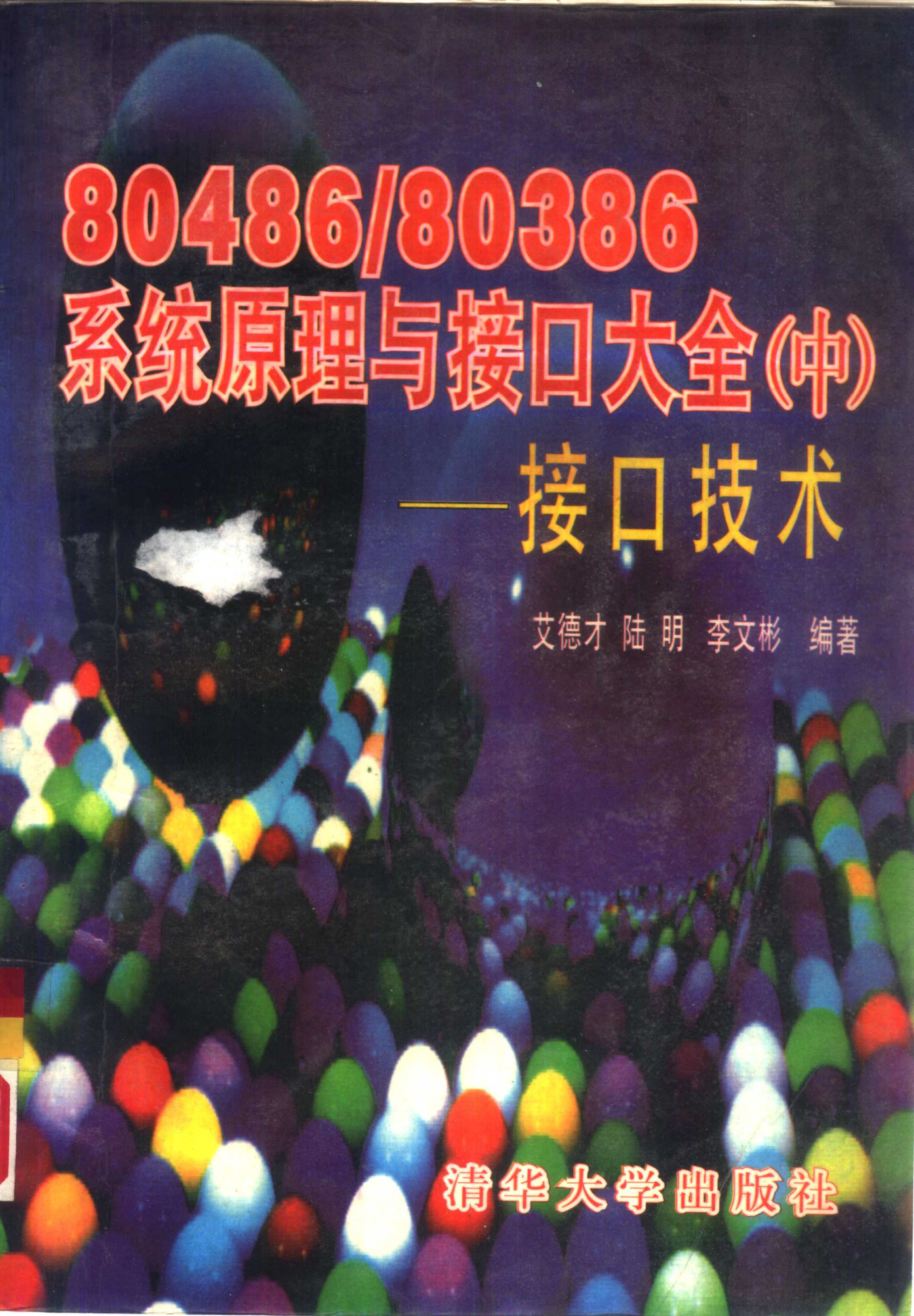




80486/80386 系统原理与接口大全(中) ——接口技术

艾德才 陆明 李文彬 编著

清华大学出版社

80486/80386
系统原理与接口大全(中)
——接口技术

艾德才 陆 明 李文彬 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

本系列书由上、中、下三册组成,分别为 80386 系统原理、接口技术、80486 系统原理。

本书为中册,系统地介绍了目前常用的与用户密切相关的各种外围设备及其接口,包括键盘、显示器、MDA、CGA、MCGA、EGA、VGA、8514/A 等原理和功能,还介绍了常用的软磁盘、硬磁盘、打印机以及 Multibus I、Multibus II、微通道三种总线,并讨论了它们与高级语言、DOS 的接口技术。

本书内容全面、实用,是 80386、80486 接口技术方面的工具书。

本书可供计算机用户用作培训教材,也可供高校师生阅读参考,并可供计算机应用人员作为工具书使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

80486/80386 系统原理与接口大全(中): 接口技术/艾德才等编著. —北京:清华大学出版社,1995

ISBN 7-302-01859-6

I. 80… II. 艾… III. ①微型计算机,80486-接口-技术 ②微型计算机,80386-接口-技术
IV. TP364

中国版本图书馆 CIP 数据核字(95)第 06564 号

出 版 者: 清华大学出版社(北京清华大学校内,邮编 100084)

责任编辑: 焦金生

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 30.75 字数: 725 千字

版 次: 1995 年 9 月第 1 版 1995 年 9 月第 1 次印刷

书 号: ISBN 7-302-01859-6/TP·839

印 数: 0001—8000

定 价: 28.00 元

前言

90年代是微型计算机技术空前发展的10年。在此10年,国内外计算机专家都预测到90年代将是32位微型计算机时代。32位微型计算机是当今微型机的杰出代表。不论是速度还是性能,32位微型计算机系统都超过了70年代末期、80年代初期中小型计算机水平。因为32位微型机已显露出前辈机所没有的机器视觉、人工智能和声音识别的特征。所以它在专家系统、机器人以及控制系统、工程工作站、办公室自动化、事务处理、科学计算和工程计算、人工智能、过程控制、软件开发、CAD/CAM、公共服务、教育和训练等各方面的应用潜力已初见端倪。在本世纪内,32位微型计算机将会占领整个微型机市场。由于80386、80486是当代32位微处理机中的杰出代表之一,微型计算机的主流产品——膝上机的CPU将会全部采用80386、80486甚至是80586。为紧紧盯住90年代微型计算机的潮流,紧紧盯住国际微型计算机潮流,我们不失时机地编写本书,奉献给急需了解80386、80486以及微型计算机接口的广大用户和科技人员。

本书共分三册。上册部分描述的是80386系统原理。围绕80386系统原理,主要介绍了80386CPU、存储管理方式、数值协同处理器80387、高性能的DMA控制器82380、高速缓冲存储器Cache及其控制器82385、图形协同处理器82786等外围芯片的体系结构和工作原理等。同时也列专题讨论了与其系列机兼容的有关问题。

中册部分描述的是与我们用户密切相关的微处理机外围设备及其接口。其中包括我们用户须臾离不开的键盘、显示器、打印机、磁盘等看得见摸得着的外围设备。同时也对支持显示器工作的MDA、CGA、MCGA、EGA、VGA、8514/A等适配器以及各种视频服务器原理和功能给以剖析解释。对80386、80486常采用的Multibus I、Multibus II总线以及IBM的最新专利技术微通道都作了详尽剖析。与高级语言接口、与DOS接口等也给以适当说明。

下册介绍80486系统。该册介绍别具特色,因为80486是由提高了效率的80386微处理机、增强了性能的80387数值协同处理器、一个完整的Cache及其控制器组合而成,根据其特征,下册着意对80486有别于80386的部分给以论述,象对80486CPU、流水线操作原理、支持部件等给以独到论述。对80486特有的片内Cache、二级Cache都给以详尽论述,对有别于80386的存储管理、总线、自测试等方面也进行了有针对性的论述。也就是说,对80486的论述就不再包括与80386相同的部分。

在编写本书过程中力求全面具体,丰富实用,力求使其成为一本有关80386、80486及其接口的微型计算机系统大全和工具书。

在本书编写工作中,得到刘晓月、陈毓弘、苗君秋、胡琳、周士松、孙丙国、刘秀云、程家莲等同志的帮助,在此深表谢意。

由于编者水平所限,时间仓促,新技术新词汇难于仔细推敲,书中不足和谬误在所难免,敬请计算机界老前辈、同仁和读者不吝指正。

编 者

1994 年 10 月于天津大学

目 录

第 14 章 键盘	1
14.1 键盘概述	1
14.2 扫描码	3
14.3 中断 9H	4
14.4 中断 16H	6
14.5 举例	9
第 15 章 显示器	11
15.1 视频基础	12
15.2 视频方式	17
15.3 CRT 控制部件	20
15.4 分辨率	38
第 16 章 MDA、CGA 和 MCGA	43
16.1 单色显示适配器(MDA)	43
16.2 彩色图形适配器(CGA)	46
16.3 文本方式	50
16.4 图形方式	51
16.5 多色图形阵列(MCGA)	54
第 17 章 EGA	62
17.1 EGA 概述	62
17.2 EGA 寄存器的结构	63
17.3 文本方式	96
17.4 图形方式	97
17.5 属性控制	103
17.6 属性控制器的程序设计	104
17.7 RAM 字符发生器	106
17.8 EGA64 介绍	107
第 18 章 VGA	108
18.1 VGA 结构	109
18.2 VGA 寄存器	113
18.3 视频方式	120
18.4 图形控制器	124

18.5	属性控制	127
18.6	字符发生器	131
18.7	视频 DAC	131
18.8	VGA 子系统	134
第 19 章	视频服务程序	139
19.1	视频服务程序概述	139
19.2	置视频方式	141
19.3	光标控制	144
19.4	读光笔位置	146
19.5	选择现役显示页	147
19.6	显示屏幕滚动	148
19.7	读写字符	149
19.8	置颜色调色板	152
19.9	读/写像素	153
19.10	调色板寄存器	156
19.11	颜色寄存器	162
19.12	字符生成	167
19.13	替换选择	179
19.14	写字符串	185
19.15	VGA 返回条件	187
19.16	总结	189
第 20 章	8514/A 适配器	190
20.1	前言	190
20.2	图形加速器	192
20.3	角色	193
20.4	能力	195
20.5	高级功能接口	196
20.6	结构	199
20.7	算法	201
20.8	颜色	205
第 21 章	EGA 程序设计举例	209
21.1	光栅显示器	210
21.2	一个简单的 EGA 库	210
21.3	Bresenham 线段算法	213
21.4	Bresenham 算法的实现	214
21.5	大屏幕上的 EGA 图形	221
第 22 章	磁盘	225
22.1	实际磁盘	225

22.2	软盘	227
22.3	硬盘	239
22.4	磁盘和适配器	256
22.5	磁盘格式化	258
22.6	INT 13H 的服务程序功能	259
第 23 章	打印机	262
23.1	并行端口信号	262
23.2	并行端口硬件	263
23.3	中断 17H	265
23.4	打印屏幕	265
23.5	PS/2 并行端口	266
第 24 章	串行端口	267
24.1	串行通信基本概念	268
24.2	串行通信概述	269
24.3	调制解调器	270
24.4	RS-232 接口标准	273
24.5	串行位流的参数	274
24.6	串行端口	275
24.7	串行端口寄存器	277
24.8	中断 14H	279
24.9	举例	282
第 25 章	MULTIBUS I 总线	284
25.1	前言	284
25.2	Multibus I 技术规格	286
25.3	iLBX 局部总线扩展	307
25.4	iSBX 扩充的输入/输出总线	309
25.5	iSBX 机械参数	318
25.6	Multibus I 应用	320
第 26 章	Multibus II 总线	333
26.1	引言	333
26.2	功能分配	334
26.3	子总线	337
26.4	互联空间	340
26.5	信息传送机构	343
26.6	信息空间	347
26.7	系统总线特点	350
26.8	总线周期	353
26.9	Multibus II 应用	355

26.10 Multibus II 与兼容	367
第 27 章 微通道	371
27.1 微通道特征	371
27.2 微通道构成	378
27.3 传送周期操作	385
27.4 系统配置	394
27.5 总线控制权	403
27.6 微通道接口芯片	406
27.7 16 位微通道	424
27.8 32 位微通道	432
27.9 微通道适配器	437
第 28 章 与高级语言接口	443
28.1 参数传递	443
28.2 Turbo Pascal 直接插入码	444
28.3 Turbo Pascal 接口(版本 3)	445
28.4 新 Turbo Pascal 接口	447
28.5 与 C 接口	448
第 29 章 DOS 接口	451
29.1 前言	451
29.2 DOS 的结构	452
29.3 DOS 的装入	454
29.4 设备驱动程序	456
29.5 磁盘文件	456
29.6 磁盘数据存取	458
29.7 引导记录	459
29.8 根目录	460
29.9 文件分配表	462
29.10 子目录	464
第 30 章 DOS 中断	466
30.1 DOS 中断	466
30.2 DOS 功能调用	467
30.3 DOS 程序执行	471
30.4 .EXE 程序文件的结构	474
30.5 .EXE 程序示例	475
30.6 .COM 文件	478
参考文献	481

第14章 键 盘

键盘是微机系统中不可缺少的一个重要组成部分,它接收用户通过按键发出来的信息,并将这些信息传送给系统部件。

对键盘功能进行比较深入的了解后可以发现,在键盘与系统的其它部分之间存在着相当复杂的相互作用,这种相互作用正是本章所要讨论的内容所在。

在 PC 微机系统的键盘上按下 a 键,并不产生可由系统直接使用的字符 a(比如,以 ASCII 码形式表示的字符 a),而是在部分键盘电路和系统部件中引起一系列的连锁反应,这些反应包括键盘部件与系统部件之间的通信以及若干级的转换和筛选。PC 微机系统的键盘之所以不采用直接输出 ASCII 码的方式,而采取这种比较复杂的处理方法,目的在于增加灵活性。

PC 微机系统中,键盘负责把按下的和释放的键的位置报告给相应的处理程序,然后再由系统软件来解释和处理这些报告。键盘本身并不知道,也“看”不见通过各个键写入的信息。本章我们将详细介绍在键盘与等待被激活键的程序之间的信息流。我们的讨论首先从基本的键盘功能及由键盘传送出的信息格式开始,然后再讨论系统部件中几个负责解释、处理键盘信息的软件。此外,还将对两个键盘中断服务程序 INT 9H 和 INT 16H 进行概要性介绍。

14.1 键 盘 概 述

在不同时期的 PC 系列微机上,配有物理上各不相同的键盘。早期的 PC 机和 PC/XT 机,使用的是有 83 个按键的键盘(PC 键盘)。首批 PC/AT 机的键盘(AT 键盘),在 PC 键盘的基础上增加了一个新键 SysRq;并对键盘的布局进行了少量的调整。在此之后生产的所有的机器采用的是有 101 或 102 个按键的增强型键盘,由于按键数量的增加,使得键盘的功能也随之有所扩充。从 AT 键盘开始,所有键盘均为 Num Lock、Caps Lock 和 Scroll Lock 键配备了发光二极管显示器。此外,AT 键盘和增强型键盘还具有硬件级的有限可编程性。

键盘在专用微处理器或微控制器(Intel 80486 或与之类似的微处理器)的控制下可实现多种功能。其中有一组功能就是负责对键盘进行内部测试,当电源加电时就使用这组功能对键盘进行内部测试。另一种功能是负责处理键盘最新可编程能力和部分地可编程性。键盘的主要功能就是报告被按下的那些键的情况,然后由另两个功能组给予相应处理。第一步是观察扫描各个键的位置,然后记下每个键被按下或被释放的操作情况。第二步是将被操作的那些键的情况经由串行通信线路传送给系统部件,如图 14.1 所示。

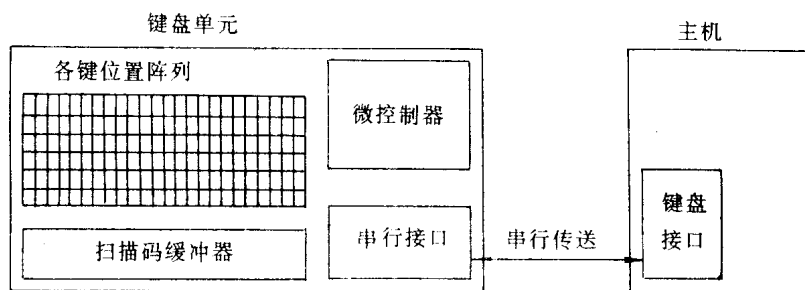


图 14.1 键盘部件与系统部件间的相互作用

从键盘微控制器的角度看,键盘上的键以阵列形式排列(虽然并非所有阵列位置都是实际的键位置),微控制器对键盘上阵列中的各行和列上的键不断地进行快速扫描,以确定有无键的按下或释放。我们可以认为是以下述方式进行扫描的:首先,对第一行中的所有各键位置都逐一地给以探查,然后再对第二行中所有各键的位置给以探查,如此重复操作,直到把整个键盘都探查一遍之后,然后再从第一行开始继续一行行地第二次对键盘进行扫描,第三次对键盘进行扫描,如此往复地进行。这一点与屏幕在显示画面时电子束的扫描十分类似,只不过是键盘扫描的方向与信息流动的方向相反而已。

每次扫描所涉及的某个键的动作信息是:在前一次扫描时还没有按下的那个键现被按下去了,或者是先前按下的一个键现又被释放掉。有关键的动作的信息由微控制器负责实施检测,并且将其以扫描码的形式存放到内部缓冲存储器中。这种内部缓冲存储器规模大约是 16 个字节或 20 个字节,这取决于键盘。

每次对一个键动作的扫描所产生的扫描码都是由一个或多个字节信息构成的。当系统部件不太忙时,再由微控制器以串行方式将其发送给系统部件。最后,微控制器还要负责处理键的重复功能,即当一个键被按下时间超过半秒钟时,就将以每秒 10 个字符的速率重复生成扫描码。

系统部件接收由键盘部件传来的串行信号,并按 8 位一组把它们装配起来,接收电路为每组信号产生一个中断,并告诉系统微处理器键盘出现了动作。在早期的 PC 机中,接收电路由简单的分离元件组成,但从 PC/AT 开始,系统部件中的键盘接口由专用的微处理器操纵,这就可以使键盘与系统部件间能够进行更为复杂的双向通信,同时也可以使系统部件能够向键盘发送操作命令。

通常是由键盘接口产生的中断信号 INT 9H 来启动一个对键盘信息进行预筛选和传送来自键盘的信息的服务程序。例如,该服务程序可截获引起系统复位的 Ctrl-Alt-Del 组合键;将具有变换功能的各种键的激活情况及有关 Shift 变换键状态的信息存放在为键盘保留的数据区内;将通常的键转换成等价的 ASCII 码,并将其存放到应用程序可以访问到的键盘缓冲存储器中(系统存储器的一个组成部分)等等。对 INT 9H 中断服务程序的功能在后面还将进行详细讨论。

将扫描码转换为 ASCII 码信息,是 INT 9H 中断服务程序的重要功能之一。通常,我们总是希望这些信息与实际写在键盘键上的字符相一致。由于在不同国家中,使用的键

盘的布局不同,所以扫描码到 ASCII 码信息的转换应以使用的键盘为准,即 INT 9H 中断服务程序的版本应与键盘布局相匹配。ROM BIOS 中装有美国版本的 INT 9H 服务程序,在其它国家里,这一版本通常被适合这个国家情况的新版本所取代,新版本通常驻留在磁盘上,在系统启动时才将其装入。

对键盘缓冲区中的 ASCII 码(或与 ASCII 码类似的码)信息,应用程序可通过 ROM BIOS 的另一个服务程序 INT 16H 对其进行访问。INT 16H 中断服务程序,除在应用程序与键盘缓冲区的交互作用中提供帮助外,还对具有可编程能力的键盘提供键盘功能控制。关于 INT 16H 服务程序的情况,我们还将以后讨论。

INT 9H 和 INT 16H 服务程序的主要作用是在系统键盘缓冲区与键盘硬件(键盘接口)之间提供一种连接手段。系统键盘缓冲区与键盘接口之间的连接就是由 INT 9H 中断服务程序提供的。而应用程序与键盘缓冲区之间的连接通常是由 INT 16H 中断服务程序提供的。这两个中断服务程序又常被一些应用程序所接管。例如,把一个筛选程序插到这些中断服务程序中的其中一个中断服务程序之上,就是诸多驻留在存储器中实用程序常采用的方法。因为这里的绝大多数信息都是经过筛选程序直接进行传送,而且还能捕获到某些专用功能的组合功能,用来激活这些实用程序。一个程序也可以通过改变指向其自身程序段中中断向量的方法,实现筛选功能,当然使用的键都是调试过的,若发现是一个热键,就启动这个实用程序。否则就把控制传送到旧的中断处理程序。当然,在实施这种类型的键盘功能时,最好还是要高度注意,但在大多情况下还是能运行的。

14.2 扫描码

由上节讨论可知,当按下或释放任一键时,键盘都会向系统部件送出一扫描码,扫描码标明键的位置以及在该键位置上发生的活动(键的按下或释放)。依据键盘的不同,扫描码可以由单字节组成,也可以由一个或多个字节组成。

键盘上动作有两种:一种是打入键操作动作,即把一个键按下;另一种动作是断开键操作动作,即把按下的键松开或者说是释放。

早期 PC 机配置的键盘,按下键的扫描码的字节对应于数值 1—83。这些值是以下述方式分配给各个键的。首先是分配给打字机各键,从最上一行开始自左至右各键分配的扫描码分别是 1—58。然后再给各功能键分配,依从上至下从左到右的顺序分配给它们的扫描码依次是 59—68。最后再给数字袖珍键分配,依与上相同的顺序给各数字键分配的扫描码依次为 69—83,而断开键的扫描码就是那个按下键的扫描码再加上 128。其实,实现起来非常简单,只需将按下键的扫描码的最高有效位置成 1,也就相当于加上了 128,即为断开键的扫描码。键盘上所有各个键的位置都有重复起动功能,每当一个键被按下后保持半秒以上时间,就会重复扫描并重复生成这个键的扫描码,并传送出去,直到松手释放掉这个键,生成了断开键的扫描码方终止重复扫描过程。

AT 键盘是部分可编程的。其扫描码的分配和使用与 PC 键盘也不尽相同。AT 键盘中的每个键位也被安排与一数字相对应,像打字机上的键被分配的数值从上至下分别是:1—28、30—41、43—44、46—55、57—58、61、64;它的功能键被分配的数值从上至下依

次为 65—74；为数字袖珍键盘上的键分配的数值是从键盘中央开始向两边延伸依次为 90—94、96—107。其实，这些数字并不是扫描码，但它们是确定按下键扫描码的基础。AT 键盘中，断开键（释放键）的扫描码在这里由两个字节组成，即由断开前缀 OFOH 再加上按下键位置的扫描码组成。正象 PC 机上配备的键盘那样，所有的键位置都具有重复起动功能，当把一个键按下时，AT 键盘将产生该键位的重复扫描码，至于重复起动的速率和延迟时间都是可编程的（详见后面的 INT 16H 中断服务程序）。

对于具有多种新的键位和许多布局变化的增强型键盘来说，扫描码的安排要更加复杂些。增强型键盘扫描码的分配方式为部分可编程的，可以选择与 PC 键盘类似的扫描码编码方式，也可以选择与 AT 键盘基本相同的处理方法。增强型键盘中，对除 Pause 键外的所有键都是可重复起动的（即产生按下键的重复扫描码），而且，有些扫描码，特别是袖珍键盘上新光标键的代码，要由 Shift 键及 Num Lock 键的状态来决定。

本节所讨论的按下键或释放键时所产生的扫描码，主要由 INT 9H 中断服务程序在将键的活动动作转换为信息的过程中使用。

14.3 中 断 9H

如前所述，每当系统部件中的键盘接口在接收到由键盘部件以串行方式发来的一个整字节后，将产生一个 INT 9H 中断。INT 9H 中断为硬件中断，它负责处理接收的扫描码（参阅图 14.2）。首先，INT 9H 中断服务程序从输入/输出端口（即端口 60H）读入扫描码字节信息，然后根据读入的实际的扫描码情况决定下一步的动作（见图 14.2）。

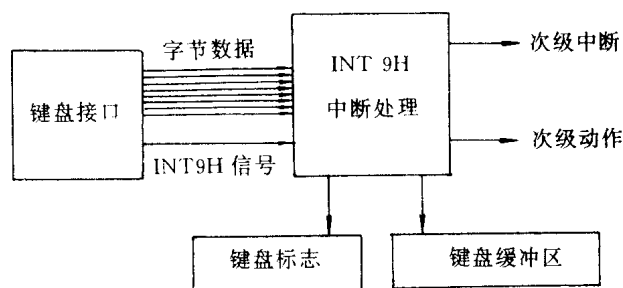


图 14.2 中断 9H 处理

如果读入的扫描码为专用键或像 Print Screen 或 Ctrl-Alt-Del 这样的组合键，就要执行专用键或组合键所具有的功能，INT 9H 中断服务程序将产生一个次级中断来完成这些动作。例如，为 Print Screen 产生 5H 中断；为 Ctrl-Break 产生 1BH 中断，等等。对系统复位组合键，Ctrl-Alt-Del，INT 9H 中断服务程序将产生另外一些次级动作。当然，Ctrl-Alt-Del 引起的系统复位工作只有在 CPU 允许并接受中断后才能进行。

键盘上 Shift 变换键的活动情况被记录在一个称之为键盘标志(keyboard flag)的内部数据区里，如图 14.2 所示，在键盘标志区中，含有各个 Shift 变换键的状态信息（详见表 14.1）。

表 14.1 键盘标志(控制)字节编码

位	意义(40:17H)	意义(40:18H)
7	封锁插入	按下 In 键
6	封锁 Caps Lock 键	按下 Caps Lock 键
5	封锁 Num Lock 键	按下 Num Lock 键
4	封锁 Scroll Lock 键	按下 Scroll Lock 键
3	按下 Alt 键	封锁暂停
2	按下 Ctrl 键	按下 Sys Rq 键*
1	按下左边的 Shift 键	按下左边的 Alt 键*
0	按下右边的 Shift 键	按下左边的 Ctrl 键*

对读入的普通键的扫描码,INT9H 中断服务程序将它们转换为相应的 ASCII 码,并把这些 ASCII 码放入一个可由 DOS 和其它应用程序访问到的另一个专用数据区——ROM BIOS 键盘缓冲区中(keyboard buffer)

下面列出 ROM BIOS 在完成键盘控制时所用到的数据区:

- 键盘标志(双字节,地址在 40:17H—18H)
- 交替袖珍键盘输入缓冲区(单字节,地址在 40:19H)
- 键盘缓冲区头指针(单字,地址在 40:1AH)
- 键盘缓冲区尾指针(单字,地址在 40:1CH)
- 键盘缓冲区(32 个字节,地址在 40:1EH—3DH)

表 14.1 中还清清楚楚地给出了由双字节控制的键盘码,但凡是标有 * 号的那些码对所有键盘都不合法。

在按着 Alt 键不放的情况下,在计算由数字袖珍键盘上的数字键直接输入的 ASCII 码值时,交替袖珍键盘输入缓冲区的作用是,暂时保存输入的 ASCII 码的中间值。

键盘缓冲区是由从 40:1EH 始到 40:3DH 止的 32 个字节连续地址组成,它还配有两个字指针,其地址分别为 40:1AH 和 40:1CH。键盘缓冲区是以循环队列方式组织起来的,用两个缓冲区指针分别指示着现行缓冲区内容的开始与结束位置。键盘缓冲区及其指针如图 14.3 所示。

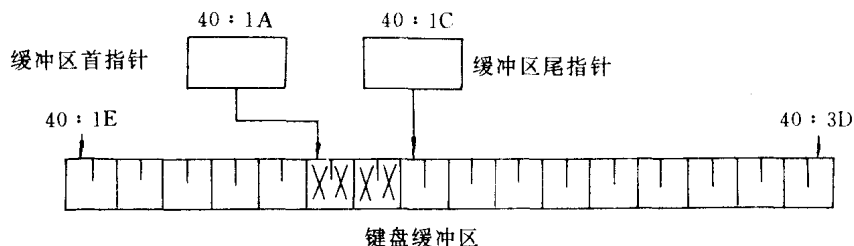


图 14.3 键盘缓冲区

通常是把新信息以两个字节信息的形式存放到队列中,所用方法是把这两个字节信

息写到由尾指针指示的存储单元内,然后修改尾指针。而欲从队列中移出信息(要用到 INT16H,参阅下面的例子)则要由缓冲区首指针指示出要从队列中移出的信息的地址。待信息移出后,缓冲区中的首指针内容就要增值 2。首指针和尾指针都具有在 32 个字区结束时自动环绕回缓冲区始端的功能。事实上,这两个指针中保存的是针对基地址为 40H 的 ROM BIOS 数据区段的偏移量。

INT 9H 中断服务程序的主要功能是,解释由键盘发出的扫描码流同时将这些扫描转换成 ASCII 码那样的格式。当然,在此转换过程中由于使用了各种内部键盘标记,所以还应把各种 Shift(变换键)的状态考虑进去。转换的结果是把按下的每一个键解释成两字节长的类 ASCII 码形式保存于键盘缓冲区。

对那些可解释为 ASCII 码的按键,转换结果中第一个字节为 ASCII 码值,第二个字节为一伪扫描码(在早期的 PC 键盘中,伪扫描码实际上是与键相对应的扫描码)。不可解释为 ASCII 码的其它键,其第一个字节为 00H,第二个字节为一标识被按下的那个键的扩充码。

在增强型键盘上,有些处于不同位置上的键具有相同的功能(如重复的光标键),实际上,这些键经 INT 9H 中断程序解释后,与键盘缓冲区中的解释内容是不一样的。例如,老的光标键经转换后,是带有首字节为 00H 的扩充码,而新增加的光标键经转换后是首字节为 0E0H 的扩充码。为使功能相同的按下键在应用程序面前以相同的形式出现,需要增加一级额外的转换,该级转换一般加在 INT 16H 中断服务程序中。

利用 DEBUG 程序,通过察看 ROM BIOS 数据区的前面部分,可很容易地获得键盘缓冲区中的内容。使用 D40:0 命令并结合按下的键,可将键盘缓冲区的活动情况显示出来。

INT 9H 中断服务程序通常是在没有应用程序的直接干预下工作的。每当键盘上的一个键被激活后,INT 9H 中断程序将更改相关的键触发器或将一扫描码及与它等价的 ASCII(或扩充码)放到键盘缓冲区内。需说明的一点是,如果发出的是喇叭声和按下的键没起作用,这时提供的缓冲区就不满。与 INT 9H 中断程序相反,下面将要讨论的 INT 16H 中断服务程序是一个为应用程序访问键盘数据区提供服务的程序。

14.4 中 断 16H

对 ROM BIOS 的其它服务程序来说,INT 16H 中断服务程序实际上是一组功能块的集合。可通过在 AH(以及 AL)寄存器中放入一功能块编号(服务号)来选择所需的服务,并用在 AL 及其它寄存器中放入适当附加参数的办法调用。表 14.2 列出了 INT 16H 中断服务程序提供的服务功能。

表中,前三项服务功能(即 AH=00, AH=01 和 AH=02)为基本的键盘服务功能(basic keyboard service function),所有的 ROM BIOS 版本均可使用。AH=03 服务功能可在 PC/AT 类计算机(即 BIOS 时间在 1985 年以后的 PC/AT 机及 PC 机 286 机)和所有的 PS/2 机上实现。而 AH=05H 及 AH=10H—12H 的服务功能可在所有使用增强型键盘的机器上使用。下面对各类服务功能进行简单介绍,并说明调用参数及调用规则。

表 14.2 在 INT 16H 中断服务程序下的有效功能

服务号	功 能
00H	键盘读
01H	敲击键的状态
02H	shift 键的状态
03H	置自动重发的速率
04H	保留
05H	键盘写
06H—0FH	保留
10H	扩展键盘的读
11H	扩展的键盘敲击键状态
12H	扩展的 shift 键状态
13H—FFH	保留

14.4.1 基本键盘服务功能

首先列出基本键盘服务功能的作用：

输 入	作 用
AH=00H	等到键盘缓冲区中有按键码后,读缓冲区中内容。 AL=ASCII 码(或 00H) AH=扫描码(伪扫描码)
AH=01H	返回按键码状态 ZF=0/1(1 表示缓冲区中无按键码,0 表示缓冲区中已有按键码) AL=ASCII 码(或 00H) AH=扫描码(伪扫描码)
AH=02H	返回<Shift>变换键状态 AL=<Shift>变换键状态

子服务功能 AH=00H 从键盘缓冲区中传送出标识按下键的两字节扫描码信息(别忘了此时需修改缓冲区首指针代码),如键盘缓冲区中无按下键的扫描码,它将等待到有键按下为止。所以,从键盘缓冲区中读取信息的一般步骤是:首先利用 AH=01H 功能查看缓冲区中是否有按下键的扫描码,然后再用 AH=00H 服务功能读取数据。也就是说,从缓冲区中传送出数据。

AL 寄存器中报告的 shift 交换键状态的格式(来自子服务功能 AH=02H)与位于地址为 40:17H 中的第一个键盘控制字节相同,参阅表 14.1 所示内容。

如前所述,增强型键盘及支持它的 BIOS 把具有相同功能的不同键转换成不同的数据项存放在键盘缓冲区,说明这个问题的最好的例子就是二套光标控制键集。子服务功能 AH=00H 和 AH=01H 在把它们从缓冲区取出时,将对其进行进一步转换,即将它们转

换成(扩充的)ASCII 码。为区分具有相同功能的不同键,可利用扩充的键盘服务功能 AH=10H 和 AH=11H(详见后面的讨论)。

14.4.2 置自动重发速率

在 AT 键盘和增强型键盘中,自动重发(即按下键后发送重复的扫描码)速率是可编程的,这项功能得到了硬件支持。通过 INT 16H 中断服务程序可实现这一功能。实践证明这一功能在最近使用的 BIOS 版本上也是可行的有效的(但在早期的 PC/AT 上的 BIOS 版本上则不能使用)。下表列出有关接口的情况。

输 入	作 用
AH=03H AL=05H	置自动重发及自动延迟
BL=自动重发速率	
BH=自动重发延迟	

表中,自动重发速率(BL)的有效值为 00H—1FH,在此非线性编码体系中,00H 对应于每秒重发 30 次(这种重发速度是相当快的),而 1FH 则对应于每秒重发 2 次(重发速度非常慢)。自动重发延迟(在 BH 中)系指从开始按键到第一次发出重复码之间的时间间隔,其有效值的范围为 00H—03H,分别对应于 250ms、500ms、700ms 和 1000ms(ms 为毫秒)。

14.4.3 键盘写

利用子服务功能 AH=05H,可对 BIOS 键盘缓冲区编程,这就意味着我们可以在程序中对键盘输入进行仿真。

输 入	作 用
AH=05H	将(ASCII 码,扫描码)对送入键盘缓冲区
CL=ASCII 码值	
CH=扫描码	AL=00H/01H,00H 表示传送成功,01H 表示缓冲区已满

子服务功能 AH=05H 还可在一个简单的控制中使用,以证实扩展键盘服务功能是可用的。向键盘缓冲区中写入一随后可取出的广义值(如 0FFFFH),这样就可以将 BIOS 中的扩展功能建立起来。

14.4.4 扩展键盘服务功能

子服务功能 AH=10H 和 AH=11H 可返回有关具有相同功能的不同键位置的全部信息,而子服务功能 AH=12H 可返回一个关于扩展变换键(extended shift key)状态的报告,除以上两点外,子服务功能 AH=10H,AH=11H 和 AH=12H 与前面介绍过的三个基本键盘服务功能类似。