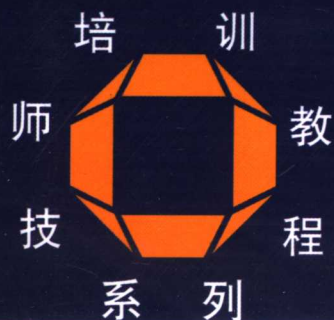
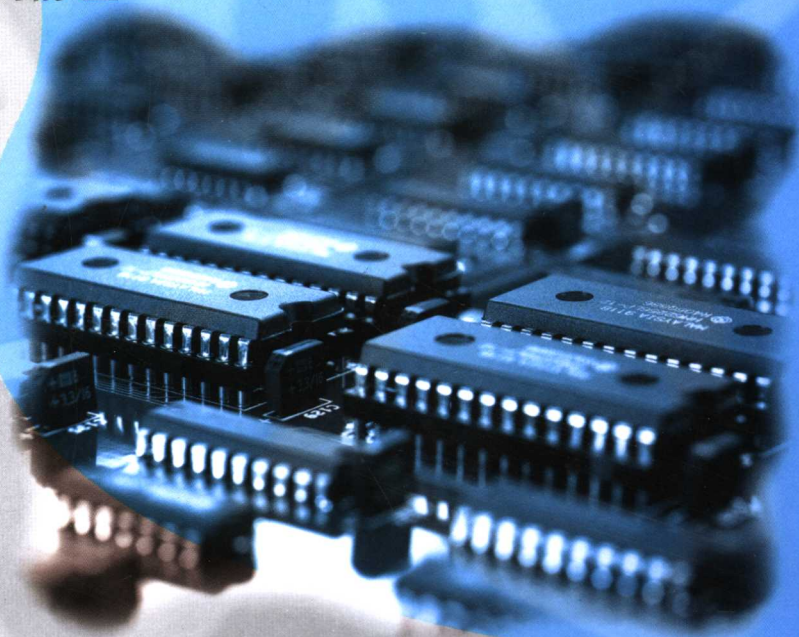




DANPIAN WEIJI
KONGZHI YINGYONG
JISHU

单片机控制 应用技术

主 编 张大明
副主编 彭旭昀 尚静基



技师培训教程系列

单片机控制应用技术

主 编 张大明

副主编 彭旭昀 尚静基



机械工业出版社

本书从实用角度出发,前半部分介绍了 MCS-51 系列单片微机的原理、结构、指令系统、程序设计方法;内部定时/计数、中断、串行通信功能及使用方法;基本系统扩展、输入/输出通道接口技术以及应用系统设计方法。后半部分介绍了单片微机 C 语言程序设计基础和大量 C51 编程实例。此外,书中还列举了大量的例题和实际操作课题,有些实训例题除提供常见的编程方法和接口电路外,还给出简单实用的电路和巧妙的程序及其调试方法。本书着重培养读者的创造性能力、软件编程能力和硬件电路设计能力,提高单片微机应用的开发能力,特别适合初学者学习。

本书叙述遵循教学规律,循序渐进,深入浅出,易于理解,便于自学,可作为大专院校以及高职、电大、函大、职大和培训班的教材及工程技术人员的自学和参考书。

图书在版编目(CIP)数据

单片微机控制应用技术 / 张大明主编. —北京:

机械工业出版社, 2006.2

(技师培训教程系列)

ISBN 7-111-18567-6

I. 单... II. 张 ... III. 单片微型计算机—控制系
统一技术培训—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2006) 第 013200 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 牛新国

责任印制: 李 妍

北京中兴印刷有限公司印刷

2006 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16·25.5 印张·629 千字

0001—4000 册

定价: 40.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68326294

编辑热线: (010) 88379768

[Http://www.machineinfo.gov.cn/book/](http://www.machineinfo.gov.cn/book/)

封面无防伪标均为盗版

技师培训教程系列

编委会名单

主 任：黄景容

副主任：王 德 张红岩 彭旭昀

顾 问：乐嘉祥 侯勇志 王晓沛

委 员：涂爱云 岳庆来 张大明 乔建伟 吴启红 刘振鹏

胡 洪 唐先军 王跃军 胡冬兵 谭 斌 黄太平

鄂永雄 廖俊龙 陈 芸 蔡 勇 官风华 赵宪龙

王继桦 李 明 梁昌龙 丘 洪

序

随着知识经济时代的到来,科学技术突飞猛进,特别是随着经济的全球化发展和我国加入 WTO,我国将逐步成为世界的“制造中心”。同时,我国经济的不断发展和产业结构的转型升级,也使得我国符合企业需求的一线工程技术人员严重匮乏,特别是技能人才的市场供给严重不足,而且正在成为影响经济进一步发展的瓶颈。

随着国家劳动和社会保障部国家高技能人才培训工程和《三年五十万新技师培养计划》的实施,高技能人才的培养有了较大的发展并面临新的契机。本系列教材就是在这种形势下组织编写的,以适应经济、社会发展对技能人才教育的新要求,满足高素质、复合型人才培养的需要。

深圳技师学院实行模块化和分段培养教学模式,已经成为深圳市高技能人才培养的主要基地和主要力量,年培养技能人才 10000 人次。在培训教学实践中;我们感到真正实用、好用的培训教材为数不多,尤其是有特色的、贴近生产实际的教材更显匮乏,教材的匮乏及教学内容的滞后已经成为培训教学的瓶颈问题。在技能人才培养的教学过程中,教材处于基础地位,是课程体系设计的核心,因此我们尝试组织专业技术人员自编部分教材,在教材编写中,结合市场需求,我们把相关知识点的学习与专业技能的训练有机地结合起来,摒弃以往“就知识讲知识”的传统做法,在培训教学中“边使用,边调整”,取得了较好的教学效果。

本系列教材的编写思想从以下四个方面出发:

一是根据国家劳动和社会保障部相关技能鉴定规范,并参考了深圳市电工等工种相关职业标准,包括深圳市电工、电梯、制冷专业技师(模块化)职业标准,力求教学内容能覆盖相应技能鉴定的要求。

二是结合当前企业的生产实际,力求教学内容能反映本工种新技术、新工艺的应用,具有一定的广度和深度,因此部分参编人员来自企业一线,能够了解企业生产需要。

三是培养目标明确,强化应用。教学中注重培养学员的职业能力,坚持技能人才的培养方向,内容安排上符合认知规律,由浅及深,由易到难。

四是内容编排上,努力做到理论与实践紧密结合,侧重实践操作,理论知识以够用为度,技能实操以培养掌握复杂操作和新技术操作的技能,并以增强分析、判断、排除各种实际故障为重点。

本系列教材的编者来自深圳技师学院从事培训教学的一线教师和企业的一部分专家,书中内容基本反映出深圳市技能培训教学和社会化考核的方向,相信本书会受到中、高职类院校广大师生和广大青年读者的欢迎。

由于时间的限制,也囿于我们的知识和水平,书中疏漏之处在所难免,恳请广大读者和业内人士批评指正!

编委会主任 黄景容

前 言

单片微型计算机 (Single Chip Microcomputer), 简称单片机 (SCM), 也称为微处理器 (MPU) 或微控制器 (MCU), 是 20 世纪 70 年代中期发展起来的一种新型微机。由于单片机在国防、工业、农业、交通、医疗、家电、通信、楼宇智能化、机器人等各高精尖技术领域的广泛应用, 促进了单片机技术的迅猛发展。近年来适应各种要求的各种类型、系列的单片机如雨后春笋般出现。速度快、体积小、功能全的单片机新品种不断涌现。MCS-51 系列单片机是美国 Intel 公司在 1980 年继 MCS-48 系列 8 位单片机之后推出的 8 位单片机。具有性价比高、稳定可靠、通用性强、体积小、价格低等优点。单片机的许多新品种都是继承了 51 系列单片机的技术内核开发出来的, 与 51 系列单片机相互兼容, 可以互换。目前, 51 系列单片机是国内单片机应用及教学的主流产品, 本书介绍单片机原理时以 8051 (80C51) 系列或目前流行的 AT89C51 (或 AT89S51) 为例进行讲述。

本书前半部分 (第 1~9 章) 以汇编语言为编程语言, 讲述 51 系列单片机的原理、结构、指令系统、程序设计方法; 内部定时/计数、中断、串行通信功能及使用方法; 基本系统扩展、输入/输出通道接口技术、应用系统开发设计的过程及抗干扰技术简介。考虑到单片机原理及应用是一门实践性很强的课程, 对于初学者不宜一次讲得很深很透。因此本书改变了以往说明书式的教材写法, 对基本原理与结构的讲述以够用为原则, 点到为止; 而对应用方面则以较大篇幅给出较多的例题, 并且有的例题给出多种解法, 以培养读者的创造性能力、软件编程能力和硬件电路设计能力, 提高单片机应用的开发能力。后半部分 (第 10 章和第 11 章) 详细介绍了单片机 C 语言程序设计基础并给出大量 C51 编程实例, 特别是在单片机的输入/输出、串行通信、LED 和 LCD 点阵图形显示方面, 给出各种应用例题。对 LED 点阵图形显示给出各种编程思路和方法, 以迅速提高读者的 C 语言编程能力, 从而尽快进入开发设计者的行列中。后半部分适用于有 51 系列单片机汇编语言基础的单片机高级培训班学习。

本书是作者多年教学实践与科研开发的经验积累, 同时为了使本书的内容更加丰富和完整, 书中也引用了部分参考书籍的内容, 主要来源见参考文献。在此谨对有关作者表示感谢。

本书由张大明担任主编, 彭旭昀、尚静基担任副主编。杨运峰编写第 1 章和第 2 章; 彭旭昀编写第 3 章; 胡洪编写第 4 章; 王跃军编写第 6 章; 吴启红编写第 7 章; 关君编写第 8 章; 尚静基编写第 10 章; 郭闰丙编写第 11 章第 11.1~11.4 节; 李翎编写第 11 章第 11.5 节; 靳小红编写第 11 章第 11.6.1 节; 张大明编写第 11 章第 11.6.2 节; 赵忻编写第 9 章和第 11 章第 11.7、11.8 节; 刘振鹏编写第 5 章、部分插图和附录。全书由张大明统稿。

由于本书中例题数量较大, 例的编号依据是, 对一些基本概念和知识的例子, 若为较简单的练习, 就只是写“例”; 若为较复杂的练习, 或者是重点例题及编程题, 就加上编号。

书中带☆号的题为深圳市电工技师职业技能鉴定考试题。

由于编者水平有限, 书中的错误与不妥之处在所难免, 恳请广大读者批评指正。

编 者

2006 年 1 月

目 录

序

前言

第1章 单片微型计算机基础.....	1	3.1.1 指令格式.....	27
1.1 微型计算机的数制与码制.....	1	3.1.2 指令分类及符号说明.....	27
1.1.1 进位计数制.....	1	3.1.3 寻址方式.....	29
1.1.2 各种进制间的转换.....	2	3.2 指令及功能.....	31
1.1.3 代码.....	5	3.2.1 数据传送类.....	31
1.1.4 带符号数的表示及转换(8位).....	6	3.2.2 算术运算类.....	37
1.1.5 二进制的算术运算(8位).....	7	3.2.3 逻辑运算类.....	39
1.1.6 二进制的逻辑运算.....	8	3.2.4 控制转移类.....	42
1.2 单片微型计算机基本原理.....	8	3.2.5 位操作指令.....	44
1.2.1 微型计算机或单片微型计算机控 制系统的结构.....	8	3.2.6 伪指令.....	46
1.2.2 单片微型计算机的工作原理.....	10	3.3 程序设计方法.....	50
1.2.3 单片微型计算机控制系统设计运 行的过程.....	12	3.3.1 顺序结构程序设计.....	50
第2章 单片机原理与结构.....	13	3.3.2 分支结构程序设计.....	52
2.1 MCS-51 系列单片机结构.....	13	3.3.3 循环结构程序设计.....	54
2.1.1 MCS-51 系列单片机内部结构.....	14	3.3.4 查表程序设计.....	57
2.1.2 MCS-51 系列单片机引脚及功能.....	14	第4章 单片机内部功能及应用.....	61
2.2 MCS-51 系列单片机存储器结构 及其用途.....	16	4.1 定时/计数功能及应用.....	61
2.2.1 存储器分配.....	16	4.1.1 8051 定时/计数器结构和工作原理.....	61
2.2.2 程序存储器.....	16	4.1.2 定时器初始化.....	63
2.2.3 数据存储器.....	17	4.1.3 定时/计数器应用程序设计.....	65
2.3 输入/输出(I/O)接口.....	21	4.2 中断控制功能及应用.....	70
2.3.1 I/O 接口结构.....	21	4.2.1 中断概念.....	70
2.3.2 用途及用法.....	22	4.2.2 中断控制.....	72
2.4 时钟和复位电路.....	25	4.2.3 中断处理过程.....	73
2.4.1 振荡器和时钟电路.....	25	4.2.4 采用中断的程序结构.....	75
2.4.2 复位电路.....	25	4.2.5 中断应用程序设计举例.....	75
2.4.3 时钟与时序.....	26	4.3 串行通信功能及应用.....	78
第3章 单片机指令系统及程序设计.....	27	4.3.1 串行接口结构.....	79
3.1 指令系统简介.....	27	4.3.2 串行接口工作方式.....	81
		4.3.3 串行通信过程.....	81
		4.3.4 串行通信举例.....	82
		第5章 单片机基本系统扩展.....	85

5.1 存储器扩展.....	85	7.4 打印机接口及应用.....	140
5.1.1 程序存储器扩展.....	85	7.4.1 TPμP-AT 微型打印机.....	141
5.1.2 数据存储器扩展.....	90	7.4.2 微型打印机与单片机的接口电路.....	141
5.2 I/O 接口扩展.....	91	7.4.3 直接用 8051 的 P1 口驱动打印机.....	143
5.2.1 简单 I/O 接口扩展.....	91	第 8 章 单片机应用系统的开发设计	144
5.2.2 8255 可编程并行 I/O 接口扩展电路.....	92	8.1 单片机应用系统的开发设计过程.....	144
5.2.3 8155 可编程 I/O 扩展.....	95	8.1.1 单片机的开发系统.....	144
5.2.4 串行接口扩展 I/O 接口.....	98	8.1.2 单片机应用系统的设计过程.....	144
第 6 章 信号输入通道接口	101	8.2 实用程序设计.....	145
6.1 开关量输入通道.....	101	8.2.1 交通灯控制程序设计.....	145
6.1.1 有触点开关接口电路.....	101	8.2.2 生产线工件自动计数器程序设计.....	147
6.1.2 无触点开关接口电路.....	102	8.2.3 电梯轿厢数码管指层器程序设计.....	149
6.2 键盘输入接口.....	102	8.2.4 多台设备自动循环控制程序设计.....	151
6.2.1 独立式按键电路.....	102	8.2.5 步进电动机驱动程序设计.....	152
6.2.2 矩阵式(行列式)按键电路.....	105	8.2.6 LED 数码管动态显示程序设计.....	160
6.2.3 中断式键盘电路.....	105	8.2.7 注塑机变频微机控制系统程序设计.....	161
6.3 模拟量输入通道.....	107	8.2.8 单片机控制内燃机散热器叶片装 配机程序设计.....	165
6.3.1 模拟量采集.....	107	第 9 章 单片机控制系统抗干扰技术	171
6.3.2 ADC0809 A/D 转换器及接口电路 (用 74LS373 和门电路驱动).....	107	9.1 干扰的来源.....	171
6.3.3 直接用 8051 的 P1 口驱动 ADC0809.....	114	9.2 抑制干扰的措施.....	171
6.3.4 TLC0831 串行 A/D 转换器接口.....	116	9.2.1 硬件抗干扰与硬件“看门狗”.....	171
6.3.5 AD654 压频(V/F)转换器.....	118	9.2.2 软件抗干扰与软件“看门狗”.....	172
第 7 章 信号输出通道接口	122	第 10 章 单片机 C 语言程序设计基本语法	175
7.1 开关量输出通道.....	122	10.1 C 语言的特点和程序结构.....	175
7.1.1 大功率晶体管驱动电路.....	122	10.1.1 C 语言的特点.....	175
7.1.2 晶闸管及其接口电路.....	123	10.1.2 C 语言的程序结构.....	176
7.1.3 固态继电器及其接口电路.....	123	10.2 C 语言的标识符和关键字.....	178
7.2 模拟量输出通道.....	123	10.3 数据类型.....	181
7.2.1 DAC0832 D/A 转换器及接口电路.....	124	10.3.1 字符类型 char.....	181
7.2.2 直接用 8051 的 P1 口驱动 DAC0832.....	126	10.3.2 整型 int.....	182
7.2.3 MAX517/518 串行 D/A 转换器接口.....	127	10.3.3 长整型 long.....	182
7.3 显示接口.....	129	10.3.4 单精度浮点型 float 和双精度浮点 型 double.....	182
7.3.1 LED 数码管的静态显示接口.....	129	10.3.5 位变量 bit.....	182
7.3.2 LED 数码管的动态显示接口.....	132	10.3.6 特殊位变量 sbit.....	183
7.3.3 直接用 8051 的 P1 口驱动 LED 数 码管.....	134	10.3.7 特殊功能寄存器 sfr 和 sfr16.....	183
7.3.4 MAX7219 专用数码管显示电路.....	136	10.3.8 BYTE 字节型数据存储器.....	184
7.3.5 液晶显示器(LCD)及接口电路设计.....	139	10.4 常量.....	185

10.4.1 整型常量	185	10.11.4 函数的参数和函数的返回值	231
10.4.2 浮点型常量	185	10.11.5 数组、指针作为函数的参数	232
10.4.3 字符型常量	186	第 11 章 C51 编程实例	234
10.4.4 字符串型常量	186	11.1 8051 单片机输出控制的 C 语言编程	234
10.5 变量及存储类型	187	11.1.1 用 4 个并行 I/O 接口 (Pi 口) 直 接输出数据	234
10.6 运算符及表达式	188	11.1.2 向片外数据存储器或 I/O 接口芯 片输出数据	239
10.6.1 赋值运算符及其表达式	188	11.2 8051 单片机输入控制的 C 语言编程	243
10.6.2 算术运算符及其表达式	189	11.2.1 用 4 个并行 I/O 接口 (Pi 口) 直 接输入数据	243
10.6.3 关系运算符及其表达式	190	11.2.2 由片外数据存储器或 I/O 接口芯 片输入数据	244
10.6.4 逻辑运算符及其表达式	191	11.3 8051 单片机定时/计数器功能的 C 语言编程	246
10.6.5 位运算符及其表达式	192	11.4 8051 单片机中断功能的 C 语言编程	259
10.6.6 自增减运算符、复合运算符及其 表达式	194	11.5 8051 单片机串行通信的 C 语言编程	264
10.6.7 逗号运算符和条件运算符	195	11.5.1 用 8051 串行接口输入或输出数据	265
10.7 控制语句	195	11.5.2 用 8051 串行接口通信	267
10.7.1 C 语言程序的结构和语句	195	11.6 LED 点阵图形显示器接口电路的 C 语言编程	290
10.7.2 条件语句 if	195	11.6.1 LED 点阵汉字图形显示器接口 电路	290
10.7.3 开关语句 switch/case	198	11.6.2 LED 大屏幕点阵汉字图形显示 系统	358
10.7.4 循环语句	199	11.7 液晶点阵图形显示器接口电路的 C 语言编程	359
10.8 数组	209	11.7.1 LCM1602 液晶点阵字符显示器 接口电路	359
10.8.1 数组的定义、初始化和使用	209	11.7.2 液晶点阵汉字图形显示器接口 电路	365
10.8.2 字符数组	210	11.8 PC 与单片机的串行通信	380
10.8.3 数组的应用	211	11.8.1 硬件电路的组成	381
10.9 指针	211	11.8.2 软件程序的设计	381
10.9.1 指针的概念	211	附录	387
10.9.2 指针变量的定义和赋初值	213	附录 A ASCII 码表	387
10.9.3 指针变量的使用	214	附录 B 常用集成电路引脚图	388
10.9.4 数组的指针	216	附录 C MCS-51 指令表	393
10.9.5 数组指针变量的运算	216	参考文献	397
10.9.6 指针数组	222		
10.10 结构体	223		
10.10.1 结构体的定义	223		
10.10.2 结构体的使用	224		
10.10.3 结构体数组	225		
10.10.4 结构体指针	226		
10.11 函数	227		
10.11.1 函数的定义	228		
10.11.2 函数的调用	230		
10.11.3 对被调用函数的说明	230		

第1章 单片微型计算机基础

本章讲述微型计算机的数制与码制及单片微型计算机的基本原理，这些基础内容对今后学习单片微型计算机的指令、程序和应用都是非常重要的。

1.1 微型计算机的数制与码制

1.1.1 进位计数制

让大家先来学习进位计数制。要想用单片机控制霓虹灯、交通灯、航标灯、多台电动机循环控制和让步进电动机依使用者的意愿运行，想用数码管随意点亮数字，甚至想要点阵式的LED（发光二极管）点亮任意的图形，就需要熟练地写出十进制数、二进制数和十六进制数并进行它们之间的转换。

1. 十进制

十进制（D）数是大家很熟悉的，有时为了与二进制数和十六进制数相区别，十进制数可以用一个括号加下角标10注明，或在后边用大写的D表示。根据数学上计数的理论， N 位计数制的数，可以按 N （又称基数）的幂指数展开求和的方法求出其值，如十进制数可按10的幂指数展开求和的方法表示。例如， $(89.75)_{10}$ （或 $89.75D$ ）可写为

$$89.75 = 8 \times 10^1 + 9 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

其中某一位数乘10的几次方，要看这一位后面的整数部分有几位。例如，上面的数字8后面，整数部分有1位，这个8就相当于8乘10的1次方。

2. 二进制

先来谈谈为什么要引进二进制（B），由于计算机和单片机中的中央处理器（CPU）、存储器和其他硬件都是用集成电路做成的，这些集成电路内部都是由一个个晶体三极管（或场效应晶体管）组成的数字电路构成的。数字电路在存储数字或进行计算时，是利用晶体三极管（或场效应晶体管）工作在开关状态下，集电极（或漏极）不是处于低电平就是处于高电平来表示数码0和1的。并排的8个晶体管，其集电极电平的高低状态可以用8位二进制数表示。人们写在纸上的一个8位二进制数“11010101”，存在计算机和单片机内部的硬件电路CPU或内存里，就是8个晶体管的集电极电平的“高高低低高低高”。可以说，计算机和单片机内部的硬件电路“只认识”二进制数，“不认识”十进制数、十六进制数及其他的任何码制。其他的任何计数制和码制，都要转换为二进制数，才能在计算机和单片机内部运行。实质上，人们写在纸上的其他任何计数制和码制，都要转换为二进制数，才能输入计算机和单片机中（内存里）。人们通过计算机键盘敲数字键或字符键时，一方面，输入的数字或字符显示在显示器的屏幕上；同时，另一方面，键盘里的单片机已经自动把代表这些数字或字符的ASCII码以二进制数的形式输入到计算机主机内部（计算机的内存）了。

二进制数只有两个数码，即0、1，加法逢二进一位。二进制数后可加一大写的B表示。例如

$$\begin{aligned}
 (1011.011)_2 &= 1011.011B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 + 0 \times 1/2 + 1 \times 1/4 + 1 \times 1/8 \\
 &= 11.375
 \end{aligned}$$

2 的幂指数与十进制数和二进制数的关系如下:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
↓	↓	↓	↓	↓	↓	↓	↓	↓
256	128	64	32	16	8	4	2	1
↓	↓	↓	↓	↓	↓	↓	↓	↓
100000000	10000000	1000000	100000	10000	1000	100	10	1

3. 十六进制

用二进制表示一个数要写很长, 难于记忆, 于是人们引进了十六进制 (H) 数。十六进制数有 16 个数码, 即 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F, 基数是 16, 加法逢十六进一位。十六进制数后可加一个大写的 H 表示。例如

$6EH = (6E)_{16} = 6 \times 16^1 + 14 \times 16^0 = (110)_{10} = 110D = 110$ (这是十进制的 110, 读作十进制数时, “1” 的读音为 “衣”, 读作二进制数或十六进制数时, “1” 要读成 “腰”, 否则与 “E” 就区别不开了)。

4. 十进制、二进制与十六进制数对照表

十进制、二进制与十六进制数对照表如下:

十进制	二进制	十六进制
0	0000	0H
1	0001	1H
2	0010	2H
3	0011	3H
4	0100	4H
5	0101	5H
6	0110	6H
7	0111	7H
8	1000	8H
9	1001	9H
10	1010	AH
11	1011	BH
12	1100	CH
13	1101	DH
14	1110	EH
15	1111	FH
16	10000	10H

1.1.2 各种进制间的转换

1. 二进制数 ↔ 十六进制数

(1) 二进制数 → 十六进制数。整数部分将二进制数从最低位 (小数点左边第 1 位) 开始,

向左数，每 4 位二进制数转换为一位十六进制数。

例 $101101 = \underline{0010} \underline{1101} = 2DH$ 。

☆例 $110110110 = \underline{0001} \underline{1011} \underline{0110} = 1B6H$ (最高位不够 4 位的前面补零凑够 4 位)。

例 $(01110001101.1100001) = 38D.C2H$ (小数部分向右数，每 4 位一组转换为一位十六进制数，最低位不够 4 位的后面补零凑够 4 位)。

(2) 十六进制数 $\rightarrow$ 二进制数。将每 1 位十六进制数转换为 4 位二进制数。

例 $35H = 00110101$, $96H = 10010110$, $C7H = 11000111$ 。

由于二进制数与十六进制数之间的转换方法很简单，所以人们在书写时常用，也总是用十六进制数去表示二进制数。一个二进制数与它的十六进制数是一一对应的，下面会“指鹿为马”地指着一个十六进制数说它是一个二进制数。不过，这些十六进制数，进入计算机内后，正如前述，统统都会变成二进制数（实际是变成晶体三极管集电极电平的高低不同状态）。

2. 二进制数、十六进制数 $\leftrightarrow$ 十进制数

(1) 二进制数、十六进制数 $\rightarrow$ 十进制数见 1.1.1 节内容。

例 $6EH = 110$

$$80H = 128 = 2^7$$

$$100H = 256 = 2^8$$

$$400H = 1024 = 2^{10}$$

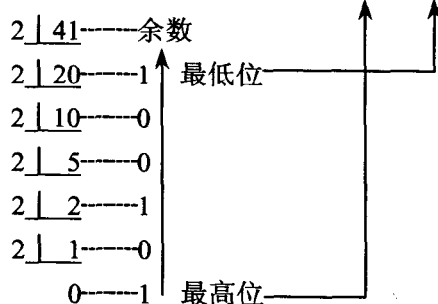
$$1000H = 4096 = 2^{12}$$

$$10000H = 65536 = 2^{16}$$

这些数据很有用，最好能记住。

(2) 十进制数 $\rightarrow$ 二进制数。

方法一：整数部分用除 2 取余法， $41 = 101001$



方法二： $41 = 32 + 8 + 1$

$= 100000 \leftarrow 32$ 的二进制数

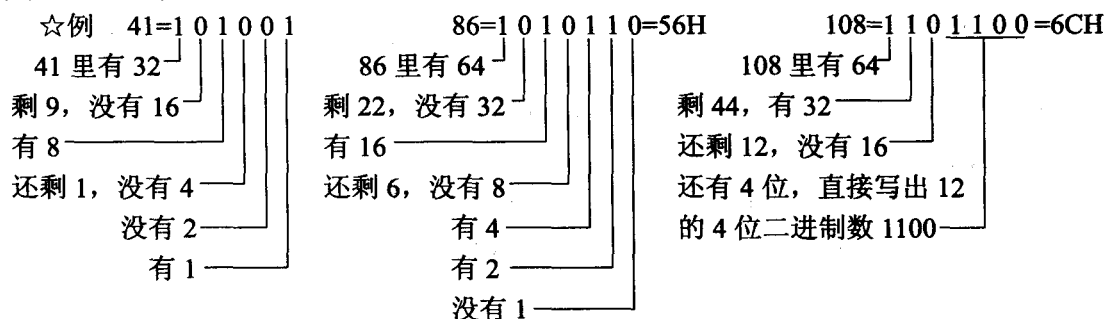
$+ 1000 \leftarrow 8$ 的二进制数

$+ 1 \leftarrow 1$ 的二进制数

$= 101001 \leftarrow 41$ 的二进制数

方法三：根据方法二，按某十进制数包含哪些 2 的幂指数，心算可一步将该十进制数直

接写成二进制数，从高位到低位，含有 2 的幂指数的位写 1，没有的位写 0。



126=1111110=7EH, 127=01111111=7FH, 128=10000000H=80H, 256=100000000=100H,
 89.75=1011001.11=59.CH (0.75 里有 $2^{-1}=1/2=0.5$, 还有 $2^{-2}=1/4=0.25$)。

(3) 十进制数→十六进制数。

方法一：按十进制→二进制→十六进制进行。

例 126=1111110=7EH

89=1011001=59H

98=1100010=62H

$2^8=256=100H$

$2^9=512=200H$

$2^{10}=1024=400H$

$2^{11}=2048=800H$

$2^{12}=4096=1000H$

$2^{13}=8192=2000H$

$2^{14}=16384=4000H$

$2^{15}=32768=8000H$

$2^{16}=65536=10000H$

例 15536=11110010110000=3CB0H

-8192

=7344

-4096

=3248

-2048

=1200

-1024

=176

-128

=48

-32

16

方法二：除 16 取余法。

15536=3CB0H

<u>971</u>	<u>60</u>	<u>3</u>
16/15536	16/971	16/60
<u>144</u>	<u>96</u>	<u>48</u>
113	余 11=B	余 12=C
<u>112</u>		
16		
<u>16</u>		
余 0		

方法三：用科学计算器（计算机里的计算器也有科学计算功能）。点击【开始】→【程序】→【附件】→【计算器】，选择【查看】→【科学型】。选中【十进制】，按数字按钮“1”、“5”、“5”、“3”、“6”，再选中【十六进制】，即出现“3CB0”。

1.1.3 代码

计算机内部的硬件电路“只认识”二进制数，而人们日常生活习惯于十进制数，进行书写习惯于用英文字符“A、B、C…”（或汉字）。另外，大量不懂计算机原理的人又必须使用计算机键盘输入文字。这就出现了一个问题，即如何把十进制数、键盘上的“A、B、C…”这些英文字符和标点符号等用二进制数表达出来，用一些特别规定好的二进制数代表这些十进制数和各种字符，让计算机懂得输入的是什，于是就出现了代码。所谓代码，就是按预先规定的方法，用二进制数码（实际写出来为了简单还是用十六进制数）表示十进制数或各种字符的一种编码。这些二进制数代码虽然是用二进制数形式写出来的，但已不是原来意义上的二进制数了，而是一些代替十进制数或字符的二进制编码，就好像一个班上的学生花名册，用排列序号代表某某学生一样。常用的代码有二—十进制编码和 ASCII 编码。

1. 二—十进制编码

二—十进制编码（BCD 码）就是用二进制形式表示十进制数的编码。常用的编码方法有 8421BCD 码、5421BCD 码、余 3 码及格雷码等。

十进制数→8421BCD 码时，是把每一位十进制数用 4 位二进制数表示。此时的“二进制数”要加括号并用下角标“BCD”表示它是一个 BCD 码，已不是原来意义上的“二进制数”了。

例 $86 = (10000110)_{\text{BCD}}$ ，如果要把这个代表十进制数 86 的 BCD 码输入到计算机里，进入计算机内存的只是“10000110”，而不是加了标注的“(10000110)_{BCD}”。人们把“10000110”分成 4 位一组再按十进制数读出来，就读成“八十六”。为了简单起见，可以把这个代表十进制数 86 的 BCD 码 $(10000110)_{\text{BCD}}$ 记成十六进制形式“86H”，此时的 86H 已不是原来意义上的“十六进制数”了，而是十进制数的 86 的 BCD 码，就代表了十进制数 86。

例 $35 = (00110101)_{\text{BCD}} = 35\text{H}$ $98 = (10011000)_{\text{BCD}} = 98\text{H}$ $89 = (10001001)_{\text{BCD}} = 89\text{H}$

BCD 码分压缩 BCD 码和非压缩 BCD 码，代表十进制数 35 的 BCD 码 35H 是压缩 BCD 码，单独表示其十位数 3 的 BCD 码 03H 和表示其个位数 5 的 BCD 码 05H 是非压缩 BCD 码。

2. ASCII 码

属于字符编码（字符的二进制编码，它是美国标准信息交换码）。例如，阿拉伯数字 0 的 ASCII 码是“0”=30H，还有“1”=31H，…，“9”=39H。人们敲键盘上的数字“0”键，输入到计算机内存中的是 30H。另外，英文大写字母的 ASCII 码为“A”=41H，“B”=42H，…，

小写字母的 ASCII 码为“a”=61H, “b”=62H, …。空格命令、回车换行命令的 ASCII 码分别为 20H、0DH。请参看本书附录 A。

1.1.4 带符号数的表示及转换 (8 位)

计算机中要表示的数有不带符号数 (没有正负之分, 只有零和正数) 和带符号数 (有正有负有零)。计算机中的数叫机器数, 机器数相当于十进制数的值叫真值。下面的叙述以 MCS-51 系列单片机为例, MCS-51 系列单片机中进行传送、运算、处理的数, 一个单元 (字节) 是 8 位二进制数。习惯上, 最右边的最低位标为 D0 (称为第 0 位, 不是第 1 位), 最高位标为 D7 (称为第 7 位)。例如, 二进制数 11010011, 各位标记为

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	1	0	0	1	1

1. 不带符号数的表示

8 位全用来表示大小, 可以表示的十进制数的范围是 0~255。

例 01001101=4DH=77 11111111=FFH=255 01111111=7FH=127 10000000=80H=128

2. 带符号数的表示

如果用 8 位二进制数表示有正有负的数, 就需要用 8 位中的 1 位表示数的正负。规定最高位 D7 是符号位, D7=0 表示 “+”, D7=1 表示 “-”。其余各位 (D6~D0) 为数值位。对于数值位, 又有如下的原码、反码、补码之分。

(1) 原码。其余各位 (D6~D0) 表示该数的绝对值。原码所能表示十进制数的范围是 -127~+127。

☆例 (+86)_原=01010110=56H (-86)_原=11010110=D6H
(+127)_原=01111111=7FH (-127)_原=11111111=FFH

(2) 反码。正数的反码与原码相同; 负数的反码: 符号位 (D7) 仍为 1, 其余各位是将该负数的原码对应的各位取反。

☆例 (+86)_反=01010110=56H (-86)_反=10101001=A9H
(+127)_反=01111111=7FH (-127)_反=10000000=80H

(3) 补码。正数的补码与原码相同; 负数的补码是, 符号位 (D7) 仍为 1, 其余各位是将该负数的反码末位加 1, 即 $(X)_{补} = (X)_{反} + 1$ 。

☆例 (+86)_补=01010110=56H (-86)_补=10101010=AAH
(+127)_补=01111111=7FH (-127)_补=10000001=81H

8 位二进制数补码可以表示的十进制数的范围是 +127~-128 (01111111~10000000 或 7FH~80H)

从上面可以看出, 正数的原码、反码、补码都是相同的, 负数的原码、反码、补码各不相同。由负数的原码求补码除了先由原码求反码, 再由反码求补码的方法之外, 还可以直接由原码求补码。利用公式 $(-X)_{补} = 100H - X$ 。注意, 式中的 -X 是负数, X 是正数。由于 8 位二进制数补码所能表示的负数范围是 -1~-128, 此公式只适用于计算 -1~-128 的负数的补码。如果要计算 16 位二进制数负数的补码, 应当用公式 $(-X)_{补} = 10000H - X$ 。以后在第 4 章讲定时/计数器时也要用到这个公式。

例 (-86)_补=100H-86=100H-56H=AAH (-127)_补=100H-127=100H-7FH=81H
(+1)_补=00000001=01H (-1)_补=100H-01H=FFH=11111111

$$\begin{array}{ll}
 (+2)_{\text{补}} = 00000010 = 02\text{H} & (-2)_{\text{补}} = 100\text{H} - 02\text{H} = \text{FEH} = 11111110 \\
 (+3)_{\text{补}} = 00000011 = 03\text{H} & (-3)_{\text{补}} = 100\text{H} - 03\text{H} = \text{FDH} = 11111101 \\
 \vdots & \vdots \\
 (+127)_{\text{补}} = 01111111 = 7\text{FH} & (-127)_{\text{补}} = 100\text{H} - 7\text{FH} = 81\text{H} = 10000001 \\
 & (-128)_{\text{补}} = 100\text{H} - 80\text{H} = 80\text{H} = 10000000
 \end{array}$$

可以看出，负数的绝对值越小，它的补码看起来越大。

3. 转换（只适用于负数）

(1) 反→原。反反得原，即 $((X)_{\text{反}})_{\text{反}} = (X)_{\text{原}}$ 。

(2) 补→原。补补得原，即 $((X)_{\text{补}})_{\text{补}} = (X)_{\text{原}}$ 。

另外，还有一个规律，即 $(X-Y)_{\text{补}} = (X)_{\text{补}} + (-Y)_{\text{补}}$ ，由于计算机中的硬件电路只有加法器（全加电路），没有减法电路，这个规律可以把减法化为加法来计算，所以计算机中的数，如果不特别声明，一律是补码。

例 $((-86)_{\text{补}})_{\text{补}} = (10101010)_{\text{补}} = 11010110 = (-86)_{\text{原}}$

例 $X=34, Y=68$ ，则 $X-Y=-34$ 在计算机中的结果为

$$(-34)_{\text{补}} = 100\text{H} - 22\text{H} = \text{DEH} = 11011110$$

实际在计算机中

$$(X-Y)_{\text{补}} = (34-68)_{\text{补}} = (34)_{\text{补}} + (-68)_{\text{补}} = 22\text{H} + \text{BCH} = \text{DEH} = 11011110$$

也可直接相减

$$34-68=22\text{H}-44\text{H}=\text{DEH}=11011110$$

1.1.5 二进制的算术运算（8位）

1. 加法规律

$0+0=0, 0+1=1, 1+0=1, 1+1=0$ 并进一位。

2. 减法规律

$0-0=0, 1-1=0, 1-0=1, 0-1=1$ 并借一位。

3. 乘法规律

$0 \times 0=0, 0 \times 1=0, 1 \times 0=0, 1 \times 1=1$ 。

加、减、乘、除可先化成十六进制运算，之后再化成二进制。注意，十六进制加法规律是满 16 进一位；减法规律是不够减时向上借一位到本位就是 16。例如， $0-1=\text{F}$ ，并借一位。

例 $01110101+10101110=100100011$

$$=75\text{H}+\text{AEH}=123\text{H}$$

$$01110101 \quad 75\text{H}$$

$$+10101110 \quad +\text{AEH}$$

$$100100011 \quad 123\text{H}$$

例 $10101100-01110101=00110111$

$$=\text{ACH}-75\text{H}=37\text{H}$$

$$10101100 \quad \text{ACH}$$

$$-01110101 \quad -75\text{H}$$

$$00110111 \quad 37\text{H}$$

例 $12\text{H} \times 5 = 5\text{AH}$

1.1.6 二进制的逻辑运算

逻辑“与”、“或”、“非”、“异或”，按位运算，不同的位不相干。

1. 逻辑“与”

$$0 \wedge 0 = 0, 0 \wedge 1 = 0, 1 \wedge 0 = 0, 1 \wedge 1 = 1.$$

可以看出，某数（这里指的是二进制的一位数，不是0就是1）跟0相“与”，结果变为0（又称为“置0”、“清0”、“复位”）；某数跟1相“与”，结果保留不变。

2. 逻辑“或”

$$0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1.$$

可以看出，某数跟0相“或”，结果保留不变；某数跟1相“或”，结果变为1（又称为“置1”、“置位”）。

3. 逻辑“非”

即取反， $0 = \bar{1}$ ， $1 = \bar{0}$ 。

4. 逻辑“异或”

$$0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0.$$

可以看出，某数跟0相“异或”，结果保留不变；某数跟1相“异或”，结果被取反。

1.2 单片微型计算机基本原理

1.2.1 微型计算机或单片微型计算机控制系统的结构

微型计算机或由单片机组成的单片微型计算机控制系统主要由中央处理器（CPU）（又称微处理器 MPU）、存储器和 I/O 接口电路三大部分组成，各部分通过三态门与地址总线（AB）、数据总线（DB）和控制总线（CB）连接起来并相互传递信息（即传送数据，又叫通信、访问）。其结构如图 1-1 所示，这种结构叫三总线结构。存储器的各单元和各 I/O 接口芯片都有各自的地址，地址总线是 CPU 向存储器和 I/O 接口芯片读写数据时发布的地址信号，以确定到底是要向哪一个存储器的单元或哪一个 I/O 接口芯片进行读写。数据总线用来传送数据，CPU 向存储器各单元和各 I/O 接口芯片读写数据时，数据就被送到数据总线上，如果是 CPU 在从某存储器单元或 I/O 接口芯片（或 I/O 端口）读数据（用“MOVX A, @DPTR”指令或“MOV A, P1”等指令），数据就被 CPU 取走（送到 CPU 里的累加器 A 中）；如果是 CPU 在向某存储器单元或 I/O 接口芯片（或 I/O 端口）写数据（用“MOVX @DPTR, A”指令或“MOV P1, A”等指令），数据就被送到（写入）某存储器单元或 I/O 接口芯片（或 I/O 端口）。控制总线用来传送各种命令。在 MCS-51 系列单片机中，为了节省引脚，CPU 与外部存储器和 I/O 接口电路芯片的连接中，地址总线的低 8 位和数据总线用一个 P0 口（8 位）的 8 根线，由 CPU 控制分时切换。下面分别简单介绍各部件功能和结构。

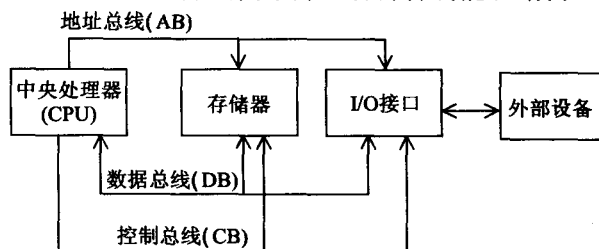


图 1-1