

汇编语言程序设计实验指导书

李兆凤 等编

中央广播电视大学出版社

汇编语言程序设计实验指导书

李兆凤 等编

中央广播电视大学出版社出版

新华书店北京发行所发行

国防科工委印刷厂印装

开本787×1092 1/16 印张6.375 千字159

1988年3月第1版 1988年4月第1次印刷

印数 1—14,000

定价1.25元

ISBN 7-304-00175-5/TP·10

目 录

第一部分	(1)
第一节 TP801 A单板机组成结构	(1)
一、主要技术特征	(1)
二、TP801 A单板计算机的结构	(2)
三、时钟电路	(2)
四、Z80-CPU	(3)
五、存储器	(3)
六、输入输出 (I/O) 接口	(5)
第二节 TP801 A单板机的使用	(8)
一、单板机的开关(按钮)	(8)
二、单板机按键	(9)
三、熟悉键盘操作及命令使用	(15)
四、程序的调试	(17)
第三节 监控程序介绍	(17)
一、概述	(17)
二、TPBUG总程序流程图、主要程序段及其说明	(18)
第四节 TRS-80微型机系统结构	(25)
一、Z-80CPU	(26)
二、ROM	(26)
三、RAM	(26)
四、地址译码器	(26)
五、显示接口	(26)
六、盒式磁带机I/O接口	(26)
七、键盘	(26)
八、电源	(27)
第五节 NEWDOS/80操作系统命令介绍	(27)
一、AUTO命令	(27)
二、BREAK命令	(28)
三、COPY 命令	(28)
四、DATE命令	(32)
五、DEBUG命令	(32)
六、DIR 命令	(33)
七、DUMP 命令	(33)
八、JKL命令	(34)
九、KILL命令	(34)
十、LIST命令	(35)
十一、LOAD 命令	(35)

十二、RENAME命令	(35)
十三、TIME命令	(35)
第六节 编辑/汇编程序及纠错和调试程序	(36)
一、编辑/汇编程序是NEWDOS/80操作系统中的EDTASM程序	(36)
二、实时查错程序是NEWDOS/80中的DEBUG程序	(43)
第七节 系统子程序	(47)
一、简要说明	(47)
二、部分子程序说明	(48)
三、部分ROM子程序	(49)
第八节 在APPLE-Ⅰ机上实现Z-80汇编语言的操作	(50)
一、实现Z-80汇编的条件	(50)
二、目标盘的格式化和CP/M化	(50)
三、Z-80汇编的实现	(51)
四、Z-80汇编文件的处理	(53)
五、A.L.D.S.软件	(55)
第二部分	(58)
实验一 键盘操作实验	(58)
实验二 简单程序的编制	(61)
实验三 循环程序的编制	(64)
实验四 代码转换实验	(66)
实验五 算术运算程序	(67)
实验六 字符显示实验	(70)
实验七 中断程序设计与输入输出实验	(71)
实验八 子程序设计	(74)
实验九 定点及浮点算术运算程序	(75)
实验十 TRS-80微机编辑/汇编程序的应用实验	(78)
实验十一 数据排序实验	(79)
实验十二 在CRT上显示活动图象	(81)
第三部分	(88)
实验参考程序清单	(88)

第 一 部 分

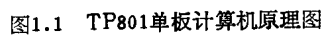
第一节 TP-801A单板机组成结构

TP-801A单板计算机是使用Z-80系列器件,做在一块印刷电路板上的微型计算机。它的结构简单,功能齐全,使用方便。

一、主要技术特征

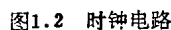
1. 中央处理单元为Z80-CPU。
2. 时钟(ϕ)频率为1.9968MHZ。晶体振荡器的频率为3.9936MHZ。
3. 随机存贮器RAM为4K字节,使用2114静态读写存贮器芯片(共八片),也可以使用2K字节。
4. 只读存贮器为2K字节,使用2716芯片或2758芯片,编入监控程序TPBU-G-A。
5. PROM插座两个,可以插入4K字节的PROM或EPROM。
6. Z80-PIO并行输入输出接口芯片一个,它有两个8位可编程的I/O端口,全部供给用户使用。
7. Z80-CTC计数器/定时器芯片一个,它有四个通道。其中通道0供用户使用,通道1—通道3由监控程序TPBUG-A使用。
8. 按键共28个,有16个为十六进制的数字键;12个为命令键。
命令键有以下功能键:
MON' (换上挡的监控)键
MON (换下挡的监控)键
STEP (单步执行程序)键
EXEC (连续执行程序)键
DISP/MEM (偏数量计算/存贮单元检查)键
REG/REG' (寄存器检查/辅助寄存器检查)键
DUMP/LAST (信息转储磁带/查上一单元)键
LOAD/NEXT (磁带输入/查下一单元)键
2FB8/PROM (用户定义功能/EPROM写入)键
2FBA/MOVE (用户定义功能/存贮块移动)键
2FBC/PORT (用户定义功能/口检查)键
2FBE/BP (用户定义功能/设置断点)键
9. 七段显示器有六位,通常左边四位LED显示地址,右边两位LED显示数据。
10. 配有盒式磁带录音机接口。用来信息的转储。
11. 布线区有2.5×7英寸
12. S-100总线插孔两组

TP801A单板计算机的原理图如图1-1所示。它有三条总线:数据总线DB、地址总线AB、



码电路、时钟电路和复位电路等等。详细的原理图见附图1。

CPU 的时钟频率略低于 2



— 2 —

四、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。这13个控制信号本机使用9个,还有4条信号线没被使用。这4个信号是 $\overline{\text{HALT}}$ 、 $\overline{\text{RFSH}}$ 、 $\overline{\text{BUSERQ}}$ 、 $\overline{\text{BUSAK}}$ ($\overline{\text{BUSERQ}}$ 输入线接高电平,使其无效)。有部分总线接到布线区和 S-100总线。用户做附加电路和实验时所需的信号可自行从总线上引出。

五、存储器

1. 存储空间分配

存储空间分配和片选信号见表1.1。

表1.1 TP801A 的存储分配

地 址	器 件	$A_{15} \sim A_{11}$	$A_{10} \sim A_0$	译码器的有效输出
3800~3FFFH	没 用	00111	可变	$\overline{Y_7} = \overline{\text{CS}_7}$
3000~37FFH	没 用	00110	可变	$\overline{Y_6} = \overline{\text{CS}_6}$
2800~2FFFH	2KRAM ₂ (U ₂₀ —U ₂₃)	00101	可变	$\overline{Y_5} = \overline{\text{CS}_5} = \overline{\text{RAM}_2\text{SEL}}$
2000~27FFH	2KRAM ₁ (U ₁₆ —U ₁₉)	00100	可变	$\overline{Y_4} = \overline{\text{CS}_4} = \overline{\text{RAM}_1\text{SEL}}$
1800~1FFFH	没 用	00011	可变	$\overline{Y_3} = \overline{\text{CS}_3}$
1000~17FFH	2KPROM ₂ (U ₈)	00010	可变	$\overline{Y_2} = \overline{\text{CS}_2} = \overline{\text{PROM}_2\text{SEL}}$
0800~0FFFH	2KPROM ₁ (U ₆)	00001	可变	$\overline{Y_1} = \overline{\text{CS}_1} = \overline{\text{PROM}_1\text{SEL}}$
0000~07FFH	2KROM(U ₇)	00000	可变	$\overline{Y_0} = \overline{\text{CS}_0} = \overline{\text{MON SEL}}$

监控程序TPBUG-A放在最下面的2K字节中、地址号为0000~07FFH。

0800~0FFFH为2K字节的PROM₁, 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大于2K字节, 则可以存放在 PROM₂中。另外, 用户可以利用监控程序对PROM₂插座中的EPROM进行写入。

1800~1FFFH, 2K字节, 没被使用。

2000~27FFH, 2K字节的RAM₁。

2800~2FFFH为2K字节的RAM₂。

RAM (RAM₁和RAM₂) 的存储分配如表 1.2 所示。

3000~37FFH为2K字节, 没被使用。

3800~3FFFH为2K字节, 没被使用。

2. 存储器译码

存储器译码采用了74LS138译码器。74LS138是八中取一译码器。具体电路如图 1.3所示。

74LS138译码器的输出线有8条, 每条线的输出可接至2K存储器的 $\overline{\text{CE}}$ 端或 $\overline{\text{CS}}$ 端, 所以, 本机可配16K存储器。译码器规定, A_{15} 和 A_{14} 应为0, 故16K的存储器的地址空间为0000—3FFFH, 占有下方的16K字节。在本机中, 译码器的输出都是接到 U₁₇的16个引线孔。如果用户要改变2K存储器的地址空间, 则可以改变译码器输出到引线孔之间的连线。

表1.2 RAM的存储分配

地址空间	用 途	字 节 数	插 座 编 号
2RFF 2FC0	TPBUG-A使用的RAM暂存区和断点表	64	U ₂₀ —U ₂₃ RAM ₂
2FBF 2FB8	TPBUG-A四个用户程序入口地址	8	
2FB7 2FA0	用户程序寄存器存放区 用户程序栈工作区	24	
2F9F 2F88	监控程序工作区	24	
2F87 2C00	RAM ₂ 的用户程序工作区(1)	904	
2BFF 2800	RAM ₂ 的用户程序工作区(2)	1K	U ₁₆ —U ₁₉ RAM ₁
27FF 2400	RAM ₁ 的用户程序工作区(2)	1K	
23FF 2000	RAM ₁ 的用户程序工作区(1)	1K	

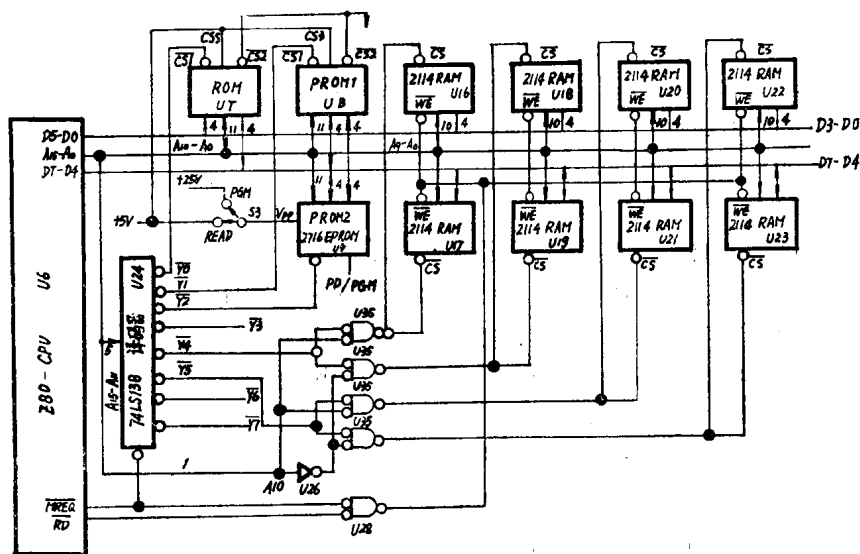


图1.3 RAM与CPU之间的连接

3. 随机存储系统RAM

本机共有4K字节的RAM，其中U₁₀—U₁₉为2K字节的RAM，U₂₀—U₂₁为另外2K字节的RAM。RAM中有部分单元用作TPBUG-A的暂存区和工作区，这在前面的RAM分配表中已经见到。RAM和CPU之间的联接如图1.3所示。

4. PROM₁

当复位信号RESET有效时，从TPBUG-A的0000H单元开始执行程序，在00CDH单元的指令检查S₁开关的位置。如果S₁开关置于MON、RST位置，则继续执行监控程序，在显示器上显示“P”，并且扫描键盘，接收键盘的输入。如果S₁开关位于PROM、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直接进入用户程序。

5. PROM₁-EPROM

这部分是用来对2716/2758型的 EPROM 芯片进行写入操作的。在对 EPROM 写入时，首先要将开关S₁置向PGM位置，即将 EPROM 的 V_{pp} 引脚接至 +25V 电源上。另外，还要将片选信号CS=1, PD/PGM 引脚上输入一个 TTL 电平的脉冲宽度为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，数据总线上应出现被写入的内容。

六、输入/输出 (I/O) 接口

(1) I/O 译码和空间分配

I/O 译码电路如图 1.4 所示。这个译码电路也是采用74LS138译码器。译码器的 8 条输

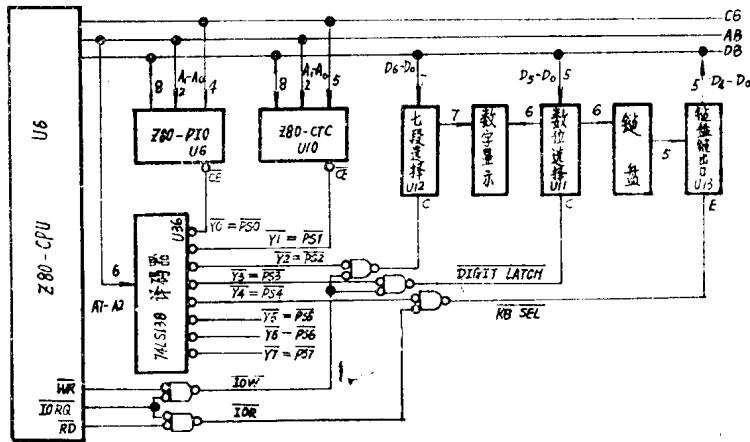


图1.4 I/O译码电路

出线中，其中有 5 条线是本机使用，还有三条线即 $\overline{PS_0}$ 、 $\overline{PS_1}$ 、 $\overline{PS_2}$ 接向布线区，以备用户作为其它使用。对于 PIO、CTC，可以用 A₁ 和 A₀ 线将表中所示的地址分配给有关单元，请看表1.3。

(2) Z80-PIO

Z80-PIO是并行的输入输出接口芯片。它是一个可以用程序来变换其工作方式的器件。它具有两个 8 位的 I/O 口，即 PA 和 PB。每个口有两根联络线，用以制控 PIO 与外设数据的传送。PIO 的两个口全部供给用户使用，这些线都接向布线区，可供用户作附加电路或扩充其它功能使用。

每个口都可以使用联络线进行中断控制，采用 IM₁ 中断方式。

3. Z80-CTC

Z80-CTC 计数器/定时器是一种具有四个独立通道的可编程器件。它有下列两个功能：

(1) 定时功能。它能定时地发出脉冲（并能在发脉冲的同时请求中断）。脉冲的时间间隔、脉冲序列的起始和终了、产生脉冲的方式以及脉冲的个数均可由程序来决定。

(2) 计数功能。它能对外界事件进行计数，当达到程序规定的数值时，向 CPU 请求中

表1.3

A_1-A_2	译 码 输 出	器 件	A_1A_0	口	口 地 址
100000	$\overline{Y}_0 = \overline{PS}_0 = \overline{PIOSEL}$	Z80-PIO	00	口 A 数据寄存器	80H
			01	口 B 数据寄存器	81H
			10	口 A 控制寄存器	82H
			11	口 B 控制寄存器	83H
100001	$\overline{Y}_1 = \overline{PS}_1 = \overline{CTCSEL}$	Z80-CTC	00	通道 0	84H
			01	通道 1	85H
			10	通道 2	86H
			11	通道 3	87H
100010	$\overline{Y}_2 = \overline{PS}_2 = \overline{SEGLH}$	74LS273 八锁存器	× ×	七段选择 (只写)	88H-8BH
100011	$\overline{Y}_3 = \overline{PS}_3 = \overline{DIGLH}$	74LS273 八锁存器	× ×	数位选择 (只写)	8CH-8FH
100100	$\overline{Y}_4 = \overline{PS}_4 = \overline{KBSEL}$	74LS244 八缓冲器	× ×	读 键 值 (只读)	90H-93H
100101	$\overline{Y}_5 = \overline{PS}_5$	没使用			94H-97H
100110	$\overline{Y}_6 = \overline{PS}_6$	没使用			98H-9BH
100111	$\overline{Y}_7 = \overline{PS}_7$	没使用			9CH-9FH

断 (并可输出一脉冲)。

CTC的四个通道中:

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。此外, 通道 0 的输入和输出都接向布线区, 以供用户使用。

通道 1 —— 用于“CASS DUMP”键的动作程序, 该程序将 RAM 中的信息转储到盒式录音机磁带中。通道 1 的中断服务程序起始地址为 0732H。

通道 2 —— 用于“PROM PROG”键、“SINGLE STEP”和“MON”键的动作程序。在 PROM PROG 键动作程序中, 通道 2 用来产生 52ms 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。它们都没有采用 IM2 中断方式, 所以也不需要使用中斷矢量。

通道 3 —— 用于“CASS LOAD”键的动作程序, 此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起始地址为 079DH。

各个通道的中断服务程序的起始地址表放在 ROM 中, 如表 1.4 所示。

4. 显示和键盘

本机有六个 LED 七段显示器, 它能显示十六进制数, 也能显示一些特定字符, 如 TPBUG。监控程序中可以显示“P”、“/”和“空”。

所要显示的数据 (或字符) 要写入 U_{11} 的八锁存器 74LS273 中, 它的口地址是 88H, 至

于要在哪一位上显示数据或字符，这要由U₁₁单元的八锁存器 74LS273 来选择，它的口地址是8CH。

表1.4

地 址	内 容
07F8	D6
07F9	2F
07FA	32
07FB	07
07FC	DD
07FD	2F
07FE	9D
07FF	07

通道 0 中断服务程序起始地址2FD6
通道 1 中断服务程序起始地址0732
通道 2 中断服务程序起始地址2FDD
通道 3 中断服务程序起始地址079D

七段显示器的图形及段号与数据位的对应关系如图 1.5 所示。

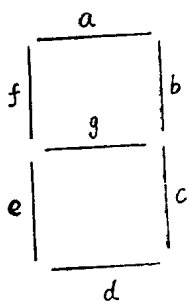


图1.5 七段显示器

LED 显示器最左边的一位叫做 DG₀ 位，其余的依次为 DG₁、DG₂、DG₃、DG₄、DG₅。例如，我们要在最左边的LED显示器DG₀位上显示“4”，可以通过下列程序来完成。

```
LD A, 19H
OUT (88H), A
LD A, 20H
OUT(8CH), A
```

在程序中，19H的二进制表示为：

D ₇	0	0	0	1	1	0	0	1	
	g	f	e	d	c	b	a		

因为是七段显示，只用 7 位数字，所以，D₇位不起作用，对于 D₆—D₀来说，0有效，所以，b c f g亮，得“4”。

20H的二进制表示为：

D ₇	0	0	1	0	0	0	0	0	D ₀
			DG ₀						DG ₅

D₇、D₆位不起作用，对于 D₅—D₀来说，1为有效，即哪位为 1，就在哪位显示，所以将“4”显示在最左边的DG₀位上。

在监控程序中，从地址07A6H-07B8H是七段字形显示表，它给出了 TP801A 单板机所能显示的数字、字母和特定符号显示表，供我们编写程序中参考。表 1.6 是七段显示的代码与字形的关系。

实际的显示过程并不是特别简单，在监控程序中，地址00F4H-0120H就是DISUP显示程序段。我们所编写的显示程序只有 4 条指令，而在监控程序中，DISUP 程序是做了大量的工作的。请参考TPBUG-A监控程序 00F4-0120H 程序段。

表 1.5

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	-

七段显示器的代码与字形的关系

第二节 TP801A单板机的使用

在这部分，我们要了解单板机各个按键的作用，学会单板机的使用。通过这部分的学习，能够在单板机上熟练操作，进行程序的输入、调试和运行。

一、单板机的开关（按钮）

本机中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 S₂置向 MON RST 位置，则在复位后，显示器上显示“P”，并且

扫描键盘的输入；如果S₂置向PROM, RST位置，则在初始化后进入PROM₁中的用户程序。开关S₁用来选择PROM₂中的EPROM处于“写入(PGM)”或者“读出(READ)”的状态。通常，S₂开关设定在MON RST位置，开关S₁设定在READ位置。

二、单板机按键

单板机共有按键28个，其中16个为十六进制数字键；12个为命令键，这些已在前一节中介绍，现在重点介绍12个命令键二十种功能的使用。本机键盘排列简图如图2.1所示。

$\frac{2FB8}{PROM}$	$\frac{2FBA}{MOVE}$	$\frac{2FBC}{PORT}$	$\frac{2FBE}{BP}$	
$\frac{DISP}{MEM}$	$\frac{REG'}{REG}$	$\frac{DUMP}{LAST}$	$\frac{LOAD}{NEXT}$	
7	8	9	A	MON'
4	5	6	B	MON
1	2	3	C	STEP
0	F	E	D	EXEC

图2.1 键盘排列

1. MON(MONITOR)键

MON键的用途是中止或暂停当前的命令状态和数据输入，返回到监控程序，准备接收新的输入和新的命令。通常，要改换和进行一种新的操作之前，都需要压下MON键，使显示器上出现“P”后，才能开始新的操作，压下新的命令键或数字键。MON键与RESET按钮所不同的是：使用RESET按钮时，将破坏前面已执行过的程序的现场；而使用MON键只是暂停当前程序的执行，停止当前的操作，但并不破坏现场。这对检查结果和排除程序故障是很有用的。请看MON键使用的例子。

按键	显示	说明
MON	P	准备接收命令
2000	2000	输入地址2000
MEMEXAM	2000 X X	检查2000单元内容
MON	P	退出以上工作方式
A	A	输入累加器A
REG'/REG	A X X	检查A中内容，X X为读出值

2. MON' (交换) 键

此键除了具有MON键的基本功能使系统进入监控程序待命状态外，还能够完成双功能键的上、下挡更换。

例如：以REG'/REG键为例，上挡键为REG'，下挡键为REG。当按下MON'后，再按下REG'/REG键，监控程序接受上挡键命令，进行REG'访问。如先按下MON键而不是MON'键，则监控程序接受下挡键命令，进行REG访问。为了在使用中引起用户注意，监控程序处于上下挡的不同状态，按下MON'和MON的显示标志也是不同的。按下MON'

时, 显示器则显示“'”, 而按下MON时, 则显示“P”。

3. REG'/REG (REGISter'/REGISter EXAmine寄存器 (检查) 键

此键用来检查或更改寄存器和辅助寄存器的内容。如果先按下MON键, 再压下REG'/REG键, 则是要检查或更改下列寄存器的内容, A, B, C, D, E, F, H, L, I, IFF, PC, IX, IY。SP寄存器的内容只能被检查, 不能被更改。使用此键的方法是: 先按下代表寄存器号的数字键, 再按下此键, 这样相对应的寄存器的内容就显示在显示器的左边两位或四位上 (视被检查的是二进制8位或16位的寄存器而定)。若需更改寄存器的内容, 只须重新打入两个或四个新数据即可。

如果先按下MON'键, 再按此键, 就是要检查或更改辅助寄存器的内容: A', B', C', D', E', F', H', L'。其操作方法与检查更改寄存器内容的方法相同。

4. MEM (内存检查) 键

MEM键是用来检查和改变存贮单元的内容, 即通过键盘命令对RAM单元进行读、写操作。也可以对ROM或PROM进行读写操作。

在使用MEM键之前, 首先压下MON键, 再将存贮地址(四位十六进制数)打在显示器的地址部分, (显示器的左边四位)然后按下MEM键, 则该地址单元的内容就显示在显示器的数据显示部份(显示器的右两位)。

若要改变某存贮单元中的原内容, 只要重新输入两个新的数字就能改变。如果要退出检查内存工作方式。按下MON键即可。

5. STEP(单步执行程序)键

STEP键用来每次执行程序中的一条指令。执行完后, 显示程序计数器PC(下一条指令的地址)和累加器A的内容。每按一次STEP键就执行一条指令。如果重复压下STEP键, 就能够以单步方式通过程序。并在执行中可以观察下一条即将执行的指令地址和A的内容。对于条件转移和转子指令, 用户可以了解转移或跳转的地址是否正确, 条件是否满足。在一步一步执行程序时, 用户可以更改PC的内容, 强迫程序从新的地址开始执行。

6. EXEC(Excute连续执行程序)键

EXEC键用来连续执行RAM、ROM或EPROM中的程序。它有两种操作方法:

(1)从程序的起始地址开始执行。先输入要执行的程序的四个数字的起始地址, 而后压下EXEC键, 这样, Z80-CPU就从起始地址起开始执行程序, 直到程序结束或者直到遇到断点和MON键时才能停止执行。

(2)从当前地址开始执行。如果现在的PC值是执行了一段程序后的值, 此时按下EXEC键, 则程序就从当前PC所指示的地址开始执行。如果现在的PC值是用寄存器检查方式送到PC中的值, 此时压下EXEC键, 就以当前的PC指示的地址为起点开始执行程序, 这种执行方式对于遇到一个断点后, 或者在使用单步执行方式后, 重新开始执行程序是十分有用的。

7. BP(BREAK POINT设置断点)键

断点就是当程序执行到该处时便自动暂停。其目的在于观察或者检查寄存器及其通道执行完一段程序后的内容和状态, 以便可以检查用户在上一段程序设计中有无差错, 进而修改程序, 或者修改I/O口和寄存器中的内容。这对程序的调试是极其方便和有用的。

在用户程序中, 允许设置一至五个断点。设置断点的操作如下:

按 键	显 示	说 明
MON	P	准备接受命令
2015	2015	先输入第一个断点地址
Break Point	2015	断点被接受(BFLG)=1 2015被送往(2FF4)(2FF5)单元
MON	P	
2024	2024	输入第二个断点地址
BP	2024	第二个断点被接受
MON	P	
2000	2000	输入程序首地址
EXEC	2015	从2000H开始执行程序, 暂停于 第一个断点处、显示断点地址 (PC)和累加器A的内容。
EXEC	2024	继续执行程序, 暂停于第二个断 点处, 显示断点地址(PC)和累 加器A的内容。上述三步可反复 进行。
MON	P	返回监控

用户每输入一个断点, 断点标志(BFLG=2FF4H单元)的内容增1, 断点地址被送往以2FF4H单元开始的断点表BPTAB中。这个表共有15个单元, 其中两个单元存放断点地址, 另一个单元存放用户程序中断点处指令操作码的第一个字节。在输入断点地址后, 按下BP键, 显示的地址先暗后明, 表示断点已被接受。如果输入的断点多于五个, 则多设置的断点输入将无效, 显示器上显示“P”。最初设置的五个断点仍然保持有效。

当执行设有断点的程序时, 先执行第一条指令, 然后进入不可屏蔽中断服务程序, 装配断点, 也就是将RST₀.(CFH)单字节指令代替各个断点指令操作码的第一个字节, 而这条指令操作码的第一个字节被送往断点表BPTAB中。然后, 继续执行程序, 直到断点处, 执行RST₀指令。接着进入起始地址为0008H的服务程序。这个服务程序暂停用户程序的执行, CPU寄存器的内容均保存在“用户寄存器存放区”, 并且把断点表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RST₀指令, 以便用户进行检查和修改, 此时显示PC和A的内容。

如果压下EXEC键, 可继续执行用户程序, 如果程序中还有其余断点, 则其过程与上述情况相同。

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

如果用户需要清除已设置的断点, 可用下列三种方法:

- (1) 按下复位按钮(RESET)。
 - (2) 当显示器显示“P”时, 按下STEP键。
 - (3) 在输入断点地址的四个数字之前, 也就是在输入到三个数字后, 紧接着压下BP键。
- 上述三种方法都是将断点标志(BFLG)清零, 从而使得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, 如果用户程序中遇到HALT 指令而停止, 此时压下MON 键, 就使程序转回到监控, 而此时用户程序中的RST₀₀指令并没有被取代; 也就是说, 用户设置的断点仍然保留在程序中。在调试程序的过程中, 应注意到这种情况的发生。

8. NEXT键

NEXT键用于检查存储器, 检查输入输出, 检查EPROM的下一个写入错误。在存储器检查和口检查时, NEXT键用来选择下一个存储器单元或口地址, 并且自动显示其内容, 也就是在对存储器和输入输出检查中起地址“增1”的作用。

9. LAST键

LAST键也是用于访问显示现行存储单元的上一个存储单元。它恰好与NEXT键相反。但是, NEXT键不仅能访问(读或写)下一个存储单元, 而且还可以访问下一个输入输出, 以及EPROM编程时的下一个错误检查。而LAST键只能检查存储器。用户在检查存储单元时, 可以连续按下LAST键, 顺序地显示上一个存储单元的内容, 而且也可交叉使用LAST和NEXT键, 灵活地修改要访问的存储单元。

10. MOVE(存储块移动)键

MOVE的作用是从所显示的地址单元开始, 每按一次MOVE, 这个地址单元的内容就向下移动一个字节。直到2EFFH单元为止。显示器原显示地址单元的内容被清零, 相当于插入了一条单字节的空操作指令NOP。在TP801A单板机中, 用户可用的区域为2000H至2F87H, 使用MOVE键可以移动的存储单元范围为2000至2EFFH。也就是说, 从2FO0H开始至2F87H为止的136个字节不能向下移动。这136个字节可存放数据表格和变量。

11. DISP(相对转移偏移量计算)键

在程序设计中, 经常要用到相对转移指令。DISP键就是进行相对转移指令中的偏移量计算, 显示计算的结果; 并且提示计算的有效性(是否超过单字节范围); 计算结果写入指令的操作数字节。在监控程序中, 从地址00C0H开始的子程序, 能自动计算并填写相对偏移量。相对转移源地址装入IY寄存器对中, 相对转移目的地址装入IX中, 并且用DISP键直接启动此程序, 使用户使用起来非常方便。

如果某JR指令的地址是2300H, 转移的目的地址是22FOH, 计算相对偏移量的操作如下:

按 键	显 示	说 明
MON	P	等待接受命令
IY	5	IY 寄存器访问
REG'/REG	5 × × × ×	
2300	5 23 00	输入源地址
MON	P	
IX	4	IX 寄存器访问
REG'/REG	4 × × × ×	
22FO	4 22 FO	输入目的地址
MON'	'	TPBUG-A进入上档键有效
DISP/MEM	FF EE	上档键DISP有效

计算的结果为EE,也就是从源地址2300H转到目的地址22FOH的相对偏移量为EE。在显示器上最左边两位显示的FF表示此转移为负跳转,如果显示的是00,则表示正跳转。如果最左两位显示的是00和FF以外的数,则说明计算结果超出了相对跳转指令所允许的范围。

12. PORT(PORT EXAMINE口检查)键

口检查键是用来检查或修改口的内容。当显示器上显示提示符“P”时,可通过键盘输入要检查的口地址。口地址是两位数字,先送高位数字,后送低位数字。然后压下2FBC/PORT(下档键有效)键,则在显示器的最右边两位上就会显示出该口的内容。如果要修改此口的内容,直接输入新的数字即可。口检查键的操作与内存检查键的操作相似,故不再做过多的文字叙述。

13. PROM(EPROM PROGRAMMER, EPROM写入)键

在本机中,可以将RAM中起始地址为2000H的数据写入EPROM芯片上,首先将要编程的EPROM芯片2716/2758插在PROM2插座上(起始地址为1000H)。要对EPROM进行写入,还需要有一个 $+25 \pm 1V$, 30mA的辅助电源接到单板机左边标有+25V的点上,检查无误后才能接通单板机的电源和+25V电源。

对EPROM进行写入的操作如下:

(1)将需写入的旧的EPROM芯片用紫外线灯通过芯片窗口进行照射,以擦除原来的内容,也就是将各单元内容全变为FF。若是新的芯片,就不必照射。紫外线的波长为 $2537\text{\AA}$,照射强度为 $12000\mu W/cm^2$,照射能量为 $15W\text{-sec}/cm^2$ 。一般情况下,照射距离为2.5~5.0cm,照射时间为20~40分钟。

(2)在电源关闭的情况下,将EPROM芯片插在PROM2插座上。

(3)接通+5V和+25V电源。

(4)将需写入EPROM的数据送入RAM中。当显示器上显示“P”标志时,输入四位十六进制数字(高位在前,低位在后),表示从RAM中传送到PROM2的字节数。

(5)将开关S₁置于PGM位置,按下PROM键,这时显示器熄灭,表示已开始了写入过程。当重新显示提示符“P”时,表示EPROM已经编程完毕,并且已经证实了EPROM中的内容与RAM中的完全一致。

若EPROM中的内容与RAM中不完全相同,则显示器上就会显示出第一个与RAM中内容不同的单元的地址和内容(前四位显示地址,后两位显示内容)。按下NEXT键,监控程序将继续检查EPROM中的内容。如果没有错误,显示器显示“P”;若再次出现错误,则又会显示出错误单元的地址及其内容。EPROM写入过程中出现错误的原因,一般是使用的EPROM芯片没有擦净,或者是废品。

(6)编程完毕后,将开关S₁拨回到READ位置

14. DUMP(CASS DUMP信息转储)键

当用户需要将RAM中的程序保存起来,以便下次继续使用或者调试时,可以用转储的方法将RAM中的程序转储到录音机的磁带中。操作方法如下:

(1)用转录线将单板机的J₁(AUX)端与录音机的“AUXILIARY”或“MIC”输入端相连。

(2)将磁带装入录音机。

(3)将内存中需要转储的信息的起始地址的高字节送入2FC0H单元,低字节送入