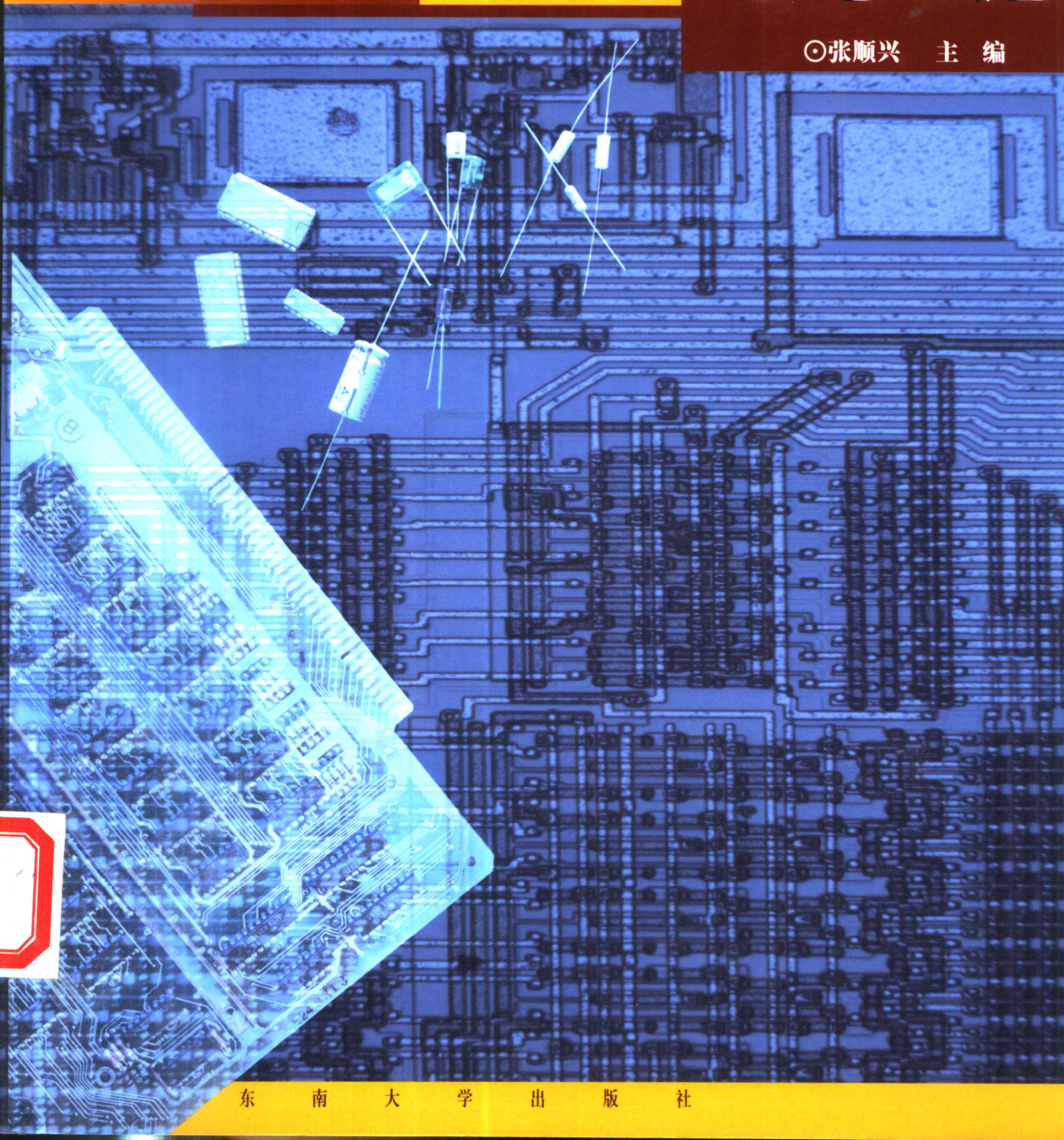
SHUZIDIANLU YU XITONG

数字电路与系统

⊙张顺兴　主　编



东　南　大　学　出　版　社

内 容 简 介

本书较详细地介绍了数字电路与系统的基本理论及其设计方法。全书内容参照原国家教委颁布的本课程教学基本要求进行编写，并充实了反映当前数字电子技术发展水平的 PLD 和 EDA 等方面的内容，符合当前我国高等学校工科本课程教学内容的实际需要。

全书共 11 章，内容包括数制与码制、逻辑代数基础、集成逻辑门、组合逻辑电路、触发器、时序逻辑电路、脉冲信号的产生和变换、D/A 和 A/D 变换、半导体存储器、可编程逻辑器件、数字系统设计(含 VHDL)。书末有附录英汉名词术语对照表。

本书可作为高等学校电气信息类通信工程、电子信息工程、计算机科学与技术、自动化等专业的教科书，也可供其他相关专业选用和社会读者阅读参考。

图书在版编目(CIP)数据

数字电路与系统/张顺兴主编 .—南京：东南大学出版社，2001.2
ISBN 7-81050-711-7

Ⅰ.数… Ⅱ.张… Ⅲ.①数字电路②数字系统 Ⅳ.TN311.12

中国版本图书馆 CIP 数据核字(2000)第 84021 号

东南大学出版社出版发行
(南京四牌楼 2 号 邮编 210096)
出版人：宋增民
江苏省新华书店经销 南京邮电学院印刷厂印刷
开本：787 mm×1092 mm 1/16 印张：21.5 字数：550 千字
2001 年 2 月第 1 版 2002 年 8 月第 2 次印刷
印数：4051—7050 册 定价：26.00 元
(凡因印装质量问题，可直接向发行科调换。电话：025-3792327)

前　言

本书是面向高等学校工科电气信息类各专业及其他类相关专业的专业基础教科书。其先修课程为模拟电子线路、程序设计语言，后续课程为计算机组成原理、微机原理和接口技术等。

为了适应大规模集成电路(LSI)和超大规模集成电路(VLSI)日益广泛的应用以及计算机已经普遍使用的现实，本书在介绍了数字电路课程的基本内容之后，着重介绍了可编程逻辑器件(PLD)和数字系统的设计方法，并在数字系统设计中介绍了超高速集成电路硬件描述语言(VHDL)及其应用。

本书承教育部高等学校工科电工课程指导委员会电子技术与电子线路课程指导小组委员、南京航空航天大学沈嗣昌教授审阅。沈教授以其深厚的理论造诣和丰富的教学经验对本书从结构到文字都提出了许多极为宝贵的修改意见，编者表示深切的谢意！

本书由张顺兴主编。具体分工为：第1、2、5、6、7章及附录由张顺兴编写；第3、4、8章由盛利编写(后又经杨恒新对第3、4章作了较大修改，黄丽亚对第8章作了修改)；第9章由黄丽亚编写；第10章由杨恒新编写；第11章由杨秀侠编写。全书由张顺兴统稿。

在本书的编写过程中，南京邮电学院王锁萍教授对编写的指导思想提出了具体意见，并审阅了初稿；糜正琨教授、朱洪波教授、沈元隆教授、桂志波博士提出了许多极为中肯的宝贵意见；教务处处长梅杓春教授、电子工程系副主任蒋国平博士、教材科方小玉科长对本书的出版给予了大力支持和帮助；东南大学出版社陈跃编辑为本书的及时出版做了大量缜密细致的工作。编者在此一并表示衷心的感谢！

在本书所介绍的集成电路的逻辑符号中，对现行国标(GB/T 4728.12—1996)符号及国内外曾用、常用的符号都作了介绍，并尽可能采用国标符号，但为了便于教学，对于应用电路中的MSI器件则使用了示意性的简化符号。在简化符号中，对于低电平有效的信号在其端子上加小圈“○”表示，并在信号名上加反号“—”，而且带反号的信号名可视需要写在逻辑框的外部或内部，这与国标是不完全一致的(在国标中，逻辑框内不使用带反号“—”的变量)。

限于编者水平和成书时间仓促，书中定有许多错漏和不妥之处，期望读者批评指正。

编　者

2000.12

目　录

1 数制与码制

内容提要 本章介绍数制与码制两部分内容。在数制部分首先概述数制中的基本概念，然后介绍各种常用数制及其互相间的转换，最后介绍二进制数的算术运算；在码制部分介绍码制的基本概念以及常用的二进制码和 BCD 码。

1.1 数制

1.1.1 概述

数制是计数体制的简称，有累加计数制和进位计数制两种。累加计数制是原始的计数方法，计多大的数要使用与所计数目个数相等的各不相同的计数符号，很不方便。比较方便的计数方法是进位计数制，本章所述数制即指进位计数制。常用的进位计数制有十进位计数制(简称十进制)以及二、八、十六进制。在介绍各种数制之前，首先介绍数制的基础知识。

1) 数码

数码是基本数码的简称，是指计数制中使用的基本数字符号。例如，十进制中的 0、1、2、3、4、5、6、7、8、9 便是。

2) 基数(R)

计数制中所使用的数码的个数称为基数，亦称底数。基数常用R表示，在十进制中 $R=10$。在基数为 R 的数制中，每一个数位上可以使用的数码包括 0 在内共有 R 个，最大数码是 $R-1$，而没有 R，因此计数时当某位数计到 R 时，则在该位记作 0，并向高位进一，即逢 R 进一，故基数为 R 的计数制称为 R 进制计数制。

3) 数位(i)

在由一串数码构成的数中，数码所在的位置称数位。数位的排序用 i 表示，i 的计算以小数点为界，向左依次为第 0 位、第 1 位、……向右依次为第 -1 位、第 -2 位、……例如，在十进制数 123.45 中，3 是第 0 位数，2 是第 1 位数，4 是第 -1 位数等等。

4) 位权(weight)

位权亦称权值。在进位计数制的由一串数码构成的数中，各个数位上的数码所表示的数值的大小不但和该数码本身的大小有关，而且还和该数码所处的数位有关。例如，在十进制数 44 中，十位数 4 表示 4×10^1，个位数 4 表示 4×10^0。可见，不同的数位赋予该位上的数码以不同的表示数的大小的权力。我们把数位上的数码在表示数时所乘的倍数称为该数位的位权。

在 R 进制数中，第 i 位数位的权值用 W_i 表示，$W_i=R^i$。其中 R 是基数，i 是数位的位数。例如，十进制数的第 0 位(个位)、第 1 位、第 -1 位的位权分别为 10^0、10^1 和 10^{-1}。

在由一串数码表示的数中，相邻两个数位中左边数位的位权是右边的 R 倍。

5) 数的表示方式

在进位计数制中数的表示方式有位置记数法、按权展开式、和式 3 种。

(1) 以十进制数 123.45 为例,分别可以表示为

$N^{①}_{10} = 123.45$ ……①　位置记数法

$= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$ ……②　按权展开式

$= \sum_{i=-2}^{2} D^{②}_i \times 10^i$ ……③　和式

上述①式用一串数码来表示数,称为位置记数法或并列记数法。②式称为按权展开式,这是用多项式来表示一个数,又称多项式表示法。多项式中的各个乘积项由各个数位上的数码加权(加权即乘上权值)构成。③式是把按权展开式表示为$\sum$的形式,称为和式。式中 D_i 表示第 i 位数位上的十进制数码,10^i 为第 i 位的权值。

(2) 对于任意一个 R 进制数$(N)_R$,可以分别用 3 种方式表示如下:

$(N)_R = a_{m-1}a_{m-2}\cdots a_1 a_0 a_{-1} a_{-2}\cdots a_{-n}$ …… ① 位置记数法

$= a_{m-1} \times R^{m-1} + a_{m-2} \times R^{m-2} + \cdots + a_1 \times R^1$

$+ a_0 \times R^0 + a_{-1} \times R^{-1} + \cdots + a_{-n} \cdot R^{-n}$ …… ② 按权展开式

$= \sum_{i=-n}^{m-1} a_i R^i$ …… ③ 和式

式中　$a_{m-1}, a_{m-2}, \cdots, a_{-n}$——分别为第 $m-1, m-2, \cdots -n$ 位上的数码;

R——基数;

m、n——分别为整数部分的位数和小数部分的位数;

i——数位的序号。

1.1.2　常用数制

1) 二进制

数码:0、1

基数:$R = 2$

第 i 位的权值:$W_i = 2^i$

例　表示方式:

$$(101.01)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= \sum_{i=-2}^{2} B_i \times 2^i$$

2) 八进制

数码:0、1、2、3、4、5、6、7

基数:$R = 8$

第 i 位的权值:$W_i = 8^i$

例　表示方式:

$$(25.6)_8 = 2 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} = \sum_{i=-1}^{1} O_i \times 8^i$$

---

① 脚标 10 表示 N 为十进制数。二、八、十、十六进制数分别用脚标 2、8、10、16 或 B、O、D、H 表示。

② D 表示十进制数码。二、八、十六进制数的数码分别用 B、O、H 表示。

3）十六进制

数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F

其中数码 A、B、C、D、E、F 依次与十进制数 10、11、12、13、14、15 等值，但 A、B、C、D、E、F 在十六进制中是 1 位数。

基数：$R=16$

第 i 位的权值：$W_i=16^i$

例　表示方式：

$$(12D.23)_{16}=1\times16^2+2\times16^1+13\times16^0+2\times16^{-1}+3\times16^{-2}$$

$$=\sum_{i=-2}^{2}H_i\times16^i$$

表 1.1.1 列出了十进制数 0～20 等数在其他各种数制中的写法。

表 1.1.1　十、二、八、十六进制数对照表

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	12	1100	14	C
1	1	1	1	13	1101	15	D
2	10	2	2	14	1110	16	E
3	11	3	3	15	1111	17	F
4	100	4	4	16	10000	20	10
5	101	5	5	17	10001	21	11
6	110	6	6	18	10010	22	12
7	111	7	7	19	10011	23	13
8	1000	10	8	20	10100	24	14
9	1001	11	9	32	100000	40	20
10	1010	12	A	100	1100100	144	64
11	1011	13	B	1000	1 111 101000	1750	3E8

1.1.3　数制转换

所谓数制转换是指将一种数制的数变换成等值的用另一种数制表示的数。

1）二进制数（八、十六）→十进制数

将二进制、八进制、十六进制数转换成十进制数只要把原数写成按权展开式再相加即可。

例 1.1.1　分别将 $(101.01)_2$，$(74.5)_8$，$(3C.A)_{16}$ 转换成十进制数。

解　①$(101.01)_2=1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$

$=(5.25)_{10}$

②$(74.5)_8=7\times8^1+4\times8^0+5\times8^{-1}$

$=(60.625)_{10}$

③$(3C.A)_{16}=3\times16^1+12\times16^0+10\times16^{-1}$

$=(60.625)_{10}$

2）十进制数→二（八、十六）进制数

十进制数转换成二、八、十六进制数只需将整数部分和小数部分分别转换成二、八、十六

进制数，再将转换结果连接在一起即可。

(1) 整数的转换用“除基数取余法” 将十进制数除以目标数制(所要转换成的数制)的基数，得商和余数，取下余数作为目标数制数的最低位数码；将所得的商再除以基数又得商和余数，取下余数作为目标数制数的次低位数码……如此连续进行，直至商为0余数小于目标数制的基数 R 为止，末次相除所得的余数为目标数制数的最高位数码。

例 1.1.2 将$(60)_{10}$分别转换成二、八、十六进制数。

解 ①

```
2 | 60
2 | 30 ……余 0……最低位
2 | 15        0
2 |  7        1
2 |  3        1
2 |  1        1
     0        1……最高位
```

所以 $(60)_{10}=(111100)_2$。

②

```
8 | 60
8 |  7 ……余 4……最低位
     0        7……最高位
```

所以 $(60)_{10}=(74)_8$。

③

```
16 | 60
16 |  3 ……余 12……最低位
      0        3……最高位
```

所以 $(60)_{10}=(3C)_{16}$

(2) 小数的转换用“乘基数取整法”

将被转换的十进制小数乘以目标数制的基数 R，取下所得乘积中的整数作为目标数小数的最高位；再将乘积中的小数部分乘以基数 R，取下乘积中的整数作为目标数小数的次高位……如此反复进行，直到乘积的小数部分为0或达到所需精度为止。各次相乘所得的整数即可构成目标数的小数，第一次相乘所得的整数为小数的最高位，末次相乘所得的整数为小数的最低位。

例 1.1.3 将十进制数$(0.625)_{10}$分别转换成二、八、十六进制数。

解 ①

```
    0.625
  ×     2
  ---------
(1).250      最高位
  ×     2
  ---------
(0).500
  ×     2
  ---------
(1).000      最低位
```

所以 $(0.625)_{10}=(0.101)_2$

②

```
    0.625
  ×     8
  ---------
(5).000
```

所以　　$(0.625)_{10}=(0.5)_8$

③

$$\begin{array}{r} 0.625 \\ \times \quad 16 \\ \hline 3.750 \\ +\ 6.25 \\ \hline (10).000 \end{array}$$

所以　　$(0.625)_{10}=(0.A)_{16}$

(3) 剩余误差及转换位数的确定

一个 R 进制 n 位小数的精度为 R^{-n}，例如 $(0.95)_{10}$ 的精度为 10^{-2}。采用乘基数取整法将十进制小数转换成二、八、十六进制小数时，可能出现多次相乘后乘积的小数部分仍不为 0 的情况。如果转换取了 n 位，则转换的剩余误差 Δ 小于该 n 位小数的精度，即 $\Delta < R^{-n}$。例如，$(0.95)_{10}=(0.746314631\cdots)_8$，当转换取3位时，可得 $(0.95)_{10}=(0.746)_8$，则 $\Delta=(0.0003146314631\cdots)_8$，$\Delta<8^{-3}$。

一个 j 位的 α 进制小数，当其转换成 β 进制小数时，如要求转换后保持或不低于原来的精度，其转换成的 β 进制小数应取的最少位数(设为 k 位)计算如下：

由于原精度为 α^{-j}，转换后的精度为 β^{-k}，根据转换精度的要求应有 $\alpha^{-j}\geqslant\beta^{-k}$，可以求得 $k\geqslant j\dfrac{\lg\alpha}{\lg\beta}$。

k 应取满足上式的最小正整数，即

$$j\frac{\lg\alpha}{\lg\beta}\leqslant k<j\frac{\lg\alpha}{\lg\beta}+1$$

3) 两种任意数制的转换

把一个 α 进制数转换成 β 进制数可以利用十进制数作为桥梁：

$$\alpha\text{ 进制数}\xrightarrow{\text{按权展开相加}}\text{十进制数}\xrightarrow{\text{基数乘除法}}\beta\text{ 进制数}$$

如果 α 和 β 都是 2 的正整数幂，则可进行直接转换。例如，当进行二⇌八(十六)进制转换时，由于3(4)位二进制数可以用1位八(十六)进制数表示，因此，二进制数表示为八(十六)进制数时可把二进制数以小数点为界分别向左和向右每 3(4)位组成一组，组成时不足 3(4)位时整数部分高位添 0，小数部分低位添 0，使其补足 3(4)位，然后将每组 3(4)位二进制数用 1 位八(十六)进制数表示即可，反之亦然。例如：

$$(0\,\underline{10}\,\underline{101}.\,\underline{001}\quad\underline{110})_2\rightleftharpoons(25.16)_8$$

$$(0\,0\,\underline{10}\quad\underline{1101}.\underline{0100}\quad\underline{1000})_2\rightleftharpoons(2D.48)_{16}$$

在用上述方法将八(十六)进制数转换为二进制数时需要舍去多余的 0(整数部分最高位的 0 和小数部分最低位的 0)。

需要说明的是，在数字技术中用得最多的是二进制。这是因为二进制中只有 0 和 1 两个数码，很容易用高低电平来表示；如果采用十进制，则需要用 10 个不同的状态来表示十进制中 10 个不同的数码，很不容易。此外，二进制数的运算也比较简单，因此机器数都用二进制数表示。但是用二进制表示的数往往很长，不便于书写和记忆，相对而言，八进制、十六进制数的书写和记忆比较方便，而且和二进制数之间的转换也很方便，因此，又常用八进制和

十六进制数作为二进制数的缩写形式用于书写、输入和显示。

1.1.4 二进制数的算术运算

当两个二进制数用来表示数量的大小时，它们可以进行数值运算，这种运算称为算术运算。二进制数算术运算的法则和十进制数的运算法则基本相同。唯一的区别在于十进制数是逢十进一、借一当十，而二进制数是逢二进一、借一当二。

以两个二进制数 1001 和 0101 为例，其算术运算有：

加法运算

```
  1 0 0 1
+ 0 1 0 1
---------
  1 1 1 0
```

减法运算

```
  1 0 0 1
- 0 1 0 1
---------
  0 1 0 0
```

乘法运算

```
        1 0 0 1
      × 0 1 0 1
---------------
        1 0 0 1
      0 0 0 0
    1 0 0 1
+ 0 0 0 0
---------------
  0 1 0 1 1 0 1
```

除法运算

```
          1.1 1…
       ---------
0101) 1 0 0 1
      -0 1 0 1
      ---------
        1 0 0 0
        -0 1 0 1
        ---------
          0 1 1 0
          -0 1 0 1
          ---------
            0 0 1
```

需要说明的是，在数字电路中为了简化运算电路，通常二数相减的运算是用其补码的加法来实现的，乘法运算则用移位和加法两种操作来完成，而除法运算是用移位和减法操作来完成，因此，二进制数的加、减、乘、除都可以用加法电路完成。所以在数字设备中加法器是极为重要的运算部件。

1.2 码制

用文字、符号或数码来表示各个特定对象的过程称为编码，编码所得的每组符号称为代码或码字，代码中的每个符号称为基本代码或码元。在数字电路中通常用二进制数码 0 和 1 构成的代码来表示各有关对象(如十进制数、字符等)。码制是指编码的制式，不同的码制编码时遵循不同的规则。

1.2.1 二进制码

所谓二进制代码是指用二进制数码 0 和 1(这里代码中的码元 0、1 不一定有大小的概念)构成的代码。n 位的二进制代码可以有 2^n 个代码。表 1.2.1 给出了几种典型的二进制代码。

表 1.2.1 典型二进制码

十进制数	4 位自然二进制码	典型格雷码（循环码）	8421 奇(偶)校验码			
			信息码	P(奇)	信息码	P(偶)
0	0 0 0 0	0 0 0 0	0 0 0 0	1	0 0 0 0	0
1	0 0 0 1	0 0 0 1	0 0 0 1	0	0 0 0 1	1
2	0 0 1 0	0 0 1 1	0 0 1 0	0	0 0 1 0	1
3	0 0 1 1	0 0 1 0	0 0 1 1	1	0 0 1 1	0
4	0 1 0 0	0 1 1 0	0 1 0 0	0	0 1 0 0	1
5	0 1 0 1	0 1 1 1	0 1 0 1	1	0 1 0 1	0
6	0 1 1 0	0 1 0 1	0 1 1 0	1	0 1 1 0	0
7	0 1 1 1	0 1 0 0	0 1 1 1	0	0 1 1 1	1
8	1 0 0 0	1 1 0 0	1 0 0 0	0	1 0 0 0	1
9	1 0 0 1	1 1 0 1	1 0 0 1	1	1 0 0 1	0
10	1 0 1 0	1 1 1 1	1 0 1 0	1	1 0 1 0	0
11	1 0 1 1	1 1 1 0	1 0 1 1	0	1 0 1 1	1
12	1 1 0 0	1 0 1 0	1 1 0 0	1	1 1 0 0	0
13	1 1 0 1	1 0 1 1	1 1 0 1	0	1 1 0 1	1
14	1 1 1 0	1 0 0 1	1 1 1 0	0	1 1 1 0	1
15	1 1 1 1	1 0 0 0	1 1 1 1	1	1 1 1 1	0

1）自然二进制码

自然二进制码是通常用以表示数值的一种二进制码。从编码的角度看，二进制数也是一种表示数的代码，称为自然二进制码。例如，1100 既可以说它是数十二的二进制数，又可以说它是数十二的自然二进制码。不过，虽然一个数的自然二进制码和其二进制数在写法上完全一样，但在概念上是不一样的，前者是码制中的概念，后者是数制中的概念。表 1.2.1 中给出了 4 位自然二进制码，代码中每个码元的位权自左至右分别为 8、4、2、1，16 个代码依次分别用来表示数 0～15。

2）格雷码、循环码

在一组数的编码中，若任意两个相邻数的代码中只有一位对应的码元不同，则称这种编码为格雷码(Gray Code)。格雷码的种类很多，循环码是其中的典型。在循环码中，不仅相邻的两个代码只有 1 位码元不同，而且首尾两个代码也是如此，表 1.2.1 中给出了 4 位循环码。使用循环码可以减少电路工作时出错的可能。通常把两个代码中取值不同的码元的位数称为两个代码的间距，把两个相邻代码中只有一位对应码元取值不同的特点称为单位间距特性。格雷码和循环码都具有单位间距特性，因此，它们都是单位间距码。如表 1.2.1 循环码中虚线所示，循环码的各个代码除最左位以外，其他各位均以最左位 0 和 1 的水平分界线为轴镜像对称；循环码的最左位，分界线以上均为 0，以下均为 1。上述特点称为循环码的反射特性。利用反射特性可以由 n 位循环码方便地写出($n+1$)位循环码。例如，1 位、2 位、3 位循环码如下所示：

1位	2位	3位
0	0 0	0 0 0
1	0 1	0 0 1
	……	0 1 1
	1 1	0 1 0
	1 0	……
	↑	1 1 0
	反射位	1 1 1
		1 0 1
		1 0 0
		↑
		反射位

3) 奇(偶)校验码

奇(偶)校验码示于表1.2.1中。这种码由信息码加一个奇校验位P(奇)或偶校验位P(偶)构成,其中校验位的取值(0或1)将使各个代码(包括信息码和校验位)中"1"的个数都为奇数或都为偶数。若使"1"的个数为奇数,称为奇校验码;为偶数,则称为偶校验码。8421奇(偶)校验码是以8421码作为信息码构成的奇(偶)校验码,如表1.2.2中所示。任何一种n位二进制码,只要增加一个校验位,便可构成$(n+1)$位的奇校码或偶校码。奇(偶)校验码的用处是可以通过检测经传输以后的代码中"1"的个数是否为奇数(或是否为偶数),即进行奇校验(或偶校验)来判断信息在传输过程中是否有一位码元出错。

格雷码、循环码、奇(偶)校验码都是一种可靠性编码。奇、偶校验码具有一定的检测错码的能力,是一种误差检验码,但只能检错而不能纠错。使用海明码则可以达到检测并纠正错码的要求。

1.2.2 二—十进制(BCD)码

所谓二—十进制码又称BCD(Binary Coded Decimal)码,是指用二进制数码0和1的编码来表示的十进制数码0,1,…,9,或者说是十进制数码0,1,…,9的二进制编码。要表示0~9这10个十进制数码,使用的代码至少需有4位码元。由于4位二进制数可以构成16个不同的代码,因此,构成BCD的方案可以多达$A_{16}^{10}\approx 290$亿种,不过最常用的也只是几种。BCD码可以分为有权码和无权码两大类,表1.2.2列出了8种最常用的BCD码。

表1.2.2 常用BCD码

十进制数	8421码	5421码	2421(B)码	631-1码	余3码	余3循环码	格雷码(2)	8421奇校码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 1 1	0 0 1 0	0 0 0 0	0 0 0 0 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 1 0	0 1 0 0	0 1 1 0	0 0 0 1	0 0 0 1 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 1 0 1	0 1 1 1	0 0 1 1	0 0 1 0 0
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 1	0 1 1 0	0 1 0 1	0 0 1 0	0 0 1 1 1
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 0	0 1 1 1	0 1 0 0	0 1 1 0	0 1 0 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 1	1 0 0 0	1 1 0 0	0 1 1 1	0 1 0 1 1
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 0 0	1 0 0 1	1 1 0 1	0 1 0 1	0 1 1 0 1
7	0 1 1 1	1 0 1 0	1 1 0 1	1 0 1 0	1 0 1 0	1 1 1 1	0 1 0 0	0 1 1 1 0
8	1 0 0 0	1 0 1 1	1 1 1 0	1 1 0 1	1 0 1 1	1 1 1 0	1 1 0 0	1 0 0 0 0
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 0 0	1 1 0 0	1 0 1 0	1 0 0 0	1 0 0 1 0