微型计算机软件专辑

上海交通大学 微机研究所
科技交流室
一九八五年四月

目　录

8086 简易编辑程序

唐长钧

一般的单板计算机不具有软件开发能力。因此，用户使用单板计算机时只能使用机器代码，这对程序编制、修改以及程序进入单板计算机都带来很大的麻烦。16 位单板计算机具有较强的功能，如能配备适当的语言(如汇编，BASIC，FORTH 等)，则可大大加强它开发软件的能力。为了使用这些语言，作为基础首先必须配备文本编辑程序，这就是编制本编辑程序的目的。目前它与汇编程序一起，已固化在 MIC－867(8086＋8087)单板计算机上。本编辑程序也可用于任何 CPU 是 8086 或 8088 的单板计算机。

设计的一个指导思想是所占用的内存容量尽可能地要小，以便有更大的空间提供给用户使用。本编辑程序的另一个特点是，无需改动就可把本编辑程序置于本代码段(现为70)，64KB 内存寻址范围的任意区域。这样用户可根据需要任意改变编辑程序在存贮器的位置，这对固化和移植带来了极大的方便。此外，只要改动二处(共 4 个字节)便可改变生成文本的源程序区在内存中的位置，而且可任意改变该源程序区的大小。这对有效使用内存创造了条件。

一、编辑命令

本编辑程序提供给用户 10 条编辑命令，说明如下。

•B 命令：

命令形式：Bh↵(↵为回车符号)

功能：指点器指向从文本顶部开始的 h 行。h 为 16 进制数。

•E 命令

命令形式：E ↵

功能：返回监控程序。

•F 命令

命令形式：F C_1 C_2 C_3…Cn ↵

功能：从现在指点器所指的位置开始，向下寻找某一行，该行的第 1 个字符开始应为 C_1 C_2…Cn。如找到，则指点器指向该行；如找不到，指点器指向文本底部(此时显示 *)。

•I 命令

命令形式：a) I C_1 C_2…Cn 或

b) I↵ 输入 n 行↵↵

功能：使用形式 a)时，就在指点器所指位置的上面插入新的一行 C_1 C_2…Cn，指示器位置不变。使用形式 b)时，就在指点器所指位置的上面插入任意行，指点器仍指向原来行的位置。当指点器指向文本顶部时，使用 I 命令，就在文本的顶部加入新的行；当指点器指向文本底部时，使用 I 命令，就在

文本的底部加入新的行。

*K 命令

命令形式：Kh ↵

功能：从现在指点器所指的位置开始，削除 h 所指的行数，指点器指向削除部分的下一行。h 为 16 进制数。

*L 命令

命令形式：Lh ↵

功能：从指点器所指位置开始，向上移 h 行。

*N 命令

命令形式：Nh↵

功能：从指点器所指位置开始，向下移 h 行。

*S 命令

命令形式：S C_1 C_2…Cn↵

功能：将现在指点器所指的一行，用新的一行 C_1 C_2…Cn 来取代，指点器所指位置不变。

*T 命令

命令形式：Th↵

功能：显示指点器所指位置开始的 h 行。指点器所指位置不变。h 为 16 进制数。

*Z 命令

命令形式：Z↵输入 n 行↵↵

功能：指点器指向文本底部，并可接着输入 n 行。指点器指向输入 n 行的底。

二、编辑程序内部数据结构

图 1 为本编辑程序源程序区存放的文本以及执行几条主要编辑命令后，指点器移动的位置。TOP 和 BOTTOM 分别是源程序区的顶和底(TOP 和 BOTTOM可变动)，也就是说源程序最大不能超过(BOTTOM-TOP)个字节数。整个源程序区分为上下二部分，即上部分指点器 UP 以上(左)部分和下部分指点器 DN 以下(右)部分。图 1-①为现在源程序区的状态，即上部分有二行，为a和b，UP指向b 行最低位字符的下一个空字节位置；下部分也有二行，为d和e，DN 指向 d 行最高位字符的上一个空字节位置。图 1-②为执行一条“NI”命令后的状态，该命令执行后，源程序区下部分的第一行 d 移到了上部分，UP 就指向 d 行最低位字符的下一个空字节位置。DN 就指向 e 行最高位字符的上一个空字节位置。图1-③是执行一条“L1”命令后的状态，该命令将上部分最末行b移到下部分，UP、DN 作如图的变动。图 1-④为执行一条插入命令 I 后的状态，如插入一行为 C，则此行插在上部分的下面，一旦插入，UP 就指向 C 行最低位字符后的一个空字节。图 1-⑤为执行一条“K1”命令后的状态，该条命令从下部分取消一行C，使 DN 指向 e 行最高位字符的上一个空字节位置。

图 2 为源程序区中一行的结构。每行的长度是不定的。左边为一行b 行，b 行由字符 b_0，b_1，b_2，b_3 组成，b 行在源程序区的上部分。这一行中每个字符占高 7 位二进制位，最低一位为行界标志，当字符的行界标志位为0时，说明此行没结束(从右向左看)；为 1 时，说

明此行结束。UP 指向该行最低位字符下一个字节位置。右边一行为 d 行，d 行由字符 d_0，d_1，d_2，d_3 组成，d 行在源程序区的下部分。这一行中每个字符占低 7 位二进制位，最高一位为行界标志，当字符中行界标志为1时，说明此行结束(从左向右看)。DN 指向 d 行最高位字符的上一个字节位置。由此可见：源程序区空区域在中间部分。这样，当要将一行从上部分移至下部分，或从下部分移至上部分时，可采用图 3 的程序。

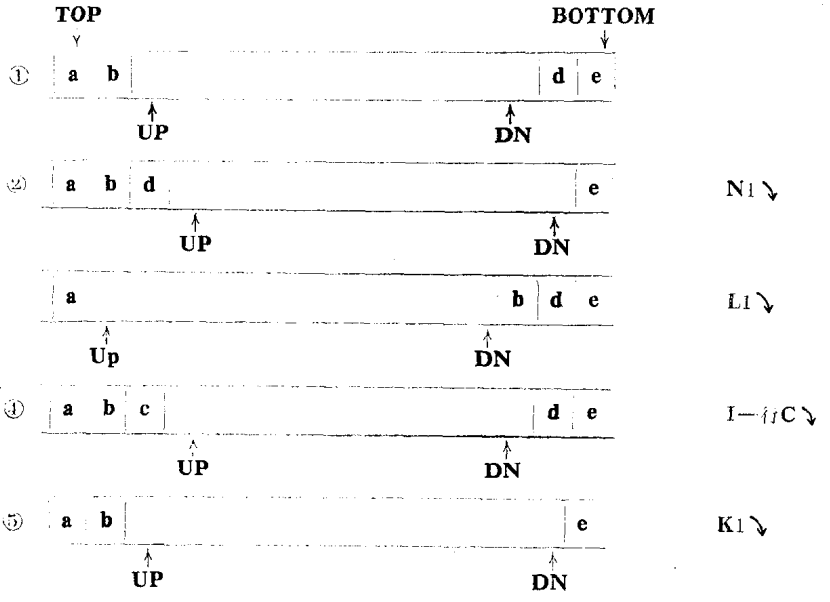


图 1　指点器状态

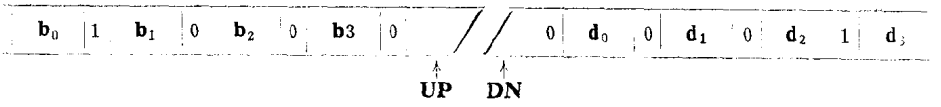


图 2　行的结构

```
MOV     BX, Up      ;  将上部分指点器 up 内容，即所指的地址送 BX。
MOV     DX, DN      ;  将下部分指点器 DN 内容，即所指的地址送 DX。
STC                 ;  置 1 进位位 CF。
→INC    DX          ;  DX←(DX) + 1，即 DX指向下部分第一行的一个字符。
XCHG    BX, DX      ;  交换 BX，DX 的内容。
MOV     AL, [BX]    ;  将下部分第一行的一个字符送至 AL。
XCHG    BX, DX      ;  交换 BX，DX 内容。
RCL     AL, 1       ;  连进位位左移(第一次移位时因 CF = 1，故移位后 AL 的末
                       位必为 1，成为上部分行结束标志)。
MOVMB [BX], AL      ;  将该字符送至上部分 up 所指位置。
INC     BX          ;  BX←(BX) + 1。
JNB                 ;  若到行末，则必 CF = 1，循环结束，否则返回继续，传送
                       下一个字符。
```

(a)　下部分一行移至上部分的程序

```
MOV     BX, Up    ;  将上部分指点器 up 的内容，即所指的地址送 Bx。
MOV     DX, DN    ;  将下部分指点器 DN 的内容，即所指的地址送 Dx。
STC               ;  置 1 进位位 CF。
DEC     BX        ;  BX←(BX) - 1,即BX 指向下部分最末行的最末一个字符。
MOVB    AL, [BX]  ;  将下部最末行最末一个字符送至 AL。
RCRB    AL, 1     ;  连进位位右移(第一次移位时，因 CF = 1，故移位后 AL 的
                     最高位必为 1，形成下部分的结束标志。
XCHG    BX, DX    ;  交换BX, DX 的内容。
MOVMB [BX], AL    ;  将该字符送至下部分 DN 所指位置。
XCHG    BX, DX    ;  交换 BX, DX 内容。
DEC     DX        ;  DX←(DX) - 1。
JNB               ;  若到行末，则必 CF = 1，循环结束。否则返回，继续传送
                     下一个字符。
```

(b) 上部分一行移至下部分的程序。

图 3 移动一行的程序及其说明。

三、使用方法

在启动本编辑程序之前，用户必须事先设置几个参量。它们是：

UP(2 个字节)——源程序区上部分指点器；

DN(2 个字节)——源程序区下部分指点器；

OVER(2 个字节)——源程序区上部分总行数；

UNDR(2 个字节)——源程序区下部分总行数。

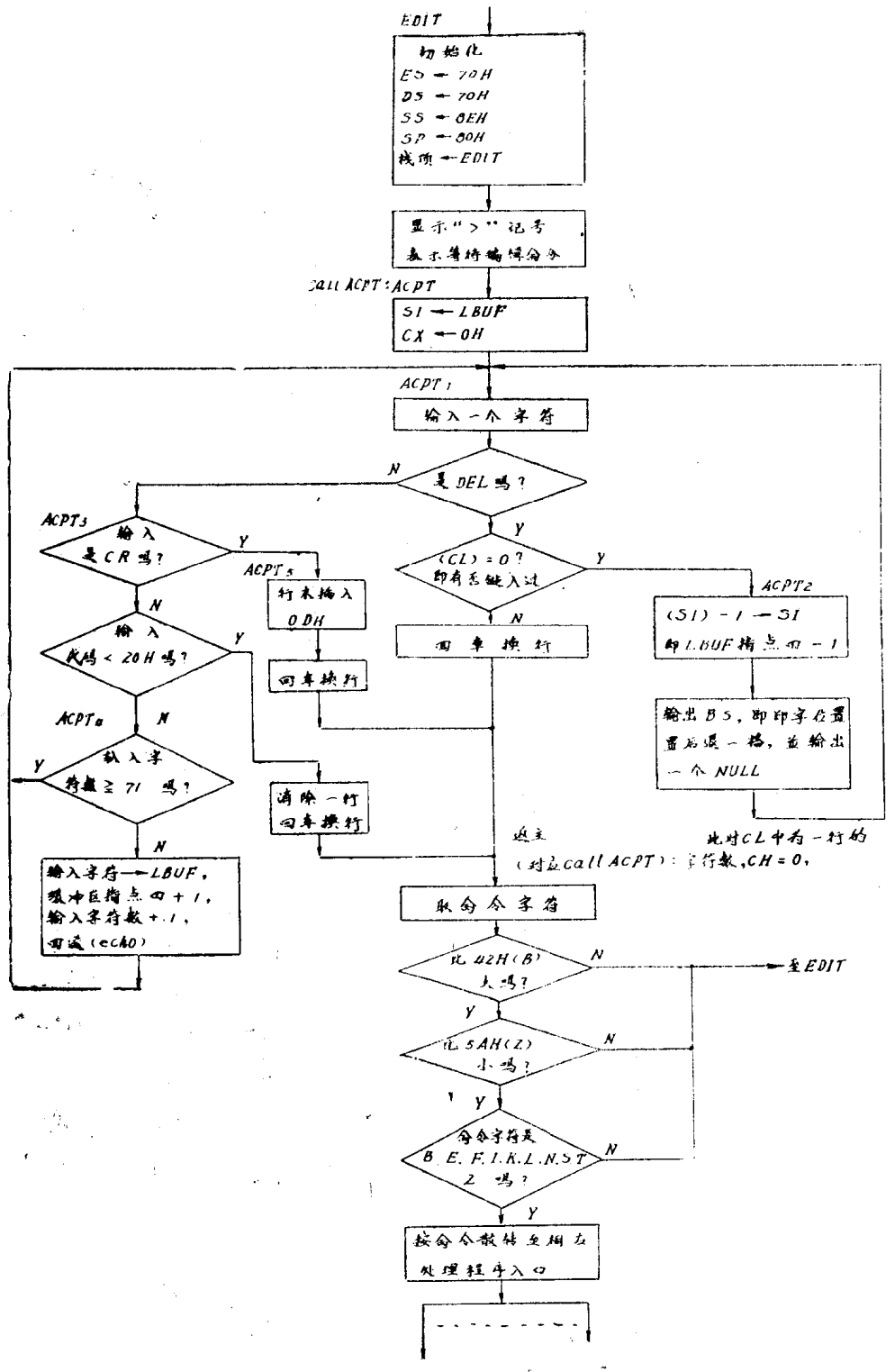
如这四个参量依次放在存储器 70：260～267 八个单元，初始值为00，1A，FE，38，00，00，00，00。其含义是源程序区上部分指点器 UP 指向内存地址 1A00H，源程序区下部分指点器 DN 指向内存地址 38FEH。也就是说用户编辑的源程序可放在从 1A00H～38FEH 的存储器范围内。OVER，UNDR 开始时都设置为0，即这时在源程序区内没有一行内容。可见要改变源程序区的范围，只要在开始启动编辑程序之前用监控命令对 UP、DN 置上不同的地址就可以了。

设置这四个参量后，最好在 DN 所指内存地址的下一个地址置上 AA 的内容，上例中即在70：38FF地址置AA内容。这是为了在表示文本结束时，在 CRT 显示“*”字符。

接着使用监控命令，从地址 F800：0000 启动编辑命令，这时 CRT 将显示提示符“>”。用户在提示符后，即可使用上述的十条命令进行文本编辑。在文本编辑过程中，UP、DN、OVER、UNDR 四个参量的内容将自动变化。文本编辑结束后由E命令将状态返回到监控。

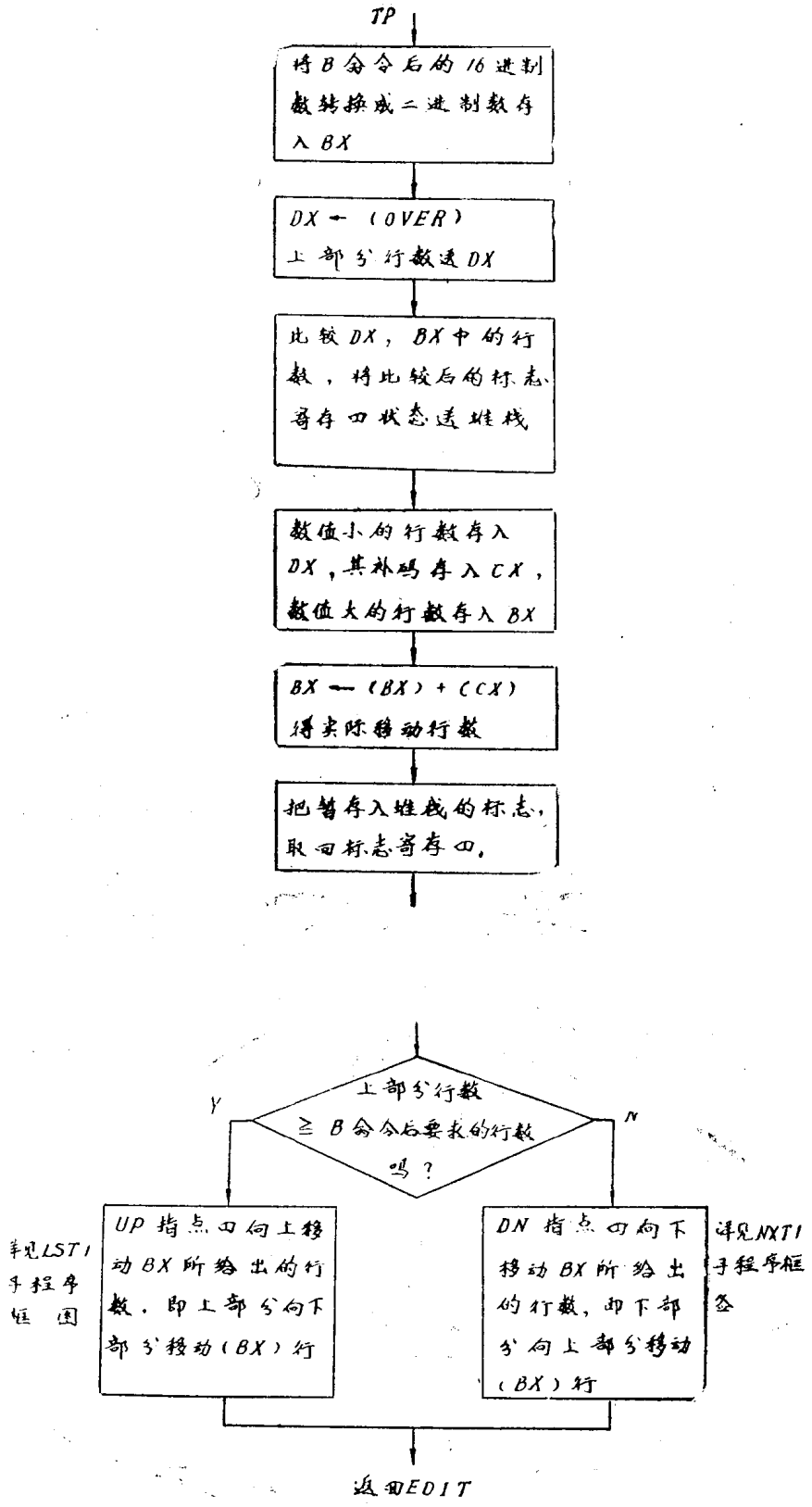
四、编辑程序框图及清单

本编辑程序除了上述四个参量外，还有一个 LBUF(行缓冲器)参量，用以存放键入的一行。


EDIT
初始化
ES ← 70H
DS ← 70H
SS ← 8EH
SP ← 80H
栈顶 ← EDIT
显示">"记号
表示等待输入命令
call ACPT:ACPT
SI ← LBUF
CX ← 0H
ACPT1
输入一个字符
是DEL吗?
N
Y
ACPT3
输入是CR吗?
Y
ACPT5
行未输入0DH
回车换行
N
输入代码<20H吗?
Y
ACPT6
N
输入字符是否≥71吗?
Y
N
输入字符→LBUF,缓冲区指点(SI)+1,输入字符数+1,回送(echo)
消除一行回车换行
(CL)=0?即有无键入过
Y
N
回车换行
ACPT2
(SI)-1→SI即LBUF指点(SI)-1
输出BS,即印字位置置后退一格,并输出一个NULL
此时CL中为一行的字符数,CH=0,
返主(对应call ACPT)
取命令字符
比42H(B)大吗?
N
至EDIT
Y
比5AH(Z)小吗?
N
Y
命令字符是B.E.F.I.K.L.N.S.T.Z吗?
N
Y
按命令散转至相应处理程序入口

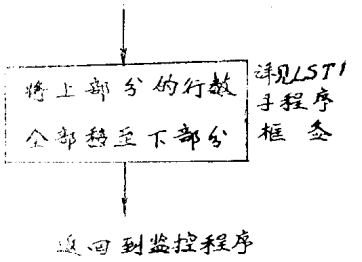
接下图

B 命令处理程序框图

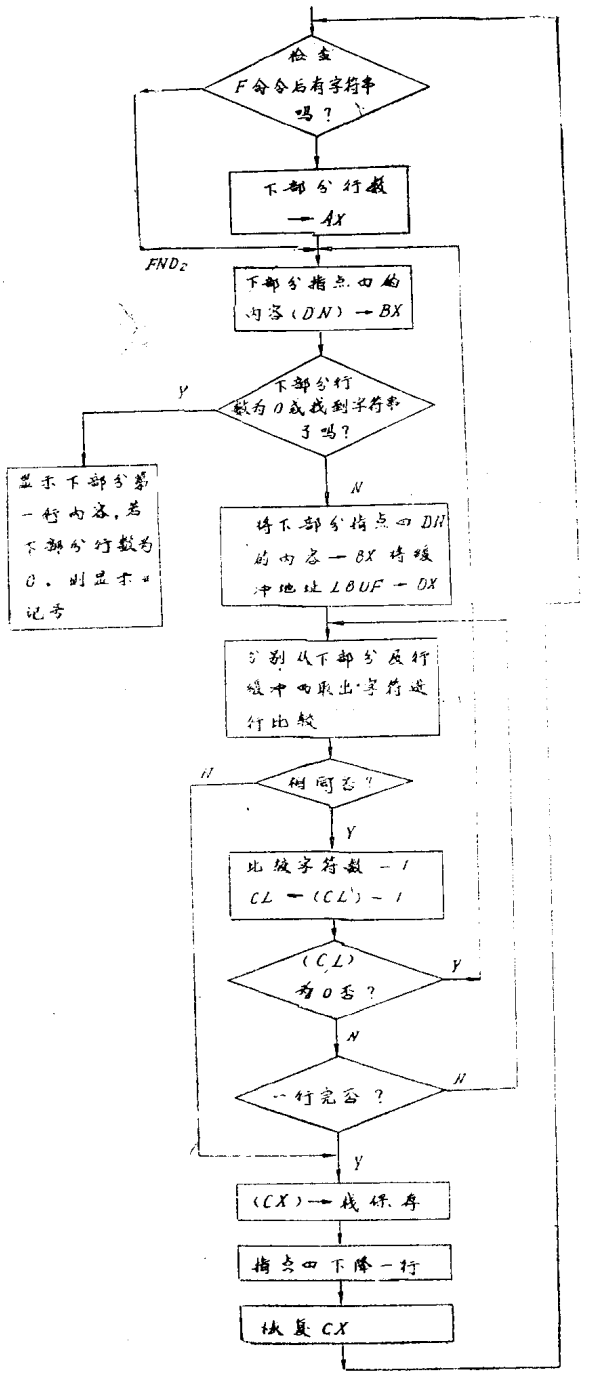


接下图

E 命令处理程序框图：

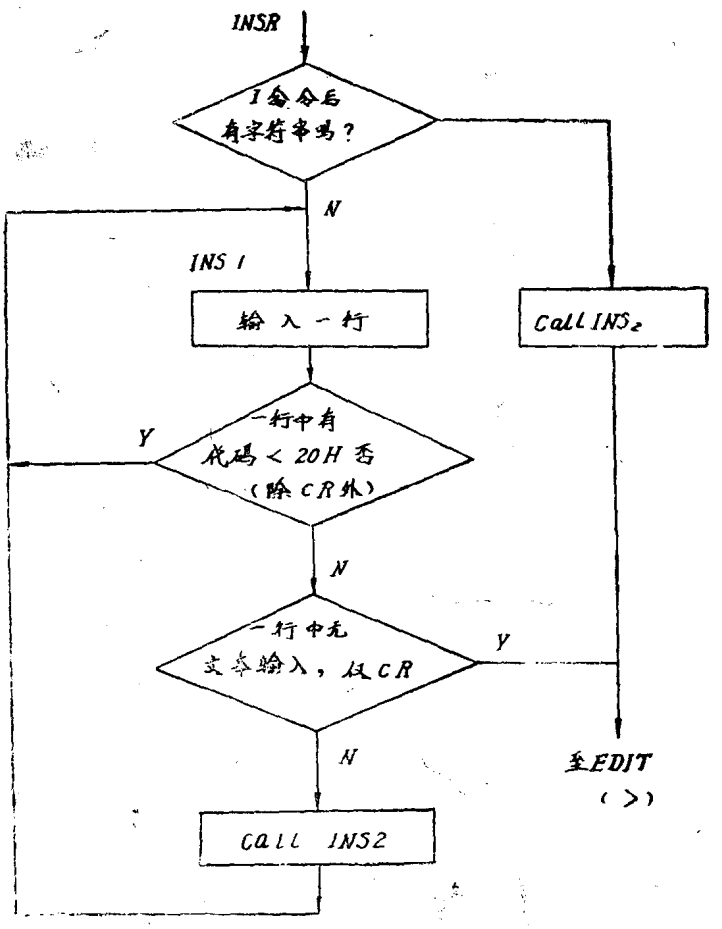


F 命令处理程序框图

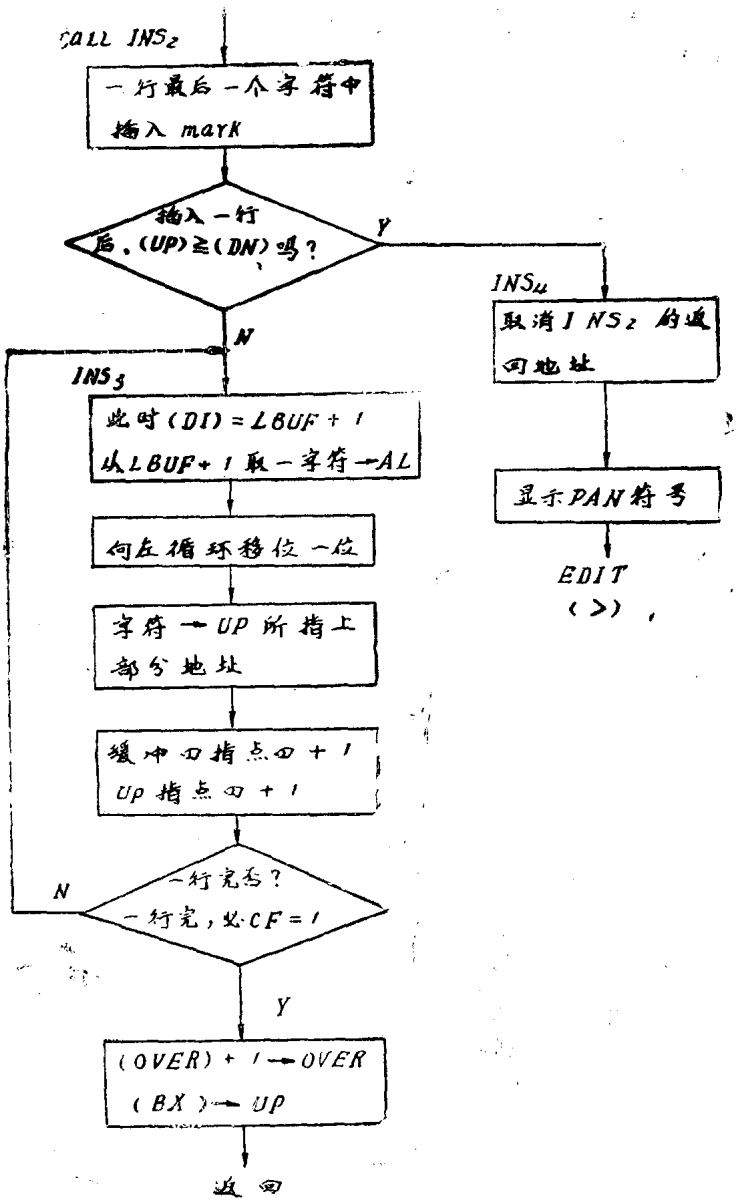


接下图

1 命令处理程序框图：

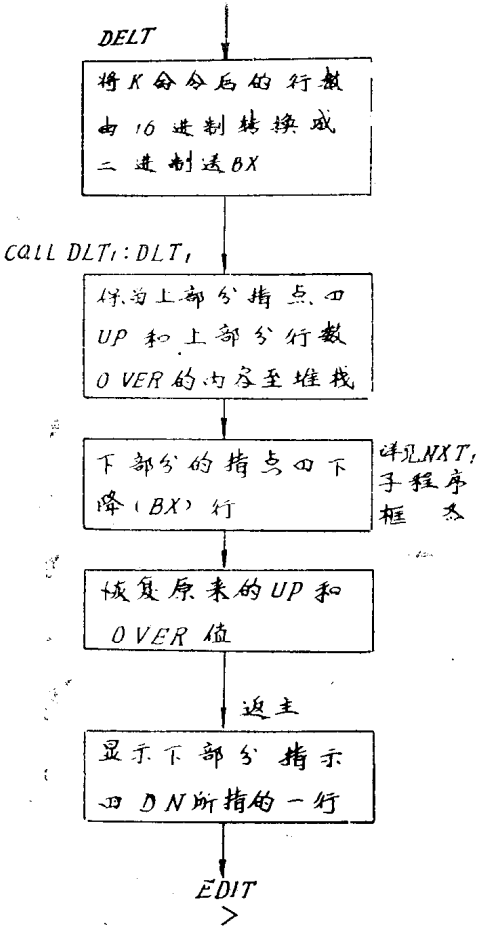


接下图

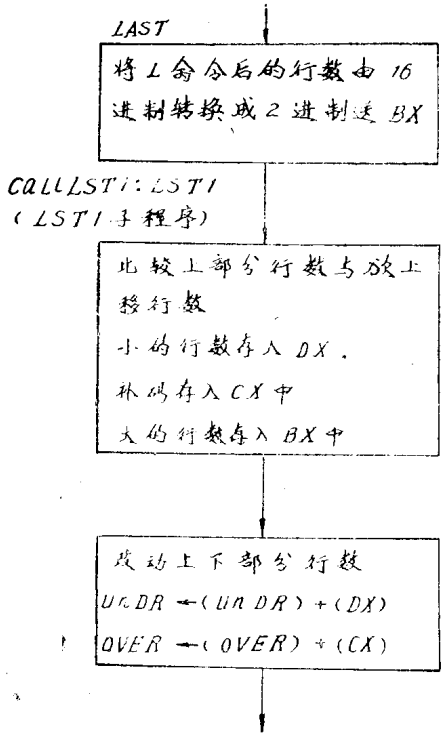


接下图

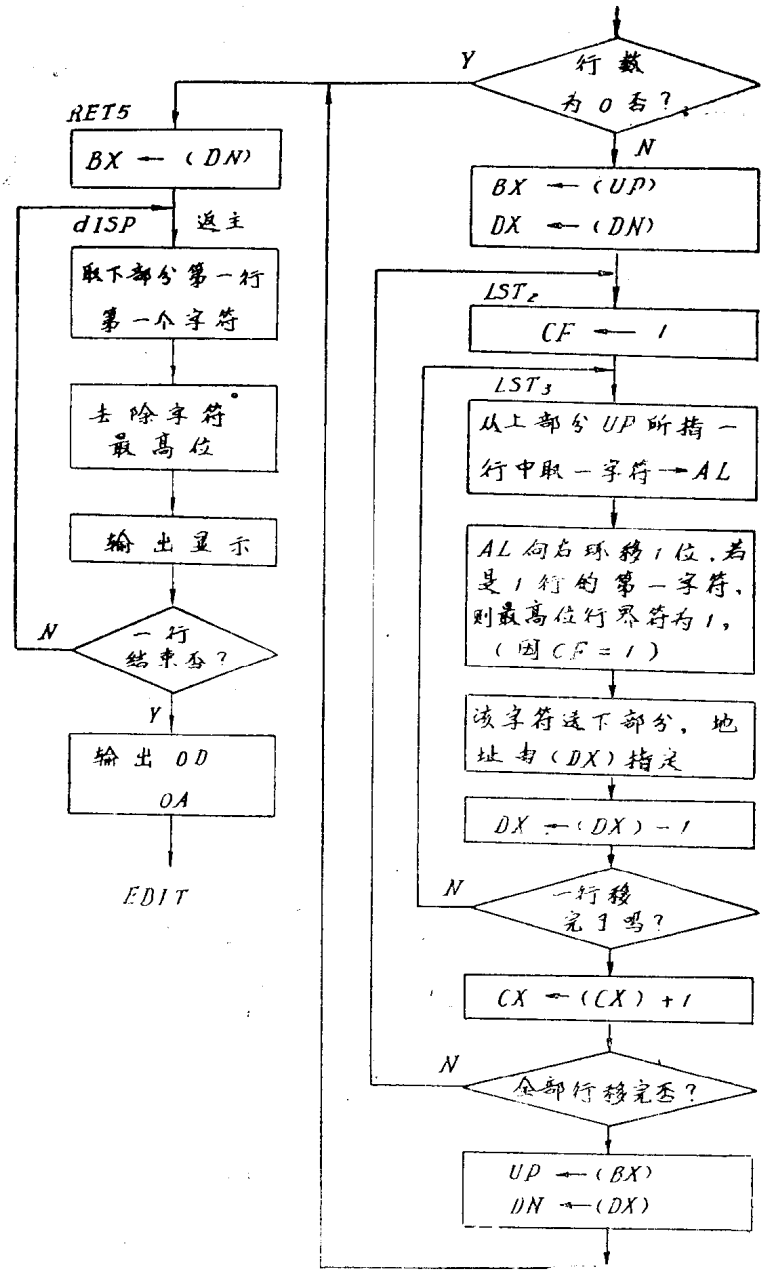
K 命令处理程序框图：



L 命令处理程序框图

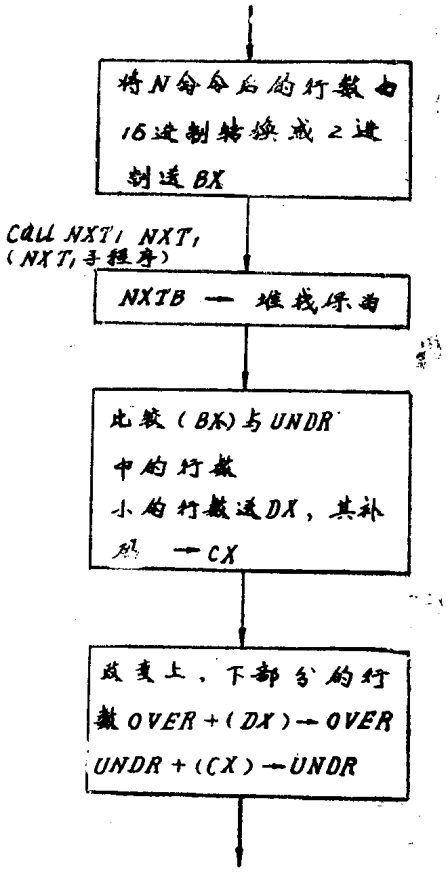


接下图

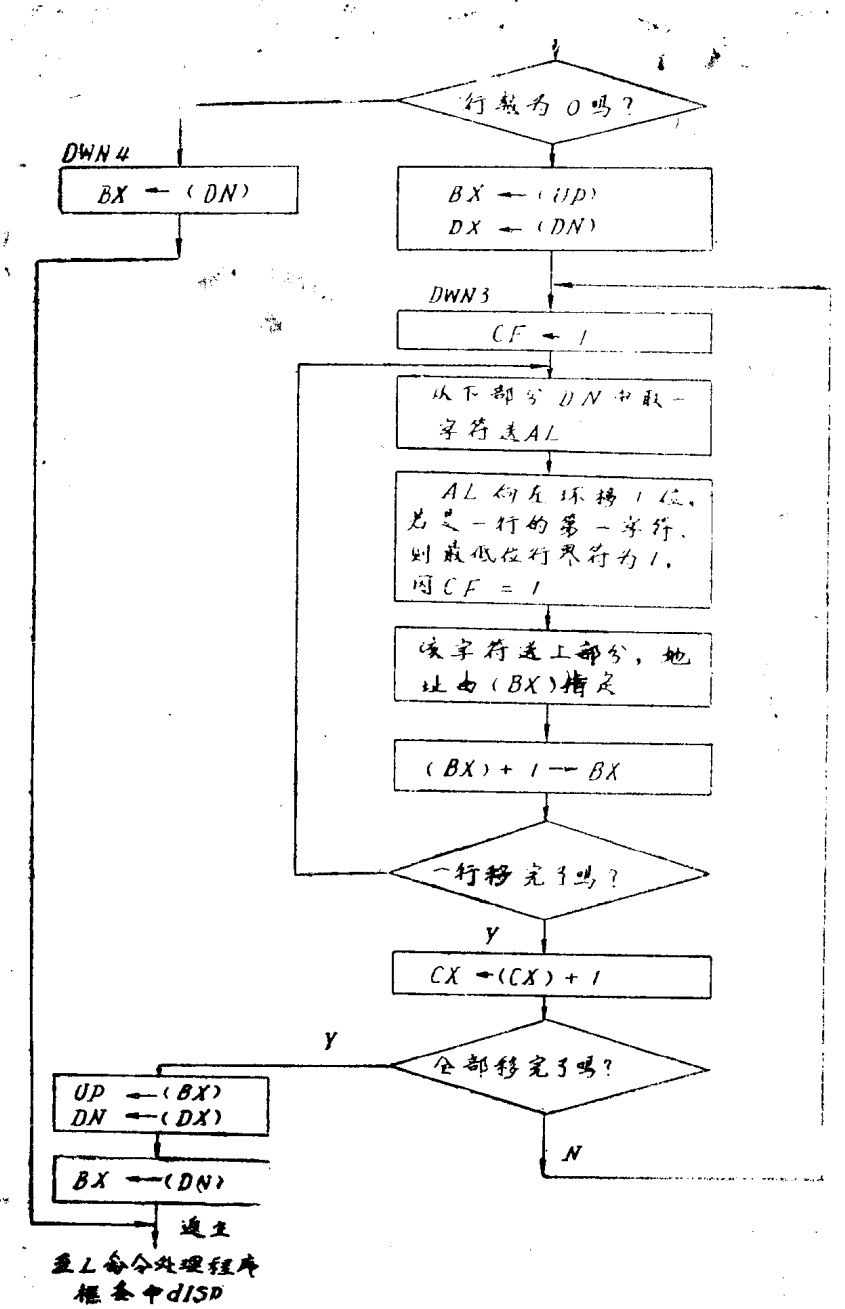

Y
行数
为0否?
N
RET5
BX ← (DN)
BX ← (UP)
DX ← (DN)
dISP
返主
取下部分第一行
第一个字符
LST2
CF ← 1
LST3
去除字符
最高位
从上部分UP所指一
行中取一字符→AL
输出显示
AL向右环移1位,若
是1行的第一字符,
则最高位行界符为1,
(因CF=1)
N
一行
结束否?
Y
该字符送下部分,地
址由(DX)指定
输出 0D
0A
DX ← (DX) - 1
N
一行移
完了吗?
CX ← (CX) + 1
N
全部行移完否?
EDIT
UP ← (BX)
DN ← (DX)

接下图

N 命令处理程序框图。

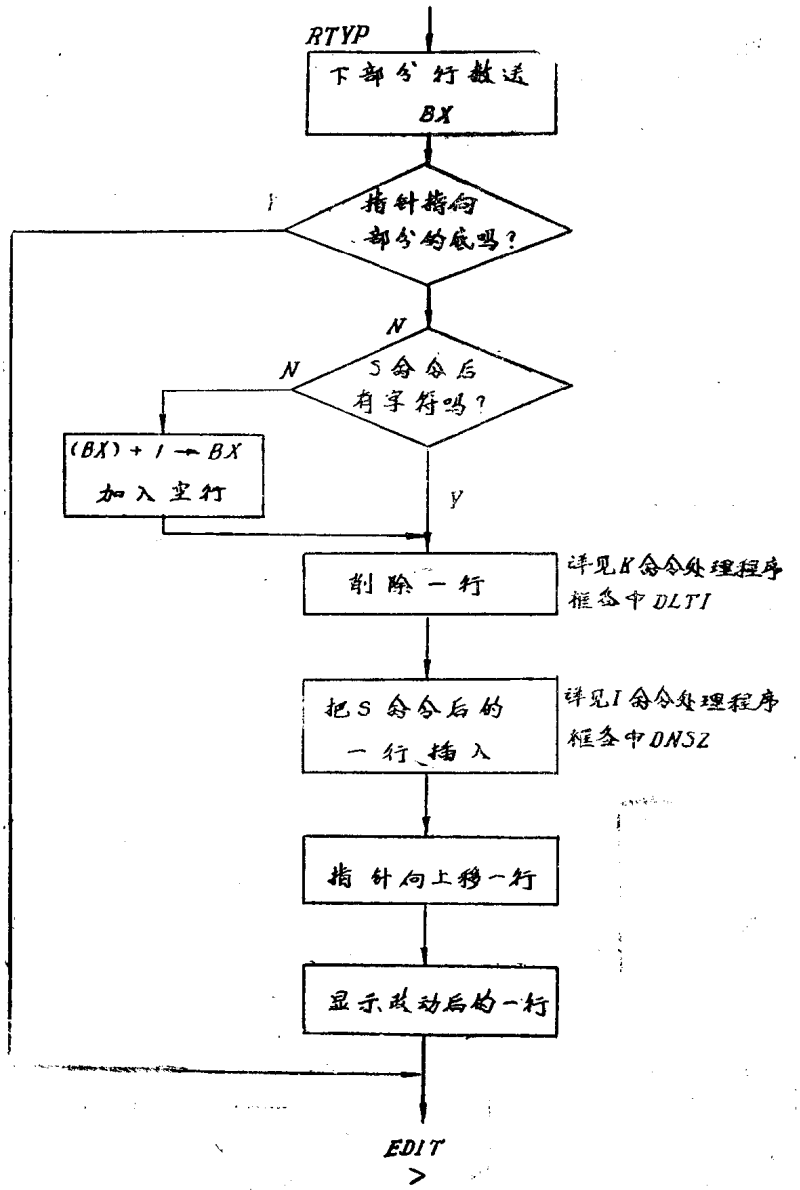


接下图


行数为0吗?
DWN4
BX ← (DN)
BX ← (UP)
DX ← (DN)
DWN3
CF ← 1
从下部分DN中取一字符送AL
AL内左环移1位,若是一行的第一字符,则最低位行界符为1,同CF = 1
该字符送上部分,地址由(BX)指定
(BX) + 1 → BX
一行移完了吗?
Y
CX ← (CX) + 1
Y
全部移完了吗?
UP ← (BX)
DN ← (DX)
BX ← (DN)
N
退出
至L命令处理程序框套中DISP

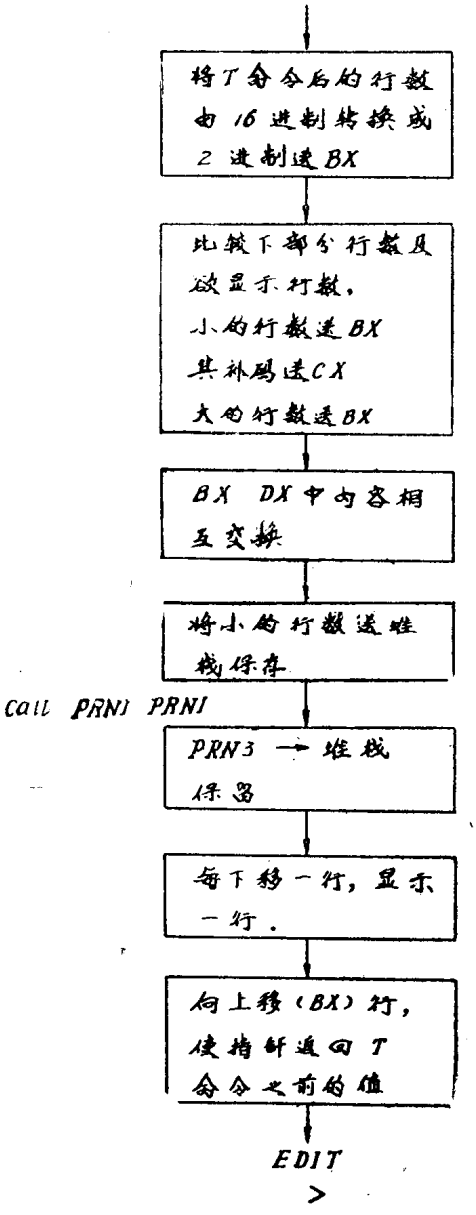
接下图

S 命令处理程序框图：



接图下

T 命令处理程序框图：



接下图