

前 言

微型机是一门实践性很强的课程，只有通过大量上机实验和较充分的编程练习，才能比较牢固的掌握课程内容。为此我们编写本实验和习题指导书。指导书分两部分，第一部分是编程实验和习题，这部分是想通过几个程序的导入，运行以及根据给出的习题进行编程练习，并对自己所编的程序上机调试，以求熟悉 TP 801 单板机键盘的操作；掌握调试程序的基本方法；熟悉 Z 80 指令功能；掌握程序设计的基本要领和技巧。第二部分是接口实验和习题，这部分包括存储器扩充和可编程序接口 PIO，CTC 的实验和习题。通过这些实验和习题练习，以求掌握存储器扩充、PIO，CTC 可编程序接口的功能和应用。为今后设计组成微型机系统和微型机的应用打下初步基础。

本指导书的内容，由浅入深，由简到繁，并按微型机的授课章节安排实验顺序。教材中的例题，实验内容和习题互相配合互相补充，所以实验时间要和授课进程紧密配合。才能收到较好的效果。

由于实验和习题较多，可以根据实验的课时，由指导教师指定必做和选做内容。

思考题和习题答案，暂不在指导书中列出。

目 录

第一部分 编程实验和习题

- 实验一、 单板机 TP801 的操作
- 实验二、 数据传送和交换
- 实验三、 数据块传送
- 实验四、 基本算术和逻辑运算
- 实验五、 跳转·检索
- 实验六、 循环移位
- 习题一、 传送与交换指令练习
- 习题二、 算术和逻辑运算指令练习
- 习题三、 简单的循环程序编制
- 习题四、 字符编码程序
- 习题五、 代码转换程序
- 习题六、 算术运算程序

第二部分 接口实验和习题

- 实验一、 存储器扩充实验
- 实验二、 数据显示实验
- 实验三、 Z80—PIO 接口实验(1)
- 实验四、 Z80—PIO 接口实验(2)
- 实验五、 Z80—CTC 接口实验
- 习题一、 存储器扩充和测试
- 习题二、 字符显示编程练习
- 习题三、 PIO 编程练习
- 习题四、 CTC 编程练习
- 习题五、 TP801 实时控制

第一部分 编程实验和习题

实验一 单板机TP801的操作

目的: 熟悉 TP801 键盘操作

掌握程序导入, 存储器检查和寄存器检查方法

掌握用单步, 设断点和连续执行键运行程序的方法

器材: TP801 单板计算机一台

5V 直流稳压电源一台

要求: 通过下面程序的导入、检查和执行, 一方面熟悉通过键盘命令导入和检查程序的操作; 另一方面通过显示器的显示, 观察指令执行结果, 并熟悉 MEM, NEXT, LAST, MON, REG, STFP, B·P 和 EXEC 等命令键的使用。初步了解程序的调试过程。

程序:

地址(H)	机器码(H)	汇编语言程序
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, 0AA
200A	76	HALT

步骤:

(1) 通过键盘操作将上面程序导入单板机。(被操作的按键均标以横线, TP-BUG 响应在显示器上均以方框表示。

按 键	显 示
RESET (或 MON)	P
2000 MEM	2000 x x
<u>3</u> <u>E</u>	2000 3 E
NEXT <u>A</u> <u>A</u>	2001 A A

按 键

显 示

NEXT 0 6

2002 0 6

NEXT B B

2003 b b

NEXT 7 8

2004 7 8

NEXT 0 E

2005 0 E

NEXT C C

2006 C C

NEXT 7 9

2007 7 9

NEXT 3 E

2008 3 E

NEXT A A

2009 A A

NEXT 7 6

200A 7 6

(2) 用 LAST 键检查目标程序是否正确无误

按 键

显 示

200A 7 6

LAST

2009 A A

LAST

2008 3 E

LAST

2007 7 9

LAST

2006 C C

LAST

2005 0 E

LAST

2004 7 8

LAST

2003 b b

LAST

2002 0 6

LAST

2001 A A

LAST

2000 3 E

注意：如果检查发现某个单元内容和你的目标指令不一致，可以用数字键立即修

改。但修改时，一定要同时按两位数字键才能更改。

例如 2007 单元内容应为 79，但用 LAST 键检查时，若显示器显示出 2007 78 说明该单元内容不正确，须要修改，此时你必须按下 7 再按 9 两个数字键后，2007 单元的内容才能更改为 79，同时，更改后的内容在显示器上显示出。 2007 79。

(8) 用 REG 键检查未执行程序之前有关寄存器的内容：

按 键		显 示	
MON		P	
<u>A</u>	REG	A	× ×
MON		P	
<u>B</u>	REG	b	× ×
MON		P	
<u>C</u>	REG	C	× ×
MON		P	

注：未执行程序前，有关寄存器的内容是随机数，此处用 × × 表示，你可以在 × × 符号边上记下这个随机数，以便和执行程序后比较。REG 键不仅可以用来检查寄存器内容，还可以更改寄存器内容。如果要更改该寄存器内容，只要通个数字键，按下相应的数字。下列 CPU 寄存器 A, B, C, D, E, F, H, L, I, IFF, PC, IX 和 IY 可以被检查，也可以被更改。但 SP 寄存器的内容只能被检查，不能被更改。IFF 是中断允许触发器，其内容为 00 表示禁止中断，04 表示允许中断。

(4) 用 STEP 键单步执行程序。

STEP 键可以用来每次执行程序中的一条指令。执行完后，在左边四位显示器上显示 PC 内容（即下一条指令的地址）在右边二位显示器上显示 A 的内容。此时，你可以使用 MON 键以及其他按键来检查或修改程序，I/O 口或 CPU 寄存器。

当你反复地压下 STEP 键，将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地址。对于条件转移和条件转了指令，这种功能是很有用的，你可以了解程序是否发生转移以及跳转到什么地址。单步执行程序操作步骤如下：

按 键		显 示	
MON		P	
PC		1	

按 键	显 示	
REG	1 × ×	× ×
<u>2</u> <u>0</u> <u>0</u> <u>0</u>	1 20	0 0
STEP	2002	A A
STEP	2004	A A
MON	P	
<u>B</u> REG	b	b b
STEP	2005	b b
STEP	2007	b b
MON	P	
<u>C</u> REG	C	C C
STEP	2008	C C
STEP	200A	A A

(5) 用 B·P 键设置断点

B·P 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，利用设置断点，可以使程序在某处暂停执行。以便你用 MON 键和其他按键检查上一段程序设计中是否有发生差错。进而修改程序或寄存器和 I/O 口中的内容。

在 2002、2005、2008、200A 单元设置断点，操作步骤如下：

按 键	显 示	
MON	P	
<u>2</u> <u>0</u> <u>0</u> <u>2</u> B·P	2002	
MON	P	
<u>2</u> <u>0</u> <u>0</u> <u>5</u> B·P	2005	
MON	P	
<u>2</u> <u>0</u> <u>0</u> <u>8</u> B·P	2008	

按 键	显 示
MON	P
<u>2</u> <u>0</u> <u>0</u> <u>A</u> B·P	200A
MON	P

注：断点只能设在指令操作码的第一个字节，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，每个断点占用三个单元，其中两个存放断点地址，另一个存放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 B·P 键，显示的地址应暗后复明，表示断点已被接收。本机最多能设置五个断点，如果断点数多于五个，则多设的断点无效，此时显示器上显示出提示符‘P’。最初输入的五五个断点仍保持不变。如果你要知道有没有断点，有几个断点，可以检查 BFLG 单元。

（6）用 EXEC 键执行设置有断点的程序，注意观察显示器的变化，并和用 STEP 键方式执行程序比较。

按 键	显 示
MON	P
<u>2</u> <u>0</u> <u>0</u> <u>0</u>	2000
EXEC	2002 A A
EXEC	2005 b b
MON	P
<u>B</u> <u>REG</u>	b b b
EXEC	2008 C C
MON	P
<u>C</u> <u>REG</u>	C C C
EXEC	200A A A

（7）清除断点

你可以用下述三种方法之一来清除断点：

- ① 按下 RESET 键

2 按下 STEP 键

③ 输入零到三个数字之后，紧接着压下 B·P 键，上述三种方法都是将断点标志 2FF4H 单元清零。因此，你也可以将 2FF4H 单元清零来清除断点。

(8) 用 EXEC 键连续执行程序。

EXEC 键可以用来连续执行 RAM、ROM 或 EPROM 的程序。先输入要执行的程序的起始地址，然后按下 EXEC 键，即从起始地址开始执行程序。一般设置 HALT 指令作为程序的结束。

按 键	显 示
MON	P
<u>2</u> <u>0</u> <u>0</u> <u>0</u>	2000
EXEC	

(9) 检查程序执行结果

按 键	显 示
MON	P
<u>A</u> REG	A A A
NON	P
<u>B</u> REG	b b b
MON	P
<u>C</u> REG	C C C
MON	P
<u>PC</u>	1 20 0 b

问题:

(1) 用 EXEC 键连续执行程序，为什么显示器不显示？若要显示器自动显示提示符‘P’，应如何修改程序？

(2) HALT 指令的作用？程序末尾不放 HALT 指令行吗？

(3) 总结一下用 EXEC，设断点和 STEP 键执行程序的步骤。并对比分别用三种方式执行程序时 PC 的内容和显示器的状态。

(4) 总结一下什么时候用 EXEC 键执行程序，什么时候用单步键，什么时候用设断点。

实验二 数据传送和交换

目的：掌握 TP801 的操作

练习传送指令和交换指令的使用

练习手工编译

器材：TP801 单板计算机一台

5V 直流稳压电源一台

练习2.1 8位数据传送

要求：将立即数 12H→A 累加器，

34H→B 寄存器，56H→C 寄存器，

78H→D 寄存器，9AH→E 寄存器，

BCH→H 寄存器，DEH→L 寄存器。

步骤：(1)将下列源程序手工编译成机器码程序（目标程序）。

(2)将目标程序由 MEM 和 NEXT 键盘命令导入本机内存单元（由 2000H 单元开始存放程序）。

(3)用 LAST 和 MEM 键，检查程序无误后，用 STEP 或 EXEC 键执行程序。

(4)按压 MON 键，回到监控状态，显示“P”提示符后，用寄存器代号键（数字键）、RFG 和 MON 键配合，检查 A、B、C、D、E、H、L 等寄存器的内容是否与你的设想一致。

程序：

地址(H)	机器码(H)	汇编语言程序
2000	3E12	LD A, 12H
2002		LD B, 34H
		LD C, 56H
		LD D, 78H
		LD E, 9AH
		LD H, 0BCH
		LD L, 0DEH
	76	HALT

练习2.2 16位数据传送

要求：将练习 2.1 中，各寄存器数据一一对应传送到辅助寄存器中去，并要求保留原寄存器中的数据不变。

步骤：除同练习 2.1 外，再用 REG' 和 MON 键配合检查程序执行前后辅助寄存器的内容变化。

程序：

地址(H)	机器码(H)	汇编语言程序
2000	3E 12	LD A, 12H
2002	01 56 34	LD BC, 3456H
2005		LD DE, 789AH
		LD HL, BCDEH
		PUSH AF
		PUSH BC
		PUSH DE
		PUSH HL
		EX AF, AF'
		EXX
		POP HL
		POP DE
		POP BC
		POP AF
		HALT

实验三 数据块传送

目的: 掌握相对转移时偏移量的计算方法
掌握标志位的检查方法
练习数据块传送指令的使用
熟悉寄存器对间接寻址和变址寻址方式

器材: TP801 单板计算机一台
5V 直流稳压电源一台

练习 3.1 相对转移时偏移量的计算及数据传送

要求: 通过下面程序的导入和执行, 掌握相对转移时偏移量的计算方法和标志位的检查方法, 下面程序的功能是: 将 2200H 到 2204H 存储区的源数据块, 传送到 2300H 起的存储区去。

程序:

地址(H)	机器码(H)	汇编语言程序
2000	21 00 22	LD HL, 2200H
2003	11 00 23	LD DE, 2300H
2006	0E 05	LD C, 5
2008	7E	LOOP: LD A, (HL)
2009	12	LD (DE), A
200A	23	INC HL
200B	13	INC DE
200C	0D	DEC C
200D	20 7	JR NZ, LOOP
200F	76	HALT

数据区

传送前			传送后		
源	目的		源	目的	
2200 A0	2300 × ×		2200 A0	2300 A0	
2201 A1	2301 × ×		2201 A1	2301 A1	
2202 A2	2302 × ×		2202 A2	2302 A2	
2203 A3	2303 × ×		2203 A3	2303 A3	
2204 A4	2304 × ×		2204 A4	2304 A4	

步骤:

(1) 操作键盘导入程序和数据, 但是, 200E 单元暂不输入。因为要将计算的偏移量填入此单元。

(2) 按如下步骤计算偏移量:

按 键	显 示	说 明
MON	P	
IX REG	4 × × × ×	显示 IX 寄存器内容
<u>2</u> <u>0</u> <u>0</u> <u>8</u>	4 20 0 8	目的地址→IX
MON	P	
IY REG	5 × × × ×	显示 IY 寄存器内容
<u>2</u> <u>0</u> <u>0</u> <u>D</u>	5 20 0 D	源地址→IY
MON		上档键有效
DI SP	FF F 9	偏移量为 F9 它是 -07H 的 2 的补码, FF 表示偏移量为负数即负跳转, 若 00 表示偏移量为正数
MON	P	
<u>2</u> <u>0</u> <u>0</u> <u>E</u> MEM	2 0 0 E F 9	200 E 单元已自动填入了偏移量 F9

(3) 从 2000H 地址起单步执行每一条指令, 结果填入空格内

地址(H)	汇编程序	结果
2000	LD HL, 2200H	HL =
2003	LD DE, 2300H	DE =
2006	LD C, 5	C =
2008	LOOP: LD A, (HL)	A =
2009	LD (DE), A	(DE) =
200A	INC HL	HL =
200B	INC DE	DE =
200C	DEC C	C =
		Z =
200D	JR NZ LOOP	PC =
		Z =
200F	RST 8	PC =

(4) 在 200A 设断点, 每执行一次循环体 (即从 2000H 起每按一次 EXEC 键), 检查下列寄存器的状态, 并记录入下表。

状态 次数	PC	HL	DE	C	Z	(DE)
1						
2						
3						
4						
5						
6						

(5) 改用 LDI 指令编制能完成上面功能的程序。并采用在 LDI 指令前设断点方法运行程序, 以观察 P/V 标志的变化。

(6) 改用 LDD, LDIR, LDDR 指令编制程序, 并在单板机上运行。

(7) 改用变址寻址编制程序, 并采用设断点方法运行程序, 观察变址寄存器的变化。

练习 3.2 数据传送指令的应用

要求: 将内存 2400H~2C00H 存入 FFH 数据

步骤: 略

程序:

地址(H)	机器码(H)	汇编语言程序
2000	21 00 24	LD HI, 2400H
2003		LD D, H
		LD E, L
		INC DE
		LD (HL), 0FFH
		LD BC, 0800H
		LDIR
		HALT

问题:

(1) 若要求从 2400H 开始到 2C00H 单元区域均填入空格符 (20H), 如何修改本程序来实现?

(2) 若改用 LDDR 指令, 如何修改本程序来实现?

(3) 采用数据块传送指令编制程序, 实现将 0000H~03FFH 1K 字节的监控程序块移动到 2000H~23FFH 存贮区中。

实验四 基本算术和逻辑运算

目的：熟悉算术和逻辑指令（如：ADD、ADC、INC、DEC、CPL、NEG、SUB、SBC 等）的功能；
进一步了解标志位的意义和指令对它的影响；练习数据在 CPU 寄存器和存储器中的分配

练习 4.1 8 位算术运算

要求：将 D 寄存器和 E 寄存器的数据相加，结果存于 HL 寄存器对。

程序：

地址(H)	机器码(H)	源程序
2000	_____	LD A, E
	_____	ADD A, D
	_____	LD L, A
	_____	LD A, 0
	_____	ADC A, 0
	_____	LD H, A
	_____	HALT

步骤：

- (1) 将源程序手工编译成目标程序（机器码）；
- (2) 将目标程序由键盘操作导入本机指定内存单元；
- (3) 按表格要求，由键盘命令予置 D、E 寄存器的初值和检查 HL 的内容，并将 HL 这时的内容填入表格中；
- (4) 采用单步执行程序方式（按动 STEP 键执行程序），将结果填入表中；
- (5) 请你仿效本例，予置不同的 D、E 初值，用手算方式求出结果，与本机运算结果相对比，看看是否一致。

予置初值		结 果		执行的指令
D	E	HL	HL CY,S,Z,P/V	
5AH	A6H			未执行指令时，
				ADD A, D
				ADC A, 0
				HALT
78H	CDH			
FEH	A8H			

问题：LD A, 0 和 ADC A, 0 这两条指令在本程序中的作用是什么？这两条指令和下面的一条指令 LD H, A 一起能否用别的指令来代替，也能完成同样的功能？

练习 4.2 16位算术运算

要求：将存储单元 2040H、2041H 的 16 位数据（高位在 2041H）与 DE 寄存器对的数据相加，假定和数也是 16 位数。将和数存于 2042H 和 2043H 单元（高位在 2043H 单元）。

程序：

地址(H)	机器码(H)	源程序
		ORG 2000H
		LD HL, (2040H)
		ADD HL, DE
		LD (2042H), HL
		HALT

步骤：

- (1) 将源程序编译成目标程序（机器码）；
- (2) 将目标程序装入指定内存单元；
- (3) 由键盘命令予置 DE 寄存器初值和 2040H、2041H 初值；
- (4) 采用 STEP 或 EXFC 方式执行程序，将结果填入表中；
- (5) 请你仿效本例，予置不同 DE、与 2040H、2041H 数值，用手算求出的结果与本机运算相比，看看是否一致。

予置值				执行结果	
寄存器		存储器		存储器	存储器
D	E	2041H	2040H	2043H	2042H
34	56H	78H	9AH		
A5	47H	19H	F4H		

问题：• 若 DE 寄存器是被减数，2041H 和 2040H 数据为减数，如何修改本程序进行减法运算？要求将差数存于内存 2043H、2042H 单元。

• 若本题两个 16 位数相加，和数大于 16 位，请问如何编制程序？

练习 4.3 逻辑运算 (一)

要求: 将 2100H 单元的数据求反码和求补码, 并将求得的反码存入 2101H, 补码存入 2102H 单元。

程序:

地址(H)	机器码(H)	源程序
		ORG 2000H
		LD HL, 2100H
		LD A, (HL)
		LD B, A
		CPL
		INC HL
		LD (HL), A
		LD A, B
		NEG
		INC HL
		LD (HL), A
		HALT

步骤:

- (1) 将源程序编译成目标程序(机器码);
- (2) 将目标程序装入指定内存单元;
- (3) 采用 STFP 或 EXEC 方式执行程序。执行程序前应予置 2100H 单元的初值。
- (4) 检查内存单元 2101H 和 2102H 的数据与您手算的结果是否一致?
- (5) 反复予置 2100H 单元的数据, 检查结果是否与你予想的都一致?

练习 4.4 逻辑运算 (二)

要求: 将存储单元 2040H 的内容拆成两段, 每段 4 位, 并将它们分别存入存储单元 2041H 和 2042H, 即存储单元 2040H 的高 4 位存入存储单元 2041 的低 4 位, 存储单元 2040H 的低 4 位存入存储单元 2042H 的低 4 位。并将 2041H 和 2042H 的高 4 位清零。

程序:

地址(H)	机器码(H)	源程序
		LD HL, 2040H
		LD A, (HL)
		LD B, A
		RRA ; 数据右移 4 次
		RRA
		RRA
		RRA

地址(H)	机器码(H)	源程序
		AND 00001111B
		INC HL
		LD (HL), A
		LD A, B
		AND 00001111B
		INC HL
		LD (HL), A
		HALT

步骤:

- (1) 将源程序译成目标程序(机器码)
- (2) 将目标程序装入指定内存单元
- (3) 将数据存入 2040H 单元。
- (4) 采用 STEP 或 EXEC 方式执行程序。
- (5) 检查内存 2041H 和 2042H 内容, 看与你手算的结果是否一致?
- (6) 反复予置 2040H 单元的数据, 检查结果是否与你想的都一致?