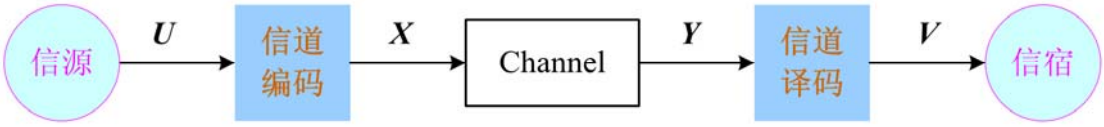
BUPT

# 信道编码



## Contents

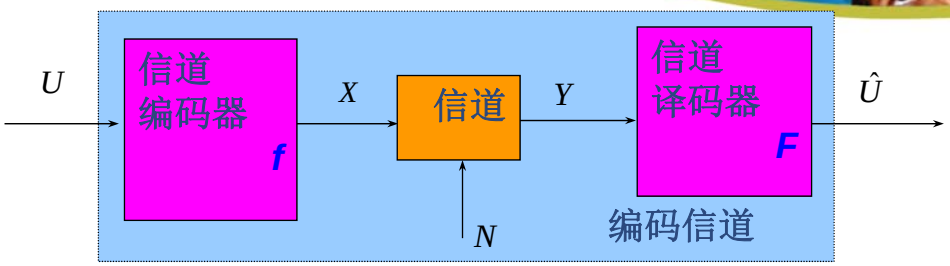
# 信道编码是干什么的？




- 信道中存在干扰与噪声，数据的传输出现差错在所难免。信道编码提供一种差错的保护形式
- 一组$k$个比特通过信道裸传，难免会出现差错。
  - 除非信噪比无限大，否则误码的概率总大于0
- 接收端不知道是否存在错或哪个比特错了
  - 例如：接收到1110，那么在接收机看来，可能是
    - 发送的本来就是1110
    - 发送的本来是0000，但因为前3个比特出错，所以我看到了1110
    - 一共有16种可能性，接收机自己不可能排除任何一种可能性



# 问题的提出



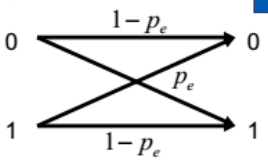

- 衡量信息传输可靠性的**指标**：平均差错率$P_e$。
- 降低$P_e$的**方法**：
  - 前向纠错编码（FEC）—— 增加冗余，提供链路保护能力
  - 反馈重传（ARQ）—— 按需重复传输
  - 混合重传（H-ARQ）—— 结合前两者的优点

# 常用差错控制方法-ARQ

## ❖反馈重传（检错重传/自动请求重复:ARQ）

- 发端发送有一定检错能力的码，收端译码时如发现有错，则通知发端重发，直到正确接收或达到最大重传次数
- 优点：检错简单，码的效率和结构简单，译码电路简单。
- 缺点：需要反馈信道，不能单向通信；实时性差。


□对称二进制信道BSC




□1个bit，只传1次，出错概率$p_e$

□1个bit，重传3次，大数判决，即：收到2个及以上的1，判为1；收到2个及以上的0，判为0

□此时，出错概率 $[3p_e^2(1-p_e)+p_e^3]\approx p_e^2$



# 常用差错控制方法-FEC

## ■前向纠错(FEC)

- 发端发送有一定纠错能力的码，若传输中产生的差错的数目在码的纠错能力内，收端可以纠正。
- 优点：单向通信（不需要反馈信道），实时性好。
- 缺点：码的构造复杂，译码电路复杂。

# 常用差错控制方法-HARQ

- 混合差错控制（HARQ）
  - FEC和ARQ的结合
  - 需要反馈信道。
  - 实时性和译码复杂性是FEC和ARQ两种方式的折衷。




# 有噪信道编码定理

- 为降低平均差错率，通过对消息增加冗余方式进行信道编码，这将降低信息传递的速度。
- 问题：是否能找到一种信道编码方法能同时保证差错率和信息传输速度的要求呢？

香农编码定理(1948)： 若信道是离散、无记忆、平稳的，且信道容量为$C$，只要待传送的信息率$R<C$，就一定能找到一种信道编码方法，使得码长足够大时，平均差错率任意接近于零。

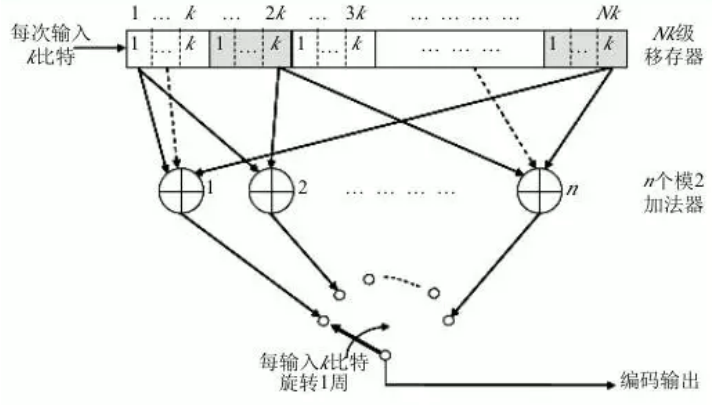
# 信道编码的发明历史-1

- 1950年， Bell Lab的研究员Hamming提出了第一种结构化纠错码
  - 其理论基础是Galois提出的群论
  - 从此，代数学被广泛应用于纠错码
- 瞻仰牛人：
  - Elwyn Berlekamp 代数编码理论
  - Goppa代数几何码
  - Bob Li环论网络编码，代数交换理论



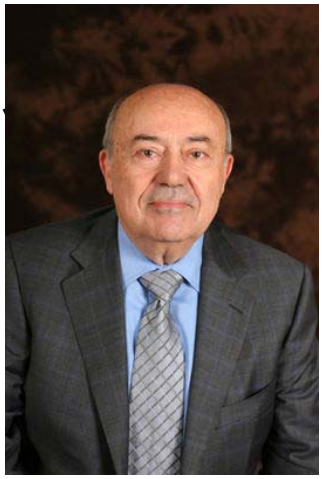

# 信道编码的发明历史-2

- 接下来的10年，无线通信性能出现跳跃式的发展，这主要归功于Elias在1955年提出的卷积码。
  - 与分组码的不同：它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。




# 信道编码的发明历史-3

- 要提高信号编码效率达到信道容量，就要使编码的分段尽可能加长而且使信息的编码尽可能随机。
  - 但是，困难在于“计算复杂性”问题。
- 1967年，Viterbi提出了Viterbi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、IS-95CDMA、3G、商业卫星通信系统等
  - 还应用于：DNA检测、语音识别……



- 其后，编码专家们苦苦思索，设计编码和算法，以提高效率，但其增益与香农理论极限始终都存在2~3dB的差距。



# 信道编码的发明历史-4

- 直到1993年，在日内瓦召开的IEEE通信国际会议上，两位当时名不见经传的法国电机工程师C. Berrou和A. Glavieux声称他们发明了一种Turbo编码，可以使信道编码效率接近香农极限。
  - 凭着电机工程师的经验，他们发现在电子学中经常用到的反馈概念似乎被数学家们忽略。
  - 他们摈弃了“纯粹”的数字化概念。
    - 在典型的数字化方法中，总是先把某一电平设定为阈值。信号电平高于这一阈值就判决为“1”低于就判决为“0”。

在Turbo码解码过程中，某一特定比特的电平被量化为整数，例如从-127 到+127。其数值就作为判决该比特为“1”或“0”的可置信度的度量（例如-110意味该比特非常非常可能是“0”，而+40 意味该比特也许是“1”但把握不大）。

  - Turbo码成为了始于本世纪初的3G/4G移动通信技术的核心，直到今天4.5G

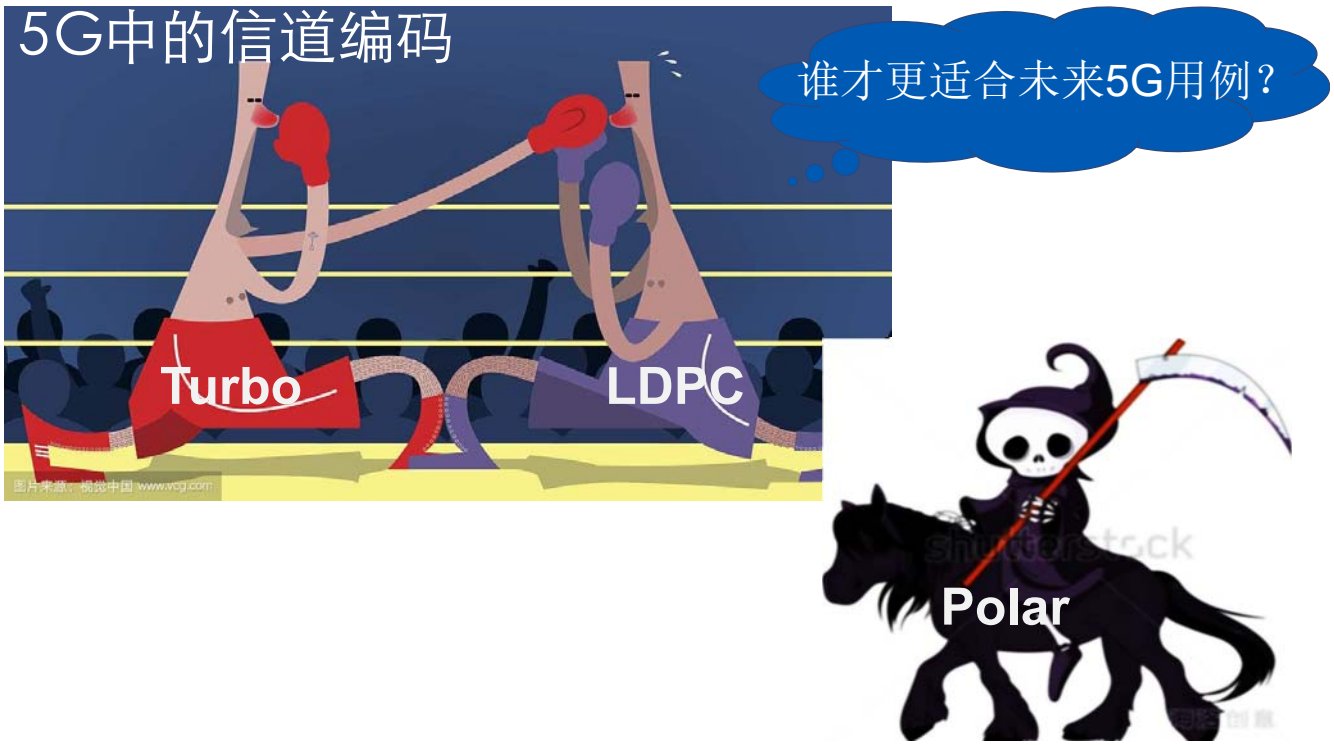
# 信道编码的发明历史-5

- 在1999年，LDPC在被人们遗忘了几十年后被重提
  - LDPC（ low-density parity check），即低密度奇偶校验码。它于1962年由Gallager提出。
  - 其基本专利到1999年就到期了
  - 而Turbo码要到2013年才到期

- LDPC利用校验矩阵的稀疏性，使得译码复杂度只与码长成线性关系，在长码长的情况下具有更简单的译码算法。
- 随着人们对 LDPC码重新进行了研究，发现LDPC 码与Turbo一样具有逼近香农极限的性能。
  - 接着，LDPC被应用于IEEE 802.11n 以及802.16的协议中。DVB-S2也决议以LDPC替代Turbo码。
  - 有人认为，LDPC是终极纠错编码，极有可能成为未来主流编码技术。



# 信道编码的今天-6



# 信道编码的今天-7

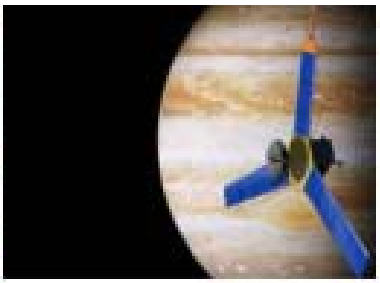
- 2007年，土耳其比尔肯大学教授E. Arikan基于信道极化理论提出的一种线性信道编码方法，即Polar码。
  - 能够达到香农限。
- Polar码的理论基础就是信道极化。
  - 信道极化包括信道组合和信道分解部分。
  - 当组合信道的数目趋于无穷大时，则会出现极化现象：一部分信道将趋于无噪信道，另外一部分则趋于全噪信道。
- 编解码的复杂性是Polar的问题，
  - 在使用改进后的SCL（Successive Cancelation List）译码算法时能以较低复杂度的代价，接近最大似然译码的性能。

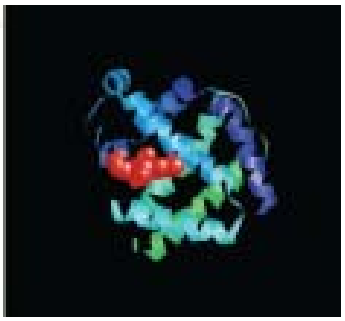


# 信道编码的未来-8

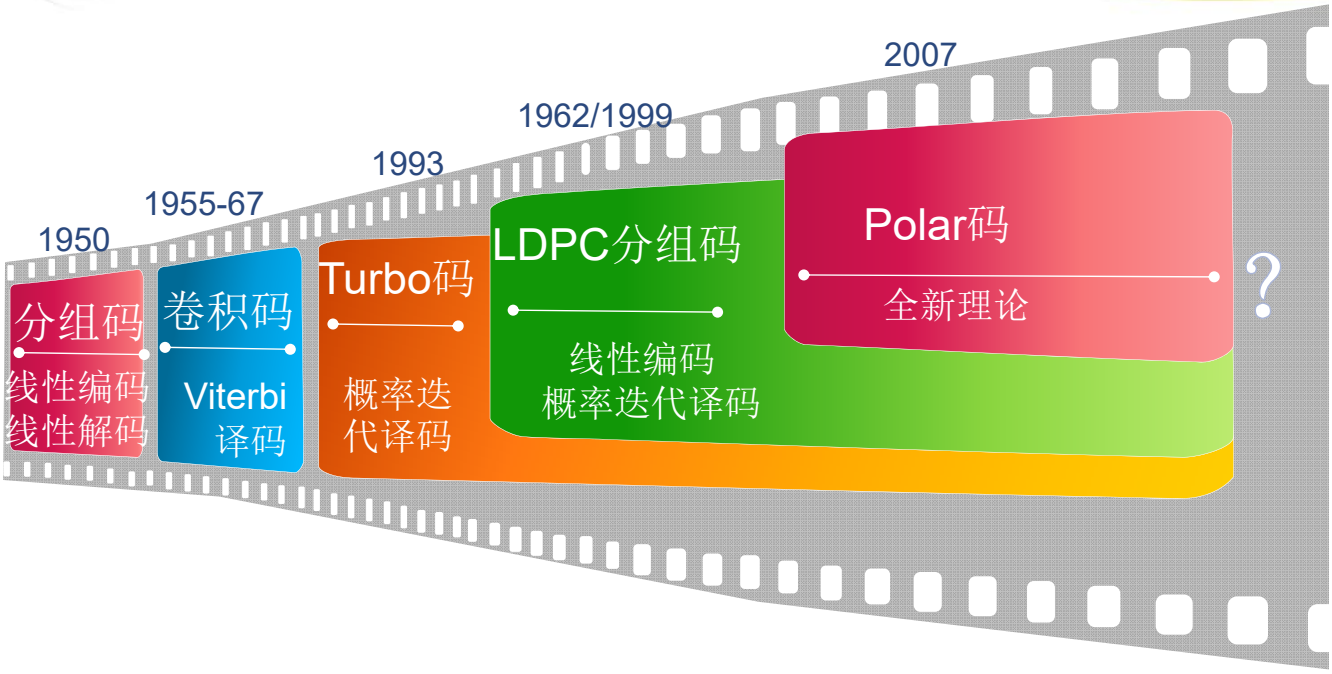
- 应用领域
  - 下一代移动通信
  - 深空通信与探测
  - 可靠计算
  - 电磁，光存储设备
  - 基因工程

基于纠错编码理论DNA序列编码特性的分析





# 简单总结






# Contents

# 数学知识预备

## 有限域

Galois Field (2)

- 我们主要讨论二进制码，使用GF(2)
- 运算规则如下

加法

$$0+0=0$$
$$0+1=1+0=1$$
$$1+1=0$$

乘法

$$0\times0=0$$
$$0\times1=1\times0=0$$
$$1\times1=1$$



# 分组码回顾

- 定义：监督码元仅与本码组的信息码元有关

  具体来说，将数字序列分段，每段根据规则映射成一个码字

  0 0 | 1 0 | 0 1 | 1 1——〉

  000 | 101 | 011 | 110

- 优点：简单并且易于分析

  容易建立系统的数学理论

  容易搭建低复杂度的编解码电路

# 检错码的例子

- 注意是检错！
- 奇偶监督码（码字长为n）

$$a_1 a_2 \cdots a_{n-1} a_n$$

根据n-1个信息码元累加，计算出最后一位码元

$$a_n = a_1 + a_2 + \cdots + a_{n-1}$$

检错方法：正确接收的码字一定满足

$$\begin{aligned} & a_1 + a_2 + \cdots + a_{n-1} + a_n \\ &= a_1 + a_2 + \ldots + a_{n-1} + a_1 + a_2 + \quad + a_{n-1} \\ &= 0 \end{aligned}$$

如果计算结果是1，说明出现了差错！

When检错失败？ 如果错偶数个码元，则刚好抵消



# 纠错码的直观表示

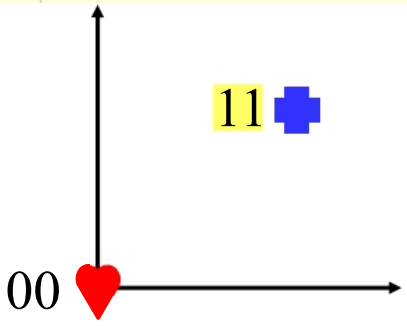
```
case 4
    %QPSK modulation
    s4=[-1-j; -1+j; 1-j; 1+j]/sqrt(2);
    c4 = [0 0; 0 1; 1 0; 1 1];
    L = length(demod_in);
    [tmp, index] = min(abs(s4(:,ones(1,L))-demod_in(ones(4,1),:)));% 最小值对应的星座点序号的二进制表示即为解调结果
    demod_out = c4(index,:);
    demod_out = reshape(transpose(demod_out),1,2*L);
```

## ❖ 距离

- 以00和11为例

Euclidean距离 $\sqrt{2}$

Hamming距离 $2$




Hamming距离的定义是，两个码字之间不同码元的个数

# Hamming距离

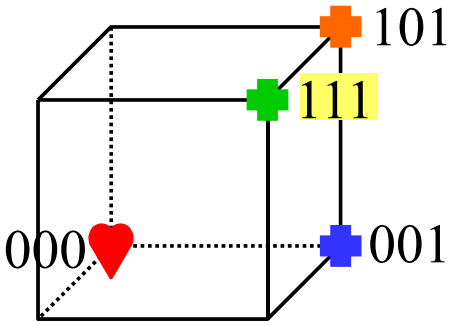
❖ 信道编码常用Hamming距离

更易于与判决、概率分析等关联起来

错成001，概率 $p_e$ ，Hamming距1

错成101，概率 $p_e^2$ ，Hamming距2

错成111，概率 $p_e^3$ ，Hamming距3




Euclidean距离也是有用的——比如网格编码调制（TCM）



# 信道编码的设计

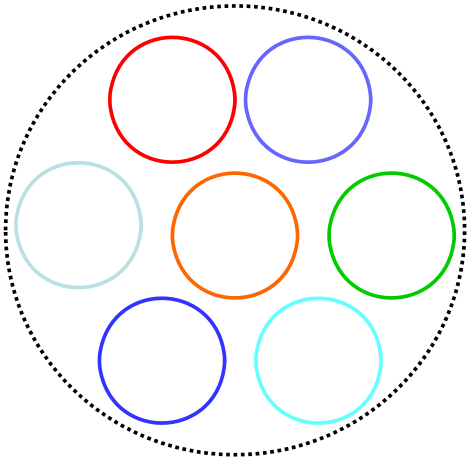
❖ 两个准则：

- 可靠性准则：最大化最小码距
- 有效性准则：给出尽可能多的许用码字

❖ 有效性

- 码率：k/n

  （k为信息码位，n是码长）
- 许用码字的数量为 $2^k$



# 线性分组码

## 线性分组码设计

- 编码过程即信息序列与生成矩阵相乘

$$\mathbf{A}=\mathbf{XG}=\mathbf{X}[\mathbf{I}:\mathbf{Q}]$$

注意：这些都是GF(2)上的运算

## 线性分组码是系统码

$$\mathbf{A}=\mathbf{X}[\mathbf{I}:\mathbf{Q}]$$

因为有单位矩阵，所以信息码元会原样出现

## Q矩阵的宽r是监督码位，线性分组码的码率是

$$\frac{k}{n}=\frac{k}{k+r}$$



# 监督矩阵H

## 还是类似的套路，从编码看解码

看HammingCode.m

- 编码 $\mathbf{A}=\mathbf{X}[\mathbf{I}:\mathbf{Q}]$

注意矩阵的维数

- 注意如下恒等式

$$\mathbf{A}\begin{bmatrix}\mathbf{Q}\\ \mathbf{I}\end{bmatrix}=\mathbf{X}[\mathbf{I}:\mathbf{Q}]\begin{bmatrix}\mathbf{Q}\\ \mathbf{I}\end{bmatrix}=\mathbf{X}(\mathbf{Q}+\mathbf{Q})=\mathbf{X}\mathbf{0}=\mathbf{0}$$

## 若以上等式无法满足，即出现了错误 H

$$\mathbf{A}\begin{bmatrix}\mathbf{Q}\\ \mathbf{I}\end{bmatrix}\neq\mathbf{0}$$

注意矩阵的维数

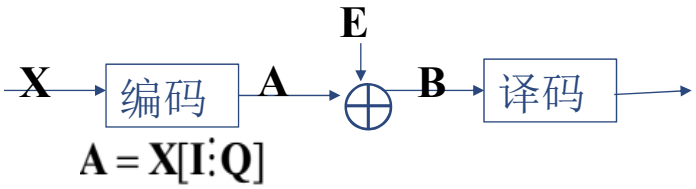
$$[\mathbf{Q}^T:\mathbf{I}]\mathbf{A}^T\neq\mathbf{0}$$

## 线性分组码如何纠错

- 线性码不仅可以检错，还能够纠错
- 解码信号可以表示为

$$\mathbf{B}=\mathbf{A}+\mathbf{E}$$

其中 $$\mathbf{E}(i)=\begin{cases}0 & a_i=b_i\\ 1 & a_i\neq b_i\end{cases}$$




- 定义校正子

$$\mathbf{BH}^T=(\mathbf{A}+\mathbf{E})\mathbf{H}^T=\mathbf{AH}^T+\mathbf{EH}^T=\mathbf{EH}^T$$



## 线性分组码如何纠错

- 基本思想，用校正子指示错误的位置，从而实现纠错
- 一个基本假设：只需要纠一个误码
  - 因为有n位码，需能指示n+1种不同情况

为什么是n+1?

  - 校正子有r位，最多能指示 $2^r$ 种情况
  - 所以，n位码中，至少需 $\lceil\log_2(n+1)\rceil$ 位校正子，亦即 $\lceil\log_2(n+1)\rceil$ 个监督位

## 线性分组码如何纠错

- 考虑最理想的情况
  - 恰好满足 $2^r = n+1$
  - 此时r位校正子，恰好指示n种误码，即错的那一个在哪位

$$s = EH^T$$

  - 若错的是第v位，则校正子为

$$\mathbf{s} = \mathbf{H}^T(v,:)$$

由此可以判断差错的位置

  - 即，监督矩阵转置后的第v行



## 线性分组码如何纠错

- 监督矩阵的设计
  - 显然，不是任何矩阵都能当监督矩阵
  - 监督矩阵需要满足：转置后各行均不相同
  - 换言之：监督矩阵各列均不相同
  - 回顾监督矩阵的转置

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{Q} \\ \mathbf{I} \end{bmatrix}$$

Q中每行至少有2个1

# 线性分组码的设计举例

- 满足 $2^r = n+1$ 的线性分组码称为Hamming码
- 下面，我们自己设计一个Hamming码
  - 取r=3，不能太大（为什么？）
  - 计算码长

$$n = 2^3 - 1 = 7$$

  - 计算信息码位：k=n-r=4
  - 确定转置监督矩阵的维度

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



# 线性分组码的设计举例

- 确定Q
  - 选择和000, 100, 010, 001不同的行构成Q
  - 如果我们希望

注意这种映射可以随便选

111表示第一位错
110表示第二位错
101表示第三位错
011表示第四位错

- 那么

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

# 线性分组码的设计举例

❖ 由监督矩阵确定生成矩阵

$$[\mathbf{I}:\mathbf{Q}]=\begin{bmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&1&0\\0&0&1&0&1&0&1\\0&0&0&1&0&1&1\end{bmatrix}$$



# Hamming码的基本性质

- 码长 $n=2^r-1$
- 监督码位 $r$
- 信息码位 $k=n-r=2^r-1-r$
- 纠错位数t=1（这是由其设计原理决定的）
- 码率 $\dfrac{2^r-1-r}{2^r-1}$

能否通过提高r，使得码率趋近于1？

- 最小码距 $d_{\min}=3$

看HammingCode.m

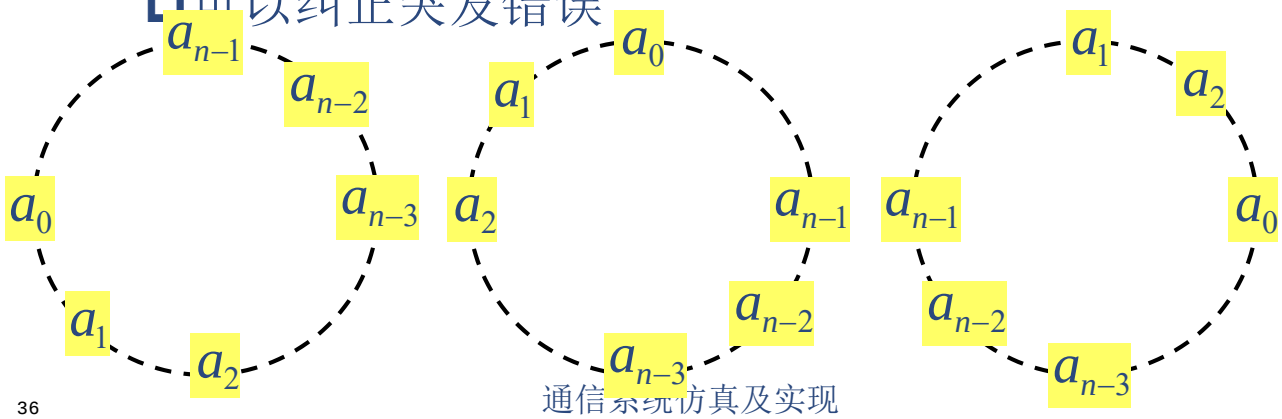
# Contents





# 循环码

- 循环码基本概念
  - 是一种线性分组码
  - 具有循环性质，若 $a_{n-1}\cdots a_1a_0$ 是许用码字，则 $a_{n-2}\cdots a_0a_{n-1}$ 也是许用码字
  - 便于用移位寄存器实现
  - 可以纠正突发错误



# 循环码示例

- 下面这个$C$是循环码：
  - 0000000　0001011　0010110　0011101
  - 0100111　0101100　0110001　0111010
  - 1000101　1001110　1010011　1011000
  - 1100010　1101001　1110100　1111111
- 这个码同时也是(7,4)汉明码

4个循环圈分别由$C_1$=[0000000], $C_2$=[0001011], $C_4$=[0011101], $C_{16}$=[1111111] 循环移位构成。



# 循环码的多项式描述

- **一个$n$元码字可以用次数不超过$n$-1的多项式唯一的表示**

$$c=\left(c_{n-1}c_{n-2}\ldots c_1c_0\right)$$

$$c\left(x\right)=c_{n-1}x^{n-1}+c_{n-2}x^{n-2}+\ldots+c_1x^1+c_0$$

  - $x$ 为哑元，无具体取值，我们仅关心其幂次数所代表的相应码元的位置
  - $c(x)$为$c$的码多项式
  - 码多项式最高次数为称为多项式的次数或阶数
- 用多项式的语言来说，循环封闭性就是

$$\text{if}\quad c(x)\in C(x)$$

$$\text{then}\quad \left[xc(x)\right]_{\bmod(x^n+1)}\in C(x)$$

$$
\begin{aligned}
x\,c(x) &= x\left(c_{n-1}x^{n-1}+c_{n-2}x^{n-2}+\cdots+c_1x+c_0\right)\\
&= c_{n-1}x^{n}+c_{n-2}x^{n-1}+\cdots+c_1x^{2}+c_0x\\
&= c_{n-1}x^{n}+c_{n-2}x^{n-1}+\cdots+c_1x^{2}+c_0x+c_{n-1}+c_{n-1}\\
&= c_{n-1}\left(x^{n}+1\right)+\left(c_{n-2}x^{n-1}+c_{n-3}x^{n-2}+\cdots+c_0x+c_{n-1}\right)
\end{aligned}
$$

- 除以$x^n+1$后取余得到

$$c_{n-2}x^{n-1}+c_{n-3}x^{n-2}+\cdots+c_0x+c_{n-1}$$

  - 对应的码字是

$$(c_{n-2},c_{n-3},\cdots,c_0,c_{n-1})$$

板书一个例子



- 集合$C(x)$中除0多项式外，次数最低的那个多项式$g(x)$称为该循环码的生成多项式
  - 唯一
  - 零次项是1
  - $C(x)$中的多项式都是$g(x)$的倍式
  - 任何$g(x)$的倍式，若次数不超过$n-1$，一定在$C(x)$中
  - $g(x)$的次数等于$n$-$k$
  - $g(x)$是$x^n+1$的一个因式

$$x\,c(x)=c_{n-1}\left(x^{n}+1\right)+\left(c_{n-2}x^{n-1}+c_{n-3}x^{n-2}+\cdots+c_0x+c_{n-1}\right)$$

## 循环码的生成矩阵G

❖对于n=7

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

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

■ 例：（7, 3）循环码的生成矩阵

$$G'=\begin{pmatrix}1&0&0&1&1&1&0\\0&1&0&0&1&1&1\\0&0&1&1&1&0&1\end{pmatrix}\xrightarrow[c2+c3\to c2]{c1+c2+c3\to c1}G=\begin{pmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\0&0&1&1&1&0&1\end{pmatrix}$$

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



## 循环码的编码

❖一般的，对k个信息码元分组进行编码可表示为

- $g(x)$ 取自$x^n+1$的一个因式

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

$$c(x)=uG=(u_{k-1}u_{k-2}\cdots u_0)\begin{bmatrix}x^{k-1}g(x)\\x^{k-2}g(x)\\\vdots\\xg(x)\\g(x)\end{bmatrix}=[u_{k-1}x^{k-1}+u_{k-2}x^{k-2}+\cdots+u_0]g(x)$$

每个码字都能被生成多项式整除

# 系统循环码的译码

## ❖监督矩阵

- 先求 $h(x)=(x^n+1)/g(x)$
- 再求$h(x)$的互反多项式 $h^*(x)$
- 根据 $h^*(x)$，对齐循环移位写出监督矩阵H

看代码CyclicCode.m

## ❖译码过程

- 与线性分组码译码相同，先求校正子$s(x)$，再确定错误图样$e(x)$，最后根据错误图样纠正错误。
- 错误图样$e(x)$的确定由于循环特性而简化实现电路



# BCH码、RS码

## ❖BCH码和RS码都属于循环码

- 对于较短的码长，BCH是目前已知的非常好的码
- RS码是多进制BCH码的一个特例
  - 所谓多进制的意思是：编码中的符号不是0、1，而是例如0、1、2、3这样的四进制符号。同时遵循GF(4)的算术规则
  - 对于高码率（$k/n$大）及擦除信道，RS经常是首选
    - 擦除信道：数据或者正确到达，或者丢失。

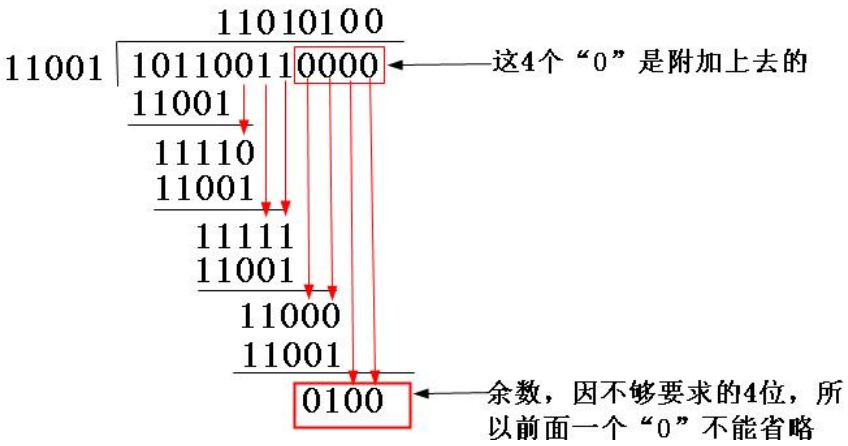
# 循环冗余校验码CRC

- 是一种检错编码
- 思路
  - 循环码的码字多项式能被生成多项式整除。如果收到的码字不能被$g(x)$整除，则可以认为有错误bit。
  - 发送码字只需要满足是$g(x)$倍数这个要求，不一定要有循环封闭性，不限制是 $x^n+1$的因子。
  - 不一定是循环码
  - 一般用系统码形式，校验码的位数比生成多项式的位数少1



# CRC举例

- 生成多项式 $g(x)=x^4+x^3+1$，二进制比特串为11001
- 待编码序列是10110011




- 把余数作为校验位，即编码序列为101100110100
- 接收端，用接收序列除以生成多项式，不整除即错。