

Principle and Application
of Single Chip Computer

高等院校电子信息类
卓越工程师培养系列教材

单片计算机 原理及应用

尹 勇 撒继铭 编著
刘 涛 李京蔚

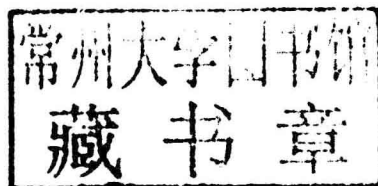


科学出版社

高等院校电子信息类卓越工程师培养系列教材

单片计算机原理及应用

尹 勇 撒继铭 编著
刘 涛 李京蔚



科学出版社

北京

内 容 简 介

“单片计算机原理及应用”是一门重要的技术基础课,它为电子信息技术领域的学习、工作和研究奠定计算机应用基础。近年来,国家启动了“卓越工程师教育培养计划”,为了配合该计划的实施,我们编写了本教材,希望能为卓越工程师的培养作出贡献。

全书共 11 章,首先介绍了微型计算机的基础知识,如数制与编码、总线、接口、堆栈、中断、定时与计数等;在此基础上,详细讲解了 51 单片机的基本结构、寄存器、指令系统、基本接口、应用编程和外部扩展等内容。本教材力图体现基础性、实用性和先进性,深入浅出、循序渐进,以基础知识为铺垫,从实际应用的角度将软、硬件相结合,并精心安排了大量习题。

本教材可作为计算机类、电子信息类和机械类专业本科生的教材,也可作为相关专业学生的教材或相关工程人员的参考用书。

图书在版编目(CIP)数据

单片计算机原理及应用 / 尹勇等编著. —北京: 科学出版社, 2013.6
(高等院校电子信息类卓越工程师培养系列教材)
ISBN 978-7-03-037720-3

I. ①单… II. ①尹… III. ①单片微型计算机—高等学校—教材
IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2013)第 120054 号

丛书策划: 匡 敏 潘斯斯

责任编辑: 潘斯斯 张丽花 / 责任校对: 郑金红

责任印制: 阎 磊 / 封面设计: 迷底书装

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市文林印务有限公司 印刷

科学出版社发行 各地新华书店经销

*

2013 年 6 月第 一 版 开本: 787×1092 1/16

2013 年 6 月第一次印刷 印张: 18

字数: 460 000

定价: 40.00 元

(如有印装质量问题, 我社负责调换)

前言

“卓越工程师教育培养计划”(简称“卓越工程师计划”)是为贯彻落实《国家中长期教育改革和规划纲要(2010—2020年)》而提出的高等教育重大改革计划。当前,为适应经济社会发展和产业结构优化升级对本科工程专业人才培养的新要求,深化本科工程专业人才培养模式改革,着力培养具有较强工程实践能力、优秀创新能力和良好综合素质的优秀工程应用型人才,国内各高校正在积极组织和落实“卓越工程师计划”建设工作。

根据“卓越工程师计划”的要求,本科阶段的培养仍然是4年制,推荐“3+1”培养模式,即3年在校学习,累计1年在企业学习和做毕业设计(简称“本科3+1”模式)。然而,传统的电类课程教学培养模式都遵循一个顺序:从高等数学开始,接着是电路分析基础、模拟电子技术基础、数字电子技术基础、微机原理、C语言、单片机等。这样的顺序虽然符合教学规律,但会带来一个问题:一些应用技术类课程,如单片机、嵌入式系统、DSP和FPGA等的开课时间会由于基础课程的滞后而被推后,甚至到了大学四年级才能开课。这样的课程安排将无法满足“本科3+1”模式的培训安排,很难满足卓越工程师的培养要求。为了解决这个问题,部分专业在制定卓越工程师培养方案时,不得不以牺牲基础理论教学学时为代价来强化工程应用技术的培训。

毋庸置疑,机械地把电类课程往前提是有违教学规律的。为了配合“卓越工程师计划”,我们编写了“卓越工程师计划”系列教材,充分考虑了“本科3+1”模式培养问题。在编写《单片计算机原理及应用》教材时,我们进行了以下改进。

(1) 取缔了以往《微机原理》中关于8086和8088的内部结构、寄存器、汇编语言、信号与时序等内容。这些内容与8086和8088芯片紧密相关,但从目前的发展看,学生在实际工作中没有机会应用8086和8088芯片。因此,本教材仅介绍微机原理中与单片机相关的基本概念、基本接口等知识。

(2) 将“单片计算机原理及应用”课程提前到大学二年级,以便让学生及早地接触广泛使用的单片机知识,为大学三年级进一步学习ARM、DSP和FPGA等应用性课程打下基础。

(3) 为了与本课程配套,“电路分析基础”、“数字电子技术基础”、“模拟电子技术基础”等课程也要进行一系列的相应改变:将“常用半导体器件基础”纳入到“电路分析基础”中,大学一年级第一学期开始学习“电路分析基础”;这样做的目的是可以将“数字电子技术基础”课程提前到大学一年级的第二学期讲授,而将“数字电子技术基础”课程提前的目的则又是可以将“单片计算机原理及应用”课程提前到大学二年级的第一学期,模拟电子技术课程也在这学期开设。

通过这样的调整,本科的理论教学任务基本能在大学前三年完成,为“卓越工程师计划”安排学生在大学四年级去企业实地实践创造了良好机会。结合多年的教学经验和大量的学生反馈,经过一年多的努力,我们于2013年3月完成了本教材的撰写工作,教材的主要内容安排如下。

第1章:主要介绍微型计算机的基础知识,包括计算机的产生和发展、计算机的分类及特点、计算机的架构和指令集、计算机中常用的数制和编码、数据在计算机中的表示方法,以及定点数与浮点数等。

第2章：主要介绍计算机的基本组成单元，包括中央处理单元、存储器、I/O接口、总线、计算机中数据的基本单位等。本章举例介绍了8086单板机、家用微型计算机(PC)和51单片机的组成与工作原理，进行了对比和分析，并进一步介绍了流水线等概念。

第3章：在第1章和第2章的基础上，以51单片机为核心，介绍了单片机的分类和发展及应用。以8051单片机系统为例，介绍了51单片机的存储器组织、51单片机CPU内部结构、51单片机的时钟与复位、51单片机的引脚功能和最小系统等。结合51单片机的特点，在本章中对微机原理的堆栈部分进行了详细介绍。

第4章：考虑到学生毕业后很少使用8086的汇编语言，本章以51单片机的指令集为主线，介绍51单片机的寻址方式和指令系统，详细分析了各种寻址方式的特点，并逐一介绍了51单片机的基本指令。

第5章：基于第4章的寻址方式和基本指令，对单片机的汇编语言编程进行了讲解。

第6章：鉴于C语言在实际开发中的广泛运用，包括多种单片机、DSP、ARM等均把C语言作为开发首选，本章主要对基于51单片机的C语言进行讲解。除存储类型、指针和一些特殊功能寄存器外，51单片机的C语言与标准C语言在语法和结构方面基本一致，故本章可作为选讲部分。

第7章：介绍了微机原理中计算机中断的基本概念，包括中断、中断源、中断系统、中断优先级和中断嵌套。结合51单片机，详细给出了中断的具体处理过程和应用举例。

第8章：在第2章介绍的I/O接口概念的基础上，针对51单片机的I/O接口进行了详细讲解，并给出了LED数码管和矩阵键盘在I/O接口中的实际应用。

第9章：介绍了微机原理中定时/计数的基本概念，结合51单片机，详细介绍了其定时器/计数器的结构、组成、工作方式、基本寄存器、初始值设置等，给出了51单片机定时和计数的应用举例。

第10章：对微机原理中的串行通信技术进行了介绍，包括同步和异步通信、串行通信的方向、串行通信的波特率及基本的RS232连接。结合51单片机，详细讲解了51单片机系统中串行通信的结构、工作原理、基本寄存器、工作方式等，并给出了应用举例。

第11章：介绍了微机原理中的扩展方法，包括系统总线和扩展的基本概念、译码、片选和总线的基本概念，以51单片机为例，详细讲解了存储器的扩展、I/O接口的扩展、定时器/计数器的扩展、D/A和A/D的扩展，并给出了综合应用。

本教材深入浅出、通俗易懂，具有良好的可读性，是大学本科电子信息类专业的教材，也可供自学考试和成人教育有关专业选用，还可供研究生及科研人员参考使用。

本教材共11章，由尹勇负责第1、2、7、8章的编写，撒继铭负责第4、5、6、10、11章的编写，刘涛负责第3章的编写、李京蔚负责第9章的编写。刘岚负责全书统稿工作。阮天宇、张洁、王在振、陶小鹏、潘永峰参与了本教材部分资料的搜集整理工作，在此特表感谢。

本教材参考和引用了大量书籍和网络文献，教材的完成离不开这些宝贵的资源和同行默默无闻的奉献，在此特向他们表示崇高的敬意和衷心的感谢。

由于编者水平有限，教材中难免存在疏漏之处，恳请广大读者批评指正。

编 者

2013年3月

目 录

前言

第 1 章 计算机基础知识	1
1.1 概述	1
1.2 数制与编码	5
1.3 数据在计算机内的表示	10
1.4 常用编码	18
*1.5 定点数和浮点数	21
习题	22
第 2 章 计算机的基本工作原理	23
2.1 计算机的组成单元	23
2.2 中央处理单元	23
2.3 存储器	24
2.4 I/O 接口	33
2.5 总线	36
2.6 数据的基本单位	39
2.7 计算机系统	40
2.8 计算机的程序执行	42
2.9 流水线	43
习题	44
第 3 章 MCS-51 单片机内部结构	45
3.1 单片机简介	45
3.2 51 单片机的内部结构	50
3.3 存储器	52
3.4 特殊功能寄存器	56
3.5 时钟电路与复位电路	60
3.6 引脚功能	62
3.7 单片机的最小系统	64
习题	65
第 4 章 51 单片机的指令系统	67
4.1 程序设计语言	67
4.2 寻址方式	68
4.3 基本指令	72
习题	86

第 5 章 51 单片机汇编程序设计	89
5.1 概述	89
5.2 汇编语言的语句格式	90
5.3 伪指令	92
5.4 汇编语言程序设计步骤	93
5.5 顺序程序设计	94
5.6 分支程序设计	95
5.7 循环程序设计	97
5.8 位操作程序设计	100
5.9 子程序	101
习题	105
第 6 章 51 单片机 C 程序设计	107
6.1 概述	107
6.2 C51 的程序结构	107
6.3 C51 的数据类型	109
6.4 数据的存储器类型和存储器模式	110
6.5 指针	112
6.6 C51 对 SFR、可寻址位、存储器和 I/O 口的定义	115
*6.7 C51 的运算符	117
*6.8 函数	118
6.9 汇编语言和 C 语言的混合编程	124
*6.10 开发工具简介(Keil C)	132
习题	137
第 7 章 51 单片机中断系统	138
7.1 中断的基本概念	138
7.2 51 单片机的中断系统	141
7.3 MCS-51 中断处理过程	144
7.4 中断的程序设计	147
习题	150
第 8 章 51 单片机 I/O 接口	152
8.1 P0~P3 口的功能和内部结构	152
8.2 I/O 口编程举例	155
8.3 用并行口设计 LED 数码显示器	156
8.4 用并行口设计键盘电路	161
习题	165
第 9 章 51 单片机定时/计数器	167
9.1 概述	167

9.2 51 单片机定时/计数器工作原理	168
9.3 51 单片机定时/计数器的工作方式	170
9.4 51 单片机定时/计数器的应用	172
习题	175
第 10 章 51 单片机的串行接口	177
10.1 概述	177
10.2 单片机串行口的结构与工作原理	186
10.3 串行口的控制寄存器	188
10.4 串行口的工作方式	190
10.5 串行口的应用编程	191
10.6 利用串行口方式 0 扩展 I/O 口	196
习题	199
第 11 章 51 单片机的扩展	201
11.1 单片机系统总线和系统扩展方法	201
11.2 存储器的扩展	206
11.3 I/O 接口的扩展	215
11.4 存储器和 I/O 综合扩展举例	226
11.5 定时器/计数器扩展	230
11.6 中断的扩展	234
11.7 D/A 扩展	238
11.8 A/D 扩展	244
11.9 隔离与驱动	252
习题	255
参考答案	257
参考文献	276
附录 MCS-51 指令表	277

第 1 章 计算机基础知识

本章主要对计算机的产生和发展、计算机的分类和特点、计算机的架构、计算机中的数制和编码等内容进行介绍,所涉及的知识对所有类别的计算机都适用。

1.1 概 述

1.1.1 计算机的产生和发展

计算机(全称为电子计算机;别称为电脑)是 20 世纪最伟大的科学技术发明之一,它是一种能够按照程序的运行对各种数据和信息进行自动加工和处理的电子设备。计算机的产生和发展到目前为止共经历了以下几个阶段。

第一个阶段: 1946 年以前。第一台计算机是著名科学家帕斯卡(Pascal)发明的机械计算机。帕斯卡的计算机是一种由系列齿轮组成的装置,其外形像一个长方盒子,它需要使用某种钥匙旋紧发条后才能转动,只能够做加法和减法。这一时代的计算机主要应用于简单的加、减运算,是基于机械运行方式的。尽管有个别产品开始引入一些电学内容,却都是从属于机械的。

第二个阶段: 1946~1959 年。这段时期称为“电子管计算机时代”。计算机内部元件使用的是电子管,主要用于军事目的和科学研究,具有代表性的计算机有电子数值积分计算机(Electronic Numerical Integrator and Computer, ENIAC)、电子离散变量计算机(Electronic Discrete Variable Computer, EDVAC)、电子延迟存储自动计算器(Electronic Delay Storage Automatic Calculator, EDSAC)和通用自动计算机(Universal Automatic Computer) UNIVAC-I 型。商业用计算机起源于美国国际商业机器(International Business Machine Corporation, IBM)公司。1952~1954 年,IBM 公司先后推出了用于科学计算的 IBM 701(1952 年),用于数据处理的 IBM 702(1953 年),以及它们的后继产品 IBM 703(1954 年)。这些计算机后来被称为 IBM 700 系列。这一时代的计算机除 ENIAC 外,一般都是按存储程序模式工作的。

第三个阶段: 1960~1964 年。由于在计算机中采用了比电子管更先进的晶体管,所以这段时期称为“晶体管计算机时代”。晶体管比电子管小很多,不需要暖机时间,消耗能量较少,处理更迅速,更可靠。这一时代的计算机程序语言从机器语言发展到汇编语言。接着,高级语言 FORTRAN 语言和 COBOL 语言相继开发出来并被广泛使用。这时,开始使用磁盘和磁带作为辅助存储器。晶体管计算机的体积和价格都有所下降,计算机工业迅速发展。这一时代的计算机主要用于商业、大学教学和政府机关。

第四个阶段: 1965~1970 年。在此期间,集成电路被应用到计算机中。集成电路(Integrated Circuit, IC)是做在晶片上的一个完整的电子电路,晶片比指甲还小,却包含了几千个晶体管元件。这一时期的主要特征是以中、小规模集成电路为电子器件,并且出现操作系统,使计算机的功能越来越强,应用范围越来越广。它们不仅用于科学计算,还用于文字处理、企业

管理、自动控制等领域，并出现了计算机技术与通信技术相结合的信息管理系统，可用于生产管理、交通管理、情报检索等领域。这个时代的计算机代表是 IBM 公司用 50 亿美元开发的 IBM 360 系列。

第五个阶段：1971 年到现在。这一时代被称为“大规模集成电路计算机时代”。计算机使用的元件依然是集成电路，不过这种集成电路已经大大改善，它包含着几十万到上百万个晶体管，被人们称为大规模集成电路 (Large Scale Integrated Circuit, LSI) 和超大规模集成电路 (Very Large Scale Integrated Circuit, VLSI)。在这一阶段，微处理器和微型计算机应运而生，字长 4 位、8 位、16 位、32 位和 64 位的微型计算机相继问世和广泛应用，高级语言种类进一步增加，操作系统日趋完善，并且具备批量处理、分时处理、实时处理等多种功能。同时数据库管理系统、通信处理程序、网络软件等也不不断增添到软件系统中，软件系统的功能不断增强。

新一代计算机：从 20 世纪 80 年代开始，日本、美国以及欧洲共同体都相继开展了新一代计算机的研究。新一代计算机是把信息采集、存储、处理、通信和人工智能结合在一起的计算机系统，它不仅能进行一般信息处理，而且能面向知识处理，具有形式推理、联想、学习和解释能力，能帮助人类开拓未知领域和获取新的知识。新一代计算机的典型研究方向有：根本改变传统的计算机依据事先安排的既定程序处理问题的模式，而根据用户提出的问题，自动选择内置在知识库机中的规则，通过推理来处理问题的“知识信息处理系统(KIPS)”的智能计算机；用简单的数据处理单元模拟人脑的神经元，并利用神经元结点的分布式存储和关联，模拟人脑活动的“神经网络计算机”(Neural Network Computer, NNC)；使用由生物工程技术产生的蛋白质分子为主要原料的生物芯片，使之具有生物体自调节能力、自修复能力以及再生能力，更易于模拟人脑机制的“生物计算机”(Biological Computer)。新一代计算机的系统结构将突破传统的冯·诺依曼机器的结构和概念，实现高度并行处理。

1.1.2 计算机的分类和特点

按照比较流行的分类法，计算机可以根据其规模和处理能力分为大型(超级)计算机、小型计算机、工作站和微型计算机等几大类。

1) 大型(超级)计算机

大型(超级)计算机包括通常所说的(超)大、中型计算机，通常指大、快、贵的计算机。目前世界上运行最快的超级机速度为每秒 1704 亿次浮点运算。例如，美国的克雷公司生产的 Cray-1、Cray-2 和 Cray-3，IBM 公司生产的 IBM 360、370、4300、3090 以及 9000 系列等都是著名的巨型机。这些计算机主要应用于大范围天气预报、流体湍流分析、海洋环境与污染分析、卫星照片整理、原子核物理探索及核试验模拟、洲际导弹和航天器的辅助设计和模拟等。中国目前自主研发的大型计算机有“天河”、“星云”、“曙光”、“银河”、“神威”等系列。大型计算机被誉为计算机中的珠穆朗玛峰，是理论和试验之外的第三种科学研究手段。以“天河一号”为代表的超级计算机的发展，意味着我国在战略高技术和大型基础科技装备研制领域实现了重大突破，对维护我国国家安全，提升国家经济与科技实力意义重大。

2) 小型计算机

由于大型计算机价格昂贵、操作复杂,只有大单位才能买得起,因此,随着集成电路的问世,20世纪60年代DEC推出了一系列小型计算机。小型机(Minicomputer),通常能满足部门性的要求,为中小企事业单位采用。例如,美国DEC公司的VAX系列、DG公司的MV系列、IBM公司的AS/400系列,以及富士通公司的K系列都是有名的小型机。

3) 工作站

工作站(Workstation)有自己鲜明的特点,它的运算速度通常比微型计算机要快,需要配置大屏幕显示器和大容量的存储器,并且要有比较强的网络通信功能,主要用于特殊的专业领域,如图像处理、计算机辅助设计等方面。工作站又分为初级工作站、工程工作站、超级工作站,以及超级绘图工作站等。典型机型有HP-Apollo工作站、Sun工作站等。

4) 微型计算机

微型计算机(Microcomputer)又称微机或个人计算机,广泛应用于办公、教育领域及普通家庭。随着微型计算机的用户日益增加,人们对其要求也不断提高。目前微型计算机正在由桌上型向笔记本型发展,功能也由单一的办公用具发展为集音视频、电话、传真和电视等一体化的多媒体计算机,不但可以满足日常办公的需要,还成为人们的一大娱乐工具。其中,读者接触比较多的台式机(Personal Computer, PC)、笔记本、PDA、单片机、DSP、ARM等计算机,都属于微型计算机范畴。

计算机按用途又可分为专用计算机和通用计算机。专用计算机与通用计算机在效率、速度、配置、结构复杂程度、造价和适应性等方面有所区别。例如,在导弹和火箭上使用的计算机很大部分就是专用计算机。这些计算机非常先进,但用户不能用它来编辑文档或者玩游戏。而通用计算机适应性很强,应用面很广,但运行效率、速度和经济性依据不同的应用对象会受到不同程度的影响。

1.1.3 计算机的架构和指令集分类

1) 冯·诺依曼架构和哈佛架构

在计算机中,CPU与其部件(如内存、接口等)进行交互必须经过三种类型的总线,即数据总线、地址总线和控制总线,这是目前为止任何计算机都遵循的原则。从计算机硬件架构的角度,计算机分为冯·诺依曼体系结构和哈佛体系结构两大类。

1945年,冯·诺依曼首先提出了“存储程序”的概念和二进制原理,后来人们把利用这种概念和原理设计的电子计算机系统统称为“冯·诺依曼型结构”计算机。使用冯·诺依曼结构的处理器将数据和程序存放在同一个存储器空间中,经由同一组数据总线进行传输,如图1.1所示。目前使用冯·诺依曼结构的中央处理器和微控制器很多,如英特尔公司的8086系列、51单片机系列、ARM的ARM7系列等。

冯·诺依曼体系结构是“共享数据,串行执行”的一维计算模型。按照这种结构,指令和数据存放在共享的存储器中,CPU从存储器中取出指令和数据进行相应的运算。由于存储器的存取速度远低于CPU的运算速度,而且每一时刻只能访问存储器的一个单元(代码和数

据不能同时进行存储),从而使计算机的运算速度受到很大限制,CPU 与共享存储器间的数据交换造成了高速计算和系统性能的“瓶颈”。

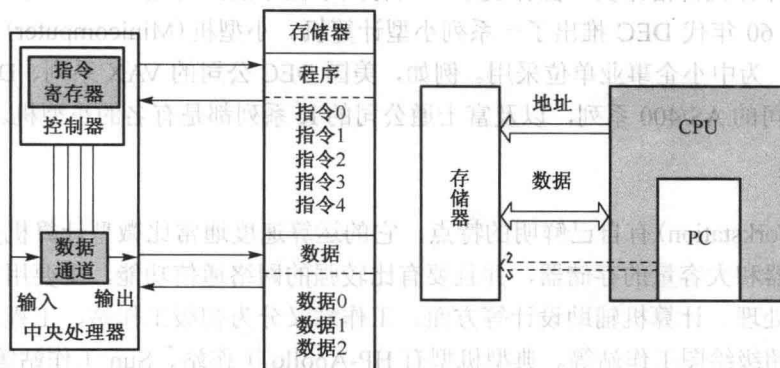


图 1.1 冯·诺依曼结构的计算机系统

与冯·诺依曼体系结构相对应的是哈佛结构。哈佛结构是一种并行体系结构,是将程序指令存储和数据存储分开的存储器结构,程序和数据存储在不同的存储空间中,即程序存储器和数据存储器独立编址、独立访问,是两个独立的存储器。这样,程序存储器和数据存储器相对应的是系统的 4 条总线:程序的数据总线与地址总线,数据的数据总线与地址总线,如图 1.2 所示。这种分离的程序总线和数据总线允许在一个机器周期内同时获得指令字(来自程序存储器)和操作数(来自数据存储器),从而提高了执行速度。又由于程序和数据存储器在两个分开的物理空间中,因此,取指和执行能完全重叠。使用哈佛结构的计算机有 TI 公司的 DSP, Microchip 公司的 PIC 系列,摩托罗拉公司的 MC68 系列, Zilog 公司的 Z8 系列, ATMEL 公司的 AVR 系列和 ARM 公司的 ARM9、ARM10 和 ARM11 系列等。

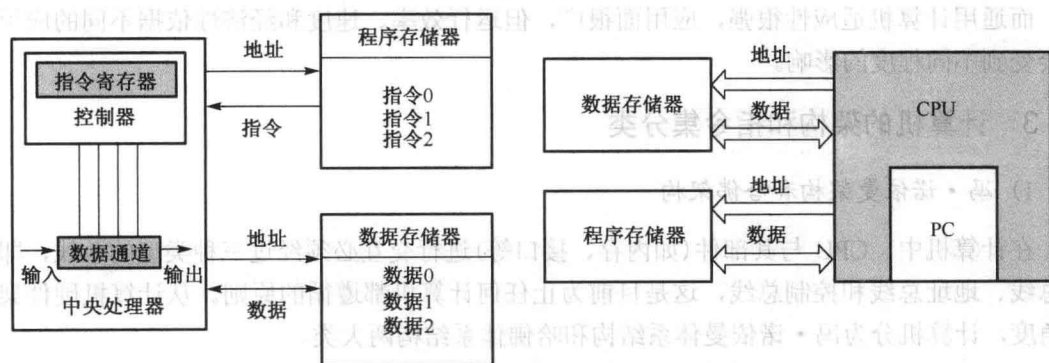


图 1.2 哈佛结构的计算机系统

2) 复杂指令集和精简指令集

从计算机的指令集的角度看,计算机又分为复杂指令集(Complex Instruction Set Computer, CISC)和精简指令集(Reduced Instruction Set Computer, RISC)两种指令架构。

早期的计算机部件比较昂贵,主频低,运算速度慢。为了提高运算速度,人们不得不将越来越多的复杂指令加入到指令系统中,以提高计算机的处理效率,这种指令集为复杂指令集(CISC)结构。早期的计算机几乎全部采用了 CISC。复杂指令集一个最大的特点就是计算机

指令数目多而且复杂, 每条指令编译成机器码以后的长度也不相等。从计算机诞生以来, 人们一直沿用 CISC 方式。早期流行的桌面计算机 x86 体系结构是按 CISC 设计的, 并一直沿用。很多其他微处理器厂商一直在走 CISC 的发展道路, 包括 Intel、AMD、TI(德州仪器)、Cyrix 以及 VIA(威盛)等。

CISC 的优点在于指令越多功能越强, 它强调代码效率, 容易和高级语言接轨, 可以对存储器直接操作, 从而实现从存储器到存储器的数据转移。然而, CISC 存在许多缺点。首先, 日益庞大的指令系统不仅使计算机研制周期变长, 而且还有难以调试、难以维护等一些自身无法克服的缺点。各种指令的使用率相差悬殊: 一个典型程序的运算过程所使用的 80% 的指令, 仅占该计算机处理器指令系统的 20%, 而且这 20% 的指令大多是取、存和加法运算等最简单的指令, 而剩下 80% 的复杂指令实际上很难在程序中得到使用。同时, 复杂的指令系统必然带来结构的复杂性, 这不但增加了设计的时间与成本, 还容易造成设计失误。

针对 CISC 的这些弊端, 以帕特逊教授为首的一批科学家提出了精简指令的设想, 即指令系统应当只包含那些使用频率很高的少量指令, 这种精简指令集仅仅包含最简单、最基本的指令, 通过这些简单、基本的指令, 就可以组合成复杂指令。同时, 每条指令的长度都是相同的, 大多数的指令都可以在一个机器周期内完成, 并且允许处理器在同一时间内执行一系列的指令。按照这个原则发展而成的计算机被称为精简指令集。

RISC 指令集的优点是: 各种指令编译成机器码以后, 长度相等, 目前的典型长度是 4 个字节; 寻址方式少且简单, 一般为 2 或 3 种, 最多不超过 4 种, 不可能出现存储器间接寻址方式; 指令集中的指令数目一般少于 100 种, 指令格式一般少于 4 种; 指令功能简单, 指令集中的各种指令都能得到充分的应用。然而, 与 CISC 相比, 编写同样功能的程序, RISC 产生的代码量比较大; 同时, RISC 不能对存储器直接访问, 无法实现存储器到存储器的数据转移。现在很多嵌入式微处理器都是 RISC, 如常用的 51 系列的单片机、ARM、MIPS、POWERPC 等。

CISC 和 RISC 两大体系各自的优缺点使它们的融合成为不可阻挡的趋势, 这种趋势在处理器的霸主 Intel 身上体现得尤为明显。Intel 在 20 世纪 90 年代后期设计的 x86 架构(CISC 的典型代表)的处理器中开始使用部分 RISC 技术, 如内置一级缓存和容量更大的二级缓存来克服内存速度的瓶颈问题; 使用超标量、超流水线技术以增强指令的执行效率; 使用指令乱序执行和分支预测技术来降低由于流水线空闲带来的性能损失。在新一代的 64 位处理器中, Intel 更进一步融合了这两大体系, 形成了清晰并行指令计算(EPIC)体系。

1.2 数制与编码

1.2.1 数制的基本概念

人们在生产实践中创建了各种表示数的方法, 按进位的原则进行计数, 这种数的表示系统称为“数制”。在日常生活中通常以十进制进行计数, 除了十进制计数以外, 还有许多非十进制的计数方法。例如, 60 分钟为 1 小时, 用的是六十进制计数法; 1 星期有 7 天, 是七进制计数法; 1 年有 12 个月, 是十二进制计数法。当然, 在生活中还有许多其他各种各样的进制计数法。

对于十进制, 它由 10 个阿拉伯符号“0~9”来表示, 并且是“逢十进一”; 对于六十进制, 它用 60 个阿拉伯符号“0~59”来表示, 并且是“逢六十进一”; 对于十二进制, 它用 12 个阿拉伯符号“0~11”来表示, 并且是“逢十二进一”……那么, 对于 N 进制数, 它可以用 N 个阿拉伯符号“0~ $N-1$ ”来表示, 并且是“逢 N 进一”。一般地说, 任何一个 N 进制数都可表示为

$$K_{(N\text{进制数})} = a_{n-1}N^{n-1} + a_{n-2}N^{n-2} + \cdots + a_1N^1 + a_0N^0 + a_{-1}N^{-1} + a_{-2}N^{-2} + \cdots + a_{-m}N^{-m}$$

$$= \sum_{i=m}^{n-1} a_i \times N^i$$

式中, a_i ($n-1 \geq i \geq -m$) 为各位数上的数字, 它可以在 0, 1, $\cdots$, $N-1$ 共 N 种数种任意取值; N 为进制的基数; N 的 i 次方为第 i 位的权; m 和 n 为正整数, 其中 n 为整数部分的位数, m 为小数部分的位数。

1.2.2 常用的数制分类

在计算机中, 有如下的常用进位计数制。

1) 十进制 (Decimal)

十进制数是日常生活中使用最广的计数制。组成十进制数的符号有 0、1、2、3、4、5、6、7、8、9 共 10 个符号, 称这些符号为数码。在十进制中, 每一位有 0~9 共 10 个数码, 最大数字是 9, 最小数字是 0, 基数为 10。超过 9 就必须用多位数来表示。十进制数做加法时满足“逢十进一”, 减法时满足“借一当十”。对于任意十进制数可表示为

$$K_{(10)} = a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \cdots + a_{-m} \times 10^{-m}$$

例如, 对于十进制数 435.05 可表示为

$$435.05 = 4 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 + 0 \times 10^{-1} + 5 \times 10^{-2}$$

2) 二进制 (Binary)

与十进制相似, 二进制数也遵循两个规则: 仅有两个不同的数码, 即 0 和 1; 满足加法时“逢二进一”, 减法时“借一当二”。二进制数中最大数字是 1, 最小数字是 0, 基数为 2。对于任意一个二进制数可表示为

$$K_{(2)} = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m}$$

例如, 对于二进制数 1101.101 可表示为

$$1101.101\text{B} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

3) 八进制 (Octal)

类似地, 八进制数有 8 个不同的数码, 即 0、1、2、3、4、5、6、7, 并满足加法时“逢八进一”, 减法时“借一当八”。八进制数中最大数字是 7, 最小数字是 0, 基数为 8。对于任意一个八进制数可表示为

$$K_{(8)} = a_{n-1} \times 8^{n-1} + a_{n-2} \times 8^{n-2} + \cdots + a_1 \times 8^1 + a_0 \times 8^0 + a_{-1} \times 8^{-1} + a_{-2} \times 8^{-2} + \cdots + a_{-m} \times 8^{-m}$$

例如, 对于八进制数 107.13 可表示为

$$107.13Q = 1 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} + 3 \times 8^{-2}$$

4) 十六进制数(Hexdecimal)

十六进制数有 16 个不同的数码，即 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，并满足加法时“逢十六进一”，减法时“借一当十六”。十六进制数中最大数字是 F，最小数字是 0，基数为 16。对于任意一个十六进制数可表示为

$$K_{(16)} = a_{n-1} \times 16^{n-1} + a_{n-2} \times 16^{n-2} + \cdots + a_1 \times 16^1 + a_0 \times 16^0 + a_{-1} \times 16^{-1} + a_{-2} \times 16^{-2} + \cdots + a_{-m} \times 16^{-m}$$

比如，对于十六进制数 E09.F3 可表示为

$$E09.F3H = E \times 16^2 + 0 \times 16^1 + 9 \times 16^0 + F \times 16^{-1} + 3 \times 16^{-2}$$

1.2.3 数制的表示方法

有两种数制表示方法，即后缀表示法和下标表示法，这两种表示方法如图 1.3 和图 1.4 所示。

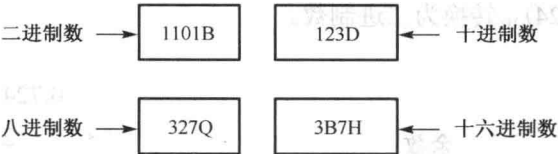


图 1.3 后缀表示法

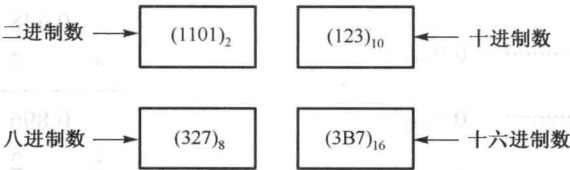


图 1.4 下标表示法

计算机中常用的几种进位数制如表 1.1 所示。

表 1.1 计算机中常用的进位数制的表示

进位制	二进制	八进制	十进制	十六进制
规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数	$r=2$	$r=8$	$r=10$	$r=16$
数符	0, 1	0, 1, 2, ..., 7	0, 1, 2, ..., 9	0, 1, 2, ..., 9, A, B, ..., F
权	2^i	8^i	10^i	16^i
表示符	B	Q	D	H

值得注意的是，如果一个数的尾部没有后缀，也没有下标，则该数默认是十进制。例如，11.725 表示该数是一个十进制数。

1.2.4 数制转换

1) 其他进制与十进制之间的转换

二进制数转换成十进制数的方法是：将二进制数的每一位数乘以它的权，然后相加，即可求得对应的十进制数值。同理，若将任意进制数转换为十进制数，只需将数 $(N)_R$ 写成按权展开的多项式表示式，并按十进制规则进行运算，便可求得相应的十进制数 $(N)_{10}$ 。

【例 1-1】 把二进制数 10.01 转换成相应的十进制数。

解: $(10.01)_2 = 1 \times 2^1 + 1 \times 2^{-2} = (2.25)_{10} = 2.25D$

【例 1-2】 把八进制数 13.7 转换成相应的十进制数。

解: $(13.7)_8 = 1 \times 8^1 + 3 \times 8^0 + 7 \times 8^{-1} = (11.875)_{10} = 11.875D$

【例 1-3】 把十六进制数 1A.AF 转换成相应的十进制数。

解: $(1A.AF)_{16} = 1 \times 16^1 + A \times 16^0 + A \times 16^{-1} + F \times 16^{-2} = (26.68)_{10} = 26.68D$

2) 十进制与其他进制之间的转换

将十进制数转换成二进制数时, 整数部分和小数部分分别转换, 然后再合并。十进制整数转换为二进制整数的方法是“除 2 取余”; 十进制小数转换为二进制小数的方法是“乘 2 取整”。同理, 若将十进制数转换为 N 进制数, 需将整数部分和小数部分分别转换, 整数部分是“除 N 取余”; 小数部分是“乘 N 取整”。

【例 1-4】 将 $(57.724)_{10}$ 转换为二进制数。

解:

	整数	小数	0.724
2 57	余数		$\times 2$
2 28	1 = a_0		1.448
2 14	0 = a_1		0.448
2 7	0 = a_2		$\times 2$
2 3	1 = a_3		0.896
2 1	1 = a_4		$\times 2$
0	1 = a_5		1.792
			0.792
			$\times 2$
			1.584

$$(57)_{10} = (111001)_2$$

$$(0.724)_{10} = (0.1011)_2$$

故 $(57.724)_{10} = (111001.1011)_2$ 。

【例 1-5】 将十进制数 226.125 转换为十六进制数。

解: 对整数部分的转换采用除 16 取余法; 对小数部分的转换采用乘 16 取整法。

余数	取整
16 226	0.125
16 14	$\times 16$
0	2 ← 2.0

$$(226)_{10} = (E2)_{16}$$

$$(0.125)_{10} = (0.2)_{16}$$

故 $(226.125)_{10} = (E2.2)_{16}$ 。

注意: 当在不同数制间进行转换时, 把二进制、八进制、十六进制数转换为十进制数或十进制整数转换为其他数制的整数时, 都能做到完全准确。但把十进制小数转换为其他数制时, 除少数没有误差外, 大多存在误差。

【例 1-6】 求 $(0.5678)_{10}$ 的二进制数。

解:

$0.5678 \times 2 = 1.1356 \cdots$ 取出整数 1

$0.1356 \times 2 = 0.2712 \cdots$ 取出整数 0

$0.2712 \times 2 = 0.5424 \cdots$ 取出整数 0

$0.5424 \times 2 = 1.0848 \cdots$ 取出整数 1

:

从例 1-6 可以看出, 无论将转换计算到多少位, 也不能把小数点后面的数变成 0, 也就是说总避免转换误差, 只是小数后位数越长误差越小精度越高而已。

3) 二进制、八进制、十六进制之间的转换

二进制、八进制、十六进制由于它们存在着关系: $2^3=8$, $2^4=16$, 因此, 它们之间的转换非常简单。

把二进制数转换为八进制数时, 按“三位并一位”的方法进行。以小数点为界, 将整数部分从右向左每三位一组, 最高位不足三位时, 添 0 补足三位; 小数部分从左向右, 每三位一组, 最低有效位不足三位时, 添 0 补足三位。然后, 将各组的三位二进制数按权展开后相加, 得到一位八进制数。

将八进制数转换成二进制数时, 采用“一位拆三位”的方法进行。即把八进制数每位上的数用相应的三位二进制数表示即可。例如:

$$(1011.0101)_2 = (001 \ 011.010 \ 100)_2 = (13.24)_8 \quad (46.7)_8 = (100 \ 110.111)_2$$

把二进制数转换为十六进制数时, 按“四位并一位”的方法进行。以小数点为界, 将整数部分从右向左每四位一组, 最高位不足四位时, 添 0 补足四位; 小数部分从左向右, 每四位一组最低有效位不足四位时, 添 0 补足四位。然后, 将各组的四位二进制数按权展开后相加, 得到一位十六进制数。将十六进制数转换成二进制数时, 采用“一位拆四位”的方法进行。即把十六进制数每位上的数用相应的四位二进制数表示即可。例如:

$$(10010.01)_2 = (0001 \ 0010.0100)_2 = (12.4)_{16}$$

$$(79B.FC)_{16} = (111 \ 1001 \ 1011.1111 \ 11)_2$$

不同进制之间的对照表如表 1.2 所示。

表 1.2 不同进制数的对照

十进制	二进制	八进制	十六进制
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9