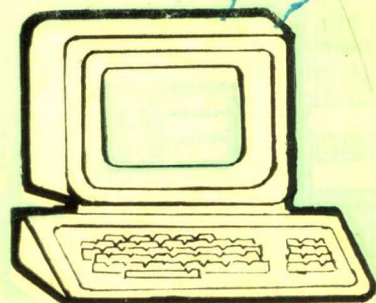


**Z-80**

# 微 型 电 脑

## 汇编语言程序设计 与实用子程序手册



江苏工学院微型机应用研究室

1984.8

# 前 言

近几年来，Z—80微型电脑的应用有了很大的发展，如CMC—80、TP—801等单板微型机日益广泛地用于生产过程的自动检测、实时控制、数据的采集和处理。由于单板机内存容量小，不可能用高级语言编制各种应用程序，这就需要用户对Z80的机器语言和汇编语言有较深入的了解。但用这类语言编制程序难度较大，现成的参考程序也不多，这对Z80类型单板机的进一步推广普及造成困难，不少用户迫切希望有关单位和部门能提供一些标准的实用子程序，以使用户编制程序时选用和参考。为此，我们从去年三月份开始，陆续收集和编制了各种典型的实用子程序103个，编辑成《CMC—80微型电脑实用子程序手册》于同年七月作为内部资料出版。一年来经用户使用，证明本《手册》所提供的程序，有广泛的实用意义，为用户用汇编语言编制各种应用程序带来了很大方便，深受读者欢迎，并纷纷来函要求再版。一年来我们根据《手册》使用的情况和读者的要求决定在原《手册》的基础上增写了一部份程序设计的内容，为初学者介绍了一些汇编语言程序设计的基础和技巧；同时对原有的103个子程序作了进一步的说明，便于读者阅读；在程序数量上也有原有的103个扩充到124个，内容更为广泛、实际。书中所有程序仍保持了原《手册》的特点，即不仅有程序清单，还附有较详细的说明和框图，使之结构清晰，阅读方便，而且全部程序都在CMC—80单板机上调试通过，因此内容正确可靠。

本《手册》对Z80类型单板机用户有一定的参考价值，亦可供高等、中等学校及培训班的师生阅读参考。

本《手册》由陈金华主编，参加编写工作的还有宋顺林、李芷、毕建良、赵耀华、俞炘、盛平、王勤等同志。袁云良同志为“手册”的出版工作做了不少工作，赵守先同志为《手册》设计了封面。

由于编者水平有限，书中内容难免有不当之处，敬请读者批评指正。

编 者

一九八四年七月

# 目 录

## 第一部分 Z80汇编语言程序设计概论

### 第一章 Z80指令系统

- 1—1 指令格式.....( 1 )
- 1—2 寻址方式.....( 2 )
- 1—3 指令系统.....( 6 )

### 第二章 Z80汇编语言程序设计

- 2—1 机器语言和汇编语言.....( 25 )
- 2—2 汇编语言的结构.....( 26 )
- 2—3 伪指令.....( 27 )
- 2—4 汇编语言程序设计的特点和步骤.....( 28 )
- 2—5 汇编语言程序设计.....( 29 )

附录：按字母顺序排列的Z80指令系统表.....( 36 )

## 第二部分 Z80实用于程序

### 第一章 代码转换程序

- 1—1 十六进制→ASCII 码转换程序.....( 46 )
- 1—2 ASCII码→十六进制数转换程序.....( 46 )
- 1—3 十六进制数的 ASCII 码→四位二进制数转换程序.....( 47 )
- 1—4 ASCII 码→二进制转换程序.....( 49 )
- 1—5 ASCII 码→BCD 码转换程序.....( 49 )
- 1—6 BCD 码→ASCII 码转换程序.....( 50 )
- 1—7 BCD 码→二进制转换程序.....( 52 )
- 1—8 十六位二进制数→BCD 码数的转换程序—I.....( 53 )
- 1—9 二进制数→ASCII 码字符串转换程序.....( 55 )
- 1—10 一位十六进制数→ASCII 码转换程序.....( 56 )
- 1—11 十进制(BCD码)→七段代码转换程序.....( 56 )
- 1—12 十六进制→七段代码转换程序.....( 57 )

|      |                      |      |
|------|----------------------|------|
| 1—13 | 十六位二进制数→BCD码数转换程序—II | (58) |
| 1—14 | 一组无符号的十进制数→二进制数转换程序  | (61) |
| 1—15 | 格雷码→二进制码转换程序         | (62) |
| 1—16 | 二进制码→格雷码转换程序         | (64) |

## 第二章 二进制定点运算

|      |                |      |
|------|----------------|------|
| 2—1  | 多字节加法          | (66) |
| 2—2  | 多字节减法          | (66) |
| 2—3  | 单字节无符号数乘法      | (67) |
| 2—4  | 单字节有符号数乘法(纯小数) | (68) |
| 2—5  | 单字节有符号数乘法(整数)  | (69) |
| 2—6  | 多字节无符号数乘法      | (71) |
| 2—7  | 多字节有符号数乘法      | (73) |
| 2—8  | 单字节无符号数除法(整除)  | (76) |
| 2—9  | 单字节有符号数除法(一)   | (78) |
| 2—10 | 单字节有符号数除法(二)   | (80) |
| 2—11 | 多字节无符号数除法(整数)  | (82) |
| 2—12 | 多字节有符号数除法(整数)  | (85) |

## 第三章 BCD码运算

|      |                  |       |
|------|------------------|-------|
| 3—1  | 单字节BCD码加法        | (89)  |
| 3—2  | 多字节BCD码加法        | (89)  |
| 3—3  | 单字节BCD码减法        | (90)  |
| 3—4  | 多字节BCD码减法        | (90)  |
| 3—5  | 一位BCD码乘法         | (91)  |
| 3—6  | 单字节BCD码乘法        | (92)  |
| 3—7  | 多字节BCD码乘法        | (94)  |
| 3—8  | 单字节BCD码除法        | (96)  |
| 3—9  | 多字节BCD码除法(一)     | (99)  |
| 3—10 | 多字节BCD码除法(二)     | (102) |
| 3—11 | 求单字节BCD码整数平方根子程序 | (104) |
| 3—12 | 求双字节BCD码整数平方根子程序 | (106) |
| 3—13 | BCD码浮点运算加法       | (107) |
| 3—14 | BCD码浮点运算减法       | (111) |
| 3—15 | BCD码浮算运算除法       | (116) |
| 3—16 | BCD码浮点运算乘法       | (121) |

#### 第四章 浮点运算及函数

|      |                           |       |
|------|---------------------------|-------|
| 4—1  | 整数转化 成浮 点数 FLOT.....      | (129) |
| 4—2  | 浮点数规格化子 程序 FNOR.....      | (130) |
| 4—3  | 浮点数加 法子 程序 FADD.....      | (133) |
| 4—4  | 浮点数减法子程序FSUB.....         | (137) |
| 4—5  | 浮点数乘法 子程序 FMUL.....       | (137) |
| 4—6  | 浮点 数除法子程序FDIV.....        | (140) |
| 4—7  | 浮点数平方根子程序FSQT.....        | (145) |
| 4—8  | 浮点数绝对值子程序FABS.....        | (149) |
| 4—9  | 存浮点 数子 程序 FGET.....       | (150) |
| 4—10 | 浮点数退栈子程序FPOP.....         | (151) |
| 4—11 | 浮点数进栈子程序FPSH.....         | (152) |
| 4—12 | 浮点数求补子程 FNEG 及 FNGX.....  | (152) |
| 4—13 | 浮点数求积(商)符号 子程序 FSIG.....  | (154) |
| 4—14 | 浮点数对阶右移子程序FASR.....       | (155) |
| 4—15 | 多字节左移一位子程序FRAL及 FRL0..... | (157) |
| 4—16 | 初始化子程序 FSPS.....          | (157) |
| 4—17 | 单字节带进 位加法 子程序 FADS.....   | (158) |
| 4—18 | 单字节带进位减 法子程序 FSBS.....    | (158) |
| 4—19 | B、C、D寄存器 左移 子程序 RLBD..... | (158) |
| 4—20 | 浮点数平方子程序FSQU.....         | (159) |
| 4—21 | 浮点数立方子程序 FCUBE.....       | (159) |
| 4—22 | 取浮 点数子 程序 FPUT.....       | (160) |
| 4—23 | 角度 化弧度子 程序 CDTR.....      | (161) |
| 4—24 | 弧度化 角度子 程序 CRTD.....      | (161) |
| 4—25 | 角 度余弦 函数 FCOD.....        | (161) |
| 4—26 | 角 度正 弦函数FSID.....         | (162) |
| 4—27 | 弧度余弦函数FDOS.....           | (162) |
| 4—28 | 弧度正弦函数FSIN.....           | (162) |
| 4—29 | 正 切 函数 FTAN.....          | (167) |
| 4—30 | 常用对 数函数 FLOG.....         | (168) |
| 4—31 | 自然对数 函数 FLN.....          | (168) |
| 4—32 | 反正 切 函数 FATN.....         | (173) |
| 4—33 | 指数函数FEXP.....             | (178) |
| 4—34 | 多项式 计算 子程序 FCAL.....      | (182) |
| 4—35 | 取整 函数 FIXF.....           | (184) |
| 4—36 | 取整 子程序 FIXI.....          | (184) |

|      |                     |       |
|------|---------------------|-------|
| 4—37 | B、C、D 寄存器右移子程序 RRBD | (186) |
| 4—38 | 浮点数比较子程序 FCMP       | (186) |

## 第五章 检索和排序

|      |                    |       |
|------|--------------------|-------|
| 5—1  | 找最大数(无符号)          | (188) |
| 5—2  | 找最大数(带符号)          | (189) |
| 5—3  | 顺序检索               | (190) |
| 5—4  | 对分检索               | (192) |
| 5—5  | 交换法八位排序(无符号)       | (194) |
| 5—6  | 交换法八位排序(带符号)       | (194) |
| 5—7  | 交换法十六位排序(无符号)      | (196) |
| 5—8  | 泡冒法八位排序(无符号)       | (198) |
| 5—9  | 泡冒法八位排序(带符号)       | (199) |
| 5—10 | Shell 交换法八位排序(无符号) | (200) |

## 第六章 A/D与D/A转换和数据采集

|     |                 |       |
|-----|-----------------|-------|
| 6—1 | A/D 转换程序        | (204) |
| 6—2 | 锯齿波产生程序         | (205) |
| 6—3 | 三角波产生程序         | (207) |
| 6—4 | 梯形波产生程序         | (209) |
| 6—5 | 方波、矩形波及脉冲波产生程序  | (209) |
| 6—6 | 正弦波产生程序         | (210) |
| 6—7 | 数据采集程序(流水线法)    | (211) |
| 6—8 | 数据采集程序(中断方式)(一) | (213) |
| 6—9 | 数据采集程序(中断方式)(二) | (216) |

## 第七章 数据块移动、合并与校验

|     |           |       |
|-----|-----------|-------|
| 7—1 | 数据块移动     | (218) |
| 7—2 | 数据块合并     | (218) |
| 7—3 | 数据块校验     | (221) |
| 7—4 | 数据校验和(8位) | (222) |
| 7—5 | 条件选择转移    | (224) |

## 第八章 数据处理程序

|     |            |       |
|-----|------------|-------|
| 8—1 | 求算术平均值     | (225) |
| 8—2 | 双字节二进制数开平方 | (228) |
| 8—3 | 五中取三求平均值   | (229) |
| 8—4 | 解线性方程组     | (231) |

|     |                                        |       |
|-----|----------------------------------------|-------|
| 8—5 | 计算 $\sum_{i=1}^M A_i \times B_i$ ..... | (237) |
|-----|----------------------------------------|-------|

## 第九章 延时程序

|     |                         |       |
|-----|-------------------------|-------|
| 9—1 | 1—255 MS 软件延时 .....     | (240) |
| 9—2 | 1—255 秒 软件延时 .....      | (240) |
| 9—3 | 1—255 秒 CTC 中断 延时 ..... | (242) |
| 9—4 | 1—255 分 CTC 中断延 时 ..... | (244) |
| 9—5 | 时、分、秒 实时钟程序 .....       | (246) |

## 第十章 其 它

|      |                                |       |
|------|--------------------------------|-------|
| 10—1 | RAM 存贮器读写检查 程序 .....           | (249) |
| 10—2 | 程序执行时间测定程序 .....               | (252) |
| 10—3 | 显示子程序 .....                    | (253) |
| 10—4 | 直接七段码转换及逆显示子程序 .....           | (255) |
| 10—5 | 七段显示计数程序 .....                 | (256) |
| 10—6 | 用CMCP—16微型打印机打印X轴的箭头部份程序 ..... | (259) |
| 10—7 | 用CMCP—16微型打印机打印Y轴的箭头部分程序 ..... | (260) |
| 10—8 | 实时打印输入数据的连续曲线 .....            | (262) |
| 附录:  | ASCII 表 .....                  | (265) |

# 第一部分 Z80汇编语言程序设计概论

微电脑的结构是以大规模集成电路为基础的，微电脑的用户在电子线路设计方面的工作量大大减少，而主要精力集中在接口技术和程序设计上。微电脑程序设计的基本工具是汇编语言，必须熟练地掌握。为了帮助Z80微电脑的用户更好地使用本手册所包含的众多的实用子程序，本书第一部分给大家介绍一些程序设计的基本知识和常用的程序设计方法。

## 第一章 Z80的指令系统

### 1—1 指令格式

微型电脑是按照一定的命令来执行特定的操作的，这种命令称为指令。把一系列的指令按一定的要求和规则组合起来就构成程序，一个程序能使微型电脑执行一个完整的任务。每种微型电脑都配有自身所能识别的指令，这种指令的集合称为指令系统。指令系统的好坏是微型电脑功能强弱的重要标志。

通常一条指令应包括两部分内容：

(一) 操作码 指明微型电脑做什么操作，如做加法，操作码为ADD，做传送数据，操作码为LD等等。

(二) 操作数 指明所要操作的数来自什么地方，操作的结果数应放在什么地方。如加法指令

ADD A, B

告诉我们加数放在B寄存器里，被加数放在累加器A里，相加的结果数应放在累加器里。

指令的格式较多，一般由1—4字节的二进制代码来表示的。

1. 单字节指令 如空操作指令 NOP它只有一字节的操作码 00H，没有操作数。又如传送指令

LD A, B

它虽然也只有一字节的代码78H，但既包含着操作码，又包含着操作数。

2. 双字节指令 如传送指令

LD (HL), n

它是由两个字节的代码来代表的，如图，其中36H是操作码，n是8位的操作数。

|    |
|----|
| 36 |
| n  |



### 3. 三字节指令 如传送指令

LD HL, 6600H

它是用三字节的代码表示的, 如图。其中第一个字节21为操作码, 其余两字节为操作数。

|    |
|----|
| 21 |
| 00 |
| 66 |

### 4. 四字节指令 如传送指令

LD IX, 45A2H

它是用四字节的代码表示的, 如图。其中DD21为操作码, A245为操作数。

|    |
|----|
| DD |
| 21 |
| A2 |
| 45 |

## 1——2 寻址方式

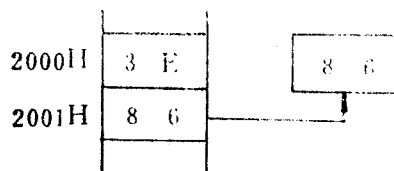
操作数是指令的重要组成部分, 由它来指定参加运算的数或数所在的单元地址。所谓寻址方式就是指指令查找操作数的方法。Z80 有十种寻址方式, 可以方便灵活地在整个内存区域内和CPU寄存器组内找到所需要的操作数。

### 1. 立即寻址 (Immediated Addressing)

这类指令的操作码后面直接放着实际的 8 位操作数。如指令

LD A, 86H

它的操作码为3EH, 后面跟着的86H就是立即数, 这条指令使立即数86H传送到累加器A, 如图。



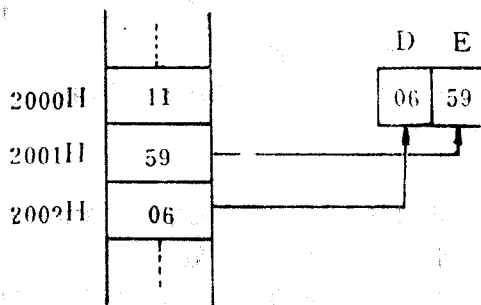
### 2. 立即扩展寻址 (Immediate Extended)

这类指令的操作码后面直接放着实际的 16 位操作数, 相当于对 8 位操作数的扩展。

如指令

LD DE, 0659H

它的操作码为11H, 16位操作数为 0659H, 这条指令把16位立即数放入DE 寄存器对。注意16位立即数存放在内存中时, 低8 位在上, 高8位在下。



### 3. 寄存器寻址 (Register Addressing)

这类指令, 在操作码中有若干位代码用来指定CPU中哪个寄存器的内容作为操作数。如指令

LD C, A

这是条一字节指令, 它的操作码为

|    |     |     |
|----|-----|-----|
| 01 | 001 | 111 |
|----|-----|-----|

即 4FH

其中最高两位01为操作码, 紧跟着的三位 001 是目标寄存器C的代码, 低三位111是源寄

存器A的代码。各CPU寄存器的代码如表所示。

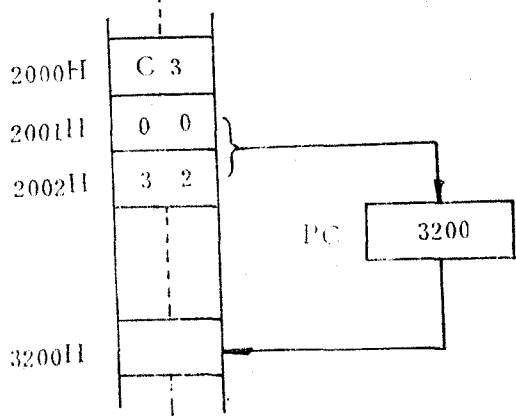
这条指令使寄存器A的内容放入寄存器C，前者通称为源寄存器，后者通称为目标寄存器。

4. 扩展寻址 (Extended Addressing)

这类指令操作码后面紧跟着两字节的16位地址，如跳转指令  
JP 3200H

这是一条三字节指令，C3H为操作码，后面紧跟着的16位地址3200H就是程序要转移的目标地址。因此这条指令能使程序转移到3200H单元处继续执行，如图所示。扩展寻址实际上是直接寻址，可以从整个内存区中找到任一单元。

| γ | 编 码 |
|---|-----|
| B | 000 |
| C | 001 |
| D | 010 |
| E | 001 |
| H | 100 |
| L | 101 |
| A | 111 |



又如传送指令

LD A, (4400H)

这条指令可以从内存中找到4400H单元，并把该单元的内容取出后存入寄存器A中（注意指令中括号的含义）。

这类寻址方式与立即扩展寻址不同，前者指令中所包含的16位地址用来寻找内存中的相应单元，以便进行读数或存数，也可以作为程序的转移地址，因此16位地址并不是实际的操作数。而后者指令中的16位数是作实际操作数用的。

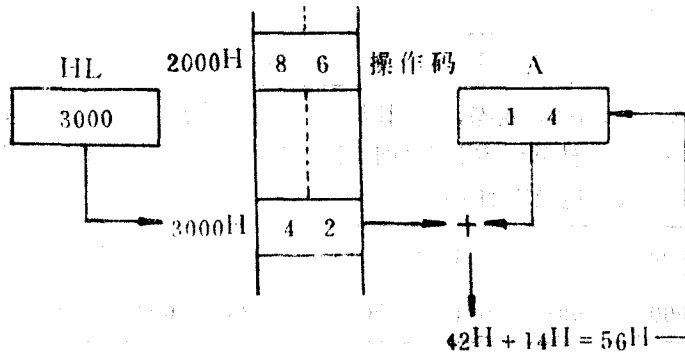
的，以便进行16位数的传送。

5. 寄存器间接寻址 (Register Indirect Addressing)

这类指令指定某寄存器对的内容作为实际操作数的地址。如指令

ADD A, (HL)

指令执行过程如图所示



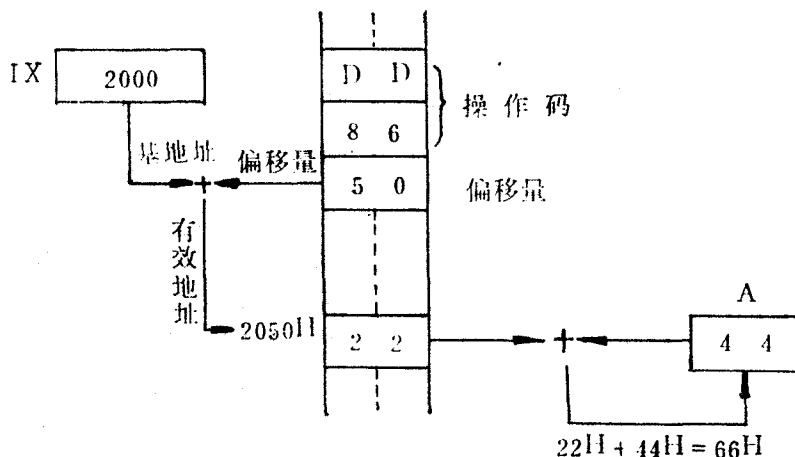
这是一条加法指令，操作码为86H，被加数14H放在A寄存器，加数42H放在内存单元中，该单元的地址3000H就是HL寄存器对的内容。指令执行的结果放A寄存器。

## 6. 变址寻址 (Indexed Addressing)

由这类指令指定的变址寄存器的内容 (称为基地址) 和指令中给定的偏移量相加, 得到寻找单元的有效地址。如加法指令

ADD A, (IX + 50H)

基地址2000H, 已放在变址寄存器IX中, 50H为偏移量, A中放44H。该指令执行的过程如图所示。



注意: 偏移量是一个带符号的8位数, 故用2的补码表示。

变址寻址主要用于处理数组和表格。它的优点是: 用同样的一组指令能处理数组和表格中所有的元素, 因此比较灵活方便。其缺点是: 执行速度慢, 因为CPU要做一次加法后才能获得有效地址。

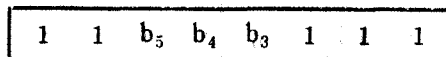
## 7. 零页寻址 (Modified Page Zero Addressing)

Z80有一条单字节的重新启动指令

RST n

用来指定零页中 (0000H—00FFH为零页) 某些规定的单元作为子程序的入口地址。

RST n的指令操作码为



其中三位二进制数 $b_5b_4b_3$ 有8种状态, 分别以0—7之间的整数表示, 故这条指令有8种不同的形式, 可分别把程序转移到零页上的相应的8个地址之一作为子程序的入口。

RST n指令的入口地址表如下:

| n    | 00H   | 08H   | 10H   | 18H   | 20H   | 28H   | 30H   | 38H   |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 入口地址 | 0000H | 0008H | 0010H | 0018H | 0020H | 0028H | 0030H | 0038H |

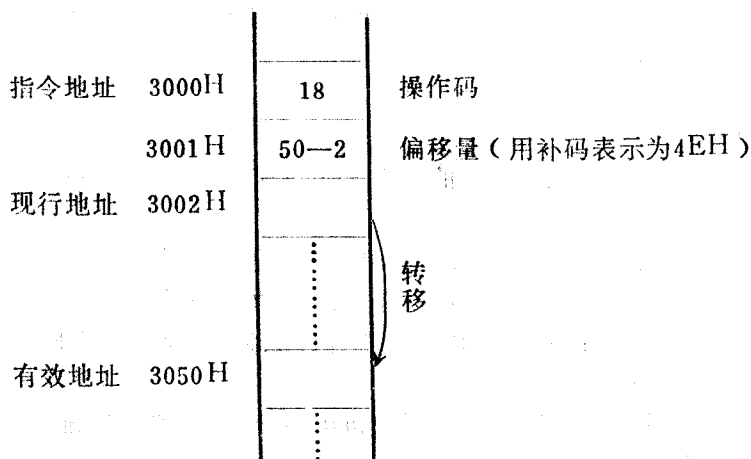
这种寻址方式比较简单, 只要用一个字节就可以确定下一条要执行的指令地址, 从而节省了内存单元。这种寻址方式主要用于形成中断矢量, 此时RST n指令不是由程序员编的, 而是由申请中断的外设提供的。

## 8. 相对寻址 (Relative Addressing)

相对寻址指令的第一字节为操作码，第二字节为偏移量，它是由带符号数的 2 的补码来表示的。如转移指令

JR 50H

产生需要寻找单元的有效地址的过程如图所示。



执行这条指令后，程序转向 3050H 继续执行。有效地址 3050H 产生的方法是

$PC \leftarrow \text{现行地址} + \text{偏移量}$

3050H    3002H    50H - 2

注意：① 指令中给出的偏移量（如 50H）放在内存中时必须减 2（即  $50H - 2$ ），这是因为转移指令取出后，现行地址往前跑了两个单元（这是两字节指令，PC 指向 3002H）。转移是以现行地址（此处为 3002H）为基准的。

② 偏移量的范围是 -128——+127，而跳转是以现行地址（指令地址加 2）为基准的，因此，偏移量的范围应改为 -126——+129

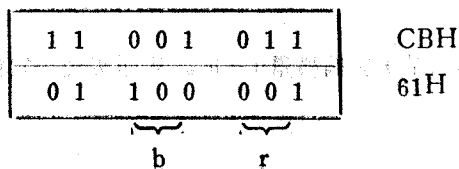
对大多数程序来说，有关的程序段都在附近，可以采用上述相对寻址的方法来转移程序，因为这种指令的偏移量只有一个字节，可以节省指令的存储空间。

## 9. 位寻址 (Bit Addressing)

Z80 中有位操作指令，可以对某一个字节（可以是寄存器的内容，也可以是内存单元的内容）的任一位进行操作（置 1、置 0 或测试）。字节的地址可用寄存器寻址或变址寻址等方法找到，至于字节中的某一位可由操作码中的三位代码来寻址。如指令

BIT 4, C

用来对 C 寄存器中的  $D_4$  位进行测试，即是“1”还是“0”，测试的结果由标志位 Z 来指示，其指令码的形式为



r表示要操作的寄存器，由最低三位代码表示，b表示对寄存器中的哪一位进行操作，由中间三位表示（8种状态中的每一种对应于被测试字节中的某一位）。

### 10. 隐含寻址 (Implied Addressing)

在指令的操作码中，有一个操作数的地址没有指定。但是这类指令中，隐含了以某一寄存器（通常是累加器A）中的内容作为一个操作数。Z80中的算术或逻辑指令，就是这种寻址方式的指令。如指令 `OR 7` 是把立即数7与累加器A中的内容进行逻辑或，其中累加器A在指令中是隐含的。

## 1—3 指令系统

Z80共有158种指令，其中包括8080A的全部78种。下面把指令系统分成8类简介如下：

### 一、8位数传送指令

8位字长的微电脑在加工、处理数据时，数据在CPU内各寄存器之间、寄存器与内存单元之间的传送是以8位数的形式进行的。这类指令具有下列特点：

1. 指令必须指出数据源地址和目标地址，而且目标地址在前面，源地址在后面，如 `LD A, (2300H)`

其中A为目标地址，2300H为源地址，意义是把2300H单元的8位数传送到累加器A。

2. 传送后源地址所存数据不变。

3. 这类指令中的绝大部分在执行后不影响标志位。

按寻址方式分，这类指令又可分六种：

1. 寄存器之间传送 如 `LD r1, r2`

其中r<sub>1</sub>和r<sub>2</sub>为CPU内8位寄存器的任一个，

如 `LD A, B`

这种指令的功能是把B寄存器的内容传送到累加器A。

2. 立即数送寄存器 如 `LD r, n`

这种指令的功能是把立即数n传送到由r所指定的CPU内部寄存器，如

`LD A, 7FH`

3. 寄存器间接寻址传送

如 `LD r, (HL)`

`LD (HL), r`

`LD (HL), n`

这种指令是以HL寄存器对的内容为地址，可把任一8位寄存器的内容或立即数传送到指定的内存单元，也可以把指定内存单元的内容传送到任一寄存器。

4. 变址寄存器寻址的传送

如 `LD r, (IX+d)`

`LD (IX+d), r`

`LD (IX+d), n`

内存单元用变址寄存器（IX或IY）加偏移量d寻址。从而实现指定内存单元的内容传送到任一8位寄存器的功能。

### 5. 寄存器对BC或DE间址传送

如 LD A, (BC)  
LD (BC), A

这种指令用于累加器和内存单元之间传送数据。

### 6. 扩展寻址传送

如 LD A, (nn)  
LD (nn), A

这种指令用16位立即数来指定内存单元的地址, 以实现累加器和内存单元之间传送数据。

表1—1列出了所有8位数传送指令的机器码, 机器码用十六进制表示。根据此表可以方便地查到与指令符相对应的机器码, 其方法如下: 表中最上面一行为源地址, 最左面一列为目标地址。由指定的源地址垂直向下, 从指定的目标地址向右, 两者相交处方格内的数据就是所要找的机器码。如

LD (HL), C 指令的机器码为71H

LD B, (IX+08H) 指令的机器码为DD4608H

表中未列出机器码的空间, 表示没有这种指令。

## 二、16位传送指令

### 1. 立即数传送到寄存器对

如 LD dd, nn

这种指令的功能是把16位立即数(nn)传送到某个指定的寄存器对 dd(即BC、DE、HL、SP、IX、IY中的任一个), 常用于置地址初值。

### 2. 16位扩展寻址的数的传送

如 LD dd, (nn)  
LD (nn), dd

这种指令的功能是以 nn 和 nn+1 作为地址, 把这两个连续单元的内容传送到指定的寄存器对(如BC、DE、HL、SP、IX、IY中的任一个), 或作相反传送。

### 3. 16位数传送到堆栈指示器SP

如 LD SP, HL  
LD SP, IX

这种指令用来在内存的合适区域设置栈顶地址。

### 4. 堆栈操作指令

如 PUSH qq  
PUSH IX  
POP qq  
POP IX

其中PUSH是把信息推入堆栈的指令, 用于保护现场。qq 表示AF、BC、DE寄存器对中的任一个; POP指令把保存在堆栈中的信息弹出来, 用于恢复现场。

表1—2列出了16位数传送指令的机器码。

表 1—1 8 位数传送指令LD  
源

|         |           | 隐 含  |          | 寄 存 器         |               |               |               |               |               |               | 寄存器间接    |          |          | 变 址        |               | 扩充地址          | 立即                 |         |
|---------|-----------|------|----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|----------|----------|----------|------------|---------------|---------------|--------------------|---------|
|         |           | I    | R        | A             | B             | C             | D             | E             | H             | L             | (H<br>L) | (B<br>C) | (D<br>E) | (IX<br>+d) | (IY<br>+d)    | (nn)          | n                  |         |
| 目 标     | 寄 存 器     | A    | ED<br>57 | ED<br>5F      | 7F            | 78            | 79            | 7A            | 7B            | 7C            | 7D       | 7E       | 0A       | 1A         | DD<br>7E<br>d | FD<br>7E<br>d | 3A<br>n<br>n       | 3E<br>n |
|         |           | B    |          |               | 47            | 40            | 41            | 42            | 43            | 44            | 45       | 46       |          |            | DD<br>46<br>d | FD<br>46<br>d |                    | 06<br>n |
|         |           | C    |          |               | 4F            | 48            | 49            | 4A            | 4B            | 4C            | 4D       | 4E       |          |            | DD<br>4E<br>d | FD<br>4E<br>d |                    | 0E<br>n |
|         |           | D    |          |               | 57            | 50            | 51            | 52            | 53            | 54            | 55       | 56       |          |            | DD<br>56<br>d | FD<br>56<br>d |                    | 16<br>n |
|         |           | E    |          |               | 5F            | 58            | 59            | 5A            | 5B            | 5C            | 5D       | 5E       |          |            | DD<br>5E<br>d | FD<br>5E<br>d |                    | 1E<br>n |
|         |           | H    |          |               | 67            | 60            | 61            | 62            | 63            | 64            | 65       | 66       |          |            | DD<br>66<br>d | FD<br>66<br>d |                    | 26<br>n |
|         |           | L    |          |               | 6F            | 68            | 69            | 6A            | 6B            | 6C            | 6D       | 6E       |          |            | DD<br>6E<br>d | FD<br>6E<br>d |                    | 2E<br>n |
|         | 寄 存 器 间 接 | (HL) |          |               | 77            | 70            | 71            | 72            | 73            | 74            | 75       |          |          |            |               |               |                    | 36<br>n |
|         |           | (BC) |          |               | 02            |               |               |               |               |               |          |          |          |            |               |               |                    |         |
|         |           | (DE) |          |               | 12            |               |               |               |               |               |          |          |          |            |               |               |                    |         |
| 变 址     | (IX + d)  |      |          | DD<br>77<br>d | DD<br>70<br>d | DD<br>71<br>d | DD<br>72<br>d | DD<br>73<br>d | DD<br>74<br>d | DD<br>75<br>d |          |          |          |            |               |               | DD<br>36<br>d<br>n |         |
| 址       | (IY + d)  |      |          | FD<br>77<br>d | FD<br>70<br>d | FD<br>71<br>d | FD<br>72<br>d | FD<br>73<br>d | FD<br>74<br>d | FD<br>75<br>d |          |          |          |            |               |               | FD<br>36<br>d<br>n |         |
| 扩 充 地 址 | (nn)      |      |          | 32<br>n<br>n  |               |               |               |               |               |               |          |          |          |            |               |               |                    |         |
| 隐 含     | I         |      |          | ED<br>47      |               |               |               |               |               |               |          |          |          |            |               |               |                    |         |
|         | R         |      |          | ED<br>4F      |               |               |               |               |               |               |          |          |          |            |               |               |                    |         |

表 1-2 16位数传送指令 ‘LD’, ‘PUSH’和 ‘POP’  
源

|                       |           | 寄 存 器 |    |                    |                    |              |                    |                    | 立即扩充               | 扩充地址     | 寄存器<br>接间 |          |                    |                    |          |
|-----------------------|-----------|-------|----|--------------------|--------------------|--------------|--------------------|--------------------|--------------------|----------|-----------|----------|--------------------|--------------------|----------|
|                       |           | A     | F  | B                  | C                  | D            | E                  | H                  | L                  | SP       | IX        | IY       | nn                 | (nn)               | (SP)     |
| 目<br>标<br>器           | 寄 存 器     | AF    |    |                    |                    |              |                    |                    |                    |          |           |          |                    |                    | F1       |
|                       |           | BC    |    |                    |                    |              |                    |                    |                    |          |           |          | 01<br>n<br>n       | ED<br>4B<br>n<br>n | C1       |
|                       |           | DE    |    |                    |                    |              |                    |                    |                    |          |           |          | 11<br>n<br>n       | ED<br>5B<br>n<br>n | D1       |
|                       |           | HL    |    |                    |                    |              |                    |                    |                    |          |           |          | 21<br>n<br>n       | 2A<br>n<br>n       | E1       |
|                       |           | SP    |    |                    |                    |              |                    |                    | F9                 |          | DD<br>F9  | FD<br>F9 | 31<br>n<br>n       | ED<br>7B<br>n<br>n |          |
|                       |           | IX    |    |                    |                    |              |                    |                    |                    |          |           |          | DD<br>21<br>n<br>n | DD<br>2A<br>n<br>n | DD<br>E1 |
|                       |           | IY    |    |                    |                    |              |                    |                    |                    |          |           |          | FD<br>21<br>n<br>n | FD<br>2A<br>n<br>n | FD<br>E1 |
| 压<br>入<br>指<br>令<br>→ | 扩充地址      | (nn)  |    | ED<br>43<br>n<br>n | ED<br>53<br>n<br>n | 22<br>n<br>n | ED<br>73<br>n<br>n | DD<br>22<br>n<br>n | FD<br>22<br>n<br>n |          |           |          |                    |                    |          |
|                       | 寄存器<br>间接 | (SP)  | F5 | C5                 | D5                 | E5           |                    |                    | DD<br>E5           | FD<br>E5 |           |          |                    |                    |          |

↑  
弹出指令

### 三、16位数交换指令

这类指令有下列几种:

EX DE, HL , DE与HL的内容交换

EX AF, AF' , AF与辅助寄存器组的AF'交换

EXX , BC与BC', DE与DE', HL与HL'之间同时交换

EX (SP), HL 堆栈内容与HL内容互换

EX (SP), IX 堆栈内容与IX内容互换



表1—3列出了交换指令的机器码。

表 1—3 交换指令“EX”和“EXX”

四、数据块传送指令  
这类指令有四条：LDI、LDIR、LDD和LDDR。具有下列特点：

1. 这类指令用于内存区中成批数据的搬家即可以从内存区的一处搬到另一处。
2. 这类指令执行后，原内容不变。因此这种传送具有复制的含义。
3. 要用到三个寄存器对，并由程序员事先予置好一定的数据。

|           |                | 隐 含 寻 址 |             |    |          |          |
|-----------|----------------|---------|-------------|----|----------|----------|
|           |                | AF      | BC' DE' HL' | HL | IX       | IY       |
| 隐<br>含    | AF             | 08      | D9          |    |          |          |
|           | BC<br>DE<br>HL |         |             |    |          |          |
|           | DE             |         |             | EB |          |          |
| 寄存器<br>间接 | (SP)           |         |             | E3 | DD<br>E3 | FD<br>E3 |

- HL 放传送字节的源地址
- DE 放传送字节的目标地址
- BC 作为字节计数器，放数据块的字节数

上述四条指令又可分成两类：

1. 单个传送

- LDI , (DE)←(HL), DE←DE + 1, HL←HL + 1  
BC←BC - 1 从首地址单元开始传送。
- LDD , (DE)←(HL), DE←DE - 1, HL←HL - 1  
BC←BC - 1 从末地址单元开始传送。

其功能是把HL所指的内存单元内容传送到由DE所指的内存单元中去。并且该指令每执行一次，只能传送一个数据，同时伴有自动修改地址（地址加1或减1）和字节计数器减1。

2. 成组传送

- LDIR , 从数据块的首地址单元开始传送。
- LDDR , 从数据块的末地址单元开始传送。

其功能与单个传送类似，所不同的是成组传送指令只要一条就能完成整个数据块的传送。但它并不能完全代替单个传送指令，若在数据传送过程中需要插入一些其它操作时，就只能用单个传送指令。

表1—4列出了数据块传送指令的机器码。

五、数据块查找指令

这类指令有四条：CPI、CPIR、CPD和CPDR。具有下列特点：

1. 查找指令的功能是在64K内存范围内迅速找出某一个字节数或ASCII字符。
2. 查找的方法为：把需要查找的内容预先放在累加器A中，然后以A的内容与由HL所指示的每个内存单元的内容逐个进行比较，即A - (HL)，当找到时，A - (HL) = 0，