L \ H	0000	0001	0010	0011	0100	0101	0110	0111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	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	VT	ESC	+	;	K	[	k	{
1100	FF	FS	’	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

比特与进制

十进制数

- 使用十个不同的数字符号（0,1,2,3,4,5,6,7,8,9），基数是 10，逢10进1
- 这些数字符号处于十进制数中不同位置时，其权值各不相同，十进制数各位的权值是10的整数次幂
- 十进制数的标志： 尾部加“D”或缺省

$$2004.96D = 2\times10^3 + 0\times10^2 + 0\times10^1 + 4\times10^0 + 9\times10^{-1} + 6\times10^{-2}$$

八进制数

- 使用0、1、2、3、4、5、6、7 八个数字符号表示，基数为8，逢8进1
- 八进制数各位的权值是8的整数次幂
- 八进制数的标志：尾部加Q

$$365.2Q = 3\times8^{2}+6\times8^{1}+5\times8^{0}+2\times8^{-1} = 245.25$$

十六进制数

- 使用0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F十六个符号表示，其中A、B、C、D、E、F分别代表十进制的10、11、12、13、14、15，基数为16，逢16进1
- 十六进制数各位的权值是16的整数次幂
- 十六进制数的标志：尾部加H

$$F5.4H = 15\times16^{1}+5\times16^{0}+4\times16^{-1} = 245.25$$

任意（R）进制数

- 每种进位制都有固定的数码——基数
- 按基数进位或借位——逢R进一
- 位权与基数的关系：位权的值等于基数的若干次幂。

$$(K_nK_{n-1}\ldots K_1K_0 . K_{-1}K_{-2}\ldots K_{-m})_R$$
$$= K_n\times R^n+K_{n-1}\times R^{n-1}+\ldots+K_1\times R^1+K_0\times R^0$$
$$+K_{-1}\times R^{-1}+K_{-2}\times R^{-2}+\ldots+K_{-m}\times R^{-m}$$

不同进制数的相互转换

- 熟练掌握不同进制数相互之间的转换，在编写程序和设计数字逻辑电路时很有用

- 只要学会二进制数与八进制、十六进制数、十进制数之间的转换，其他进制的转换一样引刃而解

十进制数与二进制数的相互转换

- 二进制数→十进制数

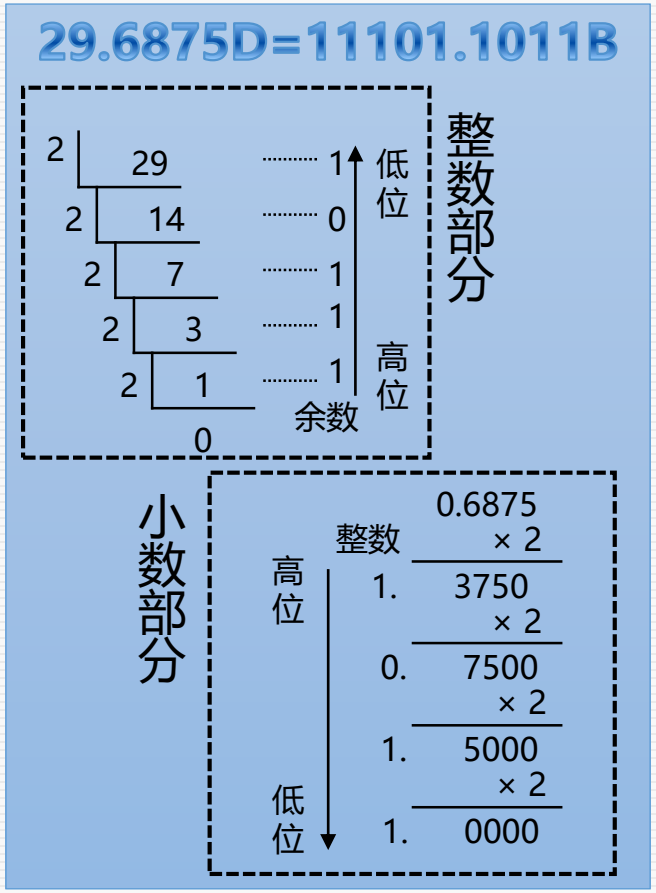
  按位权展开

- 十进制数→二进制数

  整数部分：除2取余，结果倒写

  小数部分：乘2取整，结果顺写

注意转换可能存在的误差问题



二进制数与十六进制数的相互转换

∵ $16=2^4$

∴ 采用1-4组合法

5A2.FC H → 0101 1010 0010.1111 1100 B

0011 0100 1110.1100 1100 B → 34E.CC H

十六进制	0	1	2	3	4	5	6	7
二进制	0000	0001	0010	0011	0100	0101	0110	0111
十六进制	8	9	A	B	C	D	E	F
二进制	1000	1001	1010	1011	1100	1101	1110	1111

• 整数的编码表示

- 整数的特点

  不使用小数点，或者认为小数点固定隐含在个位数的右面，因此整数也叫做“定点数”。

- 整数的分类
  - 无符号的整数 (unsigned integer)

    正整数。可用于表示字符编码、地址码、索引码等
  - 带符号的整数(signed integer)

    正整数或负整数。可用于表示某个物理量的值

• 不带符号整数的编码表示

- 用一个机器数表示一个不带符号的整数。

  其取值范围由机器数的位数决定

  最小值：00000000……B，最大值：11111111……B

8位：	可表示0～255(2^8 - 1)范围内的所有正整数
16位：	可表示0～65535(2^{16} - 1)范围内的所有正整数
n位：	可表示0～2^n - 1范围内的所有正整数

注意由于超出机器数范围而导致的溢出（overflow）问题

带符号整数的编码表示

- **“原码”**编码方法：机器数的最高位表示整数的符号（0代表正数，1代表负数），机器数的剩余位以二进制形式表示数据的绝对值。
- 表示范围：

8位：$-2^7+1 \sim 2^7-1$(-127～127)范围内所有整数

16位：$-2^{15}+1 \sim 2^{15}-1$(-32767～32767)范围内所有整数

n位：$-2^{n-1}+1 \sim 2^{n-1}-1$范围内所有整数

- 原码举例（8位原码）：

$[+43]_{原码} = 00101011$

$[-34]_{原码} = 10100010$

• 带符号整数的编码表示

- 原码表示的优点：

  与日常使用的表示方法比较一致，简单、直观

- 原码表示的缺点：

  – 加法运算与减法运算的规则不统一，减法运算烦琐，实现电路复杂，增加了成本

  – 整数0有“00000000”和“10000000”两种表示形式

计算机内部通常不采用“原码”
而采用“补码”表示带符号整数

• 带符号整数的编码表示

- **“补码”**编码方法：正整数的补码与其原码形式相同；负整数的补码等于其原码除最高符号位保持不变外，其它每一位取反，再在末位加“1”后所得到的运算结果。
- 补码举例（8位原码）：

$[+43]_{补码} = [+43]_{原码} = 00101011$

$[-34]_{原码} = 10100010$

$[-34]_{反码} = 11011101$

$[-34]_{补码} = 11011110$

• 带符号整数的编码表示

- 补码运算规则：

$$[X \pm Y]_{原码} = [\,[X]_{补码} + [\pm Y]_{补码}\,]_{补码}$$

- 补码表示范围：

最小值：10000000……B，最大值：01111111……B

8位：$-2^7 \sim 2^7-1$ (-128～127)范围内的所有整数

n位：$-2^{n-1} \sim 2^{n-1}-1$ 范围内的所有整数

带符号整数的编码表示

- 补码的优点：
  - 能将减法运算转换为加法运算，便于CPU作运算处理
  - n位原码表示整数“0”时，有“1000…00”与“0000…00”两种形式，而在补码表示中，整数“0”只有“0 000…00”一种表示形式，“1000…00”则用来表示整数值-2^{n-1}，因而，若原码和补码的表示位数相同，补码可表示整数的个数比原码多一个。（例，8位补码能表示-128，8位原码则不能）
- 补码的缺点：不直观

• 实数的编码表示

- 实数的特征

  实数也叫做“浮点数”，包含小数点，既有整数部分又有小数部分。整数和纯小数是实数的特例。

- 实数的编码表示（浮点表示法）

  任何一个实数总可以表达成一个乘幂和一个纯小数之积。乘幂中的指数部分用来指出实数中小数点的位置，纯小数部分决定了有效数字。

  例如：$56.72=0.5672\times10^{2}$； $-0.00347=-0.347\times10^{-2}$

• 实数的编码表示

- 二进制实数：

  $1001.011B = 0.1001011B \times 2^{100}B$

  $-0.0010101B = -0.10101B \times 2^{-10}B$

- 任一个二进制实数 N 均可表示为：

  $N = \pm S \times 2^{\pm P}$ （0 < S < 1时为规范化表示）

  S：称为 N 的**尾数**（纯小数）

  ±P：称为 N 的**阶码**（有符号整数）

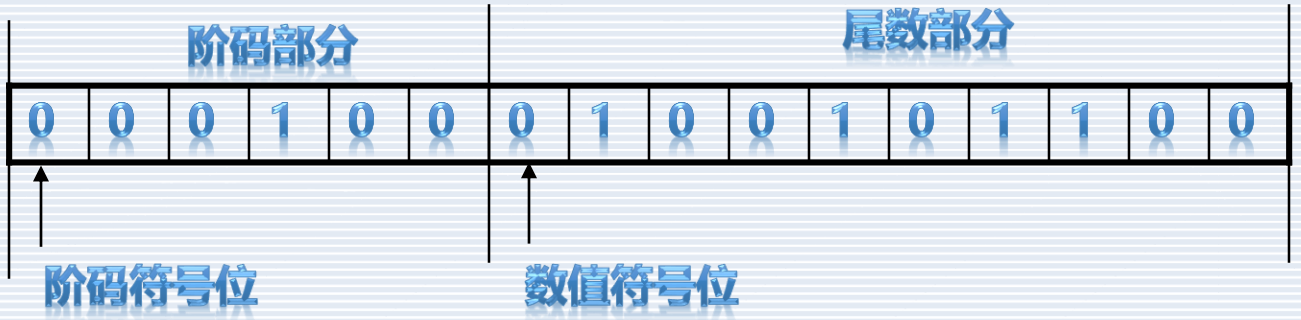
• 实数的编码表示

- 二进制实数的编码

  将“阶码”和“尾数”进行编码，合在一起即为实数编码。这种表示法称为“浮点表示法”。

  - 阶码是有符号整数，可采用原码或补码表示
  - 尾数的数值范围和小数点的位置可以有不同的约定

  例如， $1001.011B = 0.1001011B \times 2^{+100B}$



• 实数的编码表示

浮点数的长度 = 阶码的编码位数 + 尾数的位数

浮点数的长度：一个或多个机器数（例如：32位、64位等）

一般来说，

阶码位数越多，可表示的实数的范围越大

尾数位数越多，可表示的实数的精度越高

举例：16位机器数能够表示的实数的范围：

011111 1111111111——011111 0111111111（原码）

最小值　　最大值

$-(1-2^{-9})\times 2^{2^5-1}$ —— $(1-2^{-9})\times 2^{2^5-1}$

• 实数的编码表示

Pentium处理器32位浮点数的表示格式(工业标准 IEEE 754)

包括三个组成部分:

- 符号s：**1位**，s=0表示此数为正数，s=1表示为负数
- 偏移阶码(e)：**8位**，e = 指数 + 127

  带有偏移量127的无符号整数
- 尾数(f)：**23位**，原码表示，绝对值在1与2之间，其中1和小数点都是隐含的，不直接表示出来

1位	8位	23位
符号位s	偏移阶码e	尾数 f ($b_1b_2b_3...b_{23}$)