

计算机实用技术系列丛书(三)

P R O G R A M M E R ' S G U I D E T O

PC&PS/2[®]

系列机视频

高级程序设计技术

VIDEO

Systems

MAXIMUM VIDEO PERFORMANCE

FROM THE EGA, VGA, MCGA,

AND HERCULES GRAPHICS CARD

RICHARD WILTON

THE C GAZETTE
"1989 BOOK OF THE YEAR"
THE C GAZETTE

希望

Microsoft
PRESS

学苑出版社

计算机实用技术系列丛书(三)

PC 和 PS/2 系列机视频高级程序设计技术

—— EGA、VGA、HGC、MCGA 最佳视频功能实现

Richard Wilton	著
闻 钟	译
吴 平	审校

学苑出版社

(京)新登字 151 号

内 容 提 要

本书详细、全面地介绍了 PC、PS/2 系列机的视频模式及高级程序设计技术。书中内容丰富,有大量程序实例,是从事计算机工作人员必备的参考书。

需要本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码 100080,电话 2562329。

版 权 声 明

本书英文版名为《PC&PS/2 VIDEO SYSTEMS》,由 Microsoft 公司下属的 Microsoft 出版社(Microsoft Press)出版。版权归 Microsoft 公司所有。本书中文版由 Microsoft Press 授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

计算机实用技术系列丛书(三)

PC 和 PS/2 系列机视频高级程序设计技术

著 者:Richard Wilton

译 者:闻 钟

审 校:吴 平

责任编辑:甄国宪

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:双青印刷厂

开 本:787×1092 1/16

印 张:29.625 字数:694 千字

印 数:1~5000

版 次:1994 年 6 月北京第 1 版第 1 次

ISBN 7-5077-0777-6/TP.9

本册定价:53.00 元

学苑版图书印、装错误可随时退换

目 录

致谢	(1)
本书介绍	(2)
第一章 IBM 视频硬件与固件	(5)
1.1 IBM PC 和 PS/2 的视频硬件	(5)
1.2 ROM BIOS 界面介绍	(9)
第二章 硬件程序设计	(14)
2.1 IBM PC 和 PS/2 视频子系统的功能部件	(14)
2.2 显示刷新周期	(16)
2.3 CRT 控制器程序设计	(18)
2.4 基本 CRTC 算法	(22)
2.5 CRT 状态寄存器	(24)
2.6 视频模式	(26)
2.7 硬件视频模式控制	(27)
2.8 视频子系统的组合	(36)
第三章 字母数字的模式	(39)
3.1 使用字母数字的模式	(39)
3.2 字母数字数据的表示法	(40)
3.3 属性	(42)
3.4 灰度和	(54)
3.5 边界颜色	(55)
3.6 避免 CGA 雪花	(57)
3.7 使用所有的视频缓冲区	(64)
3.8 光标控制	(66)
第四章 图形模式	(72)
4.1 使用图形模式	(72)
4.2 映射像素到屏幕	(72)
4.3 像素坐标	(77)
4.4 像素显示属性	(85)
第五章 像素程序设计	(95)
5.1 位平面程序设计	(95)
5.2 读一个像素的值	(106)
5.3 设置一个像素的值	(117)
5.4 填充视频缓冲区	(134)

第六章 线	(141)
6.1 有效的画线算法	(141)
6.2 最优化	(145)
6.3 PC 和 PS/2 的性能	(148)
6.4 线属性	(190)
6.5 剪取	(190)
第七章 圆和椭圆	(196)
7.1 圆和像素比例	(196)
7.2 画椭圆算法	(197)
7.3 最优化	(207)
7.4 剪取	(214)
7.5 真正的圆	(215)
第八章 区域填充	(216)
8.1 什么是区域	(216)
8.2 利用水平直线简单填充	(217)
8.3 区域填充的三种算法	(218)
8.4 算法比较	(235)
第九章 图形的文本	(238)
9.1 字符定义表	(238)
9.2 软件字符生成器	(240)
9.3 设计一个软件字符生成器	(241)
9.4 实现一个软件字符生成器	(244)
第十章 字母数字型字符集	(264)
10.1 字符定义表	(264)
10.2 更新字符生成器 RAM	(270)
10.3 使用 RAM 基本字符集	(280)
10.4 改变显示字符矩阵	(287)
10.5 在字母数字模式下的图形窗口	(300)
第十一章 位块和动画制作	(305)
11.1 位块移动	(305)
11.2 像素逐位操作	(320)
11.3 位块覆盖	(323)
11.4 动画制作	(324)
11.5 图形模式光标	(328)
第十二章 视频程序设计的高级技术	(331)
12.1 垂直中断管理器	(331)
12.2 EGA 和 VGA 上的全景	(342)
12.3 位平面分层	(350)
12.4 EGA 和 VGA 分离屏幕	(351)

12.5 光笔接口	(355)
第十三章 在高级语言中的图形子程序	(361)
13.1 链接图形子程序	(361)
13.2 全程数据区	(375)
13.3 分层图形接口	(376)
附录 A 视频 BIOS 概要	(384)
A.1 ROM 视频 BIOS 支持的硬件	(384)
A.2 视频 BIOS 数据区	(386)
A.3 IBM PC 和 PS/2 视频 BIOS 功能	(396)
附录 B 打印屏幕	(437)
B.1 字母数字的模式	(437)
B.2 图形模式	(437)
附录 C 识别视频子系统	(453)
C.1 CGA 及其系列	(453)
C.2 其他视频适配器	(453)
C.3 PS/2 系列	(453)
词汇表	(465)

致 谢

第六、七、八章的资料归功于几个在计算机图形领域倍受推崇的人员所做的大量的独创性工作。在每一章节中,我引用了他们最著名的出版物的内容。如果你对这些章节中的算法描述有兴趣,无论如何你自己要得到最初的出版物并进行研究。

本书如果没有我的家人、朋友、同事的支持鼓励是编写不了的。我要对他们的耐心和支持表示衷心的感谢。我也向 Andy Fischer 和 Charles Petzold 表示感谢,他们二人对本书的部分内容提出了各自见解并提出了正确的批评和建议。

此外,我还特别感谢 Microsoft Press—Claudette Moore, Jeff Hinsch 及其他许多人,他们不辞辛苦的将该书草稿整理为完整的出版物。

本书介绍

：我还清楚地记得那一天，我第一次将一个新的 IBM 增强型图形适配器(EGA)插进 IBM PC 机里。有一个 IBM 新的“增强型”视频硬件是好的，它的分辨率和控制颜色是任何 IBM 早期的 PC 机视频硬件所不具有的，现在我准备编写一些真正清晰的图形应用程序。

遗憾的是，我不能理解如何设计这个新发明的玩意儿，我完全没有技术性文件(直到花了 6 个月时间和 125 美元邮寄费)，我试图分解 EGA 的 ROMBIOS，但研究了 6000 条无注释的机器指令，很快得到比它已回答的更多的问题，我不顾死活地试图猎取一种方法——改变内存的内容定位及匹配寄存器，正如看到的一样将发生什么——但是这就像一辆随意切成碎片的汽车一样，最终不得不停工。

我所缺乏的是详细叙述——硬件设计的概念描述，描述程序设计界面的目录以及所有的关于一些典型的程序设计技巧的源代码例子。选取一个源代码例子将花去许多调试时间，当我试图弄明白如何设计视频适配器时而遭到了挫折。

辛苦的内存试验给予了本书灵感，本书中充满了源代码例子，它用文本形式描述源代码。本书也有许多表和硬件程序设计的扼要描述。简单地讲，本书是我希望在开始设计 PC 视频硬件时所得到的一切。

本书内容

本书第一章节是视频显示环境的一般评述。本书中其他部分描述了 PC 机及 PS/2 视频硬件的一般使用，它亦向你介绍(如果你不准备说术语)了众所周知的 ROM BIOS 视频支持的程序。

接下来的十章内容包含了 IBM 视频程序设计的难题与关键问题。前面一些章节包括了基本法则，包含了硬件体系结构、视频显示模式和你的程序与硬件之间的自然接口。后面的章节基于基本法则来论述许多用于产生文本和图形输出的技术。

本书最后两章给出了低级和高级的视频图形程序设计，第十二章为硬件粗略补充章节——如果你打算用垂直中断工作或转动位面，则该章节的内容你可参阅。最后，第十三章告诉你如何链接视频硬件驱动到高级程序，及介绍一些商业视频程序包。

使用本书你需要什么

本书不是为初学者准备的，这并不是说一个刚学习如何编写工作代码的程序员从本书资料中不能获益，相反，许多有用的源代码工作例子对任何一个打算为 PC 机或 PS/2 机认真编写程序设计的人都是有价值的。然而，你的程序设计背景越宽，你将有更多的方法解决视频程

序设计中所包含的各种不同的难易问题。

语 言

本书中,我用汇编语言和 C 语言编写了大部分程序设计举例,虽然我有意避免一些 C 语言更多的神秘句法构成。如果用汇编语言和用 C、Pascal、FORTRAN、PL/1 或结构化 BASIC 是很顺手的,那么,你阅读源代码举例将是没有问题的。

此外,第十三章用不同的内存模式和子例程调用协议讨论了一些高级语言界面,并根据其说明用自己喜欢的语言翻译器,将可调用的 C 源代码举例转换为子程序调用协议。

你也许打算用一些其他程序设计工具,如果你打算用下一个源代码举例试验一下。例如,一个好的汇编语言调试器有很大帮助。如果你打算从高级语言程序调用本书中的汇编语言子程序,那么,你将需要一个目标链接器。这样,像源文件和目标模块激增一样,你也许发现 UNIX 会使实用程序非常有用。

操作系统

本书中的所有程序都可以运行于 MS-DOS 或 PC-DOS 2.0 版或更高版本。无论如何,任何一个源代码在所用的操作系统上验证是没有问题的。但是,如果你将代码传送到 MS-DOS 的早期版本或其他操作系统时要十分小心。

硬 件

拥有一个所喜爱的 PC 或 PS/2 视频显示是不可缺少的。视频程序设计就像游泳:观赏它是一回事,自己亲身试一下又是另外一回事。事实上,如果你打算编写大量的视频程序设计,最好在 PC 机里安装两种不同的视频子系统和显示器,在同样的计算机里有两套单独的视频硬件。当在一个屏幕上显示测试程序输出时,可以在另一个屏幕上运行调试器。这种双重显示硬件配置是真正的时间保存器。特别是,当开发视频图形程序时,可以像第 5 至第 9 章描述的那样去做。

下面列出本书中用于讨论技术开发的各种不同的计算机和视频适配器表:

计算机:

IBM PC/XT

IBM PC/AT

IBM PS/2 型号 30

IBM PS/2 型号 60

适配器:

IBM 单色显示适配器(MDA)

IBM 彩色显示适配器(CDA)

IBM 增强型图形适配器(EGA)

IBM PS/2 显示适配器(DA)

Hercules 图形卡(GC)

Hercules 图形卡 Plus(GCP)

Hercules 彩色卡(CC)

Hercules InColor Card

如果你正在使用这些计算机或适配器或硬件兼容系统中的一个,那么,可以运行源代码例子。

手 册

知道硬件设计内容以及软件和系统 BIOS 如何交互使用,对设计 IBM PC 视频硬件十分有用。这个基本信息可在关于 PC、PC/XT、PC/AT 的 IBM 技术参考手册以及选项和适配器技术参考手册中找到。大部分的 IBM PC 视频设备经销商在硬件中也提供了详细的技术信息。

本书的资料打算补充制造商技术文件中的讨论。我试图与制造商的术语和硬件描述相一致,然而,制造商提供的文件时常出错,如果你发现正式提供的文件和本书之间的差别,你可以(我也希望)能依据本书得到正确的信息。

至此,在本书的内容中,不可避免地存在错误,我真诚地欢迎大家评论、批评和建议。

我已发觉编写一个好的视频软件对自己是一个挑战,但是,你得到的回报也是特别令人满意的。我希望在本书中,你遇到一些尖锐的挑战,也得到一些满意的回报。

第一章 IBM 视频硬件与固件

微型计算机视频系统越来越完善。据 1981 年 IBM PC 机的介绍,工程技术有了很大的改进,视频硬件的市场更加广阔。IBM 和其他竞争对手都以高级的视频适配器和显示器的迅速发展作了回答,它可以像软件一样漂亮地与大众要求保持同步。

本章提供了 IBM PC 和 PS/2 视频硬件发展的一个总体看法,这个看法决不是理解,但它覆盖了 IBM 及 Hercules 提供的用最广泛的视频装置。本章结束于介绍 IBM 的视频 BIOS,这是所有 IBM PCS 及 PS/2 中,建立在 ROM 中的驱动程序集合,为视频应用提供了一个基本的程序设计界面。

1.1 IBM PC 及 PS/2 视频硬件

一个“Plain Vanilla”(无色的香草)IBM PC/XT 或 PC/AT 不包含内部的视频硬件,所以你必须选择并安装自己的视频硬件。在一个典型的配置里,一个视频显示器(监控器)由一个 9 芯电缆连接着 PC 机里安装的视频适配器。这个典型的视频适配器是一块印刷电路板,带有一个九针插座,用于连接显示器的电缆,还有一个 2×31 连接卡管脚,用于插到 PC 机主板上的一个扩展槽中。图 1-1 展示了这些插座,像有些集成电路一样,为许多 IBM 视频适配器共有。在视频适配器中的集成电路,生成一个信号,控制在监控器屏幕上的显示。

当购置一台 IBM PC 机时,必须决定用哪个适配器和监控器。与大部分软件一起应用最广泛的视频适配器是:IBM 单色显示适配器、彩色图形适配器、增强型图形适配器及由 Hercules 制造的单色图形卡。

对比而言,所有的 IBM PS/2 系列计算机都配有内部视频子系统,所以购置一台分离的视频适配器是没有必要的。PS/2 型号 25 和 30 的视频子系统称为多色彩图形阵列。在型号 50、60、70 及 80 中,组合视频子系统是众所周知的视频图形阵列。图形视频阵列子系统也可用作 PC/XT、PC/AT 及 PS/2 型号 30 的适配器。该适配器本质上与 PS/2 一体化的子系统有同样的硬件特征。

1.1.1 IBM 单色显示适配器和彩色图形适配器

当 PC 机在 1981 年被引进时,IBM 提供了两个视频适配器:单色显示适配器(MDA)和彩色适配器(CGA),所设计的 MDA 用于 80 列 25 行字母数字的文本显示的单色监控器(IBM 单色显示器)。CGA 支持或者是一个 RGB 显示器(一个红、绿、蓝色分开输入信号的监控器),或者是一台家用电视机的设置(使用复合信号)。CGA 当然能像字母数字文本一样在逐点的基础上显示图形信息。

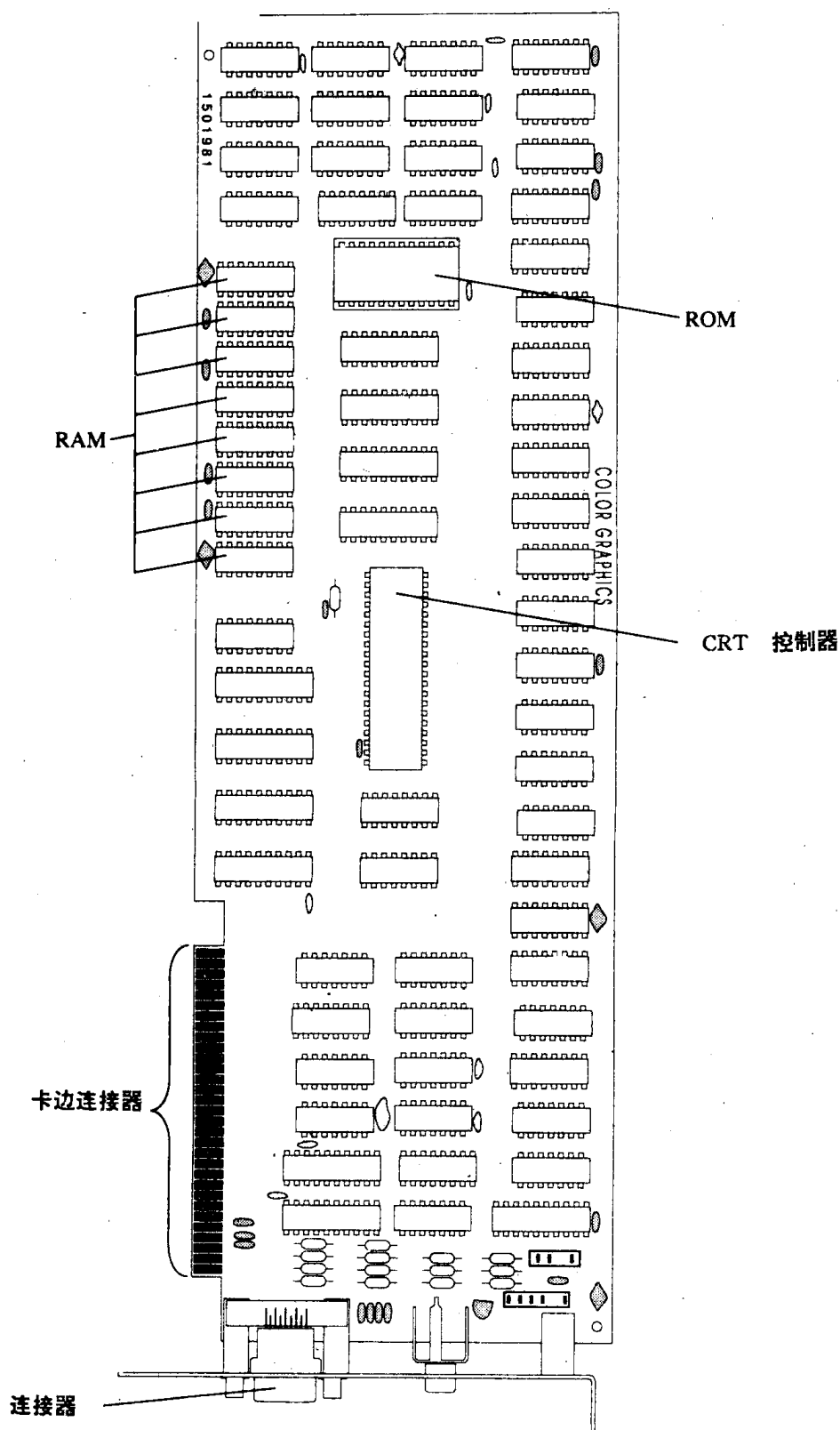


图 1-1. 一个典型 IBM PC 视频适配器。

尽管 MDA 和 CGA 两者都能显示 25 行 80 列文本,但大多数人发现 MDA 的单绿色显示对眼睛更舒服些,这是因为用 MDA 的单色显示器比你所用的 CGA 一些监控器有比较有效的较高分辨率,它的分辨率为 720 点宽 $\times$ 350 点高。CGA 驱动显示的最大分辨率为 640 $\times$ 200。

两个适配器以矩形点阵显示字符,一个简单的计算表明了单色显示器每个字符的点阵为 9 $\times$ 14 而 CGA 显示器只有 8 $\times$ 8 点阵。MDA 的高分辨率提供了更清晰的定义字符,使得更容易阅读,基于这一理由,大部分 PC 机用户阅读文本时更喜欢用 MDA。

另一方面,许多计算机用户需要显示曲线图,及其他除字母数字文本之外的图形信息,亦即在屏幕上显示彩色对许多计算机应用是基本的,因为 MDA 只能显示单色文本,需要图形输出的 PC 机用户只能通过使用 CGA 的点阵彩色图形性能解决,但它不能读文本。

为什么不将高分辨单色显示与彩色图形适配器结合起来以得到最好的呢?遗憾的是,由 MDA 生成的视频信号与驱动 CGA 监控器及替代的要求是不兼容的,不匹配的监控器和适配器导致代替高分辨率显示的监控器的功能失常。

如果你需要可以像彩色图形一样清晰可读的文本,并且可以提供额外的设施,那么你可以在同样的 PC 机上安装 MDA 和 CGA,你能够用单色显示(参照 MDA)来处理文本,并且用 RGB 彩色显示(由 CGA 驱动)来处理彩色图形。

1.1.2 Hercules 图形卡

Hercules 通过增加单色显示适配器的图形功能来解决在同样监控器上显示可读文本和点阵图形的问题。在 1982 年介绍的单色 Hercules 图形卡(HGC)用一个 IBM MDA 在同种绿色单色显示器上显示图形和字母数字文本(除它的图形功能之外,HGC 极好地保留了 IBM 原始 MDA 的功能),显示可读文本及单色图形组合的功能对许多应用是足够的,所以许多 PC 机用户选择了 HGC。因为它从高级软件销售商那里获悉,HGC 在市场上销路极好。

1.1.3 Hercules 图形卡 Plus

HGC+在 1986 年 6 月推出,其最大不同之处在于由最初的 HGC 升级。它能显示按规格改进的、RAM 基本的字母数字字符集,而 MDA 和 HGC 只能显示一个预先定义的 ROM 基本字母数字字符集,因为字母数字比点阵图形字母显示要快得多,在同样的文本应用中 HGC+能二倍或三倍地提高速度。

1.1.4 IBM 增强型图形适配器

一个对更好的文本和图形分辨率要求的不同响应是在 1985 年早些时候推出的 IBM 增强型图形适配器(EGA),EGA 的配置能仿真 MDA 或 CGA。EGA 的“增强”的含义为它可以做到它的先期产品做不到的事情,不像 MDA,EGA 可以在单色显示器上产生点阵图形。进一步说,EGA 改进 CGA 具有的功能生成 16 色字母数字或 640 $\times$ 350 分辨率的图像。

虽然 EGA 的分辨率和色彩性能不比 CGA 好很多,但文本和图形 EGA 比 CGA 显示的更清晰,但 EGA 低廉的价格及高质量的软件应用程序、和适配器的性能使它在市场上具有了硬件水平。

1.1.5 Hercules Incolor 卡

在 1987 年 4 月推出的 Hercules Incolor 卡, 实质上是 HGC+ 的一个 16 色版本。Incolor 卡硬件完全能与 HGC+ 相竞争, 所以运行于 HGC+ 上的程序可以不加修改地运行于 Incolor 卡上, Incolor 卡的分辨率与 HGC 及 HGC+ 的一样为: 720×348 像素。适配器的色彩兼容性等价于 EGA 的, 它可以从 64 色调色板中一次显示 16 色, 适配器必须使用有 350 条垂直分辨率的 EGA 兼容彩色显示器。

注意: 不要将 Incolor 彩色卡与 Hercules 彩色卡相混淆, 一个增加的 CGA 设计用于在同样的计算机上是用一个 HGC 或 HGC+ 卡。

1.1.6 多色图形阵列

多色图形阵列是一个集成于 PS/2 型号 25 和 30 的视频子系统, 从一个程序设计人员的角度看, MCGA 在许多方面与 CGA 相似, 而且 MCGA 具有比较好的分辨率(最大为 640×480 点阵)和改进彩色显示的功能。

MCGA 与上述视频适配器之间的一个重要区别在于 MCGA 生成模拟 RGB 的视频信号。其他则产生数字 RGB 信号。数字和模拟 RGB 之间的区别有些像状态开关(开——关)与无级调节开关之间的区别。数字 RGB 信号视频显示能且只能识别特定的颜色(红、绿或蓝)信号是开还是关。另一方面, 用模拟 RGB 信号的视频显示器将每一个信号的电压变成一定宽度范围的相应颜色强度, 只有模拟视频显示能用于 MCGA。

注意: 有些视频监控器可以配置到模拟或数字视频集中上。如果使用正确的连线且它们是为模拟视频配置的, 那么, 这些监控器可用于 MCGA。

对于一个模拟视频的判断在于它能显示较宽范围的颜色。MCGA 有一个视频数模转换器(DAC)能够使子系统从 262,144 (256K 或 218) 颜色调色板中一次显示 256 种不同的颜色, 除了模拟彩色显示外, IBM 提供了一个用于 MCGA 的模拟单色显示。在这个单色监控器上, MCGA 可以显示 64 个灰度级。

1.1.7 视频图形阵列

视频图形阵列术语明确参照了 PS/2 型号 50、60 及 80 的视频子系统的线路部分。VGA 通常是一个在 EGA 中几个芯片完成同样功能的单个集成芯片。不过, 人们通常用简略的 VGA 去描述全部的视频子系统。

VGA 的程序设计界面与 EGA 的程序设计界面是相似的。许多关于 EGA 的程序可以不加改动地运行于 VGA。VGA 具有较高的显示分辨率的水平(如 720×400 文本或 640×480 图形方式)。与 MCGA 一样, 不管如何, VGA 包含一个视频 DAC, 可以从 262,144 种可能性中一次生成 256 种颜色, 因为 VGA 可以生成与 MCGA 同样的模拟 RGB 信号, 它必须用同样的模拟单色或彩色监控器。

1.2 · ROM BIOS 界面介绍

ROM 中 BIOS(基本输入/输出系统)程序在 IBM PC 机及 PS/2 中都有,ROM BIOS 程序为标准硬件特征提供了一个界面,包括日时钟、键盘、软硬盘,当然还有视频子系统。视频 BIOS 例行程序由一组完成基本视频程序设计任务的简单工具组成。如写字符串到屏幕、清屏、改变颜色,等等……。

尽管 ROM BIOS 视频例行程序有时是比较慢的并且相对是非老练的,但是用于它们的程序在 IBM PC 机和 PS/2 机的不同视频子系统之间是可移植的。更进一步说,大多数 IBM PC 制造商在他们的机器里复制了 IBM 的 BIOS 功能,这样,一个用于访问视频硬件的 BIOS 例行程序比其他程序更具有可移植性。

1.2.1 中断 10H

BIOS 例行程序是用汇编语言写成的。这样,当你用汇编语言编写程序时,可以比较方便地访问它们,所有的 BIOS 视频例行程序都是通过运行 80x86 软件中断 10H 进行访问的(80x86 这一术语参照 Inter8086 系列中的微处理器:8086、8088、80286 及 80386)。基于这一理由,ROM BIOS 视频界面与所知的 INT 10H 界面一样。ROM BIOS 提供了一个视频输入/输出功能序号,通过运行中断 10H 进行访问,这些功能被编了号。运行中断 10H 时,必须在描述这些功能的 80x86 寄存器 AH 中放入这个号。

一直地运行中断,意味着通常带有参数的 80x86 寄存器传递给 BIOS 例行程序。如果 INT 10H 函数返回数据到你的程序,可以通过保留一个或多个 80x86 寄存器中的数据而做到。这个寄存器基本的参数传递方案是用汇编语言程序完成的。

让我们看一下如何使用 INT 10H 界面,举例 1-1 为汇编语言例行程序 SetVMode()的测试。这个例行程序可以链接到供 Microsoft C 语言编写的程序(在产生名字之前加下划线,在 PROC 说明中的 near 关键字,以及下列 Microsoft C 转换中在传递参数时堆栈的使用),例行程序的核心是它们调用 ROM BIOS 来配置特定视频模式的视频硬件(该操作的详细说明在第 2 章及附录 A 中讨论)。

举例 1-1. SetVmode()。

```
;
;NAME:          SetVmode
;
;Function:      Call IBM ROM BIOS to set a video display mode.
;
;              viod    SetVmode(n);
;
;              int     n;                /* video mode */
;
```

```

ARGn          EQU      byte prt [bp+4]      ;stack frame addressing
EQUIP_ FLAG    EQU      byte prt ds:[10h]
CGAbits        EQU      00100000b           ;bits for EQUIP_ FLAG
MDAbits        EQU      00110000b
- TEXT         SEGMENT byte public 'CODE'
                ASSUME  cs:- TEXT
                PUBLIC  - SetVmode
- SetVmode     PROC      near
                push    bp                    ;preserve caller registers
                mov     bp,sp
                push    ds
                mov     ax,10h
                mov     ds,ax                ;DS-->Video Display Data Area
                mov     bl,CGAbits           ;BL:=bits indicating presence of CGA
                mov     al,ARGn              ;AL:=desired video mode number
                mov     ah,al
                and     ah,7
                cmp     ah,7
                jne     L01                  ;jump if desired mode not 7 or 0FH
                mov     bl,MDAbits           ;BL:=bits indicating presence of MDA
L01:           and     EQUIP_ FLAG,11001111b
                or      EQUIP_ FLAG,bl      ;set bits in EQUIP_ FLAG
                xor     ah,ah                ;AH:=0 (INT 10h function number)
                push    bp
                int     10h                  ;call ROM BIOS to set the video mode
                pop     bp
                pop     ds                    ;restore caller registers and return
                mov     sp,bp
                pop     bp
                ret
- SetVMode     ENDP
- TEXT         ENDS
                END

```

实际上调用视频 BIOS 很简单。首先,描述功能的号放入寄存器 AH(XOR AH,AH),然后,保护寄存器 BP 的内容到栈(PUSH BP)之后。例行程序通过运行中断 10H(INT 10H)调用 ROM BIOS 功能。

在举例 1-2 中,一个称为 GetVmode()的互补的例行程序查询 BIOS 得到当前视频模式值。例行程序通过运行中断 10H 的 0FH 功能得到这个值。ROM BIOS 的功能在寄存器 AL 中保留这个值。GetVmode()然后返回这个值到调用程序中。

举例 1-2. GetVmode()。

```

;
;NAME:          GetVmode
;
;Function:      Call IBM ROM BIOS to set a video display mode.
;
;Caller:        Microsoft C;
;
;              int    GetVmode();
;
;
;
_TEXT          SEGMENT byte public 'CODE'
               ASSUME  cs: _TEXT
               PUBLIC  _GetVmode
_GetVmode      PROC    near
               push    bp                      ;preserve caller registers
               mov     bp,sp
               mov     ah,0Fh                  ;AH:=0FH (INT 10h function number)
               push    bp
               int     10h                      ;call ROM BIOS to get Video mode number
               pop     bp
               xor     ah,ah                    ;AX:=video mode number
               mov     sp,bp
               pop     bp
               ret
_GetVmode      ENDP
_TEXT          ENDS
               END

```

1.2.2 视频显示数据区

在举例 1-1 中, 实际调用 ROM BIOS 之前的代码修改反映 PC 机视频子系统状态的几个全局变量中的一个全局变量。这些变量通过所有的 ROM BIOS 视频程序更改和引用。他们集中在一个 RAM 的块中。在 IBM 的技术文献资料中称为视频显示数据区(或视频控制数据区), 视频显示数据区由 RAM 的两个块组成, 第一块对应内存的 0040:0049 和 0040:0066 之间, 第二块对应 0040:0084 和 0040:008A 之间。

一些视频 BIOS 程序也引用 0040:0010(在 IBM 技术文献资料中称为 EQUIP-FLAG)的一个全局变量中的 2 位域。该变量的第 4 和第 5 位指示当计算机开机引导时的缺省视频模式。在 SetVmode() 中的代码更改该位的值来完成视频模式的选择。例如, 如果一个单色显示适配器(MDA)请求所希望的视频模式, 则 EQUIP-FLAG 中的位值被相应地更新(再则, ROM BIOS 视频模式的详细说明可在第 2 章和附录 A 中找到)。

注意:贯穿于本书中, 有引用 INT 10H 界面, BIOS 的视频显示数据区, 以及特别感兴趣的视频显示数据