第8章　差错控制编码

- 数字信号在传输过程中，由于信道传输特性不理想及加性噪声的影响，不可避免地会发生错误。香农编码理论认为，可通过以下三方面的措施来减小误码率：
- (1)提高信道容量。合理设计基带信号，选择适当的调制、解调方式，增大发送功率，采用均衡技术，滤波，采用低噪声器件等都有利于提高信道容量。
- (2)降低编码效率。在一定的信道容量下，降低编码效率，就意味着增加信道容量的冗余度，等效于增大信道容量。
- (3)增加码长。保持编码效率不变，码长增加，码字间的距离就加大，从而提高了可靠性。但是码长越长，延迟也越长，编解码算法越复杂，编解码器越昂贵。
- 差错控制不仅广泛应用于各种通信系统中，而且在计算机、磁记录与存储设备中也得到大量的应用。

8.1 差错控制的基本概念

按错码分布规律的不同，可分为三类:

随机性错码。错码的出现是随机的，且错码的出现是统计独立的。它由高斯白噪声引起。

突发性错码。错码是成串集中出现的。也就是说，在短时间内会出现大量错码，而在这些短促的时间区间之间却又存在较长的无错码区间。产生突发错码的主要原因是脉冲干扰和信道中的衰落。

混合性错码。既有随机错码又有突发错码，且哪一种都不能忽略不计。对于不同类型的错码，应采用不同的差错控制技术。

常用的差错控制方法有以下几种:

检错重发(ARQ)：如接收端检测出错码，通知发端重发，直到接收正确为止。此方法只能判断是否有错码，不能判断具体的错码位置。所以，只能检错不能纠错，且需要双向通道。

前向纠错(FEC)：收端能检测出错码，并可以确定错码的位置，并予纠正。此方法只需要单向通道。实时性好，但设备复杂。

反馈校验：接收端将收到的信号原封不动的发回发端，由发端将其与原发信号相比较，如果有错则重发。这种方法需双向通道，效率低，设备简单

检错删除：如：重复发送的的遥测信号。

差错控制编码之所以能进行差错控制，其基本原理可归结为两条:

1. 利用冗余度

差错控制编码就是在信息码元序列后面增加一些监督码元，这些监督码和信码之间有一定的关系，接收端利用这种关系来发现或纠正错码。监督码不荷载信息，它的作用是用来监督信息码在传输中有无差错，对用户来说是多余的，最终也不传送给用户，所以称它是冗余的。

2. 噪声均化(随机化)

就是设法把集中出现的突发性差错分摊开来，变成随机性差错。噪声均化的方法主要有三种。

(1)增加码长。码长越长，每个码组中误码的比例越接近统计平均值，译码产生错误的概率就越小。

(2)卷积。在相邻的若干个码组之间加进了相关性，译码时，结合多个码组的信息来作出判决。加上适当的编译码方法，使错码分散到不同的码组上。

(3)交织。使交织器输出码流的顺序不同于输入的顺序，那么在信道中码流的传输顺序和解交织器输出的顺序也不一样，则信道中的突发性错码能够被均化。

在信息码序列中加*监督码元*，监督码和信息码之间存在一种逻辑关系。因此，收端可以利用这种逻辑关系发现或纠正存在的错码。

一般来说，*监督码元越多*，检、纠错*能力越强*。用降低传输速率换取传输可靠性的提高。

不同的编码方法，有不同的检错或纠错能力。

目标：监督码元要少，检、纠错能力要强。

例：表示天气

信源	发送信息码
晴	0 0
云	0 1
阴	1 0
雨	1 1

错 1 位

接收信息码	判别(错误)
0 1	云
11	雨
0 0	晴
1 0	阴

结论：虽然接收码组有错，但接收端无法识别。

增加一位监督码

信源	发送信息码	监督码
晴	0 0	0
云	0 1	1
阴	1 0	1
雨	1 1	0

错 1 位

接收码组	判别
001、010、100	×
010、001、111	×
100、111、001	×
111、100、010	×

错 2 位

接收码组	判别(错误)
011、110、101	云、雨、阴
000、101、110	晴、阴、雨
110、000、011	雨、晴、云
101、000、011	阴、晴、云

许用码组：有效信息码组

禁用码组：非信息码组

结论：可以检测出 1 位错码，但不能纠错。

码距

增加三位监督码

信源	发送信息码	监督码
晴	0 0	000
云	0 1	011
阴	1 0	101
雨	1 1	110

错 1 位

接收码组	判别
00001,00010,00100,01000,10000	晴
01010,01001,01111,00011,11011	云
10100,10111,10001,11101,00101	阴
11111,11100,11010,10110,01110	雨

错 2 位

接收码组	判别
11000,10100,10010,10001,01100,01010,01001,00110,00101,00011	×
10011,11111,11001,11010,00111,00001,00010,01101,01110,01010	×
01101,00001,00111,00110,11001,11110,11101,10011,10000,10100	×
00110,01010,01100,01111,10010,10100,10111,11000,11011,11111	×

结论：能纠正 1 位错码，或检测出 2 位错码。

▼ 分组码

分组码定义：将信息码分组，为每组信息码后附加若干监督码元形成的码集合。

特点：分组码中的监督码元仅监督本码组中的信息码元。

符号：(n, k)

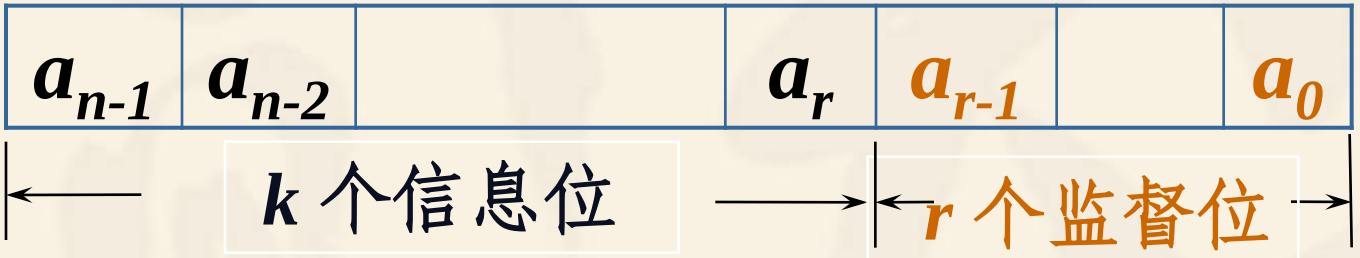
k：码组中信息码元的数目。

n：码组的总位数，又称为码组长度。

$r = n - k$：码组中监督码元的数目。

编码效率：k/n；冗余度：(n-k)/k

结构



码长 $n = k + r$

码组重量：码组中 “1” 的数目。

码距 d：两个码组对应位上不同的码元个数，称为汉明距离。

最小码距 d_0：码集合中任意两两码组间距离的最小值。

▼ 码距与码集合检、纠错能力的关系

- 检测 e 个错码，要求最小码距　$d_0 \geq e+1$
- 纠正 t 个错码，要求最小码距　$d_0 \geq 2t+1$
- 纠正 t 个错码、同时检测 e 个错码，要求最小码距

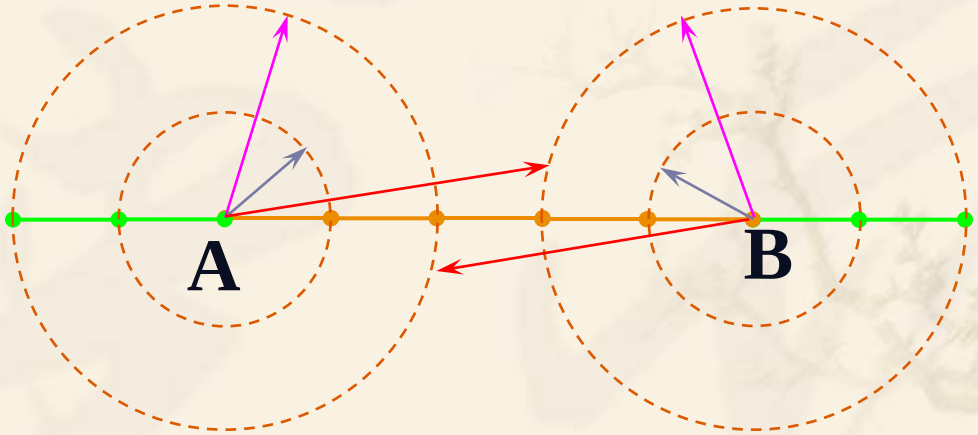
$$d_0 \geq e+t+1 \quad (e>t)$$

例：　$A=(00000)$、$B=(11111)$，　$d_0=5$

$d=1$

$d=2$

$d=3$



结论：$e=4$ 或 $t=2$ 或 $t=1$、$e=3$

1、奇偶监督码

码距为2，能检测奇数个错码

- 一维奇偶监督码： *1* 位监督码；

❖ 奇数监督码：使码组中 *“1”* 的个数为奇数

$$a_{n-1}\oplus a_{n-2}\oplus\ldots\oplus a_0=1$$

❖ 偶数监督码：使码组中 *“1”* 的个数为偶数

$$a_{n-1}\oplus a_{n-2}\oplus\ldots\oplus a_0=0$$

常用

- 二维奇偶监督码（矩阵码）

生成规则： 许用码组写成一行（包括信息码和*1* 位监督码），设共有*m* 行。第 *m+1* 行为按列增加的监督码。（构成监督码行）

例 二维偶数监督码

a_2	a_1	a_0
0	0	0
0	1	1
1	0	1
1	1	0
0	0	0

通式

$$
\begin{array}{cccc}
a_{n-1}^{1} & a_{n-2}^{1} & \cdots\cdots & a_{0}^{1} \\
a_{n-1}^{2} & a_{n-2}^{2} & \cdots\cdots & a_{0}^{2} \\
\vdots & & & \\
a_{n-1}^{m} & a_{n-2}^{m} & \cdots\cdots & a_{0}^{m} \\
c_{n-1} & c_{n-2} & \cdots\cdots & c_{0}
\end{array}
$$

1）设 a_{n-1}^{1} 和 a_{0}^{1} 发生错码，按行无法检测出有错，而按列可检测。

2）能检测突发性错码；适用于突发信道。

3）若仅一行有奇数个错码时，可通过列确定错码位置并纠正。

4）当 $a_{n-1}^{1}\ a_{0}^{1}$ 同时出错，则按行按列均不能检测出有错。
$a_{n-1}^{m}\ a_{0}^{m}$

5）方阵码除了在行列上的错码都为偶数时，无法检测外，其余均能检测。

2. 恒比码

在恒比码中，每个码组均含有相同数目的“1”(和“0”)。这种码在检测时，只要判断接收码组中“1”的数目是否正确，就能判断有无错误。

P286表9-2中的保护电码，每个码组的长度为5，其中恒有3个“1”，称为5/3恒比码。用于我国的汉字电传编码。

从5中取3的组合数$C_3^5=5!/(3!\ 2!)=10$。这10种许用码组恰好可用来表示10个阿拉伯数字。用4位阿拉伯数字表示一个汉字。

在无线电报通信中，广泛采用的是 7/3恒比码，这种码组中总是有3个“1”。共有$7!/(3!\ 4!)=35$种许用码组，它们可用来代表26个英文字母及其他控制符号。

8.3　线性分组码

（可以纠错）

线性码：监督码和信息码之间的关系是线性关系

汉明码的编码原理

一般线性分组码的编码原理

8.3.1 汉明码

分析偶数监督码，寻找逻辑组合

监督方程 $a_{n-1}\oplus a_{n-2}\oplus\cdots\cdots\oplus a_0=0$

一位监督码对应一个监督方程，即对应一个校正子

所以解码就是要计算

校正子 $s=a_{n-1}\oplus a_{n-2}\oplus\cdots\cdots\oplus a_0$

$$s=\begin{cases}0 & \text{无错}\\ 1 & \text{有错}\end{cases}$$

只能表示出错
不能描述错码位置

结论：若增加监督码元，建立多个监督方程，多个校正子就能形成逻辑组合描述错码位置。

r位监督码对应r个校正子，就有2^r种组合，用其中一种组合表示无错，其余2^r-1种组合表示错码的位置。

只纠正一位错码

如果只错一位，分组码(n,k)中的错码有n个可能的位置，要用r位监督码表示这n个错码的位置，$2^r-1\geq n$

为提高编码效率，r取最小值

例：已知$(7,4)$码，$r=3$　　$2^3-1=7=n$

∴ 共有3个监督方程，
构成3个校正子$S_1S_2S_3$
确定监督关系表

监督码出错只与一个校正子有关

$S_1S_2S_3$	
000	无错
001	a_0错
010	a_1错
100	a_2错
110	a_3错
011	a_4错
111	a_5错
101	a_6错

建立监督方程

$$a_6\oplus a_5\oplus a_3\oplus a_2=0$$
$$a_5\oplus a_4\oplus a_3\oplus a_1=0$$
$$a_6\oplus a_5\oplus a_4\oplus a_0=0$$

- 建立编码方程

$$a_2=a_6\oplus a_5\oplus a_3$$
$$a_1=a_5\oplus a_4\oplus a_3$$
$$a_0=a_6\oplus a_5\oplus a_4$$

求码组集合

$k=4$，信息码组有 16 个

能纠正一位错码，且$2^r-1=n$的线性分组码，称为汉明码。

其编码效率为

$k/n=(2^r-1-r)/(2^r-1)=1-r/(2^r-1)=1-r/n$

$a_6\ a_5\ a_4\ a_3$	$a_2\ a_1\ a_0$
0 0 0 0	0 0 0
0 0 0 1	1 1 0
0 0 1 0	0 1 1
0 0 1 1	1 0 1
……	
1 1 0 0	0 1 0
1 1 0 1	1 0 0
1 1 1 0	0 0 1
1 1 1 1	1 1 1

当n很大时，则编码效率接近1。可见，汉明码是一种高效码。

8.3.2 一般线性分组码的编码原理

汉明码的监督方程为

$$a_6 \oplus a_5 \oplus a_3 \oplus a_2 = 0$$
$$a_5 \oplus a_4 \oplus a_3 \oplus a_1 = 0$$
$$a_6 \oplus a_5 \oplus a_4 \oplus a_0 = 0$$

用矩阵表示

$$\left[\begin{array}{c:ccc:c} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array}\right] \cdot \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H = \left[\begin{array}{ccc:c} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array}\right]$$ 监督矩阵

$$A = [a_6 a_5 a_4 a_3 a_2 a_1 a_0]$$

码组向量

记为：$H \cdot A^T = 0$

当 $H = [P \mid I_r]$ *称 H 为典型监督矩阵（含单位阵）*

P 为 $r \times k$ 阶 I_r 为 $r \times r$ 阶

错误图样

根据监督方程确定了编码方程

$$a_2 = a_6 \oplus a_5 \oplus a_3$$
$$a_1 = a_5 \oplus a_4 \oplus a_3$$
$$a_0 = a_6 \oplus a_5 \oplus a_4$$

$$\begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1101 \\ 0111 \\ 1110 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \end{bmatrix} = P \cdot \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \end{bmatrix}$$

两边同取转置 $[a_2 a_1 a_0] = [a_6 a_5 a_4 a_3] P^T = [a_6 a_5 a_4 a_3] \cdot Q$

其中 $Q = P^T$ $k \times r$ 阶

构造生成矩阵 $G = [I_k \mid Q]$ G 为典型生成矩阵

∴ 编码矩阵方程 $A = [a_6 a_5 a_4 a_3 a_2 a_1 a_0] = [a_6 a_5 a_4 a_3] \cdot G$

由典型生成矩阵得出的码组 A是系统码

特点：信息位不变，监督位附加于其后。

生成矩阵 $G=[I_k\ Q]$

$$=\left(\begin{array}{c|c} 1000 & 101 \\ 0100 & 111 \\ 0010 & 011 \\ 0001 & 110 \end{array}\right)_{k\times n} =[I_k\ P^T]$$

G 中每行均为一个码组，且线性无关

若能找到k个线性无关的已知码组，就能构成矩阵G。

- 错误图样——收发码组的关系

令　发码组为 A、收码组为 B

$\therefore$ 错码图样 $E = B - A$

译码运算，当　$BH^T = \begin{cases} 0 & \text{无错} \\ 1 & \text{有错} \end{cases}$

令 $B = E + A$　　$\therefore$　$BH^T = (A + E)H^T$

$\therefore$　$AH^T + EH^T = S$

$S = EH^T$

S 为校正子。说明 S 与 E 有确定的线性关系

若　E 的数目有限，能与 S 一一对应，
则　说明 S 能描述错码的位置，具有纠错能力。

例：（7，4）汉明码，

$$H=\begin{pmatrix}1101100\\0111010\\1110001\end{pmatrix}$$

发码组 $A=1\ 1\ 0\ 0\ 0\ 1\ 0$

收码组 $B=1\ 0\ 0\ 0\ 0\ 1\ 0$

∴ 译码运算

$$BH^T=(1000010)\begin{pmatrix}101\\111\\011\\110\\100\\010\\001\end{pmatrix}=(111)$$ ∴ a_5 错

$$EH^T=(111)$$

含义：错码图样 $E=(0100000)$ 只有一位错码

$S_1S_2S_3$	
000	无错
001	a_0 错
010	a_1 错
100	a_2 错
110	a_3 错
011	a_4 错
111	a_5 错
101	a_6 错

- 线性分组码的性质：封闭性

——线性码中任意两个码组之和仍为这种码中的一个码组

证：　设 A_1 、A_2 为线性码中两个许用码组

$$A_1H^T=0 \qquad A_2H^T=0$$

两式相加　$A_1H^T+A_2H^T=(A_1+A_2)H^T=0$

$$\therefore\ A_1+A_2 \text{ 是许用码组}$$

推广：　1）两个码组间的距离必是另一码组的重量

2）除 全0 码组之外，编码的最小码重是码集合的最小码距。

3)线性分组码中必有全0码；

（信息码全0，监督码全0）

8.4 循环码

循环码是线性分组码中一种重要的编码。它是在严密的代数理论基础上建立起来的。其编码和解码不太复杂，但检(纠)错的能力较强。循环码除了具有线性码的一般性质外，还具有循环性。

码多项式

码多项式——以码组中各码元为系数的多项式

$$T(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

x 仅为码元位置的标记

例：*(110 0101)*　　$T(x) = x^6 + x^5 + x^2 + 1$

码多项式的模运算：

设 多项式 $F(x)$、除式为 $N(x)$　　$\dfrac{F(x)}{N(x)} = Q(x) + \dfrac{R(x)}{N(x)}$

$$F(x) \equiv R(x) \quad [\text{模 } N(x)]; \quad R(x)\text{：余式}$$

注：多项式按模 $N(x)$ 运算过程中，其系数均为**模2 运算**。

例：$\dfrac{x^4+x^2+1}{x^3+1}$

解：

$$\begin{array}{r} x \\ x^3+1 \overline{)\, x^4+x^2+1} \\ \underline{x^4+x} \\ x^2+x+1 \end{array}$$

记为：$x^4+x^2+1 \equiv x^2+x+1$　余式

系数为二进制，只能取0或1，二进制的加减都是一样的

循环码的特性

编码中任意一个码组，左移或右移一位得到的新码组必是该码集合中另一码组。

用码多项式的运算来表示：若$T(x)$对应一个码长为n的许用码组，则$x^i T(x)$按模x^n+1运算后余式$T'(x)$仍为许用码组。

证：令 $x^i \cdot T(x) \equiv T'(x)$ 模 x^n+1

$$\because\ T(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

$$\therefore\ x^i T(x) = a_{n-1}x^{n-1+i} + \cdots + a_{n-1-i}x^{n-1} + \cdots + a_0 x^i$$

$$[x^i T(x)] \equiv a_{n-1-i}x^{n-1} + \cdots + a_0 x^i + a_{n-1}x^{i-1} + \cdots + a_{n-i} \quad \text{模 } x^n+1$$

$\therefore$ $T'(x)$的系数是$T(x)$中系数向左循环移位i次的结果

例：（7，3）循环码，码组为（1100101），验证 $x^3T(x)$ 按模 x^7+1 运算后余式仍是一个许用码组。

解：∵ $T(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$

∴ $T(x)=x^6+x^5+x^2+1$

∵ $x^3T(x)=x^9+x^8+x^5+x^3$

∴ $x^3T(x)\equiv x^5+x^3+x^2+x$

$$
\begin{array}{r}
x^2+x \\
x^7+1\overline{)\,x^9+x^8+x^5+x^3} \\
\underline{x^9+x^2} \\
x^8+x^5+x^3+x^2 \\
\underline{x^8+x} \\
x^5+x^3+x^2+x
\end{array}
$$

∴ 余式$T'(x)$对应码组为（0101110）是$T(x)$码组循环左移三位的结果

8.5 循环码的编译码方法

思路：由码的循环性，可知找到一个码多项式，就能得到其他的码多项式

一个(n,k)码有2^k个不同码组。用$g(x)$表示其中前$(k-1)$位皆为“0”的码组。则$g(x)$，$xg(x)$，$x^2g(x)$，…，$x^{k-1}g(x)$都是该循环码的码组，而且这k个码组是线性无关的，用它们可以构成此循环码的生成矩阵G。

循环码的生成矩阵 G $G(x)=\begin{pmatrix} x^{k-1}\cdot g(x) \\ x^{k-2}\cdot g(x) \\ \vdots \\ x\cdot g(x) \\ g(x) \end{pmatrix}$

k是一个码组中信息码的长度

对g(x)的说明：

是该循环码中阶数最低的码多项式。

在循环码中除全“0”码组外，即连“0”的长度最多只能有(k-1)位。否则在经过若干次循环移位后将得到一个k个信息位全为“0”，但监督位不全为“0”的码组。这显然是不可能的。

——g(x)必须是一个常数项不为“0”的(n-k)次多项式。

而且还是唯一的。因为如果有两个，则由码的封闭性，把这两个相加也应该是一个码组，且此码组多项式的次数将小于(n-k)，即连续“0”的个数多于(n-k)，这是不可能的。

我们称这唯一的(n-k)次多项式g(x)为码的生成多项式。一旦确定g(x)，则整个(n-k)循环码就被确定了。

(7,3)循环码的码组：

$$T(x)=[a_6a_5a_4]G(x)=[a_6a_5a_4]\begin{bmatrix} x^2g(x) \\ xg(x) \\ g(x) \end{bmatrix}$$

$$=(a_6x^2+a_5x+a_4)g(x)$$

这表明，所有码多项式$T(x)$都是$g(x)$倍式，而且任一次数不大于$(k-1)$的多项式乘$g(x)$都是码多项式。

所有的信息码组合

在表11-5中找出生成多项式……

码生成多项式 $g(x)$ 的求解

定理： 循环码(n,k)的$g(x)$是x^n+1的一个$(n-k)$次因子。

证： ∵ 任意一个码多项式$T(x)$都是$g(x)$倍式

令 $T(x)=h(x)g(x)$

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

$$=1+\frac{T(x)}{x^n+1}$$

g(x)为一(n-k)次多项式，故x^kg(x)为一n次多项式。余式$T(x)$也是一个许用码组。

∴ $x^k g(x)=x^n+1+T(x)$

∴ $x^n+1=x^k g(x)+T(x)$

$=x^k g(x)+h(x)g(x)=[x^k+h(x)]g(x)$

例：已知 (7,3) 循环码，求码组集合、监督矩阵 H。

解：$\because n = 7$

$$\therefore\ x^7+1=(x+1)(x^6+x^5+x^4+x^3+x^2+x+1)$$

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

$$\therefore\ g_1(x)=(x+1)(x^3+x^2+1)=x^4+x^2+x+1$$

$$\therefore\ g_2(x)=(x+1)(x^3+x+1)=x^4+x^3+x^2+1$$

互逆

(生成多项式的逆多项式也是生成多项式)

取 $g(x)=g_1(x)=x^4+x^2+x+1$

$$\therefore\ G(x)=\begin{pmatrix} x^{k-1}\cdot g(x) \\ x^{k-2}\cdot g(x) \\ \vdots \\ x\cdot g(x) \\ g(x) \end{pmatrix}=\begin{pmatrix} x^6+x^4+x^3+x^2 \\ x^5+x^3+x^2+x \\ x^4+x^2+x+1 \end{pmatrix}$$

$$G=\begin{pmatrix} 101\ 1100 \\ 010\ 1110 \\ 001\ 0111 \end{pmatrix}$$

表11-5（2）

$$\therefore\quad A=(a_6\,a_5\,a_4\,a_3\,a_2\,a_1\,a_0)$$

$$=(a_6a_5a_4)G$$

$$=(a_6a_5a_4)\begin{pmatrix}101\;1100\\010\;1110\\001\;0111\end{pmatrix}$$ 非典型

$$\because\quad G=\begin{pmatrix}1011100\\0101110\\0010111\end{pmatrix}=\begin{pmatrix}100\;1011\\010\;1110\\001\;0111\end{pmatrix}$$

$a_6\,a_5\,a_4$	$a_6\,a_5\,a_4\,a_3\,a_2\,a_1\,a_0$
0 0 0	0 0 0 0 0 0 0
0 0 1	1）0 0 1 0 1 1 1
0 1 0	2）0 1 0 1 1 1 0
0 1 1	3）0 1 1 1 0 0 1
1 0 0	5）1 0 1 1 1 0 0
1 0 1	4）1 0 0 1 0 1 1
1 1 0	7）1 1 1 0 0 1 0
1 1 1	6）1 1 0 0 1 0 1

非系统

$$\therefore\quad H=\begin{pmatrix}110\;1000\\011\;0100\\111\;0010\\101\;0001\end{pmatrix}$$

$\therefore\ d_0=3,\ t=1$

监督方程：

$$a_6\oplus a_5\oplus a_3=0$$

$$a_5\oplus a_4\oplus a_2=0$$

$$a_6\oplus a_5\oplus a_4\oplus a_1=0$$

$$a_6\oplus a_4\oplus a_0=0$$

循环码的编、解电路

1、循环码的编码电路

设$m(x)$为信息码多项式，其次数小于k。

(1)用x^{n-k}乘$m(x)$。得到的$x^{n-k}m(x)$的次数必小于n。也就是把信息码后附加上$(n-k)$个“0”，这是监督位的位置。

(2)用$g(x)$除$x^{n-k}m(x)$: $x^{n-k}m(x)/g(x)=Q(x)\cdots\cdots r(x)$

得到余式$r(x)$。

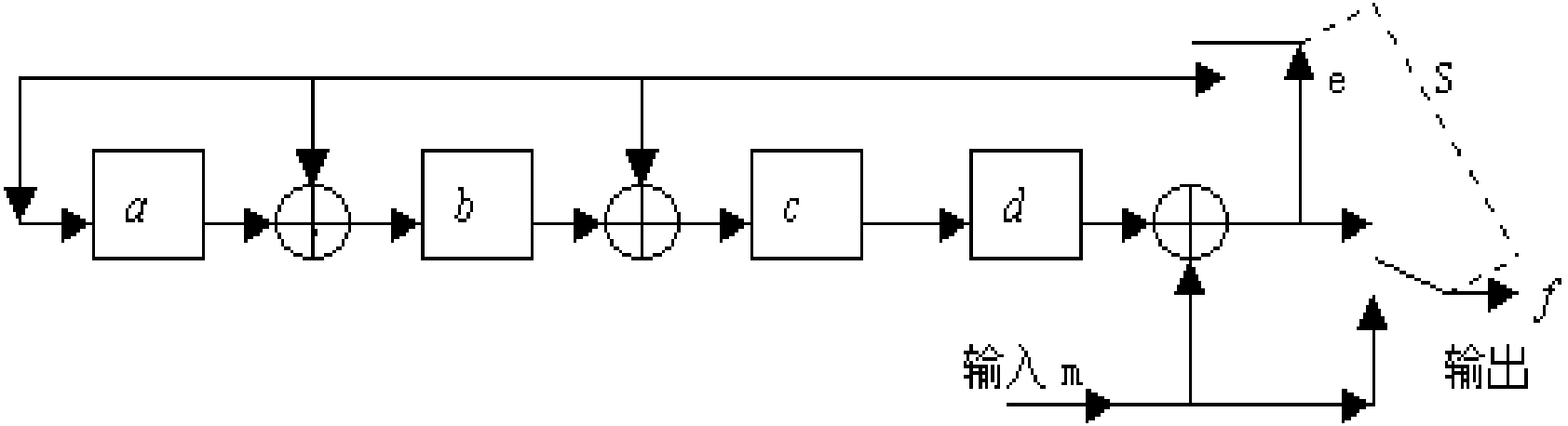
(3) $T(x)=x^{n-k}m(x)+r(x)$

由于$x^{n-k}m(x)+r(x)=Q(x)g(x)$，能被$g(x)$整除，因此$T(x)$必为一码多项式。

余式$r(x)$就是监督码多项式。

上述三步运算，可由除法电路来实现。

（移存器的反馈抽头取决于生成多项式）



时钟脉冲	输入 m	反馈 e	a	b	c	d	输出 f
		$m \oplus d^{-1}$	e	$a^{-1} \oplus e$	$b^{-1} \oplus$ e	c^{-1}	
0	0	0	0	0	0	0	$f=m$
1	1	1	1	1	1	0	
2	1	1	1	0	0	1	
3	0	1	1	0	1	0	
4	0	0	0	1	0	1	$f=d$
5	0	0	0	0	1	0	
6	0	0	0	0	0	1	
7	0	0	0	0	0	0	

时钟脉冲	输入 m	反馈 e	a	b	c	d	输出 f
		$m \oplus d^{-1}$	e	$a^{-1} \oplus e$	$b^{-1} \oplus$ e	c^{-1}	
0	0	0	0	0	0	0	$f=m$
1	1	1	1	1	1	0	
2	1	1	1	0	0	1	
3	0	1	1	0	1	0	
4	0	0	0	1	0	1	$f=d$
5	0	0	0	0	1	0	
6	0	0	0	0	0	1	
7	0	0	0	0	0	0	

2. 循环码的解码电路

接收端解码的要求有两个：检错和纠错。

用于检错的解码电路比较简单。

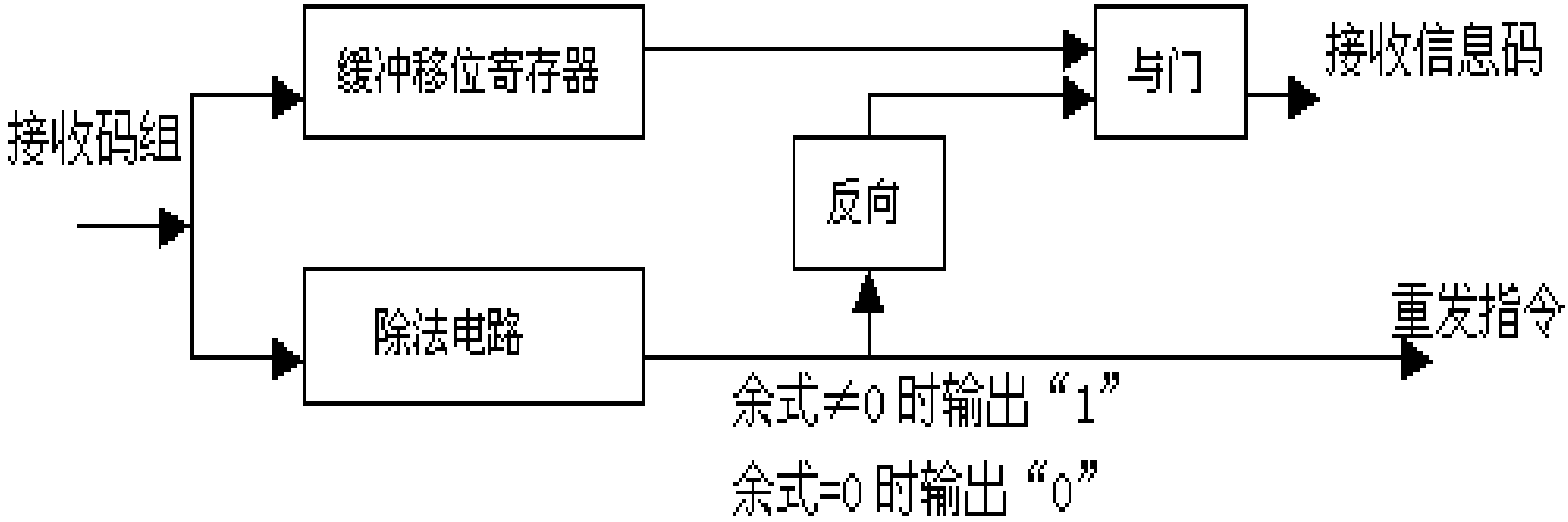
把接收码组$R(x)$除以生成多项式$g(x)$。

当传输中未发生错误时，接收码组与发送码组相同，即$R(x)=T(x)$，故接收码组$R(x)$必定能被$g(x)$整除

若码组在传输中发生错误，则$R(x)\neq T(x)$，$R(x)$被$g(x)$除时可能除不尽而有余项，即有

$$R(x)/g(x)=Q(x)+r(x)/g(x)$$

当错码数超过了这种编码的检错能力时，有错码的接收码组也可能被$g(x)$整除，这时的错码就不能检出了。这种错误称为不可检错误。


接收码组
缓冲移位寄存器
除法电路
与门
反向
接收信息码
重发指令
余式≠0 时输出“1”
余式=0 时输出“0”

用于纠错的解码方法比较复杂。要求每个可纠正的错误图样必须与一个特定余式有一一对应关系。可按下述步骤进行:

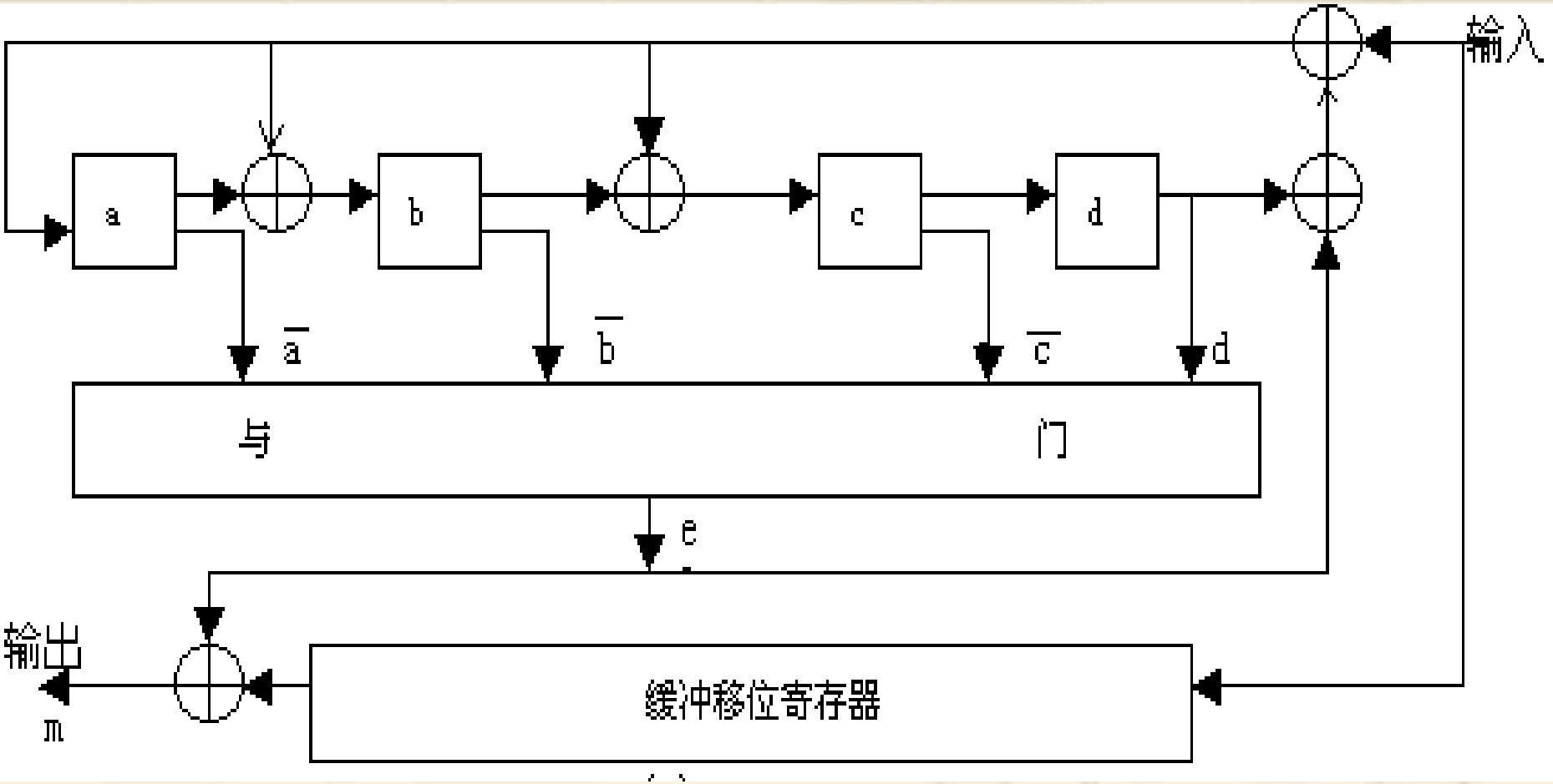
(1)用生成多项式g(*x*)除接收码组*R*(x)=*T*(*x*)+*E*(*x*)，得出余式*r*(*x*);

(2)按余式r(*x*)用查表的方法或通过某种运算得到错误图样E(*x*)。

(3)从R(x)中减去E(*x*)，便得到已纠正错误的原发送码组T(x);

上述运算第(2)步较复杂，并且在计算余式和决定E(*x*)的时候需要把整个接收码组R(*x*)暂时存储起来。

对于纠正突发错误或单个错误的编码还算简单，而对于纠正多个随机错误的编码却是十分复杂的。



这种解码方法称为捕错解码法。一种编码可以有几种不同的纠错解码法。对于循环码来说，可用捕错解码、大数逻辑解码等——现在主要用软件来做编解码

缩短循环码

- 并不是在所有码长(n,k)码中，都能找到相应的满足某纠错能力的循环码。但在系统设计中，码长n、信息位数k和纠错能力常常是预先确定的。
- 这时可采用缩短循环码来满足要求。
- 把一个$(n+i，k+i)$循环码的信息位减少到k位。就得到一位新的(n,k)的线性码，我们称这种码为缩短循环码。由于监督码没有变化，缩短循环码与原循环码至少具有相同的纠错能力；缩短循环码的编码和译码可用原循环码使用的电路完成。
- 在实际中，为了增加检错性能，在原循环码上增加一个偶校验位，得到$(n+1,k+1)$ 码——扩展码。扩展码已不再具有循环性。

8.6 实用循环码

- 8.6.1 循环冗余校验码（**CRC**）
- CRC码是一种缩短循环码，不再具有循环性，但循环码的内在特性依然存在，它的编、译码仍可用原循环码的编、译码电路完成。它的最小码重等于生成多项式的项数。CRC码的信息位长度可变，只要不大于原循环码的信息位长度即可，主要用于检错。被广泛应用于帧校验。
- 国际上常用的CRC码有以下几种：
- (1) CRC-12。生成多项式g(x)= x12+ x11+ x3+ x2+x+1，能检出以下类型的错码：
- ①所有奇数个差错；
- ②所有≤5个的随机差错；
- ⑧所有长度≤12的单串突发差错；
- ④以1-2-13的概率检出长度为17的单串突发差错；
- ⑤以1-2-12的概率检出长度大于17的单串突发差错；
- ⑥所有长度≤2的两串突发差错。

8.6 实用循环码

- (2) CRC-ITU-T。生成多项式g(x)=x16+ x12+ x5+1，用于HDLC、SDLC、X.25、7号信令、ISDN等处。能检出以下类型的错码：
- ①所有奇数个差错；
- ②所有≤3个的随机差错；
- ⑧所有长度≤16的单串突发差错；
- ④以1-2-17的概率检出长度为17的单串突发差错；
- ⑤以1-2-16的概率检出长度大于17的单串突发差错；
- ⑥所有长度各≤2的两个突发差错。
- (3)CRC-16。生成多项式g(x)=x16+ x15+ x2+1，用于美国二进制同步系统。检错能力同CRC-ITU-T。

8.6 实用循环码

- (4)CRC-32。生成多项式g(x)= x32+ x26+ x23+ x22+ x16+x12+ x11+ x10+ x8+ x7+ x5+ x4+ x2+x+1，用于以太网及ATM AAL-5适配层。能检出以下类型的错码：
- ①所有奇数个差错；
- ②所有≤14个的随机差错；
- ⑧所有长度≤32的单个突发差错；
- ④以1-2-33的概率检出长度为33的单个突发差错；
- ⑤以1-2-32的概率检出长度大于33的单个突发差错；
- ⑥所有长度各≤2的两个突发差错。
- (5) CRC-IS-95 CDMA。生成多项式g(x)= x30+x29+x21+x20+x15+x13+ x12+x11+x8+x7+x6+x2+ x+1 。
- 在一些UART通信控制芯片中都集成有CRC码编解码电路，如INTEL8273、MC6854及Z80-SIO　等。

8.6.2 BCH码

- 在已提出的许多纠正随机错误的码中，BCH码是至今用得最广泛和很有效的一种码，BCH码是以发明这种码的三个人的名字来命名的。BCH码是一类纠正多个随机错误的循环码。
- BCH码分两类，即本原BCH和非本原BCH码。本原BCH码的码长为$n=2^m-1$，(m是≥3的任意正整数)，它的生成多项式g(x)中含有最高次数为m次的本原多项式；
- 非本原BCH码的码长n是2^m-1的一个因子，它的生成多项式g(x)中不含有最高次数为m的本原多项式。

- 能纠正$t<m/2$错码的BCH码，其码长为$n=2^m-1$，监督位$n-k\leq mt$。
- 若码长$n=(2^m-1)/i[i>1$，且除得尽$(2^m-1)]$，则为非本原码。
- 具有循环性的汉明码就是能纠正单个错码的本原BCH码
- 表11-7中的(23,12)码称为戈莱(Golay)码，它是一个纠正三个随机错误的码，且容易解码，实际中使用的比较多。

- 8.6.3 里德-索洛蒙码（**RS**码）
- RS码是一种多进制的BCH码，每个符号由m个比特组成。一个能纠正t个错码的RS码码长为n=2m-1，监督位码长2t。特别适于纠正突发性错码，可纠正的错误图样有：
- 总长度b1=(t-1)m+1的单个突发错码
- 总长度b2=(t-3)m+3的两个突发错码
- ……
- 总长度bi=(t-2i+1)m+2i-1的i 个突发错码
- RS码适用于衰落信道及计算机的存储系统。它的译码方法与BCH码类似，也有彼得森译码和迭代译码两种。

- 8.6.4 法尔码（Fire码）
- Fire码是可纠正单个突发错码的一类循环码。
- 令p(x)是一个m阶的既约多项式，*l*与m互素，则Fire码的生成多项式为
- $$g(x)=p(x)+(xi+1) \qquad (7.6\text{-}1)$$
- 该码码长 $$n=LCM(l，e) \qquad (7.6\text{-}2)$$
- 其中e=2m-1，该码的监督码长
- $$r=l+m \qquad (7.6\text{-}3)$$
- Fire码的纠错能力为
- 当$l \geq bt+be-1$，$m \geq bt$时，能纠正长度≤bt的单个突发错码，并能发现长度≥bt而≤be的突发错码；
- 若用于检错，能发现长度≤$l+m$的单个突发错码，或两个突发错码的组合，两个突发错码长度之和≤$l+1$，其中一个长度≤be。

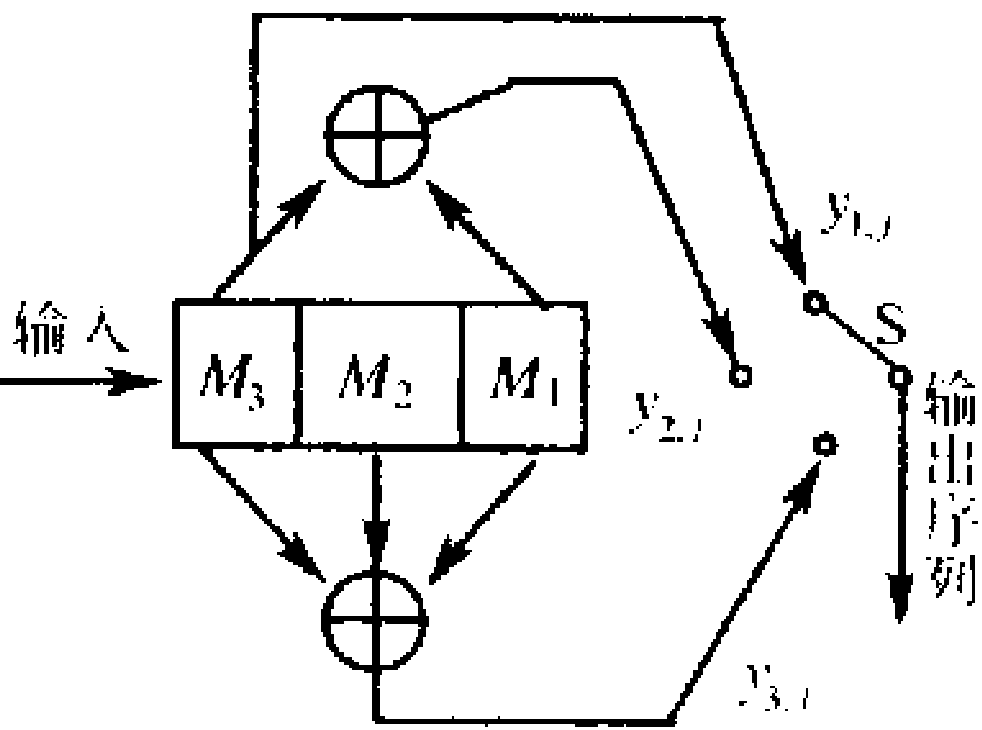
8.7 卷积码

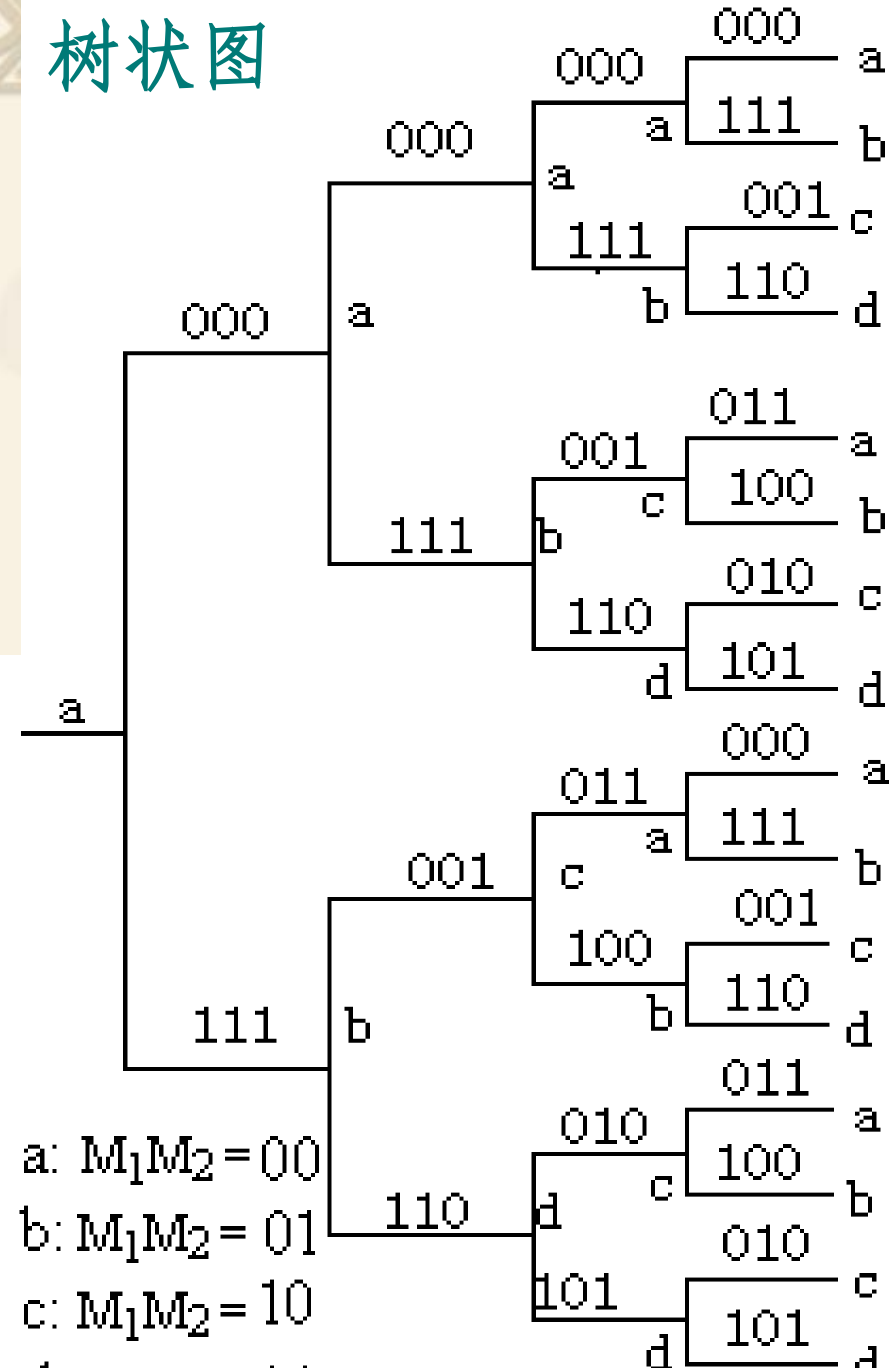
- 卷积码是伊莱亚斯于1955年提出的一种非分组码。与线性分组码相比存在着许多差别，大体表现在以下几个方面：
- (1)线性分组码的编码是将信息序列明确地分组，每个码组中校验码仅仅与本码组中的信息码有关，编码后形成固定长度、互不相关的码组序列，这种编码无记忆性。卷积码每个码组中的监督码不但与本码组的信息码有关，还与前边(N-1)个码组中的信息码有关。设一个码组的码长为n，n×N称为约束长度，N称为约束度。卷积码的纠错能力也随N的增大而增强，卷积码是具有记忆性。一般用(n，k，N-1) 表示卷积码。
- (2)为了兼顾纠错能力与编码效率，线性分组码的码组长度n一般都较大。随着n增大，编、译码电路复杂度迅速增加，并带来较大的译码延时。卷积码则将信息码与校验码之间的相关性分布在N个码组之间。这样卷积码的k和n值可以为比较小的值，编、译码延时小，特别适合以串行方式传输信息的应用场合。**更适用于前向纠错。**因此在相同的传信率和设备复杂度的条件下，卷积码的性能一般优于线性分组码。

8.7 卷积码

- (3)线性分组码多采用系统码，而卷积码则不然。当N值确定后，非系统卷积码可获得更大的自由距，更易达到最佳编码效果。对卷积码的译码而言，系统码和非系统码的译码难度是一样的，故卷积码常采用非系统码。
- (4)线性分组码有严格的代数结构，而卷积码的纠错能力与编码结构之间缺乏明确的数学关系。在构造许用的卷积码(也称为好码)时，只能是依码距性能，利用计算机对大量的码进行搜索得到的。
- (5)线性分组码的编码器可视为一个有k个输入变量、n个输出变量的线性网络。卷积码可视为输入信息序列与编码器的特定结构所决定的另一个序列的卷积，卷积码也就由此得名。

树状图





a: $M_1M_2 = 00$

b: $M_1M_2 = 01$

c: $M_1M_2 = 10$

d: $M_1M_2 = 11$

a状态

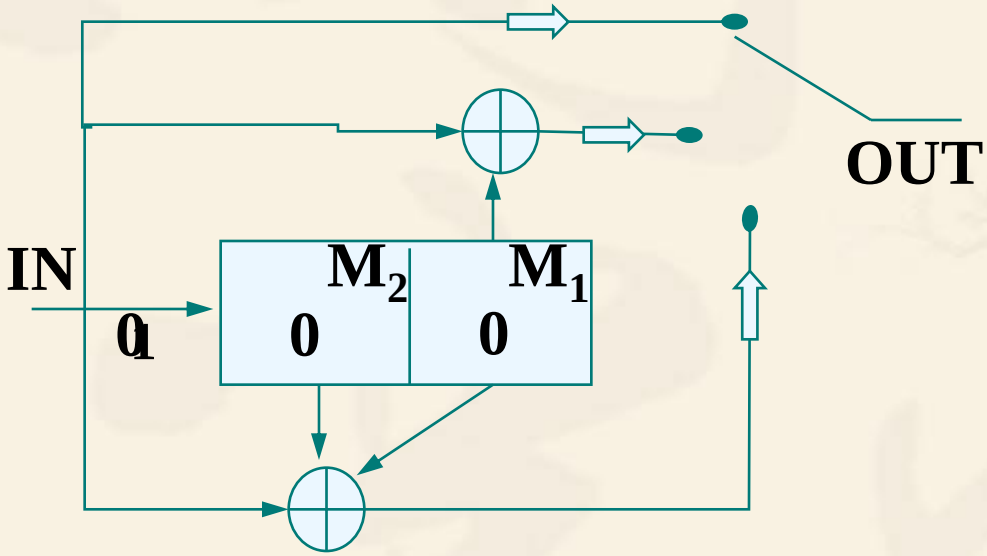
a: $M_1M_2=00$

b: $M_1M_2=01$

c: $M_1M_2=10$

d: $M_1M_2=11$

输入0　　移位 → a状态



复位

输入1　　移位 → b状态

b状态

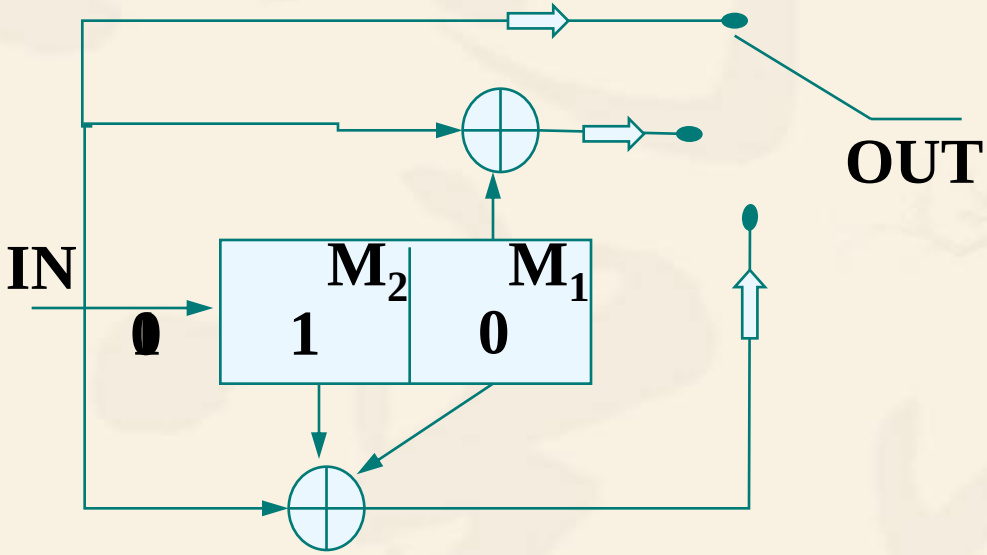
a: $M_1M_2=00$

b: $M_1M_2=01$

c: $M_1M_2=10$

d: $M_1M_2=11$

输入0



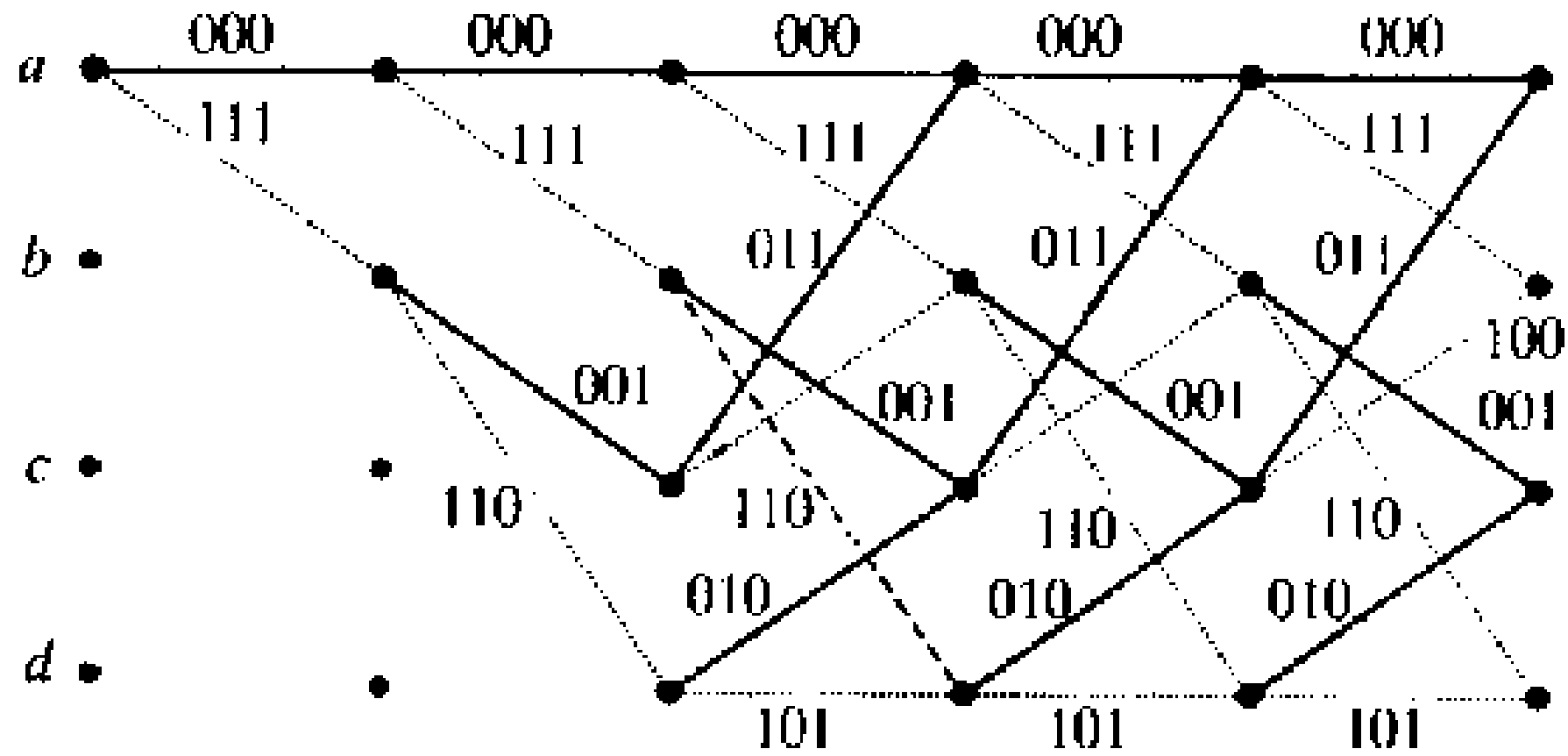
输入1

c状态

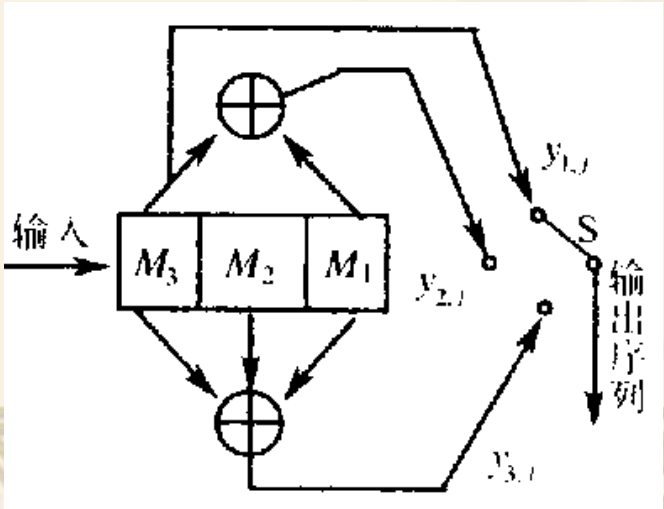
移位

d状态

复位



网格图



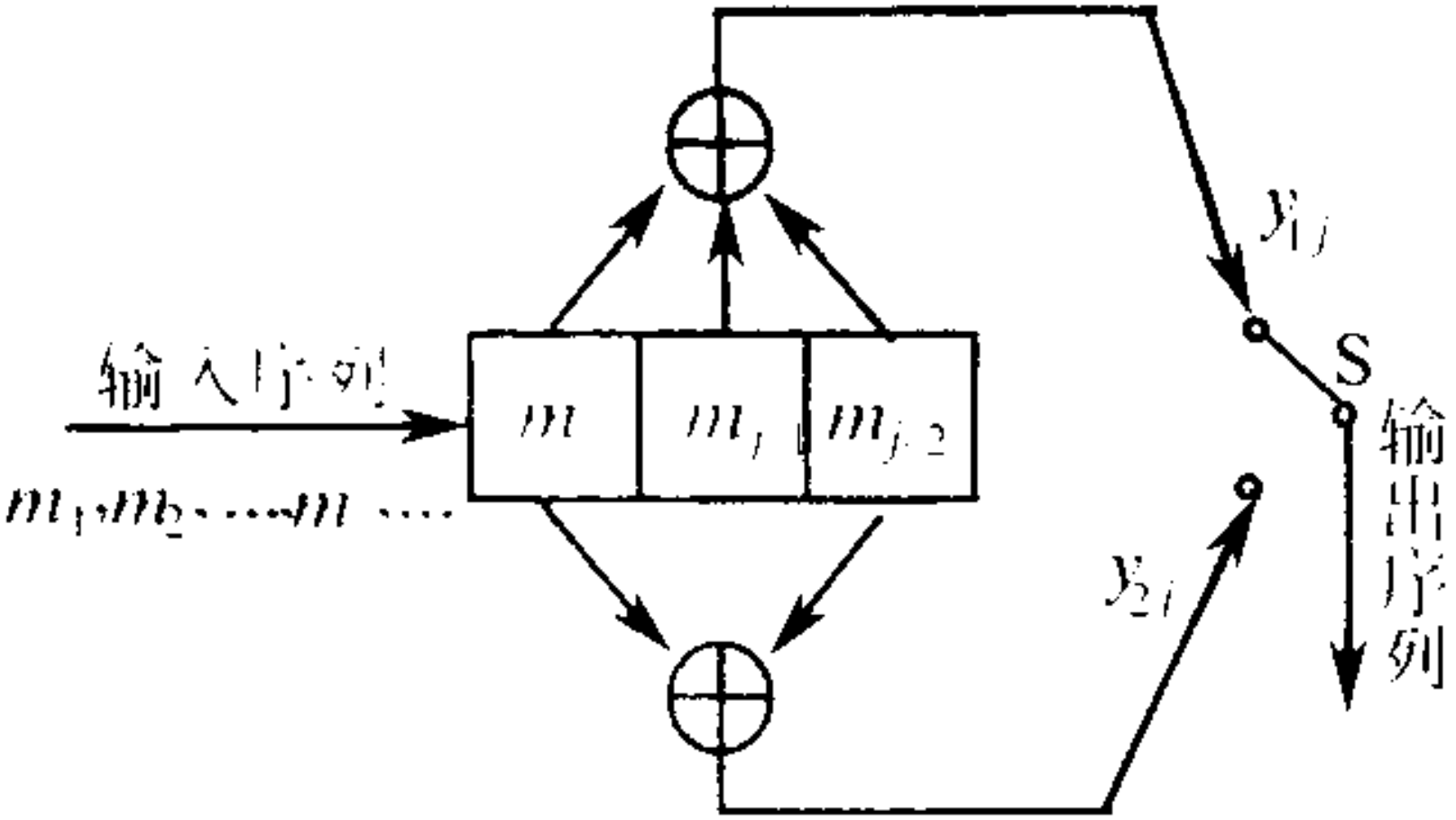
a: $M_1M_2=00$

b: $M_1M_2=01$

c: $M_1M_2=10$

d: $M_1M_2=11$

非系统码



- 8.7.3卷积码的距离特性
- 卷积码码距的概念与分组码不同，有最小距离*dmin*和自由距离*dfree*两种，从网格图上能得到很好的表述。最小距离*dmin*定义为由零状态零时刻分叉、长度为*nN*的两个编码序列间的最小距离。也就是在零状态零时刻输入非零信息码、长度为*nN*的编码序列的最小码重，如图7-15中路径abcb所对应的编码序列111001100的码重*w*=5就是（3，1，2）码的最小距离。自由距离*dfree*定义为由零状态零时刻分叉、任意长的两个编码序列间的最小距离。也就是在零状态零时刻输入非零信息码、然后又回到零状态的所有编码序列中的最小码重。仍以图7-15为例，路径abca所对应的编码序列111001011的码重w=6就是（3，1，2）码的自由距离。一般来说，*dmin*≤*dfree*。采用哪种码距来度量纠错能力，与译码方法有关。采用门限译码时，就以最小距离来度量；采用维特比译码和序列译码时，就以自由距离来度量。
- 目前卷积码的许用码组（好码）都是由计算机根据距离特性搜索得到的，表7-11中列出了部分具有最大自由距离的非系统卷积码。

8.7.4 卷积码译码

卷积码的译码方法有两类：一类是建立在代数译码基础上的门限译码，又称大数逻辑译码；另一类是最大似然译码，又称概率译码，概率译码又分为维特比译码和序列译码两种。

1. 维特比译码

在离散无记忆信道中，输入一个二进制符号序列**X**，而输出**Y**则是具有**J**种符号的序列。***x***序列每发一个符号***xi***，则信道输出端收到一个相应的符号***yj*(j=1,2,3,…,J)**。由于是无记忆，故***yj***只与***xi***有关。如果**J=2**，则离散无记忆信道输出是二进制序列。该信道称为硬量化(硬判决)信道。如果**J≥2**，即信道输出符号数大于**2**，则称为软量化(软判决)信道。已经证明，对高斯白噪声来说，**3**比特软量化(即**J＝8)**与硬量化相比可获得**2dB**的编码增益。

维特比译码算法简称VB算法，是1967年由Viterbi提出。是最大似然译码的一种。最大似然译码的基本思路是：把已接收序列与所有可能的发送序列做比较，选择其中码距最小的一个序列作为发送序列。如果发送L组信息比特，对于(n,k)卷积码来说，可能发送的序列组合有2kL个，需要存储所有这些序列并进行比较，以找到码距最小的那个序列。当传信率和信息组数L较大时，译码器将变得非常复杂。VB算法则对上述的思路做了简化，成为了一种实用化的概率算法。它并不是在网格图上一次比较所有可能的2kL条路径(序列)，而是接收一段，计算和比较一段，选择一段有最大似然可能的码段，从而达到整个码序列是一个有最大似然值的序列。

下面将用图7-14的(2,1,2)卷积码编码器所编出的码为例，来说明维特比译码硬判决的运算过程。该码网格图同图7-12，只是路径上的输出码组不同。设编码器初始状态为a状态。网格图的每一条路径都对应着不同的输入信息序列，而所有的可能输入信息序列共有2^{kL}个，因此网格图中所有可能路径也有2^{kL}条。

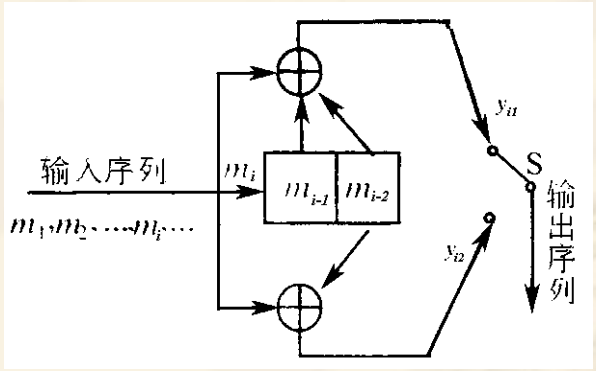
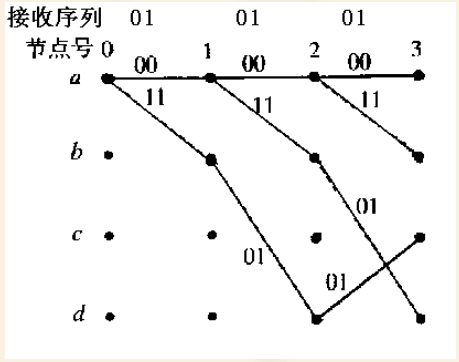


图7-14 (2,1,2)卷积码编码器

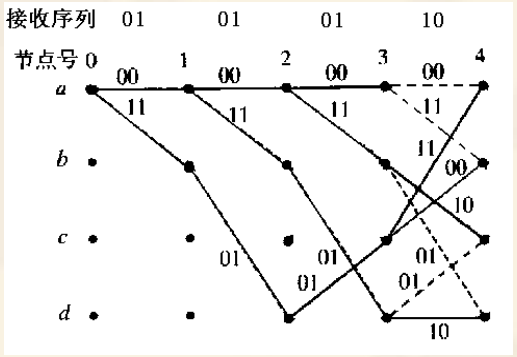
输入编码器的信息序列为(11011000)，则由编码器输出的序列Y=(1101010001011100)，编码器的状态转移路线为abdcbdca。若收到的序列R=(0101011001011100)，对照网格图来说明维特比译码的方法。

由于该卷积码的约束长度为6位，因此先选择接收序列的前6位R_1=(010101)同到达第3时刻的可能的8个码序列(即8条路径)进行比较，并计算出码距。该例中第3时刻到达a点的路径序列是(000000)和(111011)，它们与R_1的距离分别是3和4；到达b点的路径序列是(000011)和(111000)，它们与R_1的距离分别是3和4；到达c点的路径序列是(001110)和(110101)，与R_1的距离分别是4和1；到达d点的路径序列是(001101)和(110110)，与R_1的距离分别是2和3。上述每个节点都保留码距较小的路径作为幸存路径，幸存路径码序列分别是(000000)、(000011)、(110101)和(001101)，如图7-15(a)所示。

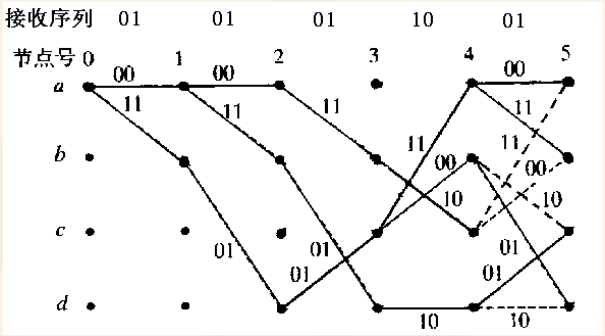
00
01
10
11



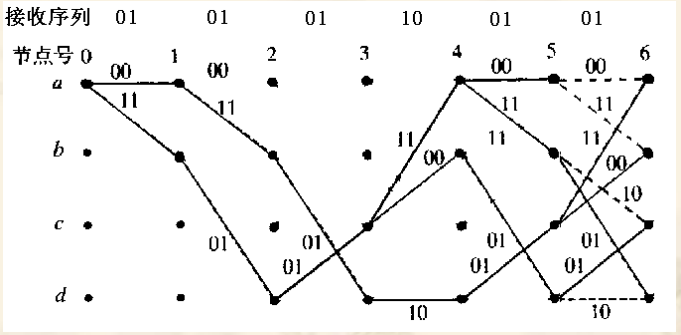
(a)第3时刻幸存路径；



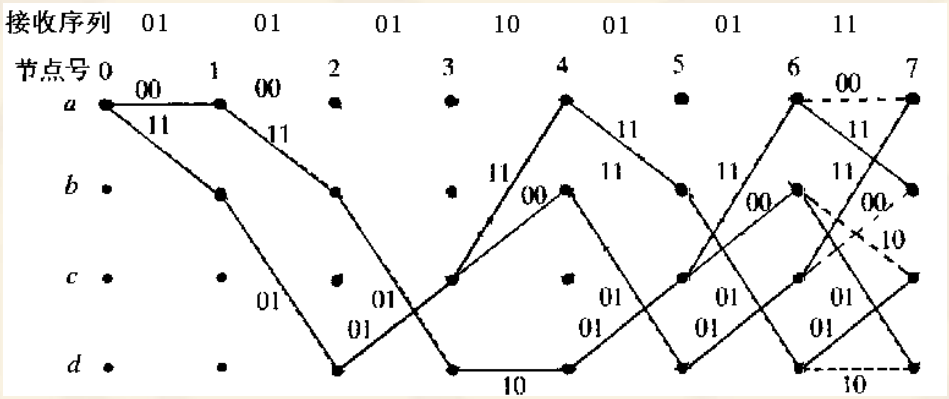
(b)第4时刻幸存路径；



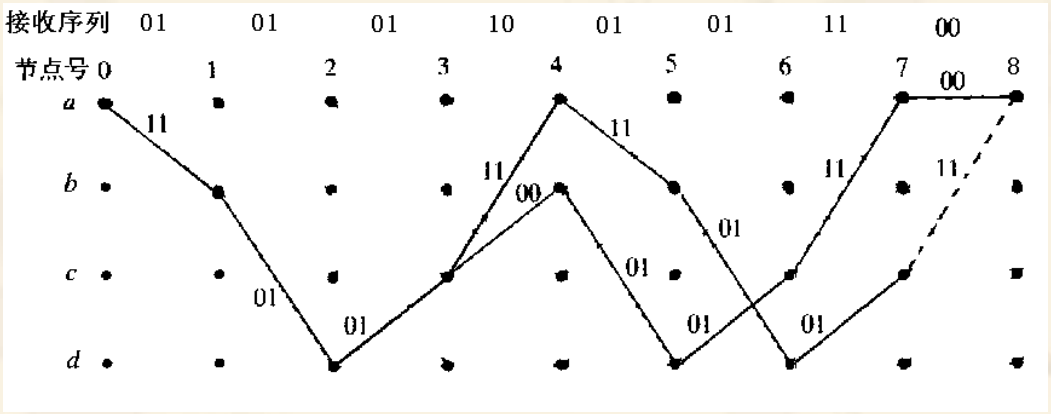
(c)第5时刻幸存路径



(d)第6时刻幸存路径



(e)第7时刻幸存路径



(f) 第8时刻幸存路径

- 用与上面同样的方法可以得到第4时刻的幸存路径。选择接收序列的前8位R2=(01010110)同到达第4时刻的可能的8个码序列(即8条路径)进行比较。到达a点的路径序列是(00000000)和(11010111)，它们与R2的距离分别是4和2；到达第3时刻b点的路径序列是(00000011)和(11010100)，它们与R2的距离分别是4和2；到达c点的路径序列是(00001110)和(00110101)，与R2的距离分别是3和4；到达d点的路径序列是(00001101)和(00110110)，与R2的距离分别是5和2。上述每个节点都保留码距较小的路径作为幸存路径，幸存路径码序列分别是(11010111)、(11010100)、(00001110)和(00110110)，如图7-18(b)所示。

- 如果到达某一个节点的两条路径与接收序列的码距相等，则可任选一路径作为幸存路径，此时不会影响最终的译码结果。当信息码传输结束时，编码器一定会回到a状态，所以最后在a状态得到一条幸存路径即可，如图7-15(f)所示。由此看到译码器输出是R’=(1101010001011100)=Y，说明在译码过程中已纠正了在码序列第1和第7位上的差错。当然，差错出现太频繁，超出卷积码的纠错能力时，则会发生误纠。
- 通过上面的分析可以看出，随着信息码的增加，译码时要比较的码序列的长度（称为译码深度）也不断增长，这将使译码器变得非常复杂。当然译码深度也不可能随着信息码的增加而不断增大。实践证明，当硬判决时，译码深度（又称译码约束长度）取编码约束长度的3~5倍；软判决时，量化比特取3；编码增益已接近极限。

- 8.8 网格编码调制
- 在传统的数字传输系统中，编解码与调制解调是各自独立设计和实现的。在20世纪70年代中期，梅西（Messey）根据信息论，证明了将编码与调制作为一整体考虑的最佳设计，可以明显地改善系统性能。1982年，昂格尔博克(Ungerbook)提出了卷积码与调制相结合的网格编码调制(即TCM)。
- TCM码的典型结构如图7-18所示。第一部分是差分编码，它与第三部分的合理结合可以避免解调时信号的倒π现象，和DPSK的原理相同。第二部分是卷积编码器，将m 位信码中的k位编成(k+1)位卷积码。第三部分叫做分集映射(mapping by set partitioning)，其任务是将一个(m+1)bit的码组对应为一个调制符号输出。(m+1)bit组有2m+1种可能的组合，调制后也必须有2m+1个信号。

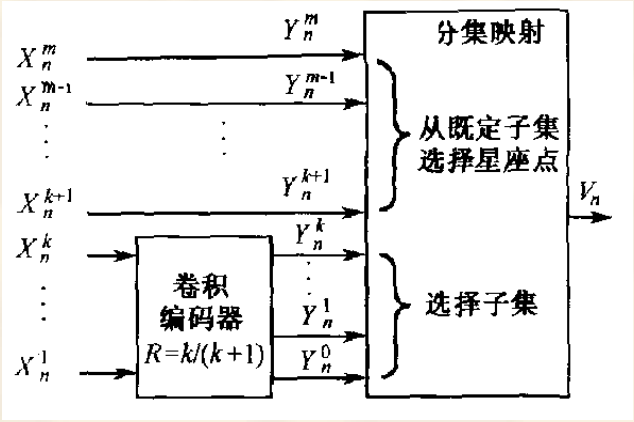


图7-18 TCM编码器的一般结构

7.8.2 TCM码设计原则

为了得到TCM好码，分集映射必须遵守如下规律:

(1)从某一状态发出的子集源于同一个上级子集，如从a状态发出的码字有000，100，010，110分属两个子集C0和C1，它们源于同一个上级子集B0。

(2)到达某一状态的子集源于同一个上级子集。

(3)各子集在编码矩阵中出现的次数相等，并呈现一定的对称性。

真正好的实用TCM编码都是由计算机搜索得到的。表7-14和表7-15列出了部分由计算机搜索得到的TCM好码。表中的生成多项式的系数用8进制数表示，以4状态8PSK TCM编码为例，它的生成多项式的系数h_2不存在，说明X_n^2不参与卷积编码，$Y_n^2=X_n^2$；$h_1=(5)_8=(101)_2$，$h_0=(2)_8=(010)_2$，其生成多项式分别为

$$g_1(x)= 1+x^2,$$
$$g_0(x)= x,$$

则卷积码编码输出为

$$Y_n^1= X_n^1(1+x^2)$$
$$Y_n^0= X_n^1 \cdot x= X_{n-1}^1$$

表7-14 PSK TCM好码

状态数	编码位数k	生成多项式系数			编码增益(dB) $G_{8PSK/4PSK}$	编码位数k	生成多项式系数			编码增益(dB) $G_{16PSK/8PSK}$
		h_2	h_1	h_0			h_2	h_1	h_0	
4	1	/	5	2	3.01	1	/	2	5	3.54
8	2	04	02	11	3.60	1	/	04	13	4.01
16	2	16	04	23	4.13	1	/	04	23	4.44
32	2	34	16	45	4.59	1	/	10	45	5.13
64	2	066	030	103	5.01	1	/	024	103	5.33
128	2	122	054	277	5.17	1	/	024	203	5.33
256	2	130	072	435	5.75	2	374	176	427	5.51

表7-15 QAM TCM好码

状态数	编码位数k	生成多项式系数			编码增益(dB) $G_{16QAM/8PSK}$
		h_2	h_1	h_0	
4	1	/	2	5	3.01
8	2	04	02	11	3.60
16	2	16	04	23	4.13
32	2	10	06	41	4.59
64	2	064	016	101	5.01
128	2	042	014	203	5.17
256	2	304	056	401	5.75