

KẾ HOẠCH GIẢNG DẠY MÔN HỌC

Bộ môn: **Tự động & Kỹ thuật tính**
 Khoa: **Kỹ thuật điều khiển**
 Học kỳ: 2
 Năm học: 2017-2018

Môn học: **Cấu trúc máy tính**
 Ký hiệu môn học: 04DU
 Dùng cho lớp: TLPK, TH
 Giáo viên phụ trách chính: **Nguyễn Trần Hiệp**
 Các giáo viên tham gia: **Nguyễn Văn Xuân**

Tổng số tiết: **60**
 Bài giảng: 52
 Bài tập: 8
 Thí nghiệm:
 Thảo luận:

Số học sinh:

Hình thức khác:

Số TT bài	Số TT đề mục	Hình thức huấn luyện	Tên gọi các phần, các đề mục	Số tiết	Phòng học chuyên dùng hoặc phòng thí nghiệm chuyên dùng	Giáo viên phụ trách	Ngày tháng
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	1.1	Bài giảng 1-4 Tuần 1	Tổng quan về cấu trúc máy tính Định nghĩa máy tính điện tử Biểu diễn thông tin trên máy tính Lịch sử phát triển Cấu trúc chung của máy tính Cấu trúc theo nguyên lý Von-Neumann Hệ thống Bus kết nối các thành phần - Bus địa chỉ, độ rộng bus địa chỉ, - Bus dữ liệu, độ rộng bus dữ liệu, - Bus điều khiển, - Thao tác ghi/đọc Phân loại các hệ máy tính	4	Phòng học chuyên dùng		
2	1.2	Bài giảng 5-8 Tuần 2	Cấu trúc chung của BVXL Giới thiệu chung - Bố trí các nhom chân của BVXL Tổ chức BVXL - Tổ chức của EU, BIU, - Địa chỉ vật lý, - Địa chỉ logic, Các chế độ làm việc	4	Phòng học chuyên dùng		

			<ul style="list-style-type: none"> - Chế độ thực, - Chế độ bảo vệ, <p>Giản đồ thời gian hoạt động của BVXL</p> <p>Các bộ vi xử lý tiên tiến của Intel</p> <p>Nguyên tắc xây dựng BVXL của Intel</p> <p>Các thế hệ BVXL của Intel</p> <p>Công nghệ Clocking double</p> <p>Kiến trúc đường ống</p> <p>Kiến trúc Superscalar</p> <p>Công nghệ siêu phân luồng</p> <p>Cấu trúc Dual Core Processors</p> <p>Cấu trúc Chip Set</p> <ul style="list-style-type: none"> - North and South Bridge 			
3	1.3	Bài giảng 9-12 Tuần 3	<p>Hệ thống hỗ trợ BUS</p> <p>Mạch tạo dao động 8224</p> <p>Mạch điều khiển BUS 8288</p> <p>Hệ thống hỗ trợ Bus</p> <ul style="list-style-type: none"> - Mạch chốt địa chỉ và đệm dữ liệu - Mạch giải mã <p>Nguyên tắc của giải mã.</p> <p>Mạch giải mã bộ nhớ</p> <p>Mạch giải mã các cổng vào/ra</p> <ul style="list-style-type: none"> - Giản đồ thời gian chu kỳ ghi/đọc <p>Các chế độ làm việc của BUS</p> <p>Các hệ Bus tiêu chuẩn</p> <ul style="list-style-type: none"> - Khái niệm về hệ bus tiêu chuẩn, <ul style="list-style-type: none"> + Thông lượng của Bus + Giao thức Bus + Tự động định dạng cấu hình - Bus ISA, MCA, EISA, PCI 	4	Phòng học chuyên dùng	
4	1.4	Bài giảng 13-20 Tuần 4,5	<p>Lập trình ASM và Tập lệnh của BVXL 8086</p> <p>Lập trình hợp ngữ</p> <p>Cấu trúc file *.com và *.exe</p> <p>Cú pháp câu lệnh hợp ngữ</p>	8		

			Các kiểu dữ liệu trong ASM Giới thiệu tập lệnh của BVXL 80x86 Các nhóm lệnh Chuyển dữ liệu Lệnh số học Lệnh Logic Lệnh xử lý chuỗi Lệnh chuyển điều khiển Các lệnh vào/ra Mô tả các cấu trúc lập trình bậc cao dùng hợp ngữ. Các ví dụ Giới thiệu EMU 8086 Giới thiệu chương trình mô phỏng Proteus			
5	2.1	Bài giảng 21-24 Tuần 6	Bộ nhớ và hệ thống hỗ trợ bộ nhớ Định nghĩa Bộ nhớ, dung lượng nhớ Các loại bộ nhớ (ROM, RAM) Bản đồ bộ nhớ trên máy tính IBM PC Tổ chức giải mã bộ nhớ - Giải mã tìm vi mạch nhớ - Giải mã tìm ô nhớ Bộ nhớ trên các máy tính hiện đại Khái niệm bộ nhớ mở rộng, bộ nhớ phát triển, Vùng nhớ cao, nạp HĐH lên vùng nhớ cao.	4	Phòng học chuyên dùng	
6	2.2	Bài giảng 25-32 Tuần 7,8	Hệ thống hỗ trợ Vào/Ra Định nghĩa quá trình vào/ra Các hình thức vào ra - Vào ra sử dụng không gian riêng - Vào ra sử dụng không gian bộ nhớ Các lệnh vào ra sử dụng không gian riêng Nối ghép vào/ra - Vào ra bằng vi mạch đơn giản Mạch vào/ra song song sử dụng	8	Phòng học chuyên dùng	

			<p>8255</p> <ul style="list-style-type: none"> - Chức năng của của 8255, - Nối ghép với Bus, - Nối ghép với ngoại vi, - Các thanh ghi bên trong của 8255 - Lập trình các chế độ làm việc cho 8255 <p>Vào/ra song song trên máy tính</p> <ul style="list-style-type: none"> - Vào/ra song song trên Board, - Vào/ra song song trên cổng LPT. 			
7		Bài giảng 32-36 Tuần 10	<p>Tổ chức một hệ ghép nối Máy tính với bộ nhớ và vào/ra</p> <ul style="list-style-type: none"> - Mạch quét Led - Giao tiếp với ADC nhiều kênh - Mạch bàn phím 	4		
7	2.3	Bài giảng 32-36 Tuần 9	<p>Truyền dữ liệu nối tiếp</p> <ul style="list-style-type: none"> - Định nghĩa truyền dữ liệu nối tiếp - Truyền đơn công, song công, - Thiết bị đầu cuối, thiết bị truyền dữ liệu - Chuẩn giao thức RS -232 <p>Mạch và/ra nối tiếp sử dụng USART 8250</p> <ul style="list-style-type: none"> - Nối ghép 8250 với Bus, - Các thanh ghi bên trong của 8250, - Lập trình truyền số liệu nối tiếp thông qua 8250, - Lập trình truyền số liệu nối tiếp thông qua ngắt int 14h, <p>Giao diện tuần tự đa năng USB</p> <ul style="list-style-type: none"> - Truyền thông sử dụng Bộ khuếch đại vi sai, - Giao thức truyền thông trên USB, - Các tính năng và chuẩn USB, - Các chế độ truyền thông của USB 	4	Phòng học chuyên dùng	
13		Bài tập	Chữa bài tập và kiểm tra giữa kỳ	4	Phòng học	

		40-44 Tuần 15		chuyên dùng		
8	2.4	Bài giảng 37-40 Tuần 10	<p>Quản lý quá trình vào/ra - Phương pháp hỏi vòng, - Phương pháp ngắn,</p> <p>Quản lý vào ra bằng phương pháp ngắn - Khái niệm ngắn, - Số hiệu ngắn, - Chương trình phục vụ ngắn - ISR - Cấu trúc bảng vector ngắn - Cấu trúc con trỏ ngăn xếp</p> <p>Mạch điều khiển ngắn 8259 - Chức năng của mạch điều khiển ngắn, - Nối ghép 8259 với Bus và ngoại vi, - Cấu trúc các thanh ghi ICW1-ICW4 - Cấu trúc các thanh ghi OCW1-OCW3</p> <p>Mạch điều khiển ngắn trên máy tính hiện đại.</p> <p>Mạch thời gian 8353</p>	4	Phòng học chuyên dùng	
9	3.1	Bài giảng 41-44 Tuần 11	<p>Màn hình và Card điều khiển màn hình</p> <p>Tổ chức màn hình</p> <p>Tổ chức Card điều khiển màn hình</p> <p>Làm việc chế độ văn bản</p> <p>Làm việc chế độ đồ họa</p> <p>Quan hệ giữa độ phân giải, số bits mã hóa màu và bộ nhớ Video RAM</p> <p>Lập trình các chế độ cho màn hình</p> <p>Card tăng tốc đồ họa AGP, PCI Express</p> <p>Bài tập về lập trình</p>	4	Phòng học chuyên dùng	
10	3.2	Bài giảng 45-48	<p>Truy cập bộ nhớ trực tiếp DMA</p> <ul style="list-style-type: none"> - Khái niệm về DMA, - Nối ghép Vi mạch 8237 với bus 	4	Phòng học chuyên dùng	

		Tuần 12	<p>- Cấu trúc các thanh ghi của 8237</p> <p>Mạch điều khiển DMA trên máy tính hiện đại.</p> <p>Các thiết bị ngoại vi điển hình</p> <p>Bàn phím</p> <ul style="list-style-type: none"> - Quét bàn phím, mã quét bàn phím - Nối ghép bàn phím với bảng mạch - Cấu trúc con trỎ đầu, con trỎ đuôi - Lập trình bàn phím <p>Ô đĩa</p> <ul style="list-style-type: none"> - Tô chức đĩa mềm, đĩa cứng, đĩa CD - Cấu hình và định dạng đĩa - Cấu trúc bảng FAT, NTFS 			
13		Bài tập 43-45 Tuần 15	Ôn tập cuối môn học	4	Phòng học chuyên dùng	
Tổng số giờ				60		

Các đề nghị đặc biệt:

Ngày 23 tháng 10 năm 2017
Người lập kế hoạch

Ngày 25 tháng 10 năm 2017
CN Bộ môn thông qua

Nguyễn Trần Hiệp

Tài liệu tham khảo

TT	Tên giáo trình tài liệu	Có ở thư viện	Tham khảo trên mạng hoặc mua	Giáo viên cung cấp
1	Cấu trúc máy tính tập 1 - HVKTQS	x		
2	Cấu trúc máy tính tập 2 - HVKTQS	x		
3	www.karbosguide.com/books/pcarchitecture/chapter00.htm		x	
4	Kỹ thuật vi xử lý và lập trình Assembler – HVKTQS	x		
5	Tập lệnh Bộ VXL Intel x86		x	
6	Lập trình mô phỏng BVXL x86 Emu8086.com		x	
7	Phần mềm thiết kế và mô phỏng Proteus		x	
8	Tập bài giảng Cấu trúc máy tính Đề cương chi tiết Đề cương ôn tập			x
9	https://www.slideshare.net/congthanhbkit03/basic-assembly (Hướng dẫn lập trình ASM)		x	

≥

**BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn**

ĐỀ CƯƠNG BÀI GIẢNG

GIÁO VIÊN

Truong Đăng Khoa

Nguyễn Trần Hiệp

Bài giảng: Tổng quan về cấu trúc máy tính

Chương 1 mục 1.1;

Tiết thứ: 1-4

Mục đích, yêu cầu: Cấu trúc máy tính theo Nguyên lý Von Neumann

- Hình thức tổ chức dạy học:

Lý thuyết, bài tập, tư học, tư nghiên cứu

- Thời gian:

Lý thuyết: 4 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 8 tiết

- Địa điểm:

Giảng đường do P2 phân công

- Nội dung chính:

* Làm quen với sinh viên (học viên), giới thiệu môn học

1.1. Tổng quan về cấu trúc hệ máy tính

Định nghĩa máy tính: Là tổ hợp các thiết bị điện, điện tử và cơ khí - hoạt động theo chương trình được dùng vào mục đích tự động hóa các quá trình **thu thập, xử lý, lưu trữ và truyền đạt** thông tin.

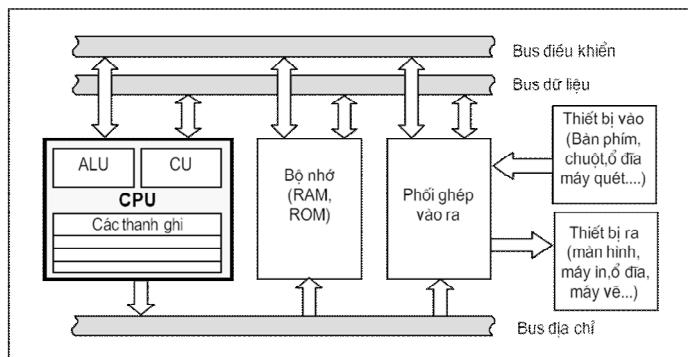
Thông tin trong máy tính được biểu diễn dưới dạng mã nhị phân.

Thành phần cơ bản của máy tính là phần tử biểu diễn số: Có hai trạng thái 0 và 1 biểu diễn được 1 số nhị phân (còn gọi là 1 bit).

Thông tin được biểu diễn trong máy tính dưới dạng 1 dãy số nhị phân có độ dài thường là 2^m bits ($m \geq 3$); 8, 16, 32 hoặc 64 bits (1 byte, 2 bytes, 4 bytes, 8 bytes).

Để thuận lợi trong giao tiếp người – máy thường sử dụng các hệ đếm trung gian như hệ đếm mười, hệ đếm mười sáu.

- 1.1.1. Lịch sử phát triển
 - 1.1.2. Cấu trúc chung hệ máy tính
 - 1.1.2.1. Các thành phần chính của hệ máy tính



- CPU;
- Bộ nhớ vĩnh cửu (ROM);
- Bộ nhớ làm việc (RAM);
- Thiết bị vào (Input);
- Thiết bị ra (Output).

Hệ thống kết nối vật lý giữa các thành phần: System Bus

Tín hiệu địa chỉ (Bus địa chỉ - address Bus): Dùng để định danh các ô nhớ và các thiết bị vào/ra

Tín hiệu số liệu (Bus số liệu – data Bus): Thông tin trao đổi giữa CPU với bộ nhớ và các thiết bị vào/ra.

Tín hiệu điều khiển (Bus điều khiển – control Bus): Cho phép xác định đích và nguồn của số liệu,

Độ rộng bus địa chỉ: BVXL có n dây địa chỉ (ký hiệu $A_0 - A_{n-1}$) biểu diễn khả năng định danh (địa chỉ hóa) được 2^n ô nhớ hay 2^n cổng vào/ra.

Độ rộng bus số liệu: BVXL có m dây số liệu (ký hiệu $D_0 - D_{m-1}$) biểu diễn số lượng bits thông tin được trao đổi giữa CPU với bộ nhớ hay các thiết bị vào/ra.

Theo nguyên lý Von Neumann tại một thời điểm CPU chỉ có khả năng trao đổi số liệu với 1 ô nhớ hoặc 1 cổng vào hoặc 1 cổng ra. Điều đó đồng nghĩa với việc tại 1 thời điểm CPU chỉ thực hiện 1 lệnh, quá trình thực hiện lệnh như thế gọi là xử lý tuần tự (step – by – step) . Đây là nhược điểm lớn nhất của nguyên lý này vì trong thực tế hiện nay đa số các bài toán là quá trình thu thập, lưu trữ, xử lý và truyền đạt đồng thời nhiều tín hiệu (xử lý song song).

1.1.2.2 Phân loại các hệ máy tính

Phân loại theo BVXL;

Phân loại theo tính năng;

Phân loại theo công nghệ:

Máy tính thế hệ 1 (sử dụng công nghệ đèn điện tử) 1950-1959

Máy tính thế hệ 2 (Sử dụng công nghệ bán dẫn) 1960-1963

Máy tính thế hệ 3 (sử dụng công nghệ mạch tơ hợp cỡ nhỏ) 1964-1974

Máy tính thế hệ 4 (sử dụng công nghệ mạch tơ hợp cỡ lớn) 1974 – nay

Đặc điểm: Quan hệ người máy là quan hệ trực tiếp - máy tính cá nhân.

Máy tính thế hệ thứ 5: Máy tính sử lý song song trên cơ sở mạng Nơron (Neural Network).

- Yêu cầu SV chuẩn bị:

Đọc trước tài liệu TL1, trang 7-25.

Câu hỏi bài 1:

1. Trình bày cấu trúc của máy tính hoạt động theo nguyên lý Von Neumann (Thành phần, các hệ bus, 2 thao tác cơ bản của máy tính khi làm việc). Nêu nhược điểm cơ bản của cấu trúc này?

2. Trình bày chức năng của các bus địa chỉ, bus dữ liệu, bus điều khiển. Thế nào gọi là độ rộng Bus? Độ rộng của bus địa chỉ và bus dữ liệu có ý nghĩa như thế nào?

3. Trình bày 2 thao tác cơ bản (đọc/ ghi – chu kỳ bus đọc và chu kỳ bus ghi) của máy tính hoạt động theo nguyên lý Von Neumann. Hai thao tác đó sẽ tương ứng với 4 chu kỳ của tín hiệu clock. Trình bày sự xuất hiện của các tín hiệu địa chỉ, dữ liệu và điều khiển tương ứng với từng chu kỳ tín hiệu clock?

4. Tại sao lại có chu kỳ đợi xuất hiện ở giữa xung T3 và T4.

**BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn**

ĐỀ CƯƠNG BÀI GIẢNG

(Dùng cho 4 tiết giảng)

GIÁO VIÊN

Học phần: CTMT

Bộ môn: Tự động và KTT

Khoa: KTĐK

Trương Đặng Khoa

Nguyễn Trần Hiệp

Bài giảng: Cấu trúc Bộ vi xử lý 8086/88

Chương 1 mục 1.2;

Tiết thứ: 5-8 Tuần thứ: 2

Mục đích, yêu cầu: Giới thiệu cấu trúc và nguyên lý hoạt động của BVXL
8086/88

- *Hình thức tổ chức dạy học:*

Lý thuyết, bài tập, tư học, tư nghiên cứu

- Thời gian:

Lý thuyết: 4 tiết; bài tập: 0 tiết; tư học, tư nghiên cứu: 8 tiết

- Địa điểm:

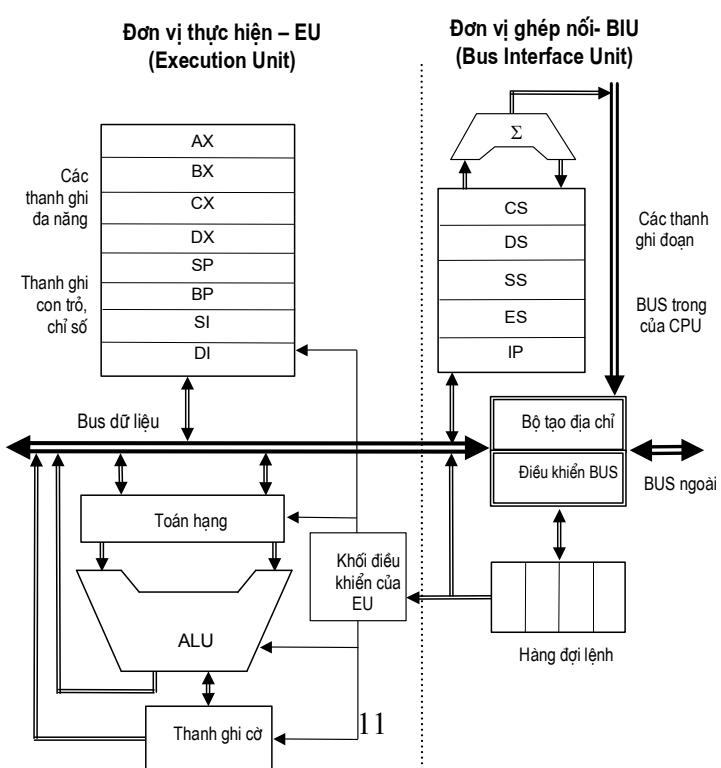
Giảng đường do P2 phân công

- Nội dung chính:

1.2. Cấu trúc Bộ vi xử lý 8086/88

1.2.1. Tổ chức chung của MP 8088/86

Sơ đồ khối của Mn 8086/88

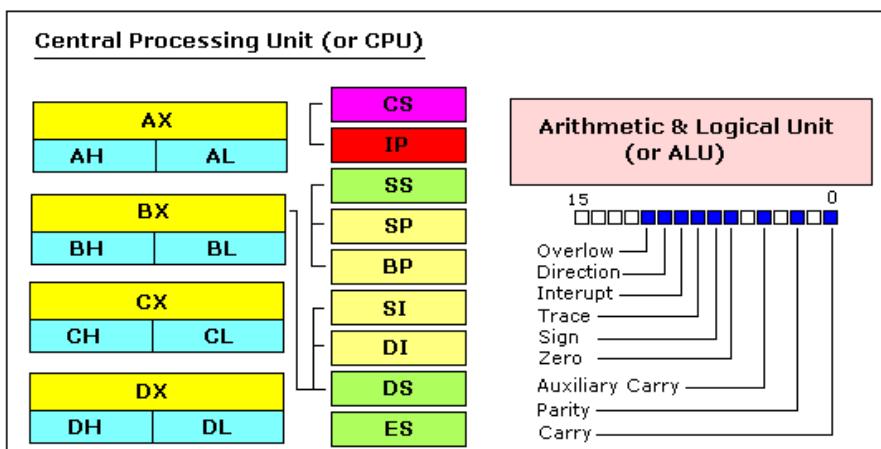


1.2.2. Tổ chức của EU

Đơn vị số học logic (ALU);

Khối điều khiển của EU;

Tập các thanh ghi đa năng 16 bits AX, BX, CX, DX;



Thanh ghi cờ trạng thái

- SF là cờ dấu (sign flag).
- CF là cờ nhớ (carry flag).
- ZF là cờ không (zero flag).
- OF là cờ tràn (overflow flag).
- AF là cờ trung gian (auxiliary flag) hay còn gọi là cờ tràn phụ (half-carry flag).

AF = 0 khi không có sự tràn về dung lượng 4 bit.

AF = 1 khi có sự tràn về dung lượng 4 bit.

Ví dụ : phép toán $00001001 + 00000111 = 00010000$ sẽ lập cờ AF lên 1.

Cờ AF thường được dùng trong các phép toán BCD (là các số dùng hệ nhị phân 4 bit để biểu diễn số thập phân từ $0 \div 9$).

- PF là cờ kiểm tra chẵn lẻ (parity flag).

PF = 1 nếu số bit 1 của kết quả là số chẵn.

PF = 0 nếu số bit 1 của kết quả là số lẻ. Ví dụ : sau khi thực hiện $(00000101 \text{ AND } 00000101)$ thì PF = 1.

- DF là cờ định hướng (direction flag).

DF = 0 : định hướng giảm địa chỉ cho các lệnh xử lý chuỗi.

DF = 1 : định hướng tăng địa chỉ cho các lệnh xử lý chuỗi.

- IF là cờ ngắt quãng (interrupt enable flag).

IF = 0 : cấm ngắt cứng INTR.

IF = 1 : cho phép ngắt cứng INTR.

- TF là cờ bẫy (trap flag). Dùng để chạy từng bước khi cần kiểm tra hoạt động của CPU

1.2.3. Tổ chức của BIU

Đơn vị điều khiển Bus: Bus số liệu ngoài 8086: 16 bits, của 8088: 8 bits

Đơn vị điều khiển địa chỉ: 20 bits địa chỉ

Hàng đợi lệnh (8086 là 3 thanh ghi 16 bits; 8088 là 4 thanh ghi 8 bits)

Thanh ghi đoạn và sự hình thành địa chỉ 20 bits

Các thanh ghi trong BVXL chỉ có độ dài 16 bits do đó để biểu diễn 1 địa chỉ 20 bits phải kết hợp 2 thanh ghi 16 bits với nhau (thanh ghi đoạn: con trỏ, chỉ số)

Quy tắc:

CS: IP chỉ đến ô nhớ chứa lệnh sẽ được thực hiện

DS:SI;DI;BP;BX chỉ đến ô nhớ chứa số liệu

SS: SP chỉ đến ô nhớ là đính của ngăn xếp

ES:DI; BP chỉ đến ô nhớ chứa số liệu.

Địa chỉ logic: Biểu diễn đồng thời cả 2 thanh ghi đoạn và thanh ghi con trỏ chỉ số được dùng trong lập trình và giao tiếp người máy

Địa chỉ vật lý: Biểu diễn vị trí chính xác của ô nhớ trong tập hợp bộ nhớ.

Cách xác định địa chỉ vật lý từ địa chỉ logic:

Thanh ghi đoạn * 16 + Thanh ghi con trỏ chỉ số.

Một địa chỉ vật lý sẽ tương ứng với bao nhiêu địa chỉ logic???

1.2.4. Giản đồ thời gian của các quá trình ghi/đọc

Đọc và ghi là 2 thao tác cơ bản trên máy tính tương ứng với 2 tín hiệu điều khiển Read (\overline{RD}) và Write (\overline{WR})

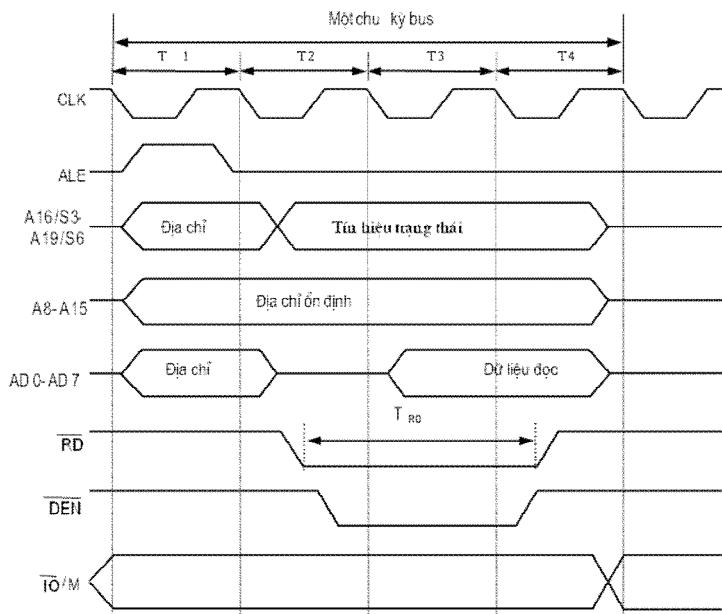
BVXL của Intel còn có tín hiệu trạng thái M/ \overline{IO} , kết hợp 3 tín hiệu M/ \overline{IO} , \overline{RD} và \overline{WR} ta có các tín hiệu điều khiển sau:

M/ \overline{IO}	\overline{RD}	\overline{WR}		Thao tác
0	0	1	\overline{IOR}	Đọc cổng vào
0	1	0	\overline{IOW}	Ghi cổng ra
1	0	1	\overline{MEMR}	Đọc bộ nhớ
1	1	0	\overline{MEMW}	Ghi bộ nhớ

Một thao tác đọc hay ghi được gọi là 1 chu kỳ bus (bus cycle).

Một chu kỳ bus gồm 4 chu kỳ của xung đồng bộ (từ T1 đến T4)

Giản đồ thời gian của một chu kỳ bus

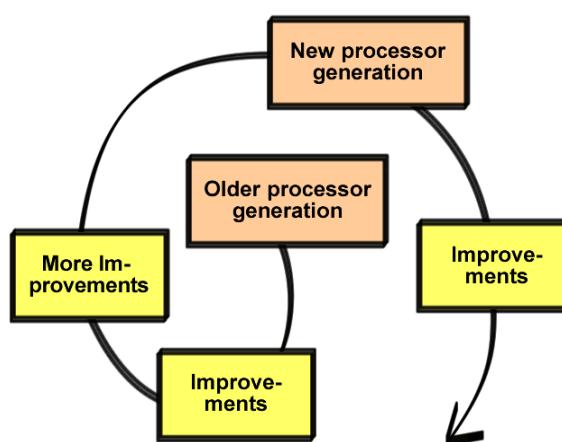


BVXL 8086/88 có hàng đợi lệnh, do đó khi EU đang thực hiện lệnh thì lệnh tiếp theo sẽ được BIU nạp vào hàng đợi lệnh. Cấu trúc hàng đợi lệnh làm cho Bus không có chu kỳ nghỉ. Tốc độ làm việc của BVXL được tăng lên. (Cấu trúc đường ống).

Các bộ vi xử lý tiên tiến của Intel

Các BVXL tiên tiến

Nguyên tắc xây dựng BVXL của Intel



1.4.1 Các thế hệ BVXL của Intel

BVXL 16 bits – 8086, 8088 và 80286.

Khả năng truy cập hơn 1MB bộ nhớ ở chế độ Max.

Khả năng Multitasking – Tại một thời điểm HĐH có thể chạy một vài chương trình.

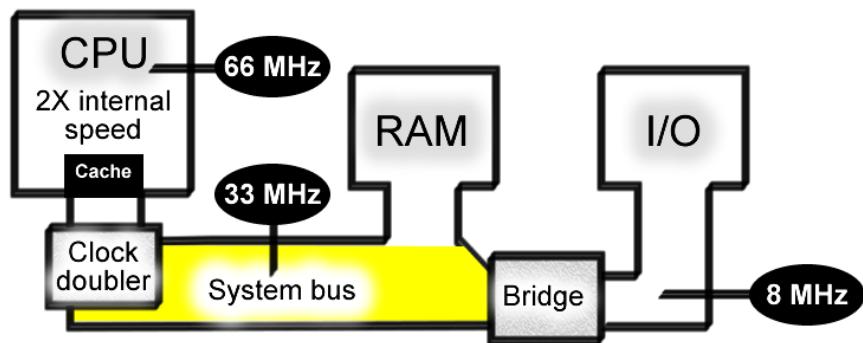
Khả năng sử dụng bộ nhớ ảo *virtual memory* (Sử dụng HDD làm RAM phát triển).

Chế độ bảo vệ mở đường để chuyển đổi từ DOS sang WINDOWS (1990)

Bộ VXL 32 bits – 80386 và 80486

BVXL 80486 sử dụng Cache L1; có tần số l/v gấp đôi tần số của BUS; Bộ đồng xử lý nằm trong BVXL.

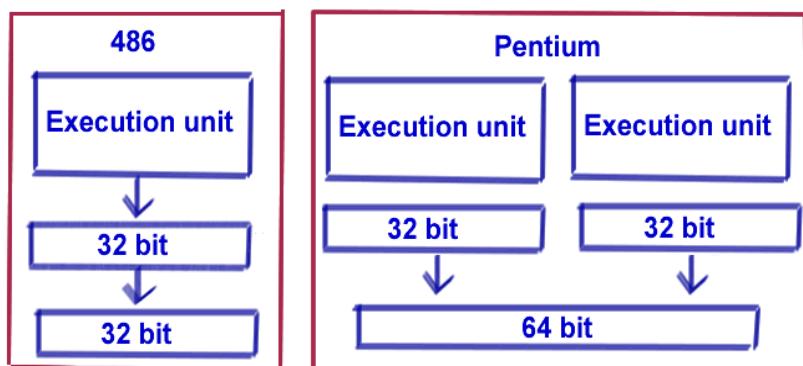
Từ BVXL 486 Intel bắt đầu sử dụng “*clock doubling*” trong BVXL.



BVXL Pentium

Sử dụng kiến trúc *superscalar* – Tại một thời điểm BVXL có thể thực hiện một vài lệnh nhờ sử dụng đường ống nhiều tầng.

Độ rộng BUS của RAM tăng lên 64 bits.



BVXL Pentium 4 với đường ống nhiều tầng cho phép:

Cache L1 có khả năng giải mã lệnh.

Đường ống lên đến 20 tầng (về sau lên đến 31 tầng).

Trong ALU có hai đơn vị thực hiện.

CPU nối với RAM qua north bridge có khả năng truyền 4 gói số liệu trên một clock (tương đương 4*100 Mhz; 4*133 Mhz; 4*200 Mhz; 4*266 Mhz).

Có chế độ Hyper Threading – Trong một số trường hợp nó có thể hoạt động như 2 BVXL riêng biệt.

Cấu trúc đường ống trong BVXL

Với cấu trúc hàng đợi lệnh ở BVXL 8086/88 ta thấy khi EU đang thực hiện lệnh, thì lệnh tiếp theo sẽ được thực hiện (có địa chỉ tại CS:IP) sẽ được nạp vào hàng đợi lệnh. Cấu trúc này cho phép quá trình đọc bộ nhớ được thực hiện đồng thời với quá trình xử lý lệnh. Xuất phát từ ý tưởng đó, ta thấy: nếu chia quá trình thực hiện một lệnh thành nhiều giai đoạn, cấu trúc hàng đợi lệnh sẽ cho phép thực hiện đồng thời các giai đoạn khác nhau của các lệnh kế tiếp nhau. Đó chính là bản chất của cấu trúc đường ống. Giả sử một lệnh máy được chia thành 5 giai đoạn:

Đọc lệnh (IF: Instruction Fetch).

Giải mã lệnh (ID: Instruction Decode).

Thi hành lệnh (EX: Execute).

Thâm nhập bộ nhớ trong hoặc nhảy (MEM: Memory access)

Lưu trữ kết quả (RS: Result Storing).

Mỗi giai đoạn có thể được thực hiện trong 1 hay nhiều xung Clock.

Chuỗi lệnh

Chu kỳ xung đồng hồ

	1	2	3	4	5	6	7	8	9
Lệnh thứ i	IF	ID	EX	MEM	RS				
Lệnh thứ i+1	IF	ID	EX	MEM	RS				
Lệnh thứ i+2		IF	ID	EX	MEM	RS			
Lệnh thứ i+3			I F	ID	EX	MEM	RS		
Lệnh thứ i+4				IF	ID	EX	MEM	RS	

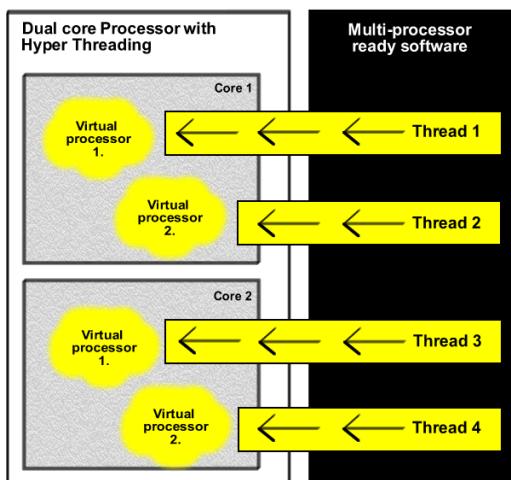
Trong tiến trình trên ta thấy trong chu kỳ của 1 xung clock nhiều giai đoạn khác nhau của lệnh được thực hiện đồng thời trên các phần tử khác nhau trong BVXL. Với 5 lệnh kế tiếp nhau, nếu không có cấu trúc đường ống sẽ cần đến 45 chu kỳ để thực hiện. Nhưng 5 lệnh đó chỉ thực hiện trong 9 chu kỳ nếu BVXL có cấu trúc đường ống đó là ưu điểm nổi bật của cấu trúc đường ống. Cấu trúc đường ống cũng là khởi đầu cho các cấu trúc vô hướng, siêu vô hướng, phân luồng và siêu phân luồng trên các BVXL tiên tiến của Intel.

Công nghệ siêu phân luồng (Hyper-Threading Technology)

Để khai thác hết cấu trúc đường ống; BVXL P4 cho phép tại một thời điểm xử lý 2 lệnh. Các BVXL thế hệ trước chỉ xử lý 1 lệnh. (Tương đương với việc sử dụng 2 CPU trong một máy tính).

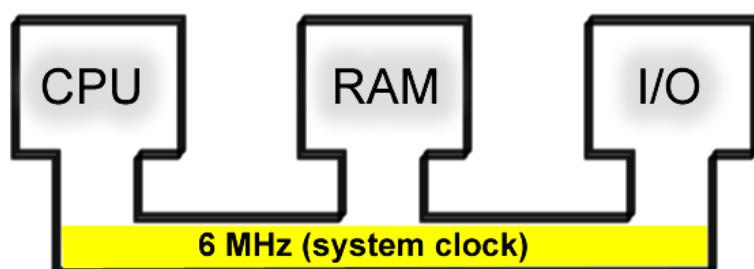
Dual-core processors

Sử dụng 2 CPU trong một chip

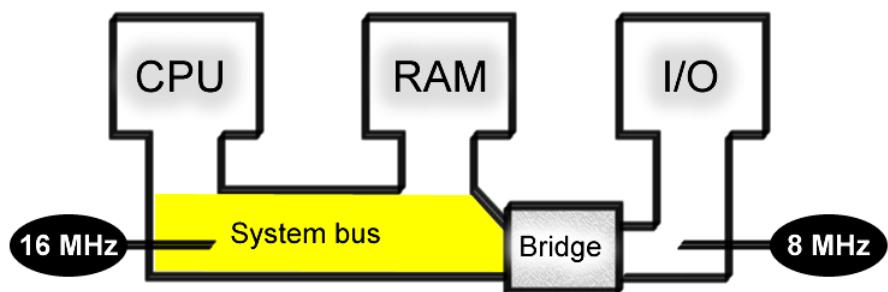


1.4.2 Chip set

Trên máy XT CPU, RAM và I/O cùng được nối vào 1 BUS.



Năm 1987 Compaq có ý tưởng chia đồng hồ hệ thống thành hai hệ khác nhau CPU và RAM làm việc cùng một tần số độc lập với tần số của I/O



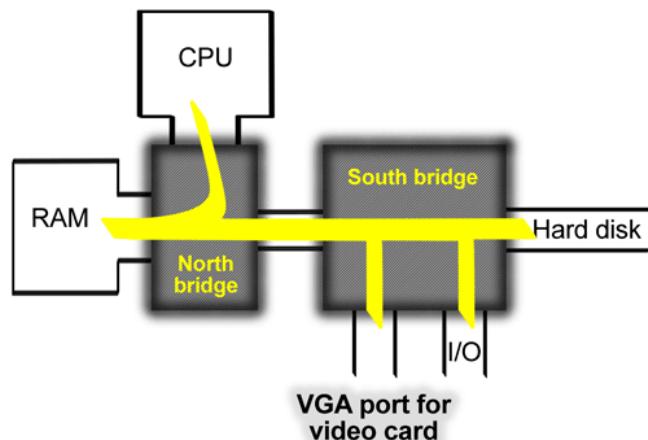
Kết nối giữa hai hệ bus sử dụng thiết bị được gọi là cầu nối “Bridge” hay còn gọi là **chipset**.

Bus của MainBoard ®được điều khiển bởi rất nhiều mạch phần cứng có các chức năng khác nhau. Tất cả chúng được đóng trong một mạch tích hợp cực lớn gọi là **chipset**.

Cấu trúc phổ biến nhất của chipset gồm 2 chip gọi là **north and south bridges**.

North bridge: Điều khiển truyền số liệu giữa CPU, RAM và cổng AGP và cổng PCI Express x16.

South bridge: Liên kết với **North bridge** và truyền số liệu đến tất cả các ngoại vi còn lại.



Bài tập: chia bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 16-27

Đọc trước TL2 trang 344-348

Câu hỏi bài 2:

1. Các thanh ghi của BVXL 8086/88 đều có độ dài 16 bits. Làm thế nào BVXL tạo được Bus địa chỉ 20 bits. Thế nào là địa chỉ Logic, địa chỉ vật lý? Tại sao nói không gian địa chỉ 20 bits của BVXL 8086/88 lại được chia thành 16 đoạn (segment)?

2. Nếu phương pháp xác định địa chỉ Vật lý từ địa chỉ Logic. Xác định ít nhất 5 địa chỉ logic của ô nhớ có địa chỉ vật lý 3682F (H).

3. Một địa chỉ vật lý sẽ có tối đa và tối thiểu bao nhiêu địa chỉ logic?

4. Các thanh ghi đoạn CS, DS, SS, ES được kết hợp với các thanh ghi con trỏ chỉ số (SI, DI, SP, BP, IP.v.v.) theo quy tắc như thế nào để tạo được địa chỉ 20 bits tương ứng với các vùng làm việc khác nhau của bộ nhớ RAM khi thực hiện các lệnh của Hệ điều hành hay các chương trình ứng dụng.

5. Nêu cấu trúc hàng đợi lệnh của BVXL 8086/8088 và điểm khác biệt về cấu trúc hàng đợi lệnh của BVXL 8086 và BVXL 8088? Điều đó có liên quan gì đến độ rộng Bus dữ liệu của Bộ VXL 8086 và 8088 hay không? Trình bày ưu điểm của cấu trúc này so với cấu trúc của các BVXL thế hệ trước?

6. Xác định trạng thái của thanh ghi cờ trạng thái khi BVXL thực hiện 2 câu lệnh sau:

MOV AX, 0ffffh (1256H)

ADD AX, 0001h (0fff0H)

7. Trình bày tư tưởng xây dựng BVXL của Intel.

8. Trình bày cấu trúc đường ống (giả thiết 1 lệnh được chia thành 5 vi lệnh, mỗi vi lệnh được thực hiện trong 1 clock tick). Nêu ưu, nhược điểm của cấu trúc đường ống?

9. Giải thích thé là là chip 2 lõi, 4 lõi. Cấu trúc này có khắc phục được hoàn toàn nhược điểm của cấu trúc VonNeumann không?

10. Nói rằng trên các máy tính hiện nay CPU, các vi mạch nhớ và các thiết bị ngoại vi không làm việc ở cùng 1 tần số là đúng hay sai? Giải thích tại sao?

11. Nêu vai trò và chức năng của North bridge và South bridge?

BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn

Trương Đăng Khoa

ĐỀ CƯƠNG BÀI GIẢNG
(Dùng cho 4 tiết giảng)

Học phần: CTMT
Bộ môn: Tự động và KTT
Khoa: KTĐK

GIÁO VIÊN

Nguyễn Trần Hiệp

Bài giảng: Hệ thống hỗ trợ trên máy tính IBM PC

Chương 1 mục 1.3;

Tiết thứ: 9-12

Tuần thứ: 3

Mục đích, yêu cầu: Nghiên cứu các hệ thống hỗ trợ cho BVXL trên máy tính

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 4 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 8 tiết

- **Địa điểm:**

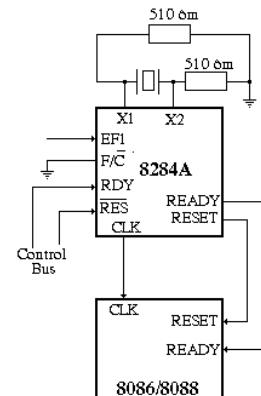
Giảng đường do P2 phân công

- **Nội dung chính:**

1.3. Hệ thống hỗ trợ

1.3.1. Mạch tạo đồng bộ 8224

Tần số gốc cho 8284 làm việc có thể được tạo từ Mạch tạo dao động thạch anh hoặc từ 1 nguồn dao động khác. Trên máy tính IBM PC XT, 8284 làm việc từ nguồn dao động được nối đến chân X₁, X₂ của 8284 thạch anh với tần số 4,7 MHz



1.3.2. Hệ thống hỗ trợ Bus

Mạch tách tín hiệu địa chỉ/ số liệu: Do BVXL 8086/88 có chung các đường địa chỉ/số liệu (AD₀ – AD₇ – với 8088 và AD₀ – AD₁₅ với 8086) do đó cần phải tách tín hiệu địa chỉ ra bus địa chỉ và tín hiệu số liệu ra bus số liệu. Mạch tách tín hiệu địa chỉ số liệu sử dụng tín hiệu điều khiển là ALE, \overline{DEN} cùng với mạch chốt địa chỉ sử dụng vi mạch 74LS373 và bộ đếm số liệu 2 chiều sử dụng vi mạch 74LS245. Trong một chu kỳ bus, tại thời điểm của xung T1 tín hiệu địa chỉ xuất hiện trên các chân địa chỉ/ số liệu, tín hiệu ALE sẽ cho phép các vi mạch 74LS373 chốt địa chỉ tại đầu ra của nó, tại thời điểm của xung T3 số liệu sẽ xuất hiện và tín

hiệu cho phép mở bộ đệm 2 chiều 74LS245 nối bus số liệu với tín hiệu số liệu của BVXL (Sử dụng biểu đồ thời gian ghi/đọc, hình vẽ ghép nối kênh địa chỉ/số liệu, sơ đồ và bảng chân lỳ vi mạch 74LS373, vi mạch 74LS 245)

1.3.3 Mạch chốt địa chỉ và đệm dữ liệu.

1.3.4 Mạch giải mã

Máy tính làm việc theo nguyên lý VonNewmann do đó tại một thời điểm, CPU chỉ thực hiện việc trao đổi dữ liệu với 1 ô nhớ hay 1 cổng vào/ra. Quá trình tìm ra được ô nhớ hay cổng vào/ra sẽ thực hiện trao đổi dữ liệu với CPU được gọi là quá trình giải mã.

Nguyên tắc của giải mã.

Có 2 loại tín hiệu liên quan đến quá trình giải mã.

+ Tín hiệu lựa chọn bộ nhớ hay thiết bị ngoại vi (vào/ra) thực hiện trao đổi dữ liệu (M/\overline{IO})

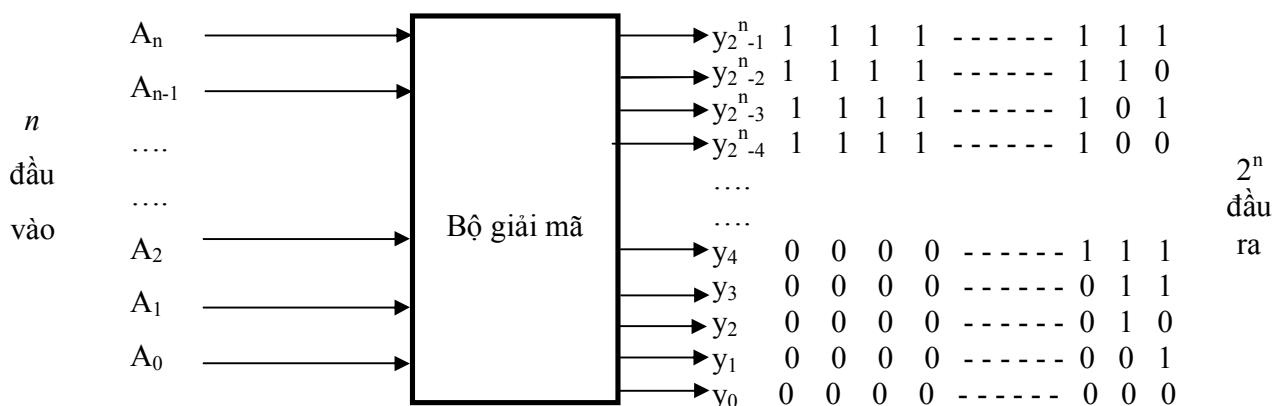
+ Tín hiệu lựa chọn cụ thể ô nhớ hay cổng, tín hiệu địa chỉ ($A_n - A_0$)

Nếu Bus địa chỉ có n dây dẫn – n bits ta có các tín hiệu địa chỉ ($A_n - A_0$), cho phép biểu diễn được vị trí của 2^n ô nhớ hay 2^n cổng.

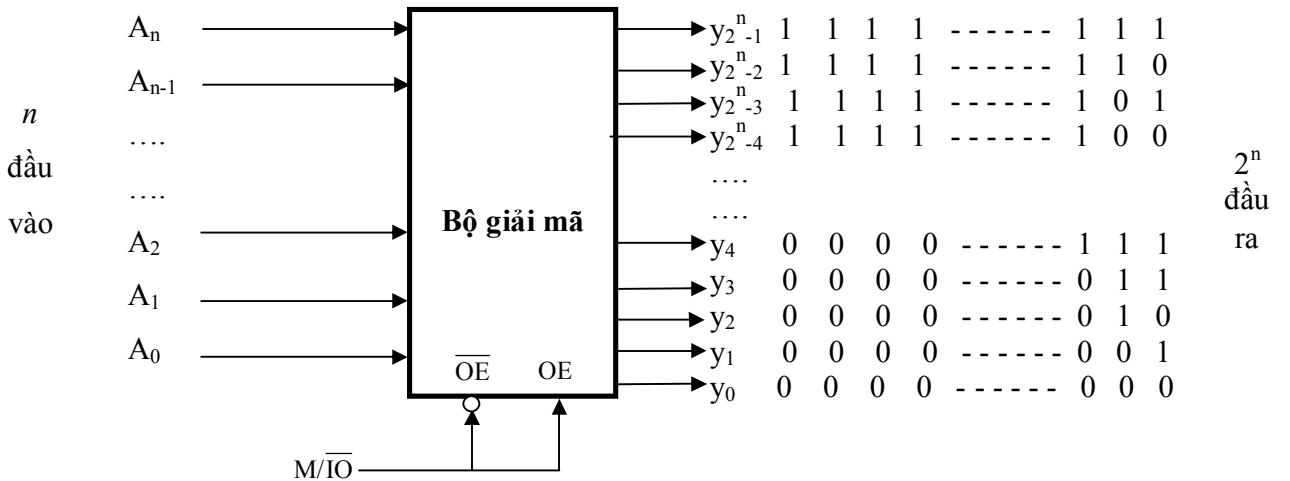
	A_n	A_{n-1}	A_{n-2}	-----	A_1	A_0
Ô nhớ có thứ tự thấp nhất (đầu tiên)	0	0	0	00000	0	0
Ô nhớ thứ hai	0	0	0	00000	0	1
Ô nhớ thứ ba	0	0	0	00000	1	0

Ô nhớ có thứ tự cao nhất (cuối cùng)	1	1	1	11111	1	1

Như vậy bộ giải mã có thể hình dung là 1 thiết bị phần cứng có n đầu vào (các bits địa chỉ) và có 2^n đầu ra chính là tín hiệu lựa chọn ô nhớ hay cổng.



Thêm vào đó, các tín hiệu điều khiển M/\overline{IO} (\overline{MEMR} , \overline{MEMW} , \overline{IOR} , \overline{IOW}) cũng được bổ sung vào mạch giải mã để cho phép chọn bộ nhớ, hay công và chiều của dữ liệu là hướng vào CPU hay hướng ra từ CPU.



Khi M/\overline{IO} có mức logic 1: Thực hiện giải mã chọn ô nhớ

Khi M/\overline{IO} có mức logic 0: Thực hiện giải mã chọn công

Vẽ sơ đồ cho trường hợp giải mã chọn công vào và giải mã chọn công ra?

Giải mã chọn ô nhớ trong vi mạch nhớ ROM và chọn ô nhớ trong vi mạch nhớ RAM?

1.3.5. Mạch điều khiển BUS 8288

Mạch điều khiển bus 8288 được sử dụng khi BVXL làm việc ở chế độ tối đa hay chế độ bảo vệ, khi đó các tín hiệu điều khiển sẽ quá trình đọc/ghi sẽ được tạo từ vi mạch này chứ không từ 3 tín hiệu M/\overline{IO} , \overline{RD} và \overline{WR} .

Các chân 26,27,28 của BVXL sẽ tương ứng với các trạng thái để mã hóa các chu kỳ máy:

\bar{S}_2	\bar{S}_1	\bar{S}_0	Chu kỳ Bus	\bar{S}_2	\bar{S}_1	\bar{S}_0	Chu kỳ Bus
0	0	0	Trả lời ngắn	1	0	0	Gọi lệnh
0	0	1	Đọc số liệu từ cổng vào	1	0	1	Đọc bộ nhớ
0	1	0	Ghi số liệu ra cổng ra	1	1	0	Ghi bộ nhớ
0	1	1	Dừng	1	1	1	Thụ động

1.3.6. Bus hệ thống trên máy tính

Các thiết bị ngoại vi kết nối với hệ thống nhờ các khe cắm mở rộng (expansion slot).

Bus hệ thống (Bus system) sẽ kết nối tất cả các thành phần lại với nhau.

Có 3 loại bus :bus dữ liệu (data bus), bus địa chỉ (address bus) và bus điều khiển (control bus).

Address Bus : nhóm đường truyền nhận diện vị trí truy xuất trong thiết bị đích : thông tin được đọc từ đâu hoặc ghi vào đâu.

Data Bus : nhóm đường truyền để tải data thực sự giữa các thiết bị hệ thống do địa chỉ trên address bus đã xác định. Độ rộng của data bus (số đường dây dẫn) xác định data trong mỗi lần truyền là bao nhiêu.

Control Bus : nhóm đường truyền cho các tín hiệu điều khiển như : tác vụ là đọc hay ghi, tác vụ thực thi trên bộ nhớ hay trên thiết bị ngoại vi, nhận dạng chu kỳ bus và khi nào thì hoàn tất tác vụ

Độ rộng bus địa chỉ: là số dây dẫn hợp thành bus bus có độ rộng n thì có thể nhận biết được 2^n địa chỉ.

Độ rộng bus số liệu: được thiết kế theo nguyên tắc là bội của 8 (8,16,32,64 bit) như thế mỗi lần truyền 1 byte/2 bytes/4 bytes tùy theo máy. Độ rộng Data bus càng lớn thì data truyền càng nhanh.

2.1. Các loại BUS trên máy tính

2.1.1 Giới thiệu về BUS máy tính

Bus trên máy tính là tập hợp các tín hiệu địa chỉ, số liệu, điều khiển cùng với các tín hiệu đồng bộ, nguồn được chế tạo tuân theo một tiêu chuẩn nhất định đảm bảo cho việc trao đổi số liệu giữa CPU với bộ nhớ và các thiết bị ngoại vi.

Một hệ Bus tiên tiến luôn đảm bảo được tốc độ truyền số liệu giữa CPU bộ nhớ và ngoại vi là nhanh nhất, đồng thời có khả năng nhận dạng được sự thay đổi của thiết bị ngoại vi hay bộ nhớ để đảm bảo cho việc truyền số liệu không bị xung đột.

2.1.2 Thông lượng, giao thức BUS, tự động định dạng cấu hình

Tốc độ mà bus có thể truyền dữ liệu từ thiết bị chủ tới thiết bị tớ gọi là *thông lượng* hay *độ rộng dải* của bus (bandwidth) hoặc *năng suất truyền* của bus (throughput). Đơn vị đo là MB/sec. Thông lượng của bus phụ thuộc vào tốc độ, độ rộng và giao thức bus:

Dải thông của Bus = tần số làm việc của bus * độ rộng bus số liệu.

Mục đích phát triển của các hệ bus trên máy tính IBM PC và tương thích là các loại bus thế hệ sau luôn có dải thông lớn hơn thế hệ trước để đảm bảo truyền số liệu nhanh hơn.

Plug and play

Microsoft và Intel cộng tác với nhau để trang bị bus ISA thêm khả năng của cấu hình tự động. Đặc điểm này còn gọi là *cắm là chạy* (plug and play). PCI với cấu hình tự động có thể làm việc hoàn chỉnh chỉ sau khi card EISA và BIOS được trang bị cấu hình tự động. ISA và MCA đều được bổ sung đặc tính quan trọng này. Có 3 trường hợp liên quan tới Plug and Play như sau:

1. Cả BIOS trên board mẹ và card bổ sung đều không phải là Plug and play.
2. BIOS trên board mẹ được trang bị Plug and play nhưng card bổ sung thì không. Trường hợp này cần một phần mềm cài đặt sẽ giúp sắp xếp địa chỉ I/O, IRQ và các kênh DMA.
3. Cả BIOS trên board mẹ và card bổ sung được trang bị Plug and play. Trường hợp này, cấu hình tự động sẽ thực hiện mọi công việc. Tự sắp xếp địa chỉ I/O, IRQ và các kênh DMA không cần người dùng can thiệp.
4. Hot Plug and Play là khả năng bus hoàn toàn có thể nhận biết sự thay đổi (ngắt và kết nối ngoại vi đến bus) và cài đặt điều khiển cho ngoại vi ngay cả khi máy tính vẫn đang làm việc. Khả năng này cho phép máy tính có thể làm việc liên tục không phải dừng khi thay thế các ngoại vi.

2.2.1 Các chuẩn BUS

ISA, MCA, EISA

PCI

Bài tập: chũa bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 27-34 từ 85-99

Đọc trước TL2 trang 340-344

Câu hỏi bài 3:

1. BXL 8086/88 có chung các đường địa chỉ/số liệu AD0 – AD7 và AD0 – AD15. Làm cách nào nó có thể tách địa chỉ và số liệu ra riêng biệt. Vai trò của các tín hiệu ALE và DEN.

2. BVXL 8086/88 có hai chế độ l/v Min và Max. Hai chế độ này được dùng khi nào? Điểm khác biệt lớn nhất của 2 chế độ này khi làm việc?

3. Phân biệt sự giống và khác nhau giữa BVXL 8086 và 8088 về bus dữ liệu ngoài, cấu trúc hàng đợi lệnh.

4. Định nghĩa về thông lượng, mối quan hệ giữa thông lượng với độ rộng bus dữ liệu và tần số làm việc của bus. Tiêu chí để xây dựng các hệ Bus EISA, MCA, VLBUS, PCI là gì?

5. Thế nào là Plug-and-play. Trình bày các mức độ plug and play. Trong các hệ bus tiêu chuẩn ISA EISA, MCA, VLBUS, PCI hệ bus nào có hỗ trợ plug and play?

BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn

Trương Đăng Khoa

ĐỀ CƯƠNG BÀI GIẢNG
(Dùng cho 8 tiết giảng)

Học phần: CTMT
Bộ môn: Tự động và KTT
Khoa: KTĐK

GIÁO VIÊN

Nguyễn Trần Hiệp

Bài giảng: Tập lệnh của BVXL 8086/88

Chương 1 mục 1.4;

Tiết thứ: 13-20

Tuần thứ: 4,5

Mục đích, yêu cầu: Giới thiệu tập lệnh BVXL 8086/8088 và lập trình hợp ngữ

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 6 tiết; bài tập: 2 tiết; tự học, tự nghiên cứu: 8 tiết

- **Địa điểm:**

Giảng đường do P2 phân công

- **Nội dung chính:**

1.4.1 Giới thiệu về lập trình hợp ngữ.

Ngôn ngữ máy (**Machine language**): Là tập các lệnh của BVXL biểu diễn dưới dạng nhị phân (hoặc hex) - mã máy, nó được nạp trực tiếp vào bộ nhớ để BVXL thực hiện.

Hợp ngữ (**Assembly language**): Là ngôn ngữ bậc thấp, một chương trình viết bằng hợp ngữ bao gồm một chuỗi các lệnh của BVXL biểu diễn bằng các cú pháp dễ nhớ mà khi được dịch bằng một trình dịch hợp ngữ, thì chúng có khả năng nạp được vào bộ nhớ đồng thời thực thi được bởi BVXL.

Trình dịch hợp ngữ (**Assembler**): Là một chương trình dịch bậc thấp, dịch các chương trình hợp ngữ thành các mã máy, nạp được vào bộ nhớ đồng thời thực thi được. VD trình dịch Emu8086.

db 0B4h,12h,0B0h,23h,02h,0C4h,0BEh,34h,12h,8Ah,1Ch,0CDh,20h

Mã máy (Hex)	Lệnh hợp ngữ	Ghi chú
	ORG 100h	Chỉ thị dẫn hướng cho trình dịch đặt mã lệnh từ đ/c 100h
B412	MOV AH,12H	Gán AH=12H
B023	MOV AL,23H	Gán AL=23H
02C4	ADD AL,AH	AL=AL+AH
BE3412	MOV SI,1234H	Gán SI=1234H
8A1C	MOV BL,[SI]	Đọc ô nhớ có đ/c 1234h vào BL
CD20	INT 20H	Lệnh gọi ngắt 20h kết thúc C/T
	END	Từ khóa kết thúc C/T hợp ngữ

Cú pháp một câu lệnh hợp ngữ

Một lệnh hợp ngữ đầy đủ gồm 4 thành phần sau:

[Tên nhãn:] <Tên lệnh> [Các toán hạng] [;Chú thích]

Vdụ Next: MOV AH, 12h ; gán AH = 12h

Trong đó:

+ [Tên nhãn]: Là một dãy các kí tự đứng trước câu lệnh (kết thúc (:)), nó cho biết địa chỉ của câu lệnh ngay sau tên nhãn, tên nhãn được dùng trong các câu lệnh lặp, lệnh nhảy, hoặc tên một chương trình con... Do đó, nó chỉ được sử dụng khi cần.

- Trong một chương trình hợp ngữ không thể có hai tên nhãn ở hai vị trí trùng tên nhau.

+ <Tên lệnh>: Cho người dùng biết hoạt động của lệnh và thường là chữ tiếng anh viết tắt như MOV, ADD(Addition), SUB(subtract), INC, DEC ...

- Lệnh hợp ngữ không phân biệt chữ hoa/thường. Trong chương trình hợp ngữ mỗi dòng chỉ có thể chứa 1 lệnh và mỗi lệnh phải được đặt trên 1 dòng.

+ [Các toán hạng]: Là đối tượng mà lệnh tác động vào. Một lệnh hợp ngữ của 8088/8086 có thể không có toán hạng, có một toán hạng, hoặc có hai toán hạng. Nếu có hai toán hạng thì toán hạng đứng trước gọi là [Toán hạng đích], toán hạng đứng sau gọi là [Toán hạng nguồn]. [Toán hạng đích] không thể là một hằng số.

+ [;Chú thích]: Lời giải thích của người viết chương trình, trình biên dịch bỏ qua chú thích này, thường được dùng để làm rõ ý nghĩa của câu lệnh. Lời giải thích phải nằm sau dấu chấm phẩy (;).

Dưới đây là một đoạn chương trình hợp ngữ:

```

        mov CX, 5    ; gán cx=5, hai TH
        Mov BX, 100 ; gán bx=100, hai TH
BACK: DEC BX  ; giảm bx 1 đơn vị, 1 TH
        NOP          ; lệnh không có TH
        LOOP BACK ;Tên nhãn BACK trong lệnh lặp.

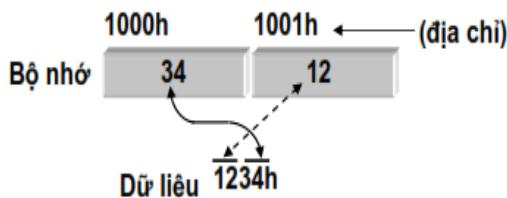
```

Khai báo biến, hằng

- Khai báo biến 1byte sử dụng từ khóa **DB**
 - VD khai báo 2 biến a,b: **a db 00h**
b db ?
- Khai báo biến 2byte (Word): **DW**
 - VD khai báo biến c 2 byte: **c dw 1234h**
- Khai báo chuỗi: **db**
 - VD khai báo chuỗi str: **str db "Hello"**
- Khai báo mang: **m db 100 dup(?)**
- Khai báo hằng sử dụng từ khóa: **EQU**
 - VD: **kytuA EQU 65**

Vị trí khai biến, hằng

- File COM: Hằng khai báo ở đầu chương, biến ở cuối chương trình, ở vị trí chưa dữ liệu.
- File EXE: Khai báo trong đoạn dữ liệu
- Lưu trữ 16 bit trong bộ nhớ (địa chỉ cao có trọng số lớn hơn) ;BE3412 ;
MOV SI, 1234h



Chỉ rõ kích thước ô nhớ

- Xác định rõ kích thước của ô nhớ dùng từ khóa PTR.
- **Hoạt động 8 bit:** **BYTE PTR [1000h]**

Tham chiếu đến 1 byte có địa chỉ 1000h

- **Hoạt động 16 bit:** **WORD PTR [1000h]**

Tham chiếu 2 byte có địa chỉ 1000h, 1001h

CÁC CHẾ ĐỘ ĐỊNH ĐỊA CHỈ

. Chế độ định địa chỉ thanh ghi

Ví dụ: ADD AX, BX

2. Chế độ định địa chỉ tức thi

Ví dụ: MOV AX, 0B800H

Các chế độ định địa chỉ bộ nhớ

3. Chế độ định địa chỉ bộ nhớ trực tiếp

Ví dụ: MOV DL, [100]

4. Chế độ định địa chỉ bộ nhớ gián tiếp thanh ghi

Ví dụ: MOV AX, [BX] DS:BX

5. Chế độ định địa chỉ quan hệ cơ sở

Ví dụ: MOV CX, [BX]+10 DS:BX+10

6. Chế độ định địa chỉ quan hệ chỉ số

Ví dụ: MOV DX, [SI] + 5 DS:SI+5

7. Chế độ định địa chỉ chỉ số cơ sở

MOV CL, [BX] [DI]+8 DS:BX+DI+8

MOV CH, [BP] [SI]+12 ES:BP+SI+12

Địa chỉ hiệu dụng

- Địa chỉ hiệu dụng là tổ hợp của 3 nhóm địa chỉ sau đặt trong dấu []
 - Nhóm thanh ghi chỉ số: SI, DI
 - Nhóm thanh ghi cơ sở: BX, BP
 - Địa chỉ trực tiếp: số 16 bit

Đ/C hiệu dụng không có 2 thanh ghi trong cùng một nhóm và phù hợp với một trong các chế độ địa chỉ.

Ví dụ :

. Địa chỉ hiệu dụng hợp lệ :

[1000h], [SI], [DI], [BX], [BP]

[SI+BX], [SI+BP], [DI+BX], [DI+BP], [SI+1000h], [DI+100h], [BX+1], [BP+1]

[SI][BX][1000h], [SI+ BP+1000h], [DI+BX][1000h], [DI+1000h][BP]

. Địa chỉ hiệu dụng không hợp lệ :

[70000], [AX], [SI+DI+1000h], [BX][BP]

2. Cấu trúc file chương trình COM, EXE

```

C. trúc CT hợp ngữ tạo ra file .COM
MaA EQU 65 ;Khai báo các hằng số
ORG 100h ; Đặt mã lệnh từ đ/c 100h
;Viết các lệnh hợp ngữ
MOV AH,12H
MOV AL,23H
ADD AL,AH
INT 20H
; Khai báo các biến
str db "Hello"
END
- Hợp ngữ không phân biệt chữ hoa,
thường.
- Chú thích viết sau dấu chấm phẩy (;)

```

```

Cấu trúc CT hợp ngữ tạo ra file .EXE
MaA EQU 41h
data segment ; Khai báo đoạn data
    pkey db "press any key...$"
ends
stack segment ; khai báo đoạn stack
    dw 128 dup(0)
ends
code segment ; Khai báo đoạn mã lệnh
start:
    mov ax, data
    mov ds, ax
    mov es, ax
;Viết các lệnh hợp ngữ
    mov ax, 4c00h
    int 21h
ends
end start

```

VD 1. Viết chương trình hợp ngữ dạng COM hiển thị ký tự 'A' lên màn hình;
Dùng chức năng 02h của ngắt 21h.

```

MaASCII EQU 'A'
org 100h
    mov ah, 02h ;          02h mã chức năng xuất ký tự
    mov dl, MaASCII      ; nạp mã ascii vào dl
    int 21h              ; gọi ngắt 21h
    mov a, dl
    int 20h              ; gọi ngắt kết thúc CT
    a db 0
end

```

Đặc điểm file COM, EXE

File chương trình .COM là file chương trình có cấu trúc đơn giản nhỏ gọn, mã lệnh và dữ liệu đặt trong cùng một đoạn bộ nhớ. Kích thước file <=64Kb. 4 thanh ghi đoạn đặt cùng một đoạn bộ nhớ (CS=DS=SS = ES).

File chương trình .EXE là file chương trình có cấu trúc phức tạp hơn file .COM, mã lệnh và dữ liệu có thể đặt trong các đoạn bộ nhớ khác nhau. Kích

thúréc file .exe cho phép lớn hơn 64Kb. 4 thanh ghi đoạn CS, DS, SS, ES có thể trỏ đến các đoạn bộ nhớ khác nhau.

Viết chương trình COM hiển thị 2 ký tự A, B lên màn hình dùng chức năng AH=02h, của ngắt int 21h.

1.4.2 Tập lệnh BVXL 8086/8088

Các ký hiệu quy ước

- reg: chỉ một thanh ghi bất kỳ
- reg16: thanh ghi 16 bit
- segreg: thanh ghi đoạn
- accum: thanh ghi tích lũy AX, hoặc AL
- mem: địa chỉ bộ nhớ trực tiếp
- mem16: địa chỉ từ nhớ 2 byte
- immed: hằng số
- immed8: hằng số 8 bit
- thđ: toán hạng đích (toán hạng trước)
- thn: toán hạng nguồn (toán hạng sau)

Nhóm lệnh chuyển dữ liệu

1. Lệnh MOV

- Dạng lệnh :

MOV	reg,reg	MOV	reg,immed
MOV	mem,reg	MOV	mem,immed
MOV	reg,mem	MOV	mem16,segreg
MOV	reg16,segreg	MOV	segreg,mem16
MOV	segreg,reg16		

- Giải thích : **thđ** ← **thn**

- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
_____	_____	_____	_____	_____	_____	_____	_____

- Chép toán hạng nguồn vào toán hạng đích.

- Ví dụ :

MOV	AX,CX	;	AX ← CX
-----	-------	---	---------

2. Lệnh PUSH – Cất dữ 2 byte vào Stack

- Dạng lệnh : $PUSH \ reg16$ $PUSH \ segreg$
 $PUSH \ mem16$
- Giải thích : $SP \leftarrow SP-2$
 $[SS:SP+1,SS:SP] \leftarrow thd$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
- Đẩy toán hạng nguồn 16 bit vào chồng (địa chỉ định chồng là SS:SP).
- Ví dụ : $PUSH \ DI ; [SS:SP+1,SS:SP] \leftarrow DI$
 $PUSH \ CS ; [SS:SP+1,SS:SP] \leftarrow CS$
 $PUSH \ [SI] ; [SS:SP+1,SS:SP] \leftarrow [SI+1,SI]$

3. Lệnh POP – Lấy 2 byte ra từ Stack

- Dạng lệnh : $POP \ reg16$ $POP \ segreg$
 $POP \ mem16$
- Giải thích : $thd \leftarrow [SS:SP+1,SS:SP]$
 $SP \leftarrow SP+2$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
- Lấy dữ liệu từ địa chỉ định chồng vào toán hạng đích.
- Ví dụ : $POP \ AX ; AX \leftarrow [SS:SP+1,SS:SP]$
 $POP \ ES ; ES \leftarrow [SS:SP+1,SS:SP]$
 $POP \ [BX+1] ; [BX+2,BX+1] \leftarrow [SS:SP+1,SS:SP]$

4. Lệnh LEA – lấy địa chỉ ô nhớ

- LEA reg16, mem
- Giải thích : $thd \leftarrow địa\ chỉ$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
- Nạp địa chỉ hiệu dụng vào thanh ghi 16 bit.
- Ví dụ : $LEA \ BX,[1000h] ; BX \leftarrow 1000h$
 $LEA \ SI,[DI][BX][2000h] ; SI \leftarrow DI+BX+2000h$

5. Lệnh IN – đọc cổng IO

- Dạng lệnh : IN *accum,immed8*
IN *accum,DX*
- Giải thích : **btl** \leftarrow [cổng IO]
- Tác động cờ : **OF DF IF SF ZF AF PF CF**

--	--	--	--	--	--	--	--	--
- Nhập dữ liệu từ cổng xuất nhập vào thanh ghi bộ tích lũy AL hay AX. Trường hợp AX sẽ nhập byte thấp trước, byte cao sau.
- Dạng lệnh có *immed8* dùng trong trường hợp địa chỉ cổng xuất nhập 8 bit.
- Ví dụ : IN AL,61h
IN AX,40h
- Dạng lệnh có thanh ghi DX dùng cho trường hợp địa chỉ cổng 16 bit. Tuy nhiên dạng này vẫn có thể dùng cho cổng xuất nhập có địa chỉ 8 bit và có lợi khi sử dụng địa chỉ cổng để nhập nhiều lần.
- Ví dụ : MOV DX,378h
IN AL,DX

6. Lệnh OUT – Ghi cổng IO

- Dạng lệnh : OUT *immed8,accum*
OUT *DX,accum*
- Giải thích : [cổng IO] \leftarrow *accum*
- Tác động cờ : **OF DF IF SF ZF AF PF CF**

--	--	--	--	--	--	--	--	--
- Xuất dữ liệu từ thanh ghi bộ tích lũy AL hoặc AX ra cổng xuất nhập có địa chỉ 8 bit là số tức thời *immed8* hay có địa chỉ 16 bit trong thanh ghi DX.
- Ví dụ : OUT 20h,AL
MOV DX,2F8h
OUT DX,AL

Nhóm lệnh số học

7. Lệnh ADD – lệnh cộng 8, 16bit

- Dạng lệnh : ADD *reg,reg* ADD *reg,immed*
 ADD *mem,reg* ADD *mem,immed*
 ADD *reg,mem* ADD *accum,immed*
 - Giải thích : $\text{thd} \leftarrow \text{thd} + \text{thn}$
 - Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	x	x	x
 - Cộng toán hạng nguồn vào toán hạng đích. Kết quả cất vào toán hạng đích.
 - Ví dụ : ADD CX,SI ; CX $\leftarrow CX + SI$
 ADD DH,BL ; DH $\leftarrow DH + BL$
 ADD [1000h],BX ; [1001h,1000h] $\leftarrow [1001h,1000h] + BX$
 ADD [2000h],CL ; [2000h] $\leftarrow [2000h] + CL$
 ADD AL,[0000h] ; AL $\leftarrow AL + [0000h]$
 ADD BYTE PTR [SI+8],5 ; [SI+8] $\leftarrow [SI+8] + 05h$
8. Lệnh ADC – lệnh cộng thêm cờ nhớ
- Dạng lệnh : ADC *reg,reg* ADC *reg,immed*
 ADC *mem,reg* ADC *mem,immed*
 ADC *reg,mem* ADC *accum,immed*
 - Giải thích : $\text{thd} \leftarrow \text{thd} + \text{thn} + CF$
 - Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	x	x	x
 - Cộng toán hạng đích với toán hạng nguồn với cờ nhớ. Kết quả cất vào toán hạng đích. ADC dùng cho phép cộng 2 số có chiều dài nhiều byte.
 - Ví dụ : ADC BX,AX ; BX $\leftarrow BX + AX + CF$
 ADC BYTE PTR [1000h],7Ah ; [1000h] $\leftarrow [1000h] + 7Ah + CF$

9. Lệnh SUB – lệnh trừ 8,16 bit

- Dạng lệnh : SUB *reg,reg* SUB *reg,immed*
 SUB *mem,reg* SUB *mem,immed*
 SUB *reg,mem* SUB *accum,immed*
- Giải thích : $\text{thd} \leftarrow \text{thd} - \text{thn}$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	x	x	x
- Trừ toán hạng đích cho toán hạng nguồn. Kết quả cất vào toán hạng đích.
- Ví dụ : SUB DL,AL ; DL $\leftarrow DL - AL$
 SUB CX,[DI] ; CX $\leftarrow CX - [DI+1,DI]$
 SUB BP,4 ; BP $\leftarrow BP - 4$

10. Lệnh SBB – lệnh trừ thêm với cờ nhớ

- Dạng lệnh : SBB reg,reg SBB reg,immed
 SBB mem,reg SBB mem,immed
 SBB reg,mem SBB accum,immed
- Giải thích : $thd \leftarrow thd - thn - CF$
- Tác động cờ : OF DF IF SF ZF AF PF CF

x			x	x	x	x	x
---	--	--	---	---	---	---	---
- Trừ toán hạng đích cho toán hạng nguồn và cờ nhớ. Kết quả cắt vào toán hạng đích.
- Ví dụ : SBB SI,BX ; $SI \leftarrow SI - BX - CF$
 SBB BYTE PTR [BX],2 ; $[BX+1,BX] \leftarrow [BX+1,BX] - 2 - CF$

11. Lệnh MUL – lệnh nhân

- Dạng lệnh : MUL reg MUL mem
- Giải thích : **Toán hạng nguồn 8 bit thì :** $AX \leftarrow AL * thn8$
Toán hạng nguồn 16 bit thì : $DX AX \leftarrow AX * thn16$
- Tác động cờ : OF DF IF SF ZF AF PF CF

x			?	?	?	?	x
---	--	--	---	---	---	---	---
- Nhân hai số không dấu 8 bit hay 16 bit. Số bit thực hiện được xác định bằng chiều dài của toán hạng nguồn.
 - ♣ Phép nhân 8 bit : thực hiện nhân AL với toán hạng nguồn, kết quả 16 bit cắt trong thanh ghi AX.
 - ♣ Phép nhân 16 bit : thực hiện nhân AX với toán hạng nguồn, kết quả 32 bit cắt trong 2 thanh ghi DX và AX. DX giữ 16 bit cao, AX giữ 16 bit thấp.
- Ví dụ : Nếu AL=5, CH=4, sau khi thực hiện lệnh

$$\begin{aligned} & \text{MUL CH} \\ & \text{ta có } AX = AL * CH = 0014h. \\ & \text{Nếu } AX=500h, [1001h,1000h]=401h, \text{sau khi thực hiện lệnh} \\ & \text{MUL WORD PTR [1000h]} \\ & \text{ta có } DXAX = AX * [1001h,1000h] = 500h * 401h = 00140500h \\ & \text{Nghĩa là } DX=0014h \text{ và } AX=0500h. \end{aligned}$$

12. Lệnh DIV – lệnh chia

- Dạng lệnh : DIV reg DIV mem
- Giải thích : **Toán hạng nguồn 8 bit thì :** $AL \leftarrow (AX / thn8)$
 $AH \leftarrow$ số dư của $(AX / thn8)$
- Toán hạng nguồn 16 bit thì :** $AX \leftarrow (DXAX / thn16)$
 $DX \leftarrow$ số dư của $(DXAX / thn16)$

- Tác động cờ : OF DF IF SF ZF AF PF CF

?		?	?	?	?	?
---	--	---	---	---	---	---

- Chia hai số không dấu.
- Nếu toán hạng nguồn là thanh ghi hay bộ nhớ 8 bit, thực hiện chia số 16 bit trong thanh ghi AX cho toán hạng nguồn 8 bit. Kết quả 8 bit cát trong thanh ghi AL. Số dư 8 bit cát trong thanh ghi AH.

13. Lệnh INC – lệnh tăng 1 đơn vị

- Dạng lệnh : INC reg INC mem
- Giải thích : **thđ** \leftarrow **thđ + 1**
- Tác động cờ : OF DF IF SF ZF AF PF CF

x			x	x	x	x
---	--	--	---	---	---	---
- Tăng tức là cộng 1 vào toán hạng đích nhưng không ảnh hưởng cờ nhớ.
- Ví dụ : INC CH
INC WORD PTR [1000h]

14. Lệnh DEC – lệnh giảm 1 đơn vị

- Dạng lệnh : DEC reg DEC mem
- Giải thích : **thđ** \leftarrow **thđ - 1**
- Tác động cờ : OF DF IF SF ZF AF PF CF

x			x	x	x	x
---	--	--	---	---	---	---
- Giảm tức là trừ 1 vào toán hạng đích nhưng không ảnh hưởng cờ nhớ.
- Ví dụ : DEC AX
DEC BYTE PTR [SI][2000h]

15. Lệnh CMP – lệnh so sánh

- Dạng lệnh :

CMP	reg,reg	CMP	reg,immed
CMP	mem,reg	CMP	mem,immed
CMP	reg,mem	CMP	accum,immed
 - Giải thích : **thđ - thn**
 - Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	x	x	x
 - So sánh. Thực hiện trừ toán hạng đích cho toán hạng nguồn, không lưu lại kết quả mà chỉ giữ lại tác động của phép trừ lên các cờ.
 - Ví dụ :

CMP	AL,8	; AL - 8
CMP	WORD PTR [1000h], 3	; [1001h,1000h] - 3

Nhóm lệnh logic

16. Lệnh TEST lệnh kiểm tra

- Giải thích : *thđ AND thn.*

- Tác động cờ :	OF DF IF SF ZF AF PF CF
	0 x x ? x 0

- Và luận lý hai toán hạng nhưng không giữ lại kết quả mà chỉ lập các cờ. Xóa cờ nhớ và cờ tràn về 0. Thường dùng để kiểm tra bit. Lúc đó toán hạng nguồn là một mặt nạ bit cần thiết.
 - Ví dụ : TEST DX,1 ; kiểm tra bit 0
 TEST BYTE PTR [2000h],10000000b ; kiểm tra bit 7

- Dạng lệnh :	TEST reg,reg	TEST reg,immed
	TEST mem,reg	TEST mem,immed
	TEST reg,mem	TEST accum,immed

- Giải thích : *thđ AND thn.*

- Tác động cờ :	OF	DF	IF	SF	ZF	AF	PF	CF
	0			x	x	?	x	0

- Và luận lý hai toán hạng nhưng không giữ lại kết quả mà chỉ lập các cờ. Xóa cờ nhớ và cờ tràn về 0. Thường dùng để kiểm tra bit. Lúc đó toán hạng nguồn là một mặt nạ bit cần thiết.

- Vi du : TEST DX,1 ; kiểm tra bit 0
TEST BYTE PTR [2000h],10000000b ; kiểm tra bit 7

18. Lệnh AND – lệnh và bít

- Dạng lệnh : AND reg,reg AND reg,immed
AND mem,reg AND mem,immed
AND reg,mem AND accum,immed
- Giải thích : $thd \leftarrow thd \text{ AND } thn.$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	?	x	0
- Vài luận lý. Xóa cờ nhớ về 0.
- Ví dụ : AND CH,AH
AND [SI],DX
AND BYTE PTR [1000h],10000000b
AND AX,0FFF0h

19. Lệnh OR - lệnh hoặc bít

- Dạng lệnh : OR reg,reg OR reg,immed
OR mem,reg OR mem,immed
OR reg,mem OR accum,immed
- Giải thích : $thd \leftarrow thd \text{ OR } thn.$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	?	x	0
- Hay luận lý. Xóa cờ nhớ về 0.
- Ví dụ : OR DL,CH
OR BP,[2000h]
OR WORD PTR [1000h],000Fh

20. Lệnh XOR lệnh hoặc loại trừ bít (cộng modul 2)

- Dạng lệnh : XOR reg,reg XOR reg,immed
XOR mem,reg XOR mem,immed
XOR reg,mem XOR accum,immed
- Giải thích : $thd \leftarrow thd \text{ XOR } thn.$
- Tác động cờ :

OF	DF	IF	SF	ZF	AF	PF	CF
x			x	x	?	x	0
- Hay ngoại luận lý. Xóa cờ nhớ về 0.
- Ví dụ : XOR DL,80h ; đảo bit 7
XOR [2000h],AL ; [2000h] \leftarrow [2000h] XOR AL

21. Lệnh NOT – lệnh đảo bít

- NOT reg NOT mem
- Giải thích thđ = bù 1 của thđ

(đảo tất cả các bít của thđ rồi gán lại cho thđ)

- Cờ tác động: không
- Ví dụ: NOT AL

NOT WORD PTR [BX]

Nhóm lệnh điều khiển

Lệnh CALL – gọi chương trình con

- CALL Tên chương trình con
- Ví dụ: CALL INIT

INIT:

MOV AX,0000h

MOV BP, 0000h

RET

- RET: Lệnh kết thúc chương trình con

23. Lệnh JMP – lệnh nhảy không điều kiện

- JMP tên nhãn
- Nhảy đến thực hiện lệnh ngay sau tên nhãn
- Ví dụ:

MOV AL, 00h

MOV BL, 00h

NEXT:

INC AL

.....

INC BL

JMP NEXT

24. Lệnh LOOP – lệnh lặp với số lần lặp trong CX

- LOOP tên nhãn
- CX=CX-1, nếu CX khác 0 nhảy đến thực hiện lệnh sau tên nhãn.
- Ví dụ:

MOV AL, 00h

MOV CX, 05h

NEXT:

INC AL

LOOP NEXT

Kết thúc vòng lặp AL=5

Nhóm liên quan đến ngắn

- Lệnh gọi ngắn INT mm, 00h<=mm<=0FFh

Gọi CT con phục vụ ngắt có số hiệu mm.

Vd int 10h, int 20h, int 21h....

- IRET: Lệnh kết thúc ngắt nó thường đặt ở cuối chương trình con phục vụ nhất.
- CLI: Xóa cờ ngắt IF cấm ngắt cứng vào chân INTR.
- STI: Thiết lập lại cờ ngắt IF.

Nhóm lệnh nhảy có điều kiện

- Cú pháp: Jxxx label
 - xxx là viết tắt của điều kiện
 - label là nhãn để nhảy đến
- Nếu điều kiện được thỏa mãn thì nhảy
 - Nếu không thì tiếp tục thực hiện lệnh tiếp theo
 - JNZ == jump if not zero
- Điều kiện
 - Thanh ghi cờ
- Phạm vi nhảy
 - Không quá 126 bytes
- Các lệnh nhảy có dấu

SYMBOL	Ý nghĩa	DESCRIPTION	CONDITION
JG/JNLE	>	jump if greater than jump if not less than or equal to	ZF=0 and SF=OF
JGE/JNL	>=	jump if greater than or equal to jump if not less or equal to	SF=OF
JL/JNGE	<	jump if less than jump if not greater or equal	SF<>OF
JLE/JNG	<=	jump if less than or equal jump if not greater	ZF=1 or SF<>OF

- Các lệnh nhảy không có dấu

SYMBOL	Ý nghĩa	DESCRITION	CONDITION
JA/JNBE	>	jump if above jump if not below or equal	CF=0 and ZF=0
JAE/JNB	>=	jump if above or equal jump if not below or equal	CF=0
JB/JNAE	<	jump if below jump if not above or equal	CF=1
JBE/JNA	<=	jump if below or equal jump if not above	CF=1 or ZF=1

Các lệnh nhảy 1 cờ

SYMBOL	DESCRITION	CONDITION
JE/JZ	jump if equal jump if equal to zero	ZF=1
JNE/JNZ	jump if not equal jump if not zero	ZF=0
JC	jump if carry	CF=1
JNC	jump if no carry	CF=0
JO	jump if overflow	OF=1
JNO	jump if not overflow	OF=0
JS	jump if sign negative	SF=1
JNS	jump if non-negative sign	SF=0
JP/JPE	jump if parity even	PF=1
JNP/JPO	jump if parity odd	PF=0

Ví dụ: Cho AH=02h, BL=01h

```
MOV CX,0  
CMP AH, BL  
JNLE NEXT  
    MOV CX,01
```

NEXT:

KQ: CX=0;

Ví dụ 2: Tạo vòng lặp

```
MOV CX, 100
```

NEXT:

```
....           ; lặp lại CX=100 lần  
DEC CX
```

JNZ NEXT ; nhảy nếu khác 0

Ví dụ 3: Cộng các số: 1,2,3...,100

```
MOV BX, 1  
MOV AX, 0
```

BACK:

```
ADD AX, BX  
INC BX  
CMP BX, 100
```

JLE BACK ; nhảy nếu <=

4. Mô tả các lệnh if, for, while...

Cấu trúc IF – THEN – END IF

IF (Condition) THEN

 Thực hiện lệnh nếu Condition = true

END IF

Ví dụ:

```
; IF (AX<0)  
    CMP AX,0  
    JNL END_IF ; nhảy nếu không bé hơn (>= )  
; THEN  
    MOV CL,01h ; AX <0  
END_IF:
```

Cấu trúc IF – THEN – ELSE – END IF

IF Condition THEN

 Thực hiện ở đây nếu Condition=True

ELSE

 Thực hiện ở đây nếu Condition=False

END_IF

IF AL<=BL THEN
 character in display AL
ELSE
 display character in BL
END_IF

```
MOV AH,2
;IF AL<=BL
CMP AL,BL ; AL<=BL?
JNLE ELSE ; nhảy nếu >
;THEN
MOV DL, AL
JMP END_IF
ELSE:
MOV DL, BL
END_IF:
INT 21H
```

Tìm BL=max (CL, CH)

org 100h

CMP cl,ch

JL behon

 MOV bl, cl

 JMP exit

behon:

 MOV bl, ch

exit:

end

Cấu trúc lặp FOR

- LOOP label

Lặp khi CX khác 0

Số đếm trong thanh ghi CX giảm đi 1

- Mẫu thực hiện lệnh LOOP

 MOV CX, 20 ; gán cho CX số lần lặp là 20

TOP:

 ... ; thân vòng lặp ở đây

LOOP TOP

 ... ; các lệnh tiếp theo sau vòng lặp

5. Vào/ra BF, MH với ngắt 21h

1. **Hàm 01:** Đọc 1 ký tự (có hiện) từ bàn phím

```
mov AH, 01
```

```
int 21h
```

- Output: AL= mã ASCII của ký tự đã gõ,
AL= 0 nếu gõ vào phím chức năng.

2. **Hàm 02:** Hiện 1 ký tự lên màn hình

```
mov AH, 02
```

```
mov DL, mã ASCII của ký tự cần hiển thị
```

```
int 21h
```

Hàm 09: Hiện 1 xâu ký tự kết thúc bởi '\$' lên màn hình: Chức năng AH = 09; DX = địa chỉ offset của xâu ký tự; của ngắt 21h

```
mov ah, 09h
```

```
lea dx, str
```

```
int 21h
```

```
int 20h
```

```
str db "Xin chao cac ban hoc vien$"
```

6. Các ví dụ minh họa.

Ví dụ 1: Các lệnh sau đây sẽ tính tổng nội dung của 100 ô nhớ (100 byte nhớ) trong bộ nhớ, bắt đầu tại địa chỉ 0A00:0120. Kết quả được lưu vào word nhớ ngay trước vùng nhớ này

Ví dụ 1: Các lệnh sau đây sẽ tính tổng nội dung của 100 ô nhớ (100 byte nhớ) trong bộ nhớ, bắt đầu tại địa chỉ 0A00:0120. Kết quả được lưu vào word nhớ ngay trước vùng nhớ này.

```
Mov Ax, 0A00h
```

```
Mov DS, Ax
```

```
Mov SI, 0120h; trả DS:SI tới nguồn 0A00:0120
```

```
Mov DI, SI ; trả DS:DI tới nguồn 0A00:0120
```

```
Sub DI,2 ; trả DS:DI về word trước vùng nhớ nguồn
```

```
Mov Cx, 100
```

```
Mov Dx, 0 ; DX chứa tổng
```

TTong: Add Dx, byte PTR [SI] ; cộng vào DX

Inc SI ; trả vào ô nhớ tiếp

Loop TTong

Mov Word PTR [DI], DX

Ví dụ 2: Copy toàn bộ 16 kí tự từ biến Xau1 vào biến Xau2. Giả sử Xau1 và Xau2 đã được khai báo trước như sau: Xau1 DB "Khoa KTDK HVKTQS"

Xau2 DB 18 Dup ("")
Lea SI, Xau1
Lea DI, Xau2
Mov Cx, 18

Lap_Copy:

Mov A1, [SI]
Mov [DI], A1
Inc SI
Inc DI
Loop Lap_Copy

Ví dụ 3: Viết chương trình dạng COM: Nhập vào một kí tự thường, chương trình sẽ in ra kí tự in hoa tương ứng.

ORG 100h

Mov Ah, 09h ; xuất thông báo 1
Lea Dx, TB1
Int 21h
Mov Ah, 01 ; chờ nhập ký tự
Int 21h
Mov Bl, Al
Mov Ah, 09h ; xuất thông báo 2
Lea Dx, TB2
Int 21h
Mov Ah, 02 ; xuất k/tự và chuyển in hoa
Mov Dl, Bl
Sub Dl, 20h
Int 21h
Int 20h
TB1 DB 'Nhập vào một kí tự thường: \$'
TB2 DB 0Ah,0Dh,'Kí tự hoa tương ứng: \$'
end

Ví dụ 4: Viết chương trình dạng COM: Nhập vào hai số (số thứ nhất: nhỏ hơn 5; số thứ hai: nhỏ hơn hoặc bằng 5), sau đó in ra tổng của hai số vừa nhập.

```
ORG    100h
Jmp    Main
TBN1  DB  'Nhap so hang thu nhat (nho hon 5): $'
TBN2  DB  0Ah,0Dh,'Nhap so hang thu hai (<=5): $'
TBX   DB    0Ah,0Dh,'Tong cua hai so la: $'
```

Main:

```
Mov    Ah, 09h
Lea    Dx, TBN1
Int    21h      ; xuất thông báo 1 nhập số thứ 1
Mov    Ah, 01 ; nhận số thứ nhất
Int    21h
Mov    Bl, A1
Sub    Bl, 30h ; chuyển thành số
Mov    Ah, 09h ; Thông báo nhập số thứ 2
Lea    Dx, TBN2
Int    21h
Mov    Ah, 01 ; nhận số thứ hai
Int    21h
Sub    Al, 30h ; chuyển thành số
Add    Bl, Al ; cộng hai số
Mov    Ah, 09h ; hiển thị thông báo kết quả
Lea    Dx, TBX
Int    21h
Mov    Ah, 02 ; hiển thị kết quả
Mov    Dl, B1
Add    Dl, 30h ; chuyển thành ascii
Int    21h
Int    20h ; kết thúc CT
```

End

Ví dụ 5. Đọc xâu kí tự từ bàn phím

```
Lea    DI, LuuXau ; địa chỉ lưu xâu
Mov    Cx, 5      ; số ký tự đọc
```

```

Mov      Ah, 01h
Nhap_Xau:
    Int     21h
    Mov     [DI], AL
    Inc     DI
Loop    Nhap_Xau
    mov     ah, 09h ; xuất xâu ra màn hình
    Mov     Byte PTR [DI], '$'
    Lea     DX, LuuXau
    int 21h
    int 20h
LuuXau   DB     30 Dup (" ")

```

BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn

Trương Đăng Khoa

ĐỀ CƯƠNG BÀI GIẢNG
(Dùng cho 3 tiết giảng)
Học phần: CTMT
Bộ môn: Tự động và KTT
Khoa: KTĐK

GIÁO VIÊN

Nguyễn Trần Hiệp

Bài giảng: Bộ nhớ và hệ thống hỗ trợ bộ nhớ

Chương 2 mục 2.1;

Tiết thứ: 13-18

Tuần thứ: 5 -6

Mục đích, yêu cầu: Nguyên lý làm việc, cơ chế giải mã bộ nhớ để xác định ô nhớ mà BVXL sẽ truy cập

- Hình thức tổ chức dạy học:

Lý thuyết, bài tập, tự học, tự nghiên cứu

- Thời gian:

Lý thuyết: 6 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 12 tiết

- Địa điểm:

Giảng đường do P2 phân công

- Nội dung chính:

2.1. Bộ nhớ và hệ thống hỗ trợ bộ nhớ

Định nghĩa: Bộ nhớ trên máy tính là nơi lưu trữ các chương trình điều hành hoạt động của máy tính và là môi trường cho Hệ điều hành và các chương trình ứng dụng hoạt động.

Bộ nhớ trên máy tính là tổ hợp của nhiều vi mạch nhớ được sắp xếp với nhau theo một thứ tự nhất định.

Bộ nhớ của máy tính và vi mạch nhớ có những điểm khác biệt

Dung lượng bộ nhớ máy tính:	Dung lượng vi mạch nhớ:
Được xác định bằng độ rộng Bus địa chỉ	Được xác định bằng số chân địa chỉ của vi mạch.
Ô nhớ trên máy tính được tính bằng đơn vị Byte.	Ô nhớ của vi mạch nhớ được xác định bằng đơn vị bit
Máy tính có n bits địa chỉ: Có dung lượng bộ nhớ là 2^n ô nhớ	Vì mạch nhớ có k chân địa chỉ: Có dung lượng là 2^k ô nhớ
Dung lượng bộ nhớ $2^n * 1$ (2,4) Byte	Dung lượng vi mạch nhớ $2^k * 1$ (4,8) bit

Dung lượng của vi mạch nhớ được tính bằng bit, do đó chân số liệu của vi mạch nhớ có thể là 1, 2, 4 hoặc 8 bit, Vì vậy để tạo ra được 1 ô nhớ tương ứng với bộ nhớ của máy tính, phải kết nối các vi mạch nhớ với nhau để tạo được ô nhớ có độ dài là 1, 2, 4, 8 bytes.

2.2.1 Cơ sở bộ nhớ bán dẫn

Bộ nhớ chỉ đọc ROM (Read Only Memory)

ROM là vi mạch nhớ có đặc điểm thông tin ghi trong nó không bị mất đi khi ta ngắt nguồn điện cung cấp cho vi mạch. Vi mạch nhớ ROM thường được sử dụng để chứa các chương trình điều khiển hệ thống. Trong máy tính IBM PC ROM lưu trữ chương trình điều hành hoạt động của máy tính được gọi là ROM BIOS.

Theo sự phát triển của công nghệ có nhiều loại vi mạch ROM khác nhau như PROM, EPROM, EEPROM và Flash ROM, tùy theo từng ứng dụng mà người ta sử dụng loại ROM nào cho phù hợp.

Để ghi thông tin lên vi mạch nhớ ROM người ta phải dùng các thiết bị và chương trình đặc biệt (ghi ROM). Thao tác ghi bộ nhớ của CPU không thể ghi được thông tin lên ROM.

Bộ nhớ ghi đọc RAM (Random Access Memory)

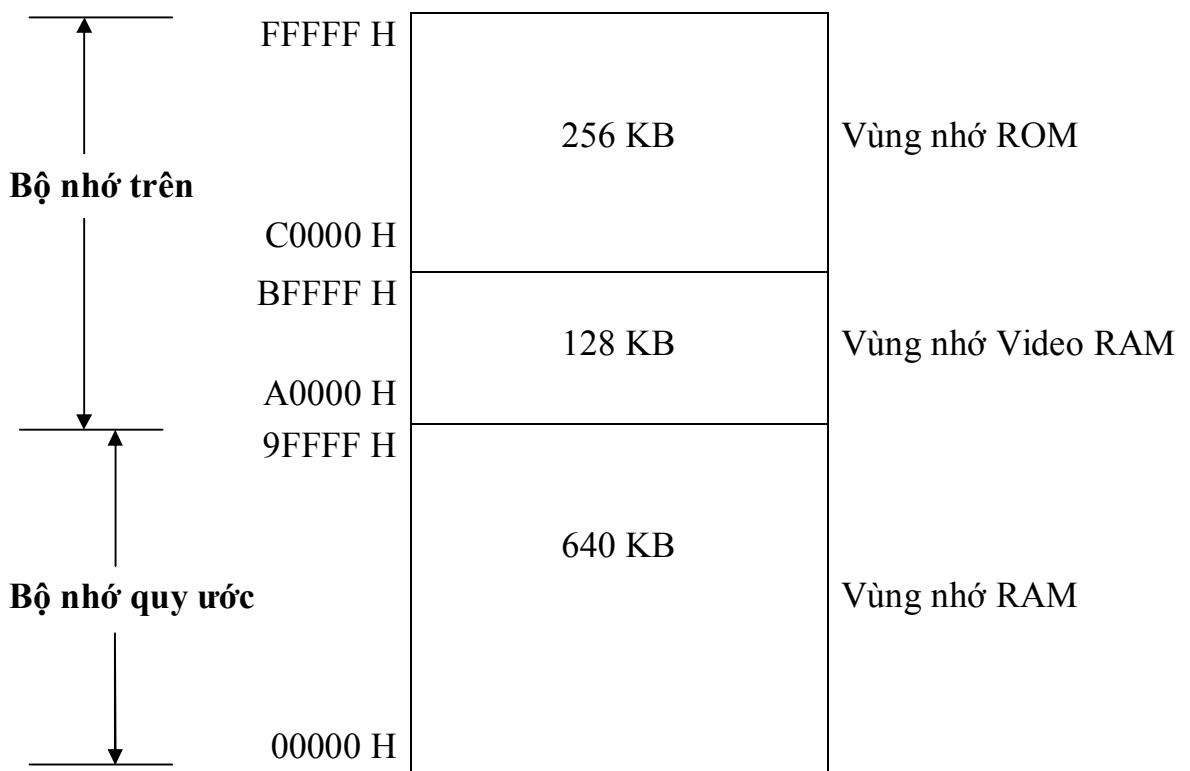
Còn gọi là bộ nhớ thay đổi, khi ta cắt nguồn điện thì toàn bộ thông tin trong vi mạch nhớ RAM sẽ bị xóa hết. CPU có thể ghi và đọc thông tin của RAM thông qua thao tác ghi và đọc bộ nhớ.

Bộ nhớ RAM là môi trường làm việc của HDH và chương trình ứng dụng, do đó cần có dung lượng lớn để đáp ứng được các yêu cầu của HDH và chương trình. Khi vi mạch RAM có dung lượng lớn nghĩa là có mật độ linh kiện cao. Về nguyên tắc để biểu diễn 1 bit thông tin cần đến 6 Tranzitor, loại RAM đó còn được gọi là RAM tĩnh, vi mạch RAM tĩnh có dung lượng lớn thì số Tranzitor sẽ lớn, khi đó các vấn đề cần giải quyết là: tỏa nhiệt cho vi mạch và bố trí dây dẫn trong nội tại vi mạch, điều đó làm cho vi mạch RAM tĩnh với dung lượng lớn có giá thành rất cao.

Để tạo ra vi mạch RAM có dung lượng lớn và giá thành hạ người ta sử dụng 1 Tranzitor và 1 tụ điện để biểu diễn 1 bit thông tin, do đó có thể chế tạo được vi mạch RAM có dung lượng lớn có giá thành thấp hơn RAM tĩnh. Loại RAM đó gọi là RAM động, tuy nhiên do hiện tượng dò của tụ điện nên thông tin biểu diễn bằng RAM động sẽ tự bị mất đi theo thời gian. Để khắc phục hiện tượng này sau 1 khoảng thời gian nhất định người ta phải tiến hành nạp lại điện áp của tụ điện, quá trình đó được gọi là quá trình làm tươi, khi đó không thể ghi và đọc thông tin trên vi mạch RAM động, do đó tốc độ làm việc của RAM động thấp hơn tốc độ làm việc của RAM tĩnh, thêm vào đó mạch điều khiển của RAM động cũng phức tạp hơn do phải có thêm mạch làm tươi.

2.2.2 Bản đồ bộ nhớ trong máy tính IBM PC

BVXL 8086/88 của Intel để xây dựng máy tính IBM PC XT có 20 tín hiệu địa chỉ do đó máy tính IBM PC có dung lượng bộ nhớ là 2^{20} ô nhớ có địa chỉ bắt đầu từ 00000 H đến FFFF H và bộ nhớ của máy tính phải bao gồm cả bộ nhớ ROM và RAM do đó không gian bộ nhớ trên máy tính được chia thành các vùng như sau:



Vùng nhớ dành cho ROM BIOS bao giờ cũng nằm trên đỉnh cao nhất của không gian nhớ. Do đó khi khởi động máy tính, BVXL sẽ có trạng thái như sau:

$$CS = FFFF; DS = 0000; SS = 0000; ES = 0000; IP = 0000$$

Ô nhớ mà BVXL sẽ truy cập khi khởi động là CS:IP là FFFF:0000 tương ứng với địa chỉ vật lý là FFFF0 H.

2.2.3 Tổ chức giải mã bộ nhớ

Theo nguyên lý Von Neumann, nếu BVXL làm việc với bộ nhớ ($M/\overline{IO}=1$ – truy cập bộ nhớ) thì tại 1 thời điểm CPU chỉ có thể đọc hoặc ghi thông tin tại 1 ô nhớ. Để tìm được ô nhớ nào sẽ trao đổi thông tin với BVXL người ta sử dụng phương pháp giải mã cụ thể như sau:

Không gian địa chỉ hóa bộ nhớ trên máy tính có n bits địa chỉ, khi đó bus địa chỉ của máy tính sẽ có các đường dây tín hiệu từ A_0 đến A_{n-1}

Không gian địa chỉ hóa của vi mạch nhớ là k bits địa chỉ ($k < n$), khi đó vi mạch nhớ sẽ có các chân địa chỉ từ A_0 đến A_{k-1} .

Khi đó sẽ tiến hành giải mã theo 2 cấp để xác định ô nhớ sẽ trao đổi thông tin với BVXL:

$A_{n-1}, A_{n-2}, A_{n-3}, \dots, A_k$	$A_{k-1}, A_{k-2}, \dots, A_0$
Nối đến mạch giải mã	Nối đến các chân tương ứng của vi mạch nhớ

Các tín hiệu địa chỉ $A_{n-1}, A_{n-2}, A_{n-3}, \dots, A_k$ nối đến mạch giải mã để tìm được vi mạch nhớ nào chứa ô nhớ sẽ trao đổi thông tin với BVXL

Các tín hiệu $A_{k-1}, A_{k-2}, \dots, A_0$ nối đến vi mạch nhớ cho phép giải mã bên trong vi mạch nhớ để tìm ra ô nhớ sẽ trao đổi thông tin với BVXL.

Mạch giải mã của máy tính thường sử dụng vi mạch 74LS 138.

Mạch giải mã bên trong vi mạch nhớ: gồm có mạch giải mã hàng và cột để tạo được ma trận điểm có 2^k phần tử, mỗi một phần tử sẽ tương đương với tín hiệu chọn 1 ô nhớ.

2.2.4 Bộ nhớ trên các máy tính hiện đại

Hiện nay, bộ nhớ RAM trên các máy tính hiện đại được chia thành các bank nhớ, người ta chia ra các loại bank nhớ RAM như sau:

SIMM RAM: Bộ nhớ một hàng chân,

DIMMRAM: Bộ nhớ 2 hàng chân, được chia thành 3 loại:

SDRAM - Synchronous DRAM;

DDR RAM - Double Data Rate DRAM,

RD RAM - RAM Bus DRAM

2.2.5. Bộ nhớ mở rộng

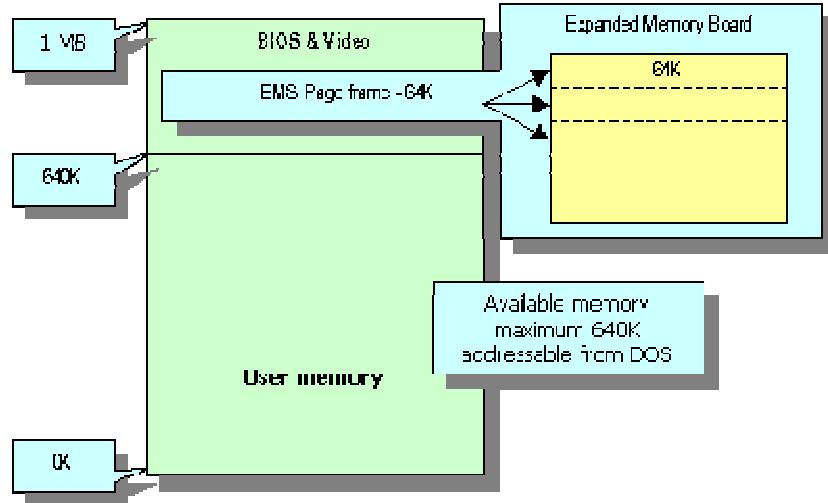
Khi các chương trình ứng dụng phát triển, không gian 640 KB của bộ nhớ làm việc (RAM) trở nên quá ít, do vậy Lotus và Intel đã xuất phương pháp mở rộng để quản lý bộ nhớ RAM nằm ngoài không gian chuẩn của máy tính.

Theo phương pháp này, sử dụng vùng nhớ có địa chỉ E0000 H đến EFFFF H (dung lượng 64 Kb) làm cửa sổ để quy chiếu ra vùng nhớ mở rộng, cửa sổ này được chia làm 4 khung trang, mỗi khung có dung lượng 16 KB. Việc xác định địa chỉ của vùng nhớ mở rộng sử dụng 1 chương trình là EMS. EXE

Theo sự phát triển các Version HĐH, chương trình EMS cũng phát triển theo.

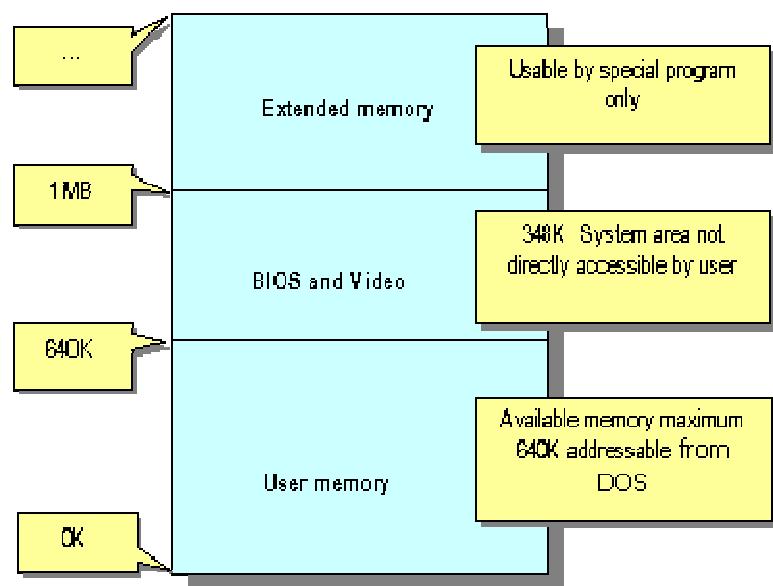
EMS 3.0 chỉ cho phép vùng nhớ mở rộng chưa số liệu mà không chứa mã lệnh.

EMS 4.0 cho phép vùng nhớ mở rộng chứa được cả số liệu và mã lệnh và có thể quản lý vùng nhớ mở rộng đến 32 MB.



2.2.6 Bộ nhớ phát triển

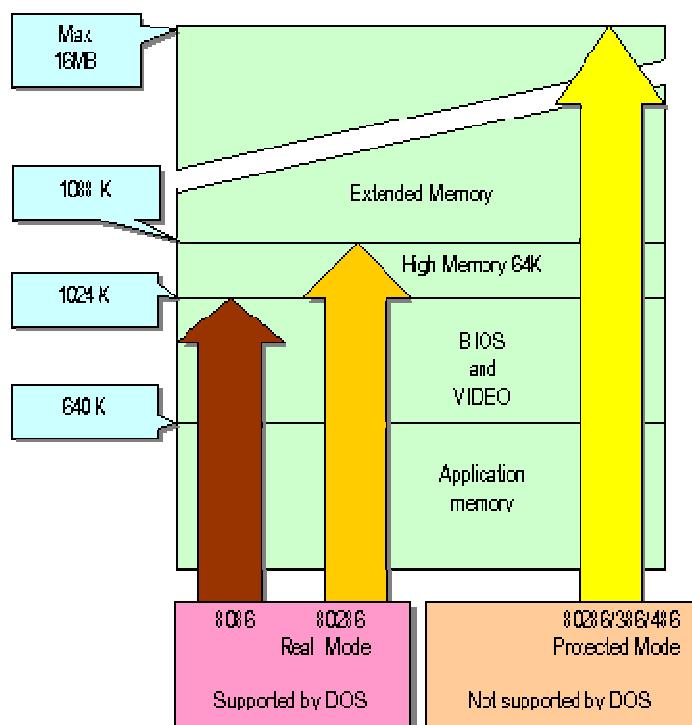
BVXL 80286 có 24 bits địa chỉ cho phép địa chỉ hóa được 2^{24} ô nhớ, ở các BVXL tiến tiến hơn còn có khả năng địa chỉ hóa nhiều hơn, do đó có thể có 1 vùng không gian bộ nhớ có địa chỉ gắn liền với bộ nhớ tiêu chuẩn 1 MB ban đầu của máy tính,



Người ta sử dụng một phần mềm là XMS. EXE để quản lý không gian bộ nhớ như vừa trình bày. Như vậy Bộ nhớ phát triển chỉ có thể tồn tại trên các máy tính sử dụng BVXL từ 80286 trở đi.

2.2.7 Vùng nhớ cao

Do có sự tồn tại của tín hiệu địa chỉ A20, do đó ở chế độ thực, các BVXL từ 80286 trở đi đều có khả năng truy cập 65520 bytes ô nhớ có địa chỉ từ 100000H đến 10FFEF. Không gian này được gọi là vùng nhớ cao (HMA)



Để mở rộng không gian bộ nhớ làm việc cho việc thực hiện các chương trình ứng dụng, HĐH sẽ chuyển các lệnh nội trú của HĐH từ vùng nhớ của bộ nhớ RAM tiêu chuẩn lên vùng nhớ cao, như vậy làm tăng không gian bộ nhớ làm việc cho các chương trình ứng dụng.

Bài tập: chia bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 57-93

Câu hỏi bài 5:

- Phân biệt dung lượng Bộ nhớ máy tính và dung lượng của Vi mạch nhớ.

2. Với vi mạch nhớ RAM có dung lượng $2^{10} * 1$ bit. Cần bao nhiêu vi mạch để ghép nối chúng với nhau để tạo được 1 vùng nhớ có dung lượng $2^{10} * 8$ bits? Trình bày sơ đồ ghép nối.

3. Thiết kế mạch giải mã RAM trên máy tính, với vùng địa chỉ trong khoảng (00000 H – 9FFFF H)

- Giả thiết sử dụng vi mạch nhớ RAM có dung lượng: $2^{12} * 8$ bits.

- Sử dụng vi mạch giải mã là 74138. Hãy cho biết cần dùng bao nhiêu vi mạch giải mã và vi mạch nhớ RAM như đã cho trong giả thiết để điền đầy không gian nhớ của vùng nhớ quy ước?

4. Thiết kế mạch giải mã ROM trên máy tính, với vùng địa chỉ trong khoảng (C0000 H – FFFFF H)

- Giả thiết sử dụng vi mạch nhớ ROM có dung lượng: $2^{11} * 8$ bits.

- Sử dụng vi mạch giải mã là 74138. Hãy cho biết cần dùng bao nhiêu vi mạch giải mã và vi mạch nhớ ROM như đã cho trong giả thiết để điền đầy không gian nhớ của bộ nhớ ROM?

- Hãy cho biết địa chỉ của ROM BIOS nằm trong khoảng giá trị nào?

5. Bản chất của bộ nhớ phát triển và bộ nhớ mở rộng qua hình vẽ.

6. Giải thích ý nghĩa của Slide trình bày về không gian nhớ của máy tính trước và sau khi chạy lệnh DEVICE=C:\DOS\HIMEM.SYS và DOS = HIGH.

7. Giải thích tại sao dung lượng của vùng nhớ cao lại là 65520 bytes.

8. Xác định vùng địa chỉ của ROM BIOS nếu sử dụng vi mạch nhớ ROM có dung lượng 8k *8 bits.

9. Thế nào là chương trình chạy thường trú, lệnh nội trú, lệnh ngoại trú./

10. Bộ nhớ phát triển có thể tồn tại trên máy tính IBM PC XT sử dụng Bộ VXL 8086/8088 hay không? Tại sao?

11. Hãy cho biết địa chỉ đầu tiên của:

Bộ nhớ mở rộng

Bộ nhớ phát triển khi chưa thực hiện lệnh
DEVICE=C:\DOS\HIMEM.SYS và DOS = HIGH.

Bộ nhớ phát triển khi đã thực hiện lệnh
DEVICE=C:\DOS\HIMEM.SYS và DOS = HIGH.

**BỘ MÔN DUYỆT
Chủ nhiệm Bộ môn**

Truong Đăng Khoa

ĐỀ CƯƠNG BÀI GIẢNG
(Dùng cho 3 tiết giảng)

GIÁO VIÊN

Nguyễn Trần Hiệp

Bài giảng: Lập trình ASM trên máy tính IBM PC

Chương 2 mục 2.2;

Tiết thứ: 19-21

Tuần thứ: 7

Mục đích, yêu cầu: Ôn lại kỹ năng lập trình ASM làm công cụ để tìm hiểu sâu vào bên trong máy tính IBM PC

- Hình thức tổ chức dạy học:

Lý thuyết, bài tập, tư học, tư nghiên cứu

- Thời gian:

Lý thuyết: 6 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- Địa điểm:

Giảng đường do P2 phân công

- Nội dung chính:

2.3.1. Lập trình ASM trên máy tính IBM PC

2.3.1.1. Tập lệnh của BVXL 8086/88

2.3.1.2. Chương trình ASM

Đặc điểm của ASM: 1 lệnh ASM tương ứng với 1 lệnh máy do đó ASM thường được dùng cho lập trình quá trình thời gian thực.

Chỉ thực hiện trên nền HĐH DOS

Không có các công cụ hỗ trợ như soạn thảo, dịch và chạy trong một môi trường.

File có phần mở rộng .com

File có phần mở rộng .exe

2.3.1.3 Lập trình ASM sử dụng debug

Chức năng: Debug là một lệnh ngoại trú của của HĐH nó cho phép ta có thể can thiệp đến từng bit trong các thanh ghi của BVXL, đến từng ô nhớ, từng sector trên ổ đĩa.

Thực hiện chạy debug trên máy tính với HĐH DOS và HĐH Windows

Debug có 19 lệnh cho phép thao tác với các thanh chỉ của BVXL, hiển thị và thay đổi nội dung của bộ nhớ, soạn thảo và chạy 1 đoạn chương trình trong bộ nhớ, dịch thuận và dịch ngược 1 đoạn chương trình.v.v.

2.3.2 Lập trình ASM dùng Emu 80x86

Hướng dẫn cài đặt Emu 80x86 trên máy tính,

Đặc điểm: Emu 80x86 là phần mềm giả lập các lệnh của BVXL Intel 80x86 trong môi trường Windows

Có các công cụ hỗ trợ trên môi trường Windows

Lập trình ASM thường sử dụng một số hàm ngắt của HĐH

Hàm ngắt int 21h với các giá trị khác nhau của ah

Hàm ngắt int 10h với các giá trị khác nhau của ah

Hàm ngắt int 16h với các giá trị khác nhau của ah

Bài tập: chũa bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước Slide bài giảng từ trang 50-69

Câu hỏi bài 6:

1. Phân biệt thế nào là file *.com và file *.exe.
2. Trình bày tổ chức của 1 segment khi thực hiện file *.com.
3. Viết 1 chương trình ASM có phần mở rộng .COM thực hiện công việc sau:

Bước 1. Hiển thị ra màn hình dòng chữ: Ký tự nhan duoc la:

(Sử dụng ngắt int 21h với ah = 09h)

Bước 2. Chờ nhận 1 phím từ bàn phím và hiển thị ký tự đó ra màn hình.
(Sử dụng ngắt int 21h với ah = 01h)

Bước 3: Xuống dòng và làm lại bước 1.

(Sử dụng int 21h với ah = 02) mã ASCII của ký tự xuống dòng là 0ah.

Mã ASCII của ký tự về đầu dòng là 0dh

Chương trình sẽ kết thúc khi bạn ấn phím ECS có mã ASCII là 1bh.

4. Viết chương trình in ra màn hình 6 ký tự đầu tiên của bảng mã ASCII mỗi ký tự cách nhau hai khoảng trắng. [Đoan chương trình](#)

5. Viết chương trình hiển thị họ và tên của sinh viên tại vị trí hiện thời của con trỏ.

Sử dụng ngắt int 21h với ah=09h.. [Đoan chương trình](#)

6. Viết chương trình nhận 1 ký tự từ bàn phím và hiển thị ra màn hình. Chương trình sẽ kết thúc khi ấn phím ESC (1bh)

Sử dụng ngắt int 21h; với ah = 08h.

Bài giảng: Màn hình và Card điều khiển màn hình

Chương 3 mục 3.1;

Tiết thứ: 22-24

Tuần thứ: 8

Mục đích, yêu cầu: Giới thiệu về các thiết bị vào ra điện hình trên máy tính IBM PC

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 3 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- **Địa điểm:**

Giảng đường do P2 phân công

- **Nội dung chính:**

3.4. Các thiết bị vào ra điện hình

3.4.1. Màn hình và card điều khiển màn hình

Màn hình là thiết bị ra cho phép hiển thị kết quả của các chương trình ứng dụng và HĐH. Là thành phần chính trong giao tiếp trực tiếp người – máy.

Card điều khiển màn hình thực chất là một hệ VXL có chung vũng nhớ với Video RAM của máy tính, khi cần hiển thị thông tin ra màn hình, CPU của máy

tính sẽ ghi thông tin ra vùng nhớ Video RAM và Hệ VXL của Card điều khiển sẽ đọc thông tin trên vùng nhớ Video RAM, biến đổi nó để hiển thị ra màn hình.

Các chuẩn màn hình và card điều khiển màn hình:

Card CGA tương ứng với màn hình CGA

Card EGA tương ứng với màn hình EGA

Card VGA và SVGA tương ứng với màn hình VGA và SVGA

Card điều khiển màn hình và màn hình có 2 chế độ làm việc: Chế độ văn bản (TEXT) và chế độ đồ họa (GRAPHIC)

3.4.1.1 Lập trình chế độ văn bản

Ở chế độ văn bản, đối tượng điều khiển của Card điều khiển màn hình là 1 ma trận điểm cho phép hiển thị được 1 ký tự có mã ASCII. Ở chế độ văn bản có hai loại độ phân giải của ký tự trên màn hình: 2000 ký tự/màn hình ($80*25$) và 1000 ký tự/màn hình ($40*25$).

Khi đó thông tin của 1 ký tự được hiển thị trên màn hình gồm:

Vị trí của ký tự trên màn hình.

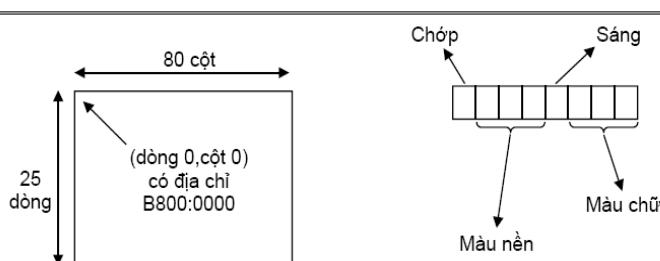
Mã ASCII của ký tự được hiển thị.

Màu sắc và độ sáng của ký tự sẽ được hiển thị (thuộc tính).

Mỗi ký tự trên màn hình cần 2 byte thông tin về nó: Byte đầu tiên chứa mã ASCII của ký tự và byte thứ hai chứa thuộc tính của ký tự.

Với độ phân giải màn hình là 2000 ký tự/màn hình sẽ cần dung lượng là 4000 byte trong bộ nhớ Video RAM để hiển thị.

Byte thuộc tính:



Lập trình chế độ văn bản - Sử dụng ngắt int 10h với:

Ah = 00h: Đặt chế độ cho card màn hình và màn hình.

Ah = 02h: Xác lập vị trí con trỏ; Dh – tọa độ hàng; Dl – tọa độ cột

• • • • •

3.4.1.2 Lập trình chế độ đồ họa

Ở chế độ đồ họa, đối tượng của card điều khiển màn hình là từng điểm sáng trên màn hình.

Thông tin một điểm sáng trên màn hình bao gồm: Vị trí của nó trên màn hình và màu sắc của nó.

Để cung cấp thông tin về vị trí của điểm sáng trên màn hình phải sử dụng 2 thanh ghi 16 bits (CX cung cấp tọa độ cột và DX cung cấp tọa độ hàng)

Để hình ảnh ở chế độ đồ họa có chất lượng cao, phải dùng nhiều bit thông tin để mã hóa màu (8 bits: 256 màu, 16 Bits màu, 24 bits và 32 bits) do đó 1 điểm sáng kèm theo nó là số bits mã hóa màu tương ứng. Như thế toàn màn hình tùy theo độ phân giải (800*600 hoặc 1024 * 768) sẽ bao gồm rất nhiều bits thông tin, do vậy bộ nhớ dùng cho Card điều khiển màn hình ở chế độ đồ họa là rất lớn.

Lập trình chế độ đồ họa – Sử dụng ngắn int 10h

Thiết lập chế độ đồ họa: chế độ 06h, 13h

Tọa độ của điểm ảnh được chứa trong thanh ghi CX tọa độ cột, DX tọa độ hàng,

Tắt, sáng điểm ảnh ah = 0ch – Al = 0 tắt điểm ảnh, Al = 1 bật điểm ảnh.

Al có các giá trị khác nhau sẽ cho điểm ảnh có màu sắc khác nhau.

Card tăng tốc đồ họa AGP.

Ở chế độ đồ họa, để biểu diễn một trang màn hình có độ phân giải cao, màu sắc rõ nét sẽ cần một dung lượng nhớ rất lớn. Khi thực hiện xử lý hình ảnh chuyển động thì trong 1 giây cần phải đưa đến 30 trang lên màn hình, lượng thông tin trao đổi giữa máy tính và Card điều khiển màn hình sẽ rất lớn (dải thông đòi hỏi cao). Thực tế bus PCI của máy tính không đủ dải thông để truyền lượng thông tin này, do vậy người ta thiết kế 1 giao diện truyền số liệu giữa Card điều khiển màn hình và CPU của máy tính trên cơ sở tốc độ truyền số liệu giữa CPU và RAM. Card màn hình trên giao diện đó gọi là card tăng tốc đồ họa AGP.

Xem thêm trong tài liệu.

Trên cơ sở dải thông của bus PCI là 266MBps, người ta thiết kế Card AGP với các tốc độ truyền:

$$2X = 2 * 266MBps$$

$$4X = 4 * 266MBps$$

$$8X = 8 * 266MBps$$

Ngoài việc tăng dải thông, để tăng tốc độ truyền AGP còn cải tiến cả giao thức truyền số liệu.

Hiện nay khi mà PCI được nâng cấp PCI Exp có dải thông đến 16X thì người ta không sử dụng AGP nữa mà sử dụng chuẩn PCI Exp cho các máy tính.

Câu hỏi bài 7:

1. Phân biệt chế độ đồ họa và chế độ văn bản.
2. Trình bày cấu trúc byte thuộc tính trong Video RAM ở chế độ Text.
3. Vì sao nói ở chế độ đồ họa, khi độ phân giải tăng, số bit mã hóa màu tăng thì bộ nhớ Video Ram phải tăng.
4. Lập trình vẽ 1 hình tam giác ABC trên màn hình với các tọa độ sau: A(50,50), B(0,100), C(100,100). Thay đổi AL với các giá trị khác nhau để nhận được các màu khác nhau.
5. Lập trình thiết lập màn hình làm việc ở chế độ 02, đưa con trỏ về vị trí (hàng 15 – 0fh, cột 20 -14h), hiển thị ra màn hình 03 ký tự “C” với thuộc tính 42h (nền đỏ chữ xanh lục) – sử dụng int 10h với ah=09h. Sau đó thực hiển thị ra màn hình 01 ký tự “A” – Sử dụng int 10h với ah=0eh. Nhận xét về sự xuất hiện của ký tự “A” trên màn hình và rút ra kết luận về dịch chuyển con trỏ khi sử dụng ngắt int 10h với ah=09h.
6. Lập trình thực hiện hiển thị ra màn hình họ tên của bạn tại vị trí hàng 10-0ah, cột 30 – 1ch, với thuộc tính 2fh (nền xanh, chữ trắng). – Sử dụng int 10h với ah=13h.
7. Lập trình thiết lập chế độ làm việc của màn hình ở chế độ phân giải 80*25 (chế độ 2) rồi sau đó thiết lập lại ở chế độ phân giải 40*25 (chế độ 0). So sánh độ rộng của màn hình trong 2 chế độ này, giải thích tại sao. Sử dụng EMU 80x86.
8. Lập trình chế độ văn bản (chế độ 02), đặt con trỏ về vị trí hàng 20 – 14h, cột 2. Chờ nhận từ phím và hiển thị phím được án ra màn hình với thuộc tính nền đỏ chữ vàng (4eh), dịch chuyển con trỏ sang vị trí bên cạnh và chờ nhận tiếp một ký tự mới. Chương trình sẽ kết thúc nếu nhập phím ESC có mã ASCII là 1bh.
9. Lập trình chế độ văn bản, (chế độ 01) với ah=13h hiển thị lên màn hình dòng chữ HVKTQS có thuộc tính 74h (nền trắng – chữ đỏ) bắt đầu từ vị trí hàng 20 (14h) và cột 32 (20h).
10. Lập trình vẽ một đường dọc trên màn hình từ vị trí hàng 10, cột 20 đến vị trí hàng 30 cột 20 với độ dày của đường là 4 điểm ảnh.

Bài tập và kiểm tra

Chương 1,2,3 mục ;

Tiết thứ: 25-27 Tuần thứ: 9

Mục đích, yêu cầu: Chữa bài tập và kiểm tra

- Hình thức tổ chức dạy học:

Lý thuyết bài tập tư học tư nghiên cứu

- Thời gian:

Jý thuyết: 3 tiết; bài tập: 0 tiết; tư học, tư nghiên cứu: 6 tiết

- Địa điểm:

Điều kiện:

- NỘI DUNG CHÍNH -

Bài tân chương 1.

- Bài tập chương 1:**

 1. Trình bày cấu trúc của máy tính theo nguyên lý Von Neumann, chỉ ra nhược điểm cơ bản của nguyên lý này?
 2. Những thao tác nào là thao tác cơ bản nhất trong hoạt động của máy tính theo nguyên lý Von Neumann.

3. Trình bày công dụng của các tín hiệu địa chỉ, số liệu, điều khiển trong hệ bus của máy tính, ý nghĩa độ rộng bus địa chỉ và số liệu.
 4. Tại sao nói chia BVXL 8086/88 thành 2 thành phần cơ bản EU và BIU và với việc có cấu trúc hàng đợi lệnh là một cải tiến đáng kể của Intel.
 5. Các thanh ghi trong nội bộ BVXL 8086/88 có 16 bits, để biểu diễn được địa chỉ 20 bits người ta dùng giải pháp nào?
 6. Nguyên tắc kết hợp giữa các thanh ghi đoạn với con trỏ lệnh và các thanh ghi con trỏ chỉ số để biểu diễn địa chỉ 20 bits. Cách xác định địa chỉ vật lý từ địa chỉ logic
 7. Thế nào là địa chỉ vật lý, và địa chỉ logic. Khi nào sử dụng phương pháp địa chỉ vật lý, khi nào sử dụng phương pháp địa chỉ logic. Một địa chỉ vật lý có tối đa bao nhiêu địa chỉ logic.
 8. BVXL 8086/88 có chung các chân địa chỉ, số liệu, làm cách nào để tách được tín hiệu trên kênh địa chỉ số liệu (trình bày giản đồ thời gian của quá trình đọc/ghi và sơ đồ hệ thống hỗ trợ bus). Vai trò của các tín hiệu ALE và $\overline{\text{DEN}}$.
 9. Ưu điểm lớn nhất của BVXL 16 bits 8086/88 của Intel là gì?
 10. Trình bày cấu trúc đường ống, nêu ưu, nhược điểm của cấu trúc này?
- Bài tập chương 2:**
1. Trình bày bản đồ bộ nhớ trên máy tính IBM PC XT, khi khởi động máy tính ô nhớ nào sẽ được truy cập đầu tiên.
 2. Người ta dùng giải pháp nào để xác định được ô nhớ sẽ trao đổi số liệu với CPU? Trên máy tính IBM PC XT sử dụng vi mạch nhớ ROM có dung lượng $2^{13} \times 8$ bits thì các tín hiệu địa chỉ nào sẽ được nối từ bus địa chỉ đến chân vi mạch nhớ và tín hiệu địa chỉ nào sẽ được nối đến mạch giải mã.
 3. Thiết kế mạch giải mã bộ nhớ RAM trên máy tính.
 - Địa chỉ vùng nhớ RAM (00000 H – 9FFFF H)
 - Giải thích sử dụng vi mạch nhớ RAM có dung lượng: $2^{12} \times 8$ bits.
 4. Thế nào là file *.com và file *.exe, Cấu trúc của bộ nhớ làm việc khi thực hiện những file dạng *.com và file *.exe.
 5. Bản chất của bộ nhớ mở rộng, người ta sử dụng công cụ nào để quản lý bộ nhớ mở rộng.
 6. Bản chất của bộ nhớ phát triển, người ta sử dụng công cụ nào để quản lý bộ nhớ phát triển.

7. Thế nào là vùng nhớ cao, người ta sử dụng vùng nhớ cao vào mục đích gì để nâng cao hiệu suất làm việc của máy tính.

Bài giảng 8: Hệ thống hỗ trợ vào/ra

Chương 3 mục 3.2;

Tiết thứ: 28-30 Tuần thứ: 10

Mục đích, yêu cầu: Vào ra song song trên máy tính sử dụng vi mạch lập trình.

- *Hình thức tổ chức dạy học:*

Lý thuyết, bài tập, tự học, tự nghiên cứu

- Thời gian:

Lý thuyết: 3 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- Địa điểm:

Giảng đường do P2 phân công

- Nội dung chính:

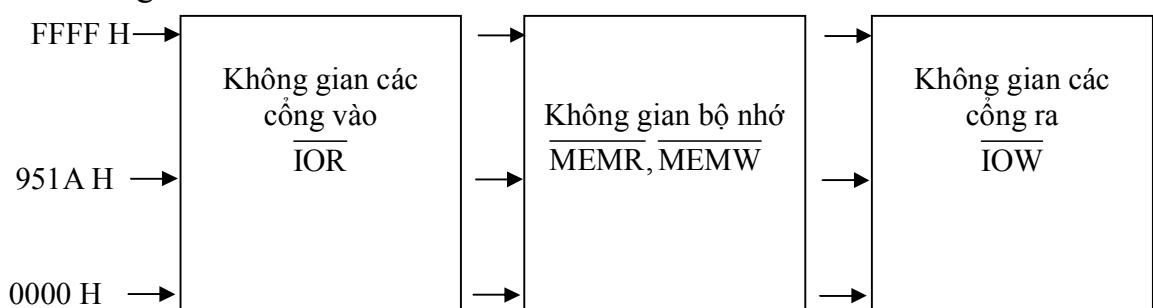
3.1. Tổ chức vào/ra

3.1.1 Các lệnh vào ra

CPU kết nối với các thiết bị ngoại vi được gọi là kết nối vào/ra. Có 2 dạng kết nối vào/ra:

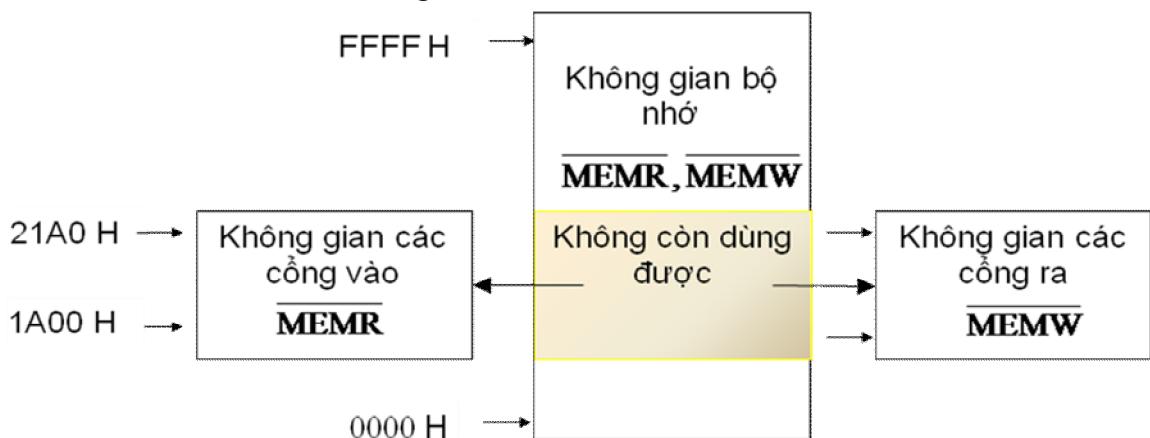
Với BVXL có hỗ trợ vào/ra: khi đó BVXL có phần cứng và tập lệnh hỗ trợ vào ra. Cụ thể sẽ có tín hiệu điều khiển vào/ ra riêng IOR, IOW và có các lệnh vào IN và lệnh ra OUT riêng biệt. Bộ VXL của Intel là 1 ví dụ về BVXL có hỗ trợ vào ra.

Với BVXL có hỗ trợ vào/ra, khi đó bên cạnh không gian bộ nhớ, máy tính còn có thêm không gian các cổng vào và không gian các cổng ra có thể có cùng địa chỉ nhưng có các tín hiệu điều khiển khác nhau.



Đặc điểm của BVXL này là sẽ sử dụng thanh ghi AL làm thanh ghi thao tác nếu trao đổi số liệu 8 bits và thành ghi AX là thanh ghi thao tác nếu trao đổi số liệu 16 bits.

Còn có BVXL không có hỗ trợ vào/ra riêng biệt, khi đó không gian vào/ra sẽ phải sử dụng 1 phần không gian bộ nhớ làm không gian vào/ra. Lúc đó các lệnh điều khiển vào/ra sẽ là các lệnh trao đổi giữa BVXL với bộ nhớ, tín hiệu đọc thiết bị vào là MEMR và tín hiệu ghi thiết bị ra là MEMW.



Các lệnh vào/ra khi đó có các chế độ địa chỉ:

Chế độ địa chỉ trực tiếp: không gian công vào và không gian công ra được địa chỉ hóa bởi 8 bits địa chỉ từ A₀ đến A₇ do đó ta có thể có được 2⁸ cổng vào và 2⁸ cổng ra có địa chỉ từ 00 H đến FF H.

Cú pháp lệnh vào: IN AL, #port; địa chỉ cổng nằm trực tiếp trong lệnh;
IN AX, # port

Cú pháp lệnh ra: OUT #Port, AL
OUT #port, AX

Chế độ địa chỉ gián tiếp: Không gian công vào và cổng ra được địa chỉ hóa bởi 16 bits địa chỉ từ A₀ đến A₁₅ do đó ta có thể có được 2¹⁶ cổng vào và 2¹⁶ cổng ra có địa chỉ từ 0000 H đến FFFF H. Trong các lệnh vào/ra lúc này địa chỉ cổng được nằm trong thanh ghi DX,

Cú pháp lệnh vào: MOV DX,#port (địa chỉ 16 bits)
IN AL, DX hoặc:

MOV DX, #Port
IN AX, DX

Cú pháp lệnh ra MOV DX, #port
OUT DX, AL hoặc
MOV DX, #port
OUT DX, AX

3.1.2 Nối ghép vào ra song song

Để nối ghép với các cổng vào/ra có thể có 2 phương pháp:

Sử dụng các mạch vào/ra đơn giản

Mạch vào sử dụng bộ đếm 1 chiều là vi mạch 74LS 244,

Mạch nối ghép vào , mạch giải mã có thêm tín hiệu \overline{IOR} ,

Mạch ra sử dụng chốt số liệu là vi mạch 74LS 374. Mạch giải mã có thêm tín hiệu \overline{IOW} .

Sử dụng vi mạch lập trình 8255.

3.1.3 Vi mạch lập trình 8255A

Vi mạch lập trình 8255 là vi mạch lập trình cho phép nối ghép vào/ra song song. Cấu trúc của vi mạch gồm:

Có 8 chân số liệu từ D₀ đến D₇ cho phép kết nối với các chân tương ứng của bus số liệu,

Có 2 chân địa chỉ A_1 và A_0 nối trực tiếp từ bus địa chỉ đến vi mạch cho phép lựa chọn $2^2 = 4$ thanh ghi bên trong vi mạch,

Có 3 cổng vào/ra có thể lập trình được, mỗi cổng có 8 bits lần lượt là Cổng A gồm các chân từ PA_0 đến PA_7 , Cổng B gồm các chân từ PB_0 đến PB_7 , Cổng C gồm các chân từ PC_0 đến PC_7 . Bên trong vi mạch có 4 thanh ghi có thể lập trình được là thanh ghi 8 bit tương ứng với cổng A được gọi là thanh ghi PA, thanh ghi 8 bits tương ứng với cổng B được gọi là thanh ghi PB và thanh ghi 8 bits tương ứng với cổng C được gọi là thanh ghi PC. Một thanh ghi để xác lập chế độ làm việc của 8255 được gọi là thanh ghi điều khiển. Việc lựa chọn các thanh ghi kể trên với các tín hiệu A_1 A_0 như sau:

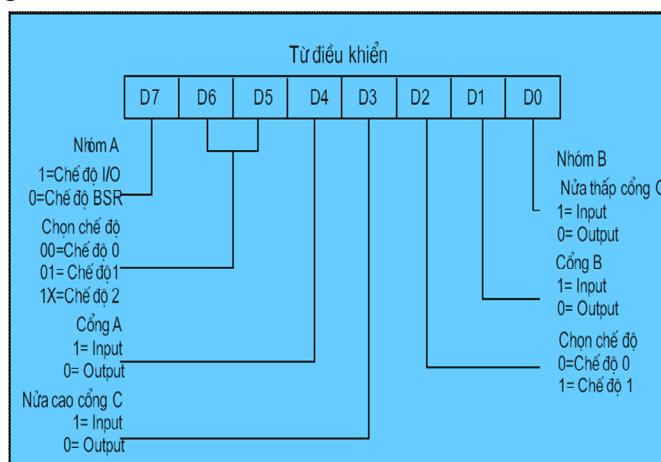
A_1	A_0	
0	0	Thanh ghi cổng A được chọn
0	1	Thanh ghi cổng B được chọn
1	0	Thanh ghi cổng C được chọn
1	1	Thanh ghi điều khiển được chọn.

Thiết lập chế độ cho các cổng của 8255 được thông qua thanh ghi điều khiển, Các cổng của 8255 được thiết lập thành 2 nhóm làm việc có thể lập trình độc lập qua thanh ghi điều khiển;

Nhóm 1 gồm tất cả các bits của cổng A và 4 bits cao của cổng C (PC_4 đến PC_7) có 3 chế độ làm việc là mode 0, mode 1 và mode 2.

Nhóm 2 gồm tất cả các bits của cổng B và 4 bits thấp của cổng C (PC_0 đến PC_3) có 2 chế độ làm việc là mode 0 và mode 1.

Thiết lập chế độ làm việc cho các nhóm của 8255 thông qua việc ghi vào thanh ghi điều khiển như sau:



Mode 0 còn được gọi là chế độ vào/ra đơn giản; khi đó các cổng PA, PB, PC có thể được lập trình để tạo ra 1 vi mạch 74LS244 khi nó là cổng vào và 1 vi mạch 74LS 374 khi nó là cổng ra.

Mode 1 và mode 2 được gọi là vào ra có bắt tay; khi đó các cổng PA và PB có thể được lập trình thành các cổng vào ra còn các bít cao của PC với nhóm 1 và các bít thấp của PC với nhóm 2 được sử dụng làm các tín hiệu điều khiển.

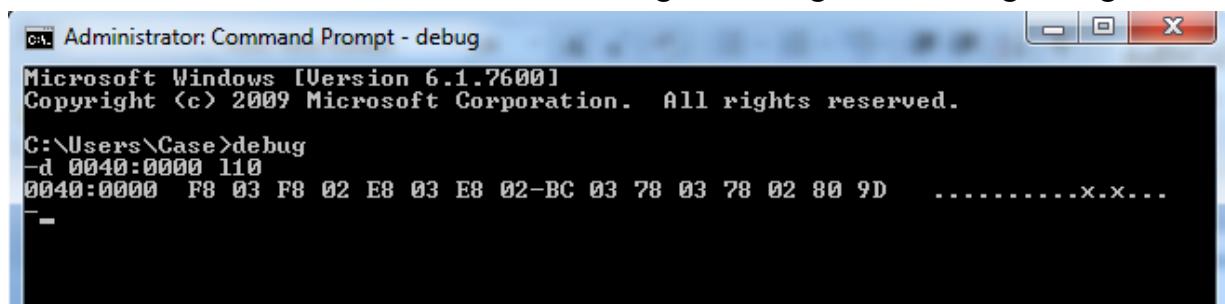
Vi mạch 8255 trên máy tính IBM PC

Máy tính IBM PC sử dụng 2 vi mạch 8255. Vi mạch thứ nhất nằm trên Mother Board của máy tính có địa chỉ giải mã là 60 H được lập trình ở mode 0 cổng PA có nhiệm vụ đọc cổng bàn phím, cổng PB là cổng ra cho phép điều khiển các ngoại vi, cổng PC là cổng vào biểu diễn cấu hình của máy tính.

Vi mạch 8255 thứ 2 nằm trên Card vào/ra với địa chỉ giải mã 0378 H và 0278 H là cổng LPT của máy tính. Cổng LPT có thể nối đến rất nhiều thiết bị như máy in, scanner, HDD Box.v.v. do đó 8255 sẽ được lập trình để phù hợp từng loại thiết bị nối đến cổng LPT.

Khi khởi động máy tính, HDH sẽ kiểm tra sự có mặt của các cổng LPT và ghi địa chỉ của các cổng này vào vùng nhớ có địa chỉ từ ô nhớ 00408h đến ô nhớ 0040fh.

Có thể kiểm tra sự có mặt của các cổng LPT bằng lệnh D trong debug:



```
Administrator: Command Prompt - debug
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Case>debug
-d 0040:0000 110
0040:0000 F8 03 F8 02 E8 03 E8 02-BC 03 78 03 78 02 80 9D .....x.x...
---
```

Chế độ lập và xóa bít của 8255

Vi mạch 8255 còn có một chế độ làm việc đặc biệt; Nếu bít D7 của thanh ghi điều khiển có giá trị 1 thì ta có thể tạo ra trên các chân tương ứng của cổng PC các mức logic 1 hay không tùy thuộc vào việc chọn các bits từ D1 đến D3 của thanh ghi điều khiển.

D7	D6	D5	D4	D3	D2	D1	D0	
1	x	x	x	0	0	0	1	PC0 có mức 1

				0	0	0	0	PC0 có mức 0
				0	0	1	1	PC1 có mức 1
				0	0	1	0	PC1 có mức 0
				0	1	0	1	PC2 có mức 1
				0	1	0	0	PC2 có mức 0
				0	1	1	1	PC3 có mức 1
				0	1	1	0	PC3 có mức 0
				1	1	1	1	PC7 có mức 1
				1	1	1	0	PC7 có mức 0

Lập trình cổng LPT sử dụng ngắt int 17h

Để đơn giản khi lập trình vào/ra song song trên các cổng LPT HĐH cung cấp ngắt int 17h với các chức năng phụ thuộc vào giá trị của ah như sau:

Int 17h với Ah = 0: Khởi tạo cổng LPT, giá trị cổng được ghi trong DX, (DX = 0 - 3) tương ứng với LPT 1- LPT 4.

Int 17h với Ah = 1: Đưa 1 ký tự trong AL ra cổng LPT, giá trị cổng được ghi trong DX, (DX = 0 – 3) tương ứng với LPT 1- LPT 4.

Int 17h với Ah = 2: Nhận trạng thái cổng LPT về thanh ghi AH, giá trị cổng được ghi trong DX, DX = 0 -3 tương ứng với LPT 1- 4.

AH	Ý nghĩa các bit	AH	Ý nghĩa các bit
D7	1=Printer busy, 0=printer not busy	D3	1=I/O error
D6	1=Acknowledge from printer	D2	Not used
D5	1=Out of paper signal	D1	Not used
D4	1=Printer selected	D0	Time out error

3.2.1 Bộ định thời 8253

Máy tính sử dụng một tín hiệu đồng bộ để đồng bộ mọi hoạt động. BVXL làm việc ở tần số cao nhất còn các ngoại vi làm việc ở tần số thấp hơn. Thiết bị tạo

thời gian cho máy tính sử dụng vi mạch 8253 được dùng để tạo tần số cho các ứng dụng.

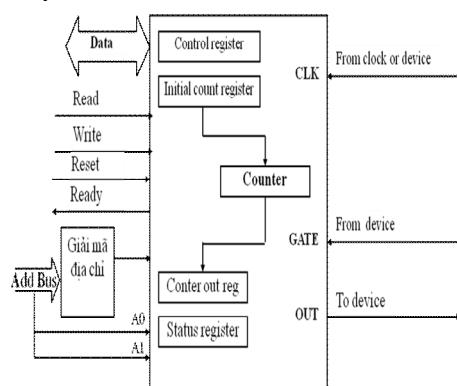
Cấu tạo vi mạch 8253.

Vi mạch lập trình 8253 có 8 chân số liệu từ D₀ đến D₇ cho phép kết nối với các chân tương ứng của bus số liệu,

Có 2 chân địa chỉ A₁ và A₀ nối trực tiếp từ bus địa chỉ đến vi mạch cho phép lựa chọn $2^2 = 4$ thanh ghi bên trong vi mạch,

Bên trong 8253 có 3 bộ đếm với độ dài 16 bits, được gọi là con đếm 1, 2 và 3. cùng với 1 thanh ghi điều khiển 8 bits cho phép thiết lập chế độ làm việc của các con đếm trong 8253.

Một con đếm 16 bits có các đầu vào/ra như sau:



CLK là tín hiệu đầu vào con đếm (Tín hiệu cần chia tần)

Out là tần số ra của con đếm sau khi đã được chia tần.

Gate là tín hiệu khởi động con đếm.

Tín hiệu địa chỉ A₁ và A₀ cho phép lựa chọn con đếm:

A ₁	A ₀	Lựa chọn
0	0	Con đếm 0 (độ dài 16 bits)
0	1	Con đếm 1 (độ dài 16 bits)
1	0	Con đếm 2 (độ dài 16 bits)
1	1	Thanh ghi điều khiển (độ dài 8 bits)

3.2.2 Ghép nối và lập trình 8253

Mỗi con đếm của 8253 được lập trình riêng biệt bằng cách ghi thông tin lên thanh ghi điều khiển:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CS1	CS0	RL1	RL0	M2	M1	M0	BCD

CS1	CS0	Lựa chọn
0	0	Con đếm 0
0	1	Con đếm 1
1	0	Con đếm 2
1	1	Cảm

RL1	RL0	Lựa chọn
0	0	Chốt đầu ra con đếm
0	1	Đọc/Nạp chỉ phần cao của con đếm
1	0	Đọc/Nạp chỉ phần thấp của con đếm
1	1	Đọc/Nạp cả phần thấp và phần cao của con đếm, byte thấp đọc trước

M2	M1	M0	Mode	BCD	
0	0	0	Mode 0	0	Đếm 16 bit
0	0	1	Mode 1	1	Đếm thập phân
x	1	0	Mode 2		
x	1	1	Mode 3		
1	0	0	Mode 4		
1	0	1	Mode 5		

Các chế độ làm việc của con đếm:

Mode 0: Ngắt khi đếm kết thúc, khi chân gate được nâng lên mức 1 con đếm bắt đầu đếm, đầu ra sẽ có mức 0 độ rộng xung ra được tính như sau:

$$\text{Độ rộng xung} = N * T$$

N là số đếm được nạp vào con đếm

T là chu kỳ của xung đưa vào đầu vào CLK

Khi con đếm kết thúc, đầu ra sẽ nâng lên mức cao và được giữ nguyên cho đến khi có một giá trị mới được nạp.

Mode 1: Chế độ xung Sau khi con đếm được nạp, nếu có 1 xung từ 0 lên 1 tác động vào chân gate con đếm bắt đầu đếm và khi đếm hết (đầu ra hạ xuống 0 và nâng lên 1 khi đếm xong).

Mode 2: Phát tín hiệu tỷ lệ

Mode 3: Tạo xung vuông

Mode 4: Tạo xung bằng phẳng mềm

Mode 5: Tạo xung mẫu bằng phần cứng.

3.2.3 Sử dụng 8253 trong máy tính IBM PC

Trong máy tính IBM PC, vi mạch 8253 có địa chỉ giải mã là 40h,

Con đếm 0 có địa chỉ giải mã 40h

Con đếm 1 có địa chỉ giải mã 41h

Con đếm 2 có địa chỉ giải mã 42h

Thanh ghi điều khiển có địa chỉ giải mã 43h

Cả 3 con đếm có đầu vào CLK được nối đến 1 tần số 1,1931817 Mhz lấy tại chân PLK của vi mạch 8284,

Con đếm 0 được sử dụng để tạo ngắt thời gian IRQ0

Con đếm 1 được sử dụng để tạo xung tuần hoàn cho mạch làm tươi DRAM

Con đếm 2 được dùng để tạo âm thanh

Bài tập: chửa bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 102-132

Đọc trước TL1 trang 135-159

Câu hỏi bài 8:

1. Phân biệt thế nào là vào/ra theo không gian riêng và vào/ra theo không gian địa chỉ bộ nhớ. (Hỗ trợ phần cứng, phần mềm, không gian vào/ra, các lệnh vào/ra, tín hiệu điều khiển)

2. Ở các BVXL của Intel sử dụng vào/ra ở chế độ nào? Trình bày chi tiết vào/ra ở chế độ địa chỉ trực tiếp và chế độ địa chỉ gián tiếp?

3. Tại sao nói khi sử dụng vi mạch vào/ra song song lập trình được là 8255 làm việc ở chế độ 0, sẽ tương đương với việc sử dụng các vi mạch đơn giản như mạc chốt dữ liệu 74LS374 hay bộ đếm dữ liệu 74LS244.

4. Tại sao khi thực hiện ghép nối vi mạch 8255 vào hệ Bus, lại có 2 tín hiệu địa chỉ A0 và A1 được nối từ Bus đến vi mạch?

5. Giả sử vi mạch 8255 được nằm trên card ISA của máy tính với địa chỉ giải mã là 0248h. Xác định địa chỉ của các thanh ghi PA, PB, PC và thanh ghi từ điều khiển. Lập trình cho 2 nhóm của vi mạch này làm việc ở mode 0 với PA là cổng ra,

PB, PC là cổng vào. Thực hiện đọc dữ liệu từ PB, xét 2 bits D5, D6. Nếu 2 bits này đồng thời bằng 1 thì gửi dữ liệu đọc được ra cổng PA.

6. Trên máy IBM PC và tương thích, vi mạch 8255 được sử dụng tại những vị trí nào? Trình bày địa chỉ giải mã, chức năng và chế độ làm việc của các vi mạch đó?

7. Thiết kế một mạch nối ghép vào, chế độ địa chỉ trực tiếp, địa chỉ cổng vào 55H để đọc trạng thái các cánh cửa của 1 lớp học sử dụng công tắc tiếp xúc (“1” cửa đóng, “0” cửa mở) sử dụng vi mạch 74244 (bộ đếm vào)

Thiết kế mạch giải mã, tín hiệu điều khiển

Thiết kế mạch nối ghép giữa 74244 với Bus dữ liệu (8 bits D0- D7).

8. Con đếm 0 của vi mạch giải mã 8253 có địa chỉ giải mã là 40h làm việc ở mode 3. Với tần số đầu vào clock 0 là 1,8432 Mhz tính giá trị nạp vào con đếm nếu muốn tần số trên chân out của là 50hz? Viết đoạn chương trình thiết lập chế độ làm việc cho con đếm và nạp giá trị cho con đếm?

Bài giảng 9: Quản lý quá trình vào/ra

Chương 3 mục 3.3;

Tiết thứ: 31-33

Tuần thứ: 11

Mục đích, yêu cầu: Các phương pháp quản lý ngoại vi trên máy tính, cơ chế ngắt và vi mạch điều khiển ngắt cứng 8259

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 3 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- **Địa điểm:**

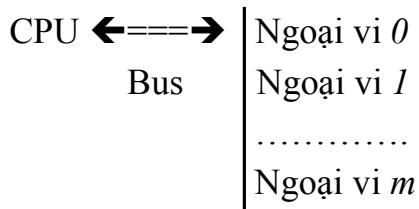
Giảng đường do P2 phân công

- **Nội dung chính:**

3.3.1 Quản lý ngoại vi

Trên máy tính CPU trao đổi dữ liệu với các thiết bị ngoại vi thông qua Bus được gọi là quá trình vào/ra

Có thể có nhiều ngoại vi được nối với máy tính, tuy nhiên tại một thời điểm CPU chỉ có thể trao đổi dữ liệu với 1 trong số các ngoại vi đó.



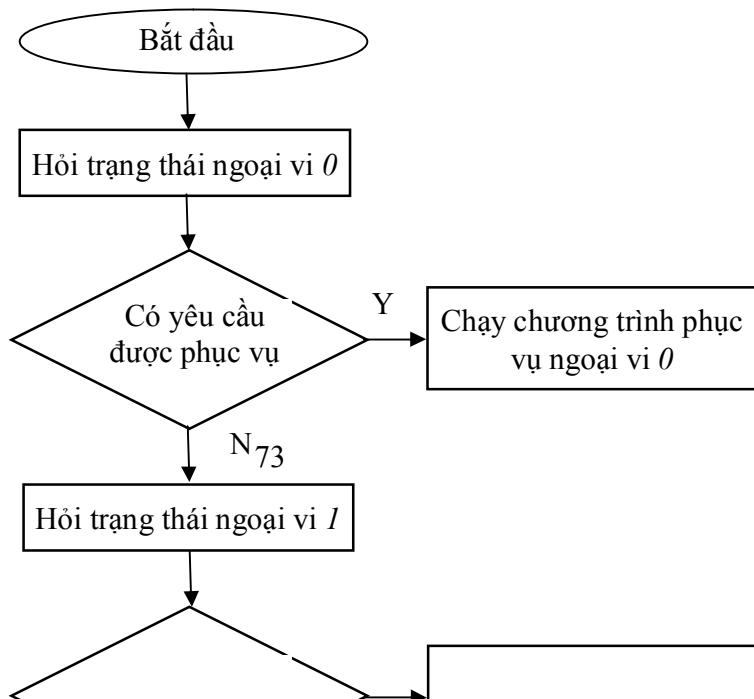
Mỗi ngoại vi được nối với máy tính thực hiện các nhiệm vụ khác nhau, ví dụ Máy in, Scanner, HDD Box.v.v. Do đó, với mỗi ngoại vi khác nhau HĐH cũng phải sử dụng các chương trình điều khiển quá trình trao đổi dữ liệu khác nhau, các chương trình đó có sẵn được gọi là **chương trình phục vụ ngoại vi**. Chúng có thể được nạp sẵn trong bộ nhớ ROM BIOS hay được HĐH nạp vào bộ nhớ RAM trong quá trình khởi động máy tính.

Nhiệm vụ Hệ điều hành là phải xác định ngoại vi nào được chọn để trao đổi dữ liệu với CPU, quá trình xác định đó được gọi là quá trình quản lý ngoại vi hay quản lý vào ra.

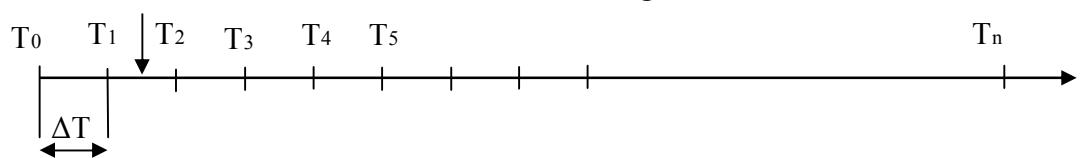
Có 2 nguyên tắc quản lý thiết bị ngoại vi:

Quản lý bằng cách thăm dò: Mỗi thiết bị ngoại vi thường có 1 hay nhiều tín hiệu trạng thái làm việc của nó được biểu diễn bằng các bits trạng thái. Sau một khoảng thời gian nhất định CPU sẽ thực hiện kiểm tra các bits các trạng thái đó để phục vụ. Theo nguyên tắc này, nếu 1 ngoại vi yêu cầu được phục vụ nhưng CPU chưa kiểm tra trạng thái thì nó sẽ chưa đáp ứng yêu cầu của ngoại vi.

Giả sử, máy tính có m ngoại vi được cài đặt, quá trình quản lý ngoại vi theo phương pháp thăm dò sẽ thực hiện như lưu đồ thuật toán sau:



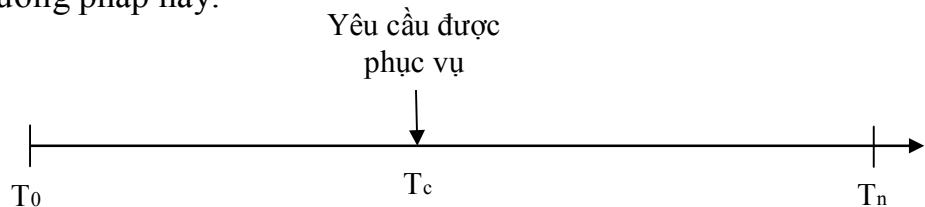
Có thể biểu diễn qua trình thăm dò ngoại vi theo một tiến trình thời gian như sau: Giả sử HĐH thực hiện 1 tiến trình theo thời gian từ T_0 đến T_n



Theo phương pháp thăm dò, sau 1 khoảng thời gian là ΔT HĐH lại thực hiện lại chu kỳ của thuật toán vừa mô tả. Nếu có 1 ngoại vi yêu cầu được phục vụ tại thời điểm giữa T_1 và T_2 , thì nó sẽ phải đợi đến thời điểm T_2 máy tính mới nhận biết được yêu cầu đó. Nghĩa là đáp ứng của máy tính với yêu cầu của ngoại vi là không tức thời.

Nếu trong thời gian từ T_0 đến T_n nếu không có ngoại vi nào yêu cầu được phục vụ, HĐH vẫn phải thực hiện từng đó thao tác, như thế sẽ gây ra lãng phí thời gian và tài nguyên của máy tính.

Quản lý băng ngắt: Ngắt là đáp ứng tức thời của CPU với yêu cầu phục vụ của ngoại vi, nguyên tắc này tận dụng tốt khả năng sử dụng CPU cho xử lý nhiều công việc. Hiện nay tất cả các ngoại vi tiêu chuẩn trên máy tính đều được quản lý bằng phương pháp này.



3.3.2 Các ngắt trên máy tính IBM PC

Cơ chế ngắt: trên máy tính có 2 cơ chế ngắt là ngắt cứng và ngắt mềm. Ngắt cứng là đáp ứng của CPU với yêu cầu được phục vụ của ngoại vi. Ngắt mềm là đáp ứng của CPU với yêu cầu của người lập trình hay yêu cầu do chương trình sinh ra trong quá trình thực hiện chương trình.

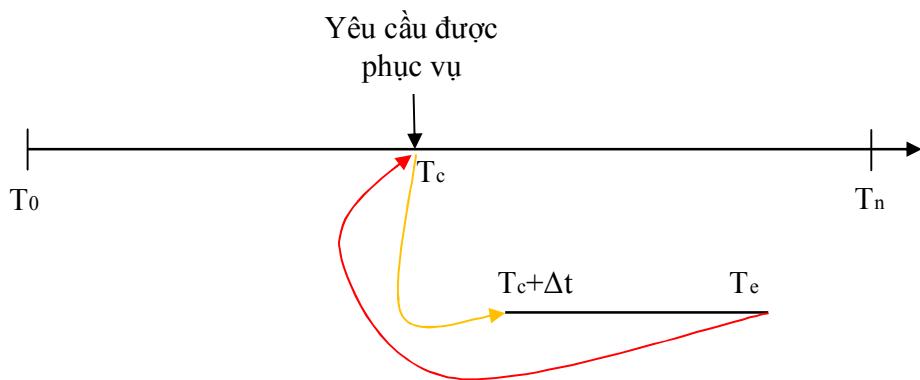
Máy tính có khả năng phục vụ được 256 ngắt được đánh số từ 00 – FF H, thứ tự của mỗi một ngắt như vậy được gọi là 1 số hiệu ngắt. Thứ tự ưu tiên của các ngắt được xác định từ thấp đến cao tức là ngắt có số hiệu 00 H sẽ có mức ưu tiên cao nhất và ngắt có số hiệu FF H sẽ có mức ưu tiên thấp nhất. Trong 256 ngắt đó chỉ có 15 ngắt cứng, còn lại là ngắt mềm và dự trữ. Ý nghĩa của từng ngắt được mô tả trong tài liệu đi kèm máy tính.

Mỗi một ngắt bao giờ cũng có 1 phần mềm để đáp ứng yêu cầu của ngắt đó được gọi là chương trình phục vụ ngắt (ISR). Với 256 ngắt sẽ có 256 số hiệu ngắt và 256 chương trình phục vụ ngắt. Như đã trình bày các chương này có sẵn và được nằm trong bộ nhớ ROM BIOS hoặc RAM của máy tính.

3.3.3 Cơ chế ngắt và phục vụ ngắt

Khi một ngắt phát yêu cầu đến CPU, ngay lập tức CPU sẽ ngừng công việc mà nó đang thực hiện để chạy chương trình phục vụ tương ứng với ngắt đó. Sau khi chương trình phục vụ ngắt kết thúc, máy tính lại khôi phục trạng thái trước khi ngắt và tiếp tục công việc của nó trước khi ngắt.

Δt là thời gian xử lý ngắt (rất ngắn) do vậy có thể nói đáp ứng của CPU với yêu cầu ngắt là tức thời.



Vấn đề đặt ra ở đây là:

- Cắt giữ trạng thái khi bắt đầu thực hiện ngắn tại thời điểm T_c và khôi phục trạng thái sau khi ngắn;
- Bằng cách nào xác định được điểm vào của chương trình phục vụ ngắn và dấu hiệu nào để biết được chương trình phục vụ ngắn đã kết thúc.

Để đáp ứng được những yêu cầu trên người ta đưa ra giải pháp:

- Sử dụng cấu trúc ngăn xếp và con trỏ ngăn xếp;
- Sử dụng bảng vector ngắn;
- Quy định lệnh cuối cùng của một chương trình ngắn là lệnh IRET.

Bảng vector ngắn: Là vùng nhớ trên máy tính có địa chỉ từ 00000H đến 003FFH có dung lượng 1024 ô nhớ được HĐH sử dụng để chứa điểm vào của các chương trình phục vụ ngắn. Điểm vào của một chương trình phục vụ ngắn chính là vị trí của 1 ô nhớ trong không gian bộ nhớ của máy tính. Vị trí đó được xác định chính là địa chỉ vật lý của ô nhớ đó trong không gian bộ nhớ máy tính, khi thực hiện chương trình địa chỉ được biểu diễn dưới dạng địa chỉ logic (*đoạn:độ lệch*). Như vậy điểm vào của một chương trình phục vụ ngắn được biểu diễn bởi 2 thanh ghi 16 bits (4 bytes), với 256 chương trình phục vụ ngắn sẽ cần $4 * 256 = 1024$ bytes => bảng vector ngắn có độ dài 1024 bytes.

Cách ghi điểm vào của các chương trình phục vụ ngắn trong bảng vector ngắn được bố trí như sau:

003FF	Đoạn	Chứa điểm vào của ISR của ngắn có số hiệu là OFF H
003FE		
003FD	Độ lệch	
003FC		

0000C		
0000B	Đoạn	Chưa điểm vào của ISR của ngắt có số hiệu là 02 H
0000A		
00009	Độ lệch	
00008		
00007	Đoạn	Chưa điểm vào của ISR của ngắt có số hiệu là 01 H
00006		
00005	Độ lệch	
00004		
00003	Đoạn	Chưa điểm vào của ISR của ngắt có số hiệu là 00 H
00002		
00001	Độ lệch	
00000		

Nhìn vào quy tắc sắp xếp trên ta thấy ô nhớ trong bảng vector ngắt chưa điểm vào của 1 chương trình phục vụ ngắt luôn gấp 4 lần số hiệu ngắt. Như vậy để xác định được điểm vào của một chương trình phục vụ ngắt chỉ cần xác định được số hiệu ngắt đó, lấy số hiệu ngắt nhân 4 sẽ xác định được địa chỉ của ô nhớ đầu tiên trong 1 cụm 4 ô nhớ có nội dung chính là điểm vào của chương trình phục vụ ngắt tương ứng với số hiệu ngắt đó.

Cách xác định số hiệu ngắt với ngắt mềm: Số hiệu ngắt nằm ngay trong lệnh ngắt; ví dụ lệnh int 21h; int 10h; int 17h thì 21h, 10h và 17h chính là số hiệu ngắt.

Cách xác định số hiệu ngắt với ngắt cứng: Quy định số hiệu ngắt cho các đầu vào yêu cầu ngắt IRQ bằng cách lập trình cho thanh ghi ICW2 của vi mạch điều khiển ngắt 8259.

Ví dụ: Xác định điểm vào của chương trình phục vụ ngắt của ngắt có số hiệu ngắt là 17h.

Địa chỉ của ô nhớ đầu tiên trong cụm 4 ô nhớ chứa điểm vào của chương trình phục vụ ngắt của ngắt có số hiệu 17h là: $17h * 4 = 0005C H$.

Địa chỉ 2 ô nhớ có nội dung là địa chỉ độ lệch là: $17h * 4 + 0 = 0005C H$

$$17h * 4 + 1 = 0005D H$$

Địa chỉ 2 ô nhớ có nội dung là địa chỉ đoạn là: $17h * 4 + 2 = 0005E H$

$$17h * 4 + 3 = 0005F H$$

Để xem nội dung của các ô nhớ trên có thể dùng lệnh D trong debug

```
Administrator: Command Prompt - debug
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Case>debug
-d 0000:0050 110
0000:0050  EE 09 0E 02 0D 02 E1 02-00 0A 0E 02 8B 05 0E 02  .......
```

Nội dung của ô nhớ 0005C H là: 8B H

Nội dung của ô nhớ 0005D H là: 05 H

Nội dung của ô nhớ 0005E H là: 0E H

Nội dung của ô nhớ 0005F H là: 02 H

Với biểu diễn số trong máy tính sử dụng BVXL của Intel (little endian). Ta có

Giá trị của địa chỉ độ lệch là: 058B H

Giá trị của địa chỉ đoạn là: 020E H

Vậy địa chỉ vật lý của điểm vào (ô nhớ đầu tiên) của chương trình phục vụ ngắt có số hiệu ngắt là 17h là:

CS →	0	2	0	E	0
IP →		0	5	8	B
	0	2	6	6	B

Khi đó HĐH sẽ thực hiện nạp cho thanh ghi con trả lệnh IP = độ lệch = 058B

nạp cho thanh ghi đoạn mã lệnh CS = đoạn = 020E

CPU sẽ đọc lệnh có địa chỉ được ghi trong CS:IP để thực hiện. Đó chính là điểm vào của chương trình phục vụ ngắt của ngắt có số hiệu là 17h.

Cấu trúc ngăn xếp

Ngăn xếp là một vùng nhớ nằm trong bộ nhớ quy ước được HĐH dùng để cất giữ trạng thái của CPU khi thực hiện các thao tác ngắt và các lệnh cất giữ Push và hồi phục Pop trong tập lệnh của BVXL.

Khi có một yêu cầu ngắt được gửi đến chân INT của BVXL, BVXL thực hiện cắt giữ trạng thái của nó tại thời điểm đó vào ngăn xếp để tiến hành quá trình phục vụ ngắt, sau khi hoàn thành quá trình phục vụ ngắt, BVXL lại thực hiện khôi phục trạng thái tại thời điểm xảy ra ngắt và tiếp tục thực hiện tiến trình.

Trong quá trình cắt giữ và hồi phục trạng thái của BVXL, vị trí của ô nhớ mà tại đó sẽ thực hiện quá trình cắt giữ hay hồi phục được gọi là đinh ngăn xếp – Stack point (SP) và nó luôn thay đổi. Đinh ngăn xếp là địa chỉ của ô nhớ có địa chỉ SS:SP.

SS là giá trị của thanh ghi đoạn ngắn xếp,

SP là giá trị của thanh ghi con trỏ ngắn xếp.

Trạng thái của BVXL tại thời điểm xảy ra ngắt là:

Giá trị của thanh ghi cờ (RF) có độ dài 16 bits.

Địa chỉ của ô nhớ chứa lệnh sẽ được thực hiện tại thời điểm xảy ra ngắt, địa chỉ đó chính là giá trị của cặp thanh ghi CS:IP.

Như vậy trạng thái của BVXL tại thời điểm xảy ra ngắt là 3 thanh ghi 16 bits: RF, IP, CS và sẽ cần 6 ô nhớ trong ngăn để cắt giữ trạng thái này.

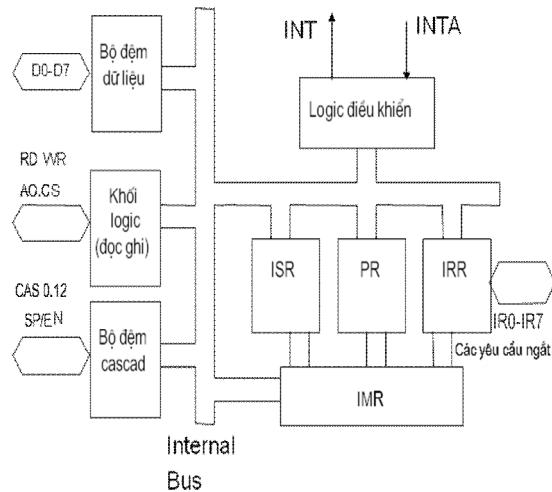
Quá trình cắt giữ và hồi phục trạng thái BVXL qua ngăn xếp:

Quá trình cắt giữ	Địa chỉ ô nhớ	Ngắn xếp	Địa chỉ ô nhớ	Quá trình khôi phục
	FFFF			
Cắt RF vào 2 ô nhớ FFFE và FFFD	Khởi tạo SP = FFFE	Thanh ghi RF	SP = SP+2 = FFFE	Chuyển nội dung 2 ô nhớ FFFD và FFFE vào RF
	FFFD		FFFD	
Cắt IP vào 2 ô nhớ FFFC và FFFB	SP = SP-2 = FFFC	Thanh ghi IP	SP = SP+2 = FFFC	Chuyển nội dung 2 ô nhớ FFFB và FFBC vào IP
	FFFB		FFFB	
Cắt CS vào 2 ô nhớ FFFA và FFF9	SP=SP-2 =FFFA	Thanh ghi CS	SP = SP+2 = FFFA	Chuyển nội dung 2 ô nhớ FFF9 và FFFA vào CS
	FFF9		FFF9	
	SP=SP-2 =FFF8			

Quá trình hồi phục được tiến hành ngược lại với quá trình cắt giữ, Thanh ghi RF được cắt vào đầu tiên khi cắt giữ được lấy ra sau cùng khi khôi phục. Quá trình này được gọi là LIFO (last in – first out).

3.3.4 Vi mạch điều khiển ngắt 8259

Là vi mạch lập trình hỗ trợ ngắt cứng, mở rộng ngắt cho máy tính và sắp xếp thứ tự ưu tiên các yêu cầu ngắt.



Nối ghép với Bus hệ thống: 8 chân số liệu; 1 chân địa chỉ A₀, các chân điều khiển, 1 chân yêu cầu ngắt INT và chân nhận trả lời ngắt INTA.

Nối ghép với ngoại vi: 8 chân yêu cầu ngắt IRQ₀ đến IRQ₇

8259 có 4 từ lệnh khởi tạo là ICW1, ICW2, ICW3 và ICW4.

CS	A0	Tù lệnh khởi tạo
0	0	ICW1
0	1	ICW2, ICW3, ICW4

Tù lệnh khởi tạo 1: ICW1

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	1	LTIM	0	SNGL	IC4

Tù lệnh khởi tạo 2: ICW2

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
T7	T6	T5	T4	T3	T2	T1	T0

Tù lệnh khởi tạo 3 trên 8259 chủ: ICW3

A0	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	S7	S6	S5	S4	S3	S2	S1	S0

S_n = 1 IR lối vào có thiết bị tớ

S_n = 0 IR lối vào không có thiết bị tớ

Tù lệnh khởi tạo 3 trên 8259 tớ: ICW3

A0	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	S7	S6	S5	S4	S3	S2	S1	S0

	0	0	0	Nối đến IR0 của IC chủ
	0	0	1	Nối đến IR1 của IC chủ
	1	1	1	Nối đến IR7 của IC chủ

Tù lệnh khởi tạo 4: ICW4

Tù lệnh điều khiển hoạt động: OCW1, OCW2, OCW3.

\overline{CS}	A0	Tù lệnh điều khiển hoạt động
0	0	OCW2, OCW3.
0	1	OCW1

Tù lệnh điều khiển hoạt động 1: OCW1

A0	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	M7	M6	M5	M4	M3	M2	M1	M0

Tù lệnh điều khiển hoạt động 2: OCW2

Chế độ quay tự động:

A0	IR ₇	IR ₆	IR ₅	IR ₄	IR ₃	IR ₂	IR ₁	IR ₀
0	0	0	1	0	0	1	0	0

Chế độ quay riêng:

Có thể đặt bất kỳ IR nào có giá ưu tiên cao nhất

Tù lệnh điều khiển hoạt động 3: OCW3

Cho phép đọc các thanh ghi IRR và ISR của 8259.

3.3.5. Nối ghép ngắt trong máy tính IBM PC

Máy tính IBM PC XT sử dụng 1 vi mạch 8259 để điều khiển được 8 ngắt cứng, máy tính IBM PC AT sử dụng thêm 1 vi mạch 8259 mắc nối tầng với vi mạch của PC XT để rộng ngắt cứng lên 15.

AEN	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Địa chỉ	
1	0	0	0	1	1	x	x	x	x	0	20h	ICW1
1	0	0	0	1	1	x	x	x	x	1	21h	ICW2, ICW3,ICW4
1	0	0	0	1	1	x	x	x	x	0	20h	OCW2, OCW3
1	0	0	0	1	1	x	x	x	x	1	21h	OCW1

Ở máy tính IBM PC AT, vi mạch 8259 bổ sung làm việc ở chế độ tách rời (chế độ tách rời) nối với vi mạch trước đó làm việc ở chế độ chủ qua chân IRQ2.

IRQ2 của vi mạch chủ nối với chân INT của vi mạch tách rời,

CAS0, CAS1, CAS2 của vi mạch chủ khi đó là các đầu ra được nối đến các chân tương ứng của vi mạch tách rời là các lối vào.

Thiết lập cho 8259 làm việc ở chế độ chủ và tách rời qua cài đặt phần cứng (chân SP/ \overline{EN}) và lập trình qua các thanh ghi ICW1, ICW2, ICW3 và ICW4

Bài tập: chia bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 160-194

Câu hỏi bài 9:

1. Phân biệt quản lý thiết bị ngoại vi bằng phương pháp hỏi vòng và phương pháp ngắt. Trên máy tính IBM PC và tương thích sử dụng phương pháp nào để quản lý ngoại vi.

2. Thế nào là ngắt cứng và ngắt mềm? Trình bày khái niệm về số hiệu ngắt? Tại sao nói việc xác định số hiệu ngắt là bước quan trọng để tiến hành thực hiện chương trình phục vụ ngắt? Trình bày cách xác định số hiệu ngắt của ngắt cứng và ngắt mềm.

3. Trình bày cơ chế phục vụ ngắt trên máy tính:

Máy tính có khả năng đáp ứng bao nhiêu ngắt cứng, bao nhiêu ngắt mềm? Thế nào là chương trình phục vụ ngắt, khi máy tính làm việc, điểm vào của các chương trình này được nằm ở đâu?

4. Trình bày cấu trúc, chức năng của bảng vector ngắt? Phương pháp xác định địa chỉ vào của một chương trình phục vụ ngắt khi có một ngắt xảy ra? Xác định điểm vào của chương trình phục vụ ngắt có số hiệu là 28h?

5. Trình bày cấu trúc, chức năng và hoạt động của con trỏ ngăn xếp khi máy tính thực hiện một thao tác ngắt?

6. Trình bày vai trò của vi mạch quản lý ngắt 8259 trên máy tính IBM PC, nêu địa chỉ giải mã của các thanh ghi của vi mạch 8259 trên máy tính IBM PC XT và IBM PC AT (sử dụng cho các máy tính có BXVLT từ 80286 trở đi)?

Bài giảng 10: Truyền thông nối tiếp

Chương 3 mục 3.4;

Tiết thứ: 34-36

Tuần thứ: 12

Mục đích, yêu cầu: Phương pháp truyền thông nối tiếp và lập trình cho truyền thông nối tiếp

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 3 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

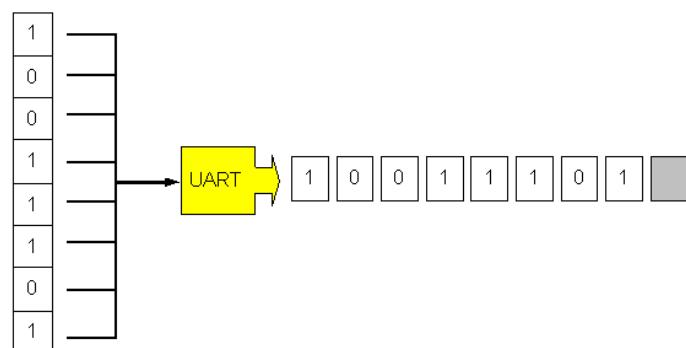
- **Địa điểm:**

Giảng đường do P2 phân công

- **Nội dung chính:**

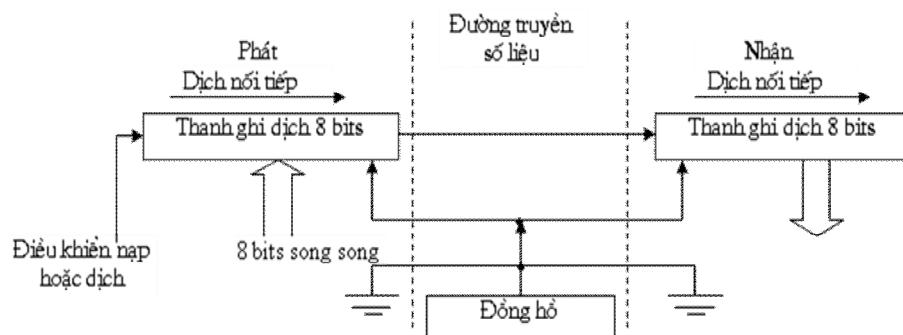
3.5. Truyền thông nối tiếp

Truyền thông nối tiếp là phương pháp truyền số lần lượt theo từng bits.



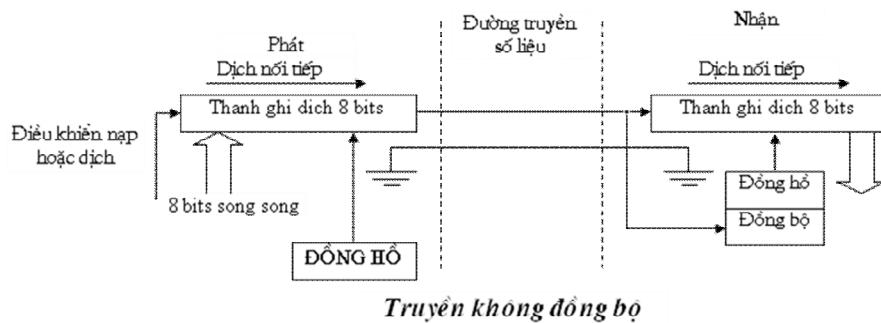
Các thủ tục truyền số liệu:

Truyền đồng bộ:



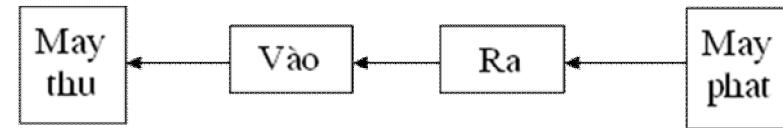
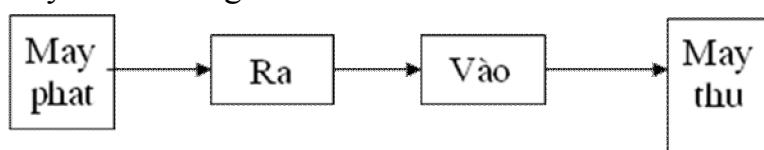
Truyền đồng bộ

Truyền không đồng bộ:

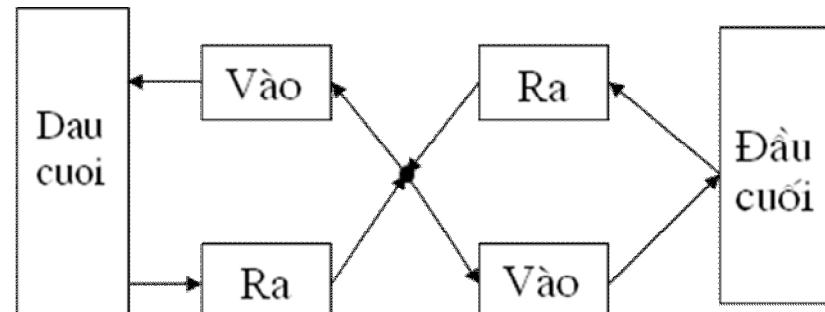


Phương pháp truyền số liệu:

Truyền đơn công

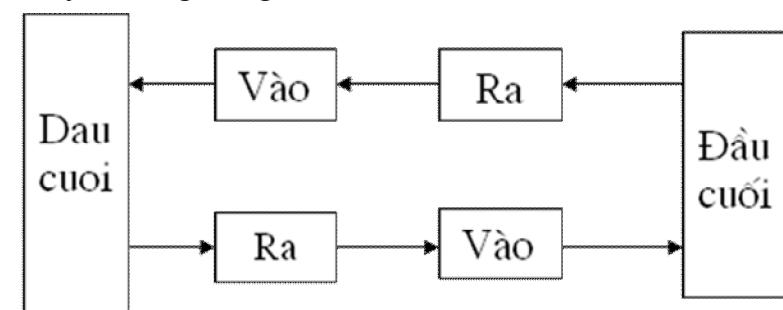


Truyền bán song công



**Truyền
bán
song
công**

Truyền song công



**Truyền
song
công**

Phân biệt loại thiết bị truyền thông:

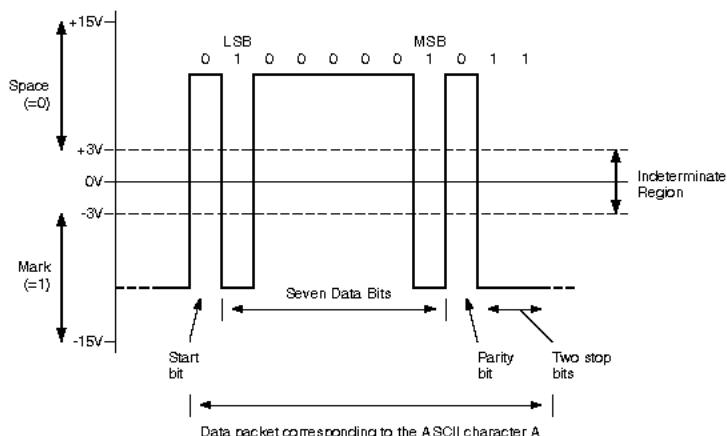
Thiết bị đầu cuối – DTE (Data Terminal Equipment): là đích và nguồn của số liệu (máy tính, máy điện thoại, máy fax.v.v.).

Thiết bị truyền số liệu – DCE (Data Communication Equipment) là thiết bị truyền số liệu như khuyếch đại, điều chế, giải điều chế số liệu (Switch, Repeater, Modem, tổng đài.v.v.)

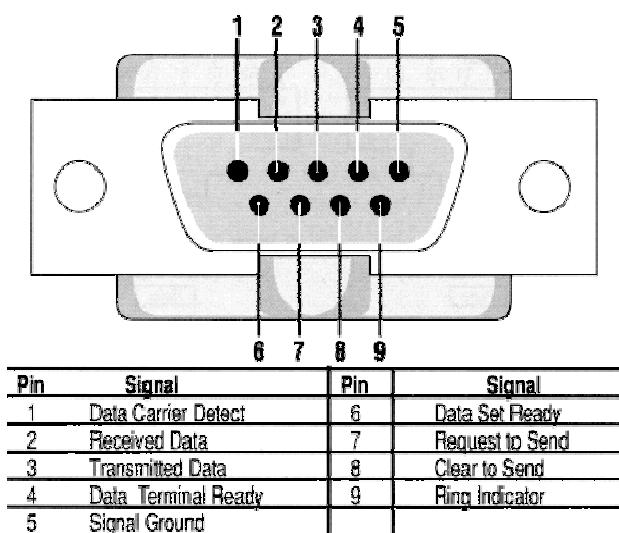
Tốc độ truyền số liệu: Tốc độ truyền số liệu nối tiếp được tính bằng đơn vị bit/second (bps, kbps).

Định khung truyền số liệu: Để đảm bảo việc truyền số liệu chính xác, mỗi lần truyền một block số liệu cần có các thủ tục bắt tay.

Với truyền nối tiếp không đồng bộ thủ tục bắt tay gồm có: 1 bits khởi động có mức logic 0; khôi số liệu cần truyền (5, 6, 7 hoặc 8 bits tùy theo mã truyền thông); 1 bits kiểm tra chẵn, lẻ cũng có mức logic 0, và cuối cùng là 1, hoặc 2 bits kết thúc có mức logic 1.



3.5.1 Chuẩn RS 232



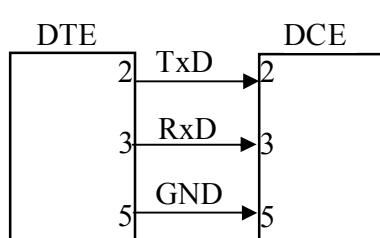
Chuẩn RS 232 là chuẩn truyền số liệu nối tiếp được sử dụng từ những năm 1960. Trên máy tính, cổng truyền số liệu nối tiếp theo chuẩn RS 232 được gọi là cổng COM. 1 máy tính có khả năng quản lý được 4 cổng COM (thứ tự COM1, COM2, COM3, COM4). Khi khởi động máy tính, HĐH sẽ kiểm tra sự có mặt của các cổng COM này và ghi địa chỉ của chúng vào các ô nhớ có địa chỉ từ 00400h đến 00407h. (Sử dụng lệnh D trong Debug để xem trực tiếp).

Các tín hiệu trên đảm bảo đảm bảo cho RS 232 truyền số liệu nối tiếp giữa:

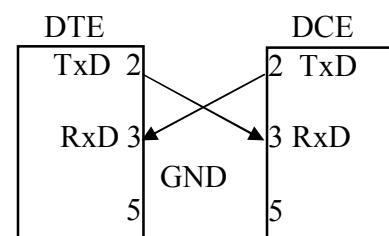
DTE với DTE – Nối trực tiếp 2 máy tính với nhau,

TDE với DCE – Nối máy tính với modem,

DCE với DCE – Nối hai modem với nhau.

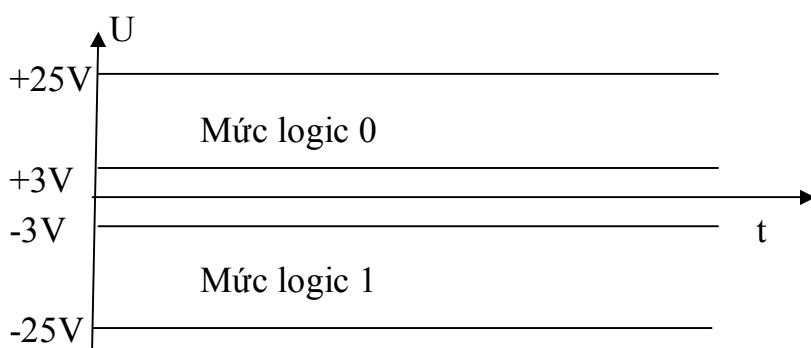


Nối ghép giữa DTE và DCE



Nối ghép giữa DTE và DCE

Chuẩn RS 232 quy định:



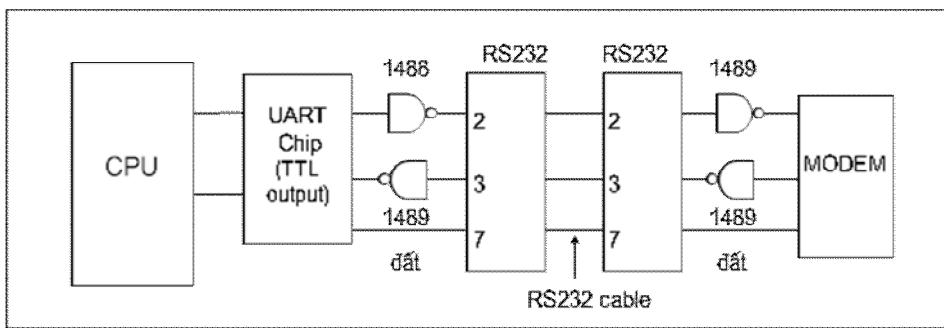
Mức logic 0 Nằm trong dải từ +3V đến + 25V

Mức logic 1 nằm trong dải từ - 3V đến - 25 V

Mức từ - 3v đến + 3 V là mức không xác định.

Máy tính IBM PC sử dụng mức logic 1 là -12V và mức logic 0 là +12V.

Trên máy tính sử dụng mức TTL do đó trên mạch UART phải có mạch chuyển đổi điện áp cho phù hợp giữa mức TTL và mức RS 232.



Hình 9.4. Sử dụng chip MC1488 và MC1489 nối UART với RS232

3.5.2 Lập trình RS 232 bằng ngắt và vi mạch 8250

Vi mạch lập trình nối tiếp 8250 là vi mạch có chức năng nhận và gửi số liệu nối tiếp trên các máy tính họ 80x86.

Kết nối với bus số liệu: Các chân số liệu D₇ đến D₀ được nối từ bus số liệu đến các chân của vi mạch cho phép lập trình để thiết lập chế độ làm việc cho vi mạch và các thao tác nhận và truyền số liệu.

Kết nối với bus địa chỉ: có 3 tín hiệu địa chỉ là A2, A1, A0 kết hợp với tín hiệu giải mã CS0, CS1 và $\overline{CS2}$ cho phép chọn 2³ thanh ghi bên trong vi mạch

TT	A2	A1	A0	Tên thanh ghi
1	0	0	0	Thanh ghi đệm số liệu thu và lưu số liệu phát
2	0	0	1	Thanh ghi cho phép ngắt
3	0	1	0	Thanh ghi nhận biết ngắt (chỉ đọc)
4	0	1	1	Thanh ghi điều khiển đường truyền
5	1	0	0	Thanh ghi điều khiển modem
6	1	0	1	Thanh ghi trạng thái
7	1	1	0	Thanh ghi trạng thái modem
8	1	1	1	Thanh ghi xóa (reset)

Trên máy tính IBM PC cổng COM1 có địa chỉ giải mã từ 03f8h đến 03ffh.

Khi xét truyền nối tiếp giữa DTE và DTE trên các cổng COM ta sẽ xét các thanh ghi đệm số liệu thu và phát (03f8h), thanh ghi điều khiển đường truyền (03fbh) và thanh ghi trạng thái đường truyền (03fdh).

Xét việc truyền số liệu giữa DTE và DTE.

Khi đó cáp kết nối theo quy định như hình vẽ đã trình bày ở trên. Ta cần phải thiết lập cả ở máy thu và phát những tham số sau:

- Định dạng số liệu truyền,
- Tốc độ truyền.

Để thiết lập định dạng số liệu truyền ta lập trình vào thanh ghi điều khiển đường truyền (03fbh) có các bits được định nghĩa như sau:

D7	D6	D5	D4	D3	D2	D1	D0	
DLAB	Break					0	0	5 bits data
0	0				Stop bit	0	1	6 bits data
						1	0	7 bits data
						1	1	8 bits data
					0			1 bit stop
						1		2 bit stop
		0	0	0				Không kiểm tra chẵn lẻ
		0	0	1				Kiểm tra lẻ
		0	1	1				Kiểm tra chẵn

Để thiết lập tốc độ truyền số liệu ta cần phải đặt bit D7 của thanh ghi điều khiển đường truyền lên 1 và sẽ nạp vào 2 thanh ghi có địa chỉ 03f8h và 03f9h một số được tính như sau:

$$\text{Số chia} = \frac{X_{in}}{\text{Tốc độ} * 16}$$

Để truyền và nhận số liệu cần phải hỏi trạng thái của thanh ghi trạng thái đường truyền, nếu được phép lúc đó ta mới được nhận và truyền số liệu. Thanh ghi trạng thái đường truyền có địa chỉ 03fdh có các bits được định nghĩa như sau:

D7	D6	D5	D4	D3	D2	D1	D0
0	TEMT	THRE	BI	PE	PE	OE	DR

Thanh ghi lưu giữ và thanh ghi dịch phát rỗng

1= _____

Thanh ghi lưu giá trị phát rỗng

1= _____

=1 sẵn sàng nhận số liệu

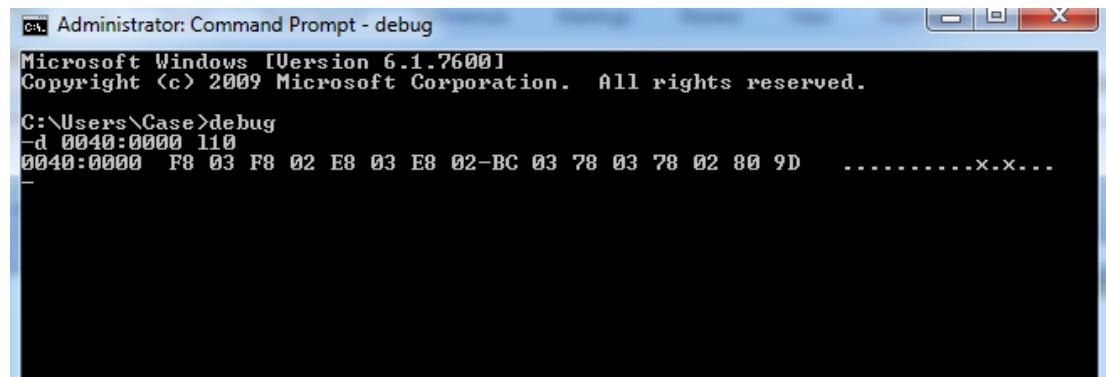
=1 lỗi tràn số liệu

=1 lỗi chẵn lẻ

=1 lỗi định dạng khung

=1 có ngắt

3.5.3 Truyền nối tiếp sử dụng ngắt mềm int 14h



```
Administrator: Command Prompt - debug
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Case>debug
-d 0040:0000 110
0040:0000 F8 03 F8 02 E8 03 E8 02-BC 03 78 03 78 02 80 9D .....x.x...
```

Để giảm nhẹ khối lượng cho người lập trình, HĐH có hỗ trợ vào/ra nối tiếp sử dụng ngắt int 14h với các chức năng được xác định bởi giá trị của thanh ghi AL như sau:

Ngắt Int 14h với Ah= 0: Khởi tạo cổng COM.

DX = 0 - 3: lưu giá trị cổng COM 1 – COM4

D7 D6 D5 D4 D3 D2 D1 D0

x x x	Tốc độ truyền
x x	Kiểm tra chẵn lẻ
x	Số bit Stop
x x	Số bits dữ liệu

D7 D6 D5 : Tốc độ truyền

0 0 1	: 150 bps
0 1 0	: 300 bps
0 1 1	: 600 bps
1 0 0	: 1200 bps
1 0 1	: 2400 bps
1 1 0	: 4800 bps
1 1 1	: 9600 bps

D4 D3 : Kiểm tra chẵn lẻ

0 0	: No Parity
0 1	: Odd Parity
1 0	: No Parity
1 1	: Even Parity

D2 : Số bit Stop

0	: Một bit Stop
1	: Hai bits Stop
D1 D0	: Số bits dữ liệu
1 0	: 7 bits dữ liệu
1 1	: 8 bits dữ liệu

Ví dụ: Khởi tạo cổng COM1: tốc độ 2400 bps, no parity, eight bit data, and two stop bits-

```

    mov ah, 0           ;Initialize opcode
    mov al, 10100111b   ;Parameter data.
    mov dx, 0           ;COM1: port.
    int 14h

```

Ngắt Int 14h với Ah = 1: Truyền dữ liệu qua cổng COM

DX = 0 - 3: lưu giá trị cổng COM 1 – COM4

AL: Chứa Nội dung cần truyền

Ví dụ: Truyền ký tự “a” ra cổng COM1

```

    mov dx, 0           ;Select COM1:
    mov al, 'a'         ;Character to transmit
    mov ah, 1           ;Transmit opcode
    int 14h
    test ah, 80h        ;Check for error
    jnz SerialError

```

Ngắt Int 14h với Ah = 2: Nhận dữ liệu qua cổng COM

DX = 0 - 3: lưu giá trị cổng COM 1 – COM4

AL: Chứa Nội dung nhận được

Ví dụ: Đọc 1 ký tự từ cổng COM1

```

    mov dx, 0           ;Select COM1:
    mov ah, 2           ;Receive opcode
    int 14h
    test ah, 80h        ;Check for error
    jnz SerialError

```

Ngắt Int 14h với AL = 3: Trả về trạng thái cổng COM

AX	Ý nghĩa các bit	AX	Ý nghĩa các bit
D15	Time out error	D7	Receive line signal detect
D14	Transmitter shift register empty	D6	Ring indicator
D13	Transmitter holding register empty	D5	Data set ready (DSR)
D12	Break detection error	D4	Clear to send (CTS)
D11	Framing error	D3	Delta receive line signal detect
D10	Parity error	D2	Trailing edge ring detector
D9	Overrun error	D1	Delta data set ready
D8	Data available	D0	Delta clear to send

3.5.4 Chuẩn truyền thông nối tiếp USB

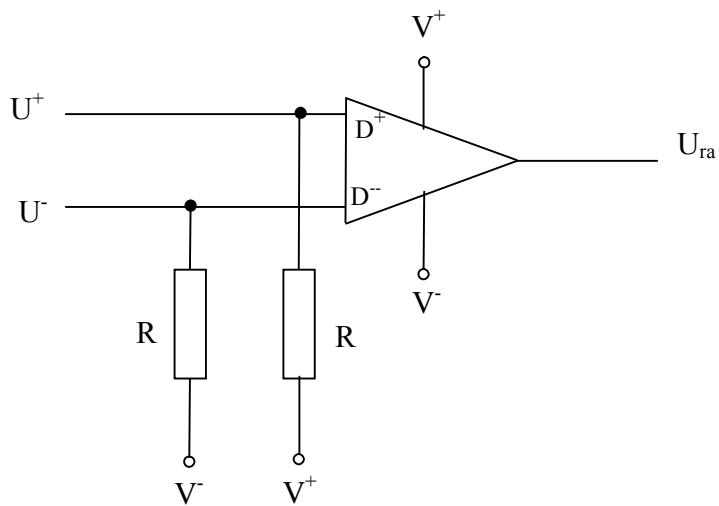
Trên máy tính cho phép có tối đa 4 cổng COM và 4 cổng LPT như vậy chỉ có thể nối tối đa 8 ngoại vi đến máy tính.

Mặt khác tốc độ truyền của các cổng COM và LPT rất hạn chế, không đảm bảo được khi máy tính kết nối với các thiết bị xử lý ảnh như camera, scanner tốc độ cao.v.v.

Xuất phát từ thực tế trên, để theo kịp được công nghệ của Macintosh (giao diện 1394) Intel đã đề xuất 1 giao diện tuần tự đa năng – USB cho phép nối được nhiều ngoại vi và có tốc độ truyền lớn.

Giao diện USB sử dụng giao thức IP (Internet Protocol) nên có thể quản lý đến 127 ngoại vi nhờ sử dụng hub kết nối.

Về mặt truyền số liệu, USB sử dụng phương pháp truyền vi sai (dây truyền tín hiệu có 2 dây D⁺ và D⁻ kết hợp với bộ khuỷch đại vi sai đảm bảo truyền số liệu với tốc độ cao.



Tín hiệu sẽ được truyền trên hai dây tín hiệu là U+ và U- được đưa vào hai chân D+ và D- tương ứng của bộ khuếch đại vi sai. Với sơ đồ trên tín hiệu trên đầu ra Ura sẽ là:

$$U_{ra} = \begin{cases} V^+ & \text{if } |D^+| \geq |D^-| \\ V^- & \text{if } |D^+| < |D^-| \end{cases}$$

Việc sử dụng bộ khuếch đại vi sai có ưu điểm với hai tín hiệu U+ và U- được xoắn với nhau cho phép triệt tiêu được nhiễu tác động lên đường truyền. Nếu có một nhiễu loạn U_{nh} tác động lên đường truyền thì tín hiệu đó sẽ tác động đồng thời lên cả hai dây tín hiệu U⁺ và U⁻, khi đó:

Tín hiệu tác động lên đầu vào D⁺ và D⁻: $\begin{cases} D^+ = U^+ \pm U_{nh} \\ D^- = U^+ \mp U_{nh} \end{cases}$

Khi truyền qua khuếch đại vi sai: $D^+ - D^- = (U^+ \pm U_{nh}) - (U^- \mp U_{nh}) = U^+ - U^-$

Rõ ràng tín hiệu nhiễu tác động lên đường truyền đã bị triệt tiêu khi truyền qua bộ khuếch đại vi sai.

USB cho phép máy tính nối được rất nhiều thiết bị, vì vậy với mỗi loại thiết bị khác nhau cần có những phương thức truyền số liệu phù hợp, vì vậy có rất nhiều phương thức truyền số liệu ở giao thức USB

Truyền điều khiển

Truyền ngắt

Truyền đồng bộ cách biệt

Truyền khôi

Khi sử dụng cụ thể ngoại vi nào, chương trình điều khiển sẽ sử dụng giao thức phù hợp với ngoại vi đó.

- Yêu cầu SV chuẩn bị:

Đọc trước TL1 trang 266-301

Câu hỏi bài 10:

- Định nghĩa truyền thông nối tiếp, thế nào là truyền đồng bộ, không đồng bộ.
- Định nghĩa các phương thức truyền đơn công, bán song công và song công. Định nghĩa chuẩn RS 232. Nêu các mức điện áp của chuẩn RS 232. Trên máy tính IBM PC, mức “0” và “1” theo chuẩn RS 232 có các giá trị điện áp là bao nhiêu?
- Định nghĩa DTE và DCE. Cách nối ghép giữa DTE – DTE, DTE với DCE. Vì sao phải có mạch chuyển đổi mức điện áp tương thích với chuẩn RS 232 sang mức điện áp tương thích với chuẩn TTL. Mạch chuyển đổi dùng loại vi mạch nào?

4. Thực hiện nối ghép 2 máy tính với nhau qua cổng COM (RS 232).

Vẽ sơ đồ ghép nối hai máy tính với nhau qua cổng COM?

Thực hiện chương trình truyền và nhận dữ liệu giữa hai máy tính theo các chế độ sau:

a. Định nghĩa 1 máy chỉ phát dữ liệu, 1 máy chỉ nhận dữ liệu.

Chương trình trên Máy truyền dữ liệu:

Nhận phím từ bàn phím và truyền ký tự nhận được sang máy nhận qua cổng RS 232. Khi ấn phím ESC thì truyền nốt ký tự này rồi kết thúc chương trình.

Chương trình trên máy Nhận dữ liệu:

Chờ nhận 1 ký tự gửi đến từ máy phát, hiển thị ký tự nhận được ra màn hình.

Nếu ký tự nhận được là 1bh (Mã của phím ESC) thì kết thúc chương trình

Chương trình chạy trên máy này cũng chủ động kết thúc chương trình nếu ta ấn phím ESC mà không cần chờ nhận được lệnh dừng từ máy phát.

b. Thực hiện truyền dữ liệu giữa 2 máy tính ở chế độ bán song công.

5. So sánh truyền dữ liệu qua các cổng USB với truyền dữ liệu qua cổng COM và LPT ở các khía cạnh sau:

Số lượng ngoại vi tối đa,

Khoảng cách truyền tối đa,

Tốc độ truyền tối đa,

Mức độ chống nhiễu,

Mức độ tự động định dạng cấu hình (plug and Play)

Bài giảng 11: Các thiết bị vào ra điện hình

Chương 3 mục 3.5;

Tiết thứ: 37-42

Tuần thứ: 13, 14

Mục đích, yêu cầu: Giới thiệu về các thiết bị vào ra điện hình trên máy tính IBM PC

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 6 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- **Địa điểm:**

Giảng đường do P2 phân công

- **Nội dung chính:**

3.5 Các thiết bị vào ra điện hình

3.5.1 Truy cập bộ nhớ trực tiếp DMA

Khái niệm DMA

Khi máy tính cần phải chuyển một số lượng lớn số liệu giữa thiết bị ngoại vi và bộ nhớ. Nếu dùng CPU để thực hiện thì đầu tiên CPU nhận thông tin rồi sau đó mới chuyển đến nơi nhận. Quá trình nhận và giải mã lệnh còn cần thêm các thông tin bổ xung do đó công việc này sẽ rất chậm. Vì vậy Intel thiết kế bộ điều khiển và truy cập trực tiếp 8237 với chức năng bỏ qua CPU và truyền số liệu trực tiếp giữa bộ nhớ và ngoại vi, nhờ vậy mà làm cho quá trình sẽ nhanh lên nhiều.

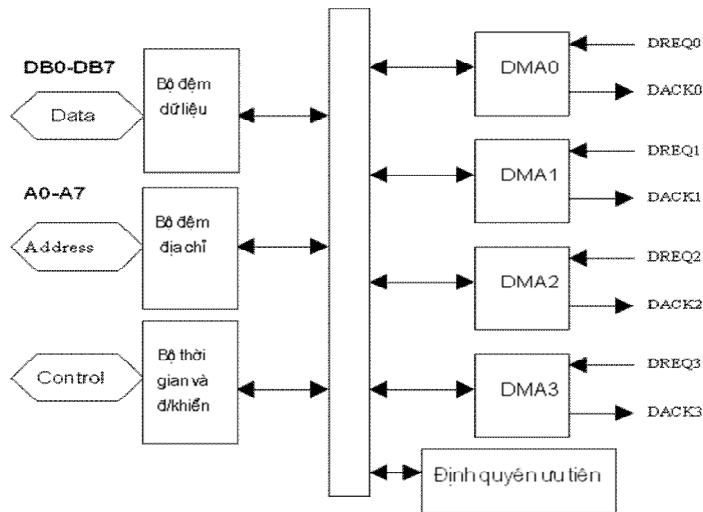
DMA (Direct Memory Access — Truy cập bộ nhớ trực tiếp). Là một dạng của phương pháp vào ra, trong đó dữ liệu được di chuyển trực tiếp giữa bộ nhớ máy tính và mạch ngoài mà không dùng bộ vi xử lý.

Ta có thể so sánh: nếu 8237 có thể truyền byte dữ liệu giữa thiết bị ngoại vi I/O và bộ nhớ trong 4 chu kỳ đồng hồ thì 8088 phải thực hiện trong 39 chu kỳ đồng hồ.

Số chu kỳ đồng hồ

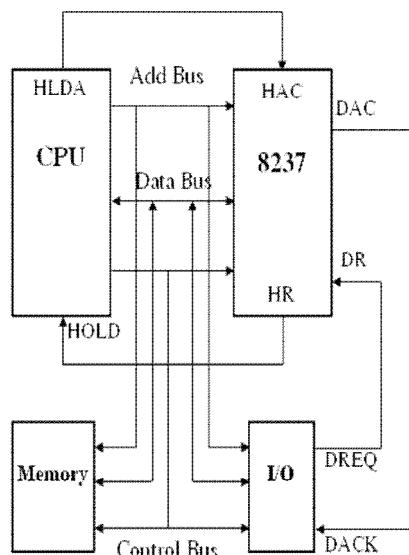
BACK:	MOV	AL,[SI]	10
	OUT	PORT,AL	10
	INC	SI	2
	LOOP	BACK	<u>17</u>
Tổng			39

Vị trí mạch điều khiển DMA – 8237



SƠ ĐỒ KHỐI ĐƠN GIẢN CỦA 8237

Tiến trình DMA



TIẾN TRÌNH DMA

Cấu trúc bên trong của 8237

Bốn kênh DMA được định các mức ưu tiên khác nhau bằng lập trình. Mỗi kênh DMA trống khi sử dụng phải được khởi tạo để xác định địa chỉ và kích thước của khối số liệu. Quá trình khởi tạo là ghi vào từng kênh các nội dung:

1. Thanh ghi địa chỉ cơ sở: ghi địa chỉ đầu của khối dữ liệu cần truyền
 2. Thanh ghi đếm từ cơ sở: Ghi độ dài của khối dữ liệu cần truyền DMA
- Các thanh ghi điều khiển của 8237

A3	A2	A1	A0	Thao tác	\overline{IOR}	\overline{IOW}	Địa chỉ
1	0	0	0	Đọc vào thanh ghi trạng thái	0	1	x8h
1	0	0	0	Ghi vào thanh ghi lệnh	1	0	x8h
1	0	0	1	Ghi vào thanh ghi yêu cầu	1	0	x9h
1	0	1	0	Ghi vào thanh ghi mặt nạ đơn	1	0	x0ah
1	0	1	1	Ghi vào thanh ghi chế độ	1	0	x0bh
1	1	0	0	Xóa con trỏ byte	1	0	x0ch
1	1	0	1	Đọc thanh ghi tạm	0	1	x0dh
1	1	1	0	Xóa thanh ghi mặt nạ	1	0	x0eh
1	1	1	1	Ghi tất cả các bits của thanh ghi mặt nạ	1	0	x0fh

Thanh ghi lệnh - x8h: Là thanh ghi 8 bits được định nghĩa như sau:

D0 =1 Khóa bộ nhớ - Bộ nhớ	D4=0: Mức ưu tiên cố định
D0=0 Mở bộ nhớ - Bộ nhớ	D4=1: Mức ưu tiên quay
D1 = 0: Khóa giữ đ/c kênh 0	D5= 0: Chọn ghi trễ
D1= 1: Mở giữ đ/c kênh 0	D5= 1: Chọn ghi mở rộng
D1 không xác định nếu D0 = 0	D5 không xác định nếu D3 = 1
D2 = 0: Mở bộ điều khiển	D6 = 0: DREQ có mức tích cực cao
D2 = 1: Khóa bộ điều khiển	D6 = 1: DREQ có mức tích cực thấp
D3 = 0: Định thời bình thường	D7 = 0: DACK có mức tích cực thấp
D3 = 1: Định thời nén	D7 = 1: DACK có mức tích cực cao
D3 không xác định nếu D0 = 1	

Thanh ghi trạng thái - x8h Là thanh ghi 8 bits được định nghĩa như sau:

D0 =1 Kênh 0 đạt TC	D4 =1 Kênh 4 đạt TC
D1 =1 Kênh 1 đạt TC	D5 =1 Kênh 5 đạt TC
D2 =1 Kênh 2 đạt TC	D6 =1 Kênh 6 đạt TC
D3 =1 Kênh 3 đạt TC	D7 =1 Kênh 7 đạt TC

Thanh ghi chế độ - x8bh Là thanh ghi 8 bits được định nghĩa như sau:

D1D0 = 00 Chọn kênh 0	D7D6 = 00 Chọn chế độ yêu cầu
D1D0 = 01 Chọn kênh 1	D7D6 = 01 Chọn chế độ đơn
D1D0 = 10 Chọn kênh 2	D7D6 = 10 Chọn chế độ khối
D1D0 = 11 Chọn kênh 3	D7D6 = 11 Chọn chế độ nối tầng
D3D2 = 00 Chuyển kiểm tra	D5 = 0 Khóa khởi tạo tự động
D3D2 = 01 Chuyển ghi	D5 = 1 Mở khởi tạo tự động
D3D2 = 10 Chuyển đọc	
D3D2 = 11 Cấm	D4 = 0 Chọn địa chỉ tăng
D3D2 Không xác định nếu D7D6 =11	D4 = 1 Chọn địa chỉ giảm

Ghép nối 8237 trên máy tính IBM PC XT

Sử dụng 1 vi mạch, thực hiện DMA 8 bits

Kênh DMA0 được sử dụng để làm tươi RAM

Kênh DREQ1 – DREQ3, và DACK0 - DACK3 được bố trí trên rãnh ISA phần 62 Chân

Kênh 1 dự trữ :

Kênh 2 : dùng cho điều khiển đĩa mềm

Kênh 3 : dùng cho điều khiển đĩa cứng

Kích thước tối đa của 1 lần truyền là 64KB

DMA ở máy tính IBM PC AT

Sử dụng thêm 01 vi mạch 8237 nâng số kênh DMA lên 7

Vi mạch lắp thêm sẽ được làm việc ở chế độ master, vi mạch cũ làm việc ở chế độ slave.

3 kênh mở rộng của vi mạch 8237 được đưa ra trên phần 36 chân mở rộng của Bus ISA thực hiện truyền dữ liệu 16 bits nâng kích thước tối đa của 1 lần truyền lên 128 KB.

Địa chỉ giải mã của các kênh DMA trên máy tính bắt đầu từ địa chỉ 80H

3.5.2 Bàn phím

Bàn phím là một thiết bị vào, cho phép người sử dụng có thể vào số liệu hay ra lệnh cho HDH thực hiện 1 thao tác nào đó. Bàn phím là thành phần chính trong giao tiếp trực tiếp người – máy.

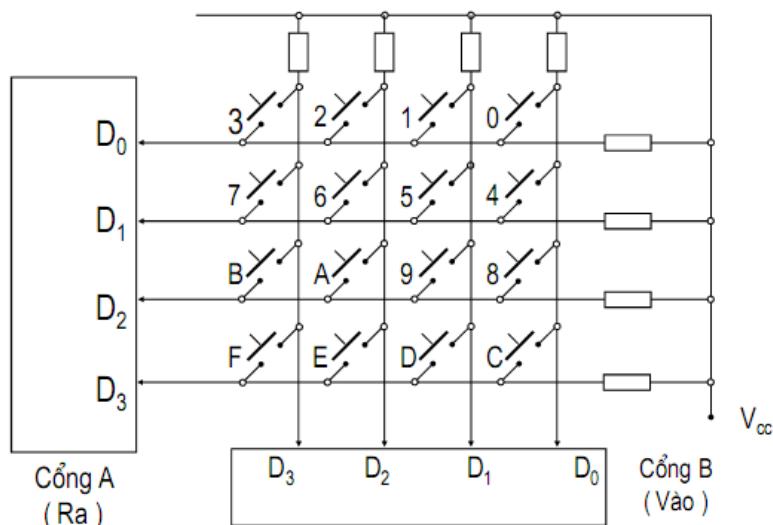
Bàn phím thực chất là một Hệ VXL có nhiệm vụ nhận biết phím được ấn, gán cho phím ấn 1 mã gọi là mã bàn phím và gửi sang máy tính dưới dạng truyền nối tiếp.

3.5.2.1 Cấu tạo

Các phím ấn sử dụng nguyên lý tiếp điểm

Nhận biết phím được ấn: Chia các phím thành các hàng và cột,

Sử dụng phương pháp quét bàn phím: Lần lượt cấp mức logic “0” cho các hàng và lần lượt đọc giá trị trên các cột.



Khi 1 cột có mức “0” thì giao điểm của hàng được cấp “0” và cột đọc được giá trị “0” chính là phím được ấn.

Một phím chỉ có 1 mã bàn phím nhưng có khả năng hiển thị 2 ký tự ASCII, kết hợp mã bàn phím với byte trạng thái bàn phím.

Vai trò của ngắt int 09h và int 16h.

Bàn phím là ngoại vi được quản lý bằng ngắt yêu cầu ngắt bàn phím được nối đến chân IRQ 1 của vi mạch 8259, tương ứng với ngắt có số hiệu 09h. Khi ngắt 09 được kích hoạt chương trình phục vụ ngắt 09 sẽ đọc mã bàn phím trong bộ đệm bàn phím thứ nhất, gán mã ASCII cho phím và chuyển mã ASCII và mã phím vào bộ đệm bàn phím thứ 2. IBM còn dùng thêm 1 ngắt mềm nữa là int 16h để đọc mã ASCII từ bộ đệm bàn phím thứ 2, kết hợp với byte trạng thái bàn phím để hiển thị ký tự ra màn hình.

Khi int 09 không kịp lấy thông tin từ bộ đệm bàn phím thứ nhất sẽ xảy ra hiện tượng tràn bộ đệm bàn phím thứ nhất.

Khi int 16h không kịp đọc mã bàn phím từ bộ đệm bàn phím thứ hai cũng xảy ra tràn bộ đệm bàn phím thứ 2. Để nhận biết tràn bộ đệm bàn phím thứ 2, sử dụng khái niệm con trỏ đầu và con trỏ đuôi.

3.5.2.2 Lập trình bàn phím

Lập trình bàn phím sử dụng ngắn int 16h với:

Ah = 00, 01: Kiểm tra sự có mặt của ký tự trong bộ đệm bàn phím

Ah = 02 Nạp byte trạng thái bàn phím hiện hành vào AL

3.5.3 Ổ đĩa

Đĩa là một thiết bị lưu trữ thông tin được CPU quản lý như một thiết bị ngoại vi tiêu chuẩn.

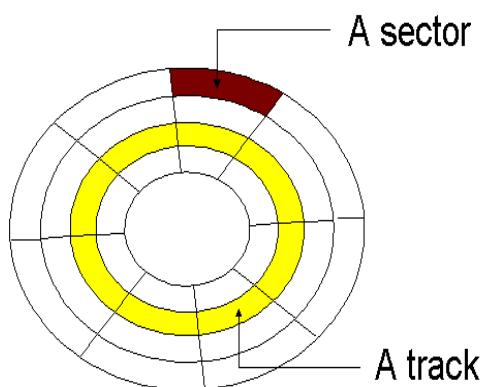
Có các loại đĩa và ổ đĩa: Đĩa mềm, đĩa cứng, đĩa quang.

Khái niệm mặt đĩa mềm, đĩa cứng: Tấm nhựa hình tròn kích thước 3,5' hay 2,5" được phủ một lớp oxit sắt lên bề mặt, quá trình ghi và đọc thông tin trên đĩa là quá trình từ hóa và đọc từ hóa trên bề mặt đĩa.

Khái niệm đĩa quang: Tấm nhựa hình tròn kích thước 3,5' hay 2,5" phủ một lớp đặc biệt, quá trình ghi thông tin lên bề mặt đĩa là dùng ánh sáng laze khắc lên bề mặt đĩa những rãnh nông, sâu khác nhau, quá trình đọc thông tin là đọc độ nông sâu của các rãnh đó.

Khái niệm rãnh – Track: bề mặt đĩa được chia thành nhiều vòng tròn đồng tâm đánh số thứ tự từ ngoài vào trong gọi là track;

Khái niệm cung - sector: Trên mỗi rãnh bao gồm nhiều cung nhỏ được gọi là 1 sector, sector là đơn vị chứa thông tin nhỏ nhất trên đĩa.



Số rãnh trên một măt đĩa, số cung trên một rãnh và dung lượng của một cung phụ thuộc vào từng loại đĩa và dung lượng của đĩa.

Khái niệm liên cung – Clusster: 1 cluster bao gồm nhiều sector, số sector có trong 1 cluster phụ thuộc vào từng loại đĩa. Khi cấp dung lượng đĩa để ghi 1 file thông tin lên đĩa, HĐH sẽ lấy một số nguyên làn số Cluster để lưu trữ 1 file thông tin.

Định dạng đĩa - Format: Trước khi ghi được thông tin lên đĩa HĐH cần định dạng đĩa, quá trình định dạng đĩa sẽ chia không gian đĩa thành 4 vùng:

Vùng khởi động – Boot Record: Cung cấp thông tin về HĐH, về các tham số của đĩa.

Bảng định vị file – FAT: Luôn nằm sau Boot record cùng cấp cho HĐH lô trình tìm kiếm file trên đĩa. Một phần tử của FAT có độ dài là 12 bits, 16 bits hay 32 bits. Tương ứng với nó, bảng FAT sẽ có 2^{12} , 2^{16} hay 2^{32} phần tử và mỗi phần tử sẽ tương ứng với 1 clusster. Nội dung của phần tử trong bảng FAT biểu diễn trạng thái của cluster tương ứng với nó.

Nội dung của phần tử trong FAT	Trạng thái của Cluster tương ứng
0000	Cluster đó rỗng
FFF0 – FFF6	Cluster dự phòng
FFF7	Cluster hỏng
FFF8 – FFFF	Cluster cuối cùng của 1 file
xxxx	Cluster này chứa thông tin, xxxx là cluster tiếp theo chứa thông tin của file

Danh sách các file – Directory: Chứa danh sách các file, kích thước, thuộc tính, phần mở rộng, ngày tháng tạo file.v.v. và số thứ tự cluster đầu tiên của file này.

Vùng chứa thông tin – Data

Phân chia đĩa: Ổ đĩa vật lý và ổ đĩa logic

Chứa bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước TL1 trang 226-256

Câu hỏi bài 11:

1. Khi một phím được ấn sinh ra chỉ 1 mã bàn phím, tại sao lại có thẻ có 2 mã ASCII (1 phím có 2 khả năng – Chữ a: có mã ASCII là 61h và chữ A: có mã ASCII là 41h). máy tính sử dụng biện pháp gì để phân biệt được điều này?
2. Làm rõ vai trò của ngắt int 09h và int 16h khi máy tính thao tác với bàn phím.
3. Giải thích tiến trình DMA và tại sao 1 lần truyền DMA 8 bits chỉ truyền tối đa được 64KB, và 16 bits là 128 KB.
4. Lập trình thực hiện nhận phím từ bàn phím.
Nếu phím nhận được có giá trị từ 0-7 thì thiết lập màn hình ở chế độ tương ứng và hiển thị trên màn hình dòng chữ: "**che do man hinh la:**" (chế độ tương ứng).
 - Nếu phím nhận được là ESC (1bh) thì kết thúc chương trình.
 - Nếu phím nhận được là các phím khác thì mặc định đặt chế độ màn hình là 02Quay lại nhận phím tiếp theo.

Ôn tập cuối môn học

Tiết thứ: 43-45

Tuần thứ: 15

Mục đích, yêu cầu: Tổng kết lại toàn bộ chương trình, giới thiệu đề cương ôn tập

- **Hình thức tổ chức dạy học:**

Lý thuyết, bài tập, tự học, tự nghiên cứu

- **Thời gian:**

Lý thuyết: 3 tiết; bài tập: 0 tiết; tự học, tự nghiên cứu: 6 tiết

- **Địa điểm:**

Giảng đường do P2 phân công

- **Nội dung chính:**

Giới thiệu đề cương ôn tập

Bài tập: chia bài tập

- **Yêu cầu SV chuẩn bị:**

Đọc trước toàn bộ chương trình đã học