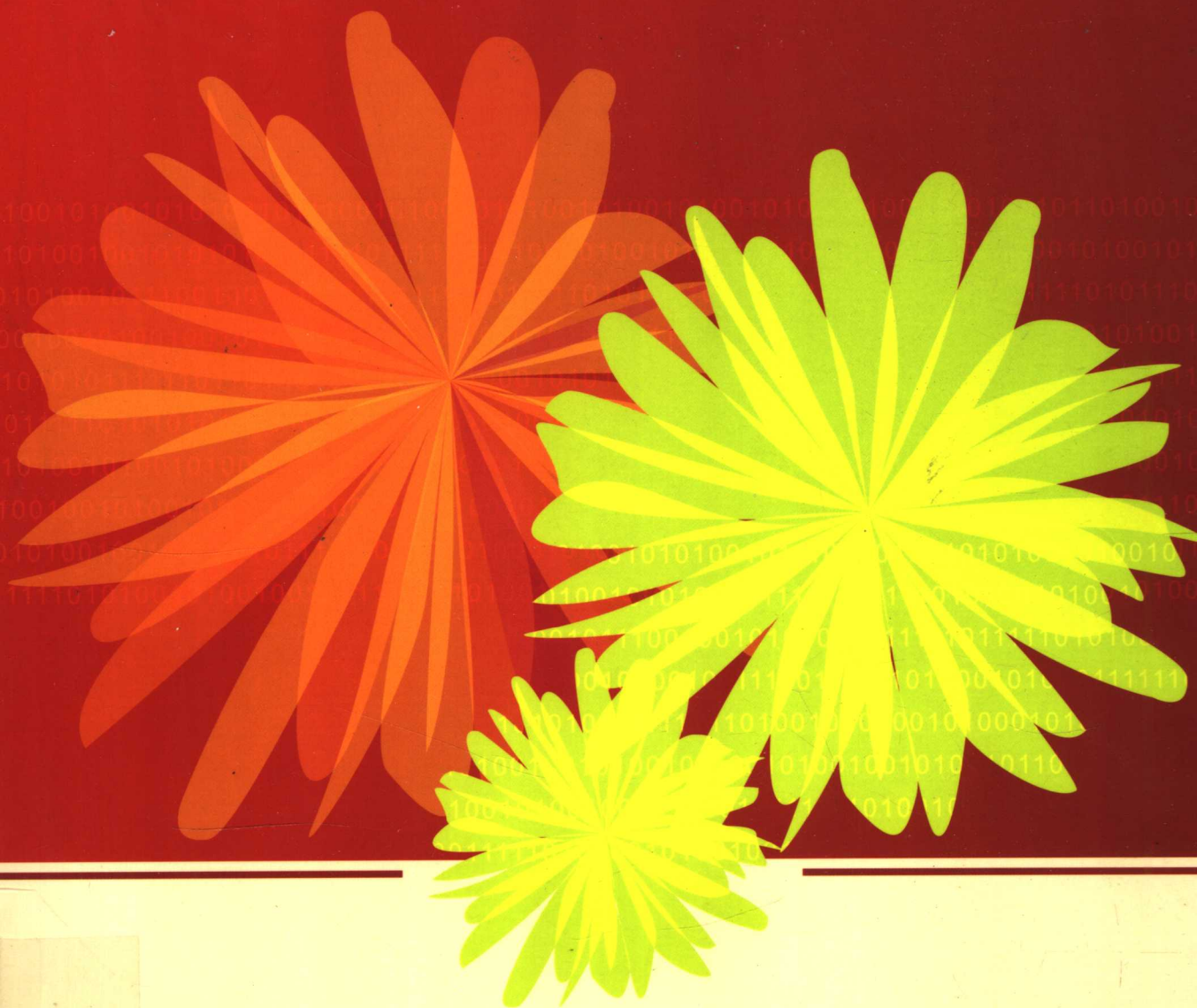




微机原理与接口技术简明教程

主 编 王 丰

副主编 田红芳 邱俊杰 谭 娟

 北京航空航天大学出版社

TP36

448

2007



高职高专“十一五”规划示范教材

微机原理与接口技术简明教程

主 编 王 丰

副主编 田红芳 邱俊杰 谭 娟

北京航空航天大学出版社

内 容 简 介

本书较系统地介绍了16位微型计算机的基本原理及接口技术。全书共分12章,主要包括:微型计算机的组成及工作原理、指令系统、MASM伪指令系统、汇编语言程序设计、8086微处理器结构、半导体存储器、输入/输出技术、中断、并行接口芯片8255A、串行通信、其他常用接口、总线等。

本书以培养学生应用能力为主要目标,以介绍基础知识和基本技术为主。内容组织深入浅出,语言叙述浅显易懂,在兼顾学科知识完整性的同时,针对高职高专院校的教学特点进行了内容优化,以达到重点突出、内容简洁实用的目的。

本书可作为高职高专院校计算机专业的教材,或作为自动化、机电等非计算机专业的本科生教材,也可供工程技术人员、自学人员参考使用。

图书在版编目(CIP)数据

微机原理与接口技术简明教程/王丰主编. —北京:北京航空航天大学出版社,2007.5
ISBN 978-7-81077-969-2

I. 微… II. 王… III. ①微型计算机—理论—高等学校—教材②微型计算机—接口—高等学校—教材 IV. TP36

中国版本图书馆CIP数据核字(2007)第020430号

微机原理与接口技术简明教程

主 编 王 丰

副主编 田红芳 邱俊杰 谭 娟

策划编辑 蔡 喆

责任编辑 刘晓明

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:15.25 字数:342千字

2007年5月第1版 2007年5月第1次印刷 印数:4 000册

ISBN 978-7-81077-969-2 定价:21.00元

前言

微机原理与接口技术是计算机及相关专业的一门重要专业基础课,是计算机组成原理、计算机系统结构和计算机控制系统等课程的前导课。

本书以 PC/XT 为主要背景机,以 8086/8088 CPU 为基础来介绍汇编语言程序设计、微机工作原理及接口开发的相关知识。从市场角度看,16 位机作为 20 世纪 70 至 80 年代的产品,已经过时;但从教学的角度看,它却是一类极佳的机型。其结构简单明了,同时又能完整地体现现代计算机各方面的理念和技术,即所谓的“麻雀虽小,五脏俱全”。以该机型为模型机教学,可有效避免现代 586 类计算机过度的复杂性和过多、过杂的技术细节,使学生能掌握计算机工作原理和框架的脉络结构。同时,考虑到计算机技术本身的发展,本书中对一些新技术也作了相应的介绍。

本书的前身《微机原理与接口技术》(王丰、王兴宝编著)于 2005 年 3 月由北京航空航天大学出版社出版。该书是作者在十多年教学科研经验的基础上精心编纂而成的,概念清晰,内容翔实,深入浅出,出版后得到了全国多所高职高专院校的充分肯定并被选为教材。在充分听取了使用院校的意见后,本书在原书的基础上作了较大改动:主要是精简了中断控制器 8259A、DMA 控制器 8237 等内容,同时改写了串行通信及总线等内容,以达到在保持原教材风格的基础上针对专科教学特点精简原教材的目的。

全书共分 12 章,内容包括:微型计算机的组成及工作原理、指令系

统、MASM 伪指令系统、汇编语言程序设计、8086 微处理器结构、半导体存储器、输入/输出技术、中断、并行接口芯片 8255A、串行通信、其他常用接口及总线等。在附录部分,还给出了详细的汇编语言上机指南。

书中凡以完整格式给出的程序均可直接使用,但须注意在向计算机内录入程序时,所有标点符号均应为西文符号;而书中为了显示清晰,将程序代码中的逗号、引号和分号等改用了中文符号。

王丰编写了本书的第 1~4,6,8,10,11 章及附录;田红芳编写了第 5,12 章;邱俊杰编写了第 7 章;谭娟编写了第 9 章。全书由王丰统稿。此外,参加本书编写及资料整理的还有栾学德、谭树芹、潘明寒等。

本书编写过程中,得到了北京航空航天大学出版社教材编辑室的大力支持和指导,在此一并表示衷心感谢。

由于作者水平有限,书中如有错误和不足,恳请广大读者批评指正。

本书配有电子教案及教学大纲等教学材料,如有需要请发送邮件到 bhkejian@126.com,或者致电 010-82317027,向北京航空航天大学出版社申请索取。

作 者

2007 年 1 月

目 录

第 1 章 微型计算机的组成及工作原理	1
1.1 编码与运算规则	1
1.1.1 无符号数与带符号数	1
1.1.2 补码的表示范围、运算规则及溢出	2
1.1.3 ASCII 码与 BCD 码	3
1.2 微型计算机系统及其工作原理	4
1.2.1 微型计算机的发展及主要技术指标	4
1.2.2 硬件系统结构	5
1.2.3 三类语言	8
1.2.4 微型计算机的工作原理	10
1.3 PC 机的编程结构	11
1.3.1 8086 CPU 的内部结构	11
1.3.2 8086/8088 的寄存器结构	13
1.3.3 内存储器基础	16
1.3.4 内存分段	18
习 题	20
第 2 章 指令系统	22
2.1 寻址方式	22
2.1.1 指令格式及操作数类型	22
2.1.2 8086/8088 的寻址方式	23
2.2 8086/8088 指令系统	27
2.2.1 数据传送指令	27
2.2.2 算术运算指令	33
2.2.3 逻辑运算及移位指令	40
2.2.4 转移控制指令	49
2.2.5 处理器控制指令	53

2.2.6 简单程序设计举例	53
习 题	56
第3章 MASM 伪指令系统	59
3.1 汇编语言语句种类与格式	59
3.1.1 汇编语言语句的种类	59
3.1.2 汇编语言语句的格式	60
3.2 常用伪指令	62
3.2.1 符号定义伪指令	62
3.2.2 数据定义伪指令与变量	63
3.2.3 标号及其属性	68
3.2.4 析值/属性操作符	68
3.2.5 段定义伪指令	70
3.3 程序的段结构	73
习 题	76
第4章 汇编语言程序设计	78
4.1 汇编语言软件开发步骤及输入/输出问题	78
4.1.1 汇编语言软件开发步骤	78
4.1.2 汇编语言编程时的输入/输出问题	78
4.2 顺序程序设计	81
4.2.1 字符串的输入/输出	81
4.2.2 查 表	82
4.3 分支程序设计	83
4.3.1 基本分支程序设计	83
4.3.2 跳转表法实现分支	84
4.4 循环程序设计	85
4.4.1 循环程序的结构	86
4.4.2 循环程序设计	86
4.5 子程序设计	87
4.5.1 子程序的定义、调用与返回	87
4.5.2 子程序设计的相关技术	90
习 题	93

第 5 章 8086 微处理器	95
5.1 8086 微处理器工作模式与引脚功能	95
5.1.1 8086 的工作模式	95
5.1.2 两种工作模式下共用引脚	95
5.1.3 最小模式引脚	97
5.1.4 最大模式引脚	98
5.1.5 8088 与 8086 的差别	98
5.2 系统组成	99
5.2.1 锁存与收发	99
5.2.2 最小模式下的系统组成	101
5.2.3 最大模式下的系统组成	102
5.3 8086 的总线操作与时序	103
5.3.1 指令周期、总线周期与 T 状态	103
5.3.2 8086 总线读/写周期	103
习 题	108
第 6 章 半导体存储器	109
6.1 概 述	109
6.1.1 半导体存储器的分类与技术指标	109
6.1.2 静态读/写存储器芯片介绍	111
6.2 CPU 与存储器的连接	112
6.2.1 内存容量的扩展方法	112
6.2.2 存储器的译码	115
6.2.3 8088 与存储器连接实例	117
6.2.4 8086 与存储器的连接	118
习 题	121
第 7 章 输入/输出技术	123
7.1 有关接口的基本概念	123
7.1.1 接口的概念与功能	123
7.1.2 接口信号的分类	124
7.1.3 端口地址及编址方式	124
7.1.4 8086/8088 的 I/O 指令	125
7.1.5 端口地址的译码问题	126

7.2 接口数据的传送方式	127
7.2.1 无条件传送方式	127
7.2.2 查询传送方式	128
7.2.3 中断传送方式	130
7.2.4 DMA 传送方式	131
习 题	132
第 8 章 中 断	133
8.1 中断概述	133
8.1.1 中断的概念	133
8.1.2 中断源	133
8.1.3 中断类型	134
8.1.4 中断类型号	135
8.1.5 中断向量表	135
8.2 中断处理过程	136
8.2.1 中断请求	136
8.2.2 中断响应	138
8.2.3 中断处理	139
8.2.4 中断返回	140
习 题	141
第 9 章 并行接口芯片 8255A	142
9.1 8255A 概述	142
9.1.1 8255A 的主要特性	142
9.1.2 8255A 的内部结构	142
9.1.3 8255A 的引脚	143
9.2 8255A 的控制字	144
9.2.1 工作方式控制字	144
9.2.2 C 口按位置位/复位控制字	145
9.3 8255A 的工作方式	146
9.3.1 方式 0:基本输入/输出	146
9.3.2 方式 1:带选通的输入/输出	147
9.3.3 方式 2:带选通的双向输入/输出	149
9.4 8255A 的编程	150

习 题.....	153
第 10 章 串行通信	155
10.1 串行通信概述.....	155
10.1.1 串行通信的特点.....	155
10.1.2 串行通信的基本概念.....	156
10.2 RS-232C 串行接口标准	159
10.2.1 电气特性及传输距离.....	160
10.2.2 接口信号.....	161
10.2.3 通信连接.....	162
10.3 串行通信接口芯片 8251A	163
10.3.1 8251A 的内部结构和引脚	163
10.3.2 8251A 的连接使用	166
10.3.3 8251A 的控制字	167
10.3.4 8251A 的初始化与编程	170
习 题.....	174
第 11 章 其他常用接口	175
11.1 定时/计数器 8253	175
11.1.1 8253 定时/计数器概述	175
11.1.2 8253 的工作方式	177
11.1.3 8253 的控制字与编程	180
11.2 D/A 接口	182
11.2.1 D/A 转换概述	182
11.2.2 D/A 转换器 DAC0832	184
11.3 A/D 接口	187
11.3.1 A/D 转换概述	187
11.3.2 A/D 转换芯片 ADC0809	188
11.4 人机交互设备.....	191
11.4.1 键 盘.....	191
11.4.2 LED 显示器	193
11.4.3 打印机接口.....	196
习 题.....	199

第 12 章 总 线	200
12.1 总线概述	200
12.1.1 总线的分类、指标、标准、意义	200
12.1.2 总线数据传输过程	202
12.2 系统总线	204
12.2.1 系统总线的特点及发展	204
12.2.2 IBM PC 总线	205
12.3 局部总线	207
12.3.1 局部总线概述	207
12.3.2 VL 总线	207
12.3.3 PCI 总线	208
12.4 常用外部总线	210
12.4.1 外部总线概述	210
12.4.2 USB 总线	210
12.4.3 IEEE 1394 总线	212
习 题	213
附 录	214
附录 A ASCII 码表	214
附录 B 汇编语言上机指导	215
参考文献	231

第 1 章 微型计算机的组成及工作原理

计算机分为巨型机、大型机、中型机、小型机和微型机等多种类型,尽管其组织结构差别较大,但基本工作原理是相同的。微型机结构最简单且应用最广泛,故本章以微型机为例介绍计算机的组成及工作原理。

1.1 编码与运算规则

日常生活中的各类数据在计算机内表示时,必须首先用二进制数将其编码。对不同特点的数据可以采用不同的编码方案。下面简要介绍常用的编码方案。

1.1.1 无符号数与带符号数

1. 无符号数

若计算机要处理的数据全为正数,在计算机内表示时可以使所有的二进制位都表示数值,这种数称为无符号数,其机内结构如图 1-1 所示。



图 1-1 无符号数的结构

例如:100 表示成 8 位无符号数为 0110 0100;255 表示成 8 位无符号数为 1111 1111。

无符号整数有一定的表示范围。在无符号数中,最高位也表示数值。8 位二进制数可表示 $2^8=256$ 个数,因此,8 位无符号数可表示 256 个数,范围为 $0 \leq N \leq 255$ 。同理可推得 16 位无符号数可表示 65 536 个数,范围为 $0 \leq N \leq 65\,535$ 。 n 位无符号数可表示 2^n 个数,范围为 $0 \leq N \leq 2^n - 1$ 。

2. 带符号数

对于有正有负的数,计算机内必须表示出数的符号,一般用机器数的最高位表示符号,其余位表示数值部分,如图 1-2 所示。对数值部分,可以直接以绝对值的形式表示,也可以进行各种变换,为此产生了原码、反码、补码等多种表示方案;但考虑到运算的方便性,一般表示成补码形式。

原码:机器数的最高位作为符号位,0 表示正数,1 表示负数,后面的各位表示数的绝对



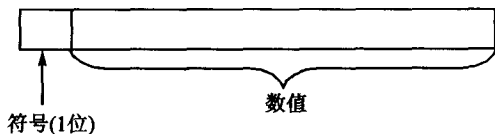


图 1-2 带符号数的结构

值。例如： $[+3]_{\text{原}} = 00000011$ ， $[-3]_{\text{原}} = 10000011$ 。

反码：正数的反码与其原码相同，负数的反码是由其对应的正数连同符号位按位取反求得。例如： $[+9]_{\text{反}} = 00001001$ ， $[-9]_{\text{反}} = 11110110$ 。

补码：正数的补码与其原码相同，负数的补码是其反码加 1。例如： $[+9]_{\text{补}} = 00001001$ ， $[-9]_{\text{补}} = 11110111$ 。

1.1.2 补码的表示范围、运算规则及溢出

1. 补码的表示范围

8 位补码中，最高位表示符号，有 7 位表示数值。7 位数值位可表示 $2^7 = 128$ 个数，加上最高位可正可负，故可表示正负各 128 个数，总计为 256 个数，范围为 $-128 \leq N \leq +127$ 。正数的最大值达不到 +128，是因为 0 占用了正数的一种表示形式。

同理，可推得 16 位补码能表示 65 536 个数，范围为 $-32\,768 \leq N \leq +32\,767$ 。 n 位补码能表示 2^n 个数，范围为 $-2^{n-1} \leq N \leq +2^{n-1} - 1$ 。

2. 补码的运算规则

补码的加减法运算比较简单(乘法运算复杂一些，本书不作介绍)。

补码的加法规则是： $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$ ；

补码的减法规则是： $[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$ 。其中的 $[-Y]_{\text{补}}$ 只要对 $[Y]_{\text{补}}$ 求补就可得到。

例 1.1 用 8 位补码完成下列加法运算。

① $23+36$ ；② $36+(-23)$ ；③ $23-36$ 。

$$\begin{array}{r} 00010111 \text{ (23)} \\ + 00100100 \text{ (36)} \\ \hline 00111011 \text{ (59)} \end{array}$$

①

$$\begin{array}{r} 00100100 \text{ (36)} \\ + 11101001 \text{ (-23)} \\ \hline \text{(进位1) } 00001101 \text{ (13)} \end{array}$$

②

$$\begin{array}{r} 00010111 \text{ (23)} \\ + 11011100 \text{ (-36)} \\ \hline 11110011 \text{ (-13)} \end{array}$$

③

说明：

① 在第②小题“ $36+(-23)$ ”的运算中，两个 8 位补码运算后的结果为 9 位二进制数(即最高位向前有一个进位)。由于机器字长的限制，最高进位位会自动丢失，运算结果只有低 8 位二进制数，但这并不影响运算结果的正确性。这是补码编码原理上的巧妙之处。事实上，

若不丢失最高进位位,结果反而不正确。

② 补码减法是用对减数求补后,把减法转换为加法进行的,它能自动得到正确的结果。

3. 补码的溢出

补码只能表示一定范围内的数。对于8位补码来说,其范围是 $-128 \sim +127$ 。若参加运算的两个数都在此范围之内,但运算结果超出此范围,则这时运算结果不正确。这称为溢出。

例如: $64+67=131$,但这两个补码相加的结果为10000111,其对应的真值为 -121 ,显然出错了。错误原因是 $+131 > +127$ 。这种超过上限的溢出称为正向溢出。又例如: $(-64)+(-67)=-131$,但补码相加的结果为101111001,最高位的进位自然丢失,机器中实得结果为01111001,其对应的真值为 $+121$,显然不对。错误原因是 $-131 < -128$ 。这种超过下限的溢出称为负向溢出。

防止溢出的办法是使用更多位数的补码进行运算。在上面的例子中,将参加运算的数64,77, -64 和 -67 改用16位补码表示,再进行运算,则运算结果无溢出。这是因为16位补码能表示 $-32\,768 \sim +32\,767$ 之间的数,而 $+131, -131$ 均在此范围之内。

4. 符号扩展

当将8位补码改为16位补码时,并不需要根据补码的定义重新求一遍16位补码,而只需将已有8位补码作为16位补码的低8位,再用8位补码时的符号位(即 D_7 位的值)填充16位补码的高8位即可。将较少位数的补码通过前面填充符号位的方法变为较多位数补码的方法称为符号扩展。

例如:

-2 的补码(8位)为11111110; -2 的补码(16位)为11111111 11111110;

$+2$ 的补码(8位)为00000010; $+2$ 的补码(16位)为00000000 00000010。

1.1.3 ASCII码与BCD码

1. ASCII码

计算机中除了要处理各种数据外,还要处理字符。字符在计算机内也必须用二进制数编码表示,计算机内字符的编码采用美国国家信息交换标准字符码,即ASCII码。

ASCII码以ASCII码表的形式提供,具体见本书附录A。ASCII码表的下列特点请读者注意掌握。

① 数字 $0 \sim 9$ 的ASCII码用十六进制表示为 $30H \sim 39H$,即任一数字字符的ASCII码=该数字值 $+30H$ 。

② 小写字母 $a \sim z$ 的ASCII码表示成十六进制为 $61H \sim 7AH$,而大写字母 $A \sim Z$ 的ASCII码表示成十六进制为 $41H \sim 5AH$ 。同一字母的大小写ASCII码不同,且小写字母的ASCII码比大写字母的ASCII码大 $20H$ 。



2. BCD 码

十进制是人们常用的数制,但计算机内只能处理二进制数,为此需要在输入时将十进制数转换为二进制数,然后才能进行运算;而在输出运算结果时又要将二进制数再转换为十进制数。当输入/输出的数据量很大,而要求的运算却很简单时,这种转换过程使得系统的绝大部分时间都浪费在了两种数制间的转换上,因而效率很低。为此,可以考虑不采用将十进制数转换为等值的二进制数的编码方案,而是将十进制数用二进制数直接表示出来,这称为二进制编码的十进制数,简称 BCD 码。

BCD 码有多种编码方法,最常用的是 8421BCD 码。编码方案为:将十进制数的每一位数字直接用等值 4 位二进制数表示。例如:十进制数 39 表示为 00111001,十进制数 90 表示为 10010000。

由于 4 位二进制数有 16 种组合,而 1 位十进制数字只有 0~9 十种状态,故 8421BCD 码中二进制数不允许出现 1010~1111 这 6 种组合。

在上面的方案中一个字节可表示两位十进制数,这种 BCD 码称为组合型 BCD 码。还有一种非组合型 BCD 码,它用一个字节表示一位十进制数字,高 4 位为 0,低 4 位用于表示对应的十进制数字。例如:39 表示为 0000001100001001,需占两个字节。

BCD 码表示法中,十进制数与二进制数的转换过程比较简单,但编码后的二进制数不能直接按原二进制数的运算规则进行运算,必须在二进制数运算的基础上再进行调整操作,因而增加了运算的复杂性,只适用于数据运算比较简单的情況。

1.2 微型计算机系统及其工作原理

1.2.1 微型计算机的发展及主要技术指标

1. 微型计算机的发展

微型计算机是计算机所有分类中发展最快、应用最广泛的一类。从 1971 年 Intel 公司推出 4004 微处理器芯片以来,短短 30 多年的时间里,微型计算机的发展已经历了 4 位机、8 位机、16 位机、32 位机和 64 位机五代。

微型计算机的性能主要由 CPU 决定,人们通常以 CPU 一次能处理的二进制数的位数(即计算机的字长)作为微型计算机的分类依据,字长为几位就称为几位机。其中,Intel 4004 为 4 位机,Intel 8086/8088 为 16 位机,80386,80486,Pentium 为 32 位机。

2. 微型计算机的主要技术指标

评价一台微型计算机系统性能好坏的主要技术指标如下。

(1) 机器字长

字长是指 CPU 能够一次处理的二进制数的位数。字长越长,多字节数据运算越方便。



通常,微型计算机的字长都是字节的整数倍,如8位、16位、32位、64位。如8086为16位字长;80386,80486为32位字长。

(2) 主 频

微型计算机的时钟频率(通常称主频)在一定程度上反映了机器的运算速度,主频越高,运算速度越快。主频的单位是MHz(兆赫[兹])。如386的主频为20~33MHz,486的主频为25~100MHz,Pentium的主频为66~200MHz,Pentium III的主频为336~1400MHz。

(3) 内存容量

内存容量指微型计算机可存储信息的多少,通常以字节为计量单位。内存容量越大,储存的程序和数据就越多,处理能力也就越强。目前微机使用的内存容量为256~1024MB不等。

(4) 运算速度

运算速度的一种表示方式是MIPS,即每秒百万条指令,它主要是对整数运算而言。对于浮点运算,一般使用MFLOPS表示,即每秒百万次浮点运算。

1.2.2 硬件系统结构

一个完整的计算机系统由硬件和软件两大部分组成。本小节介绍硬件系统的组成及其工作原理。

1. 硬件的逻辑组成部分

计算机的硬件,不论是微型计算机,还是其他各类计算机,从逻辑功能上可划分为五个部分,分别是:运算器、控制器、存储器、输入设备和输出设备,如图1-3所示。

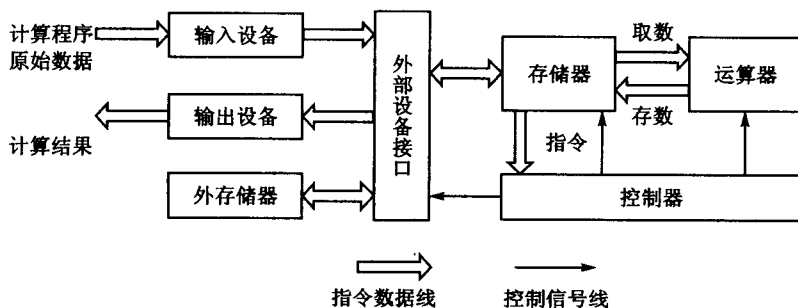


图 1-3 计算机的硬件组成

各部分的功能如下。

① 运算器:核心部件,负责对信息进行处理和运算。经常进行的运算是算术和逻辑运算,因此运算器的核心是算术逻辑部件,简称ALU。

② 控制器:是整个计算机的指挥中心,它的主要功能是按照人们预先确定的操作步骤,控

制整个计算机有条不紊地工作。控制器从内存中逐条取出指令进行分析,根据指令的要求来安排操作顺序,向计算机的各部件发出相应的控制信号,控制它们执行指令所要求的任务。

③ 存储器:用来存放数据及程序,是计算机的记忆装置。

④ 输入设备:输入设备的任务是把人们编制好的程序和原始数据送到计算机中去,并且把它们转换成计算机内部所能识别和接收的信息方式。最常用的输入设备是键盘和鼠标。

⑤ 输出设备:输出设备的任务是将计算机的处理结果以人或其他设备所能接受的形式送出计算机。最常用的输出设备是显示器和打印机。

在上述五部分中,由于运算器与控制器的关系密切,所以人们直接把这两部分做在一块集成电路芯片上,这样的芯片称为微处理器,简称 CPU。

计算机的存储器实际上又可分为内存和外存两部分。上面所说的存储器指的是内存,它位于机箱内部,由半导体器件构成。而像磁盘、光盘等则称为外存,外存上的数据不能被 CPU 直接使用,必须先读入内存才可使用。在学习计算机原理时,主要关心的是内存的使用方法及组织结构。

习惯上,人们常把输入设备、输出设备、外存储器统称为外部设备,简称外设或 I/O 设备;把运算器、控制器和内存存储器统称为主机。外设与主机之间的信息交换是通过外部设备接口(简称 I/O 接口)实现的,而且不同的外设使用不同特点的 I/O 接口。

2. 三类总线

计算机的硬件由五个逻辑部分组成,而每个逻辑部分又分别由多个电路来实现,在将各个部件连接成一台计算机时,要保证信号能在部件间方便地传递,而且当系统增减部件时又不影响其他部件。总线结构提供了解决这一问题的良好方案,各部件间通过总线连接的基本情况如图 1-4 所示。

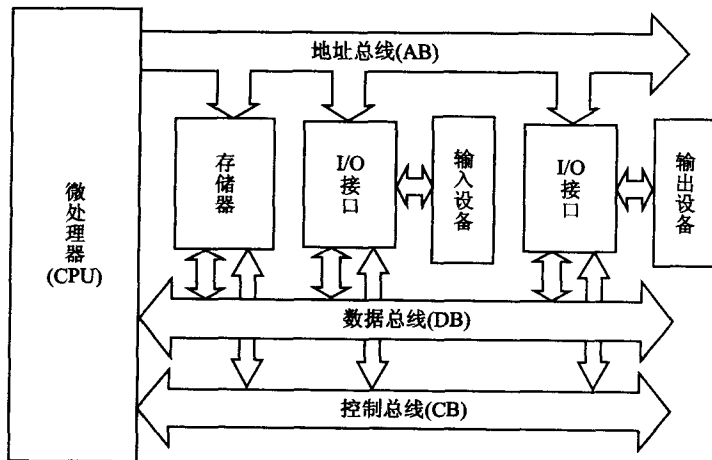


图 1-4 三类总线