

Z80单板计算机监控 程序分析

余金龙 编

3.87138
4585

电讯工程学院出版社

DAN BAN JI SUAN JI JIAN KONG CHENG XU FEN XI

Z80单板计算机监控程序分析

余金龙 编

西北电讯工程学院出版社

1984

内 容 简 介

本书对Z80系列的单板计算机的软件——监控程序作了详细的分析，给出了TPBUG-A的全部程序清单(用汇编语言写的)和每个程序块的详细框图及其说明。本书的特点是分析细致、层次分明、解释周详。

本书既适合于作为大专院校师生学习和掌握微机使用的教材，也适合于作为广大科技人员、生产人员实际应用Z80系列单板计算机软件的参考材料。

Z80单板计算机监控程序分析

余金龙 编

西北电讯工程学院出版社出版

西北电讯工程学院印刷厂印刷

开本787×1092 1/16 印张4 4/16 印刷字数98千字
1984年11月第一版 1984年11月第一次印刷 印数13,000

书号：15322·14

定价：0.65元

目 录

第一章 概述	1
1.1 引言.....	1
1.2 监控程序的作用.....	1
1.3 TPBUG—A 监控程序的特点和功能	1
第二章 TPBUG—A 监控程序的分析	3
2.1 几点说明	3
2.2 总体结构.....	3
2.3 初始化和部分初始化程序.....	4
2.4 显示程序.....	4
2.5 键盘扫描程序.....	5
2.6 键盘译码程序.....	5
2.7 数字键处理程序.....	5
2.8 MON和MON' 命令执行程序	5
2.9 NEXT命令执行程序.....	5
2.10 LAST命令执行程序.....	6
2.11 EXEC命令执行程序.....	6
2.12 STEP单步命令执行程序	6
2.13 自定义命令执行程序.....	6
2.14 PORT命令执行程序	7
2.15 REG和REG' 命令执行程序.....	7
2.16 MEM命令执行程序	7
2.17 BP命令执行程序	7
2.18 DISP命令执行程序	7
2.19 DUMP命令执行程序	8
2.20 LOAD命令执行程序.....	9
2.21 PROM命令执行程序	10
2.22 MOVE命令执行程序	10
2.23 子程序.....	10
第三章 监控程序的框图	12
附录: TPBUG—A 监控程序清单	27
结束语.....	63
参考书和资料.....	63

第一章 概 述

1.1 引言

自七十年代以来,微处理器与微型计算机的发展十分迅速。由于应用了大规模和超大规模集成电路技术,所以它突出的优点是体积小,价格便宜,性能可靠,使用方便,因而应用极为广泛。已经深入到工业、农业、交通、国防、文教、科研,以及日常生活等各个领域。

几年来,我国引进了大量的单板机和微型计算机系统,特别是 Z80 单板机的数量,更是屈指首位。Z80 单板机作为“智能”部件,可用于生产过程控制,各种仪器和仪表的自动测试,数据处理以及检测、管理、数控、数据通讯、信号控制等。

如果想要把 Z80 单板机很好地应用到各个领域,则必须深入地熟悉和掌握 Z80 单板机的灵魂——监控程序。只有把监控程序吃透了,才能使 Z80 单板机运用自如,甚至能够根据应用的需要编制出新的监控程序。我国各厂家引进的八位 Z80 单板机监控程序大体相同,它们的设计思想和程序结构基本一致。在本书中我们详细地分析了 TP-801 单板机的监控程序 TPBUG-A,给出了全部的程序清单(用汇编语言写的)和每个程序块的详细框图和说明。便于工程技术人员和大专院校师生学习。

1.2 监控程序的作用

单板机在它的硬件和软件(监控程序)的共同作用下,才能实现全部功能。任何一个单板机在加电后要能够正常启动,扫描键盘,控制输入装置输入信息,输入用户程序,实现各种功能,以及运行和输出给外部世界等等,都必须要有个监控程序。微处理机正是由于利用事先储存在 ROM 中的监控程序而成为快速可用的工具,取得了强有力和完美精巧的地位。单板机作为“智能”部件所具有的全部功能,都是通过监控程序实现控制的。因此,监控程序是单板机的灵魂。没有监控程序,单板机便是一台死机器,毫无用处。

1.3 TPBUG-A 监控程序的特点和功能

目前引进的八位 Z80 单板机的监控程序大体相同。不论是北京工业大学引进的 TP801-Z80 单板机的 TPBUG 监控程序,还是福建省电子计算机厂引进的 DBJ-Z80 单板机的 ZBUG 监控程序,其程序结构基本一致。而 TPBUG-A 监控程序是北京工业大学在原 TPBUG 监控程序基础上作了一定的改进后,配备在 TP801-Z80 单板机上的。在 TPBUG-A 监控程序的命令键中,有 8 个是双功能键。这样,监控程序的命令就由原来的 12 增加到 20 个,这是 TPBUG-A 较之 TPBUG 监控程序的主要改进之处,由于操作命令的增加,使单板机功能更加齐全,使用更加方便。

TPBUG-A 监控程序固化在一块 2K 字节的只读存储器 ROM 中。有了 TPBUG-A 监控程序,就可以通过十六进制的键盘向计算机输入机器语言的程序和数据,并可调试 Z80 程序;可以为用户提供相对转移指令的偏移量计算;可以用便宜的盒式磁带录音机作单板机的外存储器,将存储区中的程序转录到磁带中或将磁带中的信息装入到单板机的 RAM 中。

TPBUG-A还具有简易的开发功能：配有一个可编程2716/2758EPROM片子的EPROM编程器，用户可将程序存入 EPROM片中随机使用；具有设置和清除断点，单步执行指令以及显示和修改寄存器(包括辅助寄存器)内容的功能，为用户提供了先进的诊断能力。此外，TPBUG-A 监控程序还具有将RAM中的程序块移动的功能；具有显示和修改存储器和各 I/O 端口内容的功能；具有启动和连续执行用户程序的功能；具有显示下一个或上一个存储单元的内容，以供检查和修改的功能；具有中止现行程序的执行，中止或者退出当前的命令状态，使控制返回到监控程序初始化部份的功能。最后，还留有四个扩充功能键供用户使用。

总之，TPBUG-A 监控程序为 Z80 单板机的用户开发微机应用提供了强有力的手段。

第二章 TPBUG-A 监控程序的分析

2.1 几点说明

TPBUG-A 监控程序，固化在 2K 字节的 ROM 片中。在存储空间中，它占有的地址为 0000H—07FFH。整个监控程序的汇编语言程序清单见附录。

TP801-Z80 单板机的存储空间分配如下：

0000H—07FFH	存放 2K 字节的监控程序
0800H—0FFFH	为 2K 字节的 PROM ₁ ，可存放用户的应用程序
1000H—17FFH	为 2K 字节的 PROM ₂ ，可存放用户的应用程序。此外，监控程序还可对该插座中的 EPROM 进行写入。

1800H—1FFFH	没被使用。
-------------	-------

2000H—2FFFH	为 4K 字节的 RAM 存储区。
-------------	-------------------

3000H—FFFFH	该地址区没有被使用。
-------------	------------

监控程序还占用内存 RAM 中的 120 个单元作为堆栈工作区和存放变量的工作单元之用。

具体分配如下：

2F88H—2F9FH	监控堆栈区。
-------------	--------

2FA0H—2FB7H	用户寄存器存放区和用户堆栈工作区。
-------------	-------------------

2FB8H—2FFFH	监控程序的变量(标志、指针、断点)表和缓冲区。
-------------	-------------------------

单板机的外部设备口地址如下：

80H	为PIO A口的数据寄存器地址。
-----	------------------

81H	为PIO B口的数据寄存器地址。
-----	------------------

82H	为PIO A口的控制寄存器地址。
-----	------------------

83H	为PIO B口的控制寄存器地址。
-----	------------------

84H	为CTC0 通道口地址。
-----	--------------

85H	为CTC1 通道口地址。
-----	--------------

86H	为CTC2 通道口地址。
-----	--------------

87H	为CTC3 通道口地址。
-----	--------------

88H	为显示器段锁寄存器地址。
-----	--------------

8CH	为显示器数位锁寄存器地址。
-----	---------------

90H	为键盘输入端口地址。
-----	------------

以上这些地址，在 TPBUG-A 监控程序中均要用到。

2.2 总体结构

TPBUG-A 监控程序由四个主要程序段组成：

(1) 初始化，显示和键盘分析程序(0000H—0230H)

- (2) 键盘动作程序, 即命令键和数字键处理程序(0231H-0633H)
- (3) 实用子程序(0634H-07A5H)
- (4) 表格(07A6H-07FFH)

TPBUG-A 监控程序的总框图, 如图 1 所示。

从 0000H 地址开始进入初始化, 为监控程序的正常工作作好准备。然后判断 S_2 开关, 若 $S_2 = 0$ 则转入 0800H 地址开始的用户程序, 若 $S_2 = 1$ 则执行 TPBUG-A 监控程序。经过从左到右的显示一遍后, 进入键盘扫描和译码程序, 判断是否有键按下以及找出那一个键按下, 从而决定程序转移的方向。如果按了数字键, 则进行数字键处理程序。如果按下的是命令键, 则转向该命令的执行程序。如果没有键按下, 则程序又返回到显示程序, 显示一轮后又进入键盘扫描程序, 搜索是否有键按下。所以, TPBUG-A 监控程序是不断地在显示和键盘扫描之间交替, 不断地搜索键盘, 呈现动态的等待状态, 一旦有键按下就进入该键所对应的执行程序, 这就是 TPBUG-A 监控程序的设计思想。

2.3 初始化和部分初始化程序(从 0000H 开始)

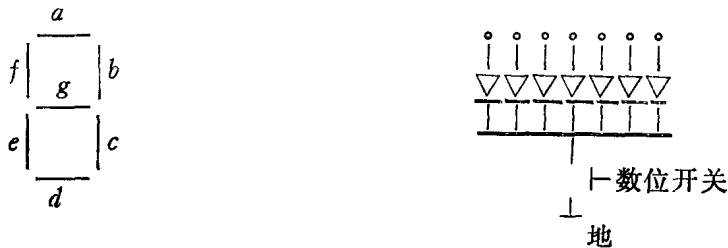
单板机接通电源后自动复位或按下复位按钮(RESET)都能强迫 Z80-CPU 复位, 并开始执行起始地址为 0000H 的监控程序的初始化程序。这时最左边显示器显示 TPBUG-A 监控程序的提示符“P”, 其余五个显示器皆为空白。表明监控程序已经启动, 正准备接受键盘的命令。

具体来说, 在初始化程序中, 首先设置用户堆栈和监控堆栈的栈底, 清除断点标志, 用户中断标志和各种工作方式标志。然后将 P 和空白显示符送显单元。如果开关 S_2 处于 PROM1 位置, 则监控程序执行完初始化程序后, 自动转向起始地址为 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。程序框图见图 2。

部分初始化是从 RESTR1(009CH 地址)开始的初始化程序。当中止或者退出当前命令状态, 而返回到监控程序初始化时, 则就是从部分初始化开始。

2.4 显示程序(从 00F4H 开始)

单板机是采用六个七段数字显示器, 每一段为一发光二极管, 其结构如下图所示。



七段数字显示器结构

通过数位锁存器(口地址为 8CH)输出, 来选通一位显示器工作。通过段锁存器(口地址为 88H)输出字形码, 使相应的二极管发光, 在被选通的那位显示器上显示出字符。

显示程序的功能，就是把存放在显示缓冲区前六个单元(DISMEM 至 DSMEM5)的数送到显示器上显示。

当要显示某个 16 进制数，首先把 IX 寄存器指向七段显示字形表的首地址，把从显缓单元中取出的数作为偏移量，查字形表，从中取出被显示的数所对应的字形码送段锁存器，输出到显示器上显示。

显示程序框图见图 3。

2.5 键盘扫描程序(从 0123H 开始)

首先粗扫描键盘矩阵，检查有无键闭合，若无键闭合则转显示程序。若有键闭合时，则逐行扫描，检查列值，直到找到闭合的键，然后转至键盘译码程序以确定此键值。

程序框图见图 4。

2.6 键盘译码程序(从 014FH 开始)

由所按键的行值与列值，合并成键码。再根据键码，在键值表中查找键值。当键值 < 10H，闭合的是数字键，则转数字键处理程序。当键值 > 10H，闭合的是命令键，则转到相应的命令键处理程序的入口地址，以完成该命令的功能。

程序框图见图 5。

2.7 数字键处理程序(从 0179H 开始)

把数字放到 KEYPTR 所指示的显示缓冲区中的下一个空单元中。若已输入两个数，置标志 DIG2。若已输入四个数，置标志 DIG4，并分别转显示程序显示。当输入八个数，调用修改子程序 ALTER。根据(2FDE)，(2FDF)，(2FE0)，(2FE1)单元中的检查标志，分别转至通道口内容检查的修改程序；寄存器，辅助寄存器内容检查的修改程序；或存储器单元内容的检查修改程序。

数字键处理程序的程序框图见图 6。

2.8 MON 和 MON' 命令执行程序(从 0097H, 0200H 开始)

MON 键有两种功能：第一，中止或者退出当前输入的命令状态和数据，使控制返回到监控程序的初始化部分。显示提示符“P”，等待下一个按键的输入。若按下的是八个双功键之一，则执行的是下档命令。第二，当正在执行用户程序时，按下 MON 键，将中止现行程序的执行。通过硬件产生一个非屏蔽中断，CPU 响应非屏蔽中断，转入 0066H 开始的非屏蔽中断服务程序。保护 CPU 寄存器内容，使控制返回到监控程序的初始化部分。这种功能在分析程序故障时很有用。

MON' 命令也是使控制返回到监控程序的初始化部分，所不同的是显示提示符“'”。执行的是双功键的上档命令。

程序框图见图 7、图 8。

2.9 NEXT 命令执行程序(从 02AEH 开始)

NEXT 键与存储器检查，通道检查和下一个 EPROM 编程错误等三种操作方式一起使

用。在存储器检查和通道检查方式中，NEXT 键用于选择下一个存储器单元或通道口，并自动显示这个单元或通道的内容。在 EPROM 的编程期间如果出现错误，NEXT 键可用于步进通过这个错误，而继续核对 EPROM 中的内容。

程序框图见图 9。

2.10 LAST 命令执行程序(从 02A9H 开始)

LAST 键是用来显示上一个存储器单元的地址和内容。只适用于存储器检查命令状态，对于其它命令状态使用 LAST 键，则按无意义处理并返回到监控程序初始化部分。

程序框图见图 9。

2.11 EXEC 命令执行程序(从 0231H 开始)

该命令用来连续执行 RAM，ROM 或 EPROM 中的程序。

如果要从程序的起始部分开始执行，应先输入表示执行程序起始地址的四个数字，然后按下 EXEC 键，即可从此地址开始执行程序。如果未输入地址，仅仅按下 EXEC 键，CPU 将从保存在“用户寄存器映象区”内的 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，此后又要求连续执行程序，则只需按下 EXEC 键即可。

在 EXEC 执行过程中，分两种情况：(1)无断点，则执行很简单。设置允许 MON 键中断，置 CTC2 为计数工作方式，以保证按下 MON 键时能产生 $\overline{\text{NMI}}$ 请求。然后把各个寄存器内容从堆栈取回，从 PC 所指示的地址开始执行用户程序，直至结束。(2)有断点，则转入执行单步一次，在单步处理程序中，设置 CTC2 为定时工作方式，定时时间为 176T。使得在恢复各个寄存器状态后，刚返回用户程序时，CTC2 定时到，向 CPU 发出 $\overline{\text{NMI}}$ 请求。因此，在执行了一条用户程序后，CPU 响应非屏蔽中断 $\overline{\text{NMI}}$ 请求，转入非屏蔽中断服务程序 NMIS。在 NMIS 程序中，先保护所有寄存器状态，装入断点和保留用户操作码，然后进入允许用户用 MON 键中断的程序。当执行用户程序到断点处时转入 RST8 程序：把用户栈中 PC 减 1 换成断点地址，然后在所有断点处又换成用户程序原来的操作码，显示断点的地址和当前累加器 A 的内容。

程序框图见图 10。

2.12 STEP 单步命令执行程序(从 0272H 开始)

单步命令键使用户能够每次执行程序中的一条指令，执行完后显示当前累加器 A 的内容和下一条指令的地址。这样便于用户检查和修改程序。重复地按单步键，将使程序一步一步地执行。

程序首先清除所有断点，建立单步标志，然后置 CTC2 为定时器方式。每执行完一条指令后，CTC2 定时到，发出 $\overline{\text{NMI}}$ 请求，CPU 响应转入非屏蔽中断服务程序 NMIS，保护 CPU 内部寄存器，显示下一条指令地址和当前累加器 A 的内容。

程序框图见图 10。

2.13 自定义命令执行程序

在 2FB8H，2FBAH，2FBCH，2FBEH 单元中，分别放置了四个独立的程序段的起

始地址。这四个独立的程序段，为用户自己定义命令功能的执行程序。当按下 MON' 键，显示“/”后，按下 2FB8 或 2FBA 或 2FBC 或 2FBE 键，即按 EXEC 方式执行用户编的程序段。

程序框图见图 11。

2.14 PORT 命令执行程序(从 0368H 开始)

该命令键是用来显示通道地址和通道内容。当显示器上出现提示符“P”后，输入一个两位数的通道地址，随后按下 PORT 命令键即可。若再输入两位新数据，将改变该通道的内容。

程序框图见图 12。

2.15 REG 和 REG' 命令执行程序(从 02FFH 和 02EDH 开始)

REG 命令用来显示和修改寄存器 A, B, C, D, E, F, H, L, IX, IY, PC 以及 IFF 的内容。SP 寄存器的内容只能被检查，不能被更改。当提示符“P”出现后，先按下代表寄存器的数字键，在最左边显示器显示该数字，然后按下 REG 命令键，显示器右边显示寄存器中的内容。若再输入新的数据则改变寄存器的内容。

REG' 命令用来显示和修改辅助寄存器 A', B', C', D', E', H', L', F' 的内容。该键的操作和使用与 REG 命令键相同。

程序框图见图 13。

2.16 MEM 命令执行程序(从 0394H 开始)

在显示提示符“P”后，通过键盘输入存储器地址的四个十六进制数字，然后按 MEM 命令键，显示器显示该存储单元的地址和内容。输入两位新数据可改变存储单元的内容。但是，对于 ROM 中的单元内容，只能检查，不能修改。

程序框图见图 14。

2.17 BP 命令执行程序(从 049DH 开始)

该命令键用来在用户程序中设置 1 至 5 个断点，当用户程序执行到断点处就停止执行，以使用户检查和调试程序。

在程序中设置断点的方法是，通过键盘输入四个十六进制数字(断点地址)，然后按下 BP 键，就完成了断点的设置。当要在程序中再设置断点，则按下 MON 键出现提示符“P”后，重复前面的步骤。当设置的断点数超过 5 个时，则多设的断点无效。

采用下列三种方法，可将已设置的断点清除：

- (1) 按下复位按钮(RESET)。
- (2) 按下 STEP 单步键。
- (3) 在输入不到 4 个数字时，紧接着按下 BP 键。

程序框图见图 15。

2.18 DISP 命令执行程序(从 02A1H 开始)

该命令键为计算相对转移指令的位移量提供了方便。

利用 REG 命令, 置 IX 为相对转移的目的地址, 置 IY 为相对转移指令所在地址。按下 MON' 键, 显示“'”后按 DISP 键, 在最右边两个显示器上显示偏移量, 而最左边两个显示器的显示内容作判断用。若不是 FF 或 00, 则表示相对位移量超出规定范围, 说明结果不正确。

程序框图见图 16。

2.19 DUMP 命令执行程序(从 04CAH 开始)

DUMP 命令用于信息转储, 通过转录线将单板机内存中的程序块转录到录音机的磁带上保存起来。数据记录到磁带上, 要按一定的格式来记录。这里要用到两个标准格式:

1. Kansas 城标准

该标准用来规定记录的信息是“1”还是“0”。标准规定如下:

(1) 一个逻辑“1”用 8 个频率为 2400Hz 的正弦波表示。一个逻辑“0”用 4 个频率为 1200Hz 的正弦波表示。

(2) 数据传递速率为 300 波特, 即每位宽 3.33ms。

(3) 只传递 ASCII 字符。一个字符的排列顺序为: 开始是一个起始位, 规定为“0”, 然后是 7 位 ASCII 码(最低有效位在前, 最高有效位在后), 最后是两个终止位, 规定为“1”。

(4) 整个数据块的开头有 40 秒的空闲位, 为全“1”。数据块结束, 有 5 秒的空闲位为全“1”作为间隔。

2. Intel Hex 标准

该标准是用来表示数据块记录的格式, 其规定如下:

(1) 一个数据块可分成许多个记录。每个记录用:(冒号)开始, 用 CR(回车)和 LF(换行)作为结束。

(2) 数据(信息)用 ASCII 码表示(七位)。

(3) 数据记录格式

字节 1 是:(冒号)

字节 2—3 是记录长度(表示这个记录中的字节数), 一个记录最多为 32 个 ASCII 字符, 所以一个数据块记录只能传 16 个内存单元的信息。

字节 4—5 是数据起始地址的高八位。

字节 6—7 是数据起始地址的低八位。

字节 8—9 是记录类型。若记录结束, 则记录类型用 0 表示。否则, 用 00 表示。

字节 10 从第 10 个字节开始为用 ASCII 码表示的数据字节, 在一个记录内最多为 32 个字节。

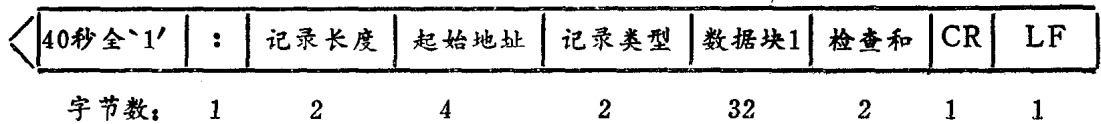
其后两个字节存放的是除定义符, 回车符, 换行符以外的所有字节的校验和, 这个校验和是此记录中所有二进制数的和的负数。

最后是 CR(回车)和 LF(换行)。

信息在磁带上的排列如下页上的图所示。

转储时将程序块的起始地址置于 2FC0H 和 2FC1H 单元, 终止地址置入 2FC2H 和

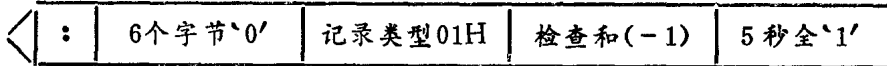
开头记录格式



中间记录格式



结尾记录格式



2FC3H单元。按下 MON' 键，当显示了提示符“'”后，再按下 DUMP 键，就将程序块转储到磁带上。

磁带上记录的“1”或“0”是以不同频率的信号表示的，而不同频率的信号是利用 CTC1 通道来产生的。所以首先要对 CTC1 道通进行初始化，并将要传送的程序块的起始地址和终止地址送至 DE 和 HL 中，建立 40 秒的引导。按 Intel Hex 格式，首先输出冒号，然后比较数据块的长度是否比一个记录(最大为 32 个 ASCII 字节)长。若比它长，则按 Intel Hex 格式组织数据块输出。若比它短，则检查是否是文件的结尾段。若是则按文件结束记录格式输出。

在 DUMP 命令执行程序中，调用了把二进制数转换成 ASCII 码且输出到磁带上的子程序 UAPCCS，以及按 KC 标准把以 ASCII 码表示的字符输出到磁带上的子程序 OTCHR。DUMP 命令的程序框图见图 17，OTCHR 子程序的框图见图 18。

2.20 LOAD 命令执行程序(从0581H 开始)

该命令是用来将存放在磁带上的信息装到单板机内存中。用转录线将单板机与录音机相连，按 MON' 键，显示提示符“'”，然后按下 LOAD 键，“'”标志将消失。录音机置成放音方式，增大音量，机上右上角 LED(发光二极管)发亮。如果输入成功，则显示提示符“'”。若在输入过程中发现某个记录的“检查和”有差错，则显示下一个记录的第一个字节的地址。

程序开始有 15 秒的引导，在输入一个记录时，调用从磁带上输入字符的子程序 INCHR 检查是否为冒号，查到冒号后才是一个记录的开始，然后按 Intel Hex 格式输入一个记录的数据。最后检查“校验和”是否正确，若正确则输入下一个记录，否则进行出错处理，显示下一个记录的起始地址。

INCHR 是输入字符的子程序，先检查是否为起始位，检查到后，再在一位的中间处(延时半位)检查一下，确认为起始位“0”后，就在数据位的中间每隔一位的时间采样数据(以便取得稳定的信息)，把串行的变为并行，且去掉停止位，保留在 A 中。所以，该子程序是把一个字符由 Kansas 城格式变成 ASCII 码形式放在 A 中。

LOAD 命令的执行程序框图见图 19。子程序 INCHR 的框图见图 20。

2.21 PROM 命令执行程序(从 05D3H 开始)

该命令的功能,是将 RAM 中的程序写入插在 PROM2 插座中的 2716/2758 型 EPROM 中。当按下 MON 键出现提示符“P”后,通过键盘向单板机输入四个十六进制数字,以表示要向 EPROM 写入的字节数。然后将开关 S₅置于 PGM 位置,按下 PROM 键,显示器熄灭,并进行写入。

PROM 执行程序是将 RAM 中以 2000H 为起始地址开始的信息传送到起始地址为 1000H 的 EPROM 中。传送完毕后,经过核对,如果没有错误,则显示器上出现“P”。如果有错误,则显示错误地址及内容。此时若按下 NEXT 键,将继续往下进行核对。若还有错,则继续显示错误地址及内容,直至用 NEXT 键检查完毕为止。

程序框图见图21。

2.22 MOVE 命令执行程序(从 0672 开始)

该命令的功能是将内存中的数据块移动一个单元。当出现提示符“P”后,将需要空出的单元地址输入,然后按下 MOVE 键,则从输入地址到 2EFEH 的整个程序块向下移动一个单元,将原输入地址单元空出(清零)。显示器上显示原输入地址的下一个地址及其内容(该内容就是原输入地址单元中的内容)。每按一次 MOVE 键,则就空出一个单元

程序框图见图22。

2.23 子程序

TPBUG-A 监控程序中安排了多个不同功能的子程序,通过子程序的调用,使监控程序得到简化。同时也为用户自己在需要编制新的监控程序时,提供了方便。

这些子程序是:

(1) ALTER——调用地址: 03B4H

功能: 用于内容修改。在输入 8 个数的标志为“1”后,就转入该子程序。在子程序中,首先检查是那一类修改(通道,存储器,寄存器或辅助寄存器内容的修改?)然后转入相应类的修改程序,进行数据处理。

(2) ALTER7——调用地址: 0400H

功能: 累加器 A 中内容左移四次。

(3) UIX3——调用地址: 0634H

功能: $IX + 3, B - 1$, 用于查断点表。

(4) UFOR1——调用地址: 063CH

功能: 把累加器 A 中的高四位和低四位数分别装入所指定的两个显缓区单元中。

(5) D20MS——调用地址: 064FH

功能: 延时 20ms。

(6) UFOR2——调用地址: 0659H

功能: 将显示缓冲区中的前四位数组成地址,写入 HL 中。

(7) UFOR3——调用地址: 067CH

功能: IX 指向断点表首址,断地标志(BFLG)送入 B 中。

(8) UFOR4——调用地址: 06A7H

功能: 将“空白”码在字形表中的序号 10H 送显示缓冲区中的八个单元。

(9) UABIN——调用地址: 06B3H

功能: 将累加器 A 中的一个 ASCII 码转换成二进制数(即转换成一位对应的十六进制数)。

(10) UPACC——调用地址: 06C4H

功能: 将累加器 A 中的两个字符变成 ASCII 码, 分别按 KC 格式转储磁带。

(11) UPACCS——调用地址: 06D5H

功能: 把累加器 A 中两个字符求累加和, 然后转储到磁带上。

(12) ULLCC——调用地址: 06DBH

功能: 把从磁带上连续输入的两个字符的 ASCII 码转换成二进制数, 合并成一个字节放在 A 中, 且加至累加和 C 中。

(13) OTCHR——调用地址: 06F4H

功能: 把需要转储的一个字符的四位二进制数, 转换成 ASCII 码, 按 KC 标准写入磁带。

(14) INCHR——调用地址: 0758H

功能: 把从磁带上输入的一个字符, 由 KC 格式变成 ASCII 码形式, 放入累加器 A 中。

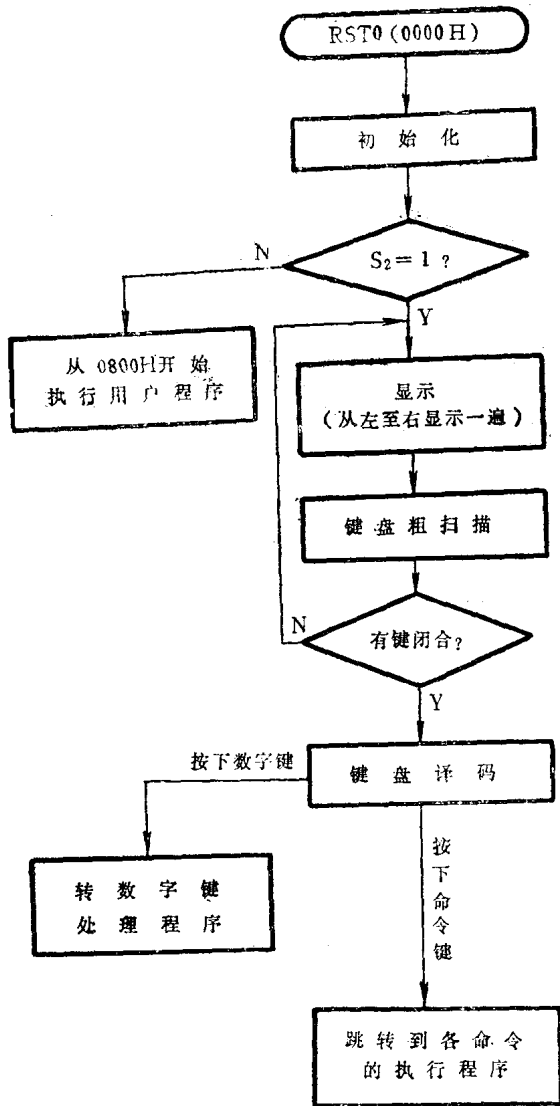


图1 总体框图

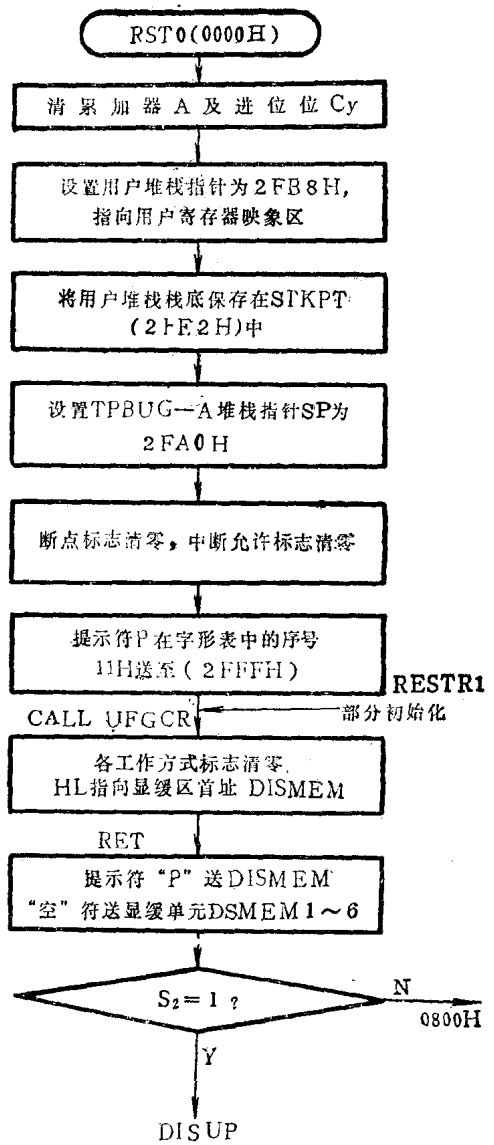


图2 初始化程序框图

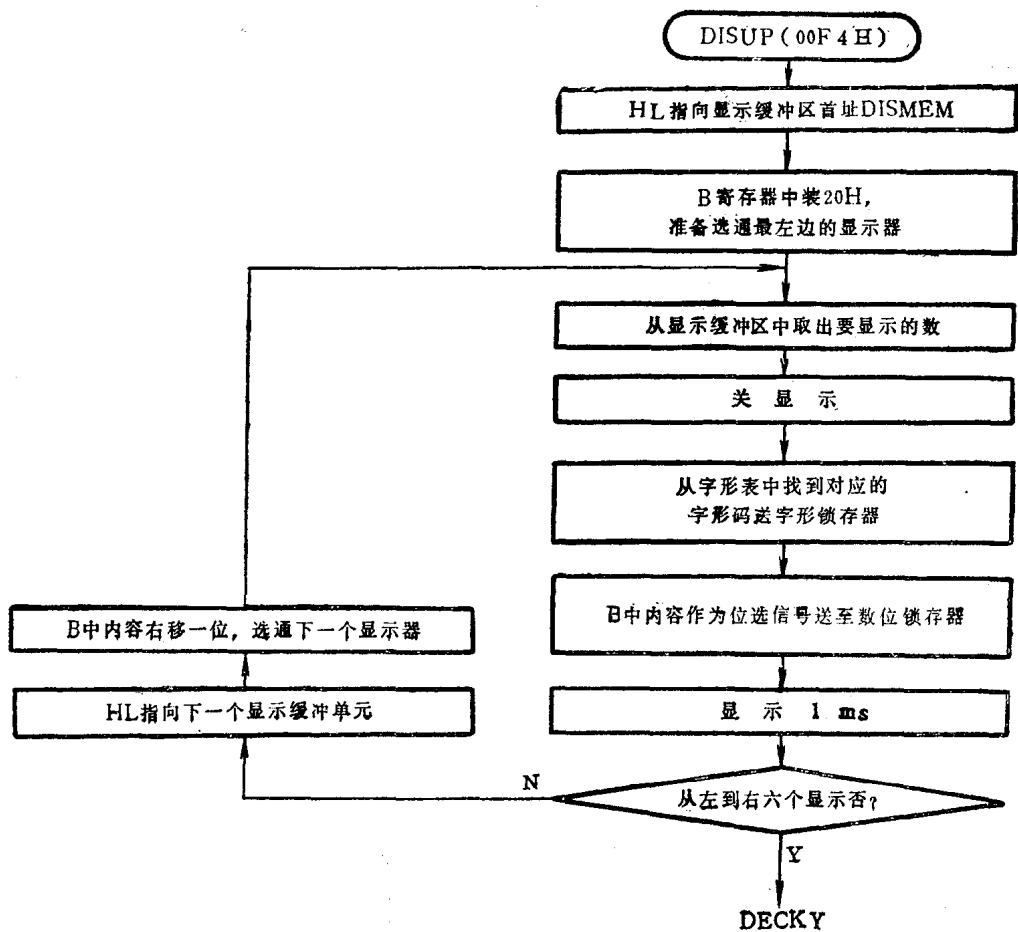


图 3 显示程序框图

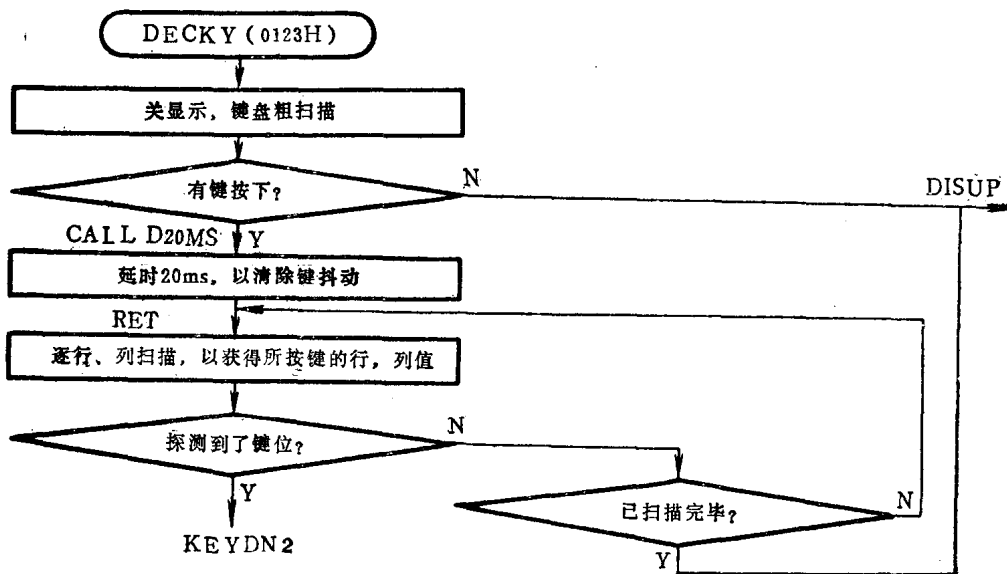


图 4 键盘扫描程序框图