BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC HÀNG HẢI
BỘ MÔN: KỸ THUẬT MÁY TÍNH
KHOA: CÔNG NGHỆ THÔNG TIN

BÀI GIẢNG

KIẾN TRÚC MÁY TÍNH

TÊN HỌC PHẦN : **KIẾN TRÚC MÁY TÍNH**
MÃ HỌC PHẦN : **17302**
TRÌNH ĐỘ ĐÀO TẠO : **ĐẠI HỌC CHÍNH QUY**
DÙNG CHO SV NGÀNH : **CÔNG NGHỆ THÔNG TIN**

HẢI PHÒNG - 2010

MỤC LỤC

YÊU CẦU VÀ NỘI DUNG CHI TIẾT

Tên học phần: **Kiến trúc máy tính và thiết bị ngoại vi** *Loại học phần:* **2**
Bộ môn phụ trách giảng dạy: **Kỹ thuật máy tính** *Khoa phụ trách:* **CNTT**
Mã học phần: **17312** *Tổng số TC:* **4**

TS tiết	Lý thuyết	Thực hành/Xemina	Tự học	Bài tập lớn	Đồ án môn học
75	60	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 Hệ điều hành, Cấu trúc dữ liệu, các Ngôn ngữ lập trình cấp cao, Mạch và tín hiệu, Kỹ thuật số.

Mục tiêu của học phần:

Giới thiệu cho sinh viên có khái niệm tổng quan về kiến trúc máy tính, các kỹ thuật cơ bản về phần cứng, các kỹ thuật điều khiển và ghép nối.

Nội dung chủ yếu

- Phần I: Kiến trúc máy tính - giới thiệu về các phương pháp biểu diễn thông tin trong máy tính, kiến trúc chung của máy tính điện tử và các thành phần bên trong cũng như chức năng và cấu trúc trong của các thành phần này.

- Phần II: Các thiết bị ngoại vi - giới thiệu chung về hệ thống vào ra, các thiết bị ngoại vi, chức năng và cấu trúc trong của chúng cũng như quá trình ghép nối các thiết bị này với máy tính điện tử.

Nội dung chi tiết của học phần:

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	**LT**	**BT**	**TH**	**KT**
Chương I *:* Giới thiệu chung	**6**	**6**			
1. Lịch sử phát triển		1			
2. Phân loại		1.5			
3. Hệ đếm		0.5			
4. Các loại mã		1			
5. Đơn vị thông tin		0,5			
6. Biểu diễn số nguyên trong máy		1			
7. Mảng và ngăn xếp		0,5			
Chương II : Bộ xử lý trung tâm	**14**	**8**		**5**	**1**
1. Tổ chức của bộ xử lí trung tâm		1			
2. Tập các thanh ghi		1			
3. Đơn vị số học và Logic		1,5			
4. Đơn vị điều khiển		2			
5. Cấu trúc kết nối		0,5			
6. Tập lệnh Assembly và các Mode địa chỉ		2			
Chương III: Hệ thống nhớ	**13**	**12**			**1**
I. Tổng quan		2			

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	BT	TH	KT
1. Hệ thống nhớ					
2. Phân cấp hệ thống nhớ					
II. Bộ nhớ bán dẫn		2			
1. Tổ chức của bộ nhớ bán dẫn					
2. Các loại bộ nhớ bán dẫn					
II. Bộ nhớ Cache		3			
1. Khái niệm					
2. Các phương pháp ánh xạ dữ liệu					
IV. Kỹ thuật giải mã địa chỉ		4			
1. Khái niệm					
2. Kĩ thuật giải mã địa chỉ					
V. Kỹ thuật quản lí bộ nhớ		1			
1. Các kí thuật quản lí bộ nhớ					
2. Quản lí bộ nhớ ảo					
Chương IV: Hệ thống vào ra	**9**	**8**			**1**
I. Tổng quan		2			
1. Các Thiết bị ngoại vi					
2. Các Mô đun vào ra					
II. Ghép nối vào ra		3			
1. Ghép nối nối tiếp					
2. Ghép nối song song					
III. Các phương pháp vào ra		3			
1. Vào ra bằng phương pháp thăm dò					
2. Vào ra bằng Ngắt					
3. Vào ra bằng phương pháp truy nhập trực tiếp bộ nhớ					
Chương V: Thiết bị nhập dữ liệu	**5**	**5**			
1. Khái niệm		0,5			
2. Bàn phím		3			
3. Chuột		1			
4. Các thiết bị nhập liệu tiên tiến		0,5			
ChươngVI : Thiết bị xuất dữ liệu	**5**	**5**			
1. Giới thiệu chung		0,5			
2. Màn hình		3,5			
4. Máy in		1			
Chương VII : Thiết bị lưu trữ	**11**	**6**		**5**	
1. Các khái niệm		1			
2. Đĩa từ		3			
3. Đĩa quang		0,5			
4. Các thiết bị lưu trữ khác		1,5			

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	BT	TH	KT
Chương VIII : Thiết bị ghép nối và truyền thông	**12**	**6**		**5**	**1**
1. Giới thiệu chung		1			
2. Bộ chuyển đổi ADC – DAC		2			
3. Các chuẩn ghép nối và truyền thông		1			
4. Thiết bị điều khiển ghép nối truyền thông		2			

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, hoàn thành bài tập lớn theo yêu cầu.

Tài liệu học tập :

- William Stalling - Computer Organization and Architecture, 1997
- Mc. Graw - Computer Architecture, 1997
- Văn Thế Minh - Kỹ thuật vi xử lý - NXB giáo dục 1997
- Nguyễn Kim Khánh - Giáo trình kiến trúc máy tính - ĐHBK Hà nội
- Nguyễn Đình Việt - Giáo trình kiến trúc máy tính - ĐHQG Hà nội
- Trần Thái Bá - Điều khiển và ghép nối thiết bị ngoại vi - NXB thống kê 2000
- Trần Quang Vinh - Cấu trúc máy tính - NXB giáo dục 1997
- Võ Văn Thành - Máy vi tính sự cố chẩn đoán và cách giải quyết - NXB thống kê 1996.
- William Stalling- *Computer Architecture Advanced,* 1997

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: 15 / 6 / 2010

Trưởng Bộ môn: ThS. Ngô Quốc Vinh

Chương I: GIỚI THIỆU CHUNG

1.1. Lịch sử phát triển và phân loại

1.1.1. 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 Mĩ - chiếc máy tính được đặt tên là ENIAC (Electronic Numerial Itergrator 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 tuỳ 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 thuât.

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 lí 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 vưc 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 lí. 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 chíp 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.

1.1.2. Phân loại máy tính

Máy tính (computer) là một khái niệm tương đối rộng, tuỳ 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:

1.1.2.1. Phân loại theo khả năng

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

Máy tính lớn (mainframe computer):

Là những máy tính cỡ lớn, có khả năng hỗ trợ cho hàng trăm đến hàng ngàn người sử dụng. 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. Thường được sử dụng trong chế độ các công việc sắp xếp theo lô lớn (Large-Batch-Job) hoặc xử lý các giao dịch (Transaction Processing), ví dụ trong ngân hàng.

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 (microcontroler) 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):

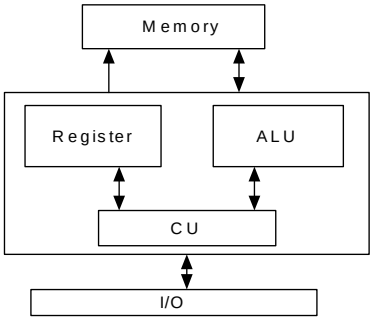
Chỉ được một người sử dụng. Giá thành của chúng rẻ do cấu hình đơn giản, được chuẩn hoá, và được sản xuất hàng loạt với số lượng lớn.Cùng với sự phát triển của khoa học công nghệ mà máy tính cá nhân ngày nay đã có thể làm được những công việc mà trước kia vốn chỉ là đặc quyền của máy tính lớn.

1.1.2.2. Phân loại theo nguyên lý

- Máy tính cơ khí.
- Máy tính tương tự
- Máy tính số

1.1.2.3. 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 (regiser)
- ✓ ALU (Arithmetic Logical Unit)
- ✓ CU (Control Unit).

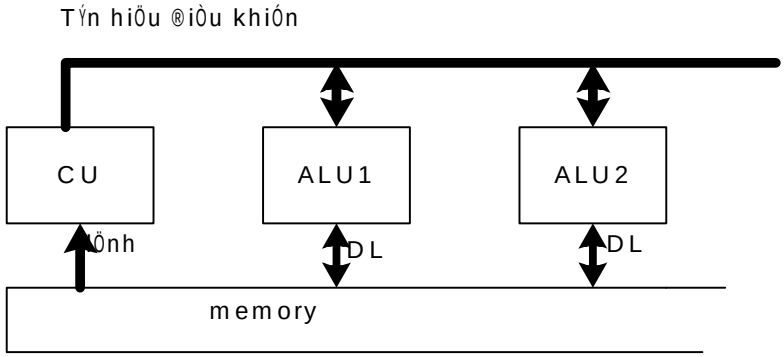
Đặc điểm :

- ✓ 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

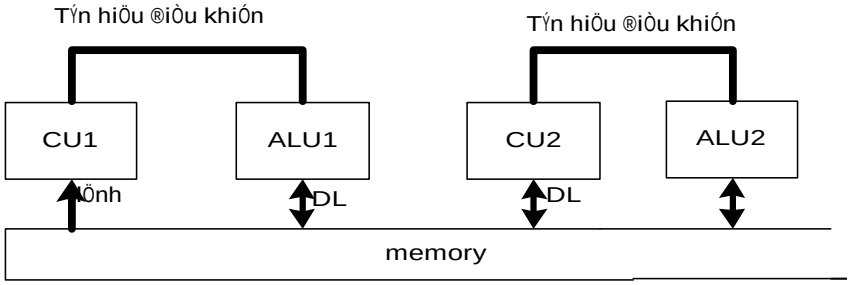
+ SIMD (Single Instruction Stream-Multiple Data Stream)



Đặc điểm:

- o Có một đơn vị điều khiển, n phần tử xử lý
- o 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 tren các DL khác nhau.
- Tất cả các CPU cùng chia xẻ 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

1.2. Biểu diễn thông tin trên máy tính

1.2.1. Hệ đếm

a. Hệ đếm bất kỳ

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

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

Trong đó N là một số nguyên có n chữ số. Chữ số a_i tại vị trí i (i=0...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:

$$N = 1547d = 1.10^3 + 5.10^2 + 3.10^1 + 7.10^0$$

c. Hệ đếm nhị phân

Được hình thành trên cơ sở đại số lôgic Boole, xuất hiên 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 luc 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:

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

1.2.2. Đổi số thập phân ra số nhị phân hoặc ngược lại

Để đổi số 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 lsg LSB (least significant bit) và trị số cuối cùng (ngoài cùng bên trái) được gọi là MSB (most signifcant bit).

Ví dụ: N = 113*d*

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	

Kết quả = 113*d* = 110001*b* = 71*h*

1.2.3. 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

1.2.4. 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

1.2.5. 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

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

Trong đó R số thực cần biểu diễn gồm n trị số đứng trước và m trị số đứng sau dấu chấm. Tuỳ 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 $X = (-1)^S.M.R^E$

Ví dụ: R=-750*d* = $-0,75.10^3 = -0,75E3$

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:

$$R = (-1)^S.(1 + M_1.2^{-1} + ...+M_n2^{-n}).2^{E7...E0-127}$$

S	E_7-E_0	Định trị(M_1- M_{23})

Ví dụ:

428CE9FC*h* = 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 phảy 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

1.2.6. Biểu diễn các dạng thông tin khác

a. Biểu diễn hình ảnh

b. Biểu diễn âm thanh

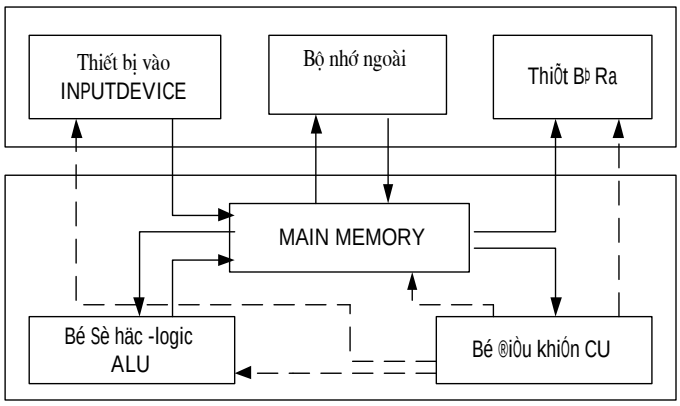
c. Các đai lương vât lý khác

1.3. Các loại 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



C¸c ®¬n vÞ chøc n¨ng c¬ b¶n cña m¸y tÝnh ®iÖn tö

(Các đường vẽ nét đứt chỉ mối quan hệ. Các đường nét liền là đường truyền dữ liệu)

- 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
- 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.

- 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.
- 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.
- 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.

CÂU HỎI VÀ BÀI TẬP

1.1. Trình bày hiểu biết của anh chị về các giai đoạn phát triển của máy tính điện tử

1.2. Trình bày về phân loại máy tính theo kiến trúc

1.3. Đổi số 23786d, -17456d sang số nhị phân nguyên có dấu 16 bít

1.4. Đổi số 3476,0655d sang số thực nhị phân 32 bít theo chuẩn IEEE754 – 1995

1.5. Trình bày cấu trúc chung một máy tính điện tử

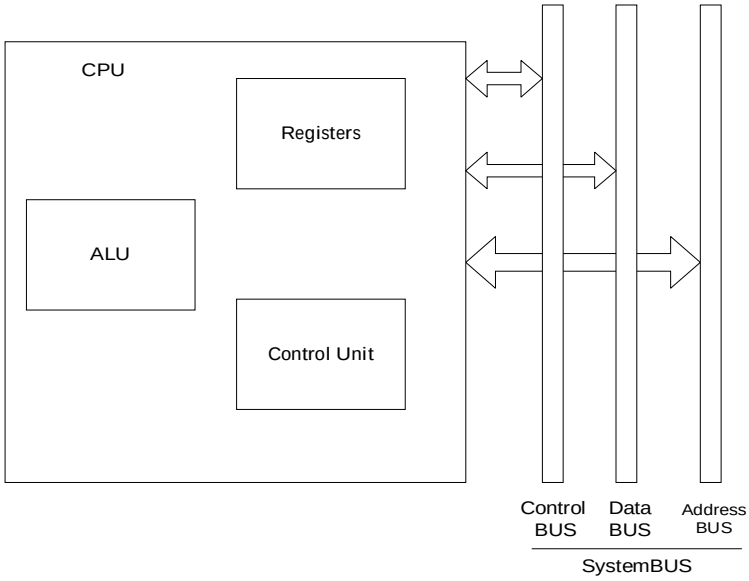
Chương II: BỘ XỬ LÝ TRUNG TÂM

2.1. 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:

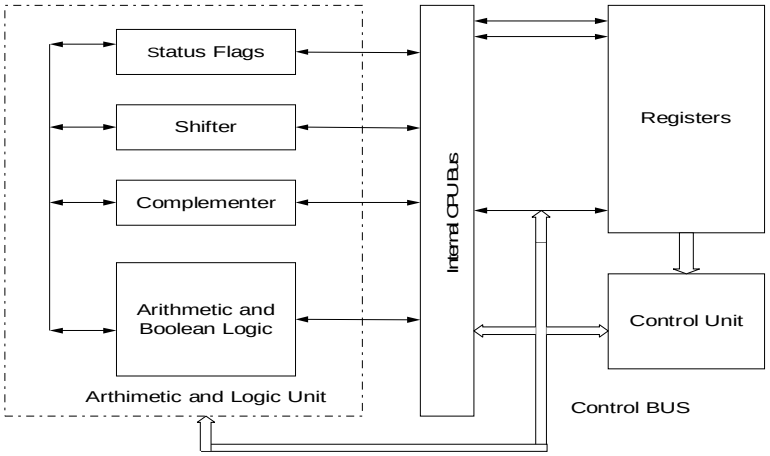
- Fetch Instructions(chỉ lệnh tìm nạp): CPU phải đọc các chỉ lệnh từ bộ nhớ.
- Interpret Instructions: chỉ lệnh phải được giải mã để xác định hành động nào được yêu cầu.
- Fetch data (dữ liệu tìm nạp): Sự thi hành một chỉ lệnh có thể yêu cầu thực hiện một vài thao tác số học hoặc lôgi trên dữ liệu.
- 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 trên xác định các kết nối của CPU với phần còn lại của hệ thống thông qua bus hệ thống. Một giao diện tương tự có thể cần đến cho tất các 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 ảnh chi tiết hơn cấu trúc CPU được mô tả ở trang sau. 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.



2.2. 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 Asembly 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

2.2.1. 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. Thỉnh thoảng, 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 thuờng, 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 địa chỉ hoá (ví dụ thanh ghi gián tiếp, dịch chuyển). Trong các trường hợp khác, có một phần hoặc sự phân chia rõ ràng giã 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 sốvà 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.

2.2.2. 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. Với các máy tính khác nhau sẽ có tổ chức thanh ghi khác nhau và sử dụng thuật ngữ khác nhau.

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 Fuffer Register): chứ một từ dữ liệu được ghi vao 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 [STRI79].

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 luỹ 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 sl 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

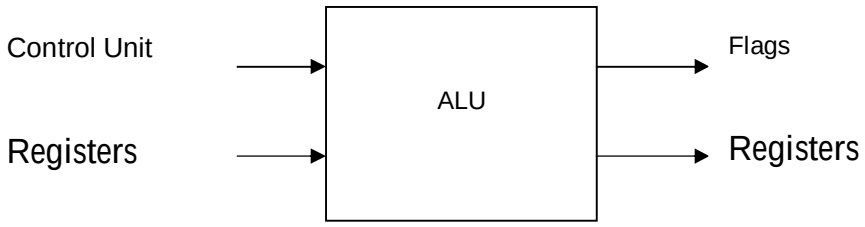
9 trong số 16 bit của thanh ghi này được sử dụng, mỗi bit có thể được thiết lập hay xoá dể chỉ thị kết quả của mỗi thao tác trước đó hoặc trạng thái hiện thời bộ XL

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

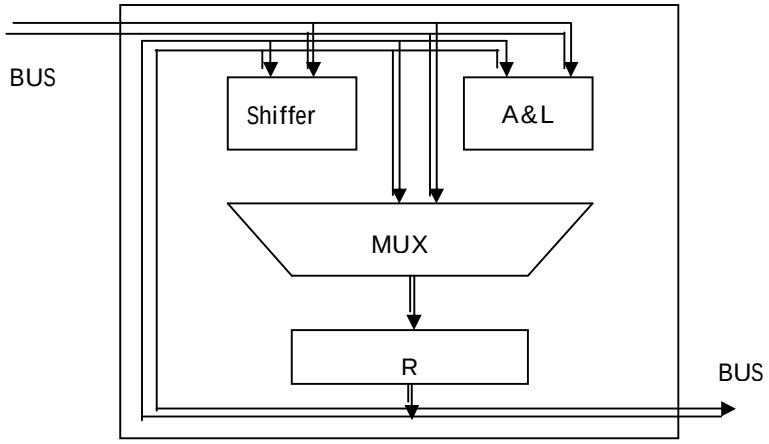
2.3. Đơn vị số học và logic ALU (Arithmetic and logic unit)

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 của 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 trang sau)



Các phép toán cơ bản của ALU

Bộ cộng, trừ:

2.4. 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)

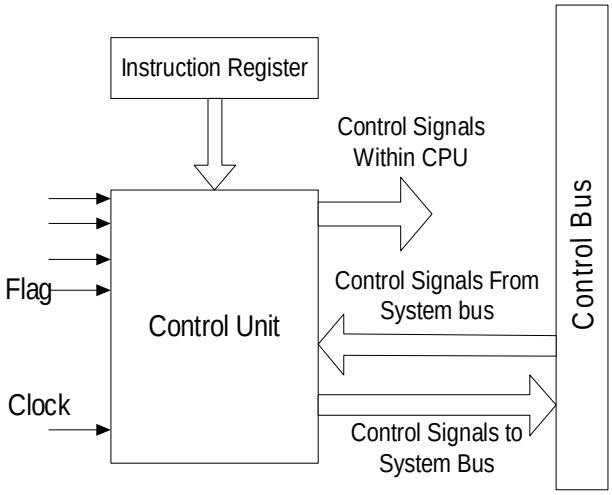
Đơn vị điều khiển thực hiện hai công tác chính:

- 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.**

2.4.1 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 logíc 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.



Các tín hiệu của đơn vị điều khiển được thể hiện trong hình trên. Bao gồm:

- ✓ Tín hiệu vào
- ✓ Tín hiệu ra

a. Các tín hiệu vào

- 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 mộ vài lần nhắc đến như là chu kỳ thời gian xử lý, hoặc chu kì 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 kì 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 kiển sẽ lượng gia 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.

b. Các tín hiệu ra

- 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.

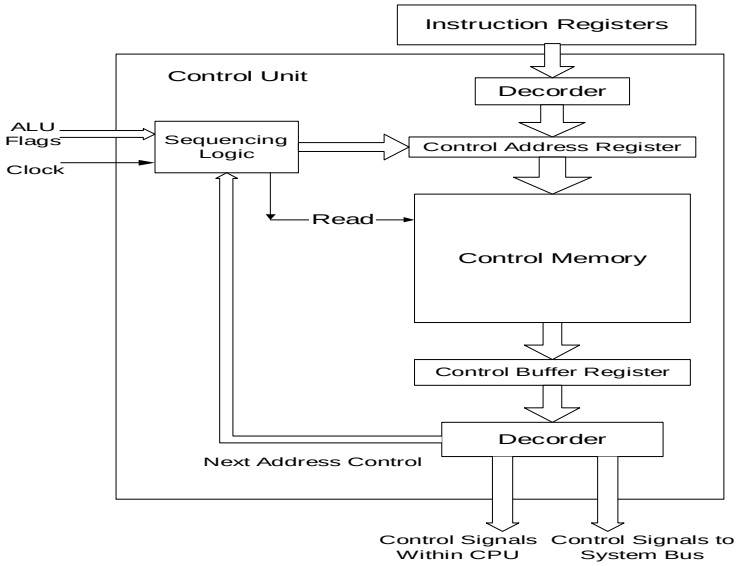
Chúng ta hãy xem xét lại chu kỳ tìm nạp để xem cách đơn vị điều khiển duy tri diều khiển. Đơn vị điều khiển giữ dấu vết nơi nó là trong chu kỳ tìm nạp. Tại điểm quy định, nó biết chu kỳ tìm nạp đượoc thực hiện tiếp. Bước đầu tiên là di chuyển các nội dung của PC và MAR. Đơn vị điều khiển dùng nó bằng 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 gia lượng PC. Đơn vị điều khiển làm việc này bằng vệc gửi các tín hiệu kèm theo đồng thời.

- Một tín hiệu điều khiển mở các cổng cho phép nội dung của MAR đưa vào bus địa chỉ.
- Bộ nhớ đọc tín hiệu điều khiển trên bus điều khiển.
- Một tín hiệu điều khiển các cổng cho phép nội dung bus dữ liệu được chứa trong MBR.
- 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 (Memory buffer register) và IR (Instruction Register)

2.4.2. 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

- Từ mà địa chỉ được xác định trong thanh ghi địa chỉ điều khiển được đọc vào thanh ghi bộ đệm điều khiển.
- Nội dung của thanh ghi bộ đệm điều khiển phát ra tín hiệu điều khiển và thong tin địa chỉ tiếp theo cho đơn vị lôgic 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.

2.4.3. 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. Mộ 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 hon 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 truyền, 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.

2.5. Cấu trúc kết nối - 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 lạc, 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 tuỳ 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 rã 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 kì nhất định. Hoạt động của vi xử lý đòi hỏi thời gian là bội số của chu kì 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.

2.6. Tập lệnh và các Mode địa chỉ

2.6.1. Tập lệnh của CPU

Chức năng máy tính:

- ✓ Xử lí tin
- ✓ Truyền thông

Về cơ bản việc xử lí thông tin và truyền thông đều dựa trên nguyên tắc thự 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

2.6.2. 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. Tuỳ 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 đễ 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.

a. Nhóm lệnh cơ sở

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

b. 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 : đợi 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

b. Nhóm lệnh số học

**** 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 MUL (Multiply) và IMUL (Integer MUL):*** Lệnh nhân giá trị số học. Khi xét các số có dấu và không dấu thì kết qủa 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):*** Lệnh chia số học. Khi xét các số có dấu và không dấu thì kết qủa 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

c. 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,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

d. Các lệnh dịch và quay

Quay và dịch các bit trong toán hạng đích sang trái hoặc phải 1 hoặc 1 số vị trí.

- ✓ *Lệnh dịch:* các bit bị dịch ra khỏi toán hạng sẽ bị mất
- ✓ *Lệnh quay:* các bit bị dịch ra 1 phía của toán hạng đích sẽ đựoc đưa trở lại phía bên kia quay 1 vị trí

Cú pháp:

Quay, dịch 1 vị trí

Mã lệnh Toán hạng đích,1

Quay, dịch 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

Lệnh SAL(Shift Arithmetic Left): Tương tự như lệnh SHL

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 trái 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

Lệnh SAR(Shift Arithmetic Right): Tương tự như lệnh SHR

*** Các lệnh quay**

Lệnh ROL(Rotate Left):

Dịch các bít 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 bít 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 (Rotate Carry Left): quay trái qua cờ nhớ

Dịch các bít 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(Rotate Carry Right): quay phải qua cờ nhớ

Dịch các bít của toán hạng đích sang phải. Bit LMSB được đặt vào CF, giá trị của CF được đưa vào bit phải nhất (LSB) của toán hạng đích

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
```

e. 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 Greate, 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
JAE/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

Kí hiệu	*Chức năng*	*Điều kiện nhảy*
JNO	Nhảy nếu không tràn	OF=0
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

2.6.3. Hợp ngữ (Assembly)

a. 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
```

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

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

Lời 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

b. 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

c. Các hằng có tên

Dùng các tên tượng trưng để biểu diễn các hằng số

Cú pháp:

Tên EQU Hằng số

ví dụ:

CONST EQU 0Ah

cũng có thể dùng chuỗi:

MSG EQU "hello"

chú ý: bộ nhớ không dành chỗ cho các hằng có tên

d. 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 trỏ tới PSP
- ✓ IP được gán giá trị 100h
- ✓ SP được gán giá trị FFFEh

Cấu trúc chương trình

```
.MODEL TINY
.CODE
        ORG         100h
START: JMP CONTINUE
        ; khai báo dữ liệu
CONTINUE:
        MAIN PROC
                ; đoạn mã
                ...
                INT    20 h; 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 tuỳ 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
```

e. 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
```

f. 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"

2.6.4. 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 như một con trỏ trỏ đến các ô nhớ

Khuôn 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, lưu kết quả vào AX

```
        XOR   AX,AX
        LEA   SI,W
Tong:
        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 được 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 đợc 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 thờng 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
```

CÂU HỎI VÀ BÀI TẬP

2.1. Trình bày kiến trúc chung của máy tính theo nguyên lý VonNewman? Nêu chức năng từng đơn vị? Phân loại máy tính theo kiến trúc?

2.2 Trình bày sơ đồ cấu trúc, chức năng nhiệm vụ của bộ xử lý trung tâm (Nêu rõ chức năng của từng đơn vị)

2.3. Trình bày tổ chức thanh ghi trong vi xử lý 8086?

2.4. Trình bày chức năng nhiệm vụ cấu tạo của đơn vị số học và logic ALU?

2.5. Các loại tín hiệu điều khiển của đơn vị xử lý trung tâm (Vẽ sơ đồ giải thích)?

2.6. Trình bày sơ đồ cấu trúc, chức năng nhiệm vụ của đơn vị điều khiển vi chương trình ?

2.7. Trình bày khái niệm về BUS? Phân loại BUS

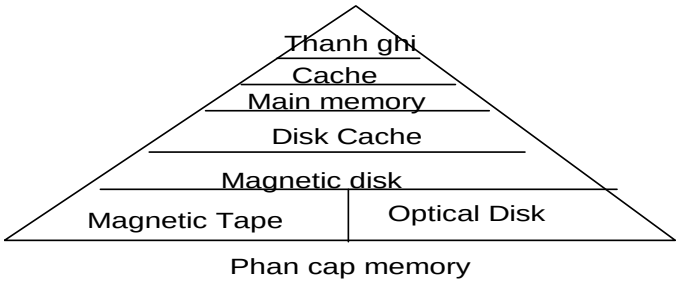
Chương III: HỆ THỐNG NHỚ

3.1. Khái quát về hệ thống nhớ

- 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 dụng 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 bít 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 vớ 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*: 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

3.2. Phân cấp bộ nhớ



Phan cap memory

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.

3.3. Bộ nhớ bán dẫn

3.3.1. Các loại bộ nhớ bán dẫn

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

a. 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:*** 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 .

b. 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 (Programable 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 chíp. Flash memory sử dụng một transistor trên một bit, và do đó giành được mật độ cao.

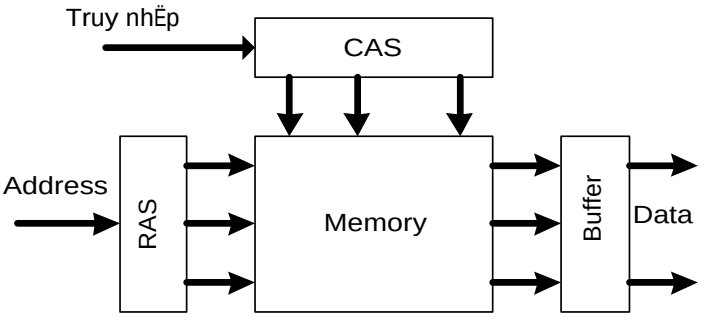
3.3.2. Tổ chức bộ nhớ

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



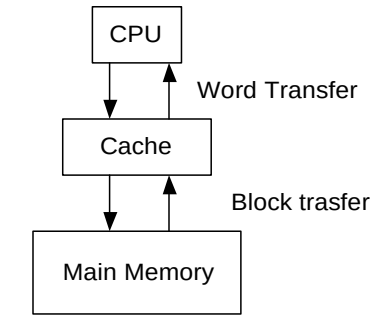
CAS: Colum Address Select

RAS: Row Address Select

3.4. Cache Memory

3.4.1. Nguyên tắc

Cache memory được dùng có 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.

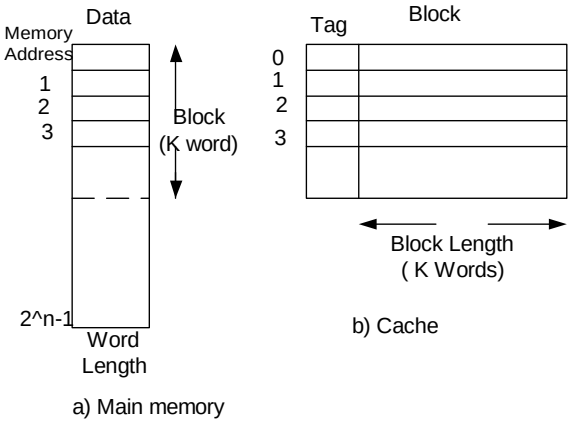


Cache and Main Memory

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 dó 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.

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<< M). Tại bất kỳ thời điểm nào, một vài tập con của khối của bộ nhớ lưu trú trong 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 độc 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.

3.4.2. 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 $K=2^{N1}$ byte -. có $M=2^N/k$ khối.

Chia Cache thành C đường, mỗi đường k byte nhớ: C<<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 như thế nào.

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
- ✓ Bock i -> line (i mod C)

Giả sử cache có 2^{n2} ngăn nhớ (đường), địa chỉ do CPU phát ra là n bit

Tag	n2	n1

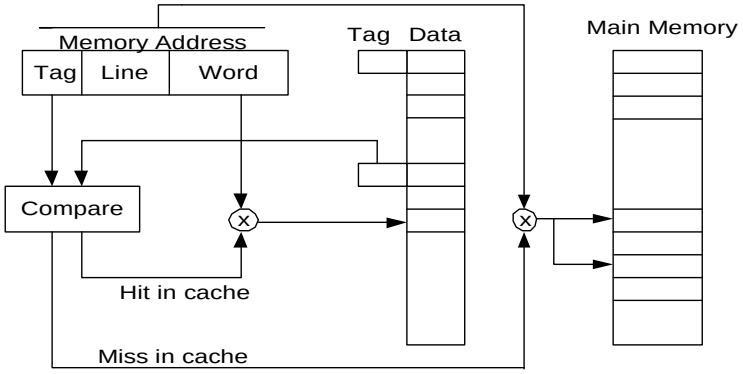
n_1 xác định số byte trong khối 2^{n1} ->byte

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

n_2 bít 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)

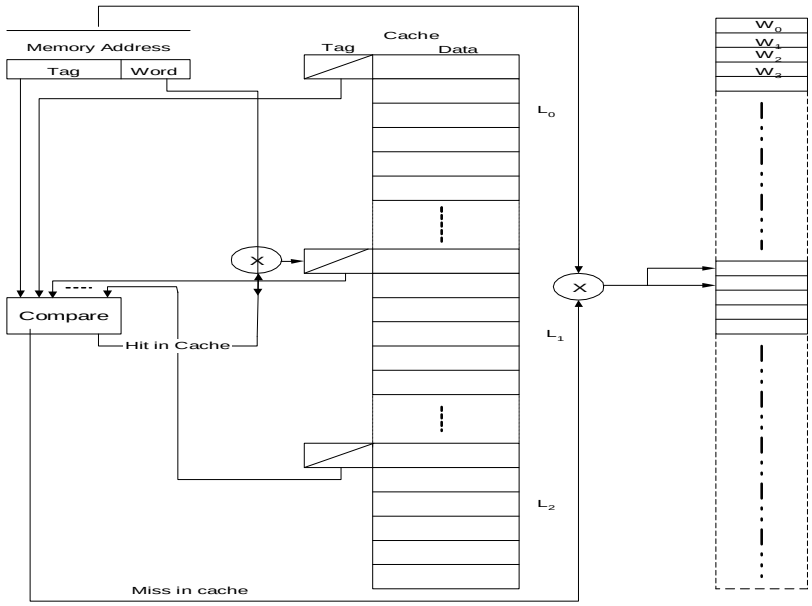


Bộ so sánh : thông dịch địa chỉ và so sánh nối vào.

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.

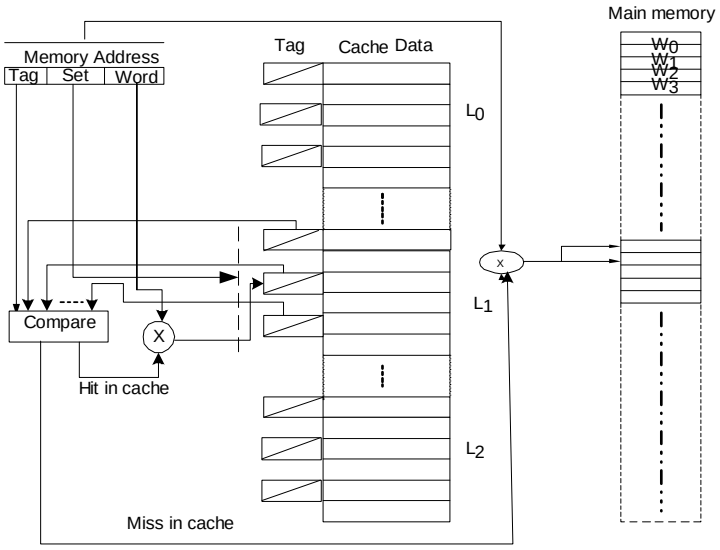


Trong trường hợp này n bit chia ra 2 trường:

Tag	word

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 : Cho bộ nhớ chính 4 GB, kích thước cache 16 kB, kích thước khối 32 byte. Xác định địa chỉ do CPU phát ra theo 3 phương pháp ánh xạ

Bộ nhớ chính có dung lượng 4GB = 2^{32}B -> Số bit địa chỉ do CPU phát ra là n=32

1 khối (block) có kích thước là 32B = 2^5B -> n1=5

* Xét trong trường hợp ánh xạ trực tiếp

Số đường trong cache: n2 = $\frac{16kb}{32byte} = \frac{2^{14}}{2^5} = 2^9$ -> n2 = 9 bit

➔ Tag = 32 – 9 – 5 = 18 bit

* Với liên kết hoàn toàn

➔ Tag=32 – 5 = 27 bit

* Với liên kết tập hợp (2 đường)

Số đường trong cache: n2 = $\frac{16kb}{32byte.2line} = \frac{2^{14}}{2^5.2} = 2^8$ -> n2 = 8 bit

➔ Tag = 32 – 8 – 5 = 19 bit

Trực tiếp

18	9	5

Hoàn toàn

27	5

Tập hợp

19	8	5

3.5. Quản lý bộ nhớ

3.5.1. 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 dành cho hệ điều hành (resident monitor)
- Một phần dành cho chương trình đang được thực hiện.

Trong một hệ thống lập trình đa nhiệm, phần của bộ nhớ dành cho chương trình cần 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).

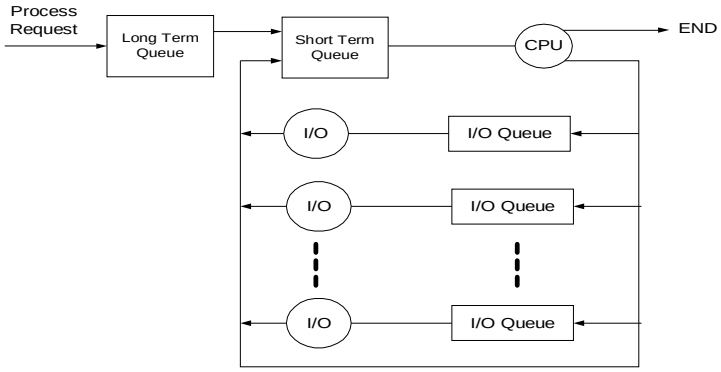


Figure 7.14 Queuing diagram representation of processor scheduling

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)

Trên hệ thống luôn tồn tại 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.

Thực tế, các hoạt động vào ra chậm hơn rất nhiều so với khả năng tính toán của CPU, nên trong các hệ thống lập trình đơn nhiệm CPU nhàn rỗi trong hầu hết thời gian.

Trong quá trình hoạt động, bộ nhớ lưu giữ các tiến trình và dữ liệu, CPU có thể chuyển tới tiến trình khác khi một tiến trình đang đợi. Nhưng vi 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 trong khi có dung lượng bộ nhớ lớn hơn.

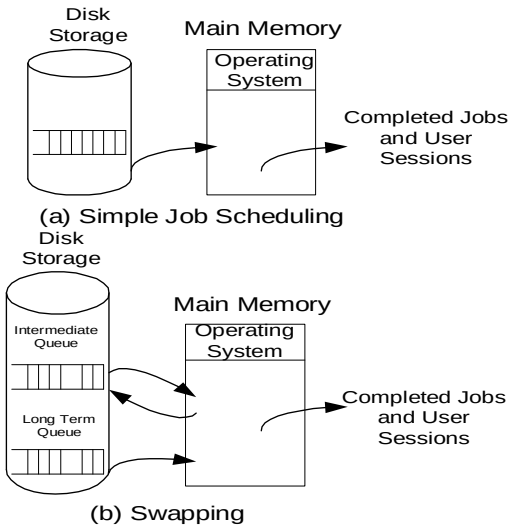


Figure 7.15. The use of swapping

Một giải pháp khác là tráo đổi (swapping). 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 tráo 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).

Kỹ thuật tráo đổi, dù sao cũng là thao tác vào ra (I/O) và vì vậy cũng có thể phát sinh các sự cố về dữ liệu. 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), tráo đổi sẽ nâng cao sự thực thi. Một giản đồ tinh vi hơn là bộ nhớ ảo sẽ cải thiện sự thi hành hơn sự tráo đổi đơn giản.

b. Phân vùng (Partioning)

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. Lược đồ đơn giản nhất cho bộ nhớ có thể phân vùng là sử dụng "các phân vùng kích thước cố định" (fixed-size partitions)

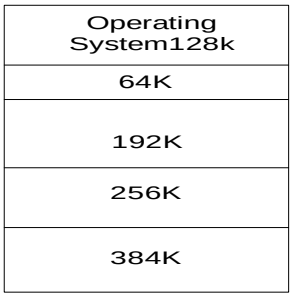


Figure 7.16 Example of Fixed Partioning

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. 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.

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 đại 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 kết 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 cùng 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 vẽ)

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 lôgic-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).

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 rỗi và một bảng trang được thiết lập.

3.5.2. 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), có thể 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.

Như 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 tráo 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 tráo đổi các trang hơn là thực hiện các chỉ lệnh. Việc tránh thrashing đã là mộth 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 hạnh 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.

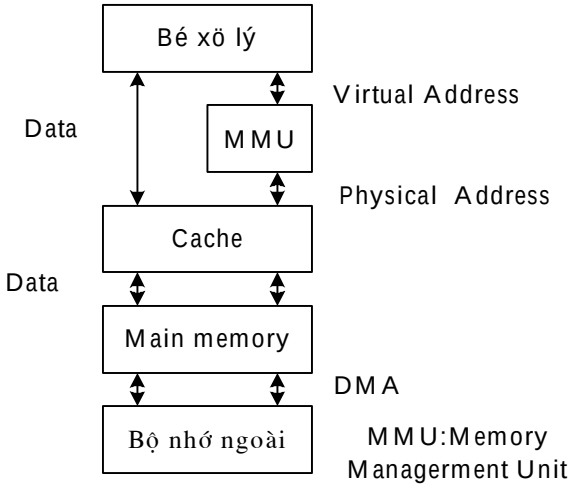
Bởi vì một tiến trình thực hiện chỉ trong bộ nhớ chính, chỉ bộ nhớ chính được tham chiếu như là bộ nhớ thực sự. Nhưng một lập trình viên, hoặc người dùng nhật thấy một bộ nhớ lớn hơn rất nhiều- được định rõ trên đĩa cứng. Nó được tham chiếu đến như là bộ nhớ ảo.

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 vaat 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.

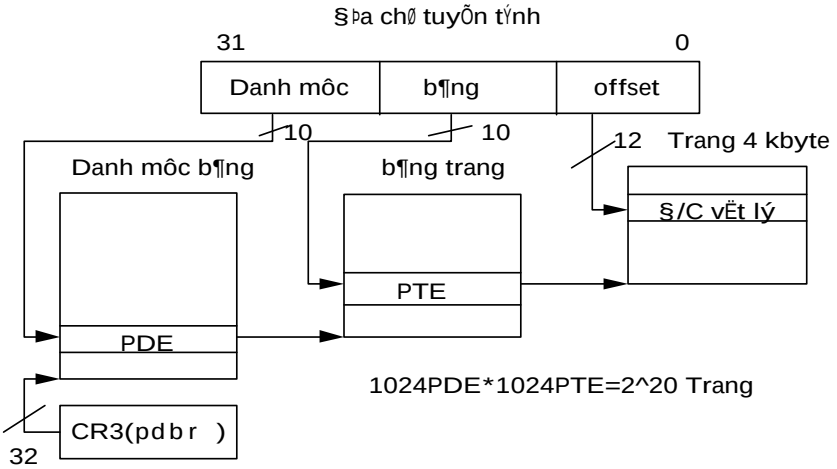
c. Cách quản lý trang và phương pháp biên dịch địa chỉ tuyến tính

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 trỏ danh mục trang : tổ hợp 3 giá trị 64 bit, mỗi giá trị trỏ 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)



Biªn dÞch ®Þa chØ tuyÕn tÝnh (trang 4 kbyte)

Hình vẽ cho ta thấy cách dùng danh mục trang và bảng trang khi ánh xạ địa chỉ tuyến tính sang trang4 kbyte. Giá trị trong danh mục trang trỏ đến một vị trí trong bảng trang, giá trị tương ứng trong bảng trang trỏ 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).

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 trọn 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.

3.5.3. 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.

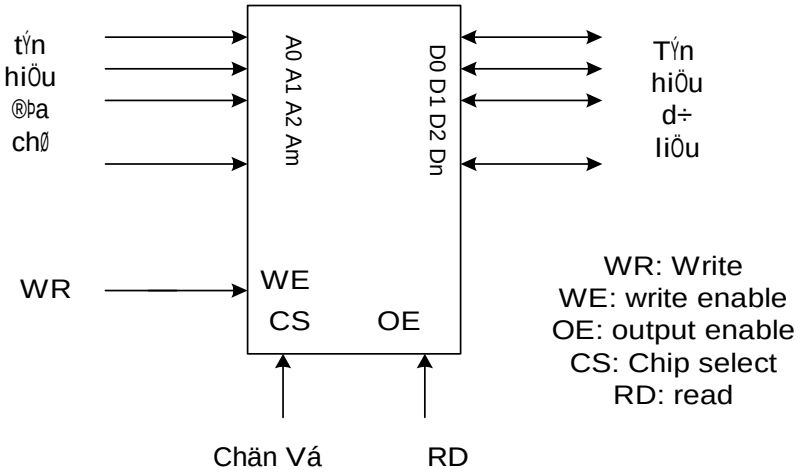
- ✓ Nó làm đơn giản cách trình bày sự phát triển của các cấu trúc dữ liệu. Nếu lập trình viên không biết kích thước của một cấu trúc dữ liệu riêng biệt sẽ là bao nhiêu, nó sẽ không cần hướng dẫn. Cấu trúc dữ liệu có thể được phân bổ và đoạn của nó và hệ điều hành sẽ mở rộng hoặc co rút lại đoạn nếu cần thiết.
- ✓ 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 .
- ✓ 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.
- ✓ Nó tự bảo vệ mình. Khi một đoạn có thể được xây dựng để chứa đượng 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 .

3.6. Kỹ thuật giải mã địa chỉ

3.6.1. 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 tuỳ 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.

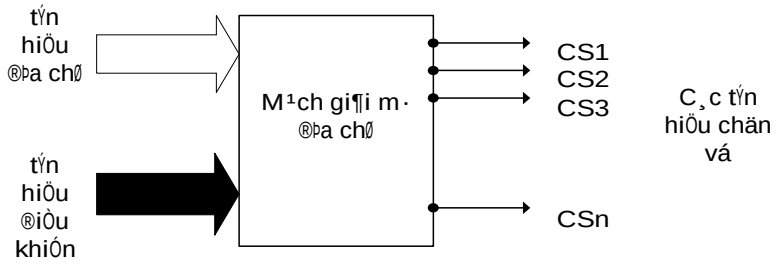
3.6.2. Giải mã địa chỉ cho bộ nhớ

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. Cấu tạo chung của bộ giải mã địa chỉ:



M¹ch gi¶i m· ®Þa chØ tæng qu¸t

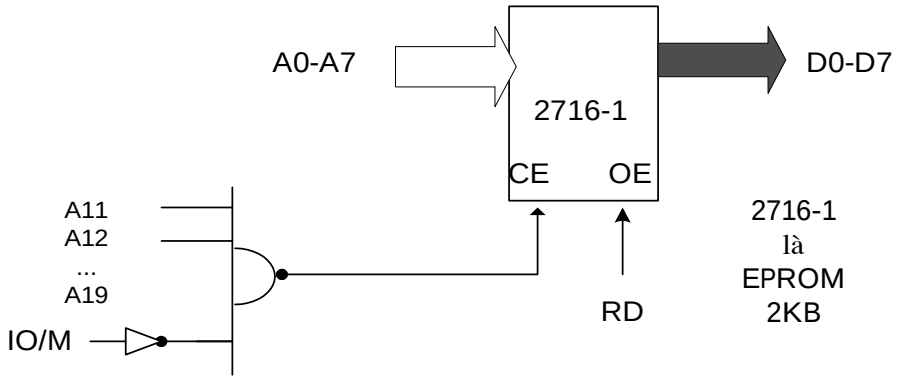
- Tín hiệu điều khiển : IO/ M dùng để phân biệt đối tượng mà CPU chon làm việc là bộ nhớ hay thiết bị vào ra.
- Tín hiệu địa chỉ : là các bit địa chỉ có quan hệ nhất định đến việc chọn vỏ ở đầu ra.

Thông thường khi thiết kế mạch giải mã người ta thường tính dôi ra để dự phòng, sao cho sau này co thể tăng thêm dung lượng bộ nhớ.

b. 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 FF800*h*-FFFFF*h* (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 =FF800h

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 =FFFFFh

Số chân địa chỉ trong CPU 8088 là 20 chân đánh số từ A_0 đến A_{19}. Trong mạch giải mã này ERPROM dung lượng 2KB -> sử dụng 11 bít địa chỉ thấp từ A_0 đến A_{10} để chọn từ nhớ trong ERPROM . Các bít 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

c. Giải mã dùng mạch giải mã kiểu 74LS138

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 G2b

A_{19} được đưa vào G1

Nếu dùng EPROM 2Kb -> cần sử dụng 32 mạch nhớ.->sử dụng 4 bộ giải mã LS138 để chọn chíp 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 G2A của bộ giải mã LS138 1 và 3; qua mạch NOT đổi thành xung âm đưa vào chân G2A của bộ giải mã LS138 2và 4.

Bit A_{15} được đưa vào chân G2B của bộ giải mã LS138 1 và 2; qua mạch NOT thành xung âm đưa vào chân G2B của bộ giải mã 3,4 .

Các chân $A_{16} \div A_{19}$ 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:

CÂU HỎI VÀ BÀI TẬP

3.1. Trình bày khái niệm, nguyên tắc hoạt động của bộ nhớ Cache?

2.2. Trình bày kỹ thuật quản lý bộ nhớ (Khái niệm, kỹ thuật hoán đổi, phân vùng, phân trang)

3.3. Trình bày khái niệm bộ nhớ ảo (Virtual memory)? Cách quản lý trang và phương pháp biên dịch địa chỉ tuyến tính?

3.4. Cho bộ nhớ chính 32 MB, kích thước cache 8 kB, kích thước khối 32 byte. Xác định địa chỉ do CPU phát ra theo 3 phương pháp ánh xạ

3.5. Trình bày sơ đồ 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: HỆ THỐNG VÀO RA

4.1. 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:

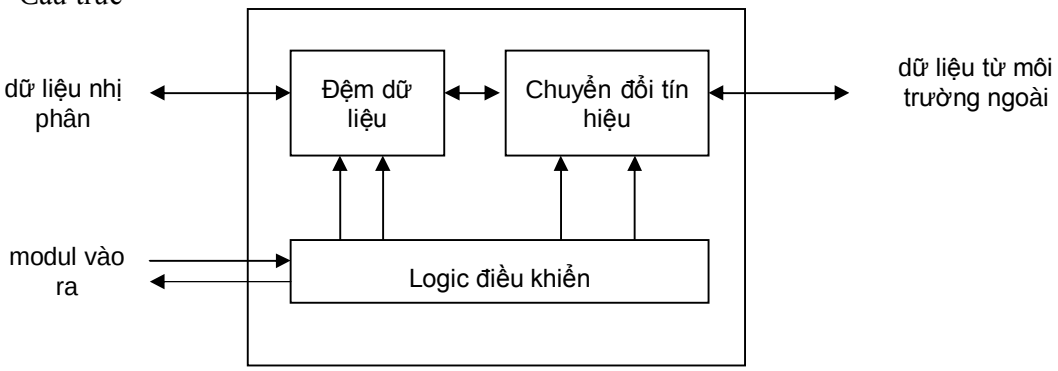
- ✓ Địa chỉ hoá tách biệt (vào ra trực tiếp): không gian địa chỉ cổng độc lập với không gian địa chỉ bộ nhớ.
- ✓ Địa chỉ hoá theo bản đồ bộ nhớ: không gian địa chỉ cổng vào ra nằm trong không gian địa chỉ bộ nhớ.

4.1.1. Các thiết bị ngoại vi

Chức năng:

- ✓ Trao đổi thông tin người - máy
- ✓ Trao đổi thông tin máy - máy

Cấu trúc



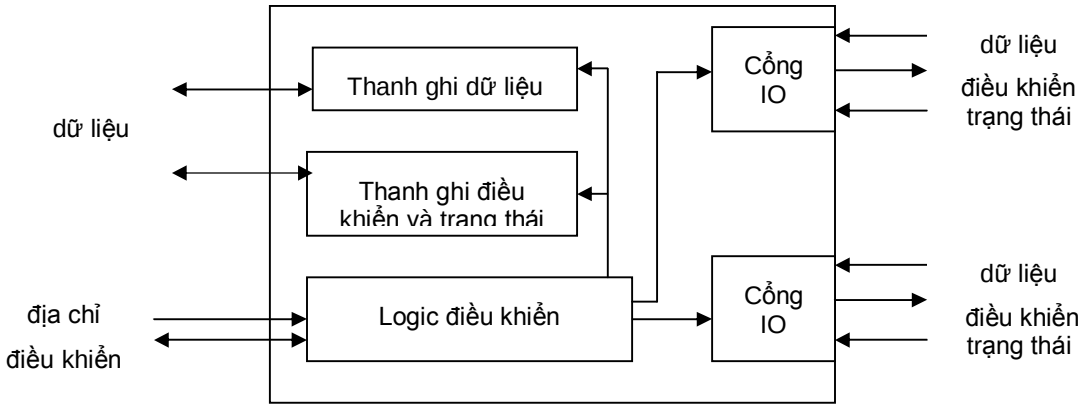
Logic điều khiển:

- ✓ 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.

4.1.2. 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: (Hình vẽ trang trước)

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

4.2. Ghép nối máy tính với thiết bị ngoại vi

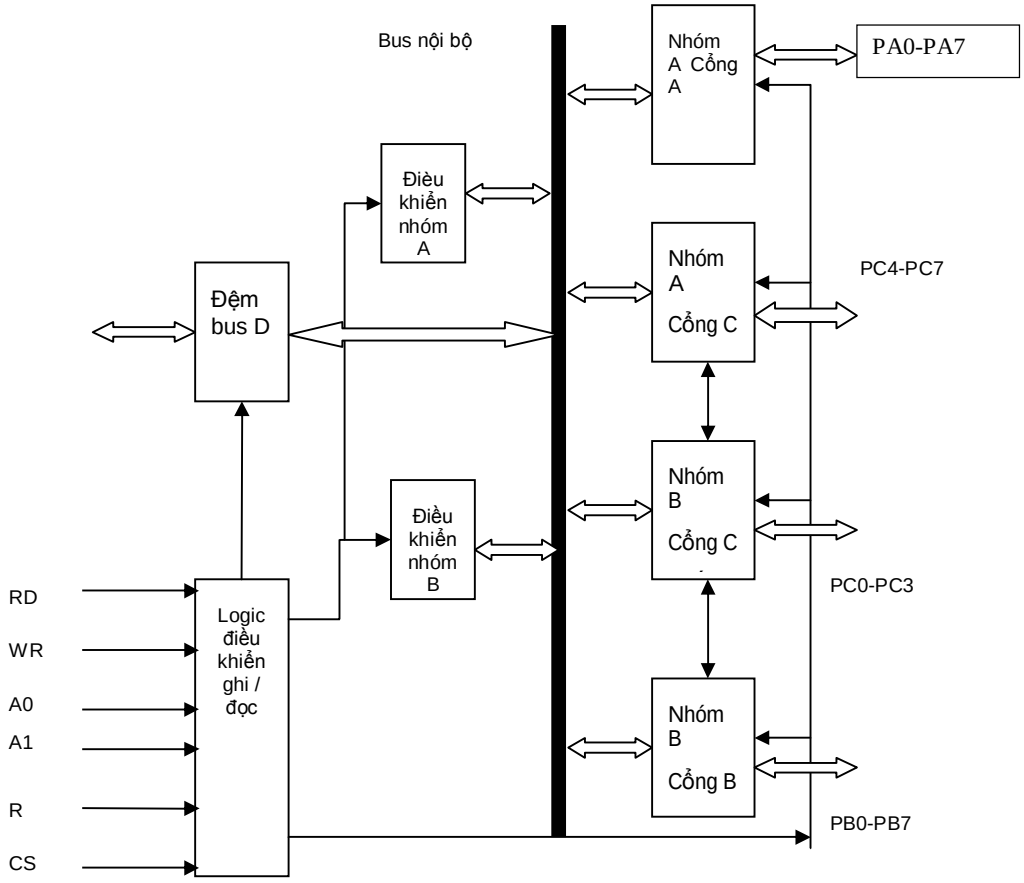
- ✓ Ghép nối giữa CPU và Modul vào ra: song song
- ✓ Ghép nối giữa Modul vào ra và TBNV: song song sẽ tạo ra cổng LPT, nối tiếp sẽ tạo ra cổng COM

4.2.1. 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 inteface)

Mạch ghép nối vào ra song song lập trình được 8255A

**** 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.

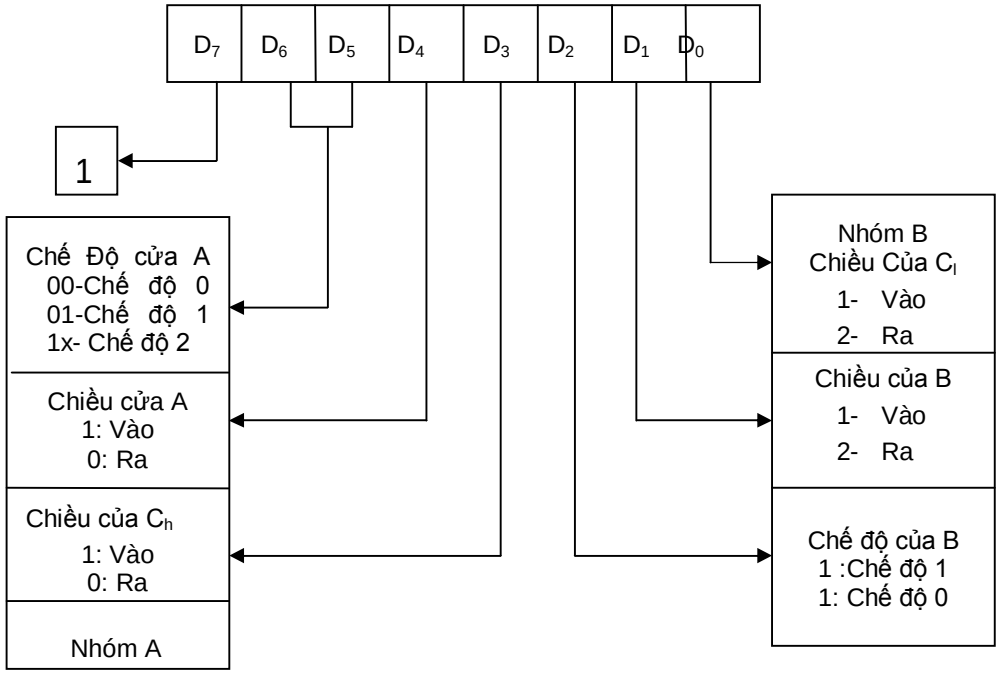
Có hai loại từ điều khiển cho 8255A:

- ✓ Từ điều khiển định nghĩa cấu hình cho các cổng PA,PB,PC.
- ✓ Từ điều khiển lập/xoá 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



**** Các chế độ hoạt động của 8255***

<u>Chế đô 0 (chế đô vào ra cơ sở):</u>

- ✓ Các cổng A, B, C được sử dụng độc lập nhau
- ✓ Các cổng A, B, C có thể là cổng vào hoặc cổng ra tuỳ 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 (PC7- PC4) 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.

**** Ghép nối 8255A với MVT và TBN***

Nguyên tắc chung:

- ✓ Phần ghép nói với máy vi tính (MVT)
- ✓ Phần ghép nối với TBN

**** Lập trình cho 8255A***

4.2.2. 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 kí 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.

4.3. Các phương pháp điều khiển vào ra

4.3.1. Vào ra điều khiển bằng cách thăm dò

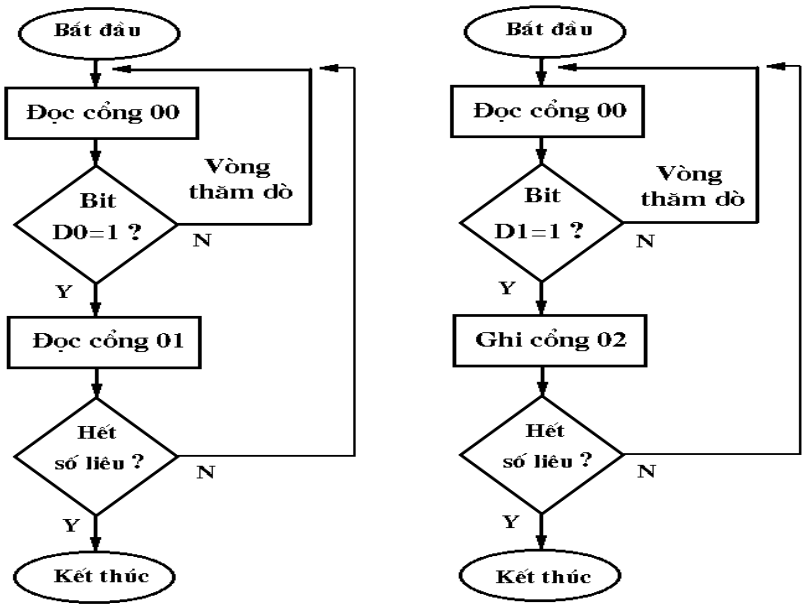
Vấn đề điều khiển vào/ra dữ liệu sẽ trở thành rất đơn giản nếu thiết bị ngoại vi lúc nào cũng sẵn sàng chờ để làm việc với CPU. Ví dụ, bộ phận đo nhiệt số (như là một thiết bị vào) lắp sẵn trong một hệ thống điều khiển lúc nào cũng có thể cung cấp số đo về nhiệt độ của đối tượng cần điều chỉnh, còn một 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 vơi 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 thông 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.



Đọc dữ liệu từ cổng 01 **Ghi dữ liệu ra cổng 02**

4.3.2. Vào ra điều khiển bằng Ngắt

a. 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.

b. Các loại ngắt trong 8088

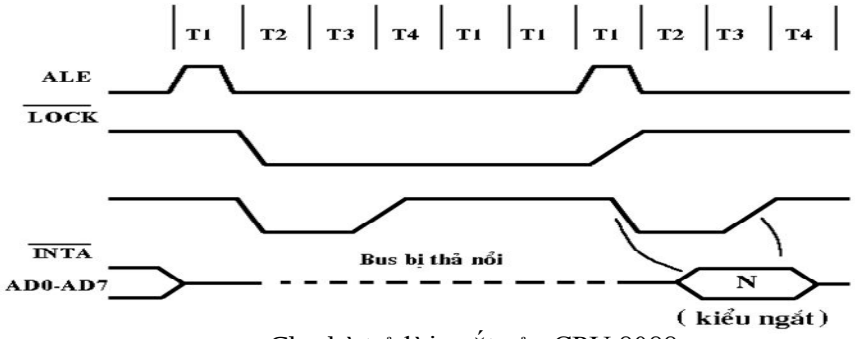
Trong hệ vi xử lý 8088 có thể xếp các nguyên nhân gây ra ngắt CPU vào 3 nhóm sau:

**** 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:***

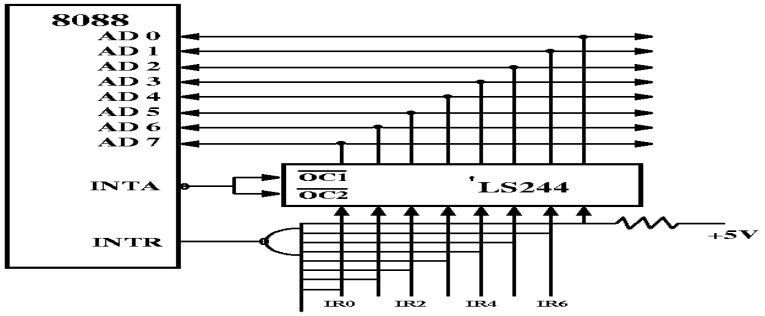
Khi CPU thực hiện các lệnh ngắt dạng INT N, trong đó N là số hiệu (kiểu) ngắt nằm trong khoảng 00-FFH (0-225).

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.



c. Đưa số hiệu ngắt vào Bus dữ liệu

Giả thiết trong một thời điểm nhất định chỉ có một yêu cầu ngắt IRi được tác động và khi đó ở đầu ra của mạch NAND sẽ có xung yêu cầu ngắt đến CPU. Tín hiệu IRi 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 IRi và số hiệu ngắt N

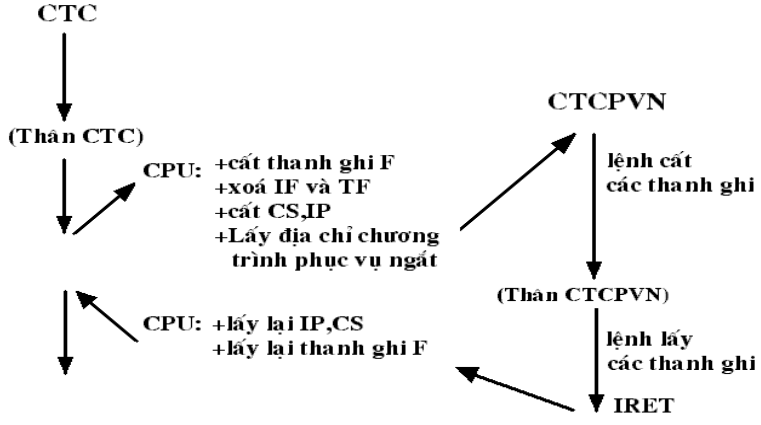
AD7	IR6	IR5	IR4	IR3	IR2	IR1	IR0	N
1	1	1	1	1	1	1	0	FEH(254)
1	1	1	1	1	1	0	1	FDH(253)
1	1	1	1	1	0	1	1	FBH(251)
1	1	1	1	0	1	1	1	F7H (247)
1	1	1	0	1	1	1	1	EFH(239)
1	1	0	1	1	1	1	1	DFH(233)
1	0	1	1	1	1	1	1	BFH(191)

d. 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 ← SP-2, SP ← FR,trong đó SP là ô nhớ do SPchỉ ra (chỉ ra đỉnh mới của ngăn xếp,cất thanh ghi cờ vào đỉnh ngăn xếp)
2. IF ← 0 , TF ← 0.
(cấm các ngắt khác tác động vào CPU,cho CPU chạy ở chế độ bình thường).
3.SP ← SP-2, SP ← CS
(chỉ ra dỉnh mới của ngăn xếp,cất phần địa chỉ đoạn củađịa chỉ trở về vào đỉnh ngăn xếp)
4. SP ← SP-2 , SP ← IP.
(chỉ ra đỉnh mới của ngăn xếp,cất phần địa chỉ lệch của địa chỉ trở về vào đỉnh ngăn xếp).
5. N*4 → IP , N*4+2 → 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 → IP, SP ← SP+2
SP → CS, SP ← SP+2
SP → FR, SP ← 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 CTCPVN 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 CTCPVN.

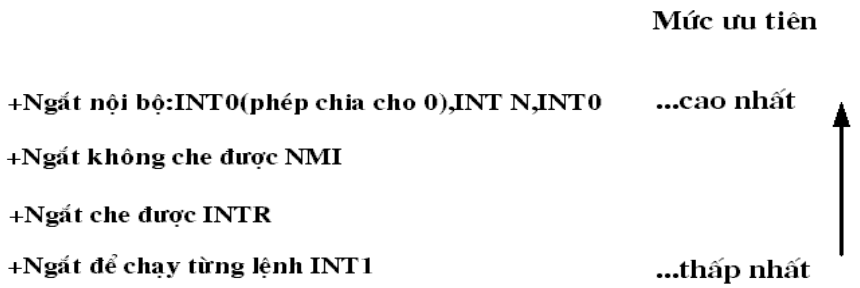
Bảng véctơ ngắt của 8088 tại 1 KB RAM đầu tiên được chỉ ra như hình dưới đây:

03FEH-03FFH	CS Của CTPVN INT FFH
03FCH-03FDH	IP Của CTPVN INT FFH
0082H-0083H	CS Của CTPVN INT 20H
0080H-0081H	IP Của CTPVN INT 20H
000AH-000BH	CS Của CTPVN INT 2
0008H-0009H	IP Của CTPVN INT 2
0006H-0007H	CS Của CTPVN INT1
0004H-0005H	IP Của CTPVN INT1
0002H-0003H	CS Của CTPVN INT0
0000H-0001H	IP Của CTPVN INT0

e. 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 thự 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:



f. Mạch điều khiển ưu tiên ngắt 8259A (PIC: Priority interrupt control)

g. Các khối chức năng chính của PIC: (bổ sung)

4.3.3. 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 khớ 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

* 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 của 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 vủa 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.

* 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.

* 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:

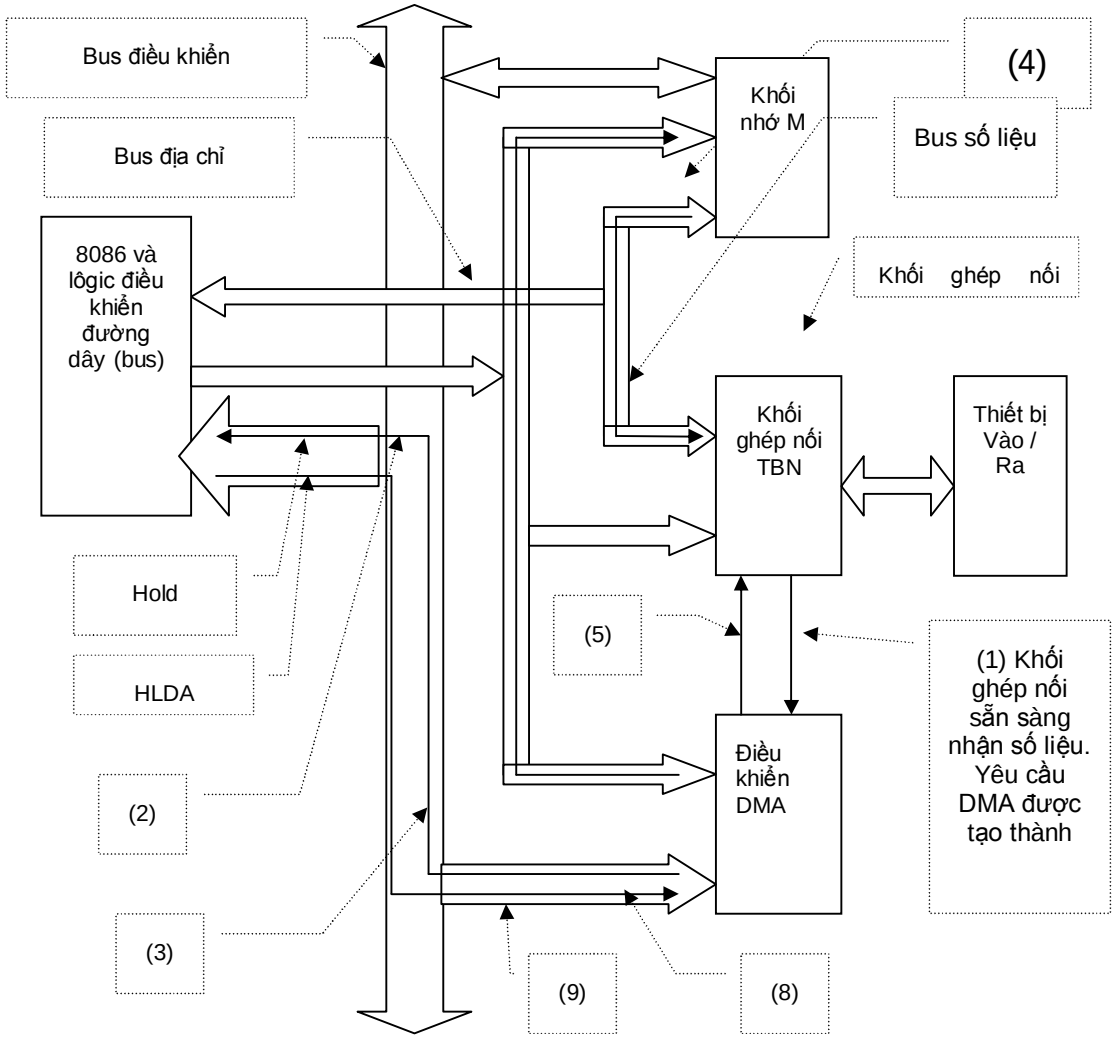
- ✓ Tín hiệu Hold (cho các VXL 8085,8086 ở chế độ MIN)
- ✓ Tín hiệu RQ_0 (ở chế độ MAX của 8086)

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 (GT_0, GT_1 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.

* Chuỗi hành đông 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 tráo 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

c. Các chế độ trao đổi DMA

* Trao đổi tin khối :

Trao đổi nhiều (khối) lời tin lần lượt từ giá trị đếm lời tin n tới 0 (hết)

* 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 Tw tới khi DMAC thực hiện trao đổi xong một phần của trao đổi tin và tiếp tục nốt ở phần lấy lén chu kỳ sau cho tới khi kết thúc trao đổi tin khối tin DMA.

* Lấy lén chu kỳ trong suốt:

Chế độ này cũng giống chế độ trên là lấy lén chu kỳ nhưng bắt VXL chờ với Tw lớn hơn cho tới khi trao đổi xong một khối tin trọn vẹn.

Khối điều khiển DMAC

a. Nhiệm vụ của khối

b. Cấu trúc khối DMAC

c. Vi mạch 8237,8257

CÂU HỎI VÀ BÀI TẬP

4.1. Trình bày vai trò, nhiệm vụ, phân loại khối ghép nối? Vẽ sơ đồ cấu trúc chung của một khối ghép nối?

4.2. Vi mạch điều khiển vào ra bằng chương trình 8255A của một máy tính IBM/PC có địa chỉ cơ sở là 60h. Hãy xác định địa chỉ các cổng PA, PB, PC, thanh ghi từ điều khiển CWR?

4.3. Vi mạch điều khiển vào ra bằng chương trình 8255A của một máy tính IBM/PC có địa chỉ cơ sở là 60h. Hãy xác định giá trị thanh ghi từ điều khiển CWR để 8255A hoạt động ở chế độ vào ra cơ sở với PA, PCh là cổng vào, và với PB, PCl là cổng ra?

4.4. Khái niệm, phân loại về ngắt? Nêu phương pháp điều khiển vào ra bằng ngắt.

4.5. Trình bày khái niệm, các thủ tục trao đổi tin, các chế độ trao đổi DMA.

4.6. Trình bày phương pháp vào ra bằng chương trình? So sánh phương pháp vào ra bằng chương trình và phương pháp vào ra bằng DMA

Chương V: THIẾT BỊ NHẬP DỮ LIỆU

5.1. Giới thiệu chung

5.2. Bàn phím

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:

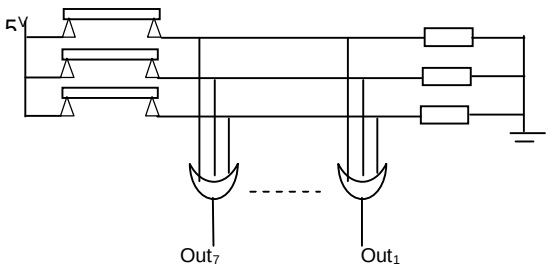
- ✓ Bàn phím ký tự: dùng để đưa chữ cái, chữ số, dấu hiệu và một số ký tự điều khiển vào máy tính.
- ✓ Bàn phím số: chuyên dùng để đưa số vào máy tính. Thường được ghép cùng bàn phím ký tự(bên phải bàn phím hiện đại).
- ✓ Bàn phím đặc nhiệm: dùng cho các thiết bị điều khiển tự động (có sử dụng bộ vi xử lý...).

Công nghệ:

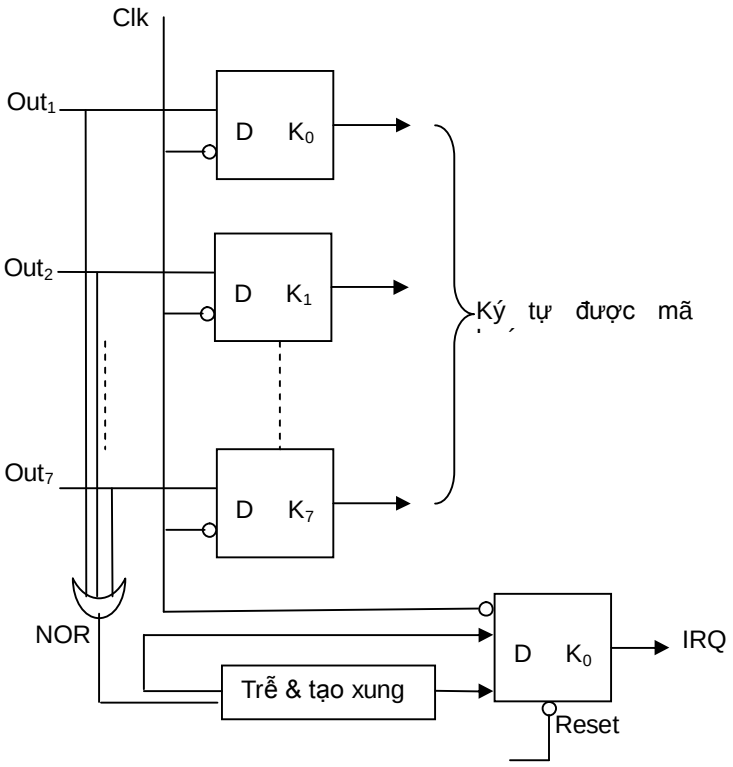
- ✓ Điện dung: hai trạng thái của phím ấn khác nhau về điện dung (hình vẽ)
- ✓ Hiệu ứng Hall (hình vẽ)
- ✓ Bàn phím quang điện
- ✓ Bàn phím lõi Pherit

5.1.1. Kĩ thuật dò phím

Bàn phím được tổ chức thành ma trận, giao của hàng và cột sẽ cho phép xác định phím được nhấn:



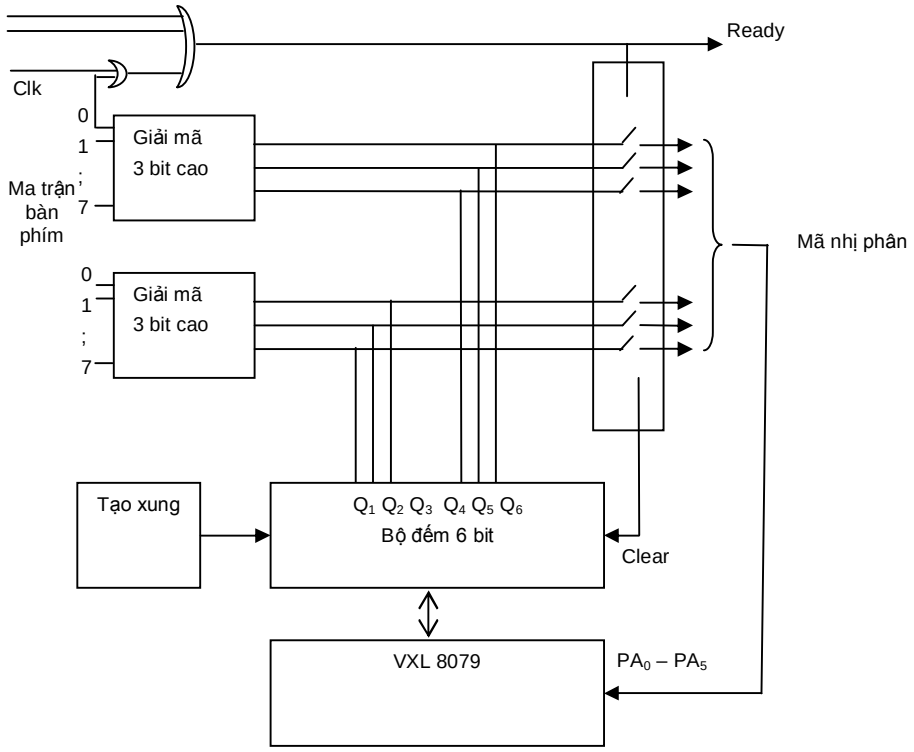
Mạch phối ghép chốt ký tự:



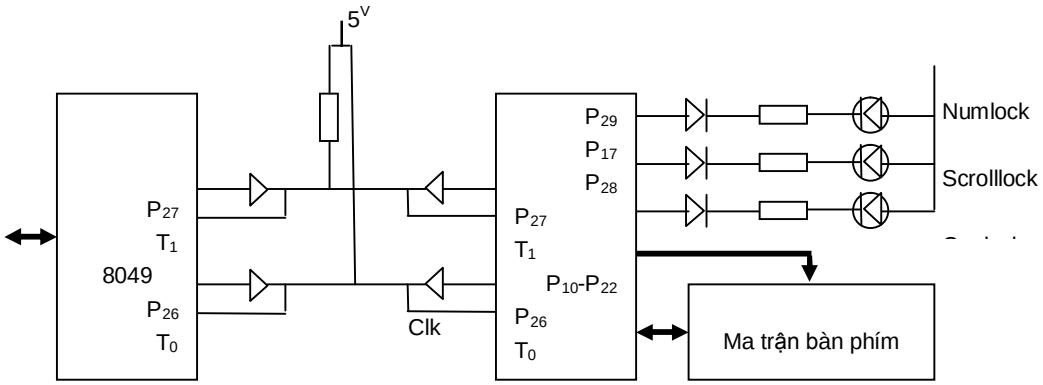
5.1.2. Kĩ thuật quét phím (Scan)

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.

5.3. Chuột

5.4. Các thiết bị nhập liệu tiên tiến

CÂU HỎI VÀ BÀI TẬP

5.1. Trình bày kỹ thuật dò phím.

5.2. Trình bày kỹ thuật quét phím

Chương VI: THIẾT BỊ XUẤT DỮ LIỆU

6.1. 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 hoạ (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.

6.1.1. 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.

6.1.2. 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 tuỳ theo độ đậm nhạt (gray scale). Độ sâu màu là số màu có thể hiển thị được cho một điểm ảnh.

Tuỳ theo số bít 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 tôc độ quét màn hình

f_{lt}=30Hz đến 60Hz.

6.2. 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.

6.2.1. 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 (Pricision In Line)
- ✓ Các chấm sáng theo tam giác đều: Trinitron.

Sơ đồ cấu tạo CRT màu

<u>*Bộ phận súng điện tử*</u>

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

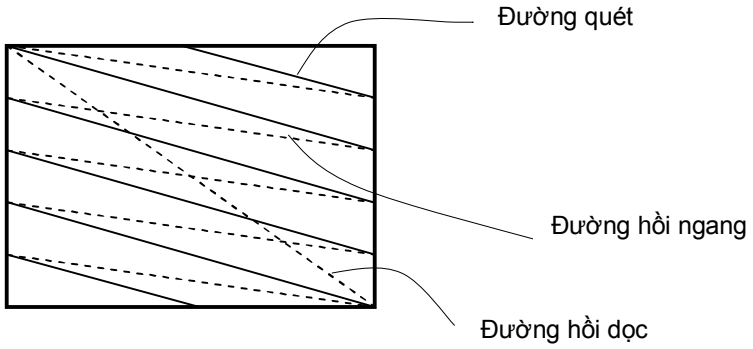
Khi súng điện tử quét lần lượt các dòng của mành, nếu làm thay đổi cường độ của chùm tia điện tử theo qui luật thay đổi của tín hiệu hình ảnh nơi phát đi, thì sẽ tái tạo lại hình ảnh trên màn hình. Việc điều khiển dòng điện tử theo quy luật của hình ảnh cần hiển thị được gọi là điều chế dòng điện tử.

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.

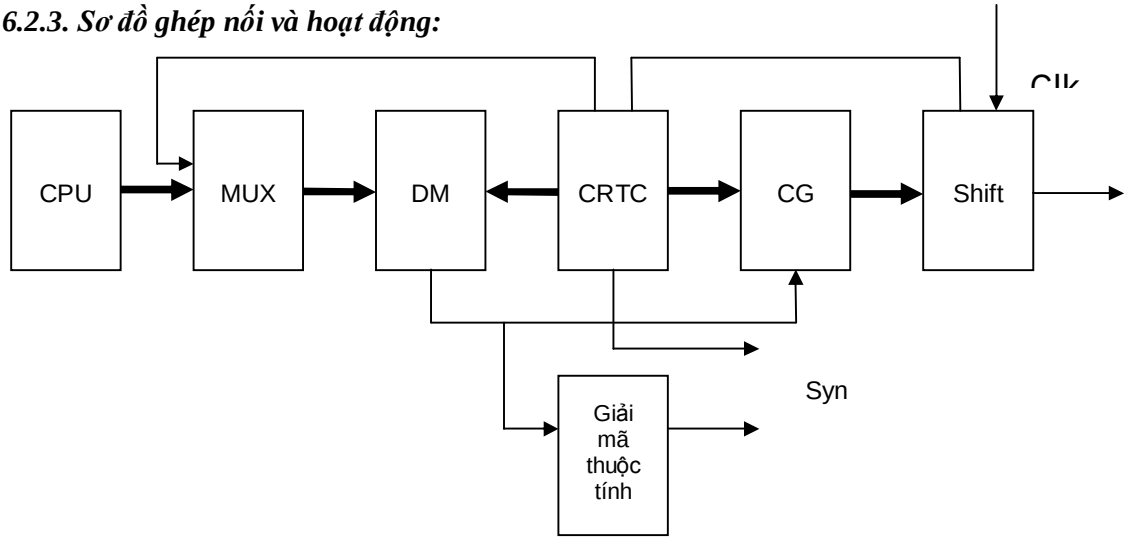
6.2.2. Phương pháp quét dòng

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.



6.2.3. 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 hinhg trong chế độ Text.

CG (Character General): lưu trữ các mẫu bit của kí tự, các Font chữ trong chế độ Text.

Bộ Shift:

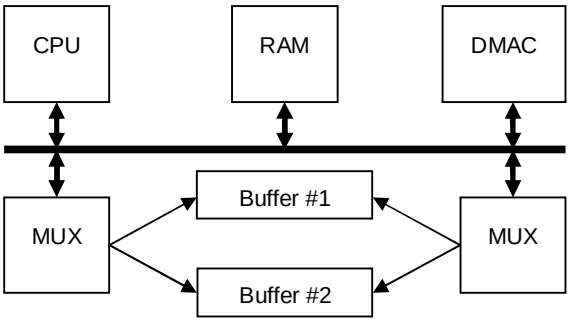
- ✓ Nhận số liệu từ CG và đẩy ra tín hiệu Video
- ✓ Kết hợp với bộ giải mã thuộc tính để tạo ra các tín hiệu cho súng RGB
- ✓ Kết hợp với các tín hiệu đồng bộ từ CRTC để đưa ra tín hiệu hỗn hợp.

6.2.4. 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 lập 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.

6.3. Máy in

CÂU HỎI VÀ BÀI TẬP

6.1. Trình bày các nguyên lý của phương pháp hiển thị hình ảnh video

6.2. Trình bày những đặc điểm chung của màn hình

6.3. Trình bày cấu tạo, nguyên lý hoạt động của màn hình CRT màu

6.4. Trình bày sơ đồ tổng quát, nguyên lý hoạt động của khối ghép nối điều khiển màn hình CRT.

6.5. Trình bày kỹ thuật làm tươi màn hình

Chương VII: THIẾT BỊ LƯU TRỮ

7.1. Giới thiệu chung

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, Memory card…

7.2. Đĩa từ (Magetic)

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

7.2.1. 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:

Qúa 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 trục 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

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.

7.2.2. Tham số đĩa từ

a. 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)

b. Cấu trúc các vùng thông tin trên đĩa

- ✓ Vùng hệ thống
- ✓ Vùng dữ liệu

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

Ý nghĩa	*Độ dài*	*Số hiệu*
Tên tệp, thư mục	8 byte	0
Phần mở rộng	3 byte	1
Thuộc tính	1 byte	2
Để 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 tệp 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:…

72.3. 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) tuỳ theo các chuẩn ATA (AT Attachment)
- ✓ Tốc độ có thể lên tới 4 Mbyte/s

7.2.4. Chuẩn bị một đĩa cứng để đưa vào sử dụng

Đinh dang 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)

Đinh dang cấp cao (High Level Format)

- ✓ Tạo các phân vùng hệ thống, dữ liệu trên đĩa
- ✓ Đánh dấu các vùng đĩa lỗi

*Thực hi*ện: FORMAT

7.3. Đĩa Quang (Optical Disk)

7.3.1. Đặc điểm:

- ✓ Mật độ ghi thông tin cao
- ✓ Dung lượng lớn
- ✓ Giá thành: thấp
- ✓ Tốc độ truy xuất: nhỏ hơn đĩa cứng

7.3.2. Nguyên tắc đọc/ghi thông tin

a. Ghi thông tin:

- 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 chảy 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 trôn 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)

b. Đọc thông tin:

- Đầu dò (Detector): đo năng lượng phản xạ từ bề mặt đĩa khi chiếu lên bề mặt 1 tia lazer công suất nhỏ.
- Dữ liệu được đọc với vận tốc 75inch/s tương đương 153.60 Kbyte/s

7.3.3. 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..

7.4. Các thiết bị lưu trữ khác

CÂU HỎI VÀ BÀI TẬP

7.1. Trình bày các phương pháp mã hóa số liệu trên đĩa từ.

7.2. Trình bày sơ đồ tổng quát của các vùng thông tin trên đĩa từ. Giải thích ý nghĩa, tác dụng của các vùng thông tin.

7.3. Trình bày cấu tạo chung của một đĩa cứng? Các chuẩn giao tiếp đĩa cứng? Các khái niệm về side/head, track,cylinder, sector, cluster?

7.4. Thực hiện thao tác định dạng cấp thấp, định dạng cấp cao của đĩa cứng cụ thể.

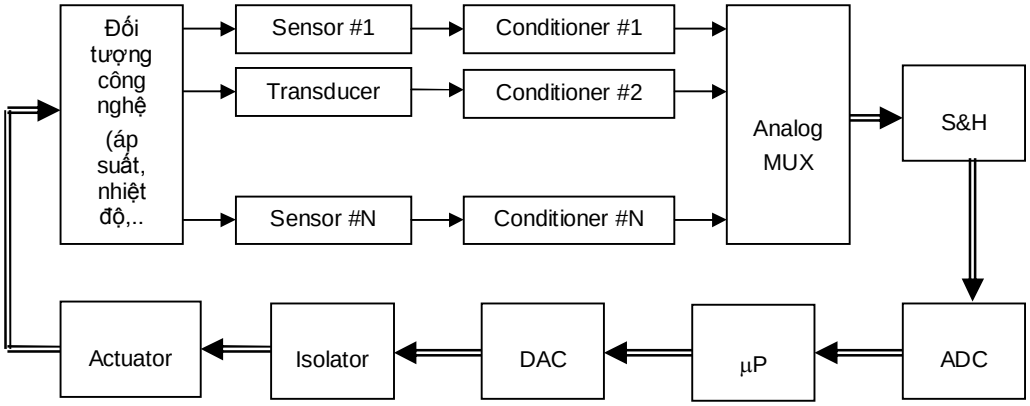
7.5. Trình bày nguyên tắc đọc, ghi thông tin trên đĩa quang.

Chương VIII: THIẾT BỊ GHÉP NỐI VÀ TRUYỀN THÔNG

8.1. Giới thiệu chung

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.

8.2. Bộ chuyển đổi tín hiệu

8.2.1. 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.

$$E_{Out} = E_{in}(B_1.2^{-1} + B_2.2^{-2} + \ldots + B_n.2^{-n})$$

n: số bit

B_i: các bit.

Đặc điểm

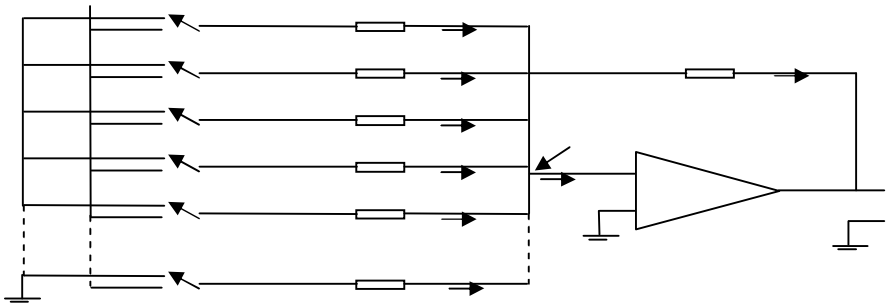
Lượng ra không liên tục

Giá trị ra lớn nhất khi tất cả các bit là 1

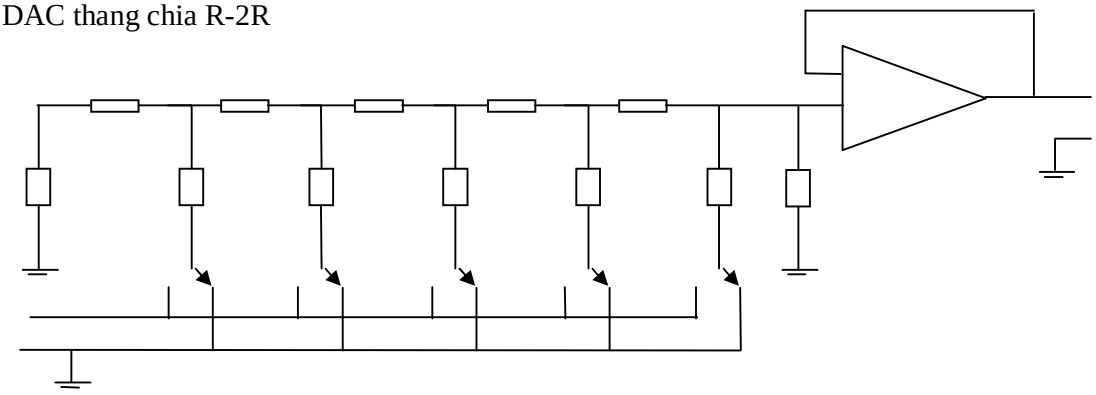
Sự thay đổi của điện áp hay dòng ứng với sự thay đổi của LSB.

Các loại DAC:

DAC thang chia nhị phân



DAC thang chia R-2R



8.2.2. Bộ chuyển đổi tín hiệu tương tự - số: ADC (Analog Digital Converter)

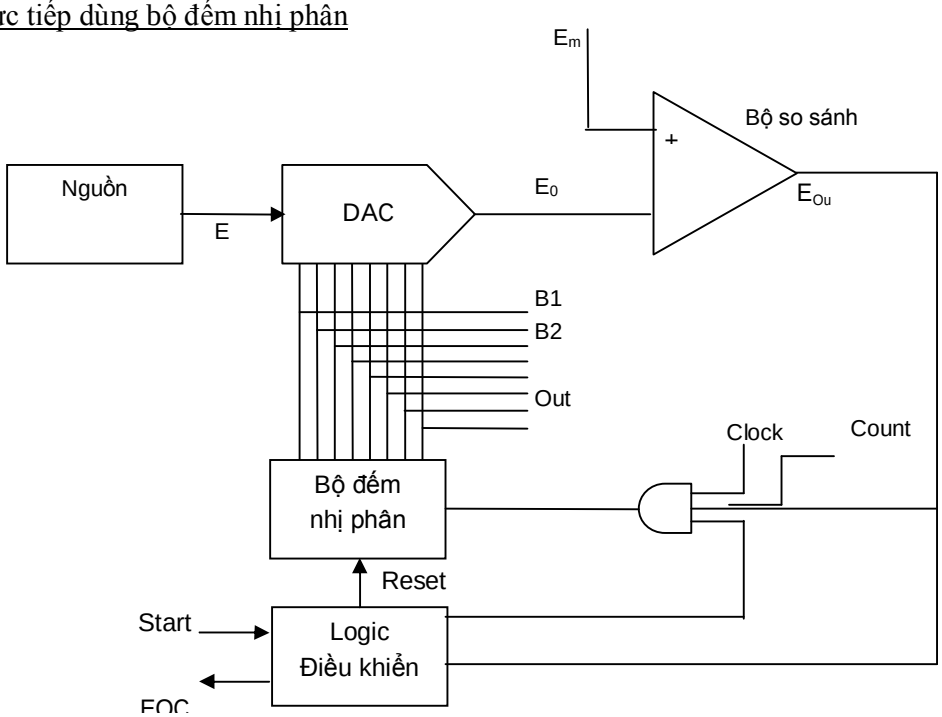
Nguyên tắc: rời rạc hoá và mã hoá tín hiệu

Phân loại:

Trực tiếp: từ giá trị mã điện áp sang mã xung

Gián tiếp: thông qua khâu biến đổi

<u>ADC trực tiếp dùng bộ đếm nhị phân</u>

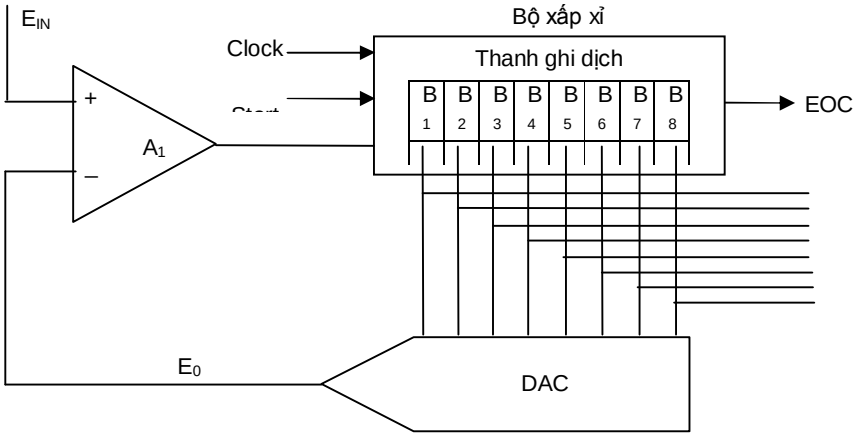


Khi có tín hiệu xoá bộ đếm $E_0 = 0$.

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), xoá bộ đếm

ADC gián tiếp dùng bộ xấp xỉ (Successive approximation)



8.2.3. Modem (Modulation - Demodulation) điều chế và giải điều chế

Thiết bi

- Chuyển đổi tín hiệu: số → tương tự và tương tự → 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 bít 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

- S_0: 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

8.3. Các chuẩn giao tiếp

8.3.1. Các chuẩn chung:

Hầu hết các thiết bị xử lí 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 dã liệu đầu cuối (DTE, DCE).

Mỗi thiết bị xử lí 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

8.3.2. Các chuẩn về giao diện gữa DTE và DCE bao gồm:

- ✓ *Chuẩn về cấu trúc*: xác định kết nối vật lí giữa DTE và DCE (tín hiệu và mạch điều khiển thông qua cáp nối và giắc cắm)
- ✓ *Chuẩn về tín hiệu:* xác địn 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.

8.3.3. Chuẩn EIA-RS 232 (Electronic Industry Association – Recomand 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 tuỳ 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 V đến +25 V. Các mức điện áp trong phạm vi –3 V đến +3 V là trạng thái chuyển tiếp, trong khi các phạm vi từ ±3 V đến ±5 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.

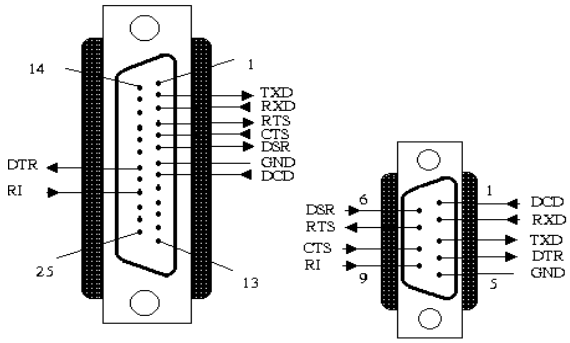
Các đặc trưng điện của tiêu chuẩn RS-232 quy định cụ thể điện áp cực tiểu và cực đại của mức logic '1' và '0'. Mức điện áp bằng 0 V ở bộ nhận, được hiểu như việc đường truyền bị đứt hoặc xảy ra chập mạch.

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 ± 12 V, chứ không dùng giới hạn ±25 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 (-12V).

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 V đến +12 V, khoảng từ + 5 V đến +12 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 cấp 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: =>: Lối vào <=: Lối 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 *(Xoá 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.

8.4. Mạch điều khiển truyền số liệu

8.4.1. Giới thiệu chung

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 dung 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 USART tuỳ 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 ngắt và địa chỉ cổng cơ bản

Cổng	Địa chỉ cơ bản	IRQ
COM1	3F8h	IRQ4
COM2	2F8h	IRQ3
COM3	3F8h	(IRQ4)
COM4	2F8h	(IRQ3)

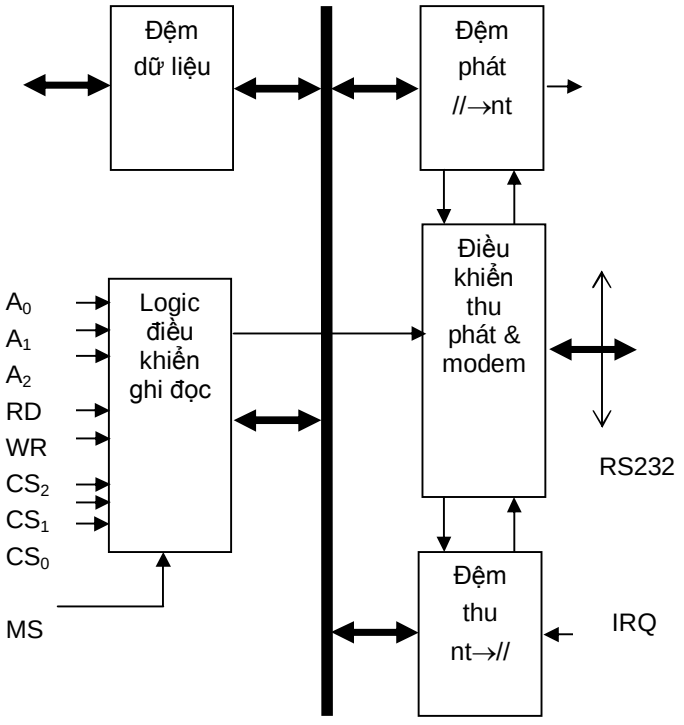
Địa chỉ cơ sở và ngắt của các cổng nối tiếp.

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

8.4.2. 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 vỉ 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. Tuỳ 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	Trạn 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.

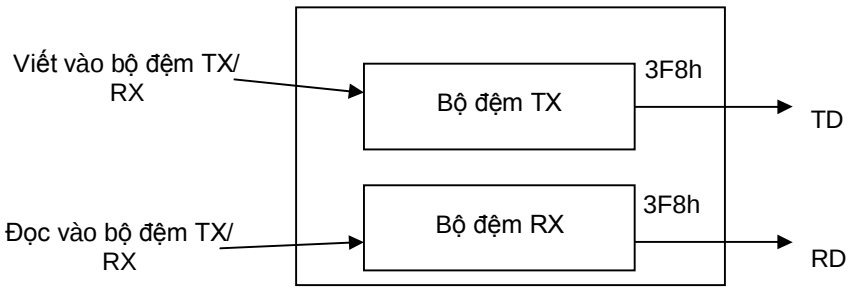
Vi mạch UART 8250 có tất cả 10 thanh ghi, sau đây ta sẽ lần lượt tìm hiểu các thanh ghi này:

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.



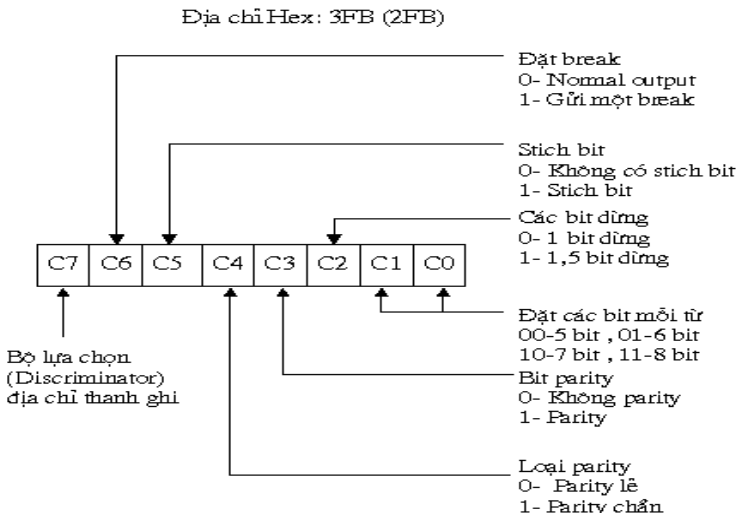
Đọc ra và ghi vào từ bộ đệm TX/ RX.

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 đấy. Chức năng các bit của thanh ghi LCR.

- Các bit 0 và 1. Giá trị được cất giữ trong hai bit nhi 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 dùng 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 Boud

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.

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

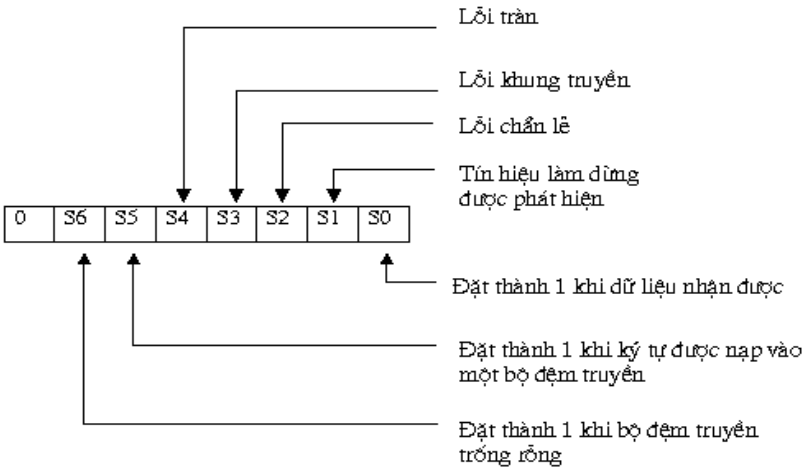
với hai chốt số chia. Các chốt số chia bao gồm 16 bit hay hai byte, được sắp xếp thành các bit có giá trị thấp LSB (Least Significant byte) và bit có giá trị cao hơn MSB (Most Significant bit), được sử dụng trong việc đặt tốc độ baud của hệ thống truyền thông.

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).

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 đệm 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.



- 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 dừng 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 Resgister) 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.

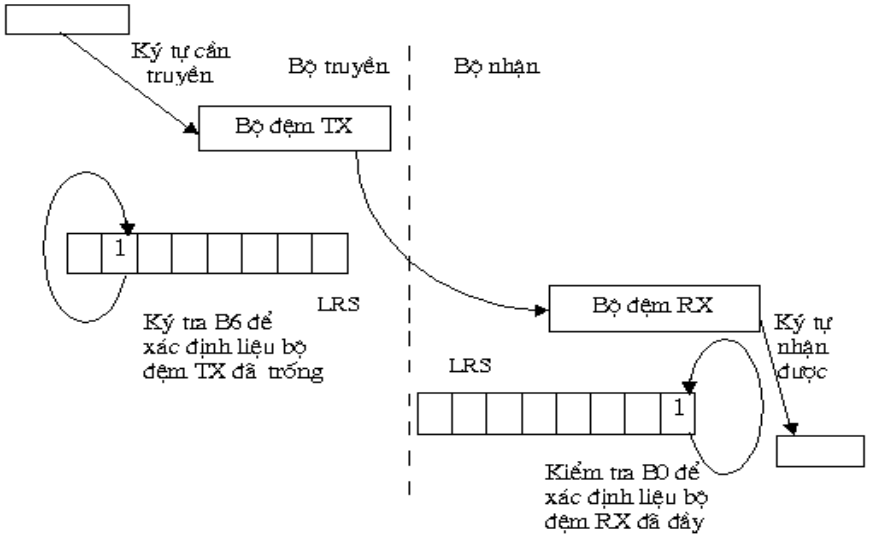
Một điều rủi ro có thể xảy ra khi truyền dữ liệu một ký tự mới có thể được viết vào bộ đệm truyền trước khi ký tự trước đấy đã được gửi. Khi đó ký tự mới này sẽ viết đè lên nội dung của ký tự đang được truyền. Để tránh tình trạng rủi ro này S5 được giao nhiệm vụ thông báo kết quả kiểm tra xác định liệu 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;)*



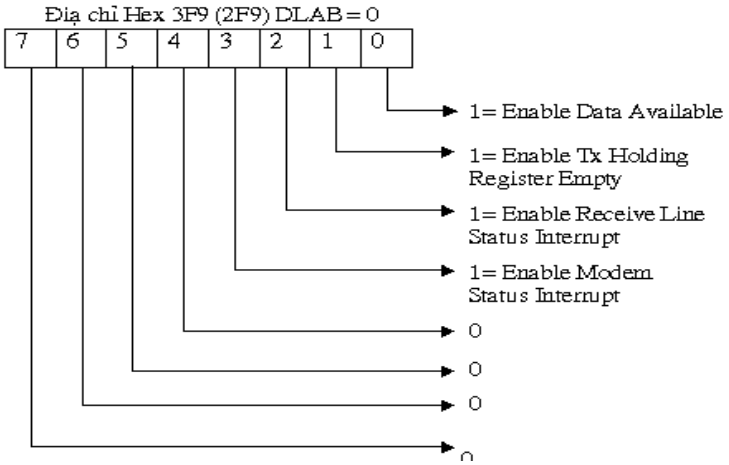
Kiểm tra thanh ghi LSR để truyền và nhân các ký tự.

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).

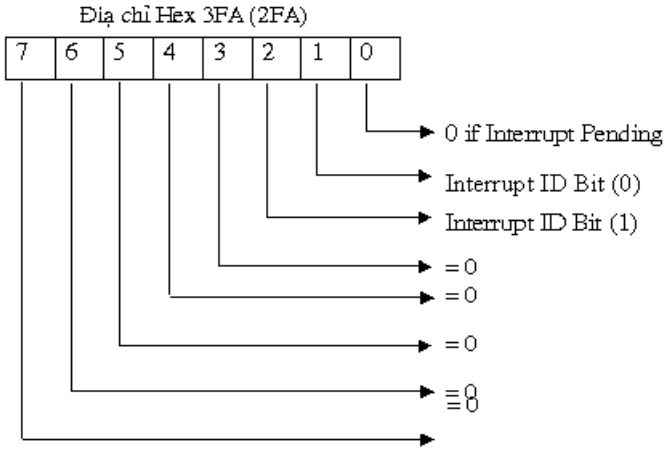
Nếu như khả năng ngắt của vi mạch đã cho phép và một ngắt xuất hiện thì bit xuất ra ngắt từ 8250 chiếm lấy mức logic 1. Tín hiệu này được nối với bus ngắt cứng của máy tính. Logic 1 trên bus này báo hiệu cho bộ xử lý biết là cần phải chú ý tới cổng nối tiếp. Hình 15 minh hoạ sự phân bố của các bit trên 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 chẵn 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

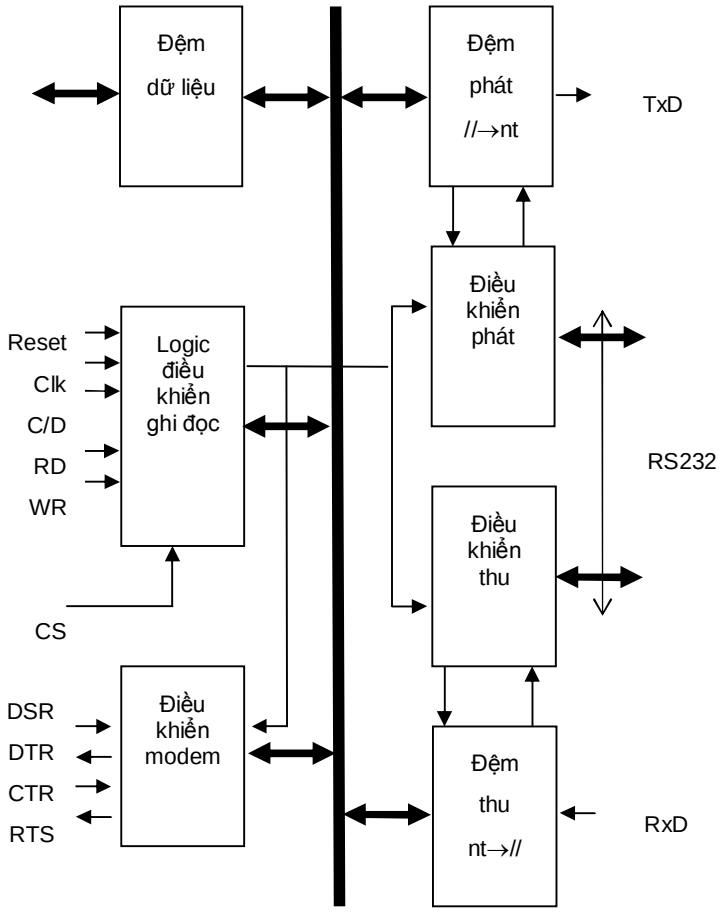
Các mức ưu tiên của từng ngắt.

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.

8.4.3. Mạch điều khiển truyền thông đồng bộ - dị bộ vạn năng USART (VXL 8251A)

Vi mạch 8251A là một USART được dùng rộng rãi trong các máy IBM PC tại vỉ 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 la 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 đè
- Bit 5: Lỗi Frame
- Bit 6: Kí tự đồng bộ
- Bit 7: Modem sẵn sàng

CÂU HỎI VÀ BÀI TẬP

8.1. Trình bày hiểu biết của anh, chị về bộ biến đổi ADC, DAC.

8.2. Trình bày hiểu biết của anh, chị về Modem

8.3. Trình bày ý nghĩa các bit của thanh ghi điều khiển đường truyền trong mạch điều khiển truyền thông dị bộ vạn năng UART – 8255A

8.4. Trình bày ý nghĩa các bit của thanh ghi trạng thái đường truyền trong mạch điều khiển truyền thông dị bộ vạn năng UART – 8255A

8.5. Trình bày ý nghĩa các bit của thanh ghi cho phép ngắt trong mạch điều khiển truyền thông dị bộ vạn năng UART – 8255A

8.6. Trình bày ý nghĩa các bit của thanh ghi nhận dạng ngắt trong mạch điều khiển truyền thông dị bộ vạn năng UART – 8255A

8.7. Trình bày ý nghĩa các bit của thanh ghi từ chế độ trong mạch điều khiển truyền thông đồng bộ, dị bộ vạn năng USART – 8251A

8.8. Trình bày ý nghĩa các bit của thanh ghi từ lệnh trong mạch điều khiển truyền thông đồng bộ, dị bộ vạn năng USART – 8251A

8.9. Trình bày ý nghĩa các bit của thanh ghi từ trạng thái trong mạch điều khiển truyền thông đồng bộ, dị bộ vạn năng USART – 8251A

8.10. Xây dựng chương trình đọc và thiết lập các thông số cho các thanh ghi của cổng COM

8.11. Xây dựng chương trình truyền dữ liệu qua cổng COM giữa hai máy tính

ĐỀ THI THAM KHẢO

Đề 1: ***(Thời gian làm bài 75 phút)***

1. Trình bày sơ đồ cấu trúc, chức năng nhiệm vụ của bộ xử lý trung tâm (Nêu rõ chức năng của từng đơn vị)
2. Cho năm lệnh i1, i2, i3, i4, i5 mỗi lệnh cần bốn giai đoạn FI, DI, EI, WB. Mỗi giai đoạn cần T0 đơn vị thời gian. Tính thời gian thực hiện năm lệnh trên cho các trường hợp
   a. Pipeline
   b. Superpipelined
   c. Superscalar
3. Trình bày phương pháp vào ra bằng chương trình? So sánh phương pháp vào ra bằng chương trình và phương pháp vào ra bằng DMA
4. Thiết kế bộ nhớ có dung lượng 64 Kbytes hoạt động trong khoảng địa chỉ F0000h – FFFFFh (Sử dụng mạch giải mã LS138 và EPROM 2Kb)

Đề 2: ***(Thời gian làm bài 75 phút)***

1. Đổi số 42,00125 (trong hệ thập phân) sang số dạng IEEE 754.
2. Trình bày ý nghĩa tác dụng của bộ nhớ ảo ? Kỹ thuật phân trang trong quản lý bộ nhớ?
3. Cho đoạn vi lệnh sau: T1: MBR←(PC)
   T2: MAR←Save-Address
   PC←Routine-Address
   Anh (chị) hãy giải thích từng vi lệnh trong đoạn vi lệnh trên và nó tương ứng với chu kỳ nào trong quá trình xử lý lệnh?
4. Thiết kế bộ nhớ dung lượng 32 KBytes hoạt động trong khoảng địa chỉ F4000h÷ FBFFFh (sử dụng mạch giải mã LS138 và EPROM 4k*8)

Đề 3: ***(Thời gian làm bài 90 phút)***

1. Đổi số thực —5,0315 (trong hệ thập phân) sang dạng IEEE 754.
2. Trình bày kiến trúc chung của máy tính theo nguyên lý VonNewman? Nêu chức năng từng đơn vị? Phân loại máy tính theo kiến trúc?
3. Cho một lệnh máy: **SUB R,X**
   Lệnh này lấy nội dung của thanh ghi R trừ đi nội dung của ô nhớ X. Hãy chỉ ra các vi lệnh mà Bộ điều khiển (Control Unit) phải thực hiện.
4. Cho hệ thống máy tính với không gian địa chỉ bộ nhớ chính là 4 GBytes, dung lượng bộ nhớ Cache là 512 KBytes, kích thước đường là 64 byte, tìm dạng địa chỉ truy nhập Cache cho 3 trường hợp:
   a. Ánh xạ địa chỉ trực tiếp
   b. Ánh xạ địa chỉ liên kết hoàn toàn
   c. Ánh xạ địa chỉ liên kết tập hợp 2 đường.
5. Thiết kế bộ nhớ dung lượng 32 KBytes hoạt động trong khoảng địa chỉ F8000h÷ FFFFFh (Sử dụng mạch giải mã LS138 và EPROM 1k*8)

Đề 4: ***(Thời gian làm bài 90 phút)***

1. Đổi số 428DA9FC*h* (theo chuẩn IEEE 754) sang số thực hệ thập phân
2. Trình bày vai trò, nhiệm vụ, phân loại khối ghép nối? Vẽ sơ đồ cấu trúc chung của một khối ghép nối?
3. Giả sử có một lệnh máy như sau: **ISZ x**

Lệnh này tăng nội dung ô nhớ lên 1. Nếu kết quả là 0 thì lệnh tiếp theo bị bỏ qua. Hãy xác định dãy các vi lệnh mà bộ điều khiển (Control Unit) phải thực hiện để xử lý lệnh trên.

4. Vi mạch điều khiển vào ra bằng chương trình 8255A của một máy tính IBM/PC có địa chỉ cơ sở là 60h. Hãy xác định:
   - Địa chỉ các cổng PA, PB, PC, thanh ghi từ điều khiển CWR?
   - Giá trị thanh ghi từ điều khiển CW để 8255A hoạt động ở chế độ vào ra cơ sở với PA, PC_h là cổng vào, và với PB, PC_l là cổng ra?
5. Thiết kế bộ nhớ dung lượng 64 KBytes hoạt động trong khoảng địa chỉ F0000h÷FFFFFh (sử dụng mạch giải mã LS138 và EPROM 4k*8)

Đề 5: ***(Thời gian làm bài 90 phút)***

1. Đổi số thực 4,625 (trong hệ thập phân) sang dạng IEEE 754.
2. So sánh hai phương pháp vào ra theo ngắt và vào ra truy nhập bộ nhớ trực tiếp DMA.
3. Cho một lệnh ở ngôn ngữ cấp cao: **Y=(A+B)*(C+D/E)**
   Sử dụng các lệnh máy hai địa chỉ để mã hóa lệnh cấp cao trên. Mỗi lệnh máy cần năm giai đoạn FI: tải lệnh, DI: giải mã lệnh, FO: tải toán hạng, EI: xử lý lệnh, WB: ghi lại kết quả. Tính thời gian thực hiện các lệnh máy ở trên, với giả thiết mỗi giai đoạn cần T0 đơn vị thời gian.
4. Cho hệ thống máy tính với không gian địa chỉ bộ nhớ chính là 4 G bytes, dụng lượng bộ nhớ Cache là 256 KBytes, kích thước đường là 64 byte, tìm dạng địa chỉ truy nhập Cache cho 3 trường hợp:
   a. Ánh xạ địa chỉ trực tiếp
   b. Ánh xạ địa chỉ liên kết hoàn toàn
   c. Ánh xạ địa chỉ liên kết tập hợp 4 đường.
5. Thiết kế bộ nhớ dung lượng 32KBytes hoạt động trong khoảng địa chỉ F0000h÷F7FFFh (sử dụng mạch giải mã LS138và EPROM 2k*8)

GỢI Ý LÀM BÀI:

Đề 1:

Câu 1:

- Những giai đoạn thực hiện lệnh của CPU
- Sơ đồ cấu trúc tổng quát của CPU
- Chức năng nhiệm vụ của từng thành phần CU, ALU, Register...

Câu 2:

Dựa vào các giai đoạn thực hiện lệnh, các công nghệ xử lý CPU

Câu 3

- Nêu phương pháp vào ra bằng chương trình
- So sánh kỹ thuật, ưu nhược điểm của 2 phương pháp vào ra bằng chương trình, vào ra bằng ngắt

Câu 4: Tham khảo ví dụ về thiết kế bộ nhớ

Đề 2:

Câu 1: Tham khảo ví dụ đổi số

Câu 2: Nêu ý nghĩa tác dụng của bộ nhớ ảo, Trình bày kỹ thuật phân trang

Câu 3:

Câu 4: Tham khảo ví dụ về thiết kế bộ nhớ

Đề 3:

Câu 1: Tham khảo ví dụ đổi số

Câu 2:

- Trình bày sơ đồ cấu trúc chung, giải thích chức năng của từng đơn vị CU, ALU…
- Phân loại kiến trúc máy tính: tuần tự, song song

Câu 3

- Dựa vào các vi lệnh cuả CPU để giải thích và xác định vi lệnh cần thực hiện

Câu 4: Tham khảo ví dụ về xác định từ điều khiển của CPU

Câu 5: Tham khảo ví dụ về thiết kế bộ nhớ

Đề 4 + 5: Tham khảo đề 3