

内容提要

本书主要介绍 8086/8088 和 Pentium 微型计算机的汇编语言，内容包括指令系统、源程序结构、输入输出程序设计、过程设计、模块化程序设计等。书中配有大量例题和插图，以方便学生理解基本概念和掌握基本技巧。

本书可作为大、中专院校计算机及其相关专业的教材，也可供软件开发与应用人员参考。

计算机系列教材 汇编语言程序设计

- ◆ 编 著 许曰滨 李 潼
责任编辑 潘 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
北京汉魂图文设计有限公司制作
北京朝阳隆昌印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本：787×1092 1/16
印张：13.5
字数：331 千字 2000 年 7 月第 1 版
印数：1-6 000 册 2000 年 7 月北京第 1 次印刷

ISBN 7-115-08211-1/TP?1385

定价：19.8 元

前 言

随着计算机的不断普及，用于描述计算机工作流程的语言也得到迅速发展。在种类繁多的计算机语言中，高级语言因其语法规则接近于自然语言而得到广泛应用。但是，高级语言对硬件的“透明度”差，使一些依赖硬件较强的程序的开发受到严重制约。很自然地，人们会想到计算机的低级语言——汇编语言。

汇编语言是面向计算机硬件的语言。程序设计者可以随心所欲地控制 CPU 进行各种运算、访问主存储器的任何一个存储单元、从外部设备上输入或输出数据……因而，汇编语言是一种功能很强的计算机语言。

人们将汇编语言归入低级语言，部分原因是它的语法规则太繁琐，远离于自然语言。许多程序设计人员为其严谨的语法所困扰。然而，学生们学好、用好汇编语言，必将为其日后开发计算机软件打下良好的基础。

在本书编撰过程中，为了避免枯燥地、连篇累牍地介绍语法，许多内容都以程序设计为主线，围绕应用示例进行讲解，学生在完成前两章的学习后就可以上机操作。这样做的目的是，尽快增加学生上机调试程序的感性认识，提高其学习汇编语言的兴趣。

本书第 1 章介绍计算机硬件和系统软件的一些基本知识；第 2 章引入汇编语言的段式结构，以便使学生掌握程序设计的整体框架；第 3 章至第 5 章讲述键盘输入程序、屏幕显示程序和数据处理程序的设计方法；第 6 章介绍磁盘文件读写方法。我们相信，通过前 6 章的学习，学生的汇编语言设计思路就可以基本建立起来。第 7 章对多模块程序设计方法、外部设备接口中的端口地址等较深的问题进行了进一步的讨论。

本书既注意了内容的深入浅出，同时又注意到知识介绍的系统性。书中带有“*”的章节属于选学内容，不同专业的学生可根据本专业特点适当取舍，即使不学也不影响对基本概念的掌握。

本书承蒙张昆藏、李桂芬两位教授指点，并得到李咏梅、刘伟两位同志精心审阅，谨此表示深深的感谢。

书中若有疏漏之处，敬请读者批评指正。

作 者
2000 年 5 月

序 言

为了适应“逐步实现教材多样化，增加不同品种、不同档次、不同风格、不同改革实验的教材”要求，我们组织编写了这套《计算机系列教材》，以适应大、中专计算机教学的需要。

本套教材的基本任务是系统地阐述计算机的基本概念和基本操作，这些基本概念和基本操作将是未来掌握计算机知识的基础。因此，本套教材的构架并不打算定位在不断变化的、十分活跃的研究和应用领域，而是立足于对基础知识的介绍上。本套教材共包括以下 10 本，计划 1999 年出版前 5 本，2000 年出齐。

1. 《微型计算机应用基础》
2. 《中文 Windows 98 实用教程》
3. 《操作系统概论》
4. 《C 语言程序设计》
5. 《PASCAL 语言程序设计》
6. 《数据结构》
7. 《计算机实用软件》
8. 《计算机网络基础》
9. 《微机系统原理与维修》
10. 《汇编语言程序设计》

本套教材根据计算机技术的最新发展，在取材的深度和广度方面作了精心的优化选择，从而使其在新颖性、系统性和准确性方面达到了新的高度。在写作方面，力求深入浅出、通俗易懂，并特别注重实例的选择和说明。为了加深对基本概念的理解，各章的末尾均给出大量习题供学员课外练习。同时，每本教材的后面都附有实验题配合学员上机实习使用。

由于编写时间紧促，而且限于水平和经验的不足，书中肯定存在不少错误和遗漏，我们诚心希望使用本套教材的广大教师和同学们提出宝贵的批评建议。

教材编委会

1999 年 6 月

编委会名单

主任：王熙法（中国科学技术大学计算机系主任，教授）

委员：陆钟辉（北京大学计算机系教授）

师书恩（北京师范大学计算机系教授）

杨一平（首都经济贸易大学信息管理系教授）

许曰滨（青岛大学计算机系教授）

沈长宁（北京师范大学电子学系副教授）

沈精虎（青岛大学副教授）

于久威（北京师范大学物理学系副教授）

目 录

第 1 章 汇编语言运行环境	1
1.1 计算机系统组成	1
1.1.1 计算机的组成结构	1
1.1.2 中央处理器 (CPU)	2
1.1.3 内存储器	3
1.1.4 输入设备	4
1.1.5 输出设备	4
1.2 微处理器	5
1.2.1 8086/8088 寄存器结构	5
1.2.2 8086/8088 的地址计算	9
1.2.3 堆栈地址生成	9
1.3 计算机内的数据表示	11
1.3.1 二进制数	11
1.3.2 十六进制数	14
1.3.3 补码	15
1.4 微型计算机软件	16
1.4.1 操作系统	16
1.4.2 文本编辑程序 EDIT	19
1.4.3 汇编程序与连接程序	21
习 题	21
第 2 章 程序基本结构	25
2.1 汇编语言源程序格式	25
2.1.1 程序头和程序尾	26
2.1.2 程序体	27
2.2 指令概况	30
2.2.1 操作指令的格式与分类	31
2.2.2 数据传送指令	32
2.2.3 处理器控制指令	33
2.3 一个小程序	34
2.3.1 “计算圆周长”的程序	35
2.3.2 程序注释	37
2.3.3 指令标号	38
2.3.4 几种常用的寻址方式	38
2.4 程序上机的运行与调试	39

2.4.1 源程序编辑.....	39
2.4.2 程序汇编.....	40
2.4.3 程序连接.....	41
2.4.4 程序运行与调试.....	42
习 题.....	46
第 3 章 键盘输入程序设计.....	49
3.1 一位数输入程序设计.....	49
3.1.1 ASCII 码.....	49
3.1.2 一位数输入程序设计举例.....	50
3.1.3 算术运算指令.....	53
3.2 多位数输入程序设计.....	57
3.2.1 多位数的输入举例.....	57
3.2.2 转移指令.....	59
3.3 多字符输入程序设计.....	60
3.3.1 多字符输入举例.....	60
3.3.2 标识符与变量.....	62
3.3.3 几种特殊的寻址方式.....	64
3.4 字符串输入程序设计.....	66
3.4.1 定义输入缓冲区.....	66
3.4.2 字符串输入程序设计举例.....	67
习 题.....	69
第 4 章 屏幕输出程序设计.....	75
4.1 BCD 码与二进制数.....	75
4.1.1 BCD 码.....	75
4.1.2 将二进制数转换为 BCD 码.....	76
4.1.3 数据显示程序设计.....	79
4.2 字符串输出程序设计及调试.....	83
4.2.1 显示字符串程序设计举例.....	83
4.2.2 用 Debug 查看存储分配情况.....	85
4.3 系统调用.....	86
4.3.1 系统调用的分类.....	87
4.3.2 01H~0CH 号功能.....	88
4.3.3 系统调用举例.....	89
4.4 顺序、分支与循环.....	92
4.4.1 顺序结构.....	92
4.4.2 分支结构.....	93
4.4.3 循环结构.....	93

4.4.4 循环指令	94
习 题	97
第 5 章 数据处理	101
5.1 位操作	101
5.1.1 移位指令	101
5.1.2 利用逻辑尺的程序设计	103
5.2 逻辑运算	105
5.2.1 逻辑指令的格式与功能	105
5.2.2 应用举例	107
5.3 过程	109
5.3.1 过程结构	110
5.3.2 调用指令	112
5.3.3 堆栈	112
5.3.4 应用举例——求随机数序列	114
5.4 数据处理举例	118
5.4.1 求一维数组平均值	118
5.4.2 求最大元	120
5.4.3 汽泡法排序问题	121
5.4.4 跨越段的过程调用举例*	123
习 题	125
第 6 章 磁盘文件	129
6.1 磁盘文件概述	129
6.1.1 文件概念	129
6.1.2 磁盘文件访问方式	131
6.2 利用文件句柄进行磁盘文件管理	132
6.2.1 创建新文件	133
6.2.2 打开文件与关闭文件	134
6.2.3 磁盘文件写入与读出	135
6.3 文件修改	141
6.3.1 文件读写指针	141
6.3.2 在文件中添加记录	144
6.4 错误码处理	148
习 题	150
第 7 章 高级应用*	153
7.1 多模块程序设计	153
7.1.1 段的组合	153

7.1.2 交叉引用	156
7.1.3 应用示例	158
7.2 端口地址与 I/O 指令	161
7.3 音乐程序设计	165
7.3.1 乐曲的频率与节拍	165
7.3.2 音符的频率因子	165
7.3.3 乐曲节拍的计算	166
7.3.4 音乐演奏	167
7.4 串行通信	172
7.4.1 串行通信原理	172
7.4.2 利用 BIOS 的串行接口访问	174
7.5 Pentium 处理器及汇编语言简介	179
7.5.1 Pentium 的寄存器结构	180
7.5.2 Pentium 的寻址方式	181
7.5.3 Pentium 的指令系统	182
7.5.4 Pentium 支持的源程序结构	184
习 题	186
实验指导书	191
实验一 程序编译实验	191
实验二 调试程序 DEBUG 实验	194
实验三 源程序结构实验	195
实验四 排序	197
实验五 磁盘文件访问	199
实验六 乐曲演奏	201
附录 汇编程序出错信息一览表	203

第1章 汇编语言运行环境

自从 20 世纪第一台电子计算机诞生以来，计算机技术飞速发展。特别是从电子管到晶体管，从集成电路到超大规模集成电路的研究与应用，使电子计算机的性能大幅度提高，其体系结构（Computer Architecture）也得到了重大改革。

本章针对微型计算机的体系结构，简要介绍汇编语言的运行环境，包括硬件结构与功能，软件配置及使用方法。

1.1 计算机系统组成

众所周知，电子计算机经历了电子管、晶体管、集成电路、大规模集成电路和超大规模集成电路等多个发展时期，其体系结构日益复杂。从工作方式上看，目前可供使用的大部分电子计算机，基本上遵循着著名数学家冯·诺依曼提出的所谓“程序驻留内存”（Stored Program）方式。也就是说，顺序地取出预先存放在内存中的程序，按照程序的要求控制计算机各部件进行操作。

1.1.1 计算机的组成结构

电子计算机通常由运算器、控制器、内存储器、I/O 控制装置、外部设备（输入和输出设备）组成。图 1-1 所示是计算机的基本结构。

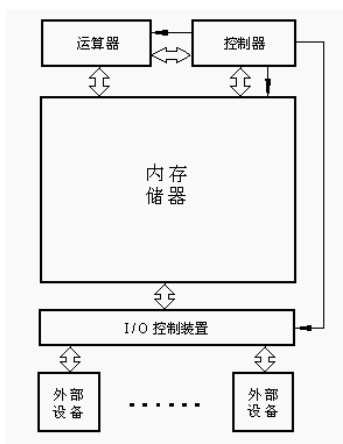


图 1-1 计算机基本结构

其中，运算器和控制器组合起来又称作“中央处理器”(Central Processing Unit)，简称 CPU。应用程序就是在 CPU 中运行的。

计算机的工作过程是：

- (1) 将用户编制的源程序通过输入设备读入计算机的内存储器中。
- (2) 对内存储器中的源程序进行必要的翻译或者解释，使之成为可执行的程序。
- (3) 执行内存储器中的可执行程序，实现用户的要求。比如，用户程序要求计算机完成数据运算的处理步骤可以是：
 - 从输入设备读入初始数据，存储于内存存储器的指定区域。
 - 将内存存储器的初始数据送到 CPU 中，让运算器进行运算，并将运算结果送到内存存储器中保存。
 - 将内存存储器中所存的结果通过输出设备输出。

1.1.2 中央处理器 (CPU)

中央处理器是计算机的核心，由极其复杂的电子线路和大量电器元件组成。它的任务是实现各种运算，并产生各种控制信号，控制计算机各个部件有条不紊地工作。

1. 运算功能

电子计算机的功能十分强大，可以完成十分复杂的数学运算。但这些复杂的数学运算，都是建立在简单的算术运算、逻辑运算和关系运算的基础之上的。

(1) 算术运算

中央处理器可以完成两个量的加、减、乘、除和乘方等操作。实现的过程是，将两个运算分量送入运算器进行运算，运算结束后将结果送入内存中的某一指定位置保存起来。

(2) 逻辑运算

中央处理器可以完成两个逻辑分量的“逻辑与”、“逻辑或”运算以及单个逻辑分量的“逻辑非”运算，所得结果为 1 或者 0，分别表示“真”或“假”。

(3) 关系运算

中央处理器可以比较两个运算分量的大小，相应的关系运算有：大于 ($>$)、大于等于 (\geq)、小于 ($<$)、小于等于 (\leq)、等于 ($=$) 和不等 (\neq)。运算结果同样为 1 或者 0，分别表示“真”或“假”。

2. 控制功能

计算机的各个部件都有独立的功能，它们分别在系统中扮演着不同的角色。只有在中央处理器的统一指挥下，整个系统才能协调一致地工作。中央处理器中负责指挥工作的是控制器，它所实现的功能包括读取指令和执行指令。

(1) 读取指令

我们知道，指令是用户对计算机提出的要求，是下达给计算机的执行命令。用户的

所有意图都通过一系列指令体现出来。用户事先将完整的指令序列存储于内存储器中，供计算机中的控制器执行。

控制器中设有存放指令的寄存器，用来保存从内存储器中读出的指令。此外，控制器还设有“地址生成”功能部分，专门产生指令地址以读取内存储器中的指令。通常，指令在内存储器中是按执行的先后次序存放的，因此，控制器在读出一条指令之后可以按照“地址递增”方式极容易地产生出下一条指令的地址。不过，有时侯情况也有例外，当指令的执行次序违反了其存储次序时，就不能用“地址递增”方式产生地址了，这种情况下需要进行“地址计算”来产生。

通常，我们将地址生成过程称作“寻址”。

(2) 执行指令

指令的执行是一个信号译码的过程。当读出的指令被存到指令寄存器之后，通过一系列的、与指令寄存器输出端连接的信号译码电路产生不同的状态，在时钟的配合下输出一系列控制信号。

控制器产生的控制信号被送到计算机的各个部件中，指挥相关部件的操作。

1.1.3 内存储器

内存储器 (Memory)，简称内存，是用半导体电路制成的一种数据存储设备。它通过连接到 CPU 的一组“数据线”与 CPU 进行数据交换。另外，它还有与 CPU 连接的一组“地址线”和一组“控制线”，分别接收 CPU 送来的地址码和控制信号，供内存储器进行读/写数据时使用。

内存储器由存储体和辅助电路组成。其中，存储体中包含若干存储单元。每个存储单元可存放一个字节信息（即 8 位二进制数）。内存中数据的存入和读出都是以存储单元为基本单位进行的，每个存储单元有一个编号，叫做“内存地址”，简称“地址” (Address)。

一个内存单元，作为基本的存储单位，可以存储一个独立的数据。若干存储单元联合起来，可以存储大的程序、数据块及其他信息。为了实现数据的存入或读出，内存储器除了要有一个存储体外，还应设有配套的辅助电路。比如地址寄存器、数据寄存器和读写控制电路等。

图 1-2 所示是内存储器的逻辑图。

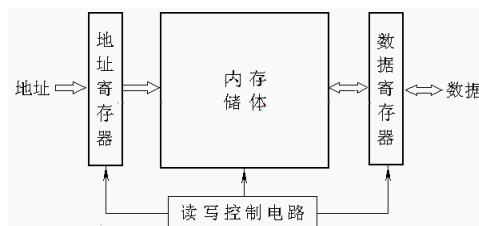


图 1-2 内存储器的组成

其中，读写控制电路包括读写内存储体的控制逻辑、内存地址译码电路、读/写放大电路，以及驱动逻辑等。

1.1.4 输入设备

计算机上用于进行数据输入输出的设备称作“外部设备”(Peripheral)，它们通过一些接口部件与计算机连接起来。由于硬件采取模块式结构，设备与计算机的连接可以方便地拆卸或安装，使系统的配置具有极大的灵活性。

输入设备是专门为计算机提供信息的。常用的输入设备是键盘和鼠标。

1. 键盘

键盘是最普通的输入设备，它可以输入的信息有：阿拉伯数字、英语字母、数学符号、汉字、标点符号及一些具有特殊含义的信息。它是目前输入设备中所能输入的信息的种类和数量最大的一种，因此几乎所有的计算机系统都能支持这种设备。

图 1-3 所示是一个普通键盘的平面图。

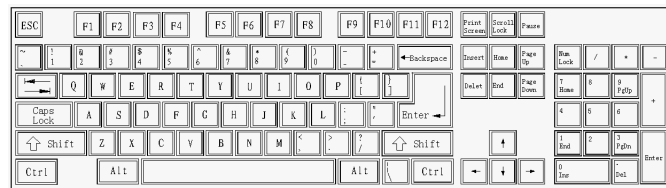


图 1-3 键盘

常用的标准键盘有 101 键、102 键和 104 键。另外还有带“跟踪球”或带“触摸板”的键盘。

2. 鼠标

鼠标是一种操纵光标移动的设备。它的工作原理是，当用户移动鼠标时，鼠标的移动距离及方向通过一连串的电脉冲信息传送给计算机，计算机收到这些信息后转化成一串平面坐标值来控制屏幕上的光标移动，从而达到操纵光标的目的。

通常，鼠标上设有左右两个按键，用户按下其中一个按键时，鼠标将自动把按键信号及鼠标的当前位置送给计算机，以表示用户确定了屏幕上当前光标处的一项选择。

1.1.5 输出设备

输出设备主要担负计算机的信息输出。输出的信息形式与键盘输入的信息大体相同，除了键盘包括的信息外，还包括一些输出控制信息，比如清屏幕、移动光标、控制走纸、换行和换页等。

常用的输出设备有显示器和打印机。

1. 显示器

计算机在运行过程中时常需要把运行状态和计算结果告诉用户，以便得到用户的响应。显示器即担负着这种显示任务。

常见的显示器有 35.6 cm (14 英寸)、38.1 cm (15 英寸) 和 43.2 cm (17 英寸) 等几种规格，可以显示多种字符，并支持多种颜色。

目前，一种功能奇特的显示器正流行起来，这就是“触摸屏显示器”。它的屏幕能对物体的接触作出反应，因此可将用户的触摸位置输入给计算机。计算机收到触摸信号后可由软件进行相应的处理。

2. 打印机

打印机是一种信息输出设备。它将输出的信息打印在纸上，作为硬拷贝可被用户保留起来。常见的打印机是点阵式的，十分耐用，但是打印速度比较慢，一般为每秒 200 个字符左右，噪声也很大。

近年来，激光打印机和喷墨打印机开始流行起来，它们的打印速度比点阵式打印机快得多，而且噪声也很小，受到用户普遍欢迎。

1.2 微处理器

微型计算机诞生于 20 世纪 70 年代，是大规模集成电路发展的产物。微型计算机一问世，就以体积小、重量轻、可靠性强、结构灵活等优点得到广泛的推广和应用，并逐步成为当今社会不可缺少的重要工具。

微型计算机的 CPU 又称作“微处理器”(Microprocessor)，通常采用超大规模集成电路制成。随着集成电路制造及其相关技术的不断进步，微处理器的性能也在不断提高，最具代表性的产品是美国 Intel 公司生产的微处理器系列。从 70 年代末开始，美国 Intel 公司先后研制出了微处理器 8086/8088、80286、80386、80486 和 Pentium 系列产品，其功能不断增强，内部结构也越来越复杂。正是由于这些性能各异的微处理器的支持，才使得微型计算机世界五彩缤纷。

然而，无论哪种档次的微处理器对汇编语言的支持都遵循着最基本的特征，即源程序框架、指令结构和寻址方式都是相同的。另外，由于档次高的微处理器汇编系统是档次低的汇编系统的延伸，且完全兼容档次低的，因此本书主要介绍最基本的语言，即 8086/8088 微处理器汇编语言。

1.2.1 8086/8088 寄存器结构

为了便于运算和控制，在微处理器 8086/8088 中设有“算术逻辑单元”(ALU)、“控制逻辑”部件和一些寄存器。其中：

- ALU：实现算术运算功能。

- 控制逻辑部件：包含指令寄存器（以保存从内存读出的指令）及指令译码逻辑电路（以实现控制信号的产生）。
- 寄存器：用于寄存一些算术运算分量及地址运算分量。

图 1-4 所示是它的组成结构图。

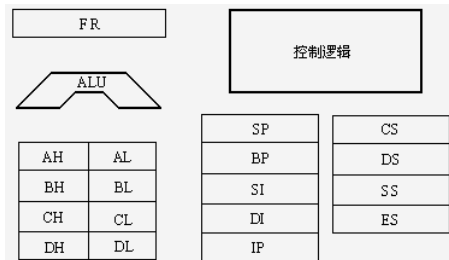


图 1-4 8086/8088 CPU 结构

下面是各类寄存器的用途。

1. 数据寄存器

微处理器 8086/8088 中设有 4 个 16 位的数据寄存器：AX、BX、CX 和 DX。它们中每一个都可以用于存放一个 16 位的二进制数。其中：

- AX 称作“累加器”，是算术运算中的主要寄存器。
- BX 称作“基址寄存器”，既可以存放算术运算中的数据，又可以存放地址分量供地址计算用。
- CX 称作“计数器”，既可以存放算术运算中的数据，又可以存放循环控制中的次数。
- DX 除了可以存放算术运算中的数据外，也可以与 AX 构成 32 位的寄存器，用来存放两个 16 位数据的乘积，或存放 32 位的被除数。

应当指出，这 4 个寄存器都可以作为 8 位的寄存器使用。比如说，AX 用作两个 8 位的使用时，可记作 AH 和 AL。由此，我们可以有下述 8 位的寄存器：

- AH：对应 AX 的高 8 位
- AL：对应 AX 的低 8 位
- BH：对应 BX 的高 8 位
- BL：对应 BX 的低 8 位
- CH：对应 CX 的高 8 位
- CL：对应 CX 的低 8 位
- DH：对应 DX 的高 8 位
- DL：对应 DX 的低 8 位

2. 段寄存器

8086/8088 中设有 4 个 16 位的段寄存器：CS、DS、ES 和 SS。它们中每一个都可

以用于存放一个 16 位的二进制数，以指示内存存储器的某一存储区的位置。其中：

- CS 称作“代码段寄存器”，所保存的数据是代码段在内存存储器中的位置。代码段就是指程序段，由一系列汇编指令组成。
- DS 称作“数据段寄存器”，所保存的数据是数据段在内存存储器中的位置。数据段由一系列的内存单元组成，主要用来存放初始数据、中间运行数据和最终结果。
- ES 称作“附加段寄存器”，所保存的数据是另一个数据段在内存存储器中的位置。
- SS 称作“堆栈段寄存器”，所保存的数据是堆栈段在内存存储器中的位置。关于堆栈概念将在下一小节介绍。

在 CPU 中设立段寄存器可以支持“分段存储”的模式，可方便地将程序区、数据区和堆栈区相互隔离。

图 1-5 是一个分段存储示意图。

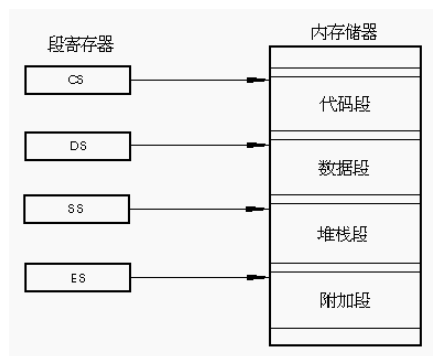


图 1-5 分段存储示意图

3. 指针寄存器及变址寄存器

CPU 中设有 2 个 16 位的指针寄存器 SP 和 BP，以及 2 个 16 位的变址寄存器 SI 和 DI。它们通常被用作地址计算，以指示内存存储器中某一存储单元的位置。其中：

- SP 称作“堆栈指针寄存器”，可与段寄存器 SS 联用，以指示内存存储器的堆栈段中处于栈顶的存储单元地址。
- BP 称作“基址指针寄存器”，可与段寄存器 SS 联用，以指示内存存储器的堆栈段中某个存储单元地址。
- SI 称作“源变址寄存器”，可与段寄存器 DS 联用，以指示内存存储器的数据段中某个存储单元地址。
- DI 称作“目的变址寄存器”，可与段寄存器 ES 联用，指示内存存储器的附加段中某个存储单元地址。

在某些场合，也将 BP 与 BX 称作“基址寄存器”，而将 SI 和 DI 称作“变址集训队器”。应当指出，这些寄存器除了上述功能外，还有其他一些用途，这要依据程序的

需要而定。

4. 控制寄存器

控制寄存器包括两个 16 位的寄存器 IP 和 FR（又称 Flag），其中：

- IP 称作“指令计数器”，是保存指令地址的寄存器。它可与段寄存器 CS 联用，以指示内存储器的代码段中某条指令的存储地址。CPU 将把此地址送往内存储器以读出所存的指令。
- FR 称作“标志寄存器”，用来指示 CPU 的运行状态。它由 16 位组成，其中每一位代表一种状态。

FR 各数据位所代表的含义如图 1-6 所示。

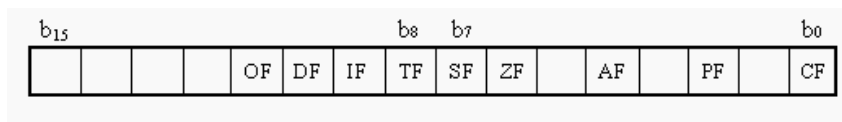


图 1-6 8086/8088 的标志寄存器 FR

图中各数据位的含义见表 1-1 所示。

表 1-1 8086/8088 的标志寄存器 FR 各数据位的含义

数据位	标识符	含义	为 1 时的含义
0	CF	进位标志	运算结果有进位或者借位
1	保留	保留	保留
2	PF	奇偶标志	运算结果的二进制数中有偶数个“1”
3	保留	保留	保留
4	AF	辅助进位标志	运算结果的低 4 位有进位或者借位
5	保留	保留	保留
6	ZF	零标志	运算结果为 0
7	SF	符号标志	运算结果的二进制数中最高位为“1”
8	TF	跟踪标志	CPU 按单步方式运行
9	IF	中断标志	开中断
10	DF	方向标志	在进行串操作时，内存地址按递减方式生成
11	OF	溢出标志	运算结果超出了机器的表示范围
12	保留	保留	保留
13	保留	保留	保留
14	保留	保留	保留
15	保留	保留	保留

表中只给出了各位为 1 时的含义，为 0 时的结果正好相反。

1.2.2 8086/8088 的地址计算

微处理器 8086/8088 产生的地址是由 20 位二进制数组成的，可以表示成 5 位的 16 进制数。其中，最小的地址码是 00000H，最大的是 0FFFFFFH。每个地址码只与一个内存单元对应。CPU 每次需要读取内存的一条指令或一个数据时，就将地址码通过 20 条地址线送往内存。

我们知道，CPU 的 4 个段寄存器分别对应内存的 4 个存储区域。这 4 个区域分别称为代码段、数据段、堆栈段和附加段。段寄存器中的 16 位二进制数指示的内存段地址，是其后添加 4 个二进制数 0 的一个 20 位的地址码，该地址码指示的是段的首地址。

已知 CS 的值等于 502AH，那么，它所对应的代码段首地址为 502A0H，二进制的表示形式为：01010000001010100000。

为了叙述方便，我们将用某寄存器与小括弧的联合来表示它的值，比如 (CS) 等。

当一个段在内存中的起始位置确定后，段内某个存储单元的地址就可以用位移量来指明。位移量又称作“位移地址”，由 16 位的二进制数组成。

图 1-7 描述了段地址、位移量与内存地址的对应关系。

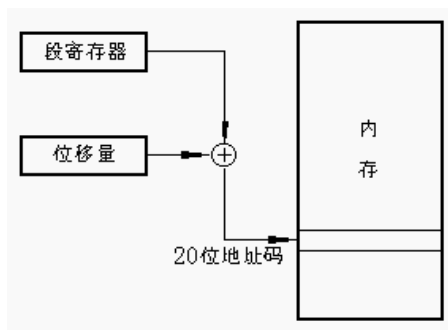


图 1-7 段地址、位移量与内存地址的对应关系

【例 1-1】已知 (DS) = 9C01 H, (BX) = 5016 H, 那么，数据段的首地址为 9C010H，求以 (BX) 为位移量的内存单元地址。

解：以 (BX) 为位移量的内存单元地址计算为：

$$\begin{array}{r} 9C010H \\ + 5016H \\ \hline 0A1026H \end{array}$$

1.2.3 堆栈地址生成

堆栈是内存的一个存储区域。CPU 访问该区域使用的段寄存器是 SS，位移量是 SP。