

IBM^{PCXT} 微电脑

技术参考手册

下

-62

-2

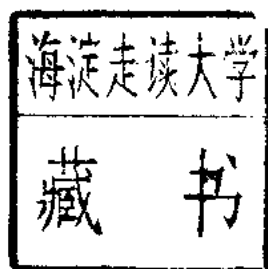
IBM PC XT

微电脑技术参考手册

翻译：韩 迈
尹 光

校对：丁伟之

(下册)



010330

国营长城科学仪器厂

下册 ROM和系统

用法及附录

ROM BIOS	2-2
键盘编码和用法	2-11
附录A	
附录B	
附录C	
附录D	
附录E	
附录F	
附录G	
词汇表	

ROM BIOS和系统用法

ROM BIOS

基本输入/输出系统 (BIOS) 驻留在系统板上的ROM中, 并且为系统中的主要I/O设备提供设备一级的控制。附加的ROM模块可以放在选择的适配器上, 为选择的适配器提供设备一级的控制。BIOS例行子程序使得汇编语言的程序员无需关心设备的地址和操作的特性便可执行硬盘和软盘或字符一级的I/O操作。BIOS提供了系统服务, 如时钟和内存容量的确定。

这样做的目的是为了给系统提供一个操作接口, 以减轻程序员的负担, 不必更多了解硬件设备的特性。BIOS接口使得用户与硬件隔离, 这样就允许给系统增加新的设备, 同时为这设备在BIOS一级保留接口。采用这种方法, 用户对于硬件的修改和增加是透明的。

IBM个人计算机的宏汇编语言手册和IBM个人计算机硬盘操作系统 (DOS) 手册提供了与这一节有关的有用的程序设计内容。在附录A中给出了一个完整的BIOS列表。

BIOS的使用

对BIOS的访问是通过8088软件中断来实现的。每个BIOS入口点都可以通过它自己的中断进入, 这在“8088软件中断列表”中可以找到。

软件中断, 从十六进制的10到1A, 每一个都可访问一个不同的BIOS例行子程序。例如, 要确定系统中可使用的内存数量,

INT 12H

就可请求BIOS子程序来确定存储池大小, 并且返回一个值给调用程序。

参数传递

所有传递给BIOS子程序或从BIOS子程序返回的参数都要通过8088

寄存器。每个BIOS功能的序言中都指出了调用和返回使用的寄存器。内存容量这一例，没有参数传递。内存容量（按1k字节增加）返回在AX寄存器中。

如果某一BIOS功能有几种可能的操作，就在输入时使用AH寄存器指出所需的操作。例如，要设置时钟，需要如下代码：

MOV AH,1 : 功能是要设置时间

MOV CX,HIGH-COUNT : 建立当前时间

MOV DX,LOW-COUNT

INT 1AH : 设置时间

为读这一时间：

MOV AH,0 : 功能是要读时间

INT 1AH : 读时钟

一般情况下，BIOS程序除了AX和标志以外保护所有的寄存器。仅当其它寄存器正在给调用者返回一个值时才被修改，寄存器的真正使用可以在每个BIOS功能的开头找到

8088软件中断列表

地 址 (十六进制)	中 断 (十六进制)	名 称	BIOS初始化
0-3	0	除零	D11
4-7	1	单步	D11
8-B	2	不可屏蔽的	NMI_INT
C-F	3	断点	D11
10-13	4	溢出	D11
14-17	5	打印屏幕	PRINT_SCREEN
18-1B	6	保留	D11
10-1F	7	保留	D11
20-23	8	时钟	TIMER_INT
24-27	9	键盘	KB_INT
28-2B	A	保留	D11
2C-2F	B	通讯	D11
30-33	C	通讯	D11
34-37	D	硬磁盘	D11
38-3B	E	软盘	DISK_INT
3C-3F	F	打印机	D11
40-43	10	终端	VIDEO_IO
44-47	11	设备检查	EQUIPMENT
48-4B	12	内存	MEMORY_SIZE_DETERMINE
4C-4F	13	软盘/硬盘	DISKETTE_IO
50-53	14	通讯	RS232_IO
54-57	15	磁带	CASSETTE_IO
58-5B	16	键盘	KEYBOARD_IO
5C-5F	17	打印机	PRINTER_IO
60-63	18	驻留BASIC	F600'0000
64-67	19	引导	BQOT_STRAP
68-6B	1A	时钟	TIME_OF_DAY
6C-6F	1B	键盘死锁	DUMMY_RETURN
70-73	1C	时钟滴答声	DUMMY_RETURN
74-77	1D	终端初始化	VIDEO_PARMS
78-7B	1E	软盘参数	DISK_BASE
7C-7F	1F	终端图形特性	0

具有特殊意义的向量:

中断码IB—键盘锁住地址

当按了键盘的Ctrl和Break键时,这个向量指向键盘死锁的代码程序。响应键盘中断时,请求这一向量,控制应通过IRET指令返回。加电子程序初始化这个向量使其指向一条IRET指令,所以当按下Ctrl和Break键时不会发生什么问题,除非这个应用程序设置了不同的值。

控制被这个子程序保持时,具有以下问题。Break也许在中断处理过程中已经发生,所以必须给8259控制器发送一个或多个中断命令的结束,同时,所有的I/O设备应当复位以防有某一操作正在进行。

中断IC—时钟滴答声

这个向量指向按每个系统时钟滴答声执行的代码程序,当响应时钟中断时请求这个向量,并且控制应通过IRET指令返回。加电子程序初始化这个向量指向IRET指令,所以除非应用程序修改了这个指针,否则不会发生什么问题。应用程序要对所有做修改的寄存器进行保护和恢复。

中断ID—终端参数

这个向量指向一数据域,这数据域含有初始化终端插件板上的6845所需的参数。注意有四个不同的表,如果要支持所有的操作方式,这四个表必须都建立。加电子程序初始化这个向量,指向ROM终端子程序中所含的参数。

中断码1E—软盘参数

这个向量指向软盘驱动器所需的参数域，加电子程序初始化这个向量，使其指向ROM软盘子程序所含的参数。这些缺省参数是为安装到机器上的IBM驱动器指定的值。当另外接上驱动器时，可能要对这个参数块加以修改，以便与另接的驱动器的特性相符。

中断码1F—图形特性扩展

采用IBM彩色/图形显示器适配器（320×200,640×200）的图形方式操作时，读/写字符接口根据ASCII码使用一系列点阵形成字符。头128个字符的点阵存在ROM中。为了访问后128个字符码点，必须使该向量指向一个多达1K字节的表，其中每一个字符码点用八个字节的图形数据来表示。刚加电时，该向量被初始化为000:0。如需要附加字符码点，用户应改变这个向量。

中断码40H—保留

当装上IBM硬盘驱动器适配器时，BIOS子程序用中断向量码40H作为软盘指针向量。

中断码41H—硬盘参数

这个向量指向存贮着硬盘驱动器所需参数的数据区域。加电子程序初始化该向量，使其指向ROM磁盘子程序中的参数。这些缺省值就是连到计算机上的任何IBM硬盘驱动器所需的特定值。当接上其它类型的硬盘驱动器时，可能要修改这个参数，以与这种类型的硬盘驱动器的特性相符。

其它读/写存储器用途

IBM BIOS子程序占用内存256个字节，从绝对地址100（十六进制）到4FF（十六进制）。从400到407（十六进制）的地址用来存放接到系统中的RS-232C插件板的基地址。408到40F（十六进制）用来存放打印机适配器的基地址。

内存地址300~3FF（十六进制）用做加电初始化过程中的堆栈域，当控制从加电转向引导加载时，代表加载程序使用的栈域。如果用户希望该栈区位于别的区域，则这个区域应由应用程序设定。

BASIC和DOS保留的中断向量

地 址 (十六进制)	中 断 (十六进制)	功 能
80-83	20	DOS程序结束
84-87	21	DOS功能调用
88-8B	22	DOS结束地址
8C-8F	23	DOS CTRL-BRK 退出地址
90-93	24	DOS致命错误向量
94-97	25	DOS全值磁盘读
98-9B	26	DOS全值磁盘写
9C-9F	27	DOS结束，固定在存储器中
A0-FF	28-3F	为DOS保留
100-17F	40-5F	保留
180-19F	60-67	为用户软件中断保留
1A0-1FF	68-7F	不用
200-217	80-85	由BASIC保留
218-3C3	86-FO	当BASIC运行时，由BASIC解释程序使用
3C4-3FF	F1-FF	不用

保留的存储器单元

地址 (十六进制)	方 式	功 能
400—48F	ROM BIOS	见BIOS清单
490—4CF	DOS	由DOS方式命令使用
4D0—4EF		保留
4F0—4FF		为应用程序保留, 作为内部应用通信区
500—5FF		为DOS和BASIC保留
500	DOS	打印屏幕状态标志存储区
		0—打印屏幕未做或者打印屏幕操作成功
		1—打印屏幕在进行中
		255—打印屏幕操作期间遇到了错误
504	DOS	单驱动器方式状态字节
510—511	BASIC	BASIC段地址存储区
512—515	BASIC	时钟中断向量段: 偏移存储区
516—519	BASIC	间断键中断向量段: 偏移存储区
51A—51D	BASIC	磁盘错误中断向量段: 偏移存储区

BASIC 工作空间的变量

	偏 移	长 度
当前正执行的行号	'2E'	2
最后出错的行号	'347'	2
到程序文本的起始段的偏移	'30'	2
到变量的起始处的偏移 (程序文本1—1的结尾)	'358'	2
键盘缓冲区内容	'6A'	1
0—缓冲区无字符		
1—字符在缓冲区		
如果执行POKE&H6A, 0		
将冲掉缓冲区的任何字符		

例:

```
100 PRINT PEEK (&H2E) + 256 * PEEK (&H2F)
```

L

H

X'64'

X'00'

十六进制的开始地址

00000	BIOS 中断向量
00080	可使用的 中断向量
00400	BIOS 数据域
00500	用户读/写内存
C8000	磁盘适配器
F0000	只读存储器
FE000	BIOS 程序域

BIOS编程提示

BIOS代码是通过软件中断请求的，程序员不应当将BIOS地址加在应用程序中。BIOS中的内部过程和绝对地址禁止随意更改。

如果磁盘或软盘代码报告一个错误，应当复位这个驱动器并重复这一操作。在做软盘读操作时应需要指定重复的次数，以确定这个问题不是由于驱动器的启动引起的。

修改 I/O 端口位的值时，程序员应当只改变当前任务所需的那些位，完成后，程序员应恢复原来的环境。没有做这一工作可能会与当前的和以后的应用程序不相配。

系统可访问的ROM组件的适配器插件板

ROM BIOS为板上带有ROM程序的适配器插件板一并进入系统提供了便利条件。在POST期间，为BIOS调用建立了中断向量，缺省向量设置好之后，就对附加的ROM组件扫描。此时适配器板上的ROM程序可以获得控制。子程序可以建立或阻止将其自身挂到系统上的中断向量。

为寻找有效的适配器插件板ROM，要对绝对地址C8000H到C4000H以2K为单位进行扫描。一个有效的ROM按下列定义：

字节0:55H

字节1:AAH

字节2:长度指示器，表示ROM中512个字节为一块的信息块个数（长度/512）。

为测试ROM组件的完整性，需求一个检查和。在被定义的ROM中，每一字节按模100H求和，求和必须为0，该模块才被认为有效。

当POST找出一个有效的ROM时，它对ROM的字节3（将是一个可执行的码）做一次远调用（far call）。适配器板现在可以执行它的加电初始化任务。适配器ROM应通过执行一条段间返回（far return）指令，使控制返回到BIOS程序。

键盘编码和用途

编码

键盘程序由IBM的ROM BIOS提供，它用来转换键盘扫描码为“扩展的ASCII码”。

扩展的ASCII码是可能取值0到255的一个字节的字符码，这些扩展码，用于某些键盘扩充功能和在键盘程序中通过中断而处理的功能。

字符编码：

通过BIOS键盘子程序，下列字符码被送到系统或应用程序中。

“-1”表示这种组合在键盘子程序中是禁止的。字符的编码被回送到AL中。更确切的编码请看附录C。并参阅“键盘扫描编码图”（在第一节）

字符编码

键 号	下 档	上 档	Ctrl	Alt
1	Esc	Esc	Esc	-1
2	1	!	-1	注1
3	2	@	Nul (000) 注1	注1
4	3	#	-1	注1
5	4	\$	-1	注1
6	5	%	-1	注1
7	6	^	RS(030)	注1
8	7	&	-1	注1
9	8	*	-1	注1
10	9	(	-1	注1
11	0	)	-1	注1
12	.	_	US(031)	注1
13	=	+	-1	注1
14	Backspace (008)	Backspace (008)	Del (127)	-1
15	→ (009)	← (Note 1)	-1	-1
16	q	Q	DC1 (017)	注1
17	w	W	ETB (023)	注1

字符码 (续)

键 号	下 档	上 档	Ctrl	Alt
18	e	E	ENQ (005)	注 1
19	r	R	DC2 (018)	注 1
20	t	T	DC4 (020)	注 1
21	y	Y	EM (025)	注 1
22	u	U	NAK (021)	注 1
23	i	I	HT (009)	注 1
24	o	O	SI (015)	注 1
25	p	P	DLE (016)	注 1
26	[	{	Esc (027)	-1
27	]	}	GS (029)	-1
28	CR	CR	LF (010)	-1
29 Ctrl	-1	-1	-1	-1
30	a	A	SOH (001)	注 1
31	s	S	DC3 (019)	注 1
32	d	D	EOT (004)	注 1
33	f	F	ACK (006)	注 1
34	g	G	BEL (007)	注 1
35	h	H	BS (008)	注 1
36	j	J	LF (010)	注 1
37	k	K	VT (011)	注 1
38	l	L	FF (012)	注 1
39	:	:	-1	-1
40	;	;	-1	-1
41	'	'	-1	-1
42 Shift	-1	-1	-1	-1
43	\		FS (028)	-1
44	z	Z	SUB (026)	注 1
45	x	X	CAN (024)	注 1
46	c	C	ETX (003)	注 1
47	v	V	SYN (022)	注 1
48	b	B	STX (002)	注 1
49	n	N	SO (014)	注 1
50	m	M	CR (013)	注 1
51	,	<	-1	-1
52	.	>	-1	-1
53	/	?	-1	-1
54 Shift	-1	-1	-1	-1
55	*	(Note 2)	(Note 1)	-1
56 Alt	-1	-1	-1	-1
57	SP	SP	SP	SP
58	-1	-1	-1	-1
Caps Lock				
59	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)
60	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)
61	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)
62	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)
63	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)
64	Nul (注 1)	Nul (注 1)	Nul (注 1)	Nul (注 1)

字符码 (续)

键 号	下 档	上 档	Ctrl	Alt
65	Nul (注1)	Nul (注1)	Nul (注1)	Nul (注1)
66	Nul (注1)	Nul (注1)	Nul (注1)	Nul (注1)
67	Nul (注1)	Nul (注1)	Nul (注1)	Nul (注1)
68	Nul (注1)	Nul (注1)	Nul (注1)	Nul (注1)
69 Num Lock	-1	-1	Pause (注2)	-1
70	-1	-1	Break (注2)	-1
Scroll Lock				
注1: 参见本节的“扩展码”部分 2: 参见本节的“特殊操作”部分				

71到83号键仅在下档, Num Lock (或shifted) 状态, 或Ctrl 状态时有意义。应注意, shift键会暂时使当前的Num Lock状态无效。

字符码 (续)

键 号	Num Lock	下 档	Alt	Ctrl
71	7	Home (注1)	-1	清除屏幕
72	8	↑ (注1)	-1	-1
73	9	Page Up (注1)	-1	到文本顶部光标回位
74	-	-----	-1	-1
75	4	← (注1)	-1	倒退一个字(注1)
76	5	-1	-1	-1
77	6	→ (注1)	-1	前进一个字(注1)
78	+	+	-1	-1
79	1	End (注1)	-1	一直删除到行尾(注1)
80	2	↓ (注1)	-1	-1
81	3	Page Down (注1)	-1	一直删除到屏幕末尾
82	0	Ins	-1	-1 (注1)
83		Del (注1, 2)	注2	注2
注1: 参见本节的“扩展码”部分 2: 参见本节的“特殊操作”部分				

扩展码

扩展功能

某些功能无法用标准的ASCII编码表示，这时就要使用扩展码。字符编码000（空）被回送到AL中。这表明，系统或应用程序应检查第二个编码，给出了实际功能。通常，但不总是，第二个编码是所按过的下档键的扫描码。这个码被回送到AH中。

键盘扩展功能

第二个代码	功 能
3	Nul Character
15	←
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
30-38	Alt A, S, D, F, G, H, J, K, L
44-50	Alt Z, X, C, V, B, N, M
59-68	F1 to F10 Function Keys Base Case
71	Home
72	↑
73	Page Up and Home Cursor
75	←
77	→
79	End
80	↓
81	Page Down and Home Cursor
82	Ins (Insert)
83	Del (Delete)
84-93	F11 to F20 (Upper Case F1 to F10)
94-103	F21 to F30 (Ctrl F1 to F10)
104-113	F31 to F40 (Alt F1 to F10)
114	Ctrl PrtSc (Start/Stop Echo to Printer)
115	Ctrl ← (Reverse Word)
116	Ctrl → (Advance Word)
117	Ctrl End [Erase to End of Line (EOL)]
118	Ctrl PgDn [Erase to End of Screen (EOS)]
119	Ctrl Home (Clear Screen and Home)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = (Keys 2-13)
132	Ctrl PgUp (Top 25 Lines of Text and Home Cursor)

Shift状态

大多数shift状态是在键盘子程序中处理的，对系统和应用程序是透明的。在任何情况下，均可调用ROM键盘子程序的一个入口点来得到一组现行起作用的shift状态。使用下列字键将会改变shift状态。

Shift

这个字键暂时将字键2—13, 15—27, 30—41, 43—53, 55 和 59—68 变换成上档字符。（如在Caps Lock状态下是下档）。并且，shift 键还能将字键71—73, 75, 77和79—83的Num Lock或non-Num-Lock 状态翻转。

Ctrl

这个键暂时变更 3, 7, 12, 14, 16—28, 30—38, 43—50, 55, 59—71, 73, 75, 77, 79和81字键为Ctrl状态。它还可以与Alt和Del键一起使用来完成“system reset”(系统复位)功能;和Scroll Lock (滚动锁定)键同时使用来完成“break”(间断)功能;和Num Lock 键同时使用来完成“pause”(停机)功能。系统复位, 间断和停机功能在后面的“Special Handling”(特殊处理)中有所描述。

Alt

这个键将 2—13, 16—25, 30—38, 40—50, 59—68号字键暂时变成Alt状态。它还可与Ctrl键和Del键同时使用来完成后面章节“special Handling”(特殊处理)所描述的“system reset”(系统复位)功能。

Alt键还有其它用途。这个键允许用户从键盘将编码为0—255的字符输入给系统。用户按下Alt键并使用数字键盘(字键71—73, 75—77, 79—82)打入所要求字符的十进制数值。然后释放Alt键。如果打入的数字多于3位, 则产生一个以256为模的数。这三位数字被解释成一个字符的编码并通过键盘子程序传输给系统或应用程序。Alt键由键盘子程序内部处理。