高等学校规划教材

汇编语言程序设计

Huibian Yuyan Chengxu Sheji

主编 张虹

中国矿业大学出版社

HUIBIAN YUYAN CHENGXU SHEJI

前　言

《汇编语言程序设计》是煤炭工业部"九五"规划教材。1997 年 1 月，在徐州召开的煤炭工业部普通高校"九五"规划教材会议上，确定由中国矿业大学等七院校编写。

汇编语言程序设计是计算机硬件、软件及计算机应用专业必修的基础核心课程。通过本课程的学习，使学生掌握程序设计的基本技能，提高编程及用汇编语言解决实际问题的能力，进一步了解 80386、80486、Pentium CPU 的工作模式及扩充功能，以便跟踪微机新技术的发展和应用水平。

本书共分为十六章，在重点介绍 8086/8088 汇编语言寻址方式、指令系统以及各种程序设计方法的基础上，进一步介绍程序应用技巧、高级宏汇编技术、模块化程序设计方法和常用高级语言或数据库语言与汇编语言的连接，最后是宏汇编语言上机操作、实验和附录，为读者用汇编语言进行程序设计提供了方便。

本书的编写特点是：① 由浅入深、系统完整、条理清楚、重点突出，加强基础知识的讲析，重视实践环节，以达到培养学生分析、综合和解决实际问题等能力的目的；② 本书在编写中为了跟踪微机新技术的发展，介绍了 Pentium 和 Pentium Ⅱ 的 CPU 体系结构、运行模式及其扩充功能，并介绍了最新宏汇编 MASM6.0 版本的特点及使用方法；③ 书中附有大量的例题、习题及实验示例，其中例题和实验示例均已上机通过，有相当部分例题来源于科研实践；④ 本书的实验部分内容丰富，覆盖了各章节的有关内容，有利于巩固和掌握所学的理论知识。

该教材由中国矿业大学、山西矿业学院、淮南矿业学院、焦作工学院、湘潭工学院、河北建筑科技学院和黑龙江矿业学院等七院校编写。本书由张虹主编，李东生副主编，山东矿业学院吴哲辉教授主审。张虹编写第三、十三、十六章，李东生编写第四、十四章，潘瑜编写第七、八章，李明学编写第二、十二、十五章，杨立身编写第五、六章，沈宏远编写第九、十、十二、十六章，杨荣爱编写第一、十二、十五章。

本书的出版，得到了煤炭工业部所属各院校有关领导和教师的关心与支持，并得到了教材规划室瓮立平副编审和中国矿业大学出版社何戈编辑的热心帮助，主审吴哲辉教授认真审阅了全书并提出指导意见，中国矿业大学张振环老师为实验示例的调试做了许多工作，在此谨向他们表示衷心感谢。

编　者

1997 年 12 月

目　录

第一章　基础知识

第一节　计算机语言

为了让计算机完成某一任务，就必须给计算机下达指令，且计算机也能以特定的方式给出处理结果，这个过程就涉及到一个“语言”问题，我们把这种人和计算机交流信息所使用的语言称为计算机语言，又称为程序设计语言。

计算机语言分为机器语言、汇编语言和高级语言三种类型。

一、机器语言

由于计算机只能识别和执行由二进制代码 0 和 1 表示的指令，所以在计算机应用的初期，人们直接使用二进制代码形式的语言来编写程序，这种语言称为机器语言。例如，完成将数值 10 送累加器 AL，并将 AL 的内容加 10，结果仍存放在 AL 中的操作，用 8086 的机器指令可表示为

1011000000001010

0000010000001010

机器语言是面向机器的程序设计语言，即机器指令与计算机的硬件是密切相关的，它是为特定计算机系列或计算机设计的，所以机器语言有通用性差、指令难以记忆、书写困难、容易出错且调试修改麻烦等缺点。但是机器语言也有它的优点，即用机器语言编写的程序，不需要任何处理，可直接送入计算机运行，并且能够直接控制计算机硬件的操作，因此用机器语言编写的程序，能充分发挥计算机硬件的功能，且运行速度快、占用内存空间少，所编写的程序较精练。

二、汇编语言

由于机器语言是二进制代码形式的语言，难记忆，易出错，所以可用符号来代替二进制代码，即用助记符来代替操作码，用地址符号来代替地址码，于是就产生了机器指令符号化描述的语言，即汇编语言，所以汇编语言有时也称为符号语言。每一条汇编语言指令对应于一条机器指令，它们之间的关系是一一对应的。例如：

汇编语言指令	机器指令	
MOV　AL，　10	1011000000001010	；数值 10 送累加器 AL
ADD　AL，　10	0000010000001010	；AL 的内容加 10，和值存放在 AL 中

从上例可以看出，汇编语言比机器语言有了很大的进步，但仍是面向机器的程序设计语言，它是为特定的计算机系统设计的，不同的计算机具有各不相同的汇编语言，所以汇编语言的通用性较差。但汇编语言除了保持机器语言的编程质量高、执行速度快、占用内存少的

优点外，还具有易记忆、易理解、书写调试较方便等优点。

用汇编语言编写的程序称为汇编语言源程序，它不能被计算机直接识别和执行，必须经过逐条翻译，变成由 0 和 1 代码组成的机器语言表示的目标程序，计算机才能理解和执行。完成翻译工作的程序称之为汇编程序。汇编程序的工作过程如图 1-1 所示。

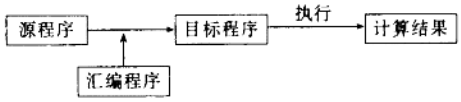


图 1-1　汇编工作过程示意图

三、高级语言

高级语言是由“词”和数学公式按照一定的语法规则组成，这就与我们使用的自然语言和数学表示十分接近。例如 BASIC 语言中，以 Y＝X＋5 表示将 X 变量的值加 5，和值送给变量 Y；用语句 INPUT A，B 表示让用户从键盘键入两个数据分别送给变量 A 和 B。由此可以看出，高级语言与自然语言的语法体系十分相近，易学易记、编程容易、调试修改方便，同时它又与具体的计算机无关。用某一高级语言编写的源程序，不经修改或仅作少量修改，可以在任何一种类型的计算机上运行。所以高级语言具有很好的通用性。

与汇编语言类似，用高级语言编写的源程序不能被计算机直接执行，必须经过“翻译程序”或“解释程序”将它翻译成等价的机器语言表示的目标程序，计算机执行目标程序才能得到预期的结果。

计算机执行用高级语言编写的源程序有两种方式：编译执行方式和解释执行方式。

编译执行方式：用户将用高级语言编写的源程序输入计算机后，调用系统中的编译程序将源程序全部进行翻译，生成目标程序，再对目标程序连接和装配形成可执行文件，然后执行该可执行文件，得到计算结果。如果在翻译过程中，发现源程序有错误，编译程序会给出错误信息。用户可根据错误信息修改源程序，然后再重复上述过程。虽然编译方式在每次修改源程序后，都要重新编译，比较麻烦，但它执行速度快。编译工作过程如图 1-2 所示。

解释执行方式：用户将源程序输入到计算机内存后，可调用系统的解释程序对源程序进行扫描，解释一句，执行一句，不生成目标程序。如果发现错误随时修改，然后再重新运行。因此解释执行的工作方式运行速度慢。解释工作过程如图 1-3 所示。

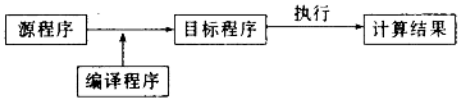


图 1-2　编译工作过程示意图

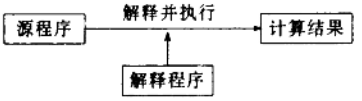


图 1-3　解释工作过程示意图

四、汇编语言的用途

高级语言的出现，使不懂计算机内部结构的人员也能很方便地使用计算机，这对计算机的推广和普及起到了很大的促进作用，但它并不能取代汇编语言。因为用高级语言编写的源程序，经翻译后形成的目标程序，与有经验的程序员用汇编语言编写的程序相比，前者比后者程序要长 15%～200%，执行时间要长 15%～300%。所以高级语言比汇编语言要求的内存容量大，运行时间长。另外，汇编语言能够利用计算机所有硬件特性并能直接控制硬件，因

此汇编语言广泛应用于计算机系统软件设计、计算机实时控制和计算机通讯等领域。

第二节　数字与字符在机器内部的表示

众所周知，计算机内部是采用二进制进行操作和运算的，二进制中的“0”和“1”分别对应于表示数据的物理器件的两个稳定状态。采用二进制计数系统，数字与字符在计算机中都采用由 0 或 1 组成的一串代码表示。

一、数字的表示

1. 十进制数和二进制数的表示

十进制采用“逢十进一”的计数方法。它使用的数码有十个，即 0，1，2，…，9，其基数为 10。对于十进制数 1234 可表示为

$1234=1\times10^3+2\times10^2+3\times10^1+4\times10^0$

对任意十进制正数可表示为

$N=K_n\times10^n+K_{n-1}\times10^{n-1}+\cdots+K_0\times10^0+K_{-1}\times10^{-1}+\cdots+K_{-m}\times10^{-m}$

二进制则采用“逢二进一”的计数方法，它使用的数码有两个，即 0 和 1，其基数为 2。

二进制数 1010 可表示为

$1010=1\times2^3+0\times2^2+1\times2^1+0\times2^0=8+0+2+0=(10)_{10}$

对于任意二进制正数则可表示为

$S=K_n\times2^n+K_{n-1}\times2^{n-1}+\cdots+K_0\times2^0+K_{-1}\times2^{-1}+\cdots+K_{-m}\times2^{-m}$

其中，2^R(R=n，n−1，…，0，−1，…，−m)称为相应位的权。二进制整数前 8 位的权位如表 1-1 所示。

表 1-1

二进制位	7	6	5	4	3	2	1	0
权	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
十进制数值	128	64	32	16	8	4	2	1

将十进制整数转换为二进制数可采用“除 2 取余”法，即将待转换的十进制整数不断除以 2，并记下余数，直到商为 0 为止，则余数的倒排序列即为该数对应的二进制数。

例 1-1　将十进制数 46 转换为二进制数

商	余数
46/2=23	0
23/2=11	1
11/2=5	1
5/2=2	1
2/2=1	0
1/2=0	1

则$(46)_{10}=(101110)_2$

将十进制小数转换为二进制小数，可将小数部分乘以 2，然后取结果的整数部分，重复此操作，直到小数部分为 0 或得到的整数序列满足精度要求为止。此整数序列的正序排列即

为该十进制小数对应的二进制数。

例 1-2 将十进制小数 0.65625 转换为二进制小数

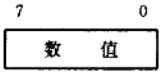
	结果小数部分	结果整数部分
0.65625×2	0.3125	1
0.3125×2	0.625	0
0.625×2	0.25	1
0.25×2	0.5	0
0.5×2	0.0	1

则 $0.65625=(0.10101)_2$

对于既包括整数部分又包括小数部分的十进制数可采用上述方法分别将整数和小数部分分别转换为对应的二进制数，最终形成该十进制数对应的二进制数。

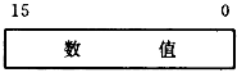
2. 无符号二进制数的表示

对于 8 位无符号二进制数，其机器内部表示格式为



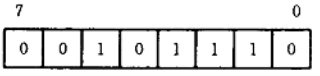
其表示数值的范围为 0～255。

对于 16 位无符号二进制数，其机器内部表示格式为



其表示数值的范围为 0～65535。

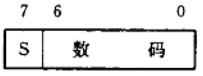
例 1-3 $(46)_{10}=(101110)_2$，其机内 8 位表示为



3. 有符号数的表示

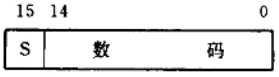
为了能在计算机内表示正数和负数，可将每个表示数据的字节或字的最高位设置成符号位。当符号位为 0 时，表示该数为正数，当符号位为 1 时，表示该数为负数。IBM—PC 系列机采用补码(2's Complement)表示有符号数。对于正数，其补码同其原码，负数的补码则采用将其原码除符号位以外的每位求反且末位加 1 的方法求得。

8 位有符号二进制数的格式



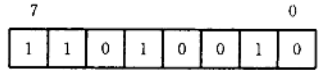
其中，S 为符号位，数值的表示范围为－128～＋127

16 位有符号二进制数的格式



其中，S 为符号位，数值的表示范围为－32768～＋32767

例 1-4 $(-46)_{10}$其 8 位补码为 11010010，机内表示为



4. BCD(Binary Coded Decimal)码的表示

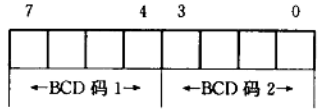
计算机采用二进制进行数据运算和处理是十分方便和必要的。考虑到人习惯于使用十进制数，为便于在计算机中直接使用十进制数，产生了用二进制编码的十进制数，即 BCD 码。BCD 码用 4 位二进制数表示一位十进制数码，由于此处 4 个二进制位的权值分别是 8421，所以 BCD 码又称为 8421 码。十进制数中每一个数码对应的 BCD 码如表 1-2 所示。

表 1-2

十进制数码	0	1	2	3	4	5	6	7	8	9
BCD 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

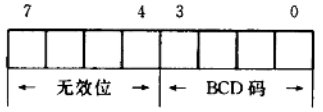
BCD 码有以下两种不同的存储方式。

第一种方式：组合型 BCD 码。



在组合型 BCD 码中，一个字节可以表示两个 BCD 码，又称压缩 BCD 码。

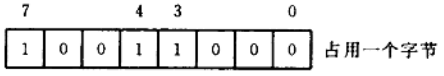
第二种方式：非组合型 BCD 码。



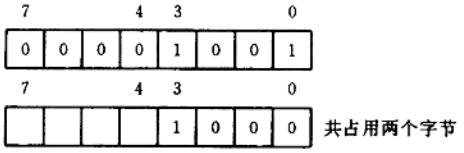
非组合型 BCD 码，一个字节表示一个 BCD 码，其高半字节的内容与 BCD 码无关，一般清 0，又称非压缩 BCD 码。

例 1-5 十进制数 98 的 BCD 码表示

若采用组合型 BCD 码可表为



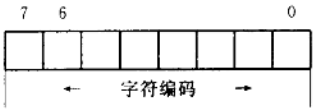
若采用非组合型 BCD 码可表为



二、字符的表示

IBM—PC 系列机中字符的表示采用的是 ASCII 码(American Standard Code for Infor-

mation Interchange)，即美国标准的信息交换码。标准的 ASCII 码是用一个字节中的七位二进制表示字符编码，其中最高位(D7 位)是奇偶校验位。



参见本书附录 1，标准的 ASCII 码表中共有 128 个字符，分为可打印的 ASCII 字符和非打印的 ASCII 字符(控制字符)。

可打印的 ASCII 字符包括：

数字 0～9，其 ASCII 码为 30H～39H

大写字母 A～Z，其 ASCII 码为 41H～5AH

小写字母 a～z，其 ASCII 码为 61H～7AH

空格符(space)，其 ASCII 码为 20H

非打印的 ASCII 字符(控制字符)包括：

BEL，响铃，其 ASCII 码为 07H

CR，回车，其 ASCII 码为 0DH

LF，换行，其 ASCII 码为 0AH

非打印的 ASCII 码主要用于系统控制。

第三节　数字与字符在程序中的表示

一、各种进制数在程序中的表示

在计算机内部，所有的数据信息都是以二进制的形式存放的。对人而言，在程序设计当中使用二进制数是相当不方便的。所以在汇编语言程序设计中，除了使用二进制外，还经常用到八进制、十进制、十六进制以及科学计数法。下面分别对几种常用的数字表示方法进行讨论。

1. 十进制数(Decimal)

在程序中，数字序列后面加字母 D 或省略 D，都表示此数是一个十进制数。例如：

ADD AX，10D 或 ADD AX，10

ADD AX，－40D 或 ADD AX，－40

上两条指令中的 10 和－40 都是十进制数。

2. 二进制数(Binary)

程序中对二进制数的表示是在二进制数后面加字母 B。例如：

ADD AX，1010B

此语句中的 1010 表示的是一个二进制数值。在程序中使用二进制数时应注意负数的书写形式。

3. 八进制(Octal)

八进制是基数为 8 的计数方式，它遵循的原则是“逢八进一”，组成八进制的数码共有 8

个，它们分别为0、1、2、3、4、5、6、7。这8个数码与二进制、十进制数的关系如表1-3所示。

表1-3　　八进制数码与二进制、十进制数的对应关系

十进制数	0	1	2	3	4	5	6	7
二进制数	000	001	010	011	100	101	110	111
八进制数	0	1	2	3	4	5	6	7

因八进制的基数8是2的幂，所以二、八进制转换非常容易。把一个二进制数转换为八进制数的方法是：按照从低位到高位的顺序三位一组进行划分，最后不足三位用0补齐，然后直接用每三位二进制数对应的八进制数码来表示即可。例如把二进制数1010110转换为八进制数的方法为：

```
001 010 110
 1   2   6
```

即1010110的八进制数为126。

在程序设计中，对八进制的表示是在0～7数字序列后面加字母O或Q。例如：

```
ADD AX,12Q
```

若数值为负数时，应注意书写形式。

4. 十六进制(Hexadecimal)

十六进制是汇编语言程序设计中常用的一种计数方式。它的基数为16，遵循“逢十六进一”的原则。十六进制的数码有16个，分别是数字0～9，英文字符A到F。这16个数码与十进制数、二进制数的关系如表1-4所示。

十六进制的基数16是2的幂，因此二进制与十六进制数的转换类似于二、八进制转换。把一个二进制数转换为十六进制数的方法为：按照从二进制数的低位到高位的顺序四位为一组进行划分，最后不足四位，用0补足高位，然后直接用十六进制数码替换各组的值即可。例如：

```
0101  1001  0011  1101
 5     9     3     D
```

即二进制数101100100111101对应的十六进制数为593D。

表1-4　　十六进制数码与十进制、二进制数的对应表

十进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
十六进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

在程序中，十六进制数的表示方法为：在十六进制数码序列后加上英文字符H。例如：

```
ADD AX,12H
```

在此语句中，12是十六进制数。

在程序中使用十六进制数应注意以下两点：

① 若数值的第一个字符是英文字母A到F中的一个，那么在数值之前应加一个0以和

符号名相区别。例如：

ADD AX,0A123H

② 若数值是负数时，应用补码表示。如把－5 加到 AX 寄存器上，－5 的补码用两个字节来表示为 1111111111111011B，转换为十六进制数码为 FFFBH，如：

ADD AX,0FFFBH

二、字符的表示

在程序中，字符或字符串被括在单引号或双引号中，也可以使用字符的 ASCII 码值表示。如下语句中，把字符 A 传送给寄存器 AL：

MOV AL,'A'

其中，'A'可用 ASCII 值 41H 表示，上语句可写为：MOV AL,41H。

例 1-6
```
MOV  DL,0DH
MOV  AH,2
INT  21H
```
实现光标换行功能

习 题 一

1.1 什么是机器语言、汇编语言和高级语言？比较三种语言的优缺点。

1.2 何谓汇编、编译和解释？它们适合于什么语言。

1.3 将下列十进制数转换为二进制数和十六进制数。

(1)102　(2)23767　(3)67　(4)1032

1.4 将下列二进制数转换为十进制数和十六进制数。

(1)10111010　(2)11111　(3)10000000　(4)11110000

1.5 写出下列带符号数的二进制补码表示(字节存放)。

(1)＋78　(2)－10　(3)－128　(4)＋125

1.6 下列各数为十六进制表示的 8 位二进制数，如果它们分别被看作是有符号数和无符号数时，请写出它们所表示的十进制数。

(1)A0　(2)FF　(3)12　(4)F1

1.7 请写出下列字符或操作的 ASCII 码值。

(1)0　(2)a　(3)B　(4)空格　(5)回车　(6)换行　(7)$

1.8 写出下列各浮点数的内存表示。

(1)1.23E2　(2)－0.625E－1　(3)1.0E1　(4)3.90626E－2

1.9 用指令完成下列十六进制数的加法运算。

(1)1234H＋0034H　(2)01ABH＋5BA3H　(3)5567H＋0881H。

第二章　Intel 8086/8088 CPU 的功能结构

第一节　计算机系统组成

计算机系统包括硬件和软件两部分。硬件包括中央处理机、存储器、输入/输出设备等。软件则包括操作系统软件、系统应用软件和用户应用软件等各种类型的软件。

一、硬件

典型计算机的结构可用图 2-1 表示。其中包括中央处理机 CPU (Central Processing Unit)、存储器(Memory)和输入/输出(I/O)子系统三个主要组成部分，通过系统总线把它们连接在一起。

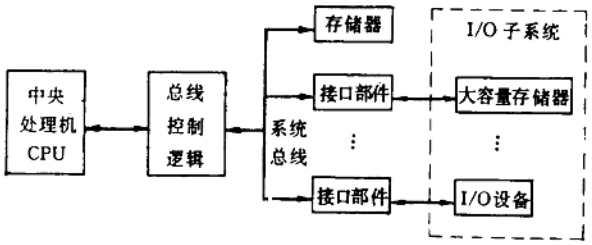


图 2-1　计算机结构

中央处理机包括执行单元 EU(Execution Unit)和总线接口单元 BIU(Bus Interface Unit)两部分。执行单元 EU 负责对指令进行分析执行以及算术运算和逻辑运算等操作，而总线接口单元 BIU 则负责从存储器中读取指令和数据、向存储器中存数据、形成物理地址以及与外部接口的操作。

存储器是计算机的记忆部件。人们编写的程序(由指令序列组成)在执行或编辑时就存放在这里，它也可以存放程序中所使用的数据和中间结果。常把该存储器视为机器内部的存储器，或称为内存。

I/O 子系统一般包括 I/O 设备和大容量存储器两类外部设备。I/O 设备是指计算机与外部世界进行信息交流用的输入/输出设备，如显示终端、键盘、打印机和扫描仪等多种类型的外部设备。大容量存储器则是指可存储大量信息的外部存储器，如磁盘、磁带、光盘等外存储器，常称为外存。由于内存的容量有限，所以计算机用外存作为内存的后备存储设备。外存的容量可以比内存大很多，但存取信息的速度要比内存慢得多，所以除必要的系统程

序外，一般程序（包括数据）是存放在外存中的。只有当运行时，才把它从外存传送到内存的某个区域中，再由中央处理机控制执行。

系统总线把 CPU、存储器和 I/O 设备连接起来，用来传送各部分之间的信息。系统总线包括数据总线、地址总线和控制总线三大类，除此之外还有电源线和地线等。数据总线是用来传送数据的；地址总线是用来指明存储器和外设地址的；控制总线则是传送控制信息的。系统总线的工作是由总线控制逻辑负责指挥。接口部件起总线驱动和信息缓冲等作用。

二、软件

计算机软件是计算机系统的重要组成部分，它可以分为操作系统软件、系统应用软件和用户应用软件三大类。操作系统软件主要是对计算机硬件系统和软件系统进行管理的一组程序，例如，PC—MSDOS 操作系统。系统应用软件主要是给用户提供一个开发环境，用来开发有特殊用途的一组程序。例如，文本编辑程序、调试程序、连接程序等。用户应用软件则是由用户在系统应用软件基础上自行编制的各种程序。图 2-2 表示了计算机软件的层次。

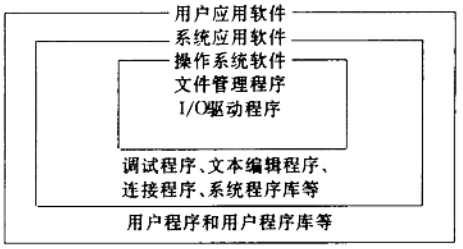


图2-2　计算机软件层次图

三、计算机系统

基于 Inter CPU 的 PC 机是由美国 IBM 公司 1981 年推出的个人计算机发展起来的。系统基本组成如图 2-3 所示。其基本配置包括主机板、显示器、键盘、软硬盘驱动器、电源、扬声器等几大部分。

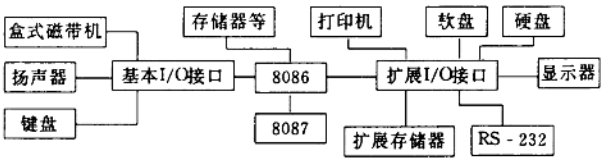


图2-3　PC机系统组成

CPU 与外部设备交换信息是通过 I/O 接口进行的。PC 机有两部分 I/O 接口：一部分是基本 I/O 接口，它与盒式磁带机、扬声器和键盘相连；另一部分是扩展 I/O 接口，它就是我们常说的扩展总线，即 PC/XT 总线。随着 CPU 的升级，PC/XT 总线已经不能满足要求了，因此逐渐产生了 ISA（Industry Standard Architecture）总线、MCA（Micro Channel Architecture）总线（微通道结构总线）、EISA（Extended Industry Standard Architecture）总线、VESA（Video Electronics Standards Association）局部总线（VL－Bus）和 PCI（Peripheral Compponent Inter-

connect)总线。外部扩展总线接口以主板插槽形式与外部设备连接。在标准配置的 PC 机插槽中插有显示适配器和驱动适配器(包括软盘接口、硬盘接口、并口、串口、游戏口等,也称多功能卡)。有的 PC 机的多功能卡和显示适配器等已被制造在主机板上了。有的 PC 机中还插有扩展存储器板。当然,还可以插其他板卡,如网卡、声卡、视卡、解压卡、MODEM 卡等。

第二节 8086/8088 CPU 的寄存器

在 8086/8088 CPU 的执行单元 EU 和总线接口单元 BIU 中有 14 个寄存器,在计算机中起着重要的作用,每一个寄存器相当于存储器中的一个存储单元,但它的存取速度比存储器要快得多。用它来存放计算过程中所需要的各种信息,例如,操作数地址、操作数和运算的中间结果等。这些寄存器按用途可分为通用寄存器组,段寄存器组,指针、变址寄存器和标志寄存器。如图 2-4 所示。

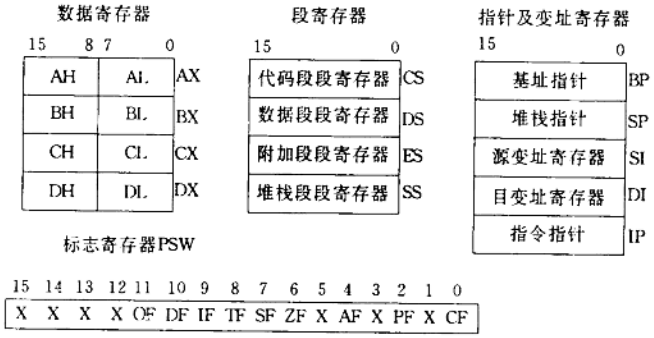


图2-4 8086/8088 CPU内部寄存器

一、通用寄存器组

通用寄存器中包括数据寄存器、地址指针和变址寄存器。

1. 数据寄存器

数据寄存器包括 AX、BX、CX、DX,用来暂存计算过程中所用到的操作数、结果或其他信息。它们都可以以字(16 位)的形式访问,也可以以字节(8 位)的形式访问,例如对 AX、BX、CX、DX 可以分别被访问高 8 位字节 AH、BH、CH、DH 或者低 8 位字节 AL、BL、CL、DL。这四个寄存器都是通用寄存器,但它们各自又有专门的用途。

AX(Accumulator) 累加器。是算术运算的主要寄存器,特别是进行乘、除法运算时,所有的 I/O 指令都使用这一寄存器与外部设备传送信息。

BX(Base) 基址寄存器,常用来存放地址。

CX(Count) 计数寄存器,在循环(LOOP)和串处理指令中作为隐含的计数器。

DX(Data) 数据寄存器,一般在作双字长运算时,例如乘、除运算,把 DX 和 AX 组合在一起存放一个双字长的数据,DX 用来存放高位字,AX 用来存放低位字。对于某些 I/O 操作,DX 可用来存放 I/O 的端口地址。

2. 指针与变址寄存器

SP、BP、SI、DI 四个 16 位的通用寄存器。它们可以像数据寄存器一样在运算过程中存放操作数，但只能以字(16 位)为单位使用。此外，它们更多的用途是在段内寻址时提供偏移地址。SP(Stack Pointer)为堆栈指针寄存器，它指向堆栈的栈顶。BP(Base Pointer)为基址指针寄存器，用来表示堆栈段的偏移地址。SI(Source Index)源变址寄存器和 DI(Destination Index)目的变址寄存器一般与 DS 连用，用来确定数据段中某一存储单元的地址。这两个变址寄存器有自动增量和自动减量的功能，所以用于变址是很方便的。在串处理指令中，SI 和 DI 作为隐含的源变址和目的变址寄存器使用时，SI 和 DS 联用，DI 和 ES 联用，分别达到在数据段和附加段中寻址的目的。

二、段寄存器

段寄存器是用来存放段基地址的，总线接口单元 BIU 设置四个段寄存器，CPU 可访问存储器中四个不同的段(每段 64K 字节)。四个段寄存器的作用分别是：

CS(Code Segment)　代码段寄存器　它存放当前代码的段基地址。CS 的内容左移四位再加上指令指针 IP 的内容就是下一条要执行指令的地址。

DS(Data Segment)　数据段寄存器　它存放当前数据段的段基地址。DS 的内容左移四位再加上按指令中存储器寻址方式计算出来的偏移地址，即为对数据段指定单元进行读写的物理地址。

SS(Stack Segment)　堆栈段寄存器　它存放当前堆栈段的段基地址。

ES(Extra Segment)　附加数据段寄存器　附加数据段又称附加段，它存放当前附加段的段基地址。

DS、ES 和 SS 都要由用户用程序设置初值，其内容在第三章中介绍，这里不再赘述。

三、指令指针和标志寄存器

IP(Instruction Pointer)指令指针寄存器，用来存放代码段中的偏移地址。在程序运行过程中，它始终指向下一条指令的首地址，与 CS 寄存器连用确定下一条指令的物理地址。当这一地址送到存储器后，控制器可以取得下一条要执行的指令；控制器一旦取得这条指令就马上修改 IP 的内容，使它指向下一条指令的首地址。由此可见，计算机就是用 IP 寄存器来控制指令序列的执行流程的，因此 IP 寄存器是计算机中很重要的一个寄存器。

PSW(Program Status Word)标志寄存器，为程序状态字寄存器，是一个 16 位寄存器，由条件码标志(OF、SF、ZF、AF、PF、CF)和控制码标志(DF、IF、TF)构成。其中，条件码标志用来记录指令执行结果的状态信息。由于这些状态信息常常作为后续条件转移指令的转移控制条件，所以称为条件码标志。它包括以下 6 种：

CF(Carry Flag)进位标志，记录运算时从最高有效位产生的进位或借位值。若最高有效位有进位时 CF 置 1，否则 CF 为 0，或者最高有效位有借位时 CF 置 1，否则 CF 为 0。

AF(Auxiliary carry Flag)辅助进位标志，记录运算时结果的低半部向高半部产生的进位或借位值。若有进位或借位，则 AF 置 1，否则 AF 为 0。对于加法运算判别第 3 位(8 位运算)或第 7 位(16 位运算)是否有进位，对减法运算判别相应位是否有借位。

ZF(Zero Flag)零标志，运算结果为 0 时 ZF 置 1，否则 ZF 为 0。

OF(Overflow Flag)溢出标志，运算的结果若超出了机器所能表示的范围则称为溢出，此时 OF 置 1，否则为 0。

SF(Sign Flag)符号标志，记录运算结果符号位的值，若符号位为 1，则 SF 置 1，否则为