通信原理

第11章差错控制编码

第11章差错控制编码

11.1 概述

- 信道分类：从差错控制角度看
  - 随机信道：错码的出现是随机的
  - 突发信道：错码是成串集中出现的
  - 混合信道：既存在随机错码又存在突发错码
- 差错控制技术的种类
  - 检错重发　ARQ
  - 前向纠错　FEC
  - 反馈校验
  - 检错删除

第11章差错控制编码

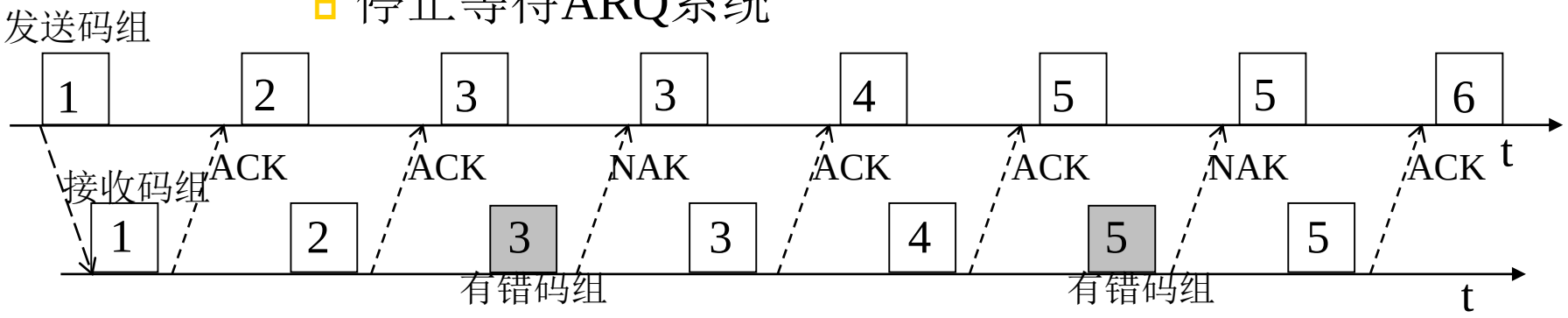
- 差错控制编码：常称为**纠错编码**

监督码元：上述4种技术中除第3种外，都是在接收端识别有无错码。所以在发送端需要在信息码元序列中增加一些差错控制码元，它们称为监督码元。

- 不同的编码方法，有不同的**检错**或**纠错**能力。
- **编码效率(简称码率)**：设编码序列中信息码元数量为k，总码元数量为n，监督码元数量为r, 则比值k/n 就是码率。
- **冗余度：**监督码元数r与信息码元数 k之比 r/k。
- 差错控制以降低信息传输速率为代价换取提高传输可靠性。
- **多余度：**就是指增加的监督码元多少 r与n之比。如：编码序列中平均每两个信息码元就添加一个监督码元，则编码的多余度为1/3。

第11章差错控制编码

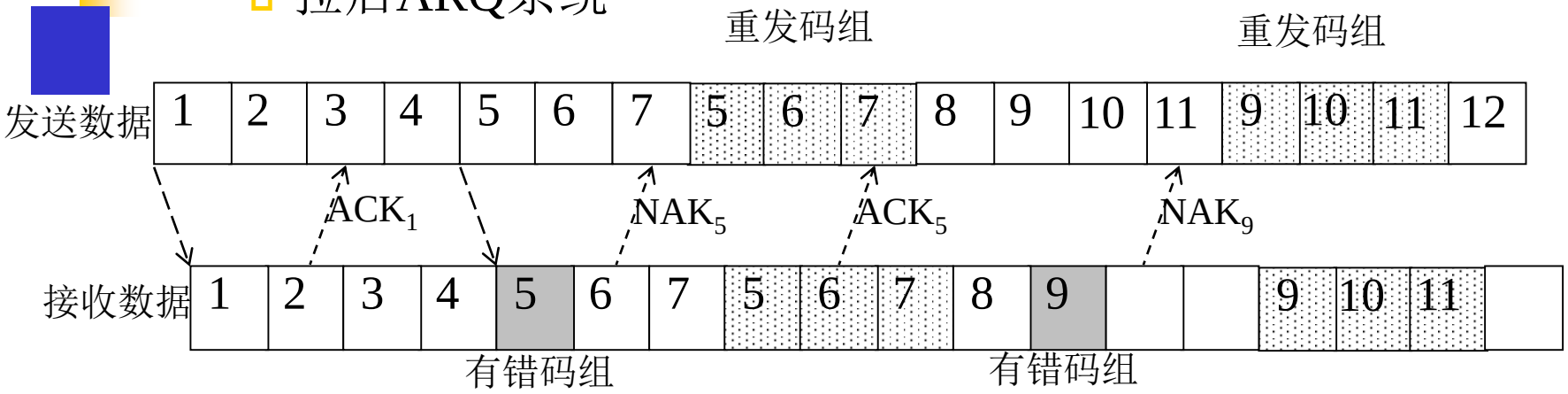
- 自动要求重发(ARQ)系统（检错重发）
  - 3种ARQ系统
    - 停止等待ARQ系统



- 数据按分组发送。每发送一组数据后发送端等待接收端的确认(ACK)答复，然后再发送下一组数据。
- 图中的第3组接收数据有误，接收端发回一个否认(NAK)答复。这时，发送端将重发第3组数据。
- 系统是工作在半双工状态，时间没有得到充分利用，传输效率较低。

第11章差错控制编码

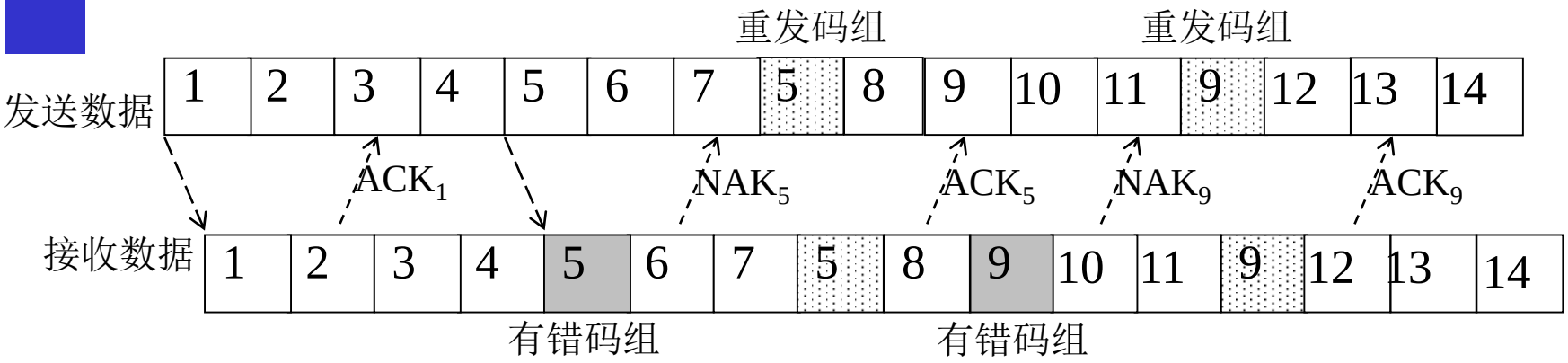
- 拉后ARQ系统



- 发送端连续发送数据码组，接收端对于每个接收到的数据码组都发回**确认(ACK)**或**否认(NAK)**答复。
- 例如，图中第5组接收数据有误，则在发送端收到第5组接收的否认答复后，从第5组开始重发数据组。
- 在这种系统中需要对发送的数据码组和答复命令进行编号，以便识别。显然，这种系统需要全双工信道

第11章差错控制编码

- 选择重发ARQ系统



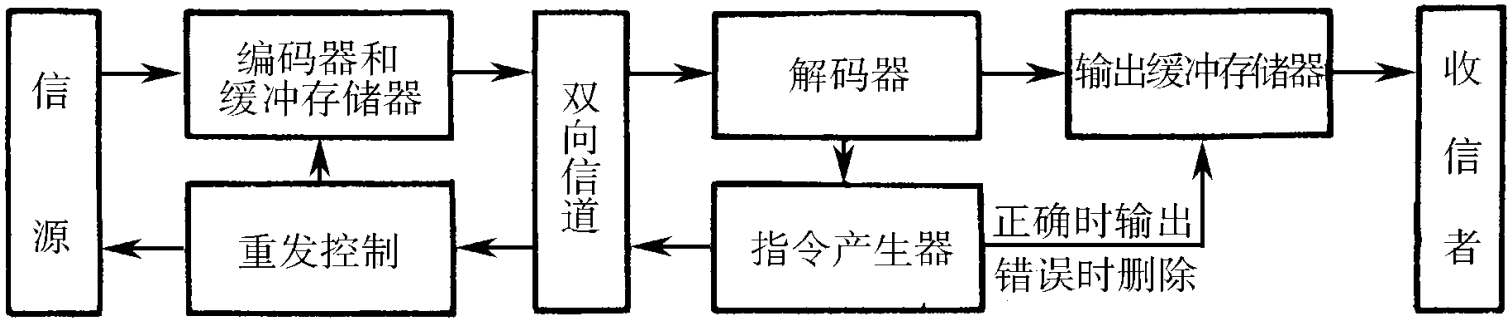
- 它只重发出错的数据组，因此进一步提高了传输效率。
- 在这种系统需要对发送的数据码组和答复命令进行编号。

第11章差错控制编码

- ARQ的主要优点：和前向纠错方法相比
  - 监督码元较少即能使误码率降到很低，即**码率较高**；
  - 检错的计算复杂度较低；
  - 检错用的编码方法和加性干扰的统计特性基本无关，能适应不同特性的信道。
- ARQ的主要缺点：
  - **需要双向信道来重发**，不能用于单向信道，也不能用于一点到多点的通信系统。
  - 因为重发而使ARQ系统的**传输效率降低**。
  - 在信道**干扰严重**时，可能发生因不断反复重发而造成事实上的**通信中断**。
  - 在要求**实时通信的场合**，例如电话通信，往往**不允许使用ARQ法**。

第11章差错控制编码

- ARQ系统的原理



- 在发送端，输入的信息码元在编码器中被分组编码（加入监督码元）后，除了立即发送外，还暂存于缓冲存储器中。若接收端解码器检出错码，则由解码器控制产生一个重发指令。此指令经过反向信道送到发送端。由发送端重发控制器控制缓冲存储器重发一次。
- 接收端仅当解码器认为接收信息码元正确时，才将信息码元送给收信者，否则在输出缓冲存储器中删除接收码元。
- 当解码器未发现错码时，经过反向信道发出不需重发指令。发送端收到此指令后，即继续发送后一码组，发送端的缓冲存储器中的内容也随之更新。

第11章差错控制编码

11.2 纠错编码的基本原理

- 分组码基本原理：举例说明如下。

- 设有一种由3位二进制数字构成的码组，它共有8种不同的可能组合。若将其全部用来表示天气，则可以表示8种不同天气，

  例如：“000”（晴），“001”（云），
  “010”（阴），“011”（雨），
  “100”（雪），“101”（霜），
  “110”（雾），“111”（雹）。

- 其中任一码组在传输中若发生一个或多个错码，则将变成另一个信息码组。这时，接收端将无法发现错误。

第11章差错控制编码

- 若在上述8种码组中只准许使用4种来传送天气，例如：

“000”＝晴　“011”＝云　“101”＝阴　“110”＝雨

这是**许用码组**

- 这时，虽然只能传送4种不同的天气，但是接收端却有可能发现码组中的一个错码。
- 例: 若“000”（晴）中错了一位，则接收码组将变成“100”或“010”或“001”。这3种码组都是不准使用的，称为**禁用码组**。
- 接收端在收到禁用码组时，就认为发现了错码。当发生3位错码时，“000”变成了“111”，它也是禁用码组，故这种编码也能检测3个错码。
- 但是这种码不能发现一个码组中的两个错码，因为发生两个错码后产生的是**许用码组**。

第11章差错控制编码

检错和纠错

- 上面这种编码只能检测错码，不能纠正错码。例如，当接收码组为禁用码组“100”时，接收端将无法判断是哪一位码发生了错误，因为晴“000”、阴“101”、雨“110”三者错了一位都可以变成“100”。
- 要能够纠正错误，还要增加多余度。例如，若规定许用码组只有两个：“000”（晴），“111”（雨），其他都是禁用码组，则能够检测两个以下错码，或能够纠正一个错码。
- 例如，当收到禁用码组“100”时，若当作仅有一个错码，则可以判断此错码发生在“1”位，从而纠正为“000”（晴）。因为“111”（雨）发生任何一位错码时都不会变成“100”这种形式。
- 但是，这时若假定错码数不超过两个，则存在两种可能性：“000”错一位和“111”错两位都可能变成“100”，因而只能检测出存在错码而无法纠正错码。

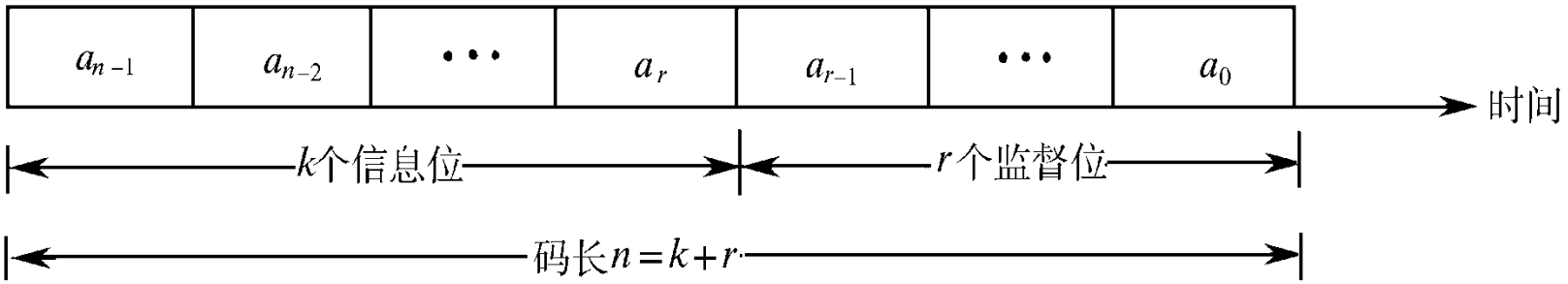
第11章差错控制编码

- 分组码的结构
  - 将信息码分组，为每组信息码附加若干监督码的编码称为分组码。
  - 在分组码中，监督码元仅监督本码组中的信息码元。
  - 信息位和监督位的关系：举例如下

	信息位	监督位
晴	00	0
云	01	1
阴	10	1
雨	11	0

第11章差错控制编码

- 分组码的一般结构



- 分组码的符号：(n, k)
  - n ——码组的总位数，又称为码组的长度（码长），
  - k ——码组中信息码元的数目，
  - $n-k=r$ ——码组中的监督码元数目，或称监督位数目。

第11章差错控制编码

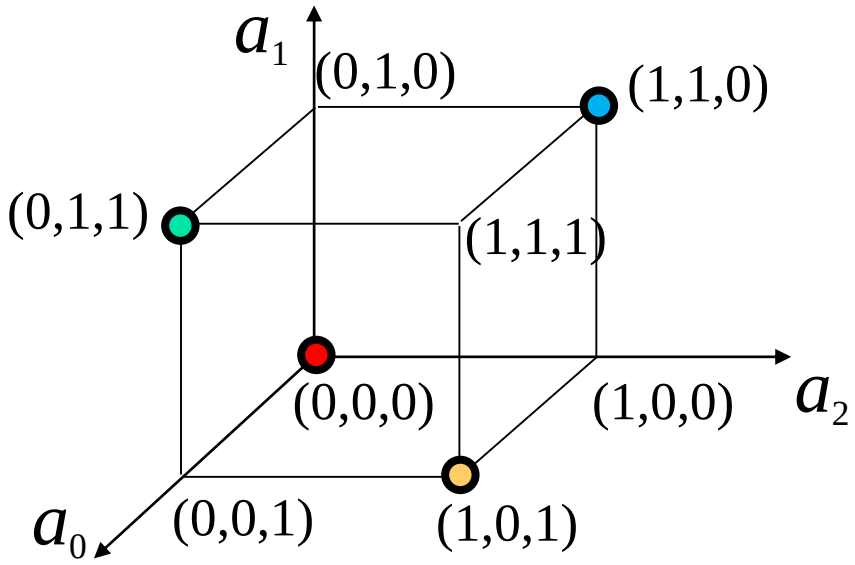
- 分组码的码重和码距
  - 码重：把码组中“1”的个数，称为码组的重量，简称**码重**。
  - 码距：把两个码组中对应位上数字不同的位数，称为码组的距离，简称**码距**。码距又称**汉明距离**。
  - 例如，“000”＝晴，“011”＝云，“101”＝阴，“110”＝雨，4个码组之间，任意两个码组的距离均为2。
  - 最小码距：把某种编码中各个码组之间距离的最小值，称为**最小码距**(d_0)。例如，上面的编码的最小码距$d_0 = 2$。

第11章差错控制编码

- 码距的几何意义

  (a_2, a_1, a_0)

  - “0 0 0”＝晴
  - “0 1 1”＝云
  - “1 0 1”＝阴
  - “1 1 0”＝雨



- 对于3位的编码组，可以在3维空间中说明码距的几何意义。
- 每个码组的3个码元的值(a_2, a_1, a_0)就是此立方体各顶点的坐标。而上述码距概念在此图中就对应于各顶点之间沿立方体各边行走的几何距离。
- 由此图可以直观看出，上例中4个许用码组之间的距离均为2。

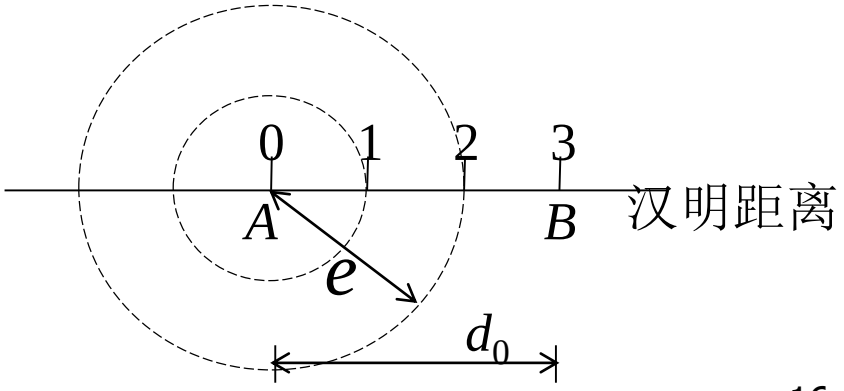
第11章差错控制编码

- 码距和检错、纠错能力的关系
  - 一种编码的最小码距d_0的大小，决定这种编码的检错和纠错能力。
  - **要求检测e个错码，最小码距应满足** $d_0 \geq e+1$

【证】设一个码组A位于O点。若码组A中发生一个错码，则我们可以认为A的位置将移动至以O点为圆心，以1为半径的圆上某点，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O点为圆心，以2为半径的圆。因此，只要最小码距不小于3，码组A发生两位以下错码时，

不可能变成另一个准用码组，因而能检测错码的位数等于2。



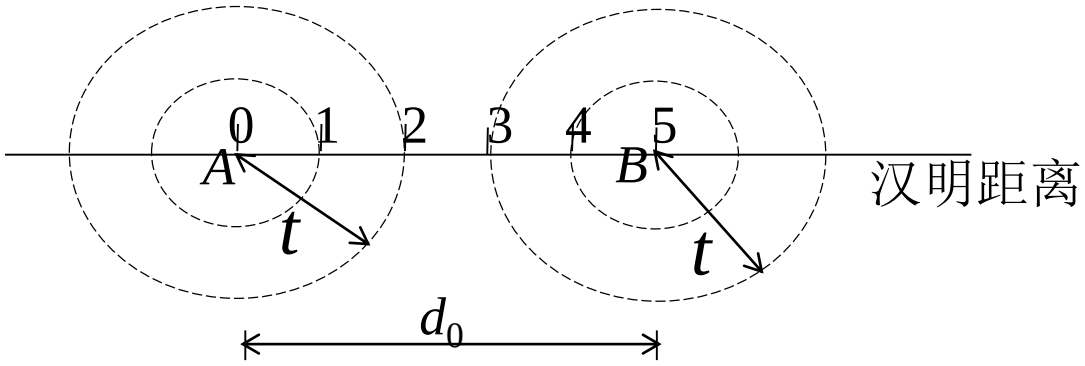
第11章差错控制编码

同理，若一种编码的最小码距为d_0，则将能检测$(d_0 - 1)$个错码。反之，若要求检测e个错码，则最小码距d_0应大于等于$(e + 1)$。

- **要求纠正t个错码，最小码距满足** $d_0 \geq 2t + 1$

【证】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半径为2以原位置为圆心的圆。这两个圆是不重叠的。判决规则为：若接收码组落于以A为圆心的圆上就判决收到的是码组A，若落于以B为圆心的圆上就判决为码组B。

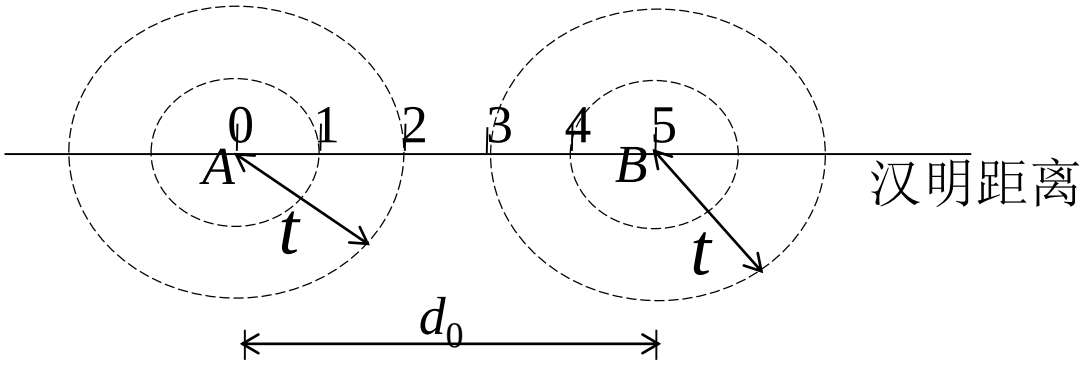
这样，就能够纠正两位错码。



第11章差错控制编码

若这种编码中除码组A和B外，还有许多种不同码组，则要求，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为半径画出之圆都不会互相重叠。

这样，每个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$d_0=5$时，能够纠正2个错码，且最多能纠正2个。若错码达到3个，就将落入另一圆上，从而发生错判。故一般说来，为纠正t个错码，最小码距应不小于$(2t+1)$。

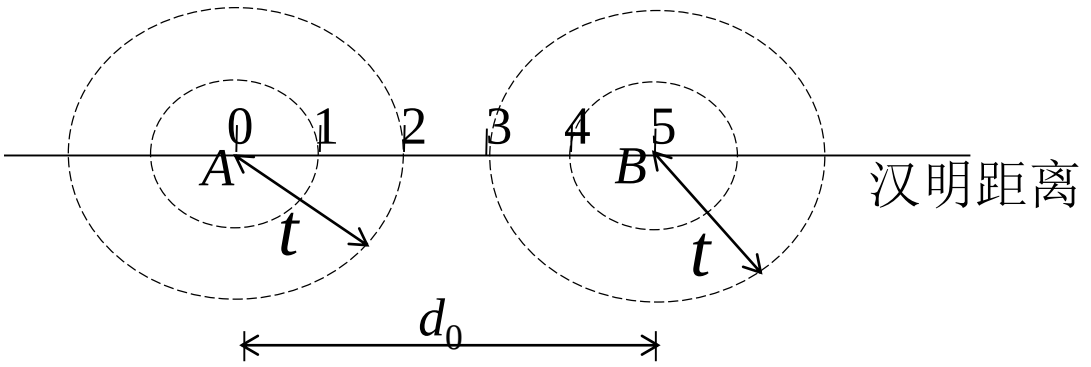


第11章差错控制编码

- 要求纠正t个错码，同时检测e个错码，最小码距需要满足

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

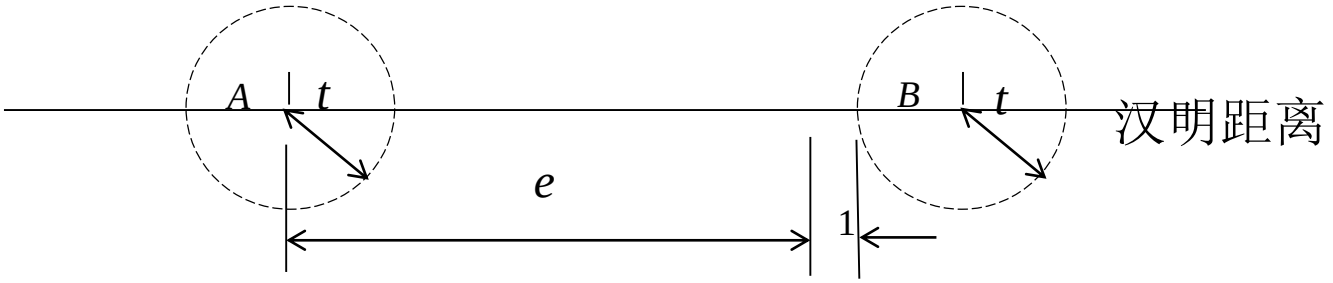
在解释此式之前，先来分析下图所示的例子。图中码组A和B之间距离为5。按照检错能力公式，最多能检测4个错码，即$e = d_0 - 1 = 5 - 1 = 4$，按照纠错能力公式纠错时，能纠正2个错码。但是，不能同时作到两者，因为当错码位数超过纠错能力时，该码组立即进入另一码组的圆内而被错误地“纠正”了。例如，码组A若错了3位，就会被误认为码组B错了2位造成的结果，从而被错“纠”为B。这就是说，检错和纠错公式不能同时成立或同时运用。



第11章差错控制编码

所以，为了在可以纠正t个错码的同时，能够检测e个错码，就需要像下图所示那样，使某一码组（譬如码组A）发生e个错误之后所处的位置，与其他码组（譬如码组B）的纠错圆圈至少距离等于1，不然将落在该纠错圆上从而发生错误地“纠正”。因此，由此图可以直观看出，要求最小码距

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



这种纠错和检错结合的工作方式简称**纠检结合**。

第11章差错控制编码

这种工作方式是自动在纠错和检错之间转换的。当错码数量少时，系统按前向纠错方式工作，以节省重发时间，提高传输效率；当错码数量多时，系统按反馈重发方式纠错，以降低系统的总误码率。所以，它适用于大多数时间中错码数量很少，少数时间中错码数量多的情况。

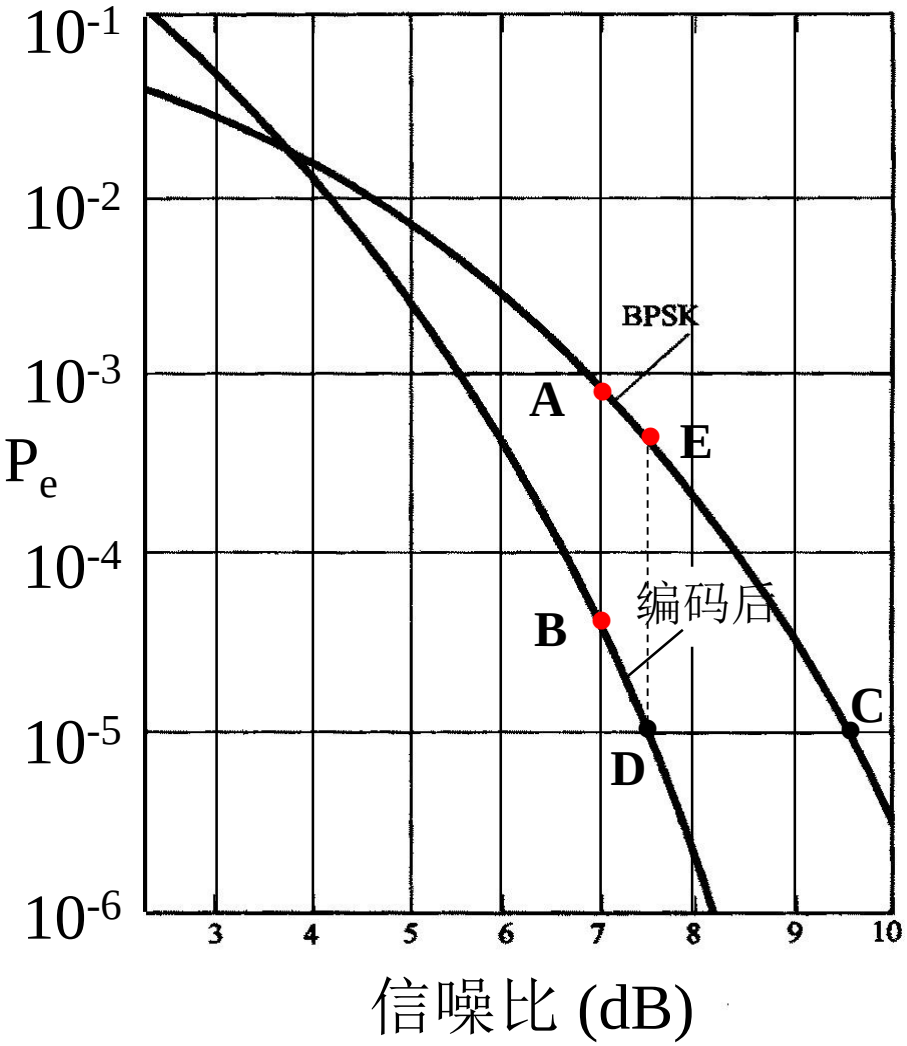
第11章差错控制编码

11.3 纠错编码的性能

- 系统带宽和信噪比的矛盾：
  - 由上节所述的纠错编码原理可知，为了减少接收错误码元数量，需要在发送信息码元序列中加入监督码元。这样作的结果使发送序列增长，冗余度增大。
  - 若仍须保持发送信息码元速率不变，则传输速率必须增大，因而增大了系统带宽。系统带宽的增大将引起系统中噪声功率增大，使信噪比下降。信噪比的下降反而又使系统接收码元序列中的错码增多。一般说来，采用纠错编码后，误码率总是能够得到很大改善的。改善的程度和所用的编码有关。

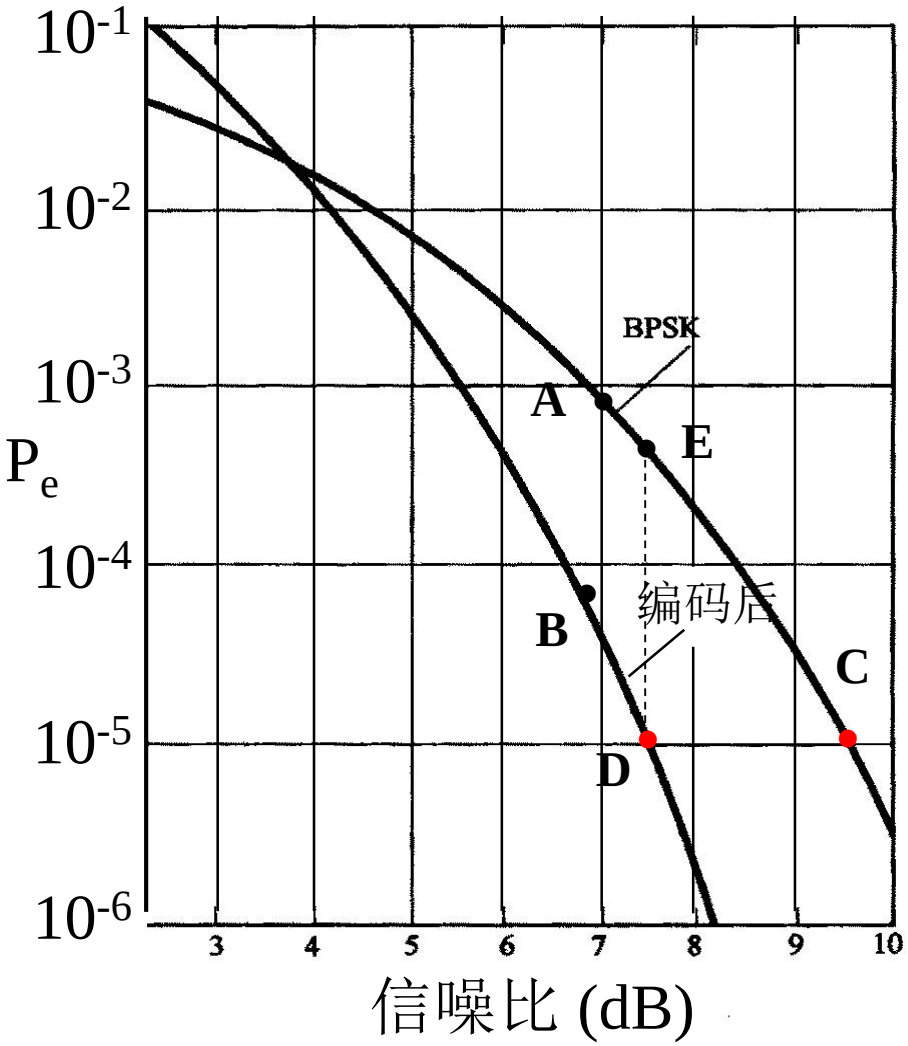
第11章差错控制编码

- 编码性能举例
  - 未采用纠错编码时，若接收信噪比等于7dB，编码前误码率约为8×10^{-4}，图中A点，在采用纠错编码后，误码率降至约4×10^{-5}，图中B点。这样，不增大发送功率就能降低误码率约一个半数量级。



第11章差错控制编码

- 由图还可以看出，若保持误码率在10^{-5}，图中C点，未采用编码时，约需要信噪比$E_b / n_0 = 9.5$ dB。在采用这种编码时，约需要信噪比7.5 dB，图中D点。可以节省功率2 dB。通常称这2 dB为编码增益。
- 上面两种情况付出的代价是带宽增大。



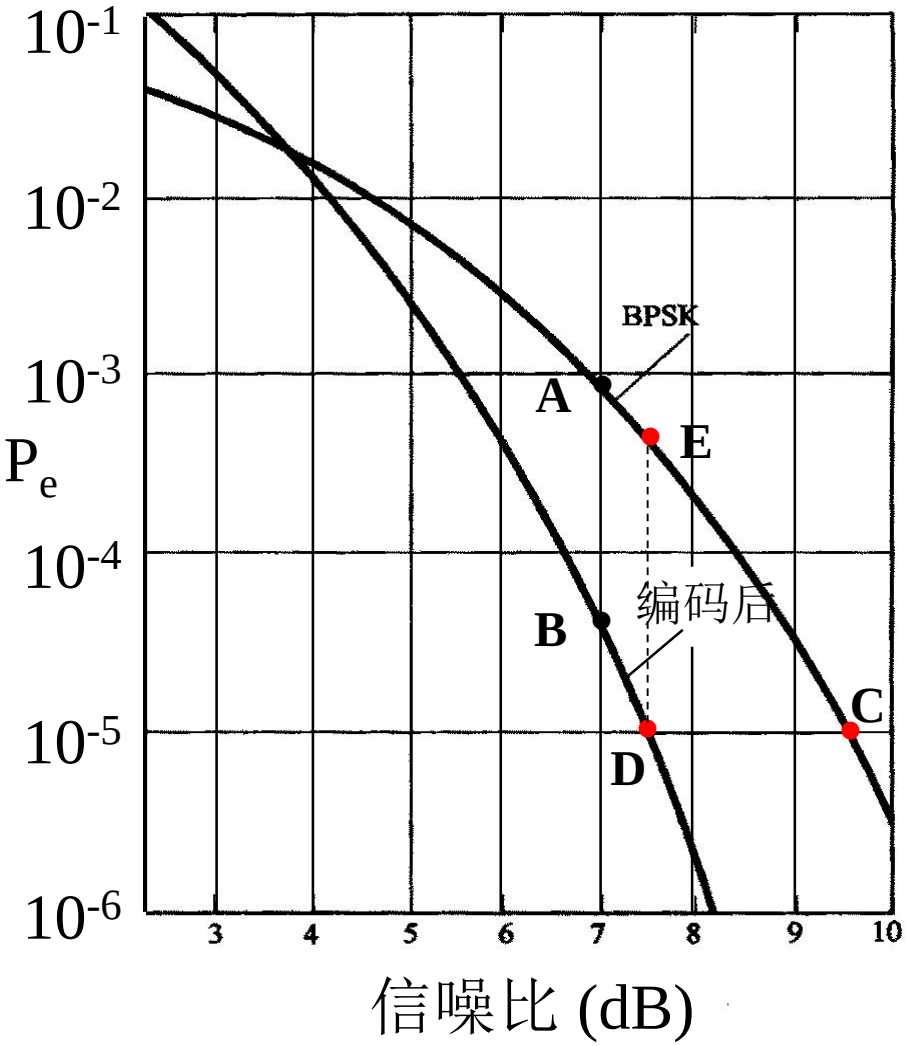
第11章差错控制编码

- 传输速率和E_b/n_0的关系

对于给定的传输系统

$$\frac{E_b}{n_0}=\frac{P_sT}{n_0}=\frac{P_s}{n_0(1/T)}=\frac{P_s}{n_0R_B}$$

式中，R_B为码元速率。若希望提高传输速率，由上式看出势必使信噪比下降，误码率增大。假设系统原来工作在图中C点，提高速率后由C点升到E点。但加用纠错编码后，仍可将误码率降到D点。这时付出的代价仍是带宽增大。



第11章差错控制编码

11.4简单的实用编码

11.4.1 奇偶监督码

- 奇偶监督码分为奇数监督码和偶数监督码两种，两者的原理相同。在偶数监督码中，无论信息位多少，监督位只有1位，它使码组中“1”的数目为偶数，即满足下式条件：

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

式中a_0为监督位，其他位为信息位。

这种编码能够检测奇数个错码。在接收端，按照上式求“模2和”，若计算结果为“1”就说明存在错码，结果为“0”就认为无错码。

奇数监督码与偶数监督码相似，只不过其码组中“1”的数目为奇数：

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

第11章差错控制编码

11.4.2 二维奇偶监督码（方阵码）

- 二维奇偶监督码的构成

它是先把上述奇偶监督码的若干码组排成矩阵，每一码组写成一行，然后再按列的方向增加第二维监督位，如下图所示

$$
\begin{array}{cccccc}
a_{n-1}^1 & a_{n-2}^1 & \cdots & a_1^1 & a_0^1 \\
a_{n-1}^2 & a_{n-2}^2 & \cdots & a_1^2 & a_0^2 \\
\cdots & \cdots & & \cdots & \\
a_{n-1}^m & a_{n-2}^m & \cdots & a_1^m & a_0^m \\
c_{n-1} & c_{n-2} & \cdots & c_1 & c_0
\end{array}
$$

图中$a_0^1\ a_0^2\ \ldots\ a_0^m$为$m$行奇偶监督码中的$m$个监督位。

$c_{n-1}\ c_{n-2}\ \ldots\ c_1\ c_0$为按列进行第二次编码所增加的监督位，它们构成了一监督位行。

第11章差错控制编码

- 二维奇偶监督码的性能

  - 这种编码有可能检测偶数个错码。因为每行的监督位虽然不能用于检测本行中的偶数个错码，但按列的方向有可能由$c_{n-1}\ c_{n-2}\ \ldots\ c_1\ c_0$等监督位检测出来。有一些偶数错码不可能检测出来。例如，构成矩形的4个错码，譬如图中

$$a_{n-2}^{2} \quad a_{1}^{2} \quad a_{n-2}^{m} \quad a_{1}^{m}$$

    错了，就检测不出。

  - 这种二维奇偶监督码适于检测突发错码。因为突发错码常常成串出现，随后有较长一段无错码的时间区间。

  - 由于方阵码只对构成矩形四角的错码无法检测，故其检错能力较强。

  - 二维奇偶监督码不仅可用来检错，还可以用来纠正一些错码。例如，仅在一行中有奇数个错码时。

第11章差错控制编码

11.4.3 恒比码

- 在恒比码中，每个码组均含有相同数目的“1”（和“0”）。即：“1”的数目与“0”的数目之比保持恒定，故得此名。
- 这种码在检测时，只要计算接收码组中“1”的数目是否对，就知道有无错码。
- 恒比码的主要优点是简单、适合用来传输电传机或其他键盘设备产生的字母和符号。对于信源来的二进制随机数字序列，这种码就不适合使用了。

第11章差错控制编码

11.4.4 正反码

- 正反码的编码：
  - 它是一种简单的能够纠正错码的编码。其中的监督位数目与信息位数目相同，监督码元与信息码元相同，或者监督码元与信息码元相反，由信息码中“1”的个数而定。
  - 例如，若码长$n = 10$，其中信息位$k = 5$，监督位$r = 5$。其编码规则为：
    - 当信息位中有奇数个“1”时，监督位是信息位的简单重复；（即：监督码元与信息码元相同）
    - 当信息位有偶数个“1”时，监督位是信息位的反码。
    - （即：监督码元与信息码元每位相反）
    - 例如，若信息位为11001，则码组为11001 11001；若信息位为10001，则码组为10001 01110。

第11章差错控制编码

- 正反码的解码

1. 生成合成吗

先将接收码组中信息位和监督位按模 2 相加，得到一个5位的合成码组。然后，由此合成码组产生一个校验码组。

2.由合成码组**产生校验码组**

- 若接收码组的信息位中有奇数个“1”，则校验码组与合成码组相同；“重”
- 若接收码组的信息位中有偶数个“1”，则校验码组是合成码组取反码。
- 最后，观察校验码组中“1”的个数，按下表进行判决及纠正可能发现的错码。

第11章差错控制编码

- 校验码组和错码的关系

	校验码组的组成	错码情况
1	全为“0”	无错码
2	有4个“1”和1个“0”	信息码中有1位错码，其位置对应校验码组中“0”的位置
3	有4个“0”和1个“1”	监督码中有1位错码，其位置对应校验码组中“1”的位置
4	其他组成	错码多于1个

例如，若发送码组为11001 11001，

则合成码组应为11001⊕11001=00000。由于接收码组信息位中有奇数个“1”，所以校验码组就是00000。按上表判决，结论接收码组中无错码。

第11章差错控制编码

若传输中产生了差错，使接收码组变成1000111001，则合成码组为10001⊕11001＝01000。由于接收码组中信息位有偶数个“1”，所以校验码组应取合成码组的反码，即10111。由于其中有4个“1”和1个“0”，按上表判断信息位中左边第2位为错码。

若接收码组错成1100101001，则合成码组变成11001⊕01001＝10000。由于接收码组中信息位有奇数个“1”，故校验码组就是10000，按上表判断，监督位中第1位为错码。

最后，若接收码组为1001111001，则合成码组为10011⊕11001＝01010，校验码组与其相同，按上表判断，这时错码多于1个。

- 上述长度为10的正反码具有纠正1位错码的能力，并能检测全部2位以下的错码和大部分2位以上的错码。

第11章差错控制编码

11.5 线性分组码

- 问题1：通信时由于信道影响使接收信息中常出现错误。这在商业、军事等应用中都会产生严重的后果。在1946年前后相当长一段时间内谁也找不出解决的办法。这个问题成**摆在科学家面前的一大难题。到1947年Hamming(海明)终于**发明了纠错编码，较好的解决了这一问题。

第11章差错控制编码

- **Richard Wesley Hamming**

1915年-1998年，1937年芝加哥大学获数学学士学位，1939年在内布拉斯加大学获得硕士学位，1942年在伊利诺伊大学获博士学位。1946-1976年30年时间在贝尔实验室担任计算机科学部的主任。美国工程院院士。

获得美国计算机协会ACM图灵奖,“计算机界的诺贝尔奖”；贡献：纠错编码，数字滤波器 。

第11章差错控制编码

11.5 线性分组码

- 本节将以汉明码为例讲解线性
- 分组码的基本原理。
  - 基本概念
    - **代数码**：建立在代数学基础上的编码。
    - **线性码**：按照一组线性方程构成的代数码。在线性码中信息位和监督位是由一些线性代数方程联系着的。
    - **线性分组码**：按照一组线性方程构成的分组码 。

第11章差错控制编码

汉明码

能够纠正1位错码且编码效率较高的一种线性分组码

- 汉明码的构造原理。
  - 在偶数监督码中，由于使用了一位监督位a_0，它和信息位$a_{n-1}\ \ldots\ a_1$一起构成一个代数式：

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

在接收端解码时，实际上就是在计算

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

若$S=0$，就认为无错码；若$S=1$，就认为有错码。现将上式称为**监督关系式**，S称为**校正子**。由于校正子S只有两种取值，故它只能代表有错和无错这两种信息，而不能指出错码的位置。

第11章差错控制编码

- 若监督位增加一位，即变成两位，则能增加一个类似的监督关系式。由于两个校正子的可能值有4中组合： 00，01，10，11，故能表示4种不同的信息。若用其中1种组合表示无错，则其余3种组合就有可能用来指示一个错码的3种不同位置。同理，r个监督关系式能指示1位错码的(2^r-1)个可能位置。
- 一般来说，若码长为n，信息位数为k，则监督位数$r=n-k$。如果希望用r个监督位构造出r个监督关系式，来指示1位错码的n种可能位置，则要求

$$2^r-1\geq n \qquad \text{或} \qquad 2^r\geq k+r+1$$

下面通过一个例子来说明如何具体构造这些监督关系式。

第11章差错控制编码

- 例：设**分组码(*n*, *k*)**中$k=4$，为了纠正1位错码，由上式$2^r \geq k+r+1$ 可知，要求监督位数 $r \geq 3$。若取 $r=3$，则$n=k+r=7$。我们用$a_6\, a_5 \ldots a_0$表示这7个码元，用S_1、S_2和S_3表示3个监督关系式中的校正子，则S_1、S_2和S_3的值与错码位置的对应关系可以由下表规定：

$S_1\,S_2\,S_3$	错码位置	$S_1\,S_2\,S_3$	错码位置
001	a_0	101	a_4
010	a_1	110	a_5
100	a_2	111	a_6
011	a_3	000	无错码

第11章差错控制编码

由表中规定可见，仅当一位错码的位置在a_2、a_4、a_5或a_6时，校正子S_1为1；否则S_1为零。这就意味着a_2、a_4、a_5和a_6四个码元构成偶数监督关系：

$$S_1 = a_6 \oplus a_5 \oplus a_4 \oplus a_2$$

$S_1 S_2 S_3$	错码位置	$S_1 S_2 S_3$	错码位置
001	a_0	101	a_4
010	a_1	110	a_5
100	a_2	111	a_6
011	a_3	000	无错码

a_1、a_3、a_5和a_6构成偶数监督关系：

$$S_2 = a_6 \oplus a_5 \oplus a_3 \oplus a_1$$

a_0、a_3、a_4 和a_6构成偶数监督关系

$$S_3 = a_6 \oplus a_4 \oplus a_3 \oplus a_0$$

第11章差错控制编码

- 在发送端编码时，信息位a_6、a_5、a_4和a_3的值由输入信号决定，因此它们是随机的。监督位a_2、a_1和a_0应根据信息位的取值按监督关系来确定，即监督位应使上3式中S_1、S_2和S_3的值为0（表示编成的码组中应无错码）：

$$S_1 = a_6 \oplus a_5 \oplus a_4 \oplus a_2$$

$$S_2 = a_6 \oplus a_5 \oplus a_3 \oplus a_1$$

$$S_3 = a_6 \oplus a_4 \oplus a_3 \oplus a_0$$

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

上式经过移项运算，解出监督位

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

给定信息位后，可以直接按上式算出监督位， 结果见下表：

第11章差错控制编码

信息位 $a_6a_5a_4a_3$	监督位 $a_2a_1a_0$	信息位 $a_6a_5a_4a_3$	监督位 $a_2a_1a_0$
0000	000	1000	111
0001	011	1001	100
0010	101	1010	010
0011	110	1011	001
0100	110	1100	001
0101	101	1101	010
0110	011	1110	100
0111	000	1111	111

第11章差错控制编码

- 接收端收到每个码组后，先计算出S_1、S_2和S_3，再查表判断错码情况。例如，若接收码组为0000011，按上述公式计算可得：$S_1 = 0$，$S_2 = 1$，$S_3 = 1$。由于$S_1\ S_2\ S_3$ 等于011，故查表可知在a_3位有1错码。

- 按照上述方法构造的码称为汉明码。表中所列的(7, 4)汉明码的最小码距$d_0 = 3$。因此，这种码能够纠正1个错码或检测2个错码。由于码率$k/n = (n - r)/n = 1 - r/n$，故当$n$很大和$r$很小时，码率接近1。可见，汉明码是一种高效编码。

$$S_1 = a_6 \oplus a_5 \oplus a_4 \oplus a_2$$

$$S_2 = a_6 \oplus a_5 \oplus a_3 \oplus a_1$$

$$S_3 = a_6 \oplus a_4 \oplus a_3 \oplus a_0$$

第11章差错控制编码

线性分组码的一般原理

线性分组码的构造

*H*矩阵

由上面(7, 4)汉明码的例子：

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

现在将上面它改写为

$$\left.\begin{array}{l} 1\cdot a_6 + 1\cdot a_5 + 1\cdot a_4 + 0\cdot a_3 + 1\cdot a_2 + 0\cdot a_1 + 0\cdot a_0 = 0 \\ 1\cdot a_6 + 1\cdot a_5 + 0\cdot a_4 + 1\cdot a_3 + 0\cdot a_2 + 1\cdot a_1 + 0\cdot a_0 = 0 \\ 1\cdot a_6 + 0\cdot a_5 + 1\cdot a_4 + 1\cdot a_3 + 0\cdot a_2 + 0\cdot a_1 + 1\cdot a_0 = 0 \end{array}\right\}$$

上式中已经将“⊕”简写成“+”。

第11章差错控制编码

$$\left.\begin{aligned}
&1\cdot a_6+1\cdot a_5+1\cdot a_4+0\cdot a_3+1\cdot a_2+0\cdot a_1+0\cdot a_0=0\\
&1\cdot a_6+1\cdot a_5+0\cdot a_4+1\cdot a_3+0\cdot a_2+1\cdot a_1+0\cdot a_0=0\\
&1\cdot a_6+0\cdot a_5+1\cdot a_4+1\cdot a_3+0\cdot a_2+0\cdot a_1+1\cdot a_0=0
\end{aligned}\right\}$$

上式可以表示成如下矩阵形式：

$$\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}\begin{bmatrix}a_6\\a_5\\a_4\\a_3\\a_2\\a_1\\a_0\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}\qquad（模2）$$

第11章差错控制编码

定义 $\boldsymbol{H}=\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}$

$\boldsymbol{A}=[a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0]$

$\boldsymbol{0}=[000]$

$$\begin{bmatrix}1110100\\1101010\\1011001\end{bmatrix}\begin{bmatrix}a_6\\a_5\\a_4\\a_3\\a_2\\a_1\\a_0\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

该矩阵关系可以简写成

$$\boldsymbol{H}\cdot\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{0}^{\mathrm{T}}\ \text{或}\quad \boldsymbol{A}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$$

将H称为监督矩阵。 右上标“T”表示将矩阵转置。例如，H^T是H的转置，即H^T的第一行为H的第一列，H^T的第二行为H的第二列等等。

只要监督矩阵H给定，编码时监督位和信息位的关系就完全确定了。

第11章差错控制编码

- ***H***矩阵的性质：

  1) ***H***的行数就是监督关系式的数目，它等于监督位的数目r。***H***的每行中“1”的位置表示相应码元之间存在的监督关系。例如，***H***的第一行1110100表示监督位a_2是由a_6 a_5 a_4之和决定的。***H***矩阵可以分成两部分，例如

$$\boldsymbol{H}=\left[\begin{array}{c:c}1110 & 100\\ 1101 & 010\\ 1011 & 001\end{array}\right]=\left[\boldsymbol{P}\boldsymbol{I}_r\right]$$

  式中，***P***为$r\times k$阶矩阵，***I***r为$r\times r$阶单位方阵。我们将具有[***P*** ***I***r]形式的 ***H*** 矩阵称为**典型阵**。

第11章差错控制编码

2) 由代数理论可知，***H***矩阵的各行应该是线性无关的，否则将得不到 *r*个线性无关的监督关系式，从而也得不到 *r*个独立的监督位。若一矩阵能写成典型阵形式[***P I***r]，则其各行一定是线性无关的。因为容易验证[***I***r]的各行是线性无关的，故[***P I***r]的各行也是线性无关的。

- ***G***矩阵：　　上面汉明码例子中的监督位公式为

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

也可以改写成矩阵形式：

$$\begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1110 \\ 1101 \\ 1011 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \end{bmatrix}$$

$$\begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1110 \\ 1101 \\ 1011 \end{bmatrix} \begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \end{bmatrix}$$

或者写成

$$[a_2 a_1 a_0] = [a_6 a_5 a_4 a_3] \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \end{bmatrix} = [a_6 a_5 a_4 a_3]\boldsymbol{Q}$$

式中，$\boldsymbol{Q}$为一个$k \times r$阶矩阵，它为$\boldsymbol{P}$的转置，即 $\boldsymbol{Q} = \boldsymbol{P}^{\mathrm{T}}$

上式表示，在信息位给定后，用信息位的行矩阵乘矩阵$\boldsymbol{Q}$就产生出监督位。

第11章差错控制编码

我们将$\boldsymbol{Q}$的左边加上1个$k \times k$阶单位方阵，就构成1个矩阵$\boldsymbol{G}$

$$\boldsymbol{G} = [I_k \boldsymbol{Q}] = \begin{bmatrix} 1000 \vdots 111 \\ 0100 \vdots 110 \\ 0010 \vdots 101 \\ 0001 \vdots 011 \end{bmatrix}$$

$\boldsymbol{G}$称为**生成矩阵**。由$\boldsymbol{G}$**矩阵**可以产生整个码组，即有

$$[a_6 a_5 a_4 a_3 a_2 a_1 a_0] = [a_6 a_5 a_4 a_3] \cdot \boldsymbol{G}$$

或者

$$\boldsymbol{A} = [a_6 a_5 a_4 a_3] \cdot \boldsymbol{G}$$

因此，如果找到了码的生成矩阵$\boldsymbol{G}$，则编码的方法就完全确定了。具有$[\boldsymbol{I}_k \boldsymbol{Q}]$形式的生成矩阵称为**典型生成矩阵**。由典型生成矩阵得出的码组$\boldsymbol{A}$中，信息位的位置不变，监督位附加于其后。这种形式的码称为**系统码**。

第11章差错控制编码

- ***G***矩阵的性质：

  1) ***G***矩阵的各行是线性无关的。因为由上式可以看出，任一码组***A***都是***G***的各行的线性组合。***G***共有k行，若它们线性无关，则可以组合出2^k种不同的码组***A***，它恰是有k位信息位的全部码组。若***G***的各行是线性相关的，则不可能由***G***生成2^k种不同的码组了。

  2) 实际上，***G***的各行本身就是一个码组。因此，如果已有k个线性无关的码组，则可以用其作为生成矩阵***G***，并由它生成其余码组。

$$[a_6a_5a_4a_3a_2a_1a_0]=[a_6a_5a_4a_3]\begin{bmatrix}1000:111\\0100:110\\0010:101\\0001:011\end{bmatrix}$$

第11章差错控制编码

错码矩阵和错误图样

- 一般说来，$\boldsymbol{A}$为一个n列的行矩阵。此矩阵的n个元素就是码组中的n个码元，所以发送的码组就是$\boldsymbol{A}$。此码组在传输中可能由于干扰引入差错，故接收码组一般说来与$\boldsymbol{A}$不一定相同。 $\boldsymbol{A}=[a_6a_5a_4a_3a_2a_1a_0]$
- 若设接收码组为一n列的行矩阵$\boldsymbol{B}$，即

$$\boldsymbol{B}=[b_{n-1}b_{n-2}\cdots b_1b_0]$$

则发送码组和接收码组之差为

$$\boldsymbol{B}-\boldsymbol{A}=\boldsymbol{E}\ (\text{模}2)$$

它就是传输中产生的**错码**行**矩阵**

$$\boldsymbol{E}=[e_{n-1}e_{n-2}\cdots e_1e_0]$$

式中

$$e_i=\begin{cases}0, & \text{当}b_i=a_i\\ 1, & \text{当}b_i\neq a_i\end{cases}$$

第11章差错控制编码

因此，若$e_i = 0$，表示该接收码元无错；若$e_i = 1$，则表示该接收码元有错。

$$\boldsymbol{B} - \boldsymbol{A} = \boldsymbol{E} \qquad \text{可以改写成} \qquad \boldsymbol{B} = \boldsymbol{A} + \boldsymbol{E}$$

例如，若发送码组$\boldsymbol{A}$ = [1000111]，错码矩阵$\boldsymbol{E}$ = [0000100]，则接收码组$\boldsymbol{B}$ = [1000011]。

错码矩阵有时也称为**错误图样**。

第11章差错控制编码

- 校正子$\boldsymbol{S}$

当接收码组有错时，$\boldsymbol{E} \neq \boldsymbol{0}$，将$\boldsymbol{B}$当作$\boldsymbol{A}$代入公式$(\boldsymbol{A} \cdot \boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0})$后，该式不一定成立。在错码较多，已超过这种编码的检错能力时，$\boldsymbol{B}$变为另一许用码组，则该式仍能成立。这样的错码是不可检测的。在未超过检错能力时，上式不成立，即其右端不等于$\boldsymbol{0}$。假设这时该式的右端为$\boldsymbol{S}$，即

$$\boldsymbol{B} \cdot \boldsymbol{H}^{\mathrm{T}} = \boldsymbol{S}$$

将$\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{E}$代入上式，可得

$$\boldsymbol{S} = (\boldsymbol{A} + \boldsymbol{E})\,\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{A} \cdot \boldsymbol{H}^{\mathrm{T}} + \boldsymbol{E} \cdot \boldsymbol{H}^{\mathrm{T}}$$

由于$\boldsymbol{A} \cdot \boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$，所以

$$\boldsymbol{S} = \boldsymbol{E} \cdot \boldsymbol{H}^{\mathrm{T}}$$

式中$\boldsymbol{S}$称为校正子。它能用来指示错码的位置。

$\boldsymbol{S}$和错码$\boldsymbol{E}$之间有确定的线性变换关系。若$\boldsymbol{S}$和$\boldsymbol{E}$之间一一对应，则$\boldsymbol{S}$将能代表错码的位置。

第11章差错控制编码

- 线性分组码的性质
  - **封闭性：**是指一种线性码中的任意两个码组之和仍为这种码中的一个码组。

    这就是说，若$\boldsymbol{A}_1$和$\boldsymbol{A}_2$是一种线性码中的两个许用码组，则$(\boldsymbol{A}_1+\boldsymbol{A}_2)$仍为其中的一个码组。这一性质的证明很简单。若$\boldsymbol{A}_1$和$\boldsymbol{A}_2$是两个码组，则有

    $$\boldsymbol{A}_1 \cdot \boldsymbol{H}^{\mathrm{T}} = 0, \qquad \boldsymbol{A}_2 \cdot \boldsymbol{H}^{\mathrm{T}} = 0$$

    将上两式相加，得出

    $$\boldsymbol{A}_1 \cdot \boldsymbol{H}^{\mathrm{T}} + \boldsymbol{A}_2 \cdot \boldsymbol{H}^{\mathrm{T}} = (\boldsymbol{A}_1 + \boldsymbol{A}_2)\, \boldsymbol{H}^{\mathrm{T}} = 0$$

    所以$(\boldsymbol{A}_1 + \boldsymbol{A}_2)$也是一个码组。

    由于线性码具有封闭性，所以两个码组($\boldsymbol{A}_1$和$\boldsymbol{A}_2$)之间的距离（即对应位不同的数目）必定是另一个码组$(\boldsymbol{A}_1 + \boldsymbol{A}_2)$的重量（即“1”的数目）。因此，码的最小距离就是码的最小重量（除全“0”码组外）。

第11章差错控制编码

11.6 循环码

11.6.1 循环码原理

- **循环性**：循环性是指任一码组循环一位（即将最右端的一个码元移至左端，或反之）以后，仍为该码中的一个码组。在下表中给出一种(7, 3)循环码的全部码组。

码组编号	信息位	监督位	码组编号	信息位	监督位
	$a_6a_5a_4$	$a_3a_2a_1a_0$		$a_6a_5a_4$	$a_3a_2a_1a_0$
1	000	0000	5	100	1011
2	001	0111	6	101	1100
3	010	1110	7	110	0101
4	011	1001	8	111	0010

例如，表中的第2码组向右移一位即得到第5码组；第3码组向左移一位即得到第6码组。

第11章差错控制编码

一般说来，若$(a_{n-1}\ a_{n-2}\ \ldots a_0)$是循环码的一个码组，则循环左移位后的码组

$$(a_{n-2}\ a_{n-3}\ \ldots\ a_0\ a_{n-1})$$

$$(a_{n-3}\ a_{n-4}\ \ldots\ \ a_{n-1}\ a_{n-2})$$

$$\ldots\ \ldots\ \ldots$$

$$(a_0\ a_{n-1}\ \ldots a_2\ a_1)$$

也是该编码中的码组。

第11章差错控制编码

- 码多项式
  - 码组的多项式表示法

把码组中各码元当作是一个多项式的系数，即把一个长度为n的码组表示成

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

例如，上表中的任意一个码组可以表示为

$$T(x)=a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$$

其中第7个码组可以表示为

7	110	0101

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

$$=x^6+x^5+x^2+1$$

这种多项式中，x仅是码元位置的标记，例如上式表示第7码组中a_6、a_5、a_2和a_0为“1”，其他均为0。因此我们并不关心x的取值。

第11章差错控制编码

- 码多项式的按模运算
  - 在整数运算中，有模n运算。例如，在模2运算中，有

$$1+1=2\equiv 0 \quad (\text{模}2),$$

$$1+2=3\equiv 1 \quad (\text{模}2),$$

$$2\times 3=6\equiv 0 \quad (\text{模}2)$$

等等。一般说来，若一个整数m可以表示为

$$\frac{m}{n}=Q+\frac{p}{n}, \qquad p<n$$

式中，Q — 整数，

则在模 n 运算下，有

$$m\equiv p \quad (\text{模}n)$$

即，在模 n 运算下，一个整数m等于它被 n 除得的余数。

第11章差错控制编码

- 在码多项式运算中也有类似的按模运算。

  若一任意多项式$F(x)$被一 n 次多项式$N(x)$除，得到商式$Q(x)$和一个次数小于n的余式$R(x)$，即

$$F(x)=N(x)Q(x)+R(x)$$

  则写为　　$F(x)\equiv R(x)$　　（模$N(x)$）

  这时，码多项式系数仍按模2 运算，即系数只取 0 和1。

  例如，　$x^4+x^2+1\equiv x^2+x+1$　　（模(x^3+1)）

$$\begin{array}{r} x \\ x^3+1 \overline{)\, x^4+x^2+0+1} \\ \underline{x^4+0+x+0} \\ x^2+x+1 \end{array}$$

应当注意，由于在模2运算中，用加法代替了减法，故余项不是x^2-x+1，而是x^2+x+1。

同理：x^3 被(x^3+1)除，得到余项1。所以有

$$x^3\equiv 1 \qquad （模(x^3+1)）$$

第11章差错控制编码

- 循环码的码多项式
  - 在循环码中，若T(x)是一个长为n的许用码组，则$x^i \cdot T(x)$在按模x^n+1运算下，也是该编码中的一个许用码组，即若

$$x^i \cdot T(x) \equiv T'(x) \qquad (\text{模}(x^n+1))$$

则$T'(x)$也是该编码中的一个许用码组。

【证】

设$T(x)$是一个长为n的许用码组，即n-1次多项式

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

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

对 $xT(x)$ 除 x^n+1 运算有：

$$
\begin{array}{r}
a_{n-1} \\
x^n+1 \overline{)\, a_{n-1}x^n + a_{n-2}x^{n-1} + \ldots + a_1x^2 + a_0x + 0} \\
\underline{\quad a_{n-1}x^n + 0 \qquad + \ldots + 0 \qquad + a_{n-1}} \\
a_{n-2}x^{n-1} + a_{n-3}x^{n-2} + \ldots + a_0x + a_{n-1}
\end{array}
$$

$$xT(x) \equiv a_{n-2}x^{n-1} + \ldots + a_1x^2 + a_0x + a_{n-1}$$

$$T(x) \overset{\text{对应}}{\leftrightarrow} (a_{n-1}, a_{n-2} \ldots a_1, a_0)$$

$$T'(x) = xT(x) \overset{\text{对应}}{\leftrightarrow} (a_{n-2}, a_{n-3} \ldots a_1, a_0, a_{n-1})$$

该码组是上一码组循环左移一位

【证】续 所以，这时有

$$T'(x) = xT(x) \equiv a_{n-2}x^{n-1} + \ldots + a_1 x^2 + a_0 x + a_{n-1}$$

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

$$\equiv a_{n-1-i}x^{n-1} + a_{n-2-i}x^{n-2} + \cdots + a_0 x^i + a_{n-1}x^{i-1} + \cdots + a_{n-i}$$

$$T'(x) = a_{n-1-i}x^{n-1} + a_{n-2-i}x^{n-2} + \cdots + a_0 x^i + a_{n-1}x^{i-1} + \cdots + a_{n-i}$$

$xT(x)$仍是一个$(n-1)$次多项式。

$$T'(x) = x^i T(x) \leftrightarrow (a_{n-1-i}, a_{n-2-i}, a_1, a_0, a_{n-1}, \cdots a_{n-i})$$

$$(a_{n-1,}, a_{n-2.}, \cdots a_1, a_0)$$

第11章差错控制编码

$$T'(x) = a_{n-1-i}x^{n-1} + a_{n-2-i}x^{n-2} + \cdots + a_0 x^i + a_{n-1}x^{i-1} + \cdots + a_{n-i}$$

上式中$T'(x)$正是$T(x)$代表的码组向左循环移位i次的结果。因为原已假定$T(x)$是循环码的一个码组，所以$T'(x)$也必为该码中一个码组。例如，循环码组

$$T(x) = x^6 + x^5 + x^2 + 1$$

其码长$n = 7$。现给定$i = 3$，则

$$\begin{aligned} x^3 \cdot T(x) &= x^3(x^6 + x^5 + x^2 + 1) \\ &= x^9 + x^8 + x^5 + x^3 \\ &= x^5 + x^3 + x^2 + x \qquad (\text{模}(x^7 + 1)) \end{aligned}$$

其对应的码组为0101110，它正是表中第3码组。

要点：一个长为n的循环码必定为按模$(x^n + 1)$运算的一个余式。

第11章差错控制编码

- 循环码的生成矩阵$\boldsymbol{G}$
  - 由上节中公式

$$\boldsymbol{A}=[a_6a_5a_4a_3]\cdot\boldsymbol{G}$$

可知，有了生成矩阵$\boldsymbol{G}$，就可以由k个信息位得出整个码组，而且生成矩阵$\boldsymbol{G}$的每一行都是一个码组。例如，在此式中，若$a_6a_5a_4a_3$ = 1000，则码组$\boldsymbol{A}$就等于$\boldsymbol{G}$的第一行；若$a_6a_5a_4a_3$ = 0100，则码组$\boldsymbol{A}$就等于$\boldsymbol{G}$的第二行；等等。由于$\boldsymbol{G}$是k行n列的矩阵，因此若能找到 k 个已知码组，就能构成矩阵$\boldsymbol{G}$。如前所述，这k个已知码组必须是线性不相关的，否则给定的信息位与编出的码组就不是一一对应的。

$$\boldsymbol{G}=\begin{bmatrix}1000:111\\0100:110\\0010:101\\0001:011\end{bmatrix}$$

  - 在循环码中，一个(n, k)码有2^k个不同的码组。若用$g(x)$表示其中前$(k\text{-}1)$位皆为“0”的码组，则$g(x)$，$x\,g(x)$，$x^2\,g(x)$，…，$x^{k-1}\,g(x)$都是码组，而且这k个码组是线性无关的。因此它们可以用来构成此循环码的生成矩阵$\boldsymbol{G}$。

第11章差错控制编码

- 在循环码中除全“0”码组外，**存在、且只有一个$(k-1)$位连续“0”的码组，没有连续k位均为“0”的码组。**否则，在经过若干次循环移位后将得到一个k位信息位全为“0”，但监督位不全为“0”的一个码组。这在线性码中显然是不可能的。**$g(x)$必须是一个常数项不为“0”的$(n-k)$次多项式，而且这个$g(x)$还是这种(n, k)码中次数为$(n-k)$的唯一多项式。**

在一个(n ,k)循环码中，存在一个且仅有一个g(x)多项式：

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_2 x^2 + g_1 x + 1$$

第11章差错控制编码

要点： $g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_2 x^2 + g_1 x + 1$

1．g(x)的常数项必须不为零，最高项系数为“1”。

2．g(x)是(n ,k)码中次数为(n-k)的唯一多项式。

3．循环码中，除全“0”码组外，再没有连续k-1位均为“0”的码组，连“0”的长度最多只能有k-1位。

- 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$(k-1)$。显然，这与前面的结论矛盾，故是不可能的。称这唯一的$(n-k)$次多项式$g(x)$为码的**生成多项式**。一旦确定了$g(x)$，则整个(n, k)循环码就被确定了。

第11章差错控制编码

- 因此，循环码的生成矩阵$\boldsymbol{G}$可以写成

$$\boldsymbol{G}(x)=\begin{bmatrix} x^{k-1}g(x) \\ x^{k-2}g(x) \\ \vdots \\ xg(x) \\ g(x) \end{bmatrix}$$

- 例：在上表所给出的(7, 3)循环码中，$n=7, k=3, n-k=4$。由此表可见，唯一的一个$(n-k)=4$次码多项式代表的码组是第二码组0010111，与它相对应的码多项式（即生成多项式）$g(x)=x^4+x^2+x+1$。将此$g(x)$代入上式，得到

$$\boldsymbol{G}(x)=\begin{bmatrix} x^{2}g(x) \\ xg(x) \\ g(x) \end{bmatrix} \quad \text{或} \quad \boldsymbol{G}(x)=\begin{bmatrix} 1011100 \\ 0101110 \\ 0010111 \end{bmatrix}$$

第11章差错控制编码

由于上式不符合$\boldsymbol{G}=[\boldsymbol{I}_k\boldsymbol{Q}]$的形式，所以它不是典型阵。不过，将它作线性变换，不难化成典型阵。

我们可以写出此循环码组，即

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

$$=a_6x^2g(x)+a_5xg(x)+a_4g(x)$$

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

上式表明：所有码多项式$T(x)$都可被$g(x)$整除，**而且任意一个次数不大于$(k-1)$的多项式乘$g(x)$都是码多项式。**

说明：两个矩阵相乘的结果应该仍是一个矩阵。上式中两个矩阵相乘的乘积是只有一个元素的一阶矩阵，这个元素就是$T(x)$。为了简洁，式中直接将乘积写为此元素。

第11章差错控制编码

- 如何寻找任一(n, k)循环码的生成多项式

由上式可知，任一循环码多项式$T(x)$都是$g(x)$的倍式，故它可以写成

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

$$T(x) = h(x)\cdot g(x)$$

而生成多项式g(x)本身也是一个码组，即有

$$T'(x) = g(x)$$

由于码组$T'(x)$是一个$(n-k)$次多项式，故$x^k\,T'(x)$是一个n次多项式。由下式(11.6-10)

$$x^i \cdot T'(x) \equiv T(x) \qquad (\text{模}(x^n+1))$$

可知，$x^k\,T'(x)$在模(x^n+1)运算下也是一个码组，故可以写成

$$\frac{x^kT'(x)}{x^n+1} = Q(x) + \frac{T(x)}{x^n+1}$$

第11章差错控制编码

$$\frac{x^k T'(x)}{x^n+1} = Q(x) + \frac{T(x)}{x^n+1}$$

上式左端分子和分母都是n次多项式，故商式$Q(x) = 1$。因此，上式可以化成：($T(x) = h(x)\cdot g(x)$; $T'(x) = g(x)$)

$$x^k T'(x) = (x^n+1) + T(x) \qquad (x^n+1) = x^k T'(x) + T(x)$$

将$T(x)$和$T'(x)$表示式代入上式，经过化简后得到

$$x^n + 1 = g(x)[x^k + h(x)]$$

上式表明，生成多项式$g(x)$应该是$(x^n + 1)$的一个因子，为我们寻找循环码的生成多项式指出了一条道路。

结论：循环码的生成多项式是$(x^n + 1)$的一个$(n - k)$次因式。

例如，$(x^7 + 1)$可以分解为

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

为了求(7, 3)循环码的生成多项式$g(x)$，需要从上式中找到一个$(n - k) = 4$次的因式。不难看出，这样的因式有两个，即

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

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

以上两式都可作为生成多项式。不过，选用的生成多项式不同，产生出的循环码码组也不同。

第11章差错控制编码

11.6.2 循环码的编解码方法

- 循环码的编码方法
  - 编码原则

    在编码时，首先要根据给定的(n, k)值选定生成多项式$g(x)$，即从$(x^n + 1)$的因子中选一个$(n - k)$次多项式作为$g(x)$。

由于所有码多项式$T(x)$都可以被$g(x)$整除。根据这条原则，就可以对给定的信息位进行编码：

设$m(x)$为信息码多项式，其次数小于k。用x^{n-k}乘$m(x)$，得到的$x^{n-k}\,m(x)$的次数必定小于n。用$g(x)$除$x^{n-k}\,m(x)$，得到余式$r(x)$，$r(x)$的次数必定小于$g(x)$的次数，即小于$(n-k)$。将此余式$r(x)$加于信息位之后作为监督位，即将$r(x)$和$x^{n-k}\,m(x)$相加，得到的多项式必定是一个码多项式。因为它必须能被$g(x)$整除，且商的次数不大于$(k-1)$。

第11章差错控制编码

- 编码步骤：
  - 用x^{n-k}乘$m(x)$。这一运算实际上是在信息码后附加上$(n-k)$个“0”。例如，信息码为110，它相当于$m(x)=x^2+x$。当$n-k=7-3=4$时，$x^{n-k}m(x)=x^4(x^2+x)=x^6+x^5$，它相当于1100000。
  - 用$g(x)$除$x^{n-k}m(x)$，得到商$Q(x)$和余式$r(x)$，即

$$\frac{x^{n-k}m(x)}{g(x)}=Q(x)+\frac{r(x)}{g(x)}$$

例如，若选定$g(x)=x^4+x^2+x+1$，则

$$\frac{x^{n-k}m(x)}{g(x)}=\frac{x^6+x^5}{x^4+x^2+x+1}=(x^2+x+1)+\frac{x^2+1}{x^4+x^2+x+1}$$

上式相当于

$$\frac{1100000}{10111}=111+\frac{101}{10111}$$

第11章差错控制编码

- 编出的码组$T(x)$为

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

在上例中，$T(x) = 1100000 + 101 = 1100101$，它就是上表中的第7码组。

第11章差错控制编码

- 循环码的解码方法
  - 解码要求：检错和纠错。
  - 检错解码原理：由于任意一个码组多项式$T(x)$都应该能被生成多项式$g(x)$整除，所以在接收端可以将接收码组$R(x)$用原生成多项式$g(x)$去除。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

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

因此，就以余项是否为零来判别接收码组中有无错码。

需要指出，有错码的接收码组也有可能被$g(x)$整除。这时的错码就不能检出了。这种错误称为不可检错误。不可检错误中的误码数必定超过了这种编码的检错能力。

第11章差错控制编码

- 纠错解码原理：为了能够纠错，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。因为只有存在上述一一对应的关系时，才可能从上述余式唯一地决定错误图样，从而纠正错码。因此，原则上纠错可按下述步骤进行：
  - 用生成多项式$g(x)$除接收码组$R(x)$，得出余式$r(x)$。
  - 按余式$r(x)$，用查表的方法或通过某种计算得到错误图样$E(x)$；例如，通过计算校正子S和查表，就可以确定错码的位置。
  - 从R(x)中减去$E(x)$，便得到已经纠正错码的原发送码组$T(x)$。
- 通常，一种编码可以有几种纠错解码方法，上述解码方法称为捕错解码法。

- 目前多采用软件运算实现上述编解码运算。

作业

- 作业**17**

- 思考题：11－1；11－ 3 ；11－5；
- 11－9；11－ 10 ；
- 习题：11－1；11－2；11－5；
- 11－7；11－8；