

IBM PC 8086/8088

宏汇编语言程序设计及实验

谭云福 编著

机械工业出版社

宏汇编语言程序设计及实验

机械工业出版社

313

VF/1

IBM PC 8086/8088

宏汇编语言程序设计及实验

谭云福 编著

机械工业出版社

(京)新登字054号

内 容 简 介

本书是 IBM PC (Intel 8086/8088) 微型机宏汇编语言程序设计与实验结合性教材。全书共9章, 分别介绍了指令功能、程序设计基本方法和技巧、宏汇编及条件汇编程序设计、DOS 功能调用及应用、中断程序及常驻内存程序设计、图形与音响程序设计、DEBUG命令和DOS命令。为便于应用, 各章均列出大量程序设计实例。

本书可作为大中专院校计算机和自动化类专业教学用书或参考书, 亦可作为工程技术人员或微机工作者自学用书。

**IBM PC 8086/8088
宏汇编语言程序设计及实验**

谭云福 编 著

*

责任编辑: 严蕊琪

*

机械工业出版社出版(北京阜成门外百万庄南街一号)

(北京市书刊出版业营业许可证出字第117号)

北京通县建新印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 787×1092 1/16 · 印张 16 · 字数 385 千字

1993年3月北京第1版 · 1993年3月北京第1次印刷

印数 0001—3 500 册 · 定价: 11.80 元

*

ISBN 7-111-03581-X/TP·176

前 言

汇编语言是面向计算机硬件的指令语言，它在微机工业控制、自动检测、无线电通讯、智能仪器仪表等领域应用十分广泛。然而，汇编语言比较抽象，学习汇编语言应紧密结合实验才能达到良好效果，因此编写了这本《IBM PC宏汇编语言程序设计及实验》教材。该书是在本科生教学的基础上编写的，适用于IBM-PC、XT、286、386及其兼容机。该书主要特点是，紧密结合实验讲述了汇编语言程序设计及调试过程，书中给出的所有程序均经过上机验证。考虑到不同用户和工程技术人员自学的需要，书中还给出了大量的程序设计实例，读者只要有上机条件，都可以很快掌握汇编语言程序设计方法，并能直接应用于实际工作。

全书共分九章：第1章是学习汇编语言必要的硬件基础；第2章是指令系统及指令实验；第3章是汇编语言程序设计基本方法，讲述了子程序设计、代码转换、浮点数运算、表的处理等程序设计技巧；第4章是宏汇编及条件汇编程序设计，讲述了分模块程序设计及连接方法；第5章是DOS功能调用格式及应用实例；第6章是中断程序设计及常驻内存程序设计；第7章是图形程序设计和音响程序设计；第8章是DEBUG命令及应用；第9章是DOS组成及常用DOS命令描述。

本书在编写过程中，得到了胡笔蕊教授、赵树春副教授的热情指导，并对原稿提出了修改意见，在此深表谢意。全书由赵振玉副教授主审。

因编写时间紧迫，且编者水平有限，缺点和不妥之处在所难免，敬请读者批评指正。

编 者

目 录

第1章 微机基础	1
1 微机系统组成	1
1.1 微机系统基本结构	1
1.2 微机系统硬件配置	2
2 中央处理机 (CPU) 与可编程寄存器	4
2.1 8086/8088CPU 功能结构	4
2.2 8086/8088 可编程寄存器	4
3 存储器分段结构及堆栈操作	6
3.1 存储器分段结构	6
3.2 堆栈操作	6
4 数据类型及字符编码	7
4.1 数据类型	7
4.2 数的进位制	8
4.3 字符编码	9
第2章 指令系统	10
1 指令寻址方式	10
2 标志寄存器 (FLAG)	12
3 指令系统	13
3.1 数据传送指令	14
3.2 算术运算指令	15
3.3 逻辑运算指令	19
3.4 移位指令	20
3.5 串操作指令	21
3.6 转移控制指令	22
3.7 处理器控制指令	24
4 检验指令功能	25
4.1 分析指令	25
4.2 实验	26
4.2.1 实验准备	26
4.2.2 启动微机系统和DEBUG 程序	28
4.2.3 输入并调试程序	28
5 数据传送指令程序设计	32
5.1 建立内存数据块	32
5.2 用传送指令完成数据块传送	34
5.3 用串传送指令完成数据块传送	35
6 串搜索和串比较指令程序设计	36

6.1 建立字符区并搜索关键字	36
6.2 用串搜索和串比较指令搜索关键字	38
7 加减法指令程序设计	39
7.1 多字节加法运算	39
7.2 多字节减法运算	41
8 乘除及移位指令程序设计	42
8.1 乘除指令编程	42
8.2 移位指令编程	45
9 子调用指令程序设计	45
9.1 显示十进制数	45
9.2 输出字符图形	46
第3章 汇编语言程序设计	49
1 汇编语言语句及汇编运算符	49
1.1 汇编语言语句格式	49
1.1.1 指令语句格式	49
1.1.2 伪指令语句格式	50
1.2 汇编运算符	50
2 伪指令	52
2.1 符号定义伪指令(赋值语句)	52
2.2 内存数据定义伪指令	52
2.3 段定义伪指令	53
2.4 段寄存器说明伪指令	54
2.5 过程(子程序)定义伪指令	54
2.6 模块开始伪指令	55
2.7 模块结束伪指令	55
2.8 定位伪指令	55
2.9 列表伪指令	55
2.10 系统隐含进位制伪指令	55
2.11 连接伪指令	55
2.12 记录伪指令	56
2.13 结构伪指令	57
2.14 块注释伪指令	57
3 程序设计一般步骤	57
4 程序结构基本形式	59
4.1 顺序结构	59
4.2 分支结构	60
4.3 循环结构	60
4.4 子程序结构	61
5 程序编辑、汇编及连接	62
5.1 汇编语言程序实例	62
5.2 程序编辑、汇编及连接实验	64
6 伪指令和汇编运算符编程练习	72

6.1 伪指令和汇编运算符检验性编程	72
6.2 程序实验	73
7 子程序设计	74
7.1 子程序近程直接调用程序设计	74
7.2 子程序近程间接调用程序设计	76
7.3 程序实验	77
8 代码转换程序设计	81
8.1 多字节二进制数对BCD码转换	81
8.2 多位BCD码对十进制数ASCII码转换	81
8.3 十进制数ASCH码对二进制数转换	83
8.4 大小写字母互换	86
9 浮点数运算	88
9.1 浮点数在存储器中存储格式	88
9.2 浮点数加减运算	89
9.3 浮点数乘法运算	91
9.4 程序实验	93
10 表的处理	94
10.1 读表程序设计	95
10.2 表删除、插入程序设计	95
10.3 排序程序设计	95
11 EXE文件和COM文件	102
11.1 程序段前缀	102
11.2 EXE文件	103
11.3 COM文件	104
11.4 EXE文件对COM文件转换	105
第4章 宏汇编语言程序设计	106
1 宏定义和宏调用	106
1.1 宏语句	106
1.2 宏调用编程练习	107
1.3 宏调用显示字符图形	107
1.4 程序实验	113
2 条件汇编程序设计	114
2.1 条件汇编语句	114
2.2 条件汇编程序设计	115
2.3 程序实验	115
3 分模块程序设计	118
3.1 分模块程序设计(一)	118
3.2 分模块程序设计(二)	121
3.3 程序实验	124
第5章 输入输出DOS功能调用	126
1 字符输入输出DOS功能调用	126
1.1 字符输入输出DOS功能调用格式	126

1.2 字符输入输出DOS功能调用应用.....	129
2 磁盘文件管理.....	131
2.1 磁盘文件管理DOS功能调用格式.....	131
2.2 磁盘文件管理程序设计.....	133
2.2.1 创建、查找和删除磁盘文件.....	133
2.2.2 输出磁盘文本文件.....	136
2.2.3 在磁盘上建立文件.....	137
3 取、置微机系统日期程序设计.....	139
3.1 取系统日期程序设计.....	139
3.2 置系统新日期程序设计.....	140
3.3 程序实验.....	144
4 取、置微机系统时间程序设计.....	145
4.1 取系统时间程序设计.....	145
4.2 置系统新时间程序设计.....	145
4.3 程序实验.....	147
第6章 中断程序设计.....	150
1 中断技术.....	150
1.1 中断概念.....	150
1.2 中断源优先级.....	150
1.3 中断向量.....	151
1.4 中断指令和HLT指令.....	152
2 中断程序设计.....	153
2.1 程序设计.....	153
2.2 程序实验.....	153
3 中断嵌套程序设计.....	157
4 常驻内存程序设计.....	160
第7章 图形与音响程序设计.....	164
1 图形显示基础.....	164
1.1 显示器.....	164
1.2 显示模式和显示属性.....	164
2 图形显示中断调用格式.....	165
3 图形程序设计.....	169
3.1 输出变色三角图形.....	169
3.2 输出带属性彩色字符.....	171
3.3 画圆程序设计.....	171
4 计算机发音基础.....	177
4.1 定时/计数器 8253.....	177
4.2 扬声器控制.....	178
4.3 音频计算方法.....	178
5 发音程序设计.....	179
5.1 简单发音程序设计.....	179
5.2 升降调发音程序设计.....	179

5.3 乐曲程序设计.....	182
5.3.1 《雪里送炭》乐曲程序设计.....	182
5.3.2 《一个美丽的传说》乐曲程序设计.....	183
第8章 DEBUG命令及应用	189
1 DEBUG功能.....	189
2 DEBUG启动.....	189
3 DEBUG命令应用明说.....	190
4 DEBUG命令及应用.....	192
4.1 A(Assemble) 汇编.....	192
4.2 U(Unassemble) 反汇编.....	193
4.3 D(Dump)显示存储器.....	193
4.4 E(Enter) 修改存储器	194
4.5 R(Register) 显示或修改寄存器	194
4.6 G(GO) 执行程序	195
4.7 T(Trace)跟踪命令.....	196
4.8 Q(Quit)退出DEBUG命令.....	196
4.9 N(Name)命名.....	196
4.10 L(Load)读磁盘.....	197
4.11 W(Write)写磁盘.....	198
4.12 I(Input Command) 输入命令.....	200
4.13 O(Output Command) 输出命令.....	200
4.14 H(Hexarithmetic) 运算.....	200
4.15 M(Move) 传送	201
4.16 F(Fill) 填充.....	201
4.17 S(Search)检索.....	202
4.18 C(Compare) 比较.....	202
4.19 P(Proceed) 进程.....	203
第9章 行编辑及常用 DOS 命令	204
1 行编辑EDLIN	204
1.1 A(Append Lines) 增补行.....	204
1.2 C(Copy Lines) 复制行.....	204
1.3 D>Delete Lines) 删除行.....	204
1.4 编辑行(Edit Line) 命令	205
1.5 E(End Edit)结束编辑.....	205
1.6 I(Insert Lines)插行.....	205
1.7 L(List Lines) 列行.....	205
1.8 M(Move)移动行.....	206
1.9 P(Page) 分页	206
1.10 Q(Quit Edit)退出编辑.....	206
1.11 R(Replace Text)替换文本.....	206
1.12 S(Search Text) 检索文本.....	206
1.13 T(Transfer Lines)传送行.....	207

1.14 W(Write Lines)写行.....	207
2 DOS概述.....	207
2.1 DOS发展进程.....	207
2.2 DOS组成.....	208
2.3 DOS命令类型.....	208
2.4 准备软磁盘.....	209
2.5 启动DOS.....	209
2.6 DOS常用键及编辑键.....	210
3 文件说明.....	211
3.1 文件标识符.....	211
3.2 DOS通配(? 、 *)	211
3.3 DOS保留设备名.....	212
3.3 文件分类.....	212
4 常用DOS命令描述.....	212
附录1 DEBUG命令	223
附录2 EDLIN命令.....	225
附录3 DOS常用命令	226
附录4 8086/8088指令系统总表.....	231
附录5 ASCII 码字符表.....	245
参考文献.....	246

第1章 微机基础

1 微机系统组成

1.1 微机系统基本结构

微机系统的基本结构由中央处理机 (CPU)、只读存储器 (ROM)、随机读/写存储器 (RAM)、外部接口电路 (I/O) 以及系统总线等组成, 如图1-1所示。

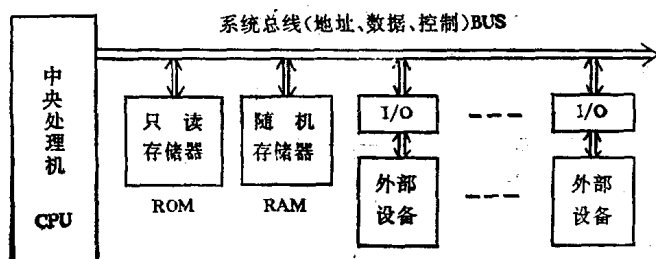


图1-1 微机系统基本结构

1. 中央处理机 (CPU)

中央处理机是一片集成电路, 它是微机系统的核心部件, 其主要功能有指令译码、微操作信号生成、数值运算以及总线控制等。

2. 只读存储器 (ROM)

只读存储器由一片或多片集成电路组成。在ROM中固化了基本的输入输出设备驱动程序和微机启动自检等程序, 统称为 BIOS 系统程序 (BASIC Input/Output System)。BIOS程序是微机系统的重要组成部分, 它与磁盘操作系统 DOS (Disk Operating System) 构成了微机的“神经”系统。磁盘操作系统通过ROM中的设备驱动程序实现对基本输入输出设备的管理以及它们之间的信息传送。

在有些微机中, 还配置了固化BASIC语言解释系统, 在ROM中装有磁带BASIC解释程序, 机器通电后, 不需要DOS系统启动, 能自动进入BASIC状态, 并等待执行磁带 BASIC 应用程序。

在 ROM 中的程序是只读工作方式, 它只能被随机读出, 不能被重新写入, 因此, 在ROM中的程序是被保护的。

3. 随机读/写存储器 (RAM)

RAM (Random Access Memory) 存储器又称为内存储器 (内存), 它由多片集成电路组成, 存储器的容量以KB (1024字节) 为计算单位, 每个字节表示八位二进制信息, 一般微机的内存容量在512KB~1MB (1024KB) 之间。目前流行的各种牌号的286或386微机内存容量在2~4MB左右, 销售商根据用户的需要可以扩充。

RAM存储器用于存储程序和数据, 任何要执行的程序和要处理的数据 (如磁盘程序或数据文件) 必须先装入RAM才能工作。

4. I/O接口电路

I/O 接口电路由多种集成电路芯片、电阻、电容等器件组成, 它是主机与外设、外设与外设之间的硬件接口, 不同的外部设备, 可以通过配套的接口电路实现它们之间或它们与CPU之间的数据缓冲与传送、以及信号转换等。

I/O 接口电路应用十分广泛, 但就其接口电路的产品而言, 是针对某类设备而设计的,

且装配在印刷电路板上，在应用时，将其插在微机主机箱内的I/O总线扩展槽中，再用软扁连线把接口板与相应外设连接起来，实现不同外设与主机的接口。

5. 总线 (BUS)

总线是一族公共数据线、地址线和控制信号线，它把系统配置的各个设备及部件连接起来，构成微机的硬件系统。总线在工作时，数据及各种信息传送是分时操作的。

1.2 微机系统硬件配置

微机系统硬件配置如图1-2所示。

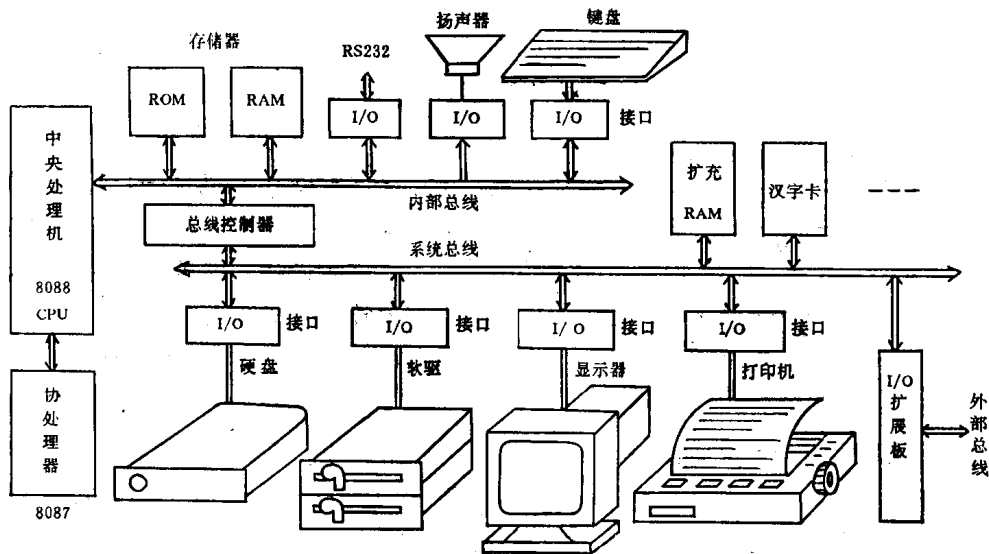


图1-2 微机系统硬件配置

1. 内部总线

内部总线又称为板内总线，它把CPU、ROM、RAM、基本I/O接口、定时器以及总线控制逻辑等部件连成一个系统，成为微机的基本硬件系统板电路。内部总线是微机系统的一级总线。

2. 系统总线

系统总线又称为板间总线，它由CPU通过总线控制逻辑进行管理，在该总线上装有5~8个通用I/O扩展插槽，不同设备的接口电路都将通过该扩展槽与主机连接。系统总线是微机系统的二级总线。

3. 协处理器

中央处理器CPU的运算是定点运算，若要实现浮点数运算，以及某些函数运算，都需要进行仿真编程，因此，运算速度比较慢。协处理器是专用于浮点数及某些函数运算的器件，它是一片集成电路。协处理器扩充了CPU寄存器和指令系统，并增加了新的数据类型。对某些科学计算（如常用函数、实数四则运算等）可直接由硬件实现，因此，协处理器大大提高了计算机的数值处理能力和运算速度。

4. 磁盘存储器

磁盘存储器是计算机的外部存储器（外存），它是对内存（内存）的补充。磁盘存储器是机电一体化设备，它由电机驱动、磁盘以及磁头等部件组成，称为磁盘驱动器。它的

特点是靠磁盘表面存储信息，由读/写磁头进行读/写操作。磁盘存储器受到机械传动的影响，随机启动速度不是很快，但在启动后每次数据的读/写速度较快。

磁盘存储器是利用磁滞现象存储信息，因此在停电后，它所存储的信息不会丢掉。磁盘存储器根据装配的结构不同又可分为硬磁盘存储器和软磁盘存储器。

硬盘存储器有密封的外壳，内装传动机构、读/写磁头（多个）和硬盘片（一片或多片），驱动电路在密封壳外边。硬盘存储器的容量在10MB到几百MB之间（硬磁盘的盘片不准用户拆卸）。

软盘存储器一般没有密封的外壳，其驱动和读/写原理与硬磁盘有些类似，最大的区别是软磁盘片可以随意更换，在应用时可以无限地备份程序和数据文件。在应用软盘存储器时要注意两个问题：一是软盘片的容量，常见的有360KB、720KB和1.2MB等，一般把1.2MB的磁盘片称为高密磁盘，而把360KB的磁盘称为低密磁盘，相应的磁盘驱动器，也称为高密驱动器和低密驱动器。目前流行的微机一般都配有一个高密驱动器和一个低密驱动器，以便于高低密磁盘上的文件相互交换；二是就磁盘驱动器和盘片的尺寸而言，有3in、5in不等，有些机器同时配有5in和3in两种磁盘驱动器。^①

5. 外部设备

(1) 扬声器

扬声器是输出设备，它用于机器报警和输出音响。扬声器与定时/计数器、控制门以及驱动电路相连，可由用户编程控制音响效果。

(2) 键盘

键盘是输入设备，用于人机对话。键盘可以向计算机输入数据和程序，或向计算机发布命令。

(3) 显示器

显示器是输出设备，用于人机对话。在计算机中的程序和数据以及计算机的运行结果（如数值运算结果、绘图及制表等），都可以通过显示器显示出来。目前，新技术产品不断出现，显示器也可以用作输入设备使用，如通过光笔在显示器上绘图或作其它信息的输入。

(4) 打印机

打印机是输出设备，它能够把源程序或数据以及程序运行结果（如数值运算结果、绘图、制表等）打印出来。根据打印的方法，目前常见的有针式打印机和激光打印机等几种类型。

6. 外部总线

外部总线用于机外的进一步扩展。例如，有些自制的接口电路机内放不下，或系统总线的I/O扩展插槽数量不够，就可通过该总线进行连接。

7. 汉字卡

汉字卡为汉字操作系统提供汉字字型（汉字库）。汉卡的类型比较多，有些汉卡除固化了汉字库以外，还固化了某些系统程序和应用程序，这样系统工作便可提高运行速度，并能预防计算机病毒对这些程序的侵入。

^① 1in=0.0254m。

2 中央处理机 (CPU) 与可编程寄存器

2.1 8086/8088CPU功能结构

CPU功能结构如图1-3所示。

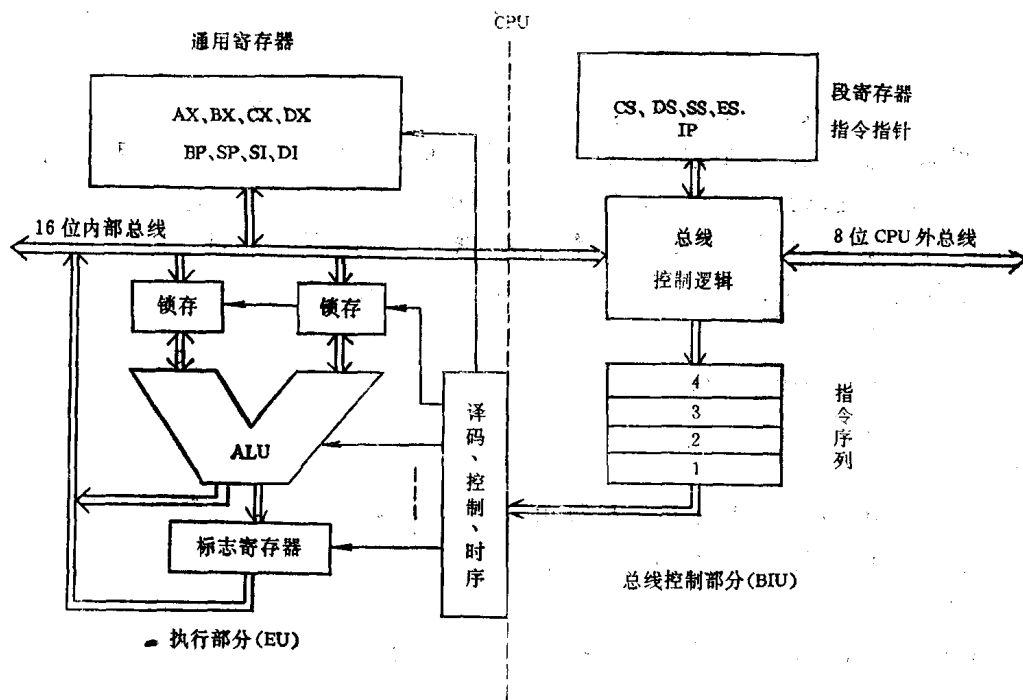


图1-3 8086/8088CPU功能结构

1. 执行部分 (EU)

执行部分由运算器 (ALU)、通用寄存器、标志寄存器、暂存器 (锁存) 及控制逻辑等部分组成。运算器完成算术及逻辑运算，通用寄存器用于存储数据以及设置偏移地址，标志寄存器存放结果特征，控制逻辑电路用来生成各种微操作信号去控制各种操作。

2. 总线控制部分 (BIU)

总线控制部分主要完成对CPU外总线的管理，CPU通过总线控制逻辑实现CPU与存储器以及外部I/O接口电路之间的数据传送。

微机在工作时，首先从存储器中读取指令，经CPU的外总线送入指令排队寄存器，指令经控制逻辑电路进行译码，并与内部的时序电路组合生成相应的微操作信号去完成指令规定的操作。若从存储器中取出的是数据，则送入运算器或寄存器。

指令排队寄存器用于取指令的缓冲，因而实现了取指令和执行指令并行处理，提高了系统的运行速度。

2.2 8086/8088可编程寄存器

可编程寄存器可作为指令的操作数，8086/8088可编程寄存器符号及名称如图1-4所示。

1. 通用寄存器

通用寄存器分为数据寄存器和指针寄存器，数据寄存器包括AX、BX、CX和DX，这些寄存器可以按16位使用，也可以按8位使用，它们包括AH、AL、BH、BL、CH、CL、DH、

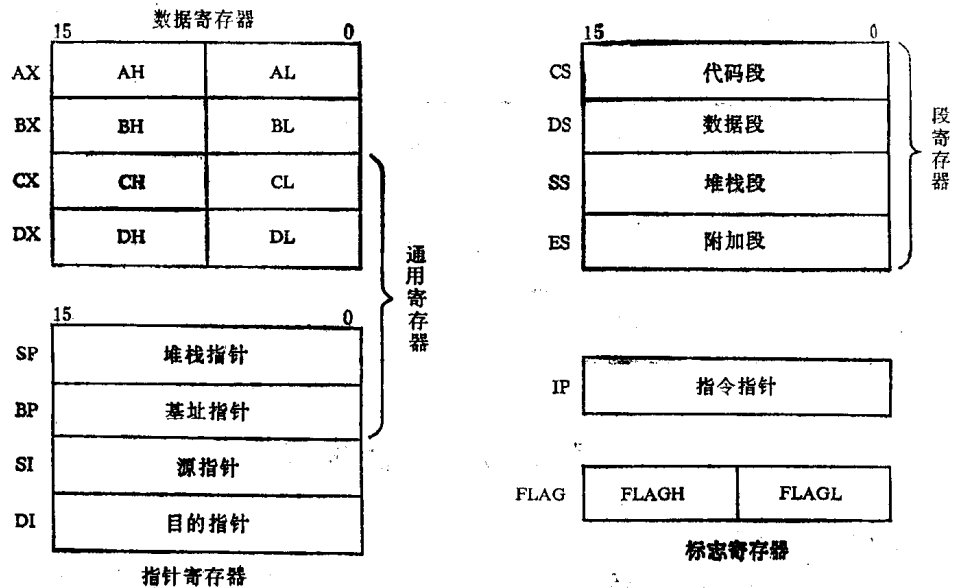


图1-4 8086/8088可编程寄存器

DL。指针寄存器包括SP、BP、SI和DI，它们是16位寄存器。指针寄存器可以作为数据寄存器使用，但多数情况下作为存储器寻址指针使用。SP专用于堆栈操作地址指针。

BX、BP、SI和DI寄存器可用于存储器间接寻址。

2. 段寄存器 (CS、DS、SS、ES)

在对存储器寻址时，16位指针寄存器或16位直接偏移地址寻址范围是64KB，段寄存器（16位）是专为扩大存储器寻址范围而设置的，它们与指针寄存器或直接偏移地址组合，形成较大的物理地址。例如，IBM PC微机形成的20位物理地址，可在1MB的存储空间寻址，1MB的内存空间分成16个（每个64KB）逻辑段，各段由段寄存器设定。

计算机在工作时，指令寻址总是在当前段内进行，当前段又称为活动段，它们是由代码段CS、数据段DS、堆栈段SS以及附加段ES指定的，当要改变某个活动段时，只要修改这些段寄存器的值即可。当然，有时也可以在指令操作数前加段超越前缀（如DS:、ES:、CS:、SS:）来实现对其它段的访问。

3. 指令指针寄存器 (IP)

指令指针寄存器又称为指令地址计数器或程序计数器，它被规定与CS段寄存器组合对指令寻址。CS:IP总是指向将要执行的指令地址，CPU每执行一条指令，IP被自动修改并指向下一条要执行的指令偏移地址。

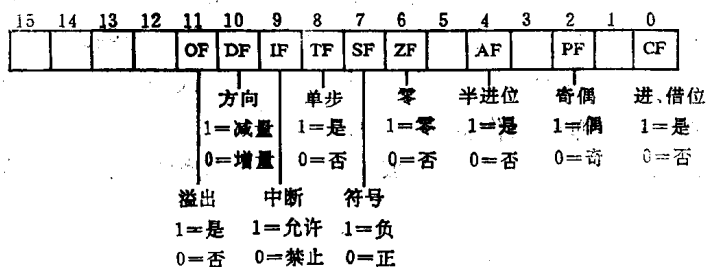


图1-5 标志寄存器 (FLAG)

4. 标志寄存器 (FLAG)

标志寄存器长两个字节，有效位是9位，它记录着某些指令的运行结果特征，标志寄存器为程序分支和转移提供了判断条件。标志寄存器的标志位符号和含意如图1-5所示。

在用DEBUG软件调试程序时，检查标志寄存器的状态，其标志位是

用一些符号表示的，见表1-1所列。

表1-1 标志寄存器标志位符号

状态	OF 溢出	DF 串方向	IF 中断	SF 符号	ZF 结果零	AF 半进位	PF 奇偶	CF 进位
0	NV否	UP增	DI禁	PL正	NZ否	NA否	PO奇	NC否
1	OV是	DN减	EI允	NG负	ZR是	AC是	PE偶	CY是

3 存储器分段结构及堆栈操作

3.1 存储器分段结构

所谓存储器分段结构是指存储器在逻辑上分成若干段（64KB），各段地址由段寄存器指定，段内的偏移地址（低16位）由指针寄存器或指令中的直接偏移地址指定。存储器分段寻址如图1-6所示。

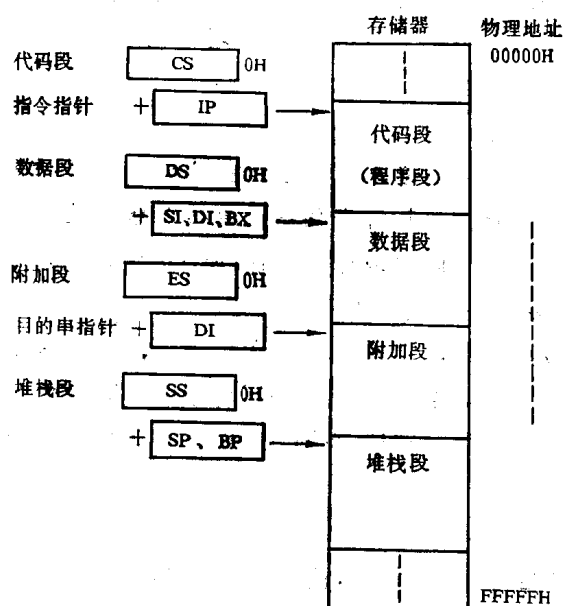


图1-6 存储器分段结构

在存储器寻址时，段寄存器（16位）内容向左移4位（二进制位），再与指针寄存器（如IP、SI、DI、BX、SP、BP）或直接偏移地址相加形成物理地址。在应用时，段与段之间可以重叠或部分重叠，但有用的信息不能被重叠。

3.2 堆栈操作

在存储器中，从某个固定地址开始，读/写的信息按后进先出的原则进行操作，这称为堆栈操作。进栈称为压入，出栈称为弹出，每次压入和弹出操作是两个字节，通常堆栈指令PUSH和POP就是用来完成这些操作的。

存储器的这种堆栈结构也是逻辑上的。栈区的起点（固定端）称为栈底，它可通过修改SS和SP寄存器内容重新选择栈区的起点。栈区活动一端称为栈顶，它是小地址一端。

堆栈操作总是要伴随SP指针自动修改，每次栈压入操作，SP减2；每次栈弹出操作，SP加2。SP内容总是指向栈顶位置。堆栈压入和弹出操作如图1-7所示。

除了堆栈操作指令外，子程序调用与返回、中断调用与返回也伴随堆栈的压入与弹出操作，详见以后有关章节。

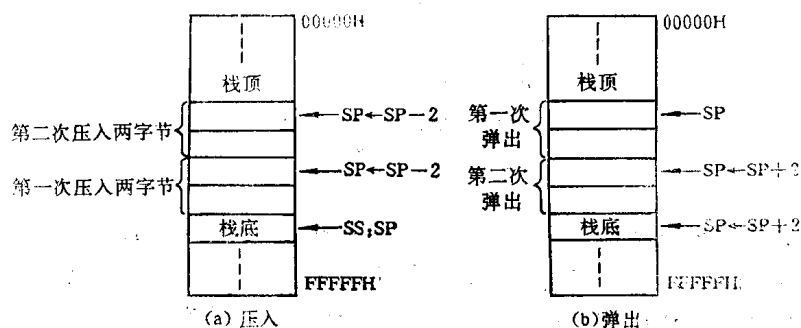


图1-7 堆栈压入与弹出操作示意图

4 数据类型及字符编码

4.1 数据类型

1. 无符号二进制数

8位无符号数 (1个字节), 取值范围0~255

16位无符号数 (2个字节), 取值范围0~65535

2. 有符号二进制数

8位有符号数 (1个字节), 取值范围-128~127

16位有符号数 (2个字节), 取值范围-32768~32767

有符号数的最高位是符号位, 0表示正, 1表示负。

3. 无符号十进制数 (BCD码)

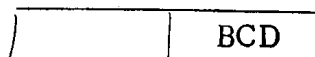
在计算机应用中, 常用4位二进制数来表示一位十进制数, 称为二-十进制数或BCD码。

用4位二进制代码表示十进制数有许多方法, 常用的8421码、5421码、余3码等, 每个BCD码占半字节。

BCD码在存储时分为非压缩式和压缩式两种。

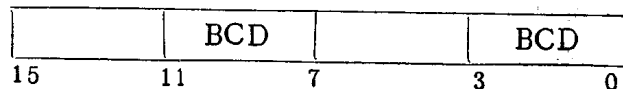
非压缩十进制数:

一字节:



数值范围0~9

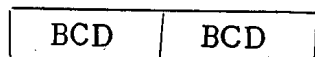
二字节:



数值范围0~99

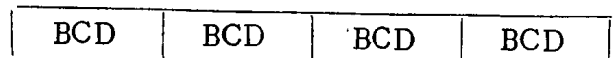
压缩十进制数:

一字节:



数值范围0~99

二字节:



数值范围0~9999

4. 浮点数

不同的汇编程序对浮点数约定可能有些不同, 但其基本结构是类同的, 它们都是由阶码和尾数两部分组成。例如, 8086/8088宏汇编程序对用DD伪指令定义的浮点数约定如下: