# 信息论与编码
# Information Theory and Coding

# 第7章　信道编码技术

张建国

西安交通大学
电子与信息工程学院

2016年 6月



## 第7章　信道编码技术







## 编码目的及问题

### 编码目的

编码是指为了达到某种目的而对信号进行的一种变换。其逆变换称为译码或解码。根据编码的目的不同，编码理论包括以下分支：

1. 为了提高通信有效性的编码—信源编码；
2. 为了提高通信可靠性的编码—信道编码（纠错码）；
3. 为了提高通信保密性的编码—保密编码。

### 编码问题

1. 如何寻找性能优异的码？搜索？
2. 编译码方法是否易于实现？
3. 编码的性能分析应如何进行？

本章主要讨论信道编码技术，即纠错码的编译码方法。

信道编码历史上的几次突破（里程碑）。汉明码、卷积码的维特比译码、Turbo码。





## 编码问题举例

分组码、卷积码都是因为有良好的数学结构而被广泛应用。分组码的数学结构是近世代数，卷积码的结构是有限状态机。

**例：** 设输入序列长$L = 60$，编成长为$N = 100$的码字。这在通信应用中是很常见的。如果没有数学结构，只能使用查找表法。

源序列个数为$2^{60} = 1.15 \times 10^{18}$，存储源序列需要的存储空间为：

$$60 \times 2^{60}/8 = 8.65 \times 10^{18}\text{Byte}$$

类似地，存储码序列需要：

$$100 \times 2^{100}/8 = 1.58 \times 10^{31}\text{Byte}$$

这些均是天文数字，显然是不现实的。

而如果利用数学结构，线性分组码的核心就是一个编码矩阵（生成矩阵，元素取$0, 1$）。对该例来说，生成矩阵是$100 \times 60$维的，需要的存储空间为$100 \times 60/8 = 750$字节。

# 纠错码的分类

差错控制的基本形式：前向纠错（FEC）、反馈重发（ARQ）、混合纠错（HEC）和信息反馈（IRQ）等。

从纠错码的特点、性能或构造方法等角度出发，可以对其进行分类。

1. 按纠正错误的类型分
   纠随机差错码、纠突发差错码、纠混合差错码。
2. 按码的结构中对信息序列的处理方式分
   分组码、卷积码。
3. 按码的数学结构中校验元与信息元的关系分
   线性码、非线性码。
4. 按构造码的数学理论分
   代数码、几何码、算术码和组合码。
5. 按码是否具有循环特性分
   循环码与非循环码。





# 分组码

一个分组码是一组$q$进制$n$位长的符号序列，这些序列称为码字。对目前已发明的分组码，总有码字个数$M = q^k$，其中$k$为源序列的长度。这种编码通常被记为$(n, k)$码，表示$k$位长的源序列编为$n$位长的码序列。

例：$q = 2, n = 5, M = q^2 = 4$的$(5, 2)$分组码。其解码表如下：

| 源符 | $s_1$ | $s_2$ | $s_3$ | $s_4$ | |
|---|---|---|---|---|---|
| 码字输入 | 11000 | 00110 | 10011 | 01101 | |
| 输出 | 11001 | 00111 | 10010 | 01100 | 错1位 |
| | 11010 | 00100 | 10001 | 01111 | |
| | 11100 | 00010 | 10111 | 01001 | |
| | 10000 | 01110 | 11011 | 00101 | |
| | 01000 | 10110 | 00011 | 11101 | |
| | 11110 | 00000 | 01011 | 10101 | 错2位 |
| | 01010 | 10100 | 11111 | 00001 | |





# 卷积码（树形码）

卷积码给每个源序列（可能半无限长）分配一个码序列（可能半无限长），它可以用一个码树来描述。

卷积码的特点：

1. 与分组码不同，卷积码是有记忆的。
2. 和分组码相比，达到相同的误码率时，卷积码的$(n, k)$比较小

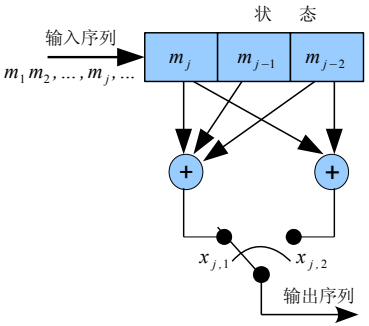
卷积码的描述方式：

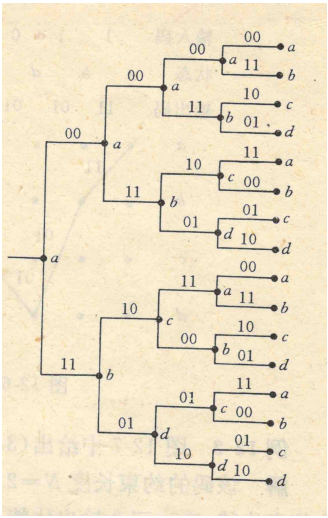
1. 移位寄存器方式（硬件电路）
2. 树状图
3. 状态转移图
4. 网格图（篱笆图）



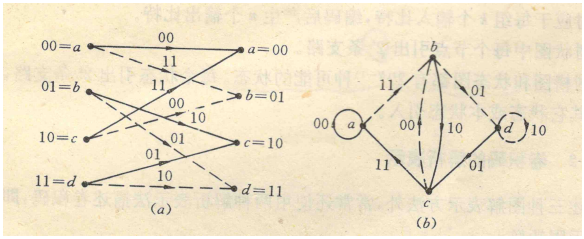


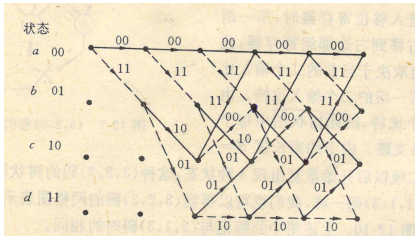
# 卷积码的描述方式—寄存器方式与树状图



# 卷积码的描述方式—状态流图与网格图








# 群的定义

定义 (群, Group)

设非空集合$G$内定义了一种代数运算$\odot$，若运算满足：

1. 封闭性—“群胚”
   $\forall a \in G, b \in G$，恒有$a \odot b \in G$。
2. 结合律—“半群”
   $\forall a, b, c \in G$，恒有$(a \odot b) \odot c = a \odot (b \odot c)$。
3. 单位元
   $\exists e \in G$，$\forall a \in G$，满足$a \odot e = e \odot a = a$，称$e$为单位元或恒元。
4. 逆（反）元
   $\forall a \in G$，$\exists a^{-1} \in G$，使$a \odot a^{-1} = a^{-1} \odot a = e$，称$a^{-1}$为$a$的逆元。

称$G$为一个群。

若群$G$的二元运算满足交换律，即若$\forall a, b \in G$，有$a \odot b = b \odot a$，则称群$G$为交换群或Abel群。





# 群的基本性质

性质一（恒元唯一）

群$G$中的恒元是唯一的。

**证明**：设$G$中有两个恒元$e$和$e'$，则有：$e' = e' \odot e = e$.
所以群$G$中恒元是唯一的。 □

性质二（逆元唯一）

群中任意元素的逆元是唯一的。

**证明**：设$a \in G$，在$G$中有两个逆元$a_1^{-1}$和$a_2^{-1}$，则

$$a_1^{-1} = e \odot a_1^{-1} = (a_2^{-1} \odot a) \odot a_1^{-1} = a_2^{-1} \odot (a \odot a_1^{-1}) = a_2^{-1} \odot e = a_2^{-1}.$$

所以群$G$中逆元是唯一的。 □





# 举例

1. 整数集合$\mathbb{Z}$及加法运算—Abel群
   整数对加法满足“闭”、“结”，单位元是0，逆元是$-a$，此外还满足交换律。
2. 自然数集合$\mathbb{N}$及加法运算
   自然数对加法满足“闭”、“结”，但没有单位元，也没有逆元，所以是半群但不是群。
3. 除零外的实数集合$\mathbb{R} \setminus \{0\}$及乘法运算
   该集合对乘法满足“闭”、“结”，单位元是1，逆元是$a^{-1}$，此外还满足交换律，所以是Abel群。
4. $N \times N$的非奇异矩阵集合，矩阵乘法
   非奇异矩阵的乘积仍是非奇异的，且满足结合律，单位元是单位阵，逆元是逆矩阵，所以是群。但矩阵乘法不满足交换律，所以不是Abel群。

# 举例—有限集合群

$G=\{0,1\}$，运算为模2加“$\oplus$”，也叫异或。

- “闭”：$0\oplus 0=0, 0\oplus 1=1, 1\oplus 0=1, 1\oplus 1=0$
- “结”
- “单”，单位元素是0。
- “反”，每个元素的逆元是自身。
- “交”

所以，$G$是一个Abel群。

**推论**

$G$是一个由长为$L$的$0,1$序列构成的集合，运算为模2加时构成一个群。

**群的重要性质**

$-(a+b)=(-b)+(-a)$.





# 子群和陪集

**证明：** $(a+b)+(-b)+(-a)=a+[b+(-b)]+(-a)$
$=a+0+(-a)=a+(-a)=0.$
同理可证：$(-b)+(-a)+(a+b)=0.$ □

**定义（子群）**

若群$G$的非空子集$H$对于$G$中定义的代数运算也构成群，则称$H$是$G$的子群。

**定义（陪集）**

设$G$的子群$H=\{h_1=e,h_2,\cdots,h_r\}$，$\forall a\in G$，但$a\notin H$，将它与$H$中的元素依次相加，得$a+H=\{a+h_i, i=1,2,\cdots,r\}$，称$a+H$为$H$的一个陪集（*Coset*），$a$称为该陪集的陪集首。

$H$的陪集可能有许多个，因此可以将$H$进行陪集展开。





# 陪集展开与陪集阵

| 子群 | $h_1=0$ | $h_2$ | $\cdots$ | $h_n$ | $h_i\in H$ |
|---|---|---|---|---|---|
| 陪集$\{g_1\}$ | $g_1+h_1=g_1$ | $g_1+h_2$ | $\cdots$ | $g_1+h_n$ | $g_1\notin H$ |
| 陪集$\{g_2\}$ | $g_2+h_1=g_2$ | $g_2+h_2$ | $\cdots$ | $g_2+h_n$ | $g_2$为前两行未出现元素 |
| | | $\vdots$ | | | |
| 陪集$\{g_m\}$ | $g_m+h_1=g_m$ | $g_m+h_2$ | $\cdots$ | $g_m+h_n$ | $g_m$为前$m$行未出现元素 |
| 陪集阵，每行为一个陪集$g_j$ | | | | | |





# 举例—子群、陪集与陪集展开

**例：** 试构造出全体二进制4-重矢量组成的模2加法群的任意一个子群，并求其陪集展开。

**解：** 二进制4-重矢量集合$V_4$的全部元素为：

| | | | |
|---|---|---|---|
| 0000 | 0100 | 1000 | 1100 |
| 0001 | 0101 | 1001 | 1101 |
| 0010 | 0110 | 1010 | 1110 |
| 0011 | 0111 | 1011 | 1111 |

- 要构造的子群$H$对模2加要是群，所以需要包括恒元0000；
- 根据模2加规则，各元素的逆元为其本身；
- 为满足封闭性，可在$V_4$中任取两元素，模2加后得到第4个元素，这样得到的4个元素必然构成群，故应是$V_4$的一个子群；
- 模2加运算满足结合律和交换律。

# 举例—子群、陪集与陪集展开(续)

任取两元素0110和1101，模2加得1011，由0000, 0110, 1101和1011这4个元素组成的集合$H$对模2加构成群。

按照陪集的定义，得$V_4$的陪集展开如下：

| | | | | | |
|---|---|---|---|---|---|
| $H$ | 0000 | 0110 | 1101 | 1011 | $1011 = 0110 \oplus 1101$ |
| $0001 + H$ | 0001 | 0111 | 1100 | 1010 | $g_1 = 0001 \in G, \notin H$ |
| $0010 + H$ | 0010 | 0100 | 1111 | 1001 | $g_2 = 0010 \in G, \notin H \cup \{g_1\}$ |
| $1110 + H$ | 1110 | 1000 | 0011 | 0101 | $g_3 = 1110 \in G,$ |
| | ↑ | | ↑ | | $\notin H \cup \{g_1\} \cup \{g_2\}$ |
| | 陪集首 | | 陪集展开 | | |

其中陪集首分别为0001, 0010和1110。





# 群的元素、子群和陪集

**定理 (群的元素、子群和陪集)**

群$G$中的每一个元素都位于子群$H$的一个且仅一个陪集中。

**证明：**先证出现，如果某个元素还未在陪集中出现，只要用它做陪集首形成新的陪集即可。

再证唯一，假设元素$a$在一个陪集$g_i$中出现两次，
设$a = g_i + h_k = g_i + h_j$，则
$(-g_i) + a = (-g_i) + g_i + h_k = (-g_i) + g_i + h_j \Rightarrow h_k = h_j$
所以，元素$a$在一个陪集中只能出现一次。
假设元素$a$在不同的陪集中出现两次，即$a = g_i + h_k = g_m + h_j$，
其中$h_k \neq h_j$，否则$g_i = g_m$，与假设矛盾。因此，有
$a + (-h_k) = g_i + h_k + (-h_k) = g_m + h_j + (-h_k)$
$\Rightarrow g_i = g_m + h_j + (-h_k) = g_m + h_n$
所以$g_i \in g_m$，这与陪集的产生方式矛盾。 □





# 群的元素和陪集

**定理 (群的两个元素位于同一陪集的充要条件)**

群$G$中任意两个元素$a_1, a_2$位于$G$的子群$H$的同一左陪集的充要条件是：$(-a_1) + a_2 \in H$。

**证明：**

**必要性**

设$a_1, a_2$同属于$\{g_i\}$，$a_1 = g_i + h_k, a_2 = g_i + h_j$，则
$(-a_1) + a_2 = (-h_k) + (-g_i) + g_i + h_j = (-h_k) + h_j \in H$

**充分性**

若$(-a_1) + a_2 = h_i \in H$，则
$a_1 + (-a_1) + a_2 = a_1 + h_i \Rightarrow a_2 = a_1 + h_i$
设$a_1 = g_i + h_k \in \{g_i\}$，那么
$a_2 = a_1 + h_i = g_i + h_k + h_i = g_i + h_j \in \{g_i\}$ □





# 环的定义

**定义 (环, Ring)**

若在非空集合$R$中定义了两种代数运算加和乘，且满足：

1. 集合$R$在加法运算下构成*Abel*群；
2. 乘法有封闭性，即$\forall a, b \in R$，有$ab \in R$。
3. 乘法结合律成立。即$\forall a, b, c \in R$，有
   $(ab)c = a(bc)$
4. 加法和乘法的左右分配律成立，即$\forall a, b, c \in R$，有
   $a(b + c) = ab + ac, (b + c)a = ba + ca$

则称$R$是一个环。

若环对乘法满足交换律，即对任意$a, b \in R$，恒有$ab = ba$，则称此环为交换环或Abel环。

# 环的基本性质

根据环的定义，不难看出环具有如下的基本性质。

**环的基本性质**

$\forall a, b \in R$，有：

$a \cdot 0 = 0 \cdot a = 0$

$a(-b) = (-a)b = -ab$

**定义 (零因子环)**

**设$a, b \in R$，且$a \neq 0, b \neq 0$，若$ab = 0 \in R$，则$a, b$为零因子，称$R$为有零因子环。**

例如：模6的剩余类环$Z_6$有6个元素：$\{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}, \bar{5}\}$，其中$\bar{2} \neq 0$，$\bar{3} \neq 0$，但$\bar{2} \cdot \bar{3} = \bar{6} = 0$，所以$\bar{2}, \bar{3}$ 是模6剩余类环的零因子，故$Z_6$为有零因子环。

把所有对模$n$同余的整数构成的一个集合称为模$n$的一个剩余类。





# 无零因子环和除环

在无零因子环中，乘法消去律成立，称为整环，如整数环；在有零因子环中，乘法消去律不成立，如$\bar{2} \cdot \bar{3} = \bar{0} \cdot \bar{3} = 0$，但$\bar{2} \neq \bar{0}$。

**定义 (除环)**

**乘法有单位元且每个非零元素有逆元、非可换的环，称为除环。**

如：$R = \{\text{复数对}(\alpha, \beta)\}$

加法：$(\alpha_1, \beta_1) + (\alpha_2, \beta_2) = (\alpha_1 + \alpha_2, \beta_1 + \beta_2)$

乘法：$(\alpha_1, \beta_1)(\alpha_2, \beta_2) = (\alpha_1\alpha_2 - \beta_1\bar{\beta}_2, \alpha_1\beta_2 + \beta_1\bar{\alpha}_2)$

易验证：对加法是Abel群，乘法封闭且结合律成立，分配律成立，所以$R$是环。

此外，对乘法有单位元$(1, 0)$，令$\gamma = \alpha\bar{\alpha} + \beta\bar{\beta}$，则非零元素的逆元为$(\frac{\bar{\alpha}}{\gamma}, \frac{-\beta}{\gamma})$。但乘法交换律不成立：$(i, 0)(0, 1) = (0, i)$，而$(0, 1)(i, 0) = (0, -i)$。所以$R$是除环。通常称之为四元数(素)除环。





# 环的例子

根据环的定义，可以考察一个集合是不是环。例如：

- 全体整数、全体偶数，构成环；
- 某一整数$m$的倍数全体，构成环；
- 模整数$m$的全体剩余类，构成环，称为剩余类环$Z_m$；
  把所有对模$n$同余的整数构成的一个集合称为模$n$的一个剩余类。
- 实系数多项式全体，构成环；
- $n$阶方阵全体，构成环，记为$R_n$。





# 域的定义

**定义 (域, Field)**

**若在非空集合$F$中定义了加和乘两种运算，且满足：**

1. **$F$关于加法构成*Abel* 群，其加法恒元记为*0*；**
2. **$F$中非零元素全体对乘法构成*Abel* 群，乘法恒元记为*1*；**
3. **加法和乘法间有如下分配律：**
   $a(b + c) = ab + ac \qquad (b + c)a = ba + ca$

**则称*F*是一个域。**

域是一个可换的、有单位元的、非零元素有逆元的环。

# 域的基本性质

域具有如下的基本性质：

**域的基本性质**

1. 域中一定无零因子
   若$a,b\in F$，$a\neq 0,b\neq 0$，则$ab\neq 0$ 。
2. $\forall a,b,c\in F$，有
   - $a\cdot 0=0\cdot a=0$;
   - $-(ab)=a(-b)=(-a)b$;
   - 若$ab=0$且$a\neq 0$，则$b=0$;
   - 若$a\cdot b=a\cdot c$且$a\neq 0$，则$b=c$。

**证明：**(1) 假设$F$中有零因子，则由定义$\exists a,b\in F$，$a\neq 0,b\neq 0$，使得$ab=0$；则有$0=a^{-1}0=a^{-1}(ab)=(a^{-1}a)b=b$，与$b\neq 0$的假设矛盾，所以$ab\neq 0$。

(2) 根据域的定义，不难证明所给结论。





# 域的例子

根据域的定义，可以考察一个集合是不是域。

1. 无限域
   有理数全体、实数全体、复数全体对加、乘，都构成域，且均为无限域；
2. 有限域$GF(q)$，又称伽罗华（Galois）域。
   域中有$q$个元素，称$q$为域的阶，亦称$GF(q)$为$q$元域。
   - 二元域$GF(2)$:
     集合$\{0,1\}$在模2加法和模2乘法下，是两个元素的域$GF(2)$。
   - $p$元域$GF(p)$:
     令$p$为素数，集合$\{0,1,\cdots,p-1\}$在模$p$加法和模$p$乘法下是阶为$p$的域，称为素域$GF(p)$。
   - 扩域$GF(p^m)$:
     将素域$GF(p)$扩展成有$p^m=q$个元素的域，即得$GF(p)$的$m$次扩域$GF(q)$。扩域$GF(p^m)$可用一个多项式$p(x)$来描述。





# 扩域中的运算举例

以$GF(2^2)$为例。
加法为模2加，如下：

| $\oplus$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 1 | 0 | 3 | 2 |
| 2 | 2 | 3 | 0 | 1 |
| 3 | 3 | 2 | 1 | 0 |

例如：$3+2$
$(11)_2+(10)_2=(01)_2$

乘法，多项式乘法：

| $\otimes$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 |
| 2 | 0 | 2 | 3 | 1 |
| 3 | 0 | 3 | 1 | 2 |

超出最高幂次对本原多项式取余。$GF(2^2)$的本原多项式为$x^2+x+1$。





# 有限域的性质—$GF(q)$的特征

**定义 (有限域的特征)**

**考察$GF(q)$中单位元素1的和序列，由封闭性，必然存在一个最小正整数$\lambda$，使得$\sum_{i=1}^{\lambda}1=0$，称$\lambda$为有限域$GF(q)$的特征。**

若域中单位元素为$e$，上述定义同样成立。

关于$GF(q)$特征的几个结论（证明略）：

1. 二元域$GF(2)$的特征是2，素数域$GF(p)$的特征是$p$。
2. 有限域的特征$\lambda$是素数。
3. $GF(\lambda)$是$GF(q)$的子域；若$\lambda\neq q$，则$q$是$\lambda$的幂。
   定理：有限域的阶必为其特征之幂。

# 有限域的性质—域元素的阶

**定义 (域元素的阶)**

对$GF(q)$中任意非零元素$a$，必然存在一个最小正整数$n$，使得$a^n = 1$，称$n$为元素$a$的阶。

如果$a$的阶是$q-1$，则称$a$是本原的，$a$为**本原元素**（primitive element），简称本原元。本原元素的各次幂生成$GF(q)$的所有非零元素。

关于域元素阶的几个结论（证明略）：

1. 设$a^i, a^j \in GF(q)$，
   若$i+j \le n$，则$a^i \cdot a^j = a^{i+j}$；
   若$i+j > n$，且$i+j = n+r, 0 < r \le n$，则$a^i \cdot a^j = a^r$。
   元素$a^n = 1$和$a^1, a^2, \cdots, a^{n-1}$在$GF(q)$乘法下形成的群称**循环群**。
2. 若$a \in GF(q)$，且$a \neq 0$，则$a^{q-1} = 1$
3. 若$a \in GF(q)$，且$a \neq 0$，令$n$是$a$的阶，则$q-1$能被$n$除尽。
4. 若$GF(p^m)$的特征为$p$，且$a, b \in GF(p^m)$，则$(a+b)^p = a^p + b^p$





# $GF(2)$域上的多项式

二进制及$2^m$进制在编码中应用最为广泛，所以重点讨论二元域$GF(2)$及其扩域$GF(2^m)$。

$(n+1)$比特的二进制数字序列，可用如下的多项式来描述：

$$f(x) = f_n x^n + f_{n-1} x^{n-1} + \cdots + f_1 x + f_0 : f_i \in GF(2), 0 \le i \le n$$

并称之为$GF(2)$上的多项式。其中，$x^i$代表对应系数所在的位置。例如二进制序列1001001100111001，其对应的多项式为：

$$f(x) = x^{15} + x^{12} + x^9 + x^8 + x^5 + x^4 + x^3 + 1$$

$GF(2)$上的多项式具有如下性质：

1. 可按普通方法进行加、减、乘、除运算；
2. 满足交换律、结合律、分配律；
3. 可做长除法，即
   $f(x) = q(x)g(x) + r(x), g(x) \neq 0$





# 既约多项式

**定义 (既约多项式)**

若$GF(2)$上的$m$次多项式不能被$GF(2)$上的任何次数小于$m$但大于零的多项式除尽，就称它是$GF(2)$上的既约多项式。

如：$x^2+x+1, x^3+x+1, x^4+x+1$分别为2、3、4次既约多项式。

- 对任意$m \ge 1$，存在有$m$次既约多项式；
- 能被分解因式的，都不是既约多项式；
- $GF(2)$上的任意$m$次既约多项式，除尽$x^n+1$，其中$n = 2^m - 1$。

例如：2次既约多项式除尽$x^3+1$，3次既约多项式除尽$x^7+1$，4次既约多项式除尽$x^{15}+1$，依次类推。





# 本原多项式

**定义 (本原多项式)**

若$m$次既约多项式$p(x)$除尽的$x^n+1$的最小正整数$n$满足$n = 2^m - 1$，称$p(x)$为本原多项式。

例如，既约多项式$x^3+x+1$的$m=3$，它能除尽$x^7+1$，但除不尽$x^4+1, x^5+1, x^6+1$，所以它是本原多项式。

- 本原多项式一定是既约多项式；
- 对于给定的$m$，可能有不止一个$m$次本原多项式；
  例如对于$m=5$，$x^5+x^3+1$是本原多项式，$x^5+x^2+1$也是。
- 可以根据本原多项式的定义编制计算机程序来计算各次本原多项式。

# 部分本原多项式表

：$m \le 24$的部分本原多项式

| 次数$m$ | 本原多项式$p(x)$ | 次数$m$ | 本原多项式$p(x)$ |
|---|---|---|---|
| 3 | $x^3+x+1$ | 14 | $x^{14}+x^{10}+x^6+x+1$ |
| 4 | $x^4+x+1$ | 15 | $x^{15}+x+1$ |
| 5 | $x^5+x^2+1$ | 16 | $x^{16}+x^{12}+x^3+x+1$ |
| 6 | $x^6+x+1$ | 17 | $x^{17}+x^3+1$ |
| 7 | $x^7+x^3+1$ | 18 | $x^{18}+x^7+1$ |
| 8 | $x^8+x^4+x^3+x^2+1$ | 19 | $x^{19}+x^5+x^2+x+1$ |
| 9 | $x^9+x^4+1$ | 20 | $x^{20}+x^3+1$ |
| 10 | $x^{10}+x^3+1$ | 21 | $x^{21}+x^2+1$ |
| 11 | $x^{11}+x^2+1$ | 22 | $x^{22}+x+1$ |
| 12 | $x^{12}+x^6+x^4+x+1$ | 23 | $x^{25}+x^5+1$ |
| 13 | $x^{13}+x^4+x^3+x+1$ | 24 | $x^{24}+x^7+x^2+x+1$ |





# $GF(2)$上多项式的$2^l$次幂

考察系数为$GF(2)$上的元素的多项式$f(x)$的二次方，有

$$\begin{aligned} f^2(x) &= (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0)^2 \\ &= (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0) \times \\ &\quad (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0) \\ &= f_n (x^n)^2 + f_{n-1}(x^{n-1})^2 + \cdots + f_1 x^2 + f_0 = f(x^2) \end{aligned}$$

进而考察多项式$f^2(x)$的二次方，有

$$\begin{aligned} f^4(x) &= (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0)^4 \\ &= (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0)^2 \times \\ &\quad (f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0)^2 \\ &= f_n (x^n)^4 + f_{n-1}(x^{n-1})^4 + \cdots + f_1 x^4 + f_0 = f(x^4) \end{aligned}$$

进一步推广，对任意$l \ge 0$，有$[f(x)]^{2^l} = f(x^{2^l})$。





# 由GF(2)构造出$GF(2^m)$

由$GF(2)$构造出含有$2^m, m>1$个元素的有限域$GF(2^m)$。

设$p(x)$是$GF(2)$上的$m$次本原多项式，$\alpha$为$p(x)$的根，即$p(\alpha)=0$，有：

$$\begin{aligned} x^{2^m-1}+1 &= q(x)p(x) \\ \alpha^{2^m-1}+1 &= q(\alpha)p(\alpha) \\ \alpha^{2^m-1}+1 &= 0 \\ \alpha^{2^m-1} &= 1 \end{aligned}$$

**定理 (有限域)**

**若$p(x)$是$GF(2)$上的$m$次本原多项式，$\alpha$为$p(x)$的根，则集合$F* = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\}$构成一个有限域。**





# 由GF(2)构造出$GF(2^m)$–续

**证明：**先证明乘法的封闭性。

设$0 \le i, j < 2^m - 1$，

若$i+j \le 2^m-1$，则；$\alpha^i \cdot \alpha^j = \alpha^{i+j}$是$F*$中的一个非零元素；

若$i+j > 2^m-1$，且$i+j = (2^m-1)+r, 0<r\le 2^m-1$，则$\alpha^i \cdot \alpha^j = \alpha^{i+j} = \alpha^{2^m-1}\cdot\alpha^r = \alpha^r$。$\alpha^r$也是$F*$中的一个非零元素。

所以$F*$中的非零元素在乘法下是封闭的。 □

$F*$中的非零元素在乘法下构成$2^m-1$阶群，且满足交换律和结合律，群中有恒元，任一非零元素有逆元（$\alpha^j$与$\alpha^{2^m-1-j}$互为逆元）。即$F*$中的非零元素在乘法下构成一可交换群。

**$F*$在加法下构成一个可交换群**

对于$0 \le i < 2^m-1$，用$p(x)$除多项式$x^i$，得$\boxed{x^i = q_i(x)p(x) + a_i(x)}$式中$q_i(x)$是商式，$a_i(x)$是余式。因为$p(x)$是$m$次多项式，所以$a_i(x)$的次数小于或等于$m-1$，将其记作：

$$a_i(x) = a_{i,m-1}x^{m-1} + a_{i,m-2}x^{m-2} + , \ldots, + a_{i,1}x + a_{i,0}$$

# 由GF(2)构造出$GF(2^m)$–续

设$\alpha$是$p(x)$的根，则

$\alpha^i = a_i(\alpha) = a_{i,m-1}\alpha^{m-1} + a_{i,m-2}\alpha^{m-2} + , \ldots, + a_{i,1}\alpha + a_{i,0}$

$F*$中的$2^m-1$个非零元素，可表示为$\alpha^0, \alpha^1, \ldots, \alpha^{2^m-2}$ 这就是域$F*$的幂描述。

$F*$中的非零元素也可表示成$GF(2)$上$2^m-1$个次数小于或等于$m-1$的不同的非零多项式，加上零元素（可以看作零多项式），这就是域$F*$的多项式描述。

在模2加法下，对$0 \le i, j < 2^m - 1$，有

$$0 + \alpha^i = \alpha^i + 0 = \alpha^i$$

$$\begin{aligned}\alpha^i + \alpha^j &= (a_{i,m-1}\alpha^{m-1} + a_{i,m-2}\alpha^{m-2} + , \ldots, + a_{i,1}\alpha + a_{i,0}) \\ &\quad + (a_{j,m-1}\alpha^{m-1} + a_{j,m-2}\alpha^{m-2} + , \ldots, + a_{j,1}\alpha + a_{j,0}) \\ &= (a_{i,m-1} + a_{j,m-1})\alpha^{m-1} + , \ldots, + (a_{i,1} + a_{j,1})\alpha \\ &\quad + (a_{i,0} + a_{j,0})\end{aligned}$$





# 由GF(2)构造出$GF(2^m)$–续

若$i = j$，则上式为零；否则不为零且必为$F*$中某个$\alpha^k$的多项式，即$F*$在定义的加法下有恒元0，逆元是其自身，且域$F*$是封闭的、可交换的，故域$F*$在定义的加法下构成一个可交换群。

$F* = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}\}$在加法运算下是可交换群，非零元素在乘法运算下也是可交换群，且利用多项式表示式，乘法对加法是可分配的。

所以$F*$是$2^m$个元素的伽逻华（Galois）域$GF(2^m)$，是$GF(2)$的扩域，称$GF(2)$是$GF(2^m)$的基域，根据前面的分析，$GF(2^m)$的特征是2。

**例：** 已知$p(x) = x^4 + x + 1$是$GF(2)$上的本原多项式，$\alpha$为其本原元，试由此构造其4次扩域$GF(2^4)$。





# 由GF(2)构造出$GF(2^m)$–续

**解：** $m = 4, 2^m - 2 = 14$，若将该域的元素用幂形式表示，有

$$GF(2^4) = \{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7, \alpha^8, \alpha^9, \alpha^{10}, \alpha^{11}, \alpha^{12}, \alpha^{13}, \alpha^{14}\}$$

下面考虑用多项式形式来表示域中的元素。

因为$p(\alpha) = \alpha^4 + \alpha + 1 = 0$，所以

$$\begin{aligned}&\alpha^4 = \alpha + 1; && \alpha^5 = \alpha \cdot \alpha^4 = \alpha^2 + \alpha; \\ &\alpha^6 = \alpha \cdot \alpha^5 = \alpha^3 + \alpha^2; && \alpha^7 = \alpha \cdot \alpha^6 = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1; \\ &\alpha^8 = \alpha^4 \cdot \alpha^4 = \alpha^2 + 1; && \alpha^9 = \alpha \cdot \alpha^8 = \alpha^3 + \alpha; \\ &\alpha^{10} = \alpha^5 \cdot \alpha^5 = \alpha^2 + \alpha + 1; && \alpha^{11} = \alpha \cdot \alpha^{10} = \alpha^3 + \alpha^2 + \alpha; \\ &\alpha^{12} = \alpha^3 + \alpha^2 + \alpha + 1; \\ &\alpha^{13} = \alpha \cdot \alpha^{12} = \alpha^3 + \alpha^2 + 1; && \alpha^{14} = \alpha \cdot \alpha^{13} = \alpha^3 + 1;\end{aligned}$$

说明：$\alpha^{12} = \alpha^9 \cdot \alpha^3 = \alpha^6 + \alpha^4 = \alpha^3 + \alpha^2 + \alpha + 1$





# 由GF(2)构造出$GF(2^m)$–续

这样就建立了域元素的幂描述与多项式描述之间的关系。同时，一个$(m-1)$次多项式对应着一个$m$-重二进制序列，它是该多项式的系数。由此可得$GF(2^m)$域元素的三种描述方法，分别是幂、多项式和$m$-重二进制序列的表示式。本例中三种描述的对应关系如下表所示。

| 幂 | 多项式 | 4-重序列 | 幂 | 多项式 | 4-重序列 |
|---|---|---|---|---|---|
| 0 | 0 | 0000 | $\alpha^7$ | $\alpha^3 + \alpha + 1$ | 1011 |
| 1 | 1 | 0001 | $\alpha^8$ | $\alpha^2 + 1$ | 0101 |
| $\alpha$ | $\alpha$ | 0010 | $\alpha^9$ | $\alpha^3 + \alpha$ | 1010 |
| $\alpha^2$ | $\alpha^2$ | 0100 | $\alpha^{10}$ | $\alpha^2 + \alpha + 1$ | 0111 |
| $\alpha^3$ | $\alpha^3$ | 1000 | $\alpha^{11}$ | $\alpha^3 + \alpha^2 + \alpha$ | 1110 |
| $\alpha^4$ | $\alpha + 1$ | 0011 | $\alpha^{12}$ | $\alpha^3 + \alpha^2 + \alpha + 1$ | 1111 |
| $\alpha^5$ | $\alpha^2 + \alpha$ | 0110 | $\alpha^{13}$ | $\alpha^3 + \alpha^2 + 1$ | 1101 |
| $\alpha^6$ | $\alpha^3 + \alpha^2$ | 1100 | $\alpha^{14}$ | $\alpha^3 + 1$ | 1001 |

## $GF(2)$域上多项式的根

$f(x)$是$GF(2)$域上的多项式，虽然它的系数取自二元域$GF(2)$，但它的根可能不在$GF(2)$中，理论上已证明它的根全部在其扩域$GF(2^m)$中。

例如：$x^4+x^3+1$在$GF(2)$上是既约的，但0和1都不是它的根，但在$GF(2^4)$中可以找到$\alpha^7,\alpha^{11},\alpha^{13}$和$\alpha^{14}$都是它的根，而且是它的全部根。

$$\begin{aligned}(\alpha^7)^4+(\alpha^7)^3+1&=\alpha^{28}+\alpha^{21}+1=\alpha^{13}+\alpha^6+1\\&=\alpha^3+\alpha^2+1+\alpha^3+\alpha^2+1=0\end{aligned}$$

由$[f(x)]^{2^l}=f(x^{2^l})$易知，$\alpha^{14}$、$\alpha^{28}=\alpha^{13}$以及$\alpha^{56}=\alpha^{11}$也是它的根。

多项式的根具有重要的意义，下面不加证明地列出$GF(2)$域上多项式根的基本性质及推论。





## $GF(2)$上多项式根的基本性质—续

**性质一（共轭元）**

$f(x)$是$GF(2)$上的多项式，$\beta$是$GF(2)$扩域的元素，若$\beta$是$f(x)$的根，则对任何$l\geq 0$，$\beta^{2^l}$也是$f(x)$的根。称$\beta^{2^l}$为$\beta$的共轭元。

由性质一可知，若$\beta$是$f(x)$的根，则$GF(2^m)$中$\beta$所有不相同的共轭元也是$f(x)$的根。

**性质二（$x^{2^m-1}+1$的全部根）**

$GF(2^m)$中$2^m-1$个非零元素形成$x^{2^m-1}+1$的全部根。

由性质二推广可知，$GF(2^m)$中的元素形成$x^{2^m}+x$的所有根，任何$\beta$都是$x^{2^m}+x$的根，且$\beta$有可能是$GF(2)$上次数小于$2^m$的多项式的根。





## $GF(2)$上多项式根的基本性质—续

$GF(2^m)$中的元素$\beta$有可能是多个多项式的根。其中，次数最低的多项式具有重要意义。

**定义 (域元素的最小多项式)**

设$\beta$是$GF(2^m)$中的一个元素，$\phi(x)$是$GF(2)$上使$\phi(\beta)=0$的最低次多项式，称$\phi(x)$为$\beta$的最小多项式。

**性质三（最小多项式是既约的）**

$GF(2^m)$中的元素$\beta$的最小多项式$\phi(x)$是既约的。

**性质四（最小多项式除尽）**

令$f(x)$是$GF(2)$上的多项式，$\phi(x)$是域$GF(2^m)$中元素$\beta$的最小多项式，若$\beta$是$f(x)$的根，则$f(x)$可被$\phi(x)$除尽。





## $GF(2)$上多项式根的基本性质—续

**性质五（最小多项式除尽$x^{2^m}+x$）**

$GF(2^m)$中元素$\beta$的最小多项式$\phi(x)$除尽$x^{2^m}+x$。

**性质六（既约多项式与最小多项式）**

令$f(x)$是$GF(2)$上的既约多项式，$\beta$是$GF(2^m)$中的元素，$\phi(x)$是$\beta$的最小多项式，若$f(\beta)=0$，则$\phi(x)=f(x)$。

**推论 (既约多项式与最小多项式)**

若一既约多项式有根$\beta$，它就是$\beta$的最小多项式$\phi(x)$。

# $GF(2)$上多项式根的基本性质—续

**性质七（既约多项式的构造）**

令$\beta$是$GF(2^m)$中的元素，$e$是使$\beta^{2^e}=\beta$的最小非负整数，则$f(x)=\prod_{i=0}^{e-1}(x+\beta^{2^i})$是$GF(2)$上的既约多项式。

**性质八（最小多项式的构造）**

令$\phi(x)$是$\beta$的最小多项式，$e$是使$\beta^{2^e}=\beta$的最小非负整数，则$\phi(x)=\prod_{i=0}^{e-1}(x+\beta^{2^i})$

**性质九（最小多项式次数）**

若令$\phi(x)$是$GF(2^m)$中元素$\beta$的最小多项式，$e$是$\phi(x)$的次数，则$e$是使$\beta^{2^e}=\beta$的最小整数，而且$e\le m$。





# $GF(2)$上多项式根的基本性质—续

**推论（最小多项式次数）**

$GF(2^m)$中任一元素的$\phi(x)$的次数除尽$m$。

例如$GF(2^4)$，$m=4$，它的所有元素的最小多项式的次数分别是1、2和4次，没有3次的最小多项式。

**性质十（共轭元与本原元素）**

若$\alpha$是$GF(2^m)$的本原元素，则它的所有共轭元$\alpha^2,\alpha^4,\ldots$也是$GF(2^m)$的本原元素。

**性质十一（共轭元的阶）**

若$\beta$是$GF(2^m)$中的一个$n$阶元素，则它的所有共轭元有同样阶数。





# 码字、信息元与校验元

编码器将输入的信息序列，每$k$个信息符号序列分成一段，构成信息组，记为$\mathbf{m}=(m_{k-1},m_{k-2},\ldots,m_0)$，其中$m_i,i=0,\ldots,k-1$称为**信息元**。

为了传输的可靠性，编码器将每个信息组按一定的规则增加$r$个多余的符号，形成长为$n=k+r$的序列$\mathbf{c}=(c_{n-1},c_{n-2},\ldots,c_1,c_0)$，称此序列为**码字（或码组、码矢）**。码字中的符号$c_i,i=0,\ldots,n-1$称为码元。所增加的码元称为**校验元**。通常用$\mathbf{C}$表示所有码字的集合。

当码字中所增加的校验元只由本组的信息元按一定规律产生，而与其他信息组无关时，所形成的码字集合称为分组码，记为$(n,k)$。

$n$长的$q$进制码符号序列共有$q^n$个。把选作$(n,k)$分组码码字的$q^k$个称为许用码组，把其余$q^n-q^k$个称为禁用码组。





# 线性分组码的定义

**定义（线性分组码）**

**码长为$n$，有$2^k$个码字的分组码，当且仅当其$2^k$个码字构成$GF(2)$域上所有$n$重矢量空间的一个$k$维子空间时，称该分组码为$(n,k)$线性分组码。**

$n$重二进制码元组成的集合$V_n$称为二进制的矢量空间。它包含两种运算，即模2加和乘。为了简单起见，也常用普通的“+”号代替“$\oplus$”。

矢量空间$V_n$的一个子集$S$如果满足：1）包含全0矢量；2）封闭性，即任意两个矢量的和也在$S$中。则称其为$V_n$的一个子空间。

假设$\mathbf{c}_i$和$\mathbf{c}_j$是$(n,k)$二进制分组码中的两个码字（或码矢量），当且仅当$\mathbf{c}_i+\mathbf{c}_j$也是一个码矢量时，这个码才是线性的；特别地，当$\mathbf{c}_i=\mathbf{c}_j$时，$\mathbf{c}_i+\mathbf{c}_j=0$。

**二元分组码是线性的充要条件是两个码字的模2和也是码字。**

# 线性分组码相关概念

分组码的线性只与选用的码字有关，而与消息序列怎样映射到码字无关。

例如：一个$(5,2)$分组码$\mathbf{C}=\{00000,01011,10101,11110\}$，假设消息序列与码字的映射关系为：

$$00\to 00000, 01\to 01011, 10\to 10101, 11\to 11110$$

容易验证它是线性的；如果将映射关系改变为：

$$00\to 11110, 01\to 10101, 10\to 01011, 11\to 00000$$

容易验证它仍是线性的。

但也容易验证前者满足如下的关系：如果消息序列$\mathbf{x}_1$映射为码字$\mathbf{c}_1$，$\mathbf{x}_2$映射为$\mathbf{c}_2$，则$\mathbf{x}_1+\mathbf{x}_2$映射为码字$\mathbf{c}_1+\mathbf{c}_2$。而后者尽管是线性的，却不满足上述映射关系。

称满足$\mathbf{x}_1+\mathbf{x}_2$映射为码字$\mathbf{c}_1+\mathbf{c}_2$的关系为线性分组码的特殊关系。一般来讲，**在线性分组码中，只要全0的消息序列映射为全0的码字，都能满足特殊关系**。因此在后面的讨论中，如果没有特别说明，均假定满足所述的特殊关系。





# 码字的重量

定义 (码字重量)

**码字的汉明重量是指码字中所含非零码元的个数。设码字$\mathbf{c}\in\mathbf{C}$，通常用$\mathrm{w}(\mathbf{c})$表示其汉明重量，也称汉明势，简称重量。**

设$\mathbf{c}=(c_{n-1},c_{n-2},\ldots,c_1,c_0)$为二元码，则有$\mathrm{w}(\mathbf{c})=\sum_{i=0}^{n-1}c_i$。

定义 (最小重量)

**码$\mathbf{C}$中所有非零码字重量的最小值称为码$\mathbf{C}$的最小重量。即**
$W_{\min}(\mathbf{C})=\min\{\mathrm{w}(\mathbf{c})|\mathbf{c}\in\mathbf{C},\mathbf{c}\neq 0\}$.

定义 (线性分组码的最小距离)

**在某一码$\mathbf{C}$中，任意两个码字汉明距离的最小值称为该码的最小距离，即**

$$d_{\min}=\min\{D_H(\mathbf{c_i},\mathbf{c_j})\}\quad \mathbf{c_i}\neq\mathbf{c_j}\quad \mathbf{c_i},\mathbf{c_j}\in\mathbf{C}$$





# 最小距离与最小重量

两个码字的汉明距离定义为对应码元不相同的个数。对二元编码，它等于两个码字模2和所得序列的重量。

如：$D_H(10110,01111)=\mathrm{w}(10110\oplus 01111)=\mathrm{w}(11001)=3$

定理 (线性分组码的最小距离与最小重量)

**线性分组码的最小距离等于非零码字的最小重量。**

**证明**：设线性分组码$\mathbf{C}$的最小距离为$d_{\min}$，最小重量为$W_{\min}$，则根据定义，有

$$\begin{aligned}d_{\min}&=\min_{\substack{\mathbf{v}_i,\mathbf{v}_j\in\mathbf{C}\\ i\neq j}}d(\mathbf{v}_i,\mathbf{v}_j)=\min_{\substack{\mathbf{v}_i,\mathbf{v}_j\in\mathbf{C}\\ i\neq j}}d(0,\mathbf{v}_i+\mathbf{v}_j)\\&=\min_{\substack{\mathbf{v}\in\mathbf{C}\\ \mathbf{v}\neq 0}}d(\mathbf{v})=W_{\min}\qquad\square\end{aligned}$$





# 错误图样

由于信道中存在噪声和干扰，接收序列中某些码元可能会发生差错。为便于描述所发生的差错，引入错误图样（差错图样）的概念。

对常见的二元数字通信系统。码元错误不外乎是“1”错变成“0”，或“0”错变成“1”，因此错误图样也可以用二元序列表示。

$\mathbf{e}=(e_{n-1},e_{n-2},\ldots,e_1,e_0)\qquad e_i\in\{0,1\},i=0,1,\ldots,n-1$

$e_i$取0表示第$i$位码元传输正确；而$e_i$取1表示该码元发生了差错。

设信道输出的接收序列为

$\mathbf{r}=(r_{n-1},r_{n-2},\ldots,r_1,r_0)\qquad r_i\in\{0,1\},i=0,1,\ldots,n-1$

则，码字、接收序列和错误图样三者间的关系为$\mathbf{r}=\mathbf{c}\oplus\mathbf{e}$。

根据模二和的运算性质，显然还有$\mathbf{c}=\mathbf{r}\oplus\mathbf{e}$或$\mathbf{e}=\mathbf{c}\oplus\mathbf{r}$。

错误图样的重量$W(\mathbf{e})$。对BSC信道，设$p$为错误传递概率，$\bar{p}=1-p$为正确传递概率，则$n$次无记忆扩展信道中，随机差错错误图样$\mathbf{e}$出现的概率为：$P(\mathbf{e})=\bar{p}^{n-W(\mathbf{e})}p^{W(\mathbf{e})}$。

# 纠错能力与最小重量

**定理 (线性分组码的纠错能力)**

**对任意$(n,k)$线性分组码C，其最小距离为$d_{\min}$，那么：**

1. **若要检测$a$个随机错误，则要求$d_{\min} \geq a+1$；**
2. **若要纠正$t$个随机错误，则要求$d_{\min} \geq 2t+1$；**
3. **若要纠正$t$个随机错误，且同时检测$a$（$a > t$）个随机错误，则要求$d_{\min} \geq a+t+1$。**

先说明什么是“检测$a$个并纠正$t(t < a)$个错误”（简称“纠检结合”）。

在某些情况下，要求对于出现较频繁但错码数很少的码组，按前向纠错方式工作，以节省反馈重发时间；同时又希望对一些错码数较多的码组，在超过该码的纠错能力后，能自动按检错重发方式工作，以降低系统的总误码率。这种工作方式就是“纠检结合”。

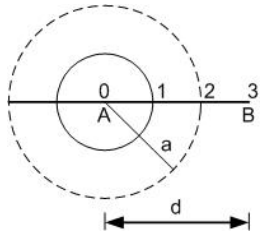




# 线性分组码的纠错能力（续）

在上述“纠检结合”系统中，差错控制设备按照接收码组与许用码组的距离自动改变工作方式。若接收码组与某一许用码组间的距离在纠错能力$t$范围内，则按纠错方式工作；若与任何许用码组间的距离都超过$t$，则按检错方式工作。

**证明：**可以由下图简单证明检错能力，

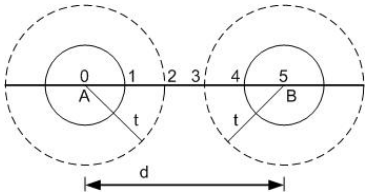




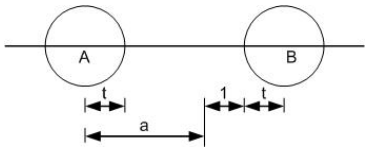



# 线性分组码的纠错能力（续）

纠错能力证明：




检错纠错能力证明：








# 线性分组码的纠错能力（续）

（1）设一码组A位于0点。

若码组A中发生一位错码，则可以认为A的位置将移动至以0点为圆心、以1为半径的圆（球壳）上某点，但其位置不会超出此圆；

若码组A中发生两位错码，则其位置不会超出以0点为圆心、以2为半径的圆。

因此，只要最小码距不小于3（如图中B点），在此半径为2的圆上及圆内就不会有其它码组。这就是说，码组A发生两位以下错码时，不可能变成其它任何许用码组，因而能检测错码的位数等于2。

同理，如果一种编码的最小距离为$d$，则将能检测$(d-1)$个错码；反之，若要求检测$a$个错码，则最小码距$d$至少应不小于$(a+1)$。

（2）图中画出码组A和B的距离为5。

码组A或B若发生不多于两位错码，则其位置均不会超出以原位置为圆心，以2为半径的圆。

# 线性分组码的纠错能力（续）

若接收码组落在以A为圆心的圆上，就判决收到的是码组A；若落在以B为圆心的圆上，就判决为码组B。这样，就能够纠正两位错码。

若这种编码中除码组A和B外，还有许多种不同码组，但任两码组之间的码距均不小于5，则以各码组的位置为中心、以2为半径画出的圆都不会互相重叠。

每种码组如果发生不超过两位错码都将能被纠正。

因此，当最小码距$d=5$时，能够纠正两个错码，且最多能纠正两个。若错码达到3个，就将落于另一圆上，从而发生错判。

为纠正$t$个错码，最小码距$d$应不小于$(2t+1)$。

（3）设码的检错能力为$a$，则当码组A中存在$a$个错码时，该码组与任一许用码组（例如图中码组B）的距离应至少为$t+1$，否则将进入许用码组B的纠错能力范围内，而被错纠为B。这样就要求最小码距$d$至少为$a+t+1$。





# 生成矩阵

构成线性分组码的一种方法。（为方便起见，下述向量均为行向量）

在$(n,k)$线性分组码中，假设消息序列分别为：

$$\begin{aligned}\mathbf{u}_1&=(1000\cdots00)\\ \mathbf{u}_2&=(0100\cdots00)\\ \mathbf{u}_3&=(0010\cdots00)\\ &\cdots\\ \mathbf{u}_{k-1}&=(0000\cdots10)\\ \mathbf{u}_k&=(0000\cdots01)\end{aligned}$$

这$k$个消息序列的长度都是$k$-bit，其对应的$k$个线性独立（无关）的码字分别为$\mathbf{g}_1,\mathbf{g}_2,\ldots,\mathbf{g}_k$，均是长度为$n$的二进制序列。

对于任意的消息序列$\mathbf{x}=(x_1,x_2,x_3,\ldots,x_{k-1},x_k)$，都可以表示为

$\mathbf{x}=\sum_{i=1}^{k}x_i\mathbf{u}_i$。





# 生成矩阵

对应的码字为：$\mathbf{c}=\sum_{i=1}^{k}x_i\mathbf{g}_i=(c_1,c_2,\ldots,c_n)$

定义：

$$\mathbf{G}=\begin{bmatrix}\mathbf{g}_1\\ \mathbf{g}_2\\ \vdots\\ \mathbf{g}_k\end{bmatrix}=\begin{bmatrix}g_{11}&\cdots&g_{1n}\\ \vdots&\ddots&\vdots\\ g_{k1}&\cdots&g_{kn}\end{bmatrix}$$

为该分组码的**生成矩阵**。则有

$$\mathbf{c}=\mathbf{x}\mathbf{G}\tag{1}$$

上式说明生成矩阵行向量的任意线性组合都是一个码字。生成矩阵$\mathbf{G}$是秩为$k$的$k\times n$矩阵，它完整地描述了编码的过程。

有了生成矩阵$\mathbf{G}$，编码器的结构就很容易确定，即上式事实上给出了编码的实现方法。





# 校验矩阵

为了在接收端进行正确译码，可以定义一个对应于生成矩阵$\mathbf{G}$的矩阵$\mathbf{H}$，称为一致校验矩阵或监督矩阵，满足：

$$\mathbf{G}\mathbf{H}^T=\mathbf{0}$$

即生成矩阵$\mathbf{G}$的行与一致校验矩阵$\mathbf{H}$的行相互正交。

由于$\mathbf{G}$是$k\times n$阶矩阵，故$\mathbf{H}$是$(n-k)\times n$阶矩阵，$\mathbf{0}$是一个$k\times(n-k)$阶的零矩阵。显然有：

$$\mathbf{c}\mathbf{H}^T=\mathbf{x}\mathbf{G}\mathbf{H}^T=\mathbf{0}\tag{2}$$

由于$\mathbf{c}$是$1\times n$阶的行矩阵，故式中$\mathbf{0}$为$1\times(n-k)$阶的行矩阵。

上式说明：任意码字必然与$\mathbf{H}$正交。它事实上给出了译码的实现思路。因为一致校验矩阵$\mathbf{H}$是已知的，如果接收到的码矢与它的转置的乘积为$\mathbf{0}$，则说明接收无误，否则说明存在错误。

# 生成矩阵与校验矩阵的意义

有了生成矩阵与校验矩阵，说明线性分组码：

1. 可以节省存储空间；
2. 容易判断接收到的码字是否有错。

例：某$(3,2)$码的生成矩阵为

$$\mathbf{G}=\begin{bmatrix}1&0&1\\0&1&0\end{bmatrix}$$

那么

$$\mathbf{c}_1=[00]\mathbf{G}=000 \qquad \mathbf{c}_2=[01]\mathbf{G}=010$$
$$\mathbf{c}_3=[10]\mathbf{G}=101 \qquad \mathbf{c}_4=[11]\mathbf{G}=111$$

**注意：一个线性分组码的生成矩阵不是唯一的。**





# 等价码

**定义 (等价码)**

**两个$q$元线性分组码被称为等价的，如果一个可以由另一个按下列方式得到**

1. **在一个码字内进行位置置换。**
2. **用非零常量去乘码字。**

对生成矩阵通过以下的一些变换所产生的线性分组码是等价的。

1. 行置换；
2. 对某行乘以非零常量；
3. 一行乘以某常量加到另一行上；
4. 列置换；
5. 任意列乘以一个非零的常数。





# 系统码与标准生成矩阵

**定义 (系统码)**

**若一个$(n,k)$线性分组码码字的前（或后）$k$个码元是信息元本身，就称该码为系统码。**

**定义 (标准生成矩阵)**

**若生成矩阵能把信息元保留在各码字的前$k$位上，就称其为标准生成矩阵。**

$$\mathbf{G}=\begin{bmatrix}&p_{11}&\cdots&p_{1,n-k}\\ \mathbf{I}_k&\vdots&\ddots&\vdots\\&p_{k1}&\cdots&p_{k,n-k}\end{bmatrix}=[\mathbf{I}_k\ \mathbf{P}]$$

其中，$\mathbf{I}_k$为$k\times k$阶单位阵，$\mathbf{P}$为$k\times(n-k)$阶矩阵。





# 标准生成矩阵

显然，很容易可得一致校验矩阵：$\mathbf{H}=[-\mathbf{P}^T\ \mathbf{I}_{n-k}]$。

$$\mathbf{G}\mathbf{H}^T=[\mathbf{I}_k\ \mathbf{P}]\begin{bmatrix}-\mathbf{P}\\ \mathbf{I}_{n-k}\end{bmatrix}=-\mathbf{I}_k\mathbf{P}+\mathbf{P}\mathbf{I}_{n-k}=-\mathbf{P}+\mathbf{P}=-\mathbf{0}$$

任何一个线性分组码的生成矩阵都可以通过等价码的方式化为标准生成矩阵。

例如，$(7,4)$汉明码的标准生成矩阵与对应的校验矩阵为：

$$\mathbf{G}=\begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix}\qquad \mathbf{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

# 一致校验矩阵的性质

若$\mathbf{C}$是一致校验矩阵为$\mathbf{H}$的$(n,k)$线性码，则对汉明重量为$l$的每个码矢，在$\mathbf{H}$中都有$l$列使得这$l$列的矢量和等于0；反之，若在$\mathbf{H}$中有$l$列，其矢量和是零矢量，则在$\mathbf{C}$中必有一个汉明重量为$l$的码矢。

若$\mathbf{H}$中没有$d-1$列或更少的列相加为 0 ，则码的最小重量至少为$d$；码$\mathbf{C}$的最小重量等于$\mathbf{H}$中列和为 0 的最小列数。

由于$\mathbf{H}$是$(n-k)\times n$维矩阵，因此其任意$n-k+1$个列向量必然线性相关。所以$(n,k)$线性分组码的最小距离$d\le n-k+1$，称为辛莱顿限。

$\mathbf{c}=\mathbf{xG}$已给出了一般线性分组码的编码实现方法。如果采用硬件实现，只要使用移位寄存器、乘法器和模2加法器等器件就可以了。特别地，对于系统码的编码，无论是用硬件还是软件都很容易实现。

分组码的消息部分越长，编码的实现也会越复杂，但随着集成电路和微处理器技术的成熟，设计中受编码复杂性制约的成份越来越小，注重的是其具体应用中的性能，这必须和译码特性统筹考虑。





# 译码的基本思路

$\mathbf{cH}^T=\mathbf{0}$已给出了一般线性分组码的检错方法。

设发端发送的码字为$\mathbf{c}=(c_1,c_2,\ldots,c_n)$，它是$2^k$个$n$重二进制许用码组中的一个。由于在传输过程中可能受到干扰或噪声的影响，设接收矢量为$\mathbf{r}=(r_1,r_2,\ldots,r_n)$，它是$2^n$个$n$重二进制码元组成的集合$V_n$中的一个。用$\mathbf{e}$表示错误图样，则有

$$\mathbf{r}=\mathbf{c}+\mathbf{e} \tag{3}$$

显然，在集合$V_n$中存在着$2^n-1$个不为0的潜在错误图样，纠错译码的任务就是确定错误图样。

如果$\mathbf{e}=\mathbf{0}$或者是码字，则$\mathbf{rH}^T=0$；如果$\mathbf{e}\neq\mathbf{0}$且不是码字，则$\mathbf{rH}^T\neq 0$。也就是说，$\mathbf{rH}^T$中含有接收矢量中的全部错误信息。





# 错误图样出现的概率

设某错误图样$\mathbf{e}$中错误的出错位数为$i$，则该错误图样出现的概率为$p=\varepsilon^i(1-\varepsilon)^{n-i}$。重量轻的错误图样比重量大的错误图样出现概率大。

即：$i_1<i_2$时，$p_1>p_2$。

$$p_1=\varepsilon^{i_1}(1-\varepsilon)^{n-i_1}$$
$$p_2=\varepsilon^{i_2}(1-\varepsilon)^{n-i_2}$$

那么

$$\frac{p_1}{p_2}=\varepsilon^{i_1-i_2}(1-\varepsilon)^{i_2-i_1}=\left(\frac{1-\varepsilon}{\varepsilon}\right)^{i_2-i_1}$$

当$\varepsilon<\frac{1}{2}$时，$\frac{1-\varepsilon}{\varepsilon}>1$，所以有：若$i_2-i_1>0$，则$\frac{p_1}{p_2}>1$。





# 伴随式的定义

前面提到$\mathbf{rH}^T$中含有接收矢量中的全部错误信息。为此给出如下定义：

定义 (伴随式)

称矢量$\mathbf{rH}^T$为接收码矢$\mathbf{r}$的伴随式或校正子，表示为

$$\mathbf{s}=\mathbf{rH}^T \tag{4}$$

如果$\mathbf{r}$是码字，则$\mathbf{s}$的值将为$\mathbf{0}$，否则将不为$\mathbf{0}$；如果$\mathbf{r}$包含着可纠正的错误，$\mathbf{s}$将有可能具有特殊的非零值并和特定的错误图样相对应。这就是伴随式译码的基本原理。

由于

$$\mathbf{s}=\mathbf{rH}^T=(\mathbf{c}+\mathbf{e})\mathbf{H}^T=\mathbf{cH}^T+\mathbf{eH}^T=\mathbf{eH}^T$$

根据一致校验矩阵的性质，错误图样将和伴随式相对应，这样，译码器就可根据要求实现FEC或ARQ，从而实现了译码。

# 伴随式的性质

**伴随式性质一**

**当且仅当$\mathbf{r}$是码字时$\mathbf{s}=0$，当且仅当$\mathbf{r}$不是码字时$\mathbf{s}\neq 0$。**

**伴随式性质二**

**存在着$2^k-1$个不可检测的错误图样。**

这是错误图样落到许用码组中的情形，即

$$\mathbf{v}+\mathbf{e}=\mathbf{w}$$

式中$\mathbf{v}$是发送的码字而$\mathbf{w}$是许用码组中的一个码字但不是发送的码字，这时虽然$\mathbf{s}=0$，但$\mathbf{r}=\mathbf{v}+\mathbf{e}=\mathbf{w}\neq\mathbf{v}$，造成无法检测的错误。

这一性质可等效为发端发的是一全0码矢而收端收的是非全0码矢，一共有$2^k-1$个。





# 伴随式的性质（续）

**伴随式性质三**

**令$A_i$是$(n,k)$线性码$\mathbf{C}$中重量为$i$的码矢数目，$p$是二元对称信道（BSC）的转移概率，若在BSC上$\mathbf{C}$只用来检错，用$P_u(E)$表示未检出错误的概率，则有 $P_u(E)=\sum_{i=1}^{n}A_i p^i(1-p)^{n-i}$**

根据性质二，只有当错误图样是一个非全零码的码字时，该错误图样才是不可检测的，即一种编码的不可检测错误概率，就是错误图样恰好为码字的概率。因此，求不可检测错误概率的问题就转化为求发端发全0码字而收端收到非全0码字的概率。

对于任意$i$，这种情况的概率为$p^i(1-p)^{n-i}$，而重量为$i$的码矢数目为$A_i$，考虑到全部的$i$，故有以上结论。

**伴随式性质四**

**$\mathbf{s}$是一个$n-k$维行矢量。**





# 不可检测错误概率的计算

**例：**假设某$(7,4)$码的重量分布为$A_0=1, A_1=A_2=0, A_3=A_4=7, A_5=A_6=0, A_7=1$，求其不可检测错误概率。

**解：**由伴随式性质三，有

$$P_u(E)=7p^3(1-p)^4+7p^4(1-p)^3+p^7$$

如果$p=10^{-2}$，则$P_u(E)\approx 7\times 10^{-6}$；
如果$p=10^{-4}$，则$P_u(E)\approx 7\times 10^{-12}$。

由数值计算，不可检测错误概率最主要决定于最小的$i$，也就是码字的最小重量或码字的距离。码字的距离越大，不可检测错误概率就越小。





# 标准阵的定义

**定义 (标准阵)**

**把$2^n$个可能的接收矢量划分成$2^k$个不相交的子集，使每个子集（列）只含有一个码矢，这个阵列称为标准阵（$2^{n-k}\times 2^k$）。**

标准阵的实质是将$V_n$集合中的所有$n$重二进制序列进行陪集展开，它的第一行以全0码字（$\mathbf{c}_1$或$\mathbf{e}_1$）开始，包括了所有的码字，而第一列即陪集首包括所有可纠正的错误图样，如图所示。

| | | | | | |
|---|---|---|---|---|---|
| $\mathbf{c}_1/\mathbf{e}_1$ | $\mathbf{c}_2$ | $\cdots$ | $\mathbf{c}_i$ | $\cdots$ | $\mathbf{c}_{2^k}$ |
| $\mathbf{e}_2$ | $\mathbf{c}_2+\mathbf{e}_2$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_2$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_2$ |
| | | $\cdots$ | | $\cdots$ | |
| $\mathbf{e}_j$ | $\mathbf{c}_2+\mathbf{e}_j$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_j$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_j$ |
| | | $\cdots$ | | $\cdots$ | |
| $\mathbf{e}_{2^{n-k}}$ | $\mathbf{c}_2+\mathbf{e}_{2^{n-k}}$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_{2^{n-k}}$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_{2^{n-k}}$ |

# 标准阵的性质

**标准阵性质一**

在标准阵的同一行中，没有两个$n$重序列是相同的，每个$n$重序列在且仅在一行中出现。

**标准阵性质二**

每个$(n,k)$线性分组码都能纠正$2^{n-k}$种错误图样，它们就是标准阵的陪集首。根据最大似然准则，重量较小的错误图样比重量较大的错误图样更可能出现，因此应当选择重量最小的那些矢量作为陪集首。

**标准阵性质三**

令$\alpha_i$表示重量为$i$的陪集首的数目，$\alpha_0,\alpha_1,\ldots,\alpha_m$ 称为陪集首的重量分布，当且仅当错误图样不是陪集首时才出现译码错误，故对于转移概率为$p$的二元对称信道而言，采用标准阵译码方法所产生的不可检测错误概率为 $P(E)=1-\sum_{i=0}^{m}\alpha_i p^i(1-p)^{n-i}$





# 标准阵的性质（续）

由性质二，在陪集首中小重量的错误图样占的比例较大，因此由上式计算其不可检测错误概率很小。

**标准阵性质四**

一个陪集的所有$2^k$个$n$重有同样的伴随式，不同陪集的伴随式互不相同。

**证明：** 设陪集首为$\mathbf{e}_l$的陪集中任一矢量$(\mathbf{e}_l+\mathbf{c}_i)$，其伴随式为

$$\mathbf{s}=(\mathbf{e}_l+\mathbf{c}_i)\mathbf{H}^T=\mathbf{e}_l\mathbf{H}^T+\mathbf{c}_i\mathbf{H}^T=\mathbf{e}_l\mathbf{H}^T$$

即同一陪集首所在的行中任一矢量的伴随式等于陪集首的伴随式。

再令$\mathbf{e}_j$和$\mathbf{e}_l$分别为第$j$个和第$l$个陪集的陪集首，其中$j\neq l$。假定这两个陪集的伴随式相等，则$\mathbf{e}_j\mathbf{H}^T=\mathbf{e}_l\mathbf{H}^T\Rightarrow(\mathbf{e}_l+\mathbf{e}_j)\mathbf{H}^T=0$。这表明$\mathbf{e}_l+\mathbf{e}_j$是码字，令$\mathbf{e}_l+\mathbf{e}_j=\mathbf{v}_i$，则$\mathbf{e}_l=\mathbf{v}_i+\mathbf{e}_j$，和标准阵的构成规则相矛盾，所以没有两个陪集有相同的伴随式。 □





# 标准阵译码

设接收系列为$\mathbf{r}$

方法一：在标准阵中搜索序列$\mathbf{r}$，将其译码为所在列的列首码字。

方法二：根据标准阵性质四，给标准阵增加一列伴随式。

| | | | | | | |
|---|---|---|---|---|---|---|
| $\mathbf{s}_1$ | $\mathbf{c}_1/\mathbf{e}_1$ | $\mathbf{c}_2$ | $\cdots$ | $\mathbf{c}_i$ | $\cdots$ | $\mathbf{c}_{2^k}$ |
| $\mathbf{s}_2$ | $\mathbf{e}_2$ | $\mathbf{c}_2+\mathbf{e}_2$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_2$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_2$ |
| | | $\cdots$ | | $\cdots$ | | |
| $\mathbf{s}_j$ | $\mathbf{e}_j$ | $\mathbf{c}_2+\mathbf{e}_j$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_j$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_j$ |
| | | $\cdots$ | | $\cdots$ | | |
| $\mathbf{s}_{2^{n-k}}$ | $\mathbf{e}_{2^{n-k}}$ | $\mathbf{c}_2+\mathbf{e}_{2^{n-k}}$ | $\cdots$ | $\mathbf{c}_i+\mathbf{e}_{2^{n-k}}$ | $\cdots$ | $\mathbf{c}_{2^k}+\mathbf{e}_{2^{n-k}}$ |

1. 根据接收序列计算伴随式$\mathbf{s}$，在伴随式列搜索$\mathbf{s}$；
2. 在$\mathbf{s}$所在行搜索接收序列$\mathbf{r}$。

**标准阵译码是最大似然译码。**





# 伴随式译码

性质四说明**陪集首和伴随式之间有着一一对应的关系**，由此可以构成一张译码表，对于接收到的码序列先计算其伴随式，通过查表找到对应的陪集首从而得到错误图样，这就是查表法译码的基本原理。

伴随式译码的一般步骤：

1. 计算接收矢量$\mathbf{r}$的伴随式：$\mathbf{s}=\mathbf{r}\mathbf{H}^T$；
2. 由伴随式$\mathbf{s}$，找到对应于它的陪集首$\mathbf{e}_l$；
3. 纠正错误：$\mathbf{v}=\mathbf{r}+\mathbf{e}_l$，$\mathbf{v}$即为译码输出的码字。

关键在于第(2)步。由标准阵可以看出，当$n-k$的数值不是很大时，用前面所述的查表法比较容易由伴随式得到所对应的陪集首，但对于大的$n-k$数值，查表法需要大量的存储器或复杂的电路，因此寻找更为有效的方法将变得很有意义。

**伴随式译码是最大似然译码。**

# 译码

1. 校验子、伴随式$\mathbf{s}=\mathbf{u}\mathbf{H}^T$：1行$(n-k)$列；
2. 当接收码字是许用码字时：校验子为零矢量；
3. 当接收码字是禁用码字时：校验子为非零矢量。

**定理 (校验子相等)**

**两个接收向量$\mathbf{u}_1$、$\mathbf{u}_2$位于同一陪集的充要条件是两者的校验子相等。**

**证明：** 由前面的定理知：$\mathbf{u}_1$、$\mathbf{u}_2$位于同一陪集的充要条件是$\mathbf{u}_1-\mathbf{u}_2$是另一个码字，记为$\mathbf{v}$，那么有

$$\mathbf{v}\mathbf{H}^T=0\Leftrightarrow(\mathbf{u}_1-\mathbf{u}_2)\mathbf{H}^T=0\Leftrightarrow\mathbf{u}_1\mathbf{H}^T=\mathbf{u}_2\mathbf{H}^T\Leftrightarrow\mathbf{s}_1=\mathbf{s}_2$$

解码：建立陪集首元素与校验子对照表。

标准阵译码、伴随式译码





# 译码举例

某$(4,2)$线性码

$$\mathbf{G}=\begin{bmatrix}1&1&1&0\\0&1&0&1\end{bmatrix}\Rightarrow\begin{bmatrix}1&0&1&1\\0&1&0&1\end{bmatrix}\qquad\mathbf{H}=\begin{bmatrix}1&0&1&0\\1&1&0&1\end{bmatrix}$$

标准阵列：

| | | | |
|---|---|---|---|
| 0000 | 1011 | 0101 | 1110 |
| 1000 | 0011 | 1101 | 0110 |
| 0100 | 1111 | 0001 | 1010 |
| 0010 | 1001 | 0111 | 1100 |

陪集首与伴随式：

| | | | |
|---|---|---|---|
| 0000 | 1000 | 0100 | 0010 |
| 00 | 11 | 01 | 10 |

如果收到码字是1100，用标准阵译码，结果为？

如果收到码字是0001，用伴随式译码，结果为？





# 编码效率的Plotkin上界

编码的目标：

1. 尽量小的误码率；
2. 尽量大的编码效率。

步骤：

1. 求出给定$d$后的$k/n$值；
2. 求出给定$d$后的$P_e$值；
3. 求相应的最佳编码。

**定理 (编码效率$k/n$的Plotkin上界)**

**如果$n\geq(qd-1)/(q-1)$，为获得最小码字重量$d$，校验位数$n-k$应不小于$\frac{qd-1}{q-1}-1-\log_q d$。**





# 编码效率的汉明上界

**定理 (编码效率$k/n$的汉明上界)**

**任何最小重量为$d=2t+1$的$(n,k)$分组码，它的校验位数须满足：$n-k\geq\log_q\left[1+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\cdots+\binom{n}{t}(q-1)^t\right]$。**

**证明：** $(n,k)$分组码码字的最小距离为$2t+1$，那么以每个码字为球心，Hamming距离$t$为半径做球，这些球不会相交。每个小球包含的序列数目为$\left[1+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\cdots+\binom{n}{t}(q-1)^t\right]$。

这样的球共有$q^k$个，空间中的总序列数为$q^n$。显然，

$$q^k\left[\sum_{i=0}^{t}\binom{n}{i}(q-1)^i\right]\leq q^n$$

即$n-k\geq\log_q\left[\sum_{i=0}^{t}\binom{n}{i}(q-1)^i\right]$。

使等号成立的线性分组码被称为完备的。Hamming码就是完备的。

## BSC的Hamming最佳编码及其误码率下界

定理 (误码率下界)

**任何二元$(n,k)$码的误码率下界为**

$$P_e \geq \left[\binom{n}{t+1} - \alpha_{t+1}\right] P^{t+1}Q^{n-t-1} + \sum_{i=t+2}^{n}\binom{n}{i}P^iQ^{n-i}$$

**其中所有$t$个或小于$t$个错可纠，部分$t+1$个错可纠，**
$\alpha_{t+1} = 2^{n-k} - 1 - \binom{n}{1} - \binom{n}{2} - \cdots - \binom{n}{t} \geq 0$

**证明：**根据汉明上界定理：

$$n - k \geq \log_2\left[\sum_{i=0}^{t}\binom{n}{i}(q-1)^i\right]$$





## BSC的汉明最佳编码及其误码率下界（续）

$$2^{n-k} \geq \sum_{i=0}^{t}\binom{n}{i} = 1 + \binom{n}{1} + \cdots + \binom{n}{t}$$

根据$\alpha_{t+1}$的定义$2^{n-k} = 1 + \binom{n}{1} + \cdots + \binom{n}{t} + \alpha_{t+1}$。
因此，最大正确解码的概率为

$$P_c \leq Q^n + \binom{n}{1}PQ^{n-1} + \cdots + \alpha_{t+1}P^{t+1}Q^{n-t-1}$$

$$P_e = 1 - P_c \geq \left(\binom{n}{t+1} - \alpha_{t+1}\right)P^{t+1}Q^{n-t-1} + \sum_{i=t+2}^{n}\binom{n}{i}P^iQ^{n-i}$$

注意：$1 = (P+Q)^n = Q^n + \binom{n}{1}PQ^{n-1} + \cdots + \binom{n}{n}P^n$。
当$2^{n-k} = 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}$或$\alpha_{t+1} = 0$时，称为完备码。
当$2^{n-k} > 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}$或$\alpha_{t+1} > 0$时，称非完备码。





## 完备码（$t=1$）

当$t=1$时，完备码应满足$1 + \binom{n}{1} = 2^{n-k}$或$n = 2^{n-k} - 1$

| 取$n-k=1,2,\ldots$ | | | | |
|---|---|---|---|---|
| $n-k$ | $n$ | $k$ | $k/n$ | |
| 1 | 1 | 0 | 0 | 无信息位，无用 |
| 2 | 3 | 1 | 1/3 | ↓效率增加 |
| 3 | 7 | 4 | 4/7 | |
| 4 | 15 | 11 | 11/15 | |
| 5 | 31 | 26 | 26/31 | |
| 6 | 63 | 57 | 57/63 | |





## 完备码（$t=2$）

当$t=2$时，完备码应满足$1 + \binom{n}{1} + \binom{n}{2} = 2^{n-k}$或$n(n+1) = 2\cdot 2^{n-k} - 2$

| 取$n-k=1,2,\ldots$ | | | | | |
|---|---|---|---|---|---|
| $n-k$ | $n$ | $k$ | $k/n$ | | |
| 1 | 1 | 0 | 0 | 无用 | ↓效率增加 |
| 2 | 2 | 0 | 0 | | |
| 3 | 3 | 0 | 0 | 非完备 | |
| 4 | 5 | 1 | 1/5 | 完备 | |
| 5 | 7 | 2 | 2/7 | | |
| 6 | 10 | 4 | 2/5 | | |
| 7 | 15 | 8 | 8/15 | | |
| 8 | 22 | 14 | 7/11 | 非完备 | |
| 9 | 31 | 22 | 22/31 | | |
| 10 | 44 | 34 | 17/22 | | |
| 11 | 63 | 52 | 52/63 | | |
| 12 | 90 | 78 | 39/45 | 完备 | |

# 汉明码定义

**定义 (汉明码)**

**最小距离为3的二元线性分组码$(2^m-1, 2^m-1-m)$称为汉明码，其中$m=n-k$为任何不小于2的整数。它是完备的，可纠正一位错。**

$2\cdot t+1=3$，可以纠正一位错，根据完备码的条件$n=2^{n-k}-1$，令$m=n-k$，即有$n=2^m-1$，$k=n-m=2^m-1-m$。其编码效率为

$$R=(2^m-1-m)/(2^m-1)$$

汉明码的编码效率随着$m$的增大而提高，当$m$很大时，$R$将接近于1，但由于它只能纠1个错，故实际应用中只选用适当的$m$值。

对应于$m=2\sim 8$的汉明码为：

$$(3,1);(7,4);(15,11);(31,26);(63,57);(127,120);(255,247)$$





# 汉明码的几何解释

汉明码只纠正1个错误，因此它的陪集首是重量为0和1的所有矢量。对于汉明码的标准阵，其每一列都正好是以其列首的码矢量（即标准阵第1行）为中心，以1为半径所作的一个球，而且这$2^{2^m-1-m}$个球又恰好能把标准阵所表示的矢量空间全部填满。

汉明码的陪集首是其纠错能力范围内的全部错误图样（其中0表示没有错误）。但要注意的是并非所有的线性分组码都有这一特性的。

由标准阵的性质二已知，每个$(n,k)$线性分组码都能纠正$2^{n-k}$种错误图样，它们是标准阵的陪集首。可是，能纠正$2^{n-k}$种错误图样并没有说明能纠正几个比特的错误。

例如，$(8,2)$线性分组码的陪集首的个数为$2^6=64$，表示可以纠正63种错误图样的错误，但具有1、2、3比特错误的错误图样的个数分别为$\binom{8}{1}=8;\binom{8}{2}=28;\binom{8}{3}=56$。





# 汉明码

用标准阵译码$(8,2)$线性分组码，除了能够纠正全部1比特和2比特错误外，还能纠正部分3比特的错误。由于它不能纠正全部的3比特错误，故其最大纠错能力只能说是2比特。

反映到陪集首的个数，它包含全部1、2比特错误的图样还有余，包含全部3比特错误的图样尚不足。

根据完备码的定义可知，汉明码是一种完备码。同时，汉明码也是到目前为止已知的二元线性分组码中编码效率最高的码。

汉明码是一种性能良好的码，它是在纠错编码的实践中较早发现的一类具有纠单个错误能力的纠错码，在通信和计算机工程中都有应用。如果对汉明码作进一步推广，就得出了能纠正多个错误的纠错码，其中最典型的是BCH码，而且汉明码是只纠1比特错误的BCH码，可将它们都归纳到循环码中。





# 汉明码设计

关键在于$\mathbf{H}$的设计。$\mathbf{s}=\mathbf{e}\mathbf{H}^T$，如果

$$\mathbf{e}=0,0,\ldots,\underset{i}{1},\ldots,0,\ldots,\underset{j}{1},0$$

那么$\mathbf{s}=\mathbf{h}_i+\mathbf{h}_j$，即$\mathbf{s}$是$\mathbf{H}$中与出错位对应的列矢量之和。

为了保证能纠正1位错，首先$\mathbf{H}$中不能有全零列；其次$\mathbf{H}$中各列要均不相同。而$\mathbf{H}$的维数为$(n-k)\times n$，列数为$n=2^m-1$。刚好就是由除全0向量外的全部长为$m$的二元向量组成。

**例：** $m=3, n=7, k=n-m=4$

$$\mathbf{H}=\begin{bmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{bmatrix}=\begin{bmatrix}0&1&1&1&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

$$=[-\mathbf{P}^T\ \mathbf{I}_3]$$

# 汉明码设计（续）

$$\mathbf{G}=[\mathbf{I}_4\ \mathbf{P}]=\begin{bmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{bmatrix}$$

$$\mathbf{E}\mathbf{H}^T=\mathbf{S}$$

其中，$\mathbf{E}$为错误向量矩阵，$(2^{n-k}-1)\times n$

$\mathbf{H}$为校验矩阵，$(n-k)\times n$

$\mathbf{S}$为伴随式矩阵，非零校验子矩阵，$(2^{n-k}-1)\times(n-k)$

$$\mathbf{H}^T=\mathbf{E}^{-1}\mathbf{S}$$

对$t=1$的最佳码，$\mathbf{H}^T=\mathbf{E}^{-1}\mathbf{S}=\mathbf{I}^{-1}\mathbf{S}=\mathbf{S}$。





# 汉明码设计举例

**例**：$t=1,n=15$，最佳码为$(15,11,3)$

$$\mathbf{H}=\begin{bmatrix}0&0&0&\cdots&1&0&0&0\\0&1&1&\cdots&0&1&0&0\\1&0&1&\cdots&0&0&1&0\\1&1&0&\cdots&0&0&0&1\end{bmatrix}=[-\mathbf{P}^T\ \mathbf{I}_4]$$

$$\mathbf{G}=[\mathbf{I}_{11}\ \mathbf{P}]=\begin{bmatrix}1&0&0&\cdots&0&0&1&1\\0&1&0&\cdots&0&1&0&1\\0&0&1&\cdots&0&1&1&0\\&&\vdots&\ddots&&&\vdots&\\0&0&0&\cdots&1&1&1&1\end{bmatrix}$$

Hamming码的$\mathbf{S}$很容易得到。$\mathbf{S}=\mathbf{E}\mathbf{H}^T$，当$\mathbf{E}$取遍所有的一位错序列时。$[100000\cdots0][010000\cdots0]\cdots[00000\cdots1]$，恰好构成一个单位阵，所以$\mathbf{S}=\mathbf{H}^T$。





# 循环码的特点与定义

循环码在性能上，具有明确的纠、检错能力，对于给定的码长$n$和信息位数$k$，已提出的各类循环码都有确定的纠、检错能力的理论计算值；

在实现上，编码和译码都可以通过简单的反馈移位寄存器来完成。

在结构上，它的循环性使得更容易用数学语言来描述。

本章将首先研究如何对循环码进行描述。然后讨论其编码和译码方法。接着以二元BCH码为例对其性能进行详细分析。最后对多元BCH码、RS码和其他循环码进行简要的讨论。

定义 (循环码)

**一个$(n,k)$线性分组码$\mathbf{C}$，若它一个码矢的每一循环移位都是$\mathbf{C}$的一个码字，则$\mathbf{C}$是一个循环码。**





# 循环码的举例

循环码首先是一种线性分组码，因此线性分组码的一切特性均适合于循环码；但它的特殊性是其循环性，码字集合或者说码组中任意一个码字的循环移位得到的序列仍是该码字集合中的码字，即它对循环操作满足封闭性。

**例1**：$\mathbf{C}_1=\{0000,0101,1010,1111\}$是循环码。

**例2**：$\mathbf{C}_2=\{0000,0110,1001,1111\}$虽然与$\mathbf{C}_1$是等价的，但它不是循环码。

注意：并不要求所有的码字都是由一个码字的循环移位构成的。

三种显然的循环码：

1. $(n,1)$码，即$n$重码；
2. $(n,n-1)$码，奇偶校验码；
3. $(n,n)$码。

# 循环码的举例

考虑一个$(7,3)$码，其生成矩阵为

$$\mathbf{G}=\begin{bmatrix}1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix}$$

非零码字为：

$$\begin{aligned}\mathbf{c}_1&=1011100 & \mathbf{c}_1&\to\mathbf{c}_2\\ \mathbf{c}_2&=0101110 & \mathbf{c}_2&\to\mathbf{c}_3\\ \mathbf{c}_3&=0010111 & \mathbf{c}_3&\to\mathbf{c}_1+\mathbf{c}_3\\ \mathbf{c}_1+\mathbf{c}_2&=1110010 & \mathbf{c}_1+\mathbf{c}_2&\to\mathbf{c}_2+\mathbf{c}_3\\ \mathbf{c}_1+\mathbf{c}_3&=1001011 & \mathbf{c}_1+\mathbf{c}_3&\to\mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3\\ \mathbf{c}_2+\mathbf{c}_3&=0111001 & \mathbf{c}_2+\mathbf{c}_3&\to\mathbf{c}_1\\ \mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3&=1100101 & \mathbf{c}_1+\mathbf{c}_2+\mathbf{c}_3&\to\mathbf{c}_1+\mathbf{c}_2\end{aligned}$$





# 循环码的举例

**例：**判断二进制码组

$$\begin{aligned}\mathbf{C}_1&=\{000,110,101,011\}\\ \mathbf{C}_2&=\{00000,01111,10100,11011\}\\ \mathbf{C}_3&=\{0000,1101,0111,1011,1110\}\end{aligned}$$

是不是循环码。

**解：**看码组符不符合线性和循环的条件。

对于码组$\mathbf{C}_1$，它既是线性分组码又是循环码。事实上，它是对$00,01,10,11$进行偶校验得到的码，是$(3,2)$循环码。

对于码组$\mathbf{C}_2$，它是线性分组码但不是循环码。事实上，它是对消息序列$00,01,10,11$进行编码得到的线性分组码$(5,2)$码。

对于码组$\mathbf{C}_3$，它尽管满足循环性但由于不是线性分组码，故也不是循环码。





# 循环码的研究工具—多项式

二元线性分组码可以看作扩域$GF(2^m)$的元素，它与$GF(2)$域上的多项式一一对应，利用多项式工具可以更好地分析循环码的性能。

可以建立码序列和码多项式的一一对应关系。

设码序列为$\mathbf{v}=\{v_{n-1},v_{n-2},\ldots,v_1,v_0\}$则它可用多项式表示为

$$v(x)=v_{n-1}x^{n-1}+v_{n-2}x^{n-2}+\cdots+v_1x+v_0$$

对码序列循环左移$i$次得到的新的码字，可以表示为

$$\begin{aligned}v^{(i)}(x)=v_{n-i-1}x^{n-1}+v_{n-i-2}x^{n-2}+\cdots+v_1x^{i+1}+v_0x^i\\ +v_{n-1}x^{i-1}+\cdots+v_{n-i+1}x+v_{n-i}\end{aligned}$$

称$v^{(i)}(x)$为码序列循环移位$i$次后的码多项式。





# 多项式的运算

循环码码字的循环移位与自身的和仍为一个码字。运算过程可以用多项式表示。

例如：$01101\leftrightarrow x^3+x^2+1$。

该码与其左移一次形成的码字仍为有效码字，

$$01101\oplus 11010=10111$$

如果用多项式运算，相当于乘以$x+1$，如下

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

我们还可以使用多项式的除法，$v(x)=q(x)p(x)+r(x)$。如

$$x^7+x^6+x^5+x^3=1\cdot(x^7+1)+(x^6+x^5+x^3+1)$$

# 多项式循环移位定理

**定理 (多项式的循环移位)**

**设循环码的码多项式为$v(x)=v_{n-1}x^{n-1}+v_{n-2}x^{n-1}+\cdots+v_1x+v_0$，它循环移位$i$次后的码多项式为$v^{(i)}(x)$，则$v^{(i)}(x)$是$x^n+1$除多项式$x^iv(x)$所得之余式。**

**证明：**将要证明的命题表示为

$$v^{(i)}(x)=x^iv(x) \mod (x^n+1)$$

即

$$x^iv(x)=q(x)(x^n+1)+v^{(i)}(x) \tag{5}$$

其中$q(x)$为商。

显然，由于$x^iv(x)=x^iv(x)+v^{(i)}(x)+v^{(i)}(x)$，我们只需证明$x^iv(x)+v^{(i)}(x)$是$x^n+1$的倍式即可。





# 多项式循环移位定理

由于

$$\begin{aligned}x^iv(x)=&v_{n-1}x^{n+i-1}+v_{n-2}x^{n+i-2}+\cdots+v_{n-i+1}x^{n+1}\\&+v_{n-i}x^n+v_{n-i-1}x^{n-1}+\cdots+v_1x^{i+1}+v_0x^i\end{aligned}$$

$$\begin{aligned}x^iv(x)+v^{(i)}(x)=&v_{n-1}x^{n+i-1}+v_{n-2}x^{n+i-2}+\cdots+v_{n-i+1}x^{n+1}\\&+v_{n-i}x^n\underline{+v_{n-i-1}x^{n-1}+\cdots+v_1x^{i+1}+v_0x^i}\\&\underline{+v_{n-i-1}x^{n-1}+\cdots+v_1x^{i+1}+v_0x^i}\\&+v_{n-1}x^{i-1}+\cdots+v_{n-i+1}x+v_{n-i}\\=&\left(v_{n-1}x^{i-1}+\cdots+v_{n-i+1}x+v_{n-i}\right)(x^n+1)\end{aligned}$$

故命题成立。 □





# 最低次码多项式

**定理 (最低次码多项式性质)**

**循环码$\mathbf{C}$中，次数最低的非零码多项式是惟一的。**

**证明：**

令

$$g(x)=x^r+g_{r-1}x^{r-1}+\cdots+g_1x+g_0$$

是$\mathbf{C}$中次数最低的非零码多项式。

若$g(x)$不是唯一的，则必存在有另一个次数为$r$的码多项式

$$g'(x)=x^r+g'_{r-1}x^{r-1}+\cdots+g'_1x+g'_0$$

因为$\mathbf{C}$是线性的，故$g(x)+g'(x)$是一个次数小于$r$的码多项式，必有$g(x)+g'(x)=0$，否则与假设相矛盾，故$g(x)$是惟一的。 □





# 最低次码多项式

**定理 (最低次码多项式必有常数项)**

**令$g(x)=x^r+g_{r-1}x^{r-1}+\cdots+g_1x+g_0$是$(n,k)$循环码$\mathbf{C}$中最低次数的非零码多项式，则常数项$g_0$必为1。**

**证明：**设$g_0=0$，则

$$\begin{aligned}g(x)&=x^r+g_{r-1}x^{r-1}+\cdots+g_1x+g_0\\&=x(x^{r-1}+g_{r-1}x^{r-2}+\cdots+g_1)\end{aligned}$$

将$g(x)$循环右移1位或循环左移$n-1$位，记为$g^{(1)}(x)$，有

$$g^{(1)}(x)=x^{r-1}+g_{r-1}x^{r-2}+\cdots+g_1$$

它是一个次数小于$r$的非零码多项式，与$g(x)$是最低次数的非零码多项式的假设相矛盾，故$g_0$必为1。因此

$$g(x)=x^r+g_{r-1}x^{r-1}+\cdots+g_1x+1$$ □

该定理实质上是论证了生成多项式必有常数项。

# 码多项式的充要条件

定理 (码多项式充要条件)

**令$g(x)=x^r+g_{r-1}x^{r-1}+\cdots+g_1x+1$是一个$(n,k)$循环码$\mathbf{C}$中次数最低的非零码多项式，一个次数等于或小于$n-1$次的二元多项式，当且仅当它是$g(x)$的倍式时，才是码多项式。**

**证明：** 充分性，令$v(x)$是一个次数小于或等于$n-1$次的二元多项式，如果$v(x)$是$g(x)$的倍式，则

$$\begin{aligned}v(x)&=(u_{n-r-1}x^{n-r-1}+\cdots+u_1x+u_0)g(x)\\&=u_{n-r-1}x^{n-r-1}g(x)+\cdots+u_1xg(x)+u_0g(x)\end{aligned}$$

由循环码的线性叠加和循环特性，知$v(x)$必是$\mathbf{C}$中的一个码多项式。

必要性，用$g(x)$去除$v(x)$，得

$$v(x)=u(x)g(x)+b(x)$$

式中$b(x)$的次数小于$g(x)$次数。





# 码多项式的充要条件

因此，

$$b(x)=u(x)g(x)+v(x)$$

上式右端两项均为码多项式，因此$b(x)$要么是非零码多项式，要么是0。由于$b(x)$的次数小于$g(x)$的次数，而定理假设$g(x)$是循环码$\mathbf{C}$中次数最低的非零码多项式，故$b(x)$只能是0。因此

$$v(x)=u(x)g(x)\qquad\square$$

该定理说明，一个次数等于或小于$n-1$次的二元多项式是码多项式的充要条件，就是它是$g(x)$倍式。这实际上给出了循环码的一种编码方法。因为：假如$u(x)$是消息多项式，则它和$g(x)$相乘就得到码多项式。

由此可见：码字集合的全部码字都可以由$g(x)$来给出，因此称$g(x)$为$(n,k)$循环码的生成多项式，其次数$r=n-k$。





# 循环码的生成多项式是$x^n+1$的因式

定理 (生成多项式)

**$(n,k)$循环码的生成多项式$g(x)$是$x^n+1$的因式。**

**证明：** 将生成多项式$g(x)$乘以$x^k$，由式(5)，得

$$x^kg(x)=q(x)(x^n+1)+g^{(k)}(x)$$

由于$x^kg(x)$次数为$n$，故上式中$q(x)=1$，而$g^{(k)}(x)$是$g(x)$循环左移$k$次所得，由前面分析知它是$g(x)$的倍式，设$g^{(k)}(x)=u(x)g(x)$，故有

$$x^n+1=[x^k+u(x)]g(x)=f(x)g(x)\qquad(6)$$

证完。 $\square$

$x^n+1$不但有$g(x)$这一因子还可能有其他因子。

该定理实质上给出了如何求得生成多项式的方法。





# $x^n+1$的$n-k$次因式生成$(n,k)$循环码

定理 ($x^n+1$的$n-k$次因式生成$(n,k)$循环码)

**若$g(x)$是一个$n-k$次多项式且是$x^n+1$的因式，则$g(x)$生成一个$(n,k)$循环码。**

**证明：** 考虑$k$个次数小于等于$n-1$的多项式$g(x),xg(x),\ldots,x^{k-1}g(x)$，令码多项式为

$$\begin{aligned}v(x)&=u_{k-1}x^{k-1}g(x)+\cdots+u_1xg(x)+u_0g(x)\\&=(u_{k-1}x^{k-1}+\cdots+u_1x+u_0)g(x)\end{aligned}$$

式中的第一个等式将码多项式表示为生成多项式及其$k$个左移多项式的线性组合，第二个等式将其表示为$g(x)$的倍式，一共可构成$2^k$个码字，因此它可构成$(n,k)$线性分组码。

再考虑其循环性，令$v(x)$是$(n,k)$线性分组码的$2^k$个码字中的一个。

$$v(x)=v_{n-1}x^{n-1}+v_{n-2}x^{n-1}+\cdots+v_1x+v_0$$

## $x^n+1$的$n-k$次因式生成$(n,k)$循环码

$$\begin{aligned} xv(x) &= v_{n-1}(x^n+1)+(v_{n-2}x^{n-1}+\cdots+v_0x+v_{n-1}) \\ &= v_{n-1}(x^n+1)+v^{(1)}(x) \end{aligned}$$

式中$v_{n-1}(x^n+1)$能被$g(x)$整除。因为$v(x)$是$g(x),xg(x),\ldots,x^{k-1}g(x)$的一个线性组合，它必能被$g(x)$整除。所以$v^{(1)}(x)$也是$g(x)$的倍式，也即$v^{(1)}(x)$是一个码多项式。

类似地，可以证明$x^iv(x)$也是码多项式，因此由$g(x)$生成的码是$(n,k)$循环码。 □

$(n,k)$循环码的生成多项式$g(x)$在代数结构上是$x^n+1$的一个$(n-k)$次因式。但由式(6)可见，$x^n+1$可能有不止一个$(n-k)$次因式。其含义是，对于一个$(n,k)$循环码可能会有多个生成多项式。

因此分析这些生成多项式所生成的码的性能，从而选择好的生成多项式，是研究循环码的主要目的之一，尤其当$n$较大时，需要用计算机搜索的方法来寻找好的生成多项式，通常称为搜索好码。





## 举例

**例：** 多项式$x^7+1$可以分解为

$$x^7+1=(x^3+x+1)(x^3+x^2+1)(x+1)$$

可以看出有两种$(7,4)$循环码的生成矩阵$g(x)=(x^3+x+1)$和$g(x)=(x^3+x^2+1)$。

由于$x^n+1$总可以分解为

$$x^n+1=(x+1)(x^{n-1}+\cdots+x+1)$$

因此对任意的$n$总存在以下两种循环码。

1. $n$重码：$(n,1)$，$g(x)=x^{n-1}+\cdots+x+1$:
2. 奇偶校验码：$(n,n-1)$，$g(x)=x+1$

如上例$x^7+1=(x+1)(x^6+x^5+x^4+x^3+x^2+x+1)$可产生$(7,1)$ 循环码（7重码）和$(7,6)$循环码（奇偶校验码）。





## 举例

**例：** 试构造$n=10$和$n=15$时可能的二进制循环码。

**解：** 对各种可能的$k$值求出其对应的生成多项式。

对于$n=10$的情况，求$k=1,2,\ldots,8,9$时对应的生成多项式，就是求$x^{10}+1$的各$(10-k)$次因式；

对于$n=15$的情况，求$k=1,2,\ldots,13,14$时对应的生成多项式，就是求$x^{15}+1$的各$(15-k)$次因式。

两种情况的计算结果如下表所示，表中给出了$g(x)$以及对应的最小重量，其中$g(x)$用其系数表示，例如110101 代表$x^5+x^4+x^2+1$。





## $n=10,15$时的循环码生成多项式

| $k$ | $g(x)(n=10)$ | $d_{\min}$ | $g(x)(n=15)$ | $d_{\min}$ |
|---|---|---|---|---|
| 1 | 1,111,111,111 | 10 | 111,111,111,111,111 | 15 |
| 2 | 101,010,101 | 5 | 11,011,011,011,011 | 10 |
| 3 | | | 1,001,001,001,001 | 5 |
| 4 | 1,100,011 | 4 | 111,101,011,001/110,001,100,011 | 8/6 |
| 5 | 100,001 | 2 | 10,100,110,111/10,000,100,001 | 7/3 |
| 6 | 11,111 | 2 | 1,011,001,101/1,100,111,001 | 6/6 |
| 7 | | | 111,010,001/110,111,011 | 5/3 |
| 8 | 101 | 2 | 11,010,001/11,100,111 | 4/4 |
| 9 | 11 | 2 | 1,011,101/1,111,001 | 4/3 |
| 10 | | | 110,101/100,001 | 4/2 |
| 11 | | | 10,011/11,111 | 3/2 |
| 12 | | | 1,001 | 2 |
| 13 | | | 111 | 2 |
| 14 | | | 11 | 2 |

# 循环码生成多项式

由表可见，$x^{10}+1$不含有$(10,3)$、$(10,7)$循环码，因为它没有7次和3次因式；而对于$k=5,6,8,9$，其距离都是2，故它们的纠、检错能力是一样的，但编码效率显然大不一样，说明同是循环码，性能却有所不同，这就提出了所谓搜索好码的问题。

$n=15$情况，$x^{15}+1$有$[1,14]$区间的各次因式，但有超过一半的$k$值有2个同次因式，对应的情况就可以构造出2个循环码。但注意到对应的$d_{\min}$值就会发现，有的情况却不相等，这说明注入同样的冗余度得到的效果却不一样，同样说明寻找性能好的循环码的重要性。

通过实例可以发现，随着$n$的增大，同次生成多项式的个数将会增加，因此搜索好码具有重要的意义。

给定$n$，对于不同的$k$，寻找$g(x)$，计算$d_{\min}$及重量分布，可以构造出符合需要的循环码。循环码中的许多好码都是这么发现的。

直到$n=99$的循环码，已有表格。





# 非系统码生成矩阵

循环码是一种线性分组码，因此它的编码和译码也可以用生成矩阵和一致校验矩阵来描述，同样也分为系统码和非系统码。

设$(n,k)$循环码：

生成多项式：$g(x)=g_{n-k}x^{n-k}+\cdots+g_1x+g_0$，$g_i\in[0,1], g_0=g_{n-k}=1$

信息多项式：$\mathbf{m}(x)=m_{k-1}x^{k-1}+\cdots+m_1x+m_0$，$m_i\in[0,1]$

码多项式：$\mathbf{c}(x)=c_{n-1}x^{n-1}+\cdots+c_1x+c_0$，$c_i\in[0,1]$

由码多项式充要条件，有

$$\begin{aligned}\mathbf{c}(x)&=\mathbf{m}(x)g(x)\\&=m_{k-1}x^{k-1}g(x)+\cdots+m_1xg(x)+m_0g(x)\\&=(m_{k-1}\ \cdots\ m_1\ m_0)\cdot\begin{bmatrix}x^{k-1}\cdot g(x)\\ \vdots\\ x\cdot g(x)\\ 1\cdot g(x)\end{bmatrix}\end{aligned}$$





# 非系统码生成矩阵

$$=\mathbf{m}\begin{bmatrix}g_{n-k}&\cdots&g_1&g_0&0&0&\cdots&0&0\\0&g_{n-k}&\cdots&g_1&g_0&0&\cdots&0&0\\0&0&g_{n-k}&\cdots&g_1&g_0&\cdots&0&0\\0&\vdots&\ddots&&&\vdots&\ddots&&\vdots\\0&0&0&\cdots&0&g_{n-k}&\cdots&g_1&g_0\end{bmatrix}\begin{bmatrix}x^{n-1}\\x^{n-2}\\\vdots\\x\\1\end{bmatrix}$$

所以，码字$\mathbf{c}=\mathbf{mG}$，其中

$$\mathbf{G}=\begin{bmatrix}1&g_{n-k-1}&\cdots&g_1&1&0&0&\cdots&0&0\\0&1&g_{n-k-1}&\cdots&g_1&1&0&\cdots&0&0\\0&0&1&g_{n-k-1}&\cdots&g_1&1&\cdots&0&0\\0&\vdots&\ddots&&&&\ddots&&\vdots&\\0&0&0&\cdots&0&1&g_{n-k-1}&\cdots&g_1&1\end{bmatrix}$$

即为循环码的生成矩阵，它是$k\times n$阶循环矩阵。显然，它不是系统的。

由$\mathbf{G}\cdot\mathbf{H}^T=0$可得非系统形式的一致校验矩阵$\mathbf{H}$。





# 系统循环码的编码

系统形式的循环码生成矩阵，由非系统形式的生成矩阵通过行变换而获得。也可以根据下面的定理直接求得。

**定理 (系统循环码)**

**设循环码的生成多项式为$g(x)$，待编码的消息多项式为$u(x)$，且$g(x)$和$u(x)$的次数分别是$n-k$和$k-1$，则**

$$\mathbf{c}(x)=u(x)x^{n-k}+u(x)x^{n-k}|_{\ \mod g(x)} \tag{7}$$

**为码多项式，用此方法编得的码字为系统循环码。**

**证明：**系统循环码是消息部分在左、一致校验部分在右的结构，即码多项式的第$n-1$次至$n-k$次的系数是信息位，而其余为校验位。

因为码多项式一定是生成多项式的倍式，故有

$$\mathbf{c}(x)=u(x)x^{n-k}+r(x)\equiv 0|_{\ \mod g(x)} \tag{8}$$

## 系统循环码的编码

其中，$g(x)=x^{n-k}+g_{n-k-1}x^{n-k-1}+\cdots+g_1x+1$是生成多项式，
$u(x)=u_{k-1}x^{k-1}+u_{k-2}x^{k-2}+\cdots+u_1x+u_0$是信息多项式，
$r(x)=r_{n-k-1}x^{n-k-1}+r_{n-k-2}x^{n-k-2}+\cdots+r_1x+r_0$是校验多项式，其系数是消息码元的校验位。

由式(8)，可得

$$r(x)=\mathbf{c}(x)+u(x)x^{n-k}\equiv u(x)x^{n-k}\pmod{g(x)}$$

因此，由式(7)构造的码为系统循环码。 □

由式(8)，$u(x)x^{n-k}+r(x)=q(x)g(x)$，有

$$u(x)x^{n-k}=q(x)g(x)+r(x)$$





## 循环码编码举例

**例：**$(7,4)$码的生成多项式$g(x)=x^3+x+1$，已知信息多项式$u(x)=x^2+1$，分别求其非系统形式和系统形式的编码输出。

**解：**非系统形式，直接将信息多项式与生成多项式相乘，得

$$\mathbf{c}(x)=g(x)u(x)=(x^3+x+1)(x^2+1)=x^5+x^2+x+1$$

即$\mathbf{c}=0100111$。也可用生成矩阵求。

系统形式，由式(7)，

$$r(x)=u(x)x^{n-k}\mod g(x)=x^5+x^3\mod g(x)$$

而$x^5+x^3=x^2g(x)+x^2$，所以$r(x)=x^2$。
因此

$$\mathbf{c}(x)=x^3u(x)+r(x)=x^5+x^3+x^2$$

即$\mathbf{c}=0101100$。





## 系统码的生成矩阵

根据上述定理，可得如下系统码生成矩阵的求解方法。

对所有的$i=0,1,\ldots,k-1$，用生成多项式$g(x)$除$x^{n-k+i}$，有

$$x^{n-k+i}=a_i(x)g(x)+b_i(x)$$

其中$b_i(x)=b_{i,n-k-1}x^{n-k-1}+\cdots+b_{i,1}x+b_{i,0}$是余式。

因此$x^{n-k+i}+b_i(x)$是$g(x)$的倍式，即码多项式。由此得到系统形式的生成矩阵为

$$\mathbf{G}=\begin{bmatrix}1&0&\cdots&0&b_{k-1,n-k-1}&\cdots&b_{k-1,1}&b_{k-1,0}\\0&1&\cdots&0&b_{k-2,n-k-1}&\cdots&b_{k-2,1}&b_{k-2,0}\\\vdots&\vdots&\ddots&\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&\cdots&1&b_{0,n-k-1}&\cdots&b_{0,1}&b_{0,0}\end{bmatrix}$$

由$\mathbf{G}\cdot\mathbf{H}^T=0$可得系统形式的一致校验矩阵。





## 系统码的一致校验矩阵

由于，

$$\mathbf{G}=\begin{bmatrix}x^{n-1}+x^{n-1}&\mod g(x)\\x^{n-2}+x^{n-2}&\mod g(x)\\\vdots&\\x^{n-k}+x^{n-k}&\mod g(x)\end{bmatrix}=\begin{bmatrix}&x^{n-1}&\mod g(x)\\\mathbf{I}_k&x^{n-2}&\mod g(x)\\&\vdots&\\&x^{n-k}&\mod g(x)\end{bmatrix}$$

所以：

$$\mathbf{H}^T=\begin{bmatrix}x^{n-1}&\mod g(x)\\x^{n-2}&\mod g(x)\\&\vdots\\x^{n-k}&\mod g(x)\\x^{n-k-1}&\\&\vdots\\&1\end{bmatrix}=\begin{bmatrix}x^{n-1}\\x^{n-2}\\\vdots\\x^{n-k}\\x^{n-k-1}\\\vdots\\1\end{bmatrix}\mod g(x)$$

# 系统码生成矩阵求解举例

**例：** 求生成多项式$g(x)=x^3+x+1$的$(7,4)$循环码的系统生成矩阵和一致校验矩阵。

**解：** 由生成多项式，有

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

即

$$b_0=x+1 \quad b_1=x^2+x \quad b_2=x^2+x+1 \quad b_3=x^2+1$$

由此可得系统生成矩阵和一致校验矩阵为

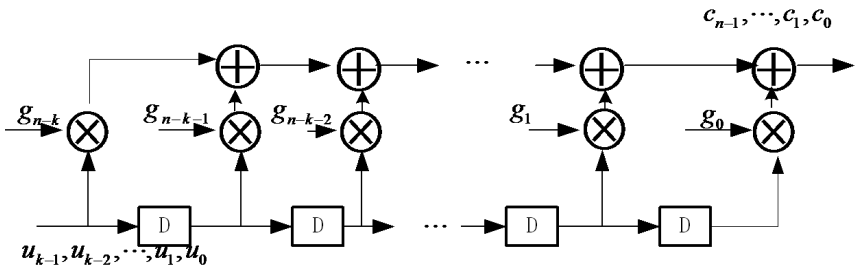
$$\mathbf{G}=\begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix} \qquad \mathbf{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$





# 非系统循环码的编码电路

消息多项式与生成多项式直接相乘得到的码多项式是非系统形式的循环码。编码电路如下图所示。

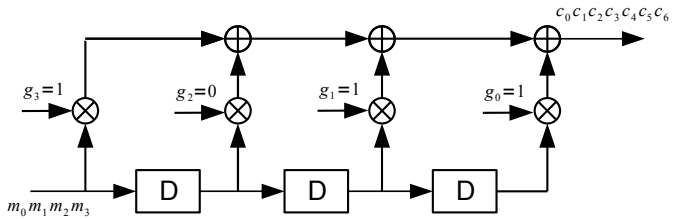







# 非系统循环码的编码电路—举例

$(7,4)$码，$g(x)=x^3+x+1$。




| 输入 | 寄存器状态 | 输出 | 输入 | 寄存器状态 | 输出 |
|---|---|---|---|---|---|
| | 0 0 0 | | 1 | 1 0 1 | 0 |
| 0 | 0 0 0 | 0 | 0 | 0 1 0 | 1 |
| 1 | 1 0 0 | 1 | 0 | 0 0 1 | 1 |
| 0 | 0 1 0 | 0 | 0 | 0 0 0 | 1 |



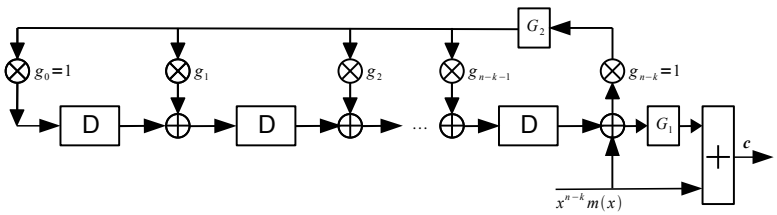


# 系统循环码的编码电路

由系统循环码编码定理，可以得到如下的编码方法：

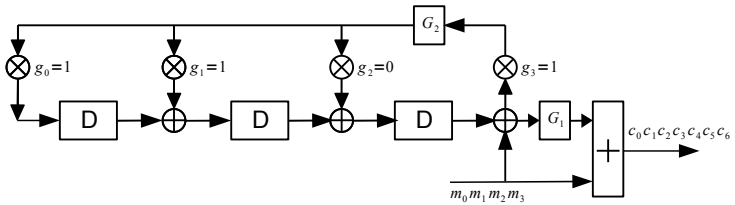
1. 信息序列$m(x)$乘以$x^{n-k}$；
2. 用$g(x)$除$m(x)x^{n-k}$，得到余式$r(x)=m(x)x^{n-k} \mod g(x)$；
3. 输出码字序列$m(x)x^{n-k}+r(x)$。

系统循环码的编码主要是求校验位，而求校验位要以$g(x)$为模做除法。除法电路可用反馈移位寄存器等数字电路来实现，下图为用$n-k$级移位寄存器实现系统循环码编码的一般结构。



# 系统循环码的编码电路—举例

$(7,4)$码，$g(x)=x^3+x+1$。




| 输入 | 寄存器状态 | 输出 | 输入 | 寄存器状态 | 输出 |
|---|---|---|---|---|---|
| | 0 0 0 | | 1 | 0 0 1 | 1 |
| 0 | 0 0 0 | 0 | 0 | 0 0 0 | 1 |
| 1 | 1 1 0 | 1 | 0 | 0 0 0 | 0 |
| 0 | 0 1 1 | 0 | 0 | 0 0 0 | 0 |





# 循环码的伴随式译码

循环码是线性分组码的一种，故线性分组码的译码也完全适用于循环码。但循环码具有循环特性，各种译码算法、电路等有可能利用循环码的循环特性来简化译码。

到目前为止用得最多的还是伴随式译码，常见的译码电路有梅吉特译码器、捕错译码器等。除了伴随式译码外，还有软件译码、大数逻辑译码、软判决译码等译码方案。

设发送的码字为$\mathbf{c}$、错误图样为$\mathbf{e}$，则接收端收到的序列为

$$\mathbf{r}=\mathbf{c}+\mathbf{e}$$

由伴随式定义可知，相应的伴随式为

$$\mathbf{s}=\mathbf{r}\cdot\mathbf{H}^T=(\mathbf{c}+\mathbf{e})\cdot\mathbf{H}^T=\mathbf{e}\cdot\mathbf{H}^T$$

伴随式也可以用多项式表示。





# 伴随式的多项式表示

用生成多项式$g(x)$去除接收多项式$r(x)$，有

$$r(x)=a(x)g(x)+s(x)$$

式中$s(x)$为$g(x)$除$r(x)$所得的余式，次数最高为$n-k-1$，当且仅当$r(x)$是码多项式时，$s(x)$为零。对接收多项式，$s(x)$的$n-k$个系数同样构成伴随式矢量$\mathbf{s}$，因此$s(x)$是伴随多项式。

接收矢量的伴随多项式中同样包含有接收矢量中错误图样的信息，因此可以用它来译码。而且可以用一个和发送端编码器相类似的除法电路来实现伴随式的计算。





# 伴随式的循环性

定理 (伴随式的循环性)

**令$s(x)$是接收多项式$r(x)=r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\cdots+r_0$ 的伴随多项式，则用生成多项式$g(x)$除$xs(x)$所得之余式$s^{(1)}(x)$，就是$r(x)$循环移位一次$r^{(1)}(x)$的伴随式。**

**证明：**因为

$$\begin{aligned}xr(x)&=r_{n-1}x^n+r_{n-2}x^{n-1}+\cdots+r_1x^2+r_0x\\&=(r_{n-1}x^n+r_{n-1})+r_{n-2}x^{n-1}+\cdots+r_1x^2+r_0x+r_{n-1}\\&=r_{n-1}(x^n+1)+r^{(1)}(x)\end{aligned}$$

所以

$$r^{(1)}(x)=r_{n-1}(x^n+1)+xr(x)\tag{9}$$

# 伴随式的循环性

用$g(x)$去除$r^{(1)}(x)$，得

$$r^{(1)}(x) = q(x)g(x) + \rho(x) \tag{10}$$

式中$\rho(x)$是$r^{(1)}(x)$的伴随式。

因为$(x^n+1)$是$g(x)$的倍式，式(9)的右端可写为

$$r_{n-1}(x^n+1) + xr(x) = r_{n-1}g(x)h(x) + x\left[a(x)g(x) + s(x)\right] \tag{11}$$

将式(10)和(11)代入式(9)，有

$$xs(x) = \left[q(x) + r_{n-1}h(x) + xa(x)\right]g(x) + \rho(x)$$

即$\rho(x) = s^{(1)}(x)$ □

**推论：** 用生成多项式$g(x)$除$x^i s(x)$所得的余式$s^{(i)}(x)$，是$r(x)$的$i$次循环移位$r^{(i)}(x)$的伴随式。





# 伴随式译码步骤

伴随式译码步骤：

1. 由$r(x)$做除法求余数得$s(x)$；
2. 利用伴随式和标准阵中的陪集首之间的一一对应关系，由伴随式可以确定$e(x)$；
3. 将$e(x)$和$r(x)$模2相加，即得译码输出$v(x)$。

上述译码步骤可以用软件实现，也可以用硬件电路来实现。梅吉特译码器是一个$(n,k)$循环码的通用译码器，由三部分组成：伴随式移位寄存器、错误图样检测器和存储接收矢量的缓冲寄存器。接收多项式从左端移入伴随式寄存器。只要简单地把一个错误数据由左端通过异或门供给移位寄存器就可以消除错误数据对伴随式的影响。

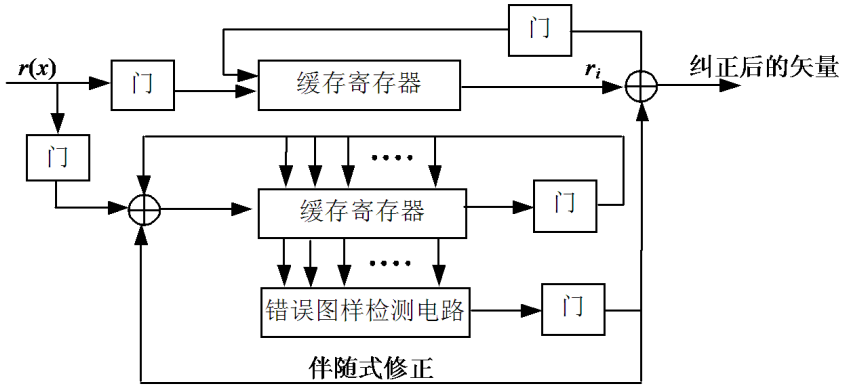
梅吉特译码器原则上可用于任何循环码，其核心部件是错误图样检测电路，整个译码的复杂度也取决于它，因此设计合适的错误图样检测电路是梅吉特译码器的关键。





# 梅吉特译码器

循环码的梅吉特译码器如下图所示。





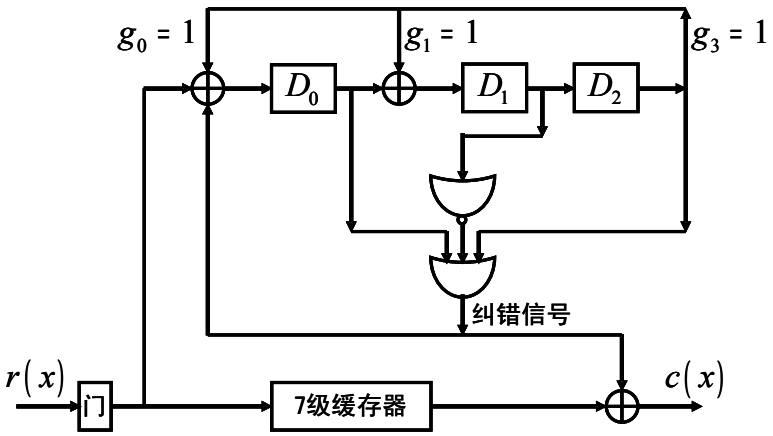



# 译码算法描述

1. 接收矢量全部移入伴随式寄存器从而得到伴随式。与此同时，接收矢量也存入了缓冲寄存器。
2. 将伴随式读入检测器，以探测相应的错误图样。
   检测器思想为：当且仅当伴随式移位寄存器中的伴随式与在最高位$x^{n-1}$有错的错误图样相对应时，才输出“1”。其输出是一个与缓冲寄存器输出符号相应的错误估计值。
3. 由缓冲寄存器读出第一个符号，同时伴随式移位寄存器移位一次。
   如果检测到第一个接收符号是错误的，那么检测器的输出给予纠正。检测器的输出也反馈到伴随式移位寄存器以修正伴随式，消除错误符号在伴随式中的影响。同时得到一个新的伴随式，对应于向右移位一次后所得的修正接收矢量的伴随式。
4. 由步骤3所得的新伴随式，用来检测此刻位于缓冲寄存器最右边一级的第二个接收符号是否有错，译码器重复步骤2和步骤3。第二个接收符号的纠正方法和第一个符号一样。
5. 按步骤2 ～ 4对接收矢量逐个符号地实施译码，直到从缓冲寄存器中读出全部接收矢量为止。

# 梅吉特译码器举例-$(7,4)$循环码

$(7,4)$循环码的梅吉特译码器如下图所示。








# BCH码的特点

循环码中应用最为广泛的二进制BCH码，是以三个发明者的名字Bose、Chaudhuri和Hocquenghem命名的。BCH码具有良好的通用性和较高的编码效率，解码算法简单实用，当码长较长时，BCH码的性能超过了所有具有相同码长和编码效率的其他编码。

BCH码是循环码中的一类，因此它具有分组码、循环码的一切性质。

BCH码的最大特点是用生成多项式根的个数（包含$2t$个连续幂次的根）保证码的最小距离。它明确界定了码长、一致校验位数目、码的最小距离。在同样的编码效率情况下，纠、检错的能力均较强；特别适合于不太长的码，在无线通信系统中获得广泛应用。另外，BCH码也是现阶段比较容易实现的一种码。

所谓构造BCH码，实质上就是找出它的生成多项式。除了找$x^n+1$的因式外，BCH码的构造有更简单的办法，即生成多项式由它的取自伽罗华域$GF(2^m)$中的根来确定。





# BCH码的定义

定义 (二元本原BCH码定义一)

**令$\alpha$为$GF(2^m)$中的本原元，若多项式$g(x)$是以$\alpha,\alpha^2,\alpha^3,\ldots,\alpha^{2t}$为根的$GF(2)$上的最低次多项式，则由它生成的码长为$2^m-1$、纠$t$个错误的码为二元本原*BCH*码。**

若$g(x)$的根是$GF(2^m)$中的任意若干元素，则生成的循环码是一般循环码。

若$g(x)$以非本原元$\beta$的连续幂次为根，则生成非本原BCH码。此时$n\neq 2^m-1$，但是它的因子。Golay码即为非本原BCH码。

定义 (二元本原BCH码定义二)

**对任何$m\geq 3$的正整数和$t, t<2^{m-1}$，若存在一循环码，其码长$n=2^m-1$；一致校验位数目$n-k\leq mt$；最小距离$d_{\min}\geq 2t+1$，能纠$t$个或更少个错误的任何组合，称之为纠正$t$个错误的二元本原*BCH*码。**

通过后面的分析可以知道，这两种定义是一致的。





# 本原BCH码的生成多项式

若$\alpha$为$GF(2^m)$中的本原元，码长为$2^m-1$、纠$t$个错误的二元BCH码的生成多项式$g(x)$是$GF(2)$上的**最低次多项式**，它以$\alpha,\alpha^2,\alpha^3,\ldots,\alpha^{2t}$为根，即对$1\leq i\leq 2t$，$g(\alpha^i)=0$。根据$GF(2^m)$域的性质，这些根的共轭元也是$g(x)$的根。

令$\phi_l(x)$是$\alpha^l$的最小多项式，则必有

$$g(x)=\mathrm{LCM}\{\phi_1(x),\phi_2(x),\ldots,\phi_{2t}(x)\}$$

对偶数$i$，总可以表示为$i=i'\cdot 2^l$（$i'$为奇数，$l\geq 1$），因此$\alpha^i=(\alpha^{i'})^{2^l}$是$\alpha^{i'}$的共轭元，所以$\alpha^i$和$\alpha^{i'}$有相同的最小多项式，从而可知BCH码的生成多项式$g(x)$可写为

$$g(x)=\mathrm{LCM}\{\phi_1(x),\phi_3(x),\ldots,\phi_{2t-1}(x)\}$$

由于每个最小多项式的次数小于等于$m$，所以$g(x)$次数至多为$mt$，即码的一致校验位数目$n-k$至多等于$mt$。

由于$\alpha$为本原元（它生成$GF(2^m)$中的所有非零元素），又由于是用$\alpha$和$\alpha$的各次幂的最小多项式构造了BCH码的生成多项式，故称上述方法生成的BCH码为本原BCH码。

# 本原BCH码–举例

**例：**考虑由$p(x)=x^4+x+1$为本原多项式而生成的$GF(2^4)$域，其本原元为$\alpha$，试求出可纠1个错、2个错、3个错的BCH码的生成多项式。

**解：**对于由本原多项式$p(x)=x^4+x+1$生成的$GF(2^4)$，可以求出其所有元素的最小多项式：

| | |
|---|---|
| $\alpha,\alpha^2,\alpha^4,\alpha^8$ | $x^4+x+1$ |
| $\alpha^3,\alpha^6,\alpha^9,\alpha^{12}$ | $x^4+x^3+x^2+x+1$ |
| $\alpha^5,\alpha^{10}$ | $x^2+x+1$ |
| $\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$ | $x^4+x^3+1$ |

对$t=1,n=15$，有$g(x)=\mathrm{LCM}\{\phi_1(x)\}=x^4+x+1$。即BCH $(15,11,1)$的生成多项式为$g(x)=x^4+x+1$。

$\alpha^7$也是本原元，它的连续两个幂次的最小多项式为$x^4+x^3+1$ 也可以生成BCH $(15,11,1)$。





# 本原BCH码–举例

另外一个4次多项式$x^4+x^3+x^2+x+1$是非本原元素$\alpha^3$（阶为5）及其共轭元的最小多项式，由于不是本原元素的连续幂次，所以用该多项式生成的$(15,11)$循环码的最小距离是2，性能很差，没有纠错能力。

实际上，该多项式的根是$\alpha^3$的$1-4$次幂，它整除$x^5+1$，所以用它可以生成$(5,1)$非本原BCH码，最小距离为5，纠两位错。

对$t=2,n=15$，有$g(x)=\mathrm{LCM}\{\phi_1(x),\phi_3(x)\}=(x^4+x+1)(x^4+x^3+x^2+x+1)=x^8+x^7+x^6+x^4+1$。即BCH $(15,7,2)$的生成多项式为$g(x)=x^8+x^7+x^6+x^4+1$。

以本原元素$\alpha^7$的连续四个幂次为根的多项式为$g(x)=x^8+x^4+x^2+x+1$也生成BCH $(15,7,2)$。

两个4次本原多项式相乘可以得到一个8次多项式

$$(x^4+x+1)(x^4+x^3+1)=x^8+x^7+x^5+x^4+x^3+x+1$$

用该多项式可以生成$(15,7)$循环码，最小距离为3，可以纠1位错，但由于不是用本原元素的连续幂次生成的，所以性能差。





# 本原BCH码–举例

对$t=3,n=15$，有

$$\begin{aligned}g(x)&=\mathrm{LCM}\{\phi_1(x),\phi_3(x),\phi_5(x)\}\\&=(x^4+x+1)(x^4+x^3+x^2+x+1)(x^2+x+1)\\&=x^{10}+x^8+x^5+x^4+x^2+x+1\end{aligned}$$

即BCH $(15,5,3)$的生成多项式$g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1$。

由实际计算可知，用这种求几个最小多项式乘积的方法来求生成多项式，数学模型清晰，计算较为简单，可以编制计算机程序得出各种$(n,k)$的二元本原BCH码的生成多项式。

下表给出了部分二元本原BCH码的生成多项式，表中的数字代表$g(x)$的系数，是以八进制形式给出的，比如$(15,5)$码系数为2467，化为二进制是10 100 110 111，最右边的数字对应于$g(x)$的零次系数。





# 部分二元本原BCH码

| $n$ | $k$ | $t$ | $g(x)$ |
|---|---|---|---|
| 7 | 4 | 1 | 13 |
| 15 | 11 | 1 | 23 |
| | 7 | 2 | 721 |
| | 5 | 3 | 2,467 |
| 31 | 26 | 1 | 45 |
| | 21 | 2 | 3,551 |
| | 16 | 3 | 107,657 |
| | 11 | 5 | 5,423,325 |
| | 6 | 7 | 313,365,047 |
| 63 | 57 | 1 | 103 |

| $n$ | $k$ | $t$ | $g(x)$ |
|---|---|---|---|
| 63 | 51 | 2 | 12,471 |
| | 45 | 3 | 1,701,317 |
| | 39 | 4 | 166,623,567 |
| | 36 | 5 | 1,033,500,423 |
| | 30 | 6 | 157,464,165,547 |
| | 24 | 7 | 17,323,260,404,441 |
| | 18 | 10 | 1,363,026,512,351,725 |
| | 16 | 11 | 6,331,141,367,235,453 |
| | 10 | 13 | 472,622,305,527,250,155 |
| | 7 | 15 | 5,231,045,543,503,271,737 |

摘自：陈运主编，信息论与编码（第2版），电子工业出版社

## BCH码结构

设BCH码的码多项式为

$$v(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \cdots + v_1 x + v_0$$

由生成多项式$g(x)$能除尽$v(x)$可知，对$1 \le i \le 2t$，若$\alpha^i$是生成多项式$g(x)$的根，则必然也是$v(x)$的根，故有

$$v(\alpha^i) = v_{n-1}\alpha^{i(n-1)} + v_{n-2}\alpha^{i(n-2)} + \cdots + v_1\alpha^i + v_0 = 0$$

写成矩阵形式，有

$$\begin{bmatrix} v_0 & v_1 & \cdots & v_{n-1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \alpha^i \\ (\alpha^i)^2 \\ \vdots \\ (\alpha^i)^{n-1} \end{bmatrix} = 0 \tag{12}$$





## BCH码结构—一致校验矩阵

对$1 \le i \le 2t$，BCH码的校验矩阵为

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^3)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \cdots & (\alpha^{2t})^{n-1} \end{bmatrix} \tag{13}$$

由式(12)和式(13)可得：

$$\mathbf{v} \cdot \mathbf{H}^T = 0$$

所以码$\mathbf{v}$是$\mathbf{H}$的零化空间，$\mathbf{H}$是码$\mathbf{v}$的一致校验矩阵，码$\mathbf{v}$的各元素取自$GF(2^m)$。

**定理 (二元本原BCH码的最小距离)**

纠$t$个错误的二元本原*BCH*码的最小距离$d_{\min} \ge 2t+1$。





## BCH码结构—码的最小距离$d_{\min} \ge 2t+1$

**证明：** 根据码的最小重量与一致校验矩阵的关系，即要证明$\mathbf{H}$的任意$2t$列均线性无关，也即任选$\mathbf{H}$的$2t$列构成的矩阵应该是满秩的。

任取$\mathbf{H}$第$0 \le j_1, j_2, \ldots, j_{2t} \le n-1$列，构成矩阵

$$\mathbf{A} = \begin{bmatrix} \alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{2t}} \\ (\alpha^2)^{j_1} & (\alpha^2)^{j_2} & \cdots & (\alpha^2)^{j_{2t}} \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{2t})^{j_1} & (\alpha^{2t})^{j_2} & \cdots & (\alpha^{2t})^{j_{2t}} \end{bmatrix}$$

$$= \begin{bmatrix} \alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{2t}} \\ (\alpha^{j_1})^2 & (\alpha^{j_2})^2 & \cdots & (\alpha^{j_{2t}})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{j_1})^{2t} & (\alpha^{j_2})^{2t} & \cdots & (\alpha^{j_{2t}})^{2t} \end{bmatrix}$$

$\alpha$是本原元素，$t < 2^{m-1} < 2^m - 1$，故$\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_{2t}}$是$2t$个不同的元素，即$\mathbf{A}$是范德蒙阵，所以满秩。





## BCH码结构—H矩阵的简化

若$\alpha^j$是$\alpha^i$的共轭元，则当且仅当$v(\alpha^i) = 0$时，$v(\alpha^j) = 0$。即若$\mathbf{v} = (v_{n-1}, v_{n-2}, \cdots, v_1, v_0)$和$\mathbf{H}$矩阵第$i$行的内积为零，则$\mathbf{v}$和$\mathbf{H}$的第$j$行的内积也为零，因此可消去$\mathbf{H}$的第$j$行。对于BCH码的情况，校验矩阵$\mathbf{H}$可以改写为

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \cdots & (\alpha^3)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2t-1} & (\alpha^{2t-1})^2 & \cdots & (\alpha^{2t-1})^{n-1} \end{bmatrix}$$

**例：** $m=3$，$p(x) = x^3 + x + 1$，$n = 2^m - 1 = 7$，$t < 2^{m-1} = 4$。

当$t=1$时，$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \end{bmatrix}$。

当$t=2$时，

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \end{bmatrix}$$

## 多进制BCH码

多进制有时具有更佳的性能。

考虑一个二进制码$(n,k)=(7,3)$，整个$n$元组空间共有$2^n=2^7=128$个$n$元组，其中$2^k=2^3=8$是码字，许用码组占的比例为$1/16$；

再考虑一个非二进制码$(n,k)=(7,3)$，每个码元由$m=3$比特组成，$n$元组空间共有$2^{mn}=2^{21}=2097152$个$n$元组，其中$2^{km}=2^9=512$是码字，许用码组占的比例为$1/4096$。

这个比例随着$m$的增大而减小，而更重要的一点是，$n$元组中用于码字的比例越小，获得的$d_{\min}$越大。因此研究多进制的编码具有明确的实用意义。

对于二元BCH码，码长$n=2^m-1$，若要纠$t$个或更少个错的任意组合，最多需要$mt$个一致校验元。

对$q$进制BCH码，码长$n=q^s-1$，纠$t$个或更少个错的任意组合的多元BCH码，最多需要$2st$个一致校验元。





## 多进制BCH码

令$\alpha$是$GF(q)$中的本原元，生成多项式$g(x)$的系数取自$GF(q)$，且以$\alpha,\alpha^2,\alpha^3,\ldots,\alpha^{2t}$为根的最低次多项式，即

$$g(x)=\mathrm{LCM}\{\phi_1(x),\phi_2(x),\ldots,\phi_{2t}(x)\}$$

$g(x)$的次数至多是$2st$，即用$g(x)$生成的码的一致校验元数目不多于$2st$个。显然，当$q=2$时为二元BCH码。





## R-S码（Reed-Solomon）

当$s=1$时的多元BCH码就是R-S码。所以，R-S码是非二进制BCH码的一个特殊子类。

定义 (R-S码)

**如果多元*BCH*码具有如下参数：码长$n=q-1$，一致校验数目$n-k=2t$，最小距离$d_{\min}=2t+1$ 则称它为*R-S*码。**

在通信中的多进制，通常是$2^m$时的情况，对R-S码重点讨论$q=2^m$的情况。

R-S码也分为本原R-S码和非本原R-S码。





## R-S码的编码

令$\alpha$是$GF(2^m)$中的本原元，能够纠正$t$比特错误的长为$2^m-1$的本原R-S码，其生成多项式为

$$\begin{aligned}g(x)&=(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{2t})\\&=x^{2t}+g_{2t-1}x^{2t-1}+\cdots+g_2x^2+g_1x+g_0\end{aligned}$$

$g(x)$的系数取自$GF(2^m)$域，以$\alpha,\alpha^2,\ldots,\alpha^{2t}$作为全部根，则生成码是$(n,n-2t)$循环码，R-S码也称为$(n,n-2t)$循环码。码多项式的次数最高为$n-1$，但系数取自$GF(2^m)$。

设消息多项式为

$$a(x)=a_{k-1}x^{k-1}+\cdots+a_2x^2+a_1x+a_0$$

式中$k=n-2t$，系数$a_i$取自$GF(2^m)$。

## R-S码的编码

对于非系统编码，只要将消息多项式与生成多项式相乘就得到码多项式；

对于系统编码，检验多项式为

$$b(x) = x^{2t}a(x)|_{\bmod g(x)} = b_{2t-1}x^{2t-1} + \cdots + b_1 x + b_0$$

码多项式为$v(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \cdots + v_1 x + v_0$

**例：**考虑$GF(2^8)$上的R-S码，要求$t = 16$。

**解：**$q = 256$，$2^8 - 1 = 255$，选$n = 255$，$d_{\min} = 2t + 1 = 33$，$n - k = 2t = 32$，$k = n - 2t = 255 - 32 = 223$ 该R-S码为$(255, 223)$码。
$g(x) = \prod_{i=1}^{32}(x + \alpha^i)$是$x$的32次多项式，系数取自$GF(256)$。

该编码器已做成VLSI，该码在美国国家航空和宇宙航行局、欧洲太空署均作为深空编码。





## R-S码的编码

R-S码由于性能优良而得到了广泛应用，它的主要优点包括：

1. R-S码的距离特性好，其纠错能力已发挥到极限，是MDC码（极大最小距离码）；
2. 存在一种有效的硬判决译码算法，使得在许多需要长码的应用场合，该码能够被实现；
3. $q$进制R-S码的二进制衍生码具有良好的抗突发差错能力。

对一个编好的$q = 2^m$进制$(n, k)$ R-S码，如果不以$q$进制调制发送（1符号间隔发1码元），而是将每码元对应为$m$比特后以二进制发送（用$m$符号间隔发1码元），实际上就是把$q$进制$(n, k)$R-S码化作了$(mn, mk)$二进制衍生码，这样的二进制衍生码特别适用于纠突发差错。





## 卷积码

在卷积码的约束长度内，前后各组是密切相关的，一个组的监督元不仅取决于本组的信息元，而且取决于前$m$组的信息元，其中$m$是编码记忆深度。

由于卷积码充分利用了各组之间的相关性，$n$和$k$可以用比较小的数，在与分组码同样的信息传输速率和设备复杂性时，卷积码的性能一般比分组码好。

卷积码分析，至今还缺乏分组码那样有效的数学工具，一些好码的参数往往借助于计算机搜索。

卷积码可以用卷积运算的线性方程组来描述。

卷积码的每个$(n, k)$码段称为子码，通常较短，子码内的$n$个码元不仅与该码段内的信息位有关，而且与前面$m$段内的信息位有关。

卷积码常用子码长度、子码中的信息位数目以及编码存储三个参量描述，记为$(n, k, m)$。

如果卷积码的各个子码是系统码，则称该卷积码为系统卷积码，否则为非系统卷积码。

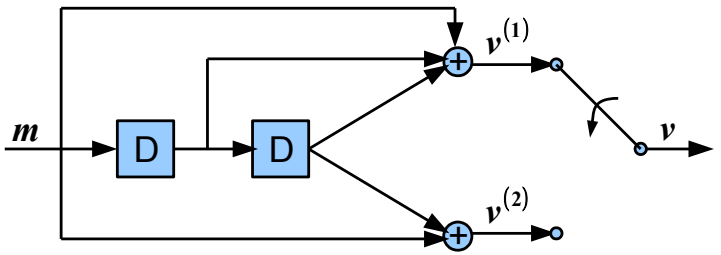




## 卷积码的生成矩阵描述

卷积码的描述方法很多，生成矩阵、状态图、树图等。这里仅介绍生成矩阵。

**例：**下图所示为二元$(2, 1, 2)$卷积码编码器，若信息序列$m = (1101000)$，试求编码输出。



# 卷积码的生成矩阵描述

如果输入一个单位冲激$\delta$，那么$v^{(1)}$,$v^{(2)}$的输出为$v^{(1)} = (1\ 1\ 1)$,$v^{(2)} = (1\ 0\ 1)$。整个过程是线性的，输出序列可以认为是输入序列通过了两个FIR滤波器，那么输出序列就是输入序列同$(1\ 1\ 1)$和$(1\ 0\ 1)$的卷积。这就是卷积码的由来。

测试一下：

$$m = (1\ 1\ 0\ 1\ 0\ 0\ 0) \quad g_1 = (1\ 1\ 1) \quad g_2 = (1\ 0\ 1)$$

$$v_1 = (1\ 0\ 0\ 0\ 1\ 1\ 0) \quad v_2 = (1\ 1\ 1\ 0\ 0\ 1\ 0)$$

$g$也被称为生成序列。

写成矩阵形式$G = [111, 101]$或者

$$G(D) = [g_1(D), g_2(D)] = [1 + D + D^2, 1 + D^2]$$

设：$v^{(1)}(D) = u(D) * g_1(D), v^{(2)}(D) = u(D) * g_2(D)$，则

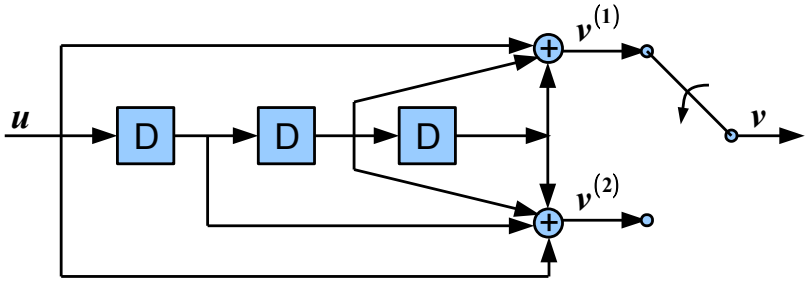
$$v(D) = u(D) * G(D) = v^{(1)}(D^2) + Dv^{(2)}(D^2)$$





# 卷积码的生成矩阵描述

**例：**下图所示为二元$(2, 1, 3)$卷积码编码器，若信息序列$u = (10111)$，试求编码输出。








# Viterbi译码

维特比（VB，Viterbi）译码算法是一种**最大似然译码**算法，由维特比提出的，它在下述意义上是最佳的，即它对整个信息比特序列译码的差错概率最小。

它的基本思想是把接收到的矢量，和网格图上诸种可能的路径比较，删去距离大的路径，保留距离小的路径（发生的可能性大），以距离最小路径作为发送码字的估计。

通过观察网格图，可以发现卷积码的以下特点：

1. $(n, k, m)$卷积码的状态数共有$2^{km}$个。
2. 每个状态有$2^k$种可能的输出。
3. 输出只与当前时刻的输入以及状态有关，与之前的输出无关。
   这提醒我们，若两条路径在某点汇聚，则这两条路径在此后就合成一条路径。即在汇聚点比较两条路径与比较整个序列的两条路径是一致的。





# Viterbi译码

下面通过一个例子来说明维特比译码过程。

**例：**已知编码器是本节开始时给出的$(2, 1, 2)$卷积编码器。现接收矢量为

$$\mathbf{r} = (00\ 10\ 01\ 00\ 00\ 00\ 00\ 00)$$

试求译码输出。

**解：**画出译码过程的网格图，如图所示，接收矢量在图的上方标识出。

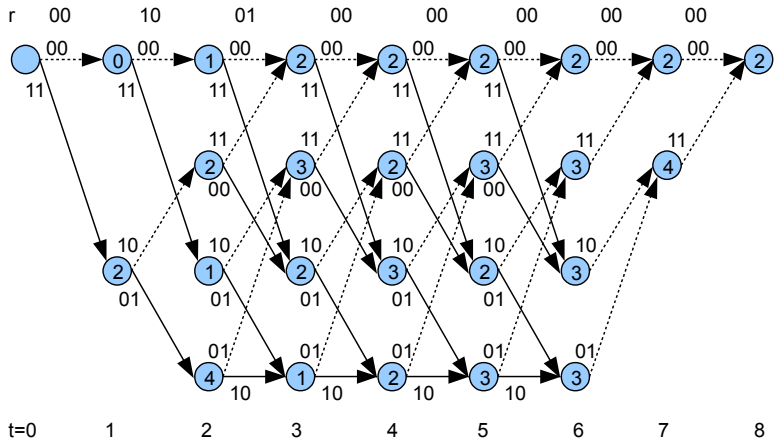
前$m$条支路互不相交，因此只要计算其汉明距离（等价于计算似然函数）并依次累加即可。从第$m+1$条支路开始，每个状态都有$2^k$条路径汇聚，根据前面的分析。需要：

1. 对每个状态，计算各条路径的汉明距离。计算方法为与前面的汉明距离累**加**。
2. **比**较计算出的各条路径的汉明距离。
3. **选**择汉明距离最小的路径保留，删除其它的路径。

每个状态保留一条路径，每一级有$2^2$个状态，因此共保留$2^2$条路径。

# Viterbi译码

最终的译码结果如下图所示。








# Viterbi译码

结果：

$$\begin{array}{cccccccccc} & S_0 & \to S_0 & \to S_0 & \to S_0 & \to S_0 & \to S_0 & \to S_0 & \to S_0 & \to S_0 \\ t= & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$

被选择为最佳译码路径，对应的译码矢量为

$$\hat{\mathbf{v}} = (00\ 00\ 00\ 00\ 00\ 00\ 00\ 00)$$

译码消息为$\hat{\mathbf{c}} = (000000)$。

注意：卷积码的结束状态一般都是已知的。另外，当两个以上的路径都是最小汉明距离时，可以任意选择一条路径作为“幸存路径”。

译码复杂度。设序列长为$L$，则最大似然译码$O(2^L)$，而维特比译码$O(L \cdot 2^m)$。

维特比译码并不能纠正所有可能发生的错误，当错误模式超出其纠错能力时，译码后的输出序列就会有错误。





# 本章小结

1. 编码概述
   编码的目的、问题；纠错码的分类；分组码举例，卷积码的描述。
2. 近似（抽象）代数基础
   群，子群，陪集及陪集展开；环；域，伽罗华域，二元域及其扩域，多项式及根。
3. 线性分组码
   定义，最小重量、最小距离、纠错能力；生成矩阵与一致校验矩阵；等价码；系统码与标准生成阵；伴随式译码、标准阵译码；分组码的性能，汉明界与完备码；汉明码。
4. 循环码
   定义，多项式的移位与循环移位；生成多项式及性质；码多项式的充要条件；非系统及系统码的生成矩阵、编码电路；循环码的伴随式译码，Meggit译码器；BCH码及RS码。
5. 卷积码
   生成矩阵描述；Viterbi译码。