* 第 3 章　汇编语言与程序设计

计算机强大的功能体现在程序设计上，汇编语言是最基本的编程语言。本章首先介绍 80x86 微处理器寻址方式、指令系统、汇编语言语句格式、保护方式编程指令、与保护方式编程接口、与 C/C++ 编程接口，然后介绍汇编语言程序设计、宏以及 DOS 功能调用，最后简要说明上机过程。

3.1　概　　述

1. 指令与程序

计算机的所有工作就是执行程序，而程序是为求解某一问题或进行某项工作的若干条指令的有序集合。所谓指令，就是指示计算机进行某一工作的命令，在计算机中用二进制数表示，其格式如图 3.1 所示，至少包括两个部分，即操作码域和操作数域。

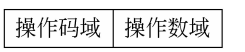

图 3.1　指令的基本格式

其中，操作码域用来说明指令的类型与功能，如加法、减法、存储器读、存储器写或数据输入/输出等。操作数域也称为地址域，用来说明参加操作（或运算）的数据在存储器中的地址及操作结果存放的地址。

通常，表示一条指令的二进制数称为指令字，按其长度可分为单字节、双字节、三字节以至四字节指令等；按操作数的个数又可分为无操作数、单操作数、双操作数以至三操作数指令，也称为无地址、一地址、二地址以至三地址指令。其中，存放结果的称为目的操作数或目的地址，其他称为源操作数或源地址。一台计算机全部指令的集合称为该计算机的指令系统，不同的计算机有不同的指令系统。

通常，人们将用二进制数表示指令的语言称为机器语言，其程序称为机器语言程序。例如，以下两条指令可构成一段程序，其作用是将数据 20H 经寄存器 CL，与累加器 AL 中的数相加，结果在 AL 中。

```
10110001 00100000
00000010 11000001
```

2. 汇编语言

汇编语言是用英文单词或其缩写字母表示指令的语言，如用 ADD、SUB、JMP 表示加、减、转移等操作，这些符号称为助记符。这种用助记符表示的指令称为汇编语言指令，用汇编语言指令编写的程序称为汇编语言程序。通常，一条汇编语言指令对应于一条机器语言指令。例如，上一节的机器语言程序可表示如下：

```
MOV  CL, 20H
ADD  AL, CL
```

但是，这种程序必须转换成机器语言才能被计算机识别和执行，其过程称为汇编，由专

门的程序进行，这种程序称为汇编程序。其中汇编语言程序称为源程序，经汇编而生成的机器语言程序称为目标程序或目标代码。

在汇编语言中，除了对应于机器语言的指令系统外，还有一些用来对汇编过程进行辅助说明的指令，称为伪指令。因此，汇编语言是由汇编语言指令系统、伪指令以及相应的语法规则组成的。其优点是可阅读性强，易于编程、理解和记忆。而且执行速度快，常用来设计一些对实时性要求比较高的程序。

3.2 80x86 寻址方式

在指令中，表示操作数或操作数的地址有多种，统称为寻址方式。同一条指令可有多种寻址方式，以说明不同的操作对象。

3.2.1 数据类型

在计算机中，数据有多种类型。一种用来表示指令，称为指令字；一种是指令处理的对象，称为操作数；第三种是存储器单元的序号，称为地址。其中操作数又分为两种，一种是参加运算或由计算机处理的对象，称为数据操作数，简称为数据；另一种用来表示转移指令的目标地址，称为转移地址操作数，简称为转移地址。

1. 数据操作数

数据操作数常见的有 3 种。一种是在指令中直接给出，称为立即数；另一种存放在寄存器中，称为寄存器数据；第三种存放在存储器中，称为存储器数据。

2. 转移地址操作数

转移地址操作数用来表示转移指令的目标地址。其表示方式可以在转移指令中直接给出，也可以由某种寻址方式，经计算后求得。

3.2.2 寻址方式

根据操作数的两种类型，80x86 微处理器的寻址方式也可分为两大类型，即数据寻址方式和转移地址寻址方式。

1. 数据寻址方式

数据寻址方式有以下 7 种。

1）立即数

立即数是指令操作数域直接给出参加运算或者进行某种操作的数据，又称即时数。它跟在操作码之后作为指令的一部分直接存放在代码段中，因此称为立即数，有 8 位和 16 位等形式。对于 16 位数据，低字节在先，高字节在后。例如指令：

```
MOV  AL, 5
```

执行后(AL)＝05H，其示意如图 3.2(a)所示。

又如指令：

```
MOV  AX, 2790H
```

执行后(AX)＝2790H，操作示意图如图 3.2(b)所示。

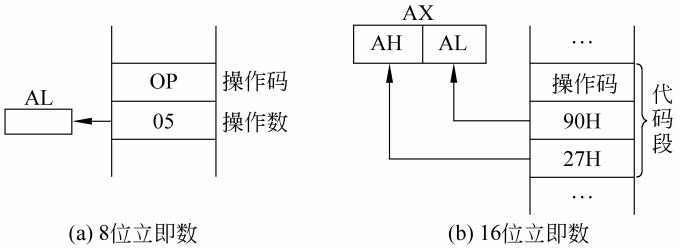

图 3.2　立即寻址示意图

对于 32 位微处理器，传送指令如下：

```
MOV  EAX, 12345678H              ;执行后 (EAX)=12345678H
```

立即数只能是源操作数，不能是目的操作数。这种寻址方式执行速度快，主要用来对寄存器或存储器赋值。

2）寄存器寻址

寄存器寻址的指令操作数域给出的是存放操作数的寄存器。对于 8 位操作数，寄存器可以是 AL、AH、BL、BH、CL、CH、DL、DH；对于 16 位操作数，寄存器可以是 AX、BX、CX、DX、SI、DI、SP、BP；对于 32 位微处理器，可使用 32 位寄存器 EAX、EBX、ECX、EDX、ESI、EDI、ESP、EBP。例如指令：

```
MOV  BX, AX                      ;操作过程如图 3.3 所示
```

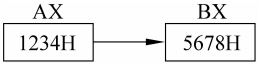

图 3.3　寄存器寻址示例

又如指令：

```
MOV  ECX, EDX                    ;将寄存器 EDX 中的数据传送到 ECX 中
```

3）直接寻址

直接寻址的指令操作数域给出的是存放操作数地址的偏移量，也称为有效地址，常用 EA 表示。如果操作数存放在数据段以外的其他段中，应在指令中指定段寄存器。直接寻址如图 3.4 所示。例如指令：

```
MOV  BX, [2000H]
```

设执行前(DS)＝4000H，EA＝2000H，(42000H)＝1FH，(42001H)＝75H，则执行后(BX)＝751FH，如图 3.5 所示。

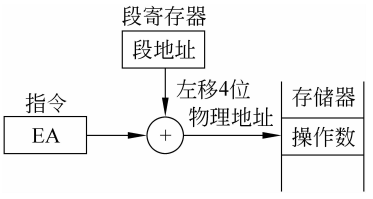

图 3.4　直接寻址示意图

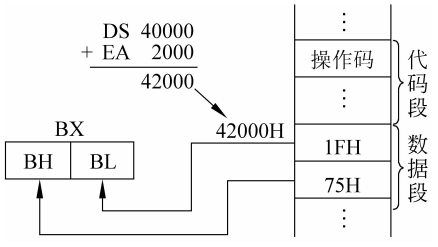

图 3.5　直接寻址示例

在汇编语言中，也可以使用符号表示地址，称为符号地址。例如指令：

```
MOV  AX, [VALUE]
```

其中，VALUE 表示存放操作数单元的符号地址，书写时可省略方括号，例如：

```
MOV  AX, DATA
```

4）寄存器间接寻址

寄存器间接寻址的指令操作数域给出存放操作数地址偏移量的寄存器。其寄存器主要有 BX、BP、SI 和 DI。如果没有特别说明，使用寄存器 BX、SI、DI 寻址时，默认数据段寄存器 DS；使用 BP 寻址时，默认堆栈段寄存器 SS。物理地址的生成如图 3.6 所示。例如指令：

```
MOV  AX, [BX]
```

设执行前(DS)＝3000H，(BX)＝3000H，则执行后(AX)＝2C7DH，其示意如图 3.7 所示。

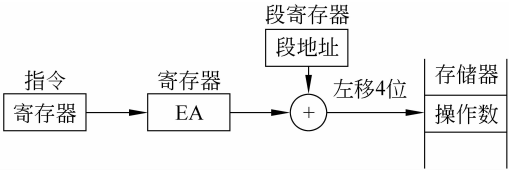

图 3.6　寄存器间接寻址示意图

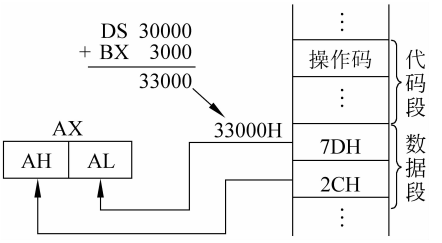

图 3.7　寄存器间接寻址示例

又如指令：

```
MOV  AX, ES:[SI]              ;段寄存器是 ES,表示数据在附加段中
```

5）寄存器相对寻址

寄存器相对寻址的指令操作数域给出的是存放基地址的基址寄存器和一个偏移量。基地址与偏移量相加，作为操作数地址的偏移地址，也就是有效地址 EA。段地址左移 4 位，与有效地址相加，生成 20 位的物理地址，如图 3.8 所示。例如指令：

```
MOV  AX, DATA[DI]
```

设执行前(DS)＝3000H，(DI)＝2000H，DATA＝300H，则执行后(AX)＝201AH，其示意如图 3.9 所示。

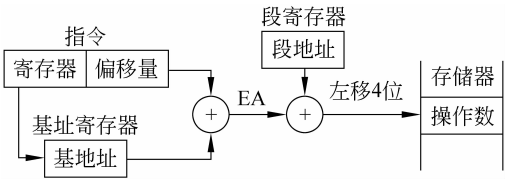

图 3.8　寄存器相对寻址示意图

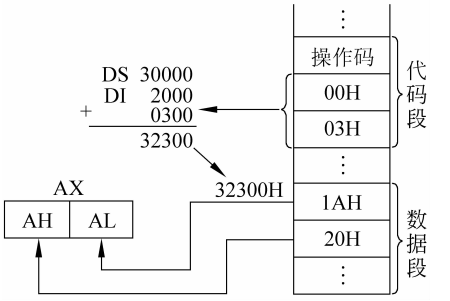

图 3.9　寄存器相对寻址示例

6）基址加变址寻址

基址加变址寻址的指令操作数域给出的是一个存放基地址的基址寄存器和一个存放变址基值的变址寄存器，两寄存器中的数据相加，作为操作数的偏移地址，即有效地址 EA。基址寄存器主要有 BX、BP，变址寄存器主要有 SI 和 DI。其中，基址寄存器 BX 默认段寄存器 DS，基址寄存器 BP 默认段寄存器 SS，允许使用段跨越，其示意如图 3.10 所示。

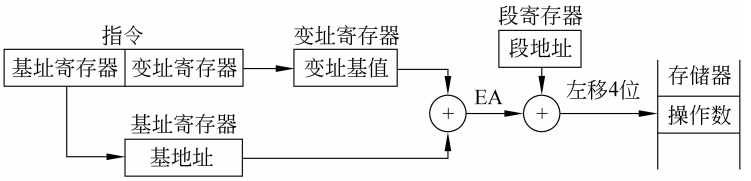

图 3.10　基址加变址寻址示意图

例如指令：

```
MOV  AX, [BX][DI]
```

或

```
MOV  AX, [BX+DI]
```

设执行前（DS）＝2100H，（BX）＝0158H，（DI）＝10A5H，则执行后（AX）＝1234H，如图 3.11 所示。

7）相对基址加变址寻址

相对基址加变址寻址的指令操作数域给出的是一个存放基地址的基址寄存器、一个存放变址基值的变址寄存器和一个偏移量，两寄存器中的数据及偏移量相加，作为操作数的偏移地址，也就是有效地址 EA。基址寄存器主要有 BX、BP，变址寄存器主要有 SI 和 DI。其中，基址寄存器 BX 默认段寄存器 DS，基址寄存器 BP 默认段寄存器 SS。例如指令：

```
MOV  AX, MASK[BX][DI]
MOV  AX, MASK[BX+DI]
MOV  AX, [MASK+BX+DI]
```

设执行前(DS)＝3000H，(BX)＝2000H，(DI)＝1000H，MASK＝0250H，则执行后(AX)＝4675H，其示意如图 3.12 所示。

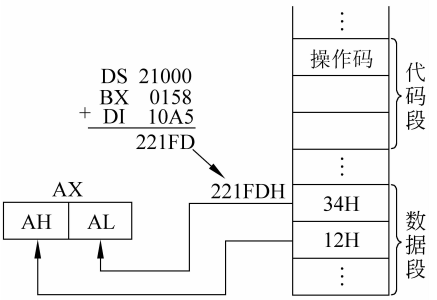

图 3.11　基址加变址寻址示例

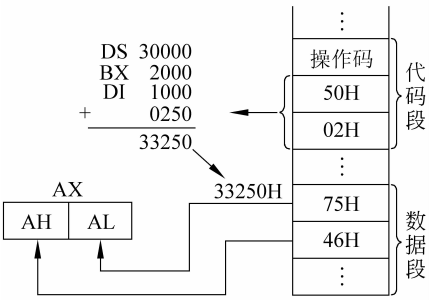

图 3.12　相对基址变址寻址示例

这种寻址方式为数组、表格及堆栈的访问，提供了方便。

2. 转移地址寻址方式

这种寻址方式主要用来确定转移指令和子程序调用指令 CALL 的目标地址，其目标地址可能在当前代码段内，也可能在当前代码段外，故有 4 种方式。

1）段内相对寻址

段内相对寻址是转移地址由指令指针 IP 的当前值与指令中给出的 8 位或 16 位偏移量相加而生成。偏移量是一个带符号的数，用补码表示。其中 8 位偏移量称为段内短转移(SHORT)，转移范围为 −128～＋127；16 位偏移量称为段内近转移(NEAR)，转移范围为 − 32 768～＋32 767。其示意如图 3.13 所示。

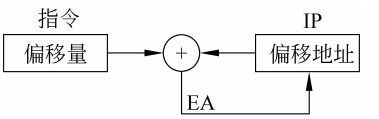

图 3.13　段内相对寻址示意图

段内相对寻址的特点是代码段寄存器 CS 的值保持不变，偏移量可以是 8 位或者 16 位。指令格式如下：

```
JMP  SHORT  OPR                    ;IP←(IP)+8 位偏移量
JMP  NEAR   PTR  OPR               ;IP←(IP)+16 位偏移量
```

2）段内间接寻址

段内间接寻址是转移指令的偏移地址存放在寄存器或存储器单元中。寄存器或存储器单元可由数据寻址方式中除立即数之外的任何一种方式得到，如图 3.14 所示。

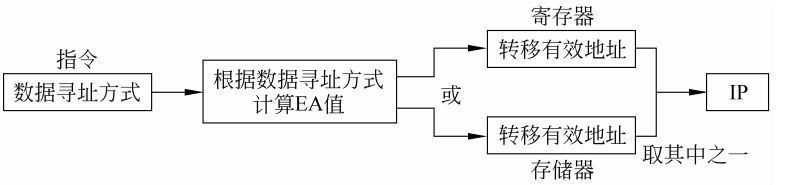

图 3.14　段内间接寻址示意图

例如指令：

```
JMP  BX                            ;将寄存器 BX 中的数送入指令指针 IP
```

又如指令：

```
JMP  WORD  PTR [BX][SI]            ;按照基址加变址的方式获取操作数，送入 IP
```

3）段间直接寻址

段间直接寻址是在指令中直接给出转移的 16 位段地址和 16 位偏移地址，分别送入代码段寄存器 CS 和指令指针寄存器 IP 中。

例如指令：

```
JMP  FAR  PTR  NEXTSUB
```

段间寻址由 FAR 来说明，其中 NEXTSUB 是符号地址，在指令中直接给出，包括转移目标的偏移地址和段地址。

4）段间间接寻址

在段间间接寻址方式中，转移地址存放在存储器中的两个连续字单元中，其中第一个字是偏移地址，第二个字是段地址。存储器单元地址可用数据寻址方式中除立即数和寄存器直接寻址以外的任何一种寻址方式来确定，如图 3.15 所示。

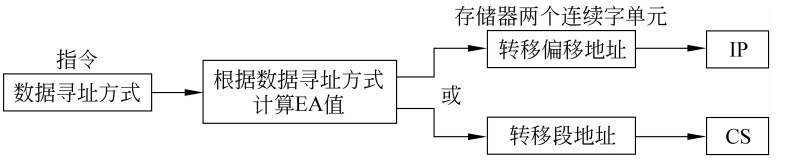

图 3.15　段间间接寻址示意图

例如指令：

```
JMP   DWORD   PTR [BX]
                        ;由寄存器 BX 确定连续两个存储器单元,取出其中的数据分别送入 IP 和 CS
```

又如指令：

```
JMP   DWORD   PTR [BP+SI+100H]
                    ;由[BP+SI+100H]确定连续两个存储器单元,取出其中的数据分别送入 IP 和 CS
```

3.3　80x86 指令系统

80x86 微处理器的指令系统按功能可分为 6 种类型，即数据传送指令、算术运算指令、逻辑运算指令、串操作指令、程序控制指令和处理器控制指令。

3.3.1　数据传送类指令

数据传送类指令用于寄存器、存储器单元或输入输出端口之间传送数据或地址，共 14 条，可分为 4 种类型：通用数据传送指令、地址传送指令、状态标志寄存器传送指令和输入/输出指令。

1. 通用数据传送指令

1）传送指令 MOV

格式：

```
MOV   DST, SRC
```

功能：DST←(SRC)

说明：DST 表示目的操作数地址，SRC 表示源操作数地址，其作用是将源操作数地址中的数据传送到目的操作数地址单元中。其中目的操作数不能是立即数、CS 和 IP 寄存器，两存储器单元之间和两段寄存器之间不能直接传送，立即数不能直接传送到段寄存器，允许段跨越，不影响标志位。源/目的操作数传送关系如图 3.16 所示。

例如指令：

```
MOV   AX, DX                        ;源/目的操作数均为 16 位
```

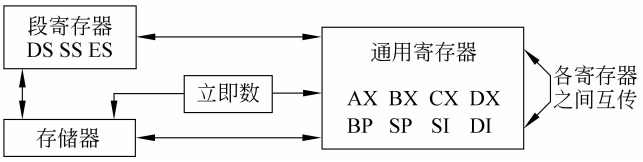

图 3.16　源操作数与目的操作数规定

```
MOV  AL, ES:[BX]                    ;目的操作数为 8 位,说明源操作数也为 8 位
```

对于 32 位微处理器,可实现 32 为数据传送,例如指令:

```
MOV  EAX, [EBX+ECX * 4]             ;目的操作数为 32 位
```

【例 3.1】 将地址为 ADDR1 存储器单元中的数据传送到同段内地址为 ADDR2 的存储器单元中。

解:使用传送指令,程序如下。

```
MOV  AL, ADDR1
MOV  ADDR2, AL
```

【例 3.2】 将立即数 3B30H 传送到段寄存器 ES 中。

解:对 ES 不能直接传送立即数,故需借用其他寄存器来实现,程序如下。

```
MOV  AX, 3B30H
MOV  ES, AX
```

【例 3.3】 将立即数 3AB0H 传送到 ADDR 单元中。

解:按字立即数传送,程序如下。

```
MOV  WORD PTR ADDR, 3AB0H           ;WORD PTR 表示按字传送
```

2) 堆栈操作指令

(1) 压栈指令 PUSH

格式:

```
PUSH  SRC
```

16 位功能:SP←(SP)－2
　　(SP)＋1,(SP)←(SRC)

32 位功能:ESP←(ESP)－4
　　((ESP)＋3,(ESP)＋2,(ESP)＋1,(ESP))←(SRC)

(2) 弹栈指令 POP

格式:

```
POP  DST
```

16 位功能:DST←((SP)＋1,(SP))
　　SP←(SP)＋2

32 位功能:DST←((ESP)＋3,(ESP)＋2,(ESP)＋1,(ESP))
　　(ESP)←(ESP)＋4

说明：堆栈操作指令以字为单位进行，可以使用除立即数以外的其他寻址方式，允许段跨越；PUSH CS 合法，但 POP CS 非法；不影响状态标志位。

【例 3.4】 将 CS 中的数据传送到段寄存器 DS 中去。

解：使用堆栈操作指令，程序如下。

```
PUSH  CS
POP   DS
```

3）交换指令 XCHG

格式：

```
XCHG  OPR1, OPR2
```

功能：(OPR1)↔(OPR2)

说明：XCHG 指令不适用段寄存器和立即数，不允许两存储器单元直接交换，两操作数可以是 8 位、16 位；允许段跨越，不影响状态标志位。例如指令：

```
XCHG  AL, BL                    ;两操作数均为 8 位
XCHG  AX, SS:[SI]               ;AX 寄存器指明操作数为 16 位
```

对于 32 位微处理器，指令如下：

```
XCHG  EAX, EBX                  ;两操作数均为 32 位
```

4）换码指令 XLAT

格式：

```
XLAT  OPR
```

或

```
XLAT
```

功能：AL←((BX)+(AL))

说明：OPR 表示表格中的地址，BX 存放表格首地址，AL 存放待查值(偏移量)，查得结果送 AL 中，不影响状态标志位。

【例 3.5】 在以 DATA 为首地址的主存区域中连续存放着 0～9 的平方表，试编写查表程序。

解：首地址存入寄存器 BX 中，待查数据 X 送入 AL 中。

```
MOV  BX, OFFSET DATA
MOV  AL, X
XLAT DATA  或  XLAT
```

2. 地址传送指令

1）有效地址送寄存器指令 LEA

格式：

```
LEA  REG,SRC
```

功能：REG←SRC

说明：将源操作数的有效地址传送到指定的寄存器 REG 中；目的寄存器为 16 位通用寄存器，源操作数可以是除立即数和寄存器寻址以外的其他寻址方式；允许段跨越；不影响状态标志位。

【例 3.6】 将例 3.5 中 DATA 首地址存入寄存器 BX 中。

解：使用 LEA 指令，程序如下。

```
LEA  BX, DATA
```

2）向指定寄存器与段寄存器 DS 传送地址指令 LDS

格式：

```
LDS  REG, SRC
```

功能：REG←(SRC)

DS←(SRC+2)

说明：将存储器中连续两个字单元中的数据传送到指定寄存器和段寄存器 DS 中，不影响状态标志。

例如指令：

```
LDS  SI, [1508H]
          ;将 DS 和[1508H]所指定单元中的两个字操作数依次送入寄存器 SI 和段寄存器 DS 中
```

3）向指定寄存器与段寄存器 ES 传送地址指令 LES

格式：

```
LES  REG, SRC
```

功能：REG←(SRC)

ES←(SRC+2)

说明：与指令 LDS 功能相同，段寄存器是 ES。

例如指令：

```
LES  DI, [DI]
```

3. 状态标志寄存器传送指令

1）状态标志寄存器低 8 位送 AH 指令 LAHF

格式：

```
LAHF
```

功能：AH←(PSW 的低字节)

说明：不影响状态标志位。

2）寄存器 AH 送状态标志寄存器低 8 位指令 SAHF

格式：

```
SAHF
```