

# Input /Output Devices

## Mục tiêu

- Nắm nguyên lý cấu tạo và đặc điểm của thiết bị I/O.
- Nhiệm vụ và yêu cầu của thiết bị I/O.
- Cách giao tiếp giữa CPU và thiết bị I/O.
- Hiểu các bước trong quá trình ngắt quãng.
- Nắm được cơ chế DMA

# Nội dung

- Nguyên lý xuất nhập trong máy tính
- Cách CPU giao tiếp với thiết bị I/O.
- Ngắt quãng
- DMA
- Các thiết bị I/O :  
Hard Disk, Floppy Disk, Printer,  
Keyboard, Mouse

## Thiết bị I/O :

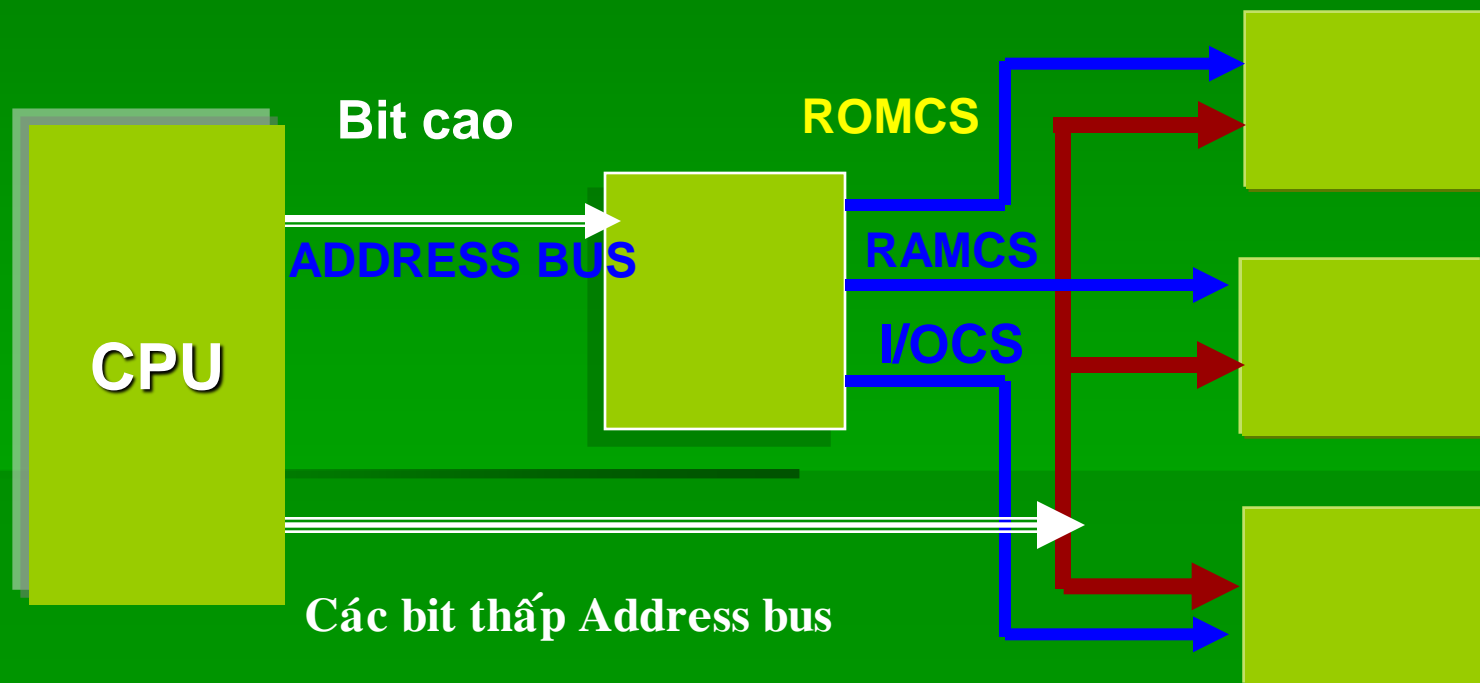
- Thiết bị I/O là 1 thiết bị có khả năng cung cấp dữ liệu khi CPU yêu cầu trong tác vụ đọc và có khả năng ghi dữ liệu vào khi CPU thực thi 1 tác vụ ghi.



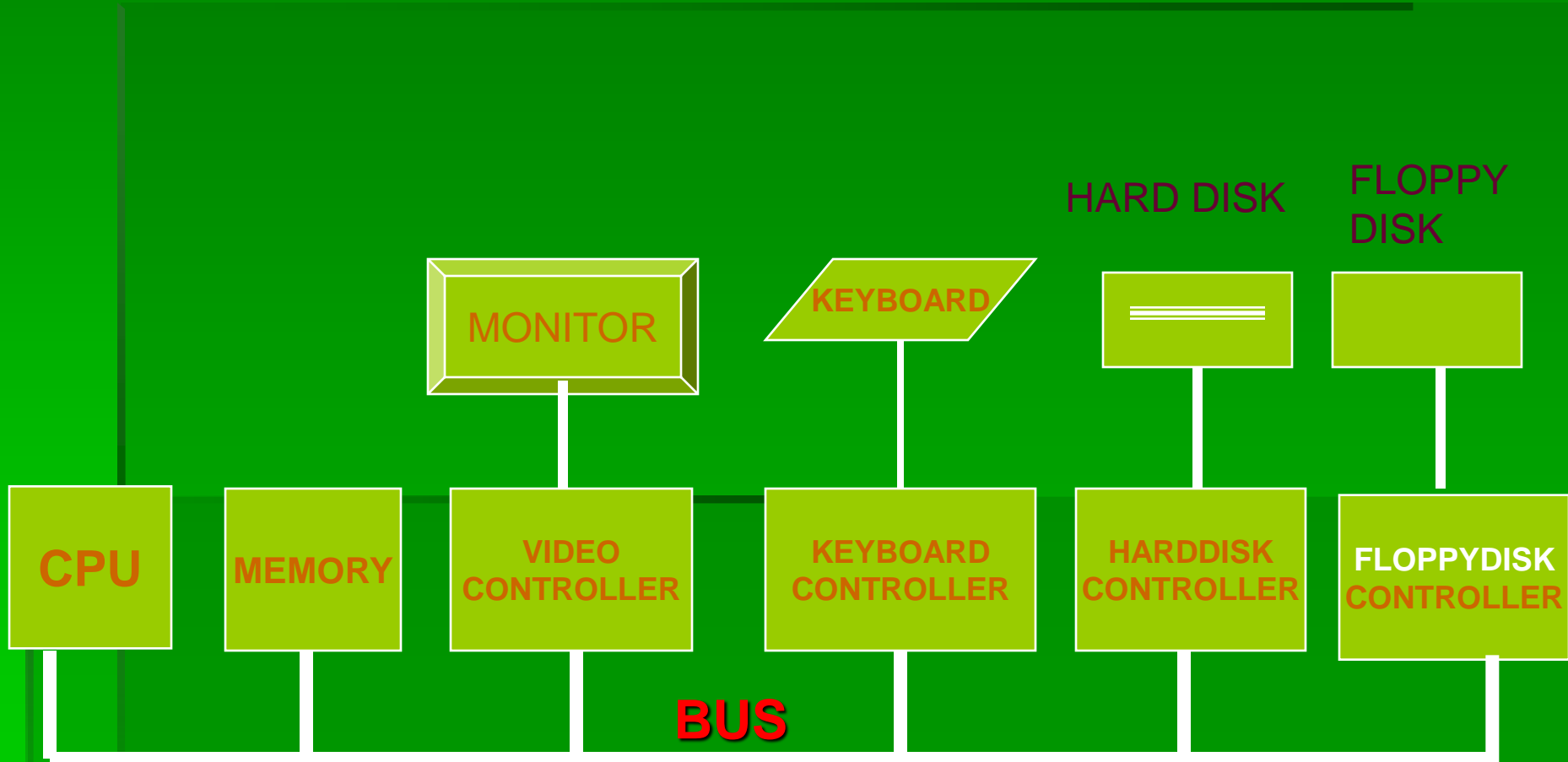
# Làm sao CPU nhận biết một I/O

- Mỗi I/O có 1 địa chỉ riêng gọi là cổng (port). Khi CPU truy xuất I/O, CPU xuất ra 1 địa chỉ.
- Một số bit cao của địa chỉ đi vào bộ giải mã, trên đường ra của bộ giải mã sẽ có tín hiệu Chip select tương ứng với I/O mà CPU muốn truy xuất.
- Các địa chỉ thấp còn lại sẽ đi đến mọi I/O nhưng chỉ có I/O nào có đường Chip Select tích cực mới được truy xuất.

# Sơ đồ giải mã địa chỉ



# Minh họa



# CPU liên lạc với thiết bị I/O

- Thiết bị ngoại vi liên lạc với CPU thông qua các cổng I/O .

Các thiết bị I/O có tốc độ làm việc chậm hơn tốc độ của CPU rất nhiều → để khắc phục nhược điểm này người ta dùng vùng nhớ đệm.

Sự truyền thông tin giữa thiết bị I/O và CPU được thực hiện theo 2 bước :

# CPU liên lạc với thiết bị I/O

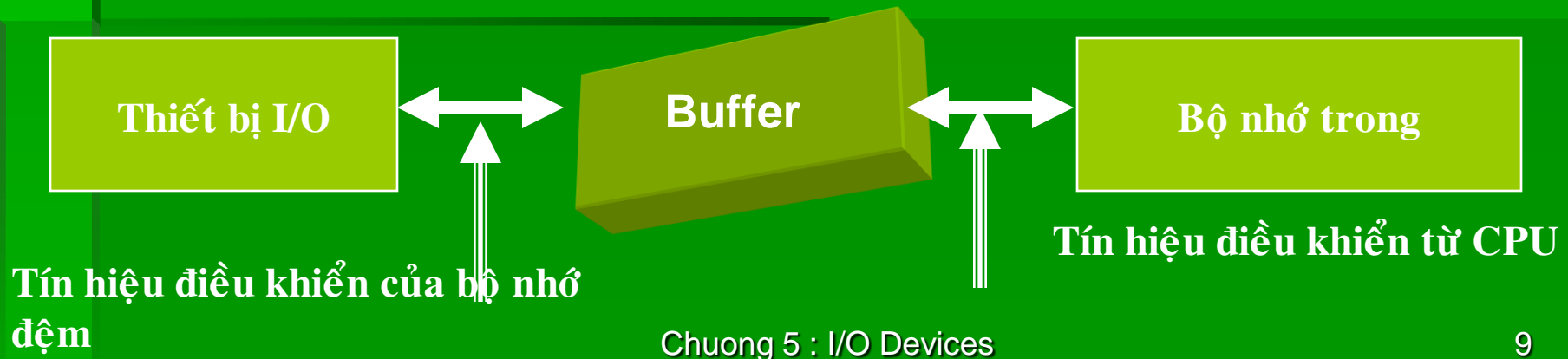
**Bước 1** : truyền thông tin giữa bộ nhớ trong và bộ nhớ đệm.

**Bước 2** : truyền thông tin giữa bộ nhớ đệm và thiết bị I/O.



# CPU liên lạc với thiết bị I/O

- Có thể tổ chức để 1 CPU làm việc đồng thời với nhiều thiết bị ngoại vi bằng cách phân chia thời gian.



# Ngắt quãng (Interrupt)

- Ngắt (Interrupt) là gì ? :  
Ngắt là sự làm ngừng chương trình đang chạy.
- Một interrupt xuất hiện khi 1 chương trình đang thực thi bị ngừng.
- Interrupt được tạo ra bởi nhiều lý do khác nhau

# Ngắt quãng (Interrupt)

- Do user lập trình có lệnh `INT <number>` yêu cầu phục vụ ngắt quãng (như xuất nhập chẳng hạn).
- Do hệ thống gây ra vì 1 lý do nào đó không mong muốn (như lỗi của phép chia 0, phép tính bị tràn số...)
- Do thiết bị I/O gây ra : máy in, bàn phím, ổ đĩa ...

# Software Interrupt

- Ngắt mềm :

Do thi hành lệnh INT trong chương trình.

Xảy ra khi cần 1 chương trình con trong hệ điều hành và thường là chương trình con xuất nhập.

Cú pháp gọi 1 ngắt mềm trong chương trình :

**INT number**

# Software Interrupt

- Một số ngắt mềm thông dụng :

**INT 10h : Video services**

**INT 16h : Keyboard services**

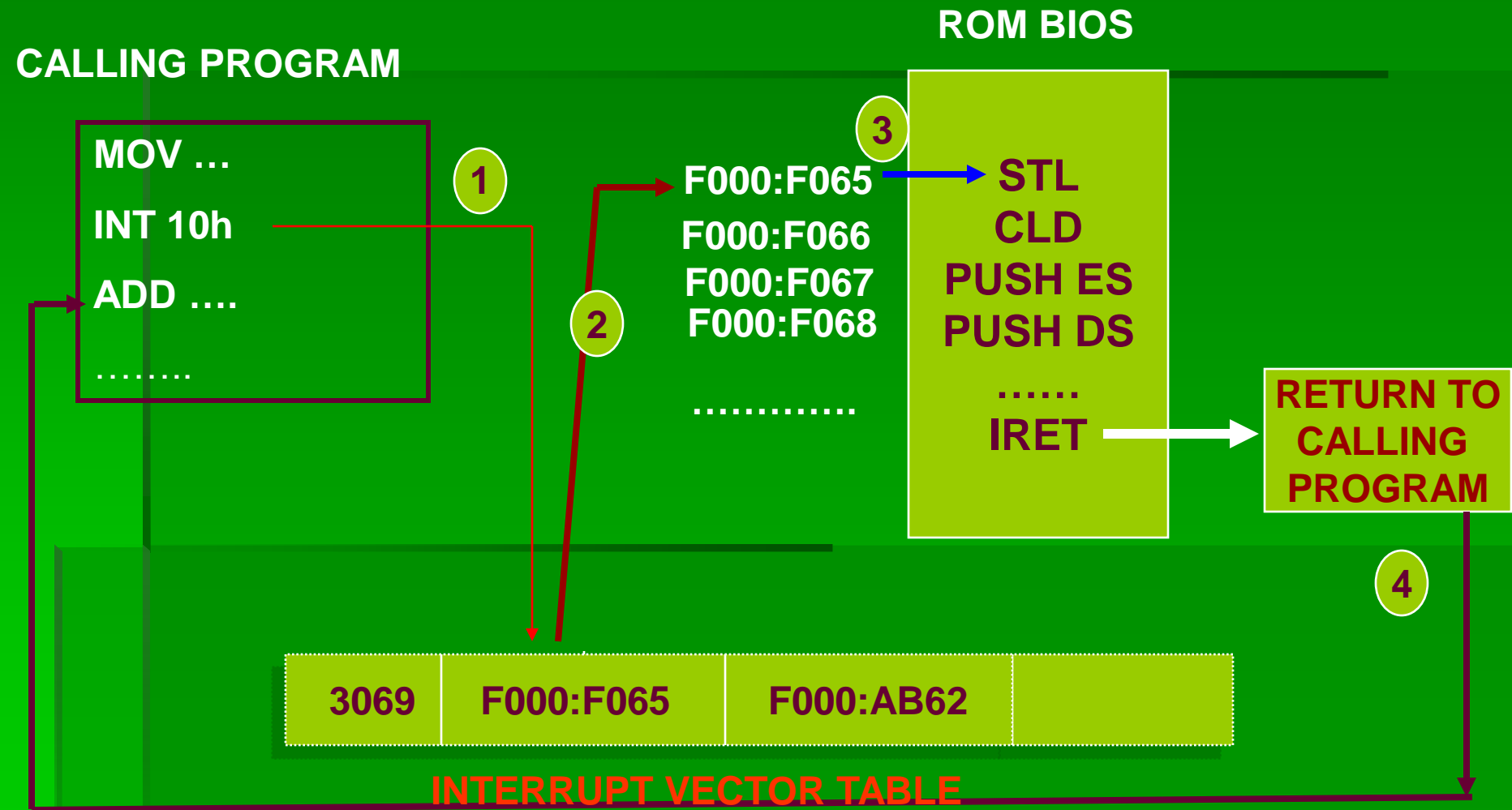
**INT 17h : Printer services**

**INT 1AH : Time of Day**

**INT 1CH : User Time Interrupt**

**INT 21H : Dos Service**

# Thí dụ minh họa gọi ngắt mềm



# Giải thích

1. Con số theo sau INT báo cho CPU biết phải định vị mục nào trong bảng vector ngắt quãng.
2. CPU nhảy đến địa chỉ lưu trong bảng vector ngắt quãng (F000:F065).
3. Một chương trình con (điều khiển ngắt) tại F000:F065 bắt đầu được thi hành và hoàn tất khi gặp lệnh IRET.
4. lệnh IRET giúp CT quay trở lại ngay sau lệnh gọi ngắt và tiếp tục thi hành lệnh này.

# Hệ thống ngắt IBM PC/XT

Ngắt	địa chỉ logic	địa chỉ VL	công dụng
0	00E3:3072	03EA2	lỗi phép chia
1	0600:08ED	068ED	chạy từng lệnh
2	F000:E2C3	FE2C3	ngắt không che NMI
3	0600:08E6	068E6	điểm dừng
4	0700:0147	07147	tràn khi làm việc với số có dấu
5	F000:FF54	FFF54	In màn hình (BIOS)
6, 7	dự trữ		
8 đến F các ngắt của chip 8259			
10	F000:F065	FF065	Vào ra cho Video (BIOS)
11	F000:F84D	FF84D	kiểm tra cấu hình tbị (BIOS)
12	F000: F841	FF841	kiểm tra kích thước bộ nhớ (BIOS)



# Hệ thống ngắt IBM PC/XT

Ngắt	địa chỉ logic	địa chỉ VL	công dụng
13	F000:EC59	FEC59	Vào/ra đĩa (BIOS)
14	F000:E739	FE739	vào/ra RS 232 (BIOS)
15	F000:F859	FF859	vào/ra cassette (BIOS)
16	F000:E82E	FE82E	Vào/ra bàn phím (BIOS)
0700:0147		07147	trần khi làm việc với số có dấu
2	F000:FF54	FFF54	In màn hình (BIOS)
3, 7	dự trữ		
8 đến F	các ngắt của chip 8259		
10	F000:F065	FF065	Vào ra cho Video (BIOS)
11	F000:F84D	FF84D	kiểm tra cấu hình tbị (BIOS)
12	F000: F841	FF841	kiểm tra kích thước bộ nhớ (BIOS)
.....			

# Hệ thống ngắt IBM PC/XT

Ngắt	địa chỉ logic	địa chỉ VL	công dụng
1A	F000:FE6E	FFE6E	thời gian hệ thống (BIOS)
1B	F000:0140	00840	điều khiển Ctrl+Break
.....			
20	PSP:0000	-----	Kết thúc chương trình DOS
21	Có thể đặt lại	-----	gọi chức năng DOS
-----			

F1 – FF không sử dụng

Tùy version DOS, dạng MT một số địa chỉ logic có thể khác nhau

# Một số ví dụ minh họa

Ex : Xem bảng vector ngắt quãng trên MT

- a. Sử dụng DEBUG để hiển thị nội dung của các ô nhớ 0000:002Fh
- b. Tìm CS:IP của lỗi phép chia, NMI và INT 8

Đối với lỗi phép chia INT 0, CS:IP được đặt ở địa chỉ 0,1,2,3.

```
D:\>DEBUG
-D 0000:0000 002F
0000:0000  68 10 A7 00 8B 01 70 00-16 00 96 03 8B 01 70 00
0000:0010  8B 01 70 00 B9 06 0C 02-40 07 0C 02 FF 03 0C 02
0000:0020  46 07 0C 02 0A 04 0C 02-3A 00 96 03 54 00 96 03
-
```

Có thể dữ liệu trên máy PC của bạn khác với dữ liệu trên vì còn phụ thuộc vào version của DOS, ngày tháng của BIOS, việc sử dụng bộ nhớ kép (shadow memory).

Đối với ngắt lỗi phép chia (INT 0), CS:IP được đặt ở địa chỉ 0,1,2,3.

→ CS = 00A7 IP = 1068

Còn INT 8 thì sao ?

**Ex : minh họa INT 0 được gọi và thi hành**

**MOV AL,92**

**SUB CL,CL**

**DIV CL**

**INT 0**

**INT 3**

**Đoạn CT trên sẽ xuất thông báo  
Divide Error**

# Interrupt nội

Ngắt nội : ngắt tự thân do CPU sinh ra còn được gọi là ngắt không che NMI.

Ngắt nội xảy ra khi CPU ở 1 trạng thái không mong muốn như lỗi phép chia 0 (DIV 0), phép chia bị tràn, điện áp nguồn bị giảm thấp ...

# Hardware Interrupt

- Ngắt cứng :

được tạo ra khi thiết bị ngoại vi cần đến CPU.

Ngắt cứng được phát sinh bởi chip 8259 Interrupt Controler, phát tín hiệu cho CPU tạm đình chỉ sự thi hành của CT hiện hành và xử lý ngắt.

Đặc trưng của ngắt cứng là tín hiệu yêu cầu ngắt quăng **INTR**.

Ngắt bàn phím là 1 điển hình ngắt cứng.

Khi cần thiết chương trình có thể cấm ngắt cứng.

# Hardware Interrupt

Đặc trưng của ngắt cứng là tín hiệu yêu cầu ngắt quăng **INTR**.

Ngắt bàn phím là 1 điển hình ngắt cứng.

Khi cần thiết chương trình có thể cấm ngắt cứng bằng lệnh CLI (Clear Interrupt Flag).



# Bảng vector Interrupt

- Một vùng nhớ dài 1024 bytes đặt ở đầu bộ nhớ chính (0h – 400h) , chứa 256 phần tử, mỗi phần tử là 1 bộ 4 bytes đánh số từ 0h-FFh và được gọi là các vector ngắt , tạo thành bảng vector ngắt

Mỗi vector ngắt chứa địa chỉ của 1 chương trình phục vụ ngắt đặt trong bộ nhớ.

Các chương trình phục vụ này liên lạc trực tiếp với các thiết bị I/O thông qua 1 số thanh ghi gọi là cổng (port) vào/ra.

## Bảng Interrupt vector (cont)

- Khi 1 ngắt được yêu cầu, CPU không cần biết địa chỉ của chương trình con phục vụ ngắt này mà chỉ quan tâm đến **số hiệu i** của ngắt và số này chỉ đến phần tử thứ i của bảng interrupt vector .

Ex : Khi ta gõ vào 1 phím, 1 tín hiệu sẽ tạm thời ngắt ngang công việc của CPU. CPU sẽ tìm đến vector ngắt số 9 (của bàn phím). Vector này ở địa chỉ 0:24h.

- CPU sẽ lấy ra địa chỉ của thủ tục chuyên phục vụ bàn phím (có sẵn trong ROM BIOS).
- Thực hiện thủ tục này xong
- Quay trở lại chỗ bị ngắt để tiếp tục thực hiện công việc dở dang



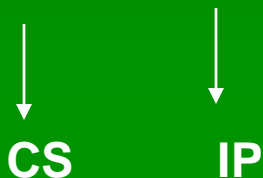
Những chỉ thị ngắt này lấy địa chỉ CTC ở đâu ?

Ex : INT 21h

Để tính địa chỉ của CTC phục vụ ngắt ta :

$21h * 4 = 84h \rightarrow$  cần dùng 2 word (4 bytes) cho mỗi vector ngắt hay địa chỉ CTC.

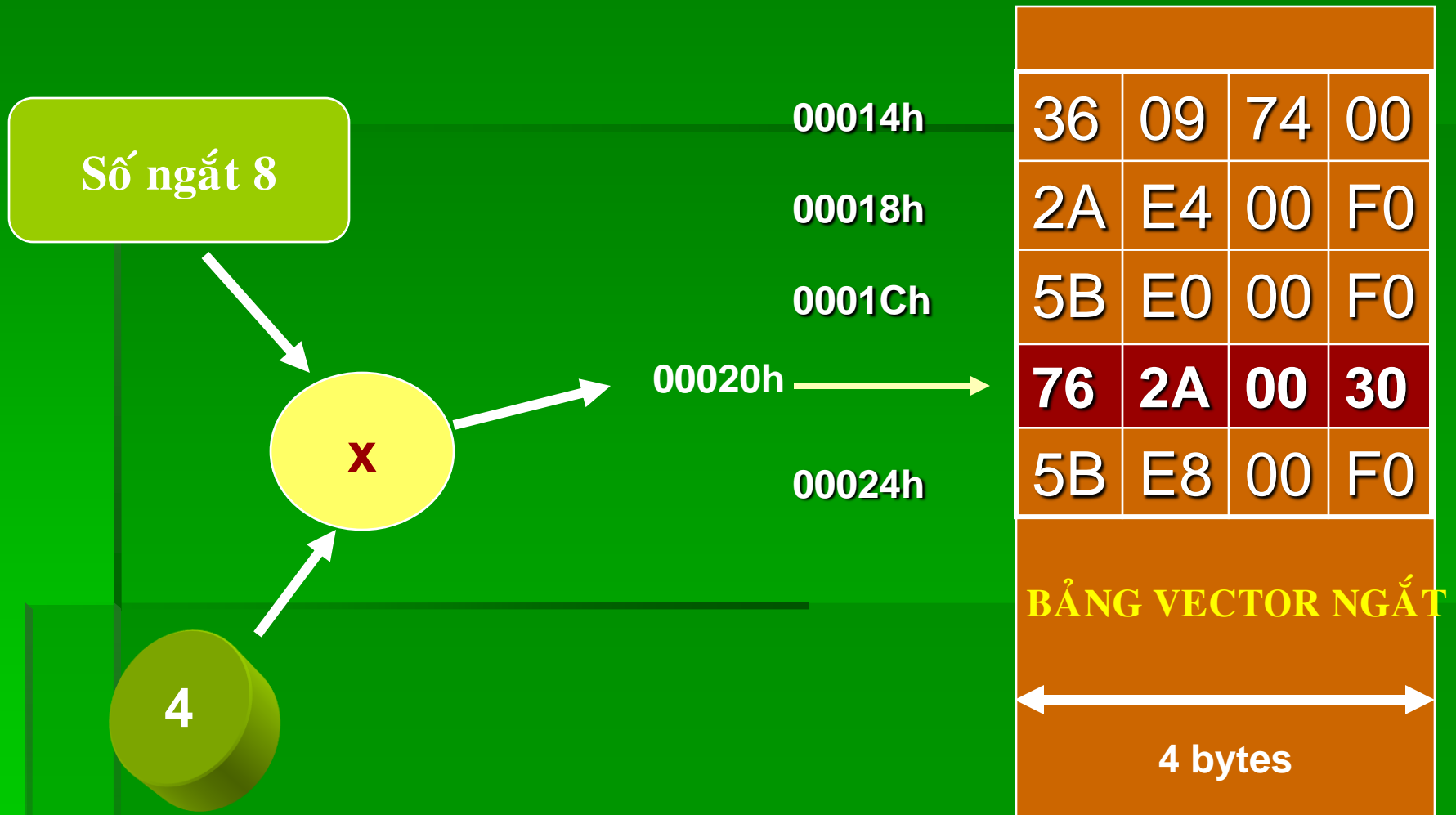
Địa chỉ 00A7:107C



# Các vector ngắt trở đến các thủ tục có sẵn trong **ROM BIOS**

Vector ngắt	Địa chỉ	Chức năng
5	14H	In màn hình ra máy in.
8	20H	Đo thời gian
9	24H	Mã scan từ bàn phím
10	40H	Video display
11	.....	
12	.....	
13	.....	

# Thí dụ minh họa Interrupt



# Các loại cổng vào ra

- Cổng nối tiếp (serial port) : IBM PC cung cấp 2 cổng nối tiếp : COM1 hay AUX và COM2  
2 cổng này ở địa chỉ 400h và 402h trong vùng dữ liệu BIOS.

Cổng nối tiếp dùng cho modem điện thoại, một máy in nối tiếp hay nối trực tiếp với 1 máy tính khác.

# Các loại cổng vào ra

- Cổng song song (parallel port) : IBM PC cho phép sử dụng 3 cổng song song : PRN hay LPT1, LPT2 và LPT3.

Địa chỉ của các cổng này lưu trong bộ nhớ tại 408, 40A, 40C.

Tên cổng	địa chỉ	Nơi chứa địa chỉ
COM1	3F8H	400
COM2	2F8H	402
PRN	3BCH	408
LPT2	378H	40A

# DMA (Direct Memory Access)

## DMA là gì ?

- Kỹ thuật cho phép I/O device hay Bus điều khiển việc truyền dữ liệu vào/ra MT mà không thông qua CPU.  
→ Nhờ thế CPU vẫn điều khiển các quá trình xử lý khác trong quá trình nhập xuất dữ liệu.



# Thiết bị đầu cuối



- **Keyboard : Thiết bị nhập đơn giản.**  
**tập hợp các công tắc bố trí thành 1 ma trận.**  
**Tín hiệu ngõ ra của ma trận công tắc này được đưa vào mạch tạo mã bàn phím.**

**Mỗi tổ hợp phím xác định được ấn xuống mạch sẽ tạo ra 1 con số nhận diện cho phím đó, sau đó con số này sẽ gửi cho CPU.**

## Scan code của Bàn phím

- Chip 8048 xử lý điều khiển bàn phím :

Theo dõi có phím nào được ấn không thì báo cho CPU thông qua ngắt 09h.

**Nếu có phím nào bị ấn quá  $\frac{1}{2}s$  , 8048 sẽ lặp lại phím này sau những khoảng thời gian nhất định (typematic)**

Mỗi lần ấn 1 phím, các mạch điện tử của bàn phím sẽ tạo ra 1 mã dài 1 byte gọi là mã scan , đặc trưng cho vị trí trên bàn phím của phím tác động, giá trị nằm trong khoảng 1 ..83

## Scan code của Bàn phím

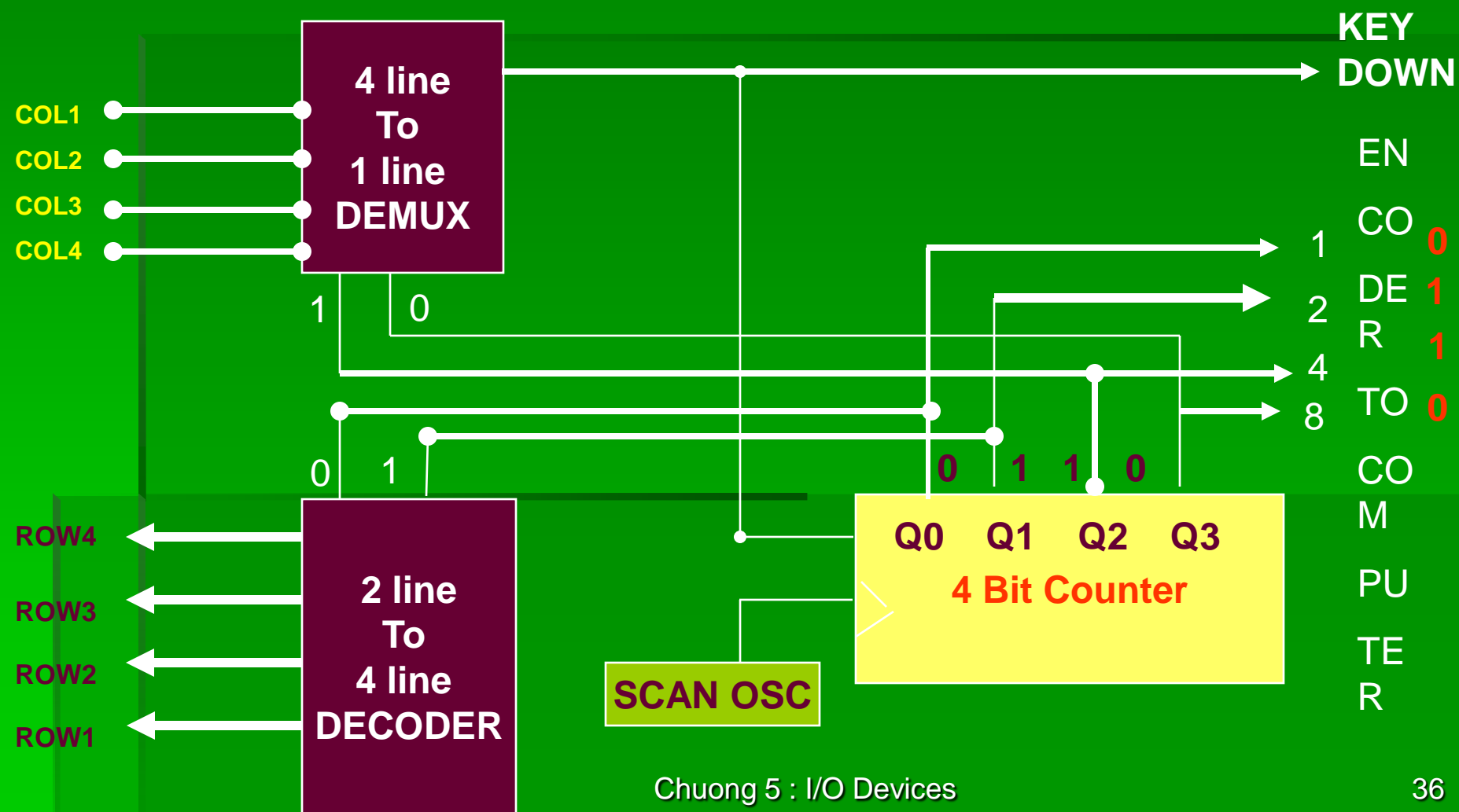
Làm sao MT phân biệt được khi 1 phím được nhấn và khi phím đó được nhả ?

- Khi nhả phím bị ấn, bàn phím tạo ra 1 mã scan khác với mã scan lúc phím bị ấn, có giá trị bằng mã trước cộng thêm 128 (80h) , nghĩa là đổi bit 7 của byte mã scan trước từ 0 → 1

**Ex : khi ta ấn chữ z , scan code là 44**

**Nhả phím này ra bàn phím tạo mã scan 172**

# Bộ mã hoá quét trên ma trận



# HOẠT ĐỘNG NGẮT QUÃNG CỦA IO

Khi 1 IO có yêu cầu giao tiếp với CPU (xuất nhập data), IO này sẽ kích khởi 1 đường tín hiệu IRQ của mình (Interrupt request) để báo là mình cần phục vụ.

Các bước trong tiến trình ngắt quãng :

- IO có yêu cầu CPU phục vụ, sẽ gửi tín hiệu IRQ đến Interrupt controller.
- Nếu có nhiều I/O cùng yêu cầu ngắt , IntController sẽ giải quyết các yêu cầu bằng chế độ ưu tiên.
- IntController phát 1 tín hiệu đến CPU xin ngắt, CPU sẽ hoàn tất lệnh đang thực hiện , cất giá trị của thanh ghi IP và CS vào stack để biết địa chỉ trở về sau khi phục vụ ngắt hoàn tất.

## Liên lạc giữa bàn phím và CPU(cont)

- Cứ mỗi lần có 1 tác động ấn phím
- → mạch bàn phím gây ra ngắt 9
- → gọi 1 chương trình con phục vụ ROM BIOS.

**INT 9 sẽ đọc cổng 60H để biết tác động phím nào đã xảy ra (đọc mã scan tương ứng).**

**INT 9h chuyển mã scan này thành mã dài 2 bytes , byte thấp chứa mã ASCII của phím đó, byte cao chứa mã scan.**

# HOẠT ĐỘNG NGẮT QUÃNG CỦA I/O

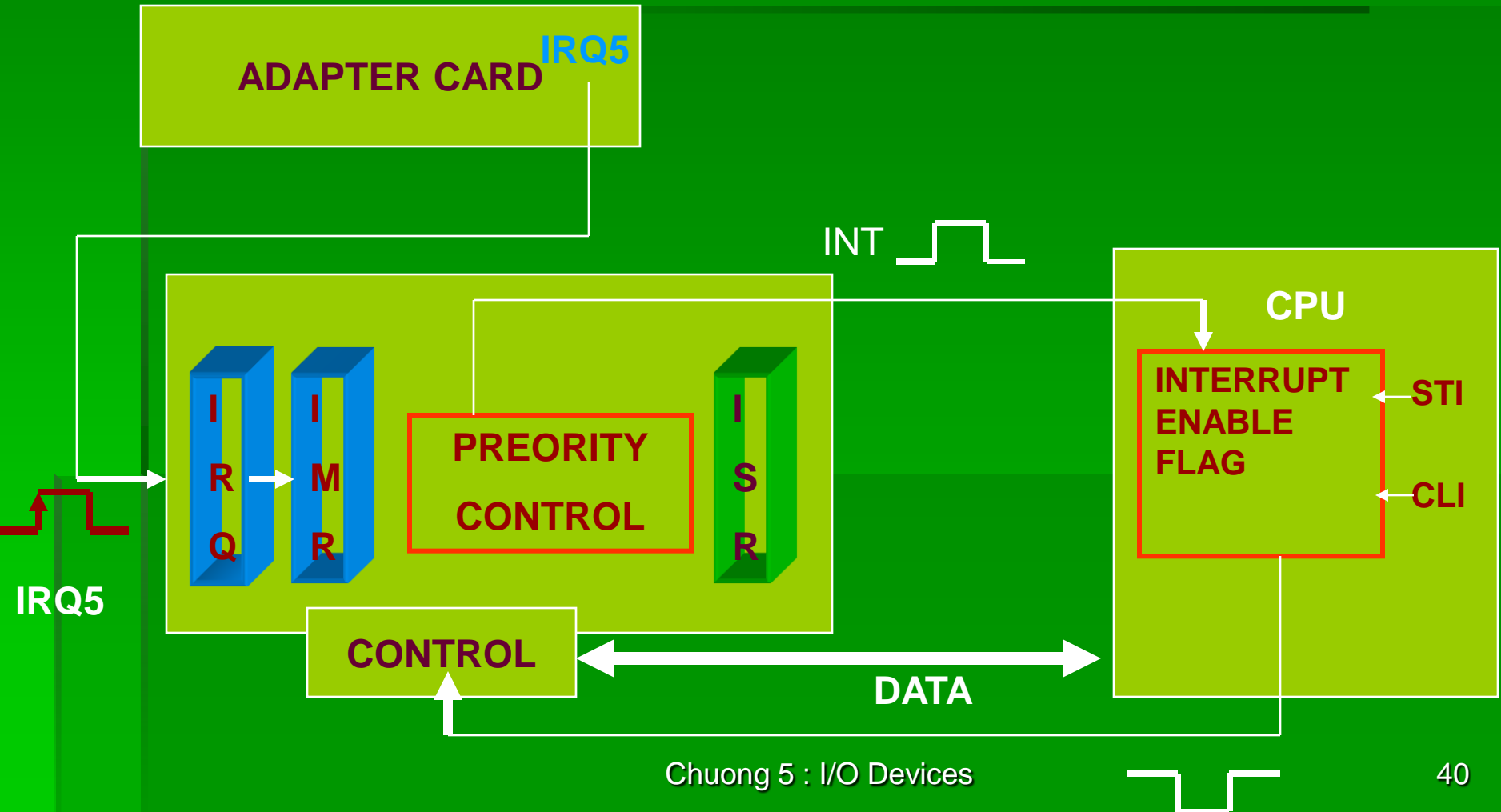
■ CPU yêu cầu mã nhận dạng để biết phục vụ cái gì?

Nhờ mã này CPU vào bảng Interrupt vector để biết địa chỉ bắt đầu của chương trình con phục vụ ngắt nằm đâu trong bộ nhớ.

■ CPU chép địa chỉ bắt đầu của chương trình con phục vụ ngắt vào CS và thực thi mã lệnh của chương trình này.

■ Sau khi thực hiện xong tác vụ của ngắt, lệnh cuối cùng là INTR, CPU sẽ lấy giá trị cũ của CS và IP trong stack ra để tiếp tục thực thi các lệnh còn lại của ứng dụng

# Thí dụ minh họa về interrupt





# Giải thích

- Card IO yêu cầu phục vụ bằng cách đưa đường tín hiệu IRQ trên card lên mức 1.

Đường IRQ nối với bộ điều khiển ngắt, yêu cầu này sẽ làm bật bit 5 của thanh ghi IRR.

Interrupt controller sẽ so sánh IRR với thanh ghi mặt nạ IMR để xem hiện tại có cho phép IRQ5 hay không. Nếu cho phép thì sẽ kiểm tra tiếp.

**Kiểm tra xem có 1 ngắt có ưu tiên cao hơn IRQ5 hay không. Nếu có thì IRQ5 phải chờ cho đến sau khi ngắt ưu tiên thực hiện xong.**

# Giải thích (cont)

- Lệnh EOI sẽ xóa bit 5 trong thanh ghi ISR để IntController có thể tiếp nhận 1 yêu cầu khác.

Chương trình phục vụ ngắt phát lệnh IRET kết thúc. CPU phục hồi giá trị CS và IP từ Stack để có thể tiếp tục thực hiện quá trình trước đó.

# Giải thích (cont)

- Interrupt controller đưa đường tín hiệu INT lên mức 1 để báo cho CPU biết có ngắt quãng.

**CPU nhận tín hiệu INT, sẽ báo lại cho Interrupt controller tín hiệu đồng ý bằng cách đưa tín hiệu INTA về mức 0.**

Sau khi CPU chấp thuận ngắt, Interrupt controller gửi data cho CPU với trị 0Dh (giả sử IRQ5 tương ứng với ngắt 0Dh).

Bật bit 5 của thanh ghi đang phục vụ ISR, xóa bit 5 của IRR vì IRQ5 đã được giải quyết.

# Giải thích (cont)

- CPU tạm dừng quá trình hiện hành, lưu giá trị CS và IP vào stack. Nhảy đến địa chỉ 0000:4\*0Dh, lấy địa chỉ offset của chương trình con phục vụ ngắt 0Dh.

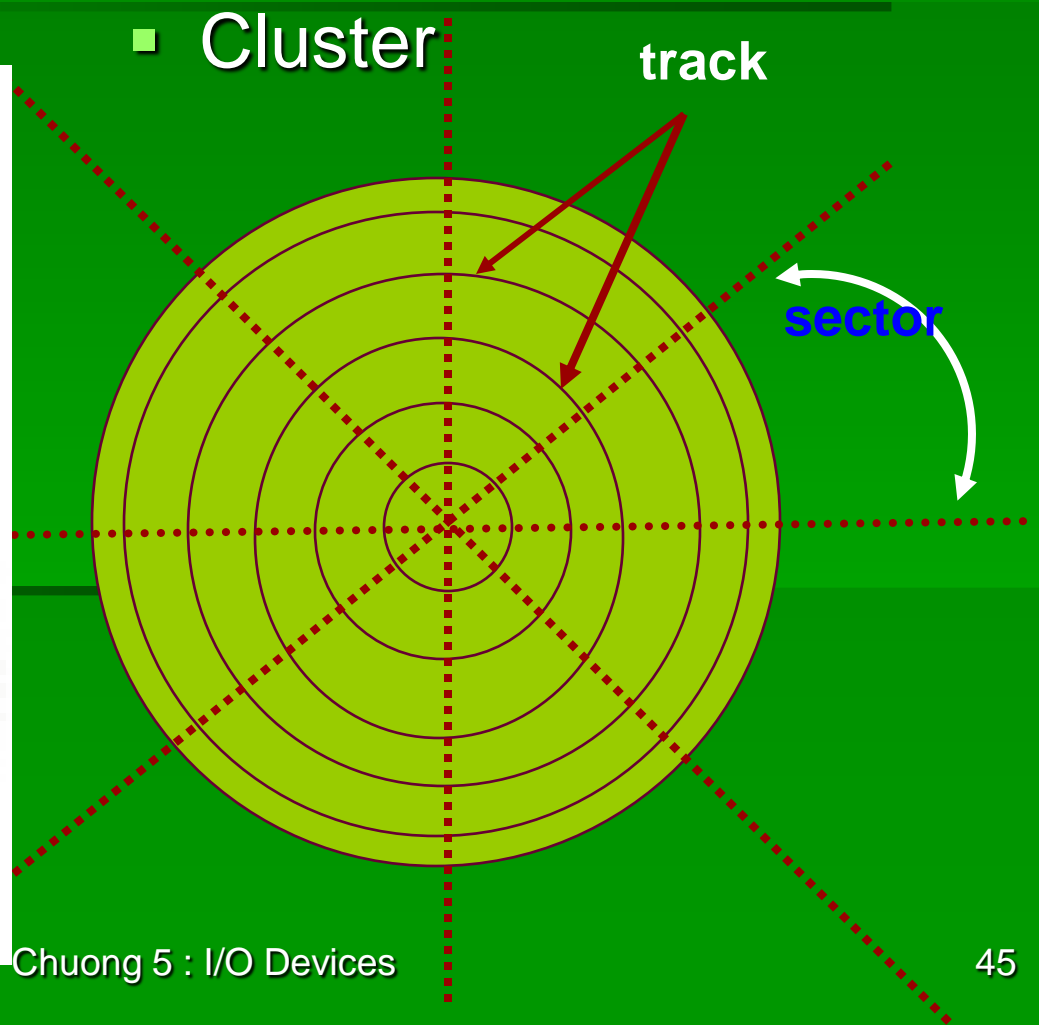
Nhảy đến nơi chứa các lệnh của ngắt này và thực thi các lệnh tương ứng.

Ở cuối chương trình phục vụ ngắt, CPU gửi giá trị báo kết thúc phục vụ ngắt EOI (End of Interrupt =20h) cho IntControler.

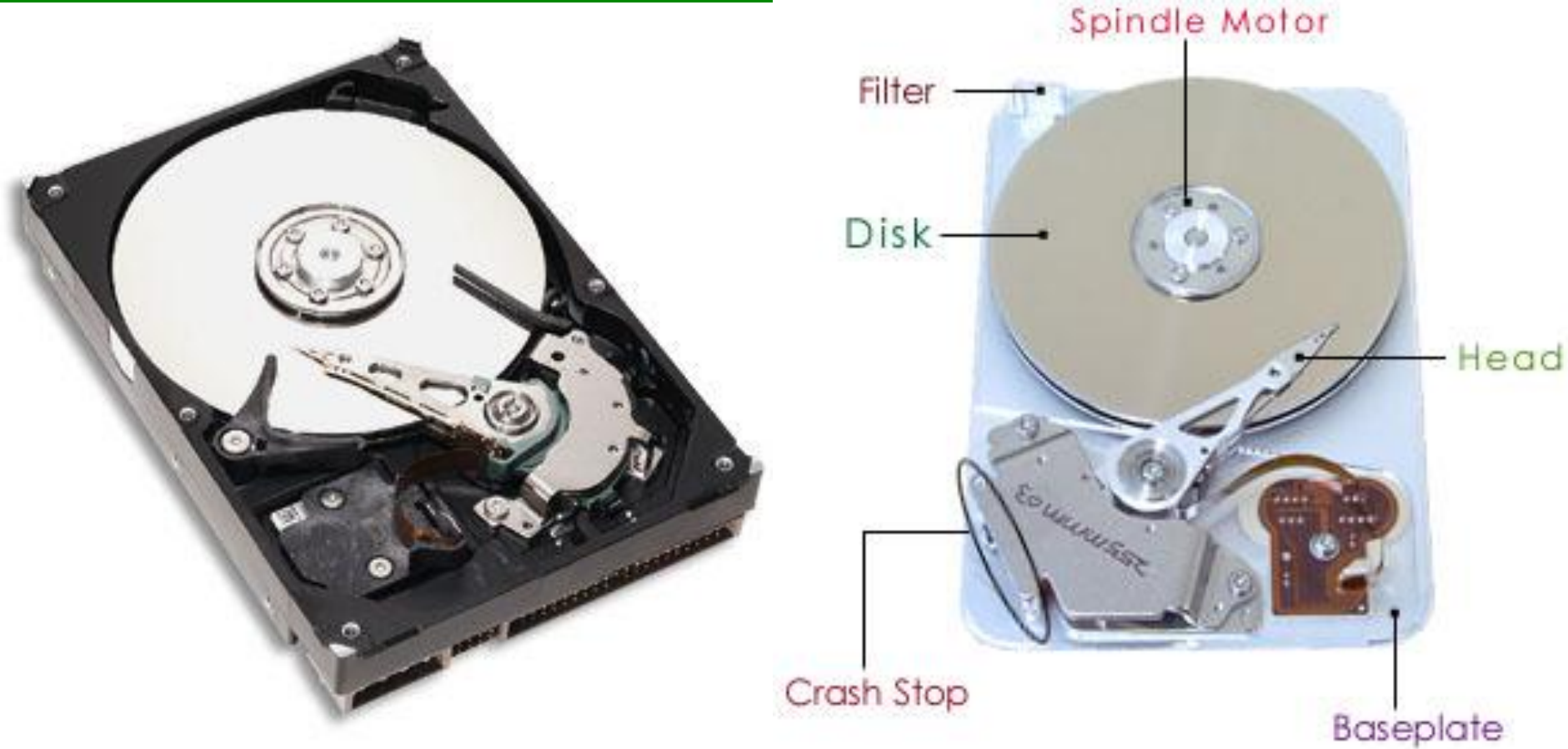
# Hard Disk

Các thuật ngữ

- Track
- Cylinder
- Sector
- Cluster



# Hard Disk



# Hệ thống tập tin của DOS và điều khiển đĩa

- **Bảng FAT : (File Allocation Table)**
- **Nằm ngay Boot Sector (sector 0).**
- **Bảng FAT được tạo ra khi ta partion đĩa cứng**
- **Nội dung Bảng FAT mô tả trạng thái của các cluster còn tốt hay đã hư (vật lý), đã dùng hay chưa dùng...**

# Summary slide



- I/O là gì ?
- Mô tả tiến trình phục vụ ngắt quãng.
- Tính toán vị trí của vector ngắt của interrupt 20h.
- Viết các lệnh sử dụng ngắt 21h , hàm 9 để hiển thị ngày hiện tại.
- Bảng FAT là gì ?



# Summary slide



- Thế nào là ngắt nội. Cho 1 thí dụ minh họa.
- Khi lập trình, ta thường gọi 1 chương trình phục vụ xuất nhập, lúc đó ta sử dụng loại ngắt nào. Cách gọi.
- Làm sao để phân biệt ngắt cứng và ngắt mềm.
- Khi dùng INT 21h để hiển thị 1 ký tự ra màn hình, thanh ghi nào chứa ký tự sẽ hiển thị?.