



新世纪高职高专实用规划教材

机电系列

单片机原理与 应用技术

姚国林 主 编
张同光 副主编
卢宇清 主 审
苏 闯

赠送
电子课件

清华大学出版社



新世纪高职高专实用规划教材 机电系列

单片机原理与应用技术

姚国林 主 编

苏 闯 张同光 副主编

卢宇清 主 审

清华大学出版社

北 京

内 容 简 介

本书以国内广泛使用的 MCS-51 系列单片机中的 8051 为对象,介绍了它的基本结构、工作原理、指令系统和基本的程序设计方法,以及 MCS-51 内部的主要资源,包括定时/计数器、中断系统、内部接口、串行通信接口的使用方法,重点介绍了 MCS-51 单片机的常用接口及控制技术和单片机应用系统开发及应用技术。针对单片机原理及应用,本着理论必须够用的原则,突出实用性、操作性,在编排上由浅入深,循序渐进,精选内容,突出重点,适当增加一些当今流行的新器件和新技术;对于接口技术和应用系统提供了详细的原理说明、电路图、完整的程序代码及程序流程图。

本书可作为高职高专院校自动化、电子信息、机电、电力和计算机等专业的教材,也可以作为工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

单片机原理与应用技术/姚国林主编;苏闯,张同光副主编;卢宇清主审.—北京:清华大学出版社,2009.7
(新世纪高职高专实用规划教材 机电系列)
ISBN 978-7-302-20351-3

I. 单… II. ①姚… ②苏… ③张… ④卢… III. 单片微型计算机—高等学校:技术学校—教材
IV. TP368.1

中国版本图书馆 CIP 数据核字(2009)第 095546 号

责任编辑:石 伟 宋延清

封面设计:山鹰工作室

版式设计:杨玉兰

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:16.5 字 数:392 千字

版 次:2009 年 7 月第 1 版 印 次:2009 年 7 月第 1 次印刷

印 数:1~4000

定 价:26.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:031887-01

前 言

单片机作为嵌入式微控制器在工业测控系统、智能仪器和家用电器中得到了广泛的应用。虽然单片机的种类很多,但 MCS-51 系列单片机仍然是单片机中的主流机型。

本书以 MCS-51 系列单片机为主介绍单片机的原理与应用,内容系统、全面,论述深入浅出、循序渐进,注重接口技术和应用。

本书由从事教学工作的一线教师编写,在编写过程中,融入了编者多年教学、科研的经验与应用实例。从应用的角度出发,对单片机的硬件结构、工作原理、指令系统进行了简明扼要的介绍;对程序设计方法、接口电路设计、应用系统等进行了详细的介绍。并提供了详细的原理、电路图、完整的程序代码及程序流程图。

本书以单片机应用能力培养为主线,从应用的角度出发,按照“知识为技能服务,技能为综合能力和素质服务”的思想精心组织内容。在教学中,采用“学、练、用”相结合的构架,使学生能够循序渐进地学习和使用单片机,实现学习基础知识与开展课题训练的巧妙融合——在学中做,在做中学,为综合应用打基础;在必要的学习和训练环节结束后,综合运用所学知识完成工程性实习项目的设计和调试。

本书在编写过程中,承蒙青岛伟立精工塑胶有限公司副总经理王明伟、经理田野给予帮助和指导,在此特别致谢。

本书共分 9 章,主要内容包括绪论、单片机系统开发、MCS-51 单片机的体系结构、MCS-51 指令系统、汇编程序设计、MCS-51 中断系统及定时/计数器、MCS-51 单片机的串口通信、单片机接口及控制技术、MCS-51 单片机应用系统的设计。

本书由河南农业职业学院的姚国林老师任主编,由河南农业职业学院的苏闯老师和新乡学院的张同光老师任副主编,河南农业职业学院的张剑锋、陈慕君、王海娜、郑传琴、史兴燕以及南阳幼儿师范学校的刘海申老师也参加了编写。

具体编写分工为:郑传琴编写第 1 章、附录 1 及附录 2,张剑锋编写第 2 章,张同光编写第 3 章,陈慕君编写第 4 章,王海娜编写第 5 章,刘海申编写第 6 章,苏闯编写第 7 章,姚国林编写第 8 章,史兴燕编写第 9 章,最后由姚国林统稿。

本书由河南农业职业学院卢宇清副教授主审,在审稿过程中提出了许多建设性的建议和意见。在编写过程中得到了许多专家和同行的大力支持和热情帮助,同时,我们也参考了有关教材、论文和著作,在此一并表示衷心的感谢。

鉴于一线教师教科研工作繁重,加之新的单片机芯片不断涌现,其应用技术也在不断发展,书中难免会有错误或不妥之处,恳请广大同行及读者不吝指正。

编 者

目 录

第 1 章 绪论 1

1.1 数制与编码的简单回顾 1

1.1.1 计算机中的数制及相互转换 1

1.1.2 二进制数的运算 4

1.1.3 带符号数的表示 6

1.1.4 带符号数运算时的溢出问题 8

1.1.5 定点数和浮点数 8

1.1.6 BCD 码和 ASCII 码 9

1.2 单片机概述 10

1.2.1 电子计算机的问世及其

经典结构 10

1.2.2 微型计算机的组成及其

应用形态 11

1.2.3 单片机的发展过程 12

1.2.4 单片机的特点 13

1.2.5 单片机的应用领域 13

1.2.6 单片机的产品近况 14

习题 1 15

第 2 章 MCS-51 单片机的体系结构 16

2.1 MCS-51 单片机的基本组成 16

2.1.1 80C51 单片机的基本结构 16

2.1.2 MCS-51 单片机的内部组成

及信号引脚 18

2.1.3 存储器结构 21

2.1.4 80C51 单片机的特殊

功能寄存器 24

2.2 并行输入/输出口结构 26

2.2.1 P0 口 27

2.2.2 P1 口 28

2.2.3 P2 口 29

2.2.4 P3 口 30

2.2.5 并行接口的负载能力 31

2.3 时钟及复位电路 31

2.3.1 时钟电路及时序 31

2.3.2 单片机的复位电路 34

2.4 MCS-51 单片机的最小系统 35

2.4.1 单片机最小应用系统举例 35

2.4.2 最小应用系统设计 35

习题 2 36

第 3 章 MCS-51 指令系统 38

3.1 指令系统概述 38

3.1.1 机器指令编码格式 38

3.1.2 符号指令格式 39

3.1.3 符号指令格式及注释中的

常用符号 40

3.2 寻址方式 40

3.2.1 寄存器寻址 41

3.2.2 直接寻址 41

3.2.3 寄存器间接寻址 42

3.2.4 立即寻址 43

3.2.5 变址寻址 43

3.2.6 相对寻址 44

3.2.7 位寻址 45

3.3 数据传送类指令 45

3.3.1 一般传送类指令 45

3.3.2 特殊传送类指令 47

3.4 算术运算类指令 49

3.4.1 加法 50

3.4.2 减法 52

3.4.3 乘法 53

3.4.4 除法 53

3.5 逻辑运算与循环类指令 53

3.5.1 逻辑与 53

3.5.2 逻辑或 54

3.5.3 逻辑异或 54

3.5.4 累加器清 0 和取反 55

3.5.5 累加器循环移位 55

3.6 控制转移类指令.....	56	5.1.1 MCS-51 的中断系统结构	96
3.6.1 无条件转移	56	5.1.2 MCS-51 的中断源	97
3.6.2 条件转移	57	5.1.3 MCS-51 中断的控制	99
3.6.3 调用与返回	58	5.2 MCS-51 单片机中断处理过程	101
3.6.4 空操作	59	5.2.1 中断响应条件和时间	101
3.7 位操作类指令	59	5.2.2 中断响应过程	102
3.7.1 位传送	60	5.2.3 中断返回	102
3.7.2 位状态设置	60	5.2.4 中断程序举例	103
3.7.3 位逻辑运算	60	5.3 MCS-51 的定时/计数器	107
3.7.4 位判跳(条件转移).....	61	5.3.1 定时/计数器的结构和 工作原理	107
习题 3	61	5.3.2 定时/计数器的控制	108
第 4 章 汇编程序设计	64	5.3.3 定时/计数器的工作方式	109
4.1 汇编程序设计概述	64	5.3.4 定时/计数器用于外部 中断扩展	111
4.1.1 程序编制的方法和技巧	64	5.3.5 定时/计数器应用举例	111
4.1.2 伪指令	67	习题 5	114
4.2 顺序程序设计	69	第 6 章 MCS-51 单片机的串口通信	115
4.2.1 数据传送	69	6.1 串口通信的基本知识	115
4.2.2 查表程序	70	6.1.1 通信的基本概念	115
4.2.3 简单运算	71	6.1.2 串行通信的分类	115
4.3 分支程序设计	71	6.1.3 串行通信的制式	116
4.3.1 分支程序实例	72	6.1.4 串行通信接口标准	117
4.3.2 分支程序结构	75	6.2 MCS-51 单片机的串口及控制 寄存器	119
4.4 循环程序设计	76	6.2.1 MCS-51 串行口结构	119
4.4.1 循环程序实例	76	6.2.2 MCS-51 串行控制寄存器	120
4.4.2 循环程序结构	81	6.3 串口的工作方式	121
4.5 子程序及其调用	83	6.3.1 方式 0	121
4.5.1 子程序的调用	83	6.3.2 方式 1	122
4.5.2 现场的保护和恢复	83	6.3.3 方式 2 和方式 3	123
4.5.3 参数传递	84	6.3.4 波特率的计算	124
4.6 常用汇编子程序	86	6.4 串口的应用	125
4.6.1 代码转换程序	86	6.4.1 双机通信	125
4.6.2 算术运算程序	89	6.4.2 多机通信	129
4.6.3 查找、排序程序	91	习题 6	135
习题 4	94		
第 5 章 MCS-51 中断系统及 定时/计数器	96		
5.1 MCS-51 的中断系统	96		

第7章 单片机接口及控制技术.....136

7.1 简单 I/O 口控制	136
7.1.1 原理及流水灯电路	136
7.1.2 控制程序及流程图	137
7.2 数码管显示	138
7.2.1 原理及控制电路	138
7.2.2 控制程序及流程图	141
7.3 键盘及接口	145
7.3.1 键盘原理及控制电路	145
7.3.2 控制程序及流程图	147
7.4 键盘及显示综合实例——秒表	150
7.4.1 功能说明	150
7.4.2 关键技术及控制电路	150
7.4.3 控制程序及流程图	151
7.5 LCD 点阵字符型液晶显示器	158
7.5.1 TC1602A 简介	158
7.5.2 控制电路	162
7.5.3 控制程序及流程图	162
7.6 DS1302 实时时钟	165
7.6.1 实时时钟 DS1302	165
7.6.2 控制电路	168
7.6.3 控制程序及流程图	169
7.7 ADC0809 电压检测电路	177
7.7.1 A/D 转换器概述	177
7.7.2 典型 A/D 转换器芯片	
ADC0809	178
7.7.3 ADC0809 电压测量电路	181
7.7.4 控制程序	182
习题 7	187

第8章 MCS-51 单片机应用系统的设计.....188

8.1 单片机应用系统概述	188
8.1.1 单片机应用系统的特点	188
8.1.2 MCS-51 单片机应用系统设计方法	188
8.2 课程设计——16×16 LED 显示	191

8.2.1 设计要求	191
8.2.2 16×16 LED 显示总体设计方案	191
8.2.3 硬件设计	193
8.2.4 编程要点及软件设计	193
8.3 课程设计——电脑钟	199
8.3.1 设计要求	199
8.3.2 电脑钟总体设计方案	200
8.3.3 硬件设计	200
8.3.4 软件设计及流程图	201
习题 8	216

第9章 MCS-51 单片机的 C51 程序设计.....217

9.1 C51 语言概述和程序结构	217
9.1.1 C51 语言的特点	217
9.1.2 程序结构	218
9.2 标识符和关键字	221
9.3 C51 语言数据类型和运算符	223
9.3.1 C51 语言的数据类型	223
9.3.2 C51 语言的运算符	231
9.4 C51 程序的基本结构	235
9.4.1 if 语句	235
9.4.2 switch case 语句	236
9.4.3 循环语句	237
9.5 C51 函数和预处理命令	238
9.5.1 函数的分类和定义	238
9.5.2 中断服务函数	240
9.5.3 C51 的库函数	241
9.5.4 编译预处理命令	243
9.6 C51 程序设计举例	244
习题 9	245

附录1 MCS-51 指令表.....246

附录2 ASCII 表.....250

参考文献.....251

第1章 绪 论

冯·诺依曼提出了“程序存储”和“二进制运算”的思想，并构建了计算机的经典结构。以此为基础，随着社会的发展，为了满足工业控制的需要，产生了单片机。单片机是一片集成芯片，它具有极适合于在智能仪表和工业控制的前端装置中使用的特点。在当前，80C51 系列单片机应用广泛、生产量大，在单片机领域里具有重要的影响。本章主要介绍有关 80C51 系列单片机学习的基础知识，以及单片机的发展、特点、应用领域及产品近况，和单片机应用系统的开发过程。

1.1 数制与编码的简单回顾

1.1.1 计算机中的数制及相互转换

1. 进位计数制

按进位原则进行计数的方法，称为进位计数制。在我们的日常生活中经常用到的是十进位计数制，简称十进制。十进制数的主要特点如下。

有 10 个不同的数字符号：0、1、2、...、9。

低位向高位进位的规律是“逢十进一”。因此，即使同一个数字符号，若在不同的数位，它所代表的数值也是不同的。如 555.5 中 4 个 5 分别代表 500、50、5 和 0.5，这个数可以写成 $555.5 = 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1}$ ，式中的 10 称为十进制的基数， 10^2 、 10^1 、 10^0 、 10^{-1} 称为各数位的权。

任意一个十进制数 N 都可以表示成按权展开的多项式：

$$N = d_{n-1} \times 10^{n-1} + d_{n-2} \times 10^{n-2} + \dots + d_i \times 10^i + \dots + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \dots + d_{-m} \times 10^{-m}$$

式中， d_i 是 0~9 (共 10 个数字) 中的任意一个， i 是数位的序数， m 是小数点右边的位数， n 是小数点左边的位数，10 是基数。

如 543.21 可表示为 $543.21 = 5 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1} + 1 \times 10^{-2}$ 。

一般而言，对于用 R 进制表示的数 N ，可以按权展开为：

$$N = a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \dots + a_i \times R^i + \dots + a_0 \times R^0 + a_{-1} \times R^{-1} + \dots + a_{-m} \times R^{-m}$$

式中， a_i 是数字 0、1、...、 $(R-1)$ 中的任一个， i 是数位的序数， m 是小数点右边的位数， n 是小数点左边的位数， R 是基数。在 R 进制中，每个数字所表示的值是该数字与它相应的权 R^i 的乘积，计数原则是“逢 R 进一”。

(1) 二进制数

当 $R=2$ 时，称为二进位计数制，简称二进制。在二进制数中，只有两个不同数码：0 和 1，进位规律为“逢二进一”。任何一个数 N ，用二进制可以表示为：

$$N = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_0 \times 2^0 + a_{-1} \times 2^{-1} + \dots + a_{-m} \times 2^{-m}$$

【例 1.1】二进制数 1011.01 可表示为:

$$(1011.01)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

(2) 八进制数

当 $R=8$ 时,称为八进制。在八进制数中,有 0、1、2、...、7,共 8 个不同的数码,采用“逢八进一”的原则进行计数。

【例 1.2】八进制数 503 可表示为:

$$(503)_8 = 5 \times 8^2 + 0 \times 8^1 + 3 \times 8^0$$

(3) 十六进制

当 $R=16$ 时,称为十六进制。在十六进制数中,有 0、1、2、...、9、A、B、C、D、E、F,共 16 个不同的数码,进位方法是“逢十六进一”。

【例 1.3】 $(3A8.0D)_{16}$ 可表示为:

$$(3A8.0D)_{16} = 3 \times 16^2 + 10 \times 16^1 + 8 \times 16^0 + 0 \times 16^{-1} + 13 \times 16^{-2}$$

表 1-1 列出了几种常见进制的对应关系。

表 1-1 几种常见进制的对应关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8				

一般在书写时,二进制数的后面加字母“B”,八进制数的后面加字母“Q”,十进制数的后面加字母“D”或什么也不加,十六进制数的后面加字母“H”。

2. 不同进制间的相互转换

(1) 二、八、十六进制转换成十进制

【例 1.4】将 $(10.101)_2$ 、 $(46.12)_8$ 、 $(2D.A4)_{16}$ 转换为十进制数。

$$(10.101)_2 = 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 2.625$$

$$(46.12)_8 = 4 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2} = 38.15625$$

$$(2D.A4)_{16} = 2 \times 16^1 + 13 \times 16^0 + 10 \times 16^{-1} + 4 \times 16^{-2} = 45.64062$$

(2) 十进制数转换成二、八、十六进制数

任意十进制数 N 转换成 R 进制数时,需将整数部分和小数部分分开,采用不同方法分别进行转换,然后用小数点将这两部分连接起来。

① 整数部分:除基取余倒序法。

分别用基数 $R(R=2、8$ 或 $16)$ 不断地去除 N 的整数,直到商为零为止,每次所得的余

数依次排列,即为相应进制的数码。最初得到的为最低位有效数字,最后得到的为最高位有效数字。

【例 1.5】 将 $(168)_{10}$ 分别转换成二、八、十六进制数。

商	余数	商	余数	商	余数
$168 \div 2 = 84 \dots 0$		$168 \div 8 = 21 \dots 0$		$168 \div 16 = 10 \dots 8$	
$84 \div 2 = 42 \dots 0$		$21 \div 8 = 2 \dots 5$		$10 \div 16 = 0 \dots A$	
$42 \div 2 = 21 \dots 0$		$2 \div 8 = 0 \dots 2$		所以 $(168)_{10} = (A8)_{16}$	
$21 \div 2 = 10 \dots 1$		所以 $(168)_{10} = (250)_8$			
$10 \div 2 = 5 \dots 0$					
$5 \div 2 = 2 \dots 1$					
$2 \div 2 = 1 \dots 0$					
$1 \div 2 = 0 \dots 1$					
所以 $(168)_{10} = (10101000)_2$					

② 小数部分: 乘基取整法。

分别用基数 $R(R=2, 8 \text{ 或 } 16)$ 不断地去乘 N 的小数,直到积的小数部分为零(或直到所要求的位数)为止,每次乘得的整数依次排列,即为相应进制的数码。最初得到的为最高位有效数字,最后得到的为最低位有效数字。

【例 1.6】 将 $(0.645)_{10}$ 分别转换成二、八、十六进制数。

整数	0.645	整数	0.645	整数	0.645
	$\times 2$		$\times 8$		$\times 16$
1 ...	1.290	5 ...	5.160	A ...	10.320
	0.29		0.16		0.32
	$\times 2$		$\times 8$		$\times 16$
0 ...	0.58	1 ...	1.28	5 ...	5.12
	0.58		0.28		0.12
	$\times 2$		$\times 8$		$\times 16$
1 ...	1.16	2 ...	2.24	1 ...	1.92
	0.16		0.24		0.92
	$\times 2$		$\times 8$		$\times 16$
0 ...	0.32	1 ...	1.92	E ...	14.72
	0.32		0.92		0.72
	$\times 2$		$\times 8$		$\times 16$
0 ...	0.64	7 ...	7.36	B ...	11.52

所以 $(0.645)_{10} = (0.10100)_2 = (0.51217)_8 = (0.A51EB)_{16}$

【例 1.7】 将 $(168.645)_{10}$ 分别转换成二、八、十六进制数。

根据例 1.5 和例 1.6 可得:

$(168.645)_{10} = (10101000.10100)_2 = (250.51217)_8 = (A8.A51EB)_{16}$

(3) 二进制与八进制、十六进制之间的相互转换

由于 $2^3=8$ (或 $2^4=16$),故可采用“合三为一”(或合四为一)的原则,即从小数点开始,

分别向左、右两边各以 3(4)位为一组进行二进制与八进制(十六进制)之间的换算: 若不足 3(4)位的以 0 补足, 便可将二进制数转换为八(十六)进制数。反之, 采用“一分为三”(“一分为四”)的原则, 每位八(十六)进制数码用三(四)位二进制数表示, 就可将八(十六)进制数转换为二进制数。

【例 1.8】 将 $(101011.01101)_2$ 转换为八进制数。

101	011	.	011	010
↓	↓	↓	↓	↓
5	3	.	3	2

所以 $(101011.01101)_2 = (53.32)_8$

【例 1.9】 将 $(123.45)_8$ 转换成二进制数。

1	2	3	.	4	5
↓	↓	↓	↓	↓	↓
001	010	011	.	100	101

所以 $(123.45)_8 = (1010011.100101)_2$

【例 1.10】 将 $(110101.011)_2$ 转换为十六进制数。

0011	0101	.	0110
↓	↓	↓	↓
3	5	.	6

所以 $(110101.011)_2 = (35.6)_{16}$

【例 1.11】 将 $(4A5B.6C)_{16}$ 转换为二进制数。

4	A	5	B	.	6	C
↓	↓	↓	↓	↓	↓	↓
0100	1010	0101	1011	.	0110	1100

所以 $(4A5B.6C)_{16} = (100101001011011.011011)_2$

1.1.2 二进制数的运算

1. 二进制数的算术运算

二进制数只有 0 和 1 两个数字, 其算术运算较为简单, 加、减法遵循“逢二进一”、“借一当二”的原则。

(1) 加法运算

规则: $0+0=0$; $0+1=1$; $1+0=1$; $1+1=10$ (有进位)。

【例 1.12】 求 $1001B+1011B$ 。

被加数	1001	
加数	+1011	
进位	10010	
和	10100B	

所以 $1001B+1011B=10100B$

(2) 减法运算

规则: $0-0=0$; $1-1=0$; $1-0=1$; $0-1=1$ (有借位)。

【例 1.13】求 $1100\text{B}-111\text{B}$ 。

$$\begin{array}{r} \text{被减数} \quad 1100 \\ \text{减数} \quad \quad -111 \\ \hline \text{借位} \quad 0110 \\ \text{差} \quad \quad 0101 \end{array}$$

所以 $1100\text{B}-111\text{B}=101\text{B}$

(3) 乘法运算

规则: $0 \times 0=0$; $0 \times 1=1 \times 0=0$; $1 \times 1=1$ 。

【例 1.14】求 $1011\text{B} \times 1101\text{B}$ 。

$$\begin{array}{r} \text{被乘数} \quad 1011 \\ \text{乘数} \quad \quad \times 1101 \\ \hline \quad \quad \quad 1011 \\ \quad \quad 0000 \\ \quad 1011 \\ +1011 \\ \hline \text{积} \quad 10001111 \end{array}$$

所以 $1011\text{B} \times 1101\text{B}=10001111\text{B}$

(4) 除法运算

规则: $0/1=0$; $1/1=1$ 。

【例 1.15】求 $10100101\text{B}/1111\text{B}$ 。

$$\begin{array}{r} \quad \quad \quad 1011 \\ \overline{) 10100101} \\ \underline{1111} \\ 1011 \\ \underline{0000} \\ 10110 \\ \underline{1111} \\ 1111 \\ \underline{1111} \\ 0 \end{array}$$

所以 $10100101\text{B}/1111\text{B}=1011\text{B}$

2. 二进制数的逻辑运算

(1) “与”运算

“与”运算是实现“必须都有, 否则就没有”这种逻辑关系的一种运算, 其运算符为“ $\cdot$ ”, 其运算规则为: $0 \cdot 0=0$; $0 \cdot 1=1 \cdot 0=0$; $1 \cdot 1=1$ 。



【例 1.16】若 $X=1011B$, $Y=1001B$, 求 $X \cdot Y$ 。

$$\begin{array}{r} 1011 \\ \cdot 1001 \\ \hline 1001 \end{array}$$

所以 $X \cdot Y=1001B$

(2) “或”运算

“或”运算是实现“只要其中之一有，就有”这种逻辑关系的一种运算，其运算符为“+”，“或”运算规则为： $0+0=0$ ； $0+1=1+0=1$ ； $1+1=1$ 。

【例 1.17】若 $X=10101B$, $Y=01101B$, 求 $X+Y$ 。

$$\begin{array}{r} 10101 \\ + 01101 \\ \hline 11101 \end{array}$$

所以 $X+Y=11101B$

(3) “非”运算

“非”运算是实现“求反”这种逻辑的一种运算，如变量 A 的“非”运算记作 $\bar{A}$ ，其运算规则为： $\bar{1}=0$ ； $\bar{0}=1$ 。

【例 1.18】若 $A=10101B$, 求 $\bar{A}$ 。

$$\bar{A} = \overline{10101} = 01010B$$

(4) “异或”运算

“异或”运算是实现“必须不同，否则就没有”这种逻辑的一种运算，运算符为“ $\oplus$ ”，其运算规则为： $0 \oplus 0=0$ ； $0 \oplus 1=1$ ； $1 \oplus 0=1$ ； $1 \oplus 1=0$ 。

【例 1.19】若 $X=1010B$, $Y=0110B$, 求 $X \oplus Y$ 。

$$\begin{array}{r} 1011 \\ \oplus 0110 \\ \hline 1100 \end{array}$$

即 $X \oplus Y=1100B$

1.1.3 带符号数的表示

1. 机器数及其真值

数在计算机内的表示形式称为机器数。而这个数本身称为该机器数的真值。例如：

正数 $+100\ 0101B(+45H)$ 可以表示成 $0100\ 0101B$ ；机器数 $45H$ 。

负数 $-101\ 0101B(-55H)$ 可以表示成 $1101\ 0101B$ ；机器数 $D5H$ 。

“ $45H$ ”和“ $D5H$ ”为两个机器数，它们的真值分别为“ $+45H$ ”和“ $-55H$ ”。

2. 原码和反码

(1) 原码

带符号二进制数(字节、字或双字)，直接用最高位表示数的符号，数值用其绝对值表

示的形式称为该数的原码。

(2) 反码

正数的反码与其原码相同；负数的反码符号位为 1，数值位为其原码数值位逐位取反。二进制数采用原码和反码表示时，符号位不能同数值一道参加运算。

3. 补码

在计算机中，带符号数的运算均采用补码。正数的补码与其原码相同；负数的补码为其反码末位加 1。例如：

正数+100 0101B，反码为 0100 0101B，补码为 0100 0101B，(45H)。

负数-101 0101B，反码为 1010 1010B，补码为 1010 1011B，(ABH)。

已知一个负数的补码，求真值的方法是：对该补码求补(符号位不变，数值位取反加 1)即得到该负数的原码(符号位+数值位)，依该原码可知其真值。例如有一数：

补码为 1010 1011B

求补得 1101 0101B

真值为-55H

补码的优点是可以将减法运算转换为加法运算，同时数值连同符号位可以一起参加运算。这非常有利于计算机的实现。例如：

45H-55H=-10H，用补码运算时表示为：

[45H]补+[-55H]补=[-10H]补

[45H]补：0100 0101

+ [-55H]补：1010 1011

结果：1111 0000

结果 1111 0000B 为补码，求补得到原码为 1001 0000B，真值为-001 0000B(即-10H)。

可见，采用反码时，“0”有两种表示方式，即有“+0”和“-0”之分，单字节表示范围是+127~-127；而采用补码时，“0”只有一种表示方式，单字节表示的范围是+127~-128。表 1-2 列出了几个典型的带符号数据的原码、反码和补码。

表 1-2 几个典型的带符号数据的 8 位编码表

真 值	原 码	反 码	补 码
+127	0111 1111B	0111 1111B	0111 1111B (7FH)
+1	0000 0001B	0000 0001B	0000 0001B (01H)
+0	0000 0000B	0000 0000B	0000 0000B (00H)
-0	1000 0000B	1111 1111B	0000 0000B (00H)
-1	1000 0001B	1111 1110B	1111 1111B (FFH)
-127	1111 1111B	1111 0000B	1000 0001B (81H)
-128	...	...	1000 0000B (80H)

1.1.4 带符号数运算时的溢出问题

两个带符号数进行加减运算时，若运算结果超出了机器所允许表示的范围，得出错误的结果，这种情况称为溢出。如 8 位字长的计算机所能表示的有符号数的范围为-128~+127，若运算结果超出此范围，就会发生溢出。

判断的方法：对加(减)运算，判断最高位与次高位的进(借)位情况是否相同，若相同，则无溢出；若不同，则有溢出。

【例 1.20】判断下列运算的溢出情况。

(1) $(+93) + (+54)$

0101	1101B	[+93]补
+ 0011	0110B	[+54]补
1001	0011B	[-109]补

由上式可以看出次高位有进位，最高位无进位，所以，有溢出发生，结果出错。

(2) $(-63) + (+70)$

1100	0001B	[-63]补
+ 0100	0110B	[+70]补
[1]0000	0110B	[+7]补

由上式可以看出次高位有进位，最高位有进位，所以，无溢出发生，结果正确。

1.1.5 定点数和浮点数

1. 定点法

定点法就是规定一个固定的小数点位置，一般来说，小数点规定在哪个位置上并没有限制，但为了方便，通常把数化为纯小数或纯整数，那么定点数就有下面两种表示方法。

符号位	数值位	或	符号位	数值位
-----	-----	---	-----	-----

2. 浮点法

浮点法就是数据中的小数点位置不是固定不变的，而是可浮动的。因此，可将任意一个二进制数 N 表示成：

$$N = \pm M \cdot 2^{\pm E}$$

其中， M 为尾数，为纯二进制小数， E 称为阶码。可见，一个浮点数有阶码和尾数两部分，且都带有表示正负的阶码符与数符，其格式为：

阶符	阶码 E	数符	尾数 M
----	--------	----	--------

设阶码 E 的位数为 m 位，尾数 M 的位数为 n 位，则浮点数 N 的取值范围为：

$$2^n 2^{-2m+1} \leq |N| \leq (1-2^{-n}) 2^{2m-1}$$

为了提高精度,发挥尾数有效位的最大作用,还规定尾数数字部分原码的最高位为1,叫做规格化表示法。

如 0.000101 表示为 $2^{-3} \times 0.101$ 。

1.1.6 BCD 码和 ASCII 码

1. 字符的二进制编码——ASCII 码

字符的编码经常采用美国标准信息交换码(American Standard Code for Information Interchange, ASCII)。

一个字节的 8 位二进制码可以表示 256 个字符。当最高位为“0”时,所表示的字符为标准 ASCII 码,共 128 个,用于表示数字、英文大写字母、英文小写字母、标点符号及控制字符等,如本书附录 2 的 ASCII 表所示。

ASCII 码常用于计算机与外围设备的数据传输。如通过键盘的字符输入,通过打印机或显示器的字符输出等,常用字符的 ASCII 码如表 1-3 所示。

表 1-3 常用字符的 ASCII 码

字 符	ASCII 码	字 符	ASCII 码	字 符	ASCII 码	字 符	ASCII 码
0	30H	A	41H	a	61H	SP(空格)	20H
1	31H	B	42H	b	62H	CR(回车)	0DH
2	32H	C	43H	c	63H	LF(换行)	0AH
⋮	⋮	⋮	⋮	⋮	⋮	BEL(响铃)	07H
9	39H	Z	5AH	z	7AH	BS(退格)	08H

通常,7 位 ASCII 码在最高位添加一个“0”组成 8 位代码,因此,字符在计算机内部存储正好占用一个字节。在存储和传送时,最高位常用作奇偶校验位,用于检查代码传送过程是否出现差错。偶校验时,每个二进制编码中应有偶数个“1”,奇校验时,每个二进制编码中应有奇数个“1”。例如,字母“F”的 ASCII 码为 1000110,因有 3 个“1”,若采用偶校验传送该字符,则奇偶校验位应为“1”,传送的代码为 11000110;若采用奇校验传送该字符,则奇偶校验位应为“0”,传送的代码为 01000110。

应当注意,字符的 ASCII 码与其数值是不同的概念。例如,字符“9”的 ASCII 码是 00111001B(即 39H),而其数值是 00001001B(即 09H)。

在 ASCII 码字符表中,还有许多不可打印的字符,如 CR(回车)、LF(换行)及 SP(空格)等,这些字符称为控制字符。控制字符在不同的输出设备上可能会执行不同的操作(因为没有非常规范的标准)。

2. 二进制编码的十进制数 8421 BCD 码

十进制数是人们在生活中最习惯的数制,人们通过键盘向计算机输入数据时,常用十进制输入。显示器显示数据时,也多采用十进制形式。

由于十进制数有十个不同的数码，因此需要 4 位二进制数来表示。而 4 位二进制数码有 16 种不同的组合，所以表示 0~9 这十个数有多种方案。所以，BCD 码也有多种方案。最常用的编码是 8421 BCD 码，它是一种恒权码， $8(2^3)$ 、 $4(2^2)$ 、 $2(2^1)$ 、 $1(2^0)$ 分别是 4 位二进制数的权值。用二进制码表示十进制数的代码称为 8421 BCD 码。十进制数 0~9 所对应的 8421 BCD 码如表 1-4 所示。

表 1-4 0~9 所对应的 8421 BCD 码表

十进制数	BCD 码	十进制数	BCD 码
0	0000B	5	0101B
1	0001B	6	0110B
2	0010B	7	0111B
3	0011B	8	1000B
4	0100B	9	1001B

用 1 个字节表示 2 位十进制数的代码，称为压缩的 BCD 码。相对于压缩的 BCD 码，用 8 位二进制码表示的 1 位十进制数的编码称为非压缩的 BCD 码。这时高 4 位无意义，低 4 位是 BCD 码。采用压缩的 BCD 码比采用非压缩的 BCD 码节省存储空间。当 4 位二进制码在 1010B~1111B 范围时，不属于 8421 BCD 码的合法范围，称为非法码。两个 BCD 码的运算可能出现非法码，这时要对所得结果进行调整。

1.2 单片机概述

1.2.1 电子计算机的问世及其经典结构

1946 年 2 月 15 日，第一台电子数字计算机 ENIAC 问世，这标志着计算机时代的到来。ENIAC 是电子管计算机，时钟频率仅有 100kHz，但能在 1 秒钟的时间内完成 5000 次加法运算。与现代的计算机相比，有许多不足，但它的问世开创了计算机科学技术的新纪元，对人类的生产和生活方式产生了巨大的影响。

匈牙利籍数学家冯·诺依曼在方案的设计上做出了重要的贡献。1946 年 6 月，他又提出了“程序存储”和“二进制运算”的思想，进一步构建了计算机由运算器、控制器、存储器、输入设备和输出设备组成这一计算机的经典结构，如图 1-1 所示。

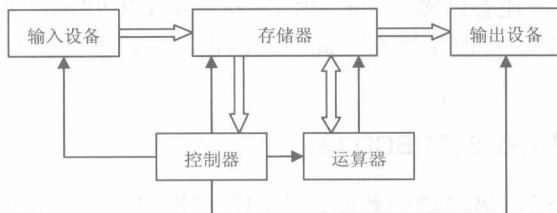


图 1-1 电子计算机的经典结构