



普通高等教育“十一五”国家级规划教材

数字电路 及 系统设计

赵曙光 刘玉英 崔葛瑾 编著



高等教育出版社
HIGHER EDUCATION PRESS



普通高等教育“十一五”国家级规划教材

数字电路及系统设计

SHUZI DIANLU JI XITONG SHEJI

赵曙光 刘玉英 崔葛瑾 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容简介

本书是普通高等教育“十一五”国家级规划教材。本书共10章,主要内容包括:数制与码制,逻辑代数基础,逻辑门电路,组合逻辑电路,触发器,时序逻辑电路,脉冲波形的产生和整形,半导体存储器,数-模转换和模-数转换,现代数字系统设计与实现方法(具体包括可编程逻辑器件、电子设计自动化、硬件描述语言、现代数字系统的设计方法与流程等)。

本书以培养分析、设计数字电子系统的能力为核心,力求做到内容全面、深度适中、注重基础、兼顾实用、结构合理、组合灵活;既利用大量的精选例题、习题,帮助读者打牢基础,又提供许多源于科研实践、具有实用价值的设计实例,帮助读者扩展视野,提高应用、创新能力。

本书可作为高等学校电气信息类各专业本科、专科“数字电路”类课程的教材和参考书,亦可供其他专业师生及相关工程技术人员选用和参考。

图书在版编目(CIP)数据

数字电路及系统设计/赵曙光,刘玉英,崔葛瑾编著. —北京:高等教育出版社,2011.6

ISBN 978-7-04-032219-4

I. ①数… II. ①赵… ②刘… ③崔… III. ①数字电路-电路设计-高等学校-教材 IV. ①TN79

中国版本图书馆CIP数据核字(2011)第083999号

策划编辑 吴陈滨	责任编辑 曲文利	封面设计 李卫青	版式设计 马敬茹
插图绘制 尹 莉	责任校对 胡晓琪	责任印制 刘思涵	

出版发行 高等教育出版社	咨询电话 400-810-0598
社 址 北京市西城区德外大街4号	网 址 http://www.hep.edu.cn
邮政编码 100120	http://www.hep.com.cn
印 刷 北京明月印务有限责任公司	网上订购 http://www.landaco.com
开 本 787×1092 1/16	http://www.landaco.com.cn
印 张 18.5	版 次 2011年6月第1版
字 数 450 000	印 次 2011年6月第1次印刷
购书热线 010-58581118	定 价 27.20元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 32219-00

前 言

本书是根据教育部高等学校电子电气基础课程教学指导分委员会制定的“数字电路与逻辑设计”课程教学基本要求,结合我们多年的教学、科研经验编写而成。

“数字电路与逻辑设计”课程是高等院校电子信息类专业、计算机专业及其他相关专业的专业技术基础课程。它对于培养学生分析、解决问题特别是分析、设计电子系统的能力,起着承前启后的重要作用。但由于数字电子技术发展迅速,新的方法、技术和成果不断涌现,如何在教材和教学中兼顾基础性和先进性、实用性,是需要不断研究和解决的问题。

本书作者力图在精炼经典性内容的同时,恰当地引入反映学科发展趋势和成果的新颖内容,并采用与之相适应的教材结构体系,努力兼顾基础性与先进性、知识传授与能力培养;恰当地处理理论教学与实践教学,以及该课程与相关课程的配合、衔接关系。

本书共分 10 章。第 1、2 章介绍了数制、编码和逻辑代数等基础知识。第 3、5 章分别介绍了逻辑门、触发器等基本的逻辑单元,着重介绍了原理、功能和用法,对内部电路的分析、计算做了适当精简。第 4、6 章分别介绍了组合逻辑电路、时序逻辑电路的分析方法和设计方法。这部分内容是“数字电路与逻辑设计”课程的教学重点和难点。本书为了兼顾基础和应用,既对传统的逻辑电路分析、设计方法做了简要介绍,又重点介绍了常用的 MSI 器件和以其为基础的逻辑电路分析、设计方法;为加深理解和贴近实际,还选配了较多的例题和习题,其中有些直接来自于科研实践。第 7 章简要介绍了脉冲信号的产生与整形。第 8 章介绍了半导体存储器的原理和使用。第 9 章介绍了数-模(D/A)转换和模-数(A/D)转换的原理和应用。第 10 章简要介绍了现代数字系统的设计、实现方法,具体内容包括:可编程逻辑器件的原理和开发简介,电子设计自动化概述,硬件描述语言 VHDL 和 Verilog-HDL 的简介和实例,数字系统设计的方法和流程等。

本书第 1~4 章由刘玉英、赵曙光编写,第 5~9 章由崔葛瑾、赵曙光编写,第 10 章由赵曙光编写,全书由赵曙光策划和统稿。

本书力求做到内容全面、深度适中、注重基础、兼顾实用、结构合理、组合灵活,可作为“数字电路”类课程的教材和参考书,以及工程技术人员的自学读物和参考书。

本书作为普通高等教育“十一五”国家级规划教材,在编写过程中得到了高等教育出版社的大力帮助,以及作者的亲友、领导和同事所给予的充分理解和支持,在此一并表示衷心感谢。书中参考和引用了许多专家、学者的著作和研究成果,特此说明并表示衷心感谢!

同济大学张申科教授审阅了全部书稿并提出了许多宝贵意见,特此表示衷心感谢!

受作者的水平、能力等限制,本书难免存在疏漏和不足,欢迎广大读者和同行批评指正。

作者谨识

2011 年 2 月

目 录

第 1 章 数制与码制	1
1.1 数制	1
1.1.1 计数体制	1
1.1.2 不同数制之间的转换	3
1.2 编码	5
1.2.1 二 - 十进制编码	5
1.2.2 可靠性编码	6
1.2.3 字符编码	8
1.3 二进制数的算术运算	9
1.3.1 二进制算术运算的特点	9
1.3.2 原码、反码、补码和补码运算	10
本章习题	14
第 2 章 逻辑代数基础	15
2.1 基本运算、公式和定理	15
2.1.1 基本逻辑运算	15
2.1.2 公式和定理	17
2.2 逻辑函数的表示方法	19
2.2.1 真值表	19
2.2.2 逻辑函数表达式	20
2.2.3 卡诺图	23
2.2.4 逻辑图	25
2.3 逻辑函数的化简方法	26
2.3.1 化简的意义和最简的概念	26
2.3.2 公式法化简	27
2.3.3 图形法化简	28
2.3.4 具有无关项的逻辑函数化简	33
本章习题	35
第 3 章 逻辑门电路	39
3.1 概述	39
3.1.1 逻辑门电路的基本概念和原理	39
3.1.2 数字集成电路的分类和特点	40

3.2 半导体元件的开关特性	41
3.3 TTL 门电路	44
3.3.1 TTL 与非门	44
3.3.2 其他类型的 TTL 门电路	51
3.3.3 TTL 器件的选择和使用要点	57
3.4 CMOS 门电路	59
3.4.1 CMOS 基本门电路	60
3.4.2 其他类型的 CMOS 门电路	61
3.4.3 CMOS 器件的选择和使用要点	64
本章习题	65
第 4 章 组合逻辑电路	70
4.1 概述	70
4.2 组合逻辑电路的分析方法	71
4.3 常用组合逻辑电路与器件	72
4.3.1 编码器	72
4.3.2 译码器	76
4.3.3 数据选择器	82
4.3.4 加法器	85
4.3.5 数值比较器	89
4.4 组合逻辑电路的设计方法	91
4.4.1 用 SSI 器件设计组合逻辑电路举例	91
4.4.2 用 MSI 器件设计组合逻辑电路举例	93
4.5 组合逻辑电路中的竞争冒险	96
4.5.1 竞争冒险的产生原因和分类	97
4.5.2 冒险现象的判别	98
4.5.3 消除竞争冒险的方法	99
本章习题	101
第 5 章 触发器	104
5.1 概述	104
5.2 RS 触发器的基本特性和电路结构	105
5.2.1 基本 RS 触发器	105
5.2.2 同步 RS 触发器	108
5.2.3 主从 RS 触发器	109
5.2.4 触发器的时间参数	111
5.3 D 触发器	112
5.4 JK 触发器	115
5.5 T 触发器和 T' 触发器	119

5.6 常用集成触发器及其激励功能转换	120
本章习题	121
第6章 时序逻辑电路	126
6.1 概述	126
6.1.1 时序逻辑电路的基本概念	126
6.1.2 时序逻辑电路的分类	127
6.1.3 常用的时序逻辑电路模块	127
6.2 时序逻辑电路的基本结构和描述方法	131
6.3 时序逻辑电路的一般分析方法	133
6.4 同步时序逻辑电路的一般设计方法	138
6.4.1 同步时序逻辑电路的设计	138
6.4.2 时序逻辑电路的 ASM 图描述	142
6.4.3 状态化简的一般方法	145
6.4.4 状态编码分配的一般规则	146
6.5 常用时序逻辑器件及其应用	149
6.5.1 基于 MSI 的时序电路分析和设计方法	149
6.5.2 集成计数器及其应用	150
6.5.3 集成移位寄存器及其应用	156
6.5.4 集成寄存器和集成锁存器	161
本章习题	162
第7章 脉冲波形的产生和整形	171
7.1 概述	171
7.2 多谐振荡器	172
7.2.1 CMOS 逻辑门组成的多谐振荡器	172
7.2.2 555 定时器构成的多谐振荡器	174
7.2.3 石英晶体振荡器	178
7.3 单稳态触发器	179
7.3.1 CMOS 微分型单稳态触发器	179
7.3.2 555 定时器构成的单稳态触发器	181
7.3.3 可重复触发的单稳态触发器	182
7.3.4 集成单稳态触发器	183
7.3.5 单稳态触发器的应用	184
7.4 施密特触发器	186
7.4.1 用门电路组成的施密特触发器	186
7.4.2 常用的集成施密特触发器	187
7.4.3 施密特触发器的应用	187
本章习题	189

第 8 章 半导体存储器	194
8.1 概述	194
8.1.1 分类与特点	194
8.1.2 主要性能指标	195
8.2 只读存储器	196
8.2.1 只读存储器的结构	196
8.2.2 只读存储器的编程及分类	197
8.3 随机存储器	200
8.3.1 随机存储器的基本结构	200
8.3.2 随机存储器的主要类型	202
8.3.3 静态随机存储器	203
8.3.4 动态随机存储器	204
8.4 存储器容量的扩展	205
8.5 利用存储器实现逻辑函数	207
本章习题	209
第 9 章 数 - 模转换和模 - 数转换	210
9.1 数 - 模转换	210
9.1.1 数 - 模转换器的基本原理	210
9.1.2 常用的数 - 模转换技术	211
9.1.3 数 - 模转换器的主要技术参数	215
9.1.4 集成 DAC 的选用	218
9.1.5 数 - 模转换器输出极性的扩展	219
9.1.6 数 - 模转换器的典型应用	221
9.2 模 - 数转换	223
9.2.1 模 - 数转换器的主要参数	223
9.2.2 常用的模 - 数转换技术	225
9.2.3 集成 ADC 的选用	237
9.2.4 采样 - 保持器的原理和指标	237
本章习题	240
第 10 章 现代数字系统的设计与实现方法	247
10.1 现代数字系统的主要构件——可编程逻辑器件	247
10.1.1 可编程器件概述	247
10.1.2 可编程逻辑器件的分类与特点	250
10.1.3 CPLD 的典型结构和原理	254
10.1.4 FPGA 的典型结构和原理	259
10.1.5 可编程逻辑器件的开发流程	265
10.2 现代数字系统的设计手段——电子设计自动化	267

10.3 现代数字系统的描述工具——硬件描述语言	270
10.3.1 硬件描述语言 Verilog - HDL 简介	271
10.3.2 硬件描述语言 VHDL 简介	274
10.4 现代数字系统的设计方法和流程	280
10.4.1 传统的电路设计流程及其困境	280
10.4.2 现代电子设计的基本流程和方法	280
本章习题	282
参考文献	284

第 1 章

数制与码制

1.1 数制

自然界中存在的各种各样的物理量,根据其是否随时间连续变化可分为模拟量和数字量两大类。这些物理量有些本身就是电信号,其他的也大多可以用传感器转换成电信号,再由电路对其进行处理。能够产生、处理模拟信号的电路称为模拟电路;而能够产生、处理数字信号的电路称为数字电路。与在时间和幅值上均连续变化的模拟信号相比,在时间和幅值上均离散变化的数字信号具有噪声容限大、抗干扰能力强、易于加工和存储等优点。因此,数字电路和系统获得了越来越广泛的应用,在电子系统中已占据绝对的统治地位。

数字电路和系统的基本功能就是加工和处理数据,包括数值信息和经过编码的非数值信息,因此要学习和掌握数字电子技术,首先必须学好数制与码制方面的知识。

1.1.1 计数体制

人类的一切活动都离不开计数。最常用的十进制据说就起源于人类双手的十指。当需要计的数较大时仅用 1 位数字便不够用,于是创造和采用了多位数。同样,在细分数量的需求驱动下小数也被创造和采用了。在多位数中(包括整数和小数),同一个数码在不同的位置就代表不同的数量,(较右边)低位和(较左边)相邻高位之间相差一个“数量级”,因而存在着进位关系。按进位的原则进行计数,就称为进位计数制。任何一种进位计数制都有一组特定的数码,这些不同

数码的个数称为其基数或底数 R 。相应的,常将基数为 R 的进位计数制简称为 R 进制。所谓的数制(计数体制)就是某种进位计数制。数字电路中经常使用的数制包括十进制、二进制、十六进制和八进制。以下分别加以简要介绍。

1. 十进制(Decimal Number System)

十进制是日常生活中最常使用的进位计数制。十进制数的基本元素是 10 个不同的数码 0、1、2、…、9,以及小数点“.”。显然它的基数是 10,10 以上的数必须用多位数表示,其中低位和相邻高位之间的进位关系是“逢十进一”。

多位数中某一位置上数字 1 所代表的数值大小称为该位的“权”。整数部分从低位到高位每位的权依次为 $10^0, 10^1, 10^2, \dots$ 。小数部分从高位到低位每位的权依次为 $10^{-1}, 10^{-2}, 10^{-3}, \dots$ 。因此,一个多位数表示的数值等于每一位的基数乘以它的权,然后相加。例如, $125.3 = 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1}$ 。所以任意一个十进制数 N_D 的按权展开式为

$$N_D = \sum_{i=-m}^{n-1} k_i 10^i \quad (1-1-1)$$

其中, k_i 是第 i 位的系数,可以是 0~9 中的任何一个。 i 是整数且 $-m \leq i \leq n-1$,其中 n 是十进制数 N_D 的整数部分的位数, m 是其小数部分的位数。

同理,任意一个 R 进制,都可以按式(1-1-1)的形式展开,并求出相应的十进制数

$$N_R = \sum_{i=-m}^{n-1} k_i R^i \quad (1-1-2)$$

式中 k_i 是第 i 位的系数, R 是计数的基数, R^i 称为第 i 位的权, i 的取值范围同式(1-1-1)。

2. 二进制(Binary Number System)

在二进制数中,每一位数只有 0 和 1 两种可能的取值,所以计数的基数是 2。低位向高位进位的规则是“逢二进一”。根据式(1-1-2)可以将任意一个二进制数按每位的权展开,并按十进制数相加,即得到它所表示的十进制数的大小。例如

$$(1010.1111)_B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} = (10.9375)_D$$

将上式表示的关系写成一般形式,就得到二进制的按权展开式为

$$N_B = \sum_{i=-m}^{n-1} k_i 2^i \quad (1-1-3)$$

式中 k_i 的取值只有 0 或 1 两种可能, i 的取值范围与式(1-1-1)相同。

由于二进制数每位只可能有两种数值:0 或者 1,在数字系统中可用电子器件的两种不同状态来表示 1 位二进制数,例如:晶体管的截止和饱和,继电器的接通和断开,灯的亮和灭,电平的高和低,都可将其中一个状态定义为 0,另一个状态定义为 1。因此,二进制数的电路实现较简单、可靠,而且存储和传送也方便。

但与十进制相比较,二进制的缺点是人们对它不熟悉,使用不习惯,因此,在数字系统中,通常先将人们所熟悉的十进制数进行“十翻二”(十进制转换为二进制)的运算,变成数字系统能接受的二进制数输入,输出时再把运算结果进行“二翻十”(二进制转换为十进制)的转换,变成人们熟悉的十进制数。

二进制的另一缺点是用其表示一个数所要用的位数很多,不便于书写和记忆。因此,在数字系统中,通常采用八进制和十六进制作为二进制的缩写。

3. 八进制 (Octal Number System)

在八进制数中, 每一位数有 8 个不同的数码: 0 ~ 7。低位向高位进位的规则是“逢八进一”。所以计数的基数是 8。任意一个八进制数的按权展开为

$$N_o = \sum_{i=-m}^{n-1} k_i 8^i \quad (1-1-4)$$

并按十进制数相加, 即得到它所表示的十进制数的大小。

例如 $(23.4)_o = 2 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} = (19.5)_D$

4. 十六进制 (Hexadecimal Number System)

在十六进制数中, 每一位数有 16 个不同的数码: 0 ~ 9、A (代表 10, 其他依次类推)、B (11)、C (12)、D (13)、E (14)、F (15)。低位向高位进位的规则是“逢十六进一”。所以计数的基数是 16。任意一个十六进制数可按权展开为

$$N_H = \sum_{i=-m}^{n-1} k_i 16^i \quad (1-1-5)$$

再按十进制数相加, 即得到它所表示的十进制数的大小。例如

$$(5AE)_H = 5 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 = (1454)_D$$

1.1.2 不同数制之间的转换

1. 二进制数与十进制数的相互转换

(1) 二进制数转换成十进制数

二进制数转换成十进制数时, 只要将二进制数按权展开, 然后将各项数值按十进制数相加, 便可得到等值的十进制数。例如

$$(1011.11)_B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (11.75)_D$$

(2) 十进制数转换成二进制数

① 整数转换——除 2 取余法

对于任意的十进制整数, 可用 2 不断连除十进制数, 直到商是 0 为止, 所得的余数就是对应的二进制数 (先得的余数为最低位, 其他以此类推)。其原理如下: 为简便起见 (不失一般性), 假设某一个十进制数 N 可转换为 3 位二进制数

$$(N)_D = (K_2 K_1 K_0)_B$$

把该二进制数按权展开: $(K_2 K_1 K_0)_B = (K_2 \times 2^2 + K_1 \times 2^1 + K_0 \times 2^0)_B = [2(K_2 \times 2 + K_1) + K_0]_B$, 因此显然有

$$(N)_D \div 2 = K_2 \times 2 + K_1 \quad \cdots \cdots \text{余数为 } K_0$$

用商再除以 2 得

$$(K_2 \times 2 + K_1) \div 2 = K_2 \quad \cdots \cdots \text{余数为 } K_1$$

不断用前次的商除以 2。直到最后的商为零

$$K_2 \div 2 = 0 \quad \cdots \cdots \text{余数为 } K_2$$

可见, 将每次除以 2 所得的余数依序合并即得到十进制数 N 对应的二进制数 $(K_2 K_1 K_0)_B$ 。

例 1-1 将十进制数 25 转换为二进制数。

解: 将十进制数 25 不断地除以 2 取其余数, 并按顺序合并。

2 25	余数			
2 12	1	(K ₀)		
2 6	0	(K ₁)		
2 3	0	(K ₂)		
2 1	1	(K ₃)		
0	1	(K ₄)		

↑
读
数
次
序

故

$$(25)_D = (11001)_B$$

② 小数转换——乘2取整法

任何十进制数的纯小数部分均可用“乘2取整”的方法,求得相应的二进制数(先得的整数为最高位,其他以此类推)。其原理是:十进制纯小数转换为 m 位二进制纯小数时,可用按权展开式表示为

$$(N)_D = K_{-1} \times 2^{-1} + K_{-2} \times 2^{-2} + \cdots + K_{-m} \times 2^{-m}$$

将上式两边分别乘以2,得

$$2 \times (N)_D = K_{-1} + K_{-2} \times 2^{-1} + \cdots + K_{-m} \times 2^{-(m-1)} = K_{-1} + N_1$$

可知上式的整数部分即是 K_{-1} ,将其小数部分 N_1 再乘以2,得

$$2 \times N_1 = K_{-2} + K_{-3} \times 2^{-1} + \cdots + K_{-m} \times 2^{-(m-2)}$$

上式右边的整数部分即是 K_{-2} 。重复上述乘法运算,即可依次求得 K_{-1} ,直至求得 K_{-m} 。

上述计算过程中,如还未求到 K_{-m} 时小数部分已为零,说明此小数已精确转换,余下位数均为零;如求到 K_{-m} 时,小数部分仍不为零,则说明转换有误差,可按照误差要求取舍。

例1-2 将纯小数 $(0.8341)_D$ 转换成误差 ε 不大于 2^{-4} 的二进制纯小数。

解:

$0.8341 \times 2 = 1.6682$	取整	(K ₋₁)		
$0.6682 \times 2 = 1.3364$	 1	(K ₋₂)		
$0.3364 \times 2 = 0.6728$	 1	(K ₋₃)		
$0.6728 \times 2 = 1.3456$	 0	(K ₋₄)		
	 1	(K ₋₅)		

↓
读
数
次
序

所以

$$(0.8341)_D \approx (0.1101)_B$$

将一个带有整数和小数的十进制数转换成二进制数时,需将整数部分和小数部分分别按除2取余法和乘2取整法进行转换,然后再将两者的转换结果合并起来即可。

同理,若将十进制数转换成任意 R 进制数 N_R ,则整数部分转换应采用除 R 取余法;小数部分转换应采用乘 R 取整法。

2. 二进制数与八进制数、十六进制数之间相互转换

八进制数和十六进制数的基数分别为 $8=2^3$, $16=2^4$, 所以3位二进制数恰好对应于1位八进制数,4位二进制数对应于1位十六进制数,它们之间的相互转换因而十分方便。

二进制数转换成八进制数的方法是:从小数点开始,分别(整数部分)向左、(小数部分)向右,将二进制数按每3位一组分组(不足3位的补0),然后按组写出等值的八进制数。例如,求 $(01101111010.1011)_B$ 的等值八进制数:

二进制	001	101	111	010	101	100
八进制	1	5	7	2	5	4

所以 $(01101111010.1011)_B = (1572.54)_O$

二进制数转换成十六进制数的方法和二进制数转换成八进制数相似,即从小数点开始分别向左、向右将二进制数按每 4 位一组分组(不足 4 位补 0),然后写出每组等值的十六进制数。

例如,将 $(1101101011.101)_B$ 转换为十六进制数:

二进制	0011	0110	1011	1010
十六进制	3	6	B	A

所以 $(1101101011.101)_B = (36B.A)_H$

八进制数或十六进制数转换为二进制数时,可以采用与前面相反的步骤,即,只需按原有位序将每位八进制数或十六进制数用相应的 3 位或 4 位二进制数代替即可。例如

$(375.46)_O = (011\ 111\ 101.100\ 110)_B, (678.A5)_H = (0110\ 0111\ 1000.1010\ 0101)_B$

3. 任意进制数与十进制数之间的相互转换

任意进制数与十进制数之间的转换原理及方法,同二进制数与十进制数之间的转换原理和方法相类似。为了方便和减少差错,通常的做法是先由一种进制数转换为二进制数,再由二进制数转换为另一进制数,以二进制数作为桥梁进行过渡。

1.2 编 码

在数字系统中,常用预先规定的方法将文字、数字或其他对象映射为二进制的数码以便处理,这种给信息分配的二进制代码称为对信息的编码。

1.2.1 二 - 十进制编码

数字系统在其内部处理二进制数据和代码,人们日常却习惯使用十进制数,为了便于编程、调试等工作,遂研究和采用了 4 位二进制数表示 1 位十进制数的方法。这种用于表示十进制数的二进制代码称为二 - 十进制代码(Binary Coded Decimal),简称 BCD 码。

十进制数有 0~9 共 10 个数码,所以表示 1 位十进制数至少需要 4 位二进制数,但 4 位二进制数可以产生 $2^4 = 16$ 种组合,因此用 4 位二进制数表示 1 位十进制数时有 6 种组合是多余的。因而,十进制数的二进制编码可以有多种编码方案,每种编码方案都有它的特点。表 1-2-1 列举了目前常用的几种编码方案。

表 1-2-1 4 种常用的 BCD 编码

十进制数	8421	5421	2421	余 3 码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110

续表

十进制数	8421	5421	2421	余 3 码
4	0100	0100	0100	0111
5	0101	1000	0101	1000
6	0110	1001	0110	1001
7	0111	1010	0111	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

8421 码是一种最简单、最常用的 BCD 编码(二 - 十进制代码)。这种编码是将每个十进制数码用 4 位二进制数表示,按自然二进制数的规律排列,并且指定前面 10 个代码依次表示数码 0 ~ 9。8421 码是一种有权码,每位都有固定的权。各位的权从左到右分别是 8、4、2、1。必须指出,在 8421BCD 码中不允许出现 1010 ~ 1111 这些代码,因为在十进制中没有数码同它们对应。

5421 码是另一种有权码,自高位到低位,每位的权分别为 5、4、2、1。

2421 码也是一种有权码,每位的权从高位到低位为 2、4、2、1。2421 码是一种“对 9 的自补”代码。在这种编码中,十进制数 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的对应码位互为补码。

余 3 码是一种特殊的 8421 码,它是由 8421 码加 3 后形成的,所以叫余 3 码。它是一种无权码。例如,十进制数 4 在 8421 BCD 码中表示为 0100,在余 3 码中则表示为 0111。

1.2.2 可靠性编码

数字信息在产生和传输的过程中,难免由于速度差异、噪声、干扰等原因出现错误。为了减少这种错误,研究并采用了一种称为可靠性编码的方法。

1. 格雷码

可靠性编码有很多种,其中之一是格雷(Gray)码,又称循环码。其特点是任意两个相邻的格雷码仅有 1 位不同,所以在传输过程中不容易出错。用来表示十进制数的格雷码也属于无权码,并且和 BCD 码一样也有多种形式,表 1-2-2 列出了其中两种。

表 1-2-2 几种格雷码和二进制码对照表

十进制数	二进制码	格雷码 1(十进制)	格雷码 2(十进制)	典型格雷码	步进码
0	0000	0000	0000	0000	00000
1	0001	0001	0001	0001	00001
2	0010	0011	0011	0011	00011
3	0011	0010	0010	0010	00111
4	0100	0110	0110	0110	01111
5	0101	1110	0111	0111	11111

续表

十进制数	二进制码	格雷码 1(十进制)	格雷码 2(十进制)	典型格雷码	步进码
6	0110	1010	0101	0101	11110
7	0111	1011	0100	0100	11100
8	1000	1001	1100	1100	11000
9	1001	1000	1000	1101	10000
10	1010			1111	
11	1011			1110	
12	1100			1010	
13	1101			1011	
14	1110			1001	
15	1111			1000	

另一种常用的格雷码是步进码。这种步进码在加 1 计数时,只需将最左 1 位求反后移到最右 1 位,而其余各位都向左移 1 位。表 1-2-2 中列出了 5 位的步进码,它的循环长度为 10。

2. 奇偶校验码

奇偶校验码是一种能检验出二进制信息在传递过程中是否出现错误的代码。这种代码由两部分组成:一部分是信息位,这就是需要传送的信息本身;另一部分是奇偶校验位,其定义和作用是使整个代码中 1 的个数按预先的规定成为奇数或偶数。当信息位和校验位中 1 的总个数为奇数时,称为奇校验,而 1 的总个数为偶数时,称为偶校验。由 4 位信息位及 1 位奇偶校验位构成的 5 位奇偶校验码见表 1-2-3。

表 1-2-3 十进制数码的奇偶校验码

十进制数码	带奇校验的 8421 码		带偶校验的 8421 码	
	信息位	校验位	信息位	校验位
0	0000	1	0000	0
1	0001	0	0001	1
2	0010	0	0010	1
3	0011	1	0011	0
4	0100	0	0100	1
5	0101	1	0101	0
6	0110	1	0110	0
7	0111	0	0111	1
8	1000	0	1000	1
9	1001	1	1001	0

这种编码的特点是:每个代码中含有 1 的个数总是奇(或偶)数,这样,一旦某代码在传递过程中出现 1 的个数不是奇(或偶)数时就会被发现。它因此常被用于校验(即查错)。

1.2.3 字符编码

计算机等数字系统需要处理的数据不仅有数字,还有字母、标点符号、运算符号及其他特殊符号。这些数字、字母和专用符号统称字符。字符在计算机内部都必须用二进制代码来表示。它们的编码称为字符编码。

ASCII 码,即美国标准信息交换码(American Standard Code for Information Interchange)。它已被国际标准化组织确定为国际标准字符代码。该代码由 7 位二进制数组成,共可表示 128 个字符,包括 0~9 这 10 个十进制数码、26 个英文字母(分大、小写)及其他一些符号和标记。7 位 ASCII 码表见表 1-2-4。其中,以词组缩写为名称的字符(例如前 32 个字符)为非显示的控制符,主要用于主机与外设间的通信,其具体含义见表 1-2-5;其余字符主要用于人机交互,包括键盘输入、屏幕显示等。

值得注意的是,在 ASCII 码表中不仅字符和代码一一对应,而且数字、字母均被按照从小到大、从前到后的顺序,很有规律地编码,有了这种规律,通过代码之间的比较就可以获知相应的数字、字母之间的大小、先后,也很容易实现大、小写字母的相互转换。

此外,在 7 位 ASCII 码的基础上又研发和推广了 8 位 ASCII 码,称为扩展 ASCII 码,它向下兼容于 7 位 ASCII 码。中文字符也采用了类似的编码原理,但(基本)编码位数增加到了 16 位(因为汉字的个数远超英文字母),而且也 and ASCII 码兼容。

表 1-2-4 美国标准信息交换码

$b_6b_5b_4 \backslash b_3b_2b_1b_0$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOM	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VF	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	!
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	↓	o	DEL