第三章 信道编码

3.4 循环码

（第12讲　2007.11.13.）

本节的主要内容

- 码多项式
- 循环移位的数学表达
- 循环码的生成多项式
- 循环码的编码
- 循环码的译码
- 编、译码的电路实现

外语关键词

循环码：cyclic code

码多项式：code polynomial

生成多项式：generator polynomial

求模运算：modular arithmetic

系统码：systematic（regular）code

循环移位运算：cycle shift operation

上节回顾：线性分组码

- 基本概念：

表达方式：(n,k)码，k是信息位数，r是监督位数，n=k+r是码长。

- 编码：已知信息K（k位二进序列），求相应码字的方法是 C=KG，G叫生成矩阵，是k行n列的， 一般G具有$[I_k\ Q]$的形式， I_k 是k行k列单位方阵，Q是k行r列的矩阵。

生成矩阵的设计，应使许用码字之间的最小汉明距离尽量地大。

- 译玛：当收到码字R时，首先计算伴随子向量：$S=RH^T$；若S=0，则R=C为正确码字；若$S \neq 0$，则R≠C为错误码字。

  这里H叫一致监督矩阵，是r行n列的。一般H具有[P I_r]的形式，I_r 是r行r列单位方阵，P是r行k列的矩阵，P与Q互为转置关系。

- 纠正1位错：当$S \neq 0$时，由$S=R \cdot H^T$ 求出S，比较S 与H^T，H^T的那一行与S相同，相应的错误格式向量E的那一位就等于1。于是R的那一位就是错误的，根据C=R+E进行将其纠正。

- 纠正多位错错：当$S \neq 0$时，根据$S=R\cdot H^T = E\cdot H^T$，可以预先由$S=E\cdot H^T$计算出各种错误格式$E$所对应的伴随子向量S，得到E~S对照表。查表就能找到接收码字R（即$S=R\cdot H^T$）所对应的E。
- 纠错能力不等式：

$$2^r \geqslant C_n^{\,o}+C_n^{\,1}+C_n^{\,2}+\cdots\cdots+C_n^{\,t}$$

这是因为伴随子S是1行r列的向量，它有2^r种不同的状态，除了用全零态表示正确码之外，最多只能区别开2^r-1种不同的错误格式。

3.4 循环码

- 引言：

构造线性分组码关键是设计出一个好的生成矩阵，使所有码字之间的汉明距离尽量大。怎样找这样的矩阵呢？循环码的出现提供了一整套理论和方法，使人们能够借助数学工具来寻找更好的线性分组码，并由此引发出一大类很常用检、纠错编译码。

- 不难发现 上节所讨论过的(7, 3）线性分组码具有循环移位特性：

C_0=(0000000);　　C_0=(0000000);

C_1=(0011101);　　C_1=(0011101);

C_2=(0100111);　　C_3=(0111010);

C_3=(0111010);　　C_7=(1110100);

C_4=(1001110);　　C_6=(1101001);

C_5=(1010011);　　C_5=(1010011);

C_6=(1101001);　　C_2=(0100111);

C_7=(1110100);　　C_4=(1001110);

- 循环码是线性分组码中的一个子集。
- 对于循环码，有了一个的码字，按循环移位规律就能写出n个码字。从中选出k个来构造生成矩阵G，就能生成全部2^k个许用码字。
- 循环码与近代数学有密切联系。由此我们便可以借助数学工具来设计编码。

3.4.1 码多项式

- 二进制自然码可表达为以2为底的多项表达形式，如：

  C = (1010111)=

  $=1\times2^6+0\times2^5+1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0$ ；

- 把底换为x，则得到“码多项式”：

  $C(x)=1\cdot x^6+0\cdot x^5+1\cdot x^4+0\cdot x^3+1\cdot x^2+1\cdot x^1+1\cdot x^0$

  $=x^6+x^4+x^2+x+1$

- 码长为n时，可写：

  $C(x)=c_{n-1}x^{n-1}+c_{n-2}x^{n-2}+\cdots\cdots+c_1\times x^1+c_0x^0$

- 如三位二元码的8个码字对应的码多项式为：

  000，001，010，011，100，101，110，111；

  0，1，x，$x+1$，x^2，x^2+1，x^2+x，x^2+x+1

3.4.2 循环移位的数学表达

- 对二进数，左移一位相当于乘以2，而将最高位的进位(2 n位)上的数码拿回到2 0位，叫做循环移位，相当于作模2^n-1运算。

  如：101→1010→011 实际是 5×2 mod 7 = 3

- 码多项式的循环移位，实际是乘x后作模 x^n-1运算。

  如： 1100010 → 11000100 → 1000101

$$(x^6+x^5+x) \rightarrow x(x^6+x^5+x) \mod (x^7-1)$$

$$= (x^7+x^6+x^2) \mod (x^7-1) = x^6+x^2+1$$

用循环移位得到（7, 4）循环码的码多项式

序号	信息	循环码	循环次数	码多项式	模x^7-1运算后
1	0001	0001011	0	x^3+x+1	x^3+x+1
2	0010	0010110	1	$x(x^3+x+1)$	x^4+x^2+x
3	0101	0101100	2	$x^2(x^3+x+1)$	$x^5+x^3+x^2$
4	1011	1011000	3	$x^3(x^3+x+1)$	$x^6+x^4+x^3$
5	0110	0110001	4	$x^4(x^3+x+1)$	x^5+x^4+1
6	1100	1100010	5	$x^5(x^3+x+1)$	x^6+x^5+x
7	1000	1000101	6	$x^6(x^3+x+1)$	x^6+x^2+1

3.4.3 循环码的生成多项式

- 循环码的码多项式中幂次最低的非零多项式叫做生成多项式，记做$g(x)$。如(7,4)码的x^3+x+1。有了它，其它码字都可由$x^i \cdot g(x)$ 的模 x^n-1得到。
- 生成多项式的常数项为1。否则，通过循环移位还能继续降低幂次，它就不是幂次最低的多项式了。
- 生成多项式的幂次为r。因为幂次最低的码多项式是信息为000……01,后面跟上r个监督位的那个码字所对应的码多项式，它的最高位是 x^r，是r次的多项式。

码多项式的两个性质：

1. 任意码多项式$T(x)$都是生成多项式 $g(x)$ 的倍式。

证明：（n, k）循环码作为线性分组码，其生成矩阵G是k行n列的，可由k个不同的码字构成：

$$G(x)=\begin{pmatrix} x^{k-1}g(x) \\ x^{k-2}g(x) \\ \cdots\cdots\cdots \\ xg(x) \\ g(x) \end{pmatrix}$$

- 任给一个信息码$K = (c_{n-1}c_{n-2}\cdots\cdots c_{n-k})$，利用生成矩阵和公式$C = K \cdot G$，不难求出它对应的码字

$$T(x) = K \cdot G = (c_{n-1}c_{n-2}\cdots\cdots c_{n-k}) \cdot \begin{pmatrix} x^{k-1}g(x) \\ x^{k-2}g(x) \\ \cdots\cdots\cdots \\ xg(x) \\ g(x) \end{pmatrix}$$

$$= （c_{n-1}x^{k-1} + c_{n-2}x^{k-2} + \cdots\cdots + c_{n-k}） \cdot g(x);$$

即：$T(x) = h(x) \cdot g(x)$;

表明任意码多项式T(x)都应能被g(x)整除。

2．$g(x)$是x^n-1的一个因式。

证明：因为$g(x)$乘以x^k的模x^n-1运算仍为一个码多项式：

$$\frac{x^k \cdot g(x)}{x^n-1} = Q(x) + \frac{T(x)}{x^n-1}$$

$x^k \cdot g(x)$为n次多项式，除以x^n-1的商式必为$Q(x)=1$，

于是：　$x^k \cdot g(x) = x^n-1 + T(x) = x^n-1 + h(x) \cdot g(x)$

移项即证得：　$x^n-1 = g(x) \cdot [x^k - h(x)]$；

3.4.4 循环码的编码

（1）确定生成多项式$g(x)$：

由性质2知，$g(x)$是x^n-1的一个因式。可根据设定的码长n和监督位r，将x^n-1因式分解，从中选择一个r次的因子作为$g(x)$。

- 例如： (7, 4) 码，$n = 7$，$k = 4$，$r = 3$；

  应分解：$x^7-1 = (x-1)(x^3+x+1)(x^3+x^2+1)$
- 生成多项式$g(x)$ 应是x^7-1 的一个$r = 3$次的因子，可取为： $g(x) = x^3+x+1$ 或 $g(x) = x^3+x^2+1$；
- 二者任选其一，一旦选定，就不再考虑另一个了。

（2）计算待编信息的监督元:

- 以(7, 4)码为例，设信息位为$k_3\,k_2\,k_1\,k_0$，对应的码多项式为：$k(x)=k_3x^3+k_2x^2+k_1x+k_0$;

  设监督位为$r_2\,r_1\,r_0$，监督位对应的多项式为:

  $$r(x)=r_2x^2+r_1x+r_0\ ;$$

  故码字$C=(k_3\,k_2\,k_1\,k_0\,r_2\,r_1\,r_0)$对应的码多项式为:

  $$C(x)=k_3x^6+k_2x^5+k_1x^4+k_0x^3+r_2x^2+r_1x+r_0\ ;$$

  $$=x^3(k_3x^3+k_2x^2+k_1x+k_0)+(r_2x^2+r_1x+r_0)$$

  $$=x^3k(x)+r(x)$$

- 推广到一般，码多项式为:

  $$C(x)=x^r\cdot k(x)+r(x);$$

- 因为任何码多项式一定能被$g(x)$整除：

$$C(x) \ mod \ g(x) = 0$$

即：$[x^r \cdot k(x) \ \text{mod} \ g(x)] + [r(x) \ \text{mod} \ g(x)] = 0$；

移项，并考虑到模2运算可把负号变正号，于是：

$r(x) \ \text{mod} \ g(x) = x^r \cdot k(x) \ \text{mod} \ g(x)$；

考虑到$r(x)$是r-1次多项式，$g(x)$是r次多项式，

$r(x) \ \text{mod} \ g(x) = r(x)$

- 所以计算监督多项式的公式是：

$$r(x) = x^r \cdot k(x) \ \text{mod} \ g(x)$$

（3）写出编码C:

由$C(x) = x^r \cdot k(x) + r(x)$直接写出编码C；

[例1]求（7, 4）循环码中信息位K = (0100) 对应的的码字：

解：$k(x) = x^2$，$r = 3$，$x^r \cdot k(x) = x^5$，

$g(x) = x^3+x+1$；

∵ $r(x) = x^5 \bmod x^3+x+1 = x^2+x+1$；

∴ $C(x) = x^5 + x^2 + x + 1$；

∴ $C = (0100111)$；

同法可得到16个信息（0000~1111）的码字。

小插件1：系统码和非系统码

- 有人会问，求出了生成多项式$g(x)$，等于得到了一个码字，通过循环移位不就得到其它码字了吗？何必按上述办法来计算。
- 实际上，通过循环移位最多可写出n个码字，得不到全部2^k个许用码字。

例如(7, 4)循环码共有16个许用码字。

取$g(x)= x^3+x+1$ ，等于知道了中信息$K=(0001)$所对应的那个码字：　$C_1=(0001\ \ 011)$。

- C_1经过循环移位只能得到7个码字：

序号	信息位	许用码字
C_1	**0001**	（0001 011）
C_2	0010	（0010 110）
C_5	0101	（0101 100）
C_{11}	1011	（1011 000）
C_6	0110	（0110 001）
C_{12}	1100	（1100 010）
C_8	1000	（1000 101）

- 要想得到所有码字，需要写出生成矩阵G

- (7, 4)码的生成矩阵应当由4个许用码字构成，比如取循环组的前4个，构成的生成矩阵为G_1：

$$G_1 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

- 利用$C = K \cdot G_1$虽然能求得全部许用码字。

  比如：(0100)$\cdot G_1$=(0101100);

  (1000)$\cdot G_1$=(1011000);

- 然而发现码字不具备信息位在前，监督位在后的形式。原因是G_1不具备$G=[I_k\ Q]$的形式，这样编码叫非系统码，非系统码在译码时是比较困难的。

- 要构造系统码，生成矩阵应具备$G=[I_k\ Q]$的形式。可以经过线性变换，将G_1最下行加到第二行上，将最下面两行加到第一行上，就得到$[I_k\ Q]$的形式的系统码生成矩阵G:

$$G_1 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

- 实际上，为了构造$G=[I_k\ Q]$形式的生成矩阵。

  需要如下的4个码字：

  1 0 0 0 □□□

  0 1 0 0 □□□

  0 0 1 0 □□□

  0 0 0 1 □□□

  才能排列出4×4的单位矩阵I_k。

- 通过对C_1=(0001 011)的循环移位可以得到（0010 110）和（1000 101），

  但是却无法得到(**0100 □□□**)

- 原因何在？ 原来**0100**所对应的码字（0100 111）位于另一循环组中

第一循环组			第二循环组		
序号	信息	许用码字	序号	信息	许用码字
C_1	**0001**	（0001 011）	C_4	**0100**	（0100 111）
C_2	0010	（0010 110）	C_9	1001	（1001 110）
C_5	0101	（0101 100）	C_3	0011	（0011 101）
C_{11}	1011	（1011 000）	C_7	0111	（0111 010）
C_6	0110	（0110 001）	C_{14}	1110	（1110 100）
C_{12}	1100	（1100 010）	C_{13}	1101	（1101 001）
C_8	1000	（1000 101）	C_{10}	1010	（1010 011）

- 还有全零码C_0=(0000 000)和全1码C_{15}= (1111 111)，二者分别各自形成一个循环体。
- (7, 4)循环码 4位信息共16个许用码字，分别位于四个不同的循环体中。
- 系统码的生成矩阵G由来自两个不同循环体中的码字构成：C_1、 C_2 、 C_8来自第一循环组， C_4来自另一组。

$$G=\begin{pmatrix}C_8\\C_4\\C_2\\C_1\end{pmatrix}=\begin{pmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix}$$

- 结论：用 $C(x) = x^r \cdot k(x) + r(x)$ 直接就能得到系统码。

3.4.5 循环码的译码（纠错）

1. 由接收码字R，写出其接收码多项式$R(x)$；

2. 求伴随子多项式$S(x) = R(x) \mod g(x)$；

3. 若 $S(x) = 0$（即接收码多项式能被$g(x)$整除），则表明接收码无误。

4. 若 $S(x) \neq 0$，表明接收码有误，此时定义错误格式多项式：

$$E(x) = e_6x^6+e_5x^5+e_4x^4+e_3x^3+e_2x^2+e_1x+e_0;$$

5. 由$S(x)$ 求对应的$E(x)$。

因$S(x) = [C(x)+E(x)] \mod g(x)$

$= E(x) \mod g(x)$；

为了找到$S(x)$ 对应的$E(x)$， 我们可以预先将纠错能力t 位 以内的各种错误格式$E(x)$除以$g(x)$ 的余式都计算出来，列成一张$S(x)$—$E(x)$ 对照表，由$S(x)$就能直接查出$E(x)$。

6. 纠错译码：

由$C(x) = R(x) + E(x)$就能写出译码C。

[例2] 已知（7，4）码是纠正一位错的汉明码，且$g(x) = x^3+x+1$；请为$R = (0110010)$纠错。

解：设接收码为$R = (r_6\, r_5\, r_4\, r_3\, r_2\, r_1\, r_0)$；由

$S(x) = E(x) \mod g(x)$；可列出$S(x)$—$E(x)$ 对照表：

误码位置	r_0	r_1	r_2	r_3	r_4	r_5	r_6
$E(x)$	1	x	x^2	x^3	x^4	x^5	x^6
$S(x)$	1	x	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1

若$R = (0110010)$，$R(x) = x^5+x^4+x$；

$S(x) = (x^5+x^4+x) \mod (x^3+x+1) = x+1$；

查表知：$E(x) = x^3$；

于是：　$C(x) = R(x) + E(x) = x^5+x^4+x^3+x$；

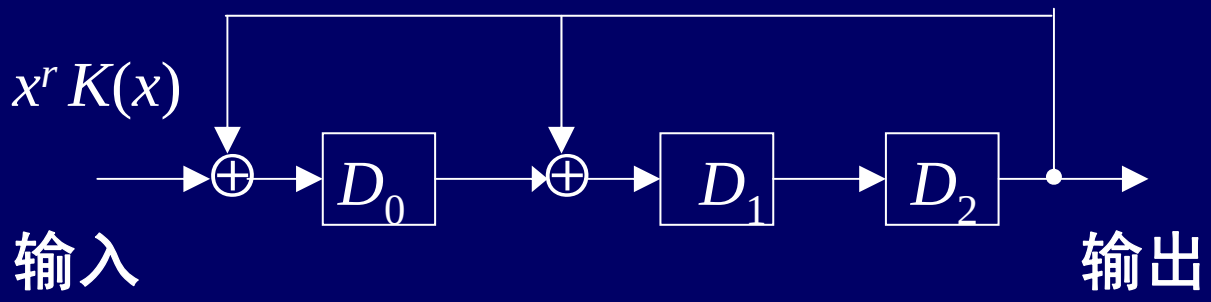
即：　$C = (0111010)$；

3.4.6 编、译码的电路实现

1. 除法求余电路：

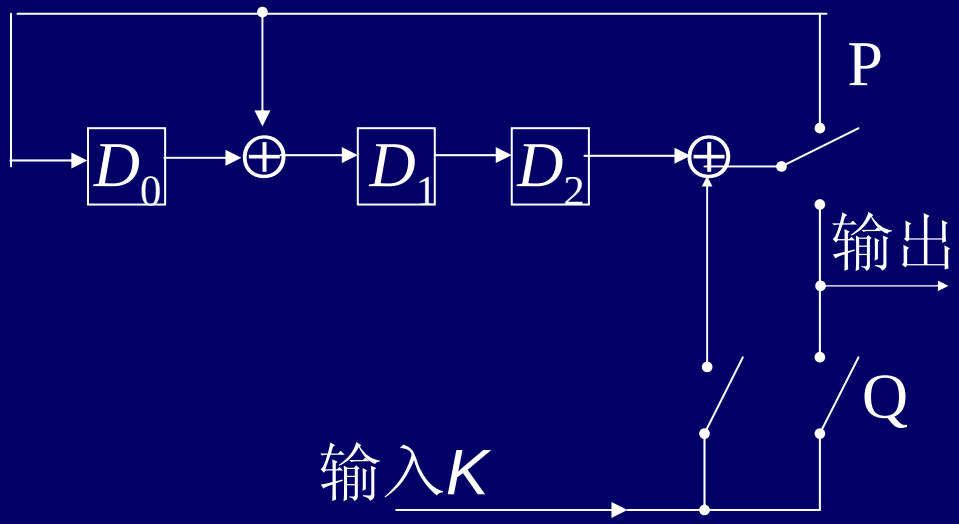
- 计算机中对公式的计算其实仍归结为数值计算，对所有的赋值都能正确得到结果，就等于对公式的计算。
- 笔算时是从被除数的高位开始，依次对除数求商、求积、求余，然后右移一位，继续前述过程，直至到被除数末尾 。
- 现在的道理相同，只不过把除数固定在电路上（就是反馈的位置），让被除数逐位右移通动寄存器，由于是二进制代码，商只有0和1两个可能值，积就是除数本身或是零，模二减等于模二加。用反馈电路很简单地实现了求商、求积、求余，输出的就是商，留在寄存器中的就是余数。

设被除数是$x^r \cdot k(x) = x^5$=0100000，除数是$g(x) = x^3+x+1$=1011，相除的过程见表所示。



$x^r K(x)$	输入	$D_0D_1D_2$	输出
x^6位	0	0 0 0	0
x^5位	1	1 0 0	0
x^4位	0	0 1 0	0
x^3位	0	0 0 1	0
x^2位	0	1 1 0	1
x^1位	0	0 1 1	0
x^0位	0	1 1 1	1

2. 编码电路：



$K(x)$	输入	$D_0D_1D_2$	输出
x^3位	0	0 0 0	0
x^2位	1	1 1 0	1
x^1位	0	0 1 1	0
x^0位	0	1 1 1	0
			1
			1
			1

把输入的$K(x)$从D_0端移到D_2后面，就得到了如图所示的实用编码电路。

首先开关P置向上，开关Q闭合，得到上面四行的数据；输出的就是信息K。然后P置向下，Q开启，继续将寄存器中的余数输出；就是接在后面的监督。

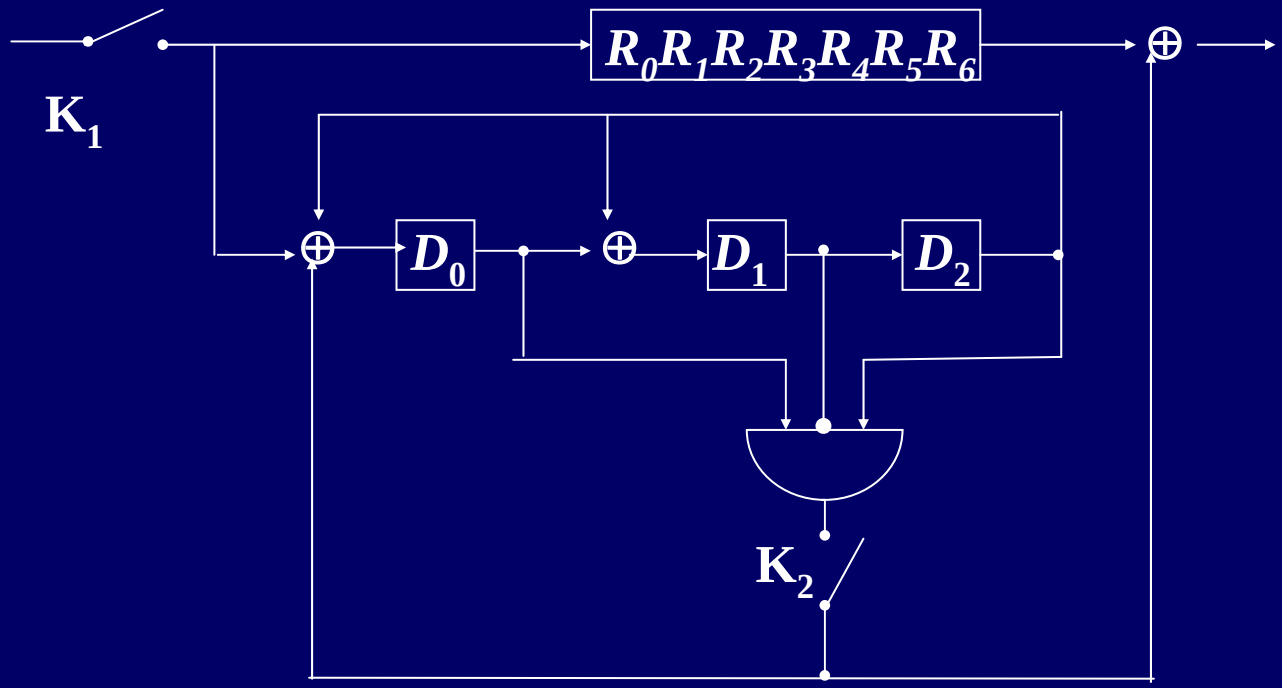
电路工作原理：

(1) 在D_2端加入$K(x)$，就等于在D_0端加$x^r \cdot k(x)$，这样就省去了前置的乘以x^3的乘法(移位)器。

(2) 将Q闭合，P置向上，就可以在进行除法运算的同时，将信息K送到输出，形成编码的前半段。

(3) 无须移位7次，只要移位4次，就可以完成求模运算的工作，接着把P置下，Q开启，就可以把$D_0D_1D_2$中的余数，顺序接在编码的后半段上，形成完整的编码。

3、译码电路：



- 首先断开K_2，接通K_1，利用除法求余电路，把接收码字$R(x)$ 除以$g(x)$ 的余式，即伴随子$S(x)$计算出来，存于$D_0D_1D_2$中；与此同时，$R(x)$ 也被缓存在R_0~R_6中。

- 然后，断开K_1，接通K_2。
- 与门设计是对输入（**101**）有响应。若e_6位错，则求余结果S = 101，恰使与门有输出，正好纠正R_6位。此后，移位继续进行，$D_0D_1D_2$值发生变化，与门关闭，不再影响$R_5 \ldots R_0$的输出。
- 若e_5位错，则求余得出的S =（111），与门无输出，但经过一个节拍后$D_0D_1D_2$变成(101)，与门变得有输出，正好轮到缓存器中R_5位的输出，将它纠正；再继续移位，与门又关闭了。
- 其它位有错时，同样引起上述类数据变化似的纠错过程。巧就巧在正好轮到R有错的那一位输出时，寄存器恰变为101，与门便输出纠错信号“1”，将该位错码纠正。

- 如此巧妙的璇玑在哪里呢?
- 因为伴随子$S(x) = R(x) \bmod g(x) = E(x) \bmod g(x)$，所以分析电路对$R(x)$的作用与分析$E(x)$是等价的。
- 当$R(x)$为正确码时， $E(x)$是全0，除法器求余结果为0，与门不会打开， $R(x)$从缓冲器中原样输出。
- 当$R(x)$有一位不正确时， $E(x)$的相应位是1，其它全0；除法器求余逻辑是按照x^3+x+1设计的, 初值为000, 当有1输入时才变为100, 此后由于输入全为0, 寄存器则按照100→010 →001→110→011→111→101的规律变化, 共7步变到101。
- $E(x)$为1的码位进入运算器的同时也进入缓冲器，经过7步才能缓冲才能输出，这时正好与门打开，将其纠正。

译码电路逐次移位的数据变化

错误格式 $e(x)$	伴随式 $S(x)$	寄存器 $D_0D_1D_2$	继续移位次数N	又移位后 $D_0D_1D_2$	纠正位 R
$e^6=1$	x^2+1	1 0 1	0	1 0 1	R_6
$e^5=1$	x^2+x+1	1 1 1	1	1 0 1	R_5
$e^4=1$	x^2+x	0 1 1	2	1 0 1	R_4
$e^3=1$	$x+1$	1 1 0	3	1 0 1	R_3
$e^2=1$	x^2	0 0 1	4	1 0 1	R_2
$e^1=1$	x	0 1 0	5	1 0 1	R_1
$e^0=1$	1	1 0 0	6	1 0 1	R_0

本节要点

1. 循环码的基本概念：

（1）码字的循环移位

（2）码多项式及其两个重要性质

（3）循环码的生成多项式

2. 循环码的编码：

根据信息K，直接由

$$r(x) = x^{r} \cdot k(x) \mod g(x)$$

求出监督位，添在信息位后，即得到编码。

3. 循环码的译码：

（1）由接收码字R，写出其接收码多项式$R(x)$；

（2）求伴随子多项式$S(x) = R(x) \mod g(x)$；

（3）若 $S(x) = 0$（即接收码多项式能被$g(x)$整除）， 则表明接收码无误。

（4）若 $S(x) \neq 0$，表明接收码有误，此时应将纠错能力t 位以内的各种错误格式$E(x)$除以$g(x)$ 的余式都计算出来，列成一张$S(x)$-$E(x)$ 对照表。

（5）由$S(x)$ 直接查表得到$E(x)$。

（6）由$C(x) = R(x) + E(x)$进行纠错。

- 思考：

  是否任意码长n和任意信息位k都能构成(n,k)循环码？（提示：考虑生成多项式）。

- 作业：

  P114页：14、17题

小插件2： x^n-1的因式分解

- 在循环码的一整套编码译码方案中，生成多项式是关键。它应当根据问题所需要的码长n和监督为位r，从x^n-1分解的因式中选取一个r次的因式获得。
- 如果肯动脑筋的话，你可以追问以下几个问题：

(1) 是不是任意码长都能找到r次的生成多项式？

(2) 码长n给定后有哪些可能幂次的生成多项式？

(3) 如何将x^n-1分解因式？

- 实际上第三个问题包含了前两个问题。

查表分解x^n-1的方法

（1）并非所有的x^n-1都具有r次的既约(不能再分解)的因式，只有满足$n=2^r-1$关系的x^n-1才具有r次的既约因式。因此 P194 页表4中只列出满足$n=2^m-1$的x^n-1的分解情况。

（2）不论n取何值，x^n-1总有一个$m_0(x)=x+1$的因式。

（3）x^n-1其它因式是$m_i(x)$，$i=1,3,5,7\ldots\ldots$

（4）$m_i(x)$的表达由8进制数给出，将它换成二进制自然码就是$m_i(x)$各位的系数。如$m=5$阶时，$n=31$，可分解$x^{31}-1$为：

第$i=1$类因式查表得到$(45)_8=(100101)_2$，表示$m_1(x)=x^5+x^2+1$；
第$i=3$类因式查表得到$(75)_8=(111101)_2$，$m_3(x)=x^5+x^4+x^3+x^2+1$；
第$i=5$类因式查表得到$(67)_8=(110111)_2$，$m_5(x)=x^5+x^4+x^2+x+1$；

（5）表中并未列出x^n-1所有的因式，与已列出因式对称的因式都被省略了。所谓对称指的是将二进制自然码高低位倒置的表达，如：

与$(100101)_2$对称的是$(101001)_2$，表示$m_{15}(x)=x^5+x^3+1$；
与$(111101)_2$对称的是$(101111)_2$，表示$m_7(x)=x^5+x^3+x^2+x+1$；
与$(110111)_2$对称的是$(111011)_2$，表示$m_{11}(x)=x^5+x^4+x^3+x+1$；

值得注意的是，有的二进制自然码本身就是对称的，如：$(11111)_2$与$(10001)_2$，高低位倒置后不变，不会出现新的因式。

（6）x^n-1分解为以上因式之积，诸因式中幂次最高为r。

（7）类序号i与n互素的那些因式$m_i(x)$被称为本原多项式；类序号i与n可约的那些因式$m_i(x)$被称为非本原多项式。如果n为素数，所有的因式都是本原多项式。

[例]查表分解$x^{63}-1$

因为$2^6-1=63$，所以应查$m=6$阶。

$i=1$：$(103)_8=(1000011)_2$，得知$m_1(x)=x^6+x+1$；

其对称因式由$(1100001)_2$，得知$m_{31}(x)=x^6+x^5+1$；

$i=3$：$(127)_8=(1010111)_2$，得知$m_3(x)=x^6+x^4+x^2+x+1$；

其对称因式由$(1110101)_2$，得知$m_{15}(x)=x^6+x^5+x^4+x^2+1$；

$i=5$：$(147)_8=(1100111)_2$，得知$m_5(x)=x^6+x^5+x^2+x+1$；

其对称因式由$(1110011)_2$，得知$m_{23}(x)=x^6+x^5+x^4+x+1$；

$i=7$：$(111)_8=(1001001)_2$，得知$m_7(x)=x^6+x^3+1$；

对称因式还是自己。

i=9：$(015)_8=(1101)_2$，得知$m_9(x)=x^3+x^2+1$；

其对称因式由$(1011)_2$，得知$m_{27}(x)=x^3+x+1$；

i=11：$(155)_8=(1101101)_2$，得知$m_{11}(x)=x^6+x^5+x^3+x^2+1$；

其对称因式由$(1011011)_2$，得知$m_{13}(x)=x^6+x^4+x^3+x+1$；

i=21：$(007)_8=(111)_2$，得知$m_{21}(x)=x^2+x+1$；

其对称因式仍是自己；

最终结果：

1）$x^{63}-1=m_0(x)m_1(x)m_3(x)m_5(x)m_7(x)m_9(x)m_{11}(x)$

$\cdot m_{13}(x)m_{15}(x)m_{21}(x)m_{23}(x)m_{27}(x)m_{31}(x)$；

2)本原多项式是$m_1(x)$, $m_5(x)$, $m_{11}(x)$, $m_{13}(x)$, $m_{23}(x)$和$m_{31}(x)$；

第三章 信道编码

3.5 循环码的扩展

（第13讲　2007.11.15.）

本节的主要内容

- 增余汉明码
- 截短循环码
- 循环冗余校验码
- 二元本原**BCH**码
- 二元非本原**BCH**码

外语关键词

增余汉明码：extended Hamming code

截短循环码：shortened cyclic code

循环冗余校验码：Cyclic Redundancy Check Code (CRC)

本原**BCH**码：primitive BCH code

非本原**BCH**码: **non-**primitive BCH code

上节回顾：循环码

- 基本概念：

循环码的特点，码多项式，循环移位的数学表达。

- 生成多项式：

（1）码多项式中幂次最低（幂次为r）常数项为1的非零多项式。

（2）通过对$g(x)$的循环移位可获得其它一些码多项式。

（3）任意码多项式$T(x)$都应能被$g(x)$整除。

（4）$g(x)$是x^n-1的一个因式。

- 循环码的编码：

（1）分解x^n-1，以获得生成多项式$g(x)$；

（2）求监督多项式：$r(x) = x^r \cdot k(x) \mod g(x)$；

（3）写出相应码字：$C(x) = x^r \cdot k(x) + r(x)$；

- 循环码的译码：

（1） 根据$S(x) = E(x) \mod g(x)$，算出纠错能力t位以内的各种错误格式$E(x)$的$S(x)$对照表；

（2）求接收码的伴随子向量$S^*(x) = R(x) \mod g(x)$；

（3）从对照表中查出$S^*(x)$对应的$E^*(x)$

（4） $C(x) = R(x) + E^*(x)$

3.5.1 增余汉明码

- 汉明码是能纠正一位错的完备码，具有较高的编码效率。如果给它的每一个许用码字增加一位奇偶校验位，使其成为 **(*n*+1, *k*)** 码，其编码效率降低不多，但监督能力却得到提高，不仅能纠正1位错，还能发现2位错。
- 如 **(7, 4)**码改为 **(8, 4)**码，第8列的码元满足偶校验关系：

$$c_7 = c_6+c_5+c_4+c_3+c_2+c_1+c_0;$$

一致监督矩阵由**H(7,3)**变为**H(8,4)**：

$$H(7,4)=\begin{pmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix}$$

$$H(8,4)=\begin{pmatrix}1&1&1&0&1&0&0&0\\0&1&1&1&0&1&0&0\\1&1&0&1&0&0&1&0\\1&1&1&1&1&1&1&1\end{pmatrix}$$

- 校验矩阵增加全1的一行，以便于检查两位错：
- 当只有一位错时，伴随子$S=RH^T=EH^T$与H^T的某一行相同（即S^T与H的某一列相同），这时伴随子最后一列的元素必然为1。
- 如果出现两位错时，E中有两列为1，S^T等于H中某两列相加，结果最后一行之和必然为0。伴随子最后一个元素为0，就表明有两个错。当然，它无法纠正，只能要求系统重发。

3.5.2 截短循环码

- 求生成多项式，需要分解x^n-1，从中找出一个r次的因子来， 但而这不是任何n 和r都能办到的。比如x^5-1 中就不含$r=2$和$r=3$的因子，难以构造(5,2)码。
- 可以从效率较高的（7, 4）汉明码出发，在(7, 4) 码的生成矩阵中去掉前两行和前两列，剩下的部分构成一个5×2的矩阵，可以用它来生成(5, 2)码。

$$G(7,4)=\begin{pmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix}$$

$$G(5,2)=\begin{pmatrix}1&0&1&1&0\\0&1&0&1&1\end{pmatrix}$$

- 相对于原来的循环码，码长被截短了，码字也减少了。这样生成的编码称之为截短循环码。
- 截短循环码的普遍表达是$(n-i, k-i)$，i是截短位数，监督位长度$r=(n-i)-(k-i)=n-k$不变。
- 截短循环码的引入，扩大了循环码的覆盖范围，使人们能根据所需要的n、k值灵活地设计各种$(n-i, k-i)$码。
- 截短循环码具有循环码的许多结构特点，监督位没有变，纠错能力不变，纠错方法不变等。但由于“截短”，只取用了循环组中的部分许用码，会使码组失去循环移位性。

3.5.3 循环冗余校验码(CRC)

- 循环冗余校验码是截短循环码的一个典型应用。
- 在数据通信和软盘、光盘存储器中，常常需要对较多信息（一个数据帧或一个记录轨道中的数据）进行差错监督。数据的长短往往不确定，但校验位的长度*r*却是固定的。
- 根据$n=2^r-1$设计出一个 (*n*, *k*) 循环码后，当信息长度小于*k*时，只要同时截短信息位和码字的长度，而保持监督位不变，便得到一个(*n-i*, *k -i*)截短循环码，这就是循环冗余校验码（CRC）。

- 常取 r =16 ， 这时 $g(x) = x^{16}+x^{12}+x^5+1$

  =1000100000100001(B)=11021(H)

- $g(x)$作为最轻的码字，它的重量为4，表明该码组中最小汉明距离d_0= 4，能纠正1位差错同时还能检查到第2位差错。

- 例如信息为$K(x)$=4D6F746F(H) ，则可以在信息后面添上CRC监督码：$R(x)=x^{16}K(x) \bmod g(x)$ =

  =4D6F746F0000(H) mod 11021(H)= B944(H)

- 有时也取32位的CRC校验码，生成多项式为：

  $g(x)=(x^{16}+x^{15}+x^2+1)\cdot(x^{16}+x^2+x+1)$

- 循环冗余校验码不仅实现起来比较简单，而且具有很强的检测能力。它能检测出：

（1）绝大部分连续长度不大于$n-k+1$的突发错误。

（2）相当一部分连续长度大于$n-k+1$的突发错误。

（3）所有许用码字汉明距离不大于d_0-1的错误。

（4）所有奇数个错位。

- 当错位较少时，它能自动纠正，当差错超过纠错能力时，根据校验位很容易进行检错，采用重发反馈 **(ARQ)** 方式保证数据的正确性。

3.5.4 二元本原BCH码

因式分解的数学理论解决了循环码的存在问题，并为它奠定了坚实的基础。

1. 能不能由码长决定循环码多项式？

首先，上述数学理论指出，当$n=2^r-1$时，x^n-1 必存在最高幂次为r 的本原多项式。可构成纠一位错的汉明码。

例如，r =3, 4, 5, 6 时：

$n=2^3-1=7$： x^7-1有3次本原因式 x^3+x+1，得到(7, 4)码

$n=2^4-1=15$：$x^{15}-1$有4次本原因式 x^4+x+1，得到(15, 11)码

$n=2^5-1=31$：$x^{31}-1$有5次本原因式 x^5+x^2+1，得到(31, 26)码

$n=2^6-1=63$：$x^{63}-1$有6次本原因式 x^6+x+1，得到(63, 57)码

2. 能不能构造能纠正多位错的循环码多项式？

- 纠错位数多，码字间距就得大，而线性分组码的最小汉明距离$d_0 \leqq r+1$，为此监督位r就得多一些，也就是说生成多项式的幂次r需要更大一些。
- x^n-1既然可分解为多个因式，我们不妨取几个因子的乘积作为生成多项式，就能增加它的幂次。而几个因子的乘积必定仍然还是x^n-1的因式。
- 比如 $x^7-1=(x+1)(x^3+x+1)(x^3+x^2+1)$；取生成多项式：

  $g(x)=(x^3+x+1)(x^3+x^2+1)=x^6+x^5+x^4+x^3+x^2+x+1$

  现在监督位增加到 $r=6$，就能构成（7,1）码(即七连重复码)，它可以纠正3位错（因为$2^6=1+7+21+35$）。

- 又如$x^{15}-1=(x+1)(x^4+x+1)(x^4+x^3+1)(x^2+1)(x^4+x^3+x^2+x+1)$

取生成多项式：

$$g(x)=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)$$

$$=x^{10}+x^8+x^5+x^4+x^2+x+1$$

现在监督位增加到 $r=10$，就能构成（15, 5）码，它可以纠正3位错。(因为$2^{10}>1+15+105+255$)

定义：取码长为$n=2^m-1$，取包括本原多项式在内的若干个因式之积为生成多项式的循环码叫做本原BCH码。BCH是能纠正多位错的循环码。

3. 能不能根据纠错位数*t*来设计生成多项式？

- 进一步的研究得知，给定*t* 就能决定在x^n-1中选取哪些因式。
- 设 *t* 是所设计的纠错位数， $m_i(x)$ 是x^n-1的因式，这里 $i=1,3,5,7,\cdots\cdots$ 则生成多项式

$$g(x)=LCM[m_1(x)\ m_3(x)\ m_5(x)\cdots\cdots m_{2t-1}(x)]$$

式中：*LCM* [·] 表示求最小公倍式。

• 这种码长为$n=2^m-1$，生成多项式由上述公式定义的循环码，叫做本原BCH码。这是更具实际意义的定义。

•P195页表5给出了$n\leqslant 255$的所有本原BCH码的码长、信息位、纠错位数，并以8进制方式给出了生成多项式。

[例1] 构造码长为15，分别能纠正1位、2位、3位和4位错的本原BCH码生成多项式。

解：查最小多项式系数表（参见附录二表4)知：

$x^{15}-1= m_0(x)m_1(x)\ m_3(x)\ m_5(x)\ m_7(x)$

$=(x+1)\ (x^4+x+1)\ (x^4+x^3+x^2+x+1)\ (x^2+x+1)\ (x^4+x^3+1)$

（1）$t=1$时：$2t-1=1$，$g(x)=LMC\ [m_1(x)] = x^4+x+1$；

得到（15, 11）码。

（2）$t=2$时：

$2t-1=3$，$g(x)=LMC\ [m_1(x)\ m_3(x)] =$

$(x^4+x+1)(x^4+x^3+x^2+x+1)=x^8+x^7+x^6+ x^4+x+1$；

得到（15, 7）码。

（3）t =3时： $t-1=5$，

$g(x)=LMC[m_1(x)m_3(x)m_5(x)]$

$=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)$

$=x^{10}+x^8+x^5+x^4+x^2+x+1$；

得到（15, 5）码。

（4）t =4时：$2t-1=7$，

$g(x)=LMC[m_1(x)\ m_3(x)\ m_5(x)\ m_7(x)]$

$=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)(x^4+x^3+1)$

$=x^{14}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8$

$+x^7+x^6+x^5+x^4+x^3+x^2+x+1$；

得到（15, 1）码，即15连重复码。

3.5.5 二元非本原BCH码

- 本原BCH码在码长n给定的条件下，通过增大监督位数r来提高纠错能力。这样必然导致编码效率k/n降低。
- 如果在纠错能力不变（即r不变）的条件下减小n值，是否有可能得到编码效率较高且能纠多位错的编码呢？非本原的BCH码就是这样的编码。
- 因为非本原多项式的共厄类i是n的因子，$m=n/i$是它的循环级，根据数学理论，x^m-1的因式分解中就能包含共厄类i 相应的那个非本原多项式。以此非本原多项式为生成多项式，可得到的码长为m循环码，r 没变， 码长m却比n小多了，使效率大大提高。

- 比如：$x^{15}-1= m_0(x)m_1(x)\ m_3(x)\ m_5(x)\ m_7(x)$

  $= (x+1)(x^4+x+1)\ (x^4+x^3+x^2+x+1)\ (x^2+x+1)\ (x^4+x^3+1)$

式中：$m_3(x)= x^4+x^3+x^2+x+1$是非本原多项式，因15/3=5，所以它是x^5-1的因式；以此为生成多项式，可得到（5,1）码。

$m_5(x)= x^2+x+1$也是非本原多项式，因15/5=3，所以它是x^3-1的因式；作为生成多项式，得到（3,1）码。

- 又如，$n=2^8-1=255$， $x^{255}-1$中有一个8次的非本原多项式 $m_{15}(x)=x^8+x^7+x^6+x^4+x^2+x+1$。
- 因为n/i =255/15=17，所以$x^{17}-1$中会包含这个8次的因子。由此可构造(17,9)码。
- 由纠错不等式$2^r=256$大于$C^0_{17}+C^1_{17}+C^2_{17}=154$不难知道，线性分组循环码(17, 9)可纠正两位错。它只付出8位冗余的代价就传输了9位信息，k/r比值 达到9/8，t/n比值达到 2/17。
- 利用非本原BCH码，人们构造了不少好码。(23,12) 和 (47,24) 码，效率超过50%，分别可以纠正$t=3$和$t=5$位错误码元。特别是(23,12)叫Golay码，是迄今为止人们所找到的少有的能纠正多个错误的完备码。

表3.11 某些非本原BCH码的生成多项式

r	i	n/i	k	t	d_0	以8进数表示$g(x)$
6	3	21	12	2	5	(127)(15)
8	15	17	9	2	5	(727)
9	7	73	46	4	9	(1210)(1027)(1401)
10	31	33	22	2	5	(3043)(3)
10	31	33	13	4	9	(3043)(3777)
11	89	23	12	3	7	(5343)
12	63	65	53	2	5	(10761)
12	63	65	40	4	9	(13535)(10761)(3)
23	178481	47	24	5	11	(43073357)

本节要点

1. 增余汉明码
2. 截短循环码
3. 循环冗余校验码
4. 二元本原BCH码
5. 二元非本原BCH码

- 思考：

  计算本原BCH码的生成多项式的公式为什么用最小公倍数？为什么式中的共厄类只有奇数？

- **P114页：15、16、18题**

附录:因式分解的数学理论

1. 复数域中的分解:

分解 x^n-1，相当于求解方程 $x^n-1=0$;

如 $x^2-1=0$的根是$x=1$和$x=-1$，则 $x^2-1=(x-1)(x+1)$

同理若 $x^n-1=0$的n个根是 $x=x_1, x_2, \cdots, x_n$,

则： $x^n-1=(x-x_1)(x-x_2)\cdots\cdots(x-x_n)$;

显然，由 $x^n=1$知，这n个根应当是1的n次方根。

写 $1=1\cdot e^{j2k\pi}$ ，根据复数开方公式:

若 $z=\rho e^{j\varphi}$ 则 $\sqrt[n]{z}=\sqrt[n]{\rho}\,e^{j(\varphi+2k\pi)/n}$ $(k=0,1,2,\cdots,n-1)$

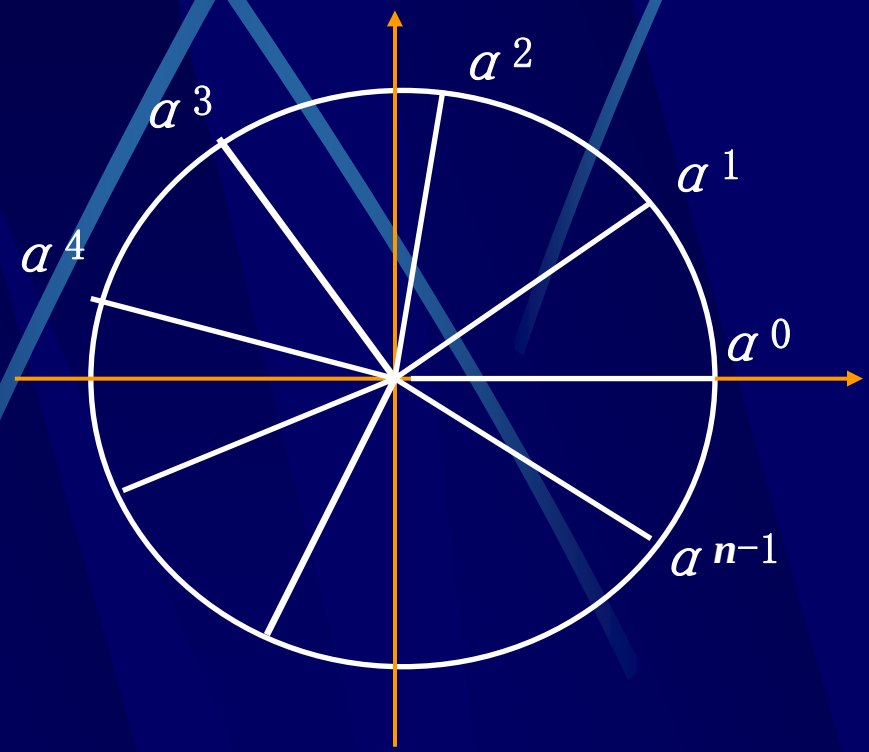
$$\therefore x_k = \sqrt[n]{1}\, e^{j(0+2k\pi)/n} = e^{j2k\pi/n} \quad (k = 0,1,2,\cdots,n-1)$$

这n个根模长均为1，辐角等份单位园的圆周。

令：$\alpha = e^{j2\pi/n}$；

则：$x_k = \alpha^k$

(k=0,1,……，n-1)



于是：

$$x^n - 1 = \prod_{k=0}^{n-1}(x - \alpha^k)$$

结论：复数域中x^n-1分解为n个1次因式的乘积。

2. 共厄类：

x^n-1的n个根 α^k，可以按k分成若干类，

这里 $k=i\cdot 2^m$ （m=0, 1, 2……）；

不妨以n=15，$x^{15}-1=0$的15个根为例。

i=1的类包含：α^1，α^2，α^4，α^8这4个根；

$\alpha^{16}=\alpha^1$，$\alpha^{32}=\alpha^2$，……又会重复出现这4个根。

i=3的类包含：α^3，α^6，α^{12}，$\alpha^{24}=\alpha^9$这4个根；

$\alpha^{48}=\alpha^3$，$\alpha^{96}=\alpha^6$，……又会重复出现这4个

同理得到其它共厄类：

共厄类	根	根的个数
i=1类	α^1，α^2，α^4，α^8	4个
i=3类	α^3，α^6，α^{12}，$\alpha^{24}=\alpha^9$	4个
i=5类	α^5，α^{10}	2个
i=7类	α^7，α^{14}，$\alpha^{28}=\alpha^{13}$，$\alpha^{56}=\alpha^{11}$	4个
i=0类	$\alpha^0=1$	1个

结论：$x^{15}-1=0$ 有 5个共厄类，各个类互不重复，且恰巧取尽了全部15个根。

3. 循环级：

因为 α^k是$x^n=1$的根，所以 $(\alpha^k)^n=1$，表明任何一个根自乘n次必然归1。

然而，例中第$i=3$类中的根 α^k，因为i值是n值的因子，每个根 α^k无须自乘$n=15$次，只要自乘$m=n/i=5$次，就能回到 $\alpha^0=1$；同理，第$i=5$类中的根 α^k，只要自乘$m=n/i=3$次，就能回到 $\alpha^0=1$。我们把通过自乘能回到 $\alpha^0=1$ 的最小自乘次数m叫做该类元素的循环级。

例中第$i=1$类和$i=7$类，k值与n值互素，因而每个根 α^k因式都必须自乘$n=15$次才能回到 $\alpha^0=1$，表明这些根的循环级$r=n=15$

4. 本原根与非本原根：

$x^n=1$的n个根中，凡是循环级m等于n的那些根被称为本原根。循环级小于n 的根，被称为非本原根。

- 同一个共厄类中的根，有同样的循环级，必有相同的本原或非本原属性。 $x^{15}=1$ 的15个根中，第$i=3$类与第$i=5$类中的根，是非本原根。而第$i=1$类和$i=7$类中的根是本原根；
- 若n为素数，则x^n-1的所有的根都是本原根；
- 若n为合数，则与n互素的那几个共厄类中的根是本原根，而与n有公因子的那几个共厄类中的根则是非本原根。

循环级与类中所包含的元素数目的关系：

- 循环级为m的类中所包含的元素数目r满足：

$$2^r \bmod m = 1$$

例如$x^{15}=1$的15个根中，

第$i=3$类，循环级$m=5$，由公式求出$r=4$；

第$i=5$类，循环级$m=3$，求出$r=2$；

第$i=1$类与第$i=7$类，循环级$m=15$，求出$r=4$；

5. x^n-1在GF(2)域中的分解：

因式分解与定义域是密切关联的。比如在复数域中$x^2+1=(x+j)(x-j)$，但在实数域中，x^2+1已经不能再分解了。

GF(2)域只有0和1两个数字，其它数是不存在的。

我们来看$x^{15}-1=0$的第$i=5$类中α^5与α^{10}这2个1次因式：

$$\because \quad \alpha^5 = e^{j(5\times 2\pi)/15} = e^{j2\pi/3};$$

$$\alpha^{10} = e^{j(10\times 2\pi)/15} = e^{j4\pi/3} = e^{-j2\pi/3};$$

$$\therefore (x-\alpha^5)(x-\alpha^{10}) = (x-e^{j2\pi/3})(x-e^{-j2\pi/3}) =$$

$$= x^2-(e^{j2\pi/3}+e^{-j2\pi/3})x+1 = x^2-2\cos(2\pi/3)x+1$$

$$= x^2+x+1$$

在 GF(2)域 x^2+x+1已经最简，不能再分解了。我们把它叫做=5类对应的最小多项式，记作：$m_5(x)= x^2+x+1$；

同理，属于每个共厄类的1次因式相乘，都可得到GF(2)域中一个相应的最小多项式，其幂次等于共厄类中根的个数：

$m_1(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^4)(x-\alpha^8)= x^4+x+1$；

$m_3(x)=(x-\alpha^3)(x-\alpha^6)(x-\alpha^9)(x-\alpha^{12})= x^4+x^3+x^2+x+1$；

$m_5(x)=(x-\alpha^5)(x-\alpha^{10})=x^2+x+1$；

$m_7(x)=(x-\alpha^7)(x-\alpha^{11})(x-\alpha^{13})(x-\alpha^{14})= x^4+x^3+1$；

$m_0(x)= (x-\alpha^0) = x+1$

于是： $x^{15}-1=m_0(x)m_1(x)m_3(x)m_5(x)m_7(x)$

6. 本原多项式与非本原多项式：

既然最小多项式是由同一个共厄类的根的1次因式相乘得到，就可以根据该类根的属性将最小多项式进行分类：

本原根的最小多项式被称为本原多项式。非本原根的最小多项式被称为非本原多项式。

$x^{15}-1$因式分解的最小多项式中，$m_1(x)$与$m_7(x)$是本原多项式；$m_3(x)$与$m_5(x)$是非本原多项式；

本原多项式与非本原多项式的幂次：

显然，每个最小多项式的幂次r，等于该类根的个数，而它由该类根的循环级决定：

$$2^r \bmod m=1$$

因此，本原多项式的幂次r 满足：

$$2^r \bmod n=1 \quad (\text{因为本原根的循环级} m=n)$$

非本原多项式的幂次r 满足：

$$2^r \bmod (n/i)=1 \quad (\text{因为第} i \text{类循环级} m=n/i)$$

x^n-1的根、共厄类及其最小多项式

类	根	循环级	是否本原	最小多项式
$i=1$	$\alpha^1, \alpha^2, \alpha^4, \alpha^8$	15	是	x^4+x+1
$i=3$	$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	5	非	$x^4+x^3+x^2+x+1$
$i=5$	α^5, α^{10}	3	非	x^2+x+1
$i=7$	$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	15	是	x^4+x^3+1
$i=0$	$\alpha^0=1$	无	恒元	$x+1$

小结：

- x^n-1有n个复数根 α^k，它们可分为若干个共厄类。
- 同一个共厄类的根有相同的循环级。循环级m等于n的根是本原根，循环级m等于n的某个因数(n/i)的根是非本原根。
- 同一个共厄类的逐根的1次因式相乘，得到GF(2)域的一个既约因式。每类都有自己的既约因式，x^n-1就可分解为这些既约因式之积。
- 各即约因式幂次r由该共厄类的循环级m决定：$2^r \bmod m=1$
- 本原根所在共厄类的即约因式叫做本原多项式，非本原根所在共厄类的即约因式叫做非本原多项式。反之也可以说，本原多项式的根是本原根，非本原多项式的根是非本原根。

7. 查表分解x^n-1 的方法：

- P194页表4以8进制形式列出了$r \leqq 10$的x^n-1(这里$n=2^r-1$)的分解结果。

如 $r=4$ 时，查4阶，共厄类 $i=1$ 指第1个因子，记作$m_1(x)$, 查得系数是23F, 即二进数010011，表示x^4+x+1；

共厄类 i=3 系数是37D, 即二进数011111，表示

$m_3(x)=x^4+x^3+x^2+x+1$; 同样方法可查到：

i=5类， 系数是07D, $m_5(x)=x^2+x+1$;

i=7类， 系数是31F, $m_7(x)=x^4+x^3+1$;

所以：$x^{15}-1= m_0(x)m_1(x)\ m_3(x)\ m_5(x)\ m_7(x)$

$= (x+1)(x^4+x+1)\ (x^4+x^3+x^2+x+1)\ (x^2+x+1)\ (x^4+x^3+1)$