

自动化

专业本科系列教材

Weiji Yuanli yu Jiekou Jishu

微机原理与接口技术

李捍东 主 编

罗文广 副主编

重庆大学出版社

微机原理与接口技术

李捍东 主 编
罗文广 副主编

重庆大学出版社

内 容 简 介

本书以 Intel8086 微处理器为主线,介绍了微型计算机原理及接口技术。主要包括微型计算机基础及 CPU 原理、8086 微处理器及其指令系统、8086 汇编语言程序设计、存储器、接口与总线技术、接口程序及技术。鉴于目前微型计算机已广泛采用 32 位微处理器,本书还充实了 IA—32 微处理器方面的知识。

本书力求条理清晰、语言流畅、深入浅出,又不失系统性和先进性。本书系高等院校自动化专业本科教材,也可作为非计算机专业的本专科生教材。

图书在版编目(CIP)数据

微机原理与接口技术/李捍东主编. —重庆:重庆大学出版社,2004. 10

(自动化专业本科系列教材)

ISBN 7-5624-2833-6

I. 微... II. 李... 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(2004)第 065289 号

微机原理与接口技术

李捍东 主 编

罗文广 副主编

责任编辑:谭 敏 版式设计:谭 敏

责任校对:廖应碧 责任印制:秦 梅

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400030

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(市场营销部)

全国新华书店经销

重庆科情印务有限公司印刷

*

开本:787×1092 1/16 印张:21 字数:524 千

2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

印数:1—4 000

ISBN 7-5624-2833-6/TP·388 定价:28.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有 翻印必究

前言

微型计算机技术经历了 20 多年日新月异的发展,继 4 位、8 位、16 位微处理器后,相继出现了 32 位乃至 64 位微处理器。目前广泛使用的 Intel Pentium 系列微处理已使用了 32 位结构(简称 IA—32)。IA—32 虽然比起早期的 16 位微处理器 8086(简称 IA—16)在许多方面都有极大的提高,但仍然保持与 8086 的高度兼容,因此 8086 微处理器是系统学习微机原理及接口技术的基础。

本书以 Intel8086 微处理器为主线,系统介绍了微型计算机原理及接口,同时也对 IA—32 计算机的关键技术做了详细地介绍。目前,《微机原理与接口技术》已作为自动化专业的一门重要课程,是学习后继课程《计算机控制技术》、《单片机原理及接口技术》的一门重要基础课。

本书是笔者多年从事微型计算机教学和科研工作的总结,在本书编写过程中也对当前国内外有关微型计算机的大量资料进行了提炼和总结。

本书分为 4 个部分,共 10 章。第一部分是微型计算机基础及 CPU 原理部分,包括第 1 章和第 2 章,重点介绍了微型计算机基础知识,计算机中信息的表示方式以及算术逻辑运算在计算机中的实现方法,并以初级计算机为基础介绍了 CPU 内部结构、工作过程、指令和控制单元,同时简要介绍了当前 CPU 中采用的一些先进技术;第 2 部分以 8086 微处理器为基础,介绍了 8086 内部结构和有关知识,8086 指令系统以及汇编程序设计,内容包括第 3 章、第 4 章和第 5 章;第 3 部分涉及存储器和接口技术方面的内容,包括第 6 章、第 7 章、第 8 章和第 10 章,主要介绍存储器结构及体系、芯片级接口技术、中断系统、总线技术、设备接口程序等,此外还介绍了当前微型计算机中广泛采用的先进存储器和总线技术、Windows 系统下的设备驱动等;第 4 部分(第 9 章)主要介绍目前广泛使用的 IA—32(如 Pentium 系列)微处理器的内部结构、指令系统、IA—32 的保护方式及其编程。

本书由李捍东担任主编,罗文广任副主编,参编何志琴、龚德红、徐棱桦、张发斌。其中,第1章1.1由何志琴编写,第1章1.2、第2、3、6、8、9章由李捍东编写,第4、5章、第10章10.2及附录由罗文广编写,第7章、第8章8.4由龚德红编写,第8章8.5由张发斌编写,第10章10.1、10.3、10.4由徐棱桦编写,全书由李捍东统稿,此外龚德红做了大量的图形绘制工作。

本书的出版得到了重庆大学出版社编辑与领导的大力支持,并参考了国内外有关教材及著作,在此一并表示衷心感谢。由于编者水平有限,书中难免有疏漏和不当之处,敬请读者提出宝贵意见,以便再版时修订。

编者

2004年3月

目 录

第 1 章 微型计算机基础	1
1.1 概述	1
1.2 计算机中信息的表示方法	7
习题	18
第 2 章 CPU 原理	19
2.1 概述	19
2.2 算术逻辑部件 ALU	20
2.3 CPU 运算方法	23
2.4 初级计算机结构及原理	33
2.5 指令系统	40
2.6 组合逻辑控制器	48
2.7 微程序控制器	49
2.8 提高 CPU 性能的措施	51
习题	54
第 3 章 8086 微处理器	55
3.1 8086 内部结构	55
3.2 8086 主存储器和堆栈	60
3.3 8086 工作模式和外特性	61
3.4 8086 总线及操作时序	67
习题	71
第 4 章 8086 指令系统	72
4.1 8086 寻址方式	72
4.2 8086 指令系统	79
习题	96
第 5 章 汇编语言程序设计	99
5.1 概述	99
5.2 汇编语言的表达式	101
5.3 伪指令	104
5.4 宏指令	108
5.5 汇编语言程序设计	111

5.6	实用程序设计举例	120
5.7	汇编语言程序的开发过程	123
	习题	130
第6章	存储器	132
6.1	存储器概述	132
6.2	半导体存储器	133
6.3	主存储器组织	140
6.4	存储器系统结构	148
6.5	存储器的研究与发展	155
	习题	156
第7章	接口技术与芯片级接口	158
7.1	接口技术	158
7.2	可编程并行接口	170
7.3	可编程串行接口	179
7.4	中断系统及中断控制器	187
7.5	DMA 控制器	208
7.6	常见外围芯片	215
	习题	229
第8章	总线技术	232
8.1	总线概述	232
8.2	系统总线	235
8.3	局部总线	236
8.4	设备总线	240
8.5	现场总线	251
	习题	255
第9章	IA—32 微处理器结构及编程	256
9.1	IA—32 微处理器概述	256
9.2	Pentium 微处理器	257
9.3	IA—32 微处理器指令系统	268
9.4	保护方式下的 IA—32 微处理器及其编程	278
9.5	编程实例	304
9.6	高性能 Pentium 处理器和具有 MMX™ 技术的 Pentium 处理器	309
	习题	310
第10章	接口程序及技术	311
10.1	接口(驱动)程序概述	311
10.2	DOS 系统的接口程序及实现	315

10.3 Windows 操作系统下的接口程序	322
10.4 Windows 软件接口技术	324
习题	325
附录	327
参考文献	328

第 1 章

微型计算机基础

计算机是一种能够自动、高速、精确地进行信息处理的现代化电子设备,并具有存储与记忆能力,由程序控制着计算机的运行,广泛应用于人类社会的各个领域。

本章主要介绍微型计算机的基本组成、工作过程、特点及性能指标,以及计算机中信息的表示方法等基本知识。其目的主要是对微型计算机有一个初步的了解,为学习以后的内容打下基础。

1.1 概 述

1.1.1 计算机基本概念

现在广泛使用计算机,其全称是数字电子式计算机,俗称电脑。简单地讲,计算机是一种能够存储程序,并能自动连续地执行程序,对各种数字化信息进行运算的现代化电子设备。

首先,计算机是能够进行各种运算的设备,运算可分为两类:算术运算和逻辑运算。算术运算的对象是数值型数据,以四则运算为基础,许多复杂的数学问题都可以通过各种算法转换成若干四则运算;逻辑运算用来解决逻辑问题,如信息检索、逻辑判断和分析等。因此,计算机的工作实际上就是对各种信息进行处理。

其次,计算机如何表示这些信息呢?简单来说,是用数字代码(即二进制数)来表示各种信息,因此称为数字计算机。

最后,计算机如何对这些信息进行处理。它采用的是一种存储程序工作方式,即先编写程序,再由计算机将这些程序存储起来,然后自动连续地、快速地执行程序,从而实现各种运算处理。

为了存储程序与数据,需要存储器;为了进行运算,需要运算器;为了输入程序和数据以及输出运算结果,需要有输入和输出设备;此外还需要控制器对计算机各个部件的工作进行控制和管理。

上述要领是由计算机技术的先驱冯·诺依曼首先提出的,他在 1945 提出了数字计算机的若干思想,后被称为冯·诺依曼体制,这是计算机发展史上的一个里程碑。几十年来计算机的

体系结构发生了深刻的变化,但冯·诺依曼体制的核心概念仍沿用至今。冯·诺依曼体制的要归纳如下:

- 采用二进制代码表示数据和指令。
- 采用存储程序工作方式,即先编写程序,然后存储程序,最后自动连续执行程序。
- 由运算器、存储器、控制器、输入设备和输出设备组成计算机硬件系统。

下面首先阐述其中两点:存储程序工作方式、信息的数字化表示。

1. 存储程序工作方式

计算机的工作最终体现为执行程序。前面提到,计算机采用存储程序工作方式,这是冯·诺依曼体制最核心的思想。它有3点含义,体现了计算机解决问题的过程。

(1) 编写程序

为了使用计算机解决问题,需要先编写程序。在程序中规定了计算机需要做哪些工作,按什么步骤去做。程序还包括需要处理的原始数据,此外还规定计算机什么时候从输入设备去获取数据。一件事情一般要分成几步来完成,每步执行的操作命令称为一条指令。计算机最终执行的程序是一系列指令序列,即若干指令的有序集合。换句话说,我们事先编写的程序最终变成指令序列和原始数据。

(2) 存储程序

编写完成的程序经输入设备送入计算机,存放在存储器中。编写程序时是用字符书写的,通过键盘将字符变成二进制编码,然后再输入计算机。

(3) 自动、连续地执行程序

由于程序已存储在存储器中,启动计算机后,计算机就可以按照一定顺序自动从存储器中逐条读取指令,按照指令的要求完成相应的操作,直到程序被执行完毕。原则上,程序在执行过程中不需要人工干预。当然有些工作本身需要以人机对话形式进行,例如通过计算机进行查询时,计算机通过屏幕向操作人员询问,操作人员通过键盘或鼠标进行选择。这种情况要求计算机能分段执行程序,中间允许用户进行人工干预。所以计算机在自动、连续执行程序的过程中,往往允许使用者以外部请求方式进行干预。

上述计算机的基本工作方式,称为控制流驱动方式。在程序执行过程中,根据指令发出的控制信息成为一种控制信息流,简称控制流,它驱动着计算机的工作;而处理的数据信息成为一种数据信息流,简称数据流,它是被调用(处理)的对象。

2. 信息的数字化表示

上面讲到,现在广泛使用的计算机,其全称是电子式数字计算机。

“电子”是指计算机的主要部件是由电子电路组成,计算机内传送与处理的信息是电子信号。例如,计算机中的算术运算单元(ALU)主要由加法器构成,而加法器由各种门电路(与门、非门等)组成。

“数字”则表示计算机中的信息(控制信息和数据信息)均采用数字化表示方法。例如,二进制11001表示-9,01000001表示字符‘A’,0001表示加法指令。

1.1.2 微型计算机组成

计算机是依靠硬件和软件的协同工作来执行一给定任务,其中硬件(Hardware)是构成计算机的各种实体的总称,如集成电路芯片、内外存储器、输入输出设备、电源等,它是计算机的

物质基础。软件(Software)是计算机可运行的全部程序和有关信息的总和,其作用是有效地发挥和扩大计算机的功能。硬件和软件是相互依存的两个部分,两者缺一不可。

1. 硬件系统

根据冯·诺依曼体制,现代计算机的硬件可分为 4 个部分:运算器、控制器、存储器和输入输出设备。这 4 部件之间可通过不同方式联系起来,可采用直接通路的连接方式,如图 1.1 所示,也可采用总线的连接方式,如图 1.2 所示。

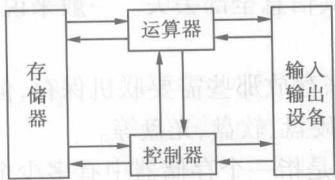


图 1.1 直接通路的连接方式

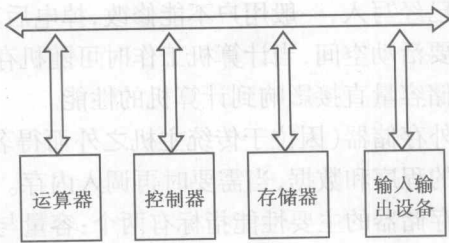


图 1.2 总线的连接方式

(1) 运算器

运算器是对数据进行加工处理的部件。无论计算机具有多强的功能,能处理多么复杂的问题,但运算器本身进行数据处理的功能只有最基本的二进制运算,如加、减、乘、除四则运算以及与、或、非等逻辑运算,比较复杂的运算需要通过程序来实现。

运算器由两部分组成,一部分是算术逻辑单元 ALU,这是运算器的核心,它主要由并行加法器及其他逻辑运算部件和各种数据通路组成;另一部分是寄存器,用以存储参加运算的数据。其基本结构示意图如图 1.3 所示。参与运算的数据分别存储在 A、B 中,参加运算时,两者同时传送至算术逻辑单元中,其结果则送至指定的存放单元中。

运算器的性能是影响整个计算机系统性能的重要因素。一般而言,运算器所能并行处理二进制代码的位数通常称为机器的字长。位数的多少影响了计算机的精度,位数越多,所能处理的数值范围就越大,精度就越高。运算器进行基本运算的速度,将直接影响系统的速度。所以精度和速度是运算器的重要性能参数。

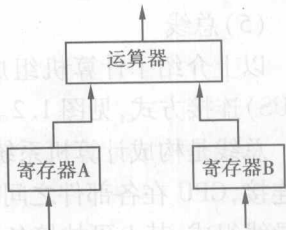


图 1.3 运算器结构示意图

(2) 控制器

控制器是控制计算机各个部件按照要求协调地工作的部件。其主要功能有：

- 1) 取出一条指令,并指出下一条指令在存储器的位置。
- 2) 对指令进行译码和分析,并产生相应的操作控制信号,以便启动相应的动作。如内存读/写操作、算术逻辑运算操作或输入/输出操作。

控制器由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器等组成。

运算器和控制器构成了计算机的中央处理器,即 CPU,是计算机系统的核心部分。

(3) 存储器

存储器是计算机系统记忆设备,是用来存放程序和数据的。其主要功能是保存大量的二进制代码,并可根据需要可从中读出代码或写入代码。程序和数据均是以二进制形式存

放的。

一般情况总希望存储器的容量越大、存取速度越高,但这往往是有矛盾的,所以常将存储器分成几级。

主存储器由半导体材料构成,工作速度较高,它主要用来存放当前 CPU 需要的程序和数据,由于它位于传统主机内,因此又称为内存储器(内存)。内存储器又分为只读存储器(ROM)和随机读写存储器(RAM)。ROM 用于存放系统初始化程序以及自检程序等,机器出厂时已经写入,一般用户不能修改,掉电后其信息不会丢失。RAM 是内存的主体,是程序运行的主要活动空间,当计算机工作时可随机存取信息,掉电后其信息全部丢失。一般来说,RAM 的存储容量直接影响到计算机的性能。

外存储器(因位于传统主机之外而得名,简称外存)用来存放那些需要联机保存,但暂不需要的程序和数据,当需要时再调入内存。外存储器主要有硬盘、软盘、光盘等。

存储器的主要性能指标有两个:容量与速度。存储容量是指一个存储器中有多少个存储单元,也就是能存放多少二进制信息。容量通常用存放字节(8 位二进制为一个字节,用 B 表示)的数量来表示,如 4 KB、64 MB、1 GB 等,其中 $\text{KB} = 1\,024\text{ B}$, $\text{MB} = 1\,024\text{ KB}$, $\text{GB} = 1\,024\text{ MB}$, $\text{TB} = 1\,024\text{ GB}$ 。存储器的速度一般用读写周期来衡量,它是指两次访问(读或写)存储器之间的最小时间间隔。

(4) 输入/输出设备

输入/输出设备也称为外围设备,又称为 I/O 设备,其功能是完成计算机系统与外部世界的信息传递。所谓外部世界,是指需要与计算机系统联系的人或物,如控制台、各种仪器设备、其他计算机等。操作人员必须通过各种设备与计算机联系,如键盘、打印机、显示器等。常用的输入设备有键盘、鼠标等,常用的输出设备有显示器、打印机、绘图仪等。其中,键盘和显示器是标准输入/输出设备。

(5) 总线

以上介绍了计算机组成的 4 大部件,它们之间的联系方式,目前广泛采用的就是总线(BUS)连接方式,见图 1.2。

总线是构成计算机系统的骨架,是多个系统部件之间进行数据传送的公共通路。通过总线连接,CPU 在各部件之间传送地址、数据和控制信息以实现对相应部件的操作。总线由若干导线组成,其上可挂接各个部件。按照功能可把总线分为 3 类:地址总线、数据总线和控制总线。

当今,微机中的总线都采用了国际标准化总线,目前广泛采用 PCI 总线标准。

2. 软件系统

简单地说,计算机软件就是程序和相关信息的总和,其中程序规定了计算机如何去完成某一任务,是某种算法的体现;相关信息包括系统说明、帮助信息和其他参考信息。从软件配置与功能角度,可将软件分为系统软件和应用软件两大类。

(1) 系统软件

这是一组为使计算机系统良好运行而编写的基础软件。从软件配置角度看,它是计算机系统的一部分,是管理、监控和维护计算机资源的软件,是支持应用软件运行的软件。它通常为用户开发应用系统提供一个平台,用户可以在其基础上操纵和管理计算机。一般常用的系统软件有:

1) 操作系统 OS(Operating System)

为了使计算机系统的所有资源(包括中央处理器、存储器、各种外部设备及各种软件)协调一致,有条不紊地工作,就必须有一组软件来进行统一管理和统一调度,这种软件称为操作系统。它的功能就是管理计算机系统的全部软硬件资源及数据资源,使计算机系统所有资源最大限度地发挥作用,为用户提供方便的、有效的、友好的人机界面。

任何计算机可以没有其他软件,但不能没有操作系统。计算机启动时需要运行的第一个软件就是操作系统,任何其他软件必须在操作系统的支持下才能工作。

随着计算机的发展,许多操作系统不断出现,目前广泛流行的有 Windows 系统、OS/2 系统、UNIX 系统、LINUX 系统等。

2) 程序语言

编写计算机程序所用的语言是人与计算机进行信息交流的工具,一般可分为机器语言、汇编语言和高级语言。

机器语言:机器语言是计算机系统惟一能识别的,不需要翻译直接供机器使用的程序设计语言。机器语言中的每一条语句(机器指令)实际是一条二进制形式的指令代码,它由二进制操作码和二进制操作数组成。它的二进制指令代码随 CPU 型号的不同而不同(同系列 CPU 一般向上兼容)。显然用机器语言编写程序非常不方便,阅读、修改困难,因此通常不采用机器语言直接编写程序。

汇编语言:汇编语言是一种面向机器的程序设计语言,它是为特定计算机或计算机系列设计的。汇编语言采用一定的助记符号代替了二进制的机器指令,这种替代使得机器语言“符号化”,所以也称汇编语言为符号语言。每条汇编语言的指令就对应了一条机器语言的代码,不同型号的计算机系统一般有不同的汇编语言。

高级语言:高级语言是一种面向用户的,与特定机器属性无关的语言,它允许编程人员以习惯的自然语言(多为英语)和数学语言编写程序。相对于机器语言和汇编语言等“低级语言”,高级语言更接近于人们的语言和思维习惯,是人们开发各种应用软件的主要工具。常用的高级语言有 BASIC、FORTRAN、PASCAL、C 等。

无论是高级语言,还是汇编语言,CPU 都不能直接识别。用这些语言编写的程序称为源程序。必须将源程序转换成用机器语言表示的程序,计算机才能识别和执行,这种机器语言表示的程序称为目标程序。

执行高级语言程序有两种方式,一种方式是将源程序通过一个编译程序,将其翻译成目标程序,并将此目标程序提供给机器执行,高级语言的编译程序称为编译器,编译汇编语言的编译程序称为汇编器,每种语言都有它自己的编译程序;另一种方式是不生成目标程序,根据高级语言的语句逐条解释并执行,如早期的 BASIC 程序就是以解释方式执行的,完成这种功能的程序称为解释器。

3) 诊断程序

诊断程序主要用于对计算机系统硬件的检测。它可对 CPU、内存、显示器等硬件设备的性能和故障进行检测。

4) 数据库系统

数据库系统主要用于档案管理、财务管理、图书资料管理及仓库管理等的数据管理。这类数据的特点是数据量比较大,数据处理的主要内容是数据的存储、查询、修改、排序、分类、统

计、打印等,目前微型计算机上广泛使用的数据库系统软件有 FoxPro、ORACAL 等。

(5) 数据仓库

数据仓库是近年来迅速发展起来的一种存储技术,是面向主题的、集成化的、稳定的、随时时间变化的数据集合,用以支持决策管理的一个过程。数据仓库的主要服务对象是企业或机构中高层领导或决策人士,可提供分析型战略数据。显然,数据仓库的基础是数据库,但它不同于数据库,它存储大量的、决策分析所必需的、历史的、分散的、详细的操作数据,经过处理能将这些数据转换成集中统一、随时可用的信息。目前,许多主要的数据库厂商和软件厂商都已加入到数据仓库产品的开发中来。

(2) 应用软件

应用软件是指计算机用户利用计算机的软、硬件资源为某一专门的应用目的而开发的软件。由于计算机的应用领域极其广泛,无所不在,因此应用软件不胜枚举,一般包括科学计算、工程设计、数据处理、事务处理、工业控制等。

当然对系统软件和应用软件的划分不是一成不变的。比如一些具有通用价值的应用软件也可纳入系统软件之中,作为一种软件资源提供给用户。

1.1.3 计算机的工作过程

计算机的工作过程实质上就是执行程序的过程,而程序是由一系列指令组成的,因此执行程序的过程就是按顺序执行指令的过程。

通常,计算机要运行某个程序时,则该程序预先要调入到内存中的一系列单元中,在程序运行过程中完全由计算机自动执行而不要人工干预,具体包含下列几个步骤:

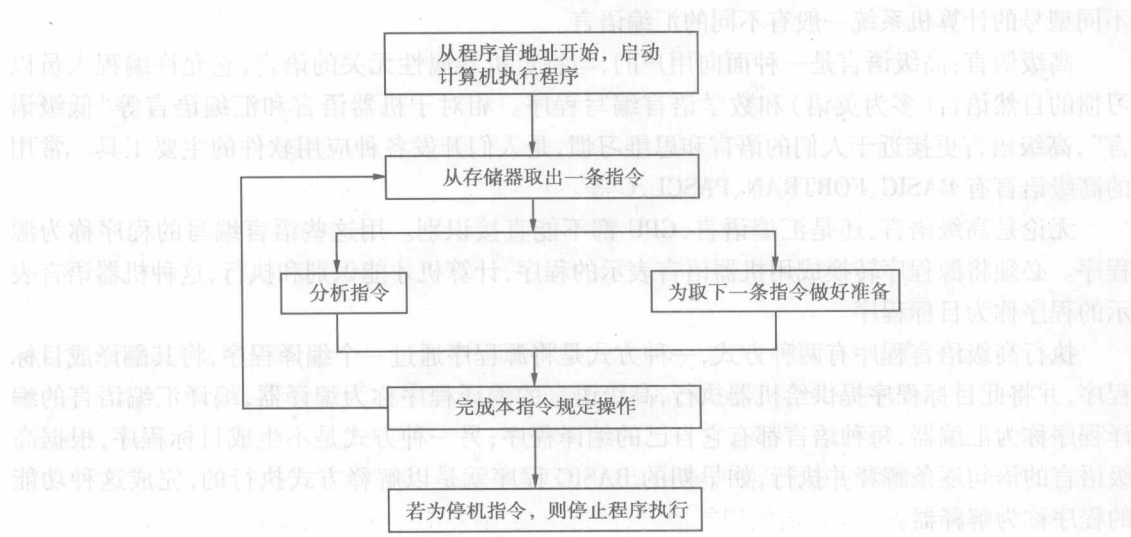


图 1.4 程序的执行过程

- (1) 取出指令 从存储器某个地址中取出要执行的指令送到 CPU 内部的指令寄存器暂时保存。
- (2) 分析指令 把指令寄存器中的指令送到指令译码器,分析出该指令对应的操作。
- (3) 执行指令 根据指令译码结果,向各个部件发出相应控制信号,完成指令规定的各种

操作。

(4)为执行下一条指令做好准备,即形成下一条指令地址。

具体过程如图 1.4 所示。

1.1.4 计算机系统的层次化结构

计算机作为一种工具,是一种复杂的软、硬件结合系统。为了能有效地组织计算机的各个部分,便于理解系统的构成,可将计算机系统分成若干个相互独立的功能层次,如图 1.5 所示。

计算机的纯硬件部分位于最底层,称为裸机。其功能是在程序的控制下自动完成计算机的物理操作。裸机是构成计算机系统物质基础,也是计算机可见的实体。

紧靠裸机层之上的是硬件的驱动程序,它直接控制着底层硬件的运行。驱动程序的存在,使得上层软件只要调用驱动程序就可以实现对硬件的功能调用,从而简化调用的过程。用户对硬件的功能调用不必了解其内部细节,也不必考虑硬件变化对功能的实现造成的影响。

位于驱动程序之上的是操作系统的功能管理程序。操作系统是所有软、硬件资源的控制者,并为用户提供一个应用环境,用户可通过操作系统提供的各种功能操纵计算机。

位于操作系统之上的是应用程序,其功能是完成具体的应用。

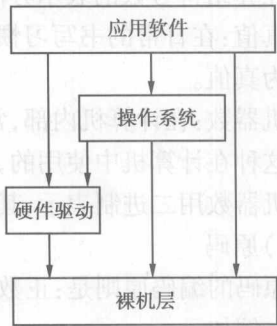


图 1.5 计算机系统的层次化结构

1.2 计算机中信息的表示方法

计算机中的信息有数据信息和控制信息两大类,且均采用二进制代码来表示。而数据又有数值型数据和非数值型数据,控制信息又称为指令信息。本节重点阐述数据和指令在计算机中的表示方法。

1.2.1 数值型数据

表示一个数值型数据需要 3 个要素:①采用什么数制? ②如何表示符号数? 也就是如何使符号数字化。③如何表示带小数点的数? 有两种方法,即定点数和浮点数。

1. 数制

数制有非进位制和进位制两种,其中非进位制的特点是:表示数值的大小与它在数中的位置无关。而进位制是当今普遍使用的数制,其特点是:表示数值大小的数码与它在数中所处的位置有关。例如,十进制数 123.45,数码 1 位于百位上,它代表 $1 \times 10^2 = 100$,则它所处的位置的权为 10^2 ,依此类推,数码 2 的权为 10^1 ,数码 3 的权为 10^0 ,数码 4 的权为 10^{-1} ,数码 5 的权为 10^{-2} 。

数据是计算机所处理的对象,在计算机中所有的数据都是以二进制的形式处理、存储和传送的,为了方便用户的输入输出或书写,经常用到十进制、八进制和十六进制。

2. 无符号数与符号数

(1) 无符号数

在数据处理的过程中,数据符号没有变化时,则不必设置符号位,用全部字长表示数值,凡数值不足定长的位,则高位可以补0。无符号数在计算机内直接用对应的二进制表示。由于符号数需要使用最高位来表示正负,因此,同样位数的无符号数的表示范围与符号数不同,如 $n=5$,符号数的表示范围为 $-16 \sim 15$,而无符号数为 $0 \sim 31$ 。

(2) 符号数

在介绍符号数的表示方法之前,首先必须明确几个基本概念。

真值:在日常的书写习惯中,经常用正负符号加上数的绝对值表示数值。这种形式表示的数称为真值。

机器数:在计算机内部,常用某一位二进制(最高位)来表示符号,如0表示正数,1表示负数。这种在计算机中使用的、连同符号位一起被数字化的数,称为机器数。

机器数用二进制表示,其位数与机器字长有关。符号数的编码形式有原码、反码和补码。

1) 原码

原码的编码原则是:正数的符号位用0表示,负数的符号位用1表示,其余的数值位保持不变。例如:

十进制数	+49	-49
二进制数	+0110001	-0110001
原码	00110001	10110001

可见,正数的原码是它本身,负数的原码就是符号位为1而数值位不变。0的原码有两个:00000000(+0)和10000000(-0)。

原码的特点是:简单且与真值转换方便,用于乘除法运算很方便,但对于加减法运算比较麻烦。为了简化运算规则,通常使用反码和补码。

2) 反码

反码的编码规则是:正数的反码表示与原码相同;负数的反码是,符号位为1,数值部分按位取反(0变为1,1变为0)。

例如:

二进制数	+1001110	-1001110
反码	01001110	10110001

0的反码也有两个:00000000和11111111

3) 补码

补码的编码规则为:正数的补码与原码相同;而负数的补码是,符号位为1,数值部分按位取反,并且最低位加1,即反码加1。

例如:

二进制数	+1001110	-1001110
补码	01001110	10110010

0的补码只有一个:00000000

补码在日常生活中也很常见。例如时钟,众所周知,时钟以12为计数循环,即以12为模。设当前时针指向10时,如需调整时间为6时,则调时间有两种方法:一是将时针倒拨4小时,

二是将时针顺拨 8 小时。可见,在以 12 为模的系统中,加 8 和减 4 都是一样,凡是减 4 的问题都可以用加 8 来替代,此时称 4 和 8 互为补数,这样就可以把减法问题转化为加法问题了。对于 n 位计算机,设 $n=8$,它所能表示的最大数为 $(11111111)_2$,相当于十进制数 255,若再加 1,成为 $(10000000)_2$,但因只有 8 位,最高位 1 将自然丢失。本来应该是 256,却回到了 0。所以 8 位二进制系统的模为 256。如同时钟一样,在这样的系统中,减法问题可化为加法问题,只要将减数用相应的补数表示就可以了。可见,把补数用到计算机对机器数的处理上,就是补码。

3. 定点数与浮点数

在计算机中对二进制小数的表示,根据约定的小数点位置是否固定,分为定点数和浮点数,它们关系到数的表示范围、精度以及电路复杂程度等。

(1) 定点数

定点数的小数点隐含固定在某个位置不变。采用定点表示法的计算机称为定点计算机。原则上,小数点固定在哪一位并不重要,一般总是把小数点放在数的最前面或数的最后面,即总是把所有的数化为纯小数或纯整数,选择哪一种在硬件上并无区别,只是在程序中约定的。

定点数的表示简单,运算电路也简单,但由于在字长一定的情况下,所表示的数的范围小,运算精度有限。

(2) 浮点数

浮点数的小数点的位置在数中可以变动。采用浮点表示法的计算机称为浮点机。

浮点表示法类似于科学表示法,任一数可通过改变其指数部分,其小数点位置随之发生移动,如 123.45 可以表示成 1.2345×10^2 , 0.12345×10^3 , 123.45×10^0 等不同形式,二进制浮点表示的一般形式为:

$$N = 2^E \times D$$

其中: E 称为阶码(含符号)

D 称为尾数(含符号)

2 是数制基数(隐含的,不必出现在数据表示中)

浮点数在计算机中的存放形式如下:

J	E_{n-1}	E_{n-2}	...	E_0	S	D_{m-1}	D_{m-2}	...	D_0
-----	-----------	-----------	-----	-------	-----	-----------	-----------	-----	-------

J 称为阶符,即指数部分的符号位;

$E_{n-1} \cdots E_0$ 称为阶码,表示幂次,可使用补码或移码表示;

S 称为数符,即尾数部分符号位;

$D_{m-1} \cdots D_0$ 尾数部分,可用补码或移码表示。

不同机器,其阶码和尾数各占用的位数都有具体规定。目前微型计算机广泛采用了 IEEE754 标准的浮点数格式,如表 1.1 所示。

表 1.1 IEEE754 标准浮点数格式

数据类型	数 符	阶 码	尾 数	总位数
单精度实数	1	8	23	32
双精度实数	1	11	52	64
双扩展精度实数	1	15	64	80