







第1章 数制与码制

Leng Ximo

数制与码制

-  几种常用的数制
-  不同数制间的转换
-  二进制算术运算
-  几种常用的编码

几种常用的数制

数制的概念

数制，又称“计数制”，指用一组固定的符号和统一的规则来表示数值的方法；

数制关注的内容：每一位的构成（使用什么符号即数码）和进位的规则；

基本概念：

- 基数 —— 使用的数码的个数，即每一位最多能用多少个符号来表示；
基数是 N 则称为 N 进制，即“逢 N 进一”；
- 权 —— 每一位的 1 代表的大小；

几种常用的数制

○ 几种常用的数制

	基数	展开形式	脚注符号
十进制	10 每一位用 0 ~ 9 表示	$D = \sum k_i 10^i$	D
二进制	2 每一位用 0 / 1 表示	$D = \sum k_i 2^i$	B
八进制	8 每一位用 0 ~ 7 表示	$D = \sum k_i 8^i$	O
十六进制	16 每一位用 0 ~ F 表示	$D = \sum k_i 16^i$	H

几种常用的数制

○ 几种常用的数制 —— 不同进制对照表

十进制数	二进制	八进制	十六进制
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

必须熟练掌握！

不同数制间的转换

○ 其他进制 → 十进制

只需要按照 $D = \sum k_i N^i$ 展开即可；

其中： k_i 为第 i 位的系数， N^i 为第 i 位的权， N 为基数；

○ 十进制 → 二进制

整数部分反复除 2 取余数，倒排（从右向左排列）（从低位到高位排列）；

小数部分反复乘 2 取整数部分，正排（从左向右排列）（从高位到低位排列）；

不同数制间的转换

○ 二进制 → 十六进制

以小数点为分割：

整数部分从右向左（从低位到高位）每四个划为一组（不够的在左边补 0），

小数部分从左向右（从高位到低位）每四个划为一组（不够的在右边补 0）；

○ 十六进制 → 二进制

将每一位用对应的四位二进制数代替即可；

不同数制间的转换

○ 二进制 → 八进制

以小数点为分割：

整数部分从右向左（从低位到高位）每三个划为一组（不够的在左边补 0），

小数部分从左向右（从高位到低位）每三个划为一组（不够的在右边补 0）；

○ 八进制 → 二进制

将每一位用对应的三位二进制数代替即可；

不同数制间的转换

○ 十进制 → 八进制、十六进制

先转换为二进制，再将二进制转换为八进制、十六进制；

○ 八进制与十六进制之间的转换

先转换为二进制，再将二进制转换为目标进制；

关键： 理解数制的概念与转换的原理，不要机械式地背下来！

二进制算术运算

○ 二进制算术运算的特点

- ① 和十进制算数运算的规则基本相同，区别在于“逢二进一”；
- ② 乘法和除法可以转化为移位操作与加减操作；
- ③ 通过补码运算（即将引入的概念），可以将减法运算转换为加法运算；

因此，数字电路普遍采用二进制进行算术运算；

二进制算术运算

○ 如何表示二进制数的正负？ —— 在前面增加一个符号位，0 表示正数，1 表示负数；

注意：一个二进制数的最高位是符号位，符号位的 0 / 1 不代表大小，只是代表正负；

○ 原码、反码、补码

正数（符号位为 0） —— 原码 = 反码 = 补码；

负数（符号位为 1） —— 反码为所有数字位取反，补码等于反码加 1；

补码的补码即为原码；

二进制算术运算

○ 关于原码、反码、补码的易错点：

①反码取反是指对数字位取反，不改变符号位！

例如 -7 即 1111 的反码是 1000，不是 0000 ！

②补码是不对称的，例如补码为 1000 代表 -8，而不代表 -0 ！

即同样位数的二进制补码，负数会比正数 “多一个” ；

③补码的 “加 1” 指的是在最低位加 1 的二进制加法运算，不是指数值 1，
即如果含有小数部分，1 是加在整体的最右边；

二进制算术运算

利用二进制补码运算求解带符号数的加法运算

Step 1 —— 分析需要使用多少个数字位；（培养思维习惯！不能忽略！）

Step 2 —— 用带符号位的二进制数表示，求解每个二进制数的补码；

Step 3 —— 对补码进行加法运算，再对结果求补码，即为最终结果；

易错：注意最后一步，对补码进行加法运算时，数字位最高位产生的进位要进到符号位上去！（教材中描述的“舍弃产生的进位”指的是舍弃符号位产生的进位）

几种常用的编码

编码的概念

广义的编码指的就是将信息从一种形式转换为另一种规范的形式；

对于数字电路，就是将现实世界中的各种物理信息转换为离散的电压序列，来表示、处理和分析信息；

事实上，前面学习的进位计数制也是一种编码，用不同进制数来表示“量”；本章的标题将数制与码制分开，首先单独讲解数制是因为数制的一个重要的特点是其编码后能够参与算术运算；而一般来说“码制”是对非数值信息的表示；

几种常用的编码

○ 常见十进制代码

十进制数	8421码	余3码	2421码	5211码	余3循环码
0	0000	0011	0000	0000	0010
1	0001	0100	0001	0001	0110
2	0010	0101	0010	0100	0111
3	0011	0110	0011	0101	0101
4	0100	0111	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1010	1101	1100	1111
8	1000	1011	1110	1101	1110
9	1001	1100	1111	1111	1010

十进制代码：用四位二进制数（最多16个）来表示 0~9；

8421、2421、5211
每个数字指的是每个位的权；

余3码是在 BCD码即8421码的基础上加 3，即用 3~12对应的二进制数来表示 0~9；

余3循环码是取格雷码（见下页）中的 3~12；

几种常用的编码

格雷码

教材中的格雷码特指的是四位格雷码，即用四位代码来表示 0 ~ 15 共十六种状态；

编码特点：

每一位的状态变化都按一定的顺序循环：

0110、00111100、0000111111110000、0000000011111111（四位16个，只循环半个周期）
（其他位数的格雷码表依次类推，自右向左每一位状态循环中连续的 0 和 1 数目加倍）

按表中顺序变化时，相邻的代码只有一位改变状态，这是格雷码最大的特点和优点；
（应用：减少代码转换过程中的过渡噪声）

几种常用的编码



四位格雷码表

编码顺序	四位二进制数	格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

必须熟练掌握！

重点是熟悉编码的规则和特点！

几种常用的编码

ASCII码

美国信息交换标准代码，用 7 位二进制代码（共128个）来表示信息，广泛应用于计算机和通信领域中；

关于编码一节，理解编码的概念，明白为什么要编码、编码用来做什么，了解常见的几种编码即可，第一次学习时不需要严格地死记硬背教材上列举的各种编码，后期继续学习过程中用到时再复习，重点是每种编码的编码规则和特点；

数制与码制——习题

本章习题小结

本章的习题考察形式一般为选择填空，题目较基础，但一些易错的细节也需要注意，重点是对各个基本概念的理解；

涉及知识点：

- 数制的概念（可能考察其他进制如三进制、五进制、六进制等）；
- 不同数制间的转换；
- 反码与补码的概念；
- 利用补码实现带符号数的加法运算（加减运算）

数制与码制——习题

例1：

现用三进制代码为电信学院的 400 名同学编制代号，则至少需要_____位编码；
如需要进一步在代号中表示来自哪个班级（假设一共 10 个班级），则还需要
增加_____位；

关键：对进位计数制与编码概念的理解！

数制与码制——习题

例2：完成以下数制转换

(1) $(01011110.10110010)_B \rightarrow H ?$

(2) $(8FAC6)_H \rightarrow O ?$

(3) $(173.8125)_D \rightarrow H ?$

数制与码制——习题

○ 例3：用二进制补码运算计算下列各式

(1) $13-10$

(2) $20-25$

(3) $-12-5$

(4) $-16-14$

养成解题习惯：

将二进制代码分为符号位和数字位！

注意首先考虑数字位的位数！

请尊重个人版权，请勿上传至其他公共平台，谢谢理解！
内容问题可联系 bow33@163.com



谢谢！