



数字逻辑

第一章 信息的表示

北京理工大学计算机学院

提纲

1. 常用数制及转换
2. 数值数据的表示
3. 字符数据的表示
4. 其它编码及应用
5. 现代典型计算系统中的数据表示

1.1 数制

■ 数的表示规则称为数制

■ 基底 (r) : 一个数制所包含的数字符号的个数

二进制	0, 1
八进制	0, 1, 2, 3, 4, 5, 6, 7
十进制	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
十六进制	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

■ 权 (r^i) : 数字符号的位置所决定的值

数字

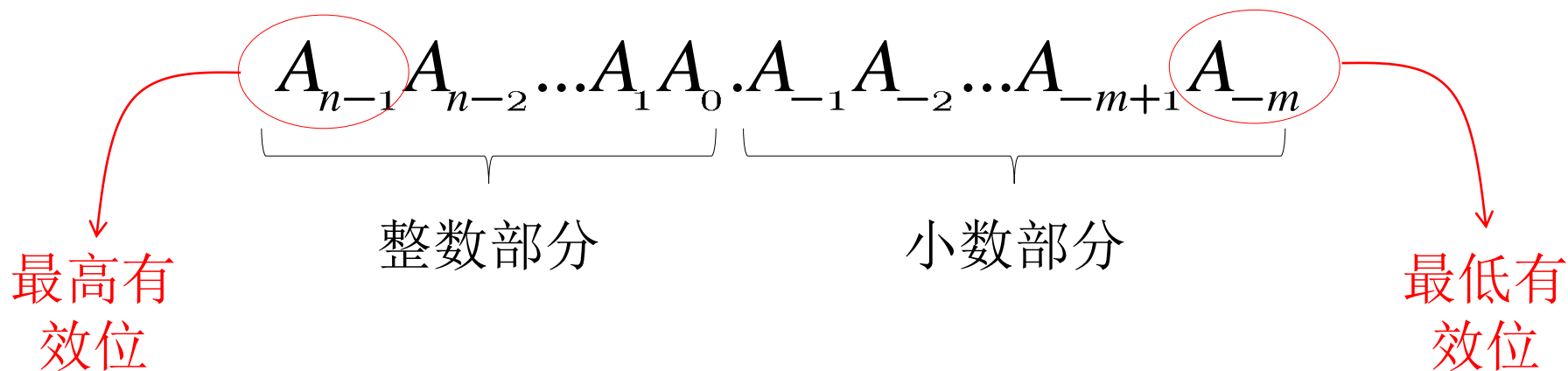
权

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

1.1 数制

■ 按位计数制

- 任何一个数值，都是各位数字本身的值与其权之积的总和，是一种按位计数制

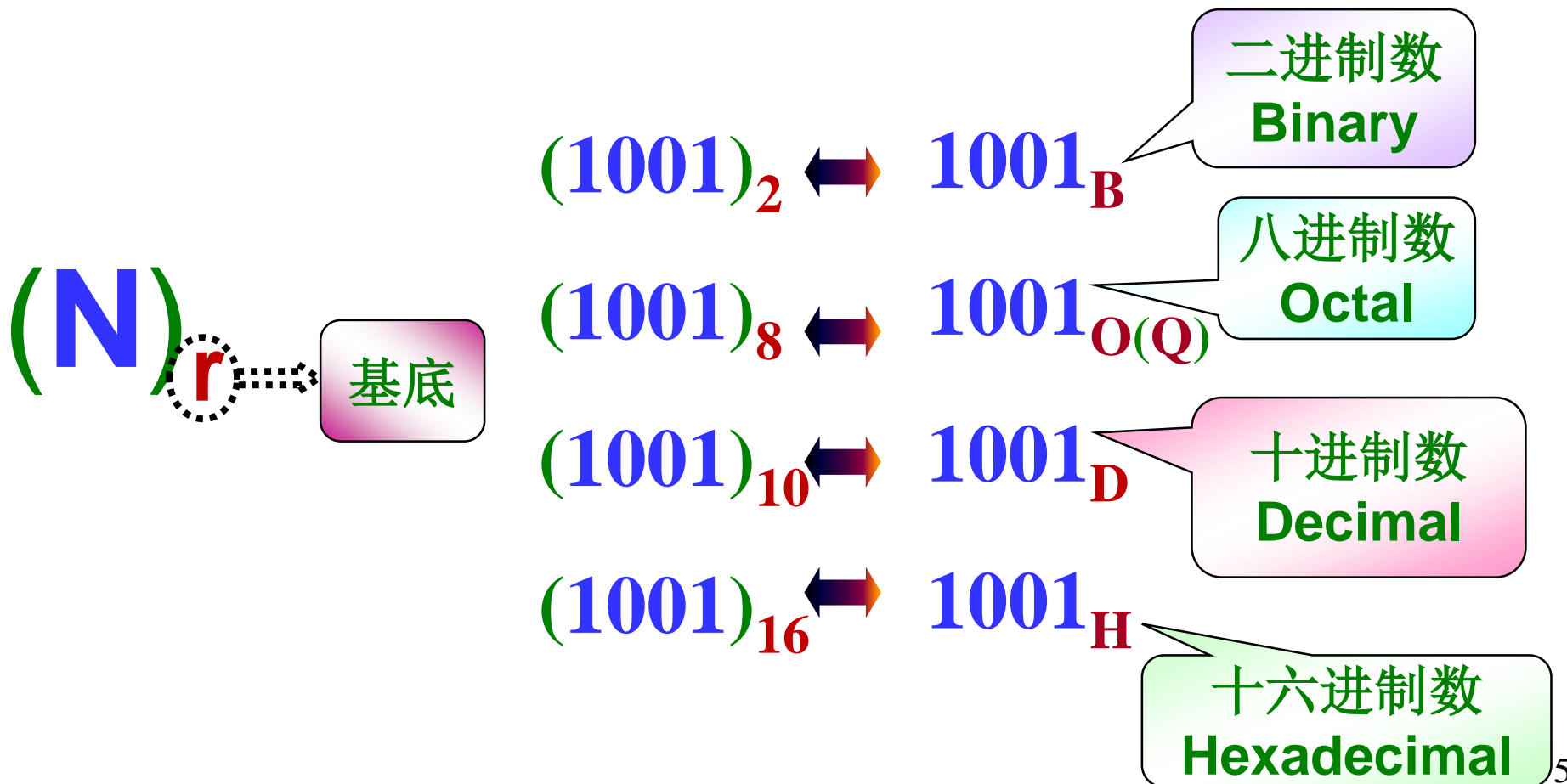


$$(number)_r = \left(\sum_{i=0}^{i=n-1} A_i \cdot r^i \right) + \left(\sum_{j=-m}^{j=-1} A_j \cdot r^j \right)$$

1.1 数制

■ 常见的按位计数制表示

- 二进制、八进制、十进制、十六进制



1.1 数制

■ 二进制

- 基底为2
- 逢二进一

$$(110.11)_2$$

$$=1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

$$=6.75$$

2^{10} (1024) 千, 记为 "K"

2^{20} (1,048,576) 兆, 记为 "M"

2^{30} (1,073, 741,824) 十亿, 记为 "G"

2^{40} (1,099,511,627,776) 万亿, 记为 "T"

1.1 数制

■ 八进制

- 基底为8
- 逢八进一
- 1个八进制位相当于3个二进制位

$$(702.41)_8$$

$$\begin{aligned} &= 7 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} + 1 \times 8^{-2} \\ &= 450.515625 \end{aligned}$$

1.1 数制

■ 十六进制

- 基底为16
- 逢十六进一
- 1个十六进制位相当于4个二进制位

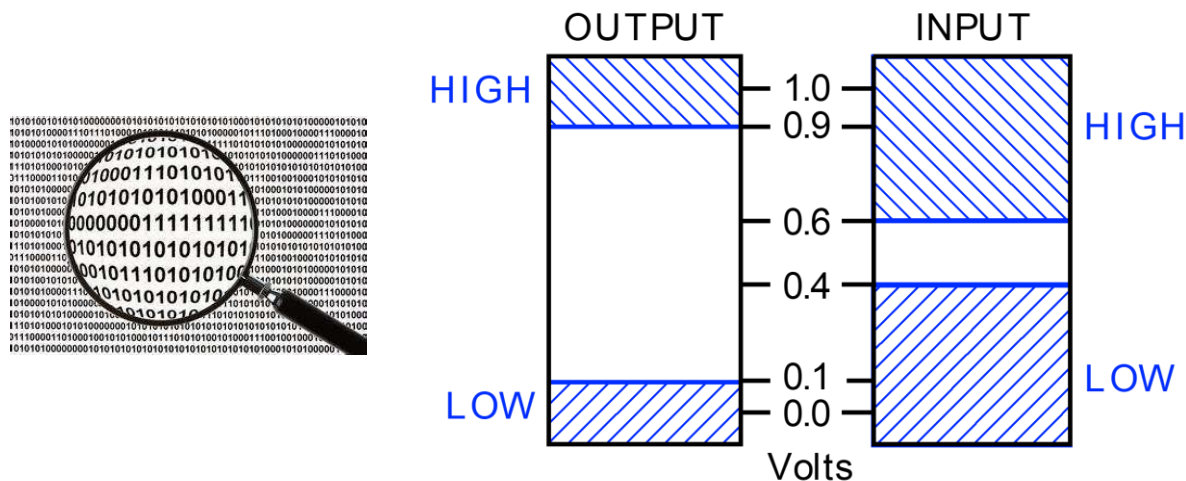
$(70F.4A)_{16}$

$$= 7 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 + 4 \times 16^{-1} + 10 \times 16^{-2}$$

$$= 1807.2890625$$

1.1 数制

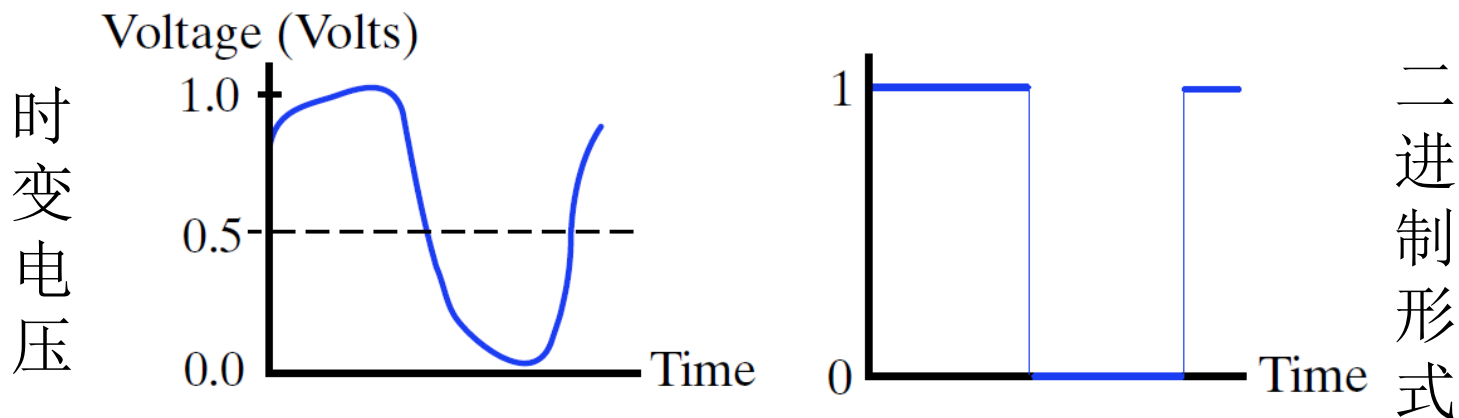
- 数字系统的信号广泛采用两个离散值的二进制
- 数字“0”和“1”
- 文字（符号）“真”（T）和“假”（F）
- 文字（符号）“高”（H）和“低”（L）
- 文字（符号）“开”（On）和“关”（Off）



1.1 数制

■ 数字系统的二进制实现

- 通过特定物理量（电压）的取值进行按位设置



- 一个二进制数字称为一位（bit）
- 数字系统的信息是通过一组一组的位来表示

0 0 1 1 1 1 1 0 1 0

1.1 数制

■ 思考题

$(45.6)_{10}$ 和 $(52.1)_8$ 两数谁大?



$=42.125$

$$\begin{array}{c} (\quad 4 \quad 5 \quad . \quad 6 \quad)_{10} \\ \downarrow \quad \downarrow \quad \downarrow \\ 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} \end{array}$$

$$\begin{array}{c} (\quad 5 \quad 2 \quad . \quad 1 \quad)_8 \\ \downarrow \quad \downarrow \quad \downarrow \\ 5 \times 8^1 + 2 \times 8^0 + 1 \times 8^{-1} \end{array}$$

1.2 数制转换

■ 四种进制关系

■ 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	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	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
16	10000	20	10

1.2 数制转换

■ 非十进制数转换为十进制

按权展开

- 二进制 \Rightarrow 十进制 $(101.101)_2 = (5.625)_{10}$
- 八进制 \Rightarrow 十进制 $(304.6)_8 = (196.75)_{10}$
- 十六进制 \Rightarrow 十进制 $(5CA)_{16} = (1482)_{10}$

1.2 数制转换

■ 十进制数转换为非十进制

十进制数分为两类：整数
小数

■ 十进制整数 \Leftrightarrow 二进制

$$(215)_{10} = (11010111)_2$$

除二取余

■ 十进制小数 \Leftrightarrow 二进制

$$(0.6875)_{10} = (0.1011)_2$$

乘二取整

1.2.2 数制转换

■ 二-八-十六进制数间的转换

■ 二进制数 \rightleftharpoons 八进制数

$$2^3 = 8$$

$$(\underline{111}\underline{010}\underline{100}\underline{11}.\underline{101}\underline{11})_2 = (3523.56)_8$$

以小数点为界，分别向左、向右每三位一组进行分割，不足三位补0。写出每三位对应的八进制数。

■ 八进制数 \rightleftharpoons 二进制数

$$(3740.562)_8 = (111111000000.10111001)_2$$

1.2.2 数制转换

■ 二-八-十六进制数间的转换

■ 二进制数 \rightleftharpoons 十六进制数

$$2^4 = 16$$

$$(1111\mathbf{0101}0011.\mathbf{1011})_2 = (\text{F53.B8})_{16}$$

以小数点为界，分别向左、向右每四位一组进行分割，不足四位补0。写出每四位对应的十六进制数。

■ 十六进制数 \rightleftharpoons 二进制数

$$(\mathbf{2AF.C5})_{16} = (1010101111.11000101)_2$$

1.2.3 基本算数运算

■ 二进制加法

■ X: 被加数; Y: 加数; Z: 进位

Z	0 0 0 0 0	1 0 1 1 0 0
X	0 1 1 0 0	1 0 1 1 0
Y	+ 1 0 0 0 1	+ 1 0 1 1 1
和	1 1 1 0 1	1 0 1 1 0 1

和大于1时向高位产生进位

1.2.3 基本算数运算

■ 二进制减法

■ X: 被减数; Y: 减数; Z: 借位

Z	0 0 0 0 0	0 0 1 1 0
X	1 0 1 1 0	1 0 1 1 0
Y	- 1 0 0 1 0	- 1 0 0 1 1
差	0 0 1 0 0	0 0 0 1 1

差小于0时向高位产生借位

1.2.3 基本算数运算

■ 二进制乘法

■ $0 \times 0 = 0$ 、 $1 \times 0 = 0$ 、 $0 \times 1 = 0$ 、 $1 \times 1 = 1$

$$\begin{array}{r} \text{被乘数:} \quad 1\ 0\ 1\ 1 \\ \text{乘数:} \quad \times \quad 1\ 0\ 1 \\ \hline \quad \quad 1\ 0\ 1\ 1 \\ \quad 0\ 0\ 0\ 0 \\ 1\ 0\ 1\ 1 \\ \hline \text{积:} \quad 1\ 1\ 0\ 1\ 1\ 1 \end{array}$$

提纲

1. 常用数制及转换
- 2. 数值数据的表示**
3. 字符数据的表示
4. 其它编码及应用
5. 现代典型计算系统中的数据表示

2.1 二进制编码

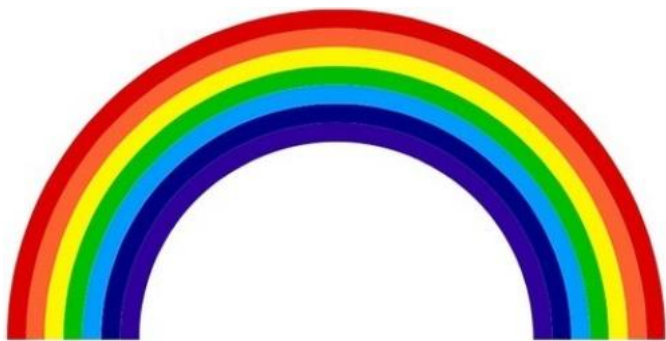
- 通过0、1排列的组合方式表示数据
- n 位二进制编码可以表示 2^n 个数据
- 数据类型
 - 数值型：能够进行算术运算的数据
 - 整数、小数、...
 - 非数值型：一般不需要算术运算的数据
 - 字符、控制符、音频、图像、视频、...



2.1 二进制编码

■ 思考题

- 七彩色需要几位二进制编码？



颜色	二进制
红	000
橙	001
黄	010
绿	011
蓝	101
靛	110
紫	111

M个数据需要的二进制位数 $\lceil \log_2 M \rceil$ 。

2.2 数值数据的二进制编码

■ 2.2.1 数值数据的类型

- 整数
- 小数

	正数	0	负数
整数	5	0	-5
小数	5.25	0.0	-5.25

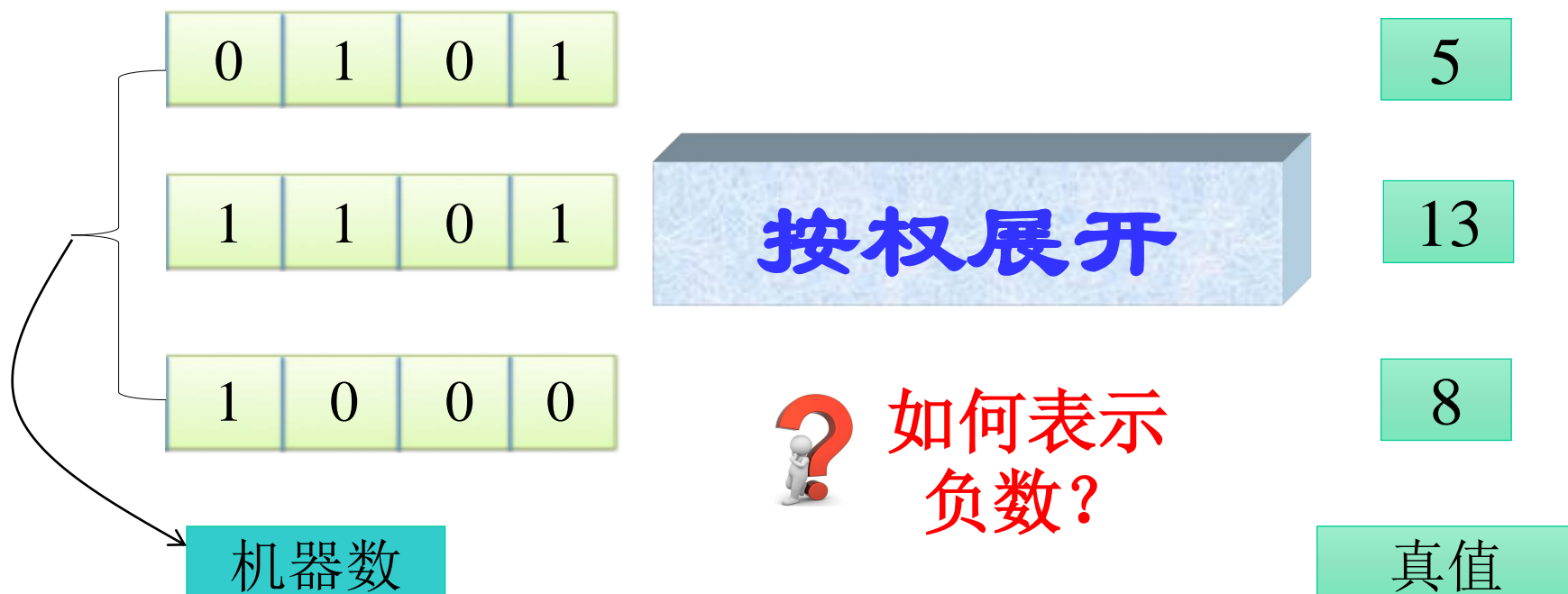
真值：二进制对应的真实数值

2.2 数值数据的二进制编码

■ 2.2.1 数值数据的类型

■ 机器数：整数或小数的二进制编码表示

■ 例如，四位二进制编码



2.2 数值数据的二进制编码

■ 2.2.1 数值数据的类型

- 符号数：最高位0表示正数；1表示负数
- 无符号数：每一位都作为权

0	1	0	1	5
---	---	---	---	---

1	1	0	1	-5
---	---	---	---	----

1	0	0	0	-0
---	---	---	---	----

符号数：符号-数值

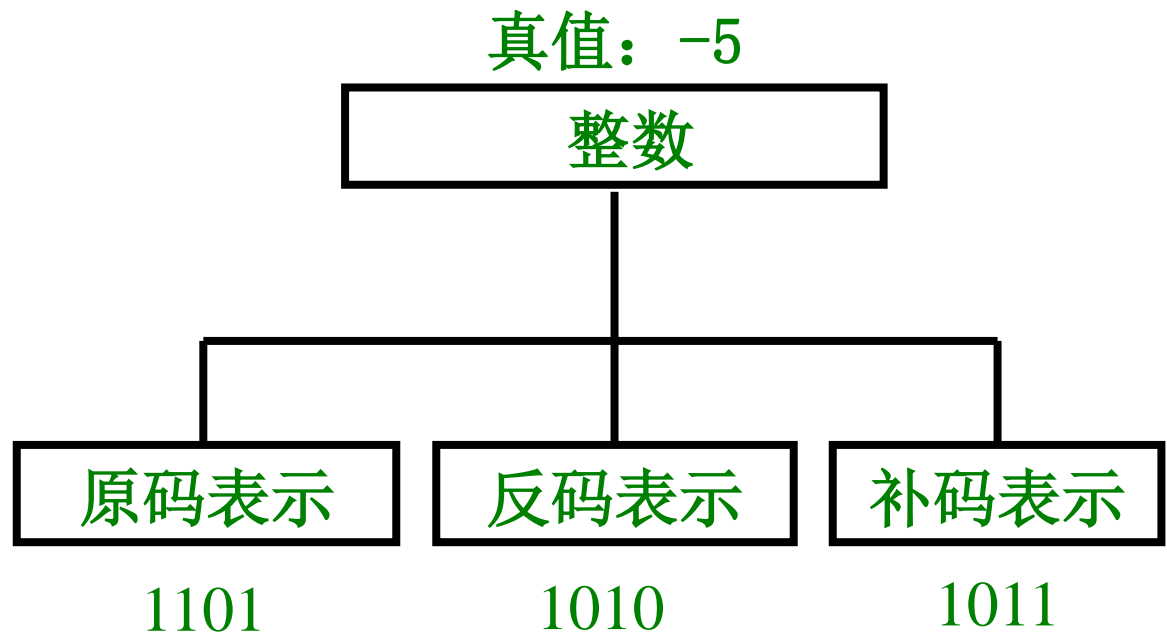
十进制	符号-数值
+3	0011
+2	0010
+1	0001
+0	0000
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100

2.2 数值数据的二进制编码

■ 2.2.2 整数的二进制编码

■ （无）符号数的三种编码形式

- 原码
- 反码
- 补码



2.2 数值数据的二进制编码

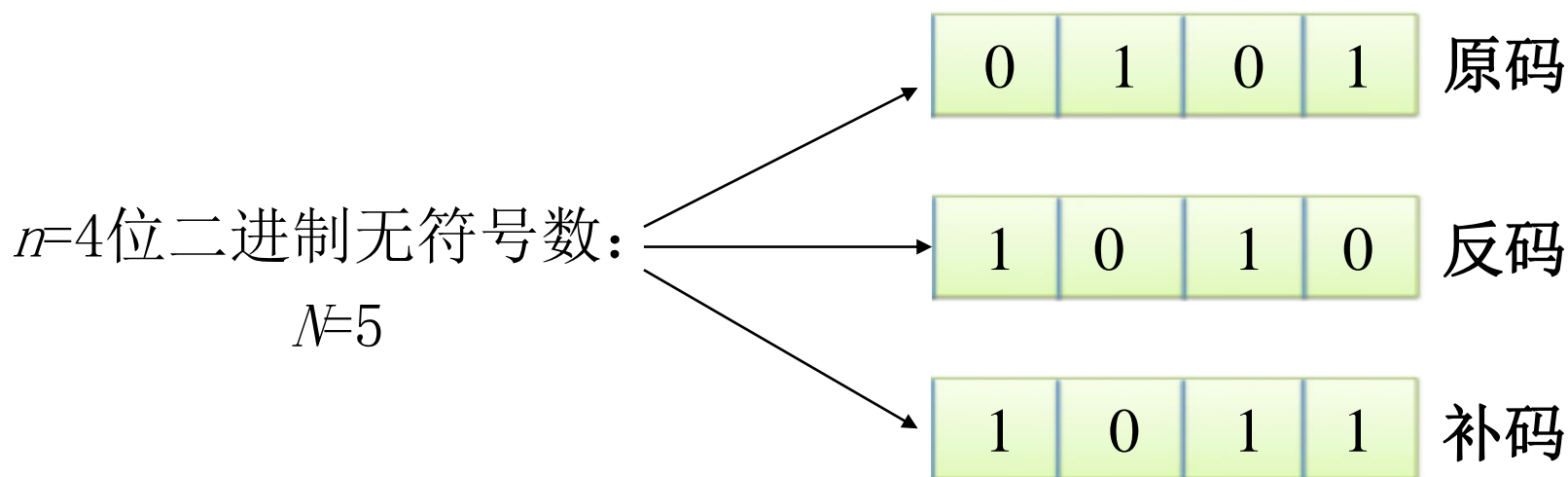
■ 2.2.2 整数的二进制编码—无符号数

■ 给定一个 n 位的二进制无符号数 N （真值）

■ 原码： N 对应的机器数

■ 反码： $(2^n - 1) - N$ 对应的机器数

■ 补码： $2^n - N$ 对应的机器数



2.2 数值数据的二进制编码

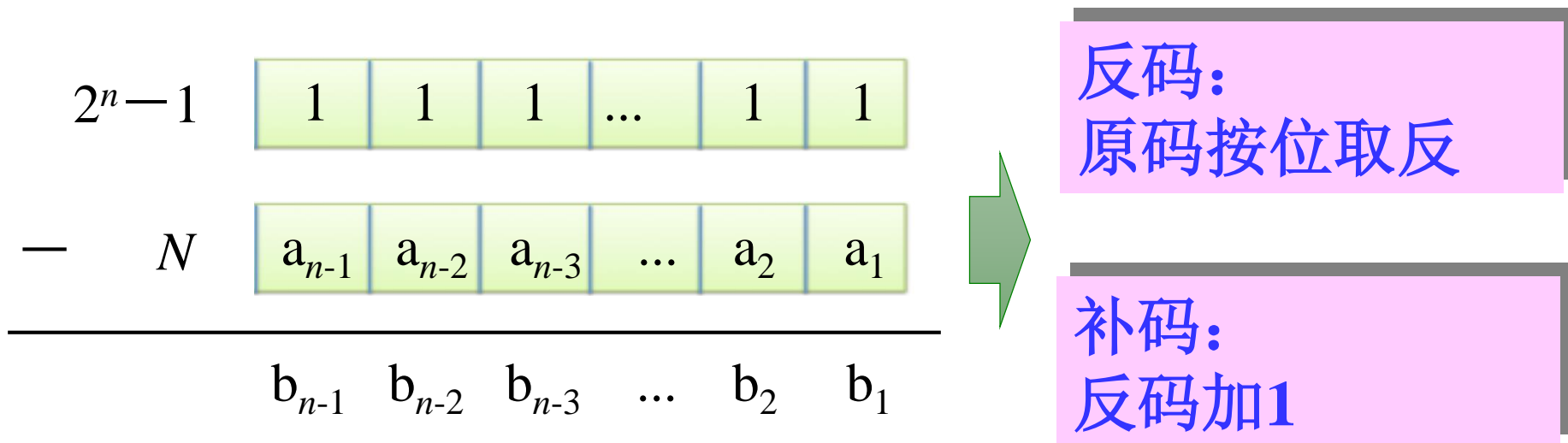
■ 2.2.2 整数的二进制编码—无符号数

■ 给定一个 n 位的二进制无符号数 N （真值）

■ 原码： N 对应的机器数

■ 反码： $(2^n - 1) - N$ 对应的机器数

■ 补码： $2^n - N$ 对应的机器数



2.2 数值数据的二进制编码

■ 2.2.2 整数的二进制编码—带符号数

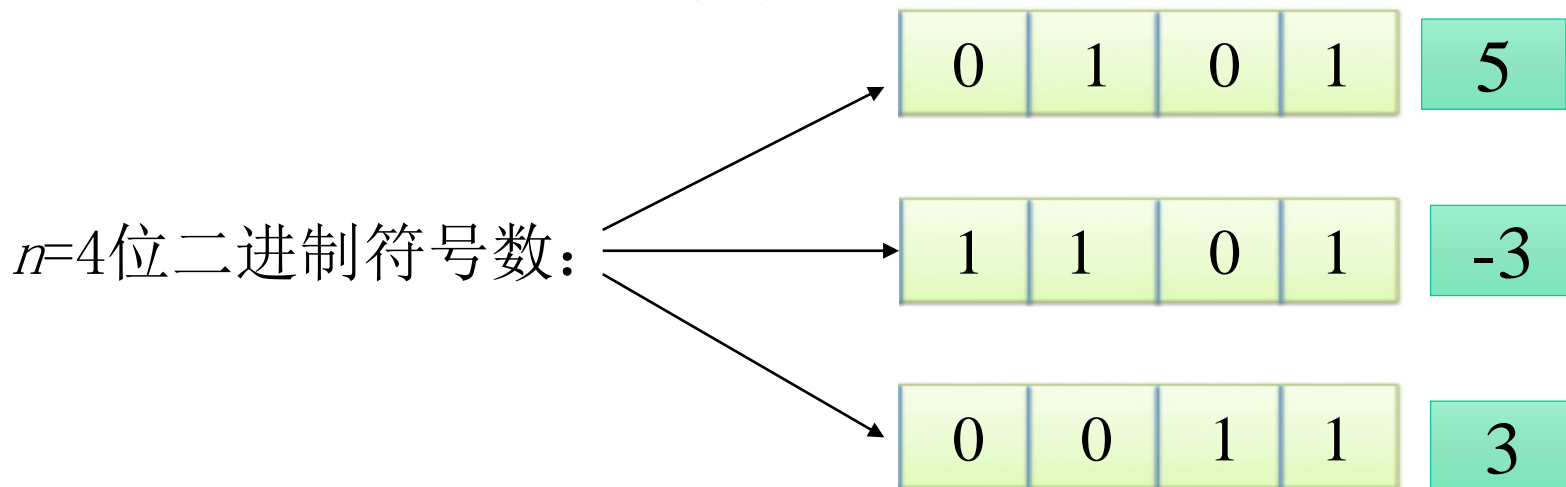
■ n 位的二进制数

■ 最高位符号位：0表示正数；1表示负数

■ 剩下的 $n-1$ 位有以下两种表示方式

■ 符号-数值：正数、0

■ 符号-补码：负数、0



2.2 数值数据的二进制编码

■ 2.2.2 整数的二进制编码——带符号数

■ n 位的二进制数

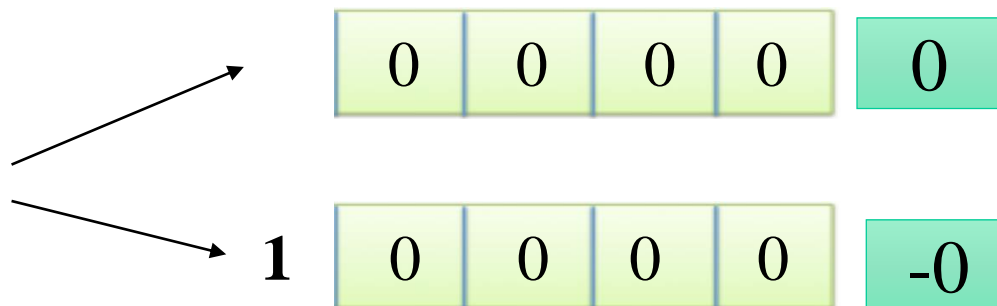
■ 最高位符号位：0表示正数；1表示负数

■ 剩下的 $n-1$ 位有以下两种表示方式

■ 符号-数值：正数、0

■ 符号-补码：负数、0

➤ 0 的表示唯一：



➤ 加、减法统一： $(\pm)A - (B) = (\pm)A + (-B)$

2.2 数值数据的二进制编码

■ 2.2.2 整数的二进制编码—带符号数

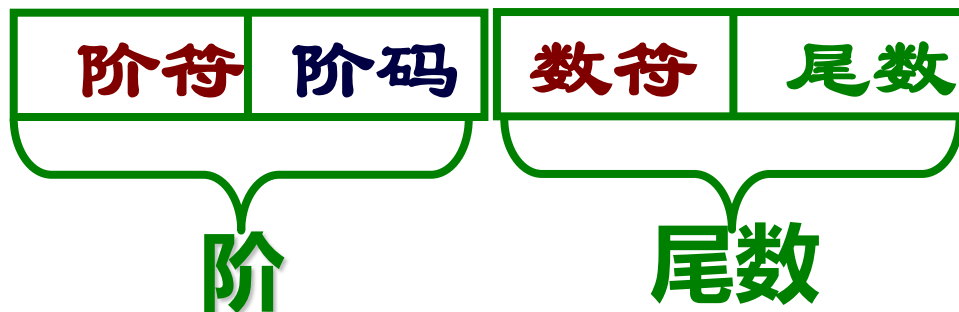
■ 两种表示方式的对比

十进制	符号-补码	符号-数值
+ 7	0111	0111
+ 6	0110	0110
+ 5	0101	0101
+ 4	0100	0100
+ 3	0011	0011
+ 2	0010	0010
+ 1	0001	0001
+ 0	0000	0000
- 0	—	1000
- 1	1111	1001
- 2	1110	1010
- 3	1101	1011
- 4	1100	1100
- 5	1011	1101
- 6	1010	1110
- 7	1001	1111
- 8	1000	—

2.2 数值数据的二进制编码

■ 2.2.3 小数的二进制编码

- 典型的浮点法：小数点位置可任意移动



- 例如：32位浮点数，用8位作阶，24位作尾数

$$\begin{aligned} & (72.45 \times 10^5)_{10} \\ & \approx (0.1101110)_2 \times (2^{23})_{10} \\ & = (0.1101110)_2 \times (2)_{10} \times (10111)_2 \end{aligned}$$

0	0010111	0	1101110...
---	---------	---	------------

提纲

1. 常用数制及转换
2. 数值数据的表示
- 3. 字符数据的表示**
4. 其它编码及应用
5. 现代典型计算系统中的数据表示

3.1 西文字符编码

- 用二进制编码表示数字、字母、字符等
- 英文：ASCII码
 - 美国信息交换标准编码
 - 7位二进制编码 ($B_6 \dots B_0$)
 - 94个可打印字符+34个控制字符=128个

换行	0A _H	10
回车	0D _H	13
空格	20 _H	32
‘0’~ ‘9’		
‘A’~ ‘Z’		
‘a’~ ‘z’		

3.1 西文字符编码

ASCII码表

b6b5b4 b3b2b1b0	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	'	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	EXT	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

3.2 汉字字符编码

- 用二进制编码表示汉字字符等
- 根据用途的差异具有以下形式：
 - 输入码：在键盘上利用数字、符号或拼音字母将汉字以代码的形式输入的代码
 - 国标码：我国1981年公布的“中华人民共和国国家标准信息交换汉字编码（GB2312—80）”代码，包含6763个汉字和682个其他基本图形字符，共7445个字符
 - 机内码：一个汉字被计算机系统内部处理和存储而使用的代码
 - 字形码：一个汉字供显示器和打印机输出的字形点阵代码

3.2 汉字字符编码

■ 举例：“英”

内 码	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D3A		印	英	樱	婴	鹰	应	纓	莹	萤	莖	苙	蝇	迎	赢	盈
D3B	影	颖	硬	映	哟	拥	佣	雍	痈	庸	雍	踊	蛹	咏	泳	涌
D3C	永	愚	勇	用	幽	优	悠	尤	由	庸	邮	铀	犹	油	游	酉
D3D	有	友	右	佑	幽	诱	又	迂	淤	于	雨	孟	榆	虞	愚	與
D3E	余	俞	逾	鱼	釉	渝	渔	隅	娱	雨	愈	与	屿	禹	宇	语
D3F	羽	玉	域	芋	愉	吁	遇	喻	御	欲	垣	欲	狱	育	誉	
D4A		浴	寓	裕	郁	豫	馭	鸳	冤	元	院	曰	袁	原	援	轅
D4B	园	员	圆	猿	源	缘	远	苑	愿	允	运	垣	约	越	跃	钥
D4C	岳	粤	月	悦	阅	耘	云	邕	陨	允		运	蕴	酝	晕	韵
D4D	孕															

D3A2

机内码

	0	7	8	15	
0					04, 10
					04, 10
					7F, FF
					04, 10
					04, 90
5					00, 80
					1F, FC
					10, 84
					10, 84
					10, 84
10					7F, FF
					01, 40
					02, 20
					04, 10
					08, 08
15					70, 07

字形码

3.3 统一字符编码

- 不同语言文字的不同的编码字符，能够覆盖世界上基本书写文字
 - **Unicode（统一码、万国码）**
 - 统一码联盟指定的标准
 - 采用16位二进制
 - 涵盖世界上几乎所有语言中的字符与文字
 - **UCS（通用字符集）**
 - 国际标准化组织ISO指定的统一编码标准
 - 采用最多32位二进制
 - 与Unicode2.0以上标准等同

3.3 统一字符编码

■ 举例：Unicode不同的编码方案

- UTF-8: 变长 1~4字节; 与ASCII码兼容。
- UTF-16: 变长2或者4字节;
- UTF-32: 定长4字节。

Code point range (hexadecimal)	UTF-8 encoding (binary, where bit positions with x are the bits of the code point value)
U+0000 0000 to U+0000 007F	0xxxxxxx
U+0000 0080 to U+0000 07FF	110xxxxx 10xxxxxx
U+0000 0800 to U+0000 FFFF	1110xxxx 10xxxxxx 10xxxxxx
U+0001 0000 to U+0010 FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

提纲

1. 常用数制及转换
2. 数值数据的表示
3. 字符数据的表示
- 4. 其它编码及应用**
5. 现代典型计算系统中的数据表示

4.1 BCD码

■ 二进制编码的十进制数（BCD码）

■ 用4位二进制表示十进制的10个数字

十进制符号	BCD码	十进制符号	BCD码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

BCD码是十进制数（**B**inary **C**oded **D**ecimal）！

4.1 BCD码

■ 编码方式

■ 逐位转换BCD码

$$(185)_{10} = (0001\ 1000\ 0101)_{\text{BCD}} = (10111001)_2$$

■ 编码特点

■ BCD码又称8421码，权值8、4、2、1

$$1001\ (9) = 1000\ (8) + 0001\ (1)$$

■ 1010到1111在BCD码中没有意义，也就是存在6个冗余的二进制码

■ 同一个8位二进制码表示的数，作为二进制数和二进制编码的十进制数时，所代表的数值是不相同的

4.2 奇偶校验码

■ 奇偶校验位

- 为了检测数据传输过程中可能存在的错误，通常在二进制编码中额外增加一个校验位（parity bit），用于表示编码中1的个数是奇数还是偶数。
- 偶校验：偶数个“1”，校验位“0”
- 奇校验：奇数个“1”，校验位“0”

传输数据	偶校验	奇校验
1 0 0 0 0 0 1	0 1 0 0 0 0 0 1	1 1 0 0 0 0 0 1
1 0 1 0 1 0 0	1 1 0 1 0 1 0 0	0 1 0 1 0 1 0 0

4.2 奇偶校验码

■ 带有奇偶校验位的检验码称为奇偶校验码，通常表示为原始二进制编码附加奇偶校验位

- 奇校验码：原编码+校验位，总共有奇数个1
- 偶校验码：原编码+校验位，总共有偶数个1

偶校验码

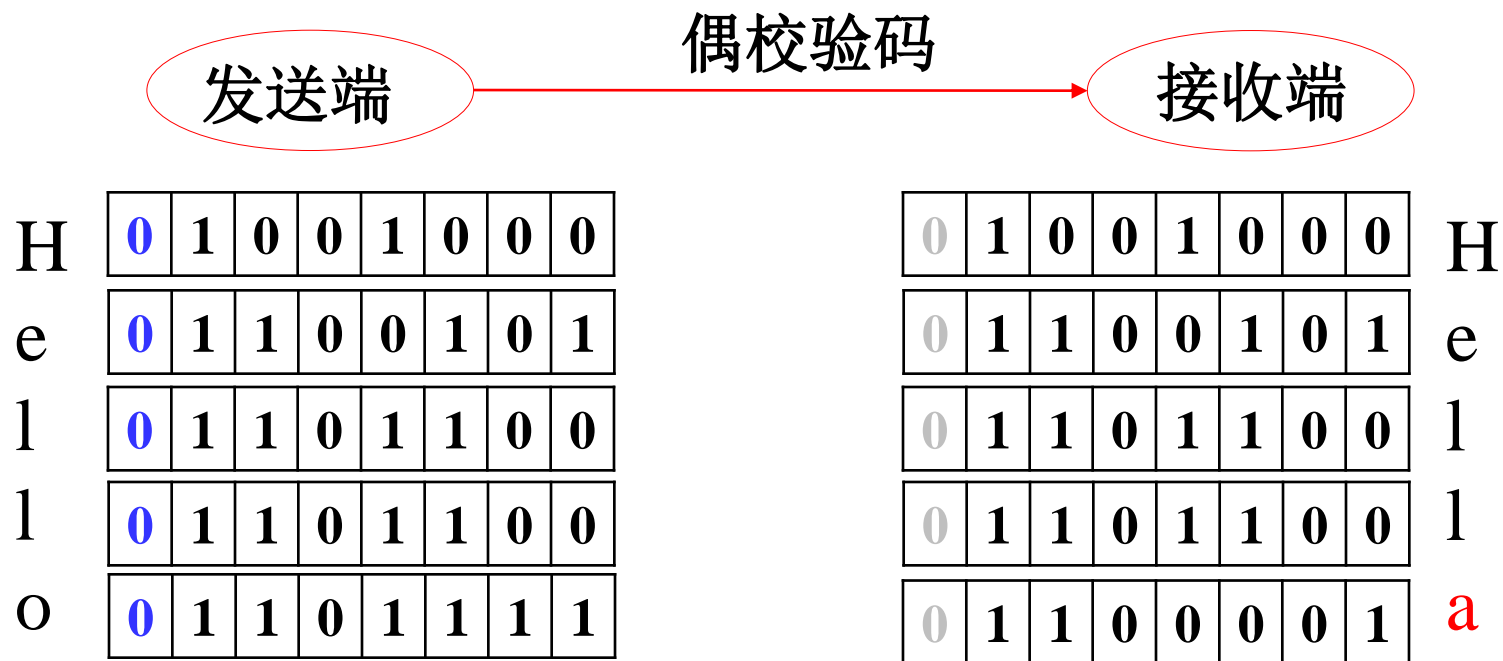
0	1	0	0	0	0	0	1
1	1	0	1	0	1	0	0

奇校验码

1	1	0	0	0	0	0	1
0	1	0	1	0	1	0	0

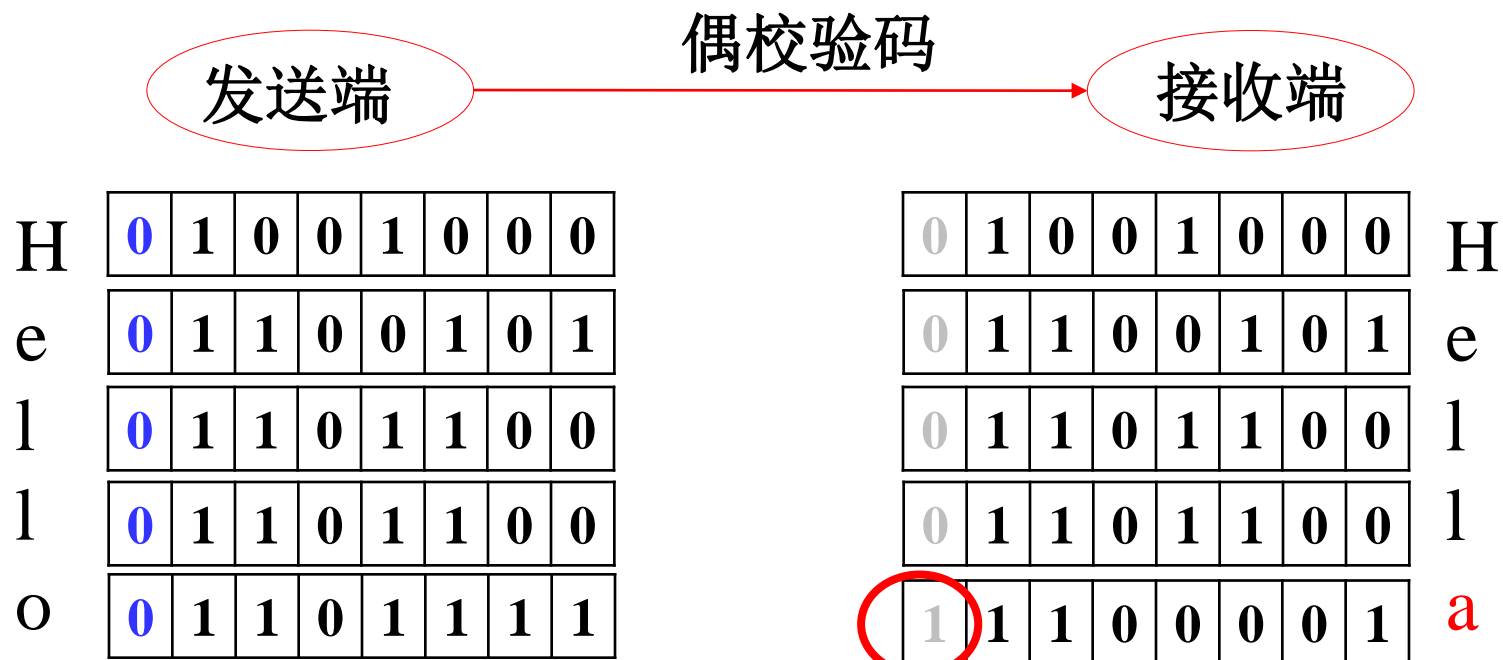
4.2 奇偶校验码

- 奇偶校验码是最简单的错误检验码，但无法确定哪一位出错



4.2 奇偶校验码

- 奇偶校验码是最简单的错误检验码，但无法确定哪一位出错



只具有部分检验能力

4.3 格雷码

■ 定义

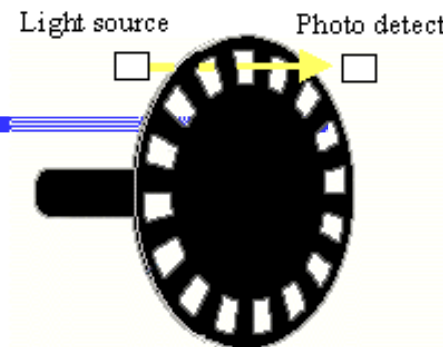
- 在一组数的编码中，若任意两个相邻的代码只有一位二进制数不同，则称这种编码为格雷码（Gray code）

- 美国贝尔实验室

Frank Gray在1953年
获得格雷码的专利

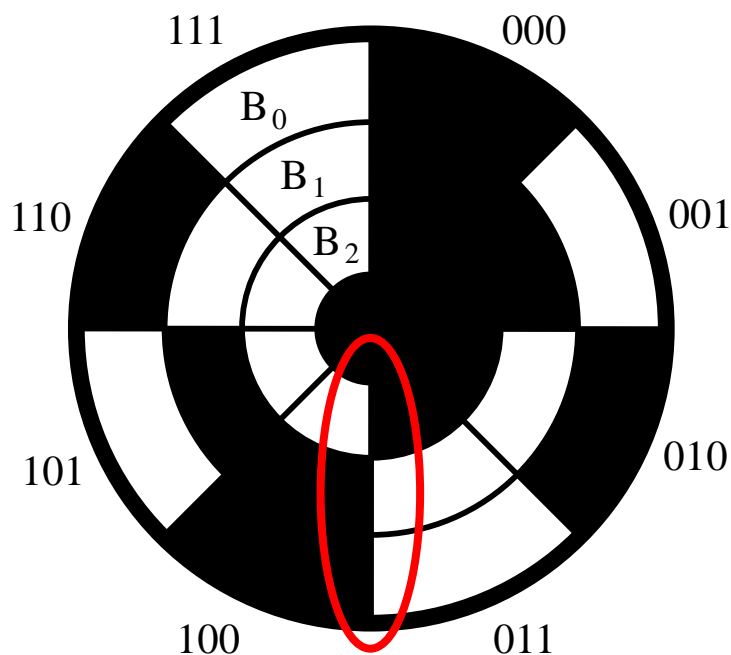
十进制	二进制	Gray
0	0000	0000
1	0001	0100
2	0010	0101
3	0011	0111
4	0100	0110
5	0101	0010
6	0110	0011
7	0111	0001
8	1000	1001
9	1001	1000

4.3 格雷码

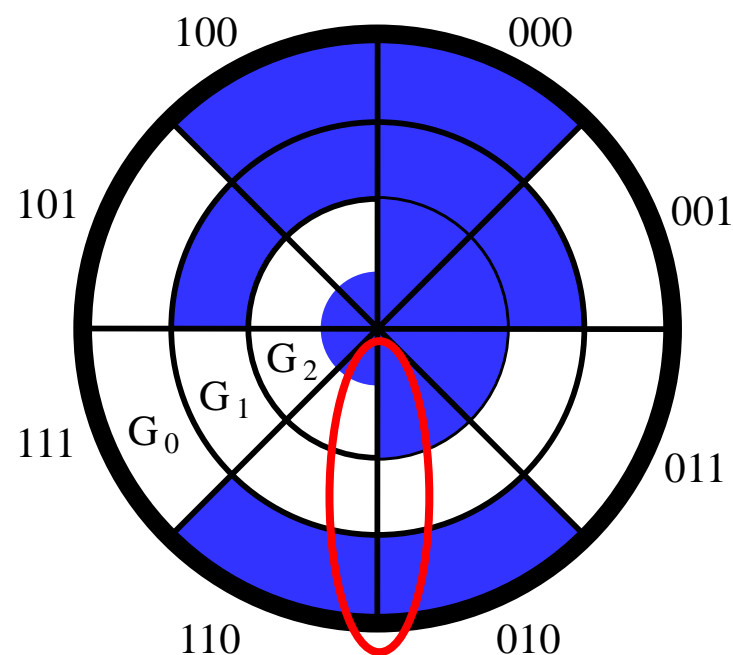


■ 用途1

- 在模-数转换过程中减少错误编码，
是一种可靠性编码，如光学轴角编码器



二进制码



格雷码

4.3 格雷码

■ 用途2:

■ 低功耗

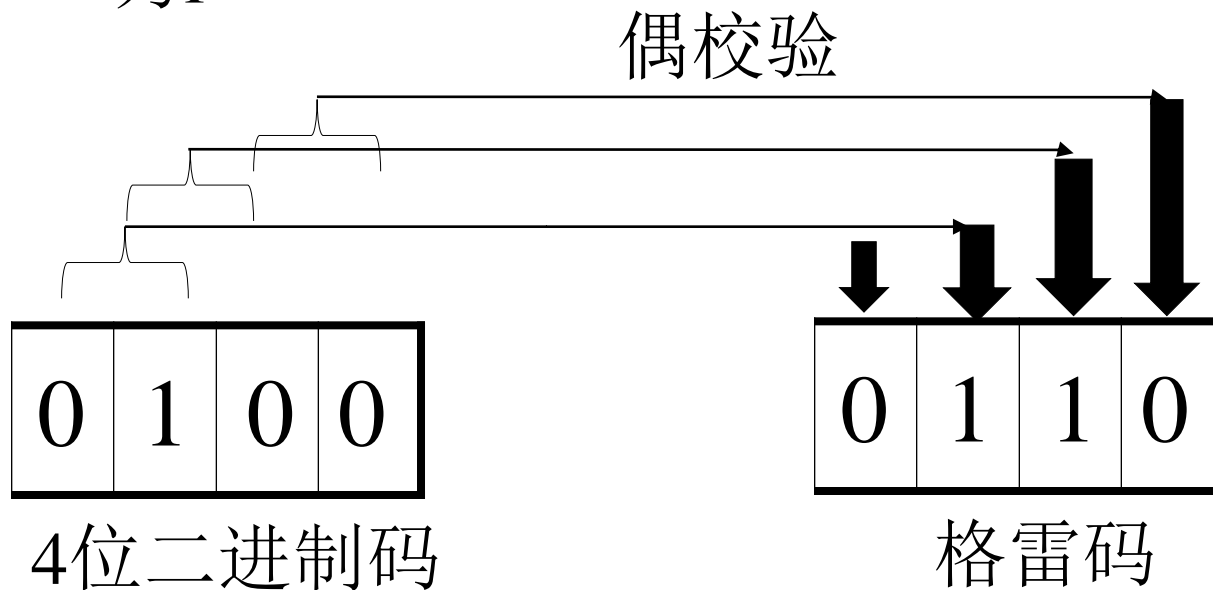
Binary Code	Bit Changes	Gray Code	Bit Changes
000		000	
001	1	001	1
010	2	011	1
011	1	010	1
100	3	110	1
101	1	111	1
110	2	101	1
111	1	100	1
000	3	000	1

4.3 格雷码

■ 编码方式

■ n位（偶数）二进制计数序列中的数值

- 前一半数值：左边最高位为0，往右各位由原二进制编码的每一位与它左边相邻位的偶校验构成
- 后一半数值：前一半逆序排列，并将左边最高位设为1



4.3 格雷码

■ 编码方式

■ 例子：4位二进制格雷码

十进制数	自然二进制数	格雷码	十进制数
0	0000	0000	8
1	0001	0001	9
2	0010	0011	10
3	0011	0010	11
4	0100	0110	12
5	0101	0111	13
6	0110	0101	14
7	0111	0100	15



4.3 格雷码

■ 编码方式

■ $n+1$ 位二进制计数序列中的数值

- 前 2^n 个码字等于 n 位格雷码的码字，按顺序书写，加前缀0
- 后 2^n 个码字等于 n 位格雷码的码字，按逆序书写，加前缀1

4.3 格雷码

■ 编码方式

■ n+1位二进制计

2位格雷码	3位格雷码	4位格雷码	4位自然二进制码
00	000	0000	0000
01	001	0001	0001
11	011	0011	0010
10	010	0010	0011
	110	0110	0100
	111	0111	0101
	101	0101	0110
	100	0100	0111
		1100	1000
		1101	1001
		1111	1010
		1110	1011
		1010	1100
		1011	1101
		1001	1110
		1000	1111

提纲

1. 常用数制及转换
2. 数值数据的表示
3. 字符数据的表示
4. 其它编码及应用
- 5. 现代典型计算系统中的数据表示**

5.1 现代典型计算机系统

■ IA-32/64架构：Intel Architecture 32位/64位



■ MIPS架构



索尼游戏机



任天堂游戏机



特斯拉

5.1 现代典型计算机系统

■ ARM架构



诺基亚6110



iPhone 1

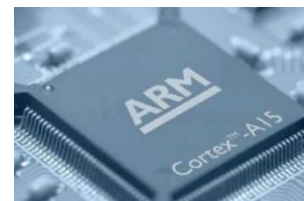


安卓

■ RISC V架构

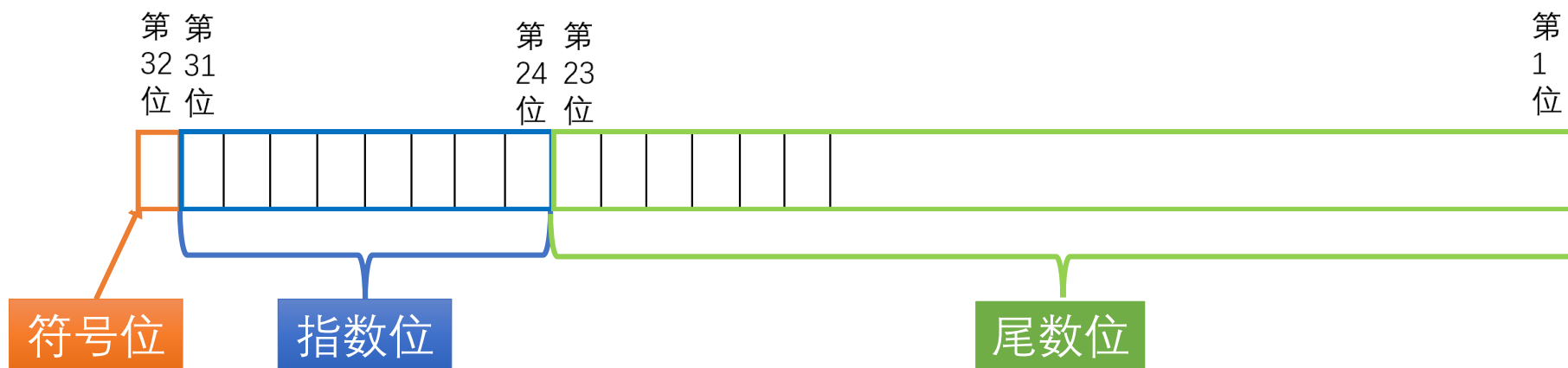
技术与市场	x86或ARM架构	RISC-V
架构篇幅	数千页	少于三百页
指令数	指令数繁多，且不同分支不兼容	基本指令集40余条
模块化	不支持	支持
可扩展性	不支持	支持
硬件实现	复杂度高	硬件设计与编译实现非常简单
商业运作	x86封闭，ARM架构授权昂贵	开源、免费
生态环境	成熟	快速起步中
应用市场	服务器与桌面PC (x86)，移动和嵌入式 (ARM)	从物联网切入，可覆盖从微控制器到超级计算机的全计算领域

5.2 典型数据表示



■ IA-32和ARM架构下单精度浮点数

■ IEEE 754标准



编码规则

- 符号位：0正，1负
- 指数位（阶码）：移位存储，阶码=指数+127
- 尾数位：科学计数法的小数部分

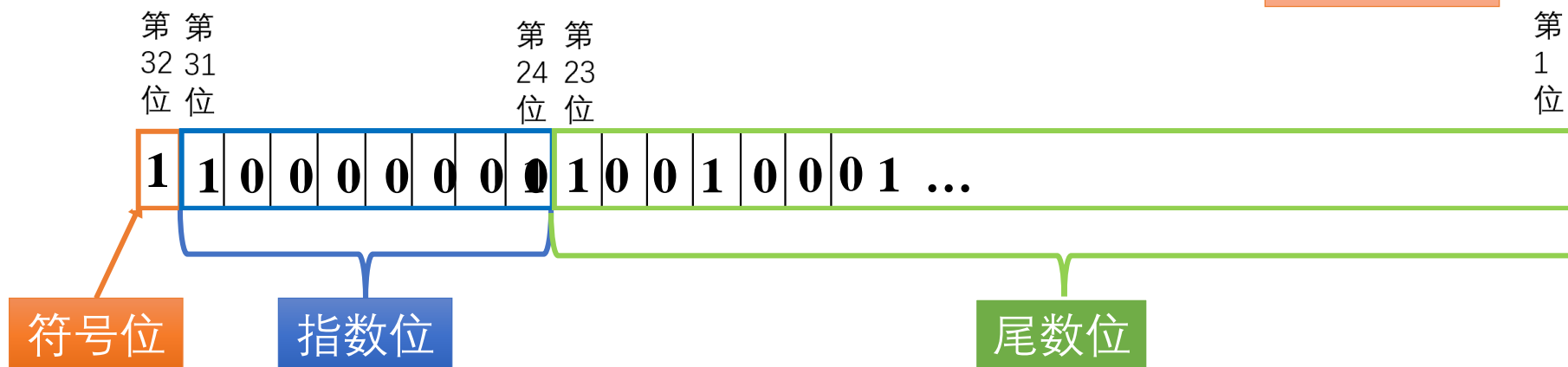
5.2 典型数据表示



■ IA-32和ARM架构下单精度浮点数

■ IEEE 754标准

-3.14



编码规则

- 符号位: 0正, 1负
- 指数位 (阶码): 移位存储, 阶码=指数+127
- 尾数位: 科学计数法的小数部分

$$\begin{aligned} &11.0010001 \\ &\quad \swarrow \\ &1.10010001 \times 2^1 \end{aligned}$$

小结

- 数制的概念
- 不同数制的转换
 - 二进制与十进制
 - 十进制与非十进制
- 符号数的编码方式
- 十进制编码方式
- ASCII字符编码方式
- 格雷码的概念