TP801

Z80单板计算机使用手册

北京工业大学电子厂　　广州无线电专用设备厂

香港京業公司

前　言

本《使用手册》共分四章。第一章概述 TP801—Z80 单板计算机的主要技术特性、功能、操作步骤及使用注意事项，并对《Z80袖珍设计手册》作了简单说明。第二章扼要介绍了监控程序 TPBUG；结合实际操作举例较为详细地叙述了各个按键的操作使用。第三章是 TP801 的结构和原理说明，主要介绍了 CPU、存储器、I/O 接口三个部件。第四章为一些实用程序举例，以供读者学习一些程序设计的方法和技巧。

为便于读者从软件和硬件两方面了解 TP801，本《用户手册》的附录编入了 TP801 的监控程序 TPBUG、线路原理图、安装位置图和零件表。此外，还编入 TP801 所使用的全部集成电路的引脚图，以便用户维修和测试本机时查对。

微处理器应用软件的编制往往使用手编程序。为方便广大 TP801 用户，我们同时提供美国 MOSTEK 公司编的《Z80 袖珍设计手册》（北京工业大学微处理机应用及自动化研究室译）。其中编入了 Z80—CPU 的指令系统和 Z80 其它芯片的编程要点。

考虑到 TP801 用户中初学微处理器的读者在使用本手册时可能会遇到一些困难，我们还随产品一起提供北京工业大学微处理机应用及自动化研究室编写的《微处理器实用教材——基础知识和 Z80 器件的使用》。

本《使用手册》与上述《微处理器实用教材》两本书，将作为对 TP801 用户提供技术培训时的教材。当然，这两本书也可供各类专业的技术人员和大专院校学生学习使用微型计算机入门参考。讲授这两本书约需 40～60 学时，实验课程约 20～30学时，无需其它先修课程。学完本课程后，学员将能初步掌握 Z80 系列器件的使用，TP801 单板机的使用，并学会手编微型计算机程序的初步技巧。有条件的话，也可以通过讲课或自学来读通监控程序 TPBUG(2K字节)。

本《使用手册》第一、三章由北京工业大学微处理机应用及自动化研究室徐家栋编写，第四章由该室吴定荣编写，第二章由吴定荣、徐家栋合写。该室侯伯文参加了本《使用手册》的修改定稿工作，并和徐家栋一起对监控程序的文本进行了修订和注释。

恳切希望广大 TP801 的用户及本书读者对书中的错误和不当之处提出指正。

北京工业大学电子厂
广州无线电专用设备厂
香　港　京　業　公　司
1981 年 4 月 30 日

目　　录

第一章 概 述

1.1 引 言

近年来，微处理器/微型计算机的发展十分迅速。它的应用已深入到工业、农业、国防、科研、教育、管理以及日常生活（如家用电器、玩具）等各个领域。在自动控制和仪器仪表方面的应用尤为突出。随着大规模集成电路的发展，微处理器/微型计算机必将对计算机工业和计算机应用产生深远的影响。

为了推广微型计算机的应用，北京工业大学和香港京業公司合作，研制了一种新型的价格便宜而性能优良的 TP801—Z80 单板计算机。该机是使用 Z80 系列器件，做在一块印刷电路板上的完整的微型计算机。它结构简单，布局合理，功能齐全，用途广泛。

TP801—Z80 单板计算机作为"智能"部件，可用于生产过程控制、各种仪器和仪表或机械的单机控制、数据处理等等。它既可独立应用在小型自动控制系统中，又可用于分布系统中的前沿控制。该机尤其适合于初学者学习微型计算机的硬件、指令系统、编写程序的方法和技巧。因此，对大专院校学生和各行各业需要应用微型计算机的科技工作者来说，TP801—Z80 单板计算机也是一种经济实用的实验教学设备。

TP801—Z80 单板计算机还具有简易的开发功能，如在用户程序内可设置多至五个断点，可单步执行存于 RAM 或 PROM 中的程序，可对 2716/2758 EPROM 进行编程等。因而，可以将它用作调试样机，也可以直接将它用于微型机化（μp—based）产品中。

TP801 作为 TP80 系列的基础部件，在其两个 S100 总线插座上附加 TP 802 扩展板后，内存容量将增大 16K，可使用 BASIC 语言，并增配字符键盘、显示器（或以黑白电视机代用）、微型打印机等外围设备，这样就构成一种经济实用、功能更为完善的简易微型计算机系统。

1.2 主要技术特性

1．中央处理单元为 Z80—CPU。

2．时钟（ϕ）频率为 1.9968MHz，便于使用 8080A 的接口电路。晶体振荡器的频率为 3.9936MHz。

3．RAM 为 4K 字节的 2114 静态读写存储器，也可只使用 2K 字节。

4．ROM 为 2K 字节，编入监控程序 TPBUG。

5．PROM 插座两个。可插入 4K 字节的 PROM或 EPROM。

6．Z80—PIO 并行 I/O接口芯片一个，它有两个 8 位的可编程的 I/O口，全供用户使用。

7．Z80—CTC 计数器/定时器芯片一个，它有四个通道：通道 0 供用户使用，其余由 TPBUG 使用。

8．按键共 28 个，16 个为十六进制数字键；12 个为命令键，其中包括：

MEM EXAM（存储器检查）键

PORT EXAM（口检查）键

REG EXAM（寄存器检查）键

REG′ EXAM（辅助寄存器检查）键

EXEC（连续执行程序）键

SINGLE STEP（单步执行程序）键

MON（监控）键

NEXT 键

BREAK POINT（设置断点）键

PROM PROG（EPROM写入）键

CASS LOAD（磁带输入）键

CASS DUMP（信息转储磁带）键

9．显示器有六位 LED 数字显示，通常左四位显示地址，右两位显示数据。

10．配有音频盒式磁带机接口。

11．布线区为 2.5×7 英寸。

12．S—100 总线插孔二组。

13．电源为 +5V±5%，1A。本机可以增配 7805 稳压器件，这时输入须接≥+7V 直流电源。若要对 EPROM 进行写入，尚须接入 +25V±1V，30mA 的电源。

1.3 功能简介

TP801—Z80 单板计算机系使用 Z80 系列器件。Z80—CPU 的指令系统有指令 158 条，其中包含了 8080A 的全部指令。此外，还增加了数据块传送与查找、位操作（置 1、置 0、测试）、相对转移、变址寻址、16 位的算术运算等。Z80—CPU 的内部寄存器也较 8080A多，增设了 IX、IY，以及一套辅助寄存器。这些特点大大增强了Z80—CPU 的处理功能，并非常便于程序设计，Z80 的软件可以与 8080A 的软件相容。

本机使用八个廉价的静态读写存储器 2114，容量为 4K 字节。板上配有 2K 字节的 ROM，存放监控程序 TPBUG，便于用户利用键盘或盒式磁带输入程序和数据，以及提供其他的软件功能。板上设有一个完整的 EPROM编程器，允许用户将信息写入$2716/275_8$ EPROM中。编程所需的电压为直流 +25V±1V，最大电流为 30mA。

本机配有音频盒式磁带机接口，用转录线将盒式磁带录音机与机上插孔相连，可以方便地存取数据和程序。单板机上的红色发光二极管能指示磁带上有无信息。利用这只二极管的指示，可在一盘磁带上录制几个文件。RAM和磁带之间的存取由命令键控制。板上配有六位数码显示，供显示地址和数据。通常左四位显示地址，右二位显示数据（对于 PC，SP，IX，IY 等 16 位寄存器而言，右四位显示数据）。板上还配有 16 个十六

进制的数字键和12个命命键，它们的功能在第二章里介绍。

本机配有三个按钮和开关：按钮S1用来对整机提供RESET（复位）信号。开关S2有两个位置可供选择，如果置向MON RST位置，则在复位后，显示器上出现“-”，并扫描键盘的输入；如果指向PROM1 RST位置，则在初始化后进入PROM1中的用户程序。开关S3用来选择PROM2中的EPROM处于“写入（PGM）”或“读出(READ)”的状态。

此外，板的左上方有两组S—100总线的插座孔。Z80—CPU的地址、数据、控制总线都接到S—100总线插孔，以便在插座上扩充存储器板及I/O接口板。板的左方有2.5×7英寸的布线区，约可安装25片IC（集成电路）。连接到布线区的信号有：

Z80—CPU的数据、地址和控制总线。

Z80—PIO两个口（Port）的数据线和联络线。

Z80—CTC通道0的输入和输出。

1.4 操作步骤及注意事项

一、请按以下规定连线：

1．直流稳压电源、盒式录音机均接至交流220V。

2．用转录线将单板机上的AUX插孔与录音机上的MIC插孔相连；

3．用转录线将单板机上的EAR插孔与录音机上的MONITOR OUT或EARPHONE插孔相连；

4．将单板机上的电源线接入直流稳压电源的+5V输出。

二、设定单板机上的开关位置：

1．通常将开关S2设定在MON RST位置；

2．通常将开关S3设定在READ位置。

三、开机：

1．接通直流稳压电源上的+5V开关；

2．按下RESET按钮S1，计算机进入监控程序TPBUG，显示器上显示“—”，表明机器可接收命令。

四、输入用户程序，并进行相应的调试和操作。请参阅第二、四章。

五、如果进行第三步时，不出现“—”，则须立即关闭电源，仔细检查导线连接有否出错，器件与插座的接触是否可靠，各开关位置的设定是否正确，直流稳压电源的输出电压是否满足+5V±5%。

六、切忌用手直接触摸集成电路，切忌对本机直接使用电源电压为交流220V的电烙铁，以防损坏芯片。必须使用电烙铁时，应将电源断开，利用烙铁的余热进行焊接。

1.5 《Z80袖珍設計手册》使用說明

TP801—Z80单板计算机提供用户一本《Z80袖珍设计手册》。这本小册子是扼要叙述Z80—CPU指令系统及其他芯片的编程要点。

本册第一页是Z80—CPU 内可供用户使用的寄存器。其中有一套与8080A兼容的主寄存器，另有一套相同的辅助寄存器。此外还有一套专用寄存器。它们的使用请参考《微处理器实用教材—基础知识和Z80器件的使用》。第2、3页是摘要表示指令是如何影响各个标志位的。第4、5页是8位传送指令组，黑体字的指令操作码与8080A相兼容，其余的是Z80所独有的新指令。新指令是变址的传送指令。第5页是对这些指令进行更为详细的说明。第6、7页是16位传送指令组，BC和DE的内容可以直接由存储单元输入，而不必通过HL寄存器。第8、9页是交换指令组和数据块传送和查找指令组。第10、11页是8位算术和逻辑指令组，其中大多数与8080A相兼容，新增加的指令也是一些变址操作指令。第12、13页是其他指令组，其中有求补码指令和中断方式选择指令。中断方式0是仿照8080A的，而方式2为Z80系列接口芯片所使用。

第14、15页是16位算术指令组。除增加了变址操作外，还增加了两种只有在小型计算机才具备的ADC和SBC指令。第16、17页提供了远较8080A更丰富的移位和循环移位操作的指令组。这些操作还能在以HL，IX和IY作为指针的存储单元上进行，就象6800那样。此外还能将存储单元中的BCD数字与累加器A进行移位操作。第18、19页是位操作指令。每一位都可被置位、复位和测试。第20、21页是转移指令组，其中有二字节的相对转移，而8080A只有三字节的转移指令。第22、23页是子程序调用和返回指令组，其中只增加了两条新指令，RETI和RETN，这两条是从中断和不可屏蔽中断返回的指令。第24、25页是输入/输出指令，新增加的指令为以C寄存器作为I/O的指针，数据传送的对象可以是CPU内8位寄存器。此外还可以进行数据块的输入/输出。第26页是Z80—CPU的中断结构。第27、28页是Z80—PIO和CTC的编程要点。第29、30、31页是Z80—SIO的编程要点。最后是ASCII字符表。

第二章　键盘操作說明

在这一章，逐个介绍本机中各个按键的使用，并扼要介绍监控程序TPBUG。更为详细的说明请参阅第四章及附录三。

2.1 监控程序TPBUG简介

TPBUG固化在可擦的可编程序只读存储器（EPROM）中，在存储空间中它占用的地址为0000—07FFH，此外，TPBUG还使用了112个RAM单元。有了TPBUG，就可以通过键盘向计算机输入机器语言水平的程序和数据。

TPBUG可以划分为下列四个主要程序段：（1）初始化、显示和键盘分析程序；（2）键盘动作程序；（3）实用子程序；（4）表格。

一、初始化、显示和键盘分析程序

1．初始化（0000—00F3H）

它设定栈指针SP＝2FC0H，并将其保存在2FE2——2FE3H单元，然后设定TPBUG的栈指针SP＝2FA8H。清除键盘状态标志。在最左端显示器对应的显示单元DISMEM中填入“—”的代码11H，其余填入空格的代码10H。

2．显示DISUP（00F4—0122H）

它将六个用于显示的显示缓冲单元(DISMEM—DSMEM5)的内容依次轮流取出，送往LED显示器，每位数字保持显示一毫秒。

3．键盘分析（0123—01CAH）

它等待并搜索键盘的输入。若没有按键压下，则又回到显示DISUP程序。若有按键压下，则对所按下的键进行分析。如果是数字键，则送入相应的显示单元；如果是命令键，则进入与该键功能相对应的键盘动作程序。

二、键盘动作程序（0230—0633H）

本机有十二个命令键，相应的有十二个按键动作程序。这些动作程序的内容将在后面关于键盘操作的叙述中作简要介绍。

三、实用子程序（0634—07A5H）

这些子程序都是供TPBUG中上述程序段使用的，也可以由用户调用。其中也包括一些中断服务程序。

下面介绍几种常用的子程序。

1.UIX3（起始地址为0634H）

它使 IX 寄存器增 3，B寄存器减 1。该子程序只影响 IX，B，F三个寄存器。

2．UFOR1（起始地址为 063CH）

它将累加器A中的高 4 位作为一个十六进制数字写入（IX）单元，将低 4 位写入（IX＋1）单元。该程序影响的寄存器有：A，B，F。

3．D20MS（起始地址为064FH）

它仅起延时 20ms 的作用，即调用这个子程序后，经过 20ms 后才返回。该程序影响的寄存器有H，L，F。

4．UABIN（起始地址为06B3H）

它将累加器A中的一个 ASCII 字符转换为二进制数，再送回累加器A。该程序使用的寄存器有A和F。

5．UBASC（起始地址为06BBH）

它将累加器A中的低 4 位作为一个十六进制数字，转换为 ASCII 字符，再送回累加器A。该程序使用的寄存器有A和F。

四、表格（07A6—07FFH）

2.2 复位(RESET)按钮

本机中有两种实现复位的方法：一种是上电复位，即一合上电源即自动地提供复位信号；另一种是压下印刷线路板右方按钮开关S1。

当开关S3处于 READ 位置，S2处于MON RST位置时，复位按钮的作用如下：

1．使Z80—CPU 处于初始状态。这些初始状态包括：

1）中断允许触发器处于禁止状态；

2）置寄存器I，R的内容为00H；

3）置程序计数器 PC 的内容为 0000H；

4）置中断方式为 IM0。

在复位信号有效期间，地址总线和数据总线处于浮动状态，所有的输出信号均为无效。

2．使本机从 0000H 开始执行监控程序的初始化部分，详细内容请参阅上节及附录三。此时，显示器的最左端应显示TPBUG的标志符号“—”。

3．如果在用户程序中用 BREAK POINT 键设置了断点，则压下复位按钮后，所设置的断点被清除。

4．如果开关 S2 处于 PROM1 RST 位置，则 TPBUG 执行完初始化程序后，自动转移到 PROM1 中起始地址为 0800H 的用户程序。这个功能使用户能方便地转入 PROM1 中的用户程序。而不必从键盘输入命令。

2.3 16个十六进制数字键

这些键用来向计算机输入十六进制数字。这些数字可以是存储单元的地址、I/O口

地址、寄存器标号、指令码，也可以是一些数据。每压下一个数字键，此数字即存入相应的显示缓冲单元。最先输入的数字送（DISMEM = 2FF7H）单元，依次为（DSMEM1 = 2FF8H），……（DSMEM5 = 2FFCH），（DSMEM6 = 2FFDH），（DSMEM7 = 2FFEH）。并将前六个单元的内容显示于六位LED显示器上，（DISMEM）单元在最左边的 LED上显示出来。具体操作如表 2.1 所示。

数字键与命令键的联合使用见以下各节。

表 2.1　　数 字 键 使 用 举 例

按　　鍵	显　　示	說　　明
MON	—	压下 MON 鍵或按复位按鈕，显示“—”
2	2	压下第一个数字鍵 2
0	2 0	再压下数字鍵 0
3　4	2 0 3 4	先后压下数字鍵 3，4
B　6	2 0 3 4　　B 6	先后压下数字鍵 B，6
7	2 0 3 4　　B 6	压下第七个数字鍵 7 沒有任何反应
8	—	压下第八个数字鍵，TPBUG进行处理，由于沒有命令鍵輸入，連續輸入 8 个数字毫无意义，所以控制进入TPBUG的初始化部分

2.4 MON(MONitor 监控)鍵

MON 鍵有两种功能。一是中止现行程序的执行，当计算机执行 RAM中的用户程序时（也即用户已压下过 EXEC 鍵），压下MON鍵，通过 U15 单稳电路向 CTC 通道 2 的 C/T2 输入一个脉冲。由于在EXEC鍵的鍵盘动作程序（起始地址为 0230H）中，将 CTC 通道 2 设定为计数器工作方式，其初始常数为01H，因而，C/T2的脉冲导致ZC2 上产生一个脉冲。这个脉冲通过U33和U34送往CPU的$\overline{\text{NMI}}$输入端，从而产生不可屏蔽中断，程序转入0066H及01CBH的中断服务程序。

这种功能在分析程序故障时尤其有用。例如，如果由于程序设计上的错误，或者由

于执行了一条暂停（HALT）指令，或者由于陷入无休止循环，那么用户可以使用MON键，使控制返回到监控程序TPBUG而能保护CPU寄存器内容。这样，用户就能知道压下MON键时程序执行到何处。

MON键的另一种功能是，中止或者退出当前的命令状态或输入数据，使控制返回到TPBUG的初始化部分，等待下一个按键输入，以便输入新的命令或数字。

通常要进行一种新的操作方式之前，均需压下MON键，使显示器上出现“—”后，才能压下新的命令键或数字键。

MON键与复位按钮的作用相似，都是使控制返回到TPBUG。其不同点是：MON键能保护CPU寄存器的状态以及TPBUG所使用的一些RAM专用单元（主要是键盘状态标志）。表2.2是说明MON键使用的例子。

表 2.2　　MON 键 使 用 举 例

按　　键	显　　示		說　　明
MON	—		准备接受命令
2 0 0 0	2 0 0 0		輸入四个数字，作为存儲单元地址
MEM EXEC	2 0 0 0	× ×	检查2000单元，××为讀出值
MON	—		退出以上工作方式
8 4	8 4		輸入口地址
PORT EXAM	8 4	× ×	检查口 84，××为讀出值

2.5 MEM EXAM (MEMory EXAMine 存儲器检查)键

MEM EXAM键用来检查或更改RAM单元的内容，也就是通过键盘命令对RAM单元进行读、写操作。此外，也可对ROM或PROM进行读操作。

当显示器最左端出现“—”标志后，通过键盘输入表示地址的四个十六进制数字。先送地址的高位数字，后送低位数字，每送入一个数字，就立即显示在显示器上，最高位显示在最左端。然后压下MEM EXAM键，则在最右边两个显示器上出现该存储单元的内容。如果还没有送完四个数字的地址就压下MEM EXAM键，那么压下此键不起任何作用，右边

两个显示器上不会出现数字，也不影响以前输入的数字。在这种情况下，如果继续输入数字，在送完地址的四个数字之后再一次压下MEM EXAM键，那么在显示器上仍能出现该单元的内容。

表 2.3　　MEM EXAM 键 使 用 举 例

按　　鍵	显　　示	說　　明
2　1	2　1	压下两个数字鍵
MEM　EXAM	2　1	此鍵不起任何作用
3　4	2　1　3　4	繼續輸入数字 3、4
MEM　EXAM	2　1　3　4　　5　6	显示該单元內容 5、6

其次，介绍如何更改存储单元的内容。（参阅表2.4）

在完成上述检查存储单元内容的操作之后，显示器上有六个数字，左边四个数字为地址，右边两个数字为存储单元的内容。如果再输入两个数字，这两个数字即被写入该存储单元，并被显示出来，原先的存储单元内容随即消失。值得注意的是，只有送完两个数字后，显示才会更新。如果只输入一个数字，则这个数字保存在(DSMEM6 = 2FFDH)单元中，不起任何作用。只有当第二个数字输入后，才会将这两个数字写入该存储单元中，然后从此存储单元中送至DSMEM4和DSMEM5单元中，并被显示出来。如果由于偶然的疏忽，用户企图更改ROM或空单元的内容，虽然在DSMEM6和DSMEM7单元会保存新输入的数字，但是它无法通过存储单元而进入DSMEM4和DSMEM5单元，从而不会更改右边显示的内容。此时用户会意识到自己的操作错误。

如果连续压下MEM EXAM键，则其结果只是重复读出并显示出该存储单元的地址和内容。如果压下NEXT键，则显示下一个单元的地址和内容。

2.6 PORT EXAM(PORT EXAMine 口检查)鍵

PORT EXAM 键的功能与MEM EXAM键相似，它用来检查或更改口的内容。

当显示器最左端出现“—”标志后，通过键盘输入口地址的两个十六进制数字，先送地址的高位数字，后送地址的低位数字。由于最多只能访向256个I/O口，所以口地址只需要两个十六进制数字。然后压下 PORT EXAM 键，则在最右边两个显示器上将出现该口的内容。由于它的功能 与 MEM EXAM 键相似，请读者参阅上节和本节的例题，例中将扼要地介绍这些特性，从而不再作文字的叙述。

表 2.4　　用 MEM EXAM 键更改存储单元的内容

按　　键	显　　示		說　　明
MON	—		或按复位按鈕
2 0 0 0	2 0 0 0		依次輸入存儲单元地址
MEM EXAM	2 0 0 0	× ×	××为原讀出值
0 6	2 0 0 0	0 6	用0 6 更改原来內容
4	2 0 0 0	0 6	只輸入一个数字，不会更改原来內容
5	2 0 0 0	4 5	送完两个数后，更改原来內容
NEXT	2 0 0 1	× ×	显示下一单元的地址和內容
7 B	2 0 0 1	7 B	用7 B 更改原来內容
MON	—		准备检查新单元 0000H
0 0 0 6	0 0 0 6		地址号錯了，应是 0000
MON	—		消去上述数字
0 0 0 0	0 0 0 0		輸入正确的地址号
MEM EXAM	0 0 0 0	3 1	讀出 0000H单元內容
4 5	0 0 0 0	3 1	再輸入两个数字，但无效，因为不能更改 ROM 的內容

值得注意的是，如果连续压下 PORT EXAM 键，则将连续显示该口的内容。这种特性对于CTC芯片是很有用的，通过这种操作可以了解 CTC 芯片中减 1 计数器的内容的变化情况。

压下 MON 键，可以使之退出 PORT EXAM 工作方式。

本机中共用11个I/O口，如表2.5所示。顺便指出，有些口只能读出，有些口只能写入，有些口（如PIO）包含若干个寄存器，读出和写入的操作与操作方式有关，使用时应予注意。

表 2.5　　TP801 使用的 I/O 口

口　地　址	口	說　　明
80 H	PIO A 口数据寄存器	口內有数据輸入寄存器和数据輸出寄存器，讀写过程与操作方式有关
81 H	PIO B 口数据寄存器	同　　上
82 H	PIO A 口控制寄存器	只能写入，不能讀出
83 H	PIO B 口控制寄存器	同　　上
84 H	CTC 通道 0	可讀写，但讀出的是減 1 計数器內容
85 H	CTC 通道 1	同　　上
86 H	CTC 通道 2	同　　上
87 H	CTC 通道 3	同　　上
88—8BH	七段选择鎖存器 SEGLH	只能写入，不能讀出
8C—8FH	数字选择鎖存器 DIGLH	同　　上
90—93H	鍵盘选择三态輸入緩冲器 KBSEL	只能讀出，不能写入

2.7　REG EXAM (REGister EXAMine 寄存器检查)鍵

REG EXAM 键用来检查或更改下列 CPU 寄存器的内容：A，B，C，D，E，F，H，L，I，IFF，PC，IX 和 IY。SP 寄存器的内容只能被检查，不能被更改。此键的使用与 MEM EXAM 相似，唯一的不同是，它不能和 NEXT 键配合使用，因为 CPU 寄存器没有地址。

先压下代表寄存器号的数字键，在显示器的右边出现该数字。然后压下REG EXAM 键，显示器右边的两个或四个数字即为寄存器中的内容。IFF 是中断允许触发器的状态，00表示禁止中断，04表示允许中断。

表 2.6　　PORT EXAM 键使用举例

按　　鍵	显　　示		說　　明
MON	—		准备接受命令
8	8		輸入口地址
PORT EXAM	8		未輸完口地址即压下 PORT EXAM 鍵，不起任何作用
0	8　0		輸完口地址 80
PORT EXAM	8　0	×　×	查口的內容，××为讀出值
NEXT	8　1	×　×	查下一个口的內容，××为讀出值
MON	—		准备接受命令，检查口90
8　8	8　8		輸入錯誤的口地址88，应为 90
MON	—		准备接受新的命令
9　0	9　0		輸入正确的口地址 90
PORT EXAM	9　0	7　F	检查口 90 的內容

如果要改变所显示寄存器的內容，只需再压下两个数字鍵（若是IX、IY或PC，则需要输入四个数字）。

用户可以压下MON鍵使之退出REG EXAM工作方式，或检查另一个寄存器。

在TPBUG的大多数按鍵工作方式下，CPU 寄存器的內容都保存在RAM的“用户寄存器存放区”。每当压下EXEC鍵（连续执行程序）或SINGLE STEP鍵（单步执行程序）而进入这两个按鍵的鍵盘动作程序时，此动作程序首先将“用户寄存器存放区”的內容送入各CPU寄存器而后进入用户程序。每当执行程序时遇到了断点或者单步执行程序结束时，或者用 MON 鍵中止用户程序的执行时，也就是说，每当退出用户程序而进入TPBUG的控制时，CPU中各个寄存器的內容又推入“用户寄存器存放区”，以便用户进行更改和读出。

2.8　REG′ EXAM(alternate REGister EXAMine 輔助寄存器检查)鍵

REG′ EXAM 鍵用来检查或更改下列CPU 辅助寄存器的內容：A′，B′，C′，D′，

表 2.7　　REG EXAM 键使用举例

按　　键	显　　示		說　　明
MON	—		准备接受命令
A　REG EXAM	A	× ×	检查累加器 A 的內容，××为讀出值
8	A	× ×	用8 4写入 A，輸入第一个数字 8 沒有反应
4	A	8 4	两个数字 8 4 都輸入后，A 內容被更改
MON	—		准备接受命令
2　REG EXAM	2　2 F	C 0	检查 SP 的內容，为2 FCOH
2　F　C　2	—		企图更改 SP 的內容为 2FC2，沒有成功
1　REG EXAM	1　2 0	0 0	检查 PC 的內容，为 2000
2	1　2 0	2 0	2 进入显示
3	1　2 0	2 3	3 进入显示
4	1　2 0	2 3	沒有响应
5	1　2 3	4 5	用 2 3 4 5 更改 PC 的內容
NEXT	—		NEXT 鍵不起作用

E′，F′，H′和L′。用“′”来表示辅助寄存器。该鍵的操作和使用与 REG EXAM相同，不再赘述。

2.9 BREAK POINT(設置断点)鍵

该键用来在用户程序中设置一至五个断点。在调试用户程序时，利用断点可以在用户程序的某处暂停执行，以便用户检查上一段程序设计中有否发生差错，进而修改程序

表 2.8　　设 置 断 点 的 操 作 举 例

按　　鍵	显　　示	說　　明
MON	—	准备接受命令
2 0 1 5	2 0 1 5	先輸入第一个断点地址
BREAK POINT	2 0 1 5	断点被接受，(BFLG)=1 2015被送往(2FE4) (2FE5)单元
MON	—	
2 0 2 4	2 0 2 4	輸入第二个断点地址
BREAK POINT	2 0 2 4	第二个断点被接受,(BFLG)=2,2024 被送往(2FE7)(2FE8) 单元
MON	—	
2 0 0 0	2 0 0 0	輸入要执行的程序 的 起始地址
EXEC	2 0 1 5 × ×	从 2000H 单元开始执行程序，暫停于第一个断点处显示断点地址(即PC) 及累加器A的內容
EXEC	2 0 2 4 × ×	繼續从 2015H 单元执行程序暫停于第二个断点处，显示該断点地址 (PC) 及累加器 A 的內容，上述三步可反复进行
MON	—	控制轉回 TPBUG
2	2	輸入数字 2
BREAK POINT	2	由于沒有送完 四个数 就压下BREAK POINT鍵，从而使 (BFLG) = 0，断点表 BPTAB 中的信息不起作用，显示保持原状

或者修改寄存器和I/O口中的内容。

设置断点的操作如表2.8所示。每当输入一个断点后，断点标志（BFLG = 2FF4H单元）的内容增1，断点地址被送往起始地址为2FE4H的“断点表”BPTAB。该表共有15个单元，每个断点占用三个存储单元，其中两个存放断点地址，另一个存放用户程序断点处指令操作码的第一个字节。在输入断点地址后按下 BREAK POINT键，显示的地址应暗后复明，表示断点已被接收。如果输入的断点数多于五个，则多设的断点输入将是无效，此时显示器上出现标志“—”。最初输入的五个断点仍保持不变。

当执行设有断点的用户程序时，先执行第一条指令，然后进入不可屏蔽的中断服务程序，装配断点，即将RST8（CFH）单字节指令代替各个断点的指令操作码的第一个字节，而此指令操作码的第一字节均被送往BPTAB。然后继续执行程序，直到断点处执行RST8指令。接着进入起始地址为0008H的服务程序，它暂停程序的执行，CPU寄存器的内容均保存在“用户寄存器存放区”，并将断点表BPTAB中所有的断点指令操作码的第一个字节（不管是否已经执行）送回用户程序，取代RST8指令，以便用户进行检查或更改，此时显示PC和A的内容。

如果按下 EXEC 键，可继续执行用户程序；如果程序中还有其余断点，则其过程与上述相同。

在每个断点处暂停用户程序时，用户可用MON等键对已执行的程序和执行结果（寄存器和I/O口）进行检查或修改。

如果用户要清除已设置的断点，可以使用下列三种方法：

1．按下复位按钮；

2．按下SINGLE STEP键；

3．在输入断点地址的四个数字之前，也就是说在输入零到三个数字后，紧接着压下BREAK POINT键。

上述三种方法都是将断点标志（BFLG）清零，从而使断点表中的数据无效。

通常MON键对设置断点没有任何影响，但是下列情况是例外。如果程序遇到 HALT 指令而停止，此时压下MON键，使控制转回 TPBUG，但是用户程序中的RST8指令并没有被取代，而仍然留在用户程序中。在调试过程中，用户应该注意这种情况的发生。

2.10 SINGLE STEP(单步执行程序)键

此键可以用来每次执行程序中的一条指令。执行完后，显示PC（即下一条指令的地址）和A的内容。此时，用户可以使用MON键以及其他的按键来检查或修改程序、I/O口或CPU寄存器。

用户可以反复地压下 SINGLE STEP 键，使程序一步一步地执行，显示器上可以看到下一次要执行的指令的地址。对于条件转移（conditional jump）和条件转子（conditional call）指令，这种特性是很有用的。用户可以了解程序是否发生转移以及跳转到什么地址，也就是说，测试的“条件”是否满足。在一步一步地执行程序时，用户可以更改PC的内容，强使程序转移到一个新的地址。

不仅RAM中的程序可以单步地执行，而且ROM，PROM1，EPROM中的程序也可以单步地执行。值得注意的是，单步执行与CTC通道2有关的指令，很易产生差错，用户应尽量避免这种情况。

2.11 EXEC (EXECute 連續执行程序)键

这个键用来连续执行RAM，ROM，或EPROM的程序。它有两种使用方式：

1．先输入要执行的程序起始地址的四个数字，然后压下EXEC键，即从此地址开始执行程序。如果要从程序的起始部分开始执行，通常使用这种操作方式。见表2.8。

2．仅仅压下EXEC键。此时从保存在“用户寄存器存放区”中的PC的现行地址开始执行程序。如果在连续执行程序时遇到断点，或是单步执行程序而暂停程序的执行，而此后要求连续执行程序，则只需压下EXEC键即可。这种操作方式请参阅表2.8。

2.12 CASS DUMP(CASSette DUMP 信息轉儲)键

此键用来将RAM中的信息转储到盒式录音机的磁带中。在转储过程中使用美国“堪萨斯（Kansas）城标准”。传送信息的速率为300波特（Baud），即以八个2400赫音频脉冲信号表示“1”，以四个1200赫音频脉冲信号表示“0”。在一个文件的转储过程中，开头有40秒全“1”的导引信号，末尾有5秒全“1”的结尾信号。

转储的操作过程如下：

1．用转录线将本机的J2(AUX)端与录音机的“AUXIARY”或“MIC”输入端相连。

2．将磁带装入录音机。

3．利用MEM EXAM键，将内存中需要转储的信息的起始地址置入2FCO和2FC1H单元（高字节置入2FCOH，低字节置入2FC1H）。

4．利用MEM EXAM键，将内存中需要转储的信息的最终地址置入2FC2和2FC3H单元（高字节置入2FC2H，低字节置入2FC3H）。

5．压下MON键，使显示器出现“—”。将录音机置成录音方式，然后压下CASS DUMP键。“—”将消失。

6．不需要进行音量调节，因为录音机内有AGC自动增益控制。当转储完成后，显示器上再次出现标志“—”。此时按下录音机的STOP键，停止走带。整个转储过程至少要45秒。

2.13 CASS LOAD (CASSette LOAD 磁带輸入)键

此键用来将录音机磁带中的信息输入RAM。输入的操作过程如下：

1．用转录线将录音机的“MONITOR OUT”或“EARPHONE”孔与本机的J1(EAR)孔相接。

2．将磁带走到相应位置。

3．将录音机的高音控制调到最大，低音控制调到最小，音量控制调到最小。

4．压下MON键，显示器上出现标志“—”。然后压下CASS LOAD键，“—”标志消失。

5．将录音机置成放音方式，逐渐增大音量，直至LED（发光二极管）出现亮光。然后再增大音量20%。在整个输入过程中，LED将保持发亮。

6．如果输入过程获得成功(即核实了各个记录的“检查和”Checksum*)，则TPBUG将显示标志“—”。此时可以关闭录音机。

7．如果在输入过程中发现某个记录的“检查和”有差错，则显示下一个要输入的记录的第一个字节的存放地址。这种情况表明，刚才输入的记录有差错，而在这个有差错记录以前的输入是成功的。此时应重新进行输入，并核实音量和音调的设置是否恰当。

8．在同一条磁带上可以记录好几个文件（file）。当走带进入一个文件时，LED显示器发亮；而进入文件之间的间隙时，则LED不发亮。用户可以利用这个特点来识别不同的文件。

用户也可以使用语言标志或录音机上的走带计数器对文件进行识别。

2.14 PROM PROG(ePROM PROGrammer，EPROM写入)键

在本机中，可以将RAM中起始地址为2000H的数组写入插在PROM2插座（起始地址为1900H）中的2716/2758型EPROM。对EPROM进行写入，还需要有一个+25±1V，30mA的辅助电源。此电源应接到印刷线路板上右边标有标记+25V的焊点上。

对EPROM进行写入的操作过程如下：

1．将需要写入的EPROM用紫外灯穿过器件窗口进行照射，以擦除其中原有的内容，即各单元内容均应为FF。如果是未使用过的新器件，则可以免去此步。

所用紫外线的波长为2537Å，照射强度为12000μW/cm^2，照射能量为15W—sec/cm^2。通常可使用医用的紫外线灯（15～30W），照射距离为2.5～5.0cm，照射时间为20—40分钟，即可擦除EPROM的内容。

2．在关闭电源的情况下，将EPROM插入PROM2插座。

3．合上+5V和+25V电源。

4．用MEM EXAM键或CASS LOAD键将所需写入EPROM的数据输入RAM（从2000H开始）。

5．压下MON键，LED显示器上出现标志“—”。

6．向微型机输入四个十六进制数字，表示要向EPROM写入的字节数。先送高位数字。

7．将开关S3置于PGM位置，压下PROM PROG键，显示器将熄灭，并进行写入。每个字节的写入需要约52ms。

---

* 常通要輸入文件(可以是一批数据，也可以是一个程序)由好几个記录构成，記录的格式詳見第四章，每个記录有一个“检查和”。

写入完毕后，有两种可能的结果：

一（多半）是显示器上出现标志“—”。这表示写入 EPROM 的内容与 RAM 核对后无误。

另一是在显示器上出现六个数字，分别表示 EPROM 中第一个与 RAM 内容不相符的单元的地址(左边四个数字)和内容（右边两个数字）。如果压下 NEXT 键，键盘动作程序将继续核对 EPROM 中的内容。如果没有错误，则显示器上出现“—”。如果有错误，则继续给予显示，直至用 NEXT 键检查完毕为止。在写入 EPROM 过程中出现错误，往往是由于使用了没有“擦净”的 EPROM，或者所用的 EPROM 是废品。

在写入过程完成以后，将开关 S3 置于 READ 位置。

还有一点需要提醒用户注意，在 EPROM 写入过程中要插入 52.5ms 的等待（Wait）状态。在等待状态中 Z80—CPU 暂停对动态存储器的刷新，从而使动态存储器中的内容丢失。如果用户需要扩充使用动态存储器，则应对此问题予以注意。本机中使用静态存储器，故不会产生上述问题。

在第四章中我们将介绍一种程序，它能够从 RAM 或 PROM1，ROM 的任意单元向 PROM2 的任意单元进行写入。在 RAM 顶部（2F90—2FFFH）有 112 个供 TPBUG 使用的专用单元，不能存放向 EPROM 写入的内容。

2.15 NEXT 键

NEXT 键用于下述三种操作方式：检查存储器，检查口，检查 EPROM 的下一个写入错误。它们的操作和特性已如前述，此处不再重复。

如果在其他操作方式中使用 NEXT 键，则不会产生任何反应。此时在显示器上出现“—”，控制转入 TPBUG 的初始化部分。

第三章 TP801—Z80单板计算机的结构和原理

3.1 概 述

TP801—Z80 单板计算机的原理框图如图 3.1 所示。它有三条总线：数据总线DB、地址总线 AB、控制（总）线 CB。挂到这三条总线上的器件有下列三类：

1. Z80—CPU；
2. 存储器：有 ROM，PROM1，PROM2，RAM；
3. 接口电路：有 Z80—PIO，Z80—CTC，键盘和显示，录音机的接口；

此外，尚有译码电路、时钟和复位电路等。详细的原理图请见附录一。

下面对电路的各个部分以及本机的一些功能作进一步的介绍。

3.2 时钟电路

时钟信号用来协调微型机内各部分的动作，使其有条不紊地进行操作。时钟电路示

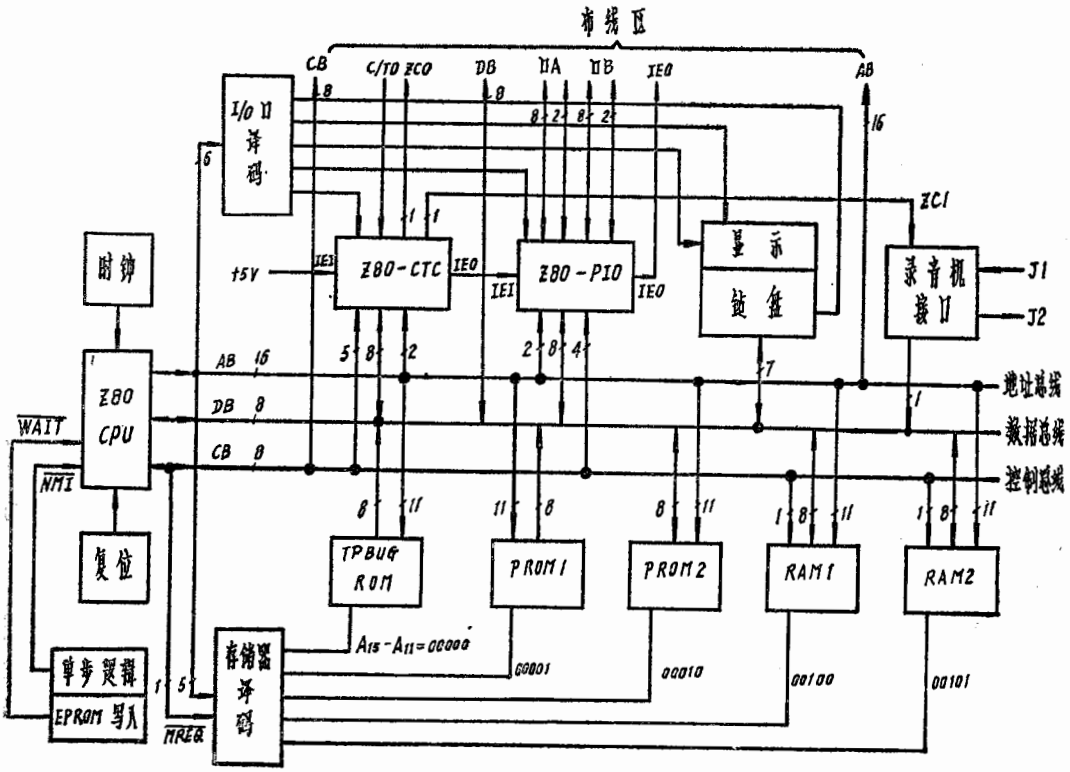


图 3.1 TP 801 单板計算机原理框图

于图3.2。晶体振荡电路的振荡频率为3.9936MHz（兆赫），这个数值恰好是堪萨斯标准音频磁带机接口所要求的1200/2400Hz（赫）和300波特（Baud）的整倍数。这个频率由D触发器U30分频而成为1.9968 MHz的CPU时钟频率。CPU时钟频率略低于2MHz，时钟周期接近于500ns，从而可以使用8080A的接口芯片以及廉价的存储器。这个D触发器的输出通过74LS04反相器接到Z80—CPU，PIO和CTC。

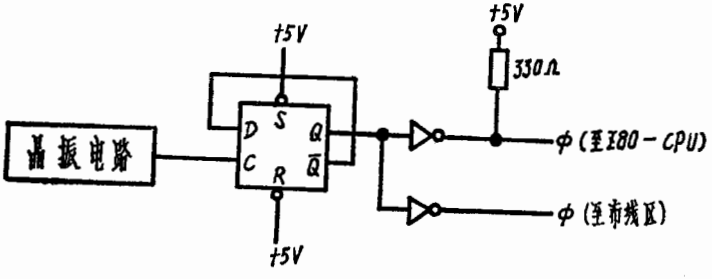


图3.2 时鐘电路

3.3 Z80-CPU

Z80—CPU有一条16位的地址总线、一条8位的双向数据总线、一条13位的控制(总)线。13个控制信号中本机使用9个，未使用$\overline{\text{HALT}}$、$\overline{\text{RFSH}}$、$\overline{\text{BUSRQ}}$、$\overline{\text{BUSAK}}$信号（$\overline{\text{BUSRQ}}$输入线接高电位，使其无效）。所有的总线均接向布线区和S—100总线插孔，以供用户附加电路时使用。关于CPU的详细介绍请参考《微处理器实用教材：基础知识和Z80器件的使用》及Z80—CPU技术手册。

3.4 存储器

一、存储空间分配

存储空间的分配及片选信号见表3.1。

TPBUG监控程序安排在最下面的2K字节（其地址号为0000—07FFH）。

0800—0FFFH为2K字节的PROM1，其中可存放用户的应用程序。

1000—17FFH为2K字节的PROM2，如果用户的应用程序超过2K字节，则可继续存放在PROM2。此外，TPBUG还可对该插座中的EPROM进行写入。

1800—1FFFH　2K字节，没被使用。

2000—27FFH　为2K字节的RAM1。其中2400—27FFH的1K字节为任选。

2800—2FFFH　为2K字节的RAM2。其中2800—2BFFH的1K字节为任选。

RAM（即RAM1和RAM2）的存储分配如表3.2所示。

3000—37FFH　为2K字节，没被使用。

3800—3FFFH　为2K字节，没被使用。

表 3.1　　TP 801 的 存 储 分 配

地　　址	器　　件	A15-A11	A10-A0	譯碼器的有效輸出
3800—3 FFFH	沒　　用	00111	可　变	$\overline{y7}=\overline{CS7}$
3000—37 FFH	沒　　用	00110	可　变	$\overline{y6}=\overline{CS6}$
2800—2 FFFH	2 K RAM 2(U 20-U 23)	00101	可　变	$\overline{y5}=\overline{CS5}=\overline{RAM2\ SEL}$
2000—27 FFH	2 K RAM 1(U 16-U 19)	00100	可　变	$\overline{y4}=\overline{CS4}=\overline{RAM1\ SEL}$
1800—1 FFFH	沒　　用	00011	可　变	$\overline{y3}=\overline{CS3}$
1000—17 FFH	2 K PROM 2 (U 9)	00010	可　变	$\overline{y2}=\overline{CS2}=\overline{PROM2\ SEL}$
0800—0 FFFH	2 K PROM 1 (U 8)	00001	可　变	$\overline{y1}=\overline{CS1}=\overline{PROM1\ SEL}$
0000—07 FFH	2 K ROM (U7)	00000	可　变	$\overline{y0}=\overline{CS0}=\overline{MON\ SEL}$

表 3.2　　RAM 的 存 储 分 配

地 址 空 間	用　　途	字 节 数	插 座 編 号
2 FFFH 2 FCOH	TPBUG 使用的 RAM 暫存区 和断点表	64	U 20—U23 RAM 2
2 FBFH 2 FA 8 H	用戶寄存器存放区 用戶栈工作区	24	
2 FA 7 H 2 F 90 H	TPBUG 栈工作区	24	
2 FF 8 H 2 C 00 H	RAM 2 的用戶工作区 (1)	912	
2 BFFH 2800 H	RAM 2的用戶工作区(2)—任选	1 K	
27 FF H 2400 H	RAM 1的用戶工作区(2)—任选	1 K	U 16—U 19 RAM 1
23 FFH 2000 H	RAM 1的用戶工作区(1)	1 K	

二、存储器译码

存储器的译码是由 U24 的 74LS138 八中取一译码器来完成的，具体电路如图3.3所示。译码器的每根输出接至 2K 存储器 的$\overline{CE}$端或$\overline{CS}$端，故本机可配16K存储器。译码器规定 A15和A14 应为 0，故16K存储器的地址空间为 0000—3FFFH，占有下方的 16K字节。

译码器的输出都接向 U27 的16个引线孔，如图Ⅰ.1所示。如果要改变各个 2K 存储器的地址空间，则可以切断印刷线路板引线孔之间的铜泊连线，按装16只引脚的插座，其中插入具有相应连线的插头。

三、系统 RAM

U16—U19为总计 2K字节的 RAM（2114型），U20—U23为另外的2K 字节RAM。RAM中部分单元用作 TPBUG 的暂存区和栈区，如表3.2所示。

RAM 与 CPU 之间的连接如图3.3所示。

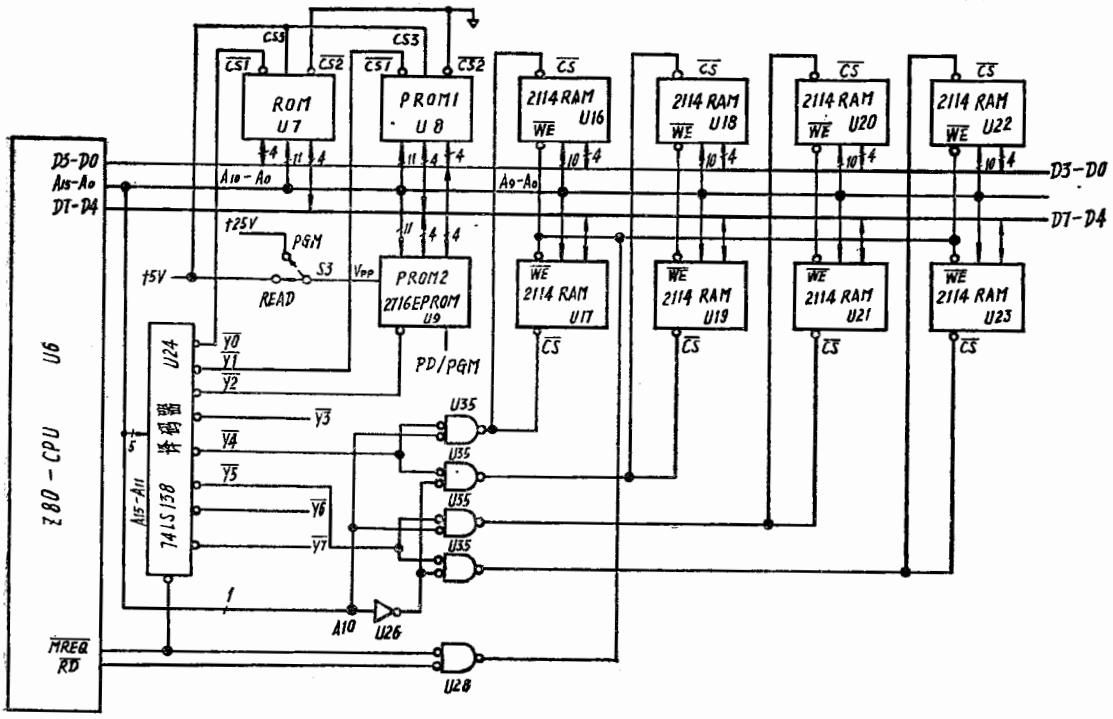


图3.3 RAM与CPU之間的連接

四、PROM1

当 RESET 信号为有效时，从 TPBUG 的 0000H 单元开始执行程序。在 00CDH 单元的指令检查开关 S2 的位置。如果 S2 置于 MON RST 位置，则继续执行 TPBUG 监控程序，在显示器上显示“—”，并扫描键盘的输入。如果 S2 置于 PROM1 RST 位置，则转移到起始地址为 0800H 的 PROM1 中的程序，从而可以不必通过键盘输入命令而进入用户程序。

五、PROM2—EPROM

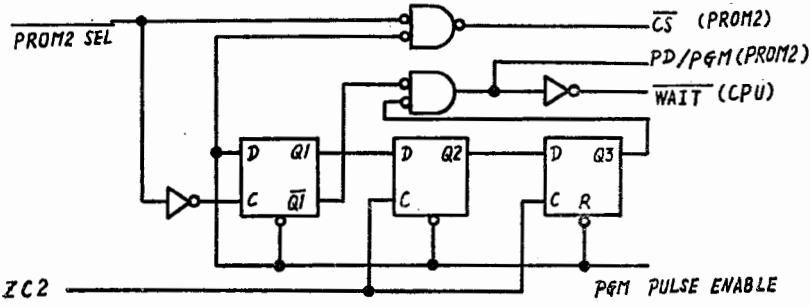
本节结合硬件和软件来解决 EPROM 的写入过程。

首先复习 2716/2758 型 EPROM 的操作方式，如表3.3所示。为了对 EPROM 进行写入，要求 S3 开关置于 PGM 位置，即 EPROM 的 Vpp 引脚接至 +25V 电源。此外要求CS = 1、PD/ PGM 引脚上输入一个 TTL 电平的脉宽为 50—55 毫秒的正脉冲。在写入时，地址总线上应出现 EPROM 被写入单元的地址，数据总线应出现被写入的内容。

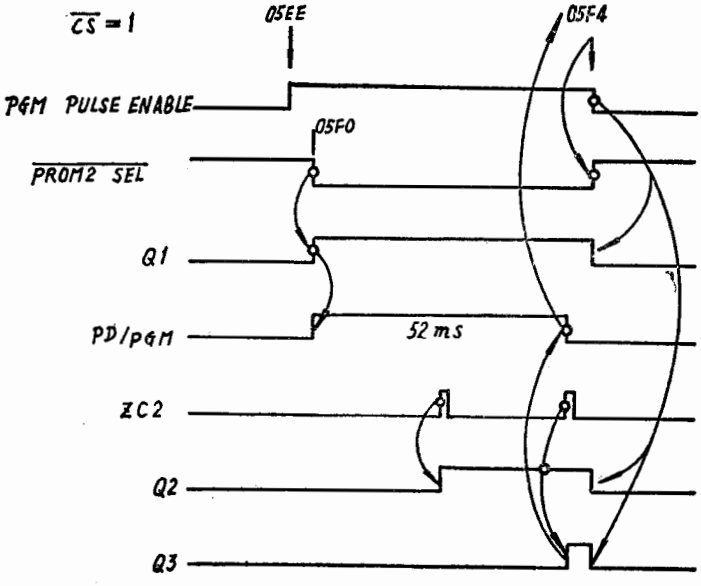
EPROM 写入的电路如图3.4 (a) 所示，操作的时间图如图3.4 (b) 所示。在写入前应做完一切准备工作，即用紫外线擦除 EPROM 中的内容（即各单元内容都为全1）。写入时，应在未接入电源的情况下插入插座 U9，再接上电源 +5V和 +25V，将要写入的内

表 3.3　　EPROM 的 工 作 方 式

工作方式 \ 管脚	PD/PGM	$\overline{CS}$	Vpp	数据线状态
讀	0	0	+5V	D_{out}
禁止	0	1	+5V	高阻抗
待机	1	×	+5V	高阻抗
写入	52ms	1	+25V	D_{in}
校核	0	0	+25V	D_{out}
禁止写入	0	1	+25V	高阻抗



(a) 电路



(b) 时間图

图 3.4　EPROM 写入的电路和时間图

容输入起始地址为 2000H 的 RAM 单元，将需要写入的字节数用数字键送入，字节数用四个数字表示，显示在左端四个 LED 显示器上，也就是说，它们存放在 DISMEM—DS

MEM3 存储单元内。然后将 S3 开关置于 PGM 位置，压下 MON 键使显示“—”。最后压下 PROM PROG 键，进入该键的动作程序（起始地址为 05D3H）：

首先将 CTC 通道 2 设定为定时器工作方式，每隔26毫秒送出一个脉冲，但并不需要产生中断。其次执行 05EEH 单元的指令，使 PGM PULSE ENABLE 信号变高电平，从而撤除对 Q1—Q3 触发器的封锁。下一条为 05FOH 单元的 LDI 指令，开始对 EPROM 进行写入，地址总线上出现 1000H，数据总线上出现 2000H 单元的内容，$\overline{\text{PROM2 SEL}}$ 变为低电平，Q1 变为高电平，PD/PGM 变为高电平，$\overline{\text{WAIT}}$ 变为低电平（有效），从而使 CPU 进入等待状态。当 CTC 通道 2 的 ZC2 发出第二个脉冲（即 52 毫秒以后）时，Q3 变高，从而使 PD/PGM 变低，$\overline{\text{WAIT}}$ 变高，CPU 结束等待状态。执行完 LDI 指令后，继续执行下一条指令（05F4H），使 PGM PULSE ENABLE 变低，从而使 $\overline{\text{PROM2 SEL}}$ 变高、Q1—Q3 变低。这一单元的写入过程到此结束。写入下一单元的过程与此相同。

2758 与 2716 的写入过程基本相同，两者只有一只引脚不同，即 2758 的引脚 19 为 AR 而不是 2716 的 A10。通过跳接线（jumper）可将 AR 端接至 +5V，或接至地，或接至地址总线的 A10。

3.5 I/O 接口

一、I/O 译码及空间分配

I/O 译码电路如图 3.5 所示。U36 单元的八中取一译码器将 I/O 口每四个为一组，每次选中一组，如表 3.4 所示。表中 $\overline{\text{PS5—PS7}}$ 连向布线区以供用户使用。对于 PIO 和 CTC，可以用 A1 和A0 线将表中所示地址分配给有关单元。

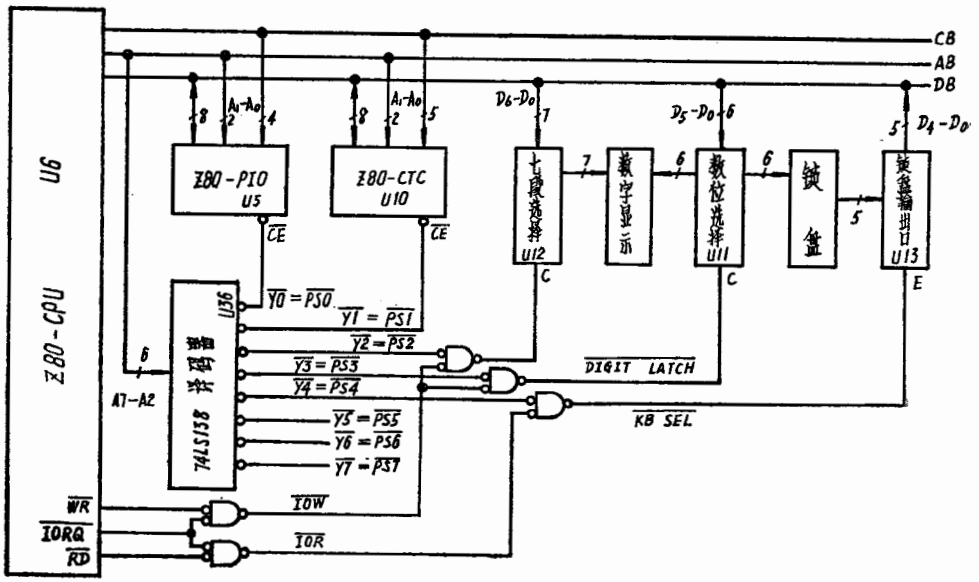


图 3.5 I/O 譯碼电路

表 3.4

A7—A2	譯碼器輸出	器　件	A1	A0	口	口地址
100000	$\overline{Y0}=\overline{PS0}=\overline{PIO\ SEL}$	Z80—PIO U5	0	0	口A数据寄存器	80H
			0	1	口B数据寄存器	81H
			1	0	口A控制寄存器	82H
			1	1	口B控制寄存器	83H
100001	$\overline{Y1}=\overline{PS1}=\overline{CTC\ SEL}$	Z80—CTC U10	0	0	通道0	84H
			0	1	道通1	85H
			1	0	通道2	86H
			1	1	通道3	87H
100010	$\overline{Y2}=\overline{PS2}=\overline{SEG\ LH}$	74LS273 U12 八鎖存器	×	×	七段选择 (只写)	88—8BH
100011	$\overline{Y3}=\overline{PS3}=\overline{DIG\ LH}$	74 LS 273 U11 八鎖存器	×	×	数位选择 (只写)	8C—8FH
100100	$\overline{Y4}=\overline{PS4}=\overline{KB\ SFL}$	74 LS 244 U13 八緩冲器	×	×	讀鍵值 (只讀)	90—93H
100101	$\overline{Y5}=\overline{PS5}$	沒使用				94—97H
100110	$\overline{Y6}=\overline{PS6}$	沒使用				98—9BH
100111	$\overline{Y7}=\overline{PS7}$	沒使用				9C—9FH

二、Z80—PIO

Z80—PIO 是通用的并行 I/O 口芯片。它有两个 8 位的口。每个口有两根联络线。Z80—PIO 的两个口全部供用户使用，本机并不予占用，因而这些线都接向布线区，可供用户附加电路。

每个口都可以使用联络线进行中断控制，并采用 IM2 中断方式。

三、Z80—CTC

Z80—CTC 有四个通道：

通道 0 —供用户使用，中断服务程序的起始地址为 2FD6H。用户可在 2FD6 处安排一个转移指令，转移到中断服务程序。

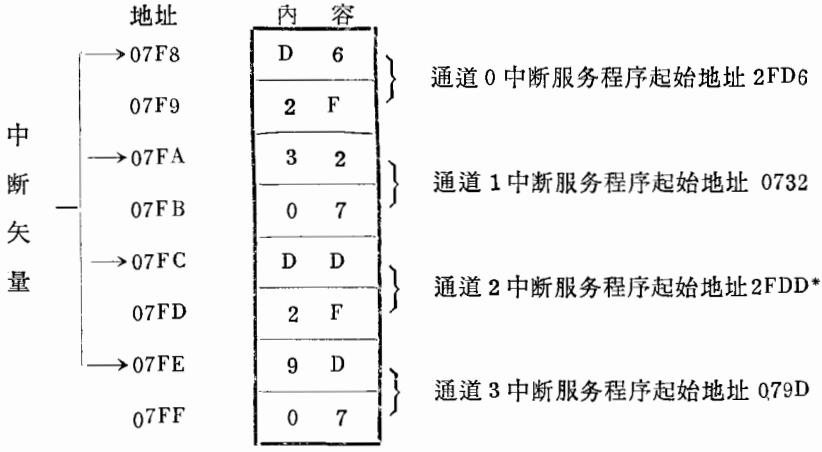
通道 1 —在本机中用于“CASS DUMP”键的动作程序，该程序将 RAM 中信息转储到盒式录音机的磁带中。该通道的中断服务程序的起始地址为 0732H。

通道 2 —在本机中用于 PROM PROG 键、SINGLE STEP 键和 MON 键的动作程序，

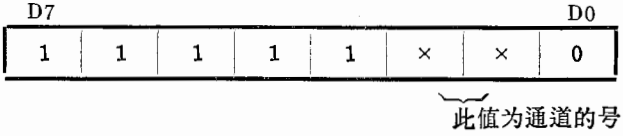
在PROM PROG 键动作程序中，通道2仅用来产生52 ms 的脉冲。在SINGLE STEP 键和MON 键的动作程序中，通道2用来产生不可屏蔽中断。它们都没有使用IM2 中断方式，因而也不需要使用中断矢量。

通道3—在本机中用于“CASS LOAD”键的动作程序，该程序将盒式录音机磁带中的信息传送到RAM中。该通道的中断服务程序的起始地址为079 DH。

各个通道的中断服务程序的起始地址表放在ROM 中，如下表所示：



CTC 四个通道的中断矢量按偶数依序排列，中断矢量低字节写入CTC 通道0，其格式如下：



通道0 的输入和输出都接向布线区以供用户使用。

四、键盘和显示

本机有六个七段LED 显示器，可以显示十六进制数，也可以显示其它的一些特定字符，在TPBUG 监控程序中可显示“—”和“ ’ ”和“空”。

所显示的数据（或字符）写入U12 的八锁存器74LS273（口地址为88H），至于要显示哪一位则由U11 单元的八锁存器74LS273（口地口为8CH）来选择。

七段LED 显示器的图形以及段号与数据位的对应关系如图3.6 所示。LED 显示器最左边的一位命名为DG0 位，其余的依次为DG1，DG2，DG3，DG4，DG5。例如，要在最左边的LED 显示器DG0 位上显示“4”，可以通过下列程序段来完成：

```
LD    A，15H
OUT   (88)，A
LD    A，20H
OUT   (8C)，A
```

* 如果用戶需要将通道2 設定为IM2 中断方式，则（2FDD）—（2FDF）将存放中断服务程序的第一条指令，通常是一条轉移指令，从而这三个单元不能被TPBUG 用作（IF），（PUFLR），（RFLG），用戶应予注意。监控程序不要求通道2 請求IM2 方式的中断。

其中 15H 的二进制表示为：

D7							D0
0	0	0	1	0	1	0	1
g	f	e	d	c	b	a	

D7 不起作用，对于D6—D0 来说 0 为有效，故 b，c，f，g 段亮，得“4”。

数 20H 的二进制表示为：

D7							D0
0	0	1	0	0	0	0	0
		↑					↑
		DG0					DG5

D7，D6 不起作用，对于 D5—D0 来说，1 为有效，故显示最左边的 DG0 位。

实际的显示过程并非如此简单。请参考 TPBUG 监控程序中00F4—0120H 的 DISUP 程序段以及起始地址为 07A6H 的七段显示表 SEGPT。

晶体管 Q1—Q7 用来提高驱动能力，U1—U3 驱动器用来增加吸收电流的能力。U11 不仅扫描显示，还能扫描键盘。例如，如图Ⅰ.1所示，当口（8C）U11 中数为 08H（即 L3 线为低电平，其余线为高电平）时，可检查 4，5，6，B，MON 键有否被压下。如果键 6 被压下，则 R2 线呈低电平，其余线呈高电平，即口（90）中低五位为 11011B。

由 08H 和11011B 两个数字可以求出键的偏移量（功能键的键偏移量都大于等于 10H。

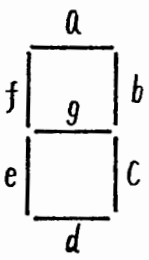


图 3.6 （a）七段显示器；

数字	D7 D6 D5 D4 D3 D2 D1 D0 g f e d c b a	字形
0	1 0 0 0 0 0 0	0
1	1 1 1 1 0 0 1	1
2	0 1 0 0 1 0 0	2
3	0 1 1 0 0 0 0	3
4	0 0 1 1 0 0 1	4
5	0 0 1 0 0 1 0	5
6	0 0 0 0 0 1 0	6
7	1 1 1 1 0 0 0	7
8	0 0 0 0 0 0 0	8
9	0 0 1 1 0 0 0	9
A	0 0 0 1 0 0 0	A
B	0 0 0 0 0 1 1	b
C	1 0 0 0 1 1 0	C
D	0 1 0 0 0 0 1	d
E	0 0 0 0 1 1 0	E
F	0 0 0 1 1 1 0	F
'	1 1 1 1 1 0 1	'
空	1 1 1 1 1 1 1	
-	0 1 1 1 1 1 1	-

（b）七段显示的代码与字形的关系

例如EXEC键为10H，SS键为11H等，详见起始地址为07B9H的键值表KYTBL）。

详细的键盘分析程序DECKY请参考附录三的TPBUG。

五、录音机接口电路（一）—转储（Dump）

将存放在RAM中易失的程序或信息转储到廉价的盒式磁带中，通常可以使用市场上出售的廉价的录音机和录音磁带。在转储过程中我们采用了“堪萨斯（Kansas）城标准”的记录技术。

用于盒式磁带上记录数据的格式必须遵守两个标准：记录“1”和“0”的堪萨斯城标准和记录数据块的Intel Hex Format。

堪萨斯城标准是1975年9月7日和8日在美国堪萨斯城由BYTE杂志主持的专题讨论会上制定的。这次专题讨论会的目的是，为业余爱好者的录音磁带记录技术制订标准。堪萨斯城标准的要点是（本机是符合这些规定的）：

1）八个频率为2400Hz的周波表示逻辑1；

2）四个频率为1200Hz的周波表示逻辑0；

3）一个记录字符的构成为：一个逻辑0的起始位、7或8个数据位、两个或两个以上的停止位（本机使用一个七位的ASCII数据字符和一个停止位）；

4）七个ASCII数据位的次序为：最低位在先，最高位在后；

5）在数据块之前有30秒以上的全1导引段，之后有5秒的全1结尾段；

6）数据的传送速率为300波特（每位用3.33ms）；

7）数据块内容不作规定。

因为堪萨斯城标准没有对所记录数据块的内容作出规定，所以选用另一个标准—Intel Hex Format—来规定数据块的结构。该标准的要点是：

1）在数据块内的每个记录以冒号（：）开始，以回车和换行符号结束；

2）一切信息用ASCII（7位，无奇偶校验）表示；

3）数据记录格式：

字节1	冒号（：）分解符
字节2—3	该记录中二进制的字节数，最多为16个二进制字节（即32个ASCII字节）。
字节4—5	该记录起始地址的高位字节。
字节6—7	该记录起始地址的低位字节。
字节8—9	记录类型：ASCII“0 0”。
字节10—11	除了分解符与回车和换行符以外的所有字节的“检查和”，此“检查和”为各二进制字节和的负数。
回车和换行	

4）“文件结束”记录格式：

字节1	冒号（：）分解符
字节2—3	ASCII“0”
字节4—5	ASCII“0”
字节6—7	ASCII“0”

字节8—9	“01”记录类型：ASCII “01”
字节10—11	“检查和”

用于转储的接口电路如图Ⅰ.1下部所示。利用程序(详细内容请参考TPBUG中起始地址为04CAH的CASS DUMP键盘动作程序）设定CTC通道1为定时器工作方式，使ZC1产生4800Hz或2400Hz的脉冲，通过U14的分频，即可得到2400Hz或1200Hz的脉冲，再通过阻容（R33和C7）滤波，滤去其中的高频分量，即可接至录音机的Auxiliary或MIC输入端。

六、录音机接口电路（二）—输入（Load）

从J1端输入录音机的脉冲信息，U4为限幅与整形电路，使U5能有一个不畸变的方波输入。J1端输入的脉冲的峰—峰值应为2V左右，输入电平的高低可以用U4驱动LED显示器来指示。U14和U15是频率检侧器，用来鉴别1200Hz和2400Hz。当U7输出1200Hz脉冲（“0”）时，U14的$\overline{Q}$输出为0，当U7输出2400Hz脉冲（“1”）时，U14的$\overline{Q}$输出为1。$\overline{Q}$的数据由CPU通过U13来读取，并拼装成一个ASCII字符，读入的ASCII字符再由程序转换成二进制数。

可见，在TPBUG的控制下，Z80—CPU和Z80—CTC能实现UART（通用异步接收发送器）的功能，以接收异步的串行数据，并形成并行的字，然后存入存储器。

3.6 其他部分

一、单步逻辑

当压下SINGLE STEP键后，进入该键的动作程序（起始地地为0272H）。它将CTC通道2设定为定时器工作方式，并恢复现场，即从栈（用户寄存器存放区）中弹出各个寄存器的内容，然后执行用户的指令。在开始执行用户指令时，CTC通道2的ZC2输出一个脉冲，这个脉冲通过U33和U34送向CPU的$\overline{NMI}$输入端，因而在这一条用户指令执行完毕后，即返回TPBUG，执行0066H单元的指令。此后程序段（起始地址为01CBH）将关闭CTC通道2，并保护现场。最后进入起始地址为007EH的程序段，显示PC和累加器A的内容。

二、中断链路

Z80系列中I/O接口器件内均设有中断控制电路，从而与Intel 8080/8085系列不同，不必使用专用的中断控制器件。中断的优先级根据该芯片在链路中的位置来决定。本机中CTC具有较高的中断优先级，PIO具有较低的中断优先级。PIO芯片的IE0线接向布线区，供用户再增加I/O芯片时使用。

三、复位电路（Reset）

图3.7示出了本机所用的复位电路。有两种情况可以产生复位信号：一是上电复位，在刚接通电源时，由于电容C17的作用，使$\overline{RESET}$短暂的保持低电平信号另；一是使用RESET按钮，在压下此按钮期间，$\overline{RESET}$保持低电平。

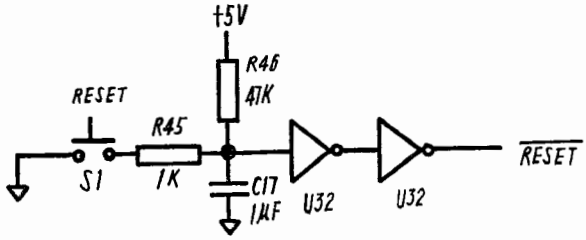


图3.7　复位电路

四、电压保护电路（任选）

本机使用7805集成电路。它的输入应大于+7V。它的输出将稳定在+5V。

五、S—100总线

本机可安装两个S—100总线插座。这种接口通常能与静态存储器和I/O扩充板兼容。如果接口需要8080所特有的控制信号，例如SYNC，INTA，DBIN，POC，PWR，PRD等，则尚需在布线区附加一些逻辑电路，以进行信号变换；并/或将附加的信号接到S—100总线。在印刷线路板上方有+8V，±18V电源的接线端，以便接入相应的电源。

六、布线区

布线区可供用户附加25—30个集成电路，以便扩充存储器、增加CRT接口及其他电子实验电路。Z80—CPU，PIO，CTC（部分）的信号线以及译码出来的系统信号都引到布线区的附近，以便用户增加集成电路时使用这些信号。电源线和地线在印刷线路板的背面平行走线，使集成电路紧挨着这一低阻抗的电源，从而减少用户电路的噪声干扰。

第四章　程序举例

本章目的是想通过几个程序的导入、执行和调试，使用户进一步熟悉键盘的操作功能和本机的使用方法，以及程序设计的一些基本知识。

4.1　熟悉键盘操作和 TPBUG 命令的使用

设有以下程序使用 TPBUG 命令，通过键盘操作，输入单板机*。

```
                ORG    2000
2000    3E  AA  LD  A,0AAH; 将立即数 AA 送入 A 累加器
2002    06  BB  LD  B,0BBH ; 将立即数 BB 送入 B 寄存器
2004    78      LD  A,B    ; 寄存器 B 内容送入 A 累加器
2005    0E  CC  LD  C ,0CCH ; 将立即数 CC 送入 C 寄存器
2007    79      LD  A,C    ; 寄存器 C 内容送入 A 累加器
2008    3E  AA  LD  A,0AAH; 立即数 AA 送入 A 累加器
200A    76      HALT       ; 暂停
```

使用 TPBUG 命令进行的操作如表 4.1：

表 4.1

按　　鍵	显　　示 ADDRESS	DATA	說　　明
RESET　(S1)	—		按下开关 S 1—RESET，立刻显示标志 "—"
2　0　0　0	2　0　0　0		輸入程序的首地址 2000
MEM　EXAM	2　0　0　0	××	使用 MEM EXAM 鍵讀出存儲单元2000的內容 ××
3　E	2　0　0　0	3 E	对 2000 存儲单元写入操作碼 3E
NEXT	2　0　0　1	××	用 NEXT 鍵順序对存儲单元进行讀出操作
A　A	2　0　0　1	A A	对存儲单元 2001 写入操作数 AA
NEXT	2　0　0　2	××	讀出 2002 存儲单元
0　6	2　0　0　2	0 6	对存儲单元 2002 写入操作 碼 06

* 在本例以及下面各例中，所有用戶按下的按鍵，都在它下面划一横綫，而 TPBUG 的响应在显示器上显示的就表示在方框中。且数字如无特殊标記，均为十六进制数字。

续表 4.1

按　鍵	显　示 ADDRESS	DATA	說　明
NEXT	2 0 0 3	××	讀出 2003 存儲单元
B B	2 0 0 3	b b	对存儲单元 2003 写入操作数 BB
NEXT	2 0 0 4	××	讀出存儲单元 2004
7 8	2 0 0 4	7 8	对存儲单元 2004 写入单字节指令 78
NEXT	2 0 0 5	××	讀出存儲单元 2005
0 E	2 0 0 5	O E	对存儲单元 2005 写入操作碼 0E
NEXT	2 0 0 6	××	讀出存儲单元 2006
C C	2 0 0 6	C C	对存儲单元 2006 写入操作数 CC
NEXT	2 0 0 7	××	讀出存儲单元 2007
7 9	2 0 0 7	7 9	对存儲单元 2007 写入单字节指令 79
NEXT	2 0 0 8	××	讀出存儲单元 2008
3 E	2 0 0 8	3 E	对存儲单元 2008 写入操作碼 3 E
NEXT	2 0 0 9	××	讀出存儲单元 2009
A A	2 0 0 9	A A	对存儲单元 2009 写入操作数 AA
NEXT	2 0 0 A	××	讀出存儲单元 200A
7 6	2 0 0 A	7 6	对存儲单元 200A写入单字节暫停指令 76
MON	—		用 MON 鍵中止对 存 儲单元的讀、写操作，以 便进行其他操作
A	A		讀出 A 累加器

续表 4.1

按　　鍵	显　　示 ADDRESS	DATA	說　　明
REG　EXAM	A	××	在程序未执行前,A 的內容是随机数
MON	—		
B	b		讀出 B 寄存器
REG　EXAM	b	××	程序未执行，寄存器 B 的內容是随机数
C	C		讀出 C 寄存器
REG　EXAM	C	××	程序未执行，C 寄存器的內容是随机数
MON	—		中止上述操作。下面用 SINGLE STEP 鍵，采用单步方式执行 ORG 2000程序
PC (1)	1		
REG　EXAM	1　× ×	××	讀出程序計数器 PC
2　0　0　0	1　2　0	0 0	单步执行程序方式，首先要修改程序計数器 PC 內容,使它指向程序首地址
SINGLE　STEP	2　0　0　2	A A	执行完 2000 指令，立即数 AA 送入累加器 A，PC 指向下一条指令的地址。“DATA” 显示器显示 A 的內容 “AA”
SINGLE　STEP	2　0　0　4	A A	2002 指令执行完毕，立即数 BB 送入 B 寄存器，PC 指向下一条指令地址，这时累加器 A 內容不变，“DATA” 仍然显示 “AA”
MON	—		
B	b		
REG　EXAM	b	b b	讀出 B 寄存器的內容，驗証指令的执行
SINGLE　STEP	2　0　0　5	b b	B 寄存器的內容送入 A 累加器，PC 指向下一条指令的地址

续表 4.1

按　　鍵	显　　示 ADDRESS　　DATA	說　　明
SINGLE STEP	2 0 0 7　　b b	立即数 CC 送入 C 寄存器，PC 指向下一条指令的地址。A累加器的內容不变
SINGLE STEP	2 0 0 8　　C C	C 寄存器的內容送入 A 累加器，PC 指向下一条指令的地址
SINGLE STEP	2 0 0 A　　A A	
MON	—	中止 SINGLE STEP 操作方式
A REG EXAM	A　　A A	先按下 A，后按 REG EXAM 鍵，讀出 A 累加器內容
MON	—	
B REG EXAM	b　　b b	讀出 B 寄存器內容
MON	—	
C REG EXAM	C　　C C	讀出 C 寄存器的內容
MON	—	
PC REG EXAM	1 2 0　　O A	讀出 PC 的內容，它現在指向 200 A 存儲单元，是存貯暫停指令的地址
MON	—	中止讀出方式，下面开始設置多个断点操作，如在 2002、2005、2008、200 A 設置断点
2 0 0 2	2 0 0 2	先送入四位数碼表示断点单元地址
BREAK POINT	2 0 0 2	按下 BREAK POINT 鍵，断点被接受
MON	—	
2 0 0 5 BREAK POINT	2 0 0 5	先按四位地址数碼鍵，后按 BREAK POINT 鍵，在 2005 单元設置断点

续表 4.1

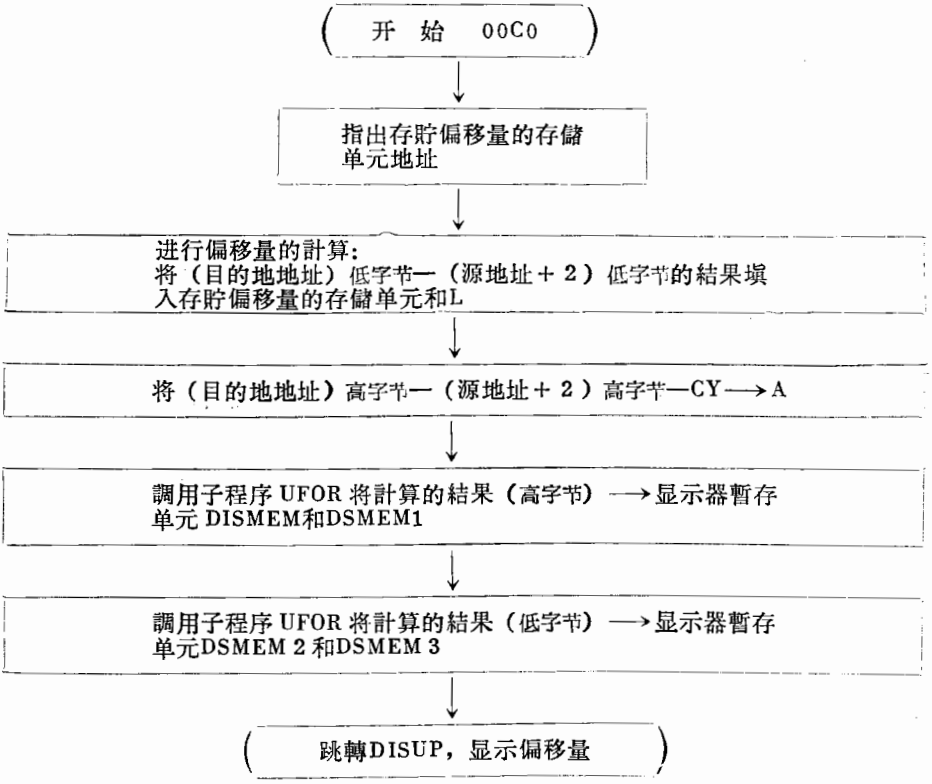
按　　鍵	显　　示 ADDRESS　　DATA	說　　明
MON	—	
2 0 0 8 BREAK POINT	2 0 0 8	在2008单元設置断点
MON	—	
2 0 0 A BREAK POINT	2 0 0 A	在200A 单元設置断点
MON	—	中止設置断点的操作,下面用EXEC鍵执行程序
2 0 0 0	2 0 0 0	送入程序首地址2000，按EXEC鍵，执行程序
EXEC	2 0 0 2　　A A	程序从2000单元开始执行，显示第一个断点地址和执行第一条指令后，A的內容AA
EXEC	2 0 0 5　　b b	显示第二个断点地址和执行此断点前面的指令后，A的內容“bb”
EXEC	2 0 0 8　　C C	显示第三个断点地址以及完成此断点前面的指令后，A的內容“CC”
EXEC	2 0 0 A　　A A	显示第四个断点,以及完成此断点前面的指令后，A的內容“AA”
RESET	—	按动RESET或SINGLE STEP或BREAK POINT鍵，均可取消断点
2 0 0 0	2 0 0 0	輸入程序首地址
EXEC		从程序首地址开始执行程序，一直执行到暫停指令。显示器不显示
MON	—	按下MON鍵,回到TPBUG控制,显示标志“—”
PC REGEXEC	1　　2 0　　O b	检查PC，它指向存貯HALT指令的下一个存儲单元的地址
MON	—	重新回到TPBUG控制,可相应检查各寄存器內容，与上述执行程序方式相比較

4.2 相对轉移指令中偏移量的計算

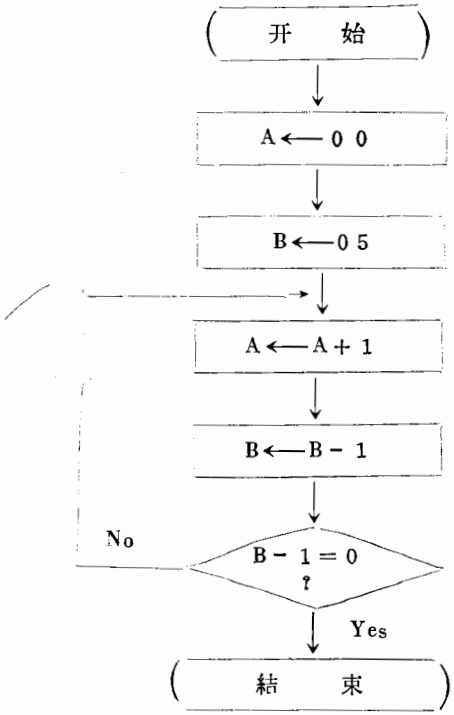
本例的内容是计算相对转移指令的偏移量。偏移量是以十六进制表示的带符号的2的补码。TPBUG具有自动计算并填写相对偏移量的子程序。子程序的入口地址是00C0H。相对转移源地址装入DE寄存器对，目的地地址装入HL寄存器对，偏移量则装入以(DE+1)内容为地址的存储单元。

```
；子程序
                        ORG   00C0
00C0    1813            JP    RESTR3—$
00D5    13    RESTR3:   INC   DE      ；指出填写偏移量所在存储单元地址
00D6    D5              PUSH  DE      ；将存贮偏移量的存储单元地址转入IX
00D7    DDE1            POP   IX      ；IX←DE
00D9    13              INC   DE
00DA    7D              LD    A,L
00DB    93              SUB   E       ；A←(目的地地址)低字节
                                        -(源地址+2)低字节
00DC    6F              LD    L,A
00DD    DD7700          LD(IX+0),A；将计算结果(低字节)填入存贮偏移量的
                                    存储单元
00E0    7C              LD  A, H;
00E1    9A              SBC A, D;     A←(目的地地址)高字节
                                      —(源地址+2)高字节-CY
00E2    DD21F72F        LD  IX,DISMEM
00E6    CD3C06          CALL  UFOR1；利用TPBUG的子程序UFOR1，将计
                                    算结果（高字节）写入显示暂存单元
                                    2FF7(DISMEM)和2FF8(DSMEM1)
00E9    DD21F92F        LD  IX,DSMEM2
00ED    7D              LD  A, L
00EE    CD3C06          CALL  UFOR1；调用子程序UFOR1，将计算结果（低
                                    字节）写入显示暂存单元2FF9（DSM-
                                    EM2）和2FFA（DSMEM3）
00E1    C3F400          JP    DISUP；调用TPBUG子程序DISUP显示计算
                                    结果（四位数字）
```

计算相对转移指令偏移量的子程序流程图



主 程 序 流 程 图



通过键盘操作，输入下面程序：

```
                    ORG 2000
2000    3E  00      LD A,00
2002    06  05      LD B,05H
2004    3C          LOOP:INC A
2005    10  ?       DJ NZ,LOOP-$
2007    76          HALT
```

表 4.2

按　　鍵	显　　示 ADDRESS　　DATA	說　　明
RESET	—	等待鍵盘命令
2 0 0 0 MEM EXAM	2 0 0 0　× ×	輸入程序
3　E	2 0 0 0　3 E	
NEXT 0 0	2 0 0 1　0 0	
NEXT 0 6	2 0 0 2　0 6	
NEXT 0 5	2 0 0 3　0 5	
NEXT 3 C	2 0 0 4　3 C	
NEXT 1 0	2 0 0 5　1 0	
NEXT	2 0 0 6　× ×	2006 单元不装入信息，这是准备填入計算的結果——相对轉移偏移量
NEXT 7 6	2 0 0 7　7 6	
MON	—	中止輸入程序的操作，准备进行計算偏移量的操作
H　REG EXAM	7　× ×	目的地地址→HL
2 0	7　2 0	
MON	—	
L　REG EXAM	8　× ×	

续表 4.2

按　　鍵	显示 ADDRESS	显示 DATA	說　　明
0 4	8	0 4	
MON	—		
D REG EXAM	d	× ×	源地址→ DE
2 0	d	2 0	
MON	—		
E REG EXAM	E	× ×	
0 5	E	0 5	
MON	—		
0 0 C 0	0 0 C 0		計算相对轉移偏移量的子程序从 00C0 单元开始执行
EXEC	F F F d		显示偏移量 FFFDH，即—0003H的 2 的补碼，因Z80指令系統規定相对偏移量为两字节(即从+127D—-128D)，所以实际偏移量为FDH
MON	—		
2 0 0 6 MEM EXAM	2 0 0 6	F d	检查2006单元 已自动填入偏移量 FD
MON	—		下面用SINGLE STEP方式执行主程序
PC REG EXAM	1 × ×	× ×	修改程序計数器
2 0 0 0	1 2 0	0 0	将PC 指向程序的首地址
SINGLE STEP	2 0 0 2	0 0	执行完第一条指令，显示 A 的內容和下一条指令的地址
SINGLE STEP	2 0 0 4	0 0	
SINGLE STEP	2 0 0 5	0 1	第一次执行“INC A”指令，A第一次增量

续表 4.2

按　　鍵	显示 ADDRESS	DATA	說　　明
SINGLE STEP	2 0 0 4	0 1	循环回去，这时若检查 B寄存器，B 的內容是多少？
SINGLE STEP	2 0 0 5	0 2	累加器 A第二次增量
SINGLE STEP	2 0 0 4	0 2	循环回去
SINGLE STEP	2 0 0 5	0 3	A 第三次增量
SINGLE STEP	2 0 0 4	0 3	循环回去
SINGLE STEP	2 0 0 5	0 4	A 第四次增量
SINGLE STEP	2 0 0 4	0 4	循环回去
SINGLE STEP	2 0 0 5	0 5	A 第五次增量，这时若检查 B 的內容，它是 01
SINGLE STEP	2 0 0 7	0 5	执行条件轉移指令“D JNZ”，当条件 B-1＝0 成立时，即依次执行下一条指令，而不作轉移
MON	—		等待鍵盘命令

4.3 軟件延时

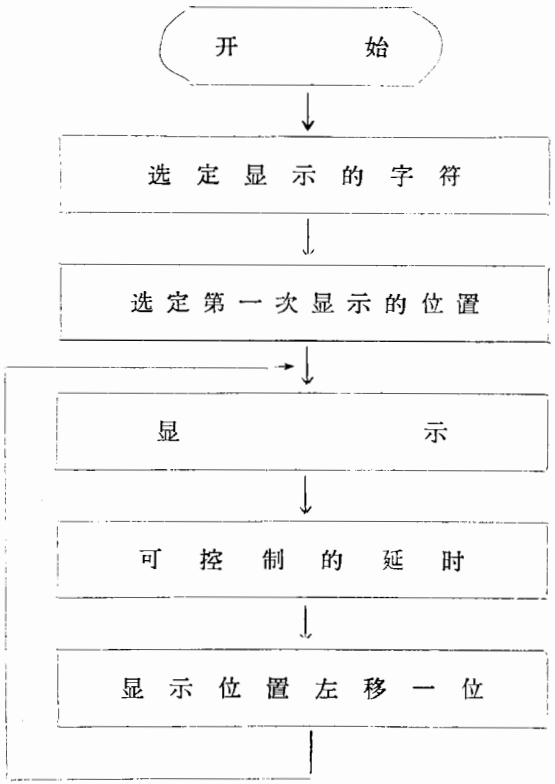
下面这个程序是使字符“8”，从显示器的右端向左端不停地循环移动。在每个位置上停留的时间，可以由改变装入 B 寄存器的时间常数来决定，约由 20ms 到 5.2sec 。

```
；程序
                        ORG 2000
2000    3E00            LD A,00
2002    D388            OUT (88H),A     ;选定字符“8”
2004    3E01            LD A,01H        ;选定左边第一个显示器亮
2006    D38C    LOOP:   OUT (8CH),A
2008    06FF            LD B,OFFH       ;B←延时时间常数
200A    CD4F06  LOOP1:  CALL D20MS      ;调用 TPBUG 中延时约 20ms 子程序
200D    10FB            DJ NZ,LOOP1-$   ;最长延时约 5.2sec
200F    07              RLCA            ;左移一位
```

```
2010    18F4        JR LOOP-$      ;循环回去
```

首先输入程序，按下MON，输入程序首地址2000，按下EXEC，字符“8”则在显示器上从右到左循环移动。按下MON，程序则停止执行。控制延时子程序D20MS的调用次数（也就是控制装入B寄存器的时间常数）就可以控制字符在显示器上的停留时间。

程　序　流　程　图



4.4 Z80-CTC 的应用

```
                ORG   2000
2000    3E21    LD    A,21H
2002    ED47    LD    I,A       ;设定中断矢量高字节I←21H
2004    310023  LD    SP,2300H  ;建立栈指示器
2007    3E00    LD    A,00
2009    D384    OUT   (84H),A   ;外部设备提供中断矢量低字节
200B    3EA5    LD    A,0A5H
200D    D384    OUT   (84H),A;设定CTC的工作方式,输入通道控制字
200F    3EFF    LD    A,OFFH
```

```
2011    D384            OUT(84H),A ;输入时间常数，延时约 33ms
2013    3E01            LD  A,01H  ;设定第一次显示 01
2015    ED5E            IM2        ;设定中断方式 2
2017    FB      LOOP:   EI
2018    76              HALT
2019    C31720          JP  LOOP
    ;中断服务程序入口地址表
2100    00  22
    ;中断服务程序
2200    FB              EI         ;开中断
2201    07              RLC  A     ; A 累加器左移一位
2202    ED4D            RETI       ;中断返回
```

本例程序是使 Z80-CTC 的 0 通道工作在定时器方式，经过一个整定延迟时间后(本例整定延迟时间约 33ms)，发出一个中断请求。改变输入到 CTC 的时间常数，可以改变请求中断的时间。

输入程序，按以下步骤进行调试：

1．采用 SINGLE STEP 工作方式，首先修改程序计数器 PC 为 2000H 。步进执行程序到显示地址为 2018H 时，稍停一会。

2．继续按动 SINGLE STEP 键，注意观察显示器显示 A 累加器的内容。每执行完一次 2201H 指令， A 的内容便左移一位，显示器按 01,02,04,08,10,20,40,80 的次序循环变化。

3．当 2201H 地址显示时，停止按 SINGLE STEP 。按下 MON 键，检查栈指示器 SP 的内容，这时应为 22FEH 。这是因为从栈顶地址 2300H 压入了两字节。检查 22FEH 和 22FFH 单元的内容，应该是 18H 和 20H ，这是中断返回的地址 2018H 。

4．按下 MON 键，然后再按下 SINGLE STEP 键，直至显示 2018H 。这时，再检查栈指示器 SP 内容，应该是 2300H 。这表明已经执行了中断返回 RETI 指令，中断返回地址 2018H 已从栈中弹出。

5．用 BREAK POINT 键，在 2200H 设置一个断点，用 EXEC 键连续执行程序。先输入首地址 2000H，按下 EXEC 键，立刻显示断点地址 2200H 和这时 A 累加器的内容。连续不断按动 EXEC， A 累加器内容不断变化，而且每按一次，改变显示一次，按 01,02,04,08,10,20,40,80 的顺序改变。

可利用 RESET 键使 CTC 复位。

下面这个程序，是将本例稍加修改后，再与 4.3 例结合起来的一个简单表演程序。

输入程序，按下 RESET 键，输入程序首地址 2000H， 按下 EXEC 键，字符“ 8 ”快速从显示器右端移动到左端，经过十次循环之后，立刻变为较慢的一次移动，接着又是十次较快的循环移动，又接着一次慢速移动……，这样不停地在显示器上循环移动。根据需要，可以很方便地改变显示的字符和循环的速度。按下 RESET 键，移动立刻停止，回到 TPBUG 控制。

```
    ; 程序
                    ORG  2000
2000    3E21        LD   A,21 H
2002    ED47        LD   I,A            ;设定中断矢量高字节
2004    3E00        LD   A,00
2006    D384        OUT  (84H),A        ;外部设备提供中断矢量低字节
2008    3EA5    ST:LD   A,0A5H
200A    D384        OUT  (84H),A        ;输入通道控制字，设定 CTC 工作方式
200C    3EFF        LD   A,0FFH
200E    D384        OUT  (84H),A        ;输入时间常数
2010    0EF6        LD   C,0F6H         ;设定快速循环次数(10D 次)
2012    ED5E        IM2                 ;设定中断方式 2
2014    FB     LOOP:EI                  ;开中断
2015    76          HALT
2016    0C          INC  C
2017    20FB        JR   NZ,LOOP-$      ;C≠0 循环返回
2019    3E03        LD   A,03H
201B    D384        OUT  (84H),A        ;禁止通道中断
201D    1E5F        LD   E,5FH          ;设定字符循环一次的时间常数
201F    CD5020      CALL DELAY          ;调用子程序
2022    C30820      JP   ST             ;循环返回，重新工作
    ; 中断服务程序入口地址表
2100    0022
    ; 中断服务程序
2200    IE0A        LD   E,0AH          ;设定字符循环速度的时间常数
2202    CD5020      CALL DELAY
2205    ED4D        RETI
    ; 子程序
2050    3E00 DELAY:LD   A,00            ;选定字符“8”
2052    D388        OUT  (88H),A
2054    3E01        LD   A,01H
2056    D38C LOOP1:OUT  (8CH),A
2058    43          LD   B,E
2059    CD4F06LOOP2:CALL D20MS
205C    10FB        DJ   NZ,LOOP2-$
205E    CB6F        BIT  5,A            ;测试 A 的第 5 位
2060    2003        JR   NZ,LOOP3-$
2062    07          RLCA
```

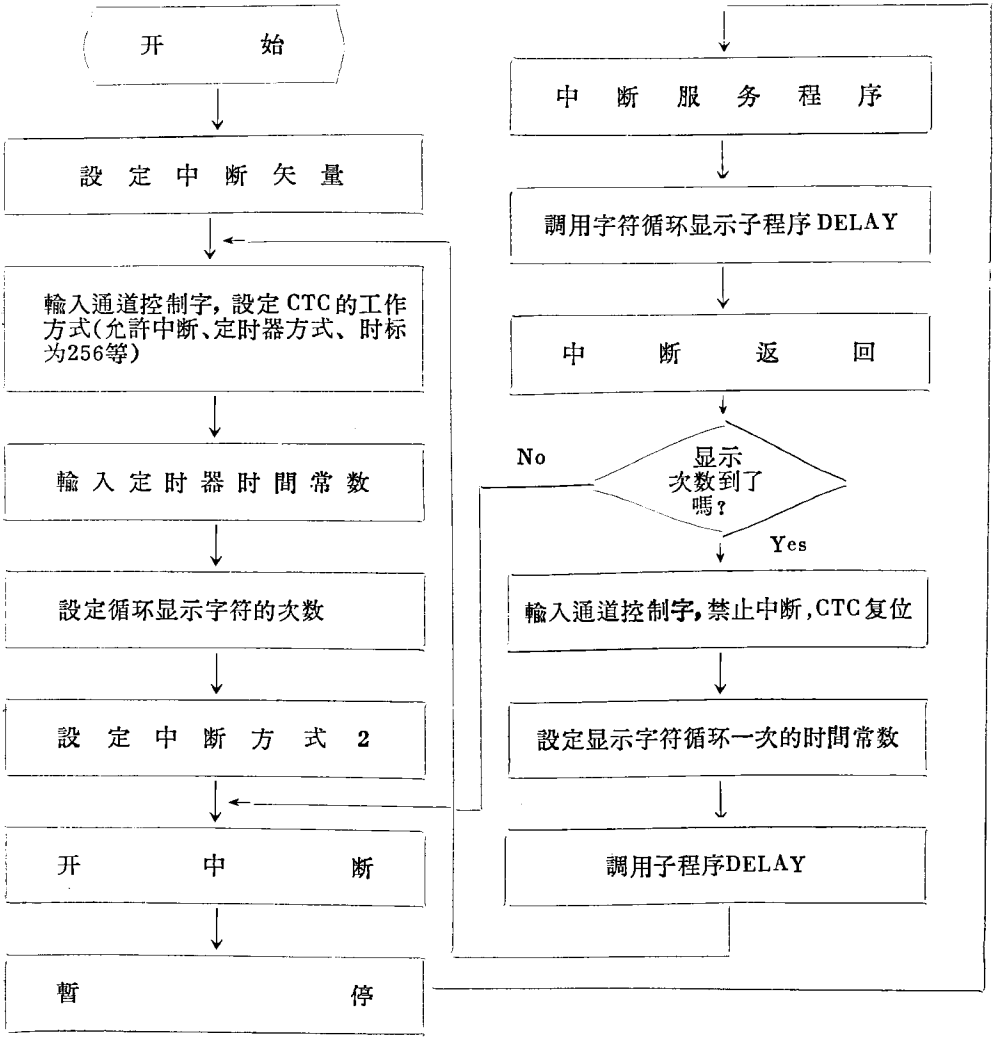
```
2063    18F1          JR  LOOP1-$
2065    C9      LOOP3:RET
```

注：(1)这个〔DELAY〕子程序，是将4.3例稍加修改而成。这种设计方法，不仅易于调试和检查，而且也节省存储单元。

(2)若想显示别的字符，可参考下表，修改DELAY子程序的第一条指令LD A，00 H中的立即数(即修改2051存储单元的内容)。

显示的字符	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
立即数	40	79	24	30	19	12	02	78	00	18	08	03	46	21	06	0E

程　序　流　程　图



4.5 Z80-PIO 的应用

首先用一个 10KΩ 电阻，将 Z80—PIO 的 $\overline{\text{ASTB}}$ 脚连接到电源 +5V 端，然后输入下列程序，并用 BREAK POINT 键在 2200H 单元设置断点。

输入程序首地址 2000H，按下 EXEC 键，CPU 执行程序到暂停指令 HALT，显示器无显示。

取一导线，使 $\overline{\text{ASTB}}$ 脚碰一下地 (GND)，相当于输入一负脉冲。此脉冲的上升沿使 PIO 发出一中断请求($\overline{\text{INT}}$)，CPU 将执行中断服务程序，把由 BREAK POINT 键设置的断点 2200H 显示出来。

```
        ; 程序
                              ORG   2000
2000    3E 21                 LD    A, 21H
2002    ED 47                 LD    I, A          ; 设定中断矢量高字节
2004    3E 00                 LD    A, 00
2006    D3 82                 OUT   (82H), A      ; 外部设备提供中断矢
                                                    量低字节
2008    3E 4F                 LD    A, 4FH
200A    D3 82                 OUT   (82H), A      ; 对 PIO 的 A 口设置
                                                    方式 1
200C    3E 87                 LD    A, 87H
200E    D3 82                 OUT   (82H), A      ; 设置允许中断控制字
2010    ED 5E                 IM2                 ; 中断方式 2
2012    FB            LOOP:   EI
2013    76                    HALT
2014    C31220                JP    LOOP—$
        ; 中断服务程序入口地址表
2100    00 22
        ; 中断服务程序
2200    FB EI
2201    ED4D RETI                                 ; 中断返回
```

可以将 4.4 例的中断服务程序及 DELAY 子程序作为本例的中断服务程序。这时，输入程序之后，按下 RESET 键，输入程序首地址 2000H，按下 EXEC 键，显示器无显示，将 $\overline{\text{ASTB}}$ 脚碰一下地，字符“8”立刻从显示器右端移到左端，最后停留在左端不动。再碰一次，再重复一次上述现象。若将 $\overline{\text{ASTB}}$ 脚碰一下地的操作得不好时，字符“8”会循环移动两次才停留在左端不动。这时，可以用一个 10KΩ 电阻把 $\overline{\text{ASTB}}$ 脚与地连接，然后输入程序，用 EXEC 键执行程序，然后通过一个 3KΩ 电阻碰一下 +5V 端，字符立即从显示器右端移到左端并停留在左端不动。再碰一次，再重复上述现象一

次。若条件许可，用 TTL 电平的单脉冲触发就更好些。

4.6 EPROM 的写入程序

本机可以将用户所需要的程序写入到 EPROM（1K 字节的2758 或 2K 字节的2716）中。在前面按键操作说明中已讲到，可以使用 PROM PROG 键，将从 2000H 单元开始的程序复制到从 1000H 单元开始的 EPROM 中去。复制的字节数，由按键盘输入的4位十六进制数字表示。这个操作虽然简便，但源程序和目标程序的首地址均被限定在2000H 和 1000H，有时使用就不方便。下列两个应用程序的例子，用户可以将存储器数据块复制到 EPROM 中的任意存储单元。

这两个应用程序，如果是经常使用的话，可以事先录制在盒式录音机的磁带上。需要时，从本系统盒式磁带机接口 J1 输入 RAM 中；或者将程序写在 EPROM 里，而将这个写有程序的 EPROM 插在 PROM1 插座上，供在线使用。

在第一个应用程序中，使用了 Z80 数据块传送指令。它可以用来将贮存在存储器中的数据块（源数据）传送到任意地址的存储单元（目的地数据）；它也可以将插在 PROM1 或 PROM2 插座上的 EPROM 数据复制到 RAM 中去。

使用这个程序之前，先用 MON 和 REG EXAM 键对 Z80 寄存器进行予置数据：

HL——源数据地址，高字节在 H；

DE——目的地数据地址，高字节在 D；

BC——数据传送的总字节数，高字节在 B。

然后执行下列数据块传送程序：

```
;
2050    EDB0                LD   IR       ; 数据块自动传送
2052    C3AE00              JP   RESTRI   ; 传送完毕，回到 TPBUG
                                            控制，显示“—”标志
```

第二个例子，是用来对 EPROM 进行编程用的应用程序。可以将贮存在 RAM(本系统原有的或用户扩充的 RAM)，ROM，或者插在 PROM1 插座上的 EPROM 中的数据，复制到插在 PROM2 插座上的 EPROM 中去。

在每次使用本程序之前，先按下 RESET 键，然后修正 2FC0，2FC1，2FC2，2FC3 等单元和 HL 寄存器对的内容。

2FC0——源数据地址的高字节；

2FC1——源数据地址的低字节；

2FC2——目的地数据的地址高字节；

2FC3——目的地数据的地址低字节；

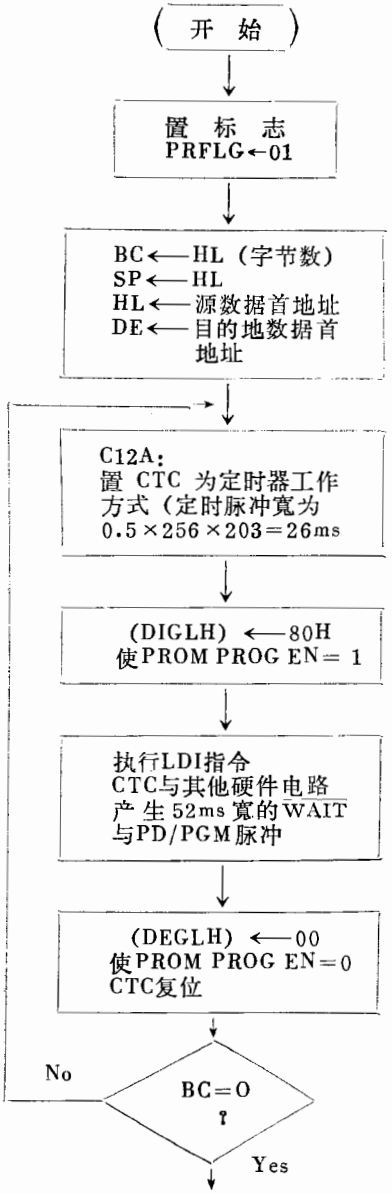
HL——用十六进制表示的要复制的总字节数，高字节在 H。

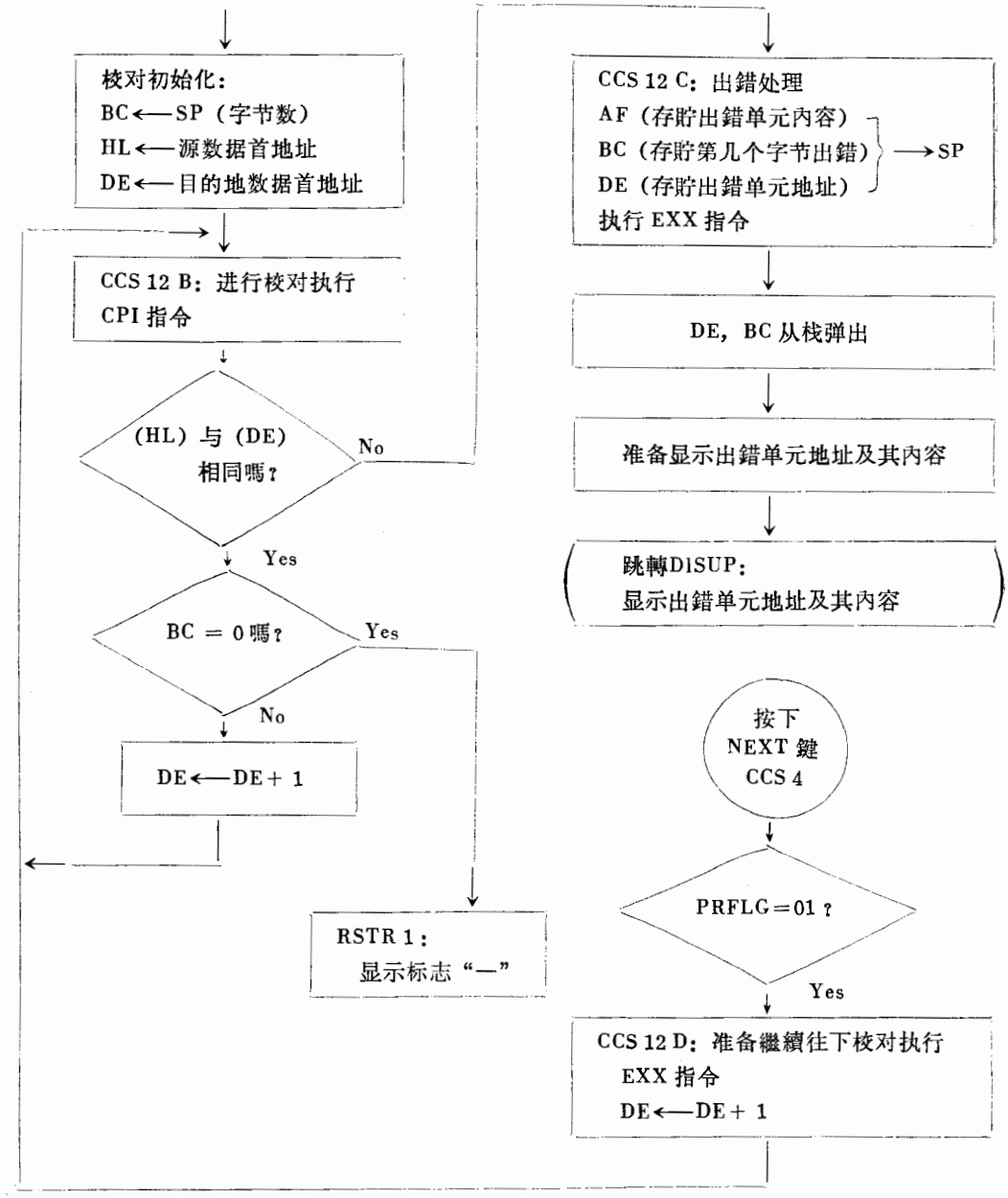
然后执行下列程序，将数据写入 EPROM。

；程序

```
                        ORG   2000
2002   3E  01    CI2:  LD    A, 01
2002   32  DA2F        LD    (PRFLG),A; 置标志
2005   E5            PUSH  HL
2006   C1            POP   BC          ; 将字节数送入 BC
2007   E5            PUSH  HL          ; 保存字节数目,预备校对用
2008   3A  CO2F      LD    A, (2FC0H)
200B   67            LD    H, A
200C   3A  C12F      LD    A, (2FC1H)
200F   6F            LD    L, A        ; 将源数据送入 HL
2010   3A  C22F      LD    A, (2FC2H)
2013   57            LD    D, A
2014   3A  C32F      LD    A, (2FC3H)
2017   5F            LD    E, A        ; 将目的地数据送入 DE
2018   3E  25    C12A: LD    A, 25H
201A   D3  86        OUT   (86H), A ; 输入控制字, 设置 CTC
                                      为定时器工作
201C   3E  CB        LD    A, 0CBH
201E   D3  86        OUT   (86H), A  ; 时间常数, 约 26ms
2020   3E  80        LD    A, 80H
2022   D3  8C        OUT   (DIGLH),A; 发出 PROM PROG EN = 1
2024   ED  AO        LDI             ; 插入等待讯号 WAIT
2026   3E  00        LD    A, 00
2028   D3  8C        OUT   (DIGLH),A; 使 PROM PROG EN = 0
202A   3E  03        LD    A, 03H
202C   3E  86        OUT   (86H), A ; CTC 复位
202E   EA  1820      JP    PE, C12A ; 若 BC - 1≒0 则返回
2031   C1            POP   BC          ; 弹出字节数, 开始进行校对
2032   3A  C02F      LD    A, (2FC0H)
2035   67            LD    H, A
2036   3A  C12F      LD    A, (2FC1H)
2039   6F            LD    L, A
203A   3A  C22F      LD    A, (2FC2H)
203D   57            LD    D, A
203E   3A  C32F      LD    A, (2FC3H)
2041   5F            LD    E, A
2042   C3  0406      JP    CCS12B    ; 转入 TPBUG
```

程 序 流 程 图


开 始
置 标 志
PRFLG←01
BC←HL（字节数）
SP←HL
HL←源数据首地址
DE←目的地数据首地址
C12A:
置 CTC 为定时器工作方式（定时脉冲宽为
0.5×256×203=26ms
(DIGLH) ←80H
使PROM PROG EN= 1
执行LDI指令
CTC与其他硬件电路
产生52ms宽的WAIT
与PD/PGM脉冲
(DEGLH) ←00
使PROM PROG EN=0
CTC复位
BC=O
?
No
Yes


校对初始化：
BC←—SP（字节数）
HL←—源数据首地址
DE←—目的地数据首地址
CCS 12 B：进行校对执行
CPI 指令
(HL) 与 (DE)
相同嗎？
No
Yes
BC = 0嗎？
Yes
No
DE←—DE + 1
RSTR 1：
显示标志“—”
CCS 12 C：出錯处理
AF（存貯出錯单元內容）
BC（存貯第几个字节出錯）
DE（存貯出錯单元地址）
→SP
执行 EXX 指令
DE，BC 从栈弹出
准备显示出錯单元地址及其內容
跳轉DISUP：
显示出錯单元地址及其內容
按下
NEXT 鍵
CCS 4
PRFLG＝01？
Yes
CCS 12 D：准备繼續往下校对执行
EXX 指令
DE←—DE + 1

4.7 求十进制数的算术和

下列程序是将两个5位十进制数(或和数不大于6位的6位数)进行算术加法运算，两数长度是相同的，和数贮存在指定存储单元，并将和数在显示器上显示出来。

；数据

2030——贮存字节数（两位十进制数为1字节，5位十进制数为3字节）；

2031、2032、2033——贮存被加数，按顺序先贮存低字节数；

2041、2042、2043——贮存加数，按顺序先贮存低字节数；

2051、2052、2053——贮存和数，按顺序先贮存低字节数。

；例子 895867 + 91787 = 987654

字节数：(2030) = 03

被加数：(2031) = 67　　加　数：(2041) = 87　　和　数：(2051) =

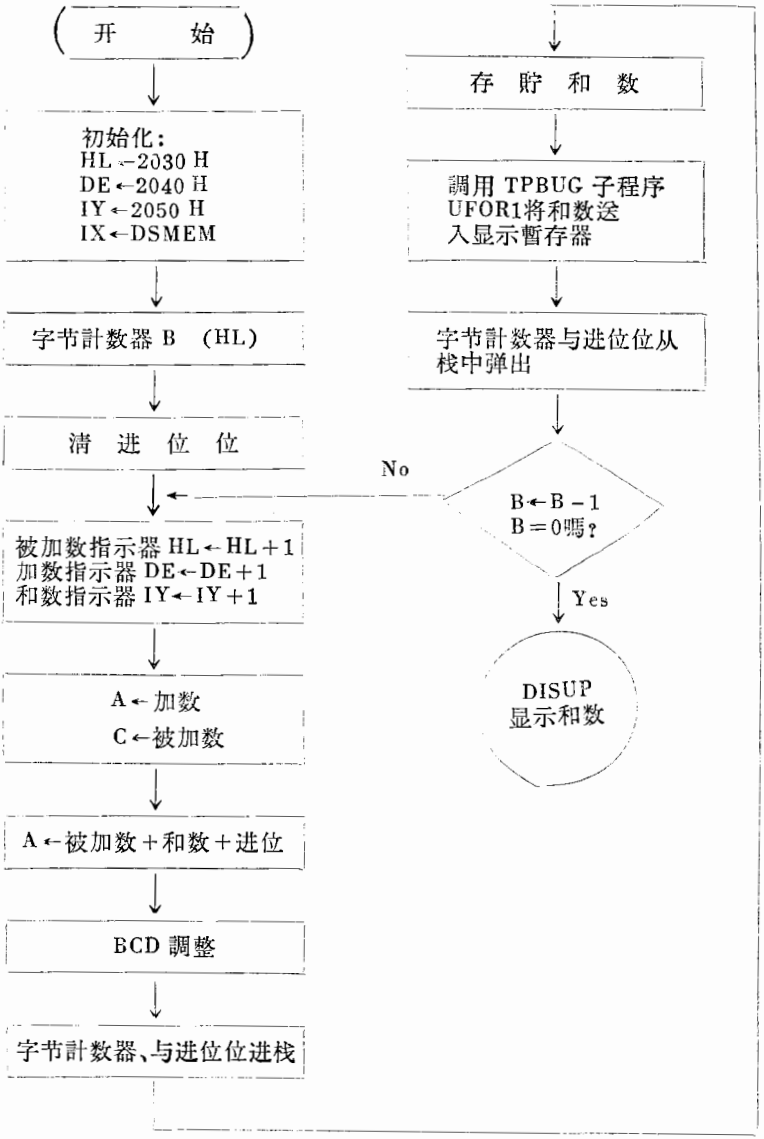
(2032) = 58　　(2042) = 17　　(2052) =

(2033) = 89　　(2043) = 09　　(2053) =

；程序

```
2000   21  30  20        LD    HL, 2030H    ；初始化
2003   11  40  20        LD    DE, 2040H
2006   FD  21  50  20    LD    IY, 2050H
200A   DD  21  FB  2F    LD    IX, (DSMEM4)
200E   46                LD    B, (HL)      ；计数器←字节数
200F   AF                XOR   A            ；清进位位
2010   23          LOOP: INC   HL           ；指向被加数存储单元地址
2011   13                INC   DE           ；指向加数存储单元地址
2012   FD  23            INC   IY           ；指向和数存储单元地址
2014   1A                LD    A, (DE)      ；取出加数
2015   4E                LD    C, (HL)      ；取出被加数
2016   89                ADC   A, C          ；带进位加
2017   27                DAA                ；BCD 调整
2018   FD  7700          LD    (IY+0), A    ；贮存和数
201B   C5                PUSH  BC           ；记忆计数器内字节数
201C   F5                PUSH  AF           ；记忆进位位
201D   CD  3C06          CALL  UFOR1        ；显示初始化
2020   DD2B              DEC   IX
2022   DD2B              DEC   IX
2024   F1                POP   AF
2025   C1                POP   BC
2026   10  E8            DJ    NZ, LOOP—$
2028   C3  F400          JP    DISUP        ；显示和数
```

程 序 流 程 图


开 始
初始化：
HL←2030 H
DE←2040 H
IY←2050 H
IX←DSMEM
字节計数器 B (HL)
清 进 位 位
被加数指示器 HL←HL+1
加数指示器 DE←DE+1
和数指示器 IY←IY+1
A←加数
C←被加数
A←被加数+和数+进位
BCD 調整
字节計数器、与进位位进栈
存 貯 和 数
調用 TPBUG 子程序
UFOR1将和数送
入显示暫存器
字节計数器与进位位从
栈中弹出
B←B-1
B=0嗎?
No
Yes
DISUP
显示和数

附　录　一

TP801 原理图、安装位置图、零件表

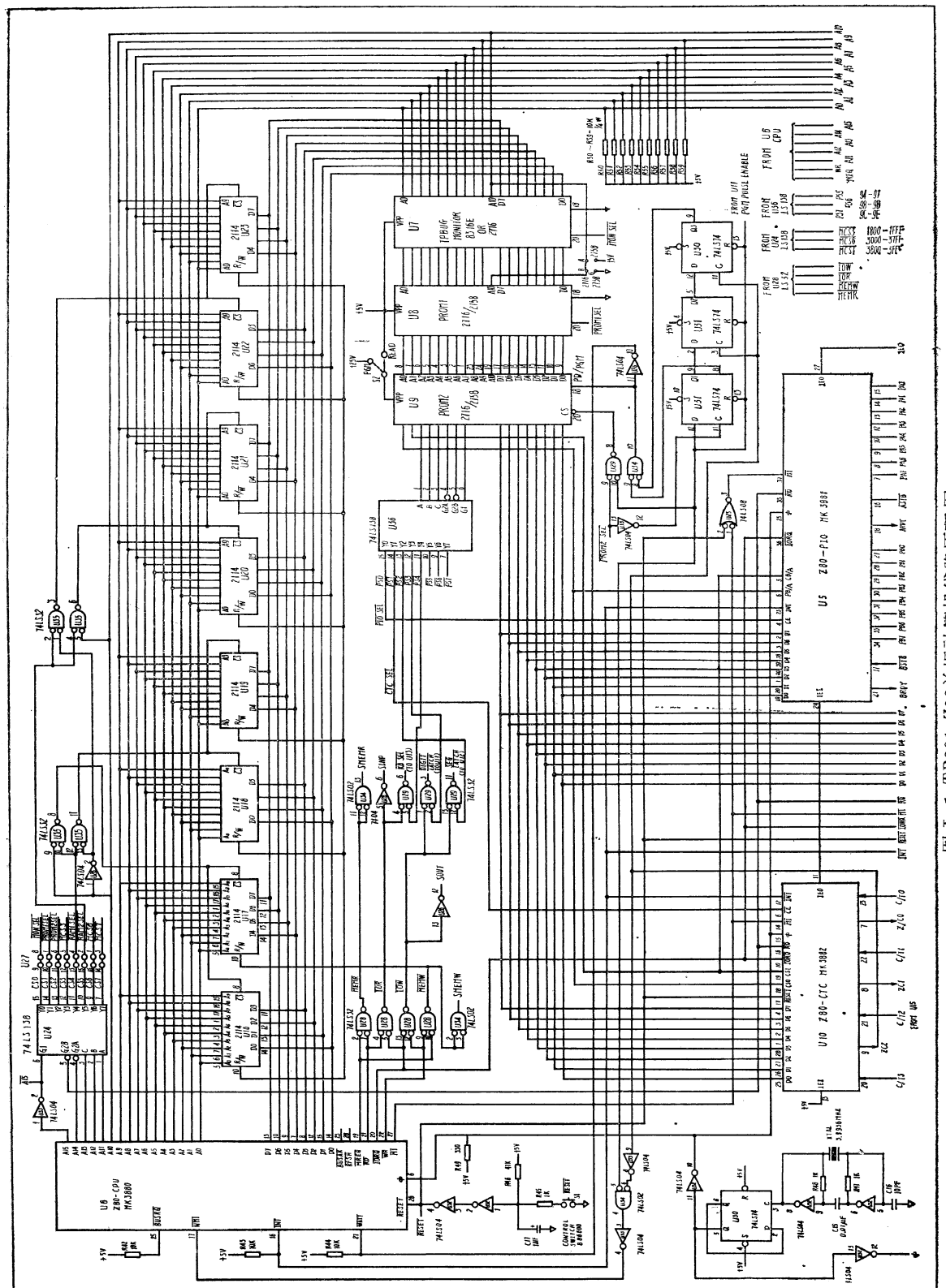


图 I.1 TP801-Z80单板計算机綫路原理图 1

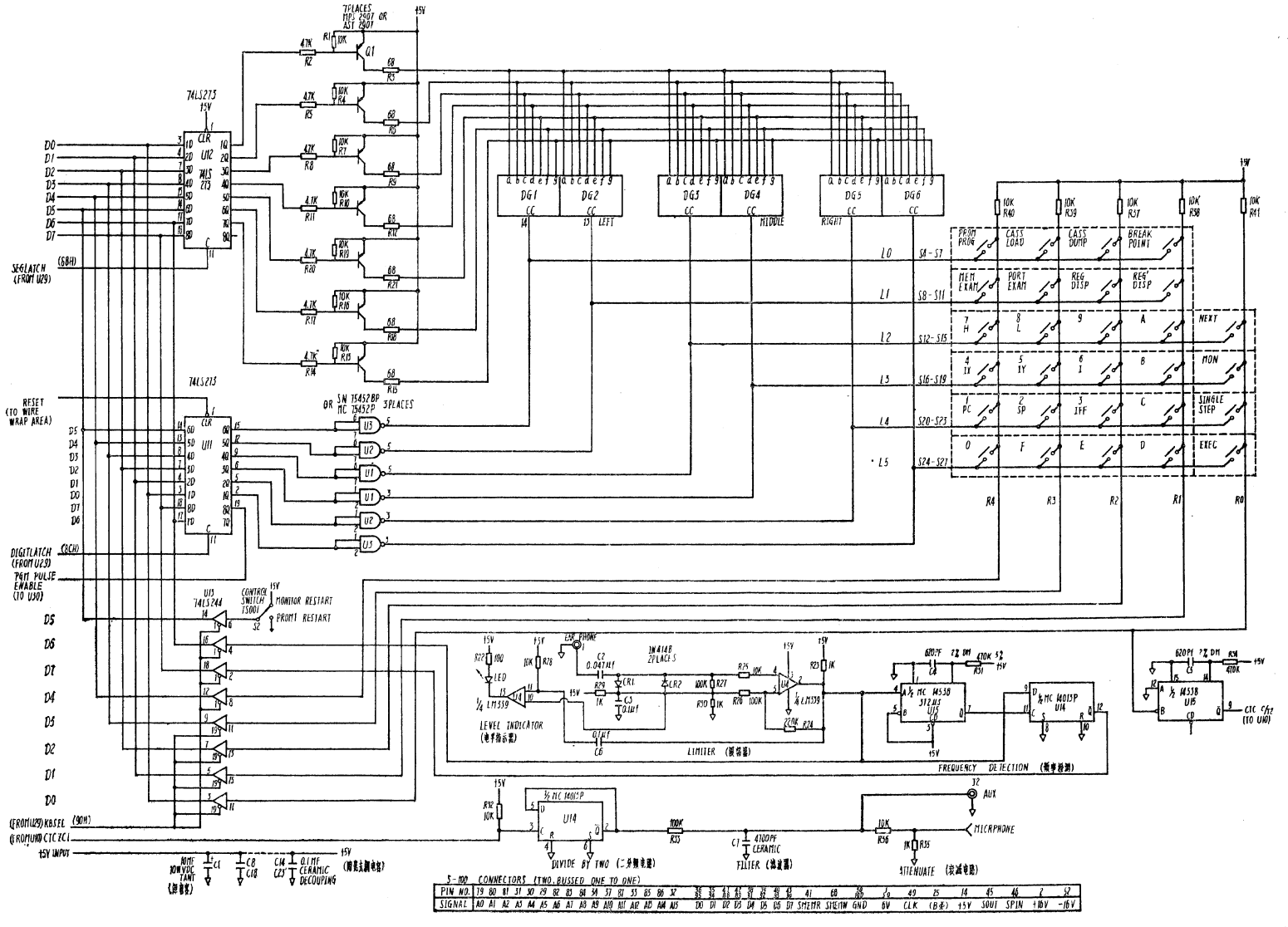
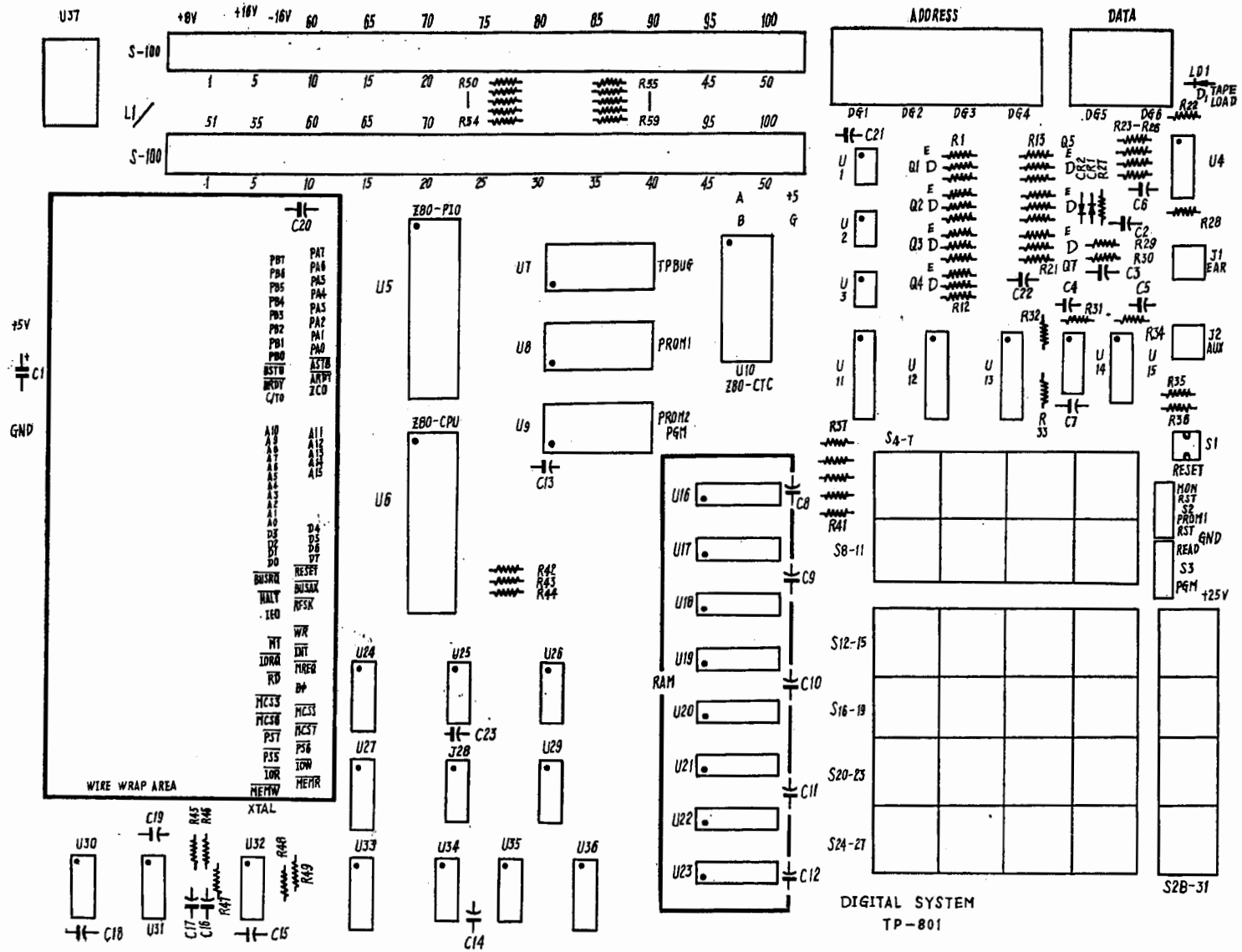


图 I.2 TP801-Z80 单板計算机綫路原理图 2



图Ⅰ.3　TP 801-Z 80 单板計算机安装位置图

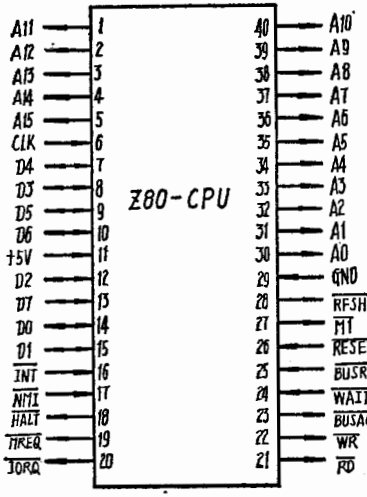
TP 801—Z80 单 板 计 算 机 零 件 表

序号	型号与名称（英文）	中文名称	数量	元器件代号
1	Z—80 CPU	中央处理器	1	U6
2	Z—80 PIO	并行輸入輸出电路	1	U5
3	Z—80 CTC	計数器定时器电路	1	U10
4	2114 RAM	450ns 讀写存儲器	4/8	U16 U19/U20 U23
5	ROM—TPBUG	只讀存儲器 TPBUG	1	U7
6	74 LS 138 Decoder	譯碼器	2	U24、U36
7	74 LS 32 Quad OR Gate	四或門	3	U28、U29、U35
8	74 LS 04 Hex Inverting Buffer	六反相器	2	U26、U33
9	7404 Hex Inverting Buffer	六反相器	1	U32
10	74 LS 08 Quad And Gate	四与門	1	U25
11	74 LS 244 Octal Buffer	8 位緩存器	1	U23
12	74 LS 273 Octal Latch	8 位 D 鎖存器	2	U11、U12
13	74 LS 74 Dual Latch	双 D 鎖存器	2	U30、U31
14	74 LS 02 Quad NOR Gate	四或非門	1	U34
15	75452P Peripheral Driver	驅动器	3	U1、U2、U3
16	MC14538BCP CMOS Monostable	CMOS 双单稳态触发器	1	U15
17	MC14013BCP CMOS Latch	CMOS 双 D 鎖存器	1	U14
18	LM339 Quad Comparitor	四比較器	1	U4
19	Sigle Digit Display	数碼显示器	6	DG1 DG6
20	9012 PNP Transistor	PNP 三极管	7	Q1 Q7
21	1N4148 Diode	二极管	2	CR1～CR2
22	3.9936 MHZ Parallel Resonant Crgsatl	并联諧振晶体	1	XTAL
23	Red LED Indicater	紅发光二极管	1	LD1
24	4.7K Ohm Resistor $^1/_4$W 5%	47KΩ 电阻色标：黃、紫、紅	7	R2、R5、R8、R11、R14、R17、R20、R32、R36
25	10K Ohm Resistor $^1/_4$W 5%	10KΩ 电阻色标：棕、黑、橙	29	R1、R4、R7、R10、R13、R16、R19、R25、R28、R37 R44、R50 R59
26	68 Ohm Resistor $^1/_4$W 5%	68Ω 电阻 色标：兰、灰、黑	7	R3、R6、R9、R12、R15、R18、R21
27	1K Ohm Resistor $^1/_4$W 5%	1K 电阻 色标：棕、黑、紅	7	R23、R29、R30、R35、R45、R47、R48

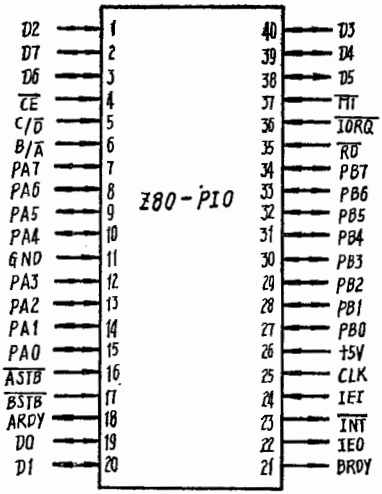
28	470K Ohm Resistor $^1/_4$W 5%	470K电阻 色标:黄、紫、黄	2	R31、R34
29	220K Ohm Resistor $^1/_4$W 5%	220K电阻 色标:紅、紅、黄	1	R24
30	100K Ohm Resistor $^1/_4$W 5%	100K电阻 色标:棕、黑、黄	3	R26、R27、R33
31	300 Ohm Resistor $^1/_4$W 5%	330Ω电阻 色标:橙、橙、黑	1	R49
32	47K Ohm Resistor $^1/_4$W 5%	47K电阻 色标:黄、紫、橙	1	R46
33	100 Ohm Resistor $^1/_4$W 5%	100Ω电阻 色标:棕、黑、棕	1	R22
34	0.1μF Capacitor 20%	0.1 μF 电容	15	C3、C6、C8 C14,18～C23
35	1μF Capacitor 20%	1 μF 鉭电容	1	C17
36	10μF Capacitor 20%	10 μF 鉭电容	1	C1
37	0.0047μF Capacitor 20%	0.047μF 电容	1	C7
38	10PF Capacitor 20%	10PF 电容	1	C16
39	0.01μF Capacitor 20%	0.01μF 电容	1	C15
40	620PF Capacitor 5%	620PF 电容	2	C4、C5
41	0.047μF Capacitor 20%	0.047μF 电容	1	C2
42	Push Button Control Switch	按鈕	1	S1
43	Slide Switch	控制开关	2	S2、S3
44	Key Switch	鍵开关	28	S4 S31
45	Key Top with Transparent Cap	带透明罩的鍵帽	28	
46	Audio Jack	录音机插孔	2	J1、J2
47	8Pin IC Socket	8 脚插座	3	U1、U2、U3
48	14Pin IC Socket	14脚插座	12	U4、U14、U25、U26、U28、U29 U35
49	16Pin IC Socket	16脚插座	3	U15、U24、U36
50	18Pin IC Socket	18脚插座	4	U16 U19
51	20Pin IC Socket	20脚插座	3	U11、U12、U13
52	24Pin IC Socket	24脚插座	3	U7、U8、U9
53	28Pin IC Socket	28脚插座	1	U10
54	40Pin IC Socket	40脚插座	2	U5、U6
55	Printed Circuit Board 354×264mm	印刷电路板	1	
56	Plastic Support 15mm	塑料支柱	14	
57	Steel Screw M3×12 S/T R/H	M3×12螺釘	14	

附 录 二

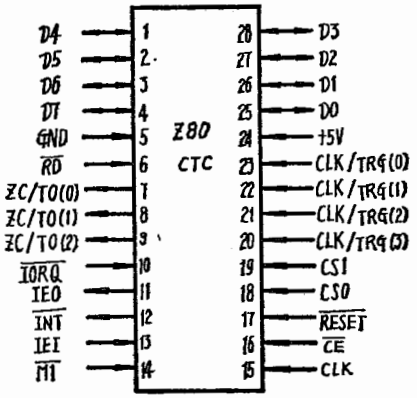
TP801 所用集成电路引脚图



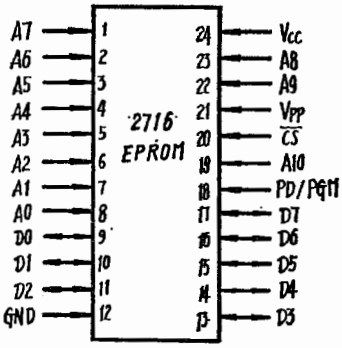
图Ⅱ.1 Z80—CPU 引脚图



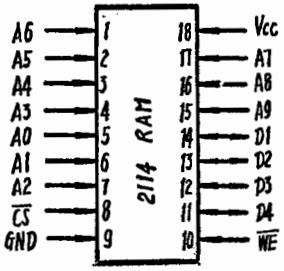
图Ⅱ.2 Z80—PIO 引脚图



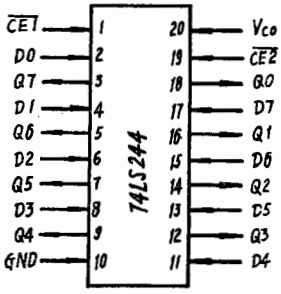
图Ⅱ.3 Z80—CTC 引脚图



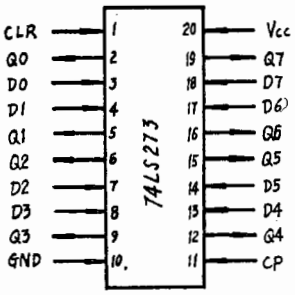
图Ⅱ.4 2716 EPROM 引脚图



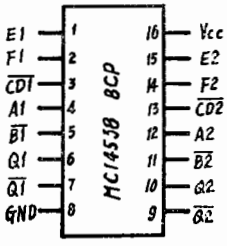
图Ⅱ.5 2114 RAM 引脚图

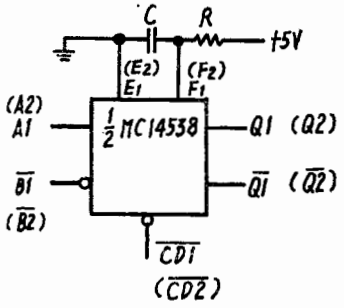


图Ⅱ.6　74LS244 八緩存器引脚图

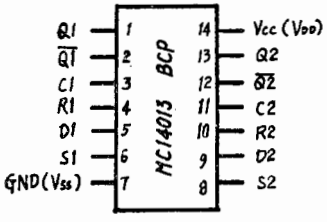


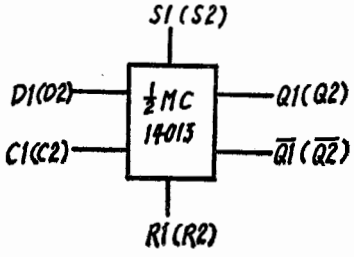
图Ⅱ.7　74LS273八D鎖存器引脚图



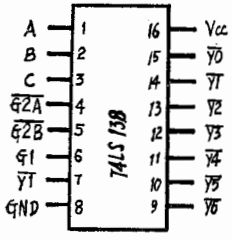


图Ⅱ.8　CMOS—MC14538BCP 双单稳态触发器引脚图及框图



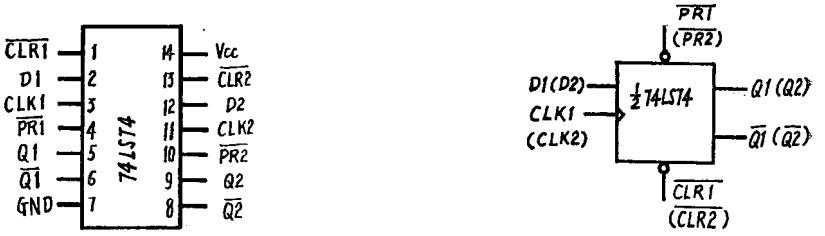


图Ⅱ.9　CMOS—MC14013BCP 双D鎖存器引脚图及框图



G1	G2A	G2B	A	B	C	輸　出
1	0	0	0	0	0	$\overline{Y0}=0$ 其余为 1
1	0	0	0	0	1	$\overline{Y1}=0$ 其余为 1
1	0	0	0	1	0	$\overline{Y2}=0$ 其余为 1
1	0	0	0	1	1	$\overline{Y3}=0$ 其余为 1
1	0	0	1	0	0	$\overline{Y4}=0$ 其余为 1
1	0	0	1	0	1	$\overline{Y5}=0$ 其余为 1
1	0	0	1	1	0	$\overline{Y6}=0$ 其余为 1
1	0	0	1	1	1	$\overline{Y7}=0$ 其余为 1
不是上述情况			×	×	×	全部輸出为 1

图Ⅱ.10　74LS138八中取一譯碼器引脚图及眞值表



图Ⅱ.11　74LS74 双D鎖存器的引脚图及框图

图Ⅱ.12　LM 339 四比較器的引脚图

图Ⅱ.13　74LS04 六反相器的引脚图

图Ⅱ.14　74LS08 四与門的引脚图

图Ⅱ.15　74LS32四或門的引脚图

图Ⅱ.16　74LS02 四或非門的引脚图

图Ⅱ.17　75452P 双驅动器的引脚图。

附 录 三

TPBUG 监 控 程 序 表

LOAD MAP(存储分配)

			起始地址		结束地址	
DK1:	RESTR.	OBJ[1]	REL	BEG ADDR 0000	END ADDR	00F3
DK1:	DISUP.	OBJ[1]	REL	BEG ADDR 00F4	END ADDR	01 22
DK1:	DECKY.	OBJ[1]	REL	BEG ADDR 0123	END ADDR	0633
DK1:	UTIL.	OBJ[1]	REL	BEG ADDR 0634	END ADDR	07F4
DK1:	UTILR.	OBJ[1]	ABS	BEG ADDR 07F8	END ADDR	07FF

GLOBAL CROSS REFERENCE TABLE (通用标号索引表)

SYMBOL (标号)	ADDR (地址)	REFERENCES (使用索引)
ARFLG	2FE0	06A4 03C5 02F0
BFLG	2FF4	0681 04BF 04BB 049E 0275 00A9
BPTAB	2FE4	067E 0250
CTC1P	07FA	04D5
CTC3L	07FE	0759
D20MS	064F	0172 0135
DECKY	0123	0121
DIG2	2FF5	068F 036C 01A6
DIG4	2FF6	0692 0498 0396 0235 01AC
DISMEM	2FF7	06AA 0687 065B 0617 05BB 0470 0443 03B6 02F4 00F5 00E4 00B4 0080
DISUP	00F4	062E 05CB 04C8 04B3 03B2 0393 0369 02EB 02D6 01BC 01A3 014D 0132 00F2 00D0 009D
DSMEM1	2FF8	017F 00B9
DSMEM2	2FF9	061F 05C3 0420 0378 0340 0334 00EB 00BC

```
DSMEM3   2FFA   037B 0187 00C3
DSMEM4   2FFB   0627 03E2 03A4 0385 0357 034A 0327 0315 0301 02E5 00C6
DSMEM5   2FFC   043A 00C9
DSMEM6   2FFD   0363
DSMEM7   2FFE   0434 0360 0192
FLG24    2FD9   0737 072D 0724 0714 0707 04D0
INCHR    0758   0591
INCHR4   079D   07FE
KEYPTR   2FDB   068A 03AF 0390 0366 0351 032B 01B9 01B5 01A0 019C 0195 018A 017A
KYTBL    07B9   0160
MFLG     2FE1   069E 03CC 039C 02A5
NMIS     01CB   0067
OTCHR    06F4
OTCHR1   06F7   050C
OTCHR6   0732   07FA
PFLG     2FDE   069B 03B9 0373 02AC
PRFLG    2FDA   0698 05D6 02B3
PUNHEH   2FC2   04EE
PUNHEL   2FC3   04F2
PUNHSH   2FC0   0552 0529 04E6
PUNHSL   2FC1   0556 0530 04EA
REGTB    07D5   047A
REGTBP   07E5   045D
REMBP2   007E   0208
RESTAR   009F
RESTR1   00AE   060A 05D1 057F 04B0 0482 0465 043F 03D0 02B8 02A2
```

RFLG 2FDF 06A1 03BF 030B
RST16 2FC4 0011
RST24 2FC7 0019
RST32 2FCA 0021
RST40 2FCD 0029
RST48 2FD0 0031
RST56 2FD3 0039
SEGPT 07A6 0102
SSFLG 2FF3 0695 027A 0202 01F9
STKPT 2FE2 0485 0468 0346 023F 01F6 01E6 01CD 00A1 0083 004B 0043 000A
STKPT1 2FE3 033C
UARIN 06B3
UBASC 06BB
UFGCR 0686 00AF
UFOR1 063C 062B 0623 061B 05C7 05BF 0423 03E5 03A7 0388 035D 034D 0343 0337 032E
0318 0304 02E8 02E1 02D3 02CB 02C3 00EF 00E7 009A 0091 0088
UFOR2 0659 05D9 04A3 03E9 03D3 039F 037E 02D9 02BE 023B
UFOR3 067C 04A6 024B 0205 0062
UFOR4 06A7 04CB
UIF 2FDD 0437 035A 0295 01F2 00AC 005F
UIX3 0634 04B6 0260 007A
ULACC 06DB 05B3 05AC 05A5 05A1 059D 0599
UUPACC 06C4
UPACCS 06D5 056B 0566 055F 054E 0549 0544 053C 0538 0534 052D 0526

GLOBAL SYMBOL TABLE (通用标号表)

ARFLG	2FE0	BFLG	2FF4	BPTAB	2FE4	CTC1P	07FA
CTC3L	07FE	D20MS	064F	DECKY	0123	DIG2	2FF5
DIG4	2FF6	DISMEM	2FF7	DISUP	00F4	DSMEM1	2FF8
DSMEM2	2FF9	DSMEM3	2FFA	DSMEM4	2FFB	DSMEM5	2FFC
DSMEM6	2FFD	DSMEM7	2FFE	FLG24	2FD9	INCHR	0758
INCHR4	079D	KEYPTR	2FDB	KYTBL	07B9	MFLG	2FE1
NMIS	01CB	OTCHR	06E4	OTCHR1	06F7	OTCHR6	0732
PFLG	2FDE	PRFLG	2FDA	PUNHEH	2FC2	PUNHEL	2FC3
PUNHSH	2FC0	PUNHSL	2FC1	REGTB	07D5	REGTBP	07E5
REMBP2	007E	RESTAR	009F	RESTR1	00AE	RFLG	2FDF
RST16	2FC4	RST24	2FC7	RST32	2FCA	RST40	2FCD
RST48	2FD0	RST56	2FD3	SEGPT	07A6	SSFLS	2FF3
STKPT	2FE2	STKPT1	2FE3	UABIN	06B3	UBASC	06BB
UFGCR	2FC4	UFOR1	063C	UFOR2	0659	UFOR3	067C
UFOR4	2FD0	UIF	2FDD	UIX3	0634	ULACC	06DB
UPACC	2FE2	UPACCS	06D5				

```
地址    目的码      语句序号      源语句
                    0002          程序名称    RESTR
                    0003 ; 重新启动(RST), 不可屏蔽中断(NMI)和移去断点(BP)程序
                    0004 ; 版本 1.7                5/18/78
                    0005          PSECT     REL
                    0006          GLOBAL    DSMEM4
                    0007          GLOBAL    DSMEM1
                    0008          GLOBAL    DISUP
                    0009          GLOBAL    UIF
                    0010          GLOBAL    UIX3
                    0011          GLOBAL    STKPT
                    0012          GLOBAL    RST16
                    0013          GLOBAL    RST24
                    0014          GLOBAL    RST32
                    0015          GLOBAL    RST40
                    0016          GLOBAL    RST48
                    0017          GLOBAL    RST56
                    0018          GLOBAL    DSMEM5
                    0019          GLOBAL    DSMEM2
                    0020          GLOBAL    DISMEM
                    0021          GLOBAL    DSMEM3
                    0022          GLOBAL    UFGCR
                    0023          GLOBAL    UFOR1
                    0024          GLOBAL    NMIS
                    0025          GLOBAL    UFOR3
                    0026          GLOBAL    RESTAR
                    0027          GLOBAL    REMBP2
```

```
地址    目的码        语句序号      源语句
                      0028          GLOBAL  RESTR 1
                      0029          GLOBAL  BFLG
>0000                 0030          ORG     0000 H
                      0031 ; ※※※※重新启动指令处理※※※※
                      0032 ; ※※※※复位入口点— RST0 ※※※※
                      0033 ; ※※※※RST0 用于复位，RST 8 用于断点，其它所有
                      0034 ;       的 RST 指令如 RST 16、RST 24、RST 32、RST
                      0035 ;       40 和 RST 56 都在 RAM 中分配存储单元，用
                      0036 ;       户可以在那里设置一条转移指令，转移到存在
                      0037 ;       RAM 中的服务程序。当进入断点后，用户程
                      0038 ;       序所有的寄存器都被推入堆栈，而后将所有的
                      0039 ;       断点从用户程序中移走
                      0040 ; ※※※※复位入口点—RST0
0000    31 C 02F      0041          LD      SP, 2 FC 0 H
0003    C 39 F 00     0042          JP      RESTAR      ; 转去完成初始化
                      0043 ; ※※※※断点入口—RST 8;
>0008                 0044          ORG     000 8H      ; 开始重新启动表
0008    ED73E22F      0045 BPENT:   LD      (STKPT),SP                ; 保存用户的栈指针 SP
000 C   F 5           0046          PUSH    AF          ; 用户寄存器进栈
000 D   C 5           0047          PUSH    BC
000 E   1803          0048          JR      BPENT2-$
                      0049 ; ※※※※用户使用 RST 16的入口
0010    C 3 C 42 F    0050          JP      RST 16      ; 转到 RAM
0013    ED 57         0051 BPENT2:  LD      A,I         ; 中断矢量寄存器 I 送累加器 A，中断允许触发器 IFF
                                                        送标志寄存器 F
```

```
地址    目的码      语句序号        源语句
0015    F3          0052            DI
0016    1803        0053            JR          BPFNT3-$
                    0054 ; ※※※※用户使用 RST 24 的入口
0018    C3C72F      0055            JP          RST24
001B    D5          0056 BPENT3:    PUSH        DE
001C    E5          0057            PUSH        HL
001D    F5          0058            PUSH        AF          ; 保护 I 和 IFF
001E    1803        0059            JR          BPENT4-$
                    0060 ; ※※※※用户使用 RST 32 的入口
0020    C3CA2F      0061            JP          RST32
0023    08          0062 BPENT4:    EX          AF,AF'      ; 取辅助寄存器的内容
0024    D9          0063            EXX
0025    F5          0064            PUSH        AF
0026    1803        0065            JR          BPENT5-$
                    0066 ; ※※※※用户使用 RST 40 的入口
0028    C3CD2F      0067            JP          RST40
002B    C5          0068 BPENT5:    PUSH        BC
002C    D5          0069            PUSH        DE
002D    E5          0070            PUSH        HL
002E    1803        0071            JR          BPENT6-$
                    0072 ; ※※※※用户使用 RST 48 的入口
0030    C3D02F      0073            JP          RST48
0033    DDE5        0074 BPENT6:    PUSH        IX
0035    00          0075            NOP                     ; 空一字节
0036    1803        0076            JR          BPENT7-$
```

```
地址     目的码        语句序号      源语句
                0077 ; ※※※※用户使用 RST 56 的入口
0038     C3D32F       0078          JP        RST56
003B     FDE5         0079 BPENT7:  PUSH      IY
003D     3E03         0080          LD        A,03H
003F     D386         0081          OUT       (CTC2),A          ; 使键盘不能通过 CTC2 请求中断
0041     DD2AE22F     0082          LD        IX,(STKPT)        ; 取用户栈指针 SP
0045     DD23         0083          INC       IX
0047     DD23         0084          INC       IX
0049     DD22E22F     0085          LD        (STKPT),IX
004D     DD7EFE       0086          LD        A,(IX-2)          ; 测试 PC 的低字节
0050     B7           0087          OR        A          ; 设置状态标志
0051     2003         0088          JR        NZ,BPENT8-$
0053     DD35FF       0089          DEC       (IX-1)            ; 高字节减 1
0056     DD35FE       0090 BPENT8:  DEC       (IX-2)            ; 低字节减 1
0059     DD7EF4       0091          LD        A,(IX-12          ; 在堆栈中的 IFF送 A
005C     E604         0092          AND       04H               ; 保留 F 中的 IFF, 屏蔽掉其它位
005E     32DD2F       0093          LD        (UIF),A           ; 保存供后面使用
0061     CD7C06       0094          CALL      UFOR3     ; 取得断点表地址, 断点数目存入 B 寄存器
0064     1803         0095          JR        BPNT8A-$
0066     C3CB01       0096          JP        NMIS              ; 不可屏蔽中断入口
0069     2813         0097 BPNT8A:  JR        Z,REMBP2-$        ; 若无断点, 转去显示 PC 和 A 寄存
                                                                 器内容
                0098 ; ※※※※移去断点※※※※
                0099 ; 当进入监控程序 TPBUG 时移去断点, 在执行或继续进行用户程序命令时, 恢复断点
                0100
```

```
地址    目的码      语句序号    源语句
006B    DD7E02      0101 REMBP: LD      A,(IX+2)          ; 取要恢复的操作码
                    0102 ; 校验是否设置了多断点
006E    FECF        0103        CP      0CFH
0070    2807        0104        JR      Z,REMBP1-$        ; 如多断点则转移
0072    DD6E01      0105 REMPP0:LD      L,(IX+1)
0075    DD6600      0106        LD      H,(IX+0)          ; 取断点地址
0078    77          0107        LD      (HL),A    ; 恢复操作码
0079    CD3406      0108 REMBP1:CALL    UIX3      ; 取下一个断点地址，断点数目计数器 B 减 1
007C    20ED        0109        JR      NZ,REMBP-$        ; 再次返回
                    0110 ; 显示 PC 和A
007E    DD21F72F    0111 REMBP2:LD      IX,DISMEM         ; 指向显示缓冲区首地址 DISMEM
0082    2AE22F      0112        LD      HL,(STKPT)        ; 栈顶置入 HL
0085    2B          0113        DEC     HL
0086    7E          0114        LD      A,(HL)    ; 取 PC 的高字节
0087    CD3C06      0115        CALL    UFOR1     ; 写入显示缓冲区
008A    DD23        0116        INC     IX
008C    DD23        0117        INC     IX
008E    2B          0118        DEC     HL
008F    7E          0119        LD      A,(HL)    ; 取 PC 的低字节
0090    CD3C06      0120        CALL    UFOR1     ; 写入显示缓冲区
0093    DD23        0121        INC     IX
0095    DD23        0122        INC     IX
0097    2B          0123        DEC     HL
0098    7E          0124        LD      A,(HL)    ; 取栈存的 A
0099    CD3C06      0125        CALL    UFOR1     ; 写入显示缓冲区
```

```
地址    目的码      语句序号    源语句
009C    C3F400      0126        JP      DISUP       ; 转显示程序
                    0127 ; ※※※※※※※※※ 完成重新启动的程序 ※※※※※※※※※※※※※※
009F    ED73E22F    0128 RESTAR: LD     (STKPT),SP              ; 保护用户的 SP
00A3    31A82F      0129        LD      SP,2FA8H                ; 对监控程序的 SP 置初值
00A6    3E00        0130        LD      A,00H
00A8    32F42F      0131        LD      (BFLG),A                ; 清断点标志 BFLG
00AB    32DD2F      0132        LD      (UIF),A                 ; 清用户 IFF2 标志
00AE    CD8606      0133 RESTAR1:CALL   UFGCR       ; 清各个标志
00B1    3E11        0134        LD      A,11H       ; 送提示符 "—"
00B3    32F72F      0135        LD      (DISMEM),A
00B6    3E10        0136        LD      A,10H       ; 送空白符
00B8    32F82F      0137        LD      (DSMEM1),A
00BB    32F92F      0138        LD      (DSMEM2),A
00BE    1802        0139        JR      RESTR2-$
00C0    1813        0140        JR      RESTR3-$                ; 相对偏移量 (OFFSET) 计算程序的
                                                                    入口
00C2    32FA2F      0141 RESTR2: LD     (DSMEM3),A
00C5    32FB2F      0142        LD      (DSMEM4),A
00C8    32FC2F      0143        LD      (DSMEM5),A
00CB    DB90        0144        IN      A,(KBSEL)               ; 检测开关 S2 的位置
00CD    CB6F        0145        BIT     5,A         ; 测试
00CF    C2F400      0146        JP      NZ,DISUP                ; 转显示程序
00D2    C30008      0147        JP      0800H       ; 转向 PROM1 的起始地址
                    0148 ; 相对偏移计算子程序—自动地计算相对偏移
                    0149 ; 量并将计算结果放入存储器并显示
```

```
地址    目的码        语句序号      源语句
                0150 ; HL 放相对转移的目的地的操作码地址
                0151 ; DE 放带有相对偏移量的相对转移指令所在的地址
00D5    13      0152 RESTR3: INC    DE          ; 指向偏移量
00D6    D5      0153         PUSH   DE
00D7    DDE1    0154         POP    IX          ; 保护偏移量的地址
00D9    13      0155         INC    DE          ; 指向下一条指令的操作码
00DA    7D      0156         LD     A,L         ; 取低字节
00DB    93      0157         SUB    E           ; 减
00DC    6F      0158         LD     L,A         ; 保存偏移量
00DD    DD7700  0159         LD     (IX+0),A                ; 将偏移量填入需要的存储单元
00E0    7C      0160         LD     A,H         ; 取高字节
00E1    9A      0161         SBC    A,D         ; 减
00E2    DD21F72F 0162        LD     IX,DISMEM
00E6    CD3C06  0163         CALL   UFOR1       ; 将高字节减法结果写入显示缓冲区，如果不是 FF 或
                                                   00，则为超越正常范围
00E9    DD21F92F 0164        LD     IX,DSMEM2
00ED    7D      0165         LD     A,L
00EE    CD3C06  0166         CALL   UFOR1       ; 写偏移量到显示缓冲区
00F1    C3F400  0167         JP     DISUP       ; 转显示程序
>0086           0168 CTC2:   EQU    86H         ; 单步/EPROM 编程
>0090           0169 KBSEL:  EQU    90H         ; MONITOR/PROM1 开关指向检测
                0170         END
```

```
地址    目的码      语句序号      源语句
                    0002          程序名称    DISUP
                    0003 ; 更新显示程序
                    0004 ; 版本0.3  3/13/78
                    0005          PSECT       PEL
                    0006          GLOBAL      DISMEM
                    0007          GLOBAL      SEGPT
                    0008          GLOBAL      DECKY
                    0009          GLOBAL      DISUP
                    0010          GLOBAL      DISUP
>00F4               0011          ORG         00F4
                    0012 ; ※※※※ 更新显示程序 ※※※※
                    0013 ; ※※ 功能：从显示缓冲区移数据到数码显示
                    0014 ; ※※ 入口：被显示的数据在六个存储单元
                    0015 ;          内，起始单元为 DISMEM
                    0016 ; ※※ 出口：刷新所有的六位数据一使用的
                    0017 ;          寄存器是：IX,AB,D,E,H,L,
                    0018 ; ※※ 寄存器的使用：HL 是指向 DISMEM 中现行单元的指示器
                    0019 ;                   IX 是指向字形表的指示器
                    0020 ;                   B 是现行数位指示器
                    0021 ;                   E 保持被显示数据
                    0022 ;                   A 和 D 是暂存寄存器
00F4    21F72F      0023 DISUP:  LD          HL,DISMEM ; 指向显示缓冲区的首地址 DISMEM
00F7    0620        0024         LD          B,020H    ; 指向左边数位
00F9    5E          0025 DISUP1: LD          E,(HL)    ; 取第一个字节
00FA    1600        0026         LD          D,00H     ; D 寄存器清零
```

```
地址   目的码      语句序号        源语句
00FC   3E00        0027            LD      A,00H
00FE   D38C        0028            OUT     (DIGLH),A  ; 关显示
0100   DD21A607    0029            LD      IX,SEGPT   ; 变址寄存器指向字形表
0104   DD19        0030            ADD     IX,DE      ; 为获得字形变址作加法
0106   DD7E00      0031            LD      A,(IX+0)   ; 得到字形
0109   D388        0032            OUT     (SEGLH),A  ; 输出字形到字形选择锁存器
010B   78          0033            LD      A,B
010C   D38C        0034            OUT     (DIGLH),A  ; 输出到数位选择锁存器
010E   1E2D        0035            LD      E,2DH      ; 设置 1 ms 延时
0110   1D          0036 DISUP2:    DEC     E
0111   3E00        0037            LD      A,00H      ; 延时循环
0113   BB          0038            CP      E
0114   20FA        0039            JR      NZ,DISUP2-$
0116   3E01        0040            LD      A,01H
0118   B8          0041            CP      B          ; B=1?
0119   2805        0042            JR      Z,DISUP3-$ ; 是，转出口
011B   23          0043            INC     HL         ; 否，指向下一位
011C   CB38        0044            SRL     B          ; 移位到下一位
011E   18D9        0045            JR      DISUP1-$
0120   C32301      0046 DISUP3:    JR      DECKY      ; 转键盘分析程序
>008C              0047 DIGLH:     EQU     8CH        ; 只写数位选择
>0088              0048 SEGLH:     EQU     88H        ; 只写字形选择
                   0049            END
```

```
地址    目的码      语句序号      源语句
                0002          ORG       0123 H
                0003  ; 键分析和执行
                0004  ; 版本  2.1  5/17/78
                0005          PSECT     REL
                0006          NAME      DECKY
                0007          GLOBAL    UFOR4
                0008          GLOBAL    UFOR1
                0009          GLOBAL    UFOR2
                0010          GLOBAL    UFOR3
                0011          GLOBAL    REMBP2
                0012          GLOBAL    D20MS
                0013          GLOBAL    KEYPTR
                0014          GLOBAL    STKPT
                0015          GLOBAL    STKPT1
                0016          GLOBAL    SSFLG
                0017          GLOBAL    UPACCS
                0018          GLOBAL    OTHCR1
                0019          GLOBAL    OTHCR6
                0020          GLOBAL    UFOR1
                0021          GLOBAL    BPTAB
                0022          GLOBAL    DIG4
                0023          GLOBAL    DIG2
                0024          GLOBAL    DIG8
                0025          GLOBAL    MFLG
                0026          GLOBAL    BFLG
```

地址	目的码	语句序号	源语句	
		0027	GLOBAL	RFLG
		0028	GLOBAL	ARFLG
		0029	GLOBAL	PFLG
		0030	GLOBAL	PRFLG
		0031	GLOBAL	INCHR
		0032	GLOBAL	PUNHSH
		0033	GLOBAL	PUNHSL
		0034	GLOBAL	PUNHEH
		0035	GLOBAL	PUNHEL
		0036	GLOBAL	REGTB
		0037	GLOBAL	RESTAR
		0038	GLOBAL	RESTR1
		0039	GLOBAL	DISUP
		0040	GLOBAL	REGTBP
		0041	GLOBAL	DISMEM
		0042	GLOBAL	RFLG
		0043	GLOBAL	UIF
		0044	GLOBAL	DSMEM1
		0045	GLOBAL	DSMEM2
		0046	GLOBAL	DSMEM3
		0047	GLOBAL	DSMEM4
		0048	GLOBAL	DSMEM5
		0049	GLOBAL	DSMEM6
		0050	GLOBAL	DSMEM7
		0051	GLOBAL	ULACC

```
地址    目的码      语句序号    源语句
                    0052        GLOBAL    FLG24
                    0053        GLOBAL    UIX3
                    0054        GLOBAL    KYTBL
                    0055        GLOBAL    CTC1P
                    0056        GLOBAL    NMIS
                    0057        GLOBAL    DECKY
                    0058 ;  功能：空白数码显示，扫描键矩阵有否键闭合，如
                    0059 ;        发现键闭合，就产生一个 20ms 的延时用以
                    0060 ;        消除键抖动。每次扫描键矩阵的一行，并检
                    0061 ;        测所有的列。如果闭合的是一个十六进制的
                    0062 ;        数字键，就将其放到存储器显示缓冲区的下
                    0063 ;        一个空白单元中（指针为 KEYPTR)。输入
                    0064 ;        两位数就设置标志（DIG2)，当已输了四位
                    0065 ;        数，就设置标志（DIG4)，当已输了八位
                    0066 ;        数，就调用存储器交换程序。如经分析识别
                    0067 ;        为一个命令键，相应的处理程序将从跳转表
                    0068 ;        （JPTAB）中找到。命令键的处理程序也包
                    0069 ;        括在本模块中。
                    0070 ;
                    0071 ;  入口：无什么要求，跟在更新显示程序后面执行。
                    0072 ;  出口：如果是命令键，在执行命令之后转出。如果是十六进制数字键，数据已写入显示缓冲
                    0073 ;        区（DLSMEM）之后转出。
                    0074 ;  使用的寄存器：
                    0075 ;        A —暂存键值
```

```
地址    目的码      语句序号      源语句
                    0076 ;        B —数位值
                    0077 ;        C —暂存由行和列数据确定的特征字
                    0078 ;        HL —键值查找表 (KYTBL) 指针
                    0079 ;        HL —显示缓冲区指针 (KEYPTR)
                    0080 ;        BC —暂存—多用于计算出要存放在 DISMEM 中的偏移量
                    0081 ;        EXX—在 EPROM 编程的校对过程中保存指针用。在显示了错误之后，使用NEXT 键
                    0082 ;             继续校对时，恢复指针用。
0123    3E7F        0083 DECKY:   LD      A, 7FH
0125    D388        0084          OUT     (SEGLH) , A                    ; 关显示
0127    3E3F        0085          LD      A, 3FH
0129    D38C        0086          OUT     (DIGLH) , A                    ; 输出，使所有的行为低
012B    DB90        0087          IN      A, (KBSEL                      ; 输入列数据
012D    E61F        0088          AND     1FH          ; 屏蔽无用的位 (D5, D6, D7)
012F    FE1F        0089          CP      1FH          ; 是否有键按下?
0131    CAF400      0090          JP      Z, DISUP     ; 否，返回显示
0134    CD4F06      0091          CALL    D20MS        ; 是为了消去键抖动而延时 20 ms
0137    0E8C        0092          LD      C, DIGLH
0139    0601        0093          LD      B, 01H
013B    ED41        0094 KEYDN1:  OUT     (C) ,B       ; 选中其中一行键
013D    DB90        0095          IN      A, (KBSEL)                     ; 输入列数据
013F    E61F        0096          AND     1FH          ; 屏蔽无用的位 (D5,D6,D7)
0141    FE1F        0097          CP      1FH          ; 是否有键按下?
0143    200A        0098          JR      NZ,KEYDN2—$                    ; 是，转键译码
0145    CB20        0099          SLA     B            ; 否，选择下一行
0147    3E40        0100          LD      A,40H
```

```
地址    目的码      语句序号    源语句
0149    B8          0101        CP      B           ；是否全扫描完?
014A    20EF        0102        JR      NZ,KEYDN1—$                 ；否，返回循环
014C    C3F400      0103        JP      DISUP       ；是，转显示
                    0104 ；※※入口：进入键分析时行存于 B 寄
                    0105 ；          存器列（已用1FH屏蔽过）存于累加器中
014F    0E00        0106 KEYDN2: LD      C,00H
0151    0D          0107 KEYDN3: DEC     C
0152    CB38        0108        SRL     B
0154    20FB        0109        JR      NZ,KEYDN3—$                 ；当 B=0 时，才往下执行
0156    CB21        0110        SLA     C
0158    CB21        0111        SLA     C
015A    CB21        0112        SLA     C
015C    CB21        0113        SLA     C           ；得到字节高四位
015E    81          0114        ADD     A, C        ；与 A 相加
015F    21B907      0115        LD      HL, KYTBL   ；设置表的指针
0162    BE          0116 KEYDN4:CP      (HL)        ；在表中找到等于 A 的项
0163    2804        0117        JR      Z,KEYDN5—$；是，找到此项
0165    23          0118        INC     HL
0166    04          0119        INC     B           ；否，调整指针
0167    18F9        0120        JR      KEYDN4—$    ；循环返回
0169    DB90        0121 KEYDN5:IN      A,(KBSEL)   ；检测键释放
016B    E61F        0122        AND     01FH        ；屏蔽其它输入位
016D    FE1F        0123        CP      01FH
016F    20F8        0124        JR      NZ,KEYDN5—$                 ；循环直到键释放为止
```

```
地置   目的码     语句序号   源语句
0171   CD4F06     0125       CALL    D20MS       ; 为消去键抖动而延时 20 ms
0174   78         0126       LD      A、B        ; 取键值
0175   FE10       0127       CP      01H
0177   3045       0128       JR      NC,KEYDN6—$                ; 若是命令键，则转到分析
0179   2ADB2F     0129       LD      HL,(KEYPTR)                ; 若是十六进制数字键，则
                                                                 存入显示缓冲区
017C   70         0130       LD      (HL),B
017D   B7         0131       OR      A
017E   01F82F     0132       LD      BC,DSMEM1
0181   ED42       0133       SBC     HL,BC       ; 输入的是第 2 位数吗?
0183   2820       0134       JR      Z,KEYDNA—$                 ; 是， DIG 2 标志置 1
0185   B7         0135       OR      A           ; 清进位标志
0186   01FA2F     0136       LD      BC,DSMEM3                  ; 清断位标志
0189   2ADB2F     0137       LD      HL,(KEYPTR)
018C   ED42       0138       SBC     HL,BC       ; 输入的是第 4 位数吗?
018E   281B       0139       JR      Z,KEYDN8—$                 ; 是， DIG 4 标志置 1
0190   B7         0140       OR      A
0191   01FE2F     0141       LD      BC,DSMEM7
0194   2ADB2F     0142       LD      HL,(KEYPTR)                ; 十六进制数字键，存入
                                                                 显示缓冲区中
0197   ED42       0143       SBC     HL,BC       ; 显示缓冲器已写入 8 位数吗?
0199   2816       0144       JR      Z,KEYDNP—$                 ; 是，跳转
019B   2ADB2F     0145 KEYDN7:LD     HL,(KEYPTR)                ; 取键指针
019E   23         0146       INC     HL          ; 指针加 1
```

```
地址    目的码      语句序号      源语句
019F    22DB2F      0147          LD      (KEYPTR),HL                ；保护HL 内容
01A2    C3F400      0148          JP      DISUP       ；转出
01A5    21F52F      0149 KEYDNA:  LD      HL,DIG2
01A8    34          0150          INC     (HL)
01A9    18F0        0151          JR      KEYDN7—$
01AB    21F62F      0152 KEYDN8:  LD      HL,DIG4
01AE    34          0153          INC     (HL)
01AF    18EA        0154          JR      KEYDN7—$
01B1    CDB403      0155 KEYDN9:  CALL    ALTER       ；输入数据到存储器、输入输出口或寄存器
01B4    2ADB2F      0156          LD      HL,(KEYPTR)
01B7    2B          0157          DEC     HL          ；把指针向回调整到进入六位数时的位置
01B8    22DB2F      0158          LD      (KEYPTR),HL                ；存 HL
01BB    C3F400      0159          JP      DISUP
                    0160 ；找到命令键处理程序的地址，此时键值在 A 中
01BE    D610        0161 KEYDN6:  SUB     10H         ；第 16 个键的偏移量为 0，第 17 个键的偏移量为
                                                        3，先求差值
01C0    4F          0162          LD      C,A
01C1    81          0163          ADD     A,C         ；二倍差值
01C2    81          0164          ADD     A,C         ；三倍差值
01C3    4F          0165          LD      C,A
01C4    0600        0166          LD      B,00H
01C6    210C02      0167          LD      HL,JPTAB                   ；取表的首地址
01C9    09          0168          ADD     HL,BC       ；得到表的项号地址（首地址加偏移量）
01CA    E9          0169          JP      (HL)
                    0170 ；不可屏蔽中断服务程序（单步或中止程序时使用）
```

```
地址    目的码        语句序号      源语句
01CB    ED73E22F      0171  NMIS:   LD      (STKPT),SP   ; 保存用户的 SP
01CF    F5            0172          PUSH    AF
01D0    3E03          0173          LD      A,03H
01D2    D386          0174          OUT     (CTC2),A     ; 关闭 CTC
01D4    ED57          0175          LD      A,I          ; I 送 A，IFF 送 F
01D6    C5            0176          PUSH    BC
01D7    D5            0177          PUSH    DE           ; 保存各寄存器
01D8    E5            0178          PUSH    HL
01D9    F5            0179          PUSH    AF
01DA    08            0180          EX      AF,AF'
01DB    D9            0181          EXX
01DC    F5            0182          PUSH    AF
01DD    C5            0183          PUSH    BC
01DE    D5            0184          PUSH    DE
01DF    E5            0185          PUSH    HL
01E0    DDE5          0186          PUSH    IX
01E2    FDE5          0187          PUSH    IY
01E4    DD2AE22F      0188          LD      IX,(STKPT)   ; 取用户 SP
01E8    DD23          0189          INC     IX
01EA    DD23          0190          INC     IX           ; 调整指针到用户 SP 的实际值
01EC    DD7EF4        0191          LD      A,(IX-2)     ; IFF 在堆栈内的位置
01EF    E604          0192          AND     04H          ; 屏蔽其它位，保留 IFF
01F1    32DD2F        0193          LD      (UIF),A      ; 存入 UIF
01F4    DD22E22F      0194          LD      (STKPT),IX   ; 保存用户 SP
01F8    3AF32F        0195  NMIS1:  LD      A,(SSFLG)    ; 单步方式吗？
```

```
地址   目的码     语句序号        源语句
01FB   B7         0196            OR      A                ；设置状态标志
01FC   CA2301     0197            JP      Z,DECKY          ；否，是一个由键盘给出的中断
01FF   3E00       0198  NMIS2：   LD      A,00H
0201   32F32F     0199            LD      (SSFLG),A        ；请标志 SSFLG
0204   CD7C06     0200            CALL    UFOR3            ；取断点数目
0207   CA7E00     0201            JP      Z,REMBP2         ；无断点，转去显示 PC 和 A
020A   1842       0202            JR      CCS1C—$          ；转装配断点，使键盘中断有效，并返回用户程序
                  0203  ；命令键转移表
020C   C33002     0204  JPTAB：   JP      CCS1             ；EXEC（连续执行程序）
020F   C37202     0205            JP      CCS2             ；SS（单步执行程序）
0212   C3A102     0206            JP      CCS3             ；MON（监控）
0215   C3A402     0207            JP      CCS4             ；NEXT（下一）
0218   C3ED02     0208            JP      CCS5             ；REG'DISP（辅助寄存器检查）
021B   C30803     0209            JP      CCS6             ；REG DISP（寄存器检查）
021E   C36B03     0210            JP      CCS7             ；PORT EXAM（口检查）
0221   C39503     0211            JP      CCS8             ；MEM EXAM（存储器检查）
0224   C39704     0212            JP      CCS9             ；BP（设置断点）
0227   C3CA04     0213            JP      CCS10            ；PUNCH TAPE（信息转储磁带）
022A   C38105     0214            JP      CCS11            ；LOAD TAPE（磁带输入）
022D   C3D305     0215            JP      CCS12            ；PROG PROM(EPROM 写入)
                  0216  ；EXEC 键处理程序
0230   3E00       0217  CCS1：    LD      A, 00H
0232   D38C       0218            OUT     (DIGLH), A       ；清显示
0234   3AF62F     0219            LD      A, (DIG4)
0237   B7         0220            OR      A                ；设置状态标志
```

```
地址    目的码      语句序号      源语句
0238    280D        0221          JR      Z, CCS1A—$      ; 如果没有输入 4 位数，则转移到继续
                                                              执行方式
023A    CD5906      0222          CALL    UFOR2           ; 将输入的四个数作为起始地址，置入 HL
023D    DD2AE22F    0223          LD      IX, (STKPT)     ; 设置用户 SP
0241    DD75FE      0224          LD      (IX-2), L
0244    DD74FF      0225          LD      (IX-1), H       ; 更换 PC 为新的值
                    0226 ; 继续执行方式
0247    CD7C06      0227 CCS1A :  CALL    UFOR3           ; 测试断点是否有效
024A    202B        0228          JR      NZ, CCS2A—$     ; 是，单步一次
024C    1816        0229          JR      CCS1D—$         ; 若断点无效，允许键盘用 NMI中断中止用户程序
                                                              并用 RET 返回用户程序
                    0230 ; 装配断点并保存用户操作码
024E    DD21E42F    0231 CCS1C:   LD      IX, BPTAB       ; 设定断点表指针
0252    DD6600      0232 CCS1CA:  LD      H, (IX+0)
0255    DD6E01      0233          LD      L, (IX+1)       ; 取操作码的地址
0258    7E          0234          LD      A, (HL)         ; 取操作码
0259    0ECF        0235          LD      C, 0CFH         ; 置 RST8操作码 CF 到 C
025B    71          0236          LD      (HL), C         ; 设置一个新断点
025C    DD7702      0237          LD      (IX+2), A       ; 保存用户操作码
025F    CD3406      0238          CALL    UIX3            ; 指向下一个单元
0262    20EE        0239          JR      NZ, CCS1CA—$    ; 为处理更多的断点而循环返回
                    0240 ; 允许键盘中止用户程序
0264    3E04        0241 CCS1D:   LD      A, 04H
0266    D38C        0242          OUT     (DIGLH), A      ; 允许 NMI 中断
```

```
地址   目的码     语句序号      源语句
0268   3E45       0243          LD      A, 45H
026A   D386       0244          OUT     (CTC2), A   ; CTC 的 ZC/TO 待命
026C   3E01       0245          LD      A, 01H
026E   D386       0246          OUT     (CTC2), A   ; 若输入一个负脉冲，则发生中断请求
0270   1812       0247          JR      CCS2B—$     ; 恢复用户寄存器
                  0248  ; SINGLE STEP 键处理程序
0272   3E00       0249  CCS2:   LD      A, 00H
0274   32F42F     0250          LD      (BFLG), A   ; 清除断点
0277   3E01       0251  CCB2A:  LD      A, 01H
0279   32F32F     0252          LD      (SSFLG), A  ; 设置单步方式标志
027C   3E07       0253          LD      A, 07H
027E   D386       0254          OUT     (CTC2), A   ; 禁止 CTC2 中断
0280   3E0B       0255          LD      A, 0BH      ; 启动CTC,时间常数为 176
0282   D386       0256          OUT     (CTC2), A   ; 等待 NMI中断
                  0257  ; 恢复寄存器
0284   FDE1       0258  CCS2B:  POP     IY
0286   DDE1       0259          POP     IX
0288   E1         0260          POP     HL
0289   D1         0261          POP     DE
028A   C1         0262          POP     BC
028B   F1         0263          POP     AF
028C   08         0264          EX      AF, AF'     ; 辅助寄存器
028D   D9         0265          EXX
028E   F1         0266          POP     AF          ; I 和 IFF
028F   ED47       0267          LD      I, A        ; 恢复 I
```

```
地址    目的码      语句序号        源语句
0291    E1          0268            POP     HL
0292    D1          0269            POP     DE
0293    C1          0270            POP     BC
0294    3ADD2F      0271            LD      A, (UIF)      ; 取 IFF
0297    B7          0272            OR      A             ; 设置状态
0298    C29E02      0273            JP      NZ, CCS2C     ; 转允许中断
029B    F1          0274            POP     AF
029C    F3          0275            DI                    ; 禁止中断
029D    C9          0276            RET                   ; 返回用户程序
029E    F1          0277  CCS2C:     POP     AF
029F    FB          0278            EI                    ; 允许中断
02A0    C9          0279            RET                   ; 返回用户程序
                    0280 ; MON 键处理程序——强迫返回监控程序 RESTR1 (00AE)
                    0281 ;                  并保存用户寄存器
02A1    C3AE00      0282 CCS3:      JP      RESTR1        ; 清标志及进行显示处理
                    0283 ; NEXT 键处理程序
                    0284 ; 用于选择下一个存储单元，下一个口或
                    0285 ; 在EPROM 写入程序中的下一个错误
02A4    3AE12F      0286  CCS4:     LD      A, (MFLG)
02A7    FE01        0287            CP      01H           ; 是存储器检查方式吗?
02A9    2812        0288            JR      Z, CCS4B—$
02AB    3ADE2F      0289            LD      A, (PFLG)
02AE    FE01        0290            CP      01H           ; 是口检查方式吗?
02B0    2826        0291            JR      Z, CCS4C—$
```

```
地址    目的码      语句序号      源语句
02B2    3ADA2F      0292          LD      A, (PRFLG)
02B5    FE01        0293          CP      01H          ; 是 EPROM 写入方式吗?
02B7    C2AE00      0294  CCS4A:  JP      NZ, RESTR1   ; 否, 按无意义键处理
02BA    C33006      0295          JP      CCS12D       ; EPROM 写入方式有效
                    0296;                              转显示下一个错误
02BD    CD5096      0297  CCS4B:  CALL    UFOR2        ; 将 DISMEM 的前四个字节形成一个地址送 HL
02C0    23          0298          INC     HL           ; 选择下一个存储单元地址
02C1    7C          0299          LD      A, H
02C2    CD3C06      0300          CALL    UFOR1        ; IX 已经指向 DISMEM, 更新前两位数
02C5    DD23        0301          INC     IX
02C7    DD23        0302          INC     IX
02C9    7D          0303          LD      A, L
02CA    CD3C06      0304          CALL    UFOR1        ; 更新次两位数
02CD    DD23        0305          INC     IX
02CF    DD23        0306          INC     IX
02D1    7E          0307          LD      A, (HL)      ; 读新的存储单元内容
02D2    CD3C06      0308          CALL    UFOR1        ; 更新最后两位数
02D5    C3F400      0309          JP      DISUP        ; 转显示
02D8    CD5906      0310  CCS4C:  CALL    UFOR2        ; 将 DISMEM 的前四个字节形成一个地址送 HL
02DB    4C          0311          LD      C, H         ; 仅 H 有意义
02DC    0C          0312          INC     C            ; 选择下一个口地址
02DD    ED78        0313          IN      A, C         ; 读入
02DF    CD3C06      0314          LD      A, (C)       ; IX 已经指向 DISMEM
02E0    79          0315          CALL    UFOR1
02E3    DD21FB2F    0316          LD      IX, DSMEM4   ; 指向最后两位数
```

```
地址    目的码      语句序号    源语句
02E7    CD3C06      0317        CALL    UFOR1       ; 更新最后两位数
02EA    C3F400      0318        JP      DISUP       ; 转显示
                    0319  ; 辅助寄存器检查 (REG′ DISP 键)
02ED    3E01        0320  CCS5: LD      A, 01H
02EF    32E02F      0321        LD      (ARFLG) , A             ; 置位辅助寄存器检查标
                                                                  志 ARFLG
02F2    DD21F72F    0322        LD      IX, DISMEM              ; 指向显示缓冲区 DISNEM
                                                                  单元
02F6    3F12        0323        LD      A,12H       ; 撇号送 A
02F8    DD7701      0324        LD      (IX+1) , A              ; 写入显示缓冲区
02FB    CD5504      0325        CALL    ALTER5
02FE    7E          0326        LD      A, (HL)     ; 送入累加器
02FF    DD21FB2F    0327        LD      IX, DSMEM4              ; 指向显示缓冲区后两位
0303    CD3C06      0328        CALL    UFOR1       ; 写 A 到显示缓冲区
0306    185A        0329        JR      CCS6C—$
                    0330  ; 主寄存器检查 ( REG DISP 键)
0308    3E01        0331  CCS6: LD      A, 01H
030A    32DF2F      0332        LD      (RFLG) ,A   ; 置位寄存器检查标志RF LG
030D    CD6E04      0333        CALL    ALTER6
0310    380A        0334        JR      C, CCS6A—$  ; 是对应键小于6的寄存器则转移
0312    7E          0335        LD      A, (HL)     ; 取 (HL)单元内容送A
0313    DD21FB2F    0336        LD      IX, DSMEM4  ; 指向显示缓冲区的后两个单元
0317    CD3C06      0337        CALL    UFOR1       ; 写A到DSMEM4,5
031A    1846        0338        JR      CCS6C—$
                    0339  ; PC,SP,IX,IY 寄存器处理
```

```
地址    目的码        语句序号      源语句
031C    FE03          0340  CCS6A:  CP      3              ; 是键3吗?
031E    2835          0341          JR      2, CCS6B—$                    ; 是, 为IFF
0320    FE02          0342          CP      2              ; 是键2吗?
0322    2817          0343          JR      2, CCS6A—$                    ; 是, 为栈指示器SP
0324    7E            0344          LD      A, (HL)        ; 低字节送A
0325    DD21FB2F      0345          LD      IX, DSMEM4
0329    DD22DB2F      0346          LD      (KEYPTR), IX                  ; 为写入四位新数做准备
032D    CD3C06        0347          CALL    UFOR1          ; 第一个字节写到DSMEM4,5
0330    23            0348          INC     HL
0331    7E            0349          LD      A,(HL)         ; 高字节送A
0332    DD21F923      0350          LD      IX, DSMEM2                    ; 指向前两位
0336    CD3C06        0351          CALL    UFOR1          ; 第二字节写到 DSMEM2,3
0339    182D          0352          JR      CCS6D—$        ; 转显示
033B    3AE32F        0353  CCS6A1:LD       A, (STKPT1)
033E    DD21F92F      0354          LD      IX, DSMEM2
0342    CD3C06        0355          CALL    UFOR1          ; 第一字节送DISMEM2,3
0345    3AE22F        0356          LD      A, (STKPT)
0348    DD21FB2F      0357          LD      IX, DSMEM4
034C    CD3C06        0358          CALL    UFOR1          ; 第二字节送DISMEM4,5
034F    DD22DB2F      0359          LD      (KEYPTR), IX                  ; 为写入四位新数作准备
0353    1813          0360          JR      CCS6D—$
                      0361  ; IFF处理
0355    DD21FB2F      0362  CCS6B:  LD      IX, DSMEM4
0359    3ADD2F        0363          LD      A, (UIF)       ; 取得UIF的值4或0
035C    CD3C06        0364          CALL    UFOR1          ; 写入DISMEM
```

```
地址    目的码      语句序号    源语句
035F    21FE2F      0365        LD      HL, DSMEM7
0362    21FD2F      0366  CCS6C: LD      HL, DSMEM6 ; 设置指针以备更换 IFF 时用
0365    22DB2F      0367        LD      (KEYPTR), HL
0368    C3F400      0368  CCS6D: JP      DISUP
                    0369  ; 口检查(PORT EXAM 键)
036B    3AF52F      0370  CCS7:  LD      A, (DIG2)
036E    FE01        0371        CP      1
0370    2020        0372        JR      NZ, CCS7A—$                ; 验证是否输入了两位数
0372    32DE2F      0373        LD      (PFLG), A                  ; 是,置位口检查标志PFLG
0375    3E10        0374        LD      A, 10H
0377    32F92F      0375        LD      (DSMEM2), A
037A    32FA2F      0376        LD      (DSMEM3), A                ; 对(DSMEM2) (DSMEM3)
                                                                   填"空"符
037D    CD5906      0377        CALL    UFOR2       ; 取得口地址送 HL
0380    4C          0378        LD      C, H        ; 传送前两位数到 C 寄存器
0381    ED78        0379        IN      A, (C)      ; 从该口输入
0383    DD21FB2F    0380        LD      IX, DSMEM4  ; 指向数据位(最后两位)
0387    CD3C06      0381        CALL    UFOR1       ; 写入显示缓冲区第 5,6 单元
038A    DD23        0382        INC     IX
038C    DD23        0383        INC     IX          ; 为口内容改变作准备
038E    DD22DB2F    0384        LD      (KEYPTR), IX               ; 设置口改变时的指针
0392    C3F400      0385  CCS7A: JP      DISUP       ; 转显示
                    0386  ; 存储器检查(MEM EXAM键)
0395    3AF62F      0387  CCS8:  LD      A, (DIG 4)
0398    B7          0388        OR      A           ; 验证是否输入了四位数
```

```
地址    目的码        语句序号        源语句
0399    2816          0389            JR      Z, CCS8A—$
039B    32E12F        0390            LD      (MFLG), A  ; 是，置位存储检查标志 MFLG 为 1
039E    CD5906        0391            CALL    UFOR2      ; 四位数形成地址送 HL
03A1    7E            0392            LD      A, (HL)    ; 取得存储器数据
03A2    DD21FB2F      0393            LD      IX, DSMEM4                    ; 指向正确的数位
03A6    CD3C06        0394            CALL    UFOR1      ; 写入显示缓冲区(DSMEM4)(DSMEM5)
03A9    DD23          0395            INC     IX
03AB    DD23          0396            INC     IX
03AD    DD22DB2F      0397            LD      (KEYPTR), IX                  ; 为更换数据准备指针
03B1    C3F400        0398            JP      DISUP
03B4    DD21F72F      0399  ALTER:    LD      IX, DISMEM
03B8    3ADE2F        0400            LD      A, (PFLG)
03BB    B7            0401            OR      A
03BC    202A          0402            JR      NZ, ALTER2—$                  ; 口内容的改变
03BE    3ADF2F        0403            LD      A, (RFLG)
03C1    B7            0404            OR      A
03C2    2035          0405            JR      NZ, ALTR3—$                   ; 寄存器内容的改变
03C4    3AE02F        0406            LD      A, (ARFLG)
03C7    B7            0407            OR      A
03C8    C24104        0408            JP      NZ, ALTR4                     ; 辅助寄存器内容的改变
03CB    3AE12F        0409            LD      A, (MFLG)
03CE    B7            0410            OR      A
03CF    CAAE00        0411            JP      Z, RESTR1                     ; 不正确操作，转回 RESTR 1
03D2    CD5906        0412            CALL    UFOR2      ; 取得地址送 HL
03D5    DD7E06        0413            LD      A, (IX+6)                     ; 取得字节的高四位(半字节)
```

```
地址   目的码      语句序号   源语句
03D8   CD8E04      0414       CALL    ALTER7
03DB   DDB607      0415       OR      (IX+7)          ;与字节的低四位相或
03DE   77          0416       LD      (HL),A          ;此字节写入存储器中
03DF   7E          0417       LD      A,(HL)          ;再读回A
03E0   DD21FB2F    0418 ALTER1:LD     IX,DSMEM4
03E4   CD3C06      0419       CALL    UFOR1           ;把新的数据写入显示缓冲区(DSMEM4)(DSMEM5)
03E7   C9          0420       RET
03E8   CD5906      0421 ALTER2:CALL   UFOR2            ;取得的地址送HL
03EB   DD7E06      0422       LD      A,(IX+6)                    ;取得字节的高四位
03EE   CD8E04      0423       CALL    ALTER7
03F1   DDB607      0424       OR      (IX+7)          ;与字节的低四位相或
03F4   4C          0425       LD      C,H
03F5   ED79        0426       OUT     (C),A           ;写入口中
03F7   18E7        0427       JR      ALTER1-$
                   0428 ;主寄存器改变内容
03F9   CD6E04      0429 ALTR3:CALL    ALTER6
03FC   380D        0430       JR      C,ALTR3A-$      ;是键号小于6的16位寄存器和IFF吗?
03FE   DD7E06      0431       LD      A,(IX+6)        ;取得字节的高四位
0401   CD8E04      0432       CALL    ALTER7
0404   DDB607      0433       OR      (IX+7)          ;与字节的低四位相或
0407   77          0434       LD      (HL),A          ;修改保护在寄存器存放区内的内容
0408   C3E003      0435       JP      ALTER1
040B   FE03        0436 ALTR3A:CP     3               ;是键3吗?
040D   2824        0437       JR      2,ALTR3B-$                  ;是,IFF
040F   FE0228      0438       CP      2               ;是键2吗?
```

```
地址    目的码      语句序号    源语句
0411    282A        0439        JR      Z, ALTR3C - $           ; 是, SP
0413    DD7E04      0440        LD      A, (IX + 4)             ; 取字节的高四位
0416    CD8E04      0441        CALL    ALTER7
0419    DDB605      0442        OR      (IX + 5)    ; 与字节的低四位相或
041C    23          0443        INC     HL
041D    77          0444        LD      (HL), A     ; 修改高字节
041E    DD21F92F    0445        LD      IX, DSMEM2
0422    CD3C06      0446        CALL    UFORI       ; 写入数据到显示缓冲区
0425    DD7E04      0447        LD      A, (IX + 4)             ; 取字节的高四位
0428    CD8E04      0448        CALL    ALTER7
042B    DDB605      0449        OR      (IX + 5)    ; 与字节的低四位相或
042E    2B          0450        DEC     HL
042F    77          0451        LD      (HL), A     ; 修改低字节
0430    C3E003      0452        JP      ALTER1
0433    3AFE2F      0453 ALTR3B :LD     A,(DSMEM7)              ; 获得新值
0436    32DD2F      0454        LD      (UIF), A    ; 修改 UIF
0439    32FC2F      0455        LD      (DSMEM5),A              ; 写入显示缓冲区
043C    C9          0456        RET
                    0457 ; SP 的内容是不容许改变的
043D    E1          0458 ALTR3C:POP     HL          ; 哑指令
043E    C3AE00      0459        JR      RESTR1
                    0460 ; 辅助寄存器改变内容
0441    DD21F72F    0461 ALTR4 : LD     IX, DISMEM              ; 指向存放在 DISMEM, 其中存有代
                                                                  表寄存器的键号
0445    CD5504      0462        CALL    ALTER5
```

```
地址   目的码      语句序号        源语句
0448   DD7E06      0463            LD      A, (IX+6)          ; 取字节的高四位
044B   CD8E04      0464            CALL    ALTER7
044E   DDB607      0465            OR      (IX+7)   ; 与字节的低四位相或
0451   77          0466            LD      (HL), A  ; 修改在寄存器存放区内的内容
0452   C3E003      0467            JP      ALTER1   ; 转写新数据到显示缓冲区
0455   DD5E00      0468 ALTER5:    LD      E, (IX+0); 取存于 DISMEM, 其中存有代表寄存器的键号
0458   0600        0469            LD      B, 00H
045A   1600        0470            LD      D, 00H
045C   21E507      0471            LD      HL, REGTBP         ; 指向辅助寄存器表
045F   19          0472            ADD     HL, DE   ; 加键值偏移量
0460   4E          0473            LD      C, (HL)  ; 取值
0461   79          0474            LD      A, C
0462   FE19        0475            CP      19H      ; 是非法值吗?
0464   CAAE00      0476            JP      Z, RESTR1          ; 是, 重新启动监控程序
0467   2AE22F      0477            LD      HL, (STKPT)        ; 否, 指向用户 SP
046A   B7          0478            OR      A        ; 清进位位
046B   ED42        0479            SBC     HL, BC   ; 从 SP 减去偏移量
046D   C9          0480            RET
046E   DD21F72F    0481 ALTER6:    LD      IX, DISMEM ; 指向 DISMEM, 其中存有代表寄存器的键号
0472   DD5E00      0482            LD      E, (IX)  ; 得到代表寄存器的键号
0475   0600        0483            LD      B, 00H
0477   1600        0484            LD      D, 00H
0479   21D507      0485            LD      HL, REGTB ; 指向寄存器表 REGTB
047C   19          0486            ADD     HL, DE   ; 加键值偏移量
047D   4E          0487            LD      C, (HL)  ; 得到表中相应项的内容
```

```
地址    目的码      语句序号      源语句
047E    79          0488          LD      A, C
047F    FE19        0489          CP      19H          ; 是非法值(即 25D)吗?
0481    CAAE00      0490          JP      Z, RESTR1    ; 是, 重新启动
0484    2AE22F      0491          LD      HL, (STKPT); 否, 指向用户 SP
0487    7B          0492          LD      A, E         ; 重新得到键值
0488    B7          0493          OR      A            ; 清进位位
0489    ED42        0494          SBC     HL, BC       ; 从用户SP 减去表中相应项内容
048B    FE06        0495          CP      6            ; 判断是何寄存器
048D    C9          0496          RET
048E    CB27        0497ALTER7:   SLA     A
0490    CB27        0498          SLA     A
0492    CB27        0499          SLA     A
0494    CB27        0500          SLA     A
0496    C9          0501          RAT
                    0502 ; 断点的设置
                    0503 ; 如断点表还有空间就在断点表中设置一个入口
                    0504 ; 实际上断点是在执行命令中才放入
                    0505 ; RAM 之中的
0497    3AF62F      0506  CCS9:   LD      A, (DIG4) ; 取标志
049A    B7          0507          OR      A         ; 设置状态
049B    2005        0508          JR      NZ, CCS90— $          ; 输入了四位数吗?
049D    32F42F      0509          LD      (BFLG), A             ; 否, 则清除所有的断点
04A0    1810        0510          JR      CCS91—$
04A2    CD5906      0511  CCS90:  CALL    UFOR2     ; 得到断点地址在HL中
04A5    CD7C06      0512          CALL    UFOR3     ; 得到断点状态
```

```
地址    目的码      语句序号    源语句
04A8    2810        0513        JR      Z,CCS9B—$ ；无断点，转去在表内设置一个断点
04AA    78          0514        LD      A, B
04AB    FE05        0515        CP      05H         ；表满了吗?
04AD    2006        0516        JR      NZ, CCS9A—$                    ；否，继续
04AF    C3AE00      0517        JP      RESTR1      ；是，清显示
04B2    C3F400      0518  CCS91:JP      DISUP
                    0519  ；在表中找到第一个未使用的空间
04B5    CD3406      0520  CCS9A:CALL    UIX3        ；IX+3 和B-1
04B8    2OFB        0521        JR      NZ, CCS9A—$                    ；返回，直到找到自由空间
                    0522  ；在表内插入新断点
04BA    3AF42F      0523  CCS9B:LD      A, (BFLG)
04BD    3C          0524        INC     A
04BE    32F42F      0525        LD      (BFLG), A                      ；标志 BFLG 增1
04C1    DD7400      0526        LD      (IX), H     ；把断点地址写入断点表内
04C4    DD7501      0527        LD      (IX+1), L
04C7    C3F400      0528        JP      DISUP
                    0529  ；按 KC 格式将信息转储到盒式磁带
                    0530  ；起始地址在 DE，结束地址在 HL
                    0531  ；在此程序中，中断始终有效
04CA    CDA706      0532  CCS10:CALL    UFOR4       ；清显示
04CD    3E01        0533        LD      A, 1
04CF    32D92F      0534        LD      (FLG24), A ；为发逻辑1而设置 FLG24 标志为1
04D2    ED5E        0535        IM      2
04D4    01FAO7      0536        LD      BC, CTC1P
04D7    78          0537        LD      A, B
```

```
地址    目的码      语句序号    源语句
04D8    ED47        0538        LD      I, A
04DA    79          0539        LD      A, C
04DB    D384        0540        OUT     (CTC0), A                   ; 中断矢量
04DD    3E85        0541        LD      A, 85H
04DF    D385        0542        OUT     (CTC1), A                   ; CTC 中断允许
04D1    3E1A        0543        LD      A, 1AH
04D3    D385        0544        OUT     (CTC1), A                   ; 送时间常数
04E5    3AC02F      0545        LD      A, (PUNHSH)
04E8    57          0546        LD      D, A
04E9    3AC12F      0547        LD      A, (PUNHSL)                 ; 得到起始地址
04EC    5F          0548        LD      E, A
04FD    3AC22F      0549        LD      A, (PUNHEH)
04F0    67          0550        LD      H, A
04F1    3AC32F      0551        LD      A, (PUNHEL)                 ; 得到结束地址
04F4    6F          0552        LD      L, A
04F5    AF          0553        XOR     A           ; 清进位位
04F6    ED52        0554        SBC     HL, DE      ; 计算数据块大小
04F8    23          0555        INC     HL
04F9    E5          0556        PUSH    HL          ; 计算结果存在 HL 中
04FA    210000      0557        LD      HL, 0000H
04FD    0603        0558        LD      B, 03H      ; 设置计数值以获得40秒延时
04FF    FB          0559        EI
0500    76          0560 CCS10A: HALT               ; 等待 CTC1 中断
                    0561 ; ※※※※※※※
0501    2D          0562        DEC     L
```

地址	目的码	语起序号		源语句		
0502	20FC	0563		JR	NZ, CCS10A—$	
0504	25	0564		DEC	H	
0505	20F9	0565		JR	NZ, CCS10A—$	；延时循环
0507	10F7	0566		DJNZ	CCS10A—$	；40 秒全 1 开头段
0509	3E3A	0567	CCS10B:	LD	A, 03AH	；以冒号开始
050B	CDF706	0568		CALL	OTCHR1	；输出冒号
050E	AF	0569		XOR	A	；清进位位
050F	011000	0570		LD	BC, 10H	
0512	E1	0571		POP	HL	
0513	ED42	0572		SBC	HL, BC	；HL 和最大字节数 16 相比
0515	3009	0573		JR	NC, CCS10C—$	；NC = >HL>10H
0517	09	0574		ADD	HL, BC	；当少于 16 个字节时，恢复数据块字节数
0518	85	0575		ADD	A, L	；如L = 0，就会设置状态标志
0519	47	0576		LD	B, A	
051A	2E00	0577		LD	L, 0	
051C	283C	0578		JR	Z, CCS10F—$	
051E	1801	0579		JR	CCS10D—$	
0520	41	0580	CCS10C:	LD	B, C	；B = 数据块大小
0521	E5	0581	CCS10D:	PUSH	HL	；保护进栈
0522	0E00	0582		LD	C, 0	；清检查和
0524	78	0583		LD	A, B	；转储记录长度
0525	CDD506	0584		CALL	UPACCS	
0528	3AC02F	0585		LD	A, (PUNHSH)	
052B	67	0586		LD	H, A	
052C	CDD506	0587		CALL	UPACCS	

```
地址  目的码   语句序号  源语句
052F  3AC02F   0588      LD      A, (PUNHSL) ; 转储起始地址高字节
0532  6F       0589      LD      L, A
0533  CDD506   0590      CALL    UPACCS
0536  AF       0591      XOR     A           ; 为转储"记录类型"置累加器为 0
0537  CDD506   0592      CALL    UPACCS
053A  7E       0593 CCS10E: LD    A, (HL)     ; 将被转储的数据送累加器 A
053B  CDD506   0594      CALL    UPACCS
053E  23       0595      INC     HL
053F  10F9     0596      DJNZ    CCS10E—$
0541  97       0597      SUB     A
0542  91       0598      SUB     C
0543  CDD506   0599      CALL    UPACCS      ; 转储"检查和"
0546  3E0D     0600      LD      A, 0DH      ; 回车字符
0548  CDD506   0601      CALL    UPACCS
054B  3E0A     0602      LD      A, 0AH      ; 换行字符
054D  CDD506   0603      CALL    UPACCS
0550  7C       0604      LD      A, H
0551  32C02F   0605      LD      (PUNHSH), A                 ; 保存地址高字节
0554  7D       0606      LD      A, L
0555  32C12F   0607      LD      (PUNHSL), A                 ; 保存地址低字节
0558  18AF     0608      JR      CCS10B—$                    ; 继续转储更多的数据
055A  0603     0609 CCS10F: LD    B, 3
055C  AF       0610 CCS10G: XOR   A
055D  4F       0611      LD      C, A
055E  CDD506   0612      CALL    UPACCS      ; 转储"文件结束"
```

```
地址    目的码      语句序号      源语句
0561    10F9        0613          DJNZ    CCS10G—$
0563    3E01        0614          LD      A, 1H
0565    CDD506      0615          CALL    UPACCS
0568    97          0616          SUB     A
0569    91          0617          SUB     C
056A    CDD506      0618          CALL    UPACCS      ; 转储"文件结束"的检查和
056D    21FFFF      0619 CCS10H:  LD      HL, 0FFFFH                ; 设置5秒全1结尾的计数值
0570    FB          0620          EI                  ; 开中断
0571    76          0621          HALT
                    0622 ; ※※※※※※※※※
0572    2D          0623 CCS10J:  DEC     L
0573    20FD        0624          JR      NZ, CCS10J—$
0575    25          0625          DEC     H
0576    20FA        0626          JR      NZ, CCS10J—$
0578    F3          0627          DI
0579    3F03        0628          LD      A, 03H
057B    D385        0629          OUT     (CTC1), A   ; 关 CTC, 停止发出脉冲波
057D    F3          0630          DI                  ; 关中断
057E    C3AE00      0631          JP      RESTR1      ; 转重新启动监控程序
                    0632 ; 从盒式磁带向存储器输入数据(Kansas city 格式)
0581    ED5E        0633 CCS11:   IM      2           ; 选择中断方式 2
0583    21FF0F      0634          LD      HL, 0FFFH   ; 开始 15 秒延时
0586    0620        0635          LD      B, 20H
0588    2D          0636 CCS11A:  DEC     L
0589    20FD        0637          JR      NZ, CCS11A - $
```

```
地址    目的码      语句序号        源语句
058B    25          0638            DEC     H
058C    20FA        0639            JR      NZ, CCS11A - $
058E    10F8        0640            DJNZ    CCS11A - $
                    0641    ; 磁带现在进入导引部分—启动主循环
0590    CD5807      0642    CCS11G: CALL    INCHR       ; 取得第一个字
0593    D63A        0643            SUB     03AH        ; 检测是否是冒号
0595    20F9        0644            JR      NZ, CCS11G—$            ; 循环返回
0597    4F          0645            LD      C, A        ; "检查和"预置 0
0598    CDDB06      0646            CALL    ULACC       ; 收到记录长度及其检查和
059B    47          0647            LD      B, A
059C    CDDB06      0648            CALL    ULACC       ; 收到地址的高字节
059F    57          0649            LD      D, A
05A0    CDDB06      0650            CALL    ULACC       ; 收到地址的低字节
05A3    5F          0651            LD      E, A
05A4    CDDB06      0652            CALL    ULACC       ; 收到记录类型
05A7    3D          0653            DEC     A           ; 记录类型为"文件结束"吗?
05A8    F5          0654            PUSH    AF
05A9    2807        0655            JR      Z, CCS11J - $           ; 如是则转移
05AB    CDDB06      0656    CCS11H: CALL    CLACC       ; 不是，接收数据
05AE    12          0657            LD      (DE), A     ; 存储数据
05AF    13          0658            INC     DE
05B0    10F9        0659            DJNZ    CCS11H—$            ; 为接收更多的数据而循环
05B2    CDDB06      0660    CCS11J: CALL    ULACC       ; 接收"检查和"
05B5    AF          0661            XOR     A           ; 清累加器
05B6    81          0662            ADD     A, C
```

```
地址    目的码      语句序号      源语句
05B7    2814        0663          JR      Z, CCS11K—$                    ; 用检查和校验无误
                    0664 ; 指示接收差错
05B9    DD21F72F    0665          LD      IX, DISMEM                     ; 指向显示缓冲区
05BD    7A          0666          LD      A, D
05BE    CD3C06      0667          CALL    UFOR1          ; 显示高字节
05C1    DD21F92F    0668          LD      IX, DSMEM2
05C5    7B          0669          LD      A, E
05C6    CD3C06      0670          CALL    UFOR1          ; 显示低字节
05C9    F1          0671          POP     AF             ; 恢复堆栈指针 SP
05CA    C3F400      0672          JP      DISUP          ; 转显示
05CD    F1          0673 CCS11K:  POP     AF             ; 判断是否为“文件结束”
05CE    20C0        0674          JR      NZ, CCS11G—$                   ; 转入下一个记录的循环
05D0    C3AE00      0675          JP                     ; 重新启动监控程序
                    0676 ; 用于 2758 和 2716EPROM 的 EPROM 编程器
                    0677 ; 从以 2000H 为起始地址的 RAM 传送数据
                    0678 ; 到起始地址为 1000H 的 EPROM
                    0679 ; 被传送的字节数(由四个十六进制数表示)存
                    0680 ; 放在显示缓冲区内
05D3    3E01        0681 CCS12:   LD      A, 01H
05D5    32DA2F      0682          LD      (PRFLG) ,A ; EPROM 编程标志置位
05D8    CD5906      0683          CALL    UFOR2          ; 得到字节数
05DB    E5          0684          PUSH    HL             ; 保护
05DC    C1          0685          POP     BC
05DD    E5          0686          PUSH    HL
05DE    210020      0687          LD      HL, 2000H      ; RAM 源地址送HL
```

```
地址    目的码      语句序号    源语句
05E1    110010      0688        LD      DE, 1000H       ; EPROM 目的地址送 DE
05E4    3E25        0689 CCS12A: LD     A, 25H          ; CTC用于 26ms 定时输出 ZC/TO2
05E6    D386        0690        OUT     (CTC2) ,A       ; 不产生中断
05E8    3ECB        0691        LD      A, 0CBH
05EA    D386        0692        OUT     (CTC2) ,A       ; 时间常数
05EC    3E80        0693        LD      A, 80 H         ; 清显示
05EE    D38C        0694        OUT     (DIGLH) ,A      ; 发出 EPROM 编程允许脉冲 (PGM PULSE
                                                        ENABLE)
05F0    EDA0        0695        LDI                     ; 插入等待状态直到
05F2    3E00        0699        LD      A, 00H          ; CTC2定时输出两次
05F4    D38C        0697        OUT     (DIGLH) ,A      ; 清EPROM编程允许脉冲
05F6    3E03        0698        LD      A, 03H          ; CTC2 复位
05F8    D386        0699        OUT     (CTC2),A
05FA    EAE405      0700        JP      PE, CCS12A      ; 如 BC－1≒0,循环返回
                    0701 ; 校验 EPROM 的内容与 RAM 的内容是否相同
05FD    C1          0702        POP     BC              ; 恢复字节数
05FE    210020      0703        LD      HL, 2000H       ; HL 指向 RAM
0601    110010      0704        LD      DE, 1000H       ; DE指向 EPROM
0604    1A          0705 CCA12B: LD     A, (DE)         ; 取 EPROM 数据
0605    EDA1        0706        CPI                     ; 与 RAM 比较
0607    2006        0707        JR      NZ,CCS12C-$     ; 有错, 指出错误
0609    E2AE00      0708        JP      PO,RESTR1       ; 无错—返回
060C    13          0709        INC     DE              ; 更新DE
060D    18F5        0710        JR      CCS12B—$        ; 检查下一个字节
                    0711 ; 错误处理程序
```

```
地址    目的码        语句序号    源语句
060F    F5            0712 CCS12C: PUSH    AF          ; 保护错误数据
0610    C5            0713         PUSH    BC          ; 保护字节计数
0611    D5            0714         PUSH    DE          ; 保护错误地址
0612    D9            0715         EXX                 ; 保护主寄存器
0613    D1            0716         POP     DE          ; DE 为 EPPOM 出错处地址
0614    C1            0717         POP     BC          ; BC 为字节计数
0615    DD21F72F      0718         LD      IX, DISMEM
0619    7A            0719         LD      A, D
061A    CD3C06        0720         CALL    UFOR1       ; 高字节送显示缓冲区
061D    DD21F92F      0721         LD      IX, DSMEM2
0621    7B            0722         LD      A, E
0622    CD3C06        0723         CALL    UFOR1       ; 低字节送显示缓冲区
0625    DD21FB2F      0724         LD      IX, DSMEM4
0629    F1            0725         POP     AF          ; 取回错误数据
062A    CD3C06        0726         CALL    UFOR1       ; 错误数据送显示缓冲区
062D    C3F400        0727         JP      D1SUP       ; 显示
                      0728 ; 在错误显示期间 NEXT 键输入
0630    D9            0729 CCS12D: EXX                 ; 取得返回的各指针
0631    13            0730         INC     DE          ; 为测试下一个错误而增 1DE
0632    18D0          0731         JR      CCS12B—$    ; 为测试其它错误而循环
                      0732 ; ※※※※※※赋值
>008C                 0733 DIGLH:  EQU     8CH         ; 只写数位选择输出口
>0090                 0734 KBSEL:  EQU     90H         ; 只读键盘选择输入口
>0084                 0735 CTC0 :  EQU     84H         ; 中断矢量
>0085                 0736 CTC1 :  EQU     85H         ; 磁带接口—转储
```

```
地址    目的码      语句序号      源语句
>0086               0737 CTC2 :   EQU     86 H        ; 单步/EPROM 编程器
>0087               0738 CTC3 :   EQU     87 H        ; 磁带接口—输入
>0088               0739 SEGLH:   EQU     88 H        ; 只写字形锁存器输出口
                    0740          END
```

```
地址     目的码          语句序号        源语句
                         0002    程序名称  UTIL
                         0003 ; 实用程序块
                         0004 ; 版本1.5  5/13 /78
>0634                    0005            ORG     0634H
                         0006            PSECT   REL
                         0007            GLOBAL  DIG2
                         0008            GLOBAL  DIG4
                         0009            GLOBAL  DIG8
                         0010            GLOBAL  UIX3
                         0011            GLOBAL  UFOR1
                         0012            GLOBAL  D20MS
                         0013            GLOBAL  UFOR2
                         0014            GLOBAL  UFOR3
                         0015            GLOBAL  UFOR4
                         0016            GLOBAL  BPTAB
                         0017            GLOBAL  SSFLG
                         0018            GLOBAL  UABIN
                         0019            GLOBAL  UBASC
                         0020            GLOBAL  UPACC
                         0021            GLOBAL  UPACCS
                         0022            GLOBAL  ULACC
                         0023            GLOBAL  OTCHR
                         0024            GLOBAL  KYTBL
                         0025            GLOBAL  REGTB
                         0026            GLOBAL  REGTBP
```

```
地址    目的码      语句序号      源语句
                    0027          GLOBAL    DISMEM
                    0028          GLOBAL    STKPT
                    0029          GLOBAL    SEGPT
                    0030          GLOBAL    KEYPTR
                    0031          GLOBAL    OTCHR
                    0032          GLOBAL    OTCHR1
                    0033          GLOBAL    INCHR
                    0034          GLOBAL    FLG24
                    0035          GLOBAL    RFLG
                    0036          GLOBAL    ARFLG
                    0037          GLOBAL    BFLG
                    0038          GLOBAL    PFLG
                    0039          GLOBAL    CTC1P
                    0040          GLOBAL    MFLG
                    0041          GLOBAL    PRFLG
                    0042          GLOBAL    CTC3L
                    0043          GLOBAL    CTC1P
                    0044          GLOBAL    UFGCR
                    0045          GLOBAL    OTCHR6
                    0046          GLOBAL    INCHR4
                    0047 ; IX 加 3 而 B 减 1
                    0048 ; 用于查断点表，该表存有
                    0049 ; 断点地址和用户操作码
0634    DD 23       0050  UIX3;   INC       IX
0636    DD 23       0051          INC       IX
```

```
地址    目的码      语句序号      源语句
0638    DD 23       0052          INC       IX
063A    05          0053          DEC       B
063B    C9          0054                                  ；转用户检测断点溢出
                    0055 ；IX 指向显示缓冲区的一个和下一个要写入的单元
                    0056 ；写入的数据来自累加器
                    0057 ；UFOR1 执行这一写操作
063C    47          0058  UFOR1:LD        B, A            ；保护 A 的内容
063D    E60F        0059        AND       00FH            ；屏蔽掉字节的高四位
063F    DD7701      0060        LD        (IX+1), A       ；写入字节的低四位
0642    78          0061        LD        A, B            ；取回完整字节
0643    CB3F        0062        SRL       A
0645    CB3F        0063        SRL       A
0647    CB3F        0064        SRL       A
0649    CB3F        0065        SRL       A               ；字节的高四位移至低四位
064B    DD7700      0066        LD        (IX+0), A       ；将字节的高四位写到显示缓冲区
064E    C9          0067        RET
                    0068 ；延时 20 毫秒后返回
064F    21FF08      0069  D20MS: LD        HL, 08FFH
0652    2D          0070 D20MS1: DEC        L
0653    20FD        0071        JR        NZ, D20MS1 - $
0655    25          0072        DEC       H
0656    20FA        0073        JR        NZ, D20MS1 - $
0658    C9          0074        RET
                    0075 ；将在显示缓冲区中的前四位数组成地址置于 HL 中，
                    0076 ；用于存储器和口的更新命令，设置断点时也用到
```

```
地址    目的码      语句序号    源语句
0659    DD21F72F    0077  UFOR2:LD      IX, DISMEM  ; 指向 DISMEM
065D    DD7E00      0078        LD      A, (IX)     ; 取得第一位数
0660    CB27        0079        SLA     A
0662    CB27        0080        SLA     A
0664    CB27        0081        SLA     A
0666    CB27        0082        SLA     A           ; 在 A 内移到高四位
0668    DDB601      0083        OR      (IX+1)      ; 引入低四位
066B    67          0084        LD      H, A        ; 存入 H
066C    DD7E02      0085        LD      A, (IX+2)   ; 取来第二字节的高四位
066F    CB27        0086        SLA     A
0671    CB27        0087        SLA     A
0673    CB27        0088        SLA     A
0675    CB27        0089        SLA     A
0677    DDB603      0090        OR      (IX+3)      ; 引入低四位
067A    6F          0091        LD      L, A        ; 完成指针 HL 的建立
067B    C9          0092        RET
                    0093 ; 令 IX 指向断点表地址而(BFLG)标志放入 B 中
067C    DD21E42F    0094  UFOR3:LD      IX, BPTAB   ; IX 指向表的首地址
0680    3AF42F      0095        LD      A, (BFLG)   ; 有效的断点数目
0683    B7          0096        OR      A           ; 设置状态
0684    47          0097        LD      B, A
0685    C9          0098        RET                 ; 转用户测试状态标志
                    0099 ; 标志初始化程序
0686    21F72F      0100  UFGCR:LD      HL, DISMEM
0689    22DB2F      0101        LD      (KEYPTR,HL; 指向显示缓冲区
```

```
地址    目的码      语句序号      源语句
068C    3E00        0102          LD      A, 00H
068E    32F52F      0103          LD      (DIG2), A  ；各个标志置0
0691    32F62F      0104          LD      (DIG4), A
0694    32F32F      0105          LD      (SSFLG), A
0697    32DA2F      0106          LD      (PRFLG), A
069A    32DE2F      0107          LD      (PFLG), A
069D    32E 12F     0108          LD      (MFLG), A
06A0    32DF2F      0109          LD      (RFLG), A
06A3    32 E02F     0110          LD      (ARFLG)A
06A6    C9          0111          RET
                    0112 ；将空码送入显示缓冲区——DISMEM
06A7    0608        0113  UFOR4:LD       B, 08H
06A9    21F72F      0114          LD      HL, DISMEM
06AC    3E10        0115          LD      A, 10H      ；空码送A
06AE    77          0116 UFOR4A: LD       (HL), A
06AF    23          0117          INC     HL
06B0    10FC        0118          DJNZ    UFOR4A - $；循环送空码直到送完为止
06B2    C9          0119          RET
                    0120 ；将累加器中的一个ASCII码数转换为二进制数
06B3    D630        0121  UABIN:SUB       030H
06B5    FE0A        0122          CP      10
06B7    F8          0123          RET     M
06B8    D607        0124          SUB     7
06BA    C9          0125          RET
                    0126 ；将累加器中的一个4位二进制数转换为ASCII码
```

```
地址    目的码      语句序号      源语句
06BB    E60F        0127  UBASC:AND       0FH         ; 屏蔽掉字节的高四位
06BD    C690        0128        ADD       A, 90H
06BF    27          0129        DAA
06C0    CE40        0130        ADC       A, 40H
06C2    27          0131        DAA
06C3    C9          0132        RET
                    0133 ; 将累加器内容按两个字进行转储——首先转储高四位
06C4    D9          0134  UPACC: EXX                 ; OTCHR 使用辅助寄存器
06C5    F5          0135        PUSH      AF        ; 保护累加器
06C6    0F          0136        RRCA
06C7    0F          0137        RRCA
06C8    0F          0138        RRCA
06C9    0F          0139        RRCA
06CA    CDF406      0140        CALL      OTCHR
06CD    F1          0141        POP       AF
06CE    E60F        0142        AND       0FH       ; 屏蔽掉高四位
06D0    CDF406      0143        CALL      OTCHR
06D3    D9          0144        EXX                 ; 恢复寄存器
06D4    C9          0145        RET
                    0146 ; 对累加器中的两个字符求检查和，而后转储
06D5    F5          0147 UPACCS: PUSH       AF        ; 保护累加器
06D6    81          0148        ADD       A, C          ; 求检查和而后送入 C 寄存器
06D7    4F          0149        LD        C, A
06D8    F1          0150        POP       AF
06D9    18E9        0151        JR        UPACC-$       ; 累加器内容转储
```

```
地址    目的码      语句序号      源语句
                    0152 ；从磁带中读两个字符并转换为二进制数，这个二进
                    0153 ；制数存放在A累加器中，并将求得的检查和保存在C寄存器
06DB    C5          0154 ULACC: PUSH    BC          ；保护C寄存器（检查和）
06DC    CD5807      0155        CALL    INCHR       ；读字符的第一位
06DF    CDB306      0156        CALL    UABIN       ；转换为二进制数
06E2    07          0157        RLCA
06E3    07          0158        RLCA
06E4    07          0159        RLCA
06E5    07          0160        RLCA                ；移位到高四位
06E6    4F          0161        LD      C, A        ；保留第一位数
06E7    CD5807      0162        CALL    INCHR       ；取第二位数
06EA    CDB306      0163        CALL    UABIN       ；转换成二进制数
06ED    B1          0164        OR      C           ；合并两个四位
06EE    C1          0165        POP     BC          ；取回检查和
06EF    F5          0166        PUSH    AF          ；保护累加器内容
06F0    81          0167        ADD     A, C        ；将八位二进制数加到检查和
06F1    4F          0168        LD      C, A        ；存入C寄存器
06F2    F1          0169        POP     AF          ；取回累加器内容
06F3    C9          0170        RET
                    0171 ；程序使用CTC的通道1作为基本时基（4800HZ或2400HZ）
                    0172 ；入口是：放一个二进制数在累加器的低四位，本程序将这个
                    0173 ；二进制数转换为ASCII码，并将此码和一个起始位以
                    0174 ；及两个停止位一起发出，从CTC 1的输出端（ZC/TO）
                    0175 ；输出的是两倍于Kansas City标准频率的频率（1为
                    0176 ；4800HZ，0为2400HZ）。
```

```
地址    目的码      语句序号      源语句
06F4    CDBB06      0177  OTCHR:CALL      UBASC          ; 转换为 ASCII 码
09F7    FB          0178 OTCHR1:EI
06F8    57          0179          LD      D, A           ; 累加器的内容送 D 寄存器
06F9    3F10        0180          LD      A, 10H         ; 为 CTC 中断将 A 予置初值
06FB    2E0A        0181          LD      L, 0AH         ; 字位的计数
06FD    CB12        0182          RL      D              ; 进位位送 D0
06FF    FE01        0183 OTCHR2:CP        01H            ; A = 1 吗? 等待到 A = 1 为止
0701    20FC        0184          JR      NZ, OTCHR2 - $ ; 等待 CTC 下一次计数到 0，直到最
                                                         后一次(A = 1)为止
0703    47          0185          LD      B, A           ; 保存累加器内容
0704    3E00        0186          LD      A, 00H
0706    32D92F      0187          LD      (FLG24), A     ; 为发出起始位而清标志(FLG24)
0709    78          0188          LD      A, B
070A    76          0189 OTCHR3:HALT
                    0190 ; ※※※※※※※※※※※
070B    37          0191          SCF                    ; 设置进位位
070C    CB1A        0192          RR      D              ; 右移D寄存器一次
070E    2D          0193          DEC
070F    2007        0194          JR      NZ, OTCHR4 - $
0711    3E01        0195          LD      A, 01H
0713    32D92F      0196          LD      (FLG24), A     ; 字发完——跟随发 1
0716    F3          0197          DI                     ; 出口，禁止中断;
0717    C9          0198          RET
0718    FE01        0199 OTCHR4:CP        1              ; 反复等待下一次计数到 0，直到最后一次
                                                         (A = 1)为止。
```

```
地址    目的码      语句序号        源语句
071A    20FC        0200            JR      NZ, OTCHR4 - $  ; 还未到，则等待
071C    CB42        0201            BIT     0, D            ; 测试下一位
071E    2009        0202            JR      NZ, OTCHR5 - $  ; 下一位是 1
0720    47          0203            LD      B, A
0721    3E00        0204            LD      A, 00H          ; 下一位是 0，将 0 送入累加器
0723    32D92F      0205            LD      (FLG24), A      ; 清 (FLG24) 标志
0726    78          0206            LD      A, B
0727    18E1        0207            JR      OTCHR3 - $
0729    47          0208 OTCHR5:    LD      B, A
072A    3E01        0209            LD      A, 01H
072C    32D92F      0210            LD      (FLG24), A      ; 置 (FLG24) 标志为 1
072F    78          0211            LD      A, B            ; 恢复累加器
0730    18D8        0212            JR      OTCHR3 - $      ; 等待字符转储结束
                    0213 ; 在转储期间的 CTC1 中断服务程序
0732    3D          0214 OTCHR6:    DEC     A               ; A 为周期计数器
0733    2020        0215            JR      NZ, OTCHR8 - $  ; 计数未完，转中断返回
0735    DD21D92F    0216            LD      IX, FLG24
0739    DDCB0046    0217            BIT     0, (IX + 0)     ; 测试 (FLG24) 标志
073D    200C        0218            JR      NZ, OTCHR7 - $  ; (FLG24)为 0 则不转移
073F    3E85        0219            LD      A, 85H
0741    D385        0220            OUT     (CTC1), A       ; 新控制字中断有效
0743    3E34        0221            LD      A, 34H
0745    D385        0222            OUT     (CTC1), A       ; 为 1200HZ 设时间常数
0747    3E08        0223            LD      A, 8            ; 计数 8 个周期
0749    180A        0224            JR      OTCHR8 - $      ; 返回
```

```
地址    目的码      语句序号      源语句
074B    3E85        0225 OTCHR7:LD       A, 85H
074D    D385        0226         OUT     (CTC1), A      ; 新控制字
074F    3E1A        0227         LD      A, 1AH
0751    D385        0228         OUT     (CTC1), A      ; 为2400HZ设时间常数
0753    3E10        0229         LD      A, 10H         ; 计数16个周期
0755    FB          0230 OTCHR8:EI
0756    ED4D        0231         RETI
                    0232 ; 输入的位率已在(BITRT)范围内均分
                    0233 ; 在出口：字存在累加器内
                    0234 ; 使用的寄存器是：A, B, H
                    0235 ; 允许中断且CTC1按位率中断
0758    21FE07      0236  INCHR:LD       HL, CTC3L
075B    7C          0237         LD      A, H
075C    ED47        0238         LD      I, A           ; 设置I寄存器
075E    7D          0239         LD      A, L
075F    D384        0240         OUT     (CTC0)         ; CTC中断矢量
0761    0608        0241 INCHR1:LD       B, 08H         ; 一个字的位计数值
0763    FB          0242         EI
0764    3E00        0243         LD      A, 0
0766    2600        0244         LD      H, 0           ; 清字
0768    DB90        0245 INCHRIA:IN      A, (KBSEL)     ; 输入数据
076A    CB7F        0246         BIT     7, A
076C    20FA        0247         JR      NZ, INCHIA - $ ; 为找到起始位“0”而循环返回
076E    3EA5        0248         LD      A, 0A5H        ; 设中断，定标系数为256
0770    D387        0249         OUT     (CTC3), A      ; CTC3控制字
```

```
地址    目的码      语句序号        源语句
0772    3E0D        0250            LD      A, 0DH
0774    D387        0251            OUT     (CTC3), A       ; 给出 CTC3时间常数，得位宽(时间)
                                                              的一半，即在每位的中点读入数据
0776    3EA5        0252            LD      A, 0A5H
0778    D387        0253            OUT     (CTC3)A
077A    3E1A        0254            LD      A, 01AH
077C    D387        0255            OUT     (CTC3), A       ; 下一个时间常数按全位宽度
077E    76          0256            HALT
                    0257 ; ※※※※※※※※※※※※※※※※※※※※
077F    CB7F        0258            BIT     7, A
0781    2014        0259            JR      NZ, INCHR3 - $  ; 再读一次,若是假启动位,则重新开始
0783    76          0260 INCHR2: HALT                       ; 等待字符的第一位
                    0261 ; ※※※※※※※※※※※※※※※※※※※※
0784    E680        0262            AND     80H             ; 屏蔽掉其它位输入
0786    B4          0263            OR      H
0787    67          0264            LD      H, A            ; 保留
0788    1018        0265            DJNZ    INCHR5 - $
078A    CB7F        0266            BIT     7, A
078C    2809        0267            JR      Z, INCHR3 - $   ; 甄别错误，重新启动
078E    F3          0268            DI
078F    3E03        0269            LD      A, 03H
0791    D387        0270            OUT     (CTC3), A       ; CTC 复位
0793    7C          0271            LD      A, H
0794    E67F        0272            AND     7FH             ; 屏蔽掉启始位
0796    C9          0273            RET
```

```
地址    目的码      语句序号    源语句
0797    3E03        0274 INCHR3: LD      A, 03H
0799    D387        0275         OUT     (CTC3), A      ; CTC3 复位，甄别错误
079B    18C4        0276         JR      INCHR1 - $     ; 为接收另外的字符循环返回
                    0277 ; 在输入期间 CTC 中断服务程序
079D    DB90        0278 INCHR4: IN      A, (KBSEL)     ; 输入数据
079F    FB          0279         EI
07A0    ED4D        0280         RETI
07A2    CB0C        0281 INCHR5: RRC     H
07A4    18DD        0282         JR      INCHR2 - $     ; 等待下一个
                    0283 ; ※※※※※※※※※※※※赋值表※※※※※※※※※※※※
                    0284 ; CTC 读为取减一计数器的值，CTC 的写为设置计数器
>0084               0285 CTC0:   EQU     84H            ; 为用户使用
>0085               0286 CTC1:   EQU     85H            ; 音频磁带—信息转储
>0086               0287 CTC2:   EQU     86H            ; 单步和 EPROM 编程器
>0087               0288 CTC3:   EQU     87H            ; 音频磁带—磁带输入
>0090               0289 KBSEL:  EQU     90H            ; 磁带数据
                    0290 ; ※※※七段显示字形
07A6    40          0291 SEGPT   DEFB    40H            ; 0
07A7    79          0292         DEFB    79H            ; 1
07A8    24          0293         DEFB    24H            ; 2
07A9    30          0294         DEFB    30H            ; 3
07AA    19          0295         DEFB    19H            ; 4
07AB    12          0296         DEFB    12H            ; 5
07AC    02          0297         DEFB    02H            ; 6
07AD    78          0298         DEFB    78H            ; 7
```

地址	目的码	语句序号	源语句		
07AE	00	0299	DEFB	00H	; 8
07AF	18	0300	DEFB	18H	; 9
07B0	08	0301	DEFB	08H	; A
07B1	03	0302	DEFB	03H	; b
07B2	46	0303	DEFB	46H	; C
07B3	21	0304	DEFB	21H	; d
07B4	06	0305	DEFB	06H	; E
07B5	0E	0306	DEFB	0EH	; F
70B6	7F	0307	DEFB	7FH	; 空
07B7	3F	0308	DEFB	3FH	; 提示符“—”
07B8	7D	0309	DEFB	7DH	; 撇号标记“，”
		0310 ; ※※※查找键值表			
07B9	FF	0311 KYTBL:	DEFB	0FFH	; 0 B=01, A=0F
07BA	EF	0312	DEFB	0EFH	; 1 B=02, A=0F
07BB	F7	0313	DEFB	0F7H	; 2 B=02, A=17
07BC	FB	0314	DEFB	0FBH	; 3 B=02, A=1B
07BD	DF	0315	DEFB	0DFH	; 4 B=04, A=0F
07BE	E7	0316	DEFB	0E7H	; 5 B=04, A=17
07BF	EB	0317	DEFB	0EBH	; 6 B=04, A=1B
07C0	CF	0318	DEFB	0CFH	; 7 B=08, A=0F
07C1	D7	0319	DEFB	0D7H	; 8 B=08, A=17
07C2	DB	0320	DEFB	0DBH	; 9 B=08, A=1B
07C3	DD	0321	DEFB	0DDH	; A B=08, A=1D
07C4	ED	0322	DEFB	0EDH	; B B=04, A=1D
07C5	FD	0323	DEFB	0FDH	; C B=02, A=1D
07C6	0D	0324	DEFB	00DH	; D B=01, A=1D

```
地址    目的码      语句序号    源 语 句
07C7    0B          0325        DEFB        00BH        ; E  B=01, A=1B
07C8    07          0326        DEFB        07H         ; F  B=01, A=17
07C9    0E          0327        DEFB        0EH         ; EXEC      B=01, A=1E
07CA    FE          0328        DEFB        0FEH        ; SS        B=02, A=1E
07CB    EE          0329        DEFB        0EEH        ; MON       B=04, A=1E
07CC    DE          0330        DEFB        0DEH        ; NEXT      B=08, A=1E
07CD    CD          0331        DEFB        0CDH        ; REG'DISP  B=10, A=1D
07CE    CB          0332        DEFB        0CBH        ; REG DISP  B=10, A=1B
07CF    C7          0333        DEFB        0C7H        ; PORT EXAM B=10, A=17
07D0    BF          0334        DEFB        0BFH        ; MEM EXAM  B=10, A=0F
07D1    BD          0335        DEFB        0BDH        ; BP        B=20, A=1D
07D2    BB          0336        DEFB        0BBH        ; PUNCH     B=20, A=1B
07D3    B7          0337        DEFB        0B7H        ; LOAD      B=20H,A=17
07D4    AF          0338        DEFB        0AFH        ; PROG      B=20, A=0F
                    0339 ; ※※※寄存器表—相应键值与在用户寄存器存放区内位置的对应
07D5    19          0340 REGTB: DEFB        19H         ; 键0 不显示
07D6    02          0341        DEFB        02H         ; 键1 =PC
07D7    02          0342        DEFB        02H         ; 键2 =SP
07D8    0C          0343        DEFB        0CH         ; 键3 =IFF 显示用户IFF2
07D9    16          0344        DEFB        16H         ; 键4 =IX
07DA    18          0345        DEFB        18H         ; 键5 =IY
07DB    0B          0346        DEFB        0BH         ; 键6 =I
07DC    09          0347        DEFB        09H         ; 键7 =H
07DD    0A          0348        DEFB        0AH         ; 键8 =L
07DE    19          0349        DEFB        19H         ; 键9 不显示
```

地址	目的码	语句序号	源 语 句		
07DF	03	0350	DEFB	03 H	; 键 A = A
07E0	05	0351	DEFB	05 H	; 键 B = B
07E1	06	0352	DEFB	06 H	; 键 C = C
07E2	07	0353	DEFB	07 H	; 键 D = D
07E3	08	0354	DEFB	08 H	; 键 E = E
07E4	04	0355	DEFB	04 H	; 键 F = F
		0356 ; ※※※辅助寄存器组			
07E5	19	0357 REGTBP:	DEFB	19 H	; 键 0 不显示
07E6	19	0358	DEFB	19 H	; 键 1 不显示
07E7	19	0359	DEFB	19 H	; 键 2 不显示
07E8	19	0360	DEFB	19 H	; 键 3 不显示
07E9	19	0361	DEFB	19 H	; 键 4 不显示
07EA	19	0362	DEFB	19 H	; 键 5 不显示
07EB	19	0363	DEFB	19 H	; 键 6 不显示
07EC	13	0364	DEFB	13 H	; 键 7 = H′
07ED	14	0365	DEFB	14 H	; 键 8 = L′
07EE	19	0366	DEFB	19 H	; 键 9 不显示
07EF	0D	0367	DEFB	0DH	; 键 A = A′
07F0	0F	0368	DEFB	0FH	; 键 B = B′
07F1	10	0369	DEFB	10 H	; 键 C = C′
07F2	11	0370	DEFB	11 H	; 键 D = D′
07F3	12	0371	DEFB	12 H	; 键 E = E′
07F4	0E	0372	DEFB	0E H	; 键 F = F′
		0373	END		

```
地址    目的码      语句序号      源语句
                    0002          程序名称  UTILR
                    0003 ; 实用RAM和常数
                    0004 ; 版本 1.2  5/13/78
>07F8               0005          ORG       07F8H
                    0006          PSECT     ABS
                    0007          GLOBAL    CTC3L
                    0008          GLOBAL    CTC1P
                    0009          GLOBAL    OTCHR6
                    0010          GLOBAL    INCHR4
                    0011 ; CTC 中断矢量表
078F    D62F        0012 CTCO:    DEFW      CTC0V     ; 分配地址到 RAM
07FA    3207        0013 CTC1P:   DEFW      OTCHR6    ; 转储中断矢量FAH
>07FC               0014 CTC2:    DEFS      2         ; 不用中断
07FE                0015 CTC3L:   DEFW      INCHR4    ; 输入中断矢量 FEH
>2FC0   9D07        0016          ORG       23C0H
                    0017          GLOBAL    DIG2
                    0018          GLOBAL    DIG 4
                    0019          GLOBAL    DIG 8
                    0020          GLOBAL    BPTAB
                    0021          GLOBAL    SSFLG
                    0022          GLOBAL    UIF
                    0023          GLOBAL    DISMEM
                    0024          GLOBAL    DSMEM1
                    0025          GLOBAL    DSMEM2
                    0026          GLOBAL    DSMEM3
```

地址	目的码	语句序号	源语句	
		0027	GLOBAL	DSMEM4
		0028	GLOBAL	DSMEM5
		0029	GLOBAL	DSMEM6
		0030	GLOBAL	DSMEM7
		0031	GLOBAL	STKPT
		0032	GLOBAL	STKPT1
		0033	GLOBAL	RST16
		0034	GLOBAL	RST24
		0035	GLOBAL	RST32
		0036	GLOBAL	RST40
		0037	GLOBAL	RST48
		0038	GLOBAL	RST56
		0039	GLOBAL	KEYPTR
		0040	GLOBAL	FLG24
		0041	GLOBAL	RFLG
		0042	GLOBAL	ARFLG
		0043	GLOBAL	BFLG
		0044	GLOBAL	PFLG
		0045	GLOBAL	MFLG
		0046	GLOBAL	PRFLG
		0047	GLOBAL	PUNHSH
		0048	GLOBAL	PUNHSL
		0049	GLOBAL	PUNHEH
		0050	GLOBAL	PUNHEL
		0051	; ※※※ RAM变量	

```
地址    目的码      语句序号    源语句
>2FC0               0052 PUNHSH:DEFS    1           ;转储起始地址高字节
>2FC1               0053 PUNHSL:DEFS    1           ;转储起始地址低字节
>2FC2               0054 PUNHEH:DEFS    1           ;转储结束地址高字节
>2FC3               0055 PUNHEL:DEFS    1           ;转储结束地址低字节
>2FC4               0056 RST16: DEFS    3           ;为处理RST16—56用户插入转移指令
>2FC7               0057 RST24: DEFS    3           ;
>2FCA               0058 RST32: DEFS    3           ;
>2FCD               0059 RST40: DEFS    3           ;
>2FD0               0060 RST48: DEFS    3           ;
>2FD3               0061 RST56: DEES    3
                    0062 ;※※※为CTC0 中断用户插入转移指令
>2FD6               0063 CTC0V: DEFS    3           ;CTCO 中断将指向此处
                    0064 ;※※※ CTC3 中断用
>2FD9               0065 FLG24: DEFS    1           ;转储逻辑1时置1
>2FDA               0066 PRFLG: DEFS    1           ;EPROM 编程标志
>2FDB               0067 KEYPTR:DEFS    2           ;为了写入下一个显示缓冲区单元而设的指针
>2FDD               0068 UIF:   DEFS    1           ;用户 IFF2
>2FDE               0069 PFLG:  DEFS    1           ;口检查标志
>2FDF               0070 RFLG:  DEFS    1           ;寄存器检查标志
>2FE0               0071 ARFLG: DEFS    1           ;辅助寄存器检查标志
>2FE2               0072 MFLG:  DEFS    1           ;存储器检查标志
>2FE3               0073 STKPT: DEFS    1           ;用户栈指针的高字节
>2FE3               0074 STKPT1:DEFS    1           ;用户栈指针的低字节
                    0075 ;※※※断点表的格式为:
                    0076 ;设置断点处的地址为两个字节，跟着一个字节即是被 RST8 指令替换掉的用户
```

```
地址     目的码      语句序号      源语句
                     0077 ; ※※※操作码的第一字节
>2FE4                0078 BPTAB:  DEFS   15      ; 断点地址和移来的操作码
>2FF3                0079 SSFLG:  DEFS   1       ; 单步方式标志
>2FF4                0080 BFLG    DEFS   1       ; 已设置的断点数目
>2FF5                0081 DIG2    DEFS   1       ; 已输入2位数字的标志
>2FF6                0082 DIG4    DEFS   1       ; 已输入4位数字的标志
>2FF7                0083 DISMEM  DEFS   1       ; 存储器显示缓冲区
>2FF8                0084 DSMEM1  DEFS   1
>2FF9                0085 DSMEM2  DEFS   1
>2FFA                0086 DSMEM3  DEFS   1
>2FFB                0087 DSMEM4  DEFS   1
>2FFC                0088 DSMEM5  DEFS   1
>2FFD                0089 DSMEM6  DEFS   1
>2FFE                0090 DSMEM7  DEFS   1
                     0091         END
```

附 录 四

实 验 指 示 书

实验一　熟悉键盘操作

一、实验目的：熟悉按键操作及 TPBUG

二、实验设备：TP801 单板机一台

5 V 稳压电源一台

三、实验准备工作：

① 启动稳压电源，将其输出电压调到 5 V ± 5%；

② 将单板机的电源线“+”端、“地”端分别接到稳压电源的“+”、“-”端。

四、实验内容：

本实验通过执行下面的程序，一方面熟悉通过键盘命令输入程序的操作，另一方面通过显示器的显示，部分地观察程序的执行，并熟悉 SINGLE STEP, MON, BREAK POINT……等命令键的使用，简单了解调试程序的过程。

```
                        ORG     2000 H
2000    3E  AA          LD      A, 0AAH
2002    06  BB          LD      B, 0BBH
2004    78              LD      A, B
2005    0E  CC          LD      C, 0CCH
2007    79              LD      A, C
2008    3E  AA          LD      A, 0AAH
200A    76              HALT
```

五、实验步骤：

1. 通过键盘操作，将 ORG 2000 H 程序送入单板机。（被操作的按键均标以横线，TPBUG 响应均以方框表示显示器显示。）

按　键	显　示	
RESET (S1)	—	
2 0 0 0 MEM EXAM	2 0 0 0	× ×
3 E	2 0 0 0	3 E
NEXT A A	2 0 0 1	A A
NEXT 0 6	2 0 0 2	0 6
NEXT B B	2 0 0 3	b b

按　键	显　示	
NEXT　7　8	2　0　0　4	7　8
NEXT　0　E	2　0　0　5	0　E
NEXT　C　C	2　0　0　6	C　C
NEXT　7　9	3　0　0　7	7　9
NEXT　3　E	2　0　0　8	3　E
NEXT　A　A	2　0　0　9	A　A
NEXT　7　6	2　0　0　A	7　6
MON	—	

2．在未执行程序前，检查有关寄存器内容：

A　REG　EXAM	A	×　×
MON	—	
B　REG　EXAM	b	×　×
MON	—	
C　REG　EXAM	C	×　×
MON	—	

（注：在××符号边上记上这时数据，以便和步骤 3.对比）

3．用 SINGLE STEP 键，单步执行程序。首先调整程序计数器 PC，使 PC 指向程序首地址 2000 H，然后才按动 SINGLE STEP 键，注意观察显示器的变化。

当显示器显示地址为 200AH 时，按下 MON 键，检查并记录有关寄存器内容，与步骤 2.进行比较。

MON	—	
A　REG　EXAM	A	
MON	—	
B　REG　EXAM	b	

MON	—	
C REG EXAM	C	
MON	—	
PC REG EXAM	1	
MON	—	

4．用 BREAK POINT 键，设置断点。

在 2002、2005、2008、200 A 单元设置断点，并用 EXEC 键执行程序，注意观察显示器的变化，比较与 SINGLE STEP 方式执行程序有什么不同。

MON	—	
2 0 0 2 BREAK POINT	2 0 0 2	
MON	—	
2 0 0 5 BREAK POINT	2 0 0 5	
MON	—	
2 0 0 8 BREAK POINT	2 0 0 8	
MON	—	
2 0 0 A BREAK POINT	2 0 0 A	
MON	—	

然后输入首地址2000 H，连续按动 EXEC 键执行程序。

当执行程序到显示器显示 200 A H 地址时，停止按动 EXEC 键，与步骤 3．一样，检查有关寄存器内容：

A	A	
B	b	
C	C	
PC	1	

5．清除断点。用 EXEC 键连续执行程序。

MON

2 0 0 0 EXEC

问题：

(1) 为什么显示器不显示？若要显示器自动显示标志“—”，应如何修改程序？

(2) 检查 PC 内容，对比三种方式执行程序，PC 内容为什么有异、同？

(3) 清除断点有几种方式？

(4) 思考：当采用 SINGLE STEP 方式执行程序，若要显示 DATA 的显示器循环不止的显示 AA，BB，CC，如何修改本程序？

实验二　相对转移中偏移量的计算

一、实验目的：熟悉单板机的性能，了解 Z 80 的指令。

在 Z 80 系统中，有许多指令需要计算相对转移的偏移量，而这些偏移量是以二进制的补码表示的。在本系统中，具有自动计算并填写相对偏移量的功能。

二、实验设备：同实验一

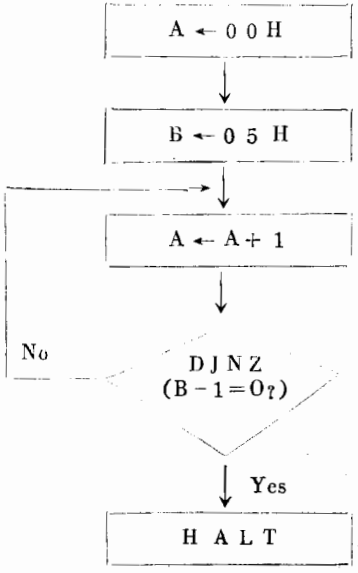
三、实验准备工作：同实验一

四、实验内容：

相 对 地 址 的 计 算

地　址	机 器 码		操 作 码		
2000	3E00		LD	A	00 H；
2002	0605		LD	B	05 H；
2004	3 C	→LOOP：	IN	A	
2005	10？		DJNZ,	LOOP-$	
2007	76		HALT		

框　图



五、实验步骤：

1．操作键盘，输入程序。

RESET	—	
2 0 0 0 MEM EXAM 3 E	2 0 0 0	3 E
MEXT 0 0	2 0 0 1	0 0
MEXT 0 6	2 0 0 2	0 6
NEXT 0 5	2 0 0 3	0 5
NEXT 3 C	2 0 0 4	3 C
NEXT 1 0	2 0 0 5	1 0
NEXT	2 0 0 6	× ×
NEXT 7 6	2 0 0 7	7 6
MON	—	

注意：在2006单元不输入内容，因为要将计算的相对偏移量填入此单元。

H REG EXAM 2 0	7	2 0
MON L REG EXAM 0 4	8	0 4

（說明：标号 LOOP 单元目的地址2004H装入 HL 寄存器对。）

MON D REG EXAM 2 0	d	2 0
MON E REG EXAM 0 5	E	0 5

（說明：现行 $ 单元源地址 2005H 装入 DE 寄存器对。）

MON 0 0 C 0 EXEC	F F F d	

（TPBUG 存有执行相对地址計算子程序，它的起始地址是 00C0H。FF 表示向上偏移，如 00 表示向下偏移，Fd 为偏移值。）

MON 2 0 0 6 MEM EXAM	2 0 0 6	F d

（检查2006，看偏移量是否已装入存储器。）

2．用 SINGLE STEP 单步键来检序程序的执行过程：

MON 1 REG EXAM	1 × ×	× ×
	使PC 2000H	
2 0 0 0	1 2 0	0 0
SINGLE STEP	2 0 0 2	0 0
SINGLE STEP	2 0 0 4	0 0
SINGLE STEP	2 0 0 5	0 1
	第一次增A	
SINGLE STEP	2 0 0 4	0 1
	循环返回	
SINGLE STEP	2 0 0 5	0 2
	第二次增A	
SINGLE STEP	2 0 0 4	0 2
SINGLE STEP	2 0 0 5	0 3
	第三次增 A	
SINGLE STEP	2 0 0 4	0 3
SINGLE STEP	2 0 0 5	0 4
	第四次增 A	
SINGLE STEP	2 0 0 4	0 4
SINGLE STEP	2 0 0 5	0 5
	第五次增 A	
		B 被减到 1
MON B REG EXAM	6	0 1
MON SINGLE STEP	2 0 0 7	0 5

(B减量，若非零則轉移，現在B 为零，所以 PC→2007H)

选作题：

下例程序使字符“8”在显示器内从右方移到左方

```
2000  3E  00              LD      A, 00H,
2002  D3  88              OUT     (88H), A
2004  3E  01              LD      A, 01H
2006  D3  8C      LOOP:   OUT     (8CH), A
2008  CD  4F  06          CALL    D20MS
200B  CD  4F  06          CALL    D20MS
200E  CD  4F  06          CALL    D20MS
2011  CD  4F  06          CALL    D20MS
2014  CD  4F  06          CALL    D20MS
2017  07                  RLCA
2018  18  EC              JR      LOOP-$
```

注："CALL D20MS"：调用管理程序中 D20MS 子程序，"D20MS"表示延时 20 ms 意思。

思考：① 能否将上面选作题稍作修改，可以很方便 地控 制字符在显示器上停留时间？

② 若要显示别的字符，应如何修改程序？

实验三　　程序设计（一）

一、实验目的：利用已掌握的 Z 80 指 令 系 统，进行一些简单程序设计，并通过实验，熟悉调试程序的过程。

二、实验准备：同实验一

三、实验准备工作：同实验一

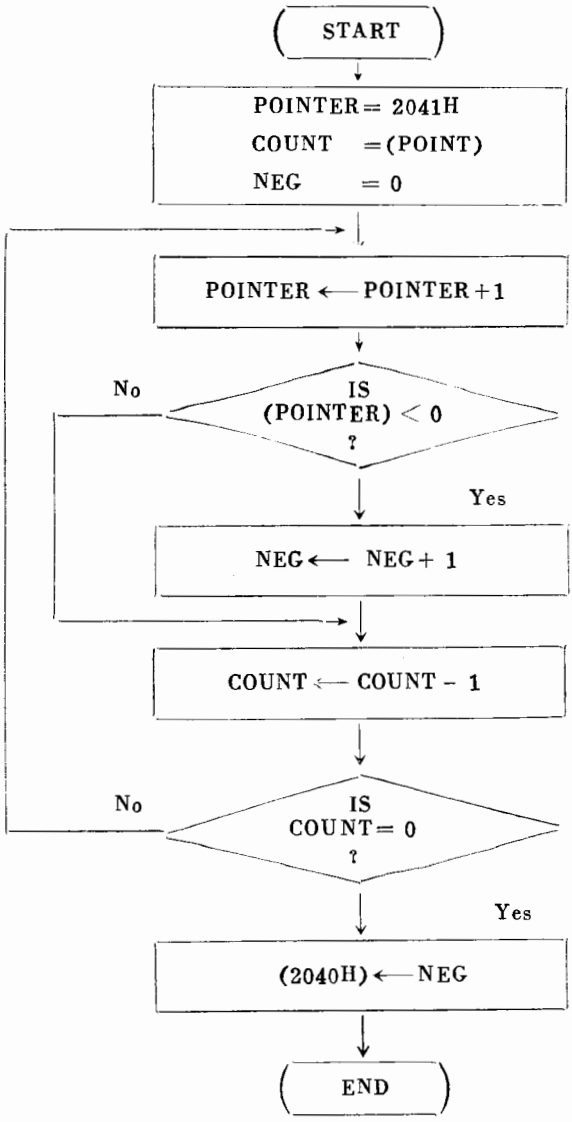
四、实验内容：

1．在一个有正、负数的数据块中，找出负数的个数。假定每个数 都 是 8 位二进制数，并将每个数的第 7 位(b 7)作为符号位。数据块的长度存放在 2041 H 单元，数据从 2042 H 单元开始存放，负数的个数存放在 2040 H 单元。

例如：

```
(2040H) =  ?
(2041H) = 06 H
(2042H) = 53 H
(2043H) = F4 H
(2044H) = 85 H
(2045H) = 30 H
(2046H) = 58 H
(2047H) = 3A H
```

流　　程　　图



请大家查找指令表，将机器码填入。

```
                ORG     2000
2000            LD      HL, 2041 H
                LD      B, (HL)
                LD      C, 00
LOOP 1 ;        INC     HL
                LD      A, (HL)
                AND     A
                JR      P, LOOP 2
```

```
            INC     C
LOOP2 :     DJ      NZ, LOOP1
            LD      A, C
            LD      (2040H), A
            HALT
```

注：AND A 这条指令是用来判别 A 的内容的第 7 位 B_7 是 0 还是 1（即 A 的内容是正数还是负数），当 $B_7=1$ 时，标志寄存器 F 的 S = 1；反之，S = 0，请问，能完成这种功能的指令还有那些？请按你的意见，修改上面程序，并通过实验，验证计算机执行程序结果如何？

2．在一个数据块中找出最大数。假定这些数都是 8 位二进制数，且不带符号位的正整数。数据块的长度存放在 2041H 单元，数据块从 2042H 单元开始存放，结果找出的最大数存放在 2040H 单元。

例如：

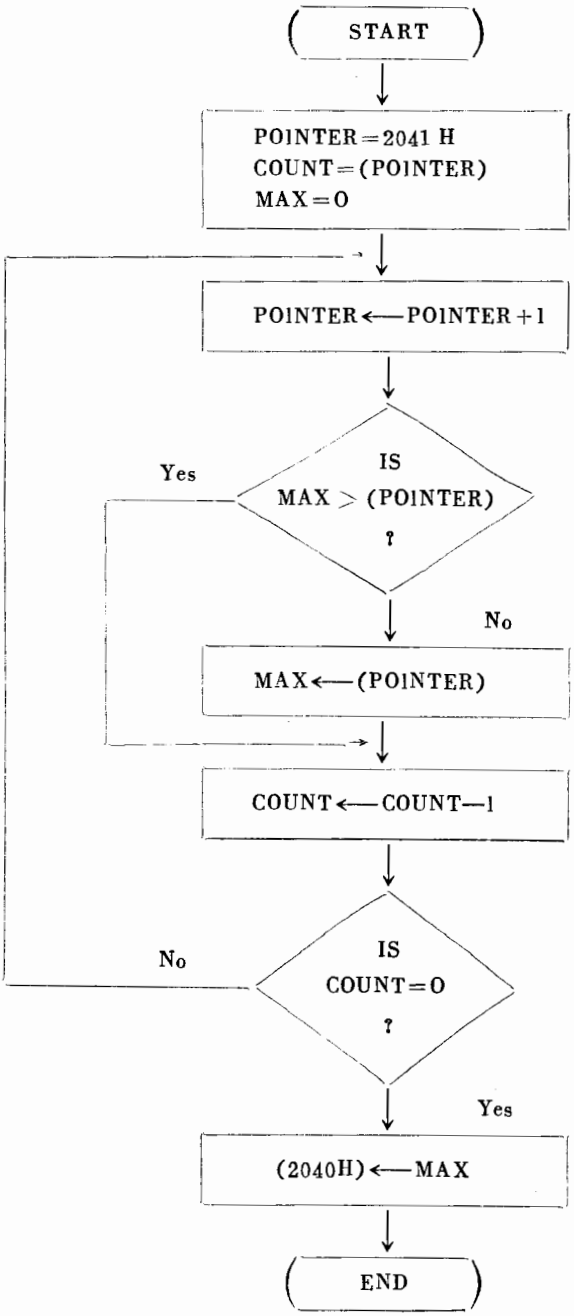
(2040H) = ?
(2041H) = 06 H
(2042H) = 57 H
(2043H) = 89 H
(2044H) = A7 H
(2045H) = 17 H
(2046H) = E8 H
(2047H) = 95 H

请大家查找指令表，将对应机器码填入下面程序

```
                ORG     2000H
2000            LD      HL, 2041H
                LD      B, (HL)
                SUB     A
LOOP1:          INC     HL
                CP      (HL)
                JR      NC, LOOP2
                LD      A, (HL)
LOOP2:          DJ NZ , LOOP1
                LD      (2040)H), A
                HALT
```

按下 MON，检查 2040 H 单元内容。

流　　程　　图


START
POINTER＝2041 H
COUNT＝(POINTER)
MAX＝0
POINTER←POINTER＋1
IS
MAX ＞ (POINTER)
?
Yes
No
MAX←(POINTER)
COUNT←COUNT—1
IS
COUNT＝0
?
No
Yes
(2040H)←MAX
END

实验四　　程序设计（二）

一、实验目的：同实验三

二、实验设备：同实验一

三、实验准备工作：同实验一

四、实验内容：

1．8位二进制数求和　　38H＋55H＋26H＋12H＋23H＝？

5个八位数相加，这组数的个数放在(2041 H)单元，和存在(2040 H)单元。这组数从(2042 H)单元开始存放(假定和数也是一个八位数，暂不进行和数大于八位数的计算)

数据：　　(2041 H) = 05 H　　(2044 H) = 26 H

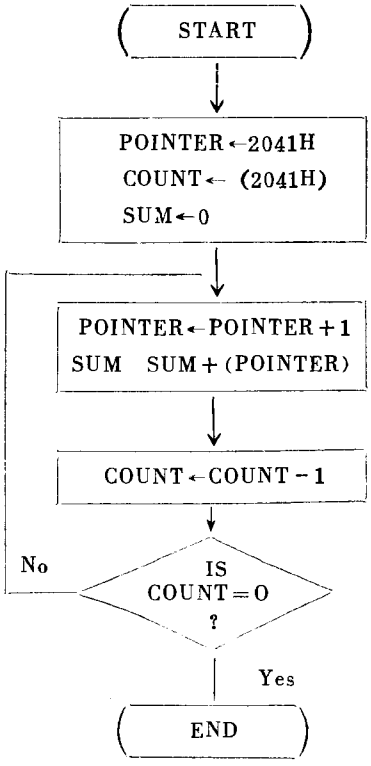
　　(2042 H) = 38 H　　(2045 H) = 12 H

　　(2043 H) = 55 H　　(2046 H) = 23 H

和数：　　(2040 H) = E 8 H

根据下面流程图，请大家进行程序设计，然后在计算机上进行调试。

流　　程　　图



输入程序，从首地址2000H开始执行，按MON键，检查2040 H地址单元内容，即是和数。

2．8位二进制数求和，和数是16位二进制数

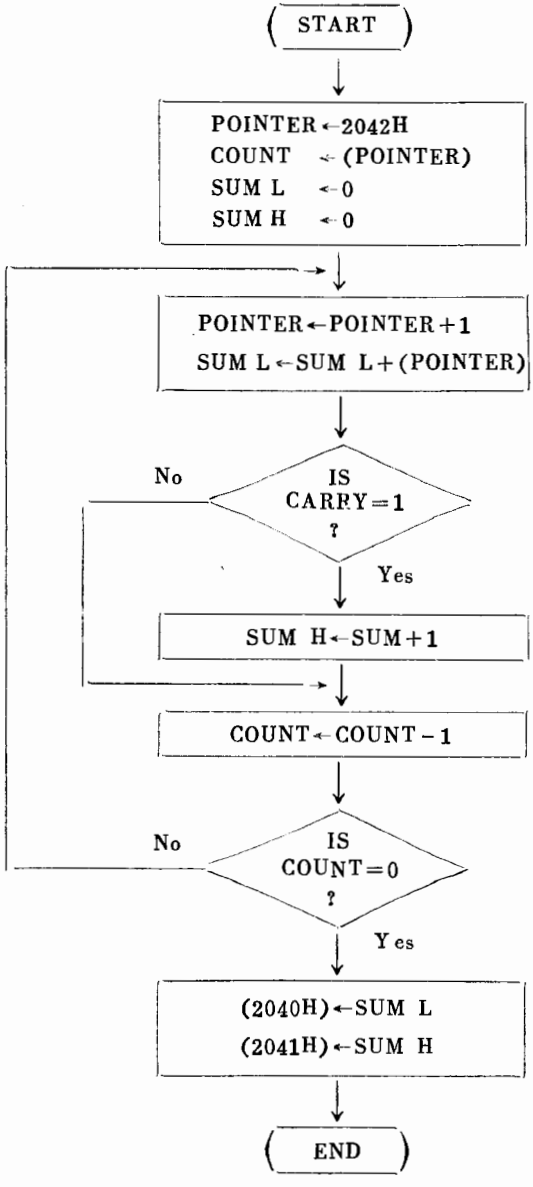
C8 H + FAH + 96H + F 9 H + E 8 H = ?

和的低位存放（2040 H）单元，高位存放（2041 H）单元，数组的个数存放（2042H）单元，数据从（2043 H）开始存放

```
;和数      (2040 H) = ?
           (2041 H) = ?
;数据      (2042 H) = 05 H          (2045 H) = 96H
           (2043 H) = C8H           (2046 H) = F9H
           (2044 H) = FAH           (2047 H) = E8H
```

输入程序，从首地址 2000 H 开始执行，按 MON 键， 检查2040 H 和 2041 H地址单元内容，即是和数。

流　　程　　图



选作题：将 BCD 码转换为二进制码

本程序是将存贮在 2040 H 和 2041 H 单元的 BCD 码 29D 转换成二进制代码，存放在 2042 H 单元。

```
        ; 数据
        (2040 H) = 02
        (2041 H) = 09
        (2042 H) = 1 DH = 29 Decimal
                   ORG 2000
                   LD        HL, 2040 H
                   LD        A, (HL)
                   ADD       A, A
                   LD        B, A
                   ADD       A, A
                   ADD       A, A
                   ADD       A, B
                   INC       HL
                   ADD       A, (HL)
                   INC       HL
                   LD        (HL), A
                   HALT
```