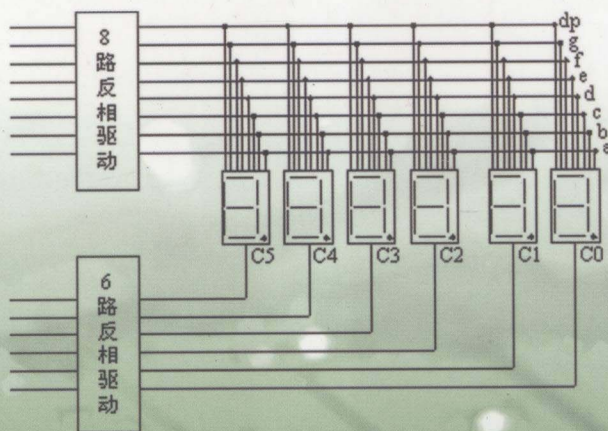


普通高等教育“十一五”电气信息类规划教材

DIANQI  
XINXILEI

# 单片机原理及其 接口技术

严洁 主编



 机械工业出版社  
CHINA MACHINE PRESS

免费  
电子课件



普通高等教育“十一五”电气信息类规划教材

# 单片机原理及其接口技术

主 编 严 洁  
副主编 刘 强 张建国  
参 编 苏博民 董建园  
主 审 申忠如



机械工业出版社

本书是面对非电专业工科学生编写的计算机硬件基础教材，主要读者为机械类、机电类专业学生。本书内容以 51 系列单片机为蓝本，内容涉及计算机基本概念、8051 编程结构、汇编语言设计、系统扩展技术、中断、定时器/计数器、串行接口、人机交互通道接口技术、前向及后向通道接口技术、单片机控制系统系统设计方法及应用实例，最后简单介绍了几种常用单片机。

### 图书在版编目 (CIP) 数据

单片机原理及其接口技术/严洁主编. —北京: 机械工业出版社, 2010. 1  
普通高等教育“十一五”电气信息类规划教材  
ISBN 978-7-111-29022-3

I. 单… II. 严… III. ①单片机—基础理论—高等学校—教材②单片机—接口—高等学校—教材 IV. TP368. 1

中国版本图书馆 CIP 数据核字 (2009) 第 205139 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)  
策划编辑: 刘丽敏 责任编辑: 任正一 版式设计: 霍永明  
封面设计: 张 静 责任校对: 纪 敬 责任印制: 杨 曦  
唐山丰电印务有限公司印刷

2010 年 1 月第 1 版第 1 次印刷

184mm × 260mm · 15 印张 · 368 千字

标准书号: ISBN 978-7-111-29022-3

定价: 26.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010)88361066

门户网: <http://www.cmpbook.com>

销售一部: (010)68326294

教材网: <http://www.cmpedu.com>

销售二部: (010)88379649

读者服务部: (010)68993821

封面无防伪标均为盗版

# 前 言

作为工科学生，在工作中面临的计算机控制系统日益增多，在学习了计算机文化基础、计算机软件基础（C语言）课程之后，有必要接触一点计算机硬件基础。基于非电专业工科学生硬件训练较少的特点，编写了这本适合他们了解计算机硬件基础的教材。

大家都有这样的经验，如果在墙上钉一个钉子，这个钉子必须是尖头的，不能是圆头的。本书提供的就是学习计算机硬件基础这个钉子的尖头。在学习本书的过程中，建议学有余力的同学进行更深入的学习，并且通过计算机+下载电缆+ISP单片机的途径开展实验，把学习计算机硬件基础的钉子逐步钉结实。

本书的一个主要特点是：没有将指令系统单列一章，所有指令分散在程序实例中介绍，使其具有强大的生命力。为照顾系统性，书后附录中给出了指令系统的全部内容，便于习惯集中讲授指令系统的教师选择使用和查找。该思路得益于外语单词的学习方法，在文章中出现的生词容易记忆，便于应用。

汇编语言始终是学习计算机硬件基础的一个瓶颈，为此专门设立8051编程结构一章，重点介绍常用寄存器、存储器的使用方法，并结合编程加深印象，而对其他结构尽量不讲或少讲，突出应用。

本书建议授课学时56，授课内容从第1章到第7章的1、2节。

本书第1章由苏博民编写，第2章、附录由严洁编写，第3章、第5章、第6章由刘强编写，第4章、第7章由张建国编写，第8章、第9章由董建园编写。全书由严洁统稿，申忠如教授主审。

本书编写过程中得到了翟俊祥、席爱民、王建校教授的指导和帮助，席洪波、张阳同学参与了本书的录入和绘图工作。借此机会向上述各位表示由衷的感谢。

限于编者的水平，书中难免存在疏漏及欠妥之处，恳请广大教师和读者指正。

编 者

2008年10月

# 目 录

## 前言

## 第 1 章 单片机的概念及其预备知识 ..... 1

### 1.1 数制、码制系统 ..... 1

#### 1.1.1 进位计数制及各计数制间的转换 ..... 1

#### 1.1.2 码制及其转换 ..... 4

### 1.2 数字电路基础 ..... 8

#### 1.2.1 数字信号 ..... 8

#### 1.2.2 门电路 ..... 8

#### 1.2.3 触发器 ..... 9

#### 1.2.4 常用集成芯片 ..... 10

### 1.3 微型计算机工作原理 ..... 13

#### 1.3.1 CPU 结构 ..... 13

#### 1.3.2 总线结构 ..... 14

#### 1.3.3 指令的执行过程 ..... 15

### 1.4 存储器基础 ..... 16

#### 1.4.1 信息的基本单位 ..... 16

#### 1.4.2 存储器系统结构 ..... 16

#### 1.4.3 存储器的主要性能指标 ..... 17

#### 1.4.4 存储器的分类 ..... 17

### 1.5 I/O 基本概念 ..... 19

#### 1.5.1 I/O 接口电路的作用 ..... 19

#### 1.5.2 I/O 与 CPU 的连接信号 ..... 20

#### 1.5.3 I/O 的编址方案 ..... 20

#### 1.5.4 I/O 的控制方式 ..... 20

### 1.6 单片机概念 ..... 21

#### 1.6.1 单片机概述 ..... 21

#### 1.6.2 8051 单片机基本结构 ..... 23

### 本章小结 ..... 24

### 习题 ..... 24

## 第 2 章 8051 编程结构 ..... 26

### 2.1 累加器和工作寄存器组 ..... 26

#### 2.1.1 寄存器 ..... 26

#### 2.1.2 工作寄存器和工作寄存器组 ..... 28

### 2.2 程序状态字寄存器 ..... 28

#### 2.2.1 程序状态字 ..... 28

#### 2.2.2 条件标志位含义 ..... 29

#### 2.2.3 标志位操作 ..... 30

### 2.3 数据地址指针 ..... 30

#### 2.3.1 数据地址指针 ..... 30

#### 2.3.2 数据地址指针的操作 ..... 31

### 2.4 单片机存储器组织 ..... 31

#### 2.4.1 程序存储器 ..... 32

#### 2.4.2 数据存储器 ..... 35

### 2.5 堆栈和堆栈指针 ..... 39

#### 2.5.1 堆栈和堆栈指针的概念 ..... 39

#### 2.5.2 8051 堆栈和堆栈指针 ..... 40

### 2.6 位处理器和位寻址单元 ..... 42

#### 2.6.1 可位寻址的单元 ..... 42

#### 2.6.2 位存储器操作 ..... 42

#### 2.6.3 位累加器操作 ..... 43

### 本章小结 ..... 44

### 习题 ..... 45

## 第 3 章 汇编语言程序设计 ..... 48

### 3.1 汇编语言源程序的设计和汇编 ..... 48

#### 3.1.1 指令和助记符 ..... 48

#### 3.1.2 伪指令 ..... 49

#### 3.1.3 汇编语言语句格式 ..... 50

#### 3.1.4 汇编语言程序设计步骤 ..... 51

#### 3.1.5 程序流程图技术 ..... 51

### 3.2 程序计数器和程序运行 ..... 52

#### 3.2.1 程序计数器 ..... 52

#### 3.2.2 简单程序运行 ..... 52

#### 3.2.3 循环程序运行 ..... 54

#### 3.2.4 子程序运行 ..... 55

### 3.3 顺序程序设计 ..... 55

### 3.4 分支程序设计 ..... 57

3.4.1 无条件转移指令	57	5.1.3 8051 中断控制	99
3.4.2 条件转移指令	59	5.1.4 8051 中断程序的设计及应用	103
3.5 循环程序设计	62	5.2 定时器/计数器	104
3.5.1 循环结构	62	5.2.1 定时器/计数器概述	104
3.5.2 循环分类	62	5.2.2 定时器/计数器的工作方式	106
3.6 子程序设计	66	5.2.3 定时器/计数器的控制寄存器	107
3.6.1 调用与返回指令	66	5.2.4 定时器/计数器初始化	109
3.6.2 主程序与子程序间的参数传递	67	5.2.5 定时器/计数器应用举例	109
3.6.3 现场保护和现场恢复	68	5.3 串行接口	117
3.7 常用子程序设计	68	5.3.1 串行通信基础知识	117
本章小结	74	5.3.2 8051 串行接口结构	119
习题	74	5.3.3 串行口控制寄存器	120
<b>第4章 存储器及 I/O 扩展</b>	<b>77</b>	5.3.4 波特率的设定	122
4.1 单片机引脚	77	5.3.5 串行工作方式	123
4.2 单片机 I/O 口	80	5.3.6 单片机与 PC 的通信	125
4.2.1 P1 口	80	本章小结	126
4.2.2 P0 口	81	习题	127
4.2.3 P2 口	82	<b>第6章 人机交互通道接口技术</b>	<b>129</b>
4.2.4 P3 口	83	6.1 显示器接口	129
4.3 存储器扩展	84	6.1.1 LED 数码显示器接口	129
4.3.1 存储器芯片	84	6.1.2 专用 LED 驱动器 MAX7219	134
4.3.2 8051 单片机的存储器扩展系统	85	6.1.3 LED 大屏幕显示	134
4.4 I/O 扩展	88	6.1.4 LCD	135
4.4.1 简单并行扩展	88	6.2 键盘接口	138
4.4.2 可编程并行扩展芯片 8255A	88	6.2.1 键盘	138
4.4.3 8255A 的内部结构与引脚	88	6.2.2 键盘的工作原理	139
4.4.4 8255A 的工作方式	91	6.3 BCD 码拨盘接口	142
4.4.5 8255A 的控制字	91	本章小结	143
4.4.6 8255A 与系统的连接	92	习题	144
4.5 非总线型扩展	93	<b>第7章 前向和后向通道接口技术</b>	<b>145</b>
4.5.1 总线型单片机的非总线扩展	93	7.1 测控系统中前向和后向通道的构成	145
4.5.2 非总线型单片机	94	7.2 D/A 转换器	146
4.5.3 串行扩展总线与串行扩展接口	94	7.2.1 基本概念	146
本章小结	96	7.2.2 典型的 D/A 转换器芯片举例	147
习题	96	7.3 A/D 转换器	151
<b>第5章 中断、定时器/计数器、</b>		7.3.1 A/D 转换器工作原理	151
<b>串行接口</b>	<b>98</b>	7.3.2 典型的 A/D 转换器芯片举例	153
5.1 中断	98	7.3.3 ADC 芯片与 CPU 接口	154
5.1.1 中断概述	98	7.4 传感器及信号调理电路	156
5.1.2 8051 中断处理步骤	99	7.4.1 电阻式传感器	156
		7.4.2 电磁式传感器	157

7.4.3 压电式传感器 .....	158	<b>第9章 其他常用单片机简介</b> .....	181
7.4.4 磁电感应式传感器 .....	158	9.1 Atmel 公司 8051 系列兼容机 .....	182
7.4.5 霍尔传感器 .....	159	9.2 Atmel 公司 AVR 单片机 .....	183
7.4.6 光电式传感器 .....	159	9.3 TI 公司 MSP430 单片机 .....	184
7.4.7 信号调理电路 .....	159	9.4 Microchip 公司 PIC 单片机 .....	186
7.5 驱动电路 .....	161	9.5 Silicon Labs 公司 C8051F 单片机 .....	188
本章小结 .....	163	<b>附录</b> .....	189
习题 .....	163	附录 A 8051 指令详表 .....	189
<b>第8章 单片机控制系统设计</b> .....	164	附录 B 8051 指令速查表 .....	214
8.1 单片机应用系统开发过程 .....	164	附录 C 8051 指令分类列表 .....	216
8.2 开发工具和语言的选择 .....	165	附录 D 常用 8051 资源表 .....	220
8.3 应用程序设计原则与方法 .....	167	附录 E ASCII 码字符表 .....	230
8.4 应用举例 .....	169	附录 F 常用 IC 查询网站 .....	231
8.4.1 过零触发双向晶闸管调功器 .....	169	<b>参考文献</b> .....	232
8.4.2 简易数控系统 .....	175		

# 第 1 章 单片机的概念及其预备知识

本章首先简单介绍数制及其转换、码制及其转换的方法，然后总结数字信号、门电路、触发器的基本概念，包括单片机系统中常用集成芯片的使用方法，使读者建立必要的知识体系。最后介绍微型计算机电路的基本工作原理，包括 CPU 结构，指令执行过程，存储器，I/O 接口，单片机发展、特点、应用，单片机硬件结构图，使读者初步认识单片机。

## 1.1 数制、码制系统

### 1.1.1 进位计数制及各计数制间的转换

数制是人们为了计数方便，选择一组数码，并将其按一定位置排列，使相邻两位保持固定进位关系的、用来表示数量大小的方法。日常生活中最常用的数制是十进制。计算机硬件电路由数字电路构成，其电路信号只有两种稳定状态，分别采用二进制数的 0 和 1 来表示。采用数字信号可以很方便地对数据进行处理和存储。在编程时，为了方便记忆和书写，人们还经常用十六进制来表示二进制数，因此可以认为，十六进制数是二进制数的缩写。如某数用二进制表示为：100010010110B，而改用十六进制表示则为：896H，比二进制简单得多。

#### 1. 进位计数制

当进位计数制采用位置表示法时，同一数码处于不同的数位所代表的数值是不同的。如十进制数 888.88，从左到右的五个 8 依次表示的数值为：8 个百、8 个十、8 个一、8 个十分之一以及 8 个百分之一。在使用不同进位计数制时，应注意以下两个基本的因素：

1) 基数  $R$  (Radix)：它代表计数制中所用到的数码个数。如：在十进制计数中用到 0~9 十个数码，基数是 10；二进制计数中用到 0 和 1 两个数码，基数是 2；而十六进制计数中用到 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 共十六个数码，基数是 16。一般地说，基数为  $R$  的数制（简称  $R$  进制）中，包含 0、1、 $\dots$ 、 $R-1$  个数码，进位规律为“逢  $R$  进 1”或“借 1 当  $R$ ”。

2) 位权  $W$  (Weight)：进位计数制中，某个数位的值是由这一位的数码值乘以处在这一位的固定常数决定的，通常把这一固定常数称之为位权值，简称位权。各位的位权是以  $R$  为底的幂。如十进制数基数  $R=10$ ，则个位、十位、百位上的位权分别为  $10^0$ ， $10^1$ ， $10^2$ ，按幂称个位、十位、百位分别为 10 的 0 权位、1 权位、2 权位。

一个  $R$  进制数  $N$ ，可以用以下两种形式表示：

1) 并列表示法，或称位置计数法

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}K_{-2}\cdots K_{-m})_R \quad (1-1)$$

2) 多项式表示法，或称按权展开式

$$(N)_R = K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_1R^1 + K_0R^0 + K_{-1}R^{-1} + \cdots + K_{-m}R^{-m} = \sum_{i=n-1}^{-m} K_i R^i \quad (1-2)$$



式中， $m$ 、 $n$  为正整数， $n$  代表整数部分的位数； $m$  代表小数部分的位数； $K_i$  代表  $R$  进制中的任一个数码， $0 \leq K_i \leq R - 1$ 。

表 1-1 给出了以上三种数制的对应关系。为避免混淆，除用  $(N)_R$  的方法区分不同进制数外，还常用数字后加字母作为标注。字母 D (Decimal) 或不加字母表示十进制数；字母 B (Binary) 表示二进制数；字母 H (Hexadecimal) 表示十六进制数。

表 1-1 十、二、十六进制数码对应表

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

### 2. 三种进制数间的相互转换

由于各种数制被用于不同的场合，因此它们之间需要经常进行转换。如输入输出时使用人们习惯的十进制数，而计算机处理和存储数据时采用二进制数，在了解各种数制的物理基础之后，它们之间的转换也就比较容易掌握了。

#### (1) 二进制数转换成十进制数

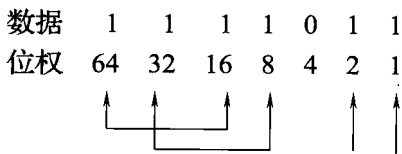
二进制数转换成十进制数的方法是：先按权展成多项式，再利用十进制运算法则求和，即可得到该数对应的十进制数。为叙述方便，本书各种数制全部采用定点整数表示。

**例 1-1** 将 1001101B 转换为十进制数。

$$1001101B = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 64 + 8 + 4 + 1 = 77$$

在实际应用中，由于二进制数只有 0 和 1 两个数码，因此按权展开式不需进行乘法运算，只需直接将 1 数位上的位权值相加即可。

**例 1-2** 将数 1111011B 转换为十进制数。



转换结果  $80 + 40 + 2 + 1 = 123$

加法尽量按补数分组进行。另外，例 1-2 中为 1 的数位多，为 0 的数位少，根据此特点，还可以使用减法加速转换过程。例中可考虑 7 个数位全 1 时转换结果为 127，该数据 2 权位数码为 0，从 127 中减去对应位权值 4 即可得 123。

#### (2) 十进制数转换为二进制数

任一十进制整数  $N$  转换成二进制数的方法是连续除以 2 倒取余。

**例 1-3** 将十进制数 118 转换成二进制数。

解：转换过程如右式所示。

简单地，还可以利用权值进行转换。将比待转换数据小的所有位权按顺序排列，从高位到低位依次累加，根据每次累加和是否小于待转换数据，决定该位是否为 1。

$$\begin{array}{r}
 2 \overline{)118} \\
 \underline{2 \overline{)59}} \quad \text{余数 } 0, K_0=0 \\
 2 \overline{)29} \quad \text{余数 } 1, K_1=1 \\
 2 \overline{)14} \quad \text{余数 } 1, K_2=1 \\
 2 \overline{)7} \quad \text{余数 } 0, K_3=0 \\
 2 \overline{)3} \quad \text{余数 } 1, K_4=1 \\
 2 \overline{)1} \quad \text{余数 } 1, K_5=1 \\
 1 \quad \text{余数 } 1, K_6=1
 \end{array}$$

位权	64	32	16	8	4	2	1
数据	1	1	1	0	1	1	0

其结果为  $118D = 1110110B$

### (3) 二进制数与十六进制数之间的相互转换

从个位开始向左每四位为一组按 8421 权进行转换, 若最左(高位)一组不足四位添 0 补足四位, 便可将二进制数转换为十六进制数。

**例 1-4** 将  $1101000B$  转换成十六进制数。

**解:** 根据“合 4 为 1”的原则, 可将该二进制数分组后, 每组按 8421 权展开为:

0110	1000	B
↑	↑	
6	8	H

其结果为  $1101000B = 68H$ 。

反之, 采用“1 分为 4”的原则, 每 1 位十六进制数用 4 位二进制数表示, 便可将十六进制数转换为二进制数。

**例 1-5** 将  $4DH$  转换成二进制数。

**解:** 根据“1 分为 4”的原则, 可将该十六进制数书写为:

4	D	H
↓	↓	
0100	1101	B

其结果为  $4DH = 1001101B$ 。

### (4) 十进制数与十六进制数之间的相互转换

由于十六进制数是二进制数的缩写, 二进制数与十进制数转换又较为方便, 故十六进制数与十进制数之间的转换可通过二进制数进行。

**例 1-6** 将  $ABH$  转换成十进制数。

**解:** 先将  $ABH$  转换成二进制数,

A	B	H
1010	1011	B

再将二进制数转换成十进制数

位权	128	64	32	16	8	4	2	1
数据	1	0	1	0	1	0	1	1
和	160		+	10		+	1	=171

## 3. 二进制数的算术运算

二进制数的加、减法遵循“逢 2 进 1”、“借 1 当 2”的原则。

1 位二进制数加法规则为

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10 \quad (\text{有进位}) \quad (1-3)$$

1 位二进制数减法规则为

$$1-0=1 \quad 1-1=0 \quad 0-0=0 \quad 0-1=1 \quad (\text{有借位}) \quad (1-4)$$

1 位二进制乘法规则为

$$0 \times 0 = 0 \quad 0 \times 1 = 0 \quad 1 \times 0 = 0 \quad 1 \times 1 = 1 \quad (1-5)$$

作为二进制数缩写的十六进制数，其加法按二进制进行，按十六进制书写结果。

例 1-7 求 D9H + BEH 的和。

$$\begin{array}{r} \text{D9H} = 1101 \quad 1001 \quad \text{B} \\ \text{BEH} = 1011 \quad 1110 \quad \text{B} \\ \quad \quad 1101 \quad 1001 \quad \text{B} \\ + \quad 1011 \quad 1110 \quad \text{B} \\ \hline 11001 \quad 0111 \quad \text{B} \end{array}$$

其结果为 D9H + BEH = 197H

### 1.1.2 码制及其转换

由于计算机只能识别 0 和 1，因此，凡是需要计算机处理的事物都需要用由 0 和 1 组成的代码表示，称为编码，如字符、声音、颜色等。计算机常用的编码有表示带符号数的原码、补码与反码，用二进制数表示十进制数的 BCD 码以及表示字符的 ASCII 码。

#### 1. 原码、反码与补码

##### (1) 原码

前面所涉及的二进制数都是无符号数，其特点是：所有数位均为表示二进制数大小的数值位，8 位二进制数所表示的十进制数的范围从 0 ~ 255。对于带符号二进制数而言，在计算机中通常将其最高位作为符号位，用 0 表示“+”，用 1 表示“-”，其余位作为数值位，称为带符号数的原码，或称机器数，用  $[X]_{\text{原}}$  表示。而用正负号表示的带符号数称为  $[X]_{\text{原}}$  的真值。注意：只有原码与真值之间具有转换关系。

例 1-8 将二进制数真值  $N_1 = +101011\text{B}$ ， $N_2 = -101011\text{B}$ ，分别用 8 位原码表示。

数位编号	D7	D6	D5	D4	D3	D2	D1	D0	
$[N_1]_{\text{原}}$	= 0	0	1	0	1	0	1	1	B
$[N_2]_{\text{原}}$	= 1	0	1	0	1	0	1	1	B
	↑	↑—————↑						↑	
	符号位		数值位						

若数为零，则在 8 位二进制数中，原码有两种表示法：

$$[+0]_{\text{原}} = 00000000\text{B}$$

$$[-0]_{\text{原}} = 10000000\text{B}$$

在原码中，由于符号位占据了最高位，故 8 位二进制数所表示的数的范围从 -127 ~ +127。

##### (2) 反码

正数的反码与其原码相同。

零的反码有两种表示法：

$$[+0]_{\text{反}} = 00000000\text{B}$$

$$[-0]_{\text{反}} = 11111111\text{B}$$

保持负数原码的符号位不变，数值位逐位取反（即 0 变 1，1 变 0），得负数的反码。

反码所表示的8位二进制数的范围为 $-127 \sim +127$ 。对反码可以用同样的方法求出其对应的原码，从而得到该原码的真值。

### (3) 补码

正数的补码与其原码相同。

零的补码为全零。

负数的补码在其反码最低位加1得出。

只有 $-128$ 的补码例外，需专门记忆。对8位二进制数而言：

$$[-128]_{\text{补}} = 10000000\text{B}$$

补码所表示的8位二进制数的范围从 $-128 \sim +127$ 。真值需将结果先还原为原码，或利用互补的特点，继续求补，得到补码对应的原码，进而计算其真值。

微型机只有加法器，没有专门的减法器，当进行减法时，对减数采用补码进行加法运算。

**例 1-9** 设 $X_1 = +63$ ， $X_2 = -63$ ，用8位字长表示 $X_1$ 、 $X_2$ 的原码、反码及补码。

**解：**先将63转换为二进制数，利用数据特点，进行如下转换

$$63 = 64 - 1 = 2^6 - 1 = 1000000 - 1 = 1111111\text{B}$$

得二进制数 $X_1 = +1111111\text{B}$ ， $X_2 = -1111111\text{B}$

$$[X_1]_{\text{原}} = [X_1]_{\text{反}} = [X_1]_{\text{补}} = 00111111\text{B}$$

$$[X_2]_{\text{原}} = 10111111\text{B}, [X_2]_{\text{反}} = 11000000\text{B}, [X_2]_{\text{补}} = 11000001\text{B}$$

#### 1) 补码的运算规则

$$[X \pm Y]_{\text{补}} = [X]_{\text{补}} + [\pm Y]_{\text{补}}$$

**注意：**运算结果为补码， $[X \pm Y]_{\text{补}}$ 的真值需将结果还原为原码或继续求补得到。

**例 1-10** 设 $X = +36$ ， $Y = -63$ ，用补码求 $X + Y$ 。

**解：** $[X]_{\text{补}} = 00100100\text{B}$ ， $[Y]_{\text{补}} = 11000001\text{B}$

$$[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 00100100 + 11000001 = 11100101\text{B}$$

$$[X + Y]_{\text{原}} = [ [X + Y]_{\text{补}} ]_{\text{补}} = 10011011\text{B}$$

$$X + Y = -27$$

**例 1-11** 设 $X = +36$ ， $Y = -63$ ，用补码求 $Y - X$ 。

**解：** $[-X]_{\text{补}} = 11011100\text{B}$ ，

$$[Y - X]_{\text{补}} = [Y]_{\text{补}} + [-X]_{\text{补}} = 11000001 + 11011100 = 1\ 10011101\text{B}$$

↑  
第9位自然丢失

$$[Y - X]_{\text{原}} = [ [Y - X]_{\text{补}} ]_{\text{补}} = 11100011\text{B}$$

$$Y - X = -99$$

**例 1-12** 设 $X = -99$ ， $Y = -63$ ，用补码运算求 $X + Y$ 。

**解：** $[X]_{\text{补}} = 10011101\text{B}$ ， $[Y]_{\text{补}} = 11000001\text{B}$

$$[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 10011101 + 11000001 = 1\ 01011110\text{B}$$

↑  
溢出错误

例 1-11 中，由于结果 $-99$ 在补码所能表示的8位二进制数的范围 $-128 \sim +127$ 内，故最高位属于自然丢失。本例结果 $-162$ 超出补码所能表示的8位二进制数的范围，故出现溢

出错误，可以观察到符号位变为0，这显然是错误的。当一个数超出计算机所能表示的数值的最大范围时，将会出现溢出，溢出是一种错误，必须及时发现并处理，由此引发了计算机的溢出判别问题。

2) 溢出判别 为了区分自然丢失和溢出错误，计算机中通常利用符号位的进位  $C_S$  及最高数值位的进位  $C_P$  的取值来判断是否发生了溢出，称双高位判别法。计算过程中  $C_S$  和  $C_P$  有进位为1，无进位为0。溢出判别符记为  $OV$  (Overflow)，定义为

$$OV = C_S \oplus C_P$$

$OV = 1$ ，有溢出， $OV = 0$ ，无溢出。

**例 1-13** 对例 1-11、例 1-12 采用双高位判别法进行溢出判别。

<p>解：</p> $\begin{array}{r} 11000001 \quad [-63]_{\text{补}} \\ + 11011100 \quad [-36]_{\text{补}} \\ \hline 110011101 \\ \uparrow \\ \text{进位} \end{array}$	$\begin{array}{r} 10011101 \quad [-99]_{\text{补}} \\ + 11000001 \quad [-63]_{\text{补}} \\ \hline 101011110 \\ \uparrow \\ \text{进位} \end{array}$
$C_S = 1, C_P = 1, OV = 0$ ，无溢出	$C_S = 1, C_P = 0, OV = 1$ ，有溢出

结论是与例 1-11、例 1-12 完全相同的。

## 2. BCD 码 (Binary Coded Decimal)

BCD 码是用二进制直接表示十进制数。1 位十进制数有 0~9 共 10 个不同数码，故需要由 4 位二进制数来编码。4 位二进制数有 16 种组合，可任取其中 10 种组合分别代表 10 个十进制数码，其余 6 种组合为非法码。常用 8421BCD 码，取十六进制数的前 10 种组合作为 10 个十进制数码，而 A、B、C、D、E、F 视为非法码，不允许出现。二进制数与 BCD 码之间的转换方法和二进制数与十六进制数之间的转换方法相同。8421BCD 码为有权码，还有无权 BCD 码，如格雷码。

**例 1-14** 78 转换成相应的 8421BCD 码。

解：采用“1 分为 4”的原则，每位十进制数用 4 位二进制数表示如下：

$$\begin{array}{cc} 7 & 8 \\ \swarrow \searrow & \swarrow \searrow \\ 0111 & 1000 \end{array}$$

其结果  $78 = (111\ 1000)_{8421\text{BCD}}$

注意 BCD 码不是二进制数，BCD 码转换为二进制数时应先转换为十进制数，反之亦然。

一位 BCD 码采用 4 位二进制数组成，高 4 位补 0000，构成一个字节的数据，称为非压缩 BCD 码，两位 BCD 码构成一个字节的数据时，称为压缩 BCD 码。前者常用来显示，后者常用来存储和处理。

BCD 码实质上是十进制数，按“逢 10 进 1”或“借 1 当 10”的规律进行运算，而计算机是二进制，如 4 位二进制数一组，则按“逢 16 进 1”或“借 1 当 16”的规律进行运算，因此在运算过程中可能出现错误，需要调整。调整规律如下：

1) 当两个一位 BCD 码相加结果大于 9 时，计算机不会进位，需对本位和加 6 修正，使其产生进位。修正后再次产生的非法码仍需修正。

2) 当两个一位 BCD 码相加结果有进位发生时，计算机是“逢 16 进 1”，因多向高位进了一个 6，故也需对本位和加 6 修正。修正后再次产生的进位不需修正。在计算机中专门设

置了进位位 CY 和辅助进位位 AC 两个标志, 以提供一个压缩 BCD 码中高半字节和低半字节的进位情况。

整个加 6 修正的工作是计算机自动完成的, 使用者只需在 BCD 码运算时, 使用专门的调整指令即可。由于调整指令应用范围较窄, 实际中二进制数处理指令是大量的。

**例 1-15** 用 BCD 码完成  $54 + 48$  的运算。

解:  $54 = (0101\ 0100)_{8421BCD}$ ,  $48 = (0100\ 1000)_{8421BCD}$

$$\begin{array}{r}
 0101\ 0100 \\
 +\ 0100\ 1000 \\
 \hline
 1001\ 1100 \quad (\text{低 4 位和 } 0CH, >9) \\
 +\ \quad\quad 0110 \quad (\text{低 4 位加 6 修正}) \\
 \hline
 1010\ 0010 \quad (\text{高 4 位和 } 0AH, >9) \\
 +\ 0110 \quad (\text{高 4 位加 6 修正}) \\
 \hline
 1\ 0000\ 0010
 \end{array}$$

其结果  $(0000\ 0010)_{8421BCD} = 02$ ,  $CY = 1$ 。  $54 + 48 = 102$ 。

### 3. ASCII 码 (American Standard Code for Information Interchange)

ASCII 码包括 10 个十进制数码 0~9, ASCII 码为 30H~39H (数码前加 3 得对应 ASCII 码, 转换方便); 大写和小写英文字母各 26 个, 大写字母 A~Z 的 ASCII 码为 41H~5AH; 小写字母 a~z 的 ASCII 码为 61H~7AH (大小写转换只需改变左起第 3 位, 如大写 A 为 01000001, 小写 a 为 01100001, 等); 32 个通用控制符号; 34 个专用符号, 共 128 个字符, 故采用 7 位二进制数进行编码。具体如表 1-2 所示。

表 1-2 ASCII 码表

高位 \ 低位		0	1	2	3	4	5	6	7
		00	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	,	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DE2	"	2	B	R	b	r
3	0011	ETX	DE3	#	3	C	S	c	s
4	0100	EOT	DE4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	—	=	M	]	m	}
E	1110	SO	RS	。	>	N	↑	n	~
F	1111	SI	US	/	?	P	←	o	DEL

## 1.2 数字电路基础

### 1.2.1 数字信号

数字信号是一种脉冲信号，多数数字信号在两个固定电流或电压值之间转换，这正好表示二进制数的0和1。

### 1.2.2 门电路

门电路是构成数字电路的基本单元，复杂的数字电路均由基本门电路组合而成。基本逻辑门的电路符号和逻辑表达式如图1-1所示，它们是具有一个或多个输入信号和一个输出信号 $Q$ 的简单电路，其逻辑表达式位于对应图形符号的下方。实际产品也有多输入的情况。门电路还可以级联，构成复杂的逻辑关系。

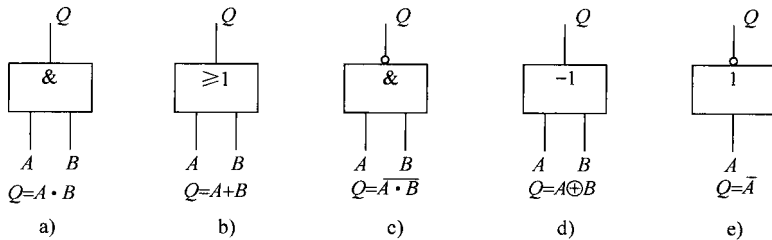


图1-1 基本门电路符号

a) 与门 b) 或门 c) 与非门 d) 异或门 e) 非门

#### 1. 逻辑与 (AND)

逻辑与运算符为 $\cdot$ 或 $\wedge$ 。运算规则如下：

$$0 \cdot 0 = 0 \quad 0 \cdot 1 = 1 \cdot 0 = 0 \quad 1 \cdot 1 = 1 \quad (1-6)$$

**例 1-16** 若逻辑序列  $X = 10101110B$ ,  $Y = 00001111B$ , 求  $X \cdot Y$ 。

解： 10101110

$$\begin{array}{r} \wedge 00001111 \\ \hline 00001110 \end{array}$$

即  $X \cdot Y = 00001110B$ 。

#### 2. 逻辑或 (OR)

逻辑或运算符为 $+$ 或 $\vee$ 。运算规则如下：

$$0 + 0 = 0 \quad 0 + 1 = 1 + 0 = 1 \quad 1 + 1 = 1 \quad (1-7)$$

**例 1-17** 若逻辑序列  $X = 1101110B$ ,  $Y = 10000000B$ , 求  $X + Y$ 。

解： 01101110

$$\begin{array}{r} \vee 10000000 \\ \hline 11101110 \end{array}$$

即  $X + Y = 11101110B$ 。

#### 3. 逻辑非 (NOT)

逻辑非又称逻辑反，运算规则如下：

$$\bar{1} = 0 \quad \bar{0} = 1 \tag{1-8}$$

例 1-18 若逻辑序列 A = 10101111B，求  $\bar{A}$ 。

解： $\bar{A} = 01010000B$

#### 4. 异或运算 (XOR)

异或运算的运算符为  $\oplus$ ，其运算规则如下：

$$0 \oplus 0 = 0 \quad 0 \oplus 1 = 1 \oplus 0 = 1 \quad 1 \oplus 1 = 0 \tag{1-9}$$

例 1-19 若逻辑序列 X = 10101111B，Y = 01011110B，求  $X \oplus Y$ 。

解： 10101111

$$\oplus \begin{array}{r} 01011110 \\ \hline 11110001 \end{array}$$

即  $X \oplus Y = 11110001B$ 。

另一种在计算机中广泛应用的门电路是三态门，如图 1-2 所示。它的输出  $b$  端除 0 和 1 两种状态外，还有第三种状态——高阻态。当输入  $a$  端出现数据时，只有使能端 EN 信号  $c = 1$ ，数据才会传送至  $b$  端，若使能端 EN 信号  $c = 0$ ，则  $b$  端为高阻态。三态门广泛用于隔离或放大。

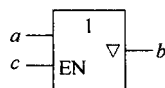


图 1-2 三态门

计算机各部件采用总线结构传递数据，总线是各部件的公共信息线，为防止总线冲突，每一时刻只能有一组信息在总线中出现。所有部件与总线采用三态门连接，当一个部件的使能信号有效时，令其他部件的使能信号均无效，与总线脱离，这就保证了数据不会发生冲突。

三态门还用于输入缓冲器。当现场数据混乱时，令使能信号  $EN = 0$ ，将计算机与输入信号隔离。待现场数据稳定后，再令  $EN = 1$ ，计算机可进行输入工作。图 1-2 为高电平有效的使能信号，也有低电平有效的使能信号，使用时应注意区分。

### 1.2.3 触发器

触发器是一种具有记忆功能的基本逻辑单元，触发器输出端  $Q$  的取值为其状态。它有两种稳定的状态：一种状态存储二进制数 0；另一种状态存储二进制数 1。两种状态在一定条件下可以转换。

按逻辑功能区分，有三种基本的触发器：RS 触发器、D 触发器和 JK 触发器，其电路符号如图 1-3 所示。本书介绍前两种。

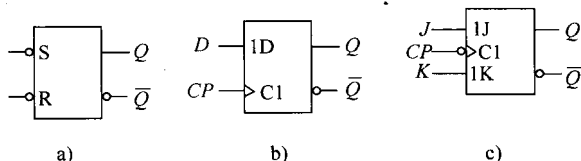


图 1-3 触发器电路符号

a) RS 触发器 b) D 触发器 c) JK 触发器

#### 1. RS 触发器

如图 1-3a 所示，输入 S 称置位端 (SET)，低电平时状态  $Q$  为 1；R 称复位端 (RESET)，低电平时状态  $Q$  为 0；两输入同时为 1 时，状态  $Q$  不变，记忆。



在数字系统设计中, RS 触发器可作为开关去抖电路。它同时也是其他触发器的基本组成单元。

## 2. D 触发器

数据输入  $D$  在时钟脉冲  $CP$  (Clock Pulse) 上沿(由 0 变 1)作用下改变其状态  $Q$ 。注意图 1-4a 中时钟端  $C1$  的三角符号, 它表示在时钟上升沿触发器改变状态。状态取值与时钟脉冲到来前的数据输入  $D$  相同。时钟脉冲  $CP$  其余时间触发器状态不变。

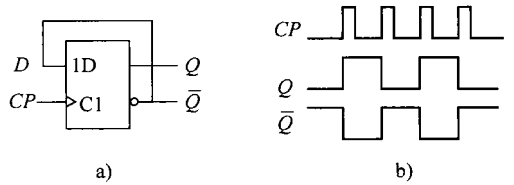


图 1-4 一位二进制计数器

$D$  触发器通常作为寄存器的一个位, 用来接收并存储数据。它还可以构成计数器, 累加  $CP$  的个数, 其结构如图 1-4a 所示。分析图 1-4b 的波形, 设触发器初始状态为 0, 第一个  $CP$  上沿状态变为 1, 第二个  $CP$  上沿状态回零, “逢 2 进 1”, 向高位进位。一个  $D$  触发器构成一位二进制计数器,  $n$  个  $D$  触发器级联, 可构成  $n$  位二进制计数器, 使其计数模达到  $2^n$ 。

我们观察图 1-4b 的波形, 还可以发现  $Q$  的频率是  $CP$  的一半, 因此计数器还被用来构成分频器, 降低系统频率。计数器的模就是分频系数, 二进制计数器计数的模是 2, 分频系数为 2, 构成二分频器。

图 1-5 所示为一位  $D$  锁存器, 其逻辑功能同  $D$  触发器, 仅时钟  $C1$  端没有三角符号, 表示在时钟为 1 时接收数据  $D$ , 时钟为 0 时数据不变, 用于总线输出部件锁存总线数据。

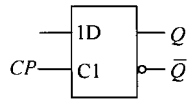


图 1-5 锁存器

## 1.2.4 常用集成芯片

制造商已开发出很多数字集成电路, 可用其构建数字系统。同一系列中的集成电路相互兼容, 彼此可以直接连接。

### 1. 分类

数字集成电路分为双极型和单极型两种。使用双极型技术的是 TTL (晶体管—晶体管逻辑), 其型号有 74 和 54 两大系列; 使用单极型技术的是 CMOS (互补—金属—氧化物半导体), 其型号有 74HC 和 54HC 两大系列。

制造商有时也根据电路的复杂程度对集成电路进行分类:

#### (1) SSI (小规模集成电路)

门电路数: 小于 12;

典型数字器件: 逻辑门和触发器。

#### (2) MSI (中规模集成电路)

门电路数: 12 ~ 99;

典型数字器件: 加法器、计数器、译码器、编码器、数据选择器和多路分配器、寄存器。

#### (3) LSI (大规模集成电路)

门电路数: 100 ~ 9999;

典型数字器件: 数字时钟、小容量存储芯片、计算器。

#### (4) VLSI (超大规模集成电路)