

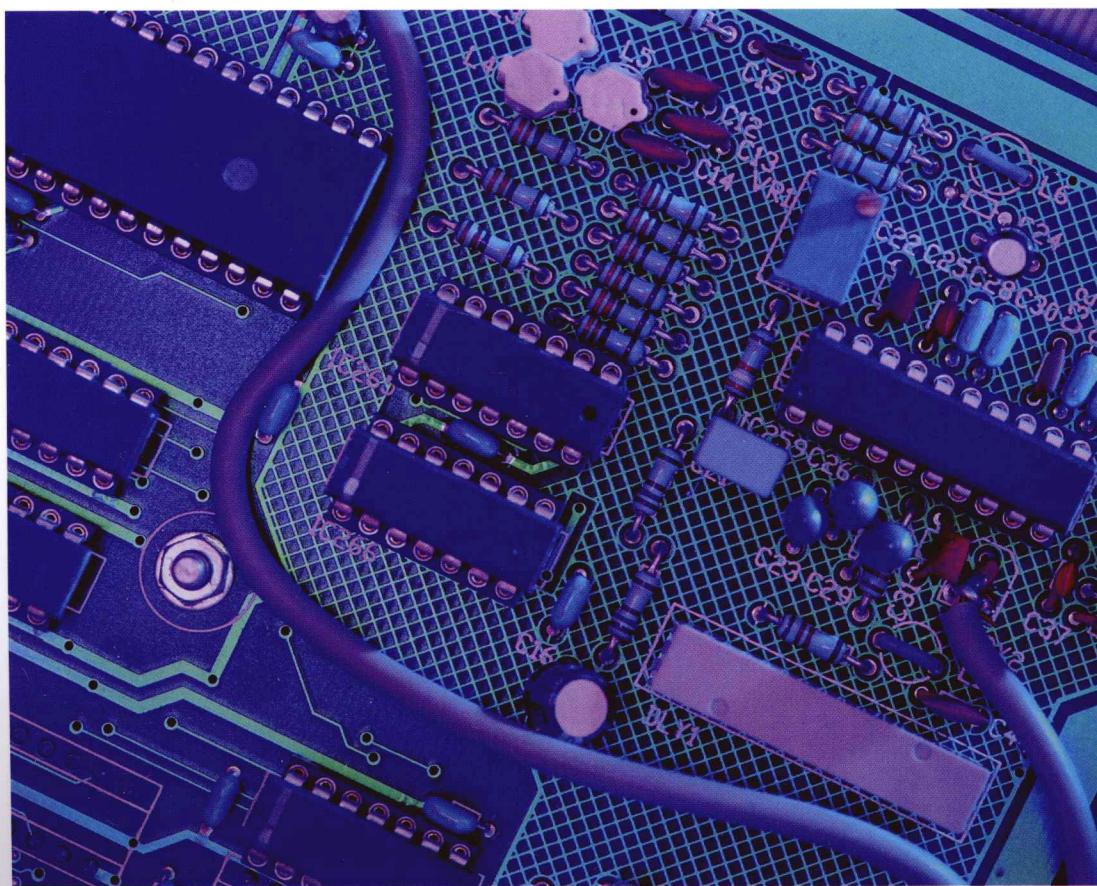


高等学校规划教材

51

单片微型机 原理和接口教程

◆ 周思跃 编著



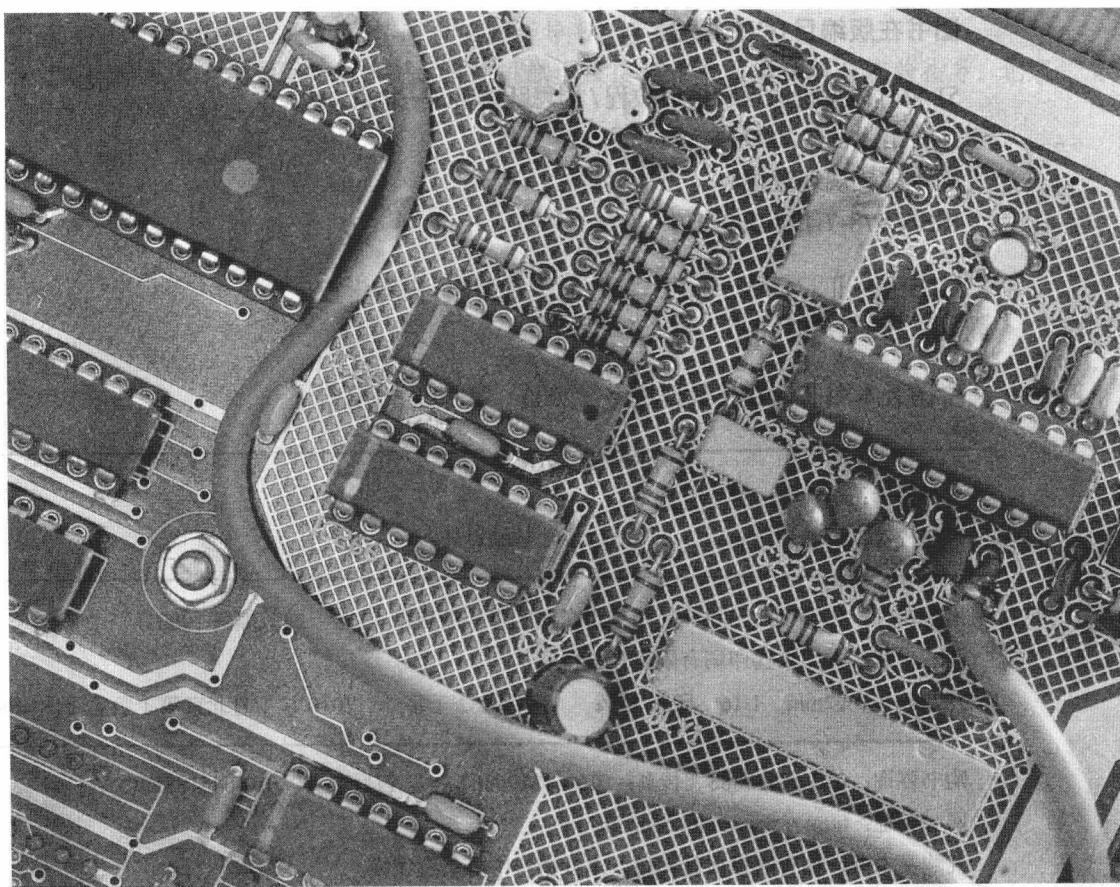
化学工业出版社

高等学校规划教材



51 单片微型机 原理和接口教程

❖ 周思跃 编著



化学工业出版社
· 北京 ·

本书共分 3 篇 12 章。第 1 篇共有 7 章的内容, 包含了 51 系列单片机的结构原理, 汇编语言程序设计。第 2 篇共有 4 章的内容, 包含了 51 系列单片机的扩展应用方式, 常用接口芯片与单片机的连接和应用, 单片机应用系统的设计。第 3 篇只有 1 章, 包含了 C51 语言程序设计的基础内容, 将读者引入单片机学习的新境界。

本书在每一章内都穿插了一些实验, 全书共穿插了十八个实验, 使教师可以在实验室里, 一边讲课, 一边做实验。学生自学时也可以看看书, 动动手。

本书在编写的文字上力求通俗、口语化, 经常用设问的方式引导学生, 增加其自学的兴趣。

本书可以作为普通高等学校机械设计制造及其自动化、电气工程及其自动化、自动化、电子信息工程等相关专业单片机课程的教材, 也可以供从事单片机设计与研究的科研人员参考。

图书在版编目(CIP)数据

51 单片微型机原理和接口教程 / 周思跃编著. —北京: 化学工业出版社, 2009.11
高等学校规划教材
ISBN 978-7-122-06971-9

I. 5… II. ①周… III. 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 (2009) 第 195656 号

责任编辑: 郝英华

责任校对: 周梦华

装帧设计: 杨 北

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 大厂聚鑫印刷有限责任公司

787mm×1092mm 1/16 印张 17¼ 字数 482 千字 2010 年 1 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 29.80 元

版权所有 违者必究

前言

笔者从事单片机原理与应用课程教学多年,对象是机械电子工程专业和机械工程与自动化专业的学生,深感本课程对学生的重要性和艰难性。每学期总有数名因感困难而放弃该课程的考试,这始终成为笔者多年来的遗憾。这也多次使笔者心存编写此书的愿望,希望编写一本学生喜欢看的专业书,编写一本教师容易讲的专业书。

本书在编写的文字上力求通俗、口语化,每一张电路图都有详细的解释,每一条程序中的指令都有注释。

本书在每一章的开头,采用设问的方式引导学生,使他们了解本章的主题,并增加他们自学的兴趣。在每一章后面都安排了本章小结,便于学生和教师对本章内容进行梳理。

本书在每一章内都插入了实验,全书共有十八个实验,使得教师可以在实验室讲解单片机原理与应用课程,一边讲课,一边做实验,克服以往课堂教学与实验教学分离的缺点。

本书在每一章里都安排了一些习题,习题的设计大多数与例题类似,便于学生对本章知识的复习巩固。

以下内容作为本书的导读。

本书在结构上分为3篇。第1篇是基础篇,共有7章(第1~7章),内容涉及51系列单片机的核心知识,是本书的重点。本篇内容可作为单片机原理课程的教材。

◇ 第1章内容把计算机与单片机之间的关系理清,把单片机中最基础的问题搞懂,自然是学生学习单片机的起点。

◇ 第2章的内容是把单片机内部最主要的部件(CPU、存储器和接口)聚在一起,使学生理解单片机的实质。考虑到学习的顺序,2.4节的内容可以暂时搁一下。

◇ 第3章的内容是51系列单片机的指令系统,是单片机的软件设计基础。

◇ 第4章的内容是汇编语言的程序设计,使一方面可以把指令的知识得以应用,一方面可以学会汇编语言编程的知识。第4章的内容学完后,建议把第2章留下的内容(2.4节)I/O接口学习完,接着学习第5章。

◇ 第5章的内容是中断系统,是本书的难点之一,需花心血,反复研磨。

◇ 第6章的内容是定时器/计数器(CTC),学习CTC的知识可结合中断技术,进一步理解中断系统的知识。

◇ 第7章的内容是串行接口,这也是本书的难点之一,但这是计算机通信的基础。

第2篇是扩展、接口和应用篇,共有4章(第8~11章),内容涉及51单片机的扩展应用模式、常用的接口和单片机应用系统的设计。本篇内容可以作为单片机接口和应用课程的教材。

◇ 第8章的内容是存储器在单片机外面的扩展和连接,并行接口的扩展和连接。这是51系列单片机扩展技术的基础。

◇ 第9章的内容是显示器/键盘电路和接口的原理及其应用,显示器/键盘电路是单片机应用系统最常用的配置。

◇ 第10章的内容是模拟接口D/A转换器和A/D转换器的原理,以及与单片机的连接和应用。

◇ 第11章的内容是单片机应用系统设计的一般方法,主要包含单片机应用系统的硬件结构和软件结构。

第3篇是C51篇，只有1章（第12章）。主要是将学生引领进单片机学习的另一种境界，为进一步学习高级的嵌入式系统作准备。其内容包括C51语言程序设计的一般方法，用C51语言设计接口控制程序。

为方便教学，本书配套的电子教案可免费提供给采用本书作为教材的大专院校使用。如有需要，请发电子邮件至 haoyinghua@cip.com.cn。

在本书编写过程中，沈林勇副研究员提出了一些建设性的意见，在此表示真诚的感谢。

借建设上海高等学校教育高地之东风，历时一年，绘制每一张图，打入每一个字，终于完成了此书的编撰。

由于编者水平有限，书中缺点和错误在所难免，恳请广大师生批评指正。

编 者
2009年10月

目 录

基础篇

1 单片机基础知识	2	实验二、认识 P0~P3 口	32
1.1 引言	2	2.5 时钟和时钟电路	33
1.1.1 计算机	2	2.6 复位和复位电路	33
1.1.2 微型机	4	2.7 本章小结	34
1.1.3 单片机与嵌入式计算机	4	习题 2	34
1.2 常用数制和码制	5	3 MCS-51 系列指令系统	36
1.2.1 常用的数制及其转换	5	3.1 指令格式和寻址方式	36
1.2.2 常用的码制	8	3.1.1 立即寻址方式	37
实验一、认识补码	12	3.1.2 直接寻址方式	37
1.3 单片机内部结构概述	13	3.1.3 寄存器寻址方式	38
1.3.1 中央处理器 CPU 概述	13	3.1.4 寄存器间接寻址方式	38
1.3.2 存储器概述	16	3.1.5 变址寻址方式	38
1.3.3 输入/输出接口概述	18	3.1.6 相对寻址方式	39
1.4 单片机产品概述	19	3.1.7 位寻址方式	40
1.4.1 单片机产品的类型	19	3.2 数据传送类指令	41
1.4.2 单片机产品的应用	20	3.2.1 内部数据传送指令	41
1.5 本章小结	20	3.2.2 内外数据传送指令	44
习题 1	21	3.2.3 查表指令	46
2 MCS-51 系列单片机系统基本组成	22	实验三、寻址方式和堆栈的认识	46
2.1 总体结构	22	3.3 数据处理类指令	47
2.1.1 内部结构	22	3.3.1 算术运算指令	48
2.1.2 外部引脚	23	3.3.2 逻辑运算指令	51
2.2 CPU 特性	24	3.3.3 移位指令	54
2.3 存储器组织	24	3.4 位操作类指令	55
2.3.1 程序存储器	25	3.4.1 位变量传送指令	56
2.3.2 数据存储器	26	3.4.2 位变量设置指令	56
2.3.3 特殊功能寄存器	28	3.4.3 位变量逻辑运算指令	57
2.3.4 外部数据存储器 and I/O 接口	29	3.5 程序转移类指令	58
2.4 I/O 接口	30	3.5.1 无条件转移指令	58
2.4.1 P1 口	30	3.5.2 有条件转移指令	60
2.4.2 P3 口	31	3.5.3 调用和返回子程序指令	62
2.4.3 P2 口	32	实验四、转移类指令和位操作类指令	
2.4.4 P0 口	32	的认识	65

3.6 CPU 执行指令的时序	66	5.3.1 中断初始化程序的编制	103
3.7 本章小结	67	5.3.2 中断服务子程序的编写	104
习题 3	67	实验八、认识中断系统和中断服务子程序编写	105
4 MCS-51 系列汇编语言程序设计	71	5.4 本章小结	106
4.1 伪指令	71	习题 5	107
4.1.1 起始指令	71	6 MCS-51 单片机定时器/计数器	108
4.1.2 结束指令	71	6.1 定时器/计数器的结构和工作原理	108
4.1.3 数据定义类指令	72	6.1.1 定时器/计数器 T0 和 T1	108
4.2 汇编语言语句格式	74	6.1.2 定时器/计数器 T2	112
实验五、认识汇编语言程序格式	74	6.1.3 定时器/计数器的使用极限	115
4.3 汇编语言程序设计方法	75	6.2 控制定时器/计数器的特殊功能寄存器	115
4.3.1 顺序结构程序设计	76	6.3 定时器/计数器的应用及其编程	117
4.3.2 分支结构程序设计	76	6.3.1 初始化编程	117
4.3.3 循环结构程序设计	78	6.3.2 定时器/计数器应用编程	118
4.3.4 子程序设计	80	实验九、定时器/计数器应用	126
4.3.5 经典程序设计举例	85	6.4 本章小结	128
实验六、查表程序设计	93	习题 6	128
实验七、排序程序设计	94	7 MCS-51 单片机串行接口	130
4.4 本章小结	95	7.1 串行通信的概念	130
习题 4	95	7.2 单片机串行接口的结构原理	132
5 MCS-51 单片机系列中断系统	98	7.3 单片机串行接口的工作方式和原理	134
5.1 中断的概念	98	7.3.1 串行口控制寄存器 SCON (Serial Control)	134
5.1.1 什么是中断	98	7.3.2 串行口工作方式 0	135
5.1.2 中断源	98	7.3.3 串行口工作方式 1	138
5.1.3 中断优先级	99	7.3.4 串行口工作方式 2、3	138
5.1.4 控制中断的特殊功能寄存器	100	7.3.5 串行口数据传送的波特率	139
5.2 中断处理过程	102	7.4 串行口应用举例	140
5.2.1 中断申请	102	实验十、串行口双机通信	147
5.2.2 中断响应条件	102	7.5 本章小结	149
5.2.3 中断处理过程	102	习题 7	150
5.2.4 中断请求信号的撤消	103		
5.3 中断程序的编制	103		

扩展、接口和应用篇

8 MCS-51 单片机的扩展技术	152	8.3.1 常用静态随机存储器 (SRAM) 芯片	158
8.1 单片机扩展原理	152	8.3.2 数据存储器扩展应用举例	159
8.2 程序存储器扩展	153	8.3.3 CPU 如何访问外部存储器	163
8.2.1 EPROM 存储器芯片	153	实验十一、数据存储器的扩展和测试	164
8.2.2 程序存储器芯片与系统的连接举例	154	8.4 并行接口的扩展	165
8.3 数据存储器的扩展	158	8.4.1 基本并行接口及其扩展	166

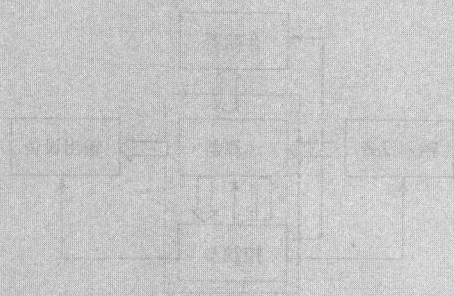
实验十二、基本并行口及其扩展	169	习题 9	210
8.4.2 8255A 并行口及其扩展	171	10 D/A 和 A/D 转换器的接口技术	211
实验十三、8255A 并行口及其扩展	180	10.1 D/A 转换器和接口技术	211
8.5 综合接口芯片 8155	181	10.1.1 D/A 转换器原理	211
8.6 本章小结	185	10.1.2 D/A 转换器性能指标	214
习题 8	185	10.1.3 DAC0832 的结构原理和	
9 键盘/显示器的接口技术	187	接口技术	214
9.1 LED 数码显示器结构原理和		10.1.4 DAC0832 的应用和编程	216
接口技术	187	实验十五、D/A 转换器应用举例	219
9.1.1 LED 数码显示器结构原理	187	10.2 A/D 转换器和接口技术	221
9.1.2 LED 数码显示器的显示码	187	10.2.1 A/D 转换器原理	221
9.1.3 LED 显示器的显示方式及其		10.2.2 A/D 转换器的主要性能指标	221
显示程序	188	10.2.3 ADC0809 的结构原理和	
9.2 键盘结构原理与接口技术	193	接口技术	222
9.2.1 键盘电路的结构原理	193	10.2.4 A/D 转换器的应用举例	223
9.2.2 键抖动问题	194	实验十六、A/D 转换器的应用和编程	227
9.2.3 键盘接口电路举例	195	10.3 本章小结	228
9.3 8279 显示器/键盘接口电路的		习题 10	228
原理和应用	198	11 单片机应用系统的研制	230
9.3.1 8279 内部结构	198	11.1 单片机应用系统设计概论	230
9.3.2 8279 的接口工作方式	199	11.2 单片机应用系统硬件设计	230
9.3.3 8279 的控制命令字、状态字和		11.3 单片机应用系统的软件设计	233
数据格式	200	11.4 单片机应用系统设计举例	235
9.3.4 8279 芯片外部引脚	202	实验十七、单片机应用系统设计举例	244
9.3.5 8279 的应用举例	203	11.5 本章小结	244
实验十四、8279 显示器/键盘接口和编程	209	习题 11	244
9.4 本章小结	209		

C51 篇

12 C51 程序设计初步	246	12.4 C51 程序设计	252
12.1 C51 概论	246	12.4.1 基本并行接口的输入/输出	
12.1.1 C51 语言与 C 语言	246	程序的设计	252
12.1.2 C51 语言与 51 系列汇编语言	246	12.4.2 8255A 并行接口的输入/输出	
12.2 C51 的数据类型	247	程序的设计	253
12.2.1 数据类型	247	12.4.3 51 串行口发送/接收程序设计	254
12.2.2 存储器类型	247	12.4.4 D/A 转换控制程序的设计	257
12.2.3 访问特殊功能寄存器	249	12.4.5 A/D 转换控制程序的设计	258
12.2.4 位变量的定义	249	12.5 C51 函数库	260
12.2.5 C51 指针	250	12.6 本章小结	261
12.2.6 绝对地址访问	250	习题 12	261
12.2.7 存储器模式	251	附录 51 系列单片机指令表	263
12.3 C51 位变量逻辑运算符	251	参考文献	267
实验十八、认识 C51 的位操作运算符	251		

● 基础篇

本篇包含了单片机原理和应用最重要、最核心的内容，也是学习后续内容的基础。



1 单片机基础知识

电脑、计算机、甚至是单片机，都已经成为人们日常生活中耳熟能详的词语了。然而，单片机与计算机究竟是什么关系？单片机究竟为何物？单片机与嵌入式系统又是什么关系？另外，电脑最为神奇作用是能处理各种各样信息，那么它是如何来表达这些信息的呢？这些问题是学习本书内容的基础，也正是本章要回答的问题。

1.1 引言

单片机究竟为何物？单片机与计算机究竟是什么关系？这要追溯到计算机的历史和发展。

1.1.1 计算机

计算机，俗称电脑，诞生于 20 世纪 40 年代，它的功能从原来的计算，不断地向控制、信息处理和通信等各个领域延伸。计算机随着电子技术的发展而发展，电子技术经历了电子管时代、晶体管时代、大规模集成电路和超大规模集成电路时代，计算机家族也从小型机、大型机，发展到巨型机和微型机。

尽管计算机有着光辉的历史和辉煌的成就，但它最初的内部结构是什么样的呢？据文献记载，计算机内部结构的蓝图是由世界级科学家冯·诺依曼绘制的，如图 1-1 所示。

在冯·诺依曼的蓝图中，计算机的硬件结构由五大部件组成，这五大部件是当代计算机硬件的雏型，学习硬件可以从这里出发。以下将这五大部件的功能简述如下。

① 运算器。运算器是计算机的中心，其功能就是进行数据运算，计算机也正是因此而被命名为“计算机”。由于运算器处在计算机的中心位置，早期计算机的一些数据传送需经过运算器。

② 控制器。控制器是执行指令的部件，它向系统中其他部件发出控制信号，协调各部件工作。计算机通过它来实现本身运行过程的自动化。

③ 存储器。存储器是存放程序和数据部件，正是因为事先将程序储存在存储器中，控制器才能自动地完成计算机交给的任务。

④ 输入设备。输入设备是输入信息的部件，输入的信息有原始的程序、数据，以及计算机操作命令等，常用的输入设备有键盘、鼠标、扫描仪等。

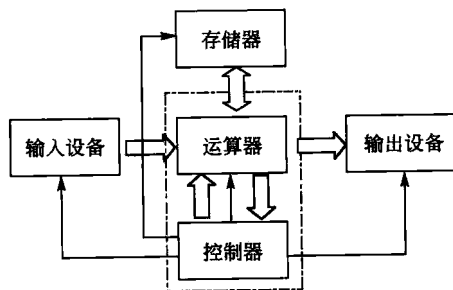


图 1-1 冯·诺依曼的计算机硬件结构图

⑤ 输出设备。输出设备是输出信息的部件，输出的信息有计算结果、绘制的图片表格，以及计算机操作提示信息等，常用的设备有显示器、打印机等。

也许现在的人们已经很难将现在的计算机结构与冯·诺依曼的计算机结构联系起来了。在现在的计算机中，已看不到运算器和控制器的身影了，它们到哪里去了呢？是不是消失了呢？这个问题将在下一小节来回答。

现在的首要问题是冯·诺依曼的计算机是如何工作的？这种计算机有什么特点？它与以后的计算机有什么关系？

首先来仔细观察一下冯·诺依曼的蓝图，即图 1-1。图中的五个方框代表了计算机的五大部件，图中的连接线路有两类，即较粗的有向线段和较细的有向线段，它们分别代表数据线和控制线。控制线的信号均来自控制器，每一部件也均有一组数据线与相关部件相连。为了使计算机按照人们的意图进行计算（早期计算机的任务就是计算），必须事先把计算方法和操作步骤用指令编制成计算机可执行的程序，通过输入设备，将一条一条指令以二进制数的形式存放在存储器中。计算机启动后，就从存储器中依次取出指令，送控制器分析，控制器根据指令的操作要求，向系统各部件发出一系列相应的操作信号，使系统各部件完成相应的工作。例如要求运算器完成加法工作等。程序执行过程中或执行结束时，可以通过输出设备将中间结果或最终结果显示出来。

根据冯·诺依曼计算机的工作原理，可以总结出以下特点。

① 采用存储程序的工作方法。将计算机要处理的数据和计算方法、步骤，事先按计算机操作命令（指令）编制成程序，存放在计算机的存储器中。计算机运行时，自动地、连续地从存储器中取出指令并执行，不需要人工干预。

② 采用二进制代码表示指令和数据。计算机内部的信息全部采用二进制编码！这样的设计主要原因是二进制的 0 和 1 两种状态用电路容易实现，运算器电路比较简单。另外，二进制信息对应的数字信号的抗干扰能力强。

令人感到惊奇的是，目前的计算机（电脑）仍然具有以上两个特性，只是人们没有感知而已。比如，在计算机中执行一个程序，只要对准这个程序的图标点击两下就行。其实，当用户点击了这个程序后，计算机中的系统软件操作系统就将该程序从硬盘装载到存储器，然后执行之。再比如，在使用计算机遇到需要输入的数据时，一般都是直接输入十进制的数据而不是二进制数据。其实这也是计算机中的系统软件在暗地里将十进制数转换成二进制数而已。

由计算机的结构原理引出了两个名词，即硬件和软件，这是计算机区别于过去其他机器的重要特征。于是对于计算机结构完整的描述是，计算机是由硬件和软件组成，其硬件主要包括控制器、运算器、存储器、输入设备和输出设备；其软件可分为系统软件、程序设计语言和应用软件，系统软件包括操作系统、编辑程序、诊断程序、调试程序、数据库管理程序、装配连接程序等；程序设计语言包括机器语言、汇编语言、高级语言和应用语言；应用软件包括通用软件、实用软件、用户程序等。计算机系统结构如图 1-2 所示。

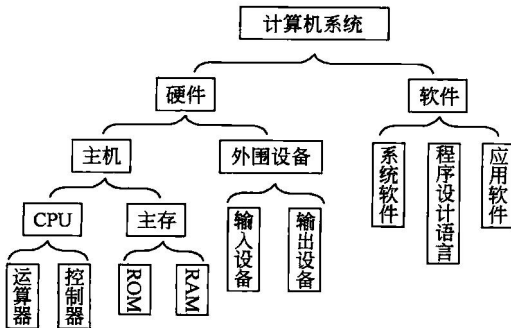


图 1-2 计算机系统结构

随着电子技术的发展，特别是大规模集成电路的发展，计算机硬件的结构发生了一个重要的变化，那就是将原来计算机中的控制器和运算器组合在一起，集成在一块半导体芯片上，并给它起了一个响亮的名字——中央处理器（Central Processing Unit, CPU）。这就是为什么在现在的计算机中看不到控制器和运算器的影子了。控制器和运算器隐身后，使计算机系统的可靠性得到了很大的提高。原来控制器与运算器之间存在大量的连线，这些连线是计算机故障的常发因素，随着两者的集成，这些故障也随之消失。

1.1.2 微型机

计算机的发展似乎注定要沿着一条与众不同的道路来发展，一方面朝着巨型机的方向发展，不仅体积庞大，而且技术上追求高精尖，满足国家发展的高端要求。另一方面走大众化的道路，满足普通民众的生活和工作要求，使计算机体积微型化，也就是人们平时看到最多的一种计算机——微型机。

那么微型机与传统的计算机在结构上有什么不同吗？答案是基本相同。先来看一下早期的微型机结构，如图 1-3 所示。

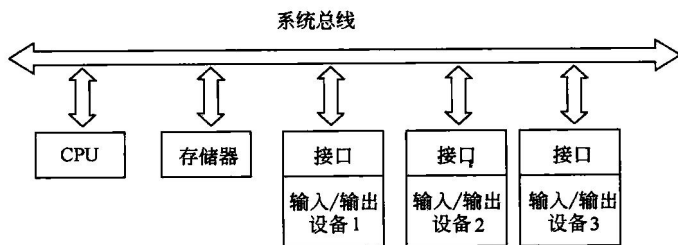


图 1-3 早期微型机结构

图 1-3 所示的是一种单 CPU、单总线的微型机结构。在这种结构的微型机中，CPU、存储器、接口各自连接在系统唯一的一条总线上，输入/输出设备分别通过接口与系统相连。

在这种结构的电脑中，CPU 通过系统总线到存储器中读取指令或数据，CPU 也是通过总线与输入/输出设备传送信息。但是系统不能在同一时刻有三个部件之间传送信息，否则不同的信息将在同一条总线上撞车。

微型机的发展是随着计算机的发展而发展的，这里的计算机是指小型机和大型机，微型机借鉴小型机和大型机发展的技术，从单 CPU、单总线结构，朝着多 CPU（多核）、多总线的结构方向发展。

对于初学者来说，微型机有关术语经常容易引起混淆，需要先强调一下，有利于今后的学习。

① 微处理器。微处理器是将控制器、运算器、寄存器集成在一块芯片上的系统，称为 MPU（Micro Processor Unit），俗称 CPU。

② 微型计算机。以 CPU 为中心，配上存储器、接口，用系统总线将三者连接起来的系统，称为 MC（Micro Computer）。

③ 微型计算机系统。以 MC 为中心，配上输入/输出设备、系统软件组成的系统，称为 MCS（Micro Computer System），俗称电脑、计算机或微机。

1.1.3 单片机与嵌入式计算机

在微型计算机系统跟着大型机、巨型机朝着通用性（多功能）方向狂奔的同时，又开辟了另一条道路来发展。1976 年 Intel 公司推出了世界上第一片单片机芯片 8048。

什么是单片机呢？说得简洁点，就是将上一节提到的微型计算机（MC）集成在一块芯片上，即将图 1-3 中的 CPU、存储器、接口等部件集成在一块芯片上的系统称为单片机 SCM（Single Chip Microcomputer）。这不能不说是一个富有深远意义的创新。

从单片机诞生这天起，就与它的“哥哥”微型计算机系统分道扬镳。微型计算机系统走的是通用化的发展道路，也就是不断增加自身的功能。而单片机走的是专用化的发展道路，即朝着控制器的方向发展，使计算机成为进入人类社会全面智能化时代的有力工具。

Intel 公司经过不断完善单片机产品，在 1980 年推出被业界公认的经典产品 MCS-51 系列单片机。经历了近三十年，MCS-51 系列仍然是单片机领域内的主流产品，这不能不说是件了不起的事情。又经过 4 年的努力，1984 年 Intel 公司研制出了高性能 MCS-96 系列单片机。

20 世纪 80 年代和 90 年代是单片机迅速发展和应用的年代，各种单片机围绕着应用对象，不断地在内部添加外围接口电路，如 A/D 转换器、D/A 转换器、PWM、日历时钟、电源监测、程序运行监测电路等，突出地显示出单片机对其应用对象的智能化控制能力。因此单片机的称呼也逐渐从 SCM 变成微控制器 MCU（Micro Controller Unit）。

随着单片机的发展，嵌入式计算机的概念被提了出来。那么单片机与嵌入式计算机有什么区别吗？回答是没有。仅仅是一样东西被冠以不同的名称而已。嵌入式计算机就是“嵌入到对象体系中的专用计算机系统”。由此可见，单片机是根据计算机的特殊结构来命名的，而嵌入式计算机是从计算机形态上来命名的。现在有一个趋势，就是将单片机统一在嵌入式计算机名下。

由于单片机或嵌入式计算机的应用领域被不断地拓展，应用系统的外围电路也不断地被集成进嵌入式计算机中，芯片最大化的趋势十分明显，由此，片上系统 SoC（System on Chip）的概念也应运而生。也就是将嵌入式应用系统中的电路元器件尽可能地集成在一块或几块芯片中，这样做的好处是，应用系统电路板非常简洁，减少体积和功耗，提高可靠性。

1.2 常用数制和码制

本书 1.1.1 节中提到，早期的计算机有两个特点，其中一个特点是计算机中的指令和数据均用二进制数来表示。很多人学计算机都有这样的感觉，计算机从小学就开始学了，用高级语言也编了不少程序，现在也已经很会用计算机了，也早就听说二进制数与计算机关系很密切，可就是没有实际的感受。这究竟是怎么回事呢？这是因为人们以前用的电脑是微型计算机系统，它配备了很多系统软件，正是这些系统软件帮助用户将各种符号转换为二进制数，递交给 CPU 处理。也就是说，CPU 是直接和二进制数打交道的部件，学会用二进制数有利于学习 CPU。

1.2.1 常用的数制及其转换

（1）常用数制

计算机中常用的数制有二进制、十六进制和十进制。实际上 CPU 能直接识别处理的只是二进制数，十六进制和十进制数只是在高级语言、汇编语言编程时使用。

为了区分不同的进制数，采用在数字后面跟一个字母来标识。比如二进制用 B(Binary)、十六进制用 H(Hexdecimal)、十进制用 D(Decimal)（通常省略）。

① 二进制数。计算机中的 CPU 可以对指令代码、数据、地址以及各种符号代码进行处理，这些代码和数都是用二进制数来表示的，用二进制数有以下优点。

a. 二进制数在物理上容易实现。在电路中用高/低电平、正/负极性、开关的开/合等来表示二进制数的“0”和“1”。

b. 二进制数运算规则简单，使运算电路结构也简单。

c. 二进制数的“0”和“1”正好与逻辑代数的“0”和“1”吻合,使逻辑运算方便。

d. 二进制与其他进制的数转换也比较简单。

对任意一个二进制数 N_2 , 均可表示为

$$\begin{aligned} N_2 &= B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 + B_{-1} \times 2^{-1} + \cdots + B_{-m} \times 2^{-m} \\ &= \sum_{i=-m}^{n-1} B_i \times 2^i \end{aligned} \quad (1-1)$$

二进制数的特点是, 基数 $R=2$, 数字符号 $B=\{0, 1\}$, 运算规则是“逢二进一”。

【例 1-1】 写出二进制数 1010.1001B 的多项式表达式。

$$\text{解 } (1010.1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$$

② 十六进制数。由于二进制数表达一个数时其长度比较长, 书写不方便, 记忆也不容易。而十六进制数与二进制数有着天然的联系, 即四位二进制数正好表达一位十六进制数, 这样数字的长度被有效的压缩, 克服了二进制数以上的缺点。与二进制数一样, 任意一个十六进制数 N_{16} , 均可以表示为

$$\begin{aligned} N_{16} &= H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \cdots + H_1 \times 16^1 + H_0 \times 16^0 + H_{-1} \times 16^{-1} + \cdots + H_{-m} \times 16^{-m} \\ &= \sum_{i=-m}^{n-1} H_i \times 16^i \end{aligned} \quad (1-2)$$

十六进制数的特点是, 基数 $R=16$, 数字符号 $H=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, 运算规则是“逢十六进一”。

【例 1-2】 试将 1001111011001.001101010111B 用十六进制数表达。

解 先将题中数的整数部分, 从小数点起向左分成四位一组, 即 1,0011,1101,1001。如果最左面一组数字不满四位, 则在其左边添若干“0”, 使该组也成四位, 变成 0001,0011,1101,1001。接着将其转换成十六进制数, 即 13D9H。然后, 将题中数的小数部分从小数点起向右分成四位一组, 即 0011,0101,0110,1。如果最右面一组数字不满四位, 则在其右边添若干“0”, 使该组也成四位, 变成 0011,0101,0110,1000。接着将其转换成十六进制数, 即 0.3568H。

③ 十进制数。十进制数是人们日常生活中使用最多的数制, 对其表示和运算都很熟悉, 其实十进制数与二进制数类似, 任意一个十进制数 N_{10} , 均可以表示为

$$\begin{aligned} N_{10} &= D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + \cdots + D_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^{n-1} D_i \times 10^i \end{aligned} \quad (1-3)$$

十进制数的特点是, 基数 $R=10$, 数字符号 $D=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, 运算规则是“逢十进一”。

(2) 数制转换

由于初学者对二进制和十六进制数不太熟悉, 往往需要找到二进制与十进制数的对应关系, 在表 1-1 中, 列出了 0~15 各数制的对应关系。

表 1-1 中的内容必须熟记于心, 这样才能在以后的二进制数运算中游刃有余! 除了这些数以外, 对于二进制数与十进制数之间转换的方法也是爱好电脑者必备的功夫。

① 十进制数转换为二进制数。十进制数转换为二进制数的基本思路应该分两步走, 即将整数部分和小数部分分别进行转换, 然后再合并起来。

【例 1-3】 将十进制数 175.625 转换为二进制数。

解 a. 将十进制整数部分转换成二进制整数。采用“除 2 取余法”! 所谓“除 2 取余法”, 就是将十进制数的整数反复除以 2, 每次取其余数。若得到的余数为 1, 则对应的二进制数的相应位

表 1-1 常用进制 0~15 的表示和对应关系

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

也为 1；若得到的余数为 0，则对应二进制数相应位也为 0。第一次得到的余数是二进制数的最低位，最后一次余数是二进制数的最高位，直至商为 0 为止。具体算法如图 1-4 所示。

由此可得， $175=10101111\text{B}$ 。

b. 将十进制小数部分转换成二进制小数。十进制小数部分转换成二进制小数可以采用“乘 2 取整法”！所谓“乘 2 取整法”，就是将十进制小数部分反复乘 2，每次取其整数，若得到的整数位 1，则对应的二进制数的相应位也为 1；若得到的整数为 0，则对应二进制数相应位也为 0。第一次乘 2 得到的整数是二进制数的最高位，从高位到低位逐次进行，直到满足精度要求或乘 2 后的小数部分变 0 为止。具体算法如图 1-5 所示。

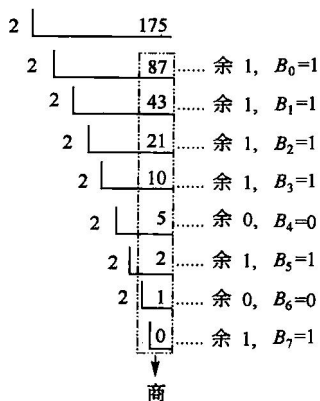


图 1-4 十进制整数转换成二进制整数

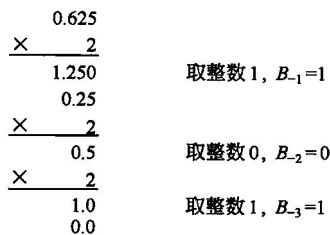


图 1-5 十进制小数转换成二进制小数

由此可得， $0.625=0.101\text{B}$ 。所以， $175.625=10101111.101\text{B}$ 。

② 二进制数转换为十进制数。将二进制数转换为十进制数，其方法是将二进制数“按权展开求和”，便可得到相应的十进制数。

【例 1-4】将二进制数 101101.1011B 转换成十进制数。

$$\begin{aligned} \text{解 } 101101.1011\text{B} &= 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 32 + 8 + 4 + 1 + 0.5 + 0.125 + 0.0625 = 45.6875 \end{aligned}$$

③ 十进制数转换为十六进制数。与十进制数转换为二进制数类似，十进制数转换为十六进制数的基本思路应该分两步走，即将整数部分和小数部分分别进行转换，然后再合并起来。

十进制整数部分转换成十六进制整数的方法与十进制整数转换为二进制整数类似，采用“除

十六取余法”。十进制小数部分转换成十六进制小数的方法与十进制小数转换为二进制小数类似,也可采用“乘十六取整法”。

十进制整数部分转换成十六进制整数也可采用间接的方法,即先将十进制数转换为二进制数,然后再将二进制数转换成十六进制数。

【例 1-5】 将十进制数 175.625 转换为十六进制数。

解 整数部分的转换如图 1-6。

小数部分的转换如图 1-7。

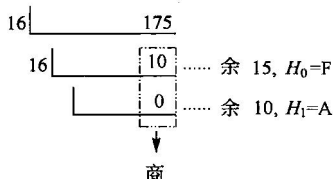


图 1-6 十进制整数转换成十六进制整数

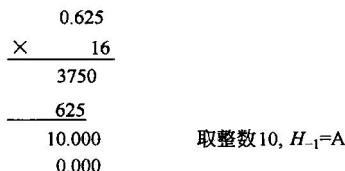


图 1-7 十进制小数转换成十六进制小数

所以, $175.625 = \text{AF.AH}$ 。

如果将数 AF.AH 转换成二进制数,则为 $10101111.1010\text{B} = 10101111.101\text{B}$,此结果与例 1-3 相同。例 1-3 与例 1-5 可以相互验证!

1.2.2 常用的码制

现在的计算机不仅能处理数值领域的问题,而且也能处理大量非数值领域的问题。所有这些都必须编成二进制格式的代码 CPU 才能进行处理。数值数据包括定点数、浮点数、无符号数等。非数值数据包括字母、数字、通用符号、控制符号、汉字等信息,还有逻辑、图形、图像、语音等信息。

这是一个庞大的符号系统,本小节主要涉及数值数据在计算机内的表示方法。为了说明计算机中数的表示方法,先说明一些与此有关的概念。

(1) 相关概念

① 机器数。众所周知,数有很多种类,如整数、小数、正数、负数等。表示这些数都有相应的规定符号。比如, 101101.1011B 是带有小数点的数, -1001100 是一个负数。然而, CPU 只能处理用“1”或“0”这两种符号表示的数,这就需要将原来数据中的一些符号用一些适合于 CPU 处理的方法来表示。最简单的方法是用一位二进制数的“0”和“1”来表示,用“0”代表“+”,用“1”代表“-”。并且,将这个符号放在一个二进制数的最高位,称为符号位。这样计算机中的符号就被数值化了,用这种方法表达的数就是所谓的机器数!

小数点的位置在机器数中隐含在一个固定位置,不占用一个数据位。如何约定小数点的位置,将在定点数中介绍。

【例 1-6】 设 $X = +1100101$, $Y = -1000101$, 则其机器数可分别表示为

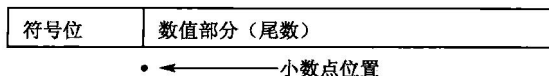
$$X = 01100101, Y = 11000101$$

② 真值。由于机器数将符号进行了数值化,所以机器数的形式值就不等于真正的数值,为了区别起见,将机器数代表的真正的数值称为真值。

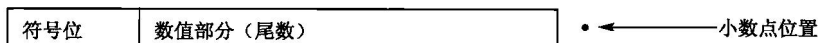
在例 1-6 中机器数 Y 的形式值为 $11000101 = 197$,而其真值为 $-1000101 = -69$ 。

③ 定点数。小数点在数中位置固定不变的数称为定点数。小数点是隐含约定的,不占用数据位。根据小数点位置的约定,分为定点小数和定点整数。

a. 定点小数。当约定小数点位置在机器数的符号位右旁,称定点小数。定点小数是绝对值小于 1 的纯小数,数据格式如下所示。



b. 定点整数。当约定小数点位置在机器数最低位右旁时，称为定点数。小数点是隐含约定的，不占数据位。数据格式如下所示。



本书以后提到的定点数均指定点整数，请读者注意。

④ 浮点数。小数点在数中位置不是固定的数称为浮点数。采用浮点数是为了扩大数的表示范围，其表达方式如下

$$B = \pm M \times R^E$$

式中， M 是尾数，一般为定点小数； R 是底数，因是二进制数，故为 2； E 是指数，也称为阶码。由于尾数的小数点已固定，故整个数的小数点位置就由指数的大小来决定，而指数是根据数值的大小而变化的，因此整个数的小数点位置也是变化的，这就是浮点数的由来。

有了以上这些概念，对于了解计算机中数的表示方法很有益处。下面具体讨论计算机中数的表示方法。

(2) 计算机中数的表示方法

① 无符号数表示方法。当需要处理的数据是正整数时，采用无符号数的表示方法是效率最高，省去符号位，使表达的数值范围更大。若计算机的字长是 8 位，则一个字长的二进制数可表示数的范围是： $0 \leq N \leq 2^8 - 1$ 。

【例 1-7】用 8 位无符号二进制数表示 167、45，然后将这两个数相加。

解 首先将十进制数 167、45 分别转换成 8 位无符号二进制数。

$$167 = 10100111\text{B}, \quad 45 = 00101101\text{B}$$

然后将以上两数相加。

$$\begin{array}{r} 167 = 10100111 \\ +) \quad 45 = 00101101 \\ \hline 212 = 11010100\text{B} = \text{D4H} = 13 \times 16 + 4 = 208 + 4 = 212 \end{array}$$

② 二-十进制数字编码（BCD 码）。在计算机中，一般是采用编码的方式来表示数。用四位二进制数编码表示一位十进制数，称为二-十进制编码-BCD 码（Binary Coded Decimal）。它既有二进制数的形式，又有十进制数的特点。

4 位二进制数有 16 个状态，只需用 10 个状态来表示 BCD 码，不同组合方式可产生不同的 BCD 码，常用的 BCD 码有 8421 码、2421 码、余 3 码、格雷码等。如表 1-2 所示。

表 1-2 几种 BCD 码

十 进 制 数	8421 码	2421 码	余 3 码	格 雷 码
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0011
3	0011	0011	0110	0010
4	0100	0100	0111	0110
5	0101	1011	1000	1110
6	0110	1100	1001	1010
7	0111	1101	1010	1000
8	1000	1110	1011	1100
9	1001	1111	1100	0100

在以上 BCD 码中，8421BCD 码用得最多。因为它就是实际的二进制数，所以只要给出