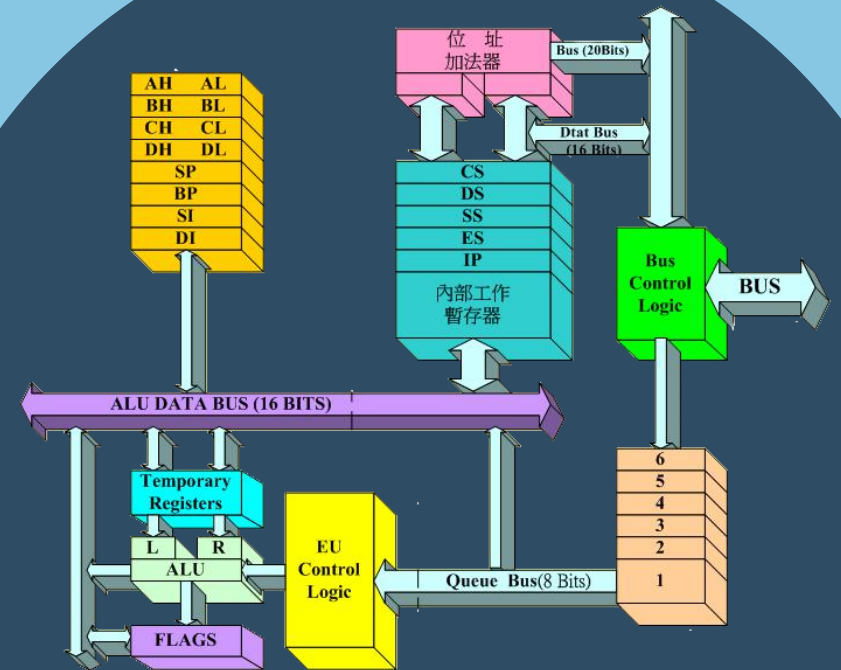


BIOS和DOS中断处理

贺利坚 主讲



汇编语言程序设计
Assembly Language

BIOS——基本输入输出系统

🖥️ BIOS，是在系统板的ROM中存放着一套程序

📄 容量：8KB

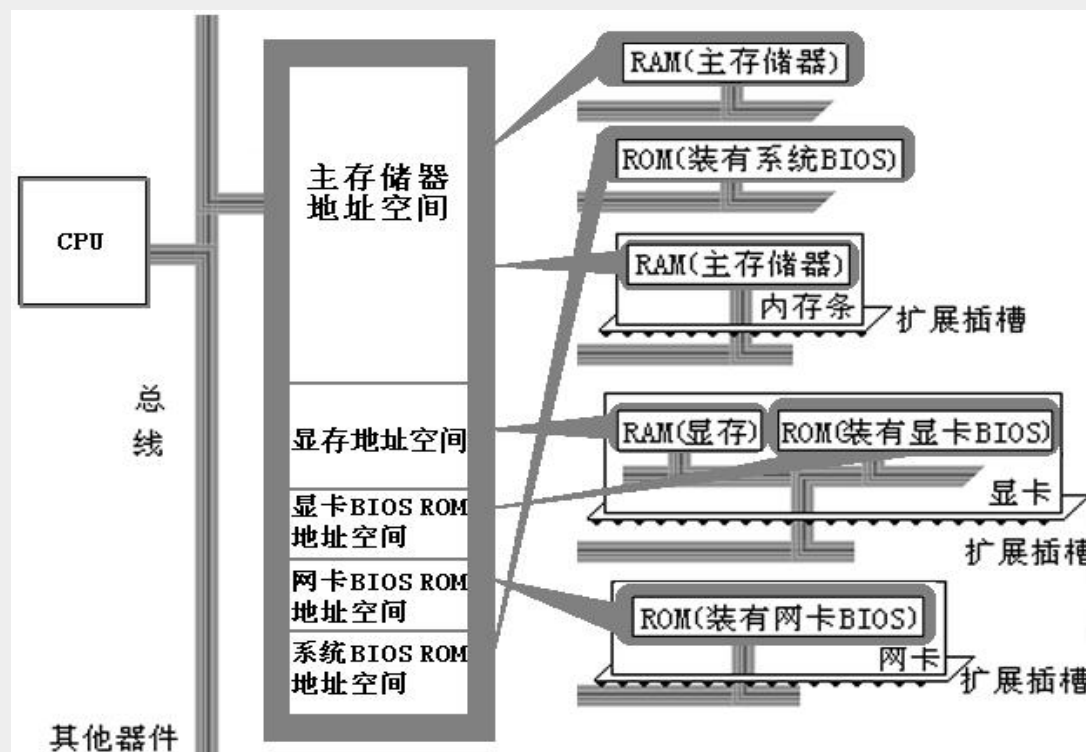
📄 地址：从FE000H开始

🖥️ BIOS中的主要内容

- (1) 硬件系统的检测和初始化程序
- (2) 外部中断和内部中断的中断例程
- (3) 用于对硬件设备进行I/O操作的中断例程
- (4) 其他和硬件系统相关的中断例程

使用BIOS功能调用，程序员不用了解硬件操作细节，直接使用指令设置参数，并中断调用BIOS例程，即可完成相关工作！

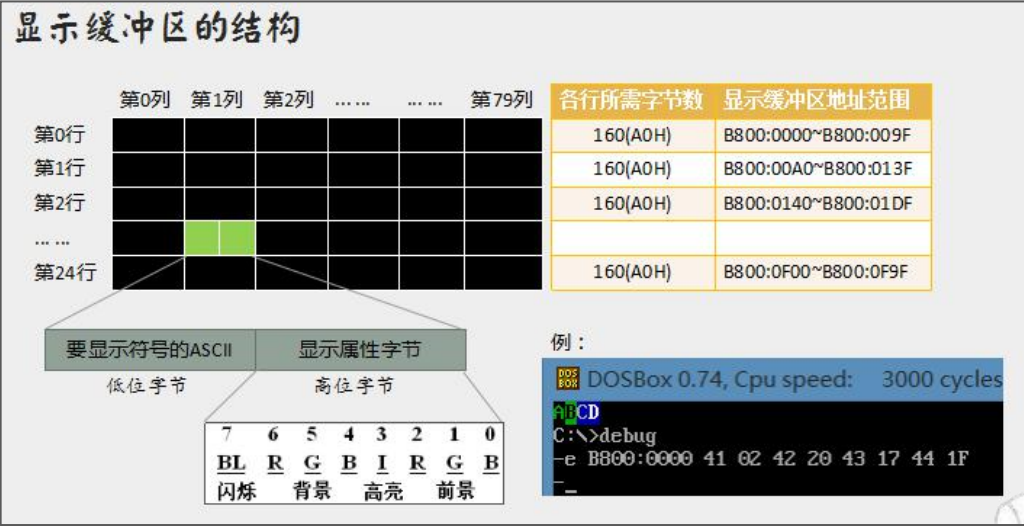
使用BIOS功能调用：(1) 方便编程；(2) 能写出简洁、可读性好、易于移植的程序。



BIOS中断调用示例

任务：在屏幕的5行12列显示3个红底高亮闪烁绿色的'a'

回顾



可以不这么细节吗？

用BIOS的10h中断

- (ah)=2时，调用第10h中断例程的2号子程序，设置光标位置
- (ah)=9时，调用第10h中断例程的9号子程序，在光标位置显示字符

```
assume cs:code
code segment
    mov ah,2      ;置光标功能
    mov bh,0      ;第0页
    mov dh,5      ;dh中放行号
    mov dl,12     ;dl中放列号
    int 10h

    mov ah,9      ;显示字符功能
    mov al,'a'    ;字符
    mov bl,11001010b;颜色属性
    mov bh,0      ;第0页
    mov cx,3      ;字符重复个数
    int 10h

    mov ax,4c00h
    int 21h
code ends
end
```

	7	6	5	4	3	2	1	0
颜色	<u>B</u> L	<u>R</u>	<u>G</u>	<u>B</u>	<u>I</u>	<u>R</u>	<u>G</u>	<u>B</u>
属性	闪烁	背景	高亮	前景				

有哪些BIOS中断，怎么用？

BIOS 中断类

- 1、显示服务(Video Service—INT 10H)
- 2、直接磁盘服务(Direct Disk Service——INT 13H)
- 3、串行口服务(Serial Port Service—INT 14H)
- 4、杂项系统服务(Miscellaneous System Service—INT 15H)
- 5、键盘服务(Keyboard Service——INT 16H)
- 6、并行口服务(Parallel Port Service——INT 17H)
- 7、时钟服务(Clock Service——INT 1AH)
- 8、直接系统服务(Direct System Service)

(9) 功能 09H

功能描述：在当前光标处按指定属性显示字符

入口参数：AH = 09H

AL = 字符

BH = 显示页码

BL = 属性(文本模式)或颜色(图形模式)

CX = 重复输出字符的次数

出口参数：无

1、显示服务(Video Service—INT 10H)

00H 一设置显示器模式	0CH 一写图形象素
01H 一设置光标形状	0DH 一读图形象素
02H 一设置光标位置	0EH 一在 Teletype 模式下显示字符
03H 一读取光标信息	0FH 一读取显示器模式
04H 一读取光笔位置	10H 一颜色
05H 一设置显示页	11H 一字体
06H、07H 一初始化或滚屏	12H 一显示器的配置
08H 一读光标处的字符及其属性	13H 一在 Teletype 模式下显示字符串
09H 一在光标处按指定属性显示字符	1AH 一读取/设置显示组合编码
0AH 一在当前光标处显示字符	1BH 一读取功能/状态信息
0BH 一设置调色板、背景色或边框	1CH 一保存/恢复显示器状态



汇编的强大功能，还有DOS中断！

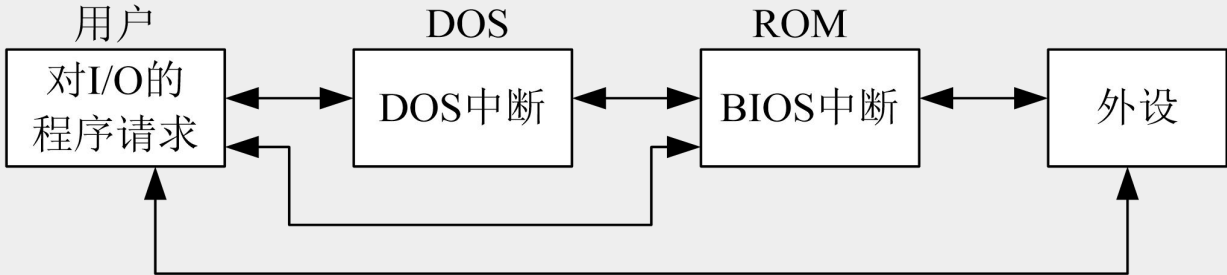
INT 20H	—终止程序运行
INT 21H	—功能调用
INT 22H	—终止处理程序的地址
INT 23H	—Ctrl+C 处理程序
INT 24H	—致命错误处理程序
INT 25H	—读磁盘扇区(忽略逻辑结构)
INT 26H	—写磁盘扇区(忽略逻辑结构)
INT 27H	—终止，并驻留在内存
INT 28H	—DOS 空闲
INT 2FH	—多重中断服务
INT 33H	—鼠标功能

DOS 中断类 (INT 21H 中断)
1、字符功能调用类(Character-Oriented Function)
2、目录控制功能(Directory-Control Function)
3、磁盘管理功能(Disk-Management Function)
4、文件操作功能(File Operation Function)
5、文件操作功能(FCB)(File Operation Function)
6、记录操作功能(Record Function)
7、记录操作功能(FCB)(Record Function)
8、内存分配功能(Memory-Allocation Function)
9、系统功能(System Function)
10、进程控制功能(Process-Control Function)
11、时间和日期功能(Time and Date Function)

```
mov ah, 2AH
int 21h
```

功能 2AH
功能描述：取系统日期
入口参数：AH = 2AH
出口参数：CX = 年 (1980~2099),
 DH = 月 (1~12),
 DL = 日 (1~31)
 AL = 星期几 (0=Sun, 1=Mon. ...)

- BIOS和DOS在所提供的中断例程中包含了许多子程序，这些子程序实现了程序员在编程的时常用到的功能。
- 和硬件设备相关的DOS中断例程中，一般都调用BIOS的中断例程。



int 21H DOS 中断例程的应用

💻 4ch号功能：程序返回

📁 功能号在ah，返回结果保存在al

💻 用法：mov ah,4ch

mov al,0

int 21h



mov ax,4c00h
int 21h

💻 09h号功能：在光标位置显示字符串

📁 ds:dx指向要显示的字符串(用'\$'结束)

💻 用法：mov ah,9

int 21h

💻 例：编程在屏幕的5行12列显示字符串
"welcome to masm!"。

```
assume cs:code
```

```
data segment
```

```
    db 'Welcome to masm!','$'
```

```
data ends
```

```
code segment
```

```
start: mov ah,2           ;置光标
```

```
        mov bh,0          ;第0页
```

```
        mov dh,5          ;dh中放行号
```

```
        mov dl,12         ;dl中放列号
```

```
        int 10h
```

```
        mov ax,data
```

```
        mov ds,ax
```

```
        mov dx,0          ;ds:dx指向字符串的首地址data:0
```

```
        mov ah,9
```

```
        int 21h
```

```
        mov ax,4c00h
```

```
        int 21h
```

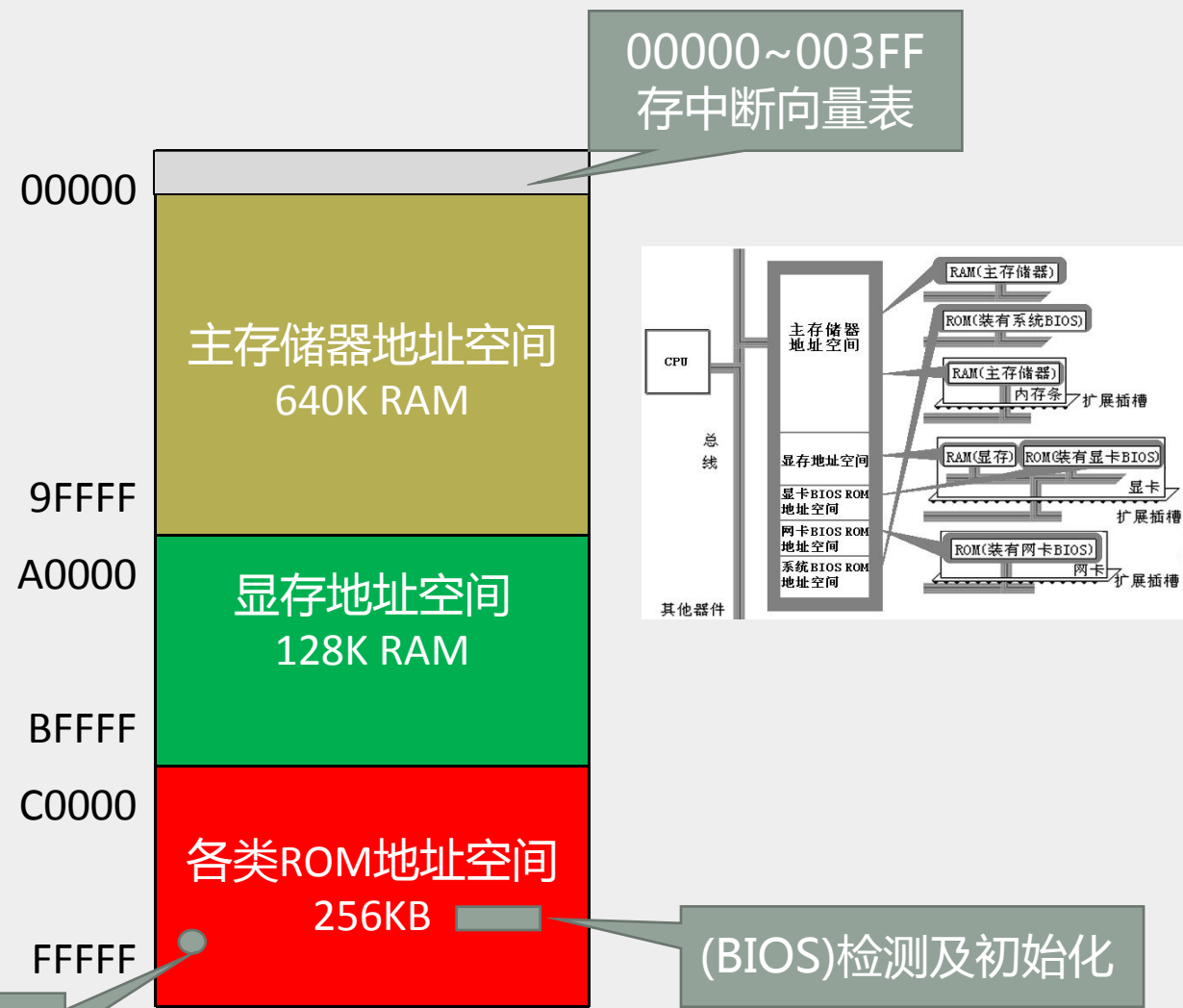
```
code ends
```

```
end start
```

BIOS和DOS中断例程的安装过程

- (1) CPU 一加电，初始化(CS)=0FFFFH，(IP)=0，自动从FFFF:0单元开始执行程序。FFFF:0处有一条转跳指令，CPU执行该指令后，转去执行BIOS中的硬件系统检测和初始化程序。
- (2) 初始化程序将建立BIOS 所支持的中断向量，即将BIOS提供的中断例程的入口地址登记在中断向量表中。
- (3) 硬件系统检测和初始化完成后，调用int 19h进行操作系统的引导。从此将计算机交由操作系统控制。
- (4) DOS 启动后，除完成其它工作外，还将它所提供的中断例程装入内存，并建立相应的中断向量。

FFFF:0 jmp...



(BIOS)检测及初始化