

# 计算机组成原理与系统结构

## 第三章 信息编码与数据表示

<http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/>





## 第三章 信息编码与数据表示

3.

数值数据的表示

3.

数据格式

3.3

定点机器数的表示

方法

3.4

浮点机器数的表示

方法

3.

非数值数据的表示

3.

校验码

3.7

现代计算机系统的数据表

示

本章小结

BACK



## 3.1 数值数据的表示



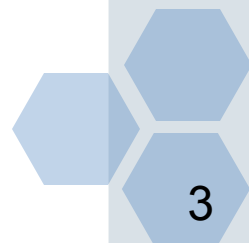
进位计数制



不同数制之间的相互转换



十进制数的编码





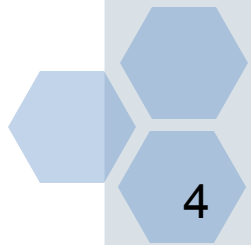
# 一、进位计数制

## ❖ 数制的两大要素：

- **基数  $R$** ：指在这种进位制中允许使用的基本数码个数。
- **权  $W_i$** ：权也称位权，指某一位  $i$  上的数码的权重值，即**权与数码所处的位置  $i$  有关**。  $W_i = R^i$ 。

## ❖ 基数为 $R$ 的数制称为 **$R$ 进制数**。

## ❖ $R$ 进制数的主要特点就是**逢 $R$ 进 1**。

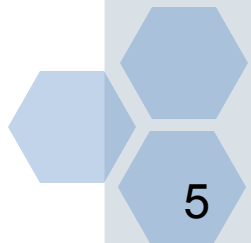




# 一、进位计数制

❖ 思考：何谓十进制、二进制、八进制、十六进制？

进制	基数 R	权 $W_i$	数码符号
十进制	$R=10$	$10^i$	$0 \sim 9$
二进制	$R=2$	$2^i$	0、1
八进制	$R=8$	$8^i$	$0 \sim 7$
十六进制	$R=16$	$16^i$	$0 \sim 9$ 、A ~ F





# 一、进位计数制

❖ 假设任意数值  $N$  用  $R$  进制数来表示，形式为：

$$N = (D_{m-1} D_{m-2} \cdots D_0 . D_{-1} D_{-2} \cdots D_{-k})_R$$

- 其中， $D_i$  为该进制的基本符号， $D_i \in [0, R-1]$ ， $i = -k, -k+1, \dots, m-1$ ；  
小数点在  $D_0$  和  $D_{-1}$  之间。

❖ 则数值  $N$  的**实际值**为：加权求和

$$N = \sum_{i=-k}^{m-1} (D_i \times R^i)$$

十进制

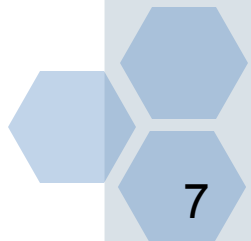


## 一、进位计数制

❖ 例 1 :  $(2345.459)_{10} = 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 9 \times 10^{-3}$

❖ 例 2 :  $(11011.011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^{-0} + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = (27.375)_{10}$

❖ 例 3 :  $(123.67)_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 6 \times 8^{-1} + 7 \times 8^{-2} = (83.859375)_{10}$





## 二、不同数制之间的相互转换

1

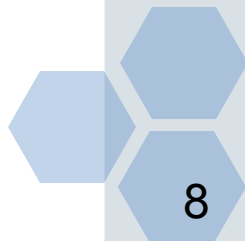
常用的几种数制的对应关系

2

二、八、十六进制转换为十进制

3

十进制转换为二、八、十六进制







# 1、常用的几种数制的对应关系

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	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F
				16	1000 0	20	10





## 2、二、八、十六进制转换为十进制

❖ 转换方法：加权求和。

$$N = \sum_{i=-k}^{m-1} (D_i \times R^i)$$

■ 例：  $(5AC.E6)_{16} = 5 \times 16^2 + 10 \times 16^1 + 12 \times 16^0 + 14 \times 16^{-1} + 6 \times 16^{-2} = (1452.8984375)_{10}$

❖ 十进制 (Decimal)、二进制 (Binary)、八进制 (Octal)、十六进制 (Hexadecimal) 数分别用 **D**、**B**、**Q**、**H** 来标志。

❖ 例如：  $(1011)_2 \rightarrow (1011)_B \rightarrow 1011B \rightarrow 1011b$

■  $(123.45)_{10} \rightarrow (123.45)_D \rightarrow 123.45D \rightarrow 123.45$

■  $(2B.D)_{16} = (2B.D)_H = (43.8125)_{10}$



### 3、十进制转换为二、八、十六进制

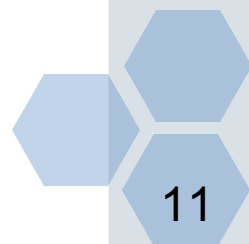
❖ **转换方法：**可以分为以下两种方法

- **直接转换：**十进制  $\rightarrow$  二、八、十六进制
- **间接转换：**十进制  $\rightarrow$  二进制  $\rightarrow$  八、十六进制



❖ (1) 十进制转化为R进制

❖ (2) 二进制转化为八、十六进制

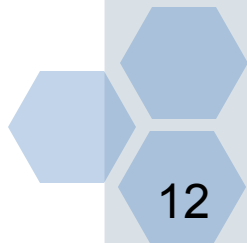




## (1) 十进制转化为 R 进制

### ❖ 转换方法

- **整数部分**：除以 R 取余，先得低位，直到商为 0。
- **小数部分**：乘 R 取整，先得高位，直到积为 0 或者达到精度要求为止。





# ( 1 ) 十进制转化为 R 进制

❖ 例：  $( 123.75 )_{10} = 1111011.11 ?$

❖ 整数部分

2	123	.....1	低位
2	61	.....1	
2	30	.....0	
2	15	.....1	
2	7	.....1	
2	3	.....1	
2	1	.....1	高位
	0		

❖ 小数部分

0.75	
× 2	
1.5	→ 0.5 .....1
	× 2
	1.0 .....1

高位  
↓  
低位

❖ 练习：  $( 123.75 )_{10} = 173.6 ?$

# 小数部分的精度要求

- ❖ 当小数部分不能整除为二进制时，则乘以 2 取整的过程中，积不会为 0；或者当小数部分转化为二进制位数很长，这时由精度来决定二进制位数。

- $(114.35)_{10} = (\quad ? \quad)_2$  无法整除

- $(0.6875)_{10} = (\quad ? \quad)_2$  位数太长

- ❖ 例： $(114.35)_{10} = (\quad ? \quad)_2$ ，要求精度大于 10%。

- 要求“=”左右两边的十进制值的差的绝对值 < 10%；因为  $10\% > 2^{-4}$ ，所以只需取 4 位二进制小数即可满足要求。

- $(114.35)_{10} = (1110010.0101)_2$

- ❖ 思考：若要求转换精度大  1%，则二进制小数



## (2) 二进制转化为八、十六进

### ❖ 二进制→八进制

- 以小数点为中心分别向两边分组，每三位一组，写出对应的八进制数字。（不够位数则在两边加 0 补足 3 位）

### ❖ 二进制→十六进制

001 011 111. 110 以小数点为中心分别向两边分组，每四位一组，写出对应的十六进制符号。（不够位数则在两边加 0 补足 4 位）

❖ 例：  $(1011111.11)_2 = 137.6_8$

$(1011111.11)_2 = (5F.C)_{16}$

0101 1111. 1100



- $$\begin{array}{r} \underline{111} \quad \underline{110} \quad \underline{101} \cdot \underline{010} \\ \underline{011} \end{array}$$

111   0110   0101.   0010  
0011





## 思考 2：计算机中为什么采用二进制表示数

- ❖ ① 具有二值状态的物理器件容易实现。
- ❖ ② 二进制数据的抗干扰性强，可靠性高。
- ❖ ③ 二进制的运算规则简单，硬件实现容易。
- ❖ ④ 具有逻辑特性，可代表“真假”、“是非”。





### 三、十进制数的编码

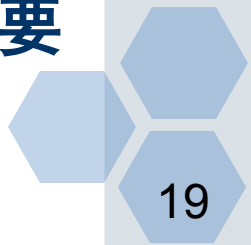
- ❖ 提出的问题：如何在计算机内使用二进制来表示十进制数据？
- ❖ 1、二—十进制码（BCD码）
- ❖ 2、十进制数串表示方法





# 1、二一十进制码（BCD 码）

- ❖ BCD（Binary Coded Decimal）码：使用二进制来编码十进制数字 0 ~ 9。
- ❖ **编码方法**：一般使用 4 位二进制编码来表示 1 位十进制数字，在 16 个编码中选用 10 个来表示数字 0 ~ 9。不同的选择构成不同的 BCD 码。
- ❖ **分类**：
  - **有权码**：编码的每一位都有固定的权值，加权求和的值即是表示的十进制数字。如 8421 码、2421 码、5211 码、4311 码、84 -2-1 码等。
  - **无权码**：编码的每一位并没有固定的权，主要包括格雷码、余 3 码等。





# 1、二一十进制码（BCD 码）

十进制数	8421 码	2421 码	5211 码	4311 码	84-2-1 码	格雷码	余3 码
0	0000	0000	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0111	0001	0100
2	0010	0010	0011	0011	0110	0011	0101
3	0011	0011	0101	0100	0101	0010	0110
4	0100	0100	0111	1000	0100	0110	0111
5	0101	1011	1000	0111	1011	1110	1000
6	0110	1100	1010	1011	1010	1010	1001
7	0111	1101	1100	1100	1001	1000	1010
8	1000	1110	1110	1110	1000	1100	1011
9	1001	1111	1111	1111	1111	0100	1100

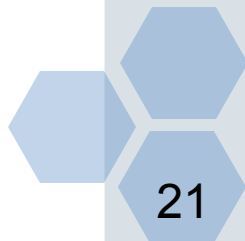


# 几种常见的 BCD 码

## ① 8421 码：

- 特点：4 位二进制数位的权从高到低依次是 8、4、2、1；8421 码实际上就是十进制数字 0 ~ 9 的二进制编码本身。
- 是最常用的一种 BCD 码，**在没有特别指出的一般情况下，所提到的 BCD 码通常就是指 8421 码。**

## ② 余 3 码：对应的 8421 码加上 0011 构成的。是一种自补码，即任何两个相加之和等于 9 的编码，互为反码 。

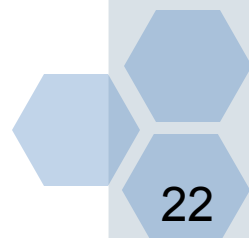




## 几种常见的 BCD 码

### ③ 格雷码：

- 特点：又叫循环码，它的任何相邻的两个编码（例如 2 和 3、7 和 8、9 和 0 等）之间只有一位二进制位不同。
- 优点：是用它构成计数器时，在从一个编码变到下一个编码时，只有一个触发器翻转即可，波形更完美、可靠。
- 格雷码的编码方案有许多种。





## 2、十进制数串的表示方法

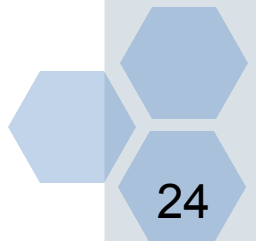
- ❖ **字符串形式：**用 ASCII 码来表示十进制数字或符号位，即 1 个字节存放 1 位十进制数字或符号位。
- ❖ **压缩的十进制数串形式：**用 BCD 码来表示十进制数字，即 1 个字节存放 2 个十进制的数字；符号位放在最低位数字位之后，一般用 C（12）表示正号，用 D（13）表示负号。
  - 例如 +258 被表示成 258CH，占用两个字节，-34 被表示为 034DH，也占用两个字节。
- ❖ **共同点：**必须给出它的主存中的首地址和位长。



## 2、十进制数串的表示方法

### ❖ 采用十进制表示数据的优点是：

- 对于需要大量地进行输入输出数据而运算简单的场合，大大**减少了二 - 十进制转换**，提高了机器的运行效率；
- 十进制数串的**位长可变**，许多机器中规定该长度从 0 到 31，有的甚至更长。不受定点数和浮点数统一格式的约束，从而提高了数据的表示范围和运算精度。







**The End !**