

DJS—21 机压缩纸带方案

计 算 站 闻人德泰

一、引 言

考虑到 DJS—21 机的 ALGOL 60 编译程序纸带是按指令格式穿孔的，以致一个指令单元需占用 15 排孔：一排特征孔，十四排信息孔。这样，整个编译程序的纸带就很长(目前为三盘)。

拥有该机的用户，如果未配磁带机，或者有磁鼓但不稳定，或者由于欲对原系统作某种修改试验，导致经常需要通过光电输入机输入三盘纸带，这时，用户就会感到非常不方便。

鉴于上述考虑，本文提出一个将纸带上的信息密度加密，以便将纸带上的信息加以压缩的设计方案。压缩后的纸带通过一个扩展的光电引导程序输入计算机后，与原纸带是等同的，而纸带长度则大大缩短了。DJS—21 机编译程序(未扩体)被压缩成一盘。估计 DJS—21 机编译程序的扩体方案经整理之后，亦可压缩成一盘纸带。这样，系统的输入手续简单了，速度加快了，同时，纸带的复制与保管也方便了。

本方案亦可用于 DJS—21 机的其它软件系统纸带，亦可供其它计算机上压缩纸带的参考。

二、特征孔的定义

在正规格式的穿孔中，曾定义了下列特征孔：

- 00001: 数特征孔
- 00100: 指令特征孔
- 00110: 地址特征孔
- 00111: 结束特征孔

本压缩方案定义了以下特征孔：

1. 开始特征孔 (00010)



该特征孔用于纸带的首部，在第一次遇到该特征孔之前的所有孔形，光电输入程序均不予“理睬”。

该特征孔的第二次以后的出现则为信息孔或空特征孔(见 2)。

2. 空特征孔 (00001, 00010)



光电输入程序视之为无特征信息的空排，就好象硬件视 00000 为黑纸带那样，但是它们

只能出现在特征孔处才作为空，在信息孔位置则分别表示 1 和 2（十六进制）。

3. 结束特征孔 (00011)



该特征孔用于纸带的结尾，光电输入程序（注：此处及以下提及此名称时均指光电输入扩展程序）遇该特征孔时执行

024 1000

以停止光电输入。

4. 地址特征孔 (00100~00111)

这四个特征孔相当于正规方式下的 00110。但它们的后两位是地址的两位高位。由于地址形式为 2444，故在地址特征孔后面紧接着三排孔是地址码的低十二位，例如：



a 地址 2657

b 地址 0610

c 地址 1005

d 地址 3050

由以上例子可见，信息“0”的表示方法采用 10000 码（结合于特征孔中的 0 例外，如 b 例 0610 中第一个 0）。

5. 指令的分类

定义：除最左八位不同时为 0 外，其余各位均为 0 的单元称为左 θ 指令。

定义：除左指令最右位和右操作码七位共八位不同时为 0 外，其余各位均为 0 的单元称为右 θ 指令。

定义：除左操作码最右二位及左地址位共 16 位不同时为 0 外，其余各位均为 0 的单元称为左 D 指令。

定义：指令的最左十位同时为 0，且又不属于上面定义三类指令，则称为特殊指令。（当然也不属于全 0 单元）。

定义：指令单元各位均为 0 时，称为全 0 单元。指令单元各位均为 1 时，称为全 1 单元。

定义：上述各类指令之外的单元称为正常指令。

6. 正常指令特征孔 (01000~01011)

由以上定义可知，此类指令没有什么特殊性，因此 42 位代码按 244444444444 格式处理，最左两位结合于特征孔中，故而在此特征孔后需紧接着十排信息孔。

为了进一步压缩纸带长度，需要着重指出下面一点：

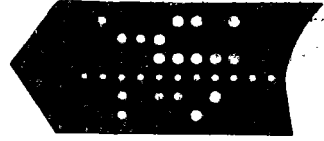
若在信息孔“0”(10000)之后尚有信息孔，则将这两排信息孔逻辑加成一排信息孔。因此，10000 表示“0”“0”两排信息孔。10111 表示“0”“7”两排信息孔的叠加。

由于这一规定，在特征孔之后实际上跟着的信息孔的排数可能少于十排（实现算法中用计数器控制就不会混乱）。

对于代码 10000 出现于信息孔的最后一排时，有两种情况：代表一个 0 (10) 或两个 0 (100)，借助于计数器同样可避免对该代码解释的二义性。

以下为几个例子：

002 0517 034 0525



028 1104 025 0553



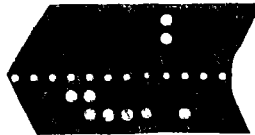
034 0590 021 0626



102 0000 004 0015



001 0000 000 0030



7. 全 0 单元特征孔 (01100)

该特征孔指出当前单元内容应为全 0。该内容可由光电输入程序直接提供，故该特征孔后没有信息孔，其后必为一特征孔。

8. 全 1 单元特征孔 (01111)

该特征孔指出当前单元内容应为全 1。该内容可由光电输入程序直接提供，故该特征孔后没有信息孔，其后必为一特征孔。

9. 特殊指令特征孔 (10000~10111)

由定义知，该类指令最左 10 位必为 0，因此最多后跟八排信息孔。

特征孔的右三位为 0~7，分别代表着后跟 1~8 排信息孔。

当后跟七排孔时，表示最左十四位全为 0。

当后跟六排孔时，表示最左十八位全为 0。

当后跟五排孔时，表示最左廿二位全为 0。

当后跟四排孔时，表示最左廿六位全为0。

当后跟三排孔时，表示最左三十位全为0。

当后跟二排孔时，表示最左卅四位全为0。

当后跟一排孔时，表示最左卅八位全为0。

例子：

000 0000 000 0004



000 0000 0000 0030



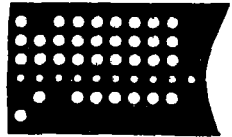
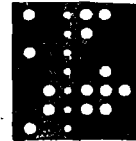
000 0000 000 0703



000 0000 004 0100



000 0020 000 2540



000 0735 135 3555



000 0400 000 1000

10. 左θ指令特征孔 (11000)

该特征孔后紧跟两排信息孔，光电输入程序将它们置于相应的位上。

例如：

021 2000 000 0000



037 0000 000 0000



003 0000 000 0000



030 0000 000 0000

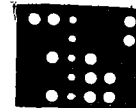


11. 左D指令特征孔 (11001)

该特征孔后跟四排信息孔。

例如：

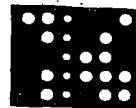
000 1264 000 0000



001 0506 000 0000



002 2651 000 0000



指 12. 右 θ 令特征孔 (11010)

该特征孔后跟二排信息孔

例如:

000 0000 134 0000



001 0001 020 0000



000 0000 003 0000



13. 废特征孔 (11011~11111)

在定义本特征孔系统时, 出发点是针对诸如编译程序纸带之类的常用专用纸带的。因此, 数特征孔未予定义, 故将所余之孔 (11011~11111) 全定义成废码特征孔。

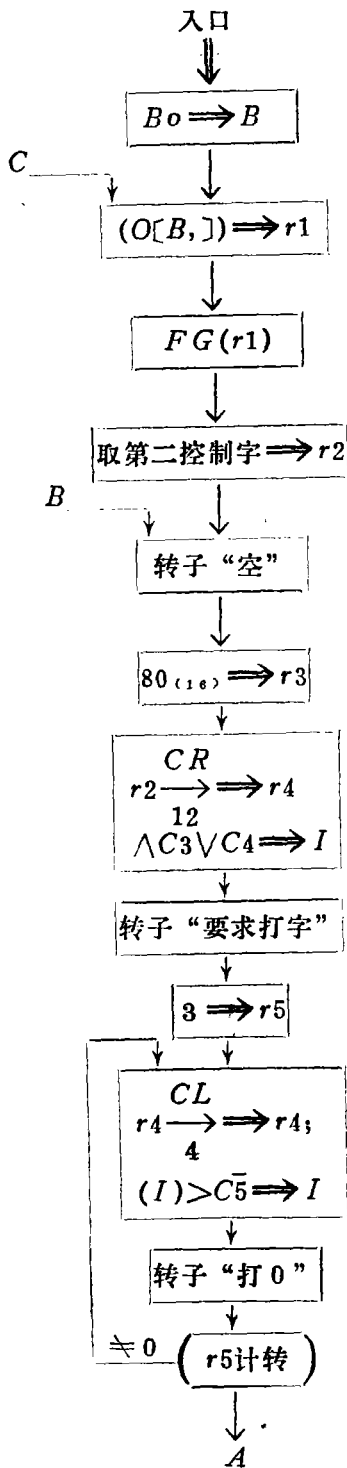
该类特征孔在产生压缩纸带时没有用处, 当欲对已压缩纸带作某些修改时, 或检查复制的纸带时, 尚具用处。

三、实现算法

显然, 按照二中定义的格式, 是无法由人工来直接实现穿孔的。

也很显然, 这一实现算法是一个很简单的非数学问题, 且电传机又具有穿孔输出的能力, 完全可以由计算机本身来自动地产生。

以下给出实现算法的框图：



B_0 给出欲穿孔之总段数及控制字表首地址

取出第一控制字

由鼓中读出欲穿孔输出之程序段

输出五个 00001

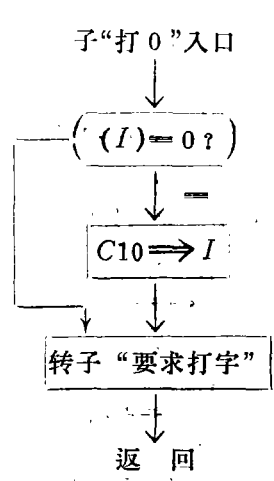
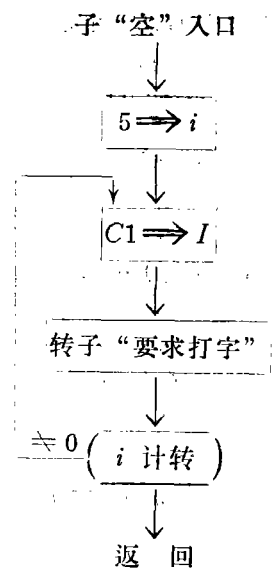
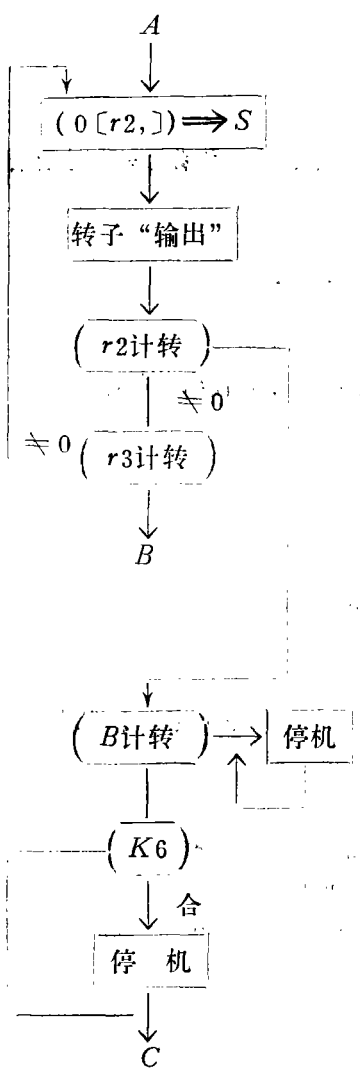
即每经 $80_{(10)}$ 个单元穿一个地址

产生并输出地址特征孔

C_3 : 000	0000	000	0003
C_4 : 000	0000	000	0004

C_5 : 00	0000	000	$000\bar{5}$
------------	------	-----	--------------

输出地址信息孔



C10: 000 0000 000 0010
 C1 : 000 0000 000 0001

子“输出”入口

全 0 ← (全 0 单元)

全 1 ← (全 1 单元)

左 θ ← (左 θ 指令)

右 θ ← (右 θ 指令)

左 D ← (左 D 指令)

特殊 ← (正常指令)

是 ↓

CL
S \wedge l1 \rightarrow VC8
2

要求打字

10 \Rightarrow i;
LL
S \rightarrow \Rightarrow I
2

F \rightarrow ↓
0 \Rightarrow I

H \rightarrow ↓
(I) \Rightarrow S;
(I) \Rightarrow S;
S \wedge l2 \Rightarrow I

D ↓

l1:138 0000 000 0000

l2:135 3800 000 0000

全 0 ↓

C 2 \Rightarrow I

E \rightarrow ↓
要求打字

返回 ↓

C 2:000 0000 000 0002

全 1 ↓

C 5 \Rightarrow I

E ↓

左 θ ↓

C18 \Rightarrow I
要求打字

CL
S \rightarrow \Rightarrow S
4

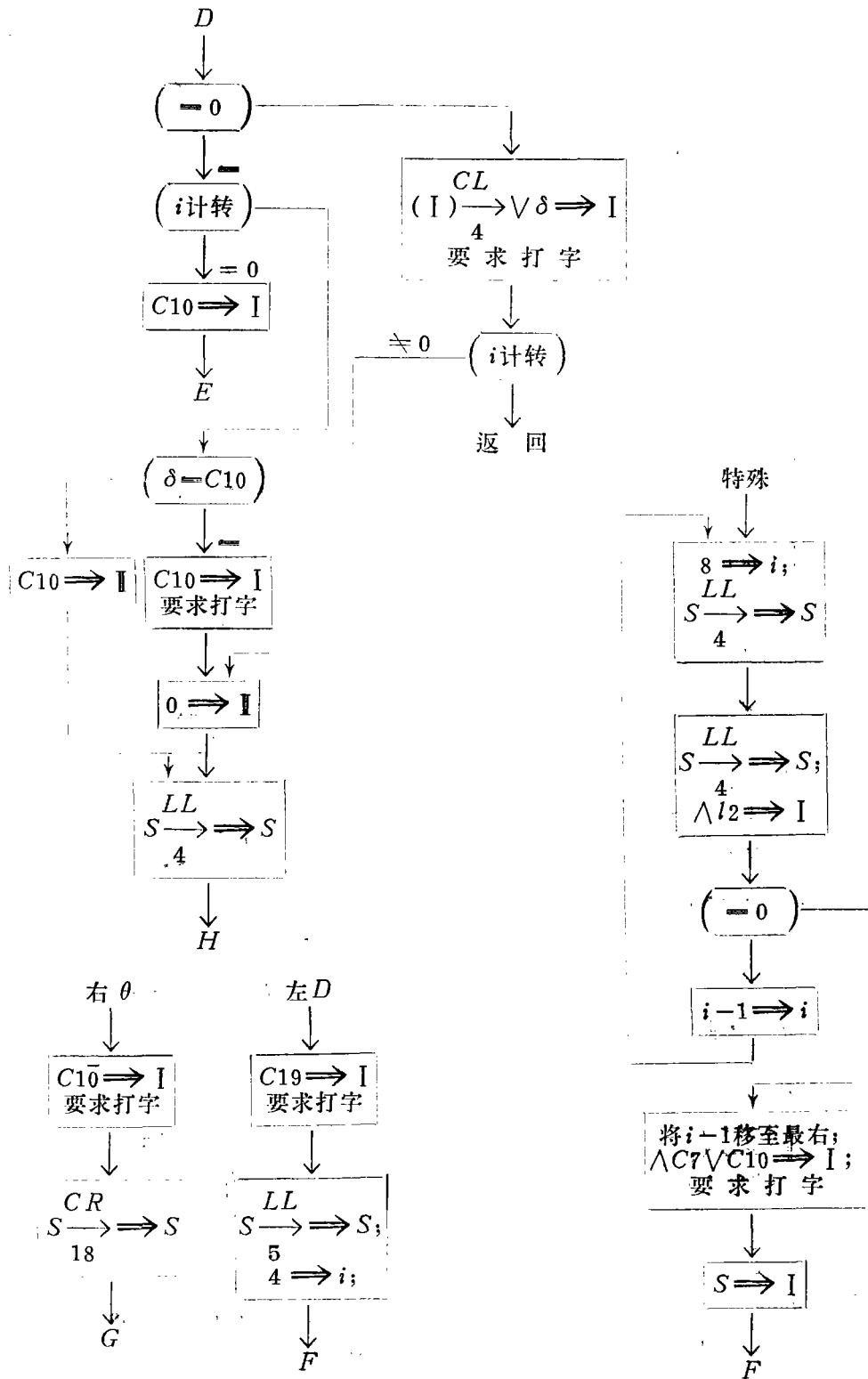
G \rightarrow ↓
S \wedge C 5 \Rightarrow I
“打 0”

CL
S \rightarrow \wedge C 5
4
“打 0”

返回 ↓

C 8:000 0000 000 0008

C18:000 0000 000 0018



框图说明

(1) 符号注解:

$d[B,]$: 表示有效地址 = $d + (B)_{\text{右址}}$;
(B)_{右址} + 变址值 $\Rightarrow B$; (37)

$d[B]$: 表示有效地址 = $d + (B)_{\text{右址}}$;

\rightarrow : 表示移位, 箭头上表示移位方式, 下面表示移位位数 (十进), 移位方式中第一字母表示移位类型 (L 表逻辑移, A 表算术移, C 表示环移), 第二字母表示移位方向 (L 表左移, R 表右移)。

(2) “要求打字”子程序的注解:

该子程序借用系统中动态子程序中的对应程序。故实际上在入口处需先调出动态子程序的后半部。

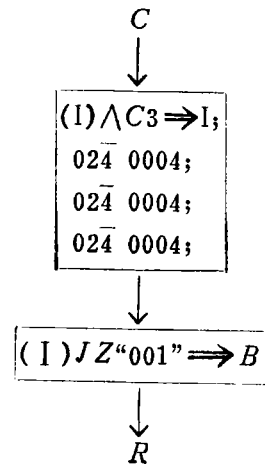
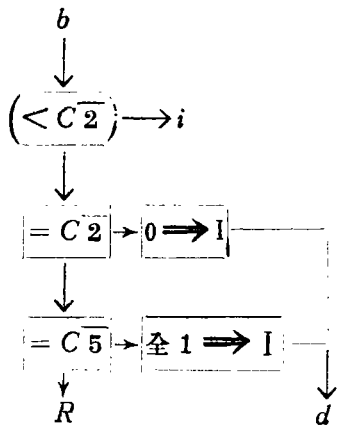
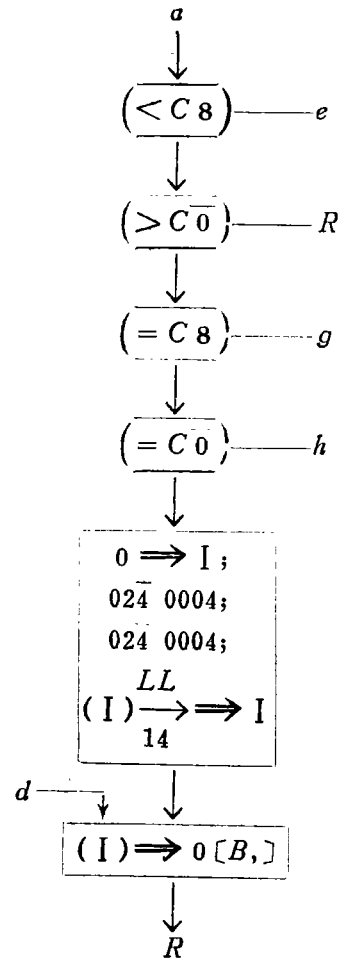
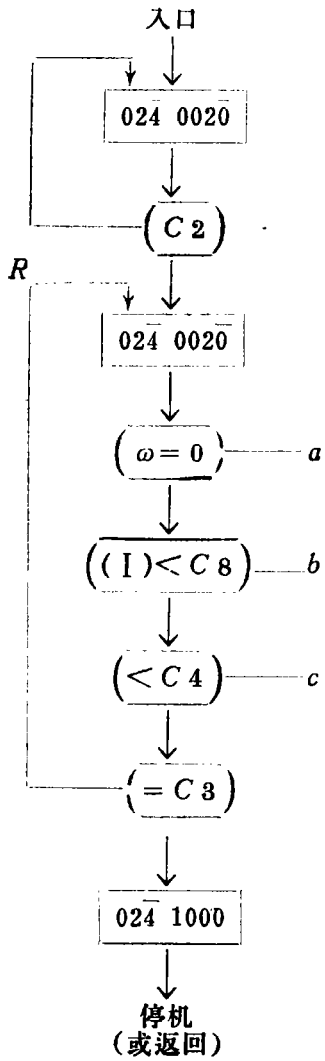
(3) 常数注解:

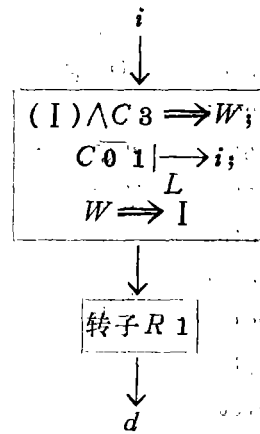
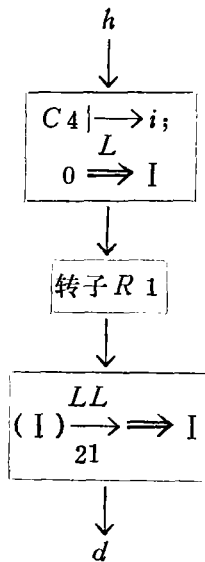
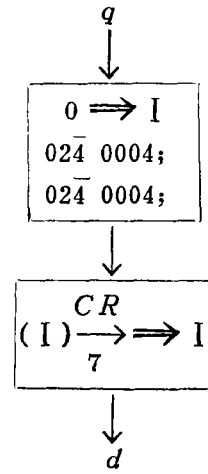
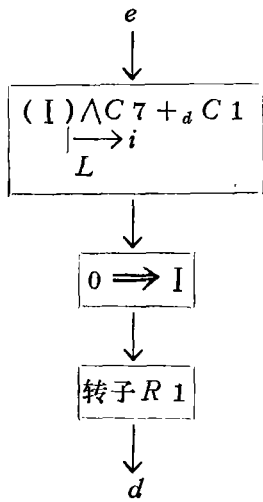
$l\ 1$:	125	3800	000	0000
$l\ 2$:	138	0000	000	0000
$C\ 1$:	000	0000	000	0001
$C\ 3$:	000	0000	000	0003
$C\ 4$:	000	0000	000	0004
$C\ 7$:	000	0000	000	0007
$C\ 8$:	000	0000	000	0008
$C\ \bar{2}$:	000	0000	000	000 $\bar{2}$
$C\ \bar{5}$:	000	0000	000	000 $\bar{5}$
$C\ 10$:	000	0000	000	0010
$C\ 18$:	000	0000	000	0018
$C\ 19$:	000	0000	000	0019

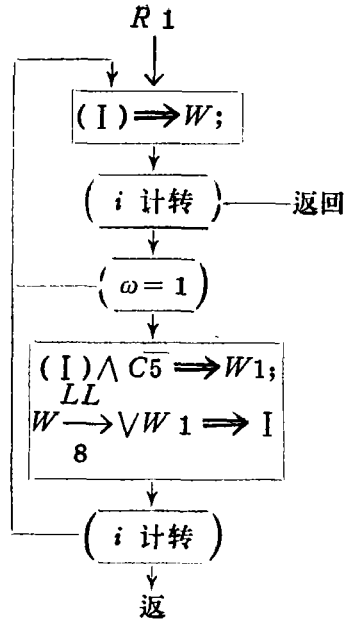
四、扩展算法

所谓扩展算法, 就是实现将压缩纸带输入内存的光电输入程序。

由压缩带的产生算法不难得到扩展算法的框图。(见下页)







框图本身很直观，不必多加解释，仅作以下注解：

符号 $a \xrightarrow{L}$ 表示填地址， $a \xrightarrow{i}$ 表示将 a 的右址填入 i 的左址 (30)；类似地， \xrightarrow{R} 为填右址 (31)。

常数 C^* 的给出类似于上一节。

附录：特征孔定义表

00000	硬件空码	10000	} 特殊指令	
00001	空	10001		
00010	开始、空	10010		
00011	结束	10011		
00100	} 地址	10100		
00101		10101		
00110		10110		
00111		10111		
01000	} 正常指令	11000		左 θ 指令
01001		11001		左 D 指令
01010		11010	右 θ 指令	
01011		11011		
01100	全 0 单元	11100	} 空 (待定义)	
01101	} 空	11101		
01110		11110		
01111		全 1 单元		11111