数字电路

第2章 数制与编码

杨 旭

北京理工大学

pyro_yangxu@bit.edu.cn

本章内容

- 2.1 数制
- 2.2 数制转换
- 2.3 二进制符号数的表示方法
- 2.4 二-十进制编码（BCD码）
- 2.5 格雷（Gray）码
- 2.6 ASCII字符集
- 2.7 检错码和纠错码

数制与编码

- 人类在日常生活中如何表示数字?

十进制

- 计算机中如何表示数字?

二进制

- 二进制和十进制之间的关系是什么?
- 还有别的常用的表示方法吗?

八进制、十六进制 ……

数制

数制

- 即记数法，人们用一组规定的符号和规则来表示数的方法

数制的两个基本要素

- 基数 —— 符号
- 位权 —— 规则

数制

基数

- 数制中每一位数所用到的数码的个数。
- 基数为N的记数制中，含有0、1、…、N-1,共N个数码，进位规律是“逢N进一”

位权

- 数制中每一固定位置对应的单位值（数值“1”）代表的值。
- 例如：十进制中第二位的位权是10，第一位的位权是1

数制

- 任意一个数均可写为多项式形式

$$(345.67)_{10}=3\times10^{2}+4\times10^{1}+5\times10^{0}+6\times10^{-1}+7\times10^{-2}$$

符号　位权

- 十进制数的基数为10，用到的10个符号为

0 1 2 3 4 5 6 7 8 9

数制

对于任意R进制数N，假设整数位数为n、小数位数为m

$$
\begin{aligned}
(N)_R &= K_{n-1}K_{n-2}\cdots K_0.K_{-1}\cdots K_{-m} \\
&= K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_0R^0 + K_{-1}R^{-1} + \cdots K_{-m}R^{-m} \\
&= \sum_{i=-m}^{n-1} K_i R^i
\end{aligned}
$$

- 其中R为R进制数的基数，K_i为0,1，……R-1范围内取值的数字。

二、八、十、十六进制数

- $(N)_{10}=\sum d_i \cdot 10^i$
- d_i的取值范围：0，1，2…9

- $(N)_2=\sum b_i \cdot 2^i$
- b_i的取值范围：0，1

- $(N)_8=\sum q_i \cdot 8^i$
- q_i的取值范围：0，1，2…7

- $(N)_{16}=\sum h_i \cdot 16^i$
- h_i的取值范围：0，1…9，A,B,C,D,E,F

二、八、十、十六进制数

二进制 Binary, B

八进制 Octal, O(Q)

十进制 Decimal, D

十六进制 Hexadecimal, H

二、八、十六进制到十进制的转换

转换方法：直接按位加权相加

例1 $(101.001)_2 = 1\cdot2^2+0\cdot2^1+1\cdot2^0+0\cdot2^{-1}+0\cdot2^{-2}+1\cdot2^{-3}$
$= (5.125)_{10}$

例2 $(32.56)_8 = 3\cdot8^1+2\cdot8^0+5\cdot8^{-1}+6\cdot8^{-2}$
$= (26.71875)_{10}$

例3 $(ED.A)_{16} = 14\cdot16^1+13\cdot16^0+10\cdot16^{-1}$
$= (237.625)_{10}$

十、二、八、十六进制对照表

十进制	二进制	八进制	十六进制
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	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

pyro_yangxu@bit.edu.cn

二←→八进制之间的转换

转换方法：每三位二进制数对应一位八进制数。转换时从小数点向左、向右每3位为一组，直接写出对应的八进制数即可。 小数点后最后一组要补足3位。

$(10010111.1101)_2$

$= (\underline{010}\ \underline{010}\ \underline{111}.\underline{110}\ \underline{100})_2$

$= (227.64)_8$

$(227.64)_Q$

$= (\underline{010}\ \underline{010}\ \underline{111}.\underline{110}\ \underline{100})_B$

$= (10010111.1101)_B$

二←→十六进制之间的转换

转换方法：四位二进制数对应一位十六进制数。转换时从小数点向左、向右每4位为一组，直接写出对应的十六进制数即可。 小数点后最后一组要补足4位。

$(110110111.011)_2$

$=(\underline{0001}\ \underline{1011}\ \underline{0111}.\underline{0110})_2$

$=(1B7.6)_{16}$

$(1C7.6)_H$

$=(\underline{0001}\ \underline{1100}\ \underline{0111}.\underline{0110})_B$

$=(111000111.011)_B$

八←→十六进制之间的转换

转换方法：八进制数与十六进制数之间的转换可以通过转换为二进制数作为中间过程。

$(1C7.6)_{16}$

$=(\underline{0001}\ \underline{1100}\ \underline{0111}.\underline{0110})_2$

$=(\underline{111}\ \underline{000}\ \underline{111}.\underline{011})_2$

$=(707.3)_8$

$(707.3)_8$

$=(\underline{111}\ \underline{000}\ \underline{111}.\underline{011})_2$

$=(\underline{0001}\ \underline{1100}\ \underline{0111}.\underline{0110})_2$

$=(1C7.6)_{16}$

十进制到二进制的转换

整数部分和小数部分分别进行转换

整数部分用连除法

例：$(59)_{10}=(\ ?\ )_2$

解：

$$0 \xleftarrow{/2} 1 \xleftarrow{/2} 3 \xleftarrow{/2} 7 \xleftarrow{/2} 14 \xleftarrow{/2} 29 \xleftarrow{/2} 59$$

1	1	1	0	1	1
b_5	b_4	b_3	b_2	b_1	b_0

$$(59)_{10}=(111011)_2$$

十进制到二进制的转换

整数部分和小数部分分别进行转换

小数部分用连乘法

例：$(0.8125)_{10}=(?)_2$

解：$0.8125 \xrightarrow{\times 2} 0.625 \xrightarrow{\times 2} 0.25 \xrightarrow{\times 2} 0.5 \xrightarrow{\times 2} 0$

	1	1	0	1
	b_{-1}	b_{-2}	b_{-3}	b_{-4}

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

十进制到二进制的转换

整数、小数分别转换，转换结果合到一起即可。

$(59)_{10}=(111011)_2$

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

$(59.8125)_{10}$

$=(111011.1101)_2$

小数位数的确定

给定位数

给定精度

$(0.2)_{10}=(\ ?\ )_2$，
保留8位小数

$(0.2)_{10}=(\ ?\ )_2$，
要求误差小于1%

$(0.2)_{10}=(0.00110011)_2$

So easy

∵0.2*1%=0.002

且$2^{-9}=0.00195<0.002$

∴ 需要保留9位小数

∴$(0.2)_{10}=(0.001100110)_2$

十进制到八、十六进制的转换

方法一：连除（乘）法。

例：$(59)_{10}=(?)_8$

解：$0 \leftarrow 7 \leftarrow 59$

$7 \quad 3$

$q_1 \quad q_0$

所以 $(59)_{10}=(73)_8$

例：$(0.8125)_{10}=(?)_8$

解：$0.8125 \rightarrow 0.5 \rightarrow 0$

$6 \quad 4$

$q_{-1} \quad q_{-2}$

所以$(0.8125)_{10}=(0.64)_8$

$$(59.8125)_{10}=(73.64)_8$$

十进制到八、十六进制的转换(续)

方法一：连除（乘）法。

例：$(59)_{10}=(?)_{16}$

解：$0 \leftarrow 3 \leftarrow 59$

3　　B(11)

h_1　h_0

所以 $(59)_{10}=(3B)_{16}$

例：$(0.8125)_{10}=(?)_{16}$

解：$0.8125 \rightarrow 0$

D(13)

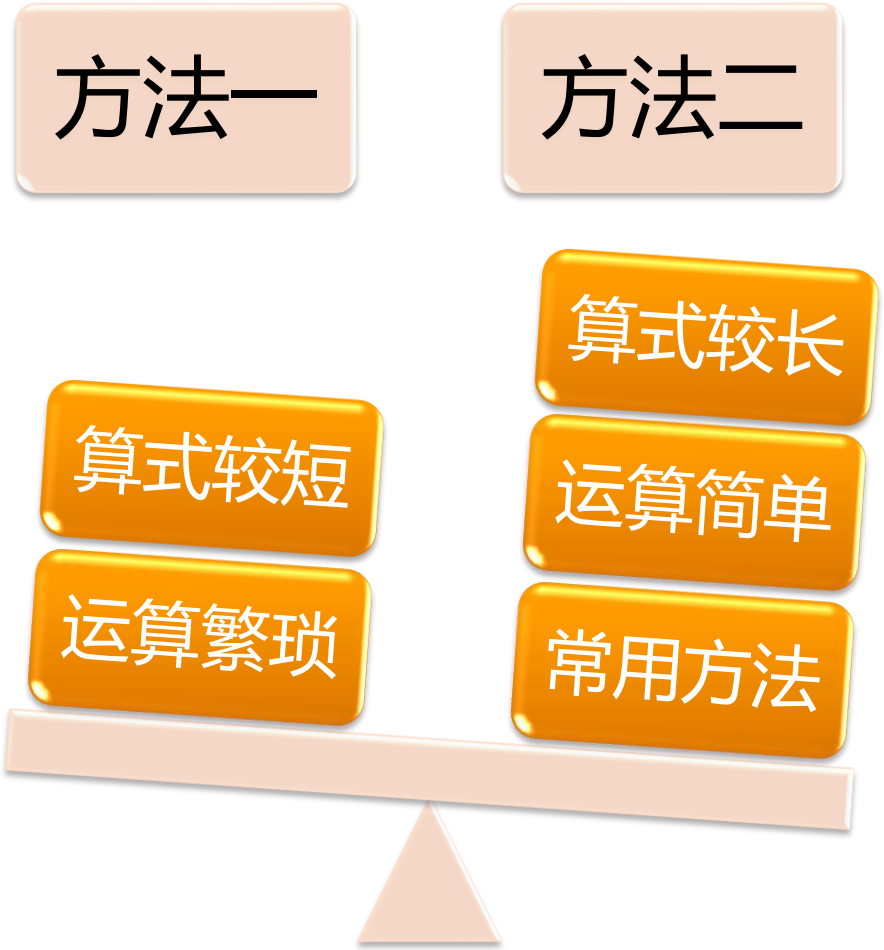
h_{-1}

所以$(0.8125)_{10}=(0.D)_{16}$

$$(59.8125)_{10}=(3B.D)_{16}$$

十进制到八、十六进制的转换

方法二：十进制 → 二进制 → 八（十六）进制数



重要的数字

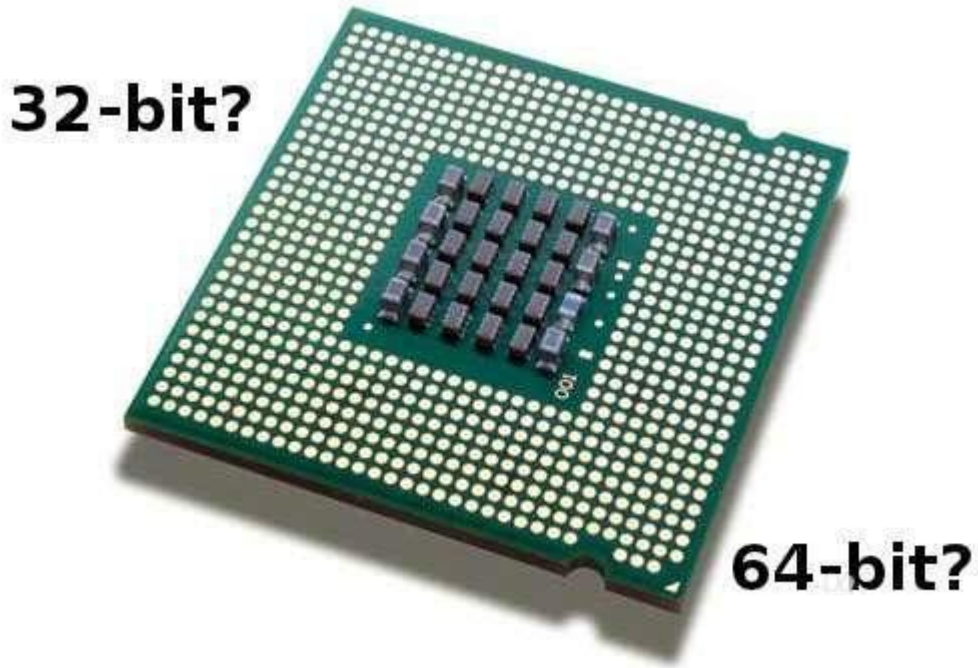
2^i, 2^i-1，i=0,1,…10

2^{10}=1024 ≈ 1000=1K

2^{20}=1M

2^{30}=1G

2^{32}=4G



二进制符号数的表示方法

符号数:

带正、负号的数

如何表示符号?

如何表示数值?

原码
表示法

反码
表示法

补码
表示法

原码表示法

符号

1位二进制数表示符号：0表示正数，1表示负数

数值

数的大小以该数的绝对值表示

符号位置

符号放在最高位

$\underline{0}\underline{0010111} = +23$　　$\underline{1}\underline{0010111} = -23$

符号　绝对值　　符号　绝对值

原码表示法

例：数据位宽n=8，最高位为符号位

十进制表示	二进制	原码表示法
$(+37)_{10}$	+0100101	00100101
$(-37)_{10}$	- 0100101	10100101
$(+0)_{10}$	+0000000	00000000
$(-0)_{10}$	- 0000000	10000000
$(+127)_{10}$	+1111111	01111111
$(-127)_{10}$	-1111111	11111111

- n位数字采用原码可表示的数据范围： $-(2^{n-1}-1)\sim+(2^{n-1}-1)$
- 0有两种表示方式：+0 和-0

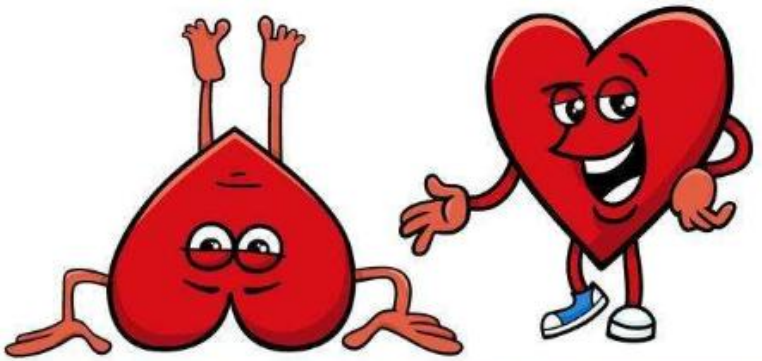
反码

反码：将二进制数的每一位分别求反而得到的二进制码

0 反码→ 1　　1 反码→ 0

N　：　10011011

$N_{反}$：　01100100



符号数的反码表示法

$\underline{0}\underline{0010111} = +23$　　$\underline{1}\underline{0010111} = -104$

符号　原码　　符号　反码

反码表示法

例：数据位宽n=8，最高位为符号位

十进制表示	二进制	反码表示法
$(+37)_{10}$	+0100101	00100101
$(-37)_{10}$	- 0100101	11011010
$(+0)_{10}$	+0000000	00000000
$(-0)_{10}$	- 0000000	11111111
$(+127)_{10}$	+1111111	01111111
$(-127)_{10}$	- 1111111	10000000

- n位数字采用反码可表示的数据范围： $-(2^{n-1}-1)\sim+(2^{n-1}-1)$
- 0有两种表示方式：+0 和-0

补码

补码：设数N为包含n位整数、m位小数的二进制数，则N的补码定义为：

$$(N)_{补n} = 2^n - N$$

N的补码与N的大小有关，还与整数位数n有关，与小数位数m无关

设n=8, 则

$(11001)_{补.8}$ = 2^8-11001=11100111

$(11001.0101)_{补.8}$ = 2^8-11001.0101
= 11100110.1011

补码的求法

方法一

- 利用补码的定义计算

较为繁琐，一般不用

方法二

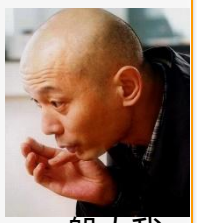
- 将原码补足n位**整数**后**整体**求反加1

常用方法

如：求(11001)在n=8时的补码

1）补齐8位：00011001
2）求反：11100110
3）加1：11100111

“加1”是在整个数的最后一位（包括小数部分）加1 ！！



一般人我不告诉他

补码的求法

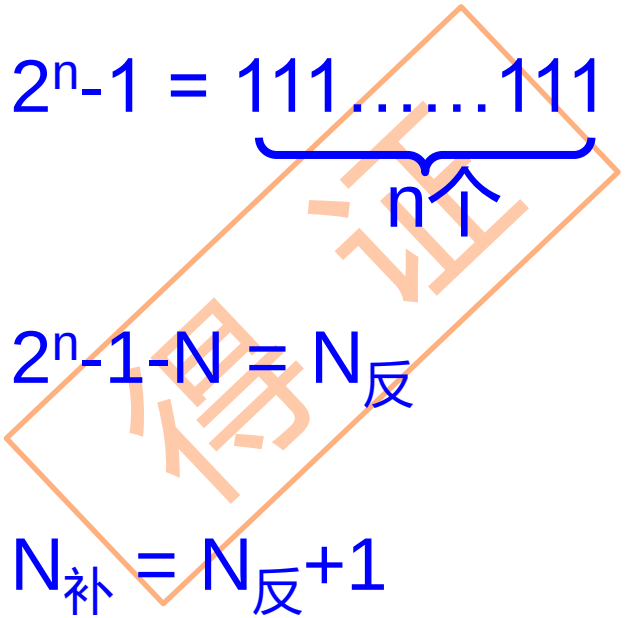
证明方法二

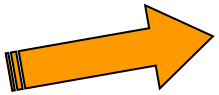
$N_{补} = 2^n - N = (2^n - 1 - N) + 1$

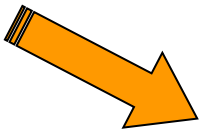
$2^n - 1 = \underbrace{111\ldots\ldots111}_{n个}$

$2^n - 1 - N = N_{反}$

$N_{补} = N_{反} + 1$







```
 100000000
-        1
----------
  11111111
```

```
  11111111
- 01101001
----------
  10010110
```

补码的求法

方法三

- 将数N补足n位整数
- 从右往左第一个1及其右边的0不变
- 其余各位求反

$N = 00100100 = \underline{00100}\ \underline{100}$

$(N)_{补.8} = \underline{11011}\ \underline{100}$

依据：100求反加1仍为100，其它位求反不变

补充概念

- 反码又称为1的补码（1’s complement)
  - 2^n-1-N
- 补码又称为2的补码（2’s complement)
  - 2^n-N
  - $N+(N)_{补}=N+2^n-N=2^n$
  - $((N)_{补})_{补}=N$

符号数的补码表示法

符号

1位二进制数表示符号：0表示正数，1表示负数

数值

正数用原码表示，负数用该数绝对值的补码表示

符号位置

符号放在最高位

$0\underline{0010111} = +23$

符号 原码

$1\underline{0010111} = -105$

符号 绝对值的补码

符号数的补码表示法

例：数据位宽n=8，最高位为符号位

十进制表示	二进制	补码表示法
$(+37)_{10}$	+0100101	00100101
$(-37)_{10}$	- 0100101	11011011
$(+0)_{10}$	+0000000	00000000
$(-0)_{10}$	- 0000000	00000000
$(+127)_{10}$	+1111111	01111111
$(-127)_{10}$	- 1111111	10000001
$(-128)_{10}$	-10000000	10000000

- n位数字采用补码可表示的数据范围： $-(2^{n-1})\sim+(2^{n-1}-1)$
- +0 和-0的表示方法相同

问题1：如何求有符号数的相反数？

A ➡ -A ？

原码表示法：将符号位取反，其余位不变

$A=(3)_{10}=(0011)_2$ ➡ $-A=(-3)_{10}=(1011)_2$

$B=(-2)_{10}=(1010)_2$ ➡ $-B=(2)_{10}=(0010)_2$

反码表示法：将整个数值连同符号位一起取反（求反码）

$A=(3)_{10}=(0011)_2$ ➡ $-A=(-3)_{10}=(1100)_2$

$B=(-2)_{10}=(1101)_2$ ➡ $-B=(2)_{10}=(0010)_2$

补码表示法：将整个数值连同符号位一起取反加1（求补码）

$A=(3)_{10}=(0011)_2$ ➡ $-A=(-3)_{10}=(1101)_2$

$B=(-2)_{10}=(1110)_2$ ➡ $-B=(2)_{10}=(0010)_2$

问题2：如何扩展数据的宽度

4位 ⇨ 8位 ？

原码表示法：在符号位和数据位之间填充0

$A=(3)_{10}=(0011)_2$ ⇨ $A=(3)_{10}=(00000011)_2$

$B=(-2)_{10}=(1010)_2$ ⇨ $B=(-2)_{10}=(10000010)_2$

反码表示法：直接进行符号扩展

$A=(3)_{10}=(0011)_2$ ⇨ $A=(3)_{10}=(00000011)_2$

$B=(-2)_{10}=(1101)_2$ ⇨ $B=(-2)_{10}=(11111101)_2$

补码表示法：直接进行符号扩展

$A=(3)_{10}=(0011)_2$ ⇨ $A=(3)_{10}=(00000011)_2$

$B=(-2)_{10}=(1110)_2$ ⇨ $B=(-2)_{10}=(11111110)_2$

有符号数的十进制值求取

- 对于原码表示法，不管是正数还是负数，将数值部分1的对应位置权值相加，忽视0的对应位置。符号位确定正负。

Ex. 1

Determine the decimal value of this signed binary number expressed in sign-magnitude: 10010101.

Solution

The seven magnitude bits and their powers-of-two weights are as follows:

2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	0	1

Summing the weights where there are 1s,

$$16 + 4 + 1 = 21$$

The sign bit is 1; therefore, the decimal number is **-21**

有符号数的十进制值求取

- 反码表示法的正数和原码表示法的求法一致

- 反码表示法的负数，先给符号位赋予一个负的权值，将1的位置的权值相加，再加1

有符号数的十进制值求取

Ex. 1

Determine the decimal value of this signed binary number expressed in 1's complement.
(a) 00010111 (b) 11101000

Solution

(a) The bits and their powers-of-two weights are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	1	1	1

Summing the weights where there are 1s,

$$16 + 4 + 2 + 1 = +23$$

(b) The bits and their powers-of-two weights are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0	0	0

Summing the weights where there are 1s,

$$-128 + 64 + 32 + 8 = -24$$

Adding 1 to the result, the final decimal number is

$$-24 + 1 = -23$$

有符号数的十进制值求取

- 补码表示法，不管是正数还是负数，给符号位赋予一个负的权值，把所有1对应位置的权值相加即可

有符号数的十进制值求取

Ex. 1

Determine the decimal value of this signed binary number expressed in 2's complement.
(a) 01010110 (b) 10101010

Solution

(a) The bits and their powers-of-two weights are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	0	1	1	0

Summing the weights where there are 1s,

$$64 + 16 + 4 + 2 = +86$$

(b) The bits and their powers-of-two weights are as follows:

-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

Summing the weights where there are 1s,

$$-128 + 32 + 8 + 2 = -86$$

- 结论: 补码表示法下，不管是正数还是负数，都是统一的求解方法，较为方便。估通常采用补码表示法来表示有符号数。

补码表示的符号数的加减运算

只需要加法器就可完成加减运算

- A+B
- A-B = A + (-B)
- B-A = (-A) + B
- -A-B = (-A) + (-B)

节省硬件，降低成本

符号位直接参与运算

结果仍为补码表示

补码表示的符号数的加减运算

设n=8，有两个正数A=10011，B=1101。试用补码求A+B，A-B，B-A，-A-B

解：$(A)_{补.8}$ = 00010011，　$(B)_{补.8}$ = 00001101

$(-A)_{补.8}$ =11101101，　$(-B)_{补.8}$ = 11110011

A+B=100000　　A-B=110　　-A+B=11111010　　-A-B=11100000

00010011	00010011	11101101	11101101
+ 00001101	+ 11110011	+ 00001101	+ 11110011
00100000	1 00000110	11111010	1 11100000

补码表示的符号数的加减运算(证明)

对任意正整数N_1和N_2,$(-N_1)_{补.n}=2^n-N_1$, $(-N_2)_{补.n}=2^n-N_2$

N_1+N_2

- 两个正数相加，结果为正数

N_1-N_2

- $N_1-N_2=N_1+(2^n-N_2)=2^n-(N_2-N_1)$,
  - 若$N_2>N_1$,则结果为负，$2^n-(N_2-N_1)$就是$-(N_2-N_1)$的补码表示形式；
  - 若$N_1>N_2$,则结果为$2^n+(N_1-N_2)$,2^n位于第n+1位，为第n位的进位，位于n位运算器之外，舍去，故结果为N_1-N_2，正确。

补码表示的符号数的加减运算(证明)

N_2-N_1

- 与N_1-N_2同理

$-N_1-N_2$

- $-N_1-N_2=（2^n-N_1）+（2^n-N_2）=2^n+[2^n-（N_1+N_2）]$
- 第1个2^n为第n位的进位，位于在第n+1位上，在n位运算器之外，舍去
- 而$[2^n-(N_1+N_2)]$就是负数$-(N_1+N_2)$的补码表示，计算结果正确

补码表示的符号数的加减运算_溢出

设n=8，有两个正数A=110011，B=1101101。试用补码求A+B，A-B，B-A，-A-B

解：$(A)_{补.8}$ = 00110011，$(B)_{补.8}$ = 01101101

$(-A)_{补.8}$ =11001101，$(-B)_{补.8}$ = 10010011

A+B=10100000　A-B=11000110　-A+B=00111010　-A-B=01100000

```
  00110011     00110011     11001101     11001101
+ 01101101   + 10010011   + 01101101   + 10010011
----------   ----------   ----------   ----------
  10100000     11000110    100111010     101100000
```

51+109=-96　51-109=-58　-51+109=58　-51-109=96

补码表示的符号数的加减运算_溢出

溢出(Overflow)：计算结果超出了n位符号数的表示范围

两个加数符号相同

- 当两个加数的绝对值之和超出符号数的表示范围时，发生溢出

两个加数符号不同

- 相加结果的绝对值一定小于某一个加数的绝对值，因此一定不会溢出

补码表示的符号数的加减运算_溢出

溢出判断

•当第n-1位（符号位）和第n-2位（最高数字位）不同时有进位（两个负数相加时）或不同时无进位（两个正数相加时）时有溢出发生

有没有简单点的办法啊

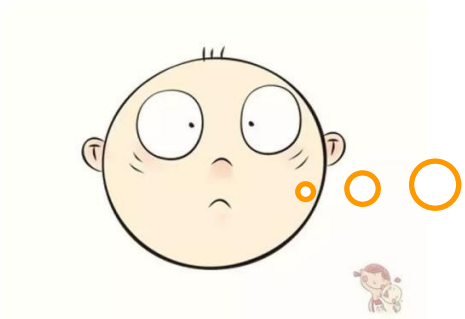
二-十进制编码（BCD码）

BCD：Binary Coded Decimal

表示0~9十个十进制数字，需要几位二进制码?

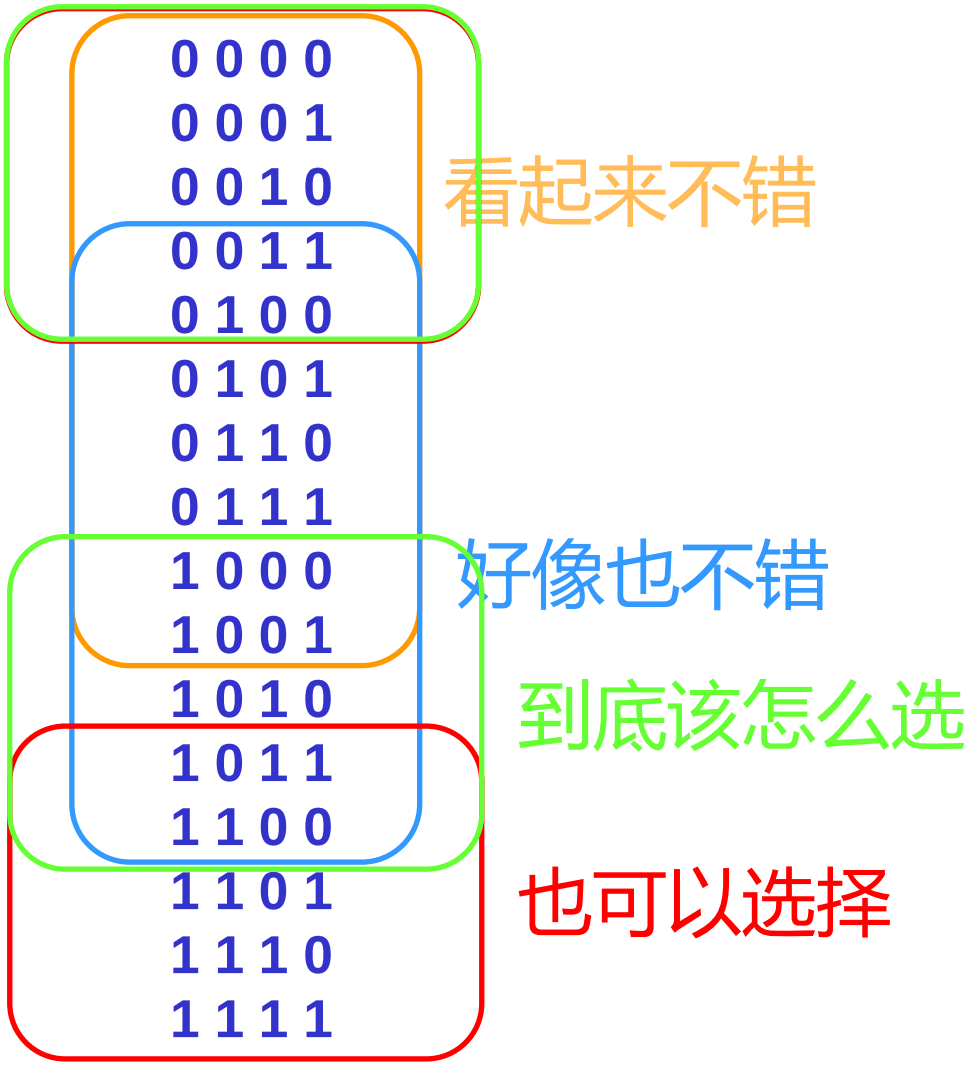
四位

如何选取有效的十个编码?



难道不是随便选?

二-十进制编码（BCD码）


0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
看起来不错
好像也不错
到底该怎么选
也可以选择

常用BCD码

有权码

无权码

十进制数	8421BCD	5421BCD	2421BCD	余3码	余3循环码
0	0000	0000	0000	0011	0010
1	0001	0001	0001	0100	0110
2	0010	0010			0111
3	0011	0011			0101
4	0100	0100	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1010		1010	1111
8	1000			1011	
9		反			
--			0101		
--	1011	0110	0110	0001	
--	1100	0111	0111	0010	0011
--	1101	1101	1000	1101	1011
--	1110	1110	1001	1110	1001
--	1111	1111	1010	1111	1000

自然码

余3码=
8421码+3

第一位自反；
其它位上下对应

自反

第一位自反，
其它位轴对称
，逻辑相邻

无效码

二-十进制编码（BCD码）

有权码-每一位都有固定的权值

- 如8421、 5421、 2421码
- 有权码所表示十进制数的大小就是各位加权相加的值

无权码-各位没有固定的权值

- 如余3码和余3循环码

二-十进制编码（BCD码）

有效编码：用以表示一位十进制数的码组

无效编码：无任何意义的码组

逻辑相邻：只有一位不同的两组编码

二-十进制编码（BCD码）

用BCD码表示十进制数时，每一位十进制数均需四位二进码表示

$(216)_{10} = (0010\ 0001\ 0110)_{8421}$

$= (0010\ 0001\ 1001)_{5421}$

$= (0101\ 0100\ 1001)_{余3}$

“0”不能省略

二-十进制编码（BCD码）

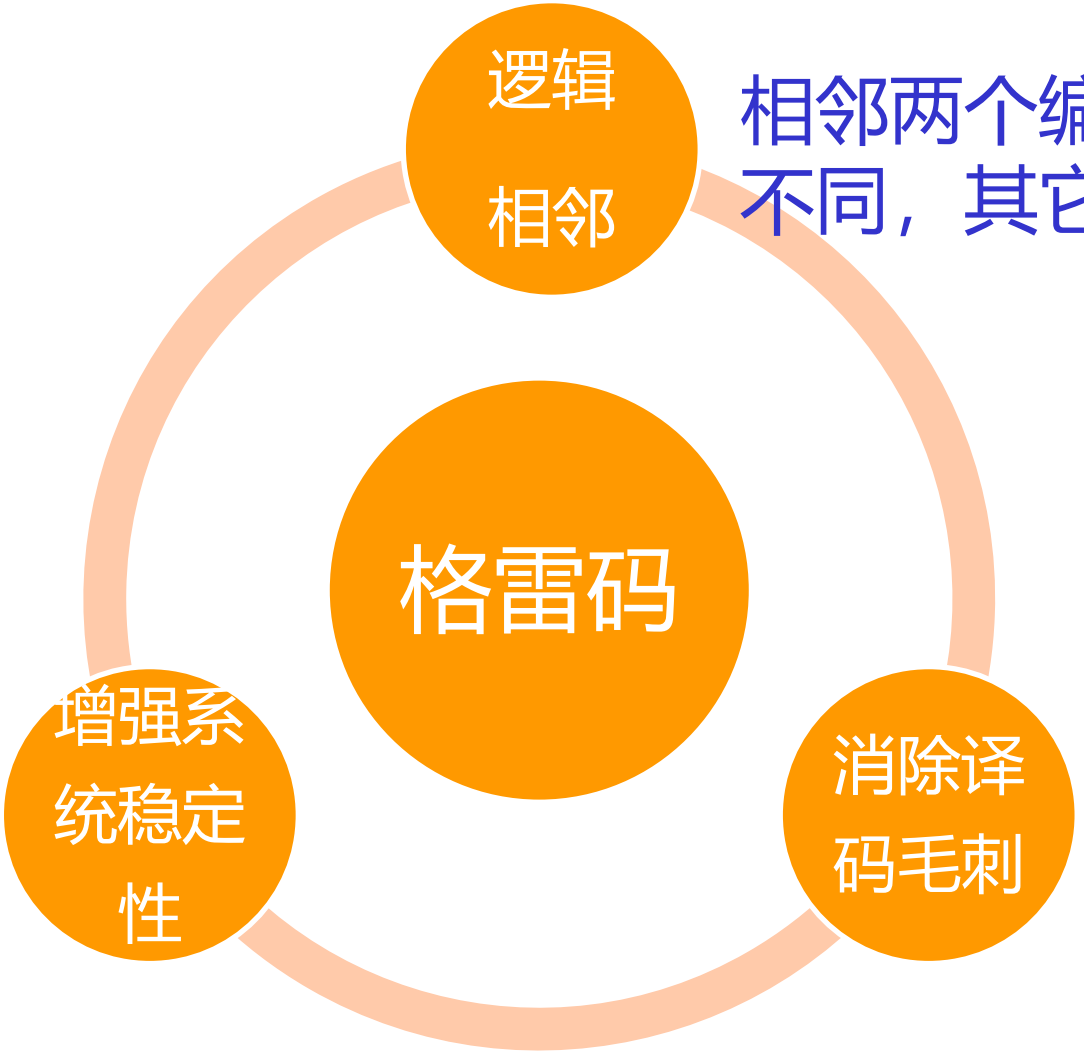
- 两个BCD码相加，结果必须还是BCD码，并且还必须是合法的BCD码

$(0010\ 0001\ 0110)_{8421}+(0011\ 1001\ 0011)_{8421}$

$=(0101\ \underline{1010}\ 1001)_{8421}$

- 其中和的第2位BCD码为1010，是非法的8421BCD码，要对其进行调整：本位加0110（10的补码），并向高位进1
- 调整后结果应为$(\underline{0110}\ \underline{0000}\ 1001)_{8421}$。

格雷（Gray）码



相邻两个编码中只有一位不同，其它各位均相同

序号	二进制码	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

余3码

余3循环码

第一位自反，其它各位关于中间轴对称

格雷（Gray）码 的特点

位置相邻的格雷码逻辑相邻

最高位自反，其它位关于中间轴对称

根据此特点可方便地写出格雷码

格雷码的写法

一位	二位		三位	
0	00	00	000	000
1	01	01	001	001
		11	011	011
		10	010	010
				110
				111
				101
				100

二进制码与格雷码转换

- 异或运算：逻辑变量A、B的取值范围为0、1，则它们的**异或**运算定义为

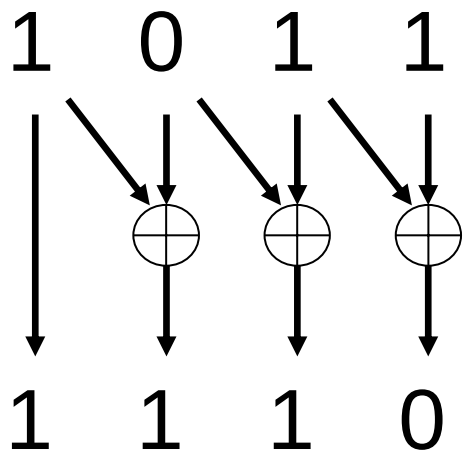
$$F=A\oplus B=\begin{cases}1 & \text{当}A\neq B\text{时}\\ 0 & \text{当}A=B\text{时}\end{cases}$$

二进制码与格雷码转换

由二进制码生成格雷码：

$G_{n-1}=B_{n-1}$，$G_i=B_{i+1}\oplus B_i$　式中i=n-2, n-3，…，2，1，0

例如：求二进制码1　0　1　1的格雷码

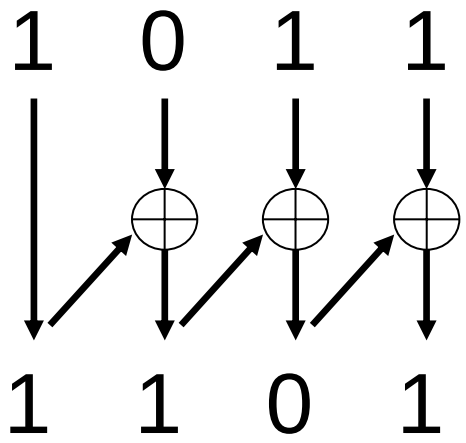


二进制码与格雷码转换

由格雷码生成二进制码：

$B_{n-1}= G_{n-1}$, $B_i=B_{i+1}\oplus G_i$ 式中i=n-2, n-3，…，2，1，0

例如：求格雷码1 0 1 1 的二进制码



ASCII符

- 除了需要表示十进制数外，还经常要表示人机交流用的其它一些信息，如大小写字母、+、-、×、÷、=、&、%…等字符，DEL、ESC、CR…等控制符
- 使用最广泛的是所谓ASCII(American Standard Codes for Information Interchange)字符集，又称为ASCII码
- 每个ASCII符号是一个7位码
- 33个控制符，95个字符，共128个

			列号（$b_6b_5b_4$）							
		B	000	001	010	011	100	101	110	111
	B	H	0	1	2	3	4	5	6	7
行号($b_3b_2b_1b_0$)	0000	0	NUL	DLE	SP	0	@	P	‘	p
	0001	1	SOH	DC1	!	1	A	Q	a	q
	0010	2	STX	DC2	“	2	B	R	b	r
	0011	3	ETX	DC3	#	3	C	S	c	s
	0100	4	EOT	DC4	$	4	D	T	d	t
	0101	5	ENQ	NAK	%	5	E	U	e	u
	0110	6	ACK	SYN	&	6	F	V	f	v
	0111	7	BEL	ETB	’	7	G	W	g	w
	1000	8	BS	CAN	(	8	H	X	h	x
	1001	9	HT	EM	)	9	I	Y	i	y
	1010	A	LF	SUB	*	:	J	Z	j	z
	1011	B	VT	ESC	+	;	K	[	k	{
	1100	C	FF	FS	,	<	L	\	l	|
	1101	D	CR	GS		=	M	]	m	}
	1110	E	SO	RS	.	>	N	^	n	~
	1111	F	SI	US	/	?	O	-	o	DEL

ASCII符

- 数字0~9的ASCII符为30H~39H
  - 字符“0”和数字0在机器中是不一样的
- A~Z的ASCII符为41H~5AH；a~z的ASCII符为61H~7AH
  - 如何进行大小写的转换？如何进行数字与字符转换？
- “BEIJING2008”的ASCII符为（每个字符用8位）4245494A494E4732303038H（８８位）
- 如果每个字符用7位，则上述字符串为1000010 1000101 1001001 1001010 1001001 1001110 1000111 0110010 0110000 0110000 0111000 B

  =10A2C9952674764C1838H（77位）

检错码和纠错码

检错码：仅能检测数字信号是否发生错误

纠错码：能够检测并纠正数字信号中的错误

奇偶校验码：除要发送的信息码之外再多发送一位奇偶校验位P，奇偶校验位一般放在最高位

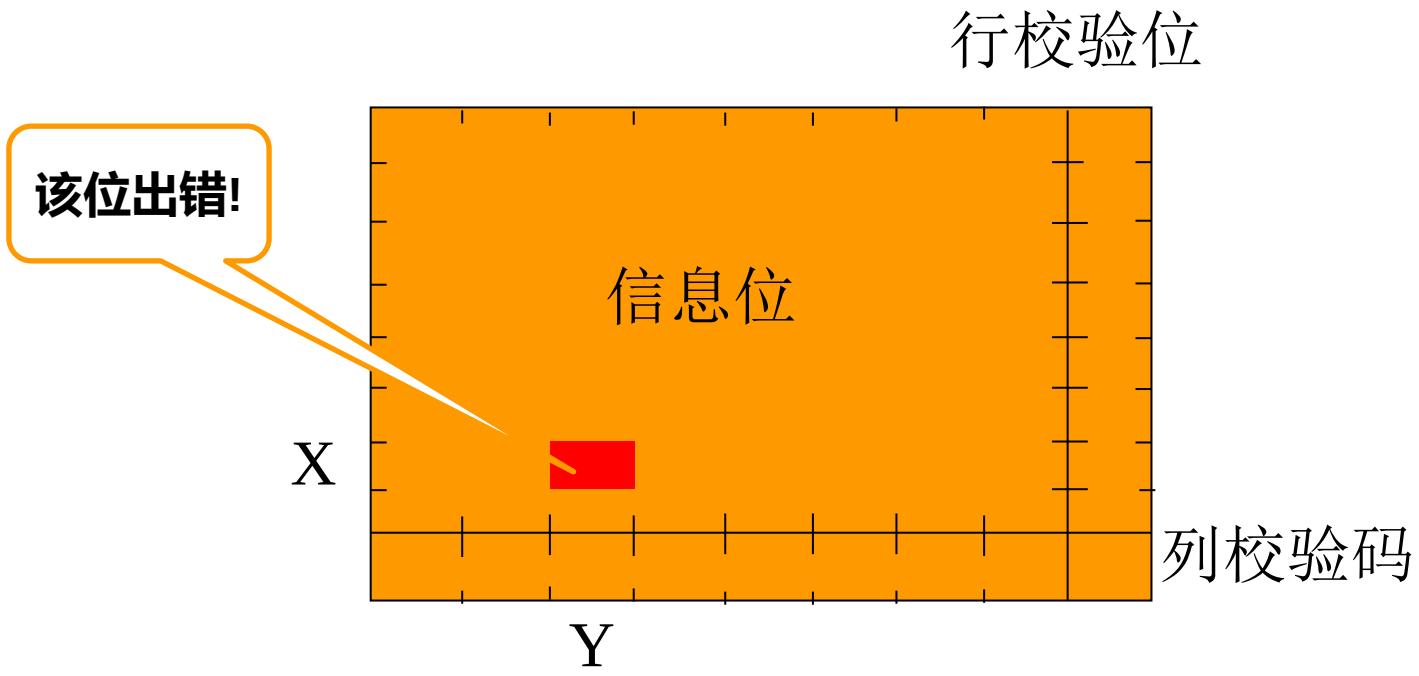
奇校验码：信息码与校验位所构成的奇偶校验码中，‘1’的个数为奇数

偶校验码：信息码与校验位所构成的奇偶校验码中，‘1’的个数为偶数

奇偶校验码

- 采用偶校验发送字符“A”
  - “A”的ASCII符为1000001
  - 加上P位使1的个数为偶数P1000001
  - 则P=0
- 传输效率η=7/8=87.5%
- 可靠性是牺牲效率换来的
- 奇偶校验只能发现奇数位错，因为偶数位错不改变码的奇偶性
- 适用于误码率较低的场合

二维奇偶纠错码



第2章小结

- 二、八、十、十六进制数及它们之间的相互转换
- 补码、反码的定义及求法
- 二进制数的计算机表示方法（符号数的表示方法）
- 利用补码进行二进制数的加减运算
- 十进制数的二进制编码(即BCD码)
- 格雷码
- ASCII字符集
- 奇偶校验码，二维奇偶纠错码