

WEI JI JI SHU SHI YAN

微机技术实验

● 陶龙芳 编

中央广播电视大学出版社

微机技术实验

陶龙芳 编

中央广播电视大学出版社

(京)新登字 163 号

微机技术实验

陶龙芳 编

中央广播电视大学出版社出版

新华书店总店科技发行所发行

一二〇二五厂印刷

*

开本 787×1092 1/16 印张 13.5 千字 308

1992 年 2 月第 1 版 1992 年 2 月第 1 次印刷

印数: 1—9000

定价: 5.25 元

ISBN 7-304-00673-0/TP·36

前 言

微计算机是一门实践性很强的学科,学习时必须理论联系实际,亲自动手做实验,才能达到预期的目的。为此,编写了这本《微机技术实验》,与《微机技术》一书(周明德、陶龙芳编著,中央广播电视大学出版社 1992 年出版)配套使用。

本书分为三章,按照单板机、单片机、PC 机分类编写,并以单板机和单片机并重。共计 33 个实验,内容包括上机操作方法、汇编语言程序设计、硬件和接口实验等。因是配套教材,不再过多叙述原理,而着重于介绍编程方法和使用方法。

本书对所使用的实验装置均作了必要的介绍。书中还给出了实验参考程序,为学生编写程序时作参考,以便提高学生的程序设计能力。程序中的相当部分是从实用程序中提取的,有利于学生从中获取实用知识,为解决问题编写更复杂的应用程序打下基础。

附录中提供了 Z80 指令和 MCS—51 指令的机器码表,以及常用集成电路芯片的引脚图,为实验带来了方便。

本书除作为广播电视大学微机技术课程的实验教材外,也可作为其它高等院校学习微机技术的实验教材,还可作为各种微机技术培训班的实验指导书。

本书由周明德同志主审。编写过程中,徐淑华同志提供了使用方便的 DWJ—1 型单片微机教学机(哈尔滨工业大学计算机科学系研制)和部分实验资料。在此表示诚挚的谢意。

由于编者水平有限,加之编写时间仓促,缺点错误在所难免,敬请广大读者批评指正。

编 者

1991 年 7 月

目 录

第一章 单板微型计算机.....	(1)
第一节 TP801—A 单板机简介	(1)
第二节 实验	(9)
一、Z80 汇编语言程序设计实验	(9)
实验一 TP801—A 单板机键盘操作练习	(9)
实验二 分支程序的运行与调试	(14)
实验三 循环程序之一——数据传送	(15)
实验四 循环程序之二——加法运算	(18)
实验五 子程序调用之一—— $\sum_{i=1}^{10} A_i B_i$ 的计算	(20)
实验六 子程序调用之二——根据程序存贮单元内容决定子程序入口地址	(21)
二、接口实验	(22)
实验七 Z80CTC 基本性能(一)	(23)
实验八 Z80CTC 基本性能(二)	(27)
实验九 Z80CTC 的应用	(31)
实验十 Z80PIO 基本性能(一)	(34)
实验十一 Z80PIO 基本性能(二)	(38)
实验十二 Z80PIO 的应用	(42)
实验十三 8255A 基本性能(一)	(44)
实验十四 8255A 基本性能(二)	(50)
实验十五 8255A 的应用	(55)
三、实验参考程序	(58)
第二章 单片微型计算机	(73)
第一节 DWJ—I 型单片微机教学机简介	(73)
第二节 实验	(77)
一、MCS—51 汇编语言程序设计实验	(77)
实验一 DWJ—I 型单片微机教学机键盘操作练习	(77)
实验二 数据传送	(84)
实验三 多精度加法	(85)
实验四 数据排序	(86)
实验五 代码转换	(87)
二、硬件和接口实验	(88)
实验六 并行口	(88)
实验七 并行口扩展	(92)

实验八 外部中断	(93)
实验九 定时器	(97)
实验十 两级中断及定时器门控方式	(99)
实验十一 显示器	(102)
实验十二 键盘	(106)
实验十三 A/D 转换	(108)
实验十四 D/A 转换	(111)
实验十五 串行通讯	(112)
三、实验参考程序	(115)
第三章 个人计算机系统	(141)
第一节 IBM 个人计算机系统简介	(141)
第二节 实验	(144)
实验一 IBM-PC DOS 的启动和常用命令	(144)
实验二 行编辑程序 EDLIN	(153)
实验三 CC-DOS 的操作与使用	(159)
附录	(170)
附录 1 Z80 指令的机器码表	(170)
附录 2 MCS—51 系列单片机	(183)
(1)指令的机器码表	(183)
(2)内部 RAM 中 20H~2FH 的位地址表	(187)
(3)特殊功能寄存器地址表	(188)
附录 3 常用芯片引脚图	(189)

第一章 单板微型计算机

第一节 TP801—A 单板机简介

TP801—A 是全部器件集中在一块印刷电路板上的微型计算机,故而称为单板微型计算机,简称单板机。它的原理框图如图 J1—1 所示。共有三条总线:数据总线 DB、地址总线 AB 和控制总线 CB。挂在这三条总线上的器件有下列三类:

1. Z80CPU;
2. 存储器:有 ROM, RAM;
3. 接口电路:有 Z80PIO, Z80CTC, 键盘, 显示器, 录音机接口。

此外,尚有译码电路、时钟电路、复位电路等。

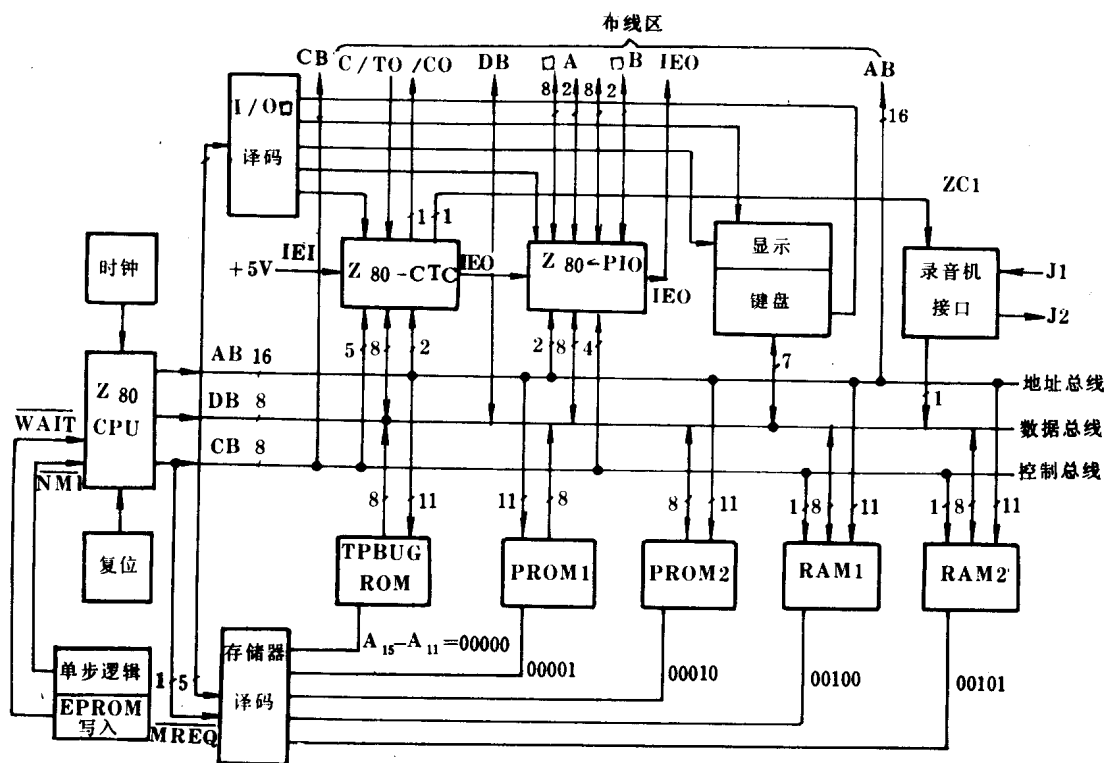


图 J1-1 TP801—A 单板机原理框图

一、存储器组成和地址分配

TP801—A 的存储器分为 ROM 和 RAM 两大部分。

ROM 又分为 ROM、PROM1 和 PROM2。每一种为 2K 字节。ROM 中存放系统的监控和调试程序 TPBUG—A, 用于管理 TP801—A 的工作, 也可用于调试用户程序。PROM1 中存放固化的用户程序。PROM2 是一片 EPROM, 可写入在 RAM 中调试过的用户程序。

RAM 又分为 RAM1 和 RAM2。每组也是 2K 字节。RAM 可用作用户的程序区和数据区, 以及系统和用户的堆栈。还有一部分空间用作系统的数据区。

TP801—A 采用一片 74LS138 作为地址译码器。译码情况如图 J1-2 所示。

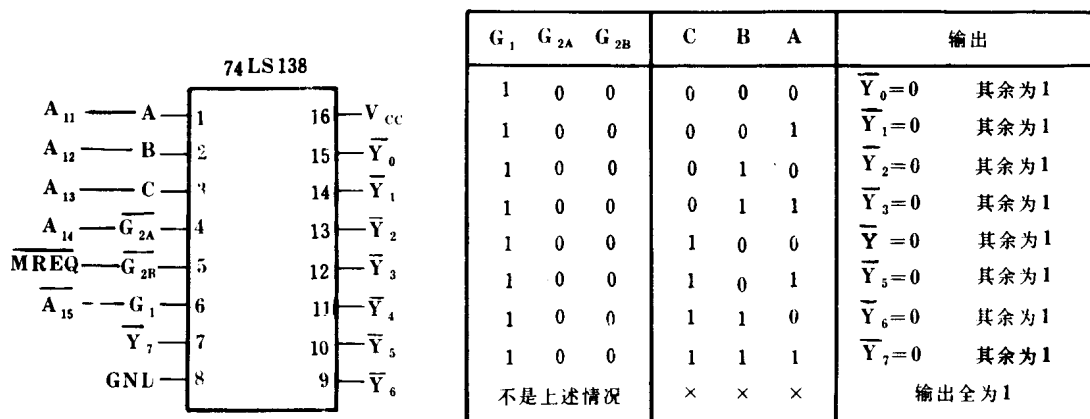


图 J1-2 存储器地址译码

得到 TP801—A 中存储器的地址分配如表 1-1 所示。从表中可见三条地址译码线 ($\overline{CS}_3$ 、 $\overline{CS}_6$ 、 $\overline{CS}_7$) 尚未使用, 用户可以用来扩充内存。

RAM 区域的分配如表 1-2 所示。

表 1-1

地 址	器 件	$A_{15}-A_{11}$	$A_{10}-A_0$	译码器的有效输出
0000—07FFH	2K ROM	0 0 0 0 0	可变	$\overline{Y}_0=\overline{CS}_0=\text{MON SEL}$
0800—0FFFFH	2K PROM1	0 0 0 0 1	可变	$\overline{Y}_1=\overline{CS}_1=\text{PROM}_1 \text{ SEL}$
1000—17FFH	2K PROM2	0 0 0 1 0	可变	$\overline{Y}_2=\overline{CS}_2=\text{PROM}_2 \text{ SEL}$
1800—1FFFFH	没用	0 0 0 1 1	可变	$\overline{Y}_3=\overline{CS}_3$ 没用
2000—27FFH	2K RAM1	0 0 1 0 0	可变	$\overline{Y}_4=\overline{CS}_4=\text{RAM}_1 \text{ SEL}$
2800—2FFFFH	2K RAM2	0 0 1 0 1	可变	$\overline{Y}_5=\overline{CS}_5=\text{RAM}_2 \text{ SEL}$
3000—37FFH	备用	0 0 1 1 0	可变	$\overline{Y}_6=\overline{CS}_6$ 没用
3800—3FFFFH	备用	0 0 1 1 1	可变	$\overline{Y}_7=\overline{CS}_7$ 没用

表 1-2

地 址 空 间	用 途	字 节 数
2000—23FFH	RAM1 的用户区	1K
2400—27FFH	RAM1 的用户区	1K
2800—2BFFH	RAM2 的用户区	1K
2C00—2F87H	RAM2 的用户区	904
2F88—2F9FH	监控程序堆栈工作区	24
2FA0—2FB7H	用户程序寄存器存放区, 用户程序堆栈区	24
2FB8—2FBFH	TP801—A4 个用户程序的入口地址	8
2FC ₀ —2FFFH	TPBUG—A 使用的 RAM 暂存区和断点表	64

二、I/O 设备和接口电路的端口地址分配

TP801—A 采用一片 74LS138 作为接口地址译码器, 译码情况如图 J1-3 所示。

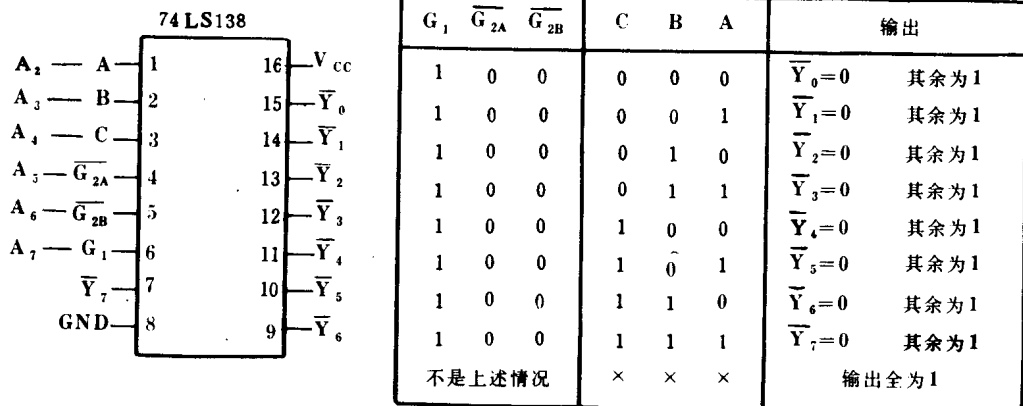


图 J1-3 接口地址译码

得到具体的接口地址分配如下表:

表 1-3

A_7-A_2	译码器输出	器 件	A_1A_0	端 口	端口地址
100000	$\overline{Y}_0 = \overline{PS}_0 = \overline{PIO SEL}$	Z80-PIO	0 0	端口 A 数据	80H
			0 1	端口 B 数据	81H
			1 0	端口 A 控制寄存器	82H
			1 1	端口 B 控制寄存器	83H

续表

A_7-A_2	译码器输出	器 件	A_1A_0	端 口	端口地址
100001	$\overline{Y_1} = \overline{PS_1} = \overline{CTC SEL}$	Z80-CTC	0 0	通道 0	84H
			0 1	通道 1	85H
			1 0	通道 2	86H
			1 1	通道 3	87H
100010	$\overline{Y_2} = \overline{PS_2} = \overline{SEGLH}$	74LS273(八锁存器)	××	七段选择(只写)	88—8BH
100011	$\overline{Y_3} = \overline{PS_3} = \overline{DIGLH}$	74LS273	××	数位选择(只写)	8C—8FH
100100	$\overline{Y_4} = \overline{PS_4} = \overline{KB SEL}$	74LS244(八缓冲器)	××	读键值(只读)	90—93H
100101	$\overline{Y_5} = \overline{PS_5}$	没使用			94—97H
100110	$\overline{Y_6} = \overline{PS_6}$	没使用			98—9BH
100111	$\overline{Y_7} = \overline{PS_7}$	没使用			9C—9FH

由表中可见, PS_5 、 PS_6 、 PS_7 未使用, 可用于接口扩充。

三、Z80PIO 编程摘要

Z80PIO 是通用的并行 I/O 接口芯片。有两个独立的 8 位端口 A 和 B, 全部供用户使用。每个口有两条联络线, 都接向布线区, 可供用户附加电路。每个口可用联络线进行中断控制, 并采用 IM_2 中断方式。

在使用 PIO 之前, 首先要进行初始化, 使其能实现我们所需要的功能。下面给出 PIO 初始化编程摘要。

1. 方式控制字

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
M_1	M_0	x	x	1	1	1	1
未使用				方式标志			
工作方式							

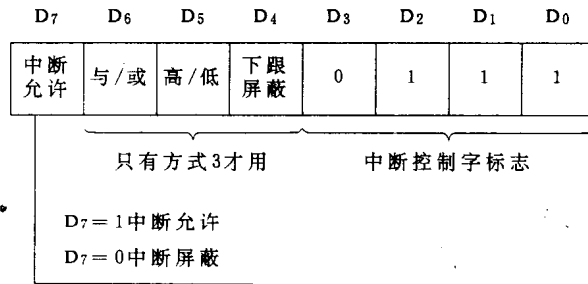
00	方式 0(输出)
01	方式 1(输入)
10	方式 2(双向)
11	方式 3(位控)

2. 输入或输出选择控制字

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I/O ₇	I/O ₆	I/O ₅	I/O ₄	I/O ₃	I/O ₂	I/O ₁	I/O ₀

任一一位, 1=输入; 0=输出。

3. 中断控制字



D₆=1, 则各位都必须有效(即相“与”), 才产生中断;

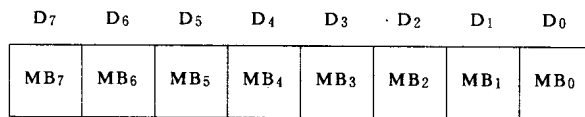
D₆=0, 则任一位有效(即相“或”), 就产生中断。

D₅ 是用来规定引起中断的端口总线的有效极性,

D₅=1, 高电平有效; D₅=0, 低电平有效。

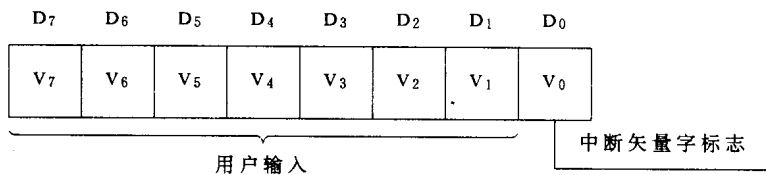
D₄=1, 则送到 PIO 的下一个控制字, 应是中断屏蔽字。

4. 中断屏蔽字

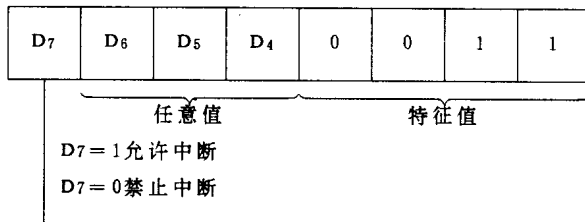


任一位, 1=屏蔽; 0=允许引起中断。

5. 中断矢量字



6. 开关中断控制字



四、Z80CTC 编程摘要

Z80CTC(计数器定时器电路)共有四个通道: 通道 0、1、2、3, 其中通道 0 留给用户使用, 可用程序选择工作在计数方式或定时方式。并可用程序规定它们的量程。下面给出 CTC 初始化编程摘要。

1. CTC 通道控制字:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
中断 允许	方式	量程	斜率	触发 脉冲	装入时 间常数	复位	1

D₇=1, 允许中断; D₇=0, 禁止中断。D₆=1, 计数器方式; D₆=0, 定时器方式。D₅ 只在定时方式有意义, D₅=1, 定标系数 256; D₅=0, 定标系数 16。

D₄=1, 定时方式——脉冲上升沿触发启动定时器工作、计数方式——脉冲上升沿使计数器减 1; D₄=0 则用下降沿。

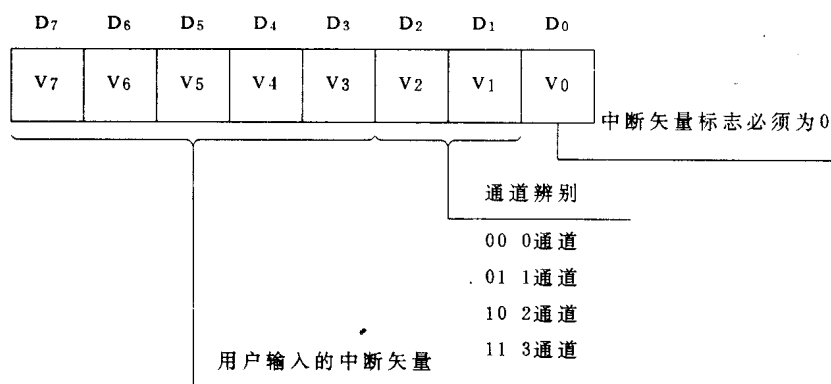
D₃ 仅用于定时方式, D₃=1, 外触发启动; D₃=0, 自动启动。

D₂=1, 表示通道控制字后一个字是要写入的时间常数; D₂=0, 没有时间常数要写入。

D₁=1, 通道复位; D₁=0, 通道继续现行操作。

D₀=1, 为通道控制字标志。

2. 中断矢量字:



3. 时间常数字:

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

它可以是 1 到 256 中的任一整数, 当放入全 0 时则表示时间常数为最大值 256。

五、显示器和键盘

TP801—A 没有汇编程序, 以机器语言进行操作。用户程序必须经过手工汇编变成机器码, 才能送入机器运行。

TP801—A 的典型输出设备是 6 个 7 段发光管显示器, 可以显示机器码。通常左面 4 个显示管显示内存单元的地址或 CPU 内部寄存器号, 或端口地址。而右面 2 个显示内存单元或寄存器或外设端口的内容。

TP801—A 的典型输入设备是键盘。有 16 个 16 进制数字键和 12 个功能键, 如图 J1-4 所

示。

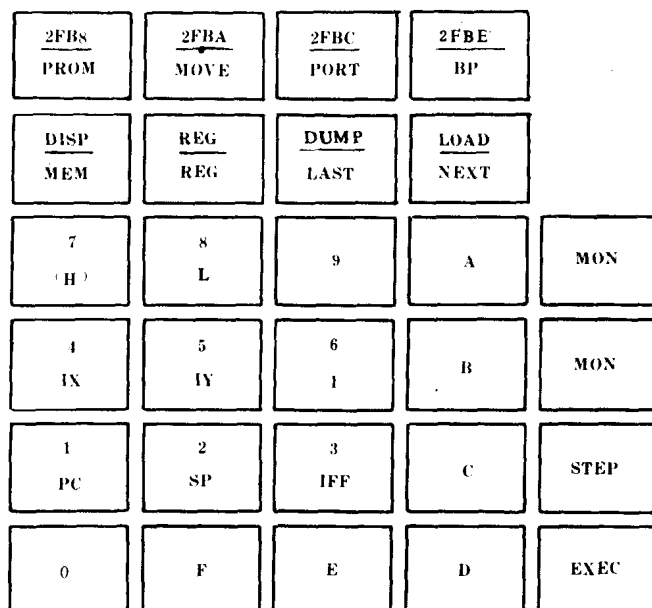


图 J1-4 TP801—A 键盘

16 个数字键是 0~9, A~F。用来输入 16 进制数字, 组成 4 位地址或 2 位数据。也用来选择寄存器, 如选寄存器 A、B、C、D、E、F 时, 则按相应数字键 A、B、C、D、E、F, 如选寄存器 H、L、IX、IY、I、PC、SP、IFF 时, 则按数字键 7、8、4、5、6、1、2、3。

12 个功能键分为两部分, 一部分在数字键的右侧, 包括 4 个键 MON'、MON、STEP、EXEC。另一部分在数字键上方, 包括 8 个键, 每个键有两种功能, 横线上方符号为上档键, 受 MON' 键控制, 横线下方符号为下档键, 受 MON 键控制。

1. MON 键: 能使机器进入监控程序; 也能中止现程序的运行, 返回监控; 还能中止当前的命令或输入的数据, 返回监控; 同时使下档功能键有效。

2. MEM 键: 能检查或修改 RAM 中某单元的内容。

3. LAST 键: 能检查或修改上一存贮单元的内容。

4. NEXT 键: 能检查或修改下一存贮单元或 I/O 端口地址的内容。

5. REG 键: 能检查或修改 CPU 内部寄存器 A、B、C、D、E、F、H、L、I、IFF、IX、IY、PC 的内容; 寄存器 SP 的内容只能读出, 不能写入。

6. EXEC 键: 用来连续执行存放在存贮器(RAM、ROM 或 EPROM)中的程序。注意程序一般以 HALT 指令结束, 显示器变暗, 当按 MON 键后, 显示器才出现提示符“P”。

7. STEP 键: 给寄存器 PC 置要执行的指令地址后, 按下 STEP 键, 机器执行一条指令。执行

完这条指令后,显示器左边4位显示下一条指令地址,右边2位显示累加器A内容。再按STEP键,继续执行下一条指令。

8. BP键:能在用户程序中设置1至5个断点。设置断点的方法是,按MON键显示提示符“P”,键入断点地址后按BP键。消除断点的方法有三种,它们是按RESET按钮,或按STEP键,或在未键入4位16进制地址前按BP键。

9. MOVE键:当显示器显示某一存储单元的地址时,按下MOVE键,则从该地址向下,各单元内容顺序向下移动一个字节,而显示地址的单元内容被清零。TP801-A使用MOVE的有效地址范围是2000H~2EFFH。

10. PORT键:用来检查或修改输入输出端口地址的内容。

11. PROM键:将RAM中的程序写入到PROM2位置上的EPROM中。

12. MON'键:使上档功能键有效,与MON键的其它功能相同。按MON'键,出现的提示符是“'”。

13. REG'键:用于显示或修改CPU辅助寄存器的内容。

14. DISP键:可计算相对转移指令中的偏移量。方法是:将相对转移指令的源地址送入IY,目的地址送入IX,按MON'键,再按DISP键。监控程序就进行偏移量的计算。计算完后,显示器左边两位显示转移方向,若为00,表示正向转移;若为FF,表示负向转移。不是00或FF,表示偏移量超出了允许范围。显示器右边两位显示用补码表示的偏移量,并自动将偏移量填入转移指令的第二字节。

15. DUMP键:能把RAM中存放的程序或数据转录到盒式录音磁带上。

16. LOAD键:能将存储在磁带上的信息输入到RAM中。

17. 2FB8、2FBA、2FBC、2FBE四个键:把四个程序(不能太长)同时输入内存,入口地址分别放在2FB8、2FB9、2FBA、2FBB、2FBC、2FBD、2FBE、2FBF等8个单元。按下四个键中的任意一个,就执行相应的一个程序。

六、TP801-A 初始状态的设置

1. 电源电压应为 $+5V \pm 10\%$,极性连接正确。

2. 开关 S_2 有两个位置:MON RST和PROM1 RST,倒向MON RST时,机器在监控程序管理下工作;倒向PROM1 RST时,机器直接运行PROM1中的程序。通常情况下, S_2 应倒向MON RST。

3. 开关 S_3 有两个位置:READ和PGM,倒向READ时,使PROM2插座上的芯片工作在只读方式,即在监控程序管理下工作;倒向PGM时,使PROM2插座上的芯片处于EPROM的写入状态。

4. 接通电源后,显示器应出现提示符“P”。否则按一下按钮 S_1 ,即RESET键,使显示器出现提示符“P”。说明机器已处于监控程序管理之下。

MON键和RESET键都使机器进入监控程序,但功能有些差别,前者能保护CPU各寄存器内容,而后者不能保护。

第二节 实 验

一、Z80 汇编语言程序设计实验

在 Z80 汇编语言表示格式中常用符号约定如下:

r: 表示寄存器 A,B,C,D,E,H,L 中的任意一个。

qq: 表示寄存器对 BC,DE,HL,AF 中的任意一个。

ss: 表示寄存器对 BC,DE,HL,SP 中的任意一个。

pp: 表示寄存器对 BC,DE,IX,SP 中的任意一个。

rr: 表示寄存器对 BC,DE,IY,SP 中的任意一个。

dd: 表示寄存器对 BC,DE,HL,SP 中的任意一个。

s: 表示 r,n,(HL),(IX+d),(IY+d)中的任意一个。

m: 表示 r,(HL),(IX+d),(IY+d)中的任意一个。

(HL): 表示以寄存器对 HL 的内容为地址的存贮单元的内容。

cc: 表示条件转移指令 JR 和 JP 的状态标志。

n: 表示范围为 0~255 的一字节表达式。

nn: 表示范围为 0~65535 的二字节表达式。

(nn): 表示以二字节表达式 nn 为地址的存贮单元的内容。

d: 表示范围为 -128~+127 的一字节表达式。

e: 表示范围为 -126~+129 的一字节表达式。

b: 表示范围为 0~7 的表达式。

←: 表示数据传送方向。

↔: 表示数据交换。

(x): 表示由 x 寻址的单元内容。

实验一 TP801-A 单板机键盘操作练习

一、实验目的

1. 通过实验了解和熟悉所使用的机器的开机、关机操作。
2. 通过实验了解和熟悉所使用的机器键盘上各个键或有关命令的功能及其使用方法。
3. 通过简单程序的练习,学会使用机器及对程序进行调试。

二、实验内容

向机器输入如下给定程序,进行调试、运行。

地址	目的程序	源程序	
		ORG	2000H
2000	110021	LD	DE,2100H
2003	210023	LD	HL,2300H
2006	010001	LD	BC,100H
2009	1A	LD	A,(DE)
200A	77	LD	(HL),A
200B	D5	PUSH	DE
200C	E5	PUSH	HL
200D	FDE1	POP	IY
200F	DDE1	POP	IX
2011	2F	CPL	
2012	DD7701	LD	(IX+1),A
2015	FD7701	LD	(IY+1),A
2018	0B	DEC	BC
2019	EB	EX	DE,HL
201A	D9	EXX	
201B	76	HALT	

三、实验准备

1. 仔细阅读你所使用的机器的使用说明书。
2. 仔细阅读本实验指导。
3. 对照实验步骤,明确实验所要测的各种数据。

四、实验步骤

1. 接通电源

- (1) 将开关 S2 置于“MON”位置,开关 S3 置于“READ”位置。
- (2) 接入电源前,以三用表测量电源电压,保证其数值为 $+5V \pm 10\%$ 。
- (3) 将单板机左测电源输入端的黑头接电源公共端,红头接电源正端,电源的输出电流要求大于 1A。

(4) 检查电源接线是否正确。单板机没有过流保护,电源极性不能接反,否则单板机上的管子会因反向击穿而损坏。

(5) 接通电源后,按 RESET 按钮 S1,数码管最左边一位显示出提示符“P”,说明单板机能正常工作。否则马上断电检查。

2. 键盘操作练习

- (1) 向机器输入给定程序

键入

显示

MON	P	(进入监控程序)
2000	2000	(输入地址)
MEM EXAM	2000 × ×	(检查 2000 存贮单元内容)
11	2000 1 1	(输入 2000 存贮单元应有内容)
NEXT	2001 × ×	(检查下一个存贮单元内容)
00	2001 0 0	(输入 2001 单元应有内容)
NEXT	2002 × ×	
21	2002 2 1	
.....		

用上述方法把整个程序输入完毕。

(2)检查存贮单元内容,看输入的机器码是否正确。

要求用 LAST 键由 201B 单元向上检查到 2000 单元。将每一存贮单元的内容,与应该输入的内容相比较,如果相同,就只作检查;如果不同,进行修正,操作的方法与 NEXT 键相同。假设把 2019 单元错误地输入 FB 了。

键入	显示
201B	201B
MEM EXAM	201B 76
LAST	201A D9
LAST	2019 FB
EB	2019 EB
LAST	2018 0B
.....	

最后检查存贮单元 2101、2100、2301、2300 的内容。

(3)检查各寄存器内容

键入	显示	
MON	P	(返回监控)
A	A	(累加器 A)
REG EXAM	A × ×	(寄存器检查命令,显示 A 的内容)
MON	P	(返回监控)
C REG EXAM	C × ×	(检查 C 寄存器,显示 C 的内容)
MON	P	
B REG EXAM	b × ×	(检查 B 寄存器,显示 B 的内容)
MON	P	
2 REG EXAM	2 × × × ×	(检查 SP 的内容)
MON	P	
4 REG EXAM	4 × × × ×	(检查 IX 的内容)