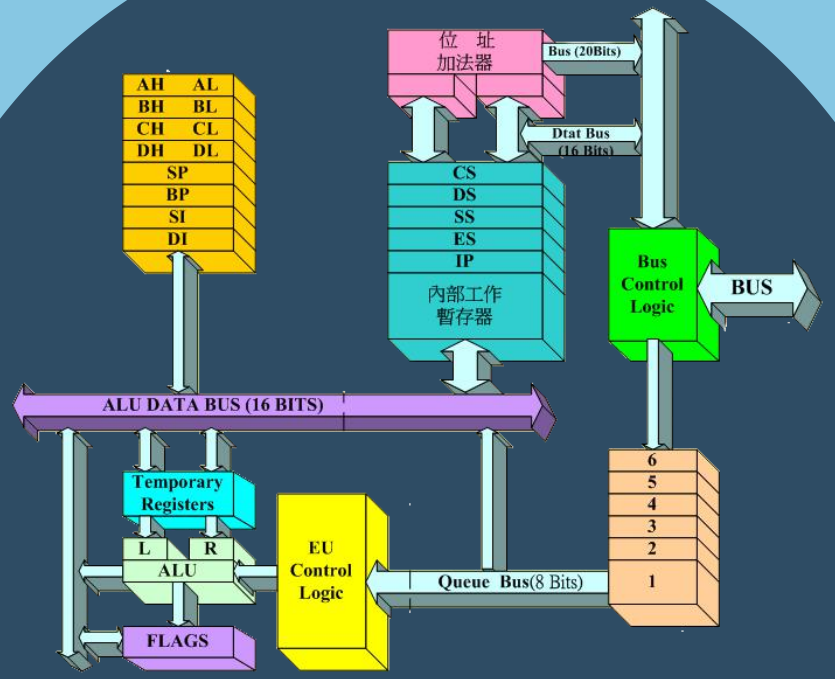
用中断响应外设

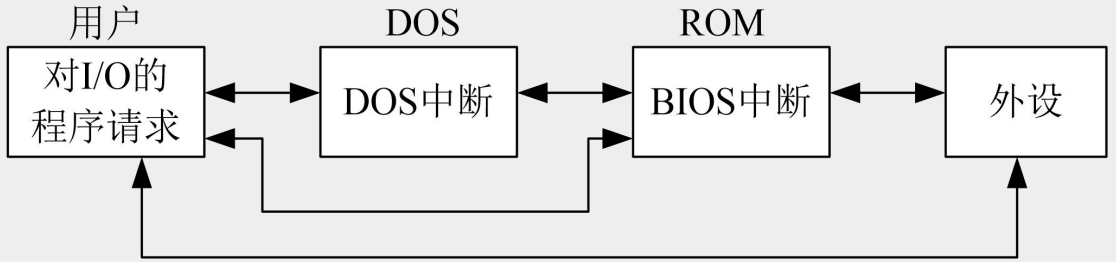
贺利坚　主讲



汇编语言程序设计

Assembly Language

如何操作外部设备?



以典型输入设计——键盘操作为例

硬件中断 int 9h	BIOS中断 int 16h	DOS中断 int 21h
由键盘上按下或松开一个键时，如果中断是允许的，就会产生int 9h中断，并转到BIOS的键盘中断处理程序。	BIOS中断提供基本的键盘操作，功能号(AH)= 00H、10H —从键盘读入字符 01H、11H —读取键盘状态 02H, 12H —读取键盘标志 03H —设置重复率 04H —设置键盘点击 05H —字符及其扫描码进栈 在使用功能键和变换键的程序中很重要。	DOS中断提供丰富、便捷的功能调用 功能号(AH)= 01H —从键盘输入一个字符并回显 06H —读键盘字符 07H —从键盘输入一个字符不回显 08H —从键盘输入一个字符，不回显，检测CTRL-Break 0AH — 输入字符到指定地址的缓冲区 0BH — 读键盘状态 0CH — 清除键盘缓冲区，并调用一种键盘功能

对键盘输入的处理的int 9h中断和int 16h中断

- int 9h将键盘输入存入缓冲或改变状态字
- 键盘输入将引发9 号中断，BIOS 提供了int 9 中断例程。
- int 9中断例程从60h 端口读出扫描码，并将其转化为相应的ASCII 码或状态信息，存储在内存的指定空间（键盘缓冲区或状态字节）中。
- 键盘缓冲区中有16 个字单元，可以存储15个按键的扫描码和对应的入ASCII 码。

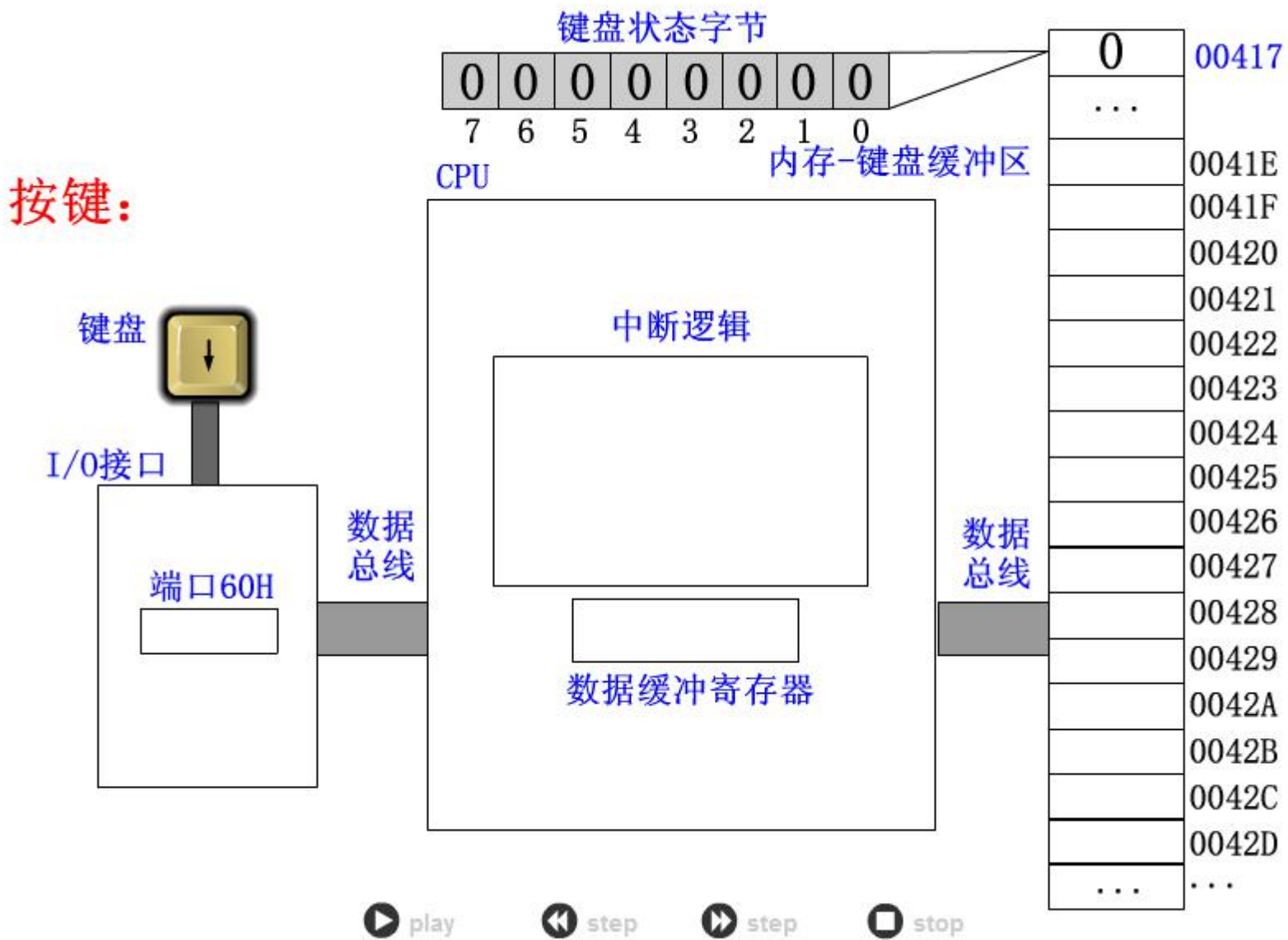
- BIOS提供了int 16h 中断例程供程序员调用，以完成键盘的各种操作。
- 例：当(AH)=0时，读取键盘缓冲区

  功能：从键盘缓冲区中读取一个键盘输入，并且将其从缓冲区中删除。

  ```
  mov ah,0
  int 16h
  ```

  结果：(ah)=扫描码， (al)=ASCII码。

演示：输入A、B、C、D、E、Shift_A、A引发的(int 9)“动作”



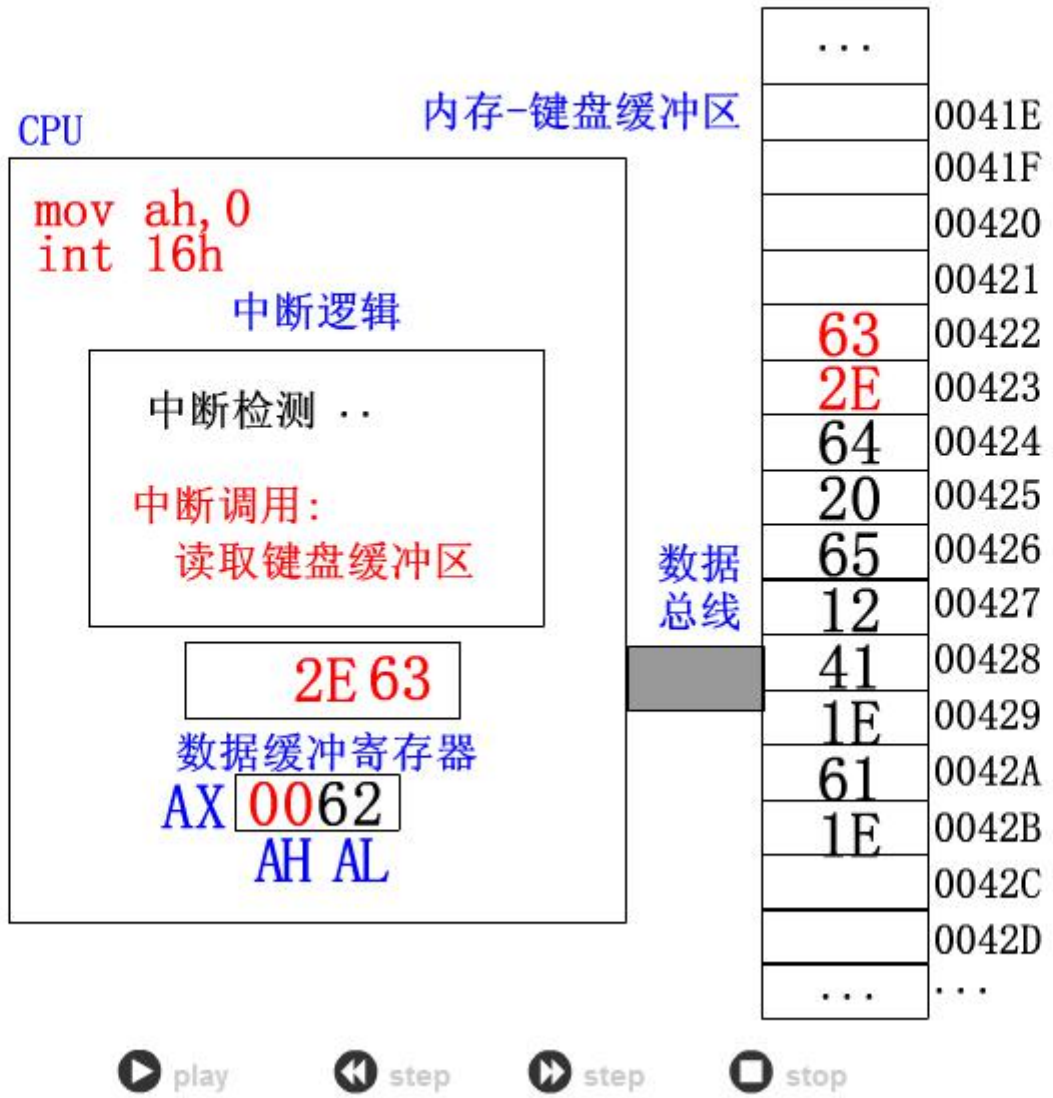
动作	扫描码
安A	1E
安B	30
安C	2E
安D	20
安E	12
安左Shift	2A
安A	1E
公Shift	(变状态字)
安A	1E

演示(续)

- 用int 16h读取出用int 9h存入缓冲区的数据

键盘缓冲区的实现

- 共16字
- 用环形队列
- 可存储15个按键扫描码



调用int 16h 从键盘缓冲区中读取键盘的输入

- 综前所述：int 16h 中断例程 0 号功能的实现过程
  - （1）检测键盘缓冲区中是否有数据；
  - （2）没有则继续做第1 步；
  - （3）读取缓冲区第一个字单元中的键盘输入；
  - （4）将读取的扫描码送入ah，ASCII 码送入al；
  - （5）将已读取的键盘输入从缓冲区中删除。
- 事实
  - B1OS 的int 9 中断例程和int 16h 中断例程是一对相互配合的程序，int 9 中断例程向键盘缓冲区中写入，int 16h 中断例程从缓冲区中读出。
  - 它们写入和读出的时机不同，int 9 中断例程在有键按下的时候向键盘缓冲区中写入数据；而int 16h 中断例程是在应用程序对其进行调用的时候，将数据从键盘缓冲区中读出。
- 应用
  - 我们在编写一般的处理键盘输入的程序的时候，根据需要用不同的方法。

应用示例：更改屏幕颜色

- 要求：接收用户的键盘输入
  - 输入“r”，将屏幕上的字符设置为红色；
  - 输入“g”，将屏幕上的字符设置为绿色；
  - 输入“b”，将屏幕上的字符设置为蓝色。

7	6	5	4	3	2	1	0
BL	R	G	B	I	R	G	B
闪烁		背景		高亮		前景	

```
assume cs:code
code segment
start:
    mov ah,0
    int 16h
    ; 调用中断，等待输入
    ; 识别按键
    mov ah,1
    cmp al,'r'
    je red
    cmp al,'g'
    je green
    cmp al,'b'
    je blue
    jmp short sret
    ; 设置屏幕颜色
red:    shl ah,1
green:  shl ah,1
blue:   mov bx,0b800h
        mov es,bx
        mov bx,1
        mov cx,2000
s:      and byte ptr es:[bx],11111000b
        or es:[bx],ah
        add bx,2
        loop s
sret:   mov ax,4c00h
        int 21h
code ends
end start
```