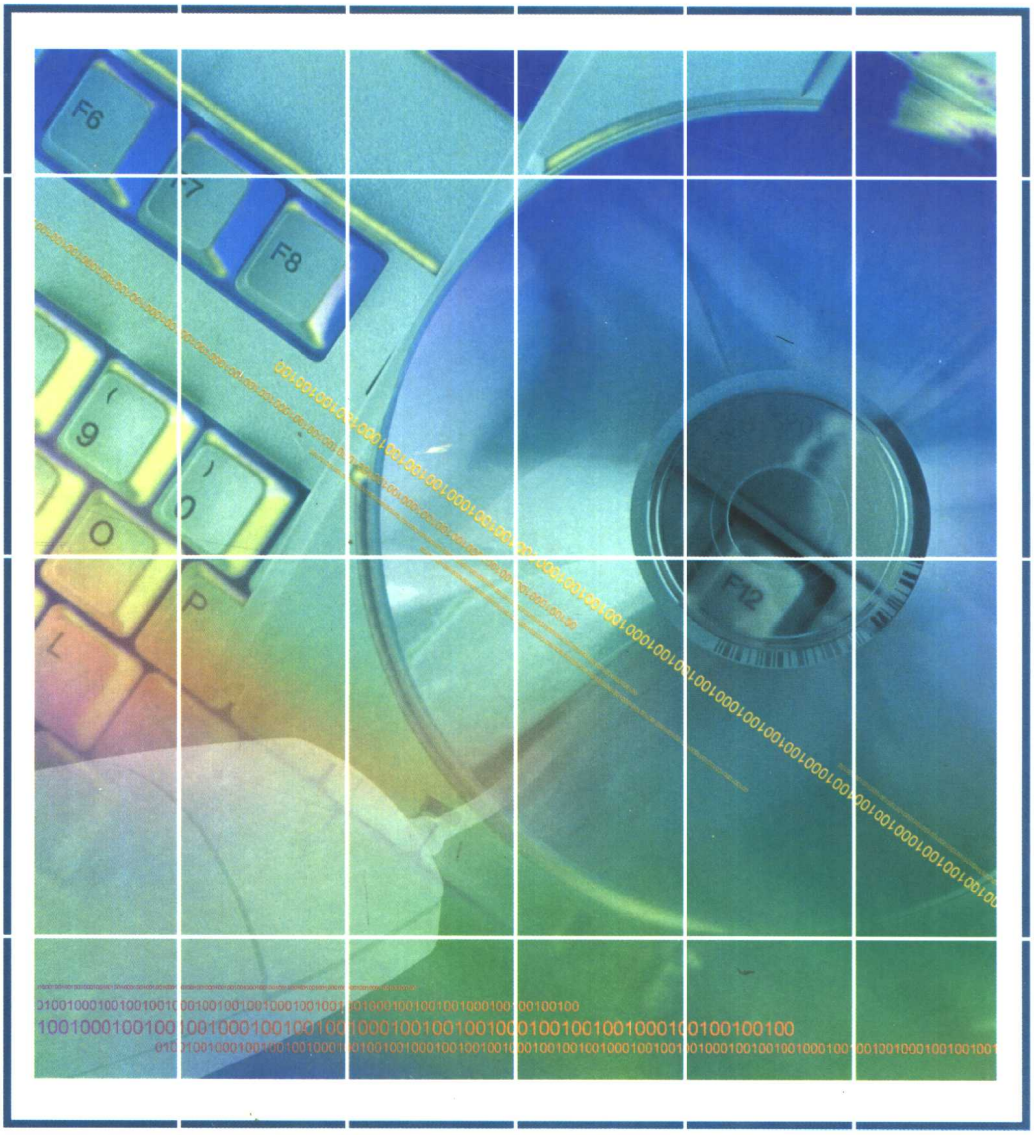
新世纪计算机类本科系列教材



PC汇编语言程序设计

王　闵　田玉敏　赵明禄　编著

西安电子科技大学出版社
http://www.xduph.com

□新世纪计算机类本科系列教材

PC 汇编语言程序设计

王　闵　田玉敏　赵明禄　编著

西安电子科技大学出版社

2001

内 容 简 介

本教材主要介绍汇编语言和汇编语言程序设计的基本技能和技巧。全书共分为11章，每章后都附有大量习题。各章的主要内容分别是：PC微型计算机的基本组成，MASM宏汇编语言，PC机的指令系统与寻址方式，DOS和BIOS功能调用，循环程序、子程序和输入/输出程序设计的基本方法，汇编语言与高级语言程序的连接。

本书可作为高等学校计算机专业本科生的教材，也可作为从事计算机专业的科技人员、电脑爱好者及各类自学人员的参考书。

图书在版编目(CIP)数据

PC汇编语言程序设计/王闵，田玉敏，赵明禄编著.
—西安：西安电子科技大学出版社，2001.6
新世纪计算机类本科系列教材
ISBN 7-5606-1019-6.

Ⅰ.P… Ⅱ.①王… ②田… ③赵… Ⅲ.个人计算机—汇编语言—程序设计—高等学校—教材 Ⅳ.TP313

中国版本图书馆CIP数据核字(2001)第20184号

责任编辑 陈宇光 戚文艳
出版发行 西安电子科技大学出版社(西安市太白南路2号)
电 话 (029)8227828 邮 编 710071
http://www.xduph.com E-mail:xdupfxb@pub.xaonline.com
经 销 新华书店
印 刷 西安兰翔印刷厂
版 次 2001年6月第1版 2001年6月第1次印刷
开 本 787毫米×1092毫米 1/16 印张 15.875
字 数 370千字
印 数 1～4000册
定 价 17.00元
ISBN 7-5606-1019-6/TP·0502

如有印装问题可调换

本书封面贴有西安电子科技大学出版社的激光防伪标志，无标志者不得销售。

前　言

本教材系新世纪计算机类本科系列教材。本教材是在编者多年教学实践的基础上编写而成的。为了使教材更符合学生的认知规律，我们将寻址方式、指令和伪指令等比较枯燥的内容分散并相互贯穿，通过例子使之具体化。例如将寻址方式放在常用指令中讲述，将指令和伪指令放在典型的程序段中讲述。本教材选材力求精练，所选例题既具实用性，又具一定的趣味性，便于学生在短时间内掌握汇编语言的基本内容及汇编语言程序设计的基本方法和技巧。

全书共分为11章，系统地介绍了宏汇编语言和汇编语言程序设计的特点和技巧。第一章简要介绍了数据编码及数据表示。第二章简要介绍了PC微型机的结构：CPU的组成及寄存器、主存的段结构和I/O端口。第三章系统介绍了操作数的寻址方式和转向地址的寻址方式。第四章分类介绍了PC机的指令系统，并通过典型的程序段介绍各类指令的使用，有助于读者体会到一些功能复杂的指令总可以由若干条最基本的简单指令实现。第五章介绍了MASM宏汇编语言中的基本知识；80X86CPU增加与增强的指令；汇编语言程序的书写格式和上机过程。第六章介绍了DOS系统功能子程序和BIOS功能子程序的调用。第七章重点介绍了循环指令，循环程序结构及各类循环控制方式。第八章介绍了过程的定义，子程序调用与返回指令，参数的传递方式以及子程序的嵌套和递归。第九章介绍了输入/输出指令，各种输入/输出控制方式的程序设计：直接方式、查询方式和中断方式以及用DOS、BIOS功能调用设计输入/输出程序。第十章介绍了MASM宏汇编语言中的宏操作，重复汇编及条件汇编。第十一章介绍了模块化程序设计以及汇编语言与高级语言的连接。本教材参考学时为40～50学时。

本教材的第一、二、三、四章由王闵编写，第五、六、七章由赵明禄编写，第八、九、十、十一章由田玉敏编写，全书由王闵统稿。老教授李伯成在百忙之中认真地审阅了全稿，并给予悉心指导和帮助。曾平教授、武波教授、裘雪红教授、顾新副教授对本书提出了许多有价值的意见和建议。在教材的编写过程中还得到计算机学院副院长武波教授的关心和支持。在此谨向上述专家学者表示衷心的感谢。

由于编者水平所限，在选材以及编排上难免存在不足，期盼同行及广大读者慷慨赐教。

编者

2001年1月

目　录

此为试读,需要完整PDF请访问: www.ertongbook.com

第一章　数值及其编码表示

计算机需要处理的信息除了数值信息外，还有各种非数值信息(如符号、文字、图像、语言等)。但计算机内部硬件只能表示两个数0和1，因而计算机只能对二进制的数字信息进行传送、处理和存储。要使计算机能处理各种各样的信息，则必须对这些信息进行编码。所谓编码，在计算机中意味着将0、1按一定的规则进行排列，用来表示并区分复杂多样的信息。

1.1　二进制数值的编码表示

为了使计算机能相对方便地进行加、减、乘、除等运算，必须对数值数据进行编码。编码除了要用"0"、"1"来表示正号(+)、负号(−)外，还要规定数值的表示方法。编码前的数据被称为真值，编码后的数据被称为码值。本章主要介绍常用的数值编码：原码和补码。

1.1.1　原码表示

原码(True Form)是比较容易理解的一种数值编码。原码用0表示正号，用1表示负号。原码的码值等于真值的绝对值。

根据原码的表示方法可知：$[+0]_{原}=000\cdots0$，$[-0]_{原}=100\cdots0$。就是说真值0在原码表示中不唯一，换句话说$[X]_{原}$与真值X不是一一对应的。

由真值转换为原码，则将正号用0表示，负号用1表示，数值位照写即可。由原码转换为真值，则将符号位0写成+，1写成−，数值位不变，可简单表示为：

$$\text{真值 X} \xleftrightarrow[\text{数值位不变}]{\text{符号}+\leftrightarrow 0,\ -\leftrightarrow 1} [X]_{原}$$

原码表示简单、直观，但用原码进行加减运算比较复杂。下面介绍应用更为广泛的另一种编码——补码。

1.1.2　补码表示

与原码相同，补码(Two's Complement)也用0表示正号，用1表示负号。当真值$X\geqslant 0$时，补码的码值等于真值X；当真值$X<0$时，补码的码值是将真值X各数值位按位取反，末位加1。

根据补码的表示方法可推知补码具有以下特点：

(1) 零的补码是唯一的，即$[0]_{补}=000\cdots0$，换句话说，$[X]_{补}$与X是一一对应的，在这

点上补码优于原码。

(2) 根据补码的表示方法可推知补码与真值、原码之间的转换关系。

当真值 $X \geqslant 0$ 时，正数的补码与原码形式相同，码值等于真值，由真值求其补码和原码，只需将“+”用“0”表示，数值位照写即可，反过来由补码或原码求真值，将符号位(最高位)的“0”写成“+”，数值位也照写即可求得，可简单表示为：

$$\text{当真值 } X \geqslant 0\text{：} X \xleftrightarrow[\text{数值位不变}]{\text{符号}+\leftrightarrow 0} [X]_{补} = [X]_{原}$$

当真值 $X<0$，即将负数的补码转换为真值时，只需将符号位 1 写为负号(−)，数值各位按位取反，末位加 1 即可，简单表示如下：

$$\text{当真值 } X<0\text{：} X \xleftrightarrow[\text{数值位按位取反，末位加1}]{\text{符号}-\leftrightarrow 1} [X]_{补}$$

因为原码体现的是绝对值，除了用“1”表示符号“−”以外，数值位与真值相同，故可将负数补码与原码之间的转换关系表示为：

$$\text{当真值 } X<0\text{：} [X]_{原} \xleftrightarrow[\text{数值位按位取反，末位加1}]{\text{符号位“1”不变}} [X]_{补}$$

例 1.1 已知真值 $X_1=+0.1011$，$X_2=0$，$X_3=-0.1101$，若用 5 位二进制数表示，求其原码及补码。

解：$[X_1]_{原}=0.1011$　　$[X_1]_{补}=0.1011$

$[X_2]_{原}=0.0000$　或$[X_2]_{原}=1.0000$　　$[X_2]_{补}=0.0000$

$[X_3]_{原}=1.1101$　　$[X_3]_{补}=1.0011$

例 1.2 已知整数补码$[X_1]_{补}=01010$，$[X_2]_{补}=10101$，$[X_3]_{补}=10000$，最高位代表符号，若用 5 位二进制数表示，求其真值及原码。

解：真值 $X_1=+01010=+10D$　　$[X_1]_{原}=01010$

真值 $X_2=-01011=-11D$　　$[X_2]_{原}=11011$

真值 $X_3=-10000=-16D$　　$[X_3]_{原}$超出表示范围

(3) 补码的算术右移。

算术右移就是除以 2 的运算。由$[X]_{补}$求$\left[\frac{1}{2}X\right]_{补}$的简便方法是把$[X]_{补}$连同符号位在内各位右移一位，同时符号位保持不变，可简单表示为：

$$[X]_{补} \xrightarrow[\text{各位右移一位}]{\text{符号位不变}} \left[\frac{1}{2}X\right]_{补}$$

例 1.3 已知$[X_1]_{补}=0.1110$，$[X_2]_{补}=1.0110$，$[X_3]_{补}=1.0000$，若用 5 位二进制数表示，求$\left[\frac{1}{2}X_1\right]_{补}=?$　$\left[\frac{1}{2}X_2\right]_{补}=?$　$\left[\frac{1}{2}X_3\right]_{补}=?$

解：$\left[\frac{1}{2}X_1\right]_{补}=0.0111$，$\left[\frac{1}{2}X_2\right]_{补}=1.1011$，$\left[\frac{1}{2}X_3\right]_{补}=1.1000$

(4) 补码的算术左移。

算术左移就是乘以 2 的运算，与算术右移不同，左移有可能产生溢出。由$[X]_{补}$求$[2X]_{补}$的简便方法是将$[X]_{补}$的各位左移一位，末位补 0，可简单表示为：

$$[X]_{补} \xrightarrow[\text{各位左移一位}]{\text{末位补0}} [2X]_{补}$$

例 1.4 已知补码$[X_1]_{补}=0.0101$，$[X_2]_{补}=1.0101$，若用 5 位二进制数表示，求$[2X_1]_{补}=?$　$[2X_2]_{补}=?$

此为试读，需要完整PDF请访问：www.ertongbook.com

解：$[2X_1]_{补}=0.1010$　　未溢出

$[2X_2]_{补}=\boxed{1}0.1010$　　溢出

(5) 补码的符号位扩展。

符号位的扩展只对定点整数补码而言，例如如何将一个 8 位表示的整数补码扩展为 16 位的整数补码，而其真值不变。

将整数补码扩大 n 位，并使其真值不变，只需将$[X]_{补}$的符号位向左复制 n 位即可。

例 1.5　已知$[X_1]_{补}=0101$，$[X_2]_{补}=1010$。将其扩展一个字节。

解：$[X_1]_{补}=00000101$

$[X_2]_{补}=11111010$

要将 n 位纯小数补码变为 2n 位，只需在末尾添加 n 个“0”即可。

1.2　二—十进制的编码表示

十进制数的基是 10，它不是 2 的整数幂，必须用二进制数对十进制数进行编码。一位十进制数由 0～9 共 10 个不同的数字字符表示，需要用四位一组的二进制来编码。而 $2^4=16>10$，这就使得二—十进制有多种编码方案，二—十进制编码简称为 BCD 码 (Binary - Coded Decimal)。二—十进制编码又可分为有权码和无权码两种。有权码与无权码都有多种编码方式。目前在计算机中应用最广泛的有权码是 8421 码，应用较广泛的无权码有余三码。这两种编码与十进制数码的对应关系如表 1 - 1 所示。

表 1 - 1　二—十进制编码对应关系

十进制	8421 码	余 3 码
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
非法	1010	1101
	1011	1110
	1100	1111
	1101	0000
	1110	0001
	1111	0010

8421 码 4 位二进制的权依次为 8，4，2，1，并由此得名。

8421 码中有效码是 0000～1001，1010～1111，共 6 种，为非法码，其编码简单，每个码相当于十进制数码所对应的二进制数。

从表 1－1 中可看到余 3 码中任何一个编码的二进制数都比它对应的十进制数多 3，故称为余 3 码。余 3 码的非法码也有 6 个，即 0000，0001，0010，1101，1110，1111。

1.3 定点数与浮点数

计算机中根据小数点位置是否固定，将数的表示分为定点数表示与浮点数表示。

1.3.1 定点数

定点数是指小数点位置固定不变的数。小数点的位置通常只有两种约定，小数点约定在最低数位右面的数称为定点整数，可用来表示一个纯整数。小数点约定在符号位右面，最高数位左面的数称为定点小数，可用来表示一个纯小数。

无符号定点整数，即正整数，不需设符号位，所有各数位都用来表示数值大小，并约定小数点在最低数位右面。

在定点整数或定点小数的表示法中，参加运算的数以及运算的结果必须在该定点数所能表示的数值范围之内，否则“溢出”。当发生溢出时，CPU 中的状态标志寄存器 FR 中的溢出标志位 OF 置位(即 1→OF)，并转入溢出处理。

1.3.2 浮点数

定点数的表示比较单一，要么纯整数，要么纯小数，表示数的范围比较小，运算过程中也很容易发生溢出。计算机中也引入了类似于十进制的科学标识法(如 $1.234\ 56\times10^4$)来表示二进制实数，这种方法用来表示值很大或很小的数，也可用它来表示既有整数又有小数的数。这种表示法称为浮点表示法，其小数点的实际位置随指数(阶)的大小而浮动。

浮点数由两部分组成：阶码 E(Exponent)和尾数 M(Mantissa)。浮点数表示的数值为 $M\times R^E$。若尾数 M 为 m 位，阶码 E 为 e 位，则典型的浮点数格式如图 1.1 所示。

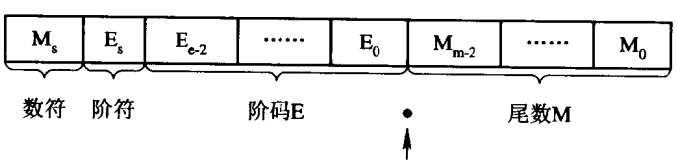


图 1.1 浮点数格式

图中，E 是阶码，即指数，为带符号定点整数，可用补码表示。E_s 为阶符，表示阶的正负。阶为正，小数点实际位置向右浮动，阶为负，小数点实际位置向左浮动。

M 是尾数，是带符号的定点纯小数，常用补码表示。M_s 是尾数的符号位，安排在最高位，表示该浮点数的正负。

小数点的位置约定在阶码最低位的右面，尾数最高数值位的左面，如图 1.1 所示。

R 是阶码的底，也就是尾数 M 的基(Radix)。一般基定为 2，它是隐含约定的。

浮点数的表示范围主要由阶码的位数决定，精度则主要由尾数的位数决定。

1.4 非数值信息的表示

在计算机中信息的概念是广泛的，除了数值信息外，还有大量的非数值信息，如逻辑信息、字符信息及汉字信息等，这些信息在计算机中也必须用二进制代码形式表示。本节简要介绍各种非数值信息的表示。

1.4.1 逻辑数据

逻辑数据是用一位二进制数来表示的。因为一位二进制数只具有两种可能的值：0 或 1，可以直接表示事物相对立的两个方面。比如一个事件成立用“1”表示，不成立则用“0”表示，另外还有“真”和“假”、“是”和“否”、“有”和“无”，都可以看作逻辑数据。

逻辑数据在计算机中虽然也具有“0”或“1”的形式，但是逻辑数据的“0”和“1”代表的是逻辑概念，完全没有“0”和“1”的数值概念，而且逻辑数据的取值只有“0”和“1”两个值，不可能再有其它值。逻辑数据表达的是事物的逻辑关系，而数值数据表达的是事物的数量关系。

计算机便是建立在与、或、非等逻辑运算基础之上的。计算机通过对逻辑数据的比较、判断、运算，可以完成逻辑推理、定理证明等一系列复杂的工作。

1.4.2 字符编码 ASCII

计算机中除了使用数字外，还大量使用英文字母及一些符号，这些符号称为“字符”。

目前使用最广泛的字符编码方案是美国国家信息交换标准代码（American Standard Code for Information Interchange），简称 ASCII 码。ASCII 码选用了 128 个常用字符，用 7 位二进制编码，如果再加上一位奇偶校验位，则正好是用一个字节表示一个字符的 ASCII 码。表 1－2 给出了 128 个字符与其 ASCII 码的对应关系，表中的 ASCII 码用十六进制码表示。

从表 1－2 中可以看出，ASCII 码包括 0～9 共 10 个数字字符（其值分别为 30H 加上相应的数字值），它与数据的二—十进制编码是两回事，26 个大写英文字母（其值分别为 40H 加上字母的顺序号），26 个小写英文字母（其值分别为 60H 加上字母的顺序号），一些通用符号和一些控制字符。这些字符的种类可满足各种程序设计语言、控制命令、西文文字等的需要。我国原电子部规定的一套部标字符编码与表 1－2 基本相同。

在计算机中，一个字符的 ASCII 码占用主存储器的一个字节单元；如果是字符序列，则占用主存多个连续的字节单元。

通用键盘的大部分键，与最常用的 ASCII 编码的字符相对应。当敲击键盘上某字符键时，由译码电路产生与该字符对应的 ASCII 码。计算机处理的结果也常以 ASCII 码形式输出，供显示与打印使用。

表 1－2　ASCII 字符编码

ASCII 码(H)	字符	ASCII 码(H)	字符	ASCII 码(H)	字符
00	NUL	2B	+	56	V
01	SOH	2C	,	57	W
02	STX	2D	—	58	X
03	ETX	2E	.	59	Y
04	EOT	2F	/	5A	Z
05	ENQ	30	0	5B	[
06	ACK	31	1	5C	\
07	BEL	32	2	5D	]
08	BS	33	3	5E	↑
09	HT	34	4	5F	—
0A	LF	35	5	60	`
0B	VT	36	6	61	a
0C	FF	37	7	62	b
0D	CR	38	8	63	c
0E	SO	39	9	64	d
0F	SI	3A	:	65	e
10	DLE	3B	;	66	f
11	DC1	3C	<	67	g
12	DC2	3D	=	68	h
13	DC3	3E	>	69	i
14	DC4	3F	?	6A	j
15	NAK	40	@	6B	k
16	SYN	41	A	6C	l
17	ETB	42	B	6D	m
18	CAN	43	C	6E	n
19	EM	44	D	6F	o
1A	SUB	45	E	70	p
1B	ESC	46	F	71	q
1C	FS	47	G	72	r
1D	GS	48	H	73	s
1E	RS	49	I	74	t
1F	US	4A	J	75	u
20	SP	4B	K	76	v
21	!	4C	L	77	w
22	"	4D	M	78	x
23	#	4E	N	79	y
24	$	4F	O	7A	z
25	%	50	P	7B	{
26	&	51	Q	7C	|
27	'	52	R	7D	}
28	(	53	S	7E	~
29	(	54	T	7F	DEL
2A	*	55	U		

习 题 一

1.1 分别写出下列各十进制数的原码和补码，用 8 位二进制小数表示(含一位符号位)：

-1，$+0$，-0，$\frac{37}{64}$，$-\frac{51}{128}$，0.375，0.5625

1.2 分别写出下列各十进制整数的原码和补码，用 8 位二进制整数表示(含一位符号位)。

-128，$+127$，-127，105，-64，-1

1.3 若认为小数点约定在 8 位二进制数的最右端(整数)，试分别写出下列各种情况下 X、Y 的十进制真值。

(1) $[X]_{补}=[Y]_{原}=00H$　　(2) $[X]_{补}=[Y]_{原}=50H$

(3) $[X]_{补}=[Y]_{原}=80H$　　(4) $[X]_{补}=[Y]_{原}=C0H$

(5) $[X]_{补}=[Y]_{原}=FFH$

1.4 已知$[X]_{补}=3EH$，$[Y]_{补}=DCH$，求：

$[2X]_{补}$，$[2Y]_{补}$，$\left[\frac{1}{2}X\right]_{补}$，$\left[\frac{1}{4}Y\right]_{补}$，$[-X]_{补}$，$[X]_{原}$，$[Y]_{原}$

1.5 维持真值不变，将下列整数补码扩展为 8 位：

1011B，　0101B，　9H，　6H

1.6 一个用 8 位二进制数表示的整数补码，如何判断其正负？如何判断其奇偶？如何判断其能否被 4 整除？

1.7 若字长为 n 位二进制数，可用来表示多少个不同的数？若用来表示无符号的整数，请写出所能表示的最大值和最小值；若为原码表示的整数，请写出所能表示的最大值和最小值；若为补码表示的小数，请写出所能表示的最大值和最小值。

1.8 下列代码若看作 ASCII 码、整数补码、8421 码时分别代表什么？

77H，37H，30H，58H，78H

第二章　PC 微型计算机的组织结构

一个微型计算机系统由运算器，控制器，存储器，输入设备，输出设备五大功能部件组成。其中，运算器和控制器通常做在一块芯片上，称为中央处理单元，简称 CPU(Central Processing Unit)，在微型机上也常称为“微处理器”。微处理器中包括算术逻辑单元 ALU (Algorithm Logical Unit)、通用寄存器和控制逻辑，它是计算机系统的核心部件。存储器由主存和辅存组成，它是存放程序和数据的场所，是组成计算机系统的存储空间。输入、输出设备通过接口与系统相连，接口中的数据缓冲寄存器和命令/标志位寄存器等组成了系统的 I/O 空间。

2.1　INTEL 8088/8086 微处理器的组成结构

INTEL 8088 是准十六位微处理器，即内部结构是十六位，外部的数据总线是八位，一个总线周期内只能输入、输出一个字节。而 INTEL 8086 的内部结构与外部总线均是十六位，一个总线周期内可输入、输出一字。但两者在组织结构上类似，指令编码和寻址方式完全相同，软件完全兼容。

INTEL 8088/8086 共有 14 个十六位寄存器，如图 2.1 所示。可分为：通用寄存器(8 个)，段寄存器(4 个)，控制寄存器(指令指针 IP 和标志字寄存器 FR)。

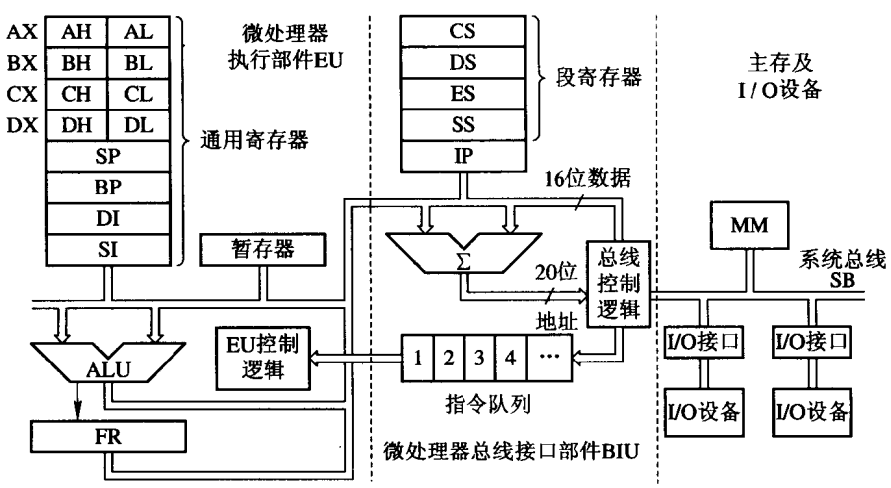


图 2.1　8088/8086 微型机的组织结构

此为试读，需要完整PDF请访问：www.ertongbook.com