

手把手教你学单片机(八)

[套件供应]

■周兴华

栈操作指令、空操作指令及伪指令、字节交换指令学习

1. 栈操作指令:

PUSH direct (堆栈指令) 其作用是先将栈指针 SP 的内容加 1, 然后将直接寻址单元中的数压入到 SP 所指的单元中。若数据已入栈, 则 SP 指向最后堆入数据所在的存储单元(即指向栈顶)。

POP direct (出栈指令) 其作用是先将栈指针 SP 所指出单元的内容送入直接寻址单元中, 然后将栈指针 SP 的内容减 1, 此时 SP 指向新的栈顶。

使用堆栈时, 一般需设定 SP 的初始值。堆栈原则上可以设在内部 RAM 的任意区域, 但为使用方便, 一般设在 30H~7FH。另外, 需注意留出足够的存储单元作栈区, 否则可能发生数据重叠, 引起程序混乱。由于入栈的第一个数必须存放在 SP+1 的存储单元, 故实际栈顶是在 SP+1 所指出的单元。

2. 空操作指令:

NOP 这是一条单字节指令, 它不作任何操作, 但要占用一个机器周期的时间, 常用于延时或等待。

3. 伪指令:

伪指令又叫做汇编控制指令, 它是在汇编过程中起作用的指令, 用来对汇编过程进行某种控制, 或者对符号、标号赋值。伪指令和指令是完全不同的, 在汇编过程中, 伪指令并不产生可执行的目标代码, 大部份伪指令甚至不会影响存储器中的内容。

ORG(汇编起始命令)

ORG 的功能为规定下面目标程序的起始地址。

格式:[标号:] ORG 16 位地址。
其中括号内是任选项, 可以没有。

ORG 伪指令总是出现在每段

源程序或数据块的起始位置, 故称为汇编起始命令。在一个源程序中, 可以多次使用 ORG 指令, 以规定不同的程序段的起始位置。但所规定的地址应该是从小到大, 而且不允许有重叠, 即不同的程序段之间不能有重叠。一个源程序若不用 ORG 指令开始, 则从 0000H 开始存放目标码。

如:

```
ORG 0030H
MAIN: MOV A,#20H
      :
```

表示主程序从 0030H 单元开始存放。

END(汇编结束命令)

格式:[标号:] END 其中括号内是任选项, 可以没有。

END 是汇编语言源程序的结束标志, 在 END 以后所写的指令, 汇编程序都不予处理。一个源程序只能有一个 END 命令。在同时包含有主程序和子程序的源程序中, 也只能有一个 END 命令, 并放到所有指令的最后。否则, 在 END 之后就有一部分指令不能被汇编(编译)。

EQU(等值命令)

EQU 的功能是将一个数或者特定的汇编符号赋予规定的字符名称。

格式: 字符名称 EQU 数或汇编符号。注意, 这里使用的是“字符名称”, 不是标号, 而且也不用“:”来作分隔符, 若加上“:”反而被汇编程序认为是一种错误。用 EQU 指令赋值以后的字符名称, 可以用作数据地址、代码地址、位地址或者直接当作一个立即数使用。因此, 给字符名称所赋的值可以是 8 位数, 也可以是 16 位二进制数。使用 EQU 伪指令时必须先赋值, 后使用。而不能先使用, 后赋值。

如:

```
FA EQU R1
```

;

表示经定义后, 允许在指令中用 FA 代替 R1。

又如:

```
ADD EQU 1000H
```

;

表示经定义后, 即给 ADD 赋以地址值 1000H。

DB 或 **DEFB**(定义字节命令)

DB 的功能是从指定的地址单元开始, 定义若干个字节作为内存单元的内容。

格式: [标号:] DB 字节形式的数据表

这个伪指令是在程序存储器的某一部分存入一组规定好的 8 位二进制数, 或者是将一个数据表格存入程序存储器。该伪指令在汇编以后, 将影响程序存储器的内容。

DB 命令所确定的单元地址可以由下述两种方法之一来确定: 若 DB 命令是紧接着其它源程序的, 则由源程序最后一条指令的地址加上该指令的字节数来确定; 由 ORG 命令来规定首地址。

如:

```
ORG 0800H
TAB: DB 53H,20H,14H,22H,
      66H,98H,87H,60H,90H,80H
      END
```

表示首地址从 0800H 开始的一组数据表格。

DW 或 **DEFW**(定义字命令)

其功能是从指定地址开始, 定义若干个 16 位数据。

格式: [标号:] DW 16 位数据表

每个 16 位数要占 ROM 的两个单元, 在 51 系列单片机中, 16 位二进制数的高 8 位先存入(低地址字节), 低 8 位后存入(高地址字节)。

如:

```
ORG 0800H
HETAB: DW 5320H,1422H,66H
      END
```

表示表示首地址从 0800H 开始的一组 16 位数据表格。

(0800)=53H (0801)=20H

(0802)=14H (0803)=22H
(0804)=00H (0805)=66H

DB、DW 伪指令都只对程序存储器起作用，即不能用它们来对数据存储器的内容进行赋值或其它初始化的工作。

DS 或 DEFS(预留存储区命令)

其功能是从指定地址开始，定义一个存储区，以备源程序使用。存储区预留的存储单元数由表达式的值决定。

格式: [标号:] DS 表达式值

如:

ORG 0500H

TEMP: DS 10

⋮

即由 0500H 地址开始保留连续的 10 个存储单元存储区。

BIT(定义位命令)

其功能用于给字符名称定义位地址。

格式: 字符名称 BIT 位地址

如:

F1 BIT P1.1

经定义后，允许在指令中用 F1 代替 P1.1。

DATA(定义数据地址命令)

其功能用于给字符名称定义 16 位地址。通常用来定义数据地址。

格式: 字符名称 DATA 16 位地址

4.字节交换指令

XCH A,Rn 表示将寄存器

Rn 中的内容与累加器 A 的内容相互交换。

XCH A,direct 表示把直接寻址单元 direct 中的内容与累加器 A 的内容相互交换。

XCH A,@Ri 表示将寄存器 Ri 中内容作为地址的单元内容(寄存器间接寻址单元)与累加器 A 的内容相互交换。

XCHD A, @Ri 表示将寄存器 Ri 间接寻址单元的低 4 位内容与累加器 A 的低 4 位内容相互交换，而各自的高 4 位维持不变。

内部 RAM

接下来做实验，具体认识所学这些指令的作用。

现在我们做实验，在 S2 板上实现：按下 0~9 键后，将立即数 0~9 输入到累加器 A 中，左边第二个数码管用作为输入显示。单片机根据累加器 A 中的数查其平方表，并且在右边的两个数码管上显示出来。在我的文档中建立一个文件目录(S13)，然后建立 S13.uv2 的工程项目，最后建立源程序文件(S13.asm)。

输入下面的程序：

```

序号:1      ORG 0000H
2          AJMP MAIN;
3          ORG 030H
4 MAIN:    LCALL SCAN_KEY;
5          JZ MAIN;
6          LCALL DEL10MS;
7          LCALL SCAN_KEY;
8          JZ MAIN;
9          MOV P3,#7FH;
10         JNB P3.0,L3;
11         JNB P3.1,L6;
12         JNB P3.2,L9;
13         MOV P3,#0BFH;
14         JNB P3.0,L2
15         JNB P3.1,L5
16         JNB P3.2,L8
17         JNB P3.3,L0
18         MOV P3,#0DFH;
19         JNB P3.0,L1
20         JNB P3.1,L4
21         JNB P3.2,L7
22         AJMP MAIN;
23 L0:     MOV A,#00H;
24         AJMP GOON;
25 L1:     MOV A,#01H;
26         AJMP GOON;
27 L2:     MOV A,#02H;
28         AJMP GOON;
29 L3:     MOV A,#03H;
30         AJMP GOON;
31 L4:     MOV A,#04H;
32         AJMP GOON;
33 L5:     MOV A,#05H;
34         AJMP GOON;
35 L6:     MOV A,#06H;
36         AJMP GOON;
37 L7:     MOV A,#07H;
38         AJMP GOON;
39 L8:     MOV A,#08H;
40         AJMP GOON;
41 L9:     MOV A,#09H;
42 GOON:   MOV 30H,A;
43         MOV DPTR,#DIS_TAB
44         MOVC A,@A+DPTR;
45         MOV P2,A;
46         MOV A,30H;
47         MOV DPTR,#TAB
48         MOVC A,@A+DPTR;
49         MOV DPTR,#DIS_TAB
50         PUSH ACC;
51         ANL A,#0FH;
52         MOVC A,@A+DPTR;
53         MOV P0,A;
54         POP ACC;
55         SWAP A;

```

```

56         ANL A,#0FH;
57         MOVC A,@A+DPTR;
58         MOV P1,A;
59         LCALL DEL10MS;
60         LJMP MAIN
61         ORG 0200H
62 SCAN_KEY:MOV P3,#0FH;
63         MOV A,P3;
64         ORL A,#0F0H;
65         CPL A;
66         RET
67         ORG 0250H
68 DEL10MS: MOV R5,#0BH
69 F1:     MOV R7,#02H
70 F2:     MOV R6,#0FFH
71 F3:     DJNZ R6,F3
72         DJNZ R7,F2
73         DJNZ R5,F1
74         RET
75         ORG 0300H
76 TAB:   DB 00H,01H,04H,09H,16H
77         DB 25H,36H,49H,64H,81H
78         ORG 0350H
79 DIS_TAB: DB 0C0H,0F9H,
80         DB 0A4H,0B0H,099H,092H,082H,0F8H
81         DB 080H,090H,088H,083H,
82         DB 0C6H,0A1H,086H,08EH
83         END

```

编译通过后，将 S13 文件夹中的 hex 文件烧录到 89C51 芯片中，将芯片插入到 S2 型数码管试验板上，按下 0 键，左边第二个数码管显示 0，右边的两个数码管上显示 00；按下 1 键，左边第二个数码管显示 1，右边的两个数码管上显示 01；……按下 9 键，左边第二个数码管显示 9，右边的两个数码管上显示 81。实现了根据输入数查其平方表的功能。

我们对程序进行分析解释。

序号 1(程序解释，下同):程序开始。

序号 2:跳转到 MAIN 主程序处。

序号 3:主程序 MAIN 从地址 0030H 开始。

序号 4:调用键扫描子程序。

序号 5:若累加器为 0，跳转到 MAIN 处。

序号 6:调用 10ms 延时子程序，避开键抖动干扰。

序号 7:再调用键扫描子程序。

序号 8:若累加器 A 为 0，说明无键输入，跳转到 MAIN 处。否则说明有键输入，顺序执行。

序号 9:向 P3 口送数 7FH，准备读取键输入状态。

序号 10:有 3 键按下转 L3。

序号 11:有 6 键按下转 L6。

序号 12:有 9 键按下转 L9。

序号 13:向 P3 口送数 BFH，准备读取键输入状态。

序号 14:有 2 键按下转 L2。

序号 15:有 5 键按下转 L5。

序号 16:有 8 键按下转 L8。

序号 17:有 0 键按下转 L0。

序号 18:向 P3 口送数 DFH，准备读取键输入状态。

序号 19:有 1 键按下转 L1。

序号 20:有 4 键按下转 L4。

序号 21: 有 7 键按下转 L7。
 序号 22: 无有效键按下, 跳转到标号 MAIN 处。
 序号 23: 向累加器 A 送立即数 00。
 序号 24: 跳转到标号 GOON 处。
 序号 25: 向累加器 A 送立即数 01。
 序号 26: 跳转到标号 GOON 处。
 序号 27: 向累加器 A 送立即数 02。
 序号 28: 跳转到标号 GOON 处。
 序号 29: 向累加器 A 送立即数 03。
 序号 30: 跳转到标号 GOON 处。
 序号 31: 向累加器 A 送立即数 04。
 序号 32: 跳转到标号 GOON 处。
 序号 33: 向累加器 A 送立即数 05。
 序号 34: 跳转到标号 GOON 处。
 序号 35: 向累加器 A 送立即数 06。
 序号 36: 跳转到标号 GOON 处。

序号 37: 向累加器 A 送立即数 07。
 序号 38: 跳转到标号 GOON 处。
 序号 39: 向累加器 A 送立即数 08。
 序号 40: 跳转到标号 GOON 处。
 序号 41: 向累加器 A 送立即数 09。
 序号 42: 将累加器 A 内容送 30H 单元暂存。
 序号 43: 将数码管字段码数据表格的首地址 (0350H) 存入 16 位数据地址指针 DPTR 中。
 序号 44: 根据 A 中内容查表。
 序号 45: 查表结果送 P2 口显示。
 序号 46: 将 30H 单元内容送回累加器 A。
 序号 47: 将 0-9 平方表数据表格的首地址 (0300H) 存入 16 位数据地址指针 DPTR 中。
 序号 48: 根据 A 中内容查表。
 序号 49: 将数码管字段码数据表格的首地址 (0350H) 存入 16 位数据地址指针 DPTR 中。

序号 50: 将累加器 A 中内容压栈。
 序号 51: 屏蔽累加器 A 高 4 位。
 序号 52: 根据 A 中内容查表。
 序号 53: 累加器 A 中内容送 P0 口显示。
 序号 54: 恢复累加器 A 中原内容。
 序号 55: 交换累加器的高、低 4 位。
 序号 56: 屏蔽累加器 A 高 4 位。
 序号 57: 根据 A 中内容查表。
 序号 58: 累加器 A 中内容送 P1 口显示。
 序号 59: 调用延时子程序, 维持数码管点亮。
 序号 60: 跳转到 MAIN 处循环执行。
 序号 61: 键扫描子程序从地址 0200H 开始。
 序号 62: 键扫描子程序开始, 向 P3 口送数 0FH, 准备读键输入。
 序号 63: 将 P3 口状态读入累加器 A 中。
 序号 64: 累加器 A 与立即数 FOH 相或, 结果送

1. 传送、交换、栈操作指令

助记符	功能说明	字节数	振荡周期
MOV A,Rn	寄存器传送到累加器	1	12
MOV A,direct	直接字节传送到累加器	2	12
MOV A,@Ri	间接 RAM 传送到累加器	1	12
MOV A,#data	立即数传送到累加器	2	12
MOV Rn,A	累加器传送到寄存器	1	12
MOV Rn,direct	直接字节传送到寄存器	2	24
MOV Rn,#data	立即数传送到寄存器	2	12
MOV direct,A	累加器传送到直接字节	2	12
MOV direct,Rn	寄存器传送到直接字节	2	24
MOV direct,direct	直接字节传送到直接字节	3	24
MOV direct,@Ri	间接 RAM 传送到直接字节	2	24
MOV direct,#data	立即数传送到直接字节	3	24
MOV @Ri,A	累加器传送到间接 RAM	1	12
MOV @Ri,direct	直接字节传送到间接 RAM	2	24
MOV @Ri,#data	立即数传送到间接 RAM	2	12
MOV DPTR,#data16	16 位数加载到数据指针	3	24
MOV A,@A+DPTR	代码字节传送到累加器	1	24
MOV A,@A+PC	代码字节传送到累加器	1	24
MOVB A,@Ri	外部 RAM(8 位地址)传送到 ACC	1	24
MOVX A,@DPTR	外部 RAM(16 位地址)传送到 ACC	1	24
MOVX @Ri,A	ACC 传送到外部 RAM(8 位地址)	1	24
MOVX @DPTR,A	ACC 传送到外部 RAM(16 位地址)	1	24
PUSH direct	直接字节压到堆栈	2	24
POP direct	从栈中弹出直接字节	2	24
XCH A,Rn	寄存器和累加器交换	1	12
XCH A,direct	直接字节和累加器交换	2	12
XCH A,@Ri	间接 RAM 和累加器交换	1	12
XCHD A,@Ri	间接 RAM 和累加器交换低 4 位字节	1	12
SWAP A	累加器内部高、低 4 位交换	1	12

3. 转移指令

助记符	功能说明	字节数	振荡周期
ACALL addr11	绝对调用子程序	2	24
LCALL addr16	长调用子程序	3	24
RET	从子程序返回	1	24
RETI	从中断返回	1	24
AJMP addr11	绝对转移	2	24
LJMP addr16	长转移	3	24
SJMP rel	短转移(相对转移)	2	24
JMP @A+DPTR	相对 DPTR 的间接转移	1	24
JZ rel	累加器为 0 则转移	2	24
JNZ rel	累加器为非 0 则转移	2	24
CJNE A,direct,rel	比较直接字节和 ACC 不相等则转移	3	24
CJNE A,#data,rel	比较立即数和 ACC 不相等则转移	3	24
CJNE Rn,#data,rel	比较立即数和寄存器不相等则转移	3	24
CJNE @Ri,#data,rel	比较立即数和间接 RAM 不等则转移	3	24
DJNZ Rn,rel	寄存器减 1, 不为 0 则转移	3	24
DJNZ direct,rel	直接字节减 1, 不为 0 则转移	3	24
NOP	空操作	1	12

2. 算术、逻辑运算指令

(下转 21 页)

助记符	功能说明	字节数	振荡周期
ADD A,Rn	寄存器加到累加器	1	12
ADD A,direct	直接字节加到累加器	2	12
ADD A,@Ri	间接 RAM 加到累加器	1	12
ADD A,#data	立即数加到累加器	2	12
ADDC A,Rn	寄存器加到累加器(带进位)	1	12
ADDC A,direct	直接字节加到累加器(带进位)	2	12
ADDC A,@Ri	间接 RAM 加到累加器(带进位)	1	12
ADDC A,#data	立即数加到累加器(带进位)	2	12
SUBB A,Rn	ACC 减去寄存器(带借位)	1	12
SUBB A,direct	ACC 减去直接字节(带借位)	2	12
SUBB A,@Ri	ACC 减去间接 RAM(带借位)	1	12
SUBB A,#data	ACC 减去立即数(带借位)	2	12
INC A	累加器加 1	1	12
INC Rn	寄存器加 1	1	12
INC direct	直接字节加 1	2	12
INC @Ri	间接 RAM 加 1	1	12
DEC A	累加器减 1	1	12
DEC Rn	寄存器减 1	1	12
DEC direct	直接地址字节减 1	2	12
DEC @Ri	间接 RAM 减 1	1	12
INC DPTR	数据指针加 1	1	24
MUL AB	A 和 B 寄存器相乘	1	48
DIV AB	A 寄存器除以 B 寄存器	1	48
DA A	累加器十进制调整	1	12
ANL A,Rn	寄存器“与”到累加器	1	12
ANL A,direct	地址字节“与”到累加器	2	12
ANL A,@Ri	间接 RAM“与”到累加器	1	12
ANL A,#data	立即数“与”到累加器	2	12
ANL direct,A	累加器“与”到直接字节	2	12
ANL direct,#data	立即数“与”到直接字节	3	24
ORL A,Rn	寄存器“或”到累加器	1	12
ORL A,direct	直接字节“或”到累加器	2	12
ORL A,@Ri	间接 RAM“或”到累加器	1	12
ORL A,#data	立即数“或”到累加器	2	12
ORL direct,A	累加器“或”到直接字节	2	12
ORL direct,#data	立即数“或”到直接字节	3	24
XRL A,Rn	寄存器“异或”到累加器	1	12
XRL A,direct	直接字节“异或”到累加器	2	12
XRL A,@Ri	间接 RAM“异或”到累加器	1	12
XRL A,#data	立即数“异或”到累加器	2	12
XRL direct,A	累加器“异或”到直接字节	2	12
XRL direct,#data	立即数“异或”到直接字节	3	24
CLR A	累加器清 0	1	12
CPL A	累加器取反	1	12
RL A	累加器循环左移	1	12
RLC A	经过进位的累加器循环左移	1	12
RR A	累加器循环右移	1	12
RRC A	经过进位的累加器循环右移	1	12

5. 线径的确定:(功率因素按 0.9 计)200W 变压器中的电流 $I=200W \div 220V \div 0.9=1.01A$ 。取漆包线的电流密度为 $2A/mm^2$, 则应选用的漆包线截面 $S=1.01A \div 2A \cong 0.505mm^2$ 。可选用标准线径 $\phi=\sqrt{0.505} mm^2 \div 3.1416 \times 2=0.8mm$ 左右的漆包线。

组装:

PCB 图见图 2, 考虑到因采用了双刀双投继电器各引脚的间距较小、故用双面板设计, 以尽量使高、低压间的电气间隙大于 3mm, 达到安全标准的要求。CZ1 的 1-2 脚为输出引脚, 3-4 脚为市电输入引脚。CZ2 是两个指示灯的引出脚。

CZ3 的 1-4 脚分别接入变压器的 170V、198V、220V、242V 抽头。

元件选用

C1 的耐压不能低于 $265V \times 1.414=375V$ 选用 400V 以上为佳, 且其输出电流要保证 100mA (两只继电器同时工作、约 80mA, 再加上其它元器件的耗电)。

继电器要选用触头过流能力大

于实际工作中过流量, 且要求接触压力大、接触电阻小于 $50m\Omega$ 为佳。

4.C5-C8 要选用耐压 100V 以上的电解电容。

5. 其余元件无特殊要求。

编者点评

此电路作者称三年前曾制作过, 近日整理、修改了当时的设计图纸, 写成此文。此电路设计比较合理, 适用于缺电的农村地区, 比较实用。但作者可能未对此电路进行过实验测试, 因而在前面有关电路原理的阐述与后面有关比较电压的逐个调定步骤中均出现了同样的失误。

此电路是利用对交流电压(经 D5、C9 半波整流滤波成直流)采样后与一个基准电压进行比较产生继电器驱动信号的, 四个比较器的基准电压都是 6V (因稳压二极管参数的离散性, 实际制作时 D5 的稳压值可能会有较大的偏差、但均可通过调采样电位器解决, 不会影响电路的正常工作), 而在市电的

不同情况下使相应继电器动作的采样电位器分压比、均可在调试时准确调定。这里选用四运放 LM324 作比较器, 不管选用任何型号的运放、或任何型号的比较器如 LM399, 因都是开环工作, 相对其高达约百万倍的开环增益、其翻转信号均非常小。如不管哪个输入端的基准电压为 6V, 只要另一个输入端比 6V 稍稍高或低一点点(约 10uV 左右)比较器会立即翻转。考虑到 LM324、LM399 这类器件的输入失调电压均在正负几 mV 左右, 对均是 6V 的基准电压, 其翻转信号仅会比 6V 高或低几 mV (是高还是低与基准电压接哪个输入端无关, 仅与该运放的失调电压是正或负相关), 而不会像文中所述要相差正负 0.1V。在比较电压的逐个调定步骤中“VR1(VR2)中点缓缓上调到 6.1V、K1(K2)刚好吸合……VR3(VR4)中点缓缓下调到 5.9V、K3(K4)刚好吸和……”的结论可能出自想象、而决非是实验的结果。

(上接 24 页)

4. 布尔指令集

助记符	功能说明	字节数	振荡周期
CLR C	清进位	1	12
CLR bit	清直接寻址位	2	12
SETB C	进位位置位	1	12
SETB bit	直接寻址位置位	2	12
CPL C	进位位取反	1	12
CPL bit	直接寻址位取反	2	12
ANL C, bit	直接寻址位“与”到进位位	2	24
ANL C, /bit	直接寻址位的反码“与”到进位位	2	24
ORL C, bit	直接寻址位“或”到进位位	2	24
ORL C, /bit	直接寻址位的反码“或”到进位位	2	24
MOV C, bit	直接寻址位传送到进位位	2	12
MOV bit, C	进位位传送到直接寻址位	2	24
JC rel	如果进位位为 1 则转移	2	24
JNC rel	如果进位位为 0 则转移	2	24
JB bit, rel	如果直接寻址位为 1 则转移	3	24
JNB bit, rel	如果直接寻址位为 0 则转移	3	24
JBC bit, rel	如果直接寻址位为 1 则转移并清该位	3	24

A 中。
序号 65:累加器 A 取反。
序号 66:返回。
序号 67:延时子程序从地址 0250H 开始。
序号 68-74:延时 10ms 子程序。
序号 75:数据表格从地址 0300H 开始。
序号 76-77:0-9 平方表数据表格内容。
序号 78:数据表格从地址 0350H 开始。

序号 79-80:数码管字段码数据表格内容。
序号 81:程序结束。

到此为止, 我们已将 MCS-51 系列单片机指令系统中的 111 条汇编指令全部学完。为方便读者查找, 现将其分四大类(传送、交换、

栈操作指令;算术、逻辑运算指令;转移指令和布尔指令集)按功能、字节数及周期数列表如下。

(下一讲学习 MCS-51 定时器/计数器及有关实验)。

配文优惠邮购(每次邮费保价费 12 元):Keil 51 Windows 集成开发环境(已汉化光盘, 邮购代号:K1):46 元。TOP851 多功能编程器(邮购代号:B1):400 元。LED 输出试验板(邮购代号:S1):90 元。LED 数码管输出试验板(邮购代号:S2):140 元。5V 高稳定专用稳压电源(邮购代号:D1):35 元。。邮购时只需在附言栏中写明邮购代号及数量并附上联系电话即可。

邮购地址:201103 上海市闵行区莲花路 2151 弄 57 号 201 室

联系人:吕超亚

电话:021-64066571 13044152947

技术支持 E-mail:zch2151@sohu.com