chỉ số

Trong các chế độ địa chỉ này địa chỉ offset của các toán hạng nhận được bằng cách cộng 1 số được gọi là độ dịch với nội dung của 1 thanh ghi. Trong đó độ dịch có thể là:

địa chỉ offset của 1 biến

1 hằng số

địa chỉ offset của 1 biên cộng hoặc trừ với 1 hằng số

Toán hạng được viết:

[thanh ghi + độ dịch]

[độ dịch + thanh ghi]

[thanh ghi] + độ dịch

độ dịch + [thanh ghi]

độ dịch[thanh ghi]

các thanh ghi phải là BX, SI, DI hay BP

dùng BX, SI hay DI thì số hiệu đoạn chứa trong DS

dùng BP thì số hiệu đoạn chứa trong SS.

Chế độ địa chỉ đọc gọi là cơ sở nếu dùng BX hay BP, gọi là chỉ số nếu dùng SI hay DI.

Ví dụ: mảng W, thanh ghi BX=4

các lệnh sau là tương đương

MOV AX,W[BX]

MOV AX,[W+BX]

MOV AX,[BX+W]

MOV AX,W+[BX]

MOV AX[BX]+W

Ví dụ 2: cho mảng ALPHA DW 0123h,0456h,0789h,0ABCh

trong đoạn đánh địa chỉ bởi DS

BX chứa 2, offset 0002 chứa 1084h

SI chứa 4, offset 0004 chứa 2BACH

DI chứa 1

với các lệnh

Lệnh	Offset toán hạng	Số đọc chuyển
Mov ax,[alpha+bx]	Alpha+2	0456h
Mov bx,[bx+2]	2+2	2BACH
Mov cx,alpha[si]	Alpha+4	0789h
Mov ax,-2[si]	-2+4	1084h
Mov bx,[alpha+3+di]	Alpha+3+1	0789h

Ví dụ 3: thay thế chữ thường trong chuỗi thành chữ hoa

ASM:

MOV CX,N

XOR SI,SI

Lap:


```
CMP      MSG[SI],''
JE       Next
AND      MSG[SI],0DFH
```

Next:

```
INC      SI
LOOP    lap
```

Chương V: HỆ THỐNG VÀO RA

I. GIỚI THIỆU CHUNG

Hệ thống vào ra: trao đổi thông tin giữa máy tính và thế giới bên ngoài, bao gồm:

- ✓ Các modul vào ra (mạch ghép nối IO): ghép nối giữa CPU và bộ nhớ với các TBNV.
- ✓ Các TBNV: mạch ghép nối vào ra tổ chức thành các cổng vào ra sao cho mỗi cổng có một địa chỉ xác định.

Địa chỉ hoá cổng vào ra:

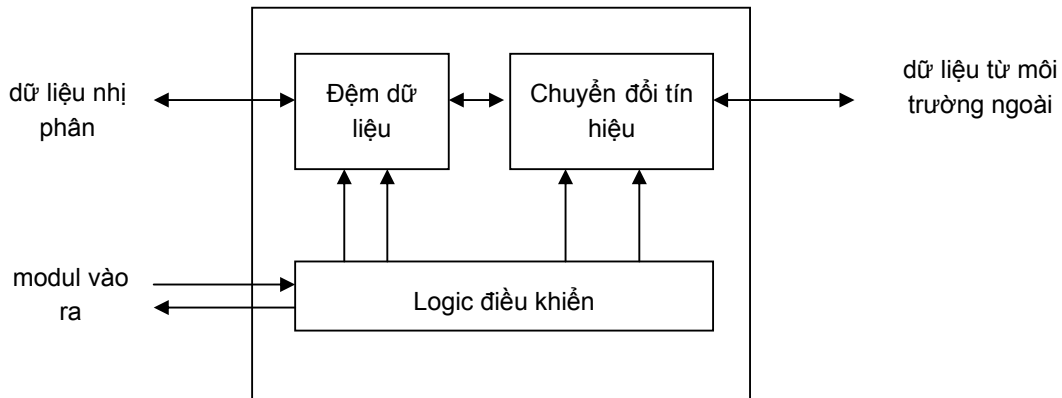
- ✓ Địa chỉ hoá tách biệt (vào ra trực tiếp): không gian địa chỉ cổng độc lập với không gian địa chỉ bộ nhớ.
- ✓ Địa chỉ hoá theo bản đồ bộ nhớ: không gian địa chỉ cổng vào ra nằm trong không gian địa chỉ bộ nhớ.

1. Các thiết bị ngoại vi

Chức năng:

- Trao đổi thông tin người - máy
- Trao đổi thông tin máy - máy

Cấu trúc



Logic điều khiển:

- ✓ Nhận tín hiệu điều khiển của CPU
- ✓ Phát tín hiệu điều khiển TBNV
- ✓ Phát tín hiệu trạng thái báo cho CPU biết trạng thái của TBNV

Đệm dữ liệu: chứa tạm thời dữ liệu trao đổi giữa TBNV và Modul vào ra

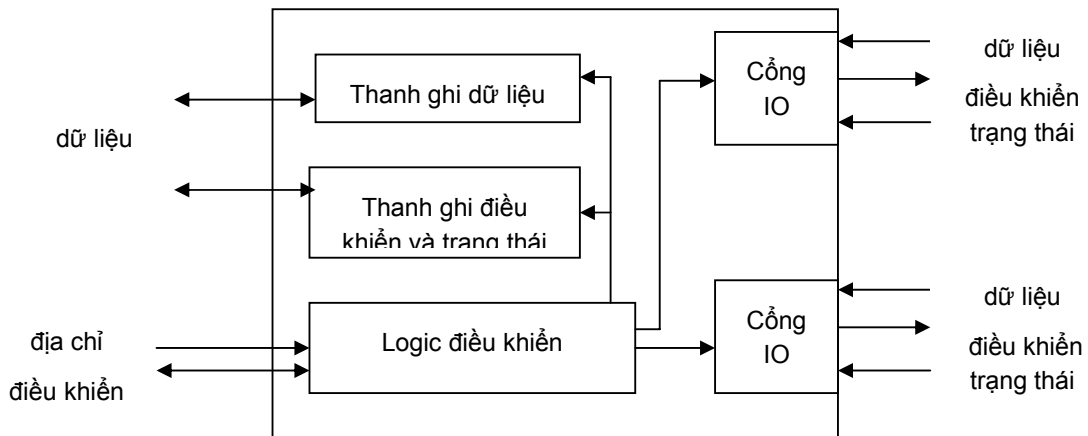
Chuyển đổi tín hiệu: chuyển tín hiệu ở dạng phi điện năng thành tín hiệu điện năng.

2. Modul vào ra

Chức năng:

- Điều khiển và định thời gian cho quá trình trao đổi
- Trao đổi thông tin với CPU
- Trao đổi thông tin với TBNV
- Đệm dữ liệu
- Phát hiện lỗi

Cấu trúc:



Người lập trình có thể can thiệp vào nội dung của các cổng và thanh ghi điều khiển trạng thái.

Nội dung thanh ghi trạng thái sẽ quyết định chế độ làm việc cho các cổng

II. GHÉP NỐI MÁY TÍNH VỚI THIẾT BỊ NGOẠI VI

Ghép nối giữa CPU và Modul vào ra: song song

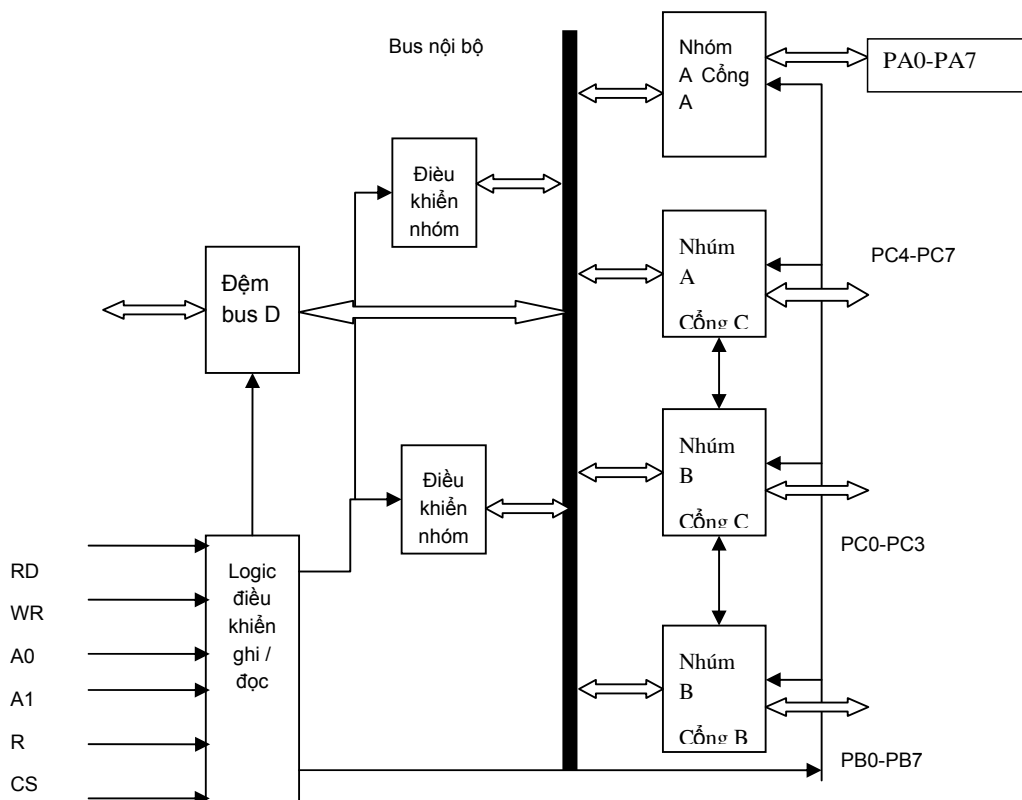
Ghép nối giữa Modul vào ra và TBNV: song song sẽ tạo ra cổng LPT, nối tiếp sẽ tạo ra cổng COM

1. Ghép nối song song

Nguyên tắc: Các cửa vào ra được ghép nối trực tiếp với bộ xử lí, ghép nối song song điều khiển bằng chương trình dùng mạch 8255A (PPI:Programable peripheral interface)

Mạch ghép nối vào ra song song lập trình được 8255A

a. Cấu trúc 8255A



Các chân tín hiệu của 8255A:

- ✓ D0-D7: Chân tín hiệu dữ liệu
- ✓ RD : chân tín hiệu yêu cầu đọc
- ✓ WR: chân tín hiệu yêu cầu ghi
- ✓ A0-A1: Các chân tín hiệu địa chỉ: chọn 4 thanh ghi bên trong 8255A

Thanh ghi từ điều khiển CWR, PA,PB,PC.

PA,PB,PC Là các thanh ghi đệm để ghi/đọc dữ liệu. Địa chỉ cho thanh ghi PA cũng là địa chỉ cơ sở của 8255.

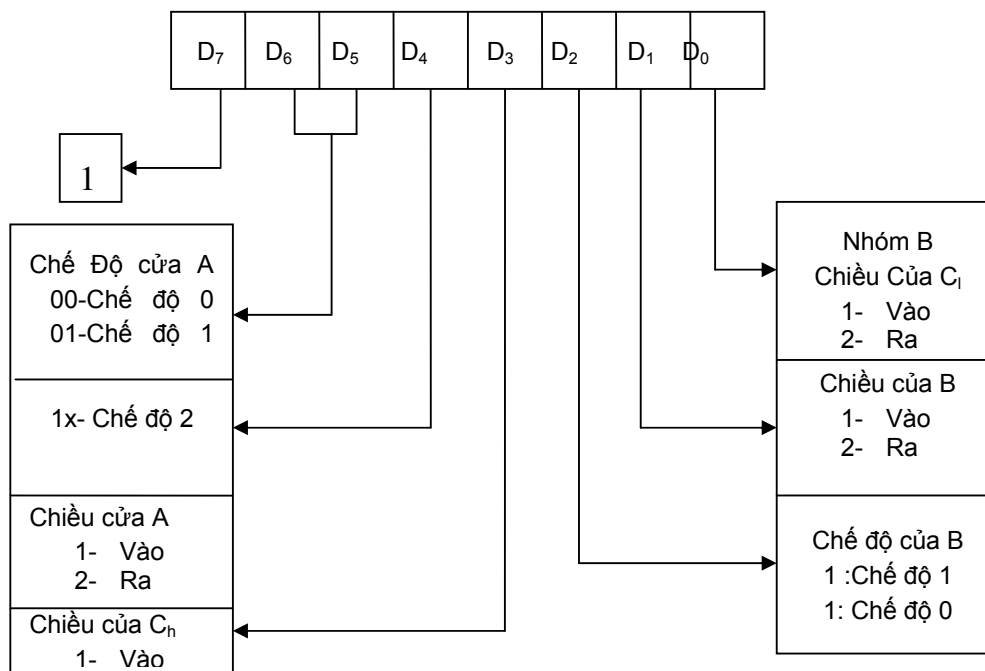
Có hai loại từ điều khiển cho 8255A:

- ✓ Từ điều khiển định nghĩa cấu hình cho các cổng PA,PB,PC.
- ✓ Từ điều khiển lập/xóa từng bit ở đầu ra của PC

Bảng chọn thanh ghi trong 8255A

CS	A1	A0	Chọn ra
1	X	X	Không chọn
0	0	0	PA
0	0	1	PB
0	1	0	PC
0	1	1	CWR

Từ điều khiển định nghĩa cấu hình



b. Các chế độ hoạt động của 8255

- Chế độ 0 (chế độ vào ra cơ sở):
 - ✓ Các cổng A, B, C được sử dụng độc lập nhau

- ✓ Các cổng A, B, C có thể là cổng vào hoặc cổng ra tùy giá trị từ điều khiển ghi trong thanh ghi từ điều khiển.

Không có sự đối thoại với VXL cũng như TBN. Nếu muốn có tín hiệu đối thoại phải dùng bit của cổng nào đó (thường là cổng C) để xác lập bit 1 và sau đó xoá về 0 bằng cách ghi số liệu (1 hoặc 0) hoặc bằng cách xác lập/ xoá một bit PC_i của cổng C bằng lệnh với D7 bằng 0.

- Chế độ 1 (chế độ vào/ ra đối thoại) với các bit cổng C.

Chia hai nhóm:

- ✓ Nhóm A gồm cổng A để trao đổi số liệu và nửa C cao ($PC_7- PC_4$) để đối thoại với VXL và TBN.
- ✓ Nhóm B gồm cổng B để trao đổi số liệu và nửa C thấp ($PC_3- PC_0$) để đối thoại với VXL và TBN.

Chiều và chế độ 1 của cổng A, B do từ điều khiển quyết định, các tín hiệu đối thoại PC_i phụ thuộc vào chiều cổng vào hay ra cho các tín hiệu PC_i . Ví dụ: (SGK)

- Chế độ 2 (vào ra có chốt):

Chế độ này chỉ dùng cho cổng A với vào ra thuận nghịch (hai chiều) và các bit $PC_3, PC_4- PC_7$ dùng làm tín hiệu đối thoại, trong đó:

- ✓ PC_3 cho tín hiệu yêu cầu ngắt $IRNT_A$ chung cho cả hai chiều và giống chế độ M=1.
- ✓ PC_4 cho tín hiệu vào STB_A khi cổng A có chiều vào (giống chế độ 1)
- ✓ PC_6 cho tín hiệu vào ACK_A khi cổng A có chiều ra (giống chế độ 1)

Chung cả hai chế độ 1, 2 các bit còn lại dùng làm đối thoại của cổng C đều là các tín hiệu ra.

Cổng B hoạt động giống chế độ 1 hoặc chế độ 0.

c. Ghép nối 8255A với MVT và TBN

Nguyên tắc chung:

- ✓ Phần ghép nối với máy vi tính (MVT)
- ✓ Phần ghép nối với TBN

d. Lập trình cho 8255A

2. Ghép nối nối tiếp

Nguyên tắc: cho phép trao đổi thông tin giữa CPU và TBNV theo từng bit, số liệu trao đổi thường được gửi theo các nhóm bit mà nó tạo thành một ký tự hay một từ.

Sử dụng:

- ✓ Khi TBNV cần trao đổi vốn đã là vào ra nối tiếp
- ✓ Khi khoảng cách giữa CPU và TBNV tương đối lớn.

Nhịp truyền: tổng số lần thay đổi tín hiệu trong một giây (baud rate)

Phương thức:

- ✓ Thời gian: đồng bộ, dị bộ
- ✓ Đường truyền: song công, đơn công, bán song công.

III. CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN VÀO RA

1. Vào ra điều khiển bằng cách thăm dò

Vấn đề điều khiển vào/ra dữ liệu sẽ trở thành rất đơn giản nếu thiết bị ngoại vi lúc nào cũng sẵn sàng chờ để làm việc với CPU. Ví dụ, bộ phận đo nhiệt số (như là một thiết bị vào) lắp sẵn trong một hệ thống điều khiển lúc nào cũng có thể cung cấp số đo về nhiệt độ của đối tượng cần điều chỉnh, còn một bộ đèn LED 7 nét (như là một thiết bị ra) dùng để chỉ thị một giá trị nào đó của một đại lượng vật lý nhất định trong hệ thống nói trên thì lúc nào cũng có thể biểu hiện thông tin đó. Như vậy, khi CPU

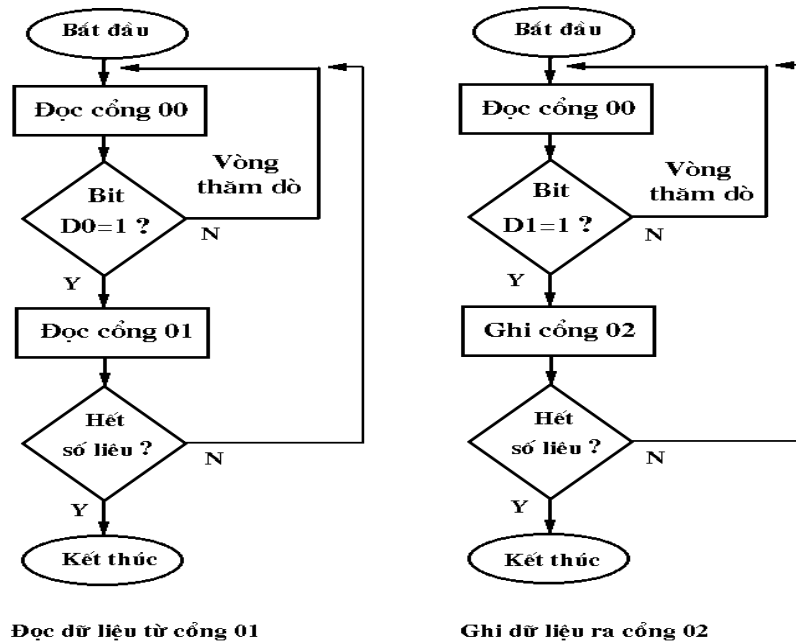
muốn có thông tin về nhiệt độ của hệ thống thì nó chỉ việc đọc cổng phối ghép với bộ đo nhiệt độ, và nếu CPU muốn biểu diễn thông tin vừa đọc được trên đèn LED thì nó chỉ việc đưa tín hiệu điều khiển tới đó mà không phải kiểm tra xem các thiết bị này có đang sẵn sàng làm việc hay không.

Tuy nhiên trong thực tế không phải lúc nào CPU cũng làm việc với các đối tượng "liên tục sẵn sàng" như trên. Thông thường khi CPU muốn làm việc với một đối tượng nào đó, trước tiên nó phải kiểm tra xem thiết bị đó có đang ở trạng thái sẵn sàng làm việc hay không, nếu có thì nó mới thực hiện việc trao đổi dữ liệu. Như vậy, nếu làm việc theo phương pháp thăm dò thì hệ thống thường là CPU phải được dành riêng cho việc trao đổi dữ liệu vì nó phải liên tục kiểm tra trạng thái sẵn sàng của thiết bị ngoại vi thông qua các tín hiệu móc nối (Handshake Signal). Các tín hiệu này được lấy từ mạch phối ghép, do người thiết kế tạo ra, để chương trình thăm dò hoạt động trên đó.

Ví dụ: Một cổng vào số 0 (Có địa chỉ 00) được dùng để đọc trạng thái sẵn sàng của 2 thiết bị ngoại vi nói trên. Tín hiệu sẵn sàng của thiết bị ngoại vi số 1 (cổng vào 01) được đặt vào bit D_0 , tín hiệu sẵn sàng của thiết bị ngoại vi số 2 (cổng vào 02) được đặt vào bit D_1 . Các thiết bị này sẽ có giá trị 1 khi thiết bị ngoại vi tương ứng ở trạng thái sẵn sàng làm việc với CPU và chúng sẽ được đưa vào Bus dữ liệu khi CPU đọc nó bằng lệnh đọc cổng vào số 0.

Mô tả hoạt động của phần mạch vào dữ liệu.

Khi thiết bị vào số 1 có 1 Byte số liệu cần trao đổi, nó đưa ra xung STB để cho phép mạch chốt 8 bit lấy Byte dữ liệu đồng thời kích cho mạch lật D (mạch tạo tín hiệu sẵn sàng) làm việc. CPU sẽ thăm dò trạng thái sẵn sàng của thiết bị vào số 1 qua bit D_0 khi nó đọc cổng D_0 . Đến khi CPU đọc 1 Byte dữ liệu vào thì nó đồng thời xoá luôn mạch tạo trạng thái sẵn sàng để chuẩn bị cho lần làm việc tới với 1 Byte dữ liệu khác.



2. Vào ra điều khiển bằng Ngắt

Ngắt và điều khiển ngắt

Trong cách tổ chức trao đổi dữ liệu thông qua việc thăm dò trạng thái sẵn sàng của thiết bị ngoại vi, trước khi tiến hành bất kỳ một cuộc trao đổi dữ liệu nào CPU phải để toàn bộ thời gian vào việc xác định trạng thái sẵn sàng làm việc của thiết bị ngoại vi. Trong hệ thống vi xử lý với cách làm việc như vậy, thông thường CPU được thiết kế chủ yếu chỉ là để phục vụ cho việc vào ra dữ liệu và thực hiện các xử lý liên quan. Trong thực tế CPU luôn có nhu cầu từ người dùng là tận dụng khả năng làm việc của CPU để làm thêm nhiều công việc khác nữa. Chỉ tới khi nào có yêu cầu trao đổi dữ liệu thì mới

yêu cầu CPU tạm dừng công việc hiện tại để phục vụ việc trao đổi dữ liệu. Sau khi hoàn thành việc trao đổi dữ liệu CPU sẽ quay về để làm tiếp công việc hiện đang bị gián đoạn. Cách làm này gọi là ngắt CPU để trao đổi dữ liệu.

Như vậy một hệ thống với cách hoạt động theo kiểu này có thể đáp ứng được rất nhanh các yêu cầu trao đổi dữ liệu trong khi vẫn có thể làm được các công việc khác. Muốn đạt được điều này, ta phải có cách tổ chức hệ thống sao cho có thể tận dụng được khả năng thực hiện các chương trình phục vụ ngắt tại các địa chỉ xác định của CPU. Vi mạch 8088 của CPU có các yêu cầu ngắt che được INTR và không che được NMI, chính các chân này sẽ được sử dụng vào việc đưa các yêu cầu ngắt từ bên ngoài đến CPU.

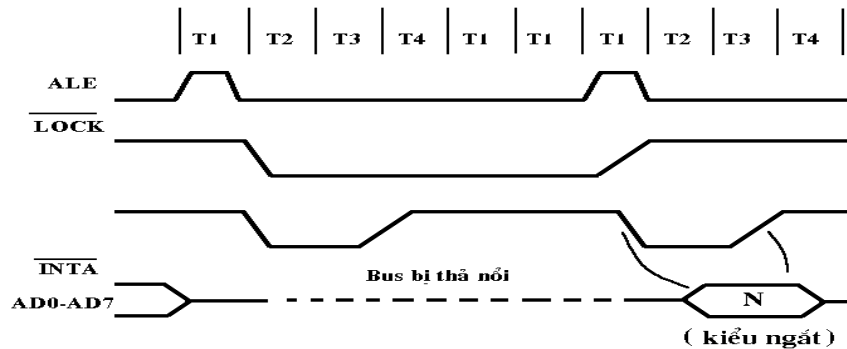
Các loại ngắt trong 8088

Trong hệ vi xử lý 8088 có thể xếp các nguyên nhân gây ra ngắt CPU vào 3 nhóm sau:

Nhóm các ngắt cứng: đó là các yêu cầu ngắt CPU do tín hiệu ngắt đến từ các chân INTR và NMI.

Ngắt cứng NMI là yêu cầu ngắt không che được, tương đương với ngắt INT2. Các lệnh CLI (xoá cờ IF) và STI (lập cờ IF) không có ảnh hưởng đến việc nhận biết tín hiệu yêu cầu ngắt NMI.

Ngắt cứng INTR là yêu cầu ngắt che được, các lệnh CLI và STI có ảnh hưởng trực tiếp tới trạng thái của cờ IF trong bộ vi xử lý, tức là ảnh hưởng tới việc CPU có nhận biết yêu cầu ngắt tại chân này hay không. Yêu cầu ngắt tại chân INTR có thể có kiểu ngắt N nằm trong khoảng 0 - FFh. Kiểu ngắt này phải được đưa vào Bus dữ liệu để CPU có thể đọc được khi có xung INTA trong chu kỳ trả lời chấp nhận ngắt.



Chu kỳ trả lời ngắt của CPU 8088.

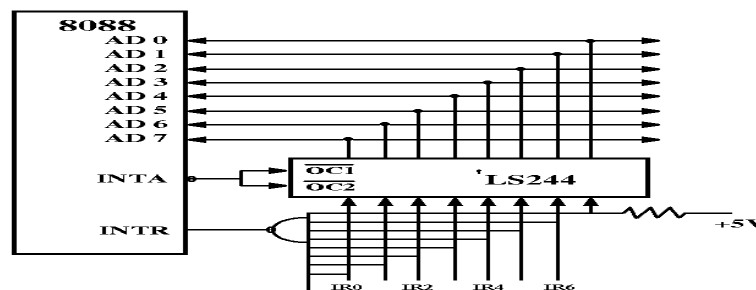
Nhóm các ngắt mềm: Khi CPU thực hiện các lệnh ngắt dạng INT N, trong đó N là số hiệu (kiểu) ngắt nằm trong khoảng 00-FFh (0-225).

Nhóm các hiện tượng ngoại lệ đó là: các ngắt do các lỗi nảy sinh trong quá trình hoạt động của CPU như phép chia cho 0, xảy ra tràn khi tính toán ...

Yêu cầu ngắt sẽ được kiểm tra thường xuyên tại chu kỳ đồng hồ cuối cùng của mỗi lệnh. Cách đơn giản để đưa được số hiệu ngắt N vào Bus dữ liệu trong khi cũng tạo ra yêu cầu ngắt đưa vào chân INTR của bộ vi xử lý 8088

Các ngắt khác:

Các ngắt xảy ra khi CPU phát hiện ra lỗi nhưng không khắc phục được lỗi đó.



Đưa số hiệu ngắt vào Bus dữ liệu

Giả thiết trong một thời điểm nhất định chỉ có một yêu cầu ngắt IR_i được tác động và khi đó ở đầu ra của mạch NAND sẽ có xung yêu cầu ngắt đến CPU. Tín hiệu IR_i được đồng thời đưa qua mạch khuếch đại đệm để tạo ra số hiệu ngắt tương ứng, số hiệu ngắt này sẽ được CPU đọc vào khi nó đưa ra tín hiệu trả lời INTA.

Quan hệ giữa IR_i và số hiệu ngắt N

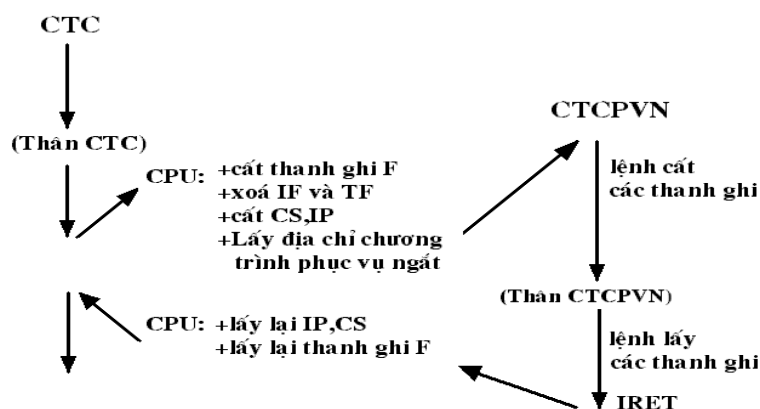
AD7	IR6	IR5	IR4	IR3	IR2	IR1	IR0	N
1	1	1	1	1	1	1	0	FEH(254)
1	1	1	1	1	1	0	1	FDH(253)
1	1	1	1	1	0	1	1	FBH(251)
1	1	1	1	0	1	1	1	F7H (247)
1	1	1	0	1	1	1	1	EFH(239)
1	1	0	1	1	1	1	1	DFH(233)
1	0	1	1	1	1	1	1	BFH(191)

Đáp ứng của CPU khi có yêu cầu ngắt

Khi có yêu cầu ngắt kiểu N đến chân CPU, và nếu yêu cầu đó được phép, CPU thực hiện các công việc sau:

1. $SP \leftarrow SP-2$, $SP \leftarrow FR$, trong đó SP là ô nhớ do SP chỉ ra (chỉ ra đỉnh mới của ngăn xếp, cắt thanh ghi cờ vào đỉnh ngăn xếp)
2. $IF \leftarrow 0$, $TF \leftarrow 0$.
(cấm các ngắt khác tác động vào CPU, cho CPU chạy ở chế độ bình thường).
3. $SP \leftarrow SP-2$, $SP \leftarrow CS$
(chỉ ra đỉnh mới của ngăn xếp, cắt phần địa chỉ đoạn của địa chỉ trở về vào đỉnh ngăn xếp)
4. $SP \leftarrow SP-2$, $SP \leftarrow IP$.
(chỉ ra đỉnh mới của ngăn xếp, cắt phần địa chỉ lệch của địa chỉ trở về vào đỉnh ngăn xếp).
5. $N*4 \rightarrow IP$, $N*4+2 \rightarrow CS$.
(lấy lệnh tại địa chỉ mới của chương trình con phục vụ ngắt kiểu N tương ứng trong bảng vector ngắt)
6. Tại cuối chương trình phục vụ ngắt, khi gặp lệnh IRET
 $SP \rightarrow IP$, $SP \leftarrow SP+2$
 $SP \rightarrow CS$, $SP \leftarrow SP+2$
 $SP \rightarrow FR$, $SP \leftarrow SP+2$
 (bộ vi xử lý quay lại chương trình chính tại địa chỉ trở về và với giá trị cũ của thanh ghi cờ được lấy ra từ ngăn xếp).

Về mặt cấu trúc chương trình, khi có ngắt xảy ra thì chương trình chính (CTC) liên hệ với chương trình con phục vụ ngắt (CTCPVN), điều này được mô tả trên hình dưới đây.



Trong thực tế các ngắt mềm INT N đã bao trùm các loại khác CPU bởi vì INTEL đã quy định một số kiểu ngắt đặc biệt được xếp vào đầu dãy ngắt mềm INT N:

INT 0: ngắt mềm do phép chia cho số 0 gây ra.

INT 1: Ngắt mềm để chạy từng lệnh ứng với trường hợp cờ TF = 1

INT 2: Ngắt cứng do tín hiệu tích cực tại chân NMI gây ra

INT3: Ngắt mềm để đặt điểm dừng của chương trình tại một địa chỉ nào đó

INT4 (hoặc lệnh INTO): Ngắt mềm ứng với trường hợp cờ tràn OF = 1

Các kiểu ngắt khác còn lại thì được dành cho INTEL và cho người sử dụng (IBM không hoàn toàn tuân thủ các quy định này khi chế tạo các máy PC/XT và PC/AT)

INT5 – INT 1FH: Dành riêng cho INTEL trong các bộ vi xử lý cao cấp khác.

INT20 – INT FFH: dành cho người sử dụng.

Các kiểu ngắt N trong INT N đều tương ứng với các địa chỉ xác định của CTPVN mà ta có thể tra được trong bảng các véc tơ ngắt. INTEL quy định bảng này nằm trong RAM bắt đầu từ địa chỉ 00000H và dài 1KB (8088 có tất cả 256 kiểu ngắt, mỗi kiểu ngắt ứng với một vectơ ngắt, mỗi véc tơ ngắt cần có 4 Byte để chứa địa chỉ đầy đủ cho CS : IP của CTPVN).

Bảng vectơ ngắt của 8088 tại 1 KB RAM đầu tiên được chỉ ra như hình dưới đây:


03FEH-03FFH	CS Cửa CTPVN INT FFH
03FCH-03FDH	IP Cửa CTPVN INT FFH
0082H-0083H	CS Cửa CTPVN INT 20H
0080H-0081H	IP Cửa CTPVN INT 20H
000AH-000BH	CS Cửa CTPVN INT 2
0008H-0009H	IP Cửa CTPVN INT 2
0006H-0007H	CS Cửa CTPVN INT1
0004H-0005H	IP Cửa CTPVN INT1
0002H-0003H	CS Cửa CTPVN INT0
0000H-0001H	IP Cửa CTPVN INT0

Xử lý ưu tiên khi ngắt

Một yêu cầu quan trọng đặt ra là, đòi hỏi CPU phải có khả năng xử lý được các yêu cầu ngắt nếu tại cùng một thời điểm có nhiều yêu cầu ngắt thuộc các loại ngắt khác nhau, đòi hỏi CPU thực hiện phục vụ. Vấn đề giải quyết theo cách sau: CPU sẽ xử lý các yêu cầu ngắt theo thứ tự ưu tiên với nguyên tắc ngắt nào có mức ưu tiên cao nhất sẽ được CPU nhận biết và phục vụ trước.

Thông thường ngay từ khi chế tạo CPU 8088 có khả năng phân biệt các mức ưu tiên khác nhau cho các loại ngắt (theo thứ tự từ cao xuống thấp) như sau:

	Mức ưu tiên
+Ngắt nội bộ:INT0(phép chia cho 0),INT N,INT0	...cao nhất
+Ngắt không che được NMI	
+Ngắt che được INTR	
+Ngắt để chạy từng lệnh INT1	...thấp nhất



Mạch điều khiển ưu tiên ngắt 8259A (PIC: Priority interrupt control)

các khối chức năng chính của PIC: (bổ sung)

3. Vào ra điều khiển bằng DMA

Khác với trao đổi tin theo chương trình do VXL điều khiển theo từng lệnh vào ra giữa thanh ghi chứa AX của VXL với cửa vào-ra của TBN, trao đổi tin DMA do khối điều khiển trao đổi trực tiếp khối nhớ DMAC điều khiển. Đây là phương pháp trao đổi tin nhanh, cho một lượng lớn tin trực tiếp giữa khối nhớ M và cửa vào ra TBN, không qua VXL. Phương pháp này thường dùng để đưa tin từ khối nhớ ra màn hình hoặc trao đổi giữa khối nhớ và đĩa từ. Khối điều khiển DMAC có thể thiết kế chế tạo bởi vi mạch rời hoặc IC lớn như 8237, 82C37.

Thủ tục trao đổi tin DMA

a. Yêu cầu trao đổi tin DMA của các TBN

a) Nhược điểm của phương pháp trao đổi tin theo CT.

Muốn trao đổi tin của Khối nhớ M và TBN nào đó ta cần:

- ✓ Đưa địa chỉ khối nhớ
- ✓ Phát lệnh đọc/ghi khối nhớ để trao đổi với thanh ghi chứa AX của VXL.
- ✓ Đưa địa chỉ của cửa nối với TBN
- ✓ Phát lệnh trao đổi tin (IN/OUT) giữa thanh ghi chứa AX với cửa vào-ra của TBN. Như vậy phải có 4 lệnh và trao đổi tin giữa khối nhớ và TBN phải thông qua thanh ghi chứa của VXL. Thời gian trao đổi tin lớn vì:
- ✓ VXL phải giải mã lệnh và thực hiện lệnh.
- ✓ Trao đổi thông qua thanh ghi chứa AX trung gian.

b) Yêu cầu trao đổi tin nhanh.

Trong hệ MVT có hai thiết bị là màn hình và đĩa từ đòi hỏi trao đổi một lượng tin lớn trong thời gian nhỏ. do đó phải dùng giải pháp cứng để điều khiển sự trao đổi tin (không dùng phần mềm). Đặc điểm của giải pháp DMA này là:

Khối DMAC hoàn toàn thay thế VXL để điều khiển sự trao đổi tin, nói cách khác DMAC hoàn toàn dành quyền sử dụng bus trong quá trình trao đổi.

VXL bị cô lập, bị treo không hoạt động, tức trạng thái điện trở cao, không liên hệ bus bên ngoài.

Khi trao đổi tin xong DMAC lại trả lại quyền sử dụng Bus cho VXL.

b. Thủ tục trao đổi tin DMA.

a) Thủ tục:

DMAC điều khiển sự trao đổi DMA giữa khối nhớ M và TBN, không có sự tham gia của VXL (bị treo hay cô lập đường dây) theo thủ tục có trình tự sau:

TBN đưa yêu cầu DRQ cho DMAC. DMAC ghi nhận (nếu chưa có ghi che, chắn trước), xét thứ tự ưu tiên (nếu có nhiều yêu cầu DRQ vào đồng thời) và đưa yêu cầu cho VXL, đề nghị chiếm giữ đường dây bởi một trong hai tín hiệu sau:

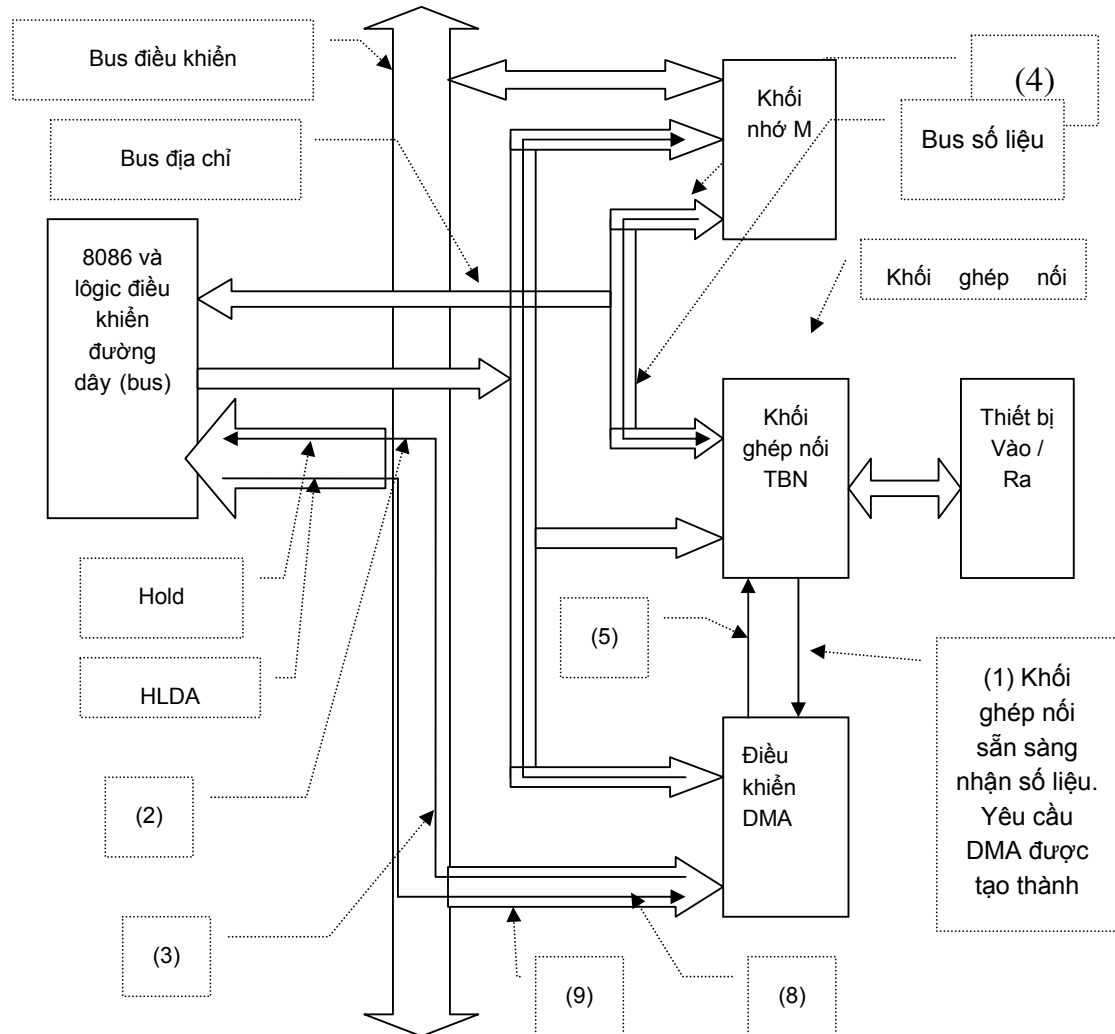
- ✓ Tín hiệu Hold (cho các VXL 8085, 8086 ở chế độ MIN)
- ✓ Tín hiệu RQ₀ (ở chế độ MAX của 8086)

VXL hoàn thành lệnh đang thực hiện, ngắt chương trình để chuyển sang chương trình con khởi phát DMA (ghi địa chỉ ban đầu khối nhớ, số lời trao đổi, hướng thay đổi địa chỉ) và đưa ra tín hiệu xác nhận DMA (GT₀, GT₁ hay HLDA) và tự treo ở trạng thái điện trở cao (cô lập) để nhường quyền sử dụng BUS cho DMAC.

DMAC tiến hành:

- ✓ Đưa tín hiệu xác nhận DACK cho TBN
- ✓ Tiến hành trao đổi DMA cho tới khi kết thúc (đếm lời trao đổi trở về 0)
- ✓ Kết thúc tín hiệu HOLD, trả quyền điều khiển bus cho VXL
- ✓ VXL nhận biết sự kết thúc tín hiệu HOLD và kết thúc tín hiệu HLDA, dành lại quyền điều khiển Bus.

b) Chuỗi hành động của DMAC trong trao đổi tin DMA



Chuỗi hành động của DMAC gồm khối ghép nối và khối điều khiển được thực hiện để đảm bảo việc trao đổi tin. Chuỗi hành động này tuân theo thứ tự:

1. Khối ghép nối (KGN) gửi khối điều khiển một yêu cầu (DRQ) cho phục vụ DMA.
2. KDK gửi yêu cầu Hold tới VXL
3. KĐK nhận xác nhận điều khiển bus (HLDA, GT_0) từ VXL
4. KĐK phát địa chỉ lên bus (từ thanh ghi đệm địa chỉ) cho khối nhớ.
5. KĐK phát xác nhận DMA cho TBN.
6. KĐK phát lệnh đọc (ghi) để trao đổi số liệu giữa khối nhớ và khối ghép nối.
7. KGN:

Chốt số liệu (khi ghi) để trao đổi với TBN. Hai hành động 6 và 7 có thể trao đổi với nhau nên ghi số liệu từ khối ghép nối vào khối nhớ thông qua thanh ghi đệm của KGN.

Thanh ghi địa chỉ tăng lên 1.

Thanh ghi đếm lời giảm đi 1. Nếu nội dung thanh ghi đếm lời này chưa bằng 0, lặp lại các bước 6, 7 để trao đổi với các lời tin khác.

8. KGN kết thúc tín hiệu yêu cầu HOLD
9. VXL kết thúc tín hiệu HLDA để dành lại quyền chiếm bus từ DMAC

3. Các chế độ trao đổi DMA

- a) Trao đổi tin khối : trao đổi nhiều (khối) lời tin lần lượt từ giá trị đếm lời tin n tới 0 (hết)
- b) Trao đổi lấy nén chu kỳ từng phần: DMAC phát hiện đường dây bus rỗi (VXL không sử dụng đường dây bus) thực hiện trao đổi DMA.

DMAC phải có :

Thiết bị phát hiện đường dây rỗi

Thiết bị bảo đảm VXL bị treo cho tới khi DMAC không sử dụng bus, khiến VXL chờ một thời gian T_w tới khi DMAC thực hiện trao đổi xong một phần của trao đổi tin và tiếp tục nốt ở phần lấy lên chu kỳ sau cho tới khi kết thúc trao đổi tin khối tin DMA.

- c) Lấy lên chu kỳ trong suốt: Chế độ này cũng giống chế độ trên là lấy lên chu kỳ nhưng bắt VXL chờ với T_w lớn hơn cho tới khi trao đổi xong một khối tin trọn vẹn.

Khối điều khiển DMAC

1. Nhiệm vụ của khối

2. Cấu trúc khối DMAC

3. Vi mạch 8237,8257

Chương VI: MÀN HÌNH

I. Những khái niệm cơ bản

Đối với các máy tính hiện nay, phương tiện đối thoại chủ yếu người máy là bàn phím và màn hình. Màn hình là phương tiện hiển thị thông tin thuận lợi và kinh tế. Thông tin hiển thị là chữ, số (text), hoặc đồ họa (graphic).

Có hai loại màn hình đang được dùng phổ biến là: màn hình tia âm cực (CRT- Cathode ray tube) và màn hình tinh thể lỏng (LCD-Liquid Crystal Display).

Lại CRT thông dụng hơn do giá thành thấp và có khả năng hiển thị thông tin phong phú, đồng thời việc điều khiển CRT đã do các mạch LSI đảm nhiệm.

Màn hình LCD có ưu điểm nhẹ mỏng, tiêu thụ ít năng lượng nhưng có giá thành cao.

1. Nguyên lý của phương pháp hiển thị hình ảnh video.

Khả năng phân giải hữu hạn của mắt người

Khả năng phân giải của mắt người là khoảng 1'(góc 1 phút), nghĩa là nếu chúng ta nhìn 2 điểm dưới một góc nhỏ hơn 1' thì sẽ cảm nhận thấy chúng dính vào nhau, góc đó còn được gọi là góc phân giải. Điều này hết sức quan trọng đối với việc hiển thị thông tin, vì chúng ta chỉ cần hiển thị một số hữu hạn điểm của màn hình ở một khoảng cách nào đó, chúng ta vẫn có cảm giác hình ảnh là mịn.

Hiện tượng lưu ảnh trên võng mạc

Khi một hình ảnh hiện rồi lại tắt với tần số lớn hơn 25lần/giây, mắt người không nhận ra được sự nhấp nháy đó và có cảm giác hình ảnh tồn tại liên tục. Đó là hiện tượng lưu ảnh trên võng mạc.

2. Những đặc điểm chung của màn hình

Điểm ảnh pixel.

Điểm ảnh pixel là phần tử nhỏ nhất của một ảnh hay một thiết bị hiển thị ảnh. Kích thước một điểm ảnh trên màn hình CRT phụ thuộc vào các tham số

- ✓ Kích thước chùm tia điện tử.
- ✓ Kích thước hạt phốt pho.
- ✓ Chiều dày lớp phốt pho.

Đối với màn hình màu, kích thước một điểm ảnh gần bằng kích thước của ba điểm màu: xanh lục, đỏ, xanh nước biển.

Kích thước ngang và dọc với đơn vị là 1 điểm ảnh được gọi là **kích thước màn hình**. Màn hình VGA cơ bản có kích thước 640x480 điểm ảnh.

Độ phân giải

Độ phân giải được định nghĩa là kích thước chi tiết nhỏ nhất và đo được của một thiết bị hiển thị. Một tham số để đo độ phân giải là số điểm ảnh trên một đơn vị chiều dài (inch hay centimét), được gọi là mật độ điểm ảnh. Mật độ điểm ảnh viết tắt là dpi(dot per inch).

Độ sáng (brighness)

Độ sáng là giá trị phát sáng tương đối của vật liệu so với một vật liệu màu trắng chuẩn. Độ phát sáng của màn hình phát sáng như ống tia âm cực được coi là độ sáng.

Độ tương phản

Là tỉ lệ giữa độ sáng hay độ phát sáng giữa hai trạng thái đóng và mở của phần tử hiển thị (điểm ảnh). Độ tương phản cho biết khả năng phân biệt hai phần tử này .

Độ sâu màu

Một màu bất kỳ có thể biểu diễn qua 3 màu cơ bản: đỏ, xanh lục, xanh nước biển tùy theo độ đậm nhạt (gray scale). Độ sâu màu là số màu có thể hiển thị được cho một điểm ảnh. Tùy theo số bit dùng để hiển thị màu ta phân loại màn hình theo màu như sau:

- ✓ Đen trắng 1 bit (2 màu)
- ✓ Màu CGA 4 bit (16 màu)
- ✓ Màu giả (pseudo color) 8 bit (256 màu)
- ✓ High color 16 bit
- ✓ True color 24 bit.

Tần số làm tươi

Tần số làm tươi chính là tốc độ quét màn hình

$$f_{it}=30\text{Hz đến }60\text{Hz.}$$

II. Màn hình màu CRT (Cathod Ray Tube)

Màn hình ống tia âm cực CRT (Cathod Ray Tube) là màn hình cổ điển và thông dụng nhất hiện nay.

1. Cấu tạo

Màn hình màu thực chất gồm 3 ống hình đơn sắc chung trong một vỏ gồm:

3 sợi đốt để điều khiển 3 chùm tia khác nhau (đỏ, xanh lơ, lục)

- ✓ Màn hình phốt pho phải được quét thành từng nhóm 3 vạch thẳng đứng (tương ứng với 3 màu) thay cho các điểm.
- ✓ Thẳng theo các vạch là một mặt nạ bằng kim loại có các lỗ (hay các khe dọc) để cho 3 chùm tia điện tử đi qua tạo thành 3 điểm riêng rẽ gần nhau tạo thành 3 chấm sáng: R, G, B, có 3 núm điều chỉnh độ sáng, tối ứng với 3 màu trên.

Vị trí đặt các màu cho ta công nghệ tương ứng:

- ✓ Các chấm sáng theo hàng ngang: PIL (Precision In Line)
- ✓ Các chấm sáng theo tam giác đều: Trinitron.

Sơ đồ cấu tạo CRT màu

(hình vẽ)

Bộ phận súng điện tử

Gồm 3 súng điện tử có nhiệm vụ bắn ra 3 chùm tia điện tử mảnh, chuyển động đủ nhanh bay lên đập vào một điểm của lớp huỳnh quang trên màn hình, làm cho điểm đó phát sáng. Các bộ phận chính của súng điện tử:

Ca-tốt: catốt của CRT khi được đốt nóng sẽ phát xạ ra một đám mây điện tử.

- ✓ Các điện cực gia tốc cho chùm tia điện tử. Điện cực Anot nằm trên màn hình có tác dụng gia tốc cho chùm tia điện tử. Hiệu điện thế đặt lên 2 cực từ 14 nghìn V đến 25.000V.
- ✓ Hệ thống các điện cực hội tụ chùm tia điện tử (G2 và G3) có tác dụng làm cho chùm tia điện tử hội tụ lại thành một tia rất mảnh. Ngoài ra chúng cũng có tác dụng tăng tốc cho chùm tia điện tử.

Hệ thống lái tia:

- ✓ Việc lái tia điện tử được thực hiện bằng điện trường hoặc từ trường. Thông thường việc lái tia điện tử được thực hiện bằng từ trường. Gần cổ đèn hình người ta đặt 2 cặp cuộn dây song song với nhau và cho các dòng điện có dạng biến thiên phù hợp, có tần số nhất định chạy qua, dòng điện sẽ tạo ra từ trường biến thiên theo phương trùng với trục của cuộn dây.. Trong lòng đèn hình có 2 vectơ từ trường theo phương nằm ngang và thẳng đứng, chúng tác động lên chùm tia điện tử, làm lệch hướng theo quy luật của dòng điện trong các cặp cuộn dây.

Điều chế dòng tia điện tử

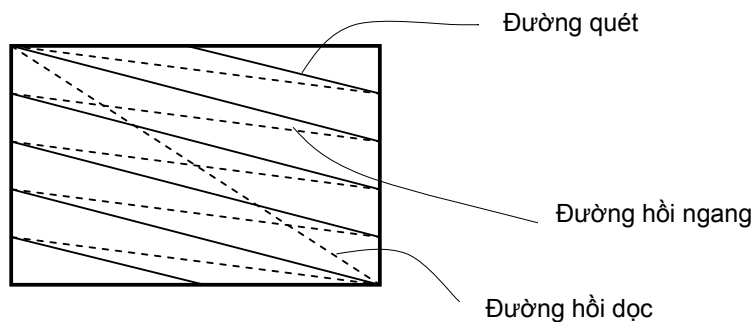
Khi súng điện tử quét lần lượt các dòng của màn, nếu làm thay đổi cường độ của chùm tia điện tử theo qui luật thay đổi của tín hiệu hình ảnh nơi phát đi, thì sẽ tái tạo lại hình ảnh trên màn hình. Việc điều khiển dòng điện tử theo quy luật của hình ảnh cần hiển thị được gọi là điều chế dòng điện tử.

Điện cực lưới G1 làm nhiệm vụ lưới điều khiển, nếu đặt giữa G1 và catốt một điện áp thay đổi theo quy luật thay đổi tín hiệu của hình ảnh thì dòng điện tử của súng điện tử sẽ bị điều chế theo quy luật của tín hiệu.

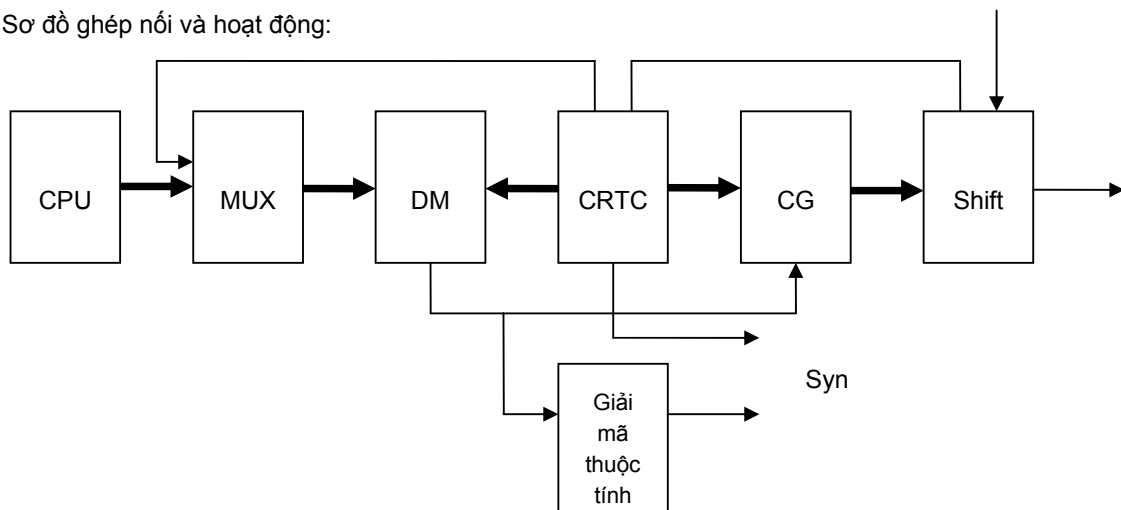
2. Phương pháp quét dòng

Dựa vào độ phân giải hữu hạn của hiện tượng lưu ảnh trên võng mạc của mắt người, người ta đã xây dựng nên phương pháp quét dòng để hiển thị hình ảnh.

Màn hình được chia thành một số hữu hạn dòng, tập các dòng tạo nên hình ảnh.



Sơ đồ ghép nối và hoạt động:



CRTC (CRT Controller): đơn vị điều khiển màn hình. Số kiệu sẽ được thể hiện trên màn hình từ bộ nhớ màn hình hay CPU gửi qua bộ tạo chữ CG để:

- ✓ Điều khiển kiểu và vị trí con trỏ màn hình
- ✓ Định chế độ dòng, màn hình và số ảnh trên một giây

DM (Display Memory): ghi thông tin sẽ được thể hiện trên màn hình trong chế độ Text.

CG (Character General): lưu trữ các mẫu bit của kí tự, các Font chữ trong chế độ Text.

Bộ Shift:

Nhận số liệu từ CG và đẩy ra tín hiệu Video

Kết hợp với bộ giải mã thuộc tính để tạo ra các tín hiệu cho súng RGB

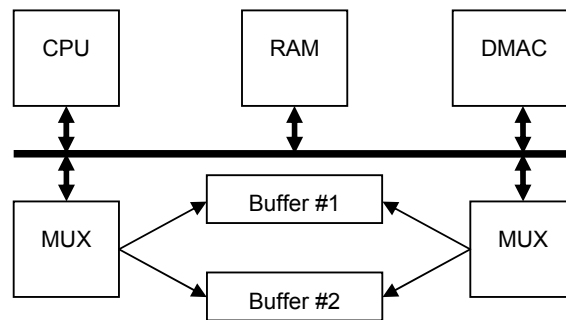
Kết hợp với các tín hiệu đồng bộ từ CRTIC để đưa ra tín hiệu hỗn hợp.

Kĩ thuật làm tươi hình ảnh

Làm tươi (Refresh): quá trình hiện hình ảnh lặp đi lặp lại liên tục với một tần suất đủ lớn để mắt người không cảm thấy lặp loè.

Nguyên tắc: chuyển nội dung bộ nhớ lên màn hình dùng phương pháp DMA

Sơ đồ:



Khi DMAC nạp số liệu vào bộ đệm Buffer #1 thì bộ đệm Buffer #2 đã được nạp đầy trước đó đẩy nội dung của màn hình ra bộ tạo chữ CG, sau đó đến lượt Buffer #1. Quá trình làm tươi được tiếp tục với sự đổi hướng của hai bộ dồn kênh MUX.

Chương VII: BÀN PHÍM

1. Khái niệm

Bàn phím là một thiết bị ngoại vi dùng để giao tiếp giữa người và máy tính (dùng để nhập số liệu, chương trình hoặc ra lệnh cho máy ...).

Chức năng chung của bàn phím:

- ✓ Phát hiện sự ấn phím
- ✓ Khử rung
- ✓ Mã hoá phím

Phân loại: bàn phím được chia làm 3 loại chính:

Bàn phím ký tự: dùng để đưa chữ cái, chữ số, dấu hiệu và một số ký tự điều khiển vào máy tính.

Bàn phím số: chuyên dùng để đưa số vào máy tính. Thường được ghép cùng bàn phím ký tự (bên phải bàn phím hiện đại).

Bàn phím đặc nhiệm: dùng cho các thiết bị điều khiển tự động (có sử dụng bộ vi xử lý...).

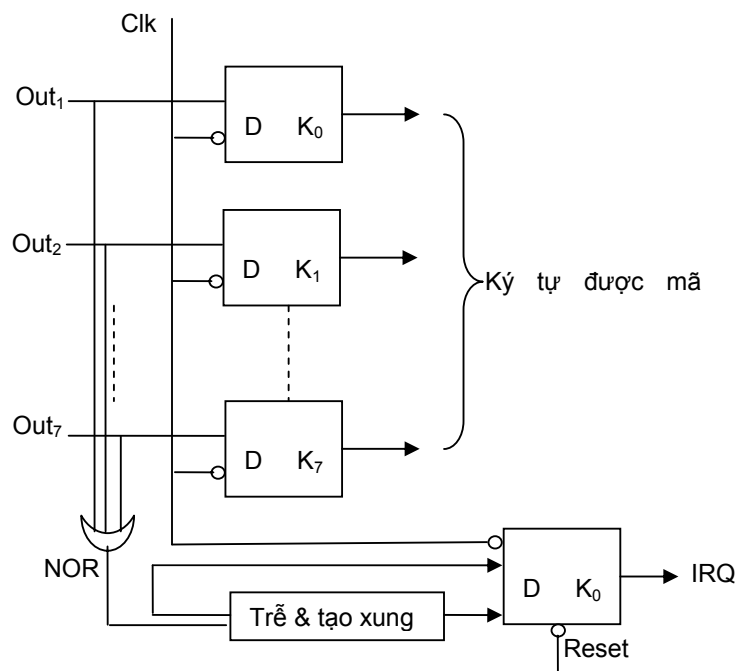
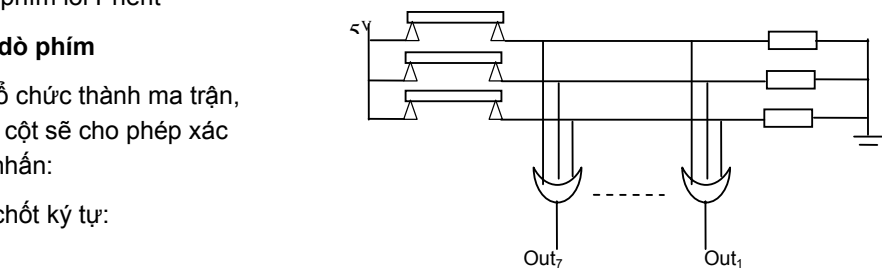
Công nghệ:

- ✓ Điện dung: hai trạng thái của phím ấn khác nhau về điện dung (hình vẽ)
- ✓ Hiệu ứng Hall (hình vẽ)
- ✓ Bàn phím quang điện
- ✓ Bàn phím lõi Pherit

2. Kỹ thuật dò phím

Bàn phím được tổ chức thành ma trận, giao của hàng và cột sẽ cho phép xác định phím được nhấn:

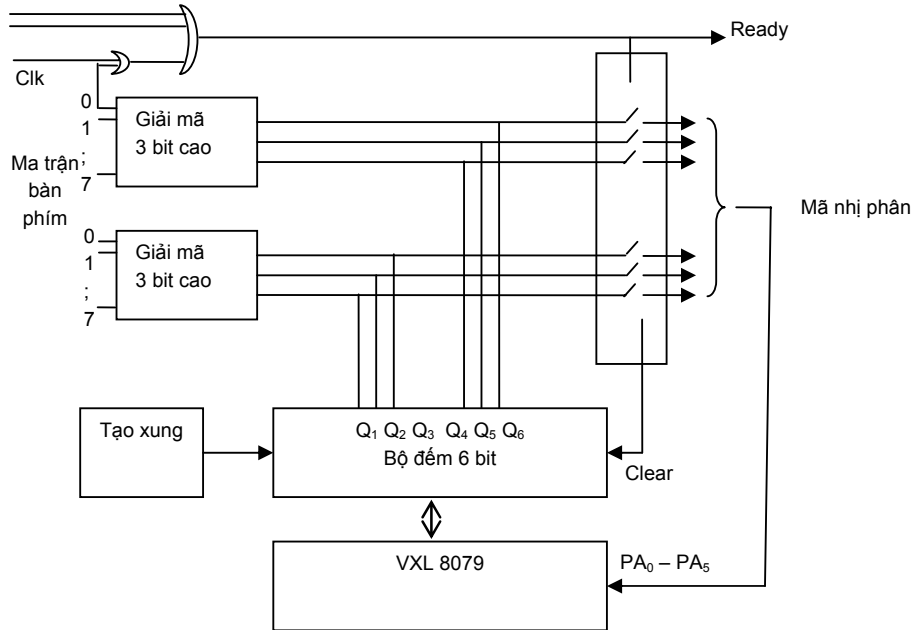
Mạch phối ghép chốt ký tự:



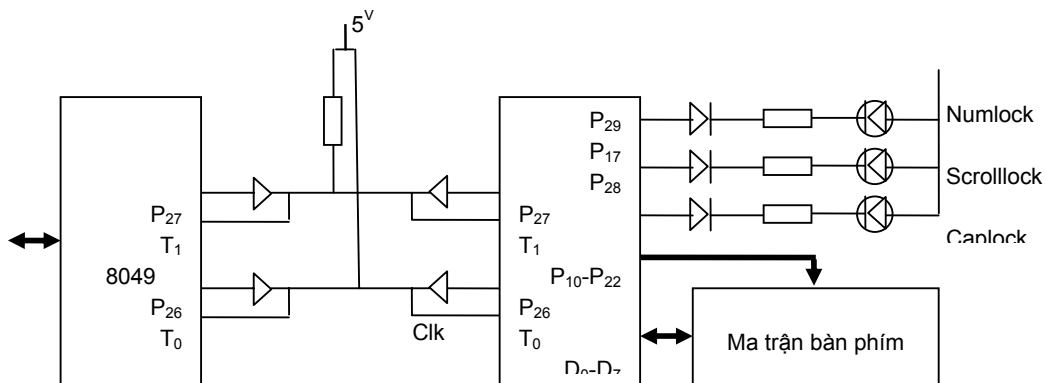
Kĩ thuật quét phím (Scan)

Xem xét trạng thái của phím theo hàng, cột. Thời gian quét phím nhỏ hơn nhiều so với thời gian nhấn một phím.

Sơ đồ: (bàn phím 64 phím)



Bàn phím 101 phím:



Mỗi khi có một phím được nhấn, gửi về ma trận mã quét của phím bằng một ngắt IRQ và CPU sẽ đọc cổng 60h để biết tác động phím nào đã xảy ra.

ROMBIOS sẽ chuyển đổi mã quét đó thành từ mã dài hai byte: byte thấp chứa mã ASCII, byte cao chứa mã quét của phím, sau đó đặt mã hai byte này vào hàng đợi bàn phím.

Các mã hai byte sau khi được lưu trong bộ đệm sẽ chờ cho tới khi có một chương trình nào đó lấy dần ra xử lí.

Với các tổ hợp phím nóng, chức năng tổ hợp được coi như một lệnh và được xử lí tức thì. Riêng tổ hợp phím CTRL + ALT + DEL luôn được kiểm tra khi có tác động của phím gửi về CPU.

Chương VIII Ổ ĐĨA

Bộ nhớ ngoài: lưu trữ dữ liệu và chương trình của người sử dụng

Đặc điểm:

- ✓ Dung lượng lớn
- ✓ Tính lưu động cao, tiện dụng
- ✓ Tốc độ truy xuất thấp

Phân loại: đĩa từ, quang

1. Đĩa từ (Magnetic)

Đĩa từ: tấm tròn mỏng, trên có phủ một lớp oxit sắt từ

- ✓ Đĩa mềm: tấm tròn bằng nhựa
- ✓ Đĩa cứng: tấm tròn bằng kim loại

Các hạt sắt từ có khả năng

- ✓ Thẩm từ (Permeable): có khả năng cho từ thông xuyên qua
- ✓ Trữ từ (Retentivity): lưu lại từ tính

Tham số:

- ✓ Đọc ghi
- ✓ Đĩa

Tham số đọc ghi (Đầu từ)

Nguyên tắc: nam châm điện

- Ghi từ: Dòng trên cuộn dây A-B tạo ra từ trường xác định trong lõi hình khuyên. Qua khe hở từ thông của từ trường đi xuyên xuống lớp oxit sắt từ sắp xếp lại các hạt chất sắt từ của lớp oxit sắt chạy qua khe hở đầu từ theo một hướng nhất định => ghi thông tin lên đĩa
- Đọc: sự thay đổi chiều sắp xếp các phần tử từ dọc theo đường ghi sẽ tạo nên sự thay đổi chiều của từ trường trong lõi đầu từ thông qua khe hở đầu từ, sinh ra dòng điện cảm ứng trong cuộn dây AB. Dòng trên cuộn dây AB sẽ mang thông tin đã được ghi trên đĩa chuyển động theo hình vành khuyên => đọc được các thông tin đó.

Các phương pháp mã hoá số liệu ghi trên đĩa:

Quá trình đọc/ghi thông tin trên đĩa đòi hỏi phải có sự đồng bộ, các dữ liệu cần ghi cũng phải được mã hoá, điều chế và giải điều chế (gắn thông tin vào vật mang tin)

Các phương pháp điều chế về cơ bản dựa vào các đặc trưng của tín hiệu

Tín hiệu hình sin chuẩn $s(t) = A \cos(2\pi f_c t + \varphi)$

Có các đặc trưng:

- ✓ Biên độ A
- ✓ Tần số f_c
- ✓ Góc pha φ

Các phương pháp điều chế

- ✓ Điều biên AM (Amplitude Modulation)
- ✓ Điều tần FM (Frequency Modulation)
- ✓ Điều pha PM (Phase Modulation)

Phương pháp điều chế sử dụng chủ yếu khi mã hoá - điều chế số liệu ghi trên đĩa từ: FM

Chia trực thời gian thành các khoảng thời gian bằng nhau gọi là ô bit

Với phương pháp FM khoảng cách giữa 2 xung đồng hồ là 1 ô bit, ở giữa ô bit xung số liệu được ghi.

- Có xung số liệu: bit dữ liệu là 1
- Không có xung số liệu: bit dữ liệu là 0 (Hình vẽ)

Như vậy, 1 byte số liệu ghi trên đĩa bao gồm giá trị thực của byte số liệu đó và giá trị FFh của byte đồng hồ.

Tuy nhiên, phương pháp FM lãng phí bộ nhớ (độ dư thừa thông tin tới 50%) => đưa ra phương pháp MFM (Modifier FM)

- ✓ Xung số liệu được ghi ở giữa mỗi ô bit
- ✓ Xung đồng hồ chỉ được ghi ở đầu mỗi ô bit nếu trong số bit này và ô bit trước đó bit số liệu là 0.

Tham số đĩa từ

Một số khái niệm

Mặt đĩa (Size): mỗi mặt tương ứng có một đầu đọc (Head)

Rãnh từ (Track): các đường tròn đồng tâm được đánh số từ ngoài vào trong (bắt đầu từ rãnh số 0)

Cung từ (Sector): mỗi rãnh được chia làm nhiều cung với Microsoft OS: dung lượng 1 sector thường là 512 byte.

Thông tin 1 cung

- ✓ Trường địa chỉ (ID)
- ✓ Số liệu
- ✓ Tín hiệu đồng bộ

Liên cung (Cluster): tập hợp của 2, 4, 6... cung từ, các cung được đánh số tuần tự nhưng 1 sector không nhất thiết phải kề với sec 2 mà được truy xuất qua các móc nối.

Từ trụ (Cylinder): các rãnh từ có cùng số thứ tự trên các đĩa từ (chỉ có ở đĩa cứng)

Cấu trúc các vùng thông tin trên đĩa

- ✓ Vùng hệ thống
- ✓ Vùng dữ liệu

(Hình vẽ)

MBR (Master Boot Record): Boot chính của đĩa (chỉ có ở đĩa cứng):

- ✓ Tham số đĩa
- ✓ Thông tin về hệ thống Format đĩa

BS (Boot Sector): liên cung khởi động (với đĩa chứa dữ liệu: để trống)

Boot Directory (Bảng thư mục gốc)

Dãy các mục vào (Entry), mỗi mục vào tương ứng với 1 thư mục con hay tệp tin có trên đĩa (Với DOS 6.22 tối đa 512 mục vào)

Thông tin mục vào

<i>Ý nghĩa</i>	<i>Độ dài</i>	<i>Số hiệu</i>
Tên tệp, thư mục	8 byte	0

Phần mở rộng	3 byte	1
Thuộc tính	1 byte	2
Đề dành	10 byte	3
Giờ tạo lập	2 byte	4
Ngày tạo lập	2 byte	5
Địa chỉ bên cung đầu tiên	2 byte	6
Kích thước	4 byte	7
Tổng	32 bite	

Cấu trúc: Danh sách móc nối

Riêng với số hiệu 0

- ✓ 00h: Chưa sử dụng
- ✓ 20h: Thư mục (.)
- ✓ 2020h: Thư mục (..)
- ✓ E5h: đã bị xoá

FAT (File Allocation Table): bảng định vị tập tin

- ✓ Quản lý danh sách các liên cung dùng lưu trữ cho tập tin
- ✓ Danh sách các liên cung còn rỗi (chưa sử dụng)
- ✓ Các liên cung bị lỗi (Bad Sector)

Mỗi bảng FAT tương ứng với 1 ổ logic, kích thước ổ logic phụ thuộc vào số bit dùng cho mỗi bảng FAT

- ✓ FAT -12: Số liên cung quản lý được: 2^{12} liên cung
- ✓ FAT -16: Số liên cung quản lý được: 2^{16} liên cung
- ✓ FAT -32: Số liên cung quản lý được: 2^{32} liên cung

Ví dụ:...

Các công nghệ sản xuất đĩa từ

Đĩa mềm

- ✓ Tốc độ quay: 360 vòng/phút
- ✓ Tốc độ truy xuất: 500Kb/s -1 MB/s
- ✓ Giao diện ISA, SCSI (Small Computer Systems Interface)
- ✓ Nguồn điện: +5v -> +12v
- ✓ Cáp dữ liệu/điều khiển: 34 pin (chân cắm) dây 1: màu đỏ
- ✓ Cáp đảo ngược (pin 10 -> pin 16): phân biệt ổ A-B

Đĩa cứng

- ✓ Tốc độ quay: 7200 vòng/phút hoặc hơn
- ✓ Tốc độ truy xuất: 1 Mb/s -> 5 Mb/s

Giao diện:

Chuẩn ST 506 (Seagate Technology):

- Các tín hiệu điều khiển 34 dây
- Tín hiệu số liệu: 20 dây
- Công nghệ MFM loại RLL (Run Length limited)
- Số Sec/Track: 17 -> 26

- Tốc độ 1 Mb/s

Chuẩn ESDI (Enhanced Small Device Interface):

- ✓ Tín hiệu tương đương ST 506
- ✓ Số Sec/Track: 34 -> 36
- ✓ Tốc độ 10 Mb/s

Chuẩn SCSI (Small Computer Device Interface):

- ✓ Tín hiệu: 50 dây hoặc 68 dây
- ✓ Tốc độ có thể lên tới 5 Mbyte/s

Chuẩn IDE (Interlligent Device Electronic):

- ✓ Tín hiệu : 40 dây (hoặc 44 dây) tùy theo các chuẩn ATA (AT Attachment)
- ✓ Tốc độ có thể lên tới 4 Mbyte/s

Chuẩn bị một đĩa cứng để đưa vào sử dụng

Định dạng cấp thấp (Low Level Format): định dạng 1 lần ngay khi chế tạo.

Phân khu đĩa (Partion): Đĩa vật lý có thể được phân thành các phân khu độc lập như 1 ổ logic. Các phân khu:

- ✓ DOS chính (Primary DOS partion)
- ✓ DOS mở rộng (Extended DOS partion)
- ✓ Phi DOS (Non- partion DOS)

Thông tin mỗi phân khu:

- ✓ Địa chỉ vật lý đầu
- ✓ Địa chỉ vật lý cuối
- ✓ Địa chỉ logic cuối
- ✓ Số Sec/phân khu

Thực hiện: FDISK (Fixed Disk)

Định dạng cấp cao (High Level Format)

- ✓ Tạo các phân vùng hệ thống, dữ liệu trên đĩa
- ✓ Đánh dấu các vùng đĩa lỗi

Thực hiện: FORMAT

2. Đĩa Quang (Optical Disk)

Đặc điểm:

- ✓ Mật độ ghi thông tin cao
- ✓ Dung lượng lớn
- ✓ Giá thành: thấp
- ✓ Tốc độ truy xuất: nhỏ hơn đĩa cứng

Nguyên tắc đọc/ghi thông tin

Ghi:

- Các đĩa CDROM được tạo bằng cách dùng 1 tia lazer mạnh đốt chảy các hốc đường kính 1 mm trên 1 đĩa chủ, từ đĩa chủ này tạo ra một khuôn để tạo ra các bản copy trên các đĩa chất dẻo.
- Sau đó phủ 1 lớp nhôm mỏng lên trên mặt đĩa và lấp chất dẻo trong suốt lên trên lớp nhôm để bảo vệ

- Các hốc nhỏ: pit, diện tích không bị đốt: land; pit và land có độ phản xạ khác nhau => phân biệt dữ liệu
- Thông tin trên CDROM được ghi theo một đường xoắn tròn ốc duy nhất từ tâm đĩa ra ngoài. Dữ liệu được ghi theo nhóm 24 byte. Giữa các nhóm có thêm các bit đặc biệt và 1 byte đồng bộ để tạo thành 1 Frame. 96 Frame -> khối (2Kb)

Đọc thông tin:

- Đầu dò (Detector): đo năng lượng phản xạ từ bề mặt đĩa khi chiếu lên bề mặt 1 tia laser công suất nhỏ.
- Dữ liệu được đọc với vận tốc 75inch/s tương đương 153.60 Kbyte/s

Phân loại:

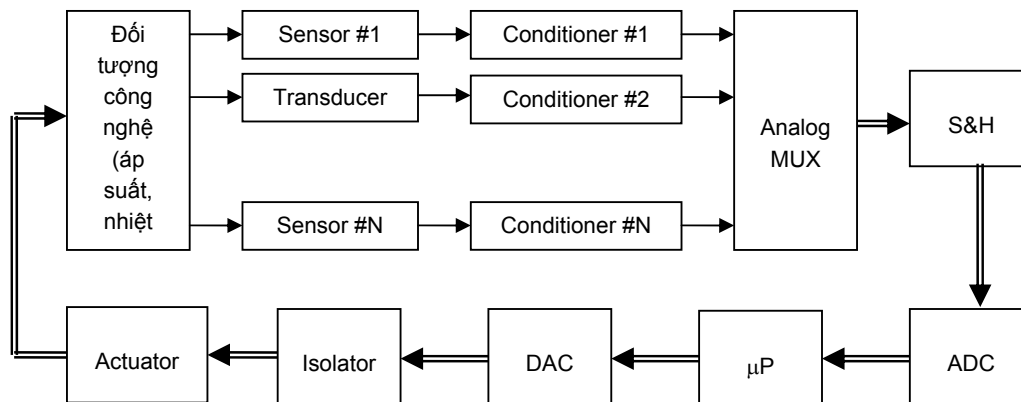
- CD-ROM (Compact Disk) – Read only Memory với định dạng dùng cho âm thanh, dữ liệu.
- WORM: Write Only-Read Multiple
- CD-RW: CD -Read/Write: bề mặt được được bao phủ bởi lớp polycarbon với 5 lớp cho phép đọc/ghi dữ liệu.
- DVD (Digital Video Disk): dung lượng lưu trữ lớn cho dữ liệu số..

Chương IX: THIẾT BỊ GHÉP NỐI VÀ TRUYỀN THÔNG

1. Bộ chuyển đổi tín hiệu

Máy tính: xử lí tín hiệu số (Digital): bit, byte,...song trong thực tế các dữ liệu tồn tại ở dạng tương tự (Analog): nhiệt độ, độ ẩm, độ dài,...âm thanh, hình ảnh,...vì vậy việc ứng dụng máy tính trong thực tế đòi hỏi phải có sự chuyển đổi các tín hiệu ở dạng tương tự sang tín hiệu số (ADC) và ngược lại (DAC).

Quá trình ghép nối:



Trong đó:

- Các Sensor: các thiết bị, vật liệu dùng biến đổi các đại lượng, các giá trị vật lí khác thành giá trị điện hay gần với điện.
- Các Transducer: thiết bị chuẩn hoá tín hiệu để tín hiệu ra là tín hiệu có giới hạn xác định.
- Conditioner : các tín hiệu từ các Sensor là các tín hiệu nhỏ, phi tuyến nên dùng các bộ khuếch đại (OpAmp) dùng bù phi tuyến và nâng mức tín hiệu cho phù hợp với giá trị đầu vào ADC.
- Analog MUX: bộ dồn kênh 2^n đầu vào một đầu ra.
- S&H (Sample and Hold): lấy mẫu tín hiệu và trích một phần tín hiệu
- Ghép với hệ thống thu thập tín hiệu biến thiên nhanh mà ADC có thời gian chuyển đổi lớn.
- Thu hẹp các cửa sổ bất định của ADC thành các cửa sổ bất định S&H.
- Isolator, Actuator: cách li bằng biến áp xung quang học.

Bộ chuyển đổi tín hiệu số - tương tự: DAC (Digital Analog Converter)

Nguyên tắc: biến đổi các mã số trực tiếp ra dòng điện hay điện áp.

$$E_{\text{Out}} = E_{\text{in}}(B_1 \cdot 2^{-1} + B_2 \cdot 2^{-2} + \dots + B_n \cdot 2^{-n})$$

n: số bit

B_i : các bit.

Đặc điểm

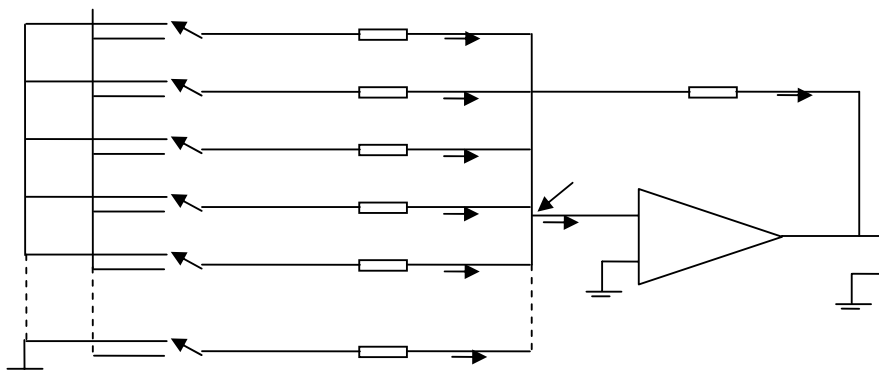
Lượng ra không liên tục

Giá trị ra lớn nhất khi tất cả các bit là 1

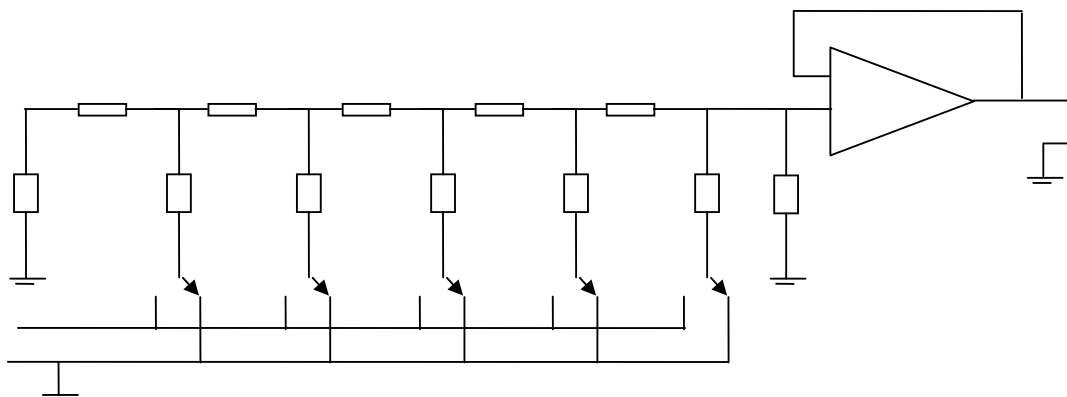
Sự thay đổi của điện áp hay dòng ứng với sự thay đổi của LSB.

Các loại DAC:

DAC thang chia nhị phân



DAC thang chia R-2R



Bộ chuyển đổi tín hiệu tương tự - số: ADC (Analog Digital Converter)

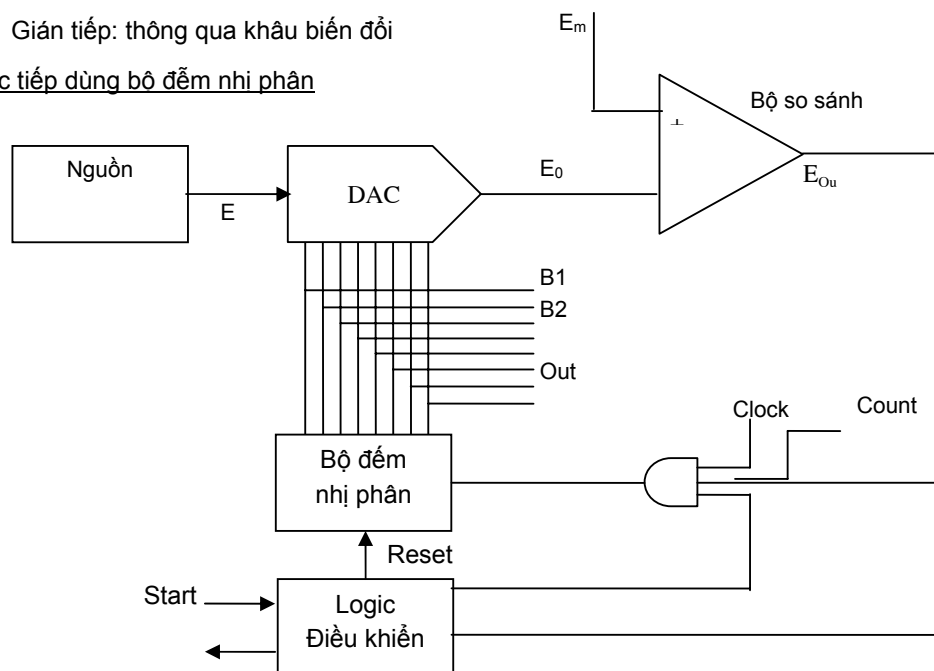
Nguyên tắc: rời rạc hoá và mã hoá tín hiệu

Phân loại:

Trực tiếp: từ giá trị mã điện áp sang mã xung

Gián tiếp: thông qua khâu biến đổi

ADC trực tiếp dùng bộ đếm nhị phân

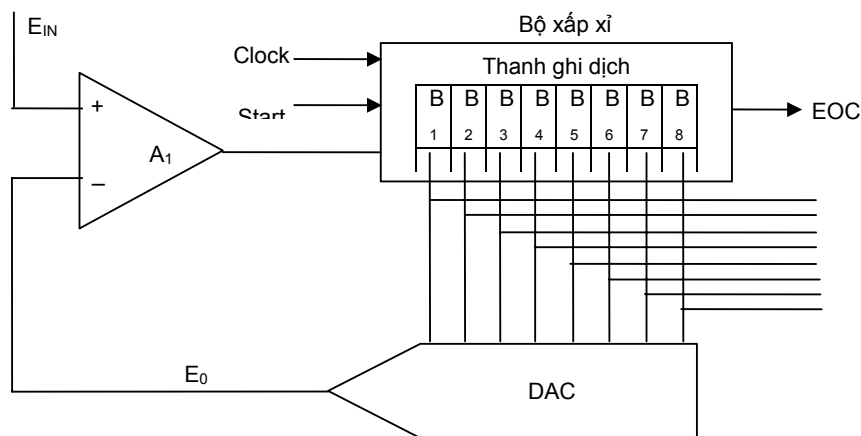


Khi có tín hiệu xoá bộ đếm $E_0 = 0$.

Khi $E_0 < E_{IN}$ thì tín hiệu ra Analog là 1, tăng bộ đếm lên 1

Khi $E_0 > E_{IN}$ thì tín hiệu ra là 0, xoá bộ đếm

ADC gián tiếp dùng bộ xấp xỉ (Successive approximation)



2. Modem (Modulation - Demodulation) điều chế và giải điều chế

Thiết bị

- Chuyển đổi tín hiệu: số \rightarrow tương tự và tương tự \rightarrow số
- Điều chế và giải điều chế tín hiệu

CCITT cho phép sử dụng các Modem vào việc truyền số liệu quốc tế.

Về nguyên lý: đáp ứng 2 yêu cầu tham số:

- Lưu lượng thông tin
- Phần tử mạng

Thao tác:

- Tự động quay số (Auto dial): gọi một Modem khác theo chế độ xung hoặc đa tần.
- Tự động trả lời (Auto answer)
- Làm ngắt quá trình kết nối với đường truyền điện thoại khi cuộc truyền dữ liệu đã hoàn tất hay có lỗi.
- Tự động thích ứng tốc độ giữa 2 Modem
- Chuyển đổi các bit sang dạng tín hiệu thích hợp với đường truyền điện thoại
- Chuyển đổi tín hiệu tương tự số và ngược lại

Phân loại:

- Modem trong: card mở rộng với khe cắm ISA
- Modem ngoài: bản mạch đóng hộp

Phương thức:

- Đồng bộ: khôi phục lại tín hiệu đồng bộ ở bộ phận nhận
- Không đồng bộ: sử dụng các bit start, stop

Các tiêu chuẩn dùng cho Modem

- V32 bit: 14,4 Kb/s

- V22 bit: 2,4 Kb/s
- V17: 14,4 bit/s
- V27: 4,8 Kb/s
- Bell: 1,2 Kb/s

Các thanh ghi trên Modem

Thanh ghi trạng thái S: cất giữ các tham số khi cài đặt

- S₀: số tiếng chuông để bắt đầu trả lời tự động
- S1: đếm số chuông gọi đến
- S2: ký tự escape
- S3: trở lại đầu dòng
- S4: về đầu dòng
- S5: back space

Thông số một số chuẩn Modem

Tên	Tốc độ (b/s)	Điều biến
V21	300	FSK
V22	1.200	PSK
V22 bit	2.400	ASK/PSK
V27	4.800	PSK
V29	9.600	PSK
V32	9.600	ASK/PSK
V32 bit	14.400	ASK/PSK
V34	28.800	ASK/PSK

3. Các chuẩn giao tiếp

Các chuẩn chung: hầu hết các thiết bị xử lý tín hiệu có khả năng truyền nhận tín hiệu hạn chế, thông thường các thiết bị này được gắn trực tiếp với các thiết bị chuyển nhận tín hiệu hoặc qua mạng, chúng được gọi là các thiết bị truyền nhận dữ liệu đầu cuối (DTE, DCE).

Mỗi thiết bị xử lý tín hiệu (trạm) thường được kết hợp với một cặp gồm một DTE và một DCE.

Hai trạm truyền tín hiệu cho nhau qua hai DCE của mỗi bên được kết nối với nhau. Hai DCE trao đổi tín hiệu với nhau trên mạng hoặc đường truyền phải tương tự nhau, nghĩa là bộ phận nhận tín hiệu bên này phải tương ứng với bộ phận phát tín hiệu của bên kia.

DTE và DCE truyền nhận tín hiệu với nhau do đó cũng phải tương thích với nhau về dữ liệu và thông tin điều khiển: các chuẩn

Các chuẩn về giao diện giữa DTE và DCE bao gồm:

- ✓ *Chuẩn về cấu trúc*: xác định kết nối vật lý giữa DTE và DCE (tín hiệu và mạch điều khiển thông qua cáp nối và giắc cắm)
- ✓ *Chuẩn về tín hiệu*: xác định mức hiệu điện thế, thời gian biến đổi tín hiệu
- ✓ *Chuẩn về chức năng*: xác định chức năng các mạch chuyển đổi
- ✓ *Chuẩn về thủ tục*: xác định thứ tự thao tác trong truyền dữ liệu dựa trên chuẩn chức năng của các đường tín hiệu.

Chuẩn EIA-RS 232 (Electronic Industry Association – Recomend Standard): chuẩn giao tiếp truyền thông công nghiệp

EIA đã công bố tiêu chuẩn RS-232C với nỗ lực nhằm tạo ra khả năng để ghép nối các thiết bị do nhiều nhà sản xuất làm ra mà không đòi hỏi có một tiêu chuẩn kỹ thuật đặc biệt cho từng trường hợp.

Ý tưởng để xây dựng tiêu chuẩn RS-232 là phải sử dụng cùng loại nối dây, thí dụ loại đầu nối 25 chân hoặc 9 chân, được nối theo cùng một cách và sử dụng cùng mức điện áp khi biểu diễn các số nhị phân 1 và 0 tương ứng. Với ý tưởng này, nếu như mọi người đều tham gia vào tiêu chuẩn theo cùng một cách thì có thể nối các thiết bị với cổng RS-232 của các hãng khác nhau, các mẫu mã khác nhau mà không cần có thêm điều kiện nào. Các môdem, các máy in và nhiều thiết bị khác có thể được nối vào giao diện RS-232.

Ngày nay, hầu hết các máy tính đều trang bị một hoặc hai cổng nối tiếp RS-232, và tất cả đều có khả năng sử dụng RS-232, ít nhất là như một khả năng tùy chọn từ nhà sản xuất máy tính hoặc từ phía người sử dụng máy tính.

Các đặc trưng điện

Các mức điện áp đường truyền

Trong RS-232B, mức logic '1' là một điện áp bất kỳ, trong phạm vi từ -5 V đến -25 V , trong khi logic '0' là bất cứ điện áp nào trong khoảng từ $+5\text{ V}$ đến $+25\text{ V}$. Các mức điện áp trong phạm vi -3 V đến $+3\text{ V}$ là trạng thái chuyển tiếp, trong khi các phạm vi từ $\pm 3\text{ V}$ đến $\pm 5\text{ V}$ không được xác định và dẫn đến các kết quả không thể dự tính trước nếu như được sử dụng: tình trạng này đã xuất hiện trong các hệ thống được thiết kế sơ sài.

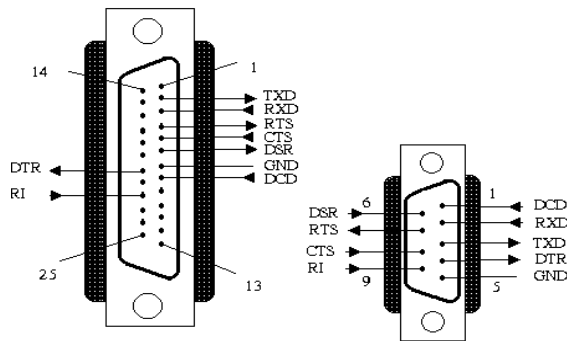
Các đặc trưng điện của tiêu chuẩn RS-232 quy định cụ thể điện áp cực tiểu và cực đại của mức logic '1' và '0'. Mức điện áp bằng 0 V ở bộ nhận, được hiểu như việc đường truyền bị đứt hoặc xảy ra chập mạch.

Trong chuẩn RS-232C, để có được tốc độ truyền dữ liệu nhanh hơn người ta đã sử dụng khoảng chênh lệch hẹp hơn giữa mức logic 0 và logic 1. Các giới hạn trên đối với mức logic 0 và logic 1 là $\pm 12\text{ V}$, chứ không dùng giới hạn $\pm 25\text{ V}$ như trong chuẩn RS-232B. Nếu không có các xung xuất hiện trên đường dẫn thì mức điện áp tương đương với mức HIGH, tức là -12 V .

Các yêu cầu về mặt điện được quy định trong chuẩn RS-232C như sau:

- Mức logic 1 (mức dấu) nằm trong khoảng: -3 V đến -12 V ; trong đó khoảng từ -5 V đến -12 V là tin cậy, mức logic 0 (mức trống) nằm trong khoảng: $+3\text{ V}$ đến $+12\text{ V}$, khoảng từ $+5\text{ V}$ đến $+12\text{ V}$ là tin cậy.
- Trở kháng tải về phía bộ phận của mạch phải lớn hơn 3.000Ω nhưng không được vượt quá 7.000Ω .
- Tốc độ truyền/ nhận dữ liệu cực đại là 100 kbit/giây .
- Các lối vào của bộ nhận phải có điện dung phải nhỏ hơn 2.500 pF .
- Độ dài của cáp nối giữa máy tính và thiết bị ghép nối qua cổng nối tiếp không thể vượt quá 15 m nếu không sử dụng môdem.
- Các giá trị tốc độ truyền dữ liệu chuẩn là 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9.600, 19.200, 28.800, ..., 56.600 baud.

Đầu nối trên máy tính PC.



Nhờ việc quy định thống nhất sử dụng một đầu nối 25 chân và về sau đã bổ sung thêm đầu nối 9 chân cho cổng nối tiếp RS-232, cụ thể hơn là ổ cắm về phía dây cáp còn ổ cắm về phía máy tính, mà tất cả các sản phẩm đều tương thích với nhau. Quy định này cũng áp dụng thống nhất cho các thiết bị ghép nối với cổng RS-232. Hình trên chỉ ra cách sắp xếp chân của đầu nối 25 chân và 9 chân dùng cho RS-232C, còn việc định nghĩa chức năng của các chân được liệt kê ở bảng kế tiếp.

Tiêu chuẩn RS-232C quy định rõ việc sử dụng đầu nối thống nhất để tất cả các sản phẩm đều tương thích với nhau. Vì vậy thứ tự và chức năng của các chân đã được quy định rất cụ thể và phải tuân thủ một cách nghiêm ngặt. Để dễ dàng nhận ra thứ tự các chân, bên cạnh các chân đều có in rõ số thứ tự trên phần nhựa của phích cắm cũng như ổ cắm. Nhận xét này cần được lưu ý khi kiểm tra cáp nối hoặc tự hàn một cáp mới.

Các chân và chức năng trên đầu nối 25 chân và 9 chân.

25 chân	9 chân	Tên	Viết tắt	Chức năng
				Chú ý: =>: Lỗi vào <=: Lỗi ra
1	-	Frame Ground (Đất - vỏ máy)	FG	Chân này thường được nối với vỏ bọc kim của dây cáp, với vỏ máy, với đai bao ngoài đầu nối hoặc đất thực sự.
2	3	Transmit Data (Truyền dữ liệu)	TXD <=	Dữ liệu được gửi từ DTE (máy tính hoặc thiết bị đầu cuối) tới DCE qua đường dẫn TD.
3	2	Receive Data (Nhận dữ liệu)	RXD =>	Dữ liệu được nhận từ DCE tới DTE (máy tính hoặc thiết bị đầu cuối) qua RD.
4	7	Request to Send (Yêu cầu gửi)	RTS <=	DTE đặt đường này lên mức hoạt động khi sẵn sàng tham gia cuộc truyền dữ liệu.
5	8	Clear to Send (Xóa để gửi)	CTS =>	DCE đặt đường này lên mức hoạt động để thông báo cho DTE là phải sẵn sàng nhận dữ liệu.
6	6	Data Set Ready (Dữ liệu sẵn sàng)	DSR =>	Tính hoạt động giống với CTS nhưng được kích hoạt bởi DTE khi nó sẵn sàng nhận dữ liệu.
7	5	Signal Ground (Đất của tín hiệu)	SG	Tất cả các tín hiệu được so sánh với đất tín hiệu (GND).
8	1	Data Carrier Detect	DCD =>	Phát hiện tín hiệu mang dữ liệu.
20	4	Data Terminal Ready (Đầu cuối dữ liệu sẵn sàng)	DTR <=	Tính hoạt động giống với đường dẫn RTS nhưng được kích hoạt bởi DCE khi muốn truyền dữ liệu.

22	9	Ring Indicate (Báo chuông)	RI =>	Chỉ cho thấy là DCE đang nhận tín hiệu rung chuông.
----	---	-------------------------------	----------	---

2. Mạch điều khiển truyền số liệu

Để thực hiện các phương pháp truyền một cách cụ thể, các nhà chế tạo đã cung cấp một loạt các IC chuyên dùng, các IC này chính là phần cứng thuộc lớp vật lí trong một hệ thống thông tin, chúng hoạt động theo nguyên tắc của kĩ thuật số và vì vậy chế độ truyền đồng bộ hay bất đồng bộ phụ thuộc vào việc sử dụng đồng hồ chung hay riêng khi truyền tín hiệu số đi xa.

Các IC đều là các vi mạch có thể lập trình. Đầu tiên lập trình chế độ hoạt động mong muốn bằng cách ghi một byte có nghĩa vào thanh ghi chế độ mode register. Sau đó ghi tiếp byte điều khiển vào thanh ghi lệnh command register để vi mạch theo đó mà hoạt động.

Vì các giao tiếp truyền nối tiếp được dùng khá rộng rãi trong các thiết bị điện tử hiện đại, các vi mạch ngoại vi LSI đặc biệt đã được phát triển cho phép thực hiện các loại giao tiếp này. Tên tổng quát của hầu hết các IC này là:

- ✓ UART (Universal Asynchronous Receiver Transmitter)
- ✓ USRT (Universal Synchronous Receiver Transmitter): mạch này đồng bộ thiên hướng ký tự.
- ✓ USART có thể hoạt động theo UART hay USRT tùy chọn.
- ✓ BOPs (Bit-Oriented Protocol Circuits) mạch này đồng bộ thiên hướng bit.
- ✓ UCCs (Universal Communication Control circuits) có thể lập trình cho cả 3 loại trên.

Cả UART và USART đều có khả năng thực hiện nhu cầu chuyển đổi song song sang nối tiếp để truyền số liệu đi xa và chuyển đổi nối tiếp sang song song khi tiếp nhận số liệu. Đối với số liệu truyền bất đồng bộ, chúng cũng có khả năng đóng khung cho ký tự một cách tự động với START bit, PARITY bit và các STOP bit thích hợp.

Các thanh ghi

Cổng	Địa chỉ cơ bản	IRQ
COM1	3F8h	IRQ4
COM2	2F8h	IRQ3
COM3	3F8h	(IRQ4)
COM4	2F8h	(IRQ3)

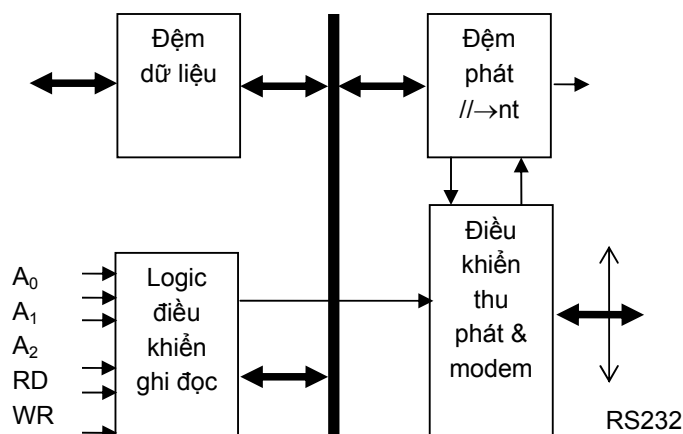
Địa chỉ cơ sở và ngắt của các cổng nối tiếp.

Các cổng nối tiếp từ thứ nhất đến thứ tư đều được phân biệt qua các vị trí địa chỉ trong vùng vào/ ra của máy tính và các số ngắt tương ứng (IRQ). Địa chỉ đầu tiên của UART, cụ thể là của thanh ghi đệm truyền/ nhận, được tính là địa chỉ cơ sở. Thông thường, địa chỉ cơ sở và IRQ được quy định nhờ các đầu nối (Jumper) trên Card vào/ ra hoặc trên bản mạch chính

Mạch điều khiển truyền thông dị bộ vạn năng UART (VXL 8250A)

Vi mạch 8250A là một UART được dùng rộng rãi trong các máy IBM PC tại vì phối ghép nối tiếp có đầu nối ra cổng thông tin nối tiếp theo chuẩn RS 232C

Sơ đồ:



Các thanh ghi có thể chia làm 3 loại:

1. Thanh ghi điều khiển (Control Register): dùng để nhận và thực hiện các lệnh từ CPU.
2. Thanh ghi trạng thái (Status Register): dùng để thông báo cho CPU biết về trạng thái của UART hay UART đang làm gì.
3. Thanh ghi đệm (Buffer Register): dùng để giữ ký tự trong lúc truyền hoặc xử lý.

Các thanh ghi này cũng giữ các ký tự nhị phân được truyền và nhận. Việc truy nhập lên các thanh ghi được thực hiện thông qua địa chỉ và khối điều khiển. Mỗi thanh ghi được gán một địa chỉ tính theo cách so sánh tương đối (Offset) với địa chỉ cơ sở của cổng nối tiếp. Các địa chỉ của hai cổng nối tiếp đầu tiên trong hầu hết các máy tính đã được tiêu chuẩn hoá.

Để viết phần mềm ghép nối qua cổng nối tiếp ta cần lưu ý là: toàn bộ hoạt động của giao diện nối tiếp đều được điều khiển qua các thanh ghi của UART, trong đó thanh ghi đệm truyền/ nhận dữ liệu thường được tính là hai thanh ghi. Do chỉ có 8 địa chỉ nên cần đến sự chuyển mạch bên trong thông qua bit DLAB (Division Latch Access Bit, bit 7 của thanh ghi điều khiển đường truyền). Các địa chỉ của từng thanh ghi đều được tính theo khoảng cách đến địa chỉ cơ sở, khoảng cách này thường được gọi là Offset. Tùy theo các thanh ghi, Offset nhận giá trị cụ thể trong khoảng từ 0 đến 7.

DLAB	A2	A1	A0	Thanh ghi	Địa chỉ
0	0	0	0	Bộ đệm đọc/ghi – RBH	3F8 (2F8)
0	0	0	1	Cho phép ngắt - IER	3F9 (2F9)
X	0	1	0	Nhận dạng ngắt (chỉ đọc) – IIR	3FA (2FA)
X	0	1	1	Điều khiển đường truyền – LCR	3FB (2FB)
X	1	0	0	Điều khiển modem – MCR	3FC (2FC)
X	1	0	1	Trạng thái đường truyền – LSR	3FD (2FD)
X	1	1	0	Trạng thái modem – MSR	3FE (2FE)
X	1	1	1	Không dùng	
1	0	0	0	Chốt số chia (LSB)	3F8 (2F8)
1	0	0	1	Chốt số chia (MSB)	3F9 (2F9)

Các thanh ghi trên vi mạch 8250.

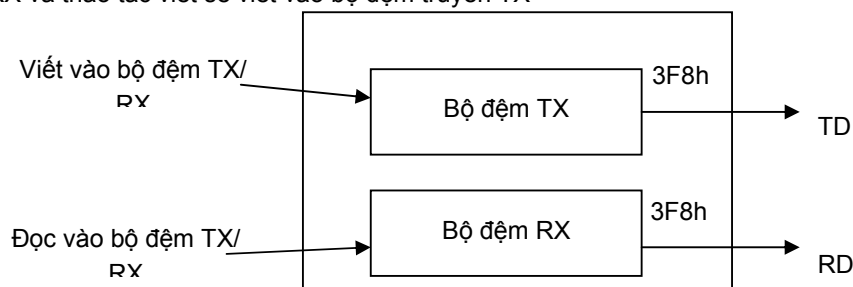
Vi mạch UART 8250 có tất cả 10 thanh ghi, sau đây ta sẽ lần lượt tìm hiểu các thanh ghi này:

Các thanh ghi lưu trữ

Như thấy rõ từ tên gọi, các thanh ghi này thực chất là các bộ đệm được chuyên dùng để giữ một ký tự, ký tự này hoặc là đã được nhận nhưng chưa được đọc, hoặc là được gửi tới cổng nối tiếp nhưng còn chưa được truyền đi. Khi mô tả quá trình truyền dữ liệu qua cổng nối tiếp, thanh ghi giữ (Holding Register) thường được gọi là bộ đệm nhận hoặc bộ đệm truyền.

Việc trang bị các bộ đệm nhận và truyền cũng là một đặc điểm của vi mạch 8250. Đặc điểm này cho phép một ký tự thứ hai được gửi tới cổng nối tiếp trước khi ký tự thứ nhất đã được truyền hoặc được đọc xong xuôi bởi bộ xử lý. Trong thời gian chờ ký tự thứ nhất được truyền hoặc được đọc, ký tự thứ hai được giữ trong bộ đệm.

Sau đây ta sẽ thấy rõ hơn là: trạng thái của bộ đệm truyền và bộ đệm nhận được quy định bởi thanh ghi trạng thái đường truyền, cụ thể hơn là ở bit 7 của thanh ghi điều khiển đường truyền LCR (Line Control Register). Khi bit này được đặt bằng '0' thì thao tác đọc từ địa chỉ cơ sở sẽ đọc từ bộ đệm nhận RX và thao tác viết sẽ viết vào bộ đệm truyền TX



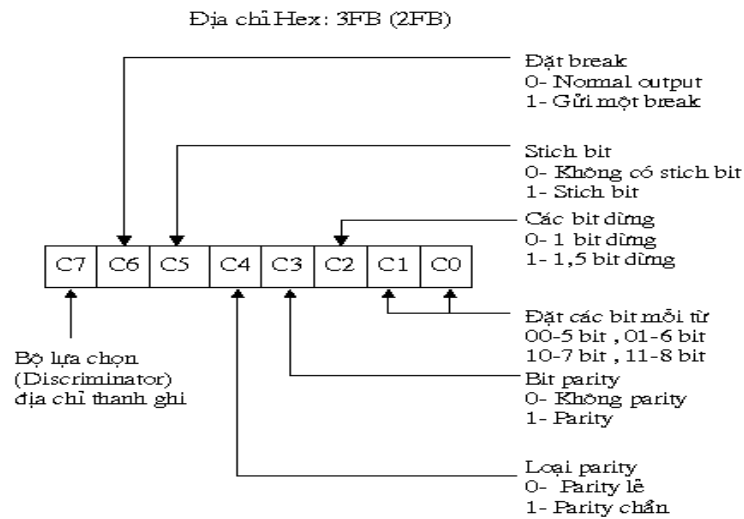
Đọc ra và ghi vào từ bộ đệm TX/ RX.

THANH GHI ĐIỀU KHIỂN ĐƯỜNG TRUYỀN

Một thanh ghi khác trong vi mạch 8250 được gọi là thanh ghi điều khiển đường truyền LCR (Line Control Register). Thanh ghi này lưu trữ các tham số được người lập trình thiết lập và xác định khuôn mẫu khung truyền của cuộc trao đổi thông tin. Các thông tin về: số các bit dữ liệu, số lượng bit dừng và kiểu chẵn lẻ được sử dụng trong khung truyền đều được cất giữ trên thanh ghi này. Dữ liệu có thể được viết vào thanh ghi này và được đọc ra sau đây. Chức năng các bit của thanh ghi LCR.

- Các bit 0 và 1. Giá trị được cất giữ trong hai bit nhị phân này chỉ rõ số các bit dữ liệu trong từng ký tự được truyền. Số các bit trên một ký tự có thể nằm trong khoảng từ 5 đến 8 bit, cho phép xác định độ dài của từ (Word). Lời giải thích cho bit 0 và 1 trên hình vẽ 12 làm sáng tỏ thêm vai trò của các bit này.
- Bit 2 chỉ rõ số các bit dừng trong mỗi khung truyền. Nếu như bit 2 có một giá trị logic bằng 0 thì số bit dừng sẽ được vi mạch 8250 tạo ra.
 Nếu ký tự được truyền có sáu, bảy hoặc tám bit dữ liệu và bit 2 được đặt vào một logic 1 thì hai bit dừng sẽ được tạo ra và "đính kèm" vào từng từ được truyền. Nếu như năm bit dữ liệu được chọn làm hệ thống mã dừng cho một ký tự thì cần đến 1,5 bit dừng chèn vào trong từ dữ liệu. Yêu cầu này cần thiết để thích ứng với các thiết bị đã cũ trên đó sử dụng năm bit dữ liệu.
- Bit 3. Được quy định là bit cho phép chẵn lẻ, nghĩa là có sử dụng bit chẵn lẻ hay không. Nếu bit này có giá trị logic 1 thì bit chẵn lẻ sẽ được tạo ra và chèn vào từng xâu ký tự. Do tính chẵn lẻ đã được cho phép nên bất kỳ ký tự nào nhận được cũng đều bị kiểm tra về tính chẵn lẻ.
- Bit 4. Kiểu chẵn lẻ đã được chọn, lẻ hoặc chẵn, được xác định bằng cách đặt bit 4. Khi cất giữ một trạng thái logic 0 ở vị trí này có nghĩa là đặt tính chẵn lẻ là lẻ và ngược lại, cất giữ một trạng thái logic 1 ở bit 4 có nghĩa là đặt tính chẵn lẻ là chẵn. Nếu như bit 3, tức là bit cho phép chẵn lẻ, bị cấm bằng cách đặt một giá trị logic 0 vào vị trí này thì bất kể là giá trị bit như thế nào được đặt ở vị trí bit 4 cũng không có tác dụng.

- Bit 5. (Bit stick parity). Nếu như bit 3 và bit 5 được đặt giá trị logic 1 thì khi bộ truyền xuất ra một ký tự, bộ nhận tại chỗ (local) sẽ phát hiện như một giá trị logic 3.
- Bit 6. Được quy định là bit BREAK (dừng). Khi bit này được đặt một giá trị logic 1 thì nó bắt buộc SOUT (Serial out hay TxD) chuyển sang mức logic trống (mức LOW) cho đến khi một giá trị logic 0 được cất giữ vào bit 6. Nhờ có bit này mà máy tính có thể báo hiệu cho thiết bị đầu cuối biết là đã được nối như một phần của hệ thống truyền thông.
- Bit 7. Phải được đặt một giá trị logic 1 để truy nhập các chốt số chia (divisor latches). Các chốt này là những thanh ghi cất giữ số chia đối với tín hiệu giữ nhịp (đồng hồ), số này quy định tốc độ baud của hệ thống truyền thông nối tiếp. Mỗi lần tốc độ baud được đặt lại thì bit này (bit 7) lại được đặt về giá trị logic 0.



Các bit trên thanh ghi điều khiển đường truyền (LCR).

THANH GHI TỐC ĐỘ BAUD

Tốc độ baud được đặt bằng cách nạp một số chia chiếm 16 bit, trong đó 8 bit thấp hơn của số chia được đặt trên địa chỉ bộ đệm TX/ RX và 8 bit phía trên đặt địa chỉ kế tiếp sau bộ đệm TX/ RX. Sự tăng gấp đôi số các thanh ghi là cần thiết vì khi bit 7 hoặc thanh ghi LCR (thường viết tắt là DLAB) được lại về giá trị logic 0 hai địa chỉ này gắn liền với bộ đệm nhận và bộ đệm truyền. Khi bit DLAB được đặt vào một giá trị logic 1 thì hai địa chỉ này gắn liền với hai chốt số chia. Các chốt số chia bao gồm 16 bit hay hai byte, được sắp xếp thành các bit có giá trị thấp LSB (Least Significant byte) và bit có giá trị cao hơn MSB (Most Significant bit), được sử dụng trong việc đặt tốc độ baud của hệ thống truyền thông.

Bởi vì các chốt số chia có độ rộng là hai byte, giá trị 060 Hex cần được chia ra để cất giữ trên hai thanh ghi LSB và MSB. Với giá trị tốc độ baud bằng 1200 trong thí dụ này, 60 Hex được cất giữ trong LSB (bit có giá trị thấp) và giá trị 0 được cất trong MSB (bit có giá trị cao hơn).

Tốc độ baud muốn có	Số chia được dùng để tạo ra: 16 x Đồng hồ		Sai số theo phần trăm (sai khác giữa mong muốn và thực tế)
	Thập phân	Hex	
50	2304	900	
75	1536	600	
110	1047	417	0,026

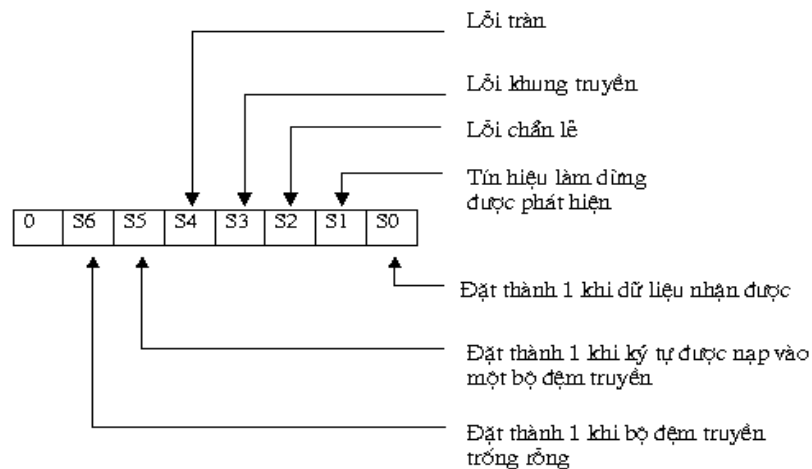
134,5	857	359	0,058
150	768	300	-
300	384	180	-
600	192	0C0	-
1200	96	060	-
1800	64	040	-
2000	58	03A	0,69
2400	48	030	-
3600	32	020	-
4800	24	018	-
7200	16	010	-
9600	12	00C	-

Bảng tốc độ baud ứng với xung nhịp 1,8432 MHz.

Một số tốc độ baud và các giá trị số chia tương ứng dưới cả hai dạng thập phân và thập lục phân (Hex). Giá trị này của số chia được nạp vào bộ đếm TX/ RX khi bit DLAB được một giá trị logic 1 đặt vào.

Thanh ghi trạng thái đường truyền

Thanh ghi trạng thái đường truyền (LSR: Line Status Register) thanh ghi 8 bit, chứa thông tin về quá trình dữ liệu qua cổng nối tiếp cần cung cấp cho bộ vi xử lý.



Thanh ghi trạng thái đường truyền.

- Bit 0, được dùng để thông báo cho biết dữ liệu đã nhận được (DR: Data Received). Khi bit 0 có giá trị logic 1 có nghĩa là dữ liệu đã được nhận và sẵn sàng để bộ xử lý đọc.
- Bit 1: Một giá trị logic 1 ở bit này có nghĩa là ký tự nhận trước đó đã bị mất vì nó không được đọc trước khi một ký tự mới được nhận nên ký tự mới đã ghi đè lên ký tự trước.

- Bit 2: Một giá trị logic 1 ở bit lỗi chẵn lẻ có nghĩa là ký tự đã được nhân có tính chẵn lẻ sai. Khi thanh ghi trạng thái đường truyền (LSR) được đọc thì bit này lại được đặt về giá trị logic 0.
- Bit 3: Đây là bit lỗi khung truyền. Nếu ký tự đã nhận không có một bit đúng hợp lệ, nghĩa là có lỗi khung truyền, thì bit 3 trong thanh ghi LSR được đặt vào một giá trị logic 1.
- Bit 4: được quy định là bit gián đoạn ngắt (break interrupt bit). Bit này được tự động đặt vào một giá trị logic 1 khi dữ liệu nhận được đã được giữ ở một mức trống trên toàn bộ chiều dài của một từ dữ liệu.
- Bit 5: được quy định là bit báo hiệu trạng thái rỗng của bộ đệm truyền (THRE: Transmit Holding Register Empty). Bit này báo hiệu là cổng nối tiếp sẵn sàng tiếp nhận ký tự khác được truyền tới.
- Bit 6: Bit này là một bit chỉ để đọc. Khi bit này có giá trị logic 1 thì bộ đệm truyền đang còn trống.
- Bit 7: không được sử dụng và luôn được đặt giá trị logic 0.

Khi viết phần mềm truy nhập thanh ghi lên thanh ghi trạng thái đường truyền ta cần lưu ý tới một số chức năng của thanh ghi này. Thanh ghi trạng thái đường truyền (LSR: Line Status Register) xác định trạng thái của bộ đệm truyền và bộ đệm nhận. Thanh ghi này chỉ dùng để đọc ra, nội dung tất cả các bit được tự động đặt bằng phần cứng.

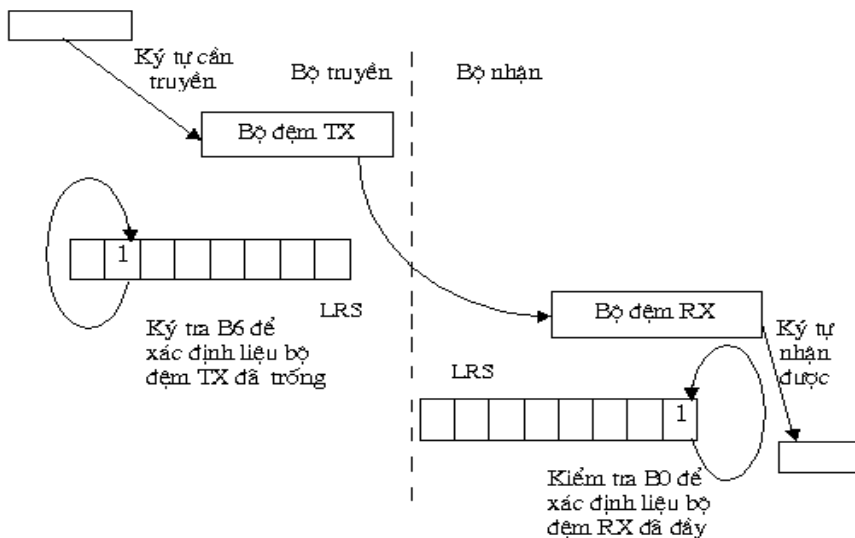
Một điều rủi ro có thể xảy ra khi truyền dữ liệu một ký tự mới có thể được viết vào bộ đệm truyền trước khi ký tự trước đây đã được gửi. Khi đó ký tự mới này sẽ viết đè lên nội dung của ký tự đang được truyền. Để tránh tình trạng rủi ro này S5 được giao nhiệm vụ thông báo kết quả kiểm tra xác định liệu vẫn còn một ký tự ở trong bộ nhớ. Nếu có thì nó được đặt thành '1', còn nếu như bit này có giá trị bằng 0 thì có nghĩa là bộ đệm truyền đang trong trạng thái trống rỗng.

Để truyền một ký tự:

- *Kiểm tra bit 6 cho đến khi được đặt; (Test Bit 6 until set;)*
- *Truyền ký tự; (Send character;)*

Để nhận ký tự:

- *Kiểm tra bit 0 cho đến khi được đặt; (Test Bit 0 until set;)*
- *Đọc ký tự; (Read character;)*



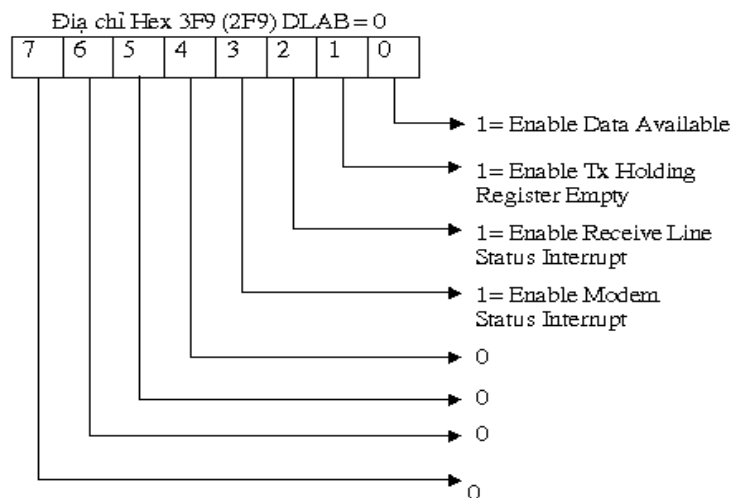
Kiểm tra thanh ghi LSR để truyền và nhận các ký tự.

Thanh ghi cho phép ngắt

Vi mạch 8250 có một nhiều khả năng ngắt. Có hai thanh ghi được sử dụng để điều khiển và xác định các nguồn ngắt. Thanh ghi đầu tiên trong hai thanh ghi đó là thanh ghi cho phép ngắt IER (Interrupt Enable Register) còn thanh ghi thứ hai là thanh ghi nhận dạng ngắt IIR (Interrupt Identification Register).

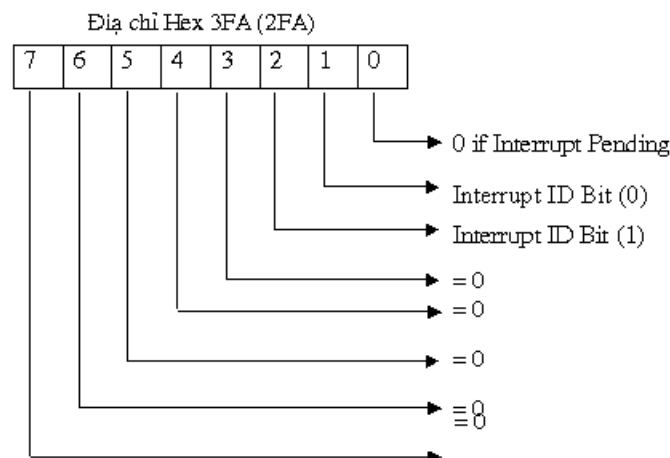
Nếu như khả năng ngắt của vi mạch đã cho phép và một ngắt xuất hiện thì bit xuất ra ngắt từ 8250 chiếm lấy mức logic 1. Tín hiệu này được nối với bus ngắt cứng của máy tính. Logic 1 trên bus này báo hiệu cho bộ xử lý biết là cần phải chú ý tới cổng nối tiếp. Hình 15 minh họa sự phân bố của các bit trên thanh ghi IER.

- Bit 0: Mỗi lần nhận một ký tự thì một ngắt lại được tạo ra. Bit này được đặt lại (Reset) sau khi ký tự đã được bộ xử lý đọc.
- Bit 1: Nếu bit này được đặt một giá trị logic 1 thì bộ đệm truyền (thanh ghi giữ truyền) trống và một ngắt xuất hiện.
- Bit 2: cho phép có sự thay đổi trong trạng thái đường truyền bộ nhận theo cách gây ra một ngắt
- Bit 3: cho phép có sự thay đổi trong trạng thái modem để ngắt bộ xử lý.
- Bit 4- 7: Các bit này luôn được đặt giá trị logic 0.



Thanh ghi cho phép ngắt.

Thanh ghi nhận dạng ngắt



Thanh ghi nhận dạng ngắt.

Nếu như một ngắt xuất hiện thì phần mềm chương trình phải thực hiện được chức năng kiểm tra thanh ghi để xác định xem sự kiện nào đang gây ra ngắt. Thanh ghi nhận dạng ngắt IIR chứa đựng mã, nhận dạng điều kiện (ngắt) nào đang yêu cầu chú ý.

Một điểm cần chú ý là: giữa các ngắt cũng có mức độ ưu tiên khác nhau, nói khác đi là có một vài ngắt tỏ ra là "quan trọng" hơn so với các ngắt khác. Về nguyên tắc, ngắt nào quan trọng hơn sẽ được ưu tiên xử lý trước.

Thanh ghi nhận dạng ngắt				Các ngắt và đặt lại chức năng		
Bit 2	Bit 1	Bit 0	Mức ưu tiên	Kiểu ngắt	Nguồn ngắt	Điều khiển đặt lại ngắt
0	0	1	-	Không dừng	Không dừng	-
1	1	0	Cao nhất	Trạng thái đường nhận	Lỗi tràn hoặc lỗi chặn lẻ hoặc lỗi khung truyền hoặc break interrupt	Đọc thanh ghi trạng thái đường truyền
1	0	0	Thứ hai	Có dữ liệu đã nhận	Có dữ liệu đã nhận	Đọc thanh ghi đệm bộ nhận
0	1	0	Thứ ba	Bộ đệm truyền trống	Bộ đệm truyền trống	Đọc thanh ghi IR (nếu là nguồn ngắt) hoặc ghi vào bộ đệm truyền
0	0	0	Thứ tư	Trạng thái modem	Xoá để gửi hoặc dữ liệu sẵn sàng hoặc báo chuông hoặc phát hiện tín hiệu đường nhận	Đọc thanh ghi trạng thái modem

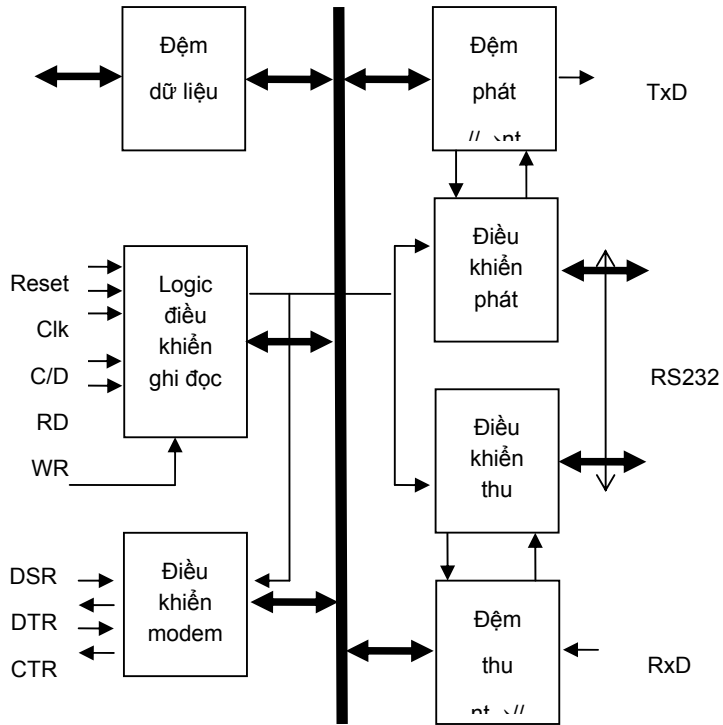
Các mức ưu tiên của từng ngắt.

Bảng trên liệt kê các mức ưu tiên của từng ngắt. Cột đặt lại ngắt liệt kê tác động nào là cần đến để đặt lại ngắt đã được chốt.

Mạch điều khiển truyền thông đồng bộ - di bộ vận năng USART (VXL 8251A)

Vì mạch 8251A là một USART được dùng rộng rãi trong các máy IBM PC tại vị phối ghép nối tiếp có đầu nối ra cổng thông tin nối tiếp theo chuẩn RS 232C

Sơ đồ:



Các thanh ghi có thể chia làm 3 loại:

- Thanh ghi điều khiển (Control Register): dùng để nhận và thực hiện các lệnh từ CPU.
- Thanh ghi trạng thái (Status Register): dùng để thông báo cho CPU biết về trạng thái của UART hay UART đang làm gì.
- Thanh ghi đệm (Buffer Register): dùng để giữ ký tự trong lúc truyền hoặc nhận

Thanh ghi từ chế độ

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S ₂	S ₁	EP	PEN	L ₂	L ₁	B ₂	B ₁

- Bit 0,1 được dùng để đồng bộ và hệ số nhân tốc độ

00	đồng bộ	01	nhân 1
10	nhân 16	11	nhân 64
- Bit 2,3: số bit mã kí tự

00	5	01	6
10	7	11	8
- Bit 4: cho phép dùng Parity hay không
- Bit 5: Parity bit
- Bit 6,7: số bit STOP

00	không hợp lệ	01	1
10	1 _{1/2}	11	2

Thanh ghi từ lệnh

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxE	DTR	TxEN

- Bit 0: cho phép phát tín hiệu
- Bit 1: DTE sẵn sàng
- Bit 2: Cho phép thu
- Bit 3: Gửi kí tự gián đoạn (kí tự với tất cả các bit là 0)
- Bit 4: Xoá cờ lỗi
- Bit 5: Yêu cầu truyền
- Bit 6: Reset nội bộ
- Bit 7: tìm kiếm kí tự đồng bộ

Thanh ghi trạng thái

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DSR	SYNDET	FE	OE	PE	TxEMPTY	RxRDY	TxRDY

- Bit 0: bên phát sẵn sàng
- Bit 1: bên thu sẵn sàng
- Bit 2: đệm phát rỗng
- Bit 3: lỗi Parity
- Bit 4: lỗi thu đề
- Bit 5: lỗi Frame
- Bit 6: kí tự đồng bộ
- Bit 7: modem sẵn sàng

Tài liệu trên được đăng bởi Đặng Vũ Thanh Hùng

(Sinh Viên Trường Cao Đẳng Kỹ Thuật Cao Thắng //// >Khóa< __2010__)

Thuộc: >Khoa< __Cơ Khí__ = \"_#@#_ \"/=____>Ngành< __Cơ Điện Tử

>Lớp< __CĐ CĐT 10B__ = \"_#@#_ \"/=____>MSSV< __307101112

[("~^_^~")Định ghi số thứ tự trong lớp nữa mà thấy "Sàm" quá __Nội nhiều đây là thấy "Quá Sàm" rùi __Hjhjhj.]

Địa chỉ thường trú: 292 Bà Hạt P9_Q10_Thành Phố Hồ Chí Minh.

SĐT [(*_*) Có nên cho không ta, mình sợ "bị khó ngủ lắm", thui vậy là được rùi]

Lưu ý: Tài liệu này, không phải tự mình viết ra, bản thân mình chỉ đăng lên, bạn thấy có nhu cầu thì Download về, xuất xứ của nó...thì...Ai mà biết

Bạn phải cảm ơn tui đó nha, vì tui đã cho bạn, cái mà bạn đang cần!

(Cho dù không cần cũng phải cảm ơn __hjhjhj_ ("~^_^~") __Mình không thích người khác mắc nợ mình)

Bạn có thể gửi ý kiến của bạn qua "Đường dây nóng" __ (Mặt dù chả thấy nóng bao giờ_ vì có thấy ai nói gì đâu __hjhj)

Email: mr_thanhhung_20111992@yahoo.com

(hoặc giolanganhseve@yahoo.com nick sài đỡ mùa mưa ấy mà! Hjhjhj.)

Một tương lai sẽ hé mở khi bạn biết cách

chia sẻ những gì bạn có cho người khác.

__ (Đặng Vũ Thanh Hùng) __

TÀI LIỆU THAM KHẢO

1. William Stalling - *Computer Organization and Architecture*, 1997
2. Mc. Graw - *Computer Architecture*, 1997
3. Văn thế Minh - *Kỹ thuật vi xử lý* - NXB giáo dục 1997
4. Nguyễn Kim Khánh - *Giáo trình kiến trúc máy tính* - ĐHBK Hà nội
5. Nguyễn Đình Việt - *Giáo trình kiến trúc máy tính* - ĐHQG Hà nội
6. Trần Thái Bá - *Điều khiển và ghép nối thiết bị ngoại vi* - NXB thống kê 2000
7. Trần Quang Vinh - *Cấu trúc máy tính* - NXB giáo dục 1997
8. Võ Văn Thành - *Máy vi tính sự cố chẩn đoán và cách giải quyết* - NXB thống kê 1996.
9. William Stalling- *Computer Architecture Advanced*, 1997

**BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC HÀNG HẢI
BỘ MÔN: KỸ THUẬT MÁY TÍNH
KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI GIẢNG
LẬP TRÌNH
GHÉP NỐI THIẾT BỊ NGOẠI VI**

**TÊN HỌC PHẦN : LẬP TRÌNH GHÉP NỐI
THIẾT BỊ NGOẠI VI
MÃ HỌC PHẦN : 17311
TRÌNH ĐỘ ĐÀO TẠO : ĐẠI HỌC CHÍNH QUY
DÙNG CHO SV NGÀNH : CÔNG NGHỆ THÔNG TIN**

HẢI PHÒNG - 2010

MỤC LỤC

Chương 1: Các chuẩn ghép nối.....	5
1.1. Cổng song song.....	5
1.1.1. Cấu trúc cổng.....	5
1.1.2. Trao đổi cổng song song với đường dẫn tín hiệu.....	6
1.1.3. Lập trình cổng song song.....	8
1.2. Cổng nối tiếp.....	8
1.2.1. Các đặc trưng điện áp.....	8
1.2.2. Khôn mẫu khung truyền.....	10
1.2.3. Các vấn đề truyền thông.....	11
1.3. Bus nối tiếp đa năng.....	12
1.3.1. Bus nối tiếp đa năng - Giới thiệu chung.....	12
1.3.2. Đầu nối và cáp tín hiệu.....	13
1.3.3. Truyền dữ liệu nối tiếp.....	14
1.3.4. Phần cứng, phần mềm và kết nối hệ thống.....	14
1.3.5. Chuẩn USB 2.0.....	15
1.4. Khe cắm mở rộng.....	15
1.4.1. BUS ISA 16 bit.....	15
1.4.2. BUS PCI.....	17
1.4.4. Ghép nối qua khe cắm mở rộng.....	18
Chương 2. Bus Ghép nối đa năng.....	20
2.1. Đặt vấn đề.....	20
2.2. Tiêu chuẩn IEEE-488 truyền thông.....	20
2.3. Cấu hình và hoạt động của Bus GPIB.....	21
Chương 3. Vòng đo điện áp.....	23
3.1. Truyền dữ liệu bằng vòng dòng điện.....	23
3.2. Vòng dòng điện 60mA.....	23
3.3. Vòng dòng điện 20mA.....	24
3.4. Vòng dòng điện 4 đến 20mA.....	25
Chương 4. Các mạch điều khiển với bộ biến đổi A/D.....	26
4.1. Card biến đổi A/D 12 bit dùng ICL7107.....	26
4.2. Card biến đổi A/D 12 bit dùng ADC547.....	26
4.3. Card biến đổi ADA 9-16 bit.....	27
Chương 5. Tự động hóa với PCL S7-200.....	29
5.1. Bộ điều khiển khả trình PCL.....	29
5.2. Soạn thảo chương trình với PCL.....	30
Chương 6. Họ vi điều khiển 8951.....	33
6.1. Cấu trúc phần cứng.....	33
6.2. Tập lệnh.....	35

YÊU CẦU VÀ NỘI DUNG CHI TIẾT

Tên học phần: **Lập trình ghép nối thiết bị ngoại vi**
Bộ môn phụ trách giảng dạy: **Kỹ thuật máy tính**
Mã học phần: **17311**

Loại học phần: **2**
Khoa phụ trách: **CNTT**
Tổng số TC: **2**

TS tiết	Lý thuyết	Thực hành/Xemina	Tự học	Bài tập lớn	Đồ án môn học
45	30	15	0	0	0

Điều kiện tiên quyết:

Sinh viên phải học xong các học phần sau mới được đăng ký học phần này:
Kiến trúc máy tính, Điện tử số, Mạch và tín hiệu, Vi xử lý, Kỹ thuật lập trình

Mục tiêu của học phần:

- Cung cấp các kiến thức cơ bản về cấu trúc, nguyên lý hoạt động của hệ thống giao tiếp với thế giới bên ngoài.
- Nắm được các phương thức điều khiển vào/ra dữ liệu.
- Hiểu rõ nguyên tắc, cách thức phối ghép các bộ điều khiển cơ bản

Nội dung chủ yếu

- Chương I. Các chuẩn ghép nối.
- Chương II. Bus Ghép nối đa năng.
- Chương III. Vòng đo điện áp.
- Chương IV. Các mạch điều khiển với bộ biến đổi A/D.
- Chương V. Họ vi điều khiển 805x.

Nội dung chi tiết của học phần:

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	BT	TH	KT
Chương I. Các chuẩn ghép nối.	25	16		9	
1.1. Cổng song song					
1.1.1. Cấu trúc cổng		1			
1.1.2. Trao đổi cổng song song với đường dẫn tín hiệu		1			
1.1.3. Lập trình cổng song song		2		3	
1.2. Cổng nối tiếp RS-232					
1.2.1. Các đặc trưng điện áp		1			
1.2.2. Khuôn mẫu khung truyền		0.5			
1.2.3. Các vấn đề truyền thông		0.5			
1.2.4. Lập trình cho cổng nối tiếp RS-232		1		3	
1.3. Bus nối tiếp đa năng					
1.3.1. Giới thiệu chung		0.5			
1.3.2. Đầu nối và cáp tín hiệu		0.5			
1.3.3. Truyền dữ liệu nối tiếp		0.5			

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	BT	TH	KT
1.3.4. Phần cứng, phần mềm và kết nối hệ thống		0.5			
1.3.5. Chuẩn USB 2.0		0.5			
1.3.6. Các kiểu truyền và phần mềm điều khiển		0.5			
1.3.7. Lập trình Bus nối tiếp đa năng		2		3	
1.4. Khe cắm mở rộng					
1.4.1. BUS ISA 16 bit		1			
1.4.2. BUS PCI		1			
1.4.3. BUS AGP		1			
1.4.4. Ghép nối qua khe cắm mở rộng		1			
Chương II. Bus Ghép nối đa năng	3	2			1
2.1. Đặt vấn đề		0.5			
2.2. Tiêu chuẩn IEEE-488 truyền thông		0.5			
2.3. Cấu hình và hoạt động của Bus GPIB		1			
Chương III. Vòng đo điện áp	3	3			
3.1. Truyền dữ liệu bằng vòng dòng điện		1			
3.2. Vòng dòng điện 60mA		0.5			
3.3. Vòng dòng điện 20mA		0.5			
3.4. Vòng dòng điện 4 đến 20mA		1			
Chương IV. Các mạch điều khiển với bộ biến đổi A/D	6	3		3	
4.1. Card biến đổi A/D 12 bit dùng ICL7107		0.5			
4.2. Card biến đổi A/D 12 bit dùng ADC547		0.5			
4.3. Card biến đổi ADA 9-16 bit		1			
4.4. Bộ đếm tần số dùng cho máy tính		1			
Chương V. Họ vi điều khiển 51	3	2			1
5.1. Bộ điều khiển khả trình PCL		1			
5.2. Soạn thảo chương trình với PCL		1			
Chương VI. Họ vi điều khiển 51	5	2		3	
6.1. Cấu trúc phần cứng		0,5			
6.2. Tập lệnh		1			
6.3. Kết nối điều khiển bằng 8951		0,5			

Nhiệm vụ của sinh viên :

Tham dự các buổi thuyết trình của giáo viên, tự học, tự làm bài tập do giáo viên giao tham dự các buổi thực hành, các bài kiểm tra định kỳ và cuối kỳ.

Tài liệu học tập :

- Ngô Diên Tập, Đo lường và điều khiển bằng máy tính, NXB KH&KT
- Ngô Diên Tập, Kỹ thuật ghép nối máy tính, NXB KH&KT
- Phan Xuân Minh, Tự động hóa với SIMATIC S7-2000, NXB NN
- Kỹ thuật lập trình PLC-SPC
- MSC 51 Microcontroller *family user's manual- Intel*

Hình thức và tiêu chuẩn đánh giá sinh viên:

- Đánh giá dựa trên tình hình tham dự buổi học trên lớp, các buổi thực hành, điểm kiểm tra thường xuyên và điểm kết thúc học phần.
- Hình thức thi cuối kỳ: thi viết.

Thang điểm: Thang điểm chữ A, B, C, D, F

Điểm đánh giá học phần $Z = 0.3X + 0.7Y$.

Bài giảng này là tài liệu **chính thức và thống nhất** của Bộ môn Kỹ thuật máy tính, Khoa Công nghệ Thông tin và được dùng để giảng dạy cho sinh viên.

Ngày phê duyệt: 25 / 10 / 2009

Trưởng Bộ môn: ThS. Ngô Quốc Vinh

Chương 1: CÁC CHUẨN GHÉP NỐI

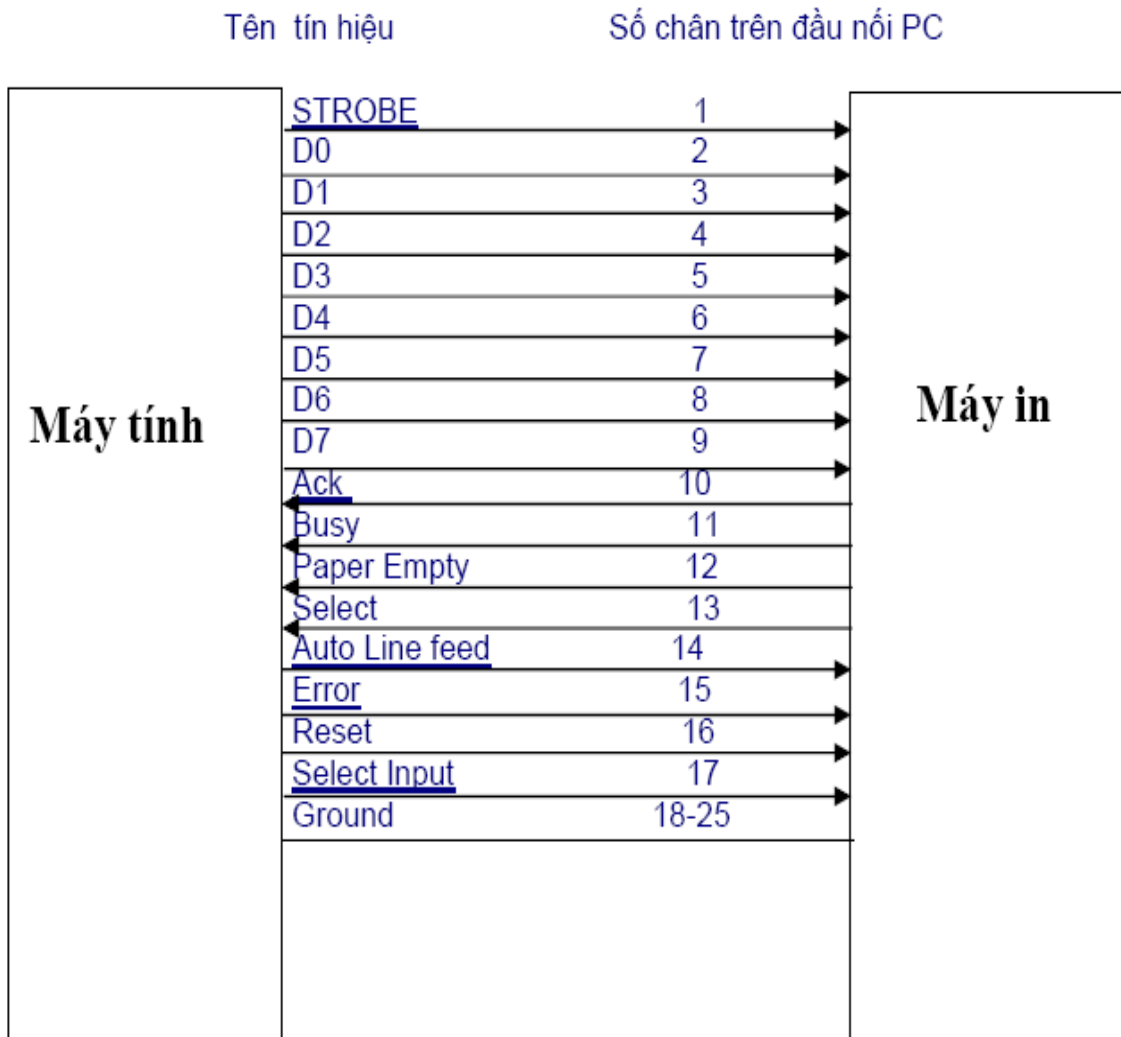
1.1. Cổng song song

1.1.1. Cấu trúc cổng

Cổng song song có hai loại: ổ cắm 36 chân và ổ cắm 25 chân. Ngày nay, loại ổ cắm 36 chân không còn được sử dụng, hầu hết các máy tính PC đều trang bị ổ cắm 25 chân nên ta chỉ cần quan tâm đến loại 25 chân.

TÊN TÍN HIỆU	VỊ TRÍ CHÂN	CHỨC NĂNG
Strobe	1	Khi đặt một mức điện áp LOW vào chân này, máy tính thông báo cho máy in biết có một byte sẵn sàng trên các đường tín hiệu để được truyền.
D0	2	Các đường dữ liệu (8 đường)
D1	3	
D2	4	
D3	5	
D4	6	
D5	7	
D6	8	
D7	9	
Acknowledge	10	Mức LOW ở chân này, máy in thông báo cho máy tính biết đã nhận được kí tự vừa gửi và có thể tiếp tục nhận.
Busy (Báo bận)	11	Máy in gửi một mức logic HIGH vào chân này trong khi đang đón nhận hay đang in ra dữ liệu để thông báo cho máy tính biết bộ đệm dữ liệu đầy hay máy in đang ở trạng thái Off-line
Paper empty (Hết giấy)	12	Máy in đặt trạng thái trở kháng cao (HIGH) ở chân này khi hết giấy.
Select (Lựa chọn)	13	Một mức HIGH có nghĩa là máy in đang trong trạng thái được kích hoạt .
Auto Linfeed (Tự động nạp dòng)	14	Mức LOW ở chân này máy tính nhắc máy in tự động nạp một dòng mới mỗi khi kết thúc một dòng.
Error (Có lỗi)	15	Mức LOW ở chân này, máy in báo cho máy tính biết đã xảy ra lỗi khi in.
Reset (Đặt lại trạng thái máy in)	16	Máy in được đặt trở lại trạng thái được xác định lúc ban đầu khi chân này ở mức LOW.
Select Input (Lựa chọn lối vào)	17	Bằng một mức LOW máy in được lựa chọn bởi máy tính.
Ground (Nối đất)	18-25	

Qua bảng trên ta thấy cáp nối giữa máy tính và máy in bao gồm 25 sợi, tuy nhiên không phải tất cả các sợi cáp đều được sử dụng như vậy chúng ta có thể tận dụng dây cáp này nếu có một vài sợi bị đứt.



Qua cách mô tả chức năng của từng tín hiệu riêng lẻ ta có thể nhận thấy là các đường dẫn tín hiệu có thể chia thành 3 nhóm:

- Các đường dẫn tín hiệu xuất ra từ máy tính PC và điều khiển máy in, được gọi là các đường dẫn điều khiển.
- Các đường dẫn tín hiệu, đưa các thông báo ngược lại từ máy in về máy tính, được gọi là các đường dẫn trạng thái.
- Đường dẫn dữ liệu, truyền các bit riêng lẻ của các ký tự cần in.

Từ cách mô tả các tín hiệu và mức tín hiệu ta có thể nhận thấy các tín hiệu **Acknowledge, Auto Linefeed, Error, Reset, và Select Input** kích hoạt ở mức **Low**. Thông qua chức năng của các chân này ta cũng hình dung được cách điều khiển máy in.

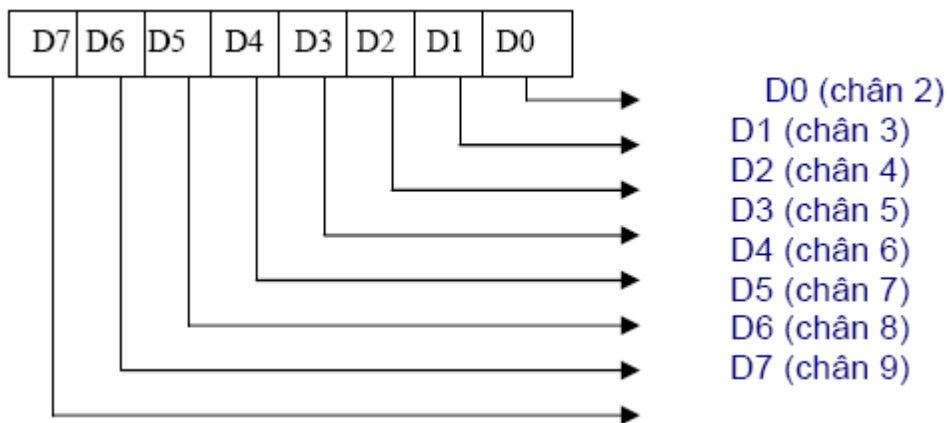
Cần chú ý là 8 đường dẫn song song đều được dùng để chuyển dữ liệu từ máy tính sang máy in. Trong những trường hợp này, khi chuyển sang ứng dụng đo lường và điều khiển ta phải chuyển dữ liệu từ mạch ngoại vi vào máy tính để thu thập và xử lý. Vì vậy ta phải tận dụng một trong năm đường dẫn theo hướng ngược lại, nghĩa là từ bên ngoài vào máy tính để truyền số liệu đo lường.

1.1.2. Trao đổi công song song với đường dẫn tín hiệu

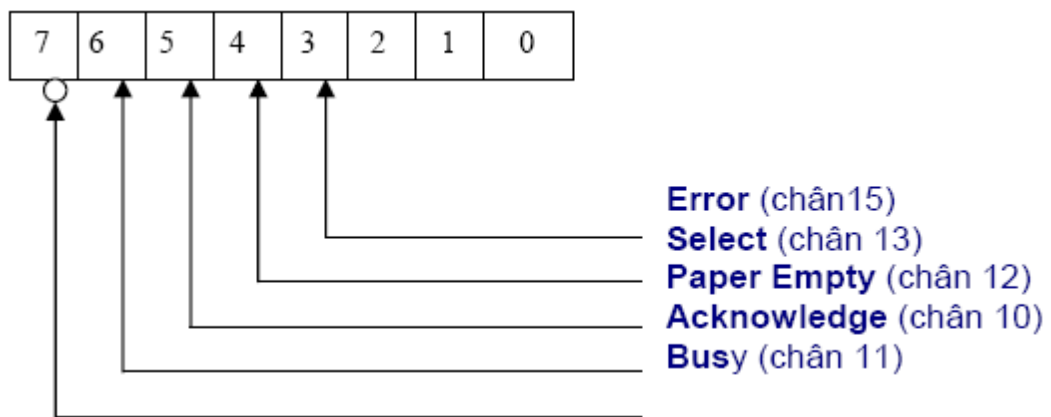
Để có thể ghép nối các thiết bị ngoại vi, các mạch điện ứng dụng trong đo lường và điều khiển với công song song ta phải tìm hiểu cách trao đổi với các thanh ghi thông qua cách sắp xếp và địa chỉ các thanh ghi. Các đường dẫn của công song song được nối với ba thanh ghi 8 bit khác nhau:

THANH GHI DỮ LIỆU

ĐỊA CHỈ (278H, 378H, 2BCH, 3BC H)

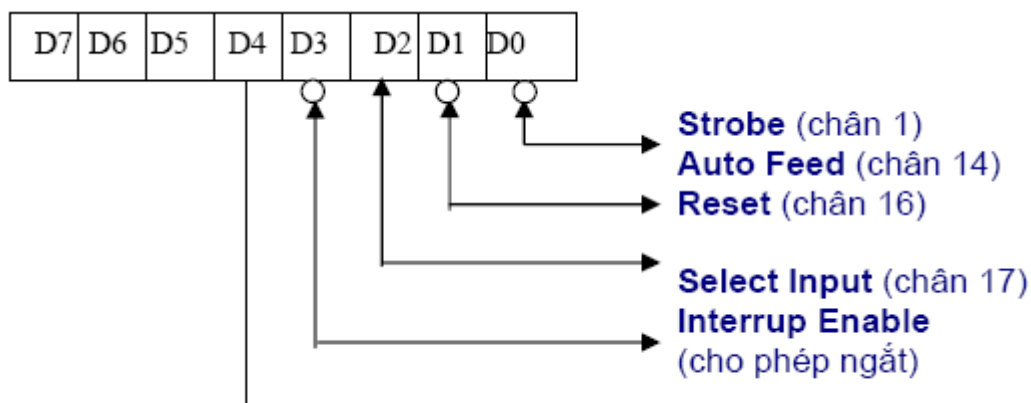


THANH GHI TRẠNG THÁI ĐỊA CHỈ (279H, 379H, 2BDH, 3BD H)



THANH GHI ĐIỀU KHIỂN

ĐỊA CHỈ (27AH, 37AH, 2BEH, 3BE H)



Như sơ đồ trên đã trình bày 8 đường dữ liệu dẫn tới thanh ghi dữ liệu còn 4 đường dẫn điều khiển **Strobe, Auto Linefeed, Reset, Select Input** dẫn tới thanh ghi điều khiển. Năm đường dẫn trạng thái **Acknowledge, Busy, Paper Empty, Select, Error** tới thanh ghi trạng thái.

Thanh ghi dữ liệu hay 8 đường dẫn dữ liệu không phải là đường dẫn 2 hướng trong tất cả các loại máy tính nên dữ liệu chỉ có thể được *xuất ra* qua các đường dẫn này cụ thể từ D0 đến D7. Thanh ghi điều khiển hai hướng, hay nói chính xác hơn: Bốn bit có giá trị thấp được sắp xếp ở các chân 1, 14, 16, 17. Thanh ghi trạng thái chỉ có thể được đọc và vì vậy được gọi là một hướng.

1.1.3. Lập trình cổng song song

Việc truy nhập trực tiếp lên các giao diện của máy tính PC, cụ thể là lên các đường dẫn riêng lẻ được tiến hành thuận lợi nhất là bằng hợp ngữ. Các ngôn ngữ bậc cao như Turbo Pascal hoặc C cũng có những lệnh đơn giản để thực hiện việc truy nhập lên các cổng tuy rằng tốc độ truy nhập có thấp hơn đôi chút.

Trước hết ta cần biết địa chỉ của các cổng mà qua đó các giao diện song song có thể được trao đổi. Sau đó sẽ quyết định đọc dữ liệu trong thanh ghi nào hoặc xuất dữ liệu ra thanh ghi nào?

Nói chung, các lệnh được sử dụng có thể viết như sau:

Bảng hợp ngữ

Để xuất ra dữ liệu OUT DX, AL

Để nhập vào dữ liệu IN AL, DX

Trong đó địa chỉ của thanh ghi cần trao đổi phải đứng ở trong DX. Ví dụ: Giả sử LPT1 có địa chỉ là 378H

MOV DX,378H; *Nạp địa chỉ của cổng LPT1 vào thanh ghi DX*

OUT DX,AL; *Xuất nội dung của thanh ghi AL lên đường dẫn dữ liệu (Từ D0 đến D7) của LPT1.*

Hai lệnh sau đây đọc các byte ở đường dẫn dữ liệu của cổng LPT1:

MOV DX, 378H; *Nạp địa chỉ của cổng LPT1 vào thanh ghi DX*

IN AL,DX; *Đọc thông tin trên các đường dẫn dữ liệu (Từ D0 đến D7) của LPT1 sang thanh ghi AL.*

Hai lệnh sau đây xuất nội dung của thanh ghi AL sang thanh ghi điều khiển của LPT1.

MOV DX,37AH; *Nạp địa chỉ của thanh ghi điều khiển cổng LPT1 vào thanh ghi DX*

OUT DX,AL; *Mang nội dung của thanh ghi AL sang thanh ghi điều khiển của LPT1.*

Hai lệnh sau đây đọc các byte ở đường dẫn trạng thái của cổng LPT1 đặt vào thanh ghi AL.

MOV DX, 379H; *Nạp địa chỉ của thanh ghi trạng thái vào thanh ghi DX*

IN AL,DX; *Đọc nội dung của thanh ghi trạng thái của LPT1 sang thanh ghi AL.*

Bảng TurboC

Để xuất ra dữ liệu Outportb(Địa chỉ cổng,Giá trị)

Để nhập vào dữ liệu Inportb(Địa chỉ cổng)

Ví dụ

Lệnh sau đây xuất giá trị 5 qua thanh ghi dữ liệu của cổng LPT1

Outportb(0x378, 5)

Lệnh sau đây đọc thông tin của thanh ghi trạng thái của cổng LPT1 sắp xếp vào biến status

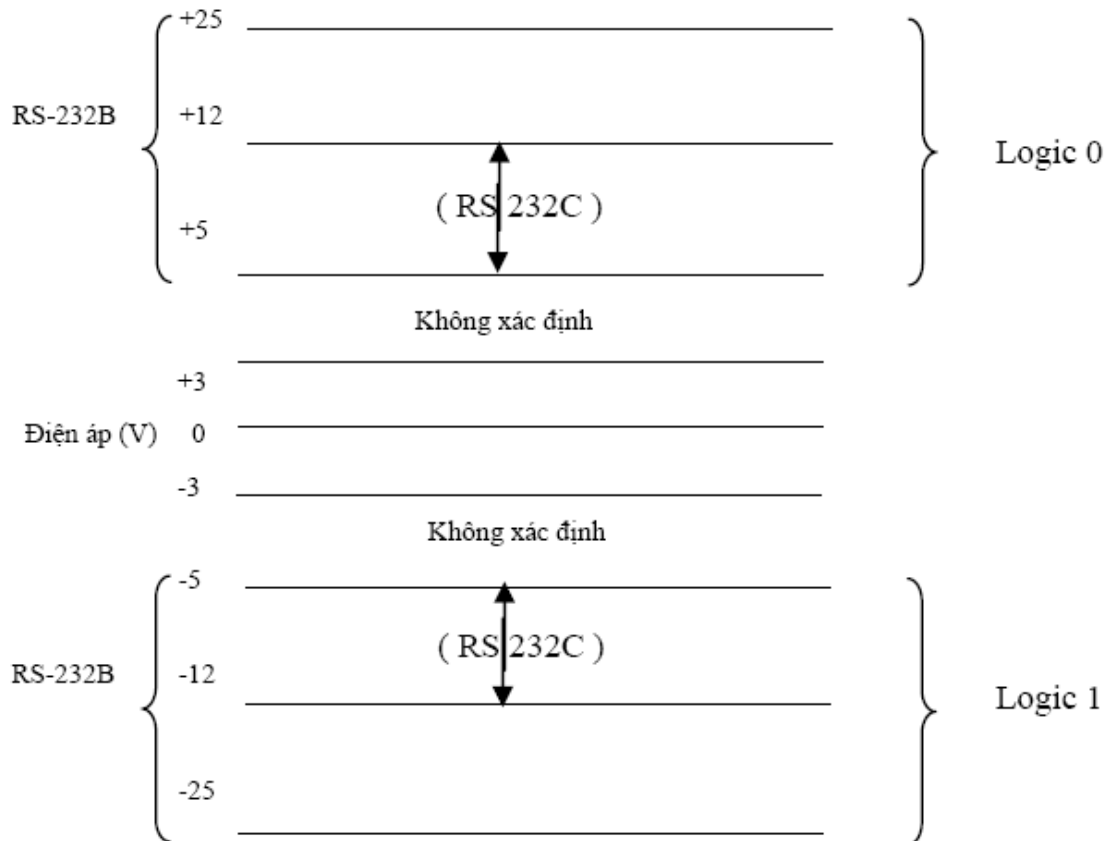
Status= Inportb(0x379)

1.2. Cổng nối tiếp

1.2.1. Các đặc trưng điện áp

+12V là mức logic 0 (LOW) Một trong những thông số quan trọng nhất của RS232 là mức điện áp trên đường truyền. RS232 đầu tiên sử dụng mức điện áp TTL giống như cổng song song. Chính vì thế ngay sau khi ra đời RS232 đã xuất hiện nhu cầu phải cải tiến. Ngoài mức điện áp thì tiêu chuẩn cũng quy định rõ giá trị của trở kháng tải đầu vào bus và trở kháng

ra của bộ phát và bộ đệm. Hướng cải tiến của mức điện áp là tăng giá trị của điện áp truyền để tăng khả năng chống nhiễu do vậy truyền được xa hơn.



Từ sơ đồ trên ta thấy cải tiến của RS232B là làm tăng mức điện áp từ $\pm 5V$ đến $\pm 25V$, Trong đó:

Mức logic 1 tính từ $-5V$ đến $-25V$.

Mức logic 0 tính từ $+5V$ đến $+25V$.

Các mức từ $-3V$ đến $+3V$ gọi là trạng thái chuyển tiếp. Các mức điện áp từ $\pm 3V$ đến $\pm 5V$ gọi là không xác định. Dữ liệu có mức điện áp rơi vào khoảng này sẽ dẫn đến kết quả không dự tính được và đây cũng là tình trạng hoạt động của những hệ thống được thiết bị kế sơ sài. Điều đáng chú ý ở đây là: Mức 1 ~ LOW, mức 0 ~ HIGH vì trước khi đưa vào xử lý còn có bộ nhớ đảo còn gọi là bộ nhớ chuẩn dạng tín hiệu.

Việc nâng mức điện áp của chuẩn RS232B dẫn đến sự hạn chế về tốc độ truyền, vì vậy người ta thấy giữa tốc độ truyền và khoảng cách truyền phải có sự dung hoà. RS232C là chuẩn hiện nay đang được áp dụng.

Điện áp sử dụng là $\pm 12V$. Trong đó:

$-12V$ là mức logic 1 (HIGH)

Cụ thể:

$+3V \rightarrow +12V$ là mức 0

$+5V \rightarrow +12V$ là mức tin cậy (của mức 0)

$-3V \rightarrow -12V$ là mức 0

$-5V \rightarrow -12V$ là mức tin cậy (của mức 1)

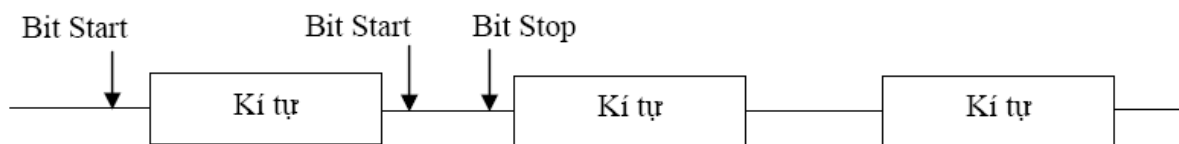
Bằng việc thu hẹp giới hạn điện áp đường truyền, tốc độ truyền dữ liệu được tăng lên đáng kể. Ngoài ra chuẩn RS232C cũng quy định trở kháng tải, giá trị này thuộc phạm vi 3000Ω đến 7000Ω ; đồng thời bộ đệm phải duy trì tăng điện áp tương đối lớn khoảng $30V/\mu s$.

Các vi mạch Motorola loại MC1488 và MC1489 đã hoàn toàn thoả mãn các thông số kỹ thuật này. Các yêu cầu về mặt điện được quy định trong chuẩn RS232C như sau:

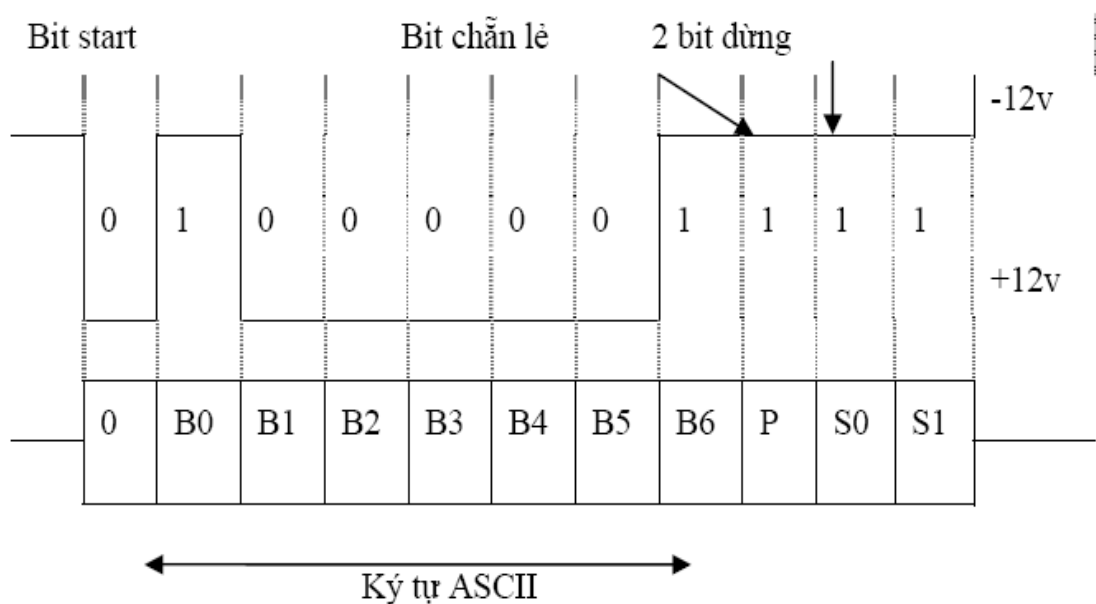
1. Mức logic 1 (mức dẫu) nằm trong khoảng $-3V \rightarrow -12V$; Mức logic 0 (Mức trống) nằm trong khoảng $+3V \rightarrow +12V$.
2. Trở kháng tải về phía bộ nhận của mạch phải nằm trong khoảng $3000\Omega - 7000\Omega$.
3. Tốc độ truyền nhận cực đại 100 Kbit/s.
4. Các lỗi vào của bộ nhận phải có điện dung $< 2500pF$.
5. Độ dài của cáp nối giữa máy tính và thiết bị ghép nối qua cổng nối tiếp không thể vượt quá 15 mét nếu không sử dụng modem.
6. Các giá trị tốc độ truyền dữ liệu chuẩn là 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 28800, ..., 56600 baud (bit/s).

1.2.2. Khuôn mẫu khung truyền

Đặc điểm của đường truyền dữ liệu qua cổng nối tiếp là tiến hành truyền và nhận trên các đường dẫn đơn lẻ, cho nên khi thiết bị truyền và thiết bị nhận được ghép nối với nhau thì đường truyền bên này sẽ được nối với đường nhận bên kia và ngược lại. Có như vậy mới hình thành được vòng kín của quá trình truyền dữ liệu. Để lưu ý mối quan hệ bất chéo tay như vậy người ta đã đưa vào dấu x ở giữa TD (TxD) và RD (RxD). Việc truyền dữ liệu qua cổng nối tiếp RS232 được tiến hành theo kiểu không đồng bộ, trong đó khuôn mẫu dữ liệu có bit bắt đầu, bit dừng được chỉ ra như hình vẽ sau:



Rõ ràng theo cách truyền này chỉ có một ký tự được truyền tại một thời điểm. Giữa các ký tự có một khoảng phân cách giữa chúng trong đó có chứa bit dừng, bit bắt đầu. Đầu tiên bộ truyền sẽ gửi một bit bắt đầu (bit Start) để thông báo cho bộ nhận biết là sau bit này sẽ là các bit dữ liệu có thể là 5,6 hoặc 7 bit. Tiếp theo là một bit chẵn lẻ và sau cùng là 1 hoặc 2 bit dừng. Điểm đáng chú ý là bao giờ bit bắt đầu cũng ở mức LOW. Khoảng thời gian phân cách của một bit đơn sẽ quyết định tốc độ truyền. Khoảng phân cách càng nhỏ thì tốc độ truyền càng lớn.



Bit bắt đầu ~ mức 0, tiếp theo là 7 bit dữ liệu 1000001, 1 bit chẵn lẻ 1, cuối cùng là 2 bit dừng 11.

Như vậy, toàn bộ khung truyền được phát ra là 01000001111. Bit chẵn lẻ dùng để kiểm tra phát hiện lỗi và sửa lỗi. Thực chất của quá trình này như sau: Khi kí tự được truyền thì máy tính sẽ đếm số kí tự 1 trong kí tự được truyền. Nếu số đó là chẵn => bit chẵn lẻ =1; Nếu số đó là lẻ => bit chẵn lẻ =0. Ở nơi nhận sẽ kiểm tra kí tự nhận được và đếm số 1, sau đó sẽ so sánh với bit chẵn lẻ. Nếu kết quả trùng khớp thì khung truyền coi như không mắc lỗi, ngược lại nó sẽ phát lệnh yêu cầu truyền lại khung truyền. Nếu tỷ lệ mắc lỗi càng nhiều thì tốc độ truyền càng giảm. Kỹ thuật mã lỗi chẵn lẻ theo kiểu này có một đặc điểm rất đơn giản, nhưng trong trường hợp bị mắc lỗi 2 lần liên hoặc 4 lỗi liên thì lại không phát hiện ra. Nhưng trên thực tế với 7 bit được truyền thì khả năng bị mắc 2 hoặc 4 lỗi là rất nhỏ có thể xem như không bao giờ xảy ra. Chính vì vậy, cách mã lỗi theo kiểu này vẫn được dùng phổ biến ở trong kỹ thuật truyền dữ liệu qua cổng nối tiếp.

Tốc độ truyền: Để đánh giá chất lượng của cuộc truyền dữ liệu qua cổng nối tiếp thì một trong những thông số đặc trưng quan trọng là tốc độ truyền/nhận dữ liệu. Trong kỹ thuật truyền dữ liệu qua cổng nối tiếp ta thấy có bit bắt đầu, bit dừng. Trong một số trường hợp có bit chẵn lẻ đã được bổ xung vào, như vậy có tới 10 bit được truyền trong khi chỉ có 7 bit dữ liệu, còn trong trường hợp sử dụng 2 bit dừng thì có tới 11 bit truyền trong khi chỉ có 7 bit dữ liệu. Như vậy nếu có 10 kí tự được gửi trong 1 giây và nếu như có 11 bit được sử dụng cho 1 kí tự thì tốc độ truyền thông sẽ là 110 bit/s. Như vậy giữa tốc độ truyền bit và tốc độ truyền kí tự là khác nhau.

Ngoài tốc độ truyền bit người ta còn sử dụng tốc độ baud. Đây là tên của một nhà kỹ thuật người Pháp đã giành nhiều công sức để nghiên cứu về truyền thông và người ta đã lấy tên ông để làm đơn vị truyền dữ liệu. Thông thường tốc độ bit và tốc độ baud là đồng nhất, chỉ trong trường hợp có môdem do có thêm quá trình biến đổi tín hiệu nên 2 tốc độ này khác nhau.

Tốc độ bps	Kí tự /s
110	11
300	30
600	60
1200	120
2400	240
4800	480
9600	960
19200	1920
56600	5660

1.2.3. Các vấn đề truyền thông

Chúng ta thấy rằng việc truyền thông nối tiếp đòi hỏi rất nhiều thao tác phải thực hiện. Ta phải chuyển một byte dữ liệu từ dạng song song thành dạng nối tiếp (bởi vì hầu hết các hệ thống số đều làm việc với các dữ liệu ở dạng song song). Tiếp theo ta phải tạo ra một lời tin theo đúng định dạng cho trước bằng cách thêm các bit Start, Stop, Parity phù hợp. Sau đó ta mới truyền dữ liệu đi dưới dạng nối tiếp. Cổng nối tiếp của máy PC có một vi mạch chuyên dùng để điều khiển truyền thông nối tiếp. Do đó, khi sử dụng cổng RS-232 của máy PC để truyền thông, tất cả công việc của chúng ta là gửi byte dữ liệu cần truyền ra thanh ghi dữ liệu của vi mạch này. Sau đó mọi thao tác của quá trình truyền thông kể trên sẽ được vi mạch thực hiện dựa theo những thiết lập trong quá trình khởi tạo.

Quá trình truyền một byte dữ liệu:

- Trong trạng thái rỗi, giá trị logic trên đường truyền luôn bằng 1. Để báo việc bắt đầu truyền dữ liệu, bên gửi đưa giá trị logic 0 lên đường truyền trong khoảng thời gian bằng độ dài một bit. Bit đó gọi là bit Start. Khi truyền với tốc độ 300 baud, một bit có độ dài là 3,3 ms, trong khi với tốc độ 9600 baud thì có độ dài 0,1 ms.
- Ngay sau bit Start, bên truyền gửi tiếp 8 bit dữ liệu kế tiếp nhau, bắt đầu bằng bit LSB. Tiếp sau đó bên truyền sẽ gửi tiếp một bit có giá trị logic 1 lên đường truyền và duy trì trong khoảng thời gian ít nhất là độ dài một bit. Ngay sau đó hoặc sau một khoảng thời gian bất kỳ, bit Start tiếp theo sẽ được gửi để bắt đầu truyền một byte mới.

1.3. Bus nối tiếp đa năng

1.3.1. Bus nối tiếp đa năng - Giới thiệu chung

Có thể nói MT từ khi ra đời đã không ngừng phát triển. Hiện nay máy tính PC vẫn đang được cải tiến nhằm nâng cao những tính năng của hệ thống. Những hướng chính là:

- + Tiếp tục cải tiến bộ vi xử lý cũng như đưa ra những bộ xử lý mới.
- + Cải tiến các hệ thống đồ họa, ví dụ: card AGP
- + Nâng cao tốc độ của đồng hồ hệ thống và của chính bộ xử lý.
- + Cải tiến các kiến trúc bus đặc biệt các cầu PCI.

+ Hoàn thiện công nghệ cắm và chạy (plug and play) và quá trình tự động cài đặt. Đặc biệt hoàn thiện cổng USB để trợ giúp cho việc dễ dàng ghép nối. Nếu như máy tính dùng nguồn AT có hai cổng RS 232 thì ở phía sau các máy tính đời mới thường dùng nguồn ATX đều có 2 ổ cắm USB. Cổng USB thực chất là BUS, bởi vì qua đó có thể đấu nối đồng thời rất nhiều thiết bị ngoại vi với những chủng loại khác nhau. Vì vậy, có thể gọi bus USB là bus nối tiếp đa năng theo đúng nghĩa của nó.

Bus USB nhằm thống nhất các kiểu ghép nối máy tính khác nhau về một dạng đầu nối và vì vậy khả năng ghép nối máy tính qua USB trở nên hết sức hấp dẫn. Các giao diện song song nối tiếp, các máy ghép hình ảnh số... đều có thể đấu nối vào bus USB. Vì vậy trong tương lai bus USB sẽ thực sự trở thành bus đa năng.

Về nguồn gốc USB được đưa ra sử dụng đầu tiên vào năm 1996, phải đến giữa năm 1998 mới thực sự được hỗ trợ đầy đủ và thể hiện vai trò của nó. Các thống kê kỹ thuật của USB đã được các công ty lớn cùng tham gia xây dựng. Trong đó phải kể đến Compaq, Digital Equipment, Northern, Telecom, IBM, Intel, Microsoft, NEC. Có thể nói bus USB đã nhanh chóng trở thành một chuẩn không chính thức. Người ta cũng sản xuất ra một card mở rộng cho phép cắm vào các máy tính đời cũ để tạo ra 2 cổng USB. Sau khi USB được giới thiệu đã có nhiều thiết bị sử dụng trong lĩnh vực điện tử dân dụng, truyền thông được thiết kế để nối vào với bus này và Microsoft đã viết phần mềm hỗ trợ cho USB từ năm 1998. Trong win 95 thì USB đã được hỗ trợ rất đầy đủ.

USB là một bus nối tiếp vì dữ liệu truyền trên bus tương tự như trong cổng nối tiếp. Cụ thể là theo từng bit một nối tiếp nhau. Nhưng có một điểm đáng lưu ý là dữ liệu được truyền trên cùng các đường dẫn theo 2 hướng trong khi theo tiêu chuẩn RS232 thì dữ liệu được truyền trên các đường dẫn khác nhau nhưng trên mỗi đường chỉ theo một hướng. Sự khác nhau cơ bản thể hiện ở chỗ các giao diện nối tiếp từ trước đến nay chỉ có thể sử dụng cho một thiết bị nhưng bus USB lại cho phép đấu nối đến 127 thiết bị. Vì vậy được gọi là một bus. Mỗi thiết bị đấu vào đều nhận một địa chỉ và thông qua địa chỉ này thiết bị có thể trao đổi dữ liệu với máy tính cũng như các thiết bị khác và địa chỉ này được mô tả bằng 7 bit.

Về mặt tốc độ, việc trao đổi dữ liệu qua bus USB nhanh hơn so với qua cổng RS 232. Trên thực tế vận tốc truyền có thể đạt được 12Mbps trên các đường dẫn dữ liệu. Dải thông sẽ được phân chia cho tất cả các thiết bị được đấu nối trên bus. Với bus USB loại 1.0 tốc độ truyền dữ liệu lên tới 12 Mbps, nhưng ở version 2.0 vận tốc đạt tới 480Mbps

vẫn giữ được tính tương thích ở phiên bản 1.0. Bus USB có mối liên quan chặt chẽ với đặc tính cắm để chạy ở các máy tính PC đời mới trong khi máy tính đang hoạt động, thiết bị có thể được đấu vào hoặc tháo ra mà không cần tắt điện nguồn nuôi trong máy tính. Đặc tính được gọi là đấu ngắt nóng. Hệ thống tự nhận biết một thiết bị mới được đấu vào thiết bị USB và lập tức nạp phần mềm điều khiển hay tệp đệm thích hợp.

Những đặc tính cơ bản của bus USB có thể kể ra là:

+ Các bộ truyền đảm thời có thể hiểu là truyền liên tục hỗ trợ các tín hiệu video và âm thanh với các đường truyền đẳng thời thì các thiết bị truyền dữ liệu theo kiểu đảm thời và theo kiểu đoán trước .

+Bus USB hỗ trợ các thiết bị không đẳng thời, các thiết bị có quyền ưu tiên cao nhất (các thiết bị đảm thời cũng như đẳng thời có thể tồn tại cùng một thời điểm).

+Các thông số kỹ thuật cắm chạy các cáp và cách kết nối đều được tiêu chuẩn hoá rộng rãi trong công nghiệp.

+Các Hub được sản xuất thành nhiều tầng với khả năng mở rộng các mức gần như vô tận và các thao tác xảy ra đồng thời.

+Tốc độ truyền là 12Mbps với các kích thước gói dữ liệu khác nhau.

+Hỗ trợ nhiều yêu cầu về giải thông từ một vài Mbps đến 19 Mbps.

+Hỗ trợ tốc độ truyền dữ liệu trên một phạm vi rộng các giá trị thông qua việc điều tiết kích thước bộ đệm gói dữ liệu và cơ chế tiềm ẩn(latency), có khả năng cắm nóng (hot plug). Nghĩa là cho phép thiết bị ngoại vi có thể được nối mà không cần phải tắt nguồn nuôi cung cấp điện cho máy tính. Có thể đấu, ngắt và thay đổi cấu hình của thiết bị ngoại vi một cách linh hoạt .

+Khả năng quản lý năng lượng được tăng cường với các chế độ nghỉ trên phạm vi hệ thống.

+Tự nhận dạng thiết bị ngoại vi kiểu mới, tự động vẽ bản đồ chức năng đối với phần mềm điều khiển và cấu hình.

+Hỗ trợ cho các thiết bị loại khác nhau với nhiều công nghệ khác nhau.

+Điều khiển luồng dữ liệu thông qua bộ đệm bằng việc quản lý giao thức đặt sẵn bên trong.

+Có thể xử lý lỗi và hoàn trả lỗi.

+Hỗ trợ khả năng nhận dạng các thiết bị mắc lỗi.

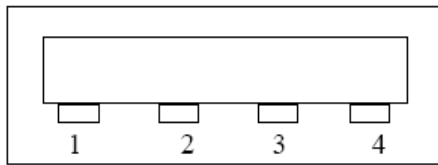
+Giao thức đơn giản trong việc thực hiện và tích hợp.

USB là một kiến trúc bus cân bằng trong quá trình hoạt động máy chủ USB đóng vai trò điều khiển dải thông của hệ thống. Mỗi thiết bị được gán một địa chỉ mặc định khi thiết bị USB được cấp điện lần đầu hoặc được đặt lại. Một đặc điểm cơ bản khác nữa của USB điện áp nguồn nuôi (+5v) có thể nhận được từ bus. Các thiết bị có công suất tiêu thụ nhỏ có thể sử dụng trực tiếp điện áp trên bus mà không cần có nguồn nuôi riêng.

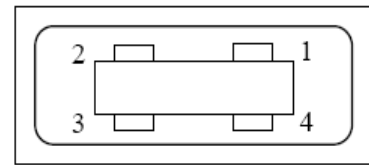
1.3.2. Đầu nối và cáp tín hiệu

USB có hai kiểu đầu nối khác nhau được gọi là A,B. Hệ thống ấy được thiết kế sao cho không xảy ra hiện tượng đầu nối nhầm. Bus USB sử dụng cáp nối 4 sợi dây để nối với các thiết bị ghép nối. Trong đó có một cặp đường truyền 2 sợi xoắn được dùng làm đường dẫn dữ liệu vi phân, ký hiệu là D+ và D-. Còn một cặp kia dùng làm đường 5V và đường nối đất chung. Cáp nối luôn được thực hiện liên kết 1:1. Sự sắp xếp các chân ở đầu nối cáp tuân theo những quy định sau:

Hai ổ cắm USB phía sau máy tính đời mới nhất đều là kiểu A, qua đó có thể đấu trực tiếp thiết bị USB vào máy tính. Các thiết bị có tốc độ thấp như chuột có thể đấu thẳng vào ổ cắm này bằng một phích cắm cũng kiểu A.



Kiểu A



Kiểu B

Trong các trường hợp khác thiết bị thường có một ổ cắm kiểu B, muốn nối với máy tính phải sử dụng một cáp kiểu A,B. Trong trường hợp cần nối dài cáp, tức là để tăng khoảng cách giữa máy tính PC tới thiết bị ghép nối, người ta sử dụng cáp A,A. Cho đến nay các cáp USB đều được các nhà sản xuất cung cấp dưới dạng hoàn chỉnh trên đó đầu cắm, độ dài, chất lượng bọc kim chống nhiễu đều không thể thay đổi được. Vì vậy, tùy theo mục đích sử dụng ta phải lựa chọn thông số cáp cho chính xác từ chiều dài cho đến đầu nối.

Chân	Tên gọi	Màu dây	Mô tả
1	vcc	Đỏ	+ 5v DC
2	D-	Trắng	DL -
3	D+	Xanh lục	DL+
4	CND	Đen	Nối đất

Qua ổ cắm USB sau máy tính có thể lấy ra điện áp + 5v với dòng điện tiêu thụ 100 mA. Trong một số trường hợp có thể lấy tới 500 mA. Hai đường dẫn dữ liệu D +, D - cho phép đầu nối với các linh kiện USB đặc biệt chẳng hạn như là một số vi điều khiển tín hiệu ở chân D +, D - là các tín hiệu vi phân với mức điện áp = 0/ 3,3 v. Điện áp nguồn nuôi cho bus có thể tăng đến 5,25 v và khi chịu dòng tải lớn có thể giảm xuống 4,2 v. Một vi mạch ổn áp trong trường hợp này có thể tạo ra một điện áp ổn định +3,3 v. Toàn bộ hệ thống có thể thiết kế sao cho khi chịu dòng tải lớn điện áp nguồn cũng không vượt quá + 4,2v. Khi thiết bị ghép nối cần dòng tiêu thụ >100mA cần xem xét kỹ khả năng cung cấp và chịu tải của các linh kiện phía trong MT để tránh những hậu quả đáng tiếc có thể xảy ra. Khi ghép nối một thiết bị với bus USB ta thường phải phân biệt rõ các thiết bị sử dụng nguồn nuôi riêng chẳng hạn như máy in với các thiết bị nhận điện áp nguồn nuôi qua bus. Trong một số trường hợp cả hai chế độ nguồn nuôi có thể cùng tồn tại để lựa chọn theo cách thiết kế của bus. Dòng tiêu thụ lấy từ bus được tự động hạn chế. Khi dòng tiêu thụ vượt quá giới hạn cho phép thì điện áp cung cấp cũng tự động ngắt.

1.3.3. Truyền dữ liệu nối tiếp

Một đặc điểm khác nữa của bus USB là chỉ có một máy chủ nghĩa là mọi hoạt động trên bus đều xuất phát từ máy tính PC quản lý. Dữ liệu được gửi lên cũng như nhận từ bus theo những gói nhỏ chứa 8 -> 256 byte. Máy tính PC có thể yêu cầu dữ liệu gửi đến từ một thiết bị nhưng ngược lại không một thiết bị nào có thể tự gửi dữ liệu đi.

Toàn bộ lượng dữ liệu đều có một khung đúng bằng 1ms. Trong phạm vi một khung nhiều gói dữ liệu kế tiếp dành cho các thiết bị khác nhau có thể được xử lý, trong đó có những gói dữ liệu cần gửi với tốc độ thấp, có những gói dữ liệu cần gửi với tốc độ cao cùng tồn tại trong một khung.

Khi cần ghép nối nhiều thiết bị USB với máy tính, ta cần có một hộp phân phối hay còn gọi là Hub cho phép tránh xảy ra tình trạng tốc độ tín hiệu cao được chuyển giao tới thiết bị có tốc độ thấp.

1.3.4. Phần cứng, phần mềm và kết nối hệ thống

Khi nối một thiết bị vào bus thì máy chủ USB sẽ tương tác với thiết bị USB thông qua bộ điều khiển của máy chủ. Khi đó máy chủ chịu trách nhiệm về những công việc sau:

Phát hiện việc kết nối hoặc hủy bỏ của các thiết bị USB

Quản lý việc điều khiển dòng dữ liệu giữa máy chủ và các thiết bị USB Thống kê trạng thái và tính hoạt động của hệ thống.

Cung cấp dòng điện đã được nạp để hạn chế công suất cho các thiết bị USB được kết nối. Trên máy chủ có phần mềm quản lý hệ thống USB, cụ thể quản lý sự tương tác giữa các thiết bị USB và phần mềm thiết bị dựa trên máy chủ. Có năm vùng tương tác giữa phần mềm hệ thống USB và phần mềm thiết bị cụ thể là:

Điểm danh và định cấu hình thiết bị.

Truyền dữ liệu trong chế độ đăng thời.

Truyền dữ liệu trong chế độ không đồng bộ.

Quản lý năng lượng Quản lý thông tin về thiết bị và về bus.

1.3.5. Chuẩn USB 2.0

Phiên bản USB 1.0 ra đời vào năm 1996 đã là một thành công rất lớn và bây giờ đang là công tiêu chuẩn trên đa số các máy tính PC đời mới. Hạn chế chính của phiên bản này là tốc độ truyền vẫn còn tương đối chậm. Đặc biệt khi tích hợp nhiều thiết bị ngoại vi thành một hệ thống kết nối riêng rẽ. Vì vậy, phiên bản 2.0 đã được ra đời cho phép đạt tới tốc độ truyền dữ liệu trên 480Mbps trong khi vẫn giữ tính tương thích với phiên bản 1.0. Các đặc tính chính của phiên bản 2.0 là:

Tốc độ thấp(1,5Mbps) dùng cho các thiết bị tương tác. thông thường 10->100Kbps.

Tốc độ cao(12Mbps) dùng cho các ứng dụng có các tín hiệu diện thoại và âm thanh. Thông thường 500Kbps->10Mbps.

Tốc độ rất cao(470Mbps) dùng cho các ứng dụng video và bộ nhớ. Thông thường 250 - >400Mbps.

Rõ ràng là phiên bản USB 2.0 sẽ đẩy nhanh quá trình hướng tới một máy tính PC sử dụng trong tương lai tức là chỉ cần dùng một loại cổng USB cho tất cả các thiết bị ghép nối.

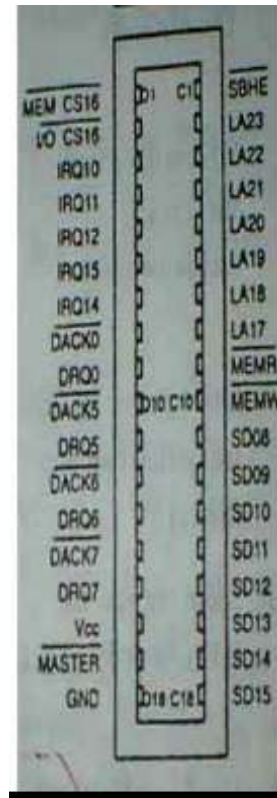
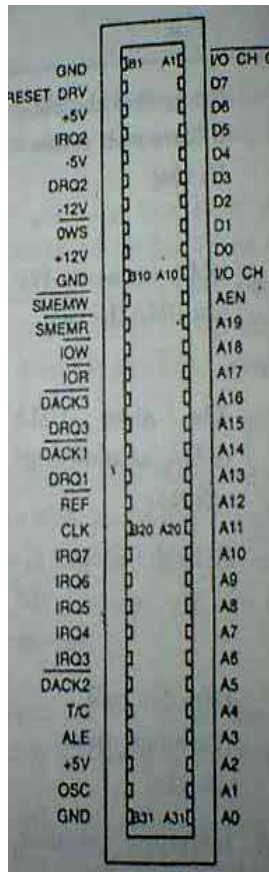
1.4. Khe cắm mở rộng

1.4.1. BUS ISA 16 bit

Từ tên gọi cho thấy đây là loại bus được kiến trúc theo tiêu chuẩn công nghiệp (Industry Standard Architecture). Một số tài liệu gọi bus PC là bus ISA 8 bit thì loại này được phân biệt rõ là ISA 16 bit. Tác giả của loại bus này là công ty IBM. Công ty này đã thiết kế ra bus ISA để dùng cho máy tính AT Advanced Technology dựa trên cơ sở của bộ xử lý 80286. Điểm mạnh nổi bật của loại bus này là có thể cho phép cùng một lúc xử lý hoặc trao đổi với 16 bit dữ liệu nghĩa là gấp đôi bus PC.

Để đảm bảo tính tương thích so với bus PC người ta đưa thêm vào một rãnh cắm thứ hai thẳng hàng so với rãnh cắm thứ nhất và có chứa 36 tiếp điểm xếp thành hai hàng mỗi hàng 18 tiếp điểm. Trên rãnh cắm thứ hai có chứa 8 bit dữ liệu và 4 đường dẫn địa chỉ. Như vậy ở trên bus ISA có tổng cộng 16 bit dữ liệu và 24 bit địa chỉ. Tốc độ truyền dữ liệu được quy định bởi tốc độ đồng hồ cố định.

Như vậy trên bus ISA có một bus dữ liệu 16 bit và chính vì lẽ đó đôi khi bus này còn gọi là bus ISA 16 bit để phân biệt với bus 8 bit ngoài ra nó còn 24 bit địa chỉ chỉ cho phép quản lý 16 Mb bộ nhớ. Giống như bus PC, nó cũng sử dụng tốc độ đồng hồ cố định nhưng khác về giá trị, cụ thể là 8MHz. Một lợi thế rất lớn của card mở rộng dùng với bus PC là chúng có thể cắm được vào rãnh cắm ISA bởi vì bus ISA bảo đảm tính tương thích kế thừa. Có thể nói card mở rộng ISA rất phổ biến bởi vì chúng thể hiện được tính năng ưu việt đối với hầu hết các ứng dụng ghép nối.



Các linh kiện được sử dụng trên card mở rộng ISA thường rất rẻ, cho nên có thể nói trên thực tế việc ghép nối bằng card mở rộng ISA tỏ ra là một công nghệ đã qua thử thách và đáng tin cậy. Ứng dụng tiêu biểu của card mở rộng ISA 16 bit có thể kể ra là: card vào ra nối tiếp và song song, card âm thanh, card mạng...

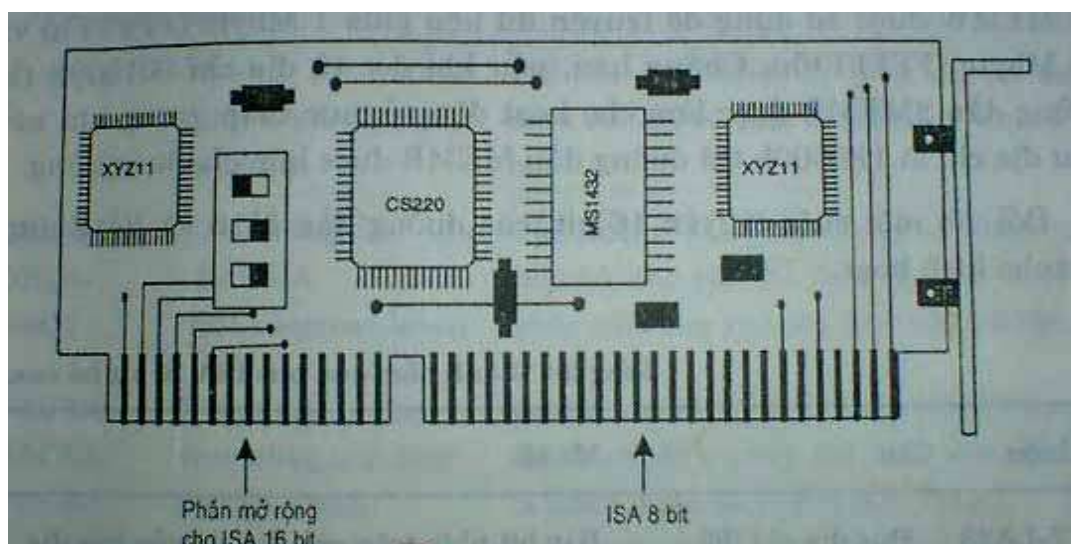
Như vậy để đảm bảo tính tương thích với loại bus PC 8 bit thì rãnh cắm mở rộng ISA 16 bit bao gồm 2 phần: Phần thứ nhất giống hệt rãnh cắm PC, các tiếp điểm ở hai mặt được đánh số theo A và B. Phần thứ hai bao gồm 36 tiếp điểm chia làm 2 hàng mỗi hàng 18 tiếp điểm đánh số là C và D.

Đầu nối bus PC chuẩn có chứa các dãy A và B. Trên dãy A có 20 địa chỉ đánh từ A0 đến A19 và 8 đường dẫn dữ liệu D0 đến D7. Dãy B có chứa các đường dẫn ngắt đánh số từ IRQ0 đến IRQ7, các đường cấp nguồn nuôi và các đường dẫn điều khiển khác. Phần rãnh cắm bổ xung bao gồm 2 dãy C và D trên đó có thêm 7 đường dẫn địa chỉ từ A17 đến A23 và các đường dẫn dữ liệu từ D8 đến D15 và các đường dẫn ngắt từ IRQ10 đến IRQ14. Như vậy bus ISA 16 bit có những đặc điểm chính sau:

Sử dụng một bus dữ liệu 16 bit từ D0 đến D15. Một bus địa chỉ 24 bit từ A0 đến A23.

Tín hiệu giữ nhịp CLK được đặt là 8,33 MHz.

Các đường dẫn SMENR và SMEMW được sử dụng để truyền dữ liệu đối với 1 Mb thấp nhất của bộ nhớ. Cụ thể là từ 0 đến FFFFh. Còn các tín hiệu MENR và MEMW được sử dụng để truyền dữ liệu trong vùng bộ nhớ giữa 1 Mb(FFFFh) và 16 Mb(FFFFFFh). Chẳng hạn nếu khi đọc từ địa chỉ 001000h thì đường dẫn SMENR được làm cho hoạt động ở mức thấp, trong khi nếu như địa chỉ là 1F0000h thì đường dẫn SMENR được làm cho hoạt động.



1.4.2. BUS PCI

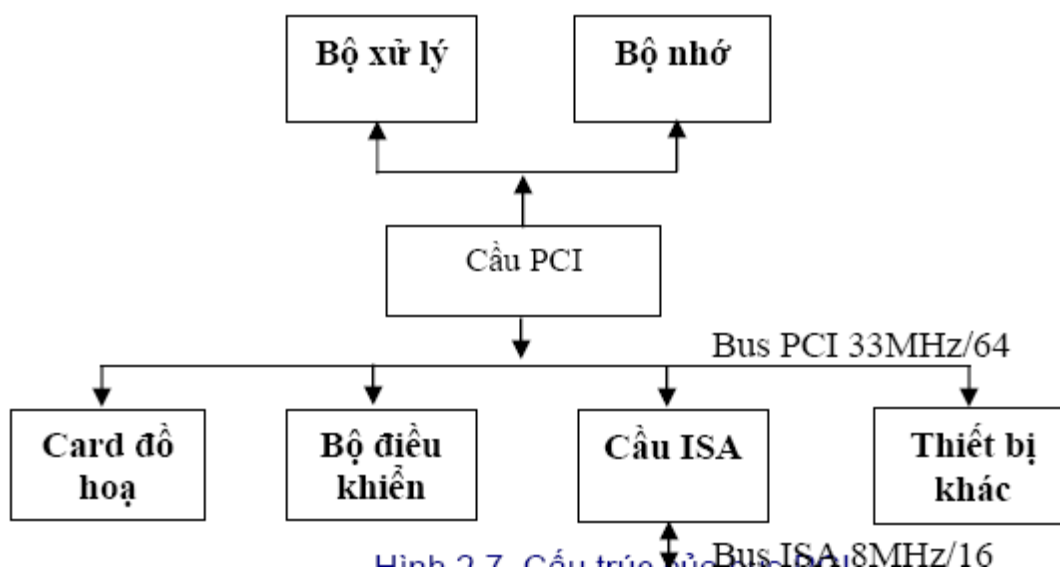
Tác giả của bus PCI (Peripheral Component Interconnection) là công ty Intel. Công ty này đã xây dựng lên một tiêu chuẩn ghép nối mới có tên là bus cục bộ PCI hay thường gọi tắt là bus PCI dùng cho bộ xử lý Pentium. Bus này được thiết kế với chỉ tiêu: Tốc độ hoạt động nhanh.

Số bit truyền trên bus cao (64 bit).

Mục đích là đẩy nhanh tốc độ truy nhập đáp ứng nhu cầu tăng tốc độ trao đổi dữ liệu giữa bộ nhớ, bộ xử lý, bộ điều khiển đĩa và card màn hình.

Một bộ vi mạch ghép nối được sử dụng cho loại bus này là chip PCI 82430 cho phép ghép nối trực tiếp với bus.

Đặc điểm: Giống VESA ở chỗ là cùng truyền dữ liệu bằng cách sử dụng đồng hồ hệ thống nhưng lại thể hiện ưu điểm hơn hẳn so với bus VESA là có thể hoạt động ở chế độ trên 32 bit thậm chí cho đến 64 bit. Do tốc độ truyền cao nên trong các máy tính có cấu trúc bus PCI có thể hạn chế các card mở rộng ghép nối xuống còn 2 hoặc 3 rãnh. Thông thường chỉ có card màn hình và card điều khiển đĩa cứng là sử dụng bus PCI. Nếu dữ liệu được truyền trong chế độ 64 bit và ở tốc độ đồng hồ là 33 MHz thì tốc độ truyền dữ liệu cực đại có thể đạt đến 264 Mbyte/s. Để phối hợp khả năng truy nhập của các thiết bị khác nhau lên bus PCI và bus ISA người ta sử dụng một khối gọi là cầu vào ra và bố trí như sau:



Hình 2.7. Cấu trúc Bus PCI

Do có thể truyền ở chế độ 64 bit có nghĩa là bus PCI ít nhất phải có 64 đường dẫn dữ liệu. Vì vậy bình thường ở bus PCI có kích thước tăng thêm gấp đôi. Tuy vậy, kích thước thực không thể tăng quá mức và người ta phải giải quyết bằng cách tăng mật độ chân trên 1 đơn vị diện tích. Chính vì lẽ đó mà chân cắm của bus PCI gần nhau hơn và không còn tương thích được với các rãnh cắm PC hoặc ISA.

Tốc độ truyền cực đại trên bus này có thể đạt tới 264 Mbyte/s. Tuy vậy, tốc độ này chỉ thực sự đạt được khi dùng phần mềm 64 bit chạy trên hệ thống có chứa bộ xử lý Pentium.

Các rãnh cắm PCI đầu tiên được lắp ráp trên mainboard với bộ xử lý 80486, mà bộ xử lý này chỉ sử dụng chế độ hoạt động 32 bit do vậy cũng chỉ đạt đến tốc độ cực đại 132 Mbyte/s.

Có thể nói bus PCI là kết quả cải tiến của các bus dùng cho máy tính PC đã có sẵn, nhưng về mặt logic thì khác hẳn với bus ISA và bus VESA. Rãnh cắm PCI có rất nhiều chân: Loại 32 bit có 62 chân trên mỗi hàng và tổng cộng 124 chân. Loại 64 bit có 94 chân x 2=188 chân. Thông thường bộ xử lý Pentium chạy với tốc độ đồng hồ hệ thống là 33MHz hoặc 50MHz.

Cần chú ý rằng bus VESA chỉ có thể hoạt động ở tốc độ truyền cực đại ứng với tần số 33MHz. Bởi lẽ mật độ các chân cắm ở rãnh cắm mở rộng PCI và VESA rất cao, cho nên khả năng để tự chế tạo ra các card mở rộng trong điều kiện không chuyên nghiệp để ghép nối với các bus này là rất nhỏ. Vì vậy hai loại bus này không được trình bày chi tiết mặc dù các mainboard được chế tạo gần đây đều có hai đến ba rãnh cắm mở rộng PCI.

1.4.4. Ghép nối qua khe cắm mở rộng

Điều kiện tiên hành: Phải có một bản mạch mở rộng, card phải hoạt động tốt. Phải có phần mềm cài đặt thích hợp để chính thức đăng ký card đó vào trong hệ thống máy tính.

Cách giải quyết:

Một số loại card mở rộng thông dụng như card modem, card âm thanh, card vào ra thì chọn giải pháp mua là thuận lợi nhất bởi giá thành giảm mà chất lượng lại đảm bảo, lý do là các card đó được sản xuất trong điều kiện công nghiệp số lượng lớn nên giá thành hạ ngoài ra việc kiểm tra xuất xưởng được tiến hành nghiêm chỉnh bởi nhà sản xuất.

Với một số mục đích chuyên dụng nghĩa là đối tượng ứng dụng tương đối hẹp và số lượng cần không nhiều thì ta thường chọn giải pháp tự thiết kế và tự làm, ví dụ card dùng trong kỹ thuật đo lường, thu thập số liệu đo lường, card điều khiển đối tượng cụ thể. Có thể về mặt kỹ thuật không phức tạp lắm nhưng ngược lại ta mua thường rất đắt hoặc không thể tìm mua được.

Trong kỹ thuật ghép nối thông dụng thường người ta chỉ sử dụng rãnh cắm ISA 8 bit hoặc 16 bit. Trên các bản mạch chính được chế tạo gần đây thường cũng có ba rãnh cắm ISA 16 bit đặt sẵn trên bản mạch chính. Đây chính là chỗ lắp vào các card biến đổi D/A và A/D hoặc là các card để tạo ra các cổng ghép nối khác như tạo ra bus GPIB hoặc RS-485... Sở dĩ bus ISA được dùng nhiều bởi vì nó có cấu trúc cũng như kích thước hình học tương đối đơn giản. Chính vì vậy trong kỹ thuật ghép nối nhiều tài liệu chỉ trình bày đến loại rãnh cắm này

CÂU HỎI VÀ BÀI TẬP

Câu 1 : Tìm hiểu DTE và DCE và các cách kết nối DTE với DEC

Câu 2 : Xác định khuôn dạng dữ liệu và biểu đồ tín hiệu khi truyền ra ngoài thông qua cổng COM chuỗi ký tự : “ACDEF”.

Câu 3: Xây dựng thủ tục thiết lập các thông số cổng COM.

Câu 4: Tìm hiểu ActiveX để lập trình cổng COM.

Câu 5: Tìm hiểu các thanh ghi của bộ UATR 8250

Câu 6: Tìm hiểu cơ chế lập trình cho UATR 8250.

Câu 7 : Tìm hiểu HID trong vấn đề lập trình USB.

Câu 8 : Trình bày các bước thiết lập thông số cổng USB

Câu 9: Xây dựng thủ tục xác định và định địa cho các thiết bị kết nối với máy tính thông qua cổng USB.

Câu 10: Xây dựng thủ tục cho phép gửi dữ liệu ra ngoài qua 1 địa chỉ cổng USB được định trước.

Câu 11: Xây dựng thủ tục cho phép nhận dữ liệu từ ngoài vào qua 1 địa chỉ cổng USB được định trước.

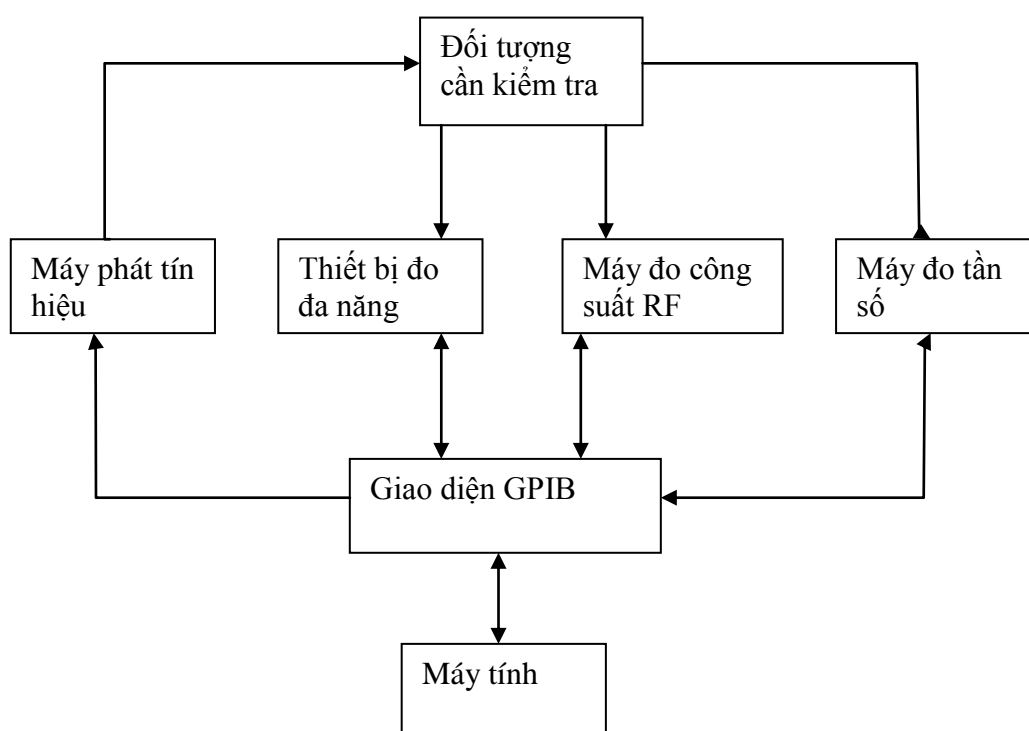
Chương 2. BUS GHÉP NỐI ĐA NĂNG

2.1. Đặt vấn đề

Một loại Bus được sử dụng nhiều trong các hệ thống đo lường với chất lượng cao là bus ghép nối đa năng, hay còn gọi là bus GPIB (General Purpose Interface Bus). Có thể nói bus GPIB là một hệ bus chuẩn cho phép thiết bị đo của nhiều hãng khác nhau cùng ghép nối để hình thành một hệ đo lường, kiểm tra và điều khiển.

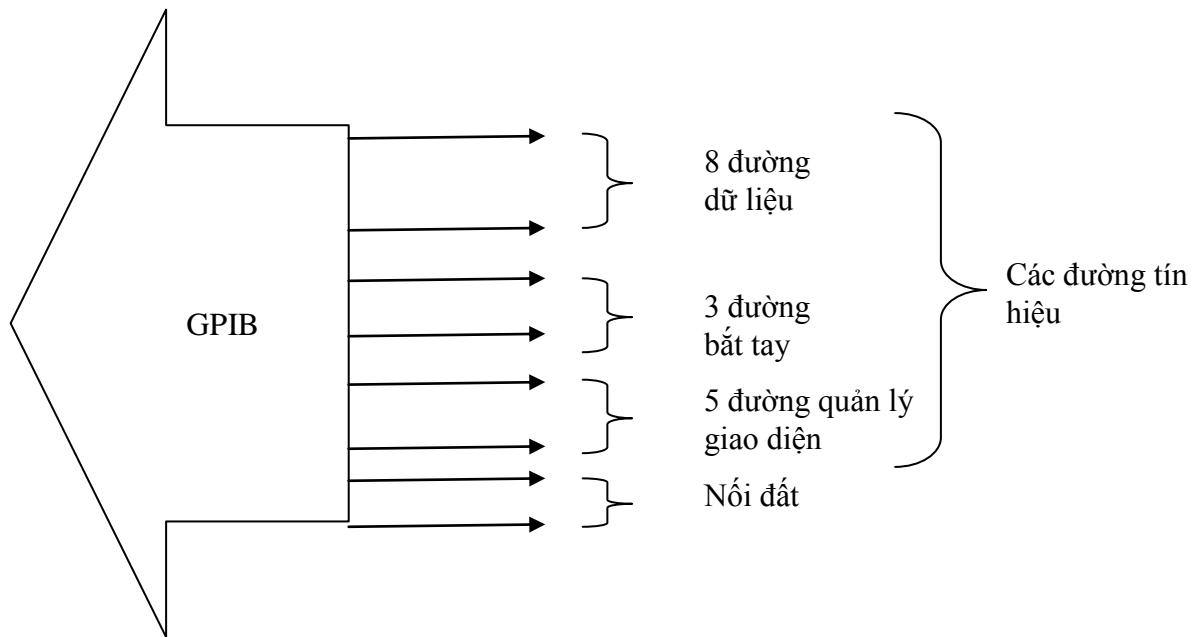
Ý tưởng khi xây dựng bus IEEE-488/GPIB là để kết hợp các thiết bị đo lường kiểm tra dùng trong công nghiệp và thương mại, với các máy tính để hình thành một hệ thống theo những tùy chọn khác nhau, có khả năng thực hiện các quá trình kiểm tra và đo lường dưới sự điều khiển của chương trình máy tính, thường được viết bằng BASIC.

2.2. Tiêu chuẩn IEEE-488 truyền thông



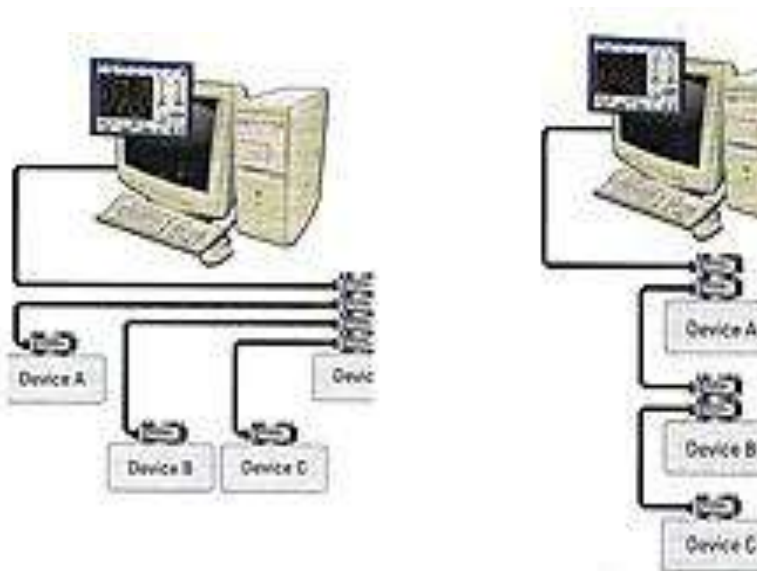
Các mức logic trên bus nói chung đều giống mức TTL : mức thấp nhỏ hơn hoặc bằng 0.8V, trong khi mức cao lớn hơn 2V. Tín hiệu logic có thể được nối với các thiết bị đo qua dây cáp nhiều sợi với chiều dài tổng cộng tới 20m. Thông thường các thiết bị đo phải đặt các nhau ít nhất là 2m. Nếu như các dây cáp đều có chiều dài ở mức thấp nhất thì chiều dài tính ra m của các dây cáp sẽ gấp 2 lần số các thiết bị được ghép nối trên hệ thống. Hầu hết các hệ thống IEEE-4888/ GPIB hoạt động bình thường với tốc độ truyền 250 kbyte/s hoặc nhanh hơn nếu chấp nhận 1 số hạn chế cụ thể.

Bus GPIB bao gồm 24 đường dẫn, các đường này được nối với đầu nối 24 chân. Các thiết bị đo lường và kiểm tra được dự tính để dùng với bus GPIB đều có lắp sẵn một số đầu nối 24chân ở mặt sau. Đồng thời cũng có 1 chuyển mạch 2 hàng (DIP) để định địa chỉ cho bus GPIB được lắp sau ở mặt sau, thường ở ngay cạnh đầu nối. Nhiệm vụ của chuyển mạch là đặt địa chỉ nhị phân 5bit, mà thiết bị được định vị trí trong hệ thống, để có thể quyết định : có hoặc không có thiết bị, hoặc chỉ là một thành viên nghe hoặc chỉ là thành viên nói và quy định một số chi tiết cụ thể khác.

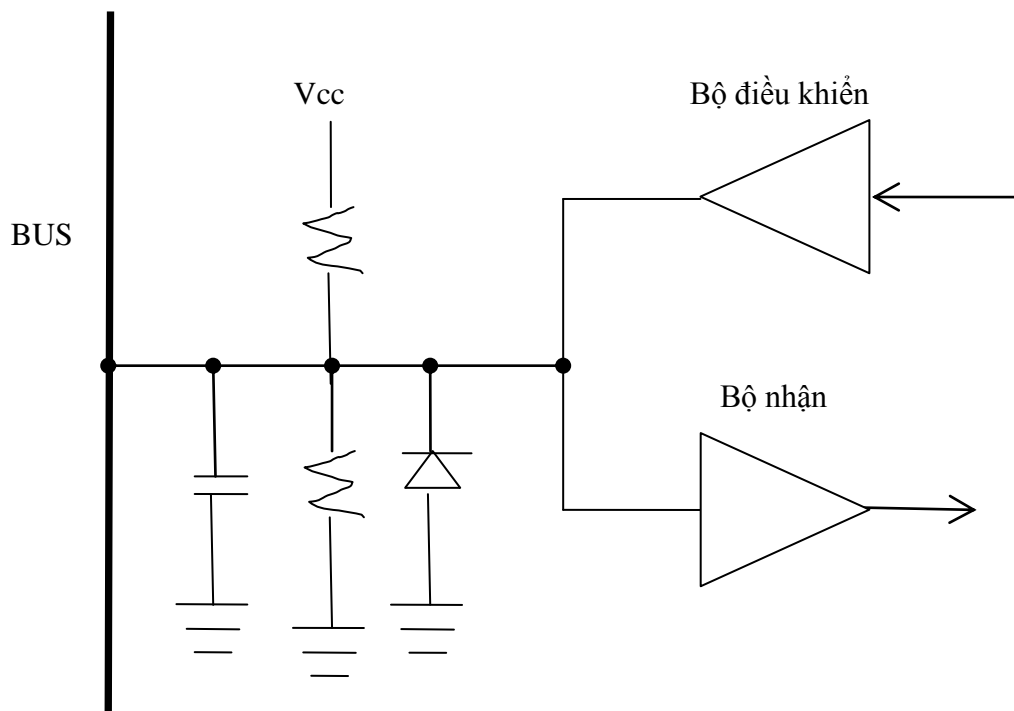


2.3. Cấu hình và hoạt động của Bus GPIB

Hệ thống IEE – 488/GPIB có thể được chia ra 2 loại cấu hình cơ bản : nối tiếp và hình sao. Các cấu hình dạng này được tạo ra nhờ cách đấu nối dây cáp giữa các thành viên của bus cụ thể (giữa các thiết bị đo lường và máy tính điều khiển). Cấu hình theo kiểu nối tiếp ghép nối các thiết bị thành một dãy kế tiếp nhau, trong đó đầu nối cho một thiết bị đo tiếp theo được cắm vào đầu nối của một thiết bị đo đứng trước đây. Trong cấu hình dạng sao, các thiết bị được nối từ một điểm chung.



Các bus trong hệ thống IEEE-488/GPIB có thể chia thành 3 loại : bus dữ liệu, bus bắt tay và bus để quản lý. Mỗi đường dẫn trong từng bus đều có 1 vạch điện tương tự với mạch điện. Bên cạnh điốt bảo vệ mắc song song vào điện dung phân bố ký sinh, cũng có các điện trở nối lên nguồn và nối xuống mass, xác định một các có hiệu quả trở kháng lỗi và đã tiêu chuẩn hóa.



Được nối tới đường bus là các mạch điện của thành viên nhận và các mạch điện của bộ điều khiển. Các mạch này tương tự với các phần tử logic TTL, bảo đảm việc nhập vào và xuất dữ liệu ra các thiết bị đo. Bộ điều khiển có một lối ra đầu vào bus và được thiết kế trên cơ sở một linh kiện 3 trạng thái. Linh kiện đó sẽ không phản ứng gì cho đến khi có lệnh đóng mạch. Như vậy, lối ra 3 trạng thái sẽ trôi nổi ở trở kháng cao cho đến khi được đóng mạch. Bộ nhận là 1 bộ đệm không đảo với lối vào trở kháng cao.

Cách bố trí của các bộ điều khiển và các bộ nhận theo cách này bảo đảm có được nhờ tải thấp từ bus và đây là lý do vì sao có thể đấu nhiều mạch tải vào một đường bus.

CÂU HỎI VÀ BÀI TẬP

1. Trình bày tiêu chuẩn IEEE-488 truyền thông
2. Trình bày cấu hình và hoạt động của Bus GPIB

Chương 3. VÒNG ĐO ĐIỆN ÁP

3.1. Truyền dữ liệu bằng vòng dòng điện

Thay cho việc dùng mức điện áp, việc truyền dữ liệu còn có thể được tiến hành nhờ sự chuyển mức dòng điện (thay đổi giá trị dòng điện).

Có thể tìm hiểu vòng dòng điện thông qua ví dụ cụ thể là vòng dòng điện 20 mA. Giao diện vòng dòng 20 mA ra đời trước khi RS-232 được xem là một tiêu chuẩn. Giao diện này còn được gọi là giao diện TTY hoặc thường được gọi là *vòng dòng điện*, mô tả trạng thái login qua tác động cho không cho 1 dòng điện với cường độ 20mA đi qua, nghĩa là việc gửi dữ liệu có thể quan niệm 1 cách đơn giản như việc đóng mạch để cho dòng điện chạy qua. Giao diện dòng 20mA được chia thành 2 phần : chủ động (có nguồn dòng không đổi 20 mA) và bị động (nguồn dòng không đổi 20 mA được tạo ra ở phía thiết bị ghép nối). Như vậy, bao giờ cũng một trong các thành viên kia là bị động. Để có thể chủ động tạo thành một mạch kín đối với dòng điện loại giao diện dòng 20 mA dùng cho máy tính PC có chứa 1 bộ biến đổi DC/DC, qua đó giao diện cung cấp năng lượng ở 2 chân nối để các vòng gửi và nhận có thể hoạt động

Ở giao diện dòng 20 mA có những tín hiệu sau :

- TxD (+) dữ liệu gửi (đường dẫn về)
- TxD (-) dữ liệu gửi (đường dẫn đi)
- RxD (+) dữ liệu nhận (đường dẫn đi)
- RxD (-) dữ liệu nhận (đường dẫn về)

Thông thường thì ở giao diện dòng 20 mA không có tín hiệu dùng để bắt tay khi thiết lập đường truyền. Với giao diện này, việc truyền dữ liệu trên khoảng cách cỡ 1.000m hoàn toàn có thể thực hiện được bởi vì nhiễu sinh ra trên đường truyền ảnh hưởng cảm ứng lên cả đường dẫn tín hiệu đi cũng như đến và do vậy bị loại trừ. Muốn thế đường dẫn đi và đường dẫn về phải đặt sát nhau, sao cho khi một tín hiệu nhiễu xuất hiện thì cùng ảnh hưởng lên cả 2 đường tín hiệu, chẳng hạn có thể dùng loại 2 dây xoắn với nhau làm đường truyền. Tất nhiên là với cách giao tiếp này không thể có được tốc độ truyền cao.

3.2. Vòng dòng điện 60mA

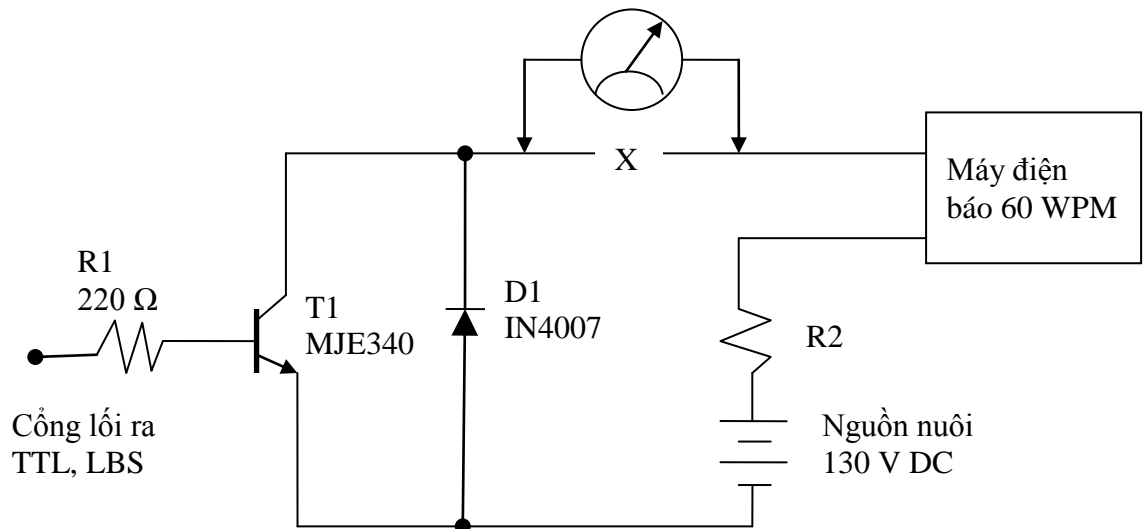
Hệ thống dòng 60 mA được mô tả theo hình vẽ dưới cho thấy một trong các máy điện báo in chữ có thể được nối với cổng lỗi ra nối tiếp tương thích TTL như thế nào. Trong nhiều trường hợp, một bit của một cổng song song sẽ được cấu hình như 1 cổng nối tiếp thông qua hoặc là phần mềm hoặc là bổ sung thêm cho phần cứng. Bit có thấp nhất (LSB) của cổng song song được chỉ định như đầu ra nối tiếp.

Mức TTL từ cổng ra nối tiếp điều khiển cực gốc của tranzito công suất NPN, chịu điện áp cao. Hai cực góp phát của tranzito được đấu nối tiếp với vòng dòng điện 60 mA và như vậy nó hoạt động như một chuyển mạch (công tắc). Khi mức TTL là HIGH, tranzito T1 mở và dòng điện chạy trong vòng. Ngược lại, khi bit TTL là LOW, tranzito chuyển sang trạng thái cấm và sẽ không có dòng điện đi qua mạch. Do đó, tranzito cung cấp một mức logic 1, khi bit TTL là HIGH, mức logic 0 khi là LOW.

Vòng dòng điện được nuôi từ nguồn một chiều 120 – 140 V, có một biến trở điều chỉnh được mắc nối tiếp, được sử dụng để đặt gần đúng mức dòng điện.

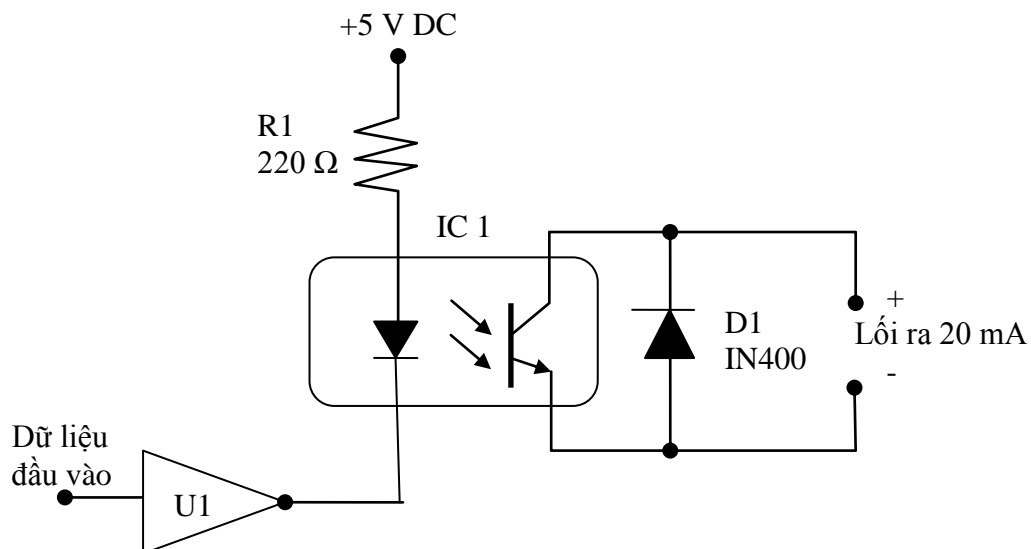
Mức dòng điện trong vòng điện 60mA được điều chỉnh bằng cách ngắt vòng và chèn vào mạch một mili ampe kế với thang đo 0 – 100 mA.

Một phím sẽ được nhấn trên bàn phím, một mức HIGH phải được viết vào cổng lỗi ra nối tiếp. Tác động này sẽ đóng mạch vòng và cho phép dòng đi qua. Chiết áp R2, được nối với biến trở, khi đó sẽ tiến hành điều chỉnh cho dòng điện đi qua xấp xỉ 60mA



Khi dòng 60 mA chạy qua các cuộn dây sẽ tạo ra một từ trường xung quanh từng cuộn. Khi bất ngờ làm gián đoạn dòng điện sẽ làm cho từ trường giảm đột ngột và xuất hiện 1 xung điện áp cao do hiện tượng cảm ứng điện từ. Kết quả là xung điện áp này có thể làm hỏng các linh kiện bán dẫn và ta phải nghĩ đến khả năng đưa ra các biện pháp bảo vệ. Chức năng này được điốt D1 đảm nhiệm, điốt này là loại chỉnh lưu có điện áp ngược tính theo đỉnh bằng 1000V (PIV). Bình thường điốt D1 được thiên áp ngược, chỉ trừ khi xung điện áp xuất hiện này mới bị phân cực thuận và dập tắt ngay xung điện áp xuất hiện trước khi nó kịp gây hại.

3.3. Vòng dòng điện 20mA

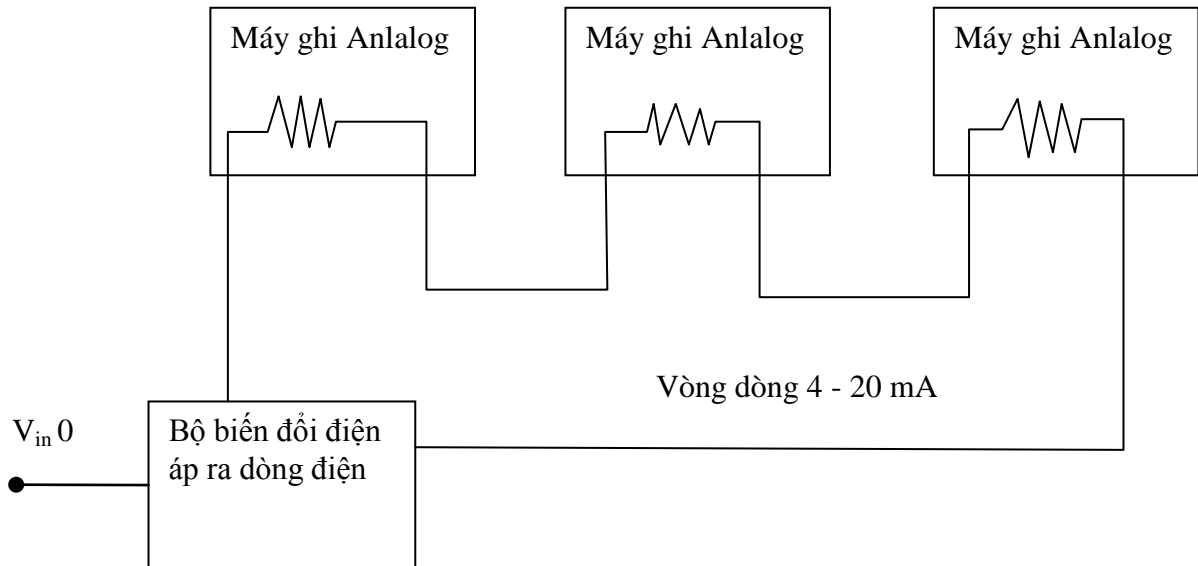


Chuẩn vòng dòng điện mới hơn đã sử dụng dòng điện với cường độ 20 mA cho mức logic 1 và dòng điện 0 cho mức logic 0. Vòng dòng điện 20mA đã được sử dụng trên máy điện báo in chữ Model 33 cả tất cả các kiểu máy kế tiếp. Các máy sử dụng dòng 20mA đều sử dụng mã ASCII.

Hình trên chỉ ra 1 phương pháp đơn giản cho phép sử dụng bộ ghép nối quang để ghép nối cổng truyền nối tiếp của máy tính với một vòng dòng điện 20 mA. Khi lối vào dữ liệu là HIGH, có nghĩa 1 trạng thái đánh dấu mức logic 1, thì lối ra của bộ đảo cực góp hờ (U1) sẽ

chuyển sang mức LOW, do đó cực âm của điốt phát quang sẽ nối đất. Tác động này sẽ làm mở tranzito, và cho phép dòng điện đi qua mạch. Lại một lần nữa, một điốt hoạt động trong chế độ phân cực ngược được sử dụng để ngăn ngừa hỏng hóc và các phiền phức khác gây ra bởi xung điện cảm ứng phát sinh khi các cuộn dây phóng điện.

3.4. Vòng dòng điện 4 đến 20mA



Trong kỹ thuật điều khiển các quá trình công nghiệp, người ta thường sử dụng hệ thống vòng dòng điện để truyền các số liệu đo lường. Hình trên thể hiện 1 hệ thống tiêu biểu, trong đó 3 thiết bị khác nhau thực hiện nhiệm vụ trong cùng 1 vòng dòng điện.

Trên một số bộ chỉ thị dòng điện dưới dạng số đang lưu hành trên thị trường, ta thấy dải đo không phải là 0-20 mA mà là 4-20 mA. Các bộ chỉ thị này cho phép từ 4 mA đến 20 mA. Mục đích của việc sử dụng vòng dòng điện này là truyền các giá trị nằm trong 1 vùng, mô tả các thông số đang được đo. Toàn bộ vùng có độ rộng là 10 mA, nhưng do việc chọn giá trị nhỏ nhất là 4 mA có thể tăng dòng điện cực đại đến 20 mA. Nguyên nhân của sự lệch, và cũng là một trong những ưu điểm của vòng dòng điện 4-20 mA so với các vòng dòng điện khác là trạng thái lỗi vào 0 được biểu diễn bởi 1 dòng điện khác 0, cụ thể là 4 mA. Do đó, trong vòng dòng điện 4 - 20 mA có thể dễ dàng phân biệt giữa 1 giá trị tham số 0 và trạng thái dòng điện bằng 0 do có sự cố trên mạch vòng (chập mạch hoặc đứt mạch) làm mất đi tín hiệu dòng điện.

CÂU HỎI VÀ BÀI TẬP

1. Trình bày vòng dòng điện 60mA
2. Trình bày vòng dòng điện 20mA
3. Trình bày vòng dòng điện 4 đến 60mA

Chương 4. CÁC MẠCH ĐIỀU KHIỂN VỚI BỘ BIẾN ĐỔI A/D

4.1. Card biến đổi A/D 12 bit dùng ICL7107

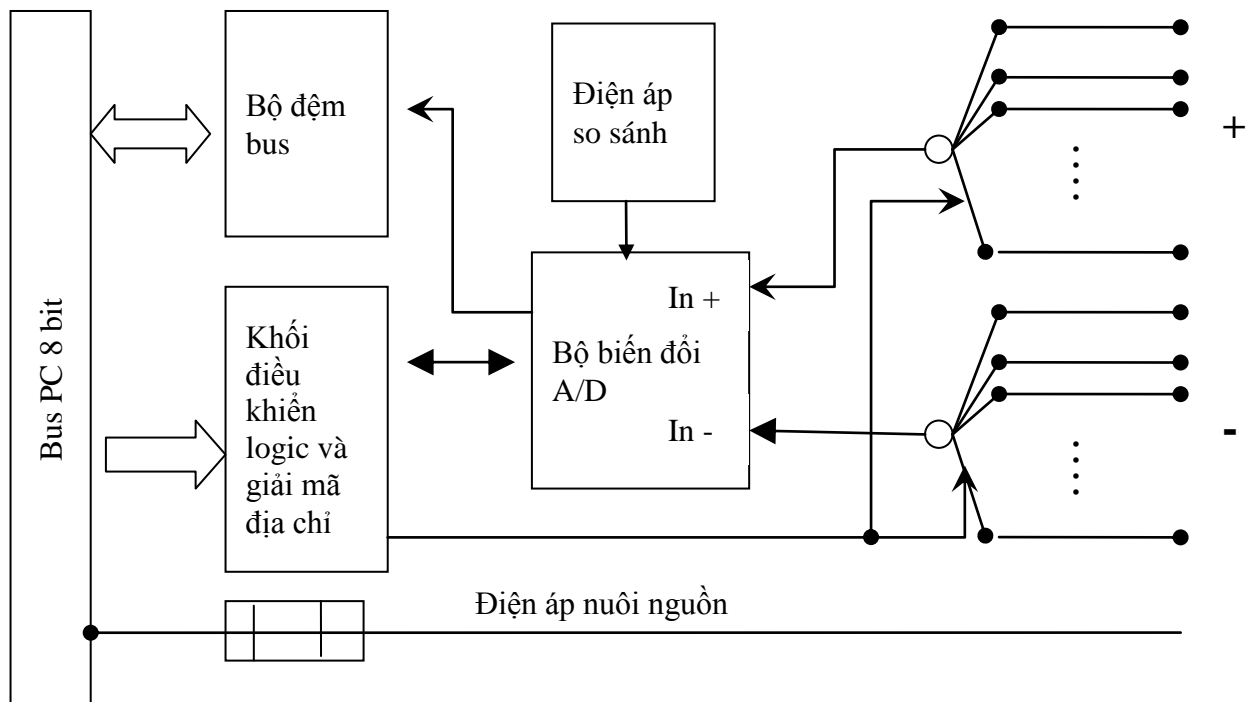
Nhiều người sử dụng máy tính quan tâm đến ứng dụng trong đo lường và điều khiển, nhưng để bắt tay vào công việc cần có thêm những phần cứng thích hợp, đóng vai trò ghép nối trung gian giữa máy tính và phần cứng bên ngoài. Một bộ phận không thể thiếu được là card biến đổi A/D và D/A mà lý do đơn giản là các tín hiệu mô tả lệnh, trạng thái hoặc dữ liệu trong máy tính đều tồn tại dưới dạng số.

Các thông số đặc trưng :

- Cho phép thực hiện 8 phép đo trong 1 giây khi hoạt động trong chế độ 1 kênh. Trong chế độ 8 kênh thì mỗi phép đo một giây.
- Cung cấp các giá trị đo trong vùng từ -1999 đến +1999.
- Độ phân giải tương ứng với 12 bit, nghĩa là 4096 bước hoặc ± 2048 bước.
- Có khả năng nhận dạng trạng thái vượt quá giới hạn đo.
- Có lối vào vi phân, cụ thể là có thể đo điện áp giữa 2 lối vào analog.

Card được mở rộng sao cho sau khi kết thúc 1 quá trình biến đổi A/D thì một ngắt được xóa và chương trình chạy trên máy tính có thể xử lý ngay giá trị đo, ngoài ra việc thu thập kết quả đo lường có thể tiến hành ngay sau máy tính, nghĩa là trong khi máy tính thực hiện các công việc khác.

Sơ đồ khối :



4.2. Card biến đổi A/D 12 bit dùng ADC547

■ Ưu điểm của ADC 547 :

- Điện trở lối vào lớn
- Vùng điện áp lối vào rất rộng
- Khả năng định địa chỉ phong phú
- Tất cả các tác động đều được điều khiển bằng phần mềm
- Giá thành không cao

- Mạch điện có độ phức tạp vừa phải
- Các thông số đặc trưng :
 - Có khả năng ghép nối qua khe cắm PC
 - Bộ biến đổi A/D có 12 bit
 - Thời gian biến đổi 25 micro giây
 - Có 16 lối vào hoạt động ở chế độ dồn kênh
 - Điện áp lối vào 0 – 5 +V hoặc -2V - + 2V
 - Có điện áp so sánh 10v bên trong
 - Địa chỉ cơ bản của card và vùng điện áp lối vào được đặt bằng cầu nối (jumper)
 - Điều khiển các quá trình trên card bằng PPI phổ dụng 8255
 - Có 8 lối vào/ra với mức analog
 - Dòng điện tiêu thụ 200 mA lấy từ khe cắm mở rộng của máy tính

4.3. Card biến đổi ADA 9-16 bit

Card ADA 9-16bit có 8 lối vào và 8 lối ra cho phép card mở rộng có thể sử dụng trong nhiều mục đích đo lường và điều khiển với độ chính xác cao.

■ Các đặc điểm chính :

- Độ phân giải cao : 9- 16 bit
- Độ chính xác cao nhất là 0.05% khi tốc độ biến đổi là 4 phép đo mỗi giây
- Bằng khả năng lập trình, có thể ấn định độ phân giải bằng 9 bit, khi tốc độ biến đổi cỡ 500 phép đo mỗi giây. Trước khi đến bộ biến đổi A/D, tín hiệu qua bộ dồn kênh với 8 kênh, vì thế có tổng cộng đến 8 điện áp lối vào có thể xử lý được.

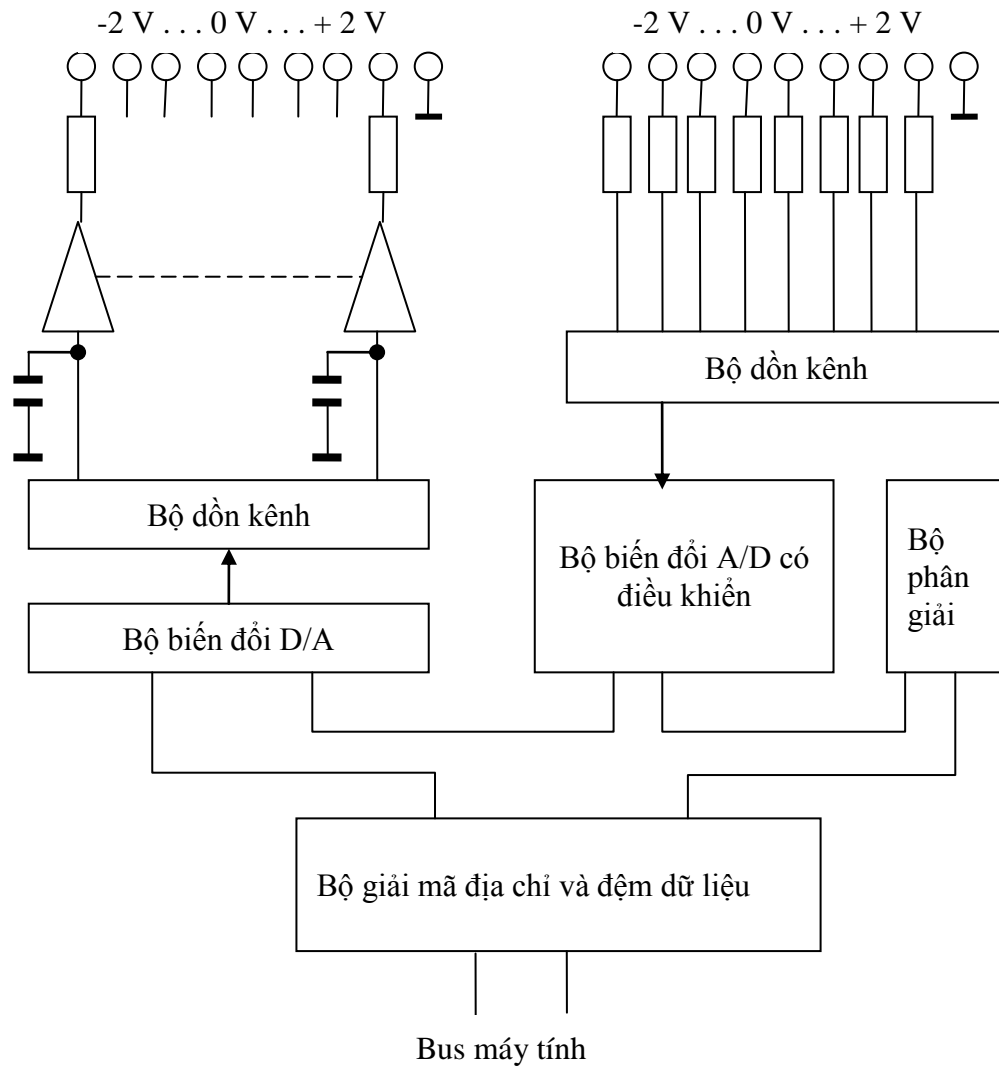
Khi xuất ra, bộ biến đổi ADA 16 có một biến đổi số / tương tự với độ phân giải 12 bit và độ chính xác là 0.025%, được mở rộng bằng một mạch lấy mẫu và giữ. Như vậy có tổng cộng 8 điện áp lối vào được xử lý cũng như 8 điện áp analog được xuất ra. Các dải điện áp đều nằm trong vùng -2v đến + 2v.

■ Sơ đồ khối :

Mạch điện của card biến đổi ADA 9-16 bit có 3 phần với 3 chức năng khác nhau :

- Bộ giải mã địa chỉ và đệm dữ liệu
- Bộ biến đổi tương tự/ số cùng với bộ dồn kênh đặt ở phía trước
- Bộ biến đổi số/ tương tự cùng với bộ dồn kênh/ đệm đặt ở phía sau

Bộ giải mã địa chỉ vào/ra giải mã 4 địa chỉ vào/ra kế tiếp nhau để điều khiển các linh kiện khác nhau của card ghép nối. Ngoài ra, bộ giải mã này còn điều khiển việc chuyển hướng dùng cho bộ đệm bus dữ liệu, nối các đường dẫn dữ liệu của máy tính với các đường dẫn dữ liệu của bộ biến đổi của A/D cũng như D/A



Phần thứ 2 là bộ biến đổi tương tự/số được bố trí ở sau bộ dồn kênh 8 lối vào, ngoài ra còn có mạch điện điều chỉnh độ phân giải cho phép ấn định độ chính xác của bộ biến đổi trong từng trường hợp cụ thể. Độ chính xác càng cao thì số phép đo được thực hiện trong một đơn vị thời gian càng nhỏ. Để có độ phân giải là 16 bit thì thời gian biến đổi cho mỗi giá trị đo cỡ 260 ms, tương ứng tốc độ xử lý là 4 phép đo mỗi giây. Khi chọn độ phân giải bằng 9 bit có tới cỡ 500 phép đo được thực hiện trong 1 giây. Tám kênh lối vào được xử lý kế tiếp nhau qua bộ dồn kênh và được đánh giá định lượng bằng bộ biến đổi A/D

CÂU HỎI VÀ BÀI TẬP

1. Trình bày Card biến đổi A/D 12 bit dùng ICL7107
2. Trình bày Card biến đổi A/D 12 bit dùng ADC547
3. Trình bày Card biến đổi ADA 9-16 bit

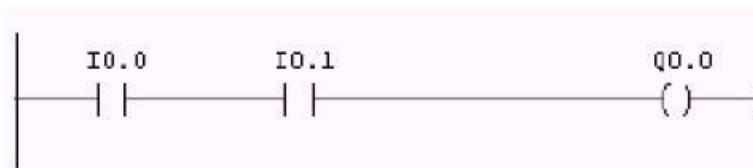
Chương 5. TỰ ĐỘNG HÓA VỚI PCL S7-200

5.1. Bộ điều khiển khả trình PCL

Hình thành từ nhóm các kỹ sư hãng General Motors năm 1968 với ý tưởng ban đầu là thiết kế một bộ điều khiển thỏa mãn các yêu cầu sau:

- Lập trình dễ dàng, ngôn ngữ lập trình dễ hiểu.
- Dễ dàng sửa chữa thay thế.
- Ổn định trong môi trường công nghiệp.
- Giá cả cạnh tranh.

Thiết bị điều khiển logic khả trình (PLC: Programmable Logic Control) là loại thiết bị cho phép thực hiện linh hoạt các thuật toán điều khiển số thông qua một ngôn ngữ lập trình, thay cho việc thể hiện thuật toán đó bằng mạch số.

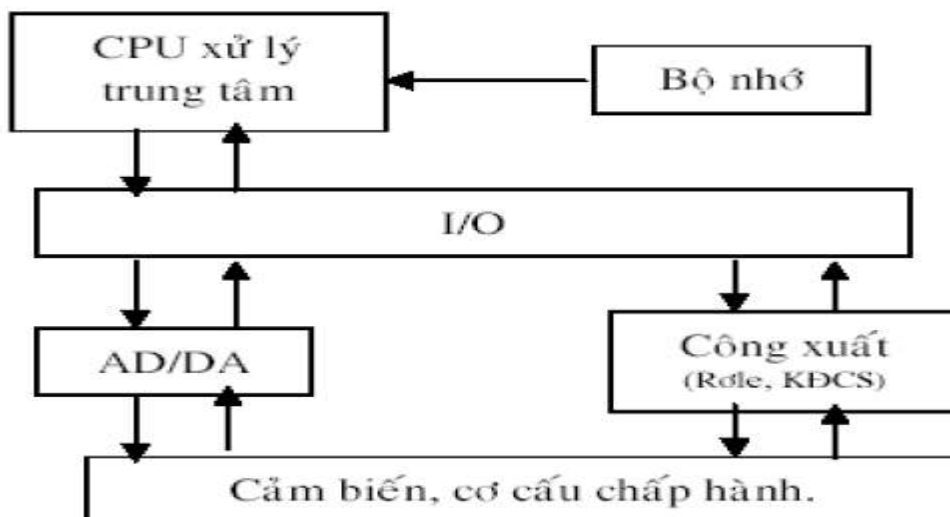


Δ	I	0.0
Δ	I	0.1
=	Q	0.0

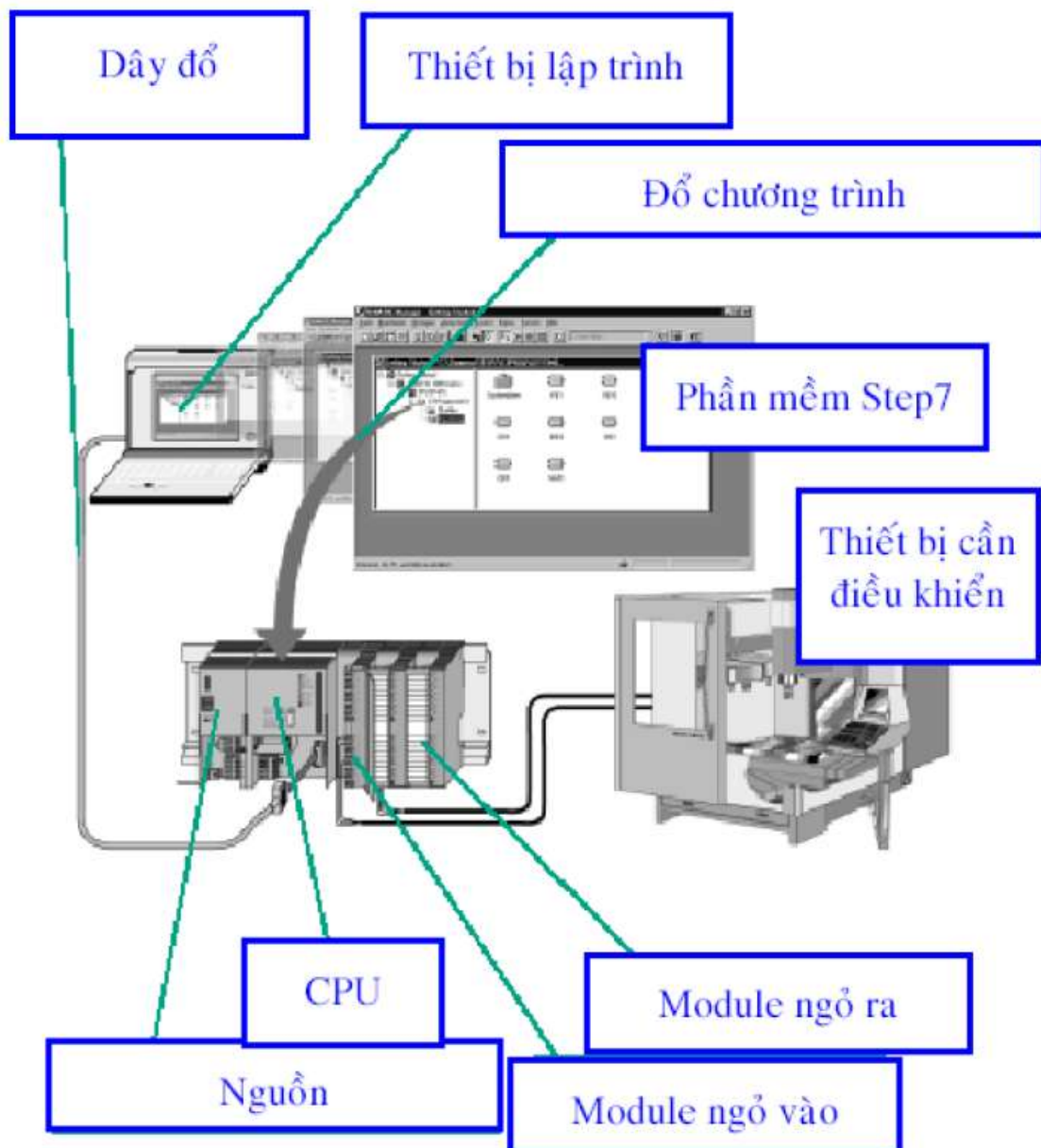
Tương đương một mạch số:



Như vậy, với chương trình điều khiển trong mình, PLC trở thành bộ điều khiển số nhỏ gọn, dễ thay đổi thuật toán và đặc biệt dễ trao đổi thông tin với môi trường xung quanh (với các PLC khác hoặc với máy tính). Toàn bộ chương trình điều khiển được lưu nhớ trong bộ nhớ PLC dưới dạng các khối chương trình (khối OB, FC hoặc FB) và thực hiện lặp theo chu kỳ của vòng quét.



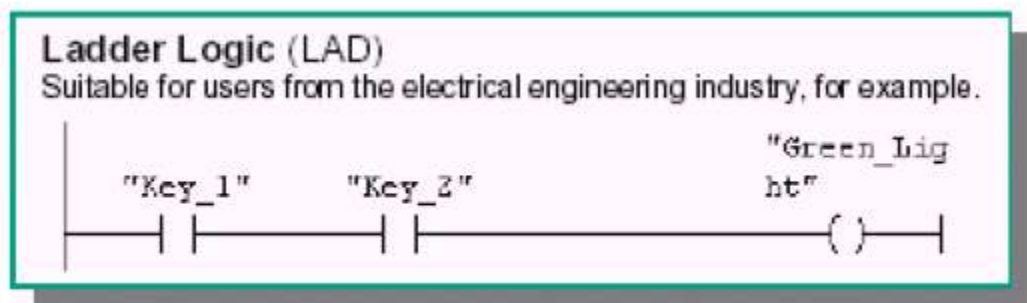
Để có thể thực hiện được một chương trình điều khiển, tất nhiên PLC phải có tính năng như một máy tính, nghĩa là phải có một bộ vi xử lý (CPU), một hệ điều hành, bộ nhớ để lưu chương trình điều khiển, dữ liệu và các cổng vào/ra để giao tiếp với đối tượng điều khiển và trao đổi thông tin với môi trường xung quanh. Bên cạnh đó, nhằm phục vụ bài toán điều khiển số, PLC còn cần phải có thêm các khối chức năng đặc biệt khác như bộ đếm (Counter), bộ định thì (Timer) ... và những khối hàm chuyên dụng.



5.2. Soạn thảo chương trình với PCL

Các loại PLC nói chung thường có nhiều ngôn ngữ lập trình nhằm phục vụ các đối tượng sử dụng khác nhau. PLC S7-300 có 5 ngôn ngữ lập trình cơ bản. Đó là:

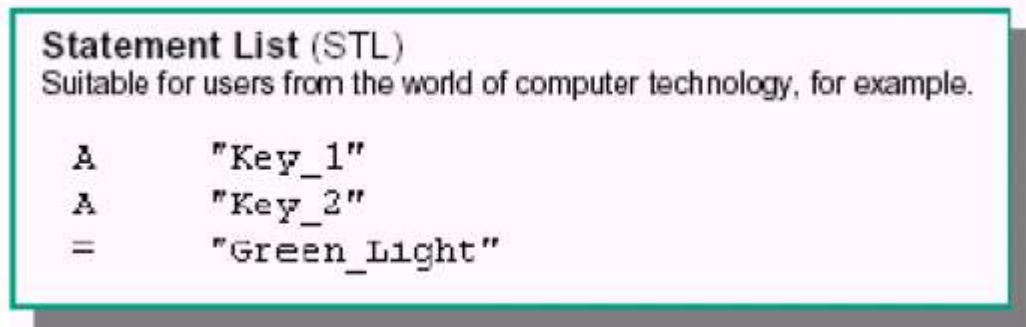
Ngôn ngữ “hình thang”, ký hiệu là LAD (Ladder logic).



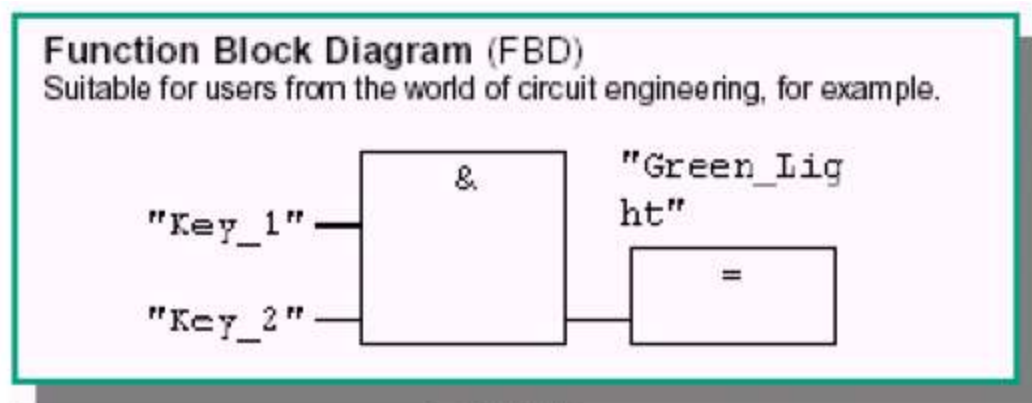
Đây là ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch logic.

Ngôn ngữ “liệt kê lệnh”, ký hiệu là STL (Statement list).

Đây là dạng ngôn ngữ lập trình thông thường của máy tính. Một chương trình được ghép gởi nhiều câu lệnh theo một thuật toán nhất định, mỗi lệnh chiếm một hàng và đều có cấu trúc chung là “tên lệnh” + “toán hạng”.



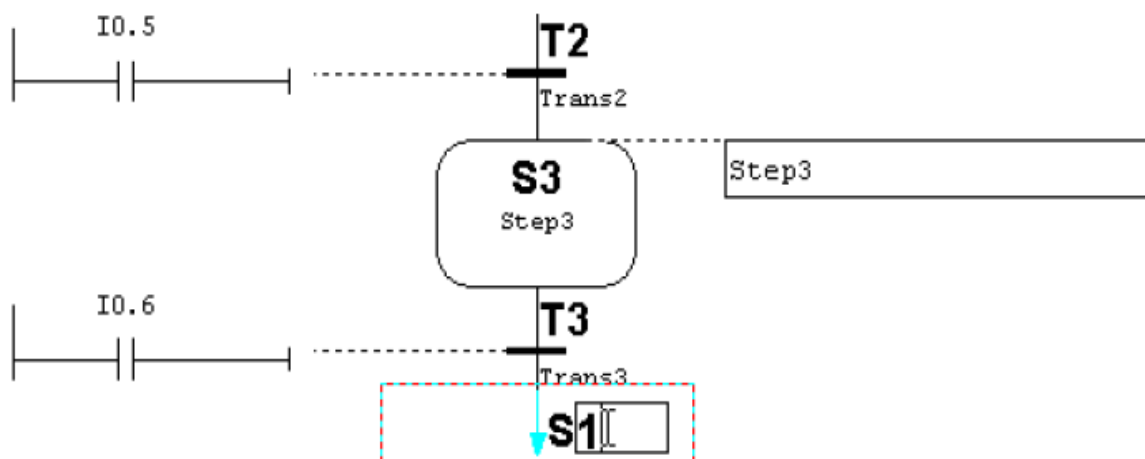
Ngôn ngữ “hình khối”, ký hiệu là FBD (Function Block Diagram).



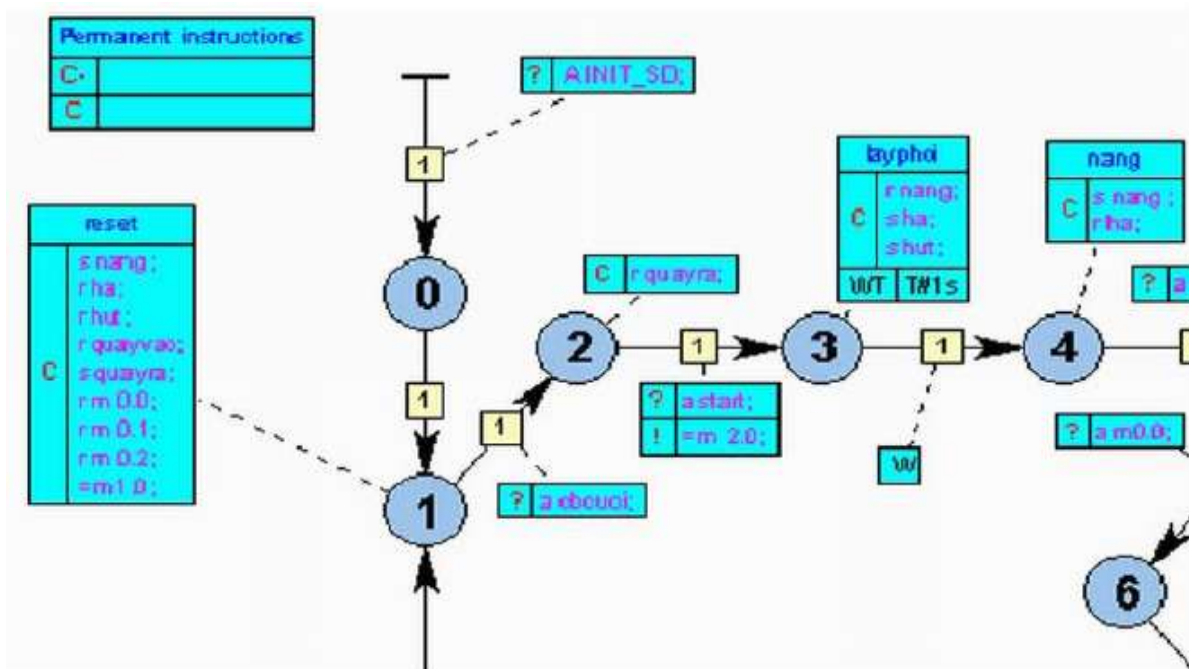
Đây cũng là ngôn ngữ đồ họa thích hợp với những người quen thiết kế mạch điều khiển số.

Ngôn ngữ GRAPH.

Đây là ngôn ngữ lập trình cấp cao dạng đồ họa. Cấu trúc chương trình rõ ràng, chương trình ngắn gọn. Thích hợp cho người trong ngành cơ khí vốn quen với giản đồ Grafset của khí nén.



Ngôn ngữ HIGH GRAPH.



CÂU HỎI VÀ BÀI TẬP

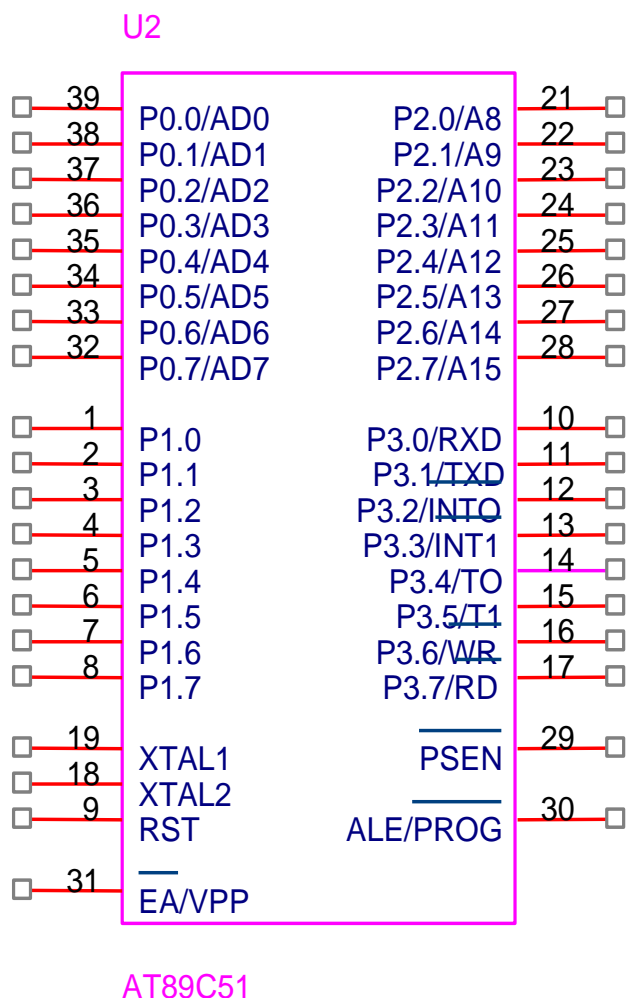
1. Trình bày Bộ điều khiển khả trình PCL
2. Thực hành lập trình PLC

Chương 6. HỘ VI ĐIỀU KHIỂN 8951

6.1. Cấu trúc phần cứng

8951 có tất cả 40 chân có chức năng như các đường xuất nhập. Trong đó có 24 chân có tác dụng kép (có nghĩa là 1 chân có 2 chức năng), mỗi đường có thể hoạt động như đường xuất nhập hoặc như đường điều khiển hoặc là thành phần của các bus dữ liệu và bus địa chỉ.

Các Port:



Port 0:

Port 0 là port có 2 chức năng ở các chân 32 - 39 của 8951. Trong các thiết kế cỡ nhỏ không dùng bộ nhớ mở rộng nó có chức năng như các đường I/O. Đối với các thiết kế cỡ lớn có bộ nhớ mở rộng, nó được kết hợp giữa bus địa chỉ và bus dữ liệu.

Port 1:

Port 1 là port I/O trên các chân 1-8. Các chân được ký hiệu P1.0, P1.1, p1.2, ... p1.7 có thể dùng cho giao tiếp với các thiết bị ngoài nếu cần. Port 1 không có chức năng khác, vì vậy chúng chỉ được dùng cho giao tiếp với các thiết bị bên ngoài.

Port 2:

Port 2 là 1 port có tác dụng kép trên các chân 21- 28 được dùng như các đường xuất nhập hoặc là byte cao của bus địa chỉ đối với các thiết bị dùng bộ nhớ mở rộng.

Port 3:

Port 3 là port có tác dụng kép trên các chân 10-17. Các chân của port này có nhiều chức năng, các công dụng chuyển đổi có liên hệ với các đặc tính đặc biệt của 8951 như ở bảng sau:

Bit	Tên	Chức năng chuyển đổi
P3.0	RXT	Ngõ vào dữ liệu nối tiếp.
P3.1	TXD	Ngõ xuất dữ liệu nối tiếp.
P3.2	INT0\	Ngõ vào ngắt cứng thứ 0
P3.3	INT1\	Ngõ vào ngắt cứng thứ 1
P3.4	T0	Ngõ vào củaTIMER/COUNTER thứ 0.
P3.5	T1	Ngõ vào củaTIMER/COUNTER thứ 1.
P3.6	WR\	Tín hiệu ghi dữ liệu lên bộ nhớ ngoài
P3.7	RD\	Tín hiệu đọc bộ nhớ dữ liệu ngoài.

Các ngõ tín hiệu điều khiển:

Ngõ tín hiệu PSEN (Program store enable):

PSEN là tín hiệu ngõ ra ở chân 29 có tác dụng cho phép đọc bộ nhớ chương trình mở rộng thường được nối đến chân OE\ (output enable) của Eprom cho phép đọc các byte mã lệnh.

PSEN ở mức thấp trong thời gian Microcontroller 8951 lấy lệnh. Các mã lệnh của chương trình được đọc từ Eprom qua bus dữ liệu và được chốt vào thanh ghi lệnh bên trong 8951 để giải mã lệnh. Khi 8951 thi hành chương trình trong EPROM nội PSEN sẽ ở mức logic 1.

Ngõ tín hiệu điều khiển ALE (Address Latch Enable)

Khi 8951 truy xuất bộ nhớ bên ngoài, port 0 có chức năng là bus địa chỉ và bus dữ liệu do đó phải tách các đường dữ liệu và địa chỉ. Tín hiệu ra ALE ở chân thứ 30 dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và dữ liệu khi kết nối chúng với IC chốt.

Tín hiệu ra ở chân ALE là một xung trong khoảng thời gian port 0 đóng vai trò là địa chỉ thấp nên chốt địa chỉ hoàn toàn tự động.

Các xung tín hiệu ALE có tốc độ bằng 1/6 lần tần số dao động trên chip và có thể được dùng làm tín hiệu clock cho các phần khác của hệ thống. Chân ALE được dùng làm ngõ vào xung lập trình cho EPROM trong 8951.

Ngõ tín hiệu EA\ (External Access):

Tín hiệu vào EA\ ở chân 31 thường được mắc lên mức 1 hoặc mức 0. Nếu ở mức 1, 8951 thi hành chương trình từ EPROM nội trong khoảng địa chỉ thấp 4 Kbyte. Nếu ở mức 0, 8951 sẽ thi hành chương trình từ bộ nhớ mở rộng. Chân EA\ được lấy làm chân cấp nguồn 12V khi lập trình cho Eprom trong 8951.

Ngõ tín hiệu RST (Reset) :

Ngõ vào RST ở chân 9 là ngõ vào Reset của 8951. Khi ngõ vào tín hiệu này đưa lên cao ít nhất là 2 chu kỳ máy, các thanh ghi bên trong được nạp những giá trị thích hợp để khởi động hệ thống. Khi cấp điện mạch tự động Reset.

Các ngõ vào bộ dao động X1, X2:

Bộ dao động được tích hợp bên trong 8951, khi sử dụng 8951 người thiết kế chỉ cần kết nối thêm thạch anh và các tụ như hình vẽ trong sơ đồ. Tần số thạch anh thường sử dụng cho 8951 là 12Mhz.

Chân 40 (Vcc) được nối lên nguồn 5V.

6.2. Tập lệnh

Tập lệnh họ MSC-51 được sự kiểm tra của các mode định vị và các lệnh của chúng có các Opcode 8 bit. Điều này cung cấp khả năng $2^8 = 256$ lệnh được thi hành và một lệnh không được định nghĩa. Vài lệnh có 1 hoặc 2 byte bởi dữ liệu hoặc địa chỉ thêm vào Opcode. Trong toàn bộ các lệnh có 139 lệnh 1 byte, 92 lệnh 2 byte và 24 lệnh 3 byte.

Các mode định vị (Addressing Mode) :

Các mode định vị là một bộ phận thống nhất của tập lệnh. Chúng cho phép định rõ nguồn hoặc nơi gởi tới của dữ liệu ở các đường khác nhau tùy thuộc vào trạng thái của người lập trình. 8951 có 8 mode định vị được dùng như sau:

- Thanh ghi.
- Trực tiếp.
- Gián tiếp.
- Tức thời.
- Tương đối.
- Tuyệt đối.
- Dài.
- Định vị.

Các kiểu lệnh (Instruction Types):

8951 chia ra 5 nhóm lệnh chính:

Các lệnh số học.

ADD A, <src, byte>

ADD A, Rn : (A) (A) + (Rn)

ADD A, direct : (A) (A) + (direct)

ADD A, @ Ri : (A) (A) + ((Ri))

ADD A, # data : (A) (A) + # data

ADDC A, Rn : (A) (A) + (C) + (Rn)

ADDC A, direct : (A) (A) + (C) + (direct)

ADDC A, @ Ri : (A) (A) + (C) + ((Ri))

ADDC A, # data : (A) (A) + (C) + # data

SUBB A, <src, byte>

SUBB A, Rn : (A) (A) - (C) - (Rn)

SUBB A, direct : (A) (A) - (C) - (direct)

SUBB A, @ Ri : (A) (A) - (C) - ((Ri))

SUBB A, # data : (A) (A) - (C) - # data

INC <byte>

INC A : (A) (A) + 1

INC direct : (direct) (direct) + 1

INC Ri : ((Ri)) ((Ri)) + 1

INC Rn : (Rn) (Rn) + 1

INC DPTR : (DPTR) (DPTR) + 1

DEC <byte>

DEC A : (A) (A) - 1

DEC direct : (direct) (direct) - 1

DEC @Ri : ((Ri)) ((Ri)) - 1
DEC Rn : (Rn) (Rn) - 1
MULL AB : (A) LOW [(A) x (B)]; có ảnh hưởng cờ OV
: (B) HIGH [(A) x (B)]; cờ Carry được xóa.
DIV AB : (A) Integer Result of [(A)/(B)]; cờ OV
: (B) Remainder of [(A)/(B)]; cờ Carry xóa
DA A : Điều chỉnh thanh ghi A thành số BCD đúng trong phép cộng BCD (thường DA A đi kèm với ADD, ADDC)
Nếu [(A3-A0)>9] và [(AC)=1] (A3A0) (A3A0) + 6.
Nếu [(A7-A4)>9] và [(C)=1] (A7A4) (A7A4) + 6.

■ **Lệnh logic.**

Tất cả các lệnh logic sử dụng thanh ghi A như là một trong những toán hạng thực thi một chu kỳ máy, ngoài A ra mất 2 chu kỳ máy. Những hoạt động logic có thể được thực hiện trên bất kỳ byte nào trong vị trí nhớ dữ liệu nội mà không qua thanh ghi A.

Các hoạt động logic được tóm tắt như sau:

ANL <dest - byte> <src - byte>

ANL A, Rn : (A) (A) AND (Rn).
ANL A, direct : (A) (A) AND (direct).
ANL A, @ Ri: (A) (A) AND ((Ri)).
ANL A, # data : (A) (A) AND (# data).
ANL direct, A : (direct) (direct) AND (A).
ANL direct, # data : (direct) (direct) AND # data.

ORL <dest - byte> <src - byte>

ORL A, Rn : (A) (A) OR (Rn).
ORL A, direct : (A) (A) OR (direct).
ORL A, @ Ri: (A) (A) OR ((Ri)).
ORL A, # data : (A) (A) OR # data.
ORL direct, A : (direct) (direct) OR (A).
ORL direct, # data : (direct) (direct) OR # data.

XRL <dest - byte> <src - byte>

XRL A, Rn : (A) (A) (Rn).
XRL A, direct : (A) (A) (direct).
XRL A, @ Ri: (A) (A) ((Ri)).
XRL A, # data : (A) (A) # data.
XRL direct, A : (direct) (direct) (A).
XRL direct, # data : (direct) (direct) # data.

CLR A : (A) 0

CLR C : (C) 0

CLR Bit : (Bit) 0

RL A : Quay vòng thanh ghi A qua trái 1 bit
(A_{n+1}) (A_n); n = 06
(A0) (A7)

RLC A : Quay vòng thanh ghi A qua trái 1 bit có cờ Carry

		$(An + 1) (An); n = 06$
		$(C) (A7)$
		$(A0) (C)$
RR	A	: Quay vòng thanh ghi A qua phải 1 bit
		$(An + 1) (An); n = 06$
		$(A0) (A7)$
RRC	A	: Quay vòng thanh ghi A qua phải 1 bit có cờ Carry
		$(An + 1) (An); n = 06$
		$(C) (A7)$
		$(A0) (C)$
SWAP	A	: Đổi chỗ 4 bit thấp và 4 bit cao của A cho nhau $(A3A0)(A7A4)$.

■ Dịch chuyển dữ liệu.

Các lệnh dịch chuyển dữ liệu trong những vùng nhớ nội thực thi 1 hoặc 2 chu kỳ máy. Mẫu lệnh MOV <destination>, <source> cho phép di chuyển dữ liệu bất kỳ 2 vùng nhớ nào của RAM nội hoặc các vùng nhớ của các thanh ghi chức năng đặc biệt mà không thông qua thanh ghi A.

Vùng Ngăn xếp của 8951 chỉ chứa 128 byte RAM nội, nếu con trỏ Ngăn xếp SP được tăng quá địa chỉ 7FH thì các byte được PUSH vào sẽ mất đi và các byte POP ra thì không biết rõ.

Các lệnh dịch chuyển bộ nhớ nội và bộ nhớ ngoại dùng sự định vị gián tiếp. Địa chỉ gián tiếp có thể dùng địa chỉ 1 byte (@ Ri) hoặc địa chỉ 2 byte (@ DPTR). Tất cả các lệnh dịch chuyển hoạt động trên toàn bộ nhớ ngoài thực thi trong 2 chu kỳ máy và dùng thanh ghi A làm toán hạng DESTINATION.

Việc đọc và ghi RAM ngoài (RD và WR) chỉ tích cực trong suốt quá trình thực thi của lệnh MOVX, còn bình thường RD và WR không tích cực (mức 1).

Tất cả các lệnh dịch chuyển đều không ảnh hưởng đến cờ. Hoạt động của từng lệnh được tóm tắt như sau:

MOV A,Rn	: (A) (Rn)
MOV A, direct	: (A) (direct)
MOV A, @ Ri	: (A) ((Ri))
MOV A, # data	: (A) # data
MOV Rn, A	: (Rn) (A)
MOV Rn, direct	: (Rn) (direct)
MOV Rn, # data	: (Rn) # data
MOV direct, A	: (direct) (A)
MOV direct, Rn	: (direct) (Rn)
MOV direct, direct	: (direct) (direct)
MOV direct, @ Ri	: (direct) ((Ri))
MOV direct, # data	: (direct) data
MOV @ Ri, A	: ((Ri)) (A)
MOV @ Ri, direct	: ((Ri)) (direct)
MOV @ Ri, # data	: ((Ri)) # data
MOV DPTR, # data16	: (DPTR) # data16
MOV A, @ A + DPTR	: (A) (A) + (DPTR)

$MOV @ A + PC : (PC) (PC) + 1$
 $(A) (A) + (PC)$
 $MOVX A, @ Ri : (A) ((Ri))$
 $MOVX A, @ DPTR : (A) ((DPTR))$
 $MOVX @ Ri, A : ((Ri)) (A)$
 $MOVX @ DPTR, A : ((DPTR)) (A)$
 $PUSH direct : \text{Cất dữ liệu vào Ngăn xếp}$
 $(SP) (SP) + 1$
 $(SP) (Direct)$
 $POP direct : \text{Lấy từ Ngăn xếp ra direct}$
 $(direct) ((SP))$
 $(SP) (SP) - 1$
 $XCH A, Rn : \text{Đổi chỗ nội dung của A với Rn}$
 $(A) (Rn)$
 $XCH A, direct : (A) (direct)$
 $XCH A, @ Ri : (A) ((Ri))$
 $XCHD A, @ Ri : \text{Đổi chỗ 4 bit thấp của (A) với ((Ri))}$
 $(A3A0) ((Ri3Ri0))$

■ Lý luận.

8951 chứa một bộ xử lý luận lý đầy đủ cho các hoạt động bit đơn, đây là một điểm mạnh của họ vi điều khiển MSC-51 mà các họ vi điều khiển khác không có.

RAM nội chứa 128 bit đơn vị và các vùng nhớ các thanh ghi chức năng đặc biệt cấp lên đến 128 đơn vị khác. Tất cả các đường Port là bit định vị, mỗi đường có thể được xử lý như Port đơn vị riêng biệt. Cách truy xuất các bit này không chỉ các lệnh rẽ nhánh không, mà là một danh mục đầy đủ các lệnh MOVE, SET, CLEAR, COMPLEMENT, OR, AND.

Toàn bộ sự truy xuất của bit dùng sự định vị trực tiếp với những địa chỉ từ 00H - 7FH trong 128 vùng nhớ thấp và 80H - FFH ở các vùng thanh ghi chức năng đặc biệt.

Bit Carry C trong thanh ghi PSW\ của từ trạng thái chương trình và được dùng như một sự tích lũy đơn của bộ xử lý luận lý. Bit Carry cũng là bit định vị và có địa chỉ trực tiếp vì nó nằm trong PSW. Hai lệnh CLR C và CLR CY đều có cùng tác dụng là xóa bit cờ Carry nhưng lệnh này mất 1 byte còn lệnh sau mất 2 byte.

Hoạt động của các lệnh luận lý được tóm tắt như sau:

$CLR C : \text{Xóa cờ Carry xuống 0. Có ảnh hưởng cờ Carry.}$
 $CLR BIT : \text{Xóa bit xuống 0. Không ảnh hưởng cờ Carry}$
 $SET C : \text{Set cờ Carry lên 1. Có ảnh hưởng cờ Carry.}$
 $SET BIT : \text{Set bit lên 1. Không ảnh hưởng cờ Carry.}$
 $CPL C : \text{Đảo bit cờ Carry. Có ảnh hưởng cờ Carry.}$
 $CPL BIT : \text{Đảo bit. Không ảnh hưởng cờ Carry.}$
 $ANL C, BIT : (C) (C) AND (BIT) : \text{Có ảnh hưởng cờ Carry.}$
 $ANL C, /BIT : (C) (C) AND NOT (BIT) : \text{Không ảnh hưởng cờ Carry.}$
 $ORL C, BIT : (C) (C) OR (BIT) : \text{Tác động cờ Carry.}$
 $ORL C, /BIT : (C) (C) OR NOT (BIT) : \text{Tác động cờ Carry.}$

MOV C, BIT : (C) (BIT) : Cờ Carry bị tác động.
MOV BIT, C : (BIT) (C) : Không ảnh hưởng cờ Carry.

■ **Rẽ nhánh chương trình.**

Có nhiều lệnh để điều khiển lên chương trình bao gồm việc gọi hoặc trả lại từ chương trình con hoặc chia nhánh có điều kiện hay không có điều kiện.

Tất cả các lệnh rẽ nhánh đều không ảnh hưởng đến cờ. Ta có thể định nhân cần nhảy tới mà không cần rõ địa chỉ, trình biên dịch sẽ đặt địa chỉ nơi cần nhảy tới vào đúng khẩu lệnh đã đưa ra.

Sau đây là sự tóm tắt từng hoạt động của lệnh nhảy.

JC rel : Nhảy đến "rel" nếu cờ Carry C = 1.
JNC rel : Nhảy đến "rel" nếu cờ Carry C = 0.
JB bit, rel : Nhảy đến "rel" nếu (bit) = 1.
JNB bit, rel : Nhảy đến "rel" nếu (bit) = 0.
JBC bit, rel : Nhảy đến "rel" nếu bit = 1 và xóa bit.
ACALL addr11: Lệnh gọi tuyệt đối trong page 2K.

(PC) (PC) + 2
(SP) (SP) + 1
((SP)) (PC7PC0)
(SP) (SP) + 1
((SP)) (PC15PC8)
(PC10PC0) page Address.

LCALLaddr16: Lệnh gọi dài chương trình con trong 64K.

(PC) (PC) + 3
(SP) (SP) + 1
((SP)) (PC7PC0)
(SP) (SP) + 1
((SP)) (PC15PC8)
(PC) Addr15Addr0.

RET : Kết thúc chương trình con trở về chương trình chính.

(PC15PC8) (SP)
(SP) (SP) - 1
(PC7PC0) ((SP))
(SP) (SP) -1.

RETI : Kết thúc thủ tục phục vụ ngắt quay về chương trình chính hoạt động tương tự như RET.

AJMP Addr11 : Nhảy tuyệt đối không điều kiện trong 2K.

(PC) (PC) + 2
(PC10PC0) page Address.

LJMP Addr16 : Nhảy dài không điều kiện trong 64K

Hoạt động tương tự lệnh LCALL.

SJMP rel :Nhảy ngắn không điều kiện trong (-128|27) byte

(PC) (PC) + 2
(PC) (PC) + byte 2

JMP @ A + DPTR: Nhảy không điều kiện đến địa chỉ (A) + (DPTR)
 $(PC) (A) + (DPTR)$

JZ rel : Nhảy đến A = 0. Thực hành lệnh kể nếu A = 0.
 $(PC) (PC) + 2$
 $(A) = 0 (PC) (PC) + \text{byte } 2$

JNZ rel : Nhảy đến A ≠ 0. Thực hành lệnh kể nếu A ≠ 0.
 $(PC) (PC) + 2$
 $(A) < > 0 (PC) (PC) + \text{byte } 2$

CJNE A, direct, rel : So sánh và nhảy đến A direct
 $(PC) (PC) + 3$
 $(A) < > (\text{direct}) (PC) (PC) + \text{Relative Address.}$
 $(A) < (\text{direct}) C = 1$
 $(A) > (\text{direct}) C = 0$
 $(A) = (\text{direct}).$ Thực hành lệnh kế tiếp

CJNE A, # data, rel : Tương tự lệnh CJNE A, direct, rel.
CJNE Rn, # data, rel : Tương tự lệnh CJNE A, direct, rel.
CJNE @ Ri, # data, rel : Tương tự lệnh CJNE A, direct, rel.
DJNE Rn, rel : Giảm Rn và nhảy nếu Rn = 0.
 $(PC) (PC) + 2$
 $(Rn) (Rn) - 1$
 $(Rn) < > 0 (PC) (PC) + \text{byte } 2.$

DJNZ direct, rel : Tương tự lệnh DJNZ Rn, rel.

CÂU HỎI VÀ BÀI TẬP

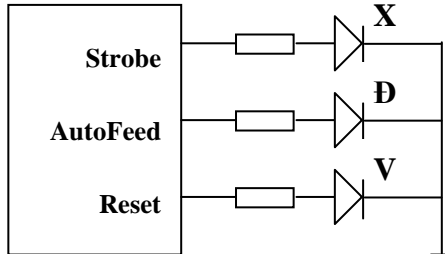
1. Trình bày Cấu trúc phần cứng vi điều khiển 8951
2. Thực hành lập trình cho vi điều khiển 8951

ĐỀ THI MẪU

Đề 1 :

Câu 1: Nêu cấu trúc cổng, ý nghĩa các chân cắm của cổng LPT.

Xác định khuôn dạng dữ liệu của cổng COM là bao nhiêu khi truyền đi 5 ký tự “EFGHI” với 1 bit start, 8bit a, 1 bit Parity, 2 bit stop?



Câu 2:

Lập trình cho các đèn LED nối với cổng LPT theo sơ đồ bên lần lượt thay nhau phát sáng theo chu kỳ 2 giây.

Câu 3 : Xây dựng thủ tục cho phép quét và cấp phát ID cho các thiết bị dạng được đầu nối giao diện kết nối USB của máy tính

Gợi ý đáp án

Câu 1 : Liệt kê danh sách các chân tín hiệu (tên, số hiệu chaanm chiều, ý nghĩa chân tín hiệu)

Chuyển đổi chuỗi ký tự sang mã ASCII 8bit , xác định khuôn dạng dữ liệu, vẽ sơ đồ tín hiệu.

Câu 2 : Xác định thanh ghi nào cần sử dụng cũng như các giá trị của thanh ghi đó theo các yêu cầu của đề bài, tiến hành viết đoạn chương trình đưa nạp giá trị vào các thanh ghi thông qua các thông số đã được xác định trước tương ứng với yêu cầu đề bài.

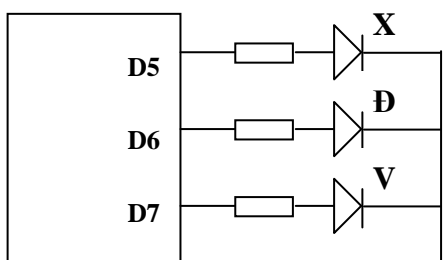
Câu 3 : Sử dụng các hàm và thủ tục API và HID để xây dựng chương trình.

Đề 2 :

Câu 1: Nêu cấu trúc đầu cắm, cáp nối của Bus ghép nối đa năng nối tiếp.

Xác định khuôn dạng dữ liệu của cổng COM là bao nhiêu khi truyền đi 5 ký tự “ABCDE” với 1 bit start, 8bit a, 1 bit Parity, 1 bit stop

Câu 2:



Lập trình cho các đèn LED nối với cổng LPT theo sơ đồ bên lần lượt thay nhau phát sáng theo chu kỳ 3 giây.

Câu 3: Xây dựng thủ tục cho phép truyền và nhận dữ liệu qua cổng USB với tham số cho trước là địa chỉ cổng

MÔN HỌC

Thiết bị ngoại vi & Kỹ thuật ghép nối (Peripherals & Interfacing Technique)

**By Bùi Quốc Anh,
Computer Engineering Dept.
(Tài liệu nội bộ - Confidence)**

<http://www.it-hut.edu.vn/~anhbq>

Email: anhbq@it-hut.edu.vn

Sách:

- **Microprocessor Interfacing techniques, R. Zaks & A. Lease, Sybex**
- **Micro Processor and Interfacing, D. Hall, McGraw Hill;**
- **IBM PC AT Technical Reference (Buses, IO Ports), IBM;**
- **Introduction to the PC Architecture, IBM PC Institute, 1997**
- **Interfacing to IBM PC – L. C. Eggebrecht, IBM Corp.**
- **Parallel Port Complete, J.Axelson, LakeViewResearch.**
- **Mastering Serial Communication, P.W. Gofton, Sybex.**
- **PC Intern (System Programming), M. Tischer, Abacus.**
- **Programming & Interfacing the 8051 MC, S. Yeralan, Addison-Wesley**
- **RTOS**
- **...**

Softwares:

- **TechHelp Ver. 4.0 / 6.0**
- **MSDN, Online Help.**
- ***Design tools: OrCAD, Protel, Cadence, Mentor Graphics...***
- **Programming Languages (C, Pascal, MASM, C++, VB, Delphi, VC++...)**
- **....**

Websites/pdf files:

- **IBM, Microsoft, Intel, Motorola ...**
- **ATMEL: atmel.com/product/microcontrollers 89Cxx (51/52/2051/8252, AVR - RISC, MSC51)...**
- **RENESAS: T-Engine embedded system**
- **National Semiconductor:
National.com/products/interface/ ADC-DAC:ADC 0804, 0809, DAC0800/1210, S&H LM198)...**
- **INTERSIL: intersil.com/products/ICL7109, 7135...**
- **Analog Devices Inc.: adi.com/products/adc, S&H...: AD574, AD1674**
- **USB: usb.org (pdf files for version 1.x & 2.x)**
- **Cypress – EZ USB, Developing Kit...**
- **...**

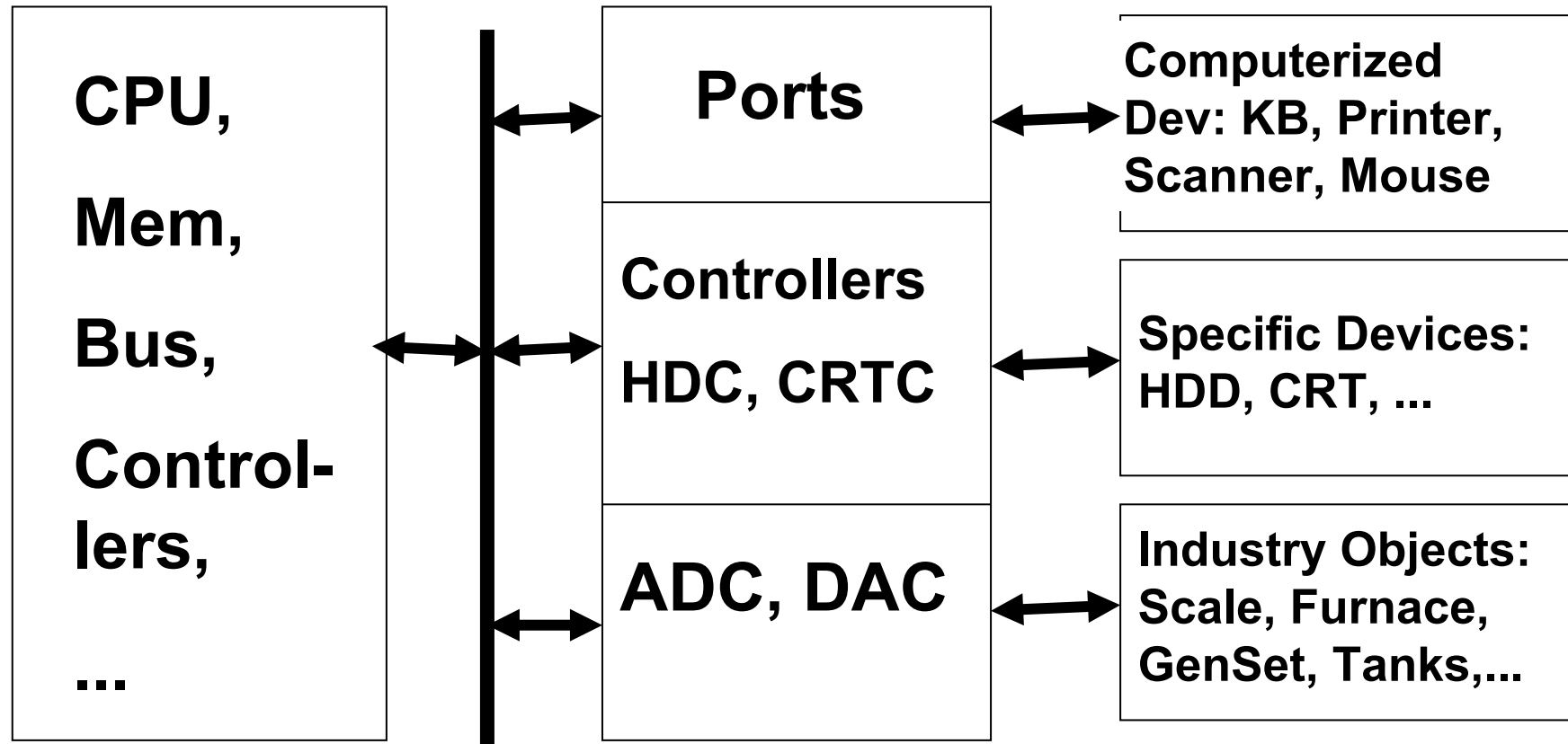
Interfacing?

1. HARDWARE

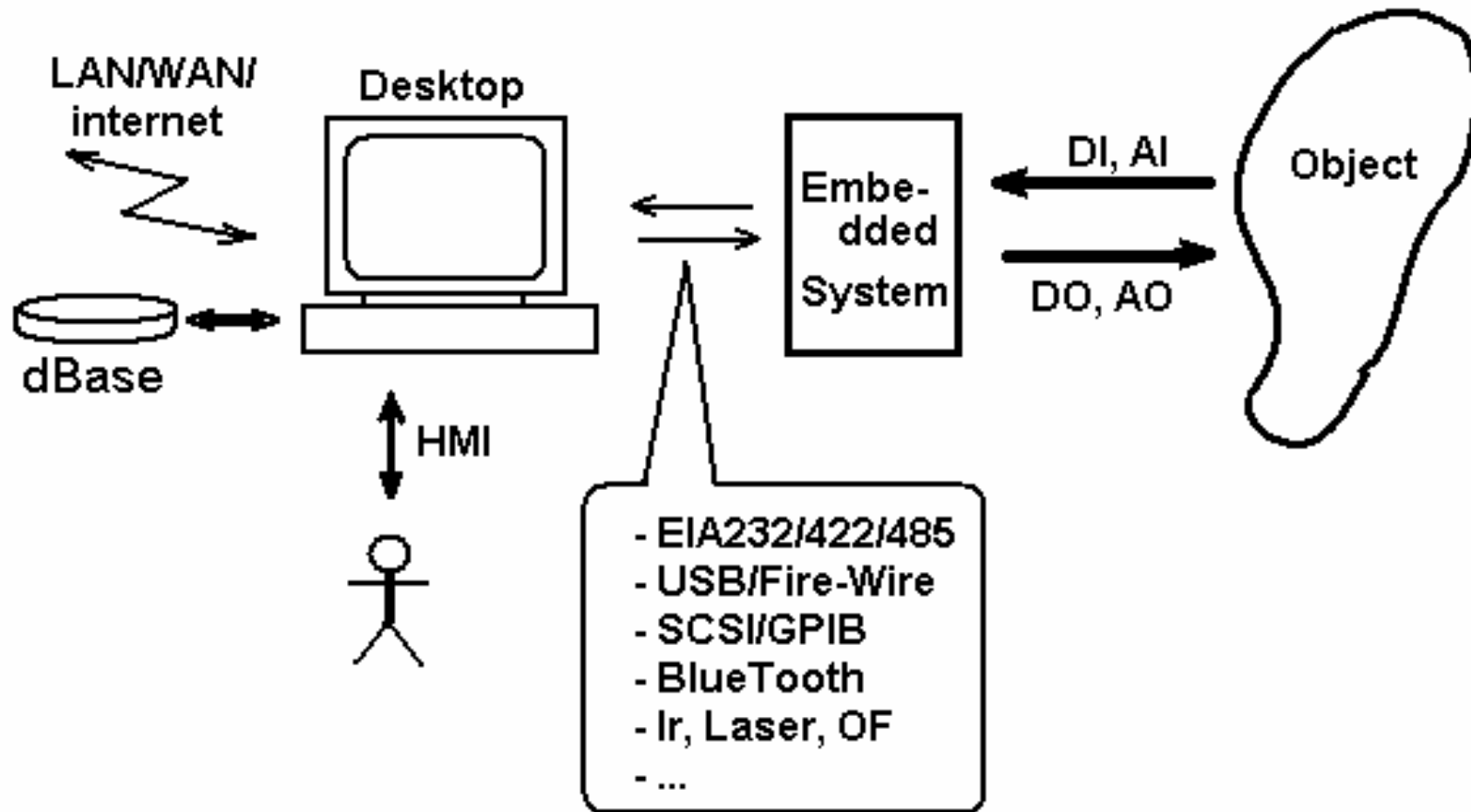
Central Sys.

Adaptor

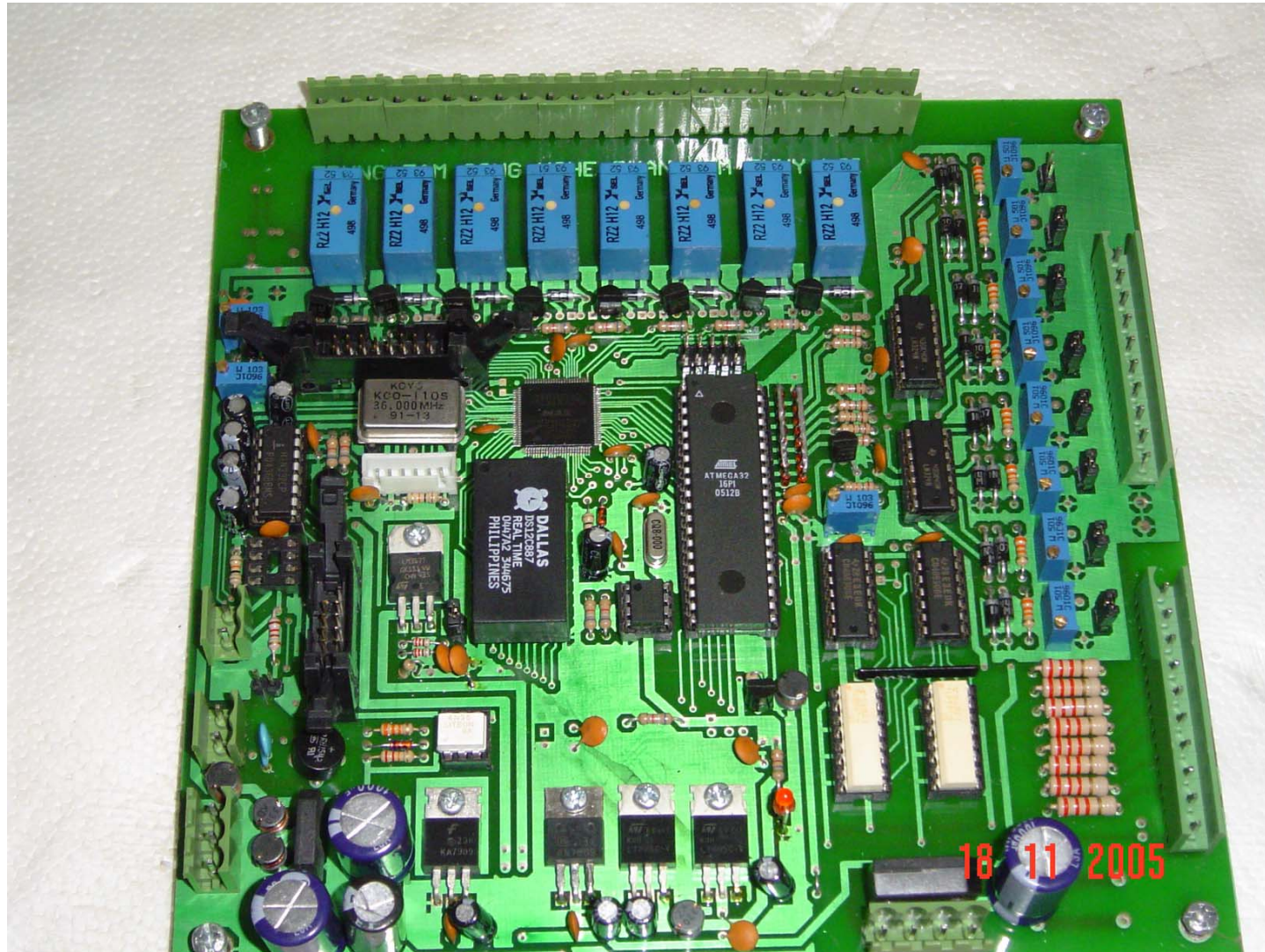
Wide world



2. SOFTWARE: Device Drivers: SLLs, DLLs, DRVs, VxDs, DCUs,⁵...



Hình 0.2. Sơ đồ ghép nối thiết bị ngoại vi hiện đại



H 0.3. Ví dụ về hệ nhúng Atmega 32, RISC

**H 5.4. RTU –
Sub system
for Tele Data
acquisition
and Control**



YÊU CẦU Đ/V HỌC VIÊN

Cần có kiến thức cơ bản các môn học:

- Lý thuyết mạch,
 - Điện tử,
 - Điện tử số,
 - Kiến trúc máy tính,
 - Vi xử lý,
 - Hệ điều hành,
 - data Base và
 - các ngôn ngữ lập trình thời gian thực.
- Kết thúc môn học, sinh viên phải hoàn thành bài tập dài, tham dự đủ các buổi lên lớp lý thuyết (theo qui chế), chữa bài tập, kiểm tra định kỳ và các buổi thí nghiệm.**

Phần A. KỸ THUẬT GHÉP NỐI (INTERFACING TECHNIQUE)

Ch. 1. Kiến trúc MT: Cấu trúc hệ vi xử lý/MT và hoạt động,

Ch. 2. Giao thức ghép nối

Ch. 3. Các phương pháp trao đổi thông tin:

Polling, Interrupt và DMA,

Ch. 4. Buses: Bus, Standardized Buses: ISA, PCI, Local, SCSI, USB, GPIB, I2C, IEEE 1394...

Ch. 5. Digital Interface: Cổng song song (PPI, LPT...), cổng nối tiếp (USART, RS-232C, RS-485...),

Ch. 6. Analog Interface: điện tử analog, DACs và ADCs

Ch. 7. Hệ thống thông tin công nghiệp và các hệ nhúng (Vi điều khiển): DAS, DCADA và DCS

Ch. 8. Chương trình điều khiển – Device Drivers:

PHẦN B. THIẾT BỊ NGOẠI VI (PERIPHERALS)

Ch. 9. Key-boards: Key-board, Key-pad

Ch. 10. Displays: LEDs, LCDs & CRT

**Ch. 11. Massive Storages: Controllers: FDD&FDC, HDD/ CD
& HDC, Tape, DiskChip,**

Ch. 12. Printers: Dot-Matrix (pin & thermal), Jet, Laser...

Ch. 13. Sound Devices: Sound card, ...

**Ch. 14. Auto ID: Scanner, Camera, Web cam, barcode reader,
RFID, Biometric...**

PHẦN C. BÀI TẬP LỚN, GỒM CÁC CHỦ ĐỀ SAU:

Tìm hiểu các thiết bị ngoại vi mới, các chuẩn ghép nối mới (buses, ports...),

- Xây dựng mạch ghép nối với hệ vi xử lý hoặc máy tính nhúng (Embedded System) hoặc PC,
- Xây dựng các chương trình điều khiển dưới môi trường Thời gian thực – RTOS (Linux – QNX, Windows-CE, VxWorks...),
- Auto ID (Voice recognition, smartcard, barcode, RFIDs, Biometric...)
- Đề tài có sản phẩm demo (không bắt buộc), làm tốt được cộng điểm:

- Tìm hiểu – thiết kế, lập trình ứng dụng các hệ nhúng (RENESAS, ARM, ATMEL, PIC, INTEL, MOTOROLA, ...)
- Ghép nối PC (comm port) \Leftrightarrow Atmel 8951 \Leftrightarrow 8 channel analog in and/or out, mô phỏng quá trình thực, viết chương trình : DD, vẽ đồ thị, CSDL.
- Ghép nối PC (comm port) \Leftrightarrow Atmel 8951 keypad+display (LED 7 seg, LCD...) và đếm xung, mô phỏng quá trình thực, viết chương trình : vi điều khiển và PC: vẽ đồ thị, CSDL.
- Điều khiển thiết bị qua mạng điện thoại, trả lời thông tin tự động

- Truyền số liệu qua modem: PC ⇔ modem ⇔ [telephone network] ⇔ modem ⇔ AT8951 ⇔ thu thập thông tin, điều khiển thiết bị,
- Thiết kế giao thức và truyền tin/điều khiển thiết bị hồng ngoại,
- Điều khiển thiết bị qua mạng điện thoại hoặc trả thông tin qua mạng điện thoại, dùng công nghệ DTMF/FSK/ PCM Codec...
- Ôtô/ Robot dò đường: step/ dc motor theo hành trình, đường đi,
- Robot Hand, tham gia ROBOTCON – có giải được miễn thi.
- Mạch Multi-IO ghép nối ISA bus: Digital In/Out, Multi channel Analog In, Analog Out, mô phỏng quá trình thực
- Ghép nối ICL 7135 với hệ Vi điều khiển/ISA bus.
- Ghép nối ICL 7109 (binary) với hệ Vi điều khiển/ISA bus.
- Ghép nối AD 574 (12 bit, 35 us) với hệ Vi điều khiển/ISA bus.
- Bảng thông tin điện tử - quang báo
- I2C bus, Blue Tooth: ghép nối các thiết bị nhúng
- ...

- Phần D. Thí nghiệm:
- Cổng nối tiếp: truyền tin và điều khiển thiết bị qua Comm port RS-232C,
- Cổng song song: điều khiển thiết bị qua LPT,
- DAC và phát hàm,
- ADC và hệ thu thập số liệu.
- USB

Thiết bị ngoại vi. Nội dung

- Thiết bị nhập dữ liệu: Key Board, Mouse
- Thiết bị xuất dữ liệu: Monitor, Printer
- Card mở rộng: Card âm thanh, card mạng
- Ổ quang: CD-ROM, DVD
- Các thiết bị khác.

Thiết bị nhập liệu.

⊙ Key Board: bàn phím



⊙ Mouse: con trỏ chuột



Thiết bị nhập liệu, keyboard

- ⊙ Keyboard ra đời từ rất sớm, trong mỗi hệ máy tính hiện nay đều có trang bị bàn phím tiêu chuẩn, những hệ máy tính đặc biệt thì có trang bị bàn phím chuyên dụng.
- ⊙ Keyboard có nhiều loại khác nhau:
 - Keyboard tiêu chuẩn
 - Keyboard cho máy xách tay
 - Keyboard ảo

Keyboard

⦿ Keyboard mini



⦿ Keyboard ảo



Con trỏ chuột

- ⊙ Con trỏ chuột ra đời muộn hơn Keyboard.
- ⊙ Sự ra đời của con trỏ chuột là một cột mốc trong ngành chế tạo máy tính
- ⊙ Giúp sự điều khiển sử dụng máy tính dễ dàng và tiện lợi hơn.
- ⊙ Từ khi ra đời đến nay con trỏ chuột đã có nhiều thay đổi trong công nghệ chế tạo.
- ⊙ Chuột Quang, Chuột Laser, Chuột bi lăn Dùng cho máy tính lớn.
- ⊙ Track ball, Track Pad, chuột cảm ứng, dành cho máy tính nhỏ.

Thiết bị nhập liệu, con chuột

⊙ Thế hệ chuột đầu tiên:

From Computer Desktop Encyclopedia
Reproduced with permission.
© 1996 The Bootstrap Institute



Thiết bị nhập liệu, con chuột

🎯 Chuột hiện nay



Thiết bị nhập liệu, con chuột

⊙ Track ball, Track Pad



Thiết bị xuất liệu.

⊙ Monitor: màn hình

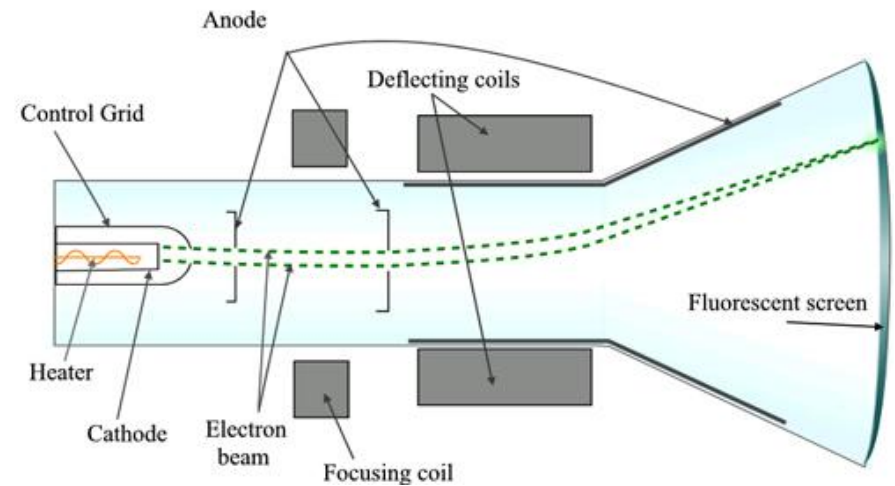
- CRT (Cathode Ray Tube)
- LCD (Liquid Crystal Display)
- Màn hình cảm ứng

⊙ Printer:

- Máy in Kim
- Máy in Phun
- Máy in Laser
- Máy in Offset

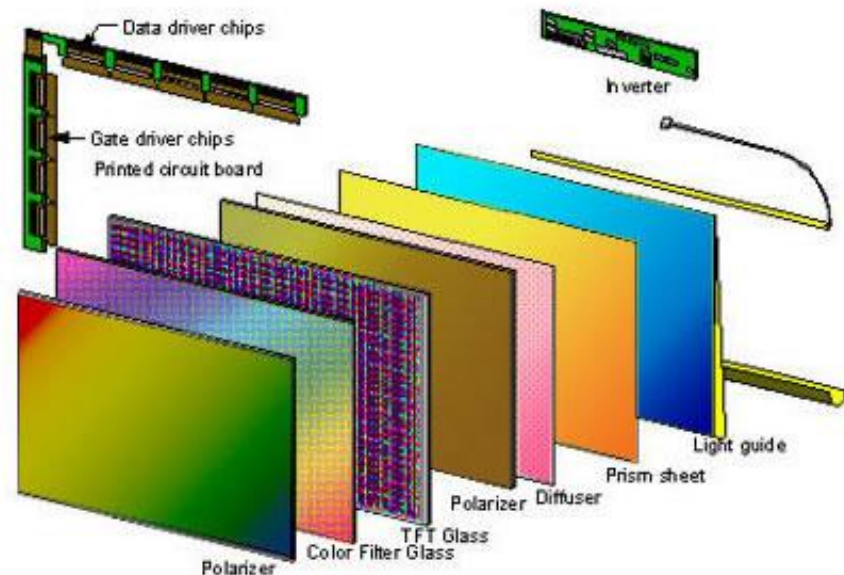
Monitor

- ⊙ Màn hình CRT: Màn hình dùng công nghệ ống cực Cathode lạnh, ra đời từ rất lâu, hiện công nghệ CRT đang dần được thay thế bằng LCD



Monitor

- ⊙ LCD: màn hình tinh thể lỏng, ra đời từ sớm tuy nhiên do hạn chế về tính năng và giá cả, nên LCD mới chỉ được dùng rộng rãi gần đây.



Monitor

- ⊙ Màn hình cảm ứng, là màn hình CRT hoặc LCD thông thường và được lắp đặt thêm tấm màn cảm ứng (cảm ứng điện trở hoặc điện dung)



Printer, máy in

- ⊙ Máy in được xem là thiết bị xuất dữ liệu cổ xưa, máy in ra đời trước khi màn hình ra đời.



Printer

- ⊙ Máy in kim: sử dụng ma trận kim và ruy băng mực (giống giấy than) để in ký tự.



Seamless ribbon

Printer

- ⊙ Máy in phun: dùng công nghệ phun mực trực tiếp lên giấy in. Thường dùng để in ảnh.



Printer

- ① Máy in Laser: dùng công nghệ định vị điểm ảnh bằng tia laser trên trống in để in ảnh, tốc độ cao.



Printer

- ⊙ Máy in Offset: Công nghệ cao dùng để in tốc độ nhanh, in chi tiết.



Ổ Quang

- ⊙ Dùng lưu trữ dữ liệu, Ổ quang là phương tiện lưu trữ dữ liệu hiệu quả, tiện dụng, ổ quang thế hệ đầu là CD-ROM có khả năng đọc dữ liệu trên các đĩa Compact dung lượng chứa tối đa 800 MB dữ liệu.
- ⊙ Thế hệ tiếp theo là công nghệ DVD với dung lượng tiêu chuẩn 4,8GB đĩa DVD 2 lớp có 9 GB dữ liệu, đĩa DVD 2 mặt 2 lớp có 18 GB dữ liệu.
- ⊙ Thế hệ mới nhất là HD-DVD và Bluray với tiêu chuẩn chứa 24 GB dữ liệu.

Ổ quang

◎ CD-ROM



Ổ quang

◎ DVD ROM



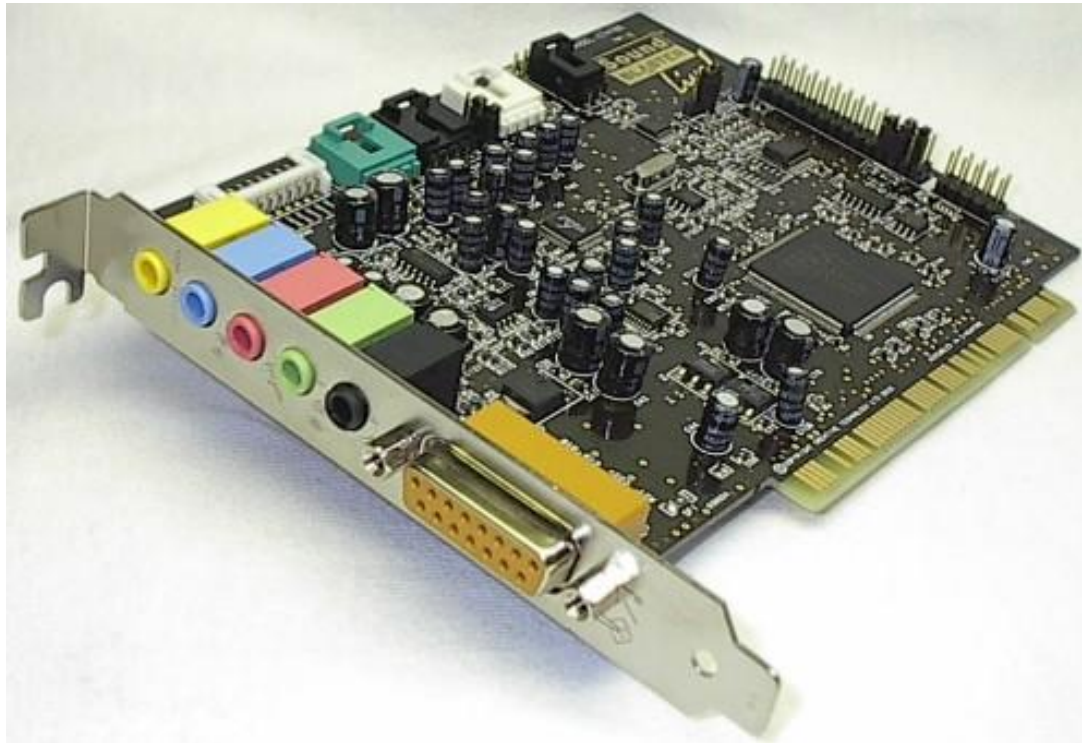
Ổ quang

◎ Bluray ROM



Thiết bị ngoại vi khác

⦿ Card âm thanh



Thiết bị ngoại vi khác

⦿ Card mạng Lan



Thiết bị ngoại vi khác

⦿ Modem Dial-up



Thiết bị ngoại vi khác

◎ Ổ đĩa mềm Floppy Disk



Thiết bị ngoại vi khác

◎ Ổ đĩa ZIP



Thiết bị ngoại vi khác

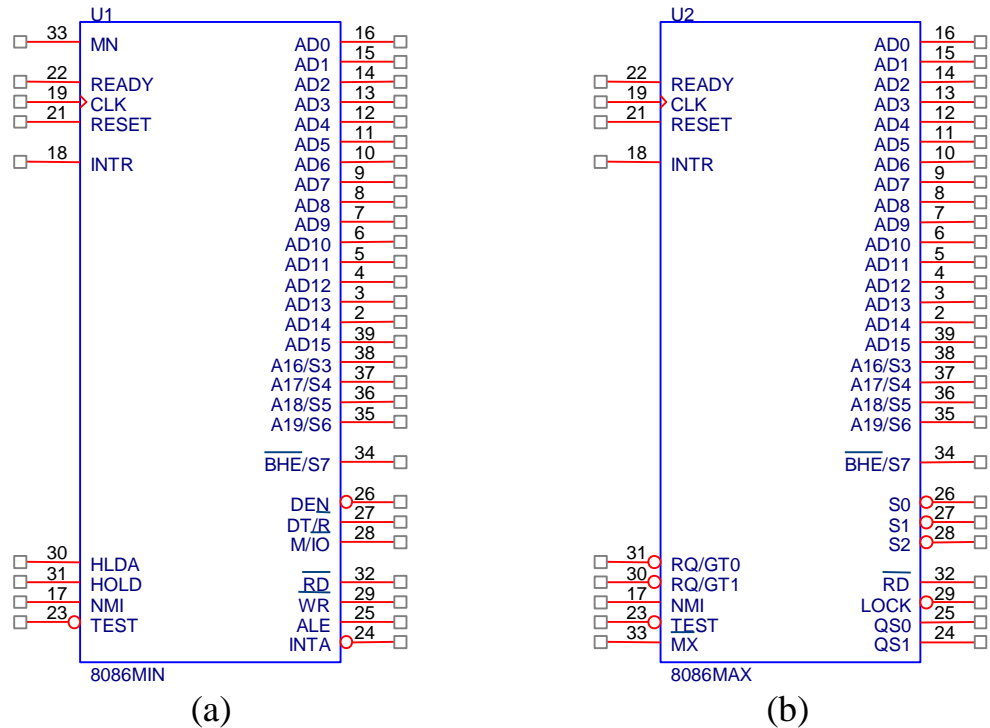
◎ Đầu đọc thẻ nhớ



CHƯƠNG V: GHÉP NỐI CPU VỚI BỘ NHỚ VÀ CÁC THIẾT BỊ NGOẠI VI

5.1. GIỚI THIỆU CÁC TÍN HIỆU CỦA 8088/8086 VÀ CÁC MẠCH PHỤ TRỢ.

CPU 8086 có các tín hiệu như biểu diễn trên hình 5.1 bao gồm:



Hình 5.1: Các tín hiệu của 8086 a) chế độ min; b) chế độ max

AD0 -AD7 (Address Data Bus): Là Bus đa hợp cho địa chỉ và dữ liệu. Trong các chu kỳ truy xuất bộ nhớ và vào ra, thì chu kỳ T1 của xung CLK các đường này sẽ đóng vai trò là các tín hiệu địa chỉ tác động cùng với tín hiệu ALE. Từ chu kỳ T2 trở đi chúng sẽ là các đường dữ liệu. Có thể thực hiện việc giải đa hợp cho bus này theo hình 5.2.

Tại T1 xung ALE sẽ cài các địa chỉ A0-A7 qua bộ cài cung cấp cho bộ nhớ và vào ra, trong các chu kỳ sau địa chỉ không còn trên các đường AD0-AD7, và lúc này chúng có thể sử dụng làm Bus dữ liệu.

A8-A15 (Address): là các đường địa chỉ cung cấp cho bộ nhớ và vào ra trong suốt các chu kỳ truy cập bộ nhớ và vào ra. Với 8086 các đường này cũng là bus đa hợp của địa chỉ và dữ liệu tương tự như các tín hiệu AD0 – AD7.

A16/S3 - A19/S6 (Address/Status): là các đường địa chỉ trong chu kỳ T1 của các chu kỳ truy xuất bộ nhớ, trong các chu kỳ truy xuất vào ra chúng luôn luôn ở mức thấp. Trong chu kỳ truy xuất bộ nhớ từ chu kỳ CLK thứ 2 chúng sẽ là các đường thông báo trạng thái đang làm việc của CPU. S6 luôn ở mức thấp, S5 thông báo trạng thái của cờ ngắt IF và kiểm tra lại ở mỗi cạnh lên của CLK, S4 và S3 được mã hóa chỉ thị các chu kỳ truy cập dữ liệu của CPU như bảng sau:

S4	S3	Chu kỳ hoạt động
0	0	Truy cập đoạn dữ liệu mở rộng.
0	1	Truy cập đoạn ngăn xếp.
1	0	Truy cập đoạn mã lệnh hoặc không.
1	1	Truy cập đoạn dữ liệu.

Các đường này cũng có thể dùng tín hiệu ALE để tách (demultiplex) riêng các đường địa chỉ và các đường trạng thái.

RD (Read): Là ngõ ra tác động mức thấp trong các chu kỳ T2 - T4, của các chu kỳ CPU đọc dữ liệu từ bộ nhớ hoặc vào ra.

READY: Là ngõ vào sử dụng để nhận biết dữ liệu sẵn sàng trên các bộ nhớ hoặc vào ra cần truy cập. Tín hiệu RDY từ bộ nhớ hoặc vào ra được đồng bộ trong bộ tạo clock 8284 tạo ra tín hiệu READY gửi tới CPU. Tín hiệu này sử dụng tạo ra các chu kỳ chờ cho việc truy cập các bộ nhớ có đáp ứng chậm.

INTR (Interrupt Request): Là ngõ vào tác động mức cao nhận tín hiệu yêu cầu ngắt quãng từ thiết bị ngoại vi. Tín hiệu này sẽ được xét tới tại cạnh xuống của chu kỳ CLK cuối cùng trong mỗi chu kỳ lệnh, nếu nó tích cực CPU sẽ kiểm tra cờ ngắt IF trong thanh ghi cờ, nếu IF đã được lập chương trình đang thực hiện sẽ tạm ngưng và chương trình ngắt được thực hiện. Nếu IF không được lập yêu cầu ngắt sẽ không được đáp ứng.

TEST: Là ngõ vào tác động mức thấp được kiểm tra bằng lệnh. Nếu tại thời điểm kiểm tra tín hiệu này ở mức thấp hoạt động của CPU sẽ tiếp tục bình thường, nếu không CPU sẽ thực hiện một chu kỳ chờ. Tín hiệu này được đồng bộ tại mỗi cạnh lên của CLK.

NMI (Non - Maskable Interrupt): Là ngõ vào nhận yêu cầu ngắt không che được bằng phần mềm. Cạnh lên ở ngõ vào này sẽ chuyển điều khiển của CPU qua chương trình phục vụ ngắt kiểu 2 tại cuối chu kỳ lệnh hiện hành.

RESET: Là ngõ vào tác động mức cao khởi động lại quá trình hoạt động của CPU, tín hiệu này phải tác động ít nhất trong 4 chu kỳ xung CLK.

CLK (Clock): Là ngõ vào cung cấp xung nhịp đồng bộ các hoạt động của CPU, Tín hiệu xung nhịp tốt nhất có 33% chu kỳ nhiệm vụ.

Vcc, GND: Là các ngõ vào cấp nguồn cho CPU, 8088 sử dụng nguồn 5Vdc 0%.

MN/MX (Minimum/Maximum): Là ngõ vào cho phép CPU8088 đổi chế độ hoạt động. Khi ngõ vào này mức cao CPU hoạt động trong chế độ cực tiểu, nó sẽ cung cấp các tín hiệu điều khiển trực tiếp tới bộ nhớ và vào ra. Khi ngõ vào này ở mức thấp CPU sẽ hoạt động trong chế độ cực đại, nó cung cấp 3 đường trạng thái S0, S1, S2 tới cho bộ điều khiển Bus 8288, bộ điều khiển Bus sẽ tạo ra các tín hiệu cần thiết để điều khiển bộ nhớ và vào ra. Sơ đồ hoạt động của CPU trong chế độ MIN như hình vẽ 5.3.

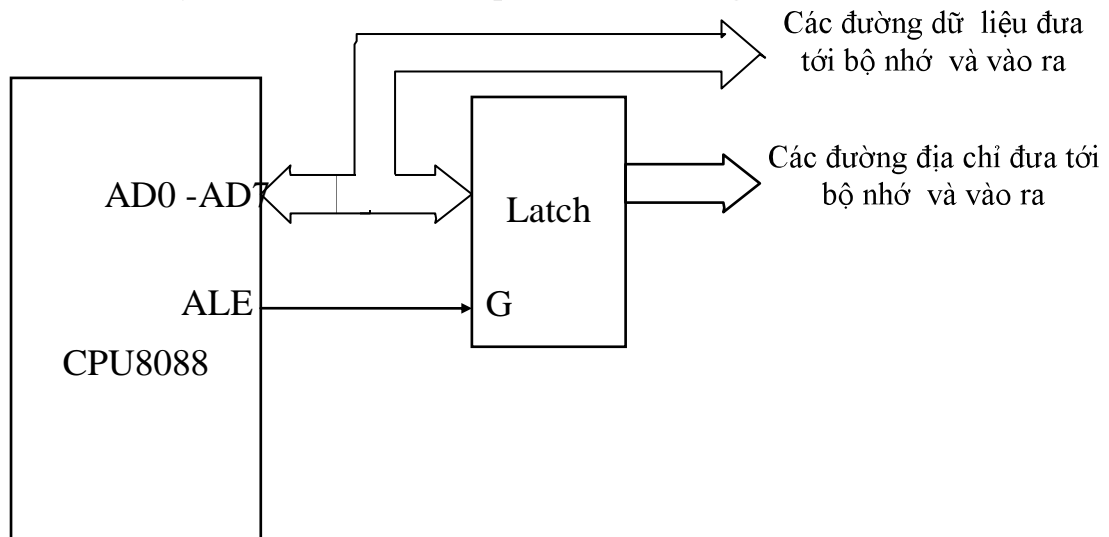
Các tín hiệu trong MIN mode bao gồm:

IO/M (Input Output/ Memory): Là ngõ ra phân biệt trạng thái CPU truy cập bộ nhớ hay vào ra. Trong chu kỳ truy xuất bộ nhớ nó tác động mức thấp, trong chu kỳ truy xuất vào ra nó tác động mức cao. Ngoài ra nó sẽ ở trạng thái trở kháng cao. Vì thế trong MIN mode tín hiệu này sử dụng cho phép các bộ giải mã địa chỉ bộ nhớ và vào ra.

WR (Write): Là tín hiệu ra tác động mức thấp thông báo trạng thái CPU ghi dữ liệu ra bên ngoài. Khi $IO/M = 1$ nó ghi tới vào ra, còn $IO/M = 0$ nó ghi tới bộ nhớ. Tín hiệu này tác động trong các chu kỳ T2-T4, nó ở mức 1 trong các chu kỳ khác và trở kháng cao khi BUS treo.

INTA (Interrupt Acknowledge): Là ngõ ra tác động mức thấp thông báo chu kỳ CPU chấp nhận yêu cầu ngắt quãng từ thiết bị bên ngoài. Tín hiệu này thường được sử dụng cho phép bộ điều khiển ngắt cung cấp vector ngắt cho CPU.

ALE (Address Latch Enable): Là ngõ ra tác động mức cao trong chu kỳ xung CLK đầu tiên trong mỗi chu kỳ CPU truy cập bộ nhớ và vào ra. Tín hiệu này được sử dụng cài các tín hiệu địa chỉ trên bus đa hợp ra ngoài BUS hệ thống cung cấp tới bộ nhớ và vào ra, từ chu kỳ T_2 trở đi BUS đa hợp sẽ được sử dụng cho nhiệm vụ khác.



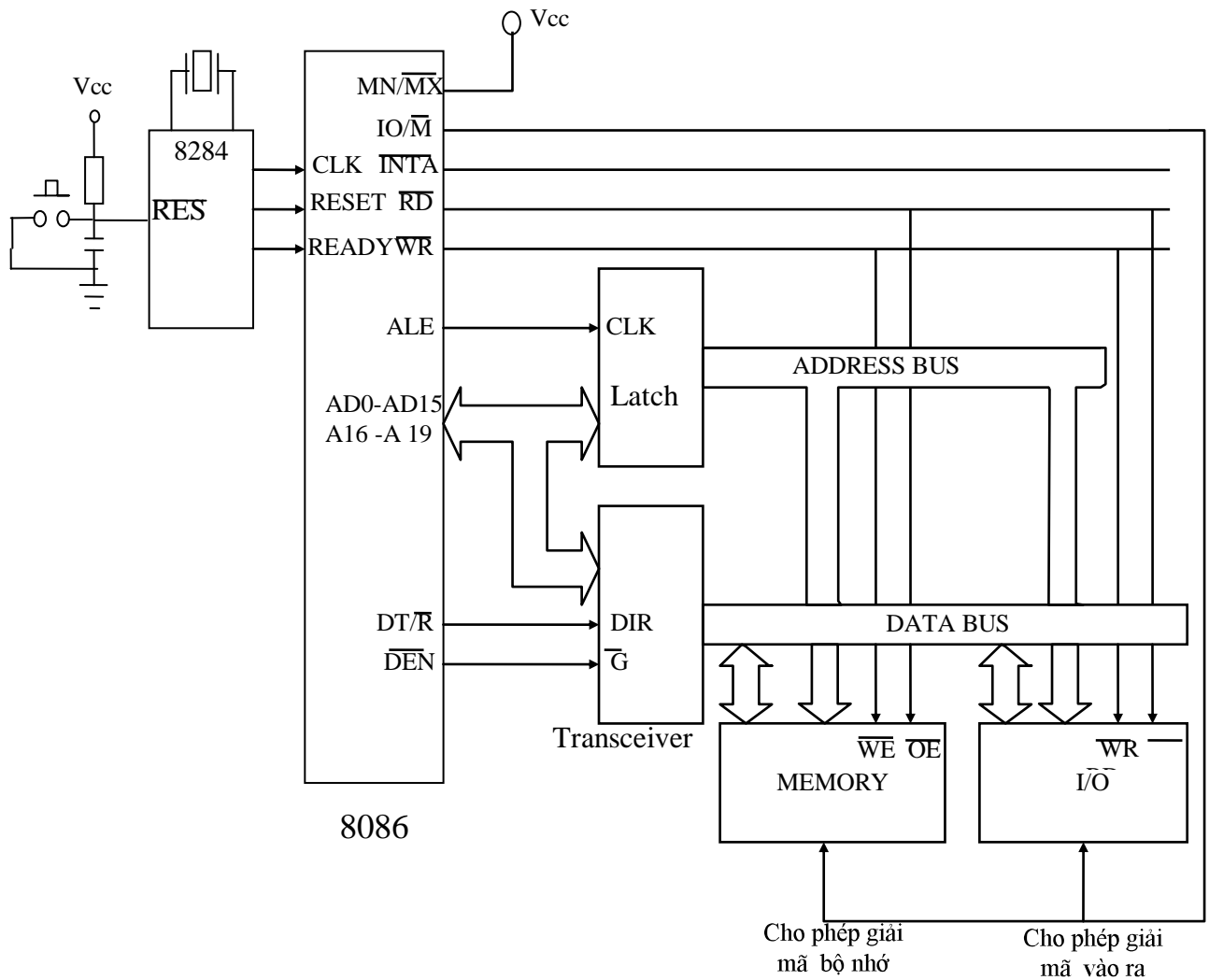
Hình 5.2: Sơ đồ tách Bus đa hợp.

DT/R (Data Transmit/Receive): Là tín hiệu ngõ ra sử dụng để xác định chiều truyền dữ liệu trên Data BUS. Khi $DT/R = 1$ CPU ghi dữ liệu ra bộ nhớ hoặc vào ra, còn khi $DT/R = 0$ nó đọc dữ liệu từ bên ngoài vào. Tín hiệu này sẽ ở trạng thái trở kháng cao khi CPU treo BUS.

DEN (Data Enable): Là ngõ ra thông báo trạng thái CPU truyền nhận dữ liệu trên bus đa hợp. Tín hiệu này thường sử dụng để cho phép bộ đệm truyền nhận của bus dữ liệu.

HOLD, HOLDA: Khi thiết bị nào đó có yêu cầu thâm nhập BUS của CPU nó cung cấp tới ngõ vào HOLD tín hiệu tác động mức cao, khi đó CPU sẽ treo các BUS của nó lên trở kháng cao và trả lời bằng cách đưa ngõ ra HOLDA lên mức cao. Khi thiết bị thực hiện xong chu kỳ chiếm bus nó sẽ kéo HOLD trở lại mức thấp, CPU sẽ tiếp tục làm việc và kéo HOLDA trở lại mức thấp. Các tín hiệu này thường được sử dụng để vi xử lý giao tiếp với bộ điều khiển DMA.

Sơ đồ các tín hiệu trong chế độ cực đại của CPU 8086/8088 trình bày trên hình 5.4.



Hình 5.3: Sơ đồ hoạt động trong Min Mode của 8086/8088.

SSO (Status line): Là ngõ ra tác động mức thấp kết hợp với các tín hiệu DT/R và IO/M để giải mã hoàn toàn trạng thái của các chu kỳ hiện hành như trình bày trên bảng sau:

IO/M	DT/R	SSO	Chu kỳ BUS
1	0	0	Công nhận ngắt
1	0	1	Đọc cổng vào ra
1	1	0	Ghi cổng vào ra
1	1	1	Treo
0	0	0	Đọc mã lệnh
0	0	1	Đọc bộ nhớ
0	1	0	Ghi bộ nhớ
0	1	1	Thụ động

Các tín hiệu trong MAX mode.

S2, S1, S0 (Status): Là các đường trạng thái mã hóa từ các chế độ hoạt động của CPU. Các tín hiệu này sử dụng cung cấp tới bộ điều khiển BUS 8288 để tạo ra các tín hiệu điều khiển bộ nhớ và vào ra. Các tín hiệu này bắt đầu tác động trong chu kỳ T4 để chỉ thị một chu kỳ BUS mới, chúng sẽ được giữ trong suốt các chu kỳ T1, T2 và trở về trạng thái thụ động trong chu kỳ T3. Chúng sẽ ở trạng thái trở kháng cao khi CPU treo BUS. Các tín hiệu sẽ được mã hóa như sau:

S2	S1	S0	Chu kỳ BUS
0	0	0	Công nhận ngắt
0	0	1	Đọc cổng vào ra
0	1	0	Ghi cổng vào ra
0	1	1	Treo BUS
1	0	0	Đọc mã lệnh
1	0	1	Đọc bộ nhớ
1	1	0	Ghi bộ nhớ
1	1	1	Trạng thái thụ động

RQ/GT0, RQ/GT1 (Request/Grant): Là các ngõ vào ra sử dụng cho các thiết bị có yêu cầu thâm nhập BUS của CPU. RQ/GT0 có mức ưu tiên cao hơn RQ/GT1. Quá trình yêu cầu và trả lời thực hiện trên các chân này thực hiện theo trình tự sau:

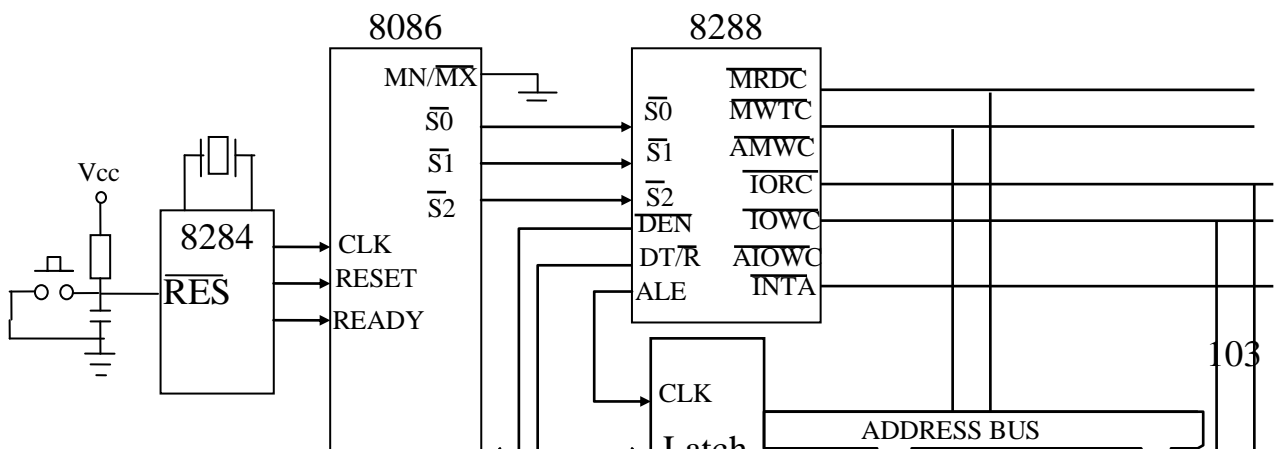
1. Thiết bị bên ngoài gửi yêu cầu bằng một xung có độ rộng bằng 1 CLK tới CPU.
2. Trong chu kỳ T4 hoặc T1 CPU sẽ trả lời thiết bị bằng cách phát ra một xung có độ rộng bằng 1 CLK, đồng thời với việc CPU treo các BUS của nó lên trở kháng cao.
3. Khi truy cập xong BUS thiết bị sẽ thông báo cho CPU bằng 1 xung tiếp theo. Lúc này CPU có thể tiếp tục làm việc trong chu kỳ xung CLK kế tiếp.

Mỗi chu kỳ yêu cầu sẽ bao gồm chuỗi 3 xung liên tiếp, và cần phải có một chu kỳ rỗi sau mỗi chu kỳ yêu cầu. Các xung yêu cầu tác động mức thấp.

Nếu yêu cầu xuất hiện khi CPU đang thực hiện một chu kỳ truy cập bộ nhớ, nó sẽ chỉ giải phóng BUS khi thỏa mãn các điều kiện sau:

1. Yêu cầu xuất hiện trước chu kỳ T2.
2. Chu kỳ hiện hành không phải là bit thấp nhất của một từ.
3. Chu kỳ hiện hành không phải là chu kỳ đầu tiên của chu kỳ công nhận ngắt quãng.
4. Không có tiền tố LOCK trong lệnh vừa thực hiện.

LOCK: Là ngõ ra tác động mức thấp khi có tiền tố LOCK trong lệnh của chương trình, nó sẽ được duy trì cho đến hết lệnh kế tiếp của chương trình. Khi tín hiệu này tác động sẽ không có thiết bị bên ngoài nào được thâm nhập BUS của CPU.



QS0, QS1 (Queue Status): Là các ngõ ra thông báo trạng thái hiện hành của hàng đợi lệnh. Chúng được mã hóa như sau:

QS1	QS0	Trạng thái hàng đợi
0	0	Không hoạt động
0	1	Có một byte mã lệnh đầu tiên trong hàng đợi
1	0	Hàng đợi rỗng
1	1	Có nhiều byte trong hàng đợi

5. Định thời hoạt động của 8086/8088.

Hình 5.5 trình bày sự tác động các tín hiệu của CPU 8086 trong các thao tác đọc ghi bộ nhớ. Hàng đầu tiên của giản đồ là dạng sóng của xung clock, trong hệ thống 8086 tín hiệu này thường được cung cấp từ bộ tạo clock 8284. Mỗi chu kỳ clock được gọi là một trạng thái (state), một trạng thái sẽ được tính từ cạnh xuống của clock này tới cạnh xuống của clock kế tiếp. Các CPU họ 8086/8088 có tốc độ đáp ứng từ 5 tới 10 Mhz nên mỗi trạng thái chiếm một khoảng thời gian từ 100 tới 200ns tùy thuộc vào tần số tinh thể thạch anh sử dụng.

Một hoạt động cơ bản của CPU như đọc một byte dữ liệu từ bộ nhớ, hoặc ghi một byte ra bộ nhớ được gọi là một chu kỳ máy (machine cycle). Trên hình vẽ một chu kỳ được ký hiệu là T_{CY} nó bao gồm một số trạng thái. Thời gian để CPU thực hiện trọn vẹn một lệnh được gọi là một chu kỳ lệnh (instruction cycle), một chu kỳ lệnh sẽ bao gồm nhiều chu kỳ máy. Tóm lại một chu kỳ lệnh được tạo bởi các chu kỳ máy, một chu kỳ máy được tạo bởi các trạng thái. Thời gian của một trạng thái sẽ tùy thuộc vào tần số tín hiệu xung clock cung cấp cho CPU.

Cách tốt nhất để phân tích giản đồ thời gian hình 5.5 là xem trục thời gian là một đường thẳng đứng cho di chuyển từ trái sang phải của sơ đồ. Bằng cách này có thể dễ dàng thấy sự tác động tuần tự của các tín hiệu của CPU trong các chu kỳ máy.

Trong T_1 của chu kỳ đọc bộ nhớ, trước hết CPU cung cấp tín hiệu IO/M mức thấp (nếu trong chu kỳ đọc công vào ra tín hiệu này sẽ ở mức cao). Trên giản đồ thời gian biểu diễn tín hiệu này ở cả hai mức vì nó có thể ở mức thấp hoặc mức cao trong chu kỳ đọc. Điểm giao nhau của hai mức chỉ thị tại thời điểm này tín hiệu bắt đầu có ý nghĩa cho chu kỳ máy hiện tại. Cũng giống như vậy trên các phần khác của giản đồ thời gian, điểm giao nhau giữa các đường sử dụng để chỉ thị thời điểm mà thông tin trên các đường hoặc nhóm đường thay đổi.

Sau khi cung cấp tín hiệu IO/M, CPU 8086 sẽ gửi mức cao ra đường ALE, tín hiệu này sẽ được nối tới cho phép ngõ vào (STB) của bộ cài. Ngõ vào của bộ cài sẽ được nối tới các đường AD0 – AD7, A16/S3 – A19/S6, BHE của 8086. Sau khi tác động tín hiệu ALE, 8086 sẽ gửi ra trên các đường này địa chỉ bộ nhớ mà nó muốn đọc. Vì bộ cài được cho phép bởi mức cao ở ngõ ra ALE nên các tín hiệu địa chỉ được chuyển qua và giữ lại ở các ngõ ra bộ cài. Sau đó 8086 sẽ kéo ngõ ra ALE của nó trở về mức thấp để cấm bộ cài. Địa chỉ ở các ngõ ra của bộ cài sẽ được giữ nguyên trên Bus địa chỉ cung cấp tới cho bộ nhớ và vào ra cho đến hết chu kỳ máy.

Chú ý hoạt động của Bus đa hợp địa chỉ/dữ liệu (A/D) biểu diễn trên giản đồ thời gian. Đầu tiên (tại điểm giao nhau) 8086 đưa giá trị địa chỉ hợp lệ ra các đường này, dạng hai mức trên sơ đồ chỉ thị các tín hiệu này không nhất thiết phải ở mức thấp hoặc cao.

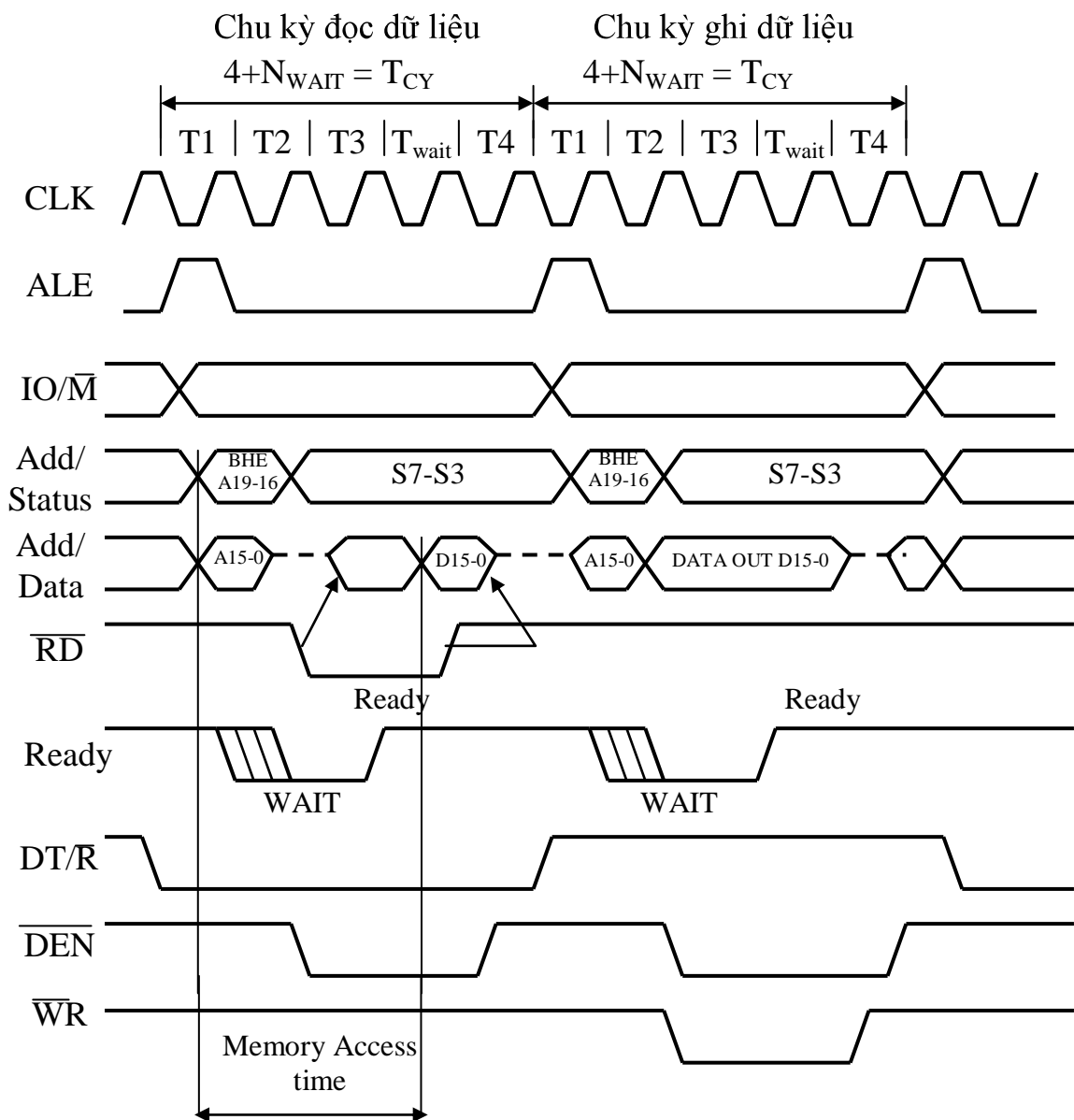
Sau đó ALE được đưa về mức thấp, 8086 không cần thiết duy trì các tín hiệu địa chỉ trên các đường AD nữa vì chúng đã được cài lại ở ngõ ra bộ cài. Do đó các đường này trở về trạng thái trở kháng cao biểu diễn bằng các đường đứt nét trên hình vẽ, 8086 thả nổi các đường AD0 – AD15 vì thế chúng có thể sử dụng để nhận dữ liệu từ bộ nhớ hoặc vào ra. Cùng thời điểm cung cấp A0 – A15, 8086 cũng cung cấp các tín hiệu BHE và A16 – A19, sau khi cài xong ALE xuống mức thấp các tín hiệu trạng thái sẽ được đưa ra trên các đường này.

8086 bây giờ sẵn sàng nhận dữ liệu của bộ nhớ hoặc vào ra tại địa chỉ mà nó đã cung cấp. Ở gần cuối trạng thái T_2 , 8086 sẽ tác động tín hiệu RD mức thấp, tín hiệu này sẽ sử dụng cho phép bộ nhớ hoặc vào ra cung cấp một byte hoặc một từ dữ liệu ra Bus dữ liệu. Điều này được chỉ thị trên hình 5.5 bằng mũi tên chỉ từ cạnh xuống của RD tới đường chỉ thị dạng sóng của các tín hiệu AD0 – AD15 (phần Bus sẵn sàng nhận dữ liệu). Dấu tròn ở cuối mũi tên chỉ thị tại cạnh hoặc mức tác động của tín hiệu, dấu mũi tên chỉ thị tác vụ mà tín hiệu sẽ thực hiện. Các mũi tên trong các giản đồ thời gian thường chỉ sử dụng để chỉ thị tác động của tín hiệu từ một thiết bị này sẽ gây ra trên một thiết bị khác, mà không sử dụng cho việc chỉ thị tác động của tín hiệu gây ra trong chính thiết bị đó.

Trở lại với dạng sóng của AD0 – AD15 ở phần cuối chu kỳ, nó chỉ thị thời gian bộ nhớ đưa dữ liệu ổn định ra Bus sau khi đã nhận được địa chỉ và tín hiệu cho phép RD. Nếu đáp ứng của bộ nhớ quá chậm, nó sẽ không đưa được dữ liệu ổn định, lúc này 8086 vẫn cứ nhận vào trạng thái bất kỳ trên Bus dữ liệu và chuyển qua chu kỳ máy kế tiếp, dữ liệu này sẽ thường làm kết quả của chương trình bị sai. Để giải quyết vấn đề này 8086 sử dụng ngõ vào READY, khi ngõ vào này ở mức cao 8086 sẽ thực hiện chu kỳ máy bình thường như đã mô tả ở trên. Nếu ngõ vào READY tác động mức thấp tại các trạng thái trước T_3 trong chu kỳ máy, 8086 sẽ chèn thêm một hoặc nhiều chu kỳ chờ giữa T_3 và T_4 . Như vậy các bộ nhớ hoặc thiết bị đáp ứng chậm sẽ phải có ngõ ra cung cấp tín hiệu READY cho CPU khi nó chưa đáp ứng kịp. Trong suốt thời gian

CPU thực hiện các trạng thái chờ, các tín hiệu địa chỉ và điều khiển sẽ vẫn giữ nguyên trạng thái của chúng. Khi READY trở về mức cao CPU sẽ thực hiện tiếp T_4 như bình thường. Với cơ chế đáp ứng bộ nhớ chậm như mô tả ở trên, hệ thống 8086 có thể sử dụng các bộ nhớ ROM đáp ứng chậm với giá thành rẻ chấp nhận thời gian thực hiện chương trình lâu hơn, khi CPU truy cập các bộ nhớ đáp ứng nhanh như RAM các trạng thái chờ sẽ không có.

Quay lại với giản đồ thời gian hình 5.5 để xem xét chức năng của các tín hiệu DEN và DT/R. Trong chu kỳ đọc 8086 tác động ngõ ra DT/R mức thấp trong T_1 , tín hiệu này sử dụng cho phép bộ đệm dữ liệu hoạt động ở chế độ nhận. Sau khi 8086 gửi xong 16 bit địa chỉ thấp ra Bus hệ thống nó tác động tín hiệu DEN mức thấp để cho phép bộ đệm Bus dữ liệu hoạt động. Dữ liệu do công vào ra hay bộ nhớ cung cấp ra trên Bus hệ thống sẽ tràn qua bộ đệm vào Bus dữ liệu của 8086.



Hình 5.5: Giản đồ thời gian hoạt động của 8086

Tóm lại sự tác động của các tín hiệu trong chu kỳ đọc bao gồm: 8086 tác động tín hiệu M/IO mức cao khi đọc bộ nhớ và mức thấp khi đọc dữ liệu từ vào ra. Ở cùng thời

điểm này 8086 tác động ALE mức cao cài địa chỉ và tín hiệu BHE ra Bus hệ thống. Sau đó các đường AD0 – AD15 được sử dụng cho việc nhận dữ liệu. Tín hiệu RD sẽ được tác động để cho phép bộ nhớ hoặc vào ra cung cấp dữ liệu ra Bus hệ thống, kết thúc chu kỳ đọc RD sẽ trở về mức cao. nếu ngõ vào READY tác động mức thấp trước hoặc trong trạng thái T₂, 8086 sẽ thực hiện thêm các trạng thái chờ cho đến khi READY trở lại mức cao, lúc này 8086 sẽ thực hiện tiếp trạng thái T₄ như bình thường. Các trạng thái chờ được sử dụng cho phép các thiết bị đáp ứng chậm có đủ thời gian cung cấp dữ liệu ổn định cho CPU. Nếu hệ thống có cấu hình lớn đòi hỏi phải sử dụng bộ đệm dữ liệu, tín hiệu DT/R sẽ được sử dụng để thiết lập chế độ truyền vào cho bộ đệm này trong suốt chu kỳ đọc, hoặc thiết lập bộ đệm ở chế độ truyền ra trong suốt chu kỳ ghi. Tín hiệu DEN sẽ cho phép bộ đệm dữ liệu hoạt động tại thời điểm thích hợp trong chu kỳ máy.

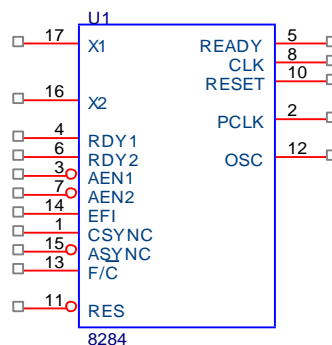
Một cách tương tự có thể giải thích sự tác động của các tín hiệu của 8086 trong các chu kỳ truy cập khác của 8086.

5.2. CÁC MẠCH PHỤ TRỢ TRONG HỆ THỐNG 8088/8086.

Trong hệ thống 8088/8086 hai mạch phụ trợ tiêu biểu thường gặp là bộ tạo xung nhịp 8284 và bộ điều khiển Bus 8288. Các vi mạch này tiếp tục được phát triển lên trong các hệ thống sau này của Intel.

5.2.1. Mạch tạo xung nhịp 8284.

Trong cả hai chế độ MIN và MAX 8088/8086 luôn cần xung đồng hồ cung cấp từ bộ tạo xung nhịp 8284, ngoài ra 8284 còn tạo đồng bộ cho các tín hiệu RESET và READY của CPU. Sơ đồ tín hiệu của 8284 trình bày trên hình 5.6, các tín hiệu bao gồm:



Hình 5.6: Sơ đồ tín hiệu của 8284

- AEN1, AEN2 (address Enable): là các tín hiệu ngõ vào sử dụng để chọn RDY1 hoặc RDY2 là ngõ vào thông báo trạng thái sẵn sàng của bộ nhớ hoặc vào ra.
- RDY1, RDY2 (Bus ready): là ngõ vào nhận tín hiệu thông báo tình trạng sẵn sàng của bộ nhớ và vào ra. Tín hiệu ngõ vào này sẽ được đồng bộ để cung cấp tới CPU, cho phép tạo ra các chu kỳ chờ khi bộ nhớ hoặc thiết bị ngoại vi chưa sẵn sàng truyền dữ liệu.
- ASYNC (ready synchronization select): Là ngõ vào chọn chế độ đồng bộ. Khi ASYNC = 1, ngõ vào RDY sẽ tạo ra tín hiệu READY cung cấp tới CPU cho tới hết cạnh xuống của xung nhịp tiếp theo. Khi ASYNC = 0, ngõ vào RDY chỉ tạo ra tín hiệu READY khi có cạnh xuống của xung nhịp tiếp theo.

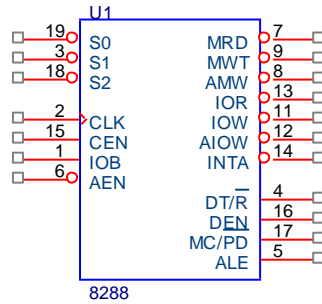
- READY: là ngõ ra cung cấp tín hiệu READY cho CPU khi có ngõ vào RDY tác động. Tín hiệu này sẽ được đồng bộ với các ngõ vào RDY.
- X1, X2 (XTAL): là các ngõ nối với tinh thể thạch anh, tần số của thạch anh nối tới hai ngõ vào này sẽ xác định tần số xung nhịp cung cấp cho toàn bộ hệ thống.
- F/C (Frequency/Crystal): là ngõ vào chọn nguồn tín hiệu chuẩn cho 8284. Khi ngõ vào này ở mức cao, xung nhịp bên ngoài sẽ được dùng làm xung nhịp cho 8284, ngược lại thì xung tạo bởi bộ dao động bên trong (sử dụng thạch anh nối tới các ngõ X1, X2) sẽ được sử dụng làm xung nhịp.
- EFI (external frequency input): là ngõ vào cung cấp xung nhịp từ bên ngoài định thời hoạt động cho 8284).
- CLK (clock): là ngõ ra cung cấp xung nhịp $f_{CLK} = f_X/3$, 77% chu kỳ nhiệm vụ đã được khuếch đại tới CPU và các vi mạch khác trong hệ thống.
- PCLK (peripheral clock): là ngõ ra cung cấp xung nhịp $f_{PCLK} = f_X/6$, 50% chu kỳ nhiệm vụ. Ngõ ra này thường được sử dụng cho việc cung cấp xung nhịp tới các bộ vào ra.
- OSC (Osc output): là ngõ ra cung cấp xung nhịp bằng f_X đã được khuếch đại.
- RES (reset input): là ngõ vào nhận tín hiệu Reset hệ thống (thường từ nút nhấn). Tín hiệu ngõ vào này sẽ được đồng bộ theo xung CLK để cung cấp cho CPU và các vi mạch khác.
- RESET: là ngõ ra cung cấp tín hiệu reset đã được đồng bộ tới CPU.
- CSYNC (Clock synchronization): là ngõ vào nhận xung đồng bộ chung trong các hệ thống mà 8284 sử dụng nguồn cung cấp xung nhịp từ dao động bên ngoài. Khi dùng mạch dao động bên trong, ngõ vào này cần được nối đất.

5.2.2. Mạch điều khiển Bus 8288.

Trong chế độ MAX, 8088/8086 không cung cấp các tín hiệu điều khiển trực tiếp tới bộ nhớ hoặc vào ra, mà các tín hiệu này được cung cấp thông qua bộ điều khiển Bus 8288. 8288 sẽ nhận các tín hiệu trạng thái S0, S1, S2 cung cấp từ CPU để tạo ra các tín hiệu điều khiển cho bộ nhớ và vào ra. Sơ đồ tín hiệu của 8288 biểu diễn trên hình 5.7 bao gồm:

- CLK (clock): là ngõ vào nhận xung nhịp của hệ thống để đồng bộ việc tạo ra các tín hiệu điều khiển của 8288.
- AEN (address enable): là tín hiệu ngõ vào sử dụng để kích hoạt các tín hiệu ngõ ra của 8288 sau thời gian trễ 150ns.
- S0, S1, S2 (status): là các ngõ vào lấy tín hiệu trạng thái cung cấp từ CPU trong chế độ MAX để tạo ra các tín hiệu điều khiển tương ứng như trong bảng sau:

S2	S1	S0	Chu kỳ điều khiển của Bus	Tín hiệu tác động
0	0	0	Chấp nhận yêu cầu ngắt	INTA
0	0	1	Đọc thiết bị ngoại vi	IORC
0	1	0	Ghi thiết bị ngoại vi	IOWC, AIOWC
0	1	1	Treo (halt)	KHÔNG
1	0	0	Đọc mã lệnh	MRDC
1	0	1	Đọc bộ nhớ	MRDC
1	1	0	Ghi bộ nhớ	MWTC
1	1	1	Bus rỗi (idle Bus)	AMWC



Hình 5.7: Các tín hiệu của 8288.

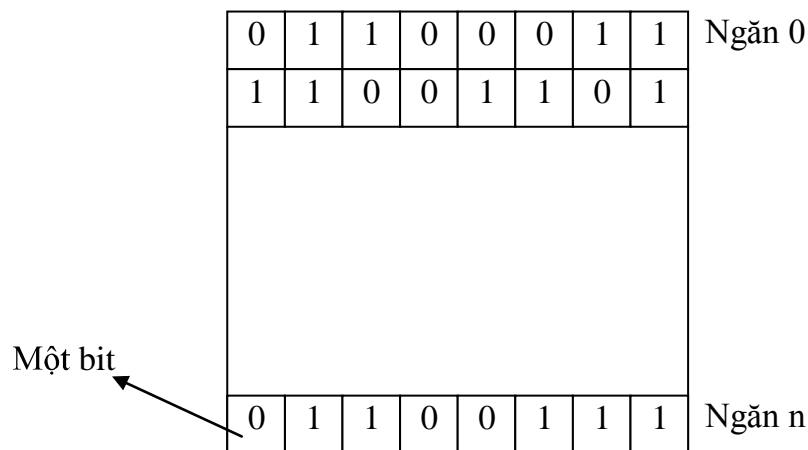
- CEN (command enable): là tín hiệu ngõ vào cho phép cung cấp tín hiệu DEN và các tín hiệu điều khiển khác của 8288.
- IOB (input/output Bus mode): là ngõ vào chọn các chế độ Bus khác nhau của 8288. Khi IOB = 1, 8288 làm việc ở chế độ Bus vào ra. Khi IOB = 0, 8288 làm việc ở chế độ Bus hệ thống.
- MRDC (memory read control): tín hiệu ngõ ra điều khiển đọc bộ nhớ, nó sử dụng để nối tới ngõ vào OE của bộ nhớ, để kích hoạt bộ nhớ cung cấp dữ liệu ra data Bus.
- MWTC (memory write control): là các tín hiệu ngõ ra điều khiển việc ghi dữ liệu vào bộ nhớ, AMWC (Advance MWTC): là tín hiệu ngõ ra điều khiển ghi bộ nhớ theo chu kỳ kéo dài, tín hiệu này được tạo ra sớm hơn so với tín hiệu MWTC để có thêm thời gian cho phép đối với bộ nhớ có đáp ứng chậm.
- IORC (input/output read control): là tín hiệu ngõ ra điều khiển việc đọc dữ liệu từ thiết bị ngoại vi.
- IOWC (input/output write control): là tín hiệu ngõ ra điều khiển việc đọc dữ liệu từ các thiết bị ngoại vi, AIOWC (Advance IOWC): là tín hiệu ngõ ra điều khiển ghi ngoại vi trong chu kỳ kéo dài, tác động sớm hơn so với IOWC.
- INTA (interrupt acknowledge): là tín hiệu ngõ ra thông báo chu kỳ CPU chấp nhận yêu cầu ngắt của thiết bị ngoại vi, lúc này thiết bị ngoại vi cần đưa vector ngắt ra Bus dữ liệu để CPU đọc.
- DT/R (data transfer/receive): là tín hiệu ngõ ra xác định chiều truyền dữ liệu của bộ đệm Bus dữ liệu trong hệ thống. (DT/R = 0: CPU đọc dữ liệu từ ngoài vào, DT/R = 1: CPU ghi dữ liệu ra bên ngoài).
- DEN (data enable): là tín hiệu ngõ ra cho phép bộ đệm Bus dữ liệu hệ thống hoạt động. Khi tín hiệu này tác động, CPU thực hiện chu kỳ truyền dữ liệu với thế giới bên ngoài.
- MCE/PDEN (master cascade enable/peripheral data enable): là tín hiệu ngõ ra cung cấp tới bộ điều khiển ngắt cho phép nó làm việc ở chế độ chủ.
- ALE (address latch enable): là tín hiệu ngõ ra cho phép việc tách địa chỉ tại các Bus đa hợp. Nó thường nối tới bộ chốt địa chỉ 74373, tác động trong chu kỳ xung nhịp đầu tiên trong mỗi chu kỳ truy cập ra bên ngoài của CPU.

5.3. NỐI GHÉP CPU VỚI BỘ NHỚ.

Bộ nhớ bán dẫn được sử dụng trong một hệ thống vi xử lý để làm nơi lưu giữ các chương trình, tức chuỗi các bit 0 và 1 cho vi xử lý làm việc. Đồng thời nó cũng là nơi lưu giữ các kết quả trung gian cũng như các kết quả cuối cùng (các hằng và biến) trong chương trình mà VXL thực hiện.

5.3.1. Khái niệm và các bộ loại bộ nhớ bán dẫn.

Có thể coi bộ nhớ bán dẫn giống như một chiếc tủ có nhiều ngăn, các ngăn được đánh số từ 0 tới hết gọi là các địa chỉ (hình 5.8). Trong mỗi ngăn lại được chia ra làm nhiều ngăn nhỏ, mỗi ngăn như vậy được gọi là một bit. Trong mỗi ngăn nhỏ có thể có hoặc không có chứa một loại đồ vật nào đó, ngăn chứa đồ tương ứng với bit 1 và ngược lại là bit 0. Nếu ta gọi các ngăn là các địa chỉ, thì tập hợp các ngăn ở trạng thái 0 và 1 trong một 1 ngăn sẽ được gọi là dữ liệu tại địa chỉ đó.



Hình 5.7: Nguyên tắc bộ nhớ bán dẫn.

Tuy nhiên bộ nhớ bán dẫn là một vi mạch số vì thế tất cả các dữ liệu và địa chỉ của nó đều là các tín hiệu số, hay các điện áp V_L và V_H tùy theo công nghệ chế tạo. Tức là việc đánh số cho các ngăn cũng sử dụng theo chuỗi số nhị phân.

Bộ nhớ bán dẫn hoạt động theo ba chế độ chính đó là: Ghi dữ liệu, đọc dữ liệu và chế độ chờ. Việc đọc ghi đều yêu cầu cung cấp một địa chỉ nhất định, các tín hiệu đọc ghi sẽ tùy thuộc vào từng loại bộ nhớ.

Dung lượng của bộ nhớ tùy thuộc và số đường địa chỉ và dữ liệu của nó. Ví dụ một bộ nhớ có mười đường địa chỉ và tám đường dữ liệu, sẽ có 2^{10} ô nhớ mỗi ô chứa 8 bit và dung lượng của nó sẽ là $2^{10} \times 8$ bit hay 1 KB.

Tính chất cơ bản nhất của bộ nhớ bán dẫn là các dữ liệu được địa chỉ hóa một cách duy nhất. Và các dữ liệu được truy cập theo địa chỉ đã qui định.

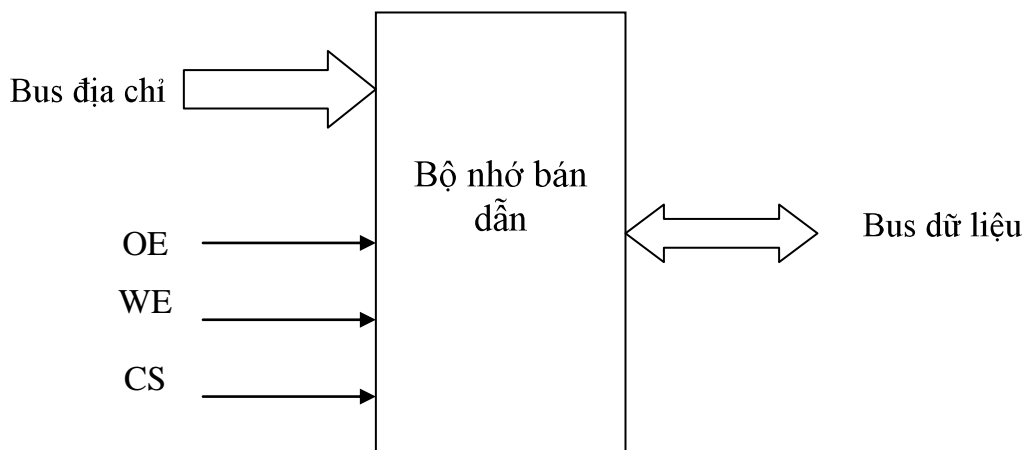
Bộ nhớ bán dẫn được chia thành hai loại chính đó là bộ nhớ ROM (Read Only Memory), và bộ nhớ có thể đọc ghi được, hay còn gọi là bộ nhớ truy cập ngẫu nhiên RAM (Random Access Memory).

Ý nghĩa chỉ đọc của bộ nhớ ROM là tương đối. Tùy theo các loại ROM mà người ta có các cách đưa dữ liệu vào trong nó khác nhau được gọi là lập trình ROM. Dữ liệu đã ghi vào sẽ không bị mất khi nguồn bị cắt. ROM được sử dụng để lưu các dữ liệu và các chương trình không đổi cho một hệ thống vi xử lý. Hầu hết các hệ thống VXL đều có một chương trình điều khiển chính nằm trong ROM ngay từ địa chỉ đầu tiên mà nó quản lý. Ví dụ chương trình ROM BIOS trong máy vi tính PC, để khởi động kiểm tra và thiết lập hoạt động cho máy.

Bộ nhớ RAM là bộ nhớ có thể đọc ghi được bằng các logic điều khiển thông thường. Tức là vi xử lý hoàn toàn có thể thay đổi được dữ liệu trong các ô nhớ RAM bằng các chu kỳ hoạt động thông thường của nó. Các biến của chương trình mà vi xử lý thực hiện thông thường đều được khai báo trong RAM. Dữ liệu lưu trong RAM sẽ bị mất khi bị mất nguồn. Ý nghĩa truy cập ngẫu nhiên là phân biệt với bộ nhớ truy cập tuần tự, như băng từ máy hát, video... Còn các loại bộ nhớ RAM và ROM đều có thể truy cập một cách ngẫu nhiên. Tức là có thể truy cập tới 1 ô nhớ bất kỳ trong nó mà không cần truy cập lần lượt từ đầu tới.

Cấu trúc tiêu biểu của một bộ nhớ bán dẫn bao gồm các tín hiệu như hình 5.8.

- Các tín hiệu địa chỉ (Bus địa chỉ) sử dụng để chọn ra vị trí của một ô nhớ cụ thể để đọc hoặc ghi. Mỗi trạng thái của các tín hiệu này sẽ định vị một ô nhớ duy nhất.
- Các tín hiệu dữ liệu (Bus dữ liệu) sử dụng cho việc truyền dữ liệu vào ra bộ nhớ tùy theo chế độ hoạt động được chọn của nó theo tín hiệu điều khiển.
- Các tín hiệu điều khiển bao gồm:
 - + Tín hiệu chọn mạch CS (chip select), hoặc CE (chip enable) sử dụng để chọn vi mạch nhớ hoạt động. Khi tín hiệu này không tích cực, vi mạch nhớ sẽ làm việc trong trạng thái chờ, lúc này nguồn tiêu thụ giảm xuống còn khoảng $\frac{1}{4}$ công suất làm việc. Các tín hiệu địa chỉ và dữ liệu sẽ ở trạng thái trở kháng cao (high Z).



Hình 5.8: Các tín hiệu của một bộ nhớ bán dẫn.

+ Tín hiệu cho phép xuất dữ liệu OE (output enable): khi bộ nhớ đã được chọn (CS tích cực). Nếu tín hiệu này tác động, dữ liệu trong ô nhớ đã được xác định bằng trạng thái Bus địa chỉ, sẽ được chuyển ra Bus dữ liệu. Lúc này bộ nhớ hoạt động trong chế độ đọc.

+ Tín hiệu cho phép ghi dữ liệu WE (Write enable): là tín hiệu cho phép bộ nhớ hoạt động ở chế độ ghi dữ liệu. Khi bộ nhớ đã được chọn (CS tích cực), nếu WE tích cực, giá trị trên Bus dữ liệu sẽ được ghi vào ô nhớ đã được định vị bằng Bus địa chỉ.

Đối với các bộ nhớ ROM, sẽ không có tín hiệu điều khiển ghi dữ liệu WE, việc ghi dữ liệu vào ROM được thực hiện bằng các qui trình đặc biệt tùy thuộc vào từng loại.

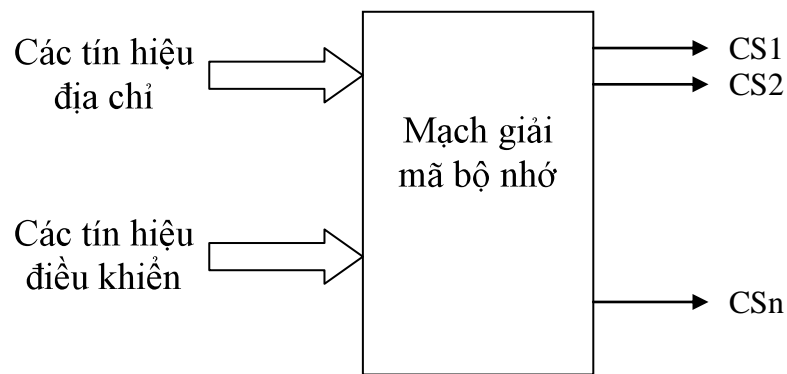
Một thông số quan trọng khác của bộ nhớ bán dẫn là thời gian truy cập t_{ac} (access time). Thông thường thời gian này được tính kể từ khi có giá trị cung cấp tới Bus địa chỉ cho đến khi có dữ liệu ổn định cung cấp ra trên Bus dữ liệu (trong chế độ đọc), và cho đến khi dữ liệu được ghi vào bộ nhớ trong chế độ ghi. Thời gian truy cập phụ

thuộc rất nhiều vào công nghệ chế tạo bán dẫn thực hiện bộ nhớ. Đối với các bộ vi xử lý có tốc độ truy cập nhanh, cần phải thực hiện mạch tạo trạng thái chờ để các bộ nhớ có tốc độ truy cập chậm đủ thời gian đáp ứng.

Chi tiết về cấu trúc của các bộ nhớ bán dẫn có thể tìm hiểu trong các giáo trình về mạch số. Trong phần này chỉ mô tả cách thức ghép nối bộ nhớ bán dẫn với bộ vi xử lý.

5.3.2. Giải mã địa chỉ cho bộ nhớ.

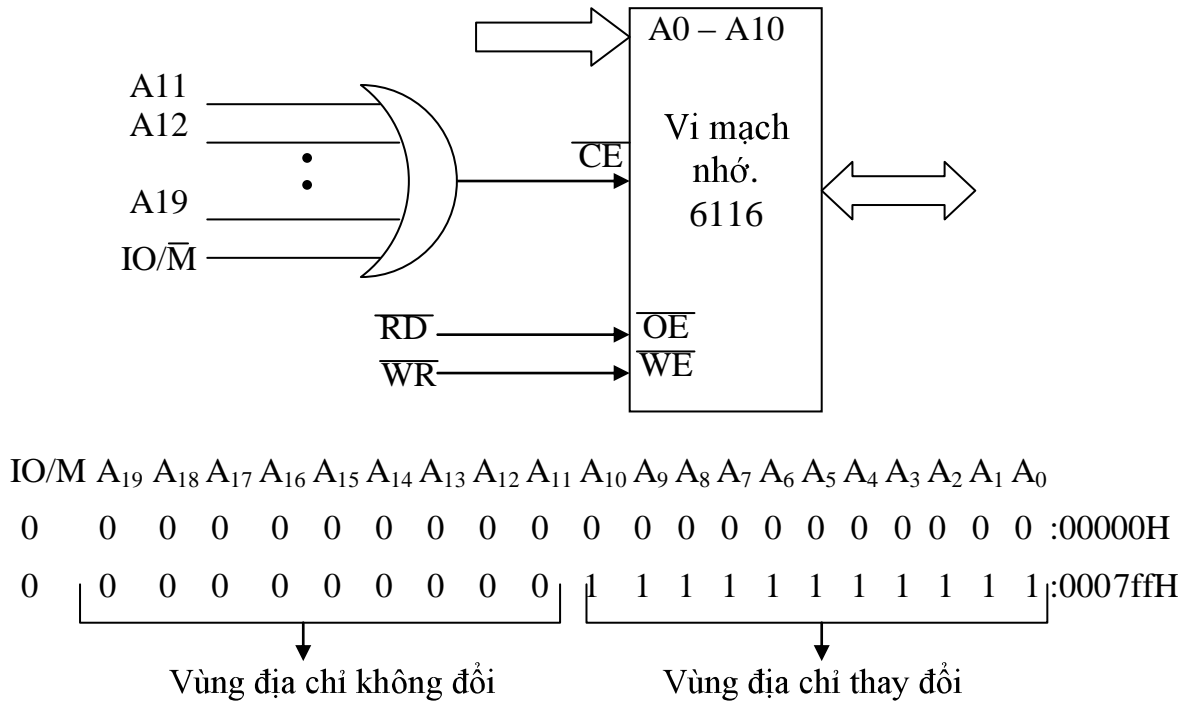
Theo nguyên tắc hoạt động của mình, vi xử lý thực hiện việc truy cập bộ nhớ theo địa chỉ mà nó cung cấp kèm theo các tín hiệu điều khiển thích hợp, và tại một thời điểm, với các tín hiệu địa chỉ cung cấp, vi xử lý chỉ truy cập tới duy nhất một trong các ô nhớ. Vì vậy, khi vùng nhớ mà vi xử lý quản lý bao gồm nhiều vi mạch nhớ, thì ứng với tất cả các giá trị địa chỉ mà nó có thể cung cấp, chúng ta phải kết nối sao cho chỉ có một vi mạch nhớ được chọn. Mạch thực hiện công việc này, được gọi là mạch giải mã bộ nhớ. Các tín hiệu của một mạch giải mã bộ nhớ có thể biểu diễn như hình 5.9.



Hình 5.9: Các tín hiệu giải mã bộ nhớ.

Ngõ vào của mạch giải mã địa chỉ bao gồm các tín hiệu điều khiển và địa chỉ. Tín hiệu điều khiển thông thường sử dụng để phân biệt chu kỳ truy xuất bộ nhớ hoặc vào/ra của vi xử lý, khi đó vi xử lý có thể sử dụng chung Bus địa chỉ cho việc truy cập bộ nhớ và vào ra. Còn các tín hiệu địa chỉ sẽ sử dụng cho việc xác định vùng nhớ cho từng vi mạch nhớ, thể hiện bằng mức thấp ở ngõ ra tương ứng chọn vi mạch nhớ đó. Ví dụ trong hệ thống 8088 ở chế độ MIN, tín hiệu điều khiển cho mạch giải mã địa chỉ sẽ là IO/M. Khi truy cập bộ nhớ IO/M=0, 20 đường địa chỉ A0 – A19 cung cấp từ 8088 sẽ được giải mã để cho phép một vi mạch nhớ làm việc. Còn khi truy cập vào ra, 8088 cung cấp 16 đường địa chỉ A0 – A15 (các đường địa chỉ cao A16 – A19 ở trạng thái mức 0), địa chỉ này có thể trùng với địa chỉ một ô nhớ nào đó, nhưng IO/M =1 nên vẫn không có vi mạch nhớ nào được chọn.

Để hiểu rõ hơn về phương pháp giải mã bộ nhớ, chúng ta xem xét một mạch giải mã bộ nhớ sử dụng cổng OR như hình 5.10. Trong mạch, bộ nhớ chỉ được phép làm việc (CE = 0) khi tất cả các tín hiệu địa chỉ ngõ vào cổng OR bằng 0 và IO/M = 0. Các đường địa chỉ thấp (A0 – A10) có thể thay đổi tất cả các trạng thái của chúng (từ 11 bit = 0 tới 11 bit =1), để chọn tất cả các ô nhớ trong vi mạch nhớ. Vùng địa chỉ cho vi mạch nhớ trong mạch từ 00000H tới 007ffH, đúng bằng dung lượng của vi mạch nhớ.



Hình 5.10: Mạch giải mã địa chỉ sử dụng cổng OR.

Khi vùng địa chỉ xác định từ vi xử lý đúng bằng dung lượng của vi mạch nhớ, người ta gọi là giải mã *đầy đủ*. Việc truy cập bộ nhớ vẫn có thể thực hiện đúng khi chúng ta giải mã *thiếu*, với vùng địa chỉ xác định từ vi xử lý lớn hơn dung lượng của vi mạch nhớ. Ví dụ trong sơ đồ giải mã hình 5.10, nếu không cung cấp tín hiệu A₁₁ tới cổng OR, thì vùng địa chỉ từ 00000H – 0007ffH hay vùng địa chỉ 000800H – 000fffH đều xác định tới vi mạch nhớ này, tức là mỗi ô nhớ của bộ nhớ này có thể truy cập bằng hai giá trị địa chỉ cung cấp từ vi xử lý. Nhưng trong trường hợp này, vùng địa chỉ dư ra sẽ không được sử dụng cho một vi mạch nhớ nào khác khi có nhu cầu tăng dung lượng nhớ cho hệ thống. Vì thế, việc giải mã thiếu chỉ sử dụng cho các hệ thống vi xử lý có yêu cầu về dung lượng nhớ nhỏ, để tiết kiệm về linh kiện cho mạch giải mã.

Khi hệ thống yêu cầu sử dụng nhiều vi mạch nhớ, có thể sử dụng nhiều cổng OR với nguyên tắc tương tự như hình 5.10. Nhưng mạch giải mã sẽ gọn hơn nếu giải mã bằng vi mạch giải mã (decoder) như hình 5.11.

Trong sơ đồ này bộ giải mã 74138 chỉ thực hiện giải mã khi A₁₉ = 1, IO/M = 0 và các đường A₁₈A₁₇ A₁₆ = 000. Nếu không thỏa các điều kiện này, toàn bộ các ngõ ra của mạch giải mã sẽ bằng 1, không có vi mạch nhớ nào được chọn. Khi đã thỏa mãn các điều kiện trên, ứng với 1 trạng thái của các ngõ vào A₁₅ – A₁₃ sẽ có một ngõ ra của bộ giải mã ở mức 0, một trong các vi mạch nhớ sẽ được cho phép truy cập. Như vậy, một ngõ ra mạch giải mã sẽ cho phép một vùng nhớ có dung lượng 2¹³ = 8K. Và toàn bộ mạch sẽ xác định được dung lượng nhớ là 8K x 8 = 68K. Nếu tín hiệu G1 không được cho phép bằng A₁₉, thì một vùng địa chỉ 64 K của hệ thống sẽ bị bỏ phí. Tương tự, cũng có thể sử dụng ngõ ra bộ giải mã để cho phép các bộ nhớ có dung lượng nhỏ hơn, nhưng khi đó sẽ có các vùng địa chỉ không bị bỏ phí.

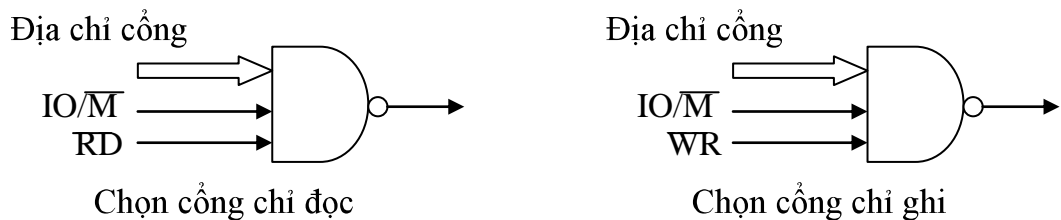
5.4. Nối ghép CPU với thiết bị ngoại vi.

Thiết bị ngoại vi đóng vai trò là các thiết bị phục vụ việc giao tiếp giữa người sử dụng và hệ thống vi xử lý, hoặc các thiết bị sản xuất chấp hành sự điều khiển của vi xử lý. Để kiểm soát việc truyền dữ liệu giữa CPU và các thiết bị ngoại vi, trong hệ thống vi xử lý sử dụng các vi mạch vào ra. Tùy theo cách hoạt động của các thiết bị ngoại vi cần phải sử dụng các vi mạch vào ra thích hợp.

Theo kiểu truyền dữ liệu số có thể phân biệt hai loại và ra: song song và nối tiếp. Theo trạng thái vật lý tín hiệu của thiết bị có : vào ra số và vào ra tương tự. Theo nguyên tắc điều khiển của CPU, thụ động hoặc có giao thức có thể phân biệt hai loại vào ra: vào ra cơ bản và vào ra có lập trình.

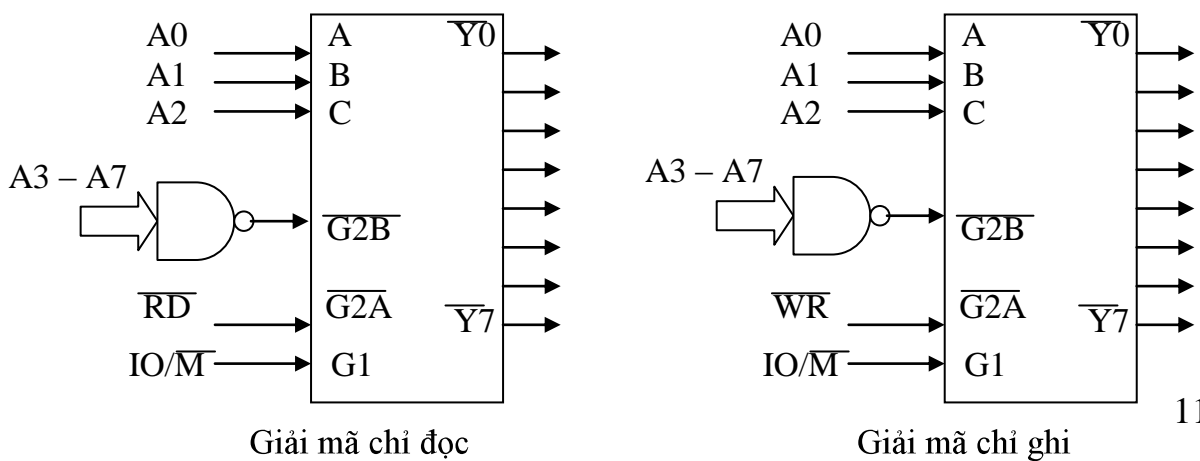
Hầu hết các loại vi xử lý đều có các tín hiệu phân biệt giữa việc truy cập bộ nhớ và truy cập vào ra. Ví dụ, với CPU 8088 tín hiệu IO/M sử dụng cho chức năng này. Khi IO/M = 0, 8088 thực hiện các chu kỳ truy cập bộ nhớ với 20 bit địa chỉ vật lý, IO/M = 1 nó thực hiện các chu kỳ truy xuất vào ra với chỉ 16 bit địa chỉ thấp. Tuy nhiên khi kết nối phần cứng tùy theo từng ứng dụng có thể sử dụng việc giải mã địa chỉ cho bộ nhớ và vào ra một cách riêng rẽ, hoặc có thể sử dụng chung. Ví dụ, có thể sử dụng chung vùng địa chỉ 1 MB của 8088 vừa cho bộ nhớ vừa cho việc truy cập vào ra, tuy nhiên lúc này việc truyền dữ liệu với vào ra cũng sẽ sử dụng các lệnh truy cập bộ nhớ như MOV, XCHG... mà không sử dụng các lệnh IN/OUT.

Việc giải mã địa chỉ vào ra cũng tương tự như giải mã địa chỉ cho bộ nhớ. Nhưng khác với các bộ nhớ (thông thường có bộ nhớ ROM chỉ đọc, bộ nhớ RAM có thể đọc ghi được, rất ít khi có bộ nhớ chỉ ghi), còn các thiết bị vào ra thì rất nhiều thiết bị chỉ đọc hoặc chỉ ghi dữ liệu. Nên ngoài các kiểu giải mã cho phép vừa đọc vừa ghi dữ liệu, có thể sử dụng thêm các tín hiệu RD và WR để giải mã cho các cổng vào ra chỉ đọc hoặc chỉ ghi, khi đó sẽ tăng thêm không gian địa chỉ cho các thiết bị ngoại vi (vì cùng một địa chỉ có thể chỉ thị hai thiết bị, một chỉ đọc và 1 chỉ ghi).



Hình 5.11: Mạch giải mã địa chỉ cơ bản cho các cổng vào ra.

Các bộ giải mã vào ra chọn một địa chỉ nhất định nào đó trong hệ thống có thể thực hiện bằng các cổng logic cơ bản. Hình 5.11 mô tả các bộ giải mã dùng cổng NAND. Trong hình vẽ, cổng vào ra sẽ được chọn khi tất cả các địa chỉ cung cấp tới cổng NAND ở mức 1, để có các địa chỉ khác có thể sử dụng thêm các cổng NOT để đảo trạng thái các tín hiệu địa chỉ thích hợp trước khi đưa vào NAND.

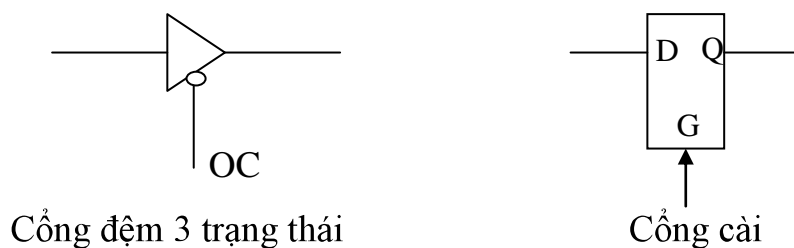


Khi sử dụng nhiều cổng vào ra với địa chỉ liên tiếp, có thể sử dụng các vi mạch giải mã như 74LS138, 74LS154 Hình 5.12 mô tả các kiểu giải mã khác nhau dùng vi mạch 74LS138.

5.4.1. Nối ghép song song.

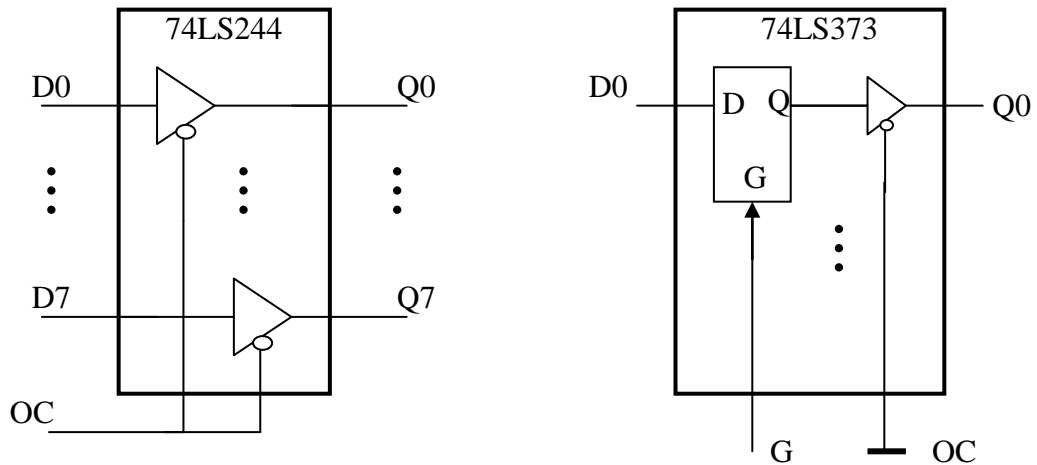
1. Các cổng vào ra song song cơ bản.

Các cổng vào ra song song được thực hiện theo nguyên tắc của các bộ đệm (buffer), cài (latch) trên hình 5.13. Với cổng đệm, khi tín hiệu cho phép OC ở trạng thái không tích cực (mức 1), ngõ ra sẽ ở trạng thái trở kháng cao, khi tín hiệu cho phép OC tích cực (mức 0), trạng thái ngõ ra sẽ có mức logic giống trạng thái ngõ vào. Còn đối với các cổng cài, khi tín hiệu cho phép G không tích cực, ngõ ra giữ nguyên trạng thái hiện tại của nó, nếu tín hiệu cho phép tích cực trạng thái logic ngõ ra sẽ tương ứng với logic ngõ vào. Trong các hệ thống vi xử lý, các bộ đệm cài này một đầu sẽ nối tới Bus dữ liệu của CPU, đầu còn lại nối tới thiết bị, ngõ vào cho phép sẽ được nối tới ngõ ra của giải mã địa chỉ. Khi truy cập dữ liệu, CPU cung cấp địa chỉ tương ứng làm ngõ vào cho phép của các cổng này tích cực, sau đó nó sẽ truyền dữ liệu với thiết bị qua cổng. Giải mã địa chỉ đảm bảo cho ứng với một giá trị địa chỉ của CPU cung cấp, chỉ có một cổng vào ra duy nhất được cho phép hoạt động.



Hình 5.13: Các thành phần tạo nên các cổng vào ra song song cơ bản.

Với nguyên tắc của các cổng đệm cài nêu trên, có rất nhiều mạch tích hợp mật độ vừa được sản xuất sử dụng làm các cổng vào ra cơ bản trong hệ thống vi xử lý. Thông thường các vi mạch này có 8 hoặc 16 đường vào ra, hình 5.14 mô tả cấu trúc của các vi mạch đệm 74LS244 và cài 74LS373.



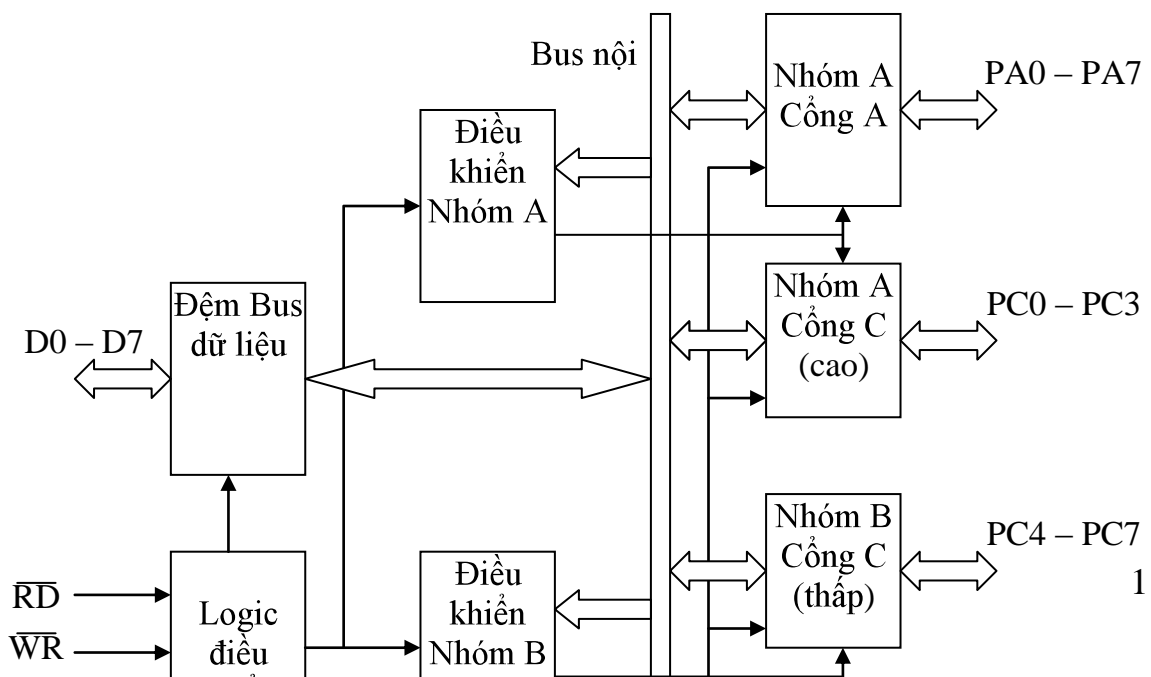
Hình 5.14: Các vi mạch vào ra cơ bản.

Với các mạch vào ra cơ bản, dữ liệu truyền giữa CPU và thiết bị ngoại vi hoàn toàn thụ động. Ví dụ, khi CPU cấp địa chỉ để đọc dữ liệu từ một cổng nào đó, có thể cổng đó chưa sẵn sàng làm việc, khi đó CPU sẽ nhận được một dữ liệu sai mà không xác định được. Để việc truyền dữ liệu với các thiết bị được đảm bảo hơn, trong các hệ thống vi xử lý thường sử dụng các bộ vào ra lập trình. Các vi mạch vào ra lập trình cho phép cung cấp các tín hiệu bắt tay giữa mạch vào ra và thiết bị nối ghép với nó, khi đó các bộ vào ra có thể thông báo các trạng thái hoạt động khác nhau của thiết bị cho CPU biết, và với cơ chế này dữ liệu truyền giữa CPU và thiết bị sẽ được đảm bảo.

Có rất nhiều bộ vào ra lập trình của các hãng khác nhau, có các chế độ hoạt động khác nhau. Phần sau sẽ mô tả bộ vào ra lập trình thông dụng của hãng Intel 8255. Các nguyên tắc hoạt động của nó có thể tham khảo để áp dụng cho các bộ vào ra lập trình song song khác.

2. Mạch vào ra song song có lập trình 8255.

Vi mạch 8255 được gọi là mạch nối ghép vào ra lập trình được (programmable peripheral interface – PPI) có sơ đồ khối biểu diễn trên hình 5.15.



8255 có ba cổng vào ra song song PA, PB và PC được chia thành hai nhóm A và B để điều khiển. Các cổng PA và PB có đệm cài cả ngõ vào và ngõ ra, còn cổng PC đệm cài ngõ ra, ngõ vào chỉ có đệm. Các cổng có các chế độ hoạt động khác nhau phụ thuộc vào sự điều khiển của các bộ điều khiển nhóm A và B. Bộ đệm Bus dữ liệu tương thích mức logic giữa CPU và thiết bị, đồng thời đóng vai trò kiểm soát việc truyền dữ liệu giữa các cổng vào ra và CPU. Khối logic điều khiển đọc ghi cung cấp tín hiệu chọn cổng và chọn chiều truyền dữ liệu tương ứng theo trạng thái của các tín hiệu điều khiển ngõ vào như mô tả trong bảng sau:

CS	A0	A1	RD	WR	Chế độ hoạt động
1	X	X	X	X	Không chọn mạch
0	0	0	0	1	Đọc PA
0	0	0	1	0	Ghi PA
0	0	1	0	1	Đọc PB
0	0	1	1	0	Ghi PB
0	1	0	0	1	Đọc PC
0	1	0	1	0	Ghi PC
0	1	1	1	0	Ghi từ điều khiển
0	1	1	0	1	Cắm

Các cổng của 8255 chỉ có thể truy cập được khi nó được lập trình trước, việc lập trình cho 8255 được thực hiện bằng cách ghi từ điều khiển tới địa chỉ cổng điều khiển của nó. 8255 có hai dạng từ điều khiển là từ điều khiển lập xoá bit cổng C và từ điều khiển chế độ, hai loại từ điều khiển này được xác định bằng trạng thái bit D7 của chúng.

0	x	x	x	C2	C1	C0	S/R
---	---	---	---	----	----	----	-----

Hình 5.16: Từ điều khiển lập xoá bit cổng C của 8255.

Từ điều khiển lập xoá bit cổng C biểu diễn trên hình 5.16, bit D7 cho từ điều khiển này luôn bằng 0, các bit D₆ D₅ D₄ có thể mang giá trị bất kỳ, các bit D₃ D₂ D₁ = C₂ C₁

C_0 mã hoá vị trí bit cổng C được lập xoá, bit $D_0 = S/R$ (Set/Reset) xác định việc lập hoặc xoá bit. Ví dụ, muốn lập bit PC2 của cổng C lên 1 có thể ghi dữ liệu 00000101 tới địa chỉ của cổng điều khiển.

1	M_{A1}	M_{A0}	A	C_H	M_B	B	C_L
---	----------	----------	---	-------	-------	---	-------

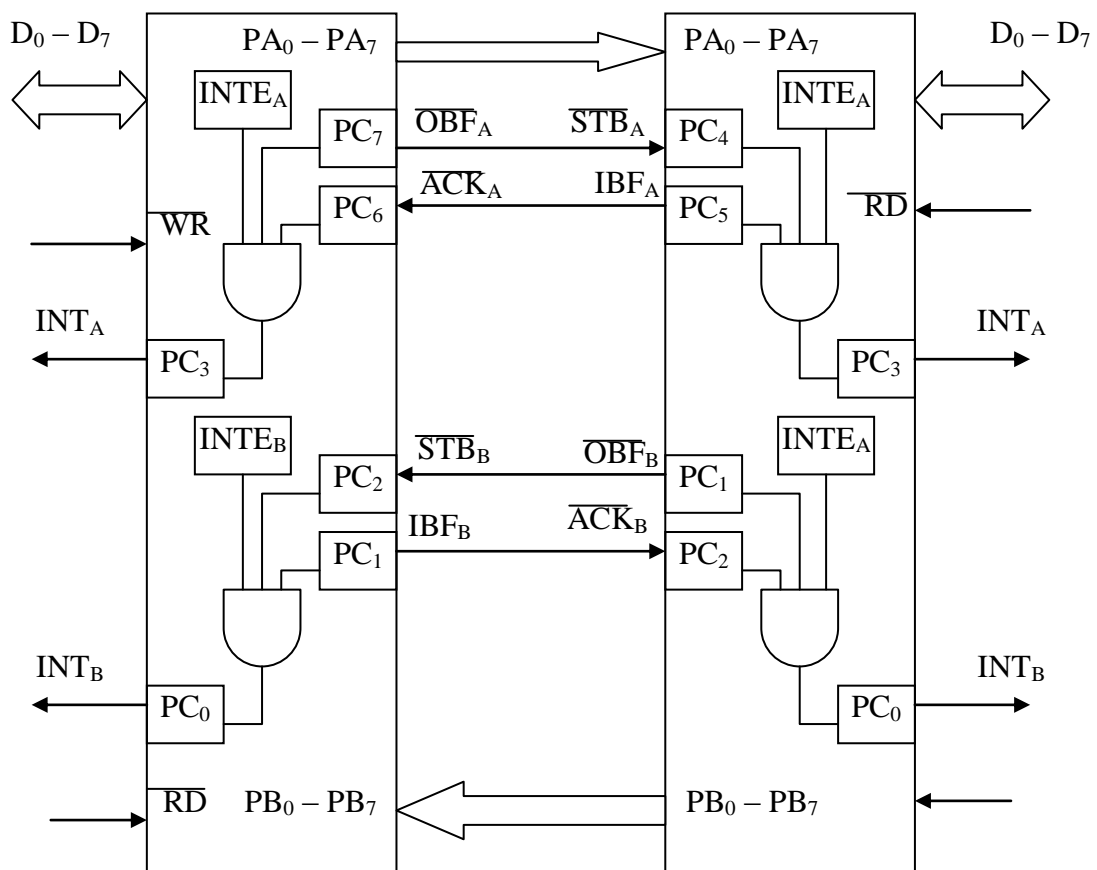
Hình 5.17: Từ điều khiển chọn chế độ của 8255.

Từ điều khiển chọn chế độ cho 8255 biểu diễn trên hình 5.17: bit D_7 của nó luôn bằng 1, $D_6 D_5 = M_{A1} M_{A0}$ chọn chế độ cho nhóm A bao gồm PA và phần cao PC (00: chế độ 0; 01: chế độ 1; 1x: chế độ 2), $D_4 = A$ sử dụng chọn chiều truyền dữ liệu cho PA (1: vào; 0: ra), $D_3 = C_H$ chọn chiều truyền dữ liệu cho PC phần cao, $D_2 = M_B$ chọn chế độ cho nhóm B gồm cổng PB và PC phần thấp (0: chế độ 0; 1: chế độ 1), $D_1 = B$ chọn chiều truyền cho PB và $D_0 = C_L$ chọn chiều truyền cho PC phần thấp.

Như vậy các cổng của 8255 có 3 chế độ làm việc:

- Chế độ 0: vào ra dữ liệu cơ bản, các cổng giống như các bộ đệm cài thông thường được chọn theo trạng thái của các ngõ vào địa chỉ A_0 và A_1 . Tùy theo việc lập trình từ điều khiển, các cổng này hoặc chỉ vào, hoặc chỉ ra.
- Chế độ 1: là chế độ truyền dữ liệu một chiều có bắt tay, chế độ này chỉ sử dụng cho các cổng PA và PB cổng PC sử dụng cho các tín hiệu bắt tay.
- Chế độ 2: là chế độ truyền dữ liệu hai chiều có bắt tay, chế độ này chỉ sử dụng cho PA, PC đóng vai trò là tín hiệu bắt tay, PB lúc này có thể hoạt động trong chế độ 0 hoặc 1.

Hình 5.17 mô tả việc truyền dữ liệu giữa hai bộ vào ra 8255 ở chế độ 1, trong đó 8255 bên trái có cổng A truyền, cổng B nhận, ngược lại 8255 bên phải có cổng A nhận, cổng B truyền dữ liệu.

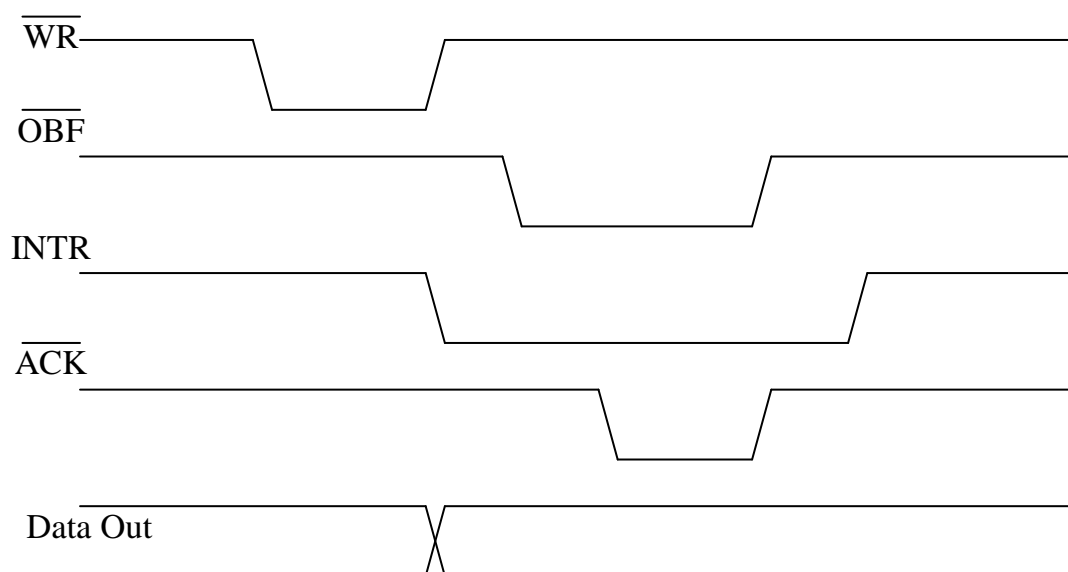


Hình 5.17: Truyền dữ liệu bằng 8255 ở chế độ 1.

Các tín hiệu của cổng truyền bao gồm:

- OBF (Output Buffer Full): là tín hiệu ngõ ra tác động mức thấp khi bộ đệm ngõ ra đầy. Tức là tín hiệu này sẽ tích cực khi CPU ghi tới cổng truyền 1 byte dữ liệu mà bên nhận chưa nhận lấy nó. Tín hiệu này sẽ được nối tới STB của cổng nhận, để chốt dữ liệu vào cổng nhận.
- ACK (acknowledge): là tín hiệu ngõ vào tác động mức thấp sử dụng cho việc nhận tín hiệu trả lời từ cổng nhận, thông báo để cổng truyền biết cổng nhận đã nhận xong dữ liệu, lúc này cổng truyền có thể truyền tiếp một dữ liệu khác.
- INTR (interrupt): là tín hiệu ra tác động mức cao thông báo một byte CPU ghi tới cổng truyền đã được truyền xong tới thiết bị. Tín hiệu này có thể sử dụng yêu cầu ngắt CPU để thực hiện việc ghi tới một dữ liệu khác.
- INTE (Interrupt Enable): là cờ cho phép ngắt truyền bên trong 8255, nó có thể lập xoá bằng bit PC₅ cho cổng A và PC₆ cho cổng B (khi truyền). Khi INTE bị xoá, yêu cầu sẽ không được gửi tới CPU khi 8255 truyền xong một dữ liệu.

Định thời các tín hiệu điều khiển cho cổng truyền biểu diễn trên hình 5.18

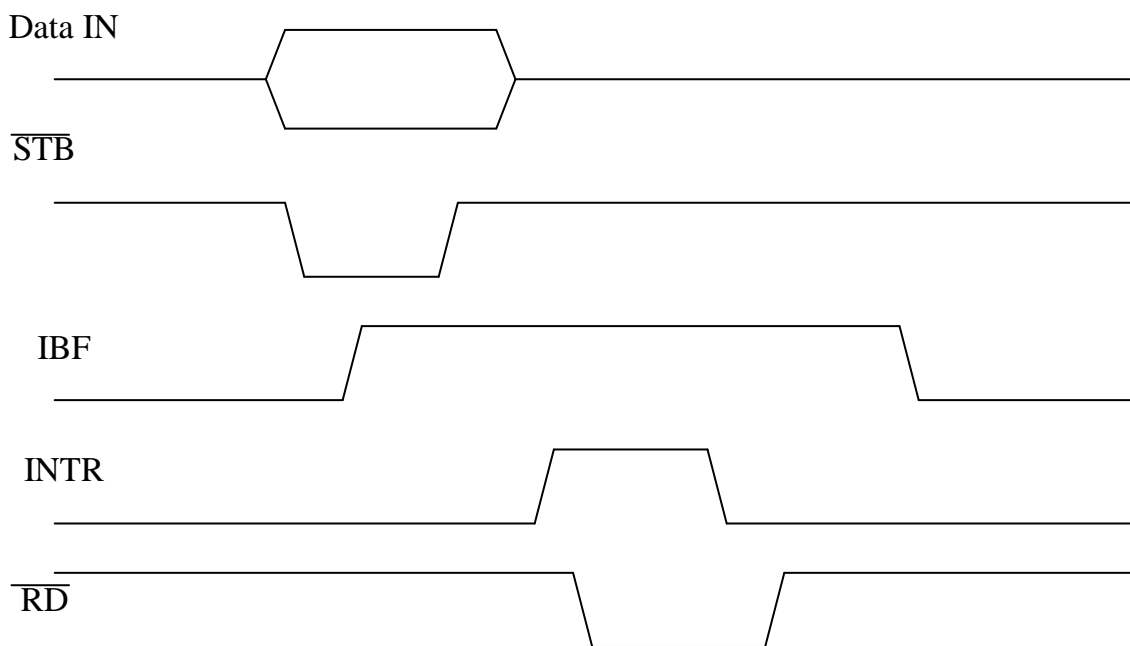


Hình 5.18: Định thời hoạt động các tín hiệu điều khiển cho PA chế độ 1
Các tín hiệu bắt tay cho cổng nhận dữ liệu trong chế độ 1 bao gồm:

- STB (Strobe): là ngõ vào, khi ngõ vào này tác động dữ liệu trên các đường nối tới cổng vào sẽ được chốt vào bộ đệm ngõ vào đó. Trên hình 5.17, tín hiệu này được cấp từ OBF của cổng truyền.
- IBF (Input Buffer Full): là tín hiệu ngõ ra tác động mức cao thông báo dữ liệu đã được chốt vào bộ đệm nhận, tín hiệu này chỉ được xoá khi CPU đọc dữ liệu.
- INTR (Interrupt Request): là tín hiệu yêu cầu ngắt, thông báo cho CPU biết đã có một dữ liệu sẵn sàng để CPU đọc, tín hiệu này chỉ được xoá khi CPU thực hiện một chu kỳ đọc kéo tín hiệu RD của nó xuống mức thấp.
- INTE (Interrupt Enable): cờ cho phép ngắt trong 8255 được lập xoá thông qua việc lập xoá bit PC₄.

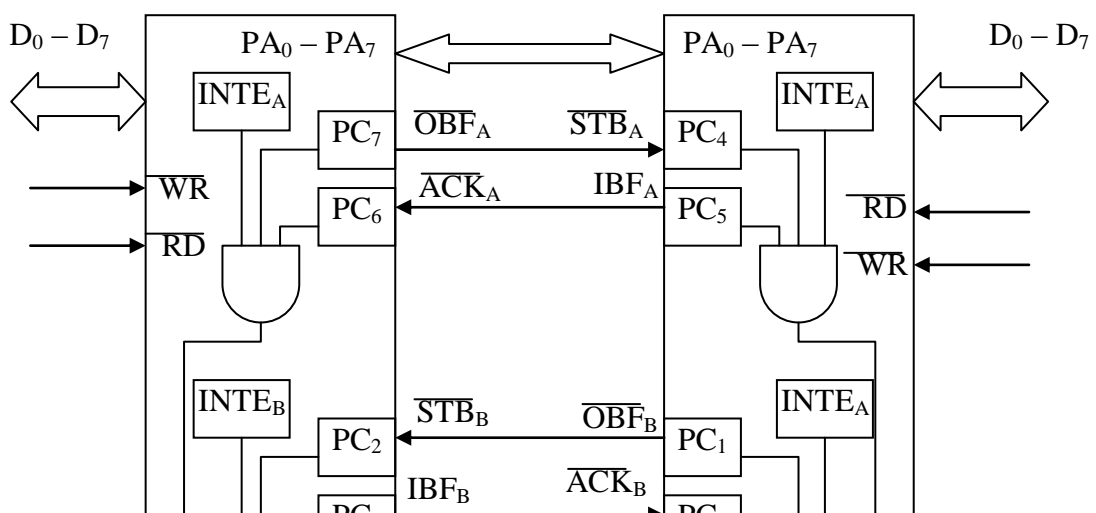
Định thời hoạt động của các tín hiệu bắt tay cho cổng nhận dữ liệu biểu diễn trên hình 5.19.

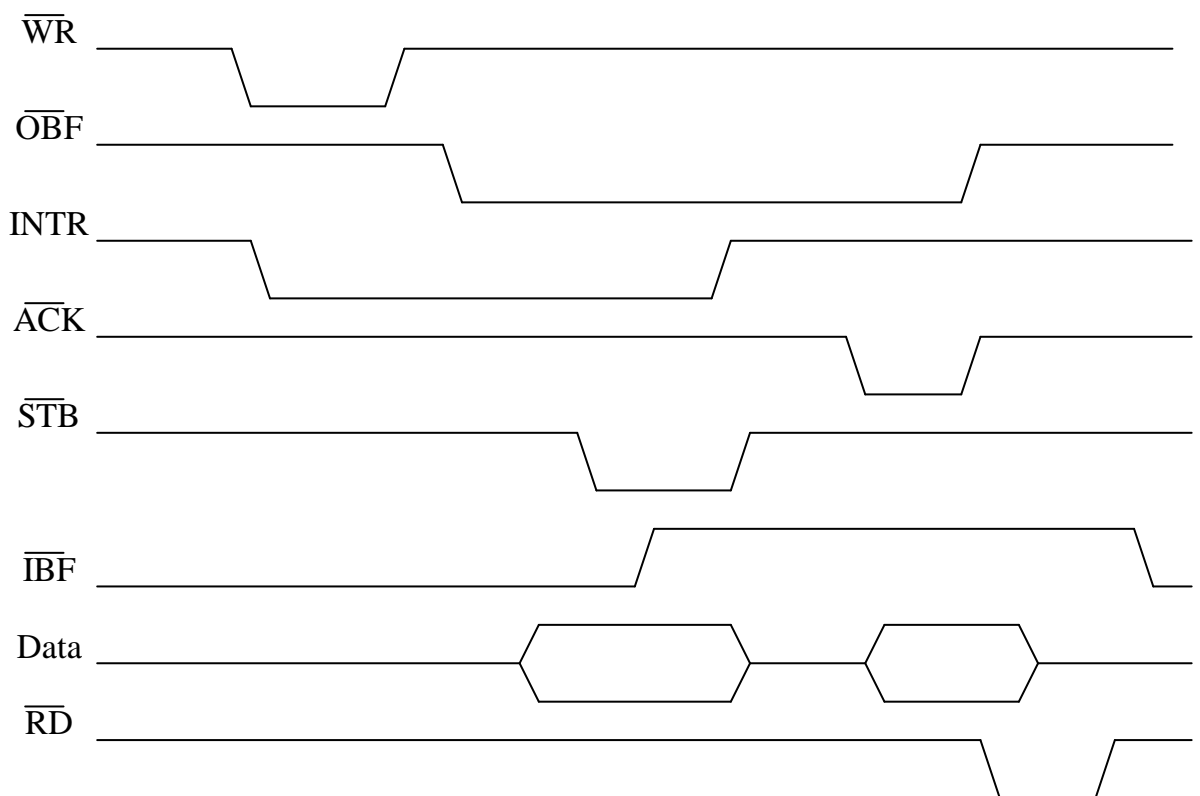
Có thể mô tả giao tiếp truyền nhận trong chế độ 1 như sau: trước hết khi cổng truyền còn trống, 8255 truyền sẽ yêu cầu CPU bên truyền ghi một byte tới cho nó bằng tín hiệu yêu cầu ngắt. Khi CPU ghi dữ liệu (tín hiệu WR tác động mức thấp), ngõ ra OBF của cổng truyền được kéo xuống mức thấp đưa tới ngõ vào STB để cài dữ liệu vào cổng nhận. Khi nhận được dữ liệu, cổng nhận thông báo trở lại cho cổng truyền bằng tín hiệu IBF mức cao đưa tới ngõ vào ACK, và đồng thời yêu cầu CPU điều khiển nhận đọc dữ liệu bằng tín hiệu INTR mức cao. Khi CPU chưa đọc, bộ nhận vẫn duy trì IBF = ACK = 1, do đó OBF của bộ truyền vẫn tác động. CPU đọc dữ liệu kéo RD của 8255 xuống mức thấp xóa tín hiệu yêu cầu ngắt, đồng thời IBF trở về mức 0 bên truyền sẽ dừng tác động tín hiệu OBF và yêu cầu ngắt để CPU ghi tới một dữ liệu mới.



Hình 5.20 mô tả nguyên tắc kết nối truyền dữ liệu của hai bộ vào ra 8255 ở chế độ 2 bằng cổng A.

Các tín hiệu bắt tay cho chế độ này có ý nghĩa tương tự như trong chế độ 1, riêng tín hiệu yêu cầu ngắt INTR sử dụng chung cả hai chiều truyền và nhận. Định thời hoạt động của các tín hiệu bắt tay trong chế độ 2 biểu diễn trên hình 5.21.





Hình 5.21: Định thời tín hiệu bắt tay trong chế độ 2.

5.4.2. Nối ghép nối tiếp.

Như đã giới thiệu, mọi hoạt động truyền dữ liệu của CPU với thế giới bên ngoài thông qua Bus dữ liệu bao gồm nhiều đường dây dẫn điện được gọi là truyền dữ liệu song song. Khi truyền dữ liệu tới các thiết bị ở xa, thì việc truyền bằng nhiều đường dây như vậy sẽ không kinh tế, mặt khác tín hiệu truyền có độ tin cậy kém. Từ các đòi hỏi ở trên xuất hiện phương pháp truyền dữ liệu nối tiếp, trong phương pháp này ở đầu phát dữ liệu song song sẽ được chuyển thành chuỗi bit nối tiếp truyền đi trên một đường dây tới bên thu. Ở đầu thu tín hiệu nối tiếp sẽ được biến đổi ngược lại thành dạng song song thích hợp cho việc xử lý tín hiệu tiếp theo.

Theo nguyên tắc trên, việc vào ra nối tiếp có thể sử dụng các vi mạch thanh ghi dịch, nhưng việc đồng bộ phải thực hiện khá phức tạp. Do đó thông thường việc truyền dữ liệu nối tiếp được thực hiện bằng các vi mạch chuyên dụng đa năng với nhiều chế độ hoạt động khác nhau có thể lập trình được. Sau đây sẽ mô tả hoạt động của một vi mạch truyền nối tiếp thông dụng: 8251 của Intel.

Hoạt động và chức năng của mạch giao tiếp nối tiếp 8251.

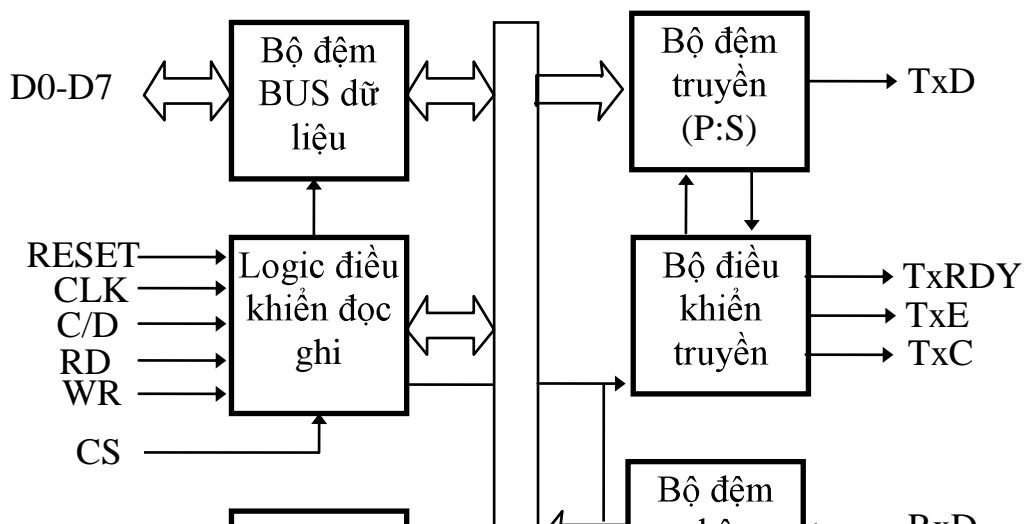
8251 là mạch giao tiếp nối tiếp trong hệ thống vi xử lý có các đặc tính chính sau:

- Hoạt động vào ra nối tiếp ở cả hai chế độ: đồng bộ và cận đồng bộ (Synchronous và Asynchronous).
- Truyền các kí tự từ 5 tới 8 bit trong chế độ đồng bộ. Đồng bộ có thể thực hiện bên trong hoặc bên ngoài. Các kí tự đồng bộ được tự động chèn vào chuỗi dữ liệu.
- Truyền các kí tự từ 5 tới 8 bit trong chế độ cận đồng bộ. Tốc độ clock cung cấp có thể bằng 1, 16 hoặc 64 lần tốc độ Baud truyền nhận dữ liệu. Có thể thực hiện việc phát kí tự ngắt truyền. Stop bit có thể là 1, 1 1/2, hoặc 2 bit. Có thể phát hiện các lỗi Start bit. Phát hiện và xử lý ngưng (break) tự động.
- Tốc độ truyền cực đại trong chế độ đồng bộ là 64K baud, trong chế độ cận đồng bộ là 19,2 K baud.
- Hoạt động truyền nhận dữ liệu đồng thời (song công), với hai bộ đệm truyền và nhận riêng biệt.
- Phát hiện các lỗi truyền như : chẵn lẻ, sai khung, overrun.
- Hoạt động tương thích với các vi xử lý họ Intel.
- Tất cả các tín hiệu vào ra tương thích logic TTL.

8251 USART (Universal Synchronous Asynchronous Receiver Transmitter) được thiết kế cho việc truyền dữ liệu giữa các vi xử lý họ Intel như MCS68, 80,85 và iPAX-86,88. 8251 sử dụng cho việc giao tiếp với các thiết bị ngoại vi nối tiếp, và khi đã được lập trình từ CPU nó có thể hoạt động trong hầu hết các chế độ truyền dữ liệu nối tiếp hiện có. Theo hướng truyền USART nhận dữ liệu song song từ CPU rồi chuyển chúng thành chuỗi dữ liệu nối tiếp liên tục để truyền. Đồng thời theo hướng nhận nó có thể nhận các chuỗi dữ liệu nối tiếp rồi biến đổi thành dạng song song để chuyển tới CPU. USART sẽ thông báo cho CPU mỗi khi nó nhận được một dữ liệu từ CPU để truyền, hoặc mỗi khi nhận được một dữ liệu từ thiết bị ngoại vi cho CPU. CPU có thể đọc được trạng thái hoạt động của 8251 tại mọi thời điểm, các trạng thái này bao gồm: các lỗi truyền dữ liệu, các tín hiệu điều khiển như SYNDET, TxEMPTY 8251 được sản xuất bằng công nghệ công bán dẫn kênh N.

1. Sơ đồ khối và chức năng hoạt động của các khối.

Nguyên tắc cấu tạo của 8251 biểu diễn trên hình 5.22 với các khối chức năng chính như sau:



*** Bộ đệm dữ liệu.**

Là bộ đệm 8 bit ba trạng thái hai chiều sử dụng cho việc giao tiếp giữa 8251 và Bus hệ thống của CPU. Dữ liệu có thể truyền nhận qua bộ đệm bằng các lệnh IN /OUT thông thường của CPU, với các tín hiệu điều khiển thích hợp đưa tới các chân C/D, RD, WR, CS. Các từ điều khiển, các từ lệnh khởi động cho 8251, các từ trạng thái, và các dữ liệu vào ra đều được truyền thông qua bộ đệm dữ liệu.

Các thanh ghi từ điều khiển và từ lệnh sẽ nhận dữ liệu từ CPU để định nghĩa các chế độ hoạt động khác nhau của 8251.

*** Khối logic điều khiển đọc ghi.**

Khối này sẽ nhận các tín hiệu cung cấp từ Bus điều khiển của hệ thống sẽ tạo ra các tín hiệu điều khiển cho toàn bộ hoạt động của 8251.

*** Khối điều khiển Modem.**

Tạo ra các tín hiệu bắt tay giao tiếp với modem cho phép truyền dữ liệu trên đường điện thoại.

*** Bộ đệm truyền.**

Khối này nhận dữ liệu song song từ bộ đệm dữ liệu, chuyển đổi thành chuỗi dữ liệu nối tiếp, chèn vào các bit hoặc các ký tự để tạo ra khung truyền tương ứng với các chế độ truyền khác nhau. Sau đó các bit sẽ được dịch ra đường TxD ở mỗi cạnh lên của xung Clock cung cấp vào TxC. Bộ truyền chỉ bắt đầu truyền khi được cho phép CTS=0. Đường TxD sẽ ở trạng thái treo khi thực hiện Reset hoặc khi CTS=1 hay khi bộ đệm truyền rỗng.

*** Bộ điều khiển truyền.**

Bộ đệm truyền quản lý tất cả các hoạt động liên quan đến việc truyền dữ liệu nối tiếp. Nó sẽ nhận các tín hiệu cả bên trong lẫn bên ngoài để thực hiện chức năng này.

*** Bộ đệm nhận.**

Bộ nhận lấy chuỗi dữ liệu nối tiếp trên ngõ RxD biến đổi thành dạng song song, kiểm tra sự duy nhất của các bit hoặc các ký tự tùy theo các chế độ thông tin và sau đó chuyển dữ liệu (đã loại bỏ các giao thức khung) tới CPU. Các bit trên đường RxD sẽ được dịch vào ở mỗi cạnh lên của tín hiệu RxC.

*** Bộ điều khiển nhận.**

Khối này có chức năng quản lý tất cả các hoạt động liên quan đến bộ nhận, bao gồm các chức năng sau:

Ngăn chặn các lỗi nhận dữ liệu của 8251, như xác định mức thấp trên đường RxD là trạng thái không sử dụng (điều kiện ngắt) hay trạng thái dữ liệu thấp. Trước khi bắt đầu nhận các kí tự nối tiếp, mức 1 phải có trên đường RxD ngay sau tín hiệu Reset. Sau đó bộ nhận sẽ kiểm tra để phát hiện bit thấp (Start bit) để xác định trạng thái bắt đầu nhận dữ liệu. Chức năng này chỉ hoạt động trong chế độ cận đồng bộ và chỉ thực hiện một lần sau mỗi lần Reset.

Ngăn chặn sai Start bit do xung nhiễu tức thời, bộ điều khiển nhận sẽ lấy mẫu start bit ở cạnh xuống và xác nhận lại nó một lần nữa bit trong khoảng giữa bit.

Phát hiện lỗi chẵn lẻ của dữ liệu nhận được và thiết lập bit thông báo lỗi tương ứng trong thanh ghi trạng thái.

Trong chế độ cận đồng bộ sẽ phát hiện lỗi sai khung khi không tìm thấy bit stop sau byte dữ liệu vừa nhận được.

2. Mô tả các tín hiệu của 8251.

* **RESET** : là một ngõ vào tác động mức cao để khởi động lại 8251, sau khi nhận được tín hiệu này 8251 sẽ trở về trạng thái ban đầu và cần phải lập trình lại các từ lệnh mới. Xung Reset cần kéo dài ít nhất trong 6 chu kỳ xung clock. Ngoài ra từ lệnh cũng có thể khởi động lại cho 8251, đó là chế độ khởi động mềm.

* **CLK**: (Clock) Là ngõ vào cung cấp xung nhịp định thời cho hoạt động cho các khối bên trong 8251. Tín hiệu này có thể lấy từ phase 2 (mức logic TTL) của bộ tạo clock. Không có ngõ vào ra nào tác động phụ thuộc vào CLK, nhưng tần số xung đưa tới ngõ vào này phải lớn hơn 30 lần tốc độ truyền nhận.

Các tín hiệu điều khiển chế độ hoạt động của 8251.

* **WR**: (Write) Ngõ vào tác động mức thấp, 8251 nhận tín hiệu này từ bus điều khiển hệ thống để xác định chế độ CPU ghi dữ liệu hoặc từ điều khiển tới 8251.

* **RD**: (Read) Là một ngõ vào tác động mức thấp nhận tín hiệu từ bus điều khiển hệ thống xác định chế độ CPU đọc dữ liệu hoặc từ trạng thái từ 8251.

* **C/D**: (Control/Data) Ngõ vào này tác động cùng với các tín hiệu RD hoặc WR để xác định tín hiệu ghi tới 8251 là dữ liệu hay từ điều khiển, và dữ liệu đọc từ 8251 là dữ liệu hay từ trạng thái.

* **CS**: (Chip Select) Là ngõ vào tác động mức thấp chọn mạch của 8251. Việc đọc ghi 8251 chỉ thực hiện được khi đã chọn mạch (CS=0). Khi CS=1 bus dữ liệu sẽ treo lên trạng thái trở kháng cao và các tín hiệu RD, WR sẽ không có tác dụng. Các chế độ hoạt động của 8251 tương ứng với các tín hiệu điều khiển được mô tả trong bảng sau:

CS	C/D	WR	RD	Chế độ
0	0	1	0	CPU ghi dữ liệu tới 8251
0	0	0	1	CPU đọc dữ liệu từ 8251
0	1	1	0	CPU ghi các từ lệnh tới 8251
0	1	0	1	CPU đọc từ trạng thái từ 8251
0	x	1	1	Data bus ở trở kháng cao
1	x	x	x	Không chọn mạch data bus ở trở kháng cao

Các tín hiệu điều khiển Modem.

8251 có một loại các tín hiệu tạo ra từ bộ điều khiển modem có thể giao tiếp một cách đơn giản tới hầu hết các modem. Các tín hiệu này tạo ra nhằm mục đích giao tiếp với các modem nhưng cũng có thể sử dụng các mục đích khác nếu cần thiết.

* **DSR** (Data Set Ready) Là một ngõ vào nối tới DSR của modem. Khi ngõ vào này tác động bit tương ứng với nó trong thanh ghi trạng thái sẽ tác động, vì thế CPU có thể kiểm tra ngõ vào này bằng phép đọc từ trạng thái. Ngõ vào này được sử dụng để kiểm tra trạng thái của modem. Modem tác động tín hiệu này bằng 0 là để thông báo tới CPU việc nó đã nối ghép hoàn chỉnh với đường điện thoại, và đang sẵn sàng cho việc truyền dữ liệu. Các modem quay số tự động sẽ gửi tín hiệu này tới CPU khi nó quay số thành công tới máy tính chủ.

***DTR** (Data Terminal Ready) Là một ngõ ra tác động mức thấp, nó có thể thiết lập bằng một bit tương ứng trong từ lệnh lập trình tới 8251. Tín hiệu này đưa tới modem để điều khiển hoạt động của modem. CPU sẽ tác động tín hiệu này khi nó sẵn sàng truyền dữ liệu với modem. Một số modem sẽ không thể thông báo trạng thái đã nối ghép được với đường điện thoại cho đến khi CPU tác động tín hiệu này. Một số modem không sử dụng tín hiệu này trong suốt quá trình truyền thông tin với hệ thống chủ. Một số modem có công tắc để chuyển chế độ sử dụng hoặc không sử dụng tín hiệu này.

* **RTS** (Request To Send) Là một ngõ ra tác động mức thấp nhờ lập trình bit thích hợp trong từ lệnh lập trình 8251. Tín hiệu này được gửi tới modem để yêu cầu gửi dữ liệu tới đường ra TxD. Tín hiệu này sử dụng kết hợp với CTS để điều khiển luồng dữ liệu giữa CPU và modem. Trong chế độ đồng bộ tín hiệu này sử dụng để điều khiển luồng tín hiệu giữa các modem.

* **CTS** (Clear to Send) Là ngõ vào tác động mức thấp cho phép 8251 truyền chuỗi dữ liệu nối tiếp khi bit TxEnable trong từ lệnh bằng 1. Còn nếu TxEnable = 0 hoặc CTS =1 trong khi đang truyền thì 8251 sẽ truyền xong hết tất cả các bit dữ liệu hiện có và ghi lệnh TxDisable trước khi ngưng. Tín hiệu này từ modem gửi tới CPU để thông báo nó sẵn sàng nhận dữ liệu. Trong chế độ đồng bộ tín hiệu này sử dụng điều khiển luồng thông tin giữa các modem.

Các tín hiệu truyền dữ liệu.

* **TxD:** (Transmit Data) Là ngõ ra sử dụng để truyền dữ liệu nối tiếp từ CPU ra ngoài thiết bị ngoại vi. Dữ liệu song song tại bộ đệm truyền sẽ dịch lần lượt ra ngõ này tại mỗi cạnh lên của TxC.

* **TxReady:** (Transmitter Ready) Ngõ ra này sử dụng thông báo cho CPU biết bộ đệm truyền đã sẵn sàng nhận một kí tự dữ liệu. Tín hiệu này có thể sử dụng ngắt CPU để yêu cầu CPU truyền tới 8251 một byte dữ liệu mới. Tín hiệu này sẽ bị che bởi bit TxEnable, CPU cũng có thể kiểm tra trạng thái ngõ ra này bởi phần mềm bằng lệnh đọc từ trạng thái của 8251 và kiểm tra bit tương ứng với nó. Ngõ ra này sẽ tự động reset ở cạnh xuống ở cạnh xuống của tín hiệu WR (tức là khi một kí tự dữ liệu được nạp tới từ CPU).

* **TxEmpty:** (Transmitter Empty) khi 8251 không có kí tự để gửi ngõ ra TxEmpty sẽ tác động mức cao. Nó sẽ được reset khi nhận được một kí tự từ CPU (nếu bộ đệm truyền được cho phép). Ngoài ra khi bộ truyền không được cho phép ngõ ra này cũng ở mức cao. TxEmpty có thể sử dụng để thông báo kết thúc một chế độ truyền, khi đó CPU sẽ hiểu cần phải chuyển đường truyền trong chế độ truyền nhận bán công.

Trong chế độ đồng bộ mức cao ở ngõ ra này thông báo không có kí tự dữ liệu nào được nạp tới, các kí tự đồng bộ bắt đầu được dịch ra một cách tự động. Trong khi các bit của kí tự đồng bộ được dịch ra ngoài thì ngõ ra TxEmpty vẫn giữ ở mức cao.

* **TxC:** (Transmitter Clock) là ngõ vào cung cấp xung nhịp điều khiển tốc độ truyền các kí tự. Trong chế độ đồng bộ tốc độ baud là 1x tức là bằng tần số xung TxC. Trong chế độ truyền cận đồng bộ tốc độ baud truyền sẽ bằng 1, 1/16, 1/64 lần tốc độ TxC tùy thuộc vào tốc độ đã lập trình là 1x, 16x hay 64x.

Ví dụ: nếu tốc độ baud là 110 baud

TxC = 110Hz trong chế độ 1x.

TxC = 1,72KHz trong chế độ 16x

TxC = 7.04 KHz trong chế độ 64x

Cạnh lên của TxC sẽ dịch các bit ra khỏi 8251.

Các tín hiệu cho việc nhận dữ liệu.

* **RxD:** (Receiver Data) Là ngõ vào sử dụng cho việc nhận dữ liệu nối tiếp từ bên ngoài vào 8251. Các bit của dữ liệu nối tiếp sẽ lần lượt được dịch vào bộ đệm nhận ở mỗi cạnh lên của RxC.

* **RxRDY:** (Receiver Ready) Là ngõ ra tác động cao chỉ thị 8251 có một dữ liệu sẵn sàng để CPU đọc. RxRDY có thể sử dụng là tín hiệu yêu cầu ngắt CPU, để CPU dừng công việc hiện tại đọc dữ liệu từ bộ đệm nhận của 8251. CPU cũng có thể kiểm tra trạng thái của ngõ vào này bằng cách đọc từ trạng thái của 8251.

Khi bit RxEnable trong từ lệnh không tác động thì RxRDY sẽ bị giữ ở trạng thái reset, ở chế độ cận đồng bộ RxRDY chỉ thiết lập khi bộ nhận được lập trình cho phép hoạt động, Start bit đã được gửi cùng với một kí tự đã được nhận xong chuyển sang dạng song song trong bộ đệm. Trong chế độ đồng bộ RxRDY cũng chỉ thiết lập khi bộ nhận được cho phép hoạt động, và kí tự dữ liệu đã được chuyển vào hoàn chỉnh trong thanh ghi dữ liệu.

Khi có một kí tự mới truyền tới, mà kí tự cũ chưa được đọc ra khỏi bộ đệm sẽ xảy ra lỗi overrun, bit báo lỗi trong thanh ghi trạng thái được thiết lập và kí tự cũ sẽ bị mất. Nếu dữ liệu bắt đầu đọc trong khi xảy ra việc truyền dữ liệu bên trong thì lỗi overrun cũng xảy ra tương tự như trên.

* **RxC:** (Receiver Clock) Ngõ vào nhận xung nhịp điều khiển tốc độ nhận dữ liệu nối tiếp. Trong chế độ đồng bộ tốc độ baud là 1x, tức là bằng với tần số của RxC. Trong chế độ cận đồng bộ tốc độ baud cũng tùy thuộc vào chế độ chọn trong khi khởi động 8251 là 1x, 16x hay 64x như khi truyền.

Ví dụ với tốc độ nhận là 300 baud thì:

- trong chế độ 1x tần số RxC phải là 300 Hz

- trong chế độ 16x tần số RxC phải là 4800 Hz

- trong chế độ 64x tần số RxC phải là 19.2 kHz

Dữ liệu được lấy mẫu vào ở mỗi cạnh lên của RxC.

Chú ý trong hầu hết các hệ thống sử dụng 8251 để truyền dữ liệu nối tiếp, thì 8251 sẽ xử lý cả hai hướng truyền và nhận trên một đường truyền bên ngoài (ví dụ như đường dây điện thoại). Do đó tốc độ truyền và nhận dữ liệu thường là bằng nhau. TxC và RxC sẽ được lấy chung từ một bộ tạo xung nhịp.

* **SYNDET:** (SYNC Detect/ BRKDET Break Detect) Tín hiệu này được sử dụng cho chức năng phát hiện đồng bộ SYNDET, và nó có thể sử dụng như một ngõ vào, hoặc một ngõ ra tùy thuộc vào từ điều khiển lập trình cho 8251. Nó sẽ bị reset về mức thấp

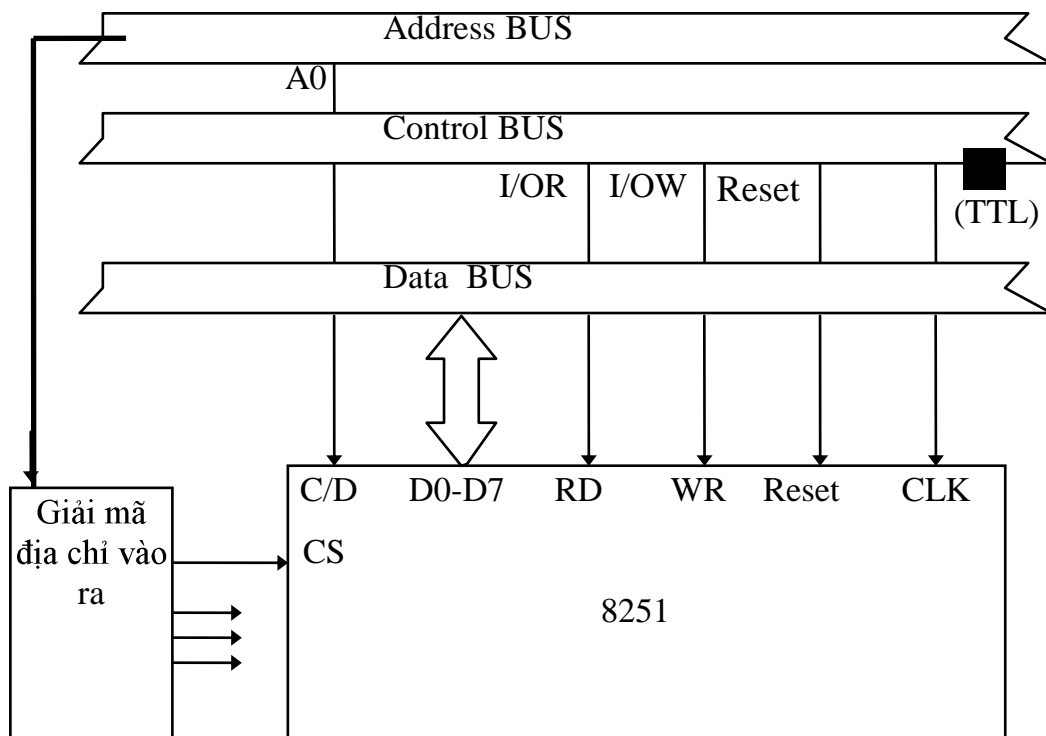
khi có tín hiệu Reset chip. Trong chế độ đồng bộ trong tín hiệu này là một ngõ ra, SYNDET sẽ tác động mức cao khi 8251 xác định được kí tự đồng bộ trong chế độ nhận. Nếu 8251 được lập trình ở chế độ có hai kí tự đồng bộ (bi-sync) thì SYNDET sẽ tác động tại thời điểm giữa của bit cuối cùng kí tự đồng bộ thứ hai. SYNDET sẽ tự động Reset khi đọc từ trạng thái.

Trong chế độ đồng bộ ngoài tín hiệu này là một ngõ vào, ngõ vào này tác động mức cao để thông báo rằng 8251 sẽ bắt đầu nhận dữ liệu vào ở cạnh lên của chu kỳ RxC kế tiếp. Khi chế độ đồng bộ ngoài được lập trình thì chế độ đồng bộ trong sẽ không được cho phép.

Trong chế độ truyền cận đồng bộ tín hiệu này mang chức năng BRKDET, nó sẽ tác động mức cao bất cứ khi nào bộ nhận phát hiện ra hai bit thấp liên tiếp sau stop bit (Trong chế độ cận đồng bộ khung dữ liệu truyền bao gồm : Start bit, các bit dữ liệu, bit chẵn lẻ và Stop bit). Phát hiện ngắt cũng có thể đọc trong từ trạng thái của 8251. Tín hiệu này sẽ bị xóa khi reset chip, hoặc khi ngõ vào RxD trở về mức cao.

3. Giao tiếp giữa 8251 và bus hệ thống.

Hình 5.23 mô tả việc nối ghép giữa 8251 và bus hệ thống của 8088, các giao tiếp khác cũng được thực hiện một cách tương tự.



Hình 5.23: Giao tiếp giữa 8251 và Bus hệ thống.

Địa chỉ mà CPU cung cấp để truy cập tới 8251 sẽ bao gồm hai phần: các bit địa chỉ cao sẽ đưa tới bộ giải mã vào ra để cung cấp tín hiệu chọn mạch cho 8251. Địa chỉ thấp A0 sử dụng cho việc xác định việc truyền dữ liệu giữa 8251 và CPU là dữ liệu thông thường hay các từ điều khiển hay trạng thái.

Bus dữ liệu sử dụng cho việc truyền dữ liệu, ghi các từ điều khiển và đọc từ trạng thái của 8251.

Các tín hiệu điều khiển còn lại được lấy từ bus hệ thống phục vụ cho các nhiệm vụ khác nhau như đã mô tả trong phần trên.

4. Hoạt động của 8251.

Mô tả chung.

Toàn bộ các chức năng hoạt động của 8251 được định nghĩa bằng phần mềm. Một loạt các từ điều khiển gửi từ CPU tới 8251 sẽ định nghĩa các dạng thông tin cụ thể. Các từ điều khiển này sẽ lập trình các thông số như: Tốc độ baud, chiều dài kí tự dữ liệu truyền nhận, số stop bit, chế độ truyền nhận là đồng bộ hay cận đồng bộ, kiểm tra chẵn lẻ là chẵn, lẻ hay không kiểm tra chẵn lẻ.... Trong chế độ đồng bộ còn định nghĩa việc đồng bộ thực hiện ở chế độ đồng bộ trong hay ngoài.

Sau khi được lập trình 8251 sẽ sẵn sàng thực hiện các chức năng thông tin của nó. Ngõ ra TxRDY sẽ tác động mức cao để thông báo cho CPU biết 8251 đã sẵn sàng nhận một kí tự dữ liệu từ CPU để truyền tới thiết bị ngoại vi. Tín hiệu này sẽ tự động được reset khi CPU ghi tới 8251 một kí tự. Theo hướng nhận 8251 nhận dữ liệu nối tiếp từ modem hay thiết bị vào ra. Khi nhận xong một kí tự vào ngõ ra RxRDY sẽ tác động mức cao để thông báo cho CPU biết 8251 đã có sẵn một dữ liệu để CPU đọc. Tín hiệu này sẽ tự động reset khi CPU đọc dữ liệu.

8251 sẽ không thể truyền dữ liệu được cho đến khi bit TxEnable (Transmitter Enable) trong từ lệnh được thiết lập, và nhận được tín hiệu CTS. Ngõ ra TxD sẽ treo ở mức cao sau khi reset.

Lập trình cho 8251.

Trước khi bắt đầu truyền nhận dữ liệu thì 8251 cần được nạp một tập các từ điều khiển ghi tới từ CPU. Các từ điều khiển này sẽ định nghĩa đầy đủ các tính năng hoạt động của 8251 và chúng cần được đưa tới ngay sau khi reset (cứng hoặc mềm) cho 8251.

Các từ điều khiển sẽ bao gồm hai dạng sau:

1. Từ chế độ (Mode Instruction)
2. Từ lệnh (Command Instruction).

Sau đây sẽ mô tả chi tiết các bit của các từ chế độ và từ lệnh và các tính năng hoạt động mà chúng quản lý.

- *Từ chế độ:*

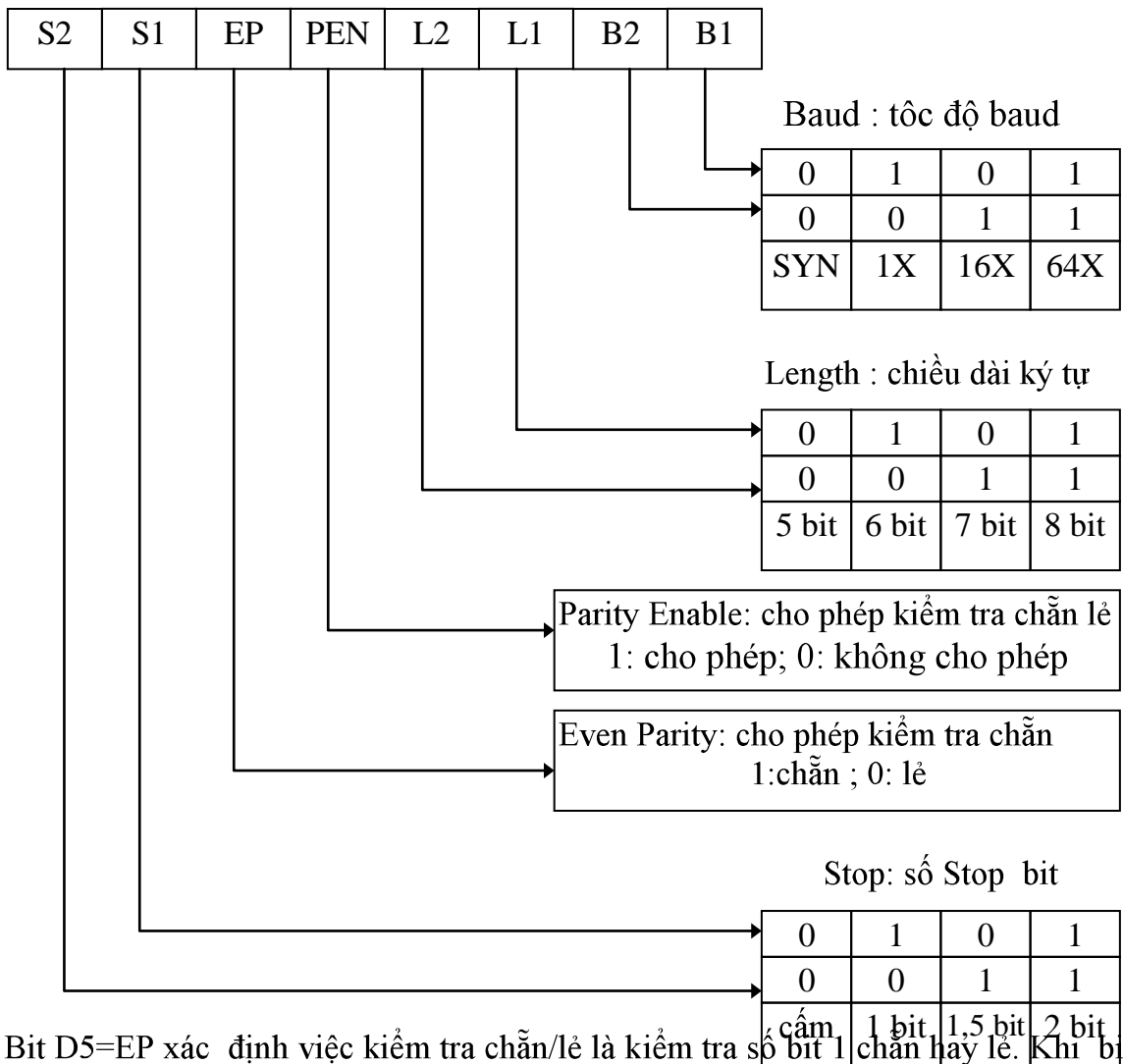
Từ chế độ sẽ định nghĩa các tính chất hoạt động chung cho 8251. Nó cần phải được ghi tới 8251 ngay sau khi reset, và chỉ khi nó đã được ghi tới thì các kí tự đồng bộ hay các từ lệnh mới có thể ghi tới được. Các bit của từ chế độ được định nghĩa như hình 5.24 bao gồm:

D1D0=B2B1 là hai bit định nghĩa tốc độ baud truyền nhận dữ liệu của 8251. Khi 2 bit này đều bằng 0 thì 8251 sẽ hoạt động trong chế độ đồng bộ, lúc này tốc độ baud sẽ bằng tần số xung nhịp. Khi hai bit này không đồng thời bằng 0, 8251 sẽ hoạt động trong chế độ cận đồng bộ và 3 tốc độ truyền có thể chọn tùy thuộc vào giá trị của các bit này.

D3D2=L2L1 là hai bit xác định chiều dài kí tự dữ liệu truyền từ 5 tới 8 bit. CPU luôn đọc và ghi dữ liệu với 8251 bằng bus dữ liệu 8 bit, do đó khi chiều dài dữ liệu chọn nhỏ hơn 8 thì khi ghi các bit có trọng số thấp sẽ bị bỏ qua, còn khi đọc thì các bit có trọng số thấp cung cấp từ 8251 sẽ bằng 0.

Bit D4=PEN xác định có cho phép kiểm tra chẵn lẻ hay không, nếu bit này bằng 1 việc kiểm tra chẵn lẻ được cho phép, lúc này 8251 sẽ tự động thêm bit chẵn lẻ vào sau

các bit dữ liệu khi truyền. Và khi nhận bit chẵn lẻ sẽ được kiểm tra xem có đúng với các bit dữ liệu đã nhận được hay không, nếu sai bit báo lỗi trong thanh ghi trạng thái sẽ được thiết lập.

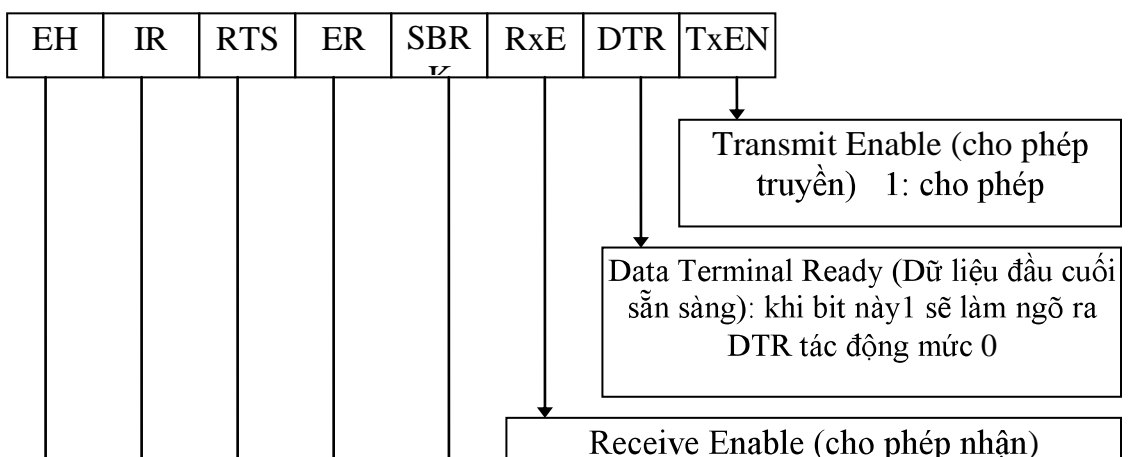


Bit D5=EP xác định việc kiểm tra chẵn/lẻ là kiểm tra số bit 1 chẵn hay lẻ. Khi bit này bằng 1 thì kiểm tra chẵn, lúc này bit chẵn lẻ sẽ bằng 1 khi số các bit 1 của dữ liệu truyền hoặc nhận là một số chẵn. Từ chế độ lập trình cho 8251.

D7D6=S2S1 là hai bit xác định chiều dài của bit Stop. Chiều dài của Stop bit có thể kéo dài trong thời gian của 1, 1 1/2, hoặc 2 bit tính theo tốc độ truyền nhận đã được qui định tùy thuộc vào giá trị của 2 bit này.

- Từ lệnh:

Từ lệnh lập trình cho 8251 được biểu diễn như hình 5.25 bao gồm 8 bit với các chức năng như sau:



Bit D0=TxEN là bit cho phép truyền dữ liệu hay không, khi bit được lập trình bằng 1 sẽ cho phép bộ truyền của 8251 và ngõ ra TxRDY, ngõ ra TxRDY sẽ tác động mức cao khi ngõ vào CTS tác động mức thấp và lúc này bộ đệm ở trạng thái sẵn sàng nhận một kí tự mới từ CPU để truyền

Bit D1=DTR là bit để thông báo trạng thái sẵn sàng của 8251, khi bit này được lập trình bằng 1 ngõ ra DTR sẽ tác động mức thấp.

Bit D2=RxE là bit sử dụng để cho phép bộ đệm nhận của 8251, khi bit này bằng 1 sẽ cho phép ngõ ra RxRDY của 8251 tác động mức cao khi bộ nhận có 1 kí tự sẵn sàng để CPU đọc vào.

Bit D3=SBRK là bit xác định có kí tự ngưng truyền gửi ra khi 8251 hết dữ liệu truyền hay không. Khi bit này bằng 1 ngõ ra TxD sẽ luôn bằng các bit 0 khi hết dữ liệu để truyền gọi là kí tự ngưng. Kí tự ngưng này sử dụng để xác định kết thúc truyền xong một khối dữ liệu.

Bit D4=ER là bit để reset các bit báo lỗi trong thanh ghi trạng thái của 8251 như bit PE báo lỗi chẵn lẻ, OE báo lỗi overrun, và FE báo lỗi sai khung truyền.

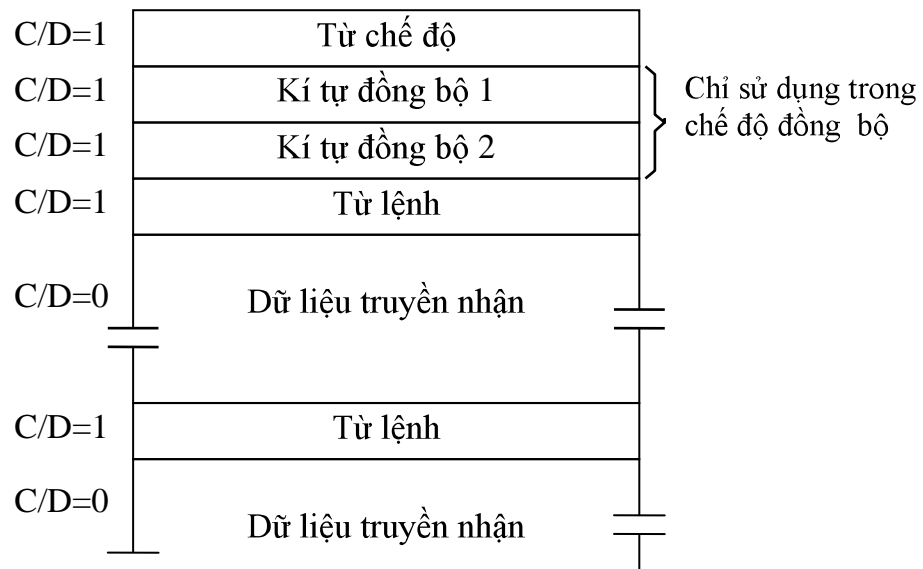
Bit D5=RTS là bit để lập trình cho ngõ ra RTS, khi bit này bằng 1 thì ngõ ra RTS sẽ tác động mức thấp, tín hiệu này sử dụng để hỏi modem đã sẵn sàng nhận dữ liệu chưa.

Bit D6=IR sử dụng để reset mềm cho 8251, khi bit này trong từ lệnh bằng 1, tác động sẽ tương đương với việc cấp xung mức cao vào chân reset của 8251. Lúc này cần phải gửi lại một từ chế độ mới.

Bit D7=EH: bit này chỉ sử dụng trong chế độ đồng bộ. Khi bit này được lập trình bằng 1 thì 8251 sẽ tìm các kí tự đồng bộ trong chuỗi bit bắt đầu dịch vào bộ nhận, cho

đến khi tìm thấy nó sẽ tác động tín hiệu ngõ ra SYNDET mức cao để thông báo cho bên phát bắt đầu truyền dữ liệu qua.

Việc lập trình cho 8251 được thực hiện theo các bước như sau:



Trước tiên khi mới bắt đầu hoạt động, hoặc ngay sau khi reset cứng hoặc mềm cần phải gửi tới cho 8251 từ chế độ. Công việc này được CPU thực hiện bằng lệnh ghi với địa chỉ đảm bảo C/D=1. Sau đó nếu chế độ đồng bộ được chọn trong từ chế độ thì cần gửi tới cho 8251 một hoặc hai kí tự đồng bộ. Các kí tự đồng bộ này được lưu bên trong 8251 để so sánh với các kí tự đồng bộ mà thiết bị bên ngoài gửi tới trong chế độ nhận đồng bộ. Và chúng sẽ được tự động phát ở mỗi đầu một chuỗi dữ liệu mà 8251 phát đi cho tới khi thiết bị bên ngoài đồng bộ được với nó. Tiếp theo là từ lệnh cần được gửi tới cho 8251.

Trong chế độ cận đồng bộ thì theo sau từ chế độ sẽ là từ lệnh. Khi từ lệnh đã được nạp 8251 có thể truyền nhận các khối dữ liệu nối tiếp như đã mô tả trong phần trên. Khi muốn thay đổi các đặc tính truyền nhận có thể lập trình lại cho 8251 một từ lệnh mới, còn muốn định nghĩa lại hoàn toàn cho 8251 thì cần thực hiện reset. Trong quá trình truyền nhận dữ liệu CPU cũng có thể kiểm tra lại trạng thái của 8251 để có một giải thuật truyền nhận hợp lý, đảm bảo độ an toàn của việc truyền nhận dữ liệu. Quá trình lập trình cho 8251 có thể mô tả trong hình 5.26. Ví dụ đoạn chương trình sau sử dụng cho việc lập trình cho 8251 hoạt động trong hệ thống vi xử lý họ Intel như sau:

```

MOV DX,PORT_ĐK      ; Nạp địa chỉ của điều khiển vào thanh ghi DX
MOV AL,00H          ; Nạp 0 vào thanh ghi lệnh rồi tạo trễ để đảm
OUT DX,AL           ; bảo 8251 đang ở chế độ nhận từ lệnh trước khi
MOV CX,02           ; được reset
D0: LOOP D0
   OUT DX,AL
   MOV CX,02
D1: LOOP D1
   OUT DX,AL
   MOV CX,02

```

```

D2: LOOP D2
    MOV AL,40H      ; Nạp từ lệnh reset mềm rồi tạo trễ để chuyển 8251 OUT
    DX,AL          ; về trạng thái chuẩn bị nhận từ lệnh chế độ
    MOV CX,02
D3: LOOP D3
    MOV AL,11001110B ; Nạp từ điều khiển định nghĩa 8251 hoạt động
    OUT DX,AL       ; với chế độ cận đồng bộ tỷ số tốc độ baud là
    MOV CX,02      ; 16x (d1d0=10), chiều dài kí tự 8 bit (d3d2=11),
D4: LOOP D4        ; không cho phép kiểm tra chẵn lẻ (d4=0), 2 stop
    MOV AL,00110111B ; bit Nạp từ lệnh định nghĩa đặc tính của 8251:
    OUT DX,AL      ; cho phép truyền, tác động ngõ ra DTR, cho phép

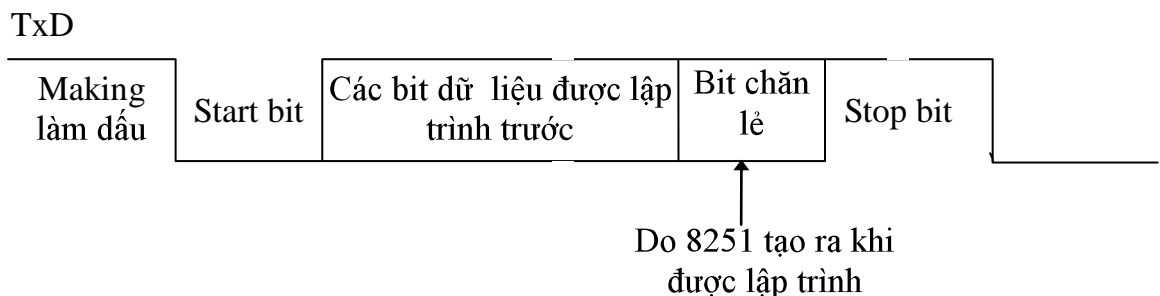
```

nhận, chế độ hoạt động thông thường (không gửi kí tự ngưng truyền), xóa tất cả các cờ lỗi, tác động ngõ ra RTS, không reset mềm, không cho phép chế độ tìm kí tự đồng bộ.

Khi bắt đầu cấp nguồn 8251 không phải luôn luôn đáp ứng đúng chế độ reset cứng, vì vậy chuỗi lệnh ban đầu gửi 3 byte 0 liên tiếp và một byte lệnh reset tới địa chỉ thanh ghi điều khiển của 8251, để đảm bảo nó được reset trước khi nhận từ lệnh chế độ. Sau khi đã được reset một cách đảm bảo từ lệnh chế độ có thể gửi tới cho 8251. Chú ý 8251 phân biệt từ chế độ và từ lệnh bằng thứ tự mà CPU ghi tới nó. Sau khi reset là từ chế độ, các từ gửi tới địa chỉ thanh ghi điều khiển sau từ chế độ sẽ được xem như các từ lệnh cho đến khi 8251 được reset trở lại. Thời gian tạo trễ sau mỗi lần gửi từ lệnh tới cho 8251 cần phải đảm bảo lớn hơn 16 lần chu kỳ xung nhịp cấp tới ngõ vào CLK. Tùy theo từng hệ thống cụ thể mà có thể thay đổi giá trị nạp vào CX để có độ trễ hợp lý. Trong ví dụ trên tần số xung nhịp cấp cho CPU và 8251 bằng nhau nên giá trị nạp cho CX là 02. Lệnh MOV CX,02 chiếm 4 chu kỳ xung nhịp, giá trị của CX được sử dụng đếm số vòng lặp cho lệnh LOOP nên lệnh này sẽ được thực hiện 2 lần. Lần đầu lệnh LOOP chiếm 17 chu kỳ xung nhịp, lần sau nó chiếm 5 chu kỳ nữa. Lệnh OUT chiếm 8 chu kỳ, như vậy với vòng lặp này thời gian trễ lớn hơn giá trị cần thiết là 16 chu kỳ xung nhịp.

Chế độ truyền cận đồng bộ.

Trong chế độ này bất cứ khi nào có một kí tự gửi tới từ CPU, sẽ được tự động cộng thêm một Start bit mức thấp, sau đó là các bit dữ liệu (bit có trọng số thấp nhất trước tiên), tiếp theo sẽ là bit chẵn lẻ nếu được lập trình trong từ lệnh chế độ, và cuối cùng là các bit Stop. Sau đó các bit trên sẽ được dịch ra ngõ TxD thành một chuỗi bit nối tiếp, chuỗi dữ liệu được dịch ra ở mỗi cạnh lên của xung TxC với tốc độ baud bằng 1, 1/16, 1/64 tốc độ TxC tùy theo lập trình trong từ lệnh chế độ. Các kí tự ngưng truyền (các bit 0 liên tục) sẽ được truyền tiếp ra ngõ ra TxD nếu từ lệnh cho phép. Khi không có dữ liệu nạp tới 8251 ngõ ra TxD sẽ được giữ ở mức cao nếu không lập trình cho phép kí tự ngắt. Hình 5.26 mô tả khung truyền dữ liệu trong chế độ truyền cận đồng bộ:

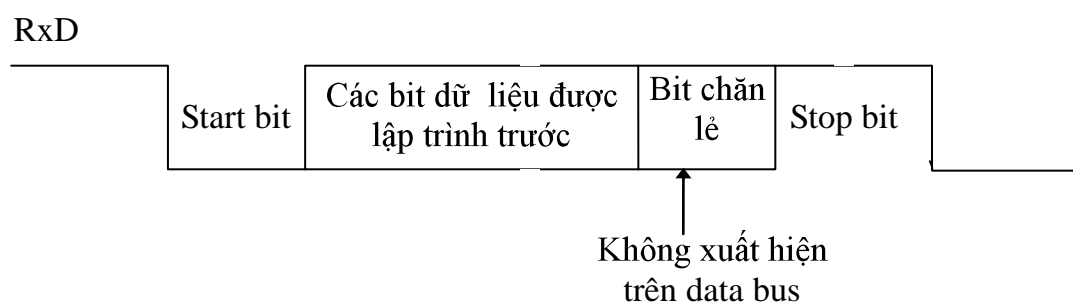


Hình 5.26: Khung truyền dữ liệu bên truyền trong chế độ cận đồng bộ.

Chế độ nhận cận đồng bộ.

Đường RxD bình thường ở trạng thái cao, khi xuất hiện một cạnh xuống trên đường mà là bắt đầu Start bit của một kí tự mới, giá trị Start bit sẽ được kiểm tra lại một lần nữa vào thời điểm giữa của nó (chỉ sử dụng trong chế độ 16x hoặc 64x). Khi kiểm tra lại nếu vẫn tìm thấy mức thấp thì Start bit được coi là hợp lệ, và bộ đếm bit dữ liệu bắt đầu hoạt động. Bộ đếm sẽ tăng giá trị của nó tại thời điểm giữa của các bit dữ liệu (kể cả bit chẵn lẻ nếu có và stop bit). Nếu có lỗi chẵn lẻ thì cờ báo lỗi chẵn lẻ sẽ được thiết lập. Dữ liệu và bit chẵn lẻ sẽ được lấy mẫu tại chân RxD tại cạnh lên của RxC. Nếu mức thấp xuất hiện tại vị trí Stop bit thì cờ báo lỗi sai khung sẽ được thiết lập, lỗi sai khung có thể xảy ra do nhiễu, hoặc do việc lập trình giao thức khung giữa bên truyền và bên nhận khác nhau. Ví dụ bên truyền lập trình truyền các kí tự có chiều dài 5 bit, mà bên nhận lập trình các kí tự có chiều dài 8 bit thì sẽ xảy ra lỗi sai khung. Stop bit sẽ thông báo kết thúc của một kí tự.

Chú ý là bên nhận chỉ yêu cầu 1 stop bit mà không quan tâm tới số stop bit được lập trình ở bên nhận. Kí tự nhận được sẽ nạp vào bộ đệm song song của 8251. Tín hiệu RxRDY lúc này sẽ tác động mức cao để thông báo tới CPU đã có một kí tự sẵn sàng để đọc vào. Nếu kí tự nhận trước chưa được CPU đọc vào, mà kí tự hiện thời tới thay thế nó trong bộ đệm nhận thì sẽ xảy ra lỗi Overrun, và cờ báo lỗi này trong thanh ghi trạng thái sẽ được lập (kí tự trước sẽ bị mất). Tất cả các cờ lỗi có thể kiểm tra được trong thanh ghi trạng thái, và có thể xóa bằng cách lập bit ER trong thanh ghi từ lệnh lập trình tới 8251. Không có lỗi nào gây ảnh hưởng tới hoạt động của 8251. Hình 5.27 biểu diễn khung truyền dữ liệu của bên nhận:



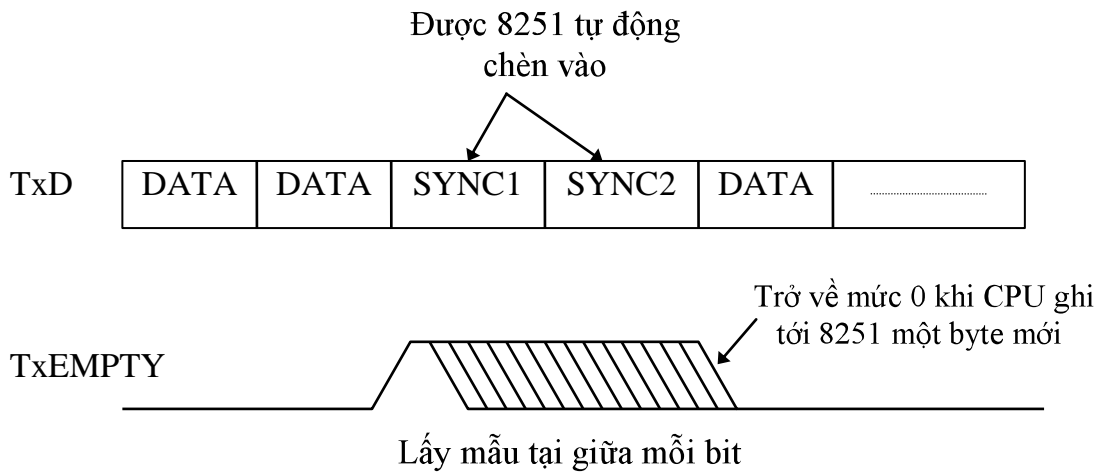
Hình 5.27: Khung truyền dữ liệu bên nhận trong chế độ cận đồng bộ.

Chế độ truyền đồng bộ.

Ngõ ra TxD sẽ giữ ở mức cao cho tới khi CPU gửi kí tự đầu tiên của nó tới 8251, thông thường đó sẽ là kí tự đồng bộ. Khi CTS tác động mức thấp kí tự đầu tiên sẽ được dịch nối tiếp tới ngõ ra. Các bit sẽ được dịch ra ngoài ở mỗi cạnh lên của TxC. Dữ liệu sẽ được dịch ra ngoài cùng tốc độ với tần số TxC.

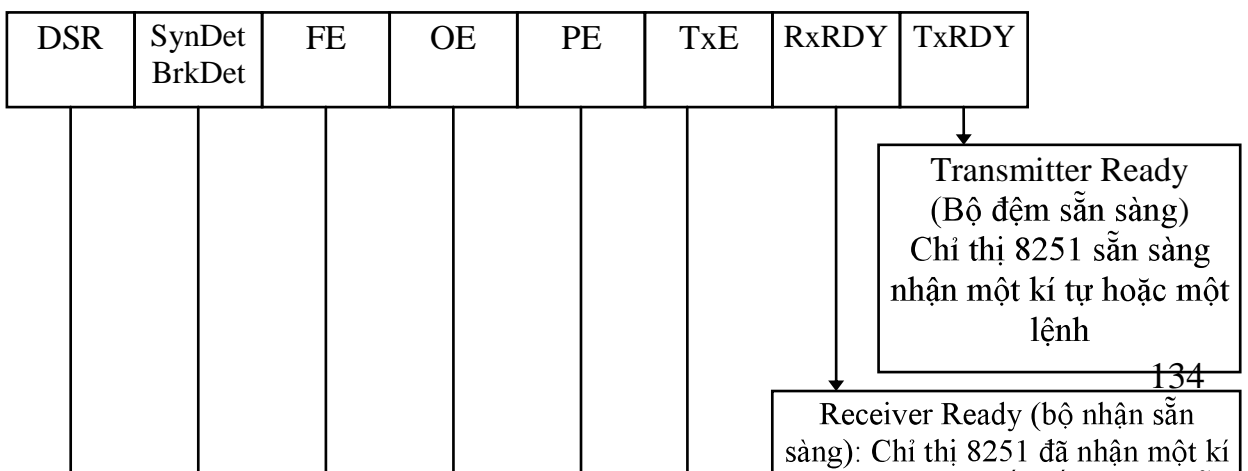
Mỗi lần bắt đầu truyền chuỗi dữ liệu ra cần giữ bằng tốc độ TxC một cách liên tục. Nếu CPU không ghi kí tự đồng bộ tới 8251 để nó truyền đi, thì bộ đếm truyền sẽ rỗng và các kí tự đồng bộ sẽ tự động được dịch ra trên đường TxD (các kí tự đồng bộ có thể bao gồm 1 hoặc hai kí tự tùy theo việc định nghĩa trong từ lệnh). Trong trường hợp này ngõ ra TxEMPTY sẽ tác động mức cao để thông báo cho bên nhận biết 8251 không truyền dữ liệu mà truyền từ đồng bộ. TxEMPTY sẽ không trở về mức thấp

trong khi 8251 truyền các kí tự đồng bộ, mà chỉ trở về mức 0 khi có một kí tự dữ liệu được ghi tới nó. Hình 5.28 mô tả khung truyền trong chế độ đồng bộ cùng với sự tác động của tín hiệu TxEMPTY:



Chế độ nhận đồng bộ Hình 5.28: Khung dữ liệu bên truyền trong chế độ đồng bộ.

Trong chế độ này có thể thực hiện đồng bộ trong hoặc đồng bộ ngoài tùy theo định nghĩa trong từ lệnh. Khi chọn chế độ đồng bộ trong từ chế độ thì phải lập bit EH lên 1 để cho phép chế độ tìm kiếm đồng bộ. Dữ liệu tới ngõ vào RxD sẽ được lấy mẫu tại mỗi cạnh lên của RxC. Giá trị trong bộ đệm nhận sẽ được so sánh tại mỗi thời điểm bắt đầu một bit cho đến khi tìm được kí tự đồng bộ. Nếu 8251 được lập trình có hai kí tự đồng bộ thì chuỗi dữ liệu nhận được sẽ được so sánh cho tới khi tìm được cả hai kí tự đó, lúc này USART sẽ kết thúc chế độ tìm kiếm đồng bộ (HUNT MODE) và bộ nhận đã đồng bộ được với bên truyền. Tín hiệu SYNDET sẽ tác động mức cao để thông báo cho bên truyền bắt đầu truyền dữ liệu, tín hiệu này sẽ tự động được xóa khi đọc từ trạng thái. Khi lập trình có kiểm tra chẵn lẻ thì SYNDET sẽ chưa được thiết lập cho đến điểm giữa của bit chẵn lẻ, thay vì tới điểm giữa của bit dữ liệu cuối cùng. Trong chế độ đồng bộ ngoài, việc đồng bộ với bên truyền được thực hiện bằng mức cao trên ngõ vào SYNDET, vì thế 8251 sẽ không thực hiện chế độ tìm kiếm từ đồng bộ. Mức cao này cần phải giữ trong thời gian lớn hơn một chu kỳ xung clock. Lập trình bit EH sẽ không có tác động gì trong chế độ cận đồng bộ. Các lỗi chẵn lẻ và overrun được thông báo và kiểm tra tương tự như trong chế độ cận đồng bộ. Lỗi chẵn lẻ luôn được kiểm tra trừ khi 8251 đang hoạt động trong chế độ tìm kiếm đồng bộ, mà không quan tâm đến việc bộ nhận có cho phép nó hay không. CPU có thể chuyển hoạt động của bộ nhận qua chế độ tìm kiếm khi mất đồng bộ. Điều này có thể thực hiện bằng cách ghi tới bộ đệm 1 kí tự có tất cả các bit bằng 1, như vậy sẽ tránh được việc phát hiện đồng bộ sai do nhầm kí tự đồng bộ với các kí tự dữ liệu thông thường khác.



Chú ý rằng SYNDET Flip Flop sẽ bị reset mỗi khi đọc từ trạng thái của 8251, mà không quan tâm tới việc nó được lập trình trong chế độ đồng bộ trong hay đồng bộ ngoài. Nhưng điều này không làm cho 8251 quay trở về chế độ tìm kiếm đồng bộ. Khi hoạt động trong chế độ đồng bộ, ngoài chế độ tìm kiếm đồng bộ, chức năng phát hiện đồng bộ vẫn được thực hiện tại biên các từ dữ liệu. Vì thế nếu hai phép đọc trạng thái liên tiếp đều phát hiện được đồng bộ thì các kí tự đồng bộ đã được phát hiện từ lần đọc trạng thái trước. Khi chọn chế độ đồng bộ ngoài, chế độ đồng bộ trong sẽ bị cấm, và SYNDET FF có thể thiết lập tại biên mỗi bit.

Từ trạng thái của 8251.

Từ trạng thái của 8251 bao gồm các bit với các ý nghĩa như hình 5.29:

Từ trạng thái có thể đọc vào bằng lệnh Input với địa chỉ là CS=0 và C/D=1. Trước khi truyền nhận dữ liệu để đảm bảo CPU cần phải kiểm tra xem 8251 đã ở trạng thái sẵn sàng chưa, việc kiểm tra có thể thực hiện bằng cách đọc giá trị của thanh ghi trạng thái và xem xét giá trị các bit của nó mỗi bit sẽ thông báo cho chúng ta biết về một đặc tính đã xảy ra tại 8251. Khi việc truyền nhận dữ đã thực hiện xong kí tự mà CPU nhận được cũng có thể sai do các tác nhân bên ngoài, các lỗi này cũng được thông báo trên

thanh ghi trạng thái, nhờ đó có thể thực hiện các giải thuật thích hợp để kiểm tra và truyền nhận lại dữ liệu mỗi khi xảy ra sai sót.

Để hiểu rõ về tác dụng của thanh ghi trạng thái có thể xem xét các đoạn chương trình truyền nhận dữ liệu bằng 8251 viết bằng tập lệnh của CPU 8086.

* **Đoạn lệnh truyền dữ liệu:**

```

MOV DX,Ctr_Add      ; Địa chỉ thanh ghi trạng thái
Test1: IN AL,DX      ; Đọc thanh ghi trạng thái
AND AL,10000001B    ; Kiểm tra DSR và TxRDY
CMP AL,10000001B
JNE Test1           ; Chỉ bắt đầu truyền khi đã sẵn sàng
MOV DX,Data_Add     ; Địa chỉ thanh ghi dữ liệu
MOV AL, Data_Send   ; Nạp dữ liệu muốn truyền vào AL
OUT DX,AL           ; Gửi dữ liệu tới cổng truyền
    
```

Trước khi muốn truyền một dữ liệu, cần phải kiểm tra xem bộ đệm truyền đã sẵn sàng truyền dữ liệu hay chưa (khi sẵn sàng bit D0 của thanh ghi trạng thái sẽ bằng 1). Ngoài ra có thể 8251 được nối tới modem để truyền dữ liệu trên đường điện thoại, nên phải xác định modem đã kết nối sẵn sàng chưa (D7=1). Khi các tín hiệu đã sẵn sàng lúc này có thể sử dụng lệnh Output để gửi dữ liệu tới 8251.

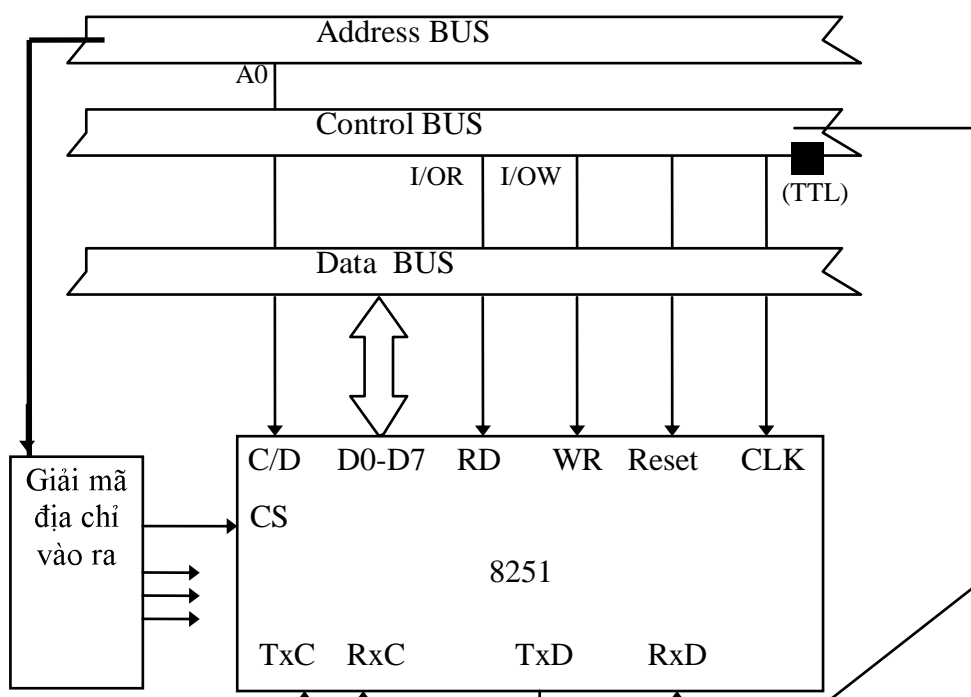
Đoạn lệnh nhận dữ liệu:

```

MOV DX,Ctr_Add      ; Địa chỉ thanh ghi trạng thái
Test2: IN AL,DX      ; Đọc thanh ghi trạng thái
AND AL,00000010B   ; Kiểm tra RxRDY
JZ Test2           ; Chỉ bắt đầu truyền khi đã sẵn sàng
MOV DX,Data_Add     ; Địa chỉ thanh ghi dữ liệu
IN AL,DX           ; Đọc dữ liệu
    
```

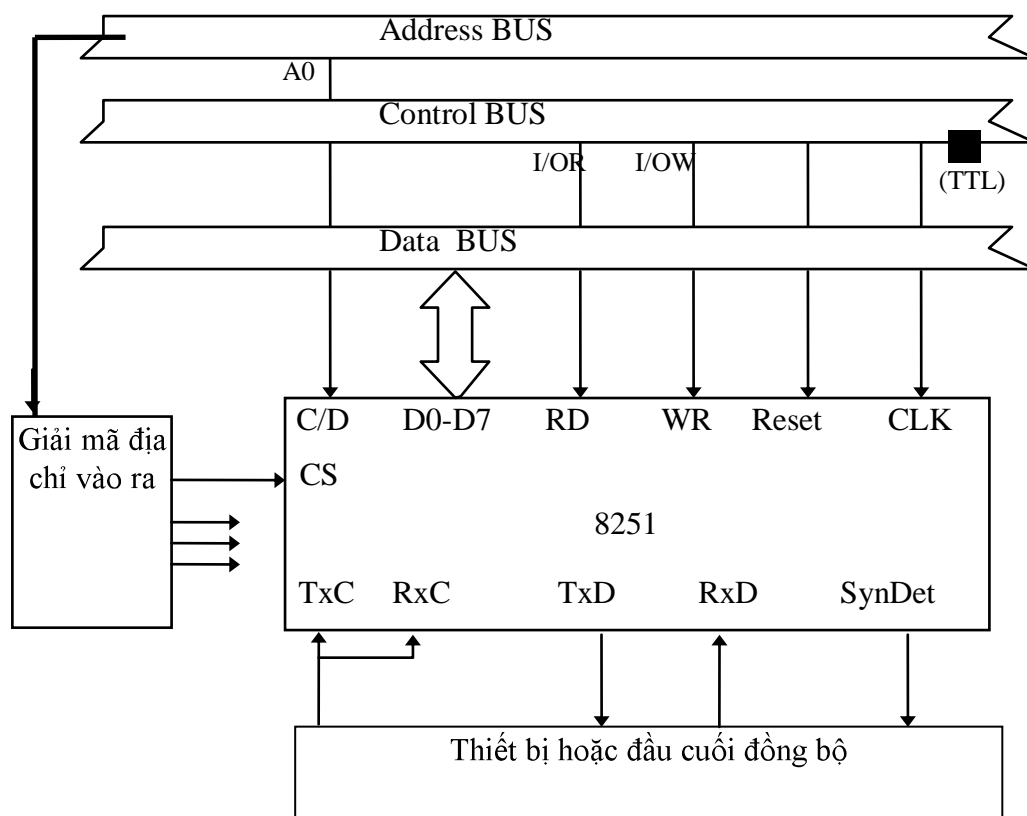
Trước tiên cần phải đọc giá trị của thanh ghi trạng thái bằng lệnh input với địa chỉ làm CS = 0; C/D=1. Sau đó kiểm tra trạng thái của bit RxRDY bằng lệnh AND 00000010B, lúc này nếu bit RxRDY (D1) = 0 thì sau lệnh AND kết quả trong thanh ghi AL sẽ bằng 0, và cờ Zero sẽ được lập làm chuyển điều khiển về nhãn Test2 để kiểm tra lại một lần nữa. Khi bit RxRDY lên mức 1, có nghĩa là trong bộ đệm của bộ nhận đã chứa một kí tự, lúc này có thể đọc kí tự vào bằng lệnh IN với địa chỉ làm CS=0 và C/D=0.

6. Một số sơ đồ ứng dụng 8251.



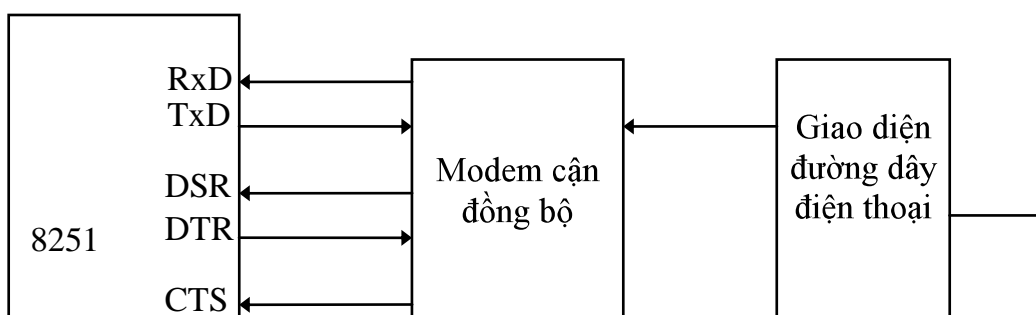
Sơ đồ hình 5.30 biểu diễn giao diện nối tiếp cận đồng bộ với thiết bị hiển thị CRT, tốc độ truyền nhận được tạo ra bằng một bộ phát xung nhịp. Việc truyền nhận dữ liệu thông qua hai đường TxD và RxD thông qua bộ đổi chuẩn tín hiệu.

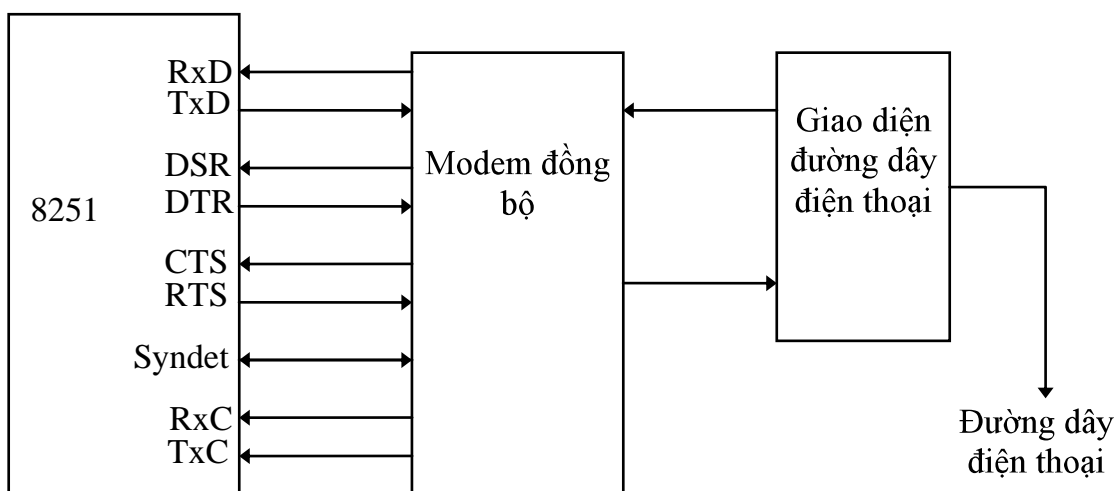
Sơ đồ hình 5.31 biểu diễn việc truyền dữ liệu theo chế độ đồng bộ bằng 8251, trong sơ đồ này cần thêm tín hiệu SYNDDET để thông báo việc đồng bộ giữa 8251 với thiết bị.



Hình 5.31: Giao tiếp giữa 8251 và thiết bị đồng bộ.

Khi truyền dữ liệu trên đường điện thoại 8251 có thể giao tiếp với các loại modem đồng bộ hoặc không đồng bộ theo các sơ đồ hình 5.32:





Hình 5.32: Giao tiếp nối tiếp qua đường điện thoại.

5.4.3. Nối ghép tương tự. (Analog interface)

Để điều khiển các máy móc trong công nghiệp, các thiết bị đo lường, các thiết bị tự động ... bằng hệ thống vi xử lý, cần phải xác định giá trị của rất nhiều đại lượng khác nhau như: áp suất, nhiệt độ, lưu lượng, số vòng quay Quá trình thu thập các đại lượng này thường được tiến hành qua nhiều bước, các tín hiệu phi điện trước hết được biến đổi thành các tín hiệu điện tương ứng với các giá trị của chúng, sau đó các tín hiệu điện sẽ được biến đổi thành dạng số để có thể đọc vào bằng vi xử lý. Bước đầu tiên của quá trình này thực hiện bằng các bộ cảm biến, chúng sẽ biến đổi các đại lượng vật lý phi điện thành dạng dòng điện hoặc điện áp. Tín hiệu điện cung cấp từ các bộ cảm biến thông thường có giá trị nhỏ, vì thế bước kế tiếp chúng cần được lọc và khuếch đại (thông thường bằng các bộ khuếch đại thuật toán). Bước cuối cùng tín hiệu dòng điện và điện áp dưới dạng tương tự sẽ được biến đổi thành tín hiệu số bằng các bộ ADC.

Ngoài ra trong thực tế, rất nhiều các thiết bị không thể điều khiển tắt mở bằng tín hiệu số cung cấp từ vi xử lý. Ví dụ tốc độ quay của một động cơ DC cần điều khiển bằng mức điện áp tương tự, âm thanh lưu trữ trong bộ nhớ dưới dạng số để nghe được cần phát ra dưới dạng tín hiệu tương tự....

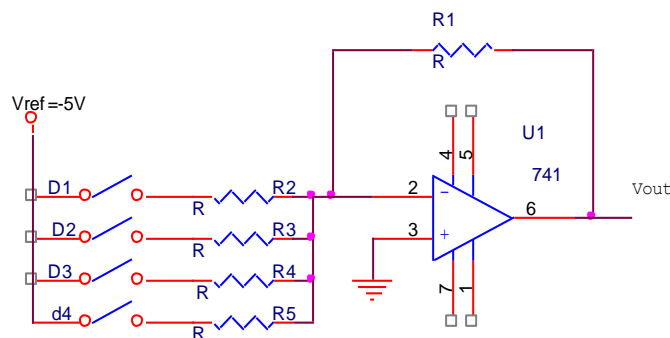
Trong phần này sẽ giới thiệu về hoạt động và nguyên tắc kết nối giữa các bộ biến đổi A/D và D/A với hệ thống vi xử lý.

5.4.3.1. Nối ghép DAC.

■ Nguyên lý hoạt động của DAC.

Bộ biến đổi DA có chức năng biến đổi dữ liệu dạng nhị phân thành dạng dòng điện hoặc điện áp liên tục. Có thể xem xét hoạt động của các bộ DAC thông qua mạch cộng điện áp 4 ngõ vào trên hình 5.33.

Vì ngõ vào không đảo của Op Amp được nối masse, nên khi hoạt động ngõ vào đảo của nó luôn được giữ mức 0 V. Trong hình vẽ ngõ vào đảo là một điểm tổng, khi một trong các công tắc đóng, dòng điện sẽ cung cấp tới ngõ vào từ nguồn $-5V$ thông qua các điện trở nối vào điểm tổng. Dòng điện ngõ vào này sẽ được khuếch đại tạo thành điện áp ngõ ra. Ví dụ khi công tắc D0 đóng, dòng điện $0,05 \text{ mA}$ sẽ đưa tới điểm tổng, qua điện trở hồi tiếp ngõ ra sẽ có điện áp là $0,05 \text{ mA} \times 10K = 0,5 \text{ V}$. Nếu đóng thêm công tắc D1, dòng tới điểm tổng sẽ là $0,15 \text{ mA}$ và điện áp ra đạt $1,5 \text{ V}$. Các điện trở nối tới điểm tổng có độ lớn giảm dần đi $\frac{1}{2}$ làm dòng điện qua nhánh dưới sẽ gấp đôi nhánh trên khi công tắc được đóng làm điện áp ngõ ra của Op Amp cũng thay đổi tương ứng. Khi thay các công tắc bằng các bit nhị phân cung cấp từ vi xử lý, ngõ ra của mạch sẽ có 16 mức điện áp tương tự tương ứng với giá trị số cung cấp, bit cung cấp tới ngõ D3 sẽ là bit có trọng số lớn nhất. Thông thường các linh kiện của bộ biến đổi DA được tích hợp trong một vi mạch, dòng điện theo kiểu nhị phân sẽ được tạo ra bằng nhiều cách khác nhau, mạch Op Amp sẽ biến đổi tổng các dòng điện thành điện áp ngõ ra.



Hình 5.33: Bộ biến đổi D/A đơn giản 4 bit.

■ Các đặc tính và các thông số kỹ thuật của DAC.

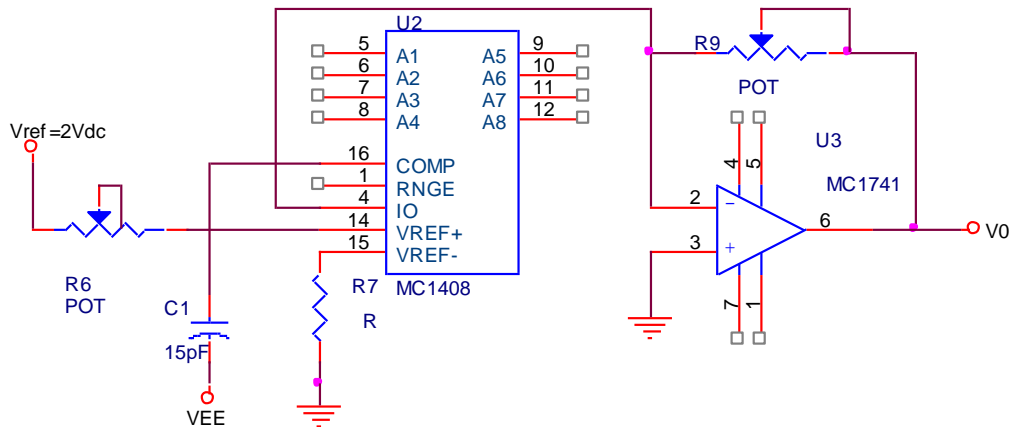
Hình 5.34 biểu diễn một bộ biến đổi D/A sử dụng IC họ MC 1408, khi sử dụng các vi mạch tích hợp này cần quan tâm tới các chỉ tiêu kỹ thuật của chúng.

Thông số kỹ thuật đầu tiên của các bộ DAC cần quan tâm là độ phân giải (resolution). Thông số này xác định số bit nhị phân ngõ vào, một ADC có 8 bit ngõ vào sẽ có $2^8 = 256$ mức điện áp ngõ ra. Độ phân giải có thể được biểu diễn theo phần trăm, ví dụ DAC 8 bit có độ phân giải biểu diễn theo phần trăm sẽ là: $(1/256) \times 100 = 0.39\%$.

Thông số tiếp theo của DAC điện áp ra đầy thang (full – scale). Ví dụ với mạch hình 5.34, dòng điện cung cấp tới tất cả các công tắc đều từ nguồn V_{ref} thông qua điện trở R14. Dòng điện ngõ ra DAC sẽ chạy qua điện trở R0 để tạo thành điện áp ngõ ra. Với công thức tính điện áp ngõ ra, nếu tất cả các ngõ vào của DAC đều bằng 1, thì điện áp ngõ ra sẽ là $10V \times (255/256) = 9,961 \text{ V}$. Mặc dù ngõ ra của DAC không bao giờ đạt tới 10 V , nhưng chúng được coi là bộ biến đổi với ngõ ra cực đại là 10 V . Như vậy giá

trị cực đại của một bộ DAC luôn có một bit có trọng số nhỏ nhất có giá trị nhỏ hơn chỉ số tên của nó. Ví dụ với DAC 12 bit biến đổi 10 V thì giá trị của 1 LSB sẽ là $10V/4096 = 2,44 \text{ mV}$. Điện áp cực đại ngõ ra sẽ là $10V - 0,0024 = 9,9976 \text{ V}$.

Độ chính xác (accuracy) của một DAC là thông số so sánh giữa điện áp ngõ ra thực sự và điện áp ngõ ra mong muốn theo lý thuyết, nó thường được biểu diễn dưới dạng phần trăm của điện áp hoặc dòng điện ra đầy thang. Ví dụ một bộ DAC có điện áp ra đầy thang là 10V với độ chính xác $\pm 0,2\%$ thì sai số cực đại cho một giá trị ra bất kỳ sẽ là $0,002 \times 10V = 20 \text{ mV}$. Sai số cực đại của một bộ DAC phải luôn đảm bảo không lớn hơn $\pm 0,5$ giá trị LSB của nó.



Hình 5.34: Biến đổi D/A MC1408 với mạch biến đổi dòng sang áp

Một thông số quan trọng khác của DAC là độ tuyến tính, nó là tham số đo độ sai lệch của ngõ ra so với đường thẳng nghiêng nối giữa điểm tương ứng với trạng thái ngõ ra khi không có công tắc nào đóng, và điểm ứng với trạng thái tất cả các công tắc đều đóng. Độ sai lệch này cần không được lớn hơn $\pm 0,5$ giá trị LSB để duy trì độ chính xác trong toàn giải biến đổi. Tuy nhiên rất nhiều DAC trong thực tế có sai số tuyến tính lớn hơn giá trị đó. Như vậy độ tuyến tính còn là một tham số quan trọng quyết định giá thành của DAC, nếu sai lệch tuyến tính càng lớn thì sự sai khác giữa tín hiệu mong muốn và tín hiệu thực tế tạo ra càng rộng.

Một tham số khác cần quan tâm của DAC là thời gian thiết lập (setting time), thông thường khi biến đổi một từ nhị phân ngõ vào thành điện áp ngõ ra tương ứng. Ngõ ra của DAC có thể vượt quá giá trị đúng ngõ ra, sau đó mới ổn định dần về giá trị đúng của nó. Thời gian để ngõ ra đạt được giá trị trong khoảng $\pm 0,5$ LSB so với giá trị đúng được gọi là thời gian thiết lập. Tham số này hết sức quan trọng khi sử dụng các DAC ở tần số cao.

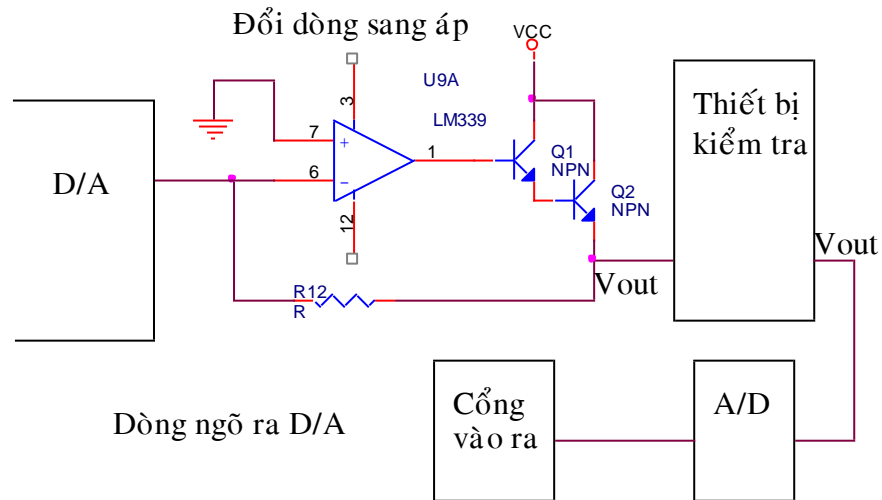
■ Các ứng dụng và giao tiếp giữa DAC và hệ thống vi xử lý.

Các DAC có rất nhiều ứng dụng khác ngoài các ứng dụng trong các hệ thống vi xử lý. Ví dụ trong một máy nhạc CD, một bộ DAC 14 hoặc 16 bit sẽ biến đổi dữ liệu nhị phân đọc từ đĩa bằng đầu đọc laser thành tín hiệu âm thanh tương tự phát ra loa. Hầu hết các vi mạch tổ hợp âm thanh đều có bộ DAC để biến đổi dữ liệu nhị phân trong bộ nhớ thành tín hiệu tương tự.

Để giao tiếp DAC với hệ thống vi xử lý, trước hết cần kết nối các ngõ vào của nó (A1 – A8 trên hình 5.34) tới Bus dữ liệu của hệ thống vi xử lý thông qua cổng vào ra. Và bằng các lệnh của chương trình có thể tạo ra điện áp ngõ ra như mong muốn. Với giao

tiếp DAC có thể thực hiện được các hệ thống điều khiển tương tự hoạt động theo chương trình.

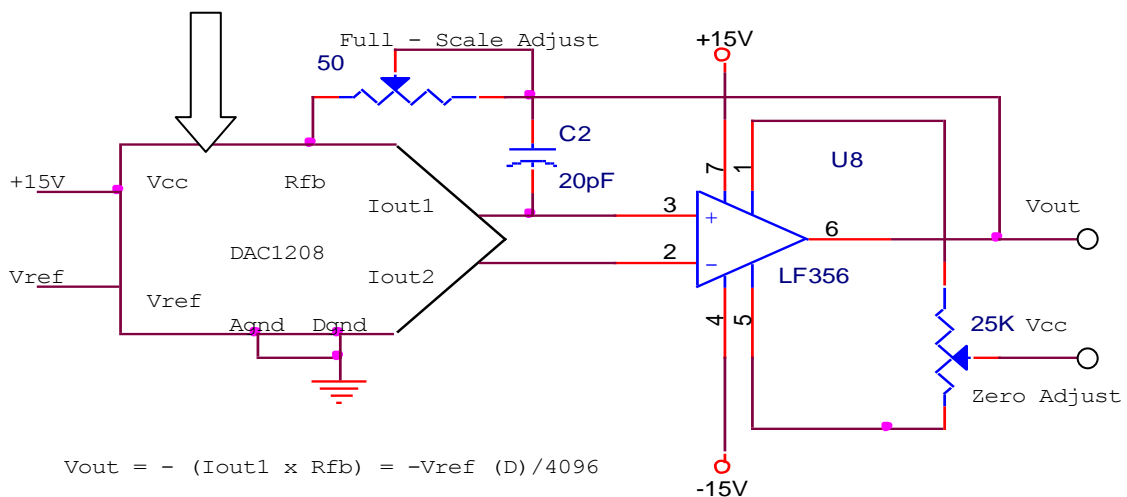
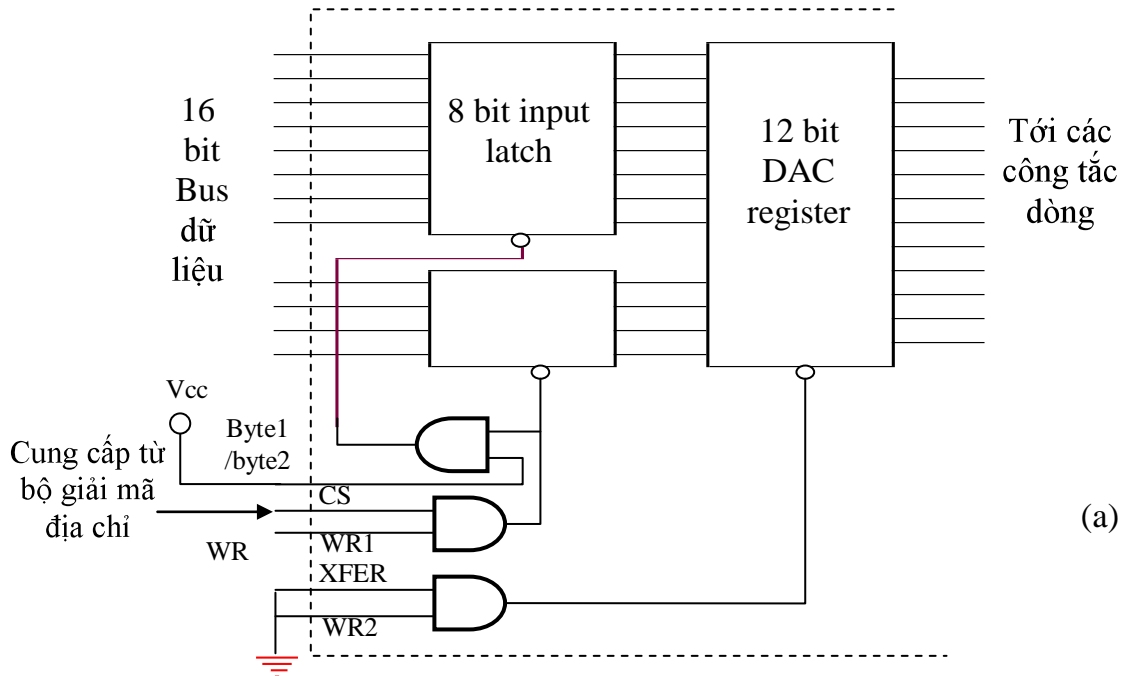
Ví dụ có thể thực hiện một hệ thống kiểm tra ảnh hưởng của điện áp nguồn cung cấp, tới điện áp ngõ ra của một số bộ khuếch đại hoạt động theo chương trình. Nếu nối ngõ ra của DAC tới ngõ vào chuẩn của một bộ nguồn lập trình, hoặc đơn giản tới một mạch đệm dòng cao như trên hình 5.35, thì có thể thay đổi được điện áp nguồn cung cấp theo chương trình. Để xác định điện áp ngõ ra của một vi mạch cần thử nghiệm, chúng ta thay đổi điện áp cung cấp cho nó và nối ngõ ra của nó tới một bộ ADC, ngõ ra của ADC sẽ được nối tới một công ngõ vào của hệ thống vi xử lý, khi đó có thể đọc được giá trị điện áp ngõ ra của vi mạch.



Một ví dụ khác, có thể thay đổi dòng điện ngõ ra của DAC từ ngõ ra của DAC thông qua bộ đệm dòng, khi đó có thể thay đổi nhiệt độ trong một buồng sấy theo chương trình. Cũng có thể điều khiển tốc độ của một động cơ DC nhỏ bằng chương trình bằng cách thay đổi dòng điện qua nó. Khi nối động cơ tới ngõ ra của một bộ DAC thông qua mạch đệm dòng, tốc độ của động cơ sẽ thay đổi tùy thuộc vào giá trị nạp cho DAC. Khi dùng điều khiển có phản hồi có thể duy trì tốc độ ổn định cho động cơ khi tải thay đổi.

Khi sử dụng một DAC 8 bit trong một hệ thống vi xử lý, một cách đơn giản có thể nối các ngõ vào của chúng tới một công ngõ ra, hoặc một số bộ DAC còn cho phép nối trực tiếp các ngõ vào của chúng tới Bus dữ liệu như đối với một công ngõ ra thông thường. Còn đối với các ứng dụng yêu cầu độ phân giải 12 bit, cần phải sử dụng các DAC 12 bit. Nhưng nếu hệ thống sử dụng vi xử lý 8 bit, trước hết phải nối 8 bit thấp của DAC tới một công ngõ vào ra, còn 4 bit cao của nó phải nối tới một công ngõ vào ra khác. Trong chương trình cần phải chuyển 8 bit thấp tới bằng một lệnh ghi, vào 4 bit cao bằng một lệnh ghi khác. Nhưng khi sử dụng hai lần ghi như vậy, ngõ ra sẽ bị đột biến, trước khi đạt được giá trị ổn định như mong muốn. Ví dụ khi muốn biến đổi giá trị của 1 DAC 12 bit từ 0000 1111 1111 thành 0001 0000 0000. Khi ghi 8 bit thấp tới DAC, ngõ ra của nó sẽ thay đổi giá trị từ 0000 1111 1111 thành 0000 0000 0000, sau đó ghi tới 4 bit cao ngõ ra với trở về giá trị mong muốn là 0001 0000 0000. Như vậy trong khoảng thời gian giữa hai lần ghi sẽ là giá trị không mong muốn. Trong nhiều hệ thống vấn đề trên sẽ gây ra hoạt động không ổn định. Để tránh được nhược điểm này có thể sử dụng các bộ cài tại ngõ vào DAC. Bộ cài sẽ được nạp giá trị hai lần sau đó chuyển

đồng thời 12 bit tới DAC. Rất nhiều bộ DAC hiện nay cho phép thực hiện công việc trên một cách dễ dàng. Hình 5.36a trình bày cấu trúc ngõ vào của DAC1230 và DAC08 của hãng national, chúng có bộ cài 4 bit thấp vì thế có thể ghi tới 12 bit ngõ vào bằng hai lệnh ghi từ các cổng vào ra 8 bit, hoặc trực tiếp từ Bus dữ liệu. Cũng có thể nối các ngõ vào của DAC này tới các đường dữ liệu của các hệ thống 16 bit như trên hình 5.36a.



$$V_{out} = - (I_{out1} \times R_{fb}) = -V_{ref} (D) / 4096$$

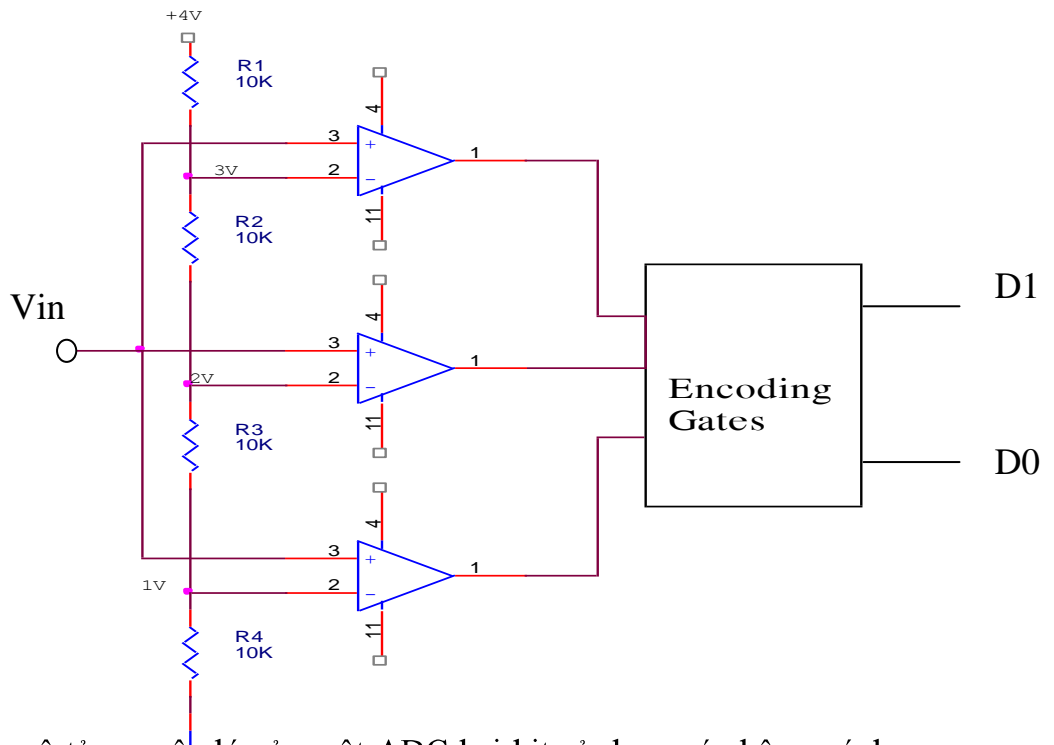
Các bộ DAC yêu cầu điện áp chuẩn chính xác, mạch hình 5.36b sử dụng điện áp chuẩn - 10,000V. Các DAC (a) cung cấp dòng ngõ ra, vì vậy cần nối thêm một mạch Op Amp ở ngõ ra để biến đổi thành điện áp ngõ ra. Thông thường sử dụng các bộ khuếch đại có ngõ vào FET, vì dòng phân cực ngõ vào của các bộ khuếch đại lưỡng cực có thể ảnh hưởng tới độ chính xác của ngõ ra. Chú ý trong mạch sử dụng masse số và masse tương tự riêng nhau, để tránh nhiễu cho tín hiệu số gây ra bởi sự méo dạng của tín hiệu tương tự.

5.4.3.2. Ghép nối ADC.

Chức năng của bộ biến đổi AD là tạo ra chuỗi số nhị phân tương ứng với độ lớn dòng điện hoặc điện áp ở ngõ vào của nó. Các thông số kỹ thuật cần quan tâm của ADC rất giống với DAC. Độ phân giải của một ADC là tham số biểu diễn số bit nhị phân ngõ ra của nó, ví dụ một ADC 8 bit sẽ có độ phân giải là $1/256$. Độ chính xác và độ tuyến tính cũng có cùng ý nghĩa như trong DAC. Một thông số quan trọng trong ADC là thời gian biến đổi, nó tính từ khi có giá trị điện áp ngõ vào ổn định cho tới khi có giá trị ngõ ra số hợp lệ. Một ADC tốc độ cao là bộ ADC có thời gian biến đổi ngắn. Có rất nhiều kỹ thuật khác nhau để thực hiện việc biến đổi ADC, sau đây sẽ xem xét một số kỹ thuật cơ bản nhất.

■ Các loại ADC.

ADC so sánh song song.

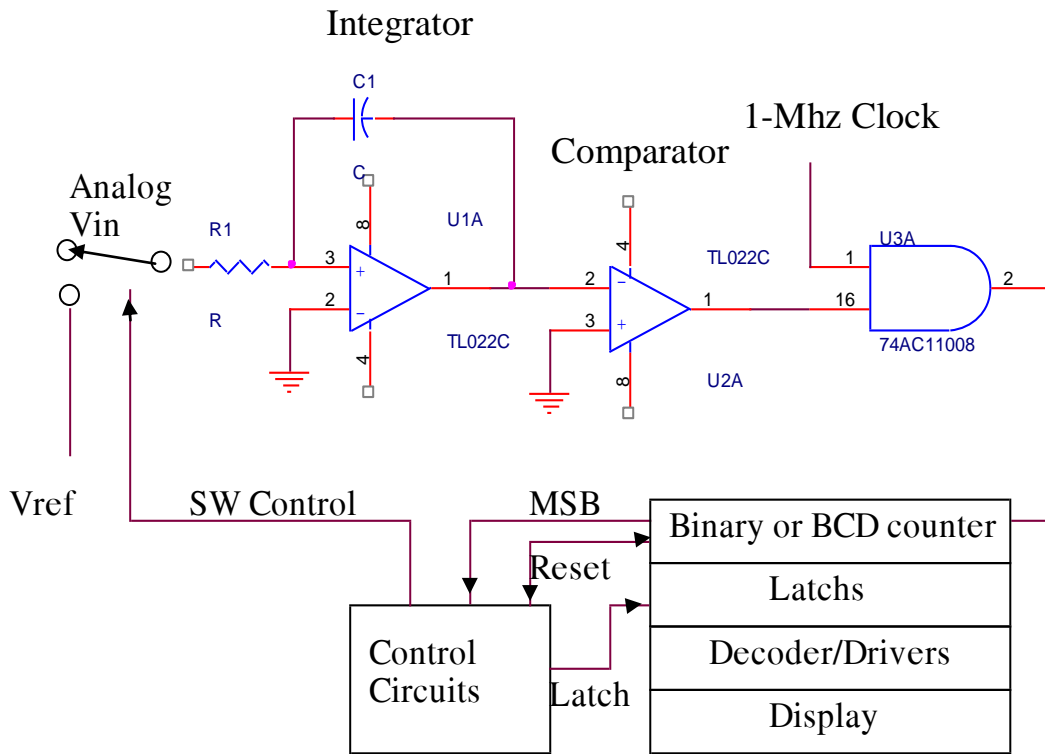


Hình 5.37 mô tả nguyên lý của một ADC hai bit sử dụng các bộ so sánh song song. Một mạch chia điện áp như Hình 5.37 sẽ biến đổi chuẩn cung cấp tới các ngõ vào đảo của các bộ so sánh. Điện áp cung cấp cho cầu phân áp này chính là trị số đầy thang của ADC. Điện áp cần biến đổi được cấp tới ngõ vào không đảo của các bộ so sánh. Nếu điện áp ngõ vào lớn hơn điện áp ngõ vào đảo của bộ so sánh nào, thì ngõ ra của nó sẽ được đưa lên mức cao. Ngõ ra của các bộ so sánh sẽ cho ra các bit số tương ứng với độ lớn của điện áp ngõ vào. Ví dụ với ngõ vào 2,6V sẽ có $A3A2A1 = 011$.

Lợi điểm của các bộ ADC so sánh song song là có tốc độ biến đổi cao, nó chỉ là thời gian trễ lan truyền của các bộ so sánh. Ngõ ra của các bộ so sánh không phải là mức logic chuẩn, nhưng chúng có thể dễ dàng biến đổi thành các mức logic thích hợp. Bất lợi chính của các ADC so sánh song song, là cần sử dụng nhiều số bộ so sánh, khi cần biến đổi N bit sẽ phải sử dụng 2^{N-1} bộ so sánh. Với ADC 10 bit sẽ cần 1023 bộ so sánh, vì vậy các bộ ADC loại này có giá thành rất cao, chúng chỉ sử dụng cho các ứng dụng có yêu cầu tốc độ biến đổi cao.

ADC hai độ dốc.

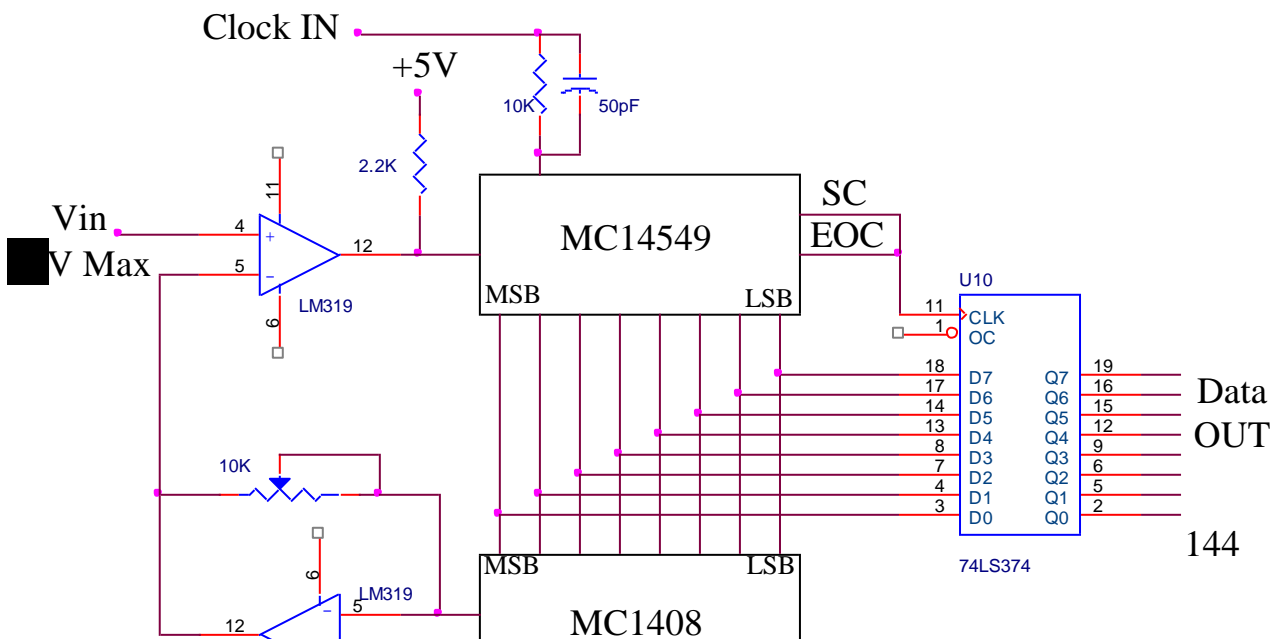
Hình 5.38 trình bày sơ đồ các khối chức năng của một bộ ADC hai độ dốc. Loại ADC này thường sử dụng trong các volt kế số vì có số bit phân giải cao với giá thành hạ.



Khi bắt đầu biến đổi mạch điều khiển sẽ xoá bộ đếm về 0 và nối ngõ vào của mạch tích phân tới tín hiệu điện áp cần biến đổi. Khi điện áp cần biến đổi dương, ngõ ra mạch tích phân sẽ là một điện áp âm. Khi ngõ ra mạch so sánh có điện áp âm khoảng vài microvolt, ngõ ra bộ so sánh vẫn ở mức cao, lúc này cổng AND cho phép xung 1 Mhz đưa vào bộ đếm. Sau một giá trị đếm xác định nào đó (thường là 1000), mạch điều khiển sẽ chuyển ngõ vào của mạch tích phân tới điện áp chuẩn âm và xoá bộ đếm. Với điện áp ngõ vào âm, điện áp ngõ ra của mạch tích phân sẽ tăng dần về 0. Khi ngõ ra mạch tích phân đạt tới 0 volt, ngõ ra mạch so sánh sẽ chuyển trạng thái về mức thấp cắt xung clock cung cấp tới bộ đếm. Giá trị đếm được trong bộ đếm cho tới khi ngõ ra mạch tích phân đạt tới 0 volt sẽ tương ứng với giá trị điện áp vào.

ADC xấp xỉ liên tiếp.

Hình 5.39 mô tả cấu tạo của một ADC xấp xỉ liên tiếp, với khối quan trọng nhất là thanh ghi xấp xỉ liên tiếp.



Tại Clock đầu tiên trong mỗi chu kỳ biến đổi, MSB của SAR sẽ ở mức cao cung cấp tới bộ biến đổi DAC. Tín hiệu dòng điện tương ứng ở ngõ ra DAC sẽ được khuếch đại và biến đổi thành tín hiệu điện áp cung cấp tới ngõ vào của bộ so sánh. Nếu điện áp này lớn hơn điện áp cần biến đổi đưa tới ngõ vào còn lại của bộ so sánh, thì ngõ ra bộ so sánh sẽ xuống mức thấp thông báo cho SAR xóa bit này, ngược lại bit này sẽ được giữ. ở chu kỳ clock kế tiếp SAR tiếp tục tác động bit có trọng số kế tiếp, và tương tự như trên, tùy theo trạng thái ngõ ra bộ so sánh mà SAR sẽ quyết định giữ lại hoặc xóa bit này. Quá trình tiếp tục được thực hiện cho tới bit cuối cùng (LSB), với ADC 8 bit quá trình biến đổi sẽ được thực hiện trong 9 xung clock. Khi quá trình biến đổi kết thúc, SAR sẽ tạo xung EOC để cài giá trị biến đổi được cung cấp ra ngoài. Nếu EOC được nối về tín hiệu bắt đầu biến đổi SC (Start Converter), thì một chu trình biến đổi tiếp theo sẽ được bắt đầu.

■ **Giao tiếp ADC với hệ thống vi xử lý.**

Giao tiếp với ADC so sánh song song.

Trong hầu hết các ứng dụng sử dụng các bộ ADC biến đổi song song, các bộ ADC này sẽ cung cấp dữ liệu ngõ ra thông thường nhanh hơn tốc độ đọc của vi xử lý. Vì vậy hoàn toàn có thể sử dụng một mạch thâm nhập trực tiếp bộ nhớ để nạp trực tiếp các dữ liệu của ADC vào bộ nhớ mà không cần sự điều khiển của vi xử lý. Bộ vi xử lý sau đó có thể xử lý các dữ liệu đã được lấy vào bộ nhớ.

Giao tiếp với ADC hai độ dốc.

Với các bộ ADC hai độ dốc, thời gian biến đổi là khá chậm so với tốc độ hoạt động của CPU. Để đọc được dữ liệu đúng, cần kiểm tra trạng thái của tín hiệu chốt dữ liệu cung cấp từ mạch điều khiển trong ADC. Khi tín hiệu này tác động mức cao, vi xử lý có thể đọc được một mã dữ liệu để ghi vào bộ nhớ.

Giao tiếp với ADC xấp xỉ liên tiếp.

Đọc dữ liệu trong các bộ ADC xấp xỉ liên tiếp cần thực hiện theo sự điều khiển của chương trình. Trước hết cần cung cấp tín hiệu khởi động quá trình biến đổi SC (Start Converter) từ vi xử lý tới ADC. Tiếp theo kiểm tra ngõ ra EOC của ADC để biết được thời điểm quá trình biến đổi kết thúc. Khi EOC tác động, vi xử lý có thể đọc được một dữ liệu hợp lệ cung cấp từ ADC.

Với các bộ ADC có nhiều kênh vào, thông thường chúng sẽ có các đường địa chỉ và tín hiệu cài địa chỉ để chọn kênh. Công việc này được thực hiện dễ dàng bằng phần mềm.

5.4.4. Một số nối ghép khác.

1. Ghép nối với bàn phím.

Bàn phím là một thiết bị vào ra cơ bản của một hệ thống vi xử lý, nó cho phép người sử dụng có thể thiết lập các thông số và lựa chọn các chế độ hoạt động khác nhau của hệ thống. Khi nhấn một phím, sẽ tác động một công tắc, có nhiều cách khác nhau để thực hiện các công tắc này. Sau đây chúng ta sẽ xem xét một số phương pháp thông dụng nhất.

a) Phím cơ khí.

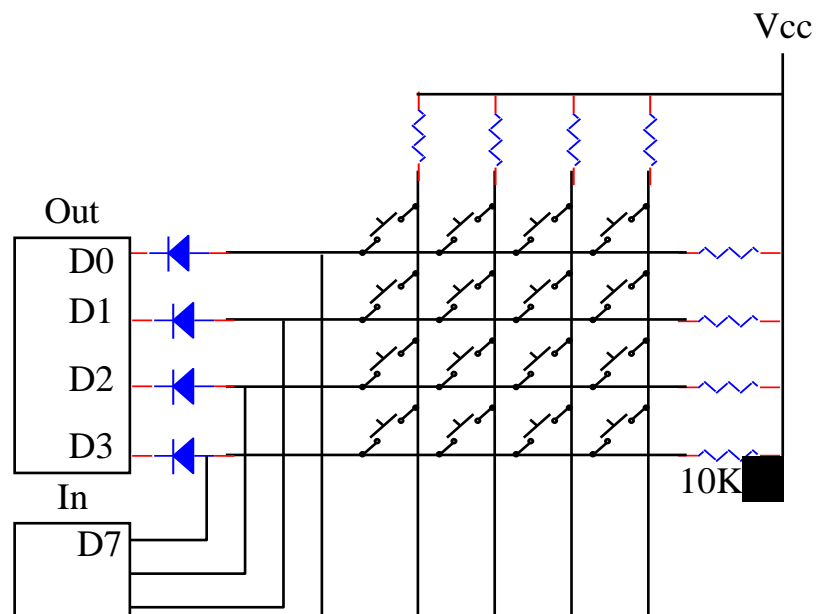
Các phím nhấn cơ khí có cấu tạo bằng một tiếp xúc của hai thanh kim loại, tiếp xúc này sẽ được dính lại khi nhấn phím. Thông thường các phím nhấn có thêm các lò xo để chúng có thể trở về trạng thái ban đầu khi không nhấn nữa. Có một số bàn phím được cấu tạo là các lớp dẫn điện tráng trên bề mặt hai miếng nhựa mỏng, và trạng thái đàn hồi thực hiện bằng các núm cao su. Các phím nhấn cơ khí còn có thể thực hiện bằng nhiều phương pháp khác nữa, nhưng nói chung chúng có giá thành rẻ nên được sử dụng khá phổ biến. Tuy nhiên chúng có một số các nhược điểm như: chúng thường có tác động dội, khi nhấn công tắc sẽ chập nhả nhiều lần trước khi thực sự được chập lại thực sự. Ngoài ra các công tắc cơ khí này thường hay bị oxy hoá hay bị dơ bẩn sau thời gian sử dụng.

b) Phím nhấn điện dung.

Phím nhấn điện dung bao gồm hai bản cực nhỏ, một bản được cố định trên mạch, bản kia nằm trên nút nhấn. Khi tác động nhấn phím, khoảng cách giữa hai bản cực thay đổi, làm điện dung của tụ thay đổi. Một mạch khuếch đại sẽ phát hiện sự thay đổi của điện dung tụ để biến đổi thành logic thích hợp chỉ thị trạng thái nhấn phím. Phím nhấn tụ có lợi điểm rất lớn là không bị oxy hoá hoặc dơ bẩn, nhưng nó cũng có nhược điểm là cần phải có một mạch điện để phát hiện sự thay đổi của điện dung tụ.

c) Phím nhấn hiệu ứng Hall.

Đây là một loại phím nhấn khác không có tiếp xúc cơ khí. Phím nhấn loại này được chế tạo bằng một tinh thể thạch anh, đặt phía trên một từ trường với các đường sức vuông góc với dòng điện chuẩn đi qua nó. Khi phím được nhấn, tinh thể thạch anh được di chuyển vào vùng đường sức từ trường, làm xuất hiện một điện áp ở hai bề mặt đối diện còn lại của tinh thể. Điện áp này được khuếch đại để chỉ thị tác động nhấn phím. Phím nhấn loại này có giá thành rất cao do có cấu tạo phức tạp, nhưng có thời gian sử dụng dài.

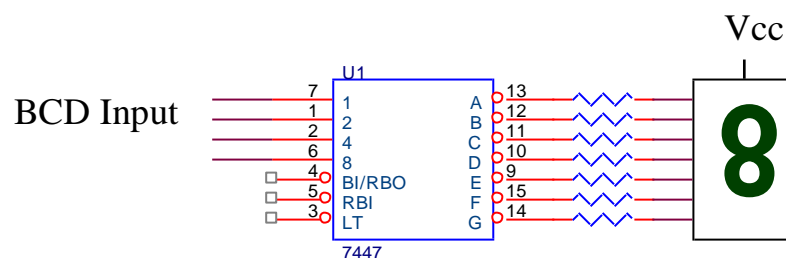


Ở hầu hết các bàn phím, thông thường các phím nhấn được kết nối thành dạng ma trận theo các hàng và cột như biểu diễn trên hình 5.40. Khi muốn kiểm tra có phím nhấn, trước tiên cung cấp mức 0 tới tất cả các hàng rồi đọc trạng thái của tất cả các cột trên bàn phím. Nếu tất cả các cột đều ở trạng thái 1, tức là không có phím nào được nhấn. Khi có một cột ở trạng thái 0, nếu là phím nhấn cơ khí có thể tạo trễ khoảng 20 msec để đọc lại trạng thái các cột để loại bỏ trạng thái dội của các phím. Khi đã xác nhận đúng trạng thái nhấn phím, có thể cung cấp mức 0 ra từng hàng rồi lần lượt đọc các cột cho tới khi gặp được mã có một bit bằng 0. Ứng với tổ hợp mã hàng và cột khi phát hiện bit 0 này có thể xác định được vị trí phím nhấn. Tổ hợp các mã này có thể chuyển đổi sang một dạng mã khác thích hợp hơn cho việc xử lý trong chương trình.

Để xác định không có phím nào hư hỏng bị dính, có thể thực hiện đoạn chương trình cấp mức 0 ra tất cả mọi hàng rồi đọc vào các cột khi khởi động hệ thống (chưa có phím nào được nhấn), và thông báo trạng thái hư hỏng khi phát hiện mức 0 ở mã hàng.

2. Giao tiếp với các bộ hiển thị cơ bản.

a) Giao tiếp với hiển thị led 7 đoạn.



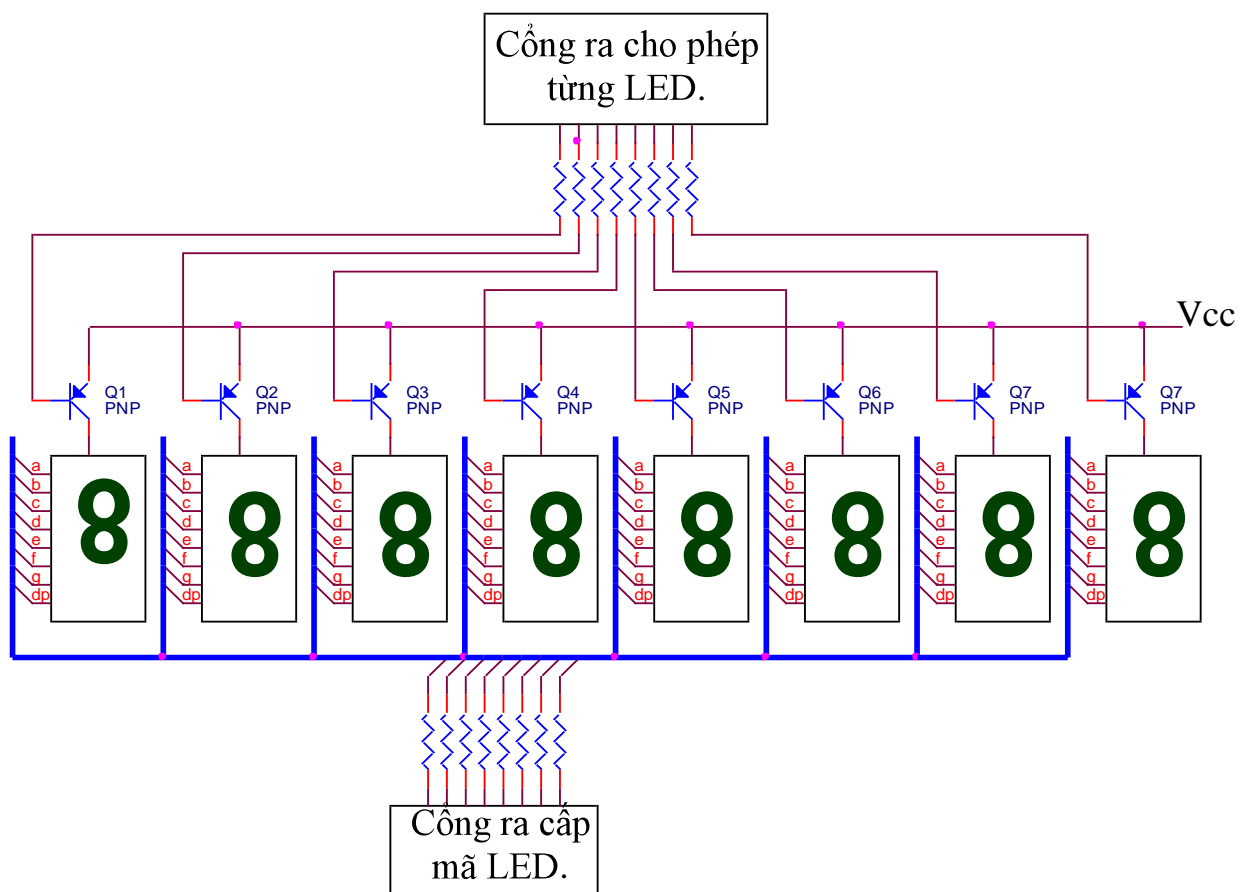
Hình 5.41: Bộ hiển thị sử dụng bằng mạch lái LED 7 đoạn.

Các bộ hiển thị led 7 đoạn là loại giao tiếp hiển thị được sử dụng phổ biến trong các hệ thống vi xử lý vừa và nhỏ, chúng rất tiện dụng trong việc hiển thị các con số cho phép xác định được các thông số mà người sử dụng muốn thiết lập cho hệ thống hoạt động.

Việc điều khiển hiển thị các con số có thể thực hiện bằng cách cung cấp mã BCD qua một cổng ra, tới bộ giải mã led 7 đoạn như trình bày trên hình 5.41. Khi sử dụng nhiều LED có thể sử dụng nhiều cổng ra cài dữ liệu tương tự.

Khi muốn hiển thị cả dấu phân cách thập phân và một số ký tự với cách hiển thị qui định trước, có thể cung cấp các mã tương ứng cho LED thẳng từ một công ra của vi xử lý. Để tiết kiệm số lượng công ra, có thể sử dụng kiểu điều khiển quét với mạch phần cứng như hình 5.42.

Mã hiển thị trên các đượ cấp chung để đưa tới tất cả mọi LED, muốn LED nào chỉ thị giá trị cung cấp ở trên chỉ cần cung cấp bit 1 để mở transistor tương ứng cấp điện vào Kathode chung. Để tất cả các đèn đều sáng, cung cấp tuần tự các mã hiển thị và bit 1 cấp điện áp cho đèn và thực hiện nhiều lần để lưu ánh sáng lại trên đèn. Nếu tần số quét đèn lớn hơn đáp ứng của mắt (khoảng 40Hz), mắt người sẽ không thấy sự chớp tắt của đèn.

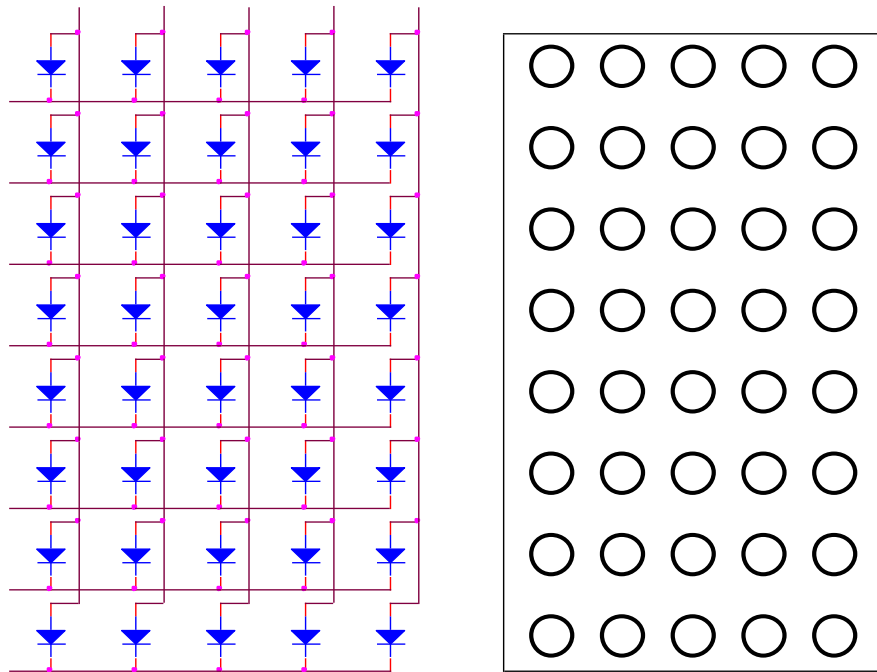


b) Giao tiếp với các loại hiển thị khác.

Để hiển thị các ký tự người ta còn sử dụng các LED ma trận điểm hoặc các màn hình tinh thể lỏng loại ma trận điểm. Với các bộ hiển thị LED dạng ma trận hình 5.43, để sáng lên một điểm cần cung cấp mức 1 vào cột và mức 0 vào hàng tương ứng với nó. Như vậy với ma trận điểm 8 hàng 5 cột trên hình 5.43, thì để hiển thị một ký tự cần cung cấp lần lượt 5 byte mã của nó ra các hàng, tương ứng với 5 cột lần lượt bằng 1.

Tương tự đối với việc điều khiển màn hình tinh thể lỏng dạng ma trận, nhưng chú ý các màn hình tinh thể lỏng cần phải điều khiển bằng tín hiệu điện áp đảo cực liên tục để tránh từ hoá các tinh thể lỏng. Vì vậy các màn hình tinh thể lỏng thường được gắn sẵn mạch lái khi thương mại, mã hiển thị cung cấp cho chúng

có thể dưới dạng ASCII, hoặc một dạng qui định trước, vị trí hiển thị trên màn hình sẽ tính theo các bit địa chỉ cung cấp cho mạch lái.



Các hệ thống vi xử lý lớn như các bộ điều khiển màn hình lớn (CRT) hoặc các màn hình tinh thể lỏng như trên máy vi tính. Để điều khiển được các màn hình loại này cần tìm hiểu cấu trúc điện tử mạch điều khiển của chúng. Các kiến thức này có thể tìm trong các tài liệu về cấu trúc máy tính.

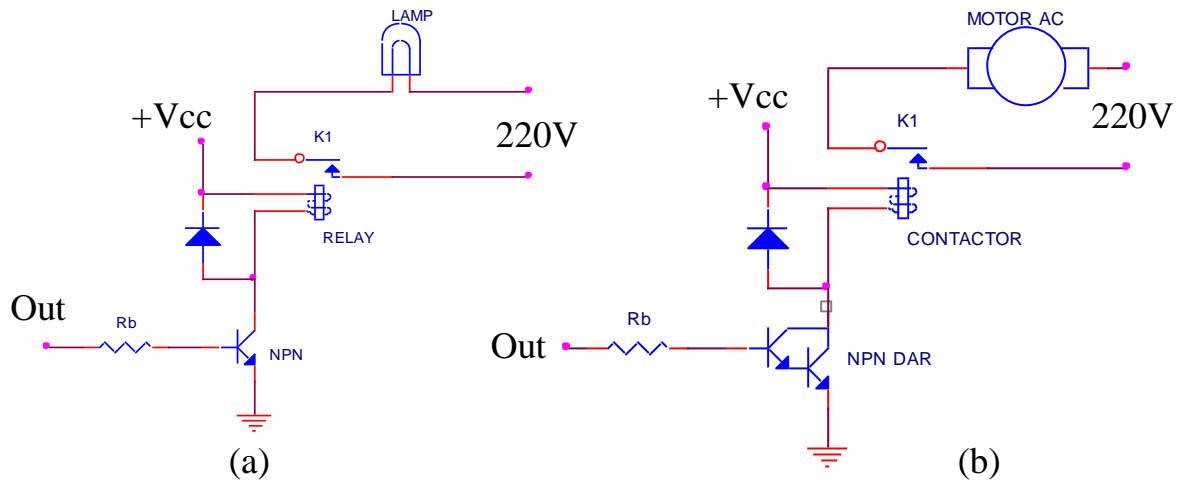
3. Giao tiếp các cổng với các thiết bị công suất lớn.

Các ngõ ra của các bộ vào ra số thông thường chỉ cung cấp dòng điện khoảng vài chục mA khi mức thấp và khoảng 1 tới 2 mA khi ở mức cao. Vì vậy nếu muốn điều khiển các thiết bị có công suất lớn như: các bóng đèn điện, các công tắc điện từ (solenoid), các động cơ ... bằng hệ thống vi xử lý, chúng ta cần phải sử dụng các bộ giao tiếp công suất lớn.

Khi cần giao tiếp với các thiết bị với công suất không lớn lắm, có thể sử dụng các bộ đệm TTL như 74LS07, 7406 Các bộ đệm loại này có thể cho dòng điện 40mA khi cung cấp mức thấp, hoặc một số IC lái thông dụng như ULN2083 ... có thể cung cấp dòng điện tới gần 1 A.

Với các tải yêu cầu dòng điện lớn hơn như các công tắc điện từ, các relay có thể sử dụng các transistor đệm làm việc ở chế độ khoá (hình 5.44a). Khi cần dòng điện lớn hơn có thể sử dụng các transistor công suất, nếu dòng điện tải yêu cầu lớn (dòng cực C lớn) sẽ yêu cầu cung cấp dòng điện cực B lớn lúc này cần ghép hai hoặc nhiều tầng transistor (hình 5.44b). Cấu trúc này có thể cho dòng điện tải lên tới vài chục Ampere. Nhưng với dòng điện vài chục đến vài trăm mA, thông thường người ta sử dụng các hệ thống kết nối Relay – Contactor khi tần số đóng ngắt không yêu cầu lớn, hoặc các linh kiện điện tử công suất như: SCR, Triac, IGBT, hay thông dụng và hiệu quả nhất là các cấu trúc quang điện Solid –

State Relay với các hệ thống yêu cầu tần số đóng ngắt lớn và không gây nhiễu nguồn điện lưới.



Hình 5.45: Mạch đệm transistor.