第一章 数制和编码

1.1 进位计数制

1.2 数制系统转换

1.3 有符号数表示和计算

1.4 数字编码

1.1 进位计数制

例：十进制数　　1 2 4 6 3 8 5 3 4 5 . 6 7 8 0 91

（1）权重

（2）不同的符号

（3）小数点

（4）计数规则

其中：“十” 为进位基数(Base / Radix)，简称基数(R)。

位置表示法和多项式表示法

	万位	千位	百位	十位	个位		十分位	百分位	千分位	万分位	十万分位	百万分位
	10^4	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
十进制数	1	2	3	4	5	.	6	7	8	0	9	1

$$N = (a_{n-1}a_{n-2}\cdots a_1 a_0 . a_{-1}a_{-2}\cdots a_{-m})_r$$

$$12345.67809 = 1\times10^4 + 2\times10^3 + 3\times10^2 + 4\times10^1 + 5\times10^0$$
$$+ 6\times10^{-1} + 7\times10^{-2} + 8\times10^{-3} + 0\times10^{-4} + 9\times10^{-5}$$

$$N = \sum_{i=-m}^{n-1} a_i r^i$$

$$(1011.11)_2 = (1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2})_{10}$$

$$= (8+0+2+1+0.5+0.25)_{10} = 11.75_{10}$$

R=10	R=2	R=3	R=4	R=8	R=16
0	0	0	0	0	0
1	1	1	1	1	1
2	10	2	2	2	2
3	11	10	3	3	3
4	100	11	10	4	4
5	101	12	11	5	5
6	110	20	12	6	6
7	111	21	13	7	7
8	1000	22	20	10	8
9	1001	100	21	11	9
10	1010	101	22	12	A
11	1011	102	23	13	B
12	1100	110	30	14	C
13	1101	111	31	15	D
14	1110	112	32	16	E
15	1111	120	33	17	F
16	10000	121	100	20	10
17	10001	122	101	21	11
…	…	…	…	…	…

二进制数为计算机内部运算的基础

（1）运算规则： ＋ 、－、×、÷

加法规则：$0 + 0 = 0$ $0 + 1 = 1 + 0 = 1$ $1 + 1 = 10$

减法规则：$0 - 0 = 0$ $1 - 0 = 1$ $1 - 1 = 0$ $10 - 1 = 1$(借位)

乘法规则：$0 \times 0 = 0$ $0 \times 1 = 1 \times 0 = 0$ $1 \times 1 = 1$

除法规则 $0 \div 1 = 0$ $1 \div 1 = 1$ (0不能作除数)

（2）常用的二进制常数要记住

i	R^i	i	R^i	i	R^i
-7	0.0078125	0	1	7	128
-6	0.015625	1	2	8	256
-5	0.03125	2	4	9	512
-4	0.0625	3	8	10	1024
-3	0.125	4	16	11	2048
-2	0.25	5	32	12	4096
-1	0.5	6	64	13	8192

（3）二进制数的单位：

1位二进制数=1bit、1B = 8b、

1K = 2^{10}、1M = 2^{20}、1G = 2^{30}、1T = 2^{40}

1.2 数制系统转换

$$(N)_\alpha \rightarrow (N')_\beta$$

多项式替代法（以β进制计算）和基数乘除法（以α进制计算）

$$(N)_\alpha = (\sum_{i=-m}^{n-1} A_i \times 10^i)_\alpha$$

1.2.1 多项式替代法

$$
\begin{aligned}
(1CE8)_{16} &= (1\times 10^3 + C\times 10^2 + E\times 10^1 + 8\times 10^0\ )_{16} \\
&= (1\times 16^3 + 12\times 16^2 + 14\times 16^1 + 8\times 16^0\ )_{10} \\
&= (4096+3072+224+8)_{10} \\
&= (7400)_{10}
\end{aligned}
$$

$$
\begin{aligned}
(N)_\alpha &= (A_{n-1}A_{n-2}\cdots A_1A_0.A_{-1}A_{-2}\cdots A_{-m})_\alpha \\
&= (A_{n-1}\times 10^{n-1} + A_{n-2}\times 10^{n-2} + \cdots + A_1\times 10^{1} + A_0\times 10^{0} + \\
&\qquad A_{-1}\times 10^{-1} + A_{-2}\times 10^{-2} A_{-2} + \cdots + A_{-m}\times 10^{-m})_\alpha \\
&= (A'_{n-1}\times \alpha^{n'-1} + A'_{n-2}\times \alpha^{n'-2} + \cdots + A'_1\times \alpha^{1} + A'_0\times \alpha^{0} + \\
&\qquad A'_{-1}\times \alpha^{-1} + A'_{-2}\times \alpha^{-2} + \cdots + A'_{-m}\times \alpha^{-m'})_\beta \\
&= (N')_\beta
\end{aligned}
$$

将$(121.2)_3$转换为二进制。

$$\begin{aligned}(121.2)_3 &= (1\times10^2+2\times10^1+1\times10^0+2\times10^{-1})_3\\ &= (1\times11^{10}+10\times11^1+1\times11^0+10\times11^{-1})_2\\ &= (1001+110+1+0.101010\ldots)_2\\ &= (10000.101010\ldots)_2\end{aligned}$$

将$(1234)_{10}$转换为十六进制

$$\begin{aligned}(1234)_{10} &= (1\times10^3+2\times10^2+3\times10^1+4\times10^0)_{10}\\ &= (1\times A^3+2\times A^2+3\times A^1+4\times A^0)_{16}\\ &= (\ ?\ )_{16}\end{aligned}$$

1.2.2 基数乘除法

1.整数转换（基数除法）

$$(N)_\alpha = (N')_\beta = (B_{n-1}B_{n-2}\cdots B_1B_0)_\beta$$

$$= (B_{n-1}\times 10^{n-1} + B_{n-2}\times 10^{n-2} + \cdots + B_1\times 10^1 + B_0\times 10^0)_\beta$$

$$= (B'_{n-1}\times \beta^{n-1} + B'_{n-2}\times \beta^{n-2} + \cdots + B'_1\times \beta^1 + B'_0\times \beta^0)_\alpha$$

两边同除以β $\quad B'_{n-1}\times \beta^{n-2} + B'_{n-2}\times \beta^{n-3} + \cdots + B'_1\times \beta^0 + B'_0/\beta$

得到余数 B'_0

将十进制的179 转换成二进制数

$179 \div 2 = 89 \ \ldots\ \ldots$ 余1（b_0）

$89 \div 2 = 44 \ \ldots\ \ldots$ 余1（b_1）

$44 \div 2 = 22 \ \ldots\ \ldots$ 余0（b_2）

$22 \div 2 = 11 \ \ldots\ \ldots$ 余0（b_3）

$11 \div 2 = 5 \ \ldots\ \ldots$ 余1（b_4）

$5 \div 2 = 2 \ \ldots\ \ldots$ 余1（b_5）

$2 \div 2 = 1 \ \ldots\ \ldots$ 余0（b_6）

$1 \div 2 = 0 \ \ldots\ \ldots$ 余1（b_7）

$179 = (10110011)_2$

将十进制的3417 转换成十六进制数。

3417÷16=213余（9）

213÷16 =13 余（5）

13÷16 =0 余（13）

$(3417)_{10} = (D59)_{16}$

2. 小数转换（基数乘法）

$$
\begin{aligned}
(N)_\alpha &= (N')_\beta \\
&= (0.B_{-1}B_{-2}\cdots B_{-m})_\beta \\
&= (B_{-1}\times 10^{-1} + B_{-2}\times 10^{-2} + \cdots + B_{-m}\times 10^{-m})_\beta \\
&= (B'_{-1}\times \beta^{-1} + B'_{-2}\times \beta^{-2} + \cdots + B'_{-m}\times \beta^{-m})_\alpha
\end{aligned}
$$

两边同乘以β $B'_{-1} + B'_{-2}\times \beta^{-1} + \cdots + B'_{-m}\times \beta^{-m+1}$

得到整数部分 B'_{-1}

将 $(0.375)_{10}$ 转换成二进制数。

$$
\begin{array}{l}
\quad 0\,.\,3\,7\,5 \\
\times \qquad\quad 2 \\
\hline
[0]\,.\,7\,5\,0 \qquad \dots\dots \quad B_{-1} = 0 \\
\times \qquad\quad 2 \\
[1]\,.\,5\,0\,0 \qquad \dots\dots \quad B_{-2} = 1 \\
\hline
\times \qquad\quad 2 \\
[1]\,.\,0\,0\,0 \qquad \dots\dots \quad B_{-3} = 1
\end{array}
$$

$$(0.375)_{10} = (0.011)_2$$

将$(0.4321)_{10}$转换成十六进制数。

$N_0 = 0.4321$

$N_0 \times \beta = 0.4321 \times 16 = 6.9136$ $N_1 = 0.9136$ $B_{-1} = 6$

$N_1 \times \beta = 0.9136 \times 16 = 14.6176$ $N_2 = 0.6176$ $B_{-2} = 14(E)$

$N_2 \times \beta = 0.6176 \times 16 = 9.8816$ $N_3 = 0.8816$ $B_{-3} = 9$

$N_3 \times \beta = 0.8816 \times 16 = 14.1056$ $N_4 = 0.1056$ $B_{-4} = 14(E)$

即 $(0.4321)_{10} \approx (0.6E9E)_{16}$

将$(1023.231)_4$转换成五进制数。

1.2.3 任意两种进制之间的转换

$$(N)_\alpha \to (N')_\beta$$

1. 若熟悉α进制的运算规则，则采用基数乘除法完成转换；
2. 若熟悉β进制的运算规则，则采用多项式替代法完成转换；
3. 若不熟悉α、β进制的运算规则：则可利用十进制作为转换桥梁

将$(1023.231)_4$转换成五进制数。

$$
\begin{aligned}
&(1023.231)_4 \\
&= (1\times10^3+0\times10^2+2\times10^1+3\times10^0+2\times10^{-1}+3\times10^{-2}+1\times10^{-3})_4 \\
&= (1\times4^3+0\times4^2+2\times4^1+3\times4^0+2\times4^{-1}+3\times4^{-2}+1\times4^{-3})_{10} \\
&= 64+0+8+3+0.5+0.1875+0.015625 \\
&= 75.703125
\end{aligned}
$$

$(75.703125)_{10}$

整数部分

$$
\begin{array}{r|l}
5 & 75 \quad \cdots\cdots\ b_0=0 \\ \hline
5 & 15 \quad \cdots\cdots\ b_1=0 \\ \hline
5 & \ \ 3 \quad \cdots\cdots\ b_2=3 \\ \hline
 & \ \ 0
\end{array}
$$

小数部分

$0\ .\ 703125$	
$\times \qquad\quad 5$	
$[3]\ .\ 515625$	$\cdots\cdots\ C_{-1}=3$
$\times \qquad\quad 5$	
$[2]\ .\ 578125$	$\cdots\cdots\ C_{-2}=2$
$\times \qquad\quad 5$	
$[2]\ .\ 890625$	$\cdots\cdots\ C_{-3}=2$
$\times \qquad\quad 5$	
$[4]\ .\ 453125$	$\cdots\cdots\ C_{-4}=4$

$$(75.703125)_{10} \approx (300.3224)_5$$

$$\therefore\ (1023.231)_4 \approx (300.3224)_5$$

1.2.4 直接转换法

二进制***Binary***，简称**B**，如 $(10)_2 = (10)_B$ ；

八进制***Octal***，简称**O** ，如 $(10)_8 = (10)_O$ ；

十六进制***Hexadecimal***，简称**H** ，如 $(10)_{16} = (10)_H$ 。

$$(N)_\alpha \to (N')_\beta$$

R=2	R=4	R=8	R=16
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	10	4	4
101	11	5	5
110	12	6	6
111	13	7	7
1000	20	10	8
1001	21	11	9
1010	22	12	A
1011	23	13	B

R=2	R=4	R=8	R=16
1100	30	14	C
1101	31	15	D
1110	32	16	E
1111	33	17	F
10000	100	20	10
10001	101	21	11
10010	102	22	12
10011	103	23	13
10100	110	24	14
10101	111	25	15
10110	112	26	16
…	…	…	…

$$(N)_2 \rightarrow (N')_{2^k}$$

$$(\mathbf{N})_2 = (\mathbf{K}_{n-1}\,\mathbf{K}_{n-2}\ldots\mathbf{K}_1\,\mathbf{K}_0\,.\,\mathbf{K}_{-1}\,\mathbf{K}_{-2}\ldots\mathbf{K}_{-m})_2$$

（1）位置计数法表示数

（2）以小数点为中心，分别**向左**、**向右**分组，每k位一组

（3）补零

（4）转换

将$(11111010.0111)_2$转换为八进制数。

$$(11111010.0111)_2 = (372.34)_8$$

$$(N)_{2^k} \to (N')_2$$

将$(213.01)_4$转换为二进制数。

$$(213.01)_4 = (100111.0001)_2$$

将$(AF.16C)_{16}$转换为八进制数。

$$(AF.16C)_{16} = (257.0554)_8$$

1.2.5 数制转换时小数位数的确定

小数部分不能精确转换时，换后的小数部分应是怎样的？

① 小数位数受机器字长的限制而确定；

② 由需求给定小数的位数；

③ 保证转换成β进制后维持与α进制相同的精度。

1.3 有符号数表示和计算

真值	符号位	数值位
＋5 －7	＋ －	5 7

1.3.1 原码

原码的形成规则：符号位+数值位

用8位二进制代码表示的原码

$x = +5$

$[x]_{原} = 00000101$

$y = -7$

$[y]_{原} = 10000111$

$$[x]_{原} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ 2^{n-1} - x, & -2^{n-1} < x \le 0 \end{cases}$$

$[x]_{原} = 01010101$ $[x]_{原} = 11010101$

$x = +85$ $x = -85$

$[x]_{原} = 01111111$ $[x]_{原} = 11111111$

$x = +127$ $x = -127$

$[x]_{原} = 00000000$ $[x]_{原} = 10000000$

$x = +0$ $x = -0$

1.3.2 反码

反码的形成规则：符号位+数值位

用8位二进制代码表示

x = ＋5

$[x]_{原}$ = 00000101 $[x]_{反}$ = 00000101

y = －7

$[y]_{原}$ = 10000111 $[y]_{反}$ = 11111000

$$[x]_{反} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ (2^n - 1) + x, & -2^{n-1} < x \le 0 \end{cases}$$

$[x]_{反}$ = 01010101 $[x]_{反}$ =11010101

x =＋85 x =－42

$[x]_{反}$ = 01111111 $[x]_{反}$ = 11111111

x =＋127 x =－0

$[x]_{反}$= 00000000 $[x]_{反}$= 10000000

x =＋0 x =－127

1.3.3 补码

补码的形成规则：符号位+数值位

用8位二进制代码表示

x = +5

$[x]_{原}$ = 00000101 $[x]_{补}$ = 00000101

y = −7

$[y]_{原}$ = 10000111 $[y]_{补}$ = 11111001

$$[x]_{补} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ 2^n + x, & -2^{n-1} \le x < 0 \end{cases}$$

$[x]_{补}$ = 01010101 $[x]_{补}$=11010101

x =+85 x =−43

$[x]_{补}$= 01111111 $[x]_{补}$= 11111111

x =+127 x =−1

$[x]_{补}$= 00000000 $[x]_{补}$= 10000000

x =+0 x =−128

$$[x]_{原} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ 2^{n-1} - x, & -2^{n-1} < x \le 0 \end{cases}$$

$$[x]_{补} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ 2^{n} + x, & -2^{n-1} \le x < 0 \end{cases}$$

$$[x]_{反} = \begin{cases} x, & 0 \le x < 2^{n-1} \\ (2^{n} - 1) + x, & -2^{n-1} < x \le 0 \end{cases}$$

$$[x]_{补} = 2^{n} - N = (2^{n} - 1 - N) + 1$$

N：原码的数值位

Signed Decimal	Sign Magnitude Binary	Two's Complement System	One's Complement System
+15	0,1111	0,1111	0,1111
+14	0,1110	0,1110	0,1110
+13	0,1101	0,1101	0,1101
+12	0,1100	0,1100	0,1100
+11	0,1011	0,1011	0,1011
+10	0,1010	0,1010	0,1010
+9	0,1001	0,1001	0,1001
+8	0,1000	0,1000	0,1000
+7	0,0111	0,0111	0,0111
+6	0,0110	0,0110	0,0110
+5	0,0101	0,0101	0,0101
+4	0,0100	0,0100	0,0100
+3	0,0011	0,0011	0,0011
+2	0,0010	0,0010	0,0010
+1	0,0001	0,0001	0,0001
0	0,0000	0,0000	0,0000
	(1,0000)	–	(1,1111)
−1	1,0001	1,1111	1,1110
−2	1,0010	1,1110	1,1101
−3	1,0011	1,1101	1,1100
−4	1,0100	1,1100	1,1011
−5	1,0101	1,1011	1,1010
−6	1,0110	1,1010	1,1001
−7	1,0111	1,1001	1,1000
−8	1,1000	1,1000	1,0111
−9	1,1001	1,0111	1,0110
−10	1,1010	1,0110	1,0101
−11	1,1011	1,0101	1,0100
−12	1,1100	1,0100	1,0011
−13	1,1101	1,0011	1,0010
−14	1,1110	1,0010	1,0001
−15	1,1111	1,0001	1,0000
−16	–	1,0000	–

1.3.4 有符号数的加、减运算

原码 加减法有不同的规则，关键是要判大小；

反码 $[x+y]_{反}=[x]_{反}+[y]_{反}$

$[x-y]_{反}=[x]_{反}+[-y]_{反}$

补码 $[x+y]_{补}=[x]_{补}+[y]_{补}$

$[x-y]_{补}=[x]_{补}+[-y]_{补}$

反码和补码的减法运算按加法运算完成

符号位S被看成一位数码，与数值位按同样的加法规则进行处理

进位的处理：补码运算时符号位产生的进位要丢掉；反码运算时符号位产生的进位加到和数最低位

求 $z = x - y$，其中 $x = +1010$，$y = +0011$

原码运算： $[x]_{原} = 01010$　　$[y]_{原} = 00011$

x 绝对值 > y 绝对值

$[z]_{原} = [01010 - 00011]_{原} = 00111$　　$z = +0111$

补码运算： $[x]_{补} = 01010$　　$[-y]_{补} = [-0011]_{补} = 11101$

$$
\begin{array}{r}
01010 \\
+11101 \\
\hline
1\ 00111
\end{array}
$$

$[z]_{补} = 00111$

$z = +0111$

反码运算： $[x]_{反} = 01010$ $[-y]_{反} = [-0011]_{反} = 11100$

$$
\begin{array}{r}
01010 \\
+11100 \\
\hline
1\ 00110 \\
+\quad\quad 1 \\
\hline
00111
\end{array}
$$

$[z]_{反} = 00111$

$z = +0111$

求 $z = x - y$， 其中 $x = +0011$， $y = +1010$

原码运算： $[x]_{原} = 00011$ $[y]_{原} = 01010$

x 绝对值 $<$ y 绝对值

$[z]_{原} = [-(01010 - 00011)]_{原} = -00111$ $z = -0111$

补码运算： $[x]_{补} = 00011$ $[-y]_{补} = [-1010]_{补} = 10110$

$$
\begin{array}{r}
00011 \\
+10110 \\
\hline
11001
\end{array}
$$

$[z]_{补} = 11001$

$z = -0111$

反码运算： $[x]_反 = 00011$ $[-y]_反 = [-1010]_反 = 10101$

$$\begin{array}{r} 00011 \\ +10101 \\ \hline 11000 \end{array}$$

$[z]_反 = 11000$

$z = -0111$

进位的处理：补码运算时符号位产生的进位要丢掉

1、两正数相加，和小于2^{n-1}

2、两正数相加，和大于2^{n-1}（溢出）

3、正数与负数相加（负数绝对值大）

4、正数与负数相加（正数绝对值大）

5、两负数相加，和绝对值小于等于2^{n-1}

6、两负数相加，和绝对值大于2^{n-1}（溢出）

$$-A+B(B>A): A^*+B=(2^n-A)+B=2^n+(B-A)>2^n$$

$$-A-B(A+B\le 2^{n-1}): A^*+B^*=(2^n-A)+(2^n-B)=2^n+2^n-(A+B)$$

进位的处理：反码运算时符号位产生的进位加到和数最低位

1、两正数相加，和小于2^{n-1}

2、两正数相加，和大于2^{n-1}

3、正数与负数相加（负数绝对值大）

4、正数与负数相加（正数绝对值大）

5、两负数相加，和绝对值小于等于2^{n-1}

6、两负数相加，和绝对值大于2^{n-1}

$$-A+B(B>A):\overline{A}+B=(2^n-1-A)+B=2^n+(B-A)-1$$

$$-A-B(A+B\le 2^{n-1}):\overline{A}+\overline{B}=(2^n-1-A)+(2^n-1-B)=2^n+[2^n-1-(A+B)]-1$$

计算机采用什么码进行整数计算？

1.4 数字编码

1.4.1 十进制数的常用代码（二进制编码的十进制数）

用四位二进制数的代码表示一位十进制数，既具有二进制数的形式，又具有十进制数的特点；

“8421”码（BCD码）、“2421”码、余3码、格雷码……

“制”表示方法。码制：编码方法；数制，计数方法。

十进制整数	8421码	2421码	余3码
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100
无效码区 Unused code wrds	1010、1011、1100、1101、1110、1111	0101、0110、0111、1000、1001、1010	0000、0001、0010、1101、1110、1111

四位二进制代码	8421码	2421码	余3码
0000	0000 0	0000 0	0000
0001	0001 1	0001 1	0001 非码区
0010	0010 2	0010 2	0010
0011	0011 3	0011 3	0011 0
0100	0100 4	0100 4	0100 1
0101	0101 5	0101	0101 2
0110	0110 6	0110	0110 3
0111	0111 7	0111 非码区	0111 4
1000	1000 8	1000	1000 5
1001	1001 9	1001	1001 6
1010	1010	1010	1010 7
1011	1011	1011 5	1011 8
1100	1100 非码区	1100 6	1100 9
1101	1101	1101 7	1101
1110	1110	1110 8	1110 非码区
1111	1111	1111 9	1111

代码表示为 $A_3 A_2 A_1 A_0$

代码	对应的十进制数值	代码直接按位转换
8421码	有权码 (*Weighted code*) $8A_3+4A_2+2A_1+1A_0$	$(13)_{10}=(00010011)_{BCD}$ $(101110101010000)_{BCD}=(1750)_{10}$
2421码	有权码、对9自补码 $2A_3+4A_2+2A_1+1A_0$	$(13)_{10}=(00010011)_{2421}$ $(1110110110000)_{2421}=(1750)_{10}$
余3码	无权码、对9自补码 $8A_3+4A_2+2A_1+1A_0$ -0011	$(13)_{10}=(01000110)_{余3}$ $(100101010000011)_{余3}=(1750)_{10}$

1.4.2 可靠性编码

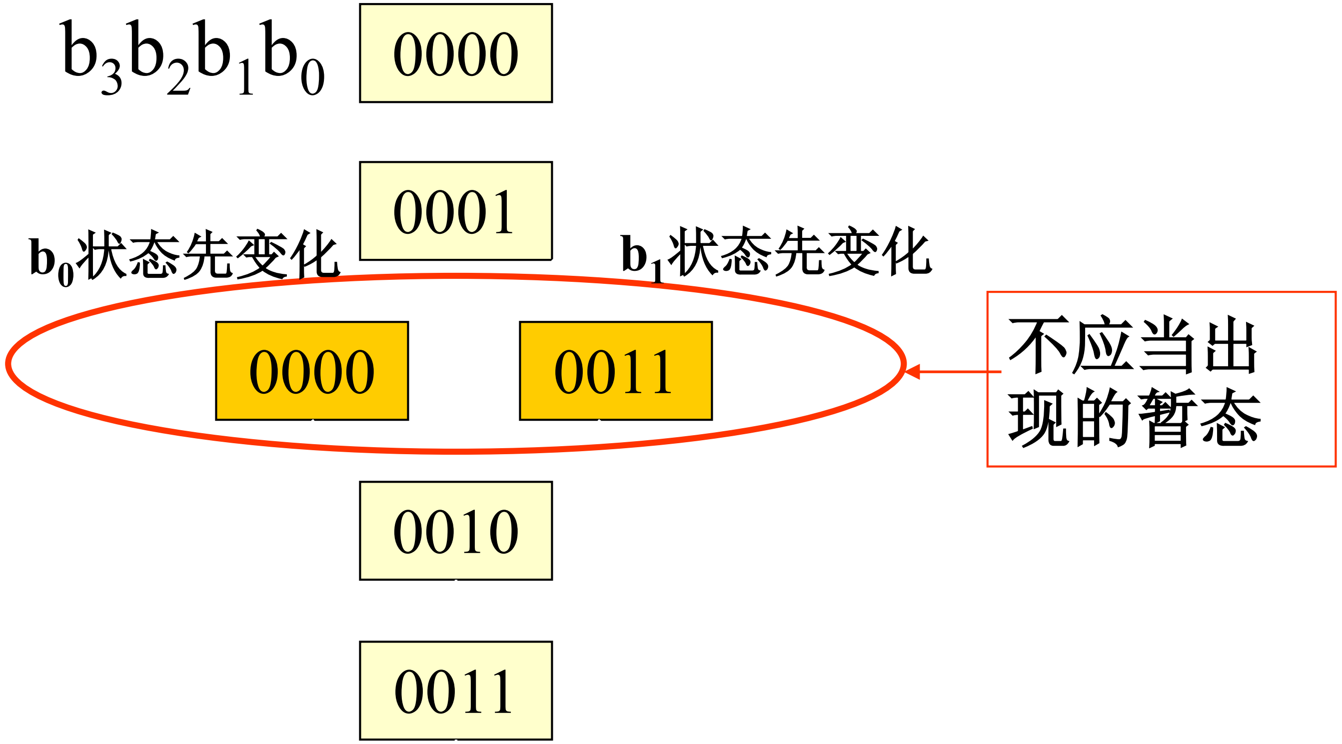
解决代码在形成或传输过程中可能会发生的错误，提高系统的安全性。

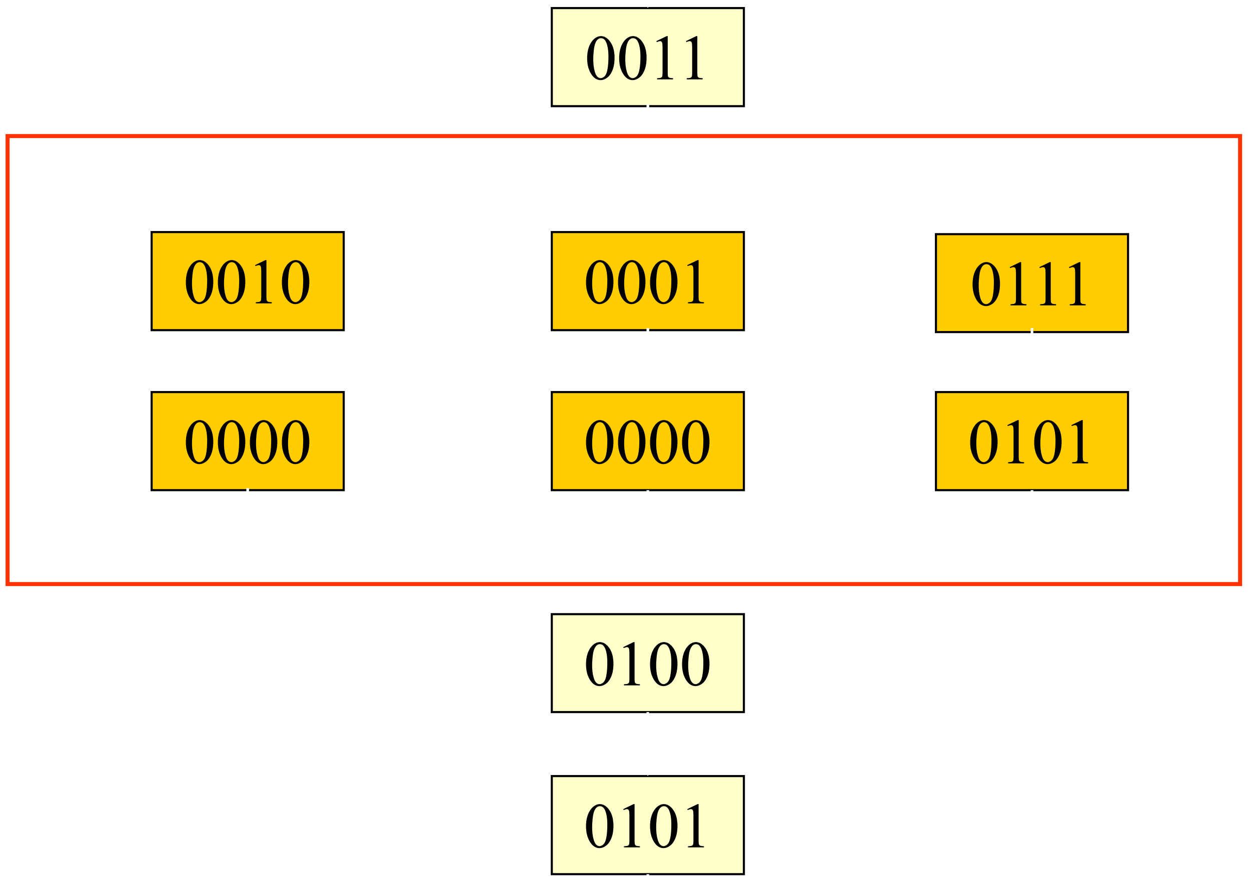
1. 使代码自身具有一种特征或能力；
2. 增加信息位之间的运算，如异或运算 $\oplus$；
3. 增加校验位 。

格雷码（Gray)

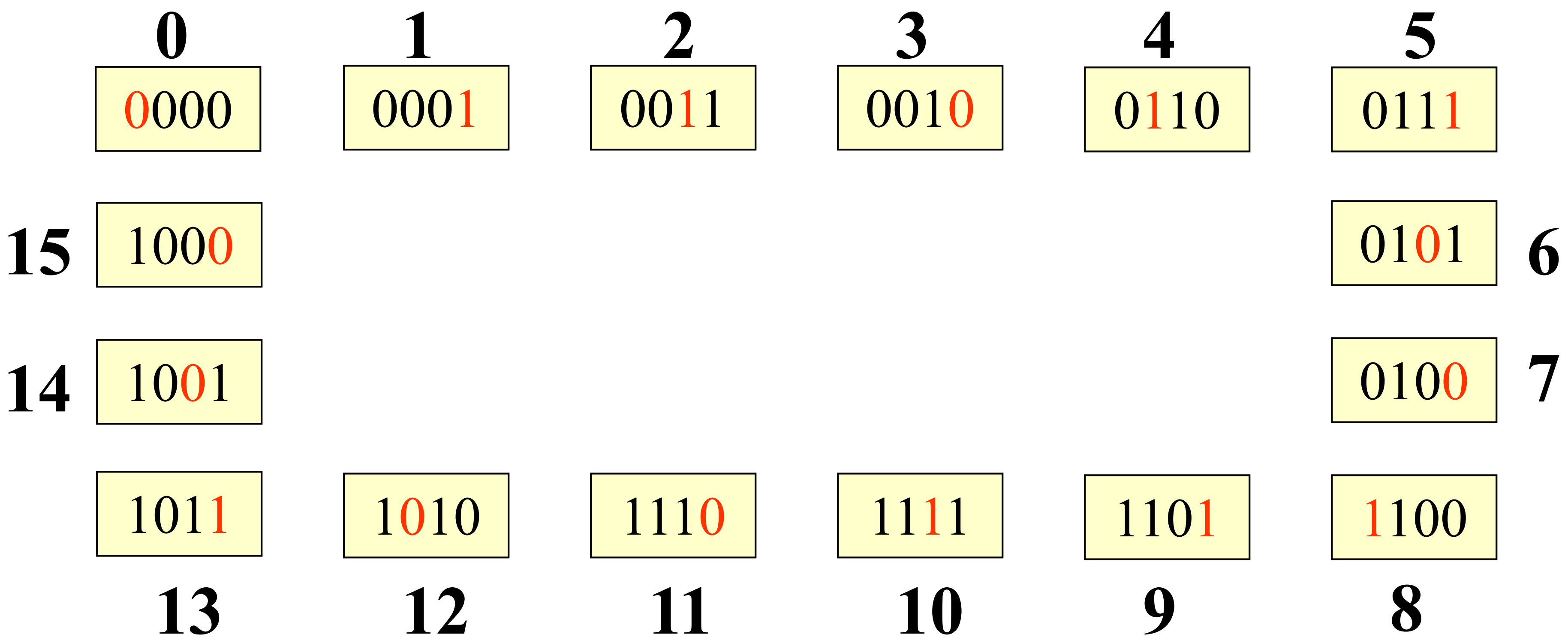
特点：任意两个相邻数的代码只有一位二进制数不同

- 例：四位二进制加1计数器，工作时有如下情况出现：





这种情况出现的最为严重的是当由1111加1计数到0000时


0
0000
1
0001
2
0011
3
0010
4
0110
5
0111
6
0101
7
0100
8
1100
9
1101
10
1111
11
1110
12
1010
13
1011
14
1001
15
1000

典型Gray码

Gray码、步进码和二进制码对照表

十进制数	二进制数	典型Gray	十进制Gray码(1)	十进制Gray码(2)	步进码
0	0000	0000	0000	0000	00000
1	0001	0001	0001	0001	00001
2	0010	0011	0011	0011	00011
3	0011	0010	0010	0010	00111
4	0100	0110	0110	0110	01111
5	0101	0111	1110	0111	11111
6	0110	0101	1010	0101	11110
7	0111	0100	1011	0100	11100
8	1000	1100	1001	1100	11000
9	1001	1101	1000	1000	10000
10	1010	1111			
11	1011	1110			
12	1100	1010			
13	1101	1011			
14	1110	1001			
15	1111	1000			

典型Gray码通过异或运算⊕完成：

$$G_i = B_{i+1} \oplus B_i$$

$$B_{n+1} = 0$$

二进制码 B	**0**	1	1	1	0
Gray码 G	**1**	0	0	1	

典型Gray码转换到二进制码：

$$G_i = B_{i+1} \oplus B_i$$

$$G_i \oplus B_{i+1} = B_{i+1} \oplus B_i \oplus B_{i+1}$$

$$B_i = G_i \oplus B_{i+1}$$

$$B_{n+1} = 0$$

$$B_i = G_i \oplus B_{i+1}$$

Gray码	G	1	1	1	0
二进制码	B	1	0	1	1

$$B_i = G_i \oplus B_{i+1}$$

$$B_{n-1} = G_{n-1} \oplus 0 = G_{n-1}$$

$$B_{n-2} = G_{n-2} \oplus B_{n-1} = G_{n-2} \oplus G_{n-1}$$

$$\vdots$$

$$B_0 = G_{n-1} \oplus G_{n-2} \oplus \cdots \oplus G_0$$

G	1	1	1	0
B	1	0	1	1

改异或运算电路为判别电路

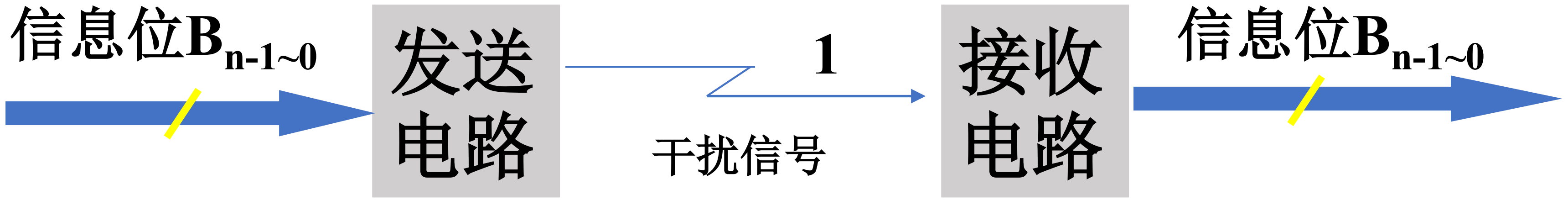
$$B_i = G_{n-1} \oplus G_{n-2} \oplus \cdots \oplus G_i$$

G = 1 1 0 1 0 0 1 0 1

B = 1 0 0 1 1 1 0 0 1

校验码和纠错码

传输系统电路



解决方法：增加校验位P（⊕）

1、奇偶校验码

校验码：

信息位 $\mathbf{B_{n-1\sim0}}$	校验位 **P**

- 偶校验：

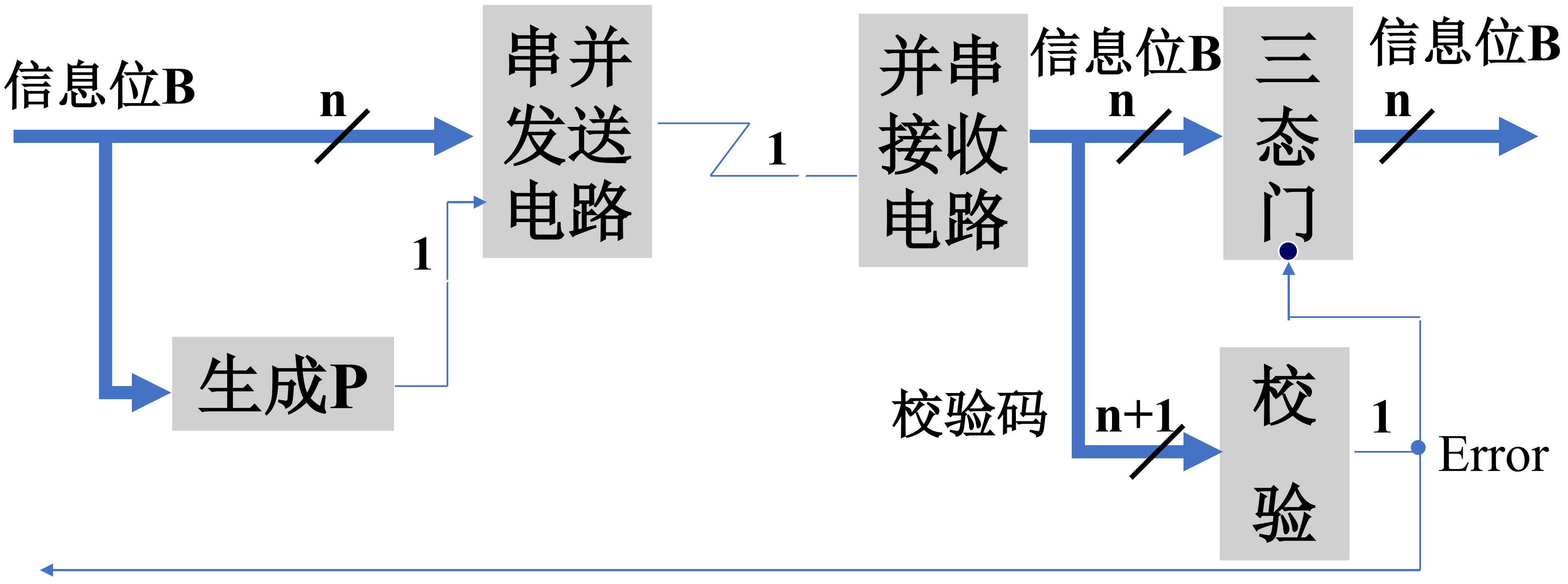
  P的取值使校验码中'1'的个数是偶数；

  $$P_{偶} = B_{n-1} \oplus B_{n-2} \oplus \ldots \oplus B_1 \oplus B_0$$

- 奇校验：

  P的取值使校验码中'1'的个数是奇数；

  $$P_{奇} = B_{n-1} \oplus B_{n-2} \oplus \ldots \oplus B_1 \oplus B_0 \oplus 1$$



奇偶校验码具有发现一位错的能力

$$\text{Error}_{偶} = B_{n-1} \oplus B_{n-2} \oplus \ldots \oplus B_1 \oplus B_0 \oplus P_{偶}$$

2、海明校验码

不仅能检测出单错，还能校正单错

以四位信息位$B_4B_3B_2B_1$为例，在传输前生成它的海明校验码：

(1) 位序：　7　6　5　4　3　2　1

$B_4\ B_3\ B_2\ P_3\ B_1\ P_2\ P_1$

(2) 校验位的生成公式：

$$P_3 = B_4 \oplus B_3 \oplus B_2$$

$$P_2 = B_4 \oplus B_3 \oplus B_1$$

$$P_1 = B_4 \oplus B_2 \oplus B_1$$

“8421”海明码

位序 N	7	6	5	4	3	2	1
	B_4	B_3	B_2	P_3	B_1	P_2	P_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

(3) 校验和：$S_3 = B_4 \oplus B_3 \oplus B_2 \oplus P_3$

$$S_2 = B_4 \oplus B_3 \oplus B_1 \oplus P_2$$
$$S_1 = B_4 \oplus B_2 \oplus B_1 \oplus P_1$$

$S_3\,S_2\,S_1 = 0$时，接收到的信息是正确的；

否则，$S_3\,S_2\,S_1$所表示的二进制值就是出错的那一位的位序值。

接收到的海明码为：

7	6	5	4	3	2	1
B_4	B_3	B_2	P_3	B_1	P_2	P_1
0	0	0	1	0	1	0

$S_3\,S_2\,S_1$ =110，表示第6位(B_3)出错，改0为1。

出错表的确定

S_3 =	$B_4 \oplus B_3 \oplus B_2 \oplus P_3$						
S_2 =	$B_4 \oplus B_3 \oplus B_1 \oplus P_2$						
S_1 =	$B_4 \oplus B_2 \oplus B_1 \oplus P_1$						
$S_3\ S_2\ S_1$	111	110	101	100	011	010	001 000
出错位序列	**7**	**6**	**5**	**4**	**3**	**2**	**1**
出错位	B_4	B_3	B_2	P_3	B_1	P_2	P_1

设：信息位n位，校验位k位

则 $(2^k-1)-k \geq n$

或 $(2^k-1) \geq n+k$

如下表所列：

校验位数k	1	2	3	4	5	6	7	8
最大信息位数n	0	1	4	11	26	57	120	247
海明码位数 (2^k-1)	1	3	7	15	32	63	127	255

第一章　作业

1、用实例说明摩尔定律

2、说明计算机中浮点数的表示格式和进行四则运算的方法

3、课后习题：1.1、1.6、1.8、1.16、1.23