



计算机组成原理

计算机中数值数据的 表示方法

* 学习目标:

- * 1.了解: 进位计数制的概念, 二进制、八进制、十进制、十六进制等常见进制的表示。
- * 2.理解: 计算机中数值数据和非数值数据表示、溢出的判断、数据校验。
- * 3.掌握: 进制转换方法, 二进制、八进制、十进制、十六进制等常见进制之间的相互转换, 原码、反码和补码之间的相互转换以及运算。

计数制及其互相转换

数值数据的表示法

❖ 进位计数制及其转换

一、进位计数制：

用少量的数字符号（也称数码），按先后次序把它们排成数位，由低到高进行计数，计满进位，这样的方法称为**进位计数制**。

❖ **基数**：进位制的基本特征数，即所用到的数字符号个数。

例如10进制：0~9 十个数码表示，基数为10。

❖ **权**：进位制中，处于某一位上的“1”所表示的数值的大小为该位的权。各种进位计数制中的权值恰好是基数的某次幂。

❖ **常见的进位制**：2，8，10，16进制。

1、十进制(Decimal)

数码为：0~9；基数是10。

Decimal：十进制

运算规律：逢十进一，即： $9 + 1 = 10$ 。

十进制数的权展开式：

$$\begin{array}{r} 5 \times 10^3 = 5000 \\ 5 \times 10^2 = 500 \\ 5 \times 10^1 = 50 \\ 5 \times 10^0 = 5 \\ \hline = 5555 \end{array}$$

10^3 、 10^2 、 10^1 、 10^0 称为十进制的权。各数位的权是10的幂。

任意一个十进制数都可以表示为各个数位上的数码与其对应的权的乘积之和，称权展开式。

同样的数码在不同的数位上代表的数值不同。

即： $(5555)_D = 5 \times 10^3 + 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0$

例如：一个十进制数143.75的表示

$$(143.75)_{10} = 1 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

注：等式左边为并列表示法

等式右边为多项式表示法

2、二进制(Binary)

数码为：0、1；基数是2。

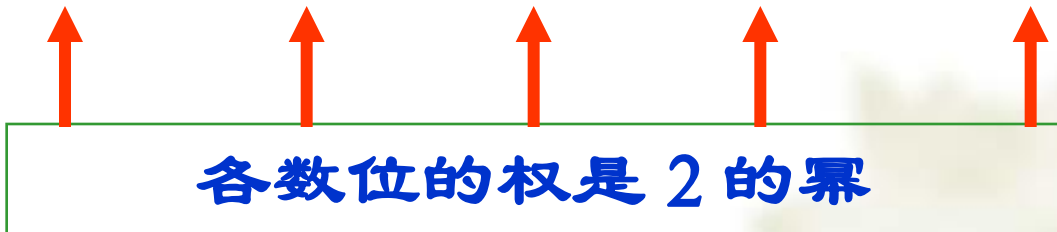
Binary：二进制

运算规律：逢二进一，即：1+1=10。

二进制数的权展开式：

$$\text{如：}(101.01)_B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$= (5.25)_{10}$



各数位的权是2的幂

二进制数只有0和1两个数码，它的每一位都可以用电子元件来实现，且运算规则简单，相应的运算电路也容易实现。

运算 规则

加法规则：0+0=0，0+1=1，1+0=1，1+1=10

乘法规则：0•0=0，0•1=0，1•0=0，1•1=1

例如:一个二进制数 $(1101.01)_2$ 的表示

$$\begin{aligned}(1101.01)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= (13.25)_{10}\end{aligned}$$

3. 二进制数的性质

移位性质：

↪ 小数点右移一位，数值扩大一倍

↪ 小数点左移一位，数值减小一半

$$1.1 = 2^0 + 2^{-1} = 1.5$$

$$11 = 2^1 + 2^0 = 2(2^0 + 2^{-1}) = 3$$

$$0.75 = 2^{-1}(2^0 + 2^{-1}) = 2^{-1} + 2^{-2} = 0.11$$

奇偶性质：

↪ 最低位为0，偶数

↪ 最低位为1，奇数

4. 二进制数的特点

优点：

- ✎ 只有0，1两个数码，易于用物理器件表示。
- ✎ 电位的高低，脉冲的有无，电路通断等都比较容易区别，可靠性高。
- ✎ 运算规则简单。
- ✎ 二进制的0，1与逻辑命题中的真假相对应，为计算机中实现逻辑运算和逻辑判断提供有利条件。

缺点：书写冗长，不易识别，不易发现错误

5、十六进制(Hexadecimal)

数码为：0~9、A~F；基数是16。

Hexadecimal：十六进制

运算规律：逢十六进一，即：F + 1 = 10。

十六进制数的权展开式：

$$\text{如：} (D8.A)_H = 13 \times 16^1 + 8 \times 16^0 + 10 \times 16^{-1} = (216.625)_{10}$$

各数位的权是16的幂

例如十六进制数 $(2A.7F)_{16}$ 的表示

$$\begin{aligned}(2A.7F)_{16} &= 2 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2} \\ &= 42.49609375\end{aligned}$$

6、八进制

数码为：0~7；基数是8。

0：八进制

运算规律：逢八进一，即：7 + 1 = 10。

八进制数的权展开式：

$$\text{如：} (207.04)_8 = 2 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 0 \times 8^{-1} + 4 \times 8^{-2}$$

$$= (135.0625)_{10}$$

各数位的权是8的幂

几种进制数之间的对应关系

十进制数	二进制数	八进制数	十六进制数
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F

进位计数制之间的转换

1、R进制转换成十进制的方法

按权展开法:先写成多项式,然后计算十进制结果。

$$\begin{aligned} N &= d_{n-1}d_{n-2}\cdots\cdots d_1d_0d_{-1}d_{-2}\cdots\cdots d_{-m} \\ &= d_{n-1} \times R^{n-1} + d_{n-2} \times R^{n-2} + \cdots\cdots\cdots d_1 \times R^1 + d_0 \times R^0 \\ &\quad + d_{-1} \times R^{-1} + d_{-2} \times R^{-2} + \cdots\cdots\cdots d_{-m} \times R^{-m} \end{aligned}$$

例如：写出 $(1101.0101)_2$ ， $(237)_8$ ， $(10D)_{16}$ 的十进制数。

$$\begin{aligned} \curvearrowright (1101.0101)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &\quad + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 8 + 4 + 1 + 0.25 + 0.0625 = 13.3125 \end{aligned}$$

$$\curvearrowright (237)_8 = 2 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 = 128 + 24 + 7 = 159$$

$$\curvearrowright (10D)_{16} = 1 \times 16^2 + 13 \times 16^0 = 256 + 13 = 269$$

练习

❖ $(11011.1)_2 = ?$

❖ $(110011.101)_2 = ?$

❖ $(1101.1100)_2 = ?$

练习

$$\diamond (11011.1)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 27.5$$

$$\diamond (110011.101)_2 = 2^5 + 2^4 + 2^1 + 2^0 + 2^{-1} + 2^{-3} \\ = 32 + 16 + 2 + 1 + 0.5 + 0.125 = (51.625)_{10}$$

$$(1101.1100)_2 = (1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_{10} \\ + (1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 0 \times 2^{-4})_{10} = (13.75)_{10}$$

2、十进制转换成二进制方法

一般分为两个步骤：

❧ 整数部分的转换

❖ 除2取余法（基数除法）

❖ 减权定位法

❧ 小数部分的转换

❖ 乘2取整法（基数乘法）

除基取余法：把给定的十进制整数除以基数，取余数作为最低位的系数，然后继续将商部分除以基数，余数作为次低位系数，重复操作直至商为0。

$$\begin{array}{r} 2 \overline{) 25} \dots \dots \text{余 } 1 \\ 2 \overline{) 12} \dots \dots \text{余 } 0 \\ 2 \overline{) 6} \dots \dots \text{余 } 0 \\ 2 \overline{) 3} \dots \dots \text{余 } 1 \\ 2 \overline{) 1} \dots \dots \text{余 } 1 \\ 0 \end{array}$$

除基数
得余数
作系数
从低位
到高位

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

问题： $(25.8125)_{10} = (?)_2$

$(0.8125)_{10} = (0.1101)_2$

0.8125	2	取整
\times	1.6250	1
	0.6250	
\times	1.2500	1
	0.2500	
\times	0.5000	0
\times	1.0000	1

若小数在连乘多次后不为 0，一般按照精确度要求 (如小数点后保留 n 位) 得到 n 个对应位的系数即可。

乘基数
取整数
作系数
从高位
到低位

练习

❖ 【例2-2】 将十进制数0.375转换成二进制数。

解：把0.375不断地乘2，取每次所得乘积的整数部分，余下的小数部分继续乘2，直到乘积的小数部分为0。

练习

❖ 【例2-2】 将十进制数0.375转换成二进制数。

解：把0.375不断地乘2，取每次所得乘积的整数部分，余下的小数部分继续乘2，直到乘积的小数部分为0，其过程如下：

0.375		
× 2		

0.750	取整数部分：0
0.750		
× 2		

1.500	取整数部分：1
0.500		
× 2		

1.000	取整数部分：1

最高位
↓
最低位

把所得整数按箭头方向从高到低排列后得到：0.375=0.011B ≠ 011B

练习

【例2-1】 将十进制数129转换成二进制数。

练习

【例2-1】将十进制数129转换成二进制数。

解：把 129 连续除以 2，直到商数为 0，余数小于 2，其过程如下：

2		129	余 1	↑ 最低位
2		64	余 0	
2		32	余 0	
2		16	余 0	
2		8	余 0	
2		4	余 0	
2		2	余 0	
2		1	余 1	
		0			最高位

把所得余数按箭头方向从高到低排列起来便可得到：129 = 10000001B

← 3. 十六进制和十进制数间的转换

(1) 十六进制数转换成十进制数

方法和二进制数转换成十进制数的方法类似，即把十六进制数按权展开后相加。

(2) 十进制数转换成十六进制数

① 十进制整数转换成十六进制整数采用“除16取余法”，即除16取余，后余先排。

【例2-3】 将十进制数3938转换成十六进制数。

解：把3938连续除以16，直到商数为0，余数小于16，其过程如下：

$$\begin{array}{r} 16 \overline{) 3938} \dots\dots\dots \text{余 } 2 \\ 16 \overline{) 246} \dots\dots\dots \text{余 } 6 \\ 16 \overline{) 15} \dots\dots\dots \text{余 } 15 \text{ (F)} \\ \quad 0 \end{array} \quad \begin{array}{l} \uparrow \text{最低位} \\ \uparrow \text{最高位} \end{array}$$

即得：3938=F62H ≠ 1562H

注意：上式中的余数 15 最后应写成 F。

❖ ②十进制小数转换成十六进制小数采用“乘16取整法”：乘16取整，整数顺排，直到所得乘积的小数部分为0或达到所需精度为止。

【例2-4】将十进制数0.566743转换成十六进制数。（小数点后取3位有效数字）

解：把0.566743连续乘以16，直到所得乘积的小数部分达到所需精度为止，其过程如下：

【例2-4】 将十进制数0.566743转换成十六进制数。（小数点后取3位有效数字）

解：把0.566743连续乘以16，直到所得乘积的小数部分达到所需精度为止，其过程如下：

0.566743		
× 16		
9.067888	取整数部分：9
0.067888		
× 16		
1.086208	取整数部分：1
0.086208		
× 16		
1.379328	取整数部分：1

} 最高位
} 最低位

即得：0.566743 ≈ 0.911H ≠ 911H

注意：别忘了加小数点。

3、其它进制之间的直接转换法

二进制数与八进制数的相互转换

(1) 二进制数转换为八进制数: 将二进制数由小数点开始, 整数部分向左, 小数部分向右, 每3位分成一组, 不够3位补零, 则每组二进制数便是一位八进制数。

$$001 \mid 101 \mid 010 \mid . 010 = (152.2)_8$$

(2) 八进制数转换为二进制数: 将每位八进制数用3位二进制数表示。

$$(374.26)_8 = 011 \ 111 \ 100 . 010 \ 110$$

二进制转换成八进制

例： $(10110111.01101)_2$

二进制： 10 , 110 , 111 . 011 , 01

二进制： 010 , 110 , 111 . 011 , 010

八进制： 2 6 7 . 3 2

$(10110111.01101)_2 = (267.32)_8$

八进制转换二进制

例如: $(123.46)_8$

$=(001,010,011 . 100,110)_2$

$=(1010011.10011)_2$

二进制数与十六进制数的相互转换

二进制数与十六进制数的相互转换，按照每4位二进制数对应于一位十六进制数进行转换。

$$0001 \mid 1101 \mid 0100 \mid . 0110 = (1D4.6)_{16}$$

$$(AF4.76)_{16} = 1010 \ 1111 \ 0100 . 0111 \ 0110$$

二进制转换成十六进制

例：(110110111.01101)₂

二进制： 1,1011,0111.0110,1

二进制： 0001,1011,0111.0110,1000

十六进制： 1 B 7. 6 8

$(10110111.01101)_2 = (1B7.68)_{16}$

练习

【例2-5】 将10101100110101.1010010111001B转换成十六进制数。

练习

【例2-5】 将10101100110101.1010010111001B转换成十六进制数。

解：

0010	1011	0011	<u>0101</u>	.	1010	0101	1100	1000	B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
2	B	3	5	.	A	5	C	8	H

即得：10101100110101.1010010111001B=2B35.A5C8H

注意：别忘了在整数部分和小数部分之间加小数点。

十六进制转换成二进制

例如: $(7AC.DE)_{16}$

$=(0111, 1010, 1100.1101, 1110)_2$

$=(11110101100.1101111)_2$

练习

【例2-6】 将十六进制数1A7.4C5H转换成二进制数。

练习

【例2-6】 将十六进制数1A7.4C5H转换成二进制数。

解：

1	A		7		.		4	C		5		H
↓		↓		↓		↓		↓		↓		↓		↓
<u>0001</u>		<u>1010</u>		0111		.		<u>0100</u>		<u>1100</u>		0101		B

即得：1A7.4C5H = 000110100111.010011000101B

注意：别忘了在整数部分和小数部分之间加小数点。

❖ 十六进制转换为八进制（或八进制转换为十六进制）时，先将十六进制（八进制）转换为二进制，然后由二进制转换为八进制（十六进制）。

2、有权码和无权码

有权码

- ❖ 表示一位十进制数的二进制码的每一位有确定的权。一般用8421码，其4个二进制码的权从高到低分别为8、4、2和1。用0000，0001，...，1001分别表示0，1，...，9，每个数位内部满足二进制规则，而数位之间满足十进制规则，故称这种编码为“以二进制编码的十进制(binary coded decimal, 简称BCD)码”。

表 4位有权码

十进制数	8421码	2421码	5211码	4311码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 1	0 0 1 1
3	0 0 1 1	0 0 1 1	0 1 0 1	0 1 0 0
4	0 1 0 0	0 1 0 0	0 1 1 1	1 0 0 0
5	0 1 0 1	1 0 1 1	1 0 0 0	0 1 1 1
6	0 1 1 0	1 1 0 0	1 0 1 0	1 0 1 1
7	0 1 1 1	1 1 0 1	1 1 0 0	1 1 0 0
8	1 0 0 0	1 1 1 0	1 1 1 0	1 1 1 0
9	1 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1

无权码

- ❖ 表示一个十进制数位的二进制码的每一位没有确定的权。用得较多的是余3码(Excess-3 Code)和格雷码(Gray Code), 格雷码又称“循环码”。
- ❖ 余3码是在8421码基础上, 把每个编码都加上0011而形成的(见表3.3), 其运算规则是:
- ❖ 当两个余3码相加不产生进位时, 应从结果中减去0011; 产生进位时, 应将进位信号送入高位, 本位加0011。

表 4位无权码

十进制数	余3码	格雷码(1)	格雷码(2)
0	0011	0000	0000
1	0100	0001	0100
2	0101	0011	0110
3	0110	0010	0010
4	0111	0110	1010
5	1000	1110	1011
6	1001	1010	0011
7	1010	1000	0001
8	1011	1100	1001
9	1100	0100	1000