

# CC-BIOS分析

钱 培 德 著

《微 计 算 机 应 用 》 编 辑 部  
中 国 科 学 院 声 学 研 究 所 情 报 室

1986年·北京

## 前 言

美国 IBM 公司于一九八一年推出了它的第一种个人计算机——IBM PC,以后又推出了它的增强型IBM PC/XT和IBM PC/AT等产品。IBM PC是用Intel 公司的8088和8087等芯片为核心开发和生产的一种微型计算机系统,它具有功能强、扩展灵活和使用方便等特点。而且它的结构设计先进,性能价格比高,具有强大的技术和经营后盾,故许多软件厂商都围绕着开发了大量的软件产品,使其成为世界上拥有软件种类最多的机种之一。因此, IBM PC 机在小型事务处理、办公室自动化、教育、通讯、控制和工程设计等许多领域中都得到了广泛的使用。它已成为当今国际微型机市场上最畅销的机种。它的兼容机已达数十种,这些兼容机在国际市场上也十分受欢迎,已成为微型机的主流产品。

IBM PC微计算机配有多个操作系统(PC-DOS, CP/M-86, UCSD-P)等,但它们均不具备汉字功能。我国于一九八三年推出了专为IBM PC机开发的汉字操作系统——CC-DOS,使得IBM PC及其兼容机具备了汉字功能,从而能普遍地被我国各部门所接受。CC-DOS是在PC-DOS的基础上开发的。目前,它已运行在国内绝大多数的IBM PC及其兼容机上。

为了更好地使用CC-DOS,为了进一步对IBM PC机的软件进行开发和汉字化工作,再加上CC-DOS使用的广泛性,所以,欲了解CC-DOS的内部结构及细节,已成为广大IBM PC微型机及其兼容机用户日趋迫切的要求。本人在研究了IBM PC系统的有关软硬件资料后,于今年二月份开始对CC-DOS进行全面的分析。分析的重点是CC-DOS实现汉字功能的部分——CC-BIOS。在分析中,得出了它的结构和工作流程。本书就是这个分析的总结。

CC-DOS自推出以来,经过了几次修改,但它的根本思想基本上没有变化。目前最流行的是其第二版本。CC-BIOS(V2.1)是用Intel 8088汇编编写的,共有5000余条指令。

本书把CC-BIOS(V2.1)按其功能分为几个部分。从键盘输入、显示输出、打印输出、引导过程等几个方面来进行分析。对其主要部分和重要程序模块,均给出了详细的流程图和注释,对其中的工作区也做了必要的说明。为便于读者阅读源程序,对文中所涉及到的全部程序模块和工作区,文章均给出了具体的地址。另外,在每部分的分析结束后,还给出了相应部分的源程序,并对源程序按其功能实现了分段。

本人在分析了CC-BIOS(V2.1)的全部内容后感到,CC-BIOS的设计和开发是成功的,它的结构紧凑,功能全面,设计独特,使用方便,它能反映出我国在汉字信息处理技术上的水平。本人认为,它不但具有使用上的价值,而且还是汉字信息处理技术方面的一个很好的范例,很值得从事这方面工作的设计和研究人员一学。另外,也适合作为大专院校中计算机专业学生的汉字信息处理教程的选讲内容。这些也是本人著此书的另一目的。

由于本书写作时间较仓促,及本人知识的限制,故文章中的缺点和谬误之处在所难免。本人诚恳地希望广大读者给予指正。在本书形成过程中,得到了《微计算机应用》编辑部的大力支持,谨在此深表谢意。

钱培德

1985年6月于苏州大学计算机教研室

# 目 录

<b>第一章 CC-DOS的概述和初步分析</b> .....	( 3 )
第一节 系统概述.....	( 3 )
第二节 CC-DOS的自举原理.....	( 3 )
第三节 CC-DOS中汉字显示输出的实现.....	( 4 )
第四节 CC-DOS中汉字输入的实现.....	( 6 )
第五节 CC-DOS中汉字打印输出的实现.....	( 9 )
<b>第二章 CC-BIOS 的引导过程</b> .....	( 13 )
第一节 概述.....	( 13 )
第二节 BIOS 的结构与修改.....	( 13 )
第三节 CC-DOS的自举过程.....	( 14 )
第四节 CC-BIOS 的引导过程.....	( 14 )
第五节 对 FILE1.EXE的分析.....	( 15 )
第六节 对CCCC.EXE 的分析.....	( 16 )
第七节 CC-BIOS 打印模块的形成.....	( 17 )
第八节 引导源程序.....	( 18 )
<b>第三章 CC-BIOS 的键盘管理模块</b> .....	( 20 )
第一节 引言.....	( 20 )
第二节 键盘中断的处理.....	( 20 )
第三节 键盘输入请求程序的结构.....	( 21 )
第四节 代码转换和扫描表.....	( 23 )
第五节 汉字输入功能的实现.....	( 24 )
第六节 读入字符处理模块.....	( 25 )
第七节 输入码处理模块.....	( 26 )
第八节 键盘管理模块的源程序.....	( 30 )
<b>第四章 CC-BIOS 的显示器控制模块</b> .....	( 54 )
第一节 总述.....	( 54 )
第二节 显示器控制模块的结构.....	( 54 )
第三节 几个重要的工作区.....	( 55 )
第四节 CRT 的初始化.....	( 57 )
第五节 光标的产生和定位.....	( 58 )
第六节 屏幕滚动的实现.....	( 60 )
第七节 读字符的实现.....	( 62 )
第八节 字符显示的实现.....	( 62 )
第九节 TTY 显示方式的实现.....	( 64 )
第十节 提示行管理模块.....	( 65 )

第十一节	显示器控制模块的源程序.....	( 67 )
<b>第五章</b>	<b>CC-BIOS 的打印驱动模块.....</b>	<b>( 92 )</b>
第一节	总述.....	( 92 )
第二节	打印屏幕驱动程序.....	( 92 )
第三节	打印机驱动程序的组成和输出字型.....	( 93 )
第四节	打印机驱动程序的工作区.....	( 94 )
第五节	打印机驱动程序的总体.....	( 95 )
第六节	17类中断程序的0号功能块 .....	( 96 )
第七节	图形态字符处理模块.....	( 98 )
第八节	打印驱动模块的源程序.....	( 100 )
<b>第六章</b>	<b>CC-BIOS 的高级打印驱动模块.....</b>	<b>( 111 )</b>
第一节	概述.....	( 111 )
第二节	高级打印驱动模块的生成.....	( 111 )
第三节	21类中断程序.....	( 112 )
第四节	17类中断程序.....	( 112 )
第五节	17类中断程序的0号功能块 .....	( 114 )
第六节	非ESC态处理模块.....	( 116 )
第七节	机内码处理模块.....	( 117 )
第八节	高级打印驱动模块的源程序.....	( 120 )
<b>参考文献</b> .....		<b>( 131 )</b>

# 第一章 CC-DOS的概述和初步分析

## 第一节 系统概述

CC-DOS 汉字操作系统已被广泛地使用在国内的 IBM PC 微型计算机及其兼容机上。CC-DOS是在PC-DOS(IBM PC微型机的主操作系统)的基础上设计的,它具有PC-DOS的全部功能,且增加了汉字功能。它的使用是与PC-DOS兼容的,可继续使用PC-DOS支持的各种软件。CC-DOS的字库存放在外存上,系统启动时一次调入内存,加快了汉字处理的速度。它具有多种汉字输入方式,向用户提供尽可能简便的操作方式。在输出方面也提供了多种选择。CC-DOS的使用不需要对IBM PC的原硬件系统进行改造,也不要增加新的硬设备,它是用软方案来实现PC-DOS的汉字化的。

在汉字处理方面,CC-DOS把汉字作为和西文一样的字符处理,汉字的容许度可以达到文件各级。在各种高级语言和应用程序中,汉字作为字符串,可以和西文混杂处理。CC-DOS的字库采用不压缩字模,每个字模32个字节,汉字点阵为 $16 \times 16$ ,字形为仿宋体。它的字库中共有6763个汉字和619个图形符号。

CC-DOS向用户提供多种输入方式,它们都有与用户进行对话的功能。用户可以方便地实现各种输入方式间的转换(通过ALT键和相应的功能键进行)。CC-DOS的汉字显示和打印输出采用图形方式。显示的汉字字符为 $16 \times 16$ 点阵。打印时可选择16种字型,同时可选择打印的行宽。配上高级打印模块后,还可打印出 $24 \times 24$ 点阵的高质量汉字。

CC-DOS是对PC-DOS的扩充,扩充了汉字输入输出功能。一般来说,涉及到汉字输入输出的只是操作系统中直接控制外部设备的那一部分。在PC-DOS中,直接控制外部设备的那一部分被称为BIOS。由于它被固化在系统板上的ROM中,故亦被称为ROM-BIOS。要为PC-DOS扩充汉字功能,只要对ROM-BIOS进行一些修改和扩充即可。经过这样修改和扩充后,就形成了CC-BIOS,它是CC-DOS中直接控制外部设备的部分。因此,我们要对CC-DOS进行分析,只要对其中的CC-BIOS进行分析即可。由此可知,CC-BIOS是CC-DOS中的精华。

## 第二节 CC-DOS的自举原理

CC-DOS自举的关键是由ROM-BIOS形成CC-BIOS。

ROM-BIOS是PC-DOS中的最底层软件。高层软件对外部设备的调用,是通过调用ROM-BIOS中的相应模块来进行的。而调用ROM-BIOS的相应模块是通过调用相应的软中断(10H类—1AH类)来完成的。软中断的响应,则是根据被调用的软中断的中断号,取中断向量表中的相应的中断指针,然后转去执行此中断指针指向处的程序。

CC-BIOS是CC-DOS的基本输入输出部分,它是由ROM-BIOS扩充而成的。主要是扩充了汉字处理部分,这部分存在RAM中(在CC-DOS自举时被引入)。CC-BIOS与ROM-BIOS

一样，也是由若干个外部设备控制模块组成的。它们完成对系统部件中主要 I/O 设备的控制，使得高层软件（包括系统程序和用户程序）均不要直接与外部设备打交道，并使得系统硬件的改进和扩充，对用户而言是“透明”的。对 CC-BIOS 中的模块的调用，也是通过 Intel 8088 提供的软中断来实现的，其参数由指定的 8088 寄存器传递。所以，这些模块又被称为软中断处理程序（简称中断程序）。

由以上可知，要把 ROM-BIOS 改为 CC-BIOS，只要把 CC-BIOS 中的模块引入内存，然后修改相应的中断指针值，使其指向相应的 CC-BIOS 的模块。以后通过软中断调用的就不再是 ROM-BIOS 中的相应模块，而是 CC-BIOS 中的相应模块。

CC-DOS 的自举过程与 PC-DOS 是一样的，先是由 ROM 中的引导程序把系统盘上的系统引导程序引入内存。然后此程序运行，把系统盘上的三个文件：IBMBIO.COM, IBMDOS.COM 和 COMMAND.COM 引到内存的指定区域。完成这些工作后就要检查系统盘上是否有 AUTOEXEC.BAT 文件存在，如存在则执行这个批命令文件。CC-DOS 的系统盘上就存在这个批命令文件，CC-DOS 自举时，就是由这个文件来完成把 ROM-BIOS 改为 CC-BIOS 的。另外，它还完成引入汉字库的工作。这个文件中有两条主要的命令，即 FILE 1 和 CCCC。前者所完成的工作是检查汉字库（CCLIB 文件）在系统盘上的完好性，再为汉字库进入内存预先申请好内存空间。后者所完成的工作是，把汉字库装入到申请好的内存空间中（由 FILE 1 申请的），再修改有关的中断指针，使它们指向 CC-BIOS 的相应模块。通过 AUTOEXEC.BAT 批命令文件的执行，就完成了 CC-DOS 自举的关键，形成了 CC-BIOS。

经过上面的分析和介绍，我们可以知道，要弄清 CC-DOS 是怎样完成汉字的输入输出功能的，只要对被修改了中断指针值的中断程序进行分析即可。这些中断是 5H 类、10H 类、16H 类和 17H 类。它们分别为 CC-BIOS 的打印屏幕驱动模块、显示控制模块、键盘管理模块和打印驱动模块。

下面对 CC-DOS 作一个初步的分析，主要分析其中实现汉字输入输出的部分。

### 第三节 CC-DOS 中汉字显示输出的实现

汉字的显示输出是由 CC-BIOS 中的 CRT 控制程序（即 10 类中断处理程序）实现的。实现汉字显示输出的过程，实际上是把汉字的机内码转换成汉字字模，把字模送屏幕输出的过程。

在 CC-DOS (V2.1) 中，汉字的机内码用国标码最高位置 1 表示（称为异形国标码），以区别于一般的 ASCII 码，故一个汉字的机内码用两字节表示。汉字字模存放在汉字库中，其存放位置也是按国标次序存放的。每个汉字字模用 32 个字节表示，即其点阵为  $16 \times 16$  的。这 32 个字节可看成 16 个字，每个字顺次表示汉字一行上 16 个点的情况。另外 CC-DOS 在显示汉字时必须工作在图形方式下，故 CRT 的字符发生器不发生作用，即使显示一般字符时，其字模亦要自己产生（其点阵为  $8 \times 8$  的）。这些字符字模存放在字符字模库中，该库已由 CCCC.EXE 带入内存。

#### 1. 有关的工作区

(1) 字模缓冲区 把获得的字模信息放到这里进行一定的排列，然后再送去显示。

它分为左半区和右半区，汉字的显示需用到两个半区，字符的显示只用到一个半区。其地址为CS:0078—0099。

(2) CRT刷新区 本区是存放屏幕显示信息的，它的每一个位(bit)与屏幕上的象元是一一对应的。改变了刷新区中某一位的内容，就改变了屏幕上相应象元的状态。其地址为B000:8000—9F3F(偶数扫描线区)和B000:A000—BF3F(奇数扫描线区)。

(3) 虚拟刷新区 其“虚”是针对CRT刷新区而言的。它是CRT屏幕所显示的字符内容的副本。虚拟刷新区分为三个部分(分区)，这三部分中的单元是一一对应的。这三个部分分别用以存放显示的字符、属性字和状态字。它们的地址分别为CS:00B0—087F(字符分区)，CS:0880—104F(属性字分区)，CS:1050—181F(状态字分区)。

## 2. CRT控制程序

CRT控制程序(即10类中断程序)由二十个功能块组成。在AH寄存器中置上功能块号(0—19)，再使用INT 10H指令就可以调用相应的功能块。以下对其中最常用的10号功能块进行分析和介绍。它的具体功能是把字符送CRT屏幕显示。在显示的字符中不包括下列特殊字符：回车(0DH)、换行(0AH)、退格(08H)和报警(07H)。对这些特殊字符的处理不包含在本功能块中(由14号功能块实现对这些特殊字符的处理)。本功能块的执行

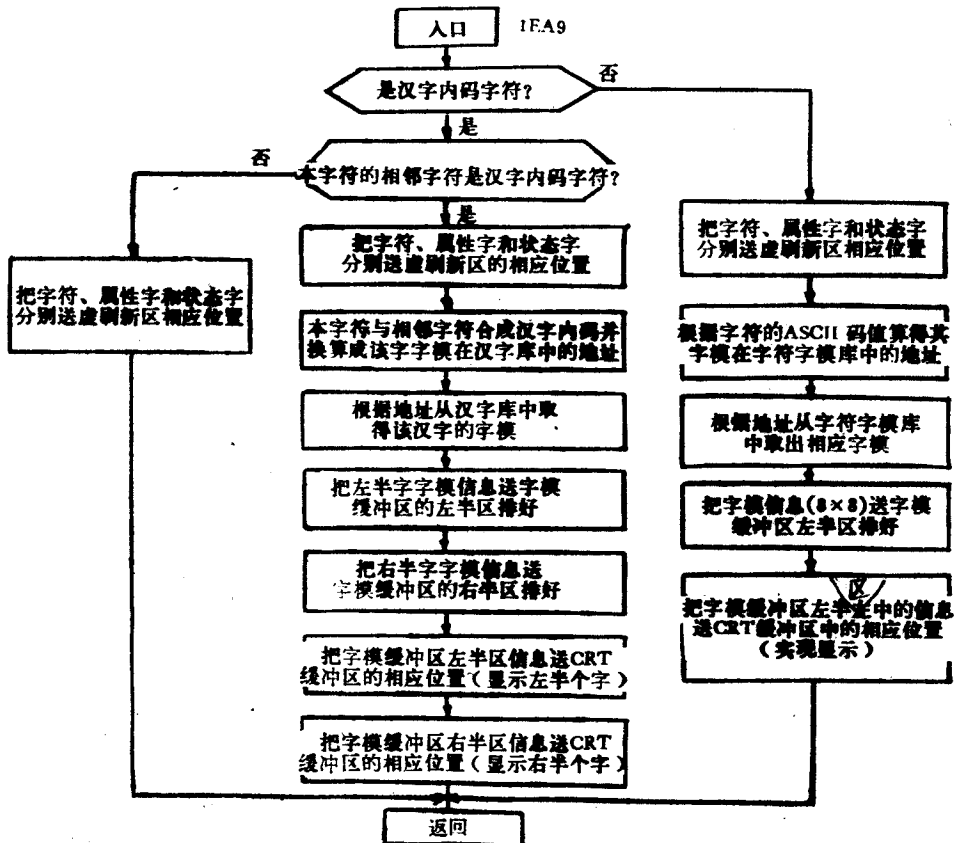


图1-1 10类中断10号功能块流程

流程如图1-1所示。

下面对图1-1中的某些部分作几点说明。

1) 判断一个字符是否是汉字内码字符, 即判其是否不于 A0H, 大于则为汉字内码字符。

2) 字符字模库的首地址就是 1F类中断指针的内容。这在 CCCC·EXE 执行时就置好了。

3) 判别一个字符的相邻字符是否是汉字内码符, 是通过判其相邻字符在虚拟刷新区中的状态字实现的, 状态字的内容可以反映出这一点。

4) 汉字库的首地址被保存在指定单元中 (CS:0075—0076) 这是在 CCCC·EXE 执行时保存的。

5) 把汉字字模信息送入字模缓冲区时, 为了要保持屏幕每行间的间隔, 故在最末要加送两行“空白”信息。把字符字模信息存入字模缓冲区时, 为了使字符与汉字保持一样的高度 (字符高为8, 汉字高为16), 故在其前与后要送几行“空白”信息。

6) 为了避免一个汉字的一半显示在一行末, 另一半显示在下一行首, 故程序中有行尾处理部分。当汉字显示的位置正好在一行末时, 则自动把其调整到下行首显示。

#### 第四节 CC-DOS 中汉字输入的实现

汉字的输入是由 CC-BIOS 中的键盘控制程序 (即 16 类中断程序) 来实现的。实现汉字输入的过程, 实际上是把由键盘输入的汉字输入码 (如区位码、首尾码、拼音码等) 转换成汉字机内码 (如异形国标码) 的过程。

与键盘有关的中断处理程序有两个。一个是 9 类中断程序, 当键盘上有字符输入时, 形成该字符的扫描码 (关于扫描码请参阅 IBM PC 机的随机资料) 和字符的 ASCII 码, 并把它们存入键盘缓冲区中。CC-BIOS 中仍用原来 ROM-BIOS 中的 9 类中断程序。另一个是 16 类中断程序, 当要取键盘输入的字符时, 就调用它。它的主要功能是把键盘缓冲区中的字符送给调用者。CC-DOS 对这个中断程序进行了较大的扩充, 使其能对输入的汉字输入码进行处理。

在 CC-DOS 中实现汉字输入码转换成机内码, 采用了两种方法。对有规律的输入码 (如区位码) 采用计算法, 对另外一些输入码 (如首尾码、拼音码等) 采用查表法, 也就是借助于一张输入码-机内码对照表 (简称扫描表), 这儿扫描表的每一个项对应于一个汉字, 表项的内容为其对应汉字的输入码。表项的排列是按国标字库顺序排的。当得到一个输入码后, 只要在扫描表中查找内容符合输入码的项, 根据得到的项的序号, 经过简单的计算即可得到其对应的机内码。

##### 1. 有关工作区

下面介绍几个与本程序有关的工作区。

(1) 键盘缓冲区 存放由键盘上输入的字符 (包括它的扫描码和 ASCII 码), 其中的内容由 9 类中断程序存放。地址为 0040:001E—003D。

(2) 汉字输入码缓冲区 存放已输入的汉字输入码字符。地址为 CS:2BB1—2BBA。



(3) 汉字内码缓冲区 存放由输入码转换成的汉字内码。地址为 CS:959D—959E。

(4) 汉字重码缓冲区 当输入码所对应的汉字不止一个时，则把这些汉字内码均存入此区（最多存10个内码）。地址为CS:2BBC—2BCF。

(5) 标志字节 它的不同值，对应于当前不同的工作方式。要知当前工作方式是什么（如拼音码方式、首尾码方式等），只要判此字节的内容即可。地址为CS:963A。

## 2. 0号功能块流程

CC-BIOS 中的16类中断程序也分几个功能块（共七个），其调用法与10类中断相同。

0号功能块是其中最主要的部分，它的功能是：从键盘缓冲区读一字符，并对其中的特殊字符（包括功能字符和汉字内码字符）进行处理，并返回相应的信息。对于一般字符，则直接返回其扫描码和ASCII码。也就是这一功能块完成了汉字输入码到汉字机内码的转换工作。

其流程如图1-2所示。下面对图1-2作几点说明。

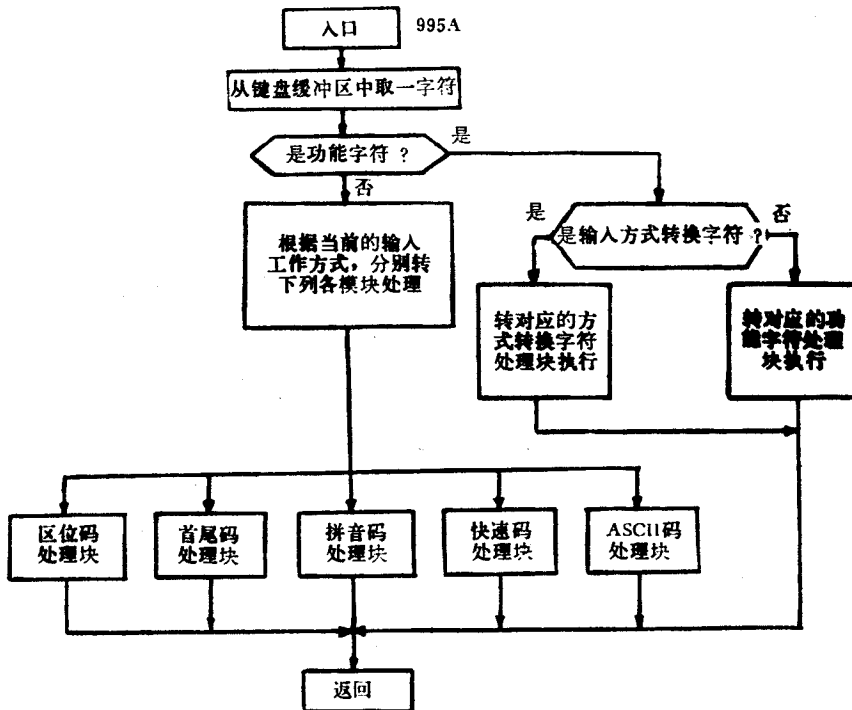


图1-2 16类中断0号功能块流程

1) 所谓功能字符，是指CTRL-F5至F10以及 ALT-F1至F6。其中 ALT-F1至F6 又称为转换字符。

2) CTRL-F5至F10这六种字符所表示的功能，请参阅CC-DOS 使用指南，这里不再重复。它们的相应处理模块就是完成它们所表示的功能。

3) ALT-F1 至F6是进行输入码方式转换，它们的相应模块所做的工作，就是在提示行

(屏幕最底下一行)中显示当前工作方式的名字和把标志字节置成与之相应的值。

4) 当前工作方式(即当前输入码方式)的判别,是根据标志字节的值确定的。

5) 各种输入码的处理模块就是完成把相应的输入码转换成机内码的工作。

### 3. 输入码的处理模块流程

下面介绍两种输入码的处理模块流程。一是区位码,二是首尾码,它们分别用计算法和查表法实现转换,有一定的代表性。

(1) 区位码处理模块的工作流程(入口为9DCB)

1) 对回车符(0DH)和退格符(08H)进行处理。

2) 如是字符“0”—“9”,则转下步执行,否则直接返回。

3) 把字符送入输入码缓冲区,当此缓冲区中的字符满四个后,即进行下一步,否则返回。

4) 把第一、二个字符转换成十进制BCD码形式作为区号;把第三、四个字符转换成十进制BCD码形式作为位号。

5) 把区号、位号转换成十六进制数(即形成国标码),再转换成异形国标码(机内码),然后返回。

(2) 首尾码的处理模块工作流程

本工作流程如图1-3所示。另外,快速码、拼音码等的处理与此相似。下面对图1-3作一些说明。

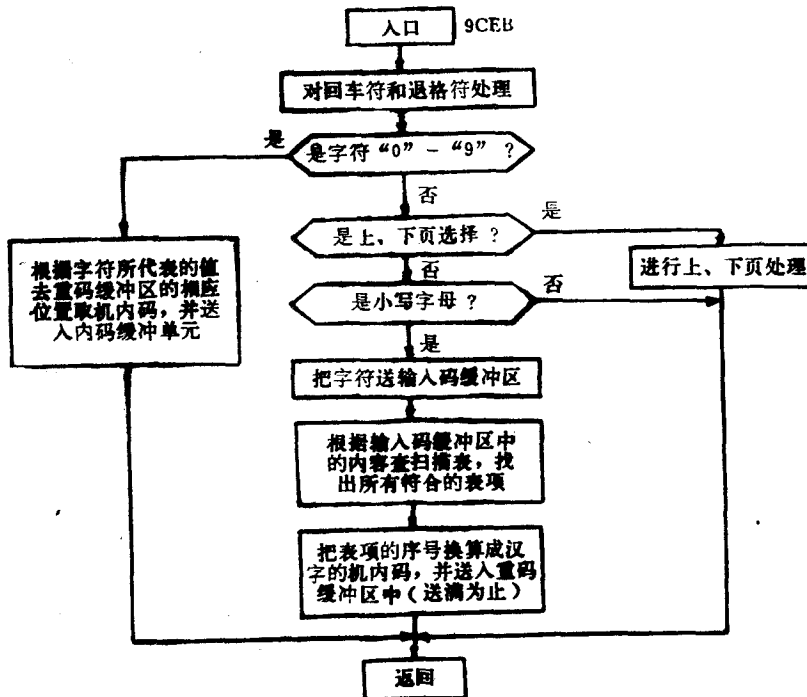


图1-3 首尾码处理流程

1) 首尾码是三字符码, 它是有重码的。也就是说, 针对一个输入码有多个汉字与它对应。CC-DOS对重码的处理是这样的, 把重码依次在提示行中显示出来, 让用户用数字键来选择其中的某个(这是CC-DOS的特色之一)。所以字符“0”—“9”表示选择键, 根据其值, 可在重码缓冲区的相应位置上取到用户所选汉字的机内码。

2) 上、下页选择是这样的, 提示行一次最多只能显示出十个重码汉字, 当重码汉字不止十个时, 则可用上、下页选择键来进行前后页的转换。对这种选择的处理, 就是把前十个或后十个重码汉字找出来送提示行显示, 并把它们的机内码存入重码缓冲区中。

3) CC-DOS规定, 首尾码、快速码、拼音码等输入码的字符须用小写字母表示, 如是大写字母, 则表示非输入码字符, 作一般ASCII码处理。

4) 输入码字符顺次存放在输入码缓冲区中, 其个数可能为一个、两个或三个。在查表处理时, 要根据输入码缓冲区中字符的个数来查找表项中最前面的相应个字符是否与输入码缓冲区中的字符相同。

5) 把表项序号转换成机内码是很容易的, 因为表项的次序是按国标字库序列排的。

## 第五节 CC-DOS中汉字打印输出的实现

用DIR命令, 可以看到在CC-DOS的系统盘上还有两个文件, 它们是 ALL9P·EXE 和 ALL24P·EXE。这两个文件是用来生成CC-DOS的17类中断处理程序(打印机控制程序)和5类中断处理程序(打印屏幕程序)的, 这两个中断程序组成了 CC-BIOS 的打印输出部分。CC-DOS就是靠这部分来实现汉字的打印输出的。因为这部分程序与系统配置的打印机有直接关系, 故CC-DOS给出了两个文件, 其中ALL9P·EXE用于九针打印机, ALL24P·EXE用于二十四针打印机。它们之所以没有被编入AUTOEXEC·BAT文件中, 是为了让用户有选择打印机的余地。一旦选定了打印机, 则可把其对应的文件编入 AUTOEXEC·BAT 文件中。以便在一开始就生成CC-BIOS的打印输出部分。下面以ALL9P·EXE为例来介绍和分析CC-DOS在九针打印机上输出汉字的实现方法。

### 1. 5类中断处理程序

用户调用打印机有两种目的, 一种是调打印机打印文件或运算结果, 另一种是调打印机打印当前屏幕上的显示内容。前者涉及到17类中断程序, 后者涉及到 5类中断程序。现先对5类中断程序进行分析。

CC-BIOS的5类中断处理程序较简单, 它的执行过程如下: (入口为CS:1FD3)

首先判CRT的工作方式。如是字符方式则直接去执行ROM-BIOS 中原来的5类中断处理程序(入口为F00:FF54), 这段程序所做的工作就是把CRT刷新区中当前页的内容, 逐个字符送打印机输出; 如果CRT的工作方式是图形方式, 则向17类中断程序发送ESC“W”控制代码, 由17类中断程序中的相应模块去完成把屏幕上的内容送打印机输出的功能(这部分内容在下面17类中断程序中介绍)。然后进行中断返回。

### 2. 17类中断处理程序

CC-BIOS的17类中断处理程序共有四个功能块组成, 其中最主要的是0号功能块。它的

功能是把字符送打印机输出。实现汉字打印输出的部分，就包含在这个功能块中。

(1) 有关的工作区

1) 打印数据缓冲区(简称缓冲区): 用来存放要送打印机输出的打印数据,并使用它对字模点阵重新排列处理。在图形方式下(打印汉字就是在此方式下进行的)要用到这缓冲区。它共分成四个部分,即缓冲区1、缓冲区2、缓冲区3、缓冲区4。每区均为816个字节。下面介绍这四个区的作用。

缓冲区1(地址为CS:00E2—0741)和缓冲区2(地址为CS:0742—0DA1)用以存放要进行纵向扩展的汉字上半部的打印数据。

缓冲区3(地址为CS:0DA2—1401)和缓冲区4(地址为CS:1402—1A61)用以存放要进行纵向扩展的汉字下半部的打印数据。对于不进行纵向扩展的汉字,缓冲区3存放汉字上半部的打印数据,缓冲区4存放汉字下半部的打印数据。对于一般字符,缓冲区4存放字符的打印数据。

2) 属性工作区: 存放要打印的字符的属性字(它表明该字符是一般图形打印,还是紧密图形打印)。它共102个字节,每一个字节对应于缓冲区中的八个字节(因一个字符有八个打印数据),即指明这八个字节的打印属性。地址为CS:0016—00E1。

3) 状态字: 用于表示打印的几种状态,比如:是否紧密打印,是否横向扩展,是否纵向扩展,是否为横向字,前一字符是否是ESC,前两字符是否是ESC“1”等等。根据此状态字可以判断出,当前字符是否是控制代码,以及要打印的汉字为何种字型(CC-DOS V2.1提供十六种字型)。它的地址为CS:000F—0010。

(2) 0号功能块的流程

0号功能块的工作流程如图1-4所示。下面对图1-4中的有关部分作些说明。

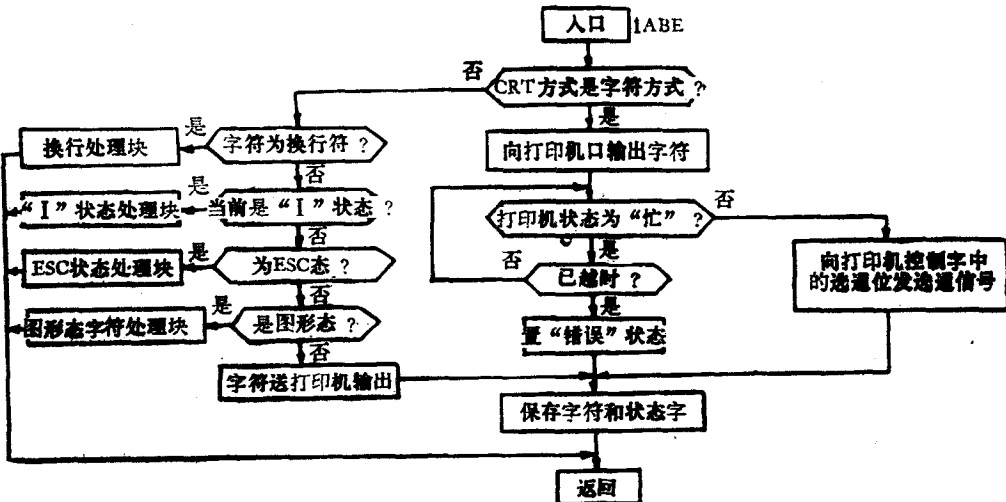


图1-4 17类中断0号功能块流程

1) 向打印机输出字符时,没有采用中断方式,而是采用查询方式。故要从打印机状态口取状态字,如打印机为“忙”,则等待,否则向其控制口发选通信号,从而完成输出一个字符。

2) 换行处理块完成以下工作, 判别缓冲区是否为空, 如为空, 则进行行间距为8/72 英寸的换行 (8/72英寸正好为半个汉字的高)。否则把缓冲区中的内容全部送打印机输出。

3) I状态处理块的功能是改变汉字的字型。在16类中断程序中, 向用户提供了一个改变打印的汉字字型的块能, 实际上就是由16类中断程序向17类中断程序发出一个 ESC “I” 控制代码, 后跟一个字母 (A—P) 代表字型的选择 (共十六种)。I 状态处理块就是对 ESC “I” 控制代码进行解释, 它根据选择的字型, 分别对状态字中的有关位置以相应的值。

4) ESC态处理块的功能是对一些控制代码进行处理。这里主要是处理由5类中断程序发来的ESC “W” 控制代码 (详见前面5类中断程序部分), 即实现把屏幕上的内容在打印机上打印出来。本处理块在判到有ESC “W” 控制码时, 就把CRT刷新区中的内容转换成打印数据送打印机输出。

5) 图形态字符处理块的功能是把字符或汉字的字模信息转换成打印数据前面已说过, 字库中的字模信息是以行表示的, 即每个字节代表字符横向的点的情况, 这适合于 CRT 显示。而打印机正好相反, 它要求字模信息以列表表示, 即每个字节代表字符纵向的点的情况 (这样的字节中的数称为打印数据), 所以, 必须把字模的横向信息转换成纵向信息后才能向打印机输出。本处理块的工作流程如图1-5所示。现对图1-5中的内容作一些说明。

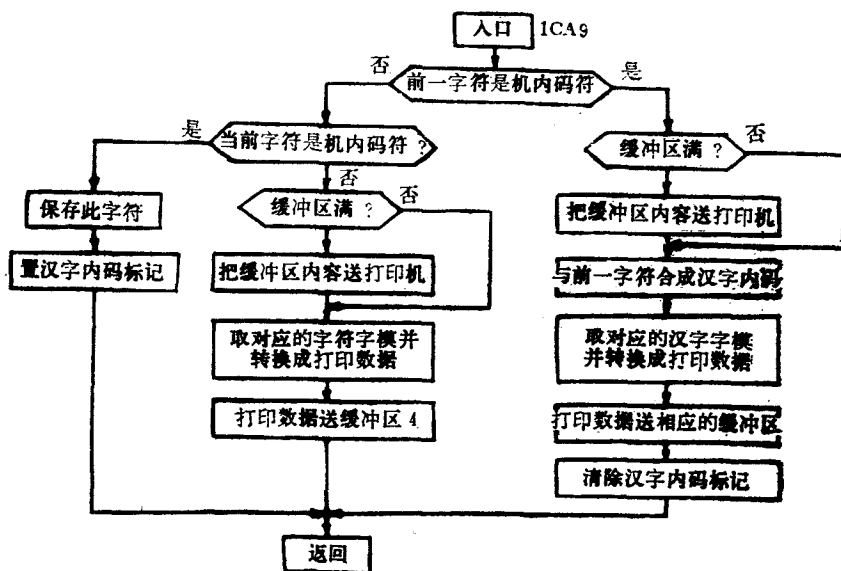


图1-5 图形态的字符处理流程

( I ) 判前一字符是否是汉字内码字符, 只要判汉字内码标记即可。

( II ) 判断缓冲区是否满, 并不是判缓冲区是否被全部装满, 而是判缓冲区是否能再装下当前要放入的全部打印数据 (如字符、一般汉字和横向扩展汉字的打印数据个数是不一样的), 如果不能全部装入, 即使缓冲区有空余字节, 也算缓冲区满。只有这样才能确保打印输出的内容的正确和完整。

( III ) 把汉字的打印数据送相应缓冲区, 要做以下工作: 根据指定的字型 (由状态字内容可知), 对打印数据进行横扩、纵扩、横向等处理。然后, 当是纵向扩展时, 则把汉字的上

半部打印数据送缓冲区1和2，把汉字的下半部打印数据送缓冲区3和4。当是非纵向扩展时，则把汉字上、下半部的打印数据分别送缓冲区3和4。

以上只是对CC-DOS操作系统进行了一个初步的分析，其目的旨在使大家对它有一定的认识，为下面对CC-BIOS的深入分析作准备。

以下各章中所附源程序的段值，除注明外一律为0984。

## 第二章 CC - BIOS的引导过程

### 第一节 概 述

CC-DSS 的汉字功能的实现,一般只与直接驱动和控制外部设备的这部分程序有关。IBM PC的系统板上有一个8K的ROM,里面装的是系统配置之外部设备的控制程序,被称为ROM-BIOS。它是IBM PC系统软件的最底层,其他软件中如要调用外部设备,则通过调用ROM-BIOS中的相应模块来实现。要使外部设备能输入和输出汉字,只要对ROM-BIOS进行修改,把它改为CC-BIOS(CC-DOS中的BIOS)。那末,怎样把ROM-BIOS改成CC-BIOS呢?这个修改的过程,就是CC-BIOS的引导过程。

### 第二节 BIOS的结构与修改

为了讨论怎样修改BIOS,首先简单地介绍一下BIOS的结构。

BIOS是由若干个独立的设备驱动模块组成的,每个模块对应于一种设备,即为该种设备的控制(驱动)程序。每个模块的入口地址被存放在系统的中断向量表中(地址为0000:0000—03FF)。每个地址均由段值和偏移量组成,占四个字节。这四个字节的内容被称为中断向量。对BIOS中的模块的调用,是通过软中断(10类—1A类)来实现的,故这些模块亦被称为软中断处理程序(简称中断程序)。

要改变BIOS中某模块的内容,我们采用“贴补代替”的办法。即把该模块在中断向量表中的相应中断向量改为新模块的所在地址即可。经这样修改后,当用相应的软中断来调用该模块时,则不再执行原来模块中的程序,而转到新的模块之入口执行。但是,新的模块应放在什么地方呢?在CC-DOS中,新模块是放在RAM中的。也就是说,CC-BIOS的程序分布在ROM和RAM中。以上就是改ROM-BIOS为CC-BIOS的基本方法。IBM PC系统向我们提供了采用这种方法来修改BIOS的可能性,因IBM PC的内存容量较大,故新的BIOS模块占用一部分内存,关系不大;另外IBM PC系统提供了使程序常驻内存的系统功能调用,使得新的BIOS模块在被引导进内存后,可以常驻内存而取代相应的ROM中的部分。

另外,在把ROM-BIOS改为CC-BIOS的过程中,并不要修改原来所有的模块,而只要修改与汉字输入输出有关的模块。这些模块如表2-1所示。

表21 与输入输出有关的BIOS模块

中断类号	实 现 功 能	中断类号	实 现 功 能
5	打印屏幕内容驱动程序	16	键盘管理程序
10	CRT控制程序	17	打印机驱动程序

### 第三节 CC-DOS的自举过程

CC-BIOS的引导过程是包含在CC-DOS的自举过程中的，让我们先来看一下CC-DOS的自举过程。

当系统启动或复位时，则自动执行 ROM中的引导程序，该程序把CC-DOS系统盘上0道1扇区中的引导记录读入内存，并使它开始执行。引导记录对系统盘进行一些正确性的检查，然后把系统盘上的 IBM bio.sys 文件和 IBM dos.sys 文件读到内存的指定区域。

首先进入内存的是 IBM bio.sys，它先进行初始化处理，然后向下重新定位 IBM dos.sys，并把控制交给它。IBM dos.sys 对有关的工作表、缓冲区和向量等进行初始化处理，并为COMMAND.COM建立程序段前缀，然后返回IBM bio.sys。IBM bio.sys把COMMAND.COM装入确定的内存位置，然后把控制交给COMMAND.COM。

COMMAND.COM查找系统盘上是否有 CONFIG.SYS文件存在，如存在则执行之。然后执行 AUTOEXEC.BAT文件（CC-DOS系统盘上一定存在此文件）。最后显示操作系统的提示符，等待用户输入命令。CC-DOS的整个自举过程到此结束。图 2-1给出了整个自举过程的流程。

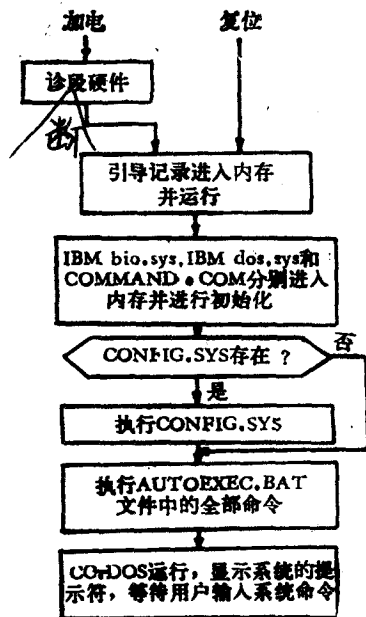


图2-1 CC-DOS自举流程

### 第四节 CC-BIOS的引导过程

让我们先了解一下CC-DOS系统盘的目录内容，其目录清单如下：

Volume in drive A is ::WV05M3

Directory of A:\

IBM bio	sys	4736	10-20-83	12:00p
IBM dos	sys	17024	8-08-84	12:00p
COMMAND	COM	17792	8-14-84	12:00p
CCCC	EXE	44356	1-01-80	12:25a
FILE1	EXE	768	7-14-84	12:00p
LOAD	EXE	768	7-14-84	12:00p
ALL24P	EXE	3456	9-05-84	1:08a
D320	EXE	3712	7-14-84	12:00p
ANSI	SYS	1664	7-14-84	12:00p
CONFIG	SYS	45	10-23-84	12:00p
AUTOEXEC	BAT	59	8-14-84	12:00p



CCLIB		237632	9-15-84	9:23a
ALL9P	EXE	8848	9-22-84	12:00p
NOF	COM	60	7-25-84	12:00p
9P	EXE	8960	8-20-84	3:12p
SYS	COM	1680	10-20-83	12:00p
		16 File(s)	0 bytes free	

从上面的清单中我们可以看到,系统盘上存在一个AUTOEXEC.BAT文件,CC-DOS自举时要执行这个文件。AUTOEXEC.BAT文件为批命令文件,它的内容为若干条命令或可执行文件名,当该文件被执行时,就把这些命令或可执行文件,逐个递交给系统执行。CC-BIOS的引导,就是利用这个文件来进行的。这个文件中存放了引导CC-BIOS的命令,其具体内容如下所示:

```

echo off
cls
echo PLEASE WAIT
file1
cccc
verify on

```

在这批命令中,前三条命令所完成的工作就是在屏幕的左上角显示“PLEASE WAIT”字样。最后一条命令是开放校验。值得我们注意的是 file1 和 cccc 两条命令。从上面给出的系统盘目录清单中我们可以知道,这两条命令的对应文件分别为 FILE1.EXE 和 CCCC.EXE。下面我们对这两个文件进行分析,分析它们的结构和它们在CC-BIOS的引导过程中所完成的功能。

### 第五节 对FILE1.EXE的分析

FILE1.EXE 的功能是做好装入汉字库之前的准备工作。具体完成了下列工作:检查汉字库(即CCLIB文件)在系统盘上的完好性,为汉字库申请好内存。其执行流程如图2-2所示。

从图 2-2 中可以看到,当汉字库(CCLIB)不完好时,则要准备再次引导系统。这实际上是这样来完成的:先显示再次引导的提示信息,然后等待用户从键盘输入信息,一旦用户输入一字符(任意字符),则通过软中断19来调用 ROM-BIOS 中的系统引导程序,执行图2-1的流程,重新引导系统。

从图2-2中还可以看到,当本命令结束时,它并没有正常退出(即交还占用的资源),而是驻留内存退出运行,也就是说,它没有交还占用的内存资源(包括结束前申请的那部分),这部分内存就不会被系统分配出去,留

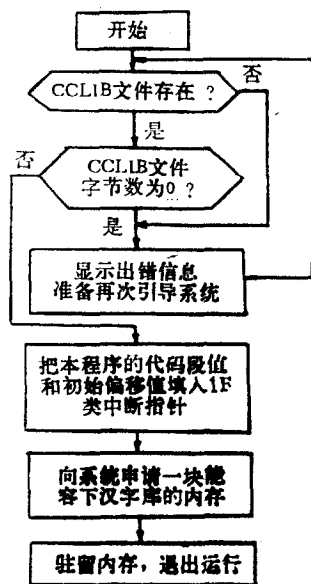


图2-2 FILE1.EXE的流程