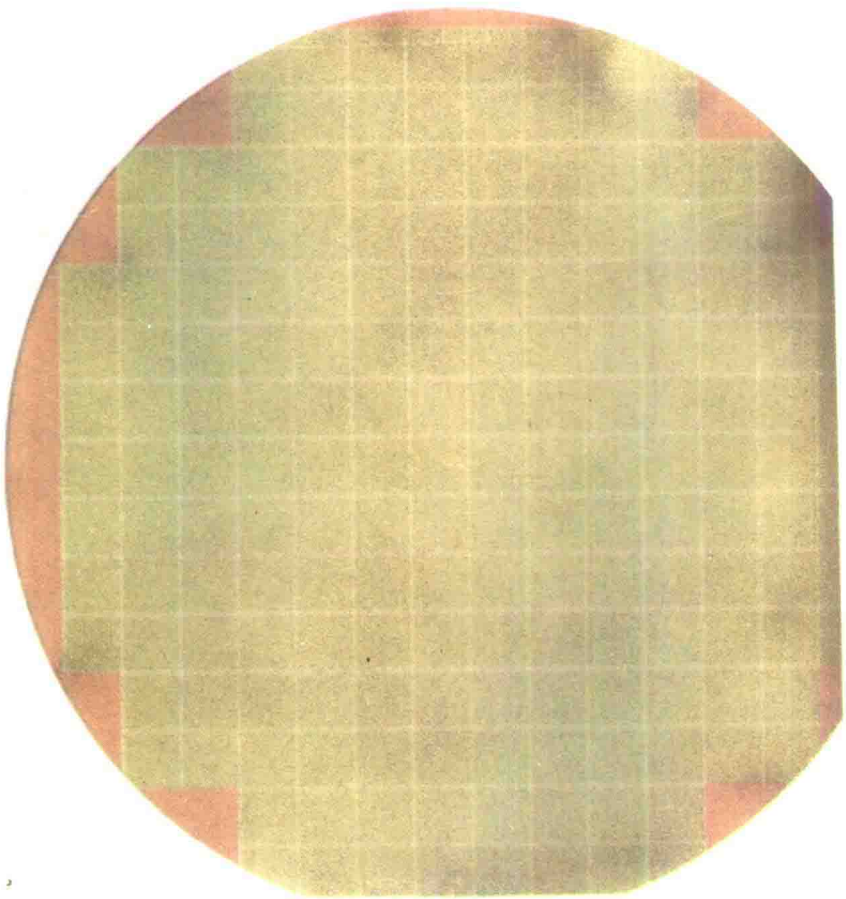
计算机
高级打印
驱动软件分析

唐建飚　唐登斌　编著



科学普及出版社

计 算 机
高级打印驱动软件分析

唐建飚 唐登斌 编著

科 学 普 及 出 版 社

内 容 提 要

计算机已成为当今高科技之一。在我国，要实现计算机的普及和应用，必须解决其输入输出的汉字化问题。该书全面、具体地讲述了计算机高级打印驱动软件的原理、性能和应用。

全书共分六章，前二章介绍了 DOS 的基础知识，后四章详细介绍了高级打印驱动软件的原理与框图，软件的分析、修改、扩充及使用。读后可使读者对该系统有一个清晰的认识，从而提高计算机编程能力和应用水平。

本书适合于从事微机系统软件开发和应用的科技人员阅读。

（京）新登字 026 号

计算机
高级打印驱动软件分析
唐建刚 唐登斌 编著
责任编辑：张 楠
封面设计：王序德
技术设计：郑爱华
科学普及出版社出版（北京海淀区白石桥路 32 号）
新华书店北京发行所发行 各地新华书店经售
北京市燕山联营印刷厂印刷
开本：787×1092 毫米 1/16 印张：6.375 字数：162 千字
1993 年 5 月第 1 版 1993 年 5 月第 1 次印刷
印数：1—3 200 册 定价：6.70 元
ISBN 7-110-02674-4/TP・49

前　言

PC 机目前在我国广为普及，随着硬件技术的发展，286 机、386 机以及 486 机呈现逐年增长的势头。但由于这些机器本身所配的操作系统是西文操作系统，不具备汉字处理功能，因此要在我国被大家普遍接受，必须对其汉化。而操作系统汉化的主要内容之一，就是打印驱动软件的汉化，以实现汉字的打印输出。本书就是对其分析的总结。

目前打印驱动软件种类繁多。这是因为不同型号的打印机需要不同的打印驱动软件来支持。本书不可能包括所有的打印驱动软件，只能就此之一进行分析。不过各种打印驱动的设计思想和基本结构相差不多，差别仅在于打印机的某些控制命令不同，因此读者通过本书的学习，不难将其修改为适合自己型号的打印机。

当前，分析各种中文操作系统的书籍不少，但就其中之一认真剖析、画出流程详细讲解的还为数不多，本书暂且作为补充之一。

本书分析的是支持 16 点阵、24 点阵、48 点阵高级打印驱动软件。虽然文件不长，但是其功能强、字体丰富、支持打印机的各种命令、运行可靠，从而赢得了用户的高度赞扬。为了进一步使用好该软件，有必要加以详细分析，然后进行改造、扩充新的功能或亲自动手编制新的高级打印驱动软件，以满足工作需要。

全书共分六章，前二章分别介绍了 CCDOS 和 PCDOS 的基础知识，这为不熟悉 DOS 的读者或初学者作一些知识准备；后三章则是对高级打印驱动的详细分析；最后一章则是高级打印驱动软件的使用说明。

由于编者水平有限，加之编写时间仓促，书中难免有缺点和错误，恳请读者批评指正。

编　者

1992 年 2 月

目　录

第一章　CCDOS 的概述

1.1　CCDOS 的形成

CCDOS 的作用是解决汉字的输入输出和机内码处理问题。它是在 PC-DOS 的基础上设计的。它通过修改 PC-DOS 中的 ROM-BIOS 中的中断向量表后形成 CCDOS 的核心 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（10 类～1A 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，取中断向量表中相应的中断指针，然后转去执行此中断指针指向的程序。

CC-BIOS 是 CCDOS 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，这部分存放在 RAM 中。CC-BIOS 与 ROM-BIOS 一样，也是由若干个外部设备控制块组成。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，对用户来讲是“透明的”的。对 CC-BIOS 中的模块调用，同对 ROM-BIOS 中模块调用一样，也是通过 Intel 8088（或 80×86）提供的软中断方案来实现的，其参数由指定的 8088（或 80×86）寄存器来传送。

由上可知，CCDOS 是由 PC-DOS 修改其中的 ROM-BIOS 后形成的；因此，CCDOS 的形成既不需要对 IBM-PC 系列微型机硬件系统进行改造，也不要增加额外的硬件设备来实现。它完全是由软方案来实现 PC-DOS 的汉化的。

CCDOS 有多种输入方式，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）。汉字的显示和打印输出采用图形方式。显示的汉字字符为 16×16 点阵，打印时可选择各种字型，同时可选择打印的行宽。配上高级打印驱动软件后，不仅可以打印 16、24 点阵的汉字，而且还可以打印 48 点阵甚至更高点阵的高质量汉字。

1.2　CCDOS 中汉字输入和显示输出的实现

一、汉　字　输　入

CCDOS 中汉字输入是由 CC-BIOS 中的键盘管理程序（即 16 类中断程序）来实现的。实现汉字输入的过程，实际上就是把键盘输入的汉字输入码（如区位码、五笔字型、拼音码等）转换成汉字机内码的过程。

与键盘有关的中断处理程序有两个：一个是 9 类中断程序（硬中断处理程序）。当键盘上有字符输入时，形成该字符的扫描码和该字符的 ASCII 码，并把它们存入键盘缓冲区中。CCDOS 中仍用原来的 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的字符送给调用者。CCDOS 对这种中断程序进行

了较大的扩充，使其能对输入的汉字输入码进行处理。

在CCDOS中实现汉字输入码转换成机内码，通常采用两种方法。对有规律的输入码（如区位码、国标码等）采用计算法，即经过简单的计算，就可将输入码转换成机内码。对另外一些输入码（如五笔字型、拼音码、首尾码等）采用查表法，也就是借助于计算机内存内的一张输入码同机内码对照表（简称扫描表），来实现汉字的输入码转换成机内码的。在这里扫描表的每一个项对应于一个汉字，表项的内容为其对应的汉字输入码。表项的排列是按国际字库顺序排列的。当得到一个输入码后，只要在扫描表中查找同输入码内容相同的项，根据得到项的序号，经过简单的换算即可得到对应的机内码。

二、汉字显示输出

CCDOS中汉字显示输出是由CC-BIOS中的CRT控制程序（即10类中断程序）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库中的地址，然后由该地址取出该字模送屏幕输出的过程。

在CCDOS中，汉字的机内码用国标码最高位置1表示，以区别于一般的ASCII码，一个汉字的机内码需用两字节表示，故两字节最多表示2^{14}个汉字。汉字字模在汉字库中，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16×16。这32个字节可看成16个字，每个字顺序表示在屏幕上汉字一行上的16个点的情况。另外CCDOS在显示汉字时必须工作在图形方式，因而，彩色（单色）图形适配器上的字符发生器不发生作用，即使在图形方式下显示一般ASCII码字符时，其字模也要自己产生（其点阵为8×8）。这些字符字模存放在字符字模库中（该库在内存RAM中），该库由系统自举时带入内存并贮留。

1.3 CCDOS中汉字打印输出的实现

CCDOS中汉字的打印输出是由CC-BIOS中的17类中断程序（打印管理程序）和5类中断程序（打印屏幕程序）来实现的。实现汉字打印输出的过程，实际上就是将汉字机内码转换成汉字字模库中的地址，然后由该地址取出字模将横向点阵变为纵向点阵送打印机输出打印。这个过程是16点阵汉字输出打印过程，在24点阵和48点阵汉字的输出过程中，由于其信息量太大，不能存放在内存，只好将其存放在外存（IBM-PC系列为硬盘），其点阵信息仅供打印驱动软件使用，不提供给CCDOS显示用，因此其点阵信息格式是纵向排列的，而不是横向排列的，这样做的目的可以将其点阵信息直接作为打印数据送打印机输出打印，免去横向点阵、纵向点阵间转换的麻烦。实现高点阵（24点阵以上）汉字打印输出的过程是：将汉字机内码转换成汉字字模库中的记录号，然后在打开的当前文件中读取该记录的点阵信息到磁盘信息交换区，再从磁盘信息交换区中将点阵信息送打印机输出打印。

第二章　DOS 中 COM 文件、EXE 文件和程序段前缀 PSP 介绍

DOS 中共有两种可执行文件，其后缀分别为.COM 和.EXE，为了分析高级打印驱动软件，有必要先介绍一下这两种文件结构。

2.1　DOS 中 COM 文件、EXE 文件结构

一、COM 文件结构

一个 COM 文件的结构定义如下：

1. 该程序只能设置一个段，且不允许建堆栈段；

2. 该程序的长度必须小于 64k 字节；

3. 该程序必须先预留 100H 的空间，且在位移 100H 处是一条可执行的指令，而不能是数据；

4. 该程序被加载的起始标号必须由 END 语句说明为开始地址；

5. 若 COM 文件是由几个不同的目标模块连接生成的，则要求所有目标模块必须具有同一代码段名和类别名（CLASS）且赋予公共属性（PUBLIC），而主模块应具有 100H 的入口指针并优先连接；

6. 该程序中的子程序必须具有近过程属性（NEAR）。

符合上述条件的源程序，经汇编、连接而生成的 EXE 文件，可由外部命令 EXE2BIN 转换成 COM 文件。

二、EXE 文件结构

一个 EXE 文件结构如下：

1. 该程序允许建若干个不同名的代码段或数据段、附加段、堆栈段；

2. 该程序的长度仅受当前内存可用空间的限制；

3. 该程序的入口随应用而定，只需起始标号与 END 语句说明的起始地址一致即可；

4. 该程序中的各个子程序的属性既可为 NEAR，也可以为 FAR，它们随段内或段间调用而定。

5. 由几个不同的目标模块连接生成的 EXE 文件，可按应用要求将每一模块内代码段、数据段或附加段取同名或独立命名。但要保证，其中只有主模块的 END 语句指出程序入口的起始标号，并至少有一个具有 STACK 属性的堆栈段。

对于 EXE 文件的这种结构，连接程序 LINK 将根据被连接的目标模块的不同连接参

数，要相应地生成一个“重定位信息块”，并将其安装在程序的最前头（俗称“文件头”）。该文件头的大小依程序加载时需重定位的段的指令条数而变化。通常是512字节的整数倍。换言之，至少占1个扇区单位的长度。下面我们来介绍文件头这512整数倍字节的具体含义。（在我们分析高级打印驱动软件中该“文件头”是512字节。）

16进制偏移量	内　　容
00～01	4DH，5AH——这是EXE文件的有效标志
02～03	映象长度以512为模（装入模块的映象长度除以512的余数）
04～05	文件的大小，增加量为512字节（页）包括文件头
06～07	重定位表的项数
08～09	头的大小，增加量为16字节（段落），用于查找文件中装入模块的起始位置
0A～0B	在装入程序的末端之上，16字节段落的最小需要数
0C～0D	在装入程序的末端之上，16字节段落的最大需要数
0E～0F	装入模块中堆栈段的以段落表示的偏移量
10～11	赋予模块控制权时，SP寄存器中的偏移量
12～13	文件所有字的负累加和
14～15	给予模块控制权时，IP寄存器中的偏移量
16～17	装入模块中代码段的段落表示的偏移量
18～19	文件中第一个重定位项的以字节表示的位移量
1A～1B	覆盖号（0是程序的驻留部分）

还有可变保留区、重定位表（该表由18～19H指出）、可变保留区……

2.2 COM文件和EXE文件之间的联系与区别

一、COM文件和EXE文件之间的联系

1. 两种文件都是在DOS环境下的可执行文件；

2. 操作系统在运行这两种文件时都形成100H字节的程序段前缀控制块PSP；

3. 在一定的条件下，EXE文件可以转化为COM文件，也就是说EXE文件在满足COM文件中的六个条件，可由外部命令EXE2BIN转化为COM文件。

二、COM文件和EXE文件之间的区别

由于COM文件和EXE文件之间的结构不同，因此它们有许多不同之处。

1. COM文件的长度限制为一个段长（64k字节），且在加载过程中没有重定位，因此，结构紧凑、装入速度快。它适宜于小型程序。

2. EXE文件不受64k字节的限制，仅受当前可用内存空间的限制，在加载过程中需要段重定位。因而，占用盘空间大，装入速度慢。它适宜中、大型程序。

3. COM文件和EXE文件加载方式不同。

COM文件存于盘上，不附加任何定位信息，因此，只需在当前可用内存空间的最低端建立一个相应的程序段前缀PSP，然后紧靠PSP的下方将COM文件装入，并把控制转到

PSP+100H 处，即可执行 COM 文件。这就说明了 COM 文件为何要预留 100H 空间的原因，并在位移 100H 处必须有一条可执行指令。

EXE 文件，由于在文件头有重定位信息，因此，加载时，在当前内存空间的最低端建立一个相应的程序段前缀，还要依据一个“文件头”指出的若干信息进行重定位；同时，对各个内部寄存器赋以与 COM 文件被加载时不同的初始化值，最后从 DOS 系统中接过控制权执行程序。

2.3 程序段前缀控制块 PSP

当一个外部命令或程序被加载时，COMMAND 确定当前内存可用空间的最低端作为程序段起点。在该起点首先建立一个程序段前缀控制块 PSP。其结构如下：

字段位移（十六进制）	字段含义	字段长度（十进制）
PSP+0→	程序终止退出指令 INT20	2
+2→	可用的内存空间高端段址	2
+4→	保　留	1
+5→	DOS 系统功能远调用入口	5
+OA→	程序结束处理中断向量原内容	4
+OE→	Ctrl-C 处理中断向量原内容	4
+12→	严重错误处理中断向量原内容	4
+16→	保　留	22
+2C→	环境块段地址	2
+2E→	保　留	46
+5C→	格式化未打开的文件控制块 1	16
+6C→	格式化未打开的文件控制块 2	20
PSP+80→	未格式化的参数（+80 字节含有存放的参数长度）或磁盘的传输区	128

注：被加载程序使用的参数是放在 PSP+82H 开始的单元中，+80H 处为该程序使用参数的个数（长度）。（其中包括定界符空格 20H）。

第三章　高级打印驱动软件的原理与框图

3.1　高级打印驱动软件的原理

一、一般打印驱动软件原理

由第一章内容可知，字符的输出打印，与 ROM-BIOS 的 17 类中断和 5 类中断有关；但西文 DOS 中的 ROM-BIOS17 类中断和 5 类中断不具备汉字的打印输出，怎么来解决这个问题呢？我们通常采用“修改贴补”的办法来解决。为了讨论怎样修改 ROM-BIOS，首先简单介绍一下 ROM-BIOS 的结构。

ROM-BIOS 是由若干个独立的设备驱动模块组成，每个模块分别对应于一种 I/O 设备，此模块即为该类设备的控制驱动程序。ROM-BIOS 中每个模块的入口地址均被放置在系统的中断向量表中，其地址为 0000：0000～03FFH。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，故这些模块也被称为“软中断处理程序。”

要改变 BIOS 中某模块的内容，我们采用“修改贴补”的办法。即把该摸块在中断向量表中的相应中断向量改为新模块所在地址。经这样修改后，当用相应的软中断来调用该模块时，就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，新模块放在什么地方呢？在 CCDOS 中，是存放在内存 RAM 中。这样，CC-BIOS 的程序分布在 ROM 和 RAM 中。以上介绍的是修改 ROM-BIOS 为 CC-BIOS 的基本方法。IBM-PC 系列微机向我们提供了采用这种方法来修改 BIOS 的可能性。因为 IBM-PC 系列的微机内存容量较大（一般有 640k 字节左右）所以新的模块占用一部分 RAM 关系不大。另外系统提供了使程序常驻内存的 31 种功能调用和 7 类中断，使新模块在被引导内存后，可以常驻内存而形成 CC-DOS。即使初始化工作结束后，也不释放内存分配块。

此外，在把 ROM-BIOS 改为 CC-BIOS 的过程中，并不要修改原来所有的模块，而只需修改与汉字输入输出有关的模块。这些模块如下所示。

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

注：与汉字（或字符）打印输出有关的是 5 类中断程序和 17 类中断程序，这两个驱动程序也就是本书将要重点分析的部分。

二、高级打印驱动软件的工作原理

高级打印驱动软件（本书分析的是 H3070A. EXE）由这样四部分组成：本身执行代码

程序、17 类中断程序、5 类中断程序和其定义的 80 类中断程序。其中 80 类中断程序同操作系统提供的 21 类中断程序完全一样，其调用方法、入口功能号安排都同 21 类中断程序，因此我们对其不作深入分析。

高级打印驱动软件的基本原理同一般打印驱动软件的原理差不多，也是修改 ROM-BIOS 中原 17 类和 5 类中断入口向量地址使其指向新的模块入口地址。但其功能和驱动高点阵汉字库是一般打印驱动程序（软件）所不及的，其包含的 17 类中断程序内容很丰富，不仅能打印出 16 点阵、24 点阵的汉字，而且能打印高点阵的 48 点阵汉字及其四种字体（宋体、仿宋体、楷体、黑体）。能够支持打印机的各种命令，其详细情况见以后各章节。

3. 2 高级打印驱动软件的框图

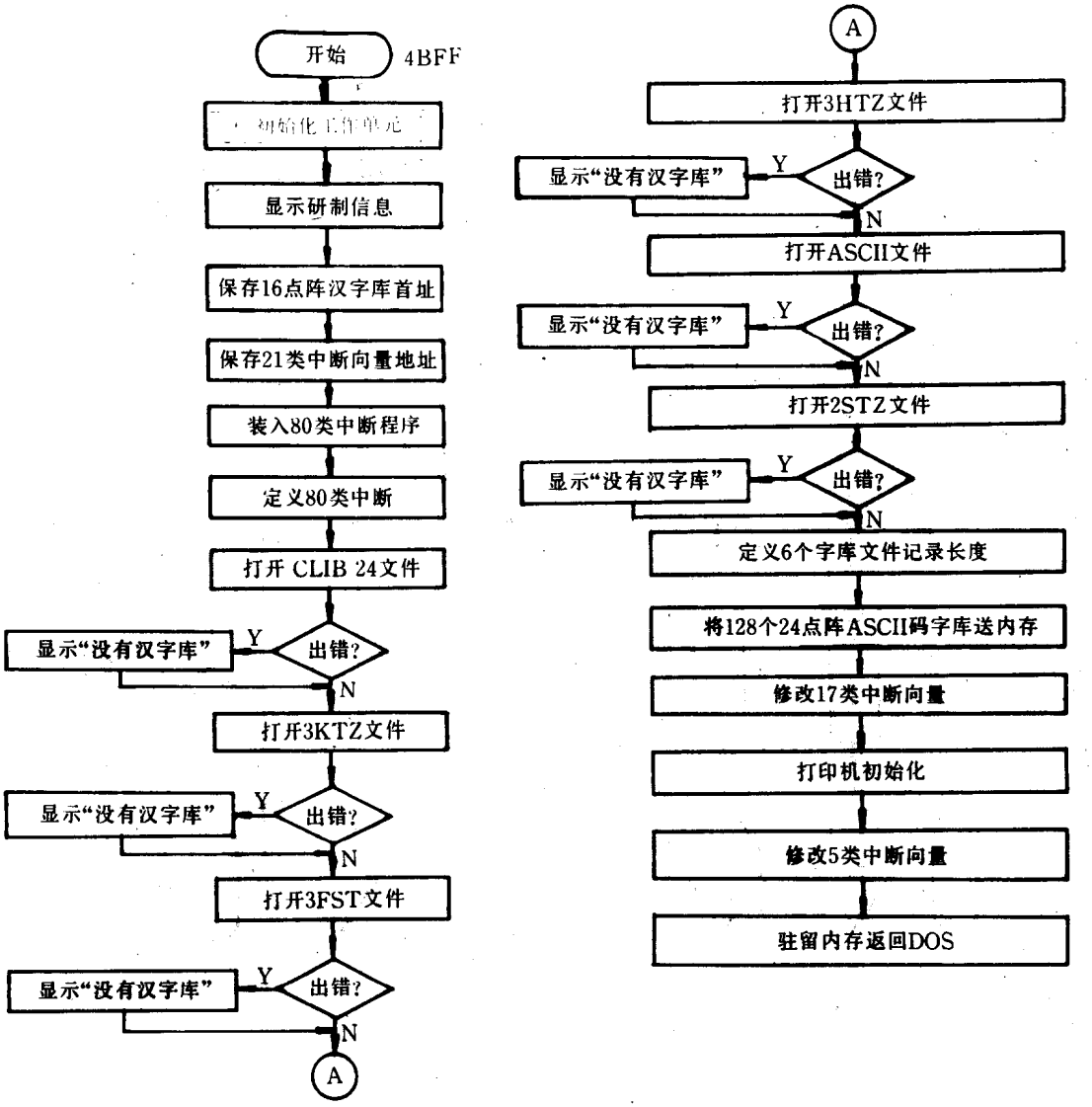


图 3-1　H3070A. EXE 本身代码执行过程

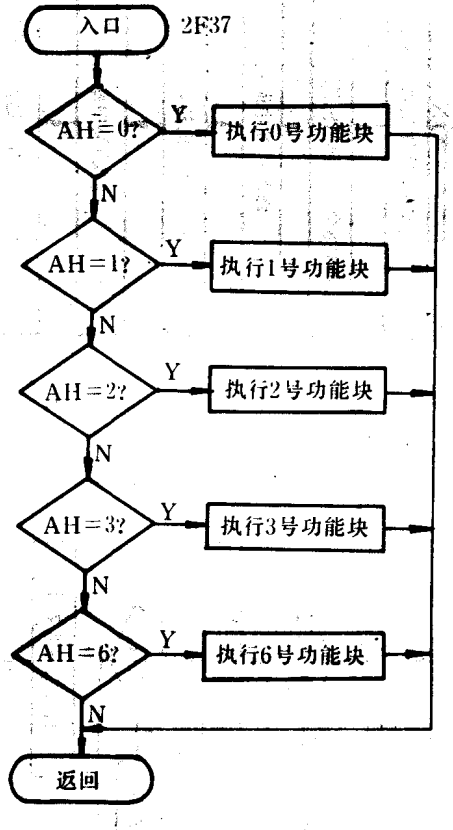


图 3-2　17 类中断程序工作流程

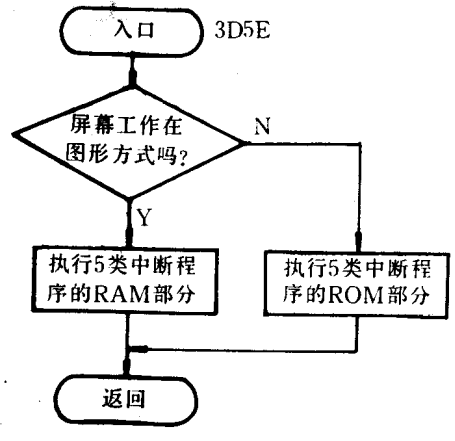


图 3-3　5 类中断程序工作流程

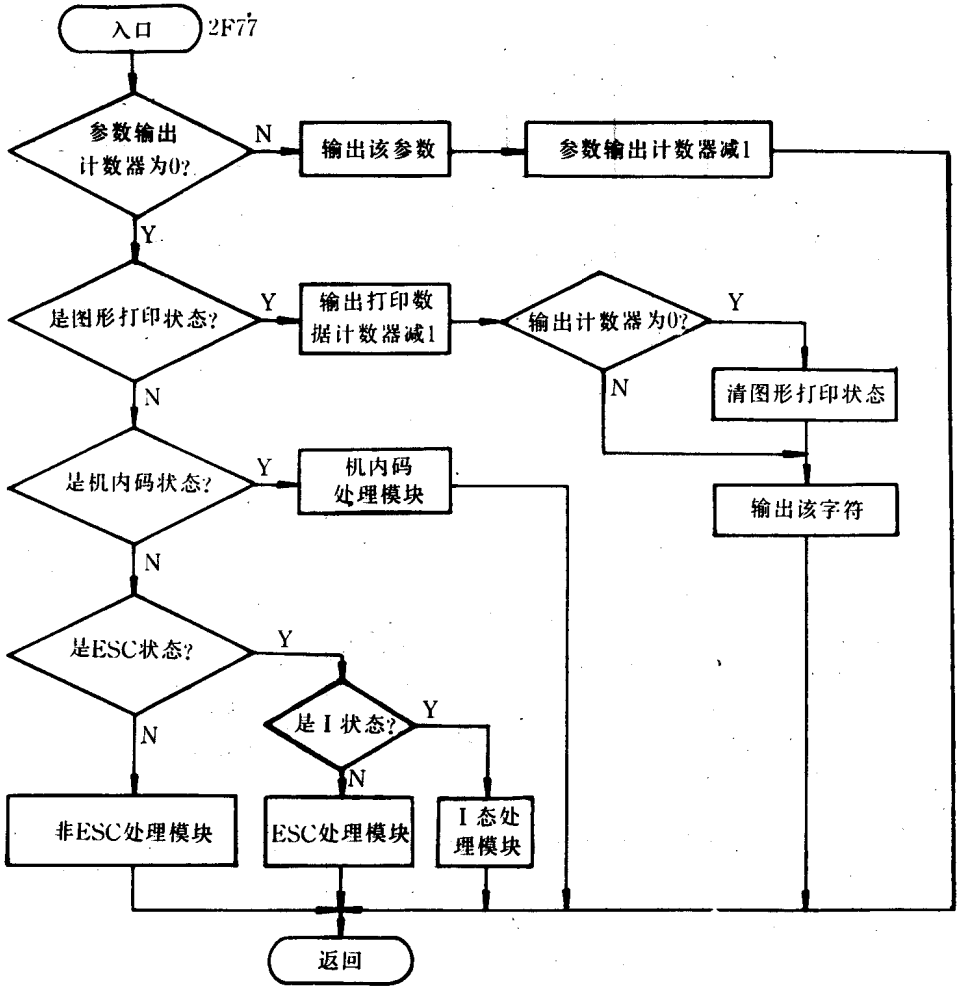


图 3-4　17 类中断 0 号功能块工作流程

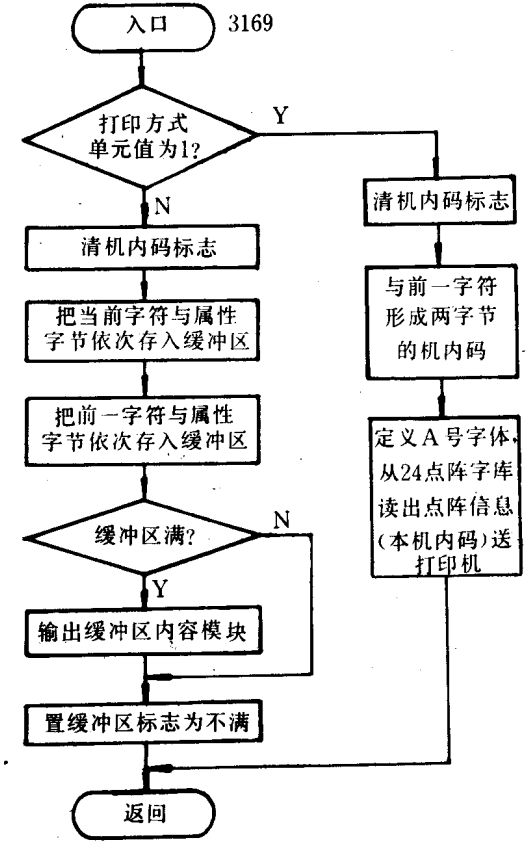


图 3-5　0 号功能块中的机内码状态处理子模块工作流程

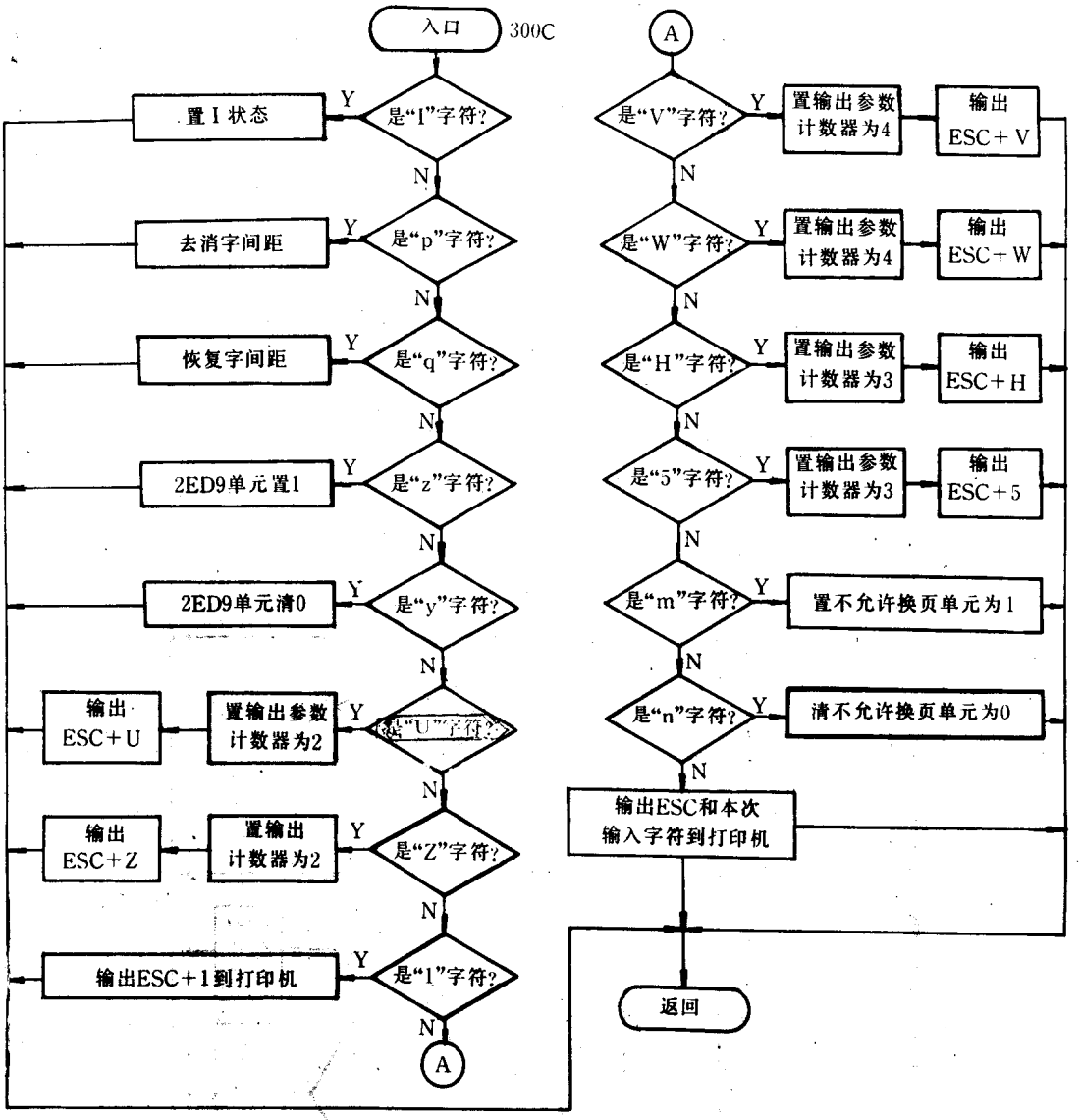


图 3-6　0 号功能块中 ESC 状态处理子模块工作流程

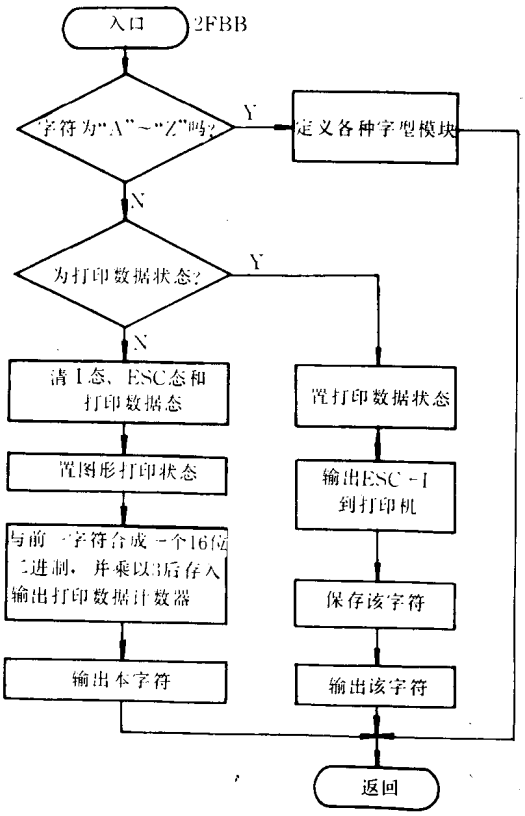


图 3-7　0 号功能块 I 态处理子模块工作流程

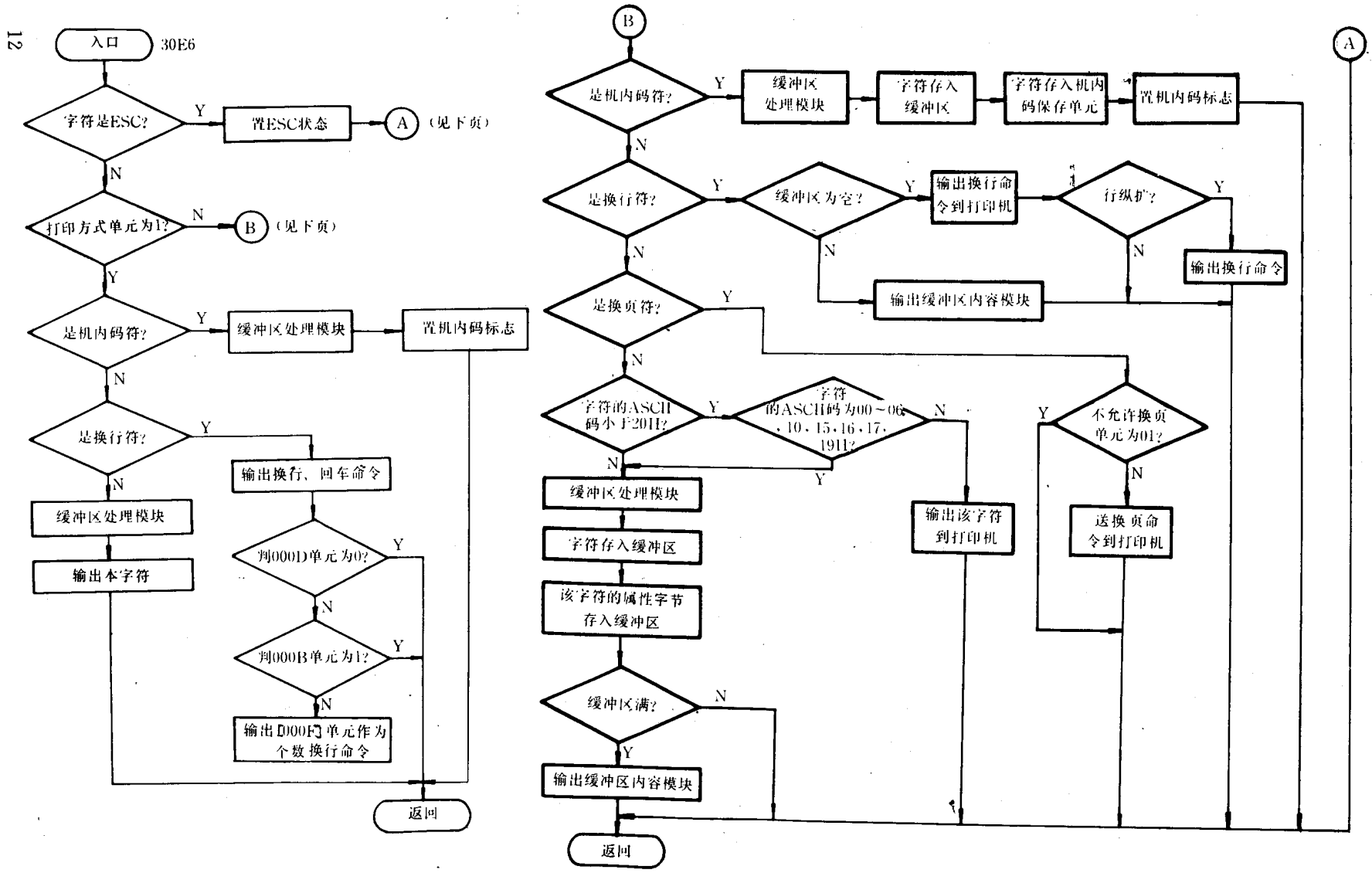


图 3-8　0 号功能块非 ESC 态处理子模块工作流程