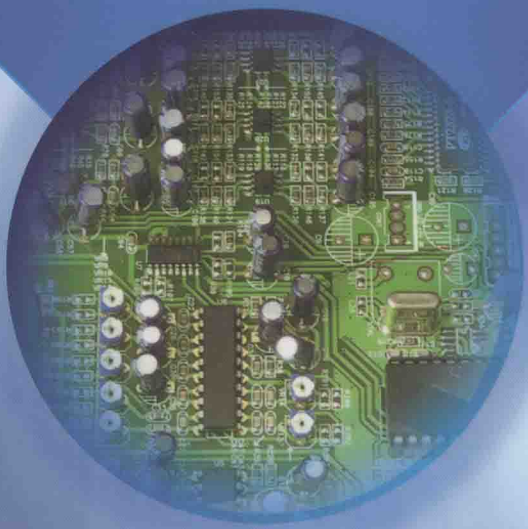
高职高专“十二五”规划教材

DANPIANJI YUANLI JI YINGYONG JIYU CYUYAN

单片机原理及应用

（基于C语言）

吴政江　张定祥　编著



化学工业出版社

高职高专“十二五”规划教材

单片机原理及应用

（基于C语言）

吴政江　张定祥　编著

化学工业出版社

·北京·

全书以 C 语言（C51）为主线，同时兼顾汇编语言，详细介绍了 51 系列单片机的原理及应用知识。本书主要内容包括单片机基础知识、MCS-51 单片机的基本结构、MCS-51 单片机的指令系统及汇编程序设计、单片机的 C51 程序设计、MCS-51 的定时和中断系统、串行接口及串行通信技术、AT89C51 单片机系统扩展、AT89C51 单片机的接口技术以及单片机应用系统设计等。同时，选用了信号灯、流水灯、数字钟、计数器、数字电压表、信号发生器、单片机双机通信、球赛记分牌等十个单片机典型应用作为实训，并且所有实训均在 Keil C51 μVision4 和 ISIS7 professional 环境下调试通过。

本书内容简洁实用，讲解通俗易懂，并有大量应用实例，实用性强，既可作为高职高专院校通信类、信息类、应用电子类、控制类、仪器仪表类以及机电类等专业单片机课程的教材，也可作为从事单片机开发应用的工程技术人员的参考书。

图书在版编目(CIP)数据

单片机原理及应用（基于 C 语言）/ 吴政江，张定祥编著. —北京：化学工业出版社，2013.5

高职高专“十二五”规划教材

ISBN 978-7-122-16929-7

Ⅰ. ①单… Ⅱ. ①吴… ②张 Ⅲ. ①单片微型计算机-C 语言-程序设计-高等职业教育-教材 Ⅳ. ①TP368.1 ②TP312

中国版本图书馆 CIP 数据核字（2013）第 065694 号

责任编辑：王听讲　　文字编辑：孙　科

责任校对：边　涛　　装帧设计：韩　飞

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 19¼　字数　498　千字　　2013 年 8 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：35.00 元

前　言

由于单片机的应用十分广泛，所以各出版社相继出版了很多关于单片机方面的教材。可以将众多的单片机教材分为传统章节式和项目驱动式两种类型，二者各有特点。传统章节式教材逻辑严谨，知识前后连贯，便于学生自学。但传统章节式教材往往偏重于理论，而且理论较深，不注重实践训练，因而不适用于高职高专学生学习。而纯项目式教材又过分注重实践训练，不注重理论知识的讲授，其所讲授的理论知识不连续、不完整，且前后不连贯，使学生学完后只会做教材上的项目，而不能触类旁通。鉴于此，我们选用师生反映良好的教案，采用传统章节式与项目式相结合的方式编写了这本基于 C 语言的单片机原理及应用教材。它综合了传统章节式与项目式两者的优点，既有较系统的理论，又有丰富的项目实践，使学生在学中做、做中学，最终掌握系统的单片机知识。本教材紧密结合高职高专教育的特点，在理论知识够用的基础上重点培养学生的实践能力，坚持理论与实践相结合，理论为实践服务，使学生毕业后尽可能实现学校与企业之间的无缝连接。

本书的编写者长期从事高等职业教育，是担任“单片机原理及应用”课程讲授的第一线教学人员和单片机应用系统开发设计人员，对单片机的历史过程和未来发展，以及单片机应用技术的途径与方法，有着较深的理解和感受。在本教材的内容取材上既考虑到单片机的基本知识，同时又充分体现了单片机的新知识与新技术。全书具有语言精练，内容新而全面，通俗易懂，结构新颖等特点。既方便教师灵活组织教学，又方便学生自学，学生通过本课程的学习能够达到熟练掌握单片机应用技术的目的。

采用汇编语言编程有利于加强对单片机的理解，而 C51 在功能、结构上以及可读性、可移植性、可维护性方面更有非常明显的优势。因此，本书以 C 语言（C51）为主线，同时兼顾汇编语言，详细介绍了 51 系列单片机的原理及应用知识。本教材在内容安排上由浅入深，循序渐进，保证即使只有《模拟电路》、《数字电路》基础的读者，也能很快入门，掌握单片机应用技术。本书主要内容包括单片机基础知识、MCS-51 单片机的基本结构、MCS-51 单片机的指令系统及汇编程序设计、单片机的 C51 程序设计、MCS-51 的定时和中断系统、串行接口及串行通信技术、AT89C51 单片机系统扩展、AT89C51 单片机的接口技术以及单片机应用系统设计等。同时，选用了信号灯、流水灯、数字钟、计数器、数字电压表、信号发生器、单片机双机通信、球赛记分牌等十个单片机典型应用作为实训，并且所有实训均在 Keil C51 μVision4 和 ISIS7 professional 环境下调试通过。学习单片机就是为了能开发与制作有具体意义的单片机应用系统。学生通过实训，可以强化单片机应用系统的概念，从而使得单片机的学习与应用变得更简单。

我们将为使用本书的教师免费提供电子教案，需要者可以到化学工业出版社教学资源网站 http://www.cipedu.com.cn 免费下载使用。

本书由贵州电子信息职业技术学院吴政江与张定祥编著，共同对本书的编写思路与内容进行了整体规划。具体分工为：吴政江编写第 1～5 章、第 9 章，并负责全书的统稿和审稿工

作；张定祥编写第 6～8 章。本书在编写过程中得到了贵州电子信息职业技术学院领导与同行的大力帮助和支持，在此表示诚挚的感谢。同时，对本书所引用的文献资料的各位作者表示诚挚的感谢。

由于编写者学识水平有限，加之编写时间仓促，疏漏与不妥之处难免，殷切希望广大读者给予批评指正。

编　者

2013 年 3 月

目　录

第 1 章　单片机基础知识 ……1
1.1　单片机的数学基础 ……1
1.1.1　数的进制及其相互转换 ……1
1.1.2　带符号数的表示方法 ……4
1.1.3　溢出的判别方法 ……5
1.1.4　ASCII 码和 BCD 码 ……7
1.2　单片机基础 ……9
1.2.1　计算机的经典组成 ……9
1.2.2　单片机的概念 ……11
1.2.3　单片机的应用范围 ……13
1.2.4　单片机的发展 ……14
1.2.5　单片机系统 ……15
1.2.6　单片机与嵌入式系统 ……18
1.3　常用单片机系列介绍 ……19
1.3.1　MCS-51 系列 ……19
1.3.2　MC68 系列(Motorola 公司的 8 位单片机) ……20
1.3.3　PIC16 系列 ……22
1.3.4　MSP430 系列 ……23
1.3.5　AVR 系列 ……23
1.3.6　STC12C5A60S2 系列 ……26
实训一　了解单片机开发环境 ……27
小结 ……30
习题与思考题 ……31
第 2 章　MCS-51 单片机的基本结构 ……33
2.1　MCS-51 单片机的内部结构 ……33
2.1.1　中央处理器（CPU） ……33
2.1.2　存储器 ……36
2.1.3　I/O 端口 ……36
2.1.4　定时器/计数器 ……37
2.1.5　中断系统 ……37
2.1.6　内部总线 ……37
2.2　AT89C51 单片机引脚及其功能 ……37
2.2.1　I/O 端口功能 ……38
2.2.2　电源线 ……41
2.2.3　外接晶体引脚 ……41
2.2.4　控制线 ……41
2.3　AT89C51 存储器 ……42
2.3.1　程序存储器 ……42
2.3.2　数据存储器 ……42
2.4　AT89C51 单片机最小应用系统 ……45
2.4.1　复位电路 ……46
2.4.2　时钟电路 ……47
2.4.3　电源电路 ……48
实训二　单片机控制信号灯亮灭 ……50
小结 ……51
习题与思考题 ……52
第 3 章　MCS-51 单片机的指令系统及汇编程序设计 ……53
3.1　指令格式和寻址方式 ……53
3.1.1　汇编语言指令格式 ……53
3.1.2　符号注释 ……54
3.1.3　寻址方式 ……54
3.2　AT89C51 指令系统 ……56
3.2.1　数据传送类指令 ……56
3.2.2　算术运算类指令 ……59
3.2.3　逻辑操作与移位指令 ……62
3.2.4　控制转移类指令 ……64
3.2.5　位操作类指令 ……66
3.3　汇编语言程序设计 ……68
3.3.1　汇编语言的构成 ……68
3.3.2　汇编语言程序设计和汇编 ……71
3.3.3　顺序程序设计 ……72
3.3.4　分支程序设计 ……74
3.3.5　循环程序设计 ……77
3.3.6　子程序设计 ……82
3.4　汇编语言程序设计举例 ……84
3.4.1　查表程序设计 ……84
3.4.2　数据检索程序设计 ……87
3.4.3　运算程序设计 ……88

实训三　单片机控制流水灯(汇编程序)…… 91
小结 …… 92
习题与思考题 …… 92
第 4 章　单片机的 C51 程序设计 …… 95
4.1　C51 程序的结构特点 …… 95
4.1.1　C 语言与汇编语言的比较 …… 95
4.1.2　C51 程序的结构特点 …… 95
4.2　C51 语法基础 …… 97
4.2.1　C 语言词汇 …… 97
4.2.2　编译预处理 …… 98
4.3　C51 的数据类型、存储类型及常量与变量 …… 100
4.3.1　C51 的数据类型 …… 100
4.3.2　C51 的数据存储类型 …… 101
4.3.3　常量与变量 …… 101
4.4　C51 对单片机主要资源的定义 …… 102
4.4.1　使用关键字定义特殊功能寄存器（SFR） …… 103
4.4.2　通过头文件访问特殊功能寄存器（SFR） …… 103
4.4.3　扩展 I/O 端口或片外 RAM 的直接访问 …… 106
4.4.4　定义和使用位变量 …… 106
4.5　C51 的基本运算 …… 107
4.5.1　C51 的算术运算 …… 107
4.5.2　C51 的关系运算 …… 108
4.5.3　C51 的逻辑运算 …… 108
4.5.4　C51 的位运算 …… 109
4.5.5　C51 的赋值运算 …… 109
4.6　C51 的构造数据类型 …… 110
4.6.1　数组 …… 110
4.6.2　结构 …… 111
4.6.3　联合 …… 113
4.7　C51 的流程控制语句 …… 116
4.7.1　选择控制语句 …… 116
4.7.2　循环控制语句 …… 118
4.8　C51 的函数 …… 122
4.8.1　函数的分类与定义 …… 122
4.8.2　函数的调用 …… 125
4.9　指针 …… 127
4.9.1　内存单元、地址和指针 …… 127
4.9.2　指针变量的定义、赋值与引用 …… 128
4.9.3　指针与数组 …… 131
4.9.4　指针变量作为函数的参数 …… 133
4.10　C51 程序设计举例 …… 135
4.10.1　在 C51 中加入汇编语言语句 …… 135
4.10.2　LED 动态显示驱动程序设计 …… 136
实训四　单片机控制流水灯(C51 程序) …… 141
实训五　计数器的 C51 程序设计与制作 …… 143
小结 …… 146
习题与思考题 …… 146
第 5 章　MCS-51 的定时与中断系统 …… 148
5.1　MCS-51 的中断系统及其应用 …… 148
5.1.1　中断的概念 …… 148
5.1.2　MCS-51 单片机中断系统的结构 …… 150
5.1.3　中断响应 …… 153
5.1.4　C51 的中断服务函数与寄存器组选择 …… 156
5.2　定时器/计数器 …… 159
5.2.1　定时器/计数器的结构及工作原理 …… 159
5.2.2　定时器/计数器的控制 …… 160
5.2.3　定时器/计数器的编程和应用 …… 163
实训六　可调时间数字钟的设计与制作 …… 166
小结 …… 171
习题与思考题 …… 171
第 6 章　串行接口及串行通信技术 …… 173
6.1　串行通信基础知识 …… 173
6.1.1　串行通信的基本概念 …… 173
6.1.2　串行通信的制式 …… 173
6.1.3　串行通信的分类 …… 174
6.2　AT89C51 单片机的串行接口 …… 175
6.2.1　串行接口的结构与控制 …… 175
6.2.2　串行接口的工作方式 …… 177
6.2.3　串行接口的波特率设计 …… 178
6.3　AT89C51 单片机串行通信举例 …… 180
6.3.1　双机通信 …… 180
6.3.2　多机通信 …… 183
6.3.3　PC 机与单片机间的串行通信 …… 184
6.4　串行通信总线标准及 RS-232C 接口 …… 187
实训七　单片机间的双机通信 …… 191

小结 …… 195
习题与思考题 …… 195
第 7 章　AT89C51 单片机系统扩展 …… 197
7.1　AT89C51 单片机系统扩展及结构 …… 197
7.1.1　系统总线 …… 197
7.1.2　存储器扩展的编址技术 …… 198
7.2　AT89C51 单片机的存储器扩展 …… 202
7.2.1　程序存储器的扩展 …… 202
7.2.2　数据存储器的扩展 …… 205
7.3　并行 I/O 口扩展 …… 207
7.3.1　基本 I/O 口的扩展 …… 207
7.3.2　可编程 I/O 口芯片 8255 及应用 …… 210
实训八　用 8255 芯片实现接口扩展 …… 214
小结 …… 216
习题与思考题 …… 216
第 8 章　AT89C51 单片机的接口技术 …… 218
8.1　键盘接口技术 …… 218
8.1.1　键盘的工作原理 …… 218
8.1.2　独立式按键 …… 220
8.1.3　矩阵式按键 …… 221
8.2　显示器接口技术 …… 225
8.2.1　LED 显示器及其接口 …… 225
8.2.2　LED 点阵显示器及其接口 …… 230
8.2.3　LCD 显示器及其接口 …… 239
8.3　A/D 转换器及其接口技术 …… 251
8.3.1　A/D 转换器概述 …… 251
8.3.2　典型 A/D 转换器芯片 ADC0809 …… 253
8.3.3　ADC0809 与 AT89C51 单片机的接口及应用 …… 255
8.4　D/A 转换器及其接口技术 …… 257
8.4.1　D/A 转换器概述 …… 257
8.4.2　典型 D/A 转换器芯片 DAC0832 …… 259
8.4.3　DAC0832 与 AT89C51 单片机的接口及应用 …… 260
实训九　基于 ADC0832 的数字电压表 …… 263
小结 …… 271
习题与思考题 …… 271
第 9 章　单片机应用系统设计 …… 272
9.1　单片机应用系统开发的一般方法 …… 272
9.1.1　确定任务 …… 272
9.1.2　总体设计 …… 272
9.1.3　硬件设计 …… 273
9.1.4　软件设计 …… 274
9.1.5　系统的仿真调试与运行 …… 275
9.2　单片机应用系统的开发工具 …… 276
9.2.1　单片机开发系统的组成结构 …… 276
9.2.2　单片机开发系统的功能 …… 277
9.2.3　单片机开发系统的类型 …… 278
9.3　单片机应用系统举例 …… 280
9.3.1　信号发生器设计 …… 280
9.3.2　抢答器系统设计 …… 283
实训十　单片机控制球赛记分牌的设计 …… 288
小结 …… 295
习题与思考题 …… 296
参考文献 …… 297

第 1 章　单片机基础知识

教学目标和要求：单片机是现代电子智能仪器仪表及嵌入式系统的主要组成部分，应用非常广泛，是现代工程技术人员必须掌握的知识之一。本章要求掌握数的进制及其相互转换、带符号数的表示方法、溢出的判别方法、ASCII 码和 BCD 码等单片机的数学基础知识；掌握单片机的概念、特点、应用范围、发展历程等基础知识；了解常用单片机系列，为后续章节的学习打下基础。

关键词汇：基数、二进制、溢出、代码、补码、反码、原码、单片机。

1.1　单片机的数学基础

1.1.1　数的进制及其相互转换

（1）数的几种常用进制

数制是人们利用数码符号来计数的方法。有很多种数制，人们熟悉的是十进制，常用的也是十进制。但由于数在机器中是以器件的物理状态来表示的，所以一个具有两种稳定状态且能相互转换的器件，就可以用来表示一位二进制数。二进制数的表示是最简单而且是最可靠的，其运算规则也是最简单的。因此，迄今为止，所有计算机中都是以二进制进行算术运算和逻辑运算的。但是在使用二进制编写程序时既烦锁又容易出错，所以人们在编写程序时又经常用到十进制、十六进制或八进制。下面分别予以介绍。

任何一种数制都有两个要素，即基数和权。基数为数制中所使用的数码符号的个数。当基数为 R 时，该数制可使用的数码为 0～（R–1）。例如在二进制中基数为 2，可使用 0 和 1 两个数码。在进行算术运算时按逢 R 进一，借 1 当 R 的规则进行。权是数制中某一数位上单位数的大小，它是一个幂，底数是基数 R，指数是数码符号的位置号。需要注意的是数码的位置号是整数。将一个数中某一位的数码与该位的权相乘，即为该位数码的数值。将所有数码的数值相加就得该数的大小。其相加的算术式就叫该数的加权系数表达式。如对于 R 进制的数 M，其加权系数表达式为

$$M = k_n R^n + k_{n-1} R^{n-1} + \cdots + k_1 R^1 + k_0 R^0 + k_{-1} R^{-1} + \cdots + k_{-m} R^{-m} = \sum_{i=-m}^{n} K_i \times R^i$$

① 十进制（Decimal）。十进制是以 10 为基数，逢十进一、借一当十的计数体制。计数符号共有十个，分别为：0、1、2、3、4、5、6、7、8、9。计数规则是逢十进一，借一当十。十进制数常用下标 D 或 10 表示。

加权系数表示：$[M]_{\mathrm{D}} = k_n 10^n + k_{n-1} 10^{n-1} + \cdots + k_1 10^1 + k_0 10^0 + k_{-1} 10^{-1} + \cdots + k_{-m} 10^{-m} = \sum_{i=-m}^{n} K_i \times 10^i$

K_i 为第 i 位的系数可取 0～9 十个数字符号中的任一个；10^i 为第 i 位的权。显然各位的权是 10 的幂。m、n 是数码符号的位置号，均为整数。

【例 1-1】 $(4258)_{10}=4\times10^3+2\times10^2+5\times10^1+1\times10^0$

$(97.51)_{10}=9\times10^1+7\times10^0+5\times10^{-1}+1\times10^{-2}$

② 二进制（Binary）。二进制是以 2 为基数，逢二进一、借一当二的计数体制。计数符号共有两个，分别为：0、1。计数规则是逢二进一、借一当二。二进制数常用下标 B 或 2 表示。

运算规则：0+0=0　0+1=1+0=1　1+1=10（读“壹零”）

0×0=0　1×0=0×1=0　1×1=1

加权系数表示：$[M]_B = k_n 2^n + k_{n-1}2^{n-i} + \cdots + k_1 2^1 + k_0 2^0 + k_{-1}2^{-1} + \cdots + k_{-m}2^{-m} = \sum_{i=-m}^{n} K_i \times 2^i$

K_i 为第 i 位的系数可取 0 或 1；2^i 为第 i 位的权，即各位的权是 2 的幂。m、n 是数码符号的位置号，均为整数。

【例 1-2】 $[1101101]_B = 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$[1101.01]_B = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$

下面再介绍有关二进制的几个概念。

位：一位二进制信息，只能是 0 或 1，也叫比特(bit)。

字节：计算机中将 8 位二进制数称为一个字节，也叫拜特（Byte）。存储器容量就是以字节为单位的。但字节太小了，不方便用于表示大容量存储器。为了表示大容量存储器的需要，人们还定义了千字节（KB）、兆字节（MB）、吉字节（GB）、太字节（TB）等单位。它们的关系为：

1KB=2^{10}Byte=1024Byte

1MB=2^{10}KB=1024KB=2^{20}Byte

1GB=2^{10}MB=1024MB=2^{20}KB =2^{30}Byte

1TB=2^{10}GB=1024GB=2^{20}MB =2^{30}KB =2^{40}Byte

字：计算机进行一次运算最多能处理的二进制位数称为一个字，也叫沃德（Word）。字是计算机中参加运算的基本单位。由于 16 位微型计算机长期占据主导地位，所以通常认为一个字为 16 位二进制数，即 1 Word=2 Byte。但对 8 位或 32 位微型计算机，一个字应为 8 位或 32 位二进制数。

③ 八进制（Octal）。八进制是以 8 为基数，逢八进一、借一当八的计数体制。计数符号共有八个，分别为：0、1、2、3、4、5、6、7。计数规则是逢八进一，借一当八。八进制数常用下标 O 或 8 表示。

加权系数表示：$[M]_O = k_n 8^n + k_{n-1}8^{n-1} + \cdots + k_1 8^1 + k_0 8^0 + k_{-1}8^{-1} + \cdots + k_{-m}8^{-m} = \sum_{i=-m}^{n} K_i \times 8^i$

K_i 为第 i 位的系数可取 0～7 八个数字符号中的任一个；8^i 为第 i 位的权。显然各位的权是 8 的幂。m、n 是数码符号的位置号，均为整数。

【例 1-3】 $[236]_O = 2\times8^2 + 3\times8^1 + 6\times8^0$

$[147.52]_O = 1\times8^2 + 4\times8^1 + 7\times8^0 + 5\times8^{-1} + 2\times8^{-2}$

八进制数有一个重要特点，那就是每位八进制数可用三位二进制数表示，反之亦然。例如：$(6)_8=(110)_2$、$(101)_2=(5)_8$。

④ 十六进制（Hexadecimal）。十六进制是以 16 为基数，逢十六进一、借一当十六的计数体制。计数符号共有十六个，分别为：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。计数规则是逢十六进一，借一当十六。十六进制数常用下标 H 或 16 表示。

加权系数表示：$[M]_H = k_n 16^n + k_{n-1}16^{n-1} + \cdots + k_1 16^1 + k_0 16^0 + k_{-1}16^{-1} + \cdots + k_{-m}16^{-m} = \sum_{i=-m}^{n} K_i \times 16^i$

K_i 为第 i 位的系数可取 0～F 十六个数字符号中的任一个；16^i 为第 i 位的权。显然各位的权是 16 的幂。m、n 是数码符号的位置号，均为整数。

【例 1-4】 $[4E6]_H=4\times16^2+14\times16^1+6\times16^0$

$[9B.C8]_H=9\times16^1+11\times16^0+12\times16^{-1}+8\times16^{-2}$

十六进制数有一个重要特点，那就是每位十六进制数可用四位二进制数表示，反之亦然。例如：$(E)_{16}=(1110)_2$、$(1011)_2=(B)_{16}$。

（2）不同进制数之间的相互转换。

① 任意进制数转为十进制数。方法：按权展开求和。

【例 1-5】 $[1101.01]_B=1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}=(13.25)_D$

$[236]_8=2\times8^2+3\times8^1+6\times8^0=128+24+6=(158)_{10}$

$[C2]_H=12\times16^1+2\times16^0=192+2=(194)_{10}$

② 十进制数转为二进制数。方法：对整数部分，连续除 2 取余反排列，直到商为 0；对小数部分，连续乘 2 取整正排列，直到乘积的小数部分为 0 或满足误差要求。

【例 1-6】 $[338.38]_D=[101010010.01100]_B$

$[25.706]_D=[11001.10111]_B$（保留 5 位小数）。

```
2 | 25   …余 1   整数最低位          0.706
2 | 12   …余 0                      × 2
  2 | 6  …余 0                      1.412   …取整 1   小数最高位
  2 | 3  …余 1                      0.412
  2 | 1  …余 1   整数最高位        × 2
    0                               0.824   …取整 0
                                    × 2
                                    1.648   …取整 1
                                    0.648
                                    × 2
                                    1.296   …取整 1
                                    0.296
                                    × 2
                                    0.592   …取整 0   小数最低位
```

由于最后舍弃的数 0.592 大于 0.5，所以按“四舍五入”原则小数最低位 0 加 1。

推广：十进制数转为任意进制数。整数部分，连续除基数取余反排列，直到商为 0；小数部分，连续乘基数取整正排列，直到乘积的小数部分为 0 或满足误差要求。

③ 八进制数与二进制数之间的相互转换。二进制转为八进制：对整数部分，从最低位开始三位三位进行分组，不足三位的前面补零；对小数部分，则从最高位开始三位三位进行分组，不足三位的后面补 0。然后每组以其对应的八进制数代替，排列顺序不变。

八进制转为二进制：将每位八进制数写成对应的三位二进制数，再按原来的顺序排列起来即可。

【例 1-7】 $[11110100010]_2=[3642]_8$　　$[6403]_8=[110100000011]_2$

④ 十六进制数与二进制数之间的相互转换。方法：跟八进制数与二进制数之间的相互转换相似，只是按四位分组即可。

【例 1-8】 $[11110100010]_2=[7A2]_{16}$　　$[B59]_{16}=[101101011001]_2$

⑤ 八进制数与十六进制数之间的相互转换。方法：通过二进制数作中间变量进行变换。

【例 1-9】 $[B59]_{16}=[101101011001]_2=[5531]_8$

$[6403]_8=[110100000011]_2=[D03]_{16}$

1.1.2 带符号数的表示方法

（1）机器数与真值

前面提到的二进制数，没有涉及符号问题，是一种无符号数。但在实际应用中，一个数显然还有正、负之分，那么符号在计算机中是怎么表示的呢？计算机中采用二进制数，对于数的符号“+”或“–”也用二进制数码表示。规定用二进制数码的最高位表示符号（称为符号位）。并规定：用数码“0”表示正数的符号“+”；用数码“1”表示负数的符号“–”。这样得到的数就称为有符号数。

【例 1-10】 X_1=+010001B；X_2= –010001B，在 8 位机中分别表示为 X_1=00010001B；X_2=10010001B。

一个数在机器中的表示形式称为机器数，而原来的实际数本身称为机器数的真值。在计算机中常用的机器数有原码、反码、补码三种形式。当真值为 X 时，其原码、反码、补码分别用$[X]_{原}$、$[X]_{反}$、$[X]_{补}$表示。

（2）原码（true form）

符号位用“0”表示正数，“1”表示负数，其余各位表示真值除符号外的尾数本身，这种表示方法称为原码表示法。即用 0、1 分别代替真值中的“+”、“–”即得原码。以八位机为例（下同）。

① 对于正数。$[X]_{原}=X$

【例 1-11】 若 X_1=+1101001B，X_2=+101101B，则$[X_1]_{原}$=01101001B，$[X_2]_{原}$=00101101B（不足 8 位应在符号位后补“0”）。

② 对于负数。$[X]_{原}=2^{8-1}-X$

【例 1-12】 若 X_1= –1101001B，X_2= –101101B，则

$[X_1]_{原}$ =11101001B=10000000B+1101001B=2^{8-1}–（–1101001B）=$2^{8-1}-X_1$

$[X_2]_{原}$ =10101101B=10000000B+101101B=2^{8-1}–（–101101B）= $2^{8-1}-X_2$

③ 对于 0。在计算机中，0 可认为它是+0，也可认为它是–0，故 0 在原码中有两种表示法。对八位机：$[+0]_{原}$=00000000B，$[–0]_{原}$=10000000B。

字长为 n 位的原码表示法的一般规律：

$$[X]_{原}=\begin{cases} X & (0\leqslant X<2^{n-1}) \\ 2^{n-1}-X & (-2^{n-1}<X\leqslant 0) \end{cases}$$

（3）反码（one's complement）

① 对于正数。其反码表示法与原码相同，即$[X]_{反}=[X]_{原}=X$。

【例 1-13】 若 X_1=+1101001B，X_2=+101101B，则$[X_1]_{反}=[X_1]_{原}$=01101001B，

$[X_2]_{反}=[X_2]_{原}$=00101101B（不足 8 位应在符号位后补“0”）。

② 对于负数。反码等于其原码符号位不变，其余各位按位取反（即“1”换成“0”，“0”换成“1”）。也可按以下公式计算：$[X]_{反}=2^8-1+X$。

【例 1-14】 若 X= –1101001B，则$[X]_{原}$ =11101001B，

$[X]_{反}$ =10010110B=2^8–1+（–1101001B）=2^8–1–1101001B。

③ 对于 0。反码有$[+0]_{反}$和$[–0]_{反}$两种表示法。对于 8 位机：$[+0]_{反}$=00000000B、$[–0]_{反}$

=11111111B。

字长为 n 位的反码表示法的一般规律：

$$[X]_{反}=\begin{cases}X & (0\leqslant X<2^{n-1})\\ 2^n-1+X & (-2^{n-1}<X\leqslant 0)\end{cases}$$

（4）补码（two's complement）

补码表示法可以把负数转换为正数，使减法转换为加法，从而使正负数的加减运算转换为单纯的正数相加的运算。因此，计算机中一般采用补码表示法。

① 对于正数。其补码就是该正数本身，即$[X]_{补}=X$

【例 1-15】 若 X=+1101001B，则$[X]_{补}$=01101001B

② 对于负数。其补码等于其反码加 1。即$[X]_{补}=[X]_{反}+1=2^n-1+X+1=2^n+X$(对八位机 n=8)。

【例 1-16】 若 X= –1101001B，则$[X]_{原}$=11101001B，$[X]_{反}$=10010110B，

$[X]_{补}$=10010110B+1=10010111B=$2^8+X=2^8+$（–1101001B）= 2^8–1101001B。

③ 对于 0。$[+0]_{补}=[-0]_{补}$=00000000B，即 0 的补码只有一种表示法。

字长为 n 位的补码表示法的一般规律：

$$[X]_{补}=\begin{cases}X & (0\leqslant X<2^{n-1})\\ 2^n+X & (-2^{n-1}<X\leqslant 0)\end{cases}$$

综上所述，对正数有$[X]_{原}=[X]_{反}=[X]_{补}=X$；对负数，用“1”代替负号“–”就得原码，再对原码除符号位（最高位）外其余各位按位取反就得反码，最后对反码加 1 就得补码。

（5）已知机器数求真值

① 先求原码。对正数(符号位为 0)，原码、反码、补码相同，无需转换；对负数(符号位为 1)，反码的数值位按位取反，可转换为原码，补码的数值位按位取反后末位加 1，可转换为原码。

② 由原码求真值。用“+”、“–”代替原码的符号位(“0”换为“+”，“1”换为“–”)即可。

【例 1-17】 若$[X]_{补}$=10011010B，求 X?

解：因符号位为 1，所以 X 为负数。

则$[X]_{原}$=11100101B+1=11100110B，X= –1100110B=$(-102)_{10}$。

1.1.3　溢出的判别方法

（1）计算机中带符号数的加减法运算

在微型计算机中，原码表示的数易于识别，但在做加减法运算时比较复杂，符号位和数值位需要分别处理。首先做两个数绝对值的减法，用绝对值大的数减去绝对值小的数，然后用绝对值大的数的符号作为结果的符号。采用补码做加减法运算时，符号位与数值位同时参与运算，减法也转换为加法运算，符号位无需单独处理。

① 补码加法运算。补码加法运算的规则是：$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

【例 1-18】 已知 X=1100011B，Y= –111B，求 $X+Y$=?

解：$[X]_{补}$=01100011B，$[Y]_{补}$=11111101B

$$\begin{array}{r}[X]_{补}=01100011\text{B}\\ +)\quad [Y]_{补}=11111101\text{B}\\ \hline \text{模溢出}\rightarrow \boxed{1}\,01100000\text{B}\end{array}$$

所以$[X+Y]_{补}=[X]_{补}+[Y]_{补}$=01100000B，$X+Y$=+1100000B，最高位的进位自动丢失（称为溢出）。

② 补码减法运算。在微型计算机中减法运算也通过补码转换为加法运算，减法运算的规则是：

$[X-Y]_{补}=[X+(-Y)]_{补}=[X]_{补}+[-Y]_{补}$，其中$[-Y]_{补}$可由$-Y$求出，也可以由$[Y]_{补}$求出。把$[Y]_{补}$的符号位与数值位一起取反，末位加 1，结果就等于$[-Y]_{补}$ 。

【例 1-19】 已知 X=1000111B，Y=1001B，求 $X-Y$=?

解：$[X]_{补}$=01000111B，$[Y]_{补}$=00001001B，$[-Y]_{补}$=11110111B

$$\begin{array}{r} [X]_{补}=01000111\text{B} \\ +)\quad [-Y]_{补}=11110111\text{B} \\ \hline \text{模溢出}\rightarrow \boxed{1}\ 00111110\text{B} \end{array}$$

$[X-Y]_{补}$=00111110B，$X-Y$=+0111110B

（2）溢出的判别方法

① 溢出的概念。在计算机内部表示数据与人工表示数据的情况不同。人工表示数据时，数据的值可以为任意大小，而在计算机内只能用有限位数来表示数据。所以计算机中所能表示的数有一定的范围，对于绝对值太大而超过一定值的数，计算机无法表示，这时会造成数据的最高位丢失，数据产生错误，这种情况称为上溢出。出现上溢出时，应停止运算，进行错误处理。对于绝对值太小的数，在计算机中同样也表示不出来，此时计算机将这个数作为 0 处理，数据产生误差，这种情况称为下溢出。由于下溢出所带来的误差很小，在允许范围之内，可不作错误处理。所以在以后提到的溢出指的是上溢出。

② 溢出的判断。当两个数作加减法运算时，如何判断运算结果是否有溢出呢？常用的有补码和变形补码两种方法。

a．补码判断法。两个用补码表示的数作加减法运算时，如果是同号相减或异号相加，只能使数据的绝对值越来越小，运算结果不可能产生溢出；如果是同号相加或异号相减，则运算结果可能会出现溢出。此时，可以把运算结果的符号与参与运算的数据符号相比较，如果出现正数加正数得负数或负数加负数得正数的情况，则可以断定运算结果出现了溢出。

【例 1-20】 已知 X=1110010B，Y=1001101B，求 $X+Y$=?

解：$[X]_{补}$=01110010B，$[Y]_{补}$=01001101B

$$\begin{array}{r} [X]_{补}=01110010\text{B} \\ +)\quad [Y]_{补}=01001101\text{B} \\ \hline 10111111\text{B} \end{array}$$

由运算结果可以看出，两个正数相加，结果为负数，可以断定是溢出造成的。出现溢出时运算结果是错误的，不再使用。

【例 1-21】 已知 X= –1100111B，Y=1001100B，求 $X-Y$=?

解：$[X]_{补}$=10011001B，$[Y]_{补}$=01001100B，$[-Y]_{补}$=10110100B

$$\begin{array}{r} [X]_{补}=10011001\text{B} \\ +)\quad [-Y]_{补}=10110100\text{B} \\ \hline \text{模溢出}\rightarrow \boxed{1}\ 01001101\text{B} \end{array}$$

由运算结果可以看出，两个负数相加，结果为正数，可以断定是溢出造成的。出现溢出时运算结果是错误的，不再使用。

【例 1-22】 已知 X=1100111B，Y=1110011B，求 $X-Y$=?

解：$[X]_{补}$=01100111B，$[Y]_{补}$=01110011B，$[-Y]_{补}$=10001101B

$$\begin{array}{rl} & [X]_{补}=01100111\text{B} \\ +) & [-Y]_{补}=10001101\text{B} \\ \hline & 11110100\text{B} \end{array}$$

由于是同号相减，运算结果不可能产生溢出。所以$[X+Y]_{补}$=11110100B，$X+Y=-0001100$B

b．变形补码判断法。变形补码是采用双符号位表示的补码，用 00 表示正数，用 11 表示负数。用变形补码判断运算结果是否有溢出时，只需要判断结果的双符号位是否相同即可。如果双符号位相同，运算结果没有溢出，否则运算结果有溢出。

【例 1-23】 已知 $X=-1100111$B，Y=1001100B，求 $X+Y$=？和 $X-Y$=？

解：$[X]_{变形补}$=110011001B，$[Y]_{变形补}$=001001100B，$[-Y]_{变形补}$=110110100B

$$\begin{array}{rl} & [X]_{变形补}=110011001\text{B} \\ +) & [Y]_{变形补}=001001100\text{B} \\ \hline & 111100101\text{B} \end{array}$$

双符号位相同，结果无溢出，$X+Y=-0011011$B。

$$\begin{array}{rl} & [X]_{变形补}=110011001\text{B} \\ +) & [-Y]_{变形补}=110110100\text{B} \\ \hline & 101001101\text{B} \end{array}$$

双符号位不同，结果溢出。出现溢出时运算结果是错误的，不再使用。

c．进位或借位判断法（以八位机为例）。在进行补码加减运算时如果最高位（即第七位）与次高位（即第六位）所产生的进位或借位相同，则运算结果没有溢出，否则运算结果有溢出。用公式表示为：$OV=C6\oplus C7$。

1.1.4　ASCII 码和 BCD 码

（1）二进制代码

数码符号不仅可以用于计数表示数值的大小，而且可以用于表示特定的对象。如电话号码、邮政编码、手机号码等就是用 0～9 这十个十进制数码符号的组合来表示特定的对象，可以称为十进制代码。同样，由 0 和 1 组成的二进制数码不仅可以表示数值的大小，而且可以用来表示特定的信息。这种具有特定含义的二进制数码称为二进制代码。建立这种代码与它表示的对象（如十进制数、字母、特定符号、逻辑值等）的一一对应关系过程称为编码；将代码所表示的特定信息翻译出来称为译码，分别由编码器、译码器来实现。

（2）二-十进制码（BCD 码）

二-十进制码就是用四位二进制数来表示 0～9 这十个十进制符号，简称为 BCD 码。由于四位二进制数从 0000～1111 共有十六种组合，而十进制只有十个数码符号，因此有很多种 BCD 码。如 8421 码、2421 码、5211 码、余 3 码等。常用的是 8421BCD 码。

① 8421 码。8421 码是用四位二进制数的前十种组合来表示 0～9 这十个十进制数。这种代码每一位的权都是固定不变的，属于恒权代码。它和四位二进制数一样，从高位到低位各位的权分别是 8、4、2、1，故称为 8421 码。其特点是每个代码的各位数值之和就是它所表示的十进制数。所以，它便于记忆，应用也比较普遍。

【例 1-24】 若 $X=(01001010)_2$，$Y=(00110111)_2$，求 $X+Y$ 的 BCD 码?

解：$X+Y=(01001010)_2+(00110111)_2$

$=(10000001)_2=(129)_{10}$

$=(000100101001)_{BCD}$

② 2421 码和 5211 码。它们也属于恒权代码，从高位到低位各位的权分别是 2、4、2、1 和 5、2、1、1，故而得名。其中 2421 码又分为(A)和(B)两种代码，它们的编码状态不完全相同。在 2421(B)码中，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 互为反码，即两码对应位的值相反。

③ 余 3 码。这种代码所组成的四位二进制数，正好比它代表的十进制数多 3，故称为余 3 码。两个余 3 码相加时，其和要比对应表示的十进制数之和多 6。因而两个十进制数之和等于 10 时，两个对应余 3 码之和相当于四位二进制的 16，刚好产生进位信号，不必进行修正。另外，余 3 码的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 也互为反码。余 3 码不能由各位二进制数的权来决定其代表的十进制数，故属于无权码。各种 BCD 码的比较见表 1-1。

表 1-1 几种常用的 BCD 码

代码种类 / 十进制数	8421 码	2421(A)码	2421(B)码	5211 码	余 3 码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 0 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	0 1 1 1
5	0 1 0 1	0 1 0 1	1 0 1 1	1 0 0 0	1 0 0 0
6	0 1 1 0	0 1 1 0	1 1 0 0	1 0 0 1	1 0 0 1
7	0 1 1 1	0 1 1 1	1 1 0 1	1 1 0 0	1 0 1 0
8	1 0 0 0	1 1 1 0	1 1 1 0	1 1 0 1	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 0 0
权	8 4 2 1	2 4 2 1	2 4 2 1	5 2 1 1	

（3）逻辑数据（逻辑代码）

逻辑数据用于表示某件事情的“真”或“假”，“成立”或“不成立”，只能参加逻辑运算。基本逻辑运算包括“与”、“或”、“非”三种运算。参加逻辑运算的数据是按位进行的，位与位之间没有进位和借位的关系。在计算机中逻辑数据也是由二进制数所组成的，但每位数没有权。通常用“1”表示“逻辑真”，“0”表示“逻辑假”。

（4）美国标准信息交换码（ASCII 码）

ASCII 码，用一个字节(8 位二进制数)来表示一个特定的字符，其中低 7 位为字符的 ASCII 码值，最高位一般用作校验位。即实际上采用 7 位二进制数，可表示 $2^7=128$ 个符号。这 128 个符号共分为两类：一类是图形字符，共 96 个；另一类是控制字符，共 32 个。96 个图形字符包括十进制数码符号 10 个、大小写英文字母 52 个和其他字符 34 个。这类字符有特定的形状，可以显示在显示器上和打印在打印纸上，其编码可以存储、传送和处理。32 个控制符包括回车符、换行符、退格符、控制符和信息分隔符等。这类字符没有特定的形状，其编码虽然可以存储、传送和起某种控制作用，但字符本身不能在显示器上显示和打印机上打印。人们可通过键盘上的字母、符号和数值向计算机发送数据和指令，每一个键可用一个二进制代

码来表示，常用的就是 ASCII 码。

在 ASCII 码表中，左边和上边为相应字符的 ASCII 码，上边为高 3 位，左边为低 4 位。例如：数字 0～9 的 ASCII 码为 0110000B～0111001B(30H～39H)，大写英文字母 A～Z 的 ASCII 码为 1000001B～1011010B(41H～5AH)，小写英文字母 a～z 的 ASCII 码为 1100001B～1111010B(61H～7AH)。控制符注释如下：NUL，空；SOH，标题开始；STX，正文结束；ETX，文本结束；EOT，传输结束；ENQ，询问；ACK，承认；BEL，响铃(Bell)；BS，退一格；HT，横向列表；LF，换行(Line Feed)；VT，垂直制表；FF，走纸控制；CR，回车(Carriage Return)；SO，移位输出；SI，移位输入；SP，空格(Space)；DLE，数据链换码；DC1，设备控制 1；DC2，设备控制 2；DC3，设备控制 3；DC4，设备控制 4；NAK，否定；SYN，空转同步；ETB，信息组传送结束；CAN，作废；EM，纸尽；SUB，减；ESC，换码；FS，文字分隔符；GS，组分隔符；RS，记录分隔符；US，单元分隔符；DEL，删除。

1.2　单片机基础

1.2.1　计算机的经典组成

（1）计算机的经典组成

计算机的经典结构如图 1-1 所示。这种结构是由计算机的开拓者——数学家约翰·冯·诺依曼最先提出的，所以就称为冯·诺依曼计算机体系结构，也叫普林斯顿结构。由该图可知，计算机的经典结构由运算器、控制器、存储器、输入设备和输出设备五个部分组成。至今为止，计算机的发展已经经历了四代，即电子管时代、晶体管时代、集成电路时代、大规模及超大规模集成电路时代，微型计算机技术也得到了充分的发展，但仍未冲出冯·诺依曼体系。当前，市场上常见的大多数型号的单片机也还遵循着冯·诺依曼体系。下面简要分析计算机各部分的作用及工作原理。

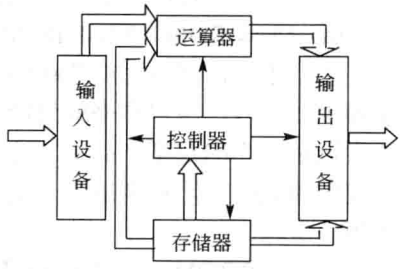


图 1-1　计算机的经典结构

如果要使计算机按照人们的需要解决某个具体问题，并不是把这个问题直接让计算机去解决，而是要用计算机可以“理解”的语言，编写出一列解决这个问题的步骤(即程序)并输入到计算机中，命令它按照这些步骤顺序执行，从而使问题得以解决。编写解决这些问题的步骤，就是人们常说的编写程序，也叫程序设计或软件开发。计算机是严格按照程序对各种数据或者输入信息进行自动加工处理的，因此必须先把程序以及数据用“输入设备”送入计算机内部的“存储器”中存起来，处理完后还要把结果用“输出设备”输送出来。输入设备包括键盘、鼠标、扫描仪等。输出设备包括显示器、打印机、绘图仪等。“运算器”完成程序中规定的各种算术和逻辑运算操作。为了使计算机各部件有条不紊地工作，由“控制器”理解程序的意图，并指挥各部件协调完成规定的任务。通常，在微型计算机中，把控制器和运算器制作在一块集成电路内，并称之为中央处理器或中央处理单元（CPU）。CPU 是计算机中最重要的部件，被喻为计算机的大脑和心脏。其具体功能有：程序控制、操作控制、时间控制、数据加工、控制程序和数据的输入与结果的输出、对异常情况和请求的处理等。目前微机常用的 CPU 芯片主要有 Intel 公司的 C800、C850、PIII866、PIII1G、PIV1.3G、PIV1.4G、PIV1.7G、PIV1.8G 等。

（2）计算机的存储器结构

① 半导体存储器。计算机的存储器通常用半导体存储器。半导体存储器是计算机的记