



中等职业学校电子信息类教材 计算机技术专业

单片机原理及应用

(第3版)

韩太林 主编
刘红 于林韬 编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

本书配有电子教学参考资料包

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

中等职业学校电子信息类教材（计算机技术专业）

单片机原理及应用 (第3版)

韩太林 主编

刘 红 于林韬 编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以通俗易懂的语言系统详尽地介绍了 MCS-51 单片机的结构、系统设计、调试方式及应用实例。内容包括单片机的结构、指令系统、程序设计、系统扩展和典型单片机系统等。本书有较强的系统性、实用性和先进性,内容由浅入深并配有习题。本书适用于中等职业学校计算机专业课程教学,也可以作为相关专业的培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

单片机原理及应用 / 韩太林主编. —3 版. —北京: 电子工业出版社, 2005.1
中等职业学校电子信息类教材·计算机技术专业
ISBN 7-120-00075-6

I. 单… II. 韩… III. 单片微型计算机—专业学校—教材 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2004) 第 062787 号

责任编辑: 蔡 葵

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 13.5 字数: 337.6 千字

印 次: 2007 年 2 月第 4 次印刷

印 数: 3000 册 定价: 17.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话: (010)68279077; 邮购电话: (010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010)88258888。

前 言



本书为读者介绍了 MCS-51 系列单片机的最新发展技术。书中包括了许多新的例题及课外习题。在结构安排上更为合理, 有利于学生对知识的接受。同时增加了目前最新的单片机理论及新的芯片扩展技术。本书由全国中等职业学校电子信息类教材编审委员会组织编写, 可以作为中等职业学校相关专业单片机课程的教材。

作者力图使本书有助于读者使用单片机为各自所从事的工作解决实际问题。因此, 在编写本书的时候, 力求深入浅出、通俗易懂, 并注重理论联系实际, 着重实际应用。全书共分 9 章。第 1 章介绍计算机基础知识。第 2 章至第 5 章分别介绍了 MCS-51 系列单片机结构、指令系统、程序设计及内部资源。第 6 章介绍了 MCS-51 系列单片机系统扩展技术及应用。第 7 章简要介绍了目前比较流行的 Flash 单片机。第 8 章介绍了单片机应用系统设计方法及设计实例, 并介绍了新颖传感器通道接口和应用系统采用的抗干扰措施。第 9 章介绍了几个实验。

本书由长春理工大学韩太林主编, 并编写了第 1 章、第 2 章、第 5 章、第 7 章和第 9 章; 长春理工大学刘红编写了第 3 章、第 6 章、第 8 章和附录; 长春理工大学于林涛编写了第 4 章。编者在教材编写出版过程中, 得到了主、参编各单位领导的大力支持, 在此表示诚挚的谢意, 并对提供大量参考文献资料的专家学者表示衷心的感谢。由于编者水平有限, 时间仓促, 错误和不妥之处在所难免, 敬请使用本书的广大教师和读者批评指正。

另外, 本书配有电子教学参考资料包(包括: 教学指南、电子教案、习题答案), 免费提供给教师使用。需要的教师请与出版社联系。E-mail: ve@phei.com.cn

编 者

2003 年 12 月于长春





第 1 章 计算机基础知识	1
1.1 常用数字转换	1
1.1.1 数制及数字之间的转换	1
1.1.2 常用二进制编码	5
1.2 数据在计算机中的表示	7
1.3 单片微型计算机	9
1.3.1 单片机的发展概况	9
1.3.2 单片机的发展趋势	9
1.3.3 单片机的应用	11
本章小结	11
习题 1	11
第 2 章 MCS-51 单片机结构	13
2.1 MCS-51 单片机内部结构	13
2.1.1 MCS-51 组成	13
2.1.2 CPU	14
2.1.3 存储器	17
2.1.4 I/O 端口	17
2.1.5 总线	19
2.1.6 复位和复位电路	20
2.1.7 MCS-51 引脚说明	21
2.2 MCS-51 存储器	22
2.2.1 程序存储器	22
2.2.2 内部数据存储器	23
2.2.3 外部数据存储器	25
2.3 特殊功能寄存器	25
2.3.1 累加器 ACC	25
2.3.2 B 寄存器	26
2.3.3 程序状态字 PSW	27
2.3.4 栈指针 SP	28
2.3.5 数据指针 DPTR	28
2.3.6 端口寄存器 P0~P3	28

2.3.7 串行数据缓冲器 SBUF	28
2.3.8 定时器/计数器	28
2.3.9 其他寄存器	28
本章小结	28
习题 2	29
第 3 章 指令系统	30
3.1 指令系统概述	30
3.1.1 指令格式	30
3.1.2 指令系统的分类及特点	31
3.1.3 指令中常用的符号说明	31
3.1.4 指令的寻址方式	32
3.2 数据传递类指令	34
3.2.1 以累加器为目的操作数的指令	34
3.2.2 以寄存器 Rn 为目的操作数的指令	35
3.2.3 以直接地址为目的操作数的指令	35
3.2.4 以间接地址为目的操作数的指令	36
3.2.5 十六位数的传递指令	36
3.2.6 累加器 A 与片外 RAM 之间的数据传递类指令	37
3.2.7 程序存储器向累加器 A 传送指令	38
3.2.8 堆栈操作类指令	38
3.2.9 数据交换指令	39
3.3 算术运算类指令	40
3.3.1 不带进位加法指令	40
3.3.2 带进位的加法指令	41
3.3.3 加 1 指令	41
3.3.4 十进制调整指令	42
3.3.5 带进位的减法指令	43
3.3.6 减 1 指令	44
3.3.7 乘法指令	45
3.3.8 除法指令	45
3.4 逻辑操作类指令	46
3.4.1 循环移位指令	46
3.4.2 累加器半字节交换指令	47
3.4.3 求反指令	47
3.4.4 清零指令	47
3.4.5 逻辑与指令	48
3.4.6 逻辑或指令	48
3.4.7 逻辑异或指令	49
3.5 控制转移类指令	50
3.5.1 无条件转移指令	50

3.5.2	条件转移指令	52
3.5.3	调用子程序及返回指令	53
3.5.4	空操作指令	54
3.6	位操作类指令	54
3.6.1	位数据传送指令	55
3.6.2	位状态控制指令	55
3.6.3	位逻辑运算指令	56
3.6.4	位条件转移指令	56
本章小结		57
习题 3		58
第 4 章	汇编语言程序设计	61
4.1	汇编语言的格式	61
4.2	汇编语言程序设计	63
4.2.1	顺序结构程序	64
4.2.2	分支程序	64
4.2.3	循环结构程序	65
4.2.4	子程序设计	67
4.3	MCS-51 汇编语言实用程序举例	67
4.3.1	代码转换类程序	68
4.3.2	运算类程序	70
本章小结		83
习题 4		84
第 5 章	MCS-51 系统内部资源	86
5.1	输入/输出的控制方式	86
5.1.1	程序查询法	86
5.1.2	中断控制方式	88
5.2	中断控制系统	88
5.2.1	中断请求源和中断请求标志	89
5.2.2	中断系统控制	90
5.2.3	中断响应过程	92
5.2.4	外部中断触发方式	93
5.2.5	中断响应时间	93
5.2.6	MCS-51 的单步操作	94
5.2.7	多个外部中断源系统设计	94
5.3	定时器/计数器	96
5.3.1	定时器/计数器内部结构及功能	96
5.3.2	定时器/计数器的工作方式	98
5.3.3	应用举例	100
5.4	串行接口	102

5.4.1 串行口控制寄存器及波特率选择位	102
5.4.2 串行接口工作方式	103
5.4.3 波特率	108
本章小结	109
习题 5	110
第 6 章 MCS-51 系统扩展技术	111
6.1 MCS-51 系统扩展原理	111
6.2 程序存储器的扩展	112
6.2.1 常用的程序存储器	112
6.2.2 MCS-51 程序存储器的扩展	115
6.3 数据存储器的扩展	119
6.3.1 常用的数据存储器	119
6.3.2 MCS-51 扩展外部数据存储器的原理	121
6.3.3 典型的 MCS-51 存储器扩展电路	122
6.4 并行接口的扩展	123
6.4.1 可编程并行接口芯片 8255A	123
6.4.2 用 74 系列器件扩展并行 I/O 口	130
6.5 A/D、D/A 转换技术	131
6.5.1 D/A 转换常用器件——DAC0832	132
6.5.2 A/D 转换常用芯片——ADC0809	135
6.6 七段发光显示器接口	137
6.6.1 显示器的结构	137
6.6.2 显示器的工作方式和显示程序设计	138
6.7 键盘及其接口	141
6.7.1 非编码键盘接口	142
6.7.2 键盘的工作方式	143
本章小结	146
习题 6	147
第 7 章 Flash 单片机简介	148
7.1 ATMEL 公司及其 89 系列单片机发展情况	148
7.1.1 ATMEL 公司简介	148
7.1.2 ATMEL 的 89 系列单片机概况	148
7.2 AT89C2051 Flash 单片机	150
7.2.1 AT89C2051 的概括功能	150
7.2.2 AT89C2051 的结构框图	150
7.2.3 AT89C2051 的引脚说明	150
7.2.4 AT89C2051 的 CPU 工作方式	152
7.2.5 Flash 存储器编程	153
7.3 AT89C51 Flash 单片机	153

7.3.1	AT89C51 的结构框图	154
7.3.2	AT89C51 的引脚功能	155
7.3.3	AT89C51 的 CPU 工作方式	157
本章小结		158
习题 7		159
第 8 章	单片机应用系统	160
8.1	单片机应用系统开发概述	160
8.1.1	应用系统的开发过程	160
8.1.2	单片机应用系统的组成	161
8.1.3	应用系统的硬件开发	161
8.1.4	应用系统的软件设计	163
8.1.5	单片机开发系统	167
8.2	传感器接口技术	174
8.2.1	概述	174
8.2.2	传感器接口电路	175
8.2.3	温度传感器及其应用	176
8.2.4	霍尔传感器及其应用	177
8.3	抗干扰技术	178
8.3.1	计算机电源系统的抗干扰措施	178
8.3.2	过程通道干扰及抗干扰措施	180
本章小结		182
习题 8		182
第 9 章	实验	183
实验 1	数据块传送实验	183
实验 2	数码转换程序实验	184
实验 3	算术运算类指令编程实验	185
实验 4	数字序列排序实验	186
实验 5	查表程序实验	187
实验 6	定时器/计数器实验	187
实验 7	低频信号发生器实验	188
实验 8	I/O 接口和中断实验	190
实验 9	串行口调试实验	191
实验 10	LED 静态显示接口实验	193
附录 A	MCS-51 指令系统分类表	195
附录 B	MCS-51 单片机常用特殊功能寄存器	199
参考文献		202

第1章 计算机基础知识



知识要点

本章主要介绍微型计算机的基本知识，包括微型计算机中采用的数制和编码，数字电路基础知识和单片机的发展概况及发展趋势。这一章是学习单片机的基础，所涉及的内容在相关的章节中有详细论述，本章不做详细的介绍。通过本章的学习使读者对微型计算机有一个初步了解，为后续章节的学习奠定了基础。

1.1 常用数字转换

1.1.1 数制及数字之间的转换

1. 数制

数制是计数的进位制。在计算机系统中常用的数制有：二进制、八进制、十进制和十六进制。其中二进制数是计算机可以进行识别和直接处理的，然而二进制数表达过于复杂，所以引入十六进制以简化表述。十进制是人们最习惯和熟悉的数制。在用单片机解决问题时这四种数制都是不可缺少的。

(1) 十进制 (Decimal)

- 十进制数常以 D 结尾，一般可以省略。
- 十进制有十个元素，即 0, 1, 2, 3, 4, 5, 6, 7, 8, 9。
- 基数为 10，逢十进一。
- 按权展开式为

$$D = \sum_{-m}^{n-1} K_i \times 10^i = K_{n-1} \times 10^{n-1} + K_{n-2} \times 10^{n-2} + \cdots + K_0 \times 10^0 + K_{-1} \times 10^{-1} + \cdots + K_{-m} \times 10^{-m}$$

其中 K_i 表示 D 的第 i 位，权为 10^i ， K_i 从 0~9 十个数字中选用； m 、 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

例如， $34.56D = 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$

(2) 二进制 (Binary)



- 二进制数常以 B 结尾。
- 二进制有二个元素即 0, 1。
- 基数为 2，逢二进一。
- 按权展开式为

$$D = \sum_{i=-m}^{n-1} K_i \times 2^i = K_{n-1} \times 2^{n-1} + K_{n-2} \times 2^{n-2} + \cdots + K_0 \times 2^0 + K_{-1} \times 2^{-1} + \cdots + K_{-m} \times 2^{-m}$$

其中 K_i 表示 B 的第 i 位，权为 2^i ， K_i 从 0, 1 两个数字中选用； m 、 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

例如， $101.101\text{B} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

(3) 八进制 (Octuple)

- 八进制数常以 O 结尾。
- 八进制有八个元素，即 0, 1, 2, 3, 4, 5, 6, 7。
- 基数为 8，逢八进一。
- 按权展开式为

$$D = \sum_{i=-m}^{n-1} K_i \times 8^i = K_{n-1} \times 8^{n-1} + K_{n-2} \times 8^{n-2} + \cdots + K_0 \times 8^0 + K_{-1} \times 8^{-1} + \cdots + K_{-m} \times 8^{-m}$$

其中 K_i 表示 O 的第 i 位，权为 8^i ， K_i 从 0, 1, 2, 3, 4, 5, 6, 7 八个数字中选用； m 、 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

例如， $27.3\text{O} = 2 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1}$

(4) 十六进制 (Hexadecimal)

- 十六进制数常以 H 结尾。
- 十六进制有十六个元素，即 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。
- 基数为十六，逢十六进一。
- 按权展开式为

$$D = \sum_{i=-m}^{n-1} K_i \times 16^i = K_{n-1} \times 16^{n-1} + K_{n-2} \times 16^{n-2} + \cdots + K_0 \times 16^0 + K_{-1} \times 16^{-1} + \cdots + K_{-m} \times 16^{-m}$$

其中 K_i 表示 H 的第 i 位，权为 16^i ， K_i 从 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 十六个字符中选用； m 、 n 为正整数， n 为小数点左边的位数， m 为小数点右边的位数。

例如， $2\text{AB.CDH} = 2 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 12 \times 16^{-1} + 13 \times 16^{-2}$

0~15 的各种数制对照表见表 1.1。

表 1.1 0~15 的各个进制之间转换表

二进制	八进制	十进制	十六进制	二进制	八进制	十进制	十六进制
0000	0	0	0	1000	10	8	8
0001	1	1	1	1001	11	9	9
0010	2	2	2	1010	12	10	A
0011	3	3	3	1011	13	11	B
0100	4	4	4	1100	14	12	C
0101	5	5	5	1101	15	13	D
0110	6	6	6	1110	16	14	E
0111	7	7	7	1111	17	15	F



2. 数制之间的转换

(1) 二进制数与十进制数之间的转换

二进制数转换为十进制数的方法是按权展开后求和。

【例 1.1】 $101.101\text{B}=1\times 2^2+0\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}=5.625\text{D}$

十进制数转换为二进制数的方法要分为整数部分和小数部分来进行。整数部分的转换遵循“除 2 取余，反向读取”的原则。小数部分转换遵循“乘 2 取整，正相读取”的原则。

【例 1.2】 将十进制数 26.625 转换为二进制数。

解：

① 将整数部分 26 转换成二进制数。

2	26	余数	
2	13	0	最低位
2	6	1	
2	3	0	
2	1	1	
	0	1	最高位

所以， $26\text{D}=11010\text{B}$ 。

② 将小数部分 0.625 转换为二进制数。

$0.625\times 2=1.25$	整数为 1	最高位
$0.25\times 2=0.50$	整数为 0	↓
$0.5\times 2=1.00$	整数为 1	最低位

所以， $0.625\text{D}=0.101\text{B}$ 。

综上所述，得

$$26.625\text{D}=11010.101\text{B}$$

(2) 八进制数与十进制数之间的转换

八进制数转换为十进制数的方法是按权展开后求和。

【例 1.3】 $27.3\text{O}=2\times 8^1+7\times 8^0+3\times 8^{-1}=23.125\text{D}$

十进制数转换为八进制数的方法要分为整数部分和小数部分来进行。整数部分的转换遵循“除 8 取余，反向读取”的原则。小数部分转换遵循“乘 8 取整，正相读取”的原则。

【例 1.4】 将十进制数 134.334 转换为八进制数。

解：

① 将整数部分 134 转换成八进制数。

8	134	余数	最低位
8	16	6	
8	2	0	
	0	2	最高位



所以 $134D=206O$ 。

② 将小数部分 0.334 转换为八进制数，取小数后三位。

$$0.334 \times 8 = 2.672 \quad \text{整数为 } 2 \quad \text{最高位}$$

$$0.672 \times 8 = 5.376 \quad \text{整数为 } 5 \quad \downarrow$$

$$0.376 \times 8 = 3.008 \quad \text{整数为 } 3 \quad \text{最低位}$$

所以， $0.334D=0.253O$ 。

综上所述，得

$$134.334D=206.253O。$$

(3) 十六进制数与十进制数之间的转换

十六进制数转换为十进制数的方法同样也是按权展开后求和。

【例 1.5】 $2ABCH=2 \times 16^3 + 10 \times 16^2 + 11 \times 16^1 + 12 \times 16^0 = 10940D$

十进制数转换为十六进制数的方法要分为整数部分和小数部分来进行。整数部分的转换遵循“除 16 取余，反向读取”的原则。小数部分转换遵循“乘 16 取整，正相读取”的原则。

【例 1.6】 将十进制数 1192.359375 转换为十六进制数。

解：

① 将整数部分 1192 转换成十六进制数。

16	1192	余数	
16	74	8	最低位
16	4	A	
	0	4	最高位

所以， $1192D=4A8H$ 。

② 将小数部分 0.359375 转换为十六进制数。

$$0.359375 \times 16 = 5.75 \quad \text{整数为 } 5 \quad \text{最高位}$$

$$0.75 \times 16 = 12.0 \quad \text{整数为 } C \quad \text{最低位}$$

所以， $0.359375D=0.5CH$

综上所述，得

$$1192.359375D=4A8.5CH。$$

(4) 二进制数与八进制数之间的转换

我们可以从前面的定义看出三位的二进制数和一位的八进制数是相对应的。所以二进制数转换为八进制数的方法是：将二进制数以小数点为界，向左右两边每三位一组分开，不足三位补零（前面不足前面补零，后面不足后面补零）。然后将三位二进制数表示为一位八进制数即可。同理将八进制数转换为二进制数时，将每一位八进制数用三位的二进制数表示即可。

【例 1.7】 将二进制数 1011000100010.001101010 转换为八进制数。

解：

$$001, 011, 000, 100, 010.001, 101, 010$$

$$1 \quad 3 \quad 0 \quad 4 \quad 2. \quad 1 \quad 5 \quad 2$$

所以， $1011000100010.001101010B=13042.152O$ 。

(5) 二进制数与十六进制数之间的转换

二进制数与十六进制数之间的转换与二进制数与八进制数之间的转换方法类似。只是四



位的二进制数和一位的十六进制数是相对应的。所以二进制数转换为十六进制数的方法是：将二进制数以小数点为界，向左右两边每四位一组分开，不足四位补零（前面不足前面补零，后面不足后面补零）。然后将四位二进制数表示为一位十六进制数即可。同理将十六进制数转换为二进制数时，将每一位十六进制数用四位的二进制数表示即可。

【例 1.8】 将二进制数 1111010100010.001101010 转换为十六进制数。

解：

0001, 1110, 1010, 0010.0011, 0101, 0000

1 E A 2 .3 5 0

所以，1111010100010.001101010B=1EA2.35H。

1.1.2 常用二进制编码

由于计算机运算采用的是二进制数，因此像数字、标点符号、运算符号、各种命令及各种文字、图形等在计算机内部都以特定的二进制码来表示，这就是二进制编码。例如我们在操作键盘时，输入的是字母、数字或功能符号，但存入机器内部的却是相应的一组二进制编码。计算机中根据信息对象的不同，编码的方式也不同，例如有 BCD 码、ASCII 码、EBCDIC 编码及汉字编码等，下面介绍常用的两种编码——BCD 码和 ASCII 码。

1. BCD 码 (Binary Coded Decimal)

BCD 编码是用 4 位二进制数表示 1 位十进制数的方法。它具有二进制数的形式，又具有十进制数的特点。常用的 BCD 码为 8421BCD 码，即每位十进制数用 4 位二进制数来表示。8421BCD 码是一种有权码，从高位到低位的权分别为 8, 4, 2, 1。四位二进制数有 0000~1111 十六种状态，而这种编码只取 0000~1001 十种状态，详见表 1.2。1010~1111 的六个状态在这种编码中无意义，因此一旦出现它们，就必须加以修正，才能得到正确的结果。

表 1.2 8421BCD 码表

十进制	BCD 码	二进制码
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001,0000	1010
11	0001,0001	1011
12	0001,0010	1100
13	0001,0011	1101
14	0001,0100	1110
15	0001,0101	1111



8421BCD 码用 0000H~1001H 代表十进制数 0~9，运算法则是逢十进一。8421BCD 码每位的权分别是“8”“4”“2”“1”，故得此名。

例如，十进制数 1969 的 BCD 码为 0001 1001 0110 1001。

2. ASCII 码（American Standard Code for Information Interchange）

在微机系统中，除了数字 0~9 以外，还经常使用其他各种字符，例如大写英文字母 A~Z、小写字母 a~z、各种标点符号及控制符号等等。这些信息都要被编成计算机能接受的二进制代码。

目前，国际上普遍采用的字符编码是 ASCII 码。ASCII 是美国标准信息交换代码的缩写。ASCII 规定每个字符用七位二进制代码的不同组合来表示，因而共可定义 $2^7=128$ 个不同的符号，称为基本 ASCII 码字符集。每个 ASCII 码字符用一个字节来存储和表示，其最高位（即左端第一位）一般置 0。这 128 个符号包括 26 个英文大写字母、26 个英文小写字母，0~9 共 10 个数字，32 个通用控制符和 34 个专用字符，见表 1.3。

表 1.3 ASCII 码字符表（美国信息交换标准码）

高位 \ 低位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	DEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

说明：

NUL	空	ETB	信息组传递结束	DLE	数据交换码
SOH	标题开始	CAN	作废	DC1	设备控制 1
STX	正文开始	EM	殖尽	DC2	设备控制 2
ETX	本文结束	SUB	减	DC3	设备控制 3
EOT	传输结束	ESC	换码	DC4	设备控制 4
ENQ	询问	VT	垂直制表	NAK	否定
ACK	承认	FF	走纸控制	FS	文字分隔符
DEL	报警符	CR	回车	GS	组分分隔符
BS	退一格	SO	移位输出	RS	记录分隔符
HT	横向列表	SI	移位输入	US	单元分隔符
LF	换行	SP	空间（空格）	DEL	作废
SYN	空转同步				

为了表示方便，一般不将 ASCII 码直接写成二进制数码，而用其对应的十进制数或十六进制数表示。例如数字 0~9 的 ASCII 码的十六进制码为 30H~39H，字母 A~Z 为 41H~5AH。

在通信中，基本 ASCII 码的第八位（即最高位）常用作奇偶校验位，以检验数据传输的正确性。该位的数值由所要求的奇偶类型确定。

近年来，在基本 ASCII 码的基础上，为了表示更多的符号，将 7 位 ASCII 码扩充到 8



位, 这样便可表示 256 个符号。其中将最高位为 1, 即码值大于十进制数 127 的 ASCII 码, 称为扩展 ASCII 码。扩展 ASCII 码可以表示一些特定的符号, 如希腊字母、数学符号等。扩展的 ASCII 码只有在不用第八位作校验位时才能使用。

我国于 1980 年制订了“信息处理交换用的 7 位编码字符集”, 即国家标准 GB1988—80。除了用人民币符号 ¥ 代替美元符号 \$ 外, 其余代码和含义都与基本 ASCII 码相同。

1.2 数据在计算机中的表示

前面我们介绍的数制, 没有考虑符号问题, 故是一种无符号数。通常意义数的正负号分别用“+”和“-”来表示。在计算机中由于采用二进制数, 只有“0”和“1”两个数字, 因此对于带符号的数, 约定数的最高位为数的符号, 一般情况下, 用 0 表示正数, 用 1 表示负数。例如,

设 $X_1 = +1101101$
 $X_2 = -1101101$

则它们在计算机中的表示为:

$X_1 = 01101101$
 $X_2 = 11101101$

在机器内部, 数字和符号都用二进制代码表示, 两者合在一起构成数的机内表示形式称为机器数; 而把这个数本身, 即用“+”、“-”号表示的数值, 称为真值。

通常机器数有三种表示方法, 即原码、反码和补码, 下面分别加以介绍。

1. 原码

正数的符号位用 0 表示, 负数的符号位用 1 表示, 其余位表示数值。这种表示方法即为原码表示法。

例如, 在字长为 8 位的微机上表示 +6 和 -6 的原码。

$[X_1]_{\text{原}} = 0$	0000110	$[X_2]_{\text{原}} = 1$	0000110
符号	数值	符号	数值

(1) 小数原码定义

设 X 是纯小数, 用小数点左面一位表示数的符号, 则真值 X 的原码表示为

$$[X]_{\text{原}} = \begin{cases} X & (0 \leq X < 1) \\ 1 - X & (-1 < X \leq 0) \end{cases}$$

例如, 设 $X_1 = 0.1100111$, $X_2 = -0.1100111$, 则 $[X_1]_{\text{原}} = 0.1100111$, $[X_2]_{\text{原}} = 1 - (-0.1100111) = 1.1100111$

(2) 整数原码定义

设机器的字长为 n 位, 最高位代表符号位, 则真值 X 的整数原码表示为

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

例如, 设 $X_1 = +1001000$, $X_2 = -1001000$, 则



$$[X1]_{\text{原}}=01001000 \quad [X2]_{\text{原}}=2^{n-1}-X2=10000000-(-1001000)=11001000$$

对于真值零, 即可以认为它是+0, 也可以认为它是-0, 则

$$[+0]_{\text{原}}=000\cdots 00$$

$$[-0]_{\text{原}}=100\cdots 00$$

因此, 在原码表示中零的表示不是惟一的, 有正零和负零之分。

(3) 原码表示数的范围

原码表示数的范围与机器字长 n 有关。

对于小数情况来讲, 所能表示的范围是

$$-(1-2^{-n-1}) \leq X \leq (1-2^{-n-1})$$

对于整数情况来讲, 所能表示的范围是

$$-(2^{n-1}-1) \leq X \leq (2^{n-1}-1)$$

原码表示简单易懂, 而且与真值的转换方便。但原码表示的数不便于计算机进行运算, 因为在两个原码数运算时, 首先要判断两数的符号, 然后再决定用加法还是用减法。例如, 正数加负数, 要先比较两数绝对值的大小, 用较大数的绝对值减去较小数的绝对值, 作为差的绝对值, 然后取绝对值较大数的符号作为差的符号。

2. 反码

一个数的反码很容易由原码求得。如果是正数, 则其反码与原码相同, 即 $[X]_{\text{原}}=[X]_{\text{反}}$ 如果是负数, 则其反码为除符号位外, 其余各位按位取反, 即 1 转换为 0, 0 转换为 1。

例如, $X1=+6$ $X2=-6$

$$[X1]_{\text{原}}=00000110 \quad [X2]_{\text{原}}=10001110$$

$$[X1]_{\text{反}}=00000110 \quad [X2]_{\text{反}}=11110001$$

又如二进制小数 $+0.1011001$ 和 -0.1011001 用反码表示分别为 0.1011001 和 1.0100110 。

对数值 0, 在反码中也不是惟一的表示, 即

$$[+0]_{\text{反}}=000\cdots 00$$

$$[-0]_{\text{反}}=111\cdots 11$$

反码表示数的范围与原码相同, 此处不再说明。

在计算机中, 反码表示法也较少使用, 实际应用主要用补码表示法。

3. 补码

在计算机中数字采用补码表示。它可以将减法运算转化为加法运算, 从而简化了计算机运算的处理过程。

(1) 模数的概念

以校对时钟为例, 现假定标准的时间是 6 点整, 有一块手表指向 8 点整, 比标准时间快 2 小时, 校准方法可以倒拨 2 小时, 也可以正拨 10 小时, 若规定倒拨是做减法, 则正拨是做加法。因此, 对手表来讲, 减 2 与加 10 是等价的, 即 $10+8=12(\text{丢失})+6=8-2$, 因为手表最大只能表示 12, 大于 12 时, 12 自然丢失, 因此减法运算可以等价于加法运算。这里自动丢失的数“12”称为模, 10 称为 -2 相对于模 12 的补。

(2) 补码的定义

一个数的补码可以由该数的反码求得。如果是正数, 则其补码与原码和反码相同, 即