

第一章

长城 0520 CH 的显示原理

第一节 概 述

长城 0520 CH 高级汉字微机采用 648×504 高分辨率的彩色显示器、高分辨率彩色 / 图形显示适配器 (俗称高显彩卡) 与标准的 IBM-PC / XT 的 640×200 分辨率的彩色 / 图形显示适配器 (俗称彩显卡) 相比 存在较大的差异。为了保证和 IBM-PC / XT 完全兼容, 主机插件槽上同时插有这两块卡, 两卡之间用扁平电缆相连, 并装有扳键开关, 可以方便地选择使用。

图 1-1、图 1-2 分别为彩显卡和高显彩卡的逻辑框图。在本章中, 先介绍使用彩显卡的标准 IBM-PC 显示方式 然后再分析高显彩卡 阐述长城 0520 CH 所特有的字符 / 图形显示原理。这里不准备详细描述它们的电子线路及有关物理过程, 主要是从软件和应用开发的角度, 来讨论它们在程序设计方面所呈现的特性和逻辑功能。

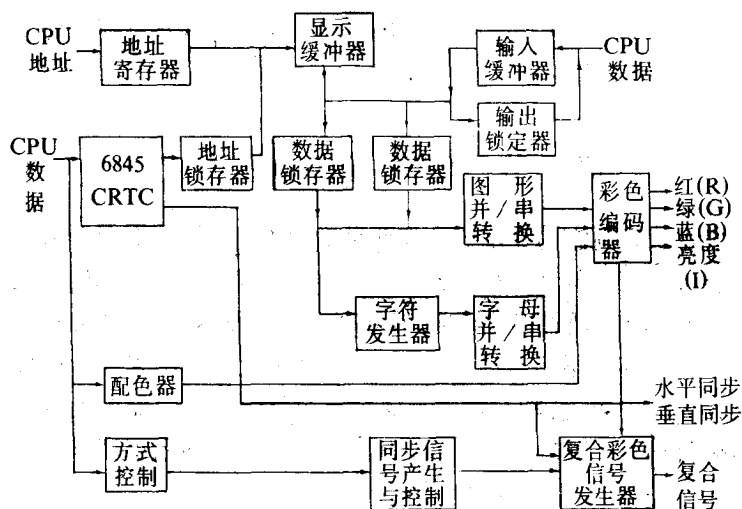


图 1-1 彩色 / 图形显示适配器方框图

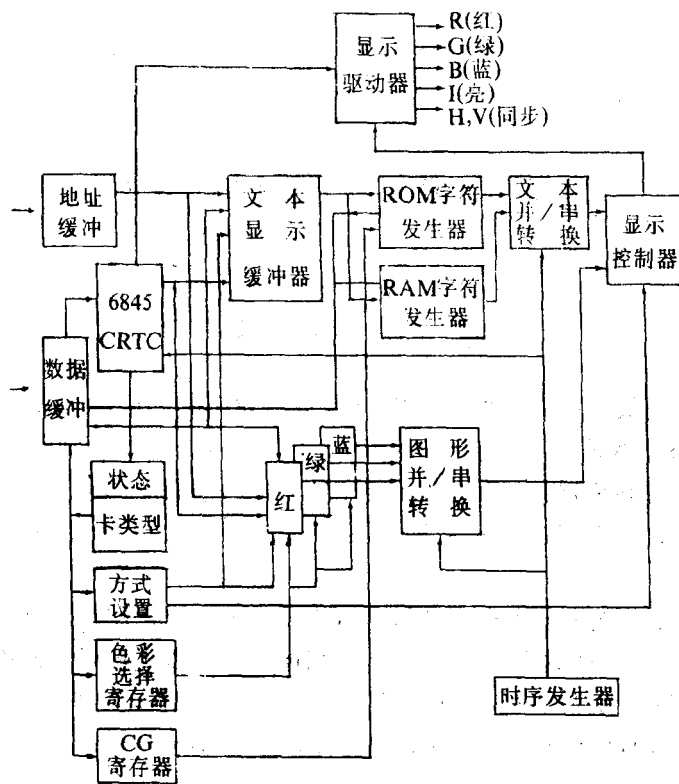


图 1-2 GW-0520 CH 高分辨率彩色 / 图形显示适配器方框图

第二节 IBM-PC 显示方式

个人计算机发送信息到它的显示屏上有两种常用的方法。一种是计算机把显示器作为一个输入输出设备来对待。计算机使用这种方法给显示器控制器发出各种命令，包括显示文本信息的命令。另一种是所谓存储变换，在这种显示方法中，计算机和显示器共用一些可寻址的存储器。显示器连续不断地读存储器，并在屏幕上显示出来。计算机只要简单地改变那些存储器所存放的内容，就能在显示器屏幕上“写”信息。存储变换方法的好处之一是程序只要简单地检查一下共用的存储器所存放的内容即可从显示屏读出信息。显示屏和共用的存储器是一致的，读写存储器就是读写显示屏。

搞计算机的人对第一种方法是比较熟悉的。这种方法的缺点是计算机和它的显示屏没有紧密的和特殊的关系，因此在显示屏上很难实现神奇的技巧。IBM 注重的是技巧，所以它吸取了第二种方法的优点，采用命令和存储变换相结合的方法。对于在屏幕上显示数据，可使用存储变换，但是应使用命令控制显示器的某些方面。

一、IBM-PC 字符显示方式

这种方式也称为字母数字模式 (A / N 模式)。在这种方式下显示缓冲器存放显示字符

的ASCII码和属性，这是和图形显示方式的最根本的区别。根据彩色显示还是黑白显示，每行40个字符还是80个字符，字符显示方式又可分成如下四种工作方式：

40×25 黑白字符方式	方式号 0
40×25 彩色字符方式	方式号 1
80×25 黑白字符方式	方式号 2
80×25 彩色字符方式	方式号 3

通常选用方式3，整个屏幕显示25行，每行80个字符。每个字符在显示屏上呈现的是8×8点阵的字符图形，并可选择16种颜色之一作为底色（背景色）显示色（前景色）甚至还能让其具有闪烁属性。当选择单色字符方式（又称黑白字符方式）时，字符的属性有加亮、闪烁和负象显示。

（一）方式设置和状态寄存器

由第一节概述中的彩显卡逻辑框图可知，整个适配器是以Motorola 6845芯片为核心的。这个器件提供了必要的接口以驱动光栅扫描显示器（CRT）它是一个很复杂而又通用的芯片，可以用来建立许多不同类型和大小的显示。它不但支持上述各种字符显示方式，还支持图形方式。长城0520CH的高分辨率彩显卡就是用它作为CRT控制器（简称CRTC）的。它一旦初始化了，芯片的控制倒是相当简单的。

6845有19个内部寄存器，用来规定和控制光栅扫描式的CRT显示器。所有这些寄存器，除了一个地址寄存器外，都使用一个输入/输出端口地址（3D5H）。地址寄存器端口地址为3D4H，又称为6845基地址寄存器，它的作用是用来指定通过3D5H端口将和18个数据寄存器之中的哪一个打交道。换句话说，通过3D5H端口想读或者写18个数据寄存器中的某一个寄存器之前，必须先把该数据寄存器的编号（或称地址，0~17）通过3D4H端口送到地址寄存器中去。表1-1列出了在各种方式下，6845的数据寄存器必须装入的数值。

需要补充说明的是：

1. 水平扫描总时间（R0）决定水平同步的频率。它的值为显示的字符时间单位加上不显示的（回扫）字符时间单位的总和减1。
2. 垂直寄存器以字符行时间或扫描线时间为单位。垂直同步的垂直频率由R4和R5决定。计算出的字符行（线）时间数实际上是一个非整数（恰好乘以行频得到准确的50Hz或60Hz的垂直刷新频率），整数部分减1存放在R4，小数部分存放在R5。
3. 隔行扫描方式寄存器（R8）是一个仅有两位的只写寄存器。其值为0或2时表示选择正常同步方式，其值为1时选择隔行同步扫描方式，其值为3时选择隔行同步及视频光栅扫描方式。
4. 最大扫描线地址寄存器（R9）决定每个字符（包括行间隔在内）的扫描线数。
5. 光标起始线寄存器（R10）的低五位规定光标起始线位置，而其第6、第5位则控制光标格式。当第6位、第5位均为0时表示光标不闪烁，当第6位为0、第5位为1时表示光标不显示，当第6位为1、第5位为0时表示以场频的1/16闪烁，当第6位、第5位均为1时表示以场频的1/32闪烁。
6. 起始地址寄存器（R12、R13）是一个14位的只写寄存器。它决定垂直消隐后第一个刷新地址，换句话说就是显示缓冲器（CPU和CRT共用存储器）的起始位置。它由8位低

地址寄存器 R13)和 6 位高地址寄存器 R12) 组成。

工作方式的设置除了要对上述 6845 的数据寄存器赋以相应的值以外，还要对彩显卡上的方式选择寄存器的值进行设置，具体可按表 1-2 来安排。

表 1-1 6845 寄存器的说明

寄存器编号	寄存器名称	寄存器类型	参数单位	40×25 字符方式	80×25 字符方式	图形方式
R0	水平扫描总时间	只写	字 符	38	71	38
R1	每行字符数	只写	字 符	28	50	28
R2	水平同步位置	只写	字 符	2D	5A	2D
R3	水平同步宽度	只写	字 符	0A	0A	0A
R4	垂直扫描总时间	只写	字符行	1F	1F	7F
R5	垂直总调节	只写	扫描线	06	06	06
R6	每帧显示行数	只写	字符行	19	19	64
R7	垂直同步位置	只写	字符行	1C	1C	70
R8	隔行扫描方式	只写	—	02	02	02
R9	最大扫描线地址	只写	扫描线	07	07	01
R10	光标起始	只写	扫描线	06	06	06
R11	光标结束	只写	扫描线	07	07	07
R12	起始地址(高位)	只写	—	00	00	00
R13	起始地址(低位)	只写	—	00	00	00
R14	光标(高位)	读写	—	× ×	× ×	× ×
R15	光标(低位)	读写	—	× ×	× ×	× ×
R16	光笔(高位)	只读	—	× ×	× ×	× ×
R17	光笔(低位)	只读	—	× ×	× ×	× ×

表 1-2 方式选择寄存器汇总表

方式选择寄存器 (7 6 5 4 3 2 1 0)	操 作 方 式
× × 1 0 1 1 0 0	40×25 黑白字符方式
× × 1 0 1 0 0 0	40×25 彩色字符方式
× × 1 0 1 1 0 1	80×25 黑白字符方式
× × 1 0 1 0 0 1	80×25 彩色字符方式
× × × 0 1 1 1 0	320×200 黑白图形方式
× × × 0 1 0 1 0	320×200 彩色图形方式
× × × 1 1 1 1 0	640×200 黑白图形方式

方式选择寄存器被赋值时，使用输出指令 (OUT) 其端口地址为 3D8H。

在彩显卡上还有一个 4 位的只读寄存器，其值反映了当时显示的某些状态。第 0 位表示 CRT 显示器是否处在水平扫描或垂直扫描的回扫期。若其位值为

1, 则 CPU 可以向显示缓冲器送入新的显示信息而不影响屏幕画面。第 1 位表示光笔触发器置位否 第 2 位表示光笔按钮开关状态。第 3 位表示视频信号输出与否。这个寄存器称为状态寄存器 其值可以通过输入指令 (IN 来读取 其端口地址为 3DAH。

(二 序符信息的显示

如上所述,IBM 采用存储变换的方式来显示字符或图形,CPU 和 CRT 显示器共用显示缓冲器,显示缓冲器是双端口读写存储器。它位于彩显卡上,其段地址为 B800H 起始偏移地址为 0 共有 16K 字节。每两个字节和屏幕上显示的一个字符相对应,所以总共可以存放 8K 个供显示的字符信息(实际使用 8000 个)。由于显示屏一次最多显示 2000 个字符(80×25 方式或 1000 个字符 40×25 方式)所以任何时候显示屏只能和整个显示缓冲器的一部分相对应。通常把这部分称为当前显示页(或当前有效页)显而易见在 80×25 方式时共有四个不同页面(0~3);在 40×25 方式时共有八个不同页面(0~7)。当前显示页面通常是第 0 页然而通过更改 6845 的起始地址寄存器(R12、R13)的值就可以很方便地把其他页面作为当前显示页面。需要指出的是 6845 使用的地址值为实际地址的 1/2。

显示屏上行的顺序规定为从屏幕最上面的行数起,每行的字符又是从左到右数起。所以第 0 行的 80 个字符,从左到右分别和当前页面起始地址开始的 160 个字节相对应第 1 行的 80 个字符,从左到右分别和当前页面起始地址+160 开始的 160 个字节相对应依次类推。

每个字符对应的两个字节中第一字节(偶地址字节)存放字符的 ASCII 码第二字节(奇地址字节)存放该字符的显示属性该字节的每位所代表的含义见图 1-3。第 0~3 位用来选择 16 种颜色之一来作为该字符的颜色(I R G B 的 16 种组合所分别代表的颜色请参阅表 1-3)。第 4~6 位 RGB 值与彩显卡上的色彩选择寄存器的第四位相结合,用以确定该字符的底色(背景)第 7 位用来控制字符显示的闪烁与否。

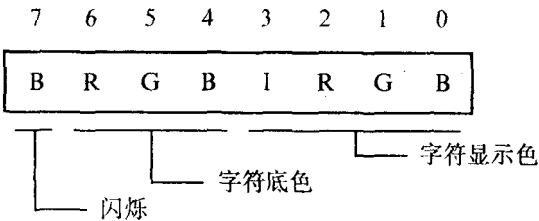


图 1-3 属性字节的格式

(三) 字符发生器

虽然在字符显示方式下,显示缓冲器中每两个字节和屏幕上显示的一个字符相对应,但是 CRT 显示器是光栅扫描式的,需要的是点阵信息。从图 1-1 可知在 6845 的控制下,由显示缓冲器取出显示字符的 ASCII 码,并以此码到一个字符发生器里取得显示字符的点阵信息。这个字符发生器是一个 8K 字节的 ROM 它不能在软件控制下进行读写操作其中存放三套不同字体的 ASCII 字符点阵字模即 7×9、7×7 和 5×7。系统一般采用的是 7×7 点阵的字体若需采用 5×7 的,可以通过彩显卡上的一个跨接线插头来选用。

表 1-3 彩色编码表

I	R	G	B	彩 色
0	0	0	0	黑
0	0	0	1	蓝
0	0	1	0	绿
0	0	1	1	青
0	1	0	0	红
0	1	0	1	洋 红
0	1	1	0	棕
0	1	1	1	淡 灰
1	0	0	0	深 灰
1	0	0	1	浅 蓝
1	0	1	0	浅 绿
1	0	1	1	浅 青
1	1	0	0	淡 红
1	1	0	1	浅洋红
1	1	1	0	黄
1	1	1	1	白

(四) 程序设计

显示字符的程序设计大致可在如下四个界面进行：

- 1. 高级语言或者数据库语言；
- 2. 操作系统 通常是 PC-DOS)；
- 3. ROM BIOS；
- 4. 裸机。

利用高级语言或数据库语言里的显示输出语句来进行程序设计 是最容易学会的 例如 ,BASIC 语言中的 PRINT 语句。但是它们的功能不够强，使用不够灵活，有些特殊效果难以实现。与此相比 在裸机界面上利用上述彩显卡所具有的接口 直接运用汇编语言来进行程序设计，可以得到执行速度最快、显示效果最好的显示程序。但是，这要求编程人员对于硬件 彩色显示适配器和 8088指令集要非常熟悉，而且编写时相当繁琐、费时间。所谓 ROM BIOS实际上是一组事先编写好的有关输入输出的子程序，它已固化 在机器内部的 ROM 芯片之中，供高层软件或用户程序通过软件中断指令来调用。其中有关显示的共有 16 个子程序 统一通过 INT 10H 软件中断指令来调用。在这个界面上进行程序，设计要比裸机界面方便得多。在操作系统界面上设计，虽然更方便，但是 PC-DOS 有关显示的功能调用只提供两个，且都是关于显示字符的，而没有关于显示图形的。这两个功能调用，一个是在当前光标位置上显示一个字符，另一个是从当前光标位置起显示一个字符串。

下面，首先介绍一下在裸机界面上如何进行字符显示的程序设计，然后再简单地介绍一下 ROM BIOS 中有关字符显示的功能调用子模块。

1. 彩色字符显示的编程方法

(1) 彩显卡输入输出端口地址

CPU 对彩显卡的控制是通过输入输出指令来进行的，彩显卡共占用 16 个输入输出端口地址 (3D0H~3DFH) 其中九个已有具体定义 (表 1-4)：

表 1-4 彩显卡输入输出地址

输入输出地址	寄存器名称
3D4(3D0)	6845 CRTC 索引寄存器
3D5(3D1)	6845 CRTC 数据寄存器
3D8	操作方式控制寄存器
3D9	色彩选择寄存器
3DA	状态寄存器
3DB	清除光笔锁存器
3DC	预置光笔锁存器

(2) 程序设计的一般步骤

1) 初始化工作方式

选定四种字符显示方式之一；把方式选择寄存器 (3D8H) 中的第 3 位置 0 即暂不允许视频信号输出；把与选定方式相对应的一组参数 (见表 1-1)置入 6845 CRTC 的数据寄存器 由选定的工作方式按表 1-2 设置方式选择寄存器。

2) 视需要设置色彩选择寄存器

色彩选择寄存器是一个 6 位只写寄存器 其端口地址为 3D9H。在字符显示方式下 第 3~0 位所指示的 IRGB 值代表屏幕上矩形显示区外部边界 (边框) 的颜色。第 4 位是决定彩色字符显示方式下字符底色的 I 值。

3) 向显示缓冲器写入要显示的字符信息 —— 字符的 ASCII 码和属性

4) 视需要作光标、滚动及选页等处理

光标通常用来指示当前显示字符的位置,它由 6845 CRTC 来维持 改变光标的位置只需要更改 6845 数据寄存器 R14 和 R15 的值。光标本身图案也只要通过更改 6845 数据寄存器 R10 和 R11 来实现。

滚动最明显的需求是因为屏幕显示满 25 行 通常只显示 24 行正文 最后一行作为提示行用)后,再显示后续行就要把第一行推出屏幕,各行依次上移一行,在最下面腾出一空行来。可以使用串传送指令来实现软件滚动。

选择当前显示页面 如前所述 只要更改 6845 数据寄存器 R12 和 R13 的值就行了。

2. ROM BIOS 中有关彩色字符显示功能子模块的简介。

ROM BIOS 中有关显示的子程序共有 16 个 它们都是通过软中断指令 INT 10H 来调用的。其中有关字符显示的有 14 个 为了便于以后在叙述长城 0520 CH 高分辨率彩色显示时作比较,现将它们的功能和用法简单介绍如下:

(1) SET-MODE AH=0

设置显示方式。包括对当前各屏幕参数单元初始化,对 6845 CRTC 的数据寄存器赋值。设定什么显示方式 由送 AL 寄存器的参数来决定:

AL = 0 40 × 25 黑白字符显示方式

AL = 1 40 × 25 彩色字符显示方式

AL = 2 80 × 25 黑白字符显示方式

AL = 3 80 × 25 彩色字符显示方式

(2) SET-CTYPE AH=1

设置光标图案的起始行和终止行,光标的宽度是不变的,始终为一个 ASCII 字符的宽度。

CH = 光标的起始行 (0~7)

CL = 光标的终止行 (0~7)

(3) SET-CPOS AH=2

设置光标在屏幕上的位置 包括光标所在行号、列号及所在页面号。

DH = 行号 DL = 列号

BH = 页面号

(4) READ-CURSORI AH=3

读指定显示页上的光标。包括光标所在行号、列号及光标的起始行和终止行。

BH = 指定的显示页面号

调用返回时:

DH = 光标所在行号

DL = 光标所在列号

CH = 光标的起始行

CL = 光标的终止行

(5) READ-LPEN AH=4

读光笔位置。

调用返回时:

AH = 0 / 1 光笔开关未按下或不触发 / 寄存器中有效的光笔值

DH = 光笔位置的字符行值

DL = 光笔位置的字符列值

CH = 光笔位置的扫描线行值 (0~199)

BX = 光笔位置的象素列值 (0~319, 639)

(6) ACT-DISP-PAGE AH = 5

根据指定的页面号，计算出该页在显示缓冲器中的起始地址及该页光标的行列号所对应的在显示缓冲器中的地址。控制 6845 使指定页面成为当前显示页。

AL = 新的当前显示页面号

(7) SCROLL-UP AH = 6

把指定的左上角和右下角所确定的屏幕窗口中的字符行上滚若干行，上滚后空出的行用指定属性的空行来填写。

AL = 上滚出的行数

CX = 左上角的行、列号

DX = 右下角的行、列号

BH = 空行的属性

(8) SCROLL-DOWN AH = 7

把指定的左上角和右下角所确定的屏幕窗口中的字符行下滚若干行，下滚后空出的行用指定属性的空行来填写。

AL = 下滚出的行数

CX = 左上角的行、列号

DX = 右下角的行、列号

BH = 空行的属性

(9) READ-AC-CURRENT AH = 8

指定页面光标所在位置上的字符的 ASCII 码和属性。

BH = 指定页面号

调用返回时：

AL = 字符 ASCII 码

AH = 字符的属性

(10) WRITE-AC-CURRENT AH = 9

把指定字符，按指定属性写在指定页面光标所指示的位置上。

BH = 指定页面号

CX = 要写字符的个数

AL = 要写字符的 ASCII 码

BL = 要写字符的属性

(11) WRITE-C-CURRENT AH = 10

把指定字符写在指定页面光标所指示的位置上，属性不改变。

BH = 指定页面号

CX = 要写字符的个数

AL = 要写字符的 ASCII 码
(12) SET-COLOR AH = 11
设置边框颜色及彩色字符的底色亮度。

BH = 0
BL 的第 3~0 位为边框颜色的 IRGB 值
BL 的第 4 位为彩色字符的底色亮度 I 值)

(13) WRITE-TTY AH = 14
把指定字符写在当前显示页面光标所指示的位置上, 然后光标移到下一个位置。

AL = 要写的字符的 ASCII 码
(14) VIDEO-STATE AH = 15
返回当前的视频状态。

AL = 当前显示方式号
AH = 当前显示方式的每一行列数
BH = 当前显示页面号

以上内容请读者查阅 IBM-PC / XT 硬件技术手册的附录部分, 那里有全部 ROM BIOS 的带注解的源程序清单, 这些功能子模块的源程序是在 3119 行至 4901 行之中。

二、IBM-PC 图形显示方式

这种方式也称为 APA 模式。在这种方式下, 显示缓冲器中所存放的是显示器屏幕上的每个点 也称为“象素”或“象元”的亮度或颜色信息。根据显示的分辨率, APA 模式又可分成四种不同的图形显示方式:

- * 高分辨率单色图形显示方式 每帧 200 线 每线 640 点 每点只能取两种颜色。方式号为 6。

- * 中分辨率单色图形显示方式 每帧 200 线 每线 320 点 每点只能取两种颜色。方式号为 4。

- * 中分辨率彩色图形显示方式: 每帧 200 线 每线 320 点, 每点可以有四种不同颜色。方式号为 5。

- * 低分辨率彩色图形显示方式: 每帧 100 线 每线 160 点 而每点有 16 种不同的颜色。目前, ROM BIOS 不能支持这种方式。

(一) 方式设置

6845 CRTC 支持图形显示方式 从表 1-1 中可以查到图形显示方式时 应该装入 6845 各数据寄存器的值。关于这些数值需要补充说明如下几点:

1. 高分辨率 (640×200) 和中分辨率 (320×200) 都使用这组数值。
2. R0~R3 中数值的单位和 40×25 字符显示方式的相同。
3. R4~R7 中数值的单位都是扫描线, 与字符显示方式的有区别。

(二) 图形信息的显示

彩显卡上的显示缓冲器 (存储地址从 B800:0000 起连续 16K 字节的 RAM 在图形显示方式下作为图形显示缓冲器。每个点的图形信息和每个字符的字符信息有很大的差异: 在 640×200 方式下 每个点只要用一个 bit 位就可以表示了。在 320×200 方式下 每个点

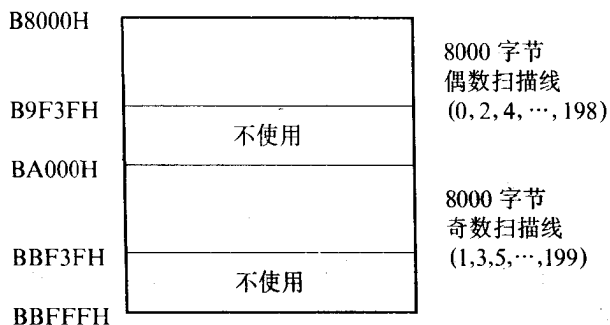


图 1-4 显示缓冲器与屏幕扫描线的对应关系

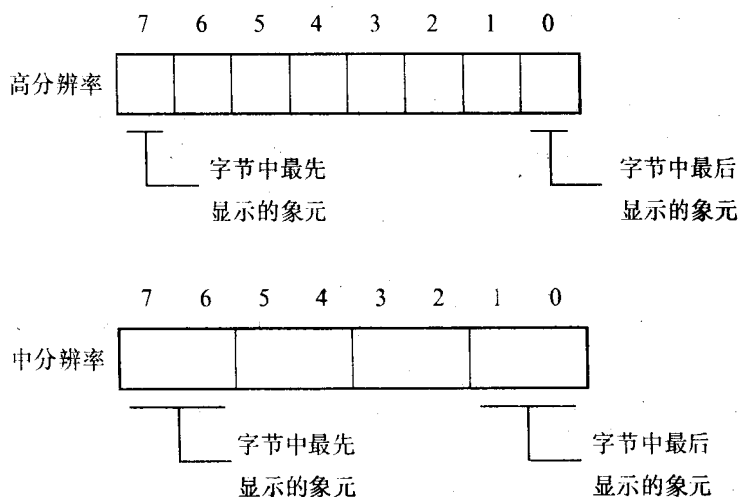


图 1-5 显示缓冲器字节内容与象元位置的对应关系

则需要用两个 bit 位来表示。这些点信息在显示缓冲器中存放的顺序也和字符方式有很大的不同。图 1-4 是显示器屏幕上的象元位置与显示缓冲器中的信息对应关系的示意图。图 1-5 是显示缓冲器中每个字节的信息内容与显示器屏幕上的象元位置对应关系的示意图。

无论哪种分辨率,16K 显示缓冲器中都是只存放一个显示页面的内容。也就是说,在图形显示方式下当前显示页面只能是第 0 页,而不存在选择其他页面的问题。

那么,象元的显示色与显示缓冲器中所存放的内容是如何对应的呢?

1. 在 640×200 单色图形显示方式下,象元在显示缓冲器中对应的内容是一个 bit 位,也就是说只能取值 1 或 0。0 代表该象元取黑色背景色(暗)而 1 表示该象元为一个亮点,这个亮点通常是白色。所以 640×200 方式常常叫做黑白方式。然而,通过对彩显卡上的色彩选择寄存器端口地址为 3D9H 的低四位(IRGB 位)重新置值,是可以使得屏幕上所有亮点都呈现如表 1-3 所示的十六种颜色中的一种的。

2. 在 320×200 彩色图形显示方式下,象元在显示缓冲器中对应的内容是两个 bit 位,可能有四种取值:0、1、2 和 3。取值 0 表示该象元显示屏幕的底色,然而,底色又是什么颜色呢?在 320×200 方式下,屏幕的底色有十六种,是由色彩选择寄存器的低四位(IRGB 值)来确定的。同样,取值 1、2、3 也不能单独决定象元的显示色,而要和色彩选择寄存器的第 5 位共同来决定。详见表 1-5。

(三) 字符发生器

这是一个容易引起误解的话题,它并不是指彩显卡上的 ROM 字符发生器。从图 1-1

上可以看出图形显示方式并不需要使用字符发生器,显示缓冲器的内容直接送到图形并/串转换部分。这里所说的字符发生器是指在填写显示缓冲器时,需要使用的字符发生器。由于在图形显示方式时,CRT显示器仍然作为控制台显示器使用,所以必须要能方便地根据字符的ASCII码来显示字符。因此,在填写显示缓冲器之前就需要根据ASCII码到某个字符发生器里去取相应的点阵信息。

表 1-5 中分辨率象元颜色定义表

色彩选择 器第 5 位	象元的值 C1 C0		象元的颜色
0	0	0	与底色相同
0	0	1	绿
0	1	0	红
0	1	1	黄
1	0	0	与底色相同
1	0	1	青
1	1	0	洋 红
1	1	1	白

由于彩显卡上的 ROM 字符发生器不能被软件调用,所以需要另外提供字符发生器。通常这个字符发生器是在主板上的 ROM BIOS 的芯片上 起始地址为 F000:FA6E 的连续 1K 字节。但须指出 这仅是前 128 个 ASCII 码相对应的点阵信息。对于后 128 个 ASCII 码从 128 到 255 怎么办呢?系统规定由用户自己来定义,在内存某个区域生成它,并把起始地址(偏移地址和段地址)填写到 0000:007C 起始的连续四字节中去(中断 1FH 向量存放单元)。

(四) 程序设计

和字符程序设计一样,图形程序设计通常也可能在四个界面上进行。但 PC DOS 并没有提供有关图形的功能调用。ROM BIOS 也只提供一个写点子程序和一个读点子程序。至于高级语言类,往往是专用绘图软件才有较完善的功能。所以,这方面往往需要使用者自己来开发。

1. 彩色图形显示的编程方法

其基本方法和一般步骤大致和彩色字符显示的编程方法相同,所不同的有如下几点:

(1) 色彩选择寄存器的使用

在图形显示方式时,色彩选择寄存器各位的含义与字符显示方式时的有不同。在 320×200 彩色图形显示方式下 低四位 IRGB 值所代表的是整个屏幕的底色。第五位是指定两种调色板之一的。在 640×200 单色图形显示方式下 低四位 IRGB 值是规定整个屏幕亮点的颜色的。

(2) 往显示缓冲器中填写的不再是字符的 ASCII 码 而是显示点的值。点值在显示缓冲器中存放的顺序也和字符显示方式时的有很大的区别,分成奇、偶扫描线区。这是需要注意的 具体请参见图 1-4 和图 1-5。

(3) 在需要显示字符时,决不能直接把 ASCII 码填入显示缓冲器。必须先到字符发生器或称字符字模库中取得该字符的点阵信息 然后按规定填写显示缓冲器。通常 对于前 128 个字符(ASCII 码从 0 到 127)使用 ROM BIOS 芯片上的字符字模库 而对于后 128 个字符(ASCII 码从 128 到 255 又称扩展 ASCII 字符)使用由奇异中断 1FH 所指示的扩展字模库。

2. ROM BIOS 中有关图形显示功能子模块的简介

前面已经说过, ROM BIOS 中有关显示的功能子模块共有 16 个 其中 14 个是与字符显示有关的, 另外两个只是在图形显示方式下使用。它们是:

(1) READ-DOT AH = 13

读出指定位置上的点值。

DX = 指定位置的扫描线行值 (0~199)

CX = 指定位置的点列值(0~639 或 0~319)

调用返回时:

AL = 读出点值

(2) WRITE-DOT AH = 12

在指定位置上写(显示)点。

DX = 指定位置的扫描线行值 (0~199)

CX = 指定位置的点列值 0~639 或 0~319)

字符显示方式下所使用的 14 个功能子模块中 其中一个初始化模块 设置显示方式 是在图形显示方式下必须使用的, 否则不能进入图形显示方式。使用这个模块时, 规定:

AL = 4 设置 320×200 黑白图形显示方式

AL = 5 设置 320×200 彩色图形显示方式

AL = 6 设置 640×200 单色图形显示方式

其余 13 个模块也可以使用, 不过要知道所显示的字符图形不是来自彩显卡上的字符发生器 而是来自 ROM BIOS 或者自己准备的字符字模库。

第三节 0520 CH 高分辨率字符显示的原理

长城 0520 CH 高分辨率彩色 / 图形显示适配器 高显彩卡 有三种基本工作方式 字符显示方式 (Text), 图形显示方式 (APA), 字符发生器方式 (CG)。在字符显示方式中 屏幕显示 28 行×80 列 ASCII 字符或 28 行×40 列国家标准汉字 也可以是 ASCII 字符与国标汉字混合显示。当第 80 列处开始显示国标汉字时, 一个附加的第 81 列会被显示 即保证一个汉字在同一行上显示。在字符显示方式中, ASCII 字符被定义在一个 18×8 的点阵中, 实际字符点阵为 16×8。国标汉字被定义在一个 18×16 的点阵中 实际字符点阵为 16×16。字符显示属性可以有前景 / 背景反转, 闪烁及高亮度。每个字符可以选择 16 种前景颜色及 8 种 (或 16 种) 背景颜色。每个字符的上、下、左、右还可以附加划线。

一、结构及原理

(一) 主要部件

由图 1-2 可以看出 高显彩卡也是以 CRTC 6845 为核心的。在字符显示方式时 要用到状态寄存器 (CTATUS REG), 文本显示缓冲器 (TXT DISPLAY BUFFER), ROM 字符发生器 (CG ROM), RAM 字符发生器 (CG RAM), 方式设置寄存器 (MODE SET REG), 字符发生器寄存器 (CG REG) 文本并 / 串转换器 (TXT SERIALIZER) 显示控制器 (DISPLAY CONTROL), 显示驱动器 (DISPLAY DRIVER) 以及时序发生器 (TIMING GENERATOR)。

1. 6845 CRT 控制器

这是一个通用性很强的芯片，它也能控制现在分辨率为 648 点 × 504 线的 28 行字符方式显示。不过，它的 18 个数据寄存器参数要按表 1-6 来填写。

表 1-6 高显彩卡 6845 寄存器参数表

地 址 寄存器	寄存 器号	寄存器类别	编程单位	I/O	显示控制参数 (28 行)	显示控制参数 (25 行)
0	R0	水平字符总数	字符个数	只写	64	64
1	R1	水平显示字符总数	字符个数	只写	50	50
2	R2	水平同步位置	字符个数	只写	54	54
3	R3	水平同步宽度	字符个数	只写	03	03
4	R4	垂直字符总数	字符行数	只写	1C	19
5	R5	垂直总数调整	扫描线	只写	1F	1D
6	R6	垂直显示字符总数	字符个数	只写	1C	19
7	R7	垂直同步位置	行列行	只写	1C	19
8	R8	隔行扫描方式	—	只写	02	02
9	R9	最大扫描线地址	扫描线数	只写	11	11
A	R10	光标起始线	扫描线	只写	4F	4F
B	R11	光标终止线	扫描线	只写	11	11
C	R12	起始地址(高)	—	只写	00	00
D	R13	起始地址(低)			00	00
E	R14	光标位置(高)	—	读/写	00	00
F	R15	光标位置(低)			00	00
10	R16	光笔(高)	—	只读	××	××
11	R17	光笔(低)			××	××
注: 所有寄存器值均用 16 进制表示						

和彩显卡一样，6845 的地址寄存器和数据寄存器的端口地址分别为 3D4H (3D0H) 和 3D5H (3D1H)。

2. 状态寄存器

其端口地址为 3DAH。这是一个 8 位的只读寄存器，各位所代表的含义与彩显卡的有所不同。具体见图 1-6。

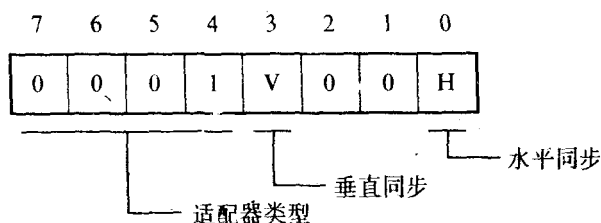


图 1-6 状态寄存器的信息格式

3. 文本字符显示缓冲器

文本显示缓冲器处于 CPU 的寻址空间中, 从 16 进制的 B8000 (即 B800:0000) 和 C0000 (C000:0000) 开始 每处提供 32K 字节 实际使用 17.5K 的动态读 / 写存储器。由于采用的是双端口存储器, 因而 CPU 和 6845 都可以访问显示缓冲器。

4. ROM 字符发生器

这个字符发生器由两块 1 兆位的 ROM 芯片组成, 实际为字符显示方式提供 7701 个字符的点阵 其中包括国标一、二级汉字。CRT 控制器在文本显示缓冲器读取字符码后就是在此查找相应字符点阵信息的。与彩显卡不同 它还可以由软件来读取 不过必须设高显彩卡的 CG 工作方式。

5. RAM 字符发生器

这是一个辅助的字符发生器, 由 $2 \times 2K$ 字节随机存储器组成, 硬件线路可以对它进行读 / 写操作。它用来存储用户自己定义的 128 个字符的点阵字模。可由软件控制对它进行操作。

6. 方式设置寄存器

其端口地址为 3DDH。这是一个只写寄存器。它控制 CPU 对显示缓冲器的访问方式及在字符显示方式下当前显示的页面。具体规定参见图 1-7。

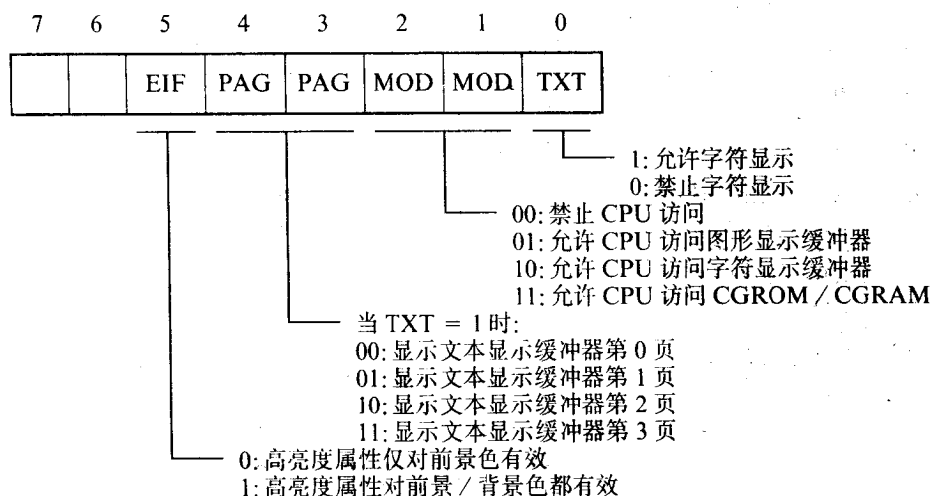


图 1-7 方式设置寄存器的位定义

7. 字符发生器寄存器

其端口地址为 3DFH。这个寄存器在高显彩卡处于字符发生器方式 (CG 方式) 时被使用 存放 CPU 访问字符发生器时的部分地址信息, 各位规定如图 1-8 所示 具体含义在下文中予以阐述。

(二) 字符显示方式工作原理

字符显示方式下显示字符的过程实际上就是把文本 (字符) 显示缓冲器中当前页面所存放的内容映象到 CRT 显示器屏幕的过程。这个过程大致如下: 在 6845 和时序发生器的支

持下 依次从显示缓冲器中当前显示页面取出字符码及属性 到 ROM 字符发生器或 RAM 字符发生器里取出相应字符的点阵字模，送到文本并 / 串转换器进行并 / 串转换。转换后产生的串行信息再发送到显示控制器，在此和图形并 / 串转换器送来的串行信息（如果当时是允许图形显示的话）作叠加等处理，最后传送到显示驱动器，由显示驱动器控制 CRT 显示器屏幕的显示。

（三）字符发生器方式

这种方式提供了用户通过 CPU 来访问高显彩卡上的 ROM / RAM 字符发生器的可能性。通常，这种需求来自下述两种情况：一种情况是由于需要显示非国标汉字或者特殊图形，这时就需要把事先准备好的扩展汉字或图形的点阵

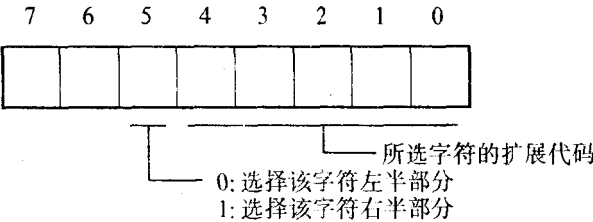


图 1-8 CG 寄存器各位定义

信息存放到 RAM 字符发生器中去。另外一种情况是有时需要查看获取 ROM / RAM 字符发生器中某个字符的点阵字模。一般，这种需求是在作系统维护时产生的。由于目前高显彩卡上 RAM 字符发生器不允许软件读，只允许软件写，所以扩展字符的点阵信息在存放到高显彩卡的 RAM 字符发生器后 还要继续保存在内存之中。这样，一旦需要读取时，可以到那里去读取。

1. 方式设置

这是 CPU 访问高显彩卡的 ROM / RAM 字符发生器之前必须要完成的。前面已经讲过，定义这种方式要使方式设置寄存器的第 2 位和第 1 位都置 1。由于方式设置寄存器的端口地址为 3DDH，所以只要使用下述四条汇编指令便可完成方式设置：

```
MOV    DX, 3DDH
MOV    AL, TXTREG
OR     AL, 6
OUT    DX, AL
```

第二条指令中的 TXTREG 是保存原来方式设置值的单元。

2. 访问地址

（1）ROM 字符发生器

ROM 字符发生器总共有 250K 字节 2×1M 位 存储空间 实际依次存储：256 个标准 ASCII 字符（256×32=8192, 8K 字节），国标 1 区～9 区全角字母、数字、符号 9×94×32=27072, 26.4375K 字节），国标 16 区～87 区汉字 72×94×32=216576, 211.5K 字节），总共是 245.9375K 字节 共存放 7870 个字符 实际为 7701 其余的是空白点阵。由于每个字符的点阵信息为 32 个字节 256 个标准 ASCII 字符只使用前 16 个字节（即左半部分点阵信息），所以寻址定位到每个字符点阵字模的首址，软件只要提供 13 位二进制地址（0～8191）就足够了。此地址码实际上就是字符在字符发生器存放的顺序码（0～7869）。由字符的 ASCII 码或国标汉字的区位码转换为地址码的方式如下：

ASCII 字符

地址码 = ASCII 码

国标汉字

地址码 = $256 + (\text{区码} - 1 - 6) * 94 + (\text{位码} - 1)$

(汉字机内码 = 汉字区位码和 A0A0H 按位加)

访问通常使用 8088 的串传送指令, 此时有关寄存器的设置是: B800H 送数据段寄存器 (DS) 作为源段地址, 上述公式计算出来的地址码的低 8 位送 SI 寄存器作为源偏移地址。把 16 送 CX 表示一次读取 16 个字节 (左半部分或者右半部分), 同时还必须把地址码的高 5 位存放到 CG 寄存器的低 5 位 (第 4 位 ~ 第 0 位) 把读取左半部分 (前 16 个字节) 或是右半部分 (后 16 个字节) 的信息存放到 CG 寄存器的第 5 位去。

(2) RAM 字符发生器

RAM 字符发生器共有 4K 字节随机读 / 写存储器, 可以用来存储 128 个用户自定义的字符字模。目前系统只允许软件对它进行写操作, 即把准备好的字符字模存放进去。访问时地址码只能是 0~127。并且通常也是使用 8088 的串传送指令, 首先要对 CG 寄存器进行设置, 第 4 位 ~ 第 0 位清 0, 第 5 位则根据是写左半部分还是写右半部分来设置。然后, 把 B800H 送入辅助段寄存器作为目的段地址, 把地址码 (0~127) 送入 DI 寄存器作为偏移地址, 把 16 送入 CX 作为本次写的字节数。

细心的读者可能已经发现, 这时地址码和读 ROM 字符发生器中前 128 个标准 ASCII 字符可能是一样的。确实如此, 所以才只允许软件通过 CPU 对 RAM 字符发生器进行写。硬件线路可以通过读或是写来确定指向 ROM 字符发生器还是指向 RAM 字符发生器。

3. RAM 字符发生器的有效利用

RAM 字符发生器最多只能同时存储某 128 个扩展字符的字模。显然, 如果把某 128 个扩展字符存储完了, 不再更动它, 那么就不能再用来显示其他扩展字符了。最好对它进行动态管理, 不断地更新它。这样就可以使得能使用的扩展字符不受限制, 只要同时在屏幕上显示的扩展字符不超过 128 个就行了。目前, 高显彩卡的 ROM BIOS 就是这样来使用高显彩卡上的 RAM 字符发生器的。具体做法是:

- * 扩展字符的字模只在要马上显示它的时候, 也就是往文本显示缓冲器当前显示页写入时, 方才打入 RAM 字符发生器。

- * 在 B800: 719C 处安排一个长度为 128 字节的地址码表, 其中存放 RAM 字符发生器的全部地址码 (0~127)。在 B800: 721C 处安排一个长度为 128 字 (256 字节) 的扩展码表, 其中存放字模目前已存在 RAM 字符发生器中的扩展字符的字符码 (对于汉字是两字节的机内码, 对于 ASCII 字符高位字节为 0, 低位字节为 ASCII 码), 让这两张表始终保持对应关系, 即扩展码所对应的地址码恰好是扩展字符字模打入 RAM 字符发生器时的地址码。

- * 一旦显示扩展字符, 即往文本显示缓冲器中写字符码及其属性时, 首先去扫描上述的扩展码表。若该扩展码不在表中, 则把该扩展码存入表中最后一项, 其他各项依次上推, 第一项被清除, 同时把地址码表作相应处理, 即把第二项至最末项依次上推, 而把第一项移到最后一项。然后, 以这对应的地址码去把字模打入 RAM 字符发生器。若扩展码在表中, 则把该项移到最后一项, 原该项下一项起到最后一项都上推一项。同时把地址码表也作相

应处理，把对应项移到最后一项。以上的维护策略可以使表中始终保持最近使用过的、最多可达 128 个的扩展字符。

(四) 文本 字符 显示缓冲器

1. 字符显示缓冲器的结构

字符显示缓冲器分成两部分：基本区和扩展区。基本区和扩展区是一一对应的。基本区的起始地址为 B800:0000H 扩展区的起始地址为 B800:8000H(或写作 C000:0000H)，每个区都分为 4 页 页间隔为 2000H(8K 字节)，每页实际使用 1180H 字节 28 行 每行 80 字符 每个字符使用两个字节)余下 E80H 字节不用。

每个显示字符由显示缓冲器中的 4 个字节来定义，在基本区中填写的是字符的显示码 (填写在该区的偶地址字节)和字符的第一属性 (填写在该区的奇地址字节)

扩展区中填写的是字符的显示码扩展部分 (填写在偶地址字节)和字符的第二属性 (填写在奇地址字节)

字符显示缓冲器示意图请参见图 1-9。

2. 字符显示缓冲器字符信息格式

显示字符有两类，一类是 ASCII 字符，另一类是汉字字符。在机器内部，ASCII 字符用一字节的 ASCII 码表示，汉字用两字节的异型国标码 GB2312-80 规定的国标码的每个字节中最高位置“1”表示 通称机内码。

		B800H 段	
基本区	0000H	第 0 页	
	1180H	空 区	
	2000H	第 1 页	
	4000H	空 区	
	6000H	第 2 页	
	8000H	空 区	
扩展区	8000H	第 0 页	
	A000H	空 区	
	C000H	第 1 页	
	E000H	空 区	
		第 2 页	
		空 区	
		第 3 页	
		空 区	