

非计算机专业Ⅲ级水平考试用书

微型计算机技术及其应用

苏广川 编著



北京理工大学出版社

微型计算机技术及其应用

非计算机专业Ⅲ级水平考试用书

苏广军 编著

北京理工大学出版社

(京)新登字 149 号

内 容 简 介

本书是按着“全国高校电子与信息类非计算机专业Ⅲ级水平考试大纲”的要求编写的。为了使本书较为实用,作者凭着多年教学和科研的实践,以 8086/8088CPU 为例,深入系统地介绍了微机的组成、原理、指令系统、接口与通讯技术,以及数据采集等。特别在多处理器系统设计;DOS 环境与进程管理;驱动程序设计方法和用汇编语言设计弹出式菜单、鼠标控制等高级图形编程技术方面均有显著特色。

本书内容丰富,选材适当,软硬件融于一体;书中配有大量应用程序实例并附有中文注释;可适用于教学、自学、应试强化。因而是高校非计算机专业教学和参加全国计算机Ⅲ级水平考试的理想用书。

微型计算机技术及其应用

非计算机专业Ⅲ级水平考试用书

苏广川 编著

*

北京理工大学出版社出版发行

各地新华书店经售

北京市万龙图文信息公司照排

北京地质印刷厂印刷

*

787×1092 毫米 16 开本 23.875 印张 583 千字

1994 年 10 月第一版 1994 年 10 月第一次印刷

ISBN 7-81013-119-2/TP·12

印数:1-7000 册

定价:18.00 元

前 言

本书是根据全国高校电子与信息类非计算机专业Ⅲ级水平考试大纲编写的,以 IBM PC/XT 机为典型,全面介绍微机系统的硬件组成、宏汇编语言程序设计以及接口技术等。

本书将软、硬件内容融于一体,能使非计算机专业学生,在较短时间内,由学习基础知识入手,系统地掌握微机原理,学会宏汇编语言程序设计,掌握解决工程实际问题的计算机软、硬件设计技术。书中提供大量应用程序实例并附有中文注释,方便读者阅读程序和理解程序设计的思想。因而是高校电子与信息类非计算机专业理想教学用书,亦可作此类专业参加全国计算机Ⅲ级水平考试参考书。

考虑到不同层次读者的需要,本书还介绍了目前流行的程序设计思想,诸如多处理器系统设计,结构化模块化设计,DOS 环境和进程管理,菜单高级图形编程技术,鼠标程序设计等,使教材具有一定的先进性。

全书共分七章,第一章介绍 8086/8088 CPU 的工作原理、微机系统组成以及多处理器系统设计技术;第二章介绍 8086/8088 宏汇编指令系统和汇编语言程序设计方法,结构化模块化设计技巧,DOS 环境与进程管理,驱动程序设计方法以及磁盘文件管理等技术;第三章介绍半导体存储器分类、内部结构组成、地址译码技术以及存储器与 CPU 连接的硬件结构等;第四章介绍输入输出接口技术,其中包括并行接口 8255A、定时器 8253、DMA 8237 控制器以及 8259A 中断控制器等;第五章介绍如何用汇编语言设计窗口管理、弹出式菜单、鼠标控制等高级图形编程技术;第七章介绍数据采集系统的设计方法以及 A/D、D/A 应用技术。

在编写本书过程中得到向茂楠教授和李光宇副教授的大力支持,在此表示衷心感谢。

由于编者水平所限,加之时间仓促,书中一定存在不少缺点和错误,敬请读者批评指正。

编 者

1994 年 6 月

目 录

第一章 微型计算机概论

第一节 微机发展概况	(1)
第二节 数制和编码	(2)
一、计算机中的数制	(2)
二、二进制编码	(4)
第三节 微机系统结构分析	(6)
一、中央处理器	(6)
二、系统总线	(7)
三、存储器	(8)
四、输入输出接口	(8)
五、系统软件	(8)
第四节 CPU 的内部结构分析	(9)
一、8086 /8088 微处理器结构	(9)
二、8086 /8088 内部寄存器	(11)
三、8086 /8088 引脚及其功能	(14)
第五节 系统时钟和等待状态产生器	(18)
一、8284 时钟发生驱动器	(18)
二、等待状态逻辑	(19)
第六节 总线控制器 8288	(20)
第七节 总线周期时序分析	(22)
一、存储器读写总线周期	(22)
二、I/O 读写周期时序	(24)
三、中断响应周期	(24)
第八节 8289 总线仲裁器	(24)
第九节 多处理器系统	(32)

第二章 8086/8088 指令系统及程序设计

第一节 寻址方式	(40)
第二节 8086/8088 指令系统	(43)
一、数据传送指令	(43)
二、算术运算指令	(46)
三、程序转移指令	(55)
四、逻辑运算和移位指令	(59)
五、数据串操作指令	(62)
六、处理机控制指令	(65)
第三节 汇编语言和汇编程序	(66)
一、汇编语言语句	(67)
二、汇编源程序结构	(68)
第四节 宏指令	(76)
第五节 条件汇编语句	(80)
第六节 结构化模块化程序设计	(81)
一、结构化程序设计步骤	(81)

二、 模块化程序的设计	(84)
第七节 汇编语言与高级语言软件接口技术	(90)
第八节 数据的输入输出与代码转换	(92)
第九节 宏汇编语言程序设计举例	(100)
第十节 DOS 环境及进程管理	(125)
一、 DOS 环境	(125)
二、 程序段前缀 PSP	(125)
三、 内存管理	(127)
四、 进程管理	(129)
五、 设备驱动程序	(144)
第十一节 磁盘文件管理	(154)
一、 磁盘记录格式	(154)
二、 PC-DOS 分区	(156)
三、 磁盘文件管理	(160)
第三章 半导体存储器	
第一节 半导体存储器分类	(178)
一、 只读存储器 ROM	(178)
二、 随机读写存储器 RAM	(179)
第二节 半导体存储器的内部结构	(179)
第三节 只读存储器	(182)
一、 掩模只读存储器 ROM	(182)
二、 可编程只读存储器 PROM	(183)
三、 可重复擦写的只读存储器 EPROM	(183)
四、 电擦可编程只读存储器 E ² PROM	(186)
第四节 随机存储器	(188)
一、 静态随机读写存储器 SRAM	(188)
二、 动态随机存储器 DRAM	(189)
三、 非易失随机读写存储器 NVRAM	(191)
第四章 输入输出接口技术	
第一节 接口技术的基本概念	(194)
一、 接口功能	(194)
二、 端口编址方式	(195)
三、 输入输出传送方式	(196)
第二节 8255A-5 可编程并行接口及应用	(198)
一、 8255A-5 芯片的结构及其功能	(199)
二、 8255A-5 在系统中的应用	(203)
第三节 8253A-5 可编程定时器及其应用	(207)
一、 8253A-5 的内部结构	(207)
二、 8253A-5 的工作方式	(209)
三、 读写说明	(210)
四、 8253A-5 在系统中的应用	(212)
第四节 8237A-5 DMA 控制器及其应用	(215)
一、 8237A-5 芯片的硬件结构和功能	(215)
二、 DMA 读写操作	(221)

三、DMA 工作方式	(222)
四、DMA 传输类型	(223)
五、几个特殊操作的说明	(223)
六、8237A-5 在系统中的应用	(225)
第五节 中断及其应用	(229)
一、中断类型和中断向量表	(229)
二、中断响应过程	(231)
三、8259A 中断控制器及其应用	(231)
第五章 异步通讯接口技术	
第一节 计算机网络概述	(246)
一、网络结构	(246)
二、网络体系结构	(248)
第二节 异步通讯接口	(250)
一、串行通讯的一般概念	(250)
二、串行总线标准 RS-232C	(254)
三、20mA 电流环标准	(257)
第三节 异步通讯适配器	(258)
一、8250 可编程异步通讯接口	(258)
二、8250 内部寄存器功能	(259)
三、INS 8250 通讯编程	(261)
第六章 图形显示高级实用编程技术	
第一节 显示模式	(269)
第二节 文本方式	(271)
第三节 图形显示	(277)
一、通过 BIOS 向显示区读写像素	(278)
二、利用图形控制寄存器对显示区读写	(279)
第四节 图形高级编程技术	(285)
一、建立宏库	(285)
二、画点	(286)
三、画直线	(289)
四、画圆	(292)
五、图形的拷贝	(297)
六、动画图形显示	(298)
七、填充矩形	(299)
八、在屏幕上开窗	(300)
九、存储屏幕图形	(302)
十、将磁盘中的图形文件写入显示缓冲区	(304)
第五节 窗口管理文件及编程技术	(307)
一、物理视频缓冲区	(307)
二、逻辑视频缓冲区	(308)
第六节 鼠标输入器及其编程	(317)
一、鼠标器程序设计接口	(318)
二、鼠标器功能调用简介 INT 33H	(320)
三、对鼠标及键盘的编程	(324)

第七节 菜单设计及其编程	(337)
第七章 数据采集系统及其应用	
第一节 数据采集系统的组成	(352)
第二节 A / D 接口电路设计	(356)
一、ADC0809 工作原理及接口电路	(356)
二、AD574A 工作原理及接口电路	(357)
三、应用 DMA 技术的高速数据采集系统	(363)
四、利用周期选通采样的高速数据采集系统	(363)
第三节 D / A 转换器及其应用	(364)
一、DAC0832 工作原理及接口电路	(364)
二、DAC1210 工作原理及接口电路	(365)
附录一 IBM PC ASCII 码字符表	(367)
附录二 8086/8088 指令系统表	(368)
参考书目	(373)

第一章 微型计算机概论

第一节 微机发展概况

微型计算机技术是工程人员最为重要的技术基础之一。以前将计算机仅仅视为一种工具的概念正在逐步被更新,现在计算机已成为解决各种复杂问题的主要手段,它是生产力中最为积极最为活跃的要素。微型计算机的广泛应用对科学技术和国民经济的发展都起着巨大的促进作用,例如计算机辅助设计(CAD)技术水平,已成为衡量一个国家工业水平的重要标志之一。在社会主义市场经济的大潮中,一个人的计算机水平高低,也成了谋求良好职业的重要砝码。

微型计算机自 70 年代初诞生以来,经历着许多重大的发展阶段,而每一发展阶段都是大规模集成电路发展的重要里程碑。

1971 年至 1973 年为微机的萌芽阶段,典型产品为 INTEL 公司的 4004(4 位机)和 8008(8 位机),采用 P-MOS 集成工艺。

1973 年至 1975 年为微处理器设计进入成熟阶段,其典型产品为 INTEL8080 和 MOTOROLA 公司的 MC6800,采用 NMOS 工艺制成。此后又相继出现集成度高、运行速度快的 8 位机,如 INTEL 8085,ZILOG 公司生产的 Z80 以及 ROCKWEL 公司生产的 6502,这些 CPU 在微机上得到广泛而较长期的运用。

1978 年至 1980 年是微机的重要发展阶段,采用硅栅 H-MOS 工艺制造的 16 位微处理器,典型产品有 INTEL 8086/8088、MC68000、Z8000,微机的功能和运算速度已初步达到小型机水平,其间与 8086/8088 CPU 相配套的有 8087 数值运算协处理器,主要支持浮点运算和函数运算。8087 有自己的指令系统,能独立地执行指令,在协处理器内部有 47 个字的现场状态寄存器,它包含当前正在执行语句的地址、程序的控制字和程序状态字以及浮点栈等内容,它是 8087 的工作现场。与 8086/8088 相配套的还有 8089 I/O 处理器,这是一个单片处理器,包含两通道,它有 53 种指令组成的指令系统,专门用于编制通道程序,它有信息装配/拆卸部件,能使不同宽度的编码设备交换信息。IOP 的通道能进行字符的输入输出,也能进行 DMA 传送,并且在传送中能对数据进行翻译、比较及校错,IOP 能方便地进行各外部设备之间、外部设备与存储器之间以及有存储器区域之间的信息传输。

80186/80188 是 8086/8088 的增强型,把 8086/8088 的外围器件如 DMA、计时器、中断控制器以及时钟发生器等集成到 CPU 内部,它用专用硬件加法器计算物理地址,而 8086/8088 是依靠微程序来实现的。80186/80188 还使用硬件乘除运算,并对指令系统有进一步增强。

80286 是 1982 年推出的一种超级 16 位机。CPU 包括了执行部件、地址部件、总线部件和指令部件,它们可以并行工作,从而提高了吞吐能力。80286 芯片具有存储器管理和保护功能,它支持实地址方式和虚拟地址保护方式下工作,80286 的实际存储空间为 16 兆字节,而每一个用户任务可使用的虚拟存储空间达 1000 兆字节。80286 采用实地址方式寻址时,能与 8086/8088 软件的目标码兼容,在保护虚地址方式时与 8086/8088 的软件的原码兼容。80286 具有实

时多任务和多道程序处理能力,它能以四层特权来支持操作系统和任务的分离,并能使程序和数据保密。

80年代中期又推出32位,超级微型机,如INTEL 80386,MC68020等,它采用了微包装技术和微组装技术,无论从性能上还是从速度上都超过了70年代末期至80年代初期的小型机水平。它在专家系统、机器人以及控制系统、工程工作站、办公室自动化、科学计算、软件开发CAD/CMA等方面应用潜力极大。80386的体系结构是全32位的,其中包括32位寄存器、32位地址总线、32位数据总线、32位内部控制总线以及32位的外部总线接口,并且还支持8位、16位和32位的数据类型。80386的芯片内有完整的存储管理部件,既可分段又可分页管理,也可以不分段不分页,80386扩展了80286的分段模式,以支持4千兆字节物理存储器分段。80386带保护的虚拟存储器容量高达64兆兆字节,由于芯片的全32位体系结构所提供的高度并行操作和快速度局部总线,由硬件所提供的多任务处理和保护机构更增强了系统的性能和完整性,使80386步入了大型机的范畴。

第二节 数制和编码

一、计算机中的数制

计算机中主要采用二进制数运算,编写程序又主要用十六进制形式表示操作数,而日常生活中大量使用的是十进制数。将十进制数转换成二进制数或将二进制数转成十进制数,这些关系在计算机输入输出的程序中经常出现。其实不管是二进制、十进制还是十六进制,我们可用一般表达式来表示:

$$\begin{aligned} N_R &= K_n R^n + K_{n-1} R^{n-1} + \dots + K_1 R^1 \\ &\quad + K_0 R^0 + K_{-1} R^{-1} + K_{-2} R^{-2} + \dots + K_{-m} R^{-m} \\ &= \sum_{i=-m}^n K_i R^i \end{aligned} \quad (1-1)$$

式中 R 为基数,表示为 R 进位制,如 $R=2$,即二进制形式表示一个实数; K_i 为第 i 位系数,可以在 $0, 1, \dots, R-1$ 范围内取任一合法单元,如 $R=16$, K_i 可以为 $0, 1, 2, 3, \dots, A, B, C, D, E, F$,其中 $A \dots F$ 分别表示 10 至 15 的数; n, m 为幂指数,正整数; R^i 为第 i 位的权。

利用式(1-1)可将任意进制数化成十进制数,其中小数点左面的各数位的权是 R 的正次幂,小数点右面的各位数的权是 R 的负次幂。如

$$\begin{aligned} (1011.11)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= (11.75)_{10} \end{aligned}$$

再如 $(2B.1)_{16} = 2 \times 16^1 + 11 \times 16^0 + 1 \times 16^{-1} = (43.0625)_{10}$

各种数制转换方法叙述如下:

1. 将十进制转换为二进制

对于十进制整数可以归纳的为“除2取余”,对十进制小数,则为“乘2取整”。若十进制为实数,则将上述规则联合使用。如十进制数21,则利用除2取余规则计算过程如下:

$$\begin{array}{r}
 2 \overline{) 21} \\
 2 \overline{) 10} \text{ 余 } 1 \\
 2 \overline{) 5} \text{ 余 } 0 \\
 2 \overline{) 2} \text{ 余 } 1 \\
 2 \overline{) 1} \text{ 余 } 0 \\
 0 \text{ 余 } 1
 \end{array}$$

在除 2 的过程中,最先的余数作二进制的低位,最后的余数作二进制的高位。这样得:

$$(21)_D = (10101)_2$$

再如十进制小数为 0.625,依据“乘 2 取整”规则,将最先得到的整数作为小数部分最高位,最后得到的整数作为小数的最右位。注意:乘 2 之后整数位不再参加运算。

$$\begin{array}{r}
 0.625 \\
 \times 2 \\
 \hline
 1.250 \\
 \times 2 \\
 \hline
 0.500 \\
 \times 2 \\
 \hline
 1.000
 \end{array}$$

由此可得

$$(0.625)_D = (.101)_2$$

同样可得

$$(21.625)_D = (10101.101)_2$$

即将一实数化为二进制数表示。

2. 将二进制转换成十六进制

只需从小数点开始,向左数 4 位二进制数用对应的十六进制数表示,不足 4 位补零,以此作为十六进制的整数部分;从小数点向右数,4 位二进制数用对应的十六进制数表示,不足 4 位亦补零。我们仍以上面的二进制数为例,表示成十六进制形式,得

$$(10101.101)_2 = (15.A)_{16}$$

3. 将十进制数转换成十六进制

可以先将十进制数转换成二进制数,然后再将二进制利用上述方法转换成十六进制。利用“除 R 取余”和“乘 R 取整”的规则,可以将十进制数化成 R 进制数,这样做更为方便。例如十进制数 235 化成十六进制时,可以采用

$$\begin{array}{r}
 16 \overline{) 235} \\
 16 \overline{) 14} \quad \dots \text{余 } B \\
 0 \quad \dots \text{余 } E
 \end{array}$$

得

$$(235)_D = (EB)_{16}$$

在 INTEL8086 宏汇编中十六进制数用“H”记。二进制数用“B”记。十进制数用“D”记,在书写时采用隐含规则不写 D,即

$$235 = EBH = 11101011B$$

这些数的形式经常在编程中使用。在汇编语言中为区别数和标号,约定:凡以字母当头的十六进制数其左面一律添加一个“0”,以表示数。否则汇编语言则认为是地址标号或变量,如标号或变量未定义,则在汇编时将显示错误信息。如 E9H 应写成 $\phi E9H$,符号“ ϕ ”记为零以区别字母

“0”。

二、二进制编码

1. BCD 码

在计算机中,不管是数还是符号都用二进制数表示。在计算指令中可以支持十进制运算,1 位十进制数用 4 位二进制数表示,权值为 8421 我们称之为 BCD 码。例如十进制数为 75,则对应的 BCD 码为 01110101B(简称为 75H)。初学者常常对 BCD 码与十六进制数两个概念区分不开。的确,从形式上看两者表现一致,但是从编程者来说应该清楚,BCD 码是十进制数的一种表现形式,它的每位限制在 0~9 的范围。从后面的讨论可知,在进行十进制运算时,所有的数都必须用 BCD 码表示,计算机按十六进制运算,并对运算结果进行二~十进制调整,我们在微型计算机中常定义 8 位二进制为一个字节(BYTE)。用一字节表示 2 位十进制数,称组合的 BCD 码。若一字节就表示 1 位十进制数,其中高 4 位为零,称未组合的 BCD 码。

2. ASCII 码

在机器中字母和符号利用 7 位或 8 位二进制数进行编码。目前在微机中最普遍采用的是 ASCII(American Standard Code for Information Lnterchange 美国标准信息交换码),IBM PC ASCII 码字符表见附录(1)。从表中可以看出 0~9 的 ASCII 码为 30H~39H,大写字母 A~Z 的 ASCII 码为 41H~5AH 小写字母 a~z 的 ASCII 码为 61~7AH,回车符的 ASCII 码为 0DH。

3. 有符号数编码

在机器中的正负数可采用多种编码方式表示,连同符号位在一起作为一个数,称为机器数;而它对应的数值为机器数的真值。常用的编码方法有原码、反码和补码,实际大量使用的是补码。

(1) 原码

正数的符号位用 0 表示,负数的符号位用 1 表示,这种表示方法为原码。如:

$$X = +100 \quad [X]_{\text{原}} = 01100100\text{B} = 64\text{H}$$

$$X = -100 \quad [X]_{\text{原}} = 11100100\text{B} = \text{E}4\text{H}$$

原码的最高位为符号位,去除符号位,余下的位表示数的绝对值。所以正数的原码,只需将正数用二进制数表示,高位用 0 补齐。例如上数 100 用 8 位原码表示为 64H,若用 16 位表示则为 0064H。对于负数的原码,因为仅和对应的正数的原码最高符号位不同,因而将对应的正数原码的最高符号位改为 1 即可。

(2) 反码

正数的反码表示与原码相同,这个基本概念常被忽视,如

$$[+100]_{\text{反}} = 01100100\text{B} = 64\text{H}$$

而负数的反码则为它对应的正数的反码按位取反(连符号位)。如

$$[+45]_{\text{反}} = 00101101\text{B} = 2\text{DH}$$

$$[-45]_{\text{反}} = 11010010\text{B} = \text{D}2\text{H}$$

(3) 补码

正数的补码表示与原码相同,而负数的补码为对应的正数补码按位取反后在最低位加 1。

如

$$[+115]_{\text{补}} = 01110011\text{B} = 73\text{H}$$

$$[-115]_{\text{补}} = 10001101\text{B} = 8\text{DH}$$

若用 16 位二进制表示补码,则

$$[+115]_{\text{补}} = 0000000001110011\text{B} = 0073\text{H}$$

$$[-115]_{\text{补}} = 1111111110001101\text{B} = \text{FF}8\text{DH}$$

从以上讨论中可以看出,反码可理解为对“1”求补,补码是对“2”求补。即一字节的正数的反码与它对应的负数反码求和为 FFH ,而补码则为 100H 。正由于补码具有这种特性,在进行减法运算时,可将减数用补码表示,与被减数相加,进位自然丢失,其和为运算的正确结果。为加深对补码概念的理解,我们举一个浅显的十进制数减法例子,如

$$\begin{aligned} 9 - 8 &= 9 + (-8) = 9 + (10 - 8) - 10 \\ &= 9 + 2 - 10 = 1 \end{aligned}$$

其中,对十进制来说,−8 的补码为 +2,这就是自然丢失进位的道理,二进制(或十六进制)运算亦然,例如

$$78\text{H} - 25\text{H} = 78\text{H} + [-25\text{H}]_{\text{补}} = 78\text{H} + \text{DBH} = 53\text{H}$$

$$\begin{array}{r} 01111000\text{B} \\ +) 11011011\text{B} \\ \hline \boxed{1}01010011\text{B} \end{array}$$

↑
自然丢失

若两数相减无借位,则结果可能为负数,如

$$25\text{H} - 78\text{H} = 25\text{H} + [-78\text{H}]_{\text{补}} = 25\text{H} + 88\text{H} = \text{ADH}$$

$$\begin{array}{r} 00100101\text{B} \\ +) 10001000\text{B} \\ \hline 10101101\text{B} \end{array}$$

在微机中,凡是带符号的数一律用补码表示。由于计算机的字长有一定限制,所以表示带符号数有一定范围,如字长为 8 位时,表达数的范围为 $+127 \sim -128$,当运算结果超过这个表达范围,结果就不正确了,这称之为溢出。解决的方法是将参加运算的数,进行符号扩展,用多字节表示。例如

$$7\text{AH} + 52\text{H} = \text{CCH}$$

两个正数相加,得出结果为负数,称之溢出,其和超过了 $+127$ 范围,为使运算结果正确,可以把两个数扩展成 16 位表示 $7\text{AH} \rightarrow 007\text{AH}$ 、 $52\text{H} \rightarrow 0052\text{H}$,这样

$$007\text{AH} + 0052\text{H} = 00\text{CCH}$$

其结果为正。

若 $(-112) + (-30)$ 用 8 位表示时,其补码为

$$90\text{H} + \text{E}2\text{H} = 72\text{H}$$

两负数相加得正数,产生溢出,若将 -112 和 -30 分别用 16 位补码表示,则变为

$$\text{FF}90\text{H} + \text{FFE}2\text{H} = \text{FF}72\text{H}$$

由此可见,正数符号为 0,将其扩展时,符号扩展位全用 00 来表示;若是负数,因为符号位为 1,所以符号扩展位为 FFH 。8086/8088 有专用指令来实现符号扩展。

第三节 微机系统结构分析

微型计算机系统(Microcomputer system)由硬件和软件两大部分组成,其中硬件部分包括:中央处理器 CPU、系统总线、存储器和输入输出接口以及电源等如图 1-1 所示。

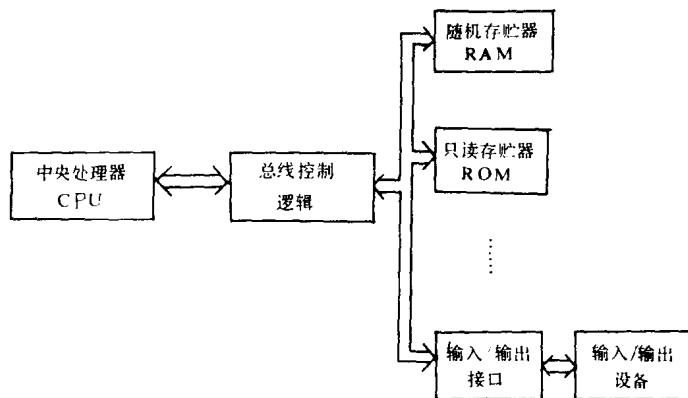


图 1-1 计算机系统结构

计算机所配置的系统软件有操作系统、编译系统、程序库和数据库等。下面对这些部分作一简要说明。

一、中央处理器(Central Processing Unit 简称 CPU)

中央处理器采用了大规模集成技术将运算器和控制器集成在一块芯片内,完成数据的处理和全机的控制。CPU 一般由三大部分组成:指令控制部件、算术逻辑单元 ALU 和寄存器(Register)。各部件之间由片内总线连接,进行信息的传送,现分述如下:

1. 指令控制部件

该部件主要功能是取指令、分析指令和执行指令,并发出相应的控制信号。它主要包括:(1)程序计数器,(2)指令寄存器,(3)指令译码器,(4)指令控制电路,产生执行此条指令所需的全部控制信号。

2. 算术逻辑单元

该部件完成算术逻辑运算指令所指定的运算操作。参加运算的操作数一般由源操作数和目的操作数组成,运算结果或暂放在寄存器中,或通过内部数据总线传送到由目的操作数所规定的地址中。而 ALU 运算结果的状态是否为 0,有没有进位,是为正还是为负等,将在状态标志寄存器中反映,然后程序根据状态标志的不同对程序的流向进行判断和转移,计算机正是利用这些特性来完成程序所规定的极其复杂的工作。

3. 寄存器组

在 CPU 内部设置了一个寄存器组,专门用于存放操作数据和地址,如数据寄存器 AX、BX、CX、DX,变址寄存器 SI、DI,堆栈指针 SP 以及段寄存器等。利用这些寄存器组可以减轻系统访问存储器的次数和减轻单总线使用压力,从而提高指令的执行速度。内部寄存器的个数及位数都反映一个 CPU 性能的高低。

CPU 执行指令的过程如下:首先由程序计数器的内容,送至地址寄存器且计数器自动加

1, 然后地址寄存器的内容送至地址总线并由控制单元发出相应读指令信号, 从存储单元中取出指令码取得指令码, 经数据总线送入 CPU 的内部数据寄存器, 并通过内部总线传至指令寄存器。然后再传至指令译码器进行译码, 按指令操作码识别指令的操作类型, 按地址码分解源操作数地址和目的操作数地址, 再由控制电路向控制总线发出各种控制信号, 取源操作数送 ALU 进行运算, 并将运算结果送至由目的操作数所规定的寄存器或存储器中。

二、系统总线

总线(BUS)是微机中连接各部件的一组公共通讯线, 一般设有发送门、接收门。根据总线不同用途, 总线可以双向传输或单向传输, 并采用三态电路(逻辑“1”、逻辑“0”、“浮空”或高阻状态)。总线可允许有多个发送端和多个接收端, 但同一时刻只能允许一个发送门打开, 向总线传送信息, 否则将引起总线竞争, 产生难以预料的结果。

总线结构使微机在系统结构上具有简单、规整和易于扩展的特点, 使整个系统中各功能部件之间的相互关系为面向总线的单一关系。这样, 系统只要将符合总线规范的功能部件接到总线上, 系统的功能就可得到扩展。

从连接方式总线可分为三大类:

① 内部总线。又称片级总线或局部总线, 用于 CPU 芯片与外围芯片间的互连。

② 系统总线。又称板级总线或内总线, 它是微机特有的一种总线, 用于系统中各插件板之间的互连, 便于系统的扩展和更新。由于模块间的连接局限于微机系统内, 连线数目多且距离近, 因而采用并行总线。目前常用的总线标准如 S-100 总线, INTEL MULTIBUS, MOTOROLA EXORCISE-86。在这些标准中, 对插件尺寸、插头线数、各引线的定义及时序作出了明确的规定。

③ 外部总线。又称通讯线或外总线, 它用于微机系统间的通讯网络或用于微机系统与电子仪器和其设备相连。这种总线不是微机所特有的, 它借用电子工业的总线标准, 如: 串行总线 EIA-RS232、并行总线 IEEE-488。

按信息的类型分, 总线又可分为: 地址总线 AB(Address Bus)、数据总线 DB(Data Bus)和控制总线 CB(Control Bus)。

① 地址总线(AB)。它是一组专门用来传送地址信息的总线, 一般为单向传送, 地址总线的宽度将决定 CPU 可直接寻址的内存容量范围, 如 8 位微型机的总线数为 16 根(A0~A15), 所以它是最大内存容量为 $2^{16}=64\text{K}$ 字节。对于 PC-XT 机, 它具有 20 根地址线, 可访问的内存空间为 $2^{20}=1\text{M}$ 字节。但对于 PC-AT 机(CPU 为 80286)地址总线为 24 根, 它具有最大内存容量为 $2^{24}=16\text{M}$ 字节空间。

② 数据总线(DB)。它是传送数据信息的总线, 具有双向传送功能。数据总线上数据线根数多少通常与 CPU 外部的数据总线宽度相一致, 如 8 位微型机的数据总线由 8 根组成(D0~D7), 16 位机 8086 的数据总线为 16 根。但 16 位机 8088 的数据总线却只有 8 根, 主要考虑与 8 位机原有外部设备兼容。由于 8088 内部的通信线是 16 位的, 故称 8088 为准 16 位机。

③ 控制总线(CB)。它传送各种控制信号, 控制信号包括 CPU 对外围芯片和 I/O 接口的控制, 如: 读信号 $\overline{\text{RD}}$, 写信号 $\overline{\text{WR}}$, 访问存储器或访问 I/O 的控制信号 $\text{IO}/\overline{\text{M}}$, 以及总线响应信号 HLDA 等。有些是外部设备送至 CPU 的控制信号, 如中断请求信号 NMI、INTR 以及总线请求 HOLD, 复位信号 RESET 信号。控制总线根据 CPU 选择的工作方式不同, 有单向双向之分。如 8086 在最大模式下工作时, $\overline{\text{RQ}}/\overline{\text{GT}}$, 总线请求和总线应答为同一根线。控制信号出现的

时间,确定了微型机的工作时序。

三、存储器

它是计算机的记忆装置,用来存放程序和数据。一个存储体由若干个存储单元组成,每个存储单元都编有号码(称为地址)。目前使用的存储器,存储单元以字节(Byte,8位二进制数)为单位。

存储器通常有内存和外存之分。内存器大都采用半导体器件组成,是一个计算机系统必不可少的组成部分。外存储器主要采用硬盘和软盘,其信息记录在磁介质表面上,存储容量大,但存取速度慢。

半导体存储器从存取方式来分,可分为随机存储器(RAM)和只读存储器(ROM)。前者存储单元内容既可读出,亦可写入,写入即修改,新信息一旦写入,原有信息就丢失。而ROM中的内容在一般情况下只能读出,不能写入和修改。在电源关闭后,ROM中存储的信息不会丢失。在只读存储器中还包括用紫外线光可改写的EPROM和用电可改写的EEPROM,这些为用户保存自己的程序提供极大的方便。

四、输入输出接口

一般计算机系统都具有输入设备键盘、输出设备打印机、显示等,而磁盘驱动器既可作输入又可作输出设备,我们常将这些输入输出设备称之为“外设”。微处理器和外设打交道远不如与存储器那样简单,输入输出(简称I/O)设备品种繁多,各种I/O设备信息传送速度差异甚大,如硬盘每秒能传输兆位信息量,而打印机每秒只能打印数百字符。此外,在各种I/O设备上,信息存储的格式各不相同,微处理器与特定的I/O设备实现信息交换,都需进行特定的格式转换。再者,各种I/O设备都有自身的定时控制逻辑,与微处理器的时序极不匹配,因此有必要在微处理器和I/O设备之间,设置专门的接口电路来解决这些问题。这样,I/O接口电路起到连接微处理机与I/O设备的媒介作用。

五、系统软件

系统软件通常是指系统应该配置的那一部分软件,用于对计算机的管理和使用。例如操作系统、编译程序、汇编程序、编辑程序、调试程序、诊断程序、标准库程序等。操作系统是紧挨着裸机的第一层软件,通过操作系统对硬件功能进行扩充,并在操作系统统一管理下运行,因而是整个计算机系统的控制和管理中心。操作系统的主要功能可归纳为处理机管理、存储器管理、设备管理和信息管理。

① 处理机管理。它的主要任务是根据一定的原则,选择较好的策略,作好处理机的调度、分配及回收工作,使处理机充分发挥效率,并尽可能合理地满足各任务的不同需要。

② 存储器管理。它主要是为多任务并发运行提供基础,因此操作系统必须随时记录存储空间的情况,进行存储空间的划分、分配及回收工作,让多个任务共存于内存,并要提供存储保护,保证各任务之间互不侵犯。

③ 设备管理。它的基本任务是使得多个用户能够共享设备,并且使用户能够方便地使用设备。为此,设备管理对设备进行合理分配(包括分配通道和控制器),并按照用户要求控制设备工作。

④ 信息管理。信息管理任务是对文件的存储空间进行组织分配、负责文件的存取,并对存

入的文件进行保护与检索。

在微型机中使用的操作系统很多,主要代表有 CP/M、PC-DOS 和 UNIX 三大类型。

CP/M:它是监督控制程序(Control Program/Monitor)的简称。其本身的结构分三层:控制台命令处理层 CCP,提供了 CP/M 与用户的接口;基本磁盘操作系统层 BDOS。它是 CP/M 的核心层;基本输入输出系统层 BIOS。CP/M 采用的文件管理系统,每一文件都以唯一的文件名标识,放在文件目录中,用户只需按文件名在磁盘上取文件,而不必了解文件映射在磁盘中的物理地址。用户利用 CP/M 提供的各种命令,能方便地生成、修改、删除、复制、读写各种文件。此外,CP/M 还提供了系统功能调用,用户能灵活地运用这些功能调用,而无需了解硬件结构的具体细节。

CP/M 操作系统在 8 位微型机中得到最为广泛的应用。CP/M-86、CP/M-68000、CP/Z8000 为单用户单任务 16 位微机的操作系统。1984 年问世的并发 CP/M-86,可同时执行几个实时应用程序,并提供先进的多窗口环境,可使 256 任务常驻在同一系统中。

PC-DOS:它是 IBM-PC 个人计算机的主流操作系统。它是单用户单任务操作系统,具有很强的文件管理能力,为用户提供了丰富的系统资源,如较多的内部、外部命令,功能强大的系统调用等。PC-DOS 自 1981 年问世以来,推出了许多新版本,DOS1.0、DOS1.1 只支持低密软盘驱动器,文件管理采用文件控制块方式。DOS2.0 是最先支持硬盘和树状目录结构,并增加了后台打印和文件重新定向等功能。DOS3.0 增加了对高密 1.2MB 的软盘支持,它将硬盘容量增加到 32MB,支持 16 位的文件分配表(FAT),DOS3.1 支持 IBM 个人计算机网络及网络程序。DOS3.2 支持 720KB 和 3.5 英寸软盘。DOS3.3 支持大于 32MB 硬盘的分区,生成逻辑驱动器。DOS4.0 支持对硬盘非分区的方法,对大于 32MB 的硬盘能够做为一个单一分区来使用,但对 DOS3.3 分区硬盘风格仍然支持。DOS4.0 支持扩展内存,但由于 DOS4.0 占用内存开销大,DOS4.0 版本的本身的内部错误太多,运行起来可靠性差以及该版本带来许多不兼容性,因而 DOS4.0 版本没有得到广泛推广。DOS5.0 于 1991 年 7 月推出,支持 2GB 硬盘分区,支持 2.88M3.5 英寸软盘,支持扩展和扩充内存,Dos shell 全屏幕编辑器以及 QBASIC 等。

1987 年推出的 OS/2,是 Microsoft 公司设计的单用户多任务操作系统,它将成为 286/386 机的主流操作系统。

UNIX:支持多用户多任务分时操作系统。系统的 90%的程序用 C 语言写成,汇编语言只占 10%,因此系统具有良好的移植性,UNIX 将作为 32 位微型机的主要操作系统。

第四节 CPU 的内部结构分析

一、8086/8088 微处理器结构

8088 微处理器内部结构由执行部件 EU(Execution Unit)和总线接口部件 BIU(Bus Interface Unit)两个独立的处理部件组成。其内部结构如图 1-2 所示,执行指令的任务由执行部件 EU 来完成,而取指令、读操作数和写运算结果等,需要执行外部的总线周期,则由总线接口部件 BIU 来完成。

1. 执行部件 EU

执行部件 EU 由通用寄存器、算术逻辑部件 ALU、标志寄存器 FR、暂存寄存器和 EU 控制系统等组成,内部总线为 16 位。EU 的功能是负责全部的指令的执行,向 BIU 输出数据和地