杜毅仁　李慕靖
王建中　陈启帆　编

16位微型计算机

SHI LIU WEI WEI
XING JI SUAN JI

——下册

上海交通大学出版社

十六位微型计算机

（下 册）

杜毅仁 李慕靖 王建中 陈启帆 编

上海交通大学出版社

内 容 简 介

本书从微型计算机的硬件、软件、系统结构等方面出发，通过 Intel 公司的 8086 系列以及 IBM 公司的个人计算机作为典型，进行系统的介绍和剖析，使读者获得有关微机的基础和应用知识。

全书共分上、中、下三册。上册通俗易懂地介绍了 8086 的硬件结构、指令系统及其程序设计语言，是全书的基础和核心部分；中册就整个 8086 系列，分别介绍了输入/输出处理器 8089、专用数值处理器 8087、操作系统固件 80130、8086 的改进型 80186 和 80286，同时，还对 MULTIBUS 这一标准的系统总线以及 8086 的其它配套系列芯片作了介绍；下册比较系统地分析了 IBM 公司的个人计算机，内容包括 PC 的硬件结构、系统固件及操作系统等。

本书具有硬件、软件相结合的特点，而且既能了解 8086，又能由此了解十六位微型计算机的一般原理及应用。因此，既可作为高等院校的教学参考书，也可作为研究院所、厂矿有关研究人员、工程技术人员乃至一般用户的工作手册。

十六位微型计算机

（下册）

杜毅仁　李慕靖　王建中　陈启帆　编

上海交通大学出版社出版

淮海中路 1984 弄 19 号

新华书店上海发行所发行

常熟文化印刷厂排版、印刷

开本：787×1092 毫米　1/16　印张 40.25　字数 998000

1985 年 6 月第一版　1986 年 1 月第 2 次印刷

印数：10000—20000

统一书号：15324·35　　科技书目：114—236

定价：7.36 元

下 册 目 录

第三篇 IBM 个人计算机系统

第 三 篇

IBM 个人计算机系统

第十七章 绪 论

1981 年 8 月，美国 IBM 公司宣布，该公司个人计算机(IBM PC)问世。这一信息，对于世界计算机制造领域，是一个不小的震动。如果说，在此之前，微机的研制、制造和应用，还停留在较狭窄的范围内，还没有引起人们的足够重视，那么，IBM 公司的介入，使得微机完全得到了“公认”和“合法化”了。

今天，IBM PC 几乎已成为社会上广泛应用微机的一个标准，它占领了当今相当大的微机系统销售额，各有关公司、厂商不断推出与它相兼容的硬件、软件或器件。IBM 的某些规范，通过 IBM PC 成为微机领域的习用标准。这种现象，很值得我们注意和重视。

一、博采众长，尽快推出新机型

IBM 公司看准了微机的发展趋势，即只有微机才能将电子计算机技术广泛应用于各行各业、千家万户，而谁能捷足先登，谁就能站稳脚跟占领市场。为此，IBM PC 不是在零部件上另搞一套，而是采用别家公司、厂商现成的原件来装备自己，如 Epson 公司的打印机、Intel 公司的 8088CPU 芯片、Tandon 公司的磁盘驱动器；其系统程序也是同样，如 Microsoft 的 MS-DOS 操作系统、信息无限公司的字处理程序等，从而，使整个 PC 的开发周期大大缩短，据报道仅用了十三周时间。

由于同一原因，使 IBM PC 的硬件资料随之公开，这就使 PC 的应用软件，有可能让其它专业软件公司生产和供应，反过来又使 PC 的软件配置，丰富于其它微机系统。

这是一种充分发挥社会大生产优势的方法。

二、IBM PC 本身反映了微机从 8 位向 16 位发展的趋势

IBM PC 采用 Intel 8088 作为中央处理器，而 8088 是 16 位 CPU 的 8 位版本，就 8088 指令系统来说，它完全与 16 位 CPU 8086 及比 8086 更高级系列芯片 80186、80286 兼容或向上兼容，在 PC 上开发的软件或为 PC 开发的软件，可以在将来原封不动地搬到数据、地址总线均为 16 位 CPU 组成的系统上去。再者，8088 的数据线(包括 I/O 通道的数据线)与现今普遍使用外设的数据宽度匹配，因此 IBM PC 机外设及接口能在已有、种类繁多的外设及接口中选用。这种做法一方面使制造商无需耗费很大的代价研制 16 位数据线的外设和接口，另一方面也使用户感到 IBM PC 的外设资源是丰富的，乐于使用。在某种程度上可以说，IBM PC 起到了一种承上启下的作用。

三、PC 的扩充性较好

支持 PC 的操作系统，目前不少于五种，其中有较简单、非常适合一般用户的 PC-DOS 和 CP/M-86，有适合事务处理的 OASIS-16，有以可移植性见长的 UCSD P-System，有支持多用户、多任务的 Unix. 用户很容易在这些操作系统支持的高级语言或提供的功能中，找到满足自己要求的语言或实用程序，这也为软件制造商开发各类面向用户的软件包，提供了有利的条件。

另外，PC 有五个外设扩展口，用户能随时更换外设，如连上单色显示器、彩色/图形显示器、打印机、扩展 RAM 板、异步通讯口、软盘机或游戏器；用户也可用目前正在研制、将来出售的具有更高性能的 I/O 设备，更新原来的老设备，扩充提高 PC 整机的性能。如果在槽口上上插一块通讯电路板，PC 就能从一个独立的设备成为大型机的一个终端或局部网络的一个结点机。

四、PC 在浮点数处理方面占有优势

PC 母板上留有一个 Intel 8087 浮点数处理器的插座，用户只要插上一块 8087 处理器片，PC 的浮点数运算就完全由硬件承担了。而且在 8087 进行浮点数运算的时候，主 CPU 8088 还能继续取指令，执行非浮点数操作，从而使 8088、8087 能协同工作，这便是协处理器概念。8087 的存在使 PC 具备相当的资格胜任多维图形显示、处理等进行大量高精度数值运算的工作，这些工作本来是要由高级小型机承担的。

下面从几个角度，将 PC 与其它型号的个人计算机作一比较，通过比较，我们能在其内存容量、整机性能、价格、高级语言适应能力方面对 PC 有个初步的认识。

1. IBM PC 与 IBM System/23、Display Write™ 微型计算机的比较。

IBM System/23 (DataMaster) 是第三代台式商用计算机，能代替 5100 系列机。IBM system/23 能进行字处理及传统的数据处理工作，其 RAM 区容量在 32K—128K 之间。

Display writer 是 IBM 公司夺回其在字处理方面优先地位的新产品，它以 8086 作 CPU，RAM 区可扩充至 256K，配有 Textpack™1、2、3、4 软件包，操作系统是 CP/M-86™

IBM 公司微型计算机硬件配置比较表

型　号	字　长	最小配备			最大配备		
		RAM	外存	价格	RAM	外存	价格
PC	16/8*	16K	磁带	$1565	256K**	320K	$4515***
system/23	8	32K	250K	$4630	128K	4,400K	$12995
Display writer	16	160K	250K	$5095	256K	2,000K	$8670

* PC 的数据线是 8 位，内部以 16 位工作.

** 配以 256K RAM 的话，PC 可接的外设：打印机、显示器、软盘机. 现已可将 RAM 区扩充到 544K.

*** 为 83 年价格

结论

(1) PC 适应能力较广，最小配备就能满足教育，娱乐应用，最大配备时的 RAM 容量与 Display writer 相同，但价格比它低得多。

(2) System/23 的字长只有 8 位，计算机能力不强，但其外存容量很大。

(3) 配有 IBM 软件如 Textpack 的 Display writer，可作为通用计算机使用。

(4) 只要花很少的钱，就能将 PC 扩充成具有 Display writer 相似运算能力的计算机。

2．IBM PC 与其他个人计算机关于最小、最大配套的比较

个人计算机的一个主要特点是可扩充性，人们决定以最低价格购买机器时，会关心机器

最小配备比较表

型　　号	显示帧	CPU	最小 RAM		最大扩充限制	
			RAM(K)	价　格	RAM	磁盘容量
Commodore VIC 20	22×23	6502A	3.5K	$330***	32K	340K
TRS-80Color Computer	32×16	6809	4K	$399	16K	N/A
Atari 400	40×24	6502	8K	$399	16K	N/A
Ohio Scientific C1P	24×24	6502	4K	$479	32K	160K
TRS-80Model III	64×16	Z-80	4K	$699	48K	313K
Ohio Scientific C4P	64×32	6502	8K	$879	24K	160K
Ohio Scientific C8P	64×32	6502	8K	$895	32K	500K
Atari 800	40×24	6502	16K	$899	48K	176K
Commodore PET	40×25	6502	16K	$995	32K	2000K
Apple II	40×24	6502	16K	$1330	64K	286K
Commodore CBM	80×25	6502	32K	$1495	32K	2000K
IBM PC	80×25	8088	16K	$1565	256K	320K
NEC 8000	80×24	Z-80A	32K	$1600	64K	328K

最大配备比较表

型　　号	RAM	显示帧	软盘机总容量	软盘尺寸(吋)	价　格
Commodore PET	32K	40×25	330K	5	$2290***
Commodore CBM	32K	80×25	330K	5	$2790
Hewlett Packard 85	32K	32×16	540K	5	$2945
Commodore PET	32K	40×25	2000K*	8	$3090
Commodore CBM	32K	80×25	2000K*	8	$5390
TRS-80 Model III	48K	64×16	334K	5	$2495
Atari 800	48K	40×24	176K	5	$2456
Apple II	64K	40×24	286K	5	$3024
Northstar Advantage	64K	80×24	720K	5	$3999
ZenithZ-90-80	64K	80×24	1380K*	5	$4890
Zenith Z-89-80	64K	80×24	2400K*	8	$6440
Hewlett Packard 125	64K	80×26	2400K	8	$10550
Commodore 8096	96K	80×25	2000K*	8	$4090
IBM System/23	128K	80×24	2200K	8	$8730
TRS-80Model II	128K	80×24	972K*	8	$5049
AppleIII	256K	80×24	286K*	5	$4979
IBM PC	256K**	80×25	320K	5	$5235

* 可接硬盘机

** 用了打印机和显示器后，就不能接通讯转接器或彩色/图形监视器转接器

*** 为 83 年价格

的最小配备，为使机器具有较高工作能力时，又要考虑机器允许的最大配备。

3．支持数据表软件 Visi Calc 的能力比较

数据表软件 Visi Calc 是一种用计算机编制报表的应用生成程序，无编程经验的用户，也能在较短的时间里掌握它。Visi Calc 的功能包括建立数据表、项目、数值、计算数据、复制数据表，以及多种模式产生报表、打印存盘数据表等。

Visi Calc 对系统的要求是：有较大的工作表区间(RAM 区)，较宽的显示屏幕，较高的运算速度。

从下一章开始，我们将详细地介绍 IBM PC 的硬件、系统软件以及一些富有特色的应用程序，体现本书"从芯片到系统"的宗旨。另外也应看到 IBM PC 本身是在不断改进着的，有关 PC 的资料也多，在这里不可能面面俱到。总的说来，PC 的硬件、BIOS、扩展槽等基本部分是不会变的，因为他们毕竟反映出 IBM PC 的基本特征。如不特别说明，下面章节出现的 PC 就指 IBM PC。

上述 PC 的各种特点，对我们不无借鉴之处。

Visi Calc 最小配备比较表

型　　号	字　长	RAM	工作表空　间	显示帧	盘容量	价　格	Visi Calc 价格	总价格
Atari 800	8	32K	6K	40×24	88K	$1757	$200	$1957
Commodore PET	8	32K	10K	40×25	170K*	$1690	$200	$1890
Commodore CBM	8	32K	10K	80×25	170K*	$2190	$200	$2390
TRS-80Model III	8	32K	10K	64×16	167K	$1995	$100*	$2095
Howlett Packard 85	8	32K	13K	32×16	200K	$2945	$200	$3145
Apple II	8	48K	18K	40×24	143K	$2334	$200	$2534
IBM PC	16	64K	23K	80×25	160K	$4045	$200	$4245
TRS-80Model II	8	64K	25K	80×24	486K*	$3899	$300	$4199
Hewlett Packard 125	8	64K	28K	80×26	512K	$6250	$200	$6450
Apple III	8	128K	71K	80×24	143K*	$3654	$250	$3904

* 可接硬盘机

Visi Calc 最大配备比较表

型　　号	字　长	RAM	工作表空　间	显示帧	磁盘容量	价　格	Visi Calc 价格	总价格
Commodore PET	8	32K	10K	40×25	2000K	$3090	$200	$3290
Commodore CBM	8	32K	10K	80×25	2000K	$3590	$200	$3790
Hewlett Packard 85	8	32K	13K	32×16	540K	$2945	$200	$3145
TRS-80Model III	8	48K	18K	64×16	334K	$2495	$200	$2695
Atari 800	8	48K	22K	40×24	176K	$2456	$200	$2656
Hewlett Packard 125	8	64K	28K	80×26	2400K	$10550	$200	$10750
Apple II	8	64K	34K	40×24	286K	$3024	$200	$3224
TRS-80Model II	8	128K	89K	80×24	972K*	$5049	$300	$5349
Apple III	8	256K	196K	80×24	286K*	$4979	$250	$5229
IBM PC	16	256K	214K	80×25	320K	$5235	$250	$5435

* 可接硬盘机

注：价格为 83 年的

第十八章 主 机

§18.1 系 统 板

系统板是一块 8.5″×11″ 通道与焊接区一一对应的多层印刷板，也称母板。系统板上装有 8088/8087 CPU 系统、ROM 区、RAM 区、I/O 通道和集成 I/O 转接口共 5 个功能区组成的系统单元。直流电源及相应的信号线，通过两个 6 脚接口送入系统板；键盘、录音机、扬声器也由接口连上；系统 I/O 通道，由导线跨连在系统板的五个 62 脚 I/O 扩展槽上。这些 I/O 插槽上，安有 16 个(只用其中 13 个)双向 DIP 开关，系统软件从这些开关得到外设选件的配置情况，如内存和外部扩展 RAM 区的容量、显示器类型、显示器帧尺寸及系统配备驱动器的数量。下面介绍系统板上的四个功能区，I/O 通道专门在 §18.3 节介绍。图 18-1 展示了五个功能区和他们的连接情况。

处理器功能区

IBM PC 采用 Intel 公司 8088 处理器作 CPU。8088 是 16 位处理器 8086 的变体，它只有 8 根数据线，但运算速度比 8 位的 CPU 快 3 至 5 倍。8088 有 20 根地址线，直接寻址 1 兆字节。安装时采用最大模式，能增加一台 8087 协处理机进行快速单指令浮点运算。8088 指令集与 8086 指令集完全兼容。IBM PC 上的 8088 时钟频率为 4.77 MHz，时钟信号由 14.31813 晶体振荡信号 3 分频后得到，该振荡信号的 4 分频 (3.58 MHz) 供彩色显示器作色彩分段信号。8088 的总线周期为 4 个时钟周期 (840ns)，单位通道周期为 5 个时钟周期 (1.05 ms)。

为便于 IBM PC 主机与高级外设的连接，CPU 功能区提供了 4 个 20 位的 DMA 通道、3 个 16 位计时器通道和 8 级中断。4 个 DMA 通道中的 3 个，用于 CPU 与 I/O 外设的高速数据交换，且这种交换沿 I/O 总线进行，另一个 DMA 通道用于系统动态存贮器的刷新；即为计时计数器编制周期性请求哑 DMA 通讯的程序，利用哑 DMA 请求产生刷新系统板存贮器及外接存贮器的读周期。处理器“准备好”线有效时，一个 DMA 传送周期占五个时钟周期，一个刷新 DMA 周期占 4 个时钟周期。

3 个计时/计数器通道的分配情况是：通道 0 是系统通用计时器，向时间时钟提供基准信号；通道 1 为 DMA 请求和计时刷新周期；通道 2 是扬声器音调发生器的计时器。3 个计时通道的最小计时分辨能力为 1.05 μs。

8 级中断的安排情况是：中断 0、中断 1 供系统板使用，其中最高级别的中断 0 接 1 号计时/计数通道，接收计时器发出的周期性中断信号；中断 1 接至键盘转接器，接收键盘发来的扫描代码中断信号。中断 2～7 分配给 I/O 扩展槽。8088 不可屏蔽中断 NMI，用来报告存贮器的奇偶校验错误。

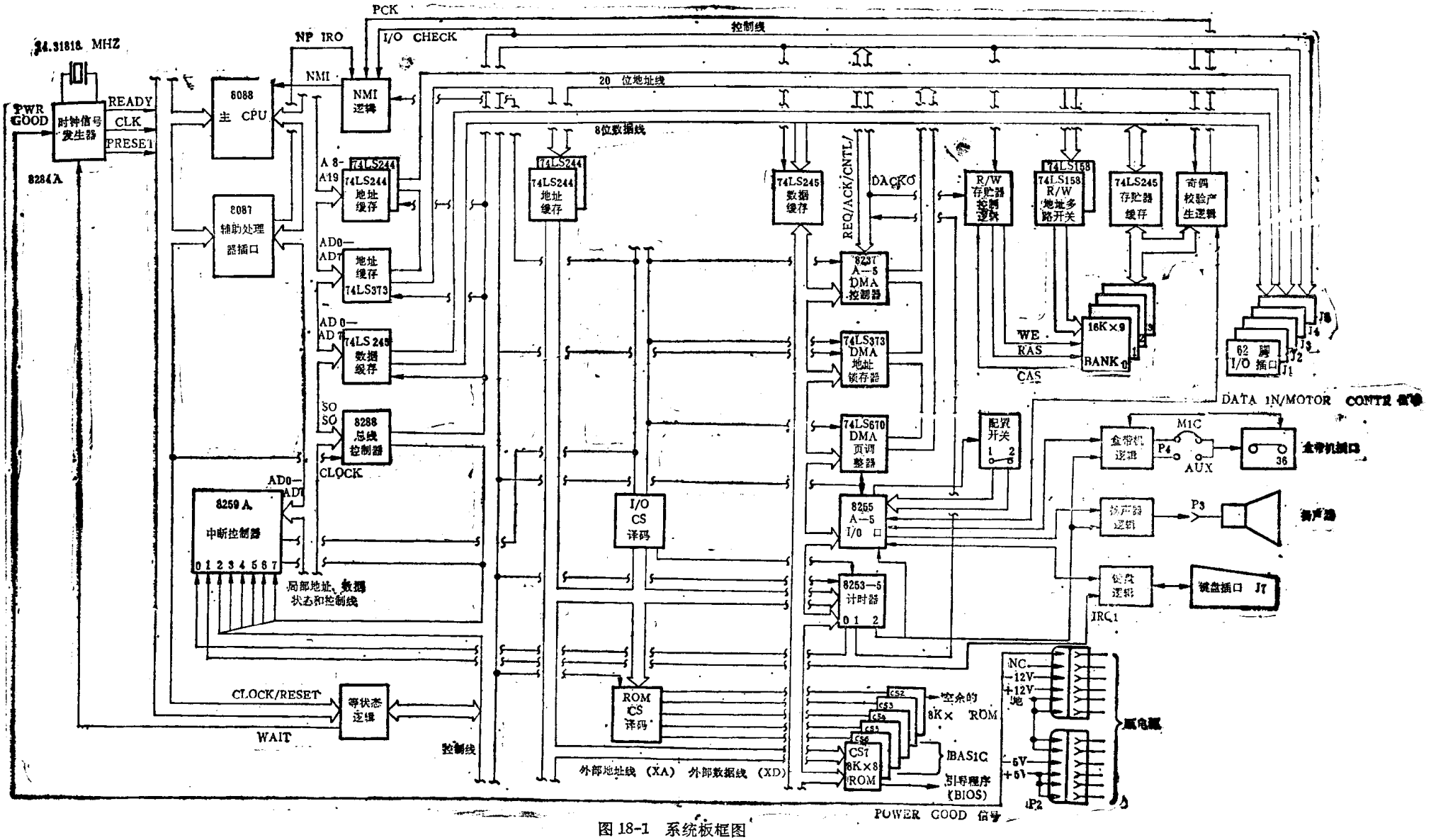

PCK
控制线
14.31818 MHZ
NP IRQ
I/O CHECK
NMI
20 位地址线
PWR GOOD
时钟信号发生器
READY
CLK
PRESET
8284A
8088 主 CPU
NMI 逻辑
8087 辅助处理器插口
8位数据线
74LS244 地址缓存
74LS244 数据缓存
地址缓存 74LS373
74LS245 数据缓存
8237 A—5 DMA 控制器
74LS373 DMA 地址锁存器
74LS670 DMA 页调整器
8288 总线控制器
8259 A 中断控制器
0 1 2 3 4 5 6 7
局部地址、数据状态和控制线
I/O CS 译码
ROM CS 译码
8255 A—5 I/O 口
8253—5 计时器
0 1 2
等状态逻辑
CLOCK/RESET
WAIT
控制线
外部地址线 (XA) 外部数据线 (XD)
REQ/ACK/CNTL/
DACK0
R/W 存贮器控制逻辑
74LS158 R/W 地址多路开关
74LS245 存贮器缓存
奇偶校验产生逻辑
16K×9
BANK 0
WE
RAS
CAS
62 脚 I/O 插口
DATA IN/MOTOR CONTR 信号
配置开关
盒带机逻辑
M1C
P4
AUX
盒带机插口
扬声器逻辑
P3
扬声器
键盘逻辑
键盘插口 J7
IRQ1
NC
-12V
+12V
地
-5V
+5V
扩展槽
P2
空余的 8K× ROM
BASIC 引导程序 (BIOS)
8K×8 ROM
POWER GOOD 信号

图 18-1 系统板框图

ROM 区及 RAM 区

系统板上留有安装 6 块 8K×8 ROM 或 EPROM (共 48 KB) 和 4 块 16K×9 RAM (共 64 KB) 的空间 (插口)。48 KB ROM 占据整个内存区 976K 至 1008K 高地址空间，64 KB RAM 占据 0～64 KB 的低地址空间。48KB ROM 中的头 40 KB 固化有基本输入输出系统 BIOS 和盒带 BASIC 解释程序，40～48 KB ROM 区是空白的。0～64 KB RAM 区供用户存放程序和数据。当实际 RAM 区只有 16 KB 时，系统处于最小系统状态，这时，若调用盒带 BASIC 解释程序，用户程序可占用 12 KB 的 RAM 区、剩余 4 KB 供系统软件用。如果要求用户 RAM 区超过 64 KB 的话，必须利用 I/O 扩展槽进行 RAM 区的外部扩充。

插于系统板的 8K×8 ROM 片子的访问时间是 250 ns，周期为 375 ns，16K×9 RAM 片子的访问时间是 250 ns，访问周期 410 ns，受奇偶校验。

集成 I/O 转接器

集成 I/O 转接器指连接盒带录音机、串行键盘和扬声器的内部转接器。盒带录音机转接器的主要输出端有二个：一是 0—1 电平输出端，它与录音机的话筒或线路输入端相连；另一是受程序控制、启停录音机马达的控制端。转接器读写磁带的波特率为 1000～2000。串行键盘转接器用来连接键盘串行接口，它在收到键盘扫描完成标识码后，向处理器发中断信号(中断 1)。以上两个转接器通过垂直安装在主机背板上的 5 脚 DIN 插座与外界联系。

扬声器转接器通过 4 脚管座以及一双线插头连接 $2^1/_4$ 吋扬声器，扬声器的驱动方式有三种：第一种为程控型，即寄存器中的一位由程序控制发出脉冲串；第二种为间接程序型，即程序控制第 2 通道计时/计数器，由它反馈给扬声器波形信号；第三种也是间接程控型，程序控制 I/O 寄存器的一位以调制形式传送给计时/计数器的时钟信号。这 3 种方式能同时使用，产生音频范围为 37～32000 Hz 的复合音频信号。

§18.2 系统板内部接口

系统板内部接口指键盘接口、扬声器接口和盒带录音机接口。在介绍键盘接口前先介绍一下键盘。

IBM PC 的键盘是同系统单元(主机)分离的外部设备，由一根五芯电缆(内有一根 +5V 电源线、一根地线、二根双向信号线、一根时钟线)与主机相连。键盘上的 83 个键分为 3 个组：处于中央组的键排成标准打字机键盘样式，右边组是 10 个功能键、用户能用软件定义它们的功能，右边组含有 16 个键，用于数字输入、屏幕光标位置设定，编辑及运算等目的。

键盘接口主要向键盘转接器提供键的接通状态，同时允许用户 (系统) 用软件定义键的工作方式如转换键的状态，产生多个同种字符等。值得注意的是从键盘接口发出的是键的扫描代码而不是 ASCII 码。扫描代码分为接通扫描代码和阻断扫描代码，后者的码值等于前者码值加 X′80′。除控制键外的所有键均能产生数个同型字符的扫描代码。

IBM PC 的键盘是智能终端，内部的 Intel 8408 微处理器负责接通键盘电源、扫描键盘、阻止键的反跳、缓存 20 个扫描代码、与系统单元进行双向通讯、执行传送扫描代码必需的信号交换协议等工作。图 18-2 是键盘接口框图，图 18-3 是键盘按键分布图，图 18-4 是键盘接口插座引脚图。表 18-1 是键盘扫描代码表。

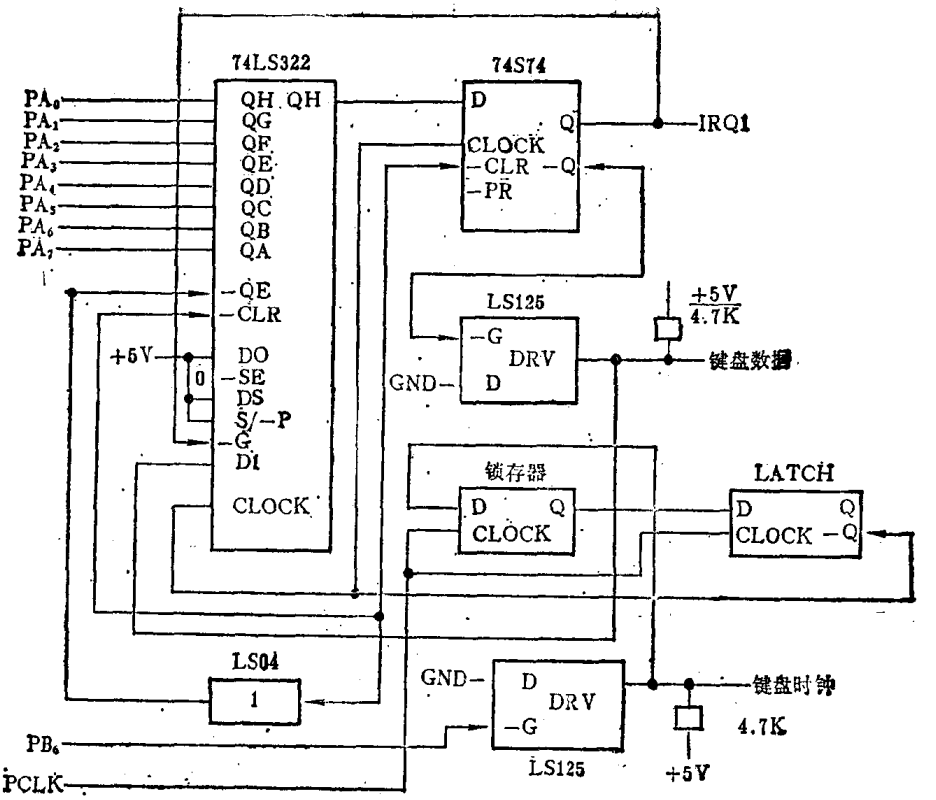


图 18-2　键盘接口框图

表 18-1　键盘扫描代码表

键　号	扫描代码	键　号	扫描代码	键　号	扫描代码	键　号	扫描代码
1	01	43	2B	22	16	64	40
2	02	44	2C	23	17	65	41
3	03	45	2D	24	18	66	42
4	04	46	2E	25	19	67	43
5	05	47	2F	26	1A	68	44
6	06	48	30	27	1B	69	45
7	07	49	31	28	1C	70	46
8	08	50	32	29	1D	71	47
9	09	51	33	30	1E	72	48
10	0A	52	34	31	1F	73	49
11	0B	53	35	32	20	74	4A
12	0C	54	36	33	21	75	4B
13	0D	55	37	34	22	76	4C
14	0E	56	38	35	23	77	4D
15	0F	57	39	36	24	78	4E
16	10	58	3A	37	25	79	4F
17	11	59	3B	38	26	80	50
18	12	60	3C	39	27	81	51
19	13	61	3D	40	28	82	52
20	14	62	3E	41	29	83	53
21	15	63	3F	42	2A		

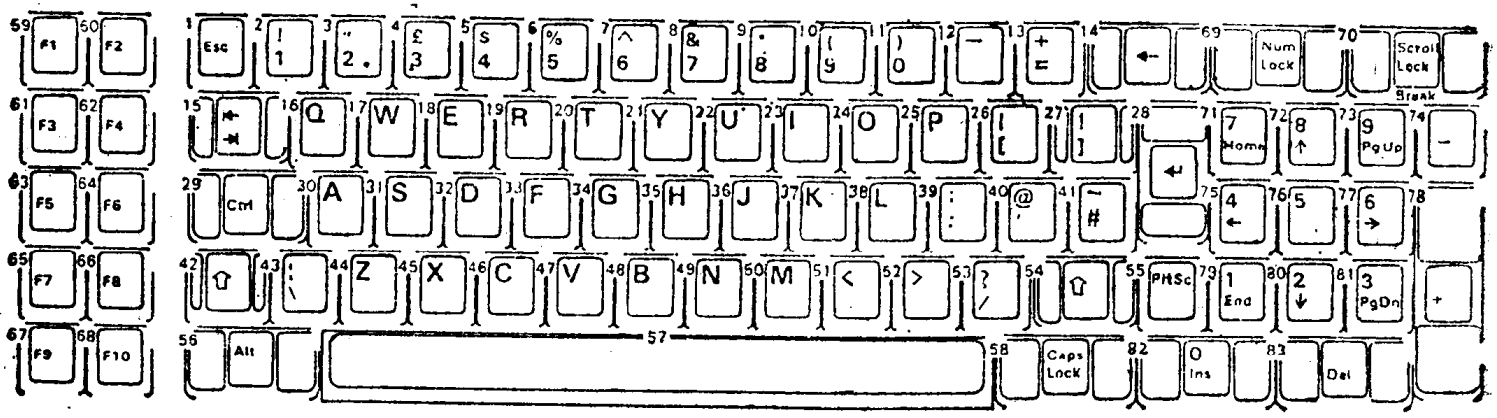


图 18-3 按键分布图(*a*)：美式键盘

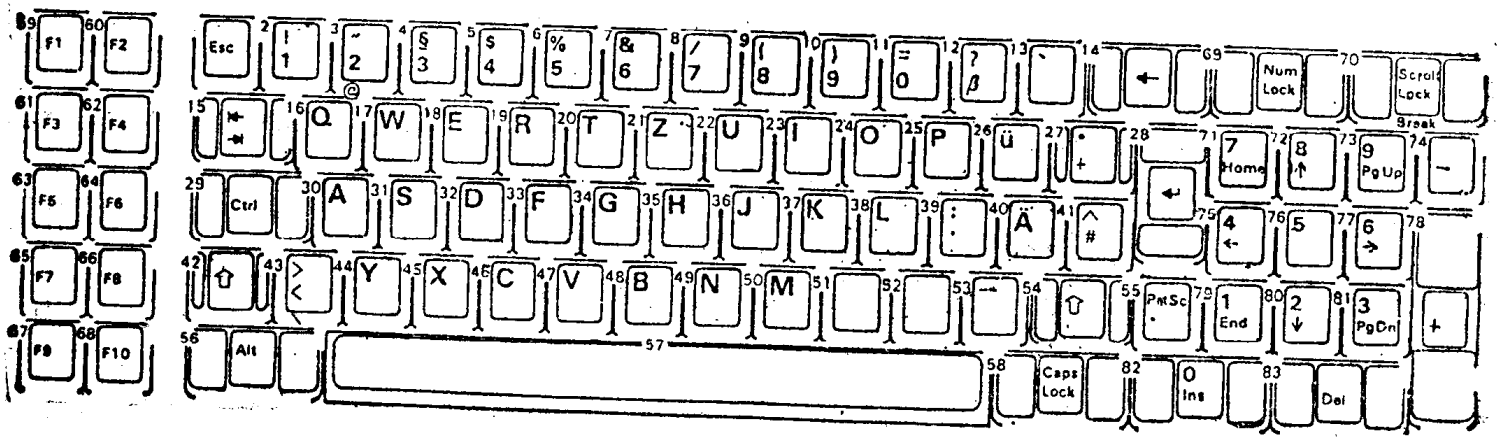


图 18-3 按键分布图(*b*)：英式键盘

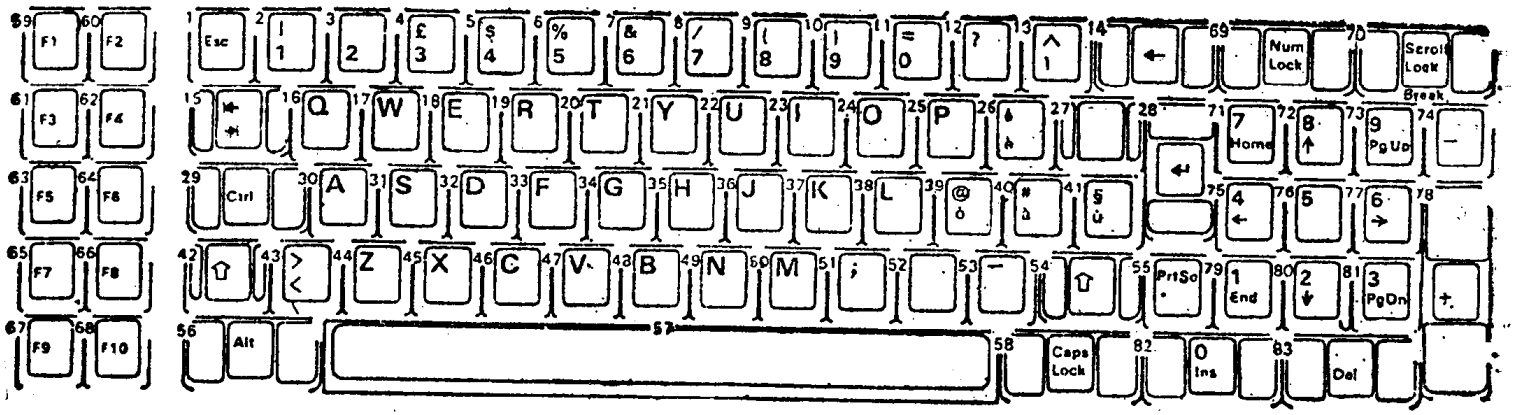


图 18-3 按键分布图(c)：意大利式键盘

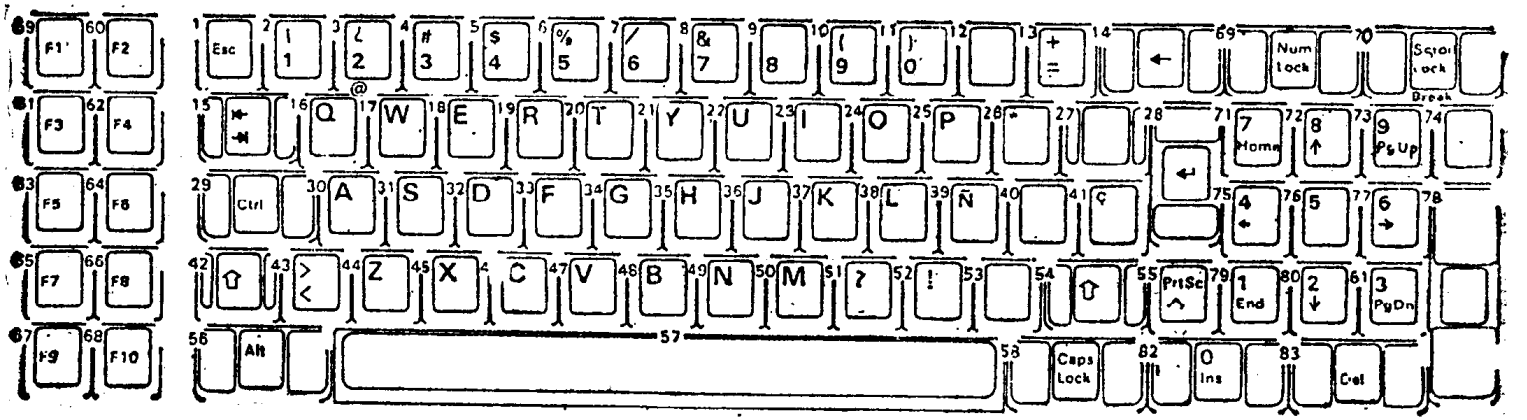


图 18-3 按键分布图(d)，西班牙式键盘

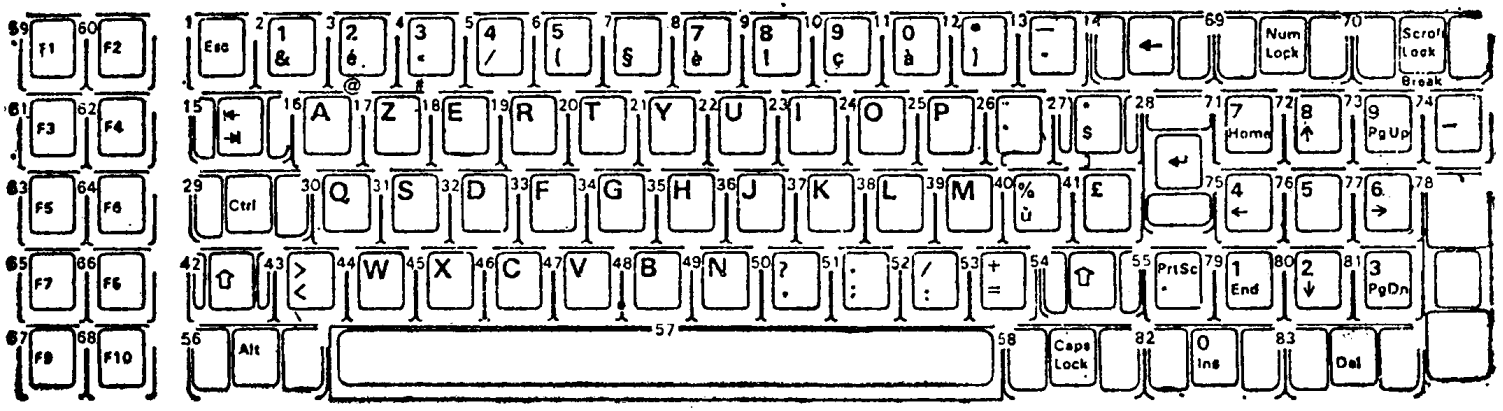


图 18-3 按键分布图(*e*)：法式键盘

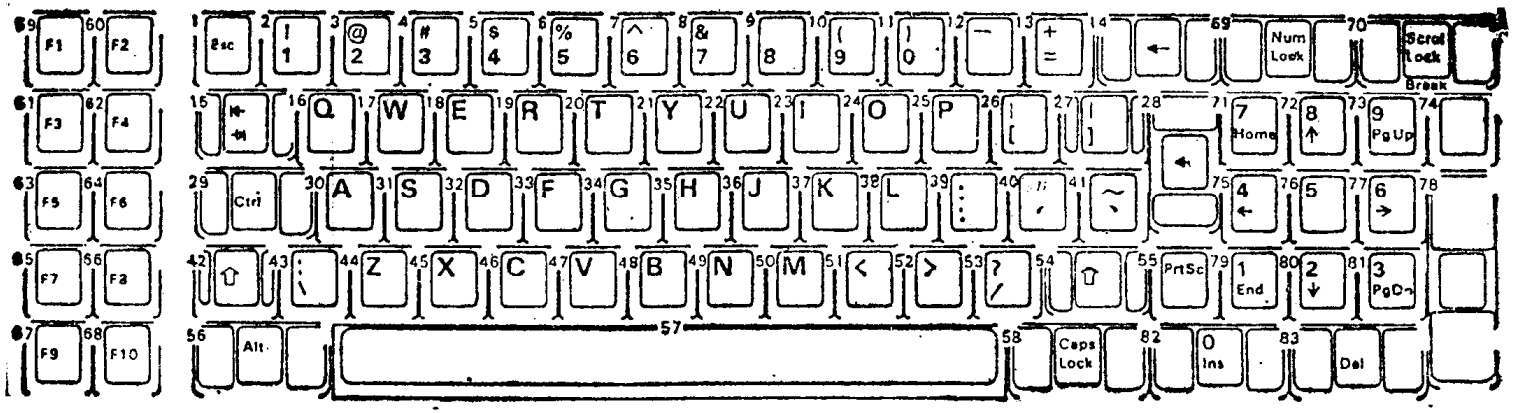


图 18-3 按键分布图(*f*)：德式键盘

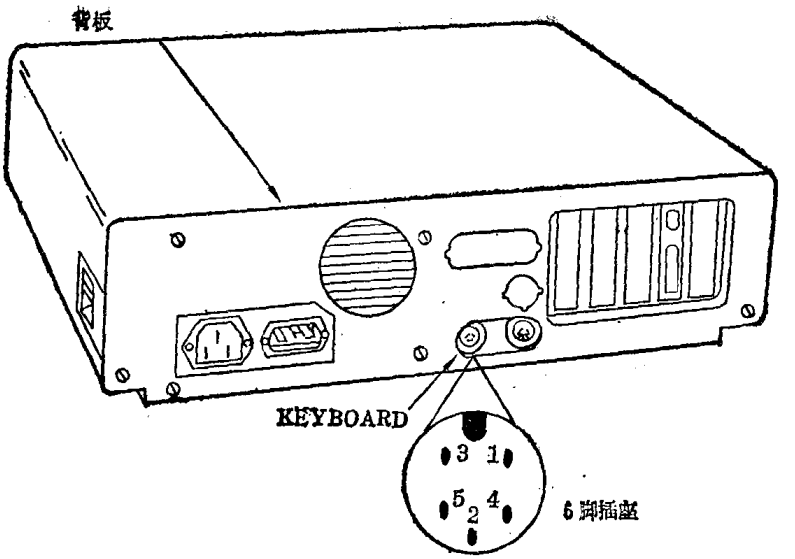


图 18-4　键盘接口插座引脚

引　脚	信　号
1	+键盘时钟
2	+键盘数据
3	−键盘复位(键盘不用)
4	地
5	+5V

扬声器接口接收从扬声器转接器送来的音频信号，这些音频信号来自：1．8255A 与 PPI 第 1 输出位，I/O 地址 X′0061′。2．计时器通道计时位，该时钟的一个输入端是 8253-5 计时器的 1.19 MHz 信号，另一端接 8255A 与 PPI 输出位，该通道时钟受程序控制。图 18-5 为扬声器驱动系统方框图，其中 CLKOUT 2 输出合成音频信号。表 18-2 是扬声器 4 脚插座引脚说明。

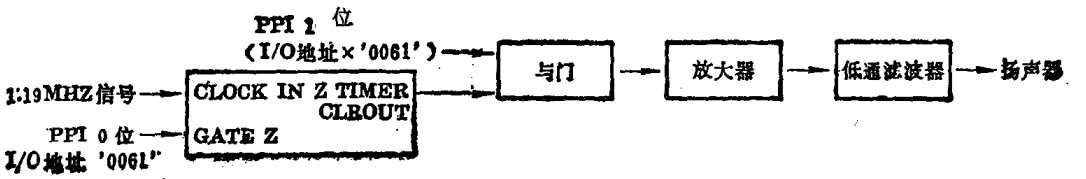
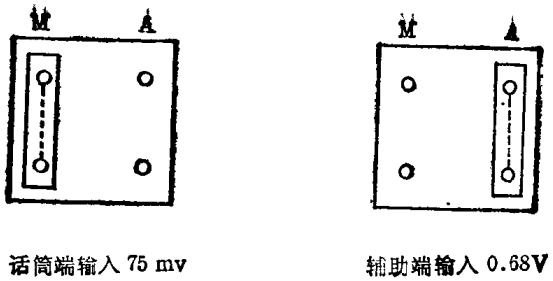


图 18-5　扬声器驱动系统框图

表　18-2

引　脚	作　用
1	数　据
2	键
3	地　线
4	+5V



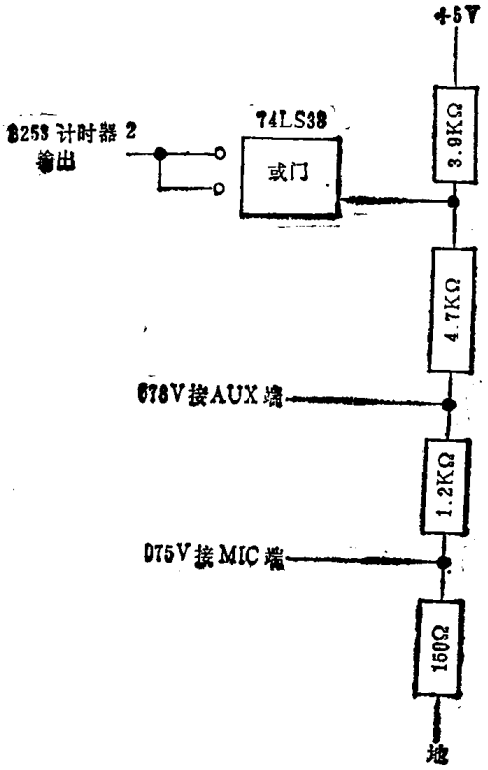


图 18-6　盒带机数据写入接口线路

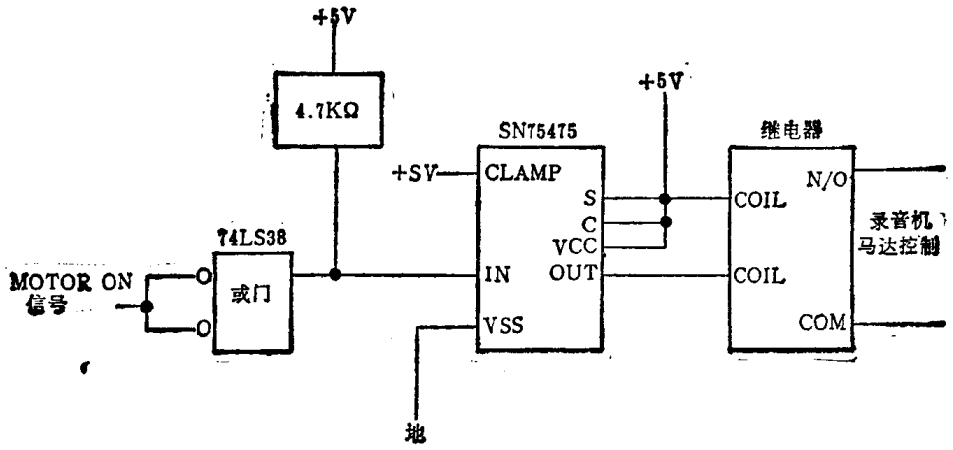


图 18-7　盒带机马达控制线路

盒带机接口也受程序控制，系统板 8253 计时器的 0—1 电平方波信号由 DIN 插座的第 5 脚送给录音机，录音机磁带输出信号从插座的第 4 脚送至 8255 A-5 可程序外设接口的输入口。录音机马达由盒带机接口插座第 1、第 3 引脚上的电平控制，而这两个电平信号又受 8255 A-PPI (I/O 地址 61 N)输出口第 3 位控制。8255 A 地址及 I/O 口分配见表 18-6。此外盒带机飞线脚上的跨接线选定录音机是话筒端输入还是辅助端输入。飞线脚接线图见 13 页最上图。

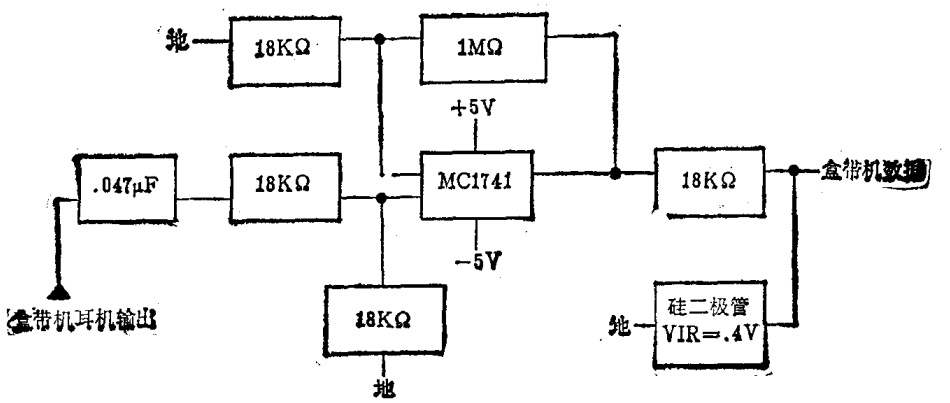


图 18-8　盒带机数据读取接口线路

§18.3　I/O　通　道

I/O 通道(见图 18-9) 是 8088 处理器总线的扩充，具有多路传送、电平再驱动、6 级中断、DMA 等特点，I/O 通道上有 8 根双向地址线、20 根地址线，6 根中断线、数根存贮器及 I/O 读写线、1 根时钟分时线、3 根 DMA 通道控制线，通道检查线、各转接器的电源线和地线。I/O 扩展槽上有 ±5V、±12V 四种直流电压。

I/O 通道连接慢速 I/O 设备或存贮器的"准备好"线的作用是：如果被寻址的设备未使"准备好"线有效的话，处理器发出的存贮器读写周期将占 4 个时钟周期(840ns/字节)，DMA 传送和 I/O 读写周期将占 5 个时钟周期(1.05 μs/字节)，在 72 个时钟周期 (15 μs) 内只提供一个刷新周期。通道检查线的作用是：检查线一旦变成有效电平，8088 不可屏蔽中断 NMI 就处激活态，所以检查线是用来报告外设如扩展存贮器出错情况的。

I/O 通道属于 I/O 映像地址空间，它最多能带 512 个外部设备。I/O 通道上的信号均经电压放大，足够驱动两个 (TTL) 负载。下面列出 I/O 通道各信号 (线) 的详细说明。图 18-9 是 I/O 通道接线图。

信号名称	I/O 方向	作　　用
OSC	O	占空度为 50% 的 14.31818 MHz 扬荡信号
CLK	O	占空度为30%、为主振频率3分频的系统时钟信号
RESET DRV	O	复位驱动信号，用于开机或低电压后系统的初始化或复位。该信号同步于时钟周期的下降沿，高电平有效。

A0～A19	O	地址线、A0 是 LSB 位，A19 是 MSB 位。他们由处理器或 DMA 控制器驱动，用于存贮器或 I/O 的寻址。高电平有效。
D0～D7	I/O	数据线，D0 是 LSB 位，D7 是 MSB 位。连接处理器、存贮器和外设，高电平有效。
ALE	O	地址锁存线，它是 8288 总线控制器的一个输出信号，用来锁存存贮器有效地址。与 AEN 配合时还作为锁存有效 I/O 地址的控制信号。在 ALE 的下降沿地址被锁存。
$\overline{\text{I/O CHCK}}$	I	I/O 通道检查信号，向 CPU 提供接在 I/O 通道上的存贮器或外设的奇偶校验信息。低电平时表示有奇偶校验错。
I/O CHRDY	I	I/O 通道准备好信号。一般慢速外设在检测到有效地址和读/写命令后立即将它拉成低电平。I/O CHRDY 线保持低电平的时间不会超过 10 个时钟周期(2.1 HS)。
IRQ2-IRQ7	I	6 级中断请求线 (2～7)，IRQ 2 的优先级最高，IRQ 7 最低。请求线在处理器响应中断前一直保持高电平。
$\overline{\text{IOR}}$	O	I/O 读写命令信号。此信号由处理器或 DMA 驱动，低电平时有效，用来通知 I/O 设备把数据打入数据总线。
$\overline{\text{ION}}$	O	I/O 写命令信号。由处理器或 DMA 驱动，处有效低电平时它通知 I/O 外设读取数据总线上的数据。
$\overline{\text{MEMR}}$	O	存贮器读命令信号，由处理器或 DMA 驱动。高电平时让存贮器把数据打入数据总线。
$\overline{\text{MEMW}}$	O	存贮器写命令信号，由处理器或 DMA 驱动。高电平时让存贮器读进数据线上的数据。
DRQ1-DRQ 3	I	1～3 DMA 请求通道。外设利用这三个异步通道获得 DMA 服务。DRQ 线有优先级别；DRQ 1 优先线最高，DRQ 2 其次，DRQ 3 最低。DRQ 电平由低变高时，DMA 请求有效。在 DACK 线变有效电平前，DRQ 必须保持高电平。
$\overline{\text{DACK0}}$-$\overline{\text{DACK 3}}$	O	DMA 应答线，回答 DRQ 1-DRQ 3 的 DMA 请求或刷新系统动态存贮器 (DACK0)。他们在低电平时有效。
AEN	O	地址允使线。处有效电平时，DMA 控制器占用地址线、数据线、读写存贮器或 I/O 命令线。

T/C　　o　　终端计数线。当 DMA 通道计数器满值时 T/C 线向外发一个脉冲。

系统板 I/O 通道电压安排：

2 个插脚 + 5 V dc

1 个插脚 – 5 V dc

1 个插脚 + 12 V dc

一个插脚 – 12 V dc

三个插脚 GND 地线

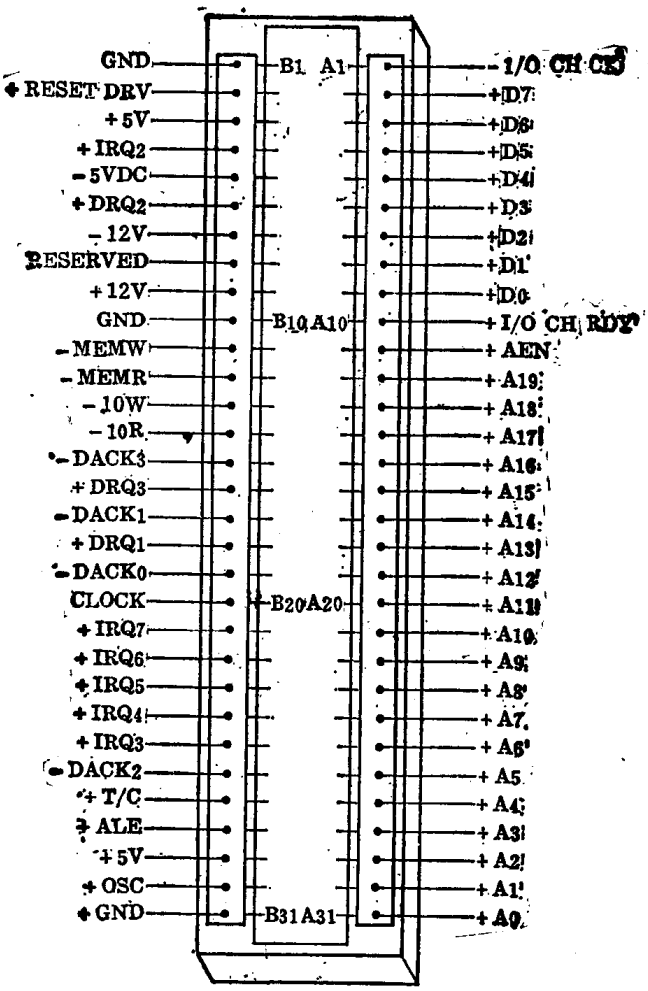


图 18-9　I/O 通道(扩展槽)

说明：Z：三态

* 开机时 NMI 中断被屏蔽，这个屏蔽可由系统软件将 X′80′ 写入 X′A0′ 单元来设置或将 X′00′ 写入 X′A0′ 单元复位。

表 18.5 I/O 地址映射表

地址范围	9	8	7	6	5	4	3	2	1	0	设备
00～0F	0	0	0	0	0	Z	A3	A2	A1	A0	8237-2 DMA 控制器
20～21	0	0	0	0	1	Z	Z	Z	Z	A0	8259 A 中断控制器
40～43	0	0	0	1	0	Z	Z	Z	A1	A0	8253-5 计时器
60～63	0	0	0	1	1	Z	Z	Z	A1	A0	PPI 8255A-5
80～83	0	0	1	0	0	Z	Z	Z	A1	A0	DMA 页寄存器
*AX	0	0	1	0	1						NMI 屏蔽寄存器
CX	0	0	1	1	0						保留
EX	0	0	1	1	1						保留
3F8～3FF	1	1	1	1	1	1	1	A2	A1	A0	TP RS-232-C 插口
3F0～3F7	1	1	1	1	1	1	0	A2	A1	A0	$5^1/_4$ 时软盘机转接器
2F8～2FF	1	0	1	1	1	1	1	A2	A1	A0	保留
378～37F	1	1	0	1	1	1	1	Z	A1	A0	并行打印机转接器
300～3DF	1	1	1	1	0	1	A3	A2	A1	A0	彩色/图形 CRT 转接器
278～27F	1	0	0	1	1	1	1	Z	A1	A0	保留
200～20F	1	0	0	0	0	0	A3	A2	A1	A0	游戏器转接器
3B0～3BF	1	1	1	0	1	1	A3	A2	A1	A0	IBM 单色显示-并行打印机转接器

表 18-6 8255 A 三口子 I/O 位映照关系表

地址 X'0060' 输入

PA 0	+ 键盘扫描码	0	或	IPC $5^1/_4$ 时软盘机 (SW1-1)
1		1		保留 (SW1-2)
2		2		系统板 RAM 容量 (SW1-3)*
3		3		系统板 RAM 容量 (SW1-4)*
4		4		显示器类型 1 (SW1-5)**
5		5		显示器类型 2 (SW1-6)**
6		6		$5^1/_4$ 时驱动器个数 1 (SW1-7)***
7		7		$5^1/_4$ 时驱动器个数 2 (SW1-8)***

地址 X'0061' 输出

PB 0	+ 计时器门 2 (扬声器)
1	+ 扬声器数据
2	+ (读 RAM 区大小)或(读空键)
3	+ 盒带机马达停
4	− 允使 RAM
5	− 允使 I/O 通道检查
6	− 保持键盘时钟为低电平
7	− (允使键盘)或(清键盘和允使读外设开关)

地址 X'0062' 输入

PC 0	扩展 RAM (SW2-1)	—扩展 RAM 容量=2 进制值×32KB] 或 [I/O RAM(SW2-5)
1	扩展 RAM (SW2-2)	
2	扩展 RAM (SW2-3)	
3	扩展 RAM (SW2-4)	
4	+ 录音机数据入(IN)	
5	+ 计时器通道 2 出(OUT)	
6	+ I/O 通道检查	
7	+ RAM PCK	

地址 X'0063' CMD/MODE 寄存器：

7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	1

X'99'

说明

* PA3 PA2 系统板 RAM 容量

SW1-4	SW1-3	
0	0	16 KB
0	1	32 KB
1	0	48 KB
1	1	64 KB

** PA5 PA4 显示器类型

SW1-6	SW1-5	
0	0	保留
0	1	40×25 彩色 CRT(黑白模式)
1	0	80×25 彩色 CRT (黑白模式)
1	1	80×25 IBM 单色 CRT

*** PA7 PA6

SW1-8	SW1-7	5－1/4 时驱动器个数
0	0	1
0	1	2
1	0	3
1	1	4

注意： PA 位＝0 表示开关“开”

位＝1 表示开关“关”

表 18-7 存贮区映照

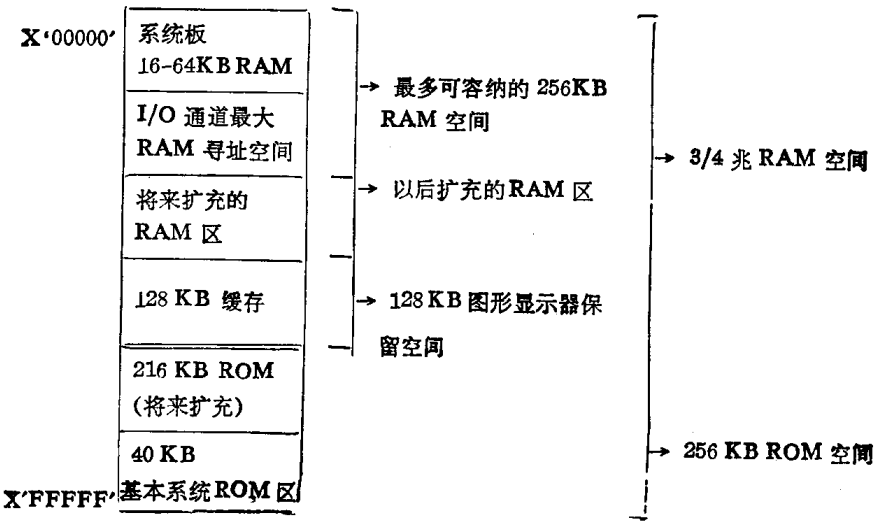


表 18-8 存贮区映照(增量 16KB)

启始地址 DECIMAL	启始地址 HEX	用途
0	00000	系统板 16-64KB RAM
16K	04000	
32K	08000	
48K	0C000	
64K	10000	至多 192KB 的 I/O 通道扩展 RAM
80K	14000	
96K	18000	
112K	1C000	
128K	20000	
144K	24000	
160K	28000	
178K	2C000	
192K	30000	
208K	34000	
224K	38000	
240K	3C000	
256K	40000	I/O 通道以后扩充的 394KB RAM
272K	44000	
288K	48000	
304K	4C000	
320K	50000	
336K	54000	
352K	58000	
368K	5C000	
384K	60000	
400K	64000	
416K	68000	
432K	6C000	
448K	70000	
464K	74000	
480K	78000	
496K	7C000	
512K	80000	
528K	84000	
544K	88000	
560K	8C000	
576K	90000	
592K	94000	
608K	98000	
624K	9C000	

启始地址 DECIMAL	启始地址 HEX	用途
640K	A0000	保留
656K	A4000	
672K	A8000	
688K	AC000	
704K	B0000	单色 CRT
720K	B4000	112KB 图形/显视缓存器
736K	B8000	彩色监视器
752K	BC000	
768K	C0000	192KB 扩充区
784K	C4000	
800K	C8000	
816K	CC000	
832K	D0000	
848K	D4000	
864K	D8000	
880K	DC000	
296K	E0000	
912K	E4000	
928K	E8000	
944K	EC000	
960K	F0000	保留
976K	F4000	48KB 基本系统 ROM 区
992K	F8000	
1.008M	FC000	

表 18-9　系统板上存贮器开关设置

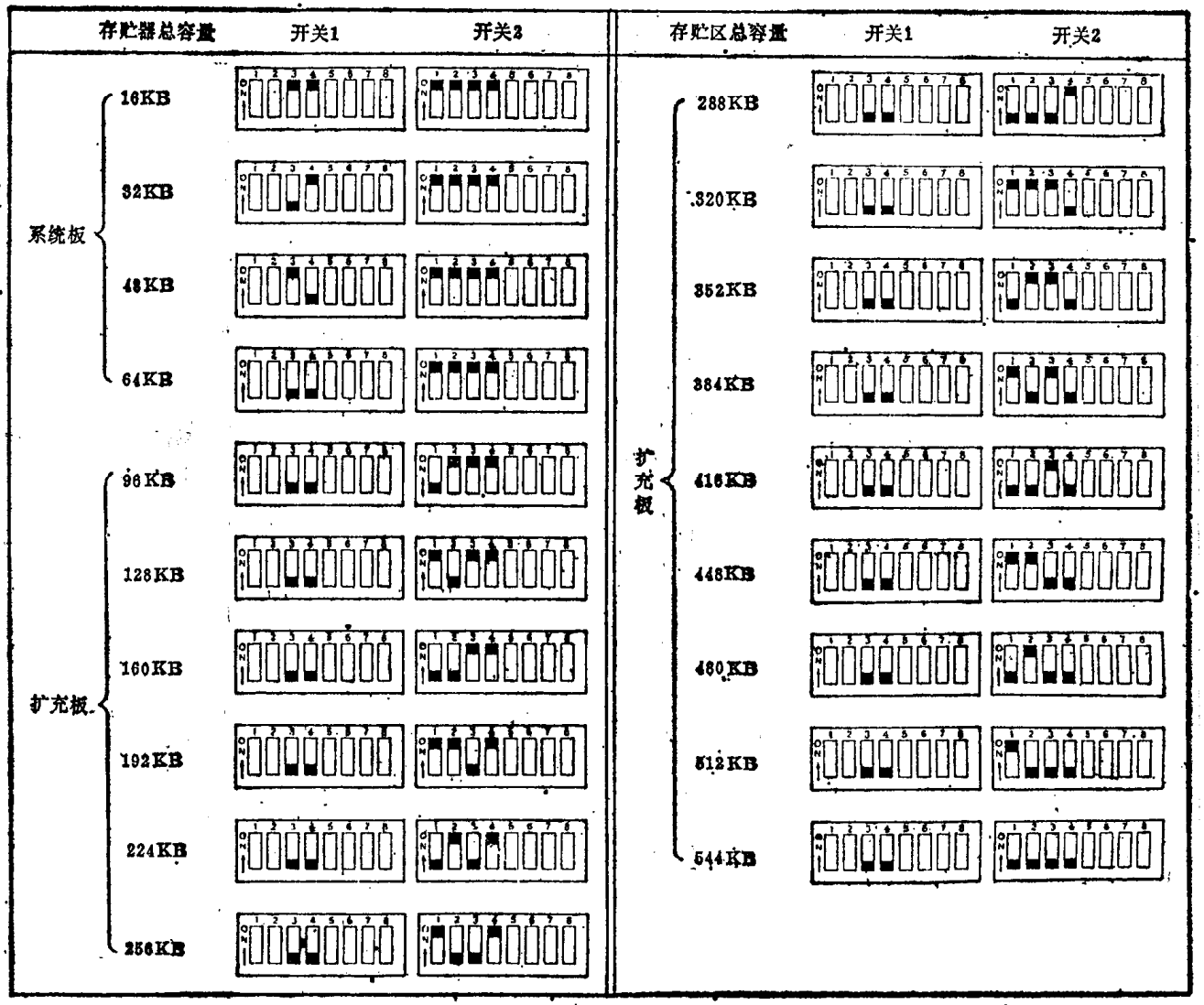
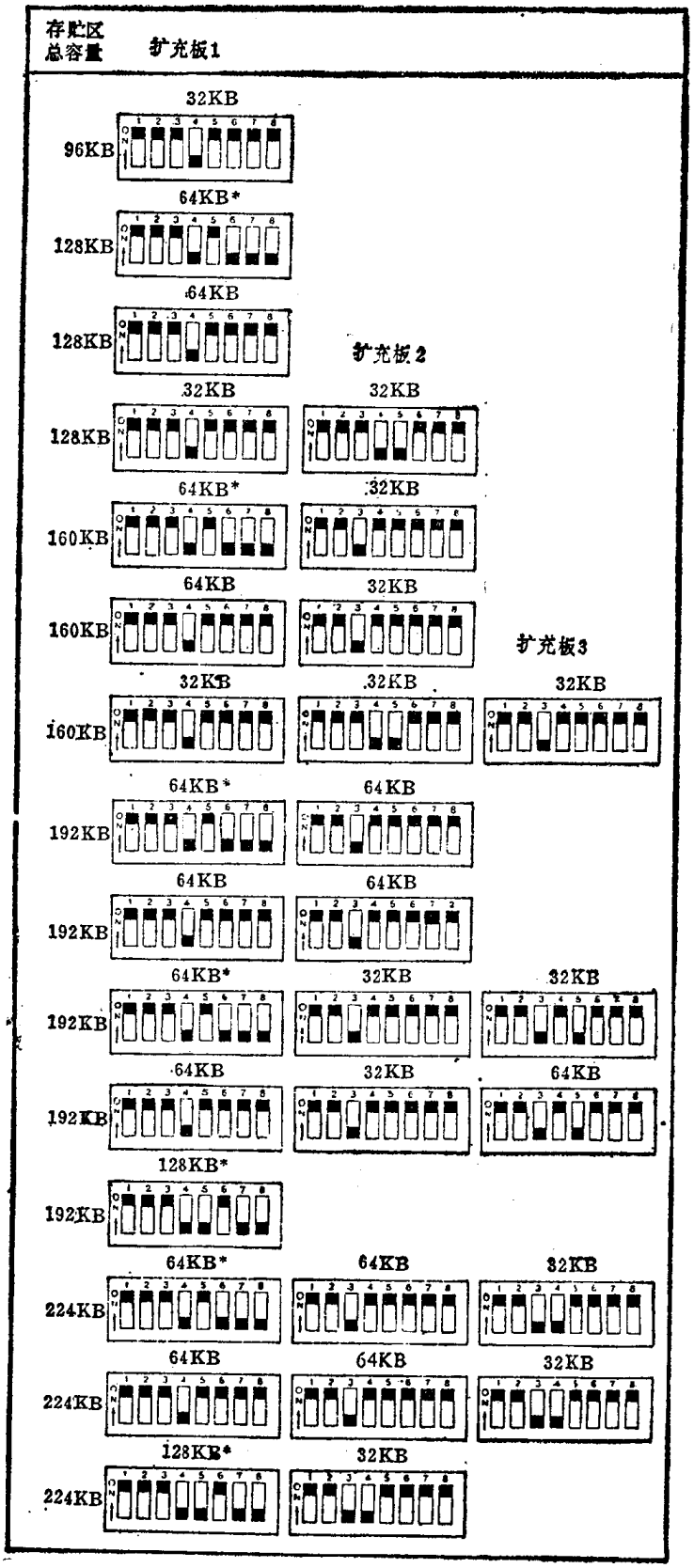
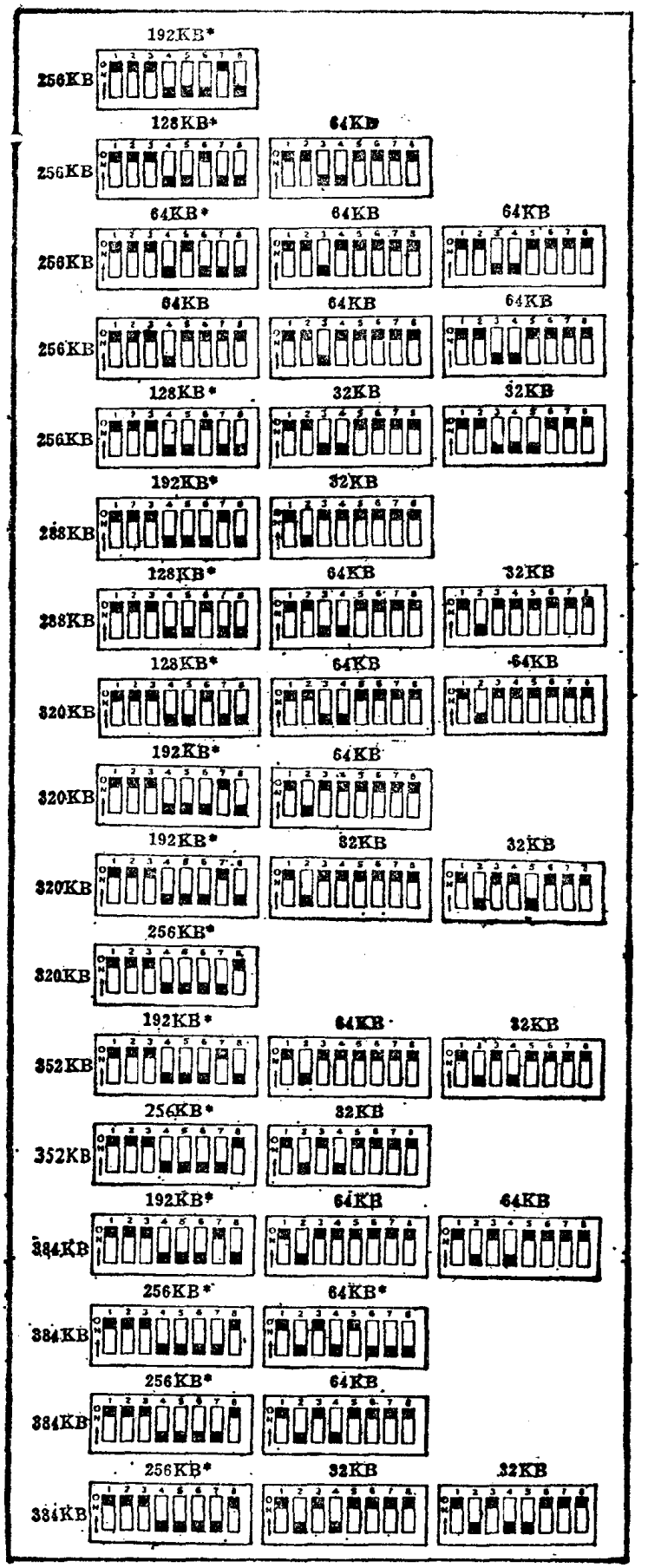


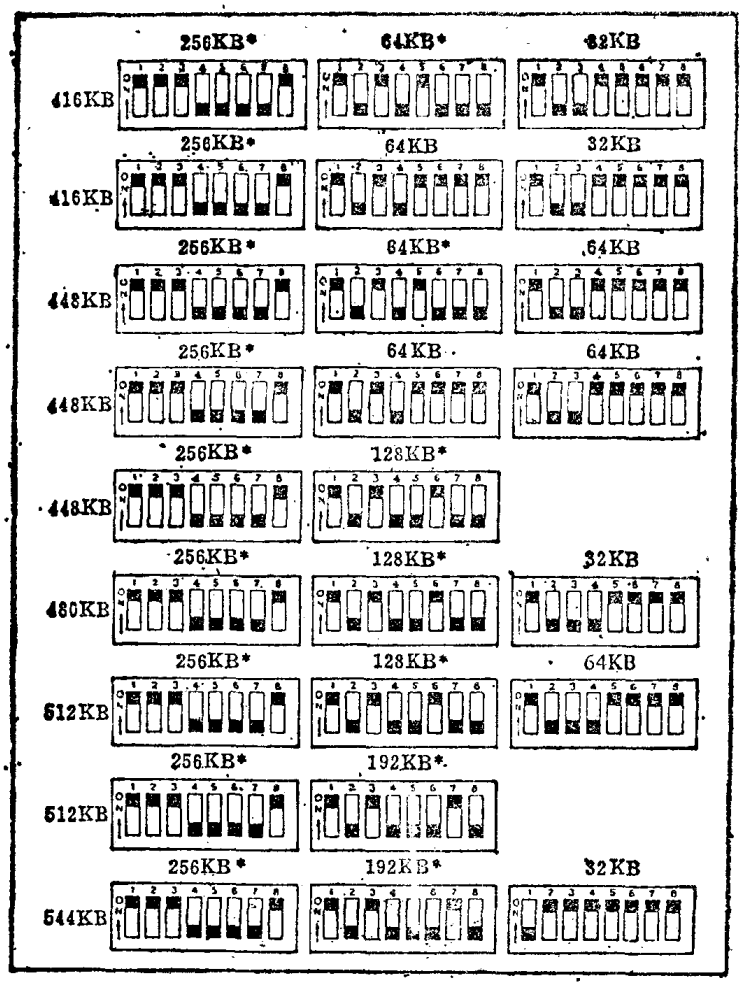
表 18-10　存贮器扩充板上的开关设置(*表示 64/256 KB 扩充板的实际容量)



（续表）


256KB 192KB*
256KB 128KB* 64KB
256KB 64KB* 64KB 64KB
256KB 64KB 64KB 64KB
256KB 128KB* 32KB 32KB
288KB 192KB* 32KB
288KB 128KB* 64KB 32KB
320KB 128KB* 64KB 64KB
320KB 192KB* 64KB
320KB 192KB* 32KB 32KB
320KB 256KB*
352KB 192KB* 64KB 32KB
352KB 256KB* 32KB
384KB 192KB* 64KB 64KB
384KB 256KB* 64KB*
384KB 256KB* 64KB
384KB 256KB* 32KB 32KB

（续表）



§18.4 电　　源

系统单元和键盘所需直流电压，由 63.5W 开关式调节器供给。调节器输出四种受过电压、过电流装置保护的直流电压：±5V±5%（7A）；+12V±5%（2A）；-5V±10%（300mA）和-12V±10%（250mA）。发生 DC 过载、过压现象时，调节器能自动切断，直至情况正常为止。四种直流电压的分配情况是：+5 Vdc：系统板吸收 3A 电流，其余 4A 给 I/O 扩展槽；12 Vdc：驱动两台内藏式 5-1/4 吋软盘机和系统动态存贮器（软盘机马达不可同时启动）；-5 Vdc：加至存贮器偏差电路和录音机转接器锁相环模拟电路；-12 Vdc：连同+12 Vdc 作为异步通讯转接器串行接口插件 EIA 驱动器和接收器的驱动电压。IBM 单色 CRT 自带电源变压器，但它的交流电压从系统单元的一个非标准插座引出，由系统板电源开关控制。下面是直流电源插座、插头的引脚示意图。

调节器中有一直流电压监视信号线，其电平范围与 TTL 器件兼容。当 4 组直流电压均高于表 18-13 列出的比较电压时，直流电压监视信号线（电压正常信号线）处高电平态（2.5V～5.5V），能放出 60HA 的电流；当 4 组电源中任一电压低于表 18-13 列出的比较电

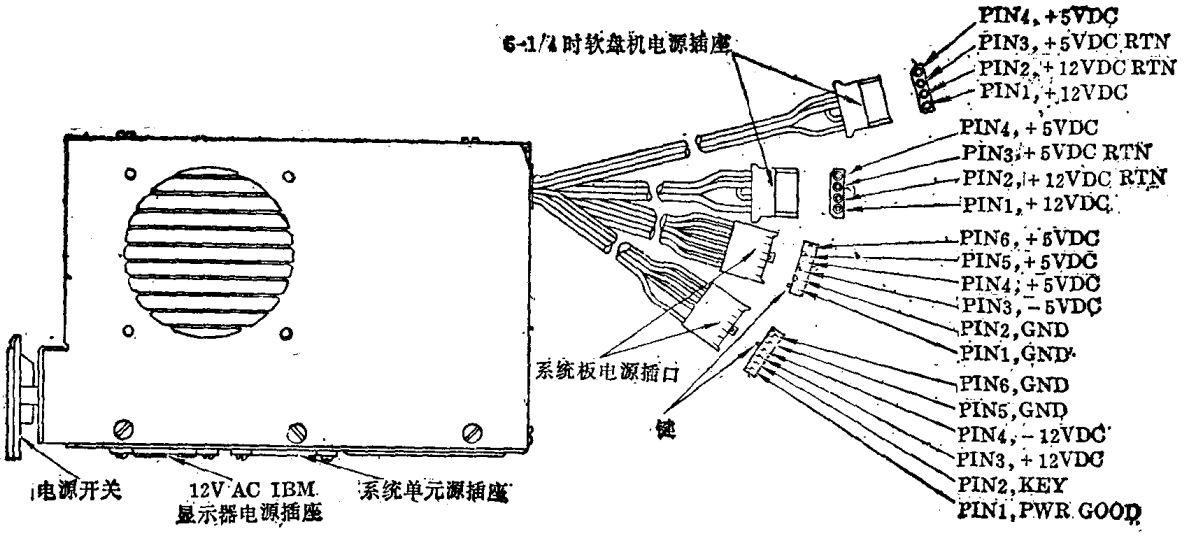


图 18-10　电源插座、插头示意图

压时，直流电压监视信号线下降至低电平（0～0.4V），能吸收 500 μA 电流。该信号线在全部直流电压恢复正常后上升至高电平，上升过程的延续时间为 100 ns～500 ns。如果交流输入电压被切断的时间不超过 5 秒，电源正常信号也被置成高电平。

表 18-11　交流输出表

电压(V)额定值	电　流　(A)		极限电压(V)	
	最小值	最大值	最小值	最大值
120.0	0.0	0.75	101.0	130.0

表 18-12　过电压/电流保护(初级端)

额定电压 (V)	保 护 方 式	标称电流 (A)
120	2 SOC SD4 60 Hz 熔丝	2

表 18-13　±5V、±12V 比较电压表

输　出	最 小 值	额定比较电压	最 大 值
+5V	+3.7	+4.0	+4.3
−5V	−3.7	−4.0	−4.3
+12V	+8.5	+9.6	+10.5
−12V	−8.5	−9.6	−10.5

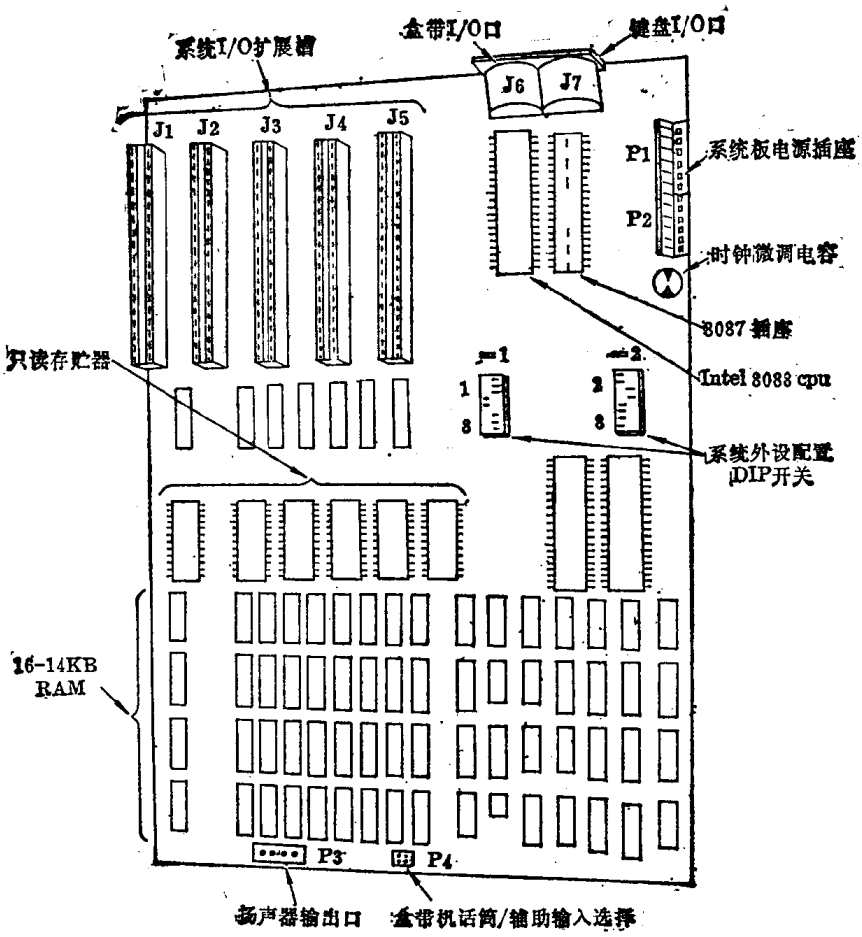

系统I/O扩展槽
盒带I/O口
键盘I/O口
J6
J7
J1
J2
J3
J4
J5
P1
系统板电源插座
P2
时钟微调电容
8087 插座
只读存贮器
Intel 8088 cpu
系统外设配置
DIP开关
16-14KB
RAM
P3
P4
扬声器输出口
盒带机话筒/辅助输入选择

图 18-11　系统板

第十九章　IBM单色显示器—并行打印机转接器

§19.1　概　　述

IBM显示器—打印机转接器，是系统单元连接IBM单色CRT或IBM 80 CPS点阵打印机的桥梁。转接器的核心是Motorala 6845 CRT控制器。转接器内设有4KB无奇偶校验的显示缓存器，起始地址B0000H，显示缓存有2个存取口，CPU可直接访问。单色显示器—打印机转接器的框图见图19-1。下面是转接器的技术参数及特点。

1．显示帧尺寸80×25的图象

2．直接驱动输出

3．不占据系统内存

4．9×14字符方框

5．7×9字符点阵

6．字符具有属性字节

7．提供256个显示字符码，他们存在8KB字符发生器中

8．如果转接器带P39磷光CRT，则光笔就不能联接

有关并行转接器部分见第21章"并行打印机转接器"。

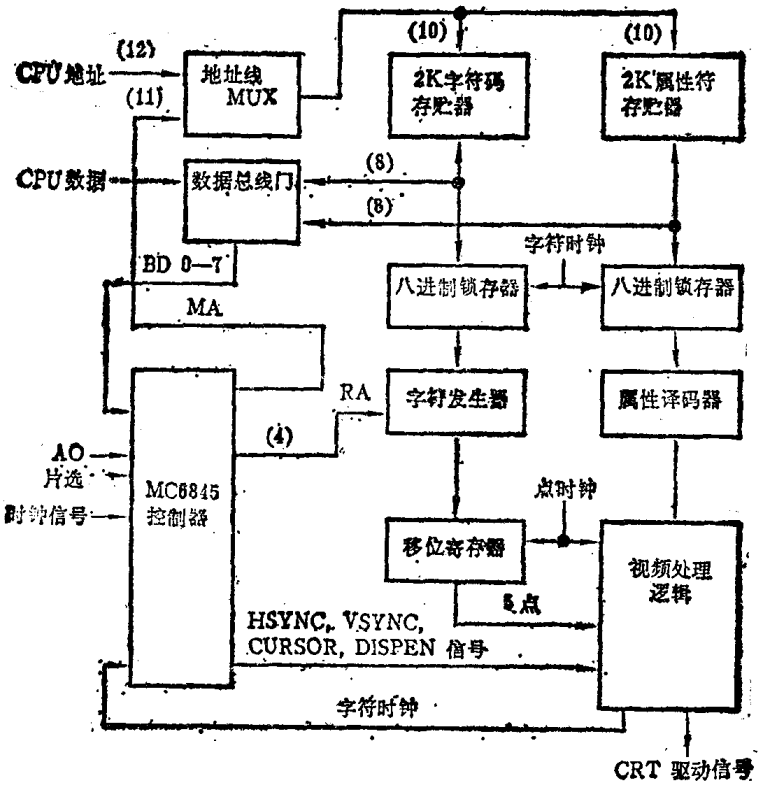


图19-1　IBM单色显示器转接器框图

§19.2 转接器与系统通道的接口及工作方式

转接器使用了I/O通道的地址线、数据线、存贮器及I/O读写线、复位线、I/O准备好线、I/O时钟线和IRQ7线。这些I/O扩展槽上的信号线除几根地址线能提供2个LS的电流外，一般只能带一个LS负载。转接器正常工作前应予以初始化，即置位插座上的高分辨率显示器输出口 (OUT PORT 3B8≈01H) 后再使用CRT。不进行初始化CPU将无法访问转接器并会损坏显示器。在任何两次对转接器显示缓存或控制寄存器的间隔，必须插入一个等待周期。由于CPU/监视器的访问要同显示转接器的字符时钟同步，这个等待周期的持续时间不固定。显示器正常工作时，从转接器缓存以每553 ns 2个字节的速度取得一个显示字符的全部信息，相当于1.8兆字节/秒。向显示缓存读、写数据可通过DMA通道进行。

转接器缓存单元是这样安排的：偶地址单元存放字母、数字或字区图(block graphics)的字符代码；偶地址+1的奇地址单元存放字符的属性代码，转接器根据字符属性码处理相应的字符代码。字符属性码中的闪烁，明暗位还能与前景、背景位混合使用以增加字符的显示效果。字符代码为一般的ASC II码。

显示字符在显示缓存中的两个字节：(R-红色、G-绿色、B-蓝色)

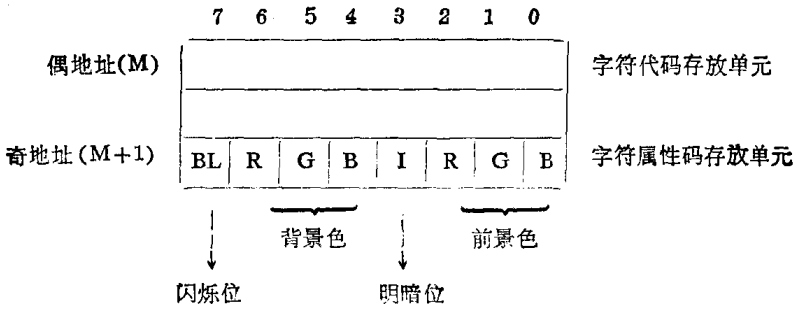


表19-1 IBM单色CRT显示字符属性位前、背景配合

前景			背景			显示效果
R	G	B	R	G	B	
0	0	0	0	0	0	不显示
0	0	1	0	0	0	下划线
1	1	1	0	0	0	白字符黑背景
0	0	0	1	1	1	视频翻转，黑字符白背景

§19.3 转接器6845 CRT控制器编程要点

6845控制器的具体初始化、预置过程在第二十章详细介绍，这里仅列出几张有用的参数表

表 19-2 6845 初始参数表(适用于 80×25 单色 CRT)

6845 寄存器号	寄存器名	单位	参数值	6845 寄存器号	寄存器名	单位	参数值
R0	水平总数	字符	61H	R9	扫描线最大地址	扫描线	0DH
R1	水平显示	字符	50H	R10	光标启始行	扫描线	0BH
R2	水平同步位置	字符	52H	R11	光标结束行	扫描线	0CH
R3	水平同步亮度	字符	0FH	R12	起始地址(H)	···	00H
R4	垂直总数	字符行	19H	R13	起始地址(L)	···	00H
R5	垂直调整	扫描线	06H	R14	光标(H)	···	00H
R6	垂直显示	字符行	19H	R15	光标(L)	···	00H
R7	垂直同步位置	字符行	19H	R16	保留	···	—
R8	交错模式	···	02H	R17	保留	···	—

表 19-3 单色显示器—并行打印机转接器地址安排(请参阅 I/O 地址映射表)

地 址	寻址对象	地 址	寻址对象
3B0	未用	3B8	CRT 控制器口 1*
3B1	未用	3B9	保留
3B2	未用	3BA	CRT 状态口**
3B3	未用	3BB	保留
3B4	6845 寻址寄存器	3BC	并行打印机数据口
3B5	6845 数据寄存器	3BD	打印机状态口
3B6	未用	3BE	打印机控制口
3B7	未用	3BF	未用

说明:

* CRT 控制口(也称 CRT 输出口) 1 的位安排:

位	0	1	2	3	4	5	6	7
作用	+高分辨模式位	无用	无用	+允使视频输出	无用	+允使闪烁	无用	无用

** CRT 状态口(I/O 地址'3BA'):

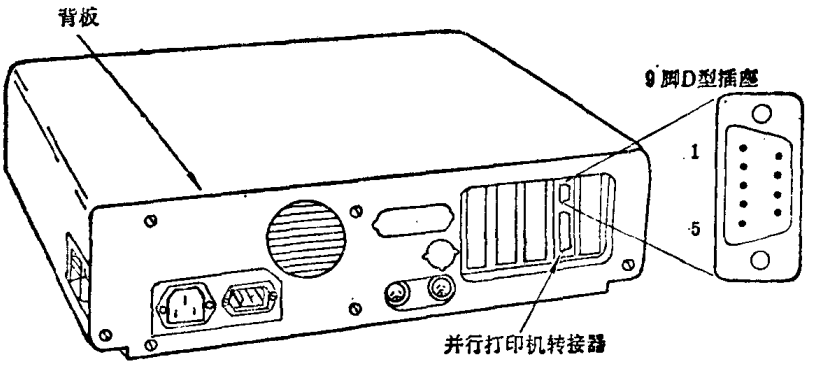
位	0	1	2	3
作用	+水平驱动	保留	保留	+B/W 视频

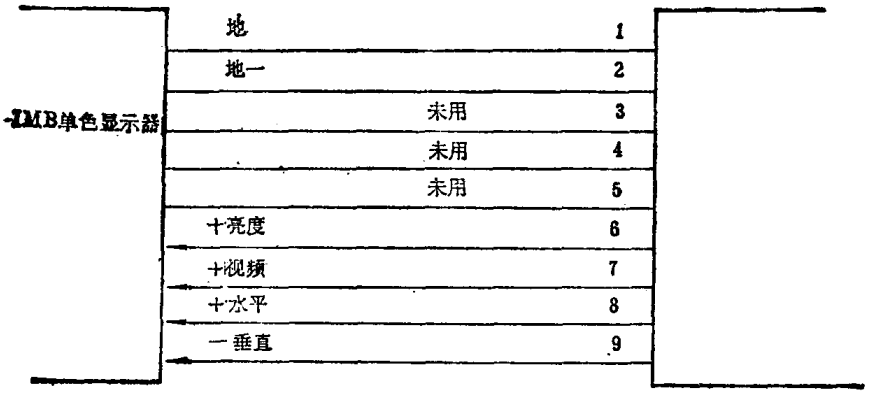
因程序不使用显示器的第 25 根线(状态线),所以它被用来产生提示符方框,方便操作。如 BASIC 用它在 CRT 上形成 10 个功能键和键号的显示字符。

§19.4 IBM 单色显示器

IBM 单色显示器是一种 12 吋高分辨率直角偏转 CRT,该显示器通过两根电缆与系统单元相连,一根接 CRT 转接器,另一根接交流电源。

技术参数：帧尺寸 80×25，9×15 字符点阵，最大视频宽度 16.27 MH2，分辨率 720（水平）×350 线（垂直），刷新频率 50Hz。行频（TTL 兼容高电平）18.432 Khz。





注意：1. 所有信号均处 TTL 电平
2. 高电平－5V，低电平－0.6V

图 19-2 IBM 单色直接驱动接口及引脚安排

第二十章　彩色/图形监视器转接器

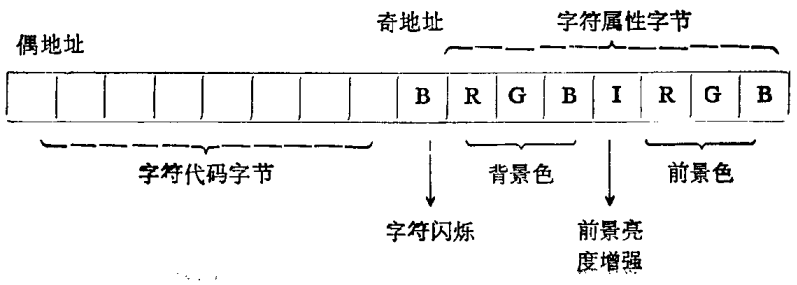
§20.1 概　述

彩色/图形转接器主要用来连接不同类型的彩色/黑白电视频率监视器或电视机，连接电视机时，要在转接器和电视机间加调制解调器。转接器内的6845控制器，负责显示器的全部显示、转换工作，用户只需写入欲显示符的代码及颜色信息。转接器自带起始地址X'B800'的16 KB字符显示缓存，能提供合成频率口、直接驱动口以及用户RF调制器连接口三个视频接口，另外还有一个光笔接口。转接器占用中断线2，根据转接器的工作方式可将视频宽度定在7 MHZ或14 MHZ。转接器封装在一个插头里，插头背面是直接驱动器口和视频合成口，使用时将转接器插头插在I/O扩展槽上。

§20.2　字母/数字显示方式(A/N式)

A/N式是以字符为单位进行显示的方式。该方式低分辨率监视器或电视机的显示帧尺寸为40×25，高分辨率监视器的显示帧尺寸为80×25。在这两种帧尺寸下，字符点阵框尺寸为8×8，小写下行字母如f、g为5×7。黑白模式时的字符属性提供视频翻转、闪烁、亮度加强等信息；彩色模式的字符属性提供16种前景色、8种背景色及闪烁信息，用户还可以选择16种屏幕边界色。

A/N式显示字符共有256种，分为七类。第一类是游戏字符类共7个字符；第二类是字处理编辑字符类共15个字符；第三类是ASC II字符类共96个字符；第四类是外国文字字符类共48个字符；第五类是图表用字符类共48个字符；第六类是常用希腊字母类共16个字符；第七类是常用科学记号字符类共15个字符。由以上256种字符组成的屏幕显示字符信息存放在转接器16 KB显示缓存内。对于40×25帧尺寸屏幕，一帧图象的所有1000个字符代码和1000个字符属性码将占用2000个字节的显示缓存，整个16 KB空间可记下8帧屏幕。对于80×25帧尺寸屏幕、整个16KB空间可记录4帧屏幕。下面是显示字符在缓存中的存放格式及缓存单元与屏幕显示位置的关系示意。



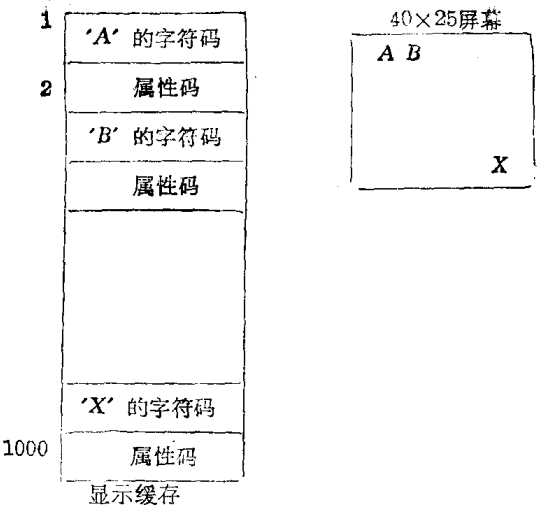


图 20-1　显示缓存示意图

表 20-1　单色-彩色字符属性比较表

属性字节 7 6 5 4 3 2 1 0	单色显示器转接器		彩色监视器转接器		显示效果
B R G B I R G B FG 背景 前景	字符色	背景色	字符色	背景色	
B 0 0 0 I 1 1 1	白	黑	白	黑	一般显示
B 1 1 1 I 0 0 0	黑	白	黑	白	视频翻转
B 0 0 0 I 0 0 0	黑	黑	黑	黑	不显示(全黑)
B 1 1 1 I 1 1 1	白	白	白	白	不显示(全白)
其余属性码定义前景色、背景色的混合色	R.G.B 其余组合得到黑背景白字符		R.G.B 其余组合的结果见表—5		

表 20-2　红、绿、兰三色混合表

原色			混合色	原色			混合色
R	G	B		R	G	B	
0	0	0	黑	1	0	0	红
0	0	1	兰	1	0	1	品红
0	1	0	绿	1	1	0	黄
0	1	1	深兰	1	1	1	白

说明：

1．8 种颜色的深淡由 I 位决定

2．彩色监视器转接器将把为 IBM 单色显示器写的带下划线属性的字符显示成兰色。

3．单色显示器将把为彩色监视器写的兰色字符代码显示成黑背景、带白色下划线的白字符。

4．并非所有的显示器都能识别出 I 位。

§20.3 全象点寻址方式(APA 式)

APA 式是按照图形的显示要求、效果，显示用户以图象象点为基本显示单位定义的帧图象的方式。用户可以定义象点的位置、颜色和明暗。APA 具有以下三种制式：

一、低分辨率制式

彩色图象帧尺寸 160×100，象点尺寸 2×2，每个象点可取 16 种颜色。低分辨率制式用到转接器中 8KB 的显示缓存。由于 BIOS 不支持这种制式，所以真正使用时，除将 6845 模式寄存器置成 40×25 制式外，还要编制特别的程序。

二、中分辨率制式

彩色图象帧尺寸 320×200，象点尺寸 1×1，每个象点可取背景色或三色色板中的一色。色板由软件指定成为红/绿/棕色板，或为深兰/品红/白色板。背景色有 16 种。中分辨率制式需要 16 KB 转接器显示缓存，缓存的每个字节记录 4 个象点的信息。下面是显示缓存区地址映照表和缓存区内每个字节的格式，#0000 号字节对应屏幕左上角象点，奇象点行点的信息与偶象点行点的信息是分开存放的。

表 20-3 地址映照表

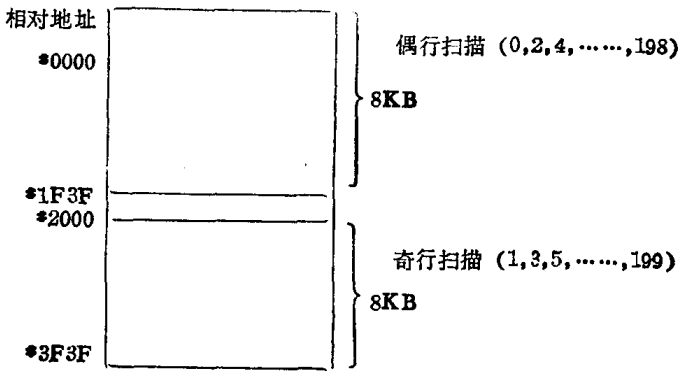
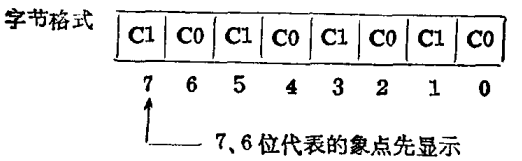


表 20-4 C1、C2 译码表



C1	C0	
0	0	背景色
0	1	预定色板第一色
1	0	预定色板第二色
1	1	预定色板第三色

表 20-5　色板颜色选择表

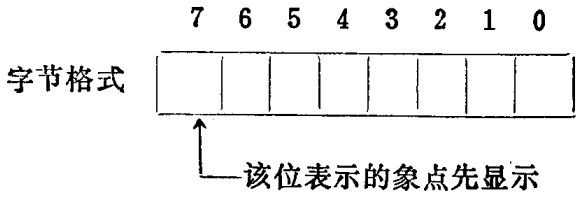
色　　号	预　定　色　板	
	1	2
1	深兰	绿
2	品红	红
3	白	棕

说明：

背景色的设置方法见 6845 编程一节中的“色彩选择寄存器”。

三、黑白高分辨率制式(监视器)

黑白图象帧尺寸 640×200，象点尺寸 1×1。高分辨率制式要求占用转接器 16 KB 显示缓存区，缓存区的映照关系同中分辨率制式，但字节格式不同，字节中的每个位将与一个象点对应。



对转接器显示缓存的一点说明：

字符代码或象点字节的缓存区起始地址，必须由 CPU 通过 6845 控制器置为偶数。除高分辨率 A/N 式外，CPU 和显示控制器 6845 能同时访问该显示缓存。对高分辨率 A/N 式，CPU 只能在显示器水平回扫期间访问显示缓存，否则图象将受随机干扰。

表 20-6　R、G、B、I 位与 16 种色彩的关系

I	R	G	B	颜　色	I	R	G	B	颜　色
0	0	0	0	黑	1	0	0	0	深灰
0	0	0	1	兰	1	0	0	1	淡兰
0	0	1	0	绿	1	0	1	0	淡绿
0	0	1	1	深兰	1	0	1	1	浅兰
0	1	0	0	红	1	1	0	0	淡红
0	1	0	1	品红	1	1	0	1	淡品红
0	1	1	0	褐	1	1	1	0	黄
0	1	1	1	淡灰	1	1	1	1	白

§20.4　转接器 6845 CRT 控制器编程要点

6845 控制器共有光栅寄存器、色彩选择寄存器、模式选择寄存器、状态寄存器四组寄存器，他们决定了 CRT 的光栅尺寸、制式、色彩，并向 CPU 返回 6845 的状态。编程 6845 r

确步骤是：1．决定显示方式；2．复位视频允使位；3．编程 6845 光栅寄存器；4．编程 6845 模式及色彩选择寄存器。具体 6845 程序可参见 BIOS 的有关程序段。

一、光栅寄存器(I/O 地址 3D4、3D5)

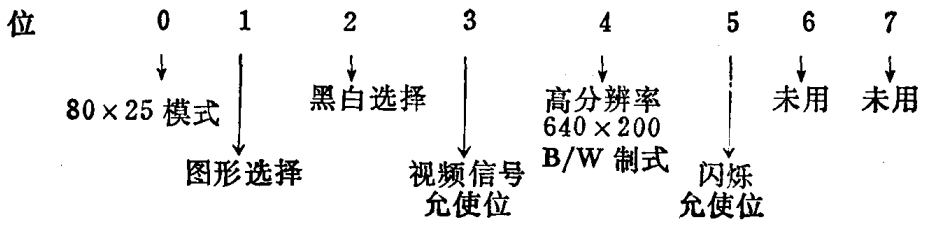
光栅寄存器实际上是 19 个寄存器组成的寄存器组，他们均通过同一个数据口与数据总线相连。光栅寄存器用来存放屏幕的光栅信息，必要时还能返回光笔在屏幕上的坐标。光栅寄存器中的地址寄存器 (I/O 地址 3D4)，是其余 18 个寄存器的指针，它的设置方法是，用 OUT 指令将其余 18 个寄存器之一的 2 进制号码值(放在 I/O 数据线的低 5 位)写入地址寄存器，使其指向某个寄存器，然后再用 OUT 指令将需要的字节写入该寄存器。写其他光栅寄存器时，必须修正地址寄存器的内容，让其指向正确的光栅寄存器，下面是 40×80 A/N 式、80×25 A/N 式、图形模色三种彩色显示制式下 18 个光栅寄存器应具有的 16 制数值表。

表 20-7　6845 初始参数表(适用于彩色/图形 CRT)

地址值	寄存器号	寄存器名	单　位	方　向	40×23 A/N	80×25 A/N	图　形
0	R0	水平总数	字符	只写	38	71	38
1	R1	水平显示	字符	只写	28	50	28
2	R2	水平同步位置	字符	只写	2D	5A	2D
3	R3	水平同步亮度	字符	只写	OA	OA	OA
4	R4	垂直总数	字符行	只写	1F	1F	7F
5	R5	垂直调整	扫描行	只写	06	06	06
6	R6	垂直显示	字符行	只写	19	19	64
7	R7	垂直同步位置	字符行	只写	1C	1C	70
8	R8	交错方式	—	只写	02	02	02
9	R9	扫描线最大地址	扫描行	只写	07	07	01
A	R10	光标起始行	扫描行	只写	06	06	06
B	R11	光标结束行	扫描行	只写	07	07	07
C	R12	起始地址(H)	—	只写	00	00	00
D	R13	起始地址(L)	—	只写	××	××	××
E	R14	光标地址(H)	—	读/写	××	××	××
F	R15	光标地址(L)	—	读/写	××	××	××
10	R16	光笔(H)	—	只读	××	××	××
11	R17	光笔(L)	—	只读	××	××	××

二、模式选择寄存器(I/O 地址 3D8)

模式选择寄存器是一个 6 位不可读、只输出寄存器，它的内容由 8088 OUT 指令设置。下面是模式寄存器各个位的定义。



说明：

位 0　为 1 时选 80×25 模式，0 时选 40×25 模式

位 1　为 1 时选 320×200 模式，0 时选 A/N 模式

位 2　为 1 时选黑白模式，0 时选彩色模式

位 3　为 1 时允许在改变模式的某个时刻输出视频信号

位 4　为 1 时选择 640×200 黑白图形模式

位 5　1 时闪烁背景色(对 A/N 式)，若没有选择高属性位，背景色可是 16 种背景色中的任一种。此位应置 1，使字符的背景能闪烁。

模式寄存器的总结(见表 20-8)

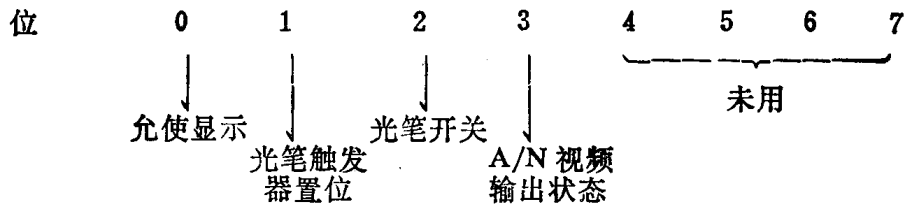
表 20-8　(Z：三态)模式寄存器

Bits

0	1	2	3	4	5	模　式　选　择
0	0	1	1	0	1	40×25　A/N　B/W 模式
0	0	0	1	0	1	40×25　A/N　彩色模式
1	0	1	1	0	1	80×25　A/N　B/W 模式
1	0	0	1	0	1	80×25　A/N　彩色模式
0	1	1	1	0	Z	320×200　B/W　图形模式
0	1	0	1	0	Z	320×200　彩色图形模式
0	1	1	1	1	Z	640×200　B/W　图形模式

三、状态寄存器(I/O 地址 X′3DA′)

状态寄存器是一用 8088 IN 指令读取其内容的 4 位只读寄存器，他向 CPU 提供视频输出状态等信息。下面是状态寄存器 4 个位的定义。



说明：

位 0　为 1 时允许 CPU 访问显示缓存，且不干扰 CRT 的工作

位 1　为 1 时表示光笔触发器被光笔输入正上升沿置位了。当电源接通或利用地址触发向 X′3DB′ 发 OUT 任何数据指令时，光笔触发器复位

位 2　该位反映光笔的开关状态。为 0 表示开关“开”。光笔开关没有锁存及反弹触发器

位 3　A/N 视频信号读取位。视频信号的发生方式(是否 RAS 式) 可通过读位 3 加以判断。

四、色彩选择寄存器(I/O 地址 X′3D9′)

色彩选择寄存器是 6 位不可读只输出寄存器，其内容用 8088 OUT 指令设置。下面是色彩选择寄存器各位的定义。

位号 0 A/N 式时选择兰边界色或 APA 式时(中分辨率模式)选择兰背景色

1 A/N 式时选择绿边界色或 APA 式时(中分辨率模式)选择绿背景色

2 A/N 式时选择红边界色或 APA 式时(中分辨率模式)选择红背景色

3 A/N 式时加深边界或 APA 式(320×200)加深背景色

4 A/N 式时交替显示背景色

5 320×200 模式色板选择

说明:

第 0.1.2.3 位在 40×25 A/N 式下选择屏幕的边界色，330×200 图形模式下选择背景色，共 16 种

第 4 位 A/N 式时交替选择加深了的背景色色板

第 5 位 选择 320×200 图形制式的背景色色板，见下表:

位 5	C1	C0	
1	0	0	位 0-3 选定的背景色
1	0	1	深兰
1	1	0	品红
1	1	1	白
0	0	0	位 0-3 选定的背景色
0	0	1	绿
0	1	0	红
0	1	1	黄

表 20-9 彩色监视器转接器各寄存器的 I/O 地址总表

I/O 地址	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	寄存器
3D0	1	1	1	1	0	1	0	Z	Z	0	6845 寄存器
3D1	1	1	1	1	0	1	0	Z	Z	1	6845 寄存器
3D4	1	1	1	1	0	1	0	1	0	0	光栅寄存器地址寄存器
3D5	1	1	1	1	0	1	0	1	0	1	光栅寄存器数据寄存器
3D8	1	1	1	1	0	1	1	0	0	0	模式寄存器
3D9	1	1	1	1	0	1	1	0	0	1	色彩寄存器
3DA	1	1	1	1	0	1	1	0	1	0	状态寄存器
3DB	1	1	1	1	0	1	1	0	1	1	光笔清除闩锁器
3DC	1	1	1	1	0	1	1	1	0	0	光笔预置闩锁器

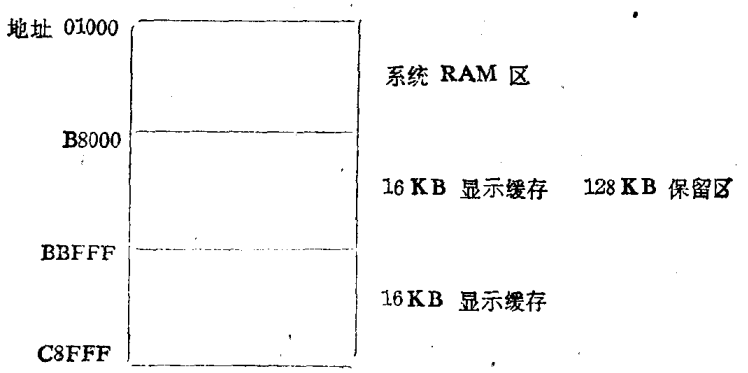


图 20-3 彩色显示器转接器显示缓存 I/O 地址映射图

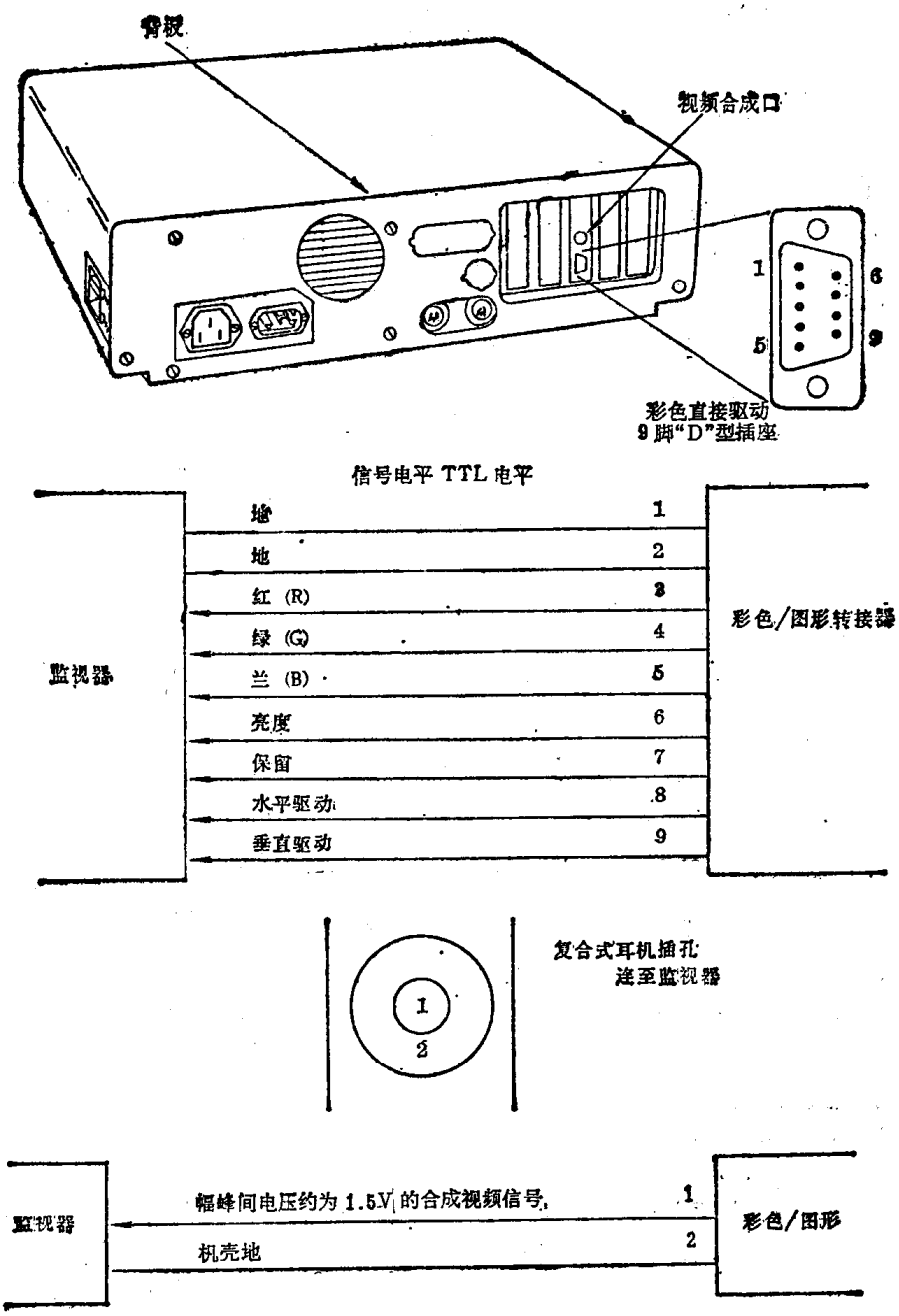


图 20-4　彩色/图形监视器直接驱动及合成接口引脚安排

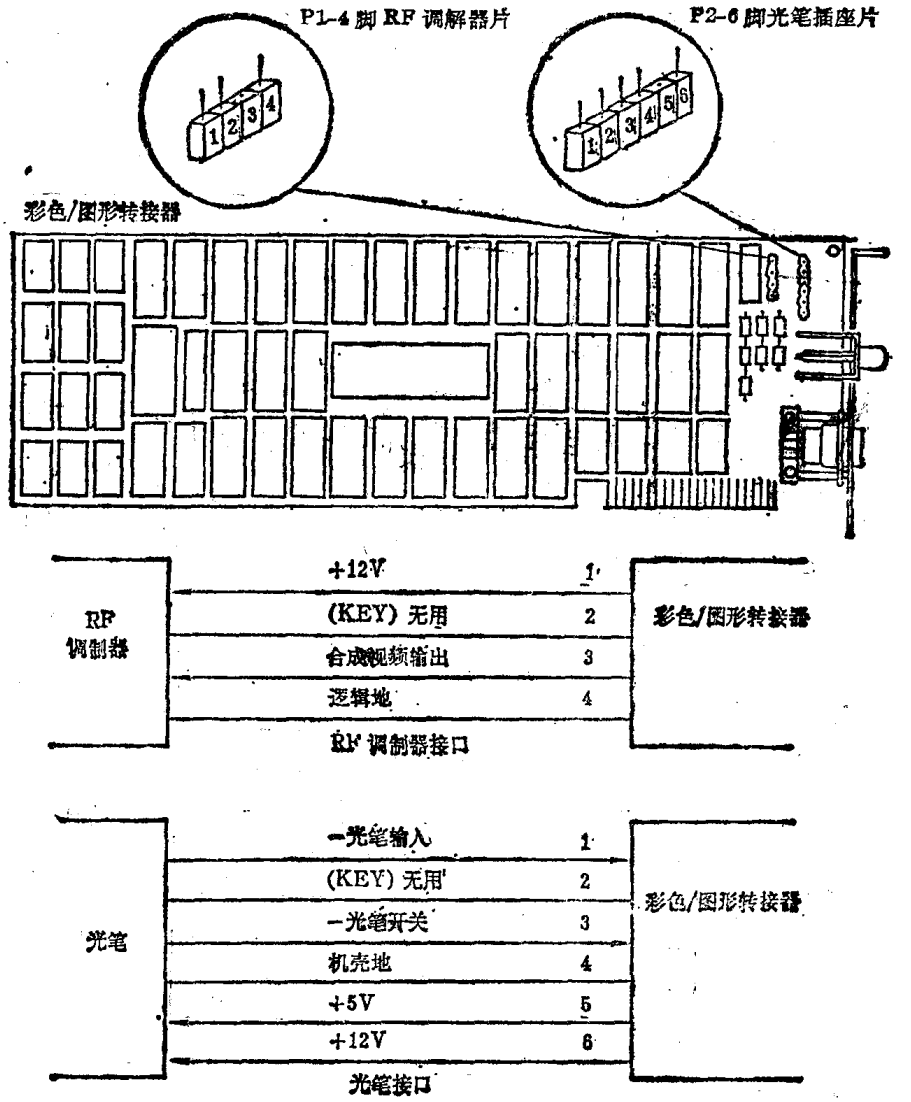


图 20-5　彩色/图形监视器转接器辅助视频插头

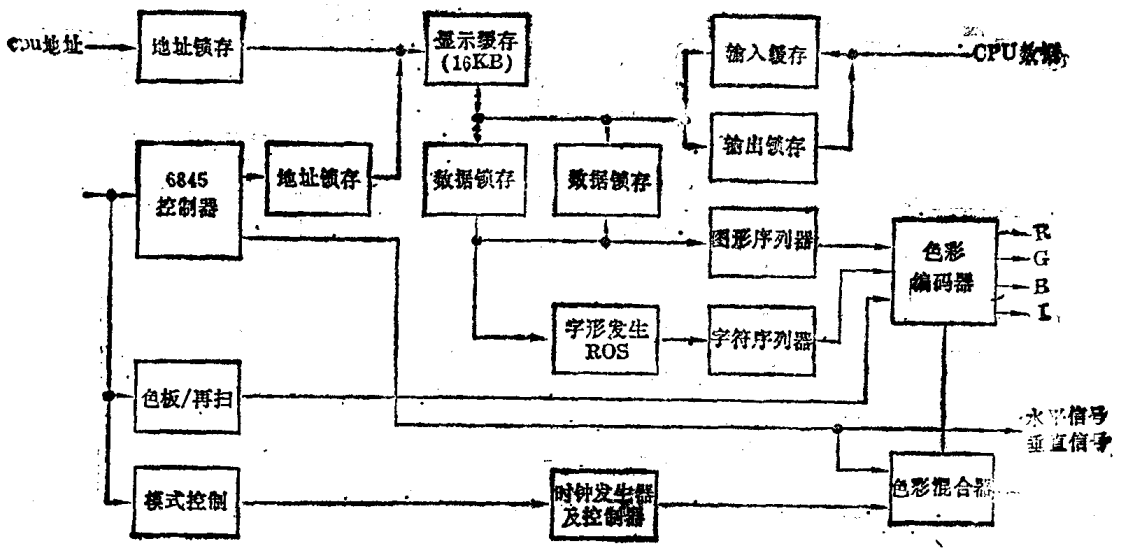


图 20-6　彩色/图形监视器转接器框图

第二十一章 并行打印机转接器

§21.1 概 述

并行打印机转接器是为连接并行打印机设计的，但也能用它连接与此转接器输入/输出电平配合的其他设备。转接器带 12 个 TTL 缓冲输出口和 5 个稳态输入口，输入口、输出口用 8088 IN 指令读取，输出口用 8088 OUT 指令写。输出口主要向 CPU 提供打印机状态信息，其中的一个输入口还能向 CPU 发中断信号，该中断能被程序"关中"或"开中"。输出口主要用来记录 CPU 发来的数据或命令，选通线有效时，数据从输出口传送给打印机。输出口的内容也能用 8088 IN 指令读取，作为判断是转接器错误还是打印机错误的依据。

转接器封装在一个插在 I/O 扩展槽上的插头里，转接器的输入、输出信号由其背部的 25 脚"D"型插座引出。并行打印机转接器的功能相当于 IBM 单色 CRT—打印机转接器打印机部分的功能。下面是转接器的框图。(图 21-1)

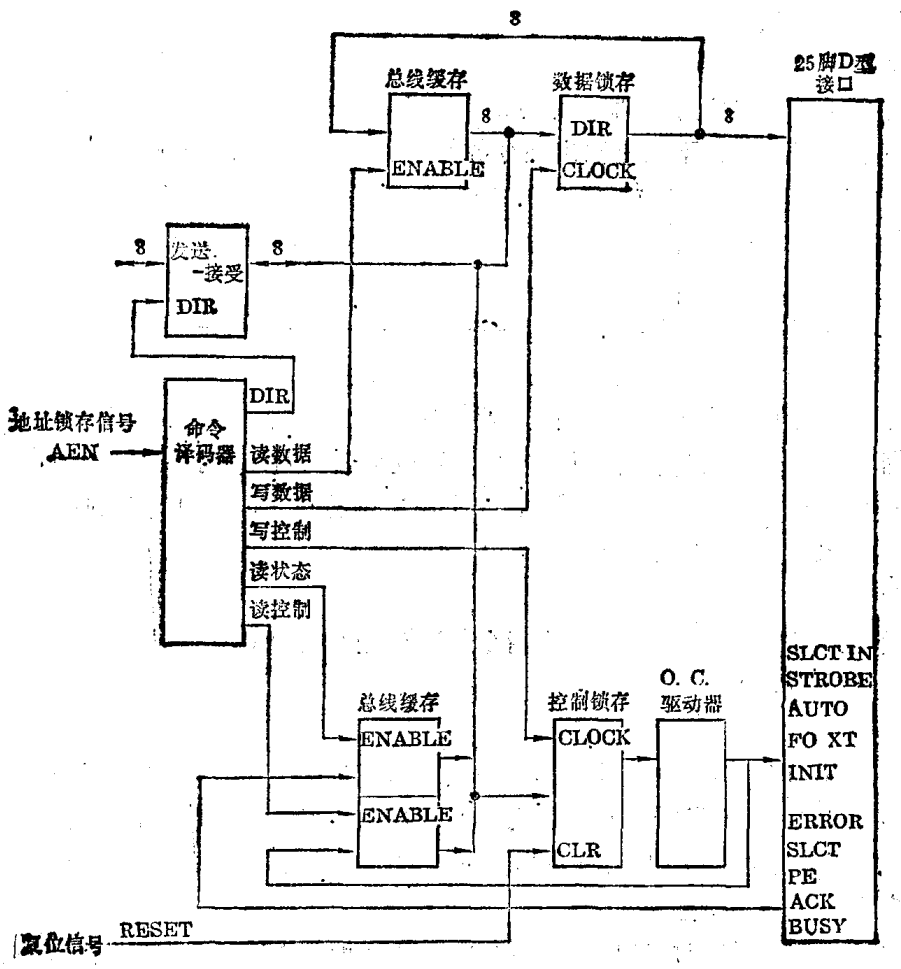


图 21-1 并行打印机转接器框图

§21.2 转接器编程要点

转接器能响应3条输入指令、2条输出指令。输出指令的作用是把数据打入锁存门，由锁存门输出端把这些数据加到25脚D型插座的一些引脚上。3条输入指令中的2条用来读取两个锁存门的内容，另一条指令读取一组插脚的实时状态。下面逐条介绍输出及输入指令。

输出指令1

对IBM单色显示器——打印机转接器本指令的I/O地址是3BCH，对并行打印机转接器，I/O地址为378H。下面是输出指令中数据字节与"D"型插座引脚的对应关系。

输出数据字节	7	6	5	4	3	2	1	0
插座引脚	9	8	7	6	5	4	3	2

说明：

本指令将数据线上的数据打到"D"型插座的某些引脚上。每个引脚可吸收电流2.4 mA，泄放电流2.6 mA。

输出指令2

对IBM单色CRT—并行转接器本指令的I/O地址是3BEH，对并行打印机转接器本指令的I/O地址为37AH。下面是输出指令数据字节与"D"型插座引脚的对应关系。

输出数据字节	4	$\bar{3}$	2	$\bar{1}$	$\bar{0}$
插座引脚	IRQ允使	17	16	14	1

说明：

本指令将数据线上的低5位数据引向"D"插座的某些引脚，且第0、1、3位被反置电平。当第4位数据为1同时插座第10引脚的电位由高变低时，转接器向CPU发中断。第1、14、16、17引脚的驱动方式为集电极开路式，由4.7K电阻拉在+5V。他们能吸收7mA左右的电流，低电平时保持在0.8V。

输入指令1

对IBM单色CRT—打印机转接器本指令的I/O地址是3BC，对并行打印机转接器本指令的I/O地址为378H。

说明：

本指令将以X′3BC′为I/O地址输出的数据字节返回给CPU。一般情况下，这个数据就是CPU最后一次写给X′3BC′口的数据。当外设正从引脚读数据（可能违反地址使用规则）时，CPU真正读到的数据将是引脚数据同锁存门数据的"或"。

输入指令2

对IBM单色CRT—打印机转接器本指令的I/O地址是X′3BD′，对并行打印机转接器本指令的I/O地址为X′379′。

说明：

本指令将下列引脚的实时状态送CPU。

读入字节	7	6	5	4	3	2	1	0
插座引脚	11	10	12	13	15	—	—	—

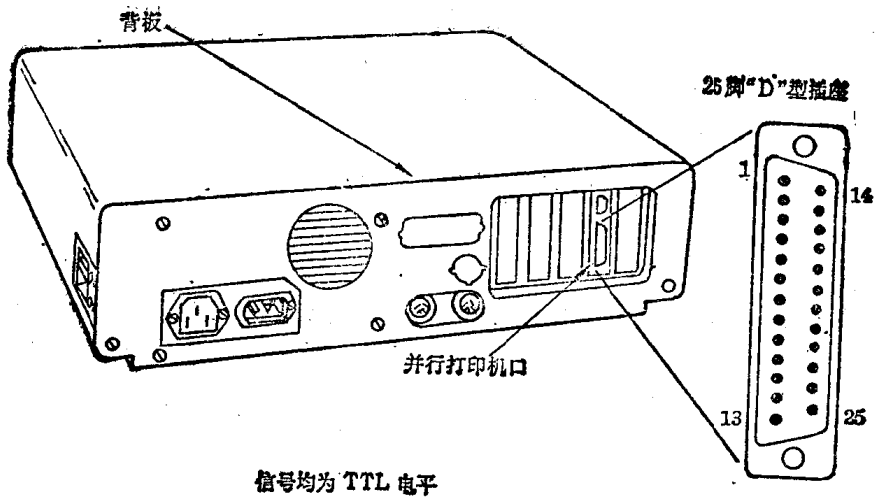
输入指令3

对于IBM单色CRT—打印机转接器本指令的I/O地址是X′3BE′，对并行打印机转接器本指令的I/O地址为X′37A′。

说明：

本指令将引脚1、14、16、17及IRQ的状态送给CPU。当外设不在使用这些引脚时，送出的数据就是最近一次写入X′3BE′口的数据。注意0-2位数据不包括在内。如果外设正访问这些引脚，读入的数据将是引脚数据同X′3BE′锁存门数据的"或"。

读入字节	7		6	5	4	3	2	1	0
插座引脚					允使IRQ $P_{or}=0$	$\overline{17}$ $P_{or}=0$	16 $P_{or}=0$	$\overline{14}$ $P_{or}=1$	$\overline{1}$ $P_{or}=1$



信号均为 TTL 电平

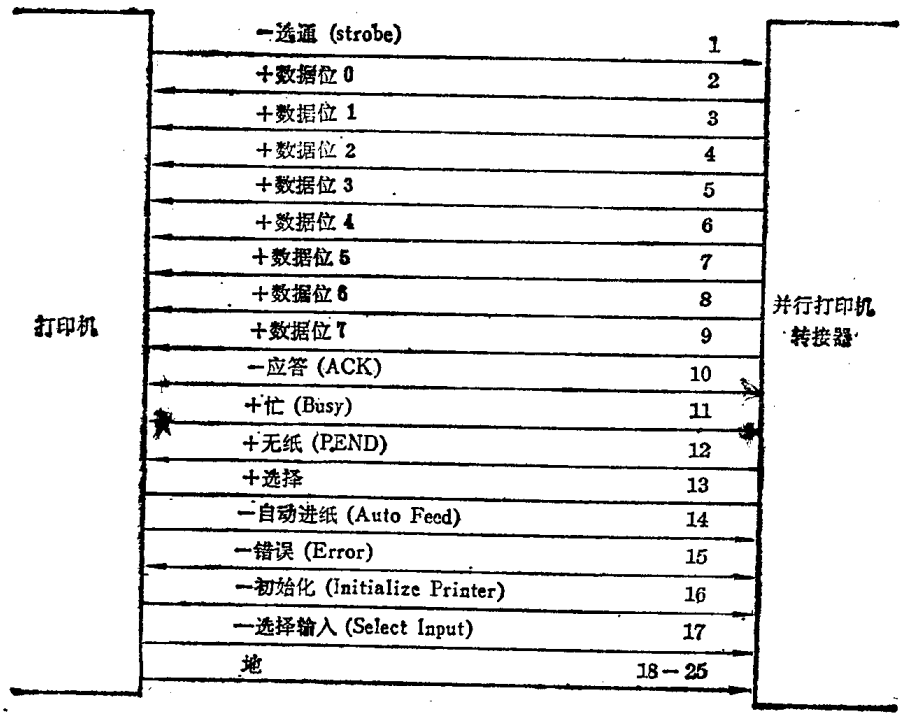


图 21-2 并行打印机转接器接口引脚安排

第二十二章　IBM 80 CPS 针极打印机

§22.1　概　　述

IBM 80 CPS 针极打印机，是一种每秒能打印 80 个字符(CPS)的双向针极式打印机，具有 9 针打印头，能打印 9×9 点阵的字符。96 个标准 ASC II 大小写字符、64 个特殊图形字符及它们的双倍字符、双点字符也都能打印。打印机每行打印字符数为：压缩方式时 132

表 22-1　IBM 80 CPM 针级打印机技术参数一览表

1. 打印方式：	串行点阵打击式		
2. 打印速度：	80 字符/秒		
3. 打印方向：	双向可逻辑搜寻打印位置		
4. 打印头针数：	9		
5. 行间距：	4.23 mm(1/6 吋)或程序选定		
6. 打印字符：			
点阵：	9×9		
字符集：	ASC II 字符全集(带译码器)+ 9 个国际字符/符号		
图形字符	64 个块符		
7. 打印尺寸	字符/吋	最多字符/吋	
一般：	10	80	
加大：	5	40	
压缩：	16.5	132	
压缩加大：	8.25	66	
8. 介质要求			
喂纸方式：	可调距链齿式		
纸宽：	101.6 mm～254 mm (4吋～10 吋)		
复印：	原纸加上两张厚度不超过 0.3 mm (0.012 吋)的碳复写纸		
纸张传送部位	背部		
9. 接口			
标准：	8 位并行数据及控制线		
10. 打印绸带			
颜色：	黑		
类型：	卷式		
寿命：	3 百万字符		
11. 使用环境			
温度：	5～35℃ (41～95°F)		
湿度：	10～80%		
12. 电源			
电压：	120 V 交流，60 HZ		
电流：	最大 1 A		
功耗：	最大 100 VA		
13. 物理尺寸			
高：	107 mm(4.2 吋)		
长：	374 mm(14.7 吋)		
宽：	305 mm(12.0 吋)		
重：	5.5 kg		

表 22-2　SWI

引　脚*	作　用	开	关	出厂状态
1	/	—	—	开
2	回车	只打印	打印、换行	开
3	缓冲区满	只打印	打印、换行	开
4	取消代码	无效	有效	关
5	删去代码	无效	有效	开
6	有错(蜂鸣器)	有声	无声	开
7	字符发生器	N.A	图形模式	关
8	选择 IN 信号	固定	不固定	开

个，标准方式时 80 个。大字方式时 66 个。另外行间距离可由程序定在 5 行/吋、8 行/吋或 10 行/吋，每行页数可定在 1～66 行/页。打印机通过电缆与系统单元连接，该电缆的打印机端是个 36 脚插头，转接器端是个 25 脚 D 型插头。

打印机 DIP 开关

打印机控制线路板上有两只 DIP 开关，拨动它们就可调节打印机的打印模式。表 22-2，22-3 列出 SW1、SW 2 DIP 开关合上、断开时的作用。图 -1 是 DIP 开关在控制板上位置示意图。

表 22-3 SW 2

引 脚	作 用	开	关	工厂状态
1	—	—	—	开
2	—	—	—	开
3	AUTO FEED $\overline{XT}$ 信号	固定	不固定	关
4	选择译码表	N.A.	标准	关

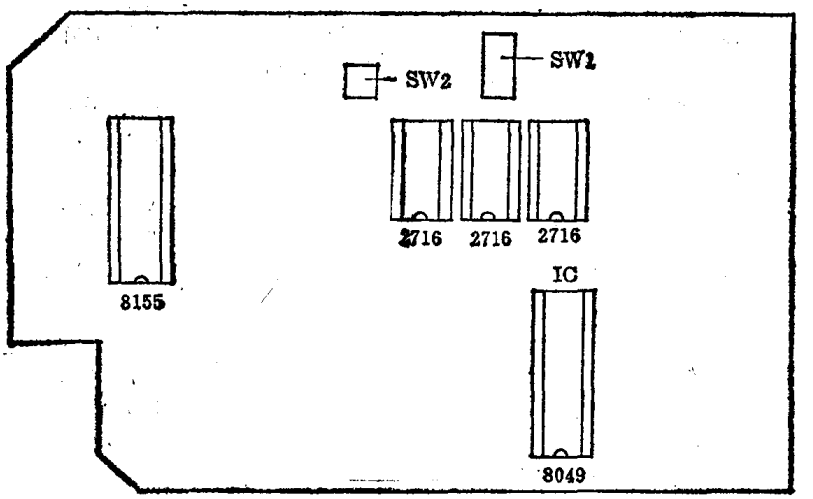


图 22-1 DIP 开关位置示意图

§22.2 打印机端并行接口

技术参数

1．数据传输率： 1000 CPS（最大值）

2．同步方式： 外部提供选通脉冲 STROBE

3．交换信号： $\overline{ACKNLG}$ 或 BUSY 信号

4．逻辑电位： 输入数据控、制信号电平与 TTL 兼容

5．插头型号： 57-30360 (AMPHENOL)

插口引脚与接口信号的关系（见表 22-4）

表 22-4

信号引脚*	返回引脚*	信　　号	方向	描　　述
1	19	STROBE	IN	高电平表示数据1，低电平表示0.
2～9	20～27	1～8位数据	OUT	应答信号，脉冲宽度5μs，低电平表示打印机已接收了当前的数据，允许送进下一个数据
11	29	BUSY	OUT	忙信号. 高电平表示打印机不能接收字符. BUSY 处高电平的原因： 1. 正输入数据；　2. 正在打印； 3. 脱机；　4. 打印机出错
10	28	$\overline{\text{ACKNLG}}$	OUT	应答信号，脉冲宽度5μs，低电平时表示打印机已接收了当前数据，允许送入下一个数据.
12	30	PE	OUT	打印纸用完信号，高电平有效
13	—	SLCT	OUT	高电平说明打印机处选择状态
14	—	AUTO FEED IN	IN	进纸信号：低电平时要求打印机在打完一行后自动进纸. 可由DIP SW 2-3 设置该信号恒为低电平.
15	—	NC		无用
16	—	0V		逻辑0电平
17	—		—	底盘地线. 打印机的底盘地线和逻辑地是分开的
18	—	NC		无用
19～30	—	GND	—	返回线，为地线电平
31	—	INIT	—	初始信号，一般保持在高电平. INIT 低电平时打印机控制器复位至初始状态，打印机缓存区清0，INIT 的低电平脉冲宽度不应小于 50 μs.
32		ERROR	OUT	出错信号. 低电平时打印机处： 1. 打印纸用完状态；　2. 脱机状态；　3. 出错状态
33	—	GND	-	作用同 19～30 线
34	—	NC	—	无用
35				由4.7 kΩ 电阻拉向 +5 V
36	—	SLCTIN	IN	低电平时数据可进入打印机，DIP SW1-8 设置该信号的状态

说明：

1) IN、OUT 均以打印机为观察方向，IN：主机→打印机，OUT：打印机→主机。

2) "返回线"表示双绞线中的返回线，即信号的地线或返回通路。双绞线应予以屏蔽，屏蔽地端与打印机和主机地端相连。

3) 接口电平均为 TTL 电平，信号脉冲上升或下降的时间不大于 0.2 μs。

4) 发送数据时不应忽略 $\overline{\text{ACKNLG}}$、BUSY 信号的作用。

5) 数据传送时间关系(见图 22-2)。

下面列出当 DIP SW 2-4 "关"、选择标准代码表时打印机能打印出的字符或执行某个动作的控制码的总表(表 22-5)。

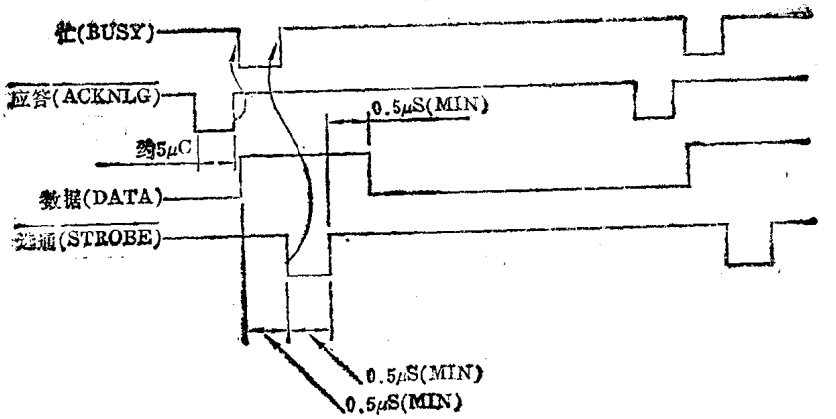


图 22-2　并行打印机接口时序图

表 22-5　ASCII 译码表

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	0000	NUL		SP	0	a	P	·	p	NUL							
1	0001		DC1	!	1	A	Q	a	q		DC1						
2	0010		DC2	"	2	B	R	b	r		DC2						
3	0011		DC3	#	3	C	S	c	s		DC3						
4	0100		DC4	$	4	D	T	d	t		DC4						
5	0101			%	5	E	U	e	u								
6	0110			&	6	F	V	f	v								
7	0111	BEL			7	G	W	g	w	BEL							
8	1000		CAN	(	8	H	X	h	x		CAN						
9	1001	HT		)	9	I	Y	i	y	HT							
A	1010	LF		*	:	J	Z	j	z	LF							
B	1011	VT	ESC	+	;	K	[	k	{	VT	ESC						
C	1100	FF		·	<	L	\	l	:	FF							
D	1101	CR		-	=	M	]	m	}	CR							
E	1110	SO		·	>	N	∴	n	~	SO							
F	1111	SI		/	?	O	—	o	DEL	SI							

§22.3 打印机控制码

ASCII 执行代码(控制码)

1．回车 (CR)

回车码 (CR) 发至打印机缓存区后，打印机将缓存区里的数据全部打出。若 $\overline{\text{AUTO FEED IN}}$ 为低电平或 DIP SW 2-2 处"开"状态，打印机在打印数据后自动进纸一行。打印机在连续收到 80 个字符(包括空格)并得知后面可能跟有有效打印数据的话，将自动打印出缓存区里的数据，如果这时 $\overline{\text{AUTO FEED IN}}$ 低电平或 DIP SW 2-2 为"开"状态，打印机也进纸一行。

2．换行 (LF)

打印机收到换行码后，将打印缓冲区里的数据打完，然后换行。若换行码前没有数据或全部是"空格"码，打印机进纸一行。

3．垂直标记(VT)

收到 VT 码后，打印机打印出 VT 码前的数据，然后进纸至 ESC B 指定的那一行。如果 ESC B 未设定垂直标记的位置，VT 码的作用与换行码一样，只进纸一行。

4．格式馈给 (FF)

收到 FF 码后，打印机在打印完缓冲区里的数据后，进纸至下一个预先定义好的格式位置。首格式位置由电源开关打开、并由 $\overline{\text{INIT}}$ 信号有效时决定。如果 "ESC C + n" 控制码未指明页格式长度的话，打印机认为页格式长度是 66 行或 77 行。

注意：只有背部印有 M 72 字样的打印机，才可设置 72 行页格式长度。

5．移出 (SO)

收到 SO 码后，打印机把与 SO 同行在SO码后的全部字符以"双倍"宽度打印出．SO 码可被在其后输入的 DC 4 码取消．利用 SO 码可在一行中混合打印出常规及双倍宽度两种字符。

实例 1： [DATA] ABC [SO] DEF [DC4] GHI [CR] [LF]

打印结果：ABC D E F GHI
（D E F：双倍宽度）

实例 2： [DATA] ABCD [SO] EFGH [CR] [LF] IJKL [SO] MNOP [CR] [LF]

打印结果：A B C D E F G H
I J K L M N O P

6．移进 (SI)

收到 SI 码后，打印机将 SI 码后的数据，以压缩字符形式打印出。如果 SI 码后跟上 SO 码，打印压缩加大字符(双倍宽度的压缩字符)，若再输入 D 4 码，字符尺寸又回到"压缩"形式。"DC 2"码取消 SI 码的作用．缓冲区每行能存贮 132 个压缩字符数据。

实例 1： [DATA] [SI] ABCDEFGHIJKL [CR] [LF]

打印结果：ABCDEFGHIJKL (压缩尺寸)

实例 2： [DATA] ABC [SI] DEF SO GHIJ KL [CR] [LF]

打印结果：ABC DEF GHIJKL
（DEF：压缩　GHIJKL：压缩加大）

7．设备控制 4 (DC 4)

DC 4 码取消 SO 码的作用。

实例：[DATA] [SI] ABCDEF [SO] GHI [DC4] JKL [CR] [LF]

打印结果：ABCDEF G H I JKL

压 缩 压缩加大 压缩

8．设备控制 2 (DC 2)

DC 2 码取消 SI 码的作用。

实例：[DATA] [SI] ABCDEF [SO] GHI [CR] [LF] [DC2] JKLMN

打印结果：ABCDEF G H I

压 缩 压缩加大

JKLMN

压 缩

9．水平标记 (HL)

HL 码将打印头移到"ESCD＋n"码规定的地方。

10．取消 CAN

收到 CAN 码后，以前存在打印缓存区里的数据(字符代码)被清除，但缓存区里的控制码 (不包括 SO) 仍有效，故 CAN 码可作清除命令使用。CAN 码的有效性，由 DIP SW 1-4 的状态决定。

11．删除 DEC

DEC 码的功能同 CAN 码，其有效性由 DIPSW 1-5 开关决定。

12．设备控制 1 (DC 1)

DC1 码将打印机置成选择状态。如果在传送数据过程中输入 DC1 码，DC1 码前的全部数据无效。

13．设备控制 3 (DC 3)

DC 3 码将打印机置成非选择状态，即打印机不能接收数据的状态。这样，打印机不能自动恢复到选择状态，除非输入 3 DC1 码。

注意：

只有当 DIP SW 1-8 处 OFF 状态时，才能使用 DC 1、DC 3 码。

实例 1：[DATA] AAAAA [DC3] BBBBB [DC1] CCCCC [CR] [LF]

打印结果：AAAAA CCCCC

实例 2：[DATA] AAAAA [DC1] BBBBB [DC3] CCCCC [DC1] [CR] [LF]

打印结果：BBBBB

下面是联机开关(ON-LTNE)、SLCTIN 开关。DC1/DC 3 码和接口信号间的相互关系表：

说明

* 表中假定打印机收到字符后立即发 ACKNLG 信号，虽然该数据还未送入打印缓存。在上面情况下，打印机将为一般输入等待 DC 1 码。

** DIP SW 1-8 处"关"态时，即接口插座第 36 脚，SLCT IN 信号为高电平 DC 1/DC 3 码有效，$\overline{\text{SLCT IN}}$ 低电平时，DC 1/DC 3 码无效。

表 22-6 DC 1/DC 3 与数据进打印机的关系

ONTINE SWITCH	SLCT IN	DC1/DC3	$\overline{\text{ERROR}}$	BUSY	$\overline{\text{ACKNLG}}$	SLCT	数据进打印机
联　机	高/低	DC1/DC3	低	高	不产生	低	不可以
脱　机	高	DC1	高	低/高	产生	高	可以(一般输入)
		DC3	高	低/高	产生	低	可以*
	低	DC1/DC3	高	低/高	产生	高	可以(一般输入)

14. 零 (NUL)

NUL 码是标定序列结束符，其作用见下面几个代码的说明。

15. 响铃 (BEL)

收到 BEL 码后，打印机内蜂鸣器鸣响 3 秒。

16. 换码 (ESC)

a. 数值控制换码符，ESC 码后跟 ASCII 数值符完成下列动作：

① ESC O (Escape O)

ESC 后跟 ASCII 码 O 表示将行间距定为 1/8 吋。ESC 2 命令、发向接口的 $\overline{\text{INIT1}}$ 信号或关机后再开机将行间距恢复至 1/6 吋。

② ESC 1 (Escapel)

ESC 后跟 ASCII 码 1 表示：将行间距定为 7/12 吋。ESC 2 命令、发向接口的 $\overline{\text{INIT}}$ 信号或关机后再开机，将行间距恢复至 1/6 吋。

③ ESC 2 (Escape 2)

ESC 后跟 ASCII 码 2 表示：把行间距定为 1/6 吋。POWER 开关打开后的行间距也是 1/6 吋。ESC 2 码也是执行"ESCA + n"码的命令之一。

④ ESC 8 (Escape 8)

ESC 后跟 ASCII 码 8 允许在打印机无纸情况下，向打印机发送数据。ESC 8 应在打印机有纸时发出，一旦打印机纸用完，该命令就使接口上的 PE 信号变高电平，从而维持 $\overline{\text{ERROR}}$ 高电平。

⑤ ESC 9 (Escape 9)

ESC 后跟 ASCII 码 9 组成的控制码，将取消 ESC 8 命令的作用。开机时打印机处 ESC9 状态，也就是说发现无打印纸的话，打印机不接收数据。

⑥ ESC SI

同 "SI"。

⑦ ESC SO

同 "SO"。

b) 控制符为字母的换码命令

ESC 后跟 ASCII 字母组成的控制码的定义如下：其中参数"n"表示 7 位 2 进制数，最高位不计。"+"号是便于阅读插进的，实际输入时不需要，

① ESC A + n

这个控制码设置换行时的行间距为 n × 1/72 吋。参数 n 的取值范围：$1_{10} \leqslant n_{10} \leqslant 85_{10}$，n = 1 表示行间距等于 1/72 吋，由于打印针头两两间距 1/72 吋，所以加宽后的行间距总是 1/72 吋的整倍数。ESC A + n 码只记录行间距的大小，打印机在收到跟在其后的 ESC 2 命令后，才根据 ESC A 码改变行间距。

实例：

[DATA] AAAAAAA [CR] [LF] BBBBBBB [CR] [LF] [ESC A + 24]
CCCCCCC [CR] [LF] DDDDDDD [ESC 2] [CR] [LF]
EEEEEEE [CR] [LF] FFFFFFF [CR] [LF]

打印结果：

AAAAAAA
BBBBBBB } 行间距 1/6 吋 (12 × 1/72 吋)
CCCCCCC
DDDDDDD
EEEEEEE } 行间距 1/3 吋 (24 × 1/72 吋)
FFFFFFF

注意：

n应以 7 位 2 进制 ASCII 码形式输入。譬如对 ESC A + 24，送给打印机的数据串应是 <1B>H<41>H<18>H。

② ESC B + n 1 + n 2 + nK + NUL ($1_{10} \leqslant n_{10} \leqslant 66_{10}$, $1 \leqslant K \leqslant 64$, $nK \leqslant nK + 1$)

ESC B 码指定垂直标记停止位置。打印机最多能识别 64 个代表停止行递增的数字，后面的数据被忽略。打印机不执行超过页长的标记定行动作。此外，打印机在收到 VT 码后才执行 ESC B 码，移向指定的标记位置，在输入新标记值前，最近一次送入的标记参数总有效。

如果没有设定标记定位值，VT 码完成进纸一行动作。

ESC B 后的数字串代表下次开始打印行的行号，应以 NUL 结尾。缺少 NUL 会产生打印错误。一般在 ESC B 码前送入 ESC C 码，定义好页格式尺寸。ESC B NUL 将使前面定义的停止行号失效。

实例：

[DATA] [ESC B] <4>H <6>H <A>H [NUL]
AAAAAAA [VT] BBBBBBB [VT] CCCCCCC [VT] DDDDDDD

打印结果：

AAAAAAA……第 1 行
BBBBBBB……第 4 行
CCCCCCC……第 6 行
DDDDDDD……第10行

③ ESC C + n ($1_{10} \leqslant n_{10} \leqslant 66_{10}$)

ESC C 码设定页的长度（页格式）为 n 行，不用 ESC C + n 指定页格式的话，每页长 66 行或 72 行。

④ ESC D + n 1 + n 2 + … + nK + NUL (1≤n_{10}≤127, K≤112)

ESC D 码指定水平标记停止位置。打印机最多能识别出 112 个以递增序排列，表示行中停止位置的数据。工作于标准字符打印格式的打印机不考虑大于 80 的列位置数据。NUL 是列位置数据序列的结束符，缺省时将出现错误。打印机在收到 HT 码后才执行 ESC D 码。无 ESC D 码的 HT 命令无效。

实例 1： 列停止位置为 5，10，21。

[DATA] [ESC D] 〈5〉 H 〈A〉 H 〈15〉 H [NUL] ABC [HT]
DEF [HT] GHI [HT] JKL [CR] [LF]

打印结果： ABC DEF GHI JKL

实例 2： 列停止位置 5，10。

DATA [ESC D] 〈5〉H 〈A〉H [NUL] ABC [HT] DEF [HT] GHI [HT] JKL
[CR] [LF]

打印结果：ABC DEF GHIJKL

实例 3： 打印字符数超过设定的列停止位置。

DATA [ESC D] 〈5〉H 〈A〉H 〈15〉H [NUL] ABCDEF [HT] GHI [HT] JKL
[CR] [LF]

打印结果：ABCDEF GHI JKL

实例 4： 一次输入两个 HT 码。

DATA [ESC D] 〈5〉H 〈A〉H 〈15〉H [NUL] ABCD [HT] [SPACE] [HT] EFGH
[CR] [LF]

打印结果：ABCD EFGH

⑤ ESC E

ESC E 码命令打印机把字符的颜色打得深一些，它可出现在一行的任何地方。打印加深颜色字符时，滑架速度减为 40 字符/秒。

实例： [DATA] [ESC E] ABCDEFGHI [CR] [LF]

打印结果： ABCDEFGHI
深 些

⑥ ESC F

ESC F 码取消 ESC E 码的作用。

⑦ ESC G

ESC G 码命令打印机完成双重字符打印动作。双重打印的步骤是：

a. 打印字符一遍

b. 进纸 1/216 吋

c. 再打印字符一遍

双重打印字符的轮廓较清楚。

实例： [DATA] [ESC G] ABCDEFGHI [CR] [LF]

打印结果： ABCDEFGHI
加 深

⑧ ESC H

ESC H 码取消 ESC G 命令的作用。

⑨ ESC K; n1; n2; V1; V2; …VK

本命令用于 IBM 80 CPS 图形打印机，它将打印机由文本模式转到位象图形模式。命令中的 n1，n2 是 V1，V2…VK 个位象数据字节的个数 ($K = n_1 + 256\,n_2$, $K \leqslant 480$) 在同一行里，位象数据可与文本数据混用。

送至打印机的数据字节序列的格式：

文本 (20 字符)	ESC K n=360	位一象数据序列	下一数据块

说明：

头 20 个字符是文本模式下打印的字符，它占据 120 个 (20×6) 位一象点，图形模式仅能打印 360 个位一象点 (480－120＝360)。

实例：在 480 位象图形模式下打印一行痕迹。

(BASIC (程序)

```
 1 'OPEN PRINTER IN RANDOM MODE WITH LENGTH OF 255
 2 OPEN"LPTI:"AS#I
 3 WIDTH"LPTI:", 255
 4 PRINT#1, CHR$(13), CHR$(10);
 5 SLASH$ = CHRS(1) + CHR$(2) + CHR$(4) + CHR$(8)
 6 SLASH$ = SLASH$ + CHR$(16) + CHR$(32) + CHR$(64) + CHR$(128)
   + CHR$(0)
 7 GAP$ = CHR$(0) + CHR$(0) + CHR$(0)
 8 NDOTS = 480
 9 'ESC K N1 N2
10 PRINT#1, CHR$(27), "K", CHR$(NDOTS MOD 256),
   CHR$(INT(NDOTS/256));
11 'SEND NDOTS NUMBER OF BIT IMAGE BYTES
12 FOR I = 1 TO NDOTS/12 'NUMBER OF SLASHES TO PRINT USING
   GRAPHICS
13 PRINT#1, SLASH$, GAP$;
14 NEXT I
15 CLOSE
16 END
```

表 22-7　图形打印机字符集 1

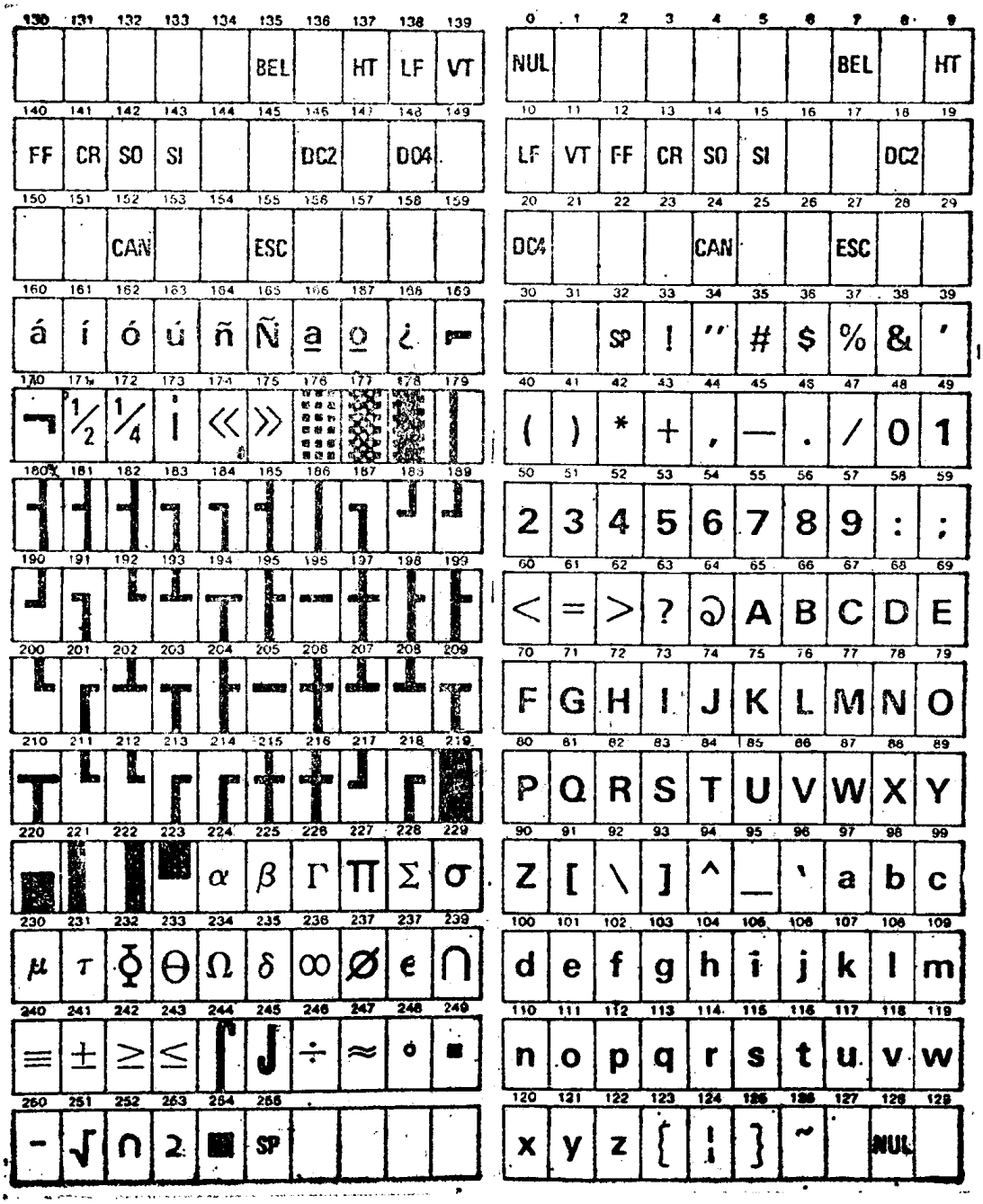
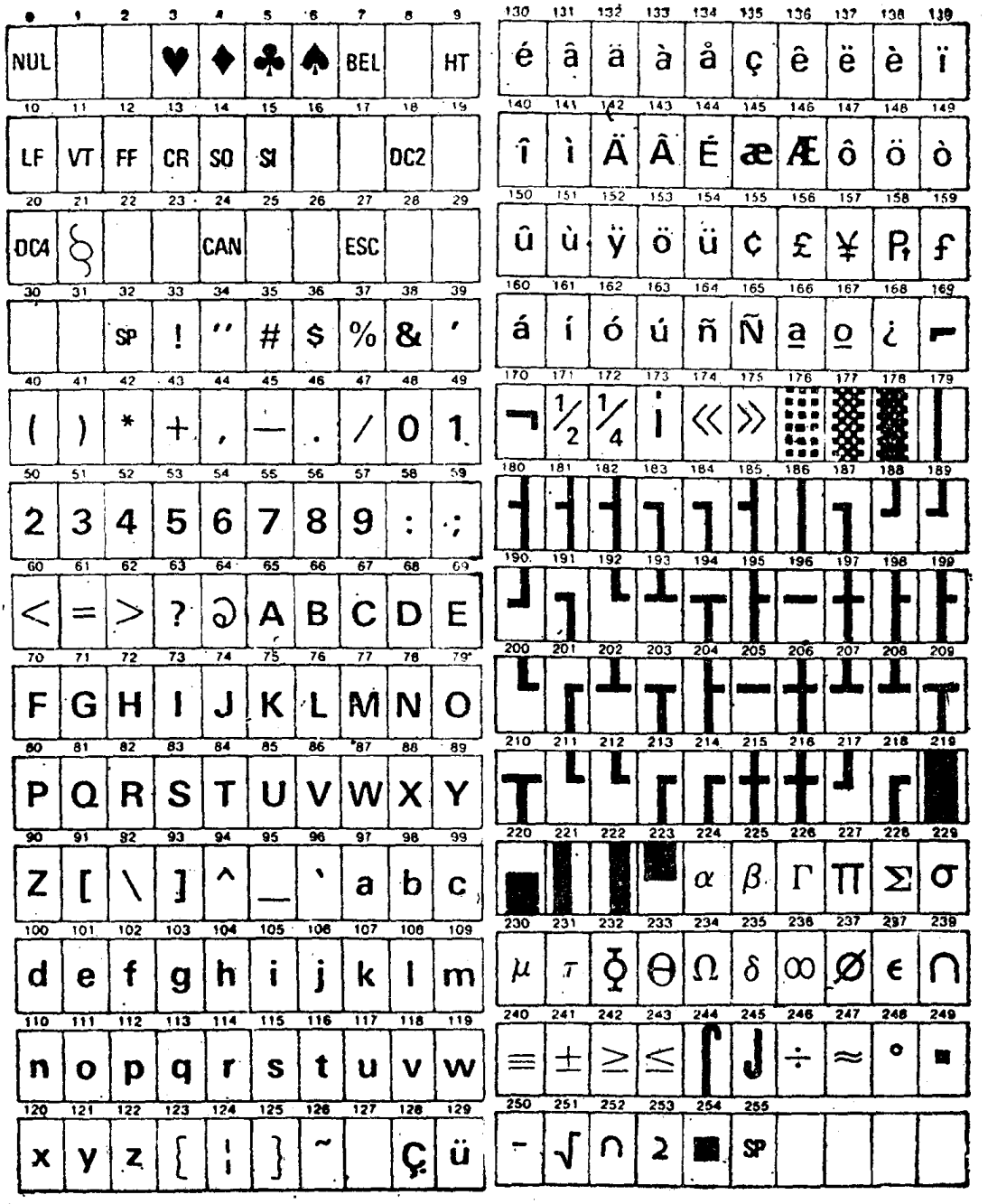


表 22-8　图形打印机字符集 2



第二十三章　5－1/4时软盘机转接器及5－1/4时软盘驱动器

§23.1　概　　述

软盘转接器（以下简称转接器）插在I/O扩展槽上，通过扁平询电缆与系统单元内的两台内部5-1/4时软盘机相连。转接器还有一个插座，通过穿出系统单元后盖板的电缆，连接两台5-1/4时外接软盘机。一个转接器可带4台软盘机。

转接器是为适配双密度调频制（MFM）软盘机设计的，采用帮助恢复时钟、数据的模拟锁相环写预压缩技术，能支持写保护操作。转接器的核心是NEC μPD765可兼容控制器。软盘机的各工作参数均可程序。

从转接器与系统的连接方式看，转接器挂在I/O总线上，数据均以DMA方式传送，操作完成等需引起注意的信号以中断方式通知CPU；从逻辑上看，转接器是系统与软盘机之间的高级命令接口。

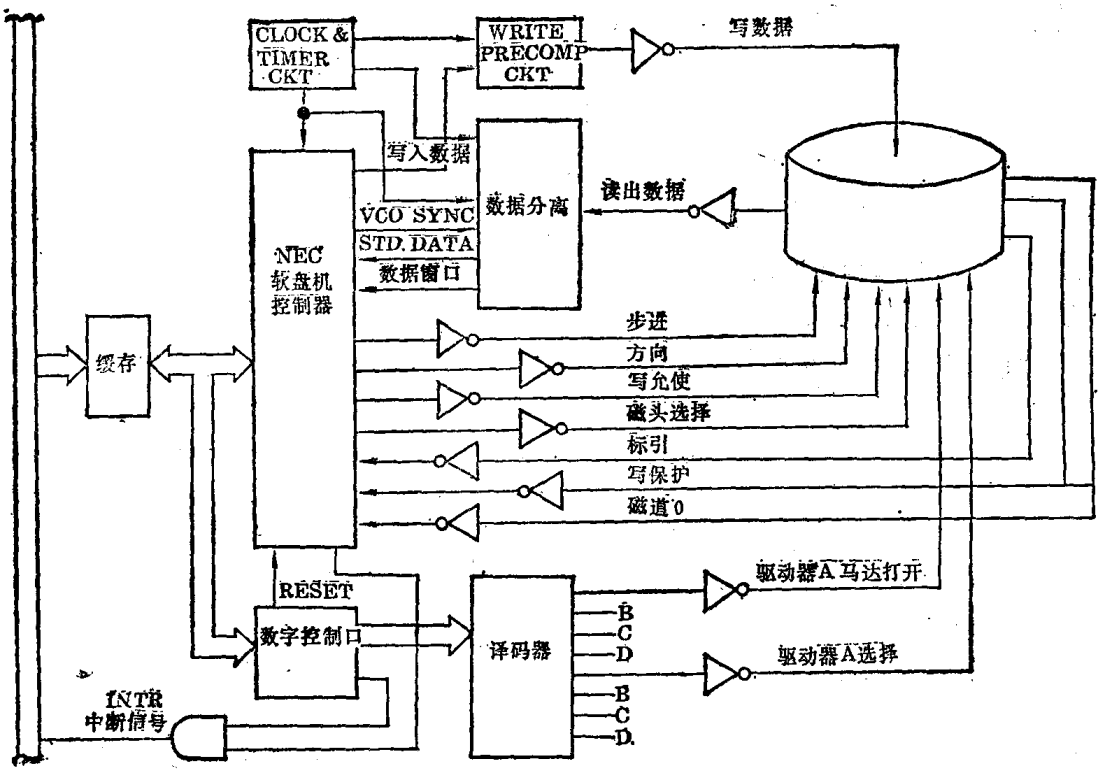


图 23-1　5-1/4 时软盘机转接器框图

§23.2 转接器功能部件

从编程角度看，转接器是一个由8位数据输出寄存器和NEC μPD 765控制器或相同功能软盘机控制器简称FDC组成的I/O口。

一、数据输出寄存器(DOR)

DOR是只输出寄存器，用以选择软盘机，允使某个动作，控制软盘机马达I/O及I/O复位线电平、请零DOR所有位。DOR各位的作用是：

位0，1　选择软盘机(如果马达正打开着)

0，1 位译码表

位		软盘机号
0	1	
0	0	0或A
0	1	1或B
1	0	2或C
1	1	3或D

位2　等于0时FDC保持复位状态，FDC的置位必须由程序进行。

位4、5、6、7这4位分别控制软盘机0、1、2、3或称A、B、C、D马达的启停。为1时马达启动。

二、软盘机控制器 FDC

FDC有两个处理器可访问的寄存器，一是记录FDC状态信息的8位主状态寄存器，只能读；另一是存贮数据、命令、参数和状态的数据寄存器。实际上FDC有多个数据寄存器，他们构成一个堆栈，但每次只有一个寄存器能与I/O数据总线相连。读、写数据寄存器的目的是编程和获取执行某条命令后的结果。

下面是主状态寄存器各个位的定义。(FDD代表软盘机)

表23-1　主状态寄存器

位*	名　称	符号	意　义
0	FDD A忙	DAB	0号FDD正处搜寻模式
1	FDD B忙	DBB	1号FDD正处搜寻模式
2	FDD C忙	DCB	2号FDD正处搜寻模式
3	FDD D忙	DDB	3号FDD正处搜寻模式
4	FDC 忙	CB	FDC正处理读/写命令
5	非DMA模式	NDM	FDC处非DMA模式
6	数据输入/输出	DIO	指出FDC和CPU间数据的传送方向．为1 FDC数据寄存器至CPU方向；为0 CPU至FDC数据寄存器方向．
7	主机请求	RQM	数据寄存器已准备好向CPU发送或从CPU接收数据DIO位和RQM位用作FDC同CPU的"Ready"和"direction"交换信号

FDC 可执行 15 条转接器命令，每条命令都是以 FDC 接收 CPU 发来的多数据字节开始，以向 CPU 返回执行结果(可能仍是多数据字节)结束。整个命令的执行过程可分成如下三个阶段：

命令阶段：FDC 从 CPU 接收完成某个软盘机动作必需的全部数据字节。

执行阶段：FDC 执行命令。

结果返回阶段：FDC 将状态信息、交换信息返回处理器。

三、FDC 编程要点

1．命令数据字节用到的符号定义：

符　号	名　称	意　义
A_0	地址线	A_0 选择主状态寄存器(A_0=0)或数据寄存器(A_0=1)
C	磁道数	当前选定的磁盘道号
D	数据	写入扇区的数据格式
D_0～D_7	数据总线	8 位数据线，D 7 为最高位，D 0 最低位
DTL	数据长度	N 定义成 00 时，DTL 代表用户要写入或读出的数据的长度
EOT	道结束	EOT 代表道上最后一个扇区的区号
GPT	间隔长度	代表不包括 VCO 同步场在内扇区间的距离即间隔 3 的长度
H	磁头位置	指示第一磁头或第二磁头，相当于 ID 标记域的内容
HLT	磁头加载时间	FDC 磁头加载时间在 4～512 ms 之间，增量 4 ms
HUT	磁头去载时间	一次磁头读/写操作后的去载时间在 0～480 ms 之间，增量 32 ms
MF	FM 或 MFM 调制方式	MF=0 选择 FM 方式　MF=1 选择 MFM 方式
N	字节数	写入一个扇面的字节数
NCN	新磁道号	搜寻操作将达到磁道的道号
ND	非 DMA	软盘机以非 DMA 方式工作
PCN	当前磁道号	完成中断状态测试命令后磁头处磁道的道号
R	记录	将进行读/写操作扇区的区号
R/W	读或写	表示读信号或写信号
SC	扇区数	每张磁盘具有的扇区数
SK	跳跃	跳跃删除数据地址标志
SRT	步距率时间	FDC 步距改变率，2～32 ms，增量 2 ms
ST 0 ST 1 ST 2 ST 3	状态 0 状态 1 状态 2 状态 3	ST 0-3 是 4 个记录一条命令执行后信息的状态寄存器组，不要把 ST 0-3 与主状态寄存器 (A_0=0 选择的)相混
STP	扫描测试	在扫描操作中，若 STP=1，连续扇区中的数据逐字节地与 CPU(DMA) 发来的数据相比较，STP=2 时只拿选择扇区的数据作比较
US 0 US 1	单元选择	US0、US1、指定软盘机，与数据寄存器 DOR 第 0、第 1 位的内容相同

2. FDC 命令 (X 表示三态, R/W 表示读/写)

1) 读数据

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
		读数据								
命令	W	MT	MF	SK	0	0	1	1	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	C								扇区标识字节
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
执行										FDD 将数据送主机返回的结果数据字节,共 7 个
结果	R	ST0								读命令后扇区标识信息(位置)
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

2) 读删除数据

阶段	R/W	数据总线 D7	D6	L5	D4	D3	D2	D1	D0	注释
命令	W	MT	MF	SK	0	1	1	0	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	C								扇区标识信息
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
执行										FDD 数据送主机
结果	R	ST0								读操作后扇区标识
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

3) 写数据

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	WT	WF	0	0	0	1	0	1	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W				C					定位信息
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
执行										主机数据送 FDD
结果	R				ST 0					写操作后扇区的位置
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					
	R				R					
	R				N					

4a) 写删除数据

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	MT	MF	0	0	1	0	0	1	命令码
	W	×	×	×	×	×	H0	US1	US0	
	W				C					定位信息
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
执行										主机数据送 FDD
结果	R				ST 0					写操作后扇区的位置
	R				ST 1					
	R				ST 2					
	R				C					
	R				H					
	R				R					
	R				N					

4b) 读一个磁道

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	MF	SK	0	0	0	1	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	C								扇区标识
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
执行										FDC 读出标引孔至 EOT 所有磁道的内容
结果	R	ST0								状态字节
	R	ST1								
	R	ST2								
	R	C								扇区新位置
	R	H								
	R	R								
	R	N								

5) 读标识

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
		读数据								
命令	W	0	MF	0	0	1	0	1	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
执行										道上第一个正确的扇区标识送数据寄存器
结果	R	ST0								执行阶段中扇区标识
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

6) 磁道格式化

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
		读数据								
命令	W	0	MF	0	0	1	1	0	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	N								
	W	SC								
	W	GPL								
	W	D								
执行										FDC 格式化整个磁道状态字节
结果	R	ST0								扇区标识(无意义)
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

7) 等同扫描

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	MT	MF	SK	1	0	0	0	1	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	C								扇区标识
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
执行										比较 FDD、主机的数据
结果	R	ST0								比较后的扇区标识
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

8) 小于等于扫描

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	MT	MF	SK	1	1	0	0	1	命令码
	W	×	×	×	×	×	HD	US1	US2	
	W	C								扇区标识
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
执行										比较 FDD、主机的数据
结果	R	ST0								比较后扇区标识
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

9) 大于等于扫描

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
		SCAN HIGH OR EQUAL								
命令	W	MT	MF	SK	1	1	1	0	1	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W	C								扇区标识
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
执行										比较 FDD、主机的数据
结果	R	ST0								比较后扇区标识
	R	ST1								
	R	ST2								
	R	C								
	R	H								
	R	R								
	R	N								

10) 重校

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	0	0	0	0	1	1	1	命令码
	W	×	×	×	×	×	0	US1	US0	
执行										磁头返回00道

11) 查中断状态

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	0	0	0	1	0	0	0	命令码
结果	R					STO				返回FDC寻找操作信息
	R					PCN				

12) 设方式

阶段	R/W	D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	0	0	0	0	0	1	1	命令码
	W	—SRT—						—HUT—		
	W	——HLT——							—ND	
结果										无

13) 取驱动器状态

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	0	0	0	0	1	0	0	命令码
	W	×	×	×	×	×	HD	US1	US0	
结果	R					ST3				FDD状态

14) 寻找

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W	0	0	0	0	1	1	1	1	命令码
	W	×	×	×	×	×	HD	US1	US0	
	W					NCN				
结果										磁头被移至指定的磁道

15) 无效

阶段	R/W	数据总线 D7	D6	D5	D4	D3	D2	D1	D0	注释
命令	W					无效命令码				命令码
结果	R					ST0				不动作，FDC转预备态ST0=80

3. ST0-3 命令状态寄存器

命令状态寄存器 ST0

位*	名　　称	符　号	意　　义
D7	中断码	IC	D7　D6
D6			0　0　命令正常结束 0　1　命令非正常结束 1　0　非法命令 1　1　执行时从 FDD 发出的“Ready”信号改变了状态，命令非正常结束
D5	搜索结束	SE	FDC 完成搜寻命令后 DS=1
D4	设备检查	EC	收到 FDD 发来的错误信号或 0 道信号在 77 个步进脉冲后未发出，D4=1
D3	未准备好	NR	D3=1 表示 FDD 处未准备好状态就发来了读写信号，或发出读写单面磁盘第一面的命令
D2	磁头地址	HD	表示中断时磁头的状态
D1 D0	软盘机选择 US0 软盘机选择 US1		D1、D0 表示中断时的驱动器号

状态寄存器 ST1

位*	名　　称	符　号	意　　义
D7	磁盘结束	EN	FDC 访问超出磁道最后扇区，D7=1
D6	—	—	无用 D3 恒等于 0
D5	数据错	DE	FDC 检查出标记域及数据域的错误后置 D5 为 1
D4	超时错	OR	传送数据时FDC在一定时间间隔内未得到 CPU 的服务，D4=1
D3	—	—	无用，D3 恒为 0
D2	无数据	ND	D2 在下面三情况下为 1： ① 执行读、写、删除数据或扫描命令时，FDC 找不到标记域指定的扇区 ② 执行读标记域命令时读出的标记域数据有错 ③ 执行读磁道命令时，未找到起始扇面
D1	不可写	NW	执行写数据命令、写删除数据命令或磁道格式化命令时，FDC检测到 FDD 的写保护信号：D1=1
D0	地址标识丢失	MA	FDC 未检测到标记域地址标志时置位 D0，状态寄存器 ST2 中的相应位也被置位

状态寄存器 ST2

位*	名　　称	符　号	意　　义
D7	—	—	无用 D7 恒等于 0
D6	控制符	CM	FDC 在执行读数或扫描命令时发现扇区中含有删除数据地址标志：D6=1
D5	数据域中的数据有错	DD	FDC 查出数据有 CRC 错：D5=1

位*	名　　称	符　号	意　　义
D4	磁道错	WC	D4 与 ND 连用，C 的内容不等于 ID 寄存器时 D4=1
D3	扫描相等	SH	执行扫描命令时满足"相等"条件：D3=1
D2	扫描不等	SN	执行扫描命令时满足"不相等"条件：D2=1
D1	磁道坏	BC	D1 与 ND 位连用，C 的内容不等于 ID 寄存器内容且 C=FF 时 D1=1
D0	数据域丢失地址标志	MD	已读进了数据但 FDC 找不到数据地址标志或删除数据地址标志：D0=1

状态寄存器 ST3

位*	名　　称	符　号	意　　义
D7	错误	FT	表示 FDD 错误信号的状态
D6	写保护	WP	表示 FDD 写保护信号的状态
D5	准备好	RY	表示 FDD 准备好信号的状态
D4	0 号磁道	T0	表示 0 号磁道信号的状态
D3	双面	TS	表示 FDD 双面信号的状态
D2	磁头地址	HD	表示 FDD 盘面选择信号的状态
D1	单元选择 1	US1	表示 FDD 单元选择 1 信号的状态
D0	单元选择 0	US0	表示 FDD 单元选择 0 信号的状态

4．FDC 编程要点总结

(1) 驱动器控制器(DPC)寄存器：

FDC 数据寄存器　　I/O 地址 3F5

FDC 主状态寄存器　　I/O 地址 3F4

数据输出寄存器　　I/O 地址 3F2

(2) 中断级别：6

(3) DMA 通道：2

(4) 磁盘格式：100

(5) 软盘技术参数：单磁头　45 磁道　每道 8 扇面　每扇面 512 字节　MFM 方式

(6) FDC 常数

N：H′02′，SC：08，HUT：F，SRT：C，GPL (格式化)：H′05′，GPL(读/写)H′：2A′，HLT：01 (8ms 磁道-磁道)

(7) 驱动器运行参数：

磁头加载时间　　35 ms

磁头稳定时间　　25 ms

马达启动时间　　500 ms

注意事项

1) 选定驱动器后再加载，读/写时应留足加载时间

2) 连续访问时应等过一个稳定时间再读/写.

3) 只有 A、B 或 C、D 两驱动器能同时使用. 注意马达有启动时间.

4) 家用电视机不得太靠近磁盘,否则易出现数据错误

§23.3 系统 I/O 通道接口

这里的系统 I/O 通道接口指软盘机转接器与系统 I/O 总线的接口. 接口上各信号的电平均与 TTL 兼容, 其中 MPUL = 5.5 Vdc, LPUL = 2.7 Vdc, MPDL = 0.5 Vdc, LPDL = −0.5 Vdc. 下面是转接口各信号线的说明:

+DO−7　双向数据线,传送命令、状态和数据,负载 74 LS, 驱动器三态 74 LS. DO 为最低位.

+AO−9　(转接器输入,负载 74 LS)
选择寄存器的地址线.

+AEN　(转接器输入,负载 74 LS)
AEN 线有效时, A 0−9 线无作用

−IOW　(转接器输入,负载 74 LS)
IOW 的下降沿将 D0−7 上的数据打入 A0−9 或 DACK 2 指定的寄存器

−IOR　(转接器输入,负载 74 LS)
IOR 的下降沿将 A0−9 或 DACK 2 指定寄存器的数据打入 D0−7 数据线

−DACK2　(转接器输入,负载 2 个 74 LS)
DACK 2有效时 DRQ 2 失效,并选择 FDC 数据寄存器作为 D0−7 数据线的源/目寄存器或间接地将 T/C 打入 IRQ 6.

+T/C　(转接器输入,负载 4 个 74 LS)
T/C 线、DACK 2 线全有效时表示初始 DMA 的数据字节正在传送中

+RESET　(转接器输入,负载 74 LS)
RESET 高电平使 FDC 的所有操作流产,同时清 DOR.

+DRQ 2　(转接器输出,驱动门;三态 74 LS)
向或从主存传送一个数据字节准备就绪时 DRQ 2 线有效. DACK 2 有效或 I/O 读取 FDC 数据寄存器时 DRQ 2 被禁止输出

+IRQ 6　(转接器输出,驱动门,三态 74 LS)
FDC 完成一个动作后 IRQ 6 线被置成有效状态, 这个有效状态作为中断信号激活检查 FDC 结果字节、复位 IRQ 6、决定结束条件的子程序.

§ 23.4 A.B 驱动器接口

A、B 驱动器接口指转接器与 A、B 驱动器的接口. 接口信号电平均与 TTL 兼容, 其中 MPUL = 5.5 Vdc; LPUL = 2.4 Vdc; MPDL = 0.4 Vdc; LPDL = −0.5 Vdc. 接口的全部输出端均为集电极开路门,必须将它们连在通有 Vcc 的终端网络上 (马达允使 1 线除外, 它由 2 KΩ 电阻接至 Vcc). 接口的每个输入端通过 150 Ω电阻接至 Vcc.

1．接口输出端

-A、B 驱动器选择（驱动门 7438）

"A 或 B 驱动器选择"线无效时，A 或 B 驱动器断开各自的驱动门和接收门（马达允使线的驱动门、接收门不包在内）。

-A、B 马达允使线（驱动门 7438）

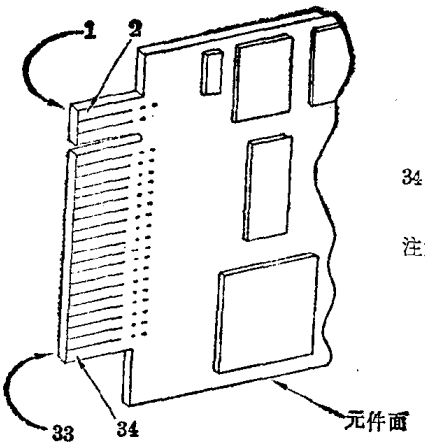
控制驱动器马达的启停。允使线上信号有效时马达开，否则关。

—步进移动（驱动门 7438）

步进移动线上的每个脉冲，使选中驱动器把读/写磁头向主轴或盘周移动一个磁道。

—方向（驱动门 7438）

决定磁头的移动方向，有效时向主轴移，无效时向盘周移。



34 脚边缘插上式插座

注意：引脚 1-33 在板的背面，
引脚 2-34（偶数）在板
的正面（元件面）

信号为 TTL 电平

IBM 5¼时软盘机	地（奇数引脚）	1-33	5¼时软盘机转接器
	未用	2.4.6	
	标引	8	
	马达A允使	10	
	驱动器B选择	12	
	驱动器A选择	14	
	马达允使B	16	
	方向（步进马达用）	18	
	步进脉冲	20	
	写入数据	22	
	写入允使	24	
	磁道0	26	
	写入保护	28	
	读出数据	30	
	磁头1选择	32	
	未用	34	

图 23-2 5—1/4 时软盘机转接器内部接口引脚安排

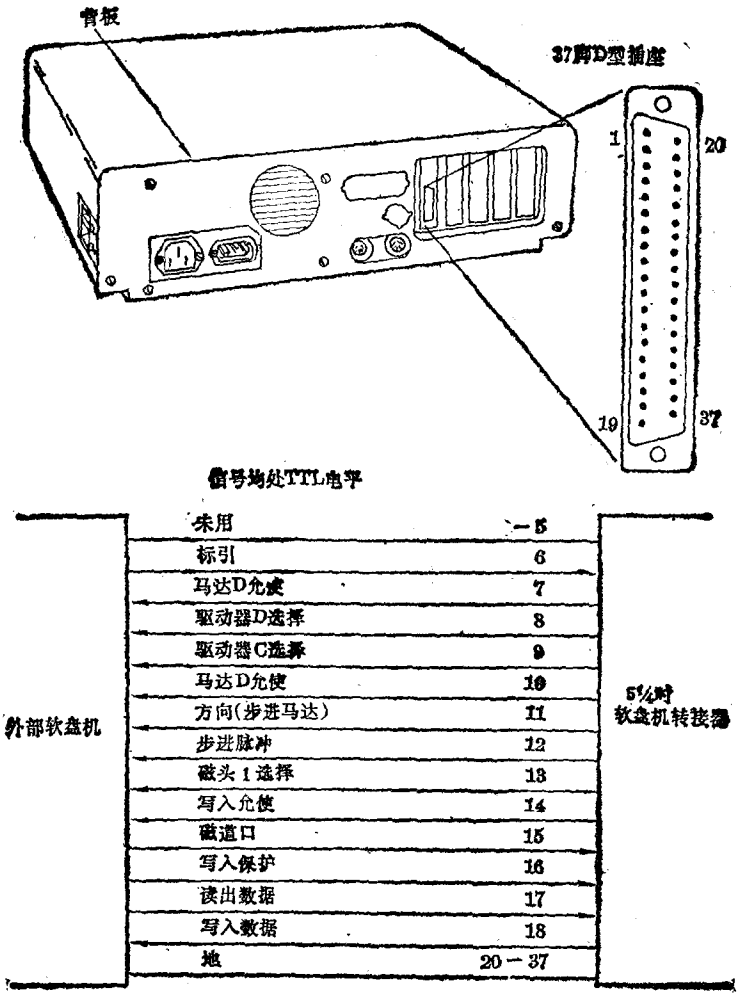


图 23-3 5—1/4 时软盘机转接器外部接口引脚安排

—写数（驱动门 7438）

"写数"线从高电平变低电平时，若写允使信号有效，则选中驱动器将磁通变化记录在软盘上。

—写允使（驱动门 7438）

禁止或允使磁头写电流

§23.5 5=1/4 时软盘驱动器

IBM 5-1/4 时软盘驱动器由主轴驱动、磁头定位及读/写/擦除三系统组成。主轴由带伺服装置的直流电机带动，4 相步进电机和相应的电路实现磁头的定位，磁盘每旋转一周磁头移动一个磁头的距离，工作时磁头接触磁盘表面。对写数据操作，磁头先在磁道上留下

0.33 mm 宽的磁迹，经擦除后留下 0.30 mm 的永久记录磁迹。读/写/擦除系统中的数据复原电路分成低电平读取放大器、微分器、过零检测器、数字转换器等若干部分，全部数据的译码工作均在转接器内完成。

驱动器内还有三个传感器，他们的作用是：

1．00 磁通开关：该开关检测磁头/滑架机构是否处在 00 磁道位置。

2．标引传感器：检测到磁盘上的标引孔时，该传感器点燃驱动器上的红色发光二极管。

3．写保护传感器：检测到插入的磁盘是写保护磁盘时，该传感器发出禁止写盘信号。

软盘机驱动器的机械、电气参数：

记录介质：工业兼容 5-1/4 吋软盘

磁道/吋：　48

磁道数：　(40)

尺寸：

高：　85.85 mm

长：　149.10 mm

宽：　203.2 mm

重：　2.04 kg

温度(不包括介质)

工作时：　10℃～44℃

不工作时：　－10℃～60℃

湿相对度(不包括介质)

工作时：　20%～80%

不工作时：　5%～95%

寻找时间：8 msec(磁道→磁道)

磁头建立时间：25 msec(已知上磁道的地址)

错误率：　10^{-9} (可恢复)

10^{-12} (不可恢复)

10^{-6} (查找)

磁头寿命：　20，000 小时(一般使用)

介质寿命：　3.6×10^{6} 遍/磁道

磁盘速度：　300 转/分±1.5%(长期)

瞬时速度变化：　±3.0%

启/停时间：　最大 500 毫秒

传送速率：　250 K 位/秒

记录方式：　MFM

功率：　±12 Vdc±0.6 V×900 mA

+5 Vdl±0.25 V×600 mA

第二十四章　扩展存贮器选件

§24.1　概　　述

通过系统扩展槽扩充的外部存贮器，应选择 32KB×9 或 64KB×9 存贮器片组。一经外部扩充，系统的 RAM 区容量便可超过 64 KB。扩展时应打开或关闭母板上的 SW 2-1～SW 2-4 开关，使他们指出正确的外接 RAM 容量。扩充 RAM 区仍属内存区，地址空间保持连续。外部扩充存贮器需要的动态刷新定时信号和地址信号由母板产生，通过 I/O 通道供出，其余的信号如总线缓冲、动态存贮器时序信号及地址多路转接电路、存贮体选择译码逻辑外部存贮器自备。

§24.2　运 行 特 点

系统板(母板)的工作主频是 4.77 MHZ，总线周期由四个时钟周期组成，所以扩展存贮器的读写周期也是四个时钟周期(840 ns)，外接 RAM 的访问时间一般等于 250 ns，周期 410 ns。

扩展存贮器的最小可装卸单元是存贮模块，它有两种规格：16 K×1 位和 32 K×1 位。每个模块要求三种电压 (+5 Vdc，−5 Vdc，+12 Vdc)，每 2 毫秒 128 个刷新周期。由于

表 24-1　存贮模块引脚说明

引　脚	16K×1 模块 (用于 32KB×8 选件)	32K×1 模块 (用于 64KB×8 选件)
1	−5V	−5V
2	Data In **	Data In **
3	—Write	—Write
4	—RAS	—RAS 0
5	A0	—RAS 1
6	A2	A0
7	A1	A2
8	+12 V	A1
9	+5V	+12V
10	A5	+5V
11	A4	A5
12	A3	A4
13	A6	A3
14	Data Out **	A6
15	−CAS	Data Ont **
16	GND	CAS 1
17	−*	—CAS 0
18	—*	GND

* 16K×1 模块只有 16 个引脚

** Data IN, Data Out 脚合用(三态总线)

32 KB 单位及 64 KB 单位的扩展存贮器是 9 位/字节的（1 位奇偶校验、8 位字节）所以对 32 KB 扩展存贮器需 18 个 16K×1 的存贮器模块，对 64 KB 扩展存贮器需要 18 个 32K×1 的存贮模块。

存贮模块的最大访问时间：RAS 250 ns、CAS 165 ns

每个扩展存贮器选件具有 8 个开关，用来设定该存贮器在整个系统存贮区中的起始地址，当 8 号开关合上时，A15 决定 64 KB 中的哪一个 32KB 部分可被访问到，8 号开关打开的话全部 64 KB 扩展内存都能访问到。

表 24-2　存贮选件始址-选件开关的关系

开　关　号	状　　态	
	开	关
1	A 19=0	A 19=1
2	A 18=0	A 18=1
3	A 17=0	A 17=1
4	A 16=0	A 16=1
5	A 15=0	A 15=1 *
6	未　用	
7	未　用	
8	只用于 64K 选件*	

* 利用 8 号开关可使 64 KB 选件中的 32 KB 接入线路

第二十五章　游戏控制转接器

§25.1　功能部件及工作原理

一、原理

游戏控制转接器最多可连接4只游戏器上的桨（paddle）或2只操纵杆（joystick）。转接器工作原理大致如下：转接器内有4套计时电路，他们的计时值是游戏器中随桨或操作杆位置变化而变化的电位器阻值的函数，CPU通过计时器完成计时需要的时间决定桨（操纵杆）的位置。游戏转接器也可作为带4个模拟阻性输入口和4个数字输入口的通用I/O转接器。使用时转接器插在I/O扩展槽上，游戏器控制电缆由转接器插头的背部引出。

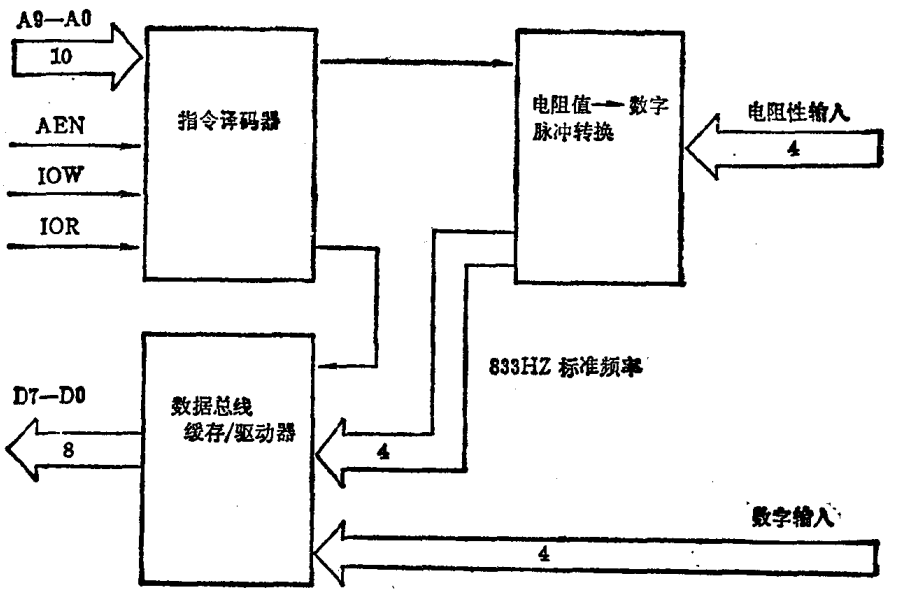


图25-1　游戏控制器转接器框图

二、转接器功能部件

1．地址译码器

转接器地址译码器由2块74 LS 138担任，转接口的有效I/O地址是X′201′，地址值有效时AEN线应处无效状态．转接器每选中一次，CPU或向游戏器发一次击发命令或读取一次击发的输出电平或扳机钮的值。

2．数据总线缓存器/驱动器

数据总线缓存器/驱动器由一块74 LS 244担任，当CPU发出IN指令（地址X′201′）时，74 LS 244驱动数据总线，其它时候缓存器为三态。

3．扳机钮

每个操纵杆/桨有一扳机钮，他的闭合状态可由INX′201′指令获得，按下扳机钮，数据

线上的某一位处'0'态，否则为'1'态。扳机钮不会自动反弹。

4．操纵杆位置

操纵杆的坐标位置由100 KΩ电位器之值决定，阻值不同使击发时间参数不同。击发动作由OUT指令（地址X'201'）触发，转接器的击发输出在触发脉冲到达后保持一段时间的高电平，这段时间由电位器中心抽头位置决定。击发输出由IN指令读入系统。

§25.2 接　口

接口有转接器I/O通道接口和转接器一游戏器接口之分。

一、I/O通道

下面是转接器占用的I/O总线的说明

A9-A0：	游戏控制器转接器地址线
D7-D0：	游戏控制器转接器数据线
IOR、IOW：	I/O读、写信号
AEN	有效时转接器、数据总线不起作用
+5V	转接器电源
GND	公共地线
A 19-A10	转接器未用
MENR、MEMW	
DACK0-DACK 3	
IRQ 7-IRQ 2	
DRQ 3-DRQ 1	
ALE、T/C	
CLK, OSC	
I/O CHCK	
I/O CH RDY	
HRQ I/O CH	
RESET DRV	
-5V, +12 V, -12 V	

二、转接器——游戏器接口

游戏器接口有8根输入线，4根为数字输入线，4根阻性输入线，他们的状态用IN指令送给CPU（地址X'201'）。

数字输入端通过1 KΩ上拉电阻接至+5V电源，这时CPU读到的是'1'。阻性输入端的输入值被转接器转换成脉冲信号，脉冲延迟时间正比于游戏器内的电位器且服从公式：

延迟时间(脉冲宽度) = 24.2 μs + 0.01 × 电位器值 μs

CPU用OUT指令（地址X'201'）触发电阻值至脉冲的转换，随后用IN指令得到阻性输入位的内容，4个阻性输位在OUT指令到来时刻同时变为高电平，在延迟了各自电位器决

定的时间后复位。

从 X'201' 口输入的数据字节格式：

7	6	5	4	3	2	1	0

数字输入位　　阻性输入位

连接操纵杆的方法：操纵杆两只(A 杆、B 杆)为一套，每只装有按钮一个，100KΩ 可变电阻两个，其中之一用于表示X坐标，另一用于表示Y坐标。操纵杆输入数据字节的格式是：

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B#2	B#1	A#2	A#1	B-Y	B-X	A-X	A-X
按　钮	按　钮	按　钮	按　钮	坐　标	坐　标	坐　标	坐　标

连接桨的方法：游戏桨两只(A 桨、B 桨)或(A.B.C.D 桨)为一套，每只桨装有按钮一个，100 KΩ 可变电阻一只。输入数据字节的变排是：

位#	7	6	5	4	3	2	1	0
	D钮	C 钮	B 钮	A 钮	D坐标	C 坐标	B 坐标	A 坐标

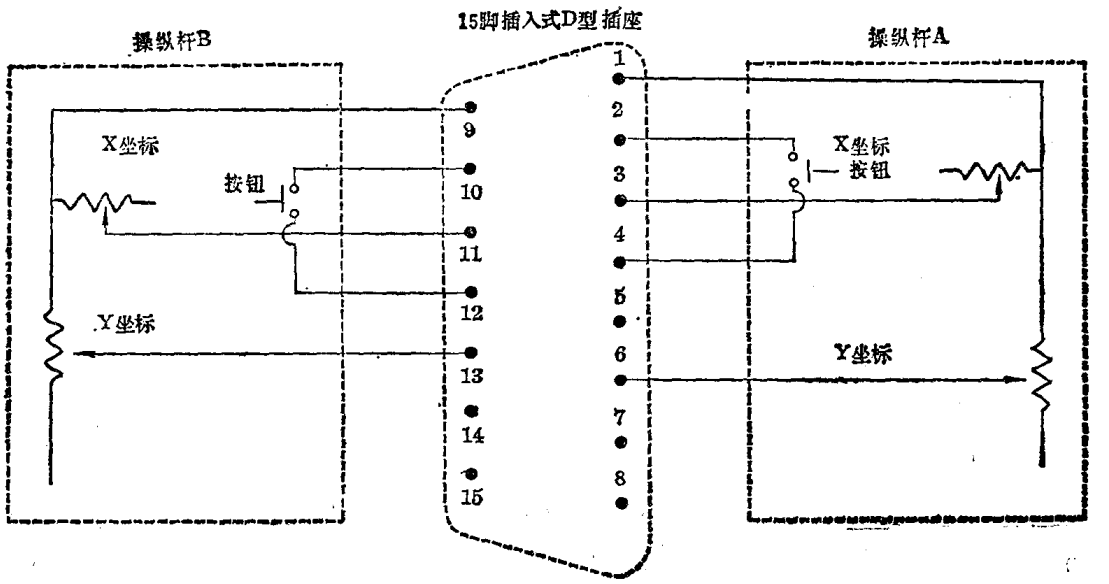


图 25-2　游戏器操纵杆连线图

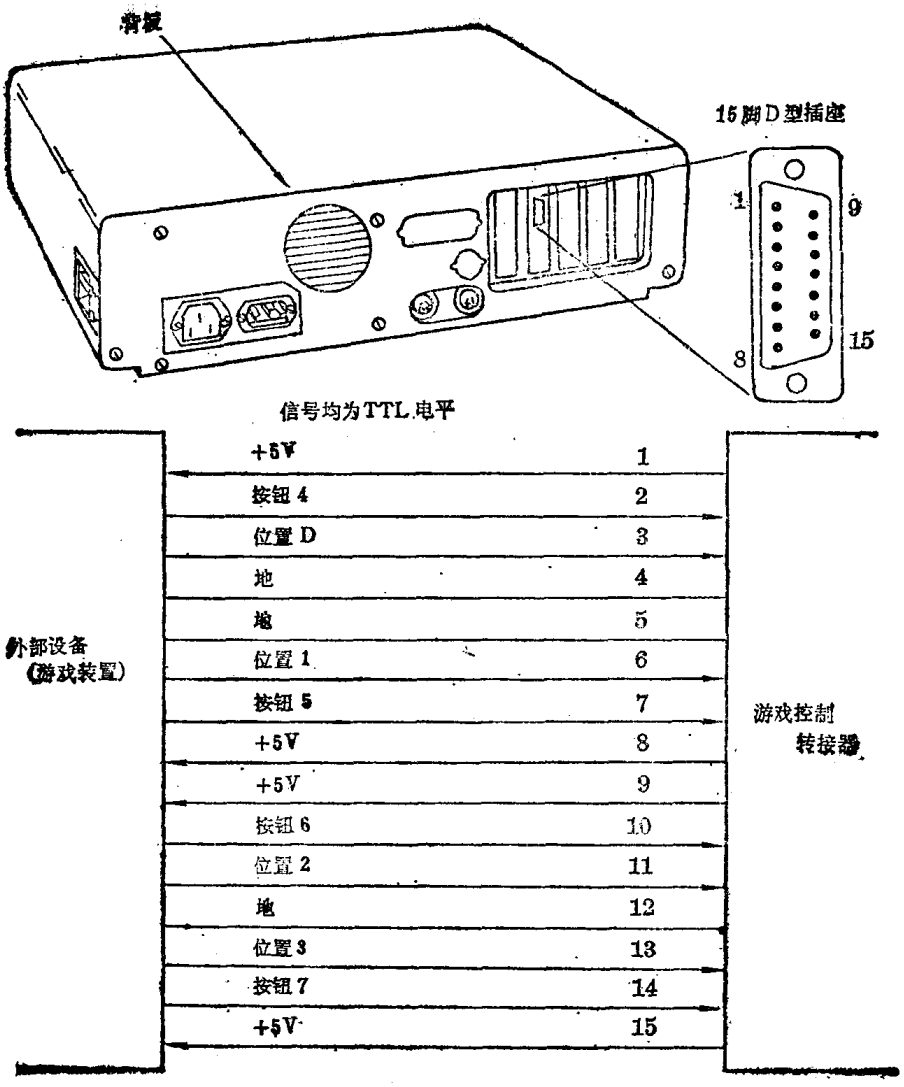


图 25-3 游戏控制器转接器(模拟量输入)插座接线说明

第二十六章　异步通讯转接器

§26.1　概　　述

异步通讯转接器(框图见图 26-1)是一个 4 吋×5 吋的插头，完全可程序，能为数据加上、除去启动位停止位和奇偶校验位。转接器的中断系统控制数据的发送和接收、错误报告、线路状态及数据装置的中断。转接器的诊断系统提供发送/接收输入输出信号的反循环功能。

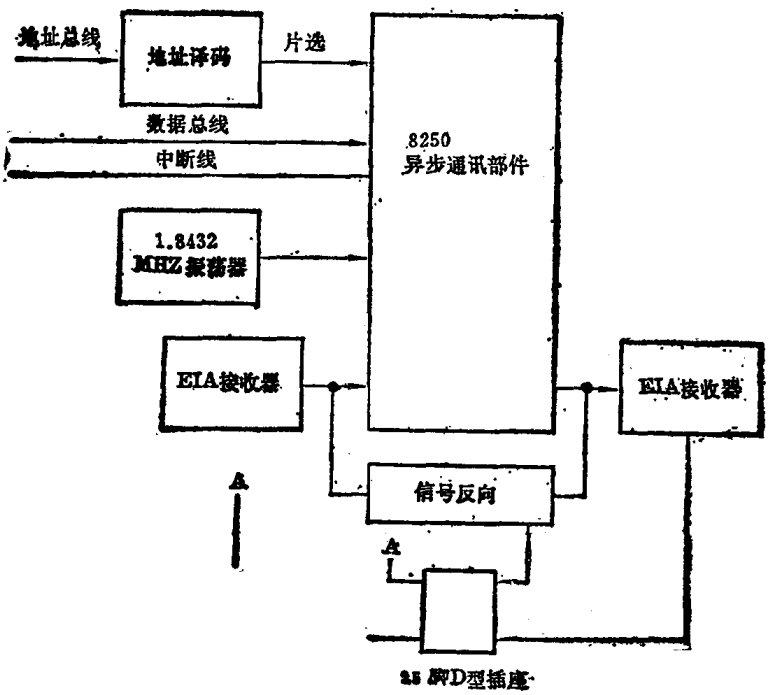


图 26--1　异步通讯转接器框图

§26.2　异步通讯控制器

Intel 公司的 INS 8250 异步通讯控制器是转接器的核心部件，其主要特点是：

1．为串行数据流加上或除去标准异步通讯位、启动位、停止位及奇偶校验位

2．完全双缓冲，无需精确同步

3．独立控制发送、接收或线路状态同数据装置的中断

4．可程序波特率发生器，对输入时钟信号进行 1 至 $2^{16}-1$ 的分频，产生内部 16×时钟信号

5．独立的接收时钟输入

6．清除后发(CTS)、请求发(RTS)、数据装置准备好(DSR)、数据结束准备好(DTR)、响

铃指示(RI)、载体检查等调制解调器功能

7．完全可程序的串行接口：

—5、6、7 或 8 位字符

—生成和检查偶奇/非偶奇位

—生成 1，1—1/2，2 位停止位

—波特率发生器(0～9600 波特)

8．检查启动位错误

9．记录所有状态信息

10．切断数据传输线及断线检查

11．内部诊断

—隔离通讯链错误、进行反循环

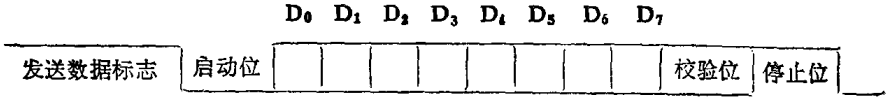
—模拟断线、奇偶校验、超时、成帧操作

12．带优先级的中断系统

通讯协议已用系统微码写成，转接器工作前应把它装入、接口和控制信号全部由系统软件负责处理。

§26.3 工作方式

通讯过程的具体工作方式通过编程 8250 设置，一般方法是把数据写入 I/O 地址为 3F8—3FF 定义工作方式的数个寄存器，行控寄存器的第 7 位(分频器门闩访问位 DLAB)也能选择这些寄存器。转接器连向 CPU 的中断线是 IRQ 4、MOdem 控制寄存器的第 3 位为 0 时，转接器通过中断线向处理器发中断信号。也只有在这时，中断允使寄存器允使中断请求信号变成真正的中断信号、发送器输出端和接收器输入端的数据格式如下：



数据字节的第 0 位先发送，先被接收。

表 26-1 8250 寄存器地址

I/O 地 址	对 应 寄 存 器	DLAB 状态
3F8	TX 缓存器	DLAB=0 (写)
3F8	RX 缓存器	DLAB=0 (读)
3F8	分频门闩最低位	DLAB=1
3F9	分频门闩最高位	DLAB=1
3F9	中断允使寄存器	DLAB=0
3FA	中断标识寄存器	
3FB	线路控制寄存器	
3FC	调制解调器控制器	
3FD	线路状态寄存器	
3FE	调制解调器状态寄存器	

表 26-2　各寄存器的地址位

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	DLAB	寄　存　器
1	1	1	1	1	1	1	×	×	×		
							0	0	0	0	接收缓存(读) 发送保持寄存器(写)
							0	0	1	0	中断允使 Reg
							0	1	0	×	中断标识 Reg
							0	1	1	×	线路控制
							1	0	0	×	Modem 控制
							1	0	1	×	线路状态
							1	1	0	×	Modem 状态
							1	1	1	×	无用
							0	0	0	1	分频门闩(LSB)
							0	0	1	1	分频门闩(MSB)

§26.4　接 口 说 明

转接器的接口信号安排大致与 EIA RS-232-C 的相同。实际使用时各信号线从 25 脚 D 型插座引出。转接器上还有电流返回接口，方便与 IBM 公司某类打印机的连接，此外还有一个选择电压接口或电流返环接口的手动开关。图 26-2 是电流返环接口图，其中 18 脚$^+$接收电流返环数据 (20 mA)，25 脚$^-$接收电流返环返回信号 (20 mA)，第四脚$^+$发送电流返环返回信号(20 mA)，第 11 脚$^-$发送电流返环数据(20 mA)。

电压接口是串行接口，它能支持的数据控制信号是：发送数据(引脚 2)、接收数据(引脚 3)、请求发送(引脚 4)、清除后发送(引脚 5)、数据装置准备好(引脚 6)、信号地(引脚 7)、载体检查(引脚 8)、数据终端准备好(引脚 20)、响铃指示(引脚 22)。

转接器把上述由通讯控制器产生的信号从 EIA(TTL) 电平转换成 TTL(EIA) 电平。系

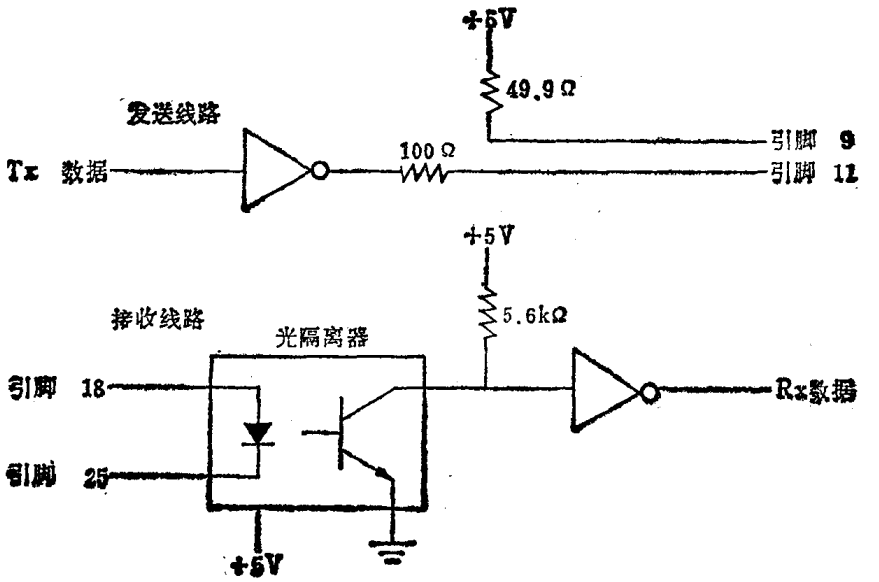


图 26-2　电流返接口

统软件利用这些电平信号决定接口或通讯外设的状态。电平的互换标准见表 26-3。发送数据时，“标准状态”表示 2 进制数据 “1”、空格状态表示二进制数据 “0”。

表 26-3 电平互换标准

互换电压 (V)	二进制状态	信号状态	接口控制
+3～15	0	空 格	ON
−3～−15	1	标 记	OFF

INS 8250 引脚功能(正逻辑)

输入信号引脚

1：选片(CS0、CS1、CS2)引脚 12～14；CS 0 = 1、CS = 1、CS 2 = 0 时 8250 被选中。有效地址打通信号 $\overline{ADS}$ 将选片信号锁存后，选片结束。

2．数据输入选通(DISTR，DISTR)引脚 21、22。DISTR 高或 DISTR 低且 8250 被选中时，CPU 可从 8250 读到数据或状态信息。因为一次读操作只要求 DISTR 有效或有输入电平，所以不读 8250 时应把两个 DISTR 脚置成低电平或高电平。

3．数据输出选通(DOSTR，$\overline{DOSTR}$)引脚 18、19。DOSTR 高或 DOSTR 低且 8250 被选中时，CPU 可把数据或命令字写入要求的寄存器。因为一次写操作只要求 DOSTR 有效或有输入电平，所以不写 8250 时把应把两个 DOSTR 都置成高电平或低电平。

4．地址选通($\overline{ADS}$)引脚 25。低电平时选择 A0、A1、A2 寄存器选择信号和 CS0.CS1.CS2 片选信号。若 A0、A1、A2 不能在读写过程中保持稳定电平的话，ADS 信号应有一个有效的输入电平。不用时将 $\overline{ADS}$ 接地。

5．寄存器选择(A0、A1、A2)引脚 26、27、28。A0、A1、A2 选中要求读出或写入字节的寄存器(见表 26-2)，DLAB 是线路控制寄存器的最高位，它能影响寄存器的选择。譬如在访问波特率发生器分频门闩时应把 DLAB 位置 1。

6．主机复位(MR)引脚 35。MR 高电平时，除接收缓存器、发送保持寄存器、分频门闩外。8250 所有寄存器和控制逻辑均被清 0，SOUT、INTRPT、OUT1、OUT2、RTS、DTR 输出信号也受影响。

7．接收器时钟(RCLK)引脚 9

接 16X 波特率接收器时钟信号。

8．串行输入(SIN)引脚 10

外设、Modem、数据装置等发来的串行数据的接收端。

9．清除后发($\overline{CTS}$)引脚 36

$\overline{CTS}$ (Modem 状态寄存器第 4 位) 是检查 Modem 控制器的一个输入脚。CPU 在读 Modem 状态寄存器时即可获得该引脚的状态电平，状态寄存器的第 0 位 DCTS 指出上次读取 Modem 状态寄存器后 CTS 位是否改变过状态，不管 $\overline{CTS}$ 何时改变状态，只要 Modem 状态中断未被禁止，中断信号总能发向 CPU。

10．数据装置准备好($\overline{CSR}$)引脚 37

DSR = 0 表示 Modem 或数据装置准备好接通线路、与 8250 进行数据传输了。DSR 的

状态可通过读取 Modem 状态寄存器获得(在寄存器的第 5 位)。Modem 状态寄存器的DDSR 位指出上次读取 Modem 状态寄存器后 DSR 位状态是否改变过。不管 DSR 何时改变状态，只要 Modem 状态中断不被禁止中断信号就能发向 CPU。

11．接收端线路信号检查(RLSD)引脚 38

RLSR = 0 表示 Modem 或数据装置已检查了数据载体。RLSD 的状态可通过读取 Modem 状态寄存器得知。上次读取 Modem 状态寄存器后，RLSD 脚状态改变与否由 DRLSD(状态寄存器的第 3 位)指出。不管 RLSD 位何时改变状态，只要 Modem 状态中断未被禁止，中断信号总能发向 CPU。

12．响铃指示器(RI)引脚 39

RI = 0 表示 Modem 或数据装置接收了电话响铃信号。读取 Modem 状态寄存器可知道 RI 的状态，在上次读取 Modem 状态寄存器后 RI 是否从低电平变成高电平，由状态寄存器的第 2 位 TERI 指出。不管 RI 位何时从高电平变成低电平，只要 Modem 状态中断未被禁止，中断信号总能发向 CPU。

13．Vcc 引脚 40

接 +5V dc。

14．Vsc 引脚 20

参考地(QV)。

输出信号

1．数据终端准备好($\overline{\text{DTR}}$)引脚 33

DTR = 0 通知 Modem 或数据装置：8250 已准备好可以通讯了。用指令将 Modem 控制寄存器第 0 位 (DTR) 置 1 时 DTR 输出引脚就变有效电平。主机复位时 DTR 被置 1。

2．请求发送($\overline{\text{RTS}}$)引脚 32

RTS = 0 通知 Modem 或数据装置：8250 已准备好，可通讯了。用指令将 Modem 控制寄存器第 1 位 (RTS) 置 1 时，RTS 输出引脚就变有效电平零电平。主机复位时 RTS 被置 1。

3．输出 1($\overline{\text{OUT1}}$)引脚 34

用户指定的输出端。用指令将 Modem 控制寄存器第 2 位 (OUT1) 置 1 时，OUT1 引脚变为有效电平(零电平)。主机复位时 OUT1 被置为 1。

4．输出 2($\overline{\text{OUT2}}$)引脚 31

用户指定的输出端。用指令将 Modem 控制寄存器第 3 位 (OUT 2) 置 1 时，OUT 2 输出引脚变成有效电平(零电平)。主机复位时 OUT2 被置 1。

5．选片输出(CSOUT)引脚 24

CSOUT = 1 时表示 8250 已被 CS0、CS1、CS2 信号选中，CSOUT 信号为逻辑 1 时数据传送才能开始。

6．禁止驱动器(DDIS)引脚 23

CPU 从 8250 读数时 DDIS 恒为 0，其他时候处逻辑 1 的 DDIS 信号，可用来禁止挂在 CPU、8250 间数据线上的内部收发器动作。

7．波特率输出($\overline{\text{BAUDOUT}}$)引脚 15

该信号为供8250发送器使用的16X时钟信号，其频率等于主参考晶体振荡频率被波特率发生器分频门闩分频后的值。BAUDOUT也可作为8250接收器的时钟信号，这时只要把它连在RCLK引脚上。

8．中断(INIRPT)引脚30

当下列类型的中断信号变成有效高电平且未被EIA禁止时，INIRPT=1；接收器错误标记、接收数据可用、发送器保持寄存器空、Modem状态，中断服务结束或主机复位后INIRPT=0。

9．串行输出(SOUT)引脚11

串行数据输出端，主机复位后SOUT被置成标记态(逻辑1)。

输入/输出信号

1．数据线(D0-D7)引脚1-8

供CPU与8250双向通讯用的引脚，传送控制字、状态以及数据。

2．外部时钟输入/输出(XTAL1、XTAL2)引脚16、17

8250连接晶体或信号时钟主时序参考信号的引脚。

§26.5 编程要点

一、异步通讯的复位(见表26-4)

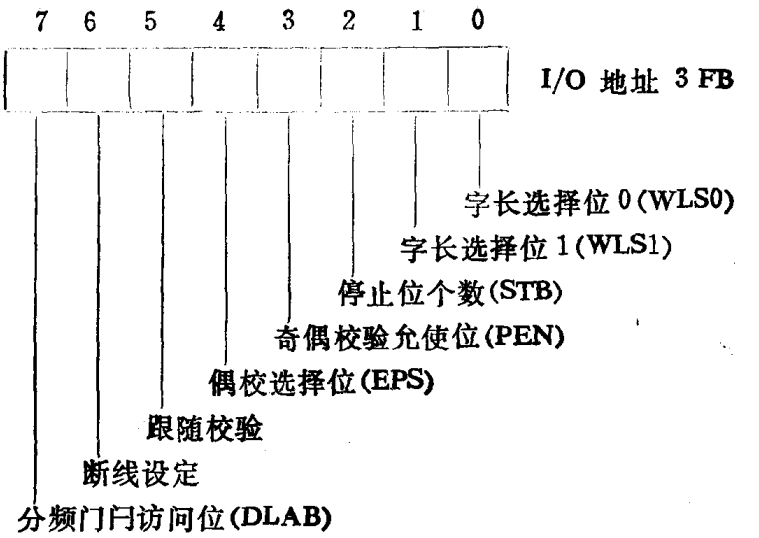
表26-4 异步通讯的复位

寄存器/信号	复位控制	复位状态
中断允使寄存器	主机复位	所有位均置成0，其中0-3为强迫0、4-7位永久0
中断标识寄存器	主机复位	0位置成1，1-2位置成0.3-7位永久0
线路控制寄存器	主机复位	所有位置成0
MODEM控制寄存器	主机复位	所有位置成0
线路状态寄存器	主机复位	仅位5、6被置成1
MODEM状态寄存器	主机复位	0-3位为0，4-7位为输入信号
SOUT	主机复位	1
INIRPT(接收器错)	读LSR/主机复位	0
INIRPT(接收器数据准备好)	读IIR写THR/主机复位	0
INIRPT MODEM状态变化	读MSR主机复位	0
OUT2	主机复位	1
RTS	主机复位	1
DTR	主机复位	1
OUT1	主机复位	1

二、INS 8250可访问寄存器值

1．INS 8850线路控制寄存器(LCR)

此寄存器用来记录用户要求的异步通讯格式，寄存器是可读，可写的。下面是LCR的格式



说明：

位 0.1：指定发送或接收串行字符的位数。

位 1	位 0	位数(位)
0	0	5
0	1	6
1	0	7
1	1	8

位 2：指定发送或接收串行字符的停止位位数。

位 2	由 0.1 位决定的字符位数	生成和检查的停止位位数(位)
0		1
1	5	1-1/2
1	6.7.8	2

位 3：第 3 位为 1 时，8250 产生(发送数据时)或检查(接收数据时) 夹在数据字最后一位与串行数据停止位间的奇偶校验位。

位 4：偶校选择位，它与位 3 连用

位 3	位 4	发送或检查数据位 + 奇偶位即累加和的奇偶性
1	0	奇
1	1	偶

位 5 跟随校验位

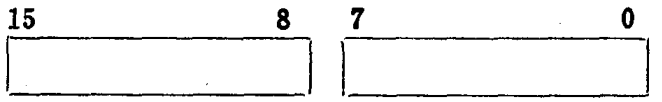
位 3	位 4	位 5	加了校验位后在接收端校验位的正常值
1	1	1	0
1	0	1	1

位 6：断线设定控制位　位 6 = 1 时，串行输出口被强制保持在空格 (逻辑 0) 状态并不受其他发送器的影响。 = 0 时禁止切断线路。

位 7. 分频门闩访问位(DLAB)　位 7 = 1 时允许在读/写操作中访问波特率发生器的分频门闩、接收器缓存器、发送器保持寄存器和中断允使寄存器。

2. 8250 可程序波特率发生器

该发生器接收 1.8432 MHZ 时钟输入信号，进行 1～2^{16}-1 的分频，发生器的输出频率＝16×波特率(波特率×16＝输入频率/分频值)。分频值在初始 8250 时装入它的两 8 个位门闩器，设置好分频门闩器的值后应立即装入 16 位的波特计算值。下面是分频门闩器的格式：



I/O 地址：3F9(DLAB＝1)　I/O 地址：3F8(DLAB＝1)

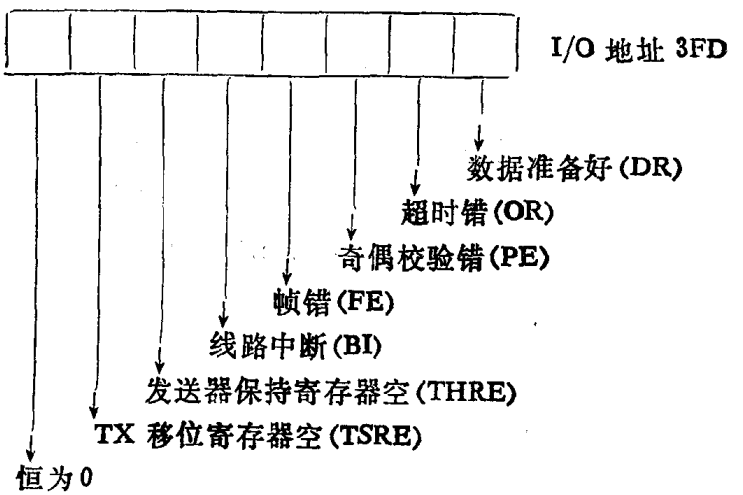
表 26-5 列出获得 15 种波特率需设置的分频值，其中 9600 是最大波特率值。由这些分频值产生成的波特率的误差是很小的。

表 26/5　(输入频率 1.843 MHZ)

要求的波特率	分频值		误差
	十进制	十六进制	
50	2304	'900'	--
75	1536	'600'	--
110	1047	'417'	0.026
134.5	857	'359'	0.058
150	768	'300'	--
300	384	'180'	--
600	192	'0C0'	--
1200	96	'060'	--
1800	64	'040'	--
2000	58	'03A'	0.69
2400	48	'030'	--
3600	32	'020'	--
4800	24	'018'	--
7200	16	'010'	--
9600	12	'00C	--

3．线路状态寄存器(LSR)

该寄存器为 CPU 提供与数据传送有关的状态信息，下面是 LSR 的格式：



说明：

位 0：接收器数据准备好指示位。位 0 = 0 时表示接收器已把整个数据字符接收了，并送至接收器缓存寄存器。CPU 读取接收器缓存寄存器或将逻辑 0 写入后该位复位。

位 1：超时错误指示位。位 1 = 1 时指出在 CPU 未把接收器缓存寄存器中的字符读走前又有新的字符被送入该寄存器，这时前字符被破坏。位 1 在 CPU 读线路状态寄存器后复位。

位 2：奇偶校验错指示位。为 1 时指出接收数据的校验和与偶校选择位指出的不同。该位在 CPU 读线路状态寄存器后复位。

位 3：帧错误指示位，指示接收字符的停止位无效。该位在检测出跟在数据最末位或校验位后的停止位为 0(空格态)时被置成 1。

位 4：线路断路中断指示位。若接收到的数据在大于整个字符传送时间内均为空格式(0)，即空格态时间 > 启动位 + 数据位 + 校验位 + 停止位传输时间，该位被置 0。

注意：

位 1-4 是错误标志位，一旦出现它们各自表示的错误 8250 就发出接收器线路状态中断。

位 5：发送器保持寄存器空指示位。位 5 表明 8250 是否准备好接收一个待发送的数据。当发送器保持寄存器空中断未被禁止的话(高电平)，该位促使 8250 发给 CPU 一个中断。数据从发送保持寄存器送入发送移位寄存器后，位 5 被置 1；数据被送入发送器保持寄存器时，位 5 被置 0。

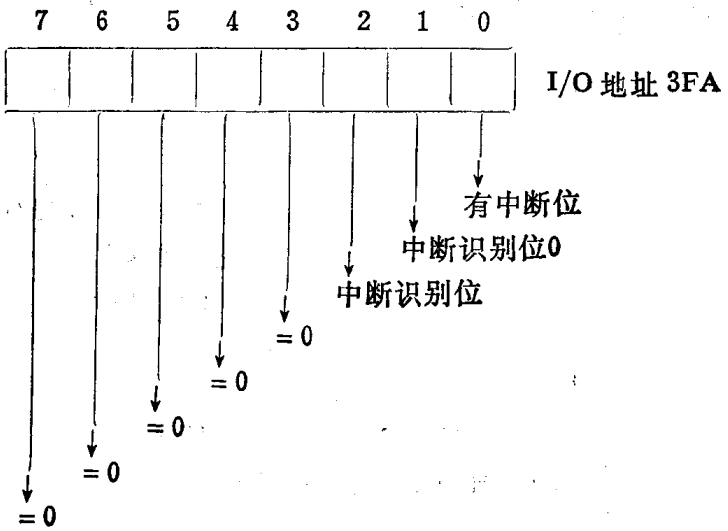
位 6：发送器移位寄存器空指示位(只能读)。位 6 = 1 时表示发送器移位寄存器空闲。数据发送器保持寄存器的内容被送入发送器移位寄存器后，该位马上复位。

位 7：总为 0。

4．中断识别寄存器(IIR)

为便于灵活地与现有的微处理机相连，减少软件在传输字符时的等待开销，8250 提供四级中断，它们是：优先级为 1 的接收器线路状态中断、优先级为 2 的接收数据准备好中断、优先级为 3 的发送器保持寄存器空中断及优先级为 4 的 Modem 状态中断。

IIR 记录中断及中断类型信息(见表 26-6)。在 8250 选片期间，IIR 允许具有最高优先



级的中断信号发向 CPU，其他中断则继续等待。下面是 IIR 的格式：

说明

位 0：在硬件分级或询问环境下指示中断是否出现。为 0 时表示有中断，这时的 IIR 内容可作为相应中断程序的指示器，为 1 时表示无中断，询问继续进行。

位 1-2：表示最高中断优先级，见表 26-6。

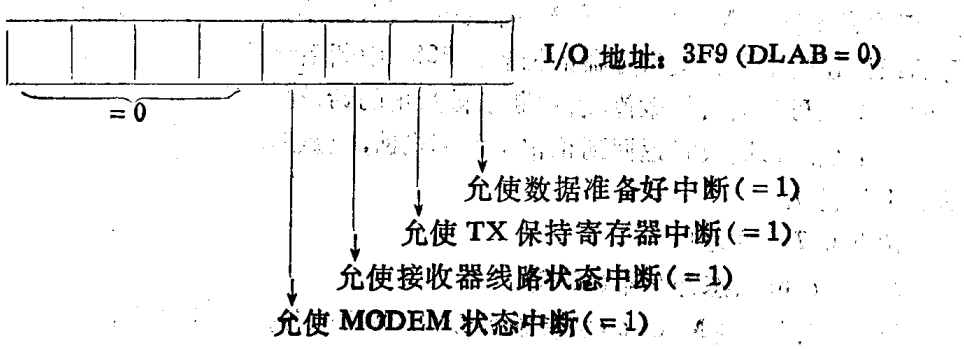
位 3-7：总为 0。

表 26-6 8250 中断类型表

中断标识寄存器			中 断 设 置 及 复 位			
Bit 2	Bit 1	Bit 0	优先级别	中 断 类 型	中 断 源	中断复位控制
0	0	1				
1	1	0	最高	接收器线路状态	超时错或 校验错或 帧错或 断线中断	读线路状态 寄存器
1	0	0	2	接收器数据好用	接收器数据好	读接收器缓存 寄存器
0	1	0	3	发送器保持 寄存器空	发送器保持 寄存器空	读 IIR(中断源) 或 写发送器保持 寄存器
0	0	0	4	MODEM 状态	清除后发 或 数据或装置准备好 或 响铃指示器 或 接收了线路检测 信号	读 MODEM 状态寄存器

5．中断允使寄存器(IER)

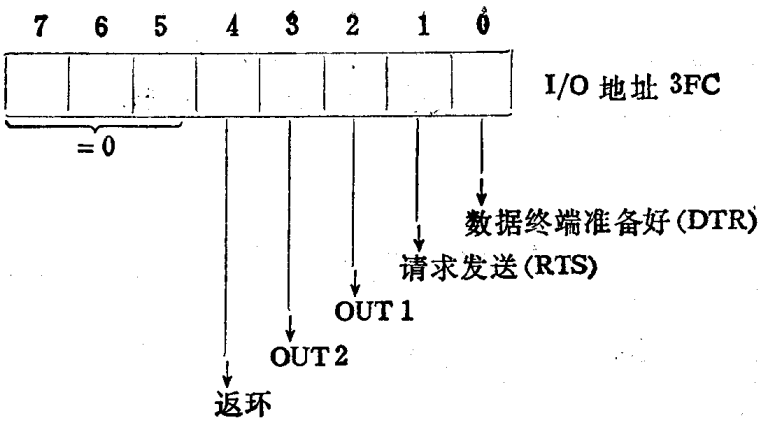
IER 使 82504 级中断信号能按优先级别逐一地通过 INTPRT 引脚发向 CPU。若把 IER 的 0-3 位全置 0，则任何中断均被禁止。IER 的某些位为 1，则相应的中断就不被禁止。下面是 IER 的格式：



6．MODEM 控制寄存器(MCR)

MCR 是个 8 位寄存器，控制与 Moder、数据装置或模拟 Modem 外设的接口。下面是

MCR 的格式。



说明

位 0：数据终端准备好(DTR)输出端控制位，他们的关系是：

位 0	DTR 输出端
1	0
0	1

DTR 可接至反相线路驱动器(如 DS1488)，使线路另一端的 Modem 或数据装置取得合适极性的电平输入。

位 1：请求发(RTS)输出端控制位，他们的关系同位 0 与 DTR 的关系

位 2：输出 1(OUT 1) 端 (用户指定辅助输出端) 控制位，他们间的关系同位 0 与 DTR 的关系。

位 3：输出 2(OUT2)端(用户指定辅助输出端)控制位，他们间的关系同位 0 与 DTR 的关系。

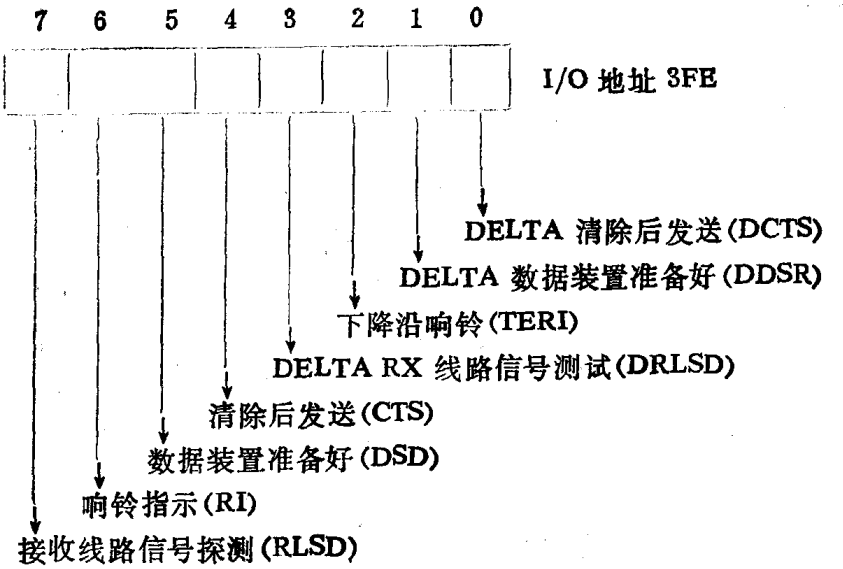
位 4：诊断测试 8250 的返环控制位，为 1 时发送器串行输出数据流被置成标记态 (逻辑 1)，接收器串行输入端开路，发送器移位寄存器的内容被送进接收移位寄存器的输入端，4 个 Modem 控制输入端(CTS、DSR、RLSD、RI)开路，改连至 4 个 Modem 控制输出端。即发送的数据立即被接收，借此检查 8250 的数据发送通路和数据接收通路。

在诊断模式下，8250 发送器及接收器的中断系统仍工作，Modem 中断控制系统也是如此，但中断源变成 Modem 控制寄存器的低 4 位，而不是 Modem 的几个控制输入端，中断信号的级别还受中断允使寄存器的控制。8250 中断系统的测试方法是：在 Modem 状态寄存器的低 4 位写入适当的数值，若中断未被禁止的话，写 1 表示发中断，这些中断位的复位操作还是不变。诊断模式返回到正常工作模式前，应重新编程 8250，并将 Modem 控制寄存器的低 4 位置 0。

位 5.6.7：总为 0。

7. Modem 状态寄存器(MSR)

MSR 记录连接 Modem (或其他外设)、CPU 的各控制线的当前状态以及控制线的变化情况。当来自 Modem 的控制信号改变状态时，MSR 反映控制线变化的低 4 位被置位，它们在 CPU 读 MSR 后复位。下面是 MSR 的格式。



说明：

位 0：DELTA 清除后发送指示位，指示在上次 CPU 读取后 CTS 输出端是否改变了状态。

位 1：DELTA 数据装置准备好指示位 (DDSR)，指示在上次 CPU 读取后 DSR 输出端是否改变了状态。

位 2：响铃指示器(TERI)下降沿探测位，指示 RI 输入端是否从 ON 变成 OFF。

位 3：DELTA 接收线信号探测器指示位，指示 RLSD 输入端是否改变过状态。

上面 4 位中任一位被置 1 后，8250 发 Modem 状态中断。

位 4：指示清除后发送输入端的反相情况。如果 MCR 的第 4 位为 1，本位等于 MCR 的 RTS 位。

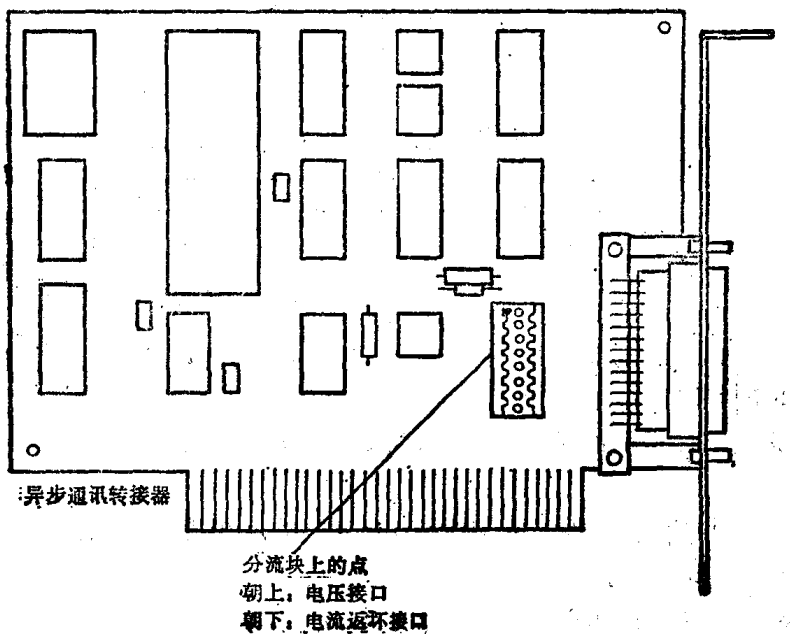
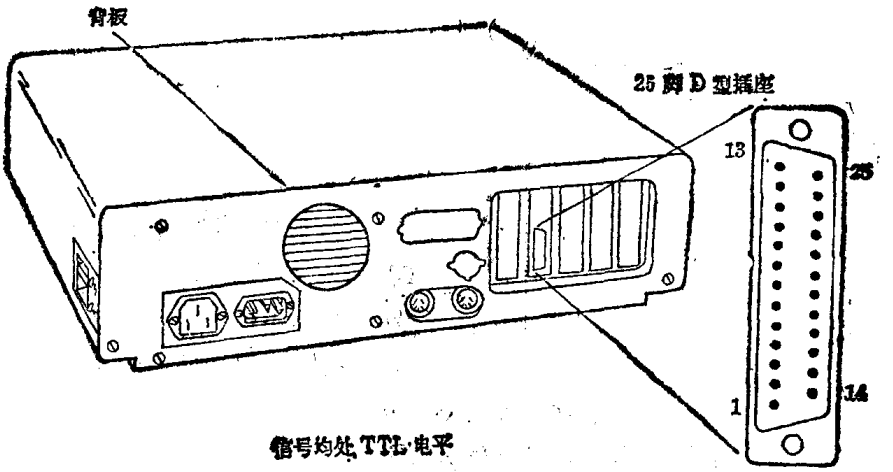


图 26-3 接口选择图



信号均处 TTL 电平

	NC	1	
	发送数据	2	
	接受数据	3	
	请求发送	4	
	清除后发送	5	
	数据装置准备好	6	
	信号地	7	
	载体检测	8	
	+	9	异步通讯转接器 (RS232C)
	NC	10	
外部设备		11	
	NC	12	
	NC	13	
	NC	14	
	NC	15	
	NC	16	
	NC	17	
		18	
	NC	19	
	数据终端准备好	20	
	NC	21	
	响铃指示	22	
	NC	23	
	NC	24	
		25	

图 26-4　异步通讯转接器插座引脚说明

位 5：指示数据装置准备好(DSR)输入端的反相情况。如果 MCR 的第 4 位等于 1，本位等于 MCR 的 DTR 位。

位 6：指出响铃指示器 (RI) 输入端的反相情况。如果 MCR 的第 4 位为 1，本位等于 MCR 的 OUT1 位。

位 7：指出接收线路信号检测(RLSD)输入端的反相情况，如果 MCR 的第 4 位为 1，本位等于 MCR 的 OUT2 位。

8．接收缓存寄存器(RBR)

RBR 保存接收到了的字符，其格式如下：

7	6	5	4	3	2	1	0	
								I/O 地址 3F8、DLAB=0、只读
D7	D6	D5	D4	D3	D2	D1	D0	

9．发送保持寄存器(THR)

THR 保存欲以串行方式发送出去的字符，其格式如下：

7	6	5	4	3	2	1	0	
								I/O 地址 3F8、DLAB=0、只写
D7	D6	D5	D4	D3	D2	D1	D0	

思 考 题

一、IBM PC 有何特点?

二、系统板有几大功能区，他们的作用是什么?

三、系统板内部接口能连接何种外设?

四、PC内部扬声器是怎样驱动的?

五、PC 的 I/O 通道是怎么回事?他的主要功能何在?不用 DMA 能读/写磁盘吗?

六、PC 采用几种电源，他们正常与否是怎样告知 8088CPU 的?

七、IBM 单色 CRT 转接器仅能适配 IBM 单色 CRT? 你认为 6845 有何特点，是否适合微机使用?

八、何为 A/N 式?何为 APA 式?他们的区别、联系在哪里?一个字符是怎样在 CRT 显示缓存中表示的? PC 有几种显示模式?

九、试叙述一个字符从数据总线送入 6845 直至显示出来的全过程。

十、试编一个为 6845 送 320×200 模式用工作参数的程序段(用 8086 指令)。

十一、打印机转接器是怎样工作的，怎样为它编程?

十二、打印机控制码有几类?试写一个输出一串空格的打印机控子程序。

十三、换码命令有何作用，怎样定页长、行长、行间距离?

十四、怎样跳行、跳列?

十五、磁盘驱动器转接器的主要部件是什么?一条磁盘机命令是怎样执行的?怎样编程 FDC?

十六、主状态寄存器与 ST0、ST1、ST2、ST3 有何区别?

十七、IBM PC 是怎样扩充其 RAM 区，扩充时用户应注意什么?做哪些工作?

十八、游戏器接口的工作原理是什么? CPU 如何知道扳机钮被按下的?

十九、异步通讯转接器中的 8250 有哪些可访问寄存器，作用是什么?

二十、电流返环的作用是什么，怎样实现?

第二十七章　IBM 硬盘驱动器转接器

§27.1　概　　述

IBM 硬盘驱动器转接器(见图 27-1)，是用来在 I/O 总线上适配硬盘驱动器的一种接口部件。它自带缓存器，利用系统板上的 DMA 传送记录的数据，控制器命令完成情况由中断形式通知主处理器，对传送数据进行 32 位 ECC 码检测，能自动检、校 11 位突发错误。每个硬盘转接器，最多能拖动两台硬盘机。硬盘驱动器转接器的设备级控由制均转接器上的 ROM 模块完成。

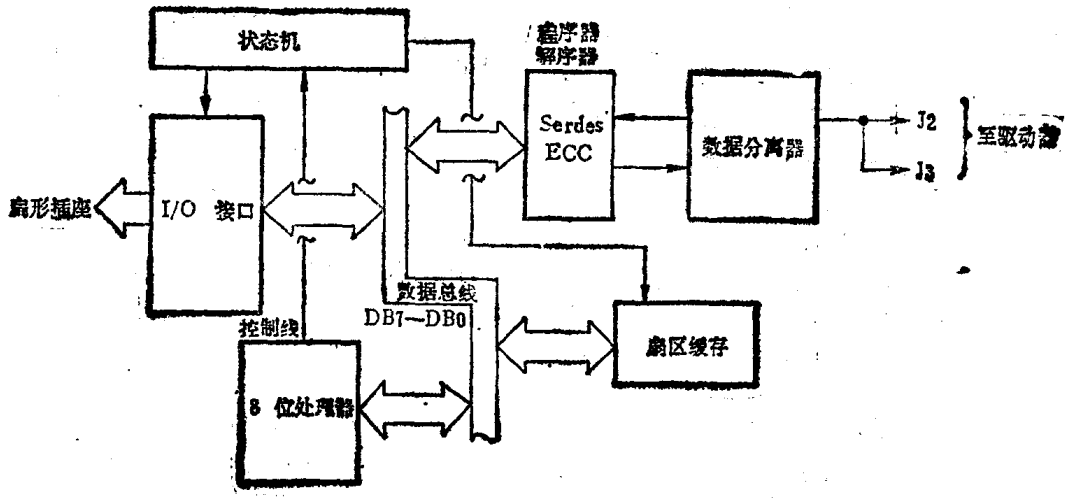


图 27-1　硬盘驱动器转接器框图

§27.2　硬盘控制器

硬盘控制器相对程序的界面，是两个可由主处理器访问的八位寄存器：一为状态寄存器，另一为数据寄存器。状态寄存器记录控制器的状态信息，只能读，主处理器通过它协调数据传输。数据寄存器实际上是通过数据总线可"看"到的寄存器栈的某一个寄存器，它可读、可写，用于记录输送的数据、命令、参数，提供具体的状态字节。控制器的选择译码由写 I/O 口 X'322' 产生。下面详细介绍这两个寄存器的用法及它们组合形成的控制器命令。

一、状态寄存器

系统板主处理器发出的任何一条硬盘驱动器命令执行完毕后，控制器都会返回一个表示命令是否正常结束的状态字节，其格式如下：

位*	7	6	5	4	3	2	1	0
	0	0	d	0	0	0	e	0

位 0.1.2.3.4.6.7：置 0

位 1：为 1 时说明命令执行有误

位 5：驱动器单元逻辑号

若中断未被禁止，控制器就在发给主处理器一个中断后，将上面的状态字送给系统板，这时的控制器忙信号不被确认。

二、检测字节

若状态寄存器收到发生错误的信号（其第 1 位置位了），硬盘控制器就会收到指明错误详细情况的 4 个检测字节，它们的安排是：

位*	7	6	5	4	3	2	1	0
0 字 节	地址有效		0		错误类型		错 误 码	
1 字 节	0	0	d	磁 头 号				
2 字 节	柱面号高部分			扇 区 号				
3 字 节	柱面号低部分							

d=驱动器

其中第 0 字节是错误类型字节，它的最高位在出错前一条是含地址命令时被置位，否则恒为 0。下表是第 0 字节中错误码，错误类型的各种组合。

表 27-1 错误一览表

	错误类型	错 误 码	描 述
位	5 4	3 2 1 0	
	0 0	0 0 0 0	控制器未查出前条命令有错
	0 0	0 0 0 1	控制器未探测出驱动器来的标引信号
	0 0	0 0 1 0	寻找操作全部非缓冲步进寻找完成后控制器未收到寻找完成信号
	0 0	0 0 1 1	最后操作中有写错误
	0 0	0 1 0 0	控制器选选驱动器，但驱动器不回送准备好信号
	0 0	0 1 0 1	未用
	0 0	0 1 1 0	跨跃过所有柱面后，控制器未收到驱动器 00 道信号

（续表）

位	错误类型 5 4	错误码 3 2 1 0	描述
	0 0	0 1 1 1	未用
	0 0	1 0 0 0	驱动器仍在寻找．本结果是测试驱动器准备好命令当驱动器未结束寻找，操作而为覆盖寻找条件返回的．控制器不对寻找操作计算超时时间．
	0 1	0 0 0 0	ID 读取错．控制器发现硬盘目标 ID 场有 ECC 错．
	0 1	0 0 0 1	数据错误：控制器在读操作中发现目标扇区有不可纠正的 ECC 错．
	0 1	0 0 1 0	地址标识错：控制器未测出硬盘上的目标地址标记(AM)．
	0 1	0 0 1 1	未用
	0 1	0 1 0 0	未找到扇面：控制器找到了正确的柱面、磁头，但找不到目标扇区．
	0 1	0 1 0 1	寻找错误：柱面或磁头地址未与予期目标地址相比较，不能作寻找结果．
	0 1	0 1 1 0	未用．
	0 1	0 1 1 1	未用．
	0 1	1 0 0 0	可纠数据错：控制器在目标场检出一个可纠 ECC 错．
	0 1	1 0 0 1	坏扇面：在最后操作中控制器查出坏扇区标记．发现本错后不得进行重试操作．

位	错误类型 5 4	错误码 3 2 1 0	描述
	1 0	0 0 0 0	非法命令
	1 0	0 0 0 1	非法硬盘地址：地址越出最大允许范围．

位	错误类型 5 4	错误码 3 2 1 0	描述
	1 1	0 0 0 0	RAM 错误：RAM 扇区—缓存诊断测试时发现有数据错误．
	1 1	0 0 0 1	程序存贮区检查和错：内部诊断时控制器发现程序存贮区检查和错．
	1 1	0 0 1 0	ECC 多项式错：控制器内部诊断测试时出现硬件 ECC 生成器失败错．

三、数据寄存器

硬盘驱动器的任何操作均是通过各类命令完成的。这些命令装配在一个由六字节组成的设备控制块 DCB 里，然后发向控制器。下面是 DCB 格式定义及说明。

位	7	6	5	4	3	2	1	0
字节 0	命令类			操作码				
字节 1	0	0	d	磁头号				
字节 2	柱面号高部			扇区号				
字节 3	柱面号低部							
字节 4	空白页或块计数							
字节 5	控制场							

字节 0：7—5 位指定命令类别，4—0 位为操作命令。

字节 1：第 5 位指定驱动器号，4—0 位为选择的硬盘磁头号。

字节 2：7—6 位为柱面号的最高、次高位、5—0 位为扇区号。

字节 3：7—0 位为柱面号低 8 位(共 10 位)。

字节 4：7—0 位指定空白页或块计数器。

字节 4：7—0 位控制场。

四、控制字节

DCB 第 5 字节是控制字节，起列选择硬盘驱动器操作类型的作用。该字节的格式如下：

位 7 6 5 4 3 2 1 0

r a 0 0 0 s s s

r = 重试位

s = 步长选择位

a = 出现数据 ECC 错误时重试选择位

说明：

位 7：置位时禁止任何访问硬盘命令失败后进行的 4 次重执过程。该位一般在测试驱动器时设立。

位 6：该位复位情况下，若在读命令执行中发现 ECC 错将再进行一次读操作。重读命令无错则结果状态不作错误标记。该位为 1 时不进行重读。

位 5，4，3：置 0，

位 2，1，0：决定驱动器类型及步长时间(见下表)。

位* 2	1	0	
0	0	0	驱动器未予指定，每个步长定为 3 毫秒。
0	0	1	N/A
0	1	0	N/A
0	1	1	N/A
1	0	0	每步 200 微秒

1 0 1　每步 70 微秒，由 BIOS 设定
1 1 0　每步 3 毫秒
1 1 1　每步 3 毫秒

5. 驱动器命令一览表

测试驱动器准备好
(0 类，操作码 00)

位	7	6	5	4	3	2	1	0
字节 0	0	0	0	0	0	0	0	0
字节 1	0	0	d	×	×	×	×	×

d=驱动器(0 或 1)
x=任意
字节 2.3.4.5.=任意

重校
(0 类、操作码 01)

位	7	6	5	4	3	2	1	0
字节 0	0	0	0	0	0	0	0	1
字节 1	0	0	d	×	×	×	×	×
字节 2	r	0	0	0	0	s	s	s

d、x 同上
r=重试
s=步长选择
字节 3.4.5=任意
ch=柱面号高部分

保留
(0 类，操作码 02)

02 操作码未用

请求检测状态
(0 类，操作码 03)

位	7	6	5	4	3	2	1	0
字节 0	0	0	0	0	0	0	1	1
字节 1	0	0	d	×	×	×	×	×

d，x 同操作码 01
字节 2.3.4.5=任意

驱动器格式化
(0 类，操作码 04)

位	7	6	5	4	3	2	1	0
字节 0	0	0	0	0	0	1	0	0
字节 1	0	0	d	磁头号				
字节 2	ch			0	0	0	0	0
字节 3	柱面低部							
字节 4	0	0	0	空白页				
字节 5	r	0	0	0	0	s	s	s

d、x 同上
r=重试
s=步长选择
ch=柱面号高部空白页：1～16
对 512 字节的扇面

准备好检验
(0 类，操作码 05)

位	7	6	5	4	3	2	1	0
字节 0	0	0	0	0	0	1	0	1
字节 1	0	0	d	磁头号				
字节 2	ch			扇面号				
字节 3	柱面号低部							
字节 4	块计数器							
字节 5	r	a	0	0	0	s	s	s

d，r，s，ch 同上
a=数据 ECC 的重试选择

（续表）

命　令	数据控制块	说　明
格式化磁道 （0类，操作码06）	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 0 1 1 0 字节1 0 0 d 磁头号 字节2 ch 0 0 0 0 0 0 字节3 柱面号低部 字节4 0 0 0 空白空 字节5 r 0 0 0 0 s s s	d,r,s,ch,a,空白页同上
格式化坏磁道 （0类，操作码07）	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 0 1 1 1 字节1 0 0 d 磁头号 字节2 ch 0 0 0 0 0 0 字节3 柱面号低部 字节4 0 0 0 空白页 字节5 r 0 0 0 0 s s s	d,r,s,ch 空白页同上
读 （0类，操作码08）	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 0 0 0 字节1 0 0 d 磁头号 字节2 ch 扇区号 字节3 柱面号低部 字节5 r a 0 0 0 s s s	d，r，s，ch 同上 a 同操作码05
保留 （0类，操作码09）		操作码09未用
写 （0类，操作码0A）	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 0 1 0 字节1 0 0 d 磁头号 字节2 ch 扇区号 字节3 柱面号低部 字节4 块计数器 字节5 r 0 0 0 0 s s s	d,r,s,ch 同操作码08

（续表）

命　令	数据控制块	说　明
寻找 (0类,操作码0B)	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 0 1 1 字节1 0 0 d 磁头号 字节2 ch 0 0 0 0 0 0 字节3 柱面号低部 字节4 × × × × × × × × 字节5 r 0 0 0 0 s s s	d,r,s,ch同上 x=任意
初始化驱动器特性* (0类,操作码0C)	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 1 0 0	字节1、2、3、4、5=任意
读ECC突发错长度 (0类,操作码0D)	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 1 0 1	字节1、2、3、4、5=任意
将数据从扇区缓存读出 (0类,操作码0E)	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 1 1 0	字节1、2、3、4、5=任意
将数据写入扇区缓存 (0类,操作码0F)	位 7 6 5 4 3 2 1 0 字节0 0 0 0 0 1 1 1 1	字节1、2、3、4、5=任意
RAM诊断 (7类,操作码00)	位 7 6 5 4 3 2 1 0 字节0 1 1 1 0 0 0 0 0	字节1、2、3、4、5=任意
保留 (7类,操作码01)		操作码01未用
保留 (7类,操作码02)		操作码02未用
驱动器诊断 (7类,操作码03)	位 7 6 5 4 3 2 1 0 字节0 1 1 1 0 0 0 1 1 字节1 0 0 d × × × × × 字节2 × × × × × × × × 字节3 × × × × × × × × 字节4 × × × × × × × × 字节5 r 0 0 0 0 s s s	d、r、s、x同0类,操作码01

（续表）

命　令	数据控制块								说　明
控制器内部诊断 (类 7，操作码 04)	位	7	6	5	4	3	2	1	0	字节 1、2、3、4、5=任意
	字节 0	1	1	1	0	0	1	0	0	
长读** (类 7，操作码 05)	位	7	6	5	4	3	2	1	0	d=驱动器(0 或 1) s=步长选择 r=重试 ch=柱面号高部
	字节 0	1	1	1	0	0	1	0	1	
	字节 1	0	0	d	磁头号					
	字节 2	ch			扇区号					
	字节 3	柱面号低部								
	字节 4	块计数器								
	字节 5	r	0	0	0	0	s	s	s	
长写*** (类 7，操作码 06)	位	7	6	5	4	3	2	1	0	d，s，r，ch 同上
	字节 0	1	1	1	0	0	1	1	0	
	字节 1	0	0	d	磁头号					
	字节 2	ch			扇区号					
	字节 3	柱面号低部								
	字节 4	块计数器								
	字节 5	r	0	0	0	0	s	s	s	

* DCB 后跟八个字节：

柱面数(2 字节)，磁头数(1 字节)，启始约减写当前柱面(2 字节)，启始写予补偿柱面(2 字节)，最大 ECC 数据突发长度(1 字节)。

** 每扇区返回 512+4ECC 数据(字节)。

*** 要求每扇区有 512+4ECC 数据(字节)。

六、编程要点

地址线低 2 位接于系统板 I/O 口译码器，该译码器分两个区：一个由 I/O 读信号(－IOR)选通，另一个由 I/O 写信号(－IOW)选通。所以共有四个 I/O 口子分配给硬盘控制器板。下面是这四个 I/O 口子的地址，作用。

R/W	口子地址	作　用
读	320	读数据(控制器至系统板)
写	320	写数据(系统板至控制器)
读	321	读控制器硬件状态
写	321	控制器复位
读	322	保留
写	322	产生控制器选择脉冲
读	323	未用
写	323	将位模式写入 DMA 及中断屏蔽寄存器

七、系统 I/O 通道接口信号

D A0-A 19：20 位正逻辑地址信号，低 10 位含有读写命令地址（I/O 范围 320 H～323 H）。20 位地址信号经译码访问转接器 8000 H～C 9 FFFH ROM 空间。

2) D0-D 7：数据信号，传送数据、状态。

3) $\overline{\text{IOK}}$：I/O 读，系统板通过程序或 DMA 读控制器状态、数据时有效。

4) $\overline{\text{IOW}}$：I/O 写，系统板通过程序或 DMA 将命令、数据写给控制器时有效。

5) AEN：地址有效信号。系统板 DMA 产生 $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ 信号并控制了地址总线、数据总线时有效。

6) RESET：复位信号，强迫硬盘控制器恢复至接通电源时的状态。

7) IRQS：主处理器确认的中断请求信号。控制器利用其在发回状态字节前中断系统板。

8) DRQ 3：转接器确认的 DMA 请求信号。DMA 控制下的数据可以发向系统板或从系统板接收时，控制器置其成有效态。DRQ 3 在系统板 DMA 应答信号（－DACK 3）有效前总为高电平。

8) $\overline{\text{DACK 3}}$：DMA 响应信号。低电平时表示系统板响应控制器的 DMA 请求。

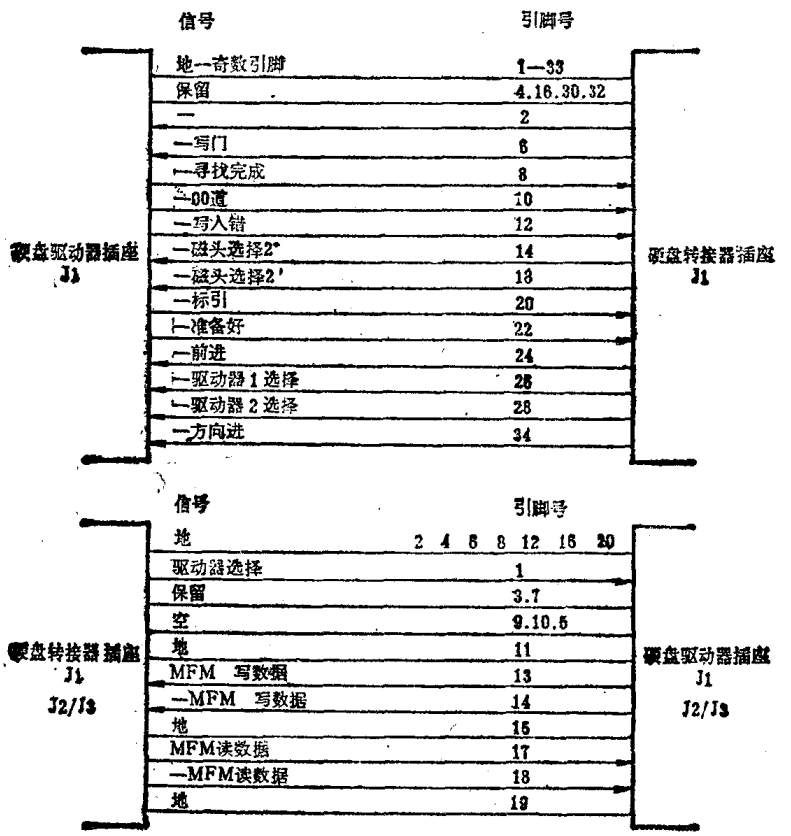


图 27-2 硬盘转接器接口说明

八、IBM 10 兆硬盘驱动器

IBM 10 兆硬盘驱动器为随机访问外设，它有两个不可移动式 $5^1/_4$ 吋存贮盘面。每个盘

面306柱面(道)，由一个活动磁头访问。硬盘机格式化后的总容量为10兆字节(17扇面/道,512字节/扇面,1224道/驱动器)。

表27-2 IBM 10兆硬盘驱动器机械、电气参数

介质	硬质盘	工作	10 Gs
磁道数	1224	不工作	20 Gs
道密度	345道/吋	访问时间	道一道间3毫秒
尺寸		平均潜在时间	8.33毫秒
高	82.55 mm	错误率	
宽	146.05 mm	软读错误	10^{-10}
深	203.2 mm	硬读错误	10^{-12}
重量	2.08公斤	寻找错误	10^{-6}
温度		设计寿命	5年(8000小时MTF)
工作时	4℃～50℃	硬盘速度	3600转/分 ±1%
不工作时	−40℃～60℃	数传率	5.0兆位/秒
相对湿度		记录方式	MFM
工作时	8%—80%(无冷凝)	电源	+12 Vdc±5% 1.8A(最大4.5A)
最大湿球	26℃		+5 Vdc±5% 0.7 A(最大1.0 A)
振动		最大波动率	等阻负载时1%

九、硬盘驱动器ROM BIOS列表文件

```
NE  SOURCE
 1  TITLE(FIXED DISK BIOS FOR IBM DISK CONTROLLER)
 2
 3    INT 13
 4  ;
 5  ; FIXED DISK I/O INTERFACE  硬盘I/O接口
 6  ;
 7  ;          THIS INTFREACE PROVIDES ACCESS TO 5 1/4 FIXED DISKS
 8  ;          THROUGH THE IBM FIXED DISK CONTROLLER
 9  ;
10  ;
11
12  ;
13  ;          THE BIOS ROUTIHES ARE MEANT TO BE ACCESSED THROUGH
14  ;          SOF TWARE INTERRUPTS ONLY ANY ADDRESSES PRESENT IN
15  ;          THE LISTIHGS ARE IHCLUDED ONLY FOR COMPLETENESS
16  ;          HOT FOR REFERENCE. APPLICATIONS WHICH REFERENCE
17  ;          ABSOLUTE ADDRESSES WITHIN THE CODE SEGMEHT
18  ;          VIOLATE THE STRUCTURE AHD DESIGN OF BIOS.
19  ;
20  ;
21  ; INPUT   (AH=HEX VALUE)  入口参数：((AH)为十六进制数)
```

```
22  ;
23  ;   (AH) = 00 RESET DISK (DL = 80H, 81H)/DISKETTE  (AH) = 0：复位硬盘
        (DL = 80 H, 81 H)/软盘驱动器
24  ;   (AH) = 01 READ THE STATUS OF THE LAST DISK OPERATION INTO
        (AL)  (AH) = 1：上次操作状态读进(AL)
25  ;           NOTE：DL < 80 H—DISKETTE  DL < 80 H 读软盘驱动器
26  ;                 DL > 80 H—DISK  DL > 80 H 读硬盘驱动器
27  ;   (AH) = 02 READ THE DESIRED SECTORS INTO MEMORY  (AH) = 2：读
        扇区
28  ;   (AH) = 03 WRITE THE DESIRED SECTORS FROM MEMORY  (AH) = 3：写
        扇区
29  ;   (AH) = 04 VERIFY THE DESIRED SECTORS  (AH) = 4：校验扇区
30  ;   (AH) = 05 FORMAT THE DESIRED TRACK  (AH) = 5：格式磁道
31  ;   (AH) = 06 FORMAT THE DESIRED TRACK ANO SET BAD SECTOR FLAGS
        (AH) = 6：格式磁道并标记出坏了的磁道
32  ;   (AH) = 07 FORMAT THE DRIVE STARTING AT THE DESIRED TRACK
        (AH) = 7：从指定的磁道开始格式化驱动器
33  ;   (AH) = 08 RETURN THE CURRENT DRIVE PARAMETERS  (AH) = 8：返
        回当前的驱动器参数
34  ;
35  ;   (AH) = 09 INITIALIZE DRIVE PAIR CHARACTERISTICS
36  ;           INTERRUPT 41 POINTS TO DATA BLOCK  (AH) = 9：初始驱动
        器特征，中断 41 指向数据块(每扇区读 512 + 4ECC 字节)
37  ;   (AH) = 0A READ LONG  (AH) = 0A：长读
38  ;   (AH) = 0B WRITE LONG  (AH) = 0B：长写
39  ;   NOTE：READ AND WRITE LONG ENCOMPASS 512 + 4BYTES ECC (每扇
        区写 512 + 4ECC 字节)
40  ;   (AH) = 0C SEEK  (AH) = 0C：寻找
41  ;   (AH) = 0D ALTERNATE DISK RESET (SEE DL)  (AH) = 0D：可选择驱动
        器的格式化(见 DL 参数的含义)
42  ;   (AH) = 0E READ SECTOR BUFFER  (AH) = 0E：读扇区的缓存
43  ;   (AH) = 0F WRITE SECTOR BUFFER
44  ;           (RECOMMENOED PRACTICE BEFORE FORMTTING)  (AH) =
        0F：写扇区缓存
45  ;   (AH) = 10 TEST ORIVE REAOY  (AH) = 10：测试驱动器是否准备好
46  ;   (AH) = 11 RECALIBRATE  (AH) = 11：重校
47  ;   (AH) = 12 CONTROLLER RAM DIAGNOSTIC  (AH) = 12：控制器 RAM 诊断
48  ;   (AH) = 13 DRIVE DIAGNOSTIC  (AH) = 13：驱动器诊断
49  ;   (AH) = 14 CONTROLLER INTERNAL DIAGNOSTIC  (AH) = 14：控制器内部
```

```
50  ;
51  ;       REGISTERS USED FOR FIXED DISK OPERATIONS  其余参数寄存器
52  ;
53  ;       (DL)   —DRIVE NUMBER (SOH-87H FOR DISX, VALUE CHECKED)
                (DL)—驱动器号(80—87 H 从指定硬盘机,进行值检查)
54  ;       (DH)   —HEAD NUMBER (0-7 ALLOWED, NOT VALUE CHECKED)
                (DH)—磁头号(0—7,不进行值检查)
55  ;       (CH)   —CYLINOER NUMBER (0-1023, NOT VALUE CHECKED) (SEE
                CL  (CH)—柱面号(0—1023,不进行值检查,见参数 CL)
56  ;       (CL)   —SECTOR NUMBER (1—17, NOT VALUE CHECKED)  (CL)—
                扇面号(1—17,不进行值检查)
57  ;
58  ;               NOTE: HIGH 2 BITS OF, CYLINDER NUMBER ARE PLACED
59  ;                     IN THE HIGH 2 BITS OF THE CL REGISTER
60  ;                     (10 BITS TOTAL)  注意:柱面号共有 10 位,其最高 2 位
                          放在 CL 寄存器的高 2 位
61  ;       (AL)   —NUMBER OF SECTORS (MAXIMUM  POSSIBLE  RANGE 1—
62  ;                                   80H,  FOR READ/WRITE LONG 1—
                                        79H)  (AL)—扇区数 (最大值 1～80
                                          H, 长读/长写为 1—79 H)
63  ;               (INTERLEAVE VALUE  FOR FORMAT  1—160)  格式化时为
                    1—160
64  ;       (ES:BX)—ADDRESS OF BUFFER FOR READS AND WRITES,
65  ;               (HOT REOUIRED FOR VERIFY)  (ES: BX)—读写缓存地址
                    (校验时无用)
66  ;
67  ; OUTPUT  出口参数:
68  ;   AH = STATUS OF CURRENT OPERATION
69  ;        STATUS BITS ARE DEFINED IH THE EGUATES BELOH  (AH) = 当
              前操作的状态,各位的定义见下面等式
70  ;   CY = 0 SUCCESSFUL OPERATION(AH = 0 ON RETURN) CY = 0 成功(AH = 0)
71  ;   CY = 1 FAILED OPERATION(AH HAS ERROR REASON)  CY = 1 失败 (AH
          内容为错误原因)
72  ;
73  ; NOTE: ERROR IIH INOICATES THAT THE DATA READ HAD A RECOV-
              ERABLE
74  ;   ERROR WHICH WAS CORRECTED BY THE ECC ALGORITHM THE DATA
75  ;   IS PROBABLY GOOD, HOWEVER THE BIOS ROUTINE INDICATES AN
```

```
76  ;   ERROR TO ALLOH THE CONTROLLING PROGRAM A CHANCE TO
        DECIED
77  ;   FOR ITSELF THE ERROR MAY NOT RECUR IF THE DATA IS
        LOC OBJ LINE SOURCE
78  ;   REWRITTEN IALI CONTAINS THE BURST LENGTH 注意：第11号
        是数据发生可修复错误，它已由ECC算法修改过来了，BIOS返回这个错
        误号让硬盘BIOS调用程序决定下一步该怎么做。重写数据可避免这种
        错误再次出现。(AL)为突发错误的长度
79  ;
80  ;   IF DRIVE PARAMETERS WERE REQUESTED
81  ;
82  ;   DL=NUMBER OF CONSECUVIVE ACKNOWLEDGING DRIVES
        ATTACHED(0—2)
83  ;       (COHTROLLER CARO ZERO TALLY ONLY) DL=连接的连续
            可识别驱动器数(0—2)(适合0标记控制器卡)
84  ;   OH=MAXIMUH USEABLE VALUE FOR NEAD NUMBER DH=磁
        头数最大可用值
85  ;   CH=MAXIMUM USEABLE VALUE FOR CYLINDER NUMBLR CH=
        柱面数最大可用值
86  ;   CL=MAXIMUM USEABLE VALUE FOR SECTOR NUHBER
87  ;       AHO CYLIHOER NUHBER MIGH BITS CL=扇区数最大可用值
            及柱面号高位部分。
88  ;
89  ;   REGISTERS MILL BE MESERVED EXCEPT WHEN THEY ARE
        USED TO RETURH
90  ;   ITFONHAYION 注意：出现错误后较好的办法是复位驱动器，再试一次
91  ;
92  ;   NOTE: IF AN ERROR IS DEPORIED BI THE DISK COOE, THE
        APPROPRIATE
93  ;           ACTIOH IS TO RESET THE OISX, THEN RETRY THE
                OPERAYIOH. 常数定义：
94  ;
95  ; ..........................................................
96
90FF 97  SENSE_FAIL        EQU  OFFH; SENSE OPERATION FAILED
0089 98  UHDEF_ERR         EQU  088H; UDEFINED ERROR OCCUR-
                                      RED
0000 99  TIHE_OUT          EQU  80H ; ATTACHMENT FAILED TO
                                      RESPOHO
0040 100 BAD_SEEK          EQU  40H ; SEEK OPERATION FAILED
```

```
0020          101  BAD_CMTLR          EQU  20H ; COHTRLLER MAS FAILED
0011          102  DATA_CORRECTED     EQU  11H ; ECC CORRECTED DATA
                                                 ERROR
0010          103  BAD_ECC            EQU  10H ; BAD ECC ON DISK READ
0008          104  BAD_TRACK          EQU  08H ; BADTRACKFLAGDETECTED
0009          105  DMA_BOUHDARY       EQU  09H ; ATIEMPT TO DMA ACROSS
                                                 64 K BOUNOARY
0007          106  INIT_FAIL          EQU  07H ; DRIVE PAPAIIETER ACTIV-
                                                 ITY FAILED
0005          107  BAD_RESET          EQU  05H ; RESET FAILED
0004          108  RECORO_NOT_FHD     EQU  04H ; REOUESTED SECTOR NOT
                                                 FOUMO
0002          109  BAD_ADOR_MARK      EQU  02H ; ADDRESS MARK MOT FOUHO
0001          110  BAD_CHD            EQU  01H ; BAD COMMAHO PASSED TO
                                                 DISK I/O
              111
              112  ; ..................................................
              113  ;      IMTERRUPT AHO SIATUS AREAS;  中断向量及状态字节区,
              114  ; ..................................................
              115
              116  DUMMY SEGMENT AT 0
0034          117          ORG     0DH*4          ; FIXED DISK IMTERR-
                                                    UPT VECIOR 硬盘机
                                                    中断向量
0034          118  HDISK_INT       LABEL   DWORD
006C          119          ORG     13H*4          ; DISK INTERRUPT VE-
                                                    CTOR 磁盘机中断向量
004C          120  ORG_VECTOR      LABEL   DWORD
0044          121          ORG     19H*4          ; BOOTSTRAP IHTERR-
                                                    UPT VECTOR 自启动
                                                    中断向量
0064          122  BOOT_VEC        LABEL   DWORD
0078          123          ORG     LEH*4          ; DISKETTE PARAME-
                                                    TERS 软盘机参数
0078          124  DISKETTE_PARM   LABEL   DWORD
0100          125          ORG     040H*4         ; NEW DISKETTE INT-
                                                    ERRUPT VECTOR 新
                                                    软盘机中断向量
0100          126  DISK_VECTOR     LABEL   DWORD
```

```
0104          127              ORG     041H*4                  ; FIXED DISK PARAM-
                                                                 ETER VECTOR 硬盘机
                                                                 参数区中断向量
0104          128  HF_TBL_VEC LABEL   DWORD
7C09          129              ORG     7C00H                   ; BOOTSTRAP LOADER
                                                                 VECTOR 自启动装入程
                                                                 序中断向量
7C00          130  BOOT_LOCH  LABEL   FAR
......        131  DUMMY       ENOS
              132
......        133  DATA        SEGMENT AT 40H
0042          134              ORG     42H
0042          135  CMO_BLOCK   LABEL BYTE
0042 (7??)    136  HD_ERROR    DB      7DUP(?)                 ; OVERLAYS DISKETTE
                                                                 STATUS 硬盘机命令块用
                                                                 字节区/检测字节区
006C          137              ORG     06CH
006C ????     138  TIMER_LOW   DN      ?                       ; TIMER LOW WORD 计时
                                                                 器低字
0072          139              ORG     72H
0072 ????     140  RESET_FLAG  DN      ?                       ; 1234H IF KEYBOARO
                                                                 RESET UNOERWAY 复
                                                                 位状态，键盘复位时等于
                                                                 1234 H
0074          141              ORG     74H
0074 ??       142  DISK_STATUS, DB      ?                       ; FIXED DISK STATUS
                                                                 BYTE 硬盘机状态字节
0075 ??       143  HF_NUH      DB      ?                       ; COUNT OF FIXED DISK
                                                                 DRIVES 硬盘机个数计数
                                                                 器
0076 ??       144  CONTROL_BYTE DB     ?                       ; CONTROL BYTE DRIVE
                                                                 OPTIONS 驱动器选择控
                                                                 制字节
0077 ??       145  PORT_OFF    DB      ?                       ; PORT OFFSET 口子的偏
                                                                 移
......        146  DATA ENDS
              147
......        148  CODE SEGMENT
              149
```

```
150 ; ..........
151 ; HAROWARE SPECIFIC VALUES  硬件特定值:
152 ;
153 ; __CONTROLLER I/O PORT  控制器I/O口内容:
154 ; >WHEN READ FROM:  (读出时)
155 ; HF__PORT+0-READ DATA (FROM CONTROLLER TO CPU; HF__
      PORT+0  读数据(控制器至CPU)
156 ; HF__PORT+1-READ CONTROLLER HARDWARE STATUS
157 ;                (CONTROLLER TO CPU) HF__PORT+1  读控制器硬件
                    状态(控制器至CPU)
158 ; HF__PORT+2-READ CONFIGURATIOH SWITCHES  HF__PORT+2
      读配置开关
159 ; HF__PORT+3-HOT USED  HF__PORT+3  未用
160 ; WHEN WRITTEN TO: (写入时)
161 ; HF__PORT+0-WRITE DATA (FROM CPU TO CONTROLLER)  HF__
      PORT+0  写入数据(CPU至控制器)
162 ; HF__PORT+1-CONTROLLER RESET  HF__PORT+1  控制器复位
163 ; HF__PORT+2-GENERATE CONTROLLER SELECT PULSE;  HF__
      PORT+2  产生控制器选择脉冲
164 ; HF__PORT+3-WRITE PATTERN TO DMA AND  INTERRURT
165 ;                MASK REGISTER  HF__PORT+3  位模写入DMA及中
                    断屏蔽寄存器
166 ;
167 ;
168
0320 169 HF__PORT1        EQU  0320H      ; DISK PORT  驱动器口子1读
                                            取内容定义:
0008 170 R1__BUSY         EQU  00001000B  ; DISK PORT 1 BUSY BIT  忙
0004 171 R1__BUS          EQU  00000100B  ; COMMAND/DATA BIT  命
                                            令/数据
0002 172 R1__IOMOOE       EQU  00000010B  ; MOOE BIT  输入/输出
0001 173 R1__REQ          EQU  00000001B  ; REQUEST BIT  请求
     174
0047 175 DMA__READ        EQU  01000111B  ; CHANIEL 3 (047H)
0048 176 DMA__WRITE       EQU  01001011B  ; CHANHEL 3 (04BH)
0000 177 DMA              EQU  0          ; DMA ADDRESS
0082 178 DMA__HIGH        EQU  082H       ; PORT FOR HIGH 4 BITS OF
                                            DMA
     179                                    硬盘机命令表,每个字均能与适
```

```
                                                    当的参数配合,组成一条
                                                    命令的命令块
0000      180 TST_RDY_CMO          EQU  00000000B  ; CNTLR READY (00H)
0001      181 RECAL_CMO            EQU  00000001B  ;       RECAL (01H)
0003      182 SENSE_CMO            EQU  00000011B  ;       SENSE (03H)
0004      183 FMTDRV_CMO           EQU  00000100B  ;       DRIVE (04H)
0005      184 CHK_TRK_CMO          EQU  00000101B  ;       T CHK (05H)
0006      185 FMTTRK_CMO           EQU  00000110B  ;       TRACK (06H)
0007      186 FMTBAO_CHO           EQU  00000111B  ;       BAD   (07H)
0008      187 READ_CMO             EQU  00001000B  ;       READ  (08H)
000A      188 WRITE_CHO            EQU  00001010B  ;       WRITE (DAH)
000B      189 SEEK_CMO             EQU  00001011B  ;       SEEK  (08H)
000C      190 IHIT_DRV_CMO         EQU  00001100B  ;       INIT  (0CH)
0000      191 RO_ECC_CMO           EQU  00001101B  ;       BURST (0DH)
000E      192 RO_BUFF_CMO          EQU  00001110B  ;       BUFFR (0EH)
000F      193 WR_BUFF_CMO          EQU  00001111B  ;       BUFFR (0FH)
00E0      194 RAM_DIAG_CMO         EQU  11100000B  ;       RAM   (E0H)
00E3      195 CHK_DRV_CMO          EQU  11100011B  ;       DRV   (E3H)
00E4      196 CNTLR_DIAG_CMO       EQU  11100100B  ;       CNTLR (E4H)
00E5      197 RD_LONG_CMO          EQU  11100101B  ;       RLONG (E5H)
00E6      198 WR_LONG_CMO          EQU  11100110B  ;       WLONG (E6H)
          199
0020      200 INT_CTL_PORT         EQU  20H        ; 8259 CONTROL PORT
                                                    8259A 控制口
0020      201 EOI                  EQU  20H        ; END OF INIERRUPT
                                                    COMMAND
          202
0008      203 MAX_FILE             EQU  8  最多容纳 8 台硬盘机
0002      204 S_MAX_FILE           EQU  2  系统单元最多容纳 2 台硬盘机
          205
          206 ASSUME               CS:CODE
0000      207 ORG                  0H
0000 55   208 DB                   055H            ; GENERIC BIOS HEADER
0001 AA   209 DB                   0AAH
0002 10   210 DB                   16D
          211
          212 ;
          213 ; FIXED DISK I/O SETUP  建立硬盘 I/O
          214 ; 完成;
```

```
                        215 ; ESTABLISH TRANSFER VECTIORS FOR THE FIXED
                              DISK  1. 生成硬盘机用向量
                        216 ; PERFORM POWER ON DIAGNOSTICS  2. 进行开机自
                              检,出错的话显示"1701"
                        217 ; SHOULD AN ERROR OCCUR A "1701" MESSAGE IS
                              DISPLAYED
                        218 ;
                        219 ;
                        220
0003                    221 DISK_SETUP   PROC FAR
0003 EBIE               222              JMP   SNORT L3
0005 35303030303539     223              DB    5000059 (C) COPYRIGHT IBM 1982
                                                     ; CORYRIGHT NOTICE
     20284329434F50
     59524947485420
     20494240203139
     3832
0023                    224 L3;
                        225          ASSUME   05;OUMMY'
0023 2BCO               226              SUB  AX, AX          ; ZERO
0025 BEDB               227              MOV  DS, AX  DS 指向 DUMMY 数据段
0027 FA                 228 CLI
0028 A14C00             229 MOV  AX, WORD PTR CRG_VECTOR
                                    ; GET  DISKETTE  VECTOR 先将软盘
                                      机参数块送给 DISK_VELTOR 向量保
                                      存
0028 A30001             230 MOV  WORD PTR DISK_VECTOR, AX
                                                ; INTO INT 40H
002E A14E00             231 MOV  AX, WORD PTR ORG_VECTOR+2
0031 A30201             232 MOV  WORD PTR DISK_VECTOR+2, AX
0034 C7064C005602       233 MOV  WORD PTR ORG_VECTOR, OFFSET DISK_IO
                                    ; HDISK HANDLER  送 DISK  IO
003A BC0E4E00           234 MOV  WORD PTR ORG_VECTOR+2, CS
003E B86007             235 MOV  AX, OFFSET HD_INT
                                    ; HDISK INTERRUPT  送硬盘中断程序
                                      首址
0041 A33400             236 MOV  WORD PTR HDISK_INT, AX
0044 BC0E3600           237 MOV  WORD PTR HDISK_INT+2, CS
0048 C70664008601       238 MOV  WORD PTR BOOT_VEC, OFFSET BOOT_STRAP
```

```
                                        ; BOOTSTRAP 送自启
                                          动程序偏移量
004E 8C0E6600    239     MOV   WORD PTR BOOT__VEC+2, CS
0052 C7060401E703 240    MOV   WORD PTR HF__TBL__VEC,OFFSET FD__TBL
                                        ; PARAMETER TBL
                                          送参数表偏移量
0058 8C0E0601    241     MOV   WORD PTR HF__TBL__VEC+2, CS
005C FB          242     STI
                 243
                 244     ASSUME DS;DATA
005D 884000      245     MOV   AX, DATA          ;ESTABLISH SEGME-
                                                  NTDS指向数据段DATA
0060 8EDB        246     MOV   DS, AX
0062 C606740000  247     MOV   DISK__STATUS, 0  ; RESET THE STATUS
                                                  INDICATOR 状态指示
                                                  器清 0
0067 C606750000  248     MOV   HF__NUM, 0       ; ZCRO COUNT OF DR-
                                                  IVES 驱动器个数计数
                                                  器清 0
006C C60643000   249     MOV   CMD__BLOCK+1, 0  ; DRIVE ZERO, SET
                                                  VALUE IN BLOCK
                                                  DCB第1字节清0(d=0)
0071 C606770000  250     MOV   PORT__OFF, 0     ; ZERO CARO OFFSET
                                                  I/O 口偏移量清 0
                 251
0076 892500      252     MOV   CX, 25H          ; RETRY COUNT 重复
                                                  次数送 CX
0079             253 L4:
0079 EBF200      254     CALL  HO__RESET__1     ; RESET CONTROLLER
                                                  复位控制器
007C 7305        255     JNC   L7
007E E2F9        256     LOOP  L4               ; TRY RESET AGAIN
                                                  不成功再试
0080 E98F00      257     JMP   ERROR__EX  重复 25 次,转 ERROR__EX
0083             258 L7:                        复位控制器成功转此
0083 890100      259     MOV   CX, 1
0086 BAB000      260     MOV   DX, 80H
                 261
0089 8800122     262     MOV   AX, 1200H        ; CONTROLLER DIA-
```

```
                                                    GNOSTICS  控制器
                                                    RAM 诊断 (AH) = 12
                                                    H, (AL) = 0
008C CD13          263      INT     13H
008E 7303          264      JNC     P7
0090 E9AF00        265      JMP     ERROR_EX
0093               266 P7:                                  RAM 诊断成功
0093 B80014        267      MOV     AX, 1400H       ; CDNTROLLER DIAG-
                                                    NOSTICS
0096 CO13          268      INT     13H  进行控制器内部诊断
0098 7303          269      JNC     P9
009A E9A500        270      JMP     ERROR_EX
0090               271 P9:                                  内部诊断成功
0090 C7066C000000  272      MOV     TIMER_LOW, 0    ; ZERO TIMER
00A3 A17200        273      MOV     AX, RESET_FLAG
00A6 303412        274      CMP     AX, 1234H       ; KEYBOARD RESET
                                                    键盘复位?
00A9 7506          275      JNE     P8
00AB C7066C009A01  276      MOV     TIMER_LOW, 4100 ; SKIP WAIT ON RES-
                                                    ET
00B1               277 PB:
00B1 E421          278      IN      AL, 021H        ; TIMER
00B3 24FE          279      ANO     AL, OFEH        ; ENABLE TIMER
00B5 E621          280      OUT     021H, AL        ; START TIMER
00B7               281 P4:
00B7 E88400        282      CALL    HO_RESET_1      ; RESET CONTROLLER
                                                    复位控制器
00BA 7207          283      JC      P10
00BC B80010        284      MOV     AX, 1000H       ; READY 驱动器准备
                                                    好?
00BF CD13          285      INT     13H
00C1 7308          286      JNC     P2
00C3               287 P10:
00C3 A16C00        288      MOV     AX, TIMER_LOW
00C6 30BE01        289      CMP     AX, 446D        ; 25 秒超时否?
00C9 72EC          290      JB      P4
00CB EB7590        291      JMP     ERPOR_EX
00CE               292 P2:
00CE B90100        293      MOV     CX, 1
```

```
00D1 BA8000      294     MOV     DX, 80H
                 295
0004 B80011      296     MOV     AX, 1100H       ; RECALIBRATE 重校
                                                   1号扇区，0号柱面，0
                                                   号磁头
00D7 CD13        297     INT     13H
00D9 7267        298     JC      ERROR_EX
                 299
00DB B80009      300     MOV     AX, 0900H       ; SET DRIVE PARAM-
                                                   ETERS 设定驱动器特
                                                   征参数
00DE CD13        301     INT     13H
00E0 7260        302     JC      ERROR_EX
                 303
00E2 B800CB      304     MOV     AX, 0C800H      ; DMA TO BUFFER
00E5 BEC0        305     MOV     ES, AX          ; SFT SEGMENT 设立
                                                   读、写缓存地址
00E7 2808        306     SUB     BX, BX
00E9 B8000F      307     MOV     AX, 0F00H       ; WRITE SECTOR BU-
                                                   FFEP 试写扇区
00EC CD13        308     INT     13H
00EE 7252        309     JC      ERROR_EX
                 310
00F0 FE067500    311     INC     HF_NUM          ; ORIVE ZERO RESPO-
                                                   NDED 成功. 还有其
                                                   他磁盘机否?
                 312
00F4 BA1302      313     MOV     OX, 213H        ; EXPANSION BOX 扩
                                                   展卡I/O口地址之一，
                                                   写00表示禁止扩展卡
                                                   工作，写01允许工作
00F7 8000        314     MOV     AL, 0
00F9 EE          315     OUT     DX, AL          ; TURH BOX DFF 禁
                                                   止扩展卡工作
00FA BA2103      316     MOV     DX, 321H        ; TEST IF CONTROLLER
                                                   测试刚才那个驱动器是
                                                   否装在系统单元内.
0FD EC           317     IN      AL, DX          ; …IS IN THE SYSTEM
                                                   UNIT 读控制器状态
```

```
                                                    (硬件),其定义见170—
                                                    173行
00FE 240F           318     INO     AL, 0FH
0100 3C0F           319     CMP     AL, 0FH
0102 7406           320     JE      BOX OM
0104 C7066C00A401   321     MOV     TIMER__LOW, 420D    ; CONTROLLER IS IN
                                                    SYSTEM UNIT 控制
                                                    器在系统单元内
010A                322 BOX__ON:
010A BA1302         323     MOV     OX, 213H            ; EXPANSION BOX
010D B0FF           324     MOV     AL, 0FFH                允许扩展卡工作
010F EE             325     OUT     DX, AL              ; TURN BOX ON
                    326
0110 B90100         327     MOV     CX, 1               ; ATTEMPT NEXT DR-
                                                    IVES
0113 8A8100         328     MOV     DX, 081H
0116                329 P3:
0116 28C0           330     SUB     AX, AX              ; RESET 复位硬盘机,
                                                    通过写I/O口321H复
                                                    位
0118 CD13           331     INT     13H
011A 7240           332     JC      POD__DCNE
011C 880011         333     MOV     AX, 01100H          ; RECAL 重校
011F CD13           334     INT     13H
0121 730E           335     JNC     P5
0123 A16C00         336     MOV     AX, TIMER__LOW      重校有错, 准备再作试一
                                                    次
0126 3DBE01         337     CMP     AX, 446D            ; 25 秒
0129 72EB           338     JB      P3
0128 E02F90         339     JMP     POO__DONE
012E                340 P5:
012E B80009         341     MOV     AX, 0900H           ; INITIALIZE CHARA-
                                                    CTERISTICS 重校无
                                                    错,现初始特征值
0131 CD13           342     INT     13H
0133 7227           343     JC      POO__DONE
0135 FE067500       344     INC     HF__NUM             ; TALLY AHOTHER
                                                    DRIVE 可用驱动器个
                                                    数加1
0139 81FA8100       345     CMP     DX, (80H + S__MAX__FILE - 1)  (S__MAX__
```

```
                                FILE = 2)
013D 731D          346      JAE     POO_DONE
013F 42            347      INC     DX
0140 EBD4          348      JMP     P3
                   349
                   350 ; ……POO ERROR
                   351                                  显示错误码
0142               352 ERROR_EX;
0142 BD0F00        353      MOV     BP, 0FN      ; POO ERROR FLAG
0145 2B0D          354      SUB     AX, AX
0147 8BF0          355      MOV     SI, AX
0149 B9060090      356      MOV     CX, F17L     ; MESSAGE CHARACTER
                                                 COUNT 输出字符个数计数
                                                 器(=6)
0140 8700          357      MOV     BH, 0        ; PAGE ZERO 0页显示缓存
014F               358      OUT_CH;
014F 2E8A846801    359      MOV     AL, CS;F17(SI) ; GET BYTE
0154 B40E          360      MOV     AH, 140      ; VIDFO DUT
0156 CD10          361      INT     10H          ; DISPLAY CHARACTER
                                                 调 VIDEO 中断输出字符
0158 46            362      INC     SI           ; NEXT CHAR
0159 E2F4          363      LOOP    OUT_CH       ; DO MORE
0158 F9            364      STC
015C               365 POO_DONE;
015C FA            366      CLI
015D E421          367      IN      AL, 021H     ; BE SUPE TIMEP IS DIS-
                                                 ABLED
015F 0C01          368      OR      AL, 01H
0161 E621          369      OUT     021H, AL
0163 FB            370      STI
0164 E8A500        371      CALL    DSDL
0167 CB            372      RET
                   373
0168 31373031      374 F17  DO      '1701', 0OH, 0AH   硬盘机错误码 1701
LOC OBJ                     LINE    SOURCE
016C 0D
016D 0A
0006               375      F17L    EQU     $-F17
                   376
```

```
016E            377 HO__RESET__1  PROC  NEAR; 过程 HD__RESET__1 (AX) 为入
                                          口号数,(AX) = 0000H
016E 51         378     PUSH  CX        ; SAVE REGISTER
016F 52         379     PUSH  DX
0170 F8         380     CLC             ; CLEAR CARRY 清 CY, 它将作为出口号
                                          数表示本次复位是否成功
0171 890001     381     MOV   CX, 0100H; RETRY COUNT 重试次数
0174            382 L6:
0174 E80706     383     CALL  PORT__1   取 I/O 地址 321 H
0177 EE         384     OUT   DX, AL    ; RESET CARO 复位
0178 E80306     385     CALL  PORT__1   取 I/O 地址 321H
0178 EC         386     IN    AL, DX    ; CHECK STATUS 读 I/O 口 321 H,其第 2
                                          位是错误位,见 170__173 行的定义
017C 2402       387     AND   AL, 2     ; ERROR BIT
017E 7403       388     JZ    R3
0180 E2F2       389     LDOP  L6
0182 F9         390     STC
0183            391 R3:                                  复位成功则 CY = 0 失败 CY = 1
0183 5A         392     POP   DX        ; RESTORE REGISTER
0184 59         393     POP   CX
0185 C3         394     RET                                              返回
                395 HD__RESET__1  ENOP
                396
                397 DISK__SETUP   ENOP
                398
                399 ; INT 19 中断 19 自启动装入程序
                400 ;
                401 ; INTERRUPT 19 BOOT STRAP LOADER 自启动装入程序完成:
                402 ;
                403 ; -THE FIXED DISK BIOS REPLACES THE INTERRURT 19 建
                      立自启动装入程序向量
                404 ;  BOOT STRAP VECTOR WITH A POINTER TO THIS BOOT
                      ROUTINE 复位缺省硬盘机和软盘机参数向量
                405 ; -RESET THE DEFAULT DISK AND OISKETTE PARAMETER
                      VECTORS 启动块读自驱动器 0 号柱面 1 号扇区
                406 ; -THE BOOT BLOCK TO BE READ IN WILL BE ATTEMPTEO
                      FROM
                407 ;  CYLINDER 0 SECTDR 1 OF THE DEVICE
                408 ; -THE BOOTSTRAP SEQUENCE IS: 自启动顺序
```

```
                    409 ;    >ATTEMPT TO LOAD FROM THE DISKETTE INTO
                             THE BOOT  1. 先读软盘机,把自启动移序装入启动区
                             (首址0000;7000),然后控制权转至该位置
                    410 ;    LOCATION (0000; 7C00)AND TRNSFER CONTROL
                             THERE
                    411 ;    >IF THE DISKETTE FAILS THE FIXED DISK IS
                             TRIED FOR A  2. 软盘机自启动失败,转硬盘机,硬
                             盘机自启
                    412 ;    VALID BOOTSTRAP BLOCK.A VALID DOOT BLOCK
                             ON THE  动程序块的末二个字节是055H, 0AAH
                    413 ;    FIXED DISK CONSISTS OF THE BYTES 055H 0AAH
                             A5 THE  3. 软、硬盘机自启动均失败,控制转至 ROM
                             BASIC
                    414 ;    LAST TWO BRTES OF THE BLOCK
                    415 ;    >IF THE ABOVE FAILS CONTROL IS PASSED TO
                             RESIDENT BASIC;
                    416 ;
                    417 ;
                    418
0186                419 BOOT_STRAP;
                    420      ASSUME  DS;DUHMY, ES;DUMMY
0186 28C0           421      SUB     AX, AX
0188 BEDB           422      MOV     DS, AX          ; ESTABLISH SEGMENT
                                                     DS 指向数据段 DOMMY
                    423
                    424 ; RESET PARAMETER VECTORS  复位参数向量
                    425
018A FA             426      CLI
0188 C7060401E703   427      MOV     WORD PTR HF_TBL_VEC, OFFSET FD_
                                     TBL  硬盘机的序数区首址 FD_TBL
0191 8C0E0601       428      MOV     WORD PTR HF_TBL_VEC+2, CS
0195 C70678000102   429      MOV     WORD PTR DISKETTE_PARM, OFFSET
                                     DISKETTE_TBL  软盘机的参数区首址 DI-
                                     SKETTE_TBL
0198 8C0E7A00       430      MOV     WORD PTR DISKETTE_PARM+2, CS
019F FB             431      STI
                    432
                    433 ; ATTEMPY BOOTSTRAP FROM DISKEITE  先进行软盘
                             机自启动尝试
```

```
                        434
01A0 B90300             435     MOV     CX, 3           ; SET RETRY COUNT 重试3次
01A3                    436 H1:                         ; IPL_SYSTEM
01A3 51                 437     PUSH    CX              ; SAVE RETRY COUNT
11A4 28D2               438     SUB     DX, DX          ; DRIVE ZERO 指定0号驱动器
01A6 2BC0               439     SUB     AX, AX          ; RESET THE DISKETTE 复位软盘器(缺省的那一台)
01A8 CD13               440     INT     13H             ; FILE IO CALL
01AA720F                441     JC      H2              ; IF ERROR, TRY AGAIN
01ACB80102              442     MOV     AX, 0201H       ; READ IN THE SINGLE SECTOR 复位成功,准备读入第1扇区
                        443
01AF 2802               444     SUB     OX, DX
01B1 BEC2               445     MOV     ES, DX          ; ESTABLISH SEGMENT 设置 DMA 地址.
01B3 BB007C             446     MOV     BX, OFFSET BOOT_LOCN
                        447
01B6 B90100             448     MOV     CX, 1           ; SECTOR 1, TRACK 0 读0号磁道第1扇区的内容
01B9 CD13               449     INT     13H             ; FILE IO CALLL
01BB 59                 450 H2: POP     CX              ; RECOVER RETRY COUNT
01BC 730A               451     JNC     H4              ; CF SET BY UNSUCCESSFUL REHO CY=0 说明读取成功,转 BOOT_LOCN
01BE 80FC80             452     CMP     AH, 80H         ; IF TIME OUT, NO RETRY 是超时错, 也就是驱动器未响应本次读取命令?
01C1 740A               453     JZ      H5              ; TRY FIXED DISK
01C8 E20E               454     LOOP    H1              ; DO IT FOR RETRY TIMES 否再试一下
01C3 EB0690             455     JMP     H5              ; UNABLE TO IPL FROM THE DISKETTE 重试了2次均失败,转硬盘机
01C3                    456 H4:                         ; IPL WAS SUCCESSFUL
01C8 EA007C0000         457     JMP     BOOT_LOCN
                        458
```

```
                459 ; ···ATTEMPT BOOTSTRAP FROM FIXED DISK 尝试调进磁
                                                        盘机上的自启动程序
                460
01CD            461 H5:
01C0 2BCD       462     SUB   AX, AX                ; RESET DISKETTE 先
                                                        将软盘机复位
01CF 2B02       463     SUB   DX, DX
01D1 CD13       464     INT   13H
01D3 B90300     465     MOV   CX, 3                ; SET RETRY COUNT 重
                                                        试硬盘机3次
01D6            466 H6:                            ; IPL_SYSTEM
01D6 51         467     PUSH  CX                   ; SAVE RETRY COUNT
01D7 BA8000     468     MOV   DX, 0080H            ; FIXED DISK ZERO 指
                                                        定硬盘机
01DA 2BC0       469     SUB   AX, AX               ; RESET THE FIXED DI-
                                                        SK
01DC CD13       470     INT   13H                  ; FILE IO CALL 先复位
                                                        0号硬盘机
01DE 7212       471     JC    H7                   ; IF ERROR,TRY AGAIN
01E0 B80102     472     MOV   AX, 0201H            ; READ IH THE SINGLE
                                                        SECTOR 复位成功,准备
                                                        读0号柱面上第1扇区的
                                                        内容
01E3 2B08       473     SUB   BX, BX
01E5 8EC3       474     MOV   E5, DX
01E7 B80007C    475     MOV   BX,OFFSET BOOT_LOCN; TO THE BOOT LOCAT-
                                                        ION 建立DMA地址
01EA BA8000     476     MOV   DX, 80H              ; DRIVE NUMBER
01ED B90100     477     MOV   CX, 1                ; SECTOR 1, TRACK 0
01F0 CD13       478     INT   13H                  ; FILE IO CALL 读自启
                                                        动0程序
01F2 59         479 H7: POP   CX                   ; RECOVER RETRY CO-
                                                        UNT
01F3 7208       480     JC    H8
01F5 A1FE70     481     MOV   AX, WORD PTR BOOT_LOCN+510D 取读出扇区
                                末两个字节
01F8 3D55AA     482     CMP   AX, 0AA55H           ; TEST FOR GENERIC
                                                        BOOT BLOCK 是自启动
                                                        程序块?
```

```
01FB 74CB   483         JZ      H4
01FD        484 H8:
01FD E2D7   485         LDOP    H 6             ; DO IT FOR RETRY
                                                  TIMES 否,再试
            486
            487 ; UNABLE TO IPL FROM THE DISKETTE OR FIXED DISK
                  0号软盘机0号硬盘机均没有自启动程序
            488
01FF CD18   489         INT     18 H            ; RESIDENT BASIC
                                                  转 ROM BASIC
            490
0201        491 DISKETTE_TBL: 软盘机用数据块
            492
0201 CF     493         DB      11001111B       ; SRT=C,HD UNLOAD
                                                  =OF-IST SPEC BYTE
0202 02     494         DB      2               ; HD LDAD=1, MODE
                                                  =DMA - 2ND SPEC
                                                  BYTE
0203 25     495         DB      25H             ; WAIT AFTER OPN
                                                  TIL MOTOR OFF
0204 02     496         DB      2               ; 512 BYTES PER SE-
                                                  CTOR
0205 08     497         DB      8               ; EOT (LAST SECTOR
                                                  ON TRACK)
0206 2A     498         DB      02AH            ; GAP LENGTH
0207 FF     499         DB      0FFH            ; DTL
0208 50     500         DB      050H            ; GAP LENGTH FOR
                                                  FORMAT
0209 F6     501         DB      0F6H            ; FILL BYTE FOR FOR-
                                                  MAT
020A 19     502         DB      25              ; HEAD SETTLE TIME
                                                  (MILLISECONDS)
020B 04     503         DB      4               ; MOTOR START TIME
                                                  (1/8 SECONO)
            504
            505 ; MAKE SURE THAT ALL HOUSEKEEPING IS DOME BEFORE
                  EXIT 下面的 DSBL 过程用来检查所有内部事务是否做好了。
            506
020C        507 DSBL    FROC    NEAR
```

```
                    508         ASSUME  DS,DATA
020C IE             509         PUSH    DS                  ; SAVE SEGMENT
0200 884000         510         MOV     AX, DATA
0210 8ED8           511         MOV     DS, AX
                    512
0212 8A267700       513         MOV     AH, PORT_OFF
0216 50             514         PUSH    AX                  ; SAVE OFFSET
                    515
0217 C606770000     516         MOV     PORT_OFF, OH
021C E86905         517         CALL    PORT_3   (DX) = HF_PORT + 3
021F 2AC0           518         SUB     AL, AL
0221 EE             519         OUT     DX, AL              ; RESET INT/DMA
                                                            MASK 复位INT/
                                                            DMA屏蔽寄存器.共进
                                                            行四次,把可能连接的4
                                                            个硬盘机 INT/DMA
                                                            屏蔽寄存器全复位掉.
0222 C606770004     520         MOV     PORT_OFF, 4H
0227 E85E05         521         CALL    PORT_3
022A 2AC0           522         SUB     AL, AL
022C EE             523         OUT     DX, AL              ; RESET INT/DMA
                                                            MASK
022D C606770008     524         MOV     PORT_OFF, 8H
0232 E85305         525         CALL    PORT_3
0235 2AC0           526         SUB     AL, AL
LOC OBJ                         LINE    SOURCE
0237 EE             527         OUT     DX, AL              ; RESET INT/DMA
                                                            MASK
0238 C60677000C     528         MOV     PORT_OFF, 0CH
0230 E8480S         529         CALL    PORT_3
0240 2AC0           530         SUB     AL, AL
0242 EE             531         OUT     DX, AL              ; RESET INT/DMA
                                                            MASK
0243 B007           532         MOV     AL, 07H
0245 E60A           533         OUT     DMA + 10, AL        ; SET DMA MOOE TO
                                                            DISABLE
0247 FA             534         CLI                         ; DISABLE INTERRU-
                                                            PTS
0248 E421           535         IN      AL, 021H
024A 0C20           536         OR      AL, 020H
```

```
024C E621       537         OUT     021H, AL            ; DISABLE INTERRUPT 5
024E FB         538         STI                         ; ENABLE INTERRUPTS
024F 58         539         POP     AX                  ; RESTORE OFFSET
0250 88267700   540         MOV     PORT_OFF, AH
0254 IF         541         FOP     DS                  ; RESTORE SEGMENT
0255 C3         542         RET
                543 DSBL ENDP
                544
                545 ; ..........................................
                546 ; FIXED DISK BIOS ENTRY POINT: 硬盘机 BIOS 入口
                547 ; ..........................................
                548
0256            549 DISK_IO PROC FAR  DISK_IO_DISK 转送程序
                550         ASSUME  DS:NUTHING, ES:NOTHING
0256 80FA80     551         CMP     DL, 80H             ; TEST FOR FIXED DISK
                                                          DRIVE 是进行硬盘机 I/O?
                                                          (≥80H 为硬盘机)
0259 7305       552         JAE     HARD_DISK           ; YES, HANDLE HERE
0258 CD40       553         INT     40H                 ; DISKTTE MANDLTEE 通
                                                          过中断 40H 转软驱动器 I/O
                                                          程序
025D            554 RET_2:
025D CA0200     555         RET     2                   ; BACK TO CALLER
0260            556 HARD_DISK; 硬盘机 I/O:
                557         ASSUME  DS:DATA
0260 FB         558         STI                         ; ENABLE INTERRUPTS 允
                                                          许中断
0261 0AE4       559         OR      AH, AH
0263 7509       560         JNZ     A3
0265 CD40       561         INT     40H                 ; RESET NEC WHEN AH
                                                          =0  AH=0 表示复位磁盘
                                                          机,先复位软盘机
0267 2AE4       562         SUB     AH, AH
0269 80FA81     563         CMP     DL, (80H+S_MAX_FILE-1)  (DL) > 81H?
                                    复位磁盘机号大于 81H 则返回
026C 77EF       564         JA      RET_2
026E            565 A3:
026E 80FC08     566         CMP     AH, 08              ; GET PARAMETERS IS A
                                                          SPECIAL CASE 要求返回
```

```
                                                    当前驱动器参数?
0271 7503       567      JNZ     A2
0273 E91A01     568      JMP     GET_PARM_N         是则转 GET PARM_N
0276            569 A2:
0276 53         570      PUSH    BX                 ; SAVE REGISTERS DUR-
                                                    ING OPERATION 保护寄
                                                    存器
0277 51         571      PUSH    CX
0278 52         572      PUSH    DX
0279 1E         573      PUSH    DS
027A 06         574      PUSH    ES
027B 56         575      PUSH    SI
027C 58         576      PUSH    DI
                577
027D E86A00     578      CALL    DISK_IO_CONT       ; PERFORM THE OPERAT-
                                                    ION 主I/O程序,完成要求
                                                    的动作(复位,取参数除外)
                579
0280 50         580      PUSH    AX
0281 E888FF     581      CALL    DSBL               ; BE SURE DISADLES OC-
                                                    CURREO 调 DSBL, 复位
                                                    屏蔽寄存器
0284 884000     582      MOV     AX, DATA
0287 8ED8       583      MOV     DS, AX             ; ESTABLISH SEGMENT
0289 58         584      POP     AX
028A 8A267400   585      MOV     AH, DISK_STATUS    ; GET STAVUS FROM OP-
                                                    ERATION 取命令执行后的
                                                    状态字节, 其非零表示有错
028E 80FC01     586      CMP     AH, 1              ; SET THE CARRY FLAG
                                                    TO INOICATE CY 取反,
                                                    使 CY 当操作成功时为0,反
                                                    之为1
0291 F5         587      CMC                        ; SUCCESS OR FAILURE
0292 5F         588      POP     DI                 ; RESTORE REGISTERS 恢
                                                    复寄存器
0293 5E         589      POP     SI
0294 07         590      POP     ES
0295 1F         591      POP     DS
0296 5A         592      FOP     DX
```

```
0297 59          593     POP     CX
0298 5B          574     POP     BX
0299 CA0200      595     RET     2                ; THROW AWAY SAVED
                                                  FLAGS 扔掉INT指令存
                                                  入的标志位字,返回
                 596 DISK_IO ENOP
                 597
029C             598 M1 LABEL  WORD               ; FUNCTION TRANSFER
                                                  TABLE 地址散转表
029C 3803        599     DW      DISK_RESET       ; 000H
029E 4003        600     DW      RETURN_STATUS    ; 001H
02A0 5603        601     DW      DISK_READ        ; 002H
02A2 6003        602     DW      DISK_WRITE       ; 003H
02A4 6A03        603     DW      DISK_VERF        ; 004H

02A6 7203        604     DW      FMT_TRK          ; 005H
02AB 7903        605     DW      FMT_BAD          ; 006H
02AA 8003        606     DW      FMT_DRV          ; 007H
02AC 3003        607     DW      BAD_COMMWAMO     ; 008H
02AE 2704        608     DW      INIT_DRV         ; 009H
02B0 CF04        609     DW      RO_LONG          ; 00AH
02B2 OD04        610     DW      WR_LONG          ; 00BH
02B4 F204        611     DW      DISK_SEEK        ; 00CH
02B6 3803        612     DW      DISK_RESET       ; 00DH
02B8 F904        613     DW      RO_BUFF          ; 00EH
02BA 0705        614     DW      WR_BUFF          ; 00FH
02BC 1505        615     DW      TST_ROY          ; 010H
02BE 1C05        616     DW      HDISK_RECAL      ; 011H
02C0 2305        617     DW      RAM_DIAG         ; 012H
02C2 2A05        618     DW      CHK_DRV          ; 013H
02C4 3105        619     DW      CNTLR_DIAG       ; 014H
002A             620 MIL EQU     $-MI
                 621
02C6             622 SETUP_A PROC NEAR
                 623
02C6 C606740000  624     MOV     DISK_STATUS,0    ; RESET THE STATUS IN-
                                                  DICATOR清状态字节,表示
                                                  暂无错误发生
02CB 51          625     PUSH    CX               ; SAVE CX
```

```
                626
                627 ; CALCULATE THE PORT OFFSET
                628
02CC 8AEA       629     MOV     CH, DL          ; SAVE DL 取驱动器号，其值
                                                  在 0～7 之间
02CE 80CA01     630     OR      DL, 1
02D1 FECA       631     DEC     DL
02D3 D0E2       632     SHL     DL, 1           ; GENERATE OFFSET 生成
                                                  偏移地址
02D5 88167700   633     MOV     PORT_OFF, DL;   STORE OFFSET 对应的硬盘
                                                  机号：
```

0	4	8	12
0,1	2,3	4,5	6,7

```
02D9 8AD5       634     MOV     DL, CN          ; RESTORE DL
02DB 80E201     635     ANO     DL, 1
                636
02DE B105       637     MOV     CL, 5           ; SHIFT COUNT
02E0 D2E2       638     SHL     DL, CL          ; DRIVE NUMBER (0, 1) 置
                                                  DCB 中的 d，偶数硬盘机 d=
                                                  0，奇数硬盘机 d=1
02E2 0AD6       639     OR      DL, DH          ; HEAD NUMBER 再取进磁
                                                  头号
02E4 88164300   640     MOV     CMD_BLOCK+1, DL 送 DCB 第 1 字节
02E8 59         641     POP     CX
02E9 C3         642     RET 返回
                643 SETUP_A ENDP
                644
02EA            645 DISK_IO_CONT PROC NEAR 主 I/O 程序
02EA 50         646     PUSH    AX
02EB B84000     647     MOV     AX, DATA
02EE 8ED8       648     MOV     DS, AX          ; ESTABLISH SEGMENT 先
                                                  建立段地址
02F0 58         649     POP     AX
02F1 80FC01     650     CMP     AH, 01H         ; RETURN STATUS 是要求读
                                                  上次操作的状态?
02F4 7503       651     JNZ     A4
02F6 EB5590     652     JMP     RETURN_STATUS
02F9            653 A4:
```

```
02F9 80EA80        654 SUB    DL, 80H              ; CONVERT DRIVE NUMBER TO
                                                     0 BASED RANGE (AH)≠01
                                                     H 硬盘号转成以 0 为基的数
02FC 80FA08        655 CMP    DL, MAX__FILE        ; LEGAL DRIVE TEST 最多允
                                                     许接 8 个硬盘机,(MAX-FILE)
                                                     =8
02FF 732F          656 JAE    BAD__COMMAND
                   657
0301 E8C2FF        658 CALL SETUP__A  建立欲 I/O 硬盘机的口子地址
                   659
                   669 ; SET UP COMMAND BLOCK
                   661
0304 FEC9          662 DEC    CL                   ; SECTORS 0—16 FOR CONT-
                                                     ROLLER 扇区号 0～16
0306 C606420000    663 MOV    CMD__BLOCK+0, 0      操作类,操作码暂送 0
0308 880E4400      664 MOV    CMD__BLOCK+2, CL;    SECTOR AND HIGH 2 BITS
                                                     CYLINDER DCB 第 2 字节:
                                                     扇区号和柱面号的高 2 位
030F 882E4500      665 MOV    CMD__BLOCK+3, CN;    CYLINDER DCB 第 3 字节:
                                                     柱面号
0313 A24600        666 MOV    CMD__BLOCK+4, AL;    INTERLEAVE/BLOCK COU-
                                                     NT DCB 第 4 字节:块计数器
                                                     /空白页
0316 A07600        667 MOV    AL, CONTROL__BYTE;   CONTROL BYTE (STEP OP-
                                                     TION)
0319 A24700        668 MOV    CMD__BLOCK+5, AL     送 DCB 第 5 字节:控制场
031C 50            669 PUSH   AX                   ; SAVE AX
037D 8AC4          670 MOV    AL, AH               ; GET INTO LOW BYTE
031F 32E4          671 XOR    AH, AH               ; ZERO HIGH BYTE
3021 D1E0          672 SAL    AX, 1                ; *2 FOR TABLE LOOKUP
0323 88F0          673 MOV    SI, AX               ; PUT INTO SI FOR BRAN-
                                                     CH (AX)=原 (AH)×2
0325 302A00        674 CMP    AX, M1L              ; TEST WITHIN RANGE 入
                                                     口参数 AH 应落在 0～14H 范
                                                     围内
0328 58            675 POP    AX                   ; RESTORE AX
0329 7305          676 JNB    BAD__COMMAND
0328 2EFFA49C02    677 JMP    WORD PTR CS:(SI+DFFSET MI)  按 AH 的内容散
                              转至各自的处理程序
```

```
0330                678 BAD_COMMANO:
0330 C606740001     679 MOV     DISK_STCTUS, BAD_CMD;  COMMAND EPROR 入口
                                                       参数不符要求产生命令不正确错误
0335 B000           680 MOV     AL, 0
LOC OBJ                 LINE    SOURCE
0337 C3             681 RET
                    682 DISK_IO_CONT  ENOP  主 I/O 程序到此结束
                    683
                    684 ; ..........................................................
                    685 ; RESET THE DISK SYSTEM (AH = 000H)  (AH) = 0;复位硬
                          盘机系统
                    606 ; ..........................................................
                    687
0338                688 DISK_RESET  PROC  NEAR
0338 E84304         689 CALL    PORT_1              ; RESET PORT  取硬盘机口子
                                                      1, 通过写 00 复位
0338 EE             690 OUT     DX, AL              ; ISSUE RESET
033C E83F04         691 CALL    PORT_1              ; CONTROLLER HARDWARE
                                                      STATUS  还是取硬盘机口1, 现
                                                      读取复位后的结果
033F EC             692 IN      AL, DX              ; GET STATUS
0340 2402           693 AND     AL, 2               ; ERROR BIT  状态字第节 1 位等
                                                      于 1 说明复位操作有误, 见硬盘
                                                      机状态字节的定义
0342 7406           694 JZ      DR1
0344 C606740005     695 MOV     DISK_STATUS BAD_RESET  复位失败, 显标志
0349 C3             696 RET
034A                677 DR1:
034A E90A00         698 JMP     INIT_DRV            ; SET THE DRIVE PARAMET-
                                                      ERS  复位成功, 转送驱动器参
                                                      数程序段
                    699 DISK_RESET  ENOP
                    700
                    701 ;
                    702 ; DISK STATUS ROUTINE (AH = 001H)  (AH) = 01H;  取上
                          次磁盘操作的状态
                    703 ;
                    704
034D                705 RETURN_STATUS PROC NEAR
034D A07400         706 MOV     AL, DISK_STATUS ; DBTAIN PREVIOUS STATUS
```

```
                                                取参数,送至 AL
0350 C606740000 707 MOV    DISK__STATUS,0; RESET STATUS  清状态保存字节
0355 C3         708 RET
                709 RETURN__STATUS ENDP
                710
                711 ;
                712 ; DISK READ ROUTINE (AH=002H)  (AH)=02H; 读扇区
                713 ;
                714
0356            715 DISK__READ PROC NEAR
0356 B047       716 MOV    AL,DMA__READ; MODE BYTE FOR DMA READ
                                                送读扇区的 DCB 操作码、操作类
0358 C606420008 717 MOV    CMO__BLOCK+0, READ__CMD
035D E9E501     718 JMP    DMA__OPN          转 DMA__OPN
                719 DISK__READ ENOP
                720
                721 ;
                722 ; DISK WRITE ROUTINE (AH=003H)  (AH)=03H; 写扇区
                723 ;
                724
0360            725 DISK__WRITE PROC NEAR
0360 B04B       726 MOV    AL, DMA__WRITE; MODE BYTE FOR DMA WRITE
                                                送写扇区的 DCB 操作码、操作类
0362 C60642000A 727 MOV    CMD__BLOCK+0, WRITE__CMD
0367 E9DB01     728 JMP    DMA__OPN          转 DMA__OPN
                729 DISK__WRITE ENDP
                730
                731 ;
                732 ; DISK VERIFY (AH =004H)  (AH)=04H; 校验扇区
                733 ;
                734
036A            735 DISK__VERT PROC NEAR
036A C606420005 736 MOV    CMD__BLOCK+0, CHK __TRK__CMD  送校验扇区的
                                                DCB 操作码,操作类
036F E9C401     737 JMP    NDMA__OPN  转 DMA__OPN
                738 DISK__VERF ENDP
                739
                740 ;
                741 ; FORMATTING (AH=005H 006H 007H)  (AH)=05H、06H、
```

```
                743
0372            744 FMT_TRK PROC NEAR; FORMAT  TRACK(AH=00SH)
                                        格式化磁道
0372 C606420006 745 MOV     CMD_BLOCK, FMTTRK_CMO
0377 EB0C       746 JMP     SHORT FMT_CONT
                747 FMT_TRK ENDP
                748
0379            749 FMT_BAD PROC NEAR; FORMAT BAD TRACK(AH=006H)
                                        格式化磁道并标出坏了的磁道
0379 C600420007 750 MOV     CMD_BLOCK, FMTBAD_CMD
037E EB05       751 JMP     SHORT FMT_CONT
                752 FMT_BAD ENDP
                753
0380            754 FMT_DRV PROC NEAR; FORMAT DRIVE (AH=007H) 从指
                                        定的磁道开始格式化驱动器
0380 C606420004 755 MOV     CMD_BLOCK, FMTDRV_CMD
                756 FMT_DRV ENDP
                757

0385            758 FMT_CONT:
0385 A04400     759 MOV     AL, CMD_BLOCK+2; ZERO OUT SECTOR FIELD
                                        清 DCB 中的扇区号
0388 24C0       760 AND     AL, 11000000B
038A A24400     761 MOV     CMD_BLOCK+2, AL
0380 E9A601     762 JMP     HDMA_OPN  转 HDMA_OPN
                763
                764 ;
                765 ; GET PARAMETERS(AH=8)  (AH)=8:返回当前驱动器参数
                766 ;
                767
0390            768 GET_PARM_N LABEL NEAR
0390            769 GET_PARM PROC FAR       ; SET DRIVE PARAMETERS
0390 1E         770 PUSH    DS              ; SAVE REGISTERS 保护 DS,
                                              ES, BX
0391 06         771 PUSH    ES
0392 53         772 PUSH    BX
                773
                774 ASSUME  DS:DUMMY
0393 2BC0       775 SUB     AX, AX          ; ESTABLISH ADDRESSING
```

```
0395 8ED8       776     MOV     D5, AX
0397 C41E0401   777     LES BX, HF__TBL__VEC  建立硬盘参数表的段、段内地址
                778     ASSUME  DS:DATA
0398 884000     779     MOV     AX, DATA
039E 8ED8       780     MOV     DS, AX          ; ESTABLISH SEGMENT
                                                  建立 DS 段 DATA
                781
03A0 80EA80     782     SUB     DL, 80H
03A3 80FA08     783     CMP     DL, MAX__FILE   ; TEST WITHIN RAHGE
                                                  硬盘机号不能大于 7
03A6 732F       784     JAE     G4
                785
03A8 E818FF     786     CALL    SETUP__A  建立欲 I/O 硬盘机的 I/O 口子地址
                787
03ABE80F03      788     CALL    SW2__OFFS  取回参数表的偏移量(在 AX 中返回)
03AE7227        789     JC      G4                参数表格式见 828—829 行
0380 0308       790     ADD     BX, AX
                791
0382 268807     792     MOV     AX, ES:(BX)     ; MAX NUMBER OF CY-
                                                  LINDERS  最大柱面号
0385 200200     793     SUB     AX, 2           ; ADJUST FOR O—N
                794                             ; AND RESERVE LAST
                                                  TRACK
0388 8AE8       795     MOV     CH, AL
038A 250003     796     AND     AX, 0300H       ; HIGH TWO BITS OF
                                                  CYL  柱面号高 2 位
038D D1E8       797     SHR     AX, 1
038F DIE8       798     SHR     AX, 1
03C1 0C11       799     OR      AL, 011H        ; SECTORS  最大扇区数
                                                  及柱面号高位部分，扇区
                                                  数定为 17
03C3 BACB       800     MOV     CL, AL
                801
03C5 268A7702   802     MOV     DH, ES:(BX)(2)  ; MEADS  磁头数
03C9 FECE       803     DEC     DH              ; O—N RANGE  调整至
                                                  O—N
03CB 8A167500   804     MOV     DL, HF__NUM     ; DRIVE COUNT  驱动器
                                                  个数
03CF2BC0        805     SUB     AX, AX
```

```
0301                    806 G5:
0301 58                 807     POP   BX                          ; RESTORE RESIST-
                                                                    ERS 恢复寄存器
0302 07                 808     POP   ES
0303 1F                 809     POP   DS
0304 CA0200             810     RET   2 返回
0307                    811 G4:
0307 C606740007         812     MOV   DISK_STATUS, INIT_FAIL ; OPERATION FAI-
                                                                    LED 置失败标志
03DC B407               813     MOV   AH, INIT_FAIL
03DE 2AC0               814     SUB   AL, AL
03E0 2BD2               815     SUB   DX, DX
03E2 2BC9               816     SUB   CX, CX
03E4 F9                 817     STC                               ; SET ERROR FLAG
03E5 EBEA               818     JMP   G5
                        819     GET_PARM ENDP
                        820
                        821 ; ..........................................................
                        822 ; INITIALIZE DRIVE CHARACTERISTICS 初始驱动器特征
                        823 ;
                        824 ; FIXED DISK PARAMETER TABLE 硬盘参数表:
                        825 ;
                        826 ; -THE TABLE IS COMPOSED OF A BLOCK DEFINED AS:
                        827 ;
                        828 ;   (I WORD) - MAXIMUM NUMBER OF CYLINDERS (1
                              WORD)—最大柱面号
                        829 ;   (1 BYTE) - MAXIMUM NUMBER OF HEADS (1 BYTE)
                              —磁头个数
                        830 ;   (1 WORU) - STARTING REDUCED WRITE CURRENT
                              CYL (1 WORD)—启始约减写当前柱面
                        831 ;   (1 WORD) - STARTING WRITE PRECOMPENSATION CYL
                              (1 WORD)—启始写予补偿柱面
                        832 ;   (1 BYTE) - MAXIMUM ECC DATA BURST LENGTH (1
                              BYTE)—最大 ECC 数据突发长度
                        833 ;   (1 BYTE) - CONTROL BYTE (DRIVE STEP OPTION (1
                              BYTE)—控制字节(驱动器步进选择)
                        834 ;           BIT 7 DISABLE DISK-ACCESS RETRIES 位
                                7 禁止重新访问硬盘机
                        835 ;           BIT 6 DISABLE ECC RETRIES 位 6 禁止重
```

```
                    试 ECC
836 ;               BITS 5-3 ZERO; 位 5-3 0
837 ;               BITS 2-0 DRIVE OPTION  位 2-0 驱动器号
838 ;  (1 BYTE) - STANDARD TIME OUT VALUE (SEE BELOW); (1 BYTE) 标
                                                    准超时值
839 ;  (1 BYTE) - TIME OUT VALUE FOR FORMAT ORIVE; (1 BYTE)格式化驱
                                                    动器用超时值
840 ;  (1 BYTE) - TIME OUT VALUE FOR CHECK DRIVE; (1 BYTE)校验驱动器
                                                    用超时值
841 ;  (4 BYTES); (4BYTE)保留
842 ;           - RESERVED FOR FUTURE USE
843 ;
844 ;     TO DYNAMICALLY DEFINE A SET OF PARAMETERS; 动态生成参数
                                                    块的方法:先在内存建立这个参数块
845 ;     BUILD A TABLE OF VALUES AND PLACE THE; 然后将第 41 号中断
                                                    向量改成该参数块的首址
846 ;     CORRESPONDING VECTOR INTO INTERRUPT 41; 首址. 缺省参数块用
                                                    于自启动(第 19 号中断).
847 ;
848 ; NOTE;
849 ;       THE DEFAULT TABLE IS VECTORED IN FOR;
850 ;       AN INTERRUPT 19H(BOOTSTRAP);
851 ;
852 ;
853 ; ON THE CAPD SWITCH SETTINGS;
854 ;
855 ;          DRIVE 0    DRIVE I;
856 ;        ..........................................
857 ;   ON   :             /          :
858 ;          —1— —2—/—3— —4— :
859 ;   OFF  :             /          :
860 ;        ..........................................
861 ;
862 ;
863 ; TRANSLATION TABLE
864 ;
865 ;     1/3    2/4     TABLE ENTRY
866 ; ..................................................
867 ;     ON  ;  ON  ;        0        ;
```

```
                868 ;    ON     OFF    1                    :
                869 ;    OFF     ON    2                    :
                870 ;    OFF    OFF    3                    :
                871 ;                                       :
                872 ;                                       :
                873 ; ..............................................
03E7            874 FD_TBL: 硬盘参数表
                875
                876 ; ……DRIVE TYPE 00     驱动器类型 00(2 磁头)
                877
03E7  3201      878         DW 03060
03E9  02        879         DB 02D
03EA  3201      880         DW 030FD
03EC  0000      881         DW 00000
03EE  0B        882         DB 0BH
03EF  00        883         DB 00H
03F0  0C        884         D8 0CH      ; STANOABD
03F1  B4        885         DB 0B4H     ; FORMAT DRIVE
03F2  28        886         DB 02BH     ; CHECK DRIVE
03F3 00000000   887         DB 0, 0, 0, 0
                888
                889 ; ……DRIVE TYPE 01   驱动器类型 01(8 磁头)
                890
03F7 7701       891         DW 0375D
03F9 08         892         DB 08D
03FA 7701       893         DW 0375D
03FC 0000       894         DW 00000
03FE 08         895         DB 0BH
03FF 05         896         DB 05H
0400 0C         897         DB 0CH      ; STHAUARD
0401 B4         898         DB 0B4H     ; FORMAT DRIVE
0402 28         899         DB 028H     ; CHECK DRIVE
0403 00000000   900         DB 0, 8, 0, 0
                901
                902 ; ……DRIVE TYPE 02   驱动器类型 02(6 磁头)
                903
0407 3201       904         DW 0306D
0409 06         905         DB 06D
040A 8000       906         DW 012BD
```

```
040C 0001          907     DW      0256D
040E 0B            908     DB      0BH
040F 05            909     DB      05H
0410 0C            910     DB      0CH     ; STANDARD
0411 B4            911     DB      0B4H    ; FORMAT DRIVE
0412 28            912     DB      028H    ; CHECK DRIVE
0413 000000000     913     DB      0, 0, 0, 0
                   914
                   915 ; ……DRIVE TYPE 03  驱动器类型 03(4 磁头)
                   916
0417 3201          917     OW      0306D
0419 04            918     DB      04D
041A 3201          919     DW      0306D
041C 0000          920     DW      00000
041E 0B            921     DB      0BH
041F 05            922     DB      05H
0420 0C            923     DB      0CH     ; STANDARD
0421 B4            924     DB      0B4H    ; FORMAT DRIVE
0422 28            925     DB      028H    ; CHECK DRIVE
0423 00000000      926     DB      0, 0, 0, 0
                   927
0427               928     INIT_DRV PROC HEAR  初始驱动器程序
                   929
                   930 ; ……DO DRIVE ZERO
                   931
0427 C60642000C    932     MOV     CMO_BLOCK+0, INIT_DRV_CMD  先把驱动
                                                              器看成 0 号的(偶数)
042C C606430000    933     MOV     CMO_BLOCK+1, 0  送命令全码及置 d=0
0431 E81000        934     CALL    INIT_DRV_R  初始
0434 7200          935     JC      INIT_DRV_OUT  初始成功转 INIT_DRV_OUT
                   936
                   937 ; ……DO DRIVE ONE
                   938
0436 C60642000C    939     MOV     CMO_BLOCK+0, INIT_DRV_CMD  驱动器是 1
                                                              号的(奇数)
0438 C606430020    940     MOV     CMD_BLOCK+1, 00100000B
0440 E80100        941     CALL    INIT_DRV_R  初始
0443               942     INIT_DRV_DUT:
0443 C3            943             RET
```

```
                944 INIT_DRV ENDP
                945
0444            946 INIT_DRV_R PROC NEAR
                947     ASSUME EC:CODE
0444 2AC0       948     SUB     AL, AL
0446 E81901     949     CALL    COMMAND ; ISSUE THE COMMAND 发出命
                                          令
0449 7301       950     JNC     B1
044B C3         951     RET
044C            952 B1:
044C 1E         953     PUSH    DS      ; SAVB SEGMENT
                954     ASSUME  DS:DUMMY  DS 指向数据段 DUMMY 准备取参
                                          数表向量
044D 2BC0       955     SUB     AX, AX
044F 8ED8       956     MOV     DS, AX  ; ESTABLESH SEGMENT
0451 C41E0401   957     LES     BX, HF_TBL_VMC 用 ES 建立一个以 HF_
                                          TBL_VEC 内容为始址的参数数据段
0455 1F         958     POP     DS      ; RESTORE SEGMFNT
                959     ASSUME  DS:DATA
0456 E83403     960     CALL    SW2-OFFS 取驱动器 I/O 口的偏移
0459 7257       961     JC      B3
045B 0308       962     ADD     BX, AX
                963
                964 ; ……SEND DRIVE PARAMETERS MOST SIGNIFICANT
                        BYTE FIRST 传送驱动器特征参数，从参数表首以字节为单位
                        传送，共 9 个字节
                965
0450 BF0100     966     MOV     DI, 1                  字节 1(柱面号)
0460 E85F00     967     CALL    INIT_DRV_S
0463 724D       968     SC      B3
                969
0465 BF0000     970     MOV     DI, 0                  字节 2(柱面号)
0468 E85700     971     CALL    INIT_DRV_S
046B 7245       972     JC      B3
                973
046D BF0200     974     MOV     DI, 2                  字节 3(磁头个数)
0470 E84F00     975     CALL    INIT_DRV_S
0473 723D       976     JC      B3
                977
0475 BF0400     978     MOV     DI, 4                  字节 4(启始约减写当前柱面)
```

```
0478 E84700      979      CALL    INIT_DRV_S
047B 7235        980      JC      B3
                 981
047D BF0300      982      MOV     DI, 3              字节 5(启始约减写当前柱面)
0480 E83F00      983      CALL    INIT_DRV_S
0483 722D        984      JC      B3
                 985
0485 BF0600      986      MOV     DI, 6              字节 6(启始写予补偿柱面)
0488 E83700      987      CALL    INIT_DRV_S
0488 7225        988      JC      B3

                 989
048D BF0500      990      MOV     DI, 5              字节 7(启始写予补偿柱面)
0490 E82F00      991      CALL    INIT_URV_S
0493 721D        992      JC      B3
                 993
0495 BF0700      994      MOV     DI, 7          字节 8(最大 ECC 数据突发长度)
0498 E82700      995      CALL    INIT_DRV_S
0498 7215        996      JC      B3
                 997
0490 BF0800      998      MOV     DI, 8                  ; DRIVE STEP OP-
                                                         TION 字节 9 (控
                                                         制字节)
04AD268A01       999      MOV     AL, ES; (BX + DI)
04A3 A27600     1000      MOV     CONTROL_BYTE, AL
                1001
04A6 2BC9       1002      SUB     CX, CX                 计时器初始化
                1003 B5:
04A8 E8D302     1004      CALL    PORT_1  指向口子 1
04ABEC          1005      IN      AL, DX
04ACAB02        1006      TEST  AL, R1_IOMODE        ; STATUS INPUT
                                                         HODE 判驱动器是
                                                         否能向主机返回结果
                                                         了
04AE7509        1007      JNZ   B6
04B0 E2F6       1008      LOOP  B5                   不能返回则等待
04B2            1009 B3:
04B2C606740007  1010      MOV     DISK_STATUS, INIT_FAIL; OPERATION FA-
                                                         ILED 初始失败标
```

```
                                        记，超过归定时间，驱动器不能返
                                        回结果
04B7 F9          1011      STC
04B8 C3          1012      RET   返回
                 1013
0489             1014 B6:
04B9 E88502      1015      CALL      PORT_0          指向口子 0
04BC EC          1016      IN        AL, DX
04BD 2402        1017      AND       AL, 2           ; MASK ERROR BIT  取结果状
                                                     态字节，第 1 位为 1 表示有错误
04BF 75F1        1018      JNZ       B3
04C1 C3          1019      RET                       返回
                 1020      ASSUME ES:NOTHING
                 1021 INIT_ORV_R EMEP
                 1022
                 1023 ; SENO THE BYTE OUT TO THE CONTROLLER  下面程序
                          将ES:[BX+DI]指定的字节从口子 0 输出。
                 1024
04C2             1025 INIT_ORV_S PROC NEAR
04C2 E8CS01      1026      CALL      HD_WAIT_REQ  等待驱动器发出请求继续输
                                                  入参数的信号
04C5 7207        1027      JC        D1
04C7 E8A702      1028      CALL      PORT_0            取 0 号口子地址，送 DX
04CA 26BA01      1029      MOV       AL, ES:[BX+DI)    取字节
04CD EE          1030      CUT       DX, AL            输出
04CE             1031      D1:
04CE C3          1032      RET
                 1033 INIT_DRV_S ENDP              INIT_DRV_S 结束
                 1034
                 1035 ;
                 1036 ; READ LONG (AH=0AH)  (AH)=0AH，长读
                 1037 ;
                 1038
04CF             1039 RD_LONG PROC NEAR
04CF E81900      1040      CALL      CHK_LONG   块计数器内容>128?
04D2 726B        1041      JC        G8         大于转 G8
04D4C6064200ES   1042      MOV       CMD_BLOCK+0, RO_LONG_CMD  送长读
                                                  命令
04D9 B047        1043      MOV       AL, DMA_READ
04DBEB68         1044 JMP      SHORT DMA_OPN     转 DMA_OPN
```

```
                1045 RD__LDNG ENDP
                1046
                1047 ;
                1048 ; WRITE LONG (AH=08H)  (AH)=OBH; 长写
                1049 ;
                1050
04D0            1051 WR__LONG FROC HEAR
04D0 E80B00     1052 CALL   CHK__LONG 块计数器内容>128?
04E0 725D       1053 JC     G8  大于转 G8
04E2 C6064200E6 1054 MOV    CMD__BLOCK+0, WR__LONG__CMD__送长写命令
04E7 B04B       1055 MOV    AL, DMA__WRITE
04E9 EB5A       1056 JMP    SHORT DMA__OPN  转 DMA__OPN
                1057 WR__LONG ENDP
                1058
04EB            1059 CHK__LONG PROC NEAR  判决计数器内容是否大于128,也即
                     传送字节数会大于
04EB A04600     1060 MOV    AL, CMD__BLDCK+4  64KB 否,大于出错。
04EE 3C80       1061 CMP    AL, 080H
04F0 FS         1062 CMC  大于 128 时 C7=1
04F1 C3         1063 RET
                1064 CHK__LONG ENDP
                1065


                1966 ; ……
                1967 ; SEEK(AH=DCH; (AH)=OCH; 寻找
                1968 ; ……
                1069
04F2            1979 DISK__SEEK PROC HEAR
04F2 C404420008 1071 MOV    CMD__SLOCK, SEEK__CMD  送寻找命令
04F7 EB30       1072 JMP    SHORT HDMA__OIN  转 HDMA__OPN
                1073 DISK__SEEK ENOP
                1974
                1975 ; ……
                1976 ; READ SECTOR SUFFER (AH=OEH); (AH)=OEH;读扇区
                     缓存
                1977 ; ……
                1978
04F9            1979 RO__BUFF PROC NEAR
04O9 C60642000E 1980 MOV    CMD__BLOCK+0, RD__BUFF__CMD  送命令
```

```
04FE C606460001   1081 MOV  CMD_BLOCK+4,1; ONLY ONE BLOCK  读1个块
0503 B047         1082 MOV  AL, DMA_READ
0505 EC3E         1083 JMP  SHORT DMA_OPN  转 DMA_OPN,(AL)为 DMA 读
                       模式字节
                  1084 RD_BUFF EMOP
                  1085
                  1086 ; ……
                  1087 ; WRITE SECTOR BUFFER (AM=OFH); (AH)=OFH; 写
                       扇区缓存
                  1088 ; ……
                  1089
0507              1090 WR_BUFF PROC HEAR
0507 C60642000F   1091 MOV  CMD_BLOCK+0, WR_BUFF_CMD  送命令
050C C606460001   1092 MOV  CMD_BLOCK+4, 1 ; ONLY ONE BLOCK  写一个块
0511 B04B         1093 MOV  AL, DMA_WRITE
0513 EB30         1094 JMP  SHORT DHA_OPN  转 DMA_OPN, (AL)为 DMA 写
                       模式字节
                  1095 WR_BUFF ENDP
                  1096
                  1097 ; ……
                  1098 ; TEST DISX READY(AH=010H); (AH)=010H 测试硬盘
                       机准备好否
                  1099 ; ……
                  1100
0515              1101 TST_ROY PROC NEAR
0515 C606420000   1102 MOV  CMO_BLDCK+0, TST_ROY_CMD  送命令
051A EBIA         1103 JMP  SHORT NDMA_OPN  转 HDMA_OPN
                  1104 TST-ROY ENDP
                  1105
                  1106 ; ……
                  1107 ; RECALIBRATE (AH=011H); (AH)=011H; 重校
                  1108 ; ……
                  1109
051C              1110 HDISK_RECAL PROC HEAR
051C C606420001   1111 MOV  CMD_BLOCK, RECAL_CMO  送命令
0521 EB13         1112 JMP  SHORT NDMA_OPN  转 NDMA_OPN
                  1113 HDISK_RECAL ENDP
                  1114
                  1115 ; ……
```

```
                        1116 ; CONTROLLER RAM DIAGNOSTICS (AH=012H): (AH)
                               =012：控制器 DMA 诊断
                        1117 ; ..........
                        1118
0523                    1119 RAM_DIAG PROC NEAR
0523 C6064200E0         1120 MOV  CMD_BLOCK+0, RAM_DIAG_CMD  送命令
0528 EB0C               1121 JMP  SHORT NDMA_OPN  转 HDMA_OPN
                        1122 RAM_DIAG ENDP
                        1123
                        1124 ; ..........
                        1125 ; DRIVE DIAGNOSTICS (AH=013H); (AH) = 013H，驱动
                               器诊断
                        1126 ; ..........
                        1127
052A                    1128 CHK_DRV PROC NEAR
052A C6064200E3         1129 MOV  CMD_BLOCK+0, CHK_DRV_CMD  送命令
052F EB05               1130 JMP  SHORT NDMA_OPN  转 HDMA_OPN
                        1131 CHK_DRV ENDP
                        1132
                        1133 ; ..........
                        1134 ; CONTROLLER INTERNAL DIAGNOSTICS (AH=014H):
                               (AH) = 014H；控制器内部诊断
                        1135 ; ..........
                        1136
0531                    1137 CNTLR_DIAG PROC NEAR
0531 C60642000E4        1138 MOV  CMD_BLDCK+0, CNTLR_DIAG_CMD  送命令
                        1139 CNTLR_DIAG ENDP
                        1140


                        1141 ; ..........
                        1142 ; SUPPORT ROUTINES：支持程序
                        1143 ; ..........
                        1144
0536                    1145 HDMA_OPN
0536 B002               1146 MOV  AL, 02H
0538 E82700             1147 CALL CDMMAND；ISSUE TNE COMMAND  送出命令
053B 7221               1148 JC   G11
0530 EB16               1149 JMP  SHORT G3  COMMAND 无错转 G3
053F                    1150 G8:
```

```
054F C606740009   1151 MOV  DISK_STATUS, DMA_BOUNDARY  DMA 负越出 64
                                KB 范围错
0544 C3           1152       RET
0545              1153 DMA_OPN;
0545 E85701       1154       CALL  DMA_SETUP ; SET UP FOR DMA OPERATION
                                                建立 DMA
0548 72F5         1155       JC    G8
054A B003         1156       MOV   AL, 03H  设屏蔽字节,送出命令块
054C E81300       1157       CALL  COMMANO   ; ISSUE THE COMMAND
054F 720D         1158       JC    G11
0551 B003         1159       MOV   AL, 03H  COMMAND 程序无错
0553 E60A         1160       OUT   DMA+10, AL ; INITIALIZE THE DISK CHAN-
                                                NEL  初始 DMA 通道
0555              1161 G3;
0555 E421         1162       IN    AL, 021N
0557 24DF         1163       AND   AL, 0DFH
0559 E621         1164       OUT   021H, AL
0558 E8AA01       1165       CALL  WAIT_INT  等待驱动器命令执行定后发来中断
055E              1166 G11;
055E E83B00       1167       CALL  ERROR_CHK  检查是否有错误发生
0561 C3           1168       RET
                  1169
                  1170 ; ..........
                  1171 ; COMMAND  COMMAND
                  1172 ; THIS ROUTINE OUTPUTS THE COMMAND BLOCK; 送出命
                         令块
                  1173 ; INPUT
                  1174 ; AL=CONTROLLER,DMA/INTERRUPT REGISTER MASK; 入
                         口参数 AL=控制器 DMA/中断寄存和屏蔽字节
                  1175 ;
                  1176 ; ..........
                  1177
0562              1178 COMMAND PROC NEAR
0562 BE4200       1179       MOV   SI, OFFSET CMD_BLOCK
0565 E81B02       1180       CALL  PORT_2
0568 EE           1181       OUT   DX, AL    ; CONTROLLER SELECT PULSE
                                                输出(AL)内容控制器选择脉冲,口
                                                子,
0569 E81C02       1182       CALL  PORT_3
056C EE           1183       OUT   DX, AL  送出控制器 DMA/中断寄存器屏蔽字节,口
```

```
056D 2BC9          1184        SUB     CX, CX 循环次数; WAIT COUNT
056F E80C02        1685        CALL    PORT_1
0572               1186 WAIT_BUSY
0572 EC            1187        IN      AL, DX   ; FET STATUS  取驱动器状态
0573 240F          1188        AND     AL, 0FH
0575 3C0D          1189        CMP     AL, RI_BUSY DR RI_BUS OR RI_REQ 驱
                                       动器能接收命令块了(忙位,总线位请求位全 0?)
0577 7409          1190        JE      CI
0579 E2F7          1191        LDOP    WAIT_BUSY
057B C606740080    1192        MOV     DISK_STATUS, TIME_OUT  循环等待
0580 F9            1193        STC
0581 C3            1194        RET                 ; ERROR RETURN  超时错, 返回
0582               1195 C1:
0582 FC            1196        CLD  正向传送标志
0583 B90600        1197        MOV     CX, 6       ; BYTE COUNT  命令块最多占六
                                       个字节.
0586               1198 CM3:
0586 EBE801        1199        CALL    PORT_0
0589 AC            1200        LODSB               ; GET THE NEXT COMMAND
                                       BYTE  用 LODSB OUT 两条指令
                                       送出命令字节
058A EE            1201        OUT     DX, AL      ; OUT IT GOES
0588 E2F9          1202        LDOP    CM3         ; DO MORE  循环六次
                   1203
058D E8EE01        1204        CALL    PORT_1      ; STATUS  指向口子 1
0590 EC            1205        IN      AL, DX
0591 A801          1206        TEST    AL, RI_REQ  判驱动器是否还在请求命令字节
0593 7406          1207        JZ      CM7
0595 C606740020    1208        MOV     DISK_STATUS, BAD_CNTLR  是则控制器错
059A F9            1209        STC
0598               1210 CM7:
0598 C3            1211        RET
                   1212 COMMAND ENDP  COMMANO 返回
                   1213
                   1214 ; ..........................................................
                   1215 ; SENSE STATUS BYTES: 检测字节(四字节).
                   1216 ;
                   1217 ; BYTE 0  字节.
```

```
                1218 ; BIT   7  ADDRESS VALID, WHEN SET; 位 7:置 1 时地址有
                                                          效
                1219 ; BIT   6  SPARE, SET TO ZERO; 位 6: 置.
                1220 ; BITS5-4 ERROR TYPE; 位 5-4: 错误类型
                1221 ; BITS3-0 ERROR CODE; 位 3-0: 错误码
                1222
                1223 ; BYTE 1; 字节,
                1224 ; BITS 7-6 ZERO; 位 7-6: 置.
                1225 ; BIT      SDRIVE(0-1): 位 S: 驱动器标识(0 或 1)
                1226 ; BITS 4-0 MEAD NUMBER; 位 4-0: 磁头号
                1227 ;
                1228 ; BYTE 2; 字节
                1229 ;       BITS 7-5 CYLINDER HIGH; 位 7-5: 柱面号高 2 位
                1230 ;       BITS 4-0 SECTOR NUMBER; 位 4-0: 扇区号
                1231 ;
                1232 ; BYTE 3; 字节 3
                1233 ; BITS 7-0 CYLINDER LOW; 位 7-0: 柱面号低 8 位
                1234 ;
                1235 ; ..........................................................
                1236
059C            1237 ERROR_CHK PROC NEAR  结果检查程序 ERROR_CHK
                1238       ASSUME  ES:DATA
057C A07400     1239       MOV     AL, DISK_STATUS ; CHECK IF THERE WAS
                                                       AN ERROR DISK_
                                                       STATUS 为 0 说明无
                                                       错
059F 0AC0       1240       OR      AL, AL
05A1 7501       1241       JNG     Z21
05A3 C3         1242       RET     无错误.
                1243
                1244 ; PERFORM SENSE STATUS  有错误, 地址控制器返回 4 个检测
                       字节, 他们能反映出错的具体情形.
                1245
05A4            1246 G21;
05A4 B84000     1247       MOU     AX, DATA
05A7 BEC0       1248       MOV     ES, AX          ; ESTABLISH SEGMENT
                                                     建立命令块数据段
05A9 2BC0       1249       SUB     AX, AX
05AB 8BFB       1250       MOV     DI, AX
```

```
05AD C606420003   1251          MOV     CMD_BLOCK+0,SENSE_CMD  建立检测命
                                          令的命令块
05B2 2AC0         1252          SUB     AL,AL
05B4 E8ABFF       1253          CALL    COMMAND                ;ISSUE SENSE ST-
                                                                 ATUS COMMAND
                                                                 送出回送检测字节的
                                                                 命令，其第0第1字
                                                                 节有效
05B7 7223         1254          JC      SENSE_ABORT            ;CANNOT RECOVER
                                                                 检测命令发送失败转
                                                                 SENSE_ABORT
0589 B90400       1255          MOV     CX,4  成功,建立计数器准备取检测字节
05BC              1256 G22:
05BC E8CB00_      1257          CALL    HD_WAIT_REQ  等待控制器执行完检测命令
                                          或输出有效的下一个字节
058F 7220         1258          JC      G24  超时错转 G24
05C1 E8AD01       1259          CALL    PORT_0
05C4 EC           1260          IN      AL,DX  指向 I/O 口 1,取检测字节
05C5 26884542     1261          MOV     ES:HD_ERROR(DI),AL;    STORE ANAY SE-
                                                                 NSE BYTES  送HD
                                                                 _ERROR 为首址的
                                                                 字节区
05C9 47           1262          INC     DI
05CA E88101       1263          CALL    PORT_1
05CD E2ED         1264          LOOP    G22  循环取4个检测字节
05CF E8B800       1265          CALL    HD_WAIT_REQ
05D2 720D         1266          JC      G24
05D4 E89A01       1267          CALL    PORT_0  最后取检测命令的状态字节，第1位
                                          为0表示无误
05D7 EC           1268          IN      AL,DX
05D8 A802         1269          TEST    AL,2
05DA 740F         1270          JZ      STAT_ERR
05DC              1271 SENSE_ABDRT:
05DC C606740DFF   1272          MOV     DISK_STATUS,SENSE_FAIL  检测命令执
                                          行失败,置CY为1,返回
05E1              1273 G24:
05E1 F9           1274          STC
05E2 C3           1275          RET
                  1276 ERROR_CHK ENDP
```

```
                        1277
05E3 1A06               1278 T_0 DW         TYPE_0  回送当前操作状态程序的地址散
                                            转表
05E5 2706               1279 T_1 DW         TYPE_1
05E7 6A06               1280 T_2 DW         TYPE_2
05E9 7706               1281 T_3 DW         TYPE_3
                        1282
05EB                    1283 STAT_ERR:
05EB 268A1E4200         1284        MOV     BL, ES:HD_ERROR ; GET ERROR BY-
                                                              TE 取首检测字节
05F0 8AC3               1285        MOV     AL, BL
05F2 240F               1286        AND     AL, 0FH
05F4 80E330             1287        AND     BL, 00110000B       ; ISOLATF TYPE
                                                                  留下类型码
05F7 2AFF               1288        SUB     BH, BH
05F9 B103               1289        MOV     CL, 3
05FB D3EB               1290        SHR     BX, CL              ; ADJUST 调整到
                                                                  2-1 位
05FD 2EFFA7E305         1291        JMP     WORD PTR CS:(BX+OFFSET T_0) 根
                                            据类型码散转
                        1292        ASSUME ES:NOTHING
                        1293
0602                    1294 TYPED_TABLE LABEL BYTE  当前操作状态表(类型 0)

0602 00204020800029     1295        DB      0, BAD_CNTLR, BAD_SEEX,
                                            BAD_CNTLR, TIME_OUT, D,
                                            BAD_CNTLR
0609 0040               1296        DB      0, BAD_SEEK
0009                    1297        TYPED_LEN   EQU S_TYPED_TABLE
0608                    1298        TYPEI_TABLE LABEL BYTE  当前操作状态表(类
                                                            型 1)
0608 1010020004         1299        DB      BAD_ECC, BAD_ECC, BAD_
                                            ADDR_MARK, 0, RECORD_NOT
                                            _FND
0610 4000001108         1300        DB      BAD_SEEK, 0, 0, DATA_CORRE
                                            CTEO, BAD_TRACK
000A                    1301        TYPEI_LEN   EQU S_TYPEI_TABLE
0615                    1302        TYPE2_TABLE LABEL BYTE  当前操作状态表(类
                                                            型 2)
```

```
0615 0102          1303      DB              BAD_CMD, BAD_ADDR_MARK
0002               1304 TYPE2_LEN    EQU S_TYPE2_TABLE
0617               1305 TYPE3_TABLE LABEL BYTE  当前操作状态表(类型3)
0617 202010        1306 DB           BAD_CNTLR, BAD_CNTLR, BAD_ECC
0003               1307 TYPE3_LEN    EQU S_TYPE3_TABLE
                   1308
                   1309 ; ……TYPE 0 ERROR  错误类型0
                   1310
061A               1311 TYPE_0:
061A BB0206        1312 MOV    BX, OFFSET TYPE0_TABLE  取状态表始址
061D 3C09          1313 CMP    AL, TYPED_LEN      ; CNECK IF ERROR IS
                                                    DEFINED 错误码个数不
                                                    能大于9
061F 7363          1314 JAE    UNDEF_ERR_L
0621 2ED7          1315 XLAT   CS:TYPE0_TABLE     ; TABLE LDOKUP 取回状
                                                    态码,由 AL 返回
0623 A27400        1316 MOV    DISK_STATUS, AL    ; SET ERROR CODE
0626 C3            1317 RET
                   1318
                   1319 ; ……TYPE 1 ERROR  错误类型1:
                   1320
0627               1321 TYPE_1:
0627 B80806        1322 MOV    BX, OFFSET TYPE1_TABLE
062A 8BC8          1323 MOV    CX, AX
062C 3C0A          1324 CMP    AL, TYPE1_LEN      ; CHECK IF ERROR IS
                                                    DEFINED 错误个数不能
                                                    大于10
062E 7354          1325 JAE    UNDEF_ERN_L
0630 2ED7          1326 XLAT   CS:TYPE1_TABLE     ; TABLE LDOKUP
0632 A27400        1327 MOV    DISK_STATUS, AL    ; SET ERROR CODE 取回
                                                    状态码交 AL 返回
0635 80E108        1328 AND    CL, 8H             ; CORRECTED ECC 是已
                                                    纠正了的 ECC 数据错?
0638 80F908        1329 CMP    CL, 08H
0638 752A          1330 JNZ    G30
                   1331
                   1332 ; ……DBTAIN ECC ERROR BURST LENGTH 是已纠正了
                          的 ECC 数据错
                   1333
0630 C606420000    1334 MOV    CMD_BLOCK+0, RD_ECC_CMD
```

```
0642 2AC0          1335        SUB     AL, AL
0644 E81BFF        1336        CALL    COMMAND  发出读取ECC突发错误长度的命令,
                                       让(AL)返回突发错误的长度
0647 721E          1337        JC      G30
0649 E83E00        1338        CALL    HD_WAIT_REQ
064C 7219          1339        JC      G30
064E E82001        1340        CALL    PORT_0  读突发长度
0651 EC            1341        IN      AL, DX
0652 8AC8          1342        MOV     CL, AL
0654 E83300        1343        CALL    HD_WAIT_REQ
0657 720E          1344        JC      G30
0659 E81501        1345        CALL    PORT_0  读I/O口0,检查读取命令执行正确与否
065C EC            1346        IN      AL, DX
065D A801          1347        TEST    AL, 01H
065F 7406          1348        JZ      G30
0661 C606740020    1349        MOV     DISK_STATUS, BAD_CNTLR
0666 F9            1350        STC
0667               1351 G30:
0667 8AC1          1352        MOV     AL, CL  突发错长度或状态码送 AL
0669 C3            1353        RET             返回
                   1354
                   1355 ; ……TYPE 2 ERROR  错误类型2:
                   1356
066A               1357 TYPE_2:
066A BB1506        1358        MOV     BX, OFFSET TYPE2_TABLE
066D 3C02          1359        CMP     AL, TYPE2_LEN    ; CHECK IF ERROR IS
                                                            DEFINED 错误个数不能
                                                            大于2
066F 7313          1360        JAE     UNDEF_ERR_L
0671 2ED7          1361        XLAT    CS:TYPE1_TABLE   ; TABLE LOOKUP  取回
                                                            状态码
0673 A27400        1362        MOV     DISK_STATUS, AL  ; SET ERROR CODE
0676 C3            1363        RET
                   1364
                   1365 ; ……TYPE 3 ERROR  错误类型3
                   1366
0677               1367 TYPE_3
0677 881704        1368        MOV1    BX, OFFSET TYPE3_TABLE
067A 3C0           1369      CMP     AL, TYPE3_LEN
```

```
067C 7306          1370    SAE    UNDEF_ERR_L  错误个数不能大于3
067E 2E07          1371    XLAT   CS:TYPE3_TABLE

0680 A27400        1372    MOV    DISK_STATUS, AL  取回状态码
0683 C3            1373    RET
                   1374
0684               1375 UNDEF_ERR_L:
0684 C6067400BB    1376    MOV    DISK_STATUS, UNDEF_ERR  错误类型非法
0689 C3            1377    RET
                   1378
068A               1379 HD_WAIT_REQ PROC NEAR  等待驱动器发出可以输入下一
                        个参数或能输出下一字节的信号
068A 51            1380    PUSH   CX
0688 28C9          1381    SOB    CX, CX  循环次数
068D E8EE00        1382    CALL   PORT_1  指向口子
0690               1383 L1:
0690 EC            1384    IH     AL, DX
0691 A801          1385    TEST   AL, R1_REQ  请求位为1?
0693 7508          1386    JHZ    L2
0695 E2F9          1387    LOOP   L1  不为1循环等待或循环次数满,出错
0697 C606740080    1388    MOV    DISK_STATUS, TIME_OUT
069C F9            1389    STC
0690               1390 L2:
0690 59            1391    POP    CX  请求位为1返回
069E C3            1392    RET
                   1393    HD_WAIT_REQ ENDP
                   1394
                   1395 ;
                   1396 ;              DMA_SETUP  建立DMA程序(DMA_SETUP)
                   1397 ; THIS ROUTINE SETS UP FOR OMA OPERATIONS:  入口
                        参数:
                   1398 ;                                  INPUT: (AL)=DMA模式字节
                   1399 ; (AL)=MOOE BYTE FOR THE DMA: (ES: BX)=DMA传
                        送始址
                   1400 ; (ES:BX)=ADDRESS TO READ/WRITE THE DATA:
                   1401 ;                                  DUTPUT: 出口参数
                   1402 ;                          (AX)DESTROYED: (AX)被破坏
                   1403 ;.....
069F               1404 DMA_SETUP PROC NEAR
```

```
069F 50        1405        PUSH   AX
06A0 A04600    1406        MOV    AL, CMD_BLOCK+4
06A3 3C81      1407        CMP    AL, 81H          ; BLOCK COUNT DUT OF RAN-
                                                    GE 块是否超过额定数目?(最
                                                    多64K/512=128块)
06A5 58        1408        PCP    AX
06A6 7202      1409        JB     J1
06A8 F9        1410        STC
06A9 C3        1411        RET    块数大于128,返回
06AA           1412 J1:
06AA 51        1413        PUSH   CX               ; SAVE THE REGISTER
06ABFA         1414        CLI                     ; NO MORE INTERRUPTS
06ACE60C       1415        OUT    DMA+12, AL       ; SET THE FIRST/LAST F/F
                                                    清字节指示器触发器
06AE50         1416        PUSH   AX
06AF58         1417        POP    AX
06B0 E608      1418        OUT    DMA+11, AL       ; OUTPUT THE MODE BYTE
                                                    送DMA模式字节,选用DMA
                                                    第3通道
06B2 BCC0      1419        MOV    AX, ES           ; SET THE ES VALUE
06B4 B104      1420        MOV    CL, 4            ; SHIFT COUNT
06B6 D3C0      1421        ROL    AX, CL           ; ROTATE LEFT ES 内容(其低
                                                    4位有效)移至高4位
06B8 8AE8      1422        MOV    CH, AL           ; GET HIGHEST NYBBLE OF ES
                                                    TO CH
06BA 24F0      1423        AND    AL, OFOH         ; ZERO THE LOW NYBBLE FR-
                                                    OM SEGMENT
06BC 03C3      1424        ADD    AX, BX           ; TEST FOR CARRY FROM
                                                    ADDITION 段寄存器与段内偏
                                                    差寄存器内容相加得内存绝对地
                                                    址
06BE 7302      1425        JNC    J33
06C0 FEC5      1426        INC    CH               ; CARRY MEANS HIGH 4 BITS
                                                    MUST BE INC (CN)为高4位
                                                    (第16—19位)内容
06C2           1427 J33:
06C2 50        1428        PUSH   AX               ; SAVE START ADDRESS
06C3 E606      1429        OUT    DMA+6, AL        ; DUTPUT LOW ADDRESS 低
                                                    地址送给第3通道基址寄存器
```

```
06C5 8AC4      1430    MOV  AL, AH
06C7 E606      1431    OUT  DMA+6, AL          ; OUTPUT HIGH ADDRESS 高
                                                 地址送给第3通基址寄状器
06C9 8AC5      1432    MOV  AL, CH             ; GET RIGH 4 BITS
06CB 240F      1433    AND  AL, OFH
06CD E682      1434    OUT  DMA__HIGH, AL      ; OUTPUT THE HIGH 4 BITS
                                                 TO PAGE REG 高4位地址送
                                                 出 DMA__HIGH = 0082
               1435
               1436 ; ……DETERMINE COUNT
               1437
06CF A04600    1438    MOV  AL, CMD__BLOCK+4 ; RECOVER BLOCK COUNT
06D2 D0E0      1439    SHL  AL, 1              ; MULTIPLY BY 512 BYTES PER
                                                 SECTOR 块数换算成字节数
06D4 FECB      1440    0DE  AL                 ; AND DECREMENT VALUE BY
                                                 DNE 字节数的低8位置全，即
                                                 至少传512个字节，且以256为
                                                 模
06D6 BAED      1441    MOV  AH, AL
06D8 BOFF      1442    MOV  AL, OFFH
               1443
               1444 ; ……HANDLE READ AND WRITE LONG (516 D BYTE
                       BLOCKS)判是长读/长写操作?
               1445
06DA 50        1446    PUSH AX                 ; SAVE REGISTER
06DB A04200    1447    MOV  AL, CMD__BLOCK+0; GET COMMAND
06DE 3CE5      1448    CMP  AL, RD__LONG__CMD  是长读?
06E0 7407      1449 JE      ADD4
06E2 3CE6      1450         CMP   AL, WR__LONG__CMD  是长写?
06E4 7403      1451         JE    ADD4
06E6 58        1452         POP   AX          ; RESTORE REGISTER 非长读
                                                 长写转J20
06E7 EB11      1453         JMP   SHORT J20
06E9           1454 ADD4:
06E9 58        1455         POP   AX          ; RESTORE REGISTER
06EAB 80402    1456         MOV   AX, 516D    ; DNE BLOCK(512)PLUS 4 BYT-
                                                 ES ECC 一块占512+4 ECC 共
                                                 516字节
06ED 53        1457         PUSH  BX
```

```
06EE 2AFF       1458        SUB     BH, BH
06F0 8A1E4600   1459        MOV     BL, CMD—BLOCK+4  取块数
06F4 52         1460        PUSH    DX
06F5 F7E3       1461        MUL     BX          ; BLDCK COUNT TIMES 516  作
                                                  块数与516的积,结果放在AX内
06F7 5A         1462        POP     DX
06F8 58         1463        POP     BX
06F9 48         1464        DEC     AX          ; ADJUST  块数X516-1这是DMA
                                                  传送的字节计算初始值
06FA            1465 J20:
                1466
06FA 50         1467        PUSH    AX          ; SAVE COUNT VALUE
06FB E60        1468        DUT     DMA+7, AL; LOW BYTE OF COUNT
06FD 8AC4       1469        MOV     AL, AH
06FF E607       1470        OUT     DMA+7, AL; HIGH BYTE OF COUNT  送出
                                                  DMA 字节计算初始值
0701 FB         1471        STI                 ; INTERRUPTS BACK ON  开中
0702 59         1472        POP     CX          ; RECOVER COUNT VALUE
                                                  (CX)为 DMA 传送字节数
0703 58         1473        POP     AX          ; RECOVER ADDRESS VALUE
                                                  (AX)为 DMA 传地送址的低 16
                                                  位
0704 03C1       1474        ADD     AX, CX      ; ADD, TEST FOR 64K OVER-
                                                  FLOW  基址+传送字节数应小
                                                  于64K
0706 59         1475        POP     CX          ; RECOVER REGISTER
0707 C3         1476        RET RETURN TO CALLER, CFL SET BY ABOVE IF
                                ERROR
                1477 DMA_SETUP ENDP
                1478
                1479 ;
                1480 ; WAIT_INT  WAIT_INT, 等待中断程序
                1481 ; THIS ROUTINE WAITS FOR THE FIXED EISK  等待硬盘
                     控制器通知中断已发生
                1482 ; CONTROLLER TO SIGNAL THAT AN INTERRUPT
                1483 ; HAS DCCURRED
                1484 ;
0708            1485 WAIT_INT PROC NEAR
0708 FB         1486 STI                    ; TURN ON INTERRUPTS  禁止中断
```

```
0709 53          1487 PUSH    BX          ; PRESERVE REGISIERS 保护寄存器
070A 51          1488 PUSH    CX
070B 06          1489 PUSH    ES
070C 56          1490 PUSH    SI
070D IE          1491 PUSH    DS
                 1492 ASSUME  DS:DUMMY
070E 2BC0        1493 SUB     AX, AX
0710 8ED8        1494 MOV     DS, AX      ; ESTABLISH SEGMENT
0712 C4360401    1495 LES     SI, HF_TBL_VEC  ES 指向硬盘特征参数表
                 1496 ASSUME  DS:DATA
0716 1F          1497 POP     DS
                 1498
                 1499 ; SET TIMEDUT VALUES
                 1500
0717 2AFF        1501 SUB     BH, BH
0719 268A5C09    1502 MOV     BL, DYTE PTR ES:(SI)(9)   STANDARD TIME
                                                        OUT 予送标准超时
                                                        值
071D 8A264200    1503 MOV     AH, COM_BLOCK+
0721 80FC04      1504 CMP     AH, FMTORV_CMD  是格式化驱动器命令?
0724 7506        1505 JNZ     W5
0726 268A5C0A    1506 MOV     BL, BYTE PTR ES:(SI)(0AH) FORMAT DRIVE
                                                        是，设格式化驱动器
                                                        用超时值
072A EB09        1507 JMP     SHORT W4
072C 80FCE3      1508 W5: CMP AH, CHK_DRV_CMD  是校验驱动器用超时值?
072F 7504        1509     JNZ W4
0731 268A5C08    1510     MOV BL, BYTE PTR ES:(SI)(08H) CHECK DRIVE 是
                                                        设校验驱动器用超时值
0735             1511 W4:
0735 2BC9        1512     SUB    CX, CX
                 1513
                 1514 ; ……WAIT FOR INTERRUPT
                 1515
0737             1516 W1:
0737 E84400      1517     CALL   PORT_1
073A EC          1518     IN     AL, DX          读驱动器 I/O 口
073B 2420        1519     AND    AL, 020H
073D 3C20        1520     CMP    AL, 020H        ; DID INTERRUPT OCCUR
```

```
                                                    中断发生否?
073F 740A          1521 JZ    W2
0741 E2F4          1522 LOOP  W1                    ; INNER LOOP  循环
0743 B2            1523 DEC   BX
0744 75F1          1524 JNZ   W1                    ; OUTER LOOP  超时值减至0超
                                                      时错
0746 C606740080    1525 MOV   DISK_STATUS, TIME-OUT
074B               1526 W2:
074B E82300        1527 CALL  PORT_0
074E EC            1528 IN    AL, DX
074F 24D2          1529 AND   AL, 2                 ; ERROR BIT  读I/O口0, 判错
                                                      误发生否
0751 08067400      1530 OR    DISK_STATUS, AL       ; SAVE
0755 E83000        1531 CALL  PORT_3                ; INTERRUPT MASK REGISTER
0758 32C0          1532 XOR   AL, AL                ; ZERO
075A EE            1533 OUT   DX, AL                ; RESET MASK  将中断层屏蔽
                                                      寄存器清.0
075B 5E            1534 POP   SI                    ; RESTORE REGISTERS  恢复寄
                                                      存器
075C 07            1535 POP   ES
075D 59            1536 POP   CX
075E 58            1537 POP   BX
075F C3            1538 RET   返回
                   1539 WAIT_INT ENDP
                   1540
0760               1541 HD_INT PROC NEAR  HD_INIT 程序段
0760 50            1542 PUSH  AX
0761 B020          1543 MOV   AL, EDI               ; END OF INTERRUPT  送中断结
                                                      束命令
0763 E620          1544 OUT   INT_CTL_PORT, AL
0765 B007          1545 MOV   AL, 07H               ; SET DMA MOOE TO DISABLE
                                                      禁止第3通道DMA
0767 E60A          1546 OUT   DMA+10, AL
0769 E421          1547 IN    AL, 021H
076B 0C20          1548 OR    AL, 020H
076D E621          1549 OUT   021H, AL  送8259A模式字节,禁止第5级中断
076F 58            1550 POP   AX
0770 CF            1551 IRET
                   1552 HD_INT ENDP
```

```
                1553
                1554 ; ……
                1555 ; PORTS
                1556 ; GENERATE PROPER PORT VALUE: 根据I/O偏移建立I/O
                       口地址(由DX返回)
                1557 ; BASED ON THE PORT OFFSET
                1558 ; ……
                1559
0771            1560 PORT_0 PROC NEAR  PORT_0,DX指向I/O口
0771 BA2003     1561 MOV    DX, HF_PORT   ; BASE VALUE: 取I/O基址320H
0774 50         1562 PUSH   AX
0775 2AE4       1563 SUB    AH, AH
0777 A07700     1564 MOV    AL, PORT_OFF  ; ADD IN THE OFFSET 取偏移量
077A 03D0       1565 AD0    DX, AX  修正DX
077C 58         1566 POP    AX
077D C3         1567 RET  返回
                1568 PORT_0 ENDP
                1569
077E            1570 PORT_1 PROC NEAR  PORT_1 DX指向I/O口
077E E8FOFF     1571 CALL   PORT_0
0781 42         1572 INC    DX              ; INCREHENT TO PORT DNE
0782 C3         1573 RET
                1574 PORT_1 ENDP
                1575
0783            1576 PORT_2 PROC NEAR  PORT_2:DX指向I/O口2
0783 EOF8FF     1577 CALL   PORT_1
0786 42         1578 INC    DX              ; INCREMENT TO PORT TWO
0787 C3         1579 RET
                1580 PORT_2 ENDP
                1581
0788            1582 PORT_3 PROC NEAR  PORT_3:DX指向I/O口3
0788 E8F8F      1583 CALL PORT_2
0788 42         1584 INC    DX              ; INCREMENT TO PORT THREE
078C C3         1585 RET
                1586 PORT_3 ENDP
                1587
                1588 ; ……
                1589 ;                                  SW2 OFFS  SW2_OFFS过程
                1590 ; OETERMINE PARAMETER TABIE OFFSET  作用: 利用控制
```

```
                        器口子地址及驱动器号标识推算出硬盘机参数表的偏移量，由
                        AX 返回
                1591 ; USING CONTROLLER RPOT TWO AND
                1592 ; DRIVE NUMBER SPECIFIER (0-1)
                1593 ; ..........................................................
                1594
0780            1595 SW2_OFFS PROC NEAR
0780 E8F3FF     1596 CALL   PORT_2
0790 EC         1597 IN     AL, DX      ; READ PORT 2  读口子 2，当为偶数号
                                            驱动器时，驱动器类型号在 2-3 位，奇数
                                            号时在 0-1 位
0791 50         1598 PUSH   AX
0792 E8E9FF     1599 CALL   PORT_1  读口子
0795 EC         1600 IN     AL, DX
0796 2402       1601 AND    AL, 2       ; CHECK FOR ERROR  状态字节第 1 位
                                            =1 表示有错
0798 58         1602 POP    AX
0799 7516       1603 JNZ    SW2_OFFS_ERR
0798 8A264300   1604 MOV    AH, CMD_BLOCK + 1

079F 80E420     1605        AND     AN, 00100000B  ; DRIVE 0 OR 1  取奇/偶号
                                                       驱动器标识位
07A2 7504       1606        JNZ     SW2_ND
07A4 D0E8       1607        SHR     AL, 1          ; ADJUST  偶数号驱动器，类
                                                       型号移至低 2 位
07A6 D0E8       1608        SHR     AL, 1
07AB            1609 SW2_ANO:
07AB 2403       1610        AND     AL, 011B       ; ISOLATE  取低 2 位的类型
                                                       号
07AA B104       1611        MOV     CL, 4
07AC D2E0       1612        SHL     AL, CL         ; ADJUST  类型号乘 4，变成该
                                                       类驱动器之参数表的偏移（与
                                                       FD_TBL 的）
07AE 2AE4       1613        SUB     AH, AH
07B0 C3         1614        RET
07B1            1615 SW2_OFFS_ERR:
07B1 F9         1616        STC
07B2 C3         1617        RET
                1618 SW2_OFFS  ENDP
```

```
                        1619
07B3 30382F31342F3832   1620        DB      08/16/82; RELEASE MARKER
                        1621
07B8                    1622 END_ADDRESS LABEL BYTE
                        1623 CODE  ENDS
                        1624        END
```

第二十八章　IBM同步数据链路控制(SDLC)通讯转接器

§28.1　概　　述

SDLC通讯转接器是一个由通讯软件控制，工作于半双工模式的接口器，最大数传率为9600波特率。转接器的系统控制信号，电压信号，数据信号均从2×31定位插卡的边脚引出。Modem接口方式为EIA驱动器、接收器式，接口插件为RS232C标准25脚D型插头。SDLC内部芯片主要有Intel的8273 SDLC协议控制器、8255 A-5可程序外设接口，8253-5时钟控制器，并且8273协议芯片及8255 A-5外设接口芯片构成了SDLC转接器外部Modem接口。下面是SDLC通讯转接器的框图。

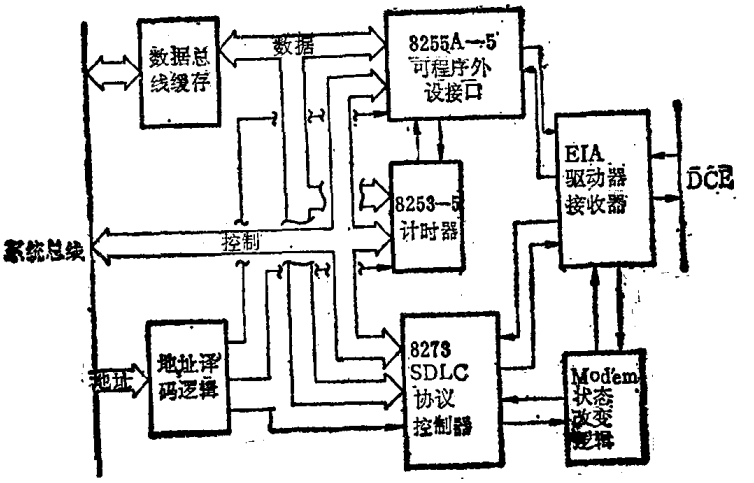


图8-1　SDLC通讯转接器框图

8273 SDLC协议控制模块具有以下五个特点。

- 自动生成，检查帧检验序列
- 自动插入、删除零位
- 与TTL电平兼容
- 内部双重处理器结构，接收帧级命令，能控制与最小处理器模式相接的数据通道
- 全部的操作(发送、接收、送口子内容)均分成命令、执行、结果返回三个阶段

§28.2　8273 协议控制器

8273(框图见图28-2)芯片具有两个通讯界面：一为处理器界面，另一为Modem界面。处理器界面在8273内部数据总线的左侧，由控制/读/写逻辑(C/R/W)、内部寄存器组、数据传送逻辑、数据总线缓存器四个块组成。Modem界面在内总线右侧，由Modem控制块、串型数据定时块组成。

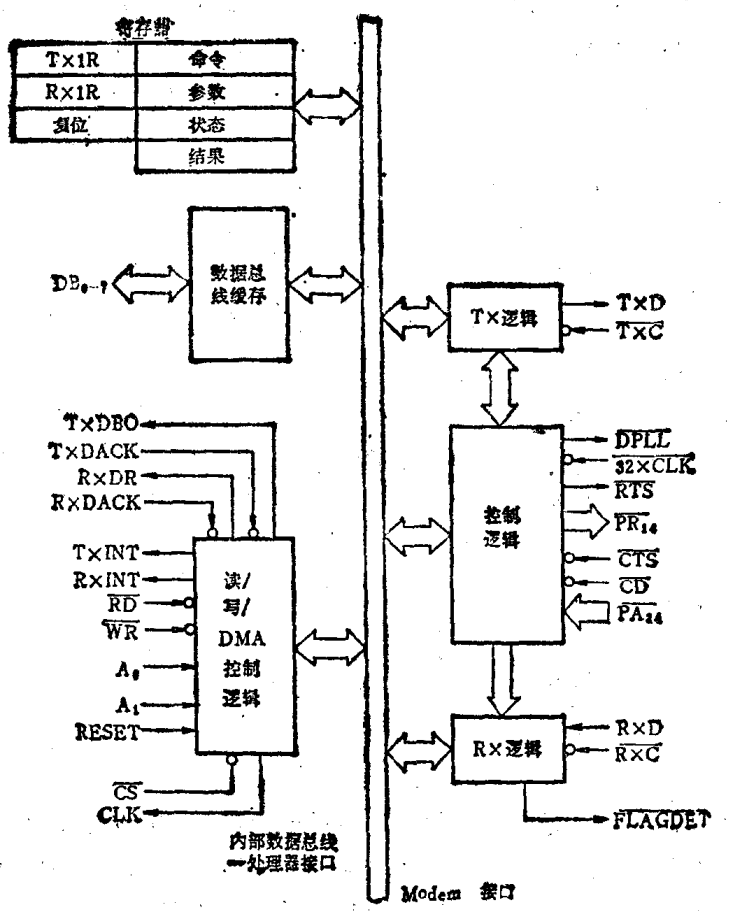


图 28-2 8273 SDLC协议控制器框图

表 28-1 SDLC协议控制器内寄存器选择

地址输入		控制输入			寄存器
A0	A1	CS	WR	RD	
0	0	0	0	1	命令
0	0	0	1	0	状态
1	0	0	0	1	参数
1	0	0	1	0	结果
0	1	0	0	1	复位
0	1	0	1	0	TxI/R
1	1	0	0	1	无
1	1	0	1	0	RxI/R

一、控制/读/写逻辑

本逻辑用来接收主处理器发出的命令，发回命令执行后的结果。逻辑块还进行地址译码，将处理器选片信号 CS、读信号 RD、写信号 WR、A1、A0 地址信号组合成七个内部寄存器的寻址选择信号(见表 28-1)，接收、发送网络中断信号。

二、内部寄存器(8273 控制/读/写寄存器)

共有七个内部寄存器，它们是：

1) 命令寄存器：记录操作要求，由主处理器写。

2) 状态寄存器：记录完成 8273 各个动作步必要的主处理器/转接器握手状态电平，这是个通用状态寄存器。

3) 参数寄存器：存放命令用参数。

4) 直接结果寄存器：记录直接执行型命令执行后的结果字节，由主处理器读取。

5) 发送器中断结果寄存器 (T_xI/R)：发送操作的结果由该寄存器传给主处理器。有效结果将产生一个给主处理器的中断信号。

6) 接受器中断结果寄存器(R_xI/R)：接收操作的结果由该寄存器传给主处理器。有效结果将产生一个给主处理器的中断信号。

7) 复位寄存器：向软件提供一种复位 8273 的功能。

三、数据接口

8273 通过数据发送逻辑支持两种相互独立的数据接口：接收数据接口，发送数据接口。它们对 DMA/非 DMA 数据传送均是可程序的。在 DMA 方式下，主处理器 DMA 控制器管理数据传送的定时、寻址，8273 则负责发出传送请求，数据块记数。8273 以第一级 DMA 传送数据。两种传送方式一般根据传递对象及中断响应时间选定，对 9600 波特率下的数传可选用非 DMA 方式。

四、数据传送接口信号

T_xDRQ/R_xDRQ：请求与主存贮器建立 DMA 通道，由 8273 设置。

T_xDACK/R_xDACK：DMA 请求应答信号，由主处理器发出，表示可进行 DMA。T_xDACK/R_xDACK 接于系统控制总线的 DACK1 上。

RD(读)：表示数据是否可从 8273 送入主存，由主处理器 DMA 控制器发出。

WR(写)：表示主存数据是否能送入 8273，它由主处理器 DMA 控制器发出。

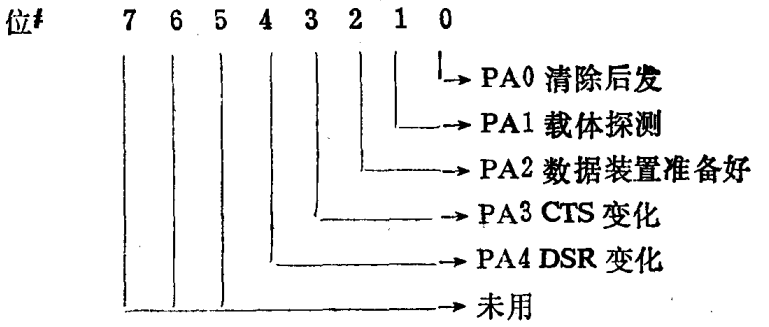
8273 与主存 DMA 传送的过程大致如下：8273 升起 DMA 请求信号，主处理器 DMA 控制器取得系统总线控制权，发回 DACK 信号和决定传送方向的 RD 或 WR 信号。该两信号到达 8273 后立即进行 DMA 传送。这种不依赖 8273 选片信号的 DMA 能省去大量读写数据寄存器的时间。

五、Modem 控制块

Modem 控制块属 Modem 界面，它提供专用或用户自定义两种 Modem 控制功能、通过 EIA 反相驱动器将 TTL 电平转换成 EIA 电平。8273 的 A 口、B 口均为 Modem 控制用 I/O

口，A 口负责输入，B 回负责输出。

8273 A 口 (Modem 控制用输入口)



说明：

位 0：反映清除后发送引脚 CTS 的逻辑状态。8273 送出帧前 CTS 必须有效。CTS 失效时正发送的那一帧作废，主处理器被中断。CTS 失效在中断状态寄存器有自己的表达电平。

位 1：反映载体探测脚 CD 的逻辑状态。接收帧地址时，CD 应在一段时间内保持有效。CD 在接收帧时失效，将导致主处理器中断。

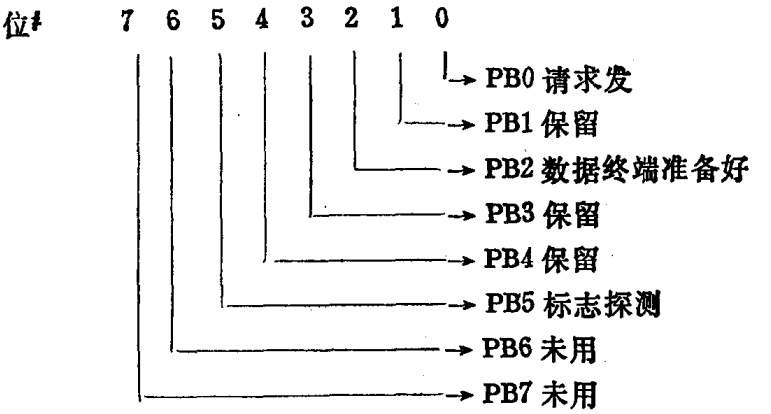
位 2：数据装置准备好(DSR)测试位。

位 3：反映 CTS 电平变化的测试位。

位 4：反映 DSR 电平变化的测试位。

位 5-7：无用，读 PA 口时他们为 1。

8273 B 口 (Modem 控制用输出口)



说明：

位 0：反映请求发送脚(RTS)逻辑状态。本位由 8273 自动处置。

位 1：保留。

位 2：数据终端准备好。

位 3：保留。

位 4：保留。

位 5：反映标记探测引脚的状态。该引脚在接收器发现标记字符后逞有效电平。

位 6-7：未用。

六、串型数据定时块

串型数据定时块也属 Modem 界面，它分为两个部分：即由发送数据输出脚 T_xD、接收输入脚 R_xO、各自时钟组成的串型数据逻辑和用于诊断时取得反环数据的数字锁相环(DDLL)有关数据传输的定时见图 28-3，其中 T_xC 的上升沿生成新的发送数据，R_xC 的下降沿用于捕获接收的数据。

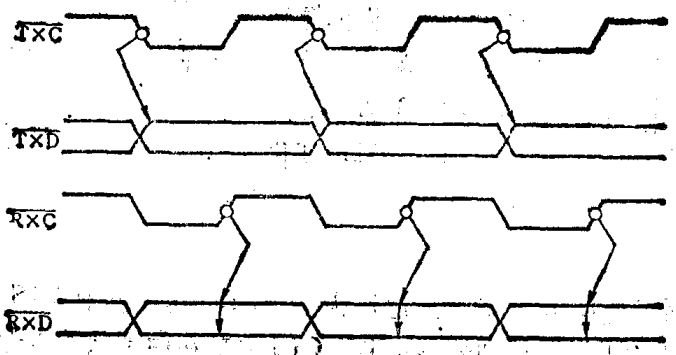


图 28-3　8273 协议控制器发送/接收定时图

§28.3　8255A-5 可程序外设接口及 8253-5 可程序间隔定时器

一、8255A-5 接口

8255A-5 PA 口*　　I/O 地址 0380H

位#	7	6	5	4	3	2	1	0

0 = 响铃指示器打开
0 = 接口数据载体探测开
振荡 = 发送时钟活跃
0 = 接口清除后发开
振荡 = 接收时钟活跃
1 = Modem 状态改变了
1 = 计时器 2 输出活跃
1 = 计时器 1 输出活跃

* PA 口定义为输入口

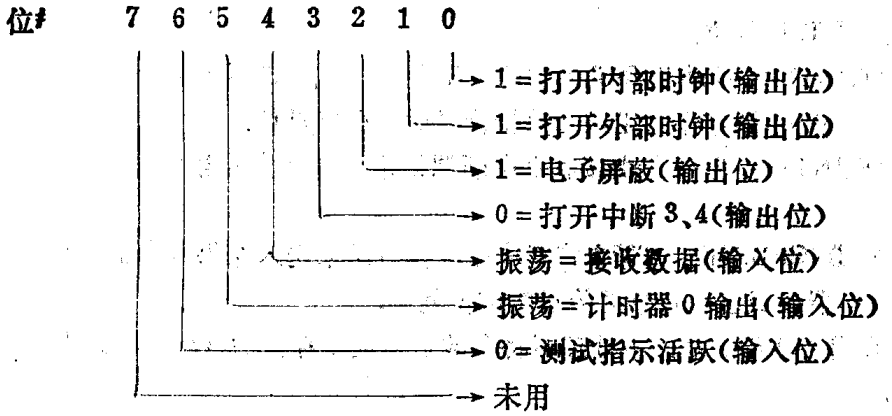
8255A-5 PB 口*　　I/O 地址 0381H

位#	7	6	5	4	3	2	1	0

0 = 打开 Modem 接口的数据信号率选择开关
0 = 打开 Modem 接口的选择准备收发开关
0 = 打开测试
1 = 复位 Modem 状态改变逻辑
1 = 复位 8273
1 = 打开计时器 2
1 = 打开计时器 1
1 = 允使第 4 级中断

* PB 口定为输出口

8255A-5 PC 口*　　I/O地址 0382H



* PC 口定义内部控制和打开功能，他有三个输出位。四个输出位。四个输出位初始时就定义好了，以后只用三个。

二、8253-5 可程序间隔定时器

8253-5 由主处理器振荡的二分频信号驱动，它的三个计时器的作用是：

计时器 0：生成方波供计时器 2 作输入信号及 8755A-5 振荡信号(C 口第 5 位)。

计时器 1：连向 8255A 口第 7 位及中断 4 引脚。

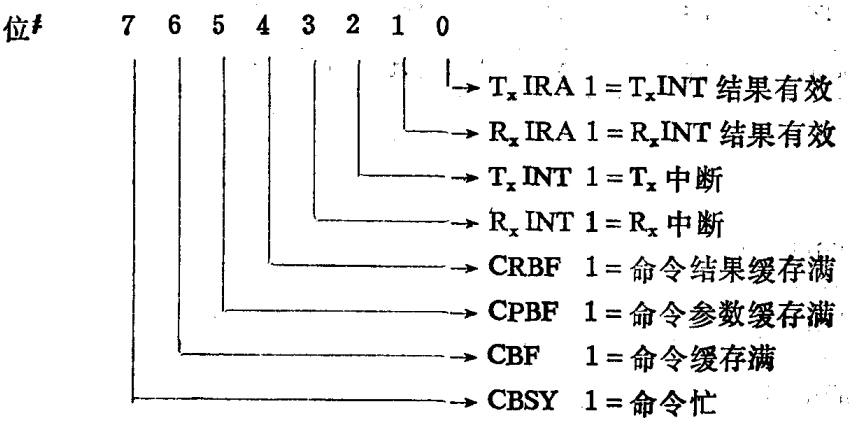
计时器 2：连向 8255A 口第 6 位及中断 4 引脚。

§28.4 编 程 要 点

一、状态寄存器

编程 8273 时主要考虑的问题是主处理器与 8273 内部寄存器的通讯。根据 8273 的内部结构安排，这种通讯是握手式的。前面介绍过的状态寄存器，被指派为握手式通讯的公共界面。下面是状态寄存器的格式以及各个位的含义。

8273 状态寄存器格式　　I/O 地址 0388H



说明：

位 0：发送器中断结果有效 (T_xIRA) 标志位。8273 在 T_xI/R 寄存器放入中断结果字节

时该位置位。主处理器读 T_xI/R 寄存器后复位。

位 1：接收器中断结果有效(R_xIRA)标志位。8273 在 R_xI/R 寄存器内放入中断结果时该位置位。主处理器读 R_xI/R 后复位。

位 2：发送器中断(T_xINT)位，反映 T_xINT 引脚的电平变化。当发送器需主处理器服务时就将 T_xINT 脚置位。主处理器读结果或正进行数据传送时该引脚复位。

位 3：接收器中断(R_xINT)位，除动作由接收器中断源初始外，该位的含义均与 T_xINT 位相同。

位 4：命令结果缓存满(CRBF)位。8273 将直接命令执行结果送入结果寄存器时该位置位，主处理器读结果寄存器后或正进行数据传送时该位复位。

位 5：命令参数缓存满 (CPRF) 位。主处理器将参数送参数寄存器后该位置位。8273 接收了参数后该位复位。

位 6：命令缓存满(CBF)位。置位时表示命令寄存器内有一字节。一般不用。

位 7：命令忙(CBSY)位。置位时表示 8273 可接收命令。复位表示命令需最后一个参数已被送至参数寄存器，并被 8273 接收。

二、初始转接器的一般步骤

初始 8273 前，应将转接器其余芯片置好操作模式。对于 8255 A-5 接口，应选好模式设定地址(参见 SDLC 通讯转接器地址表)，设好 A、B、C 三口子的 I/O 模式。其次向 C 口输出一字模，以禁止中断，选择屏蔽模式，打开外部时钟引脚(I/O 地址 0382H、0DH)。此时转接器与通讯接口相开脱。最后用 PB 口第 4 位让 8273 复位线由低电平变高电平，保持一段时间后再降下，使 8273 所有的内部寄存器被初始化。

对于 8253-5 定时器，计时器 0 设以模式 3，产生方波；计时器 1、2 设以在产生中断 4 前，就结束计数的计数初值，这样中断 4 就能向协议控制器软件报告中断 3 发生前已过去了一个予定的时间段。

表 28-2　8253-5 可程序间隔计时器控制字

控制字格式

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

各控制位的意义：

SC—选择计数器

SC1	SC0	
0	0	选择计数器 0
0	1	选择计数器 1
1	0	选择计数器 2
1	1	非法

控制字格式

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

各控制位的意义：

RL—读/装入

RL1	RL0	
0	0	计数器锁存操作
1	0	读/装入高字节(MSB)
0	1	读/装入低字节(LSB)
1	1	先读/装入低字节，后高字节

M—模式

M2	M1	M0	模式
0	0	0	模式 0
0	0	1	模式 1
×	1	0	模式 2
×	1	1	模式 3
1	0	0	模式 4
1	0	1	模式 5

BCD

0	二进制计数器 16 位
1	BCD 码计数器 4 位数

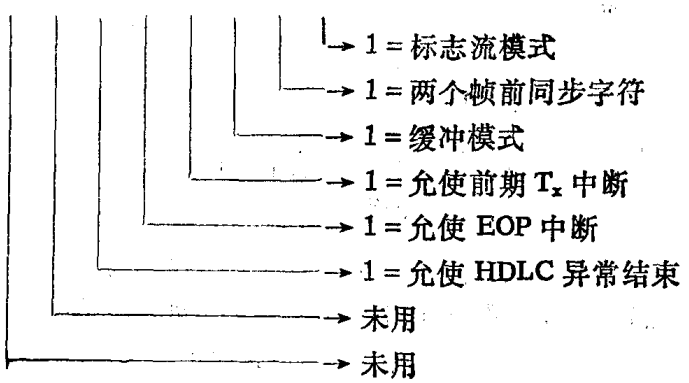
三、8273 初始/构型命令

复位后，8273 所有寄存器均被缺省成全 1，8273 初始/构型命令，就是将这些全 1 的寄存器换以工作模式。初始/构型命令，要求有一个位模式参数字节，用置位型命令或复位型命令把位参数“或入”或者“与入”模式寄存器。下面介绍操作、串型 I/O、一位延迟、数据传送四个模式寄存器的格式。

1）操作模式寄存器

格式：

位* 7 6 5 4 3 2 1 0



说明：

位 0：本位置位且发送器空闲时，标志为 1 的标志位将立即发送出去，若发送或与发送无关的命令有效，标志位将在发送完毕后马上发送出去。反环发送或做一位延迟时本位无用。本位复位则发送器无数据发送，或 0 位复位时正恰发送完毕情况下，以下一字符为界送空闲电平。

位 1：本位置位时，8273 在每帧首标识符前予发二个字符，他们可以是 00(如果 NRZI 置位)或 55H(如果 NRZI 复位)。请参见串型 I/O 模式寄存器 NRZI 译码格式。

位 2：置位时 8273 缓存接收帧的首二个字符(不传给主存)。复位则首二个字符传给主存，然后返送。

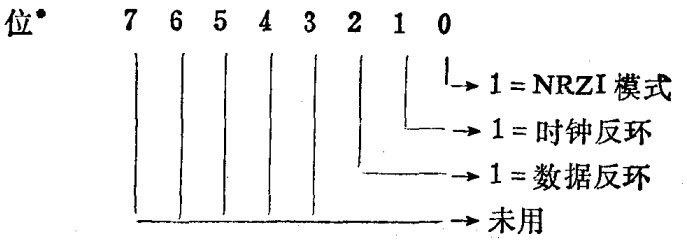
位 3：本位决定 8273 产生帧结束中断的时机。置位时，最后一个数据字符传至 8273 后即发早期中断。若主处理器响应早期中断，并在最后标识符发送前下达另一个传送命令，则最后标志中断不会形成。8273 在当前帧结束后开始一个新的帧。这样帧帧间就只有一个分隔标识符。本位复位则中断仅在最末标识后形成。

位 4：供 EOP 中断模式用，SDLC 转接器不用该位。位 4 应置成复位态。

位 5：SDLC 工作时总复位，使 8273 能识别出 8 个 1 组成的异常结束符 (011111111)

2) 串型 I/O 模式寄存器

格式：



说明：

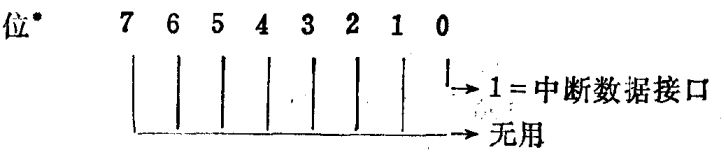
位 0：本位置位指定 NRZI 编码、解码。复位本位表示发送或接收数据流是一个正逻辑位流。

位 1：与反环位(位 2)连用。置位时发送时钟与接收时钟相连。复位时发送时钟、接收时钟依赖各自的 I/O 引脚。

位 2：置位时发送的数据被引回接收数据线路。复位时发送数据、接收数据来自各自的 I/O 引脚。

3) 数据传送模式寄存器

格式：



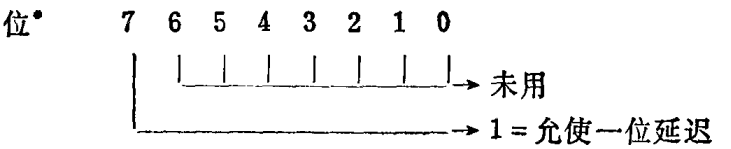
说明：

数据传送模式寄存器置位时，8273 将以中断方式通知主处理器请求发送数据字节或已接收了数据字节，此时如果状态寄存器未指出有发送或接收中断结果，中断被解释成发送或

接收数据请求中断。本寄存器复位后发送、接收以 DMA 方式进行。

4) 一位延迟模式寄存器

格式：



说明：

本寄存器第 7 位置位后，8273 将接收到的数据流延迟一位，然后发给对方(称一位延迟模式)。第 7 位复位，则 8273 停止一位延迟发送。

下表汇集了 8273 全部置位型、复位型设置模式寄存器的命令。表中复位、置位屏蔽字节是命令参数。8273 执行完这些命令后不发回中断或结果信息。

表 28-3 8273 协议控制器模式寄存器命令

寄存器	命令类型	十六进制命令码	参数
一位延迟模式	置位	A4	置位屏蔽
	复位	64	复位屏蔽
数据传送模式	置位	97	置位屏蔽
	复位	57	复位屏蔽
操作模式	置位	91	置位屏蔽
	复位	51	复位屏蔽
串型 I/O 模式	置位	A0	置位屏蔽
	复位	60	复位屏蔽

四、命令执行的三个阶段

尽管 8273 是一个完全双工的器件，但由于只有一个命令寄存器的缘故，任一时刻命令寄存器上能为一个命令序列服务，发送器或接收器不会同时据于命令阶段。命令阶段始于系统程序将命令、参数写进命令、参数寄存器。图 28-4 是命令阶段的流程图。从图中可看到握手式通讯通过状态寄存器 CBSY、CPBF 位完成。CBSY = 1 时写入新命令则原来的命令失效。参数寄存器满的测试也是必须的，不然 8273 可能会丢失刚收到的参数(即 CPBF = 1 时写新参数)。

表 28-4 8273 SDLC 协议控制器命令一览表

命令	命令码	参数	结果	结果口	完成中断
置一位延迟	A4H	设置屏蔽	无	—	无
复位一位延迟	64H	复位屏蔽	无	—	无
设数据传送模式	97H	设屏蔽	无	—	无
复位数据传送模式	57H	复位屏蔽	无	—	无
置位操作模式	91H	置位屏蔽	无	—	无
复位操作模式	51H	复位屏蔽	无	—	无
置位串型 I/O 模式	A0H	置位屏蔽	无	—	无

（续表）

命　　令	命令码	参　　数	结　　果	结果口	完成中断
复位串型I/O模式	60H	复位屏蔽	无	—	无
普通接收	C0H	80、81	R1C、R0、R1、A.C	RXI/R	有
选择接收	C1H	80、81、A1、A2	R1C、R0 R1 A.C	RXI/R	有
接收禁止	C5H	无	无	—	无
发送帧	C8H	L0.L1.A.C	TIC	TXI/R	有
发送透明	C9H	L0.L1	TIC	TXI/R	有
异常终止帧发送	CCH	无	TIC	TXI/R	有
异常终止发送透明	CDH	无	TIC	TXI/R	有
读口A	22H	无	A口内容	结　果	无
读口B	23H	无	B口内容	结　果	无
设口B	A3H	置位屏蔽	无	—	无
复位口B	63H	复位屏蔽	无	—	无

8273 命令辅助记号含义

B0—接收器缓存长度的低字节

B1—接收器缓存长度的高字节

L0—区帧长度的低字节

L1—区帧长度的高字节

A1—第一接收帧地址场匹配参数

A2—第二接收帧地址场匹配参数

C—接收帧的控制场。若指定无缓冲模式，C不提供

RXI/R—接收中断结果寄存器

TXI/R—发送中断结果寄存器

R0—收到帧的长度低字节

R1—收到帧的长度的高字节

RIC—接收器中断结果码

TIC—发送器中断结果码

2）执行阶段

执行阶段执行命令规定的动作，这里有两种情形：一为无需处理器介入的DMA数据传送，另一为中断驱动式的数据传送。对于后者，8273将升起T_xINT或R_xINT脚的电平，发出中断。主处理器响应中断，检查状态寄存器及相应的IRA位。如果IRA位等于0，说明中断源为数据传送请求；IRA等于1说明操作完毕，应读取中断状态寄存器。

3）结果返回阶段

命令正确执行完毕或出现错误都将使8273进入结果返回阶段。返回的结果有直接型、非直接型两种。读写I/O口子一般得到直接型结果。这个结果在8273结果寄存器读取，由状态寄存器的CRBF位指出当时的直接结果是否有效。非直接结果是发送器或接收器产生的，他们由发送中断结果寄存器T_xI/R、接收中断结果寄存器R_xI/R以一个字节或多个字节组成的结果代断码形式返回。这两个寄存器内容是否有效，由状态寄存器T_xIRA、R_xIRA位指出。详尽的结果码见表28-5。

五、发送、接收的典型工作过程

以下三种工作序列都是用DMA在主处理器、8273间传送数据的。

1）发送

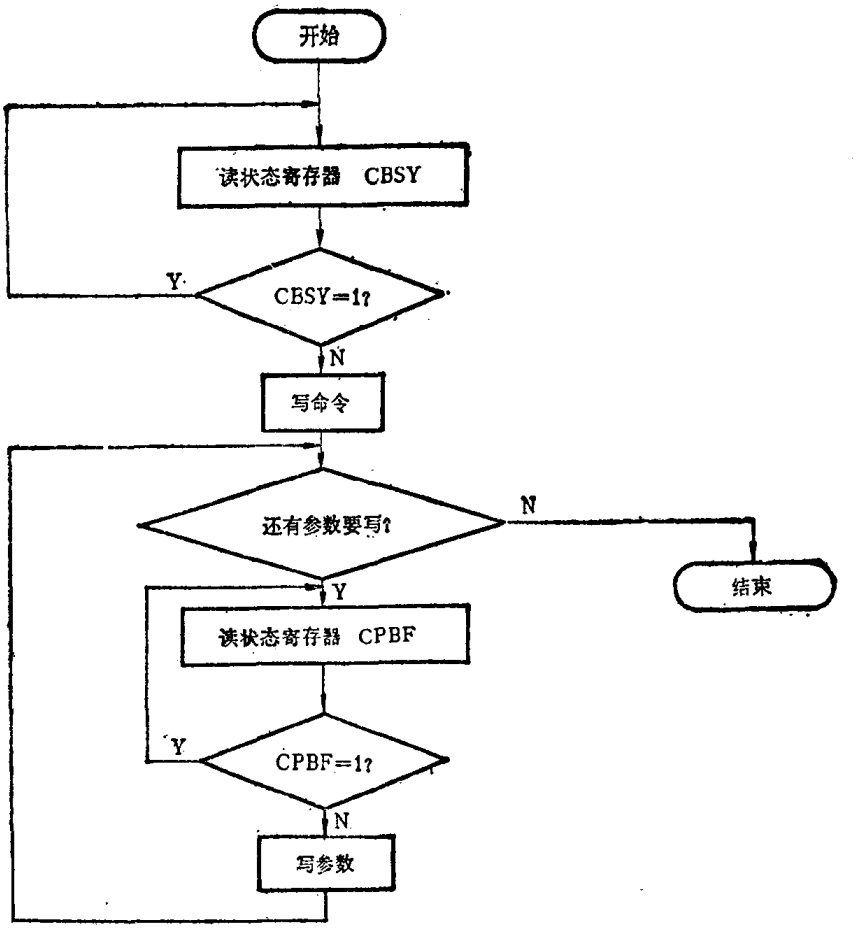


图 28-4 8273 协议控制器命令阶段流程

表 28-5 结果码汇总

	码 值	结 果	中断后的状态
发 送	0CH	早期发送器中断	发送器活跃
	0DH	帧发送完毕	空闲或标志
	0EH	DMA 漏过	异常终止
	0FH	清除后发送错误	异常终止
	10H	异常终止完成	空闲或标志
接 收	X0H	A1 匹配或普通接收	活跃
	X1H	A2匹配	活跃
	03H	CRC 错	活跃
	04H	探测出异常终止	活跃
	05H	探测出空闲	被禁止
	06H	探测出 EOP	被禁止
	07H	帧长少于 32 位	活跃
	08H	DMA 超越	被禁止
	09H	存贮器缓存溢出	被禁止
	0AH	载体探测失败	被禁止
	0BH	接收器中断超越	被禁止

注：X=收到的部分字节的位数

发送一个帧前，通讯软件预先将传送信息场的起始地址送给 DMA 控制器，再把帧发送命令及要求的 4 个参数(帧地址场字节、帧控制场字节、表示待发信息长度的二个字节)送给 8273。8273 收到命令后，将请求发送引脚 RTS 升为有效电平，等待 Modem 来的清除发信号(CTS 变有效)。CTS 一旦有效，8273 就开始帧发送，并且在发送打开标志、地址场、控制场后，与主处理器进行 DMA 通讯，读入信息，发送信息，直至长度字节规定的信息全部发完。最后 8273 停止 DMA 请求，发送一个帧检测序列 FCS 和关闭标志，升起 T_xINT 引脚电平，通知主处理器帧发送完毕，请读取结果字节。由于采用 DMA，除装入命令、参数外，主处理器并不介入真正的发送过程。

2) 普通接收

接收过程与发送过程在命令设置、DMA控制器压载等方面是极为相似的，如选定 DMA 接收缓存的地址，发给 8273 一个接收命令等。但有一点必须指出：接收命令有普通接收、选择接收两种形式。前者将收到的帧全部发给主存，后者只把收到帧中与 8273 编程地址场相匹配者送至主存。下面是普通接收的一个例子。8273 装入接收命令及表示接收器长度的两个字节，开始接收数据，主处理器则转去处理其他任务，下一帧到达接收器输入端后，马上被送至 DMA 规定的内存。收到关闭标志后，8273 检查 FCS，升起 R_xINT 线，主处理器响应中断，检查结果。对于接收帧大于接收命令规定长度的情况，主处理器会在帧结束中断前，收到一个接收器有错中断。

3) 选择接收

选择接收与普通接收相比需额外增加两个地址匹配参数 (字节 A1、A2)。处于选择接收模式的 8273，只把地址场与 A1 或 A2 相同的帧送至主存。该命令一般用于 A1 指定二级站地址，A2 指明合成线地址的二级通讯站。若只要求有一个匹配字节，则让 A1 等于 A2。象普通接收一样，8273 要检查接收帧是否超长。

六、对结果码汇总表的说明

接收栏第一、第二个结果码表示接收帧无误。因 SELC 对帧长度不作限制 (>32 位就行)，接收结果的高位部分是最后收到数据字节的有效位数。下表是该码的译码含义。

X	未字节中的接收位数
E	未字节全部位(8 位)
0	第 0 位
8	第 1 位—第 0 位
4	第 2 位—第 0 位
C	第 3 位—第 0 位
2	第 4 位—第 0 位
A	第 5 位—第 0 位
6	第 6 第—位 0 位

七、接口

SDLC 通讯转接器采用 EIA RS-232C 标准接口电平(见图 28-5)。转接器插座上有三个引脚，实际并不落入 EIA 电平范围。他们是第 11 号选择准备收、发脚，第 18 号测试引脚

和25号测试指示引脚。第8号脚用于支持开关网络中有后缓功能的Modem。第18、25号脚用于支持商用机器上具有屏蔽功能的Modem。

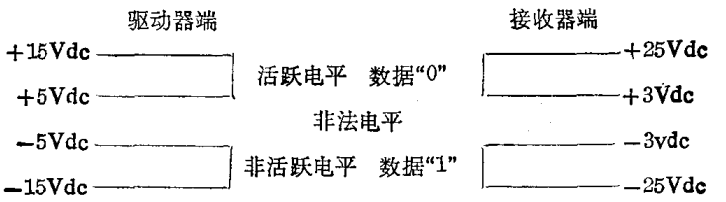


图28-5 EIA电平范围

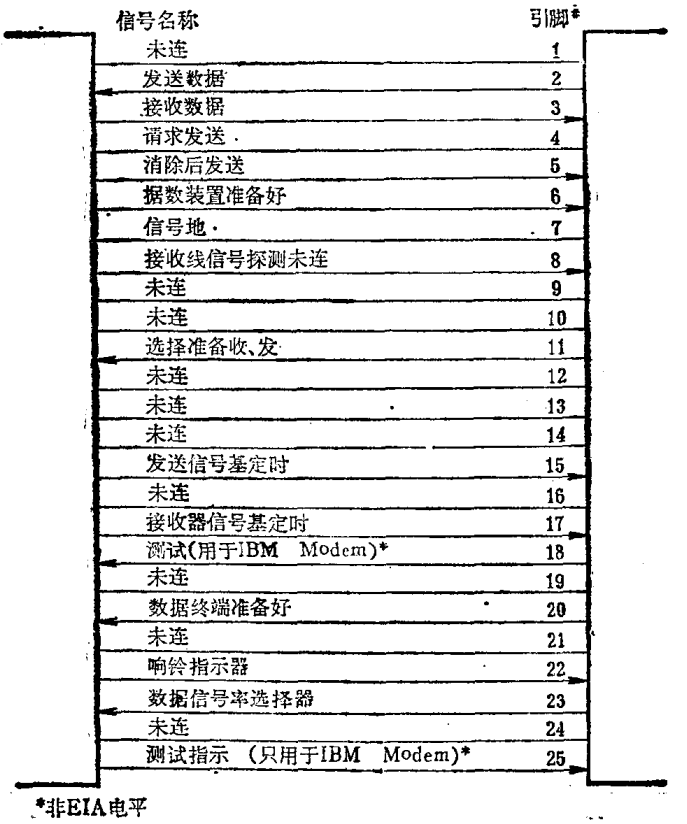


图28-6 SDLC通讯转接器接口说明

表28-6 SDLC通讯转接器设备地址

I/O 地址	设 备	寄 存 器	作 用
380H	8255	PA口数据	内部/外部检测
381H	8255	PB口数据	外部Modem接口
382H	8255	PC口数据	内部控制
383H	8255	模式设定	8255 模式初始
384H	8253	计时器0 LSB	方波发生器
384H	8253	计时器0 MSB	方波发生器
385H	8253	计时器1 LSB	暂停时间输出
385H	8253	计时器1 MSB	暂停时间输出
386H	8253	计时器2 LSB	暂停时间输出

（续表）

I/O	设　备	寄　存　器	作　用
386H	8253	计时器 2 MSB	暂停时间输出
387H	8253	模式寄存器	8253 模式设定
388H	8273	命令/状态	输出＝命令，输入＝状态
389H	8273	参数/结果	输出＝参数，输入＝状态
38AH	8273	发送 INT 状态	DMA/INT
38BH	8273	接收 INT 状态	DMA/INT
38CH	8273	数据	DPC(直接程序控制)

表 28-7　中断信息

中　断　级　别	作　用
3	发送/接收中断
4	计时器 1 中断
4	计时器 2 中断
4	清除后发送信号改变
4	数据装置准备好信号改变
DMA 0	发送，接收

第二十九章　ROM 区

§29.1　ROM　BIOS 简介

基本 I/O 系统 BIOS 是 Basic I/O System 的缩写，它占据母板上 8KB ROM区，所以它也称为 ROM BIOS。BIOS 在汇编语言级上向用户或系统程序(如 PC DOS)提供 PC 一些主要外设的设备层控制功能，包括盒带操作系统、开机自检、磁带机字节传递、显示器、通讯口、键盘和打印机的字符传送、图形发生等，这些操作均无需用户考虑外设的 I/O 地址等细节。BIOS 还能提供时间及内存容量等参考数据。BIOS 是用户与硬件间的第一隔离层，外设的更改或增减均对用户透明。

一、BIOS 的用法

进入 BIOS 各子程序的方法是利用 8088 的 INT 软中断指令，CPU 响应中断后把控制权交给某个 BIOS 子程序段，由它提供中断服务。表 29-1 是所有中断(硬、软中断)的中断向量表，其中 10H—1AH 是 BIOS 子程序的软中断号，如 INT 12H 将调用 BIOS 内存容量测试程序，并能得到实际结果。

BIOS 调用的入口参数是这样传递的：所有参数均用 8088 寄存器传送，每个 BIOS 子程序前均有传递参数及返回值寄存器名的说明。如上面的内存容量测试子程序，其说明部分指出调用无需参数，而结果(增量为 1K)也就是存贮器容量由 AX 寄存器返回。

若 BIOS 子程序能完成好几种功能，则用 AX 寄存器指定要求的功能。例如设定时间的 BIOS 调用，要求写明的 8088 指令是：

```
MOV   AX, 1;                  指定时间设定
MOV   CX, HIGH-COUNT;         建立当前时间
MOV   DX, LOW-COUNT;
INT   1AH;                    设定时间
```

若要知道当前的时间则写明：

```
MOV   AH, 0;                  指定读时间
INT   1AH;                    读计时器
```

一般情况下，BIOS 各子程序保留除 AX，状态寄存器外所有寄存器的原先内容，只有返回结果值的寄存器的原来内容被修改了。

表 29-1　中断向量表

中　断　号	名　　称	BIOS 初 始 参 数
0	除数为 0	/
1	单步	/
2	不可屏蔽中断	NMI-INT(F000:E203)
3	断点	/

续表

4		溢出	/
5		打印屏幕	PRINT-SCREEN(F000:FF54)
6		未用	
7		未用	
8	8259 向量中断	时间	TIMER-INT(F 000:FEA5)
9		键盘	KB-INT(F 000:E 987)
A		未用	
B		未用	
C		未用(保留接通讯口)	
D		未用	
E		软盘机	DISK-INT(F 000:EF 57)
F		未用(保留接打印机)	
10	BIOS 入口点	显示器	VIDEO-IO(F 000:F 065)
11		设备检查	EQUIPMENT(F 000:F 84 D)
12		存贮器容量	MEMORY-SIZE-DETERMINE (F 000:F 841)
13		软盘机	DISKETTE-IO(F 000:EC 59)
14		通讯口	RS232-IO(F 000:E 739)
15		盒带机	CASSETTE-IO(F 000:F 859)
16		键盘	KEYBOARD-IO(F000:E 82E)
17		打印机	PRINTER-IO(F 000:EF 02)
18		盒带 BASIC	(F 600:0000)
19		引导程序	BOOT-STRAP(F 000:E 6F2
1A		时间	TIMER-OF-DAY(F 000:FE6E)
1B	用户自编 服务程序	键盘阻断	DUMMY-RETURN(F000:FF53)
1C		计时器	DUMMY-RETURN(F000:FF53)
1D	BIOS 用 参数区	CRT 初始参数区	VIDEO-PARMS(F 000:F0A 4)
1E		软盘机参数区	DISK-BASE(F 000:EFC7)
1F		图形 CRT 字符	无

二、对特别中断向量的说明

1．中断 1 BH—键盘阻断

该向量指向处理按下 CTRL BREAK 键的代码段段首，该代码段的返回指令是 IRET。BIOS 初始程序将这个向量指向 IRET 指令，所以若应用程序不改变向量值的话，按不按 BREAK 键都是一样的。CTRL BREAK 键处理程序能掌握 CPU 控制权，但会出现如下的问题：处理中断时按下 BREAK 键，相应就会有一个或多个中断结束命令发向 8259 A 控制器。另外，所有正在工作的 I/O 设备均被复位了。

2．中断 1CH——计时信号

1CH 向量指向处理系统时钟计时中断子程序，该子程序的返回指令是 IRET。BIOS 初始程序将这个向量指向 IRET 指令，所以如果应用程序不修改该向量的话，1CH 中断不起作用。另外，BIOS 不保护任何寄存器。

3．中断 1 DH——CRT 参数

该中断向量指向初始显示转接器内 6845 控制器需要的参数数据区。数据区中有 4 个

表格，如果要求 6845 支持所有工作模式的话，4 个表格都要予以重新改写。BIOS 初始程序将该向量指向 ROM 显示程序参数区。

4．中断 1EH——软盘机参数

该向量指向软盘驱动器所需的参数数据区。BIOS 初始程序将该向量指向 ROM 软盘机子程序参数单元。参数缺省的话表示系统连接的是 IBM 驱动器。改动参数区内容就能连上其它类型的软盘驱动器。

5．中断 1FH——图形字符扩充

当彩色/图形转接器工作于图形模式时(帧尺寸 320×200 或 640×200)，读写字符接口利用图形点阵集得到显示字符的代码点阵。图形点阵集的第一批点阵 (128 个) 存在 ROM 区，起始地址为 FA6E，另外 128 个点阵从中断向量 1FH 指向的 1K 字节表得到。在这张表中，每个代码点阵由 8 个字节的图形信息表示。开机时 1FH 向量被初始成 00。如果要求更多的字符点阵的话，用户必须修改这个中断向量。

三、BIOS 占据的 RAM 区

IBM ROM BIOS 程序使用绝对地址 400～4FF 共 256 个 RAM 字节单元，其中 400～407 单元记录连接在系统上 RS 232 转接器插座的基本地址，单元内容全 0 表示没有连异步通讯口。408～40F 单元是打印机基本地址，其作用同 400～407 单元。300～3FF 单元是开机后初始代码段用的堆栈区。

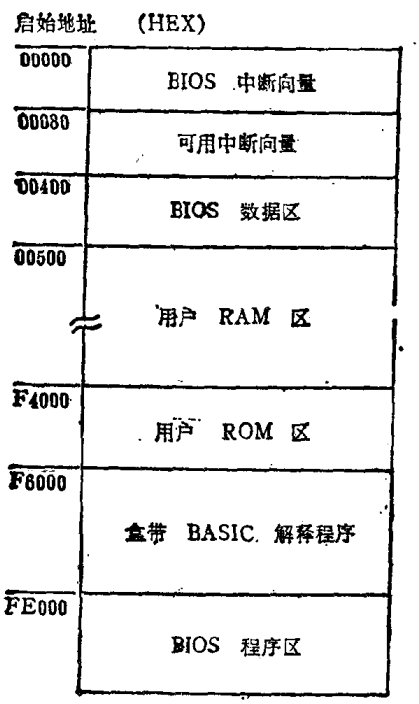


图 29-1 BIOS 存贮区映照

§29·2 BIOS 盒带机逻辑

一、中断 15 H

盒带机子程序的调用属 AX 请求型，写入/读出数据在内存的地址由 (ES) :(BX) 指出，读/写的字节数由 CX 指出，实际读出的字节数记录在 DX 中。读/写数据块开始时马达自动启动，读/写块结束时马达自动停止。AH 的请求模式是：

(AH) = 0　开盒带机马达

(AH) = 1　关盒带机马达

(AH) = 2　读块。将(CX)个字节读进(ES) :(BX)为始址的内存区，实际读出的字节数存在 DX 中，盒带机状态放在 AH 中。

(AH) = 3　写块。将(DS) :(BX)为始地址的 (CX) 个字节写入磁带。盒带机的状态由(AH)返回。

状态字节(AH 返回的)格式：

(AH) = 00　无错

(AH) = 01　读块时 CRC 错

(AH) = 02　数据未变换

(AH) = 04　无引导段

(AH) = 80　命令非法

二、写盒带机

WRITE BLOCK 子程序负责将一个数据块写入磁带。数据块的结构见下图。图中的引导头是 256 字节的全 1，同步字符为 ASCII 同步字符(X'16')，每个数据块为 256 字节，带 2 个 CRC 校验字节。

写块过程是：WRITE BLOCK 首先启动盒带机马达和同步位(0)，接着写 256 字节全 1 的引导头，然后正式写入一个或多个数据块，每写完一个数据块 WRITE BLOCK 就插入 2 个字节的 CRC 校验字节。对数据块中的每个字节，WRITE BLOCK 调用 WRITE BYTE 子程序，由它把字节拆成 8 位并逐一写入磁带。若字节中某位为 0，就写进一个周期是 500 毫微

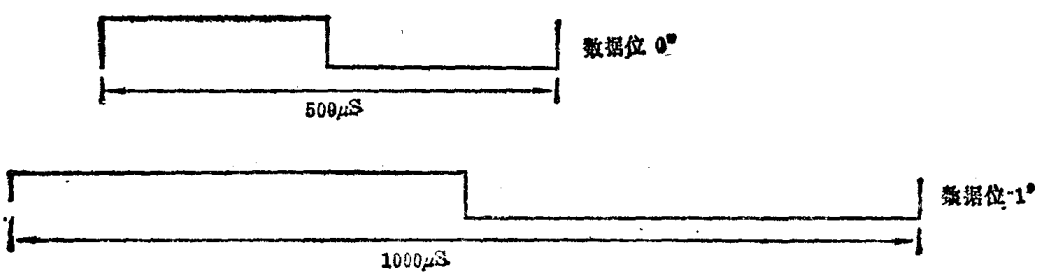


图 29-2　磁带数据位的格式

秒的方波，为1则写进一个周期是1000毫微秒的方波，这些方波均由处于模式3的第2计时器输出。如果待写入的数据字节数不是256的倍数，则在写完最后数据块的最后字节后，WRITE BYTE 重复写最后字节凑足256个字节。最后，写入4个字节的全1作为结尾。WRITE BYTE 不报告出错情况。

三、读盒带机

READ BLOCK 程序负责读取数据块。READ BLOCK 首先启动录音机马达，延迟大约0.5秒后寻找引导头，在读出1/4引导头长度的全1后 READ BLOCK 开始寻找同步字节，如果没有找到正确的同步字符 (X'16')，READ BLOCK 重新再找引导头。找到同步字符后 READ BLOCK 逐位读出数据并进行装配，每装配完一个字节就把它送入(ES):(BX)指示的单元把 BX 加1。

读完256个字节后，READ BLOCK 读磁带上的 CRC 字节，将它与产生的 CRC 相比较，若不等，说明有 CRC 错，在把进位位置1清 AH 后程序退出，这时(DX)是实际读出的字节数。注意在读盒带机时，时钟中断 IRQ 0 是被禁止的。

四、错误恢复

读写磁带时子程序利用周期冗余法(CRC)检查数据块。该 CRC 法使用的多项式 $G(X)=X^{16}+X^{12}+X^5+1$。写磁带时每个数据位从用软件实现的 CRC 寄存器通过，写完一个数据块 CRC 寄存器的补码值被写入磁带。读出一数据块后，磁带上的 CRC 字节将与生成的 CRC 寄存器内容相比较，不等表示有 CRC 错。

§29.3 键盘的译码及用法

ROM BIOS 中的键盘子程序负责将键盘扫描码转换成“扩充 ASCII 码”。“扩充 ASCII 码”是占据一个字节的 ASCII 扩展码，用来表示键盘功能或由键盘子程序、中断处理的功能。

一、字符代码

字符代码表(表29-2)列出的字符代码均由 BIOS 键盘子程序 (放在 AL) 传递给发 INT

表29-2 字符代码表

KEY*	Base CASE	UPPER CASE	CTRL	ALT
1	ESC	ESC	ESC	—1
2	1	1/2	—1	注 1
3	2	@	NUL(000)Note 1	注 1
4	3	*	—1	注 1
5	4	$	—1	注 1
6	5	%	—1	注 1
7	6	^	RS(030)	注 1
8	7	8	—1	注 1
9	8	*	—1	注 1
10	9	(	—1	注 1
11	0	)	—1	注 1
12	—	—	US(031)	注 1
13	=	+	—1	注 1

续表

14	Backspace(008)	Backspace (008)	DEL(127)	−1
15	——→(009)	←——(Note1)	−1	−1
16	q	Q	DCI(017)	注 1
17	w	W	ETB(023)	注 1
18	e	E	ENQ(005)	注 1
19	r	R	DC2(018)	注 1
20	t	T	DC4(020)	注 1
21	y	Y	EM(025)	注 1
22	u	U	NAK(021)	注 1
23	i	I	HT(009)	注 1
24	o	O	SI(015)	注 1
25	p	P	DLE(016)	注 1
26	[	{	ESC(027)	−1
27	]	}	GS(029)	−1
28	CR	CR	LF(010)	−1
29 CTRL	−1	−1	−1	−1
30	a	A	SOH(001)	注 1
31	s	S	DC3(019)	注 1
32	d	D	EOT(004)	注 1
33	f	F	ACK(006)	注 1
34	g	G	BEL(007)	注 1
35	h	H	BS(008)	注 1
36	j	J	LF(010)	注 1
37	k	K	VT(011)	注 1
38	i	L	FF(012)	注 1
39	:	:	−1	−1
40	`	"	−1	−1
41		~	−1	−1
42 SHIFT	−1	−1	−1	−1
43	/	|	FS(028)	−1
44	z	Z	SUB(026)	注 1
45	x	X	CAN(024)	注 1
46	c	C	ETX(003)	注 1
47	v	V	SYN(022)	注 1
48	b	B	STX(002)	注 1
49	n	N	SO(014)	注 1
50	m	M	CR(013)	注 1
51	)	<	−1	−1
52	.	>	−1	−1
53	/	?	−1	−1
54 SHIFT	−1	−1	−1	−1
55	*	(Note 2)	(Note 1)	−1
56 ALT	−1	−1	−1	−1
57	SP	SP	SP	SP
58 CAPS LOCK	−1	−1	−1	−1
59	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
60	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
61	NUL(注1)	NUT(注1)	NUL(注1)	NUL(注1)
62	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
63	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
64	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
65	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
66	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
67	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
68	NUL(注1)	NUL(注1)	NUL(注1)	NUL(注1)
69 NUM LOCK	−1	−1	Pause (注2)	−1
70 SCROLL LOCK	−1	−1	Break (注2)	−1
71	7	Home(注1)	注 1	Clear Screen
72	8	↑ (注1)	注 1	−1
73	9	PageUp(注1)	注 1	Top of Text & Home
74	−	−	−1	−1
75	4	← (注1)	注 1	Reverse Word(注1)
76	5	−1	注 1	−1
77	6	→ (注1)	注 1	Adv Word(注1)
78	+	+	−1	−1
79	1	End(注1)	注 1	Erase to EOL(注1)
80	2	↓ (注1)	注 1	−1
81	3	PageDown(注1)	注 1	Erase toEOS(注1)
82	0	INS	注 1	−1
83		DEL(注1、2)	注 2	注 2

注 1：请参阅本章"扩充码" 2：请参阅本章"特别处理"

16 H 软中断的程序，表中的-1 表示键盘子程序不允许同时按下几个键，71～82 号键在 NUMLOCK (或移位)态、CTRL 态时的大写字母是没有意义的，移位键会暂时翻转 NUM LOCK 的当前状态。

二、扩充码

1．扩充功能

扩充码是表示一些不能由标准 ASCII 码表示功能的代码。扩充码由两个字节组成，第一个字节是 NUL，第二字节是表示扩展功能的扫描码。扩充码由键盘子程序根据按键的组合状态生成并返回，返回时 NUL 字符放在 AL 寄存器，扩充码放在 AH 寄存器。下面是扩充码一览表。

表 29-3　扩充码及功能一览表

第二码字节(扩充码)	功　　能
3	NUL Character
15	←
16-25	ALT Q,W,E,R,T,Y U,I,O,P
30-38	ALT A,S,D,F,G,H,J,K、L
44-50	ALT Z,X,C,V,B,N,M
59-68	F1-F10 功能键(小写字母)
71	Home(复位)
72	↑
73	Page up & Home Cursor(上页、光标复位)
75	←
77	→
79	End
80	↓
81	Page Down & Home Cursor(下页、光标复位)
82	INS
83	DEL
84-93	F11-F20 (大写字母的 F1-F10)
94-103	F21-F30(CTRL F1-F10)
104-113	F31-F40(ALT F1-F10)
114	CTRL PRTSC(开始/停止打印) Key 55
115	CTRL←退字
116	CTRL→进字
117	CTRL END 擦除 EOL
118	CTRL PG DN 擦除 EOS
119	CTRL HOME 清屏幕和复位
120-131	ALT1,2,3,4,5,6,7,8,9,0,-,=(Keys 2-13)
132	文本屏幕上卷 25 行，光标复位

2．移位状态

大多数移位状态由键盘子程序处理，他们对系统和应用程序均是透明的，下面逐一介绍移位态的产生方法。

(1) 按移位键(Shift)：暂时将键 2～13、15～17、30～41、43～53、55、59～68 置成大写字母态(CAPSLOCK 态还是小写字母)，暂时翻转键 71～73、75、79～83 的 NUMLOCK/NO-

NUMLOCK 态。

(2) 按 CTRL 键：暂时将键 3、7、12、14、16～28、30～38、43～50、55、59～71、73、75、77、79、81 置为 CTRL 态，CTRL 键与 ALT、DEL 键合用可让系统复位、与 SCROLL LOCK 键合用产生"阻断"效果，与 NUMLOCK 合用使系统"暂停"。

(3) 按 ALT 键：暂时将键 2～13、16～25、30～38、44～50 和 59～68 置成 ALT 态，ALT 与 CTRL、DEL 键合用使系统复位。

ALT 键还有一个特殊的功用：让用户从键盘输入字符代码(0～255)。具体输入方法是：先按 ALT 键，再按三个数字键 (71～73、75～77、79～83) 输入字符代码值，最后释放 ALT 键。如果打入的数字超过三个，系统取最后三位。键盘子程序把这三个数字看成一个字符码，并把它传给系统程序或应用程序。

(4) 按 CAPSLOCK 键：将键 10～25、30～38、44～45 置成大写字母态，再按CAPSLOCK 键又把大写字母态翻转成小写字母态。

(5) 按 NUMLOCK 键：将键 71～73、75～77、79～83 置成数字态，再按 NUMLOCK 键翻转成原来的状态。

(6) 按 SCROLL LOCK 键：特别应用程序将该键迫使光标控制键的功能是给文本加框而不是移动光标，第二次按 SCROLL LOCK 键这些光标控制键翻转回原来的功能。键盘子程序只记录该键的当前状态，实际加框动作由应用程序或系统程序完成。

3．移位键的优先级别和混合使用。

如果 ALT、CTRL、SHIFT 键被同时按下，且这时只有一个键有效的话，键盘子程序处理具有最高优先级的键，ALT、CTRL、SHIFT 键的优先级顺序是：ALT 最高，CTRL 其次，SHIFT 最低。只有 ALT 和 CTRL 键能同时按下，混合使用。

三、特别处理

1．系统复位：同时按 ALT、CTRL、DEL (键 83)，使键盘子程序进行系统初始化工作。

2．阻断(BREAK)：同时按 CTRL、BREAK 键后，键盘子程序发出中断-1A，同时在 AX 寄存器返回全 0 的扩充字符码。开机初始程序把这个中断的处理过程删去了。如果用户要使用 BREAK 中断的话，必须在开机程序后用系统程序或应用程序段修改 -1A 向量。

3．暂停(Pause)：同时按 CTRL、NUMLOCK 键后键盘子程序进入循环，等待用户按其它键 (NUMLOCK 键除外)。这样用户就能暂停打印、列表等操作。"Unpause"键不起作用。"暂停"由键盘子程序完成。

4．下列键的打印动作可能会被键盘子程序取消：

CTRL、SHIFT、ALT、NUMLOCK、SCROLL-LOCK、CAPS LOCK、1NS。

5．打印屏幕：同时按 SHIFT-PRINT、SCREEN (键 55) 后，屏幕打印程序响应中断，打印出当前整帧屏幕。打印程序工作在字母/图形模式，不能识别的字符以空格代替。

键盘子程序有自己的缓冲区，可应付一个熟练打字员打入的字符，但尚若在缓冲区满后还打入字符，这个字符将丢失，系统响铃告警。

表 29-4 键盘及键的常用功能表

功　　能	键	使　用　场　合
光标返回出发点	Home	编辑、字处理
返回到最外层 menu	Home	Menu 驱动的应用程序
上移光标	↑	全屏幕编辑、字处理
上页，向后卷 25 行并返回出发点	PGUP	编辑、字处理
左移光标	←键75	文本、命令输入
右移光标	→	文本、命令输入
卷至文本尾，光标停在最后一行	END	编辑、字处理
下移光标	↓	全屏幕编辑、字处理
下页，向前卷 25 行并返回出发点	PGDN	编辑、字处理
在光标处开始或结束插入工作右移冲缓区里的正文	INS	文本及命令输入
删去光标处的字符	DEL	文本、命令输入
去除退格	←键14	文本、命令输入
标记向右移	→	文本输入
标记向左移	←	文本输入
清屏幕、返回出发点	CTRL HOME	命令输入
上卷	↑	处 SCROLL LOCK 态
下卷	↓	处 SCROLL LOCK 态
左卷	←	处 SCROLL LOCK 态
右卷	→	处 SCROLL LOCK 态
清光标至行尾的全部字符	CTRL END	文本、命令输入
退出/换码	ESC	编辑或处 1 层 Menu 等
开始/停止打印屏幕	PRTSC CTRL 键 55	任何场合
删去光标至 EOS 处的全部字符	CTRL PG DN	文本、命令输入
进字	CTRL→	文本输入
退字	CTRL←	文本输入
右移窗口	CTRL→	文本太宽屏幕容纳不了
左移窗口	CTRL←	文本太宽屏幕容纳不了
进入插入模式	INS	行编辑
退出插入模式	INS	行编辑
删去当前行	ESC	命令、文本输入
挂起系统(暂停)	CTRL NUMLOCK	停止列表或程序执行等工作，按任何键则继续
中断	CTRL BREAK	中断当前处理

续表

系统复位	ALT CTRL DEL	重新启动引导程序
取文件头、光标返回出发点	CTRL PGUP	编辑、字处理
标准功能键	F1-F10	基本功能键
扩充功能键	SHIFT F1-F10 CTRL F1-F10 ALT F1-F10	扩充功能键
附加功能键	ALT 键 2-13 (1-9,0,—,=)	键盘上边沿的 Striker 被按下
附属功能键	ALT A-Z	用于某功能程序的开始字母与键字母相同时

表 29-5 BASIC 屏幕编辑程序用键功能

功 能	键	功 能	镜
回车	←	插入	INS
换行	CTRL←	删除	DEL
响铃	CTRL G	清屏幕	CTRL HOME
返回出发处	HOME	暂停输出	CTRL NUMLOCK
上移光标	↑	进标记	→
下移光标	↓	停止执行 (Break)	CTRL BREAK
左移光标	←	删去当前行	ESC
右移光标	→	光标移至行尾	END
进一字	CTRL→	删至行尾的全部字符	CTRL END
退一字	CTRL←		

表 29-6 DOS 用键的特殊功能

功 能	键	功 能	键
持起	CTRL NVMLOCK	拷贝剩余字符	F3
打印屏幕	CTRL PRTSC	跳过字符	DEL
停止打印屏幕	CTRL PRTSC	跳直至匹配	F4
退出当前功能(break)	CTRL BREAK	进入插入模式	INS
退格	←键14	退出插入模式	INS
换行	CTRL←	为样板加上新行	F5
删行	ESC	生成代换(REPLACE)	F6
拷贝字符	FI 或→	中的字符串分隔符	
拷贝直至匹配	F2	键盘输入文件结束	F6

表 29-7 低地址存贮区映射(0-'0600'X)

1. 中断向量(0-7F)

地 址	中断向量	用 途	地 址	中断向量	用 途
0-3	0	除数为 0 错	4C-4F	13	软盘机 I/O 调用
4-7	1	单步	50-53	14	RS232 I/O 调用
8-B	2	不可屏蔽中断 NMI	54-57	15	盒带机 I/O 调用
C-F	3	断点指令('CC'X)	58-5B	16	键盘 I/O 调用
10-13	4	溢出	5C-5F	17	打印机 I/O 调用
14-17	5	打印屏幕	60-63	18	ROM-BASIC 入口
18-1F	6,7	保留	64-67	19	引导装入程序
20-23	8	计时器中断(18.2/秒)	68-6B	1A	日时调用
24-27	9	键盘中断	6C-6F	1B	*键盘阻断时得到控制权
28-37	A、B、C、D	保留	70-73	1C	时钟中断时得到控制*权
38-3B	E	软盘机中断	74-77	1D	**指向 CRT 初始参数表
3C-3F	F	保留	78-7B	1E	**指向盒带参数表
40-43	10	显示器 I/O 调用	7C-7F	1F	1KB 图形模式 CRT 用第 128 至 256 号字符(ASCII 码) 发生表指针,空缺为 0:0
44-47	11	设备检查调用			
48-4B	12	存贮器检查调用			

* 被开机初始程序指向 IRET 中断返回指令

** 被开机初始程序指向 ROM 区的表格

2. BASIC 及 DOS 保留中断(80-3FF)

地 址	中断向量	用 途	地 址	中断向量	用 途
80-83	20	结束 DOS 程序	9C-9F	27	DOS 结束、固定在存贮区
84-87	21	DOS 函数调用	A0-FF	28-3F	DOS 保留
88-8B	22	DOS结束地址	100-1FF	40-7F	未用
8C-8F	23	DOSCTRL-BRK 退出地址	200-217	80-85	BASIC 保留
90-93	24	DOS 致命错向量	218-3C3	86-F0	BASIC 解释程序用
94-97	25	DOS 绝对磁盘读取	3C4-3FF	F1-FF	未用
98-9B	26	DOS 绝对磁盘写入			

3. 存贮区保留地址

地 址	所 属 软 件	用 途
400-48 F	ROM BIOS	见 BIOS 列表文件
490-4CF	DOS	DOS 命令用
4DO-4EF		保留
4FO-4FF		为内部应用通讯区保留
500-5FF		为 DOS 及 BASIC 保留
500	DOS	记录打印屏幕过程中的状态信息 0—打印屏幕未动作或打印屏幕成功 1—正打印屏幕 255—打印屏幕时发生错误
504	DOS	单驱动模式状态信息字节
510-511	BASIC	BASIC 段地址记录区
512-515	BASIC	时钟中断向量段地址：偏差地址
516-519	BASIC	断键中断向量段地址:偏差地址
51A-51D	BASIC	磁盘中断向量段地址:偏差地址

4. BASIC 工作区变量

下面是进行定义缺省工作区段(DEF SEG)后一些单元记录内容的含义

含　　义	地 址 偏 差	长　　度
当前被处理行的行号	X'2E'	2
最后错误出现行的行号	X'347'	2
程序正文首在段中的偏差地址	X'30'	2
变量区首的偏差地址 (程序文本1结束处—1)	X358'	2
键盘缓冲区内容 0—缓冲区内无字符 1—缓冲区内有字符 用 POKE & H6A,0 可将缓冲区内填满字符	X'6A'	1

例子

```
100  Print  PEEK(& H2E) + 256*PEEK (& H2F)
)                L                H
100              X'64'            X'00'
```

第三十章 BIOS 列表文件

本章列出BIOS源文件及它的中、英文注释。整个BIOS源文件分成五大部分。第一部分是常数、变量定义区，在该区定义各I/O口子的地址、8088中断向量表，记录外设状态的数据区，占源文件1～189行；第二部分是15个开机检测程序和引导装入程序。这些程序将检查整个系统各组成部分，保证他们能正常工作，占源文件190—1409行；第三部分是I/O支持程序。它们均通过软中断的形式加以调用，占源文件1410—4901、4976—5501、5641—5940行；第四部分是获取系统设备配置情况的程序，他们也由软中断方式调用，占源文件4902—4975行，最后部分是处图形模式(APA式)显示器用的第一字形发生表，存贮第1至128号字符的8×8点阵，它占源文件5503—5640行。整个BIOS代码被固化有一片8KB ROM存贮模块里，不能改动。BIOS的起始地址是0F000H。本章先后顺序安排如下：一、BIOS各子程序简要说明表；二、15个开机检测程序的框图；三、BIOS源文件。

一、BIOS 各子程序简要说明表

调 用 名	起始行号	软中断号	主入口参数	功 能
STGTST	220			测试16 KB RAM 区
RESET	266			测试8088 CPU
	324			ROM 区测试
	352			初始及测试 DMA 控制器(8237)
	444			基本16K RAM 区测试
	568			8259A 中断控制器测试
	624			8253 计时器测试
	651			暂时中断服务子程序
	743			检查存贮盒带 BASIC ROM 区
	764			初始、启动 6845 CRT 控制器、测试 CRT 显示缓存
	831			CRT 接口线测试
	998			键盘测试
	1052			盒带机测试
	1096			软盘机连接测试
ERR-BEEP	1225			鸣响蜂鸣器
KBD-RESET	1282			复位键盘
BLINT-INT	1310			闪烁 LED
P-MSG	1335			在 CRT 上显示一字符
BOOT-STRAP	1371	INT19		从系统软盘机盘调入引导程序并把控制权交给它，若无系统软盘机，控制权转给盒带 BIOS
RS232-IO	1470	INT14	(AH)=0	初始 RS232 通讯口
RS232-IO	1470	INT14	(AH)=1	将 AL 中数据发向通讯线
RS232-IO	1470	INT14	(AH)=2	从通讯线接收字符，交 AL 返回
RS232-IO	1470	INT14	(AH)=3	取通讯口状态

续表

调 用 名	起始行号	软中断号	主入口参数	功 能
KEYBOARD-IO	1660	INT16	(AH)=0	读取下一键入字符，交 AL 送回
KEYBOARD-IO	1660	INT16	(AH)=1	检查是否有字符可读
KEYBOARD-IO	1660	INT16	(AH)=2	返回移位键当前状态
DISKETTE-IO	2302	INT13	(AH)=0	复位驱动器
DISKETTE-IO	2302	INT13	(AH)=1	取驱动器状态
DISKETTE-IO	2302	INT13	(AH)=2	读数个扇区数据并送至内存
DISKETTE-IO	2302	INT13	(AH)=3	写数个扇区
DISKETTE-IO	2302	INT13	(AH)=4	检验扇区
DISKETTE-IO	2302	INT13	(AH)=4	格式化磁道
NEC-OUTPUT	2628			将一个字节送 NEC 控制器数据口
GET-PARM	2676			从 DISK-BASE 参数区回送一字节
SEEK	2703			将磁头移向指定的磁道
DMA-SETUP	2759			为读/写/检验软盘建立 DMA
CHK-STAT-2	2817			重校、寻道、复位操作后 FDC 中断处理
WAIT-INT	2848			等待 FDC 发中断
DISK-INT	2878			软盘机中断处理
RESULTS	2903			返回 NEC 控制器发出中断后的状态
NUM-TRANS	2973			计算实际读/写的扇区数
PRINTER-IO	3033	INT17	(AH)=0	打印 AL 中的字符
PRINTER-IO	3033	INT17	(AH)=1	初始打印机转接器
PRINTER-IO	3033	INT17	(AH)=2	读取打印机状态
VIDEO-IO	3241	INT10	(AH)=0	设置 CRT 模式
VIDEO-IO	3241	INT10	(AH)=1	设光标类型
VIDEO-IO	3241	INT10	(AH)=2	设光标位置
VIDEO-IO	3241	INT10	(AH)=3	读光标位置
VIDEO-IO	3241	INT10	(AH)=4	读光笔位置
VIDEO-IO	3241	INT10	(AH)=5	选择活动显示页
VIDEO-IO	3241	INT10	(AH)=6	上卷活动显示页
VIDEO-IO	3241	INT10	(AH)=7	下卷活动显示页
VIEFO-IO	3241	INT10	(AH)=8	读当前光标处字符的 ASCII 码和属性符
VIDEO-IO	3241	INT10	(AH)=9	在当前光标处显示新字符
VIDEO-IO	3241	INT10	(AH)=10	在光标处写一新字符
VIDEO-IO	3241	INT10	(AH)=11	为图形模式 CRT 设置色板号
VIDEO-IO	3241	INT10	(AH)=12	在图形模式 CRT 上显示一点
VIDEO-IO	3241	INT10	(AH)=13	读取图形模式 CRT 上的一点
VIDEO-IO	3241	INT10	(AH)=14	写电传
VIDEO-IO	3241	INT10	(AH)=15	取当前 CRT 状态
SET-MODE	3303			设 CRT 模式
SET-CTYPE	3477			设光标类型
SET-CPOS	3511			将光标移向新位置
READ-CURSOR	3546			读光标坐标和模式
ALT-DISP-PAGE	3570			设置活动显示页页号
SET-COLOR	3603			为 320×200 模式置图象模式
VIDEO-STATE	3636			返回 CRT 状态
POSITION	3654			计算 A/N 式下字符在显示缓存的地址
SCROLL-UP	3681			上卷一字符块
SCROLL-DOWN	3801			下卷一字符块

续表

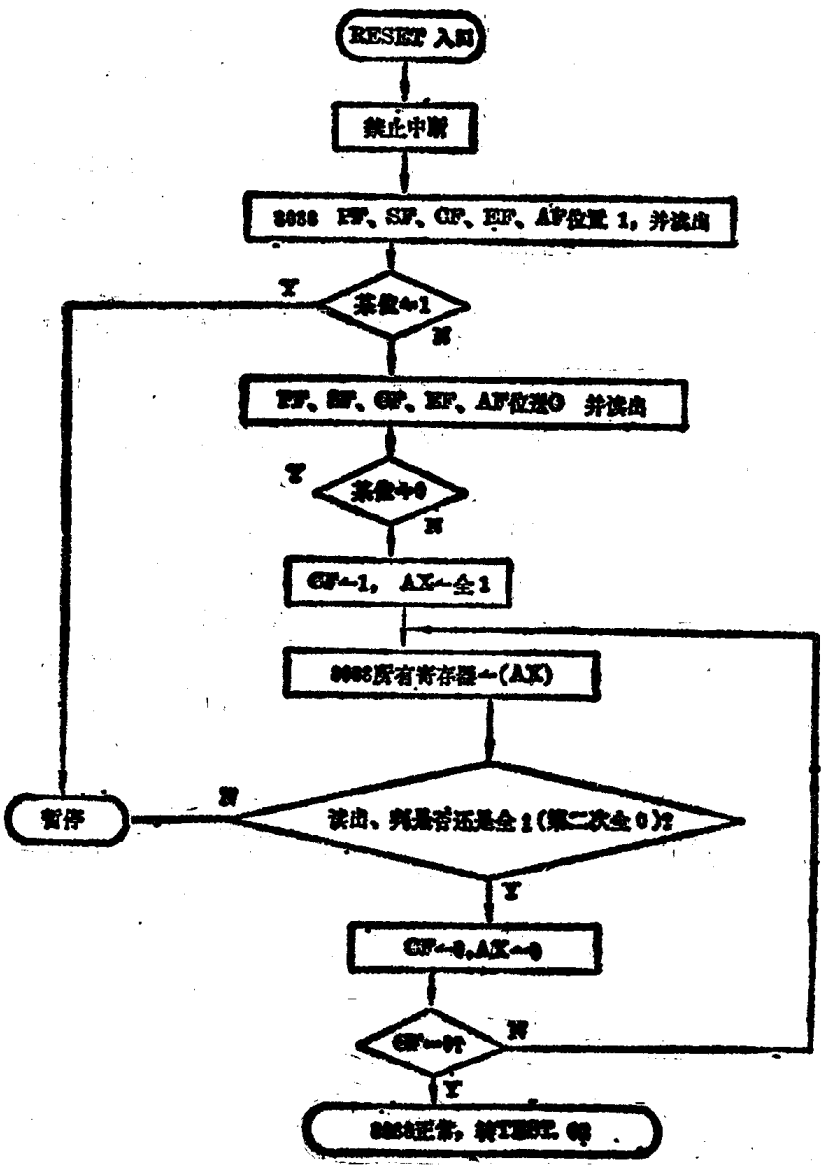
调 用 名	起始行号	软中断号	主入口参数	功 能
READ-AC-CURRENT	3849			返回光标处字符的 ASCII 码和属性码
WRITE-AC-CURRENT	3910			新字符的 ASCII 码和属性码写入显示缓存并在光标处显行出来
WRITE-C-CURRENT	3959			新字符 ASCII 码送显示缓存并在光标处显示出来
READ-DOT	4009			在 APA 式 CRT 上显示一点
WRITE-DOT	4020			在 APA 式 CRT 上显示一点
GRAPHICS-UP	4125			上卷 APA 式 CRT 上的一象点块
GRAPHICS-DOWN	4203			下卷 APA 式 CRT 上的一象点块
GRAPHICS-WRITE	4337			将一 ASCII 字符显示在 APA 式 CRT 上
WRITE-TTY	4672			模仿电传打印机在 CRT 上显示字符
READ-LPEN	4792			测试光笔状态返回光笔坐标
MEMORY-SIZE-DETERMINE	4922	INT12		取内存 RAM 区容量
EQUIPMENT DETERMINATION	4967	INT11		返回系统外设配接记录字
CASSETTE-IO	5002	INT15	(AH)=0	打开录音机马达
CASSETTE-IO	5002	INT15	(AH)=1	关闭录音机马达
CASSETTE-IO	5002	INT15	(AH)=2	从磁带读出数个数据块
CASSETTE-IO	5002	INT15	(AH)=3	向磁带写数据块
READ-BLOCK	5064			读入数个数据块
READ-BYTE	5214			读入一个字节
READ-HALF-BIT	5271			取得输入数据位半个周期的计时值
WRITE-BLOCK	5303			向磁带写数个数据块
WRITE-BYTE	5394			向磁带写一个字节
WRITE-BIT	5418			写一个数据位
CRC-GEN	5458			更新 CRC-REG 单元
BEGIN-OP	5489	INT1A	(AH)=0	读当前时钟
TIMER-OF-DAY	5658	INT1A	(AH)=1	设置时钟
TIMER-INT	5704			时钟中断处理
PRINT-SCREEN	5835	INT5		打印屏幕

二、15个开机检测程序的框图

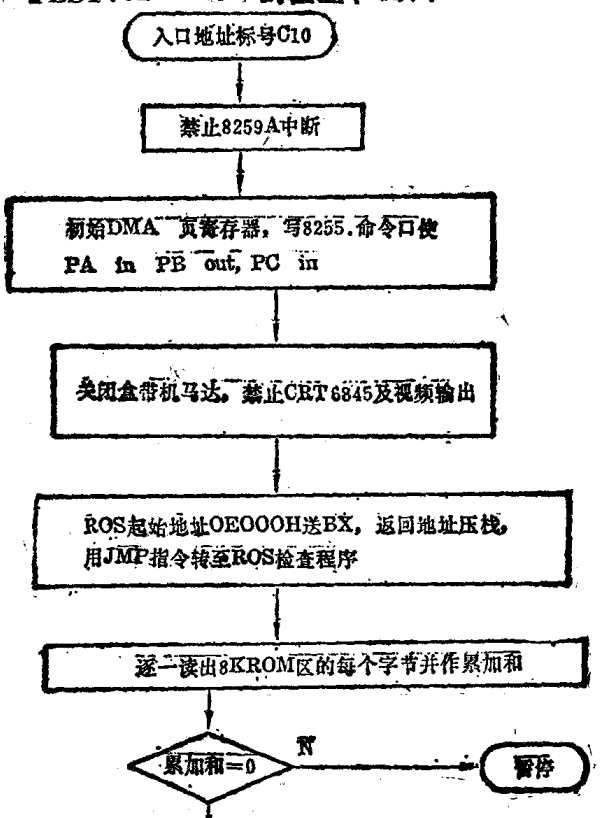
下面是 15 个进行开机后检测系统的粗略框图，其中 Test·04 用到的 STGTST 程序的框图是详尽的。读者在阅读程序前先看一下框图可以建立起程序基本结构及作用的初步概念，使阅读这些汇编程序更容易一些。另外，我们在 BIOS 源程序的英文注解旁加中文注释。这

些注释是在读懂 BIOS 的基础上写成的，十分详细。读者若能将中、英文注释及前面各硬件章节的内容结合起来，从 15 个开机检测程序入手(有框图)，一条指令一条指令地读完整个 BIOS 程序段，就一定能完全弄懂 BIOS 的每个程序、每条指令的含义，进而获得用 Intel 8088/8086 指令设计汇编程序的技巧、思想，了解 BIOS 的基本结构以及它是怎样支持键盘、软盘机、异步通讯口、打印机和显示器的，为以后利用 BIOS 提供的各种软中断服务，并在 BIOS 层上建立用户自己的应用程序乃至扩充 BIOS 打下基础。

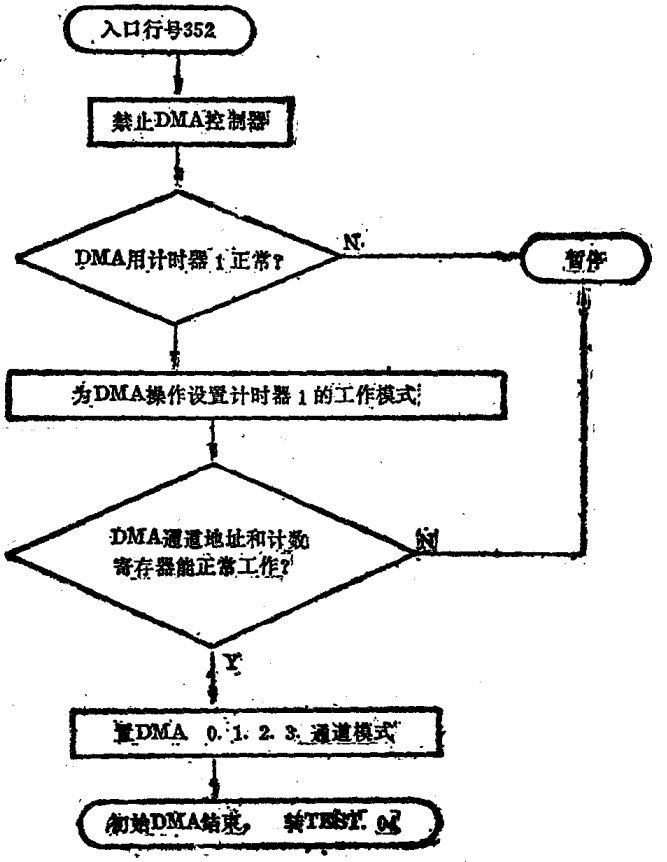
TEST.018088 处理器测试



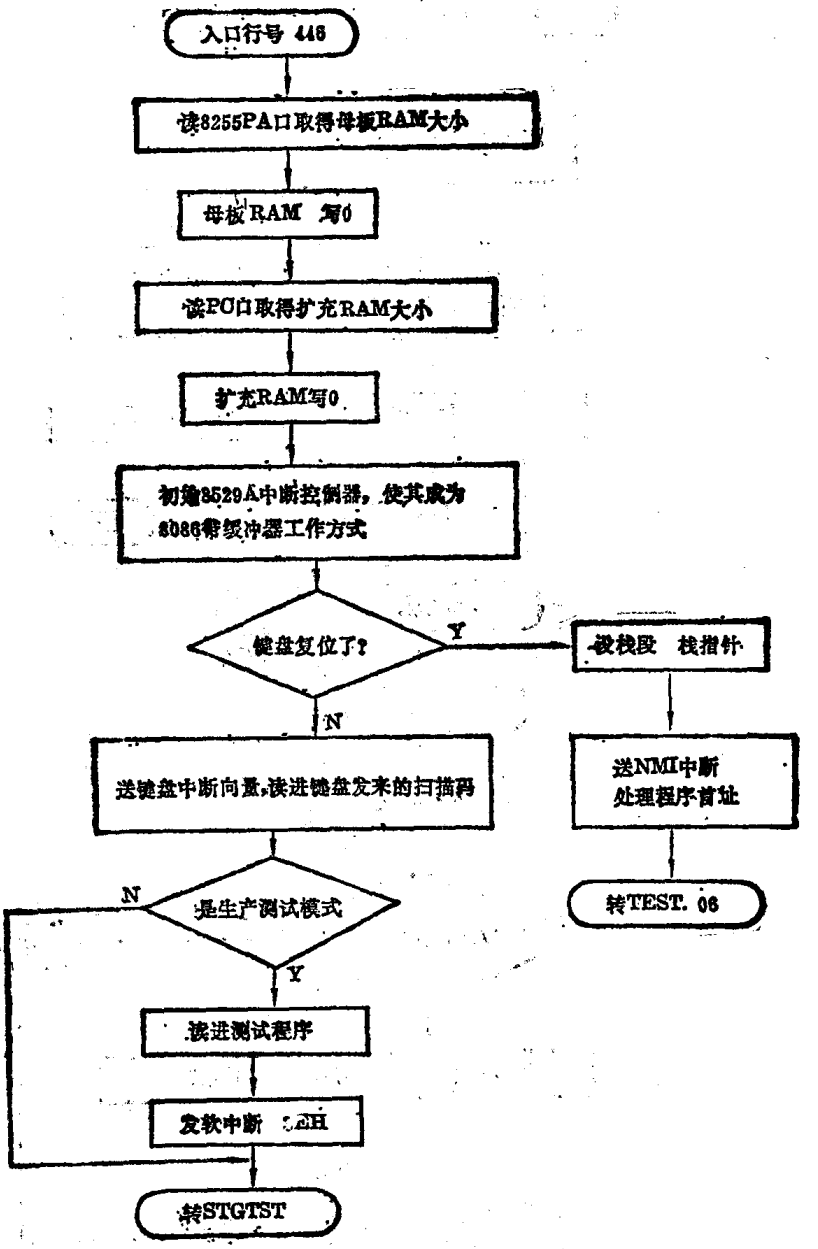
TEST.02 ROS 的检查和测试

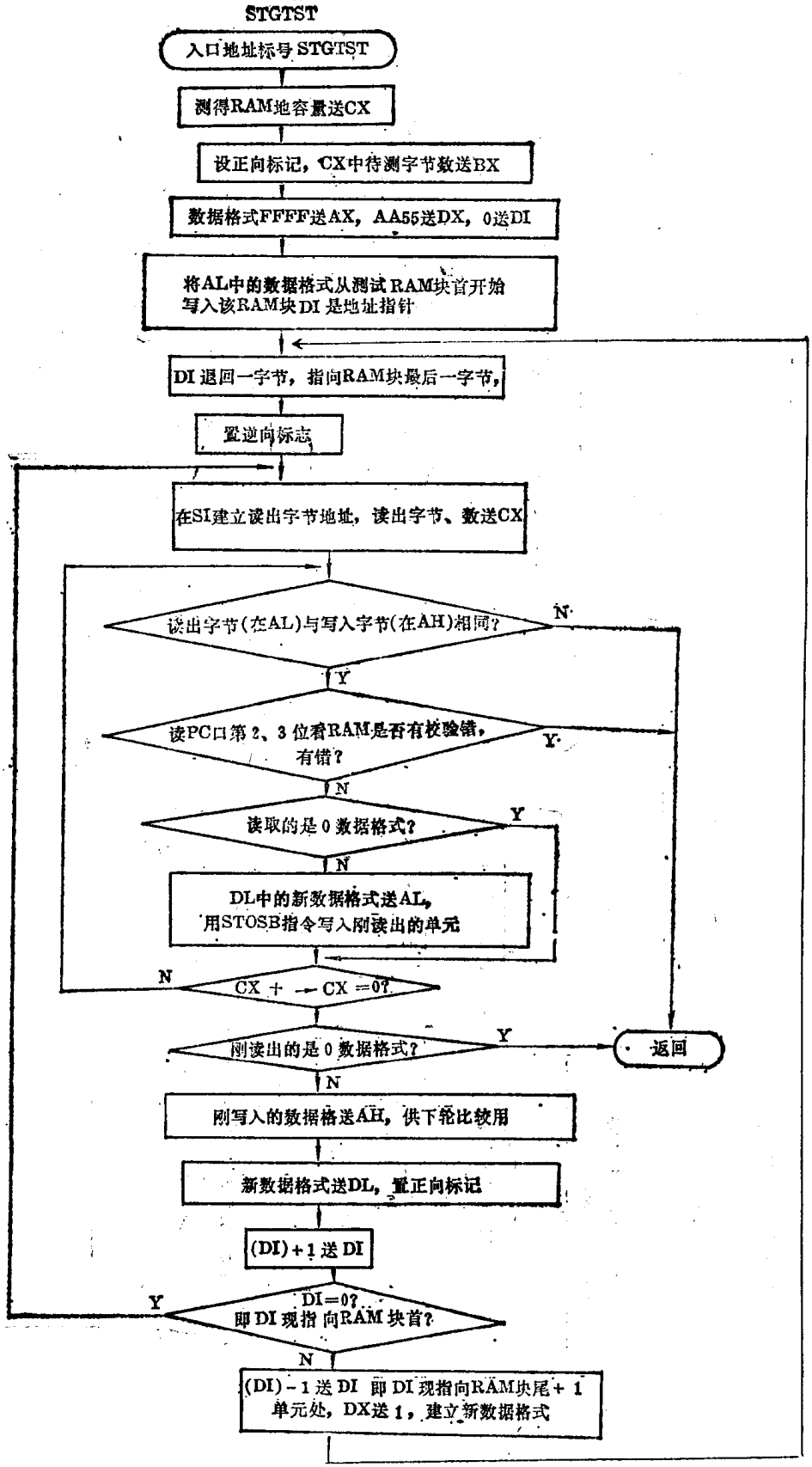


TEST.03 初始 DMA

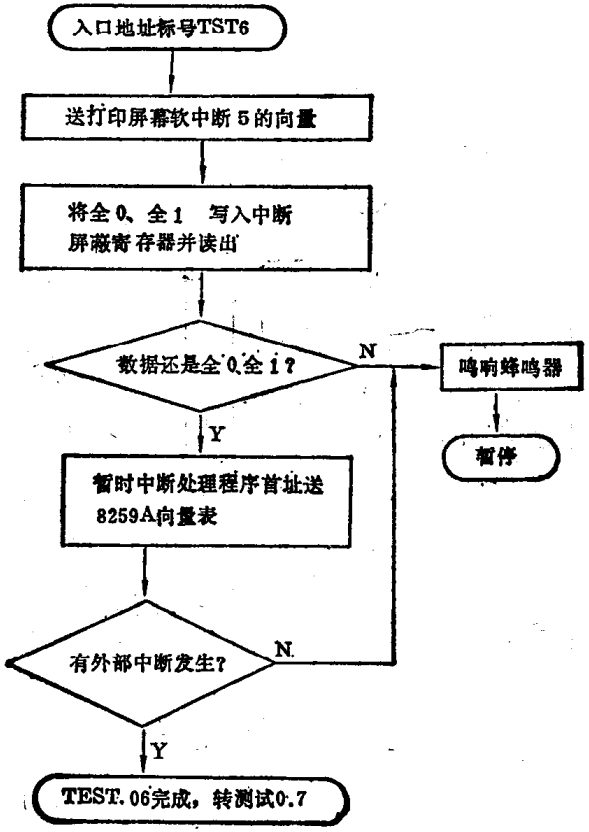


TEST·04　基本 16K RAM 区测试

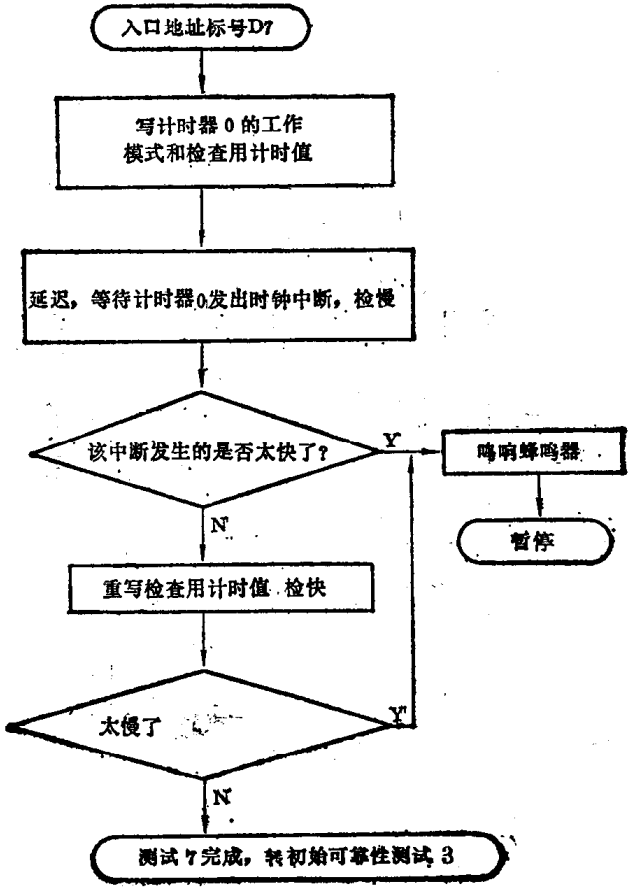



STGTST
入口地址标号 STGTST
测得RAM地容量送CX
设正向标记，CX中待测字节数送BX
数据格式FFFF送AX，AA55送DX，0送DI
将AL中的数据格式从测试RAM块首开始
写入该RAM块 DI 是地址指针
DI 退回一字节，指向RAM块最后一字节，
置逆向标志
在SI建立读出字节地址，读出字节、数送CX
读出字节(在AL)与写入字节(在AH)相同?
N
Y
读PC口第 2、3 位看RAM是否有校验错，
有错?
Y
N
读取的是 0 数据格式?
Y
N
DL中的新数据格式送AL，
用STOSB指令写入刚读出的单元
N
CX + -1 → CX =0?
刚读出的是 0 数据格式?
Y
返回
N
刚写入的数据格送AH，供下轮比较用
新数据格式送DL，置正向标记
(DI)+1 送 DI
Y
DI=0?
即 DI 现指向RAM 块首?
N
(DI)-1 送 DI 即 DI 现指向RAM块尾+1
单元处，DX送 1，建立新数据格式

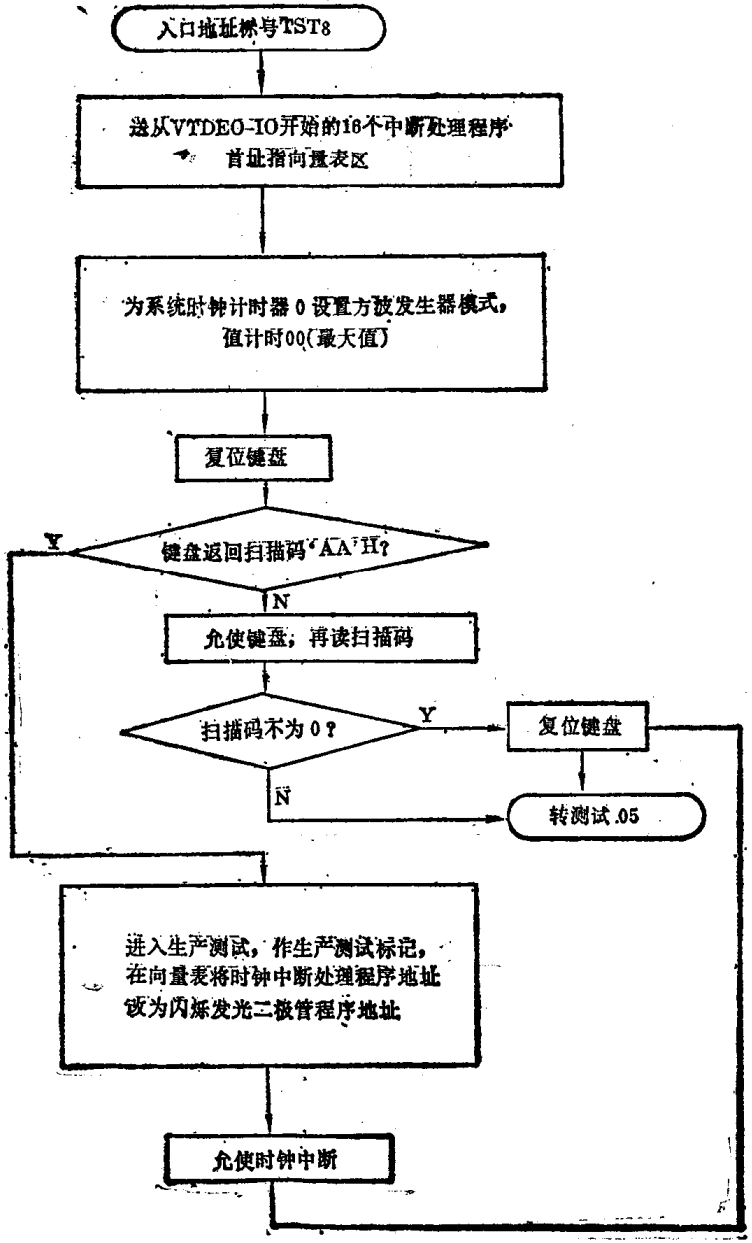
TEST·06　8259A 中断控制器测试



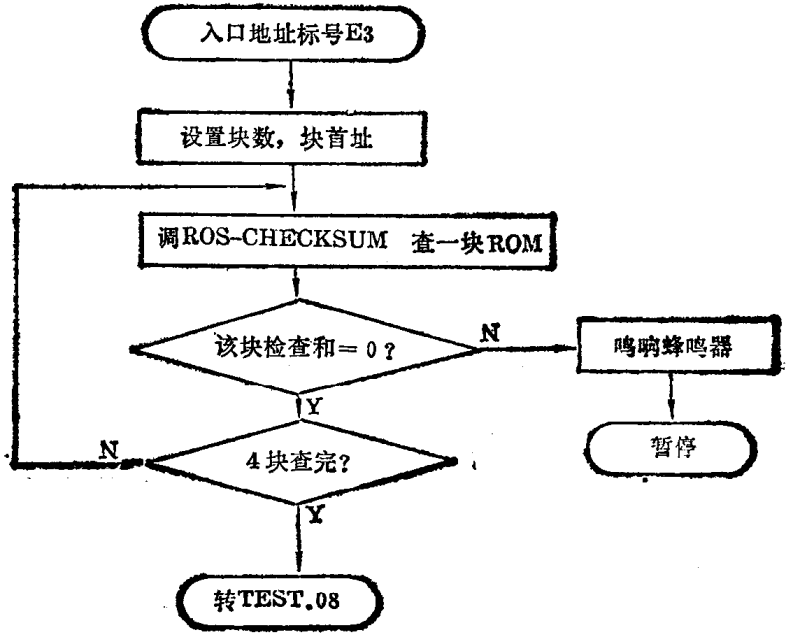
TEST·07　8253 计时器检查



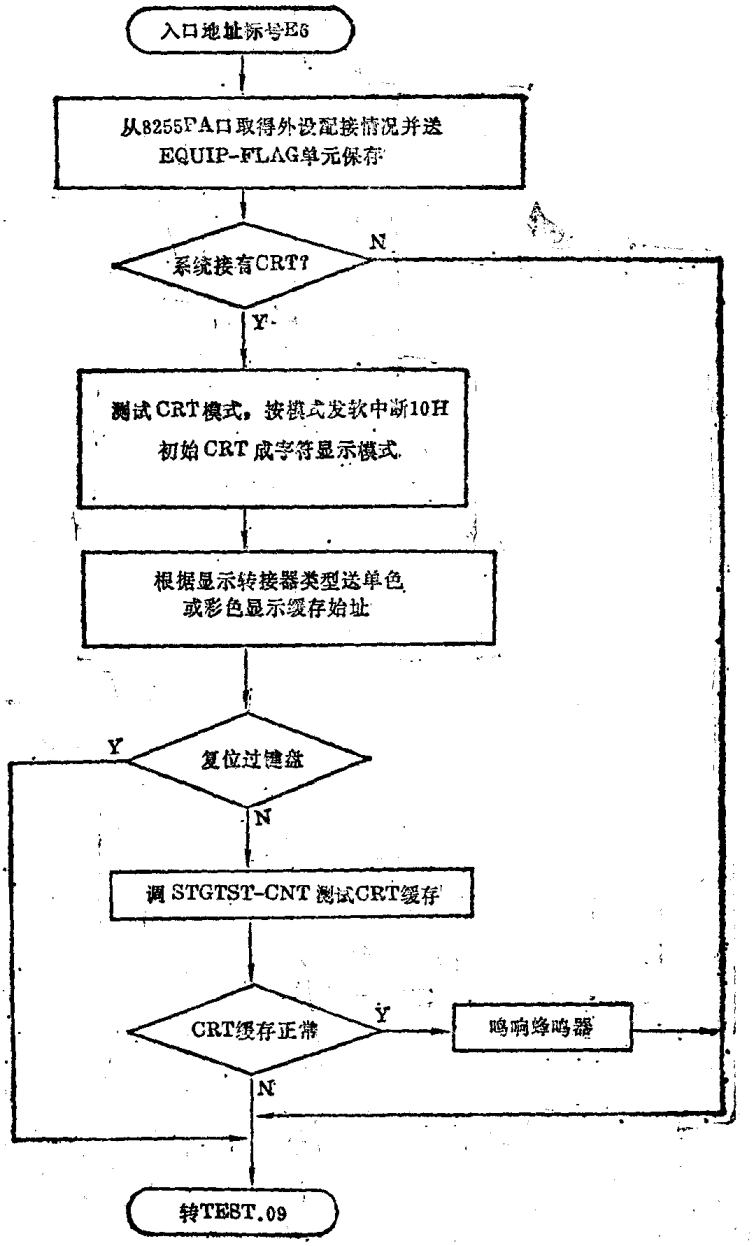
初始可靠性测试 3 建立软中断向量表



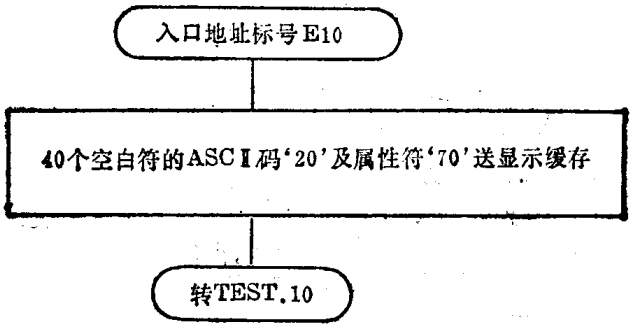
TEST.05 做4个存贮盒带BASIC代码ROM区的检查和



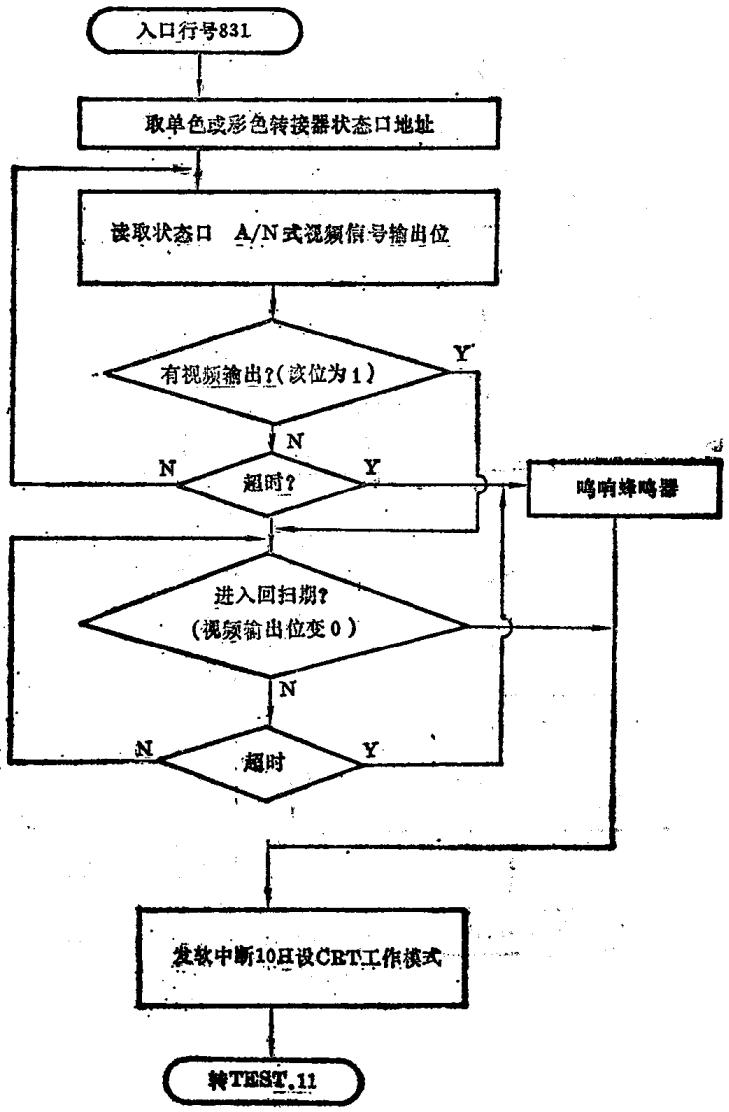
TEST.08　初始并启动 CRT 控制器 6845、测试 CRT 显示缓存



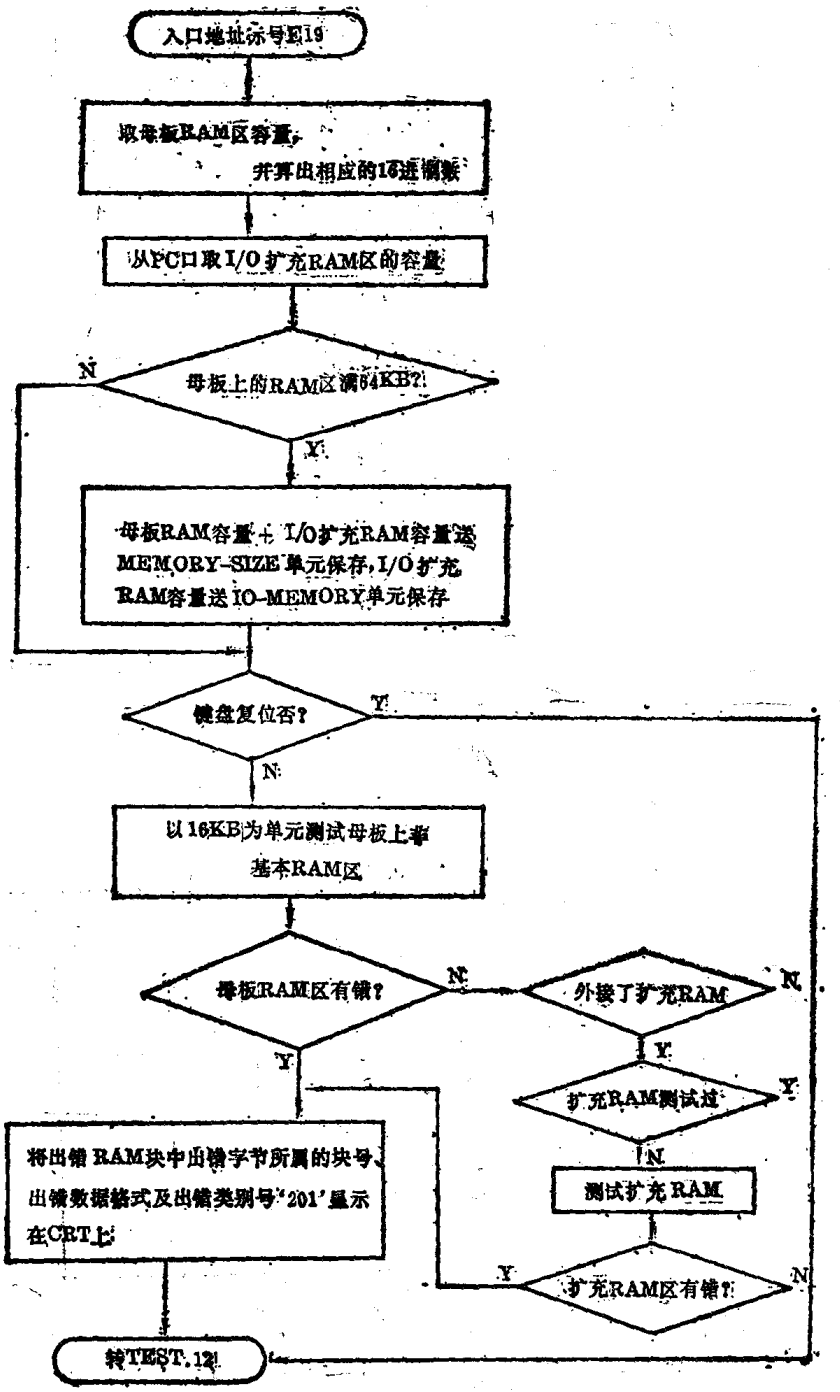
TEST.09　在CRT上显示40个视频信号翻转的空白符



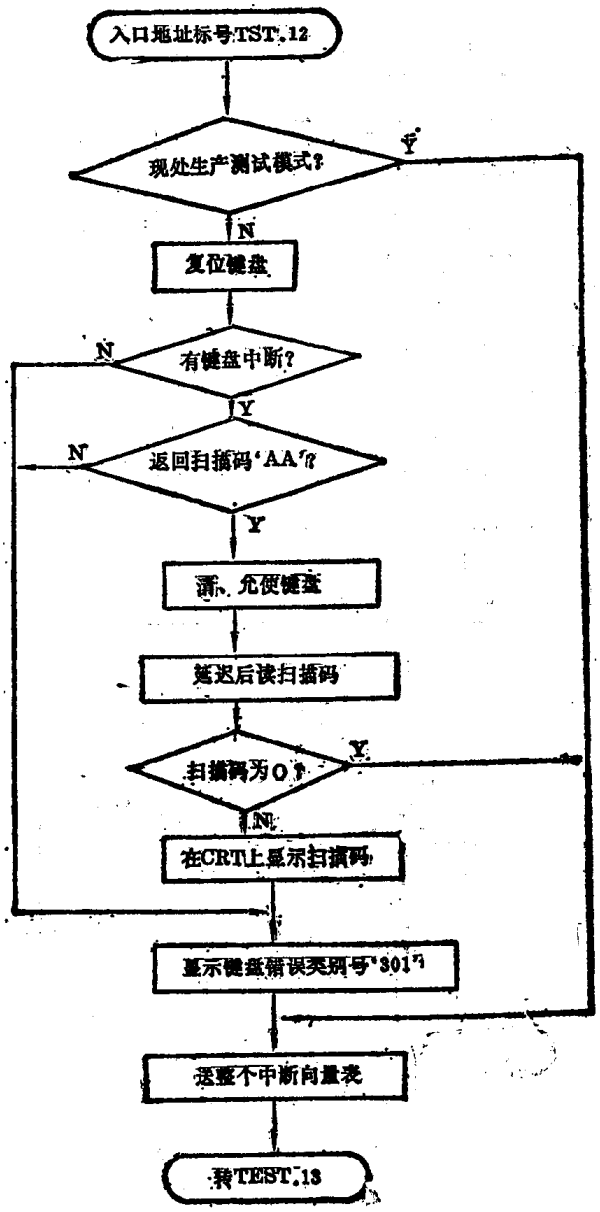
TEST.10 CRT接口线检查



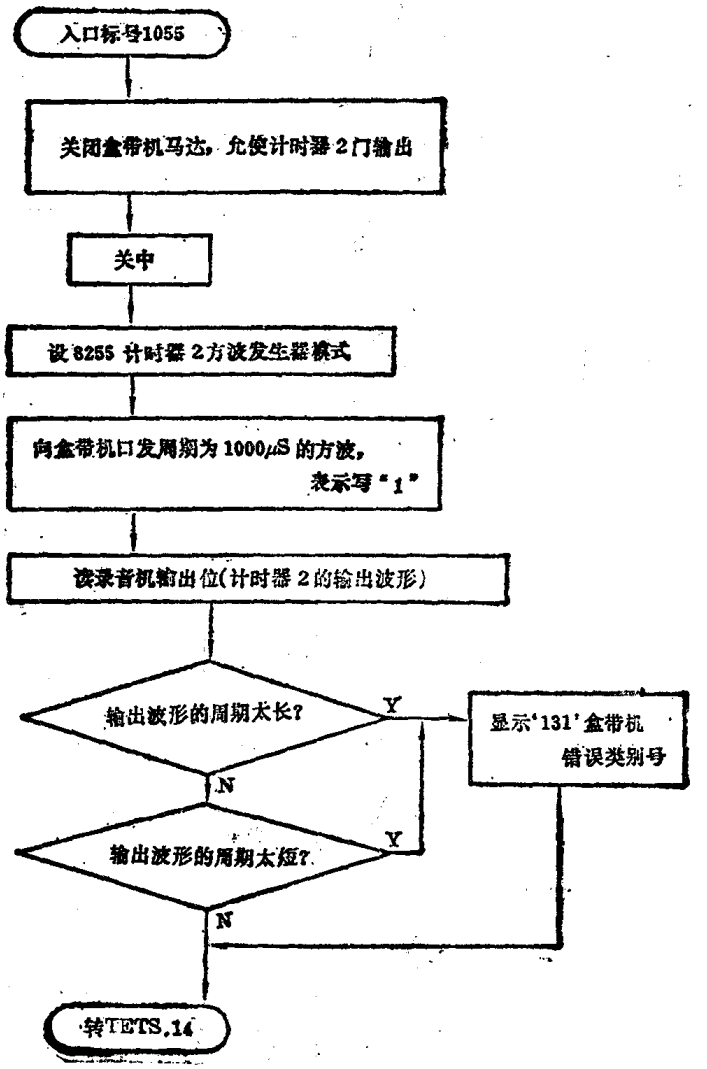
TEST.11 剩余 RAM 区测试



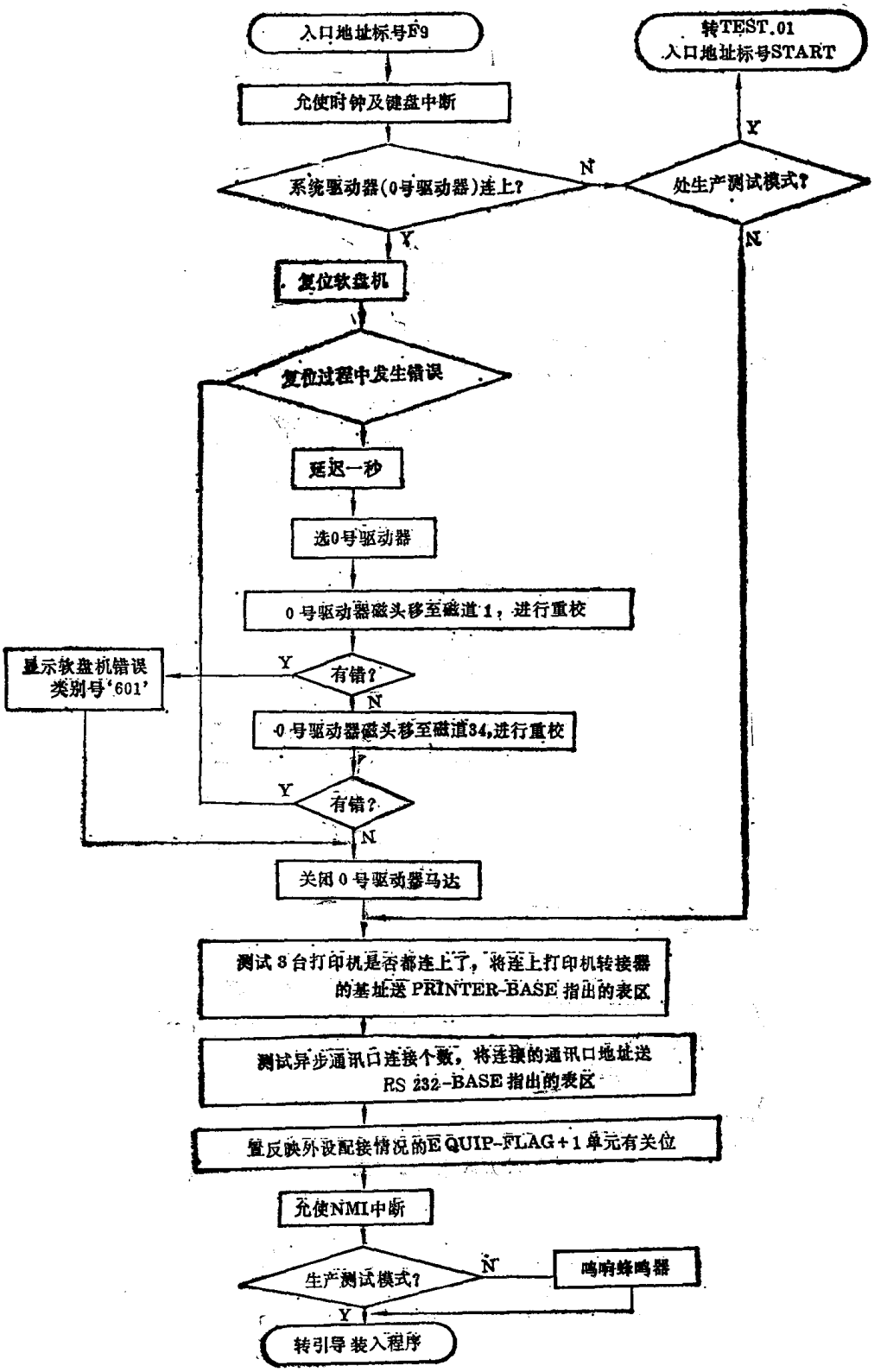
TEST.12　键盘测试

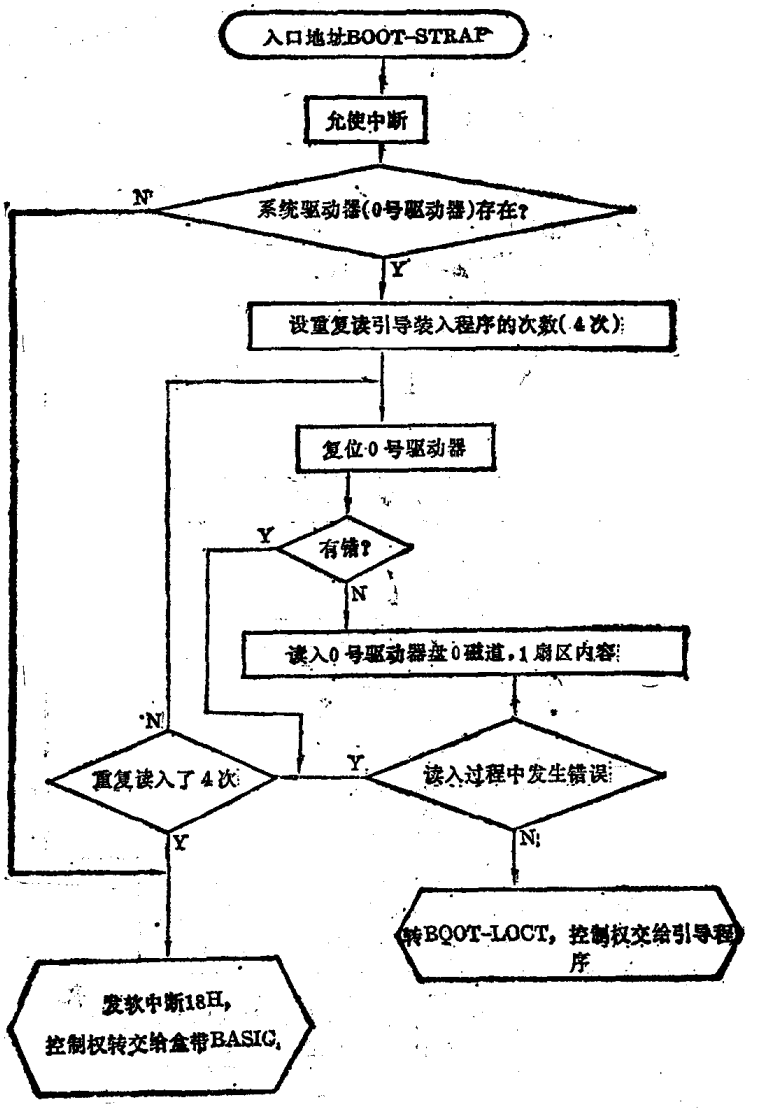

入口地址标号TST.12
现处生产测试模式?
Y
N
复位键盘
有键盘中断?
N
Y
返回扫描码'AA'?
N
Y
清、允便键盘
延迟后读扫描码
扫描码为0?
Y
N
在CRT上显示扫描码
显示键盘错误类别号'301'
还要不中断向量表
转TEST.13

TEST.13 盒带机测试



TEST.14 软盘机连接线测试





整个开机测试工作就此结束

```
LOC  OBT  LINE  SOURCE
          1  STITLE(ROM BIOS FOR IBM PERSONAL COMPUTER)
          2  ;.....
          2  ;EQUATES  常数定义段
          4  ;.....
0060      5  PORT_A          EQU  60H  ; 8255 PORT A ADDR  8255 PA 口地址
0061      6  PORT_B          EQU  61H  ; 8255 PORT B ADDR  8255 PB 口地址
0062      7  PORT_C          EQU  62H  ; 8255 PORT C ADDR  8255 PC 口地址
0063      8  CMD_PORT EQU 63H
0020      9  INTA00          EQU  20H  ; 8259 PORT  8259A 口地址
0021     10  INTA01          EQU  21H  ; 8259 PORT  8259A 口地址
0020     11  EOI             EQU  20H  ; 结束中断命令
0040     12  TIMER           EQU  40H  ; 8253 计时器基址
0043     13  TIM_CTL EQU     43H       ; 8253 TIMER CONTROL PORT ADDR
                                        计时器控制口地址
0040     14  TIMER0   EQU     40H       ; 8253 TIMER/CNTER 0 PORT ADDR
                                        计时器口地址
0001     15  TMINT    EQU     01        ; TIMER 0 INTR RECVD MASK  计时
                                        器 0 中断接收屏蔽数
0008     16  DMA08    EQU     08        ; DMA STATUS REG PORT ADDR
                                        8237DMA 片状态口地址
0000     17  DMA      EQU     00        ; DMA CHANNEL 0 ADDRESS REG
                                        PORT ADDR  8237DMA 片 0 通道地
                                        址寄存器口地址
0540     18  MAX_PERIOD EQU 540H  }
0410     19  MIN_PERIOD EQU 410H  }测试写磁带方波的半个周期用常数
0060     20  KBD_IN EQU      60H     ; KEYBOARD DATA IN ADDR PORT  键盘
                                        输入数据口地址
0002     21  KBDINT  EQU     02      ; KEYBOARD INTR MASK  键盘中断屏蔽值
0060     22  KB_DATA          EQU  60H  ; KEYBOARD SCAN CODE PORT  键盘
                                        扫描码输入口地址
0061     23  KB_CTL           EQU  61H  ; CONTROL BITS FOR KEYBOARD SENSE
                                        DATA  键盘控制口地址
          24  ; .....
          25  ; 8088 INTERRUPT LOCATIONS  下面是 8088 中断向量表
          26  ; .....
0000     27  ABS0  SEGMENT AT 0
0000     28  STG_LOC0     LABEL BYTE    ; 除数为 0 及单步操作向量
0008     29          ORG     2*4
```

```
0008          30  NMI__PTR        LABEL   WORD    ; 非屏蔽中断向量
0014          31                  ORG     5*4
0014          32  INT5__PTR       LABEL   WORD    ; 打印屏蔽中断
0020          33                  ORG     8*4
0020          34  INT__ADDR       LABEL   WORD    ; 8259A 中断向量表，占
                                                    8×4=32个字节
0020          35  INT__PTQ        LABEL   DWORD
0040          36                  ORG     10H*4
0040          37  VIDEO__INT      LABEL   WORD    ; CRT 中断向量
0074          38                  ORG     1DH*4
0074          39  PARM__PTR       LABEL   DWORD   ; POINTER TO VIDEO P-
                                                    ARMS 指向 CRT 初始
                                                    参数表
0078          40                  ORG     01EH*4  ; INTERRUPT 1EH 指向磁
                                                    盘参数表
0078          41  DISK__POINTER LABEL    DWORD
007C          42                  ORG     01FH*4  ; LOCATION OF POINTER
007C          43  EXT__PTR LABEL DWORD             ; POINTER TO EXTENSI
                                                    ON 第2字形发生表首
                                                    地址
7C00          44                  ORG     7COOH
7C00          45  BOOT__LOCN     LABEL   FAR     ; 引导程序地址
              46  ABS0 ENDS
              47
              48  ;.....................................................
              49  ; STACK…USED DURING INITIALIZATION ONLY  初始化系统
                                                          时间栈区
              50  ;.....................................
              51  STACK SEGMENT AT 30H
0000(120???)  52          DW        128 OUF(?)
0100          53  TOS     LABEL     WORD
              54  STACK ENDS
              55
              56  ;.....................................
              57  ; ROM BIOS DATA AREAS  下面是BIOS程序用到的数据存放段
                                          定义
              58  ;.....................................
0040          59  DATA SEGMENT AT 40H
0000(4????)   60  RS232__BASE      DW      4DUP(?) ; ADDRESSES OF RS232
```

```
                                                    ADAPTERS RS232通讯
                                                    口地址,最多可有4个
0008(4????)  61  PRINTER_BASE   DW   4DUP(?)   ; ADDRESSES OF PRINT-
                                                    ERS 打印机地址
0010????     62  EQUIP_FLAG    DW   ?          ; INSTALLED HARDWA-
                                                    RE 外设配接情况记录
                                                    单元
0012??       63  MFG_TST       DB   ?          ; INITIALIZATION FLAG
                                                    初始标记
0013????     64  MEMORY_SIZE   DW   ?          ; MEMORY SIZE IN K B-
                                                    YTES 系统总存贮器容
                                                    量记录单元
0015????     65  IO_RAM_SIZE   DW   ?          ; MEMORY IN I/O CHA-
                                                    NNEL 扩充 RAM 容量
                                                    记录单元
             66  ;.....
             67  ; KEYBOARD DATA AREAS 键盘数据区
             68  ;.....
0017??       69  KB_FLAG  DB   ?    ; 键盘标志
             70
             71  ; .....SHIFT FLAG EQUATES WITHIN KB_FLAG 下面是
                                                    键盘标志各个位的定义
             72
0080         73  INS_STATE      EQU  80H  ; INSERT STATE IS ACTIVE
                                                    插入态有效的话最高位为1 (7)
0040         74  CAPS_STATE     EQU  40H  ; CAPS LOCK STATE HAS BE-
                                                    EN TOGGLED CAPS键锁存状
                                                    态复位记录位 (6)
0020         75  NUM_STATE      EQU  20H  ; NUM LOCK STATE HAS BEEN
                                                    TOGGLED NUM键锁存状态复
                                                    位记录位 (5)
0010         76  SCROLL_STATE   EQU  10H  ; SCROLL LOCK STATE HAS
                                                    BEEN TOGGLED SCROLL 键
                                                    锁存状态复位记录位 (4)
0008         77  ALT_SHIFT      EQU  08H  ; ALTERNATE SHIFT KEY DE-
                                                    PRESSED 按下 ALT 键记录位
                                                    (3)
0004         78  CTL_SHIFT  EQU  04H  ; CONTROL SHIFT KEY DEPRESSED
                                                    按下 CTL 键记录位
```

```
                                                    (2)
0002        79  LEFT_SHIFT          EQU   02H   ; LEFT SHIFT KEY
                                                  DEPRESSED 按下
                                                  LEFT 键记录位(1)
0001        80  RIGHT_SHIFT         EQU   01H   ; RIGHT SHIFT KE-
                                                  Y DEPRESSED 按
                                                  下RIGHT 键记录位
                                                  (0)
            81
0018??      82  KB_FLAG_1           DB    ?     ; SECOND BYTE OF
                                                  KEYBOARD STAT
                                                  US 第二键盘状态
                                                  记录字节
            83
0080        84  INS_SHIFT           EQU   80H   ; INSERT KEY IS
                                                  DEPRESSED 按下
                                                  INS 键记录位(7)
0040        85  CAPS_SHIFT          EQU   40H   ; CAPS LOCK KEY
                                                  IS DEPRESSED
                                                  按下CAPS LOCK键
                                                  记录位(6)
0020        86  NUM_SHIFT           EQU   20H   ; NUM LOCK KEY
                                                  IS DEPRESSED
                                                  按下NUM LOCK键
                                                  记录位(5)
0010        87  SCROLL_SHIFT        EQU   10H   ; SCROLL LOCK KE
                                                  Y IS DEPRESSED
                                                  按下 SCROLL 键记
                                                  录位(4)
0008        88  HOLD_STATE          EQU   08H   ; SUSPEND KEY
                                                  HAS BEEN TOGGL
                                                  ED Hold 键状态复
                                                  位位(3)
            89
0019??      90  ALT_INPUT           DB    ?     ; STORAGE FOR AL
                                                  TERNATE KEYP-
                                                  AD ENTRY ALT
                                                  状态记录单元
001A????    91  BUFFER_HEAD         DW    ?     ; POINTER TO HE
```

```
                                                    AD OF KEY BOA-
                                                    RD BUFFER 键盘
                                                    缓存区首指针地址
001C????    92  BUFFER_TAIL     DW      ?       ; POINTER TO TA-
                                                    IL OF KEYBOARD
                                                    BUFFER 键盘缓存
                                                    区尾指针地址
001E(16 ????)  93  KB_BUFFER    DW      16DUP(?); ROOM FOR 15 EN
                                                    TRIES 缓存区
003E        94  KB_BUFFER_END LABEL   WORD
            95
            96  ; ..... HEAD=TAIL INDICATES THAT THE BUFFER IS
                  EMPTY
            97
0045        98  NUM_KEY         EQU     69      ; SCAN CODE FOR
                                                    NUMBER LOCK
                                                    NUM 键扫描码
0046        99  SCROLL_KEY      EQU     70      ; SCROLL LOCK KEY
                                                    SCROLL 键扫描码
0038       100  ALT_KEY         EQU     56      ; ALTERNATE SHI
                                                    FT KEY SCAN CO-
                                                    DE ALT 键扫描码
0010       101  CTL_KEY         EQU     29      ; SCAN CODE FOR
                                                    CONTROL KEY
                                                    CTL 键扫描码
003A       102  CAPS_KEY        EQU     58      ; SCAN CODE FOR
                                                    SHIFT LOCK
                                                    CAPS 键扫描码
002A       103  LEFT_KEY        EQU     42      ; SCAN CODE FOR
                                                    LEFT SHIFT
                                                    LEFT 键扫描码
0036       104  RIGHT_KEY       EQU     54      ; SCAN CODE FOR
                                                    RIGHT SHIFT
                                                    RIGHT 键扫描码
0052       105  INS_KEY         EQU     82      ; SCAN CODE FOR
                                                    INSERT KEY
                                                    INS 键扫描码
0053       106  DEL_KEY         EQU     83      ; SCAN CODE FOR
                                                    DELETE KEY
```

```
                107
                108 ;.....
                109 ; DISKETTE DATA AREAS  软盘机数据区
                110 ;.....
003E??          111 SEEK_STATUS     DB   ?      ; DRIVE RECALIBRATION ST
                                                   ATUS 驱动器重校状态 记
                                                   录字节
                112 ; BIT3-0 = DRIVE 3-0 NEEDS RECAL BEFORE 第3-0 位代表 3-0
                                                   号驱动器,某位为1
                113 ; NEXT SEEK IF BIT IS = 0  表示该位代表的驱动器在SEEK应重校
0080            114 INT_FLAG        EQU  080H ; INTERRUPT OCCURRENCE
                                                   FLAG
003F??          115 MOTOR_STATUS    DB   ?      ; MOTOR STATUS 马达状态字
                                                   节
                116 ; BIT3-0 = DRIVE 3-0 IS CURRENTLY RUNNING 第3-0 位中某位
                                                   为1表示某号驱动器马达在转
                117 ; BIT7 = CURRENT OPERATION IS A WRITE REQUIRES DELAY
                                                   第7位=1,写操作
0040??          118 MOTOR_COUNT     DB   ?      ; TIME OUT COUNTER FOR
                                                   DRIVE TURN OFF  超时后关
                                                   闭马达的超时计数字节
0025            119 MOTOR_WAXT      EQU  37     ; TWO SECONDS OF COUNTS
                                                   FOR MOTOR TURN OFF 常数,
                                                   让马达在超时后关闭的计算初
                                                   值
                120
                121 ;
0041??          122 DISKETTE_STATUS  DB?        ; SINGLE BYTE OF RETURN
                                                   CODE INFO FOR STATUS
                                                   下面是软盘机状态字节各位的
                                                   含义
0080            123 TIME_OUT        EQU  80H    ; ATTACHMENY FAILED TO
                                                   RESPOND  超时错(7)
0040            124 BAD_SEEK        EQU  40H    ; SEEK OPERATION FAILED
                                                   SEEK  操作不成功(6)
0020            125 BAD_NEC         EQU  20H    ; NEC CONTROLLER HAS FA
                                                   ILED  NEC 控制器坏(5)
0010            126 BAD_CRC         EQU  10H    ; BAD CRC ON DISKETTE R
```

```
                                                    EAD  CRC 校验错(4)
0009        127  DMA_BOUNDARY    EQU  09H   ; ATTEMPT TO DMA ACROSS
                                               64K BOUNDARY 传送字节数
                                               多于 64KB
0008        128  BAD_DMA         EQU  08H   ; DMA OVERRUN ON OPERA
                                               TION  DMA 超时错 (3)
0004        129  RECORD_NOT_FND  EQU  04H   ; REQUESTED SECTOR NOT
                                               FOUND  未找到扇区错(2)
0003        130  WRITE_PROTECT   EQU  03H   ; WRITE ATTEMPTED ON W
                                               RITE PROT DISK 欲写带写保
                                               护标记软盘错
0002        131  BAD_ADDR_MARK   EQU  02H   ; ADDRESS MARK NOT FOUND
                                               地址标志未发现(1)
0001        132  BAD_CMD         EQU  01H   ; BAD COMMAND PASSED TO
                                               DISKETTE I/O  软盘机命令
                                               错(0)
            133
0042(7??)   134  NEC_STATUS      DB   7DUP(?); STATUS BYTES FROM NEC
                                               7 个字节存贮一条软盘命令执
                                               行完后的状态
            135
            136  ;.....
            137  ; VIDEO DISPLAY DATA AREA  显示器数据段
            138  ;.....
0049??      139  CRT_MODE        DB   ?     ; CURRENT CRT MODE  当前
                                               CRT 模式
004A????    140  CRT_COLS        DW   ?     ; NUMBER OF COLUMNS ON
                                               SCREEN  列数
004C????    141  CRT_LEN         DW   ?     ; LENGTH OF REGEN IN BY
                                               TES  页显示缓存容量
004E????    142  CRT_START       DW   ?     ; STARTING ADDRESS IN R
                                               EGEN BUFFER 显示缓存始址
0050(8????) 143  CURSOR_POSN     DW   8DUP(?) ; CURSOR FOR EACH OF
                                               UP TO 8 PAGES  8 页图
                                               象的光标的记录单元
0060????    144  CURSOR_MODE     DW   ?     ; CURRENT CURSOR MODE S
                                               ETTING  当前光标的模式
0062??      145  ACTIVE_PAGE     DB   ?     ; CURRENT PAGE BEING DI
                                               SPLAYED  活动显示页页号
```

```
0063????   146  ADDR_6845          DW   ?   ; BASE ADDRESS FOR ACTIVE
                                               DISPLAY CARD 6845 基址
0065??     147  CRT_MODE_SET      DB   ?   ; CURRENT SETTING OF THE
                                               3×8 REGISTER 6845 模式寄存
                                               器内容记录单元
0066??     148  CRT_PALLETTE      DB   ?   ; CURRENT PALLETTE SETTI
                                               NG COLOR CARD 彩色CRT用
                                               色板号记录单元
           149
           150  ;.....
           151  ; CASSETTE DATA AREA 盒带机数据区
           152  ;.....
0067????   153  EDGE_CNT           DW   ?   ; TIME COUNT AT DATA EDGE
                                               数据位半个周期计时值暂存单元
0069????   154  CRC_REG            DW   ?   ; CRC REGISTER CRC字节记录
                                               单元
LOC OBJ    LINE SOURCE
006B??     155  LAST_VAL           OB   ?   ; LAST INPUT VALUE 前刻录
                                               音机输出电平记录字节
           156
           157  ;.....
           158  ; TIMER DATA AREA 计时器数据区
           159  ;.....
006C????   160  TIMER_LOW          DW   ?   ; LOW WORD OF TIMER COUNT
                                               时钟中断次数记录字 (低位部
                                               分)
006E????   161  TIMER_HIGH         DW   ?   ; HIGH WORD OF TIMER COU
                                               NT 时钟中断次数记录字 (高位
                                               部分)
0070??     162  TIMER_OFL          DB   ?   ; TIMER HAS ROLLED OVER
                                               SINCE LAST READ 日溢出标
                                               记字节
           163  ; COUNTS_SEC   EQU  18
           164  ; COUNTS_MIN   EQU  1092
           165  ; COUNTS_HOUR  EQU  65543
           166  ; COUNTS_DAY   EQU  1573040=1800B0H
           167
           168  ; .....
           169  ; SYSTEM DATA AREA 系统数据区
```

```
                170 ;.....
0071??          171 BIOS_BREAK   DB  ?  ; BIT 7 = 1 IF BREAK KEY HAS BEEN
                                         DEPRESSED  第7位=1表示 BREAK
                                         键被按下过
0072????        172 RESET_FLAG   DW  ?  ; WORD = 1234H IF KEYBOARD RES
                                         ET UNDERWAY  RESET-FLAG = 12
                                         34H 表示正进行键盘复位操作
                173 DATA ENDS
                174
                175 ;.....
                176 ; EXTRA DATA AREA  附加数据区
                177 ;.....
0050            178 XXDATA SEGMENT AT 50H
0000??          179 STATUS_BYTE     DB  ?  软中断5用状态记录字节
                180 XXDATA ENDS
                181
                182 ;.....
                183 ; VIDEO DISPLAY BUFFER  显示区缓存区
                184 ;.....
8800            185 VIDEO_RAM       SEGMENT AT 0B800H
0000            186 REGEN LABEL     BYTE
0000            187 REGENW LABEL    WORD
0000(16384??)   188         DB      16384DUP(?)
                189 VIDEO_RAM       ENDS
                190 ;.....
                191 ; RCM RESIDENT CODE
                192 ;.....
F000            193 CODE SEGMENT AT 0F000H
0000(57344??)   194         DB      57344 DUP(?) ; FILL LOWESY 56K
                195
E000 3537303    196         DB      5700051COPR. IBM1981 ; 常数、变量、指针定
     03035320                                              义完毕,下面是
     434F5052                                              BIOS 程序段
     2E204942
     4D203135
     3831
                197
                198 ;.....
                199 ; INITIAL RELIABILITY TESTS--PHASE  1 初始可靠性测试-1
```

```
                200 ;.....
                201       ASSUME CS:CODE, SS:CODE, ES:ABS0, DS:DATA   置段
                                                                  基寄存器
                202 ;.....
                203 ;   DATA DEFINITIONS
                204 ;.....
E016 D8E0 R     205 C1  DW  C11  ; RETURN ADDRESS
E018 EDE1 R     206 C2  DW  C24  ; RETURN ADDRESS FOR DUMMY STACK
                207 ;.....
                208 ;   THIS SUBROUTINE PERFORMS A READ/WRITE STORA
                    GE TEST ON A 16K BLOCK   本程序完成一16KB 存贮区的
                                             读/写测试
                209 ;   OF STORGAGE
                210 ; ENTRY REQUIREMENTS  入口参数:
                211 ;   ES = ADDRESS OF STORAGE SEGMENT BEING TESTED
                    ES: 被测试存贮区地址
                212 ;   DS = ADDRESS OF STORAGE SEGMENT BEING TESTED
                    DS: 被测试存贮区地址
                213 ;   WHEN ENTERING AT STGTST_CNT, CX MUST BE LOA
                    DED WITH THE BYTE COUNT  从 STGTST-CNT 进入时,
                    CX 应预置被测存贮区的字节数
                214 ; EXIT PARAMETERS  出口参数:
                215 ;   ZERO FLAG = 0 IF STORAGE ERROR(DATA COMPARE OR
                    PARITY CHECK. AL = 0  零标志 = 0: 存贮区错(数据比较错
                                          或校验错)
                216 ;                 DENOTES A PARITY CHECK.ELSE AL = X
                                      OR'ED BIT PATTERN OF THE  AL = 0 奇偶
                                      校验无错, 否则 (AL) = 应有数据格式⊕实际读
                                      出的数据格式
                217 ;                 EXPECTED DATA PATTERN VS THE AC
                                      TUAL DATA READ  AX, BX, CX, DX, DI,
                                      SI 均被破坏
                218 ;   AX, BX, CX, DX, DI, AND SI ARE ALL DESTROYED
                219 ;.....
E01A            220 STGTST   PROC NEAR
E01A B90040     221  MOV     CX, 4000H  ; SETUP CNT TO TEST A 16K BLK
                                          置字节数计数器,4000H = 16K(10)
E01D            222 STGTST_CNT:
E01D FC         223  CLD                ; SET DIR FLAG TO INCREMENT  前
```

```
                                                向测试标志 DF = 0
E01E 8BD9     224       MOV     BX, CX          ; SAVE BYTE CNT (4K FOR VID
                                                  EO OR 16K)  保存字节数,CRT 存
                                                  贮区测试时为 4K
E020 BBFFFF   225       MOV     AX, 0FFFFH      ; GET DATA PATTERN TO WRITE
                                                  准备写入数据格式
E023 BA55AA   226       MOV     DX, 0AA5511     ; SETUP OTHER DATA PATTERNS
                                                  TO USE  同上
E026 2BFF     227       SUB     DI, DI          ; DI = OFFSET 0 RELATIVE TO ES
                                                  REG  ES 段的位移初始为零
E028 F3       228       REP     STOSB           ; WRITE STORAGE LOCATIONS
                                                  重复写数据格式(0FFH)
E029 AA
E02A          229  C3:                          ; STG01
E02A 4F       230       DEC     DI              ; POINT TO LAST BYTE JUST WRI
                                                  TTEN  指向最后写入字节
E02B F0       231       STD                     ; SET DIR FLAG TO GO BACKW
                                                  ARDS  DF = 1, 逆向测试标志
E02C 8BF7     232  C4:  MOV     SI, DI          预置反向读-校-写的位移量
E02E 8BCB     233       MOV     CX, BX          ; SETUP BYTE CNT  预置字节数
E030 AC       234  C5:  LODSB                   ; READ CHAR FROM STORAGE
                                                  读一字节,送 AL, 修改 SI
E031 32C4     235       XOR     AL, AH          ; DATA READ AS EXPECTED?
                                                  AH 为应是的数据格式, 比较看相同
                                                ; 否
E033 7525     236       JNE     C7              ; NO-GO TO ERROR ROUTINE
                                                  不等, 去错误程序
E035 E462     237       IN      AL, PORT_C      ; DIDA PARITY ERROR OCCUR?
                                                  看 PORT-C 第 7.8 存贮区校验位是
                                                  否为 0
E037 24C0     238       AND     AL, 0C0H
E039 B000     239       MOV     AL, 0           ; AL = 0 DATA COMPARE OK
                                                  AL = 0, 无错
E03B 751D     240       JNZ     C7              ; 有错
E03D 80FC00   241       CMP     AH, 0           ; READING ZERO PATTERN?  是
                                                  最后的 00 数据格式
E040 7403     242       JE      C6              ; CONTINUE READING TILL END
                                                  是, 去 C6, 只读一检查, 不写
E042 8AC2     243       MOV     AL, DL          ; GET NEXT DATA PATTERN TO
```

```
                                        WRITE  不是,装入新数据格式
E044 AA     244     STOSB               ; WRITE IN BYTE LOC WE  JUST
                                        READ  写入存贮区,修改 DI
E045        245  C6:                    ; WRITE__NO__MORE
E045 E2E9   246     LOOP    C5          ; CONTINUE  TILL 16K/4K BLOCK
                                        TESTED  连续读-查-写, 直至16
                                        K(4K)查完
E047 80FC00 247     CMP     AH, 0       ; ZERO PATTERN WRITTEN TO
                                        SYG刚写入的是 0 数据格式
E04A 740E   248     JE      C7          ; YES-RETURN TO CALLER  是.
                                        检查完毕.
E04C 8AE0   249     MOV     AH, AL      ; SETUP TO NEW VALUE TO CO
                                        MPARE  否保存上次写入的数据格
                                        式,便于下次读出比较
E04E 86F2   250     XCHG    DH, DL      ; MOVE ZERO DATA PATTERN
                                        TO DL  新数据格式送 DL
E050 FC     251     CLD                 ; SET DIR FLAG TO GO FORWA
                                        RD  置前向测试标志 DF = 0
E051 47     252     INC     DI          ; SET POINTER TO BEG LOCATION
                                        DI 指向块首址
E052 74DB   253     JZ      C4          ; READ/WRIYE FORWARD IN STG
E054 4F     254     DEC     DI          DI 是在块首否?不是,恢复 DI, 使其
                                        指向块最后字节 + 1 位置
E055 BA0100 255     MOV     DX, 1       ; SETUP 01 AND 00 PATTERNS 送
                                        0001 数字格式
E058 EBD0   256     JMP     SHORT C3    ; READ/WRITE  BACKWARD  IN
                                        STG  逆向测试
E05A        257  C7:
E05A C3     258     RET  返回调用段
            259  STGTST ENDP
            260  ;.....
            261  ; TEST.01  测试 01 8088 处理器测试
            262  ;       8088 PROCESSOR TEST
            263  ; DESCRIPTION
            264  ;       VERIFY 8088 FLAGS, REGISYERS AND CONDITIONAL
                         JUMPS  检查 8088 标志位,条件转移是否能正常工作
            265  ;.....
E05B        266  RESET LABEL NEAR
E05B FA     267  START: CLI  ; DISABLE INTERRUPTS  禁止中断
```

```
E05C B4D5   268     MOV     AH, 0D5H    ; SET SF, CF, ZF, AND AF FLAGS
                                          ON PF SF, CF, ZF, AF 标志位置 1
E05E 9E     269     SAHF
E05F 734E   270     JNC     ERR01       ; GO TO ERR ROUTINE IF CF NOT
                                          SET  CF≒1, 错
E061 754C   271     JNZ     ERR01       ; GO TO ERR ROUTINE IF ZF NOT
                                          SET  ZF≒1, 错
E063 7B4A   272     JNP     EPP01       ; GO TO ERR ROUTINE IF PF NOT
                                          SET  PF≒1, 错
E065 7948   273     JNS     ERR01       ; GO TO ERR ROUTINE IF SF NOT
                                          SET  SF≒1, 错
E067 9F     274     LAHF                ; LOAD FLAG IMAGE TO AH  标志
                                          装入 AH
E068 B105   275     MOV     CL, 5       ; LOAD CNT REG WITH SHIFT CNT
                                          装入计数值 5
E06A D2EC   276     SHR     AH, CL      ; SHIFT AF INTO DARRY BIT POS
                                          右移 AH  5 位,使 AF 进入 CF 位
E06C 7341   277     JNC     ERR01       ; GO TO ERR ROUTINE IF AF NOT
                                          SET  AF≒1, 错
E06E B040   278     MOV     AL, 40H     ; SET THE OF FLAG ON
E070 D0E0   279     SHL     AL, 1       ; SETUP FOR TESTING  置 AL 溢出
E072 713B   280     JNO     ERR01       ; GO TO ERR ROUTINE IF OF NOT
                                          SET  OF≒1, 错
E074 32E4   281     XOR     AH, AH      ; SET AH = 0  AH 清 0
E076 9E     282     SAHF                ; CLEAR SF, CF, ZF, AND PF  清
                                          SF, CF, ZF, PF, AF
E077 7236   283     JC      ERR01       ; GO TO ERR ROUTINE IF CF ON
                                          CF = 1, 错
E079 7434   284     JZ      ERR01       ; GO TO ERR ROUTINE IF ZF ON
                                          ZF = 1, 错
E07B 7832   285     JS      ERR01       ; GO TO ERR ROUTINE IF SF ON
                                          SF = 1, 错
E07D 7A30   286     JP      ERR01       ; GO TO ERR ROUTINE IF PF ON
                                          PF = 1, 错
E07F 9F     287     LAHF                ; LOAD FLAG IMAGE TO AH  标志
                                          装入 AH
E080 B105   288     MOV     CL, 5       ; LOAD CNT REG WITH SHIFT CNT
                                          右移计算器
E082 D2EC   289     SHR     AH, CL      ; SHIFT AF' INTO CARRY BIT POS
```

```
                                              AF 移入 CF 位
E084 7229      290      JC      ERR01        ; GO TO ERR ROUTINE IF ON
                                              CF=1 出错
E086 D0E4      291      SHL     AH,1         ; CHECK THAT OF'IS CLEAR 置AH
                                              非溢出态
E088 7025      292      JO      ERR01        ; GO TO ERR ROUTINE IF ON
                                              OF=1 出错
               294  ; READ/WRITE THE 8088 GENERAL AND SEGMENTATION
                    REGISTERS  下面将 8088 全部通用及段寄存器写全 0，全 1，
               295  ; WITH ALL ONE'S AND ZEROES'S  然后读出检查
               296
E08A B8FFFF    297      MOV     AX,0FFFFH ; SETUP ONE'S PATTERN IN AX
                                              写全 1 至全部寄存器
E08D F9        298      STC
E0BE BE08      299  C8: MOV     DS, AX       ; WRITE PATTERN TO ALL REGS
E090 BCDB      300      MOV     BX, DS
E092 8EC3      301      MOV     ES, BX
E094 8CC1      302      MOV     CX, ES
E096 8ED1      303      MOV     SS, CX
E098 8CD2      304      MOV     DX, SS
E09A 88E2      305      MOV     SP, DX
E09C 8BEC      306      MOV     BP, SP
LOC OBJ        LINE  SOURCE
E09E 8BF5      307      MOV     SI, BP
E0A0 88FE      308      MOV     DI, SI
EDA2 7307      309      JNC     C9           ; TST1A  全 0 是否写入过?
E0A4 33C7      310      XOR     AX, DI       ; PATTERN MAKE IT THRU ALL
                                              REGS  寄存器有错?AX 清 O
E0A6 7507      311      JNZ     ERR01        ; NO-GO TO ERR ROUTINE  不等转
                                              ERR01
E6A8 F8        312      CLC                                清 CF，进行全 0 测试
E0A9 73E3      313      JNC     C8
E0AB           314  C9:                      ; TST1A
EOAB OBC7      315      OR      AX, DI       ; ZERO PATTERN MAKE IT THRU?
                                              寄存器出错?
E0AD 7401      316      JZ      C10          ; YES-GO TO NEXT TEST  无错，全
                                              0，转 ROS-CHECKSUM
OAF F4         317  ERR01: HLT               ; HALT SYSTEM  暂停
               318  ;.....
```

```
                319 ; TEST.02     测试02  ROS检查和测试1
                320 ;         ROS CHECKSUM TEST I   本段对记录POD和BIOS的
                                                    8K ROS模块进行检查和测试
                321 ; DESCRIPTION
                322 ;         A CHECKSUM IS DONE FOR THE 8K ROS MODULE
                              CONTAINING POD AND BIOS.
                323 ;.....................................................
E0B0            324 C10:
E0B0 E000       325         MOV   AL, 0          ; DISABLE NMI INTERRUPTS 禁
                                                   止不可屏蔽中断
E0B2 B6A0       326         OUT   0A0H, A1.
E0B4 E683       327         OUT   83H, AL        ; NITIALZE OMA PAGE REG 初
                                                   始DMA页寄存器
E0B6 B099       328         MOV   AL, 99H        ; SET 8255 A, C-INPUT, B-OUT
                                                   PUT
E0B8 E663       329         OUT   CMD__PORT, AL  ; WRITE 8255 CMD/MODE R
                                                   EG 写8255命令口，使PAin,
                                                   PBout, PCin.
E0BA BOFC       330         MOV   AL, 0FCH       ; OISABLE PARITY CHECKERS
                                                   AND
E0BC E661       331         OUT   PORT__B, AL;   GATE SNS SWS; CASS MOTOR
                                                   OFF 允使读取RAM大小，关断盒
                                                   带机马达(参见第一章的8255A-I/O
                                                   映射图)
E0BE 2ACO       332         SUB   AL, AL        ;
E0C0 BAD803     333         MOV   DX, 3D8H
E0C3 EE         334         OUT   DX, AL         ; DISABLE COLOR VIDEO 禁止彩
                                                   色CRT，见第二十章4模式选择寄
                                                   存器
E0C4 FECJ       335         INC   AL
E0C6 BAB803     336         MOV   DX, 3B8H
E0C9 EE         337         OUT   DX, AL         ; DISABLE B/W VIDEO, EN HIGH
                                                   RES 置单色CRT高分辨率方式，
                                                   禁止视频输出
E0CA B800F0 R338            MOV   AX, CODE       ; SETUP SS SEG REG
E0CD 8ED0       339         MOV   SS, AX         设置段寄存器
E0CF BB00E0     340         MOV   DX, 0E00H      ; SETUP STARLING ROS ADDR
                                                   ROS起始地址送BX
E0D2 BC16E0 P   341         MOV   SP, OFFSET C1  ; SETUP RETURN ADDPESS
```

```
                                        装入返回地址,位移量 C11
E0D5 E93301   342          JMP    ROS_CHECKSUM  转 541 行
E0D8 75D5     343   C11:   JNE    ERR01         ; HALT SYSTEM IF ERROR ROS-
                                                   CHECKSUM 返回处
              344   ;.....
              345   ; TEST 03  测试 03
              346   ;      8237 DMA INITIALIZATION CHANNEL REGISTER TEST
                           8237 DMA 通道寄存器初始化测试
              347   ; DESCRPITION
              348   ;      DISABLE THE 8237 DMA CONTROLLER.VERIFY THAT
                           TIMER I FUNCTIONS OK.  完成:禁止 8237DMA 控制器,
                           计时器 1 测试
              349   ;      WRITE/READ THE CURRENT ADDRESS AND WORD
                           COUNT REGISTERS FOR ALL   读/写所有通道的当前地
                           址,字计数寄存器,初始
              350   ;      CHANNELS. INITIALIZE AND START DMA FOR MEM
                           ORY REFRESM.   开始存贮器刷新的 DMA 操作
              351   ,.....
              352   ;      DISABLE DMA CONTROLLER
              353
E0DA B004     354          MOV    AL, 04        ; DISABLE DMA  CONTROLLER
                                                   禁止 DMA 控制器
E0DC E608     355          OUT    DMA08, AL
              356
              357   ;      VERIFY THAT TIMER I FUNCTIONS OK  下面进行计时
                           器 1 的测试
              358
E0DE B054     359          MOV    AL, 54H       ; SEL TIMER 1, LSB, MODE 2  写
                                                   命令字,置计时器 1,只读/装入低字
                                                   节,模式 2 工作方式
E0E0 E643     360          OUT    TIMER + 3, AL
E0E2 2BC9     361          SUB    CX, CX        CX 清 0.
E0E4 8AD9     362          MOV    BL, CL        BL 清 0,下一轮测试数据
E0E6 8AC1     363          MOV    AL, CL        ; SET INITIAL TIMER CNT TO 0
                                                   置计时器 1,计时初值 0
E0E8 E641     364          OUT    TIMER + 1, AL
E0EA          365   C12:                       ; TIMERI_BITS_ON  封锁计时器
                                                   1,0 方式
E0EA B040     366          MOV    AL, 40H       ; LATCH TIMER I COUNT
```

```
E0EC E643   367          OUT    TIMER+3, AL
E0EE E441   368          IN     AL, TIMER+1   ; READ TIMER I COUNT 读计时器1
E0F0 0AD8   369          OR     BL, AL        ; ALL BITS ON IN TIMER
E0F2 80FBFF 370          CMP    BL, 0FFH      ; YES-SEE IF ALL BITS GO OFF 计时器值为0FFH?
E0F5 7404   371          JE     C13           ; TIMERI_BITS_OFF 是,计时器1正常
E0F7 E2F1   372          LOOP   C12           ; TIMERI_BITS_ON 继续读取计时值
E0F9 EBB4   373          JMP    SHORT ERR01   ; TIMER I FAILURE, HALT SYS 循环结束,计时值不准确,出错
E0FB        374  C13:                         ; TIMERI_BITS_OFF
E0FB 8AC3   375          MOV    AL, BL        ; SET TIMER I CNT OFFH 送计时器1
E0FD 2BC9   376          SUB    CX, CX        清循环计数器
E0FF E641   377          OUT    TIMER+1, AL
E101        378  C14:                         ; TIMER_LOOP
E101 B040   379          MOV    AL, 40H       ; LATCH TIMER I COUNT 封锁计时器1,0方式
E103 E643   380          OUT    TIMER+3, AL
E105 E441   381          IN     AL, TIMER+1   ; READ TIMER I COUNT 读计时器1
E107 22D8   382          AND    BL, AL        计时器值为00H?
E109 7404   383          JZ     C15           ; WRAP_DMA_REG
E10B E2F4   384          LOOP   C14           ; TIMER_LOOP 继续读取计时值

LOC 08J     LINE SOURCE
E10D EBA0   385          JMP    SHORT ERR01
            386
            387  ;  INITIALIZE TIMER 1 TO REFRESH MEMORY 为刷新存贮器初始计时器1
            388
E10F        389  C15:                         ; WRAP_DMA_REG 覆盖DMA寄存器
E10F B054   390          MOV    AL, 54H       ; SEL TIM 1, LSB, MODE, 2 置计时器1. 低字节, 读/装入模式2工作方式
```

```
E111 E643   391          OUT   TIMER+3, AL   ; WRITE TIMER MODE REG
E113 B012   392          MOV   AL, 18        ; SETUP DIVISOR FOR REFR
                                               ESH 设置刷新分频值
E115 E641   393          OUT   TIMER+1, AL   ; WRILE TIMER 1 CNT REG
                                               送计时器1
E117 E60D   394          OUT   DMA+ODH, AL   ; SEND MASTER CLEAR TO
                                               DMA
            395
            396 ;        WRAP DMA CHANNELS ADDRESS AND COUNT REG
                         ISTERS 覆盖 DMA 通道地址和计数寄存器 1～8。
            397
E119 B0FF   398          MON   AL, 0FFH      ; WRITE PATTERN FFH TO
                                               ALL REGS 测试数据格式
                                               OFFH
E11B 8AD8   399 C16:     MOV   BL, AL        ; SAVE PATTERN FOR COM
                                               PARE
E11D 8AF8   400          MOV   BH, AL
E11F B90800 401          MOV   CX, 8         ; SETUP LOOP CNT 计数器
                                               送8.
E122 BA0000 402          MOV   OX, DMA       ; SETUP I/O PORT ADDR OF
                                               REG DMA 4 个通道的第一寄
                                               存器地址送 DX
E125 EE     403 C17:     OUT   DX, AL        ; WRITE PATTERN TO REG,
                                               LSB 计数或地址寄存器送入测
                                               试数据
E126 EE     404          OUT   DX, AL        ; MSB OF 16 BIT REG
E127 B80101 405          MOV   AX,0101H      ; AX TO ANOTHER PAT BE
                                               FORE RD
E12A EC     406          IN    AL, DX        ; READ 16-BIT DMA CH RE
                                               G, LSB 读取计数或地址寄存
                                               器低8位
E12B 8AE0   407          MOV   AH, AL        ; SAVE LSB OF 16-BIT REG
                                               送 AH
E12D EC     408          IN    AL, DX        ; READ MSB OF DMA CH R
                                               EG 读取计数或地址寄存器高
                                               8位
E12E 3BD8   409          CMP   BX, AX        ; PATTERN READ AS WRIT
                                               TEN? 比较看读出数据与写入
                                               数据是否相同
```

```
E130 7403    410          JE    C18          ; YES-CHECK NEXT REG 相同
E132 E97AFF  411          JMP   ERR01        ; NO_HALT THE SYSTEM 不同
                                             暂停系统
E135         412  C18:                       ; NXT_DMA_CH
E135 42      413          INC   DX           ; SET I/O PORT TO NEXT CH
                                             REG DX 指向下一个计数(奇数)
                                             或地址(偶数)寄存器，继续测试
E136 E2ED    414          LOOP  C17          ; WRITE PATTERN TO NEXT
                                             REG
E138 F6D0    415          NOT   AL           ; SET PATTERN TO ZERO (A
                                             L) = 0
E13A 74DF    416          JZ    C16          ; WRITE TO CHANEL REGS 8
                                             个寄存器清 0
             417
             418  ;  INITIALIZE AND START DMA FOR MEMORY REFRESH
                     为寄存器刷新初始、启动 DMA
             419
E13C B0FF    420          MOV   AL, 0FFH     ; SET CNT OF 64K FOR RAM
                                             REFRESH 设置通道 0 基本字计
                                             数器为 64KB
E13E E601    421          OUT   DMA+1, AL
E140 E601    422          OUT   DMA+1, AL
E142 B058    423          MOV   AL, 058H     ; SET DMA MODE, CH 0, REA
                                             D, AUOTINT 写 DMA 模式寄
                                             存器：选 0 通道，读传送，自动初
                                             始，地址增，单模式
E144 E60B    424          OUT   DMA+0BH, AL  ; WRITE DMA MODE REG
E146 B000    425          MOV   AL, 0        ; ENABLE DMA CONTROLLER
E148 E608    426          OUT   DMA+8, AL    ; SETUP DMA COMMAND REG
                                             置 DMA 命令寄存器
E14A E60A    427          OUT   DMA+10, AL   ; ENABLE CHANNEL 0 FOR R
                                             EFRESH 允使通道 0.
E14C B041    428          MOV   AL, 41H      ; SET MODE FOR CHANNEL 1
                                             置通道 1, 2, 3 的模式，均为检正
                                             传送，
E14E E60B    429          OUT   DMA+0BH, AL
E150 B042    430          MOV   AL, 42H      ; SET MODE FOR CHANNEL 2
                                             禁止自动初始，地址增，单模式方
                                             式
```

```
E152 E60B       431         OUT   DMA+0BH, AL ;
E154 B043       432         MOV   AL, 43H          ; SET MODE FOR CHANNEL 3
E156 E60B       433         OUT   DMA+0BH, AL
                434 ;.....................................................
                435 ; TEST. 04  测试 04
                436 ;       BASE 16K READ/WRITE STOKAGE TEST  基本 16K
                            RAM 测试
                437 ;
                438 ; DESCRIPTION  完成:将数据格式 FF, 55, AA, 01, 00 写入第
                  一块 16K RAM, 然后读出,检查是否出错
                439 ,       WRITE/READ/VERIFY DATA PATTRNS FF, 55, AA,
                            01, AND 00 TO IST 16K OF
                440 ;       STORAGE. VERIFY STORAGE ADDRESSABILITY  为
                            检查生产测试模式 2 初始 8259A 中断控制器
                441 ;       INITIALIZE THE 8259 INTERRUPT CONTROLLER C
                            HIP FOR CHECKING
                442 ;       MANUFACTURING TEST 2 MODE.
                443 ;.....................................................
                444 ;       DETERMINE MEMORY SIZE AND FILL MEMORY
                            WITH DATA  决定存贮器大小并写入数据
                445
E158 B84000   R 446         MOV   AX, DATA         ; POINT DS TO DATA SEG
E15B 8ED8       447         MOV   DS, AX           ; DS 指向数据段
E15D 8BIE7200 R 448         MOV   BX, RESET_FLAG   ; SAVE RESET_FLAG IN
                                                     BX  键盘复位时 RESET_
                                                     FLAG = 1234H
E161 2BC0       449         SUB   AX, AX           ; SET ES AND DS TO 0
E163 8EC0       450         MOV   ES, AX           ; SETUP ES SEGMENT REG
                                                     设立 ES 段首寄存器
E165 8EDB       451         MOV   DS, AX
E167 2BFF       452         SUB   DI, DI
E169 E460       453         IN    AL, PORT_A       ; DETERMINE BASE RAM
                                                     SIZE 读 8255 PA 口
E16B 240C       454         AND   AL, 0CH          ; ISOLATE RAM SIZE SWS 从 2.3 位
                                                     取得系统板 RAM 大小
                                                     00—16K
                                                     01—32K
                                                     10—48K
                                                     11—64K
E16D 0404       455         ADD   AL, 4            ; CALCULATE MEMORY SIZE
                                                     第 3 位加 1
```

```
E16F B10C   456          MOV    CL, 12
E171 D3E0   457          SHL    AX, CL   AX 左移 12 位        ┌ 01—16K
                                                              │ 10—32K
E173 8BC8   458          MOV    CX, AX   CX 最高位,以高位     │ 11—48K
                                                              └ 00—64K
E175 8AE0   459          MOV    AH, AL
E177 FC     460          CLD                    ; SET DIR FLAG TO INCR   前
                                                  向写入标志
E178 AA     461  C19:    STOSB                  ; FILL BASE RAM WITH DATA
                                                  写入数据 00H
E179 E2FD   462          LOOP C19               ; LOOP TIL ALL ZERO
            463
            464  ;       DETERMINE IO CHANNEL RAM SIZE   决定 I/O 通道
                         RAM 大小
            465
E17B E462   466          IN     AL, PORT_C      读 PCD 取低 4 位,低 4 位值×32KB
                                                  =扩展 RAM 区大小
E17D 240F   467          AND    AL, 0FH
E17F 7418   468          JZ     C21               低 4 位=0 无扩展 RAM
E181 BA0010 469          MOV    DX, 1000H       ; SEGMENT FOR I/O RAM
                                                  1000H 是扩展 RAM 区的始址
E184 8AE0   470          MOV    AH, AL            保存外接 RAM 的大小(在低 4 位)
E186 B000   471          MOV    AL, 0             写入数据 00H
E188        472  C20:                           ; FILL_IO:
E188 8EC2   473          MOV    ES, DX
E18A B90080 474          MOV    CX, 8000H       ; FILL 32K BYTES   8000H=32
                                                  KB(10)计数值
E18D 2BFF   475          SUB    DI, DI
E18F F3     476          REP    STOSB                  写数
E190 AA
E191 81C20008 477        ADD    DX, 800H        ; NEXT SEGMENT VALUE   写
                                                  完 32KRAM，再加上 32KB 的地
                                                  址偏差,指向下一32K.
E195 FECC   478          DEC    AH                计数器减 1
E197 75EF   479          JNZ    C20             ; FILL_ID   全部扩展 RAM 写完
            480  ;.....
            481  ;       INITIALIZE THE 8259 INTERRUPT CONTROLLER CHIP
                         初始 8259A 中断控制器
                 482  ;.....
```

```
E199                    483  C21:
E199 B013               484       MOV  AL, 13H          ; ICW1-EDGE, SNGL,
                                                          ICW4 INTA00 = 20H,
                                                          设置沿触发输入,间隔
                                                          8,
E198 E620               485       OUT INTA00 AL         单片,需要 ICW4
E19D B008               486       MOV  AL, B            ; SETUP  ICW2—INT
                                                          TYPE 8(8-F) A15～
                                                          A1,  插在向量地址的
                                                          高五倍上,8259A
E19F E621               487       OUT  INTA01, AL       根据中断级设定低3倍. A10～
                                                          A5 任意. 中断号 8～F
E1A1 B009               488       MOV  AL, 9            ; SETUP ICW4 - BUFF
                                                          RD, 8086 MODE
E1A3 E621               489       OUT  INTA01, AL       设置 1CW4,  为 8086
                                                          带缓冲器方式
E1A5 2BC0               490       SUB  AX, AX           ; POINT DS AND ES
                                                          TO BEGIN
E1A7 8EC0               491       MOV  ES, AX           ; OF R/W STORAGE
                                                          ES 指向 RAM 区首
E1A9 BE4000       R     492       MOV  SI, DATA         ; POINT DS TO DATA
                                                          SEG
E1AC 8EDE               493       MOV  DS, SI           ; DS 指向数据段,便于
                                                          存 RESET-FLAG 单
                                                          元
E1AE 891E7200     R     494       MOV  RESET_FLAG, BX   ; RESTORE  REST_F
                                                          LAG
E1B2 813E72003412 R     495       CMP  RESET_FLAG, 1234H  ; RESET_FLAG S
                                                            ET? 键盘复位否?
E1B8 7438               496       JE   C25              ; YES-SKIP STG TE
                                                          ST  是. 跳过下一段
E1BA 8ED8               497       MOV  DS, AX           ; POINT DS TO 1ST
                                                          16K OF STG  否.
                                                          DS 指向 16K 的第一
                                                          单元
                        498  ;.....................................................
                        499  ;    CHECK FOR MANUFACTURING TEST 2 TO LOAD
                                  TEST PROGRAMS FROM KEYBOARD.
                        500  ;.....................................
```

```
E1BC BCFO3F    501          MOV   SP, 3FFOH      ; ESTABLISH TEMPORARY
                                                    STACK  设立临时堆栈
E1BF 8ED0      502          MOV   SS, AX
E1C1 8BF8      503          MOV   DI, AX         ; DI, SS 清。
E1C3 BB2400    504          MOV   BX, 24H        ; 0024H 是键盘中断向量单元
E1C6 C707B6E2 R 505         MOV   WORD PTR (BX), OFFSET D11 ; SET UP KB
                                                    INTERRUp
                                                    T 送键盘中
                                                    断处理子程
                                                    序地址 D11
                                                    至该向量单
                                                    元
E1CA 43        506          INC   BX
E1CB 43        507          INC   BX
E1CC 8C0F      508          MOV   (BX), CS       ; 送段基址
E1CE E8B704    509          CALL  KBD_RESET      ; READ IN KB RESET C
                                                    ODE TO BL
E1D1 80FB65    510          CMP   BL, 065H       ; IS THIS MANUFACTUR
                                                    ING TEST 2?
E1D4 750E      511          JNZ   C23            ; JUMP IF NOT MAN.TE-
                                                    ST 非生产测试模式转
                                                    C23
E1D6 B2FF      512          MOV   DL, 255        ; READ IN TEST PROGR
                                                    AM
E1D8 E8BA04    513  C22:    CALL  SP_TEST        ; 读进测试程序
E1DB 8AC3      514          MOV   AL, BL
E1DD AA        515          STOSB
E1DE FECA      516          DEC   DL
E1E0 75F6      517          JNZ   C22            ; JUMP IF NOT DONE
                                                    YET
E1E2 CD3E      518          INT   3EH            ; SET INTERRUPT TYPE
                                                    62 ADDRESS F8H 调进
                                                    测试程序后发软中断3EH.
E1E4           519  C23:                        ; CONTINUE IN NORMAL
                                                    MODE
E1E4 0E        520          PUSH  CS             ; PUT SS BACK  恢复栈
                                                    段基址
E1E5 17        521          POP   SS
E1E6 FA        522          CLI                  ; 关中
```

```
E1E7 BC18E0  R     523          MOV    SP,OFFSET C2     ; SETUP RETURN
                                                         ADDR ESS  送测试
                                                         04 返回地址至栈顶
E1EA E92DFE              524          JMP   STGTST       ; GO TO RD/WRT STG
                                                         SUBROUTINE
E1ED 7403                525  C24:    JE    C25          ; GO TO NEXT TEST
                                                         IF OK
E1EF E9BDFE              526          JMP   ERR01
                         527
                         528  ;       SETUP STACK SEG AND SP   设栈
                         529
E1F2                     530  C25:
E1F2 B83000              531          MOV   AX, STACK    ; GET STACK VALUE
E1F5 8ED0                532          MOV   SS, AX       ; SET THE STACK
                                                         UP  设栈基址
E1F7 BC0001              533          MOV   SP, OFFSET TOS ; STACK IS READY
                                                         TO GO  送栈指针
                         534
                         535  ;       SETUP THE NMI INTERRUPT VECTOR PO
                                      INTER
                         536
E1FA 26C7060800C3E2 R    537          MOV   ES:NMI__PTR, OFFSET NMI__INT; 送
                                            NMI 中断处理程序首址
E201 26C7060A0000F0 R    538          MOV   ES:NMI__PTR+2, CODE; 送代码基址段
E208 E92A00              539          JMP   TST6         ; GO TO NEXT
                                                         TEST  去测试 06
                         540
E20B                     541  ROS__CHECKSUM PROC NEAR ; NEXT__ROS__MO
                                                         DULE ROS-CHE
                                                         CKSUM 子程序,
                                                         检查 8K
E20B B90020              542          MOV   CX, 8192     ; NUMBER OF BYTES
                                                         TO ADD  8192=8K
                                                         CX 是计数器
E20E 32C0                543          XOR   AL, AL       ; AL 清 0。
0E210                    544  C26:
E210 2E0207              545          ADD   AL, CS:[BX]  ; 累加 8K
E213 43                  546          INC   BX           ; POINT TO NEXT B
                                                         YTE  BX 增 1
```

```
E214 E2FA          547          LOOP C26          ; ADD ALL BYTES IN ROS
                                                    MODULE 加完 8K 否?
E216 0AC0          548          OR   AL, AL       ; SUM = 0? 是,结果是否正
                                                    常标志送 ZF 位
E218 C3            549          RET
                   550 ROS_CHECKSUM ENDP
                   551 ;.....
                   552 ; INITIAL RELIABILITY TEST—PHASE2 初始可
                       靠性测试—2
                   553 ;.....
                   554          ASSUME CS:CODE, DS:ABS0
                   555
E219 50415249545920 556 D1       DB   'PARITY CHECK 2'
     434845434B2032
000E               557 D1L EQU   $-D1; 上面'PARITY CHECK2'的长度
E227 50415249545920 558 D2       DB   'PARITY CHECK 1'
     43484543482031
000E               559 D2L  EQU  $-D2 上面'PARITY CHECK1'的长度
                   560 ;.....
                   561 ; TEST_06        测试 06
                   562 ;    8259 INTERRUPT CONTROLLER TEST
                            8259A 中断控制器测试
                   563 ; DESCRIPTION 完成
                   564 ;    READ/WRITE THE INTERRUPT MASK REG
                            ISTER(IMR)WITH ALL ONES AND ZEROES
                            用全 1 全 0 写/读其屏蔽寄存器 IMR,
                   565 ;    ENABLE SYSTEM INTERRUPTS MASK
                            DEVICE INTERRUPTS OFF. CHEC K FOR 检
                            查热中断,允使系统中断,去除外设中断屏蔽
                   566 ;    HOT INTERRUPTS(UNEXPECTED).
                   567 ;.....
E235               568 TST6:
E235 2BC0          569          SUB  AX, AX       ; SET UP ES REG
E237 8EC0          570          MOV  ES, AX
                   571
                   572 ; .....SET UP THE INTERRUPT 5 POINTER TO A
                       DUMMY
                   573
E239 26C706140054FF R 574        MOV  ES:INTS_PTR, OFFSET PRINT_SCRE
```

```
                                        EN  ; PRINT SCREEN   送打印屏蔽中
                                            断处理子程序首址及段基址
E240 26C706160000F0 R   575         MOV  ES:INTS_PTR+2, CODE  ;
                        576
                        577 ;       TEST THE IMR REGISTER  测试 IMR 寄存器
                        578
E247 FA                 579         CLI                 ; DIABLE INTERRUPTS
                                                          关中
E248 B000               580         MOV  AL, 0          ; SET IMR TO ZERO
                                                          写/读全 0。
0E24A E621              581         OUT  INTA01, AL
E24C E421               582         IN   AL, INTA01     ; READ IMR
E24E 0AC0               583         OR   AL, AL         ; IMR = 0?
E250 752B               584         JNZ  D6             ; GO TO ERR ROUTINE
                                                          IF NOT 0  读出数据不为
                                                          0。出错
E252 B0FF               585         MOV  AL, 0FFH       ; DISABLE DEVICE INT
                                                          ERRUPTS  写/读全 1
E254 E621               586         OUT  INTA01, AL     ; WRITE TO IMR
E256 E421               587         IN   AL, INTA01     ; READ IMR
E258 0401               588         ADD  AL, 1          ; ALL IMR BIT ON?  读
                                                          出数据不为全 1，出错
E25A 7521               589         JNZ  D6             ; NO-GO TO ERR ROUTI
                                                          NE
                        590
                        591 ;       CHECK FOR HOT INTERRUPTS  检查热中断
                                                          (不可预料中断)
                        592
E25C FC                 593         CLD                 ; SET DIR FLAG TO GO
                                                          FORWARD  设前向标记
E25D B90800             594         MOV  CX, B          ; SETUP TEMP INT RTNE
                                                          IN PRT TBL
LE260 BF2000            595         MOV  DI, OFFSET INT_PTR  ; GET ADDR
                                                          ESS OF INT
                                                          PROC TAB
                                                          LE 中断向量
                                                          表首址送 DI
E263                    596 D3:                         ; VECTBLO:
E263 B8B6E2  R          597         MOV  AX, OFFSET D11 ; MOVE ADDR OF I
```

```
                                                    NTR PROC TO TBL
                                                    暂时中断处理程序首
                                                    址送 AX
E266 AB          598            STOSW                          ; 送 AX 至向量表
E267 B800F0 R    599            MOV  AX, CODE  ; GET ADDR OF INTR PROC SEG
                                                    送暂时中断处理程序段基址至向量表
E26A AB          600            STOSW
E26B 83C304      601            ADD  BX, 4     ; SET BX TO POINT TO NEXT VAL
E26E E2F3        602            LOOP D3        ; VECTBL0  连续送 8 次，对应于 8 级
                                                    中断
                 603
                 604  ;         INTERRUPTS ARE MASKED OFF. CHECK THAT NO
                                INTERRUPTS OCCUR  等待中断发生
                 605
E270 32E4        606            XOR  AH, AH    ; CLEAR AH REG
E272 FB          607            STI            ; ENABLE EXTERNAL INTERRUPTS
                                                    允许中断(外部)
E273 2BC9        608            SUB  CX, CX    ; WAIT 1 SEC FOR ANY INTRS
                                                    THAT
E275 E2FE        609  D4:       LOOP D4        ; MIGHT OCCUR  等待中断，延迟 1
                                                    秒
E277 E2FE        610  D5:       LOOP D5
E279 0AE4        611            OR   AH, AH    ; DID ANY INTERRUPTS OCCUR?
                                                    有中断发生?若发生(AH)=1，参见第
                                                    652 行
E27B 7408        612            JZ   07        ; NO-GO TO NEXT TEST  发生中断
                                                    则去 TEST. 07
E27D BA0101      613  D6:       MOV  DX, 101H  ; BEEP SPEAKER IF ERROR  送鸣
                                                    响程序用计时值
E280 E8AD03      614            CALL ERR_BEEP  ; GO TO BEEP SUBROUTINE  蜂鸣
                                                    器鸣响禁中断
E283 FA          615            CLI
E284 F4          616            HLT            ; HALT THE SYSTEM  暂停系统
                 617  ;.....................................................
                 618  ; TEST. 7                                  测试 07
                 619  ;     8253 TIMER CHECKOUT  8253 计时器检查
                 620  ; DESCRIPTION  完成:
                 621  ; VERIFY THAT THE SYSTEM TIMER (0) DOESN'T COUNT
                      TOO FAST NOR TOO  计时器 0 检查，保证它能正常工作
```

```
                622  ;      SLOW.
                623  ;.....................................................
E285            624  D7:
E285 B400       625          MOV  AH, 0              ; RESET TIMER INTR RECVD
                                                     FLAG
E287 32ED       626          XOR  CH/CH              ; CLEAR THE CH REG
E289 B0FE       627          MOV  AL, 0FEH           ; MASK ALL INTRS EXCEPT
                                                     LVL 0
E28B E621       628          OUT  INTA01, AL         ; WRITE THE 8259 IMR 写8259
                                                     A OCW1 屏蔽 1~7 级中断，允使
                                                     0 级中断
E28D B010       629          MOV  AL, 00010000B      ; SEL TIM 0, LSB, MODE 0,
                                                     BINARY
E28F E643       630          OUT  TIM_CTL, AL        ; WRITE TIMER CONTROL MO
                                                     DE REG  写计时器控制模式寄
                                                     存器,0 方式,计数结束中断
E291 B116       631          MOV  CL, 16H            ; SET PGM LOOP CNT
E293 8AC1       632          MOV  AL, CL             ; SET TIMER 0 CNT REG  计
                                                     数值 = 16H, 检慢
E295 E640       633          OUT  TIMER0, AL         ; WRITE TIMER 0 CNT REG
                                                     写计数寄存器
E297 F6C4FF     634  D8:     TEST AH, 0FFH           ; DID TIMER 0 INTERRUPT
                                                     OCCUR?  时钟中断发生否? 发
                                                     生(AH) = 1
E29A 7504       635          JNZ  D9                 ; YES-CHECK TIMER OP FOR
                                                     SLOW TIME
E29C E2F9       636          LOOP D8                 ; WAIT FOR INTR FOR
                                                     SPECIFIED TIME  延迟一段时
                                                     间,但不应小于 16H 决定的时间
E29E EBDD       637          JMP  D6                 ; TIMER 0 INTR DIDN T OCC
                                                     UR-ERR  未发生时钟中断，出
                                                     错
E2A0 B112       638  D9:     MOV  CL, 18             ; SET PGM LOOP CNT  计数值
                                                     = 18，检慢
E2A2 B0FF       639          MOV  AL, 0FFH           ; WRITE TIMER 0 CNT REG
E2A4 E640       640          OUT  TIMEP0, AL         ; TIMEPO = 40H(8253 计时器/控
                                                     制器 0 地址) 写计时值
E2A6 B400       641          MOV  AH, 0              ; RESET INTR RECEIVED FLAG
E2A8 B0FE       642          MOV  AL, 0FEH           ; REENABLE TIMER 0 INTERR-
```

```
                                        UPTS 只允使中断0. 因在中断服务程
                                        序(暂时)中将其余中断允使了
E2AA E621    643        OUT  INTAOI, AL
E2AC F6C4FF 644  D10:  TEST AH, 0FFH ;  DID TIMER 0 INTERRUPT OCC-
                                        UR? 时钟中断是否发生? 发生
                                        (AH) = 1
E2AF 75CC    645        JNZ  D6         ; YES-TIMER CNTING TOO FAST,
                                        ERR 中断发生太快出错
E2B1 E2F9    646        LOOP D10        ; WAIT FOR INTR FOR SPECIFIED
                                        UPTS
E2B3 E93600  647        JMP  TST8       ; GO TO NEXT TEST ROUTINE 正常
                                        时间在[12H, 16H]决定的范围内
             648  ;.....
             649  ;      TEMPORARY INTERRUPT SERVICE ROUTINE  暂时中
                                                             断服务
                                                             子程序
             650  ;.....
E2B6         651  D11    PROC NEAR
E2B6 B401    652        MOV  AH, 1   (AH) = 1 标志已发生了中断
E2B8 50      653        PUSH AX         ; SAVE REG AX CONTENTS
E2E9 B0FF    654        MOV  AL, 0FFH   ; MASK ALL INTERRUPTS OFF  屏
                                        蔽所有中断
E2BB E621    655        OUT  INTA01, AL
E2BD B020    656        MOV  AL, EOI
E2B7 E620    657        OUT  INTAOO, AL 写结束中断命令 (OCW2) 清除 1-7 级
                                        中断屏蔽,使较低级的中断能被处理
E2C1 58      658        POP  AX         ; RESTORE REG AX CONTENTS
E2C2 CF      659        IRET            ; 中断返回,恢复原来的各标志位
             660  D11    ENDP
             661
E2C3         662  NMI_INT PROC NEAR ; 下面是不可屏蔽中断处理子程序
E2C3 50      663        PUSH AX         ; SAVE ORIG CONTENTS OF AX
                                        保存 AX
E2C4 E462    664        IN   AL, PORT_C; 读 8255 PC 口, 第 6 位为 1 表示 I/O
                                        RAM 奇偶校验错
E2C6 A840    665        TEST AL, 40H    ; IO CH PARITY CHECK?  是奇偶
                                        校验错?
E2C8 7408    666        JZ   D12        ; YES-FLAG IS SET TO 0  是: 打印
                                        错误信息"PARITY CHECK 2"
```

```
E2CABE19E2R 667          MOV  SI, OFFSET D1    ; ADDR OF ERROR MSG
E2CD B90E00   668          MOV  CX, DIL        ; MSG LENGTH  送信息首址
                                                和长度
E2D0 EB0A     669          JMP  SHORT D13      ; DISPLAY ERROR MSG  去
                                                打印程序
E2D2          670  D12:
E2D2 A880     671          TEST AL, 80H        ; PLANAR RAM P-CHECK?
                                                是系统板 RAM 奇偶校验错?
E2D4 7410     672          JZ   D14            ; NO-AUX INT
E2D6 BE27E2 R 673          MOV  SI, OFFSET D2  ; ADDR OF ERROR MSG  是
                                                打印错误信息 "PARITY CHE
                                                CK 1"
E2D9 B90E00   674          MOV  CX, D2L        ; MSG LENGTH
E2DC          675  D13:
E2DC B80000   676          MOV  AX, 0          ; INIT AND SET MODE FOR
                                                VIDEO
E2DF CD10     677          INT  10H            ; CALL VIDEO_IO PROCED
                                                URE  调 VIDEO-IO 程序, 初
                                                始,设置 CRT
E2E1 E8E603   678          CALL P_MSG          ; PRINT ERROR MSG  打印程
                                                序(实为在 CRT 上显示)
E2E4 FA       679          CLI                 ; 禁止中断
E2E5 F4       680          HLT                 ; HALT SYSTEM  暂停系统
E2E6          681  D14:
E2E6 58       682          POP  AX             ; RESTORE ORIG CONTENTS
                                                OF AX  恢复 AX
E2E7 CF       683          IRET
              684          NMI_INT ENDP        ; NMI 处理程序完
              685  ;.....
              686  ; INITIAL RELIABILITY TEST…PHASE 3  初始可靠性测试
                                                -3
              687  ;.....
              688          ASSUME CS:CODE, DS:DATA
              689
E2E8 20323031 690  E1      DB   '201';'201' 为存贮器有错标记(出错号)
0004          691  E1L     EQU  $-E1
              692
              693  ;       ESTABLISH BIOS SUBROUTINE CALL INTERRUPT
                            VECTORS  建立 BIOS 子程序调用中断向量表
```

```
                694
E2EC            695 TSTB:
E2EC FC         696         CLD                     ; SET DIR FLAG TO GO
                                                      FORWARD 前向标志
E2ED BF4000     697         MOV DI, OFFSET VIDED_INT ; SETUP ADDR TO
                                                      INTR AREA
                                                      送从 VIDED-I
                                                      O 开始 16 个中
                                                      断向量表地址
                                                      (内容)
E2F0 0E         698         PUSH CS
E2F1 1F         699         POP  DS                 ; SETUP ADDR OF VECTOR
                                                      TABLE
                                                      FF13H 是 ROM 区向量表(从
                                                      VIDEO-IO 开始)的首址
E2F2 BE13FF     700         MOV SI, 0FF13H          ; OFFSET VECTOR_TABLE+
                                                      32
E2F5 B92000     701         MOV CX, 20H             ; 计数值为 32.
E2F8 F3         702         REP MOVSW               ; MOVE VECTOR TABLE TO
                                                      RAM 向量表移进 RAM 区,参
                                                      见第 5767～5812 行
E2F9 A5
                703
                704 ;       SETUP TIMER 0 TO MODE 3 为计时器 0 设置模式 3
                705
E2FA B0FF       706         MOV AL, 0FFH            ; DISABKE ALL DEVICE IN
                                                      TERRUPTS 屏蔽所有级别的
                                                      中断
E2FC E621       707         OUT INTA01, AL
E2FE B036       708         MOV AL, 36H             ; SEL TIM 0, LSB, MSB, MODE
                                                      3 设方波频率发生器模式 (模
                                                      式 3)
E300 E643       709         OUT TIMER+3, AL         ; WRITE TIMER MODE REG
E302 B000       710         MOV AL, 0
E304 E640       711         OUT TIMER, AL           ; WRITE LSB TO TIMER 0 R
                                                      EG
E306 E640       712         OUT TIMER, AL           ; WRITE MSB TO TIMER 0
                                                      REG 设置计数值
                713
```

```
                        714 ;       SETUP TIMER 0 TO BLINK LED IF MANUF
                                    ACTURING TEST MODE
                        715
                        716     ASSUME  DS:DATA
E308 B84000       R     717     MOV     AX, DATA        ; POINT DS TO DATA
                                                          SEG
E30B 8ED8               718     MOV     DS, AX
E30D E87803             719     CALL    KBD_RESET       ; SEND SOFTWAR E
                                                          RESET TO KE YBRD
                                                          调 KBD-RE SET 复位
                                                          键盘
E310 80FBAA             720     CMP     BL, 0AAH        ; SCAN CODE 'AA'
                                                          RETURNED? 返回
                                                          了扫描码'AA'?
E313 7426               721     JE      E3              ; YES-CONTINUE(NO
                                                          N MFG MODE)
E315 B03C               722     MOV     AL, 3CH         ; EN KBO, SET KBD
                                                          CLK LINE LOW 允
                                                          使键盘,使键盘时钟线
                                                          低电平
E317 E661               723     OUT     PORT_B, AL      ; WRITE 8255
                                                          PORT B
E319 90                 724     NOP
E31A 90                 725     NOP
E31B E460               726     IN      AL, PORT A      ; RAS A BIT CLOCK
                                                          ED IN? 读PAD,取
                                                          扫描码
E31D 24FF               727     AND     AL, 0FFH
E31F 7516               728     JNZ     E2              ; YES-CONTINUE(NO
                                                          H MFG MODE)
E321 FE061200     R     729     INC     MFG_TST         ; ELSE SET SW FOR
                                                          MFG TEST MODE
                                                          设置制造测试方式并
                                                          记下
E325 26C7062000B2E6 R   730     MOV     ES:INT_ADDR, OFFSET BLINK_INT
                                        ; SETUP TIMER INTR TO BLINK LED
                                          设时钟中断处理程序为闪烁二极管程序
E32C 26C706220000F0 R   731     MOV     ES:INT_ADDR+2, CODE
E333 B0FE               732     MOV     AL, 0FEH        ; ENABLE TIMER INTER
```

```
                                        RUPT  允使时钟中断(0级
                                        中断)
E335 E621       733      OUT  INTA01, AL
E337            734  E2:                ; JUMPER__NOT__IN:
E337 BOCC       735      MOV  AL, 0CCH   ; RESET THE KEYBOARD  复位键盘
E339 E661       736      OUT  PORT__B, AL
                737  ;.....
                738  ; TEST. 05  测试05
                739  ;    ROS CHECKSUM II  ROS CHECKSOM II
                740  ; DESCRIPTION  完成:
                741  ;    A CHECKSUM IS DONE FOR THE 4 ROS MODULES
                     CONTAINING BASIC COOE  做出4个存贮盒带BASIC代
                     码模块的检查和
                742  ;.....
E33B            743  E3:
E33B B204       744      MOV  DL, 4      ; NO. OF ROS MODULES TO CHECK
                                         待查模块的个数为4
E33D BB0060     745      MOV  BX, 6000H  ; SETUP STARTING ROS ADDR  设
                                         置ROM区始址(6000H)
E340            746  E4:                ; CHECK__ROS:
E340 E8C8FE     747      CALL ROS__CHECKSUM; 查一块ROM(16KB)
E343 7507       748      JNE  E5         ; BEEP SPEAKER IF ERROR  是否
                                         错?有去E5
E345 FECA       749      DEC  DL         ; ANY MORE TO DO?  模块个数减1
E347 75F7       750      JNZ  E4         ; YES-CONTINUE
E349 EB0790     751      JMP  E6         ; NO-GO TO NEXT TEST  查完且无
                                         错去测试06
E34C            752  E5:                ; ROS__ERROR:
E34C BA0101     753      MOV  DX, 101H
E34F E8DE02     754      CALL ERR__BEEP  ; BEEP SPEAKER  检查和不为0. 出
                                         错,鸣响蜂鸣器
                755  ;.....
                756  ; TEST. 08  测试08
                757  ;    INITIALIZE AND START CRT CONTROLLER(6845)  初
                     始并启动CRT控制器6845,测试CRT缓存
                758  ;    TEST VIDEO READ/WRITE STORAGE.
                759  ; DESCRIPTION       ; 完成:
                760  ; RESET THE VIDEO ENABLE SIGNAL.  复位CRT允使信
                     号,选择彩色或黑白字符显示模式
```

```
                    761 ;   SELECT ALPHANUMERIC MODE, 40*25, B&W.   用
                            数据格式读/写缓存,检查它的可寻址能力
                    762 ;   READ/WRITE DATA PATTERNS TO STG. CHECK STG
                            ADDRESSABILITY.
                    763 ;.....................................................
E352                764 E6:
E352 E460           765     IN    AL, PORT_A        ; READ SENSE SWITCHES
                                                      取外设配备情况字节
E354 B400           766     MOV   AH, 0
E356 A31000 R       767     MOV   EQUIP_FLAG, AX    ; STORE SENSE SW INFO
                                                      保存至 EQUIP_FLAG
E359 2430           768     AND   AL, 30H           ; ISOLATE VIDEO SWS  取
                                                      CRT 配置位(5, 6位)见
E358 7503           769     JNZ   E7                ; VIDEO SWS SET TO 0?
E350 E99800         770     JMP   E19               ; SKIP VIDEO TESTS FOR
                                                      BURN_IN  未接有 CRT 转
                                                      接器
E360                771 E7:                         ; TEST_VIDEO:
E360 86E0           772     XCHG  AH, AL            ; 接有 CRT 转接器, 测试模
                                                      式
E362 80FC30         773     CMP   AH, 30H           ; B/W CARD ATTACHED?
E365 7409           774     JE    E8                ; YES-SET MODE FOR B/W
                                                      CARD  测试 B/W (单色)转
                                                      接器
E367 FEC0           775     INC   AL                ; SET COLOR MODE FOR
                                                      COLOR CD
E369 80FC20         776     CMP   AH, 20H           ; 80X25 MODE SELECTED?
                                                      测试彩色 80×25 模式
E36C 7502           777     JNE   E8                ; NO-SET MODE FOR 40×
                                                      25  测试彩色 40×25 模式
E36E B003           778     MOV   AL, 3             ; SET MODE FOR 80×25
E370                779 E8:                         ; SET_MODE:
E370 50             780     PUSH  AX                ; SAVE VIDEO MODE ON
                                                      STACK  CRT 模式:
                                                    AL = { 00 B/W 转接器   80×25
                                                         { 01 COLOR 转接器 40×25
                                                         { 11 COLOR 转接器 80×25
E371 2AC4           781     SUB   AH, AH            ; INITIALIZE TO ALPHAN
                                                      UMERIC MD
```

```
E373 CD10          782      INT   10H          ; CALL VIDEO_IO 调 VIDEO-
                                                 IO, CRT 初始成字符 (A/N) 式,
                                                 (AH)、(AL)均为入口参数
E375 58            783      POP   AX           ; RESTORE VIDEO SENSE SWS
                                                 IN AH
E376 50            784      PUSH  AX           ; RESAVE VALUE 恢复 CRT 模
                                                 式开关字节至 AH
E377 BB00B0        785      MOV   BX, 0B000H   ; BEG VIDDO RAM ADDR B/W
                                                 CD B/W CRT 缓存首址 0B00H
                                                 送 BX
E37A BAB803        786      MOV   DX, 3B8H     ; MODE REG FOR B/W B/W转
                                                 接器模式寄存器地址 3B8H
E37D B90010        787      MOV   CX, 4096     ; RAM BYTE CNT FOR B/W CD
                                                 计数器 4K
E380 B001          788      MOV   AL, 1        ; SET MODE FOR BW CARD
E382 80FC30        789      CMP   AH, 30H      ; B/W VIDEO CARD ATTACHE
                                                 D? 连接的是 B/W 转接器?
E385 740B          790      JE    E9           ; YES-GO TEST VIDEO STG
                                                 是,去缓存测试
E387 BB00B8        791      MOV   BX, 0B800H   ; BEG VIDEO RAM ADDR COL
                                                 OR CD 否,置 COLOR 转接器缓
                                                 存始址和模式寄存器地址
E38A BAD803        792      MOV   DX, 308H     ; MODE REG FOR COLOR CD
E38D B90040        793      MOV   CX, 4000H    ; RAM BYTE CNT FOR COLOR
                                                 CD
E390 FEC8          794      DEC   AL           ; SET MODE TO 0 FOR COLOR
                                                 CD AL = 0 表示 COLOR 转接器,
                                                 = 1 表示 B/W 转接器
E392               795  E9:                    ; TEST_VIDEO_STG:
E392 EE            796      OUT   DX, AL       ; DISABLE VIDEO FOR COLOR
                                                 CD 初始 B/W 转接器或禁止 CO
                                                 LOR 转接器视频输出
E393 8EC3          797      MOV   ES, BX       ; POINT ES TO VIDEO RAM
                                                 STG
E395 B84000    R   798      MOV   AX, DATA     ; POINT DS TO DATA SEGME
                                                 NT
E398 8ED8          799      MOV   DS, AX       ES 指向 CRT 缓存段首, DS 指向
                                                 数据段首
E39A 813E72003412 R 800     CMP   RESET_FLAG, 1234H ; POD INIYIAED BY
```

```
                                                KBD RESET?  键盘
                                                复位过?
E3A0 740D     801       JE    E10             ; YES-SKIP VIDEO RAM TEST
                                                键盘复位过则不用再测显示缓存
E3A2 8EDB     802       MOV   DS, I X         ; POINT DS TO VIDEO RAM STG
E3A4 E876FC   803       CALL  STGTST_CNT      ; GO TEST VIDEO R/W STG
                                                测试 CRT 缓存
E3A7 7406     804       JE    E10             ; STG OK-CONTINUE TESTING
                                                正常去下一个测试
E3A9 BA0201   805       MOV   DX, 102H        ; SETUP  OF BEEPT
E3AC E88102   806       CALL  ERR_BEEP        ; GO BEEP SPEAKER  不正常,
                                                蜂鸣器鸣响
              807 ;.....
              808 ; TEST. 09  测试 09
              809 ; SETUP VIDEO DATA ON SCREEN FOR VIDEO LINE TEST.
                    写 CRT 显示字符的属性码和字符码(空白字符的)
              810 ; DESCRIPTION  完成:允使视频, 设置模式,在屏幕上显示一水平
                  横线
              811 ;     ENABLE VIDEO SIGHAL AND SET MODE.
              812 ;     DISPLAY A HORIZONTAL BAR ON SCREEN.
              813 ;.....
E3AF          814 E10:
E3AF 58       815       POP   AX              ; GET VIDEO SENSE SWS(AH)
                                                取模式
E380 50       816       PUSH  AX              ; SAVE IT
E381 B400     817       MOV   AH, 0           ; ENABLE  VIDEO  AND  SET
                                                MODE
E3B3 CD10     818       INT   10H             ; VIDEO  调 VIDEO-IO,  允使视
                                                频,设置模式
E385 B82070   819       MOV   AX, 7020H       ; WRT BLANKS  IN  REVERSE
                                                VIDEO  "70" 是属性码: 视频翻
                                                转, 白背景黑字符, "20" 空白字
                                                符
E3B8 2BFF     820       SUB   DI, DI          ; SETUP STARTING LOC  ASCII
                                                代码
E3BA B92800   821       MOV   CX, 40          ; NO. OF BLANKS TO DISPLAY
                                                显示个数送 CX
E3BD FC       822       CLD                   ; SET DIR FLAG TO INCREME
                                                NT  前向标志
```

```
E3BE F3       823          REP  STOSW          ; WRITE VIDEO STORAGE  写
                                                 CRT 缓存
E3BF AB
              824 ;.....
              825 ; TEST. 10  测试 10
              826 ;       CRT INTERFACE LINES TEST  CRT 接口线测试
              827 ; DESCRIPTION  完成:
              828 ;       SENSE ON/OFF TRANSITION OF THE VIDEO ENABLE
                          AND HORIZONTAL 检测视频允使/水平同步转换功能(开/
                          关)
              829 ;       SYNC LINES.
              830 ;.....
E3C0 58       831          POP  AX            ; GET VIDEO SENSE SW INFO  模
                                                 式信息
E3C1 50       832          PUSH AX            ; SAVE IT
E3C2 80FC30   833          CMP  AH, 30H       ; B/W CARD ATTACHED?
E3C5 BABA03   834          MOV  DX, 03BAH     ; SETUP ADDR OF BW STATUS
                                                 PORT BW 口状态寄存器地址(3BAH)
E3C8 7403     835          JE   E11           ; YES-GO TEST LINES  是 B/W 转接
                                                 器?
E3CA BADA03   836          MOV  DX, 03DAH     ; CDLOR CARD IS ATTACHED  否,
                                                 取 COLOR 口状态寄存器地址(3DAH)
E3CD          837 E11:                        ; LINE_TST:
E3CD B4D8     838          MOV  AH, 8  8:0000  , 1000
E3CF          839 E12:                        ; OFLOOP_CNT:
E3CF 2BC9     840          SUB  CX, CX
E3D1 EC       841 E13:     IN   AL, DX        ; READ CRT STATUS PORT  读CRT
                                                 状态寄存器
E3D2 22C4     842          AND  AL, AH        ; CHECK VIDEO/HORZ LINE  取 A/
                                                 N 视频信号位
E3D4 7504     843          JNZ  E14           ; ITS ON-CHECK IF IT GOES OFF
                                                 高电平表示有视频输出
E3D6 E2F9     844          LOOP E13           ; LOOP TILL ON OR TIMEOUT  循环
E3D8 EB13     845          JMP  SHORT E17     ; GO PRINT ERROR MSG  无视频输
                                                 出,出错
E3DA 2BC9     846 E14:     SUB  CX, CX
E30C EC       847 E15:     IN   AL, DX        ; READ CRT STATUS PORT
E3DD 22C4     848          AND  AL, AH        ; CHECK VIDEO/HORZ LINE  取 A/
                                                 N 视频信号位
```

```
E3DF 74D4      849          JZ    E16          ; ITS ON-CHECK NEXT LINE 视频
                                                  信号变低电平否?
E3E1 E2F9      850          LOOP  E15          ; LOOP IF OFF TILL IT GOES ON
E3E3 EB08      851          JMP   SHORT E17    否,出错
E3E5           852  E16:                      ; NXT_LINE
E3E5 B103      853          MOV   CL, 3        ; GET NEXT BIT TO CHECK
E3E7 D2EC      854          SHR   AH, CL       ; (AH) = 0000, 0001
E3E9 75E4      855          JNZ   E12          ; GO CHECK HORIZONTAL LINE 去
                                                  测试允使显示位
E3EB EB06      856          JMP   SHORT E18    ; DISPLAY CURSOR ON SCREEN
E3ED           857  E17:                      ; CRT_ERR:
E3ED BA0201    858          MOV   DX, 102H
E3F0 E83D02    859          CALL  ERR_BEEP     ; GO BEEP SPEAKER  错误处理程
                                                  序
E3F3           860  E18:                      ; DISPLAY_CURSOR:
E3F3 58        861          POP   AX           ; GET VIDEO SENSE SWS(AH). 测
                                                  试正常转此
E3F4 B400      862          MOV   AH, 0        ; SET MODE AND DISPLAY CURSOR
E3F6 CD10      863          INT   10H          ; CALL VIDEO I/O PROCEDURE 设
                                                  置 CRT 模式
               864  ;.....
               865  ; TEST. 11  测试 11 剩余 RAM 区测试
               866  ;       ADDITIONAL READ/WRITE STORAGE TEST  完成: 为
                                                  基本 16K 后的全部 RAM 区写/读
               867  ; DESCRIPTION  数据检验格式,存贮区的寻址能力也属测试范围
               868  ;       WRITE/READ DATA PATTERNS TO ANY READ/WRI
                       TE STORAGE AFTER THE BASIC
               869  ;       16K. STORAGE ADDRESSABILITY IS CHECKED.
               870  ;.....
               871          ASSUME DS:DATA
E3F8           872  E19:
E3F8B 84000 R  873          MOV   AX, DATA
E3FB 8ED8      874          MOV   DS, AX
               875
               876  ;       DETERMINE RAM SIZE ON PLANAR BOARD  下面小
                           段决定系统板上 RAM 区大小
               877
E3FD 8A261000 R 878         MOV   AH, BYTE PTR EQUIP_FLAG ; GET SENSE SWS
```

```
                                                    INFO  取RAM区
                                                    容量信息
E401 80E40C  879        AND  AH, 0CH          ; ISOLATE RAM SIZE SWS
E404 B004    880        MOV  AL, 4
E406 F6E4    881        MUL  AH  (AH)×(AL)—AX  (AL)=10: 16K RAM
                                                         20: 32K RAM
                                                         30: 48K RAM
                                                         40: 64K RAM
E408 0410    882        ADD  AL, 16           ; ADD BASIC 16K
E40A 8BD0    883        MOV  DX, AX           ; SAVE PLANAR RAM SIZE
                                                    IN DX
E40C 8BD8    884        MOV  BX, AX           ; AND IN BX  系统区 RAM
                                                    容量暂存 BX
             885
             886 ;      DETERMINE IO CHANNEL RAM SIZE  下面程序段决定
                                                    扩展 RAM 区容
                                                    量
             887
E40E E462    888        IN   AL, PORT_C       ; READ IO  CH RAM SIZE
                                                    SWS
E410 240F    889        AND  AL, 0FH          ; ISOLATE FROM OTHER BI
                                                    TS  取扩展 RAM 容量，在
                                                    PC 口低 4 位
E412 B420    890        MOV  AH, 32
E414 F6E4    891        MUL  AH  ×32 得实际外接 RAM 容量 (AL)=00H—00 K
                                                         20H—32 K
                                                         40H—64 K
                                                         60H—96 K
                                                         80H—128K
                                                         AOH—160K
                                                         COH—192K
E416 A31500R 892        MOV  IO_RAM_SIZE, AX  ; SAVE IO CHANNEL RAM
                                                    SIZE  外接 RAM 容量送 IO
                                                    _RAM_SIZE
E419 83FB40  893        CMP  BX, 40H          ; PLANAR RAM SIZE=64K?
                                                    系统板上有 64KRAM?
E41C 7402    894        JE   E20              ; YES-ADD IO CHN  RAM
                                                    SIZE  有，加上外接 RAM
E41E 2BC0    895        SUB  AX, AX           ; NO—DON T ADD ANY IO
```

```
                                                    RAM 无,AX 清 0。
E420                        896 E20:                ; ADD_IO_SIZE:
E420 03C3                   897     ADD AX, BX      ; SUM TOTAL RAM SIZE
                                                    得系统总 RAM 容量,并送
E422 A31300       R         898     MOV MEMORY_SIZE, AX.  ; SETUP MEMO
                                                    RY SIZE PA
                                                    RM MEMOR
                                                    Y—SIZE 单元
                                                    保存
E425 813E72003412 R         899     CMP RESET_FLAG, 1234H ; POD INITIAT
                                                    ED BY KBD
                                                    RESET? 键
                                                    盘复位否?
E42B 744D                   900     JE  E22         ; YES—SKIP MEMORY
                                                    TEST 是,跳过存贮器测
                                                    试
                            901
                            902 ;   TEST ANY OTHER READ/WRITE STORAGE
                                    AVAILABLE 从上面进入本段时(DX)与系统板
                                    RAM 容量的对应关系是:      (DX)  容量
                            903                                 16—16K
E42D BB0004                 904     MOV BX, 400H                32—32K
E430 B91000                 905     MOV CX, 16                  48—48K
E433                        906 E21:                            64—64K
E433 3BD1                   907     CMP DX, CX      ; ANY MORE STG TO BE
                                                    TESTED? 还有末测完的
                                                    模块?
E435 7646                   908     JBE E23         ; NO—GO TO NEXT TEST
                                                    无,去下一个测试
E437 8EDB                   909     MOV DS, BX      ; SETUP STG ADDR IN
                                                    DS AND ES 有.
E439 8EC3                   910     MOV ES, BX
E438 83C110                 911     ADD CX, 16      ; INCREMENT STG BYTE
                                                    COUNTER 计数器加16,
                                                    表示测完 16K
E43E 81C30004               912     ADD BX, 400H    ; SET POINTER TO NEXT
                                                    16K BLK BX 指向下一待
                                                    测块首
E442 51           913 PUSH CX                       ; SAVE REGS 保存 CX,
```

```
                                                    BX, DX 之值
E443 53        914  PUSH BX
E444 52        915  PUSH DX
E445 E8D2FB    916  DALL STGTST                  ; GO TEST A 16K BLK OF
                                                  STG 测试 16K RAM
E448 5A        917  POP  DX
E449 5B        918  POP  BX                     ; RESTORE REGS恢复 DX,
                                                  BX, CX
E44A 59        919  POP  CX
E44B 74E6      920  JE   E21                    ; CHECK IF MORE STG TO
                                                  TEST RAM 模块有错? 没有的
                                                  话去 E21
               921
               922  ; PRINT FAILING ADDRESS AND XOR'ED PATTERN IF DA
                      TA COMPARE ERROR  RAM 模块有错,
               923
E44D 8CDA      924  MOV  DX, DS                 ; CONVERT FAILING HIGH-
                                                  ORDER 下面显示出错字节属
                                                  16KRAM 中哪一K 及错误数据
                                                  格式
E44F 8AE8      925  MOV  CH, AL                 ; SAVE FAILING BIT PATTE
                                                  RN
E451 8AC6      926  MOV  AL, DH                 ; GET FAILING ADDR (HIGH
                                                  BYTE) 取出错字节所在RAM
                                                  块高 8 位地址的高 4 位
E453 B104      927  MOV  CL, 4
E455 D2E8      928  SHR  AL, CL                 ; RIGHT-JUSTIFY HIGH BYTE
                                                  高 4 位调整至低 4 位
E457E 83E00    929  CALL XLAT_PRINT_CODE       ; CONVERT AND PRINT CODE
                                                  取相应 ASCII 码, 显示
E45A 8AC6      930  MOV  AL, DH
E45C 240F      931  AND  AL, 0FH                取高 8 位地址的低 4 位的 ASCII 码, 显示
E45E E83700    932  CALL XLAT_PRINT_CODE       ; CONVERT AND PRINT CODE
E461 8AC5      933  MOV  AL, CH                 ; GET FAILING BIT PATTERN
                                                  取错误字节位格式
E463 B104      934  MOV  CL, 4                  ; AND ISOLATE LEFTMOST
                                                  NIBBLE 调整至低 4 位部分
E465 D2E8      935  SHR  AL, CL
E467 E82E00    936  CALL XLAT_PRINT_CODE       ; CONVERT AND PRINT CODE
```

```
E46A BAC5       937          MOV  AL, CH                  ; GET FAILING BIT PA
                                                             TTERN AND
E46C 240F       938          AND  AL, 0FH                 ; ISOLATE RIGHTMOST
                                                             NIBBLE 取错误字节位
                                                             低4位部分
E46E E82700     939          DALL XLAT_PRINT_CODE         ; CONVERT AND PRINT
                                                             CODE 显示
E471 BEE8E2 R   940          MOV  SI, OFFSET E1           ; SETUP ADDRESS OF
                                                             ERROR MSG
E474 B90400     941          MOV  CX, E1L                 ; GET MSG BYTE COU
                                                             NT
E477 E85002     942          CALL P_MSG                   ; PRINT ERROR MSG
                                                             显示'201'出错类别号
E47A            943  E22:                                ; GO_TST12:
E47A E94A00     944          JMP  TST12                   ; GO TO NEXT TEST
                                                             去下一个测试
E47D            945  E23:                                ; STG_TEST_DONE:
E47D B84000 R   946          MOV  AX, DATA                ; POINT DS TO DATA
                                                             SEGMENT
E480 8ED8       947          MOV  DS, AX                  ; CHG MADE 3/27/81
E482 8B161500 R 948          MOV  DX, IO-RAM-SIZE         ; GET IO CHANNEL R
                                                             AM SIZE 取扩展RAM
                                                             容量字节
E486 0BD2       949          OR   DX, DX                  ; SET FLAG RESULT
E488 74F0       950          JZ   E22                     ; NO IO RAM, GO TO
                                                             NEXT TEST 无外接
                                                             RAM的无需测试去下一
                                                             个测试
E48A B90000     951          MOV  CX, 0                   设计数初值
E48D 81FB0010   952          CMP  BX, 1000H               ; HAS IO RAM BEEN
                                                             TESTED 扩充RAM区
                                                             测试过?
E491 77E7       953          JA   E22                     ; YES - GO ,TO NEXT
                                                             TEST 是,转E22
E493 BB0010     954          MOV  BX, 1000H               ; SETUP BEG LOC FOR
                                                             IO RAM 扩展RAM始
                                                             址送BX, 32K模块分2
                                                             次测完,64K模块分4次
```

测完

```
E496 EB9B      955  JMP  SHORT E21 ; GO TEST IO CHANNEL RAM
               956  ;.....
               957  ; CONVERT AND PRINT ASCII CODE  字符转换及显示子程
                    序 XLAT_PRINT_CODE
               958  ;
               959  ; AL MUST CONTAIN NUMBER TO BE CONVERTED.  AL
                    低4位是待转换成ASCII码的字符
               960  ; AX AND BX DESTROYED.
               961  ;.....
E498           962  XLAT_PRINT_CODE PROC NEAR
E498 1E        963  PUSH DS          ; SAVE DS VALUE
E499 0E        964  PUSH CS          ; POINT DS TO CODE SEG   DS 指向代
                                     段,因ASCII码表在代码区
E49A 1F        965  POP  DS
E49B BBB7E4    966  MOV  BX, 0E4B7H ; OFFSET ASCII_TBL-XLAT TABLE
                                     取ASCII码表首址 0EA4B7H
E49E D7        967  XLATB                    转换指令,取ASCII码送
E49F B40E      968  MOV  AH, 14          送入口参数,14表示显示ASCII字符
E4A1 B700      969  MOV  BH, 0                (BH)=0,置显示页为0
E4A3 CD10      970  INT  10H         ; CALL VIDEO_IO  调用显示程序
E4A5 1F        971  POP  DS          ; RESTORE ORIG VALVE IN DS   恢复
                                     DS原来的内容
E4A6 C3        972  RET
               973  XLAT_PRINT_CODE ENOP  返回
LOC OBJ        LINE  SOURCE
               974  ;.....
               975  ; INITIAL RELIABILITY TEST……PHASE 4   初始可靠性测
                    试—4
               976  ;.....
               977       ASSUME CS:CODE, DS:DATA
E4A7 20333031  978  F1  DB   '301'
   0004        979  F1L EQU  $-F1     ; KEYBOARD MESSAGE  键盘错误号
                                     '301'
E4AB 313331    980  F2  DB   '131'
   0003        981  F2L EQU  $-F2     ; CASSETTE MESSAGE  盒带机错误号
                                     '131'
E4AE 363031    982  F3  DB   '601'
   0003        983  F3L EQU  $-F3     ; DISKETTE MESSAGE  磁盘机错误号
```

```
                        984
E4B1                    985 F4    LABEL  WORD          ; PRINTER SOURCE TAB
                                                         LE
E4B1 BC03               986       DW     3BCH           显示,一打印口地址
E4B3 7803               987       DW     378H           并行打印口地址
E4B5 7802               988       DW     278H           保留
E4B7                    989 F4E   LABEL  WORD
E4B7 303132333435       990 ASCII_TBL DB '0123456789ABCDEF' 'ASCII 字符 0～
                                                         9, A～F 表
     3637383941424344
     4546
                        991 ;.....
                        992 ; TEST. 12  测试 12 键盘测试
                        993 ;       KEYBOARD TEST
                        994 ; DESCRIPTION  完成:
                        995 ;       RESET THE KEYBOARD AND CHECK THAT SC
                                    AN CODE AA' IS RETURNED  复位键盘, 检查是
                                    否返回了扫描码 AA'
                        996 ;       TO THE CPU. CHECK FOR STUCK KEYS.
                        997 ;.....
E4C7                    998 TST12:
E4C7 B84000    R        999       MOV    AX, DATA       ; POINT DS TO DATA
                                                         SEG  DS 指向数据段
E4CA 8ED8              1000       MOV    DS, AX
E4CC 803E120001 R      1001       CMP    MFG_TST, 1     ; MANUFACTURING TEST
                                                         MODE? 是制造测试模式?
E4D1 7439              1002       JE     F7             ; YES—SKIP KEYBOAPD
                                                         TEST  是,跳过本程序段
E4D3 E8B201            1003       CALL   KBD_RESET      ; ISSUE SOFTWARE RES
                                                         ET TO KEYBRD  否, 复
                                                         位键盘
E4D6 E32B              1004       JCXZ   F6             ; PRINT ERR MSG IF NO
                                                         INTERRUPT  (CX) = 0表
                                                         示无键盘中断,出错
E4D8 B04D              1005       MOV    AL, 4DH        ; ENABLE KEYBOARD
                                                         允使键盘工作(第七位为0)
E4DA E661              1006       OUT    PORT_B, AL
E40C 80FBAA            1007       CMP    BL, 0AAH       ; SCAN CODE AS EXPEC
```

```
                                                    TED? 返回了'AA'?
E4DF 7522      1008          JNE   F6              ; NO--DISPLAY ERROR MSG  否,
                                                     出错,打出'301'出错号
               1009
               1010  ;       CHEDK FOR STUCK KEYS  下面检查STUCK键
               1011
E4E1 BOCC      1012          MOV   AL, 0CCH        ; CLR KBD, SET CLK LINE HIGH
                                                     清键盘
E4E3 E661      1013          OUT   PORT_B, AL
E4E5 B04C      1014          MOV   AL, 4CH         ; ENABLE KBD, CLK IN NEXT BY
                                                     TE  允使键盘
E4E7 E661      1015          OUT   PORT_B, AL
E4E9 2BC9      1016          SUB   CX, CX
E4EB           1017  F5:                           ; KBD_WAIT
E4EB E2FE      1018          LOOP  FE              ; DELAY FOR A WHILE  等待一段
                                                     时间,延时值由(CX)决定
E4ED E460      1019          IN    AL, KBD_IN      ; CHECK FOR STUCK KEYS  KBD
                                                     _IN是PA口的地址,取扫描码
E4EF 3C00      1020          CMP   AL, 0           ; SCAN CODE = 0?  0扫描码?
E4F1 7419      1021          JE    F7              ; YES--CONTINUE TESTING  是,
                                                     转F7
E4F3 8AE8      1022          MOV   CH, AL          ; SAVE SCAN CODE
E4F5 B104      1023          MOV   CL, 4
E4F7 D2E8      1024          SHR   AL, CL          ; RIGHT_JUSTIFY HIGH BYTE
                                                     扫描码从高4位移至低4位
E4F9 E89CFF    1025          CALL XLAT_PRINT_CODE  ; CONVERT AND PRINT
                                                     显示扫描码高4位部分
E4FC 8AC5      1026          MOV   AL, CH          ; RECOVER SCAN CODE
E4FE 240F      1027          AND   AL, 0FH         ; ISOLATE LOW ORDER BYTE
E500 E895FF    1028          CALL XLAT_PRINT_CODE  ; CONVERT AND PRINT
                                                     显示扫描码低4位部分
E50 3BEA7E4 R  1029  F6:  MOV  SI, OFFSET F1 ; GET MSG ADDR
E506 B90400    1030          MOV   CX, F1L         ; GET MSG BYTE COUNT
E509 E8BE01    1031          CALL P_MSG            ; PRINT MSG ON SCREEN  显示
                                                     '301'错误类别号
               1032
               1033  ;       SETUP INTERRUPT VECTOR TABLE  送整个中断向量表
                              内容
               1034
```

```
E50C            1035  F7:                              ; SETUP_INT_TABLE:
E50C 2BC0       1036      SUB  AX, AX
E50E 8EC0       1037      MOV  ES, AX
E510 B93000     1038      MOV  CX, 24*2                  ; GET VECTOR CNT 计数
                                                         器,整个向量表占48个字
E513 0E         1039      PUSH CS;                       ; SETUP DS SEG REG
E514 1F         1040      POP  DS
E515 BEF3FE     1041      MOV  SI, 0FEF3H                ; OFFSET VECTOR_TABL
                                                         E 中断向量表存在ROM区
                                                         FEF3H起的若干单元中,现
                                                         送始址
E518 BF2000     1042      MOV  DI, OFFSET INT_PTR        RAM 区向量表始址
E51B FC         1043      CLD                                前向标志
E51C F3         1044      REP  MOVSW                         送向量表至RAM 区
E51D A5
                1045  ;.....
                1046  ; TEST. 13  测试13盒带机测试
                1047  ;   CASSETTE DATA WRAP TEST
                1048  ; DESCRIPTION  完成:关闭盒带机马达,写一位至盒带机数
                1049  ;   TURN CASSETTE MOTOR OFF. WRITE A BIT OUT TO
                          THE CASSETTE DATA BUS据线,然后读取它,检查其正确否
                1050  ;   VERIFY THAT CASSETTE DATA READ IS WITHIN A
                          VALID RANGE.
                1051  ;.....
                1052
                1053  ;   TURN THE CASSETTE MOTOR OFF   下面4条指令关闭
                          盒带机马达
                1054
E51E B84000 R   1055      MOV  AX,DATA                   ; POINT DS REG TO DATA
                                                         SEG  DS 指向数据段
E521 8ED8       1056      MOV  DS, AX
E523 B04D       1057      MOV  AL, 04DH                  ; SET TIMER 2 SPK OUT,
                                                         AND CASST
E525 E661       1058      OUT  POST_B, AL                ; OUT BITS ON, CASSETTE
                                                         MOT OFF 关闭马达,计时
                                                         器2门输出
                1059
                1060  ;   WRITE A BIT
                1061
```

```
E527 B0FN      1062        MOV  AL, 0FFH        ; DISABLE TIMER INTERRUPTS
                                                  禁止任何中断
E529 E621      1063        OUT  INTA01, AL
E52B B0B6      1064        MOV  AL, 0B6H        ; SEL TIM2, LSB, MSB, MD3
E52D E643      1065        OUT  TIMER+3, AL     ; WRITE 8253 CMD/MODE REG
                                                  设置 8253 计时器 2 方波发生器模
                                                  式
E52F B8D304    1066        MOV  AX, 1235        ; SET TIMER 2 CNT FOR 1000
                                                  USEC
E532 E642      1067        OUT  TIMER+2, AL     ; WRITE TIMER 2 COUNTER
                                                  REG 写计时值,产生周期1000HS
                                                  的方波,表示"1"。
E534 8AC4      1068        MOV  AL, AH          ; WRITE MSB
E536 E642      1069        OUT  TIMER+2, AL
               1070
               1071  ;     READ CASSETTE INPUT 读盒带机输入
               1072
E538 E462      1073        IN   AL, PORT_C      ; READ VALUE OF CASS IN
                                                  BIT
E53A 2410      1074        AND  AL, 10H         ; ISOLATE FROM OTHER BITS
                                                  读取盒带机输出位内容(PC 口第4
                                                  位)
E53C A26B00 R  1075        MOV  LAST_VAL, AL    送上次值,以便 READ_HALF_B
                                                  IT 中进行比较
E53F E83E14    1076        CALL READ_HALF_BIT
E542 E83B14    1077        CALL READ_HALF_BIT   BX 是半个周期的脉冲宽度
                                                  (READ_HACF_BIT 子程
                                                  序出口参数)
E545 E30C      1078        JCXZ F8              ; CAS_ERR (CX)=0 延时太长,
                                                  出错
E547 81FB4005  1079        CMP  BX, MAX_PERIOD  脉冲太宽?
E54B 7306      1080        JNC  F8              ; CAS_ERR 是出错。
E54D 81FB1004  1081        CMP  BX, MIN_PERIOD
E551 7309      1082        JND  F9              ; GO TO NEXT TEST IF OK 否,
                                                  脉冲太窄?脉冲宽度正常转 F9
E553           1083  F8:                       ; CAS_ERR:
E553BEABE4 R   1084        MOV  SI, OFFSET F2   ; CASSETTE WRAP FAILED 显
                                                  示盒带错误号'131'
E556 B90300    1085        MOV  CX, F2L
```

```
E559 E86E01   1086          CALL P_MSG          ; GO PRINT ERROR MSG
              1087  ;.....
              1088  ; TEST. 14  测试14
              1089  ;       DISKETTE ATTACHMENT TEST  磁盘机连接测试
              1090  ; DESCRIPTION  完成:检查IPL驱动器是否连接上，若连上，检
                    验复位
              1091  ;       CHECK IF IPL DISKETTE DRIVE IS ATTACHED TO
                    SYSTEM. IF ATTACHED,  操作后,NEC FDC状态正常
                    否. 发RECAL, SEEK命令
              1092  ;       VERIFY STATUS OF NEC FDC AFTER A RESET. ISSUE
                    A RECAL AND SEEK
                    给FDC检查状态信息. 完成系统初始操作后控制
              1093  ;       CMD TO FDC AND CHECK STATUS. COMPLESE SYS
                    TEM INIIALIZATION THEN  权转给引导程序
              1094  ;       PASS CONTROL TO THE BOOT LOADER PROGRAM.
              1095  ;.....
E55C          1096  F9:
E55C B0FC     1097          MOV  AL, 0FCH       ; ENABLE TIMER AND KBD
                                                  INTS  允使计时器及键盘中断
E55E E621     1098          OUT  INTA01, AL
E560 A01000 R 1099          MOV  AL, BYTE PTR EQUIP_FLAG  ; GET  SENSE
                                                  SWS INFO
E563 A801     1100          TEST AL, 01H        ; IPL DISKETTE DRIVE ATT
                                                  CH?  IPL驱动器连上?
E565 7503     1101          JNZ  F10            ; YES- TEST DISKETTE CONTR
E567 E9B900   1102          JMP  F22;           ; NO—SKIP THIS TEST  未连
                                                  上,跳过下面程序段,转F22
E56A          1103  F10:                       ; DISK_TEST:  连上，进行测试
E56A B0BC     1104          MOV  AL, 0BCH       ; ENATBLE DISKETTE, KEYB
                                                  OARD,
E56C E621     1105          OUT  INTA01, AL     ; AND TIMER INTERRUPTS
                                                  允使磁盘机，键盘和计时器中断
E56E B400     1106          MOV  AH, 0          ; RESET NEC FDC
E570 CD13     1107          INT  13H            ; VERIFY STATUS AFTER RE
                                                  SET  调INT 13H复位磁盘机,
                                                  出口信息(AH)的安排同DISK
                                                  ETTE_STATUS单元,见第122
                                                  ~132行
E572 F6C4FF   1108          TEST AH, 0FFH       ; STATUS OK?
```

```
E575 7520       1109            JNZ   F13             ; NO—FDC FAILED  全零表
                                                        示无错
                1110
                1111    ;       TURN DRIVE 0 MOTOR ON  无错
                1112
E577 BAF203     1113            MOV   DX, 03F2H       ; GET ADDR OF FDC CARD
                                                        03F2H 是磁盘机 FOC 数据输
                                                        出寄存器的地址
E57A B01C       1114            MOV   AL, 1CH         ; TURN MOTOR ON, EN DM
                                                        A/INT   1CH 使驱动器A马
                                                        达开，非 FDC 复位态，允使驱
                                                        动器 A
E57C EE         1115            OUT   DX, AL          ; WRITE FDC CONTROL REG
E57D 2BC9       1116            SUB   CX, CX
E57F            1117    F11:                          ; MOTOR_WAIT:
E57F E2FE       1118            LOOP  F11               WAIT FOR 1 SECOND
                                                        等待 1 秒
E581            1119    F12:                          ; MOTOR_WAIT1:
E581 E2FE       1120            LOOP  F12
E583 33D2       1121            XOR   DX, DX          ; SELECT DRIVE 0   选 0 号
                                                        (A)驱动器
E585 B501       1122            MOV   CH, 1           ; SELECT TRACK 1   让磁头
                                                        移至磁道 1
E587 88163E00 R 1123            MOV   SEEK_STATUS, DL  0 送 SEEK_STATUS
E58B E8F308     1124            CALL  SEEK            ; RECALIBRATE DISKETTE
                                                        重校驱动器 A，参见第2690行
                                                        开始的 SEEK 入口
E58E 7207       1125            JC    F13             ; GO TO ERR SUBROUTINE
                                                        IF ERR  出口参数说明
E590 B552       1126            MOV   CH, 34          ; SELECT TRACK 34  CY = 0
                                                        表示无错
E592 E8EC08     1127            CALL  SEEK            ; SEEK TO TRACK 34  磁头
                                                        移至磁道 34，重校驱动器
E595 7309       1128            JNC   F14             ; OK, TURN MOTOR OFF
E597            1129    F13:                          ; DSK_ERR: 磁盘机有错
E597 BEAEE4 R   1130            MOV   SI, OFFSET F3   ; GET ADDR OF MSG
E59A B90300     1131            MOV   CX, F3L         ; GET MSG BYTE COUNT 显
                                                        示磁盘机错误号 '601'
```

```
E59D E82A01              1132          CALL P_MSG          ; GO PRINT ERROR
                                                             MSG
                         1133
                         1134  ;       TURN DRIVE 0 MOTOR OFF  关闭A驱动器
                                       马达
                         1135
E5A0                     1136  F14:                        ; DR0_OFF:
E5A0 B00C                1137          MOV  AL, 0CH        ; TURN DRIVE 0 M
                                                             OTOR OFF 关闭马
                                                             达参数(第4位置0)
E5A2 BAF203              1138          MOV  DX, 03F2H      ; FDC CTL ADDRESS
                                                             关闭0号驱动器马达
E5A5 EE                  1139          OUT  DX, AL
                         1140
                         1141  ;       SETUP PRINTER AND RS232 BASE ADDRE
                                       SSES IF DEVICE ATTACHED 设立连接打印
                                       机和RS232通讯口基址
                         1142
E5A6                     1143  F15:                        ; JMP_BOOT
E5A6  C7061A001E00 R     1144          MOV  BUFFER_HEAD, OFFSET KB_BUFF
                                       ER  ; SETUP KEYBOARD PARAMET
                                             ERS 建立键盘缓冲区,先头尾重
                                             合
E5AC C7061C001E00  R     1145          MOV  BUFFER_TAIL, OFFSET KB_BUFFER
E5B2 BDB1E4        R     1146          MOV  BP, OFFSET F4 ; PRT_SRC_TBL
                                                             取打印机资源表首址
                                                             参见第985行
E5B5 BE0000              1147          MOV  SI, 0
E5B8                     1148  F16:                        ;PRT_BASE
E5B8 2E8B5600            1149          MOV  DX, CS:[BP]    ; GET PRINTER BASE
                                                             ADDR 取打印机口
                                                             地址(数据口)
E5BC B0AA                1150          MOV  AL, 0AAH       ; WRITE DATA TO
                                                             PORT A 写数
E5BE EE                  1151          OUT  DX, AL
E5BF 2AC0                1152          SUB  AL, AL
E5C1 EC                  1153          IN   AL, DX         ; READ PORT A
                                                             读数
E5C2 3CAA                1154          CMP  AL, 0AAH       ; DATA PATTERN
```

```
                                                    SAME  写进口子的数
                                                    据与读出的相同?
E5C4 7506              1155        JNE   F17;       , NO — CHECK  NEXT
                                                    PRT CD  相同说明该
                                                    打印机存在
E5C6 89940800   R      1156        MOV   PRINTER_BASE[SI], DX  ; YES-STOR
                                                                  E PRT BA
                                                                  SE ADDR
                                                                  该打印机存
                                                                  在，保存该
                                                                  机数据口地
                                                                  址
E5CA 46                1157        INC   SI          ; INCREMENT TO NE
                                                    XT WORD  至PRINT
                                                    _BASE  指出的表区。
E5CB 46                1158        INC   SI          指向下一个字单元
E5CC                   1159  F17:                    ; NO_STORE:
E5CC 45                1160        INC   BP          ; POINT TO NEXT BA
                                                    SE ADDR  准备取下
                                                    一打印机数据口地址
E5CD 45                1161        INC   BP
E5CE 81FDB7E4   R      1162        CMP   BP, OFFSET F4E ; ALL POSSIBLE ADD
                                                    RS CHECKED?  三个
                                                    可能的数据口均查完?
E5D2 75E4              1163        JNE   F16         ; PRT_BASE  否，再查
E5D4 BB0000            1164        MOV   BX, 0       ; POINTER  TO RS232
                                                    TABLE
E5D7 BAFA03            1165        MOV   DX, 3FAH    ; CHECK IF RS232 CD
                                                    1 ATTCH?  3FAH:中
                                                    断标识寄存器地址
E5DA EC                1166        IN    AL, DX      ; READ INTR ID REG
                                                    读该寄存器
E5DB A8F8              1167        TEST  AL, 0F8H    测其高5位是否恒为0。恒为0
                                                    表示该通讯口存在
E5DD 7508              1168        JNZ   F18
E5DF C7870000F803 R    1169        MOV   RS232_BASE[BX], 3F8H  ; SETUP RS
                                                                  232 CD #1
                                                                  ADDR  存
                                                                  在，保存通
```

```
                                                    讯口第一寄存器(TX 缓存器)
                                                    地址
E5E5 43              1170          INC   BX
E5E6 43              1171          INC   BX
E5E7 BAFA02          1172  F18:    MOV   DX, 2FAH   ; CHECK IF RS232 CD 2 AT
                                                    TCH 测试第二通讯口是否接
                                                    上
E5EA EC              1173          IN    AL, DX     ; REAE INTERRUPT ID REG
E5EB A8F8            1174          TEST  AL, 0F8H
E5ED 7508            1175          JNZ   F19        ; BASE_END
E5EF C7870000F802 R  1176          MOV   RS232_BASE[BX], 2F8H  ; SETUP RS232
                                                    CD #2 存在,
                                                    保存其 TX 缓
                                                    存器地址 2F
                                                    8H
E5F5 43              1177          INC   BX
E5F6 43              1178          INC   BX
                     1179
                     1180  ; ……SET UP EQUIP FLAG TO INDICATE NUMBER
                             OF PRINTERS AND RS232 CARDS
                     1181
E5F7                 1182  F19:                   ; BASE_END:
E5F7 8BC6            1183          MOV   AX, SI:  ; SI HAS 2* NUMBER OF
                                                    RS232 2*打印机个数
E5F9 B103            1184          MOV   CL, 3    ; SHIFT COUNT 准备移至高
                                                    4 位
E5FB D2CB            1185          ROR   AL, CL   ; ROTATE RIGHT 3 POSITIO
                                                    NS
E5FD 0AC3            1186          OR    AL, BL   ; OR IN THE PRINTER COU
                                                    NT 或入RS232 个数
E5FF A21100    R     1187          MOV BYTE PTR EQUIP_FLAG+1, AL
                                   ; STORE AS SECOND BYTE 结果见下页首
E602 BA0102          1188          MOV   DX, 201H
E605 EC              1189          IN    AL, DX
E606 A80F            1190          TEST  AL, 0FH  测试有游戏转接器否?
E608 7505            1191          JNZ   F20      ; NO_GAME_CARD 无
E60A 800E110010 R    1192          OR    BYTE PTR EQUIP_FLAG+1, 16 有:EQUIP
                                   _FLAG+1 单元第 4 位置 1
E60F                 1193  F20:                   ; NO_GAME_CARD:
```

EQUIP_FLAG+1 字节的含义

	位 7	6	5	4	3	2	1	
1	0	1	0	0	0	1	0	0
2	1	0	0	0	1	0	0	0
3	1	1	0	0	1	1	0	0
	个数		PRINTER		RS232			

```
                        1194
                        1195  ;     ENABLF NMI INTERRUPTS  允许NMI中断
                        1196
E60F B080               1197        MOV  AL, 80H          ; ENABLE NMI INTERR
                                                             UPTS
E611 E6A0               1198        OUT  0A0H, AL
E613 803E120001 R       1199        CMP  MFG_TST, 1       ; MFG MODE  是制造测
                                                             试模式?
E618 7406               1200        JE   F21              ; LOAD_BOOT_STRAP
                                                             是,马上调进引导程序
E61A EA0100             1201        MOV  DX, 1            ;
E61D E81000             1202        CALL ERR_BEEP         ; BEEP 1 SHORT TONE
                                                             蜂鸣器鸣响一会儿
E620                    1203  F21:                        ; LOAD_BOOT_STRAP:
E620 E9CF00             1204        JMP  BOOT_STRAP       ; GO TO THE BOOT LO
                                                             ADER  去引导装入程序
E623                    1205  F22:                        ; LOOP_POO:
E623 803E120001 R       1206        CMP  MFG_TST, 1       ; MANUFACTURING TE
                                                             SY MODE? 是制造测试
                                                             模式?
E628 7503               1207        JNE  F23              ; NO_GO TO BOOT LO
                                                             ADER  否,去F15测试,
                                                             RS232. 打印机连接情况
E62A E92EFA             1208        JMP  START            ; YES_LOOP POWER_
                                                             ON_DIAGS  去267行,
                                                             重新进行TEST.01
E62D                    1209  F23:                        ; GO_TO_BOOT:
E62D E976FF             1210        JMP  F15              ; JMP_BOOT 返回第1143
                                                             行
                        1211  ;.....................................................
                        1212  ; INITIAL RELIABILITY TEST...SUBROUTINS  初始可
                              靠性测试用到的子程序
```

```
                1213 ;.....
                1214        ASSUME CS:CODE, DS:DATA
                1215 ;.....
                1216 ;      SUBROUTINES FOR POWER ON DIAGNOS TICS开机
                           诊断(测试)子程序
                1217 ;.....
                1218 ;      THIS PROCEDURE WILL ISSUE ONE LONG TONE
                           (3 SECS) AND ONE OR 下面程序段让蜂鸣器鸣一声
                           长音(3 秒)和一声短音
                1219 ; MORE SHORT TONES(1 SEC)TO INDICATE A FAILUE
                           ON THE PLANAR (1 秒),以指出系统板或 RAM
                                        模块或 CRT 有错
                1220 ;      BOARD, A BAD RAM MODULE, OR A PROBLEM
                           WITH THE CRT.
                1221 ; ENTRY PARAMETERS: 入口参数
                1222 ;      DH = NUMBER OF LONG TONES TO BEEP (DH):长
                           音鸣响时间控制
                1223 ;      DL = NUMBER OF SHORT TONES TO BEEP. (DL):
                           短音鸣时间控制
                1224 ;.....
E630            1225        ERR_BEEP PROC NEAR
E630 9C         1226        PUSHF           ; SAVE FLAGS 保存所有标志位
E631 FA         1227        CLI             ; DISABLE SYSTEM INTERRUP
                                              TS 禁止中断
E632 1E         1228        PUSH DS         ; SAVE DS REG CONTENTS 保
                                              存 DS 内容
E633 B84000 R   1229        MOV AX, DATA    ; POINT DS TO DATA SEG DS
                                              指向数据段
E636 8ED8       1230        MOV DS, AX
E638 0AF6       1231        OR  DH, DH      ; ANY LONG ONES TO BEEP
                                              要鸣长音?
E63A 7418       1232        JZ  G3          ; NO, DO THE SHORT ONES 否,
                                              去鸣短音
E63C            1233 G1:                    ; LONG_BEEP:
E63C B306       1234        MOV BL, 6       ; COUNTER FOR BEEPS 一次鸣
                                              响延续时间=(BL)×0.5 秒
E63E E82500     1235        CALL BEEP       ; DO THE BEEP 调鸣响子程序
E641 E2FE       1236 G2:    LOOP G2         ; DELAY BETWEEN BEEPS 鸣响
                                              间隔
```

```
E643 FECE          1237          DEC  DH              ; ANY MORE TO DO  长音
                                                        鸣响完?
E645 75F5          1238          JNZ  G1              ; DO IT  否,
E647 80E120001 R   1239          CMP  MFG_TST, 1      ; MFG TEST MODE?  是,现
                                                        为制造测试模式?
E64C 7506          1240          JNE  G3              ; YES_CONTINUE BEEPING
                                                        SPEAKER
E64E B0CD          1241          MOV  AL, 0CDH        ; STOP BLINKING LED  制
                                                        造测试模式,停止LED闪烁
E650 E661          1242          OUT  PORT_B, AL
E652 EBE8          1243          JMP  SHORT G1
E654               1244  G3:                          ; SHORT_BEEP  鸣短音
E654 B301          1245          MOV  BL, 1           ; COUNTER FOR A SHORT
                                                        BEEP  一次鸣响时间=0.5
                                                        秒
E656 E80D00        1246          CALL BEEP            ; DO THE SOUND
E659 E2FE          1247  G4:     LOOP G4              ; DELAY BETWEEN BEEPS
                                                        鸣响间隔
E65B FECA          1248          DEC  DL              ; DONE WITH SHORTS  鸣
                                                        响完?
E65D 75F5          1249          JNZ  G3              ; DO SOME MORE
E65F E2FE          1250  G5:     LOOP G5              ; LONG  DELAY  BEFORE
                                                        RETURN  是,延迟大约1秒
E661 E2FE          1251  G6:     LOOP G6
E663 1F            1252          POP  DS              ; RESTORE ORIG CONTENTS
                                                        OF DS  恢复DS,标志位
E664 90            1253          POPF                 ; RESTORE FLAGS TO ORIG
                                                        SETTINGS
E665 C3            1254          RET                  ; RETURN TO CALLR  返回
                   1255  ERR_BEEP  ENDP
                   1256
                   1257  ;       ROUTINE TO SOUND BEEPER  鸣响子程序
                   1258
E666               1259  BEEP    PROC  NEAR
E666 B0B6          1260          MOV  AL,10110110B    ; SEL TIM  2,  LSB,  MSB,
                                                        BINARY  设置计时器2模式
E668 E643          1261          OUT  TIMER+3, AL     ; WRITE THE TIMER MODE
                                                        REG
E66A B83305        1262          MOV  AX, 533H        ; DIVISOR FOR 1000 HZ 1000
```

```
                                                    HZ分频值
E66D E642   1263        OUT  TIMER+2, AL    ; WRITE TIMER 2 CNT—LSB
                                                    写计时器2低字节
E66F 8AC4   1264        MOV  AL, AH
E671 E642   1265        OUT  TIMER+2, AL    ; WRITE TIMER 2 CNT—MSB
                                                    写计时器2高字节
E673 E461   1266        IN   AL, PORT_B     ; GET CURRENT SETTING OF
                                                    PORT
E675 8AE0   1267        MOV  AH, AL         ; SAVE THAT SETTINGH  将
                                                    PB口当前字节送AL暂存
E677 0C03   1268        OR   AL, 03         ; TURN SPEAKER ON  打开扬
                                                    声器
E679 E661   1269        OUT  PORT_B, AL
E67B 2BC9   1270        SUB  CX, ;CX        ; SET CNT TO WAIT 500 MS
E67D E2FE   1271  G7:   LOOP G7             ; DELAY BEFORE TURNING
                                                    OFF
E67F FECB   1272        DEC  BL             ; DELAY CNT EXPIRED?
E681 75FA   1273        JNZ  G7             ; NO — CONTINUE BEEPING
                                                    SPK  等待(BL)×0.5秒时间后
                                                    关闭扬声器
E683 8AC4   1274        MOV  AL, AH         ; RECOVER VALUE OF PORT
                                                    准备恢复PB口原先内容
E685 E661   1275        OUT  PORT_B, AL
E687 C3     1276        RET                 ; RETURN TO CALLER  返回
            1277  BEEP ENDP
            1278  ; ..........................................................
            1279  ;     THIS PROCEDURE WILL SEND A SOFTWARE RESET
                        TO THE KEYBOARD.  本程序段复位键盘
            1280  ;     SCAN CODE AA' SHOULD BE RETURNED TO THE
                        CPU. 复位后,扫描'AA'应返回 CPU
            1281  ; ..........................................................
E688        1282  KBD_RESET  PROC  NEAR
E688 B00C   1283        MOV  AL, 0CH        ; SET KBD CLK LINE LOW
                                                    置键盘时钟线低电平
E68A E661   1284        OUT  PORT_B, AL     ; WRITE 8255 PORT B
E68C B95629 1285        MOV  CX, 10582      ; HOLD KBD CLK LOW FOR
                                                    20 MS  延迟20毫秒
E68F E2FE   1286  G8:   LOOP G8             ; LOOP FOR 20 MS
```

```
E691 B0CC    1287          MOV  AL, 0CCH      ; SET CLK, ENABLE LINES HIGH 置时钟,允使线高电平
E693 E661    1288          OUT  PORT_B, AL
E695         1289  SP_TEST:  ; ENTRY FOR MANUFACTURING TEST 2 本入口是制造测试2入口
E695 B04C    1290          MOV  AL, 4CH       ; SET KBD CLK HIBH, ENABLE LOW 置键盘时钟高电平, 允使键盘
E697 E661    1291          OUT  PORT_B, AL
E699 B0FD    1292          MOV  AL, 0FDH      ; ENABLE KBYBOARD INTERRUPTS 允使键盘中断(中断1)
E69B E621    1293          OUT  INTA01, AL    ; WRITE 8259 IMR
E69D FB      1294          STI                ; ENABLE SYSTEM INTERRUPTS 置位IF位, 允许可屏蔽中断
E69E B400    1295          MOV  AH, 0         ; RESET INTERRURT INOICATOR 清中断发生指示单元(AH),
E6A0 2BC9    1296          SUB  CX, CX        ; SETUP INTERRUPT TIMEOUT CNT
E6A2 F6C4FF  1297  G9:     TEST AH, 0FFH      ; DID A KEYBOARD INTR OCCUR? 发生键盘中断否?全0表示未发生
E6A5 7502    1298          JNZ  G10           ; YES—READ SCAN CODE RETURNED
E6A7 E2F9    1299          LOOP G9            ; NO—LOOP TILL TIMEOUT 循环等待,直至超时
E6A9 E460    1300  G10:    IN   AL, PORT_A    ; READ KEYBOARD SCAN CODE 读扫描码
E6AB 8AD8    1301          MOV  BL, AL        ; SAVE SCAN CODE JUST READ
E6AD B0CC    1302          MOV  AL, 0CCH      ; CLEAR KEYBOARD 清键盘
E6AF E661    1303          OUT  PORT_B, AL
E6B1 C3      1304          RET                ; RETURN TO CALLER 返回
             1305  KBO_RESET  ENOP
             1306  ; ..........
             1307  ;     BLINK LED PROCEDURE FOR MFG BURN—IN AND RUN—IN TESTS 制造测试 BUR—IN, RUN—IN时闪烁 LED 程序
```

```
                1308  ;       (LED WILL BLINK APPROXIMATELY .25 SECOND)
                                        (让 LEN 闪烁 0.25 秒)
                1309  ; ..........................................................
E6B2            1310  BLINK_INT  PROC  NEAR
E6B2 FB         1311        STI   允许中断
E6B3 51         1312        PUSH CX              ; SAVE CX REG CONTENTS
E6B4 50         1313        PUSH AX              ; SAVE AX REG CONTENTS
E6B5 E461       1314        IN   AL, PORT_B      ; READ CDRRENT VAL OF
                                                   PORT B  读 PB 口
E6B7 24BF       1315        AND  AL, 0BFH                第 6 位置0.
E6B9 E661       1316        OUT  PORT_B, AL      ; BLINK LED  输出至 PB 口，点
                                                   亮 LED
E6BB 28C9       1317        SUB  CX, CX
E6BD E2FE       1318  G11:   LOOP G11                      延迟
E6BF 0C40       1319        OR   AL, 40H         ; STOP BLINKING LED  PB 口
                                                   第 6 位置 1. LED 灭
E6C1 E661       1320        OUT  PORT_B, AL
E6C3 B020       1321        MOV  AL, EOI
E6C5 E620       1322        OUT  INTA00, AL
E6C7 58         1323        POP  AX              ; RESTORE AX REG
E6C8 59         1324        POP  CX              ; RESTORE CX REG
E6C9 CF         1325        IRET
                1326  BLINK_INT  ENDP
                1327  ; ..........................................................
                1328  ;       THIS SUBROUTINE WILL PRINT A MESSAGE ON THE
                              DISPLAY  本程序段将一字符中显示在屏幕上
                1329  ;
                1330  ; ENTRY REQUIREMENTS:  入口参数
                1331  ;       SI = OFFSET (ADDRESS) OF MESSAGE BUFFER
                              (SI): 字符串缓存区偏移地址
                1332  ;       CX = MESSAGE BYTE COUNT  (CX):字符串长
                1333  ;       MAXIMUM MESSAGE LENGTH IS 36 CHARACTERS
                              字符串长不能超过 36 个字符
                1334  ; ..........................................................
E6CA            1335  P_MSG  PROC  NEAR
E6CA B84000 R   1336        MOV  AX, DATA        ; POINT DS TO DATA SEG
E6CD 8ED8       1337        MOV  DS, AX   DS 指向数据段，准备检查现是否处制造
                                                   测试模式
E6CF 803E120001 R  1833  CMP  MFG_TST, 1     ; MFG TEST MODE?
```

```
E6D4 7505      1339        JNE   G12              ; NO—DISPLAY ERROR MSG
                                                    是制造测试模式?
E6D6 B601      1340        MOV   DH, 1            ; YES—SETUP TO BEEP SPE-
                                                    AKER 是,鸣响蜂鸣器
E6D8 E955FF    1341        JMP   ERR_BEEP         ; YES—BEEP SPEAKER
E6DB           1342  G12:                        ; WRITE_MSG:
E6DB 2E8A04    1343        MOV   AL, CS:[SI]      ; PUT CHAR IN AL  非制造测
                                                    试模式,取显示字符
E6DE 46        1344        INC   SI               ; POINT TO NEXT CHAR
                                                    SI 指向下一字符
E6DF B700      1345        MOV   BH, 0            ; SET PAGE * TO ZERO  设
                                                    0 显示页
E6E1 B40E      1346        MOV   AH, 14           ; WRITE CHAR (TTY-INTERF-
                                                    ACE)  写字符方式
E6E3 CD10      1347        INT   10H              ; CALL VIDEO_IO  调 VIDEO
                                                    _IO 显示该字符
E5E5 E2F4      1348        LOOP  G12              ; CONTINUE TILL MSG WRIT-
                                                    TEN  继续显示
E6E7 B80D0E    1349        MOV   AX, 0E0DH        ; POSITION CURSOR TO NEXT
                                                    LINE  整个字符串显示完毕,现
                                                    让 CRT 换行,
E6EA CD10      1350        INT   10H              ; SEND CARRIAGE RETURN
                                                    AND  (AL) = OD 是回车码
E6EC B80A0E    1351        MOV   AX, 0E0AH        ; LINE FEED CHARS  输出换
                                                    行码 0A
E6EF CD10      1352        INT   10H
E6F1 C3        1353        RET   返回  测试子程序到此结束,开机后进行的 15 个测
                                 试程序到此结束
               1354  P_MSG  ENDP
               1355  ; ……INT 19……  软中断 19
               1356  ; BOOT STRAP LOADER  引导装入程序
               1357  ;     IF A 5 1/4" DISKETTE DRIVE IS AVATLABLE  如
                           果系统接有-5-1/4 吋磁盘机的话, 软中断 19 将该磁盘机
                           的 0 磁道, 1 扇区内容读入内存引导位置
               1358  ;     ON THE SYSTEM, TRACK 0, SECTOR 1 IS READ
                           INTO THE
               1359  ;     BOOT LOCATION (SEGMENT 0, OFFSET 7C00)
                                 (0 段,偏移地址 7C00)随后控制权转向该区
```

```
                1360  ;       AND CONTROL IS TRANSFERRED THERE.
                1361  ;
                1362  ;       IF THERE IS NO DISKETTE DRIVE, OR IF  THERE
                                IS  如果无磁盘区或有硬件错,控制权转给盒带 BASIC 入
                1363  ;       IS A HARDWARE ERROR CONTROL IS TRANSFER-
                                RED 口处
                1364  ;       TO THE CASSETTE BASIC ENTRY POINT.
                1365  ;
                1366  ; IPL ASSUMPTIONS  IPL 驱动器即 0 号驱动器或系统驱动器,
                                它的存在与否由 8255
                1367  ;       8255 PORT 60H BIT 0  PA 口第 0 位指出, 第 0 位为 1
                                说明接了.
                1368  ;       =1 IF IPL FROM DISKETTE
                1369  ;.....
                1370          ASSUME CS:OOE, DS:DATA
E6F2            1371  BOOT_STRAP  PROC  NEAR
                1372
E6F2 FB         1373          STI                     ; ENABLE INTERRUPTS
                                                        允许中断
E6F3 B84000 R   1374          MOV  AX, DATA           ; ESTABLISH  ADDRES-
                                                        SING
E6F6 8ED8       1375          MOV  DS, AX                       DS 指向数据段
E6F8 A11000 R   1376          MOV  AX, EQUIP_FLAG     ; GET THE EQUIPMENT
                                                        SWITCHES  取外设配置
                                                        情况字节 EQUIP-FLAG,
                                                        其第 0 位为 1 说明接有
                                                        IPL
E6FB A801       1377          TEST  AL, 1             ; ISOLATE  IP1  SENSE
                                                        SWITCH  驱动器
E6FD 7423       1378          JZ    H3                ; GO TO CASSETTE BA-
                                                        SIC ENTRY POINT
                1379
                1380  ; ..... MUST LOAD SYSTEM FROM DISKETTE···CX HAS
                        RETRY COUNT  有系统驱动器,准备调入引导程序
                1381
E6FF B90400     1382          MOV  CX, 4              ; SET RETRY COUNT
                                                        (CX) = 4, 最多重复读 4
                                                        次磁盘
E702            1383  H1:                            ; IPL_SYSTEM
```

```
E702 51          1384          PUSH CX       ; SAVE RETRY COUNT  保存重复计
                                               数器
E703 B400        1385          MOV  AH, 0    ; RESET THE DISKETTE SYSTEM
E705 CD13        1386          INT  13H      ; DISKETTE_IO  调 DISKETTE_IO,
                                               复位 IPL 驱动器
E707 7214        1387          JC   H2       ; IF ERROR, TRY AGAIN  CY=1 表
                                               示驱动器有错,再试一次(如果 (CX) 未
                                               减至 0 的话)
E709 B402        1388          MOV  AH, 2    ; READ IN THE SINGLE SECTOR
                                               读一扇区
E70B BB0000      1389          MOV  BX, 0    ; TO THE BOOT LOCATION
E70E 8EC3        1390          MOV  ES, BX   ES 送 0, 指向 0 段,软中断 13 中规定读
                                               出数据存放区首址由 ES:BX 指出
E710 BB007C      1391          MOV  BX, OFFSET BOOT_LOCN  送 BX
E713 BA0000      1392          MOV  DX, 0    ; DRIVE 0, HEAD 0  (DH):磁头号,
                                               (DL):驱动器号,现均为 0
E716 B90100      1393          MOV  CX, 1    ; SECTOR 1, TRACK 0  (CH): 磁道
                                               号, (CL): 扇区号, 现为 0 磁道, 1 扇
                                               区
E719 B001        1394          MOV  AL, 1    ; READ ONE SECTOR  (AL): 读入的
                                               扇区数,现为 1
E71B CD13        1395          INT  13H      ; DISKETTE_IO  调 DISKETTE_IO
E71D 59          1396  H2:     POP  CX       ; RECOVER RETRY COUNT  恢复重
                                               复计数器
E71E 7304        1397          JNC  H4       ; CF SET BY UNSUCCESSFUL  READ
                                               已重复了 4 次?
E720 E2E0        1398          LOOP H1       ; DO IT FOR RETRY TIMES
                 1399
                 1400  ; .....UNABLE TO IPL FROM THE DISKETTE  无磁盘机
                       或重复读磁盘 4 次,都有错,控制权转给盒带 BASIC
                 1401
E722             1402  H3:                 ; CASSETTE_JUMP:
E722 CD18        1403          INT  18H      ; USE INTERRUPT VECTOR TO GET
                                               TO BASIC  控制权转给盒带 BASIC
                 1404
                 1405  ; .....IPL WAS SUCCESSFUL
                 1406
E724             1407  H4:
E724 EA007C000 1408            JMP  BOOT_LOCN  引导程序已调入, 且没有发生磁盘
```

```
                                        错,控制转向 BOOT__LOCN
1409  BOOT__STRAP  ENDP
1410  ; ……INT 14……
1411  ; RS232__IO                    软中断 14—通讯口 I/O(RS232__IO)
1412  ;          THIS ROUTINE PROVIDES BYTE STREAM I/O TO
                 THE COMMUNICATIONS  本程序段完成复位 RS232 通
                 讯口,发送,接收串行数据
1413  ;          PORT ACCORDING TO THE PARAMETERS;  取通讯
                 口状态信息四大功能
1414  ;          (AH) = 0 INITALIZE THE COMMUNICATIONS PORT
        (AH) = 0  初始 RS—232 通讯口, (AL) 是初始时为通讯口设置的
1415  ;            (AL)HAS PARMS FOR INITIALIZATION 参数,其各
                   位的安排(低 5 位) 与 RS—232 转接器线路控制寄存器
                   的完全一致,(AL)格式
1416  ;
1417  ;            7     6     5      4     3          2
1418  ;            ……BAUD RATE…  —PARITY…  STOPBIT
……                (RS—232波特率)  (校验方式)  (仃止位个数)
                   1           0          见左表
                   ……WORD LENGTH……      返回参数同(AH) = 3时的
                        (字长)            返回参数
1420  ;            000—110     X0—NONE    0—1    10—7 BITS
1421  ;            001—150     01—ODD     1—2    11—8 BITS
1422  ;            010—300     11—EVEN
1423  ;            011—600
1424  ;            100—1200
1425  ;            101—2400
1426  ;            110—4800
1427  ;            111—9600
1428  ;          ON RETURN, CONDITIONS SET AS IN CALL TO
                 COMMO STATUS(AH = 3)
1429  ; (AH) = 1  SEND THE CHARACTER IN (AL) OVER THE
                 COMMO LINE  (AH) = 1 将 AL 中的数据发向通讯线
1430  ;          (AL)REGISTER IS PRESERVED
1431  ;          ON EXIT, BIT 7 OF AH IS SET IF THE ROUTINE
                 WAS UNABLE TO  出口:若数据未能由本程序段发
                 出,AH 的
1432  ;          TO TRANSMIT THE BYTE OF DATA OVER THE
                 LINE. THE  第 7 位为 1,其余位的定义见(AH) = 3 取
```

```
                        状态信息的返回参数中 AH 的第 0～第 6 位
1433  ;                 REMAINDER OF AH IS SET AS IN A STATUS
                        REQUEST,
1434  ;                 REFELECTING THE CURRENT STATUS OF THE
                        LINE.
1435  ; (AH) = 2        RECEIVE A CHARACTER IN(AL) FROM COMMO
                        LINE BEFORE  (AH) = 2. 从通讯线接收字符，并送
                        AL.
1436  ;                 RETURNING TO CALLER
1437  ;                 ON EXIT, AH HAS THE CURRENT LINE STATUS
                        AS SET BY THE  出口，(AH) 为当前线路状态，同
                        (AH) = 3时相
1438  ;                 THE STATUS ROUTINE, EXCEPT THAT THE ONLY
                        BITS  应的返回信息. 注意(AH)中 7, 4, 3, 2, 1 位
1439  ;                 LEFT ON ARE THE ERROR BITS (7, 4, 3, 2, 1)
                        指出了错误的原因，且第 7 位为 1 时指出未
1440  ;                 IN THIS CASE, THE TIME OUT BIT INOICATES
                        DATA SET  收到数据装置发来的准备好信号而不
1441  ;                 READY WAS NOT RECEIVED  是超时错误. (AH)
                        非零说明有错.
1442  ;                 THUS, AH IS NON ZERO ONLY WHEN AN ERROR
                        OCCURRED.
1443  ; (AH) = 3        RETURN THE COMMO PORT STATUS IN (AX)
                        (AH) = 3将通讯口状态信息送 AX，各位的定义是:
1444  ;                 AH CONTAINS THE LINE CONTROL STATUS
                        AH 记录线路控制状态
1445  ;                 BIT 7 = TIME OUT  = 1 超时
1446  ;                 BIT 6 = TRANS SHIFT REGISTER EMPTY  = 1发送
                        移位寄存器空
1447  ;                 BIT 5 = TRAN HOLDING REGISTER EMPTY
                        = 1 发送保持寄存器空
1448  ;                 BIT 4 = BREAK DETECT  = 1 断线检测
1449  ;                 BIT 3 = FRAMING ERROR  = 1 帧错
1450  ;                 BIT 2 = PARITY ERROR  = 1 校验错
1451  ;                 BIT 1 = OVERRUN ERROR  = 1超时错
1452  ;                 BIT 0 = DATA READY  = 1数据准备好
1453  ;                 AL CONTAINS THE MODEM STATUS  AL 记录调
                        制解调器状态
1454  ;                 BIT 7 = RECEVED LINE SIGNAL DETECT  线路信
```

```
                        号接收检测
1455  ;                 BIT 6=RING INDICATOR  响铃指示
1456  ;                 BIT 5=DATA SET READY  数据装置准备好
1457  ;                 BIT 4=CLEAR TO SEND  清除后发送
1458  ;                 BIT 3=DELTA RECEIVE LINE SIGNAL DETECT
                        DELTA 接收线信号检测
1459  ;                 BIT 2=TRAILING EDGE RING DETECTOR  下沿
                        响铃检测
1460  ;                 BIT 1=DELTA DATA SET READY  DELTA数据装
                        置准备好
1461  ;                 BIT 0=DELTA CLEAR TO SEND  DELTA清除后发
1462  其它入口参数:
1463  ; (DX)=PARAMETER INDICATING WHICH RS232 CARD(0,1
      ALLOWED)  (DX):指定用哪一个RS_232(取值0或1)
1464  ; DATA AREA-RS232-BASE CONTAINS THE BASE ADDRESS
      OF THE 8250 ON THE CARD  RS232_BASE 是通讯口地址区
      首址
1465  ; LOCATION 400H CONTAINS UP TO 4 RS232 ADDRESSES
      POSSIBLE
1466  ; OUTPUT  出口参数 AX (见上面的定义)
1467  ;           AX  MODIFIED ACCORDING TO PARMS OF
                      CALL  其他寄存器内容不变
1468  ;           ALL OTHERS UNCHANGED
1469  ; ..........................................................
1470      ASSUME CS:CODE, DS:DATA
E729  1471  A1 LABEL  WORD
E729 1704  1472          DW  1047  ; 110 BAUD  ; TABLE OF INIT
                                                VALUE  定义波特率
                                                常数
E72B 0003  1473          DW  768   ; 150
E72D 8001  1474          DW  384   ; 300
E72F C000  1475          DW  192   ; 600
E731 6000  1476          DW  96    ; 1200
E733 3000  1477          DW  48    ; 2400
E735 1800  1478          DW  24    ; 4800
E737 0C00  1479          DW  12    ; 9600
           1480
E739       1481  RS232_IO  PROC  FAR
           1482
```

```
                    1483  ; ……VECTOR TO APPROPRIATE ROUTINE  根据(AH)转
                                向不同的处理程序
                    1484
E739 FB             1485        STI                 ; INTERRUPTS BACK ON  允许
                                                      中断
E73A 1E             1486        PUSH DS             ; SAVE SEGMENT  保存寄存器值
E73B 52             1487        PUSH DX
E73C 56             1488        PUSH SI
E73D 57             1489        PUSH DI
E73E 51             1490        PUSH CX
E73F 8BF2           1491        MOV  SI, DX         ; RS232 VALUE TO SI
                                                      RS232 号送 SI
E741 D1E6           1492        SHL  SI, 1          ; WORD OFFSET  左移一位，取得
                                                      该通讯口在地址的偏移量
E743 BA4000 R       1493        MOV  DX, DATA       DX, DS 指向数据段
E746 8EDA           1494        MOV  DS, DX         ; SET UP OUR SEGMENT
E748 8B940000 R     1495        MOV  DX, RS232_BASE[SI]  ; GET BASE ADDR-
                                                             ESS  取通讯口地址
E74C 0BD2           1496        OR   DX, DX         ; TEST FOR 0 BASE ADDRESS
                                                      该地址值为0说明该通讯口不存在
E74E 7416           1497        JZ   A3             ; RETURN  通讯口不存在,返回
E750 0AE4           1498        OR   AH, AH         ; TEST FOR(AH)=0  是初始通讯
                                                      口?
E752 7418           1499        JZ   A4             ; COMMUN INIT  是,去A4
E754 FECC           1500        DEC  AH             ; TEST FOR(AH)=1  (AH)减1
E756 744E           1501        JZ   A5             ; SEND AL  发送数据?是去A5
E758 FECC           1502        DEC  AH             ; TEST FOR (AH)=2  否, (AH)
                                                      再减 1
E75A 7503           1503        JNZ  A2                返回通讯口状态?是的话去A2
E75C E98900         1504        JMP  A12            ; RECEIVE INTO AL  否，是接收
                                                      数据,去A12
E75F                1505  A2:
E75F FECC           1506        DEC  AH             ; TEST FOR(AH)=3  (AH) 再减
                                                      1,看其是否是0
E761 7503           1507        JNZ  A3  (AH)参数不合法,软中断14处理程序返回
E763 E9B900         1508        JMP  A18            ; COMMUNICATION STATUS
                                                      (AH)=3. 读取RS232状态
E766                1509  A3:                       ; RETURN FROM RS232
E766 59           1510        POP  CX               恢复各寄存器内容
```

```
E767 5F          1511          POP   DI
E768 5E          1512          POP   SI
E769 5A          1513          POP   DX
E76A 1F          1514          POP   DS
E76B CF          1515          IRET                    ; RETURN TO CALLER,
                                                         NO ACTION 返回
                 1516
                 1517  ; ……INITIALIZE THE COMMUNICATIONS PORT 下面
                       进行初始通讯口操作(有关 DX 地址部分注释最后括号
                       中的十六进制数表示第一RS232 通讯口各寄存器的地
                       址)
                 1518
E76C             1519  A4:
E76C 8AE0        1520          MOV   AH, AL            ; SAVE INIT PARMS IN
                                                         AH 入口参数送 AH
E76E 83C203      1521          ADD   DX, 3             ; POINT TO 8250 CONTROL
                                                         REGISTER
E771 B080        1522          MOV   AL, 80H
E773 EE          1523          OUT   DX, AL            ; SET DLAB = 1
                 1524
                 1525  ; ……DETERMINE BAUD RATE DIVISOR 下面决定波特
                       率分频值
                 1526
E774 8AD4        1527          MOV   DL, AH            ; GET PARMS TO DL
                                                         入口参数送 DL
E776 D0C2        1528          ROL   DL, 1
E778 D0C2        1529          ROL   DL, 1             ; GET BAUD RATE TERM
                                                         TO LOW BITS 循环左移
                                                         DL 3 位, 使其高 3 位波特率
                                                         值移入低 3 位
E77A D0C2        1530          ROL   DL, 1
E77C D0C2        1531          ROL   DL, 1             ; *2 FOR WORD TABLE
                                                         ACCESS (DL)×2, (DL)
                                                         作波特率表偏移量用
E77E 81E20E00    1532          AND   DX, 0EH           ; ISOLATE THEM 高 4 位
                                                         送.
E782 BF29E7 R    1533          MOV   DI, OFFSET A1     ; BASE OF TABLE 取波特
                                                         率表始址, 见第 1471 行
E785 03FA        1534          ADD   DI, DX            ; PUT INTO INDEX REGIS-
```

```
                                                    TER (DI)是波特率常数存
                                                    贮单元的地址
E787 8B940000 R  1535        MOV DX, RS232_BASE[SI] ; POINT TO HIGH
                                                    ORDER OF DIVIS-
                                                    OR
E78B 42          1536        INC DX                 (DX)为RS_232分频值锁存器
                                                    高字节部分的地址(DLAB = 1)
E78C 2E8A4501    1537        MOV AL, CS:[DI] + 1 ; GET HIGH ORUER OF
                                                    DIVISOR 送分频值高字节
                                                    部分
E790 EE          1538        OUT DX, AL          ; SEC MS OF DIV TO 0
E791 4A          1539        DEC DX
E792 2E8A05      1540        MOV AL, CS:[DI]     ; GET LOW ORDER OF
                                                    DIVISOR 送分频值低字节
                                                    部分
E795 EE          1541        OUT DX, AL          ; SET LOW OF DIVISOR
E796 83C203      1542        ADD DX, 3           (DX) 线路控制寄存器地址(3FB)
E799 8AC4        1543        MOV AL, AH          ; GET PARMS BACK
E79B 241F        1544        IND AL, 01FH        ; STRIP OFF THE BAUD
                                                    BITS 取入口参数(AL)的
                                                    第0～第4位
E79D EE          1545        OUT DX, AL          ; LINE CONTROL TO 8
                                                    BITS 送线路控制寄存器
E79E 83EA02      1546        SUB DX, 2  (DX):中断允许寄存器地址(3F9)
E7A1 B000        1547        MOV AL, 0
E7A3 EE          1548        OUT DX, AL          ; INTERRUPT ENABLES
                                                    ALL OFF 禁止RS232任
                                                    何中断发生
E7A4 EB79        1549        JMP SHORT A18       ; COM_STATUS 去A18取
                                                    RS_232状态信息
                 1550
                 1551  ; ……SEND CHARACTER IN (AL) OVER COMMO LINE
                           下面程序小段将(AC)逐位发往通讯线
                 1552
E7A6             1553  A5:
E7A6 50          1554        PUSH AX             ; SAVE CHAR TO SEND
                                                    保存待发数据
E7A7 83C204      1555        ADD DX, 4           ; MODEM CONTROL REG-
                                                    ISTER DX 指向 MODEM
```

```
                                                    控制寄存器(3FC)
E7AA B003       1556            MOV   AL, 3        ; DTR AND RTS
E7AC EE         1557            OUT   DX, AL       ; DATA TERMINAL READY, REQ-
                                                     UEST TO SEND 告诉 MODEM:数
                                                     据终端准备好,请求发送
E7AD 33C9       1558            XOR   CX, CX       ; INITIALIZE TIME OUT COUNT
                                                     CX 清 0.
E7AF 83C202     1559            ADD   DX, 2        ; MODEM STATUS REGISTER DX
                                                     指向 MODEM 状态寄存器(3FE)
E7B2            1560    A6:                        ; WAIT_DATA_SET_READY
E7B2 EC         1561            IN    AL, DX       ; GET MODEM STATUS 读MODEM
                                                     状态寄存器
E7B3 A820       1562            TEST  AL, 20H      ; DATA SET READY 测试其第 5
                                                     位:数据装置准备好位
E7B5 7508       1563            JNZ   A7           ; TEST_CLEAR_TO_SEND 为 1
                                                     转 A7,说明数据装置准备好
E7B7 E2F9       1564            LOOP  A6           ; WAIT_DATA_SET_READY 循
                                                     环,等待准备好
E7B9 58         1565            POP   AX
E7BA 80CC50     1566            OR    AH, 80       ; INDICATE TIME OUT 超时,置位
                                                     AH第 7 位
E7BD EBA7       1567            JMP   A3           ; RETURN 返回
E7BF            1568    A7:                        ; TEST_CLEAR_TO_SEND
E7BF 2BC9       1569            SUB   CX, CX       (CX)送 0.
E7C1            1570    A8:                        ; WAIT_CLEAR_TO_SEND
E7C1 EC         1571            IN    AL, DX       ; GET MODEM STATUS
E7C2 A810       1572            TEST  AL, 10H      ; TEST CLEAR TO SEND 读取 MO
                                                     DEM 状态寄存器第 4 位,清除后发送
                                                     位
E7C4 7508       1573            JNZ   A9           ; CLEAR_TO_SEND 清除发送准
                                                     备好?
E7C6 E2F9       1574            LOOP  A8           ; WAIT_CLEAR_TO_SEND 循环
                                                     等待
E7C8 58         1575            POP   AX           ; TIME OUT HAS OCCURRED 超
                                                     时,置位 AH 第 7 位
E7C9 80CC80     1576            OR    AH, 80H
E7CC EB98       1577            JMP   A3           ; RETURN 返回
E7CE            1578    A9:                        ; CLEAR_TO_SEND
E7CE 4A         1579            DEC   DX           ; LINE STATUS REGISTER DX 指
```

```
                                                    向线路状态寄存器(3FD)
E7CF 2BC9          1580         SUB   CX, CX    ; INITIALIZE WAIT COUNT
                                                    清 CX
E7D1               1581   A10:                  ; WAIT_SEND
E7D1 EC            1582         IN    AL, DX    ; GET STATUS  取线路状态
E7D2 A820          1583         TEST  AL, 20H   ; IS TRANSMITTER READY
                                                    测第 5 位发送器准备好
E7D4 7508          1584         JNZ   A11       ; OUT_CHAR    准备好的话去
                                                    A11
E7D6 E2E9          1585         LOOP  A10       ; GO BACK FOR MORE, AND
                                                    TEST FOR TIME OUT  循环,
                                                    等待其准备好
E7D8 58            1586         POP   AX        ; RECOVER ORIGINAL INPUT
E7D9 80CC80        1587         OR    AH, 80H   ; SET THE TIME OUT BIT  置
                                                    超时标志,返回
E7DC EB88          1588         JMP   A3        ; RETURN
E7DE               1589   A11:                  ; OUT_CHAR
E7DE 83EA05        1590         SUB   DX, 5     ; DATA PORT   DX 指向 RS232
                                                    数据口(3F8)
E7E1 59            1591         POP   CX        ; RECOVER IN CX TEMPOR-
                                                    ARILY  待发送数据送 CL
E7E2 8AC1          1592         MOV   AL, CL    ; GET OUT CHAR TO AL FOR
                                                    OUT, STATUS IN AH  (CL)
                                                    送 AL
E7E4 EE            1593         OUT   DX, AL    ; OUTPT CHARACTER  数据送
                                                    数据口
E7E5 E97EFF        1594         JMP   A3        ; RETURN  返回
                   1595
                   1596   ; ……RECEIVE CHARACTER FROM COMMO LINE
                            下面是处理接收字符的程序段, 进入时 DX 指向 RS232 基
                            地址
                   1597
E7E8               1598   A12:
E7E8 802671007F R  1599         AND   BIOS_BREAK, 07FH  ; TURN OFF BRE-
                                                          AK BIT IN BYTE
                                                          BREAK 位置.
E7ED 83C204        1600         ADD   DX, 4     ; MODEM CONTROL REGISTER
                                                    DX 指向 MODEM 控制寄存器
```

```
                                                    (3FC)
E7F0 B001          1601          MOV  AL, 1      ; DATA TERMINAL READY
E7F2 EE            1602          OUT  DX, AL     1 送该寄存器,表明数据终端准备好
E7F3 83C202        1603          ADD  DX, 2      ; MODEM STATUS REGISTER
                                                    DX 指向 MODEM 状态寄存器
                                                    (3FE)
E7F6 2BB9          1604          SUB  CX, CX     ; ESTABLISH TIME OUT COU-
                                                    NT CX 清 0.
E7F8               1605  A13:                    ; WAIT_DSR
E7F8 EC            1606          IN   AL, DX     ; MODEM STATUS 读取MOD-
                                                    EM 状态
E7F9 A820          1607          TEST AL, 20H    ; DATA SET READY 测试数
                                                    据终端准备好否
E7FB 7507          1608          JNZ  A15        ; IS IT READY YET 准备好去
                                                    A15.
E7FD E2F9          1609          LOOP A13        ; WAIT UNTIL IT IS 循环,等
                                                    待准备完毕
E7FF               1610  A14:                    ; TIME_OUT_ERR
E7FF B480          1611          MOV  AH, 80H    ; SET TIME OUT ERROR 置
                                                    超时错标志
E801 E962FF        1612          JMP  A3         ; RETURN WITH ERROR 返回
E804               1613  A15:                    ; WAIT_DSR_END
E804 4A            1614          DEC  DX         ; LINE STATUS REGISTER
                                                    DX 指向线路状态寄存器 (3FD)
E805               1615  A16:                    ; WAIT_RECV
E805 EC            1616          IN   AL, DX     ; GET STATUS 读取其状态
E806 A801          1617          TEST AL, 1      ; RECEIVE BUFFER FULL 测
                                                    试第 0 位,看看数据是否已接收
                                                    并存于接收缓存寄存器
E808 7509          1618          JNZ  A17        ; GET CHAR 是,去A17 取数据
E80A F606710080 R  1619          TEST BIOS_BREAK, 80H ; TEST FOR BREAK
                                                    KEY 进入软中断
                                                    14后若键盘BREAK
                                                    键被按下,则终止接
                                                    收数据工作
E80F 74F4          1620          JZ   A16        ; LOOP IF NOT 等待
E811 EBEC          1621          JMP  A14        ; SET TIME OUT ERROR 去
                                                    A14,置超时错
E813               1622  A17:                    ; GET_CHAR
```

```
E813 241E          1623          AND  AL, 00011110B  ; TEST FOR ERROR CONDITIONS ON RECV CHAR
                                                       保留 AL 的 1, 2, 3, 4 位,即错误指示位,让他们
E815 8AE0          1624          MOV  AH, AL         ; SAVE THIS PART OF STATUS FOR LATER OPERATION
                                                       作为出口参数的一部分
E817 8B940000 R    1625          MOV  DX, RS232_BASE[SI] ; DATA PORT DX 指向 RS232 数据口 (3F8)
E81B EC            1626          IN   AL, DX         ; GET CHARACTER FROM LINE 读入数据
E81C E947FF        1627          JMP  A3             ; RETURN 返回
                   1628
                   1629  ; ……COMMO PORT STATUS ROUTINE 下面 7 条指令读取 RS232 状态
                   1630
E81F               1631  A18:
E81F 8B940000 R    1632          MOV  DX, RS232_BASE[SI] (DX)为 RS232 通讯口基地址 (3F8)
E823 83C205        1633          ADD  DX, 5          ; CONTROL PORT DX指向线路状态寄存器(3FD)
E826 EC            1634          IN   AL, DX         ; GET LINE CONTROL STATUS 读取状态
E827 8AE0          1635          MOV  AH, AL         ; PUT IN AH FOR RETURN 送 AH
E829 42            1636          INC  DX             ; POINT TO MODEM STATUS REGISTER DX 指向 MODEM 状态寄存器 (3FC)
E82A EC            1637          IN   AL, DX         ; GET MODEM CONTROL STATUS 读取状态送 AL
E82B E938FF        1638          JMP  A3             ; RETURN 返回
                   1639  RS232_IO  ENDP
                   1640  ; ……INT 16……
                   1641  ; KEYBOARD I/O 软中断 16-键盘 I/O(KEYBOARDI/O)
                   1642  ;       THESE ROUTINES PROVIDE KEYBOARD SUPPORT
                                  完成取键入字符或状态功能
                   1643  ; INPUT 入口参数:
```

```
                1644  ;       (AH) = 0  READ THE NEXT ASCII CHARACTER
                                          STRUCK FROM THE KEYBOAR
                1645  ;                 RETURN THE RESULT IN (AL), SCAN
                                          CODE IN(AH)  (AH) = 0 读取下一个键入
                                                       字符,结果在 A2 中返回, 扫
                                                       描码在 AH 返回
                1646  ;       (AH) = 1  SET THE Z FLAG TO INDICATE IF AN
                                          ASCII CHARACTER IS AVAILABLE
                1647  ;                 TO BE READ  (AH) = 1 可读取 ASCII 字符
                                          的话置位 ZF
                1648  ;                 (ZF) = 1——NO CODE AVAILABLE
                                          (ZF) = 1……无字符可读
                1649  ;                 (IF) = 0……CODE IS AVAILABLE
                                          (ZF) = 0 字符可读
                1650  ;                 IF ZF = 0, THE NEXT CHARACTER IN
                                          THE BUFFER TO BE READ IS
                1651  ;                 IN AX, AND THE ENTRY REMAINS IN
                                          THE BUFFER  若 ZF = 0, 缓存区中下一个待
                                          读字符在 AX 中.
                1652  ;       (AH) = 2  RETURN THE CURRENT SHIFT STATUS
                                          IN AL REGISTER
                1653  ;                 THE BIT SETTINGS FOR THIS CODE ARE
                                          INDICATED IN THE
                1654  ;                 THE EQUATES FOR KB_FLAG
                                          (AH) = 2 在 AL 中返回当前移位状态, 其各位
                                          的安排同 KB_FLAG(见 76～80 行)
                1655  ; OUTPUT  出口参数:
                1656  ;       AS NOTED ABOVE, ONLY AX AND FLAGS CHAN-
                              GED
                1657  ;       ALL REGISTERS RETAINED  见上面入口参数,本程序
                              只改变 AX 与全部标志位,其余寄存器内容不变
                1658  ;.....................................
                1659          ASSUME CS:CODE, DS:DATA
E82E            1660  KEYBOARD_IO  PROC  FAR
E82E FB         1661          STI             ; INTERRUPTS BACK ON 允许中断
E82F 1E         1662          PUSH DS         ; SAVE CURRENT DS
E830 53         1663          PUSH BX         ; SAVE BX TEMPORARILY
E831 B84000     1664          MOV  BX, DATA ;
```

```
E834 8EDB       1665        MOV  DS, BX              ; ESTABLISH POINTER
                                                       TO DATA REGION
                                                       DS 指向数据段
E836 0AE4       1666        OR   AH, AH              ; AH = 0
E838 740B       1667        JZ   K1                  ; ASCII_READ
                                                       (AH) = 0 去 K1，读
                                                       ASCII 字符
E83A FECC       1668        DEC  AH                  ; AH = 1
E83C 7420       1669        JZ   K2                  ; ASCII_STATUS
                                                       (AH) = 1去K2，读状态
E83E FECC       1670        DEC  AH                  ; AH = 2
E840 742D       1671        JZ   K3                  ; SHIFT_STATUS
                                                       (AH) = 3 去 K3，取移
                                                       位状态
E842 5B         1672        POP  BX                  ; RECOVER REGISTER
                                                       恢复 BX，DS
E843 IF         1673        POP  DS
E844 CF         1674        IRET                     ; INVALID COMMAND
                                                       返回
                1675
                1676  ; ……READ THE KEY TO FIGURE OUT WHAT TO DO
                            读 ASCII 字符
                1677
E845            1678  K1:                            ; ASCII READ
E845 FB         1679        STI                      ; INTERRUPTS BACK
                                                       ON DURING LOOP
                                                       允许中断
E846 90         1680        NOP                      ; ALLOW AN INTER-
                                                       RUPT TO OCCUR
                                                       空操作，等待中断（键
                                                       盘）
E847 FA         1681        CLI                      ; INTERRUPTS BACK
                                                       OFF 关中
E848 8B1E1A00 R 1682        MOV  BX,BUFFER_HEAD      ; GET POINTER TO
                                                       HEAD OF BUFFER
E84C 3B1E1C00 R 1683        CMP  BX, BUFFER_TAIL     ; TEST END OF BUF-
                                                       FER 缓存区首尾指针
                                                       相等说明缓存区空
E850 74F3       1684        JZ   K1                  ; LOOP UNTIL SOME-
```

```
                                                    THING IN BUFFER
                                                    循环,等待按键,发生键
                                                    盘中断
E852 8B07       1685        MOV  AX, [BX]           ; GET SCAN CODE
                                                    AND ASCII CODE
                                                    ASCII 码,扫描码送AX
E854 E81E00     1686        CALL K4                 ; MOVE POINTER TO
                                                    NEXT POSITION 缓
                                                    存区指针下移一格
E857 891E1A00 R 1687        MOV  BUFFER_HEAD,BX     ; STORE VALUE IN
                                                    VARIABLE 重置缓存
                                                    区首指针
E85B 5B         1688        POP  BX                 ; RECOVER REGISTER
                                                    恢复 BX, DS
E85C 1F         1689        POP  DS                 ; RECOVER SEGMENT
E85D CF         1690        IRET                    ; RETURN TO CALLER
                                                    返回
                1691
                1692  ; ……ASCII STATUS  读 ASCII 状态
                1693
E85E            1694  X2:
E85E FA         1695        CLI                     ; INTERRUPTS OFF
                                                    关中
E85F 8B1E1A00 R 1696        MOV  BX,BUFFER_HEAD     ; GET HEAD POINTER
E863 3B1E1C00 R 1697        CMP  BX, BUFFER_TAIL    ; IF EQUAL(Z=1) TH-
                                                    EN NOTHING THERE
                                                    缓存区首尾指针重合说
                                                    明无字符输入
E867 8B07       1698        MOV  AX, [BX]           不管有没有字符先取出送 AX
E869 FB         1699        STI                     ; INTERRUPTS BACK
                                                    ON 允许中断
E86A 5B         1700        POP  BX                 ; RECOVER REGISTER
E86B 1F         1701        POP  DS                 ; RECOVER SEGMENT
                                                    恢复 BX, DS
E86C CA0200     1702        RET  2                  ; THROW AWAY FL-
                                                    AGS 扔掉标志位返回
                1703
                1704  ; ……SHIFT STATUS  读取移位状态
                1705
```

```
E86F                    1706  K3:
E86F A01700 R           1707        MOV  AL, KB_FLAG    ; GET THE SHIFT STATUS
                                                          FLAGS  移位状态内容从
                                                          KB_FLAG 指出的单元送
                                                          AL
E872 5B                 1708        POP  BX             ; RECOVER REGISTER
E873 1F                 1709        POP  DS             ; RECOVER REGISTERS
                                                          恢复 BX, DS
E874 CF                 1710        IRET                ; RETURN TO CALLER
                                                          返回
                        1711  KEYBOARD_IO  ENDP   KEYBOARD_IO 程序(软中断 16)
                                                    结束
                        1712
                        1713  ; ……INCREMENT A BUFFER POINTER  下面 6 条指令将
                              缓存区指针往下移一个字
                        1714
E875                    1715  K4    PROC NEAR
E875 83C302             1716        ADD  BX, 2          ; MOVE TO NEXT WORD
                                                          IN LIST  (BX) 加 2, 指向
                                                          下一字
E878 81FB3E00 R         1717        CMP  BX, OFFSET KB_BUFFER_END ; AT END
                                                                    OF BUF-
                                                                    FER?
                                                                    BX 正指
                                                                    向缓存区
                                                                    尾?
E87C 7503               1718        JNE  K5             ; NO, CONTINUE  否, 去 K5
E87E BB1E00 R           1719        MOV  BX, OFFSET KB_BUFFER ; YES,  RESET
                                                                TO  BUFFER
                                                                BEGINNING
                                                                是, BX 送入缓
                                                                存区首址
E881                    1720  K5:
E881 C3                 1721        RET                       返回
                        1722  K4    ENDP
                        1723
                        1724  ; ……TABLE OF SHIFT KEYS AND MASK VALUES
                              下面是键盘中断处理程序的数据区
                        1725
```

```
E882                        1726  K6    LABEL  BYTE  移位键值表
E882 52                     1727        DB     INS_KEY ; INSERT KEY  INS
                                                      INS_KEY = 82
E883 3A4546381D             1728        DB     CAPS_KEY, NUM_KEY, SCR-
                                               OLL_KEY, ALT_KEY, CTL_
                                               KEY  CAPS_KEY = 58, NUM_
                                                    KEY = 69, SCROLL_KEY
                                                    = 70, ACT_KEY = 56
E888 2A36                   1729        DB     LEFT_KEY, RIGHT_KEY
                                               CTL_KEY = 29,  LEFT_KEY =
                                               42, RIGHT_KEY = 54
0008                        1730  K6L   EQU    $_K6
                            1731
                            1732  ; ……SHIFT_MASK_TABLE  移位移屏蔽值表
                            1733
E88A                        1734  K7    LABEL  BYTE
E88A 80                     1735        DB     INS_SHIFT ; INSERT MODE
                                                      SHIFT INS_SHIFT
                                                      (插入时移位) =
                                                      80H
E88B 4020100804             1736        DB     CAPS_SHIFT,  NUM_SHIFT,
                                               SCROLL_SHIFT, ALT_SHIFT,
                                               CTL_SHIFT   CAPS_SHIFT =
                                               40H, NUM_SHIFT = 20H, SCR-
                                               OLL_SHIFT = 10H
E890 0201                   1737        DB     LEFT_SHIFT, RIGHT_SHIFT
                                               ALT_SHIFT = 08H, CTL_SHIFT
                                               = 04H, LEFT_SHIFT = 02H,
                                               RIGHT_SHIFT = 01H
                            1738
                            1739  ; ……SCAN CODE TABLES 值<59 的扫描码表
                            1740
E892 1BFF00FFFFFF1EFF       1741  K8          DB 27, -1, 0, -1, -1, -1,
                                                  30, -1
E89A FFFFFF1FFF7FFF11       1742              DB -1, -1, -1, 31, -1, 127,
                                                  -1, 17
E8A2 170512141915090F       1743              DB 23, 5, 18, 20, 25, 21, 9,
                                                  15
```

```
E8AA 101B1D0AFF0113          1744              DB 16, 27, 29, 10, -1, 1, 19
E8B1 040607080A0B0CFFFF      1745              DB 4, 6, 7,8,10, 11, 12,-1,-1
E8BA FFFF1C1A18031602        1746              DB -1, -1, 28, 26, 24, 3, 22,2
E8C2 0E0DFFFFFFFFFFFF        1747              DB 14, 13, -1, -1, -1, -1,
                                                   -1, -1
E8CA 20FF                    1748              DB '', -1
                             1749  ; ……CTL TABLE SCAN  CTL 表
E8CC                         1750  K9   LABEL BYTE
E8CC 5E5F606162636465        1751              DB 94, 95, 96, 97, 98, 99, 100,
                                                   101  94～103; CTL F1～CT-
                                                   LF10
E8D4 6667FFFF77FF84FF        1752              DB 102, 103, -1, -1, 119,
                                                   -1, 132, -1
E8DC 73FF74FF75FF76FF        1753              DB 115, -1, 116, -1, 117, -1,
                                                   118, -1
E8E4 FF                      1754              DB -1
                             1755  ; ……LC TABLE  小写字符表
E8E5                         1756  K10  LABEL BYTE
E8E5 1B                      1757              DB 01BH,  '1234567890- =',
                                                   08H, 09H
E8E6 31323334353637
     3839302D3D0809
E8F4 71776572747975          1758              DB 'qwertyuiop[]', 0DH, -1,
     696F705B5D0DFF                                'asdfghjkl; ', 027H
E902 6173646667686A
     6B6C3B27
E90D 60FF5C                  1759              DB 60H, -1, 5CH, 'zxcvbnm,
                                                   /', -1, '*', -1, ''
E910 7A786376626E6D
     2C2E2FFF2AFF20
E91E FF                      1760              DB -1
                             1761
                             1762  ; ……UC TABLE
E91F                         1763  K11  LABEL BYTE  大写字符表
E91F 1B                      1764              DB 27, '!@# $',37,05EH, '&*( )
                                                   _+', 08H, 0
E920 21402324255E
E926 262A28295F2B0800
E92E 51574552545955          1765              DB 'QWERTYUIOP', 0DH, -1,
```

```
                                        'ASDFGHJKL;"'
        494F507B7D0DFF
E93C    4153444647484A
        4B4C3A22
E947    7EFF                    1766            DB 07EH, -1,'ZXCVBNM< >?',
                                                    -1, 0, -1,'', -1
E949    7C5A584356424E
        4D3C3E3FFF00FF20FF
                                1767    ; ……UC TASLE SCAN  大写字符扫描码表(大
                                          写字母F1__F10键)
E959                            1768    K12     LABEL   BYTE
E959    5455565758595A          1769            DB 84, 85, 86, 87, 88, 89, 90
E960    5B5C5D                  1770            DB 91, 92, 93
                                1771    ; ……ALT TABLE SCAN  ACT表
E963                            1772    K13     LABEL   BYTE
E963    68696A6B6C              1773            DB 104, 105, 106, 107, 108
E968    6D6E6F7071              1774            DB 109, 110, 111, 112, 113
                                1775
                                1776    ; ……NUM STATE TABLE  NUM表
E96D                            1777    K14     LABEL   BYTE
E96D    3738392D343536          1778            DB '789-456+1230'
        2B313233302E
                                1779    ; ……BASE CASE TABLE  特别字符表(键号71
                                          ~83)
E97A                            1780    K15     LABEL   BYTE
E97A    474849FF4BFF4D          1781            DB 71, 72, 73, -1, 75, -1, 77
E981    FF4F50515253            1782            DB -1, 79, 80, 81, 82, 83
                                1783
                                1784    ; ……KEYBOARD INTERRUPT ROUTINE
                                                键盘中断处理程序
                                1785
E987                            1786    KB_INT  PROC    FAR
E987    FB                      1787            STI     ; ALLOW FURTHER INT-
                                                          ERRUPTS  允许中断
E988    50                      1788            PUSH    AX
E989    53                      1789            PUSH    BX
E98A    51                      1790            PUSH    CX
E98B    52                      1791            PUSH    DX
E98C    56                      1792            PUSH    SI
E98D    57                      1793            PUSH    DI
```

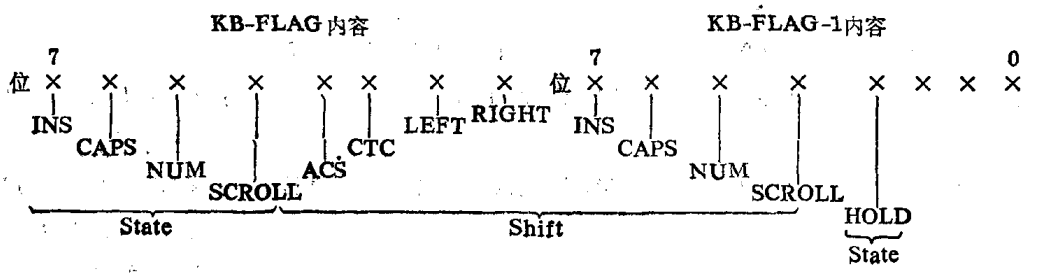
```
E98E 1E          1794          PUSH DS
E98F 06          1795          PUSH ES  保存AX, BX, CX, DX, SI, DI, DS和ES
E990 FC          1796          CLD                    ; FORWARD DIRECTION 前
                                                        向标志
E991 B84000 R    1797          MOV  AX, DATA
E994 8ED8        1798          MOU  DS, AX            ; SET UP ADDRESSING DS
                                                        指向数据段
E996 E460        1799          IN   AL, KB_DATA       ; READ IN THE CHARACT-
                                                        ER 读 8255PA 口(KB_DA-
                                                        TA = 60H) 取得扫描码
E998 50          1800          PUSH AX                ; SAVE IT 扫描码压栈
E999 E461        1801          IN   AL, KB_CTL        ; GET THE CONTROL PORT
                                                        读 8255 PB 口
E99B 8AE0        1802          MOV  AH, AL            ; SAVE VALUE
E99D 0C80        1803          OR   AL, 80H           ; RESET BIT FOR KEYBOARD
E99F E661        1804          OUT  KB_CTL, AL          PB 口第 7 位置 1 复位键盘
E9A1 86E0        1805          XCHG AH, AL            ; GET BACK ORIGINAL
                                                        CONTROL
E9A3 E661        1806          OUT  KB_CTL, AL        ; KB HAS BEEN RESET 恢
                                                        复 PB 口原先内容
E9A5 58          1807          POP  AX                ; RECOVER SCAN CODE
E9A6 8AE0        1808          MOV  AH, AL            ; SAVE SCAN CODE IN AH
                                                        ALSO 扫描码送 AH, AL
                 1809
                 1810 ;……TEST FOR OVERRUN SCAN CODE FROM KEYBOA-
                       RD
                 1811
E9AB 3CFF        1812          CMP  AL, 0FFH          ; IS THIS AN OVERRUN
                                                        CHAR 是溢出字符?
E9AA 7503        1813          JNZ  K16               ; NO, TEST FOR SHIFT KEY
                                                        否, 去 K16
E9AC E97502      1814          JMP  K62               ; BUFFER_FULL_BEEP
                                                        本次中断由键盘缓存区溢出引
                                                        起, 去 K62 进行处理
                 1815
                 1816 ;……TEST FOR SHIFT KEYS  检查是否为移位键
                 1817
E9AF             1818 K16:                           ; TEST_SHIFT
```

```
E9AF 247F        1819          AND     AL, 07FH          ; TURN OFF THE BREAK
                                                           BIT  扔掉 BREAK 位
E9B1 0E          1820          PUSH    CS
E9B2 07          1821          POP     ES                ; ESTABLISH ADDRESS
                                                           OF SHIFT TABLE  ES
                                                           指向代码段首
E9B3 BFB2E8 R    1822          MOV     DI, OFFSET K6     ; SHIFT KEY TABLE
                                                           移位键表偏移地址送
                                                           DI
E9B6 B90800      1823          MOV     CX, K6L           ; LENGTH 表长送 CX
E9B9 F2          1824          REPNE   SCASB             ; LOOK THROUGH THE
                                                           TABLE FOR A MATCH
                                                           查表，直至查出一个匹配
                                                           (ZF=1) 或查完整个表格
                                                           (CX=0)
E9BA AE
E9BB 8AC4        1825          MOV     AL, AH            ; RECOVER SCAN CODE
                                                           扫描码送 AL
E9BD 7403        1826          JE      K17               ; JUMP IF MATCH FO-
                                                           UND  键的是移位键则转
                                                           K17
E9BF E98800      1827          JMP     K25               ; IF NO MATCH, THEN
                                                           SHIFT NOT FOUND
                                                           非移位键，转 K25 再查其
                                                           它表
                 1828
                 1829 ;·····SHIFT KEY FOUND  移位键
                 1830
E9C281EF83E8 R   1831 K17:     SUB     DI, OFFSET K6+1   ; AD1UST PTR TO SCAN
                                                           CODE HTCH  调整在移
                                                           位键表中的偏移，因SCA-
                                                           SB 中 DI 多加了 1
E9C6 2EBAA58AE8 R 1832         MOV     AH, CS:K7[DI]     ; GET MASK INTO AH
                                                           取该移位键的屏蔽值
E9CB A880        1833          TEST    AL, 80H           ; TEST FOR BREAK KEY
                                                           按移位键时处BREAK态?
E9CD 7554        1834          JNZ     K23               ; BREAK_SHIFT_ FOUND
                                                           是，去 K23
```

```
                        1835
                        1836 ;.....SHIFT MAKE FOUND, DETERMINE SET OR
                                   TOGGLE  只按移位键
                        1837
E9CF 8OFC10             1838       CMP    AH, SCROLL_SHIFT    按下的是 SCROLL
                                                              NUM, CAPS, INS
                                                              键?
E902 7307               1839       JAE    K18                 ; IF SCROLL SFI-
                                                              FY OR ABOVE, TOG-
                                                              GLE KEY 是,转 K18
                        1840
                        1841 ;.....PLAIN SHIFT KEY, SET SHIFT ON  一般移位键
                                   ACT, CTL, CEFT, RIGHT
                        1842
E9D4 08261700     R     1843       DR     KB_FLAG, AH         ; TURN ON SHIFT
                                                              BIT  KB_FLAG
                                                              中的相应位置1,表
                                                              示按入了某移位键
E9D8 E98300             1844       JMP    K26                 ; INTERRUPT_
                                                              RETURN  准备返
                                                              回
                        1845
                        1846 ;.....TOGGLED SHIFT KEY, TEST FOR 1ST MAKE
                                   OR NOT  乒乓移位键,测试其是否第一次接通
                        1847
E9D8                    1848 K18;                              ; SHIFT_TOGGLE
E9DB F606170004   R     1849       TEST   KB_FLAG, CTL_SHIFT  ; CHECK CTL
                                                              SHIFT  ST-
                                                              ATE  以前
                                                              按过 CTL 移
                                                              位键?
E9E0 7568               1850       JNZ    K25                 ; JUMP IF CTL
                                                              STATE 是,去K25
E9E2 3C52               1851       CMP    AL, INS_KEY         ; CHECK FOR IN-
                                                              SERT KEY  本次
                                                              按下 INSERT 键?
E9E4 7525               1852       JNZ    K22                 ; JUMP  IF  NOT
                                                              INSERT KEY
                                                              是,去 K22
```

```
E9E6 F606170008 R  1853          TEST KB_FLAG, ALT_SHIFT ; CHECK FOR
                                                          ALTERNATE
                                                          SHIFT 以前按
                                                          过 ALT 键?
E9EB 7403          1854          JZ   K19                ; JUMP IF NOT ALT-
                                                          ERNATE SHIFT 否,
                                                          去K19
E9ED EB5B90        1855          JMP  K25                ; JUMP IF ALTERNA-
                                                          TE SHIFT 按过 ALT
                                                          键, 即处 ALT 状态, 去
                                                          K25 处理
E9F0 F606170020 R  1856 K19:     TEST KB_FLAG, NUM_STATE ; CHECK FOR
                                                          BASE STATE
E9F5 750D          1857          JNZ  K21                ; JUMP IF NUM LOCK
                                                          IS ON 以前按过NUM
                                                          键的话去K21
E9F7 F606170003 R  1858          TEST KB_FLAG, LEFT_SHIFT+RIGHT_SHIFT
                                      以前按过 LEFT 或 RIGHT 键
E9FC 740D          1859          JZ   K22                ; JUMP IF BASE STA-
                                                          TE 否, 去K22
                   1860
E9FE               1861 K20:                             ; NUMERIC ZERO,
                                                          NOT INSERT KEY 以
                                                          前按过LEFT或RIGHT
                                                          键, 现返回 ASCII数字
                                                          0 代码
E9FE B83052        1862          MOV  AX, 5230H          ; PUT OUT AN ASCII
                                                          ZERO
EA01 E90801        1863          JMP  K57                ; BUFFER_FILL 去
                                                          K57
EA04               1864 K21:                             ; MIGHT BE NUMERIC
EA04 F606170003 R  1865          TEST KB_FLAG, LEFT_SHIFT+RIGHT_SHIFT;
                                      以前按下 NUM 键, 但未按 LETT 或 RIGNT
                                      键, 返回 0。
EA09 74F3          1866          JZ   K20                ; JUMP NUMERIC, N-
                                                          OT INSERT
                   1867
EA0B               1868,K22:                             ; SHIFT TOGGLE KEY
                                                          HIT;PRODESS IT 处理
```

```
                                                    乒乓移位键
EA0B 84261800  R 1869          TEST AH, KB_FLAG_1   ; IS KEY ALREADY
                                                      DEPRESSED 以前按
                                                      过该键?
EA0F 754D       1870           JNZ  K26             ; JUMP IF KEY ALR-
                                                      EADY DEPRESSED
                                                      按过的话不作标记
EA11 08261800  R 1871          OR   KB_FLAG_1, AH   ; INDICATE THAT
                                                      THE KEY IS DEPRE-
                                                      SSED 标记已按下了
                                                      该移位键
EA15 30261700  R 1872          XOR  KB_FLAG, AH     ; TOGGLE THE SHIFT
                                                      STATE 清 KB_FLAG
                                                      的相应位，去除该键的
                                                      作用
EA19 3C52       1873           CMP  AL, INS_KEY     ; TEST FOR IST MAKE
                                                      OF INSERT KEY 本
                                                      次按下的是INSERT键?
EA1B 7541       1874           JNE  K26             ; JUMP IF NOT INSE-
                                                      RT KEY 否，去 K26
```



State：记录当前乒乓移位键的状态
Shift：为1表示相应的移位键按下过

```
EA1D B80052     1875           MOV  AX, INS_KEY*256 ; SET SCAN CODE IN
                                                      TO AH, 0 INTO AL
                                                      INSERT 键的扩充扫描
                                                      码送 AH, 0 送 AL
EA20 E98901     1876           JMP  K57             ; PUT INTO OUTPUT
                                                      BUFFER
                1877
                1878 ;.....BREAK SHIFT FOUND 处理 BREAK 态下的移位键
                1879
EA23            1880 K23:                           ; BREAK_SHIFT_FO-
```

```
                                                    UND
EA23 80FC10          1881       CMP  AH,SCROLL_SHIFT ; IS THIS A TOGGLE
                                                    KEY
EA26 731A            1882       JAE  K24            ; YES, HANDLE BRE-
                                                    AK TOGGLE 按下乒
                                                    乓移位键的话转K24
EA28 F6D4            1883       NOT  AH             ; INVERT MASK
EA2A 20261700 R      1884       AND  KB_FLAG, AH    ; TURN OFF SHIFT B
                                                    IT KB_FLAG相应位
                                                    复位
EA2E 3CB8            1885       CMP  AL, ALT_KEY+80H ; IS THIS ALTERNA-
                                                    TE SHIFT RELEASE
                                                    按下ALT乒乓开关(加
                                                    80H以恢复最高位)
EA30 752C            1886       JNE  K26            ; INTERRUPT_RETU-
                                                    RN 否转K26,返回.
                     1887
                     1888 ;.....ALTERNATE SHIFT KEY RELEASED, GET THE
                                VALUE INTO BUFFER BREAK态下的ALT键即释
                                放ALT键
                     1889
EA32 A01900 R        1890       MOV  AL, ALT_INPUT
EA35 B400            1891       MOV  AH, 0          ; SCAN CODE OF 0 0
                                                    扫描码送AH
EA37 88261900 R      1892       MOV  ALT_INPUT, AH  ; ZERO OUT THE FI-
                                                    ELD 0送ALT_INP-
                                                    UT单元
EA3B 3C00            1893       CMP  AL, 0          ; WAS THE INPUT=0
                                                    输入过数据?
EA3D 741F            1894       JE   K26            ; INTERRUPT_RETU-
                                                    RN (ACT_INPUT)
                                                    =0表示未输入过
EA3F E9A301          1895       JMP  K58            ; IT WASN'T, SO PUT
                                                    IN BUFFER 输入过,
                                                    转K58
                     1896
EA42                 1897 K24;                      ; BREAK_TOGGLE
                                                    BREAK 态下的乒乓移
                                                    位键
```

```
EA42 F6D4          1898          NOT  AH                  ; INVERT MASK
EA44 20261800 R    1899          AND  KB_FLAG_1,AH        ; INDDICATE NO LO-
                                                              NGER DEPRESSED
                                                              KB_FLAG_1 相应位
                                                              复位
EA48 EB14          1900          JMP  SHORT K26           ; INTERRUPT_RETU-
                                                              RN 准备返回
                   1901
                   1902  ; ……TEST FOR HOLD STATE 检查 HOLD (保持)态
                                 (在 CTL 态下按下移位键 (1850 行转来)或
                   1903  非移位键(第 1827 行转来))
EA4A               1904  K25:                           ; NO_SHIFT_FOUND
EA4A 3C80          1905          CMP  AL,80H              ; TEST FOR BREAK
                                                              KEY
EA4C 7310          1906          JAE  K26                 ; NOTHING FOR BRE-
                                                              AK CHARS FROM
                                                              NERE ON 处BREAK
                                                              态则立即返回
EA4E F606180008 R  1907          TEST KB_FLAG_1,HOLD_STATE ; ARE WE
                                                              IN HOLD
                                                              STATE
                                                              现处HOL
                                                              D 态?
EA53 7417          1908          JZ   K28                 ; BRANCH AROUND
                                                              TEST IF NOT 否,去
                                                              K28,再查
EA55 3C45          1909          CMP  AL,NUM_KEY
EA57 7405          1910          JE   K26                 ; CAN'T END HOLD
                                                              ON NUM_LOCK 现
                                                              处 HOLD 态,同时又处
                                                              NUM 态的话转 K26
EA59 80261800F7 R  1911          AND  KB_FLAG_1,NOT HOLD_STATE
                                 ; TURN OFF THE HOLD STATE BIT 清
                                   HOLD 态
                   1912
EA5E               1913  K26:                           ; INTERRUPT_RETU-
                                                              RN
EA5E FA            1914          CLI                      ; TURN OFF INTERRU
                                                              PTS 关中
```

```
EA5F B020          1915          MOV  AL, EOI   ; END OF INTERRUPT COM-
                                                  MAND  向 8259A 送中断结束
                                                  命令
EA61 E620          1916          OUT  020H, AL  ; SEND COMMAND TO INTE-
                                                  RRUPT CONTROL PORT
EA63               1917 K27:                    ; INTERRUPT—RETURN_NO-
                                                  EOI
EA63 07            1918          POP  ES
EA64 1F            1919          POP  DS
EA65 5F            1920          POP  DI
EA66 5E            1921          POP  SI
EA67 5A            1922          POP  DX
EA68 59            1923          POP  CX
EA69 5B            1924          POP  BX
EA6A 5B            1925          POP  AX        ; RESTORE STATE  恢复 ES,
                                                  DS, DI, SI, DX, CX, BX, AX
EA6B CF            1926          IRET           ; RETURN, INTERRUPTS BA-
                                                  CK ON WITH FLAG CHANGE
                                                  标志位不变的返回
                   1927
                   1928 ;.....NOT IN HOLD STATE, TEST FOR SPECIAL C-
                                 HARS  非 HOLD 态检查其他键
                   1929
EA6C               1930 K28:                    ; NO_HOLD—STATE
EA6C F606170008 R  1931          TEST KB—FLAG, ALT_SHIFT
                                                  ; ARE WE IN ALTERNATE S-
                                                  HIFT  正处 ALT 态?
EA71 7503          1932          JNZ  K29       ; JUMP IF ALTERNATE SNIFT
                                                  是,转 K29
EA73,E98F00        1933          JMP  K38       ; JUMP IF NOT ALTERNATE
                                                  非 ALT 态转 K38
                   1934
                   1935 ; .....TEST FOR RESET KEY SEQUENCE (CTL ALT
                                 DEL)  测试复位键(或同时按下 CTL, ALT 和 DEL
                                 键)
                   1936
EA76               1937 K29:                    ; TEST—RESET
EA76 F606170004 R  1938          TEST KB—FLAG, CTL—SHIFT
                                                  ; ARE WE IN CONTROL SHIFT
```

```
                                                    ALSO 正处CTL态?
EA7B 7431              1939         JZ    K31        ; NO_RESET否,去K31
EA7D 3C53              1940         CMP   AL,DEL—KEY SHIFT STATE IS
                                                    THERE, TEST KEY
                                                    正处DEL态
EA7F 752D              1941         JNE   K31        ; NO—RESET 否,转
                                                    K31
                       1942
                       1943 ;……CTL—ALT—DEL HAS BEEN FOUND, DO I/O
                            CLEANUP 处CTL, ALT, DEL态复位系统
                       1944
EA81 C70672003412  R   1945         MOV   RESET—FLAG, SET FLAG FOR RE-
                                          1234H      SET FUNCTION 设
                                                    置复位标记
EA87 E9D1F5            1946         JMP   RESET      ; JUMP TO POWER ON
                                                    DIAGNOSTICS 去复
                                                    位程序即从第266行开
                                                    始的15个测试程序
                       1947
                       1948 ;……ALT—INPUT—TABLE 利用ALT键输入ASC
                            II码时的数字表,定义的数表示键号,
EA8A                   1949 K30     LABEL BYTE
EA8A 524F50514B4C4D    1950         DB    82, 79, 80, 81, 75, 76, 77
LOC OBJ                LINE  SOURCE
EA91 474849            1951         DB    71, 72, 73   ; 10 NUMBERS ON K-
                                                    EYPAD
                       1952 ;……SUPER—SHIFT—TABLE A-Z 26个英文字母
                            表
EA94 1011121314151617  1953         DB    16, 17, 18, 19, 20, 21, 22, 23
                                                    ; A-Z TYPEWRITER
                                                    CHARS
EA9C 18191E1F20212223  1954         DB    24, 25, 30, 31, 32, 33, 34, 35
EAA4 2425262C2D2E2F30  1955         DB    36, 37, 38, 44, 45, 46, 47, 48
EAAC 3132              1956         DB    49, 50
                       1957
                       1958 ;……IN ALTERNATE SHIFT, RESET NOT FOUND
                            处ALT态,且非复位要求
                       1959
EAAE                   1960 K31:                     ; NO_RESET
EAAE 3C39              1961         CMP   AL, 57     ; TEST FOR SPACE
```

```
                                                    KEY 是空格键?
EAB0 7505          1962      JNE     K32            ; NOT THERE
EAB2 B020          1063      MOV     AL,''          ; SET SPACE CHAR
                                                    是，空格的ASCII码
                                                    '20'送AL
EAB4 E92501        1964      JMP     K57            ; BUFFER—FILL 转
                                                    K57 填入缓存区
                   1965
                   1966 ;.....LOOK FOR KEY PAD ENTRY
                   1967
EAB7               1968 K32:                        ; ALT—KEY_PAD
EAB7 BF8AEA R      1969      MOV     DI, OFFSET K30 ; ALT—INPUT_TABLE
                                                    取 ALT_INPUT
                                                    _TABLE 首址
EABA B90A00        1070      MOV     CX, 10         ; LOOK FOR ENTRY
                                                    USING KEYPAD 只
                                                    查10个数字
EABD F2            1971      REPNE   SCASB          ; LOOK FOR MATCH
                                                    查寻
EABEAE
EABF 7512          1972      JNE     K33            ; NO_ALT_KEYPAD
                                                    ALT 态下按入的不是
                                                    数字键,转K33
EAC1 81EF8BEA R    1973      SUB     DI, OFFSET K30+1
                                                    ; DI NOW HAS ENTRY
                                                    VALUE 调整指针
EAC5 A01900 R      1974      MOV     AL, ALT_INPUT  ; GET THE CURRENT
                                                    BYTE 取当前字节
EAC8 B40A          1975      MOV     AH, 10         ; MULTIPLY BY 10
EACA F6E4          1976      MUL     AH             当前字节*10以留下
                                                    存放当前输入数字的空
                                                    间
EACC 03C7          1977      ADD     AX, DI         ; ADD IN THE LATEST
                                                    ENTRY 送当前数字,
                                                    存入ALT—INPUT单
                                                    元
EACE A21900 R      1978      MOV     ALT_INPUT,AL   ; STORE IT AWAY
EAD1 EB8B          1979      JMP     K26            ; THROW AWAY TH-
                                                    AT KEYSTROKE 准备
```

返回

```
                        1980
                        1981 ;……LOOK FOR SUPERSHIFT ENTRY
                        1982
EAD3                    1983 K33:                          ; NO—ALT__KEYPAD
EAD3 C606190000 R       1984      MOV   ALT__INPUT, 0      ; ZERO ANY PREVIOUS
                                                             ENTRY INTO INPUT
                                                             非 ALT 态下输入 ASCII
                                                             码,置标态
EAD8 B91A00             1985      MOV   CX, 26             ; DI, ES ALREADY PO-
                                                             INTING 计数器
EADB F2                 1986      REPNE SCASB              ; LOOK FOR MATCH IN
                                                             ALPHABET 查 A-Z 26
                                                             个英文字母表
EADC AE
EADD 7505               1987      JNE   K34                ; NOT FOUND, FUNCTI-
                                                             ON KEY OR OTHER
                                                             未查到转K34
EADF B000               1988      MOV   AL, 0              ; ASCII CODE OF ZERO
                                                             查到, AL 送 0. 表示是
                                                             ALT 态下的英文字母(扩
                                                             充 AS CII 码)
EAE1 E9F800             1989      JMP   K57                ; PUT IT IN THE BUFF-
                                                             ER 转 K57, 填入缓存区
                        1990
                        1991 ;……LOOK FOR TOP ROW OF ALTERNATE SHIFT 查
                                  上排键
                        1992
EAE4                    1993 K34:                          ; ALT__TOP__ROW
EAE4 3C02               1994      CMP   AL, 2              ; KEY WITH '1' ON IT.
                                                             按入的是第 2 至 14 号键?
                                                             不是的话转 K35
EAE6 720C               1995      JB    K35                ; NOT ONE OF INTERE-
                                                             STING KEYS
EAE8 3C0E               1996      CMP   AL, 14             ; IS IT IN THE REGION
EAEA 7308               1997      JAE   K35                ; ALT__FUNCTION
EAEC 80C476             1998      ADD   AH, 118            ; CONVERT PSUEDO SC-
                                                             AN CODE TO RANGE
                                                             按入 2～14 号键之一, 键号加
```

```
                                                    118 成扩充 ASCII 码
EAEF B000       1999        MOV  AL, 0          ; INDICATE AS SUCH  AL
                                                    送 0，表示扩充 ASCII 码
EAF1 E9E800     2000        JMP  K57            ; BUFFER_FILL  转 K57 填缓
                                                    存区
                2001
                2002 ;……TRANSLATE ALTERNATE SHIFT PSEUDO SCAN
                        CODES
                2003
EAF4            2004 K35:                       ; ALT_FOUCTICN
EAF4 3C3B       2005        CMP  AL, 59         ; TEST FOR IN TABLE
EAF6 7303       2006        JAE  K37            ; ALT_CONTINUE
EAFB            2007 K36:                       ; CLDSE_RETURN
EAF8 E963FF     2008        JMP  K26            ; IGNORE THE KEY
EAFB            2009 K37:                       ; ALT_CONTINUE
EAFB 3C47       2010        CMP  AL, 71         ; IN KEYPAD REGION  按入
                                                    的是第 59～71 号键？(实际
                                                    有意义的是 59_68 号功能
                                                    键)
EAFD 73F9       2011        JAE  K36            ; IF SO, IGNORE
EAFF BB63E9 R   2012        MOV  BX,OFFSET K13  ; ALT SHIFT PSEUDO SCAN
                                                    TABLE  是，取 ALT 功能键
                                                    表首址
EB02 E92501     2013        JMP  K63            ; TRANSLATE THAT  转 K63
                                                    查寻
                2014
                2015 ;……NOT IN ALTERNATE SHIFT  非 ALT 态
                2016
EB05            2017 K38:                       ; NOT_ALT_SHIFT
EB05 F606170004 R 2018      TEST KB_FLAG, CTL_SHIFT ; ARE WE IN CO-
                                                    NTROL SHIFT  当
                                                    前处 CTL 控制态？
EB0A 745B       2019        JZ   K44            ; NOT_CTL_SHIFT  否，转
                                                    K44
                2020
                2021 ;……CONTROL SHIFT, TEST SPECIAL CHARACTERS
                        CTL态，检查特殊字符
                2022 ;……TEST FOR BREAK AND PAUSE KEYS
                2023
```

```
EB0C 3C46          2024        CMP  AL, SCROLL_KEY   ; TEST FOR BREAK
                                                     按入SCROLL键?
EB0E 7518          2025        JNE  K39             ; NO_BREAK  否，转
                                                     K39
EB10 BB1E00     R  2026        MOV  BX, OFFSET KB_BUFFER
                                                     ; RESET BUFFER TO
                                                     EMPTY  键盘缓存首
                                                     尾指针重合，清缓存区
EB13 891E1A00   R  2027        MOV  BUFFER_HEAD,BX  ;
EB17 891E1C00   R  2028        MOV  BUFFER_TAIL, BX ;
EB1B C606710080 R  2029        MOV  BIOS_BREAK, 80H ; TURN ON BIOS_BR-
                                                     EAK BIT  置 BIOS_
                                                     BREAK 位
EB20 CD1B          2030        INT  1BH             ; BREAK INTERRUPT
                                                     VECTOR  调软中断
                                                     1BH，开机程序把他短
                                                     路了，无动作，用户可自
EB22 B80000        2031        MOV  AX, 0           ; PUT OUT DUMMY
                                                     CHARACTER
                                                     编该处理程序完成某种
                                                     功能
EB25 E9B400        2032        JMP  K57             ; BUFFER_FILL  往缓
                                                     存区送哑字符
                   2033
EB28               2034 K39:                        ; NO_BREAK
EB28 3C45          2035        CMP AL, NUM_KEY      ; LOOK FOR PAUSE
                                                     KEY  按入的是 NUM
                                                     LOCK 键?
EB2A 7521          2036        JNE  K41             ; NO_PAUSE  否，去
                                                     K41
EB2C 800E180008 R  2037        OR   KB_FLAG_1, HOLD_STATE
                                                     ; TURN ON THE HO-
                                                     LD FLAG  是，置HO-
                                                     LD 标记CTRL，NUM连
                                                     用能使系统进入循环
EB31 B020          2038        MOV  AL, EOI         ; END OF INTERRUPT
                                                     TO CONTROL PORT
                                                     等待，按入非 CTRL，
```

```
                                                    NUM 键使暂停结束
EB33 E620          2039      OUT   020H, AL       ; ALLOW FURTHER KEY-
                                                    STROKE INTS  写中断结
                                                    束命令 EOI
                   2040
                   2041 ;……DURING PAUSE INTERVAL, TURN CRT BACK ON
                   2042
EB35 803E490007  R 2043      CMP   CRT_MODE, 7   ; IS THIS BLACK AND
                                                    WHITE CARD  系统连接
                                                    的是单色显示器?
EB3A 7407          2044      JE    K40           ; YES, NOTHING TO DO
EB3C BAD803        2045      MOV   DX, 03D8H     ; PORT FOR COLOR CARD
                                                    否,彩色 CRT 模式选择寄存
                                                    器地址送 DX(3D8H)
EB3F AD6500      R 2046      MOV   AL, CRT_MODE_SET
                                                  ; GET THE VALUE OF THE
                                                    CURRENT MODE  取当前
                                                    CRT 模式字节
EB42 EE            2047      OUT   DX, AL        ; SET THE CRT MODE,
                                                    SO THAT CRT IS ONM
                                                    送模式字节
EB43               2048 K40:                     ; PAUSE_LOOP  下面为暂
                                                    停循环
EB43 F606180008  R 2049      TEST  KB_FLAG_1, HOLD_STATE
EB48 75F9          2050      JNZ   K40           ; LOOP UNTIL FLAG TU-
                                                    RNED OFF  若未退出HO-
                                                    LD 态,则继续等待
EB4A E916FF        2051      JMP   K27           ; INTERRUPT—RETURN —
                                                    NO_EOI
EB4D               2052 K41:                     ; NO_PAUSE  退出 HOLD
                                                    态,返回
                   2053
                   2054 ;……TEST SPECIAL CASE KEY 55
                   2055
EB4D 3C37          2056      CMP   AL, 55   按入 PrtSc 打印屏幕键?
EB4F 7506          2057      JNE   K42           ; NOT_KEY_55
EB51 B80072        2058      MOV   AX, 114*256   ; START/STOP PRINTING
                                                    SWITCH  是, 送扩充扫描
                                                    码(启/停打印开关)
```

```
EB54 E98500          2059        JMP  K57              ; BUFFER_FILL 填缓存区
                     2060
                     2061 ;……SET UP TO TRANSLATE CONTROL SHIFT
                     2062
EB57                 2063 K42:                         ; NOT_KEY_55
EB57 BB92E8      R   2064        MOV  BX, OFFSET K8    ; SET UP TO TRANSLATE
                                                          CTL
EB5A 3C3B            2065        CMP  AL, 59           ; IS IT IN TABLE 按入的
                                                          键号≥59 的话转 K43
EB5C 7303            2066        JAE  K43              ; CTL_TABLE _ TRANSL-
                                                          ATE
EB5E EB7890          2067        JMP  K56              ; YES, GO TRANSLATE
                                                          CHAR 去K56 查 K8 表
EB61                 2068 K43:                         ; CTL—TABLE — TRANSL-
                                                          ATE
EB61 BBCCE8      R   2069        MOV  BX, OFFSET K9    ; CTL TABLE SCAN CTL
                                                          表始址送 BX
EB64 E9C300          2070        JMF  K63              ; TRANSLATE_SCAN 去
                                                          K63 查 K9 表
                     2071
                     2072 ; ……NOT IN CONTROL SHIFT 非 CTRL 态
                     2073
EB67                 2074 K44:                         ; NUT_CTL_SHIFT
                     2075
EB67 3C47            2076        CMP  AL, 71           ; TEST FOR KEYPAD RE-
                                                          GION 按入的是HOME(第
                                                          71 号) 键或键号大于 71 的
                                                          键?
EB69 732D            2077        JAE  K48              ; HANDLE KEYPAD REG-
                                                          ION
EB6B F606170003  R   2078        TEST KB_FLAG, LEFT_SHIFT + RIGHT_SHIFT
                                                          是的话转 K48
EB70 745B            2079        JZ   K54              ; TEST FOR SHIFT STATE
                                                          否,测试当前处 LEFT 或 RI
                                                          GHT ⇑键)态否
                     2080
                     2081 ;……UPPER CASE, HANDLE SPECIAL CASES 正处
                                                        LEFT 或 RIGHT 态(大写字母态)
                     2082
```

```
EB72 3C0F       2083        CMP  AL, 15          ; BACK TAB KEY  按下的是S←
                                                 →S键?
EB74 7506       2084        JNE  K45            ; NOT BACK TAB  否, 转K45
EB76 B8000F     2085        MOV  AX, 15*256     ; SET PSEUDO SCAN CODE 是,送扩充 ASCII 码(15)及标志
EB79 EB6190     2086        JMP  K57            ; BUFFER_FILL
                2087
EB7C            2088 K45:                       ; NOT_BACK_TAB  非 S←→S 键
EB7C 3C37       2089        CMP  AL, 55         ; PRINT SCREEN KEY  按入的是 PrtSc 键?
EB7E 7509       2090        JNE  K46            ; NOT_PRINT_SCREEN 否,转 K46
                2091
                2092 ;……ISSUE INTERRUPT TO INDICATE PRINT SCREEN FUNCTION ⇑ 键和 PrtSc 键混合使用产生打印屏幕动作
                2093
EB80 B020       2004        MOV  AL, EDI        ; END OF CURRENT INTERRUPT
EB82 E620       2095        OUT  020H, AL       ; SO FURTHER THINGS CAN HAPPEN  发结束中断命令
EB84 CD05       2096        INT  5H             ; ISSUE PRINT SCREEN INTERRUPT  调软中断 5H, 打印屏幕
EB86 E9DAFE     2097        JMP  K27            ; GO BACK WITHOUT EOI OCCURRING  准备返回
                2098
EB89            2099 K46:                       ; NOT_PRINT_SCREEN
EB89 3C3B       2100        CMP  AL, 59         ; FUNCTION KEYS  按入的是键号等于或大于 59 的键即 F1~F10 十个功能键?
EB8B 7206       2101        JB   K47            ; NOT_UPPER_FUNCTION 否,转 K47
EB8D BB59E9   R 2102        MOV  BX,OFFSET K12  ; UPPER CASE PSEUDO SCAN CODES  送大写字母扫描码表首址
```

```
LOC OBJ              LINE  SOURCE
EB90 E99700          2103        JMP  K63              ; TRANSLATE_SCAN 取
                                                        ASCII 码
                     2104
EB93                 2105 K47:                         ; NOT_UPPER_FUNCTION
                                                        非大写字母功能键
EB93 BB1FE9      R   2106        MOV  BX,OFFSET K11    ; POINT TO UPPER CASE
                                                        TABLE 取大写字符表首址
EB96 EB40            2107        JMP  SHORT K56        ; OK, TRANSLATE THE
                                                        CHAR 转 K56，从该字符
                                                        表查 ASCII 码
                     2108
                     2109 ;……KEYPAD KEYS, MUST TEST NUM LOCK FOR
                          DETERMINATION 处理键号等于、大于 71 的键
                     2110
EB98                 2111 K48:                         ; KEYPAD—REGION
EB98 F606170020  R  2112        TEST KB_FLAG, NUM_STATE
                                                        ; ARE WE IN NUM—LOCK
                                                        正处 NUM 态?
EB9D 7520            2113        JNZ  K52              ; TEST FOR SURE 是，转
                                                        K52
EB9F F606170003  R  2114        TEST KB_FLAG, LEET_SHIFT+RIGHT_SHIFT
                                                        ; ARE WE IN SHIFT ST-
                                                        ATE 正处移位态?
EBA4 7520            2115        JNZ  K53              ; IF SHIFTED, REALLY
                                                        NUM STATE 是，转 K53
                     2116
                     2117 ;……BASE CASE FOR KEYPAD 特殊字符
                     2118
EBA6                 2119 K49:                         ; BASE—CASE
                     2120                              ;
EBA6 3C4A            2121        CMP  AL, 74           ; SPECIAL CASE FOR A
                                                        COUPLE OF KEYS 是减
                                                        号键?
EBA8 740B            2122        JE   K50              ; MINUS
EBAA 3C4E            2123        CMP  AL, 78           加号键?
EBAC 740C            2124        JE   K51
EBAE 2C47            2125        SUB  AL, 71           ; CONVERT ORIGIN 调整
                                                        AL
```

```
EBB0 BB7AE9      R 2126        MOV  BX, GFFSET K15   ; BASE CASE TABLE
                                                       取特殊字符表首址
EBB3 EB77          2127        JMP  SHORT K64        ; CONVERT TO PSEUDO
                                                       SCAN
                   2128
EBB5 B82D4A        2129 K50:   MOV  AX, 74*256+'-'   ; MINUS  减号的 ASCII
                                                       码 4A 送 AL，键号 74 送
                                                       AH
EBB8 EB22          2130        JMP  SHORT K57        ; BUFFER_FILL
                   2131
EBBA B82B4E        2132 K51:   MOV  AX, 78*256+'+'   ; PLUS  加号的 ASCII 码
                                                       4E 送 AL，键号 78 送 AH
EBBD EB1D          2133        JMP  SHORT K57        ; BUFFER_FILL
                   2134
                   2135 ;……MIGHT BE NUM LOCL, TEST SHIFT STATUS  可
                          能是 NUM LOCK，还需检查是否处移位态
                   2136
EBBF               2137 K52:                         ; ALMOST_NUM _ ST-
                                                       ATE
EBBF F606170003  R 2138        TEST KB_FLAG, LEFT_SHIFT+RIGHT_SHIFT
EBC4 75E0          2139        JNZ  K49              ; SHIFTED TEMP OUT
                                                       OF NUM STATE  处移
                                                       位态，按将 NOM 暂时失
                                                       效处理
                   2140
EBC6               2141 K53:                         ; REALLY_NUM_STATE
                                                       真正处 NUM 态，它将键
                                                       71—73, 75—77, 79—83
                                                       转成数字键
EBC6 2C46          2142        SUB  AL, 70           ; CONVERT ORIGIN  调
                                                       整 AL
EBC8 BB6DE9      R 2143        MOV  BX, OFFSET K14   ; NUM STATE TABLE
                                                       取 NUM 表始址
EBCB EB0B          2144        JMP  SHORT K56        ; TRANSLATE_CHAR
                                                       转 K56 查相应的 ASCII
                                                       码
                   2145
                   2146 ;……PLAIN OLD LOWER CASE  非移位态
                   2147
```

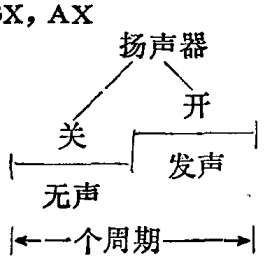
```
ECD                     2148 K54:                       ; NOT_SHIFT
EBCD 3C38               2149        CMP  AL, 59         ; TEST FOR FUNCTION
                                                          KEYS 为F1～F10,10个
                                                          功能键之一?
EBCF 7204               2150        JB   K55            ; NOT_LOWER _FUNC-
                                                          TION 否,转K55
EBD1 B000               2151        MOV  AL, 0          ; SCAN CODE IN AH
                                                          ALREADY 功能键的扩
                                                          展码已在AH中，它等于
                                                          功能键的键号,AL送。
EBC3 EB07               2152        JMP  SHORT K57      ; BUFFER_FILL
                        2153
EBD5                    2154 K55:                       ; NOT_LOWER_ FUNC-
                                                          TION
EBD5 BBE5E8       R     2155        MOV  BX, OFFSET K10 ; LC TABLE 送小写字母
                                                          母表首址
                        2156
                        2157 ;……TRANSLATE THE CHARACTER
                        2158
EBD8                    2159 K56:                       ; TRANSLATE_CHAR
EBD8 FEC8               2160        DEC  AL             ; CONVERT ORIGIN
                                                          (AL) 为欲取ASCII码在
                                                          表中的位置,(BX) ASCII
                                                          表始址
EBDA 2ED7               2161        XLAT CS:K11         ; CONVERT THE SCAN
                                                          CODE TO ASCII 查表,
                                                          ASCII字符送AL
                        2162
                        2163 ;……PUT CHARACTER INTO BUFFER
                        2164
EBDC                    2165 K57:                         BUFFR-FILL
EBDC 3CFF               2166        CMP  AL, -1         ; IS THIS AN IGORE CH-
                                                          AR 是应忽略的字符?
EBDE 741F               2167        JE   K59            ; YES, DO NOTHING W-
                                                          ITH IT
EBE0 80FCFF             2168        CMP  AH, -1         ; LOOK FOR -1 PSEUDO
                                                          SCAN 是值为-1的扩
                                                          充码?
EBE3 741A               2169        JE   K59            ; NEAR _ INTERRUFT_
```

```
                                                    RETURN  符合两种情况
                                                    之一或全部的话转 K59
                    2170
                    2171 ;……HANDLE THE CAPS LOCK PROBLEM
                    2172
EBE5                2173 K58:                      ; BOFFER_FILL _NOT-
                                                      EST
EBE5 F606170040 R   2174        TEST KB_FLAG, CAPS_STATE
                                                    ; ARE WE IN CAPS LOCK
                                                      STATE  正处 CAPS 态?
EBEA 7420           2175        JZ    K61           ; SKIP IF NOT  否,转 K61
                    2176
                    2177 ;……IN CAPS LOCK STATE  处 CAPS 态
                    2178
EBEC F606170003 R   2179        TEST KB_FLAG, LEFT_SHIFT+RIGHT_SHIFT
                                                    ; TEST FOR SHIFT STATE
                                                      正处移位态?
EBF1 740F           2180        JZ    K60           ; IF NOT SHIFT,  CONVERT
                                                      LOWER TO UPPER 否,说明
                                                      要把小写字母转换成大写字母
                    2181
                    2182 ;……CONVERT ANY UPPER CASE TO LOWER CASE
                    2183
EBF3 3C41           2184        CMP   AL, 'A'       ; FIND OUT IF ALPHABETIC
                                                      键入的是大写英文字母?
EBF5 7215           2185        JB    K61           ; NOT_CAPS_STATE
EBF7 3C5A           2186        CMP   AL, 'Z'
EBF9 7711           2187        JA    K61           ; NOT_CAPS_STATE 否,转
                                                      K61
EBFB 0420           2188        ADD   AL, 'a' - 'A' ; CONVERT TO LOWER CASE
                                                      键入大写英文字母,在其 AS-
                                                      CII 码上加 20H,变成相应的
                                                      小写
EBFD EB0D           2189        JMP   SHORT K61     ; NOT_CAPS_STATE  字母
                    2190
EBFF                3291 K59:                      ; NEAR_INTERRUPT_RET-
                                                      URN
EBFF E95CFE         2192        JMP   K26           ; INTERRUPT_RETURN 返回
```

```
                        2193
                        2194 ;……CONVERT ANY LOWER CASE TO UPPER CASE
                        2195
EC02                    2196 K60:                          ; LOWER_TO_UPPER
EC02 3C61               2197      CMP  AL, 'a'             ; FIND OUT IF ALPH-
                                                             ABETIC 键入的是小
                                                             写英文字母?
EC04 7206               2198      JB   K61                 ; NOT_CAPR_STATE
EC06 3C7A               2199      CMP  AL, 'Z'
EC08 7702               2200      JA   K61                 ; NOT_CAPS_STATE
                                                             否,转 K61
EC0A 2C20               2201      SUB  AL, 'a' - 'A'       ; CONVERT TO UPPER
                                                             CASE 键入小写英文
                                                             字母, 从其 ASCII 码减
                                                             去 20H, 变成相应的大
                                                             写字母
                        2202
EC0C                    2203 K61:                          ; NOT_CAPS_ STATE
EC0C 8B1E1C00   R       2204      MOV  BX, BUFFER_TAIL     ; GET THE END POI-
                                                             NTER TO THE BUF-
                                                             FER 键盘缓存区尾指
                                                             针送 BX, SI
EC10 8BF3               2205      MOV  SI, BX              ; SAVE THE VALUE
EC12 E860FC             2206      CALL K4                  ; ADVANCE THE TA-
                                                             IL 尾指针向下移一字
                                                             (保存在 BX 中)
EC15 3B1E1A00   R       2207      CMP  BX,BUFFER_HEAD      ; HAS THE BUFFER
                                                             WRAPPED AROUND
                                                             首尾指针相同说明当前
                                                             缓存区已满了
EC19 7409               2208      JE   K62                 ;BUFFER_FULL_BEEP
                                                             指针值相同,错误
EC1B 8904               2309      MOV  [SI], AX            ; STORE THE VALUE
                                                             键值(ASCII 码,键号或
                                                             0, 扩充 ASCII 码)送缓
                                                             存区
EC1D 891E1C00   R       2210      MOV  BUFFER_TAIL, BX     ; MOVE THE POINTER
                                                             UP 尾指针值送 BUF-
                                                             FER_TAIL 单元
```

```
EC21 E93AFE 2211        JMP  K26            ; INTERRUPT_RETURN 返回
            2212
            2213 ;……BUFFER IS FULL, SOUND THE BEEPER 缓存区满，鸣响
                   蜂鸣器
            2214
EC24        2215 K62:                       ; BUFFER_FULL_BEEP
EC24 E80D00 2216        CALL ERROR_BEEP
EC27 E934FE 2217        JMP  K26            ; INTERRUPT_RETURN
            2218
            2219 ;……TRANSLATE SCAN FOR PSEUDO SCAN CODES
            2220
EC2A        2221 K63:                       ; TRANSLATE_SCAN
EC2A 2C3B   2222        SUB  AL, 59         ; CONVERT ORIGIN TO FUNCTION
                                              KEYS 调整 AL 准备查 CTL 表，
                                              (BX)为 CTL 表始址
EC2C        2223 K64:                       ; TRANSLATE_SCAN_ORGD
EC2C 2ED7   2224        XLAT CS:K9          ; CTL TABLE SCAN 查表，扩充AS-
                                              CII 码送 AL
EC2E 8AE0   2225        MOV  AH, AL         ; PUT VALUE INTO AH (AL) 送
                                              AH
EC30 B000   2226        MOV  AL, 0          ; ZERO ASCII CODE 清 AL，表示
                                              扩充 ASCII 码
EC32 EBA8   2227        JMP  K57            ; PUT IT INTO THE BUFFER
            2228
            2229 KB_INT ENDP
EC34        2230 ERROR_BEEP PROC NEAR 鸣响蜂鸣器程序 ERROR_BEEP
EC34 50     2231        PUSH AX             ; SAVE REGISTERS 保存 AX, BX,
                                              CX
EC35 53     2232        PUSH BX
EC36 51     2233        PUSH CX
EC37 BBC000 2234        MOV  BX, 0C0H       ; NUMBER OF CYCLES FOR 1/8
                                              SECOND TONE 1/8 秒音频周期数
EC3A E461   2235        IN   AL, KB_CTL     ; GET CONTROL INFORMATION
                                              KB_CTL = 61H, 8255 PB 口地址
EC3C 50     2236        PUSH AX             ; SAVE 保存 PB 口内容
EC3D        2237 K65:                       ; BEEP_CYCLE
EC3D 24FC   2238        AND  AL, 0FCH       ; TURN OFF TIMER GATE AND
                                              SPEAKER DATA 清计时器门和扬
                                              声器数据输出端，使扬声器
```

```
EC3F E661    2239        OUT   KB_CTL, AL  ; OUTPUT TO CONTROL  不发声
EC41 B94800  2240        MOV   CX, 48H     ; HALF CYCLE TIME FOR TONE
                                           扬声器寂静时间计数器
EC44 E2FE    2241 K66:   LOOP  K66         ; SPEAKER OFF  保持扬声器静默
EC46 0C02    2242        OR    AL, 2       ; TURN ON SPEAKER BIT  打开扬
                                           声器
EC48 E661    2243        CUT   KB_CTL, AL  ; OUTPUT TO CONTROL
EC4A B94800  2244        MOV   CX, 48H     ; SET UP COUNT
EC4D E2FE    2245 K67:   LOOP  K67         ; ANOTHER HALF CYCLE  让扬声
                                           器发声,时间由(CX)决定
EC4F 4B      2246        DEC   BX          ; TOTAL TIME COUNT  周期数减
                                           为0?
EC50 75EB    2247        JNZ   K65         ; DO ANOTHER CYCLE  否,继续下
                                           一个周期
EC52 58      2248        POP   AX          ; RECOVER CONTROL  送回PB口
                                           原先内容
EC53 E661    2249        OUT   KB_CTL, AL  ; OUTPUT THE CONTROL
EC55 59      2250        POP   CX          ; RECOVER REGISTERS  恢复CX,
                                           BX, AX
```




```
EC56 5B      2251        POP   BX
EC57 58      2252        POP   AX
EC58 C3      2253        RET
             2254 ERROR_BEEP  ENDP  返回
LOC OBJ      LINE  SOURCE
             2255 ; ...INT 13.....
             2256 ; DISKETTE I/O  软中断13_磁盘机I/O(DISKETTEI/O)
             2257 ;       THIS INTERFACE PROVIDES ACCESS TO THE 5 1/4"
                          DISKETTE DRIVES  提供访问5 1/4吋驱动器种种功能
             2258 ; INPUT  入口参数
             2259 ;       (AH) = 0 RESET DISKETTE SYSTEM  (AH) = 0 复位
                                    驱动器
             2260 ;                 HARD RESET TO NEC, PREPARE COMMA-
                                    ND, RECAL RECAL REQD ON ALL DRIVES
                                    强制复位NEC磁盘控制器为全部驱动器准备命
```

```
                        令，重校
2261 ;    (AH) = 1  READ THE STATUS OF THE SYSTEM INTO(AL)
                    (AH) = 1 读取驱动器系统的状态并送 AL，读取的
2262 ;              DISKETTE_STATUS FROM LAST OP'N IS USED
                    状态是最近进行磁盘操作的状态，存于 DISKETTE_
                    STATUS
2263 ;    REGISTERS FOR READ/WRITE/VERIFY/FORMAT  下面是
          读/写/检验/格式化磁盘时 DL，DH，CL，CH，AL 寄存器的作用
2264 ;    (DL) - DRIVE NUMBER(0-3 ALLOWED, VALUE CHECKED)
          (DL)驱动器号(取值 0，1，2，3，进行值检查)
2265 ;    (DH) - HEAD NUMBER (0-1 ALLOWED, NOT VALUE CH-
          ECKED)  DH 磁头号(取值 0，1，不进行值检查)
2266 ;    (CH) - TRACK NUMBER(0-39, NOT VALUE CHECKED)
          (CH)：道号(取值 0～39，不进行值检查)
2267 ;    (CL) - SECTOR NUMBER(1-8, NOT VALUE CHECKED)
          (CL)：扇区号 取值 1～8，不进行值检验
2268 ;    (AL) - NUMBER OF SECTORS (MAX = 8, NOT VALUE CH-
          ECKED) - (AL)：扇区数(最大值为 8，不进行值检查)
2269 ;
2270 ;    (ES,BX) - ADDRESS OF BUFFER(NOT REQUIRED FOR V-
          ERIFY)  (ES:BX)缓存区首址(检验操作时不需要)
2271 ;
2272 ;    (AH) = 2  READ THE DESIRED SECTORS INTO MEMORY
                    (AH) = 2 将数个扇区读入内存
2273 ;    (AH) = 3  WRITE THE DESIRED SECTORS FROM MEMO-
                    RY  (AH) = 3 写数个扇区
2274 ;    (AH) = 4  VERIFY THE DESIRED SECTORS  (AH) = 4检验
                    数个扇区
2275 ;    (AH) = 5  FORMAT THE DESIRED TRACK  (AH) = 5 格式
                    磁道
2276 ;              FOR THE FORMAT OPERATION, THE BUFFER
                    POINTER (ES, BX) MUST  对于格式操作，ES：BX
                    应指向欲格式磁道的地址
2277 ;              POINT TO THE COLLECTION OF DESIRED AD-
                    DRESS FIELDS FOR THE  场区首，每个地址场由
                    4 个字节组成(C, H, R, N)
2278 ;              TRACK. EACH FIELD IS COMPOSED OF 4 BY-
                    TES (C. H, R, N), WHERE  其中 C = 道号，H = 磁
                    头号，R = 扇区号，N = 每扇区含
```

```
;                   C=TRACK NUMBER, H=HEAD NUMBER, R=
;                   SECTOR NUMBER, N=NUMBER  有的字节数(00
;                   =128, 01=256, 02=512, 03=1024)。
;                   OF BYTES PER SECTOR (00=128, 01=256, 02=
;                   512, 03=1024, )  道上的每个扇区只可有一项.
;                   THERE MUST BE ONE ENTRY FOR EVERY
;                   SECTOR ON THE TRACK
;                   THIS INFORMATION IS USED TO FIND THE
;                   REQUESTED SECTOR DURING
;                   READ/WRITE ACCESS
; DATA VARIABLE__DISK__POINTER 数据变量DISK__POINTER
;   DOUBLE WORD POINTER TO THE CURRENT SET OF DI-
;   SKETTE PARAMETERS  双字指针,指向当前磁盘机参数块
; OUTPUT  出口参数
;   AH=STATUS OF OPERATION  AH=操作状态, 其各位的含
;   义同DATA  段
;                   STATUS BITS ARE DEFINED IN THE EQUATES
;                   FOR DISKETTE__STATUS  DISKETTE__STATUS
;                   单元
;                   VARIABLE IN THE DATA SEGMENT OF THIS
;                   MODULE
;   CY=0    SUCCESSFUL OPERATION(AH=0 ON RETURN)
;           CY=0 操作成功(AH=0)
;   CY=1    FAILED OPERATION(AH HAS ERROR REASO-
;           N)  CY=1 操作失败(AH 中为错误信息)
;   FOR READ/WRITE/VERIFY  对读/写/检验操作
;                   DS, BX, DX, CH, CL PRESERVED  DS, BX, DX,
;                   CH, CL 内容不变
;                   AL=NUMBER OF SECTORS ACTUALLY READ
;                   AL=实际读取的扇区号
;                   ******AL MAY NOT BE CORRECT IF TIME
;                   OUT ERROR OCCURS
;   NOTE: IF AN ERROR IS REPORTED BY THE DISKETTE
;   CODE, THE APPROPRIATE  注意: 若错误由磁盘机码报告的
;   话,应采取复位
;                   ACTION IS TO RESET THE DISKETTE, THEN
;                   RETRY THE OPERATION  系统然后再进行该操
;                   作动作. 读取成功则不
;                   ON READ ACCESSES, NO MOTOR START DEL-
```

```
                                AY IS TAKEN, SO THAT  延迟马达启动，固遇
                                到该操作失败情况时，可再进行读操作(最多
                2299 ;          THREE RETRIES ARE REQUIRED ON READS TO ENSURE THAT THE
                                3 次)，避免马达启动产生错误.
                2300 ;          PROBLEM IS NOT DUE TO MOTOR START_UP
                2301 ;.....................................................
                2302            ASSUME CS:CODE, DS:DATA, ES:DATA
EC59            2303 DISKETTE_IO  PROC  FAR
EC59 FB         2304        STI             ; INTERRUPTS BACK ON  允许中断
EC5A 53         2305        PUSH BX         ; SAVE ADDRESS  保存BX, CX, DS,
                                              SI, DI, BP, DX 内容
EC5B 51         2306        PUSH CX           道号，扇区号参数压栈
EC5C IE         2307        PUSH DS         ; SAVE SEGMENT REGISTER VALUE
EC5D 56         2308        PUSH SI         ; SAVE ALL REGISTERS DURING
                                              OPERATION
EC5E 57         2309        PUSH DI
EC5F 55         2310        PUSH BP
EC60 52         2311        PUSH DX           驱动器号，磁头号参数压栈
EC61 8BEC       2312        MOV  BP, SP     ; SET UP POINTER TO HEAD PARM
                                              在原来栈顶再造一个新栈，保存有用的
                                              参数
EC63 BE4000 R   2313        MOV  SI, DATA
EC66 8EDE       2314        MOV  DS, SI     ; SET DATA REGION  DS指向数据段
EC68 E81C00     2315        CALL J1         ; CALL THE REST TO ENSURE DS
                                              RESTORED
EC6B BB0400     2316        MOV  BX, 4      ; GET THE MOTOR WAIT PARAMETER
EC6E EBFF01     2317        CALL GET_PARM   读马达等待时间参数，并送 MOTOR_COUNT 单元
EC71 88264000 R 2318        MOV  MOTOR_COUNT, AH  ; SET THE TIMER COUNT FOR THE MOTOR
EC75 8A264100 R 2319        MOV  AH, DISKETTE_STATUS  ; GET STATUS OF OPERATION
                                                      操作后状态字节送 AH
EC79 80FC01     2320        CMP  AH, 1      ; SET THE CARRY FLAG TO INDIC-
```

```
                                                    ATE  置CY位
EC7C F5          2321          CMC                  ; SUCCESS OR FAILURE  CY
                                                    位取反
EC70 5A          2322          POP   DX             ; RESTDRE ALL REGTSTERS
                                                    恢复 DX, SP, DI, SI, DS, CX,
                                                    BX 寄存器
EC7E 5D          2323          POP   BP
EC7F 5F          2324          POP   DI
EC80 5E          2325          POP   SI
EC81 IF          2326          POP   DS
EC82 59          2327          POP   CX
EC83 5B          2328          POP   BX             ; RECOVER ADDRESS
EC84 CA0200      2329          RET   2              ; THROW AWAY SAVED FL-
                                                    AGS  扔掉标志位的返回
                 2330 DISKETTE_IO  ENDP
EC87             2331 J1:      PROC NEAR
EC87 8AF0        2332          MOV   DH, AL         ; SAVE SECTORS IN DH 扇区
                                                    数送 DH
EC89 80263F007F  R 2333        AND   MOTOR_STATUS, 07FH
                                                    ; INDICATE A READ OPER-
                                                    ATION, MOTOR_STATOS 单
                                                    元第7位送0.
EC8E 0AE4        2334          OR    AH, AH         ; AH=0  复位驱动器?
EC90 7427        2335          JZ    DISK_RESET
EC92 FECC        2336          DEC   AH             ; AH=1  否,读驱动器状态?
EC94 7474        2337          JZ    DISK_STATUS
EC96 C606410000  R 2338        MOV   DISKETTE_STATUS, 0
                                                    ; RESET THE STATUS INDIC-
                                                    ATOR  否, 状态信息保存单元
                                                    清.
EC98 80FA04      2339          CMP   DL, 4          ; TEST FOR DRIVE IN 0-3 R-
                                                    ANGE  驱动器号有效?
EC9E 7313        2340          JAE   J3             ; ERROR IF ABOVE  驱动器号
                                                    是否无效?是否清转 J3
ECA0 FECC        2341          DEC   AH             ; AH=2  读磁盘?
ECA2 746A        2342          JZ    DISK_READ
ECA4 FECC        2343          DEC   AH             ; AH=3  否,写磁盘?
ECA6 7503        2344          JNZ   J2             ; TEST_DISK_VERF  否,转J2
```

```
ECA8 E99600          2345        JMP  DISK_WRITE  写磁盘,去 DISK_WRITE
ECAB                 2346 J2:                     ; TEST_DISK_ VE-
                                                    RF
ECAB FECC            2347        DEC  AH          ; AH=4  检驱磁盘?
ECAD 7468            2348        JZ   DISK_VERF
ECAF FECC            2349        DEC  AH          ; AH=5  否,  格式化
                                                    磁盘?
ECB1 7468            2350        JZ   DISK_FORMAT
ECB3                 2351 J3:                     ; BAD_COMMAND
                                                    命令无效
ECB3 C606410001 R    2352        MOV  DISKETTE_STATUS, BAD_CMD
                                                    ; ERROR CODE, NO
                                                    SECTORS TRANSF-
                                                    ERRED  置命令无
                                                    效标记(第 0 位置位)
ECB8 C3              2353        RET              ; UNDEFINED OPE-
                                                    RATION  J1 子程序
                                                    返回
                     2354 J1     ENDP
                     2355
                     2356 ;……RESET THE DISKETTE SYSTEM  复位磁盘机 (软
                             盘机)
                     2357
ECB9                 2358 DISK_RESET  PROC  NEAR
ECB9 BAF203          2359        MOV  DX, 03F2H   ; ADAPTER CONT-
                                                    ROL PORT  软盘机
                                                    数据口地址
ECBC FA              2360        CLI              ; NO INTERRUPTS
                                                    关中
ECBD A03F00      R   2361        MOV  AL, MOTOR_STATUS ; WHICH MOTOR IS
                                                    ON  测试哪一个驱
                                                    动器的马达正在旋转
ECC0 B104            2362        MOV  CL, 4       ; SHIFT COUNT
ECC2 D2E0            2363        SAL  AL, CL      ; MOVE MOTOR V-
                                                    ALUE  TO  HIGH
                                                    NYBBLE  表示马达
                                                    旋转信息的低四位移
                                                    向高 4 位
ECC4 A820            2364        TEST AL, 20H     ; SELECT  CORRES-
```

```
                                                    PONIDNG DRIVE 马
                                                    达1正工作?
ECC6 750C       2365        JNZ   J5                ; JUMP IF MOTOR O-
                                                    NE IS ON
ECC8 A840       2366        TEST  AL  40H           否,马达2正工作?
ECCA 7506       2367        JNZ   J4                ; JUMP IF MOTOR T-
                                                    WO IS ON
ECCC A880       2368        TEST  AL, 80H           否,马达3正工作?
ECCE 7406       2369        JZ    J6                ; JUMP IF MOTOR ZE-
                                                    RO IS ON  马达口正
                                                    工作转J6
ECD0 FEC0       2370        INC   AL                允使驱动器1
ECD2 FEC0       2371 J4:    INC   AL                允使驱动器2
ECD4 FEC0       2372 J5:    INC   AL                允使驱动器3
ECD6 0C08       2373 J6:    OR    AL, 8             ; TURN ON INTERRU-
                                                    PT ENABLE  第3位
                                                    置1, 允使驱动器系统
                                                    发中断信号和DMA请
                                                    求
ECD8 EE         2374        OUT   DX, AL           ; RESET THE ADAPT-
                                                    ER  控制字送数据口
ECD9 C6063E0000 R 2375      MOV   SEEK_STATUS, 0    ; SET RECAL REQUI-
                                                    RED ON ALL DRIVES
                                                    SEEK_STATUS 单元
                                                    送口,表示下次SEEK操
                                                    作前应进行重校
ECDE C606410000 R 2376      MOV   DISKETTE_STATUS, 0
                                                    ; SET OK STATUS FOR
                                                    DISKETTE  状态字节
                                                    清0无错标志
ECE3 0C04       2377        OR    AL, 4             ; TURN OFF RESET
ECE5 EE         2378        OUT   DX, AL            ; TURN OFF THE RE-
                                                    SET  送非FDC复位信
                                                    号
ECE6 FB         2379        STI                     ; REENABLE  THE
                                                    INTERRUPTS  开中
ECE7 E82802     2380        CALL  CHK_STAT_2        ; DO SENSE INTERR-
                                                    UPT STATUS FOLL-
                                                    OWING RESET  复位
```

```
                                                后去 CHK_STAT_2
                                                测试中断状态
ECEA A04200      R 2381     MOV  AL, NEC_STATUS  ; IGNORE ERROR RE-
                                                TURN AND DO OWN
                                                TEST  取测中断状态
                                                命令的返回信息的第一
                                                个字节 STO
ECED 3CC0         2382      CMP  AL, 0C0H        ; TEST FOR DRIVE
                                                READY TRANSITION
ECEF 7407         2383      JZ   J7              ; EVERYTHING OK
                                                STO 第6, 7 位全 1 表示
                                                非正常结束, FDD来的
                                                Roady 信号改变了电平
ECF1 800E410020 R 2384      OR   DISKETTE_STATUS, BAD_NEC
                                                ; SET ERROR CODE
                                                送BAD_NEC(20H)置
                                                错误位
ECF6 EB11         2385      JMP  SHORT J8        ; RESET_RET  转J8,
                                                准备返回
                  2386
                  2387 ;……SEND SPECIFY COMMAND TO NEC  NEC工作正
                                                常
                  2388
ECF8              2389 J7:                       ; DRIVE_READY
ECF8 B403         2390      MOV  AH, 03H         ; SPECIFY COMMAND
ECFA E84701       2391      CALL NEC_OUTPUT      ; OUTPUT THE COM-
                                                MAND  将 03H 指定
                                                的命令 (软盘机初始命
                                                令)送给 NEC
ECFD BB0100       2392      MOV  BX, 1           ; FIRST BYTE PARM
                                                IN BLOCK
ED00 E86D01       2393      CALL GET_PARM        ; TO THE NEC CONT-
                                                ROLLER,  送步进率
                                                (OCH), 磁头去载时间
                                                (OFH)
ED03 BB0300       2394      MOV  BX, 3           ; SECOND BYTE PARM
                                                IN BLOCK
ED06 E86701       2395      CALL GET_PARM        ; TO THE NEC CON-
                                                TROLLER  送磁头加
```

```
                                                    载时间,DMA方式
ED09                    2396 J8:                    ; RESET_RET
ED09 C3                 2397        RET             ; RETURN TO CALLER
                                                    复位软盘机子程序结束
                        2398 DISK_RESET  ENDP
                        2399
                        2400 ;……DISKETTE STATUS ROUTINE 读软盘机状态子程序
                        2401
ED0A                    2402 DISK_STATUS  PROC  NEAR
ED0A A04100         R   2403        MOV  AL, DISKETTE_STATUS  DISKETTE_ST-
                                        ATUS单元记录了软盘机状态,送 AL
ED0D C3                 2404        RET
                        2405 DISK_STATUS  ENDP  读软盘机状态子程序结束
                        2406
LOC OBJ                 LINE  SOURCE
                        2407 ;……DISKETTE READ  读软盘机子程序
                        2408
ED0E                    2409 DISK_READ  PROC  NEAR
ED0E B046               2410        MOV  AL, 046H       ; READ COMMAND FOR
                                                    DMA  送 DMA 读命令
                                                (模式字节,选第二通道,单模式传送)
ED10                    2411 J9:                    ; DISK_READ_CONT
ED10 E8B901             2412        CALL DMA_SETUP  ; SET UP THE DMA  建
                                                    立 DMA
ED13 B466               2413        MOV  AH, 066H   ; SET UP READ COMM-
                                                    AND FOR NEC CONT-
                                                    ROLLER  向 NEC 控制
                                                    器送读盘命令
ED15 EB36               2414        JMP  SHORT RW_OPN ; GO DO THE OPERAT-
                                                    ION  转 RW_OPN
                        2415 DISK_READ  ENDP
                        2416
                        2417 ;……DISKETTE VERIFY  检验软盘机子程序
                        2418
ED17                    2419 DISK_VERF  PROC  NEAR
ED17 B042               2420        MOV  AL, 042H   ; VERIFY COMMAND F-
                                                    OR DMA  DMA 检验命
                                                    令 (模式字节, 选第二通
                                                    道,单模式传送)
```

```
ED19 EBF5              2421      JMP  J9              ; DO AS IF DISK READ
                                                        转 J9
                       2422 DISK_VERF  ENDP
                       2423
                       2424 ;……DISKETTE FORMAT 格式化软盘磁道子程序
                       2425
ED1B                   2426 DISK_FORMAT  PROC  NEAR
ED1B 800E3F0080 R      2427      OR   MOTOR_STATUS, 80H  ; INDICATE WRI-
                                                          TE OPERATION
                                                          MOTOR_STATUS
                                                          单元第7位送1表
                                                          示写操作
ED20 B04A              2428      MOV  AL, 04AH           ; WILL WRITE TO THE DIS-
                                                          KETTE 模式字节，写操作，
                                                          第二通道，单模式传送
D22 E8A701             2429      CALL DMA_SETUP          ; SET UP THE DMA
                                                          送DMA写命令
ED25 B44D              2430      MOV  AH, 04DH           ; ESTABLISH THE FORMAT
                                                          COMMAND NEC 格式化命
                                                          令
ED27 EB24              2431      JMP  SHORT RW_OPN       ; DO THE OPERATION
                                                          转 RW_OPN
ED29                   2432 J10:                         ; CONTINUATION OF RW_
                                                          OPN FOR FMT
ED29 BB0700            2433      MOV  BX, 7              ; GET THE
ED2C E84101            2434      CALL GET_PARM           ; BYTES/SECTOR VALUE TO
                                                          NEC 每扇面多少字节数参数
                                                          送NEC
ED2F BB0900            2435      MOV  BX, 9              ; GET THE
ED32 E83B01            2436      CALL GET_PARM           ; SECTORS/TRACK VALUE
                                                          TO NEC 每道扇区数送NEC
ED35 BB0F00            2437      MOV  BX, 15             ; GET THE
ED38 E83501            2438      CALL GET_PARM           ; GAP LENGTH VALUE TO
                                                          NEC 间隔长度参数送NEC
ED3B BB1100            2439      MOV  BX, 17             ; GET THE FILLER BYTE
ED3E E9AB00            2440      JMP  J16                ; TO THE CONTROLLER 转
                                                          J16 取格式化填充字节并送
                                                          NEC
                       2441 DISK_FORMAT  ENDP
```

```
                        2442
                        2443 ;……DISKETTE WRITE ROUTINE 写软盘子程序
                        2444
ED41                    2445 DISK_WRITE PROC NEAR
ED41 800E3F0080 R       2446        OR   MOTOR_STATUS,80H ; INDICATE WRITE
                                                            OPERATION 置写
                                                            操作标记
ED46 B04A               2447        MOV  AL, 04AH         ; DMA WRITE CO-
                                                            MMAND 送 DMA
                                                            命令(模式字节,写操
                                                            作,第二通道,单模式
                                                            传送)
ED48 E88101             2448        CALL DMA_SETUP
ED4B B445               2449        MOV  AH, 045H         ; NEC COMMAND TO
                                                            WRITE TO DISKE-
                                                            TTE NEC 写命令
                        2450 DISK_WRITE ENDP
                        2451 ;……ALLOW WRITE ROUTINE TO FALL INTO
                                    RW_OPN
                        2452 ;……
                        2453 ; RW_OPN RW_OPN 程序段,完成读/写/检验操作
                        2454 ;        THIS ROUTINE PERFORMS THE READ/
                                    WRITE/VERIFY OPERATION
                        2455 ;……
ED4D                    2456 RW_OPN PROC NEAR
ED4D 7308               2457        JNC  J11              ; TEST FOR DMA
                                                            ERROR DMA 有错?
ED4F C606410009 R       2458        MOV  DISKETTE_STATUS, DMA_BOUNDARY
                                                          ; SET ERROR CY=
                                                            1 表示 DMA 错,送错
                                                            误标志
ED54 B000               2459        MOV  AL, 0            ; NO SECTORS TR-
                                                            ANSFERRED 实际
                                                            传送扇区数为 0
ED56 C3                 2460        RET                   ; RETURN TO MAIN
                                                            ROUTINE 子程序
                                                            结束
ED57                    2461 J11:                         ; DO_RW_OPN
ED57 50                 2462    PUSH AX                   ; SAVE THE COM-
```

```
                        2463
                        2464 ;…TURN ON THE MOTOR AND SELECT THE DRIVE
                        2465
ED58 51                 2466    PUSH CX                         ; SAVE THE T/S
                                                                  PARMS 保存磁道/
                                                                  扇区号
ED59 8ACA               2467    MOV  CL, DL                     ; GET DRIVE NUM-
                                                                  BER AS SHIFT C-
                                                                  OUNT 驱动器号送
                                                                  CL, AL
ED5B B001               2468    MOV  AL, 1                      ; MASK FOR DETE-
                                                                  RMINING MOTOR
                                                                  BIT
ED5D D2E0               2469    SAL  AL, CL                     ; SHIFT THE MASK
                                                                  BIT AL 左移一位,
                                                                  使其与 MOTOR_S-
                                                                  TATUS 单元内容对
                                                                  齐
ED5F FA                 2470    CLI                             ; NO INTERRUPTS
                                                                  WHILE DETERMI-
                                                                  NING MOTOR ST-
                                                                  ATUS 关中
ED60 C6064000FF R       2471    MOV  MOTOR_COUNT, 0FFH          ; SET LARGE COU-
                                                                  NT DURING OPER-
                                                                  ATION 马达超时
                                                                  计数器置最大值
ED65 84063F00   R       2472    TEST AL, MOTOR_STATUS           ; TEST THAT MOT-
                                                                  OR FOR OPERAT-
                                                                  ING
ED69 7531               2473    JNZ  J14                        ; IF RUNNING, SKIP
                                                                  THE WAIT 指定驱
                                                                  动器的马达正工作?
ED6B 80263F00F0 R       2474    AND  MOTOR_STATUS, 0F0H         ; TURN OFF ALL
                                                                  MOTOR BITS 否,
                                                                  复位全部马达工作位
ED70 08063F00   R       2475    OR   MOTOR_STATUS, AL           ; TURN ON THE
                                                                  CURRENT MOTOR
                                                                  置指定驱动器马达工
```

```
                                                    作标记
ED74 FB          2476        STI                    ; INTERRUPTS BACK ON
                                                    开中
ED75 B010        2477        MOV  AL, 10H           ; MASK BIT
ED77 D2E0        2478        SAL  AL, CL            ; DEVELOP BIT MASK FOR
                                                    MOTOR ENABLE 准备允使
                                                    驱动器位
ED79 0AC2        2479        OR   AL, DL            ; GET DRIVE SELECT BITS
                                                    IN 送指定驱动器选择位
ED7B 0C0C        2480        OR   AL, 0CH           ; NO RESET, ENABLE DMA/
                                                    INT 送非复位及允使 DMA,
                                                    中断位
ED7D 52          2481        PUSH DX                ; SAVE REG 保存驱动器号
                                                    和扇区数(DL, DH)
ED7E BAF203      2482        MOV  DX, 03F2H         ; CONTROL PORT ADDRESS
ED81 EE          2483        OUT  DX, AL            以上字节送驱动器数据输出口

ED82 5A          2484        POP  DX                ; RECOVER REGISTERS
                                                    恢复 DX 驱动器号和扇区数
                 2485
                 2486 ;……WAIT FOR MOTOR IF WRITE OPERATION
                 2487
ED83 F6063F0080 R 2488       TEST MOTOR_STATUS, 80H ; IS THIS A WR-
                                                    ITE 是写操作?
ED88 7412        2489        JZ   J14               ; NO, CONTINUE WITHOUT
                                                    WAIT 非写操作转 J14
ED8A BB1400      2490        MOV  BX, 20            ; GET THE MOTOR WAIT
ED8D E8E000      2491        CALL GET_PARM          ; PARAMETER 取马达启动
                                                    时间参数
ED90 0AE4        2492        OR   AH, AH            ; TEST FOR NO WAIT
ED92             2493 J12:                          ; TEST_WAIT_TIME
ED92 7408        2494        JZ   J14               ; EXIT WITH TIME EXPIR-
                                                    ED (AH)为 0 不等待
ED94 2BC9        2495        SUB  CX, CX            ; SET UP 1/8 SECOND LOOP
                                                    TIME 清 0 CX
ED96 E2FE        2496 J13:   LOOP J13               ; WAIT FOR THE REQUIRED
                                                    TIME
ED98 FECC        2497        DEC  AH                ; DECREMENT TIME VAL-
                                                    UE 延迟 1/8 秒
```

```
ED9A EBF6          2498          JMP  J12              ; ARE WE DONE YET
                   2499
ED9C               2500 J14:                          ; MOTOR_RUNNING
ED9C FB            2501          STI                  ; INTERRUPTS BACK ON
                                                          FOR BYPASS WAIT 开中
ED9D 59            2502          POP  CX                退栈;磁道/扇区号送 CX
                   2503
                   2504 ;……DO THE SEEK OPERATION 进行 SEEK (寻道)操
                                作
                   2505
ED9E E8E000        2506          CALL SEEK            ; MOVE TO CORRECT TR-
                                                          ACK 调 SEEK, 使磁头移
                                                          向指定的磁道
EDA1 58            2507          POP  AX              ; RECOVER COMMAND
                                                          取回命令并送 BH
EDA2 8AFC          2508          MOV  BH, AH          ; SAVE COMMAND IN BH
EDA4 B600          2509          MOV  DH, 0           ; SET NO SECTORS READ
                                                          IN CASE OF ERROR DH清
                                                          0 出现错误时表示未读扇区
EDA6 724B          2510          JC   J17             ; IF ERROR, THEN EXIT
                                                          AFTER MOTOR OFF CY
                                                          =1, 出错, 转 J17
EDA8 BEF3ED90  R   2511          MOV  SI, OFFSET J17  ; DUMMY RETURN ON S-
                                                          TACK FOR NEC_OUTPUT
                                                          J17 压栈, 作出错后的返回地
                                                          址
EDAC 56            2512          PUSH SI              ; SO THAT IT WILL RET-
                                                          URN TO MOTOR OFF L-
                                                          OCATION
                   2513
                   2514 ;……SEND OUT THE PARAMETERS TO THE CONT-
                                ROLLER
                   2515
EDAD E89400        2516          CALL NEC_OUTPUT      ; OUTPUT THE OPERATI-
                                                          ON COMMAND 调 NEC_
                                                          OUTPUT, 送出 AH 中的
                                                          NEC 命令
EDB0 8A6601        2517          MOV  AH, [BP+1]      ; GET THE CURRENT H-
                                                          EAD NUMBER 取磁头
```

```
                                                    号,该参数在第 2311 行被压入
                                                    栈
EDB3 D0E4       2518        SAL     AH, 1       ; MOVE IT TO BIT 2
EDB5 D0E4       2519        SAL     AH, 1
EDB7 80E404     2520        AND     AH, 4       ; ISOLATE THAT BIT  磁头
                                                    号由 AH 第 2 位表示, 1 为 1
                                                    号磁头, 0 为 0 号磁头
EDBA 0AE2       2521        OR      AH, DL      ; OR IN THE DRIVE NUMB-
                                                    ER  或入驱动器号
EDBC E88500     2522        CALL    NEC_OUTPUT      调 NEC_OUTPUT 送通道程
                                                    序的第二个字节
                2523
                2524 ;……TEST FOR FORMAT COMMAND
                2525
EDBF 80FF4D     2526        CMP     BH, 04DH    ; IS THIS A FORMAT OPER-
                                                    ATION  是格式化磁道操作?
EDC2 7503       2527        JME     J15         ; NO, CONTINUE WITH R/
                                                    W/V  否,转 J15
EDC4 E962FF     2528        JMP     J10         ; IF SO, HANDLE SPECIAL
                                                    是,转 J10 特别处理
                2529
EDC7 8AE5       2530 J15:   MOV     AH, CH      ; CYLINDER NUMBER  送磁
                                                    道号至 NEC
EDC9 E87800     2531        CALL    NEC_OUTPUT
EDCC 8A6601     2532        MOV     AH, [BP+I]  ; HEAD NUMBER FROM ST-
                                                    ACK  送磁头号至 NEC
EDCF E87200     2533        CALL    NEC_OUTPUT
EDD2 8AE1       2534        MOV     AH, CL      ; SECTOR NUMBER  送扇区
                                                    号至 NEC
EDD4 E86000     2535        CALL    NEC_OUTPUT
EDD7 BB0700     2536        MOV     BX, 7       ; BYTES/SECTOR PARM FR-
                                                    OM BLOCK  送每扇区多个
                                                    字节参数至 NEC
EDDA E89300     2537        CALL    GET_PARM    ; TO THE NEC
EDDD BB0900     2538        MOV     BX, 9       ; EOT PARM FROM BLOCK
                                                    送磁道最后扇区号参数至NEC
EDE0 E88D00     2539        CALL    GET_PARM    ; TO THE NEC
EDE3 BB0B00     2540        MOV     BX, 11      ; GAP LENGTH PARM FROM
                                                    BLOCK  送间隔长度参数至
```

```
                                                    NEC
EDE6 E88700     2541        CALL    GET_PARM        ; TO THE NEC
EDE9 BB0D00     2542        MOV     BX, 13          ; DTL PARM FROM BLOCK
EDEC            2543 J16:                           ; RW_OPN_FINISH
EDEC E88100     2544        CALL    GET_PARM        ; TO THE NEC   送数据长度
                                                    参数至NEC
EDEF 5E         2545        POP     SI              ; CAN NOW DISCARD THAT
                                                    DUMMY RETURN ADDRESS
                                                    退栈，错误发生时的返回地址
                                                    无用了
                2546
                2547 ;……LET THE OPERATION HAPPEN
                2548
EDF0 E84001     2549        CALL    WAIT_INT        ; WAIT FOR THE INTERRU-
                                                    PT   操作完成后NEC控制器
                                                    将发来中断信号，固调WAIT
                                                    _INT
EDF3            2550 J17:                           ; MOTOR_OFF
EDF3 7245       2551        JC      J21             ; LOOK FOR ERROR   CY=1
                                                    表示发生了错误，转J21
EDF5 E87301     2552        CALL    RESULTS         ; GET THE NEC STATUS
                                                    取NEC状态信息
EDF8 723F       2553        JC      J20             ; LOOK FOR ERROR
                2554
                2555 ;……CHECK THE RESULTS RETURNED BY THE CONTRO-
                        LLER  检查控制器返回的结果
                2556
EDFA FC         2557        CLD                     ; SET THE CORRECT DIRE-
                                                    CTION   前向标志
EDFB BE4200 R   2558        MOV     SI, OFFSET NEC_STATUS   ; POINT TO ST-
                                                            ATUS FIELD
                                                            状态信息区首址
                                                            送SI，状态格式
                                                            请参阅软盘机转
                                                            接口有关章节
EDFE AC         2559        LODS    NEC_STATUS      ; GET STO   STO字节送AL
EDFF 24C0       2560        AND     AL, 0C0H        ; TEST FOR NORMAL TER-
                                                    MINATION   正常结束则第
                                                    6、7位非全0
```

```
LOC OBJ      LINE  SOURCE
EE01 743B    2561      JZ    J22              ; OPN_OK  全0，转J22操作正常
                                                结束
EE03 3C40    2562      CMP   AL, 040H         ; TEST FOR ABNORMAL TERMI-
                                                NATION  第7位为0，第6位为1
                                                说明命令未能正常终止
EE05 7529    2563      JNZ   J18              ; NOT ABNORMAL, BAD NEC
                                                D7 = 0, D6 = 1 转J18 属NEC错误
             2564
             2565 ;……ABNORMAL TERMINATION, FIND OUT WHY
             2566
EE07 AC      2567      LODS  NEC_STATUS       ; GET ST1  取ST1字节
EE08 D0E0    2568      SAL   AL, 1            ; TEST FOR EOT FOUND  测ST1
                                                最高位，为1表示欲读扇区的区号
                                                大于允许最大扇区号
EE0A B404    2569      MOV   AH, RECORD_NOT_FND  欲访问扇区未找到标志
                                                (04H)送AH
EE0C 7224    2570      JC    J19              ; RW_FAIL  最高位1转J19
EE0E D0E0    2571      SAL   AL, 1
EE10 D0E0    2572      SAL   AL, 1            ; TEST FOR CRC ERROR  测ST1
                                                第5位，为1表示发生CRC错
EE12 B410    2573      MOV   AH, BAD_CRC      CRC错误标志(10H)送AH
EE14 721C    2574      JC    J19              ; RW_FAIL  第5位为1转J19
EE16 D0E0    2575      SAL   AL, 1            ; TEST FOR DMA OVERRUN  测
                                                ST1第4位，为1表示DMA错
EE18 B408    2576      MOV   AH, BAD_DMA      送DMA错标志
EE1A 7216    2577      JC    J19              ; RW_FAIL  为1转J19
EE1C D0E0    2578      SAL   AL, 1
EE1E D0E0    2579      SAL   AL, 1            ; TEST FOR RECORD NOT FOUND
                                                测ST1第2位，为1表示未找到扇
                                                区
EE20 B404    2580      MOV   AH, RECORD_NOT_FND  送扇区未找到错标志
EE22 720E    2581      JC    J19              ; RW_FAIL  为1转J19
EE24 D0E0    2582      SAL   AL, 1              测ST1第1位，为1表示写盘错误
EE26 B403    2583      MOV   AH, WRITE_PROTECT  ; TEST FOR WRITE_
                                                PROTECT  送标志
EE28 7208    2584      JC    J19              ; RW_FAIL  为1转J19
EE2A D0E0    2585      SAL   AL, 1            ; TEST MISSING ADDRESS MA-
                                                RK  测ST1第0位，为1表示FDC
```

```
                                                找不到地址标识码
EE2C B402       2586        MOV     AH, BAD_ADDR_MARK   送标志
EE2E 7202       2587        JC      J19                 ; RW_FAIL  为1转J19
                2588
                2589 ;……NEC MUST HAVE FAILED
                2590
EE30            2591 J18:                               ; RW_NEC_FAIL
EE30 B420       2592        MOV     AH, BAD_NEC         送NEC错误标记
EE32            2593 J19:                               ; RW_FAIL
EE32 08264100 R 2594        OR      DISKETTE_STATUS, AH 驱动器状态字节送
                                                        DISKETTE_STATUS单元
EE36 E87701     2595        CALL    NUM_TRANS           ; HOW MANY WERE REAL-
                                                        LY TRANSFERRED 调
                                                        NUM_TRANS,将实际传送的
                                                        扇区数送 AL
EE39            2596 J20:                               ; RW_ERR
EE39 C3         2597        RET                         ; RETURN TO CALLER  J1
                                                        子程序结束返主
                2598
EE3A            2599 J21:                               ; RW_ERR_RES  等待中断
                                                        程序中发生超时错,转此处理
EE3A E82E01     2600        CALL    RESULTS             ; FLUSH THE RESULTS BU-
                                                        FFER  取NEC状态
EE3D C3         2601        RET  返主
                2602
                2603 ;……OPERATION WAS SUCCESSFUL  操作成功
                2604
EE3E            2605 J22:                               ; OPN_OK
EE3E E86F01     2606        CALL    NUM_TRANS           ; HOW MANY GOT MOVED
                                                        实际传送的扇区数送 AL
EE41 32E4       2607        XOR     AH, AH              ; NO ERRORS  清AH成功标
                                                        记
EE43 C3         2608        RET  返主
                2609 RW_OPN  ENDP
                2610 ;……
                2611 ; NEC_OUTPUT  NEC_OUTPUT 子程序
                2612 ;      THIS ROUTINE SENDS A BYTE TO THE NEC CONT-
                            ROLLER  本程序将一个字节送NEC控制器,发送前将测试
                            传送方向及NEC准备好信号,若在一定时间内发现字节未
```

```
                        被接收,则置超时标志. 发送完毕后置软盘机状态
                ;       AFTER TESTING FOR CORRECT DIRECTION AND
                        CONTROLLER READY
                ;       THIS ROUTINE WILL TIME OUT IF THE BYTE
                        IS NOT ACCEPTED
                ;       WITHIN A REASONABLE AMOUNT OF TIME,
                        SETTING THE DISKETTE STATUS
                ;       ON COMPLETION
                ; INPUT  入口参数
                ;       (AH)  BYTE TO BE OUTPUT  (AH):待送的字节
                ; OUTPUT  出口参数
                ;       CY=0  SUCCESS  CY=0 成功
                ;       CY=1  FAILURE--DISKETTE STATUS UPDATED
                              CY=1 失败(状态信息被更新)
                ;             IF A FAILURE HAS OCCURRED, THE
                              RETURN IS MADE ONE LEVEL 发出错误
                              时的返回地址是较 NEC_OUTPUT 高一级的
                ;             HIGHER THAN THE CALLER OF NEC
                              _OUTPUT 地址(在栈中事先设好了)
                ;             THIS REMOVES THE REQUIREMENT OF
                              TESTING AFTER EVERY CALL
                ;             OF NEC_OUTPUT
                ;       (AL)  DESTROYED (AL)被破坏
                ;.....
EE44            NEC_OUTPUT  PROC  NEAR
EE44 52                 PUSH DX         ; SAVE REGISTERS  保护
                                          DX, CX
EE45 51         PUSH    CX
EE46 BAF403     MOV     DX, 03F4H       ; STATUS PORT  NEC 主状
                                          态寄存器地址(3F4H)送 DX
EE49 33C9       XOR     CX, CX          ; COUNT FOR TIME OUT
                                          CX 清 0, 设置超时计数器初
                                          值
EE4B            J23:
EE4B EC         IN      AL, DX          ; GET STATUS  取状态字节
EE4C A840       TEST    AL, 040H        ; TEST DIRECTION BIT  测
                                          方向位
EE4E 740C       JZ      J25             ; DIRECTION OK  如为 0 说
                                          明是 CPU 至 FDC 方向 (正常
```

```
                                                    方向)
EE50 E2F9          2637         LOOP    J23             循环
EE52               2638 J24:                    ; TIME_ERROR  方向错(超
                                                    时错)
EE52 800E410080 R  2639         OR      DISKETTE_STATUS, TIME_OUT  送超
                                                    过错标记
EE57 59            2640         POP     CX
EE58 5A            2641         POP     DX      ; SET ERROR CODE AND
                                                    RESTORE REGS
EE59 58            2642         POP     AX      ; DISCARD THE RETURN
                                                    ADDRESS  扔本程序返址,
                                                    靠下一返址返回(J17)
EE5A F9            2643         STC             ; INDICATE ERROR TO CA-
                                                    LLER  CY 置1错误标记
EE5B C3            2644         RET                 返主
                   2645
EE5C               2646 J25:
EE5C 33C9          2647         XOR     CX, CX  ; RESET THE COUNT  方向
                                                    正确,复位计数器
EE5E               2648 J26:
EE5E EC            2649         IN      AL, DX  ; GET THE STATUS  读取状
                                                    态字节
EE5F A880          2650         TEST    AL, 080H; IS IT READY  测其第7位,
                                                    为1说明FDC准备好接收数
                                                    据字节3
EE61 7504          2651         JNZ     J27     ; YES, GO OUTPUT  =1 转
                                                    J27
EE63 E2F9          2652         LOOP    J26     ; COUNT DOWN AND TRY
                                                    AGAIN 循环等待FDC准备好
EE65 EBEB          2653         JMP     J24     ; ERROR CONDITION  超
                                                    时错, 转J24
EE67               2654 J27:                    ; OUTPUT
EE67 8AC4          2655         MOV     AL, AH  ; GET BYTE TO OUTPUT
                                                    欲输出字节送 AL
EE69 BAF503        2656         MOV     DX, 03F5H; DATA PORT  FDC数据寄存
                                                    器地址(3F5H)送 DX
EE6C EE            2657         OUT     DX, AL  ; OUTPUT THE BYTE  送出
                                                    字节
```

```
EE6D 59          2658    POP     CX                      ; RECOVER REGISTERS
                                                           恢复 CX, DX
EE6E 5A          2659    POP     DX
EE6F C3          2660    RET                             ; CY=0 FROM TEST
                                                           INSTRUCTION 返主
                 2661 NEC_OUTPUT  ENDP
                 2662 ;.....
                 2663 ; GET_PARM  GET_PARM 子程序
                 2664 ;  THIS ROUTINE FETCHES THE INDEXED POINTER
                         FROM 本子程序从DISK_POINTER 指向的 DISK_BASE
                         参数区取一字节送 AH
                 2665 ;  THE DISK_BASE BLOCK POINTED AT BY THE DATA
                 2666 ;  VARIABLE DISK_POINTER
                 2667 ; A BYTE FROM THAT TABLE IS THEN MOVED INTO
                        AH
                 2668 ;  THE INDEX OF THAT BYTE BEING THE PARM IN BX
                 2669 ; ENTRY—入口参数
                 2670 ;  BX= INDEX OF BYTE TO BE FETCHED*2   (BX) 2*
                             待取字符偏差,若BX最低位为1,取出的字数送 NEC
                 2671 ;      IF THE LOW BIT OF BX IS ON, THE BYTE IS
                             IMMEDIATELY 控制器
                 2672 ;      OUTPUT TO THE NEC CONTROLLER
                 2673 ; EXIT--出口参数
                 2674 ;  AH=THAT BYTE FROM BLOCK  (AH),取出的字节
                 2675 ;.....
EE70             2676 GET_PARM   PROC    NEAR
EE70 1E          2677    PUSH    DS                      ; SAVE SEGMENT 保存
                                                           DS
EE71 2BC0        2678    SUB     AX, AX                  ; ZERO TO AX  AX清 0
EE73 8ED8        2679    MOV     DS, AX
                 2680    ASSUME  DS:ABS0  DS 指向 8088 中断向量表首址
EE75 C5367800    2681    LDS     SI, DISK_POINTER        ; POINT TO BLOCK DI-
                                                           SK POINTER 内容送DI,
                                                           使其指向软盘机操作参数块
EE74 D1EB        2682    SHR     BX, 1                   ; DIVIDE BX BY 2, AND
                                                           SET FLAG FOR EXIT
                                                           右环移 BX
EE7B 8A20        2683    MOV     AH, [SI+BX]             ; GET THE WORD 参数
                                                           表中由 BX 指出的字节送
```

```
                                                    AH
EE7D 1F         2684        POP     DS              ; RESTORE SEGMENT
                                                      恢复 DS
                2685        ASSUME  DS; DATA
EE7E 72C4       2686        JC      NEC_OUTPUT      ; IF FLAG SET, OUTP-
                                                      UT TO CONTROLLER
                                                      入口参数 (BX) 最低位
                                                      为 1 则将取来的参送送
                                                      NEC
EE80 C3         2687        RET                     ; RETURN TO CALLER
                                                      控制器.
                2688 GET_PARM   ENDP                  返回
                2689 ;.....
                2690 ; SEEK  SEEK 子程序
                2691 ;       THIS ROUTINE WILL MOVE THE HEAD ON THE
                             NAMED DRIVE  本程序将磁头移向指定驱动器盘上的
                             指定磁道
                2692 ;       TO THE NAMED TRACK. IF THE DRIVE HAS NOT
                             BEEN ACCESSED  若驱动器复位后未被访问过,则进行
                             重校工作
                2693 ;       SINCE THE DRIVE RESET COMMAND WAS ISSU-
                             ED, THE DRIVE WILL BE
                2694 ;       RECALIBRATED
                2695 ; INPUT  入口参数
                2696 ;       (DL) = DRIVE TO SEEK ON  (DL): 欲寻道所在的驱
                             动器号
                2697 ;       (CH) = TRACK TO SEEK TO  (DH): 欲寻道的道号
                2698 ; OUTPUT  出口参数
                2699 ;       CY = 0 SUCCESS  CY = 0 寻道成功
                2700 ;       CY = 1 FAILURE--DISKETTE STATUS SET ACCOR-
                             DINGLY  CY = 1 寻道失败,相应状态字节位被置位
                2701 ;       (AX)DESTROYED  (AX)被破坏
                2702 ;.....
EE81            2703 SEEK   PROC    NEAR
EE81 B001       2704        MOV     AL, 1           ; ESTABLISH MASK
                                                      FOR RECAL TEST
                                                      置重校标记
EE83 51         2705        PUSH    CX              ; SAVE INPUT VALU-
                                                      ES  道号暂存
```

```
EE84 8ACA       2706        MOV     CL, DL              ; GET DRIVE VALUE
                                                        INTO CL  驱动器号
                                                        送 CL
EE86 D2C0       2707        ROL     AL, CL              ; SHIFT IT BY THE
                                                        DRIVE VALUE  重校
                                                        标记与驱动器号重合
EE88 59         2708        POP     CX                  ; RECOVER TRACK
                                                        VALUE
EE89 84063E00 R 2709        TEST    AL, SEEK_STATUS     ; TEST FOR RECAL
                                                        REQUIRED  该驱动
                                                        器SEEK 前需重校? 为
                                                        0 表 RECAL
EE8D 7513       2710        JNZ     J28                 ; NO_RECAL  不需要
EE8F 08063E00 R 2711        OR      SEEK_STATUS,AL      ; TURN ON THE NO
                                                        RECAL BIT IN FLAG
                                                        置位重校位，表示下次
                                                        SEEK 操作前无需重校
EE93 B407       2712        MOV     AH, 07H             ; RECALIBRATE CO-
                                                        MMAND  送重校命令
EE95 E8ACFF     2713        CALL    NEC_OUTPUT
EE98 8AE2       2714        MOV     AH, DL
EE9A E8A7FF     2715        CALL    NEC_OUTPUT          ; OUTPUT THE DRIVE
                                                        NUMBER  送待重校
                                                        的驱动器号
EE9D E87200     2716        CALL    CHK_STAT_2          ; GET THE INTERRUPT
                                                        AND SENSE INT ST
                                                        ATUS  进入 CHK_
                                                        STAT_2等待重校结束
EEA0 7229       2717        JC      J32                 ; SEEK_ERROR  重校
                                                        有错转 J32
                2718
                2719 ;……DRIVE IS IN SYNCH WITH CONTROLLER, SEEK
                        TO TRACK  驱动器与控制器同步,现进行 SEEK 操作
                2720
EEA2            2721 J28:
EEA2 B40F       2722        MOV     AH, 0FH             ; SEEK COMMAND TO
                                                        NEC  寻道命令送NEC
EEA4 E89DFF     2723        CALL    NEC_OUTPUT
```

```
EEA7 8AE2      2724        MOV     AH, DL          ; DRIVE NUMBER 驱动器号
                                                     送NEC
EEA9 E898FF    2725        CALL    NEC—OUTPUT
EEAC 8AE5      2726        MOV     AH, CH          ; TRACK NUMBER 道号送
                                                     NEC
EEAE E893FF    2727        CALL    NEC__OUTPUT
EEB1 E85E00    2728        CALL    CHK—STAT__2     ; GET ENDING INTERRUPT
                                                     AND SENSE STATUS 等待
                                                     寻道结束
               2729
               2730 ;……WAIT FOR HEAD SETTLE
               2731
EEB4 9C        2732        PUSHF                   ; SAVE STATUS FLAGS 保
                                                     存各标志位
EEB5 BB1200    2733        MOV     BX, 18          ; GET HEAD SETTLE PAR-
                                                     AMETER 取磁头稳定时间
                                                     参数
EEB8 E8B5FF    2734        CALL    GET__PARM       送参数
EEBB 51        2735        PUSH    CX              ; SAVE REGISTER
EEBC           2736 J29:                           ; HEAD__SETTLE
EEBC B92602    2737        MOV     CX, 550         ; 1MS LOOP
EEBF 0AE4      2738        OR      AH, AH          ; TEST FOR TIME EXPIRED
                                                     稳定时间取 0 则不延迟
EEC1 7406      2739        JZ      J31
EEC3 E2FE      2740 J30,   LOOP    J30             ; DELAY FOR 1 MS 延迟 1
                                                     毫秒
EEC5 FECC      2741        DEC     AH              ; DECREMENT THE COUNT
                                                     稳定计数器减 1
EEC7 EBF3      2742        JMP     J29             ; DO IT SOME MORE 循环,
                                                     使磁头稳定
EEC9           2743 J31:
EEC9 59        2744        POP     CX              ; RECOVER STATE 恢复
                                                     CX, 标志位
EECA 9D        2745        POPF
EECB           2746 J32:                           ; SEEK__ERROR
EECB C3        2747        RET                     ; RETURN TO CALLER 返主
               2748 SEEK   ENDP
               2749 ;.....
               2750 ; DMA__SETUP DMA__SETUP 子程序
```

```
                2751 ;      THIS ROUTINE SETS UP THE DMA FOR READ/
                            WRITE/VERIFY 本程序为读/写/检验盘操作建立 DMA
                2752 ;      OPERATIONS
                2753 ; INPUT 入口参数:
                2754 ;      (AL) = MODE BYTE FOR THE DMA (AL):DMA 模式
                            字节
                2755 ;      (ES:BX) - ADDRESS TO READ/WRITE THE DATA
                            (ES:BX):欲读/写数据的始址
                2756 ; OUTPUT 出口参数:
                2757 ;      (AX)DESTROYED (AX)被破坏
                2758 ;.....
EECC            2759        DMA_SETUP PROC NEAR
EECC 51         2760        PUSH  CX              ; SAVE THE REGISTER 保存
                                                    CX
EECD E60C       2761        OUT   DMA+12, AL      ; SET THE FIRST/LAST F/F
                                                    DMA 为 8237 片起始地址,本指
                                                    令清字节指针触发器
EECF E60B       2762        OUT   DMA+11, AL      ; OUTPUT THE MODE BYTE
                                                    输出 DMA 模式字节
EED1 8CC0       2763        MOV   AX, ES          ; GET THE ES VALUE
EED3 B104       2764        MOV   CL, 4           ; SHIFT COUNT
EED5 D3C0       2765        ROL   AX, CL          ; ROTATE LEFT ES 段某寄存
                                                    器循环左移 4 位
EED7 8AE8       2766        MOV   CH, AL          ; GET HIGHEST NYBBLE OF
                                                    ES TO CH ES 高 8 位送 CH
EED9 24F0       2767        AND   AL, 0F0H        ; ZERO THE LOW NYBBLE F-
                                                    ROM SEGMENT 去除 ES 高
                                                    4 位
EEDB 03C3       2768        ADD   AX, BX          ; TEST FOR CARRY FROM A-
                                                    DDITION 加偏移地址,得绝对
                                                    地址
EEDD 7302       2769        JNC   J33       CY=1 表示有进位;应在 ES 高 4 位上加 1
EEDF FEC5       2770        INC   CH              ; CARRY MEANS HIGH 4 BITS
                                                    MUST BE INC
                                                    ES 高 4 位加 1
EEE1            2771 J33:
EEE1 50         2772        PUSH  AX              ; SAVE START ADDRES 保存
                                                    起始地址的低 16 位
EEE2 E604       2773        OUT   DMA+4, AL       ; OUTPUT LOW ADDRESS 送
                                                    低地址
```

```
EEE4 8AC4    2774        MOV     AL, AH
EEE6 E604    2775        OUT     DMA+4, AL   ; OUTPUT HIGH ADDRESS 送高地址
EEE8 8AC5    2776        MOV     AL, CH      ; GET HIGH 4 BITS
EEEA 240F    2777        AND     AL, 0FH       取绝对地址的高4位
EEEC E681    2778        OUT     081H, AL    ; OUTPUT THE HIGH 4 BITS TO PAGE REGISTER 高4位送DMA页寄存器
             2779
             2780 ;……DETERMINE COUNT          决定DMA转送的字节数
             2781
EEEE 8AE6    2782        MOV     AH, DH      ; NUMBER OF SECTORS 扇区数送AH
EEF0 2AC0    2783        SUB     AL, AL      ; TIMES 256 INTO AX, AL清0
EEF2 D1E8    2784        SHR     AX, 1       ; SECTORS*128 INTO AX 扇区数乘256并送AX
EEF4 50      2785        PUSH    AX
EEF5 BB0600  2786        MOV     BX, 6       ; GET THE BYTES/SECTOR PARM 取每扇区字节数参数(00=128, 01=256, 02=512, 03=1024)
EEF8 E875FF  2787        CALL    GET_PARM
EEFB 8ACC    2788        MOV     CL, AH      ; USE AS SHIFT COUNT (0=128, 1=256 ETC) 参数送CL作为移位计数器
EEFD 58      2789        POP     AX
EEFE D3E0    2790        SHL     AX, CL      ; MULTIPLY BY CORRECT AMOUNT 调整AX, 使其记录正确的字节/扇区
EF00 48      2791        DEC     AX          ; -1 FOR DMA VALUE AX减1
EF01 50      2792        PUSH    AX          ; SAVE COUNT VALUE DMA传送字节数压栈
EF02 E605    2793        OUT     DMA+5, AL   ; LOW BYTE OF COUNT 送计数值低字节部分至DMA通道2基字计数器
EF04 8AC4    2794        MOV     AL, AH
EF06 E605    2795        OUT     DMA+5, AL   ; HIGH BYTE OF COUNT 送高
```

```
                                                    字节部分
EF08 59         2796    POP     CX              ; RECOVER COUNT VALUE
                                                  DMA 传送字节数送 CX
EF09 58         2797    POP     AX              ; RECOVER ADDRESS VA-
                                                  LUE  内存 DMA 传送区低 16
                                                  位送 AX
EF0A 03C1       2798    ADD     AX, CX          ; ADD, TEST FOR 64K OV-
                                                  ERFLOW  相加，判断传送字
                                                  节数是否大于 64KB
EF0C 59         2799    POP     CX              ; RECOVER REGISTER  恢复
                                                  CX
EF0D B002       2800    MOV     AL, 2           ; MODE FOR 8237
EF0F E60A       2801    OUT     DMA+10, AL      ; INITIALIZE THE DISKETTE
                                                  CHANNEL 清通道 2 屏蔽位，
                                                  启动 DMA
EF11 C3         2802    RET                     ; RETURN TO CALLER, CFL
                                                  SET BY ABOVE IF ERROR
                                                  返回，CY = 1 表示越页错
                2803 DMA_SETUP  ENDP
                2804 ;.....
                2805 ; CHK_STAT_2  CHK_STAT_2 子程序
                2806 ;      THIS ROUTINE HANDLES THE INTERRUPT RECEIVED
                           AFTER  本程序处理重校，寻道，复位操作后的 FDC 中断
                2807 ;      A RECALIBRATE, SEEK, OR RESET TO THE ADAPTER
                           程序等待中断发生，然后检测中断状态，并返回它。
                2808 ;      THE INTERRUPT IS WAITED FOR,  THE INTERRUPT
                           STATUS SENSED
                2809 ;      AND THE RESULT RETURNED TO THE CALLER
                2810 ; INPUT  入口参数：
                2811 ;      NONE  无
                2812 ; OUTPUT  出口参数：
                2813 ;      CY = 0 SUCCESS  CY = 0 成功
                2814 ;      CY = 1 FAILURE--ERROR IS IN DISKETTE_STATUS
                           CY = 1 失败，错误种类在 DISKETTE_STATUS 单元
                2815 ;      (AX) DESTROYED  (AX)被破坏
                2816 ;.....
EF12            2817 CHK_STAT_2  PROC  NEAR
EF12 E81E00     2818        CALL    WAIT_INT    ; WAIT FOR THE INTERR-
                                                  UPT  进入 WAIT_INT 等待
```

```
                                                    FDC 中断
EF15 7214          2819        JC      J34          ; IF ERROR, RETURN IT
                                                    CY = 1 超时错
EF17 B408          2820        MOV     AH, 08H      ; SENSE INTERRUPT S-
                                                    TATUS COMMAND 送
                                                    检测中断状态命令
EF19 E828FF        2821        CALL    NEC_OUTPUT
EF1C E84C00        2822        CALL    RESULTS      ; READ IN THE RESU-
                                                    LTS 取回状态信息
EF1F 720A          2823        JC      J34          ; CHK2_RETURN
                                                    CY = 1 NEC 错
EF21 A04200      R 2824        MOV     AL, NEC_STATUS ; GET THE FIRST S-
                                                    TATUS BYTE 状态
                                                    字节送 AL(STO字节)
EF24 2460          2825        AND     AL, 060H     ; ISOLATE THE BITS
EF26 3C60          2826        CMP     AL, 060H     ; TEST FOR CORRECT
                                                    VALUE
EF28 7402          2827        JZ      J35          ; IF ERROR, GO MARK
                                                    IT, SEEK 命令非正常结
                                                    束的话转 J35
EF2A F8            2828        CLC                  ; GOOD RETURN, CY 送
                                                    0 成功标记
EF2B               2829 J34:
EF2B C3            2830        RET                  ; RETURN TO CALLER
                                                    返主
EF2C               2831 J35:                        ; CHK2_ERROR
EF2C 800E410040  R 2832        OR      DISKETTE_STATUS, BAD_SEEK 在 DI-
                                                    SKETTE_STATUS 单元置 SEEK 错误标记
EF31 F9            2833        STC                  ; ERROR RETURN CODE
                                                    CY = 1, 错误标记
EF32 C3            2834        RET
                   2835        CHK_STAT_2 ENDP 返回
                   2836 ;.....
                   2837 ;      WAIT_INT WAIT_INT 子程序
                   2838 ;      THIS ROUTINE WAITS FOR AN INTERRUPT TO
                                OCCUR 本程序等待 FDC 中断发生，若在规定的时间
                                内中断未发生，则出现超时错.
                   2839 ;      A TIME OUT ROUTINE TAKES PLACE DURING
                                THE WAIT, SO
```

```
                        2840 ;      THAT AN ERROR MAY BE RETURNED IF THE
                                    DRIVE IS NOV READY
                        2841 ; INPUT  入口参数:
                        2842 ;      NONE  无
                        2843 ; OUTPUT  出口参数:
                        2844 ;      CY = 0 SUCCESS  CY = 0 成功
                        2845 ;      CY = 1 FAILURE--DISKETTE_STATUS IS SET A-
                                    CCORDINGLY  CY = 1 失败,DISKETTE_STATUS 单
                                    元相应位置 1
                        2846 ;      (AX) DESTROYED  (AX)被破坏
                        2847 ;.....................................................
EF33                    2848 WAIT_INT  PROC  NEAR
EF33 FB                 2849        STI             ; TURN ON INTERRUPTS,  JUST
                                                      IN CASE  允使中断发生
EF34 53                 2850        PUSH   BX
EF35 51                 2851        PUSH   CX       ; SAVE REGISTERS
EF36 B302               2852        MOV    BL, 2    ; CLEAR THE COUNTERS  清CX
                                                      置等待 2 秒计数器
EF38 33C9               2853        XOR    CX, CX   ; FOR 2 SECOND WAIT
EF3A                    2654 J36:
EF3A F6063E0080 R       2855        TEST   SEEK_STATUS, INT_FLAG
                                                    ; TEST FOR INTERRUPT OCCU-
                                                      RRING  SEEK_STATUS 第 7 位
                                                      为 1 表示中断发生
EF3F 750C               2856        JNZ    J37      中断发生的话转 J37
EF41 E2F7               2857        LOOP   J36      ; COUNT DOWN WHILE WAIT-
                                                      ING  循环等待
EF43 FECB               2858        DEC    BL       ; SECOND LEVEL COUNTER
EF45 75F3               2859        JNZ    J36      已等了 2 秒?
EF47 800E410080  R      2860        OR     DISKETTE_STATUS, TIME_OUT
                                                    ; NOTHING HAPPENED  超时错
                                                      (等待 2 秒中断未发生)
EF4C F9                 2861        STC             ; ERROR RETURN  CY 送 1
EF4D                    2862 J37:
EF4D 9C                 2863        PUSHF           ; SAVE CURRENT CARRY  保存
                                                      标志位
EF4E 80263E007F R       2864        AND    SEEK_STATUS, NOT INT_FLAG
                                                    ; TURN OFF INTERRUPT FLAG
                                                      清中断标识位
```

```
EF53 9D              2865      POPF                ; RECOVER CARRY  恢复标志
                                                       位
EF54 59              2866      POP     CX
EF55 5B              2867      POP     BX          ; RECOVER REGISTERS
EF56 C3              2868      RET                 ; GOOD RETURN CODE COMES
                                                       FROM TEST INST  返回
                     2869 WAIT_INT  ENDP

                     2870 ;.....
                     2871 ; DISK_INT  DISK_INT 子程序
                     2872 ;    THIS ROUTINE HANDLES THE DISKETTE INTE-
                               RRUPT  本程序段处理软盘机中断
                     2873 ; INPUT  入口参数:
                     2874 ;    NONE  无
                     2875 ; OUTPUT  出口参数:
                     2876 ;    THE INTERRUPT FLAG IS SET IS SEEK_STATUS
                               SEE_STATUS 单元中断位置位
                     2877 ;.....
EF57                 2878 DISK_INT  PROC  FAR
EF57 FB              2879      STI                 ; RE ENABLE INTERRUPTS
EF58 1E              2880      PUSH    DS
EF59 50              2881      PUSA    AX
EF5A B84000      R   2882      MOV     AX, DATA
EF5D 8ED8            2883      MOV     DS, AX  DS 指向数据段
EF5F 800E3E0080  R   2884      OR      SEEK_STATUS, INT_FLAG  SEEK_ST-
                                                       ATUS 单元第 7 位(中断位)置 1
EF64 B020            2885      MOV     AL, 20H     ; END OF INTERRUPT MARK
                                                       ER
EF66 E620            2886      OUT     20H, AL     ; INTERRUPT CONTROL PORT
                                                       送中断结束(EOI)命令
EF68 58              2887      POP     AX
EF69 1F              2888      POP     DS          ; RECOVER SYSTEM
EF6A CF              2889      IRET                ; RETURN FROM INTERRUPT
                                                       恢复 DS中断处理程序返回
                     2890 DISK_INT  ENDP
                     2891 ;.....
                     2892 ; RESULTS  RESULTS 子程序
                     2893 ;    THIS ROUTINE WILL READ ANYTHING THAT
                               THE NEC CONTROLLER  本程序返回 NEC 控制器发
```

```
                        出中断后的状态
                2894 ;      HAS TO SAY FOLLOWING AN INTERRUPT
                2895 ; INPUT  入口参数:
                2896 ;      NONE  无
                2897 ; OUTPUT  出口参数:
                2898 ;      CY=0 SUCCESSFUL TRANSFER   CY=0 传送成功
                2899 ;      CY=1 FAILURE   TIME OUT IN WAITING FOR
                            STATUS  CY=1 失败,等待状态时间太长,超时错
                2900 ;      NEC_STATUS AREA HAS STATUS BYTE LOAD-
                            ED INTO IT   NEC_STATUS 起的 7 个字节存放
                            NEC 返回的状态信息
                2901 ;      (AH)DESTROYED
                2902 ;.....
EF6B            2903 RESULTS PROC  NEAR
EF6B FC         2904        CLD                          前向标记
EF6C BF4200   R 2905        MOV    DI, OFFSET NEC_STATUS
                                                  ; POINTER TO DATA AREA
                                                    DI 指向存放 NEC 状态的数据
                                                    区
EF6F 51         2906        PUSH   CX             ; SAVE COUNTER  保存CX,
                                                    DX, BX
EF70 52         2907        PUSH   DX
EF71 53         2908        PUSH   BX
EF72 B307       2909        MOV    BL, 7          ; MAX STATUS BYTES  状态
                                                    字节的最大个数
                2910
                2911 ;.....WAIT FOR REQUEST FOR MASTER
                2912
EF74            2913 J38:                         ; INPUT_LOOP
EF74 33C9       2914        XOR    CX, CX         ; COUNTER
EF76 BAF403     2915        MOV    DX, 03F4H      ; STATUS PORT  软盘转接器
                                                    主状态寄存器地址3F4H送DX
EF79            2916 J39:                         ; WAIT FOR MASTER
EF79 EC         2917        IN     AL, DX         ; GET STATUS  读取状态
EF7A A880       2918        TEST   AL, 080H       ; MASTER READY  测最高
                                                    位, 为1说明软盘机数据寄存
                                                    器准备好从/向 CPU 发送/接
EF7C 750C       2919        JNZ    J40A           ; TEST_DIR  收数据
EF7E E2F9       2920        LOOP   J39            ; WAIT_MASTER  最高位为
```

```
                                                    0，循环等待
EF80 800E410080 R 2921      OR      DISKETTE_STATUS_TIME_OUT 设置超
                                                    时错标记
EF85              2922 J40:                         ; RESULTS_ERROR
EF85 F9           2923      STC                     ; SET ERROR RETURN  CY
                                                    送1，表明有错
EF86 5B           2924      POP     BX
EF87 5A           2925      POP     DX
EF88 59           2926      POP     CX
EF89 C3           2927      RET                     返回
                  2928
                  2929 ;……TEST THE DIRECTION BIT
                  2930
EF8A EC           2931 J40A: IN     AL, DX          ; GET STATUS REG  AGAIN
                                                    读取主状态内容
EF8B A840         2932      TEST    AL, 040H        ; TEST DIRECTION BIT  第
                                                    6位为1则这时的方向是软盘
                                                    机至 CPU
EF8D 7507         2933      JNZ     J42             ; OK TO READ STATUS  读
                                                    状态第6位应为1
EF8F              2934 J41:                         ; NEC_FAIL
EF8F 800E410020 R 2935      OR      DISKETTE_STATUS, BAD_NEC  方向不
                                                    对，置相应标记
EF94 EBEF         2936      JMP     J40             ; RESULTS_ERROR  转J40
                                                    准备返回
                  2937
                  2938 ;……READ IN THE STATUS
                  2939
EF96              2940 J42,                         ; INPUT_STAT
EF97 42           2941      INC     DX              ; POINT AT DATA PORT
                                                    DX 指向数据口
EF96 EC           2942      IN      AL, DX          ; GET THE DATA  读数据
                                                    (状态字节)存入内存数据区
EF98 8805         2943      MOV     [DI], AL        ; STORE THE BYTE
EF9A 47           2944      INC     DI              ; INCREMENT THE POINTER
                                                    数据区指针加1
EF9B B90A00       2945      MOV     CX, 10          ; LOOP TO KILL TIME FOR
                                                    NEC
```

```
EF9E E2FE        2946 J43:  LOOP   J43     等待,让软盘机送出下一状态字节或完成命令
EFA0 4A          2947       DEC    DX      ; POINT AT STATUS PORT  DX 指
                                              向主状态寄存器
EFA1 EC          2948       IN     AL, DX  ; GET STATUS  取主状态内容
EFA2 A810        2949       TEST   AL, 010H ; TEST FOR NEC STILL BUSY  软
                                              盘机忙(第4位=1)说明命令尚未执
                                              行完
EFA4 7406        2950       JZ     J44     ; RESULTS DONE  为0,转J44
EFA6 FECB        2951       DEC    BL      ; DECREMENT THE STATUS CO-
                                              UNTER  读取字节计数器减1
EFA8 75CA        2952       JNZ    J38     ; GO BACK FOR MORE  再读
EFAA EBE3        2953       JMP    J41     ; CHIP HAS FAILED
                 2954
                 2955 ;……RESULT OPERATION IS DONE  返回结果工作完成
                 2956
EFAC             2956 J44:
EFAC 5B          2958       POP    BX        恢复BX, DX, CX
EFAD 5A          2959       POP    DX
EFAE 59          2960       POP    CX      ; RECOVER REGISTERS
EFAF C3          2961       RET            ; GOOD RETURN CODE FROM T-
                                              EST INST  返回
                 2962 ;.....
                 2963 ; MUM_TRANS  NUM_TRANS 子程序
                 2964 ;    THIS ROUTINE CALCULATES THE NUMBER OF SE-
                           CTORS THAT  本程序计算实际读/写的扇区数
                 2965 ;    WERE ACTUALLY TRANSFERRED TO/FROM THE D-
                           ISKETTE
                 2966 ; INPUT  入口参数
                 2967 ;    (CH) = CYLINDER OF OPERATION  (CH): 磁道号
                 2968 ;    (CL) = START SECTOR OF OPERATION  (CL): 起始扇
                           区号
                 2969 ; OUTPUT  出口参数
                 2970 ;    (AL) = NUMBER ACTUALLY TRANSFERRED  (AL):
                           实际传送的扇区数
                 2971 ;    NO OTHER REGISTERS MODIFIED
                 2972 ;.....
EFB0             2973 NUM_TRANS  PROC  NEAR
EFB0 A04500 R    2974       MOV    AL, NEC_STATUS+3  ; GET CYLINDER E-
                                                        NDED UP ON  结
```

```
                                                束时的磁道号送 AL
EFB3 3AC5       2975        CMP     AL, CH              ; SAME AS WE STA-
                                                        RTED 比较
EFB5 A04700 R   2976        MOV     AL, NEC_STATUS+5    ; GET ENDING SECT-
                                                        OR 结束时的扇区号
                                                        送 AL
EFB8 740A       2977        JZ      J45                 ; IF ON SAME CYL,
                                                        THEN NO ADJUST
                                                        磁道号对，去J45 计算
                                                        扇区数
EFBA BB0800     2978        MOV     BX, 8
EFBD E8B0FE     2979        CALL    GET_PARM            ; GET EOT VALUE
                                                        取最大扇区数,并送 AL
EFC0 8AC4       2980        MOV     AL, AH              ; INTO AL
EFC2 FEC0       2981        INC     AL                  ; USE EOT+1 FOR C-
                                                        ALCULATION 调整
                                                        AL
EFC4 2AC1       2982 J45:   SUB     AL, CL              ; SUBTRACT START
                                                        FROM END 结束时
                                                        的扇区号或最大扇区号
                                                        减起始扇区号得实际传
                                                        送的扇区数
EFC6 C3         2983        RET
                2984 NUM_TRANS  ENDP                    返回
                2985 RESULTS ENDP
                2986
                2987 ;.....
                2988 ; DISK_BASE 下面是磁盘操作必需的参数区,该区首址由
                2989 ;  THIS IS THE SET OF PARAMETERS REQUIRED FOR
                        DISK_POINTER 指出,用户可自行造一个参数区,用 DISK_
                        POINTER 指向他,但应注意参数区的格式。
                2990 ;  DISKETTE OPERATION. THEY ARE POINTED AT BY
                        THE
                2991 ;  DATA VARIABLE DISK_POINTER. TO MODIFY THE
                        PARAMETERS
                2992 ;  BUILD ANOTHER PARAMETER BLOCK AND POINT AT IT
                2993 ;.....
                2994
EFC7            2995 DISK_BASE  LABEL  BYTE
```

```
EFC7 CF    2996        DB    11001111B        ; SRT = C, HD UNLOAD = 0F - 1ST
                                                SPECIFY BYTE  SRT = C, HD 去
                                                载时间 = OF
EFC8 02    2997        DB    2                ; HD LOAD = 1,  MODE = DMA -
                                                2ND SPECIFY BYTE  磁头加载
                                                时间 = 1, DMA 传送
EFC9 25    2998        DB    MOTOR_WAIT       ; WAIT AFTER OPN TIL MOTOR
                                                OFF  操作后等待直至马达关闭
EFCA 02    2999        DB    2                ; 512 BYTES/SECTOR  512 字节/
                                                扇区
EFCB 08    3000        DB    8                ; EOT(LAST SECTOR ON TRACK)
                                                磁道最大扇区号
EFCC 2A    3001        DB    02AH             ; GAP LENGTH  间隔长
EFCD FF    3002        DB    0FFH             ; DTL  数据长
EFCE 50    3003        DB    050H             ; GAP LENGTH FOR FORMAT
                                                格式化时间的间隔长
EFCF F6    3004        DB    0F6H             ; FILL BYTE FOR FORMAT  格式
                                                化用的填充数据
EFD0 19    3005        DB    25               ; HEAD SETTLE TIME  (MILLI-
                                                SECONDS)  磁头稳定时间(毫秒)
EFD1 04    3006        DB    4                ; MOTOR START TIME (1/8 SE-
                                                CONDS)  马达启动时间(1/8 秒)

           3007 ;……INT 17……软中断 17—打印机-IO
           3008 ; PRINTER_IO
           3009 ;          THIS ROUTINE PROVIDES COMMUNICATION WITH
                           THE PRINTER  本程序提供与打印机接口的种种功能
           3010 ;          (AH) = 0 PRINT THE CHARACTER IN (AL)  (AH)
                                    = 0 打印 AL 中的字符, 返回时 AH = 1 表示该字
           3011 ;                   ON RETURN,  AH = 1 IF CHARACTER CO-
                                    ULD NOT BE PRINTED (TIME OUT)  符因
                                    超时不能打印, 其余位同 (AH) = 2 时的状态返
                                    回信息
           3012 ;                   OTHER BITS SET AS ON NORMAL STATUS
                                    CALL
           3013 ;          (AH) = 1 INITIALIZE THE PRINTER PORT  (AH) = 1
                                    初始打印机口, 状态在 AH 返回
           3014 ;                   RETURNS WITH(AH) SET WITH PRINTER
                                    STATUS
```

```
                ;       (AH) = 2 READ THE PRINTER STATUS INTO (AH)
                                (AH) = 2 读取打印机状态并在 AH 返回。
                ;
                ;                7   6   5   4   3   2-1  0  AH各位的含义见
                ;                |   |   |   |   |    |-TIME OUT 左图。
                ;                |   |   |   |   -UNUSED
                ;                |   |   |   -1=I/O ERROR
                ;                |   |   -1=SELECTED
                ;                |   -1=OUT OF PAPER
                ;                -1=ACKNOWLEDGE
                ;                -1=BUSY
                ;
                ;       (DX) = PRINTER TO BE USED (0, 1, 2) CORRESPO-
                                NDING TO ACTUAL VALUES  (DX)：打印机
                                号(0, 1 或 2)
                ;                IN PRINTER_BASE AREA
                ; DATA AREA PRINTER_BASE CONTAINS THE BASE ADD-
                  RESS OF THE PRINTER CARD (S)   可用打印机口地址存放
                  在 PRINTER_BASE 指示的数据区域中，绝对地址 408H，占 3 个
                  字
                ;  AVAILABLE (LOCATED AT BEGINNING OF DATA SEGME-
                  NT,  408H ABSOLUTE,  3 NORDS)
                ; REGISTERS      AH IS MODIFIED  全部寄存器中只有 AH 内容
                                被修改过
                ;                ALL OTHERS UNCHANGED
                ;.....................................................
                        ASSUME CS:CODE, DS:DATA
EFD2            PRINTER_IO      PROC  FAR
EFD2 FB                 STI             ; INTERRUPTS BACK ON  允使中断
EFD3 1E                 PUSH    DS      ; SAVE SEGMENT
EFD4 52                 PUSH    DX
EFD5 56                 PUSH    SI
EFD6 51                 PUSH    CX
EFD7 53                 PUSH    BX
EFD8 BE4000 R
                        MOV     SI, DATA
EFDB 8EDE               MOV     DS, SI  ; ESTABLISH PRINTER SEGMENT
                                          DS 指向数据段
EFDD 8BF2               MOV     SI, DX  ; GET PRINTER PARM  打印机号送
                                          SI
EFDF D1E6               SHL     SI, 1   ; WORD OFFSET INTO TABLE
```

```
                                        (SI)左移,得字偏移
EFE1 8B940800 R 3044    MOV   DX, PRINTER_BASE(SI)  ; GET BASB ADD-
                                                      RESS FOR PRIN-
                                                      TER CARD 取打
                                                      印机口地址
EFE5 0BD2       3045    OR    DX, DX    ; TEST DX FOR ZERO, INDICATING
                                          NO PRINTER
EFE7 740C       3046    JZ    B1        ; RETURN 地址为0说明该打印机未
                                          接
EFE9 0AE4       3047    OR    AH, AH    ; TEST FOR (AH)=0 打印字符?
EFEB 740E       3048    JZ    B2        ; PRINT_AL 是,转B2
EFED FECC       3049    DEC   AH        ; TEST FOR (AH)=1 初始打印机
                                          口?
EFEF 7442       3050    JZ    B8        ; INIT_PRT 是,转B8
EFF1 FECC       3051    DEC   AH        ; TEST FOR (AH)=2 读状态?
EFF3 742A       3052    JZ    B5        ; PRINTER STATUS 是,转B5
EFF5            3053 B1:              ; RETURN
EFF5 5B         3054    POP   BX
EFF6 59         3055    POP   CX
EFF7 5E         3056    POP   SI        ; RECOVER REGISTERS
EFF8 5A         3057    POP   DX        ; RECOVER REGISTERS 恢复寄存
                                          器
EFF9 1F         3058    POP   DS
EFFA CF         3059    IRET  返主
                3060
                3061 ;……PRINT THE CHARACTER IN (AL) 下面程序段将AL
                        中的字符送打印机打印
                3062
EFFB            3063 B2:
EFFB 50         3064    PUSH  AX        ; SAVE VALUE TO PRINT 字符保
                                          存
EFFC B30A       3065    MOV   BL, 10    ; TIME OUT VALUE 超时计时器初
                                          值送BL
EFFE 33C9       3066    XOR   CX, CX    ; ESTABLISH SHIFT COUNT CX
                                          清0
F000 EE         3067    OUT   DX, AL    ; OUTPUT CHAR TO PORT 字符送
                                          打印机
F001 42         3068    INC   DX        ; POINT TO STATUS PORT DX加
                                          1,指向打印机实时状态,读取口
```

```
F002            3069 B3:                        ; WAIT_BUSY
F002 EC         3070        IN      AL, DX      ; GET STATUS 取打印机状态,
F003 8AE0       3071        MOV     AH, AL      ; STATUS TO AH ALSO 状态送AH
F005 A880       3072        TEST    AL, 80H     ; IS THE PRINTER CURRENTLY
                                                  BUSY 判状态字节第7位,为1说明
                                                  打印机正忙,不能接收字符
F007 750E       3073        JNZ     B4          ; OUT_STROBE
F009 E2F7       3074        LOOP    B3          ; DECREMENT COUNT ON TIME
                                                  OUT 循环,等待Busy信号降下
F00B FECB       3075        DEC     BL
F00D 75F3       3076        JNZ     B3          ; WAIT FOR NOT BUSY
F00F 80CC01     3077        OR      AH, 1       ; SET ERROR FLAG AH送1,超时
                                                  错
F012 80E4F9     3078        AND     AH, 0F9H    ; TURN OFF THE OTHER BITS 清
                                                  状态字节1,2,3位,其中第3位表示
                                                  I/O错,1,2位无用
F015 EB14       3079        JMP     SHORT B7    ; RETURN WITH ERROR FLAG S-
                                                  ET 转B7,返回状态信息
F017            3080 B4:                        ; OUT_STROBE
F017 B00D       3081        MOV     AL, 0DH     ; SET THE STROBE HIGH
F019 42         3082        INC     DX          ; STROBE IS BIT 0 OF PORT C OF
                                                  8255 未位为选通位(STROBE),为
                                                  1将STROBE信号线置成高电平
F01A EE         3083        OUT     DX, AL
F01B B00C       3084        MOV     AL, 0CH     ; SET THE STROBE LOW 未位送
                                                  0,将STROBE信号线
F01D EE         3085        OUT     DX, AL      置成低电平
F01E 58         3086        POP     AX          ; RECOVER THE OUTPUT CHAR
                                                  取回输出字符
                3087
                3088 ;……PRINTER STATUS 下面程序段取打印机状态
                3089
F01F            3090 B5:
F01F 50         3091        PUSH    AX          ; SAVE AL REG
F020            3092 B6:
F020 8B
     940800 R   3093        MOV     DX, PRINTER_BASE(SI) 打印机口地址送DX
F024 42         3094        INC     DX                   DX加1,指向实时状态口
```

```
F025 EC       3095        IN      AL, DX      ; GET PRINTER STATUS  读状态
F026 8AE0     3096        MOV     AH, AL        状态送 AH
F028 80E4F8   3097        AND     AH, 0F8H    ; TURN OFF UNUSED BITS  状态字
                                                节第 1, 2 无用位清 0
F02B          3098 B7:                      ; STATUS_SET
F02B 5A       3099        POP     DX          ; RECOVER AL REG   输出字符送
                                                DL, AL
F02C 8AC2     3100        MOV     AL, DL      ; GET CHARACTER INTO AL
F02E 80F448   3101        XOR     AH, 48H     ; FLIP A COUPLE OF BITS  状态字
                                                节第 3, 6 位取反, 得正逻辑表示的状
                                                态字节
F031 EBC2     3102        JMP     B1          ; RETURN FROM ROUTINE  转B1,
                                                准备返回
              3103
              3104 ;……INITIALIZE THE PRINTER PORT  下面程序段初始打印机
                      口
              3105
F033          3106 B8:
F033 50       3107        PUSH    AX          ; SAVE AL
F034 83C202   3108        ADD     DX, 2       ; POINT TO OUTPUT PORT  DX指
                                                向命令输出口
F037 B008     3109        MOV     AL, 8       ; SET INIT LINE LOW  第 2 位的低
                                                电平将引向第 16 引脚, 由它初始打印
                                                机
F039 EE       3110        OUT     DX, AL
F03A B8E803   3111        MOV     AX, 1000      循环次数
F03D          3112 B9:                      ; INIT_LOOP
F03D 48       3113        DEC     AX          ; LOOP FOR RESET TO TAKE  等
                                                待初始完成
F03E 75FD     3114        JNZ     B9          ; INIT_LOOP
F040 B00C     3115        MOV     AL, 0CH     ; NO INTERRUPTS, NON AUTO LF,
                                                INIT HIGH  禁止中断(第 5 位为 0)
                                                初始电平置 1 态(第 2
F042 EE       3116        OUT     DX, AL        位为 1), 非自动换行 (第 1 位为 0),
                                                STROBE 脉冲高电平
F043 EBDB     3117        JMP     B6          ; PRT_STATUS_1  转 B6 取状态
              3118 PRINTER_IO ENDP
              3119 ;…INT 10……
              3120 ; VIDEO_IO  软中断 10—CRT—IO
```

```
3121 ;  THESE ROUTINES PROVIDE THE CRT INTERFACE  本程序
        提供与 CRT 接口诸功能
3122 ;  THE FOLLOWING FUNCTIONS ARE PROVIDED:
3123 ;  (AH) = 0 SET MODE (AL) CONTAINS MODE VALUE  (AH)
                = 0 设置 CRT 模式 AL 中为模式值
3124 ;          (AL) = 0  40×25 BW(POWER ON DEFAULT)  (AL)
                = 0 40×25 单色(开机时已设好)
3125 ;          (AL) = 1  40×25 COLOR  (AL) = 1 40×25 彩色
3126 ;          (AL) = 2  80×25 BW     (AL) = 2 80×25 单色
3127 ;          (AL) = 3  80×25 COLOR  (AL) = 3 80×25 彩色
3128 ;          GRAPHICS MODES  下面为图形模式
3129 ;          (AL) = 4  320×200 COLOR  (AL) = 4 320×200 彩色
3130 ;          (AL) = 5  320×200 BW     (AL) = 5 320×200 单色
3131 ;          (AL) = 6  640×200 BW     (AL) = 6 640×200 单色
3132 ;          CRT MODE = 7  80×25 B&W CARD(USED INTERNAL
                TO VIDEO ONLY)  CRT 模式=7 为 80×25 单色转接
                口
3133 ;          *** NOTE BW MODES OPERATE SAME AS C-
                OLOR MODES, BUT COLOR  上面单色模式的工作情
                况与彩色的相同,就是彩色信号不发出来
3134 ;                  BURST IS NOT ENABLED
3135 ;  (AH) = 1 SET CURSOR TYPE
3136 ;          (CH) = BITS 4-0 = START LINE FOR CURSOR
                        (AH) = 1 设光标类型
3137 ;                  **HARDWARE WILL ALWAYS CAUSE
                        BLINK  (CH)低 5 位内容为光标起始行,CH 第
                        5,6 位为 1 将造成屏幕闪烁或无光标
3138 ;                  ** SETTING BIT 5 OR 6 WILL CAUSE
                        ERRATIC BLINKING
3139 ;                      OR NO CURSOR AT ALL
3140 ;          (CL) = BITS 4-0 = END LINE FOR CURSOR  (CL)
                        低 5 位内容为光标结束行
3141 ;  (AH) = 2 SET CURSOR POSITION  (AH) = 2 设光标位置
3142 ;          (DH, DL) = ROW, COLUMN (0, 0) IS UPPER LEFT
                (DH, DL) = 行列(0, 0)表示屏幕左上角
3143 ;          (BH) = PAGE NUMBER (MUST BE 0 FOR GRAPHICS
                MODES)  (BH) = 页号(图形模式时为 0)
3144 ;  (AH) = 3 READ CURSOR POSITION  (AH) = 3 读光标位置
3145 ;          (BH) = PAGE NUMBER(MUST BE 0 FOR GRAPHICS
```

```
                MODES)  (BH)页号(图形模式时为 0)
3146 ;          ON EXIT (DH, DL) = ROW, COLUMN OF CURRENT
                CURSOR  出口参数: (DH, DL) = 当前光标处的行、列号
3147 ;                  (CH, CL) = CURSOR MODE CURRENTLY
                        SET  (CH, CL) = 光标当前模式
3148 ;  (AH) = 4 READ LIGHT PEN POSITION  (AH) = 4 读光笔位置
3149 ;          ON EXIT:  出口参数
3150 ;          (AH) = 0··LIGHT PEN SWITCH NOT DOWN/NOT
                        TRIGGERED  (AH) = 0 光笔开关未按下或
                        未触发
3151 ;          (AH) = 1··VALID LIGHT PEN VALUE IN REGISTERS
                        (AH) = 1 光笔位置在 DX, CH, BX 中:
3152 ;                  (DH, DL) = ROW, COLUMN OF CHARAC-
                        TER LP POSN  (DH, DL) = 光笔处字符所
                        在行、列
3153 ;                  (CH) = RASTER LINE(0-199)  (CH) = 光笔
                        处光栅行号
3154 ;                  (BX) = PIXEL COLUMN(0-319, 632)  (BX)
                        = 列行
3155 ;  (AH) = 5 SELECT ACTIVE DISPLAY PAGE (VALID ONLY
                FOR ALPHA MODES)  (AH) = 5 选择活动显示页(字
                符模式下有效)
3156 ;          (AL) = NEW PAGE VALUE(0-7 FOR MODES 0 & 1,
                0-3 FOR MODES 2 & 3)  (AL) 为新页号 40×25 模式
                取 0, 1, 2, 3, 4, 5, 6, 或 7, 80×25 模式取 0, 1、2
                或

3157 ;  (AH) = 6 SCROLL ACTIVE PAGE UP  (AH) = 6 上卷活动显示
                页
3158 ;          (AL) = NUMBER OF LINES, INPUT LINES BLANKED
                AT BOTTOM OF WINDOW)  (AL) = 上卷行数
3159 ;                  AL = 0 MEANS BLANK ENTIRE WINDOW
                        AL = 0 表示填空格
3160 ;          (CH, CL) = ROW, COLUMN OF UPPER LEFT COR-
                NER OF SCROLL  (CH, CL) = 上卷左上角行、列
                号
3161 ;          (DH, DL) = ROW, COLUMN OF LOWER RIGHT CO-
                RNER OF SCROLL  (DH, DL) = 上卷右下角行、列
                号
```

```
3162 ;              (BH) = ATTRIBUTE TO BE USED ON BLANK LINE
                    (BH) = 空白行用的属性字节
3163 ;  (AH) = 7 SCROLL ACTIVE PAGE DOWN   (AL) = 7 下卷活动
                    显示页
3164 ;              (AL) = NUMBER OF LINES, INPUT LINES BLANK-
                    ED AT TOP OF WINDOW   参数基本同上卷活动显示
                    页
3165 ;                        AL = 0 MEANS BLANK ENTIRE WINDOW
3166 ;              (CH, CL) = ROW, COLUMN OF UPPER LEFT CORN-
                    ER OF SCROLL
3167 ;              (DH, DL) = ROW, COLUMN OF LOWER RIGHT CO-
                    RNER OF SCROLL
3168 ;              (BH) = ATTRIBUTE TO BE USED ON BLANK LINE
3169 ;
3170 ;  CHARACTER HANDLING ROUTINES  下面为字符模式下的读/
     写操作:
3171 ;
3172 ;  (AH) = 8 READ ATTRIBUTE/CHARACTER AT CURRENT C-
                    URSOR POSITION   (AH) = 8 读当前光标位置处的字符
                    属性码和字符码
3173 ;              (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES
                    ONLY)   (BH) = 显示页号(字符模式下有效)
3174 ;              ON EXIT;  出口参数
3175 ;              (AL) = CHAR READ   (AL) = 字符码
3176 ;              (AH) = ATTRIBUTE OF CHARACTER READ (ALP-
                    HA MODES ONLY)   (AH) = 字符的属性码(字符模式)
3177 ;  (AH) = 9 WRITE ATTRIBUTE/CHARACTER AT CURRENT
                    CURSOR POSITION   (AH) = 9 在当前光标处显示新字
                    符
3178 ;              (BH) = DISPLAY PAGE (VALID FOR ALPHA MOD-
                    ES ONLY)   (BH) = 显示页(字符模式下有效)
3179 ;              (CX) = COUNT OF CHARACTERS TO WRITE
                    (CX) = 新字符个数
3180 ;              (AL) = CHAR TO WRITE   (AL) = 字符码
3181 ;              (BL) = ATTRIBUTE OF CHARACTER (ALPHA)/C-
                    OLOR OF CHAR (GRAPHICS)   (BL) = 字符属性码
                    (字符模式)或字符颜色(图
3182 ;                    SEE NOTE ON WRITE DOT FOR BIT 7 OF
                    BL = 1   形模式), 请参见"写点"(AH = 12) 有关
```

```
;  (AH) = 10 WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION  (AH) = 10 在光标处写一新字符
;            (BH) = DISPLAY PAGE (VALID FOR ALPHA MODES ONLY)  (BH) = 显示页
;            (CX) = COUNT OF CHARACTFRS TO WRITE
;            (CX) = 字符个数
;            (AL) = CHAR TO WRITE  (AL) = 字符码
; FOR READ/WRITE CHARACTER INTERFACE WHILE IN GRAPHICS MODE, THE
;            CHARACTERS ARE FORMED FROM A CHARACTER GENERATOR IMAGE  在图形模式下读/写字符时,字符代码取自系统 ROM
;            MAINTAINED IN THE SYSTEM ROM. ONLY THE 1ST 128 CHARS  区的字符表,该表只记入了 128 个字符的点阵,用户若想使
;            ARE CONTAINED THERE. TO READ/WRITE THE SECOND 128 CHARS  用其它 128 个字符的话，应通过发软中断 1FH 将指
;            THE USER MUST INITIALIZE THE POINTER AT INTERRUPT 1FH  针指向该 128 字符点阵的 1KB 数据表
;            (LOCATION 0007CH) TO POINT TO THE 1K BYTE TABLE CONTAINING
;            THE CODE POINTS FOR THE SECOND 128 CHARS (128—255)
; FOR WRITE CHARACTER INTERFACE IN GRAPHICS MODE, THE REPLICATION FACTOR  在字符模式下写字符时,(CX)的重复次数
;            CONTAINED IN (CX) ON ENTRY WILL PRODUCE VALID RESULTS ONLY  只对一个显示行有效
;            FOR CHARACTERS CONTAINED ON THE SAME ROW. CONTINUATION TO
;            SUCCEEDING LINES WILL NOT PRODUCE CORRECTLY
;
; GRAPHICS INTERFACE  下面为图形模式下的设色板、读/写点操作
; (AH) = 11 SET COLOR PALETTE  (AH) = 11 设置色板
```

```
;       (BH) = PALLETTE COLOR ID BEING SET (0-127)
;            (BH) = 欲置的色板标识(0-127)
;       (BL) = COLOR VALUE TO BE USED WITH THAT
;            COLOR ID  (BL) = 色板标识用到的色彩值
;       NOTE: FOR THE CURRENT COLOR CARD, THIS
;            ENTRY POINT HAS  注意：对于目前使用的
;            彩色 CRT 转接器，上面参数只对 320×200 模
;            式
;                 MEANING ONLY FOR 320×200
;                 GRAPHICS  有意义
;            COLOR ID = 0 SELECTS THE BACKGROUND
;            COLOR(0-15)
;            COLOR ID = 1 SELECTS THE PALLETTE TO
;            BE USED  色板标识 = 0 绿(1)，红(2)，黄 (3)
;            色板
;                 0 = GREEN (1)/RED (2)/YELLOW
;                 (3)  = 1 深兰(1)，品红(2)，白(3)
;                 色板
;                 1 = CYAN (1)/MAGENTA (2)/W-
;                 HITE(3)
;            IN 40×25 OR 80×25 ALPHA MODES, THE
;            VALUE SET FOR  在 40×25 或 80×25 模式
;            下，为色板 0 选取的
;                 PALLETTE COLOR 0 INDICATES
;                 THE BORDER COLOR  色集指定
;                 了边界色(色集值 0～31，其中 16～
;                 31
;                 TO BE USED (VALUES 0-31,
;                 WHERE 16-31 SELECT THE  选
;                 择高亮度背景色)
;                 HIGH INTENSITY BACKGROUND
;                 SET
;  (AH) = 12 WRITE DOT  (AH) = 12 写点
;       (DX) = ROW NUMBER      (DX) = 行号
;       (CX) = COLUMN NUMBER   (CX) = 列号
;       (AL) = COLOR VALUE     (AL) = 色值
;            IF BIT 7 OF AL = 1, THEN THE COLOR
;            VALUE IS EXCLUSIVE  如果(AL)第 7 位
;            为 1，则色值异或入该点
```

```
;                       OR'D WITH THE CURRENT CONTENTS OF
;                       THE DOT  的当前色值
;   (AH) = 13 READ DOT  (AH) = 13 读点
;             (DX) = ROW NUMBER        (DX) = 行号
;             (CX) = COLUMN NUMBER     (CX) = 列号
;             (AL) RETURNS THE DOT READ  (AL) = 读取的色
;             点
;
; ASCII TELETYPE ROUTINE FOR OUTPUT  下面是 ASCII 电传
  输出处理程序的入口参数
;
;   (AH) = 14 WRITE TELETYPE  (AH) = 14 写电传
;             (AL) = CHAR TO WRITE  (AL) = 待写字符
;             (BL) = FOREGROUND COLOR IN GRAPHICS MODE
;             (BL) = 图形式模下的前景色
;             (BH) = DISPLAY PAGE IN ALPHA MODE  (BH) =
;             字符式模下的显示页
;             NOTE··SCREEN WIDTH IS CONTROLLED BY PRE-
;             VIOUS MODE SET  注意:屏幕宽度由以前的模式决定
;
;   (AH) = 15 CURRENT VIDEO STATE  (AH) = 15 取当前CRT状
;             态
;             RETURNS THE CURRENT VIDEO STATE  出口参
;             数
;             (AL) = MODE CURRENTLY SET (SEE AH = 0 FOR
;             EXPLANATION)  (AL) = 当前 CRT 模式 (见 AH = 0
;             的注解)
;             (AH) = NUMBER OF CHARACTER COLUMNS ON
;             SCREEN  (AH) = CRT 屏幕上每行的字符数
;             (BH) = CURRENT ACTIVE DISPLAY PAGE  (BH)
;             = 当前活动显示负号
;
; CS, SS, DS, ES, BX, CX, DX PRESERVED DURING CALL
  调用本软中断后除 CS, SS, DS, ES, CX, DX 内容不变外其余寄存器
; ALL OTHERS DESTROYED  内容均被破坏
;.....
  ASSUME CS:CODE, DS:DATA, ES:VIDEO_RAM

```

```
F045                3243 M1         LABEL  WORD    ; TABLE OF ROUTINES WITHIN
                                                      VIDEO I/O  本软中断各子程序
                                                      入口地址表(共 16 个入口)
F045 FCF0           R 3244          DW     OFFSET   SET_MODE
F047 CFF1           R 3245          DW     OFFSET   SET_CTYPE
F049 F0F1           R 3246          DW     OFFSET   SET_CPOS
F04B 1AF2           R 3247          DW     OFFSET   READ_CURSOR
F04D A9F7           R 3248          DW     OFFSET   READ_LPEN
F04F 30F2           R 3249          DW     OFFSET   ACT_DISP_PAGE
F051 9CF2           R 3250          DW     OFFSET   SCROLL_UP
F053 41F3           R 3251          DW     OFFSET   SCROLL_DOWN
F055 7DF3           R 3252          DW     OFFSET   READ_AC_CURRENT
F057 C3F3           R 3253          DW     OFFSET   WRITE_AC_CURRENT
F059 F6F3           R 3254          DW     OFFSET   WRITE_C_CURRENT
F05B 54F2           R 3255          DW     OFFSET   SET_COLOR
F05D 38F4           R 3256          DW     OFFSET   WRITE_DOT
F05F 27F4           R 3257          DW     OFFSET   READ_DOT
F061 22F7           R 3258          DW     OFFSET   WRITE_TTY
F063 7AF2           R 3259          DW     OFFSET   VIDEO_STATE
  0020                3260 MIL      EQU    S-M1
                      3261
F065                3262 VIDEO_IO   PROC   NEAR     本软中断程序从此开始
F065 FB             3263            STI             ; INTERRUPTS BACK ON  允许
                                                      中断
F066 FC             3264            CLD             ; SET DIRECTION FORWARD
                                                      前向标记
F067 06             3265            PUSH   ES
F068 1E             3266            PUSH   DS       ; SAVE SEGMENT REGISTERS
                                                      保存段寄存器
F069 52             3267            PUSH   DX
F06A 51             3268            PUSH   CX
F06B 53             3269            PUSH   BX
F06C 56             3270            PUSH   SI
F06D 57             3271            PUSH   DI
F06E 50             3272            PUSH   AX       ; SAVE AX VALUE  保存入口参
                                                      数
F06F 8AC4           3273            MOV    AL, AH   ; GET INTO LOW BYTE
F071 32E4           3274            XOR    AH, AH   ; ZERO TO HIGH BYTE
F073 D1E0           3275            SAL    AX, 1    ; ×2 FOR TABLE LOOKUP
```

```
                                                    (AH)送 AL，并将 AX 左移一
                                                    位，得地址表偏差
F075 8BF0          3276         MOV  SI, AX         ; PUT INTO SI FOR BRANCH
F077 3D2000        3277         CMP  AX, MIL        ; TEST FOR WITHIN RANGE
                                                    地址偏差值越出有效范围?
F07A 7204          3278         JB   M2             ; BRANCH AROUND BRAN-
                                                    CH
F07C 58            3279         POP  AX             ; THROW AWAY THE PAR-
                                                    AMETER  越出，AX 退栈
F07D E94701        3280         JMP  VIDEO_RETURN   ; DO NOTHING IF NOT
                                                    IN RANGE  去VIDEO_
                                                    RETURN 准备返回
F080 B84000    R 3281 M2:       MOV  AX, DATA
F083 8ED8          3282         MOV  DS, AX                 DX 指向数据段
F085 B800B8        3283         MOV  AX, 0B800H     ; SEGMENT FOR COLOR CA-
                                                    RD  彩色转接口缓存区首址
                                                    送 AX
F088 8B3E1000  R 3284           MOV  DI, EQUIP_FLAG ; GET EQUIPMENT SET-
                                                    TING
F08C 81E73000      3285         AND  DI, 30H        ; ISOLATE CRT SWITCHES
                                                    EQUIP_FLAG 单元第 6，7 位
                                                    全 1 表示外接单色 CRT
F090 83FF30        3286         CMP  DI, 30H        ; IS SETTING FOR BW CA-
                                                    RD?
F093 7503          3287         JNE  M3                 非单色 CRT 转 M3
F095 B800B0        3288         MOV  AX, 0B000H     ; SEGMENT FOR BW CARD
                                                    单色 CRT，将其缓存区首址送
                                                    AX
F098 8EC0          3289 M3:     MOV  ES, AX         ; SET UP TO POINT AT VI-
                                                    DEO RAM AREAS  缓存区
                                                    首址送ES，建立附加数据段
F09A 58            3290         POP  AX             ; RECOVER VALUE
F09B 8A264900  R 3291           MOV  AH, CRT_MODE   ; GET CURRENT MODE
                                                    INTO AH  当前 CRT 模
                                                    式字节送 AH
F09F 2EFFA445F0 R 3292          JMP  WORD PTR CS:[SI+OFFSET M1]  根据地址
                                                    表转向不同处理子程序
                3293 VIDEO_IO ENDP
                3294 ;.....
```

```
                        3295 ; SET_MODE                    SET_MODE 子程序
                        3296 ;          THIS ROUTINE INITIALIZES THE ATTAC-
                                         HMENT TO  本程序设置显示器工作模式，设置
                                         后的屏幕不显示任何东西
                        3297 ;          THE SELECTED MODE. THE SCREEN IS
                                         BLANKED.
                        3298 ; INPUT  入口参数:
                        3299 ;          (AL) = MODE SELECTED (RANGE 0-9)
                                         (AL)选定的模式(0～9)
                        3300 ; OUTPUT  出口参数:
                        3301 ;          NONE  无
                        3302 ;.....
                        3303
                        3304 ;.....TABLES FOR USE IN SETTING OF MODE
                                         下面是设置模式时用到的参数表
                        3305
F0A4                    3306 VIDEO_PARMS  LABEL  BYTE
                        3307 ;.....INIT_TABLE  显示器工作参数表 (6845 CRT
                                         控制器用)
F0A4 38282D0A1F0619     3308            DB   38H, 28H, 2DH, 0AH, 1FH, 6, 19H
                                         ; SET UP FOR 40×25,  40×25 模式用
F0AB 1C02070607         3309            DB   1CH, 2, 7, 6, 7
F0B0 00000000           3310            DB   0, 0, 0, 0
  0010                  3311 M4         EQU  $-VIDEO_PARMS
                        3312
F0B4 71505A0A1F0619     3313            DB   71H, 50H, 5AH, 0AH, 1FH, 6, 19H
                                         ; SET UP FOR 80×25,  80×25 模式用
F0BB 1C02070607         3314            DB   1CH, 2, 7, 6, 7
F0C0 00000000           3315            DB   0, 0, 0, 0
                        3316
F0C4 38282D0A7F0664     3317            DB   38H, 28H, 2DH, 0AH, 7FH, 6, 64H
                                         ; SET UP FOR GRAPHICS  图形模式
                                           用
F0CB 7002010607         3318            DB   70H, 2, 1, 6, 7
F0D0 00000000           3319            DB   0, 0, 0, 0
                        3320
F0D4 6150520F190619     3321            DB   61H, 50H, 52H, 0FH, 19H, 6, 19H
                                         ; SET UP FOR 80×25 B&W CARD
```

```
                                                    80×25 单色转接器用
F0DB 19020D0B0C         3322            DB      19H, 2, 0DH, 0BH, 0CH
F0E0 00000000           3323            DB      0, 0, 0, 0
                        3324
F0E4                    3325 M5         LABEL   WORD    ; TABLE OF REGEN
                                                        LENGTHS 各模式每
                                                        个显示页所需字节数参
                                                        数表
F0E4 0008               3326            DW      2048    ; 40×25  40×25 2K 字
                                                        节
F0E6 0010               3327            DW      4096    ; 80×25  80×25 4K 字
                                                        节
F0E8 0040               3328            DW      16384   ; GRAPHICS 图形16K
                                                        字节
F0EA 0040               3329            DW      16384   ;
                        3330
                        3331 ; ……COLUMNS
F0EC                    3332 M6         LABEL   BYTE    CRT 列数表
F0EC 2828505028285050   3333            DB      40, 40, 80, 80, 40, 40, 80, 80
                        3334
                        3335 ; ……C_REG_TAB
F0F4                    3336 M7         LABEL   BYTE    ; TABLE OF MOOE
                                                        SETS 模式表
                                                        2CH: 40×25 单色
                                                        2AH: 320×200 彩
                                                        色
F0F4 2C282D292A2E1E29   3337            DB      2CH, 28H, 2DH, 29H,2AH,2EH,
                                                        1EH, 29H ; 28H: 40×25 彩色
                                                        2EH: 320×200 黑
                                                        白(单色)
                        3338            2DH: 80×25 单色 1EH: 640×200 单色
                                        29H: 80×25 彩色 29H: 80×25 单色转接器
F0FC                    3339 SET_MODE   PROC    NEAR
F0FC BAD403             3340            MOV     DX, 03D4H ; ADDRESS OF C-
                                                        OLOR CARD 彩
                                                        色口6845 基址 3D
                                                        4H 送 DX
F0FF B300               3341            MOV     BL, 0   ; MODE SET FOR
                                                        COLOR CARD
```

```
                                                    BL清 0 表示彩色口
F101 83FF30      3342       CMP     DI, 30H         ; IS BW CARD INSTALLED
                                                    外接彩色 CRT?
F104 7507        3343       JNE     M8              ; OK WITH COLOR
F106 B007        3344       MOV     AL, 7           ; INDICATE BW CARD M-
                                                    ODE否, AL 送 7 表示外接
                                                    单色转接口
F108 BAB403      3345       MOV     DX, 03B4H       ; ADDRESS OF BW CARD
                                                    单色口 6845 基址 3B4H 送
                                                    DX
F10B FEC3        3346       INC     BL              ; MODE SET FOR BW CA-
                                                    RD  BL 送 1, 表示单色口
F10D 8AE0        3347 M8:   MOV     AH, AL          ; SAVE MODE IN AH  模
                                                    式字节送 AH
F10F A24900  R   3348       MOV     CRT_MODE, AL    ; SAVE IN GLOBAL VARI-
                                                    ABLE  模式送 CRT_MO-
                                                    DE 单元
F112 89166300 R  3349       MOV     ADDR_6845, DX   ; SAVE ADDRESS OF BA-
                                                    SE  保存 6845 基址
F116 1E          3350       PUSH    DS              ; SAVE POINTER TO DAT-
                                                    A SEGMENT
F117 50          3351       PUSH    AX              ; SAVE MODE
F118 52          3352       PUSH    DX              ; SAVE OUTPUT PORT V-
                                                    ALUE
F119 83C204      3353       ADD     DX, 4           ; POINT TO CONTROL RE-
                                                    GISTER  DX 指向 6845 控
                                                    制寄存器
F11C 8AC3        3354       MOV     AL, BL          ; GET MODE SET FOR C-
                                                    ARD  模式标记送 AL
F11 EE           3355       OUT     DX, AL          ; RESET VIDEO  复位 6845
                                                    彩色口送0, 单色口送 1
F11F 5A          3356       POP     DX              ; BACK TO BASE REGIST-
                                                    ER  6845 基址送 DX
F120 2BC0        3357       SUB     AX, AX          ; SET UP FOR ABSO SEG-
                                                    MENT
F122 8ED8        3358       MOV     DS, AX          ; ESTABLISH VECTOR T-
                                                    ABLE ADDRESSING  DS
                                                    指向向量表首址
                 3359       ASSUME DS: ABS0
```

```
F124 C51E7400   3360        LDS     BX, PARM_PTR    ; GET POINTER TO VI-
                                                      DEO PARMS 取CRT缓
                                                      存区首址指针,送BX
F128 58         3361        POP     AX              ; RECOVER PARMS 取
                                                      回模式字节
                3362        ASSUME  DS:CODE
F129 B91000     3363        MOV     CX, M4          ; LENGTH OF EACH RO-
                                                      W OF TABLE 每种模
                                                      式需参数的个数送CX
F12C 80FC02     3364        CMP     AH, 2           ; DETERMINE WHICH
                                                      ONE TO USE 是0模
                                                      式还是1模式
F12F 7210       3365        JC      M9              ; MODE IS 0 OR 1 是转
                                                      M9
F131 03D9       3366        ADD     BX, CX          ; MOVE TO NEXT ROW
                                                      OF INIT TABLE 加增
                                                      量,指向参数表下一模式
                                                      需参数行
F133 80FC04     3367        CMP     AH, 4               设置2模式或3模式?
F136 7209       3368        JC      M9              ; MODE IS 2 OR 3 是,
                                                      转M9
F138 03D9       3369        ADD     BX, CX          ; MOVE TO GRAPHICS
                                                      ROW OF INIT_TABLE
                                                      加增量
F13A 80FC07     3370        CMP     AH, 7               是模式4,5或6?
F13D 7202       3371        JC      M9              ; MODE IS 4, 5, OR 6
                                                      是,转M9
F13F 03D9       3372        ADD     BX, CX          ; MOVE TO BW CARD
                                                      ROW OF INIT_TABLE
                                                      加增量
                3373
                3374 ;……BX POINTS TO CORRECT ROW OF INITIALIZATION
                        TABLE          BX指向待设置模式的6845工作参数行
                3375
F141            3376 M9:                            ; OUT_INIT
F141 50         3377        PUSH    AX              ; SAVE MODE IN AH
                                                      模式字节送栈
F142 32E4       3378        XOR     AH, AH          ; AH WILL SERVE AS
                                                      REGISTER NUMBER D-
```

```
                                                    URING LOOP  AH 清
                                                    0,它作为 6845 地址寄
                                                    存器
                3379
                3380 ;……LOOP THROUGH TABLE, OUTPUTTTING REG ADD-
                        RESS, THEN VALUE FROM TABLE
                3381
F144            3382 M10:                           ; INIT LOOP
F144 8AC4       3383        MOV     AL, AH          ; GET 6845 REGISTER
                                                    NUMBER  6845 光栅
                                                    寄存器号送AL(共16个
                                                    光栅寄存器，通过一个
                                                    口子访问)
F146 EE         3384        OUT     DX, AL          欲写入光栅寄存器号送
                                                    6845 地址寄存器
F149 42         3385        INC     DX              ; POINT TO DATA PO-
                                                    RT  DX 指向数据口
F148 FEC4       3386        INC     AH              ; NEXT REGISTER V-
                                                    ALUE  (AH)加1，指
                                                    向下一个 6845 寄存器
F14A 8A07       3387        MOV     AL, [BX]        ; GET TABLE VALUE
F14C EE         3388        OUT     DX, AL          ; OUT TO CHIP  参数
                                                    表字节送光栅寄存器
F14D 43         3389        INC     BX              ; NEXT IN TABLE  BX
                                                    指向参数表下一字节
F14E 4A         3390        DEC     DX              ; BACK TO POINTER
                                                    REGISTER  DX指向地
                                                    址寄存器
F14F E2F3       3391        LOOP    M10             ; DO THE WHOLE TA-
                                                    BLE  循环，送完 16 个
                                                    参数
F151 58         3392        POP     AX              ; GET MODE BACK
                                                    模式字节送 AX
F152 1F         3393        POP     DS              ; RECOVER SEGMENT
                                                    VALUE  恢复 DS
                3394        ASSUME  DS:DATA
                3395
                3396 ;……FILL REGEN AREA WITH BLANK  显示缓存区送空
```

格符

```
                        3397
F153 33FF               3398        XOR  DI, DI              ; SET UP POINTER FOR
                                                               REGEN  DI 清0
F155 893E4E00       R   3399        MOV  CRT_START, DI       ; START ADDRESS SA-
                                                               VED IN GLOBAL 起始
                                                               地址送 CRT_START 单
                                                               元
F159 C606620000     R   3400        MOV  ACTIVE_PAGE, 0      ; SET PAGE VALUE 活
                                                               动页号单元 ACTIVE_P-
                                                               AGE 送0
F15E B90020             3401        MOV  CX, 8192            ; NUMBER OF WORDS
                                                               IN COLOR CARD 彩色
                                                               口缓存区容量 (8K 字)送
                                                               CX
F161 80FC04             3402        CMP  AH, 4               ; TEST FOR GRAPHICS
                                                               图形模式?
F164 720C               3403        JC   M12                 ; NO_GRAPHICS_INIT
                                                               字符模式(彩色口)转 M12
F166 80FC07             3404        CMP  AH, 7               ; TEST FOR BW CARD
                                                               是单色口?
F169 7404               3405        JE   M11                 ; BW_CARD_INIT 单
                                                               色口转 M11
F16B 33C0               3406        XOR  AX, AX              ; FILL FOR GRAPHICS
                                                               MODE AX 清 0, 图形模
                                                               式下 CRT 将一片空白
F16D EB06               3407        JMP  SHORT M13           ; CLEAR_BUFFER 转
                                                               M13, 送缓存
F16F                    3408 M11:                            ; BW_CARD_INIT
F16F B90008             3409        MOV  CX, 2048            ; BUFFER SIZE ON BW
                                                               CARD 单色转接口, 其
                                                               缓冲区容量(2K 字)送 CX
F172                    3410 M12:                            ; NO_GRAPHICS_INIT
F172 B82007             3411        MOV  AX, '+'7*256        ; FILL CHAR FOR ALP-
                                                               HA AH 送 00, AL送
                                                               20H 分别表示空白字符的
                                                               属性码和 ASCII
F175                    3412 M13:                            ; CLEAR_BUFFER 字
                                                               符码
```

```
F175 F3              3413      REP  STOSW                ; FILL THE REGEN
                                                           BUFFER WITH B-
                                                           LANKS  空白字符
                                                           填满整个缓存区，显
                                                           示缓存区的段基址在
F176 AB                                                    ES 中
                     3414
                     3415 ;……ENABLE VIDEO AND CORRECT PORT SETTING
                     3416
F177 C70660006700 R  3417      MOV  CURSOR_MODE, 67H ; SET CURRENT CU-
                                                           RSOR MODE  置当
                                                           前光标模式
F17D A04900       R  3418      MOV  AL, CRT_MODE     ; GET THE_MODE
                                                           取回输入时 (AL) 入
                                                           口参数
F180 32E4            3419      XOR  AH, AH           ; INTO AX REGIST-
                                                           ER  清 AH
F182 8BF0            3420      MOV  SI, AX           ; TABLE POINTER,
                                                           INDEXED BY MO-
                                                           DE  模式表偏移量
                                                           (由入口参数 AL 决
                                                           定)送 SI
F184 8B166300     R  3421      MOV  DX, ADDR_6845    ; PREPARE TO OU-
                                                           TPUT TO VIDEO
                                                           ENABLE PORT
                                                           6845 基址送 DX，彩
                                                           色口为 3D4，单色口
                                                           为 3B4
F188 83C204          3422      ADD  DX, 4   DX 指向 6845 模式选择寄存器
F18B 2E8A84F4F0   R  3423      MOV  AL, CS:(SI+OFFSET M7)   取出相应的 6845
                                                           模式字节
F190 EE              3424      OUT  DX, AL           ; SET VIDEO ENA-
                                                           BLE PORT  模式字
                                                           节送 CRT 控制器
F191 A26500       R  3425      MOV  CRT_MODE_SET,AL ; SAVE THAT VAL-
                                                           UE  模式送 CRT_
                                                           MODE_SET 单元保
                                                           存
                     3426
```

```
                        3427 ;……DETERMINE NUMBER OF COLUMNS, BOTH FOR
                                ENTIRE DISPLAY 为以后的显示及电传接口决定CRT
                                列数
                        3428 ;……AND THE NUMBER TO BE USED FOR TTY INT-
                                ERFACE
                        3429
F194 2E8A84ECF0 R       3430        MOV  AL, CS:[SI+OFFSET M6]  从 CRT 列数表相
                                                                应单元取当前模式决定的列数
F199 32E4               3431        XOR  AH, AH
F19B A34A00       R     3432        MOV  CRT_COLS, AX     ; NUMBER OF COLUMNS
                                                            IN THIS SCREEN 列数
                                                            送CRT_COLS 单元保存
                        3433
                        3434 ;……SET CURSOR POSITIONS  下面设光标位置
                        3435
F19E 81E60E00           3436        AND  SI, 0EH          ; WORD OFFSET INTO
                                                            CLEAR LENGTH TABLE
F1A2 2E8B8CE4F0 R       3437        MOV  CX, CS:[SI+OFFSET M5]  ; LENGTH TO
                                                                CLEAR 取清
                                                                0 (送空白符)
                                                                显示缓存页的
                                                                长度并送 CRT
                                                                _LEN单元保
                                                                存
F1A7 890E4C00     R     3438        MOV  CRT_LEN, CX      ; SAVE LENGTH OF CR-
                                                            T_NOT USED FOR BW
F1AB B90800             3439        MOV  CX, 8            ; CLEAR ALL CURSOR
                                                            POSITIONS 所有 8 个显
                                                            示页的光标位置记录单元
                                                            清0
F1AE BF5000             3440        MOV  DI, OFFSET CURSOR_POSN  光标位置记录区
                                                                 始址送 DI
F1B1 1E                 3441        PUSH DS               ; ESTABLISH SEGMENT
F1B2 07                 3442        POP  ES               ; ADDRESSING 建立段
                                                            基
F1B3 33C0               3443        XOR  AX, AX
F1B5 F3                 3444        REP  STOSW            ; FILL WITH ZEROES
                                                            送0
F1B6 AB
```

```
                        3445
                        3446 ;……SET UP OVERSCAN REGISTER
                        3447
F1B7 42                 3448        INC   DX              ; SET OVERSCAN PORT
                                                          TO A DEFAULT  DX指向
                                                          色彩选择寄存器（单色时无
                                                          用）
F1B8 B030               3449        MOV   AL, 30H         ; VALUE OF 30H FOR ALL
                                                          MODES EXCEPT 840×200
                                                          送色彩选择常数 （非640×
                                                          200）
F1BA 803E490006 R       3450        CMP   CRT_MODE, 6     ; SEE IF THE MODE IS
                                                          640×200 BW    640×200
                                                          模式?
F1BF 7502               3451        JNZ   M14             ; IF IT IS'NT 640×200,
                                                          THEN GOTO REGULAR
                                                          否,转M14
F1C1 B03F               3452        MOV   AL, 3FH         ; IF IT IS 640×200, THEN
                                                          PUT IN 3FH   640×200,
                                                          色彩选择常数为 3FH
F1C3 EE                 3453 M14:   OUT   DX, AL          ; OUTPUT THE CORRECT
                                                          VALUE TO 3D9 PORT
                                                          送该常数
F1C4 A26600       R     3454        MOV   CRT_PALLETTE, AL ; SAVE THE VALUE
                                                          FOR FUTURE USE
                                                          色彩常数送 CRT_
                                                          PALLETTE单元保存
                        3455
                        3456 ;……NORMAL RETURN FROM ALL VIDEO RETURNS
                                全部 VIDEO 子程序正常返回处
                        3457
F1C7                    3458 VIDEO_RETURN:
F1C7 5F                 3459        POP   DI
F1C8 5E                 3460        POP   SI
F1C9 5B                 3461        POP   BX
F1CA                    3462 M15:                         ; VIDEO_RETURN_C
F1CA 59                 3463        POP   CX
F1CB 5A                 3464        POP   DX
```

```
F1CC 1F          3465            POP  DS
F1CD 07          3466            POP  ES      ; RECOVER SEGMENTS 恢复寄存
                                                器
F1CE CF          3467            IRET         ; ALL DONE  返主
                 3468 SET_MODE   ENDP
                 3469 ;.....
                 3470 ; SET_CTYPE  SET_CTYPE 置光标类型子程序
                 3471 ;          THIS ROUTINE SETS THE CURSOR VALUE  本
                                  程序置光标值
                 3472 ; INPUT  入口参数:
                 3473 ;          (CX) HAS CURSQR VALUE CH_START LINE,
                                  CL_STOP LINE  (CX) = 光标值, (CH) = 光标起始
                                  行, (CL) = 光标结束行
                 3474 ; OUTPUT  出口参数:
                 3475 ;          NONE  无
                 3476 ;.....
F1CF             3477 SET_CTYPE   PROC  NEAR
F1CF B40A        3478            MOV  AH, 10  ; 6845 REGISTER FOR CURSOR
                                                SET  选 6845 第 10 号光标启始行
                                                寄存器
F1D1 890E6000 R  3479            MOV  CURSOR_MODE, CX  ; SAVE IN DATA
                                                AREA  光标值保
                                                存
F1D5 E80200      3480            CALL M16     ; OUTPUT CX REG  调 M16 子程
                                                序送(CX)
F1D8 E8ED        3481            JMP  VIDEO_RETURN  返回
                 3482
                 3483 ; .....THIS ROUTINE OUTPUTS THE CX REGISTER
                          TO THE 6845 REGS NAMED IN AH  M16 子程
                          序将 (CX) 送(AH)及(AH)+1 号 6845 光栅寄存器
                 3484 下面的注解以光标类型设定为例
F1DA             3485 M16:
F1DA 8B166300 R  3486            MOV  DX, ADDR_6845  ; ADDRESS REGISTER
                                                6845 基址送 DX, 彩色
                                                口 3D4, 单色口 3B4
F1DE 8AC4        3487            MOV  AL, AH ; GET VALUE
F1E0 EE          3488            OUT  DX, AL ; REGISTER SET  10 送 6845 地址
                                                寄存器
F1E1 42          3489            INC  DX      ; DATA REGISTER (DX) = 数据寄
```

```
                                        存器地址
F1E2 8AC5       3490            MOV  AL, CH    ; DATA
F1E4 EE         3491            OUT  DX, AL    送光标起始行参数
F1E5 4A         3492            DEC  DX
F1E6 8AC4       3493            MOV  AL, AH
F1E8 FEC0       3494            INC  AL        ; POINT TO OTHER DATA
                                               REGISTER (AL) = 11, 光标
                                               结束行寄存器号
F1EA EE         3495            OUT  DX, AL    ; SET FOR SECOND REGIS-
                                               TER
F1EB 42         3496            INC  DX
F1EC 8AC1       3497            MOV  AL, CL    ; SECOND DATA VALUE 送
                                               结束行参数
F1EE EE         3498            OUT  DX, AL
F1EF C3         3499            RET            ; ALL DONE
                3500 SET_CTYPE      ENDP  M16 子程序返主
                3501 ;.....................................................
                3502 ; SET_CPOS  SET_CPOS 子程序
                3503 ;          THIS ROUTINE SETS THE CURRENT CURSOR
                                POSITION TO THE  本程序将光标移向屏幕另一
                                位置
                3504 ;          NEW X-Y VALUES PASSED
                3505 ; INPUT    入口参数:
                3506 ;          DX-ROW, COLUMN OF NEW CURSOR  (DX)
                                = 新光标的行、列号
                3507 ;          BH - DISPLAY PAGE OF CURSOR  (BH) = 光标
                                所在显示页
                3508 ; OUTPUT   出口参数:
                3509 ;          CURSOR IS SET AT 6845 IF DISPLAY PAGE
                                IS CURRENT DISPLAY  若光标所在页就是当前
                                显示页,则光标移向新位置
                3510 ;.....................................................
F1F0            3511 SET_CPOS       PROC  NEAR
F1F0 8ACF       3512            MOV  CL, BH
F1F2 32ED       3513            XOR  CH, CH    ; ESTABLISH LOOP COUNT
F1F4 D1E1       3514            SAL  CX, 1     ; WORD OFFSET
F1F6 8BF1       3515            MOV SI, CX     ; USE INDEX REGISTER 新
                                               光标所在页 × 2 送SI、CURSOR
                                               _POSN 指出的光标位置存贮区
```

```
F1F8 89945000  R 3516              MOV  [SI+OFFSET CURSOR_POSN], DX
                                                ; SAVE THE POINTER
F1FC 383E6200  R 3517              CMP  ACTIVE_PAGE, BH   新光标显示为当前
                                                显示页?
F200 7505        3518              JNZ  M17        ; SET_CPOS_RETURN
F202 8BC2        3519              MOV  AX, DX     ; GET ROW/COLLMN TO AX
                                                是,新光标坐标送 AX
F204 E80200      3520              CALL M18        ; CURSOR_SET  设光标
F207             3521 M17:                       ; SET_CPOS_RETURN
F207 EBBE        3522              JMP  VIDEO_RETURN  返回
                 3523 SET_CPOS     ENDP
                 3524
                 3525 ; ..........SET CURSOR POSITION, AX HAS ROW/COL-
                                   UMN FOR CURSOR  下面程序段设置新光标位置
                 3526
F209             3527 M18          PROC NEAR
F209 E87F00      3528              CALL POSITION ; DETERMINE LOCATION IN
                                                REGEN BUFFER  取新光标
                                                在显示缓区中的字节偏移,它
                                                相对于页字节偏差加上页始
                                                址
F20C 8BC8        3529              MOV  CX, AX
F20E 030E4E00  R 3530              ADD  CX, CRT_START ; ADD IN THE ST-
                                                ART ADDRESS FOR
                                                THIS PAGE
F212 D1F9        3531              SAR  CX, 1      ; DIVIDE BY 2 FOR CHAR
                                                ONLY COUNT
F214 B40E        3532              MOV  AH, 14     ; REGISTER NUMBER FOR
                                                CURSOR  第14号是光标地
                                                址寄存器
LOC OBJ          LINE  SOURCE
F216 E8C1FF      3533              CALL M16        ; OUTPUT THE VALUE TO
                                                THE 6845  送新光标坐标
F219 C3          3534              RET             返回
                 3535 M18          ENDP
                 3536 ;.....................................................
                 3537 ; READ_CURSOR  READ_CURSOR 读光标子程序
                 3538 ;            THIS ROUTINE READS THE CURRENT CUR-
                                   SOR VALUE FROM THE  本程序读取光标值,经
```

```
                                整理后送调用程序
                3539 ;          6845, FORMATS IT, AND SENDS IT BACK TO
                                THE CALLER
                3540 ; INPUT 入口参数:
                3541 ;          BH - PAGE OF CURSOR  (BH) = 光标所在页
                3542 ; OUTPUT  出口参数:
                3543 ;          DX - ROW, COLUMN OF THE CURRENT CURSOR
                                POSITION                    (DX) = 当前光标坐标
                3544 ;          CX - CURRENT CURSOR MODE    (CX) = 当前光
                                标模式
                3545 ;.....
F21A            3546 READ_CURSOR  PROC  NEAR
F21A 8ADF       3547              MOV  BL, BH
F21C 32FF       3548              XOR  BH, BH
F21E D1E3       3549              SAL  BX, 1   ; WORD OFFSET  页的字偏差
F220 8B975000  R 3550             MOV  DX, (BX + OFFSET CURSOR_POSN)  取该
                                                    页光标坐标
F224 8B0E6000  R 3551             MOV  CX, CURSOR_MODE  取光标模式
F228 5F         3552              POP  DI
F229 5E         3553              POP  SI
F22A 5B         3554              POP  BX
F22B 58         3555              POP  AX    ; DISCARD SAVED CX AND DX
                                              二次退栈 POP AX 操作,扔掉原来
                                              的 CX, DX 内容
F22C 58         3556              POP  AX
F22D 1F         3557              POP  DS
F22E 07         3558              POP  ES
F22F CF         3559              IRET          返回
                3560 READ_CURSOR  ENDP
                3561 ;.....
                3562 ; ACT_DISP_PAGE  ACT_DISP_PAGE 子程序
                3563 ;          THIS ROUTINE SETS THE ACTIVE DISPLAY P-
                                AGE, ALLOWING  本子程序设定新显示页
                3564 ;          THE FULL USE OF THE RAM SET ASIDE FOR
                                THE VIDEO ATTACHMENT
                3565 ; INPUT  入口参数:
                3566 ;          AL HAS THE NEW ACTIVE DISPLAY PAGE
                                (AL) = 新页号
                3567 ; OUTPUT  出口参数:
```

```
                    3568 ;          THE 6845 IS RESET TO DISPLAY THAT PA-
                                    GE  6845复位,显示新页
                    3569 ;.....
F230                3570 ACT_DISP_PAGE  PROC  NEAR
F230 A26200    R    3571            MOV  ACTIVE_PAGE,AL ; SAVE ACTIVE
                                                          PAGE VALUE
                                                          置新活动页
F233 8B0E4C00  R    3572            MOV  CX, CRT_LEN    ; GET SAVED L-
                                                          ENGTH OF REG-
                                                          EN BUFFER  取
                                                          显示页占缓存区字
                                                          节数
F237 98             3573            CBW                 ; CONVERT AL TO
                                                          WORD  扩展成字
F238 50             3574            PUSH AX             ; SAVE PAGE V-
                                                          ALUE
F239 F7E1           3575            MUL  CX             ; DISPLAY  PAGE
                                                          TIMES  REGEN
                                                          LENGTH  页号乘
                                                          页缓存区长得该页
                                                          在缓存区中的的位
                                                          置
F23B A34E00    R    3576            MOV  CRT_START, AX  ; SAVE START A-
                                                          DDRESS FOR LA-
                                                          TER REQUIREM-
                                                          ENTS  位置值送
                                                          CRT_START保存
F23E 8BC8           3577            MOV  CX, AX         ; START ADDRESS
                                                          TO CX  启始地址
                                                          送 CX
F240 D1F9           3578            SAR  CX, 1          ; DIVIDE BY 2 FOR
                                                          6845  HANDLING
F242 B40C           3579            MOV  AH, 12         ; 6845  REGISTER
                                                          FOR START AD-
                                                          DRESS  (AH) =
                                                          12, 6845起始地址
                                                          寄存器号
F244 E893FF         3580            CALL M16            送启始地址
F247 5B             3581            POP  BX             ; RECOVER PAGE
```

```
                                                    VALUE  取回页号
F248 D1E3       3582            SAL  BX, 1          ; ×2 FOR WORD OFF-
                                                    SET  X2
F24A 8B875000 R 3583            MOV  AX, (BX+OFFSET CURSOR_POSN)
                                                    ; GET CURSOR FOR
                                                    THIS PAGE  取该页
                                                    的光标位置坐标
F24E E8B8FF     3584            CALL M18            ; SET THE CURSOR
                                                    POSITION  转M18送
                                                    光标值
F251 E973FF     3585            JMP  VIDEO_RETURN   返主
                3586 ACT_DISP_PAGE  ENDP
                3587 ;.....
                3588 ; SET COLOR  SET_COLOR 子程序
                3589 ;          THIS ROUTINE WILL ESTABLISH THE BAC-
                                KGROUND COLOR, THE OVERSCAN COLOR
                                本子程序为中分辨率图形模式设置象图的背景色、
                                前景色
                3590 ;          AND THE FOREGROUND COLOR SET FOR
                                MEDIUM RESOLUTION GRAPHICS
                3591 ; INPUT  入口参数:
                3592 ;          (BH)HAS COLOR ID  (BH)色彩标识
                3593 ;                  IF BH=0, THE BACKGROUND CO-
                                        LOR VALUE IS SET  (BH)=0 根据
                                        BL 低5位(值在0～31间)设置背景色
                3594 ;                          FROM THE LOW BITS OF
                                                BL (0—31)
                3595 ;                  IF BH=1, THE PALLETTE SELEC-
                                        TION IS MADE  (BH)=1 根据BL第
                                        0位设色板
                3596 ;                          BASED ON THE LOW BIT
                                                OF BL:
                3597 ;                              0=GREEN, RED,
                                                    YELLOW FOR CO-
                                                    LORS 1, 2, 3
                                                    (BL)=0 取第1色板
                3598 ;                              1=BLUE, CYAN,
                                                    MAGENTA FOR C-
                                                    OLORS 1, 2, 3
```

```
                                                    (BL) = 1 取第 2 色板
                3599 ;          (BL)HAS THE COLOR VALUE TO BE USED
                3600 ; OUTPUT   出口参数:
                3601 ;          THE COLOR SELECTION IS UPDATED  按要求修
                                改 CRT 色彩
                3602 ;.....................................................
F254            3603 SET_COLOR  PROC  NEAR
F254 8B166300 R 3604            MOV  DX, ADDR_6845  ; I/O PORT FOR PAL-
                                                      ETTE
F258 83C205     3605            ADD  DX, 5          ; OVERSCAN PORT
                                                      DX 指向 6845 色彩选择
                                                      寄存器
F25B A06600   R 3606            MOV  AL, CRT_PALLETTE
                                                    ; GET THE CURRENT
                                                      PALLETTE VALUE
                                                      当前色板值送 AL
F25E 0AFF       3607            OR   BH, BH         ; IS THIS COLOR 0?
                                                      (BH)为0表示置背景色
F260 750E       3608            JNZ  M20            ; OUTPUT COLOR 1
                                                      非置背景色操作转 M20
                3609
                3610 ; ..........HANDLE COLOR 0 BY SETTING THE BACKGR-
                                OUND COLOR          下面置背景色, (BH) = 0
                3611
F262 24E0       3612            AND  AL, 0E0H       ; TURN OFF LOW 5
                                                      BITS OF CURRENT
                                                      AL 低 5 位清 0, 其第 5
                                                      位是色板号{0第 1 色板
                                                                1第 2 色板
F264 80E31F     3613            AND  BL, 01FH       ; TURN OFF HIGH 3
                                                      BITS OF INPUT VA-
                                                      LUE  BL 高 3 位清 0,
                                                      其第0～4位是背景色号
F267 0AC3       3614            OR   AL, BL         ; PUT VALUE INTO
                                                      REGISTER  混合 AL,
                                                      BL
F269            3615 M19:                           ; OUTPUT THE PALL-
                                                      ETTE
```

```
F269 EE          3616            OUT  DX, AL      ; OUTPUT COLOR SELECT-
                                                    ION TO 3D9 PORT  背景
                                                    色字节送 3D9 色彩选择寄存
                                                    器
F26A A26600    R 3617            MOV  CRT_PALLETTE, AL
                                                  ; SAVE THE COLOR VAL-
                                                    UE  色板值送 CRT_PAL-
                                                    LETTE 单元
F26D E957FF      3618            JMP  VIDEO_RETURN  返回
                 3619
                 3620 ; ..........HANDLE COLOR 1 BY SELECTING THE PA-
                                  LLETTE TO BE USED  下面设色板
                 3621
F270             3622 M20:
F270 24DF        3623            AND  AL, 0DFH    ; TURN OFF PALLETTE
                                                    SELECT BIT  暂将 AL 第
                                                    5 位复位,选 1 号色板
F272 D0EB        3624            SHR  BL, 1       ; TEST THE LOW ORDER
                                                    BIT OF BL  要求选 1 号还
                                                    是 2 号色板?
F274 73F3        3625            JNC  M19         ; ALREADY DONE, 1 号色
                                                    板,转 M19
F276 0C20        3626            OR   AL, 20H     ; TURN ON PALLETTE S-
                                                    ELECT BIT  第 5 位置 1,
                                                    选 2 号色板
F278 EBEF        3627            JMP  M19         ; GO DO IT  转 M19 送色板
                                                    值
                 3628 SET_COLOR      ENDP  返回
                 3629 ;.....................................................
                 3630 ; VIDEO STATE  VIDEO STATE 子程序
                 3631 ;  RETURNS THE CURRENT VIDEO STATE IN AX  本子
                         程序在 AX 返回 CRT 当前状态
                 3632 ;  AH = NUMBER OF COLUMNS ON THE SCREEN  出口
                         参数:
                 3633 ;  AL = CURRENT VIDEO MODE  (AH) = 每帧图象的列数,
                         (AL) = 当前 CRT 状态(模式)
                 3634 ;  BH = CURRENT ACTIVE PAGE  (BH) = 当前活动显示页
                 3635 ;.....................................
F27A             3636 VIDEO_STATE  PROC  NEAR
```

```
F27A 8A264A00 R 3637            MOV  AH, BYTE PTR CRT__COLS
                                                    ; GET NUMBER OF
                                                      COLUMNS  取列
                                                      数
F27E A04900    R 3638           MOV  AL, CRT__MODE   ; CURRENT MODE
                                                      取模式
F281 8A3E6200  R 3639           MOV  BH,ACTIVE__PAGE ; GET  CURRENT
                                                      ACTIVE PAGE
                                                      取活动页号
F285 5F          3640           POP  DI              ; RECOVER REG-
                                                      ISTERS
F286 5E          3641           POP  SI              ;
F287 59          3642           POP  CX              ; DISCARD  SAV-
                                                      ED BX
F288 E93FFF      3643           JMP  M15             ; RETURN TO C-
                                                      ALLER
                                                      返回
                 3644 VIDEO__STATE   ENDP
                 3645 ; ..........................................
                 3646 ; POSITION  POSITION 定位子程序
                 3647 ;          THIS SERVICE ROUTINE CALCULATES THE
                                 REGEN BUFFER ADDRESS  本子程序计算字符
                                 模式下某字符在显示页缓存区中的偏差
                 3648 ;          OF A CHARACTER IN THE ALPHA MODE
                 3649 ; INPUT    入口参数:
                 3650 ;          AX * ROW, COLUMN POSITION  (AX) = 字符在
                                 屏幕上的坐标(行,列)
                 3651 ; OUTPUT   出口参数:
                 3652 ;          AX = OFFSET OF CHAR POSITION IN REGEN
                                 BUFFER  (AX) = 字符在页缓存区中的偏差
                 3653 ; ..........................................
F28B             3654 POSITION  PROC NEAR
F28B 53          3655           PUSH BX              ; SAVE REGISTER
                                                      保存 BX
F28C 8BD8        3656           MOV  BX, AX
F28E 8AC4        3657           MOV  AL, AH          ; ROWS TO AL
                                                      行坐标送 AL
F290 F6264A00  R 3658           MUL  BYTE PTR CRT__COLS
                                                    ; DETERMINE BY-
                                                      TES TO ROW · (AL)
```

```
                                        ×每行字节数 ⇒ AX
F294 32FF    3659          XOR  BH, BH  清 BH
F296 03C3    3660          ADD  AX, BX  ; ADD IN COLUMN VALUE  加进
                                          列坐标
F298 D1E0    3661          SAL  AX, 1   ; ×2 FOR ATTRIBUTE BYTES
                                          (AX)×2，算进属性字节
F29A 5B      3662          POP  BX
F29B C3      3663          RET  返回
             3664 POSITION      ENDP
             3665 ; ..........
             3666 ; SCROLL UP  SCROLL UP 上卷子程序
             3667 ;         THIS ROUTINE MOVES A BLOCK OF CHARACTERS
                            UP  本程序将一个字符块上卷
             3668 ;         ON THE SCREEN
             3669 ; INPUT  入口参数：
             3670 ;         (AH) = CURRENT CRT MODE  (AH) = CRT 模式
             3671 ;         (AL) = NUMBER OF ROWS TO SCROLL
                            (AL) = 上卷行数
             3672 ;         (CX) = ROW/COLUMN OF UPPER LEFT CORNER
                            (CX) = 左上角坐标  (行/列)
             3673 ;         (DX) = ROW/COLUMN OF LOWER RIGHT CORNER
                            (DX) = 右下角坐标  (行/列)
             3674 ;         (BH) = ATTRIBUTE TO BE USED ON BLANKED LINE
                            (BH) = 空行用属性符
             3675 ;         (DS) = DATA SEGMENT  (DS) = 数据段基址
             3676 ;         (ES) = REGEN BUFFER SEGMENT  (ES) = 显示页缓
                            存区基址
             3677 ; OUTPUT  出口参数：
             3678 ;         NONE…THE REGEN BUFFER IS MODIFIED  无，页
             缓存区被修改过了
             3679 ; ..........
             3680          ASSUME CS: CODE, DS: DATA, ES: DATA
F29C         3681 SCROLL_UP  PROC  NEAR
F29C 8AD8    3682          MOV  BL, AL  ; SAVE LINE COUNT IN BL  上卷
                                          行数送 BL
F29E 80FC04  3683          CMP  AH, 4   ; TEST FOR GRAPHICS MODE  图
                                          形模式?
F2A1 7208    3684          JC   N1      ; HANDLE SEPARATELY  字符模式
```

```
                                                    转 N1
F2A3 80FC07   3685        CMP   AH, 7              ; TEST FOR BW CARD  单
                                                    色转接口?
F2A6 7403     3686        JE    N1                 单色转接口转 N1
F2A8 E9F301   3687        JMP   GRAPHICS_UP
F2AB          3688 N1:                             ; UP_CONTINUE  字符模式
                                                    (彩色或单色转接口)
F2AB 53       3689        PUSH  BX                 ; SAVE FILL ATTRIBUTE IN
                                                    BH  空白行属性符送栈
F2AC 8BC1     3690        MOV   AX, CX             ; UPPER LEFT POSITION
                                                    左上角坐标送 AX
F2AE E83900   3691        CALL  SCROLL_POSITION    ; DO SETUP FOR SCROLL
F2B1 7433     3692        JZ    N7                 ; BLANK_FIELD  上卷行数
                                                    =0说明填空格
F2B3 03F0     3693        ADD   SI, AX             ; FROM ADDRESS  左上角字
                                                    符地址加上对应数上移行占的
                                                    字节数，得起始上移行第一字
                                                    符地址
F2B5 8AE6     3694        MOV   AH, DH             ; ×ROWS IN BLOCK  字符块
                                                    行数送 AH
F2B7 2AE3     3695        SUB   AH, BL             ; ×ROWS TO BE MOVED
                                                    字符块行数减上移行数=实际
                                                    应上移的行数
F2B9          3696 N2:                             ; ROW_LOOP
F2B9 E87500   3697        CALL  N10                ; MOVE ONE ROW  上移一行
F2BC 03F5     3698        ADD   SI, BP
F2BE 03FD     3699        ADD   DI, BP             ; POINT TO NEXT LINE IN
                                                    BLOCK  移出，送入指针加上
                                                    一行字符的偏差，从而指向下
                                                    一行
F2C0 FECC     3700        DEC   AH                 ; COUNT OF LINES TO MO-
                                                    VE  上移完成?
F2C2 75F5     3701        JNZ   N2                 ; ROW_LOOP  否，转 N2
F2C4          3702 N3:                             ; CLEAR_ENTRY
F2C4 58       3703        POP   AX                 ; RECOVER ATTRIBUTE IN
                                                    AH  空白行属性送 AH
F2C5 B020     3704        MOV   AL, ''             ; FILL WITH BLANKS  空白
                                                    字符 ASC II 码送 AL
F2C7          3705 N4:                             ; CLEAR_LOOP
```

```
F2C7 E87000        3706          CALL    N11              ; CLEAR THE ROW
                                                          填一行空白符
F2CA 03FD          3707          ADD     DI, BP           ; POINT TO NEXT
                                                          LINE  DI 指向下一
                                                          行在缓存区中的始址
F2CC FECB          3708          DEC     BL               ; COUNTER OF LIN-
                                                          ES TO SCROLL 行
                                                          数计数器减 1
F2CE 75F7          3709          JNZ     N4               ; CLEAR_LOOP 清
                                                          行完毕?
F2D0               3710 N5:                               ; SCROLL_END
F2D0 B84000    R   3711          MOV     AX, DATA         ; GET LOCATION
F2D3 8ED8          3712          MOV     DS, AX           DS 指向数据段
F2D5 803E490007 R  3713          CMP     CRT_MODE, 7      ; IS THIS THE BLA-
                                                          CK AND WHITE
                                                          CARD 外接单色转
                                                          接口?
F2DA 7407          3714          JE      N6               ; IF SO, SKIP THE
                                                          MODE RESET 否,
                                                          转 N6
F2DC A06500    R   3715          MOV     AL, CRT_MODE_SET ; GET THE
                                                          VALUE OF
                                                          THE MODE
                                                          SET 取CRT
                                                          模式字节
F2DF BAD803        3716          MOV     DX, 03D8H        ; ALWAYS SET CO-
                                                          LOR CARD PORT
F2E2 EE            3717          OUT     DX, AL           模式字送 6845 模式寄存器
F2E3               3718 N6:                               ; VIDEO_RET_HE-
                                                          RE
F2E3 E9E1FE        3719          JMP     VIDEO_RETURN     准备返回
F2E6               3720 N7:                               ; BLANK_FIELD
F2E6 8ADE          3721          MOV     BL, DH           ; GET ROW COUNT
                                                          上卷行数=0 字符块
                                                          中填空格,BL 中为待
                                                          填空白符的行数
F2E8 EBDA          3722          JMP     N3               ; GO CLEAR THAT
                                                          AREA 转 N3 填空
                                                          白符
```

```
                        3723 SCROLL_UP    ENDP
                        3724
                        3725 ; ..... HANDLE COMMON SCROLL SET UP HERE
                        3726
F2EA                    3727 SCROLL_POSITION PROC  NEAR
F2EA 803E490002 R       3728        CMP   CRT_MODE, 2  ; TEST FOR SPECIAL
                                                         CASE HERE  40×25
                                                         模式?
F2EF 7219               3729        JB    N9           ; HAVE TO HANDLE
                                                         80×25 SEPARATELY
                                                         是,转 N9
F2F1 803E490003  R      3730        CMP   CRT_MODE, 3  80×25 模式?
F2F6 7712               3731        JA    N9           否转 N9
                        3732
                        3733 ; ..... 80×25 COLOR CARD SCROLL
                                     80×25 模式(彩色口)上卷
                        3734
F2F8 52                 3735        PUSH  DX     (DX)右下角坐标,(CX)左上角坐标
F2F9 BADA03             3736        MOV   DX, 3DAH   ; GUARANTEED TO BE
                                                         COLOR CARD HERE
                                                         SDAH 是6845状态寄存
                                                         器地址
F2FC 50                 3737        PUSH  AX
F2FD                    3738 N8:                     ; WAIT_DISP_ ENA-
                                                         BLE
F2FD EC                 3739        IN    AL, DX     ; GET PORT  读状态
F2FE A808               3740        TEST  AL, 8      ; WAIT FOR VERTIC-
                                                         AL RETRACE  状态
                                                         字节第3位为1表示有
                                                         视频信号输出,不处回
                                                         扫期
F300 74FB               3741        JZ    N8         ; WAIT_DISP_ENA-
                                                         BLE
F302 B025               3742        MOV   AL, 25H    非水平回扫期
F304 BAD803             3743        MOV   DX, 03D8H  模式字节第3位送0,
                                                         禁止视频输出
F307 EE                 3744        OUT   DX, AL     ; TURN OFF VIDEO
F308 58                 3745        POP   AX         ; DURING VERTICAL
                                                         RETRACE  现处水平
```

```
                                                    回扫期
F309 5A            3746      POP   DX               (AX)为左上角坐标
F30A E87EFF        3747 N9;  CALL  POSITION         ; CONVERT TO REGEN
                                                    POINTER  取该点在页
                                                    缓存区的偏差
F30D 03064E00  R   3748      ADD   AX, CRT_START   ; OFFSET OF ACTIVE
                                                    PAGE  加页基址
F311 8BF8          3749      MOV   DI, AX          ; TO ADDRESS FOR SC-
                                                    ROLL
F313 8BF0          3750      MOV   SI, AX          ; FROM ADDRESS FOR
                                                    SCROLL  在显示缓存中
                                                    的绝对地址送 DI, SI (左
                                                    上角字符)
F315 2BD1          3751      SUB   DX, CX          ; DX = ×ROWS, ×COLS
                                                    IN BLOCK  字符块的行
                                                    数,列数送 DX
F317 FEC6          3752      INC   DH
F319 FEC2          3753      INC   DL              ; INCREMENT FOR 0 O-
                                                    RIGIN  调整 DX
F31B 32ED          3754      XOR   CH, CH          ; SET HIGH BYTE OF
                                                    COUNT TO ZERO
                                                    清 CH
F31D 8B2E4A00  R   3755      MOV BP, CRT_COLS      ; GET NUMBER OF CO-
                                                    LUMNS IN DISPLAY
F321 03ED          3756      ADD   BP, BP          ; TIMES 2 FOR ATTRIB-
                                                    UTE BYTE  (BP) = 每行
                                                    字符数×2,  即实际每行
                                                    占的字节数
F323 8AC3          3757      MOV   AL, BL          ; GET LINE COUNT  上
                                                    卷行数送 AL
F325 F6264A00  R   3758      MUL   BYTE PTR CRT_COLS ; DETERMINE OF-
                                                    FSET TO FROM
                                                    ADDRESS  (AL)
                                                    ×每行字节数
F329 03C0          3759      ADD   AX, AX          ; ×2 FOR ATTRIBUTE
                                                    BYTE  (AX)×2 算进属
                                                    性字符字节
F32B 06            3760      PUSH  ES              ; ESTABLISH ADDRESS-
```

```
                                                    ING TO REGEN BUFFER
F32C 1F         3761            POP     DS          ; FOR BOTH POINTERS  ES, DS
                                                    都指向显示缓存首址
F32D 80FB00     3762            CMP     BL, 0       ; 0 SCROLL MEANS BLANK FI-
                                                    ELD  上卷行数为0?
F330 C3         3763            RET                 ; RETURN WITH FLAGS SET
                                                    返回
                3764 SCROLL_POSITION ENDP
                3765
                3766 ; ……  MOVE_ROW  MOVE_ROW 移送一行字符之字符码和
                                属性码子程序
F331            3767 N10        PROC    NEAR
F331 8ACA       3768            MOV     CL, DL      ; GET * OF COLS TO MOVE  将
                                                    移送列数送 CL, 作计数器用
F333 56         3769            PUSH    SI
F334 57         3770            PUSH    DI          ; SAVE START ADDRESS  保存
                                                    SI, DI
F335 F3         3771            REP     MOVSW       ; MOVE THAT LINE ON SCREEN
                                                    移送一行
F336 A5
F337 5F         3772            POP     DI
F338 5E         3773            POP     SI          ; RECOVER ADDRESSES  恢复
                                                    SI, DI
F339 C3         3774            RET                 返回
                3775 N10        ENDP
                3776
                3777 ; ……  CLEAR_ROW  CLEAR_ROW 送一行空白 ASC II 码
                                                    和属性码子程序
F33A            3778 N11        PROC    NEAR
F33A 8ACA       3779            MOV     CL, DL      ; GET # COLUMNS TO CLEAR
                                                    欲清除行的列数送CL, 作计数器用
F33C 57         3780            PUSH    DI
F33D F3         3781            REP     STOSW       ; STORE THE FILL CHARACTER
                                                    填清空白字符码和属性码
F33E AB
F33F 5F         3782            POP     DI          返回
F340 C3         3783            RET
                3784 N11        ENDP
                3785 ; ……
```

```
                3786 ; SCROLL_DOWN  SCROLL_DOWN 下卷子程序
                3787 ;          THIS ROUTINE MOVES THE CHARACTERS WITHIN
                                 A DEFINED 本子程序将一个字符块下卷,与SCROLL_UP
                                 基本相仿,故将注释简化了。
                3788 ;          BLOCK DOWN ON THE SCREEN, FILLING THE TOP
                                 LINES
                3789 ;          WITH A DEFINED CHARACTER
                3790 ; INPUT  入口参数:
                3791 ;          (AH) = CURRENT CRT MODE  (AH) = 当前 CRT 模式
                3792 ;          (AL) = NUMBER OF LINES TO SCROLL
                                 (AL) = 下卷行数
                3793 ;          (CX) = UPPER LEFT CORNER OF REGION
                                 (CX) = 左上角坐标
                3794 ;          (DX) = LOWER RIGHT CORNER OF REGION
                                 (DX) = 右下角坐标
                3795 ;          (BH) = FILL CHARACTER  (BH) = 填充字符
                3796 ;          (DS) = DATA SOGMENT  (DS) = 数据段首址
                3797 ;          (ES) = REGEN SEGMENT(ES) = 显示缓存区首址
                3798 ; OUPUT  出口参数:
                3799 ;          NONE…SCREEN IS SCROLLED 无,字符块下移了
                3800 ; ……
F341            3801 SCROLL_DOWN  PROC  NEAR
F341 FD         3802          STD                ; DIRECTION FOR SCROLL DO-
                                                   WN  置逆向标记
F342 8AD8       3803          MOV   BL, AL       ; LINE COUNT TO BL  下移行数
                                                   送 BL
F344 80FC04     3804          CMP   AH, 4        ; TEST FOR GRAPHICS  图形模
                                                   式?
F347 7208       3805          JC    N12                否
F349 80FC07     3806          CMP   AH, 7        ; TEST FOR BW CARD  外接单
                                                   色转接口?
F34C 7403       3807          JE    N12          是,转 N12
F34E E9A601     3808          JMP   GRAPHICS_DOWN  字符模式(彩色或单色转
                                                   接口)
F351            3809 N12:                        ; CONTINUE_DOWN
F351 53         3810          PUSH  BX           ; SAVE ATTRIBUTE IN BH  属
                                                   性符送 BH
F352 8BC2       3811          MOV   AX, DX       ; LOWER RIGHT CORNER  右下
                                                   角坐标送 AX
```

```
F354 E893FF   3812          CALL   SCROLL_POSITION  ; GET REGEN LOCATION
F357 7420     3813          JZ     N16               下卷行数=0 说明只填空格
F359 2BF0     3814          SUB    SI, AX            ; SI IS FROM ADDRESS
                                                     SI
F35B 8AE6     3815          MOV    AH, DH            ; GET TOTAL * ROWS
F35D 2AE3     3816          SUB    AH, BL            ; COUNT TO MOVE IN
                                                     SCROLL 字符块行数一下
                                                     移行数=实际下移的行数
F35F          3817 N13:
F35F E8CFFF   3818          CALL   N10               ; MOVE ONE ROW 下移
                                                     一行
F362 2BF5     3819          SUB    SI, BP
F364 2BFD     3820          SUB    DI, BP            SI, DI 减去一行字符占的
                            一行                     字节数,指向上
F366 FECC     3821          DEC    AH
F368 75F5     3822          JNZ    N13               下移字符块完毕?
F36A          3823 N14:
F36A 58       3824          POP    AX                ; RECOVER ATTRIBUTE
                                                     IN AH 下移完毕,送空
                                                     白符的 ASC II 码和属性
                                                     码
F36B B020     3825          MOV    AL, ''
F36D          3826 N15:
F36D E8CAFF   3827          CALL   N11               ; CLEAR ONE ROW 填
                                                     一行空白符
F370 2BFD     3828          SUB    DI, BP            ; GO TO NEXT ROW
                                                     指向上一行
F372 FECB     3829          DEC    BL
F374 75F7     3830          JNZ    N15               填行完毕?
F376 E957FF   3831          JMP    N5                ; SCROLL_END 完毕,
                                                     转 N5 退出 SCROLL_D-
                                                     OWN 子程序
F379          3832 N16:
F379 8ADE     3833          MOV    BL, DH  字符块全填空白符,字符块行数送 BL
LOC OBJ       LINE  SOURCE
F37B EBED     3834          JMP    N14               转 N14 填空白符
              3835 SCROLL_DOWN  ENDP
              3836 ; ..........
              3837 ; READ_AC_CURRENT  READ_CURRENT 子程序
```

```
                        3838 ;          THIS ROUTINE READS THE ATTRIBUTE AND
                                        CHARACTER AT THE CURRENT · 本子程序将光
                                        标处字符的 ASC II 码和属性码返回给调用程序
                        3839 ;          CURSOR POSITION AND RETURNS THEM TO
                                        THE CALLER
                        3840 ; INPUT    入口参数:
                        3841 ;          (AH) = CURRENT CRT MODE   (AH) = 当前 CRT
                                        模式
                        3842 ;          (BH) = DISPLAY PAGE (ALPHA MODES ONLY)
                                        (BH) = 显示页(只用于字符模式)
                        3843 ;          (DS) = DATA SEGMENT   (DS) = 数据段基址
                        3844 ;          (ES) = REGEN SEGMENT   (ES) = 显示缓存区基址
                        3845 ; OUTPUT   出口参数:
                        3846 ;          (AL) = CHAR READ   (AL) = 字符 ASC II 码
                        3847 ;          (AH) = ATTRIBUTE READ   (AH) = 字符属性码
                        3848 ; ..........................................
                        3849            ASSUME CS:CODE, DS:DATA, ES:DATA
F37D                    3850 READ_AC_CURRENT PROC  NEAR
F37D 80FC04             3851            CMP     AH, 4           ; IS THIS GRAPHICS
                                                                  图形模式?
F380 7208               3852            JC      P1
F382 80FC07             3853            CMP     AH, 7           ; IS THIS BW CARD
                                                                  外接单色转接口?
F385 7403               3854            JE      P1
F387 E9A902             3855            JMP     GRAPHICS_READ
F38A                    3856 P1:                                ; READ_AC_CON-
                                                                  TINUE  字符模式
F38A E81A00             3857            CALL    FIND_POSITION   作用见 3879～3894 行
                                                                  的 FIND_POSITION
F38D 8BF3               3858            MOU     SI, BX          ; ESTABLISH ADD-
                                                                  RESSING IN SI  光
                                                                  标位置送 SI
                        3859
                        3860 ; .....    WAIT FOR HORIZONTAL RETRACE
                        3861
F38F 8B166300 R         3862            MOV     DX, ADDR_6845   ; GET BASE ADDR-
                                                                  ESS   6845 基址送
                                                                  DX
F393 83C206             3863            ADD     DX, 6           ; POINT AT STATUS
```

```
                                                    PORT    DX 指向6845状态
                                                    寄存器
F396 06          3864           PUSH    ES      ;
F397 1F          3865           POP     DS      ; GET SEGMENT FOR QU-
                                                    ICK ACCESS   DS 指向显
                                                    示缓存区首
F398             3866 P2:                       ; WAIT FOR RETRACE L-
                                                    OW
F398 EC          3867           IN      AL, DX  ; GET STATUS  读6845 状
                                                    态寄存器
F399 A801        3868           TEST    AL, 1   ; IS HORZ RETRACE LOW
                                                    状态字节第0 位为1 表示允
                                                    许访问缓存，并且不干扰
                                                    CRT
F39B 75FB        3869           JNZ     P2      ; WAIT UNTIL IT IS
F39D FA          3870           CLI             ; NO MORE INTERRUPTS
                                                    第0 位=1，关中
F39E             3871 P3:                       ; WAIT FOR RETRACE H-
                                                    IGH
F39E EC          3872           IN      AL, DX  ; GET STATUS  状态字节送
                                                    AL
F39F A801        3873           TEST    AL, 1   ; IS IT HIGH
F3A1 74FB        3874           JZ      P3      ; WAIT UNTIL IT IS   等
                                                    待回扫期结束
F3A3 AD          3875           LODSW           ; GET THE CHAR/ATTR
                                                    取字符码/属性码，地址在
                                                    SI 中
F3A4 E920FE      3876           JMP     VIDEO_RETURN  准备返回
                 3877 READ_AC_CURRENT ENDP
                 3878
F3A7             3879 FIND_POSITION PROC  NEAR   下面程序段把光标在显示
                      缓存的位置送BX
F3A7 8ACF        3880           MOV     CL, BH  ; DISPLAY PAGE TO CX
                                                    显示页号送 CX
F3A9 32E0        3881           XOR     CH, CH
F3AB 8BF1        3882           MOV     SI, CX  ; MOVE TO SI FOR INDEX
F3AD D1E6        3883           SAL     SI, 1   ; ×2 FOR WORD OFFSET
                                                    显示页号 X2成字偏差
F3AF 8B845000 R  3884           MOV     AX, [SI + OFFSET CURSOR_POSN]
```

```
                                                ; GET ROW/COLUMN OF
                                                  THAT PAGE  取该页的
                                                  光标坐标
F3B3 33DB       3885        XOR   BX, BX        ; SET START ADDRESS
                                                  TO ZERO
F3B5 E306       3886        JCXZ  P5            ; NO_PAGE  页号=0
F3B7            3887 P4:                        ; PAGE_LOOP
F3B7 031E4C00 R 3888        ADD   BX, CRT_LEN   ; LENGTH OF BUFFER
                                                  否. BX指向下一个显示
                                                  页缓存区,
F3BB E2FA       3889        LOOP  P4              循环,直至指向要求的页
F3BD            3890 P5:                        ; NO_PAGE
F3BD E8CBFE     3891        CALL  POSITION      ; DETERMINE LOCATI-
                                                  ON IN REGEN  光标在
                                                  显示页中的偏移送AX
F3C0 03D8       3892        ADD   BX, AX        ; ADD TO START OF R-
                                                  EGEN  得光标在显示缓
                                                  存的位置
F3C2 C3         3893        RET                   返回
                3894 FIND_POSITION ENDP
                3895 ; ..........................................
                3896 ; WRITE_AC_CURRENT  WRITE_AC_CURRENT子程
                                                                  序
                3897 ;      THIS ROUTINE WRITES THE ATTRIBUTE AND
                            CHARACTER AT  本子程序将一字符的代码和属性符
                            写入显示缓存,使该
                3898 ;      THE CURRENT CURSOR POSITION  字符出现在光
                            标处
                3899 ; INPUT  入口参数:
                3900 ;      (AH)=CURRENT CRT MODE
                            AH)=当前CRT模式
                3901 ;      (BH)=DISPLAY PAGE  (BH)=显示页页号
                3902 ;      (CX)=COUNT OF CHARACTERS TO WRITE
                            (CX)=写入字符数
                3903 ;      (AL)=CHAR TO WRITE  (AL)=字符码
                3904 ;      (BL)=ATTRIBUTE OF CHAR TO WRITE
                            (BL)=字符属性
                3905 ;      (DS)=DATA SEGMENT  (DS)=数据段基址
                3906 ;      (ES)=REGEN SEGMENT  (ES)=缓存区基址
```

```
                        3907 ; OUTPUT  出口参数:
                        3908 ;         NONE  无
                        3909 ; ..........................................

F3C3                    3910 WRITE_AC_CURRENT  PROC NEAR
F3C3 80FC04             3911         CMP    AH, 4      ; IS THIS GRAPHICS  图形
                                                          模式?
F3C6 7208               3912         JC     P6
F3C8 80FC07             3913         CMP    AH, 7      ; IS THIS BW CARD  外接
                                                          单色转接口?
F3CB 7403               3914         JE     P6
F3CD E9B101             3915         JMP    GRAPHICS_WRITE
F3D0                    3916 P6:                       ; WRITE_AC_ CONTINUE
                                                          字符模式(彩色或单色转接
                                                          口)
F3D0 8AE3               3917         MOV    AH, BL     ; GET ATTRIBUTE TO AH
                                                          属性符送 AH
F3D2 50                 3918         PUSH   AX         ; SAVE ON STACK
F3D3 51                 3919         PUSH   CX         ; SAVE WRITE COUNT
                                                          属性符、写入个数压栈保护
F3D4 E8D0FF             3920         CALL   FIND_POSITION
F3D7 8BFB               3921         MOV    DI, BX     ; ADDRESS TO DI REGI-
                                                          STER  光标在显示缓存的
                                                          位置送 DI
F3D9 59                 3922         POP    CX         ; WRITE COUNT
F3DA 5B                 3923         POP    BX         ; CHARACTER IN BX REG
                                                          (CX) = 写入个数,  (BH) =
                                                          属性符, (BL) = 字符 ASC II
                                                          码
F3DB                    3924 P7:                       ; WRITE_LOOP
                        3925
                        3926 ; ......   WAIT FOR HORIZONTAL RETRACE
                        3927
F3DB 8B166300 R         3928         MOV    DX, ADDR_6845  ; GET BASE AD-
                                                          DRESS
F3DF 83C206             3929         ADD    DX, 6      ; POINT AT STATUS PORT
                                                          DX 指向 6845 状态口
F3E2                    3930 P8:
F3E2 EC                 3931         IN     AL, DX     ; GET STATUS
```

```
F3E3 A801      3932              TEST    AL, 1       ; IS IT LOW
F3E5 75FB      3933              JNZ     P8          ; WAIT UNTIL IT IS 非回扫期
                                                     则等待
F3E7 FA        3934              CLI                 ; NO MORE INTERRUPTS 关
                                                     中
F3E8           3935 P9:
F3E8 EC        3936              IN      AL, DX      ; GET STATUS 取6845状态
F3E9 A801      3937              TEST    AL, 1       ; IS IT HIGH
F3EB 74FB      3938              JZ      P9          ; WAIT UNTIL IT IS 回扫期结
                                                     束?
F3ED 8BC3      3939              MOV     AX, BX      ; RECOVER THE CHAR/ATTR
                                                     字符码及属性码送AX
F3EF AB        3940              STOSW               ; PUT THE CHAR/ATTR 写入
                                                     显示缓存,地址在DI中
F3F0 FB        3941              STI                 ; INTERRUPTS BACK ON 开中
F3F1 E2E8      3942              LOOP    P7          ; AS MANY TIMES AS REQU-
                                                     ESTED 写入数个序符
F3F3 E9D1FD    3943              JMP     VIDEO_RETURN 写入完毕,准备返回
               3944 WRITE_AC_CURRENT  ENDP
               3945 ; ..........................................
               3946 ; WRITE_C_CURRENT  WRITE_C_CURRENT 子程序
               3947 ;          THIS ROUTINE WRITES THE CHARACTER AT
                                本子程字将一字符写在当前光标处,字符属性不变
               3948 ;          THE CURRENT CURSOR POSITION, ATTRIBUTE
                                UNCHANGED
               3949 ; INPUT     入口参数:
               3950 ;          (AH) = CURRENT CRT MODE
                                (AH) = 当前 CRT 模式
               3951 ;          (BH) = DISPLAY PAGE  (BH) = 显示页页号
               3952 ;          (CX) = COUNT OF CHARACTERS TO WRITE
                                (CX) = 写入字符个数
               3953 ;          (AL) = CHAR TO WRITE  (AL) = 字符代码
               3954 ;          (DS) = DATA SEGMENT  (DS) = 数据段基址
               3955 ;          (ES) = REGEN SEGMENT  (ES) = 显示缓存区基址
               3956 ; OUTPUT
               3957 ;          NONE
               3958 ; ..........................................
F3F6           3959 WRITE_C_CURRENT PROC NEAR
F3F6 80FC04    3960              CMP     AH, 4       ; IS THIS GRAPHICS 图形模
```

```
                                                        式?
F3F9 7208       3961        JC      P10
F3FB 80FC07     3962        CMP     AH, 7           ; IS THIS BW CARD 外
                                                        接单色转接口?
F3FE 7403       3963        JE      P10
F400 E97E01     3964        JMP     GRAPHICS_WRITE
F403            3965 P10:                           字符式(彩色或单色转接口)
F403 50         3966        PUSH    AX              ; SAVE ON STACK
F404 51         3967        PUSH    CX              ; SAVE WRITE COUNT
                                                        (AX), (CX)压栈
F405 E89FFF     3968        CALL    FIND_POSITION
F408 8BFB       3969        MOV     DI, BX          ; ADDRESS TO DI 光标
                                                        在显示缓存位置送 DI
F40A 59         3970        POP     CX              ; WRITE COUNT 取回写
                                                        入个数
F40B 5B         3971        POP     BX              ; BL HAS CHAR TO W-
                                                        RITE 取回待写入字符
                                                        (在 BL)
F40C            3972 P11:                           ; WRITE_LOOP
                3973
                3974 ; ..... WAIT FOR HORIZONTAL RETRACE
                3975
F40C 8B166300 R 3976        MOV     DX, ADDR_6845   ; GET BASE ADDRESS
F410 83C206     3977        ADD     DX, 6           ; POINT AT STATUS P-
                                                        ORT DX 指向 6845状态
                                                        口
F413            3978 P12:
F413 EC         3979        IN      AL, DX          ; GET STATUS
F414 A801       3980        TEST    AL, 1           ; IS IT LOW
F416 75FB       3981        JNZ     P12             ; WAIT UNTIL IT IS 非
                                                        回扫期则等待
F418 FA         3982        CLI                     ; NO MORE INTERRUPTS
                                                        关中
F419            3983 P13:
F419 EC         3984        IN      AL, DX          ; GET STATUS
F41A A801       3985        TEST    AL, 1           ; IS IT HIGH
F41C 74FB       3986        JZ      P13             ; WAIT UNTIL IT IS 回
                                                        扫期结束?
```

```
F41E 8AC3    3987            MOV     AL, BL  ; RECOVER CHAR  结束,送字符
                                                码到 AL
F420 AA      3988            STOSB           ; PUT THE CHAR/ATTR  (AL)
                                                写入显示缓存,地址在 DI
F421 47      3989            INC     DI      ; BUMP POINTER PAST ATTRI-
                                                BUTE 地址加工 (STQSB 中已加
                                                1),跳过属性符
F422 E2E8    3990            LOOP    P11     ; AS MANY TIMES AS REQU-
                                                ESTED  写入数个字符
F424 E9A0FD  3991            JMP     VIDEO_RETURN  转 VIDEO_RETURN,
                                                返回
             3992 WRITE_C_CURRENT ENDP
             3993 ; ..........
             3994 ; READ DOT…WRITE DOT READ DOT 及WRITE DOT 子程序
             3995 ; THESE ROUTINES WILL WRITE A DOT, OR READ THE
                    本程序在图形模式 CRT 上读取或写入一点,该点位置事先指定
             3996 ;  DOT AT THE INDICATED LOCATION
             3997 ; ENTRY…入口参数:
             3998 ;  DX = ROW (0-199)  (THE ACTUAL VALUE DEPENDS ON
                    THE MODE)  (DX) = 行号(0-199)实际值由模式决定
             3999 ;  CX = COLUMN (0-639)  (THE VALUES ARE NOT RANGE
                    CHECKED)  (CX) = 列号(0～639),对列号不检查其合法性
             4000 ;  AL = DOT VALUE TO WRITE(1, 2 OR 4 BITS DEPENDING
                    ON MODE  (AL) = 写入的点值,640×200点值占 1 位,320×200
                    点值占 2 位,
             4001 ;        REQ'D FOR WRITE DOT ONLY, RIGHT JUSTIFIED)
                           160×100 点值占 4 位。点值放在 AL 低位部分
             4002 ;        BIT 7 OF AL = 1 INDICATES XOR THE VALUE IN-
                           TC THE LOCATION  (AL)的第 7 位为 1 则点值异或入显
                           示缓存
             4003 ;  DS = DATA SEGMENT  (DS)数据段基址
             4004 ;  ES = REGEN SEGMENT  (ES) = 显示缓存基址
             4005 ; ..........
             4006 ; EXIT  出口参数:
             4007 ;           AL = DOT VALUE READ, RIGHT JUSTIFIED, RE-
                              AD ONLY  (AL) = 向右调整过了的点值(读取点值)
             4008 ; ..........
             4009             ASSUME CS:CODE, DS:DATA, ES:DATA  READ
                              _DOT 子程序
```

```
F427                4010 READ_DOT  PROC  NEAR
F427 E83100         4011           CALL  R3          ; DETERMINE BYTE
                                                     POSITION OF DOT
                                                     决定点在字节中的位置
F42A 268A04         4012           MOV   AL, ES:(SI) ; GET THE BYTE  取
                                                     点所在字节
F42D 22C4           4013           AND   AL, AH      ; MASK OFF THE OT-
                                                     HER BITS IN THE
                                                     BYTE  剔除与该点无
                                                     关的位
F42F D2E0           4014           SHL   AL, CL      ; LEFT JUSTIFY THE
                                                     VALUE  左调点值到
                                                     字节高位
F431 8ACE           4015           MOV   CL, DH      ; GET NUMBER OF
                                                     BITS IN RESULT  点
                                                     值占的存贮位数送 CL
F433 02C0           4016           ROL   AL, CL      ; RIGHT JUSTIFY THE
                                                     RESULT  左环移 AL,
                                                     将点值移入AL低位,方
                                                     便读取
F435 E98FFD         4017           JMP   VIDEO_RETURN ; RETURN FROM VID-
                                                     EO IO  转 VIDEO_
                                                     RETORN返回
                    4018 READ_DOT  ENDP
                    4019
F438                4020 WRITE_DOT PROC  NEAR  WRITE_DOT 子程序
F438 50             4021           PUSH  AX          ; SAVE DOT VALUE
F439 50             4022           PUSH  AX          ; TWICE  保存点值
F43A E81E00         4023           CALL  R3          ; DETERMINE BYTE
                                                     POSITION OF THE D
                                                     OT  决定点在字节中
                                                     的位置、点值在AL高位
                                                     部分
F43D D2E8           4024           SHR   AL, CL      ; SHIFT TO SET UP
                                                     THE BITS FOR OUT-
                                                     PUT  右移点值至字节
                                                     实际存放位置
F43F 22C4           4025           AND   AL, AH      ; STRIP OFF THE OT-
                                                     HER BITS  剔除与该
```

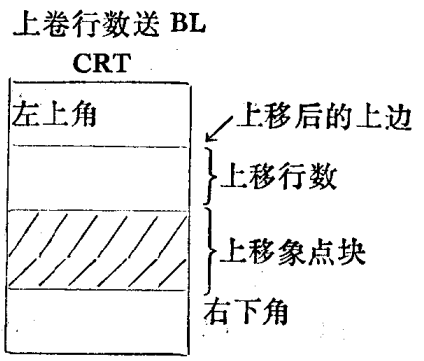
```
                                                    点无关的位
F441 268A0C   4026            MOV   CL, ES:[SI]    ; GET THE CURRENT
                                                      BYTE 取点所在字节
F444 5B       4027            POP   BX             ; RECOVER XOR FLAG
                                                      取点值字节
F445 F6C380   4028            TEST  BL, 80H        ; IS IT ON 第7位为1?
F448 750D     4029            JNZ   R2             ; YES, XOR THE DOT
F44A F6D4     4030            NOT   AH             ; SET THE MASK TO
                                                      REMOVE THE INDI-
                                                      CATED BITS 第7位
                                                      为0,非异或入,屏蔽字
                                                      节取反
F44C 22CC     4031            AND   CL, AH         字节中待写入点位置清0
F44E 0AC1     4032            OR    AL, CL         ; OR IN THE NEW V-
                                                      ALUE OF THOSE BI-
                                                      TS 点值送入记录字节
F450          4033 R1:                             ; FINISH_DOT
F450 268804   4034            MOV   ES:[SI], AL    ; RESTORE THE BYT-
                                                      E IN MEMORY 记录
                                                      字节送显示缓存
F453 58       4035            POP   AX
F454 E970FD   4036            JMP   VIDEO_RETURN   ; RETURN FROM VI-
                                                      DEO IO 转VIDEO_
                                                      RETORN返回
F457          4037 R2:                             ; XOR_DOT
F457 32C1     4038            XOR   AL, CL         ; EXCLUSIVE OR THE
                                                      DOTS 点值异或入记
                                                      录字节
F459 EBF5     4039            JMP   R1             ; FINISH UP THE W-
                                                      RITING
              4040 WRITE_DOT  ENDP
              4041 ; ..........................................
              4042 ; THIS SUBROUTINE DETERMINES THE REGEN BYTE LOC-
                     ATION OF THE 本程序根据指定的行列值找出该点在显示缓存
                     中记录字节的位置
              4043 ; INDICATED ROW COLUMN VALUE IN GRAPHICS MODE
              4044 ; ENTRY…入口参数:
              4045 ;  DX=ROW VALUE(0-199)  (DX)=行数(0-199)
```

```
                4046 ;  CX = COLUMN VALUE(0-639)  (CX) = 列数(0～639)
                4047 ;  EXIT--出口参数:
                4048 ;  SI = OFFSET INTO REGEN BUFFER FOR BYTE OF INTER-
                        EST  (SI) = 记录字节在显示缓存的的偏移
                4049 ;  AH = MASK TO STRIP OFF THE BITS OF INTEREST
                        (AH) = 字节中感兴趣位的屏蔽位
                4050 ;  CL = BITS TO SHIFT TO RIGHT JUSTIFY THE MASK IN
                        AH  (CL) = 右调整 AH 中屏蔽位的位数
                4051 ;  DH = # BITS IN RESULT  (DH) = 结果位数
                4052 ;  ..........................................
F45B            4053 R3     PROC  NEAR
F45B 53         4054        PUSH  BX        ; SAVE BX DURING OPERATI-
                                                ON
F45C 50         4055        PUSH  AX        ; WILL SAVE AL DURING OPE-
                                                RATION  保存(BX), (AX)
                4056
                4057 ; ..... DETERMINE IST BYTE IN IDICATED ROW BY MU-
                        LTIPLYING ROW VALUE BY 40, 6845 规定行数为奇数
                        的点的信息存放在奇显示缓存区
                4058 ; ..... ( LOW BIT OF ROW DETERMINES EVEN/ODD,  80
                        BYTES/ROW  行数为偶数的点的信息存放在偶显示缓存区
                4059        不管何种图形模式, 每行总占 80 个字节的显示缓存, 又由
                        于奇/
F45D B028       4060        MOV AL, 40  偶缓存的缘故,在决定(DX) 决定行的第一字
                                        节地址时,(DX)只
F45F 52         4061        PUSH  DX        ; SAVE ROW VALUE  乘 40
F460 80E2FE     4062        AND   DL,0FEH   ; STRIP OFF ODD/EVEN BIT 先
                                                不考虑奇行还是偶行
F463 F6E2       4063        MUL   DL        ; AX HAS ADDRESS OF IST B-
                                                YTE OF INDICATED ROW
                                                (DL) × 40⇒AX, (AX) 为指定行
                                                第一字符的相对偏移地址
F465 5A         4064        POP   DX        ; RECOVER IT
F466 F6C201     4065        TEST  DL, 1     ; TEST FOR EVEN/ODD  奇行还
                                                是偶行?
F469 7403       4066        JZ    R4        ; JUMP IF EVEN ROW
F46B 050020     4067        ADD   AX, 2000H ; OFFSET TO LOCATION OF O-
                                                DD ROWS  偶行, 加上偶显示缓
```

```
                                                    存的相对始址 2000H
F46E                    4068 R4:                    ; EVEN_ROW
F46E 8BF0               4069        MOV   SI, AX    ; MOVE POINTER TO SI  字节
                                                    地址送 SI
F470 58                 4070        POP   AX        ; RECOVER AL VALUE  点值
                                                    送 AL
F471 8BD1               4071        MOV   DX, CX    ; COLUMN VALUE TO DX  列
                                                    数送 DX
                        4072
                        4073 ; ..... DETERMINE GRAPHICS MODE CURRENTLY IN
                                     EFFECT  下面决定当前的图形模式，注解中的M指中
                                     分辨率，H 指高分辨率
                        4074
                        4075 ; SET UP THE REGISTERS ACCORDING TO THE MODE
                        4076 ; CH = MASK FOR LOW OF COLUMN ADDRESS (7/3 FOR
                                     HIGH/MED RES)  (CH) = 地址位的屏蔽值，对 M，因地址
                                     为 2 位，故屏蔽值为 11(3)
                        4077 ; CL = # OF ADDRESS BITS IN COLUMN VALUE (3/2
                                     FOR H/M)  (CL) = 决定点在字节中的位置需地址位数，对
                                     M，每字节有 4 点，地位位为 2
                        4078 ; BL = MASK TO SELECT BITS FROM POINTED BYTE (80
                                     H/COH FOR H/M)  (BL) = 指定字节中对应某个点的屏蔽
                                     值，如对中分辨率，取 1100，0000
                        4079 ; BH = NUMBER OF VALID BITS IN POINTED BYTE (1/2
                                     FOR H/M)  (BH) = 每个点值占的位数，高分辨率取 1，中分
                                     辨率取 2
                        4080
F473 BBC002             4081        MOV   BX, 2C0H
F476 B90203             4082        MOV   CX, 302H  ; SET PARMS FOR MED RES
                                                    先设M参数
F479 803E490006 R       4083        CMP   CRT_MODE, 6  高分辨率?
F47E 7206               4084        JC    R5        ; HANDLE IF MED ARES
F480 BB8001             4085        MOV   BX, 180H
F483 B90307             4086        MOV   CX, 703H  ; SET PARMS FOR HIGH RES
                                                    设高分辨率参数
                        4087
                        4088 ; ..... DETERMINE BIT OFFSET IN BYTE FROM COLU-
                                     MN MASK
F486                    4089 R5:
```

```
F486 22EA  4090          AND    CH, DL    ; ADDRESS OF PEL WITHIN BYTE
                                          TO CH  取列数(列坐标)的低位(3/2
                                          —H/M)，他们决定点值在一个字节
                 4091                     中的位置
                 4092 ; ……  DETERMINE BYTE OFFSET FOR THIS LOCATION IN
                            COLUMN
                 4093
F488 D3EA  4094          SHR    DX, CL    ; SHIFT BY CORRECT AMOUNT
                                          右移列坐标，移出决定点在字节中位
                                          置的位
F48A 03F2  4095          ADD    SI, DX    ; INCREMENT THE POINTER  行第
                                          一字节地址加点所在字节与第一字节
                                          的偏差，得记录字节在显示
F48C 8AF7  4096          MOV    DH, BH    ; GET THE # OF BITS IN RESULT-
                                          TO DH  缓存的位置。每个点值的位
                                          数送 DH
                 4097
                 4098 ; ……  MULTIPLY BH, (VALID BITS IN BYTE) BY CH (BIT O
                            FFSET)  每个点占的位数× 点在字节中的偏差 = 点值实际
                            存放的位置
                 4099
F48E 2AC9  4100          SUB    CL, CL    ; ZERO INTO STORAGE LDCATION
                                          清 CI
F490       4101 R6:
F490 D0C8  4102          ROR    AL, 1     ; LEFT JUSTIFY THE VALUE IN
                                          AL (FOR WRITE)  将 AL 最低位
                                          的点值移入 AL 最高位
F492 02CD  4103          ADD    CL, CH    ; ADD IN THE BIT OFFSET VALUE
                                          加进点在字节中的偏差
F494 FECF  4104          DEC    BH        ; LOOP CONTROL  点值的位数值作
                                          计数值
F496 75F8  4105          JNZ    R6        ; ON EXIT, CL HAS SHIFT COUNT
                                          TO RESTORE BITS  (CL) = 点值实
                                          际在字节中的存放位置
F498 8AE3  4106          MOV    AH, BL    ; GET MASK TO AH 点屏蔽值送 AH
F49A D2EC  4107          SHR    AH, CL    ; MOVE THE MASK TO CORRECT
                                          LOCATION  右移 AH，使之与欲写
                                          入/读出位在字节中的位置对齐
F49C 5B    4108          POP    BX        ; RECOVER REG  恢复行数
```

```
F49D C3      4109          RET                ; RETURN WITH EVERYTHING
                                                SET UP  R3 子程序结束
             4110 R3       ENDP
             4111 ; ..........................................
             4112 ; SCROLL UP  GRAPHICS-UP 图形 CRT 上卷子程序
             4113 ; THIS ROUTINE SCROLLS UP THE INFORMATION ON THE
                    CRT  本程序上卷一象点块.
             4114 ; ENTRY…人口参数:
             4115 ; CH, CL = UPPER LEFT CORNER OF REGION TO SCROLL
                    (CH, CL) = 上卷象点块左上角坐标(8 点算一行及一列)
             4116 ; DH, DL = LOWER RIGHT CORNER OF REGION TO SCROLL
                    (DH, DL) = 上卷象点块右下角坐标(8 点算一行及一列)
             4117 ; BOTH OF THE ABOVE ARE IN CHARACTER POSITIONS
             4118 ; BH = FILL VALUE FOR BLANKED LINES  (BH) = 空出行填充
                    值
             4119 ; AL = # LINES TO SCROLL  (AL = 0 MEANS BLANK THE E-
                    NTIRE FIELD)  (AL) = 上卷行数, AL = 0 表示整个象点块置成
                    空白
             4120 ; DS = DATA SEGMENT  (DS) = 数据段基址
             4121 ; ES = REGEN SEGMENT  (ES) = 显示缓存段基址
             4122 ; EXIT…出口参数:
             4123 ; NOTHING, THE SCREEN IS SCROLLED  无,图象上卷
             4124 ; ..........................................
```




```
F49E         4125 GRAPHICS_UP PROC NEAR
F49E 8AD8    4126          MOV   BL, AL    ; SAVE LINE COUNT IN BL
F4A0 8BC1    4127          MOV   AX, CX    ; GET UPPER LEFT POSITION
                                               INTO AX REG  左上角坐标送 AX
             4128
             4129 ; ……     USE CHARACTER SUBROUTINE FOR POSITIONING
             4130 ; ……     ADDRESS RETURNED IS MULTIPLIED BY 2 FROM
                           CORRECT VALUE
```

```
                        4131
F4A2 E86A02             4132        CALL    GRAPH_POSN      将左上角象点在奇/偶显示缓
                                                            存的位置(640×200)送 AX
F4A5 8BF8               4133        MOV     DI, AX          ; SAVE RESULT AS DEST-
                                                            INATION ADDRESS
                                                            (AX)送 DI
                        4134
                        4135 ;……DETERMINE SIZE OF WINDOW
                        4136
F4A7 2BD1               4137        SUB     DX, CX          象点块的行、列数=右下角
                                                            行、列数-左上角行、列数
F4A9 81C20101           4138        ADD     CX, 101H        ; ADJUST VALUES 考虑进
                                                            0行、0列
F4AD D0E6               4139        SAL     DH, 1           ; MULTIPLY # ROWS BY 4
                                                            SINCE 8 VERT DOTS/
                                                            CHAR 行数×4,得象点行
                                                            数(对应偶/奇缓存)
F4AF D0E6               4140        SAL     DH, 1           ; AND EVEN/ODD ROWS
                        4141
                        4142 ;……DETERMINE CRT MODE
                        4143
F4B1 803E490006 R       4144        CMP     CPT_MODE, 6     ; TEST FOR MEDIUM RES
                                                            中分辨率 320×200 模式?
F4B6 7304               4145        JNC     R7              ; FIND_SOURCE 否,转R7
                        4146
                        4147 ;……MEDIUM RES UP 中分辨率模式特殊处理
F4B8 D0E2               4148        SAL     DL, 1           ; # COLUMNS × 2, SINCE
                                                            2 BYTES/CHAR 列数×2
                                                            得每行字节数,以后的移送
                                                            以字节为单位
F4BA D1E7               4149        SAL     DI, 1           ; OFFSET * 2 SINCE 2 BY-
                                                            TES/CHAR (DI) ×2,中
                                                            分辨率模式下一点占二位
                        4150
                        4151 ;……DETERMINE THE SOURCE ADDRESS IN THE BUF-
                                    FER
F4BC                    4152    R7:                         ; FIND_SOURCE
F4BC 06                 4153        PUSH    ES              ; GET SEGMENTS BOTH
```

```
                                                    POINTING TO REGEN  DS,
                                                    ES 同时指向显示缓存区首
F4BD 1F          4154       POP     DS
F4BE 2AED        4155       SUB     CH, CH          ; ZERO TO HIGH OF COUNT
                                                    REG
F4C0 D0E3        4156       SAL     BL, 1           ; MULTIPLY NUMBER OF LIN-
                                                    ES BY 4   上移行×4=上移象
                                                    点行(对应奇/偶缓存)
F4C2 D0E3        4157       SAL     BL, 1
F4C4 742D        4158       JZ      R11             ; IF ZERO, THEN BLANK EN-
                                                    TIRE FIELD  上移行数=0,象
                                                    点块全部送填充值
F4C6 8AC3        4159       MOV     AL, BL          ; GET NUMBER OF LINES IN
                                                    AL
F4C8 B450        4160       MOV     AH, 80          ; 80 BYTES/ROW
F4CA F6E4        4161       MUL     AH              ; DETERMINE OFFSET TO S-
                                                    OURCE   象点行数×80=这些
                                                    象点行需占据的字节数
F4CC 8BF7        4162       MOV     SI, DI          ; SET UP SOURCE  左上角象点
                                                    地址送 SI
F4CE 03F0        4163       ADD     SI, AX          ; ADD IN OFFSET TO IT  (SI)
                                                    +AX,得欲上移象点行第一字节
                                                    的地址
F4D0 8AE6        4164       MOV     AH, DH          ; NUMBER OF ROWS IN FIELD
F4D2 2AE3        4165       SUB     AH, BL          ; DETERMINE NUMBER TO M-
                                                    OVE  象点块总象点行-上移象
                                                    点行=实际往上搬的象点行数
                 4166
                 4167 ;……LOOP THROUGH, MOVING ONE ROW AT A TIME,
                         BOTH EVEN AND ODD FIELDS
F4D4             4168 R8:                           ; ROW__LOOP
F4D4 E88000      4169       CALL    R17             ; MOVE ONE ROW  往上搬移一
                                                    行
F4D7 81EEB01F    4170       SUB     SI, 2000H-80    ; MOVE TO NEXT ROW  SI,
                                                    DI 扣除 R17 子程序中加的 2000
                                                    H,并加上 80 字节,指向下一行
F4DB 81EFB01F    4171       SUB     DI, 2000H-80
F4DF FECC        4172       DEC     AH              ; NUMBER OF ROWS TO MOVE
                                                    象点块移完?
```

```
F4E1 75F1       4173            JNZ     R8              ; CONTINUE TILL
                                                        ALL MOVED 否,
                                                        转 R8
                4174
                4175 ; ……       EILL IN THE VACATED LINE(S)
F4E3            4176 R9:                                ; CLEAR_ENTRY
F4E3 8AC7       4177            MOV     AL, BH          ; ATTRIBUTE TO F-
                                                        ILL WITH 填入值
                                                        送 AL
F4E5            4178 R10:
F4E5 E88800     4179            CALL    R18             ; CLEAR THAT ROW
                                                        为一象点行送填充值
F4E8 81EFB01F   4180            SUB     DI, 2000H-80    ; POINT TO NEXT
                                                        LINE DI 指向下一
                                                        行
F4EC FECB       4181            DEC     BL              ; NUMBER OF LINES
                                                        TO FILL
F4EE 75F5       4182            JNZ     R10             ; CLEAR_LOOP
                                                        填空完毕?
F4F0 E9D4FC     4183            JMP     VIDEO_RETURN    ; EVERYTHING DO-
                                                        NE 完毕,转VIDEO
                                                        _RETORN 返回
                4184
F4F3            4185 R11:                               ; BLANK_FIELD
                                                        象点块全部送填充值
F4F3 8ADE       4186            MOV     BL, DH          ; SET BLANK COU-
                                                        NT TO EVERYTH-
                                                        ING IN FIELD 块
                                                        之总象点行数送 BL
F4F5 EBEC       4187            JMP     R9              ; CLEAR THE FIELD
                                                        转 R9
                4188 GRAPHICS_UP  ENDP
                4189 ; ……
                4190 ; SCROLL DOWN  GRAPHICS_DOWN 图形下卷子程序
                4191 ; THIS ROUTINE SCROLLS DOWN THE INFORMATION ON
                       THE CRT  本程序下卷一象点块
                4192 ; ENTRY…入口参数:
                4193 ; CH, CL=UPPER LEFT CORNER OF REGION TO SCROLL
                       (CH, CL) = 下卷象点块左上角坐标(8 点算一行及一列)
```

```
                4194 ; DH, DL = LOWER RIGHT CORNER OF REGION TO SCROLL
                       (DH, DL) = 下卷象点块右下角坐标(8 点算一行及一列)
                4195 ; BOTH OF THE ABOVE ARE IN CHARACTER POSITIONS
                4196 ; BH = FILL VALUE FOR BLANKED LINES    (BH) = 空出行
                       填充值
                4197 ; AL = # LINES TO SCROLL (AL = 0 MEANS BLANK THE
                       ENTIRE FIELD  (AL) = 下卷行数。为 0 表示整个象点块送统
                       一的填充值
                4198 ; DS = DATA SEGMENT   (DS) = 数据段基址
                4199 ; ES = REGEN SEGMENT   (ES) = 显示缓存基址
                4200 ; EXIT…出口参数:
                4201 ; NOTHING, THE SCREEN IS SCROLLED   无,图象下卷
                4202 ; ……
                4203
F4F7            4204 GRAPHICS_DOWN  PROC  NEAR
F4F7 FD         4205         STD          ; SET DIRECTION   逆向标记
F4F8 8AD8       4206         MOV   BL, AL ; SAVE LINE COUNT IN BL
                                            下卷行数送 BL
F4FA 8BC2       4207         MOV   AX, DX ; GET LOWER RIGHT POS-
                                            ITION INTO AX REG   右
                                            右下角坐标送 AX
                4208
                4209 ; ……   USE CHARACTER SUBROUTINE FOR POSITION-
                             ING
                4210 ; ……   ADDRESS RETURNED IS MULTIPLIED BY 2 F-
                             ROM CORRECT VALUE
                4211
F4FC E81002     4212         CALL  GRAPH_POSN  将左上角象点在奇/偶显示
                                            缓存的位置(640×200)送 AX
F4FF 8BF8       4213         MOV   DI, AX ; SAVE RESULT AS DEST-
                                            INATION ADDRESS (AX)
                                            送 DI
                4214
                4215 ; ……   DETERMINE SIZE OF WINDOW
                4216
F501 2BD1       4217         SUB   DX, CX ; (DX) = 象点执行、列数
F503 81C20101   4218         ADD   DX, 101H ; ADJUST VALUES   加进第
                                            0 行,第 0 列
```

```
F507 D0E6          4219      SAL     DH, 1          ; MULTIPLY # ROWS BY 4
                                                    SINCE 8 VERT DOTS/CH-
                                                    AR 行数×4，得象点行数
                                                    (对应偶/奇显示缓存)
F509 D0E6          4220      SAL     DH, 1          ; AND EVEN/ODD ROWS
                   4221
                   4222 ;……DETERMINE CRT MODE
                   4223
F50B 803E490006 R  4224      CMP     CRT_MODE, 6    ; TEST FOR MEDIUM RES
                                                    320×200 模式?
F510 7305          4225      JNC     R12            ; FIND_SOURCE_DOWN
                                                    否转 R12
                   4226
                   4227 ;……MEDIUM RES DOWN  320×200 特殊处理
F512 D0E2          4228      SAL     DL, 1          ; # COLUMNS*2, SINCE 2
                                                    BYTES/CHAR (OFFSET
                                                    OK) 列数×2,得每行字节
                                                    数，以后的移送以字节为单
                                                    位
F514 D1E7          4229      SAL     DI, 1          ; OFFSET*2 SINCE 2 BY-
                                                    TES/CHAR (DI)×2，中
                                                    分辨率下一点占二位
F516 47            4230      INC     DI             ; POINT TO LAST BYTE
                                                    DI 指向后一字节
                   4231
                   4232 ;……DETERMINE THE SOURCE ADDRESS IN THE BUF-
                             FER
F517               4233      R12;                   ; FIND_SOURCE_DOWN
F517 06            4234      PUSH    ES             ; BOTH SEGMENTS TO R-
                                                    EGEN
F518 1F            4235      POP     DS             ES, DS 同时指向显示缓存首
F519 2AED          4236      SUB     CH, CH         ; ZERO TO HIGH OF COU-
                                                    NT REG 清 CH
F51B 81C7F000      4237      ADD     DI, 240        ; POINT TO LAST ROW OF
                                                    PIXELS 下移三个象点行，
                                                    考虑进字符间的空隙
F51F D0E3          4238      SAL     BL, 1          ; MULTIPLY NUMBER OF
                                                    LINES BY 4 下移行×4=下
                                                    移象点行(对应奇/偶缓存)
```

```
F521 D0E3      4239      SAL     BL, 1
F523 742E      4240      JZ      R16          ; IF ZERO, THEN BLANK E-
                                                NTIRE FIELD
F525 8AC3      4241      MOV     AL, BL       ; GET NUMBER OF LINES
                                                IN AL
F527 B450      4242      MOV     AH, 80       ; 80 BYTES/ROW  象点行数
                                                ×80=这些象点行需占据的字
                                                节数
F529 F6E4      4243      MUL     AH           ; DETERMINE OFFSET TO
                                                SOURCE
F52B 8BF7      4244      MOV     SI, DI       ; SET UP SOURCE
F52D 2BF0      4245      SUB     SI, AX       ; SUBTRACT THE OFFSET
                                                下搬象点行第未字节（对于象
                                                点块)的地址送SI
F52F 8AE6      4246      MOV     AH, DH       ; NUMBER OF ROWS IN FI-
                                                ELD
F531 2AE3      4247      SUB     AH, BL       ; DETERMINE NUMBER TO
                                                MOVE  象点块总象点行-下
                                                移象点行=实际往下搬的象点
                                                行数
               4248
               4249 ;……LOOP THROUGH, MOVING ONE ROW AT A TIME,
                       BOTH EVEN AND ODD FIELDS
F533           4250 R13:                    ; ROW_LOOP_DOWN
F533 E82100    4251      CALL    R17          ; MOVE ONE ROW  下移一行
F536 81EE5020  4252      SUB     SI, 2000H+80 ; MOVE TO NEXT ROW  扣
                                                除R17子程序中加的2000H,
                                                并减去80字节,指向上一行
F53A 81EF5020  4253      SUB     DI, 2000H+80
F53E FECC      4254      DEC     AH           ; NUMBER OF ROWS TO M-
                                                OVE  象点块移完?
F540 75F1      4255      JNZ     R13          ; CONTINUE TILL ALL MO-
                                                VED  否,转R13
               4256
               4257 ;……FILL IN THE VACATED LINE(S)
F542           4258 R14:                    ; CLEAR_ENTRY_DOWN
F542 8AC7      4259      MOV     AL, BH       ; ATTRIBUTE TO FILL WITH
                                                填充值送AL
F544           4260 R15:                    ; CLEAR_LOOP_DOWN
```

```
F544 E82900      4261          CALL    R18              ; CLEAR A ROW 为一象点行送填充符
F547 81EF5020    4262          SUB     DI, 2000H+80     ; POINT TO NEXT LINE DI指向上一行
F54B FECB        4263          DEC     BL               ; NUMBER OF LINES TO FILL
F54D 75F5        4264          JNZ     R15              ; CLEAR_LOOP_DOWN 填充工作完毕?
F54F FC          4265          CLD                      ; RESET THE DIRECTION FLAG 前向标记。返回
F550 E974FC      4266          JMP     VIDEO_RETURN     ; EVERYTHING DONE
                 4267
F553             4268 R16;                              ; BLANK_FIELD_DOWN 象点块全部送填充值
F553 8ADE        4269          MOV     BL, DH           ; SET BLANK COUNT TO EVERYTHING IN FIELD 块的总象点行数送 BL
F555 EBEB        4270          JMP     R14              ; CLEAR THE FIELD 转 R14
                 4271 GRAPHICS_DOWN ENDP
                 4272
                 4273 ; ..... ROUTINE TO MOVE ONE ROW OF INFORMATION
                 4274
F557             4275 R17      PROC    NEAR
F557 8ACA        4276          MOV     CL, DL           ; NUMBER OF BYTES IN THE ROW 一行需字节数送 CL, 作计数器
F559 56          4277          PUSH    SI
F55A 57          4278          PUSH    DI               ; SAVE POINTERS 保存搬出、移入指针
F55B F3          4279          REP     MOVSB            ; MOVE THE EVEN FIELD 搬送偶象点行
F55C A4
```

```
F55D 5F          4280        POP     DI
F55E 5E          4281        POP     SI
F55F 81C60020    4282        ADD     SI, 2000H
F563 81C70020    4283        ADD     DI, 2000H     ; POINT TO THE ODD FIELD
                                                     SI, DI 指向奇缓存区
F567 56          4284        PUSH    SI
F568 57          4285        PUSH    DI            ; SAVE THE POINTERS
F569 8ACA        4286        MOV     CL, DL        ; COUNT BACK  置计数器
F56B F3          4287        REP     MOVSB         ; MOVE THE ODD FIELD  搬送
                                                     奇行

F56C A4
F56D 5F          4288        POP     DI
F56E 5E          4289        POP     SI            ; POINTERS BACK  恢复指针,他
                                                     们指向奇缓存区
F56F C3          4290        RET                   ; RETURN TO CALLER  返回
                 4291 R17    ENDP
                 4292
                 4293 ;……CLEAR A SINGLE ROW  为象点行填充字符
                 4294
F570             4295 R18    PROC    NEAR
F570 8ACA        4296        MOV     CL, DL        ; NUMBER OF BYTES IN FIELD
                                                     每行占字节数
F572 57          4297        PUSH    DI            ; SAVE POINTER
F573 F3          4298        REP     STOSB         ; STORE THE NEW VALUE  送
                                                     填充值

F574 AA
F575 5F          4299        POP     DI            ; POINTER BACK
F576 81C70020    4300        ADD     DI, 2000H             ; POINT TO ODD FIELD
                                                             DI 指向奇区
F57A 57          4301        PUSH    DI
F57B 8ACA        4302        MOV     CL, DL
F57D F3          4303        REP     STOSB         ; FILL THE ODD FILELD  送填充
                                                     字符

F57E AA
F57F 5F          4304        POP     DI
F580 C3          4305        RET                   ; RETURN TO CALLER  返回
                 4306 R18    ENDP
                 4307 ; ……
```

```
4308 ; GRAPHICS WRITE  GRAPHICS WRITE 子程序
4309 ;  THIS ROUTINE WRITES THE ASCII CHARACTER TO THE
       CURRENT  本程序将一ASC II 符显示在屏幕当前位置.
4310 ;  POSITION ON THE SCREEN.
4311 ; ENTRY…入口参数:
4312 ;  AL = CHARACTER TO WRITE  (AL) = 写入字符在字符发生区
       中的位置, { MSB = 1: 第 2 字符缓存
                 { MSB = 0: 第 1 字符缓存
4313 ;  BL = COLOR ATTRIBUTE TO BE USED FOR FOREGROUND
       COLOR  (BL) = 前景色值,(BL)第 7 位为 1, 则字符异或入缓存)
4314 ;       IF BIT 7 IS SET, THE CHAR IS XOR'D INTO THE RE-
             GEN BUFFER
4315 ;       (0 IS USED FOR THE BACKGROUND COLOR)
4316 ;  CX = NUMBER OF CHARS TO WRITE  (CX) = 写入字符的个数
4317 ;  DS = DATA SEGMENT  (DS) = 数据段基址
4318 ;  ES = REGEN SEGMANT  (ES) = 显示缓存基址
4319 ; EXIT…出口参数:
4320 ;  NOTHING IS RETURNED  无
4321 ;
4322 ; GRAPHICS READ  GRAPHICS READ 子程序
4323 ;  THIS ROUTINE READS THE ASCII CHARACTER AT THE
       CURRENT CURSOR  本子程序读取光标处的字符, 然后将其与字
       符发生
4324 ;  POSITION ON THE SCREEN BY MATCHING THE DOTS ON
       THE SCREEN TO THE  区中每个字符点阵比较, 最后返回匹配
       字形号
4325 ;  CHARACTER GENERATOR CODE POINTS
4326 ; ENTRY…入口参数:
4327 ;  NONE (0 IS ASSUMED AS THE BACKGROUND COLOR  无
       (背景色值假设为 0)
4328 ; EXIT…出口参数:
4329 ;  AL = CHARACTER READ AT THAT POSITION(0 RETURNED
       IF NONE FOUND)  (AL):
4330 ;
4331 ; FOR BOTH ROUTINES, THE IMAGES USED TO FORM CHARS
       ARE CONTAINED IN ROM  对上面两子程序, 若不用 1FH 软中
       断将字符
4332 ;  FOR THE 1ST 128 CHARS. T OACCESS CHARS IN THE SEC-
       OND HALF, THE USER  发生区指针指向用户自行定义的字符
```

```
                        发生区
                4333 ; MUST INITIALIZE THE VECTOR AT INTERRUPT 1FH (L-
                       OCATION 0007CH)TO 的话，程序均用第1字符发生区(128字
                       符)。
                4334 ; POINT TO THE USER SUPPLIED TABLE OF GRAPHIC IM-
                       AGES(8×8 BO×ES) 自定义发生区时,应为每个字符规定8×8
                       点阵。
                4335 ; FAILURE TO DO SO WILL CAUSE IN STRANGE RESULTS
                4336 ; ..........................................
                4337            ASSUME  CS:CODE, DS:DATA, ES:DATA
F581            4338 GRAPHICS_WRITE PROC  NEAR
F581 B400       4339            MOV     AH, 0           ; ZERO TO HIGH OF CODE
                                                          POINT
F583 50         4340            PUSH    AX              ; SAVE CODE POINT VALUE
                4341
                4342 ; ......   DETERMINE POSITION IN REGEN BUFFER TO PUT
                                CODE POINTS
                4343
F584 E88501     4344            CALL    S26             ; FIND LOCATION IN REGEN
                                                          BUFFER 取光标在显示缓存
                                                          的位置(字节位置)
F587 8BF8       4345            MOV     DI, AX          ; REGEN POINTER IN DI
                                                          (AX)送DI保存
                4346
                4347 ; ......   DETERMINE REGION TO GET CODE POINTS FROM
                4348
F589 58         4349            POP     AX              ; RECOVER CODE POINT
F58A 3C80       4350            CMP     AL, 80H         ; IS IT IN SECOND HALF
                                                          字符点阵取自第1字符发生区还
                                                          是第2字符发生区?
F58C 7306       4351            JAE     SI              ; YES
                4352
                4353 ; ......   IMAGE IS IN FIRST HALF, CONTAINED IN ROM
                                从第1发生区取字形点阵
                4354
F58E BE6EFA     4355            MOV     SI, 0FA6EH      ; OFFSET CRT_CHAR_GEN-
                                                          OFFSET OF IMAGES 第1发
                                                          生区首址0FA6E送SI
F591 0E         4356            PUSH    CS                      ; SAVE SEGMENT
```

```
                                                    ON STACK 代码段基址送栈
F592 EB0F       4357        JMP     SHORT S2        ; DETERMINE_MODE
                4358

                4359 ;……IMAGE IS IN SECOND HALF, IN USER RAM  从第
                        2 发生区(在用户 RAM)取字形点阵
                4360
F594            4361 S1:                            ; EXTEND_CHAR
F594 2C80       4362        SUB     AL, 80H         ; ZERO ORIGIN FOR SECOND
                                                      HALF  在发生区中的位置值
                                                      减 128，得相对于第二发生区
                                                      首的位置
F596 1E         4363        PUSH    DS              ; SAVE DATA POINTER
F597 2BF6       4364        SUB     SI, SI
F599 8EDE       4365        MOV     DS, SI          ; ESTABLISH VECTOR ADD-
                                                      RESSING  DS 指向向量表首
                4366        ASSUME DS: ABS0
F59B C5367C00   4367        LDS     SI, EXT_PTR     ; GET THE OFFSET OF THE
                                                      TABLE  取第 2 字形发生区
                                                      首址,它由 1FH 中断送入
F59F 8CDA       4368        MOV     DX, DS          ; GET THE SEGMENT OF T-
                                                      HE TABLE
                4369        ASSUME DS: DATA
F5A1 1F         4370        POP     DS              ; RECOVER DATA SEGMENT
F5A2 52         4371        PUSH    DX              ; SAVE TABLE SEGMENT
                                                      ON STACK  第 2 发生区所在
                                                      段的段基压栈
                4372
                4373 ;……DETERMINE GRAPHICS MODE IN OPERATION
                4374
F5A3            4375 S2:                            ; DETERMINE_MODE
F5A3 D1E0       4376        SAL     AX, 1           ; MULTIPLY CODE POINT
                                                      指针值 ×8，因一个字符用 8
                                                      个字节表示
F5A5 D1E0       4377        SAL     AX, 1           ; VALUE BY 8
F5A7 D1E0       4378        SAL     AX, 1
F5A9 03F0       4379        ADD     SI, AX          ; SI HAS OFFSET OF DESI-
                                                      RED CODES  求该字符在发
                                                      生区的位置
```

```
F5AB 8037490006 R 4380      CMP   CRT_MODE, 6
F5B0 1F           4381      POP   DS          ; RECOVER TABLE POINTER
                                                SEGMENT  DS为发生区所在段
                                                的段基
F5B1 722C         4382      JC    S7          ; TEST FOR MEDIUM RESOL-
                                                UTION MODE  320×200模式?
                  4383
                  4384 ;……HIGH RESOLUTION MODE  640×200 模式
F5B3              4385 S3:                    ; HIGH_CHAR
F5B3 57           4386      PUSH  DI          ; SAVE REGEN POINTER
F5B4 56           4387      PUSH  SI          ; SAVE CODE POINTER  保存
                                                DI, SI
F5B5 B604         4388      MOV   DH, 4       ; NUMBER OF TIMES THROU-
                                                GH LOOP  4 次循环送数，每次
                                                送偶奇缓存各一字节
F5B7              4389 S4:
F5B7 AC           4390      LODSB             ; GET BYTE FROM CODE PO-
                                                INTS  取点阵偶字节
F5B8 F6C380       4391      TEST  BL, 80H     ; SHOULD WE USE THE FUN-
                                                CTION  字符异或入显示缓存?
F5BB 7516         4392      JNZ   S6          ; TO PUT CHAR IN  是转 S6
F5BD AA           4393      STOSB             ; STORE IN REGEN BUFFER
                                                否，点阵字节送偶缓存区
F5BE AC           4394      LODSB             取点阵奇字节
F5BF              4395 S5:                    ;
F5BF 268885FF1F   4396      MOV   ES:[DI+2000H-1], AL ; STORE IN SE-
                                                COND HALF
                                                送奇缓存区
F5C4 83C74F       4397      ADD   DI, 79      ; MOVE TO NEXT ROW IN R-
                                                EGEN  (DI)跳过 79 个字节
F5C7 FECE         4398      DEC   DH          ; DONE WITH LOOP  共循环 4
                                                次，送 8 个字节
F5C9 75EC         4399      JNZ   S4
F5CB 5E           4400      POP   SI
F5CC 5F           4401      POP   DI          ; RECOVER REGEN POINTER
                                                恢复字符点阵首字节地址
F5CD 47           4402      INC   DI          ; POINT TO NEXT CHAR POS-
                                                ITION  (DI)加上一个字节，成
                                                为下一显示字符的缓存地址
```

```
F5CE E2E3          4403          LOOP    S3              ; MORE CHARS TO WRITE
                                                           循环,写入数个字符,其个数由
                                                           (CX)决定
F5D0 E9F4FB        4404          JMP     VIDEO_RETURN    返回
                   4405
F5D3               4406 S6:                              字符异或入显示缓存
F5D3 263205        4407          XOR     AL, ES:[DI]     ; EXCLUSIVE OR WITH CU-
                                                           RRENT 与当前位置上字符的
                                                           偶字节相异或
F5D6 AA            4408          STOSB                   ; STORE THE CODE POINT
                                                           送入偶区
F5D7 AC            4409          LODSB                   ; AGAIN FOR ODD FIELD
                                                           取奇字节
F5D8 263285FF1F    4410          XOR     AL, ES:[DI+2000H-1]  ; 与当前位置上字符
                                                                 的奇字节相异或
F5DD EBE0          4411          JMP     S5              ; BACK TO MAINSTREAM
                                                           转 S5 写入奇区
                   4412
                   4413 ;……MEDIUM RESOLUTION WRITE 320×200 模式
F5DF               4414 S7:                              ; MED_RES_WRITE
F5DF 8AD3          4415          MOV     DL, BL          ; SAVE HIGH COLOR BIT
                                                           前景色(欲显示字形的颜色)送
                                                           DL
F5E1 D1E7          4416          SAL     DI, 1           ; OFFSET×2 SINCE 2 BYTES
                                                           /CHAR 偏差×2,因一字形
                                                           需 2 字节记录其一个点行(共
                                                           8 个点行)
F5E3 E8D100        4417          CALL    S19             ; EXPAND BL TO FULL W-
                                                           ORD OF COLOR 色值扩展,
                                                           见 4551 行 S19 子程序说明
F5E6               4418 S8:                              ; MED_CHAR
F5E6 57            4419          PUSH    DI              ; SAVE REGEN POINTER
F5E7 56            4420          PUSH    SI              ; SAVE THE CODE POINTER
                                                           保存 DI, SI,他们是字形写入
                                                           缓存区的地址和在发生表中的
                                                           地址
F5E8 B604          4421          MOV     DH, 4           ; NUMBER OF LOOPS 4 次
                                                           循环送数,每次送偶、奇缓存各
                                                           一字节
```

```
F5EA                4422 S9:
F5EA AC             4423        LODSB                   ; GET CODE POINT 取偶字节
F5EB E8DE00         4424        CALL    S21             ; DOUBLE UP ALL THE BITS 扩展成字
F5EE 23C3           4425        AND     AX, BX          ; CONVERT THEM TO FOREGROUND COLOR (0 BACK) 字形染上(BX)指定的颜色
F5F0 F6C280         4426        TEST    DL, 80H         ; IS THIS XOR FUNCTION 需异或入缓存?
F5F3 7407           4427        JZ      S10             ; NO, STORE IT IN AS IT IS
F5F5 263225         4428        XOR     AH, ES:[DI]     ; DO FUNCTION WITH HALF 需要
F5F8 26324501       4429        XOR     AL, ES:[DI+1]   ; AND WITH OTHER HALF 异或操作
F5FC                4430 S10:                           ;
F5FC 268825         4431        MOV     ES:[DI], AH     ; STORE FIRST BYTE 异或后的字(代表字形的偶点行)送偶显示缓存
F5FF 26884501       4432        MOV     ES:[DI+1], AL   ; STORE SECOND BYTE
F603AC              4433        LODSB                   ; GET CODE POINT 取奇字节
F604 E8C500         4434        CALL    S21             扩展成字
F607 23C3           4435        AND     AX, BX          ; CONVERT TO COLOR 字形奇字节染上(BX)指定的颜色
F609 F6C280         4436        TEST    DL, 80H         ; AGAIN, IS THIS XOR FUNCTION 异或入缓存?
F60C 740A           4437        JZ      S11             ; NO, JUST STORE THE VALUES
F60E 2632A50020     4438        XOR     AH, ES:[DI+2000H]   ; FUNCTION WITH FIRST HALF 是,进行奇字节异或
F613 2632850120     4439        XOR     AL, ES:[DI+2001H]   ; AND WITH SECOND HALF
F618                4440 S11:                           ;
```

```
F618 2688A50020   4441          MOV   ES:[DI+2000H], AH
F61D 2688850120   4442          MOV   ES:[DI+2000H+1], AL
                                      ; STORE IN SECOND PORTION
                                        OF BUFFER  奇字节送奇缓存
                                        区
F622 83C750       4443          ADD   DI, 80   ; POINT TO NEXT LOCATION
                                        (DI) 指向下一点行第一字节
F625 FECE         4444          DEC   DH
F627 75C1         4445          JNZ   S9       ; KEEP GOING  循环4次,送入
                                        16个字节
F629 5E           4446          POP   SI       ; RECOVER CODE PONTER
                                        恢复字符点阵首址
F62A 5F           4447          POP   DI       ; RECOVER REGEN POINTER
                                        恢复字符在偶显示缓存的首址
F62B 83C702       4448          ADD   DI, 2    ; POINT TO NEXT CHAR PO-
                                        SITION  指向下一字符存贮位
                                        置
F62E E2B6         4449          LOOP  S8       ; MORE TO WRITE  循环写
                                        入数个字符,个数由(CX)决定
F630 E994FB       4450          JMP   VIDEO_RETURN  转VIDEO_RETURN
                                        返回
                  4451 GRAPHICS_WRITE ENDP
                  4452 ; ..........
                  4453 ; GRAPHICS READ  下面是读字形程序
                  4454 ; ..........
F633              4455 GRAPHICS_READ  PROC  NEAR
F633 E8D600       4456          CALL  S26      ; CONVERTED TO OFFSET IN
                                        REGEN  光标在显示缓存的位
                                        置(字节位置)送AX
F636 8BF0         4457          MOV   SI, AX   ; SAVE IN SI  (AX)送SI
F638 83EC08       4458          SUB   SP, 8    ; ALLOCATE SPACE TO SAVE
                                        THE READ CODE POINT
                                        栈顶空出8个字节,作为读出字
                                        形的存放区
F63B 8BEC         4459          MOV   BP, SP   ; POINTER TO SAVE AREA
                                        重设栈指针
                  4460
                  4461 ; ..... DETERMINE GRAPHICS MODES
                  4462
```

```
F63D 803E490006 R 4463      CMP   CRT_MODE, 6    320×200 模式还是 640×
                                                 200 模式?
F642 06           4464      PUSH  ES
F643 1F           4465      POP   DS             ; POINT TO REGEN SEG-
                                                   MENT  DS, ES 指向显示
                                                   缓存首
F644 721A         4466      JC    S13            ; MEDIUM RESOLUTION
                                                   320×200 模式转 S13
                  4467
                  4468 ;……HIGH RESOLUTION READ  640×200 模式
                  4469
                  4470 ;……GET VALUES FROM REGEN BUFFER AND CON-
                          VERT TO CODE POINT
F646 B604         4471      MOV   DH, 4          ; NUMBER OF PASSES 循
                                                   环次数。共取 4 次 8 字节
F648              4472 S12:
F648 8A04         4473      MOV   AL, [SI]       ; GET FIRST BYTE  取字
                                                   形偶字节并压栈
F64A 884600       4474      MOV   [BP], AL       ; SAVE IN STORAGE AR-
                                                   EA
F64D 45           4475      INC   BP             ; NEXT LOCATION  栈指
                                                   针加 1
F64E 8A840020     4476      MOV   AL, [SI+2000H] ; GET LOWER REGION
                                                   BYTE  取字形奇字节
                                                   并压栈
F652 884600       4477      MOV   [BP], AL       ; ADJUST AND STORE
F655 45           4478      INC   BP               栈指针加 1
F656 83C650       4479      ADD   SI, 80         ; POINTER INTO REGEN
                                                   跳过一点行
F659 FECE         4480      DEC   DH             ; LOOP CONTROL
F65B 75EB         4481      JNZ   S12            ; DO IT SOME MORE  8字
                                                   节都取到了?
F65D EB1790       4482      JMP   S15            ; GO MATCH THE SAVED
                                                   CODE POINTS  是,转 S15
                  4483
                  4484 ;……MEDIUM RESOLUTION READ  320×200 模式
F660              4485 S13:                      ; MED_RES_READ
F660 D1E6         4486      SAL   SI, 1          ; OFFSET×2 SINCE 2 BY-
                                                   TES/CHAR  位置值×2,
```

```
                                                    因320×200模式下,每个字形点
                                                    行需2个字节
F662 B604       4487        MOV     DH, 4       ; NUMBER OF PASSES  共取4
                                                    次, 16字节
F664            4488 S14:
F664 E88800     4489        CALL    S23         ; GET PAIR BYTES FROM RE-
                                                    GEN INTO SINGLE SAVE
                                                    取偶字节, 压缩成一个字节后送
                                                    栈
F667 81C60020   4490        ADD     SI, 2000H   ; GO TO LOWER REGION  指
                                                    向偶缓存区
F66B E88100     4491        CALL    S23         ; GET THIS PAIR INTO SAVE
                                                    取奇字节, 压缩成一个字节后送
                                                    栈
F66E 81EEB01F   4492        SUB     SI, 2000H-80 ; ADJUST POINTER BACK IN-
                                                    TO UPPER  (SI)恢复指向偶缓
                                                    存区
F672 FECE       4493        DEC     DH
F674 75EE       4494        JNZ     S14         ; KEEP GOING UNTIL ALL 8
                                                    DONE  16字节均取到了?
                4495
                4496 ;……SAVE AREA HAS CHARACTER IN IT, MATCH IT  8
                                                    个字形字节在堆栈中
F676            4497 S15:                       ; FIND_CHAR
F676 BF6EFA     4498        MOV     DI, 0FA6EH  ; OFFSET CRT_CHAR_GEN-
                                                    ESTABLISH ADDRESSING
                                                    第一字符发生区首址送DI
F679 0E         4499        PUSH    CS
F67A 07         4500        POP     ES          ; CODE POINTS IN CS  ES指
                                                    向代码段,作为DI的自动段基
F67B 83ED08     4501        SUB     BP, 8       ; ADJUST POINTER TO BEGI-
                                                    NNING OF SAVE AREA  BP
                                                    指向字形第一字节存放处
F67E 8BF5       4502        MOV     SI, BP
F680 FC         4503        CLD                 ; ENSURE DIRECTION  前向标
                                                    记
F681 B000       4504        MOV     AL, 0       ; CURRENT CODE POINT BEI-
                                                    NG MATCHED
F683            4505 S16:
```

```
F683 16          4506          PUSH    SS          ; ESTABLISH ADDRESSING TO
                                                       STACK
F684 1F          4507          POP     DS          ; FOR THE STRING COMPARE
                                                       (DS)是SI的自动段基
F685 BA8000      4508          MOV     DX, 128     ; NUMBER TO TEST AGAINST
F688             4509 S17:
F688 56          4510          PUSH    SI          ; SAVE SAVE AREA POINTER
                                                       保存字形字节保留区指针
F689 57          4511          PUSH    DI          ; SAVE CODE POINTER 保存字
                                                       形(符)发生区中的地址
F68A B90800      4512          MOV     CX, 8       ; NUMBER OF BYTES TO MATCH
                                                       一次比较8个字节即一个字形的全
                                                       部点
F68D F3          4513          REPE    CMPSB       ; COMPARE THE 8 BYTES 比较
                                                       中有一个字节不相同就退出
F68E A6
F68F 5F          4514          POP     DI          ; RECOVER THE POINTERS 恢
                                                       复DI, SI
F690 5E          4515          POP     SI
F691 741E        4516          JZ      S18         ; IF ZERO FLAG SET, THEN M-
                                                       ATCH OCCURRED ZF=1则找
                                                       到该字形
F693 FEC0        4517          INC     AL          ; NO MATCH, MOVE ON TO NE-
                                                       XT ZF=0, 再找,AL中是欲比较
                                                       字形的地址
F695 83C708      4518          ADD     DI, 8       ; NEXT CODE POINT DI 指向下
                                                       一个字形点阵的首字节
F698 4A          4519          DEC     DX          ; LOOP CONTROL 查完整个发生
                                                       区?
F699 75ED        4520          JNZ     S17         ; DO ALL OF THEM 未查完转
                                                       S17再查
                 4521
                 4522 ;……CHAR NOT MATCHED, MIGHT BE IN USER SUPPLI-
                         ED SECOND HALF
                 4523
F69B 3C00        4524          CMP     AL, 0       ; AL<>0 IF ONLY 1ST HALF S-
                                                       CANNED 是否查过第2字形发生
                                                       区?(AL)≠0说明未查过
```

```
F69D 7412        4525          JE      S18          ; IF=0, THEN ALL HAS BEEN SCANNED 查过转S18
F69F 2BC0        4526          SUB     AX, AX       清 AX
F6A1 8ED8        4527          MOV     DS, AX       ; ESTABLISH ADDRESSING TO VECTOR DS指向向量表首
                 4528          ASSUME  DS:ABS0
F6A3 C43E7C00    4529          LES     DI, EXT_PTR  ; GET POINTER 取第2字形发生区首址
F6A7 8CC0        4530          MOV     AX, ES       ; SEE IF THE POINTER REALLY EXISTS
F6A9 0BC7        4531          OR      AX, DI       ; IF ALL 0, THEN DOESN'T EXIST (ES), (EXT_PTR)均=0表示第二发生区不存在
F6AB 7404        4532          JZ      S18          ; NO SENSE LOOKING 不存在转S18
F6AD B080        4533          MOV     AL, 128      ; ORIGIN FOR SECOND HALF 起始字形号送AL
F6AF EBD2        4534          JMP     S16          ; GO BACK AND TRY FOR IT 转S16查第2发生区
                 4535          ASSUME  DS:DATA
                 4536
                 4537 ; ..... CHARACTER IS FOUND (AL=0 IF NOT FOUND)
F6B1             4538 S18:
F6B1 83C408      4539          ADD     SP, 8        ; READJUST THE STACK, THROW AWAY SAVE 恢复栈指针的原来位置
F6B4 E910FB      4540          JMP     VIDEO_RETURN ; ALL DONE 转VIDEO_RETURN返回
                 4541 GRAPHICS_READ  ENDP
                 4542 ; ..........................................
                 4543 ; EXPAND_MED_COLOR EXPAN_MED_COLOR 子程序
                 4544 ; THIS ROUTINE EXPANDS THE LOW 2 BITS IN BL TO
```

```
                        本程序将320×200模式用的点色值扩展成一个字。
                4545 ; FILL THE ENTIRE BX REGISTER  如0000 0010扩展成10 10
                       10 10 10 10 10 10
                4546 ; ENTRY…入口参数:
                4547 ; BL=COLOR TO BE USED(LOW 2 BITS)  (BL)=色值(在低2
                       位)
                4548 ; EXIT…出口参数:
                4549 ; BX=COLOR TO BE USED(8 REPLICATIONS OF THE 2 CO-
                       LOR BITS)  (BX)=扩展的点色字
                4550 ; ……
F6B7            4551 S19    PROC   NEAR
F6B7 80E303     4552        AND    BL, 3      ; ISOLATE THE COLOR BITS  保留
                                                  色值
F6BA 8AC3       4553        MOV    AL, BL     ; COPY TO AL
F6BC 51         4554        PUSH   CX         ; SAVE RLGISTER  保存 CX
F6BD B90300     4555        MOV    CX, 3      ; NUMBER OF TIMES TO DO THIS
                                                  左移次数
F6C0            4556 S20:
F6C0 D0E0       4557        SAL    AL, 1
F6C2 D0E0       4558        SAL    AL, 1      ; LEFT SHIFT BY 2  色值左移2位
F6C4 0AD8       4559        OR     BL, AL     ; ANOTHER COLOR VERSION INTO
                                                  BL  扩展成半个字节、六位、字节
F6C6 E2F8       4560        LOOP   S20        ; FILL ALL OF BL
F6C8 8AFB       4561        MOV    BH, BL     ; FILL UPPER PORTION  再扩展成
                                                  字
F6CA 59         4562        POP    CX         ; REGISTER BACK
F6CB C3         4563        RET               ; ALL DONE  返回
                4564 S19    ENDP
                4565 ; ……
                4566 ; EXPAND_BYTE  EXPAND_BYTE 子程序
                4567 ;  THIS ROUTINE TAKES THE BYTE IN AL AND DOUBLES
                       ALL 本程序将字节扩展成字,如abcdefgh扩展成aabbccddeeffgghh
                4568 ;  OF THE BITS, TURNING THE 8 BITS INTO 16 BITS  入口
                       参数:(AL)=待扩字节
                4569 ;  THE RESULT IS LEFT IN AX  出口参数:(AX)=扩展字
                4570 ; ……
F6CC            4571 S21    PROC   NEAR
F6CC 52         4572        PUSH   DX         ; SAVE REGISTERS  保存DX, CX,
                                                  BX
```

```
F6CD 51          4573          PUSH   CX
F6CE 53          4574          PUSH   BX
F6CF BA0000      4575          MOV    DX, 0      ; RESULT REGISTER  结果寄存器
                                                   清 0
F6D2 B90100      4576          MOV    CX, 1      ; MASK REGISTER  屏蔽寄存器置
                                                   初值 1
F6D5             4577 S22;
F6D5 8BD8        4578          MOV    BX, AX     ; BASE INTO TEMP
F6D7 23D9        4579          AND    BX, CX     ; USE MASK TO EXTRACT A
                                                   BIT  利用屏蔽位选出待扩的位
F6D9 0BD3        4580          OR     DX, BX     ; PUT INTO RESULT REGISTER
                                                   送待扩位至结果寄存器
F6DB D1E0        4581          SHL    AX, 1
F6DD D1E1        4582          SHL    CX, 1      ; SHIFT BASE AND MASK BY 1
                                                   左移基、屏蔽寄存器一位
F6DF 8B08        4583          MOV    BX, AX     ; BASE TO TEMP
F6E1 23D9        4584          AND    BX, CX     ; EXTRACT THE SAME BIT  将
                                                   左移的待扩位截下
F6E3 0BD3        4585          OR     DX, BX     ; PUT INTO RESULT  待扩位送结
                                                   果寄存器，完成扩展一位工作

F6E5 D1E1        4586          SHL    CX, 1      ; SHIFT ONLY MASK NOW, MO-
                                                   VING TO NEXT BASE  向左调
                                                   屏蔽位，准备扩下一位
F6E7 73EC        4587          JNC    S22        ; USE MASK BIT COMING OUT
                                                   TO TERMINATE  CY=1 表示屏
                                                   蔽位移出 CX，扩展完成了
F6E9 8BC2        4588          MOV    AX, DX     ; RESULT TO PARM REGISTER
                                                   结果送 AX
F6EB 5B          4589          POP    BX
F6EC 59          4590          POP    CX         ; RECOVER REGISTERS  恢复
F6ED 5A          4591          POP    DX
F6EE C3          4592          RET               ; ALL DONE  返回
                 4593 S21      ENDP
                 4594 ; ..........................................
                 4595 ; MED_READ_BYTE  MEM_READ_BYTE 子程序
                 4596 ; THIS ROUTINE WILL TAKE 2 BYTES FROM THE REGEN
                        BUFFER，本子程序将 320×200 模式下字形的点行的二个字节
                        压缩
```

```
                4597 ; COMPARE AGAINST THE CURRENT FOREGROUND COLO-
                       R, AND PLACE  成一个字节,并存入内存。
                4598 ; THE CORRESPONOING DN/OFF BIT  PATTERN  INTO  THE
                       CURRENT
                4599 ; POSITION IN THE SAVE AREA
                4600 ; ENTRY…入口参数:
                4601 ; SI, DS = POINTER TO REGEN AREA OF INTEREST  (DS)、
                       (SI) = 显示缓存的基址和点行字节在显示缓存的位置
                4602 ; BX = EXPANDED FOREGROUND COLOR  (BX) = 扩展前景色
                       值字
                4603 ; BP = POINTER TO SAVE AREA  (BP) = 压缩结果存放单色指
                       针
                4604 ; EXIT…出口参数:
                4605 ; BP IS INCREMENT AFTER SAVE  BP 内容加 1
                4606 ; ……
F6EF            4607 S23    PROC   NEAR
F6EF 8A24       4608        MOV    AH, [SI]      ; GET FIRST BYTE  取第一,第
                                                     二字节
F6F1 8A4401     4609        MOV    AL, [SI+1]    ; GET SECOND BYTE
F6F4 B900C0     4610        MOV    CX, 0C000H    ; 2 BIT MASK TO TEST THE
                                                     ENTRIES  每个点的色值占 2
                                                     位,(CX)为色值屏蔽寄存器
F6F7 B200       4611        MOV    DL, 0         ; RESULT REGISTER  清结果寄
                                                     存器
F6F9            4612 S24:
F6F9 85C1       4613        TEST   AX, CX        ; IS THIS SECTION BACKGRO-
                                                     UND?  背景色点的色值为 00
F6FB F8         4614        CLC                  ; CLEAR CARRY IN HOPES T-
                                                     HAT IT IS  CY 送 0
F6FC 7401       4615        JZ     S25           ; IF ZERO, IT IS BACKGROU-
                                                     ND  ZF=1 说明是背景色
F6FE F9         4616        STC                  ; WASN'T, SO SET CARRY  非
                                                     背景色, CY 送 1, 表明字形在该
                                                     点有影
F6FF D0D2       4617 S25:   RCL    DL, 1         ; MOVE THAT BIT INTO THE
                                                     RESULT  移入 DL
F701 D1E9       4618        SHR    CX, 1
F703 D1E9       4619        SHR    CX, 1         ; MOVE THE MASK TO THE
                                                     RIGHT BY 2 BITS  向右调整
```

```
                                                    屏蔽位,检查下一色点
F705 73F2       4620            JNC     S24         ; DO IT AGAIN IF MASK D-
                                                    IDN'T FALL OUT 屏蔽位移
                                                    入 CY 位表示压缩完毕
F707 885600     4621            MOV     [BP], DL    ; STORE RESULT IN SAVE
                                                    AREA 结果送内存
F70A 45         4622            INC     BP          ; ADJUST POINTER 指针加1
F70B C3         4623            RET                 ; ALL DONE 返回
                4624 S23        ENDP
                4625 ; ..........................................................
                4626 ; V4_POSITION VS_POSITION 子程序
                4627 ; THIS ROUTINE TAKES THE CURSOR POSITION CONTA-
                       INED IN 本子程序将光标在显示缓存的字节位置返给调用程
                       序
                4628 ; THE MEMORY LOCATION, AND CONVERTS IT INTO AN
                       OFFSET
                4629 ; INTO THE REGEN BUFFER, ASSUMING ONE BYTE/CHAR.
                4630 ; FOR MEDIUM RESOLUTION GRAPHICS, THE NUMBER
                       MUST
                4631 ; BE DOUBLED.
                4632 ; ENTRY…NO REGISTERS, MEMORY LOCATION CURSOR-
                       POSN IS USED 入口参数:无
                4633 ; EXIT…出口参数:
                4634 ; AX CONTAINS OFFSET INTO REGEN BUFFER (AX) =
                       字节位置
                4635 ; ..........................................................
F70C            4636 S26        PROC    NEAR
F70C A15000 R   4637            MOV     AX, CURSOR_POSN ; GET CURRENT
                                                    CURSOR 取光
                                                    标的行、列坐标
                                                    值,这里的行(列)
                                                    指 8 个象点
F70F            4638 GRAPH_POSN LABEL NEAR 行组成的行。
F70F 53         4339            PUSH    BX          ; SAVE REGISTER
F710 8BD8       4640            MOV     BX, AX      ; SAVE A COPY OF CURRE-
                                                    NT CURSOR
F712 8AC4       4641            MOV     AL, AH      ; GET ROWS TO AL 行数送
                                                    AL
F714 F6264A00 R 4642            MUL     BYTE PTR CRT_COLS
```

```
                                        ; MULTIPLY BY BYTES/COLUMN
                                          行数×40(320×200模式)或行数×80
                                          (640×200)
F718 D1E0   4643        SHL     AX, 1   ; MULTIPLY * 4 SINCE 4 ROWS/
                                          BYTE
F71A D1E0   4644        SHL     AX, 1   再乘4，相当于行数×160(320×200)或行
                                          数×320(640×200)，考虑进奇、偶
F71C 2AFF   4645        SUB     BH, BH  ; ISOLATE COLUMN VALUE  显示
                                          缓存的因素
F71E 03C3   4646        ADD     AX, BX  ; DETERMINE OFFSET  加上列坐
                                          标，(AX)为光标在奇或偶显示缓存的
                                          位置(字节位置)
F720 5B     4647        POP     BX      ; RECOVER POINTER
F721 C3     4648        RET             ; ALL DONE  返回
            4649 S26    ENDP
            4650 ; ..........
            4651 ; WRITE_TTY  WRITE_TTY子程序
            4652 ; THIS INTERFACE PROVIDES A TELETYPE LIKE INTERFACE
                   TO THE  本程序是一个类似电传接口的CRT接口，它将字符写在
                   当前
            4653 ;  VIDEO CARD.THE INPUT CHARACTER IS WRITTEN TO T-
                   HE CURRENT  光标处并将光标向右进一字符位置。若光标越出
                   CRT最
            4654 ;  CURSOR POSITION,  AND THE CURSOR IS MOVED TO THE
                   NEXT POSITION.  后列，则光标移向下一行首；若光标越出CRT最
                   后
            4655 ;  IF THE CURSOR LEAVES THE LAST COLUMN OF THE FIELD,
                   THE COLUMN  列，则光标在整个屏幕上卷一行后移到CRT 最后
            4656 ;  IS SET TO ZERO,  AND THE ROW VALUE IS INCREMENT-
                   ED.IF THE  行首，最后行填以光标原位置的字符属性码。在
            4657 ;  ROW VALUE LEAVES THE FIELD,  THE CURSOR IS PLAC-
                   ED ON THE LAST ROW,  图形模式下，空出行填以色值0的背
                   景色。
            4658 ;  FIRST COLUMN,  AND THE ENTIRE SCREEN IS SCROLLED
                   UP ONE LINE
            4659 ;  WHEN THE SCREEN IS SCROLLED UP, THE ATTRIBUTE F-
                   OR FILLING THE
            4660 ;  NEWLY BLANKED LINE IS READ FROM THE CURSOR PO-
```

```
                SITION ON THE PREVIOUS
4661 ;          LINE BEFORE THE SCROLL, IN CHARACTER MODE. IN
                GRAPHICS MODE,
4662 ;          THE 0 COLOR IS USED
4663 ; ENTRY…入口参数:
4664 ;          (AH) = CURRENT CRT MODE   (AH) = 当前 CRT 模式
4665 ;          (AL) = CHARACTER TO BE WRITTEN   (AL) = 待显示字符
4666 ;                NOTE THAT BACK SPACE, CAR RET, BELL AND
                      LINE FEED ARE HANDLED  注意:若显示字符是回格、
                      回车、响铃,换行码的话不显示,把它们作为命令码处理。
4667 ;                AS COMMANDS RATHER THAN AS DISPLAYABLE
                      GRAPHICS
4668 ;          (BL) = FOREGROUND COLOR FOR CHAR WRITE IF CURR-
                      ENTLY IN A GRAPHICS MODE   (BL) = 图形模式下显
                      示字符的颜色值
4669 ; EXIT…出口参数:
4670 ;          ALL REGISTERS SAVED  无,所有寄存器内容均不破坏
4671 ; ……
4672                ASSUME CS:CODE, DS:DATA
F722            4673 WRITE_TTY      PROC  NEAR
F722 50         4674                PUSH  AX          ; SAVE REGISTERS
F723 50         4675                PUSH  AX          ; SAVE CHAR TO WRITE  待显
                                                      示字符两次压栈
F724 B403       4676                MOV   AH, 3
F726 CD10       4677                INT   10H         ; READ THE CURRENT CURSOR
                                                      POSITION  发软中断 10H,  读当
                                                      前光标位置(由 DX 返回)
F728 58         4678                POP   AX          ; RECOVER CHAR
                4679
                4680 ; ……     DX NOW HAS THE CURRENT CURSOR POSITION
                4681
F729 3C08       4682                CMP   AL, 8       ; IS IT A BACKSPACE  回格码?
F72B 7459       4683                JE    U8          ; BACK_SPACE
F72D 3C0D       4684                CMP   AL, 0DH     ; IS IT CARRIAGE RETURN  否,
                                                      回车码?
F72F 745E       4685                JE    U9          ; CAR_RET
F731 3C0A       4686                CMP   AL, 0AH     ; IS IT A LINE FEED  否, 换行
                                                      码?
F733 745E       4687                JE    U10         ; LINE_FEED
```

```
F735 3C07          4688      CMP     AL, 07H              ; IS IT A BELL 否,响
                                                          铃码?
F737 7461          4689      JE      U11                  ; BELL
                   4690
                   4691 ; ·····WRITE THE CHAR TO THE SCREEN
                   4692
F739 8A3E6200 R    4693      MOV     BH,ACTIVE_PAGE       ; GET THE CURRENT
                                                          ACTIVE PAGE 否,
                                                          取当前活动显示页页号
F73D B40A          4694      MOV     AH, 10               ; WRITE CHAR ONLY
F73F B90100        4695      MOV     CX, 1                ; ONLY ONE CHAR
F742 CD10          4896      INT     10H                  ; WRITE THE CHAR
                                                          发软中断 10, 将 AL 中
                                                          的字符显示在光标处,
                                                          中断返回时 (DL) 为光
                                                          标的行坐标,(DH)为光
                                                          标的列坐标
                   4697
                   4698 ;·····POSITION THE CURSOR FOR NEXT CHAR
                   4699
F744 FEC2          4700      INC     DL   (DL)指向同行下一个字符位置
F746 3A164A00 R    4701      CMP     DL,BYTE PTR CRT_COLS ; TEST FOR C-
                                                          OLUMN OV-
                                                          ERFLOW 光
                                                          标是否越出屏
                                                          幕右边界 (或
                                                          列溢出)了?
F74A 7536          4702      JNZ     U7                   ; SET_CURSOR 否,转
                                                          U7
F74C B200          4703      MOV     DL, 0                ; COLUMN FOR CURS-
                                                          OR 列溢出,DL 送 0,
                                                          准备指向下一行首
F74E 80FE18        4704      CMP     DH, 24   光标是否越出屏幕下边界(或行溢出)?
F751 752D          4705      JNZ     U6                   ; SET_CURSOR_INC
                                                          否,转 U6
                   4706
                   4707 ;·····SCROLL REQUIRED  光标越出下边界,应上卷一行
F753               4708 U1:
                   4709
```

```
F753 B402          4710          MOV     AH, 2                   设光标入口参数0显示页
F755 B700          4711          MOV     BH, 0
F757 CD10          4712          INT     10H                     ; SET THE CURSOR 将
                                                                   光标置在屏幕最下行首
                   4713
                   4714 ;……DETERMINE VALUE TO FILL WITH DURING SCR-
                                 OLL
                   4715
F759 A04900      R 4716          MOV     AL, CRT_MODE            ; GET THE CURRENT
                                                                   MODE 当前CRT模式
                                                                   字节送AL
F75C 3C04          4717          CMP     AL, 4
F75E 7206          4718          JC      U2                      ; READ-CURSOR 字符
                                                                   模式转U2
F760 3C07          4719          CMP     AL, 7                   外接单色转接口?
F762 B700          4720          MOV     BH, 0                   ; FILL WITH BACKGRO-
                                                                   UND 若图形模式则最后
                                                                   行填背景色字符模式
F764 7506          4721          JNE     U3                      ; SCROLL-UP
                                                                   读当前光标处的字符码和
                   4722                                          属性码。
F766               4723 U2:                                      ; READ-CURSOR
F766 B408          4724          MOV     AH, 8
F768 CD10          4725          INT     10H                     ; READ CHAR/ATTR AT
                                                                   CURRENT CURSOR 读
                                                                   字符码/属性码
F76A 8AFC          4726          MOV     BH, AH                  ; STORE IN BH 属性码
                                                                   送BH
                   4727
F76C               4728 U3:                                      ; SCROLL-UP
F76C B80106        4729          MOV     AX, 601H                ; SCROLL ONE LINE
                                                                   (AH)=6, (AL)=1 上卷
                                                                   屏幕一行
F76F B90000        4730          MOV     CX, 0                   ; UPPER LEFT CORNER
                                                                   屏幕左上角地址
F772 B618          4731          MOV     DH, 24                  ; LOWER RIGHT ROW
                                                                   屏幕最后行行号也即列坐
                                                                   标送DH
F774 8A164A00    R 4732          MOV     DL, BYTE PTR CRT_COLS
                                                                   ; LOWER RIGHT COLUMN
```

```
F778 FECA      4733        DEC     DL                  屏幕最右列列号也即行坐标送 DL
F77A           4734 U4:                                ; VIDED_CALL_RETURN
F77A CD10      4735        INT     10H                 ; SCROLL UP THE SCREEN
                                                       上卷屏幕一行，空出行填以
                                                       (BH)
F77C           4736 U5:                                ; TTY_RETURN
F77C 58        4737        POP     AX                  ; RESTORE THE CHARACT-
                                                       ER

LOC OBJ        LINE  SOURCE
F77D E947FA    4738        JMP     VIDEO_RETURN        ; RETURN TO CALLER  转
                                                       VIDEO_RETURN 准备返回
               4739
F780           4740 U6:                                ; SET_CURSOR_INC  光标
                                                       未越出屏幕最下行，只需将行
                                                       号加 1
F780 FEC6      4741        INC     DH                  ; NEXT ROW  (DH) 指向下一
                                                       行
F782           4742 U7:                                ; SET_CURSOR
F782 B402      4743        MOV     AH, 2               设光标入口参数
F784 EBF4      4744        JMP     U4                  ; ESTABLISH THE NEW C-
                                                       URSOR
               4746 ;……BACK SPACE FOUND                回格(退格)处理
               4747
F786           4748 U8:
F786 80FA00    4749        CMP     DL, 0               ; ALREADY AT END OF L-
                                                       INE  现是否处第 0 列?
F789 74F7      4750        JE      U7                  ; SET_CURSOR  光标保持在
                                                       行首
F78B FECA      4751        DEC     DL                  ; NO…JUST MOVE IT BACK
                                                       否，列号减 1，表示在同一行退
                                                       一格
F78D EBF3      4752        JMP     U7                  ; SET_CURSOR
               4753
               4754 ;……CARRIAGE RETURN FOUND  回车处理
               4755
F78F           4756 U9:
F78F B200      4757        MOV     DL, 0               ; MOVE TO FIRST COLUMN
                                                       光标返回至第一列
F791 EBEF      4758        JMP     U7                  ; SET_CURSOR
```

```
                4759
                4760 ; ……      LINE FEED FOUND  换行处理
                4761
F793            4762 U10:
F793 80FE18     4763            CMP     DH, 24  ; BOTTOM OF SCREEN  光标正处
                                                  最下行?
F796 75E8       4764            JNE     U6      ; YES, SCROLL THE SCREEN
                                                  正处最下行，需转 U1 上卷屏幕
                                                  上行
F798 EBB9       4765            JMP     U1      ; NO, JUST SET THE CURSOR
                4766
                4767 ; ……      BELL FOUND      响铃处理
                4768
F79A            4769 U11:
F79A B302       4770            MOV     BL, 2   ; SET UP COUNT FOR BEEP
F79C E8C7EE     4771            CALL    BEEP    ; SOUND THE POD BELL  调蜂鸣
                                                  程序
F79F EBDB       4772            JMP     U5      ; TTY_RETURN
                4773 WRITE_TTY  ENDP
                4774 ; ……
                4775 ; LIGHT PEN  LIGHT PEN 子程序
                4776 ;          THIS ROUTINE TESTS THE LIGHT PEN SWITCH
                                AND THE LIGHT  本子程序测试光笔的开关状态和触
                                发器状态，如果两
                4777 ;            PEN TRIGGER. IF BOTH ARE SET, THE LOCA-
                                  TION OF THE LIGHT  者均置位了，本程序返回光
                                  笔当前处的位置，否则
                4778 ;            PEN IS DETERMINED.OTHERWISE, A RETURN
                                  WITH NO INFORMATION  不返回任何信息。
                4779 ;            IS MADE.  入口参数：
                                              无
                4780 ;          ON EXIT:  出口参数：
                4781 ;          (AH) = 0 IF NO LIGHT PEN INFORMATION IS A-
                                        VAILABLE  (AH) = 0 光笔无作用，BX, CX,
                                        DX 内容被破坏
                4782 ;                  BX, CX, DX ARE DESTROYED
                4783 ;          (AH) = 1 IF LIGHT PEN IS AVAILABLE  (AH) = 1
                                        光笔可用
                4784 ;                  (DH, DL) = ROW, COLUMN OF CURRENT
```

```
                        ;               LIGHT PEN POSITION  (DH, DL) = 光笔当
                                        前坐标
                        4785 ;          (CH) = RASTER POSITION
                                        (CH) = 光栅线号
                        4786 ;          (BX) = BEST GUESS AT PIXEL
                                        HORIZONTAL POSITION  (BX)
                                        = 点列号最佳估计值
                        4787 ; ..........................................
                        4788            ASSUME CS:CODE, DS:DATA
                        4789 ; ......   SUBTRACT_TABLE
F7A1                    4790 V1         LABEL  BYTE
F7A1 0303050503030304   4791            DB     3, 3, 5, 5, 3, 3, 3, 4 ;
F7A9                    4792 READ_LPEN  PROC   NEAR
                        4793
                        4794 ; ......   WAIT FOR LIGHT PEN TO BE DEPRE-
                                        SSED
                        4795
F7A9 B400               4796            MOV    AH, 0    ; SET NO LIGHT PEN
                                                          RETURN CODE  清
                                                          AH
F7AB 8B166300       R   4797            MOV    DX, ADDR_6845
                                                        ; GET BASE ADDRESS
                                                          OF 6845
F7AF 83C206             4798            ADD    DX, 6    ; POINT TO STATUS
                                                          REGISTER  (DX) =
                                                          6845 状态寄存器地址
F7B2 EC                 4799            IN     AL, DX   ; GET STATUS REGI-
                                                          STER  读 6845 状态,
                                                          其第 1 位为 1 表示光笔
                                                          触发器置位，第 2 位为
                                                          0 表示光笔开关打开
F7B3 A804               4800            TEST   AL, 4    ; TEST LIGHT PEN S-
                                                          WITCH
F7B5 7578               4801            JNZ    V6       ; NOT SET, RETURN
                                                          光笔开关打开?
                        4802
                        4803 ;          NOW TEST FOR LIGHT PEN TRIGGER
                                        光笔开关打开了，现查光笔触发器
                        4804
```

```
F7B7 A802          4805    TEST    AL, 2           ; TEST LIGHT PEN TRIGG-
                                                     ER
F7B9 747E          4806    JZ      V7              ; RETURN WITHOUT RES-
                                                     ETTING TRIGGER  触发
                                                     器触发了?
                   4807
                   4808 ;…TRIGGER HAS BEEN SET, READ THE VALUE IN  光
                           笔开关,触发器状态均有效,下面取光笔的位置
                   4809
F7BB B410          4810    MOV     AH, 16          ; LIGHT PEN REGISTERS
                                                     ON 6845  6845 第16、17
                                                     号光栅寄存器记录光笔当前
                                                     位置
                   4811
                   4812 ;…INPUT REGS POINTED TO BY AH,  AND CONVERT
                           TO ROW COLUMN IN DX
                   4813
F7BD 8B166300 R    4814    MOV     DX, ADDR_6845   ; ADDRESS REGISTER FOR
                                                     6845 取6845基址也即6845
                                                     地址寄存器地址
F7C1 8AC4          4815    MOV     AL, AH          ; REGISTER TO READ  送
                                                     光栅寄存器号16至地址寄
                                                     存器
F7C3 EE            4816    OUT     DX, AL          ; SET IT UP
F7C4 42            4817    INC     DX              ; DATA REGISTER
F7C5 EC            4818    IN      AL, DX          ; GET THE VALUE  读第
                                                     16号光栅寄存器
F7C6 8AE8          4819    MOV     CH, AL          ; SAVE IN CX  读出值送CH
F7C8 4A            4820    DEC     DX              ; ADDRESS REGISTER
F7C9 FEC4          4821    INC     AH
F7CB 8AC4          4822    MOV     AL, AH          ; SECOND DATA REGISTER
F7CD EE            4823    OUT     DX, AL          送光栅寄存器号17至地址寄存器
F7CE 42            4824    INC     DX              ; POINT TO DATA REGIS-
                                                     TER
F7CF EC            4825    IN      AL, DX          ; GET SECOND DATA VA-
                                                     LUE  读第17号光栅寄存器
F7D0 8AE5          4826    MOV     AH, CH          ; AX HAS INPUT VALUE
                                                     (AX) 为读出内容(字节地
```

```
                                                        址)
                        4827
                        4828 ;……AX HAS THE VALUE READ IN FROM THE 6845
                        4829
F7D2 8A1E4900    R      4830        MOV     BL, CRT_MODE                模式送 BL
F7D6 2AFF               4831        SUB     BH, BH          ; MODE VALUE TO BX
F7D8 2E8A9FA1F7  R      4832        MOV     BL, CS:V1[BX]   ; DETERMINE AMOU-
                                                            NT TO SUBTRACT
                                                            取调整值
F7DD 2BC3               4833    SUB     AX, BX              ; TAKE IT AWAY  减
                                                            某常数
F7DF 2B064E00    R      4834        SUB     AX, CRT_START   ; CONVERT TO CORR-
                                                            ECT PAGE ORIGIN
                                                            减显示缓存首址值
F7E3 7903               4835        JNS     V2              ; IF POSITIVE, DETE-
                                                            RMINE MODE  减后
                                                            剩余值大于等于 0 转
                                                            V2
F7E5 B80000             4836        MOV     AX, 0           ; <0 PLAYS AS 0  <0
                                                            清 AX
                        4837
                        4838 ;……DETERMINE MODE OF OPERATION
                        4839
F7E8                    4840 V2:                            ; DETERMINE_MODE
                                                            下面检查模式
F7E8 B103               4841        MOV     CL, 3           ; SET ×8 SHIFT CO-
                                                            UNT
F7EA 803E490004  R      4842        CMP     CRT_MODE, 4     ; DETERMINE IF GR-
                                                            APHICS OR ALPHA
F7EF 722A               4843        JB      V4              ; ALPHA_PEN
F7F1 803E490007  R      4844        CMP     CRT_MODE, 7
F7F6 7423               4845        JE      V4              ; ALPHA_PEN
                        4846
                        4847 ;……GRAPHICS MODE                       图形模式
                        4848
F7F8 B228               4849        MOV     DL, 40          ; DIVISOR FOR GRAP-
                                                            HICS  调整后的字节
                                                            地址÷40
F7FA F6F2               4850        DIV     DL              ; DETERMINE ROW
```

```
                                                    (AL)  AND COLUMN
                                                    (AH) (AH) 为水平方向
                                                    光笔所在位置的字符号,
                                                    (AL) 为垂直方向
                        4851                        ; AL RANGE 0-99, AH
                                                    RANGE 0-39 光笔所在
                                                    行行号(只对应于奇/偶缓
                                                    存)(AH) = [0, 39], (AL)
                                                    = [0, 99]
                        4852 ;……DETERMINE GRAPHIC ROW POSITION
                        4853
F7FC 8AE8               4854    MOV   CH, AL        ; SAVE ROW VALUE IN
                                                    CH
F7FE 02ED               4855    ADD   CH, CH        ; ×2 FOR EVEN/ODD F-
                                                    IELD 行号×2, 考虑逃
                                                    奇,偶缓存因素
F800 8ADC               4856    MOV   BL, AH        ; COLUMN VALUE TO BX
                                                    列字符坐标送 BL
F802 2AFF               4857    SUB   BH, BH        ; MULTIPLY BY 8 FOR
                                                    MEDIUM RES
F804 803E490006   R     4858    CMP   CRT_MODE, 6   ; DETERMINE MEDIUM
                                                    OR HIGH RES 640×
                                                    200 模式?
F809 7504               4859    JNE   V3            ; NOT_HIGH_RES
F80B B104               4860    MOV   CL, 4         ; SHIFT VALUE FOR H-
                                                    IGH RES 640×200 模
                                                    式,列字符坐标应×16 才
                                                    等于垂直点列号
F80D D0E4               4861    SAL   AH, 1         ; COLUMN VALUE TIM-
                                                    ES 2 FOR HIGH RES
                                                    640×200模式每行可显示
                                                    80 个字符, 故应将 (AH)
                                                    ×2
F80F                    4862 V3:                    ; NOT_HIGH_RES
F80F D3E3               4863    SHL   BX, CL        ; MULTIPLY * 16 FOR
                                                    HIGH RES 640×200,
                                                    字符坐标 ×16 或 320×
                                                    200, 字符坐标×8, 均得
                        4864                        点列的估计值
```

```
                        4865 ;……DETERMINE ALPHA CHAR POSITION
                        4866
F811 8AD4               4867        MOV     DL, AH      ; COLUMN VALUE FOR RETU-
                                                          RN  字符列号送 DX
F813 8AF0               4868        MOV     DH, AL      ; ROW VALUE
F815 D0EE               4869        SHR     DH, 1       ; DIVIDE BY 4 因图形模式 CRT
                                                          垂直方向只有 200 线，每个字符
                                                          垂直方向需 8 根线，故还应
F817 D0EE               4870        SHR     DH, 1       ; FOR VALUE IN 0-24 RANGE
                                                          调整行号，使其落入[0,[199/8]]即
                                                          [0, 24]范围
F819 EB12               4871        JMP     SHORT V5;   LIGHT_PEN_RETURN_ SET
                                                          返回
                        4872
                        4873 ;……ALPHA MODE ON LIGHT PEN  字符模式
                        4874
F81B                    4875 V4:                        ; ALPHA_PEN
F81B F6364A00      R    4876        DIV     BYTE PTR CRT_COLS ; DETERMINE
                                                          ROW, COLUMN
                                                          VALUE  除每
                                                          行字符数，得光
                                                          笔处字符的行、
                                                          列号
F81F 8AF0               4877        MOV     DH, AL      ; ROWS TO DH  行号送 DH
F821 8AD4               4878        MOV     DL, AH      ; COLS TO DL  列号送 DL
F823 D2E0               4879        SAL     AL, CL      ; MULTIPLY ROWS  ×8  行号
                                                          ×8 = 水平扫描线号
F825 8AE8               4880        MOV     CH, AL      ; GET RASTER VALUE TO R-
                                                          ETURN REG
F827 8ADC               4881        MOV     BL, AH      ; COLUMN VALUE
F829 32FF               4882        XOR     BH, BH      ; TO BX
F82B D3E3               4883        SAL     BX, CL      垂直方向点列号(估计值)
F82D                    4884 V5:                        ; LIGHT_PEN_RETURN_ SET
F82D B401               4885        MOV     AH, 1       ; INDICATE EVERTHING SET
                                                          置光笔有效标记
F82F                    4886 V6:                        ; LIGHT_PEN_RETURN
F82F 52                 4887        PUSH    DX          ; SAVE RETURN VALUE (IN
                                                          CASE)  保存光笔坐标
```

```
F830 8B166300  R 4888        MOV  DX, ADDR_6845  ; GET BASE ADDRESS
F834 83C207      4889        ADD  DX, 7          ; POINT TO RESET PARM
                                                   DX 指向 6845 光笔锁存器
F837 EE          4890        OUT  DX, AL         ; ADDRESS, NOT DATA,
                                                   IS IMPORTANT 利用OUT
                                                   指令的地址清光笔锁存器,
                                                   数据与清除无关

F838 5A          4891        POP  DX             ; RECOVER VALUE 恢复
                                                   坐标值
F839             4892 V7,                        ; RETURN_NO_RESET
F839 5F          4893        POP  DI
F83A 5E          4894        POP  SI
F83B 1F          4895        POP  DS             ; DISCARD SAVED BX,
                                                   CX, DX 扔掉进入时保存
                                                   的 BX, CX, DX 内容
F83C 1F          4896        POP  DS
F83D 1F          4897        POP  DS
F83E 1F          4898        POP  DS
F83F 07          4899        POP  ES               恢复 DI, SI, DS, ES
F840 CF          4900        IRET                  返回
                 4901 READ_LPEN  ENDP
                 4902 ; ……INT 12 ……
                 4903 ; MEMORY_SIZE_DETERMINE 软中断 12_MEMORY_
                        SIZE_DETERMINE 子程序
                 4904 ;   THIS ROUTINE DETERMINES THE AMOUNT OF M-
                          EMORY IN THE SYSTEM 本子程序返回系统内存容
                          量,只有系统板上的
                 4905 ;   AS REPRESENTED BY THE SWITCHES ON THE PL-
                          ANAR. NOTE THAT RAM 满 64KB, 通过 I/O 口扩充
                          的 RAM 才算入
                 4906 ;   THE SYSTEM MAY NOT BE ABLE TO USE I/O M-
                          EMORY UNLESS THERE 系统 RAM 总量。
                 4907 ;   IS A FULL COMPLEMENT OF 64K BYTES ON THE
                          PLANAR。
                 4908 ; INPUT 入口参数:
                 4909 ;   NO REGISTERS 无
                 4910 ;   THE MEMORY_SIZE VARIABLE IS SET DURING
                          POWER ON DIAGNOSTICS 开机时系统根据 8255 PA
```

```
                        口,PC口 的读入数据算出内存容量。
                4911 ;      ACCORDING TO THE FOLLOWING HARDWARE
                            ASSUMPTIONS:
                4912 ;      PORT 60 BITS 3, 2 = 00 - 16K BASE RAM  PA口第
                            3, 2 位 = 00 - 16K 内接 RAM
                4913 ;                          01 - 32K BASE RAM  01 - 32K
                            内接 RAM
                4914 ;                          10 - 48K BASE RAM  10 - 48K
                            内接 RAM
                4915 ;                          11 - 64K BASE RAM  11 - 64K
                            内接 RAM
                4916 ;      PORT 62 BITS 3-0 INDICATE AMOUNT OF I/O
                            RAM IN 32K INCREMENTS  PC 口第3, 2, 1, 0位
                            决定 I/O 扩充 RAM 容
                4917 ;              E. G., 0000 - NO RAM IN I/O CHANNEL
                            量。容量 = 32 × 该 4 位的二进制值 KB。
                4918 ;                    0010 - 64K RAM IN I/O CHANNE-
                            L, ETC.
                4919 ; OUTPUT  出口参数:
                4920 ;      (AX) = NUMBER OF CONTIGUOUS 1K BLOCKS OF
                            MEMORY  (AK) = 内存容量(RAM 区)
                4921 ; ..........................................
                4922        ASSUME CS: CODE, DS: DATA
F841            4923 MEMORY_SIZE_DETERMINE  PROC  FAR
F841 FB         4924        STI                 ; INTERRUPTS BACK ON 允
                            许中断
F842 1E         4925        PUSH  DS            ; SAVE SEGMENT
F843 B84000  R  4926        MOV   AX, DATA      ; ESTABLISH ADDRESSING
F846 8ED8       4927        MOV   DS, AX                DS 指向数据段
F848 A11300  R  4928        MOV   AX, MEMORY_SIZE ; GET VALUE  取
                                                  内存 RAM 区容量
F84B 1F         4929        POP   DS            ; RECOVER SEGMENT
F84C CF         4930        IRET                ; RETURN TO CALLER  返回
                4931 MEMORY_SIZE_DETERMINE  ENDP
                4932 ; ..... INT 11 .....................
                4933 ; EQUIPMENT DETERMINATION  软中断 11_EQUIPMENT
                    DETERMINATION 子程序
                4934 ;      THIS ROUTINE ATTEMPTS TO DETERMINE WH-
                            AT OPTIONAL  本子程序返回一个反映系统外设配接
```

```
                情况的字。
4935 ;          DEVICES ARE ATTACHED TO THE SYSTEM
4936 ; INPUT    入口参数:
4937 ;          NO REGISTERS  无
4938 ;          THE EQUIP_FLAG VARIABLE IS SET DURING THE
                POWER ON DIAGNOSTICS  开机时系统根据下列口子置
                该字(单元)的
4939 ;          USING THE FOLLOWING HARDWARE ASSUMPTIO-
                NS;  相应位:
4940 ;          PORT 60=LOW ORDER BYTE OF EQUPMENT  8255
                PA 口、异步通讯口 8250 中断标识寄存
4941 ;          PORT 3FA=INTERRUPT ID REGISTER OF 8250  器、
                打印机转接器输出口
4942 ;                  BITS 7-3 ARE ALWAYS 0
4943 ;          PORT  378=OUTPUT PORT OF PRINTER…8255 PORT
                THAT
4944 ;                  CAN BE READ AS WELL AS WRITTEN
4945 ; OUTPUT   出口参数:
4946 ;          (AX)IS SET, BIT SIGNIFICANT, TO INDICATE ATT-
                ACHED I/O  配接情况字在 AX 返回,其各位的含义如下:
4947 ;          BIT 15, 14=NUMBER OF PRINTERS ATTACHED
                15, 14 位=打印机台数
4948 ;          BIT 13 NOT USED  13 位无用
4949 ;          BIT 12=GAME I/O ATTACHED  12 位=游戏接口数
4950 ;          BIT 11, 10, 9=NUMBER OF RS232 CARDS ATTACH-
                ED  11, 10, 9 位=RS232 异步通讯口数
4951 ;          BIT 8 UNUSED  8 位无用
4952 ;          BIT 7, 6=NUMBER OF DISKETTE DRIVES  7, 6 位=
                (AX)第 0 位为 1 时软盘机台数
4953 ;                  00=1, 01=2, 10=3, 11=4 ONLY IF BIT 0=1
                        00=1 台, 01=2 台, 10=3 台, 11=4 台
4954 ;          BIT 5, 4=INITIAL VIDEO MODE  5, 4 位=CRT 初始
                模式
4955 ;                          00-UNSED  00=未用
4956 ;                          01-40×25 BW USING COLOR CARD
                                01=彩色卡的 40×25 单色模式
4957 ;                          10-80×25 BW USING COLOR CARD
                                10=彩色卡的 80×25 单色模式
4958 ;                          11-80×25 BW USING BW CARD
```

```
                                        11 = 单色卡的 80×25 单色模式
                4959 ;          BIT 3, 2 = PLANAR RAM SISE (00 = 16K, 01 =
                                32K, 10 = 48K, 11 = 64K)  3, 2 位 = 母板 RAM 区容
                                量
                4960 ;          BIT 1 NOT USED  00 = 16K, 01 = 32K, 10 = 48K,
                                11 = 64K
                4961 ;          BIT 0 = IPL FROM DISKETTE…THIS BIT INDIC-
                                ATES THAT THERE ARE DISKETTE  1 位无用
                4962 ;                DRIVES ON THE SYSTEM  0 位 = 0 号软
                                盘机接入否, 1 表示接了
                4963 ;
                4964 ;          NO OTHER REGISTERS AFFECTED  无寄存器内
                                容被破坏
                4965 ; ……
                4966            ASSUME CS: CODE, DS: DATA
F84D            4967 EQUIPMENT  PROC FAR
F84D FB         4968            STI                          ; INTERRUPTS B-
                                                             ACK ON  允许中
                                                             断
F84E 1E         4969            PUSH  DS                     ; SAVE SEGMENT
                                                             REGISTER
F84F B84000  R  4970            MOV   AX, DATA               ; ESTABLISH AD-
                                                             DRESSING
F852 8ED8       4971            MOV   DS, AX                 DS 指向数据段
F854 A11000  R  4972            MOV   AX, EQUIP_FLAG         ; GET THE CURR-
                                                             ENT SETTINGS
                                                             取配接情况字, 并
                                                             送 AX
F857 1F         4973            POP   DS                     ; RECOVER SEG-
                                                             MENT  恢复 DS
F858 CF         4974            IRET                         ; RETURN TO CA-
                                                             LLER  返回
                4975 EQUIPMENT  ENDP
                4976 ;…INT 15 ……
                4977 ; CASSETTE I/O  软中断 15—盒带机 I/O (CASSETTE I/O)子
                     程序
                4978 ;          (AH) = 0 TURN CASSETTE MOTOR ON  (AH) = 0
                                打开录音机马达
```

```
;       (AH) = 1 TURN CASSETTE MOTOR OFF   (AH)
;       = 1 关闭录音机马达
;       (AH) = 2 READ 1 OR MORE 256 BYTE BLOCKS
FROM CASSETTE   (AH) = 2 从磁带读进数个 256
字节组成的块
;               (ES, BX) = POINTER TO DATA BUFFER
;               (ES, BX)指向读入数据的缓存区首
;               (CX) = COUNT OF BYTES TO READ
;               (CX) = 应读出的字节数
;               ON EXIT:   出口参数:
;               (ES, BX) = POINTER TO LAST BYTE
;               READ + 1   (ES, BX) = 最后读出字节在内
;               存缓存地址 + 1
;               (DX) = COUNT OF BYTES ACTUALLY
;               READ   (DX) = 实际读入的字节数
;               (CY) = 0 IF NO ERROR OCCURRED
;               (CY) = 0 无错
;                    = 1 IF ERROR OCCURRED
;                    = 1 有错
;               (AH) = ERROR RETURN IF (CY) = 1
;               (AH) = 错误号(CY = 1)
;                      = 01 IF CRC ERROR WAS DET-
;                      ECTED   = 01 CRC 错
;                      = 02 IF DATA TRANSITIONS
;                      ARE LOST   = 02 数据跳变错
;                      = 04 IF NO DATA WAS FOUND
;                      = 04 无数据错
;       (AH) = 3 WRITE 1 OR MORE 256 BYTE BLOCKS
TO CASSETTE   (AH) = 3 将数个 256 字节组成的块
写入磁带
;               (ES, BX) = POINTER TO DATA BUFFER
;               (ES, BX) = 数据缓存区首址
;               (CX) = COUNT OF BYTES TO WRITE
;               (CX) = 要求写入的字节数
;               ON EXIT:   出口参数:
;               (EX, BX) = POINTER TO LAST BYTE
;               WRITTEN + 1   (ES, BX) = 最后被写入字
;               节存贮单元地址 + 1
;               (CX) = 0   (CX) = 0
```

```
                4998 ;          (AH) = ANY OTHER THAN ABOVE VALUES
                                CAUSES (CY) = 1    (AH)不等于 0, 1, 2, 3 时
                                本程序返回参数: (CY) = 1, (AH) = 80
                4999 ;                  AND(AH) = 80 TO BE RETURNED
                                        (INVALID COMMAND).
                5000 ; ..........................................
                5001            ASSUME DS:DATA, ES:NOTHING SS:NOT-
                                HING, CS:CODE
F859            5002 CASSETTE_IO  PROC  FAR
F859 FB         5003            STI                 ; INTERRUPTS BACK
                                                      ON, 允中
F85A 1E         5004            PUSH    DS          ; ESTABLISH ADDRE-
                                                      SSING TO DATA  保
                                                      存原数据段首址
F85B 50         5005            PUSH    AX
F85C B84000  R  5006            MOV     AX, DATA
F85F 8ED8       5007            MOV     DS, AX      DS 指向数据段
F861 802671007F R 5008          AND     BIDS_BREAK, 7FH
                                                    ; MAKE SURE BREAK
                                                      FLAG IS OFF  BRE-
                                                      AK 标志位送.
F866 58         5009            POP     AX
F867 E80400     5010            CALL    W1          ; CASSETTE_IO_CO-
                                                      NT  调 W1 子程序, 完
                                                      成 (AH) 规定的盒带机
                                                      操作
F86A 1F         5011            POP     DS
F86B CA0200     5012            RET     2           ; INTERRUPT RETURN
                                                      扔掉标志位的返回
                5013 CASSETTE_IO    ENDP
F86E            5014 W1         PROC    NEAR
                5015 ; ..........................................
                5016 ; PURPOSE:  下面是 W1 子程序, 它根据 AH 内容转相应处
                     理程序
                5017 ;  TO CALL APPROPRIATE ROUTINE, DEPENDING ON
                     REG AH
                5018 ;
                5019 ;  AH              ROUTINE  AH 子程序作用
                5020 ; ..........................................
```

```
                5021 ; 0    MOTOR ON  0 开马达
                5022 ; 1    MOTOR OFF  1 关马达
                5023 ; 2    READ CASSETTE BLOCK  2 读块
                5024 ; 3    WRITE CASSETTE BLOCK  3 写块
                5025 ; ..........................................

                5026
F86E 0AE4       5027        OR      AH, AH          ; TURN ON MOTOR?  打开
                                                      马达?
F870 7413       5028        JZ      MOTOR_ON        ; YES, DO IT
F872 FECC       5029        DEC     AH              ; TURN OFF MOTOR?  否
F874 7418       5030        JZ      MOTOR_OFF       ; YES, DO IT  关闭马达?
F876 FECC       5031        DEC     AH              ; READ CASSETTE B LOCK?
                                                      否
F878 741A       5032        JZ      READ_BLOCK      ; YES, DO IT  写块?
F87A FECC       5033        DEC     AH              ; WRITE CASSETTE
                                                      BLOCK?  否
F87C 7503       5034        JNZ     W2              ; NOT_DEFINED  读块?
F87E E92701     5035        JMP     WRITE_BLOCK     ; YES, DO IT  是转 WRITE
                                                      _BLOCK

                5036
F881            5037 W2:                            ; COMMAND NOT DEFIN-
                                                      ED  命令不合法
F881 B480       5038        MOV     AH, 080H        ; ERROR, UNDEFINED OP-
                                                      ERATION 送命令错误标记
F883 F9         5039        STC                     ; ERROR FLAG  CY 送 1
F884 C3         5040        RET                     W1 返回
                5041 W1     ENDP
                5042

F885            5043 MOTOR_ON   PROC  NEAR
                5044 ; ..........................................
                5045 ; PURPOSE:  MOTOR_ON 打开录音机马达子程序
                5046 ;  TO TURN ON CASSETTE MOTOR  作用:打开马达
                5047 ; ..........................................
F885 E461       5048        IN      AL, PORT_B      ; READ CASSETTE OUTP-
                                                      UT  读 8255 PB 口
F887 24F7       5049        AND     AL, NOT 08H     ; CLEAR BIT TO TURN ON
                                                      MOTOR
F889 E661       5050 W3:    OUT     PORT_B, AL      ; WRITE IT OUT  PB 口第
```

```
                                                    4 位清 0，打开录音机马达
F88B 2AE4  5051          SUB     AH, AH         ; CLEAR AH  清 AH，命令有
                                                    效标记
F88D C3    5052          RET                    ; 返回
           5053 MOTOR_ON      ENDP
           5054
F88E       5055 MOTOR_OFF     PROC  NEAR
           5056 ; ..........................................
           5057 ; PURPOSE:  MOTOR_OFF 关闭录音机马达子程序
           5058 ;  TO TURN CASSETTE MOTOR OFF  作用:关闭马达
           5059 ; ..........................................
F88E E461  5060          IN      AL, PORT_B     ; READ CASSETTE OUTPUT
                                                    读 8255 PB 口
F890 0C08  5061          OR      AL, 08H        ; SET BIT TO TURN OFF
F892 EBF5  5062          JMP     W3             ; WRITE IT, CLEAR ERROR,
                                                    RETURN  PB 口第 4 位送 1，
                                                    关闭马达
           5063 MOTOR_OFF     ENDP
F894       5064 READ_BLOCK    PROC  NEAR
           5065 ;
           5066 ; PURPOSE:  READ_BLOCK 读块子程序
           5067 ;  TO READ 1 OR MORE 256 BYTE BLOCKS FROM CASSETTE
                   作用:读数个数据块
           5068 ;
           5069 ; ON ENTRY:  入口参数:
           5070 ;  ES IS SEGMENT FOR MEMORY BUFFER (FOR COMPACT
                   CODE)  (ES) = 内存数据块的段地址
           5071 ;  BX POINTS TO START OF MEMORY BUFFER  (BX) = 数据块
                   缓存区首址
           5072 ;  CX CONTAINS NUMBER OF BYTES TO READ  (CX) = 读入
                   字节数
           5073 ; ON EXIT:  出口参数:
           5074 ;  BX POINTS 1 BYTE PAST LAST BYTE PUT IN MFM
                   (BX) = 最后读出字节在内存缓存地址 +1
           5075 ;  CX CONTAINS DECREMENTED BYTE COUNT  (CX) = 应读入
                   字节数 - 实际读入数
           5076 ;  DX CONTAINS NUMBER OF BYTES ACTUALLY READ
                   (DX) = 实际读入的字节数
           5077 ;
```

```
                        5078 ;  CARRY FLAG IS CLEAR IF NO ERROR DETECTED
                                (CY) = 0 读入过程中无错误发生
                        5079 ;  CARRY FLAG IS SET IF CRC ERROR DETECTED
                                (CY) = 1 读入过程中发生 CRC 错误
                        5080 ;  ..........................................
F894 53                 5081        PUSA    BX              ; SAVE BX
F895 51                 5082        PUSH    CX              ; SAVE CX
F896 56                 5083        PUSH    SI              ; SAVE SI      暂存 BX,
                                                                CX, SI
F897 BE0700             5084        MOV     SI, 7           ; SET UP RETRY COUNT
                                                                FOR LEADER  最多找
                                                                引导头 7 次
F89A E8C201             5085        CALL    BEGIN_OP        ; BEGIN BY STARTING
                                                                MOTOR  启动马达,延迟
                                                                约 0.66 秒
F89D                    5086 W4:                            ; SEARCH FOR LEADER
F89D E462               5087        IN      AL, PORT_C      ; GET INTIAL VALUE
                                                                读 PC 口,其第 4 位反映录
                                                                音机输出电平(的变化)
F89F 2410               5088        AND     AL, 010H        ; MASK OFF EXTRANE-
                                                                OUS BITS
F8A1 A26B00       R     5089        MOV     LAST_VAL, AL    ; SAVE IN LOC LAST_
                                                                VAL  该电平值送 LAST
                                                                _VAL 单元, READ_
                                                                HALF_BIT 中将用到它
F8A4 BA7A3F             5090        MOV     DX, 16250       ; # OF TRANSITIONS TO
                                                                LOOK FOR
                        5091
F8A7                    5092 W5:                            ; WAIT_FOR_EDGE
F8A7 F606710080   R     5093        TEST    BIOS_BREAK, 80H ; CHECK FOR BRE-
                                                                AK KEY  用户当前
                                                                按下 BREAK 键?
F8AC 7403               5094        JZ      W6              ; JUMP IF NO BREAK
                                                                KEY
F8AE E98A00             5095        JMP     W17             ; JUMP IF BREAK KEY
                                                                HIT  按下 BREAK 键,转
                                                                W17 作超时错处理
                        5096
F8B1 4A                 5097 W6:    DEC     DX
```

```
F8B2 7503          5098          JNZ     W7                ; JUMP IF BEGINN-
                                                             ING OF LEADER
F8B4 E98400        5099          JMP     W17               ; JUMP IF NO LEA-
                                                             DER FOUND
                   5100
F8B7 E8C600        5101 W7:      CALL    READ_HALF_BIT     ; IGNORE FIRST
                                                             EDGE 转 READ_
                                                             HALF_BIT，检测引
                                                             导头前第一次电平变
                                                             化
F8BA E3EB          5102          JCXZ    W5                ; JUMP IF NO EDGE
                                                             DETECTED 无变
                                                             化转 W5，准备再找
F8BC BA7803        5103          MOV     DX, 0378H         ; CHECK FOR HALF
                                                             BITS 有电平变化，
                                                             现找引导 头。0378H
                                                             是"1"、"0"半个方波
                                                             周期计时值的平均值
F8BF B90002        5104          MOV     CX, 200H          ; MUST HAVE AT
                                                             LEAST THIS MANY
                                                             ONE SIZE 引导头
                                                             至少具有"1"的个数
                                                             之半，(256 个"1")
                   5105                                    ; PULSES BEFORE
                                                             CHECKNG FOR SY-
                                                             NC BIT(0)
F8C2 E421          5106          IN      AL, 021H          ; INTERRUPT MASK
                                                             REGISTER
F8C4 0C01          5107          OR      AL, 1             ; DISABLE TIMER
                                                             INTERRUPTS 写
                                                             8259屏蔽寄存器禁止
                                                             0 号中断即时钟中断
F8C6 E621          5108          OUT     021H, AL
F8C8               5109 W8:                                ; SEARCH_LDR
F8C8 F606710080  R 5110          TEST    BIOS_BREAK, 80H   ; CHECK FOR BRE-
                                                             AK KEY 当前按下
                                                             BREAK键?
F8CD 756C          5111          JNZ     W17               ; JUMP IF BREAK
                                                             KEY HIT 按下转
```

```
                                                    W17出错。
F8CF 51          5112      PUSH   CX                ; SAVE REG CX  否
F8D0 E8AD00      5113      CALL   READ_HALF_BIT     ; GET PULSE WIDTH
                                                    读半个方波
F8D3 0BC9        5114      OR     CX, CX            ; CHECK FOR TRANSI-
                                                    TION (CX) = 0 表示无电
                                                    平跳变,未读出波形
F8D5 59          5115      POP    CX                ; RESTORE ONE BIT
                                                    COUNTER  恢复计数器
F8D6 74C5        5116      JZ     W4                ; JUMP IF NO TRANSI-
                                                    TION  无电平跳变，转
                                                    W4 再找引导头
F8D8 3BD3        5117      CMP    DX, BX            ; CHECK PULSE WIDTH
                                                    判本次读出的是"1"的半
                                                    个波形还是"0"的半个波
                                                    形
F8DA E304        5118      JCXZ   W9                ; IF CX = 0 THEN WE
                                                    CAN LOOK  (CX) = 0
                                                    表示引导头大概读完了，
                                                    准备找同步位
                 5119                               ; FOR SYNC BIT(0)
F8DC 73BF        5120      JNC    W4                ; JUMP IF ZERO BIT (N-
                                                    OT GOOD LEADER)
                                                    引导头末读完，若读出的
                                                    是"0"的半个波形，(CY)
                                                    = 0
F8DE E2E8        5121      LOOP   W8                ; DEC CX AND READ
                                                    ANOTHER HALF ONE
                                                    BIT  继续读引导头
F8E0             5122 W9:                           ; FIND_SYNC
F8E0 72E6        5123      JC     W8                ; JUMP IF ONE BIT (ST-
                                                    ILL LEADER)  从LOOP
                                                    W8 下来 (CY) = 1，总转
                                                    W8 再读引导头
                 5124
                 5125 ; A SYNCH BIT HAS BEEN FOUND. READ SYN CHARAC-
                        TER.
                 5126
```

```
F8E2 E89B00  5127      CALL   READ_HALF_BIT  ; SKIP OTHER HALF OF
                                              SYNC BIT(0) 读出同步位
                                              另半个波形
F8E5 E86A00  5128      CALL   READ_BYTE      ; READ SYN BYTE  读同
                                              步字节
F8E8 3C16    5129      CMP    AL, 16H        ; SYNCHRONIZATION CH-
                                              ARACTER  同步字节内容
                                              = 16H?
F8EA 7549    5130      JNE    W16            ; JUMP IF BAD LEADER
                                              FOUND.  引导头不对，转
                                              W19 找引导头，最多重复 7
                                              次
             5131
             5132 ;……GOOD CRC SO READ DATA BLOCK (S)  同步字节无错，
                                                      现开始读块
F8EC 5E      5133      POP    SI             ; RESTORE REGS  恢复SI,
                                              CX, BX
F8ED 59      5134      POP    CX
F8EE 5B      5135      POP    BX
             5136 ; ……
             5137 ; READ 1 OR MORE 256 BYTE BLOCKS FROM CASSETTE
                    下面程序段读入数个数据块
             5138 ;
             5139 ; ON ENTRY:
             5140 ; ES IS SEGMENT FOR MEMORY BUFFER (FOR COMPACT
                    CODE)
             5141 ; BX POINTS TO START OF MEMORY BUFFER
             5142 ; CX CONTAINS NUMBER OF BYTES TO READ
             5143 ; ON EXIT:
             5144 ; BX POINTS 1 BYTE PAST LAST BYTE PUT IN MEM
             5145 ; CX CONTAINS DECREMENTED BYTE COUNT
             5146 ; DX CONTAINS NUMBER OF BYTES ACTUALLY READ
             5147 ; ……
F8EF 51      5148      PUSH   CX             ; SAVE BYTE COUNT  暂
                                              存应读出的字节数
F8F0         5149 W10:                       ; COME HERE BEFORE
                                              EACH
             5150                             ; 256 BYTE BLOCK IS
                                              READ
```

```
F8F0 C7066900FFFF R 5151        MOV     CRC_REG, 0FFFFH ; INIT CRC REG
                                                        初始 CRC 寄存器
F8F6 BA0001         5152        MOV     DX, 256         ; SET DX TO DA-
                                                        TA BLOCK SIZE
                                                        设数据块大小(256
                                                        字节)
F8F9                5153 W11:                           ; RD_BLK
F8F9 F606710080     R 5154      TEST    BIOS_BREAK, 80H ; CHECK FOR BR-
                                                        EAK KEY
F8FE 7523           5155        JNZ     W13             ; JUMP IF BREAK
                                                        KEY HIT  这时按
                                                        入BREAK 键，本
                                                        次读块操作流产
F900 E84F00         5156        CALL    READ_BYTE       ; READ BYTE FR-
                                                        OM CASSETTE
                                                        读一字节
F903 721E           5157        JC      W13             ; CY SET INDICA-
                                                        TES NO DATA
                                                        TRANSITIONS
                                                        (CY) = 1 表示数据
                                                        跳变错
F905 E305           5158        JCXZ    W12             ; IF WE'VE ALR-
                                                        EADY REACHED
                                                        读完要求读出的全
                                                        部字节则转 W12，
                                                        为读出 CRC 字节
                                                        空读
                    5159                                ; END OF MEMO-
                                                        RY BUFFER
                    5160                                ; SKIP REST OF
                                                        BLOCK
F907 268807         5161        MOV     ES:[BX], AL     ; STORE DATA B-
                                                        YTE AT BYTE
                                                        PTR  读出字节送
                                                        内存缓存
F90A 43             5162        INC     BX              ; INC BUFFER PTR
                                                        缓存地址加 1
F90B 49             5163        DEC     CX              ; DEC BYTE COU-
                                                        NTER  应读出字
```

```
                                                        节数减 1
F90C                    5164 W12:                       ; LOOP UNTIL DA-
                                                          TA BLOCK HAS
                                                          BEEN READ FR-
                                                          OM CASSETTE.
F90C 4A                 5165        DEC     DX          ; DEC BLOCK CNT
                                                          块中尚存字节数减
                                                          1
F90D 7FEA               5166        JG      W11         ; RD_BLK
F90F E84000             5167        CALL    READ_BYTE   ; NOW READ TWO
                                                          CRC BYTES 读
                                                          完一块，现读出两
                                                          个 CRC 校验字节
F912 E83D00             5168        CALL    READ_BYTE
F915 2AE4               5169        SUB     AH, AH      ; CLEAR AH
F917 813E69000F1D  R    5170        CMP     CRC_REG, 1D0FH ; IS THE CRC CO-
                                                          RRECT 整个数据
                                                          块有 CRC 错?
F91D 7506               5171        JNE     W14         ; IF NOT EQUAL
                                                          CRC IS BAD
F91F E306               5172        JCXZ    W15         ; IF BYTE COUNT
                                                          IS ZERO 无CRC
                                                          错，还需读块?
                        5173                            ; THEN WE HAVE
                                                          READ ENOUGH
                        5174                            ; SO WE WILL E-
                                                          XIT
F921 EBCD               5175        JMP     W10         ; STILL MORE, SO
                                                          READ ANOTHER
                                                          BLOCK (CX) = 0,
                                                          不需再读，转 W15
F923                    5176 W13:                       ; MISSING_DATA
                        5177                            ; NO DATA TRA-
                                                          NSITIONS SO
F923 B401               5178        MOV     AH, 01H     ; SET AH = 02 TO
                                                          INDICATE 置
                                                          (AH) = 02，表示数
                                                          据跳变错
                        5179                            ; DATA TIMEOUT
```

```
F925                5180 W14:                          ; BAD_CRC
F925 FEC4           5181        INC     AH             ; EXIT EARLY ON ERROR
                                                       从 W14 转入置 (AH) = 01，表示
                                                       CRC 错
                    5182                               ; SET AH=01 TO INDICATE
                                                       CRC ERROR
F927                5183 W15:                          ; RD_BLK_EX
F927 5A             5184        POP     DX             ; CALCULATE COUNT OF
F928 2BD1           5185        SUB     DX, CX         ; DATA BYTES ACTUALLY R-
                                                       EAD  (DX) = 应读入字节数 -
                                                       尚未读出字节数 = 实读入字节数
                    5186                               ; RETURN COUNT IN REG DX
F92A 50             5187        PUSH    AX             ; SAVE AX  (RET CODE)
F92B F6C403         5188        TEST    AH, 03H        ; CHECK FOR ERRORS  有错则
                                                       (AH) = 01, 02, 与 03 与后非 0
F92E 7513           5189        JNZ     W18            ; JUMP IF ERROR DETECTED
                                                       有错转W18
F930 E81F00         5190        CALL    READ_BYTE      ; READ TRAILER  读带尾
F933EB0E            5191        JMP     SHORT W18      ; SKIP TO TURN OFF MOTOR
                                                       转 W18 关录音机马达
F935                5192 W16:                          ; BAD_LEADER
F935 4E             5193        DEC     SI             ; CHECK RETRIES  重新找引导
                                                       头满 7 次?
F936 7403           5194        JZ      W17            ; JUMP IF TOO MANY RETRI-
                                                       ES  已满 7 次，转 W17。出错
F938 E962FF         5195        JMP     W4             ; JUMP IF NOT TOO MANY R-
                                                       ETRIES  未满，再寻找引导头
F93B                5196 W17:                          ; NO VALID DATA FOUND
                    5197 ;……NO DATA FROM CASSETTE ERROR, I. E. TIMEOUT
                    5198
F93B 5E             5199        POP     SI             ; RESTORE REGS
F93C 59             5200        POP     CX             ; RESTORE REGS
F93D 5B             5201        POP     BX
F93E 2BD2           5202        SUB     DX, DX         ; ZERO NUMBER OF BYTES R-
                                                       EAD  清 DX，读出字节数为 0
F940 B404           5203        MOV     AH, 04H        ; TIME OUT ERROR (NO LEA-
                                                       DER)  04 为超时错标记
F942 50             5204        PUSH    AX
```

```
F943                5205 W18:                          ; MOT_OFF
F943 E421           5206      IN      AL, 021H         ; RE_ENABLE INTERRUPTS
                                                         清 8259A 屏蔽寄存器第 0 位，允
                                                         使时钟中断
F945 24FE           5207      AND     AL, 0FFH_1
F947 E621           5208      OUT     021H, AL
F949 E842FF         5209      CALL    MOTOR_OFF        ; TURN OFF MOTOR  关闭录音
                                                         机马达
F94C 58             5210      POP     AX               ; RESTORE RETURN CODE  恢
                                                         复操作成功与否标记
F94D 80FC01         5211      CMP     AH, 01H          ; SET CARRY IF ERROR (AH>
                                                         0)
F950 F5             5212      CMC                      若(AH) > = 01, (CY) 送 1
F951 C3             5213      RET                      ; FINISHED  返回
                    5214 READ_BLOCK  ENDP
                    5215 ; ..........
F952                5216 READ_BYTE  PROC  NEAR  READ_BYTE 子程序
                    5217 ; PURPOSE:  本子程序从磁带读进一个字节的数据，该字节由
                                     AL 返回
                    5218 ; TO READ A BYTE FROM CASSETTE
                    5219 ;
                    5220 ; ON EXIT REG AL CONTAINS READ DATA BYTE
                    5221 ; ..........
F952 53             5222      PUSH    BX               ; SAUE REGS BX CX  保存BX,
                                                         CX
F953 51             5223      PUSH    CX
F954 B108           5224      MOV     CL, 8H           ; SET BIT COUNTER FOR 8 B-
                                                         ITS  字节位计数器
F956                5225 W19:                          ; BYTE_ASM
F956 51             5226      PUSH    CX               ; SAVE CX
                    5227 ; ..........
                    5228 ; READ DATA BIT FROM CASSETTE. 下面程序段从磁带读
                           入一个数据位
                    5229 ; ..........
F957 E82600         5230      CALL    READ_HALF_BIT    ; READ ONE PULSE  读
                                                         数据位的半个周期
F95A E320           5231      JCXZ    W21              ; IF CX = 0 THEN TIMEOUT
                                                         (CX) = 0超时错
                    5232                               ; BECAUSE OF NO DATA TR-
```

```
                                                ANSITIONS
F95C 53          5233    PUSH    BX             ; SAVE 1ST HALF BIT'S 保存该
                                                数据位半个周期的计时值
                 5234                           ; PULSE WIDTH (IN BX)
F95D E82000      5235    CALL    READ_HALF_BIT  ; READ COMPLEMENT-
                                                ARY PULSE 读该数据
                                                位下半个周期
F960 58          5236    FOP     AX             ; COMPUTE DATA BIT 上半周期
                                                计时值送 AX
F961 E319        5237    JCXZ    W21            ; IF CX = 0 THEN TIMEOUT DUE
                                                TO (CX) = 0 下半周期发生超时
                                                错
                 5238                           ; NO DATA TRANSITIONS
F963 03D8        5239    ADD     BX, AX         ; PERIOD 上半周期计时值 + 下半
                                                周期计时值 = 整个周期的计时值
F965 81FBF006    5240    CMP     BX, 06F0H      ; CHECK FOR ZERO BIT 比较
F969 F5          5241    CMC                    ; CARRY IS SET IF ONE BIT
                                                "1"位则 CY 送 1, "0" 位则 CY 送 0
F96A 9F          5242    LAHF                   ; SAVE CARRY IN AH 标记位送
                                                AH
F96B 59          5243    POP     CX             ; RESTORE CX 恢复 CX, 其中
                                                (CL)是字节位计数器,(CH)是已拼
                                                字节位寄存器
                 5244                           ; NOTE: 注意:
                 5245                           ; MS BIT OF BYTE IS READ FI-
                                                RST. 写磁带时字节的高位先写,
                                                读带时高位
                 5246                           ; REG CH IS SHIFTED LEFT W-
                                                ITH 先读出, 利用带 CY 位的左
                                                环移指令装配
                 5247                           ; CARRY BEING INSERTED INTO
                                                LS 读出的 8 位数据, 使其成一字
                                                节
                 5248                           ; BIT OF CH
                 5249                           ; AFTER ALL 8 BITS HAVE BE-
                                                EN
                 5250                           ; READ, THE MS BIT OF THE
                                                DATA BYTE
                 5251                           ; WILL BE IN THE MS BIT OF
```

```
                                                    REG CH
F96C D0D5       5252        RCL     CH, 1       ; ROTATE REG CH LEFT WITH
                                                    CARRY TO  读入数据位移入 CH
                5253                            ; LS BIT OF REG CH
F96E 9E         5254        SAHF                ; RESTORE CARRY FOR CRC
                                                    ROUTINE  恢复各标记位
F96F E8D900     5255        CALL    CRC_GEN     ; GENERATE CRC FOR BIT   调
                                                    CRC_GNE 产生 CRC 字节
F972 FEC9       5256        DEC     CL          ; LOOP TILL ALL 8 BITS OF
                                                    DATA  读入位数计数器-1
                5257                            ; ASSEMBLED IN REG CH
F974 75E0       5258        JNZ     W19         ; BYTE_ASM
F976 8AC5       5259        MOV     AL, CH      ; RETURN DATA BYTE IN REG
                                                    AL  读入 8 位,拼成的字节送 AL
F978 F8         5260        CLC                     (CY)送 0,无错标记
F979            5261 W20:                       ; RD_BYT_EX
F979 59         5262        POP     CX          ; RESTORE REGS CX, BX   恢复
                                                    CX, BX
F97A 5B         5263        POP     BX
F97B C3         5264        RET                 ; FINISHED  返回
F97C            5265 W21:                       ; NO_DATA  超时错处理
F97C 59         5266        POP     CX          ; RESTORE CX
F97D F9         5267        STC                 ; INDICATE ERROR  CY 置 1, 有
                                                    错标记
F97E EBF9       5268        JMP     W20         ; RO_BYT_EX
                5269 READ_BYTE ENDP
                5270 ; ..................................................
F980            5271 READ_HALF_BIT  PROC  NEAR   READ_HALF_BIT 子
                     程序
                5272 ; PURPOSE:
                5273 ; TO COMPUTE TIME TILL NEXT DATA   本程序等待录音
                     机输出电平的变化,从而得到半个周期的计时值
                5274 ; TRANSITION (EDGE)
                5275 ;
                5276 ; ON ENTRY;  入口参数:
                5277 ; EDGE_CNT CONTAINS LAST EDGE COUNT     EDGE_
                     CNT 单元记录上次电平变化时的计时值
                5278 ;
```

```
                        5279 ; ON EXIT: 出口参数:
                        5280 ; AX CONTAINS OLD LAST EDGE COUNT
                             (AX) = 上半个周期的计时值
                        5281 ; BX CONTAINS PULSE WIDTH (HALF BIT)
                             (BX) = 半个周期脉冲宽度(计时值)
                        5282 ; ..........................................
F980 B96400             5283     MOV   CX, 100          ; SET TIME TO WAIT F-
                                                          OR BIT  超时计时器送
                                                          初值
F983 8A266B00 R         5284     MOV   AH, LAST_VAL     ; GET PRESENT INPUT
                                                          VALUE  取前一刻电平
                                                          值
F987                    5285 W22:                      ; RD_H_BIT
F987 E462               5286     IN    AL, PORT_C       ; INPUT DATA BIT  读
                                                          当前录音机电平 (在第4
                                                          位)
F989 2410               5287     AND   AL, 010H         ; MASK OFF EXTRANE-
                                                          OUS BITS
F98B 3AC4               5288     CMP   AL, AH           ; SAME AS BEFORE? 比
                                                          较前刻电平和当前电平,
                                                          看是否有跳变
F98D E1F8               5289     LOOPE W22              ; LOOP TILL IT CHAN-
                                                          GES
F98F A26B00     R       5290     MOV   LAST_VAL, AL     ; UPDATE LAST_VAL
                                                          WITH NEW VALUE
                                                          有跳变或超时转此, 记录
                                                          前一刻电平
F992 B000               5291     MOV   AL, 0            ; READ TIMER'S COUN-
                                                          TER COMMAND  送封
                                                          锁计时器当前内容命令。
F994 E643               5292     OUT   TIM_CTL, AL      ; LATCH COUNTER
F996 E440               5293     IN    AL, TIMER0       ; GET LS BYTE
F998 8AE0               5294     MOV   AH, AL           ; SAVE IN AH
F99A E440               5295     IN    AL, TIMER0       ; GET MS BYTE  读计时
                                                          器0当前内容
F99C 86C4               5296     XCHG  AL, AH           ; XCHG AL, AH
F99E 8B1E6700   R       5297     MOV   BX, EDGE_CNT     ; BX GETS LAST EDGE
                                                          COUNT
F9A2 2BD8               5298     SUB   BX, AX           ; SET BX EQUAL TO H-
```

```
                                                    ALF BIT PERIOD
                                                    上次电平变化时的计
                                                    时值-当前电平变化
                                                    时的计时值
F9A4 A36700 R 5299          MOV   EDGE_CNT, AX   ; UPDATE EDGE
                                                    COUNT; (BX)为半
                                                    个周期的计时值（计
                                                    时器是递减计时的）
F9A7 C3       5300          RET                  ; 返回
              5301 READ_HALF_BIT ENDP
              5302 ; ..........................................
F9A8          5303 WRITE_BLOCK    PROC  NEAR   WRITE_BLOCK 子程序
              5304 ;
              5305 ; WRITE 1 OR MORE 256 BYTE BLOCKS TO CASSETTE
                     本子程序将一个或数个数据块写入磁带,若待写入的字节总
              5306 ;       THE DATA IS PADDED TO FILL OUT THE L-
                           AST 256 BYTE BLOCK
                           数不是 256 的整数倍,则本程序在写完最后一个数据
              5307 ;       块的最后字节后,重复写最后字节,以凑足 256 字节
              5308 ; ON ENTRY:  入口参数:
              5309 ; BX POINTS TO MEMORY BUFFER ADDRESS  (BX) = 内存
                     缓存首址
              5310 ; CX CONTAINS NUMBER OF BYTES TO WRITE   (CX) =
                     待写字节总数
              5311 ;
              5312 ; ON EXIT:  出口参数:
              5313 ; BX POINTS 1 BYTE PAST LAST BYTE WRITTEN TO CA-
                     SSETTE  (BX) = 最后写入磁带字节的地址 + 1
              5314 ; CX IS ZERO                          (CX) = 0
              5315 ; ..........................................
F9A8 53       5316          PUSH  BX
F9A9 51       5317          PUSH  CX
F9AA E461     5318          IN    AL, PORT_B     ; DISABLE SPEAK-
                                                    ER  PB 口第 1 位送
                                                    0, 禁止扬声器输出
F9AC 24FD     5319          AND   AL, NOT 02H
F9AE 0C01     5320          OR    AL, 01H        ; ENABLE TIMER
                                                    PB 口第 0 位送 1, 允
                                                    使计时器 2
```

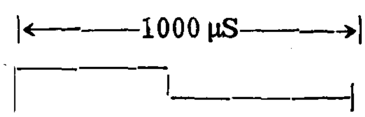
```
F9B0 E661      5321        OUT     PORT_B, AL
F9B2 B0B6      5322        MOV     AL, 0B6H      ; SET UP TIMER···MODE 3 SQU-
                                                   ARE WAVE  写8253控制口，置
                                                   计时器2模式3，方波发生器方式
F9B4 E643      5323        OUT     TIM_CTL, AL
F9B6 E8A600    5324        CALL    BEGIN_OP      ; START MOTOR AND DELAY
                                                   启动录音机马达并延迟一段时间
F9B9 B8A004    5325        MOV     AX, 1184      ; SET NORMAL BIT SIZE  "1"需
                                                   计时值
F9BC E88500    5326        CALL    W31           ; SET_TIMER  送计时值至8253
                                                   计时器2，准备写引导头(256字节
                                                   全1)
F9BF B90008    5327        MOV     CX, 0800H     ; SET CX FOR LEADER BYTE C-
                                                   OUNT  引导头计数器初始化，800
                                                   H = 2048 = 256 × 8
F9C2           5328 W23:                         ; WRITE LEADER
F9C2 F9        5329        STC                   ; WRITE ONE BITS  CY送1，写
                                                   "1"标记
F9C3 E86800    5330        CALL    WRITE_BIT     ; 写"1"
F9C6 E2FA      5331        LOOP    W23           ; LOOP TIL LEADER IS WRITTEN
                                                   写256字节个"1"
F9C8 F8        5332        CLC                   ; WRITE SYNC BIT(0)  引导头写
                                                   完，现写一个"0"即同步位
F9C9 E86200    5333        CALL    WRITE_BIT     写"0"
F9CC 59        5334        POP     CX            ; RESTORE REGS CX, BX
F9CD 5B        5335        POP     BX
F9CE B016      5336        MOV     AL, 16H       ; WRITE SYN CHARACTER  写同
                                                   步字节16H，写完同步字节后写数
                                                   据块
F9D0 E84400    5337        CALL    WRITE_BYTE ;
               5338 ; ·····
               5339 ; WRITE 1 OR MORE 256 BYTE BLOCKS TO CASSETTE
                      下面将数据块写入磁带
               5340 ; ·····
               5341 ; ON ENTRY:
               5342 ; BX POINTS TO MEMORY BUFFER ADDRESS
               5343 ; CX CONTAINS NUMBER OF BYTES TO WRITE
               5344 ;
```

```
                        5345 ; ON EXIT:
                        5346 ; BX POINTS 1 BYTE PAST LAST BYTE WRITTEN TO
                               CASSETTE
                        5347 ; CX IS ZERO
                        5348 ; ..........................................
F9D3                    5349 WR_BLOCK:
F9D3 C7066900FFFF R     5350         MOV   CRC_REG, 0FFFFH ; INIT CRC
                                                             CRC_REG
                                                             单元初始化
F9D9 BA0001             5351         MOV   DX, 256         ; FOR 256 BYTES
                                                             字节计数器初始
                                                             化
F9DC                    5352 W24:                          ; WR_BLK
F9DC 268A07             5353         MOV   AL, ES:[BX]     ; READ BYTE
                                                             FROM MEM
                                                             取一字节
F9DF E83500             5354         CALL  WRITE_BYTE      ; WRITE IT TO
                                                             CASSETTE 调
                                                             WRITE_BYTE
                                                             写该字节
F9E2 E302               5355         JCXZ  W25             ; UNLESS CX =
                                                             0, ADVANCE
                                                             PTRS & DEC
                                                             COUNT (CX)
                                                             =0 表示全部字
                                                             节均写完，转
                                                             W25
F9E4 43                 5356         INC   BX              ; INC BUFFER
                                                             POINTER 地
                                                             址加1
F9E5 49                 5357         DEC   CX              ; DEC BYTE CO-
                                                             UNTER 总字
                                                             节计数器减1
F9E6                    5358 W25:                          ; SKIP_ADV
F9E6 4A                 5359         DEC   DX              ; DEC BLOCK C-
                                                             NT 字节计数
                                                             器减1，从5355
                                                             行转此则进行块
                                                             填补工作
```

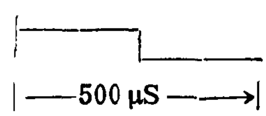
```
F9E7 7FF3       5360        JG      W24             ; LOOP TILL 256 BYTE BLO-
                                                      CK 写完一块的话继续至 5368
                                                      行
                5361                                ; IS WRITTEN TO TAPE
                5362 ; ..... WRITE CRC .....
                5363 ; WRITE 1'S COMPLEMENT OF CRC REG TO CASSETTE
                       下面是写 CRC 字的反码,将来读块时用
                5364 ; WHICH IS CHECKED FOR CORRECTNESS WHEN THE
                       BLOCK IS READ  它进行校验
                5365 ;
                5366 ; REG AX IS MODIFIED
                5367 ; ..........
F9E9 A16900 R   5368        MOV     AX, CRC_REG     ; WRITE THE ONE'S COMP-
                                                      LEMENT OF THE  取 CRC
                                                      字
                5369                                ; TWO BYTE CRC TO TAPE
F9EC F7D0       5370        NOT     AX              ; FOR 1'S COMPLEMENT  取
                                                      反
F9EE 50         5371        PUSH    AX              ; SAVE IT
F9EF 86E0       5372        XCHG    AH, AL          ; WRITE MS BYTE FIRST
F9F1 E82300     5373        CALL    WRITE_BYTE      ; WRITE IT  写 CRC 高字节
F9F4 58         5374        POP     AX              ; GET IT BACK
F9F5 F81F00     5375        CALL    WRITF_BYTE      ; NOW WRITE LS BYTE  写
                                                      CRC 低字节
F9F8 0BC9       5376        OR      CX, CX          ; IS BYTE COUNT EXHAUS-
                                                      TED?
F9FA 75D7       5377        JNZ     WR_BLOCK        ; JUMP IF NOT DUNE YET
                                                      是否写完全部字节?
F9FC 51         5378        PUSH    CX              ; SAVE REG CX  是
F9FD B92000     5379        MOV     CX, 32          ; WRITE OUT TRAILER BITS
                                                      下面写 32 个"1"作为结尾标记
FA00            5380 W26:                           ; TRAIL_LOOP
FA00 F9         5381        STC                     写"1"标记
FA01 E82A00     5382        CALL    WRITE_BIT       写"1"
FA04 E2FA       5383        LOOP    W26             ; WRITE UNTIL TRAILER
                                                      WRITTEN
FA06 59         5384        POP     CX              ; RESTORE REG CX  写完结
                                                      尾标记
FA07 B0B0       5385        MOV     AL, 0B0H        ; TURN TIMER2 OFF  写8253
```

```
                                                    控制口,置计时器2模式0
FA09 E643       5386            OUT     TIM_CTL, AL
FA0B B80100     5387            MOV     AX, 1
FA0E E83300     5388            CALL    W31             ; SET_TIMER  写计时
                                                        值,延迟片刻
FA11 E87AFE     5389            CALL    MOTOR_OFF       ; TURN MOTOR OFF  关
                                                        闭马达
FA14 2BC0       5390            SUB     AX, AX          ; NO ERRORS REPORTED
                                                        ON WRITE OP  清 AX
                                                        无错标记
FA16 C3         5391            RET                     ; FINISHED  返回
                5392 WRITE_BLOCK    ENDP
                5393 ; ..........................................
FA17            5394 WRITE_BYTE     PROC  NEAR  WRITE_BYTE  写字节子
                                                        程序
                5395 ; WRITE A BYTE TO CASSETTE. (AL)为待写入的字节
                5396 ; BYTE TO WRITE IS IN REG AL.
                5397 ; ..........................................
FA17 51         5398            PUSH    CX              ; SAVE REGS CX, AX
FA18 50         5399            PUSH    AX
FA19 8AE8       5400            MOV     CH, AL          ; AL=BYTE TO WRITE
                                                        待写字节送 CH
                5401                                    ; (MS BIT WRITTEN FI-
                                                        RST)
FA1B B108       5402            MOV     CL, 8           ; FOR 8 DATA BITS IN
                                                        BYTE  字节位计数器初
                                                        始化
                5403                                    ; NOTE:TWO EDGES
                                                        PER BIT
FA1D            5404 W27:                               ; DISASSEMBLE THE
                                                        DATA BIT
FA1D D0D5       5405            RCL     CH, 1           ; ROTATE MS BIT INTO
                                                        CARRY  移出字节一位
                                                        并送 CY 位
FA1F 9C         5406            PUSHF                   ; SAVE FLAGS  保存标
                                                        志位
                5407                                    ; NOTE, DATA BIT IS IN
                                                        CARRY  写位
FA20 E80B00     5408            CALL    WRITE_BIT       ; WRITE DATA BIT
```

```
FA23 9D                       POPF                 ; RESTORE CARRY FOR C-
                                                     RC CALC  取回刚写入的位
FA24 E82400                   CALL     CRC_GEN     ; COMPUTE CRC ON DATA
                                                     BIT  产生 CRC
FA27 FEC9                     DEC      CL          ; LOOP TILL ALL 8 BITS
                                                     DONE  位计数器减 1
FA29 75F2                     JNZ      W27         ; JUMP IF NOT DONE YET
FA2B 58                       POP      AX          ; RESTORE REGS AX, CX
                                                     写完字节
FA2C 59                       POP      CX
FA2D C3                       RET                  ; WE ARE FINISHED  恢复
                                                     AX, CX 后返回
              WRITE_BYTE      ENDP
              ; ..............................................
FA2E          WRITE_BIT       PROC   NEAR   WRITE_BIT 子程序
              ; PURPOSE:  本程序将一个数据位写入磁带
              ;
              ; TO WRITE A DATA BIT TO CASSETTE  作入口参数的 CY
                位为 1, 往磁带写"1", CY 位为 0, 往磁带写"0"
              ; CARRY FLAG CONTAINS DATA BIT  "1"的波形:
```

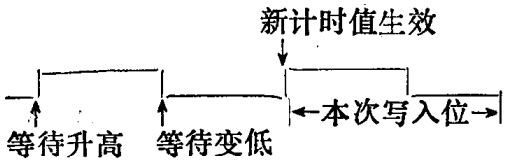



```
              ; I. E. IF SET DATA BIT IS A ONE  "0"的波形:
```




```
              ;       IF CLEAR DATA BIT IS A ZERO
              ;
              ; NOTE: TWO EDGES ARE WRITTEN PER BIT
              ;         ONE BIT HAS 500 USEC BETWEEN EDGES
              ;              FOR A 1000 USEC PERIOD (1 MILLISEC)
              ;
              ;         ZERO BIT HAS 250 USEC BETWEEN EDGES
              ;              FOR A 500 USEC PERIOD(.5 MILLISEC)
              ; CARRY FLAG IS DATA BIT
              ; ..............................................
              ; ASSUME IT'SA '1'
FA2E B8A004                   MOV      AX, 1184    ; SET AX TO NOMINAL
```

```
                                                    ONE SIZE  "1"的计时值
FA31 7203       5436        JC      W28         ; JUMP IF ONE BIT  (CY)
                                                    =1 则写 "1"
FA33 B85002     5437        MOV     AX, 592     ; NO,  SET TO NOMINAL
                                                    ZERO SIZE  写"0", 送"0"
                                                    的计时值
FA36            5438 W28:                       ; WRITE_BIT_AX
FA36 50         5439        PUSH    AX          ; WRITE BIT WITH PERI-
                                                    OD EQ TO VALUE AX
                                                    计时值压栈
FA37            5440 W29:
FA37 E462       5441        IN      AL, PORT_C  ; INPUT TIMER_0 OUTPUT
                                                    读计时器 2 输出电平
FA39 2420       5442        AND     AL, 020H
FA3B 74FA       5443        JZ      W29         ; LOOP TILL HIGN  等该电
                                                    平变成 1 态
FA3D            5444 W30:
FA3D E462       5445        IN      AL, PORT_C  ; NOW WAIT TILL TIMER'S
                                                    OUTPUT IS LOW  读计时
                                                    器 2 输出电平
FA3F 2420       5446        AND     AL, 020H
FA41 75FA       5447        JNZ     W30         等该电平变成 0 态
                5448                            ; RELOAD  TIMER  WITH
                                                    PERIOD
                5449                            ; FOR NEXT DATA BIT
```




```
FA43 58         5450        POP     AX          ; RESTORE PERIOD COU-
                                                    NT
FA44            5451 W31:                       ; SET TIMER
FA44 E642       5452        OUT     042H, AL    ; SET LOW BYTE OF TIM-
                                                    ER 2  装入欲写入位的计
                                                    时值，该值在上一次计时电
                                                    平升为 1 态后生效
FA46 8AC4       5453        MOV     AL, AH
FA48 E642       5454        OUT     042H, AL    ; SET HIGH BYTE OF TI-
                                                    MER 2
```

```
FA4A C3          5455          RET
                 5456 WRITE_BIT     ENDP          返回
                 5457 ; ..........................................
FA4B             5458 CRC_GEN       PROC  NSAR  CRC_GEN 子程序
                 5459 ; UPDATE CRC REGISTER WITH NEXT DATA BIT  本程序
                      根据(CY)更新 CRC_REG 单元
                 5460 ;
                 5461 ; CRC IS USED TO DETECT READ ERRORS  入口参数:
                 5462 ;
                 5463 ; ASSUMES DATA BIT IS IN CARRY  (CY) = 新数据位
                 5464 ;
                 5465 ; REG AX IS MODIFIED  出口参数:
                 5466 ; FLAGS ARE MODIFIED  (AX), 标记位被修改
                 5467 ; ..........................................
FA4B A16900 R    5468          MOV   AX, CRC_REG
                 5469                              ; THE FOLLOWING INSTU-
                                                     CTIONS
                 5470                              ; WILL SET THE OVERFL-
                                                     OW FLAG
                 5471                              ; IF CARRY AND MS BIT
                                                     OF CRC
                 5472                              ; ARE UNEQUAL
FA4E D1D8        5473          RCR   AX, 1
FA50 D1D0        5474          RCL   AX, 1         CRC_REG 单元最高位内容
                                                     不等于(CY)则(OF)送 1
FA52 F8          5475          CLC                 ; CLEAR CARRY
FA53 7104        5476          JNO   W32           ; SKIP IF NO OVERFLOW
                 5477                              ; IF DATA BIT XORED
                                                     WITH
                 5478                              ; CRC REG BIT 15 IS ONE
FA55 351008      5479          XOR   AX, 0810H     ; THEN XOR CRC REG W-
                                                     ITH  (OF) = 1转此,异或入
                                                     一常数 0810H
                 5480                              ;
FA58 F9          5481          STC                 ; SET CARRY  (CY)送 1
FA59             5482 W32:
FA59 D1D0        5483          RCL   AX, 1         ; ROTATE CARRY (DATA
                                                     BIT)  数据位左环移至 AX
                 5484                              ; INTO CRC REG
```

```
FA5B A36900          R   5485              MOV   CRC_REG, AX
                                                         ; UPDATE CRC_
                                                           REG (AX)送 CRC
                                                           _REG 单元
FA5E C3                  5486              RET           ; FINISHED  返回
                         5487 CRC_GEN          ENDP
                         5488 ; ..........................................
FA5F                     5489 BEGIN_OP PROC    NEAR      ; START TAPE
                                                           AND DELAY
                                                           BEGIN_OP 子程
                                                           序
                         5490 ; 本子程序启动录音机马达然后延迟 0.66 秒
                         5491 ; ..........................................
FA5F E823FE              5492              CALL  MOTOR_ON
                                                         ; TURN ON MOTOR
                                                           启动马达
FA62 B342                5493              MOV   BL, 42H ; DELAY FOR TA-
                                                           PE DRIVE  外循
                                                           环初值
                         5494                            ; TO GET UP TO
                                                           SPEED (1/2 SEC)
FA64                     5495 W33:
FA64 B90007              5496              MOV   CX, 700H ; INNER LOOP = A-
                                                           PPROX. 10 MILL-
                                                           ISEC  内循环初值
FA67 E2FE                5497 W34:         LOOP  W34
FA69 FECB                5498              DEC   BL
FA6B 75F7                5499              JNZ   W33
FA6D C3                  5500              RET             延迟结束,返回
                         5501 BEGIN_OP         ENDP
                         5502 ; ..........................................
                         5503 ; CHARACTER GENERATOR GRAPHICS FOR 320
                                ×200 AND 640×200 GRAPHICS   下面是用于
                                320×200 和 640×200 图形模式的字形发生表
                         5504 ; ..........................................
FA6E                     5505 CRT_CHAR_GEN  LABEL  BYTE
FA6E 0000000000000000    5506    DB  000H, 000H, 000H, 000H, 000H, 000H,
                                     000H, 000H ; D_00
```

```
FA76 7E81A581BD99817E    5507  DB   07EH, 081H, 0A5H, 081H, 0BDH, 099H,
                                    081H, 07EH ; D__01
FA7E 7EFFDBFFC3E7FF7E    5508  DB   07EH, 0FFH, 0DBH, 0FFH, 0C3H, 0E7H,
                                    0FFH, 07EH ; D__02
FA86 6CFEFEFE7C381000    5509  DB   06CH, 0FEH, 0FEH, 0FEH, 07CH, 038H,
                                    010H, 000H ; D__03
FA8E 10387CFE7C381008    5510  DB   010H, 038H, 07CH, 0FEH, 07CH, 038H,
                                    010N, 008H ; D__04
FA96 387C38FEFE7C387C    5511  DB   038H, 07CH, 038H, 0FEH, 0FEH, 07CH,
                                    038H, 07CH ; D__05
FA9E 1010387CFE7C387C    5512  DB   010H, 010H, 038H, 07CH, 0FEH, 07CH,
                                    038H, 07CH ; D__06
FAA6 0000183C3C180000    5513  DB   000H, 000H, 018H, 03CH, 03CH, 018H,
                                    000H, 000H ; D__07
FAAE FFFFE7C3C3E7FFFF    5514  DB   0FFH, 0FFH, 0E7H, 0C3H, 0C3H, 0E7H,
                                    0FFH, 0FFH ; D__08
FAB6 003C664242663C00    5515  DB   000H, 03CH, 066H, 042H, 042H, 066H,
                                    03CH, 000H ; D__09
FABE FFC399BDBD99C3FF    5516  DB   0FFH, 0C3H, 099H, 0BDH, 0BDH, 099H,
                                    0C3H, 0FFH ; D__0A
FAC6 0F070F7DCCCCCC78    5517  DB   00FH, 007H, 00FH, 07DH, 0CCH, 0CCH,
                                    0CCH, 078H ; D__0B
FACE 3C6666663C187E18    5518  DB   03CH, 066H, 066H, 066H, 03CH, 018H,
                                    07EH, 018H ; D__0C
FAD6 3F333F303070F0E0    5519  DB   03FH, 033H, 03FH, 030H, 030H, 070H,
                                    0F0H, 0E0H ; D__0D
FADE 7F637F636367E6C0    5520  DB   07FH, 063H, 07FH, 063H, 063H, 067H,
                                    0E6H, 0C0H ; D__0E
FAE6 995A3CE7E73C5A99    5521  DB   099H, 05AH, 03CH, 0E7H, 0E7H, 03CH,
                                    05AH, 099H ; D__0F
                         5522
FAEE 80E0F8FEF8E08000    5523  DB   080H, 0E0H, 0F8H, 0FEH, 0F8H, 0E0H,
                                    080H, 000H ; D__10
FAF6 020E3EFE3E0E0200    5524  DB   002H, 00EH, 03EN, 0FEH, 03EH, 00EH,
                                    002H, 000H ; D__11
FAFE 183C7E18187E3C18    5525  DB   018H, 03CH, 07EH, 018H, 018H, 07EH,
                                    03CH, 018H ; D__12
FB06 6666666666006600    5526  DB   066H, 066H, 066H, 066H, 066H, 000H,
                                    066H, 000H ; D__13
```

```
FB0E 7FDBDB7B1B1B1B00    5527   DB   07FH, 0DBH, 0DBH, 07BH, 01BH, 01BH,
                                     01BH, 000H ; D__14
FB16 3E63386C6C38CC78    5528   DB   03EH, 063H, 038H, 06CH, 06CH, 038H,
                                     0CCH, 078H ; D__15
FB1E 000000007E7E7E00    5529   DB   000H, 000H, 000H, 000H, 07EH, 07EH,
                                     07EH, 000H ; D__16
FB26 183C7E187E3C18FF    5530   DB   018H, 03CH, 07EH, 018H, 07EH, 03CH,
                                     018H, 0FFH ; D__17
FB2E 183C7E1818181800    5531   DB   018H, 03CH, 07EH, 018H, 018H, 018H,
                                     018H, 000H ; D__18
FB36 181818187E3C1800    5532   DB   018H, 018H, 018H, 018H, 07EH, 03CH,
                                     018H, 000H ; D__19
FB3E 00180CFE0C180000    5533   DB   000H, 018H, 00CH, 0FEH, 00CH, 018H,
                                     000H, 000H ; D__1A
FB46 003060FE60300000    5534   DB   000H, 030H, 060H, 0FEH, 060H, 030H,
                                     000H, 000H ; D__1B
FB4E 0000C0C0C0FE0000    5535   DB   000H, 000H, 0C0H, 0C0H, 0C0H, 0FER,
                                     000H, 000H ; D__1C
FB56 002466FF66240000    5536   DB   000H, 024H, 066H, 0FFH, 066H, 024H,
                                     000H, 000H ; D__1D
FB5E 00183C7EFFFF0000    5537   DB   000H, 018H, 03CH, 07EH, 0FFH, CFFH,
                                     000H, 000H ; D__1E
FB66 00FFFF7E3C180000    5538   DB   000H, 0FFH, 0FFH, 07EH, 03CH, 018H,
                                     000H, 000H ; D__1F
                         5539
FB6E 0000000000000000    5540   DB   000H, 000H, 000H, 000H, 000H, 000H,
                                     000H, 000H ; SP D__20
FB76 3078783030003000    5541   DB   030H, 078H, 078H, 030H, 030H, 000H,
                                     030H, 000H ; ! D__21
FB7E 6C6C6C0000000000    5542   DB   06CH, 06CH, 06CH, 000H, 000H, 000H,
                                     000H, 000H ; " D__22
FB86 6C6CFE6CFE6C6C00    5543   DB   06CH, 06CH, 0FEH, 06CH, 0FEH, 06CH,
                                     06CH, 000H ; # D__23
FB8E 307CC0780CF83000    5544   DB   030H, 07CH, 0C0H, 078H, 00CH, 0F8H,
                                     030H, 000H ; $ D__24
FB96 00C6CC183066C600    5545   DB   000H, 0C6H, 0CCH, 018H, 030H, 066H,
                                     0C6H, 000H ; PER CENT D__25
FB9E 386C3876DCCC7600    5546   DB   038H, 06CH, 038H, 076H, 0DCH, 0CCH,
                                     076H, 000H ; & D__26
```

```
FBA6 6060C00000000000      5547  DB   060H, 060H, 0C0H, 000H, 000H, 000H,
                                      000H, 000H ; ' D__27
FBAE 1830606060301800      5548  DB   018H, 030H, 060H, 060H, 060H, 030H,
                                      018H, 000H ; ( D__28
FBB6 6030181818306000      5549  DB   060H, 030H, 018H, 018H, 018H, 030H,
                                      060H, 000H ; ) D__29
FBBE 00663CFF3C660000      5550  DB   000H, 066H, 03CH, 0FFH, 03CH, 066H,
                                      000H, 000H ; * D__2A
FBC6 003030FC30300000      5551  DB   000H, 030H, 030H, 0FCH, 030H, 030H,
                                      000H, 000H ; +D__2B
FBCE 0000000000303060      5552  DB   000H, 000H, 000H, 000H, 000H, 030H,
                                      030H, 060H ;, D__2C
FBD6 000000FC00000000      5553  DB   000H, 000H, 000H, 0FCH, 000H, 000H,
                                      000H, 000H ; -D__2D
FBDE 0000000000303000      5554  DB   000H, 000H, 000H, 000H, 000H, 030H,
                                      030H, 000H; . D__2E
FBE6 060C183060C08000      5555  DB   006H, 00CH, 018H, 030H, 060H, 0C0H,
                                      080H, 000H ; / D__2F
                           5556
FBEE 7CC6CEDEF6E67C00      5557  DB   07CH, 0C6H, 0CEH, 0DEH, 0F6H, 0E6H,
                                      07CH, 000H ; 0 D__30
FBF6 307030303030FC00      5558  DB   030H, 070H, 030H, 030H, 030H, 003H,
                                      0FCH, 000H ; 1 D__31
FBFE 78CC0C3860CCFC00      5559  DB   078H, 0CCH, 00CH, 038H, 060H, 0CCH,
                                      0FCH, 000H ; 2 D__32
FC06 78CC0C380CCC7800      5560  DB   078H, 0CCH, 00CH, 038H, 00CH, 0CCH,
                                      078H, 000H ; 3 D__33
FC0E 1C3C6CCCFE0C1E00      5561  DB   01CH, 03CH, 06CH, 0CCH, 0FEH, 00CH,
                                      01EH, 000H ; 4 D__34
FC16 FCC0F80C0CCC7800      5562  DB   0FCH, 0C0H, 0F8H, 00CH, 00CH, 0CCH,
                                      078H, 000H ; 5 D__35
FC1E 3860C0F8CCCC7800      5563  DB   038H, 060H, 0C0H, 0F8H, 0CCH, 0CCH,
                                      078H, 000H ; 6 D__36
FC26 FCCC0C1830303000      5564  DB   0FCH, 0CCH, 00CH, 018H, 030H, 030H,
                                      030H, 000H ; 7 D__37
FC2E 78CCCC78CCCC7800      5565  DB   078A, 0CCH, 0CCH, 078H, 0CCH, 0CCH,
                                      078H, 000H ; 8 D__38
FC36 78CCCC7C0C187000      5566  DB   078H, 0CCH, 0CCH, 07CH, 00CH, 018H,
                                      070H, 000H ; 9 D__39
```

```
FC3E 0030300000203000      5567  DB   000H, 030H, 030H, 000H, 000H, 030H,
                                      030H, 000H ; : D__3A
FC46 0030300000303060      5568  DB   000H, 030H, 030H, 000H, 000H, 030H,
                                      030H, 060H ; ; D__3B
FC4E 183060C060301800      5569  DB   018H, 030H, 060H, 0C0H, 060H, 030H,
                                      018H, 000H ; <D__3C
FC56 0000FC0000FC0000      5570  DB   000H, 000H, 0FCH, 000H, 000H, 0FCH,
                                      000H, 000H ; =D__3D
FC5E 6030180C18306000      5571  DB   060H, 030H, 018H, 00CH, 018H, 030H,
                                      060H, 000H ; >D__3E
FC66 78CC0C1830003000      5572  DB   078H, 0CCH, 00CH, 018H, 030H, 000H,
                                      030H, 000H ; ? D__3F
                           5573
FC6E 7CC6DEDEDEC07800      5574  DB   07CH, 0C6H, 0DEH, 0DEH, 0DEH, 0C0H,
                                      078H, 000H ; @ D__40
FC76 3078CCCCFCCCCC00      5575  DB   030H, 078H, 0CCH, 0CCH, 0FCH, 0CCH,
                                      0CCH, 000H ; A D__41
FC7E FC66667C6666FC00      5576  DB   0FCH, 066H, 066H, 07CH, 066H, 066H,
                                      0FCH, 000H ; B D__42
FC86 3C66C0C0C0663C00      5577  DB   03CH, 066H, 0C0H, 0C0H, 0C0H, 066H,
                                      03CH, 000H ; C D__43
FC8E F86C6666666CF800      5578  DB   0F8H, 06CH, 066H, 066H, 066H, 06CH,
                                      0F8H, 000H ; D D__44
FC96 FE6268786862FE00      5579  DB   0FEH, 062H, 068H, 078H, 068H, 062H,
                                      0FEH, 000H ; E D__45
FC9E FE6268786860F000      5580  DB   0FEH, 062H, 068H, 078H, 068H, 060H,
                                      0F0H, 000H ; F D__46
FCA6 3C66C0C0CE663E00      5581  DB   03CH, 066H, 0C0H, 0C0H, 0CEH, 066H,
                                      03EH, 000H ; G D__47
FCAE CCCCCCFCCCCCCC00      5582  DB   0CCH, 0CCH, 0CCH, 0FCH, 0CCH, 0CCH,
                                      0CCH, 000H ; H D__48
FCB6 7830303030307800      5583  DB   078H, 030H, 030H, 030H, 030H, 030H,
                                      078H, 000H ; I D__49
FCBE 1E0C0C0CCCCC7800      5584  DB   01EH, 00CH, 00CH, 00CH, 0CCH, 0CCH,
                                      078H, 000H ; J D__4A
FCC6 E6666C786C66E600      5585  DB   0E6H, 066H, 06CH, 078H, 06CH, 066H,
                                      0E6H, 000H ; K D__4B
FCCE F06060606266FE00      5586  DB   0F0H, 060H, 060H, 060H, 062H, 066H,
                                      0FEH, 000H ; L D__4C
```

```
FCD6 C6EEFEFED6C6C600    5587  DB   0C6H, 0EEH, 0FEH, 0FEH, 0D6H, 0C6H,
                                    0C6H, 000H ; M D__4D
FCDE C6E6F6DECEC6C600    5588  DB   0C6H, 0E6H, 0F6H, 0DEH, 0CEN, 0C6H,
                                    0C6H, 000H ; N D__4E
FCE6 386CC6C6C66C3800    5589  DB   038H, 06CH, 0C6H, 0C6H, 0C6H, 06CH,
                                    038H, 000H ; O D__4F
                         5590
FCEE FC66667C6060F000    5591  DB   0FCH, 066H, 066H, 07CH, 060H, 060H,
                                    0F0H, 000H ; P D__50
FCF6 78CCCCCCDC781C00    5592  DB   078H, 0CCH, 0CCH, 0CCH, 0DCH, 078H,
                                    01CH, 000H ; Q D__51
FCFE FC66667C6C66E600    5593  DB   0FCH, 066H, 066H, 07CH, 06CH, 066H,
                                    0E6H, 000H ; R D__52
FD06 78CCE0701CCC7800    5594  DB   078H, 0CCH, 0E0H, 070H, 01CH, 0CCH,
                                    078H, 000H ; S D__53
FD0E FCB4303030307800    5595  DB   0FCH, 0B4H, 030H, 030H, 030H, 030H,
                                    078H, 000H ; T D__54
FD16 CCCCCCCCCCCCFC00    5596  DB   0CCH, 0CCH, 0CCH, 0CCH, 0CCH, 0CCH,
                                    0FCH, 000H ; U D__55
FD1E CCCCCCCCCC783000    5597  DB   0CCH, 0CCH, 0CCH, 0CCH, 0CCH, 078H,
                                    030H, 000H ; V D__56
FD26 C6C6C6D6FEEEC600    5598  DB   0C6H, 0C6H, 0C6H, 0D6H, 0FEH, 0EEH,
                                    0C6H, 000H ; W D__57
FD2E C6C66C38386CC600    5599  DB   0C6H, 0C6H, 06CH, 038H, 038H, 06CH,
                                    0C6H, 000H ; X D__58
FD36 CCCCCC7830307800    5600  DB   0CCH, 0CCH, 0CCH, 078H, 030H, 030H,
                                    078H, 000H ; Y D__59
FD3E FEC68C183266FE00    5601  DB   0FEH, 0C6H, 08CH, 018H, 032H, 066H,
                                    0FEH, 000H ; Z D__5A
FD46 7860606060607800    5602  DB   078H, 060H, 060H, 060H, 060H, 060H,
                                    078H, 000H ; [ D__5B
FD4E C06030180C060200    5603  DB   0C0H, 060H, 030H, 018H, 00CH, 006H,
                                    002H, 000H ; BACKSLASH D__5C
FD56 7818181818187800    5604  DB   078H, 018H, 018H, 018H, 018H, 018H,
                                    078H, 000H ; ] D__5D
FD5E 10386CC600000000    5605  DB   010H, 038H, 06CH, 0C6H, 000H, 000H,
                                    000H, 000H ; CIRCUMFLEX D__5E
FD66 00000000000000FF    5606  DB   000H, 000H, 000H, 000H, 000H, 000H,
                                    000H, 0FFH ; __D__5F
```

```
                                5607
FD6E 3030180000000000          5608   DB   030H, 030H, 018H, 000H, 000H, 000H,
                                           000H, 000H ; D__60
FD76 0000780C7CCC7600          5609   DB   000H, 000H, 078H, 00CH, 07CH, 0CCH,
                                           076H, 000H ; LOWER CASE A D__61
FD7E E060607C6666DC00          5610   DB   0E0H, 060H, 060H, 07CH, 066H, 066H,
                                           0DCH, 000H; L. C. B D__62
FD86 00007BCCC0CC7800          5611   DB   000H, 000H, 078H, 0CCH, 0C0H, 0CCH,
                                           078H, 000H ;L. C. C D__63
FD8E 1C0C0C7CCCCC7600          5612   DB   01CH, 00CH, 00CH, 07CH, 0CCH, 0CCH,
                                           076H, 000H ; L. C. D D__64
FD96 000078CCFCC07800          5613   DB   000H, 000H, 078H, 0CCH, 0FCH, 0C0H,
                                           078H, 000H ; L. C. E D__65
FD9E 386C60F06060F000          5614   DB   038H, 06CH, 060H, 0F0H, 060H, 060H,
                                           0F0H, 000H; L. C. F D__66
FDA6 000076CCCC7C0CF8          5615   DB   000H, 000H, 076H, 0CCH, 0CCH, 07CH,
                                           00CH, 0F8H ; L. C. G D__67
FDAE E0606C766666E600          5616   DB   0E0H, 060H, 06CH, 076H, 066H, 066H,
                                           0E6H, 000H ; L. C. H D__68
FDB6 3000703030307800          5617   DB   030H, 000H, 070H, 030H, 030H, 030H,
                                           078H, 000H ; L. C. I D__69
FDBE 0C000C0C0CCCCC78          5618   DB   00CH, 000H, 00CH, 00CH, 00CH, 0CCH,
                                           0CCH, 078H ; L. C. J D__6A
FDC6 E060666C786CE600          5619   DB   0E0H, 060H, 066H, 06CH, 078H, 06CH,
                                           0E6H, 000H ; L. C. K D__6B
FDCE 7030303030307800          5620   DB   070H, 030H, 030H, 030H, 030H, 030H,
                                           078H, 000H ; L. C. L D__6C
FDD6 0000CCFEFED6C600          5621   DB   000H, 000H, 0CCH, 0FEH, 0FEH, 0D6H,
                                           0C6H, 000H ; L. C. M D__6D
FDDE 0000F8CCCCCCCC00          5622   DB   000H, 000H, 0F8H, 0CCH, 0CCH, 0CCH,
                                           0CCH, 000H ; L. C. N D__6E
FDE6 000078CCCCCC7800          5623   DB   000H, 000H, 078H, 0CCH, 0CCH, 0CCH,
                                           078H, 000H ; L. C. O D__6F
                                5624
FDEE 0000DC66667C60F0          5625   DB   000H, 000H, 0DCH, 066H, 066H, 07CH,
                                           060H, 0F0H ; L. C. P D__70
FDF6 000076CCCC7C0C1E          5626   DB   000H, 000H, 076H, 0CCH, 0CCH, 07CH,
                                           00CH, 01EH ; L. C. Q D__71
FDFE 0000DC766660F000          5627   DB   000H, 000H, 0DCH, 076H, 066H, 060H,
```

```
                                        0F0H, 000H ; L. C. R D__72
FE06 00007CC0780CF800    5628  DB       000H, 000H, 07CH, 0C0H, 078H, 00CH,
                                        0F8H, 000H ; L. C. S D__73
FE0E 10307C3030341800    5629  DB       010H, 030H, 07CH, 030H, 030H, 034H,
                                        018H, 000H ; L. C. T D__74
FE16 0000CCCCCCCC7600    5630  DB       000H, 000H, 0CCH, 0CCH, 0CCH, 0CCH,
                                        076H, 000H ; L. C. U D__75
FE1E 0000CCCCCC783000    5631  DB       000H, 000H, 0CCH, 0CCH, 0CCH, 078H,
                                        030H, 000H ; L. C. V D__76
FE26 0000C6D6FEFE6C00    5632  DB       000H, 000H, 0C6H, 0D6H, 0FEH, 0FEH,
                                        06CH, 000H ; L. C. W D__77
FE2E 0000C66C386CC600    5633  DB       000H, 000H, 0C6H, 06CH, 038H, 06CH,
                                        0C6H, 000H ; L. C. X D__78
FE36 0000CCCCCC7C0CF8    5634  DB       000H, 000H, 0CCH, 0CCH, 0CCH, 07CH,
                                        00CH, 0F8H ; L. C. Y D__79
FE3E 0000FC983064FC00    5635  DB       000H, 000H, 0FCH, 098H, 030H, 064H,
                                        0FCH, 000H ; L. C. Z D__7A
FE46 1C3030E030301C00    5636  DB       01CH, 030H, 030H, 0E0H, 030H, 030H,
                                        01CH, 000H ; D__7B
FE4E 1818180018181800    5637  DB       018H, 018H, 018H, 000H, 018H, 018H,
                                        018H, 000H ; D__7C
FE56 E030301C3030E000    5638  DB       0E0H, 030H, 030H, 01CH, 030H, 030H,
                                        0E0H, 000H ; D__7D
FE5E 76DC000000000000    5639  DB       076H, 0DCH, 000H, 000H, 000H, 000H,
                                        000H, 000H, ; D__7E
FE66 0010386CC6C6FE00    5640  DB       000H, 010H, 038H, 06CH, 0C6H, 0C6H,
                                        0FEH, 000H ; DELTA D__7F
                         5641 ; ..... INT 1A .....................................
                         5642 ; TIME__OF__DAY  软中断1A__取时间(TIME__
                                OF__DAY)子程序
                         5643 ; THIS ROUTINE ALLOWS THE CLOCK TO BE
                                SET/READ  本子程序可设置或取出当今时间值
                         5644 ;
                         5645 ; INPUT  入口参数:
                         5646 ; (AH) = 0 READ THE CURRENT CLOCK SET-
                                        TING  (AH) = 0 读当前时钟值
                         5647 ;          RETURNS CX = HIGH PORTION OF
                                COUNT  返回:  (DX) = 时钟高位部分
```

```
                5648 ;                  DX = LOW PORTION OF COUNT
                                        (DX) = 时钟低位部分
                5649 ;                  AL = 0 IF TIMER HAS NOT PASSED
                                        24 HOURS SINCE LAST READ
                                        (AL) = 0 自上次读取还未过 24 小时
                5650 ;                  <>0 IF ON ANOTHER DAY
                                        <>0 本次读取同上一次读取不在同一天
                5651 ; (AH) = 1 SET THE CURRENT CLOCK   (AH) = 1 设置时钟
                5652 ;      CX = HIGH PORTION OF COUNT  (CX) = 时钟高位部
                            分
                5653 ;      DX = LOW PORTION OF COUNT   (DX) = 时钟低位部
                            分
                5654 ; NOTE: COUNTS OCCUR AT THE RATE OF 1193180/65536
                     COUNTS/SEC  注意:每秒发生 18.2 次时钟中断
                5655 ;      (OR ABOUT 18.2 PER SECOND ··· SEE EQUATES
                            BELOW)
                5656 ; ..........................................
                5657        ASSUME CS: CODE, DS: DATA
FE6E            5658 TIME_OF_DAY PROC  FAR
FE6E FB         5659        STI                 ; INTERRUPTS BACK ON  允许
                                                  中断
FE6F 1E         5660        PUSH   DS           ; SAVE SEGMENT
FE70 50         5661        PUSH   AX           ; SAVE PARM
FE71 B84000  R  5662        MOV    AX, DATA
FE74 8ED8       5663        MOV    DS, AX       ; ESTABLISH ADDRESSING TO
                                                  VALUES  DS 指向数据段
FE76 58         5664        POP    AX           ; GET BACK INPUT PARM  取
                                                  回入口参数 AH
FE77 0AE4       5665        OR     AH, AH       ; AH = 0
FE79 7407       5666        JZ     T2           ; READ_TIME  读时钟?
FE7B FECC       5667        DEC    AH           ; AH = 1
FE7D 7416       5668        JZ     T3           ; SET_TIME  置时钟?
FE7F            5669 T1:                        ; TOD_RETURN
FE7F FB         5670        STI                 ; INTERRUPTS BACK ON  允
                                                  中,因 T2, T3 中关中了
FE80 1F         5671        POP    DS           ; RECOVER SEGMENT
FE81 CF         5672        IRET                ; RETURN TO CALLER  本子程
                                                  序返回
```

```
                        5673
FE82                    5674 T2:                        ; READ_TIME 读时钟
FE82 FA                 5675        CLI                 ; NO TIMER INTERRUP-
                                                        TS WHILE READING
                                                        禁止中断
FE83 A07000       R     5676        MOV   AL, TIMER_OFL 日溢出单元TIMER_OFL
                                                        非0表示已过了24小时
FE86 C606700000   R     5677        MOV   TIMER_OFL, 0  ; GET OVERFLOW, AND
                                                        RESET THE FLAG 清
                                                        日溢出单元
FE8B 8B0E6E00     R     5678        MOV   CX, TIMER_HIGH 送时钟值
FE8F 8B166C00     R     5679        MOV   DX, TIMER_LOW
FE93 EBEA               5680        JMP   T1            ; TOD_RETURN 转T1
                                                        返回
                        5681
FE95                    5682 T3:                        ; SET_TIME 置时钟
FE95 FA                 5683        CLI                 ; NO INTERRUPTS WHI-
                                                        LE WRITING
FE96 89166C00     R     5684        MOV   TIMER_LOW, DX
FE9A 890E6E00     R     5685        MOV   TIMER_HIGH, CX
                                                        ; SET THE TIME 时钟值
                                                        送TIMER_LOW, TIM-
                                                        ER_HIGH单元
FE9E C606700000 R       5686        MOV   TIMER_OFL, 0  ; RESET OVERFLOW 日
                                                        溢出单元送0
FEA3 EBDA               5687        JMP   T1            ; TOD_RETURN 转 T1
                                                        返回
                        5688 TIME_OF_DAY  ENDP
                        5689 ; ····· TIMER_INT 时钟中断处理程序
                        5690 ; THIS ROUTINE HANDLES THE TIMER INTERRUPT
                             FROM 本程序处理计时器0的时钟中断。该计时器的
                        5691 ; CHANNEL 0 OF THE 8253 TIMER.INPUT FREQUENCY
                             IS 1.19318 MHZ 输入频率是1.19318 MHZ, 分频值
                             65536, 它大约
                        5692 ; AND THE DIVISOR IS 65536, RESULTING IN APPROX.
                             18.2 INTERRUPTS 每秒发18.2次时钟中断。
                        5693 ; EVERY SECOND.
                        5694 ;
                        5695 ; THE INTERRUPT HANDLER MAINTAINS A COUNT
```

```
                                OF INTERRUPTS SINCE POWER  程序中有一中断次数
                                计数器,可用其内容
                        5696 ;  ON TIME, WHICH MAY BE USED TO ESTABLISH
                                TIME OF DAY. 计算日时,每次进入本子程序后软盘机马
                                达
                        5697 ;  THE INTERRUPT HANDLER ALSO DECREMENTS T-
                                HE MOTOR CONTROL COUNT  控制计数器均减1,减至
                                0后马上关闭软
                        5698 ;  OF THE DISKETTE, AND WHEN IT EXPIRES,  WILL
                                TURN OFF THE DISKETTE  盘机马达,复位马达运行标
                                志。
                        5699 ;  MOTOR, AND RESET THE MOTOR RUNNING  FLAGS
                        5700 ;  THE INTERRUPT HANDLER WILL ALSO  INVOKE  A
                                USER ROUTINE THROUGH INTERRUPT  本子程序最
                                后将发软中断 1CH, 调用
                        5701 ;  1CH AT EVERY TIME TICK. THE USER MUST CODE
                                A ROUTINE AND PLACE THE  用户自编的程序段. 该
                                程序的代码和
                        5702 ;  CORRECT ADDRESS IN THE VECTOR TABLE.  入口
                                地址应由用户编写和设置
                        5703 ;  .....
FEA5                    5704 TIMER_INT  PROC  FAR
FEA5 FB                 5705        STI                     ;  INTERRUPTS BACK ON
                                                            允使中断
FEA6 1E                 5706        PUSH    DS
FEA7 50                 5707        PUSH    AX
FEA8 52                 5708        PUSH    DX              ;  SAVE MACHINE  STATE
FEA9 B84000       R     5709        MOV     AX, DATA
FEAC 8ED8               5710        MOV     DS, AX          ;  ESTABLISH  ADDRESSA-
                                                            BILITY  DS 指向数据段
FEAE FF066C00     R     5711        INC     TIMER_LOW       ;  INCREMENT TIME  时钟
                                                            寄存单元(低位部分)增1表
                                                            示进入处理程序一次
FEB2 7504               5712        JNZ     T4              ;  TEST_DAY
FEB4 FF066E00     R     5713        INC     TIMER_HIGH      ;  INCREMENT HIGH WORD
                                                            OF TIME  低位部分加1后
                                                            变0,表示应向高位部分进1
FEB8                    5714 T4:                            ;  TEST_DAY
```

```
FEB8 833E6E0018    R 5715     CMP  TIMER_HIGH, 018H
                                   ; TEST FOR COUNT EQUALLING
                                     24 HOURS
FEBD 7519            5716     JNZ  T5          ; DISKETTE_CTL
FEBF 813E6C00B000 R 5717     CMP  TIMER_LOW, 0B0H  比较若(TIMER_HIGH,
                                   TIMER_LOW) = 0018, 00B0H 则满 24 个小时
FEC5 7511            5718     JNZ  T5          ; DISKETTE_CTL  (0018, 00B0H
                                     = 1573040 次中断 = 8643276923 秒
                                     ≈24 小时)
                     5719
                     5720 ;…TIMER HAS GONE 24 HOURS  满 24 小时
                     5721
EEC7 C7066E000000 R 5722     MOV  TIMER_HIGH, 0
FECD C7066C000000 R 5723     MOV  TIMER_LOW, 0   时钟寄存单元清 0
FED3 C606700001    R 5724     MOV  TIMER_OFL, 1   日溢出单元送 1
                     5725
                     5726 ;…TEST FOR DISKETTE TIME OUT  下面处理软盘机
                            马达
                     5727
FED8                 5728 T5:                ; DISKETTE_CTL
FED8 FE0E4000      R 5729     DEC  MOTOR_COUNT  软盘机马达控制寄存器内容
                                     减 1
FEDC 750B            5730     JNZ  T6          ; RETURN IF COUNT NOT OUT
FEDE 80263F00F0    R 5731     AND  MOTOR_STATUS, 0F0H
                                   ; TURN OFF MOTOR RUNNING
                                     BITS  减至 0, 马达运行单元低 4
                                     位清 0, 表示所有马达停止
FEE3 B00C            5732     MOV  AL, 0CH
FEE5 BAF203          5733     MOV  DX, 03F2H  ; FDC CTL PORT  DX指向 NEC
                                     控制器控制寄存器
FEE8 EE              5734     OUT  DX, AL     ; TURN OFF THE MOTOR  输出
                                     控制字, 关闭全部软盘机马达
                     5735
FEE9                 5736 T6:                ; TIMER_RET:
FEE9 CD1C            5737     INT  1CH        ; TRANSFER CONTROL TO A
                                     USER ROUTINE  发软中断 1CH,
                                     控制权暂转给用户
FEEB B020            5738     MOV  AL, EOI    控制权由用户交回
FEED E620            5739     OUT  020H, AL   ; END OF INTERRUPT TO 8259
```

```
                                                    发中断结束命令
FEEF 5A          5740    POP   DX
FEF0 58          5741    POP   AX
FEF1 1F          5742    POP   DS                    ; RESET MACHINE STATE
FEF2 CF          5743    IRET                        ; RETURN FROM INTERR-
                                                       UPT  本中断处理程序返回
                 5744 TIMER_INT  ENDP
                 5745 ; ..........................................
                 5746 ; THESE ARE THE VECTORS WHICH ARE MOVED INTO
                        下面是开机检查时移入数据段的中断向量
                 5747 ; THE 8086 INTERRUPT AREA DURING POWER ON
                 5748 ; ..........................................
FEF3             5749 VECTOR_TABLE LABEL WORD ; VECTOR TABLE FOR
                                                  MOVE TO INTERRUPTS
                 5750
FEF3 A5FE    R   5751    DW    OFFSET TIMER_INT  ; INTERRUPT 8  时钟中断
                                                    向量
FEF5 00F0    R   5752    DW    CODE
                 5753
FEF7 87E9    R   5754    DW    OFFSET KB_INT     ; INTERRUPT 9  键盘中断
                                                    向量
FEF9 00F0    R   5755    DW    CODE
                 5756
FEFB 00000000    5757    DD    0   ; INTERRUPT A
FEFF 00000000    5758    DD    0   ; INTERRUPT B
FF03 00000000    5759    DD    0   ; INTERRUPT C
FF07 00000000    5760    DD    0   ; INTERRUPT D
                 5761
FF08 57EF    R   5762    DW    OFFSET DISK_INT   ; INTERRUPT E  软盘机中
                                                    断向量
FF0D 00F0    R   5763    DW    CODE
                 5764
FF0F 00000000    5765    DD    0   ; INTERRUPT F
                 5766
FF13 65F0    R   5767    DW    OFFSET VIDEO_IO   ; INTERRUPT 10H  显示器
                                                    输入输出软中断向量
FF15 00F0    R   5768    DW    CODE
                 5769
FF17 4DF8    R   5770    DW    OFFSET EQUIPMENT  ; INTERRUPT 11H  取系统
```

```
FF19 00F0  R 5771  DW   CODE
           5772
FF1B 41F8  R 5773  DW   OFFSET MEMORY_SIZE_DETERMINE
                                        ; INT 12H  取内存RAM总容
                                          量软中断向量
FF1D 00F0  R 5774  DW   CODE
           5775
FF1F 59EC  R 5776  DW   OFFSET DISKETTE_IO  ; INTERRUPT 13H  软盘机输
                                          入输出软中断向量
FF21 00F0  R 5777  DW   CODE
           5778
FF23 39E7  R 5779  DW   OFFSET RS232_IO     ; INTERRUPT 14H  RS232异
                                          步通讯口输入输出软中断向量
FF25 00F0  R 5780  DW   CODE
           5781
FF27 59F8  R 5782  DW   OFFSET CASSETTE_IO  ; INTERRUPT 15H  盒带机输
                                          入输出软中断向量
FF29 00F0  R 5783  DW   CODE
           5784
FF2B 2EE8  R 5785  DW   OFFSET KEYBOARD_IO  ; INTERRUPT 16H  键盘输入
                                          输出软中断向量
FF2D 00F0  R 5786  DW   CODE
           5787
FF2F D2EF  R 5788  DW   OFFSET PRINTER_IO   ; INTERRUPT 17H  打印机输
                                          入输出软中断向量
FF31 00F0  R 5789  DW   CODE
           5790
FF33 0000    5791  DW   00000H              ; INTERRUPT 18H  ROM
                                          BASIC 入口点
FF35 00F6    5792  DW   0F600H              ; ROM BASIC ENTRY POINT
           5793
FF37 F2E6  R 5794  DW   OFFSET BOOT_STRAP   ; INTERRUPT 19H  引导程序
                                          软中断向量
FF39 00F0  R 5795  DW   CODE
           5796
FF3B 6EFE R 5797   DW   TIME_OF_DAY         ; INTERRUPT 1AH … TIME
                                          OF DAY  取日时软中断向量
```

```
FF3D 00F0      R 5798     DW    CODE
                  5799
FF3F 53FF      R 5800     DW    DUMMY__RETURN      ; INTERRUPT 1BH…KEYB-
                                                     OARD BREAK ADDR 哑
                                                     返回中断向量(BREAK键)
FF41 00F0      R 5801     DW    CODE
                  5802
FF43 53FF      R 5803     DW    DUMMY__RETURN      ; INTERRUPT 1C…TIMER
                                                     BREAK ADDR 哑返回中
                                                     断向量(计时器 BREA键)
FF45 00F0      R 5804     DW    CODE
                  5805
FF47 A4F0      R 5806     DW    VIDEO__PARMS       ; INTERRUPT 1D…VIDEO
                                                     PARAMETERS 显示器初
                                                     始参数表地址
FF49 00F0      R 5807     DW    CODE
                  5808
FF4B C7EF      R 5809     DW    OFFSET DISK__BASE  ; INTERRUPT 1E … DISK
                                                     PARMS 取软盘机参数软
                                                     中断向量
FF4D 00F0      R 5810     DW    CODE
                  5811
FF4F 00000000     5812     DD    0                  ; INTERRUPT 1F…POINTER
                                                     TO VIDEO EXT 第2字形
                                                     发生表地址
                  5813
FF53              5814 DUMMY__RETURN;
FF53 CF           5815     IRET                     ; DUMMY RETURN FOR
                                                     BREAK FROM KEYBOA-
                                                     RD 哑返回处理程序
                  5816 ;   …… INT 5 ……
                  5817 ;   THIS LOGIC WILL BE INVOKED BY INTERRUPT 05H
                           TO PRINT 软中断 5—打印屏幕子程序
                  5818 ;   THE SCREEN. THE CURSOR POSITION AT THE TIME
                           THIS ROUTINE 本子程序打印当前屏幕内容，程序返回时当
                  5819 ;   IS INVOKED WILL BE SAVED AND RESTORED UPON
                           COMPLETION THE 前光标位置保持不变。进入本程序后若
                           按
                  5820 ;   ROUTINE IS INTENDED TO RUN WITH INTERRUPTS E-
```

```
                        NABLED 下 Prsc 键,则该键被忽略. 单元 50:0 记
                5821 ;  IF A SUBSEQUENT PRINT SCREEN KEY IS DEPRES-
                        SED DURING THE 录打印屏幕后的状态.
                5822 ;  TIME THIS ROUTINE IS PRINTING IT WILL BE IGN-
                        ORED
                5823 ;  ADDRESS 50:0 CONTAINS THE STATUS OF THE PR-
                        INT SCREEN:
                5824 ;  出口参数
                5825 ;  50:0=0 EITHER PRINT SCREEN HAS NOT BEEN CA-
                               LLED 50 : 0=0 本子程序还未被调用或打印屏幕
                               操作
                5826 ;         OR UPON RETURN FROM A CALL THIS IN-
                               DICATES 顺利完成
                5827 ;         A SUCCESSFUL OPERATION.
                5828 ;
                5829 ;      =1 PRINT SCREEN IS IN PROGRESS =1 正打印
                               屏幕
                5830 ;
                5831 ;      =377 ERROR ENCOUNTERED DURING PRINTING
                                 =377 打印屏幕操作有错
                5832 ;
                5833    ASSUME CS:CODE, DS:XXDATA
                5834
FF54            5835 PRINT_SCREEN  PROC  FAR
FF54 FB         5836    STI                     ; MUST RUN WITH INTER-
                                                  RUPTS ENABLED 允使
                                                  中断
FF55 1E         5837    PUSH  DS                ; MUST USE 50:0 FOR D-
                                                  ATA AREA STORAGE
FF56 50         5838    PUSH  AX
FF57 53         5839    PUSH  BX
FF58 51         5840    PUSH  CX                ; WILL USE THIS LATER
                                                  FOR CURSOR LIMITS
FF59 52         5841    PUSH  DX                ; WILL HOLD CURRENT
                                                  CURSOR POSITION
FF5A BB5000     5842    MOV   AX, XXDATA        ; HEX 50 附加段基址 50H
                                                  送 DS, 该段只有一个字节,
                                                  地址 STATUS_BYTE
FF5D 8ED8       5843    MOV   DS, AX
FF5F 803E000001 5844    CMP   STATUS_BYTE, 1 ; SEE IF PRINT ALREADY
```

```
                                        IN PROGRESS  50:0 单元内容为 1
                                        表示现正进行屏幕打印
FF64 745F       5845    JZ      EXIT    ; JUMP IF PRINT ALREADY IN P-
                                        ROGRESS  =1 不打印.
FF66 C606000001 5846    MOV     STATUS_BYTE, 1
                                        ; INDICATE PRINT NOW IN PRO-
                                        GRESS  置正打印屏幕标记
FF6B B40F       5847    MOV     AH, 15  ; WILL REQUEST THE CURRENT
                                        SCREEN MODE
FF6D CD10       5848    INT     10H     ; (AL) = MODE  发软中断 10H 取回
                                        光标坐标,在返回参数
                5849                    ; (AH) = NUMBER COLUMNS/LINE
                                        中(AL) = CRT 模式, (AH) = 屏幕每
                                        行字符数
                5850                    ; (BH) = VISUAL PAGE  (BH) = 活
                                        动显示页页号
                5851    ; ********************
                5852    ;       AT THIS POINT WE KNOW THE COLUMNS/
                                LINE ARE IN
                5853    ;       (AX) AND THE PAGE IF APPLICABLE IS IN
                                (BH). THE STACK
                5854    ;       HAS DS, AX, BX, CX, DX PUSHED (AL) HAS
                                VIDEO MODE
                5855    ;
                5856    ; ********************
FF6F 8ACC       5857    MOV     CL, AH  ; WILL MAKE USE OF (CX) REG-
                                        ISTER TO
FF71 B519       5858    MOV     CH, 25  ; CONTROL ROW & COLUMNS
                                        (CH) = 25(行数)、(CL) = 列数
FF73 E85500     5859    CALL    CRLF    ; CARRIAGE RETURN LINE FEED
                                        ROUTINE  调 CPLF 向打印机发回
                                        车、换行命令
FF76 51         5860    PUSH    CX      ; SAVE SCREEN BOUNDS  保存行、
                                        列数
FF77 B403       5861    MOV     AH, 3   ; WILL NOW READ THE CURSOR.
FF79 CD10       5862    INT     10H     ; AND PRESERVE THE POSITION
                                        发软中断 10H, 读光标坐标, 出口参
                                        数:(BH) = 页号,(DX) = 光标坐标
FF7B 59         5863    POP     CX      ; RECALL SCREEN BOUNDS  屏幕
```

```
                                                    行、列数送 CX
FF7C 52         5864            PUSH    DX          ; RECALL (BH) = VISUAL PAGE
                                                    保存光标坐标
FF7D 33D2       5865            XOR     DX, DX      ; WILL SET CURSOR POSITION
                                                    TO (0, 0)  清 DX，它在下面程序
                                                    中作光标坐标，清 0 表示从屏幕左
                                                    上角开始打印
                5866 ; ********************
                5867 ;          THE LOOP FROM PRI10 TO THE INSTRUCTION PR-
                                IOR TO PRI20
                5868 ;          IS THE LOOP TO READ EACH CURSOR POSITION F-
                                ROM THE SCREEN
                5869 ;          AND PRINT.
                5870 ; ********************
FF7F B402       5871 PRI10:     MOV     AH, 2       ; TO INDICATE CURSOR SET R-
                                                    EQUEST
FF81 CD10       5872            INT     10H         ; NEW CURSOR POSITION EST-
                                                    ABLISHED  发软中断 10H，设置
                                                    光标新位置
FF83 B408       5873            MOV     AH, 8       ; TO INDICATE READ CHARAC-
                                                    TER
FF85 CD10       5874            INT     10H         ; CHARACTER NOW IN (AL)  发
                                                    软中断 10H，读取当前光标处的字
                                                    符码
FF87 04C0       5875            OR      AL, AL      ; SEE IF VALID CHAR
FF89 7502       5876            JNZ     PRI 5       ; JUMP IF VALID CHAR  字符码
                                                    非 0、有效
FF85 B020       5877            MOV     AL,         ; MAKE A BLANK  字符码为 0，
                                                    送空格码
FF8D            5878 PRI15:
FF8D 52         5879            PUSH    DX          ; SAVE CURSOR POSITION  保存
                                                    当前光标位置
FF8E 33D2       5880            XOR     DX, DX      ; INDICATE PRINTER 0  清 DX，
                                                    用 0 号打印机
FF90 32E4       5881            XOR     AH, AH      ; TO INDICATE PRINT CHAR IN
                                                    (AL)
FF92 CD17       5882            INT     17H         ; PRINT THE CHARACTER  发软
                                                    中断 17H，将(AL)打印出来
```

```
FF94 5A          5883          POP     DX          ; RECALL CURSOR POSIVION
FF95 F6C425      5884          TEST    AH, 25H     ; TEST FOR PRINTER ERROR
                                                   检查打印后状态信息，若出现无
                                                   纸、I/O 错或超时错则转 ERR10
FF98 7521        5885          JNZ     ERR10       ; JUMP IF ERROR DETECTED
FF9A FEC2        5886          INC     DL          ; ADVANCE TO NEXT COLU-
                                                   MN  光标向右进一字符位置
FF9C 3ACA        5887          CMP     CL, DL      ; SEE IF AT END OF LINE
                                                   光标将移出屏幕右边缘?
FF9E 75DF        5888          JNZ     PRI10       ; IF NOT PROCEED  否，转
                                                   PRI10 继续打印同行字符
FFA0 32D2        5889          XOR     DL, DL      ; BACK TO COLUMN 0  列号
                                                   清 0
FFA2 8AE2        5890          MOV     AH, DL      ; (AH) = 0
FFA4 52          5891          PUSH    DX          ; SAVE NEW CURSOR POS-
                                                   ITION  新光标坐标暂存
FFA5 E82300      5892          CALL    CRLF        ; LINE FEED CARRIAGE R-
                                                   ETURN  让打印机回车，转行
FFA8 5A          5893          POP     DX          ; RECALL CURSOR POSITION
FFA9 FEC6        5894          INC     DH          ; ADVANCE TO NEXT LINE
                                                   行号加 1
FFAB 3AEE        5895          CMP     CH, DH      ; FINISHED?  整帧屏幕打印
                                                   完毕?
FFAD 75D0        5896          JNZ     PRI10       ; IF NOT CONTINUE  否，转
                                                   PRI10，打印下一行字符
FFAF 5A          5897 PRI20:   POP     DX          ; RECALL CURSOR POSITION
                                                   取回光标原坐标
FF89 B402        5898          MOV     AH, 2       ; TO INDICATE CURSOR SET
                                                   REQUEST  发软中断 10H，
                                                   将光标置回原先位置
FFB2 CD10        5899          INT     10H         ; CURSOR POSITION RESTO-
                                                   RED
FFB4 C606000000  5900          MOV     STATUS_BYTE, 0
                                                   ; INDICATE FINISHED  50:0
                                                   单元清 0，表示打印屏幕顺利
                                                   完成
FFD9 EB0A        5901          JMP     SHORT EXIT  ; EXIT THE ROUTINE
FFBB 5A          5902 ERR10:   POP     DX          ; GET CURSOR POSITION
FFBC B402        5903          MOV     AH, 2       ; TO REQUEST CURSOR SET
```

```
FFBE CD10          5904          INT     10H          ; CURSOR POSITION RESTO-
                                                        RED 发生打印机错误，复原
                                                        光标
FFC0 C6060000FF    5905 ERR20:   MOV     STATUS_BYTE, 0FFH
                                                        ; INDCATE ERROR 50:0 单
                                                        元送入377(FFH)，错误标记之
                   5906
FFC5 5A            5907 EXIT:    POP     DX           ; RESTORE ALL THE REGI-
                                                        STERS USED 恢复DX，
                                                        CX，BX，AX，DS
FFC6 59            5908          POP     CX
FFC7 5B            5909          POP     BX
FFC8 58            5910          POP     AX
FFC9 1F            5911          POP     DS
FFCA CF            5912          IRET                 返回
                   5913 PRINT_SCREEN ENDP
                   5914
                   5915 ; ……     CARRIAGE RETURN, LINE FEED SUBROUTINE
                                 下面程序段使打印机回车，换行
                   5916
FFCB               5917 CRLF     PROC    NEAR
FFCB 33D2          5918          XOR     DX, DX       ; PRINTER 0 选择0号打印
                                                        机
FFCD 32E4          5919          XOR     AH AH        ; WILL NOW SEND INITIAL
                                                        LF, CR TO PRINTER 清
                                                        AH
FFCF B00A          5920          MOV     AL, 12Q      ; LF 12Q=001, 010=10(10)
                                                        换行符
FFD1 CD17          5921          INT     17H          ; SEND THE LINE FEED 送
                                                        换行符，让打印机换行
FFD3 32E4          5922          XOR     AH, AH       ; NOW FOR THE CR
FFD5 B00D          5923          MOV     AL, 15Q      ; CR 15Q=001, 101=13(10)
                                                        回车符
FFD7 CD17          5924          INT     17H          ; SEND THE CARRIAGE R-
                                                        ETURN 送回车符，让打印机
                                                        回车
FFD9 C3            5925          RET
                   5926 CRLF     ENDP    CRLF         子程序返回
                   5927 CODE     ENDS    CODE         BIOS 代码段完
```

```
                    5928
                    5929 ; ..........................................
                    5930 ; POWER ON RESET VECTOR  下面是开机复位向量段
                    5931 ; ..........................................
FFFF                5932 VECTOR SEGMENT AT 0FFFFH
                    5933
                    5934 ; ..... POWER ON RESET
                    5935
0000 EA5B0000F0 R   5936    JMP  RESET  转至 RESET, 复位系统
                    5937
0005 30342F32342F38 5938    DB   '04/24/81'  ; RELEASE MARKER
     31
                    5939 VECTOR ENDS
                    5940         END
```

第三十一章　IBM PC 的操作系统

随着微型计算机的出现和迅速发展，微型计算机的两个关键部分硬件和软件已经相互渗透得很深了。这意味着：硬件中有软件，软件中亦有硬件。因此，不了解两者及相互间的关系，也就不能理解微机系统和它的应用。为了讲解方便，仍然把软件和硬件分开进行。

如果说微型机中的微处理器即 CPU 是计算机硬件核心的话，那末操作系统就是计算机软件的核心，从某种意义上说是计算机系统的核心。体现出计算机系统特点的操作系统，是程序设计语言及记帐、字处理、筹划等应用程序的基础。所以为计算机系统选定操作系统不是一件小事，它关系到在PC系统上能运行什么程序、使用何种语言，PC系统对用户来说具有何种"风格"。为此，在这一章讨论为 PC 配备的五种操作系统：IBM PC DOS、Digital Research 公司的 CP/M-86，加里福尼亚大学 San Diego Campus 的P-系统，Phase One 公司的 OASIS-16 和 Unix。显然这是从软件角度了解 IBM 个人计算机的第一步。

§31.1　CP/M-86

一、CP/M 系列

CP/M(Control Program/Microprocessor 第一字母的缩写)是 Digifal Research 公司的产品。在当时它主要用来减轻开发程序语言的负担。后来因为出现了具有大容量记录能力的软盘机，CP/M 又被扩展成软盘机操作系统，并增添了文本编辑、8080 机器代码汇编程序和查错程序。

二、CP/M-80

CP/M-80 原是为 Intel 8008 8 位微处理器研制的操作系统，它在配备 Intel 8080、8085、Zilog Z80 CPU 的系统上也能发挥作用。CP/M-80 除了在不断成熟的过程中保持 CP/M 是开发机器语言的理想工具特点外，还向高级语言如 BASIC、FORTRAN、COBOL 提供种种支持。结果 CP/M-80 就能支撑一大批应用程序。CP/M-80 和 Microsoft 公司 BASIC-80 是微型计算机世界的两个实际上的标准。

三、CP/M-86

CP/M-86 是在 16 位微机上工作的 CP/M 系列操作系统。由于 Intel 8088 和 8086 只在数据总线一次传送字节数上有差别，而在内部结构和指令系统几乎完全一样。所以，CP/M-86 在 8088 这种特殊的 8 位 CPU 上也能照常工作。下面我们就针对在 16 位微机上运行的 CP/M-86 进行一些讨论。

如图 31-1 所示，CP/M-86(CP/M 系列基本如此)由四大部分组成。它们是 BIOS(基本输入/输出系统)、BDOS(基本磁盘操作系统)、CCP (控制台命令处理程序)和 TPA (暂住程序区)。

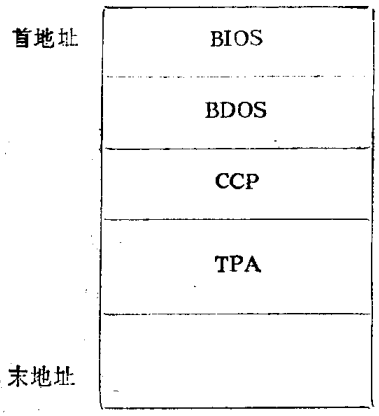


图 31-1 CP/M-86 各块在存贮区的映像

1. BIOS

BIOS 是覆盖在硬件上的第一层软件，负责处理 PC 各硬件功能块间的数据通讯。具体说来，BIOS 接收键盘发来的字符，向显示器发送待显示的字符、图形；向打印机输出要打印的字符；核查软盘机数量及其工作是否正常等情况。因为 BIOS 是 CP/M-86 中唯一同硬件打交道的软件模块，故基于 BIOS 层的 BDOS 工作时就无需考虑软盘机及其他 I/O 设备的种种细节。

2. BDOS

BDOS 是一组依赖 BIOS 工作的标准化子程序，主要负责软盘管理。功能有在磁盘上找出空白空间；为新写入的文件起名或将旧文件更改名字；有序地读出磁盘文件和写入磁盘文件；有关磁盘机自身的一些事务性信息也由 BDOS 保管。BDOS 另一项工作是初始化和协调与键盘、终端、打印机、调制解调器等其它个人计算机资源的通讯。由于用户程序到 BDOS 到 BIOS 都以系统调用方式进行，所以下面我们用一个字处理程序的典型调用来说明。假如字处理程序向 BDOS 发出希望从操作员得到一个字符的请求，也就是在 8088 CPU 特定寄存器放入适当参数后跳转至指定存贮单元的系统调用。BDOS 在掌握了控制权后知道这是一个请求字符的调用。接着 BDOS 向 BIOS 发出接收字符并加以显示的系统调用，自己则进入等待循环。一俟 BIOS 报告接收和显示已完成，BDOS 就把该字符送给 CPU，并把控制权交还给字处理程序。当用户要求把文件存入磁盘时，字处理程序便把很多条命令发给BDOS。BDOS 则执行下列命令。如寻找空白空间，在磁盘文件目录中填写新文件的名字，以一次 128 字节的速度写文件块，更新磁盘目录以及下一次写入块的地址等。同时 BDOS 也频繁地调用 BIOS，让它完成真正的磁盘写入工作。所以 BDOS 的真正任务是解释命令，作一些必要的登记工作。

3. CCP

CCP 是 CP/M 被装入工作区时一并存入的控制台命令处理程序。CCP 是操作员和 CP/M间的解释程序，它具有完成事务命令的执行和向 TPA 区装入程序两大功能。这两大功能等价于六个不同的子程序。前五个子程序(事务命令)由 DIR、ERA、REN、TYPE、USER 5 条命令调用，由于他们不时被调用且常驻在 CCP 中，所以也称为常驻命令。这 5 条命令的作用如下表所示。

当向 CCP 发出的不是上面 5 条命令时，CCP 就调用第六个子程序，由它调入并执行命令指定的程序。这里的程序可以是用户程序，也可以是 CP/M-86 提供的暂时(外部)程序，

表 31-1 CP/M-86 常驻命令/子程序

名 称	作 用	名 称	作 用
DIR	显示磁盘文件名	TYPE	显示磁盘文件的内容
ERA	擦除磁盘文件	USER	更换用户号
REN	重新命名磁盘文件		

暂时程序如下表所示：

表 31-2 CP/M-86 暂时(外部)命令/程序

名 称	作 用	名 称	作 用
ASM 86	8086 汇编程序	PIP	外设接口程序
ASSIGN	更改逻辑设备名程序	PROTOCOL	设置通讯协议
COPYDISK	拷贝磁盘程序	SPEED	设置通讯口数据传输速度和特点
DDT 86	动态检错程序	STAT	磁盘状态及统计
ED	行编辑程序	SUMIT	批处理程序
FUNCTION	为功能键指派串程序	TOD	显示/设置日期,小时
GENCOM	命令程序发生程序		
HELP	显示救援文件		
NEWDISK	磁盘格式化后把操作系统写入磁盘		

4. TPA

暂时程序区 TPA 既不是一个程序也不属于 CP/M 操作系统，它是存放暂时命令及暂时程序段的 RAM 区域，即整个 RAM 区减去 CP/M 占据的 RAM 空间后剩余的 RAM 空间。在把暂时程序装入 TPA 区之前，应将 CCP 从它占据的一部分 TPA 区域退出。暂时程序运行结束后，CCP 又被重新装进 TPA 区。这种再装入称为热启动。由于热启动的存在，在用户退出一个程序和用命令调入另一个程序之间有一段时间的延迟。

四、CP/M-86 的优点及限制

CP/M-86 的优点可从程序员角度和终端用户角度两方面来考察，因为两者的出发点和要求均是不同的。

CP/M 是微机形成时期开发的产物，至今它仍然保留着纸带穿孔机和读带机的陈旧名词。但是以 PL/M 高级语言写的 CP/M 保持至今的优点是它的可移植性。它可在 8080/8085、Z80、8086/8088 为 CPU 的微机上工作。CP/M-86 拥有在汇编语言环境中工作的必要工具。如汇编程序、行编辑程序和汇编语言检错程序。CP/M-86 也能支持高级语言而且相当可靠。CP/M 能支持 CBASIC-86、Pascal MT、PT/M-86 等高级语言。

CP/M 在外设管理及存贮区分配方面也有其特点。譬如 IBM-PC 连接羽毛球形打印机时，由于该种打印机与一般点阵式打印机的接口协议不同，所以连接必须由操作系统加以协调。CP/M 的 IOBYTE 正是这样一个协调程序。它可以对不同通讯协议的打印机进行快速

切换，这对程序员和终端用户来说是很有价值的。8088/8086 为 CPU 的系统的存贮区有 1 兆字节，分成 16 个不同的区块，每块 64KB，CPU 一次只能访问一个块。CP/M-86 的内存管理程序允许用户在高级语言中提出跨越块的请求，进而实现了缓冲空间的动态分配。终端用户看到的只是跨越块后的结果。

由于 CP/M-80 软件基础的缘故，能在 CP/M-80 环境下运行的程序并不能在CP/M-86 环境下运行，即软件不向上兼容。但另一方面，CP/M-80 同 CP/M-86 又极为相似，利用模拟程序等很容易重新编译 8 位版本的程序，使之在 CP/M-86 环境下工作，也就是说在 IBMPC 上工作。目前可用的这类转换程序有 Xedex′ Baby blue Z80 卡和 Dynamic Microprocessor Association 的 EM80/86。CP/M-86 的最大的优点可以它的另外两个变体版本 MPM-86 和并发 CP/M-86 中找到。

MP/M-86 是多道程序监控 (Multi Programming Monitor/Microcomputer)的 8086 版本，它是在微型机上支持多用户、多任务的一个操作系统。只要具有适当的计算机资源，多个用户就能同时分享机器资源。但个人计算机上有限的外设扩展口和为数不多的磁盘机限制了多用户、多任务的使用。

并发 CP/M-86 利用 8086/8088 CPU一次只能访问 64KB 区中内容的特点，允许单用户在不同的 64 KB RAM 区装入不同的程序段，并发地在 CPU 上运行，即并发 CP/M-86 实现了单用户、多任务的并发运行。见下图 31-2 CP/M-86 与并发 CP/M-86比较，图中并发 CP/M-86 的 XIOS 是扩充输入/输出系统，类似于 BIOS。XDOS 是 DOS 的扩充，拥有管理用户不同程序段的实时管理程序，Session manager 负责操作员和 RAM 区中不同程序段的通讯，*n* 个TPA 区可装入不同的任务。

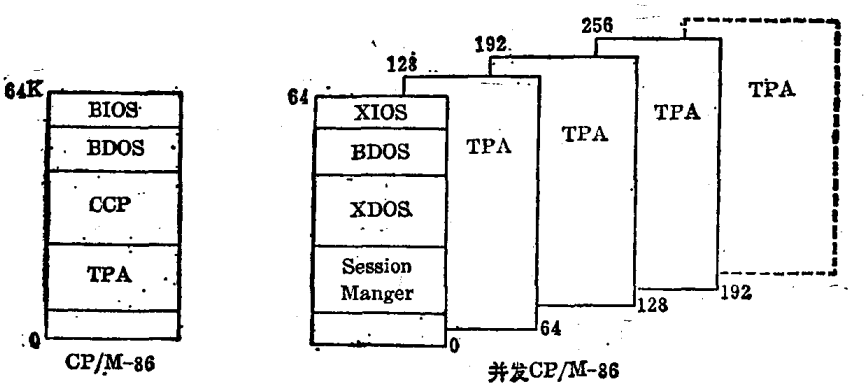


图 31-2　CP/M-86 与并发 CP/M-86 的比较

CP/M-86 的不足之处主要有三点。一是对新用户具有某种神秘感，不太容易掌握。例如复制磁盘文件的命令规定新文件名在前，当前文件名在后，这似乎在逻辑上颠倒了。第二点是 CP/M-86 的错误信息不够明确，极易混淆。譬如用户在命令、程序中用错了磁盘机号，CCP 只在 CRT 上显示一个问号和程序的名字，用户必须自己找出命令中的语法错误。另外，CP/M-86 在内存中保留有磁盘目录。假如用户不提醒 CP/M-86 就替换磁盘机中的磁盘，则下次用到这张盘时，CP/M-86 就认为它只是一张只读磁盘，用户若要写磁盘的话，CRT 上就显示出“BDOS ERROR ON A：R/O”，并中止程序。还有一个致命的错误是“BDOS ERROR ON A: BAD SECTOR”。若用户不熟悉处理这些错误的程序，以上错误信息的出现便意味着前面的工作全功尽弃。第三点是 CP/M-86 管理的磁盘的容量不能超过 8 兆字

节。所以若使用 20～40 兆字节温切斯特磁盘机的话，必须在逻辑上把它分成若干个 8 兆字节的逻辑磁盘机。对磁盘容量的限制导致任何文件的长度不能超过 8 兆字节。尽管这样大的文件并不多见，但终究给用户带来了不便。

总的说来，CP/M-86 是一个完整的、以 CP/M-80 为基础的微机操作系统，它是程序员的一个重要工具，但有时使新用户感到迷惑。CP/M 四大组成块的概括见表 31-3。

表 31-3 CP/M 四大组成块的名称及功能特点

名　　称	功　能　特　点
BIOS 基本输入输出系统	依赖于机器硬件，是 RIOS (ROM BIOS) 的扩充，在 BIOS 之上是 BDOS. BIOS 负责所有外设的输入和输出管理
BDOS 基本磁盘操作系统	不依赖机器硬件. 初始化和控制所有外设的通讯，处理终端、打印机以及磁盘的输入/输出，完成 CCP 的热启动. BDOS 还包括 CP/M-86 的文件系统，报告致命错误.
CCP 控制台命令处理程序	不依赖机器硬件. 由DIR、ERA、REN、TYPE、USER 和用户程序装入程序六大子程序组成.
TPA 暂时程序区	IBM-PC 的 RAM 区，用户程序运行及数据保存区域

§31.2 PC-DOS

PC-DOS 于 1980 年由西亚图计算机产品公司的 Tim Paterson 用 Z80 指令写成，当时的名字是 MS-DOS。后来，Paterson 又设计了一个交叉汇编程序，用于把 Z80 指令写成的 PC-DOS 翻译成 8086 汇编语言程序，这样就产生了在 8086 CPU上运行的第一个 PC-DOS：86-DOS 版本 0.1。以后 Microsoft 公司取得了 86-DOS 的出销权，并为 86-DOS 规定了商标 MS-DOS。随着 86-DOS 销售权的扩散，一些公司陆续以 MS-DOS 或其它商标命名86-DOS。例如 IBM 公司的 PC-DOS(IBM personal Computer Disk Operating System)就是其中一个。

一、PC-DOS 的结构

若把 PC-DOS 与 CP/M-80、86 作一比较，我们就能发现两者间的相同之处多于不同之处。下面就借助讨论 CP/M 时讲的例子和述语来介绍 PC-DOS。

正如 CP/M-86 能被分成四个相互分离的功能区(BIOS、BDOS、CCP 和 TPA)一样，PC-DOS 也分成好几个功能区，它们是 BIOS、DOS、命令处理程序 COMMAND 和程序存贮段(见图 29-3)。程序存贮段类似于 CP/M 中的 TPA。我们将在 CP/M-86 同 PCDOS 的比较一节中讨论两者的对比情况。

1. BIOS

BIOS 是 PC-DOS、ROM、BIOS(RIOS)的扩充，它除了管理 RIOS 负责的键盘、单色或彩色/图形显示器、盒带机、扬声器及打印机外，还管理磁盘机。在 PC 系统里，BIOS 是一个名为 IBMIO·COM 的系统文件，此文件不登在磁盘文件目录上，很难修改。

2．IBMDOS.COM

IBMDOS.COM 等价于 CP/M-86 的 BDOS。IBMDOS.COM 简称 DOS。它负责在磁盘上写入和读出信息(比 BIOS 高一级)，DOS 还有一组并行管理程序，管理非磁盘文件。

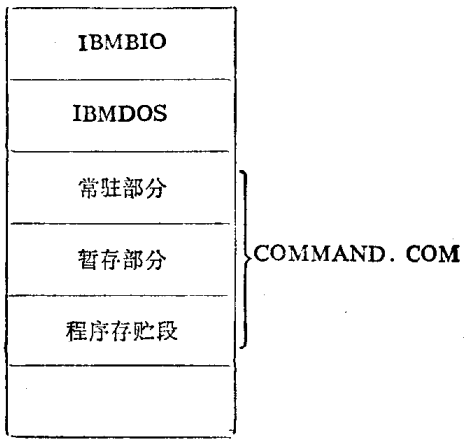


图 31-3　PC DOS 存贮区映像

3．COMMAND.COM

PC DOS 的 COMMAND.COM 相当于 CP/M-86 中的 CCP，它完成两个任务：(1)执行常驻子程序；(2)装入执行用户程序。表 31-4 列出了 PC DOS 内部常驻子程序名/命令。其中 COPY 命令和 Batch 命令 CP/M-86 是没有的。COPY 是把数据从源设备转送到目设备的通用命令，如在磁盘间传送文件；把磁盘文件送打印机打印或显示器显示；接收通讯口送来的字符，把它们装配成磁盘文件。Batch 命令启动一个写有数条操作系统命令的 Batch 文件(批处理文件)。操作系统依次执行这些命令，用户无需从键盘键入这些命令。Batch 命令的一个实用例子是在异步通讯程序软盘上复制 PCDOS。这时 Batch 命令先指导用户装入正确的磁盘，然后执行 SYS 命令，进行复制工作。另一个在启动系统时被搜寻、执行的 Batch 文件是 AUTOEXEC.BAT 文件。该文件含有 Visi Corp 公司的 Visicale 和 Sorcim 公司 Super Calc 特殊显示程序的名字，如 40/80 列显示程序。利用 Batch 文件的特点，用户可以不修改显示文件而只修改 AUTO EXEC.BAT 文件中的命令，以适应不同的显示要求。

COMMAND 与 CCP 的最后一个差别是他们进行事务处理的方法。COMMAND 处理所有的紧急中断(设备请求注意)、致命错误(主要出自磁盘)和程序终了时的结束事务。所以程序结束后 COMMAND 检查自身的完整性，如果发现结束程序使用了本由 COMMAND 占据的区间，COMMAND 便自动从磁盘把自己调进内存。固不到万不得已时，PC DOS 是不进行热启动的，这可加快撤下某一程序和装上新程序的过程。

表 31-5 列出了属于 COMMAND 的外部子程序/命令名。其中的 DATA 及 TIME 命令 CCP 是没有的，用户用他们写上创立或编辑文件的时间、日期。

下面是 PC DOS 各大部分的名称及功能特点表，我们将在它的后面较详细地介绍 PC DOS 的命令、行编辑程序和连接程序的用法。

表 31-4 PC DOS 内部常驻子程序/命令

名　　称	作　　用
Batch	批处理
COPY	把数据从一个设备复制至另一个设备
DIR	显示文件
ERASE	擦除磁盘文件
PAUSE	暂停，等待键盘输入(用于 Batch 文件)
REM	显示注释(Batch 文件用)
RENAME	更改磁盘文件名
TYPE	显示文件

表 31-5 PC DOS 外部暂时命令/程序

名　　称	作　　用
CHKDSK	磁盘和 RAM 空白区统计
COMP	磁盘文件比较
DATE	显示/设置日期
DISKCOMP	比较两张磁盘
DISKCOPY	复制整张磁盘
FORMAT	磁盘格式化后把操作系统拷贝上
MODE	设置显示器和打印机的模式
SYS	将操作系统拷贝在磁盘上
TIME	显示/设置时间

表 31-6 PC DOS 各大部分的名称及功能特点

名　　称	功　　能　　特　　点
IBMBIO	依赖机器硬件．扩充了 ROM BIOS 与 IBMDOS 接口，处理同全体外设的通讯
IBMDOS	不依赖机器硬件．与 IBMBIO、COMMAND 和应用程序接口．初始并控制与全体外设的通讯；处理终端、打印机、磁盘的输入、输出，含有完整的 PC DOS 文件系统．在操作系统启动后装入 COMMAND 文件．
COMMAND.COM	常驻部分：处理程序退出、CTRL-BREAK 键发致命错误中断，与 IBMDOS、用户程序、COMMAND、COM 、暂时部分接口．程序结束时检查暂时部分区域是否完整，不完整的话从外存调进 COMMAND.COM．报告致命错误．暂时部分：与 COMMAND．COM 常驻部分、IBMDOS 和操作员接口，含有 COPY、DIR、ERASE、RENAME、和 TYPE 五个子程序，控制批处理文件和处理 PAUSE、REM 批处理命令，初始用户程序的装入．

（续表）

名　　称	功　　能　　特　　点
程序存贮段	记录用户程序和数据，该段中的程序可复盖 COMMAND. COM 的暂时空间.

二、PC DOS 命令

1．命令参数

d：指定驱动器。A：表示系统第一台驱动器，B：表示第二台。空缺时系统自己选择缺省驱动器。

filename 磁盘文件名。长度不应超过 8 个字符。filename 后可跟扩充名。文件名的字符应取自下列字符集：A—Z，0—9

$ & # @ !
% , ( ) —
〈 〉 { } —
′ ∧ ～ ¦ ‘

PC DOS 为一些外设保留了名字，这些名字可用在磁盘文件名中但不能单独作为磁盘文件名。用保留名作为磁盘文件名时驱动器指定名将被忽略。下面是 PC DOS 的保留名及它们的含义。

CON 控制台键盘或屏幕，当它作为输入设备使用时，先按F6 键后按ENTER键能中止它。

AUX 或 COM1　第一异步通讯转接口

LPT1 或 PRN　打印机(只作为输出设备)

NUL　实际上不存在的设备(哑设备)。用来进行测试工作。作为输入设备时立即产生文件结束(end-of-file)。作为输出设备时模拟写操作，但不写数据。

ext　磁盘文件扩展名，由一个句号和 1 至 3 个字符(取自文件名字符集)组成。

filespec　指定某驱动器上的文件[d:]filename[.ext]

例如：B：myprog.COB

A：yourprog

DATAFILE.pas

Cobfile

？　全程量文件名字符，代表文件名、扩展名中的一个字符位置，该位置可以是任何字符。

例如：DIR AB?DE.XYZ 将列出所有其文件名之第一、二、三、四、五字符是 A、B、D、E；第三字符为任意字符的文件名。

＊ 全程序文件名字符，表示*位置及后面全部字符(包括扩展名若没有跟扩展名的话)。

例如：DIR AB*.XYZ 将显示出

ABCDE.XYZ

ABC357.XYZ

ABIDE.XYZ

ABIIOU.XYZ

ABO$$$.XYZ

ABODE.XYZ

2．批处理(batch)命令

格式　[d:]filename[参数]

类型　内部

作用　执行指定文件中的命令

注意：

① 不要打 batch，除非批处理文件名是 BATCH.BAT。

② 只要求输入 filename，不要跟扩展名。

③ 执行的是 filename.BAT 文件。

④ 批处理时按 CTRL-BREAK 键，屏幕上显示：

Terminate batch job(Y/N)?键入 Y，系统终止批处理作业，显示提示符。键入 N，系统只终止当前正在执行的命令并开始文件中的下一条命令的执行。

⑤ 如果含有待处理批文件的磁盘不在指定的驱动器中，PC-DOS 会提醒你让你插入这张磁盘。

AUTOEXEC.BAT 文件

AUTOEXEC.BAT 文件是一个启动系统后自动执行的文件，它必须与 PC DOS 同存于一盘。假如要在系统启动后装入 BASIC，执行 Basic 程序 MENU 的话，可创立一个含有 Basic、MENU 命令的 AUTO-EXEC。BAT 文件，具体过程如下：

创立文件　COPY CON：AUTOEXEC.BAT〈ENTER〉

写入命令　　BASIC MENU〈ENTER〉

上面的〈ENTER〉表示先按 F6 键再按 ENTER 键。

3．检查磁盘(CHKDSK)命令

格式　CHKDSK[d:]

类型　外部

作用　分析指定或缺省驱动器中磁盘的目录和文件分配表，产生错误信息及磁盘、存贮区状态报告。

状态报告的格式：(×表示数字)

```
    ××        disk file
××××××        bytes total disk space
××××××        bytes remain available
××××××        bytes total memory
××××××        bytes free
```

注意：

① CHKDSK 暂时将[d:]指定的驱动器置成缺省驱动器。假如 CHKDSK 过早结束的话(若要求它在出现一个磁盘错误后结束)，缺省驱动器就成为 CHKDSK 指定的驱动器。

② CHKDSK 不会在指定驱动器装入磁盘后再进行检查工作。在单驱动器系统中，指定驱动器应有别于缺省驱动器，除非想检查 DOS 盘。

4．比较文件(COMP)命令

格式　COMP [file spec][d:][filename[.ext]]

类型　外部

作用　以字节为单位比较第一文件与第二文件的内容。第一、第二文件的名字、所在驱动器号都应予以指定。遇到不等的字节，COMP 给出当时的 16 进制字节偏差及两不等字节的内容。

例如：Compare crror at offset × × × × × × × ×

file1 = × ×

file2 = × ×

若不等的次数超过十次，COMP 命令结束，给出如下信息：

10 Mismatches-aborting compare

若比较顺利结束(显示 Files compare OK)或比较结束 COMP 显示：

Enter primary file name

Or strike the ENTER key to end_

用户可再键入两个文件名继续比较或退出 COMP 命令。

注意：

① 两文件可同名，但必须在不同驱动器上的磁盘中。

② 若对一个文件只指定了驱动器，则该文件的名字被认为等同于另一文件的名字。

③ 文件名中使用?*字符不会得到多对文件相比的结果，COMP 只比较第一次与?、*相符合的文件。

④ 文件尺寸不等时不进行比较。

5．复制(COPY)命令

格式　COPY filespec [d:] [filename[.ext]]

类型　内部

作用　复制一份或多份 filespec 指定的文件到[d:]盘，并为之起名 filename[.ext] 或将数个文件复制在一张盘上。

注意：

?和*字符可插在源文件和复制文件的名字中。

COPY 的两种用法

① COPY filespec 或 COPY filespec d:。将文件复制到缺省或指定的驱动器盘上。文件名不改动，源、目驱动器不能相同。

实例：COPY B:MYPROG 把驱动器上的 MYPROG 文件复制到 A 驱动器盘上，文件名是 MYPROG。

COPY *. * B：A 驱动器盘上的全部文件复制到 B 驱动器盘上。

② COPY filespec filename[.ext]

或 COPY filespec d:filename[.ext]

把文件(filename)复制到缺省驱动器或指定驱动器盘上，文件名为 filename，源、目驱动器不必不相同。

实例：COPY MYPROG.ABC B:*.× × ×

注意：

保留设备名也可参加复制工作。例如：

```
COPY    CON:    filespec
COPY    CON:    AUX
COPY    CON:    LPT1:
COPY    fileA:  CON:
COPY    fileB:  AUX:
COPY    fileC:  LPT1:
COPY    AUX:    fileX
COPY    AUX:    LPT1:
COPY    AUX:    CON:
```

6．日期设置命令(DATE)

格式：DATE

类型：外部

作用：设置或更改日期。该日期存放在用户创立的全部文件的目录里。DATE 命令既可由控制台发出也可由 Batch 文件发出。DATE 命令执行后显示：

Current date is mm—dd—yy

Enter new date—

这里 mm 是 1～2 位数字，表示月份(1～12)；dd 是 1～2 位数字，表示天(1～31)；yy 或是 2 位数字 (80～99)或是 4 位 1980～2099 的数字。

注意：

① 键日期正确的话，按 ENTER。

② 日期的分隔符是—，/。

③ 只要数据不超出规定的范围，机器均认为它是今天的日期。

④ 键入错误的数据或错误分隔符后，机器显示"Invalid date"。

实例：A>DATE

Current date is 07—17—82

Enter new date 7/24/82—

7．显示目录(DIR)命令

格式：DIR [d:][filename[.ext]]

类型：内部

作用：列出全部或指定文件的目录，包括每个文件的大小 (10 进制字节数)以及文件被改动的最后日期。系统文件 IBMBIO.COM、IBMDOS.COM 和 BADTRACE 不显示。允许使用?、*字符。

DIR 命令的两种用法：

① DIR 或 DIR d: 列出缺省或 d: 指定驱动器盘上的文件目录项。

② DIR filename.ext 或 DIR d: filename.ext 列出缺省或指定驱动器盘上名为 filename 文件目录项。

8．比较磁盘(DISKCOMP)命令

格式：DISKCOMP[d:][d:]

类型：外部

作用：比较第一驱动器与第二驱动器中两盘的内容。一般在 DISKCOPY 命令后，用 DISKCOMP 检查拷贝工作是否出错。执行DISKCOMP命令前，应随便按个键。DISKCOMP 以磁道为单位进行比较，发现不相等时发出信息，指出不等的磁道号(0～39)。DISKCOMP 结束时显示：

Compare more disketles?(Y/N)_

按N结束命令，按 Y 比较在两驱动器中新插入的盘。

注意：

① 两个驱动器号省略时，进行单缺省驱动器盘比较。

② 未写第二驱动器时，缺省驱动器作第二驱动器。

③ DISKCOMP 根据内存大小，读入尽可能多的数据，减少磁盘插入次数。DISKCOMP 还会颠倒数据的读出顺序。所以对每次磁盘插入操作，DISKCOMP 能读进 2、4、6、8 或10 个磁道的数据。

④ 当比较由 COPY 命令生成的后备磁盘时，DISKCOMP 不显示 "Diskettes compare OK"。这是因为 COPY 命令压缩了文件空间，拷贝盘的物理顺序与原盘不一样。这时应用 COMP 命令进行逐个文件的比较。

⑤ 在只有一个磁盘驱动器的系统中，提示符均指驱动器 A。

9．磁盘复制(DISK COPY)命令

格式：DISKCOPY[d:][d:]

类型：外部

作用：将源驱动器盘上的内容复制到目的驱动器盘上，两驱动器同时进行单驱动器复制工作。插入盘后应按任意一个键，开始复制工作。复制完成后 CRT 显示：

COPY another?(Y/N)_

按 Y 表示在插入新盘后继续拷贝；按N结束 DISKCOMP 命令。

注意：

① 没有写明两个驱动器号时，进行缺省驱动器上的单驱动器拷贝。

② 没有写明第二驱动器号时，缺省驱动器作为目的驱动器。

③ DISKCOPY 根据内存大小读入尽可能多的数据，减少盘的插入次数，DISKCOPY 还颠倒数据的读出顺序。所以对每次盘插入操作，DISKCOPY 能读进 2、4、6、8 或 10 个磁道的数据。

④ 多次创立、删除文件，使盘的空间不按顺序分配。这种盘称为碎片盘。由于多余的磁头移动和旋转延迟，碎片盘的寻找、读、写性能将变坏。这时应用 COPY 拷贝盘上的所有文件，将目的盘恢复成非碎片盘。

例子：COPY A:*.*.B 把A驱动器上的碎片盘复制成一张非碎片盘。

⑤ DISKCOPY 后，最好用 DISKCOMP 进行检查。

⑥ 在只有一个磁盘驱动器的系统中，提示符均指驱动器 A。

10．擦除(ERASE)命令

格式 ERASE filespec

类型　内部

作用　在指定驱动器盘上，删去指定名字的文件。

注意：

① 文件名中可写有?、*字符，但系统要予以警告。

② 擦除盘上所有文件时，可键入：

ERASE [d:]*.*

③ 系统文件 IBMBIO.COM、IBMDOS.COM 和 BADTRACK 不能被擦去。

实例：A>ERASE A:myprog.1

11. 格式化(FORMAT)命令

格式：FORMAT[d:][/s]

类型：外部

作用：先把指定驱动器盘，初始成 DOS 可接受的记录格式，然后检查各个磁道，最后初始目录、文件分配表和系统装入程序。任何受 DOS 支持的磁盘，都应进行上述格式化操作。指定 /s 时，缺省驱动器盘上的三个操作系统文件依次复制到指定盘上：IBMBIO.COM，IBMDOS.COM，COMMAND.COM。

注意：

① 格式化过程要冲掉盘上写有的全部文件。

② 格式化过程中，损坏了的磁盘被分配给 BADTRACK 文件。这些磁道不能被任何文件使用。

③ 损坏磁道的道号，可通过 CHKDSK 命令的显示信息得到。未指定 /s 时，CHKDSK 报告 0 个磁盘文件。报告 1 个磁盘文件表示有损坏磁道，且这个文件就是 BADTRACK 文件。指定 /s 时，CHKDSK 会报告存在的三个文件。报告 4 个文件表示存在有损坏磁道。

实例：A>FORMAT B:/S(键入的命令)

Insert New diskette for drive B

and strike any key when Ready(准备信息)

Formatting…Format Complete(命令结束)

System transfered

Format another(Y/N)?_(问：是否还要格式化另一磁盘?)

按Y 则格式化另一张盘，按N结束 FORMAT 命令。

12. 模式(MODE)命令

格式：MODE [LPT#:][, n][, m][, T]

类型：外部

作用：为打印机或按在彩色/图形监视器转接器的显示器操作设置模式，MODE 命令对连接在 IBM 单色——并行打印机转接口上的显示器无效。未指明 m 或 n 时，表示指定外设的操作模式不变。

MODE 的两种用法：

① MODE LPT#:[, n][, m](打印机模式)

这里 # 取 1、2 或 3(打印机号)

n 取 80 或 132(每行的字符数)

m 取 6 或 8（每时的行数）

实例：MODE LPT1:,, 132, 8 把打印机置成每行打印 132 字，每时 8 行。

注意：开机时，打印机自动置成每行打印 80 个字符，每时 6 行。

② MODE[n][, m][, T]（显示器模式）

这里 n 等于 40 或 80（每行字符数）

m 取 R 或 L（显示图像左移或右移）

T 校正显示器需要测试卡

为方便显示字符的读取，显示图像可向左或右移动 1 个字符（40 列格式）或 2 个字符（80 列格式）的距离。指定 T 时，MODE 显示提示符问你屏幕图像是否还要调整。回答 Y 则命令结束，回答 N 则图像再横向移动并再一次显示上述的提示符。

13．暂停（PAUSE）命令

格式：PAUSE（remark 注释）

类型：外部

作用：PAUSE 在显示注释和"Strike a key when ready…" 后暂停。注释是不超过 128 个字节的字符串。把 PAUSE 插在 Batch 文件里能控制 Batch 文件的执行速度。执行 Batch 文件中的 PAUSE 命令时，系统停止。按 CTRL-BREAK，整个 Batch 文件的执行结束，按其他任何键 Batch 文件继续。

实例：A>PAUSE Change diskette in drive A

Strike a key when ready…_

14．注释（REM）命令

格式：REM 注释

类型：内部

作用：显示 Batch 文件中的注释。

实例：REM This is the daily checkout program

15．改名（RENAME）命令

格式：RENAME file spec filename[.ext]

类型：内部

作用：将第一参数决定的文件名改成 file-name[.ext]。参数中可有?、*字符。

实例：RENAME B:ABODE?D?B

将 B 驱动器上的 ABODE 文件改名为 ADOBE。　　RENAME B:ABODE*.XY

B 驱动器盘上的 ABODE 文件改名为 ABODE.XY。

16．系统（SYS）命令

格式：SYS d:

类型：外部

作用：将缺省驱动器盘上的操作系统文件，复制在指定驱动器盘上。复制的文件是：IBMBIO.COM 和 IBMDOS.COM。目的盘应先用 FORMAT d: /s 命令加以格式化，否则该盘上没有 DOS 文件，未开辟记录系统文件的空间。

注意：SYS 命令可使一般应用程序盘上含有 DOS。

17．时间设置 TIME 命令

格式：TIME

类型：外部

作用：设置或更改时间值，时间值可从控制台或 Batch 文件输入。TIME 命令的提示符是：

Current time is hh:mm:ss.xx

这里 hh 是 1 或 2 位数字，表示 0～23 小时。

mm是 1 或 2 位数字，表示 0～59 分钟。

ss 是 1 或 2 位数字，表 0～59 秒。

xx 是 1 或 2 位数字，表示 0～99 百分之一秒。

注意：

① 不更改时间的话只按 ENTER。

② 输入数据后再按 ENTER。修改字段后的所有字段均被置 0。

③ 在正确范围内的数字都是可接受的。

④ 小时、分钟、秒间的分隔符是“:”，秒、百分之一秒间的分隔符是“.”。

⑤ 键入错误时间值或错误分隔符后系统显示“Invalid time”。

实例：

A>TIME

Current time is 00:25:16.55

原来时间是零点 25 分 16 秒另 6.5/10。

Enter new time:13:55:00.0—

打入的时间是 13 点 55 分.秒。

18. 显示(TYPE)命令

格式：TYPE filespec

类型：内部

作用：在屏幕上显示空白(tab)字符取 8 个字符为单位的长度时格式化文件，否则显示未经格式化的文件。

注意：

① 按 CTRL—BREAK 使在显示文件的同时打印文件。

② 文本文件能以易读格式显示，其他文件如目录文件由于非字母、数字字母的存在而以不易读的格式显示。

实例：TYPE B:myprog.one

三、行编辑程序 EDLIN

EDLIN 是一个创立、修改、显示源文件(未经编译的汇编符号文件)或文本文件的程序，完成如下功能：

- 创立、写盘新源文件
- 修改旧文件并写盘修改后的文件
- 删去、编辑、显示行
- 在文本文件中寻找、删去、替换一行或数行

文本文件中的行是自动生成并显示的，但在盘文件里不存在。插入行后，所有行的行号自动增加插入的行数，删除行后所有行的行号自动减去删去的行数。

EDLIN 的调用方法：键入 EDLIN file spec。如果 file spec 指定的文件在指定驱动器盘上，文件根据内存只能被占用 75% 的原则将文件装入内存。如果整个文件都被装入了，屏幕上出现：

End-of input file

*_

这时就可编辑文件了。如果只装入了部分文件，屏幕上显示 * 号，只能编辑文件的装入部分。若想编辑未被装入内存的文件部分，应先把内存部分文件写进盘，再装入其余部分。如果 filespec 指定的文件未出现在指定的盘上，EDLIN 按 filespec 中的文件名创立一个新文件并显示：

New file

*_

编辑完成后，原文件和修改后的文件均被写入盘，原文件被加上扩展名 .BAK，修改后的文件名由 EDLIN 命令指定。带 .BAK 扩展名的文件是废文件，不能再由 EDLIN 加以编辑。必要时可用 RENAME 改名这个废文件，然后由 EDLIN 编辑。调用 EDLIN 时，原来带 .BAK 的废文件被删除，空出存放修改文件的空间。修改文件先有一个 .$$$ 的扩展名，退出 EDLIN 的 E 命令最后决定它的名字。

EDLIN 命令参数：

line　三种输入方法

① 输入表示行号在 1—65529 间的数字。若打入的数字比内存中所有行的行号值都大，则该行是现存行的最后一行。行号应用逗号或空格隔开。

② 输入句号。表示当前行

1) 指定文件最后变化的位置，

2) 不一定是上一次显示的行。

3) 行号和行之第一个字符间有 * 号

实例：10*FIRST CHARACTER OF TEXT

③ 输入#号指定当前内存最后行的下一行。

n 指定读出或写入磁盘的行数，用于写行和连接行两命令。

String 表示欲寻找、替换、删除的文本或替换其他文本的文本。

注意：

命令只用一个字符表示，命令或 String 能用大写字母、小写字母或大小写字母书写。为增加可读性，命令间和参数间可插入分隔符。相邻行号间也应加分隔符。命令以 ENTER 键启动。用 CTRL-BREAK 键终止。显示时若嫌上卷速度过快，可按 CTRL-NUMLOCK 暂停上卷。按其他任何键继续图象上卷。EDLIN 中仍能使用功能键和 DOS 编辑键。EDLIN 的提示符是星号 *。

1. 连接行命令(Append Lines)

格式：[n]A

作用：从磁盘中取出几行文件，连接到内存正被编辑的文件上，这几行加在内存文件行

的后面。若文件过长，可用A命令分批把文件调入内存，不过调入前应把内存中编辑过的行写回磁盘。

注意：

不指定行数的话，装入的行数与原有行占据75% 的内存空间。如果内存已达到75%了，则A命令不起作用。A命令把文件的最后行搬入内存时屏幕显示：End of input file。

2．删除命令(Delete lines)

格式：[line][line]D

作用：删除指定的若干行。删除行后的第一行变成当前行，当前行和其后面的所有行均被重新编号。

省略第一个参数时，Line D 命令删去当前行到 Line 指定的全部行，省略第二个参数时，line D 或 line，D 命令只删去指定的行。两个参数全部省略时，删除当前行，其后一行变为当前行。

实例：假定下面是待编辑的文本，其中第 29 行是当前行。

```
 1：  This is a sample file
 2：  used to demonstrate
 3：  line deletion
 4：  and dynamic
 5：  line number generation.
 ⋮
25：  See what happens
26：  to the lines
27：  and line numbers
28：  when lines are
29：  *deleted.
```

键入 5，25D 命令删除 5 至 25 间的全部行。结果是：

```
1：  This is a sample file
2：  used to demonstrate
3：  line deletion
4：  and dynamic
5：  *to the lines
6：  and line numbers
7：  when lines are
8：  deleted
```

其中 5～25 行被删除，26～29 行被重新定义行号 5～8。第 5 行变成当前行。

键入 6D 命令删除当前行及后一行。结果是：

```
1：  This is a sample file
2：  used to demonstrate
3：  line deletion
4：  and dynamic
```

5: *when lines are

6: deleted

5～6 行被删去，7～8 行被重新定义行号(5, 6)，当前行还是第 5 行，但内容不同了。

键入 2D 命令删去第二行。结果是：

1: This ls a sample file

2: *line deletion

3: and dynemic

4: when lines are

5: deleted

键入 D 命令删除当前行。结果是：

1: This is a sample file

2: *and dynamic

3: When lines are

4: deleted

3. 编辑行命令(Edit line)

格式：[line]

作用：显示 line 指定行的行号和内容，line 缺省时显示当前行之下一行。行号及行内容显示完毕后，CRT 换行，再一次显示行号。可用控制功能、编辑键编辑该行或在行号后重新打入新行内容。

按 ENTER 键显示行号及行内容，按 ESC 或 CTRL-BREAK 键，修改后的行不写入文本文件，原行不变。光标在行首或行尾时，按 ENTER 的效果也是如此。若光标不在行首、行尾位置，按 ENTER 将截去行在光标后的全部内容。

实例：*6

6: This is a sample unedited line.

6:

键入〈F_2〉u.光标移至 u 字母处：

*6

6: This is a sample unedited line.

6: This is a sample

键入〈DEL〉〈DEL〉〈F3〉，删除后面 2 个字符，并显示该行在删除字符后的内容：

*6

6: This is a sample unedited line.

6: This is a sample edited line.

下面可采取的动作是：

① 打 ENTER，写回修改后的行。或

② 在修改行后再打入字符即进行插入操作。或

③ 按 F5 键，编辑修改后的行，但不影响原来行。或

④ 按 ESC 或 CTRL-BREAK 键，删除修改后的行。原来行不变。

4. 结束编辑命令(End Edit)

格式：E

作用：结束编辑。修改过的文件按调 EDLIN 时指定的名字写入磁盘。如果是修改旧文件，旧文件被改名，添上。BAK 扩展名，成为废文件。创立新文件的话不建立相应的废文件。

EDLIN 返回 DOS 命令处理程序。

5．插行命令(Insert lines)

格式：[line]I

作用：在 line 指定的行前插进若干行。line 未指明时，若干行插在当前行的前面。若 line 大于现存行的所有行号，若干行将插在内存全部行的后面。

EDLIN 将显示适当数量的行，用户可利用 ENTER 键一一编辑之。按 CTRL-BREAK 退出插入命令。插入行的下一行将成为当前行，其本身和后面的全部行被重新定义行号。

注意：

创立新文件、键入文本文件前应按 I 命令。

实例：假定下面行等待编辑，当前行是第三行。

1：This is a sample file
2：used to demonstrate
3：*line deletion
4：and dynamic
5：line number generation

键入 4I 表示要在第 4 行前插入若干行。显示器显示：

4：*-

现在插入两行：

*4I
4：First new line of text
5：Second new line of text
6：〈CTRL-BREAK〉

用列表命令显示插入两行后的该文件：

1：This is a sample file
2：used to demonstrate
3：line deletion
4：First new line of text
5：Second new line of text
6：*and dynamic
7：line number generation.

如果键入的是 1I 命令即改变插入的位置，则文件变成：

1：First new line of text
2：Second new line of text
3：This is a sample file
4：used to demonstrate
5：line deletion

6: and dynamic

7: line number generation.

如果键入的是3I、.I或I命令，文件变成：

1: This is a sample file

2: used to demonstrate

3: First new line of text

4: Second new line of text

5: *line deletion

6: and dynamic

7: line number generation.

如果键入的是6I或#I命令，文件变成：

1: This is a sample file

2: used to demonstrate

3: line deletion

4: and dynamic

5: line number generation

6: First new line of text

7: Second new line of text

6．列表文件命令(list lines)

格式：[line][,line]I

作用：显示[line][,line]指定的若干行，当前行号不变。

诸如，line L形式的命令省去了第一个参数，显示从当前行前的11行处开始，一直到line指定的行结束。如果line超前当前行不止11行，按两行参数均被省略处理。诸如line L或line,L的命令省去了第二个参数，显示从line开始的23行。L命令省去了两个行参数的话，显示当前行、它的前11行、后11行，共23行。如果当前行前没有11行则以它后面的行补充，保证显示23行。

实例：假定需要编辑如下的文件，当前行在第5行。

1: This is a sample file

2: used to demonstrate

3: line deletion

4: and dynamic

5: line number generation

⋮

15: *This is the current line(note the asterisk)

⋮

25: See what happens

26: to the lines

27: and line numbers

28: when lines are

29: deleted

键入命令 5, 25L, 则显示 5 到 25 行:

5: line number generation

⋮

15: This is the current line(note the asterisk)

⋮

25: See wath happens

命令 1, 3L 显示头三行:

1: This is a sample file

2: used to demonstrate

3: line deletion

键入命令 3L, 则显示从第 3 行开始的 23 行:

3: line deletion

4: and dynamic

5: line number generation

⋮

15: *Ths is the current line(note the asterisk)

⋮

25: See what happens

键入命令 L, 则显示当前行为中心行的 23 行:

4: and dynamic

5: line number generation

⋮

15: *This is the current line(note the usterisk)

⋮

25: See what happens

26: to the lines

7. 退出编辑命令(Quit Edit)

格式: Q

作用: 在不保存前面修改文件的情况下退出编辑。EDLIN 将显示如下提示符:

Q

Abort edit(Y/N)?_

回答 Y, EDLIN 退出编辑, 改动部分不存入磁盘, 不创立。BAK 废文件; 回答 N 或不是 Y 的键, 则编辑继续。

8. 替换文本命令(Replace Text)

格式: [line][, line][?]Rstring[<F6>String]

作用: 由[line][, line] 指定行中出现第一个 String 的地方被第二个 String 替换。未指定第二 String 时, 第一 String 出现处的 String 被删除。每个被改动的行均被显示出来, 最后改动的行成为当前行。

[?] 参数表示修改后显示[O. K. ?)提示符，这时按下 ENTER 或 Y 键表示接受修改的结果，按其它字符说明不接受修改结果。在两种情况下，搜寻继续进行。

省略第一行参数表示 line = 1，省略第二行参数表示它等于内存中最后一行。省略两个行参数表示搜寻内存中的全部行。

第一 String 始于 R 后的第一个字符，终于〈F6〉或〈CTRL-Z〉或省略第二参数时的 ENTER 键。省略第一 String 的话，命令立即显示。"Not foond"。第二 String 始于〈F6〉或〈CTRL-Z〉后的第一个字符，终于 ENTER。

实例：设有文件：

1：This is a Sample file

2：used to demonstrute

3：the Replace and Search Text commands.

4：This includes the

5：optional parameter?

6：and required string

7：*Parameter。

键入命令 1，7Rand〈F6〉or〈ENTER〉将把文件中的全部 and 换成 or。结果显示：

3：The Replace or Search Text commors.

4：or required String

第 6 行成为当前行。第 1、2、4、5、7 行因未被改动不显示。

*1，7?Rand〈F6〉or〈ENTER〉命令会导致显示 O. K.? 提示符：

3：the replace or Search Text command.

O. K.?Y

3：the replace or Search Text commors.

O. K. ?N

6：or required string

O. K. ?Y

*

现在的第 3 行、第 6 行是：

3：the Replace or Search Text commands.

6：or required string

9．搜寻文本命令(Search Text)

格式：[line][，line][?]Sstring

作用：在行号参数规定的范围内搜寻与 String 匹配的内容。含有 String 的第一行被显示出来并成为当前行。没有找到匹配行时显示"Not found"，当前行不变。

[?]参数表示在找出一个匹配时显示提示符(O. K. ?)。这时按下 Y 或 ENTER 键的话，匹配行成为当前行，命令结束。按下其它键则继续搜索，直到找到新的匹配行或查遍全部该查的行为止。

省略第一行参数表示 line = 1，省略第二行参数表示它等于内存中最后一行。省略两个行参数表示搜寻内存中的所有行。

第一个 String 始于 S 后的第一字符，终于 ENTER 键。省略 String 参数，命令立即结束，显示 "Not found"。

实例：设有文件：

```
1: This is a sample file
2: used to demonstrate
3: the Search Text command
4: This includes the
5: optional parameter?
6: and required string
7: *parameter.
```

键入 1，7Sand、1，Sand 或 Sand 命令寻找文件中的"and"。结果为：

```
3: the Search Text command
*
```

尚若要寻找的不是"command"里的"and"，则按下 4，再按下 F3 键把上面 S 命令复制在屏幕上，然后再按下 ENTEK 键让 S 命令从第 4 行开始继续搜寻。

```
*4, 7 Sand
    6: and required string
*
```

现在第 6 行成为文件的当前行。也可以使用另一种方法先要求显示提示符 O. K.?，即键入 1，7？Sand 命令。

```
*1, 7?Sand
3: the Search Text command.
O. K.?N
6: and required string
o. k. ?Y
*
```

10. 写行命令(Write Lines)

格式：[n]W

作用：将内存中以第一行开始的 n 行写入磁盘。若未指明 n，写入盘的行数使得内存的占据率为 25%。即腾出内存 75% 空间以存贮从磁盘里读出的其余行。写入后内存中的剩余行被重新编号，行号从 1 开始。

四、连接程序 Linker

Linker 是一个连接各目标块的程序，完成如下功能：

① 连接数个分离目标模块。

② 搜寻文库文件以定义外部访问要求。

③ 分解外部交叉访问。

④ 生成一个可供打印的列表文件，这个文件显示出外部访问的分解结果和错误信息。

⑤ 产生可再定位的装入模块。

连接程序能处理的输入、输出和暂时文件如表 31-7、表 31-8所示。

表 31-7 输入文件

类型	.ext		生成者
	省略	跨越否	
目标	.REL	是	汇编程序
目标	.OBJ	是	汇编程序
文库	无	无	编译程序
自动响应	无	无	用户

表 31-8 输出文件

类型	.ext		使用者
	省略	跨越否	
列表	.MAP	是	用户
运行	.EXE	否	可再定位装入程序(COMMAND.COM)

1．暂时文件 VM. TMP

LINK 在定义装入模块时，将占用很大的内存空间，如果最终生成的模块内存放不下，LINK 就在缺省驱动器盘上创建名称为 VM. TMP 的暂时文件以记录模块，同时显示相应的信息。如果缺省驱动器盘中已有名为 VM. TMP 的文件，该文件将被删除，新创立的 TMP. TMP 在 LINK 完成连接工作后也被删除。

2．连接程序的命令提示符

启动连接程序后，Linker 将显示 9 个提示符，用户对这些提示符的响应(可选参数)将控

表 31-9 提示符及响应方法

提示符	响应
OBJECT MODULES	filespec[filespec…]
RUN FILE	filespec[/P]
LIST FILE	[filespec]
LIBRIAIES	filespec[filespec…]
PUBLIC	Y 或 N. Y 列出定义的全部全局符号
LINE NUMBER	Y 或 N. Y 包含进 LIST 文件中的行号
STACK SIZE	非 0，10 进制数，表示 RUN 文件中的栈大小
LOAD LOW	Y 或 N. Y 让 RUN 文件在执行前装入内存低地址部分
DSALLOCATION	Y 或 N. Y 表示在数据段的高端装入数据

制连接程序的运行路线。Linker 结束后返回 DOS。

表 31-9 列出提示符及他们的响应方法，方括号[ ]表示其中的参数可以省略。

注意：

① 对要求回答 Y 或 N 的响应，只有键入的第一个字符有效。该字符应在 Y、N 或 ENTER 三字符中选取。

② filespec 参数未指定驱动器号时，驱动器就指缺省驱动器。

③ 键入 CTRL-BREAK 可在 LINK 正常结束前结束(停止)连接过程。

3．提示符响应参数的含义

① 目标模块(Object Modules)

filespec 表示连接的目标模块名。若省略了扩展名，LINK 认为扩展名是 .OBJ 或 .REL。如果文件具有其他扩展名，这个扩展名应予以指明。filespec 应用逗号或空格分开。当指定的文件不在驱动器盘里时，LINK 显示请装入磁盘的信息。为避免与 VM. TMP 文件相冲突，模块文件可指定装在其他驱动器盘里，通过更换磁盘来调入模块。在只有一个驱动器的系统中，磁盘更换只有在 VM.TMP 文件尚未打开的情况下才能安全进行。

② 运行文件(RUN FILE)

filespec 参数用来指定连接程序生成的运行程序名和驱动器盘号。运行程序总带有扩展名，不管用户是否指定了扩展名。

假如响应参数是 filespec/P，LINK 将显示让用户插入保存运行文件的盘，该盘不应在缺省驱动器中。

③ 列表文件(LIST FILE[run-filename. MAP])

未指明被跨越时，列表文件将同运行文件一起写在磁盘上。列表文件为每个输入(目标)模块段准备一项，该项指出段在运行文件中的偏移地址。filespec 空缺时，列表文件名是运行文件名 .MAP。

④ 库文件(LIBRARIES[ ])

响应参数应是一串库文件名或 ENTER。ENTER 键表示没有库文件。库文件名的个数不能超过 8 个，名字间用逗号或空格分开。LINK 根据列出顺序搜寻库文件进行外部访问的访问。当 LINK 发现库文件名等于外部访问符号时，该库文件就作为相应的目标模块。

③ 公共(PUBLICS [n])

响应参数应是 Y、N 或 ENTER 键。键入 Y 表示 LINK 应列出输入模块中的所有全局符号、它在运行文件中的段偏差地址。

⑥ 行号(LINE NUMBER[n])

响应参数应是 Y、N 或 ENTER 键。键入 Y 表示列表文件中应包括行号和源语句在输入模块中的地址。

⑦ 堆栈大小(STACK SIZE[0 bytes])

响应的应是 0～65536 代表堆栈字节数的 10 进制数字或 ENTER 键。当数值在 0 和 512 之间时，实际堆栈占据 512 字节。

字节数用来跨越汇编程序或编译程序为待创立的装入模块准备堆栈区。0 字节数或 ENTER 键表示运行文件时的堆栈大小，等于汇编程序或编译程序的堆栈大小。运行堆栈过小将使产生的装入模块不确定。

输入的(目标)模块应至少写有一句堆栈分配语句。该语句由编译程序生成。对于汇编程序，源程序应含有一个包括 STACK 混合型的 SEGMENT 命令。若模块中没有栈分配语句，LINK 返回警告信息：NO STACK STATEMENT。

⑧ 装入低地址(LOAD LOW:[n])

响应参数应是 Y、N 或 ENTER 键。键入 Y 让 LINK 的装入程序把运行图像装在内存低地址部分，键入 N 让运行图像装在内存高地址部分，但不与 COMMAND.COM 暂存区域相冲突。COMMAND.COM 总占据内存的最高地址部分。LOAD LOW 的响应字符与 DSALLOCATION 的响应字符结合使用。对 Pascal 程序，应用 Y 响应符号。

⑨ DSALLOCATION[N]

响应符号应是 Y、N 或 ENTER 键。当存在 DGROUP 群时，该群总被放在比其他段还要低的内存地址部分。

键入 Y 响应字符，让 LINK 把 DGROUP 中定义的全部数据，装入数据段的高端部分。运行时，可访问包括群中全部数据的的数据空间可能的最低地址，是该空间的起始地址。响应字符是 N 的话，表示模块装在内存高地址空间，DGROUP 内指定分配区下的内存空间，除可作数据区域外，还可供用户应用程序动态使用。这个动态分配区域的最大值等于 64KB 减去段中分配给 DGROUP 的字节数。同时，LINK 将把全部定义在 DGROUP 群中的数据，装入数据段的低端，偏差从 0 开始。为此，数据空间指针应该定义在该数据段中。除 DGROUP 群外的其他 GROUP 中的段，同理，被装在群的低端地址部分。

4．特殊命令字符

LINK 能识别出下列两个特殊命令字符。

①&

若响应参数非常长，一个显示行容纳不下的话，用 & 号连接分行打入的响应参数，& 号在显示行的最后键入。按 ENTER 键表示整个响应参数结束了。

②;

ENTER 后键入；表示第一第二提示符后的其余提示符均取缺省响应参数。不可用感叹号！跳过一些需缺省的响应参数。

5．LINK 程序的两种调用方法

方法 1：控制台响应法

从键盘键入 LINK 和一连串响应参数。响应参数有错的话，应按 CTRL-BREAK 键退出 LINK 程序，再键入 LINK 重新进入连接程序。响应参数未进内存时，可用编辑命令、键修改。

方法 2：自动响应法

从键盘键入 LINK filespec。filespec 指明了自动提供响应参数的文件。该文件应在 LINK 被调用前写入磁盘，文件里的各响应参数的顺序应与提示符一致，过长的响应参数可分布在几行里，中间用 & 号连接。提供响应参数文件的扩展名可有可无，但没有缺省扩展名(空缺时系统指定个什么名字)。

响应参数文件的行里可有三角括号<>，它的行用是：

① 三角括号里的字符串作注释用，显示在屏幕上。

② LINK 显示完注解后暂停，等待从键盘输入响应参数，ENTER 键或单单 ENTER 键。

LINK 然后再继续连接工作。

实例：自动提供响应参数文件

① MODA…MODF 数个标准模块的名字。

② 〈your module?〉提示字符串，问是否还有其它目标模块。

③ 十个提示符的响应参数

```
MODA, MODB, MODC &
MODD, MODE, MODF &
〈YOUR MODULE?〉
RUNFILE/P
PRINTOUT
PASCAL. LIB
Y
Y
O
N
Y
```

6．段、群、类概念的定义

段、群、类是常在出错信息出现的三个述语，了解他们除能帮助看懂错误信息外，还能掌握 LINK 程序的基本工作方法。

段(segment)　段是长度不超过 64KB 字节的连续内存区域。每个段的起始单元的低四位地址总为 0 。段首地址由段寄存器提供，段内地址由 16 位偏移寄存器提供，所以访问一个段内单元要用段/段偏移两个寄存器。段段可相互覆盖。由于装入程序的存在，段中机器指令的物理地址都取偏移地址，运行时的实际物理地址由 COMMAND. COM 中的定位装入程序根据 LOAD. LOW 的响应参数产生。

组(group)　组是若干段的集合，存在一个 64KB 字节的段中。汇编程序或编译程序将这些段的名字定为组。一个程序可拥有一个或数个组。定义群的目的是为了用一个段基寄存器和一个段偏移寄存器访问属于一个组的数个段。连接程序将检查组里的目标模块是否违反了不超过 64KB 的规定。

类(class)　类也是若干段的集合，但为属于类的段起的类名将影响这些段在内存中的存贮顺序和相对位置。属于类的若干段是连续存放的。

类中段的顺序，由连接程序在目标文件中遇到这些段的先后决定。如果属于一个类的段比属于另一个类的所有段都先在输入文件里被 LINK 碰到，则第一类的顺序在第二类前。类的大小没有限制。类也可分成数个组，方便寻址。

7．调试程序 DEBUG

DEBUG 提供可控制的测试环境，使用户能够临督和控制执行需要的程序。DEBUG 还有装入、修改、显示文件的能力和执行目标文件的能力。DEBUG 由键入 DEBUG[filespec] 进入。如果打入 filespec，则 DEBUG 将指定的文件装入内存。然后用户可以进入命令：修改、显示或执行指定的文件。

当启动 DEBUG 程序时，各寄存器和标志位的内容要重新予以指定。

• 段寄存器(CS,DS,ES和SS)置到空闲内存的底部,也就是DEBUG程序端的第一端。

• IP 置为 X'0100'

• SP 置到段末端或 COMMAND. COM 暂存区的底部

• 其余寄存器(AX、BX、CX、DX、BP、SI 和 DI)均置为 0 。如果指定了 DEBUG 以及 filespec,则 CX 寄存器包含文件的长度(字节数)

• 清除标志位

• 缺省磁盘传送地址置到代码段 X'80' 单元。

下面是 DEBUG、LINK、DOS 命令一览表。

表 31-10 DEBUG 命令一览表

序 号	命 令	用 途	格 式
1	DUMP	显示内存	D【地址】或 D【范围】
2	Enter	改变内存	E 地址【表】
3	Fill	改变内存块	F 范围表
4	GO	从断点处开始执行	G【=地址】【地址【地址…】】
5	Hexarit-hmetic	16 进制加减	H值值
6	Input	读/显示输入字节	I 口地址
7	Load	装入文件或绝对磁盘扇区	L【地址【驱动器扇区扇区】】
8	Move	移动内存块	M 范围 地址
9	Name	定义文件和参数	N filespec【filespec…】
10	Output	发送输出字节	O 口地址 字节
11	Quit	退出 DEBUG 程序	Q
12	Register	显示寄存器/状态	R【寄存器名】
13	Search	寻找字符	S范围 表
14	Trace	执行和显示	T【=地址】【值】
15	Unassemble	未汇编的指令	U【地址】或 U【范围】
16	Write	写文件或绝对磁盘扇区	W【地址】驱动器扇区扇区】】

表 31-11 LINK 命令一览表

序　号	提　示	回　答
1	OBJECT.MODULES:	filespec[filespec…]
2	RUN FILE:	filespec[/P]
3	LIST FILE:	[filespec]
4	LIBRARIES:	filespec[filespec]
5	PUBLICS:	Y 或 N,Y 列出所有全程符号及其定义
6	LINE NUMBERS:	Y 或N,Y 行号也在文件 LIST 中
7	STACK SIZE	非零,10 进制数建立 RUN 文件栈大小
8	LOAD LOW:	Y 或 N,Y 引起 RUN 文件在执行时装在低阶内存区
9	DSALLOCATION:	Y 或 N,Y 把数据装入数据段的高端

表 31-12 DOS 命令一览表

命　令	类 型	用　途	格　式
(Batch)	I	调用批文件	[d:]文件名[参数]
CHKDSK	E	检查磁盘、报告状态	CHKDSK[d:]
COMP	E	比较文件	COMP[filespec][d:][文件名[.扩展名]]
COPY	I	复制文件	COPM filespec[d:][文件名[.扩展名]]
DATE	E	进入日期	DATE
DIR	I	列文件名	DIR[d:][文件名[.扩展名]]
DISKCOMP	E	比较磁盘	DISKCOMP[d:][d:]
DISKCOPY	E	复制磁盘	DISKCOPY[d:][d]
ERASE	I	删除文件	ERASE filespec
FORMAT	E	磁盘格式化	FORMAT[d:][/S]
MODE	E	设置打印机/显示器方式	MODE[LPT*:][,n][,m][,T]
PAUSE	I	系统等待	PAUSE [注释]
REM	I	显示注释	REM [注释]
RENAME	I	重新命名文件	RENAME filespec [文件名[.扩展名]]
SYS	E	传送 DOS	SYS d:
TIME	E	进入时间	TIME
TYPE	I	显示文件内容	TYPE filespec

五、PC-DOS 的优点和限制

我们准备从评价 CP/M-86 的技术特征和终端用户这两个角度来考察 PC-DOS.

从系统程序立足点看,PC-DOS 提供了行编辑程序(不是字处理程序),汇编语言排错程序和汇编程序的连接程序,但缺少一个相当重要的 8086 汇编程序,排错程序 DEBUG 中未安排一条对创立短小汇编语言程序或汇编语言排错程序这一极有用的 ASSEMBLE 汇编程序.

一般说来,Microsoft 公司版本的 Pascal,FORTRAN-77,BASIC 语言均能在 PC-DOS 支持的 PC 上使用.其中 BASIC 语言发展最快,已有解释程序 BASIC-86 和编译程序 BASCOM-86. BASIC-86 是为 8086 系统准备的一个 BASIC 解释程序版本,它融合了在不同微机上工作的各种 BASIC 语言的长处,是一个非常流行的语言.如果 BASIC-86 是被 PC-DOS 1.1 版本支持的话,由于 DOS 1.1 版本具有串行打印机、游戏装置的控制能力,BASIC 语言就能提供 VARPTR$ 功能. BASCOM-86 编译程序是一个把根据为 BASIC-86 写的 BASIC 程序翻译成能高速执行的 86 机器指令的程序.同时 BASCOM-86 还检查 BASIC 程序,给出错误信息.

PC-DOS 的另一个优点是错误信息简明易懂,用英文句子的形式给出,能指出错误出自何处以及错误的原因.譬如磁盘出错,程序员能用事务处理命令停止磁盘操作,重头开始这个操作.此外,PC-DOS 文件里还有大量的例子帮助用户掌握 PC-DOS.

§31.3 CP/M-86 与 PC-DOS 的比较

CP/M-86、PC-DOS 是现今以 8086/8088 为 CPU 的微机系统的两个流行操作系统,他们互不兼容,代表着 8 位 CPU 机操作系统向 16 位 CPU 机操作系统发展的两个分枝,所以有必要对这两个操作系统进行比较、分析.比较分析将以两个角度展开,一是两者对自己在上面运行的硬件资源的支持程度即向下支撑能力;二是向用户特别是终端用户提供的使用软盘文件系统的方便程度、高级语言、实用程序、开发软件的种类即向上的支撑能力.

一、向下支撑能力

1. 存贮管理

CP/M-86 的存贮区管理较之 PC-DOS 的先进.CP/M-86 拥有一套分配、回收动态存贮区(RAM 区)的系统调用,系统从自由内存空间池为用户分配存贮区,有些存贮区还可按绝对地址的要求予以安排.系统对一些存贮映像设备提供保护性控制,不让多个用户访问这些不可共享的设备.另外,在不超过最大段尺寸的条件下内存能按需供给.在 CP/M-86 内存管理支持下,程序的实际内存地址不一定是连续的.对于磁盘上的每个可执行命令文件,CP/M-86 都在其首部加上一记录文件执行时需最大、最小空间的信息头,所以从磁盘调入的程序均能保持完整,运行时也不会受其他内存程序段的干扰或破坏.最后 CP/M-86BIOS 层的系统存贮区段表,还能向用户系统提交整个内存区被占据和尚空白、大小等状态信息.

PC-DOS 的内存管理方法类似于 CP/M-80.在程序装入后,PC-DOS 在程序段的低地址的若干单元,还记下了当前数据段的最大地址和所有可用空间的最大地址.被装入的程序,可访问从其基地址到内存最高地址整个区域并可随时调节.被 PC-DOS 支持的系统的

内存空间必须是连续的。PC-DOS 内存第 6 号单元设置内存可用段数，指明“EXE”可执行文件装入的内存段的端边界地址，提供一些单元对边界内存管理的基本功能。PC-DOS 不具备动态内存管理功能。

2．磁盘 I/O 性能评价

① 磁盘 I/O 的缓冲能力的优劣直接影响系统性能。缓冲区大，磁盘执行文件和数据文件就能在次数不多的磁盘读取操作后装入内存，降低磁头、盘面的磨损。但是较大的缓冲区将减少用户可用内存区，使本来就不富裕的微机内存区更显得紧张。缓冲区小则适合软盘上的较小文件的读取与写入，对节省系统内存开销有利，但小缓冲区会直接影响文件的调进调出速度。从磁盘缓冲区的设置情况看，PC-DOS 优于 CP/M-86。

PC-DOS 将磁盘缓冲区设在操作系统区。它的成块折块程序以块为单位将数个块读出或写入磁盘（通过缓冲区）。所以 PC-DOS 的写盘驱动程序较简单，汇编语言程序的写盘很容易。

有些 CP/M 系统，如 Lomas 数据产品，在其 BIOS 层配备了全磁道缓冲区，使 CP/M-86 的文件调入速度接近 PC-DOS。下面是在有缓冲、无缓冲两种情况下 CP/M-80 汇编 CBIOS 的时间比较。

表 31-13　缓冲区配备对汇编速度影响的比较

操作系统	缓　冲　区	磁　盘　机	汇　编　时　间
CP/M-86	无扇区缓冲	单　密　度	3 分 30 秒
CP/M-86	有磁道缓存	单　密　度	2 分 30 秒
CP/M-86	有磁道缓存	双　密　度	2 分 20 秒

② CP/M-86 将文件分配表存于磁盘，而 PC-DOS 将类似的表格常数驻内存。下面是 PC-DOS、CP/M-86 在相同硬件环境下运算和磁盘 I/O 性能的五张比较表。表 31—14 是 BASIC 程序在 PC-DOS 及 CP/M-86 支持下的 BASIC-86 上运行的比较。由于 CP/M-86 支持的 BASIC-86 有两个版本，所以结果不一样。从表 31-14 可看出 PC-DOS 写盘操作较 CP/M-86 占较大的优势。表 31-15 是 PC-DOS 及 CP/M-86 环境下（SystemX 机）装入 BASIC-86 所需时间的比较。表 31-16 是 PC-DOS 及 CP/M-86 环境下（IBM PC 机）复制文件和装入文件的时间比较。表 31-17 是 CP/M-86 与在 PC-DOS 上的模拟 CP/M-86 程序 EM-86 的运行性能比较。程序由 Microsoft 公司的 BASIC 语言编写，运行机器是主频 5MHZ 的 Compupro 8085/8088，每次 I/O 有一等待状态。第 1 至第 5 行是不涉及 I/O 的统计值，第 6、7 行是有关 I/O 的统计值。可以看出模拟 CP/M-86 的 EM-86 程序的磁盘读取速度与 CP/M-86 相差不大，而磁盘写入速度比 CP/M-86 要快得多。表 31-18 比较了 CP/M-86、PC-DOS 和 EM-86 复制文件的时间

3．I/O 设备支持能力

CP/M-86 能支持串行或并行打印机，它的 IOBYTE 文件允许用户选择不同的外设。PC-DOS 只支持并行类型的打印机，不提供软件选择 I/O 设备功能。PC-DOS 对文件大小

表 31-14

	System 机 CP/M-86 1.0 BASIC-86 Rev 5.21	Sistem 机 CP/M-86 1.0 BASIC-86 Rev 5.22	System 机 基本 MS-DOS BASIC-86 Rev 5.21
空循环	7.2	7.3	7.8
子程序转跳	15.5	15.4	16.9
MID$(子串处理)	23.6	23.6	24.6
求质数程序	199.2	189.2	197.0
写盘程序(64KB 文件)	60.8	60.8	50.3
读盘程序(64KB 文件)	20.6	20.6	21.3
除	25.0	20.6	21.8

表 31-15

	System 机 CP/M-86 1.0	System 机基本 MS-DOS
装入 30KB 的BASIC-86	5.7	2.7

表 31-16

	IBM PC 机 Compuview CP/M-86 PIP 命令	IBM- MS.DOS 1.1COPY 命令
复制三文件，长度分别为13KB、13KB、6KB	36.9	19.3
用 VEDIT 装入 30KB 文件	10.1	18.8

表 31-17

	CP/M-86 1.0 BASIC-86 Rev.5.21	MS-DOS 模拟 CP/M-86 (EM-86)BASIC-86 Rev.5.21
空循环	6.04	6.04
除	22.12	22.12
子程序跳转	11.99	11.99
MID $(子串)	20.94	20.94
质数程序(200 个质数)	47.97	47.97
写磁盘程序(文件长 64K)	18.3	15
读磁盘程序(文件长 64K)	9.1	10.1

表 31-18

	CP/M-86 1.0 PIP 命令	MS-DOS COPY 命令	EM-86 PIP 命令
复制 22KB 文件	9.9	5.75	16.15

不作任何限制，任何容量的磁盘机都可接在 PP-DOS 支持的微机上。CP/M-86 单用户版本规定系统外接的磁盘机容量不得超过 8 兆字节，这就减少了超容量磁盘机进入微机系统的可能性，除非使用特别的软件技术分割。CP/M-86 不允许大磁盘空间，从而形成数个 8 兆字节大小的逻辑磁盘机。

二、向上支撑能力

向上支撑能力主要指操作系统用户接口的使用特性，如实用程序、系统调用、高级语言的多寡和优劣。

1 接口使用难易性

CP/M-86 具有一个很大的 HELP 程序。当用户在 CRT 前感到困惑，不知下一步该怎么办时，可键入 HELP 和命令名，CP/M-86 就立即给出该命令的使用指示。如果在磁盘空间上为 HELP 文件留下 22KB 区域的话，CP/M-86 用起来就比 PC-DOS 好多了。CP/M-86 的新用户指南也比 CP/M-80 的要齐全。只要把 CP/M-86 用户指南、CP/M-86 程序员指南和 CP/M-86 系统指南与 HELP 文件结合起来使用，CP/M-86 对新用户来说是很"友好"的，不难掌握。由于 PC-DOS 拥用错误恢复功能，所以它就不怎么需要类似于 CP/M-86 终端 HELP 的文件了。

2．实用程序

下面将两操作系统功能相近的实用程序(Utility)作比较，PC-DOS 实用程序名在前，CP/M-86 的在后。

① COPY 和 PIP

PC-DOS 的文件复制程序 COPY 是常驻内存的，而 CP/M-86 在复制前要将外设交换程序 PIP 和待复制文件的位置信息读入内存，所以 COPY 的执行速度比 PIP 要快。另一方面，PIP 的功能不仅仅是复制文件。根据 PIP 命令后跟的不同字符开关，PIP 程序能完成在 CRT 上显示传送的字符、滤除文件中的格式符、核查数据的 Intel 16 进制格式、大小写字母的转换、在目标文件中加行号、为打印机输出文件定页长，在文件中找到某个串后开始或停止复制，在文件中加空格符，验明复制文件的正确性，为每个传送字节的第八位置标记等功能。这些功能均是为汇编语言程序员准备的，对一般用户无太大用处。

② CHKDSK、DIR 和 STAT、DIR

CP/M-86 的 STAT 是报告磁盘内容及状态信息，允许用户更改外设的程序，它能给出磁盘文件的详细情况。PC-DOS 的 CHKDSK 只给出磁盘、内存 RAM 区状态的大概情况。STAT 和 CHKDSK 均驻在磁盘上，运行速度相近。

CP/M-86 和 PC-DOS 的同名文件目录显示程序 DIR，均驻在内存系统区。CP/M-86 的 DIR 列出文件名；PC-DOS 的 DIR 列出文件大小，最后写入磁盘数据、文件名。两个 DIR 的运行速度都较快。

③ DEBUG 和 DDT-86

DEBUG 程序和 DDT-86 程序大致相同，前者无汇编命令，后者有汇编命令。PC-DOS 程序员不能很快地利用 DEBUG，写出检查出错原因，修补错误语言的汇编程序。

④ 错误陷井

PC-DOS 允许应用程序捕捉磁盘错误，在捕捉到错误后调用适当的子程序使应用程序继续运行或体面地结束。CP/M-86 无错误捕捉功能，应用程序可能因磁盘错误提前结束。

⑤ 格式化程序 FORMAT

用 PC-DOS 格式化磁盘时，用户能指明被格式化磁盘有"可引导性"。

3．系统调用

下面逐一比较 CP/M-86 和 PC-DOS 对 BIOS 层或相当于 BIOS 层系统调用。由于 CP/M-86、PC-DOS 与 CP/M-80 有某些相似之处，所以这里也把 CP/M-80 列了进去。存放调用前的参数，调用后返回结果寄存器的名称比较，请见表 31-19。

表 31-19 系统调用寄存器使用对照

	CP/M-80	CP/M-86	PC-DOS
系统调用码	C	CL	CL
参数	E 或 D.E 对	DL 或 DX	DL 或 DX
调用方法	CALL5	INT224	CALL5 或 INT33，AL中是调用码
返回结果(单字节) (多字节)	A HL 对	AL 或 ES BX	AL BX

注意由于 ES 寄存器可作替换段基或自动段基寄存器，当用 ES 作结果返回寄存器时，在下述情况下可能会引起麻烦：CP/M-86 调入了一个可执行文件并在 ES 中设置好了内容，若该可执行文件在执行过程中发出 CP/M-86 BIOS 调用，调用结果又在 ES 中返回的话，如果不在调用前保存 ES 内容，调用后恢复 ES 内容，程序就可能非正常运行下去。

下面以 CP/M 的调用序号为序，逐一比较两类操作系统的系统调用。

1) 调用 0 (Function 0)——系统复位(system reset)

CP/M-80、PC-DOS 对该调用的响应相同，CP/M-86 要求用户在 DL 寄存器中指定调用失败后的处理码。0 表示结束后返回 CCP 层，1 表示程序仍保留在内存，内存的分配状况不变。

2) 调用 1(Function1)——带输出的控制台输入

CP/M-80 及 CP/M-86 扩展了标记符，输入 Control-S 表示图象卷动控制，Control-P 表示在打印机上输出。对 PC-DOS，按下 Control-Break 表示控制图象卷动。

3) 调用 2——控制台输出 (Console Output)

同调用 1。

4) 调用 3——卡片机输入(Reader Input)

在 IOBYTE 支持下，CP/M-80、86 能重新指派 I/O 设备，PC-DOS 只支持一种 I/O 设备，

无 I/O 转向功能。

5) 调用 4——穿孔机输出(Punch Output)

同调用 3。

6) 调用 5——列表输出(List Output)

同调用 3。

7) 调用 6——控制台直接 I/O(Direct Console I/O)

CP/M-80、PC-DOS 对该调用的响应是一样的。如果传递的参数是 0FFH，先进行控制台状态检查。若未输入字符时调用返回 0，输入了字符则返回这个字符。若传递了非 0FFH 参数，调用均认为它是合法的 ASCII 码，系统将它打印出来。CP/M-86 把 0FFH 看成请求状态检查的参数，把 0FFH 看成让系统等待字符输入并将其返回给用户的参数，其余值的参数作为待打印字符的输出码。CP/M-80、CP/M-86、PC-DOS 均不检查特殊字符。

8) 调用 7——取 I/O 字符(Get IOBYTE)

CP/M-80、86 能支持字符的输入或输出，工作方式也相近。PC-DOS 处理的方法同调用 1 且只输入不显示。

9) 调用 8——设置 I/O 字符(Set IOBYTE)

同调用 7。

10) 调用 9——打印串(Print String)

CP/M-80、86、PC-DOS 处理方式相同。

11) 调用 10——读控制台缓冲区(Read Console Buffer)

CP/M-80、86 利用一般 CP/M 编辑功能处理该调用。PC-DOS 除允许编辑字符缓冲区内容外，还另有一套便于命令编辑的标准方法。PC-DOS 在缓冲区最后字符后加上回车符，但字符计数器不将其算入。PC-DOS 的这种处理方法与现有的软件兼容。

12) 调用 11——取控制台状态(Get Console Status)

CP/M-80、PC-DOS 发现有字符时返回 0FFH，无字符时返回 0。CP/M-86 发现有字符时返回 1，无字符时返回 0。

13) 调用 12——返回版本名(Return Version Number)

CP/M-80、86 返回操作系统版本名，PC-DOS 把该调用处理为清键盘缓冲区，进入输入调用。存于 AL 寄存器的输入调用码，可是 1、6、7、8 或 10。以上几个参数的调用，均使系统等待用户从键盘输入新字符。

14) 调用 13——磁盘系统复位(Reset Disk System)

CP/M-80、86、PC-DOS 的处理方法相同，他们均把驱动器 A 作为缺省驱动器。

15) 调用 14——驱动器选择(Select Disk)

CP/M-80、86 的处理方法相同，他们把 DL 寄存器指出的驱动器命名为缺省驱动器。DL 内容为 0 时指定驱动A，1 时指定驱动器B……。PC-DOS 使用 DL 寄存器的方式同 CP/M，并能在 AL 寄存器返回可用驱动器的个数。

16) 调用 15——打开文件(Open file)

CP/M-80、86 处理方式相同，他们在寄存器 A 或 AL 中返回 0、1、2 或 3 表示打开文件成功，或 0FFH 表示打开文件失败。PC-DOS 用返回 0 表示打开文件成功，0FFH 表示失败。另外如果调用时要求改动缺省驱动器的话，PC-DOS 将变动文件控制块中的磁盘机指定名。

所以，打开文件控制块后，就能替换缺省驱动器。

17) 调用 16——关闭文件(Close file)

CP/M-80、86 用返回 0、1、2、3 表示操作成功，0FFH 表示失败。PC-DOS 用返回 0 表示操作成功，0FFH 表示失败。

18) 调用 17——寻找第一项(Search for first)

CP/M-80、86 用 AL 寄存器返回的 0、1、2、3 表示待找的文件存在，用 0FFH 表示寻找失败。PC-DOS 用返回 0 表示寻找成功，0FF 表示失败。

19) 调用 18——寻找下一项(Search For Next)

同调用 17。

20) 调用 19——删除文件(Delete file)

CP/M-80、86 用返回0、1、2、3 表示删除成功，0FFH 表示失败。PC-DOS 用返回 0 表示成功，0FF 表示失败。

21) 调用 20——顺序读(Read Sequential)

CP/M-80、86、PC-DOS 处理方法相同。

22) 调用 21——顺序写(Write Sequential)

CP/M-80、86、PC-DOS 处理方法相同。

23) 调用 22——建立文件(make file)

同调用 19。

24) 调用 23——文件改名(Rename file)

同调用 20。

25) 调用 24——返回 Login 向量(Return Login Vector)

CP/M-80、86 处理方法相同，PC-DOS 未用这个调用。

26) 调用 25——返回当前磁盘(Return Current Disk)

同调用 20。

27) 调用 26——设置 DMA 地址

CP/M-80、86 处理方法相同。PC-DOS 调用处理段的段基地址寄存器是 DS。

28) 调用 27——取分配向量的地址

CP/M-86 在 BX 返回分配向量的偏差地址，在 ES 返回段基地址。PC-DOS 在 DS 返回段基地址，在 DX 返回分配单元数。在 AL 返回每个单元的记录数，在 CX 返回物理扇区大小。

PC-DOS 的分配向量格式不同于 CP/M 的格式。

29) 调用 28——写保护磁盘(Write Protect Disk)

CP/M-80、86 处理方法相同，PC-DOS 未用该调用。

30) 调用 29——取只读向量

同调用 28。

32) 调用 30——设置文件属性

同调用 28

33) 调用 31——取磁盘参数块地址

CP/M-86 在 BX 返回当前选中驱动器常驻 BIOS 的参数块 DPB 的偏差地址，DPB 的

段基地址在 ES 返回。PC-DOS 不支持该调用。

33) 调用 32——设置/取用户码

同调用 24。

34) 调用 33——读随机块

CP/M-80、86 的成功返回码为 0，PC-DOS 的失败返回码不等于 CP/M 的失败返回码。

35) 调用 34——写随机块

同调用 33。

36) 调用 35——计算文件的记录个数

CP/M-80、86 无错误返回码，PC-DOS 的成功返回码是 0，失败返回码是 0FFH。

37) 调用 36——设置随机记录

同调用 35。

4. 高级语言

CP/M-86 支持 CBASIC86、Pascal/MT、Pascal M86、ANSI COBOL74、ANSI FORTRAN 77 等高级语言。PC-DOS 支持 Microsoft 公司的 Pascal、FORTRAN 77、COBOL、BASIC-86 等高级语言。

5. 两操作系统的发展方向

较 CP/M-86 高级且与 CP/M-86 基本兼容的操作系统是并发 CP/M-86。这是一个单用户、多任务操作系统，具有：扩展的错误处理及报告、为文件写上时间日期、文件口令保护、系列兼容、实时处理等能力。比并发 CP/M-86 再高一级的是 MP/M，这是多用户多任务操作系统。

MS-DOS (PC-DOS) 也有它的改进版本：MS-DOS 版本 2.0 (MS-DOS Ver.2)。MS-DOS Ver 2 新加了一个用户—系统接口层，用户可通过移动光标根据 CRT 上的名字选择文件、实用程序或应用程序。用 ANSI 标准换码序列与控制台交换信息。MS-DOS Ver 2 提供 AT&T 图形文本表现层协议，使系统可与其他电缆网络和数据库系统相连。MS-DOS Ver2 的网络分划系统，可使 MS-DOS 系统联入 MS-DOS 地方网或 MS-DOS 与 Xenix 系统的混合网。目前 Microsoft 公司正在开发 Vnix 的同类版本 Xenix，他比 MS-DOS 更为高级。

总之 MS-DOS (PC-DOS) 在磁盘 I/O 速度、磁盘空间使用效率、错误恢复方面优于 CP/M-86，适合不怎么了解计算机系统的用户；CP/M-80 则较 MS-DOS 在终端帮助 (Online help)、外设重组、内存管理方面占优势，两者的向上操作系统均已开发或正在开发。

CP/M-86、MS-DOS 间的技术性竞争，给微机操作系统领域带来了剧烈的振动，促使其向前发展。CP/M-86、MS-DOS 能在 16 位微机操作系统领域站住脚的时间取决于这两个系统的改进程度，因为 Unix 已开始进入微机领域，提供 CP/M-86、MS-DOS 不具备的对终端用户极有用的功能。

§31.4 P-system 和 UCSD P-system

一、P-systsem 概述

圣地亚哥加利福尼亚大学的 P-system，是 IBM 个人计算机的另一个操作系统，P-system 因其用 Pascal 语言编写和可移植性特点，深得软件产业界的推崇。

P-system 是由在 UCSD 教授程序设计入门课程的 Kenneth Bowles 发起写成的。早在 1974 年年底，一些研究生和本科生就开始在微机上实现 Pascal 程序设计语言。选择 Pascal 的目的，是因为 Pascal 是一个结构语言，掌握 Pascal 语言后能较快地学会其它结构语言：FORTRAN、COBOL、PL/M。考虑到微机的内存容量有限及其他因素，他们选中了解释性操作系统方案，即该操作系统将该操作系统的每条命令翻译成适当的机器语言(CPU)命令。整个UCSD操作系统几乎全用 Pascal 语言写成。这种不大的操作系统和它的编译程序，可在内存容量较小的微型机上运行，并能方便地移植到其他机器上。

P-system 的字母 P(Pseudo)是伪码的意思。P-代码是 Pascal 编译程序根据"理想"CPU 机器语言指令生成的伪码。伪码在 P-机器上执行。这里的 P-机器可以是实际的 CPU 或模拟 P-机器的解释性程序，具有 P-机器 CPU 或P-机器解释程序的系统，可以自行运行也可以运行其他系统生成的 P-代码程序。

Pascal 及可移植性，是 P-system 的两大特点。Pascal 特点的含义，是 Pascal 不但是操作系统的开发语言，而且还是其他应用语言如 FORTRAN、BASIC 的编译或解释语言。用高级语言写成的操作系统，扩充了这种高级语言的能力。高级语言和操作系统的密切关系，使应用程序与操作系统紧密榫合。在一般情况下，高级程序设计语言，是围绕操作系统开发的，对于 P-system，可以说是操作系统利用高级语言来开发。

可移植性是 P-system 的第二个特点。譬如 PDP-11 机器和具有 UCSD P-system 操作系统的 Apple 或 PC 计算机能生成相同的 P-代码。只要存贮介质特别是磁盘机兼容，为一种 UCSD 机产生的 P-代码程序无需修改，就能在其他 P-system 机器上运行。可移植性是软件产业界欢迎 P-system 的一个原因。因为只要编写能在 P-机器上运行的程序，同时考虑到存贮介质的要求，就不需要有很多的程序版本。

能运行 P-system 核心程序的机器，必须具备 P-机器 CPU 或 P-机器模拟程序。IBM 个人计算机中没有 Westen Digital™ 公司的 P-机器 CPU 芯片，故只能用 P-机器模拟程序。由于操作系统的命令先被翻成 P-代码，P-机器模拟程序再把这些 P-代码翻成机器指令。所以对同一个程序，因使用的语言不同(Pascal 或机器语言)，运行速度大不一样。P-机器模拟程序的介入，将降低计算机系统的性能。估计 P-机器模拟程序的运行速度比纯粹 8086 指令程序慢 10 到 40 倍。如果一个 8086 机器指令程序需 20 秒完成的话，P-system 程序则要花 200 到 800 秒。

P-系统性能的降低和在 P-system 上运行的程序的结构有关。一方面，如果一个程序(如字处理程序)的输入、输出数据量很大，需多次使用键盘、磁盘机、打印机，因模拟程序翻译P-代码的数量及生成的 8086 指令运行次数等因素，实际产生的性能下降是很小的，运行时间与机器指令程序的相差就不大。另一方面，如果一个程序的数据输入输出量小，CPU 运算、访内次数多，系统性能下降得就厉害。在这里 P-机器指令被处理两次是一个不能低估的原因。对于以 8088 为 CPU 的个人计算机，它的系统主频是5MHZ，平均处理速度大约为0.65MIPS (每秒百万条指令)，若从一般操作员的角度看，P-system 引起的系统总性能的明显下降又是微不足道的。

研究 P-system 的另一个出发点，是其可移植性和系统性能间的平衡。企图为产品找到理想市场的软件产业界，认为 P-system 是实现这种理想市场的工具，而注意系统性能、处理速度的计算机专业人员对 P-system 则另有看法，认为因可移植性而损失处理速度是不

合算的。一般用户的观点介于两者之间，对移植性和系统性能没有过分的要求。

二、P-system 的结构

P-system 分成P-机器模拟程序，及用 UCSD PASCAL 语言写成的操作系统程序两大部分。P-机器模拟程序，又可以分为指令解释程序和 BIOS。余下的 Pascal 操作系统程序由多个子块(Unit)组成，每个子块是一段 Pascal 程序，完成一两个基本功能，另外还有栈区、堆区、和代码池。见图 31-4。

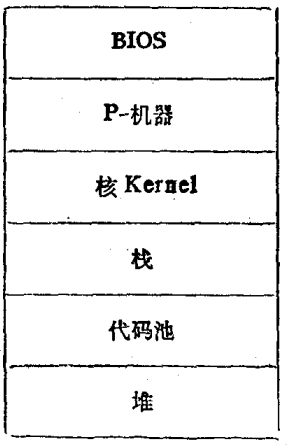


图 31-4 UCSD P-system 内存映像

1. P-机器模拟程序

P-机器模拟程序指全部涉及宿主 PC-DOS 指令和硬件的程序。使操作系统具有可移植性的一般方法是把处理打印机、磁盘驱动器、调制解调器等外设的程序集于一区，在PC 中就是 BIOS。P-system 把这概念向前发展了一步。在 P-system 中，只要 P-机器模拟程序随机器变动，P-system 就是可移植的操作系统，所以可变动的 PC BIOS 属于 P-机器模拟程序，集中在内存区的一端。P-机器模拟程序控制外设的方法与其他操作系统是一样的。

2. SYSTEM.DASCAL

这是 P-system 的第二个主要部分，也是 P-system 的核心。它由 21 个子程序块（简称子块)组成。每个子块完成特定的功能。由于 P-system 工作时一些子块总是常驻在内存，所以 SYSTEM.PASCAL 中的主要子块也称为核(KERNEL)。21 个子块的名称及作用见下表。

表中的 REALOPS 子块和 LONGOPS 子块分别是整数及浮点数处理程序，这两个子块是支持在 P-system 上运行的高级语言程序，所以 P-system 不仅仅是高级语言的载体，而且还是程序语言的一种扩充。

UCSD P-system 还具备大型机中的换页或类拟于段变换的能力。段交换功能使得超过机器内存的大程序一段一段地从磁盘调入内存运行，运行完的部分调出内存，腾出其占用的空间。P-system 至少要求有 64KB RAM 的内存，但 128KB RAM 使用起来更为方便。

3. 堆、栈和代码池区

堆、栈、代码池区占据了剩余的内存空间。栈区是记录静态变量、程序执行的簿记信息及程序命令的等价的数值和字符串的存贮区；堆区是记录程序动量变量、程序动态结构簿记

表 31-21 KERNEL 的 21 个子块(Unit)

名　　称	作　　用
HEAPOPS EXTRAHEAP PERMHEAP	堆操作
SCREENOPS	屏幕控制
FILEOPS	文件目录操作
PASCALIO EXTRAIO	文件层的I/O 操作
SOFTOPS SMALLCOMMAND COMMANDIO	I/O 的重定向和链接
STRINGOPS	串操作
OSUTIL	转换应用程序
CONCURRENCY	并发处理
REALOPS	浮点数功能和实数 I/O
LONGOPS	长整数操作
GOTOXY	屏幕光标控制
KERNEL	P-system 操作系统的常驻内存部分
GETCMD USERPROG INITILIZE PRINTERROR	P-system 暂存内状部分

信息的区域;代码池区是记录程序被存贮、执行的内存区。

根据 P-system 的安排,栈区属 P-机器,堆属核区,堆、栈两区散布在操作系统的子程序和数据中,与它们紧密相连。实际上的栈、堆是一组子程序及操作系统生成的数据。就应用程序而言是操作系统的扩充,而操作系统本身又能扩充于 P-system 环境中,因此,栈、堆与栈、P-机器是较难区分的。

除了以上介绍的内部主要部分外,P-system 还拥有支持 8087 数值处理器的宏汇编程序、编辑程序、排错程序、文件库及应用程序集。应用程序集包括 Turtlegraphics、宿主代码产生程序、交叉编译程序和 Xeonfile 四程序。

Turtlegraphis 是一组根据 P-system 以及它所支持的高级语言程序的语句命令,来描绘高分辨率 CRT 图形的子程序。Turtlegraphis(海龟图形)这个名字是从图形处理的方法得来的,该概念源于把一个海龟放在纸上,让它根据指令在爬过一定距离后转过某个角度取其尾巴轨迹形成图形这一基本思想。Turtlegraphis 同 IBM-BASIC 的绘图功能是有差别的:IBM-BASIC 根据坐标绘图,而 P-system的Turtle BASIC 根据垂直、水平坐标画点,P-system 则设置"海龟"的适当位置。两种方法因用户的要求不同各有千秋。

宿主代码产生程序,是将 UCSD 语言产生的 P-代码变换成实际系统 CPU 指令(在 IBM PC 中是 8088/86 指令)的程序。所以要求快速处理的程序能直接转换成 CPU 机器语言执行。宿主代码产生程序并不是让 CPU 指令完全替换 P-代码,而是替换其大部分而保留其很小一部分。使用宿主代码产生程序,会带来两个不良后果:宿主代码程序过长,不能将程序移植到使用不同的 CPU 系统上。目前,宿主代码产生程序能生成 8086、Z80 CPU 系列的指

令代码。

交叉汇编程序，是把为一种 CPU 写的 P-代码汇编语言程序转换成为另一种 CPU 写的 P-代码汇编语言程序的一组程序。UCSD 宏汇编程序由两个汇编程序组成，他们都有一个依赖 CPU 的核心层和一个依赖 CPU 的段。交叉汇编程序修改 CPU 依赖段，使之能与另外一种目标 CPU 配合。

Xenofile 是一组进行 CP/M-80 环境及 UCSD P-system 环境产生的文件相互变换的程序。目前只能处理 8080、Z80 支持的 CP/M-80 文件。利用 Xenofile，由 CP/M-80 格式化的磁盘文件，便能输送给 P-system 文件系统，反之 P-system 文件也能变换、转存到 CP/M-80 的磁盘上。Xenofie 特别适合于将 CP/M 开发的数据不加丢失地转变成 P-system 程序。

UCSD P-system 与 PC-DOS、CP/M-86 在内存安排上的差别，如图 31-22 所示。图 31-6 是 P-system 在 64KB 内存和 128KB＋K 内存中的映照图。

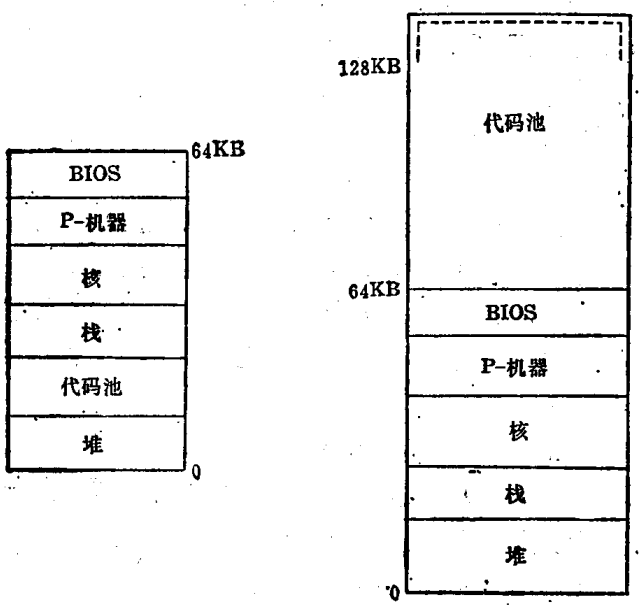


图 31-6　P-system 内存映照图

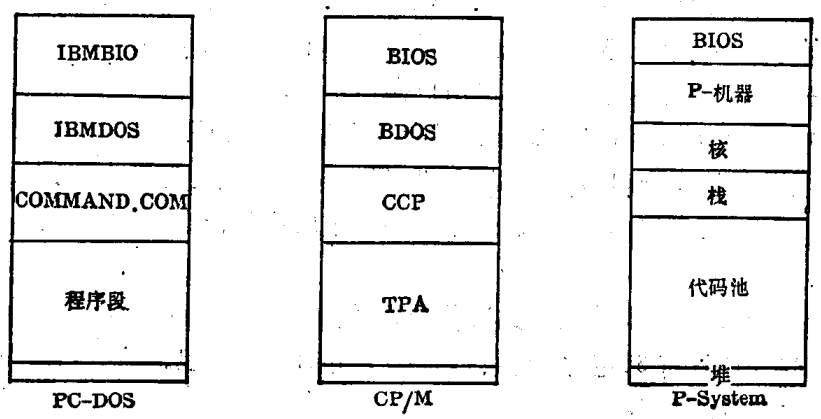


图 31-7　P-system、CP/M-86、PC-DOS 的内存映像比较

在图 31-6 右边的内存映像中，代码池单独地放在一个 64KB 的段中，由于栈、堆区变大了，较大的程序也能运行。从图 31-7 可以看出，PC-DOS 占据内存的高地址部分，事务程序

只占据一小部分低地址空间，用户程序和数据放在内存中间部分。而 P-system 把 P-机器和栈区放在高地址空间，堆区放在低地址空间，代码池或程序区位于内存中部，数据放在堆区或栈区里。

三、P-system 的优点及限制

我们打算从技术及用户两个角度来讨论这个问题。讨论前须先谈谈 P-system 的构造语言 Pascal，因为他对 P-system "风格"的影响是很大的。

Pascal 是一种结构型程序设计语言。在 UCSD-Pascal 中变量、过程均予应以说明。Pascal 源程序被编译程序编译成 P-代码，然后执行。如果拿 Pascal 语言与解释 BASIC 语言相比就会发现 Pascal 更难掌握，编写程序时需较多的事先思考、筹划，程序编制过程也较长。但另一方面，Pascal 在结构上更接近大型机、小型机上通用的 FORTRAN、RPG11 和 COBOL 语言。在弄通读懂 Pascal、P-system 的同时，能获得用户程序或系统程序中一些带有普遍性的技能，这是对 P-system 进行修改、扩充的先决条件。

段交换是 P-system 模拟较先进的虚存技术的一个例子。段交换充分利用磁盘容量大以及 8088/8086 分段访问内存的特点，能满足小内存机器运行大程序、使 CPU 及高速内存 RAM 始终处于忙状态的要求。另外 P-代码解释程序还能减少程序最终代码段的长度，这是因为解释性程序总比非解释性程序较少地占据内存空间。

P-system 的 P-机器概念，决定了 P-system 具有一程序适合多机器的性质。所以 UCSD-P-system 可以只包含少量有价值的程序，其余程序从其它操作系统经拷贝、变换后得到。UCSD 运行软件包是一个含有部分 P-system 软件能把在 P-system 上生成的程序移至非 P-system 系统运行的程序包。这种程序变换应满足磁盘介质兼容条件：配有相同的 5-1/4 吋或 8 吋软盘机。

P-system 许可证发布公司 Softech Microsystems 不久提出了"万能介质"概念。万能介质是个人计算机、AppleII、Zilog Z80 系统、Motorola 68000 系统 5-1/4 吋软盘的标准 UCSD 格式。如果这个格式能被普遍接受的话，目前软盘格式不一致的局面就能打破。这种通用磁盘格式的概念也将扩大 UCSD P-system 移植性的范围。

P-system 的"并发进程"似乎在个人计算机概念中融进了多用户、多任务、Spooling、分时处理的新鲜想法，虽然个人计算机仍是单用户系统，但"并发进程"、Spooling 技术使得操作员无需等全部结果打印出后才离机。进程并发并不意味着 P-system 能提供并发 CP/M-86 的功能，P-system 基本上还是一个单用户操作系统。

转向 I/O 是 P-system 的另一特点。转向 I/O 使得程序能从调制解调器、磁盘文件或键盘获取数据，文件的输出则可以磁盘、打印机、显示器作为最终记录介质。由于 P-system 为所有的外设起了名字，程序中的输入/输出设备能予以重新指定，所以转向 I/O 实为逻辑外设技术。

P-system 还扩充了其支持语言的数值处理功能。如P-system上的P-system语言、Pascal、FORTRAN、BASIC 具有 36 位长整数、Turtle graphics 图形包。这些功能 IBM (Microsoft) Pascal、FORTRAN 并不具备。

总的说来，P-system 是适合开发软件的一个操作系统。它的主要特点是可移植性和紧凑性。P-system 拥有较多支撑 UCSD Pascal 特殊功能的应用程序、开发软件初期必须的工具程序、完整的 Pascal、FORTRAN 语言和程序段交换、进程并发功能。P-system 的不足之处

是用户要花较多的时间掌握操作技能，程序运行速度慢，各类应用程序不多。

表 31-23　P-system 各大组成部分的名称及功能一览表

名　　称	功　　能
BIOS/P-机器	BIOS：基本输入输出系统．用 P-代码及宿主机指令写成依赖宿主机硬件程序，处理全部外设的 I/O 操作．同 P-机器模拟程序相接口．
核 Kernel	核层不依赖宿主机含有文件系统及进程控制系统．有些核代码段常驻内存，有些则在使用时调入，平时存在软盘上．整个核同 P-机器及用户程序接口，受 KERNEL、GETCMD、USERPROG、INITIALIZE 和 PRINTERROR 五子块控制．初始、控制全部外设，解释命令行，装入、执行用户程序是核的主要作用．
STACK	栈区．不依赖宿主机，记录程序静态变量、过程以及函数调用时产生的数据．栈区由 P-机器模拟程序、栈层和用户程序访问．
HEAP	堆区．不依赖宿主机，记录程序动态变量．受 HEAPOPS、EXTRHEAP、PERMHEAP 三子块控制，由核层、P-机器模拟程序和用户程序访问．
CODEPOOL	代码池．不依赖宿主机用于存放用户程序．当内存容量为 64KB 时，代码池夹在堆区和栈区中间．当内存容量大于或等于 128KB 时，代码池单独位于第二个 64KB 段．

§31.5 OASIS-16

前面介绍的三个操作系统都以程序开发为主要对象，OASIS-16 则是针对商业事务处理支持商业应用程序的操作系统。

OASIS 是加利福尼亚澳克兰 Phase One System 公司 Tom Williams 等人 76 年写成的，首先在 Z80 机上实现，属单用户操作系统。OASIS-16 是 OASIS 的 8086 CPU 系统版本，用 C 语言写成的多用户系统。

一、OASIS-16 的内部结构

OASIS6-16 的内部结构见图 31-8，其中 NUCLEUS 是机器资源管理系统调度程序，CSI 是命令串翻译程序，EXEC 是执行语言

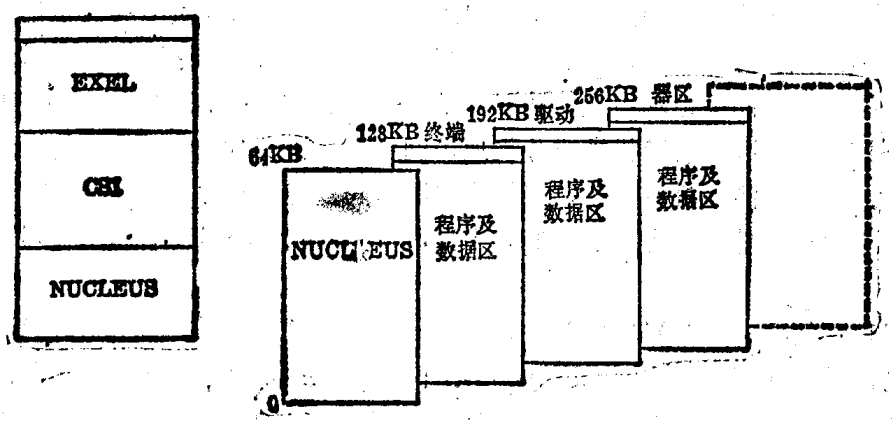


图 31-8　OASIS-16 结构

凡能支持多用户同时工作的系统都有一个多程序数据通路、指令通路的集中管理者，在多用户 OASIS-16 中它就是多用户调度管理程序 NUCLEUS. NUCLEUS 还包括管理硬件资源的 BIOS. NUCLEUS 在磁盘上的文件名是 SYSTEM.NUCLEUS.较 NUCLEUS 高一层的是命令串翻译程序 CSI,它负责翻译操作员打入的系统级命令.CSI 在内存只有一份拷贝，多用户利用再入程序技术共享 CSI. 每个用户除共享 CSI 外还各自备有一个 64KB 的程序区,见图 29-8. 绝大部分程序区空间记存用户的程序、数据. 位于程序高地址区的一些单元属终端驱动器区,存贮管理用户外设的子程序. OASIS-16 的用户接口语言是系统控制语言 EXEC, 它很象作业控制语言，用户用它规定命令文件的执行顺序和测试条件. EXEC 执行时还能从键盘得到命令信息，而 CP/M-86、PC-DOS BAT 文件、CP/M-86 的 SUBMIT, XSUB 等类似于 EXEC 的程序就没有这个功能.

以商业事务为对象的多用户 OASIS-16，还具有一套功能很强的计算机资源记帐系统. 它规定一个用户即是一个帐户，用户名与用户帐号在系统中是等价的. 用户帐目由持有相应帐号的用户访问. 系统程序也算一个用户,帐号为 0，其用户帐名为"system"或称"主"帐. 系统主帐保持在主磁盘上.

用户类型繁多的文件系统，是 OASIS-16 支持多用户商业事务的工具之一. 按访问用户的级别，OASIS-16 把文件分成三类. 第一类为私人文件. 私人文件只能由创立该文件的用户访问，常用来记录敏感信息如清算帐目程序. 第二类是公共文件. 公共文件包括 SYSTEM 帐目中的全部文件如 BASIC 解释程序、字处理程序、定义成公共文件的系统程序. 公共文件能被任何用户访问. 第三类是共享文件，它由一个用户创立. 共享文件是对部分用户"公共"的文件. OASIS-16 为这三类文件指定有特权数,只有特权数大于等于文件特权数的用户,才能访问该文件. 这种以特权数为钥匙的访问方法,能有效地减少非授权用户对敏感文件的读取和修改. 例如用户不能随便调进磁盘格式化程序，不然就可能破坏盘上有用的文件. OASIS-16 还有一个类似于总目录那样的文件,以记录每个用户进入及退出系统时间、帐名、标识名、I/O 设备名的文件 SYSTEM.HISTORY. SYSTEM.HISTORY 文件随时都可被转换成公共的 ACCOUNT.HISTORY 文件,提供用户使用系统的统计信息.

上面介绍的各种类型文件,在组织方式上,较趋近于文件逻辑含义，适合用户口味的数据库基本思想.

OASIS 还具有六种文件组织类型:ASCII 顺序型、直接型、索引型、键型、绝对型和可重定位型. ASCII 顺序型是各种操作系统常用到的文件类型，这种类型的文件记录 ASCII 字符组成的文本.直接型文件是长度固定、可随机访问的数据文件,与磁盘 BASIC 生成的文件相近. 索引型文件是同型记录组成的数据文件,每个记录的键码(如顾客名、顾客号)处记录首部,用户根据键码顺序地、随机地或分类地访问文件. 键文件与索引文件相似，但其键不按顺序存贮. 键文件能随机但不能以键的分类序访问. 绝对程序文件相当于 CP/M-86 或 PC DOS 的. COM 文件，可重定位文件相当于 PC DOS 的. EXE 文件或 CP/M-86 的.REL 文件.

在多用户环境中,实现 OASIS 六种文件组织方式,还得到文件保护机构的支持. OASIS 文件保护机构是文件锁和记录锁.文件锁只允许一用户更新文件、多用户同时读取文件. 该方法在某些场合比只允许一用户读、更新文件要好. 记录锁允许任意多用户访问、修改除去标识部分(记录)外的文件任何部分,这时用户仍遵守特权数规则,记录锁系统可让一个用户

(帐户)读、写一个记录,但不允许另一个用户(帐户)读第一个用户刚读取的记录、或写第一个用户刚写的记录。释放记录锁时,应考虑用户的交替。

文件锁和记录锁的基本思想是一致的,但发生的时机不同。记录锁在用户读取文件的一个记录——也就是读取磁盘上文件有关记录部分时就起作用了,在这个用户转向另一个记录、或关闭文件、或结束文件访问程序前,其他用户不能访问那个记录。文件锁在用户发出锁文件命令后才生效,在用户发出开文件锁命令、或文件被关闭、或处理文件的程序结束前,加了锁的文件的修改权只归一个用户。所以文件锁和记录锁是避免文件上数据不一致性、协调访问文件多个用户的机构。

OASIS-16 在启动后处于单用户状态,SYSTEM 帐的持有者——系统员发 START 命令,接纳终端用户进入系统。多用户工作时,用户与用户、用户与系统员间的消息通过"邮寄"系统传递,能得到(seek) 其他用户当前屏幕上的输出信息、或显示每个用户的当前执行情况、或强迫用户区执行某条命令,当然这些工作只能由系统员进行。

二、OASIS-16 的优点和限制

面向多用户、多任务环境的 OASIS-16 对硬件的要求较高。以 OASIS-16 作为操作系统的计算机系统至少应有 128KB 的 RAM 存贮器、两台双面双密度的小型软盘机、多个串行 RS232C 口和终端 CRT,最好能配硬盘机。所以 OASIS-16 的第一次投资额较大。如果以操作系统占 64KB 每个用户程序区也占 64KB 计算的话,内存总容量为 576KB 的 IBM PC 能带 9 个终端用户。若考虑到 8088 的处理能力及 IBMPC 五个扩展口的限制,PC 带 2～3 个终端用户是较适合的。值得强调的是 OASIS-16 不能支持一些特殊的外设——如数模转换器、数字化转换器、监视器。连接这类外设的系统程序,应由用户自己编写。另外,OASIS-16 的终端设备系统软件——"驱动器程序",放在用户 64KB 程序区中,用户实际的程序、数据空间是不足 64KB 的。

在选择 OASIS-16 时,还应考虑到该操作系统拥有应用软件的数量问题。OASIS-16 虽不至于缺乏商业及通讯软件,但各软件公司为它写的系统或用户程序,比起为 PC DOS 或 CP/M-86 写的要少。随着用 C 语言编写的程序的增多,OASIS-16 软件较缺乏的局面将会改观。选择 OASIS-16 的另一个问题是 8 位版本软件的转换。尽管 8 位 Z80 CPU 软件只占 16 位版本 OASIS 软件很少一部分,但从速度方面看,8 位至 16 位 OASIS 软件转换,优于 CP/M-80 软件到 CP/M-86 或 PC-DOS 软件的转换。

OASIS-16 是个人计算机操作系统领域中很少几个能支持多用户多任务的操作系统之

表 31-24 OASIS-16 组成部分的名称及功能一览表

名　　称	功　　能
NUCLEUS	核层. 机器硬件上的第一层软件,与外设、CSI、EXEC 和用户程序接口. 核层含有 BIOS、文件系统以及 OASIS 实时存贮器管理程序.
CSI	命令串翻译程序. 与 NUCLEUS 及操作员接口,处理用户程序的装入,NUCLEUS 与操作员、用户程序通讯. CSI 被系统全体用户共享.
EXEC	执行语言. 完成作业或处理机控制语言具有的功能,与操作员或文本文件及 CSI 接口.

一，它比较完整，且适合修改。从OASIS终端用户的角度看，OASIS-16较难掌握，熟悉使用方法要花一定的时间。另外系统硬件资源的增减变化，对用户的影响也大于单用户系统。

§31.6 Unix

Unix也是一个支持多用户多任务的操作系统。IBM PC上用的Unix版本至今尚未问世，但与Unix相近专为IBM PC编写的同类Unix不久将会出现。下面介绍Unix的历史、构造语言、内部结构，最后谈谈同类Unix。

初版Unix是美贝尔实验室在1969年配备PDP-7小型机的操作系统，用B语言写成。继B语言后开发的C语言，因其结构化以及具有CPU硬资源访问能力特点成为当今Unix的构造语言。C语言具有ALGOL、Pascal等高级语言的结构特点，又含有处理CPU、访问内存单元等低级语言成份。含有低级语言成份的C语言，从整体上说，仍是一个可移植的语言。C语言摆脱CPU等硬资源影响，保持高级语言可移植的方法是：C语言只设一个主命令核，属于核的命令组成功能子程序，数个功能子程序形成处理I/O的文库命令。由于处理字符I/O的子程序单独存贮，所以命令文库就能随系统硬件或外设而变化，让调用这些文库的命令保持不变。这样C语言在某些程序获得了与CPU外设无关的可移植性。由于Unix大部分用C语言写，C语言又有相当的可移植性，所以Unix的移植性是较好的。一般说来，配有C编译程序的系统就能使用Unix了。C语言对Unix的影响不同于Pascal语言对P-system的影响。Pascal完全是一个高级程序设计语言，C语言则是适合编写系统程序的语言。变换成P-代码的Pascal源程序，只能在P-机器或P-机器模拟程序上运行，C源程序经编译就是机器代码，能直接运行。因此，C程序执行速度快但代码空间较大。

根据定义，C语言中的函数可有递归性，函数能调用自己或相互调用。广义地说，递归调用是大型系统程序功能子程序——进程的基本属性。进程的递归性对程序的影响是：某个程序无需其他程序的帮助就能一次次地从头执行，具有再入性。再入性概念在OASIS-16和并发CP/M-86中都用到过。OASIS-16的NULLEUS层、CSI层程序，并发CP/M-86中的XIOS、BDOS、XDOS、Session Manage在内存均只有一份拷贝，但支持多个用户，Unix也是如此。全部用户共享内存仅有Unix的系统程序，但用户各自的数据区互不干扰。

把Unix从PDP-7、PDP-8机移到微机上要考虑的另一个问题是RAM区管理硬件问题。PDP-7、PDP-8机具有一套高级存贮管理硬件。在它的支持下，存贮空间的实际（物理）起始地址不同于存贮空间的逻辑地址。以PDP-7机为实现裸机的C语言充分利用了上述特点，使得在所有C语言程序看来，它们均被安排在内存初始地址开始起的一段内存中，且占据空间的大小不受限制。如果观察物理存贮器的话，这些程序分段相互错开具有可重定位性。可重定位性使用户程序的逻辑空间与用户"看到"的程序实际物理空间重合。此外C源程序的实际数据区可放在内存的任何段上，与生成数据的程序区分离。程序指令、数据的存取工作由C编译程序和内存管理硬件共同完成。上面讨论的内存管理硬件微机系统（包括IBM PC）是不具备的。因而在微机系统上C语言的一些特点就会部分丢失掉。例如若要保持C语言的递归性、在特殊场合保持再入性的话，所有C源程序必须从内存起始单元开始执行。

一、Unix的内部结构

在讨论过的四个PC操作系统中，如果说PC-DOS与CP/M-86的内部结构较为相似的

话，则 OASIS-16 与现在介绍的 Unix 除构造语言相同、适应环境相近外，它们的内部结构也是相似的。

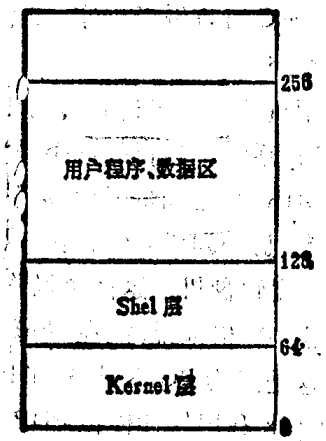


图 31-9 Unix 的结构

Unix 有三个组成部分，它们是核层 Kernel、壳层 Shell 和用户程序、数据区。见图 31-9。核层 Kernel 是系统资源管理核心，主要由文件系统（包括进程管理、内存管理、中断及时钟管理等）的实时管理程序组成。壳层 Shell 是用户能“看到”的操作系统部分，类似于 PC-DOS 的 COMMAND·COM、CP/M-86 的 CCP 和 OASIS-16 CSI。壳层含有丰富的简单命令，提供将这些简单命令组合起来使用的手段。用户通过壳层 Shell 得到 Unix 核层提供的各种服务。

值得一提的是壳层 Shell、核层 Kernel 拥有的 I/O 设备再定向服务。在大多数操作系统中，各类设备均是特殊对待的，CPU 或内存与磁盘机、终端 CRT、打印机、调制解调器等外设的通讯，分别由不同的子程序负责。外设的不同导致外设的信息传递方法不同，但归结起来任何外设总不外乎是字符型设备或块型设备。终端 CRT 一次只能处理一个字符属于字符型设备，而磁盘机一次处理一组字符块属于块型设备。字符型设备与块型设备的差异迫使应用程序在 I/O 段必须遵循一定的处理规则，从而影响 I/O 重定向的实现、程序的可读性、加重了程序员的负担。实际上，设备的统一工作可由操作系统担任。在 Unix 中，文件系统、缓冲区管理程序、设备管理程序把各类设备都当成文件处理即外设统一于磁盘文件。Unix 用户接口显露出的这种性质使得：1）从任何外设的输入都等价于从磁盘文件的输入；2）向任何外设的输出都等价于向磁盘文件的输出；3）应用程序中没有涉及外设的语句或命令；4）程序除能从任何外设输入或输出外，还能向其它程序输出数据或从其它程序输入数据，因而数个程序有可能因输入、输出关系被联在一起；5）I/O 转向成为现实，如向打印机的输出字符，可改存在磁盘上或由 CRT 显示。在 Unix 内部，这种外设统一于文件的思想，是为每类设备写一段“驱动程序”实现的。“驱动程序”包括控制外设需要的特别参数、算法和命令，使比外设“驱动程序”高一层的系统软件能对外设一视同仁。

层次比核层 Kernel 和壳层 Shell 更高的 Unix 成份是程序数据区。程序、数据在该区的具体位置随时间而变化，谁也不知道程序的物理地址。另外程序在内存中是被共享的。Shell 层在解释用户调入程序运行命令时，将检查欲调入的程序是否已在内存。如果在的话，Shell 及 Kernel 只建立描述该程序的位置及数据的各种表格，否则 Shell 先调进程序，再建立表

格。

Unix 丰富的用户软件包，也是使 Unix 成为程序员"工具箱"的因素之一。标准 Unix软件包含有 100 多个应用、调试程序，如C 语言编译程序、文本编辑程序、格式化程序、排错程序、多用户监视器，通讯程序、消息系统。

Unix 是一个证明，即能把大型机的各种先进技术用于小型机的操作系统，它没有适合微型机系统的版本。但是能在微型机系统上运行的同类 Unix 已问世，如 Mark Williams 公司的 Cohorent™ 系统，Quantum Software 公司的 Qunix 系统。Cohorent™ 需要 IBM PC 有 192KB 内存和一台或数台 5MB 硬盘机。Qunix 有单用户及多用户两种版本，需要 IBM PC 有 128KB 内存和几台小型单面双密度软盘机。这两个操作系统均保持了 Unix 程序共享、I/O 转向、支持大容量磁盘机等特点。严格地说，能在 IBM PC 上运行的 Unix，并非真正的 Unix，充其量不过是同类 Unix。这是因为 PC 不具备 PDP 机内存管理硬件、跨出 64KB 段需更改段基寄存器的缘故。同类 Unix 利用软件模拟 PDP 机内存管理功能，这一方面使运行于 PC 上的同类 Unix，保留了 Unix 的基本特点，另一方面降低了对终端用户的响应时间。

二、Unix 的优点及限制

Unix 是极适合程序员开发软件的工具系统。Unix 壳层提供的命令语言，具有程序设计语言的特征，可用来编写 Shell 过程。用户在 Shell 过程中，可以使用变量和控制流结构，调用 Shell 过程时进行参数传递。另外 Unix 允许用户自行设计、规定一套命令语言及解释程序，以代替系统提供的 Shell 命令语言和解释程序，使其工作环境发生根本性的变化。所以 Unix 的用户界面具有很大的灵活性、可更换性。Unix 的核层提供多用户多进程共享内存，以及 CPU、外存及外设转向功能。Unix 的应用软件包，向用户提供大量现成的服务、应用程序。

Unix 的完整性全面性使这个操作系统极为复杂、不易操作，对计算机系统硬资源要求较高。以上不足之处表现在：1)命令名字很短易相互混淆。2) 忽略某个操作细节将导致前面所有工作全功尽弃。如进行 I/O 设备转向时，若用户在键入一长列字符文件后，不向系统指明接收字符文件的磁盘文件的话，整个字符文件将被全部冲掉。3) Unix 要求为 Shell、Kernel 至少准备 128KB 的 RAM 区，为其余 Unix 程序准备 5 兆字节的硬盘机，要有内存硬件管理机构，在多用户环境下配备数台终端设备和串行 I/O 口。4)Unix 没有文件锁及记录锁机构，文件易被错误地读取。

表 31-26　Unix 各组成部分的名称、特点及功能一览表

名　　称	功　　能
Keinel	核层．高度依赖机器硬件，含有外设驱动程序 (BIOS)、文件系统、实时存贮管理系统．核层与外设、壳层 Shell 及用户程序接口．
Shell	壳层．负责装入用户程序，处理 I/O 转向进程并发等工作．壳层与操作员及核层接口．
用户程序数据区	存放用户程序及该程序生成的数据的RAM 区．程序区同数据区可不相连．区中的程序段可被共享．

§31.7 IBM PC操作系统小结及微机操作系统发展趋势

上面讨论了五个专门为微机系统(PC)设计或可以移植到微机系统(PC)的操作系统，它们各有特点和适用环境，其中：

PC-DOS是IBM个人计算机的第一个操作系统，使用极为便利。PC-DOS提供快速磁盘访问，适合大多数应用程序、语言的工作环境。

CP/M-86是CP/M-80的16位机版本，适合运行从CP/M-80转换来的应用程序。CP/M-86与CP/M-80的文件系统兼容，是程序员开发软件的工具。CP/M-86还有并发操作系统版本。

UCSD P-system是以可移植性和紧凑性见长的操作系统，为其编写应用程序的余地比PC-DOS、CP/M-86的要多。P-system的运行软件包，使得没有配P-system的微机系统也能使用在该操作系统上开发的软件。运行速度慢是P-system的不足之处。

OASIS-16是第一个以个人计算机为对象的多用户多任务操作系统，它具有中小型机上操作系统的一些技术特点。在IBM PC上，OASIS-16能带3～4个终端。OASIS-16适合商业事务处理。

Unix或同类Unix是从其它机器移植到个人计算机的操作系统，它能向PC提供大型机上的一些系统管理技术和程序开发、应用工具。由于IBM PC的结构限制，在PC上运行的同类Unix丢失了Unix的一些特性，因而它没有PC-DOS或CP/M-86在个人计算机上用得普遍。C语言的移植性将促使同类Unix软件渗透到其他操作系统。

从上面讨论的五个PC操作系统及整个微机操作系统领域看，微机操作系统所要达到的目标与微机所要达到的主要目标紧密相关，微机操作系统的开发不能脱离微机(PC)的发展，微机芯片等硬件的制造及系统连接必须考虑进以后操作系统的运行要求。一般认为微机操作系统可以分成二类。一类主要用来开发新的软件或软件系统。这类操作系统的编译、调试、编辑、汇编程序是向用户提供的主要开发工具，而且他们提供的软件开发等级也在不断提高。例如CP/M是单用户系统，Unix则是多终端系统。在同一时刻，Unix能支持几个人员开发软件。此外Intel公司的网络开发系统，既可为一个网络中的多用户提供支援，又能为多台处理机上的开发工作给予邦助。另一类操作系统旨在有效地执行已经开发成、经过调试的应用程序。这类操作系统数量大，对于在工业控制、通讯及字处理、交互图象处理、模拟机等领域如何经济、有效地使用微机至关重要。为应用程序的执行而设计的操作系统，除了具备简单的人机接口、预置和资源管理功能外，最好还能提供实时操作、多道程序设计、多有效任务调度等服务。

现有微机操作系统的改进，未来微机操作系统的发展趋势，集成电路芯片的制造技术技巧，用户的要求，计算机科学其它研究领域的成果都是密切相连的。具体表现在：

1．利用VLSI的发展成果

由于在微机芯片上可以集成更为复杂的功能，因此一方面微机操作系统能够依据高性能的硬件芯片支持应用范围更广的程序，满足更高的要求，另一方面微机操作系统必须利用VLSI的技术成果，与硬件芯片紧密啮合。例如由硬件实现计时程序、中断控制程序、多道程序多任务原语等一些传统的操作系统功能。譬如Intel公司的80150芯片就将CP/M-86固

化在内。固化操作系统使微机操作系统向适合微机体系结构方面发展了。

2．制定微机语言标准

制定微机语言标准有两个好处，一是打消用户的忧虑，使他们不会由于缺乏标准化的高级语言而对使用微机犹豫不决。二是使不同卖主设计的语言，可以在同一操作系统上有效地运用。从现今这种标准的实现情况看，Intel 公司 IAMX 86 操作系统的高级语言设计、运用标准是较成功的。由 IAMX86 通用开发接口(UDI)和通用运行接口(URI) 提供的标准相当于一条软件总线，不同风格、细节的语言都可在该软件总线上用，因而避免了为一个新语言配备一套新编译程序的麻烦。

3．保护软件投资

微机操作系统在近期的一个发展目标就是保护软件投资，减少用户因语言、软件的更新造成的损失，从而使用户相信他们投资购买的软件不会很快地失去作用。为达到这一目标可采取的手段是：1．软件向上兼容，8 位 CPU 版本的程序应能方便地转换成 16 位 CPU版本的程序，这种软件兼容以 CPU 指令兼容为基础。2．操作系统模块化、可构造化。这样用户就可以在不造成系统性能损失的前提下，不买或加入操作系统的一些模块。

第三十二章　IBM PC BASIC高级语言

BASIC是当今最通用的计算机高级语言。IBM PC的BASIC语言除了具备BASIC语言一般功能外还根据PC外设的配接情况加进了一些特殊语句。特殊语句分图形、中断控制、声响、通讯文件四类。下面先介绍IBM PC的三级BASIC语言，然后给出以上各类BASIC语句的定义、用途和实例。

§32.1　三级BASIC

IBM PC有三级BASIC。

一、盒带BASIC (cassette BASIC)

这是一种最简单的BASIC，在任何IBM PC机上均可运行。盒带BASIC装在内存40 KB ROM区，除具有Microsoft BASIC一切特点外，还支持在IBM中、高分辨率显示器上定点划线，让内部扬声器发声，使用光笔及游戏装置等功能。

二、磁盘BASIC(Disk BASIC)

这种BASIC工作时，至少需要32KB存贮器和一个软盘驱动器，它还占用用户存贮空间，磁盘BASIC占据的内存总量为12KB，其支持操作系统IBM DOS占12KB，整个系统程序共需24KB内存。磁盘BASIC增加了磁盘输入输出选择、实时时钟存取、存贮图形映象、使用标准RS 232 C接口、支持4台驱动器等功能。在DOS环境下键入"BASIC"即进入磁盘BASIC。

三、高级BASIC

高级BASIC至少需要48 KB的ROM区和一个软盘驱动器，它占据29 KB用户内存区。高级BASIC增加了事件陷井、高级图象和音乐播放命令。高级BASIC通过DOS环境下键入的"BASIA"命令进入。

三级BASIC都具有AUTO（自动编行号）、RETNUMB（重新为BASIC程序编行号）、MERGE（合并程序）命令。这些命令都是很有用的。

§32.2　BASIC图象类语句

1．绘椭圆语句(CIRCLE)

格式：CIRCLE(圆心坐标 x，圆心坐标 y)，半径，[，颜色[，起始角，终止角[，偏心率]]]

版本：高级BASIC

作用：在图形模式CRT上画出以(x，y)为圆心，半径等于给定半径参数的椭圆。

注释：

第一组参数决定椭圆的圆心和半径。圆心坐标可是绝对坐标或相对坐标。绝对坐标是用户可直接在图形模式 CRT 上指出的象点位置，相对坐标指相对于上次访问象点的位置，越出显示器屏幕的椭圆部分不予画出。

颜色参数缺省时椭圆呈前景色（中分辨率模式为色彩 3，高分辨率模式为色彩 1）。起始、终止角参数的单位是弧度，在 -2π 和 2π 间取值（$\pi=3.141593$）。起始角和终止角参数决定椭圆的起始和终止角度。若它们中有一为负值（不允许 -0），椭圆内有一条圆心至圆周的连线，所张的角度是负角参数的绝对值（这与在角参数上加 2π 不同）。起始角参数可以小于终止角参数。

偏心率参数决定椭圆 x 半轴与 y 半轴之比。缺省时 $x/y=5/6$（中分辨率模式）或 $x/y=5/12$（高分辨率模式）。按这种比例画出的椭圆在阔、高标准比为 4/3 的屏幕上是较直观的。偏心率小于 1 时为 x 半径，即半径（长短轴）以水平方向象点间的距离为单位；偏心率大于 1 时为 y 半径，即半径（长短轴）以垂直方向象点间的距离为单位。偏心率等于 1 时可在中分辨率模式 CRT 上得到较圆的椭圆，绘制速度也快。

实例：
```
10  CLEAR, , 8000'space for PRINT
20  KEY OFF'turn off soft key displap
25  'medium res color, black backgr, palette 0
30  SCREEN 1, 0: COLOR 0, 0
35  'set increments for start, stop of arc
40  S1 = .5 + RND*5:S2 = RND*2
50  CLS 'blank screen
55  'radius = 5, 7, 9···99
60  FOR R = 5 TO 100 STEP2
65  'Keep Stare<2*PI
70  A2 = A + S1:1F A2>6.28 THEN A2 = A2 - 6.28
75  'arc in mid screen, radivs R, Color2
80  CIRCLE(160, 100), R, 2, A, A2
85  'Keep arc within 2*PI
90  A = A + S2:1FA>6.28 THEN A = A - 6.28
100 NEXT
105 'fill in with color 3
110 PRINT(160, 100), 3, 2
```

2．着色语句（COLOR）

1）文本模式

格式：COLOR[前景][,[背景][,边界]]

这里：前景表示字符数，是一取值 0～31 的整数表达式；

背景表示背景色，取值 0～7 的整数表达式；

边界表示屏幕边界的颜色，取值 0～15 的整数表达式。

版本：盒带、磁盘、高级 BASIC

注释：显示器为彩色/图形监视器时，前景色取自下列色集：

0—黑，1—兰，2—绿，3—深兰，4—红，5—品红，6—棕色，7—白，8—灰，9—淡兰，10—淡绿，11—淡青，12—粉红，13—淡洋红，14—黄、15—加强白

前景色值加上 16 后显示的字符闪烁。

背景色取自上列色集的 0～7 号颜色。

显示器为 IBM 单色显示器时，前景色取自下列色集：

0—黑，1—白背景带下划线，7—白。

15—加强白

前景色值加上 16 后，显示出的字符闪烁。如 16 指定会闪烁的白字符，31 指定会闪烁的黑字符。

背景色取自如下色集：

0—黑，1—白

在 IBM 单色显示器上使用其他前景色、背景色值时，将产生标准白字黑底的显示效果。假如 BASIC 程序既要在带 IBM 单色显示器的 PC 又要在带彩色/图形显示器的 PC 上运行的话，选择色彩值时应注意以下的对应关系：

前景色	IBM 单色 CRT 上的显示效果
0，2～7	白字黑底
1	加下划线
8，10～15	黑底加强白字
9	加下划线，加强白字
16，18～23	闪烁
17	闪烁，加下划线
24，26～31	加强白字，闪烁
25	闪烁，加下划线，加强白字

前景色	背景色	IBM 单色 CRT 显示效果
0，8，16，24	0，8	不显示
8，16，24	7	白底黑字

注意：

1）前景色可等同于背景色，这时显示出的字符是看不见的，变动前景色或背景色后字符又可见了。

2）任何 COLOR 的参数均可省略，省略了的参数取其原先值。

3）COLOR 语句以逗号(，)结束的话，发生“少操作数”错，但色彩不变。

4）值越出[0，255]时，发生“非法调用”错。实际值取以前用过的值。

5）显示器不同时，颜色及闪烁效果可以不同。

实例 1：10 COLOR 14，1，0 语句设置一幅黄前景色、兰背景、黑边界的图象。

实例 2：
```
10  PRINT"Enter your";
20  COLOR 15, 0'highlight next word
30  PRINT"password";
40  COLOR7'return to default(white on black)
```

```
 50  PRINT"here:";
 60  COLOR 0 'invisible(black on black)
 70  INPUT PASSWORD$
 80  IF PASSWORD$ = "secret"THEN 120
 90  'blink and highlight error message
100  COLOR31:PRINT"Wrong Password":COLOR7
110  GOTO 10
120  COLOR0, 7 'reverse image(black on white) ,
130  PRINT"Program continues…";
140  COLOR7, 0 'return to default(white on black)
```

上段程序将键盘上键入的口令字符串屏蔽起来，不显示在 CRT 上。

2）图形模式

格式：COLOR[背景] [,[色板]]

这里：背景指背景色为取值为 0～15 的整数表达式，

色板指选择的色板为取值 0 或 1 的数值表达式。

版本：盒带、磁带、高级 BASIC

注释：图形模式的 COLOR 语句，只适用中分辨率图形 CRT，由 PSET、PRESET、LINE、CIRCLE、PAINT 和 DRAW 语句设置实际颜色。前景色取值同文本模式下的 COLOR 语句，背景色取自下列色集：

颜　色	色　板　0	色　板　1
1	绿	深　兰
2	红	品　红
3	棕	白

图形模式下的 COLOR 语句的参数可省略，省略参数取其原先值。另外它不可用于高分辨率模式 CRT、值不越出 0～255，否则将出现“非法函数调用”错误。

实例 1：
```
5  SCREEN 1
10 COLOR 9,0
```
上程序段将屏幕置成淡兰背景色，0 号色板。

实例 2：
```
5  SCREEN 1
20 COLOR, 1
```
上程序段设置淡兰背景色，1 号色板。

3．绘图语句(DRAW)

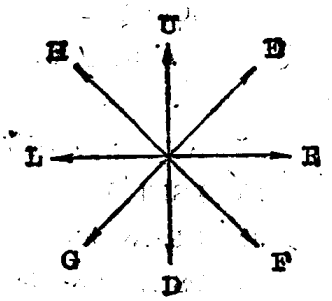
格式：DRAW string

string 是串表达式由下面介绍的绘图命令构成。

版本：高级 BASIC

作用：画出 string 指明物体的平面图形

注释：String 中的字符命令为“图形定义语言”，BASIC 根据这些由单个命令组成的串绘出图形。下列的移动命令，以“最近访问过的点”作为起始点，命令执行结束时，绘出的最后一个点作为下一个命令的“最近访问点”。



Un：上移　　En：右斜对角线上移

Dn：下移　　Fn：左斜对角线下移

Ln：左移　　Gn：右斜对角线下移

Rn：右移　　Hn：左斜对角线上移

以上八条命令中的 n 代表移动的距离，移动的象点数等于 n 乘 s 命令设置的标量因子。

Mx，y：绝对移动或者相对移动。x 前写了加号(+)或减号(-)表示相对移动。CRT 屏幕尺寸比决定了沿水平、垂直、对角线方向象点之间的距离。对标准尺寸比为 4/3 的屏幕来说，4 个水平方向点的距离，等于 3 个垂直方向点的距离。语句 DRAW“U30R40D30L40”可在 CRT 上绘出正方形。

下面两条前缀命令，可加在上述任何移动命令前。

B：移动但不绘出点

N：移动后返回命令起始点

A、C、S、X 分别是图形旋转命令、着色命令、标量设置命令和子串命令，可用他们对已有的图形进行加工。

An：设置旋转角度，n 取值 0、1、2、3 表示不旋转，逆时针转 90°、逆时针转 180°、逆时针转 270°。旋转过 90° 和 270° 的图形因标量关系，它们在 4/3 屏幕上的最终图形尺寸不变，还是在 0° 或 180° 处的样子。

Cn：设置颜色。中分辨率模式下 n 取 0、1、2 或 3，高分辨率模式下取 0 或 1。

Sn：设置标量因子。n 取值在 1～255 之间。实际标量因子 = n/4，如 n=1 时标量因子为 1/4。标量因子乘上水平、垂直或对角线移动命令给出的 n 值，就是真正的移动距离。

X string：执行子串即再次绘出 string 定义的图形的全部或部分线条。

上述命令中的参数 n 可是常数或数值变量。使用变量时，命令间应有分号(;)，在 n 为常数的 string 中分号可省略。

实例 1：绘出矩形

```
 5  SCREEN 1
10  A = 20
20  DRAW"U = A; R = A; D = A; C = A;"
```

实例 2：绘出三角形

```
10  SCREEN 1
20  DRAW"E15; F15; L30"
```

实例 3：绘出矩形及该矩形逆时针 90° 的旋转图形和逆时针 90×I° 的旋转图形。(I 应等于 0、1、2 或 3)。

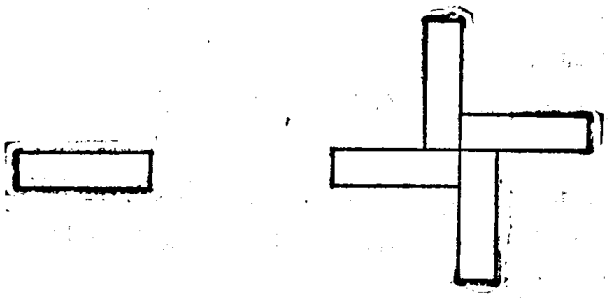
```
 5  A$ = "R40; U10; L40; D10"
10  DRAW A$
20  DRAW "A1; XA$"
```

```
30  DRAW "A=I;XA$"
```

实例 4：

```
 5  SCREEN 1
10  COLOR 4,0
15  CLS
20  A$="R40;U10;L40;D10"
25  PSET(55, 155)
30  DRAW A
35  PSET(197, 155)
40  FOR I=0 TO 3
50  B="A=I;XA;"
55  DRAW B
60  NEXT I
```

结果：



4．取图形语句 GET(Graphics)

格式：GET(X1, Y1)－(X2, Y2),数组名

版本：高级 BASIC

作用：将图形模式 CRT 上的一个点域读入数组

注释：GET 语句将指定矩形中的色彩数据送入数组，矩形由其对角点(x1, y1)、(x2, y2)坐标决定。存放矩形的数组必须是数值型的，其元素的精度任意。数组占 4＋INT((X*bitsperpixel＋7)/8)*y 个字节，其中 x、y 分别是矩形的水平边长和垂直边长，bitsperpixel 在中分辨率模式下为 2，高分辨率模式下为 1。GET 语句用于图形的快速移动。

实例：决定存放 10×12 矩形的数组大小(中分辨模式)

需要的字节数为 4＋INT((10*2＋7)/8)*12＝40。因为整型数组的每个元素占 2 字节，单精度数组的每个元素占 4 字节，双精度数组的每个元素占 8 字节，所以存放该矩形的数组至少有 20 个元素。

存放矩形数组的元素安排是：

1．水平方向边长(以点为单位)占 2 字节

2．垂直方向边长(以点为单位)占 2 字节

3．余下放矩形的点值数据

使用整数型数组，可从数组元素了解矩形的边长及点的信息。在整数型数组中，0 号元素记录矩形的 x 边长，1号元素记录 y 边长。矩形行的第一象点数据总从字节边界开始存放，

当行数据不是字节整倍数时，最后字节用 0 填充。另外要注意整型数据的存放顺序是先低字节后高字节，矩型数据存放顺序恰好相反，先高字节后低字节。当 X1 MOD4 = 0 (中分辨率模式)或 X1 MOD8 = 0 (高分辨率模式)时，由于矩形边界与字节边界吻合，GET 语句的处理速度极快。

5．划线语句(LINE)

格式：LINE [(x1, y1) - (x2, y2) [, [color] [, B[F]]]

这里(x1, y1)、(x2, y2)表示绝对坐标值或相对坐标值，Color 是色彩值，取 0、1、2 或 3。

版本：盒带、磁盘、高级 BASIC

作用：在屏幕上划线或画框

注释：在中分辨率模式 CRT 上，Color 参数选择由 COLOR 语句设置的色板颜色，0 表示取背景色，缺省时表示取色值等于 3 的前景色。在高分辨率模式 CRT 上，Color = 0 表示取黑色，Color = 1表示取白色。

B 参数表示绘出以 (x1, y1)、(x2, y2)为对角线的矩形，BF 表示除绘出矩形外还在矩形里填上 Color 参数指定的颜色。

若给出的坐标值超出了屏幕边界，则该坐标被最靠近它的有效坐标值代替。譬如页号为 0 时，大于 199 的 y 坐标值取 199，大于 639 的 x 坐标取 639。中分辨率模式下，若X值大于 319 的话，图形向下卷动一行。

LINE 语句结束时的最近访问点是(x2, y2)。相对形式的 (x2, y2) 坐标指该点相对于 (x1, y1)点的坐标。

实例 1：

```
LINE(0, 0)-(319, 199)画对角线
LINE(0, 100)-(319, 100)画横线
LINE-(X2, Y2)从最近访问点向(x2, y2)画线
```

实例 2：以任意颜色画任意方向、长度的线

```
 1  'draw random lines in random colors
10  CLS
20  LINE-(RND*319, RND*199), RND*4
30  GOTO 20
```

实例 3：画 320 条竖线，其中 160 条不可见

```
 1  'draw alternating pattern-line on, line off
10  FOR X=0 TO 319
20  line(x, 0)-(x, 199), x AND 1
30  NEXT
```

实例 4：在框中填上颜色 2

```
LINE(0, 0)-(100, 100), 2, BF
```

实例 5：在点(100, 100)和点(110, 80)间画一直线

```
LINE(100, 100)-STEP(10, 10)
```

实例 6：以任意尺寸绘出任意颜色的矩形

```
 1  'random filled box in random colors
10  CLS
20  SCREEN1, 0;color), )
30  LINE-(RND*319, RND*199)RND*2+1, BF
40  GOTO 30'boxes will overlap
```

6．涂色语句(PAINT)

格式：PAINT(x, y)[, paint[, boundary]]

这里

(x, y)表示待涂色区域中一点的绝对坐标或相对坐标，它作为起始点。

paint 表示待涂的颜色，取 0、1、2 或 3。

boundary 表示区域边界颜色，取 0、1、2 或 3。

版本：高级 BASIC

作用：用指定的颜色为屏幕上的某一区域涂色

注释：区域内部涂以 paint 指定的颜色，区域边界涂以 boundary 选定的颜色

在高分辨率模式 CRT 上，paint 参数应等于 boundary 参数。这时涂色意味着从白色区域内某点出发向外涂黑色直到碰上黑色区域，或从黑色区域中的某点出发向外涂白色直到碰上白色区域。

在中分辨率 CRT 上，指定点已有颜色边界时，PAINT 语句不起作用；若省略 paint 参数，则以前景色涂之。PAINT 语句可为任何图形涂色，但对边界不规整的图形，PAINT 语句将占用大量的堆栈空间，所以在为复杂图形着色时，应先用 CLEAR 语句扩展堆栈空间。

实例：

```
 5  SCREEN 1
10  LINE(0, 0)-(100, 150), 2, B
20  PAINT(50, 50), 1, 2
```

语句 PAINT(50, 50), 1、2 为第二行的 LINE 语句绘出的矩形涂色，色值为 1。

7．画点语句(PSET、PRESET)

格式：PSET(x, y)[, Color]

PRESET(x, y)[, Color]

这里：(x, y)为待画点的坐标

Color 为点的颜色，取 0、1、2 或 3

版本：高级 BASIC

作用：在图形模式 CRT 上画一个点，点的位置由(x, y)指出。

注释：若 PSET 语句中无 Color 参数，点的颜色为前景色。在 PRESET 语句中省略 Color 参数的话，点的颜色为背景色。不省略 Color 参数时 PSET 语句与 PRESET 语句功能相同。

点坐标越出屏幕时，PSET、PRESET 语句均不起作用，也不给出错信息。Color 参数值大于 3 时，产生"非法函数调用"错误。在高分辨率模式 CRT 上，Color 参数值等于 2 时的作用相当于等于 0 时的作用，等于 3 时的作用相当于等于 1 时的作用。

实例：从屏幕左下角开始向右上角画对角线，然后反方向抹除它

```
 5  SCREEN 1
10  'draw a diagonal line to (100, 100)
20  FOR I= 0 TO 100
30  PSET(I, I)
40  NEXT
50  'ERASE LINE BY SETTING EACH POINT TO 0
60  FOR I=100 TO 0 STEP-1
70  PRESET(I, I)
80  NEXT
```

8．写图形语句(PUT(Graphics))

格式：PUT(x1, y1), array[, action]

这里

(x1, y1)是待传送图形的左上角坐标

array 保存传送图形信息的数值数组

action 为下列语句或函数之一，缺省时取 XOR

PSET、PRESET、XOR、OR、AND

版本：高级 BASIC

作用：将颜色涂在指定的屏幕区域

注释：PUT 是 GET 的反操作，它把数组元素表示的图形及颜色显示在 CRT 上。action 参数决定数组数据与屏幕上已有图形的相互作用方式。action 取 PSET 时，PUT 完全是 GET 的反操作；取 PRESET 时，图形是原来的负象；取 AND 时应保证待"写"图形处已有图形存在；取 OR 时，传送图形迭在屏幕已有图形之上；取 XOR 可得到动画效果。

XOR 是异或操作。数组数据送上屏幕后，数组数据决定的图形象素被反置视频。如果图形被 PUT 入 CRT 两次，则图形的背景可保持不变，利用这一点可在不破坏背景的情况下移动图形，产生动画效果。得到动画图形的步骤是：

1．用 PUT 语句将图形送上 CRT(XOR)

2．计算图形的新位置

3．第二次用 PUT(XOR)语句将图形送上 CRT，擦去图形的老图象。

4．返回 1，在新位置显示图形。

按以上步骤，背景不会被破坏。减少第 1 步与第 4 步间的时间并保证第 1 步与第 3 步间有足够的延续时间，就可减少图象的跳动。如果要求映出多个动画图形，可按上述步骤逐一处理每个图形。

假如对动画图形的背景不要求十分完整，可用 PSET action 获得近似的动画效果。这时应考虑"新"、"老"图形的相对位置，使新图形能有效地覆盖掉老图形。近似动画产生方法比 XOR action 快，一次 PUT 操作就可移动图形。

如果传送图形太大超出屏幕范围，系统给出"非法函数调用"错。

对于中分辨率CRT的每个象点，AND、XOR、OR 产生下列颜色合成效果。

9．屏幕定义语句(SCREEN)

AND		数组值			
	合成色	0	1	2	3
屏幕	0	0	0	0	0
	1	0	1	0	1
	2	0	0	2	2
	3	0	1	2	3

OR		数组值			
	合成色	0	1	2	3
屏幕	0	0	1	2	3
	1	1	1	2	3
	2	2	3	2	3
	3	3	3	3	3

XOR		数组值			
	合成色	0	1	2	3
屏幕	0	0	1	2	3
	1	1	0	3	2
	2	2	3	0	1
	3	3	2	1	0

格式：SCREEN[mode][, [burst][, [apage][, vpage]]]

这里

mode　取值 0、1 或 2 的数值表达式

0 表示文本模式，屏幕宽 40 或 80 行

1 表示 320×200 中分辨率图形模式

2 表示 640×200 高分辨率图形模式

burst　取值 0 或非 0 的数值表达式，允使或禁止色彩的显示。burst 参数与 mode 参数混合使用效果如下：

黑白/彩色		模式 mode		
		0	1	2
burst	0	黑白	黑白	无用
	非0	彩色	彩色	无用

apage　置有效页即为写屏幕等语句设置显示缓存区页号。apage 应是取值为 0～7 (CRT 宽 40 行)或 0～3 (CRT 宽 80 行)的数值表达式。

vpage　置可见页号即显示页号。vpage 只用在文本模式(mode = 0)，取值同 apage。vpage 可不等于 apage。

版本：盒带、磁盘、高级 BASIC

作用：定义屏幕属性

注释：所有参数均有效的话，新屏幕属性被保存起来，原屏幕属性失效。若新屏幕属性等于原屏幕属性的话，SCREEN 语句不起作用。假如在文本模式下设定了 apage 及 vpage 参数，则显示页被更改。初始时有效页及显示页均被缺省成 0，0。利用 apage、vpage 参数可在显示某页的同时生成另一页，然后进行可见页的切换。

注意：

所有页共享一个光标，切换有效页前，应用 POS(0)和 CSRLIN 保存有效页的光标，返回时用 LOCATE 语句恢复页的光标值。

省缺参数取它们的原来值。当参数值超出允许范围时，系统产生"非法函数调用"错，参数值不改变。对于既要在单色 CRT 又要在彩色监视器上运行的程序，最好用语句 SCREEN 0，0，0 和 WIDTH 40 来设定屏幕属性。

实例 1：10 SCREEN0，1，0，0

选择文本模式，彩色，有效页、显示页定为 0，

实例 2：20 SCREEN，，1，2

mode、burst 不变，有效页号为 1，显示页号为 2。

实例 3：30 SCREEN 2

取高分辨率模式。

实例 4：40 SCREEN1，0

定义中分辨率模式、彩色 CRT。

实例 5：50 SCREEN，1

定义中分辨率模式、黑白 CRT。

§32.3 BASIC 中断控制类语句

IBM BASIS 有一种很有用的特性：它能暂时停止目前 BASIC 程序的执行，转去服务某个外部中断，然后再返回继续执行原来的 BASIC 程序。这里的中断服务程序不是机器语言写的，它也是一段 BASIC 程序。所以 IBM BASIC 语言提供的是 BASIC 一级的中断。BASIC 中断事件包括按下移位键或功能键，从 IBM 异步通讯接收器插板收到了消息、光笔输入、按下操纵杆触发钮。

中断控制类语句的形式是：

ON event GOSUB line

这里 event 是 COM(n)、KEY(n)、KEY(n)、PEN 或 STRIG(n)，line 是 BASIC 中断服务程序的起始行号。执行服务程序的另一个条件是激活事件，这由相应的一套 BASIC 命令完成。例如执行了语句 PEN ON，如果程序中有 ON PEN…语句则下次使用光笔就转入服务子程序。如果执行了 PEN OFF 语句，使用光笔就不转入服务子程序。如果执行了 PEN STOP 语句，只要在语句 PEN ON 执行后使用光笔，服务子程序便立即激活。

1．ON COM 语句

格式：ON COM(n) GOSUB line

这里 n 取 1 或 2，表示第一或第二通讯转接口。

版本：高级 BASIC

作用：为 BASIC 程序被通讯转接口收到字符事件设立服务程序的首行号。

注释：iine 参数为 0 时，禁止通讯转接口陷入事件。ON COM 语句由 COM(n)ON 语句激活，对被激活的 ON COM(n) GOSUB 语句，BASIC 在每个新语句执行前检查通讯口 n，若接收到字符，程序转入 line 参数指定的服务子程序。

如果 ON COM(n) GOSUB 语句，被 COM(n)OFF 语句屏蔽的话，无论通讯口是否收到了字符，程序都不转入程序。

如果在 ON COM(n)前，执行了 COM(n)STOP 语句。收到字符后程序不转入子程序，

但 BASIC 记下该事件，一俟执行到 COM(n)ON，程序立即转入子程序。

进入子程序后，BASIC 马上执行 COM(n)STOP 语句，以防止陷入动作重复发生。子程序的 RETURN 语句自动执行 COM(n)ON，允许再一次陷入通讯口事件的中断，除非陷入处理子程序里写了 COM(n)OFF 语句。

一般说来，通讯口服务子程序应读进整个消息而不是一个字符，否则中断缓存区会由于高波特率下的陷入读字符开销溢出。

实例：

```
150  ON  COM(1)  GOSUB 500
 ⋮
500  REM incoming characters
 ⋮
590  RETURN 300
```

2. ON KEY 语句

格式：ON KEY(n) GOSUB line

这里 n 是取值 0～14 的整数表达式，它与键的对应关系如下：

1～10：F1～F10 功能键

11：光标上移

12：光标左移

13：光标右移

14 光标下移

版本：高级 BASIC

作用：设立按下功能键或光标移位键中断事件服务子程序的首行号

注释：ON KEY 语句的用法与 ON COM (n) GOSUB line 的相同，其激活语句是 KEY(··) ON，屏蔽语句是 KEY(n)OFF，暂停语句是 KEY(n) STOP。

在 n 参数指定的键前按下其他键时，不进入服务子程序，不能用 INPUT$ 或 INKEY$ 测试中断陷入键。

实例：

```
100  on KEY(5)GOSUB 200
 ⋮
200  REM funtion key 5 pressed
 ⋮
290  RETURN 140
```

3. ON PEN 语句

格式：ON PEN GOSUB line

版本：高级 BASIC

作用：设置光笔激活事件服务子程序首行号

注释：ON PEN 语句的用法与 ON COM (n) GOSUB line 的相同，其激活语句是 PEN ON，屏蔽语句是 PEN OFF、暂停语句是 PEN STOP。

打开光笔时不应进行盒带机 I/O 操作。

实例：

```
10  ON PEN GOSUB 500
```

```
500  REM Subroutine for pen
 :
650  RETURN 30
```

4. ON STRIG 语句

格式：ON STRIG(n)GOSUB line

n 取 0 或 2 代表按钮 A 或按钮 B

版本：高级 BASIC

作用：设置操纵杆按钮下事件服务程序的首行号

注释：ON STRIG 语句的用法与 ON COM(n)GOSUB line 的相同，其激活语句是 STRIG(n)ON，屏幕语句是 STRIG(n)OFF，停止语句是 STRIG(n)STOP。

§32.4 BASIC 声响类及通讯文件类语句

IBM PC 内有一 2 吋半扬声器，可用 BASIC 声响类语句 PLAY、SOUND 让其发出声响或播放音乐。

1. 音乐播放语句(PLAY)

格式：PLAY string

这里 string 是由若干音乐命令组成的字符串。

版本：高级 BASIC

作用：根据 string 播放音乐

注释：PLAY 语句的结构与 DRAW 相仿，string 是音乐命令串。下面列出音乐命令串中可用的单字符音乐命令。

A～G：指定的音调，后可跟 #、+ 或 - 号。#、+ 表示升半音，- 表示降半音。

Ln：音符的长短。n 取值范围为 1～64，L1 表示全分音符，L2 二分音符，L4 四分音符。若只要求改变音符的长短，L 参数可写在音符参数的后面，如 A16 等价于 L16A。

MF：前景音乐。SOUND、PLAY 产生的声音以前景音乐的形式放出，即只有当某个音符播完之后，才播放下一个音符。MF 为缺省状态。按下 CTRL BREAK 键后 PLAY 语句退出。

MB：背景音乐。SOUND、PLAY 产生的声音以背景音乐形式放出，即每个音符都被送入缓存区，然后边执行某段 BASIC 程序边播放保存着的命令串决定的音乐。背景音乐命令串，一次可存放 32 个音符或休止符。

MN：正常速度。每个音符的奏出时间为 L 参数规定时间的 7/8。

ML：连奏。每个音符的奏出时间，为 L 参数规定的时间。

MS：断奏。每个音符的奏出时间，为 L 参数规定时间的 3/4。

Nn：音调 n。七种 8 音度可有 84 种音调，固 n 取值范围 0～84。n = 0 时表示休止符。

On：设置当前音度，取 0～6 间任一整数，每一整数表示 C 至 B 7 个音，音度 3 从中 C 音开始。

Pn：休止(暂停)。n 取值范围 1～64。n 的含义见参数 L。

Tn：速度也即每秒 4 分音符的个数。n 取值范围 32～255。缺省时为 120。

Xstring：执行 string 指定的音乐命令串。

. ；符号．跟在音符后的符点将使该音符以符点音符形式奏出，即延长半拍．音符后可跟任意多个符点．“A…”将A音延长至A的L参数确定长度的9/4．“A…”将A音延长至AL参数确定长度的27/8．符点也可跟在休止符的后面，使休止时间延长半拍．

以上命令中的n，既可是常数也可是变量，把音乐命令串表达式或两个音符接起来就能奏出连结音．

实例：

```
10  A$ = "BB-C"
20  B$ = "04XAS;"
30  C$ = "l1CT50N3N4N5N6"
40  PLAY"P2XA$; XB$; XC$"
```

2．声响语句(SOUND)

格式：SOUND freq，duration

这里 freq 是取值范围37～32767的整数表达式，表示发出声音的频率（以赫芝为单位）．

duration 取值范围0～63353的整数表达式，表示以时钟中断间隔为单位的声音延续时间．时钟中断间隔频率为18.2次/秒．

版本：盒带、磁盘、高级BASIC

作用：让扬声器发声

注释：SOUND语句让扬声器发出声音时，程序继续往下执行，直到碰见另一个SOUND语句．如果新SOUND语句的duration延续参数等于0，原SOUND语句不再发出声音，否则程序暂停等原SOUND语执行完后再执行新SOUND．在高级BASIC中，SOUND语句的参数可被缓存起来，从而使程序执行到一个新SOUND语句时不停下来(见PLAY语句中的MB参数)．当前执行非SOUND语句时，语句SOUNDX，0不起作用．

语句SOUND 32767，duration能使扬声器寂静一段时间．

实例：

```
10  'create rendom sounds
20  SOUND RND*1000+37, 2
30  COTO 10
```

在装有异步通讯接收器的IBM PC上，BASIC程序可以与远程设备进行交互通讯，就好象它是一个简单的磁盘文件．通讯文件类语句，包括GET、PUT及I/O语句INPUT #f、LINEINPUT#f、INPUT$、PRINT#f、PRINT#USING和WRITE#f．所有这些语句中，f是指定了通讯口COM1或通讯口COM2的文件说明．由于通讯文件类语句只是指明了COM参数的一般I/O语句，所以在此将它们省略．

§32.5 一组 BASIC 基准程序及比较结果

下面五个BASIC程序，分别用来在IBM PC及Apple II plus、4MHZ Z80、Radio Shack TRS-80II三个8位微机上测试空循环、除、子程序跳转、取子串、求质数执行结果，结果比较

表列于程序之后。

1．空循环

```
 60  A=2.71828
 80  B=3.14159
100  FOR I=1 TO 5000
320  NEXT I
```

2．除

```
 60  A=2.71828
 80  B=3.14159
100  FOR I=1 TO 5000
120  C=A/B
320  NEXT I
```

3．子程序转跳

```
 60  A=2.71828
 80  B=3.14159
100  for i=1 to 5000
120  gosub 1000
320  next i
340  end
1000  return
```

4．取子串

```
 80  A="abcdefghijklm"
100  FOR I=1 TO 5000
120  B=MID$(A$, 6, 6)
320  NEXT I
```

5．求质数

```
  1  SIZE=7000
  2  DIM FLAG(7001)
  3  PRINT"only 1 iteration"
  4  COUNT=0
  6  FOR I=1 TO SIZE
  7  FLAG(I)=1
  8  NEXT I
  9  FOR I=1 TO SIZE
 10  IF FLAG(I)=0 THEN 18
 11  PRIME=I+I+3
 12  K=I+PRIME
 13  IF K SIZE THEN 17
 14  FLAG(K)=0
```

```
15  K = K + PRIME
16  GOTO 13
17  COUNT = COUNT + 1
18  NEXT
19  PRINT COUNT, "primes"
```

结果比较表

机　　型	IBM PC	Apple II plus		4MHZ Z80		TRS-80	
BASIC 版本	IBM BASIC	Apple soft BASIC		MBASIC 4.51		Model II BASIC	
结果	时间(秒)	时间(秒)	与 IBM 之比	时间(秒)	与 IBM 之比	时间(秒)	与 IBM 之比
空循环	6.43	6.66	1.04	5.81	0.904	7.98	1.24
除	23.8	29.0	1.22	24.9	1.05	19.4	0.815
子程序转跳	12.4	13.9	1.12	9.4	0.758	17.1	1.38
求子串	23.0	32.3	1.48	18.6	0.809	24.8	1.08
求质数	190	241	1.27	151	0.795	189	0.995

§32.6 IBM BASIC 命令、语句、实例一览表

一，命令一览表

命　　令	描　　述	例　　子
AUTO	自动生成行号	AUTO 25,500
BLOAD	装入机器语言程序	BLOAD"PICTURE",O
BSAVE	保存机器语言程序	B SAVE"PICTURE",O,&H4O
CLEAR	清程序变量，设立存贮区末地址，栈空间大小	CLEAR,32768,2000
CONT	从断点起继续执行	C ONT
DELETE	删除程序中的一个或数个语句	DELETE 200-300
EDIT	规定 BASTC 程序一个待编辑的行	EDIT 100
FILES	显示磁盘上的文件名	FILES"*.BAS"
KILL	删除磁盘文件	KILL"A.BES"
LIST	显示程序里的一个或数个语句	LIST 300-400
LLIST	打印出程序里的一个或数个语句	LLIST 300-400
LOAD	装入磁盘上的文件(程序)	LOAD"B:REPORT.BAS"
MERGE	联结程序文件与内存中的现存程序	MERGE"A:NUMBERS"
NAME	更改文件名	NAME"A:ACCTS"AS"LEAD"
NEW	清现行程序	NEW
RENUM	改变程序语句的行号	RENUM 100,900,20
RESET	关闭所有磁盘文件	RESET
RUN	执行程序	RUN
SAVE	把程序或文件存入磁盘	SAVE"INVERT"
SYSTEM	退出 BASIC，返回 DOS	SYSTEM
TRON, TROFF	启动、停止程序跟踪	TRON, TROFF

二、BASIC 语句一览表

BEEP	蜂　　鸣	B EEP
CALL	从 BASIC 调用机器语言子程序	C ALL OZ(A,BS,C)
CHAIN	执行新程序、原程序的变量值不变	CHAIN"A:PRO17, 100, AL
CIRCLE	画椭圆	CIRCLE((160, 100), R, 2, A, A
CLOSE	关闭数据文件	CLOSE 1,*2,*3
CLS	清 CRT	CLS
COLOR	设置屏幕前景色、背景色	COLOR 0,7
COM…ON/OFF/STOP	激活/屏蔽 /ON COM…GOSUB 语句	COM(1)ON
COMMON	将变量传送给链接程序	COMMON A,B,C,DC3,G$
DATA	标准 DATA 语句，指派被 READ 识别的数据	DATA 1,0,"JUNE",33
DATES	设立日期	DATES="8/79/82"
DEF FN…	定义用户函数	DEF FN AREA(R)=PI*R^2
DEF SEG	定义当前内存段	DEF SEG=&HB800
DIM	分配数组存贮空间	D IM(25,10,25)
DRAW	绘图	DRAW"E15;F15;L30"
END	结束程序	END
ERASE	回收数组不使用的空间	ERASE A
ERROR	模拟给定的错误条件	ERROR15
FIELD	在随机文件缓冲区开辟单元	FIELD 1,20AS NS,12ASID
FOT…TO…STEP	标准循环语句	FOR I=1 TO 10 STEP 2
GET (磁盘I/O)	取随机文件的一个记录	GET 1
GET (图形)	将屏幕上的图形存入数组	GET(10,10),(40,50),G
GOSUB	转入子程序	GOSUB 500
GOTO	语句转向(无条件)	GOTO 500
IF…THEN…ELSE	标准 1F 语句	IF A B THEN DB=1982:GOTO 3
INPUT	从键盘或数据文件读入数据	INPUT X
KEY ON/OFF	打开或关闭 CRT 第25线指明的功能键	KEY ON
KEY	重新定义功能键	KEY 1,"FILES"+CHRS(13)
KEY…ON/OFF	激活/屏蔽 ON KEY GOSUB 语句	KEY 1 ON
LET	赋值语句	LET A=1
LINE	画线或画方框	LINE(10,10)—(20,20),2
LINE INPUT	从键盘或数据文件读入一整行	LINE INPUT*1,C$
LOCATE	设置光标	LOCATE 1,1
LPRINT	打印结果	LPRINT"Back to mormal"
LPRINTUSING	根据给定格式打印数据	LPRINT CHR$(27);"F"
LSET	向左调整字符串一个场	LSET A$=N$
MIDS	子串替换	MIDS(AS,11)="FLORIDA"
MOTOR	控制盒带机马达	MOTOR 1
NEXT	FOR 语句结束	NEXR I
ON COM/KEY/PEN STRING…GOSUB	接收给定事件的中断，转子程序	ON COM(1)GOSUB 500
ON ERROR GOTO	允许错误陷井子程序	ON ERROR GOTO 100
ON…GOSUB	转子程序开关语句	ON X GOSUB 200,240,260
ON GOTO	开关语句	ON X GOTO 460,500
OPEN	打开磁盘或通讯文件	OPEN"O",*1,"DATA"
OPTION BASE	定义数组起始元素的下标	OPTION BASE 1
OUT	向输出口送一个字节	OUT 32,100
PAINT	为屏幕上的某个区域涂色	PAINT(50,50),1,2

（续表）

PEN ON/OFF/STOP	激活/屏蔽 ON PEN GOSUB	PEN ON
POKE	将数值插入字节	POKE 106,0
PRINT	在 CRT 上显示文本或把数据存入磁盘	PRINT A$,B$
PRINTUSING	格据格式在 CRT 上显示文本或把数据存入磁盘文件	PRINTUSING"!";A$
PRESET	在 CRT 上画一个点，取背景色	PRESET(2,2),2
PSET	在 CRT 上画一个指定颜色的点	PSET (2, 2),2
PUT（磁盘I/O）	将一个记录写入随机文件	PUT#3A$,B$
PUT（图形）	将存贮图形映在 CRT 上	PUT(10,15),A,XOR
RANDOMIZE	初始随机数发生器	RANDOMIZE
READ	从 DATA 语句读取数据	READ A,B,A$,C
REM	注释语句	REM this is a remark
RESTORE	DATA语句读取指针复位	RESTORE
RESUME	从出错程序返回	RESUME
RSET	向右调整子串一个场	RSET A$=B$
SCREEN	为CRT 设置文本或图形显示模式	SCREEN 0,1,0,0
SOUND	扬声器发声	SOUND350,2
STOP	停止程序执行	S TOP
STRINR ON/OFF	允许/禁止操纵杆按钮	STRING ON
STRING…ON/OFF	激活/屏蔽 ON STRIG…GOSUB 语句	STRING(2) ON
SWAP	交换两变量的值	SWAP A1,B2
TIMES	设立时间	TIMES="0:39"
WAIT	标准 Microsoft 等待语句	WAIT 32,2
WEND	WHILE 循环结束	WEND
WHILE	循环语句，条件真就不转出	WHILE FILEA
WRITE	向CRT 或文件输出数据	WRITE A,B,CS

三、BASIC 实例一览表

ABS	取绝对值	ABS(X)
ASC	ASCII 码转换成 ASCII 代码值	ASC(AS)
ATN	反正切	ATN(X)
CDBL	双精度转换	CDBL(X)
CHRS	数值转换成相等的 ASCII 码	CHRS(X)
CINT	取整数值	CINT(X)
COS	余弦	COS(X)
CSNG	单精度数转换	CSNG(X)
CSRLIN	取光标所在行行号	CSRLIN
CVD	串转成双精度数	CVD(VS)
CVI	串转成整数	CVI(NS)
EOF	逻辑测试文件结束	EOF(1)
ERL	最后一个错误	X=ERL
ERR	最后一个错误的错误码	Y=ERR
EXP	自然指数	EXP(X)
FIX	x 的整数值	FIX(58.75)
FRE	取未用的工作空间大小	FRE(0)
HEX$	取串取应的16进制数	HEX$(X)
INKEY$	从键盘读入一个字符	X$=INKEYS
INP	从I/O 口读入一个字节	A=INP(225)

（续表）

inpu$	从文件读入字符	X$=INPUT(1)
INT	取整数值	INT(99.89)
LEFT$	取第一个字符开始的子串	LEFT$(A$,5)
LEN	求串长	LEN(X$)
LOF	取文件占据的空间大小	LOF(1)
LOG	自然对数	LOG(54/7)
LPO$	取打印机回车位置	LPO$(2)
MID$	从给定串取一子符	MID$(A$,11)="FLORID
MKD$	双精度数转换成字符串	MKD$(X)
MKI$	整数转换成字符串	MKI$(X)
MKS$	单精度数转换成字符串	MKS(X$)
OCT$	10进制数转换成8进制数	OCT$(24)
PEEK	读指定地址单元的内容	PEEK(&H410)
PEN	读光笔	PEN(3)
POINT	取图形模式 CRT 上点的颜色值	POINT(X,Y)
POS	取光标所在列的列号	POS(0)
RIGHT$	取串最后若干字符组成的子串	RIGHT(A$,7)
RND	随机数	RND(2)
SCREEN	设置文本模式CRT或图形模式CRT的属性	SCREEN(10,10)
SGN	取数值表达式的符号	SGN(X)
SIN	正弦函数	SIN(X)
SPACES	建立空串	SPACE$(I)
SPC	打印空格	SPC(15)
SQR	求平方根	SQR(X)
STICK	取操纵杆的坐标	STICK(2)
STRING$	建立由一种 ASCII 字符组成的字符串	STRING$(10,X$)
STR$	数转换成串	STR$(X)
TAB	在打印机上留空格	
TAN	正切函数	
USR	调用机器语言程序	
VAL	串转换成数值量	
VARPTR	取变量或文件控制块的地址	

第三十三章　IBM PC的两个应用实例

§33.1　三维图形显示软件包

这是一个8086/8088、8087协同处理数据的例子，介绍这个例子的目的。是让读者建立起在86系统的CPU上实现图形显示处理的一般概念，加深对前面讲过的8087浮点数处理器的认识，同时也介绍立体图形的显示方法。

一、三维图形显示及对计算机系统的要求

三维空间中的一个点，在计算机里是以一个四维向量(x，y，z，c)表示的，x，y，z是该点的坐标，c是一个便于处理的数，c一般取1。三维空间中的一条线段，可用两个四维向量表示，较简单的物体可用一系列线段表示。三维空间中以一系列点向量表示的物体经平移、旋转、标度操作(或称变换)就能在平面上显示出来，且具有立体感。所以对点向量的变换，是实现三维图形显示的关键。

根据空间解析几何原理，变换能够用矩阵与点向量的乘积表示。这是因为一个4×4的幺阵与点向量的乘积还是该点向量本身，适当定义的4×4非幺阵与点向量的乘积，可以使该点坐标发生某种有规律的变化，这种变化实际就是点在三维空间中的平移、旋转、标度。

变换操作(矩阵乘法)等显示处理，对计算机系统特别是CPU提出了大吞吐量、高处理速度，能进行硬件级浮点数运算的要求，原因有下几条：第一，实际观察面也即"窗口"越大，能看到的物体也就越多、越大，窗口的大小在计算中完全取决于它能表示的最大数；第二，图形最终在计算机内以内部坐标表示，内部坐标的单位可以是显示屏幕光栅格的长度。因而要进行外部坐标与内部的换算，为确保精度就应用32、64甚至80位长整数运算指令，使显示图形无失真或稍有失真；第三、复杂物体的点向量数目巨大，一个主体图形的产生必须先经大量的计算如矩阵联级的相乘，只有吞吐量大、处理速度高的CPU，才能胜任这种工作，在用户能容忍的时间内完成一帧图象的处理。总结上面几点就得到如下结果：适合图形立体显示的计算机系统，必须具备长整数及浮点数指令、高速度运算、大的数据吞吐量等特点，在介绍过的86系列中，8086/8088、8087的组合系统如iAPX86/20，就具备以上的条件，足以满足一般较简单物体的立体显示。图33-1是iAPX86/20的框图，有关8086、8087及系统构成的详细情况，请参阅讲解芯片的章节。下面简要介绍三维图形显示的基本要点：

1．输入表示物体形状的数据

这是图形显示软件用户接口完成的工作，主要将物体关键点的坐标，送入计算机系统，复杂物体有着大量的关键点，完全靠手工输入是困难的，较先进的办法是用电视摄象机摄下物体的图象，经特别处理后送给显示软件，不足之处是得不到物体的透视图象。

2．进行当前变换

当前变换指在整个三维空间对物体施行平移、旋转、标度操作或它们的组合。图33-2～

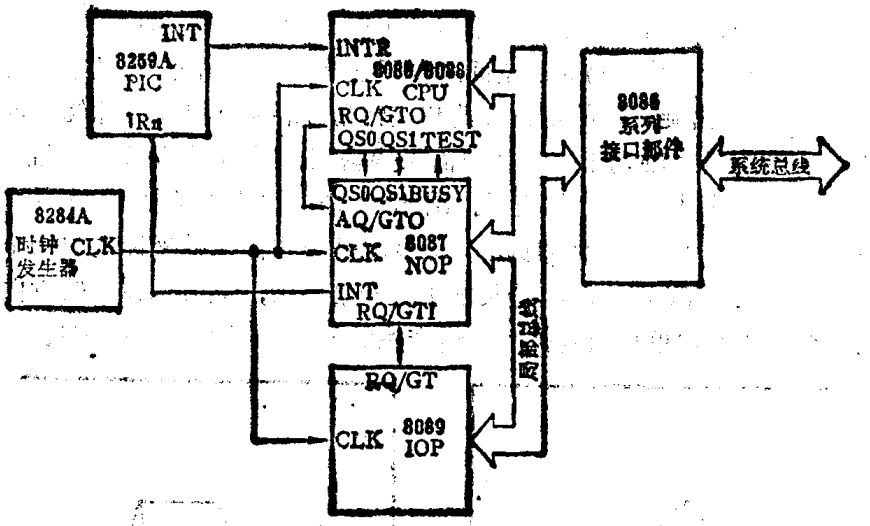


图 33-1 86/20 系统框图

11 是变换操作的例子，坐标均以 x、y、z 顺序排列，观察点定在正方向 z 轴上。图 33-2 是中心处点(0，0，0)，观察点在(0，0，10)的 4×4 立方体图象。图 33-3～33-5 是一组图形平移的例子。图 33-2 展示的物体沿 z 轴正方向移动了两个单位后，由于物体离观察点近了，所以图形变大(图 33-3)。图 33-4 展示了物体沿正 x 方向移动 2 个单位距离后图形，因为立方体边长的一半也是 2 个单位，所以立方体的一个面与 z 轴重合，反映在显示平面上就是

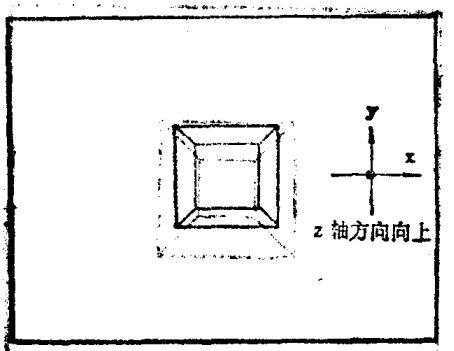


图 33-2 4×4×4立方体 中心(0,0,0) 观察点(0,0,10)

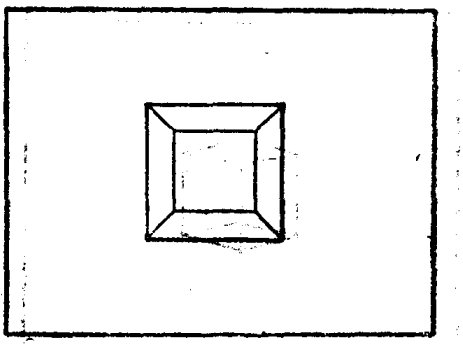


图 33-3 中Z Z 平移

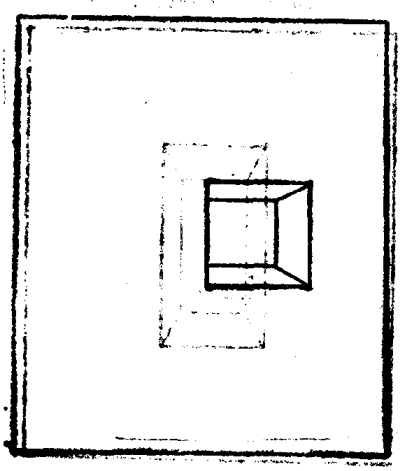


图 33-4 +2X 平移

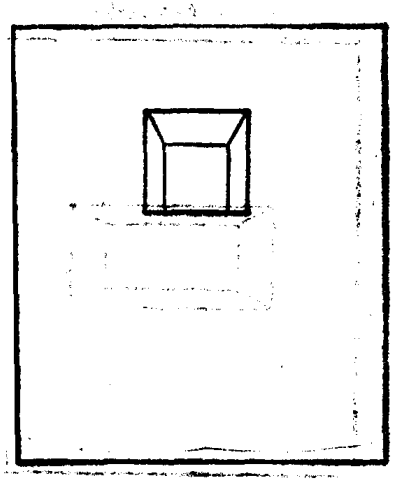


图 33-5 +2Y 平移

一条直线。图 33-5 是物体朝 y 轴正方向平移 2 个单位后的图形。图 33-6～33-8 是有关物体旋转的例子。所谓旋转就是让物体绕空间某一直线旋转一个角度、并显示旋转后物体的立体图象。旋转前必须给出旋转轴的定位参数和旋转角。图 33-6 是立方体绕 z 轴转 45° 后产生的图象，观察点所在的 z 轴仍于立方体顶面保持垂直。图 33-7 是立方体绕 x 轴旋转 45° 后产生的图象，现在 z 轴与一条边相交了。图 33-8 是立方体绕 Y 轴转 45° 后产生的图象。

图 33-9～33-11 是标度立方体的例子，标度就是将物体所处坐标系的三个分量分别放大或缩小 S_x、S_y、S_z 倍。如果将 z 轴单位长度扩大一倍的话，立方体在 z 轴方向就被拉长了(见图33-9)。将 x 轴、y 轴单位长度扩大一倍产生的效果见图 33-10、33-11。

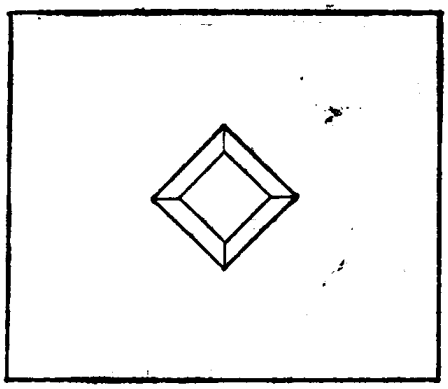


图 33-6　绕 z 轴转 45 度

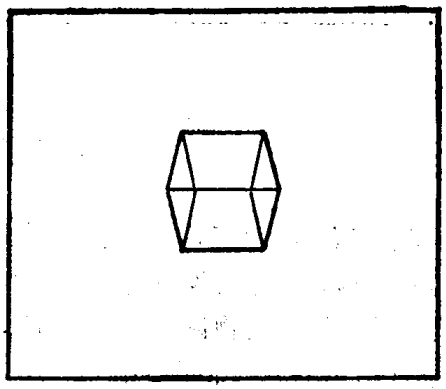


图 33-7　绕 x 轴转 45 度

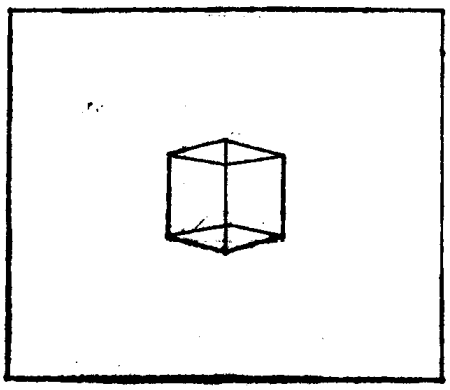


图 33-8　绕 y 轴转 45 度

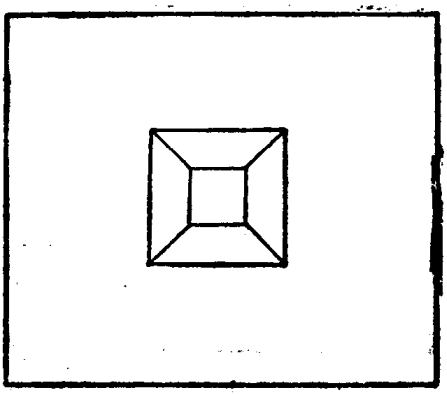


图 33-9　z 轴放大一倍

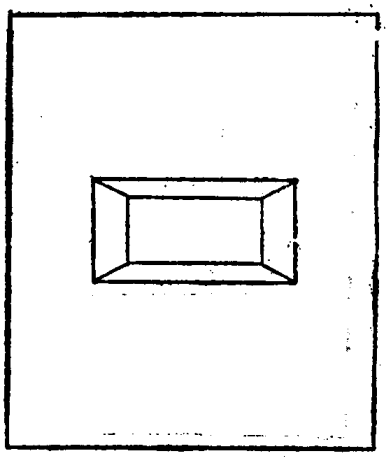


图 33-10　x 轴放大一倍

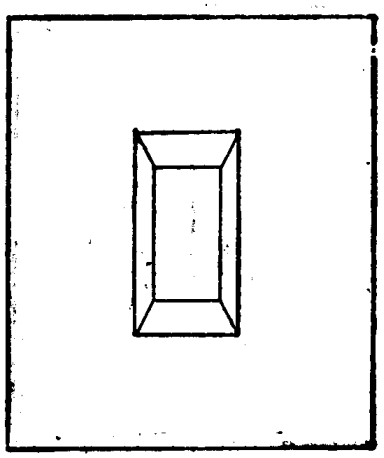


图 33-11　y 轴放大一倍

3．视点变换

视点变换是由于观察点的位置发生了变化而引入的变换，视点变换操作是平移和旋转。图 33-12 是视点变换的一个例子。

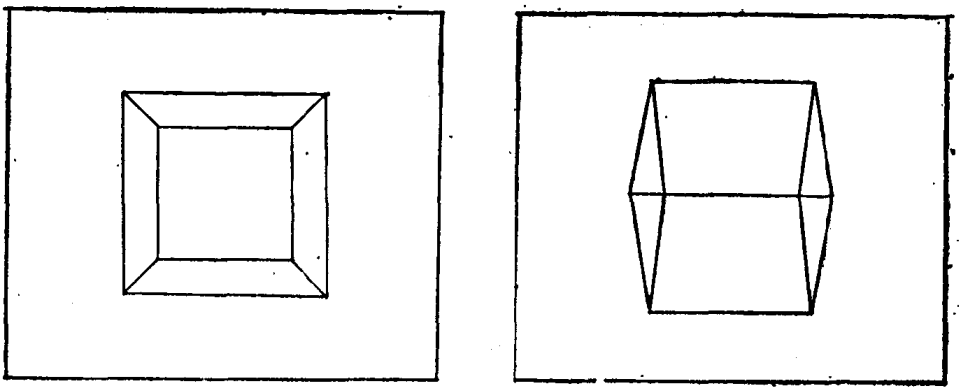


图 33-12　观察点从(0,0,10)点转移(10,10,0)点

4．对三维数据进行 z 剪切

z 剪切就是做两个垂直于 z 轴的平面，除去两平面外的三维数据使观察点出发只能看到物体夹在两个平面中间部分的图象。显然，这两个平面在视野范围之内。

5．三维数据投射到二维平面

这是在平面上显示立体图象的关键步骤。投射依据透视原理进行，改变消失点的位置就能产生不同的显示效果。图 33-13 为夸张透视、抑制透视的例子。夸张透视使得立方体表面和背面在观察方向上的距离增大了，而抑制透视使立方体表、背面在观察方向上的距离缩小了。当物体较靠近观察点时用夸张透视技术，反之用抑制透视技术。投射时还用到点向量的第四个分量。

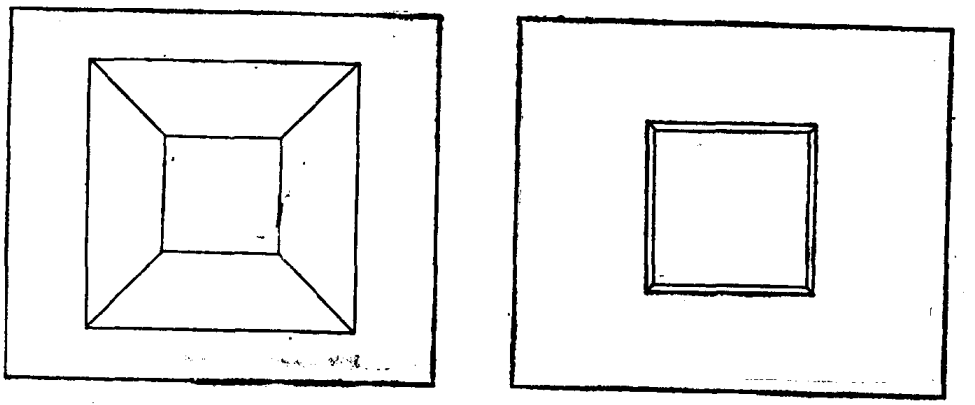


图 33-13　经夸张透视，抑制透视处理的立方体透视象

6．对投射数据施行 X-Y 剪切

X-Y 剪切就是将超出“窗口”外的线条除去。

7．“窗口”至“视区”的变换

该变换根据“窗口”、“视区”的大小进行投射数据的标度处理。

上面提到的“窗口”，描述投影图形在空间上的最大尺寸，其大小由观察者决定，受计算

机数值表示范围的限制。“视区”是屏幕或绘图仪平面上的有效显示尺寸，它取决于图形显示设备，如一种 CRT 的屏幕尺寸为 1024×1024 栅格单位。在字长一定的条件下，浮点数形式能比整数形式表示更大的数，，所以从通用性、实用性出发，用浮点数表示物体是合适的，这至少可使“窗口”开得很大。

下面给出实现平移、旋转、标度的变换矩阵。

1．把一个点(x, y, z)平移到新的位置、(x′, y′, z′)

变换式：

$$[x', y', z', 1]=[x,y,z,1]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ T_x & T_y & T_z & 1\end{bmatrix}$$

T_x、T_y、T_z 分别为 x、y、z 方向上的平移分量。

2．旋转

(1) 绕 x 坐标轴旋转 θ 度，参看图 33-14(a)

变换式：

$$[x', y', z', 1]=[x, y, z, 1]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & 0\\ 0 & \sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

(2) 绕 y 坐标轴旋转，参看图 33-14(b)

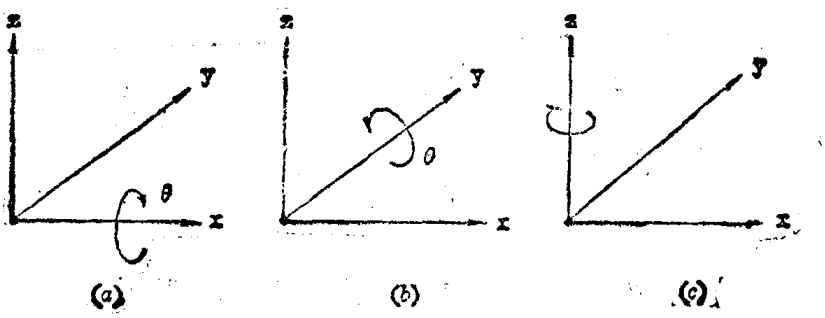
变换式：

$$[x', y', z', 1]=[x, y, z, 1]\begin{bmatrix}\cos\theta & 0 & \sin\theta & 0\\ 0 & 1 & 0 & 0\\ -\sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

(3) 绕 z 坐标轴旋转，参看图 33-14(c)

变换式：

$$[x', y', z', 1]=[x, y, z,\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$



(a) (b) (c)

图 33-14 三维旋转变换

3．标度

变换式：

$$[x', y', z', 1]=[x, y, z, 1]\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

二、显示软件包处理概述

1．变换

软件包用一个4维向量表示空间中一个点，用一个4×4矩阵表示某种变换，变换由点向量与当前矩阵的乘积完成。

在点向量输入后，4×4的当前矩阵被置成幺阵，如果输入了变换命令，则软件包根据变换要求生成一个非零的变换矩阵，并把它同当前矩阵相乘，乘得的结果仍为当前矩阵。最后的当前矩阵，完全反映了一系列变换操作，随后点向量与当前矩阵相乘，变换结束。软件包利用rotate、scale、transl、ident、push、pop子程序实施变换。

2．观察变换

由于观察点、观察方向的不同，上面变换后的点向量，需与观察矩阵相乘，考虑进观察点、观察方向对物体图象的影响。

3．z-剪切

经当前矩阵、观察矩阵处理的点向量，已落在三维空间的适当位置，但由于观察者只能看到三维空间的某个部分，如观察者不能看到观察点后的任何东西，所以必须把点向量映射到"观察者空间"中去，除去不可见点。z-剪切的主要任务是检查每一点向量的z分量，看其是否落入可观察空间，对有些线条还要用"相似三角形"原理除去可见空间外的部分。一句话，z剪切使物体的像正确地"落在"观察者的视网膜上了。

z剪切需要的两个参数，由zlip子程序设定，第一个参数决定观察者"看得到"的垂直于视轴的最近平面，第二个参数决定观察者"看得到"的垂直于视轴的最远平面，第二参数小于第一参数。z-剪切在zlipp中进行。

4．投影

投影就是将z-剪切好了的三维点向量映射到二维平面上。显示软件包只实现了z坐标的单点透视，也就是只能用z分量来修改x、y坐标分量。投影的基本思想是：点离观察点愈远，该点就愈靠近图形的中心点，也就是愈靠近消失点。找到x、y与z的一种函数关系，并让点的x、y分量随z分量变化，投影就完成了。显示软件还是用矩阵乘积实现投影，它规定投影矩阵之(3，4)元素等于透视值的4×4幺阵，点向量与投影矩阵作积仅改变点向量的第4分量，且该分量是点的z分量的函数，最后进行规范化，用第4分量除第1、第2分量。

倘若要求两点或三点透视的话，只需在投影矩阵的(1，4)、(2，4)元素放置投影值，投影矩阵的乘法，规范化操作不变。

显示软件包中的projet子程序，负责建立投影矩阵、完成透视乘法，viewpn计算透视用的消失点，norm进行向量的规范化工作。

5．X-Y剪切

X-Y剪切，除去那些不在"窗口"内的线条，X-Y剪切既可对三维数据进行，也可对二

维数据进行，后者的计算较简单。

图 33-15 是在与视轴垂直由 Viewpoint 第二参数决定的无限大平面中取一窗口的示意图。X-Y 剪切的主要工作是把点向量的 X、Y 分量与“窗口”命令设立的窗口尺寸作比较，使物体落入窗口（区域 1）的部分显示出来，落入非窗口（区域 2、3、4、5、6、7、8、9）的部分不被显示出来。

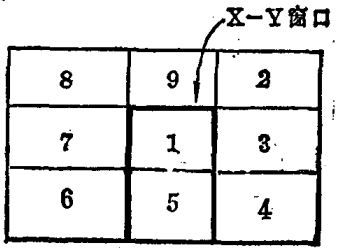


图 33-15　窗口示意图

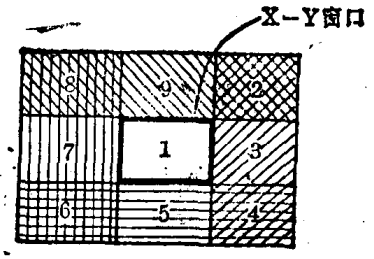


图 33-16　4 个区域显示示意图

假如某线段的一端落在区域 1 内，则该线段至少有一部分是可见的，那些两端点都不落在区域 1 里的线段，也有可能因其穿过窗口而有一段可见。这样就有一个判别线段是否完全、或部分落进窗口的问题。在显示软件包中该工作由两个指示器的与操作完成。显示软件包为线段两端点分配两个 4 位指示器，指示器的 4 个位分别表示 $>x$、$<x$、$>y$、$<y$ 区。在图 31-16 中就是 2、3、4 区；6、7、8 区；2、8、9 区；4、5、6 区。程序将端点坐标与窗口边界坐标比较，若点的 x 坐标大于窗口右边缘坐标，指示器的第 1 位置1；点的 x 坐标小于窗口左边缘坐标，指示器的第 2 位置 1，若点的 y 坐标大于窗口上边缘坐标，指示器第 3 位置 1，若点的 y 坐标小于窗口坐标的下边缘坐标，指示器的第 4 位置 1。最后进行线段两端点指示器的“与”操作，“与”结果为 0 表示该线段至少有一部分是可见的，反之，线段根本不可见。X-Y 剪切程序最后找出至少部分可见线段脱离窗口边缘的那一点，作上必要的标记。

显示软件包完成 X-Y 剪切的子程序有 wtovp（它以适当的参数调 xyclip）、xyclip（进行剪切）、code（返回端点的指示器值）、ppush（算出离开窗口的点）。

6．窗口至视区的变换

三维数据经过上述一系列的变换、剪切、最终落入了窗口尺寸决定的区域，但这里说的窗口是以现实世界常用的里、米等作为长度的，是人们假想的一个区域，不同于实际的 CRT 或绘图仪上的显示平面。窗口-视区变换就是按窗口、视区的比例关系把窗口内的一点映射到视区上去。如果它们间有等比关系，则变换就相当于缩小窗口，如果无等比关系，则变换除进行整体缩小外，还进行 X 或 Y 方向的缩小，最后显示出的图形是变了形的。

视区变换由显示软件包里的 wtovp 子程序段完成。视区建立命令允许视区是图形显示平面上的某个小区域。

三、显示软件包的命令

显示软件包，是用 Intel 公司 FORTRAN 86 编译程序支持的 FORTRAN 语言写的一个较大型的程序，在 iAPX86/20 上运行。三维数据从键盘输入图象由 H7225 A 绘图仪绘出。如果将软件包中有关绘图仪的语句改成 IBM PC 彩色监视器用的显示语句，则图象就可由彩色监视器显示。下面是显示软件包 24 条命令的使用说明，arg 表示参数，argn 表示命令要求的第 n 个参数。

1．comment arg 1

comment 命令，让软件包忽略下面 arg 1 行，它用来在命令行中插入注解。

2．define arg 1

define 命令，让软件包将下面在 enddef 命令处结束的 N 行送入内部缓存区，该N行以后用 arg1 访问。define 命令用来定义物体、视区、窗口、设置观察点。一条 define 命令只可定义 10 个物体。

3．enddef

enddef 命令指明 define 命令的结束行。

4．call arg 1

call 命令，让软件包到内部缓存区取出，并绘出名为 arg1 的图形(可能不止一个)。

5．line arg 1、arg 2、arg 3、arg 4、arg 5、arg 6

line 是线段绘制命令，线段两端点的坐标分别为(arg1 arg2 arg3)和(arg4 arg5 arg6)。line 命令涉及到旋转、平移、标度、窗口变换、视区变换等操作。

6．plot arg 1、arg 2、arg 3、arg 4

plot 也是线段绘制命令，它从上次绘出直线的末端点出发，向点(arg1 arg2 arg3)画一条直线，参数 arg 4 是笔码，等于 2 时表示画一条直线，等于 3 时直线不画出，只将绘图笔移到(arg 1 arg 2 arg 3)点。

7．ident

ident 命令，将当前矩阵置成幺阵，也即将旋转角参数定成 0 度、平移量设为 0 、标度倍数为 1。

8．push

push 命令，将当前矩阵压入一个具有 10 个存贮单元的矩阵栈，当前矩阵不变。

9．pop

pop 命令，把矩阵栈的内容送进当前矩阵。

10．rotate arg 1，arg 2，arg 3

rotate 是旋转命令，它将物体绕 X、Y、Z 轴转 arg 1、arg 2、arg 3 度，rotate 命令不改变物体的原始定义。

11．translate arg 1、arg 2、arg 3

平移命令 translate，将物体沿 x、y、z 轴三方向移动 arg 1、arg 2、arg 3 个单位。

12．Scale arg 1、arg 2、arg 3

标度命令 scale，将物体所处坐标系的 X、Y、Z 轴的长度单位乘上 arg1、arg2、arg3。

13．window arg 1、arg 2

窗口命令 window，建立起窗口，arg 1、arg 2 决定投影平面的大小。直观地说，window 命令造起一个无限高的金字塔，观察点位于金字塔的塔尖，所有被金字塔包含的物体都能在投影平面上展示出来。

14．viewport arg 1、arg 2、arg 3、arg 4

视区命令 viewport，设置绘图仪绘图平面的尺寸，arg 1、arg 2 定义该平面的中心点，arg 3、arg 4 是平面中心到平面相互垂直的两个边缘的距离。

15．viewpoint arg 1、arg 2、arg 3、arg 4、arg 5、arg 6

观察点命令 viewpoint，建立起观察变换用的观察矩阵，arg 1、arg 2、arg 3 是观察点的坐标，arg 4、arg 5、arg 6 是着眼点的坐标。这两个点的连线，形成一个从观察区通过的向量，向量的长度就是透视变量。

16. zlip arg 1、arg 2

Z-剪切命令 Zlip 建立起Z-剪切矩阵，arg 1 参数决定可观察区域与观察点的最近距离，arg 2 决定最远距离。

17. cube arg 1、arg 2、arg 3、arg 4、arg 5、arg 6

cube 命令，在平面上绘出一个中心位于(arg 1 arg 2 arg 3)、半边长 arg 4、arg 5、arg 6 的立方体。

18. arrow

arrow 命令，画出一个从(0 0 0)点出发指向(1 0 0)的箭头。

19. pyramid arg 1、arg 2、arg 3、arg 4、arg 5、arg 6

pyramid 命令，画出一个底面中心在 (arg 1 arg 2 arg 3)点，半边长分别等于 arg 4、arg 5、arg 6 的 4 边金字塔，塔高为 arg 4。

20. current

current 在终端上显示当前矩阵。

21. printdef

printdef 打印物体的定义行。

22. startt

startt 命令，启动 iSBC86/20 板上的 10 毫秒计时器。

23. endt

endt 命令，停止 iSBC86/20 板上的 10 毫秒计时器并打印计时值。

24. end

软件包结束命令，负责打印所有绘出点，显示 "success!!!"，把控制权交还给 ISIS。

四、显示软件包程序

```
      c
      c  this is the main routine of the graphics program. basically
      c     it sets up default parameters for the rest of the routines, then
      c     enters an infinite loop, alternatively fetching lines from the input
      c     (using routine getl) and sending them to be processed by the graphics
      c     processor (proc)
      c
     1        common/windoe/wxh, wyh
     2        common/viewp/vxh, vyh, vxc, vyc
     3        real*8 wxh, wyh, vxh, vyh, vxc, vyc
     4        common/matrix/currm, view, curp
     5        real*8 currm(4, 4), view(4, 4), curp(4)
     6        common/clip/hither, yon, dee
```

```
 7          real*8 hither, yon, dee
 8          common/stacks/stackp, sspace
 9          real*8 sspace(10, 4, 4)
10          integer stackp
11          common/defns/dargl, darg 2, darg 3, darg 4, darg 5, darg 6, darg 7, entry, tailp,
            ends
12          character*10 dargl(500)
13          real*8 darg 2(500), darg 3(500), darg 4(500), darg 5(500), darg 6(500), darg 7
            (500)
14          integer entry(10), ends(10)
15          integer tailp
16          common/cstack/cnum, cnump
17          integer cnum(10), cnump
18          common/penpos/xpos, ypos, pcount
19          real*8 xpos, ypos
20          integer*4 pcount
    c   initialize the plotting package
21          call plots
    c   initialize the stack pointer
22          stackp = 1
    c   set up a few defaults
23          wxh = 10.
24          wyh = 10.
25          vxh = 5.
26          vyh = 5.
27          vxc = 5.
28          vyc = 5.
29          hither = 1.
30          yon = 100.
31          dee = 10.
32          tailp = 1
33          cnump = 1
34          xpos = -1.
35          ypos = -1.
36          pcount = 0
37          print*, 'GRAPHICS program entered!!!'
    c
    c   initialize the current matrix
    c
```

```
38          call ident(currm)
    c
    c   and process all the input lines
    c
39          100         call getl
40                      call proc
41                  goto 100
42                  end
    c
    c     getl(line)
    c
    c     fetches the next line from the input file, and grabs the first7
    c     things from it, the first being an alpha command contained within
    c     (')and the rest being numbers. If less than 6 number are input
    c     the input line must be terminated by a (/)in order for the
    c     read statement to be correctly interpreted. The arguments are then
    c          placed in the common block"args". when the 'end' command is
    c          encountered, "success"is printed on the terminal, and the
    c          graphics program terminates.
    c
43          subroutine getl
44          common/args/arg1, arg2, arg3, arg4, arg5, arg6, arg7
45          character*10 arg1
46          real*8 arg2, arg3, arg4, arg5, arg6, arg7
47          read(5, *)arg1, arg2, arg3, arg4, arg5, arg6, arg7
48          if(arg1.eq.'end')then
49              call plot(0., 0., 999)
50              print*, 'success!!!'
51              stop
52          endif
53          return
54          end
    c
    c     proc
    c
    c     proc()does all the processing for a line. It gets its arguments
    c     from the common block args, and does it's thing
    c
55          subroutine proc
```

```
56          common/matrix/currm, view, curp
57          real*8 currm(4, 4), view(4, 4), curp(4)
58          common/args/arg1, arg2, arg3, arg4, arg5, arg6, arg7
59          character*10 arg1
60          real*8 arg2, arg3, arg4, rrg5, arg6, arg7
61          common/clip/hither, yon, dee
62          real*8 hither, yon, dee
63          common/cstack/cnum, cnump
64          integer cnum(10), cnump
65          integer i
66          integer*4rtimer, countt
    c
    c           determine the command entered(HUGE if-then-else if-, etc)and
    c           call the appropriate routine with the correct arguments
    c
67          if(argl.'eq.comment') then
68              i = 1
69      100     read(5, 800)
70              i = i + 1
71              if(i.le.int(arg2)) goto 100
72      800     format(al)
73              else if(argl.eq.'define') then
74              i = int(arg2)
75              call defn(i)
76              call printd(i)
77          else if(argl.eq.'call') then
78              cnum(cnump) = int(arg2)
79              cnump = cnump + 1
80              if(cnump.gt.10) then
81                  print*, 'call nesting level too deep, sorry'
82                  cnump = 10
83              endif
84              call callit(cnum(cnump - 1), cnump - 1)
85              cnump = cnump - 1
86          else if(argl.eq.'line') then
87              call pline(arg2, arg3, arg4, arg5, arg6, arg7, 2)
88          else if(arg1.eq.'plot) then
89              i = int(arg5)
90              call pplot(arg2, arg3, arg4, i)
```

```
      else if(argl.eq. 'ident') then
           call ident(currm)
      else if('argl.ep. 'push') then
           call push(currm)
      else if(argl.eq. 'pop') then
           call pop(currm)
      else if(argl.eq. 'rotate') then
           call rotate(arg2, arg3, arg4, currm)
      else if(argl.eq.'translate') then
           call transl(arg2, arg3, arg4, currm)
      else if(argl.eq. 'scale') then
           call pscale(arg2, arg3, arg4, currm)
      else if(argl.eq. 'window') then
           call window(arg2, arg3)
      else if(argl.eq. 'viewport') then
           call viewpn(arg2, arg3, arg4, arg5)
      else if(argl.eq. 'viewpoint') then
           call viewpn(arg2, arg3, arg4, arg5, arg6, arg7)
      else if(argl.eq. 'zclip') then
           call zclip(arg2, arg3)
      else if(argl.eq. 'cube') then
           call cube(arg2, arg3, arg4, arg5, arg6, arg7)
      else if(argl.eq. 'arrow') then
           call arrow
      else if(argl .eq. 'pyramid') 'then
           call pyrmd(arg2, arg3, arg4, arg5, arg6, arg7)
      else if(argl .eq. 'current') then
           call printm(currm)
      else if(argl .eq. 'printdef') then
           i = int(arg2)
           call printd(i)
      else if(argl .ep. 'startt') then
           call stimer
      else if(argl .eq. 'readt') then
           countt = rtimer( )
           print*, 'the time(in seconds) from the last startt is:', countt/100.
      else
           print*, 'error, command', argl, 'unknown'
      endif
```

```
130         return
131         end
      c
      c     ident(matrx)
      c
      c     ident() sets the given4 x 4 matrix to the idontity matrix.
      c
132         subroutine ident(matrx)
133         real*8 matrx(4, 4)
134         integer i,j
135         do 100i = 1, 4
136             do 100 j = 1, 4
137                 matrx(i, j) = 0.
138   100   continue
139         do 110 i = 1, 4
140             matrx(i,i) = 1.
141   110   continue
142         return
143         end
      c
      c     subroutine defn(number) defines figure number. the defined figure
      c         is contained in a large common block"defns"which contains
      c         enough space for a total of 500 commands. comments are not
      c         stored along with the define commands to save space. the variables
      c         entry and ends contain the starting and ending indexes of the
      c         10 possible definep figures
      c
144         subroutine defn(number)
145         integer number
146         common/defns/darg1, darg2, darg3, darg4, darg5, darg6, darg7, entry,tailp,
            ends
147         character*10 dargl(500)
148         real*8 darg2(500), darg3(500), darg 4(500), darg5(500), darg6(500), darg
            7(500)
149         integer entry(10), ends(10)
150         integer tailp
151         common/args/arg1, arg2, arg3, arg4, arg5, arg6, arg7
152         character*10 argl
153         real*8 arg2, arg3, arg4, arg5, arg6, arg7
```

```
154         integer i
155         entry(number) = tailp
156         print*, 'start of define is at', tailp
157   100   call getl
      c
      c     check for terminate of define
      c
158         if(argl .eq.'enddef') then
159             ends(number) = tailp
160             print*, 'end of figure define is at', tailp
161             return
162         else if(argl .ne. 'comment') then
163             dargl(tailp) = argl
164             darg2(tailp) = arg2
165             darg3(tailp) = arg3
166             darg4(tailp) = arg4
167             darg5(tailp) = arg5
168             darg6(tailp) = arg6
169             darg7(tailp) = arg7
170             tailp = tailp + 1
171             if(tailp .gt. 500) then
172                 print*, 'define memory overrun!!!'
173                 tailp = 500
174             endif
175         else
176                 i = 1
177   150   read(5, 800)
178                 i = i + 1
179                 if(i .le. int(arg2)) goto 150
180   800           format(al)
181         endif
182         goto 100
183         end
      c
      c     subroutine printd(number) prints the defined figure commands
      c
184         subroutine printd(number)
185         integer number
186         common/defns/dargl, darg2, darg3, darg4, darg5, darg6, darg7, entry,tailp,
```

```
            ends
187         character*10 dargl(500)
188         real*8 darg2(500)darg3(500)darg4 (500)darg5 (500), darg6(500), darg7
            (500)
189         integer entry(10), ends(10)
190         integer tailp
191         integer i
192         i = entry(number)
193   100   if(i .eq. ends(number))return
194         write(6, 800)dargl(i), darg2(i), darg3(i), darg4(i), darg5(i)darg6(i), da-
            rg7(i)
195   800   format(al0, 6f11.4)
196         i=i+1
197         goto100
198         end
      c
      c     subroutine callit(number, nest)causes the defined figure number to
      c          be input to the graphics processor, nesting level must be provided
      c     to allow pseudo-recursive type calls...
      c
199         subroutine callit(number, nest)
200         integer number, nest
201         common/defns/darg1, darg2, darg3, darg4, darg5, darg6, darg7, entry,tailp
            ends
202         character*10 dargl(500)
203         real*8 darg2(500), darg3(500), darg4(500), darg5(500), darg6(500), darg
            7(500)
204         integer entry(10), ends(10)
205         integer tailp
206         common/args/arg1, arg2, arg3, arg4, arg5, arg6, arg7,
207         character*10 arg1
208         real*8 arg2, arg3, arg4, arg5, arg6, arg7
209         integer i(10)
210         i(nest) = entry(number)
211   100   if(i(nest).eq. ends(number))return
212         arg1 = darg1(i(nest))
213         arg2 = darg2(i(nest))
214         arg3 = darg3(i(nest))
215         arg4 = darg4(i(nest))
```

```
216         arg5 = darg5(i(nest))
217         arg6 = darg6(i(nest))
218         arg7 = darg7(i(nest))
219         call proc
220         i(nest) = i(nest) + 1
221         goto 100
222         end
      c
      c     printm(matrx)
      c
      c     printm prints out the given 4 x 4 double precision matrix
      c
223         subroutine printm(matrx)
224         real*8 matrx(4, 4)
225         integer i
226         do 100 i = 1, 4
227             write(6, 800) matrx(i, 1), matrx(i, 2), matrx(i, 3), matrx(i, 4)
228   100   continue
229   800   format(4f15.4)
230         return
231         end
      c
      c     pline(x, y, z, a, b, c, s)
      c
      c     pline() draws a line from (x, y, z) to (a, b, c) with pencode s, using
      c     the current window, viewpoint, viewport, etc.
      c
  1         subroutine pline(x, Y, z, a, b, c, s)
  2         real*8x, y, z, a, b, c
  3         integer s
  4         common/matrix/currm, view, curp
  5         real*8 currm(4, 4), view(4, 4), curp(4)
  6         logical zclipp, junk
  7         real*8 tmpf(4), tmpt(4)
  8         tmpf(1) = x
  9         tmpf(2) = y
 10         tmpf(3) = z
 11         tempf(4) = 1.
 12         tmpt(1) = a
```

```
13          tmpt(2) = b
14          tmpt(3) = c
15          tmpt(4) = 1.
16          curp(1) = a
17          curp(2) = b
18          curp(3) = c
19          curp(4) = 1.
      c
      c     perform translations, anh viewing transl ation
      c
20          call mmultl(tmpf, currm, tmpf)
21          call mmultl(tmpt, currm, tmpt)
22          call mmultl(tmpf, view, tmpf)
23          call mmultl(tmpt, view, tmpt)
      c
      c     perform zclipping on both points...
      c
24          if(zclipp(tmpf, tmpt) .eq.. false.)goto 200
25          junk = zclipp(tmpt, tmpf)
      c
      c     project the vector into 2-D
      c
26          call projct(tmpf)
27          call projct(tmpt)
      c
      c     do x/y clipping, the window to viewport transform, and plot the vector
      c
28          call wtovp(tmpf, tmpt, s)
29    200   return
30          end
      c
      c     pplot(x, y, z, t)
      c
      c     plot a line from the current position to(x, y, z)using pencode t.
      c     Basically, sets up a call to pline from the current position
      c     to the new position using the appropriate pencode.
      c
31          subroutine pplot(x, y, z, t)
32          real*8 x, y, z
```

```
      integer t
      common/matrix/currm, view, curp
      real*8 currm(4, 4), view(4, 4), curp(4)
      call pline(curp(1), curp(2), curp(3), x, y, z, t)
      return
      end
c
c     push(matrix)
c
c     push() pushes the given matrix onto the matrix stack, checks
c     for stack overflow, and won't let you!!!! Does not alter the
c     curre: t matrix.
c
c
      subroutine push(matrik)
      real*8 matrix, sspace(4, 4, 10)
      integer stackp
      dimension matrix(4, 4)
      common/stacks/stackp, sspace
      if(stackp .gt. 10) then
          print*, 'stack overflow'
          return
      end if
      call copym(sspace(1, 1, stackp), matrix)
      stackp = stackp + 1
      return
      end
c
c     pop(matrix)
c
c     pop() pops the top of stack into the given matrix. Checks for
c     stack underflow, and again won't let you do it!!!!
c
      subroutine pop(matrix)
       real*8 matrix, sspace(4, 4, 10)
      integer stackp
      dimension matrix(4, 4)
      common/stacks/stackp, sspace
      stackp = stackp - 1
```

```
58          if(stackp .lt. 1) then
59              print*, 'stack underflow'
60          stackp = 1
61          return
62          end if
63          call copym(matrix, sspace(1, 1, stackp))
64          return
65          end
      c
      c     rotate(x, y, z, matrix)
      c
      c     rotate() pre-concatenates the given (x, y, z) rotation, to the
      c     supplied matrix (usually the current matrix). x, y, z are given
      c         in degrees.
      c
 1          subroutine rotate(x, y, z, matrix)
 2          real*8 x, y, z, matrix
 3          dimension matrix(4, 4)
 4          real*8 tmp
 5          dimension tmp(4, 4)
 6          call ident(tmp)
 7          tmp(2, 2) = cos(x*0.01745329)
 8          tmp(3, 3) = tmp(2, 2)
 9          tmp(2, 3) = sin(x*0.01745329)
10          tmp(3, 2) = - tmp(2, 3)
11          call mmult4(tmp, matrix, matrix)
12          call ident(tmp)
13          tmp(1, 1) = cos(y*0.01745329)
14          tmp(3, 3) = tmp(1, 1)
15          tmp(3, 1) = sin(y*0.01745329)
16          tmp(1, 3) = - tmp(3, 1)
17          call mmult4(tmp, matrix, matrix)
18          call ident(tmp)
19          tmp(1, 1) = cos(z*0.01745329)
20          tmp(2, 2) = tmp(1, 1)
21          tmp(1, 2) = sin(z*0.01745329)
22          tmp(2, 1) = - tmp(1, 2)
23          call mmult4(tmp, matrix, matrix)
24          return
```

```
25          end
      c
      c     translate(x, y, z, matrix)
      c
      c     translate() pre-concatenates the given tranlation(x, y, z) to the
      c     given matrix(usually the current matrix).
      c
26          subroutine transl(x, y, z, matrix)
27          real*8 x, y, z, matrix
28          dimension matrix(4, 4)
29          real*8 tmp
30          dimension tmp(4, 4)
31          call ident(tmp)
32          tmp(4, 1) = x
33          tmp(4, 2) = y
34          tmp(4, 3) = z
35          call mmult4(tmp, matrix, matrix)
36          return
37          end
      c
      c     pscale(x, y, z, matrix)
      c
      c     pscale pre-concatenates the given scaling(x, y, z) onto the
      c     given matrix.
      c
38          subroutine pscale(x, y, z, matrix)
39          real*8 x, y, z, matrix
40          dimension matrix(4, 4)
41          real*8 tmp
42          dimension tmp(4, 4)
43          call ident(tmp)
44          tmp(1, 1) = x
45          tmp(2, 2) = y
46          tmp(3, 3) = z
47          call mmult4(tmp, matrix, matrix)
48          return
49          end
      c
      c     window(a, b)viewport(a, b, c, d)
      c
      c     these two routines set up the global variables according to the
```

```
      c       given parameters.
      c
50            subroutine window(a, b)
51             real*8 a, b
52            real*8 wxh, wyh
53            common/windoe/wxh, wyh
54            wxh = a
55            wyh = b
56            return
57            eud
      c
58            subroutine viewpr(a, b, c, d)
59            real*8 a, b, c, d
60            real*8 vxh, vyh, vxc, vyc
61            common/viewp/vxh, vyh, vxc, vyc
62            vxc = a
63            vyc = b
64            vxh = c
65            vyh = d
66            call mplot(vxc - vxh, vyc - vyh, 3)
67            call mplot(vxc + vxh, vyc - vyh, 2)
68            call mplot(vxc + vxh, vyc + vyh, 2)
69            call mplot(vxc - vxh, vyc + vyh, 2)
70            call mplot(vxc - vxh, vyc - vyh, 2)
71            return
72            end
      c
      c       viewpoint(a, b, c, d, e, f)
      c
      c       viewpoint sets up the viewing transformation for the given
      c       to and from points - - - the eye position is(a, b, c) the lookat
      c       position is(d, e, f).
      c
1             subroutine viewpn(a, b, c, d, e, f)
2             real*8 a, b, c, d, e, f
3             real*8 angle
4             real*8 tmp(4, 4), tmpp(4)
5             common/matrix/currm, view, curp
6             real*8 currm(4, 4), view(4, 4), curp (4)
```

```
      common/clip/hither, yon, dee
      real*8 hither, yon, dee
c
c          initialize the viewing transformation
c
      call ident(view)
c     move lookat position to origin
c
      call transl(-d, -e, -f, view)
c
c          rotate view matrix per the lookat angle
c
      a = a - d
      b = b - e
      c = c - f
      angle = -atan2(a, c)
      call ident(tmp)
      tmp(1, 1) = cos(angle)
      tmp(3, 3) = tmp(1, 1)
      tmp(3, 1) = sin(angle)
      tmp(1, 3) = -tmp(3, 1)
      call mmult4(view, tmp, view)
      angle = atan2(b, sqrt(a*a + c*c))
      call ident(tmp)
      tmp(2, 2) = cos(angle)
      tmp(3, 3) = tmp(2, 2)
      tmp(2, 3) = sin(angle)
      tmp(3, 2) = -tmp(2, 3)
      call mmult4(view, tmp, view)
      a = a + d
      b = b + e
      c = c + f
      tmpp(1) = a
      tmpp(2) = b
      tmpp(3) = c
      tmpp(4) = 1.
      call mmult1(tmpp, view, tmpp)
      dee = tmpp(3)
      return
      end
```

```
c
c       zclip(a, b)
c
c       zclip()sets up the global clipping parameters, a is the hither,
c       b the yon, does not allow the hither plane to be behind the
c       viewer, nor does it allow the yon to be between the viewer
c       and the hither
c
39      subroutine zclip(a, b)
40      real*8 a, b
41      real*8 hither, yon, dee
42      common/clip/hither, yon, dee
43      if(a .lt. 0)then
44          print*, 'bad hither parameter'
45          a = 0
46      end if
47      if(b .lt. a)then
48          print*, 'bad yon parameter'
49          b = a + 100
50      end if
51      hither = a
52      yon = b
53      return
54      end
c
c       zclipping(vect 1, vect2)
c
c       zclipping()performs the zclipping on vect1 using the global
c       zclipping parameters. Modifies ONLY vect1, returns true if
c       a portion of the vector indicated by (clipped)vect1 and vect2
c       will be visible in the scene.
c
55      logical function zclipp(vect1, vect2)
56      real*8 vect1(4), vect2(4)
57      common/clip/hither, yon, dee
58      real*8 hither, yon, dee
59      real*8 htr, yn
60      htr = dee - hither
61      yn = dee - yon
```

```
62          zclipp = .true.
63          if(vect1(3).gt. htr)then
64              if(vect2(3).gt. htr)then
65                  zclipp = .false.
66              else
    c
    c       yon must modify the x and y parameters(according to like triangles)
    c       when the z parameter is modified!!!
    c
67          *       vect1(1) = (vect1(1) - vect2(1))* ((htr - vect2(3))/
                            (vect1(3) - vect2(3))) + vect2(1)
68              vect1(2) = (vect1(2) - vect2(2))*((htr - vect2(3))/
            *               (vect1(3) - vect2(3))) + vect2(2)
69              vect1(3) = htr
70              zclipp = .true.
71          end if
72          else if(vect1(3).lt. yn)then
73              if(vect2(3).lt. yn)then
74                  zclipp = .false.
75          else
76              vect1(1) = (vect2(1) - vect1(1))*((yn - vect1(3))/
        *                   (vect2(3) - vect1(3))) + vect1(1)
77              vect1(2) = (vect2(2) - vect1(2))*((yn - vect1(3))/
        *                   (vect2(3) - vect1(3))) + vect1(2)
78              vect1(3) = yn
79              zclipp = .true.
80          end if
81          end if
82          return
83          end
    c
    c       projct(vector)
    c
    c       projct() projects the given vector to a point in 2-D space using
    c       the global"dee"parameter, for single point perspective.
    c
1           subroutine projct(vector)
2           real*8 vector(4)
3           common/clip/hither, yon, dee
```

```
4         real*8 hither, yon, dee
5         real*8 tmp(4, 4)
6         call ident(tmp)
7         if(dee .ne. 0)then
8         tmp(3, 4) = -1/dee
9         else
10              tmp(3, 4) = -1000000000.
11        endif
12        call mmult1(vector, tmp, vector)
13        call norm(vector)
14        return
15        end
      c
      c   norm(vector)
      c
      c   norm()normalizes the given vector
      c
      c
16        subroutine norm(vector)
17        real*8 vector(4)
18        vector(1) = vector(1)/vector(4)
19        vector(2) = vector(2)/vector(4)
20        vector(3) = vector(3)/vector(4)
21        vector(4) = 1.
22        return
23        end
      c
      c   wtovp(from, to, pencode)
      c
      c   wtovp()takes the projected from and to points, and:
      c       1,does x/y clipping on the window
      c       2,does the window to viewport translation
      c       3,plots tne transformed points onto the device
      c
24        subroutine wtovp(from, to, pencde)
25        real*8 from(4), to(4)
26        integer pencde
27        common/windoe/wxh, wyh
28        real*8 wxh, wyh
```

```
29          common/viewp/vxh, vyh, vxc, vyc
30          real*8 vxh, vyh, vxc, vyc
31          logical xyclip
32          real*8 xp, yp
33          if(xyclip(from, to))then
34               xp = (from(1))*vxh/wxh + vxc
35               yp = (from(2))*vyh/wyh + vyc
36               call mplot(xp, yp, 3)
37               xp = (to(1))*vxh/wxh + vxc
38               yp = (to(2))*vyh/wyh + vyc
39               call mplot(xp, yp, pencde)
40          endif
41          return
42          end
      c
      c     xyclip(from, to)
      c
      c     xyclip() performs the x/y clipping on both. the from and t
      c     vectors in the window cooridinates. Returnes false if
      c     none of the vector would be visible.
      c
43          logical function xyclip(from, to)
44          real*8 from(4), to(4)
45          integer*2 cf, ct
46          xyclip = .false.
47    100        call code(from, cf)
48               call code(to, ct)
49          if((cf .and. ct).ne. 0)goto 105
50               if(cf.ne.0) call ppush(cf, from, to)
51               if(ct .ne. 0) call ppush(ct, to, from)
52          if((cf + ct).ne. 0) goto 100
53          xyclip = .true.
54    105   return
55          end
      c
      c     code(vector, flag)
      c
      c     code() returns the binary code in flag for vector indicating
      c     it's position relative to the window.
```

```
c
      subroutine code(vector, flag)
      real*8 vector(4)
      integer flag
      common/windoe/wxh, wyh
      real*8 wxh, wyh
      real*8 tmp
      flag = 0
      tmp = vector(1)
      if(tmp .lt. -wxh)flag = 1
      if(tmp .gt. wxh)flag = flag + 2
      tmp = vector(2)
      if(tmp .lt. -wyh)flag = flag + 4
      if(tmp .gt. wyh)flag = flag + 8
      return
      end
c
c     ppush(flag, to, from)
c
c     ppush() pushes"to"towards"from"according to flag, which
c     contains the code returned by code() .used to insure that the
c         line exits the window at the correct point
c
      subroutine ppush(flag, to, from)
      real*8 to(4), from(4)
      integer, flag
      common/windoe/wxh, wyh
      real*8 wxh, wyh
      if((flag .and. 1).ne. 0)then
          to(2) = ((-wxh-from(1))
     *              /(to(1)-from(1)))*(to(2)-from(2))+from(2)
          to(1) = -wxh
      endif
      if((flag .and. 2).ne. 0)then
          to(2) = ((wxy-from(1))
     *              /(to(1)-from(1)))*(to(2)-from(2))+from(2)
          to(1) = wxh
      endif
      if((flag .and. 4).ne. 0)then
```

```
30              to(1) = (( - wyh - from(2))
     *                  /(to(2) - from(2)))*(to(1) - from(1)) + from(1)
31              to(2) = - wyh
32          endif
33          if((flag .and. 8) .ne. 0)then
34              to(1) = ((wyh - from(2))
     *                  /(to(2) - from(2)))*(to(1) - from(1)) + from(1)
35              to(2) = wyh
36          endif
37          return
38          end
    c
    c       copym(dst, src)
    c
    c           copym() copies the src 4×4 matrix to the dst 4×4 matrix.
    c
39          subroutine copym(dst, src)
40          real*8 dst(16), src(16)
41          integer i
42          do 100 i = 1, 16
43          dst(i) = src(i)
44  100     continue
45          return
46          end
    c
    c       mplot(arg1, arg2, arg3)
    c
    c       mplot() calls plot with arg1, arg2, arg3 .inserted as another level
    c           of indirection in order to allow the actual plot commands to be
    c           written to a file, etc.
    c
47          subroutine mplot(arg1, arg2, arg3)
48          real*8 arg1, arg2
49          integer arg3
50          call plot(arg1, arg2, arg3)
51          return
52          end
    c
    c       cube(arg1, arg2, arg3, arg4, arg5, arg6)
```

```
c
c       cube()generates a cube centered at (arg1, arg2, arg3)with
c       arg4, arg5, arg6 as it's half widths
c
1       subroutine cube(arg1, arg2, arg3, arg4, arg5, arg6)
2       real*8 arg1, arg2, arg3, arg4, arg5, arg6
3       call pline(arg1 - arg4, arg2 - arg5, arg3 - arg6, arg1 + arg4, arg2 - arg5, arg
        3 - arg6, 2)
4       call pplot(arg1 + arg4, arg2 + arg5, arg3 - arg6, 2)
5       call pplot(arg1 - arg4, arg2 + arg5, arg3 - arg6, 2)
6       call pplot(arg1 - arg4, arg2 - arg5, arg3 - arg6, 2)
7       call pplot(arg1 - arg4, arg2 - arg5, arg3 + arg6, 2)
8       call pplot(arg1 + arg4, arg2 - arg5, arg3 + arg6, 2)
9       call pplot(arg1 + arg4, arg2 + arg5, arg3 + arg6, 2)
10      call pplot(arg1 - arg4, arg2 + arg5, arg3 + arg6, 2)
11      call pplot(arg1 - arg4, arg2 - arg5, arg3 + arg6, 2)
12      call pline(arg1 + arg4, arg2 - arg5,arg3 - arg6,arg1 + arg4,arg2 - arg5,arg3 +
        arg6, 2)
13      call pline(arg1 + arg4, arg2 + arg5,arg3 - arg6,arg1 + arg4,arg2 + arg5,arg3 +
        arg6, 2)
14      call pline(arg1 - arg4, arg2 + arg5,arg3 - arg6,arg1 - arg4,arg2 + arg5,arg3 +
        arg6, 2)
15      return
16      end
c
c
c       arrow()
c
c       arrow()draws a sort-of arrow from(0, 0, 0)to(1, 0, 0)
c
17      subroutine arrow()
18      call pline(0., 0., 0., 1., 0., 0., 2)
19      call pline(1., 0., 0., .8, .2, 0., 2)
20      call pline(1., 0., 0., .8, 0., .2, 2)
21      call pline(1., 0., 0., .8, - .2, 0., 2)
22      call pline(1., 0., 0., .8, 0., - .2, 2)
23      return
24      end
c
```

```
c       pyrmd(arg1, arg2, arg3, arg4, arg5, arg6)
c
c      pyrmp()draws a pyramid with the center of it's base at
c       (arg1, arg2, arg3)and half x, y, z widths of arg4, arg5, arg6.
c       The height is the x half width.
c
      subroutine pyrmd(arg1, arg2, arg3, arg4, arg5, arg6)
      real*8 arg1, arg2, arg3, arg4, arg5, arg6
      ealr*8 height
      call pline(arg1 - arg4, arg2 - arg5,arg3 - arg6,arg1 + arg4, arg2 - arg5,arg3 -
     arg6, 2)
      call pplot(arg1 + arg4, arg2 + arg5, arg3 - arg6, 2)
      call pplot(arg1 - arg4, arg2 + arg5, arg3 - arg6, 2)
      call pplot(arg1 - arg4, arg2 - arg5, arg3 - arg6, 2)
      height = arg4 - arg1
      call pline(arg1 - arg4, arg2 - arg5, arg3 - arg6, arg1, arg2, arg3 + height, 2)
      call pline(arg1 + arg4, arg2 - arg5, arg3 - arg6, arg1, arg2, arg3 + height, 2)
      call pline(arg1 - arg4, arg2 + arg5, arg3 - arg6, arg1, arg2, arg3 + height, 2)
      call pline(arg1 + arg4, arg2 + arg5, arg3 - arg6, arg1, arg2, arg3 + height, 2)
      return
      end
c
c           subroutine mmult4(mp1, mp2, mpr)
c
c           subroutine mmult4 multiplies the mp1 4 × 4 matrix
c           and multiplies it by the mp2 4 × 4 matrix. the result is
c           placed in the mpr 4 × 4 matrix. internal results are placed
c           in a temporary matrix, then copied over in order that one of
c           the operands may be used as the destination matrix
c
      subroutine mmult4(mp1, mp2, mpr)
      real*8 mp1(4, 4), mp2(4, 4), mpr(4, 4)
      real*8 acc
      real*8 temp(4, 4)
      integer i, j, k
      do 100 i = 1, 4
          do 100 j = 1, 4
              acc = 0.
              do 110 k = 1, 4
```

```
10                      acc = acc + mpl(i, k)*mp2(k, j)
11  110             continue
12                  temp(i, j) = acc
13  100     continue
14          do 120 i = 1, 4
15              do 120 j = 1, 4
16                  mpr(i, j) = temp(i, j)
17  120     continue
18          return
19          end
    c
    c       subroutine mmultl(mpl, mp2, mpr)
    c
    c       subroutine mmultl multiplies the mpl 4 position vector
    c       by the mp2 4 x 4 matrix. the result is put in the mpr 4
    c       position vector. results are calculated into a temporary
    c       vector, then copied over so that the mpl vector may be used
    c       as the destination of the result
    c
20          subroutine mmultl(mpl, mp2, mpr)
21          arel*8 mpl(4), mp2(4, 4), mpr(4)
22          real*8 acc
23          real*8 temp(4)
24          integer i, j, k
25          do 100 j = 1, 4
26              acc = 0.
27              do 110 k = 1, 4
28                      acc = acc + mpl(k)*mp2(k, j)
29  110         continue
30              temp(j) = acc
31  100     continue
32          do 120 i = 1, 4
33          mpr(i) = temp(i)
34  120     continue
35          return
36          end
    c
    c       subroutine plot(x, y, penc)
    c
```

```
c          subroutine plot plots a line from the current pen position to
c          the given pen position using the pencode given. The possible
c          pen codes are;
c               2;pen down
c               3;pen up
c               999;terminate plotting
c          the actual interface described here if for the serial port on the
c          iSBC 86/12a board connected to an Hp7225A flat bed plotter. no
c          handshaking is done.
c
      subroutine plot(x, y, penc)
      real*8 x, y
      integer penc
      common/penpos/xpos, ypos, pcount
      real*8 xpos, ypos
      integer*4 pcount
      pcount = pcount + 1
      if(penc .eq. 999)then
           call putout('p')
           call putout('U')
           call putout(';')
           print*, 'the number of points plotted is;', pcount
           goto 200
      endif
      if((xpos.eq. x) .and. (ypos .eq. y)) then
           if(penc .eq. 2)then
                call putout('p')
                call putout('D')
                call putout(';')
                call putout('p')
                call putout('U')
                call putout(';')
           else
                goto 200
          endif
      else
           if(penc .eq. 3)then
                call putout('p')
                call putout('U')
```

```
30              else if(penc .eq. 2)then
31                  call putout('p')
32                  call putout('D')
33                  else
34                  call putout('p')
35                  call putout('U')
36                  call putout(';')
37                  goto 200
38              endif
39              call putout(';')
40              call putout('p')
41              call putout('A')
42              if(x .gt. 12)x=12
43              call ponum(x)
44              call putout(',')
45              if(y.gt. 10)y=10
46              call ponum(y)
47              call putout(';')
48          endif
49          xpos=x
50          ypos=y
51  200     return
52          end
    c
    c       subroutine ponum(number)
    c
    c           subroutine ponum takes the given double precision real number,
    c           truncates it to integer,then runs the resultant integer out
    c           the iSBC 86/12a serial port. leading zeros are suppressed.
    c           the maximum number is 99999!!!
    c
53          subroutine ponum(number)
54          real*8 number
55          character lookup(9)
56          logical flag
57          integer multip(5)
58          integer work
59          data lookup)'1','2','3','4','5','6','7','8','9'/
60          data multip/10000, 1000, 100, 10, 1/
```

```
61          flag = .false.
62          if(number.lt. 0)number = 0.
63          number = number*800.
64          do 100 i = 1, 5
65          work = aint(number/real(multip(i)))
66          if(work .eq. 0)then
67              if(flag)call putout('0')
68          else
69              call putout(lookup(work))
70              flag = .true.
71          endif
72          number = number - work*multip(i)
73  100     continue
74          if(.not. flag)call putout('0')
75          return
76          end
    c
    c       subroutine plots
    c
    c           subroutine plots initialized the iSBC86/12 board baud rate
    c           generator(really part of the 8253 timer)and serial line.
    c           the given numbers will set it up for 600 baud, 8 bits, no
    c           parity
    c
77          subroutine plots
78          common/penpos/xpos, ypos
79          real*8 xpos, ypos
80          xpos = 10000.
81          ypos = 10000.
82          call output(#0d6h, intl(#0b6h))
83          call output(#0d4h, intl(#80h))
84          call output(#0d4h, intl(0))
85          call output(#0dah, intl(#72h))
86          call wastet
87          call output(#0dah, intl(#25h))
88          call wastet
89          call output(#0dah, intl(#62h))
90          call wastet
91          call output(#0dah, intl(#0ceh))
```

```
 92          call wastet
 93          call output(#0dah, intl(#27h))
 94          return
 95          end
     c
     c      subroutine putout(c)
     c
     c          subroutine putout puts the character given out on the iSBC 86/12
     c          board serial line(checks for transmitter empty, loops on not empty,
     c          on empty puts out the character)
     c
 96          subroutine putout(c)
 97          character c
 98          integer*1 status
 99  100         call input (#0dah, status)
100              status = status.and. 4
101          if(status.eq. 0)goto 100
102          call output(#0d8h, intl(ichar(c)))
103          return
104          end
     c
     c      subroutine wastet
     c
     c          subroutine wastet wastes a little bit of time while the 8253 gets
     c          its act together
     c
105          subroutine wastet
106          return
107          end
```

五、两个例子

下面是一段显示命令，它画出图 33-17 所示的 4 个图形。图 33-18 是从三个角度观察一架航天飞机得到的图象，最下面一个是从飞机内部朝外看。整架飞机是分批输入数据表示的零部件的合成，譬如，机翼数据只输入一次，另一机翼由这组数据的旋转、平移得到，发动机喷管由一组被标度过的圆组成，而每个圆又由 4 个张角为 90° 的扇形拼成。

```
run:f5:graph
'define'1/
'viewport'2.5 2.5 2 2/
'viewpoint'10 10 10 0 0 0/
'window'10 10/
'cube'0 0 0 2 2 2/
```

```
'viewport'7.5 2.5 2 2/
'rotate'15 15 15/
'cube'0 0 0 2 2 2/
'viewport'2.5 7.5 2 2/
'ident'/
'viewpoint'10 0 0 0 0 0/
'cube'0 0 0 2 2 2/
'viewport'7.5 7.5 2 2/
'rotate'30 30 30/
'cube'0 0 0 2 2 2/
'enddef'/
'call'1/
'end'/
```

显示命令

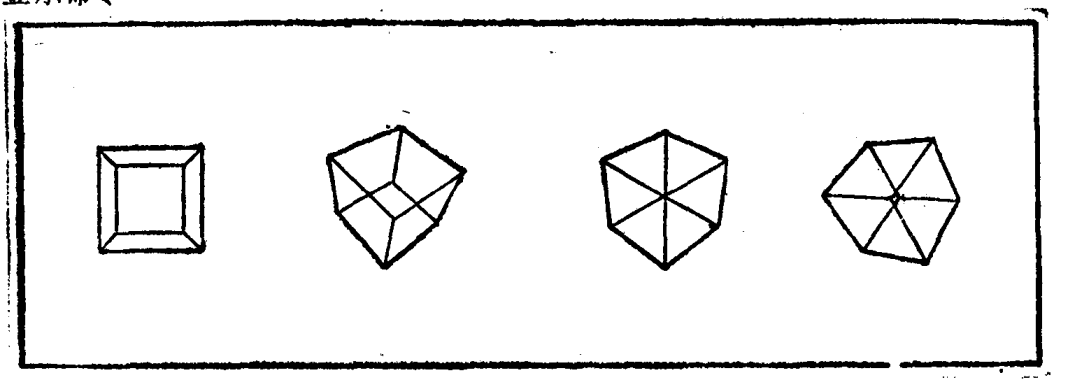


图 33-17

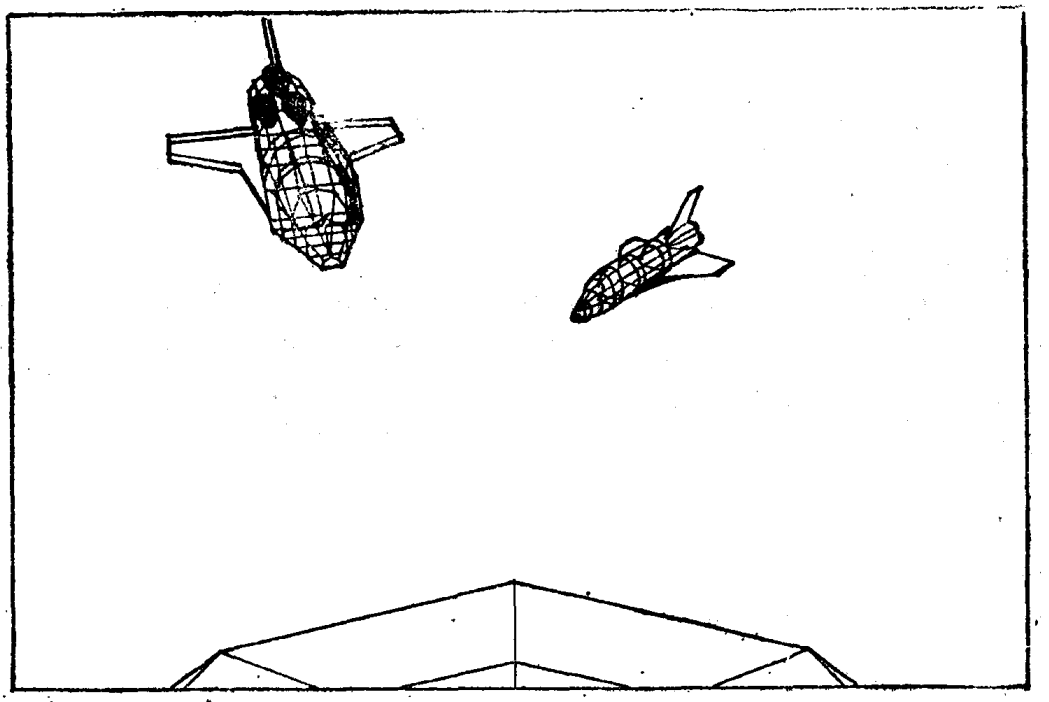


图 33-18

§33.2 以 IBM PC 为结点机的以太局部网络 Etherseries

局部网络是一种在中等规模地理区域内，例如一座办公楼、一个仓库、或一所学校中连接数台计算机及外部设备的数据通讯网，它具有中等到较高数传率的物理通信信道，该信道的误码率较低。由于微机系统的特点，在一个有限的区域内安置多台个人计算机是完全可以做到的，因而用局部网络把它们连成一个具有资源共享、信息访问、个人通讯等优点的整体是现实的、有意义的。

一、Ethernet 局部网络简介

Ethernet(以太网)是 Xerox 公司 1975 年开始研制的一个实验性局部网络，它的技术思想基本代表了局部网络的发展方向，后来出现的一些局部网络在网络配置、协议等技术方面都向 Ethernet 靠拢。Ethernet 在其开发及成熟的进程中，得到各大公司的支持。譬如 80 年 Intel 公司、Xerox 公司、DEC 公司联合宣布了 Ethernet 技术规范。

Ethernet 以基带同轴电缆作为通信的媒体，网络拓扑结构为总线式，传输速率为 10 兆位/秒。

Ethernet 的一般结构如图 33-19 所示，其中每个结点都可是一台独立工作的计算机或智能设备，如受计算机控制的磁带机、打印机、显示终端等，它们通过接口控制器。收发器同电缆连接。每段同轴电缆的长度不超过 500 米，加了中转器后不超过 2500 米，每个段最多容纳 100 个工作站，一个局部网络能拥有 1000个工作站。

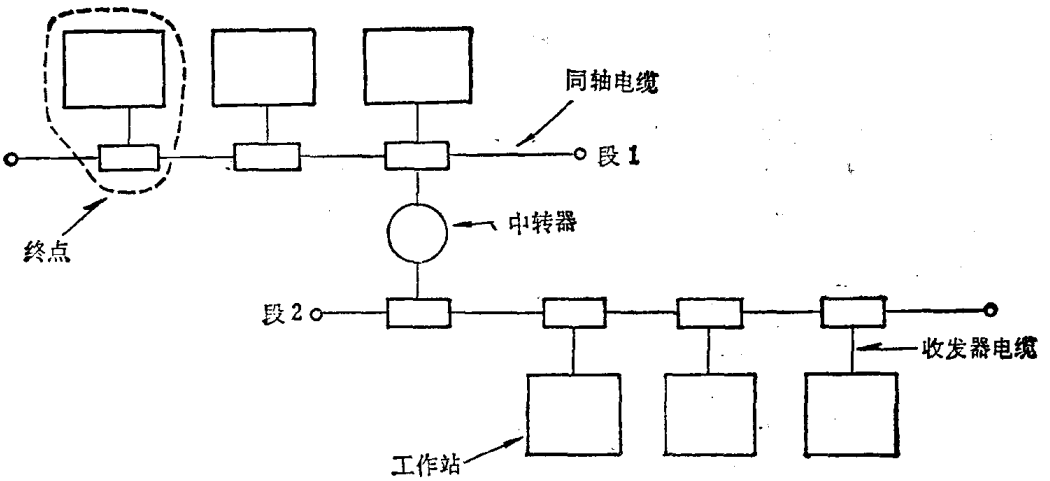


图 33-19 Ethernet 的结构

Ethernet 技术中的物理层主要由硬件实现，它提供一条与通讯介质有关的物理信道。物理层的主要功能部件是：同轴电缆、收发器、连接插座、信道时钟、数据编码器及译码器、信息前置段的产生和拆除装置、载波和冲突检测电路。物理层的主要功能有两个：一是产生及拆除前置段，为数据位编码、译码；二是信道访问，即数据位的传送和接收、载波识别和冲突

检测。

Ethernet 的数据链路层部分由硬件和部分软件实现，它在物理信道上提供一条与通讯介质无关的数据链路，其主要功能是：1．数据包装。2．链路管理。

数据链路层向用户层提供的服务主要有帧的传递和接收，高层差错恢复用的状态信息。Ethernet 各层作用见图 33-20。

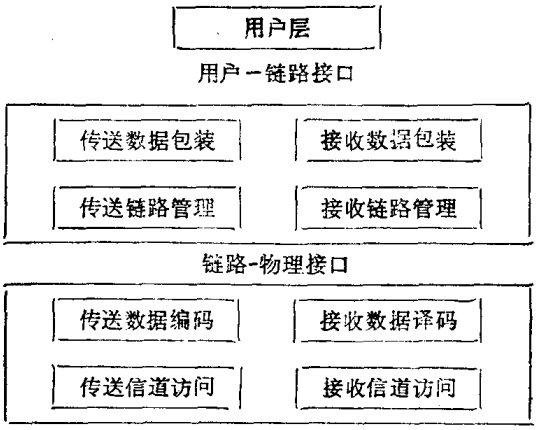


图 33-20　Ethernet 的层间接口

二、Ethernet 工作原理

Ethernet 的工作原理可概括为：分布式控制方案，高速的广播信息传输，多址访问、竞争发送与无冲突接收(CSMA/CD-Carrier Sense Multiple Access/collision Detect)。

1．分布式控制

与一般的集中控制式网络不同，Ethernet 没有设中心控制机，控制分散在各工作站上，每个结点都由一具微处理机的智能接口独立地进行数据传输。

2．广播式的信息传输方式

每个工作站所发送的信息都沿同轴电缆传播，所有站都可“听”到信息的传输，但只有站地址与信息传输目的地址相符合的工作站才能接收信息。

3.多址访问

信息在电缆上是以数据包为单位传输的，每一数据包组成一个帧。Ethernet 中帧的格式是固定的，长度可依数据量多少有所不同，见图 33-21。

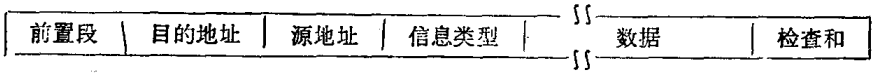


图 33-21　Ethernet 帧的格式

每帧由五个字段组成。除数据段长度可变外，其余四段的长度都是固定的。帧的地址段有 48 位，包括两种类型：一是单宿地址，指示网络中某一确定站的地址；另一种是多宿地址。其寻址又分为两种情况：一种是分组地址，它标识逻辑上相关的一组结点；另一种是广播地址，标识网内所有的结点。寻址时，用目的地址的第一位分类，为“0”表示其余 47 位是单宿地址，，为“1”表示其余 47 位是多宿地址。若 47 位全部是“1”，则指明广播地址。

4．竞争发送

Ethernet 中的每个工作站都主动争取发送信息，以提高电缆的利用率，但若同一时刻内许多站都试图发送信息的话，就会发生冲突现象，导致接收数据的失效和错误。因此在发送前每个站都先对电缆进行监视，若当前没有信息在传输，则开始本站的发送，若有信号在传输，则本站等待一段时间，待电缆空闲后开始发送。

本工作站发送过程开始后，还必须监视电缆，防止其它站的冲突干扰，此即冲突检测。若无冲突干扰就继续发送；若有冲突干扰则对电缆实行干预（继续发送一小段时间使所有的工作站都能发现冲突而使传送停止），而后按一定的算法得出的随机延迟时间等待一会儿，再重新发送。

5．无冲突接收

冲突检测只在发送站进行，一旦有冲突发生，接收站便把已收到的信息作为错误信息舍弃，随后进入准备接收状态。冲突的干预就是由发送站发出一个长度小于最小数据包的不完全帧（32 位），接收站用它判断冲突是否发生了。

6．应答式信息传输方式

为使信息传输可靠，Ethernet 采用如下应答方式：

(1) 接收站每接收完一个数据包，就向发送站发一个 ACK，回答确认数据包，告诉发送方信息已正常接收了。

(2) 如果接收的信息有错，接收站发出 NAC 数据包至发送站，表示要求复发。

(3) 每当一个由若干数据包组成的消息发送完毕，发送站再附送一个结束数据包至接收站，表示发送结束。

(4) 接收站收到结束信号时，再向发送站回送一个回答数据包，表示了解此情况。

三、Etherseries 的结构

Eherseries 是 3 cm 公司的一种开发产品，是一种以 IBM PC 为结点机的 Ethernet 局部网。由于 Etherseries 的工作原理基本与前一节介绍的 Ethernet 一样，所以在此忽略，下面简要介绍 Etherseries 因 IBM PC 的引入而具有的特点。

Etherseries 网（图 33-22）的结点，可以是独立工作的用户IBM PC，或是称作服务站的共享设备，服务站中还设有一台管理共享设备及网络通讯的服务计算机，服务计算机由 IBM PC、AP 或 VAX 机担任。以 IBM PC、IBM PC/XT、IBM PC 兼容机为服务计算机的服务站，能使局部网中的用户 PC 共享磁盘机、打印机；以 AP 机为服务计算机的服务站，能提供 30 兆硬盘，打印机、磁带机等服务；以 VAX 为服务计算机的服务站，以 Unix 作操作系统，服务程序与一般的 VAX/Unix 操作系统共存。

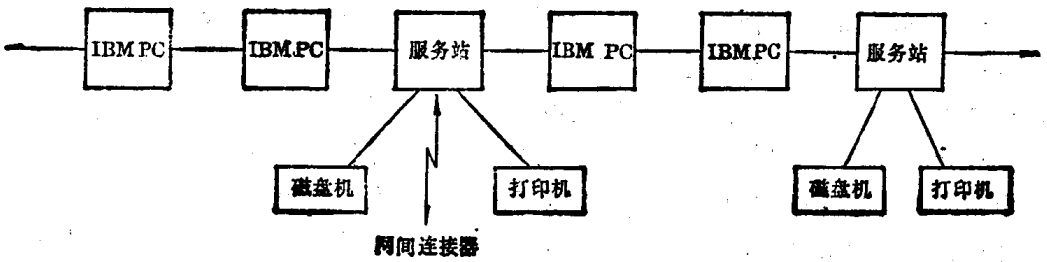


图 33-22 Etherseries 结构图

Etherseries 降低了对同轴电缆的要求，实际连网时，可用 50Ω 标准 RG-58 同轴电缆，这样局部网的造价就不高了。

在与用户 PC 的接口部分，Etherseries 也有其独到之处。Etherseries 把 Ethernet 的物理层、数据链路层做在一块可插入 IBM PC 槽口的电路板上，这块电路板容纳了所有连网用的硬件、软件，如收发器、数据链路控制器，同轴电路直接与 IBM PC 相接。

四、Etherseries 接口软件达到的服务程度

建立局部网络的目的，是使网络的用户结点能共享全网的磁盘及打印机等外部设备，以电子邮包形式与其他用户结点通讯，并且这种网间联络不应对应用程序甚至操作系统产生太大的影响，譬如，某个结点机的用户或应用程序在不知道共享磁盘机物理位置的情况下，就能访问远处的磁盘文件。让局部网络对应用程序或操作系统透明的要求是不苛刻的，它是使局部网络真正发挥作用、让用户满意的重要因素。

Etherseries 网的透明性目标，是通过其长约 85000 行的代码程序达到的。这项工作分两步进行。第一步是让局部网络隐含地影响应用程序，即网络对应用程序透明，这时网络程序与操作系统 (MS-DOS) 以某同等的身份支持，它们之间可能会有相互作用；第二步是让操作系统 MS-DOS 与网络软件的等同关系变成上下级关系，使网络对操作系统的命令如 PRINT、DIR、COPY 或整个操作系统透明，网络 IBM PC 上的用户，既可对机器携带的设备发操作命令，又可通过网络对位于它处的共享设备发操作命令，并且这些命令对操作系统来说没有什么区别。达到这种透明度的局部网络，对操作系统的影响已降到极小的程度，用户能放心地改用更高级的、向下兼容的操作系统，如 MS-DOS V. 2.0。

下面举一个 MS-DOS 发出磁盘访问的例子。(参见图 33-23)

网络的每个结点 PC 都为其配备的磁盘机以及远程共享设备分配标识名，如 A:、B:、C: 表示结点 PC 磁盘机，D:、E:、F:、G: 表示共享磁盘机 (也称虚拟磁盘机)。应用程序或用户发出的访问命令均为磁盘机标识名、文件名、记录名形式，MS-DOS 操作系统收到命令后，查带特权的磁盘文件表，算出该记录的逻辑扇区号，然后把逻辑扇区号送给磁盘驱动程序，由它把扇区号转换成磁头/道/物理扇区信息，让磁盘机完成访问。如果 MS-DOS 查出标识名是虚拟磁盘机名，则把控制转给网络软件中的 Ether 共享驱动程序，由该驱动程序把逻辑号通过 Ether 网发给服务站，服务站收到扇区号后，加以适当的处理、访问文卷，回送文件记录或整个文件。

由于 Etherseries 的通讯协议是非专用 Xerox 网络系统(XNS)协议，所以建立在它上面的数据包交换协议，允许用户以磁盘读或写请求访问远程共享设备，接收共享设备回送的数据或它对写操作的回答。

最后一个问题，是怎样把虚拟磁盘机上的文卷，分配给结点 PC。Etherseries 用其软件提供的辅助命令，来解决这个问题。

Etherseries 网中的每个结点 PC，都被指派一个服务站，使它拥有一个带名的文卷集，PC 虚拟驱动程序能动态地与集中的任何文卷集相连。访问文卷分两步进行：先用 LOGIN 命令定下指派给用户 PC 的服务站位置，建立用户的个人文卷域，呈示用户使用打印机或电子邮件系统时需要的标志。然后，用 LINK 命令把文卷纳入结点 PC 虚拟驱动程序的管辖范围，这相当于把磁盘插入 PC 的软盘机。UNK 命令执行完后，结点 PC 就能访问远程共

享设备了。利用 LINK 命令，结点 PC 用户可将他人的文卷调归自己虚拟驱动程序管理，享用他人的文件。共享设备涉及到文卷保护，并发控制等问题，Etherseries 对此设置了一个并发—安全系统，规定多种文卷类型，如只允许一个用户访问的私人文卷，允许多用户读，一个用户写的公共文卷，允许任何用户读写的共享文卷。E herseries 还用信号灯同步不同 PC 对某个文件的并发访问。

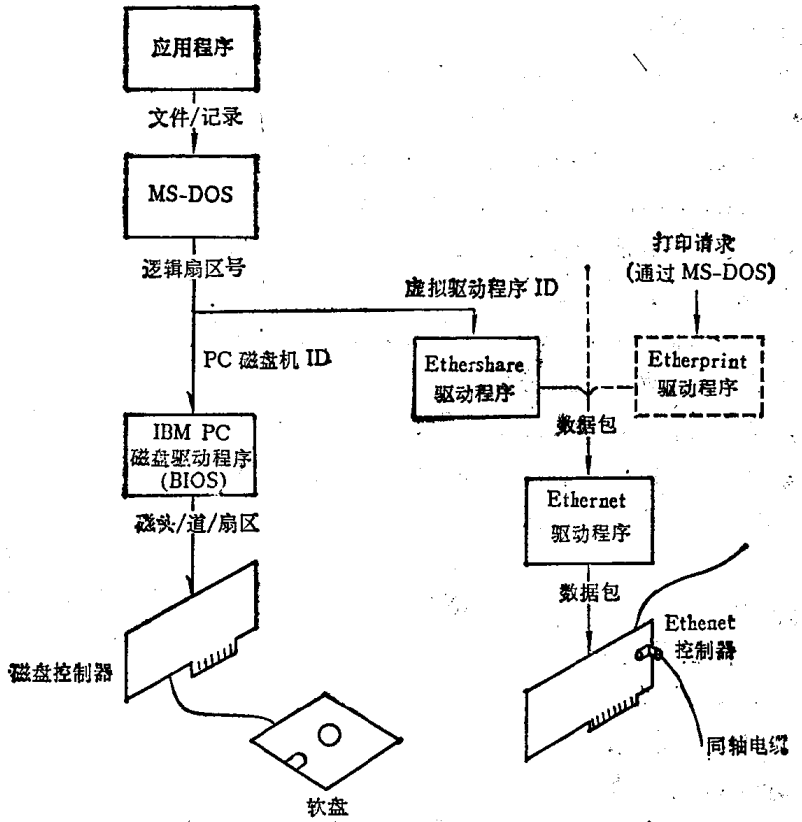


图 33-23　用户对本地软盘机或共享磁盘机的访问

思 考 题

一、IBM PC ROM 区存贮了什么系统程序?

二、在 86 汇编程序中怎样调用 BIOS 服务程序?

三、BIOS 是怎样写带、读带的?

四、试考虑编写一段读取键盘返回扩充扫描码,然后按您定义的功能加以处理的程序。

五、你对 BIOS 有何想法,开机检测是否必要?

六、为何中断向量表必须从 ROM 区送到 RAM 区?

七、DMA 是怎样建立的,计时器是怎样送初值的?

八、试画出 BIOS 键盘中断服务程序的框图。

九、试画出 BIOS 取键盘扫描码程序(用 INT 调用)的框图。

十、试画出 BIOS 显示器服务程序单色字符处理部分的框图。

十一、试画出 BIOS 软盘驱动器服务程序的框图。

十二、试画出 BIOS 打印机处理程序的框图。

十三、试画出 BIOS 盒带机处理程序的详细框图,弄清 write-bit 子程序的工作机理。

十四、从图形 CRT 用的字符发生表,能得出什么一般性的东西,你是否能独立设计一个字符发生表?

十五、弄懂时钟中断服务程序 TIMER-INT 的工作原理,建立时钟中断的概念。时钟中断除用于计时外还有什么用处?

十六、试述 CP/M-86 的特点。

十七、试述 PC-DOS 的特点。

十八、何为 P-system 的移植性?

十九、Unix 的特点是什么?你认为在 PC 上建立多用户系统是否可行,是否必要?

二十、IBM PC 有哪三级 BASIC?

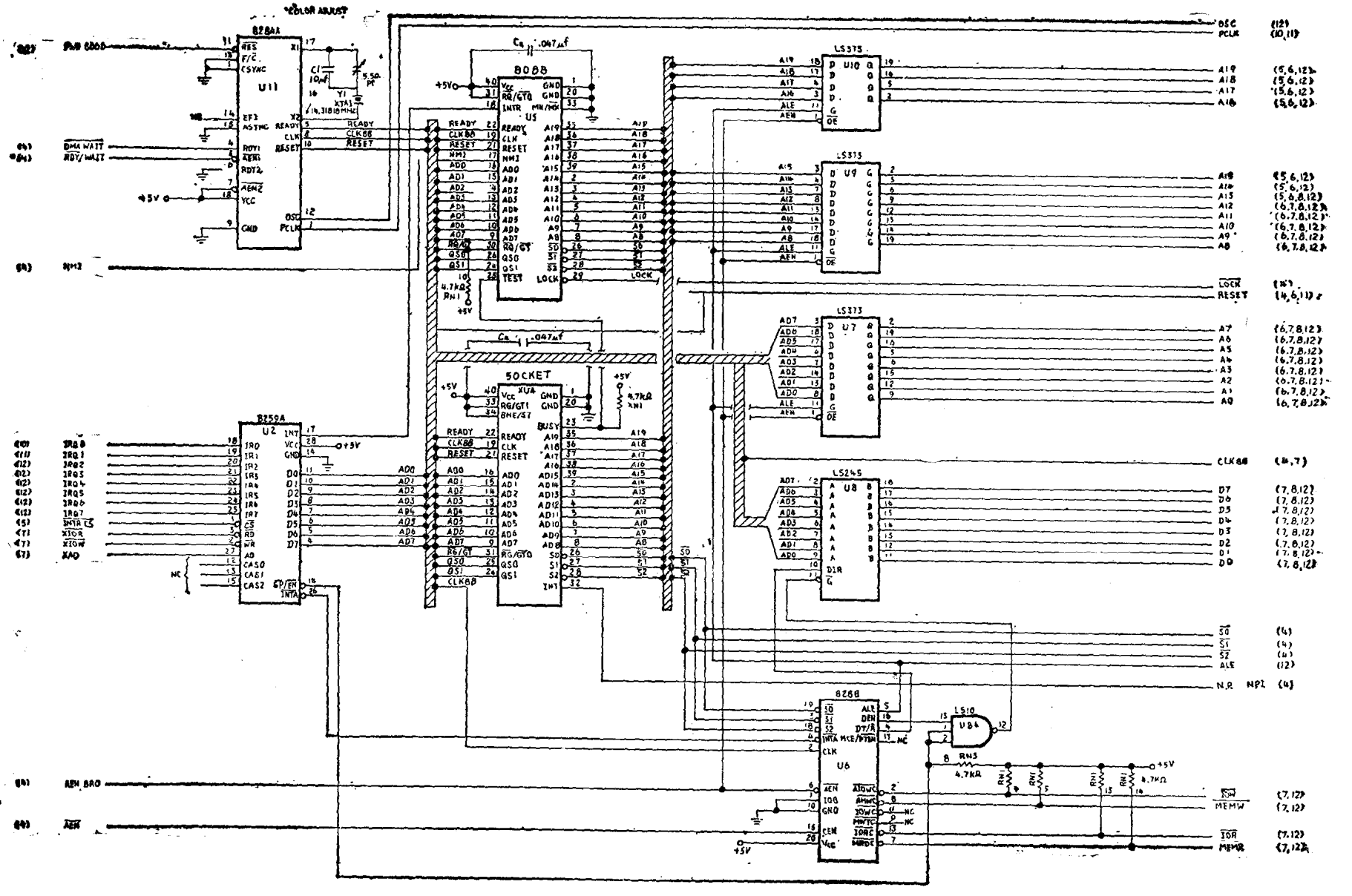
二十一、BASIC 中断控制原理是什么?

二十二、试读懂图形显示软件包,有条件的话,将其绘图仪功能段改成 CRT 功能段,在彩色监视器上显示三维图象。

二十三、Ethernet 的工作原理是什么?它是怎样支持结点机(PC)上应用程序的?

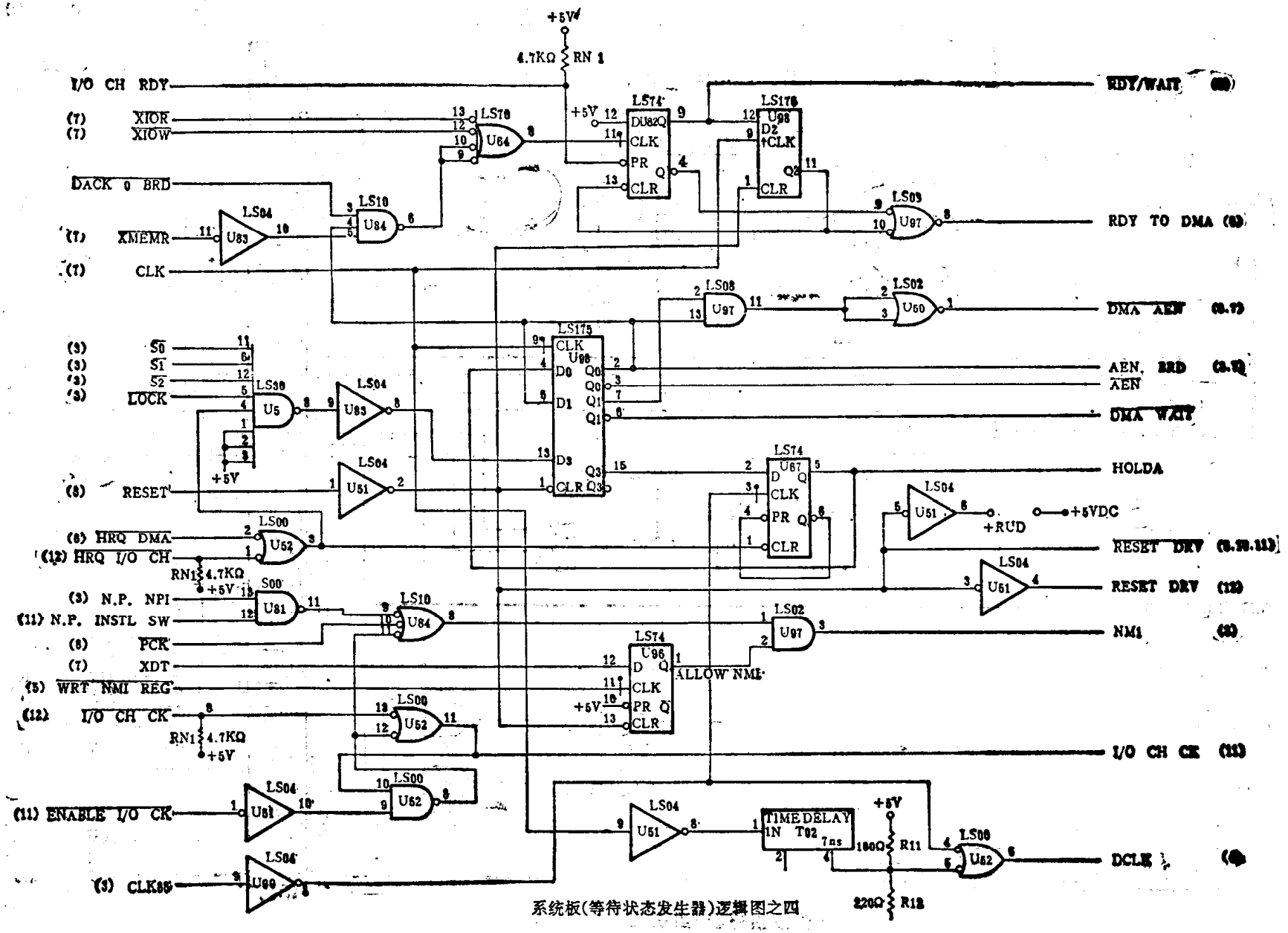
附录 1 IBM PC 逻辑线路图

系统板(处理器及支持部件)



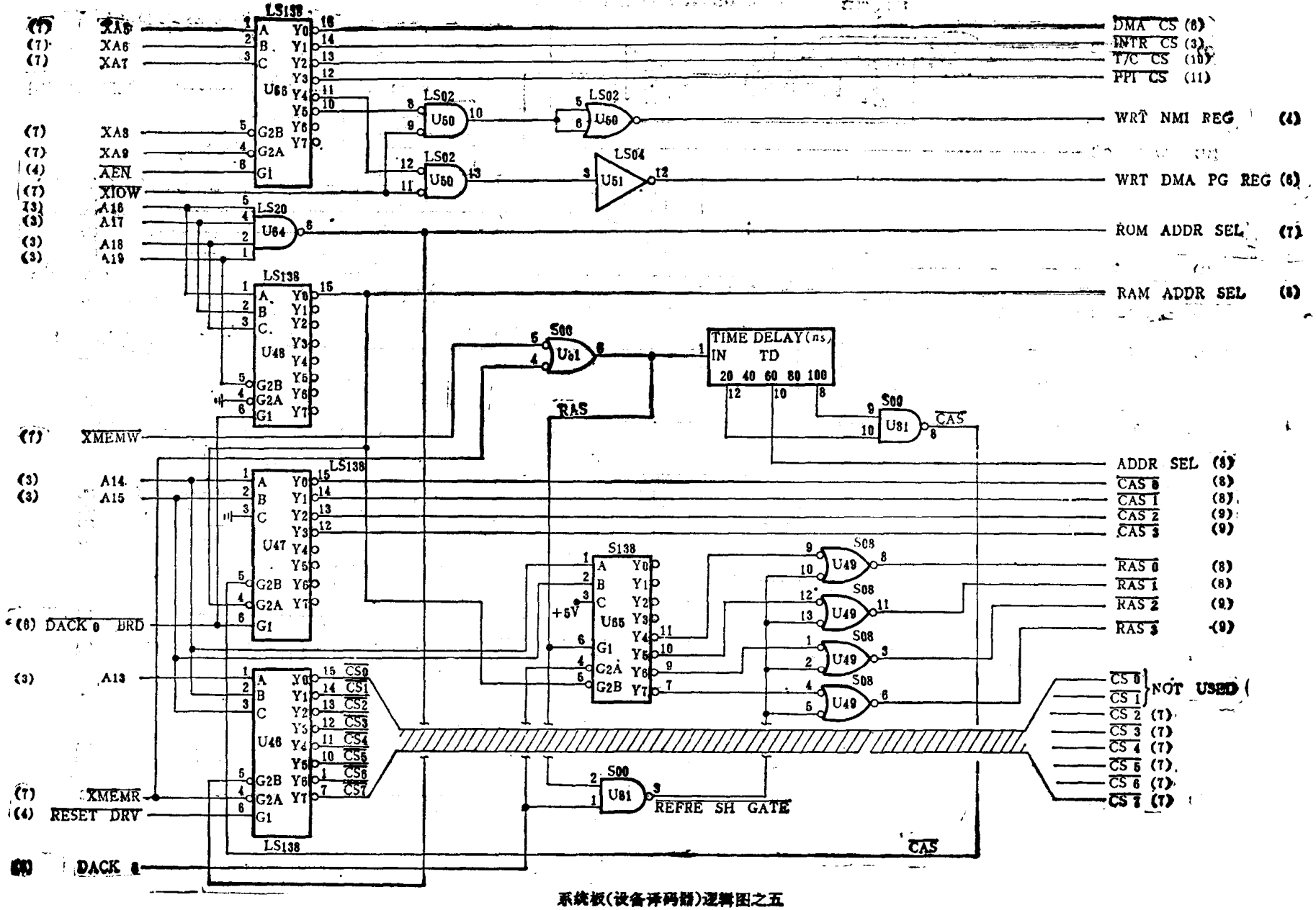
系统板(处理器及支持部件)逻辑图之五

系统板(等待状态发生器)

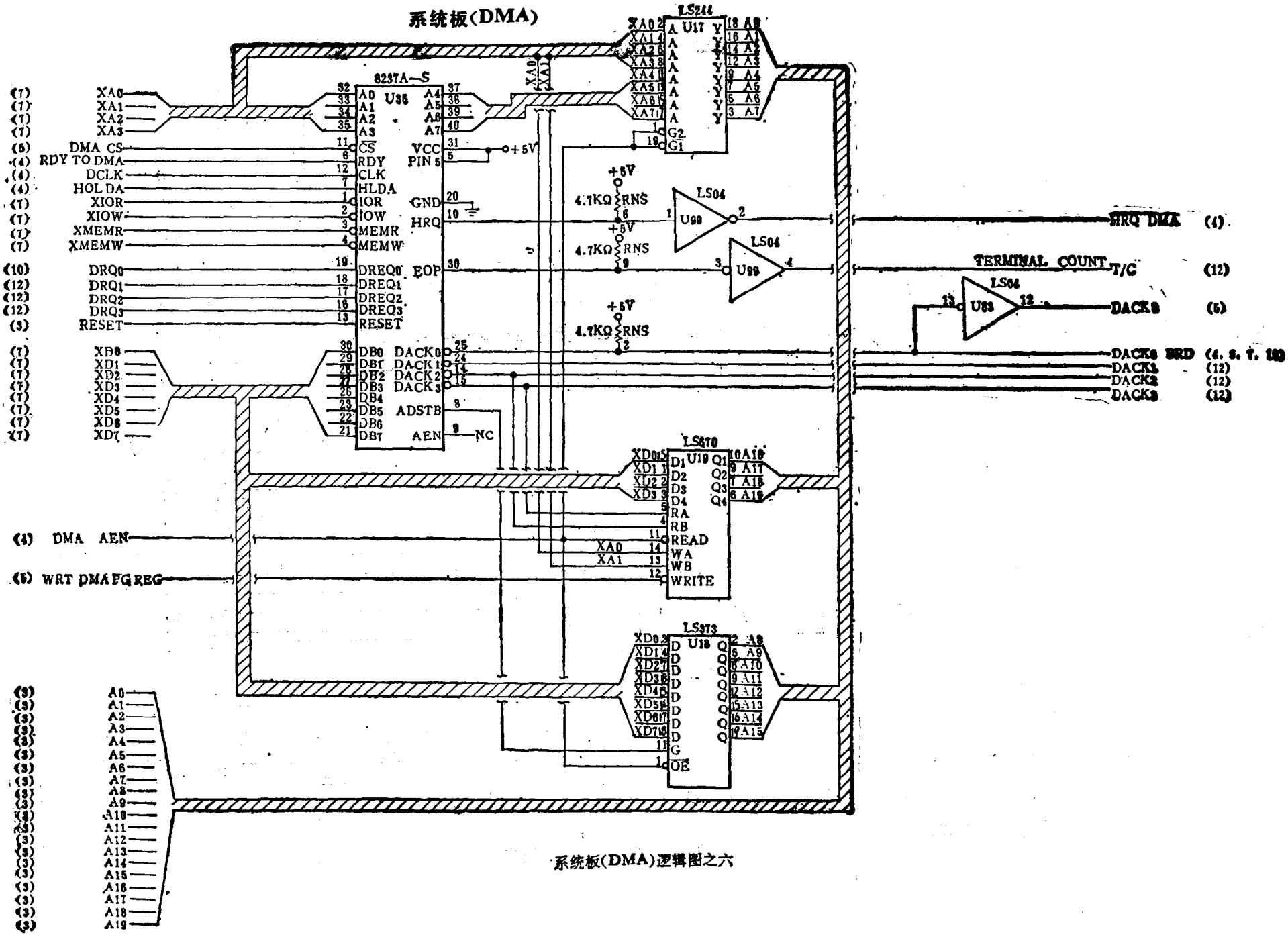


系统板(等待状态发生器)逻辑图之四

系统板(设备译码器)

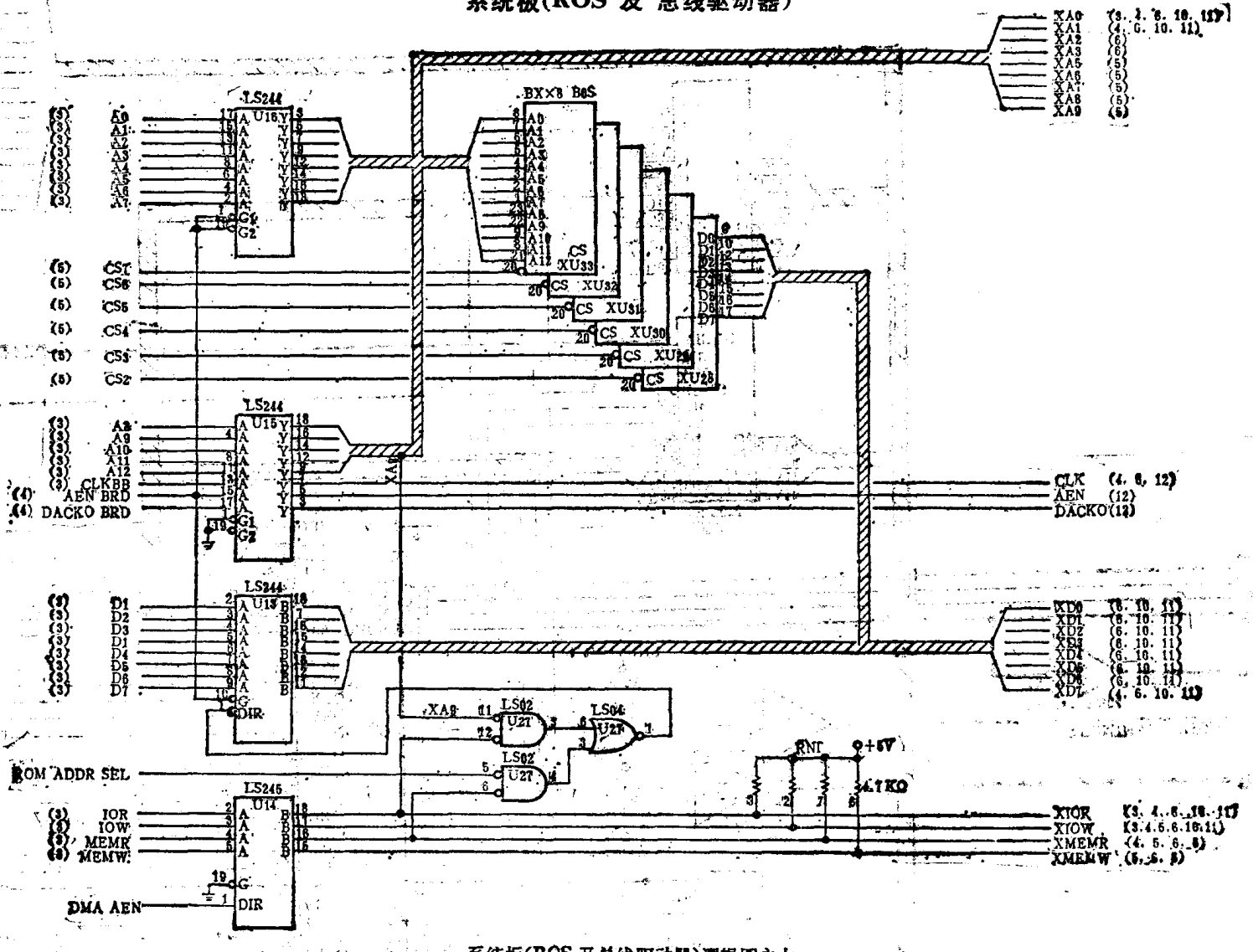


系统板(设备译码器)逻辑图之五

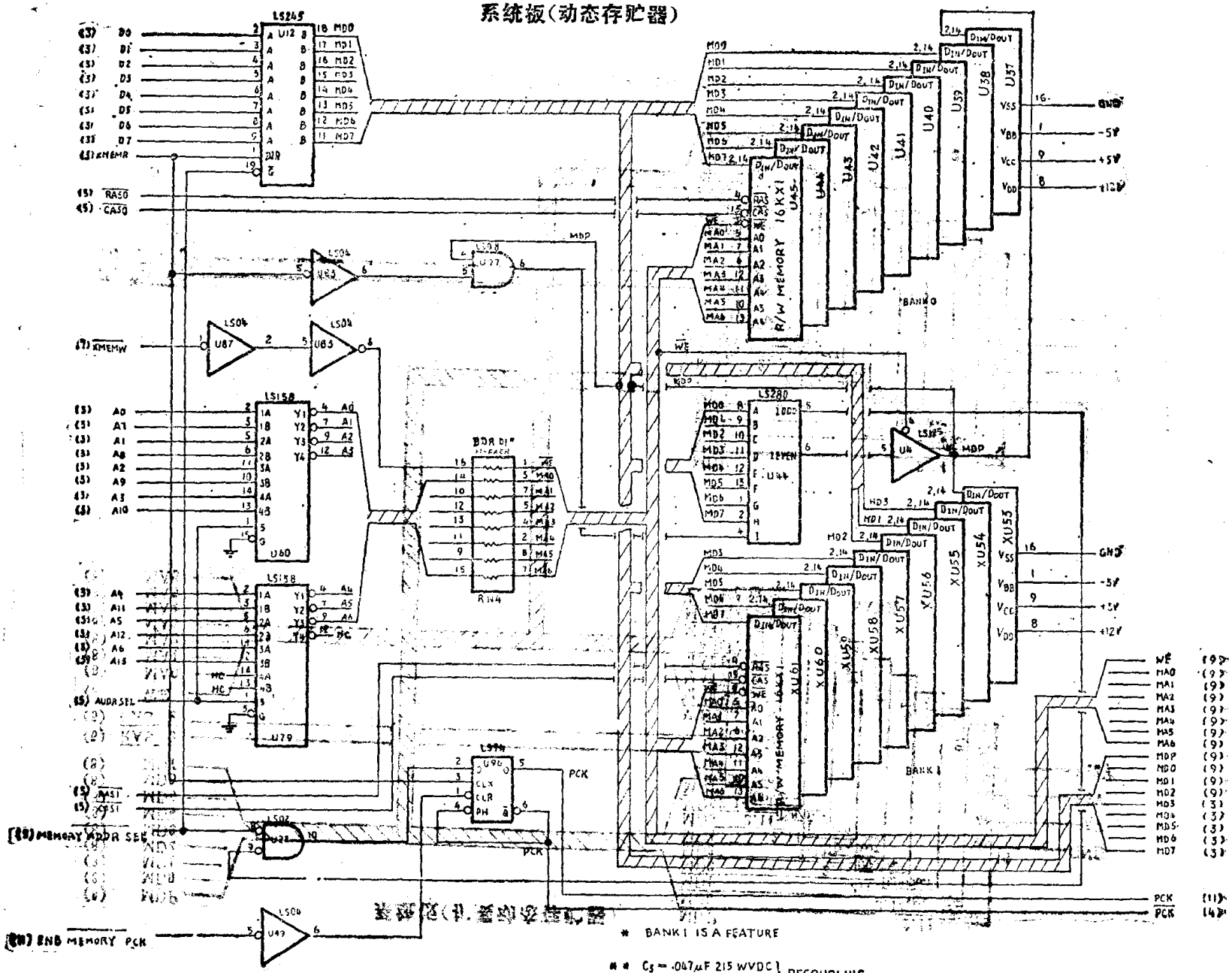


系统板(DMA)逻辑图之六

系统板(ROS 及 总线驱动器)



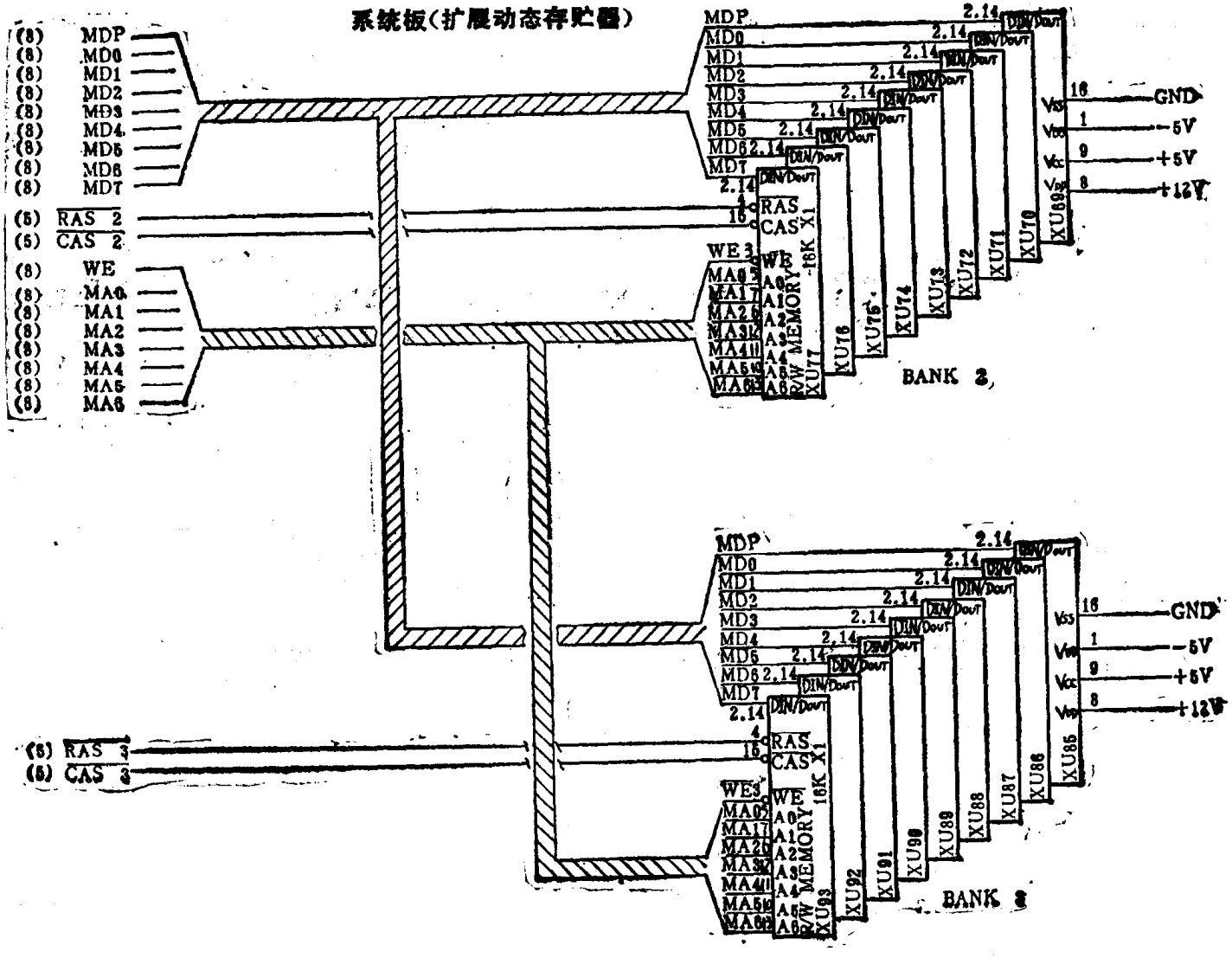
系统板(ROS 及总线驱动器)逻辑图之七



* BANK1 IS A FEATURE

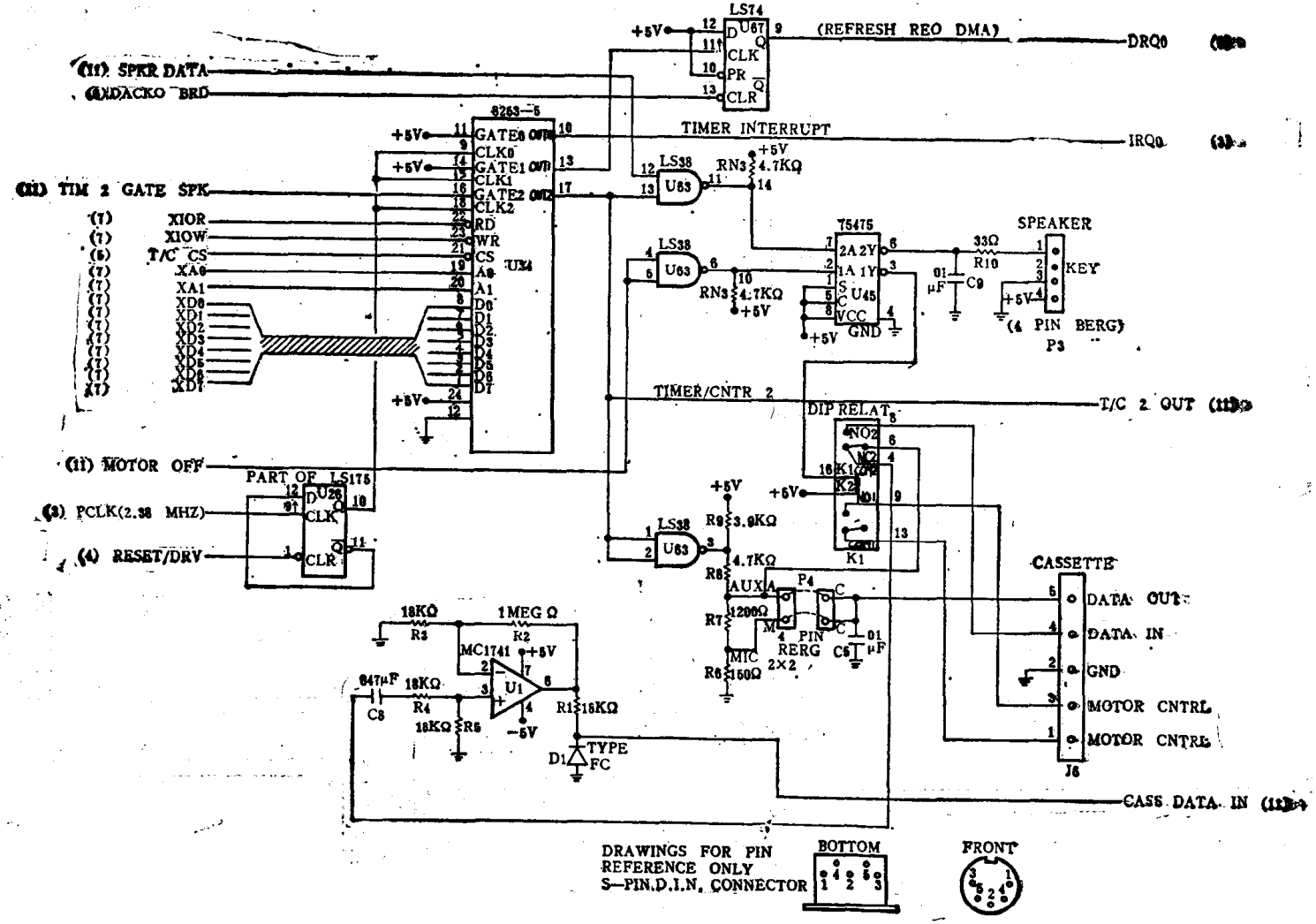
** C_3 = .047μF 215 WVDC
C_4 = .047μF / WVDC
} DECOUPLING

系统板逻辑图之八:动态存贮器

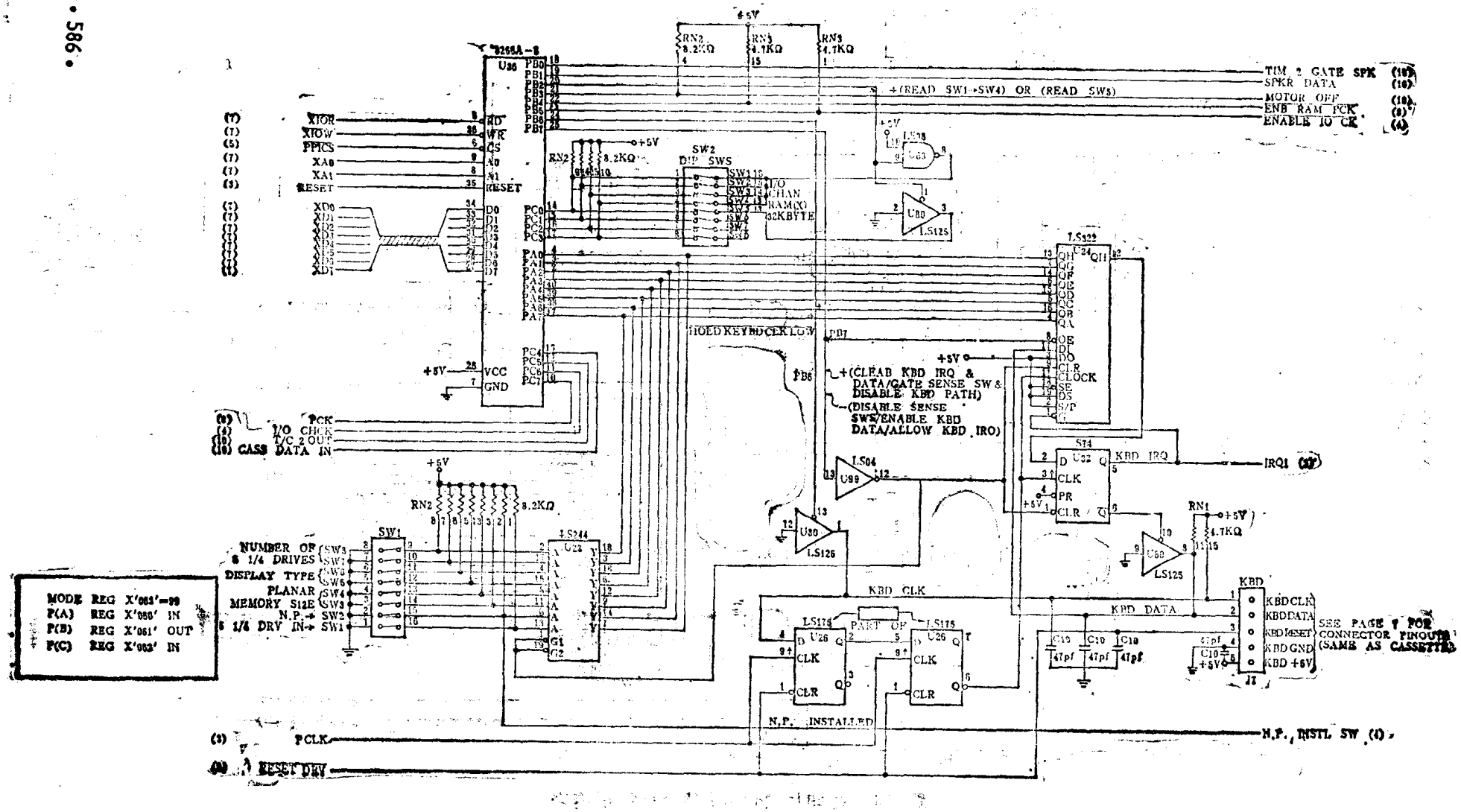


系统板逻辑图之九:扩展动态存贮器

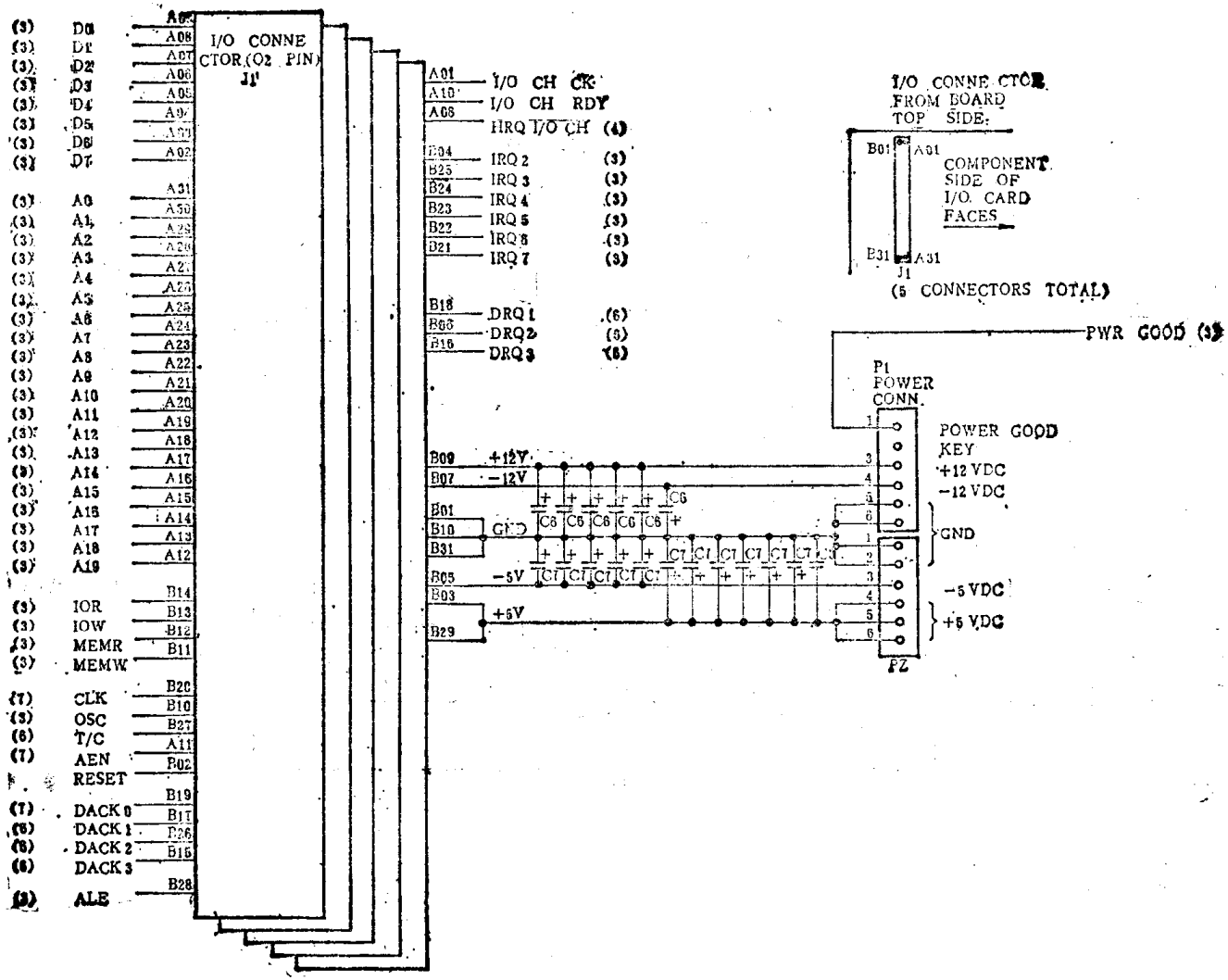
系统板(扬声器/盒带机/计时器/计数器)



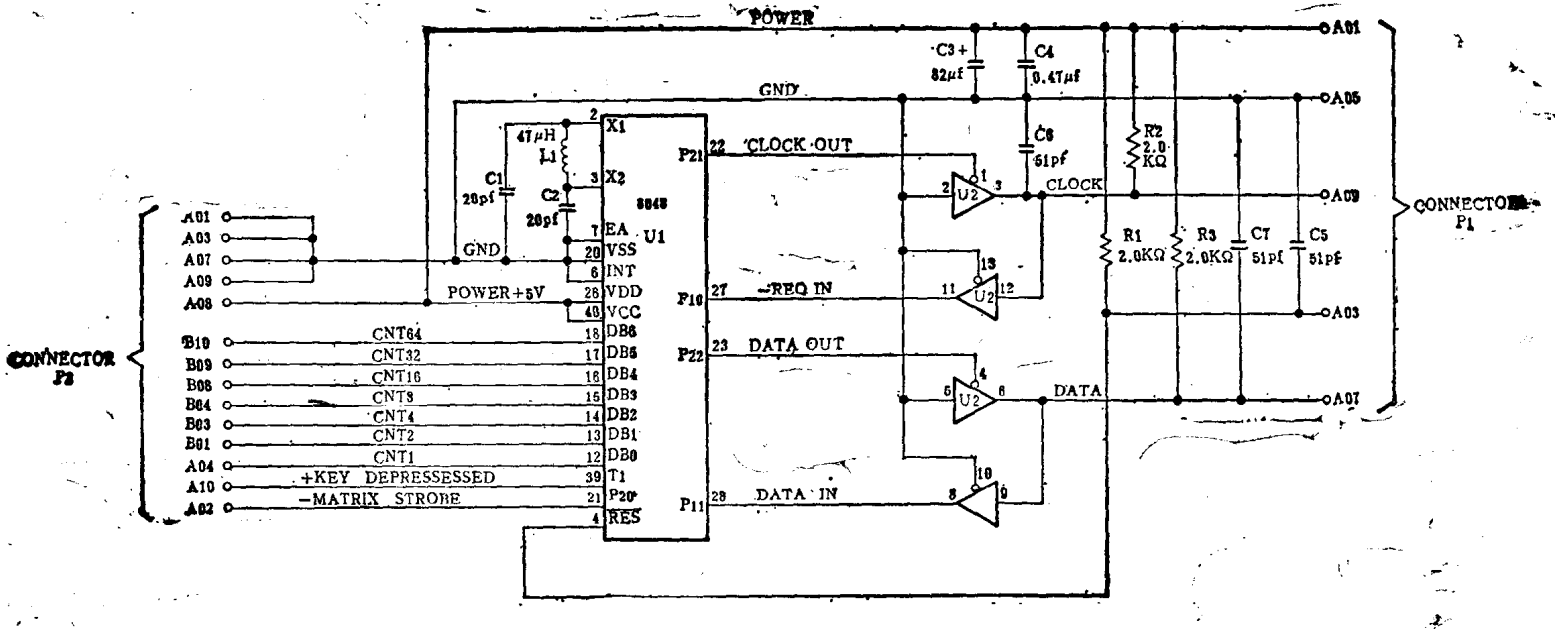
系统板逻辑图之十:扬声器/盒带机/计时器/计数器



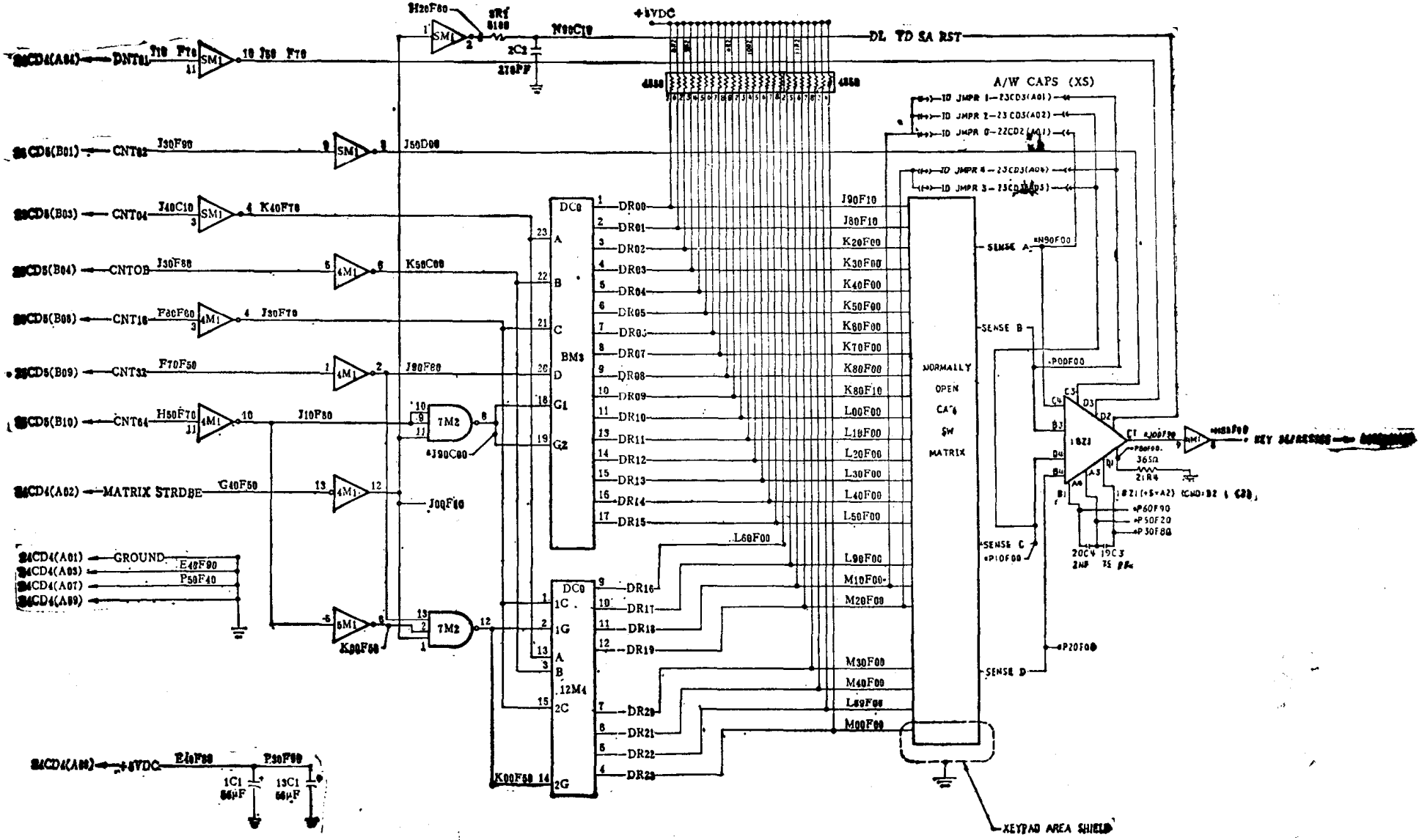
系统板逻辑图之十一：键盘接口/设备配置开关/控制器



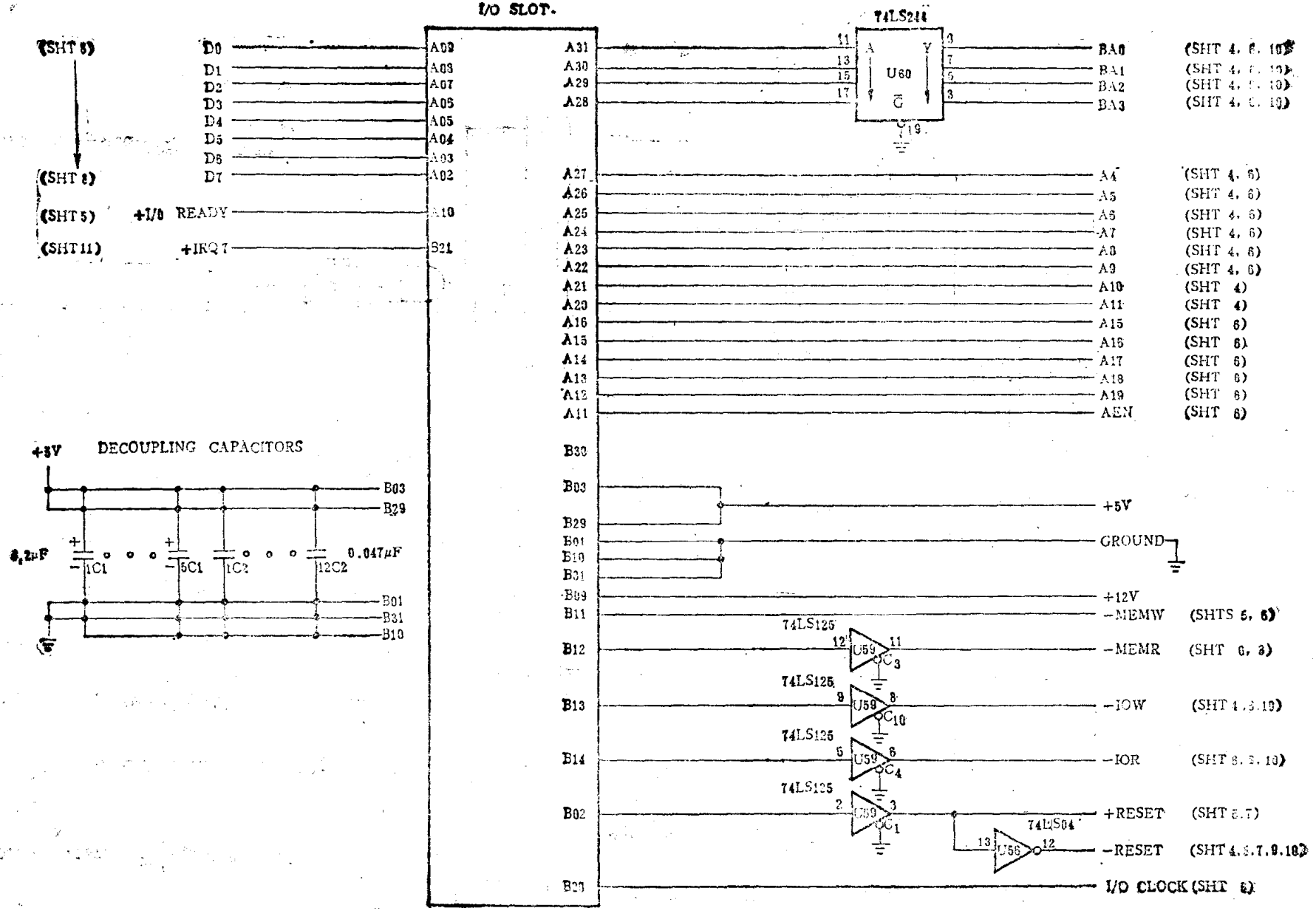
系统板逻辑图之十二：I/O 通道本图的电容均为 8.2μF 钽质电容


POWER
GND
C3+
82μf
C4
0.47μf
C6
51pf
R2
2.0
KΩ
R1
2.0KΩ
R3
2.0KΩ
C7
51pf
C5
51pf
CLOCK
DATA
U2
X1
X2
47μH
L1
C1
20pf
C2
20pf
EA
VSS
INT
VDD
VCC
DB6
DB5
DB4
DB3
DB2
DB1
DB0
T1
P20
RES
U1
P21 CLOCK OUT
P10 -REQ IN
P22 DATA OUT
P11 DATA IN
POWER+5V
CONNECTOR P2
CONNECTOR P1
A01
A03
A07
A09
A08
B10 CNT64
B09 CNT32
B08 CNT16
B04 CNT8
B03 CNT4
B01 CNT2
A04 CNT1
A10 +KEY DEPRESSESSED
A02 -MATRIX STROBE
A01
A05
A09
A03
A07

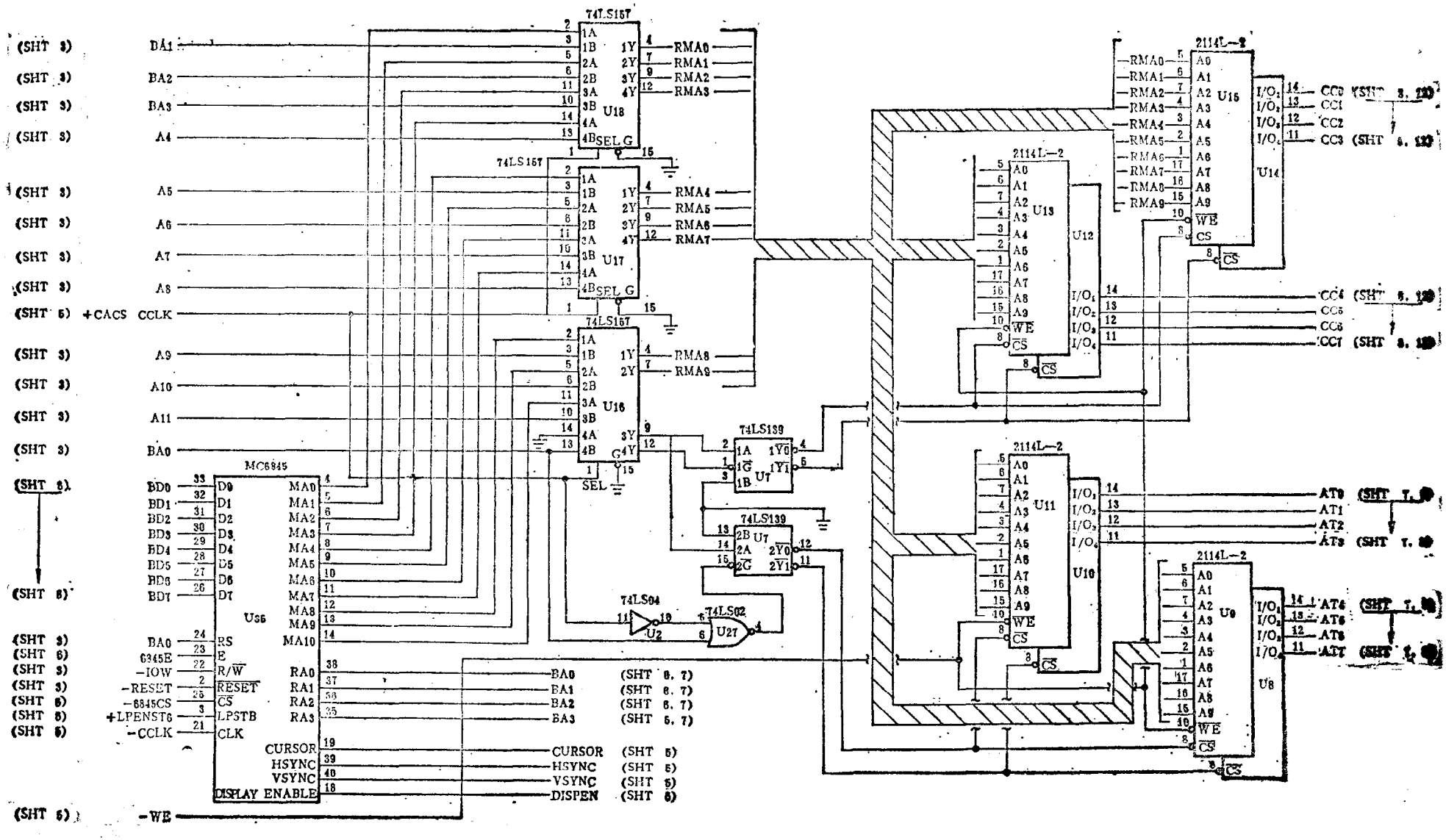
键盘逻辑图之一


A/W CAPS (XS)
DL TD SA RST
+5VDC
GROUND
MATRIX STRDBE
CNT02
CNT04
CNT08
CNT16
CNT32
CNT64
CNTOB
NORMALLY OPEN CAPS SW MATRIX
SENSE A
SENSE B
SENSE C
SENSE D
KEYPAD AREA SHIELD
DR00
DR01
DR02
DR03
DR04
DR05
DR06
DR07
DR08
DR09
DR10
DR11
DR12
DR13
DR14
DR15
DR16
DR17
DR18
DR19
DR20
DR21
DR22
DR23

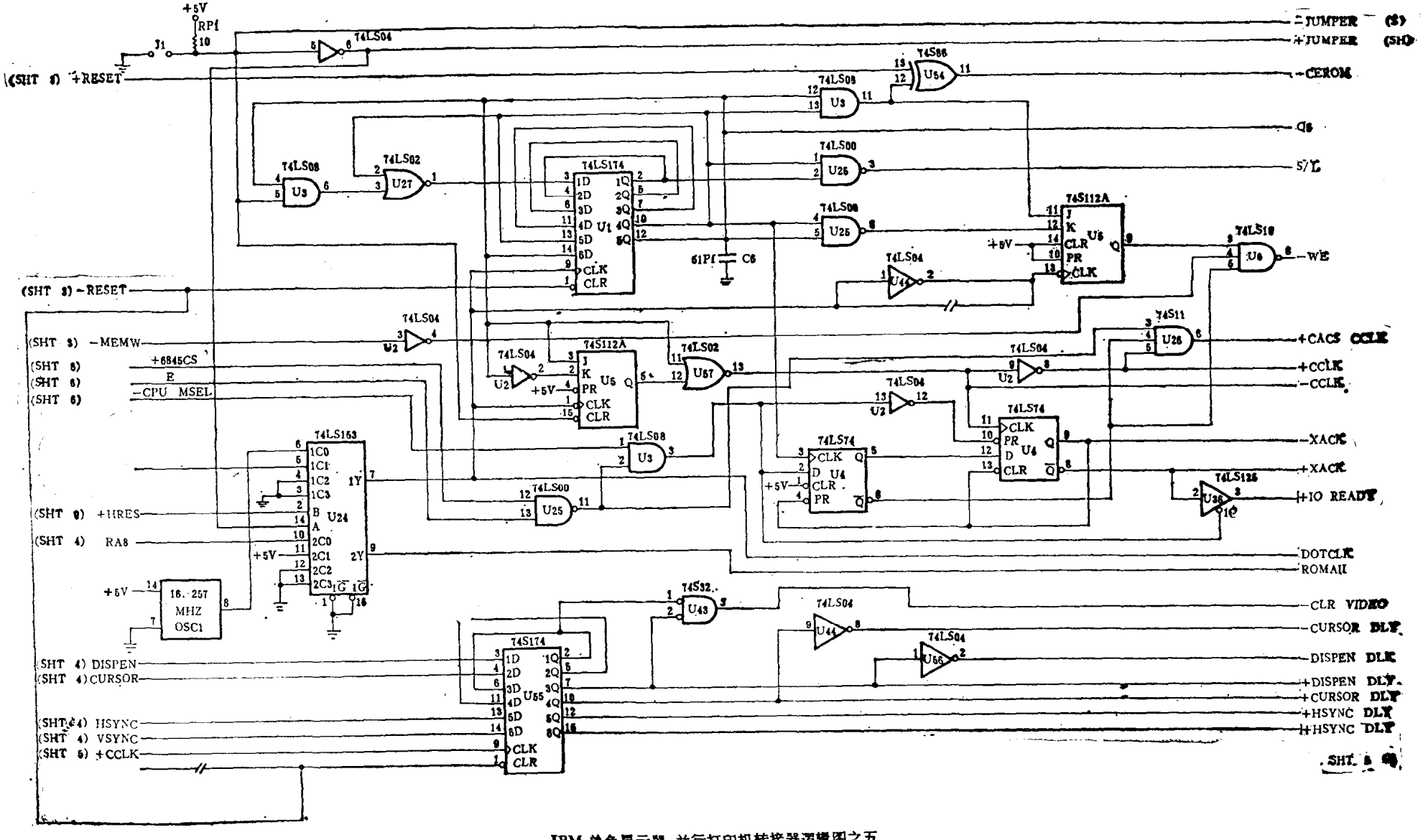
键盘逻辑图之二



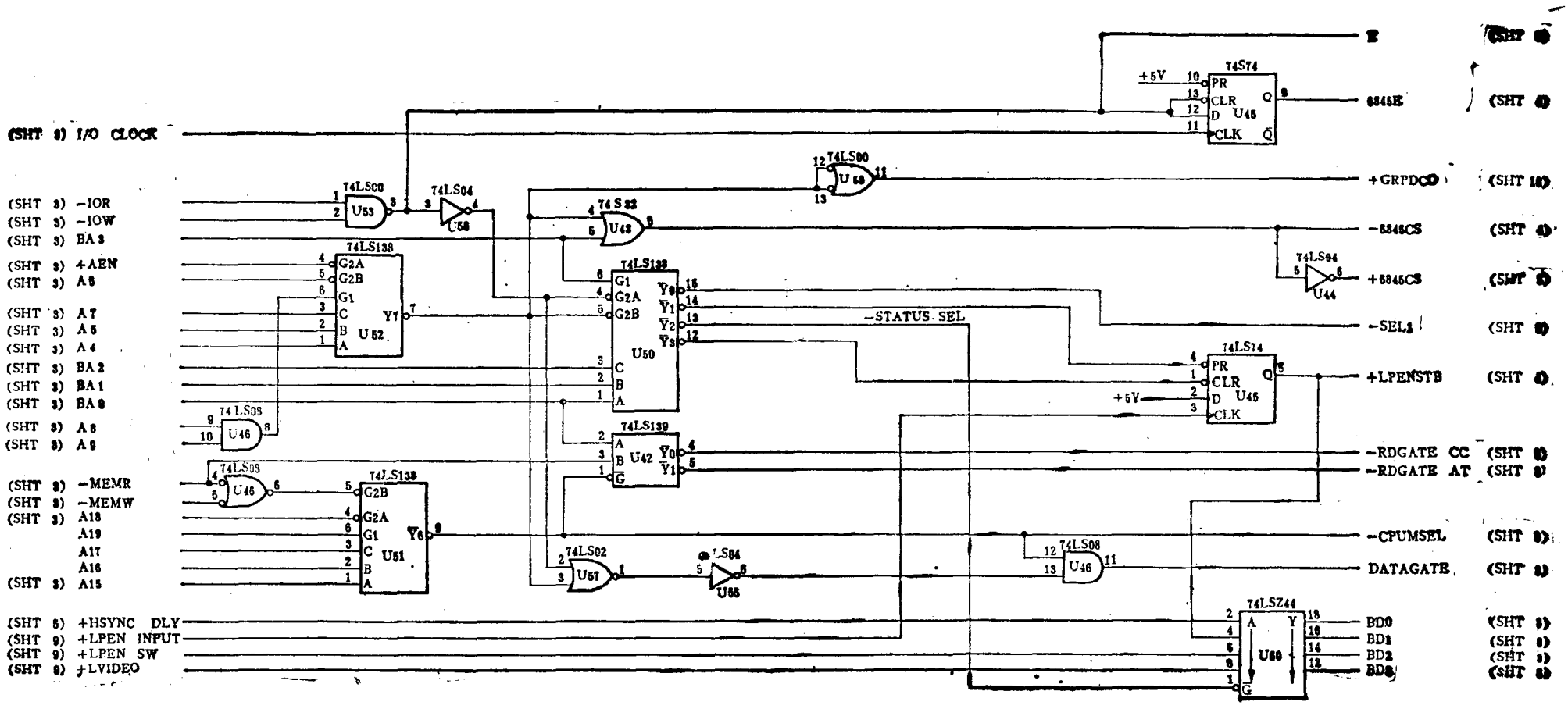
IBM 单色显示器-并行打印机转接器逻辑图之三



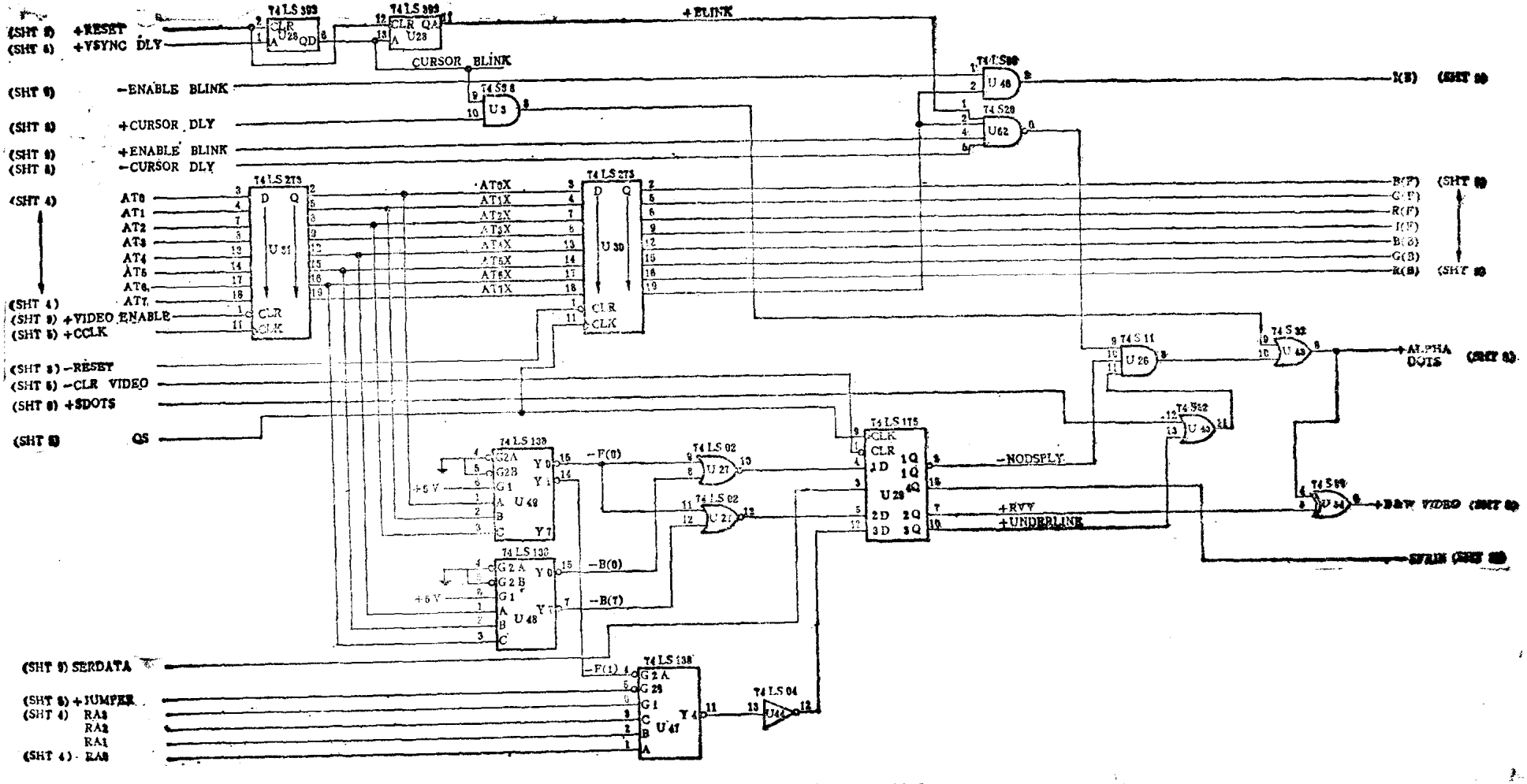
IBM 单色显示器-并行打印机转接器逻辑图之四



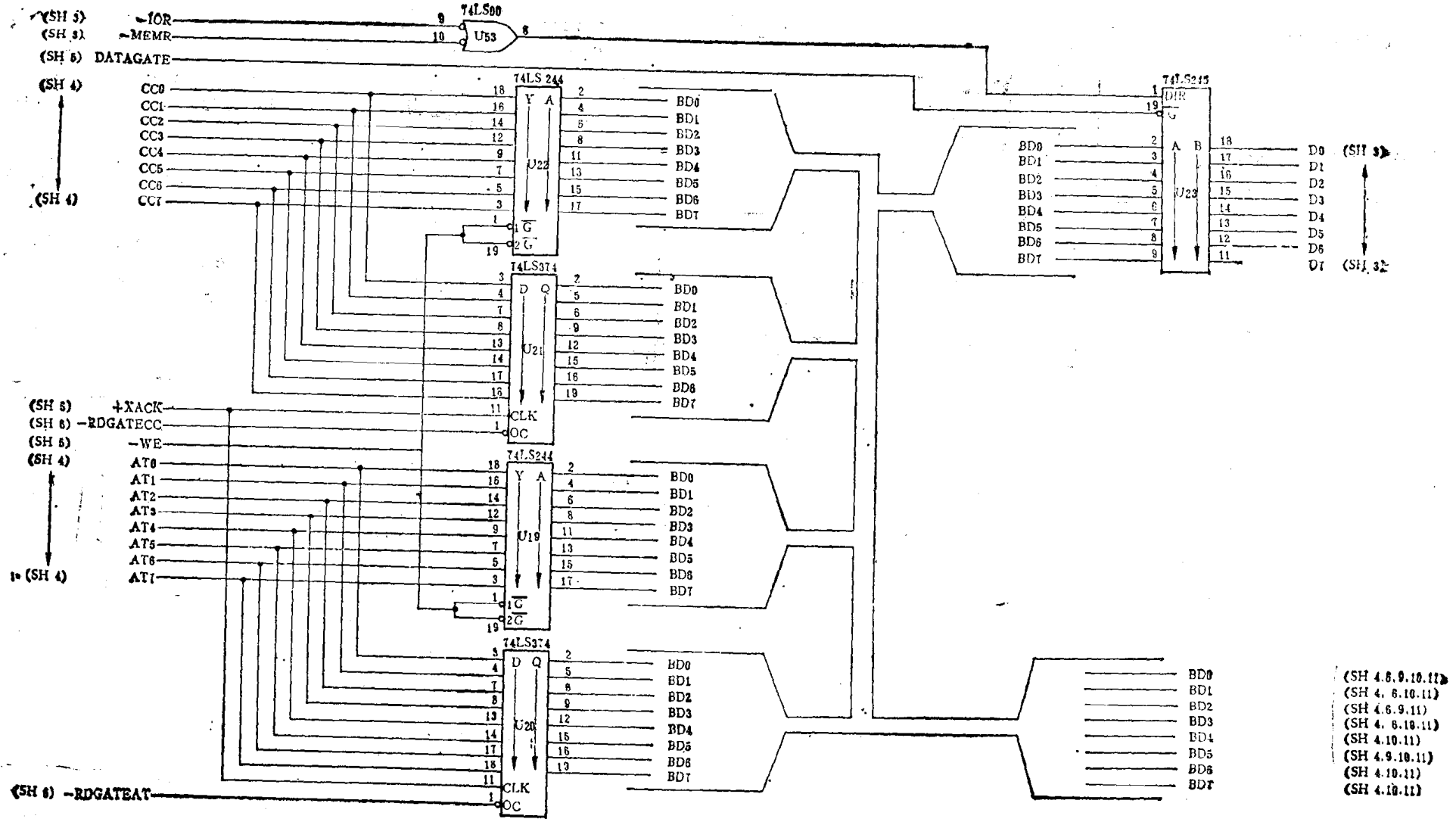
IBM 单色显示器-并行打印机转接器逻辑图之五



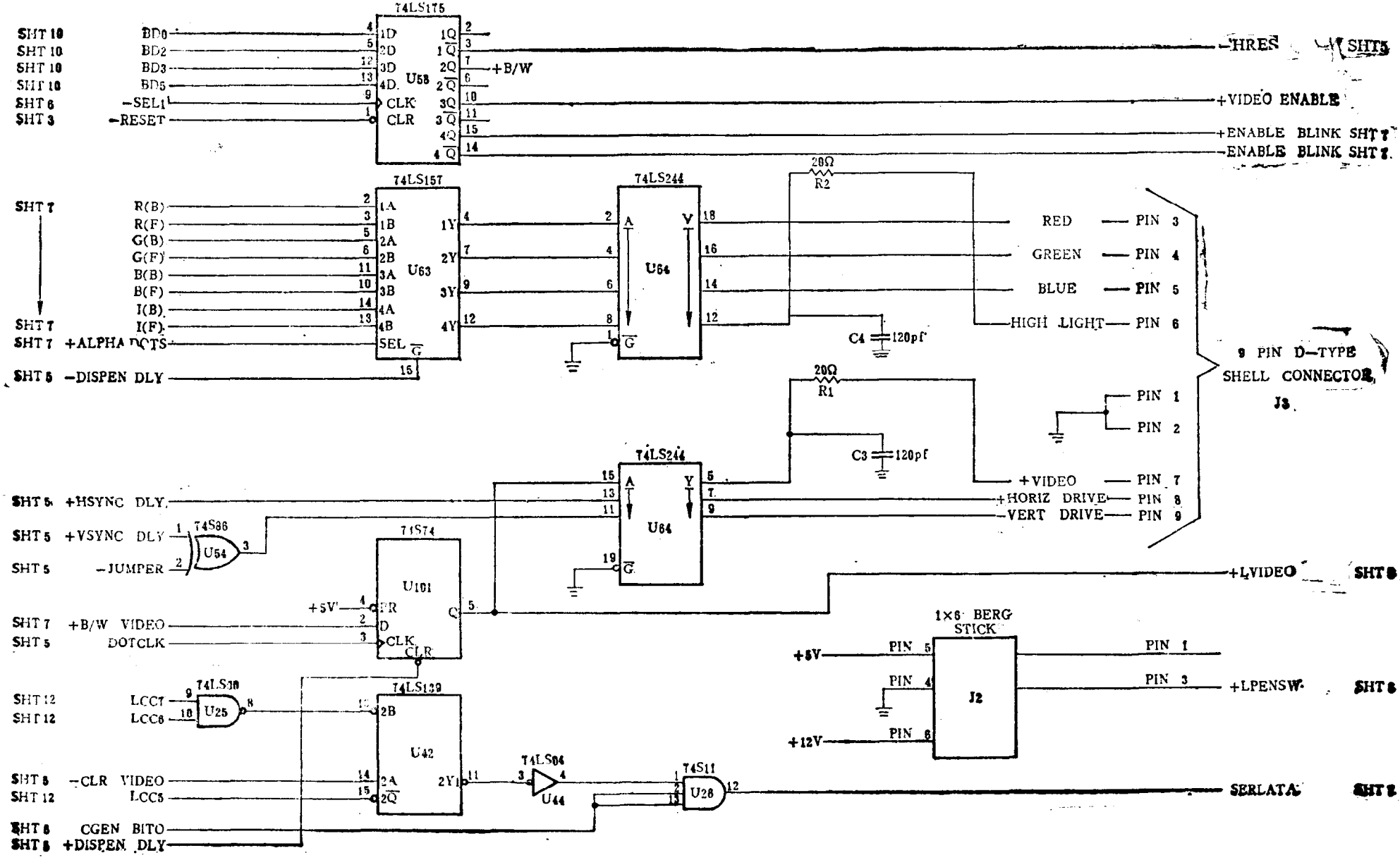
IBM 单色显示器-并行打印机转接器逻辑图之六



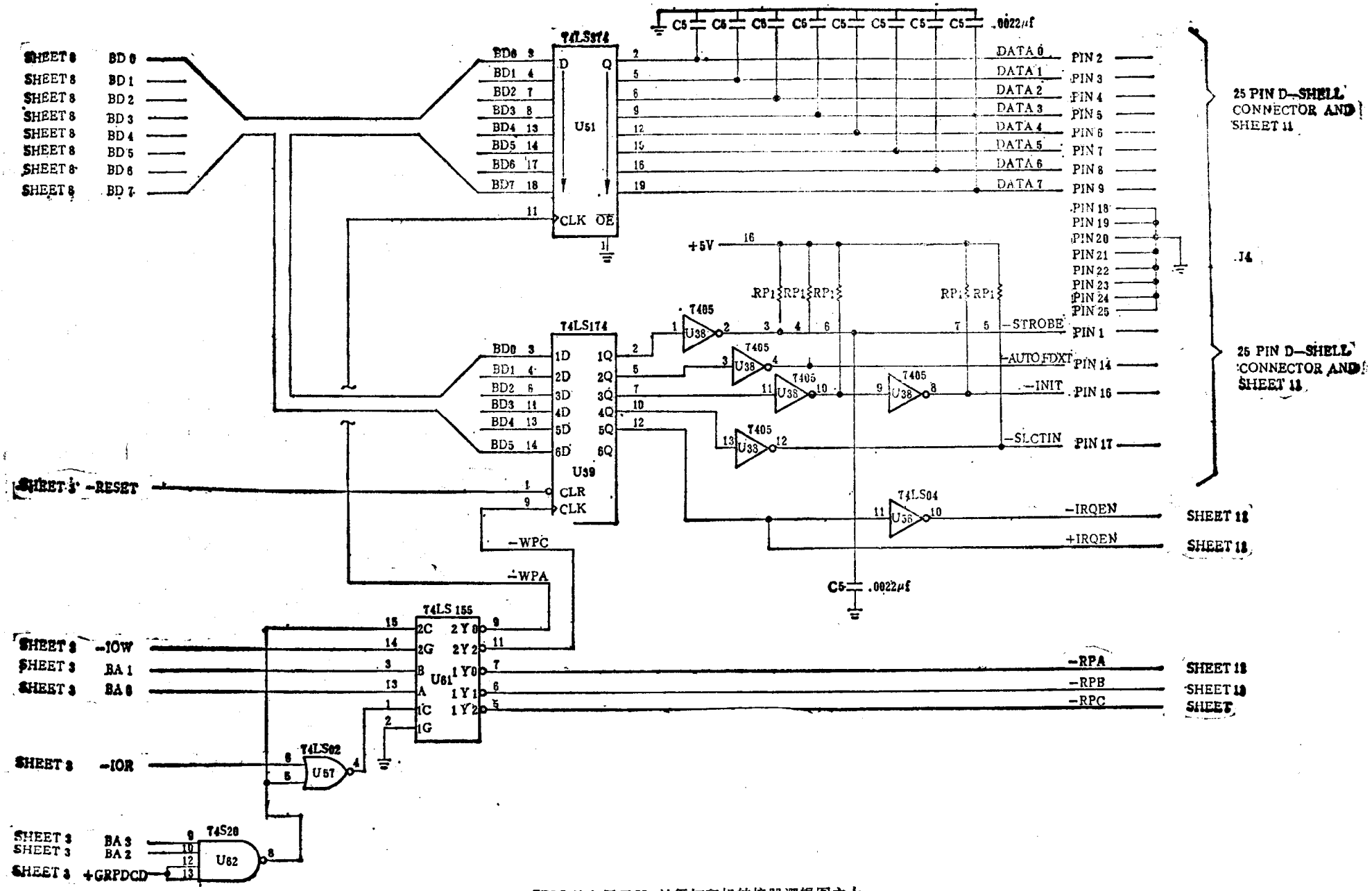
IBM 单色显示器-并行打印机转接器逻辑图之七



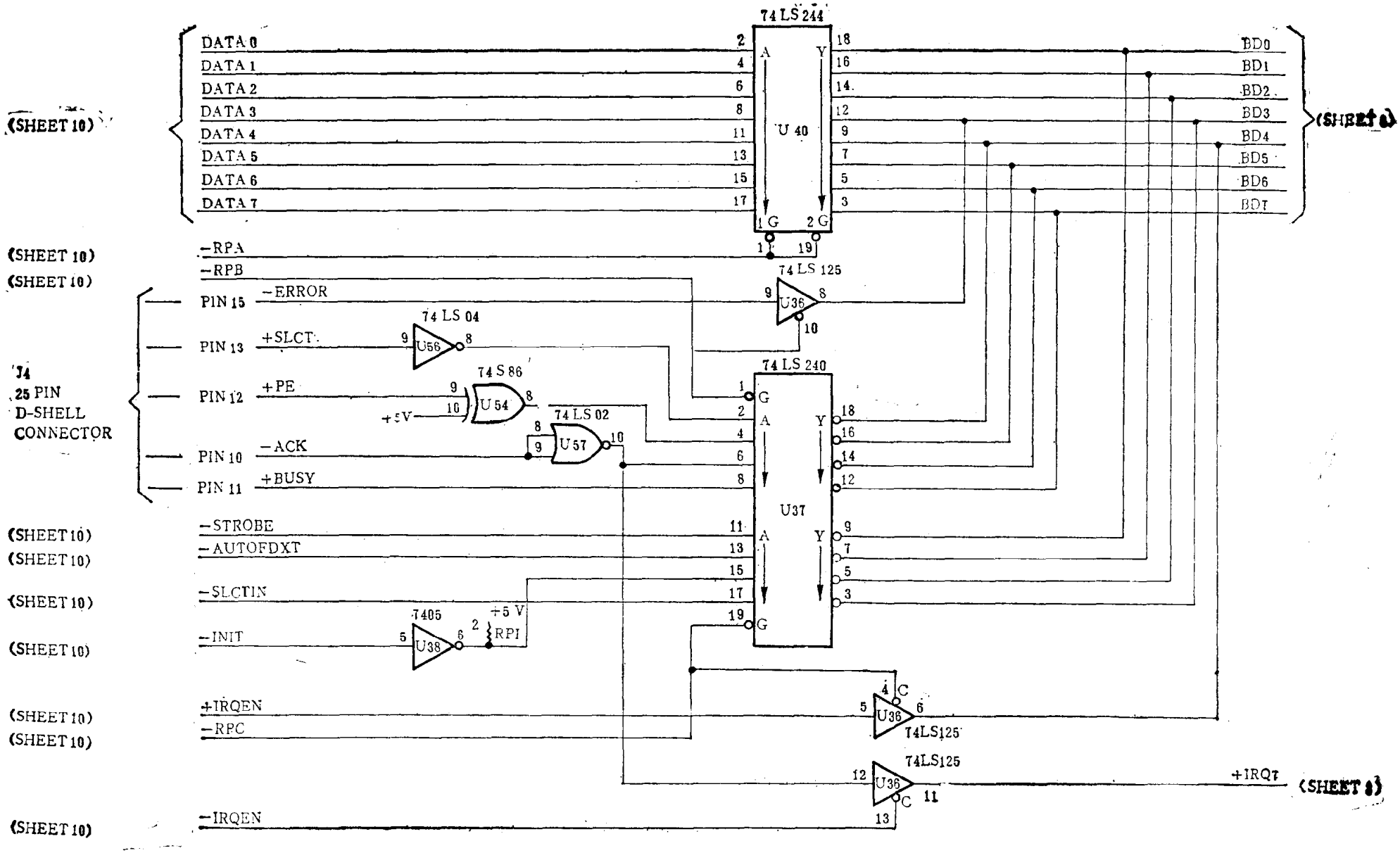
IBM 单色显示器-并行打印机转接器逻辑图之八



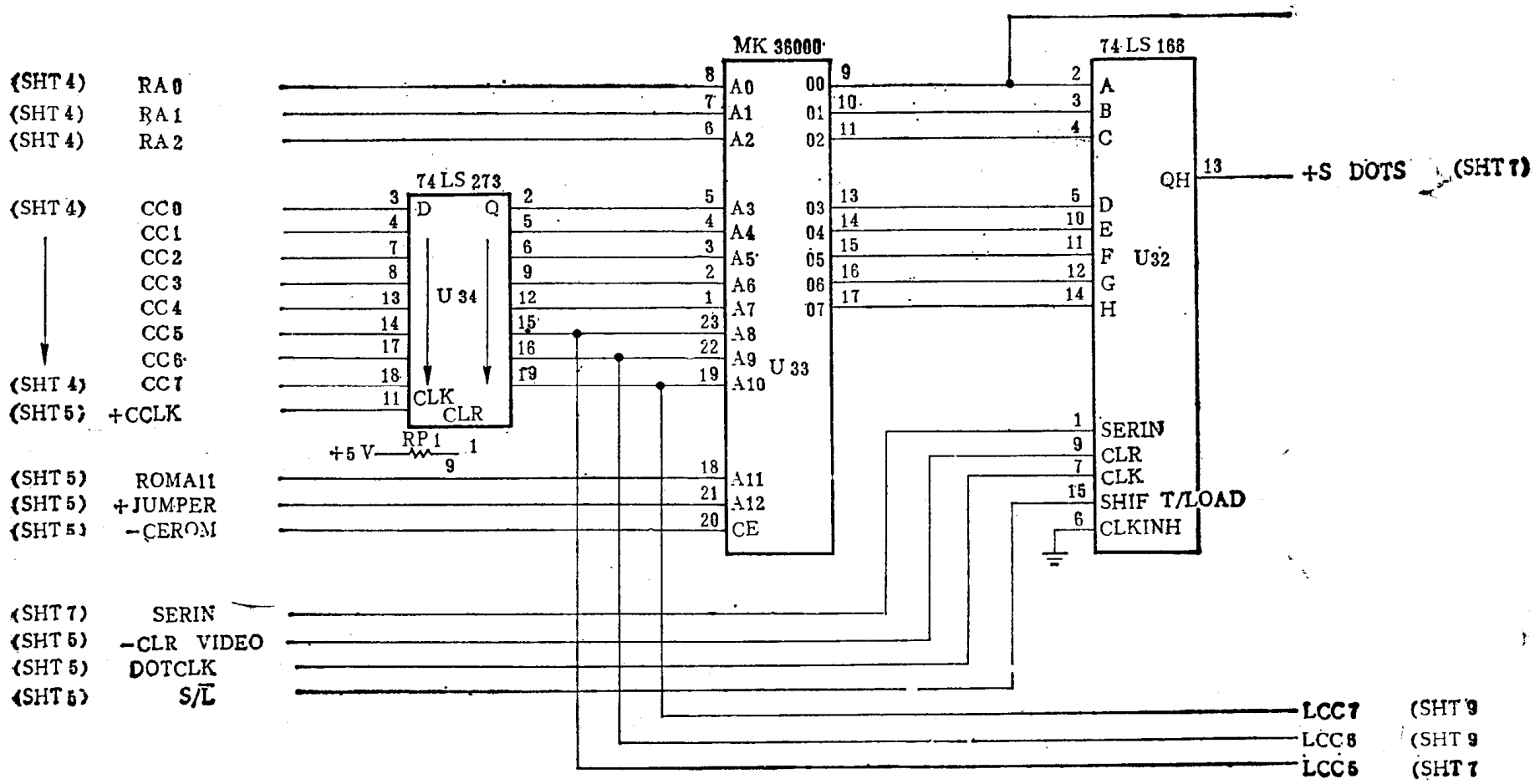
IBM 单色显示器-并行打印机转接器逻辑图之九



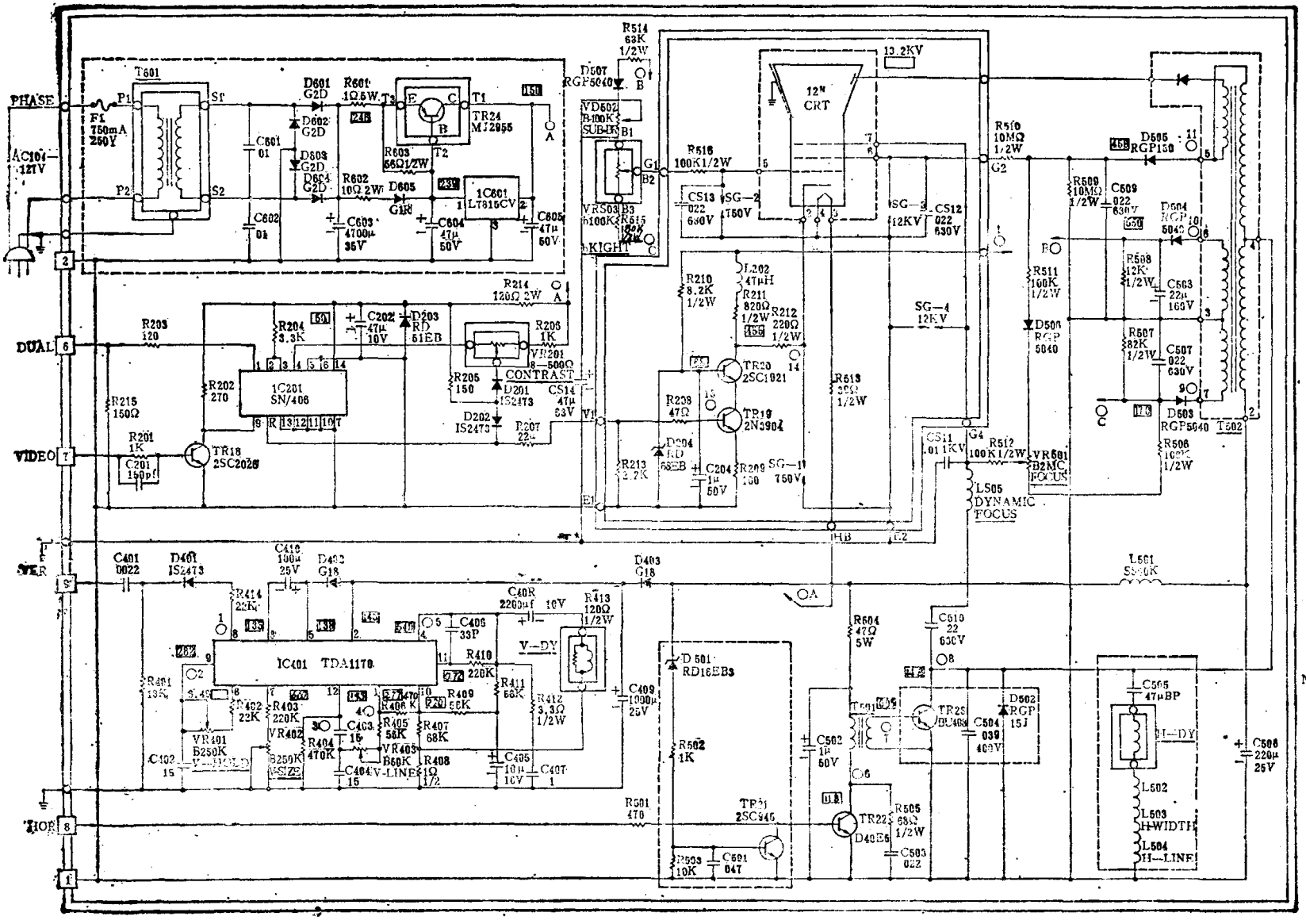
IBM 单色显示器-并行打印机转接器逻辑图之十



IBM 单色显示器-并行打印机转接器逻辑图之十一



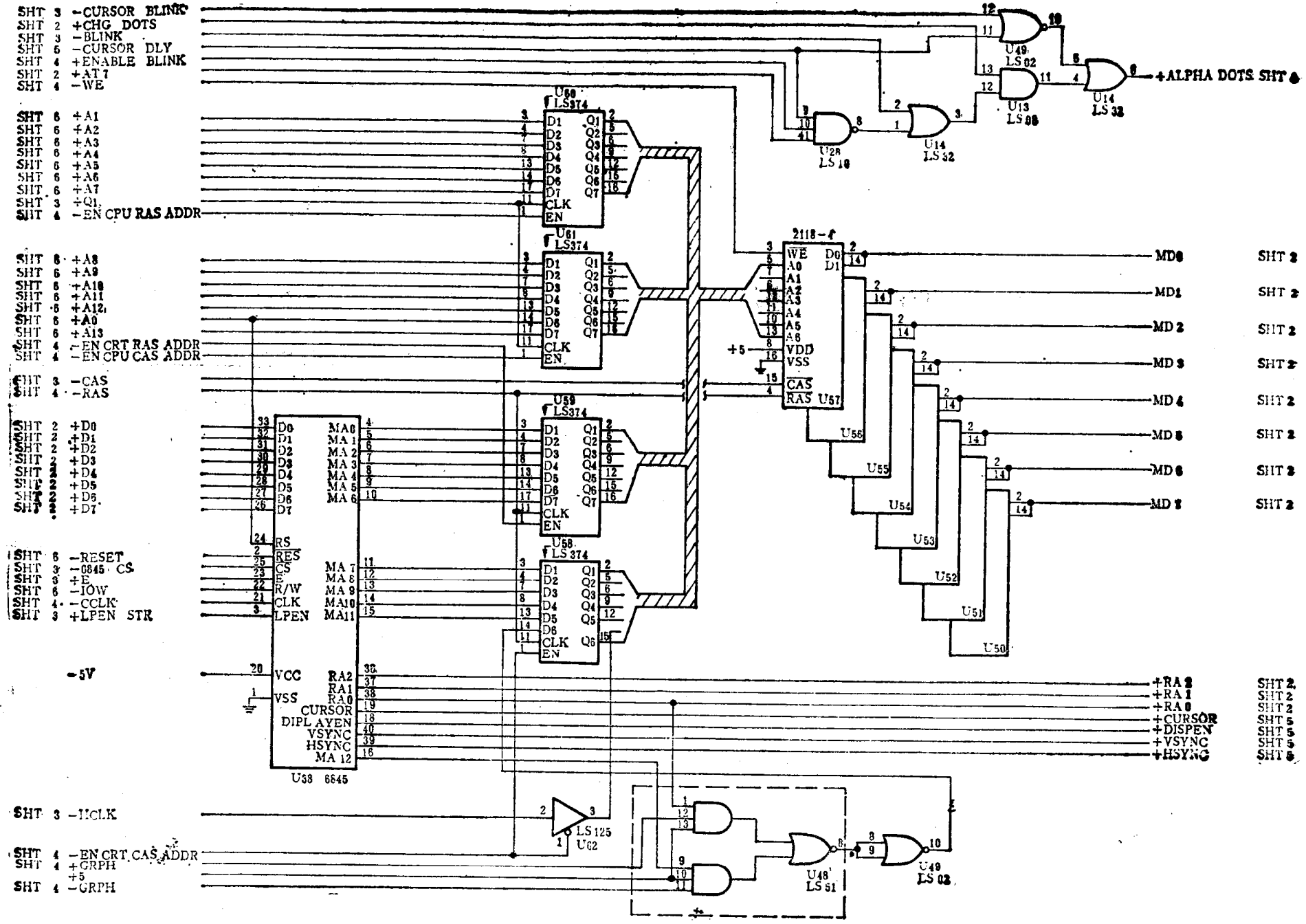
IBM 单色显示器-并行打印机转接器逻辑图之十二



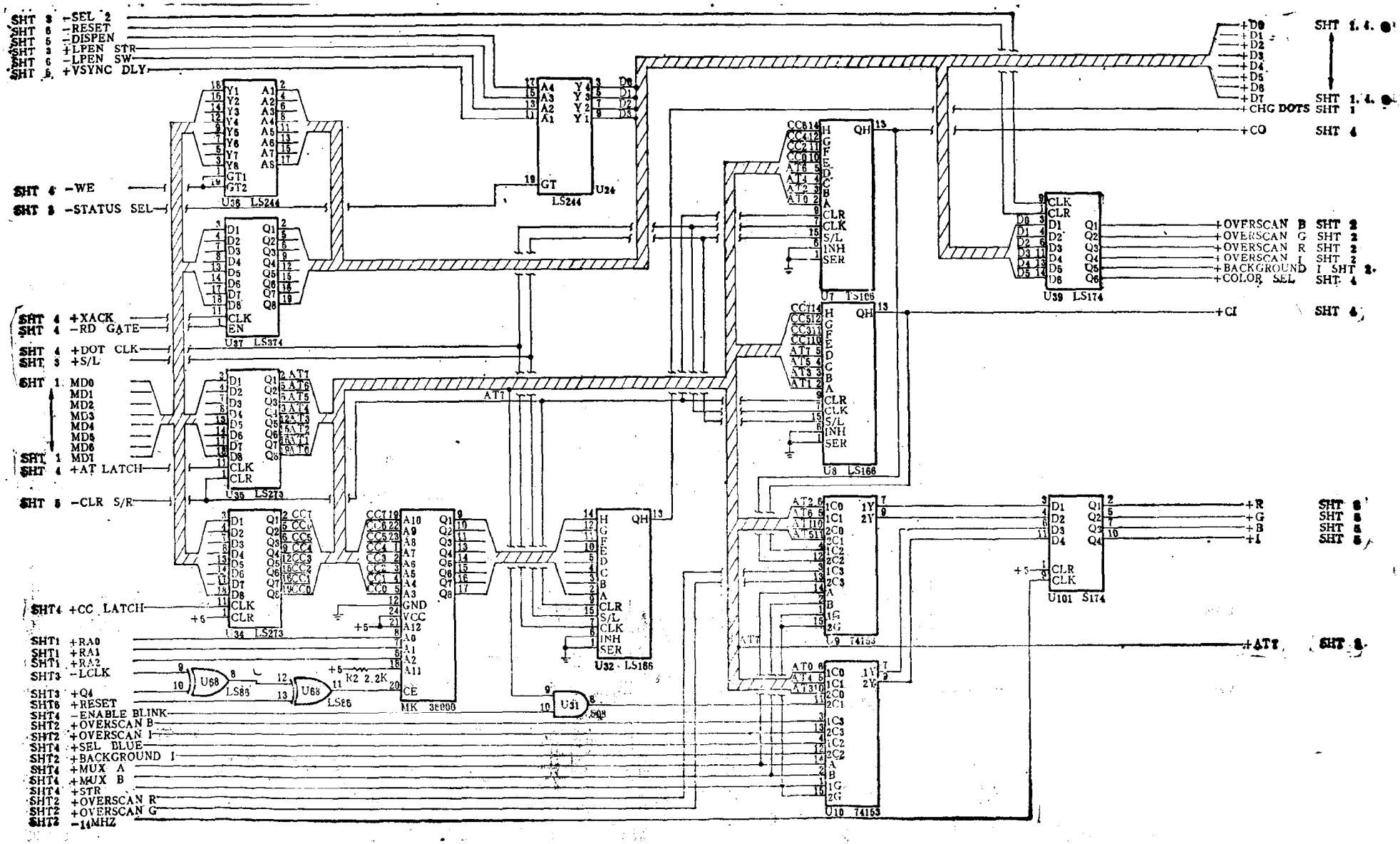
NOTES

1. RESISTOR VALUES ARE 1W (OHM)Ω K=1000Ω M=1,000,000Ω
2. ALL RESISTOR ARE 1/4W EXCEPT WHERE OTHERWISE INDICATED
3. ALL CAPACITORS ARE 50V EXCEPT WHERE OTHERWISE
4. CAPACITORS ARE VALUES μF UNLESS INDICATED μ=μF=1mF
5. AC WIRING INFORMATION PHASE=BLACK/BROWN WIRE NEUTRAL=WHITE/BLUE WIRE GROUND=GREEN and YELLOW WIRE IMPORTANT—THE PHASE WIRE MUST GO TO THE FUSED SIDE OF TRANSFORMER

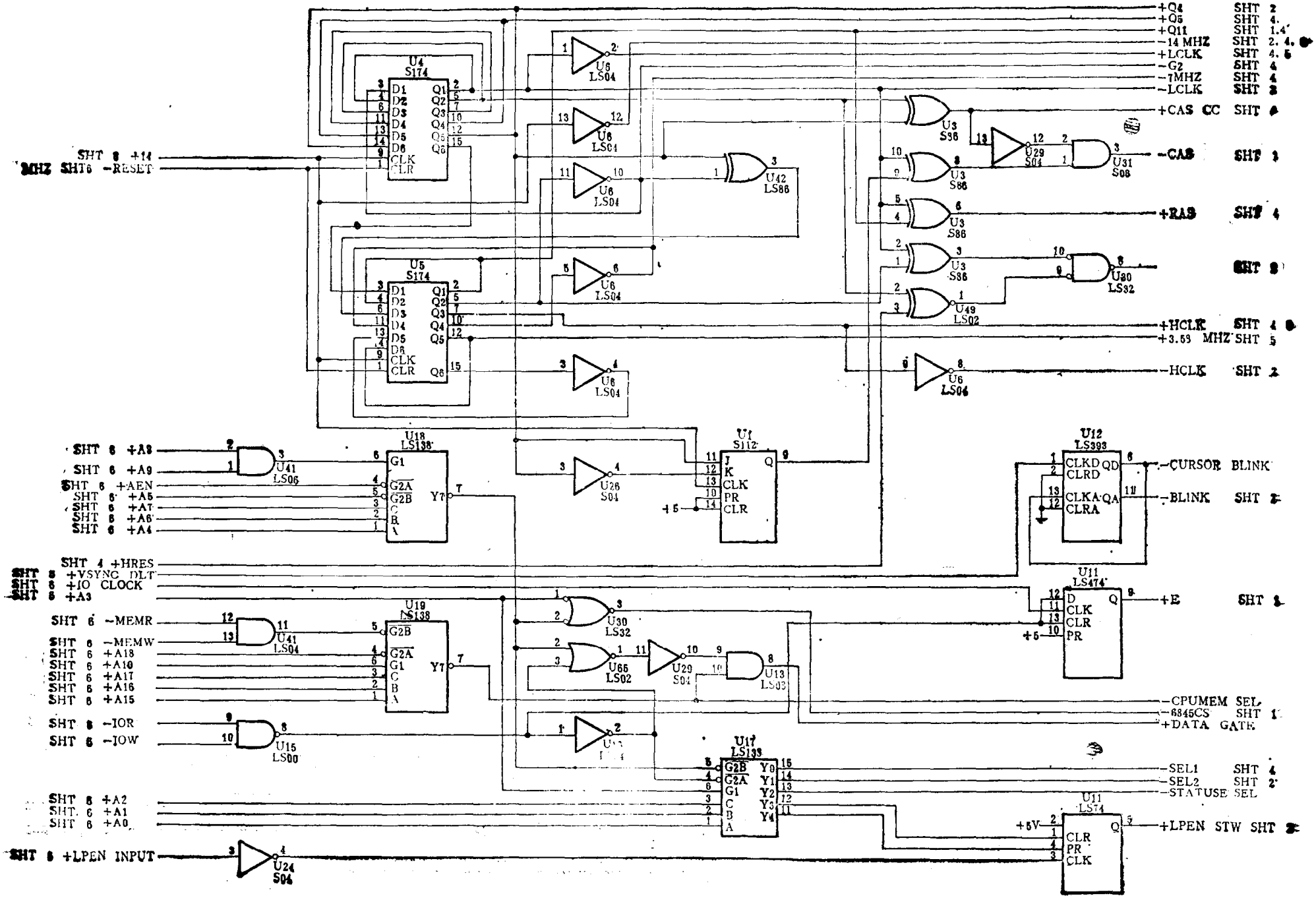
IBM 单色显示器逻辑线路图



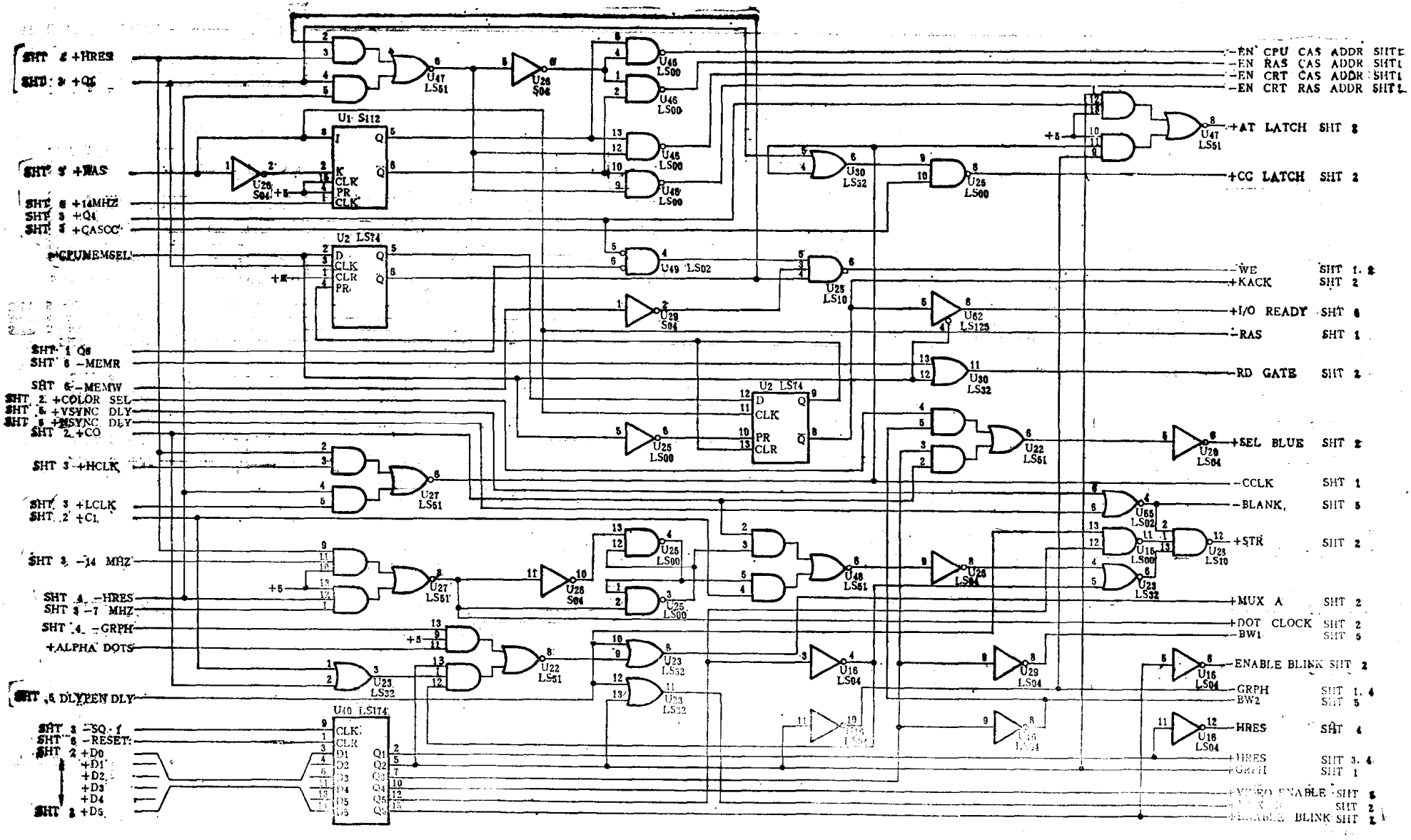
彩色/图形监视器转接器逻辑线路图之一



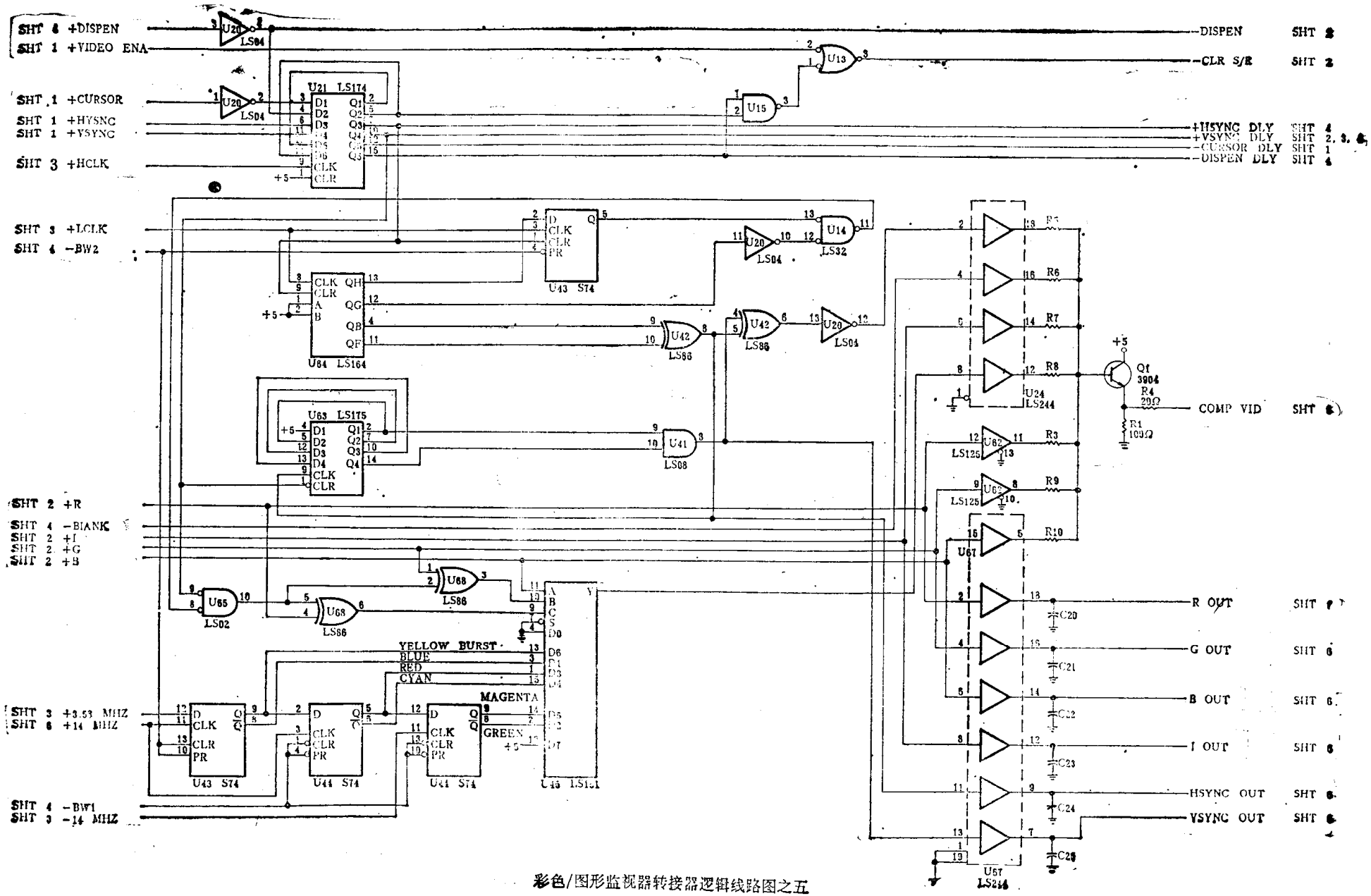
彩色/图形监视器转接器逻辑线路图之二



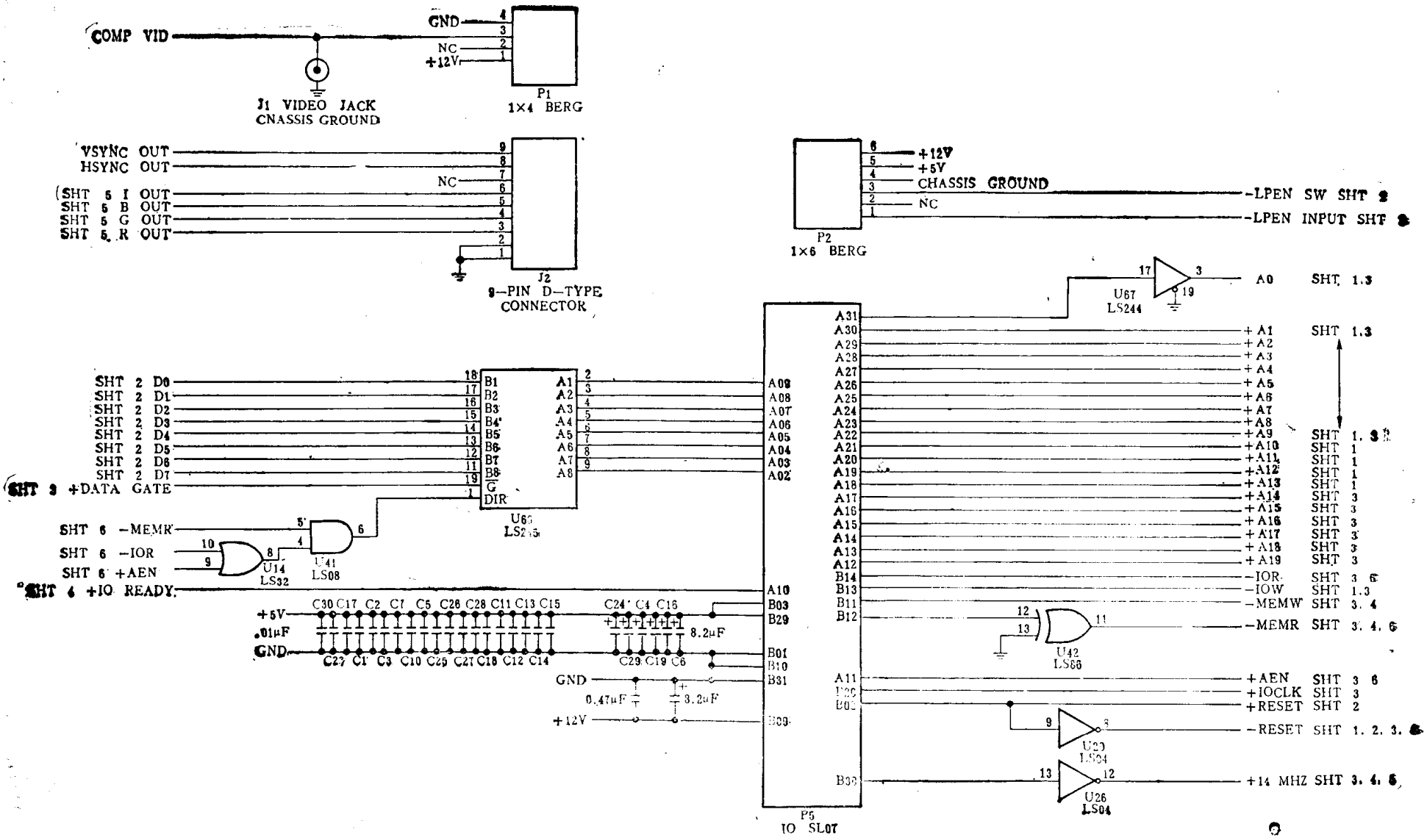
彩色/图形监视器转接器逻辑线路图之三



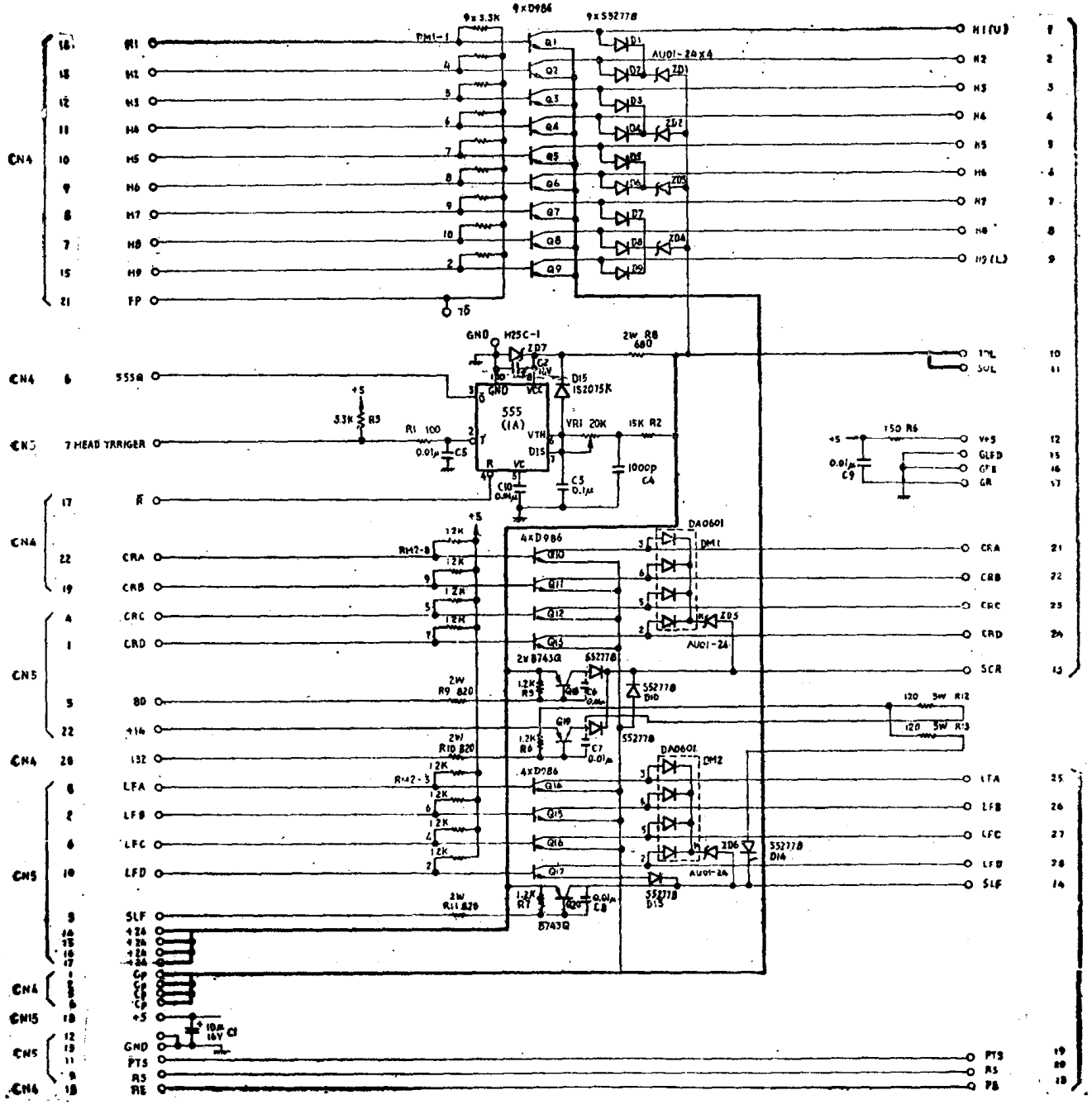
彩色/图形监视器转接器逻辑线路图之四



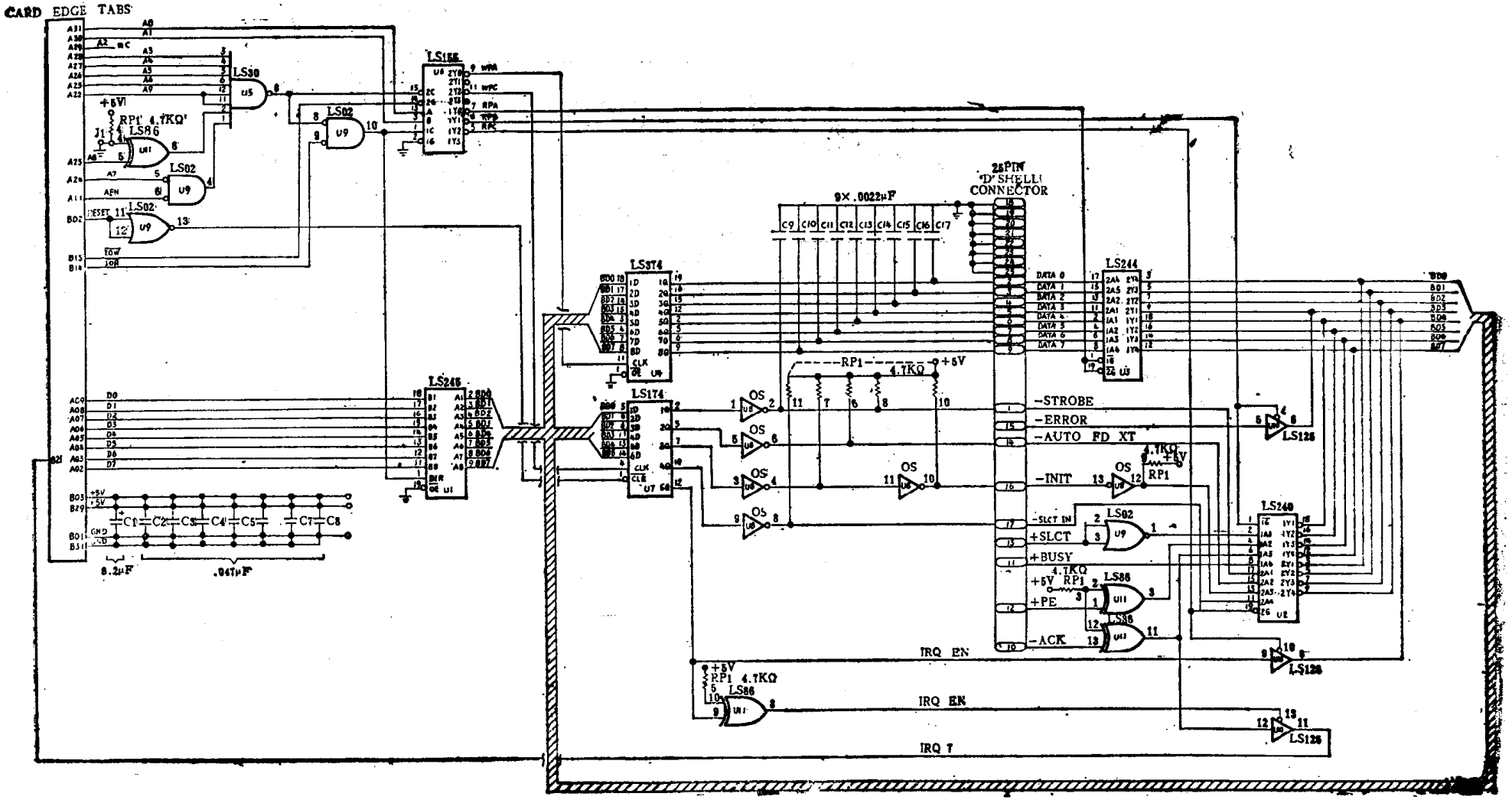
彩色/图形监视器转接器逻辑线路图之五



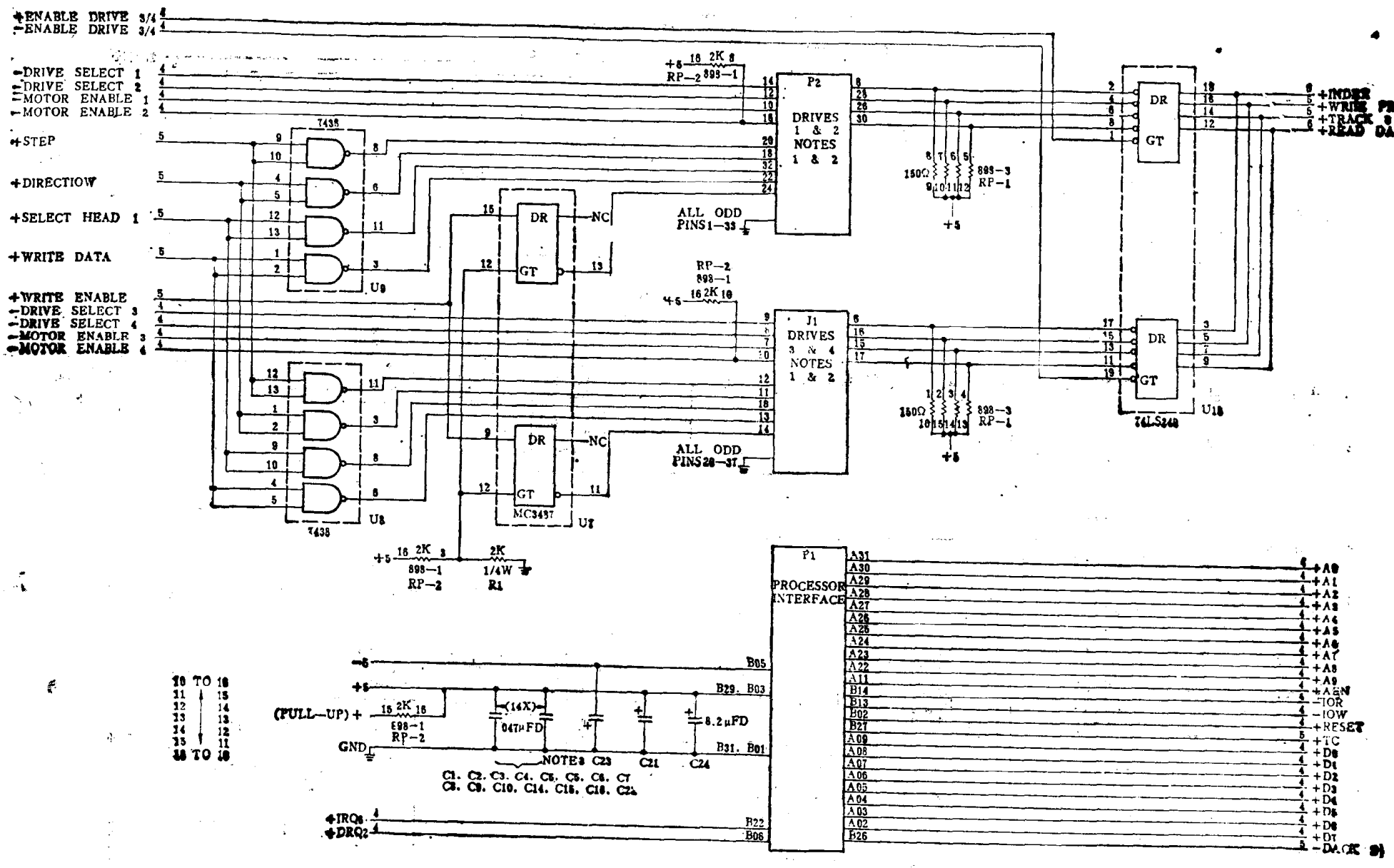
彩色/图形监视器转接器逻辑线路图之六



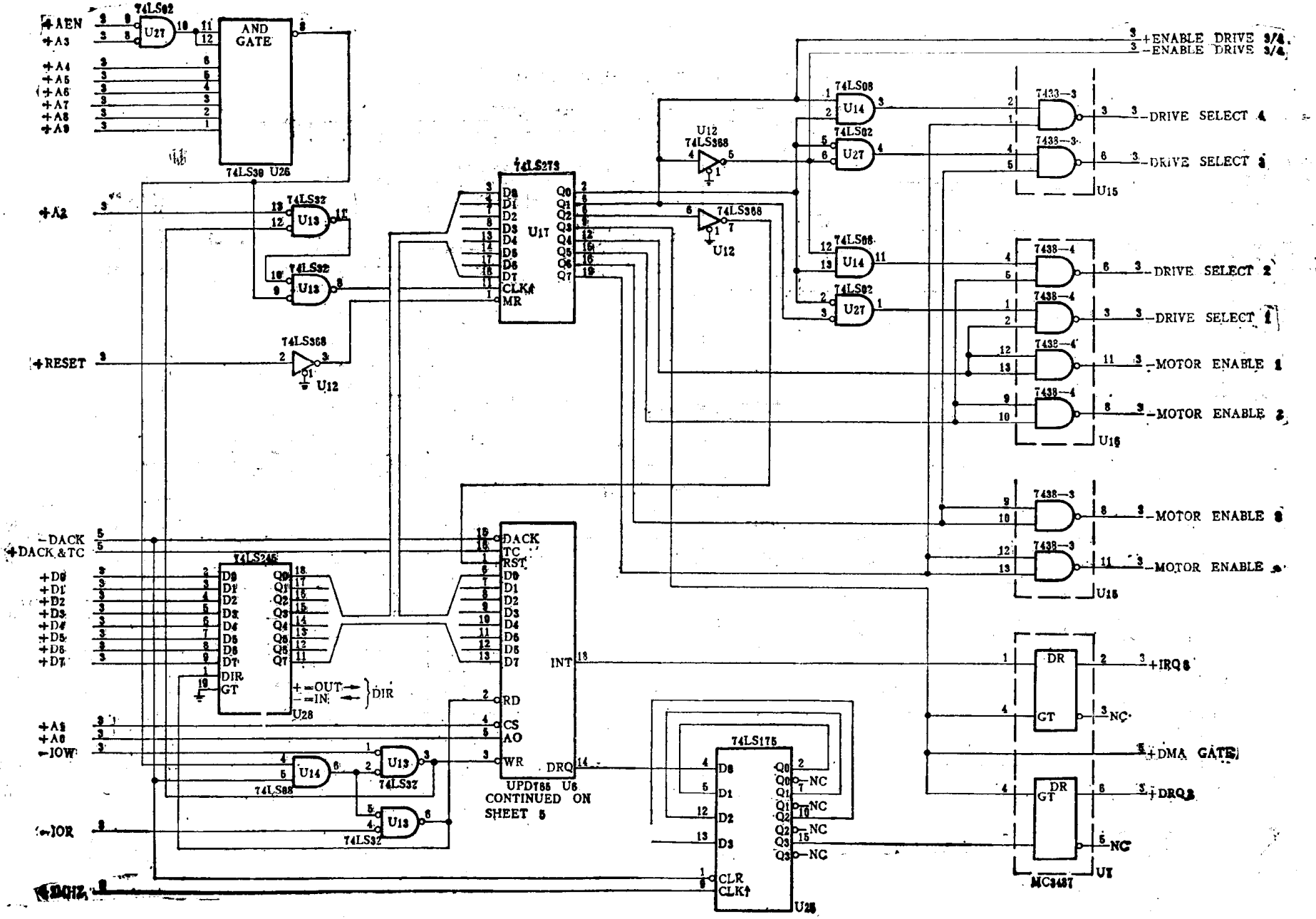
IBM 80 CPS 点阵打印机驱动线路



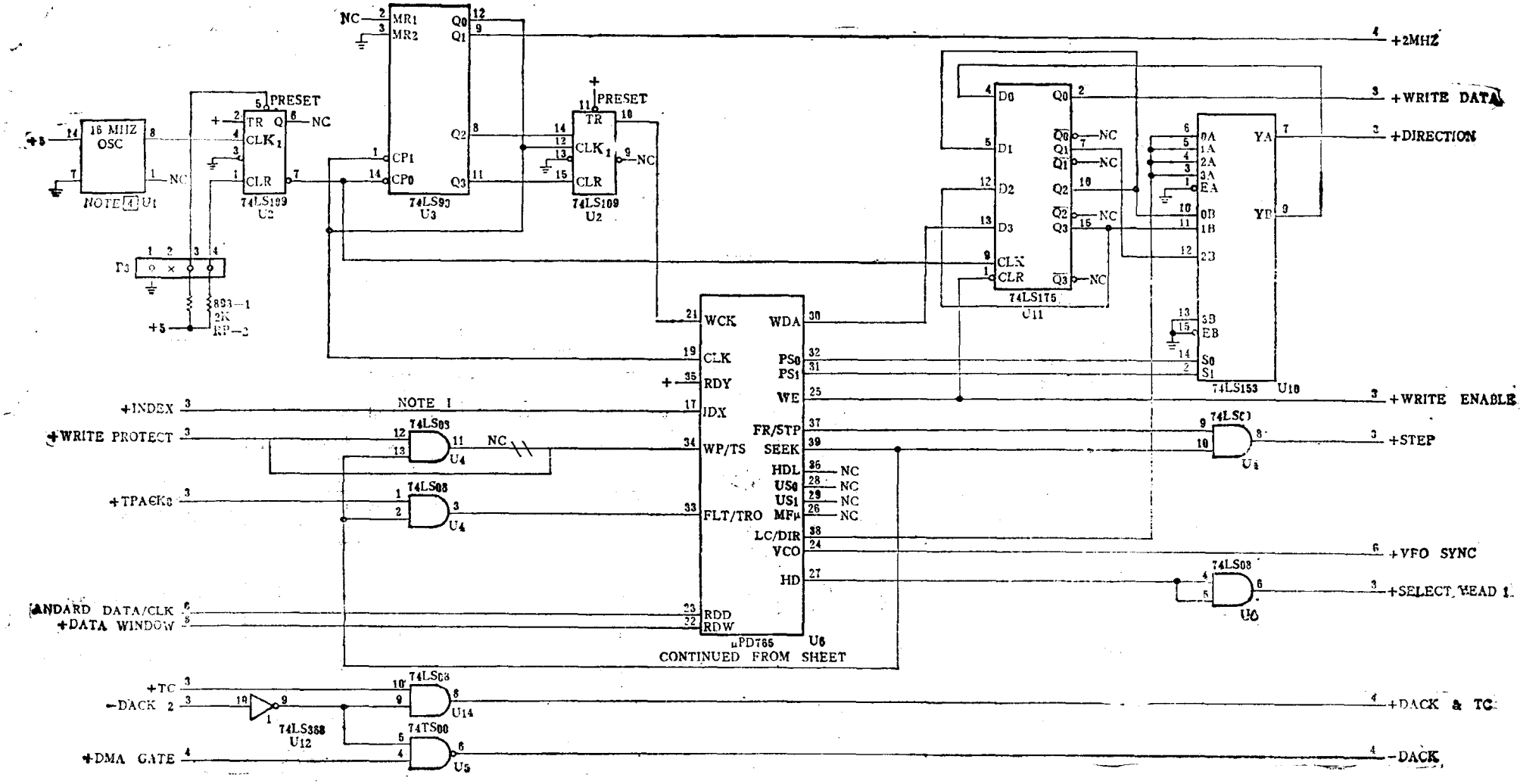
并行打印机转接器电路图



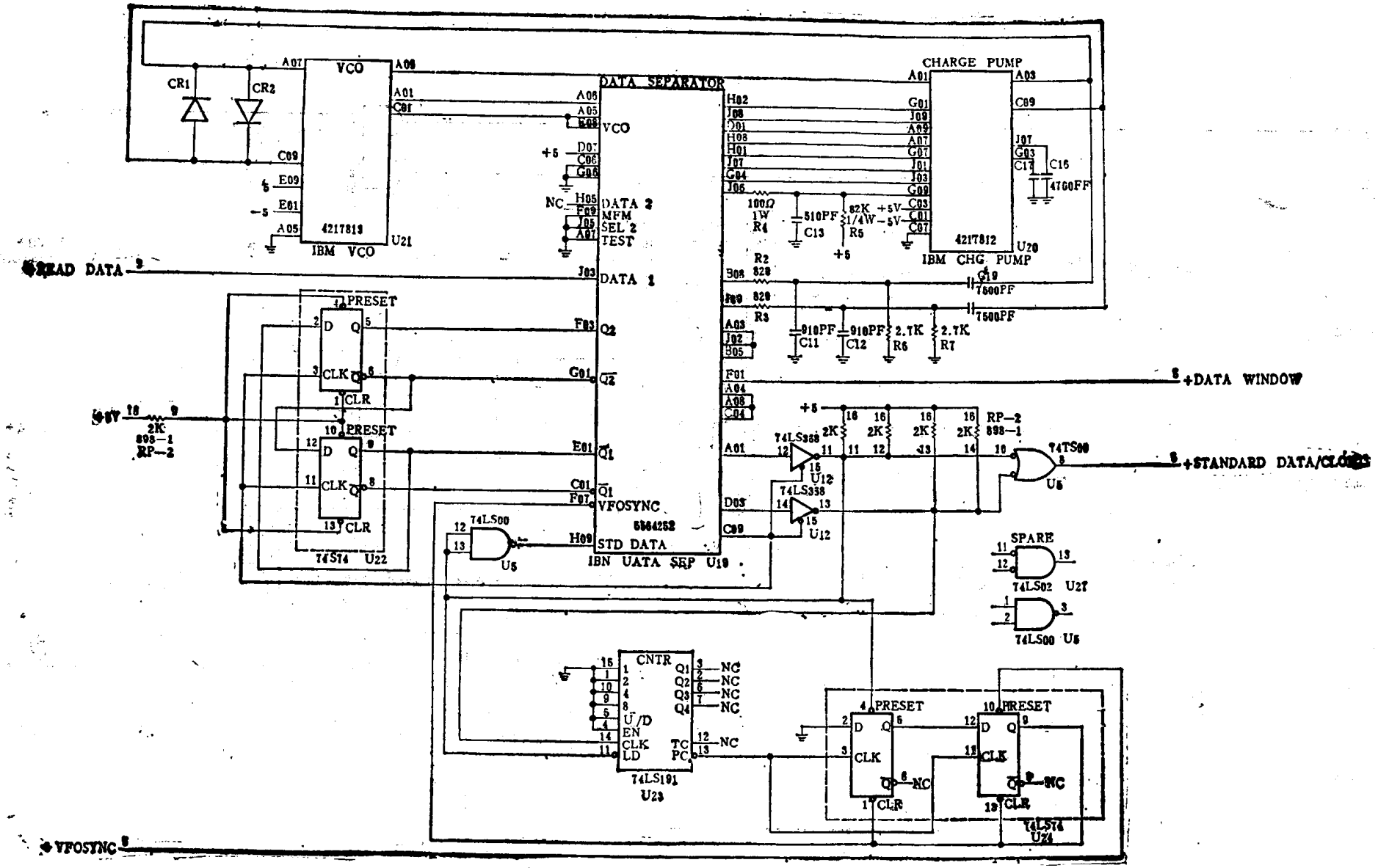
5-1/4 吋软盘机转接器逻辑线路图之三



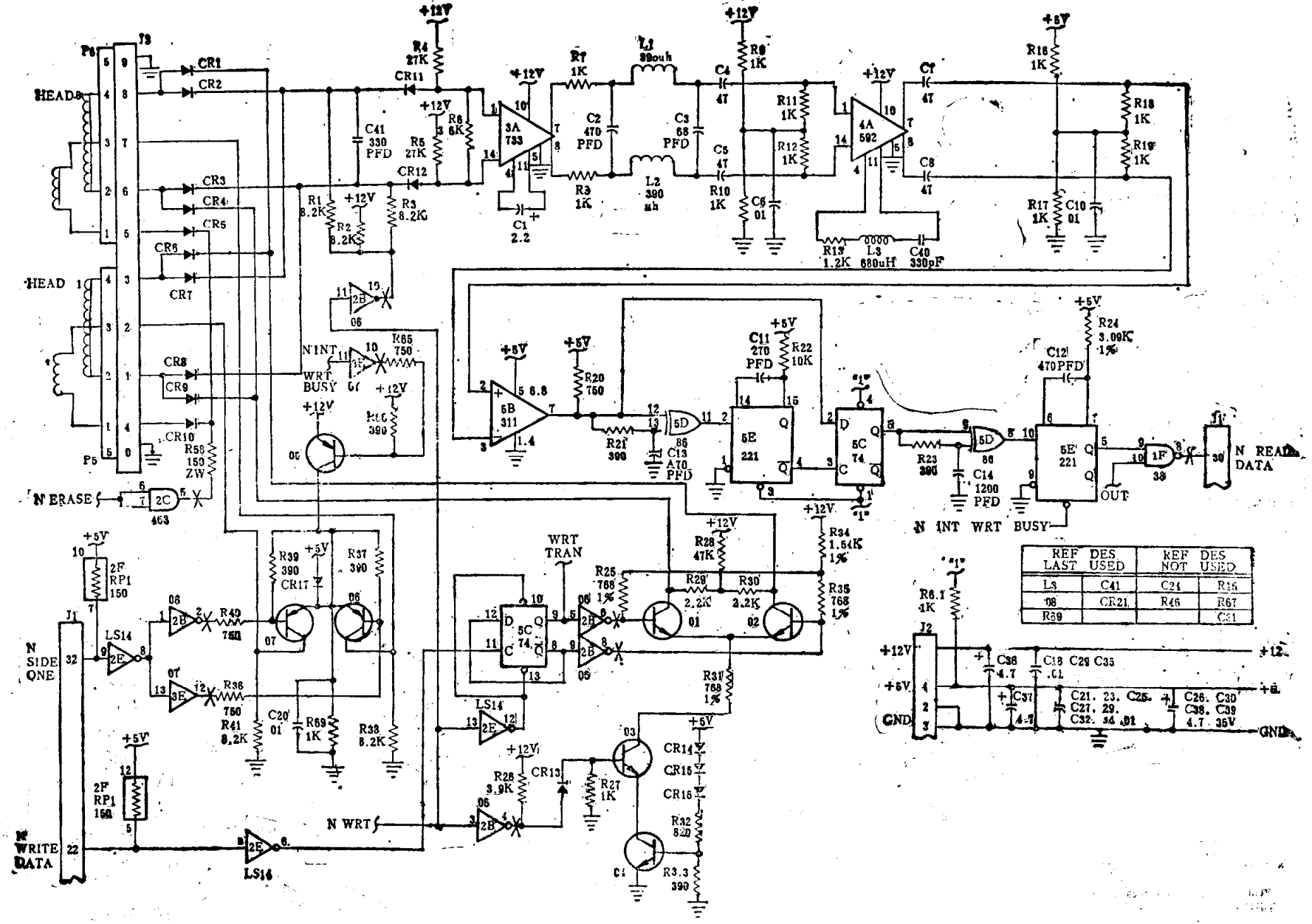
5-1/4 磁软盘机转接器线路图之四



5-1/4 时软盘机转接器线路图之五

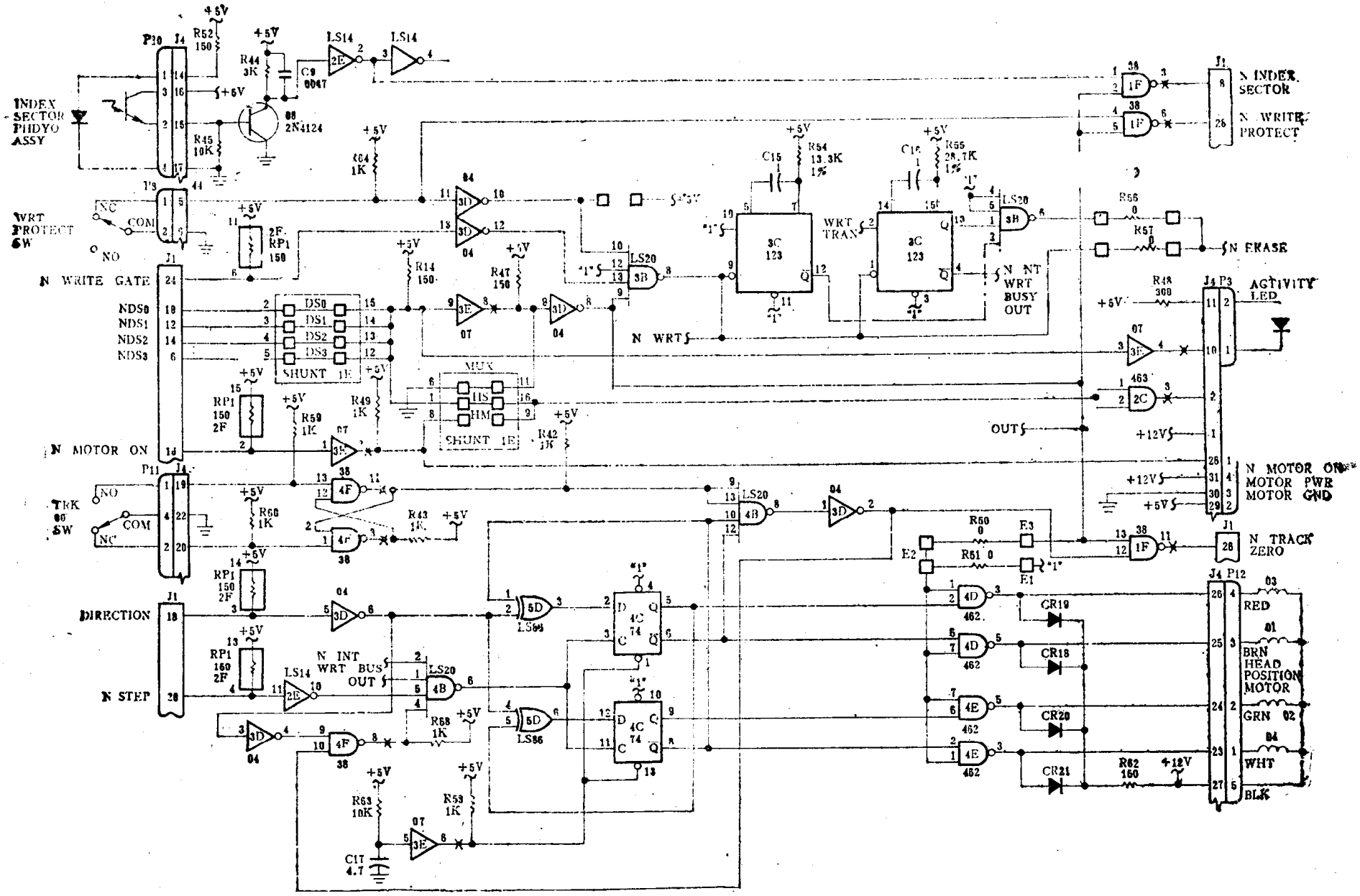


5-1/4 时软盘机转接器线路图之六

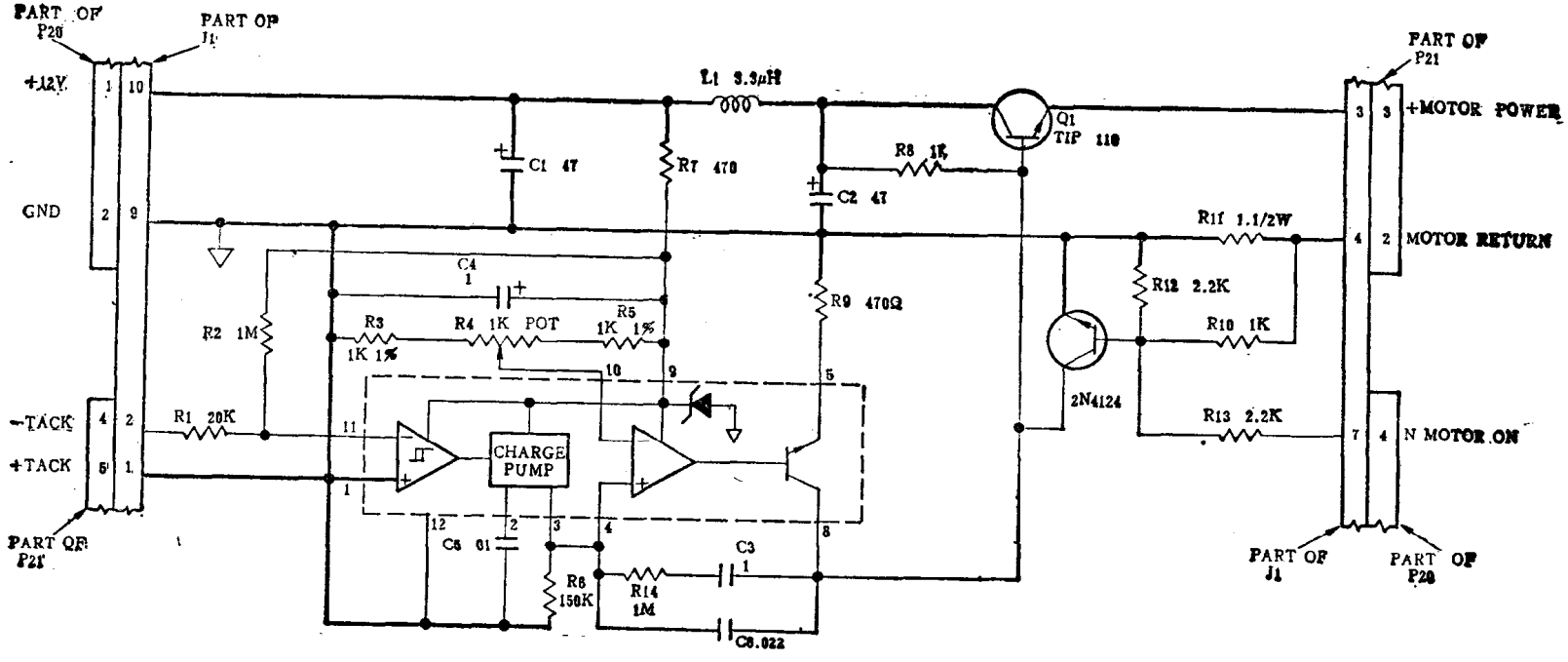


REF DES LAST USED		REF DES NOT USED	
L3	C41	C24	R16
08	CR21	R46	R67
R69			C31

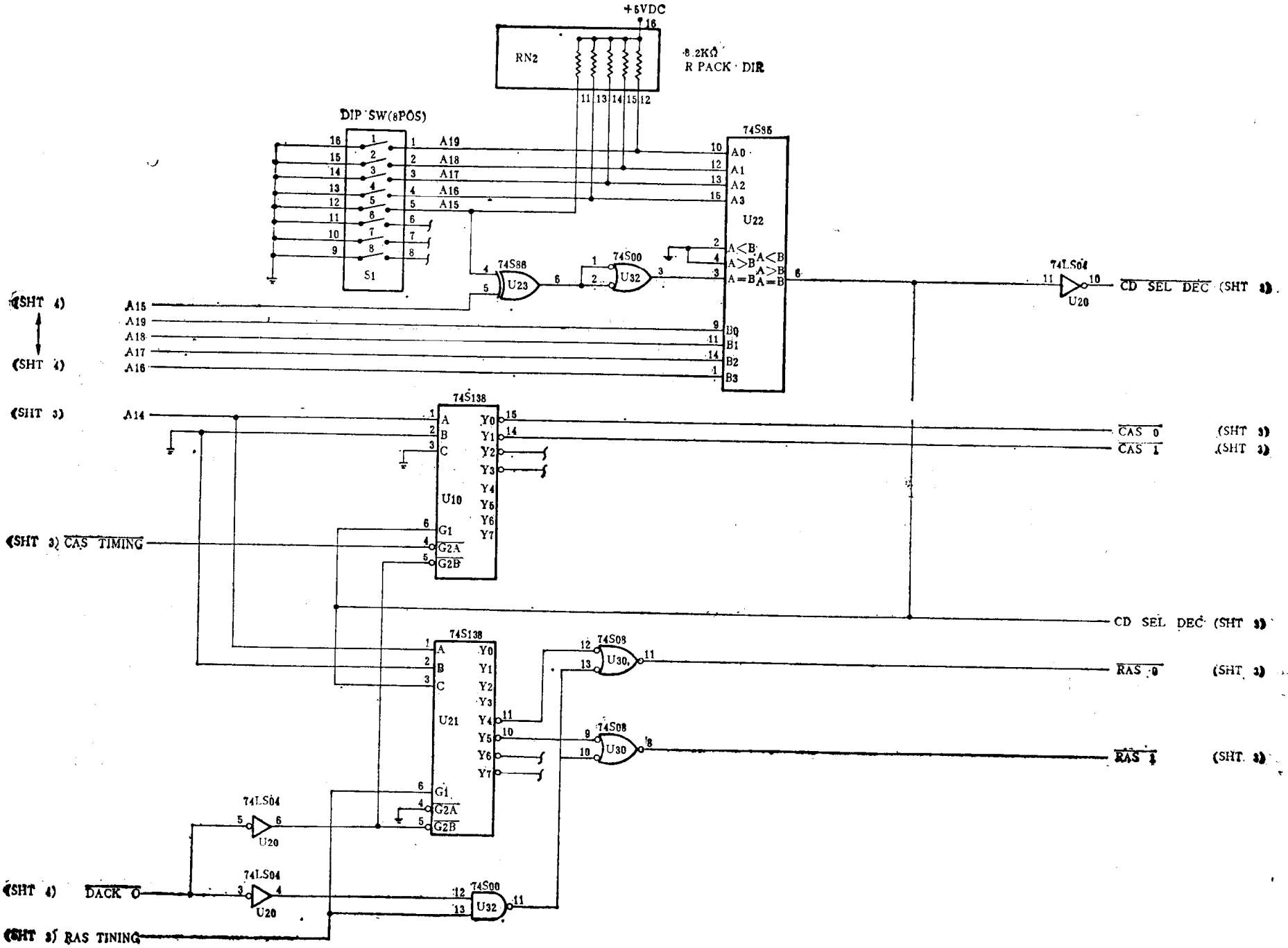
5-1/4 时软盘机驱动器线路图之一



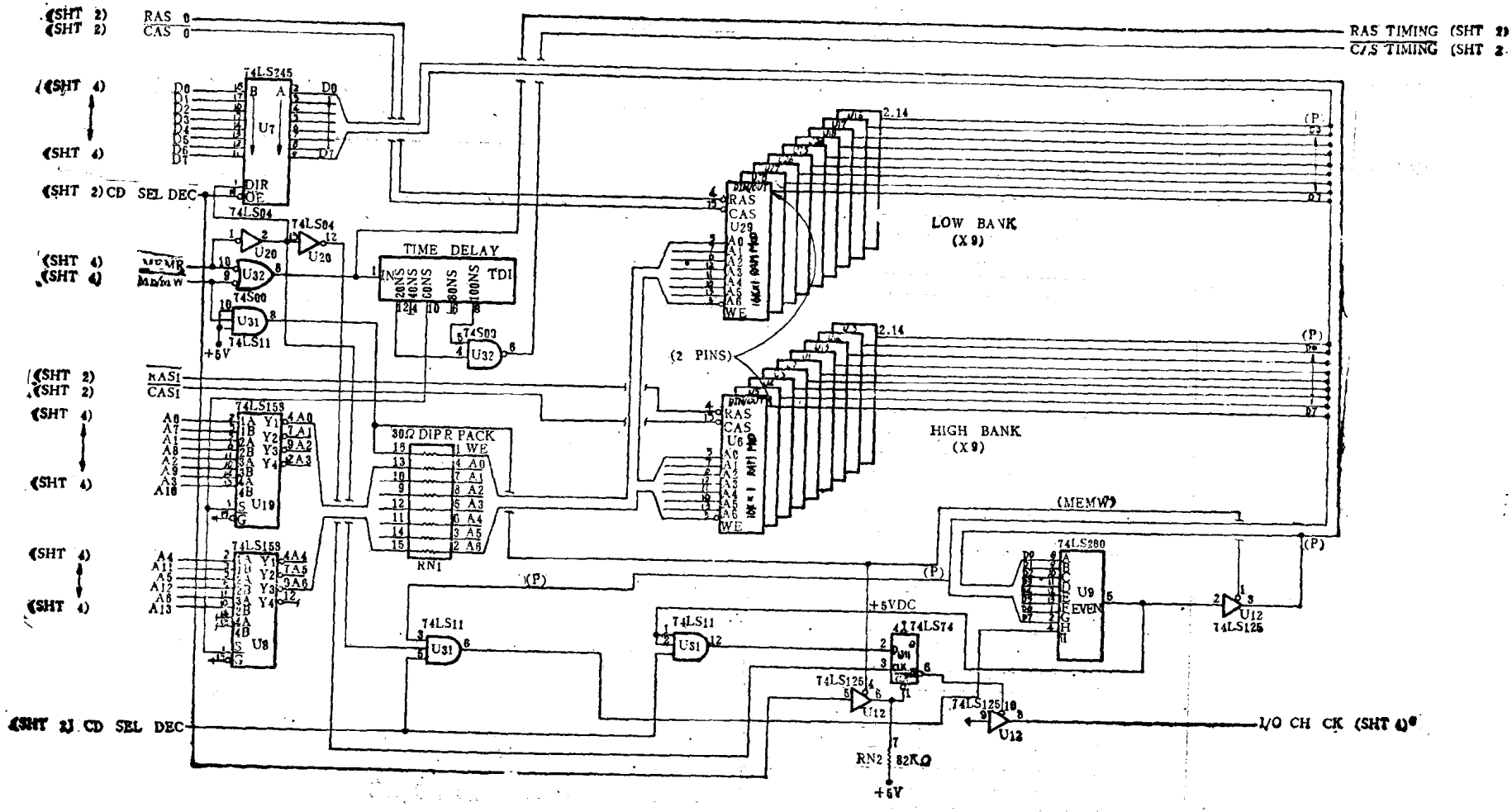
5-1/4 吋软盘机驱动器线路图之二



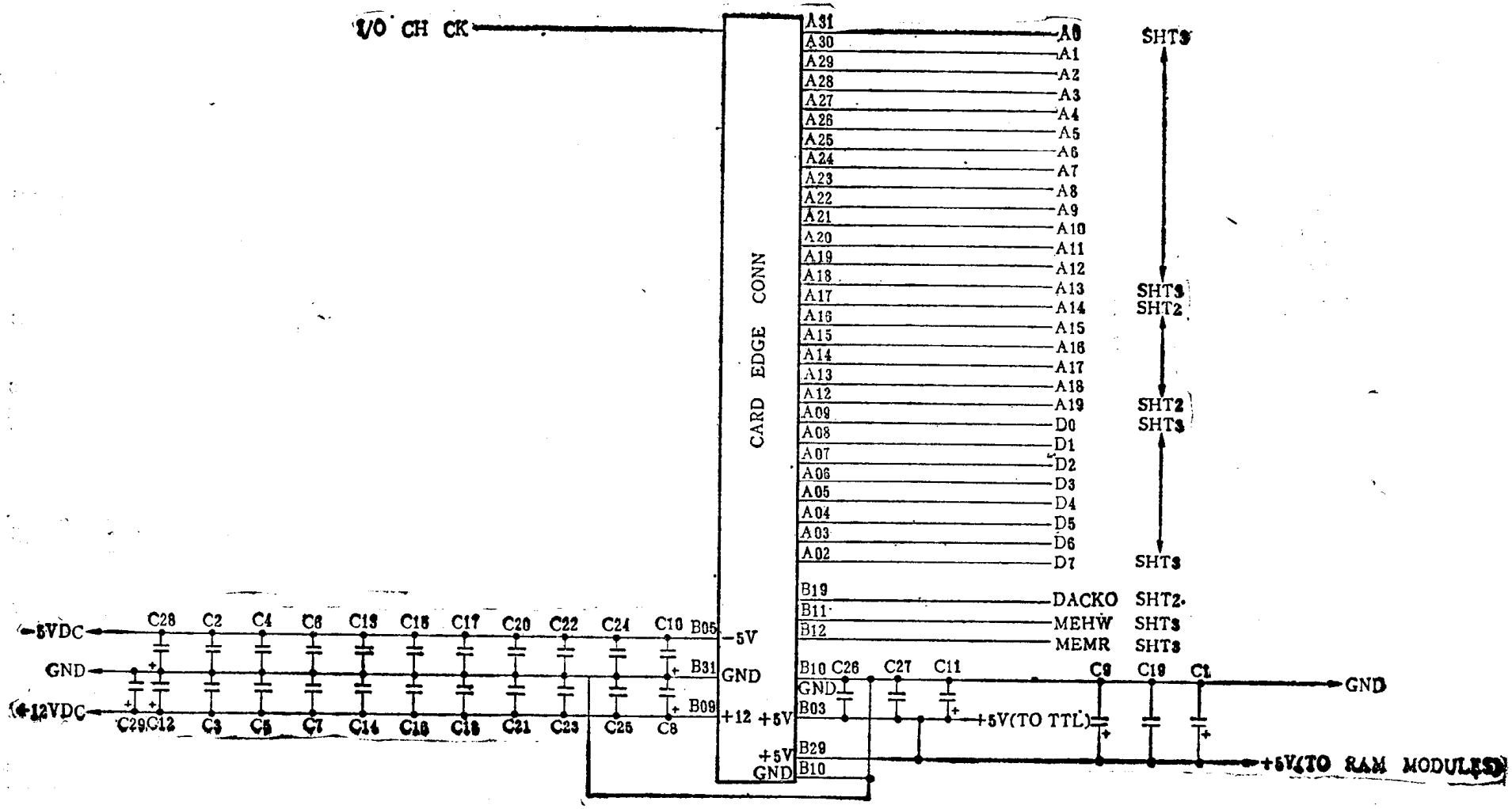
5-1/4 吋软盘机驱动器线路图之三



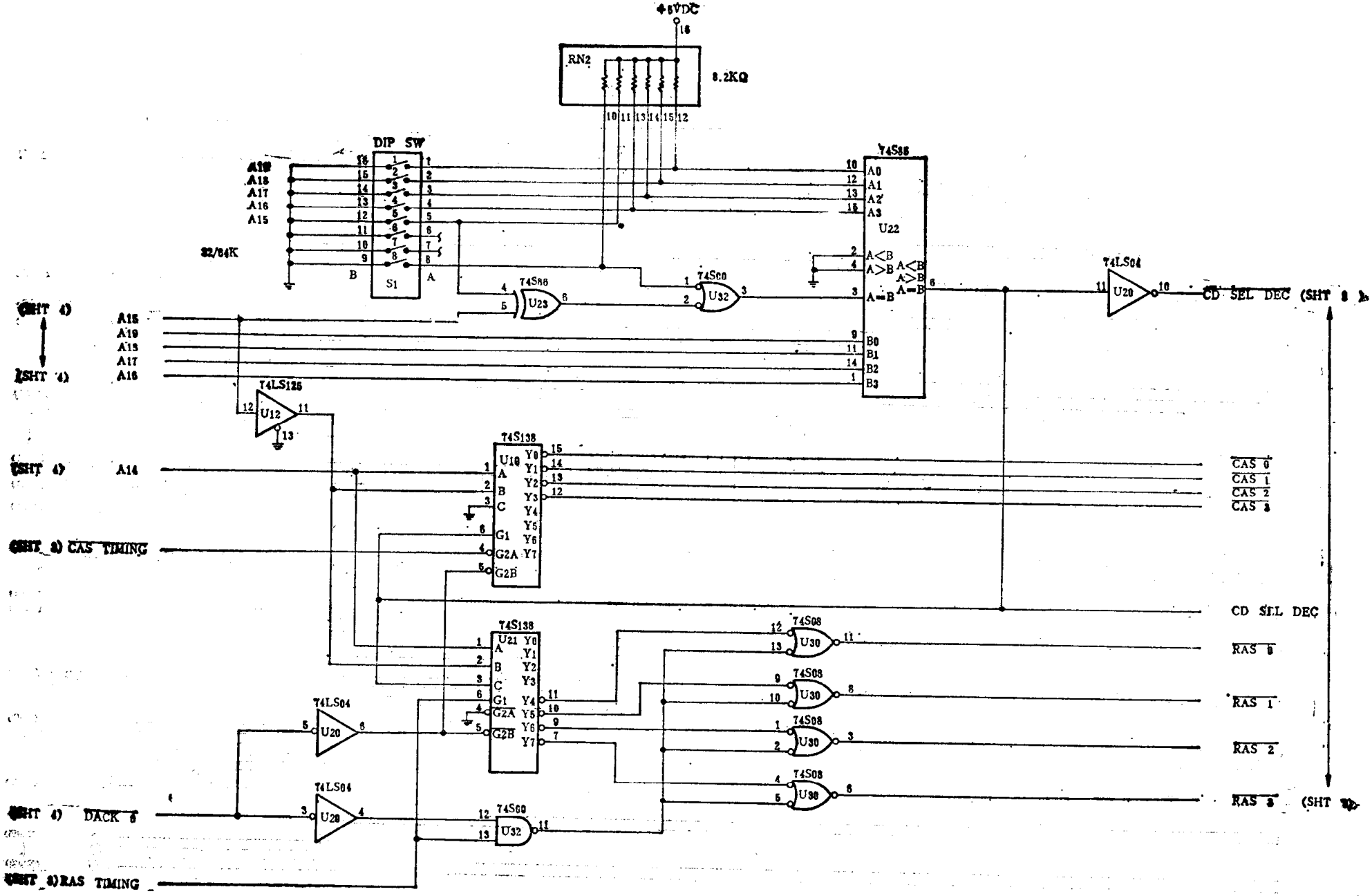
32 KB 存贮器扩展线路之二



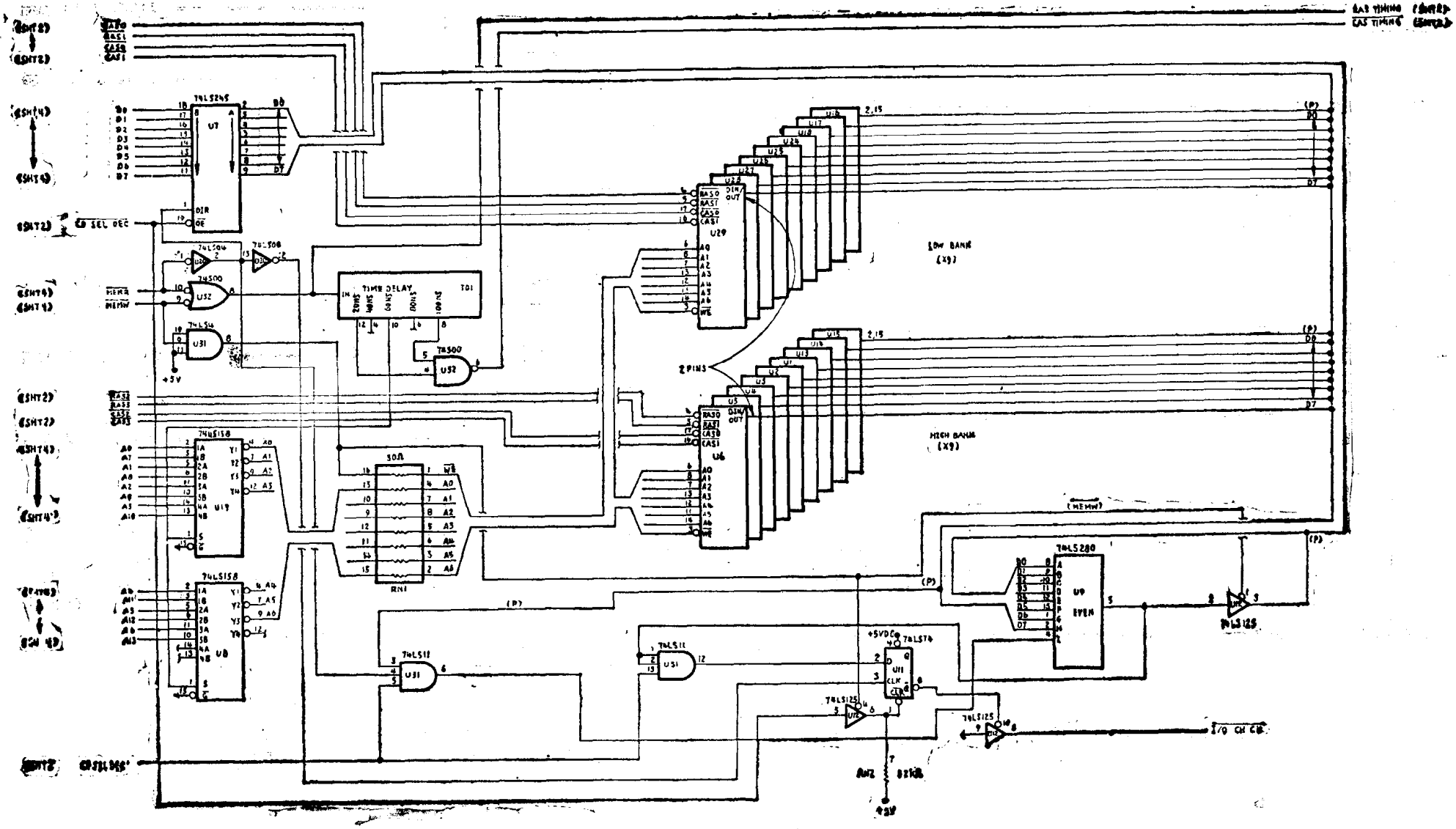
32 KB 存贮器扩展线路之三



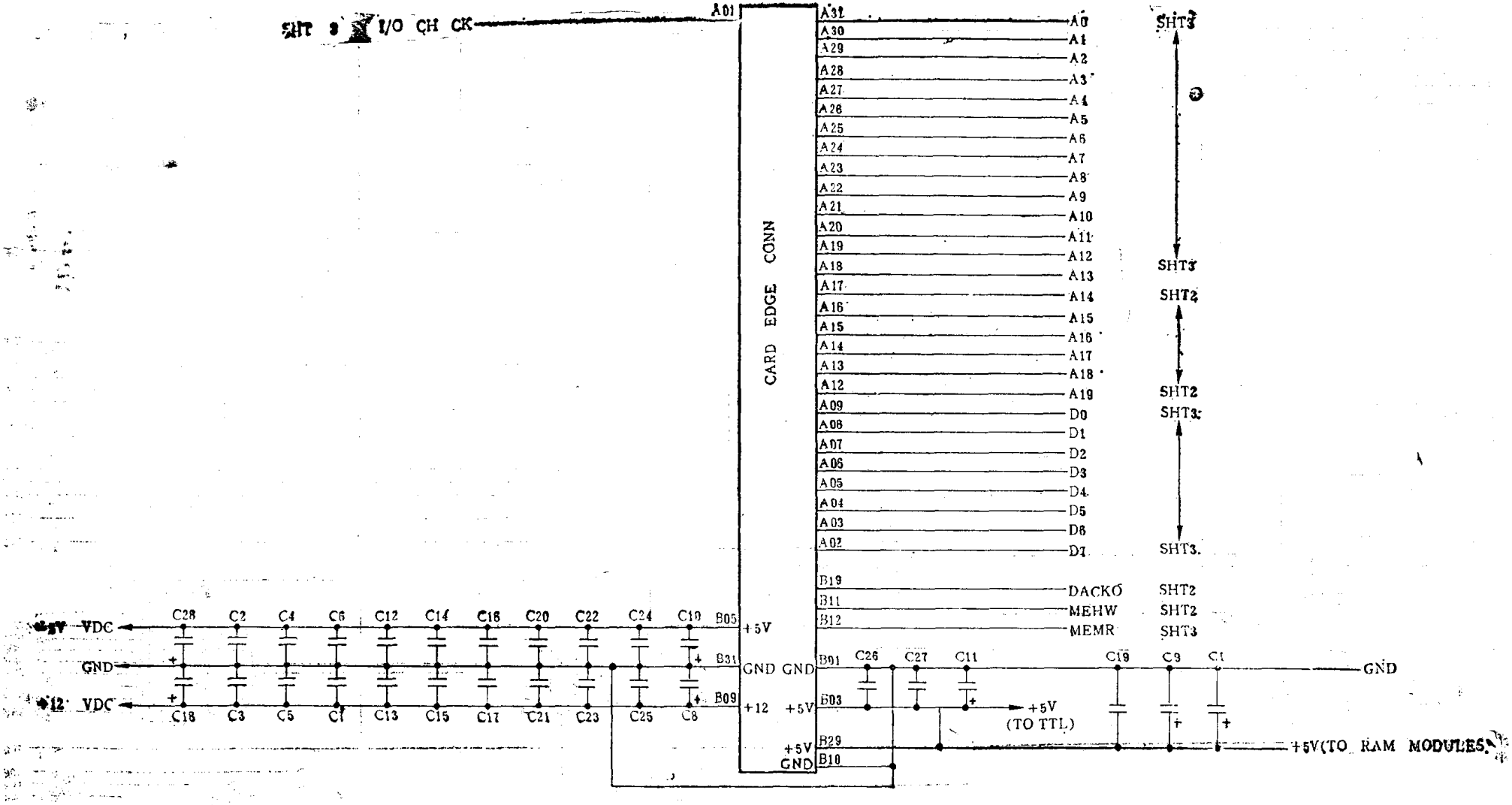
82 KB 存貯器扩展线路之四



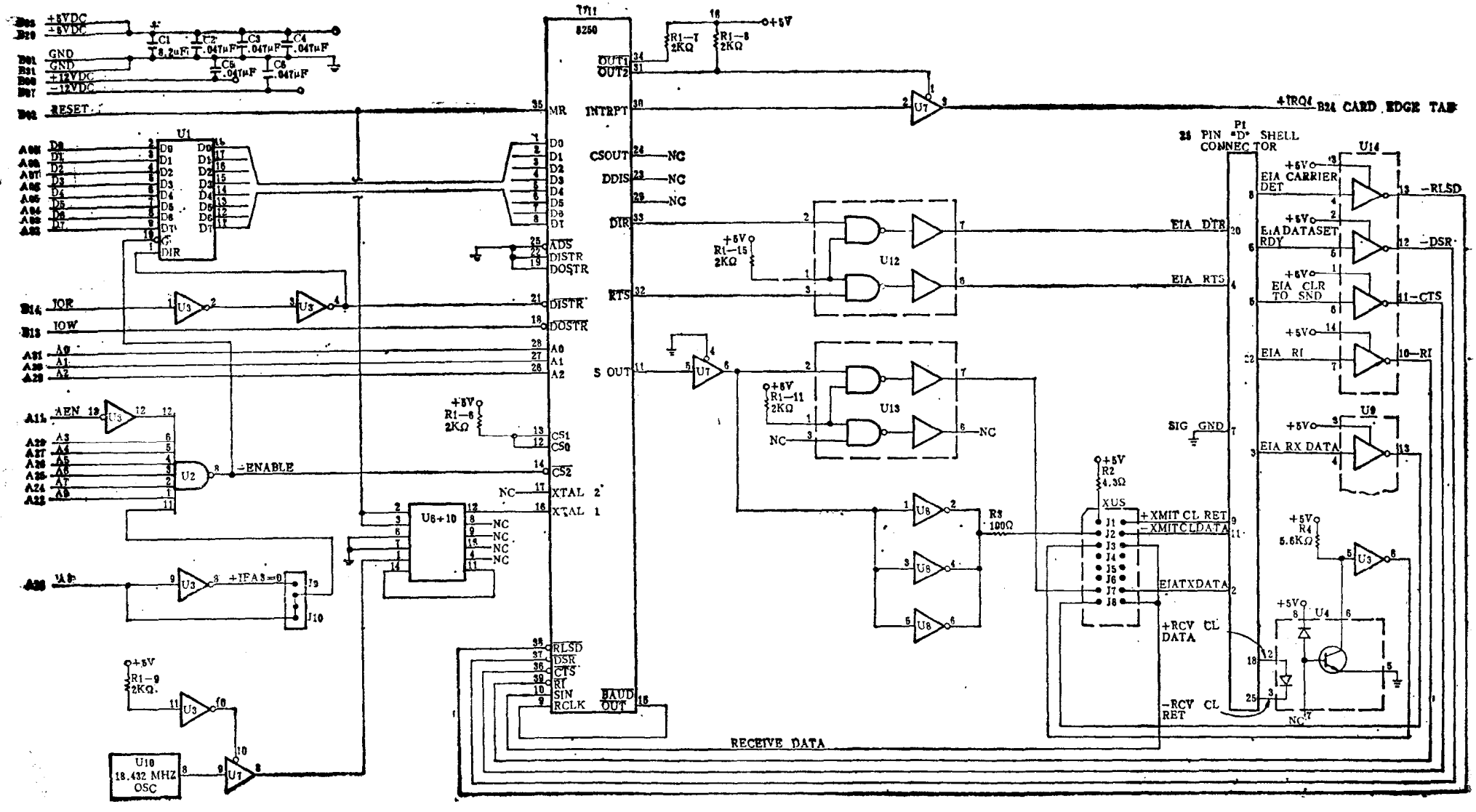
64 KB 存贮器扩展线路之一



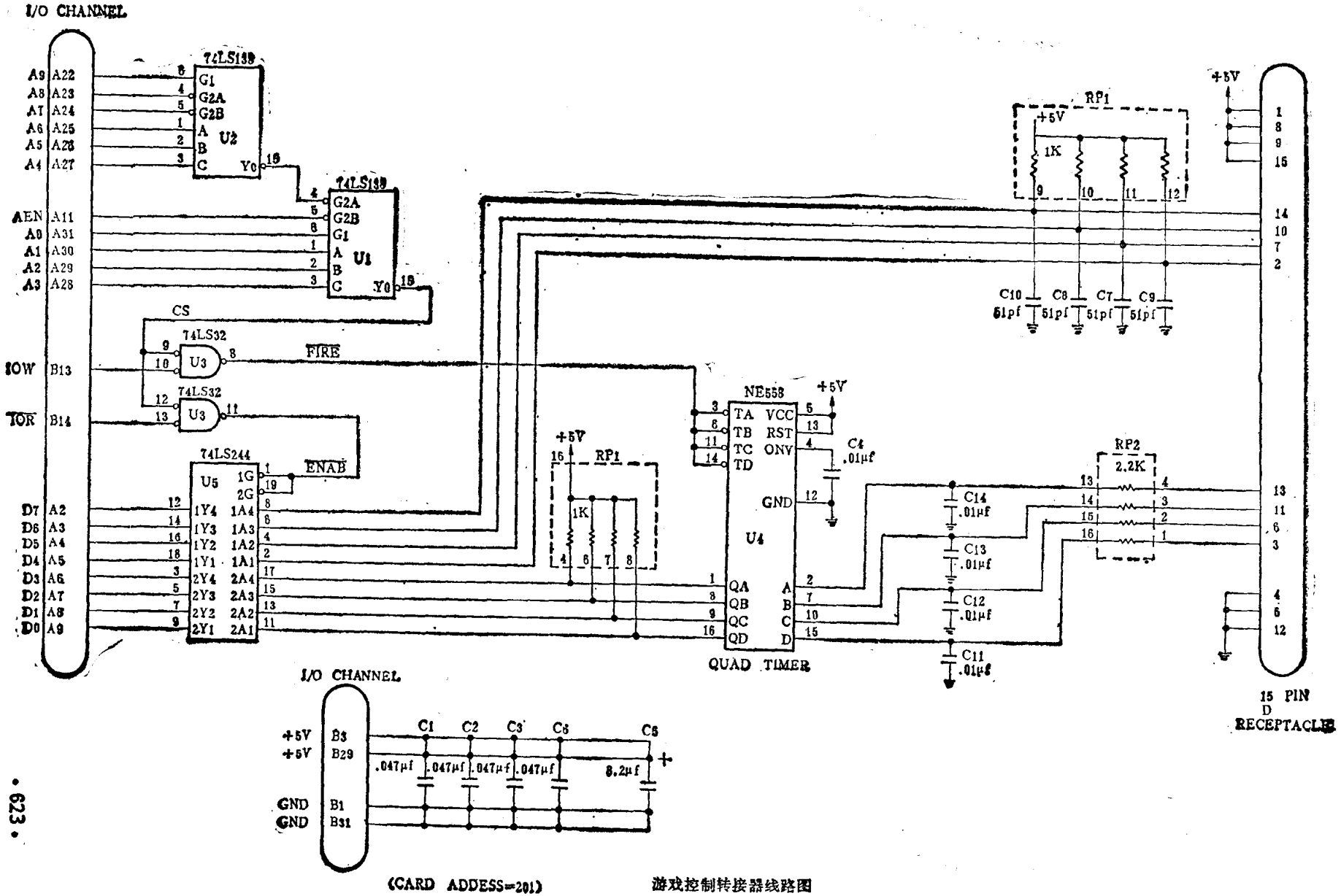
64 KB 存贮器扩展线路之三



64 KB 存贮器扩展线路之四


+5VDC
GND
+12VDC
-12VDC
RESET
C1 8.2uF
C2 .047uF
C3 .047uF
C4 .047uF
C5 .047uF
C6 .047uF
U11
8250
MR
INTRPT
OUT1
OUT2
CSOUT
DDIS
DIR
RTS
S OUT
ADS
DISTR
DOSTR
A0
A1
A2
CS1
CS0
CS2
XTAL 2
XTAL 1
BAUD OUT
RLSD
DSR
CTS
RI
SIN
RCLK
R1-7 2KΩ
R1-8 2KΩ
+5V
+IRQ4
B24 CARD EDGE TAB
U1
U2
U3
U4
U6+10
U7
U8
U9
U10 18.432 MHZ OSC
U12
U13
U14
IOR
IOW
AEN
+ENABLE
+IFAS=0
R1-9 2KΩ
R1-6 2KΩ
R1-11 2KΩ
R1-15 2KΩ
R2 4.3Ω
R3 100Ω
R4 5.6KΩ
XUS
P1
25 PIN "D" SHELL CONNECTOR
EIA DTR
EIA RTS
SIG GND
+XMIT CL RET
-XMITCLDATA
EIATXDATA
+RCV CL DATA
-RCV CL RET
EIA CARRIER DET
EIA DATASET RDY
EIA CLR TO SND
EIA RI
EIA RX DATA
-RLSD
-DSR
-CTS
-RI
RECEIVE DATA

异步通讯转接器线路图


I/O CHANNEL
74LS139
U2
U1
74LS32
U3
74LS244
U5
NE558
U4
QUAD TIMER
RP1
RP2
1K
2.2K
+5V
CS
FIRE
ENAB
IOW
IOR
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
.047μf
8.2μf
51pf
.01μf
15 PIN
D
RECEPTACLE
(CARD ADDESS=201)

游戏控制转接器线路图

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
00	0	BLANK (NULL)	CTRL2		黑	黑	不显示
01	1	☺	CTRLA		黑	兰	下划线
02	2	☻	CTRLB		黑	绿	一般
03	3	♥	CTRLC		黑	深兰	一般
04	4	◆	CTRLD		黑	红	一般
00	5	♣	CTRLE		黑	品红	一般
06	6	♠	CTRLF		黑	棕	一般
07	7	•	CTRLG		黑	淡灰	一般
08	8	■	CTRLH BACKSPACE SHIFT BACKSPACE		黑	深灰	不显示
09	9	○	CTRLI		黑	浅兰	高亮度 下划线
0A	10	◙	CTRLJ CTRL↲		黑	浅绿	高亮度
0B	11	♂	CTRLK		黑	浅绿	高亮度
0C	12	♀	CTRLL		黑	浅红	高亮度
0D	13	♪	CTRLM ↲ SHIFT ↲		黑	浅品红	高亮度
0E	14	♫	CTRL N		黑	黄	高亮度
0F	15	☼	CTRL O		黑	白	高亮度
10	16	►	CTRL P		兰	黑	一般
11	17	◄	CTRL Q		兰	兰	下划线
12	18	↕	CTRL R		兰	绿	一般
13	19	‼	CTRL S		兰	深兰	一般

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HFX	DEC	符号	按键	模式	背景色	背景	
14	20	¶	CTRLT		兰	红	一般
15	21	§	CTRLU			品红	一般
16	22	▬	CTRLV		兰	棕	一般
17	23	↨	CTRLW		兰	浅灰	一般
18	24	↑	CTRLX		兰	深灰	高亮度
19	25	↓	CTRLY		兰	浅兰	高亮度 下划线
1A	26	→	CTRLZ		兰	浅绿	高亮度
1B	27	←	CTRL[, ESC, SHIFT ESC, CTRL ESC		兰	兰	高亮度
1C	28	∟	CTRL\		兰	浅红	高亮度
1D	29	↔	CTRL]		兰	浅品红	高亮度
1E	30	▲	CTRL6		兰	黄	高亮度
1F	31	▼	CTRL-		兰	白	高亮度
20	32	BLANK (SPACE	SPACE BAR, SHIFT SPACE CTRL SPACE, ALT SPACE		绿	黑	一般
21	33	!	!	移位	绿	兰	下划线
22	34	"	"	移位	绿	绿	一般
23	35	#	#	移位	绿	深兰	一般
24	36	$	$	移位	绿	红	一般
25	37	%	%	移位	绿	品红	一般
26	38	&	&	移位	绿	棕	一般
27	39	'	'		绿	浅灰	一般

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
28	40	(	(	移位	绿	深灰	高亮度
29	41	)	)	移位	绿	浅兰	高亮度 下划线
2A	42	*	*	注 1	绿	浅绿	高亮度
2B	43	+	+	移位	绿	兰	高亮度
2C	44	'	'		绿	浅红	高亮度
2D	45	-	-		绿	浅品红	高亮度
2E	46	.	.	注 2	绿	黄	高亮度
2F	47	/	/		绿	白	高亮度
30	48	0	0	注 3	青	黑	一般
31	49	1	1	注 3	青	兰	下划线
32	50	2	2	注 3	青	绿	一般
33	51	3	3	注 3	青		一般
34	52	4	4	注 3	青	红	一般
35	53	5	5	注 3	青	品红	一般
36	54	6	6	注 3	青	棕	一般
37	55	7	7	注 3	青	浅灰	一般
38	56	8	8	注 3	青	深灰	高亮度
39	57	9	9	注 3	青	浅兰	高亮度 下划线
3A	58	:		移位	青	浅绿	高亮度
3B	59	.			青	浅青	高亮度
3C	60	<	<	移位	青	浅红	高亮度
3D	61	=	=		青	浅品红	高亮度

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
3E	62	>	>	移位	青	黄	高亮度
3F	63	?	?	移位	青	白	高亮度
40	64	@	@	移位	红	黑	一般
41	65	A	A	注 4	红	兰	下划线
42	66	B	B	注 4	红	绿	一般
43	67	C	C	注 4	红	青	一般
44	68	D	D	注 4	红	红	一般
45	69	E	E	注 4	红	品红	一般
46	70	F	F	注 4	红	棕	一般
47	71	G	G	注 4	红	浅灰	一般
48	72	H	H	注 4	红	深灰	高亮度
49	73	I	I	注 4	红	浅兰	高亮度 下划线
4A	74	J	J	注 4	红	浅绿	高亮度
4B	75	K	K	注 4	红	浅青	高亮度
4C	76	L	L	注 4	红	浅红	高亮度
4D	77	M	M	注 4	红	浅品红	高亮度
4E	78	N	N	注 4	红	黄	高亮度
4F	79	O	O	注 4	红	白	高亮度
50	80	P	P	注 4	品红	黑	一般
51	81	Q	Q	注 4	品红	兰	下划线
52	82	R	R	注 4	品红	绿	一般
53	83	S	S	注 4	品红	青	一般
54	84	T	T	注 4	品红	红	一般

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
55	85	U	U	注 4	品红	品红	一般
56	86	V	V	注 4	品红	棕	一般
57	87	W	W	注 4	品红	浅灰	一般
58	88	X	X	注 4	品红	深灰	高亮度
59	89	Y	Y	注 4	品红	浅兰	高亮度 下划线
5A	90	Z	Z	注 4	品红	浅绿	高亮度
5B	91	[	[		品红	浅青	高亮度
5C	92	\	\		品红	浅红	高亮度
5D	93	]	]		品红	浅品红	高亮度
5E	94	^	^	移位	品红	黄	高亮度
5F	95	_	_	移位	品红	白	高亮度
60	96	`	`		黄	黑	一般
61	97	a	a	注 5	黄	兰	下划线
62	98	b	b	注 5	黄	绿	一般
63	99	c	c	注 5	黄	青	一般
64	100	d	d	注 5	黄	红	一般
65	101	e	e	注 5	黄	品红	一般
66	102	f	f	注 5	黄	棕	一般
67	103	g	g	注 5	黄	浅灰	一般
68	104	h	h	注 5	黄	深灰	高亮度
69	105	i	i	注 5	黄	浅兰	高亮度 下划线
6A	106	j	j	注 5	黄	浅绿	高亮度
6B	107	k	k	注 5	黄	浅青	高亮度

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
6C	108	l	l	注 5	黄	浅红	高亮度
6D	109	m	m	注 5	黄	浅品红	高亮度
6E	110	n	n	注 5	黄	黄	高亮度
6F	111	o	o	注 5	黄	白	高亮度
70	112	p	p	注 5	白	黑	视频翻转
71	113	q	q	注 5	白	兰	下划线
72	114	r	r	注 5	白	绿	一般
73	115	s	s	注 5	白	青	一般
74	116	t	t	注 5	白	红	一般
75	117	u	u	注 5	白	品红	一般
76	118	v	v	注 5	白	棕	一般
77	119	w	w	注 5	白	浅灰	一般
78	120	x	x	注 5	白	深灰	视频翻转
79	121	y	y	注 5	白	浅兰	高亮度 下划线
7A	122	z	z	注 5	白	浅绿	高亮度
7B	123	{	{	移位	白	浅青	高亮度
7C	124	¦	¦	移位	白	浅红	高亮度
7D	125	}	}	移位	白	淡品红	高亮度
7E	126	~	~	移位	白	黄	高亮度
7F	127	△	CTRL←		白	白	高亮度
80 H-FFH 在彩色、单色模式下可闪烁							
80	128	Ç	ALT 128	注 6	黑	黑	不显示
81	129	ü	ALT 129	注 6	黑	兰	下划线

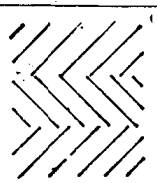
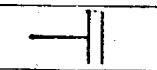
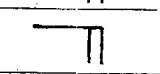
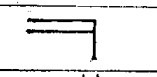
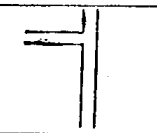
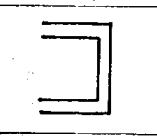
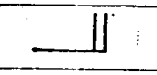
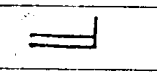
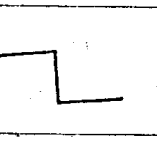
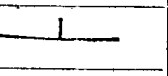
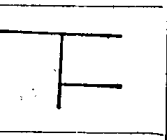
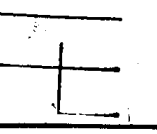
（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
82	130	é	ALT 130	注 6	黑	绿	一般
83	131	â	ALT 131	注 6	黑	青	一般
84	132	ä	ALT 132	注 6	黑	红	一般
85	133	à	ALT 133	注 6	黑	品红	一般
86	134	å	ALT 134	注 6	黑	棕	一般
87	135	ç	ALT 135	注 6	黑	浅灰	一般
88	136	ê	ALT 136	注 6	黑	深灰	不显示
89	137	ë	ALT 137	注 6	黑	浅兰	高亮度 下划线
8A	138	è	ALT 138	注 6	黑	浅绿	高亮度
8B	139	ï	ALT 139	注 6	黑	浅青	高亮度
8C	140	î	ALT 140	注 6	黑	浅红	高亮度
8D	141	ì	ALT 141	注 6	黑	浅品红	高亮度
8E	142	Ā	ALT 142	注 6	黑	黄	高亮度
8F	143	Å	ALT 143	注 6	黑	白	高亮度
90	144	É	ALT 144	注 6	兰	黑	一般
91	145	æ	ALT 145	注 6	兰	兰	下划线
92	146	FE	ALT 146	注 6	兰	绿	一般
93	147	ô	ALT 147	注 6	兰	青	一般
94	148	ö	ALT 148	注 6	兰	红	一般
95	149	ò	ALT 149	注 6	兰	品红	一般
96	150	û	ALT 150	注 6	兰	棕	一般
97	151	ù	ALT 151	注 6	兰	浅灰	一般
98	152	ÿ	ALT 152	注 6	兰	深灰	高亮度

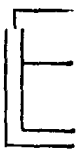
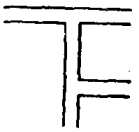
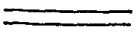
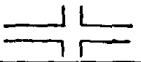
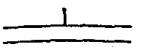
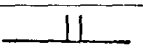
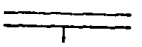
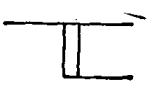
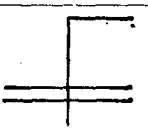
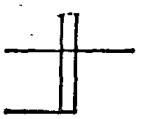
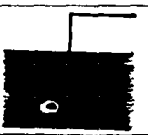
（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
99	153	ö	ALT 153	注 6	兰	浅兰	高亮度 下划线
9A	154	ü	ALT 154	注 6	兰	浅绿	高亮度
9B	155	Ç	ALT 155	注 6	兰	兰	高亮度
9C	156	£	ALT 156	注 6	兰	淡红	高亮度
9D	157	¥	ALT 157	注 6	兰	淡品红	高亮度
9E	158	Pts	ALT 158	注 6	兰	黄	高亮度
9F	159	f	ALT 159	注 6	兰	白	高亮度
A0	160	á	ALT 160	注 6	绿	黑	一般
A1	161	í	ALT 161	注 6	绿	兰	下划线
A2	162	ó	ALT 162	注 6	绿	绿	一般
A3	163	ú	ALT 163	注 6	绿	深兰	一般
A4	164	ñ	ALT 164	注 6	绿	红	一般
A5	165	Ñ	ALT 165	注 6	绿	品红	一般
A6	166	ª	ALT 166	注 6	绿	棕	一般
A7	167	º	ALT 167	注 6	绿	淡灰	一般
A8	168	¿	ALT 168	注 6	绿	深灰	高亮度
A9	169	⌐	ALT 169	注 6	绿	淡兰	高亮度 下划线
AA	170	¬	ALT 170	注 6	绿	淡绿	高亮度
AB	171	½	ALT 171	注 6	绿	兰	高亮度
AC	172	¼	ALT 172	注 6	绿	浅红	高亮度
AD	173	¡	ALT 173	注 6	绿	浅品红	高亮度
AE	174	«	ALT 174	注 6	绿	黄	高亮度

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
AF	175	»	ALT 175	注 6	绿	白	高亮度
B0	176		ALT 176	注 6	深兰	黑	一般
B1	177		ALT 177	注 6	深兰	兰	下划线
B2	178		ALT 178	注 6	深兰	绿	一般
B3	179		ALT 179	注 6	深兰	品红	一般
B4	180		ALT 180	注 6	深兰	红	一般
B5	181		ALT 181	注 6	深兰	品红	一般
B6	182		ALT 182	注 6	深兰	棕	一般
B7	183		ALT 183	注 6	青	浅灰	一般
B8	184		ALT 184	注 6	青	深灰	高亮度
B9	185		ALT 185	注 6	青	浅兰	高亮度 下划线
BA	186		ALT 186	注 6	青	浅绿	高亮度
BB	187		ALT 187	注 6	青	浅青	高亮度
BC	188		ALT 188	注 6	青	浅红	高亮度
BD	189		ALT 189	注 6	青	浅品红	高亮度
BE	190		ALT 190	注 6	青	黄	高亮度
BF	191		ALT 191	注 6	青	白	高亮度
C0	192		ALT 192	注 6	红	黑	一般
C1	193		ALT 193	注 6	红	兰	下划线
C2	194		ALT 194	注 6	红	绿	一般
C3	195		ALT 195	注 6	红	青	一般
C4	196		ALT 196	注 6	红	红	一般
C5	197		ALT 197	注 6	红	品红	一般

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
C6	198		ALT 198	注 6	红	棕	一般
C7	199		ALT 199	注 6	红	浅灰	一般
C8	200		ALT 200	注 6	红	深灰	高亮度
C9	201		ALT 201	注 6	红	浅兰	高亮度 下划线
CA	202		ALT 202	注 6	红	浅绿	高亮度
CB	203		ALT 203	注 6	红	浅青	高亮度
CC	204		ALT 204	注 6	红	浅红	高亮度
CD	205		ALT 205	注 6	红	浅品红	高亮度
CE	206		ALT 206	注 6	红	黄	高亮度
CF	207		ALT 207	注 6	红	白	高亮度
D0	208		ALT 208	注 6	品红	黑	一般
D1	209		ALT 209	注 6	品红	兰	下划线
D2	210		ALT 210	注 6	品红	绿	一般
D3	211		ALT 211	注 6	品红	青	一般
D4	212		ALT 212	注 6	品红	红	一般
D5	213		ALT 213	注 6	品红	品红	一般
D6	214		ALT 214	注 6	品红	棕	一般
D7	215		ALT 215	注 6	品红	浅灰	一般
D8	216		ALT 216	注 6	品红	深灰	高亮度
D9	217		ALT 217	注 6	品红	浅兰	高亮度 下划线
DA	218		ALT 218	注 6	品红	浅绿	高亮度
DB	219		ALT 219	注 6	品红	浅青	高亮度

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
DC	220		ALT 220	注 6	品红	浅红	高亮度
DD	221		ALT 221	注 6	品红	浅品红	高亮度
DE	222		ADT 222	注 6	品红	黄	高亮度
DF	223		ALT 223	注 6	品红	白	高亮度
E0	224	α	ALT 224	注 6	黄	黑	一般
E1	225	β	ALT 225	注 6	黄	兰	下划线
E2	226	γ	ALT 226	注 6	黄	绿	一般
E3	227	π	ALT 227	注 6	黄	青	一般
E4	228	Σ	ALT 228	注 6	黄	红	一般
E5	229	σ	ALT 229	注 6	黄	品红	一般
E6	230	ζ	ALT 230	注 6	黄	棕	一般
E7	231	τ	ALT 231	注 6	黄	浅灰	一般
E8	232	Φ	ALT 232	注 6	黄	深灰	高亮度
E9	233	ϕ	ALT 233	注 6	黄	浅兰	高亮度 下划线
EA	234	Ω	ALT 234	注 6	黄	浅绿	高亮度
EB	235	δ	ALT 235	注 6	黄	浅青	高亮度
EC	236	∞	ALT 236	注 6	黄	浅红	高亮度
ED	237	ϕ	ALT 237	注 6	黄	浅品红	高亮度
EE	238	$\in$	ALT 238	注 6	黄	黄	高亮度
EF	239	$\cap$	ALT 239	注 6	黄	白	高亮度
F0	240	$\equiv$	ALT 240	注 6	白	黑	视频翻转
F1	241	$\pm$	ALT 241	注 6	白	兰	下划线
F2	242	$\geqslant$	ALT 242	注 6	白	绿	一般

（续表）

值		作为字符			作为文本模式的属性		
					彩色/图形转接器		IBM 单色显示器转接器
HEX	DEC	符号	按键	模式	背景色	背景	
F3	243	≤	ALT 243	注 6	白	青	一般
F4	244	S	ALT 244	注 6	白	红	一般
F5	245		ALT 245	注 6	白	品红	一般
F6	246	÷	ALT 246	注 6	白	棕	一般
F7	247	≈	ALT 247	注 6	白	浅灰	一般
F8	248	○	ALT 248	注 6	白	深灰	视频翻转
F9	249	●	ALT 249	注 6	白	浅兰	高亮度下划线
FA	250	·	ALT 250	注 6	白	浅绿	高亮度
FB	251	√	ALT 251	注 6	白	浅青	高亮度
FC	252	n	ALT 252	注 6	白	浅红	高亮度
FD	253	2	ALT 253	注 6	白	浅品红	高亮度
FE	254	■	ALT 254	注 6	白	黄	高亮度
FF	255	BLANK	ALT 255	注 6	白	白	高亮度

注 1：星号(*)可用两种方法按入：1) 按 PRTSC/* 键；2) 移位态下按 8/* 键。

注 2：句号(·)可用两种方法输入：1) 按 ./> 键；2) 移位态或 NUMLOCK 态下按 del/. 键。

注 3：数字键 (0-9) 可用两种方法输入：1) 按键盘上沿的数字键；2) 移位态或 NUM LOCK 态下按键盘右边的数字键。

注 4：大写字母 (A-Z) 可用两种方法输入：1) 移位态下按字母键；2) CAPS LOCK 态下按字母键。

注 5：小写字母 (a-z) 可用两种方法输入：1) “一般” 态下按字母键；2) CAPS LOCK、移位态混合模式下按字母键。

注 6：按 ALT 键后可用 71-73、75-77、79-82 号键输三个数字，从而送入 000 至 255 号字符码。

字符集 (00-7F)

十进制值 → / 十六进制值 ↓	0	16	32	48	64	80	96	112
	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	●	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	◆	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	'	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	x
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	|
13	D	♪	↔	-	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	—	o	△

字符集 (80-FF)

十进制值 ↓	十六进制值 →	128	144	160	176	192	208	224	240
		8	9	A	B	D	C	E	F
0	0	Ç	E	á				∝	≡
1	1	ü	Æ	í				β	±
2	2	é	FE	ó				γ	⩾
3	3	â	ô	ú				π	⩽
4	4	ä	ö	ñ				Σ	⌠
5	5	à	ò	Ñ				σ	⌡
6	6	å	û	a				μ	÷
7	7	ç	ù	o				τ	≈
8	8	ê	ÿ	¿				Φ	○
9	9	ë	Ö	⌐				Θ	●
10	A	è	Ü	¬				Ω	∙
11	B	ï	¢	½				δ	√
12	C	î	£	¼				∞	η
13	D	ì	¥	¡				∅	2
14	E	Ä	Pts	«				∈	▌
15	F	Å	f	»				∩	BLANK