

专 辑

IBM-PC 奥秘

皮特·诺尔顿 著

《计算机技术》副刊编辑部

目 录

1. 使你成为个人计算机奇才

- 1.1 本书的主要内容——如何使你成为个人计算机奇才..... (1)
- 1.2 我们将讨论什么——本书内容概述..... (1)
- 1.3 需要什么手段才能从本书获得最大的教益..... (2)
- 1.4 程序、程序设计语言和软盘程序包..... (2)
- 1.5 三个重叠的范畴..... (4)
- 1.6 资料的来源..... (4)
- 1.7 对表示法所作的几点说明..... (5)

程序列表1.1演示所有的显示屏幕字符 (BASIC)

2. 硬件结构..... (9)

- 2.1 支持计算机工作的关键部件..... (9)
- 2.2 其它部件概览..... (10)
- 2.3 其它那些片子干什么呢..... (12)
- 2.4 扩充槽是如何工作的..... (12)
- 2.5 需要掌握的硬件的其它一些情况..... (13)
- 2.6 再谈三个范畴..... (14)
- 2.7 附加设备的重要性——若干种不同的IBM/PC..... (15)

3. IBM/PC是怎样思维和记忆的..... (18)

- 3.1 存贮器，第一部分：它是什么，如何把它读出来..... (18)
- 3.2 存贮器，第二部分：你的存贮地址是什么样子的..... (20)
- 3.3 PC怎样组织它的存贮器..... (23)
- 3.4 高速暂存存贮器：寄存器..... (26)
- 3.5 提醒你注意的信号：中断..... (27)
- 3.6 你的文件夹：堆栈..... (30)
- 3.7 通过端口对外联系..... (32)

程序列表3.1 查寻工作存贮器 (BASIC)

程序列表3.2 用 Beep 和 Warble (Assembler) 演示中断情况

程序列表3.3 查寻各工作端口 (Pascal)

程序列表3.4 读出端口的情况 (Assembler)

4. 基本操作系统DOS..... (39)

- 4.1 为计算机的正常运行，操作系统应做些什么..... (39)

4.2 概述六部分的工作.....	(40)
4.3 最核心的部分: ROM-BIOS.....	(41)
4.4 起动加载: 引导记录.....	(42)
4.5 完成与设备有关的工作: IBMBIO.COM.....	(43)
4.6 DOS的心脏: IBMDOS.COM.....	(44)
4.7 命令: COMMAND.COM与内部命令.....	(45)
4.8 所有其它部分: 外部命令.....	(47)
4.9 DOS的服务项目——怎样使用.....	(47)
5. 软盘的内部情况.....	(54)
5.1 软盘的结构.....	(54)
5.2 存贮格式.....	(55)
5.3 各类软盘, 以及付本保护的秘诀.....	(57)
5.4 软盘驱动器——Tandon标准.....	(58)
5.5 文件贮存策略.....	(59)
5.6 目录的细节.....	(60)
5.7 文件位置分配表 (FAT) 的细节.....	(63)
5.8 文件的空间位置分配.....	(67)
5.9 文本文件格式.....	(69)
5.10数据记录格式.....	(71)
5.11程序文件格式.....	(72)
程序列表5.1 目录译码 (Pascal)	
5.2 FAT译码 (Pascal)	
5.3 目录和FAT的处理 (Pascal)	
6. 访问ROM.....	(86)
6.1 ROM及其使用.....	(86)
6.2 了解ROM的内情, 部分1: DEBUG.....	(87)
6.3 了解ROM的内情, 部分2: 通过反汇编来进行.....	(90)
6.4 了解ROM的内情, 部分3: BASIC 译码.....	(92)
6.5 BIOS的两种版本.....	(94)
6.6 访问约定.....	(97)
6.7 三个奇异中断.....	(97)
程序列表6.1 检查ROM的发表标志 (Pascal)	
7. 访问软盘.....	(100)
7.1 什么是可利用的——软盘访问的三个级.....	(100)
7.2 ROM—BIOS软盘服务.....	(101)
7.3 软盘参数和如何实现付本保护.....	(103)

程序列表7.1控制软盘参数 (Pascal)

8. 视频终端访问, 第一部分——字符	(106)
8.1 几种视频显示屏	(106)
8.2 视频显示基本原理	(107)
8.3 存贮变换指南	(108)
8.4 彩色图形显示器上的页	(110)
8.5 显示特征	(112)
8.6 色彩的详细说明	(114)
8.7 显示信息的简易方法	(115)
8.8 光标——伪造与控制	(116)
8.9 直接的做法	(117)
8.10 字符图形	(118)
8.11 视频显示的ROM控制	(118)
程序列表8.1 演示所有的色彩和特性 (BASIC)	
程序列表8.2 产生屏幕显示 (Pascal)	
6. 视频终端访问, 第二部分——图形	(126)
9.1 图形显示的基本原理	(126)
9.2 象素	(127)
9.3 象素的变换	(128)
9.4 在图形方式下的字符	(130)
程序列表 9.1 产生图形 (Pascal)	(133)
10. 键盘	(136)
10.1 键盘的基本原理	(136)
10.2 解释键击的含义	(137)
10.3 键字符	(141)
10.4 键盘的BIOS服务	(143)
10.5 能使用键盘的语言	(144)
程序列表10.1演示键盘状况 (BASIC)	
11. 其它特点——通信、扬声器等	(146)
11.1 异步通信转接器	(146)
11.2 打印机转接器	(149)
11.3 盒式磁带机接口	(150)
11.4 其它ROM—BIOS服务	(151)
11.5 使扬声器发出声音	(154)
程序列表11.1 用定时器产生声音 (汇编语言)	

附录 1：计算机基本术语汇编 (157)

附录 2：关于Pascal的简要说明 (169)

附录 3：Pascal与汇编语言的连接 (176)

 A3.1 Pascal一方 (176)

 A3.2 汇编语言一方 (178)

附录 4：奇异字符 (182)

附录 5：附随的软盘程序包 (188)

附录 6：DOS 2.00上关于XT、IBM固定磁盘等的信息 (192)

1 使你成为个人计算机奇才

1.1 本书的主要内容——如何使你成为个人计算机奇才

本书是论述IBM个人计算机所能创造的奇迹的。

当IBM推出它的个人计算机(PC)时,它为个人计算机的质量和性能确定了一条要求很高的新标准。凡是对个人计算机已经有所了解的人都会发现IBM/PC的功能比他们以前所看到的其他PC的功能都强。而对那些常常把个人计算机看作是一种小孩玩具的人,看了IBM/PC以后也开始认识到计算机实际上究竟是干什么的——它是脑力劳动者的首要工具。

计算机入门书有许多,然而本书不在此之列。本书是为那些已经具有计算机初步知识而想进一步揭示IBM个人计算机的实际性能的人编写的。本书也适合那些想了解个人计算机的神奇之处,揭示其内在奥秘的读者阅读。IBM/PC能玩许多的“法力”,我们把所有这些“法力”都奉献给读者。不管你是一位经验丰富的计算机专门家,还是一位刚刚涉猎计算机技术的新手,只要你想要进一步透彻了解IBM/PC是怎样工作,和可以让它完成什么样的工作的,这本书都适合你阅读。

这既是一本深造用书也是一本向你提供基本知识的书。凡是想要研读IBM/PC的中级和高级技术资料的人,这本书正是为你准备的。我们将帮助你了解IBM/PC是怎样一步步工作的,还帮你了解它的全部潜在能力。还将有许多有关如何给IBM/PC设计程序的指导和提示。

但它不是一本论述PC程序设计技巧的书——那是属于设计程序书籍的内容;本书绝大部分内容是涉及IBM/PC的性能的。

在许多人的眼里,那些能干的计算机技术人员总是被看成一些奇异的魔术师,他们知道如何使那些神奇的机器唱歌和跳舞。有了这本书,你也会成为一位那样的奇才了。

1.2 我们将讨论什么——本书内容概述

本书大部分是讨论程序,即讨论软件,因为这是计算机的绝大部分奥秘之所在。但是,我们也需要看看构成IBM个人计算机的硬件——第二章将讨论这方面的内容。

我们将在第三章讨论IBM/PC是怎样思维的,重温一下PC的神经中枢——微处理机的基本原理。

第四章略述了PC的操作系统——PC-DOS,并简单介绍了DOS是怎样工作的。第四章还介绍一些可用来理解DOS特性的工具程序。

然后在第五章,我们将讲解软盘的内部构造以及数据是怎样存储在其中的。用程序实例说明怎样译出软盘的内部信息。

第六章介绍IBM/PC内部只读存储器中的程序,同时还告诉你怎样去探查了解只读存储器的内部情况。这一章是为七至十一章一个题目一个题目地讨论只读存储器中存放的例行服

务程序作准备的。其中的每一章都附有若干程序，让你通过这些程序来掌握那部分计算机的性能。

第七章告诉你如何访问软盘控制程序。第八章和第九章论述显示屏功能——字符显示方式和图形显示方式。第十章介绍键盘，第十一章综述其余各方面的内容。

本书的几个附录给出一个带解释的计算机技术词汇表，扼要介绍Pascal语言，并讨论怎样顺利地将汇编语言程序和Pascal语言程序或其他高级语言程序连接起来。

1.3 需要什么手段才能从本书获得最大的教益

如果你仅仅想学习许多有关IBM个人计算机的知识，那末你学习这本书就够了。但是，如果你还想从本书学到的知识运用到实践中去的话，你将需要学更多的东西；作为开始则先学IBM/PC。

如果你想使用本书所描述的程序，你得有一台配有64K内存和一个软盘驱动器的IBM/PC。这里所说的程序既可以用黑白显示装置也可以用彩色图形显示装置。还要有DOS(磁盘操作系统)操作系统，以及与它配合的各种工具软件，例如DEBUG(调试)程序。随便哪一种版本的DOS都可以用：原始版本1.00，临时用的非正式版本1.05以及改进的版本1.10或DOS2.00。

如果你想充分利用IBM/PC提供的所有特性，你还需要有随本书带去的单独一套软盘程序包。本书列举了利用PC各种特性的工具软件，但是，如果你想把这许多程序都用上，那是不切实际的，除非你有一个软盘付本。附录5列出该程序包的全部内容。

你不需要通过IBM宏汇编来使用本书提供的汇编语言访问例行程序。因为所有那些程序都已变换成现成可用的目标模块并存入本书所附带的软盘程序包里了。但是，如果你想修改这些汇编程序，使它们适应于你自己的需要，那末，还要有宏汇编语言，并且也要知道怎样使用汇编语言。本书向你陈述的内容中一部分就是简单介绍如何使用宏汇编的。

为能单独地或作为你自己的程序的组成部分来使用本书列出的Pascal程序，你将需要有IBM Pascal编译程序。

最后，你也许希望得到Norton实用程序的付本。这些程序包括修理损坏的软盘的程序(FileFix)，恢复被抹掉的文件的程序(UnErase)，修改软盘扇区中的内容的程序(SecMod)和管理隐文件的程序(FileHide和BatHide)，安排文件数据项的程序(DiskOpt和FileSort)，控制显示屏的程序(Reverse, ScrAtr和Clear)；再加上若干其它有用的实用程序。

1.4 程序、程序设计语言和软盘程序包

本书将给出大量的程序，而且我们将在一开始就给出一种有趣的小玩具程序；该程序在表1.1中给出。该BASIC程序可以在IBM/PC的显示终端上将256个字符码一一显示出来。令人吃惊的是，有如此多的PC用户从来没有看到这全部的256个字符，而本程序则可以把它们全部显示给你看。将所有这些字符一起显示出来对你是有好处的，它可以帮助你为了达到某种特殊的效果而从中选取你所需要的字符。该程序将这些字符在萤光屏上用网格形式显示出来，每一排有16个字符。如果你想要知道某个字符的索引号，即按CHR\$功能查找BASIC语言所指的字符，则第一排显示出CHR\$(0)到CHR\$(15)，第二排显示出CHR\$(16)

到CHR\$(31)。每个字符的16进制编码由行和列标志来指明。

本书将采用三种程序设计语言——BASIC, Pascal和汇编语言。交给用户的每一个IBM/PC都带有BASIC语言, 因此在实际工作中我经常使用BASIC语言。但是, BASIC语言不太适合于重大的程序设计, 因此, 本书将重点讨论Pascal语言。

由于Pascal语言具有功能强、简洁、安全可靠等特点, 因此它在个人计算机界的应用已非常普遍。也有几种可与Pascal语言相媲美的语言——特别值得一提的是C语言和Forth语言。不过, 在我们看来, Pascal语言具有两大优点。它比C语言和Forth语言简单易学, 而且最重要的是, IBM从一开始就支持PC采用Pascal语言。既然我们要用某种语言进行通信联络, 因此本书中的大部分程序设计将采用Pascal语言来表示。

也许你还不熟悉Pascal语言。别灰心, 学点语言方面的知识的最容易最省力的办法是读一读本书所给的例子。这些例子是经过认真撰写的, 目的是使它们尽可能的通俗易懂; 另外也是为了逐步地教大家一点Pascal语言知识。此外, 本书的附录2也为你学习Pascal提供简要指导, 向你介绍Pascal的格式并就该语言的最重要部分作些解释。

如果有读者尚未确定用什么语言来为IBM/PC编程序的话, 我诚心诚意向他推荐Pascal, 如果你计划用其他语言为IBM/PC编程, 这里给出的Pascal编程例子对你仍将是十分有用的。这些例子将会告诉你怎样在IBM/PC上做一些事情, 而且其中采用的方法可以很容易地搬到其他语言上去。

IBM/PC的许多功能最强、最有意义的特性只有通过汇编语言才能引用。我们将在本书说明这是些什么样的特性, 是如何工作的。为了解这些特性的全部用途, 我们将给出全套的汇编语言接口例程序; 利用这些程序你就可以利用IBM/PC及其操作系统的所有特性。

本书的第三个附录解释如何和汇编程序进行衔接。该附录详细论述了Pascal程序的连接细节, 而这些细节稍作改动或者不作改动就可以运用到采用标准调用机构的各种语言上去。

为本书编写了不少程序。有些程序是带示范性的, 它们告诉你应如何进行工作; 有些程序则是探索性的, 它们帮助你发现有关PC的底里, 例如隐含在PC内部的存储器的情况; 还有一些是工具程序, 它们告诉你如何访问并控制PC中的许多强有力特性。本书所给出的程序列表基本上把上述这些程序, 特别是把一些有助于读程序和学习程序的内容都包括在内了。有些工具程序, 读起来不见得有意思, 但它们在使用时有价值。本书包含了所有适合于读而且通过键盘输入到计算机中去也有实用价值的程序。其他程序放在随本书一起交给用户的单独的软盘程序包中。

该软盘程序包装有本书列出的全部程序以及许多其他有用的应用工具程序。为了让你知道该程序包中现有的内容, 本书将对这些内容进行介绍, 与此同时还介绍它们的用法。本书的附录5概述了该软盘程序包中的全部程序。

该程序包中的汇编语言程序既给出源格式也给出立即可用的汇编好的目标模块。如果你想按你的要求修改汇编语言例程序, 你可以用源代码来进行修改。但由于这些程序已经汇编好, 你不需要通晓汇编语言, 也不需要IBM宏汇编, 就能充分利用这些功能很强的操作工具。该软盘程序包中的Pascal语言程序也有源格式和编译好的模块形式, 你可以直接使用它们。

该程序包中另外一个程序, 叫DiskLook。DiskLook是一个实用程序, 利用该程序你可以把IBM/PC所用的各个软盘上的所有程序的情况全部弄清楚。它将按名称、日期或

规模大小的次序例出软盘中的文件列表,同时还列出已从盘上删掉的文件名称。DiskLook还将图示该磁盘整个空间的利用情况,指示出每个文件的所在位置,并显示该磁盘任意部分的数据。

1.5 三个重叠的范畴

IBM/PC 的问世开始建立了三个重叠在一起的范畴;虽然在这之前并没有得到广泛的承认,经过好些时间这三个重叠的范畴才逐渐明朗起来;但你必须对这些范畴以及这些范畴与本书的关系有所了解。

虽然你听起来可能会感到糊涂;IBM个人计算机的问世是第一台类IBM/PC的机器的问世,是原型机的问世。但还不仅仅是如此;IBM个人计算机的问世也是MicroSoft操作系统的问世,即MS-DOS;IBM/PC型的MS-DOS被称作PC-DOS或IBM-DOS或只叫DOS。

由此产生了三个人们感兴趣的范畴。首先是对IBM/PC本身;其次是对那些或多或少模仿IBM/PC的计算机;最后是对那些采用某种型号的MS-DOS操作系统的计算机系列感兴趣。

在这些范畴中存在大量的重叠现象,而凡是重点讨论一个范畴的资料都必将涉及到与其他两个有关的大量材料。本书将专门叙述引起人们关注的那三个范畴的创建者——IBM个人计算机;但我们要叙述的内容的大部分对那些对类IBM/PC计算机,以及对采用MS-DOS的计算机系列感兴趣的人也是有益的。

一般,只要有可能,我都将在本书中指明哪些内容适用于其他两个范畴,哪些则不适用于其他两个范畴。

1.6 资料的来源

这样的一本书不可能对IBM/PC的各个方面都进行详细论述。这里列出最重要和最有用的参考资料来源,供你详细了解IBM/PC时参考。

本书所谈的每一方面的内容几乎都取材于这些资料,再加上个人的一些实验和粗浅的见解。比起许多其他的个人计算机来,IBM/PC的资料出处是最公开的,这是由于IBM希望IBM/PC尽可能的适用于软件开发人员和硬件选配人员的要求。作为本书的作者,我并没有接触到太多你们也弄不到手的秘密。我所做的是对大家都能得到的资料加以消化,弄清其中的含义,然后从有限意义的内容中找到具有最广泛用途的资料。

当你需要了解的内容比本书所能提供给你的还要多时,下面这些资料就是你下一步要学的内容:最丰富的资料来源是IBM自己编写的个人计算机技术参考手册。你在该技术参考手册中将会找到下列全部内容以及其他一些材料:

- 个人计算机硬件组织情况概述。即使你不是计算机硬件行家,只要你浏览一下该技术参考手册也能了解IBM/PC内部的运转情况。
- 对IBM/PC的大部分电子线路进行了详细技术说明;这主要是为计算机硬件知识丰富的专家们学习用的,但它也对系统中的各个部件如何在软件控制下进行操作提供非常有益的思路。例如,如果你真想了解彩色图形显示转换器的全部潜在能力,你就应该研究对Motorola6845CRT控制器的论述。
- 全面论述作为PC的不可分割部分的ROM(只读存储器)程序的使用。

- 详细列出（加详细注释）ROM-BIOS（即只读存储器基本输入输出系统）；这些程序可为IBM/PC提供最基本的控制和服务程序设计技术。该程序列表是采用汇编语言，即说，你必须掌握8086的汇编代码才有可能完全理解该表的意思。尽管如此，该ROM-BIOS列表仍不失为一种信息宝库，它介绍适合各个程序的各种服务。即便你不懂汇编语言，只要你掌握了向你提供的各种服务和这些服务的组织安排情况，你就找到了弄懂IBM/PC情况的锁匙。（虽然给出了BIOS服务程序表，却没有给出IBM/PC内装的BASIC语言，即ROM-BASIC表。你可通过DOS的DEBUG命令得到那种语言的程序表，不过该表没有任何注解或说明，因此除非他是一位强有力的解难题行家，否则任何人拿到该表都是没有用处的。）
- 有两张表非常好，它们解说IBM/PC用的全部256个字符编码，并且说明了控制IBM/PC显示终端上文本彩色的那些“特性”字符的功能。

另一本最有用的资料是DOS操作系统使用手册。DOS使用手册中的附录有许多有关软盘的格式，DOS服务程序的各种约定，文件控制块，程序段的前缀等等方面的有用资料。DOS使用手册的内容一般都不如我们的参考资料内容丰富，因为它不提供有关DOS的全面技术说明。使用手册所叙述的内容只提供了非常有益的信息和思路。

你要了解IBM/PC内部的情况，可以去查阅几种有关Intel 8086/8088微处理机的参考书。我已发现两本特别有用的参考书。如果要了解8086微处理机的基本原理以及它是如何工作的，可以阅读Stephen P. Morse (Hayden, 1980) 著的《8086/8088初步》一书；而如果想深入了解该微处理机的详细情况，则可阅读Russell Rector和George Alexy (Osborne/McGrawHill, 1980) 合著的《8086手册》。

要了解Pascal程序设计技术，则可参阅IBM Pascal编译程序手册。这本资料虽不算是一本好的介绍Pascal程序设计技术的资料，但它是了解IBM/PC的IBM Pascal语言具体特性的唯一参考书，要学习Pascal语言的话，可以从许多Pascal入门书籍中（计算机书店有的是这类书）选一本来学。或者，你手头没有任何可以学的东西的话，就可以通读一遍IBM Pascal编译程序手册。对于你为使用IBM Pascal语言所需要知道的所有技术细节，该资料都用举例的方式来详细解释或暗示。我们认为IBM Pascal使用手册的作者在那些一时难以详细说明的内容上向读者很好地提供了必要的思路。（你还可以通过学习本书所给的Pascal举例和阅读附录2中有关Pascal语言的简短辅导材料学到许多Pascal方面的知识。）

至于汇编语言的程序设计，和Pascal语言一样，可以用IBM宏汇编语言手册作详细的参考手册。但不要希望从该IBM使用手册学会机器语言的指令系统。你若要学机器语言的指令系统，请学习上面提到的那两本8086的使用手册。IBM宏汇编使用手册在说明情况和提供必要的有关材料方面做得特别差劲；但在如何用宏汇编语言方面它是一本不可缺少的参考手册。（本书的附录3讨论使Pascal和汇编语言之间互相取得连系的技巧，而且将向你提供从何处着手去写汇编语言的许多有益的思路。）

1.7 对表示法所作的几点说明

有许多种叙述事物的方法，因而计算机中有许多令人糊涂的表示法。而在本书中我们将尽量地使这些表示法简单、明了和一致。如果你发现本书所用的某个术语不懂，请到本书的第一个附录的语汇注释那里去查找这个术语的意思。

下面谈谈我们是怎样称谓事物的。首先，让我们来谈谈数字表示法。

数字一般都采用我们小时候都学过的十进制数目字。必要时也将采用16进制的数字，这时将在该数字上标以“16”的字头。16进制数字是以16为底的数；它对我们讨论计算机很有用处，因为它们是二进制数的方便的速记形式，而二进制又是计算机的基础。十进制数字有十个，它们是0到9；而十六进制数字则有十六个，它们是0到9，接着是A（其值为十）B、C、D、E、最后是F（其值为十五）。本书有十一章（十六进制B）；美国的独立宣言是1776年（十六进制6F0）签署的。

你可能弄不清为什么本书要用十进制和十六进制这两种表示法来表示数，特别是因为在程序设计语言BASIC, Pascal和Assembler中没有任何东西要求采用十六进制。这里有三个原因。一个是，有些数特别是存储器地址单元用16进制表示其含义更明确。另一个是，DOS的工具程序DEBUG只采用十六进制表示法。最后，有关IBM/PC的有些文字资料，特别是技术参考手册，广泛采用十六进制表示法，且往往预先不通知。为了帮助你把本书的数据和其他资料上的数据联系起来，我们往往用十进制和十六进制两种形式来表示一个数。另一个与数字有关的计算机的术语是字母“K”。“K”的准确意义是1024，因此，64K就是64乘1,024，即65,536。K这个术语用得很广泛，因为它是一种表示一个大约等于一千的数的方便的缩写形式；因此，很容易理解，64K约等于64个千。K的值——1024是二进制表示法的一个完整数 2^{10} 。

偶尔，我们要按字符的索引号（其范围是0至255）去称呼字符。为此，我们将采用BASIC语言的表示法；例如，CHR\$(65)，它表示大写字母“A”。Pascal语言表示字母“A”的方式是chr(65)。

下面我们来谈谈二进制位的概念。在表示作为字节和字的组成单位的各个二进制位中，在叫法上总是使人感到糊涂。不同的资料用不同的方法来表示它们，这使我们觉得困难。最常用的二进制位系统有三种，我在这里将讲解这三种二进制表示法。很难说哪一种最好。我们将采用最简单的一种。如果你要大量阅读计算机的文献资料，你就要熟悉这三种表示法。

如果你给构成一个字节的八个二进制位进行布局，其中“最高有效位”为第一位，而“最低有效位”为最后一位，则可以给这八个二进制位分别从第一号编到第八号；你可以把这种表示法叫作“位序”。这就是我们要采用的表示法；这种表示法最简单而且最清楚，但它有一个缺点：它不具体表示什么意义。另外两种方法则注重各二进制位所在位置的数值含义。第二种方法是从右到左（和位序法相反）为各个二进制位编号，起始位为0。这是“位幂”表示法，它表示每一个二进制位相对应的二的幂。这种位幂表示法也许是技术文献资料中最普遍采用的一种方法。第三种方法叫作“位值”表示法。它给每一个二进制位的所在位置标上二的幂值。

下面这张表概述了这三种表示法：

位序法	位幂法	位值法	该位在一个八位字节中的位置
第一位	7	128	10000000
第二位	6	64	01000000
第三位	5	32	00100000
第四位	4	16	00010000
第五位	3	8	00001000

续上

位序法	位幂法	位值法	该位在一个八位字节中的位置
第六位	2	4	00000100
第七位	1	2	00000010
第八位	0	1	00000001

前面我们说过，我们在本书将采用“位序”表示法。另外两种，特别是“位幂”法，你将在别的资料中接触到。“位值”法的主要用途是可以使你明白如何把一个字节的数值与该字节的各个位的情况联系起来。例如，在一个程序中，你通过给某个字节置成值 $128+1$ 就可以确定该字节的第一位和第八位。这三种表示法中的每一种都可从一个8位字节扩展到表示16位的字，因此，有时你也会看到它们这样用。

(陈伯洲 译)

程序列表1.1

```

1000 REM Listing 1.1 -- A program to display all screen characters
1010 REM (C) Copyright 1983, Peter Norton
1020 REM
1030 GOSUB 2000 ' TITLE
1040 GOSUB 3000 ' GET WHICH DISPLAY TYPE, AND SET ADDRESS
1050 GOSUB 2000 ' SET THE TITLE AGAIN
1060 GOSUB 4000 ' BUILD THE SURROUNDING COMMENTS
1070 GOSUB 5000 ' BUILD THE DISPLAY ARRAY
1080 GOSUB 6000 ' FINISH UP AND RETURN TO DOS

2000 REM Title subroutine
2010 KEY OFF : CLS : WIDTH 80
2020 REM
2030 PRINT " Programs for INSIDE THE IBM PERSONAL COMPUTER"
2040 PRINT " (C) Copyright 1983 Peter Norton"
2050 PRINT
2060 PRINT " Program 1-1: Demonstrate all screen characters"
2999 RETURN

3000 REM Subroutine to inquire about display type
3010 PRINT
3020 PRINT "Before we go any further, is this a color-graphics display? ";
3030 GOTO 3060
3040 PRINT
3050 PRINT " (answer Y or N) ";
3060 ANSWER$ = INKEY$
3070 IF LEN(ANSWER$) < 1 THEN 3060
3080 IF LEN(ANSWER$) > 1 THEN 3040
3090 SEGVAL! = 0
3100 IF MID$(ANSWER$,1,1) = "Y" THEN SEGVAL! = &HB00 ' Color segment
3110 IF MID$(ANSWER$,1,1) = "y" THEN SEGVAL! = &HB00 ' Color segment
3120 IF MID$(ANSWER$,1,1) = "N" THEN SEGVAL! = &HB000 ' Monochrome segment
3130 IF MID$(ANSWER$,1,1) = "n" THEN SEGVAL! = &HB000 ' Monochrome segment
3140 IF SEGVAL! = 0 THEN 3040
3150 DEF SEG = SEGVAL!
3999 RETURN

4000 REM subroutine to build the surrounding messages
4010 FOR HEX.DIGIT% = 0 TO 15
4020 LOCATE 6,HEX.DIGIT% * 3 + 14
4030 PRINT HEX$(HEX.DIGIT%)
4040 LOCATE HEX.DIGIT%+8,8
4050 PRINT HEX$(HEX.DIGIT%)
4060 NEXT HEX.DIGIT%
4070 LOCATE ,,0
4999 RETURN

5000 REM subroutine to set the display array
5010 FOR ROW% = 0 TO 15
5020 FOR COL% = 0 TO 15
5030 POKE (ROW% + 7) * 160 + COL% * 6 + 26, COL% + ROW% * 16
5040 NEXT COL%
5050 NEXT ROW%
5999 RETURN

6000 LOCATE 25,1,1
6010 PRINT "Press any key to return to DOS... ";
6020 IF LEN(INKEY$) = 0 THEN 6020 ' wait for a keystroke
6030 CLS
6999 SYSTEM

9999 REM End of program listing 1-1

```

2 硬件结构

这一章我们要谈谈IBM个人计算机的物质方面，即设备和电路以及使这种计算机能够工作的“芯片”方面的情况。物理的“硬件”是和程序的“软件”相对应的。由于本书不是一本专门讨论计算机硬件的书，因此我们将不作十分深入细致的讨论。但是为了帮助你了解IBM/PC是怎样工作的，什么样的附加设备可以和IBM/PC连接，这些附加设备怎样与该PC装置的其他部件联系起来互相协同工作等这些关键的性能，我们将就这些方面进行详细讲解。

一台计算机非常象一辆汽车，是由许多部件组成的。它和汽车一样，你不必要了解各个部件之后才能使用整个机器。然而，如果你要了解整个系统的话，你至少应大致了解这些部件是怎样工作的，这就是我们要在本章讨论的内容。

2.1 支持计算机工作的关键部件

计算机系统的核心是它的处理器，又叫计算机的“大脑”。正是有了该处理器计算机才能执行计算机程序的指令。个人计算机采用微处理器；它的尺寸已缩小到适于放入到一块插入式的电路元件——“芯片”中去。IBM/PC用的微处理器是由Intel公司（该公司几年前领先开发了微处理技术）设计和制造的。

IBM/PC的微处理器型号是Intel8088，它是与它稍有不同的那种微处理器8086的小弟弟。8086和我们采用的微处理器8088执行同样的指令；从程序员的角度来看，它们是一样的（只有从电路设计人员的角度来看，它们有些差别，对此我们将作简要的说明）。有关8086的程序设计的文字资料均适用于IBM/PC采用的微处理器——8088。

区分IBM/PC与以前的个人计算机的一个主要的东西是，IBM/PC的微处理器是一种16位的微处理器。在IBM/PC之前，绝大多数的个人计算机都采用8位的处理器，例如6502（Apple II中所用的），8080以及Z80。8位和16位处理器的不同点并不很明显。最不重要的不同之处是它们的名字：8位的处理器一次一般只能处理8位数据，而16位的处理器一次却能处理16位的数据。这两种处理器能得到相同的结果，因此，这两种处理器在那方面的区别不是很重要的。

真正使16位处理器与8位处理器区别开来的是16位处理器在速度、指令系统（16位操作是该指令系统的一个组成部分）的功能和用途，以及最重要的方面，即存储器的寻址能力等方面所作的大量改进。许多8位的处理器都仅限于采用不超过64K的存储器。这个容量不能使我们充分利用个人计算机。IBM/PC采用的8088微处理器能对1024K进行编址，也就是说存储器的容量可达一百万个字节以上。你可能希望存储器的容量越大越好，不过，一百万字节的存储器完全够你用的了。

所以说，8088微处理器和8位处理器的最大不同就是，存储器的寻址能力对个人计算机要完成的任务不再是一个严重的限制条件。

8088微处理器和它的老大哥8086之间的不同点是什么？它们的内部功能是一样的；它们执行相同的指令，采用相同的数据，执行相同的程序。但是，当它们与各自的外部世界打交

道时，它们就不一样了。8086，象一台16位的计算机那样，每次与支持它的电路交换16位信息，而8088与它周围的设备每次仅交换8位数据。8086和8088的关键不同点是它们的外部数据通路大小不同——8086一次可传送16位的数据，而8088一次传送8位数据。这就使有的人说，8088不是真正的16位机，这种说法有一部分是对的，但仅一部分而已。8088的内部确实采用16位的系统结构，但它并没有采用16位的外部数据通路。

用来给整个系统传送数据的外部数据通路叫作数据总线，有关这种总线的情况我们将在2.3节中详谈。8088采用8位的数据总线。

8086和我们的8088之间的实质性区别有两个方面。第一，当要传送一个以上的数据字节时，8086的传送速度比8088快两倍。这并不意味着，8086完成操作的速度比8088快了两倍；因为，只有一部分时间是该处理器等待与它的外部设备来回传递数据，而且有时它只需要一个8位的字节。不过，当该处理器确实是在等待给它传送大量的数据时，8086处理器就可以减少等待时间，因此它能以较快的速度完成它的操作。

第二个实质性的区别是电路设计和元件选择。设计8位的电路比较容易；现在有许多价格便宜且十分可靠的8位组件供挑选。因此，IBM用8088就可以简化它的PC并降低造价，而在计算速度上只是稍有一点牺牲。

2.2 其它部件概览

要该微处理器工作，则要有许多支持部件。正好象只有发动机并不能使汽车运行一样，一台计算机不仅需要处理器还需要许多其它的支持设备。

一辆汽车尚且有许多元件可与其他牌号的汽车的元件互换，个人计算机就更应该是如此了。IBM/PC只有很少的几个工作部件是定做的；实际上大部分PC，其中首先从Intel 8088微处理器开始，都是由相当标准而现成的部件组装成的。使IBM/PC与众不同的只是把各种部件构造成一个完整的系统所采用的方法不同罢了。电子工业为计算机设计师提供了一个装满了各种标准元件的“工具箱”，设计师的任务是利用他们自己的聪明才智把它们组装成一种有用的系统。

如果把计算机的设计工作描绘成那样，则听起来似乎设计工作没有多少事情可做，而且计算机与计算机之间也没有多大区别。这等于说，一件伟大的作品不过是用在词典中找来的一个个词构成的。不错，个人计算机、包括IBM/PC在内，主要是由标准元件组成的；但其关键因素是把这些元件组织起来的方法。

有三种方法来看IBM/PC的所有元件——按它们所在的位置，按它们所起的作用以及它们彼此联系的方式。我们首先来看它的位置。

从物理上看，IBM/PC的部件分成系统板部件和扩充板部件。所有IBM/PC上有的主要电路元件都在一块叫作系统板的大板上。（Apple计算机用了一个华丽的词，叫作“母板”。）系统板上的零、部件是使计算机不需添加其他选件就能工作的各种必要的零、部件。它包括微处理器、前64K存储器、放在只读存储器片中的机内程序，例如BASIC语言的核心程序。下一节（即“其他的那些芯片干什么呢”）要讨论的部件大部分都在系统板上。图2.1是该系统板的简化图，其中较重要的元件都已标出。

该系统板平放在PC的密闭式机柜的底部。它的长度和机柜的前后长度一样长，宽度大约为机柜宽度的三分之二。如果你打开PC的机柜往里看，就会看到这块系统板躺在机柜的

底部。如果你朝这块板的中后方看，你将会看得到IBM/PC中的那个最大的电路元件——8088微处理器。

系统板右边伸入到软盘驱动器的下面（靠左边的软盘），而系统板的左边上方空间是开放的——可以放扩充板。系统板的左后方拐角处有五个空的插座，叫作扩充槽，用来连接附加到PC上去的各种选用设备。扩充板垂直插入这些扩充槽，在该系统板的上面向上伸延。

这些扩充板（有时又叫插件）可以用来支持附加到IBM/PC上去的各种设备。它们的用途分两大类：增加存储器和增加设备。如果有某个增加的设备能装到一块电路板上，则它正好就放在PC机柜里。如果该设备不适合放在里面——例如显示屏——那末支持该设备的电路部分就放在扩充板上，从PC机柜的背面与外部设备连接。每一个扩充槽都在机柜后部装有一个带盖的口子，以便与外部连接。

该系统板由IBM生产，扩充板可由其他公司生产。系统板上的每一样东西都由IBM来布局和设计；扩充板上可以设计随便什么东西，但应在尺寸、电气连接、热量等方面遵循IBM的基本原则。只要看一下人们大肆宣传的作为IBM/PC附加设备的产品，你就会看到可以插入到扩充板的的东西的品种之多实在令人吃惊。

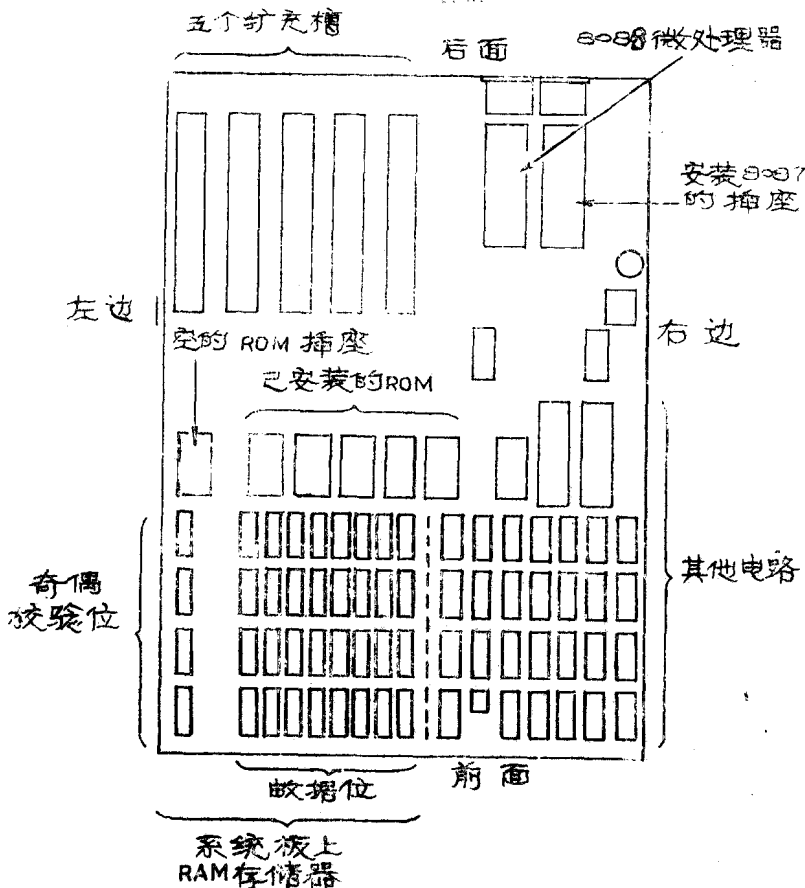


图2.1 IBM/PC的系统板

2.3 其他那些片子干什么呢

对你来说，也许没有多大用处，但了解一下IBM/PC的主要电路元件究竟是干什么的，是有意义的。因此，我们在这一节来谈谈这些元件。当这些元件中某一个元件具有某种意义时（我们将在本书的其他章节中涉及这些意义），我将专门提及它；如果不专门提及，你就可以假定，本节所提的情况和详细说明可以完全不考虑。

基本的定时信号系通过一个8284A时钟发生器芯片提供给系统。这些信号是整个计算机系统用来控制操作长度的。该时钟与一个定时器芯片——8255A-5有关，该芯片用来支持盒式磁带接口和机内喇叭。在第十一章，谈到如何控制喇叭时，我们将讨论怎样为8255A-5编程，以便使喇叭发出声音。

中断是计算机系统操作的最基本部分，我们将在下一章说明它们是怎样工作的。8259A中断控制器芯片用于控制中断的操作。

每当要在PC系统内传递数据，该数据要经过一条公用的数据通路——数据总线。这条通路通到所有的系统部件。“总线”这个概念是计算机设计中把各个部件统一起来的最巧妙的一种设计技巧，计算机的设计师不是设法把所有系统部件所需的各种专门连接电路连接起来，而是把所有的数据传送任务都放在一条公用总线上。数据以及表示该数据的用途的信号一起在该总线上传送。这种设计思想大大简化了计算机的设计，并大大增强了其灵活性。当你要增加新的部件时，不需要进行许多特殊的连接，而只要把这个部件挂到总线上就行了。正如你会想到的那样，该总线上的数据传送有可能会相当混乱，因此，由8288总线控制器芯片来维持该总线的传输秩序。

到目前为止所谈到的部件全都在系统板上。当我们看主要的扩充板时，我们发现一些其他的有趣元件。有两种IBM/PC的显示转接器。一种用来控制IBM的黑白显示，另一种用来控制彩色的图形显示（或者是可以连到彩色图形显示转接器上的简单黑白显示）。虽然这两种显示操作方式不同，而且显示能力也相差很大，但控制它们的主要电路片是相同的。6845显象管控制器是这两种显示转接器的主要部件。

软盘转接器板用了一个软盘控制芯片来控制该磁盘的驱动器。该芯片是NEC PD765 软盘控制器，或与它相似的片子。（如果你研究IBM技术参考手册中软盘控制用的ROM一览表，或者偶尔碰到“NEC”的一些秘密参考文献，这些资料都是讨论该控制器芯片的。）虽然本书不会深入讨论到那样的程度，但可以通过向该控制器发命令，对各种软盘驱动器进行直接控制，而IBM的技术参考手册列出了NEC命令的技术细节。

2.4 扩充槽是如何工作的

IBM/PC的每个外加选件都可插入到其中的一个扩充槽，槽中有62根连接线。这62根线可以把各种信号提供给加接到PC上的设备使用。这62根线并行操作，因此任何扩充板都可插入到这五个扩充槽中的任一个。送到扩充板上的任一个信号被所有这五个扩充槽同时看到，因为它们全都与公共信号线相连。这是公共数据总线思想的扩展；所有扩充板均采用一种公共的62线连接结构，叫作输入/输出通道。

这62根线的具体用法分成四类。首先，其中有8根线用来给扩充板提供各种不同电压的电源。