

第一章 引言

对于一个程序员来说,最令人沮丧的经历莫过于当他正聚精会神编写一段复杂程序时,忽然想不起一个必须使用的系统功能调用的规则和用法。IBM PC 机的程序员遇到这个问题时,为了获得有关 IBM PC、XT、便携机或 AT 机系统功能的基本信息,必须详细查阅 IBM 的有关手册,有时可能要连续查找多本手册和资料,甚至需要看晦涩难懂的汇编语言源程序。即使如此,他可能仍然需要经过反复试验,并对结果反复比较,才能掌握和运用一个特定的系统功能调用。

本书正是为这类程序员编写的。它对 IBM PC 机三个基本的编程服务:DOS 中断、DOS 功能调用和 BIOS(Basic Input /Output System)服务程序,都进行了完整的描述。用户查阅本书就象查阅字典一样方便迅速。

此外,对于愿意阅读程序而不愿看文字叙述的人,本书还提供了 200 多个使用 DOS 和 BIOS 调用的程序实例。这些实例用汇编、Microsoft C 和 Turbo Pascal 三种最流行的语言编写而成。它们个个都是独立的完整的程序,旨在说明如何准确使用 DOS 和 BIOS 提供的服务功能。

编写这本书时,我们设想一个程序员正在编写一个大型复杂的程序。他最需要的并不是程序编制的基本原则和一些不重要的细微末节,而是能够迅速查找到他需要了解的信息。因此,本书在编排上力求使读者不必翻阅许多无关章节就可以找到所需内容。我们希望本书能成为 IBM PC 机编程人员

的一本快速查询手册。

PC 机硬件概述

要想有效进行 PC 机的程序设计,就需要了解某些硬件系统的结构。本节将对 8086 及其有关的微处理器的基本的体系结构从硬件的角度上加以概述。

1. 基本寻址方式

8086 类型的处理器中,存储器的结构为 8 位字节。当处理的数量超过 8 位时,8086 则用最低地址存储最低有效字节。

Intel 系列的处理器采用的内存寻址技术为“段”(Segmentation)。简言之,一个段即为内存的一个区域。计算机可以处理多个段。在 DOS 运行的实方式中,每个段的长度定为 64K 字节,共有 65536 个段。各段重叠,每段在前一段之后的 16 字节开始。这就是为什么 DOS 的直接寻址不能超过 1M 字节(即 $65536 \times 16 = 1\,048\,576$ 字节)。8086 和 8088(XT 型机中所用的处理器)只能寻址 1M 字节。286、386 和 486 处理器可寻址更大的内存,但 DOS 不能直接访问。

各个分段的单元取值都从 0000H 到 FFFFH。由于每段为 64K 字节长,所以使用偏移量确定需要寻址的字节。8086 的完整地址应包括一个段地址和偏移量。如果段地址是 0040H,偏移量为 0102,则写为 0040 : 0102。因为段搭接每个为 16(10H)字节,所以地址 0000 : 0010 即等于地址 0001 : 0000。同样 0040 : 0000 和 0000 : 0040 相同,而且它们又都和 0020 : 0200 的单元相同。计算机中的分段地址以“反向方式”储存。例如 0040 : 1234 在计算机内存中出现的次序为:34 12

40 00。

程序形成的地址利用常数值和寄存器的组合来产生。有关寄存器设置的情况研究后再详述寻址问题。

2. 存储器类型

PC 机程序可以使用几种内存。PC XT 型的机器只能寻址 1M 字节内存。该内存称为常规存储器。在 XT 机器中可有 640K 字节的 RAM。I/O 设备、BIOS 的 ROM 以及其它系统资源使用其余 384K 字节的内存。

当程序运行超过这一空间限制时,应使用 Lotus、Intel 和 Microsoft 公司联合制定的 Expanded Memory Specification (EMS)扩展内存规范。当前常用的是 EMS 3.2 或 EMS 4.0 版本。大多数 EMS 系统由一块或几块 EMS 卡及其驱动程序的软件构成。程序通过驱动程序使用 EMS 内存,然而一次只能访问 64K 字节内存。EMS 驱动程序把这 64K 字节的内存块放在 1M 字节的地址空间上,下面通常为 BIOS 存储区。根据驱动器程序设计的要求,程序可以把 EMS 的 16K 字节页面映象到 64K 字节块之中。

即使 EMS 可以处理 32M 字节,但程序每次只能访问有限的 16K 字节的页面,所以用起来是很麻烦的,扩展内存只是 XT 机使用的一种额外内存。如果使用其它的额外内存,这种程序就可能不能在 8088/8086 机器上运行。

EMS 也可以用磁盘存储器来仿真。这需要另外的硬件,而且同硬件 EMS 系统相比是很慢的。

286、386 和 486 机都可以超过 1M 字节地址。286 可寻址 16M 字节,386 和 486 可寻址 4096M 字节(4G 字节)。从技术上讲,这种扩充内存只能在保护方式下访问。然而 BIOS 提供了一种功能,可在两个任意地址之间向前和向后传送数据,扩

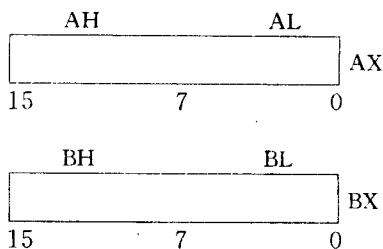
充内存中的数据也包括在内。

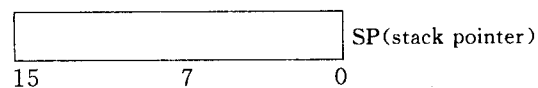
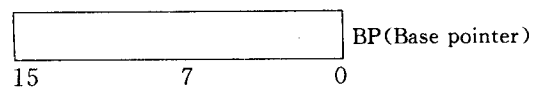
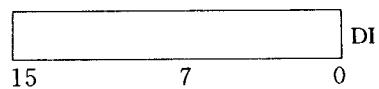
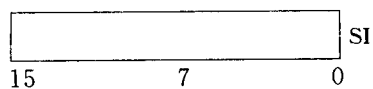
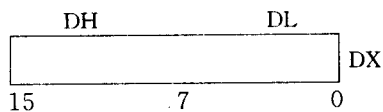
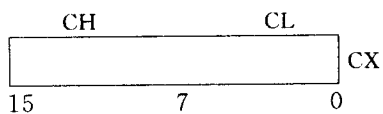
Extended Memory Specification(XMS)扩充存储器规范的驱动程序利用巧妙的寻址技术可使程序访问 64K 字节的扩充内存。注意,在保护方式下常规内存和扩充内存之间没有区别,扩充内存只从 1M 字节处开始设置。使扩充内存转换到扩展内存可以利用软件实现。在 286 机上这点不太有效,并非所有的 EMS 性能都可以仿真。然而用 386 和 486 的强大的内存管理能力,EMS 完全可以用软件有效地实现。

3. 寄存器

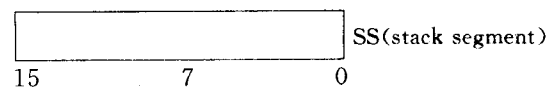
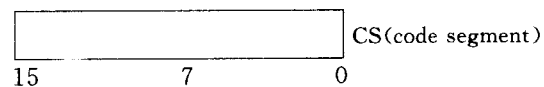
8086 处理器有 4 个通用寄存器、1 个标志寄存器、4 个段寄存器、2 个变址寄存器、1 个堆栈段寄存器和指示器、1 个基本寄存器和 1 个指令指针寄存器。图 1-1 示出了这些寄存器结构及在 286 处理器中增加的寄存器。图 1-2 示出了 386 和 486 处理器中具有寄存器。注意,286、386 和 486 处理器有许多专用寄存器。这些寄存器在 DOS 中不常用,但在保护方式扩充 DOS 时会使用它们。

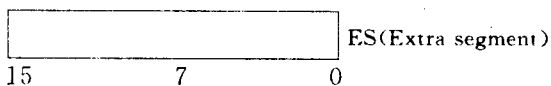
通用寄存器



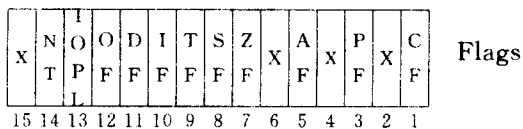
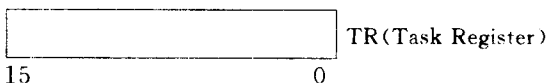
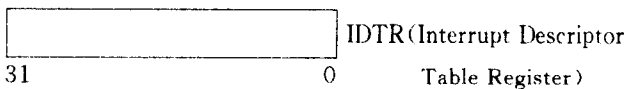
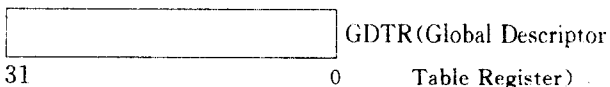
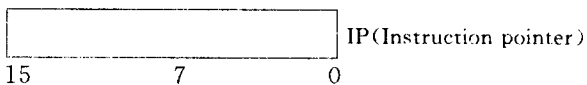


段寄存器





其它寄存器



X——保留

NT——嵌套任务标志(Nested task flag)

IOPB——输入输出特权级(I/O privilege level)(2位)

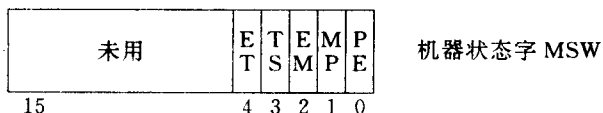
OF——溢出(overflow)

DF——方向标志(Direction flag)

IF——中断工作(Interrupt enable)

TF——陷阱标志(Trap flag)

SF ——符号标志(Sign flag)
 ZF ——零标志(Zero flag)
 AF ——辅助标志(Auxiliary flag)
 PF ——奇偶标志(Parity flag)
 CF ——进位标志(Carry flag)



ET ——协处理器类型(Type of coprocessor)
 TS ——交换任务(Task-switched)
 EM ——仿真协处理器(Emulate coprocessor)
 MP ——数学协处理器(Math coprocessor present)
 PE ——保护方式工作(Protected-mode enable)

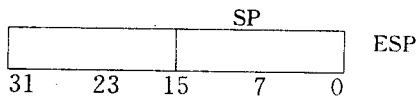
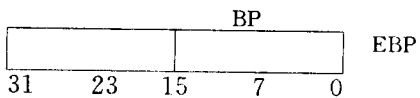
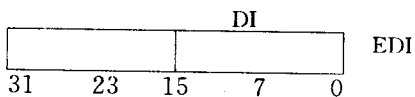
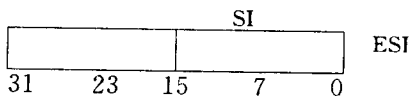
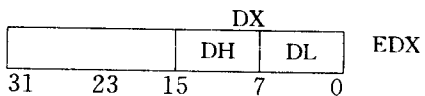
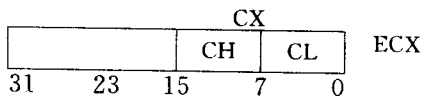
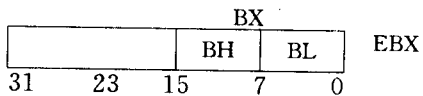
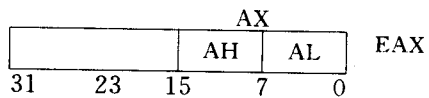
图 1-1 8088/8086/80286 寄存器

AX、BX、CX 和 DX 都是通用寄存器：

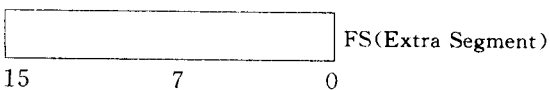
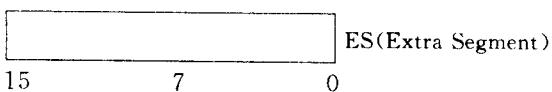
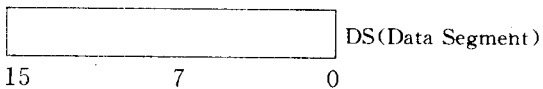
- . AX 有时可作为累加器,用于算术运算。
- . BX 只作为通用寄存器使用,可参加形成实方式下的地址。
- . CX 作循环时计数,作重复性操作。
- . DX 规定 I/O 地址,特别是大于 0FFH 的那些地址,还可在乘除法指令中同 AX 一起工作。

每个 16 位的通用寄存器都可作为双 8 位寄存器使用。例如 AH 和 AL 实际上是 AX 的半个寄存器。386/486 中有同样的通用寄存器,只是每个都是 32 位宽。32 位的寄存器开始用字母 E 表示,如 EAX、EBX 等。不能直接访问 32 位寄存器的高 16 位。

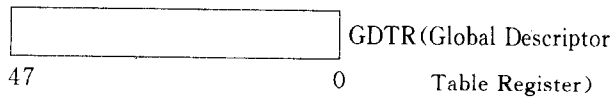
通用寄存器

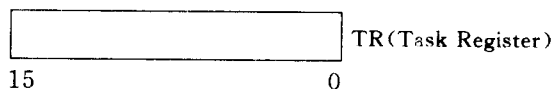
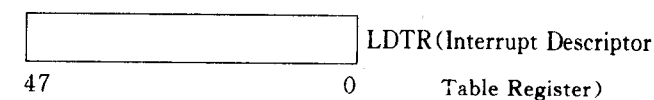


段寄存器

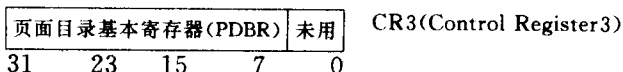
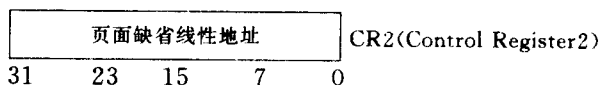
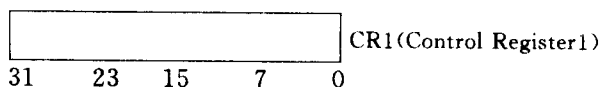


其它寄存器





(注: 8个调试寄存器 DR0~DR7 未给出)



PG —— 分页面工作 (Paging enable)

ET —— 协处理器类型 (Type of coprocessor)

TS —— 交换任务 (Task-Switched)

EM —— 仿真协处理器 (Emulate coprocessor)

MP —— 数学协处理器 (Math coprocessor present)

PE —— 保护方式工作 (Protected-mode enable)

图 1-2 80386/80486 寄存器

从理论上讲,通用寄存器可用在任何场合,然而每个寄存器都有一定的专用功能。有时必须给寄存器某个特定功能,换句话说,用某个寄存器可能比用其它的更有效。

段寄存器(CS、DS、SS 和 ES)可指出当前的代码段(CS)、数据段(DS)和堆栈段(SS),ES 是额外段,可在各段之间传送数据。这些段可参加地址计算。386/486 处理器又增加了两个段寄存器 FS 和 GS。

两个变址寄存器 SI 和 DI 主要用于形成地址,一般不作通用寄存器,它们不能分成两半。在 386/486 中可用 32 位地址的 ESI 和 EDI,或者 16 位地址的 SI 和 DI。

8086 系列为自身及程序员提供了一个堆栈,作为暂时保存数据的区域,在调用子程序和中断功能中起重要作用。该堆栈段寄存器(SS)和堆栈指针(SP 或 386/486 ESP)指向堆栈的顶部。为确定当前堆栈的顶部,使用 SS 寄存器作为段值和 SP 寄存器作为偏移值。

基本寄存器 BP(386/486 中为 EBP)指向堆栈顶。它在很多方面类似于 SI 和 DI 寄存器。使用其它寻址方式时,差别将明显不同。

指令指针寄存器 IP 是下一条指令的代码段的偏移值。它不能直接修改 IP 的内容。但跳转指令和子程序调用指令可以做到。386/486 中使用 32 位的 EIP。

标志寄存器包括一系列决定处理器状态的数据位。其中有算术运算操作的溢出位和处理器响应外部事件或中断的控制位。程序使用特定的指令即可以处理该寄存器内容,令其置位或清除。

4. 再论寻址方式

在实方式中,8086 系列处理器用某些数值的组织只能形成地址。形成一个完整地址,必须计算偏移值并确定使用哪一段。

下表说明怎样形成偏移值。在每一行都可以挑选一个或

下一行的寄存器值,把这些数值加到一起即可得到偏移值。

基本寄存器	变址寄存器	常数
BX	SI	8 位常数
BP	DI	16 位常数

例如,SI、20、BX、BX+DI、BX+SI+5 和 BX+10 等全是有数的偏移值,而 BX+SI+DI 则不是,因为它用了两个变址寄存器的内容。常数值可以为负,BX-10 也是合法值。

有关 CPU 使用段值的说明如下:通常,除非使用 BP 寄存器进行偏移值计算,否则段值本身都是指 DS 寄存器中的数值。当用 BP 做偏移值计算时,CPU 使用 SS 寄存器,然而对某个特定的前缀可能使用特定的段寄存器,如下所示:

指令	该地址上的字 装入 AX	说明
MOV AX,[BX+10]	DS:[BX+10]	DS 缺省
MOV AX,[BP]	SS:[BP]	SS 缺省
MOV AX,ES:[BP]	ES:[BP]	ES 超限
MOV AX,CS:[102H]	CS:[102]	CS 超限
MOV AX,DS:[BP]	DS:[BP]	DS 超限

有些串指令(如 MOVS)隐含使用 DS:SI 和 ES:DI。这些指令中,超限只影响 SI。指令中对某一类地址,总是使用 ES 和 DI。

5. 端口

8086 系列处理器提供 I/O 端口以把外部设备连到 CPU 上。使用专门的命令可以向这些端口读写数据。并非所有 I/

O 设备都使用这些端口,例如显示卡即使用内存的地址块来代表屏幕。这时它们使用 I/O 端口实现某些控制功能,然而,访问 CPU I/O 端口的最常用的方法是采用输入(IN)和输出(OUT)指令,例如:

IN AL,20H

即把 I/O 端口 20H 的数值读到 AL 寄存器中。其值的内容取决于在该地址上的硬件设备。又如:指令 OUT 20H,AL 是把 AL 寄存器中的数值输出到 20H 端口。该指令的作用依赖于硬件。

6. 硬件中断

硬件中断是从 I/O 设备发给计算机的一种特殊的信号。该信号通知计算机注意 I/O 设备的请求。此时处理机不管在做什么,通常都要停止下来去响应中断。当该中断完成处理时,处理机在其停止处重新开始继续执行。

8086 处理机系列可以响应 256 种中断。从单元 0000 : 0000 开始,每一中断都有一个地址表,如图 1-3 所示。在该中断向量表中,每一入口为 4 字节长,这对于段和偏移量已经足够了。例如,如果处理机接收到中断 2(INT2),则把 CS 和 IP 的标志及当前值保留在堆栈。然后处理机取得单元 0000 : 0008 的内容(图 1-3 中 F000 : 0103),并执行该地址中的中断服务程序。在 ISR 中的 IRET 指令通知中断结束,并使处理机继续执行中断前的工作。

中断可由许多中断源产生,例如对键盘每次击键或放键都会形成中断。还有由时钟、打印机、串行端口、磁盘驱动器等多种设备产生的中断。处理机本身也能产生某些中断。例如当程序中发生用 0 除某数时就会产生 INT0 的中断。以上这些都是硬件中断,本书以后将重点介绍软件中断。

0000:0000	027F:56E8	INT 00H 向量
0000:0004	460C:1180	INT 01H 向量
0000:0008	F000:0103	INT 02H 向量
0000:000C	460C:1182	INT 03H 向量
0000:0010	•	
	•	
	•	
0000:03FC		INT FFH 向量

图 1-3 中断向量表简例

在 PC 机系统的主板上设有 8259 可编程的中断控制器芯片来管理所有硬件中断。PC/XT 型微机只有一个 PIC，PC/AT 型微机有两个 PIC。这些控制器的作用是把各种设备产生的中断传给处理机以产生特殊的中断。

表 1-1 列出了硬件中断及其相应的 PIC 中断请求输入端（即 IRQ 线）。注意，勿将 IRQ 的数目同中断号（数值）相混淆。例如，键盘接到 IRQ1，而给 CPU 发出的中断为 INT9。PIC 可以再编程，对每一个 IRQ 都产生不同的中断号，除非系统在保护方式下工作，否则不要经常这样做。

PIC 还要控制中断的优先级，例如，时钟（IRQ0）的优先级就高于键盘（IRQ1）的优先级。如果处理机正服务于时钟中断程序，那么，在时钟的 ISR 复位 PIC 之前，PIC 就不会产生键盘中断。另一方面，时钟可以中断键盘的 ISR。PIC 本身可以进行编程，能用于各种优先级方案。

AT 机中的另一个 PIC 接到第一个 PIC 的 IRQ2 端。所以，增加的这个 PIC 的 IRQ 端（8~15）的优先级与 IRQ2 相

同。禁止 IRQ2 时将禁止第二个 PIC 上的所有中断。

表 1-1 硬件中断及其相应的 IRQ 线

偏移值	中断	IRQ	机型	说明
0000H	00H		所有	被 0 除或除法溢出
0008H	02H		所有	NMI
0010H	04H		所有	溢出(由 INT0 产生)
0020H	08H	0	所有	系统定时器
0024H	09H	1	所有	键盘
0028H	0AH	2	XT	通用 I/O
0028H	0AH	2	AT+	第二个 PIC 中断
002CH	0BH	3	所有	COM2
0030H	0CH	4	所有	COM1
0034H	0DH	5	XT	硬盘
0034H	0DH	5	AT	LPT2
0034H	0DH	5	PS/2	通用 I/O
0038H	0EH	6	所有	软盘
003CH	0FH	7	所有	LPT1
01C0H	70H	8	AT+	实时时钟
01C4H	71H	9	AT+	通用 I/O(同 IRQ2)
01C8H	72H	10	AT+	通用 I/O
01CCH	73H	11	AT+	通用 I/O
01D0H	74H	12	AT	通用 I/O
01D0H	74H	12	PS/2	指针设备(鼠标)
01D4H	75H	13	AT+	协处理器
01D8H	76H	14	AT+	硬盘
01DCH	77H	15	AT+	通用 I/O

注:AT+为 AT 386/486 或 PS/2

为避免某些主要的程序在一些环境下受到干扰,也可以停止处理中断。采用 CLI 指令即可以禁止所有外部中断,除了非屏蔽中断(NMI)之外。NMI 本身不通过 PIC,可以用主

机板上的电路禁止其工作。另外,对 PIC 也可用编程的方法来关闭特殊的中断。

DOS 和 PC 机都依赖于中断的处理。了解中断是掌握 PC 机的关键,用户一定要给予重视。

7. 定时器

所有 PC 机主板上都设有一定定时器芯片(XT 机为 8253, AT 机为 8254)。该定时器有三个独立时钟,或称通道。随机存储器 RAM 使用一个时钟定时进行周期刷新(该时钟通道对编程无用)。另一个时钟通道运行每天时间。每秒钟产生约 18 次信号,故该定时器产生 INT8(IRQ0),是最高优先级中断。磁盘操作使用该通道的输出,在磁盘操作以后校正每天的时间时钟。

还有一个通道是程序员最感兴趣的,它不仅可以任意使用,而且该备用通道的输出可有选择地送给扩音器,由此产生基本发声或音乐。

8. 通用 I/O

PC/XT 机使用 8255 外设接口控制器作某些功能服务。虽然 AT 机没有 PIA 控制器,但它备有大部分功能。PIA 有三种 I/O 端口,可同键盘通信,读出主板开关以及控制扩音器。

当然,AT 机不能读出主板各个开关的内容,相反它可用 SETUP 程序向其实时时钟芯片中的存储器写入配置数据。由于实时时钟用电池供电,所以当电池用完或改变配置时,只能运行 SETUP 程序。控制实时时钟端口也可以决定 NMI 是否进行工作(控制 NMI 时,XT 使用不同的端口)。

PC 机主板上还设有 8237A 直接存储器存取(DMA)控制器。它可以直接把某些设备(如硬盘)的数据迅速地传送到

计算机的内存,无须 CPU 介入。在设计上,8237A 使用计算机的 16 位地址总线工作。

在主板上还设有 DMA 页面寄存器,以适应原来的 8088/8086 CPU 芯片。当使用 DMA 时,这些寄存器保持高 4 位的地址。因此,DMA 控制器可以寻址整个 1M 字节的地址空间,但是每次只能存取 64K 字节。一个 DMA 通道在定时器芯片的控制下还可以完成动态存储器的刷新。

还有许多种 I/O 卡可以接到 PC 机上。并行口可以接打印机,串行口还可以接多种外部设备。

9. 键盘

PC 键盘本身带有 CPU,XT 机的接口经过 8255 PIA 控制器同主机相连。AT 机没设置这种 PIA,当同键盘通信时,软件还不能告之有这种差别。

AT 机的键盘功能在设计上有些技巧。例如,它的键盘可以接收命令,可以复位计算机,测试键盘本身功能,控制内存地址线以及其它任务。

每次用户按某个键时,PIA 即经由系统的 PIC 产生 IRQ1。此时处理机接收 INT9。如果处理机正在受理中断,则 ISR 从键盘上读一扫描码。准确地说,当按下或放开电键时,键盘就会发生中断。每个电键产生两个扫描码:一个是按下电键,另一个是放开电键。然而键盘不知道有关被移位的字母。例如,A,a 和 Alt-A 都产生相同的扫描码,由于每个代码代表一个物理键,即使没有 ASCII 编码的各个键,如 Alt 和 Shift,也都产生扫描码。

BIOS 处理每个 IRQ1,以决定做什么动作。某些键和组合键可直接产生结果(如 Control-Alt-Delete)。BIOS 保持内存中键盘的状态,当键本身改变后才进行更新(如 Shift 和