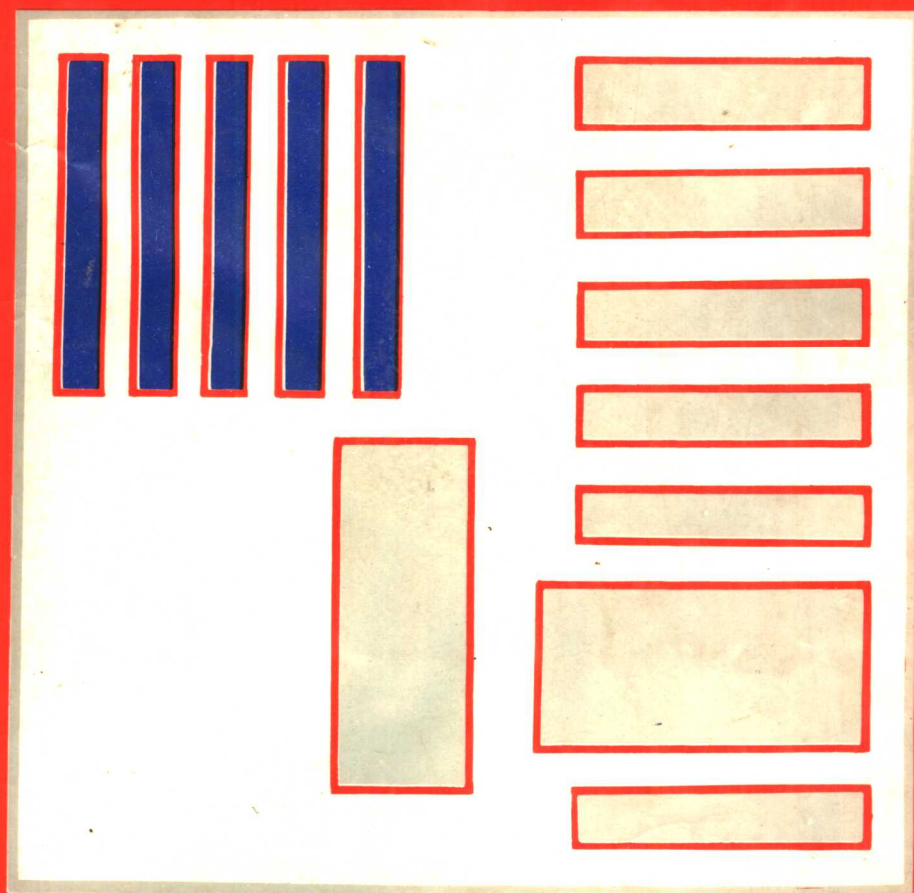


# IBM PC系列微机 接口与通讯 原理及实例

李克春 主编



大连理工大学出版社

IBM PC 系列  
微机接口与通讯原理及实例

李克春 主编

大连理工大学出版社

(辽)新登字16号

## 内 容 简 介

本书通过对 IBM PC 系统主要接口部件的剖析,从软硬件两方面对微型计算机的接口与通讯原理、编程方法进行了比较深入的研究。全书共分 7 章,前 3 章主要介绍 8088 指令系统及简单程序设计方法;第 4 章以 IBM PC 的打印机接口为例,说明基本的接口方法及硬件电路原理;第 5 章介绍 IBM PC 的系统接口部件;第 6 章介绍显示器接口原理及其编程方法;第 7 章介绍串行通讯原理,并给出 PC 机之间、PC 机与 Z-80 机、PC-1500 机之间的通讯连接电路以及程序清单。书后的附录给出了在编程中经常使用的 DOS 系统功能调用和 BIOS 各子程序列表。除第一章外,全书各章都提供了大量在 IBM PC 机上调试成功的程序实例,供读者使用。

本书可作为高等学校计算机及有关专业的教材,也适合于从事 IBM PC 系列微机应用和维护的工程技术人员使用。

### IBM PC 系列微机接口与通讯原理及实例

IBM PC Xilie Weiji Jiekou Yu Tongxun Yuanli Ji Shili

李克春 主编

---

大连理工大学出版社出版发行

大连海运学院印刷厂印刷

(邮政编码: 116021)

---

开本: 787×1092 1/16 印张: 10  $\frac{5}{8}$  字数: 255 千字

1990年6月第1版

1992年2月第2次印刷

印数: 3001-13000册

---

责任编辑: 刘晓晶

责任校对: 寸士

封面设计: 羊戈

---

ISBN 7-5611-0311-5/TP·23

定价: 3.50 元

# 前 言

自 70 年代微处理器诞生以来，以微处理器为中央处理机的微机系统层出不穷。而迄今为止，性能最强大、应用最广泛的微机系统首推 IBM PC 系列微机。现在，不仅 IBM 公司生产这种类型的微机，亚洲和欧洲的许多厂家都生产与 IBM PC 系列微机相兼容的微机系统。我国长城公司生产的 0520 系列微机也是 IBM PC 系列的兼容产品。IBM PC 不仅系统功能完善，而且有适应性极好的接口配置，它能够连接多种外部设备和应用电气装置。

接口技术是随着微处理器和可编程芯片等大规模集成电路产品的推出，在微机系统设计应用中产生的。要进行实际的接口电路的设计，首先要对计算机系统（硬件和软件）和微处理器有所了解。针对目前国内外微机发展和应用状况，本书以采用 Intel 8088 系列微处理器作为 CPU 的 IBM PC 系列微机系统为硬件范例，以 MICROSOFT 公司的 MS-DOS 为软件基础，通过对 PC 系统接口的分析，阐明接口构成原理和实用接口程序设计方法。

本书是在李克春老师主讲的“接口与通讯”课程讲义基础上编写的，共分两大部分，第一部分第一章到第三章，主要介绍 IBM PC 系统的硬件基础知识和软件知识。第二部分包括第四章到第七章，以 IBM PC 为例介绍接口原理与接口程序设计方法。其中第 2 章、第 3 章和第 7 章由李水同志编写，其余章节由李克春老师编写，全书由李克春老师审定。

本书以学懂会用为宗旨，每介绍一类接口就提供调试成功的实用程序供读者上机练习。这些程序都是作者在多年来的教学和科研中积累的经验 and 技巧。因此阅读练习这些程序，不但有助于消化理解本书内容，而且对实际工作有很大帮助。

由于本书最后定稿是在病房中，审定编写过程中遇到了许多困难，因此书中缺点错误在所难免，敬请读者批评指正。

作 者

1989 年 12 月

# 目 录

<b>第一章 IBM PC 硬件基础</b> .....	1
第一节 8088 和 IBM PC 的推出 .....	1
第二节 IBM PC 组成简介 .....	2
第三节 中央处理器 8088 .....	3
一、8088 的组成 .....	3
二、8088 的寄存器组 .....	4
三、分段结构和物理地址的形成 .....	5
四、管脚 .....	5
<b>第二章 DEBUG 调试程序和 8088 指令系统</b> .....	7
第一节 引 例 .....	7
一、声音产生 .....	7
二、字符串显示 .....	7
第二节 学习使用 DEBUG .....	8
一、对 DEBUG 的几点说明 .....	8
二、DEBUG 的基本命令及其应用 .....	9
第三节 8088 指令系统和寻址方式 .....	16
一、指令种类 .....	16
二、指令基本格式 .....	21
三、寻址方式 .....	22
第四节 程序设计初步 .....	24
一、软中断和字符输入输出系统调用 .....	24
二、算术运算 .....	26
三、无条件转移和循环 .....	28
四、接口的输入输出操作 .....	29
五、逻辑运算与位测试 .....	29
六、零转移指令和比较指令 .....	31
七、移位操作 .....	32
<b>第三章 宏汇编语言程序设计</b> .....	33
第一节 宏汇编语言简介 .....	33
一、宏汇编语言的基本概念 .....	33
二、宏汇编语言常用的伪指令 .....	34
第二节 宏汇编语言程序组织 .....	39
一、程序基本组成 .....	39

二、程序段前缀和标准序 .....	40
三、程序返回 DOS 方法 .....	41
四、过程(子程序)调用和堆栈操作 .....	43
<b>第三节 汇编过程 .....</b>	<b>45</b>
<b>第四节 宏汇编语言编程举例 .....</b>	<b>47</b>
一、大小写转换 .....	47
二、字符串匹配 .....	48
三、代码转换 .....	49
<b>第四章 基本接口方法 .....</b>	<b>52</b>
<b>第一节 接口基本概念 .....</b>	<b>52</b>
一、接口功能 .....	52
二、接口控制原理 .....	53
三、接口控制信号 .....	54
四、可编程接口芯片举例 .....	55
<b>第二节 接口寻址方法 .....</b>	<b>55</b>
一、直接地址译码 .....	55
二、SWITCH 可选地址译码 .....	56
三、间接端口寻址 .....	57
四、PROM 选择译码 .....	58
<b>第三节 数字输入输出电路 .....</b>	<b>59</b>
一、数字输出寄存器 .....	59
二、数字输入寄存器 .....	59
三、双向输入/输出寄存器 .....	60
<b>第四节 IBM PC 的输入输出系统 .....</b>	<b>60</b>
一、I/O 地址空间分配 .....	60
二、I/O 通道 .....	61
三、总线信号 .....	61
<b>第五节 IBM PC 的打印机接口插件板 .....</b>	<b>66</b>
一、打印机接口插件板简介 .....	66
二、地址译码电路 .....	67
三、输出电路和命令字 .....	68
四、输入电路和状态字 .....	68
五、打印机操作过程 .....	69
<b>第五章 系统接口部件 .....</b>	<b>71</b>
<b>第一节 IBM PC 系统板组成 .....</b>	<b>71</b>
一、8088 和 8288 .....	71
二、时钟电路 .....	72
三、ROM 芯片 .....	72
四、RAM 电路 .....	73

第二节 可编程并行接口 8255 .....	73
一、8255 的组成 .....	73
二、8255 的工作方式与控制字 .....	74
三、系统板上的 8255 .....	75
四、8255 编程举例 .....	77
五、8255 在汉卡和 EPROM 写入器中的应用 .....	78
第三节 8253 可编程定时器 .....	82
一、8253 的组成 .....	82
二、8253 的命令字和读 / 写操作 .....	83
三、8253 的工作方式及其定时图 .....	83
四、PC 系统板上的 8253 .....	86
五、声音和音乐 .....	88
第四节 可编程中断控制器 8259A .....	90
一、8259A 的组成 .....	91
二、8259A 的编程 .....	92
三、8259A 的工作方式 .....	95
四、PC 系统板上的 8259A .....	96
五、中断系统初始化和中断执行过程 .....	98
六、中断服务程序举例 .....	99
<b>第六章 显示器接口 .....</b>	<b>103</b>
第一节 显示原理 .....	103
一、光栅扫描 .....	103
二、字符图形显示和字符发生器 .....	103
三、存贮变换 .....	104
四、光标和滚行 .....	104
五、显示特征的实现 .....	104
第二节 CRT 控制器 MC6845 .....	105
一、MC6845 的组成 .....	105
二、寄存器的功能 .....	106
三、MC6845 的工作过程 .....	106
第三节 显示适配器 .....	107
一、单色显示适配器 .....	107
二、彩色 / 图形适配器 .....	110
第四节 显示器编程 .....	114
一、光标控制 .....	114
二、存贮映象 .....	116
三、滚动 .....	116
四、属性字节与彩色选择 .....	118
五、图形显示 .....	121

<b>第七章 串行通讯</b> .....	123
<b>第一节 串行通讯基础</b> .....	123
一、串并转换与设备同步 .....	123
二、串行通讯协议 .....	124
三、检错 .....	125
四、RS-232-C 标准 .....	126
五、20mA 电流环接口 .....	128
六、通讯控制字符和数据流控制 .....	128
<b>第二节 异步通讯控制器</b> .....	130
一、8250 的组成 .....	130
二、8250 的内部寄存器 .....	131
<b>第三节 异步串行通讯程序设计举例</b> .....	136
一、PC 机之间的通讯 .....	136
二、PC 机与 PC-1500 机的通讯 .....	140
三、PC 机与 Z80 机之间的通讯 .....	146
<b>附录</b>	
<b>附录一 BIOS 中的软中断调用</b> .....	152
<b>附录二 DOS 功能调用</b> .....	156
<b>参考文献</b> .....	161



# 第一章 IBM PC 硬件基础

本章介绍 IBM PC 系统的产生和组成, 以及 8088 微处理器的构造。

## 第一节 8088 和 IBM PC 的推出

1978 年以来, 在大规模集成电路技术发展的推动下, 美国最大的电子器件公司之一 INTEL 公司推出了全新的微处理器 8086。8086 能够处理 16 位字长数据的运算和传送, 其运算速度和精度都是当时最高的。鉴于 70 年代末期 8 位机系统的外部设备和各种 8 位接口器件的普遍采用, INTEL 公司随即推出了 16 位内部结构, 8 位外部数据总线的准 16 位微处理器 8088。

8088 的性能与 8086 相似, 其优点是能够方便地与 8 位机器件相连接构成性能价格俱佳的微机系统。

8088 使用与 8086 相同的指令系统, 支持各种 16 位的算术逻辑运算。由于它具有 20 条地址, 其寻址能力可达到 1M 内存空间。扩充性能良好。8088 的 8 位外部接口使它可以与 INTEL 公司的其它成熟产品(如 8259 中断控制器, 8237DMA 控制器等可编程接口芯片)共同构成功能更强的微机系统。IBM PC 就是采用 8088 作为中央处理器的成功范例。

IBM PC 是 1981 年 8 月 12 日, 由计算机行业巨人 IBM 公司推出的。一向垄断大型计算机市场的 IBM 公司首次涉足微机市场, 就获得了成功。IBM PC 充分利用了 8088 的高性能, 组合其它可编程接口芯片, 构成可靠性强, 性能超群的微机系统。IBM PC 采用开放式的接插兼容技术, 使系统扩充部件可以直接插入而不必修改原有电路。标准 IBM PC 配有 5 个扩充插槽, 继 PC 之后推出的 PC/XT 配有 8 个扩充插槽。IBM PC 的这种系统结构不仅使用户可以灵活地选择增加外设驱动插件, 也使系统本身具有良好的发展潜力。所以 IBM PC 一经推出, 就迅速成为微机市场上的主流产品, 获得了广大的用户。不仅有许多硬件厂家仿制生产 PC 的兼容产品, 还产生了许多专门为 IBM PC 编写程序的软件公司。

IBM PC 获得如此巨大的成功在于它对已制定工业标准的适应能力, 和对市场上已有产品的兼容性。实际上, IBM 公司只是 PC 的系统设计者, 其主要部件都是其它公司的产品。如中央处理器是 INTEL 公司的 8088, 早期配有 EPSON 公司的点阵打印机, 目前支持多种类型(包括来自日本的)新型打印机; 其键盘遵循西德 1980 年的 DIN 工业标准; PC 系统支持的串行通讯卡符合 RS-232 工业标准。IBM PC 这种兼容性的设计方法主要体现在其通用的接口设计上。

所谓接口, 即是在中央处理机和外部设备之间传递数据信号, 使两者动作协调一致的连接电路。IBM PC 系统中的接口电路大部分都是采用集成化的可编程接口芯片, 选用

多种接口芯片并加以编程，可以利用通用的硬件接口连接不同类型的外设，适应应用系统的实际要求。

IBM PC 不仅在硬件构成上大量采用其它公司的器件，其操作系统和主要应用软件也非本公司产品。IBM PC 最早支持的操作系统是 MACROSOFT 公司的 T. 帕蒂森先生编制的 PC DOS，除此之外，还支持 DIGITAL RESEARCH 公司的 CP/M 86 操作系统等。至于它支持的高级语言和应用软件，则更是五花八门。

IBM PC 成功的原因不仅是因为其功能完善的软硬件系统，还因为 IBM 公司在推出 PC 的同时，公布了它的几乎全部技术资料，甚至包括 BIOS 这样的操作系统底层软件和全部硬件电路图。这为 IBM PC 赢得了许多长期的用户，他们可以在 IBM PC 上进行较深入的开发工作。从这个意义上说，以 IBM PC 为例阐述接口设计方法也是非常有利的。

## 第二节 IBM PC 组成简介

IBM PC 虽然性能较强，其组成仍具有一般微机系统的共同特点。从广义上看，现代微机系统基本上都要包括：主机系统(包括中央处理器和存贮器)，外部设备(包括键盘、显示器、磁盘机和打印机)，以及负责管理系统设备的系统软件(DOS 等)。如图 1-1 所示，为 IBM PC 的一般组成。

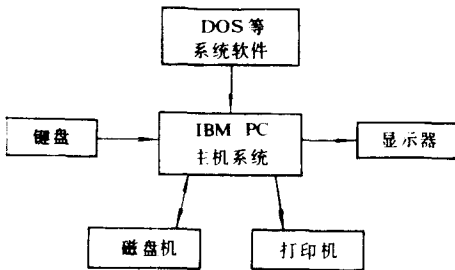


图 1-1 IBM PC 系统

在主机系统上运行着 DOS(操作系统)，管理着所有外设的动作。外部设备通过各自的接口与主机系统相连，作为微机系统的核心——中央处理器，在开机时执行操作系统程序，维持整个系统的运行。下面分别介绍 IBM PC 的主要组成部分。

### 1. 主机系统

主机系统包括中央处理器 8088，DMA 及磁盘数据交换控制电路，内存 RAM 芯片和存放底层软件的 ROM 芯片，还包括中断控制器、时钟和定时器电路，键盘、扬声器和磁带机接口电路也被集成在主机系统中。在主机系统中还连接着系统总线的外部电路——I/O 通道和系统电源等设备。

### 2. 键盘

IBM PC 的键盘共 83 个键，其主要部分为普通打字机键，数字键组、功能键组和其它专用控制键。键盘与主机通过四芯螺旋状电缆连接，在击键过程中，键盘接口芯片两次将键码送到主机，以消除击键抖动产生的错误。IBM PC 的键盘综合了传统打字机和计算器键盘布局，以提供通用的数据和文字处理功能。

### 3. 打印机

早期 IBM PC 配有每秒打印 80 个字符的针式打印机，其接口是并行标准 Centronix 接口。它能连接有类似接口的多种打印机，如目前流行的 2024 型、FX100 型等 24 针、9 针打印机。

打印机除可打印 96 个标准 ASCII 字符外，还提供一组 64 个特定图形符号。另外，主机系统通过向打印机发出由 ESC 字符(通讯控制代码)开始的控制序列代码，可以控制

打印方式。

#### 4. 显示器

IBM PC 支持两类显示器。一类是单色显示器，通过单色显示适配器与主机相连，它提供 25 行×80 列的文本方式显示。另一类是彩色图形显示器，它提供多种文本和图形方式显示。其中图形方式分低、中、高三种分辨率，高分辨率可达每屏 200×640 点。这两种显示器接入系统时要插入相应的接口驱动电路——适配器驱动。

#### 5. 磁盘系统

通过磁盘适配器，IBM PC 可连接最多 4 个磁盘驱动器，即可同时插入系统四张软磁盘。两个在主机系统内，两个在外部。这种软磁盘的存储容量为 160k 或 320k 字节，系统提供对软磁盘的读、写成组信息的功能。

PC 之后推出的 PC/XT 型微机除配有软盘驱动器外，还提供一台存储容量为 10M 字节的硬盘。

#### 6. 其它

除以上主要部件外，通过插入相应的接口适配器，IBM PC 还可以连接异步通讯设备、游戏器和实验插件板等外部设备。

在 IBM PC 的发展过程中，产生了 PC、PC/XT、PC/AT 等多种型号的微机系统，虽然在性能上彼此都有独特之处，但其基本组成结构却是一脉相承的。所以只要掌握了 IBM PC 的基本组成，弄清其它类型微机的构造就非常容易了。

### 第三节 中央处理器 8088

8088 是 INTEL 公司在 8 位机向 16 位机过渡期间推出的准 16 位机，它既能处理 8 位数据，又能处理 16 位数据。8088 微处理器是采用 N 沟道耗尽型硅栅技术，封装在 40 条引脚的管壳中的大规模集成电路。其主要特点有：

- \* 4.77MHz 的时钟频率
- \* ±5V 电源
- \* 能进行包括乘除在内的 8 位和 16 位算术逻辑运算
- \* 16 位内部结构(具有 14 个 16 位的寄存器),8 位数据总线接口
- \* 20 条地址线,寻址能力可达 1M 字节。

8088 的平均运算速度为 65 万次/秒。在当时属于性能最强的微处理器，至今仍广泛使用。

#### 一、8088 的组成

8088 在结构上采用了流水指令队列、地址计算和算术逻辑运算分布处理的先进技术。其主要组成包括执行部件和总线接口部件两部分。如图 1-2 为 8088 的组成框图。

执行单元 EU 包括 16 位通用寄存器组和算术逻辑运算部件。EU 解释执行来自指令队列中的指令，完成算术逻辑运算动作。EU 产生数据在内存中的相对地址，接受来自内部总线的的数据。

总线接口单元 BIU 专门计算实际内存地址，不仅把来自 EU 的数据相对地址转换为实际地址，还负责提供当前取指令的内存地址。BIU 包括指令指针寄存器、段寄存器、4

个字节的指令队列、地址计算逻辑和执行机构等。

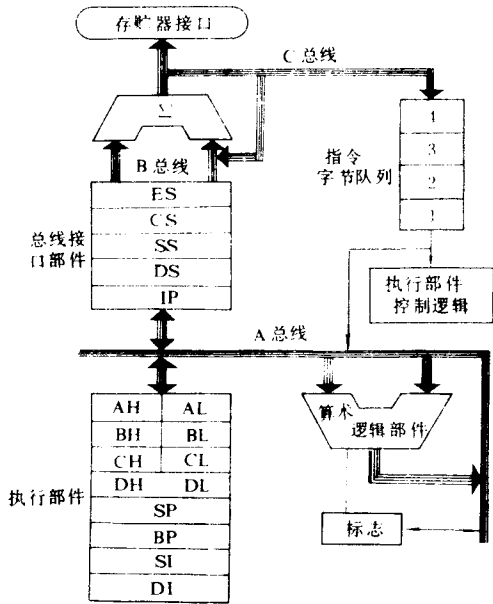


图 1-2 8088 的组成框图

引寄存器 DI、堆栈指针 SP、基址寄存器 BP。它们都是 16 位字长。

SI 和 DI 通常在数据传送时用于存放数据地址。SP 和 BP 通常在访问堆栈时使用。

### 2. 段寄存器和指令地址寄存器

段寄存器包括代码段寄存器 CS、数据段寄存器 DS、附加段寄存器 ES 和堆栈段寄存器 SS。这 4 个段寄存器在 8088 访问存储单元时用于计算实际地址。

指令地址寄存器 IP 是段寄存器 CS 基础上的指令地址偏移，其指向将要执行的下一条指令。

### 3. 标志寄存器

标志寄存器(IF)在执行单元 EU 中，它由反映算术或逻辑操作结果特征的标志位组成，如图 1-3 所示。

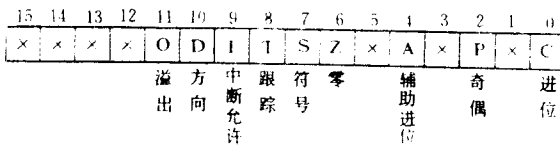


图 1-3 标志寄存器

## 二、8088 的寄存器组

在图 1-2 中有三类寄存器，第一类为位于 EU 的 16 位通用寄存器和变址寄存器，第二类为位于 BIU 的段寄存器和指令地址寄存器，第三类为标志寄存器，在 EU 中。

### 1. 通用寄存器和变址寄存器

这一类寄存器主要用于存放数据及其内存地址。

通用寄存器包括 AX、BX、CX、DX 4 个，它们既可按 16 位访问，也可分为高 8 位和低 8 位使用。如 AX 为按 16 位存取数据的寄存器符号，AH 和 AL 分别表示其按 8 位存取时的高 8 位和低 8 位。通用寄存器在运算过程中保存数据。

变址寄存器用于保存内存数据的相对地址，包括源地址索引寄存器 SI、目标地址索引寄存器 DI、堆栈指针 SP、基址寄存器 BP。它们都是 16 位字长。

这些标志位的说明如下：

(1) 辅助进位标志表示为 A，该位置“1”表示 16 位寄存器的低 8 位向高 8 位进位或借位。这个标志位可用于十进制算术指令。

(2) 进位标志表示为 C，该位置“1”表示 16 位或 8 位多字节加减运算时最高位发生进位或借位，C 也可以用循环移位指令移入的内容来决定。

(3) 溢出标志表示为 O，该位置“1”表示算术运算的结果发生溢出。有一条中断指令(INTO)能够在 O 置“1”时，产生硬件中断，使程序自动转向溢出中断处理服务子程序。

(4) 符号标志表示为 S，该位表示补码运算结果的正负号，S=1 时为负值，S=0 时为正。

(5) 奇偶标志表示为 P，该位表示操作结果“1”位的奇偶数。P=1 时表示偶数，P=0

表示奇数。这个标志位可用于数据传输时的奇偶校验。

(6) 零标志表示为 Z，该位置“1”表示操作结果为零。

(7) 方向标志表示为 D，该位标志可用程序进行设置，当 D=1 表示字符串指令执行时，地址从高到低自动减 1 或从左向右；当 D=0，表示地址从低到高或从右向左。

(8) 中断允许标志表示为 I，该位置“1”表示准许 CPU 响应外部可屏蔽的中断请求，I=0 表示禁止外部可屏蔽的中断。I 对不可屏蔽的中断和内部中断没有影响。I 标志可用中断指令 STI 和 INT 进行置位。

(9) 跟踪标志表示为 T，该位置“1”表示让 CPU 执行单步方式，以便调试程序。

在这种方式中，CPU 每执行完一条指令后就自动产生一次内部中断。T 标志可用程序进行置位。

### 三、分段结构和物理地址的形成

8088 的地址线有 20 条，而其内部逻辑都是基于 16 位的。为寻址 20 位地址指示的内存空间，8088 采用分段结构来组织存储器。如图 1-4 所示，逻辑上将整个内存空间划分为 4 个部分：代码段、堆栈段、数据段和附加段。各段地址由段寄存器给出，16 位段寄存器的基本单位是  $2^4$  代表 20 位地址的高 16 位。段空间最大为 64k 字节，各段在内存中的实际位置可以变化。

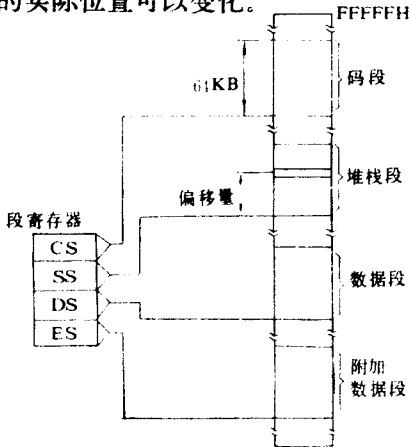


图 1-4 存储空间分段结构

在总线接口单元 BIU 中，算术逻辑专用于 20 位物理地址的产生。在指令执行时，根据指令性质和要求选定某一段寄存器，把段地址右移 4 位，再与来自执行单元 EU 的逻辑地址（偏移量）相加，其和即为实际物理地址。这一过程如图 1-5 所示。

如取指令时用代码段寄存器 CS 值右移 4 位后，与指令指针寄存器 IP 中的偏移量相加，其和即为实际指令在内存中的地址。同理，寻址堆栈时选择 SS 与 SP，寻址数据时选择 DS 或 ES 与相应偏移量相加。

分段寻址不仅解决了利用 16 位寄存地址访问 1M 内存空间的难题，而且有利于程序的再定位。在内存对于程序长度不超过 64k 的代码位置改变，只要修改代码段寄存器值为定位地址即可。

### 四、管脚

在接口硬件设计中，管脚信号往往就是对应接口信号。而中央处理器的管脚信号与各类接口芯片管脚相比，是最全面的，下面介绍 8088 的管脚定义。如图 1-6 所示，有些管脚有两种定义。

#### 1. $AD_0 \sim AD_7$

$AD_0 \sim AD_7$  这 8 个引脚在总线周期开始时，给出地址信号的低 8 位  $A_0 \sim A_7$ ，在总线周期的后半段又用作数据总线信号  $D_0 \sim D_7$ ，传送接收数据。

#### 2. $A_8 \sim A_{15}$

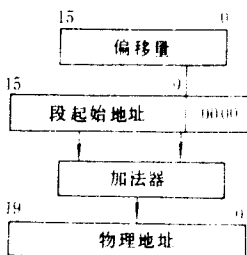


图 1-5 物理地址的形成

高 8 位地址  $A_8 \sim A_{15}$  在每个总线周期内保持稳定, 访问存储器时为内存地址高位, 访问接口时为输入输出端口地址高位。

GND	1	40	$V_{CC}$
$A_{14}$	2	39	$A_{15}$
$A_{13}$	3	38	$A_{16}/S_3$
$A_{12}$	4	37	$A_{17}/S_4$
$A_{11}$	5	36	$A_{18}/S_5$
$A_{10}$	6	35	$A_{19}/S_6$
$A_9$	7	34	(HIGH) ( $\overline{SSO}$ )
$A_8$	8	33	$MN/\overline{M\bar{X}}$
$\overline{AD}_{15}$	9	32	$\overline{RD}$
$\overline{AD}_{14}$	10	31	$\overline{RQ}/\overline{GT}_0$ (HOLD)
$\overline{AD}_{13}$	11	30	$\overline{RQ}/\overline{GT}_1$ (HLDA)
$\overline{AD}_{12}$	12	29	$\overline{LOCK}/\overline{WR}$
$\overline{AD}_{11}$	13	28	$\overline{S}_2$ (IO/ $\overline{M}$ )
$\overline{AD}_{10}$	14	27	$\overline{S}_1$ (DT/ $\overline{R}$ )
$\overline{AD}_9$	15	26	$\overline{S}_0$ ( $\overline{DEN}$ )
$\overline{AD}_8$	16	25	$QS_0$ (ALE)
NMI	17	24	$QS_1$ ( $\overline{INTA}$ )
INTR	18	23	TEST
CLK	19	22	READY
GND	20	21	RESET

图 1-6 8088 的管脚

### 3. $A_{16}/S_3 \sim A_{19}/S_6$

在访问存储器时, 这些管脚为存储地址  $A_{16} \sim A_{19}$ ; 其余时间则作为 8088 的内部状态管脚。

### 4. CLK 和 READY

CLK 为时钟输入端, 与 8284A 时钟发生芯片相连。

READY 信号也来自 8284A 时钟发生器。它用于在 8088 总线周期中插入“等待”状态信号, 响应中央处理器指示数据传递完毕信号。

### 5. NMI 和 INTR、INTA

NMI 为不可屏蔽中断输入信号线, INTR 是来自中断控制芯片的可屏蔽信号, INTA 是中断响应信号。

### 6. $\overline{RD}$ 、 $\overline{WR}$ 、和 $\overline{M}/IO$ 、ALE

执行存储器或输入/输出读周期时产生读数据  $\overline{RD}$  信号。

$\overline{WR}$  信号确定写存储器还是写输入/输出设备。

$\overline{M}/IO$  用来区分是从输入/输出设备中, 还是从存储器中存取数据。

由中央处理器提供的 ALE 信号, 将地址锁存于总线地址锁存器中。

### 7. 其它

8088 还有用于连接多处理机系统的管脚信号, 如  $\overline{RQ}/\overline{GT}_0$  和  $\overline{RQ}/\overline{GT}_1$  等, 还有指令特征信号, 如 LOCK 和 TEST 等。这些信号本书极少涉及, 这里不作详细介绍。

## 第二章 DEBUG 调试程序和 8088 指令系统

### 第一节 引 例

本节介绍的例子意在给出使用 DEBUG 编写简单的接口程序的一般步骤。

#### 一、声音产生

通过对扬声器接口编程，可以使 IBM PC 发出声音。下面程序就可以完成这个功能。

```
A>DEBUG ↓           ; 进入 DEBUG 程序。
-A100 ↓             ; “-”为 DEBUG 提示符,键入 A100 并回车,然后输入程序。
08F1: 0100  in  al,61 ↓
08F1: 0102  and al,fc ↓
08F1: 0104  xor al,2 ↓
08F1: 1006  out 61,al ↓
08F1: 0108  mov cx,140 ↓
08F1: 010B  loop 10b ↓
08F1: 010D  JMP 106 ↓
08F1: 010F  ^C
-G ↓               ; 程序输完同时按下 CTR 和 C 键, 出现“-”, 按 G 并回车,
                  ; 此时可以听到主机箱内发出蜂鸣声。
```

这是用 DEBUG 编写的一个简单程序例子。其中“A”和“G”都是 DEBUG 命令，后面将会介绍。该程序第 1 条语句从扬声器接口读入控制字，第 2、3 条语句修改控制字的有关控制位，第 4 条语句送出发声控制字。至此，接口已设定为发声状态。第 5、6 句控制发声持续时间，最后一条语句转回程序起始点，再次驱动发声。

该程序虽然简单，却是一个非常典型的接口驱动程序，结果是明显的。但是这个小程序并不完善，它执行后无法中止，只好同时按 CTR、ALT、DLE 三个键重新启动机器。

#### 二、字符串显示

汇编语言程序不仅能管理接口，而且能完成在屏幕上显示字符的“高级”功能。这是 IBM PC 系统的独特之处。下面程序执行后将在屏幕上显示字符串“good morning”。

```
A>DEBUG ↓           ; 进入 DEBUG
-A100
1141: 0100  MOV DX, 109
1141: 0103  MOV AH, 9
1141: 0105  INT 21
1141: 0107  INT 20
1141: 0109  DB 'good morning $'
1141: 0116  ^C           ; 按 CTR+C 结束
-G           ; 按 G 执行
good morning
Program terminated normally
```

这个程序利用系统调用完成在屏幕上显示字符串，并返回 DEBUG。其原理将在第

三节说明。

由上面两例可以看出：8088 指令系统功能完善，接口管理方便。另外，利用 DEBUG 编写简单程序，直观明了。所以本章首先介绍如何使用 DEBUG 程序，然后利用 DEBUG，进一步掌握指令系统。

## 第二节 学习使用 DEBUG

DEBUG 程序是 IBM PC 的 DOS 操作系统提供的一个通用工具。它向汇编语言程序设计者和系统管理员提供一个可控制的调试环境，监视和管理被调试程序的执行。使用 DEBUG 可以直接深入到机器系统内部，读写中央处理机各寄存器和内存单元，并可以访问接口和外设的寄存器。DEBUG 的具体功能如下：

- \* 读写、比较存贮单元内容；
- \* 在存贮器之间，存贮器和磁盘之间传送数据和程序；
- \* 把数据和字符串填入指定内存单元，并在指定范围内查找数据或字符串；
- \* 指定程序起址或断点，执行或分段执行程序；
- \* 显示寄存器内容、内存单元内容；
- \* 跟踪程序执行，显示处理器状态；
- \* 汇编和反汇编程序。

本章正是利用 DEBUG 的上述功能编写调试程序的。在 DEBUG 下编程并执行，不仅可以直观地了解程序的执行情况，而且便于熟悉常用指令用法，掌握基本的编程技巧。

### 一、对 DEBUG 的几点说明

DEBUG 是一个性能很强的工具，但它有一些特点，使用者应当注意。

#### 1. DEBUG 下的数据表示

在内存中，数据以二进制形式存在，但 DEBUG 采用十六进制压缩表示内存数据。1 位十六进制数字代表 4 位二进制数。小于 10 的数字与十进制数字相同，大于等于 10 到 15 的数字用字母 A 到 F 表示。如表 2-1 所示。

8088 的内存空间以字节为单位，二进制数的最小单位是比特(bit)，即 1 位二进制数。所以在 DEBUG 下，一个字节的内存单元包含 8 位二进制数，表示为两位十六进制数。如下例所示，在 DEBUG 下，内存数据表示为 3 部分：地址、十六进制数据及其对应的 ASCII 码。

```
1141: 0100 | 49 42 ..... 00 | IBMBIO COM'....
1141: 0110 | 00 00 ..... 00 | .....< >.....
1141: 0120 | 49 42 ..... 00 | IBMDOS COM'....
1141: 0130 | 00 00 ..... 00 | .....< >....B..
1141: 0140 | 43 4F ..... 00 | COMMAND COM....
1141: 0150 | 00 00 ..... 90 | .....T...E..
1141: 0160 | 54 45 ..... 00 | TETRIS BAK.....
1141: 0170 | 00 00 ..... 00 | .....e.l.*'.....
```

其中十六进制数据区中一行有 16 个字节，8 个字节间用“—”分隔。



表 2-1

二进制 (B)	十进制 (D)	十六进制 (H)
0000	0	00
0001	1	01
0010	2	02
0011	3	03
0100	4	04
0101	5	05
0110	6	06
0111	7	07
1000	8	08
1001	9	09
1010	10	0A
1011	11	0B
1100	12	0C
1101	13	0D
1110	14	0E
1111	15	0F

## 2. DEBUG 下的 ASCII 码

在上面例子中，右边区域是与中间区域数字对应的 ASCII 码。在 DEBUG 下，作为数据的字母和数字都可以表示为 ASCII 码，这在保存输入输出字符的数据区中尤为明显。

ASCII 码是美国信息交换标准码，采用 7 位二进制数字表示 128 个不同字符。IBM PC 系列微机采用扩展的 ASCII 码，可表示 256 个不同字符，增加了一些图形字符等。常用 ASCII 码数 0 到 9，表示在 30H 到 39H 之间。字母包括大写和小写两种，大写字母 A 到 Z 的 ASCII 码值在 41H 到 5AH 之间；小写字母 a 到 z 的 ASCII 码值在 61H 到 7AH 之间。

## 3. 机器语言文件

在 IBM PC 的 DOS 操作系统下支持两类机器语言文件：扩展名为 .EXE 的可执行文件和扩展名为 .COM 的命令文件。在 DEBUG 下可以将这两类文件反汇编出来，以汇编语言程序的形式显示出来。

用户在 DEBUG 下编写的汇编语言程序可以以 .COM 文件的形式存入磁盘中。这种用户 .COM 程序可以在 DOS 下直接执行。

## 二、DEBUG 的基本命令及其应用

IBM PC 系列微机上，有两种编写汇编语言程序的方法：宏汇编和 DEBUG。宏汇编比较复杂，下一章专门介绍。对于初学者，使用 DEBUG 编程简单直观。DEBUG 只有几条单字母命令，但功能很强且容易掌握。另外，在 DEBUG 下执行程序，可以直接观察机器的物理运行状态，非常适合于熟悉指令系统和系统硬件环境。下面分别介绍各个