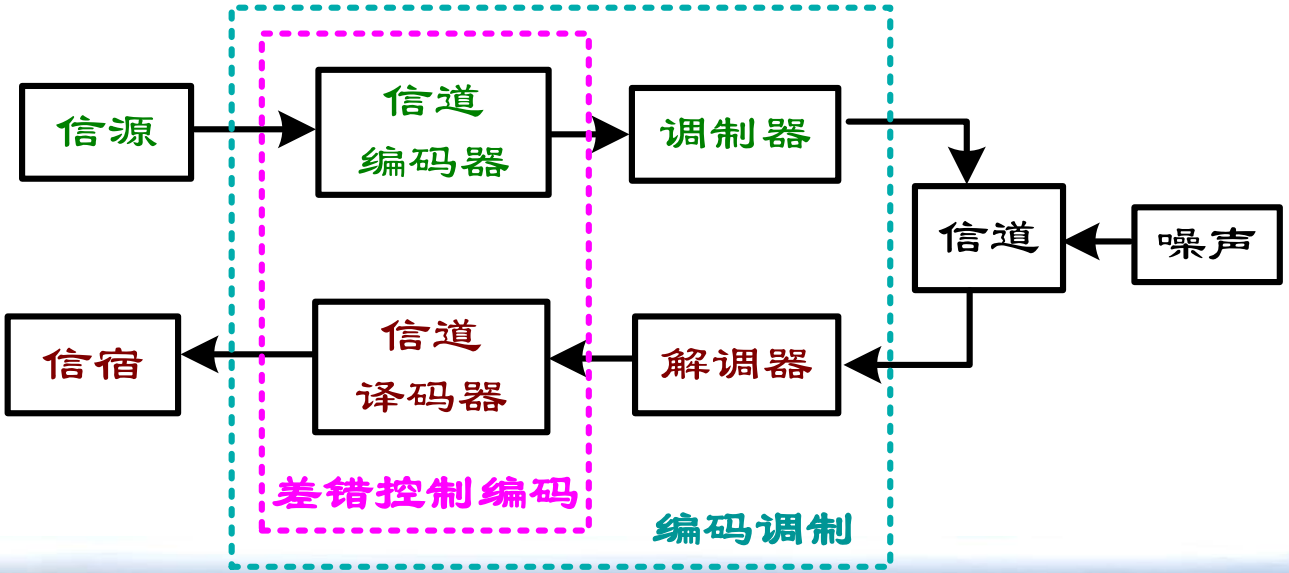
第7章 信道编码

本章内容

- 绪论
- 线性分组码
- 循环码
- 卷积码
- **Turbo码**

7.1 引言

- 在设计数字通信系统时，首先应从合理地选择调制解调方法、合适的发射功率等方面考虑，若仍不能满足系统误码率要求，则要考虑采用本章所讲的差错控制编码措施。
- 纠错码，是当消息经过有噪信道传输或要恢复存储的数据时用来纠错的。用来传输消息的物理介质叫做信道（如电话线、卫星连接、用于移动通信的无线信道等）。因为纠错码试图克服信道中噪声造成的损害，因此其编码过程又称为信道编码，它是提高数字信号传输可靠性的有效方法之一。



- 信道编码可分为两个研究领域：波形编码（信号设计）和结构化序列（结构冗余）。波形编码是将波形转变为“更好的波形”以减少错误判决；结构化序列是将数据序列转变为“更好的序列”，通过冗余比特来检测和纠正传输中的错误。
- 波形编码：最常见的是正交编码和双正交编码，通过编码使编码集合中信号间的相关度最小（即信号间的距离最大）。如对极信号、正交信号。正交编码常使用Hadamard矩阵，即1bit数据集可用两个数字的正交码字进行变换，2bit数据集可使用4个数字的正交码字进行变换，依此类推。

数据集	正交码字集
1	$H_1=\begin{pmatrix}0&0\\0&1\end{pmatrix}$
0	

$$H_k=\begin{pmatrix}H_{k-1} & H_{k-1}\\ H_{k-1} & \overline{H}_{k-1}\end{pmatrix}$$

数据集	正交码字集
0 0	$H_2=\left(\begin{array}{cc:cc}0&0&0&0\\0&1&0&1\\ \hdashline 0&0&1&1\\0&1&1&0\end{array}\right)=\begin{pmatrix}H_1 & H_1\\ H_1 & \overline{H}_1\end{pmatrix}$
0 1	
1 0	
1 1	

- 双正交码：具有M个信号的双正交信号集可由M/2个信号的正交信号集获取。

$$B_M = \begin{pmatrix} H_{M-1} \\ \overline{H}_{M-1} \end{pmatrix}$$

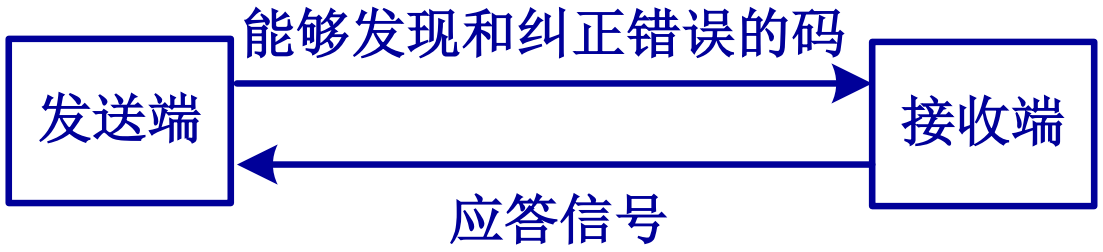
- 例如，一个3bit数据集可以转变为如下的双正交码字集

数据集	正交码字集
0 0 0	$B_3 =$ 0 0 0 0
0 1 1	0 1 0 1
0 1 0	0 0 1 1
0 1 1	0 1 1 0
	.....
1 0 0	1 1 1 1
1 0 1	1 0 1 0
1 1 0	1 1 0 0
1 1 1	1 0 0 1

- 常用的差错控制方式有三种：
  - 检错重发方式，又称为自动重发请求（ARQ），发送端发送能够发现错误的码，由接收端判断接收中有无错误发生。如果发现错误，则通过反向信道把这一判决结果反馈给发送端，然后发送端再把错误的信息重发一次。

  - 前向纠错方式（FEC）：发送端发送能够纠正错误的码，接收端收到后自动纠正传输中的错误，特点是单向传输。

- 混合纠错方式（HEC）：发送端发送既能自动纠错，又能检测的码。接收端收到码流后，检查差错情况，如果错误在纠错能力范围以内，则自动纠错，如果超过了纠错能力，但能检测出来，则经过反馈信道请求发送端重发。



7.1.1 差错控制编码理论的起源和发展

- 1948年，Bell实验室的C.E.Shannon发表的《通信的数学理论》，是关于现代信息理论的奠基性论文，它的发表标志着信息与编码理论这一学科的创立。Shannon在该文中指出，任何一个通信信道都有确定的信道容量C，如果通信系统所要求的传输速率R小于C，则存在一种编码方法，当码长n充分大并应用最大似然译码（MLD，Maximum Likelihood Decoding）时，信息的错误概率可以达到任意小。从Shannon信道编码定理可知，随着分组码的码长n或卷积码的约束长度N的增加，系统可以取得更好的性能（即更大的保护能力或编码增益），而译码的最优算法是MLD，MLD算法的复杂性随n或N的增加呈指数增加，因此当n或N较大时，MLD在物理上是不可实现的。因此，构造物理可实现编码方案及寻找有效译码算法一直是信道编码理论与技术研究的中心任务。
- Shannon指出了可以通过差错控制码在信息传输速率不大于信道容量的前提下实现可靠通信，但却没有给出具体实现差错控制编码的方法。

- 20世纪40年代，R.Hamming和M.Golay提出了第一个实用的差错控制编码方案，使编码理论这个应用数学分支的发展得到了极大的推动。通常认为是R.Hamming提出了第一个差错控制码。当时他作为一个数学家受雇于贝尔实验室，主要从事弹性理论的研究。他发现计算机经常在计算过程中出现错误，而一旦有错误发生，程序就会停止运行。这个问题促使他编制了使计算机具有检测错误能力的程序，通过对输入数据编码，使计算机能够纠正这些错误并继续运行。Hamming所采用的方法就是将输入数据每4个比特分为一组，然后通过计算这些信息比特的线性组合来得到3个校验比特，然后将得到的7个比特送入计算机。计算机按照一定的原则读取这些码字，通过采用一定的算法，不仅能够检测到是否有错误发生，同时还可以找到发生单个比特错误的比特的位置，该码可以纠正7个比特中所发生的单个比特错误。这个编码方法就是分组码的基本思想，Hamming提出的编码方案后来被命名为汉明码。

- 虽然汉明码的思想是比较先进的，但是它也存在许多难以接受的缺点。首先，汉明码的编码效率比较低，它每4个比特编码就需要3个比特的冗余校验比特。另外，在一个码组中只能纠正单个的比特错误。M.Golay研究了汉明码的这些缺点，并提出了两个以他自己的名字命名的高性能码字：一个是二元Golay码，在这个码字中Golay将信息比特每12个分为一组，编码生成11个冗余校验比特。相应的译码算法可以纠正3个错误。另外一个是三元Golay码，它的操作对象是三元而非二元数字。三元Golay码将每6个三元符号分为一组，编码生成5个冗余校验三元符号。这样由11个三元符号组成的三元Golay码码字可以纠正2个错误。

- 汉明码和Golay码的基本原理相同。它们都是将q元符号按每k个分为一组．然后通过编码得到n-k个q元符号作为冗余校验符号，最后由校验符号和信息符号组成有n个q元符号的码字符号。得到的码字可以纠正t个错误，编码码率为为k/n。这种类型的码字称为分组码，一般记为(q,n,k,t)码，二元分组码可以简记为(n,k,t)码或者(n,k)码。汉明码和Golay码都是线性的，任何两个码字经过模q的加操作之后，得到的码字仍旧是码集合中的一个码字。

- 在Golay码提出之后最主要的一类分组码就是Reed-Muller码。它是Muller在1954年提出的，此后Reed在Muller提出的分组码的基础上得到了一种新的分组码，称为Reed-Muller码，简记为RM码。在1969年到1977年之间，RM码在火星探测方面得到了极为广泛的应用。即使在今天，RM码也具有很大的研究价值，其快速的译码算法非常适合于光纤通信系统。

- 在RM码提出之后人们又提出了循环码的概念。循环码实际上也是一类分组码，但它的码字具有循环移位特性，即码字比特经过循环移位后仍然是码字集合中的码字。这种循环结构使码字的设计范围大大增加，同时大大简化了编译码结构。循环码的另一个特点就是它可以用一个幂次为n-k的多项式来表示，这个多项式记为g(D)，称为生成多项式，其中D为延迟算子。循环码也称为循环冗余校验(CRC，Cyclic Redundancy Check)码，并且可以用Meggitt译码器来实现译码。由于Meggitt译码器的译码复杂性随着纠错能力t的增加而呈指数形式的增加，因此通常CRC码用于纠正只有单个错误的应用情况，常用做检错码而非纠错码。
- 循环码的一个非常重要的子集就是分别由Hocquenghem在1959年、Bose和Ray-Chaudhuri研究组在1960年几乎同时提出的BCH码(BCH，Bose Chaudhuri Hocquenghem)，BCH码的码字长度为$n=q^m-1$，其中m为一个整数。二元BCH码（q=2）的纠错能力限为$t<(2^m-1)/2$。1960年，Reed和Solomon将BCH码扩展到非二元（q>2）的情况，得到了RS（Reed-Solomon）码。1967年，Berlekamp给出了一个非常有效的译码算法后，RS码得到了广泛的应用。此后，RS码在CD播放器、DVD播放器中得到了很好的应用。

- 虽然分组码在理论分析和数学描述方面已经非常成熟，并且在实际的通信系统中也已经得到了广泛的应用，但分组码固有的缺陷大大限制了它的进一步发展。首先，由于分组码是面向数据块的，因此，在译码过程中必须等待整个码字全部接收到之后才能开始进行译码。在数据块长度较大时，引入的系统延时是非常大的。分组码的第二个缺陷是它要求精确的帧同步，即需要对接收码字或帧的起始符号时间和相位精确同步。另外，大多数基于代数的分组码的译码算法都是硬判决算法，而不是对解调器输出未量化信息的软译码，从而造成了一定程度的增益损失。

- 分组码所存在的固有缺点可以通过采用其他的编码方法来改善，这种编码方法就是卷积码，是Elias等人在1955年提出的。卷积码与分组码的不同在于分组码在编码之前先将信息序列按照一定的数据块长度分组，然后对每一组信息进行独立编码，即对于(n,k)分组码来说，码字中的n-k个检验元仅与本码字的k个信息元有关，而与其他码字的信息元无关。同样，在分组码的译码时也是针对每一个接收码字进行独立译码的，在一个接收码字内部提取译码相关信息，与其他码字无关。而在卷积编码中则充分利用了各个信息块之间的相关性。通常卷积码记为(n，k，N)码。卷积码的编码过程是连续进行的，依次连续将每k个信息元输入编码器，得到n个码元，得到的码元中的检验元不仅与本码的信息元有关，还与以前时刻输入到编码器的信息元(反映在编码寄存器的内容上)有关。同样，在卷积码的译码过程中，不仅要从本码中提取译码信息，还要充分利用以前和以后时刻收到的码组．从这些码组中提取译码相关信息，而且译码也是可以连续进行的，这样可以保证卷积码的译码延时相对比较小。通常，在系统条件相同的条件下，在达到相同译码性能时，卷积码的信息块长度和码字长度都要比分组码的信息块长度和码字长度小，相应译码复杂性也小一些。

- 卷积码的译码通常有如下几个比较流行的译码算法：
  - 由Wozencraft和Reiffen在1961年提出、Fano和Jelinek分别在1963年和1969年进行改进了的序贯译码算法。该算法是基于码字树图结构的一种次最优概率译码算法。
  - 由Massey在1963年提出的门限译码算法。这个算法利用码字的代数结构进行代数译码。
  - 由Viterbi在1967年提出的Viterbi算法。该算法是基于码字格图结构的一种最大似然译码算法，是一种最优译码算法。
- 在Viterbi译码算法提出之后，卷积码在通信系统中得到了极为广泛的应用。如GSM、3G、商业卫星通信系统等。

- 在信道编码定理的指引下，人们一直致力于寻找能满足现代通信业务要求，结构简单、性能优越的好码，并在分组码、卷积码等基本编码方法和最大似然译码算法的基础上提出了许多构造好码及简化译码复杂性的方法，提出了乘积码、代数几何码、低密度校验码(LDPC，Low Density Parity Check)、分组-卷积级联码等编码方法和逐组最佳译码、软判决译码等译码方法以及编码与调制相结合的网格编码调制(TCM，Trellis Coded Modulation)技术。其中对纠错码发展贡献比较大的有级联码、软判决译码和TCM技术等。

- 虽然软判决译码、级联码和编码调制技术都对信道码的设计和发展产生了重大影响，但是其增益与Shannon理论极限始终都存在2～3dB的差距。
- 在1993年于瑞士日内瓦召开的国际通信会议(1CC'93)上，两位任教于法国不列颠通信大学的教授C.Berrou、A.Glavieux和他们的缅甸籍博士生P.Thitimajshima首次提出了一种新型信道编码方案——Turbo码，由于它很好地应用了Shannon信道编码定理中的随机性编、译码条件，从而获得了几乎接近Shannon理论极限的译码性能。仿真结果表明，在采用长度为65536的随机交织器并译码迭代18次情况下，在信噪比Eb/N0>=0.7dB并采用二元相移键控(BPSK)调制时，码率为1/2的Turbo码在加性高斯白噪声信道上的误比特率(BER)<=10^{-5}，达到了与Shannon极限仅相差0.7dB的优异性能。(1/2码率的Shannon极限是0dB)。

7.1.2 差错控制编码的基本原理

- 在信息码元序列中附加一些监督码元，在两者之间建立某种校验关系，当这种校验关系因传输错误而受到破坏时，可以被发现和纠正。不同的编码方法，有不同的检错和纠错能力。一般来说，付出的代价越大，检（纠）错的能力就越强。这里所说的代价，指增加的监督码元的多少。例如，若编码序列中，平均每两个信息码元就有一个监督码元，则这种编码的多余度为1/3，或者说，这种编码的编码速率为2/3，可见，差错控制编码是以降低信息传输速率来换取信息传递的可靠性提高。

7.1.3 差错控制编码的分类

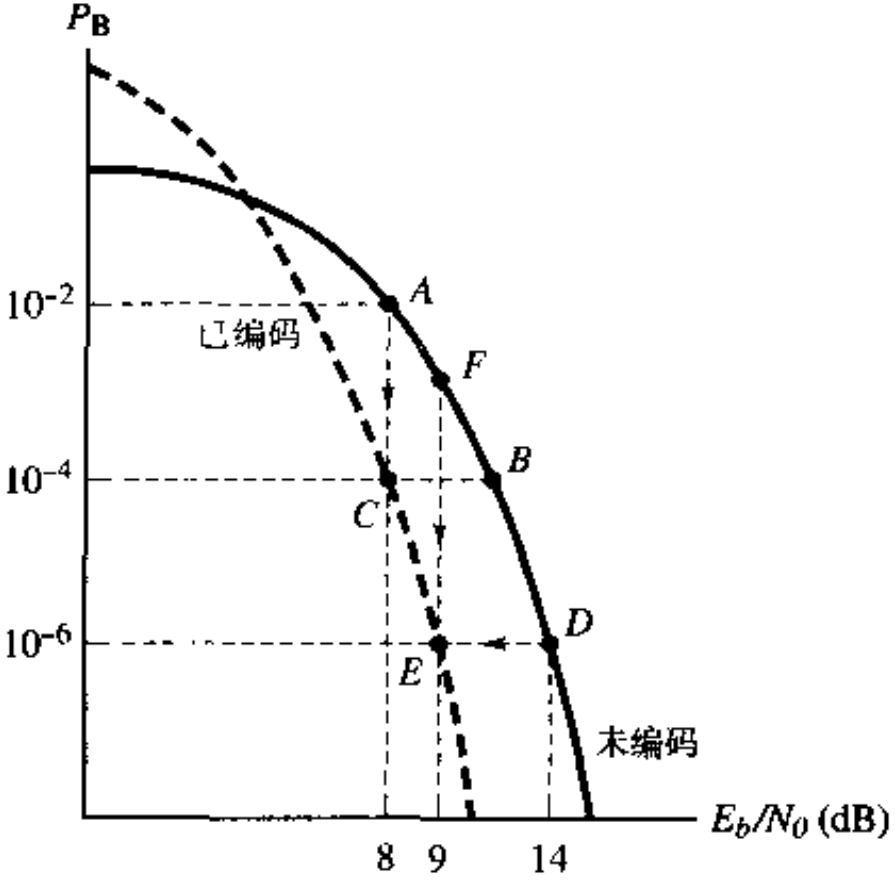
- 根据信息码元和监督码元的函数关系，可分为线性码和非线性码。
- 根据信息码元和监督码元之间的约束方式，可分为分组码和卷积码。在分组码中，监督码元仅与本码组的信息码元有关，而在卷积码中，监督码元不仅与本组的信息码元有关，还与前面若干组的信息码元有关。

7.1.4 误差控制编码的目标

- 用可以纠正的错误个数来衡量的纠错能力；
- 快速有效地对消息进行编码；
- 快速有效地对消息进行解码；
- 单位时间内所能传输的信息bit数尽量大（即尽量减少冗余度）。

使用纠错编码的原因

- 权衡1：差错性能和带宽；
- 权衡2：功率与带宽；
- 权衡3：数据速率与带宽；
- 权衡4：容量与带宽；



7.2 几种常用的简单检错码

- 奇偶监督码

  就是在原信息码元后面附加一个监督码元，使得码组中的“1”的个数为奇数或偶数。接收端译码时，对各码元进行模2加运算，其结果为0或1，如果传输过程中任何一位发生错误，就会使校验条件不满足，但当有偶数个错误发生时，这种编码就无能为力了。

- 行列监督码（水平奇偶监督码）

  对行和列都实施奇偶监督。

- 恒比码 ：即码字中“1”和“0”的数目保持恒定比例的码。

7.3 基本名词定义

- 在信道编码中，非零码元的数目称为汉明重量（Hamming Weight)，也称为码重。记为w(c)。
- 两个等长码组之间相应位取值不同的数目称为这两个码组的汉明距离（Hamming Distance），简称码距。记为$d(c_1,c_2)$,可得$d(c_1,c_2)=w(c_1-c_2)$。
- 例:考虑有两个码字{0100,1111}的码C，则w(0100)=1，w(1111)=4,这两个码字间的汉明距离为3。通过观察，有

  w(0100-1111)=w(1011)=3 =d(0100,1111)

- 码组集中任意两个码字之间距离的最小值称为最小码距d_{min}，它关系着这种编码的检错和纠错能力。
  - 为检测出e个错码， $d_{min} \geq e+1$
  - 为纠正t个错码， $d_{min} \geq 2t+1$
  - 为检测出e个错码,同时纠正t个错码，$d_{min} \geq e+t+1$且$e \geq t$

线性码具有下述性质

- 两个属于该码的码字的和仍是一个属于该码的码字；
- 全零码字总是一个有效码字；
- 一个线性码的两个码字之间的最小距离等于任何非零码字的最小汉明重量。

例：码C＝{0000,1010,0101,1111}是一个分组长度为4的线性分组码。

- 分组码一般以（n，k）表示，k是信息码元数目，n是码组总码元数，又称为码长，因此，n－k＝r就是监督码元的数目。信道编码可表示为由编码前的信息码元空间U^k到编码后的码字空间C^n的一个映射f，即 f：$U^k \rightarrow C^n$，编码速率为R=k/n。
- 在二进制情况下，共有2^k个不同的信息组，相应地有2^k个不同的码字，称为许用码组，其余2^n-2^k个就称为禁用码组。

7.4（7，4）线性分组码举例

- 7.4.1 基本概念
  - 分组码：将信息码分组，为每组信码附加若干监督码的编码，称为分组码。
  - 线性分组码：每个监督码元都是码组中某些信息码元的线性相加得到的。
- 下面以（7，4）分组码进行说明。

其码字 $A=[a_6\ a_5\ a_4\ a_3 \mid a_2\ a_1\ a_0]$

信息码元　　监督码元

$$\begin{cases} a_2=a_6+a_5+a_4 \\ a_1=a_6+a_5+a_3 \\ a_0=a_6+a_4+a_3 \end{cases}$$

- 据此，可得到16个码字，$d_{min}=3$，能检测出2个错误，纠正1个错误。

7.4.2 监督矩阵H和生成矩阵G

- 将上面的方程重写为：

$$\begin{cases} 1\cdot a_6+1\cdot a_5+1\cdot a_4+0\cdot a_3+1\cdot a_2+0\cdot a_1+0\cdot a_0=0 \\ 1\cdot a_6+1\cdot a_5+0\cdot a_4+1\cdot a_3+0\cdot a_2+1\cdot a_1+0\cdot a_0=0 \\ 1\cdot a_6+0\cdot a_5+1\cdot a_4+1\cdot a_3+0\cdot a_2+0\cdot a_1+1\cdot a_0=0 \end{cases}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

这就是监督矩阵！！

- H矩阵是一个r×n的矩阵，它的每行之间是线性无关的。 将H矩阵分为两部分：

$$H=\left[\begin{array}{cccc:ccc}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\1&0&1&1&0&0&1\end{array}\right]=[P \quad I_r]$$

所以，P矩阵是一个r×k阶的矩阵，I_r是r阶的单位阵。

- 可写为H=[P I_r]形式的矩阵称为典型监督矩阵。如果监督矩阵H不是一个典型矩阵，可以对它进行初等行变换，化为典型监督矩阵。

$$HA^T=0^T$$

- 这个式子说明H矩阵与码字的转置乘积为零，据此可作为接收码字A是否出错的依据。

- 若把监督方程补充为下列方程：

$$\begin{cases} a_6 = a_6 \\ a_5 = a_5 \\ a_4 = a_4 \\ a_3 = a_3 \\ a_2 = a_6 + a_5 + a_4 \\ a_1 = a_6 + a_5 + a_3 \\ a_0 = a_6 + a_4 + a_3 \end{cases} \Rightarrow \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \end{bmatrix}$$

- 定义：

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

则有：$A^T = G^T \cdot \begin{bmatrix} a_6 & a_5 & a_4 & a_3 \end{bmatrix}^T \Rightarrow A = \begin{bmatrix} a_6 & a_5 & a_4 & a_3 \end{bmatrix} \cdot G$

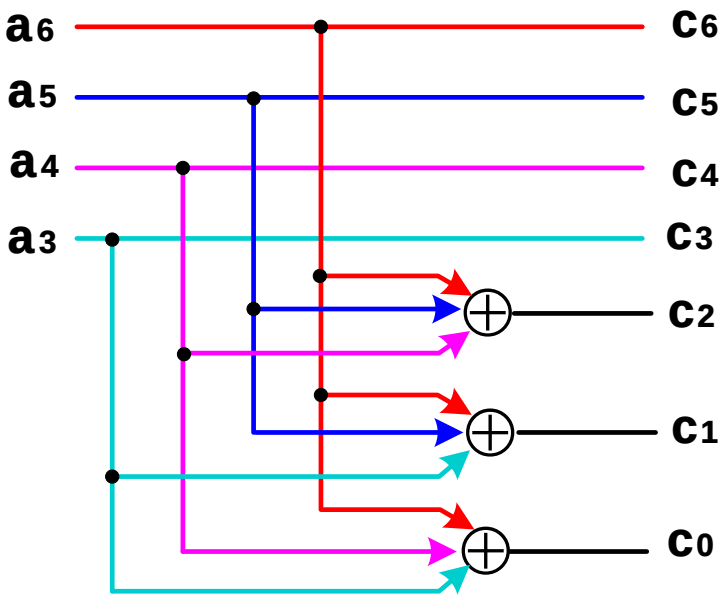
因此，由信息码元和生成矩阵G就可产生全部码字。

- 观察G，可得G=[I_k Q]

其中：$Q=\begin{bmatrix}1&1&1\\1&1&0\\1&0&1\\0&1&1\end{bmatrix}=P^T$

因此，可写为上式形式的G矩阵就称为典型生成矩阵。

(7,4)线性分组码编码器



❖ 例：已知（6，3）码的生成矩阵为 $G=\begin{bmatrix}1&0&0&1&0&1\\0&1&0&0&1&1\\0&0&1&1&1&0\end{bmatrix}$

求（1）编码码组和各个码组的码重；

（2）最小码距d_{min}和该码的差错控制能力；

➢ 解：

（1）由3位码组成的信息码组矩阵为：

$$D=\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}$$

$$A=D\cdot G=\begin{bmatrix}0&0&0&0&0&0\\0&0&1&1&1&0\\0&1&0&0&1&1\\0&1&1&1&0&1\\1&0&0&1&0&1\\1&0&1&0&1&1\\1&1&0&1&1&0\\1&1&1&0&0&0\end{bmatrix}_{8\times6}$$

码重

0

3

3

4

3

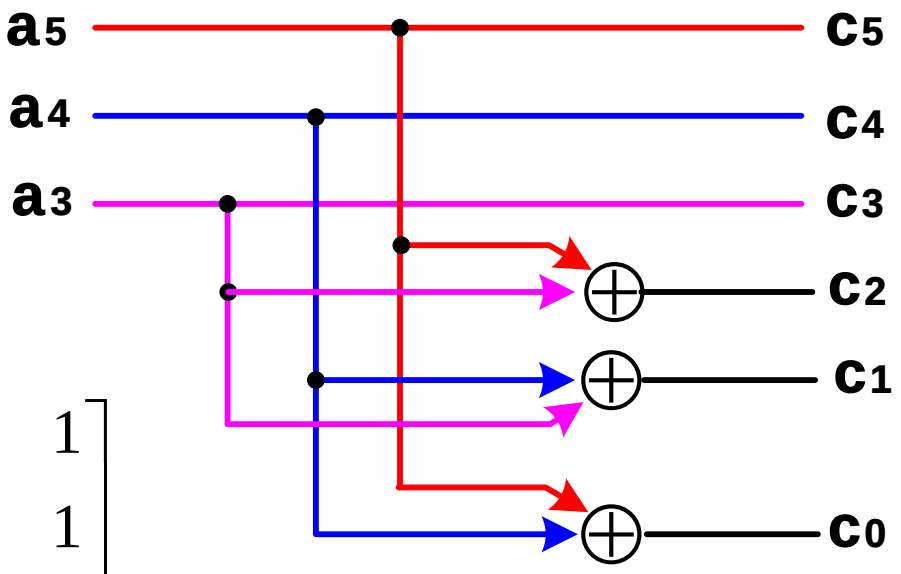
4

4

3

- （2）最小码距$d_{min}=3$，该码能检错2位，或纠错1位，或纠错1位同时检错1位的能力。

(6,3)线性分组码编码器



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

7.4.3 伴随式（校正子）S

- 码组在传输中可能由于干扰而出错，例如发送码组为A，接收到的码组却是B，它们都是n位码的行矢量，我们就定义E=B-A为错码矩阵。

  其中E=[e_{n-1} e_{n-2} … e_1 e_0]，$e_i = \begin{cases} 0 & b_i = a_i \\ 1 & b_i \neq a_i \end{cases}$

  因此有 B＝A＋E

- 定义$S=B\cdot H^T$为伴随式，则$S = (A+E)\cdot H^T = A\cdot H^T + E\cdot H^T = E\cdot H^T$

  如果传输无错，S矩阵为零矩阵；如果有错误，S就是一个非零矢量，就能从伴随式确定错误图样，然后从接收到的码字中减去错误图样，即A＝B－E，注意这里的加减都是模2加运算，就可得到正确的码组了。

- 应该注意的是，上式的解答不是唯一的。我们知道，B是一个1×n的矩阵，H^T是一个n×r的矩阵，所以S是一个1×r的矩阵，因此它有2^r种可能。而错误图样E的个数远大于2^r，因此，必然有多个错误图样对应同一个校正子S。而错误图样等于B－A，即与接收到的码组是一一对应的，为了选择正确的结果，要使用最大似然比准则，选择与B最相似的A。从几何意义上来说，就是选择与B距离最小的码组，也就是差错矢量E中1码最少的矢量。

- 对于（7，4）码来说，它的伴随式与错误图样的对应关系如下表所示：

序号	错误码位	E							S		
		e_6	e_5	e_4	e_3	e_2	e_1	e_0	s_2	s_1	s_0
0	/	0	0	0	0	0	0	0	0	0	0
1	b_0	0	0	0	0	0	0	1	0	0	1
2	b_1	0	0	0	0	0	1	0	0	1	0
3	b_2	0	0	0	0	1	0	0	1	0	0
4	b_3	0	0	0	1	0	0	0	0	1	1
5	b_4	0	0	1	0	0	0	0	1	0	1
6	b_5	0	1	0	0	0	0	0	1	1	0
7	b_6	1	0	0	0	0	0	0	1	1	1

- 由表可以看出，伴随式S的2^r种形式分别代表A码无错和2^r-1种有错的图样。

❖ 例：仍以上面的例题，已知生成矩阵G如下，列出S与E的对照表。当收到码组B＝[1 1 1 0 1 1]时，解出对应的信息码组D。

解：已知生成矩阵为：

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

I_k　　Q

$$H=[Q^T \quad I_r]=\begin{bmatrix}1&0&1&1&0&0\\0&1&1&0&1&0\\1&1&0&0&0&1\end{bmatrix}$$

S有2^3种形式，相应的码重最小的E矢量有8种。S与E的对照表如下：

E						S		
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	1	0
0	1	0	0	0	0	0	1	1
1	0	0	0	0	0	1	0	1
1	0	0	0	1	0	1	1	1

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

- （6，3）码具有纠错1位的能力，S＝111时对应双错情况，即超出了该码的纠错能力范围。

$$S = BH^T = [1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 1 \quad 1]$$

- 查表我们可知，E矢量为： **E=[0 1 0 0 0 0]**

这样我们就可得到正确的码组A，即**A=B-E=B+E=[1 0 1 0 1 1]**

所以，信息码组为：**D=[1 0 1]**

伴随式译码器

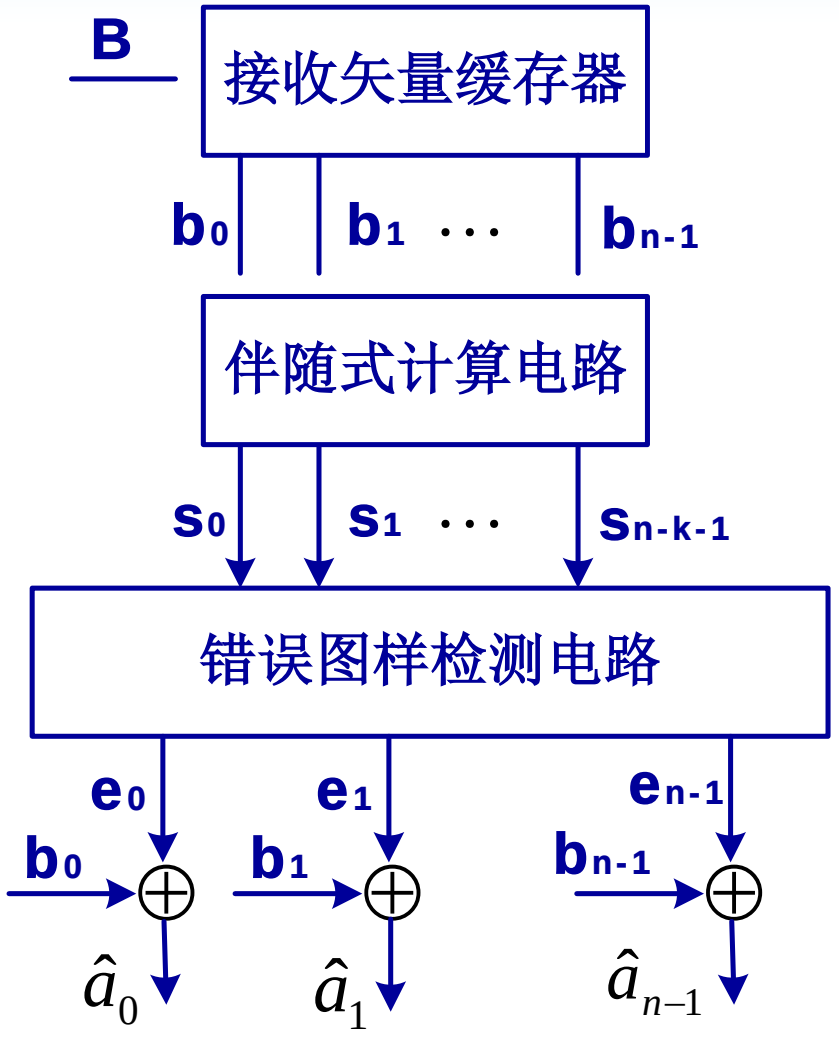
伴随式译码步骤归纳如下：

1.原始发送矢量为A；

2.计算接收矢量B的伴随式 $S=B\cdot H^T$；

3.由伴随式S决定相对应的错误图样E；

4.将B译成 $\hat{A}=B+E$ 。



- 例：(7,4)线性分组码的译码电路。前面已经给出(7,4)线性分组码的生成矩阵G和监督矩阵H为

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

假设消息序列为(0010),则经过编码后的序列为A=(0010101),接收后为B=(1010101),如何通过译码电路纠正错误!

- 根据公式$S=BH^T$,其中$S=[s_2,s_1,s_0]$, $B=[b_6,b_5,b_4,b_3,b_2,b_1,b_0]$, 可得以下关系：

$$s_2 = b_6 + b_5 + b_4 + b_2$$

$$s_1 = b_6 + b_5 + b_3 + b_1$$

$$s_0 = b_6 + b_4 + b_3 + b_0$$

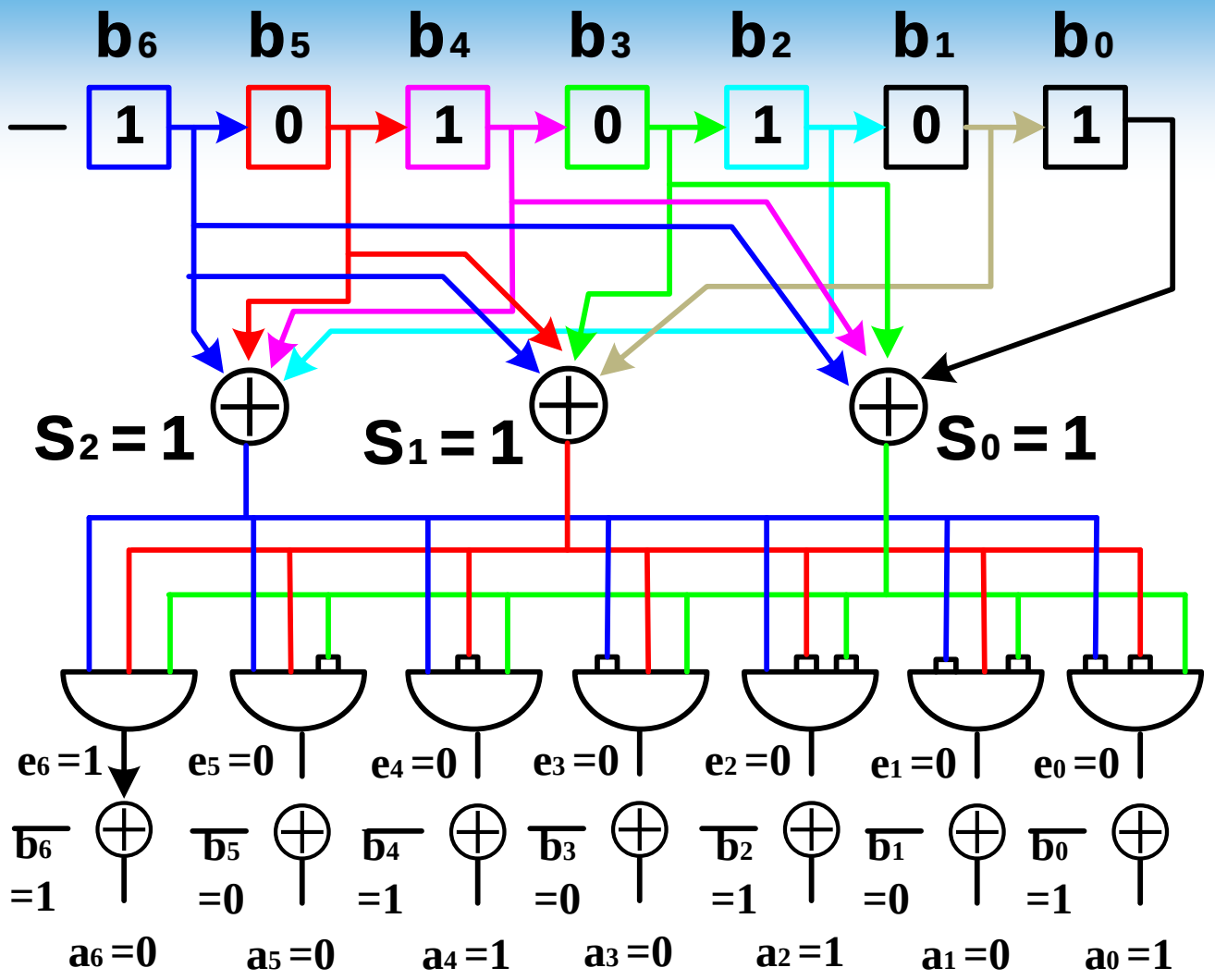
根据公式$S=EH^T$,其中$E=[e_6,e_5,e_4,e_3,e_2,e_1,e_0]$,可得以下关系：

$$e_6 = s_2 + s_1 + s_0 \;;\quad e_5 = s_2 + s_1$$

$$e_4 = s_2 + s_0 \;;\quad e_3 = s_1 + s_0$$

$$e_2 = s_2 \;;\; e_1 = s_1 \;;\quad e_0 = s_0$$

校正子S与
错误图样E
对照表



纠正了错误，输出变为了**0010101**！！

7.5 线性分组码

7.5.1 生成矩阵G和校验矩阵H

设C表示一个(n,k)线性分组码，生成矩阵G是一个k×n的矩阵，表示为

$$\mathbf{G}=\begin{pmatrix}\overline{\mathbf{v}}_0\\ \overline{\mathbf{v}}_1\\ \vdots\\ \overline{\mathbf{v}}_{k-1}\end{pmatrix}=\begin{pmatrix}v_{0,0} & v_{0,1} & \cdots & v_{0,n-1}\\ v_{1,0} & v_{1,1} & \cdots & v_{1,n-1}\\ \vdots & \vdots & \ddots & \vdots\\ v_{k-1,0} & v_{k-1,1} & \cdots & v_{k-1,n-1}\end{pmatrix}$$

任何一个码字 $\overline{\mathbf{v}}\in C$ 可表示为

$$\overline{\mathbf{v}}=u_0\overline{v}_0+u_1\overline{v}_1+\cdots+u_{k-1}\overline{v}_{k-1}$$

其中$\bar{u}=(u_0,u_1,\ldots,u_{k-1})$ 表示信息向量，$u_i \in \{0,1\}$, $1 \le i < k$

这样就有：$\overline{\mathbf{v}} = \overline{\mathbf{u}}\mathbf{G}$

生成矩阵**G**和校验矩阵**H**的关系满足$\mathbf{GH}^T=0$,从而有 $\overline{\mathbf{v}}\mathbf{H}^T = 0$

这是线性分组码解码的重要基础！

- 对系统分组码来说，其生成矩阵又可写为 $G_{sys}=(\ I_k \mid Q\ )$的形式。

其中

$$Q=\begin{pmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,n-k-1} \\ q_{1,0} & q_{1,1} & \cdots & q_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k-1,0} & q_{k-1,1} & \cdots & q_{k-1,n-k-1} \end{pmatrix}$$

由于$\mathbf{GH}^T=0$，校验矩阵的系统形式可写为 $H_{sys}=(Q^T \mid I_{n-k})$。

7.6 汉明码（Hamming）

- 为了指示所有单错位置和无错情况，线性分组码的码长n、信息码元长度k和监督码元长度r之间满足不等式：$2^r \geq n+1$；取等号时，就是汉明码。汉明码仅能纠正单个错误，是Hamming于1949年提出的。

  此时，n、k、r的关系为：$n = 2^r - 1$，$k = n - r = 2^r - r - 1$

  其中r为≥2的正整数。

- 由$2^r = n+1$，我们可知，$r = n-k = \log_2(n+1)$，$n = k+\log_2(n+1)$

上式在给定信息码组长度k后，可以求出能纠正单错的码组最小长度n，而且$d_{min}=3$。这样我们就可知道，k=1/4/11时，n=3/7/15。构成（3，1）、（7，4）、（15，11）码。

- 汉明码的编码效率为：

$$\eta = \frac{k}{n} = \frac{2^r - r - 1}{2^r - 1} = 1 - \frac{r}{2^r - 1}$$

当r很大时，η趋于1。

7.7 循环码

- 引言
- 生成多项式g(D)及生成矩阵G
- 监督多项式h(D)和监督矩阵H
- 循环码的编码和译码

7.7.1 引言

- 在处理线性分组码时，在分组码的结构上加入了线性限制的条件，这些结构上的性质可以帮助我们寻找好的能够快速、简易地编码和译码的线性分组码；本章，我们将研究线性分组码中的一个重要子类：循环码，该码在结构上有另外的限制，即一个码字任意循环移位的结果仍是一个有效码字。
- 循环码是1957年由Prange首先提出的，其特点是：(1)可以用反馈移位寄存器很容易实现编码和伴随式计算；(2)由于循环码有很多固有的代数结构，从而可以找到各种简单使用的译码方法。

多项式的运算

- 加法：$f_1(D)=D^3+D+1,\ f_2(D)=D+1$

  则 $f_1(D)+f_2(D)=D^3$

- 乘法：$f_1(D)*\ f_2(D)=D^4+D^2+D+D^3+D+1$

  $=\ D^4+\ D^3\ +D^2\ +1$

- 除法：$f_1(D)/\ f_2(D)$

$$
\begin{array}{r|l}
 & D^2+D \\
\cline{2-2}
D+1 & D^3+\quad\ D+1 \\
 & D^3+D^2 \\
\cline{2-2}
 & \quad D^2+D+1 \\
 & \quad D^2+D \\
\cline{2-2}
 & \qquad\ 1
\end{array}
$$

- 一个(n,k)线性码具有以下的属性，则称为循环码(cyclic code)：

  如果n元组$c=\{c_0,c_1,\ldots,c_{n-1}\}$是子空间S的一个码字，则经过循环移位的$c^{(1)}=\{c_{n-1}, c_0, \ldots, c_{n-2}\}$也同样是S中的一个码字，经过j次循环移位后的$c^{(j)}=\{c_{n-j},c_{n-j+1},\ldots,c_{n-1},c_0,c_1,\ldots,c_{n-j-1}\}$也是S中的一个码字。

- 码字$c=\{c_{n-1} \ \ldots \ c_1, c_0\}$的各个分量可以看作是多项式c(D)的系数，即 $c(D)=c_{n-1}D^{n-1}+\ldots+c_1D+c_0$

  每一项的存在或不存在对应了n元组中相应的位置为1或0，如果c_{n-1}非0，那么多项式的阶数为n-1。

- 如（7，3）循环码的全部码字为：

序号	码字						
0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1
2	0	1	0	0	1	1	1
3	0	1	1	1	0	1	0
4	1	0	0	1	1	1	0
5	1	0	1	0	0	1	1
6	1	1	0	1	0	0	1
7	1	1	1	0	1	0	0

- 为了便于计算，常用码多项式表示码字，如（n，k）循环码，其多项式表示为：$A(D)=a_{n-1}D^{n-1}+a_{n-2}D^{n-2}+\ldots+a_1D+a_0$

  如第2号码字可用多项式表示为：$A_2(D)=D^5+D^2+D+1$

7.7.2 生成多项式g(D)及生成矩阵G

- 如果一种码的所有码多项式都是多项式g(D)的倍式，则称g(D)为该码的生成多项式。在循环码中，次数最低的多项式（0除外）就是生成多项式g(D)，其他码多项式都是其倍数。且该g(D)的阶数为r = n – k，常数项为1，是D^n+1的一个因式。为了寻求生成多项式，必须对D^n+1进行因式分解。
- 以D^7+1为例：　　$D^7+1=(D+1)(D^3+D+1)(D^3+D^2+1)$

（n，k）	g(D)
（7，6）	$D+1$
（7，4）	D^3+D+1或D^3+D^2+1
（7，3）	$(D+1)(D^3+D+1)$或 $(D+1)(D^3+D^2+1)$
（7，1）	$(D^3+D+1)(D^3+D^2+1)$

- 循环码的生成矩阵多项式为：

$$G(D)=\begin{bmatrix} D^{k-1}g(D) \\ D^{k-2}g(D) \\ \vdots \\ Dg(D) \\ g(D) \end{bmatrix}$$

然后将系数提出就得到生成矩阵**G**。

❖ 例：已知（7，4）码的生成多项式$g(D)=D^3+D^2+1$，求生成矩阵。

解：$k=4$

$$G(D)=\begin{bmatrix} D^{k-1}g(D) \\ D^{k-2}g(D) \\ \vdots \\ Dg(D) \\ g(D) \end{bmatrix}=\begin{bmatrix} D^6+D^5+D^3 \\ D^5+D^4+D^2 \\ D^4+D^3+D \\ D^3+D^2+1 \end{bmatrix}$$

这样我们就可直接得到生成矩阵**G**为：

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

- 由A＝MG，其中M＝[m_{k-1} m_{k-2} …m_1 m_0]表示输入信息码元序列，我们可求出编码后的输出码组序列，但这样得到的循环码不是一个系统码。所谓系统码，指的是码组A的左边k位与M中的k个元素相同，而后面n－k位是M中元素的线性组合，表示监督码元。
- 为了得到系统码，就要求G矩阵的左边是一个k阶的单位阵，即是一个典型生成矩阵G ＝ [I_k Q]形式。
- 这样的系统码用多项式表示即为：

$$A(D)=D^{n-k}M(D)+r(D)$$

式中M(D)是不大于k-1次多项式，$D^{n-k}M(D)$是不大于n-1次多项式，r(D)是不大于r-1次多项式，称为监督码多项式，它等于$D^{n-k}M(D)$除以g(D)得到的余式，表示为 $r(D) = D^{n-k}M(D) \bmod g(D)$

或 $r(D) = rem\left[\dfrac{D^{n-k}M(D)}{g(D)}\right]$

- 由于典型生成矩阵$G = [I_k \ Q]$形式，与单位矩阵I_k每行对应的信息多项式为：

$$D^{n-k}m_i(D) = D^{n-k}D^{k-i} = D^{n-i}, \ i = 1,2,\ldots,k$$

$$r_i(D) = D^{n-i} \bmod g(D)$$

由此得到生成矩阵中每行的码生成多项式为：

$$C_i(D) = D^{n-i} + r_i(D), \ i = 1,2,\ldots,k$$

这样系统循环码生成矩阵多项式的一般表示式为：

$$G(D) = \begin{bmatrix} C_1(D) \\ C_2(D) \\ \vdots \\ C_k(D) \end{bmatrix} = \begin{bmatrix} D^{n-1} + r_1(D) \\ D^{n-2} + r_2(D) \\ \vdots \\ D^{n-k} + r_k(D) \end{bmatrix}$$

❖ 例7.6：我们再对前面的例题进行求解，已知$g(D)=D^3+D^2+1$，求系统循环码的生成矩阵。

解：$r_1(D)=D^6 \bmod g(D) = D^2+D$

$r_2(D)=D^5 \bmod g(D) = D+1$

$r_3(D)=D^4 \bmod g(D) = D^2+D+1$

$r_4(D)=D^3 \bmod g(D) = D^2+1$

所以，我们可写出生成矩阵多项式为：

$$G(D)=\begin{bmatrix} D^6+D^2+D \\ D^5+D+1 \\ D^4+D^2+D+1 \\ D^3+D^2+1 \end{bmatrix}$$

生成矩阵G可写为：

$$G=\left[\begin{array}{cccc:ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]=[I_k \quad Q]$$

其实这个矩阵我们也可以通过初等变换前面的生成矩阵得到。

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

第二行加到第一行 $\Rightarrow$

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

第三行加到第一行、第二行 $\Rightarrow$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

第四行加到第二行、第三行 $\Rightarrow$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

7.7.3 监督多项式h(D)和监督矩阵H

- 由于$GH^T=0$,对循环码相应的有$g(D)h(D)\equiv 0, \mathrm{mod}(D^n+1)$

$$h(D)=\frac{D^n+1}{g(D)}=D^k+h_{k-1}D^{k-1}+\ldots+h_1D+1$$

监督矩阵多项式可写为：

系数不同！！

$$H(D)=\begin{bmatrix} D^{n-k-1}h^*(D) \\ \vdots \\ D\cdot h^*(D) \\ h^*(D) \end{bmatrix}$$

其中

$$h^*(D)=D^k+h_1D^{k-1}+\ldots+h_{k-1}D+1$$

- 例：（7，3）循环码的生成多项式为$g(D)=D^4+D^3+D^2+1$，求其监督矩阵。

解：

$$h(D)=\frac{D^7+1}{g(D)}=D^3+D^2+1 \qquad h^*(D)=D^3+D+1$$

$$H(D)=\begin{bmatrix} D^3\cdot(D^3+D+1) \\ D^2\cdot(D^3+D+1) \\ D\cdot(D^3+D+1) \\ (D^3+D+1) \end{bmatrix}=\begin{bmatrix} D^6+D^4+D^3 \\ D^5+D^3+D^2 \\ D^4+D^2+D \\ D^3+D+1 \end{bmatrix}$$

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

7.7.4 循环码的编码和译码

- 我们知道，系统码用多项式表示即为： $A(D)=D^{n-k}M(D)+r(D)$，编码的关键是求出$r(D)$，而$r(D)$则要通过$r(D)=rem[D^{n-k}\cdot M(D)/g(D)]$来求解。

- 例：已知（7，4）系统循环码的生成多项式为$g(D)=D^3+D^2+1$，若信息码为1001，求编码后的循环码组。

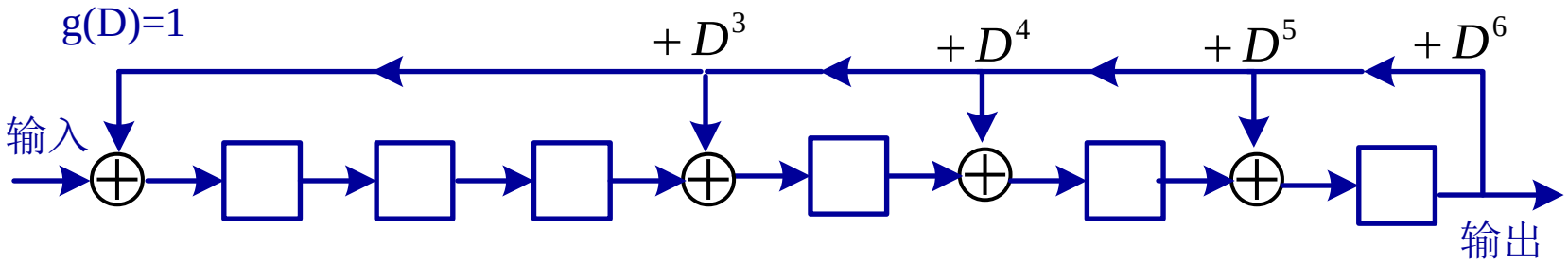
解：信息码多项式为$M(D)=D^3+1$，

$$r(D)=rem\left[\frac{D^3(D^3+1)}{D^3+D^2+1}\right]=D+1$$

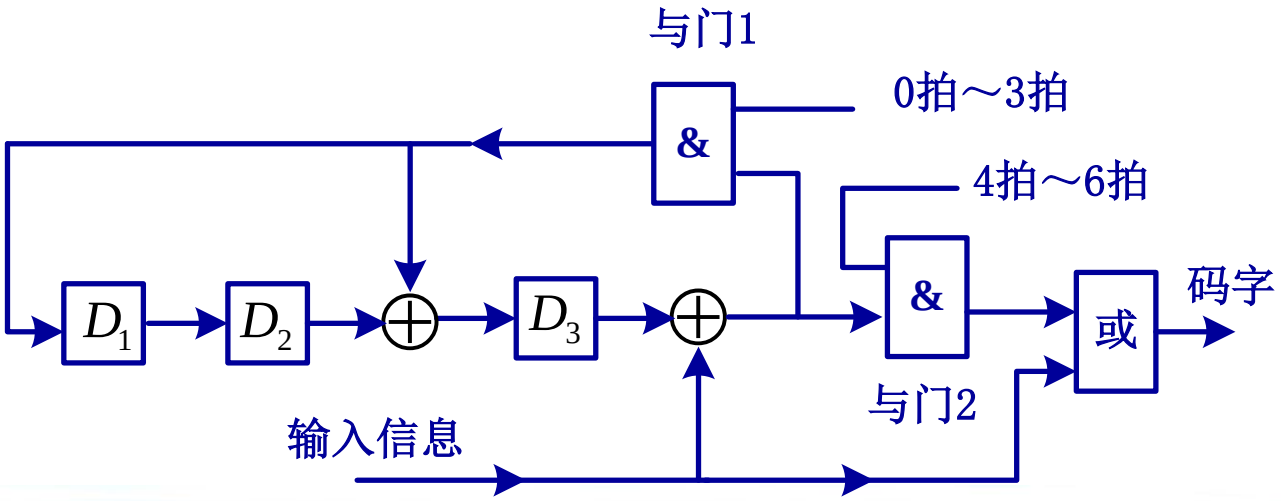
$$A(D)=D^3(D^3+1)+D+1=D^6+D^3+D+1$$

其对应的码组为1001011

- 多项式除法可以用带反馈的线性移位寄存器来实现。**g(D)**与移位寄存器的反馈逻辑相对应。如假设$g(D)=D^6+D^5+D^4+D^3+1$，则采用内接异或门的电路如下图所示，



- 以（7，4）系统循环码为例，已知它的生成多项式为$g(D)=D^3+D^2+1$，所对应的编码电路入下图所示。



- “与”门1在0拍～3拍接通，其余时间断开；“与”门2在4拍～6拍接通，其余时间断开。用3级移位寄存器D1、D2和D3以及两个模2加法器实现除法电路，反馈逻辑与g(D)相对应。“或”门把信息码元和校验码元合路，输出编码码组A(D)。由于输入信息码组直接加到除法电路的高端，相当于自动乘以D^3。当信息码组M=[1 0 1 0]时，编码过程如下表所示。在0拍时对移位寄存器状态清零。

节拍	0	1	2	3	4	5	6
信息码元	1	0	1	0	0	0	0
D_1	1	1	0	1	0	0	0
D_2	0	1	1	0	1	0	0
D_3	1	1	1	0	0	1	0
输出编码	1	0	1	0	0	0	1

- 我们知道，发送码组多项式**A(D)**是多项式**g(D)**的倍式，如果经过信道传输后发生错误，接收码组多项式**B(D)**不再是**g(D)**的倍式，可表示为：

$$\frac{B(D)}{g(D)} = x(D) + \frac{S(D)}{g(D)}$$

或写成：**S(D) = rem[B(D) / g(D)]**

其中**S(D)**是**B(D)**除以**g(D)**后的余式，是不大于**r-1**次的码组多项式，称为伴随多项式或校正子多项式。

- 接收码组多项式B(D)可表示为发送码组多项式与差错多项式之和，即：B(D) = A(D)+E(D)

$$S(D) = rem\left[\frac{A(D)+E(D)}{g(D)}\right] = rem\left[\frac{E(D)}{g(D)}\right]$$

由S(D)就可进一步确定E(D)。对于一个S(D)，E(D)可能有多种形式。由S(D)确定E(D)时同样使用最大似然比准则。对最小码重的差错多项式E(D)，由上式求出对应的伴随多项式S(D)，将E(D)与S(D)的对应关系列成译码表。当收到任一码组B(D)后，利用S(D)=rem[B(D)/g(D)]求出S(D)，对照译码表找到E(D)，再用B(D) = A(D)+E(D)求A(D)，即

A(D) = B(D) + E(D)

- 例：已知（7，4）系统循环码的生成多项式为$g(D)=D^3+D^2+1$，试构成译码表。若接收码组

B = [1 0 0 0 1 0 1]，求发送码组。

解：根据$S(D)=rem[B(D)/g(D)]$，对码重为1的差错多项式$E(D)$，求出相应的多项式$S(D)$，将其对应结果列成译码表如下：

E(D)	D^6	D^5	D^4	D^3	D^2	D	1
S(D)	D^2+D	$D+1$	D^2+D+1	D^2+1	D^2	D	1

当接收码组无误时，$E(D)=0$，则$S(D)=0$。

本题给出的接收码组为：**B = [1 0 0 0 1 0 1]**，接收码组多项式为：$B(D)=D^6+D^2+1$ 。伴随多项式$S(D)$为：

$$S(D)=rem\left[\frac{D^6+D^2+1}{D^3+D^2+1}\right]=D+1$$

查表得到：$E(D)=D^5$

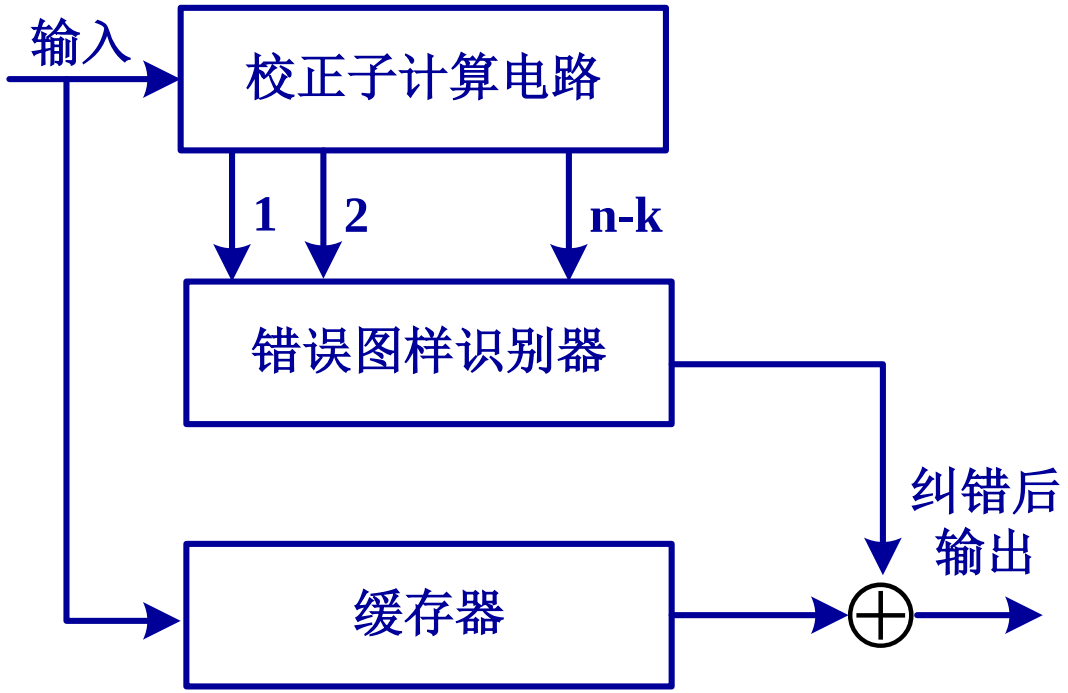
- 由B(D)和E(D)可得到译码码组多项式：

$$A(D)=B(D)+E(D)=D^6+D^5+D^2+1$$

相应的码组为：A=[1 1 0 0 1 0 1]

由于是系统循环码，所以信息码组为： M=[1 1 0 0]

- 其译码电路如下：



- 由于循环码具有很强的检测能力，因此常用于CRC校验，目前已有四个国际标准：

CRC-12: $g(D)=D^{12}+D^{11}+D^{3}+D^{2}+D+1$

CRC-16: $g(D)=D^{16}+D^{15}+D^{2}+1$

CRC-CCITT: $g(D)=D^{16}+D^{12}+D^{5}+1$

CRC-32: $g(D)=D^{32}+D^{26}+D^{23}+D^{22}+D^{16}+D^{12}+D^{11}+D^{10}+D^{8}+D^{7}+D^{5}+D^{4}+D^{2}+D+1$

作业（P288~292）

- 6.1
- 6.8
- 6.9
- 6.10
- 6.17
- 6.18
- 6.21