

# 第 5 章 纠错编码

计算机科学与技术学院 孙伟平

## 第5章

5.1 纠错编码的基本概念

5.2 线性分组码

5.3 循环码

5.4 卷积码

# 基本内容

在信息可以量度的基础上，  
研究有效地和可靠地传递  
信息的科学。

数学模型？  
怎样编码有效？  
怎么实现？

信源

数学模型？  
信息测度？

信源编码

加密

信道编码

数学模型？  
能传递多少信息？  
信道容量？

信道

数学模型？  
怎样保证传输可靠？  
怎么实现？

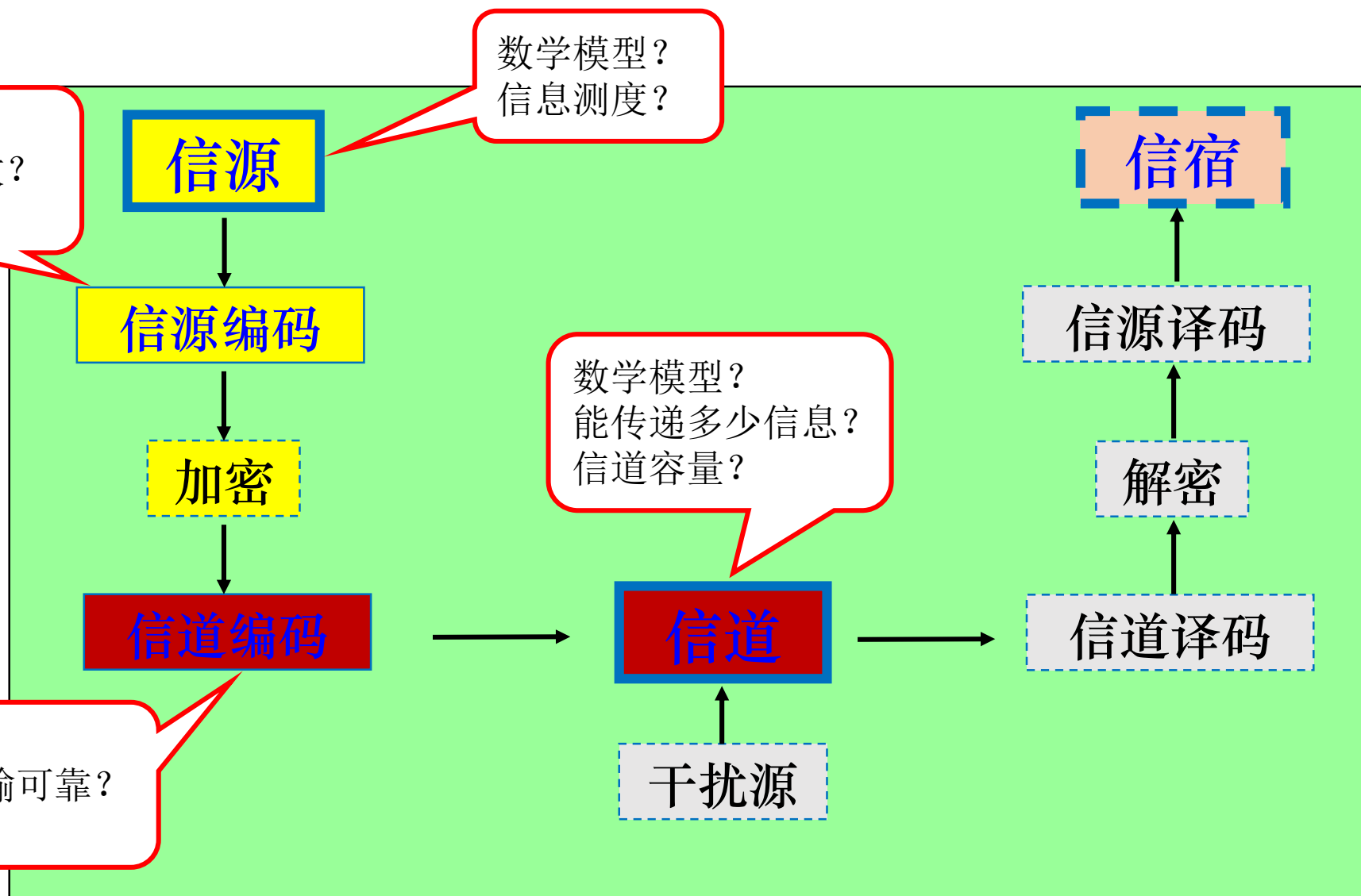
干扰源

信宿

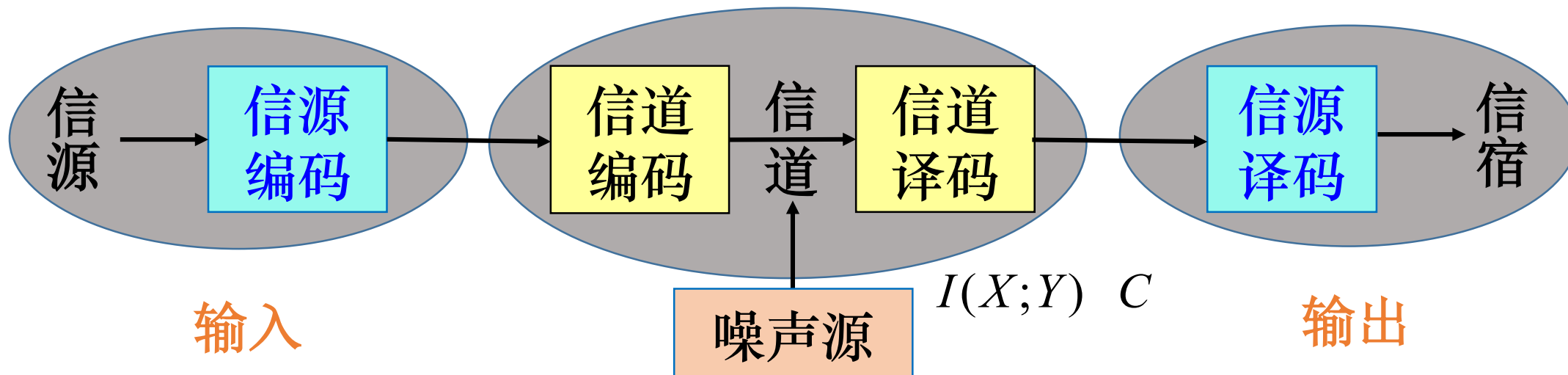
信源译码

解密

信道译码

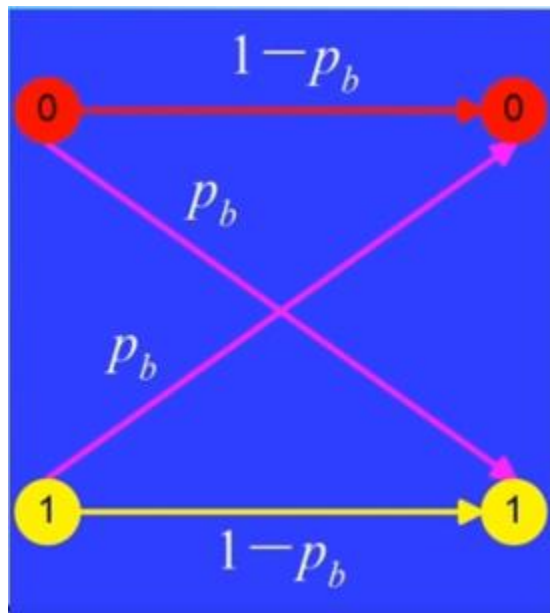


## 信道编码器在通信系统中的位置



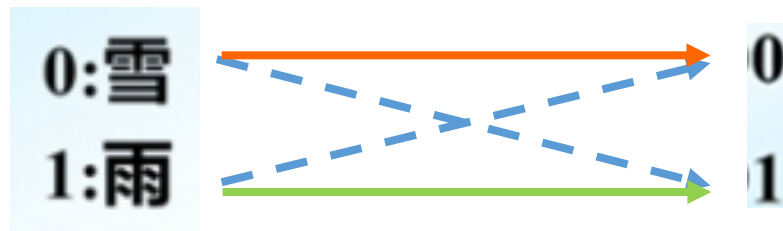
$$I(X;Y) = H(X) - H(X|Y) \quad \text{提高信息传输时的抗干扰能力}$$

- **目的** 增加信息传输的可靠性
- **手段** 增加信息冗余度
- **名称** 信道码、数据传输码、差错控制码



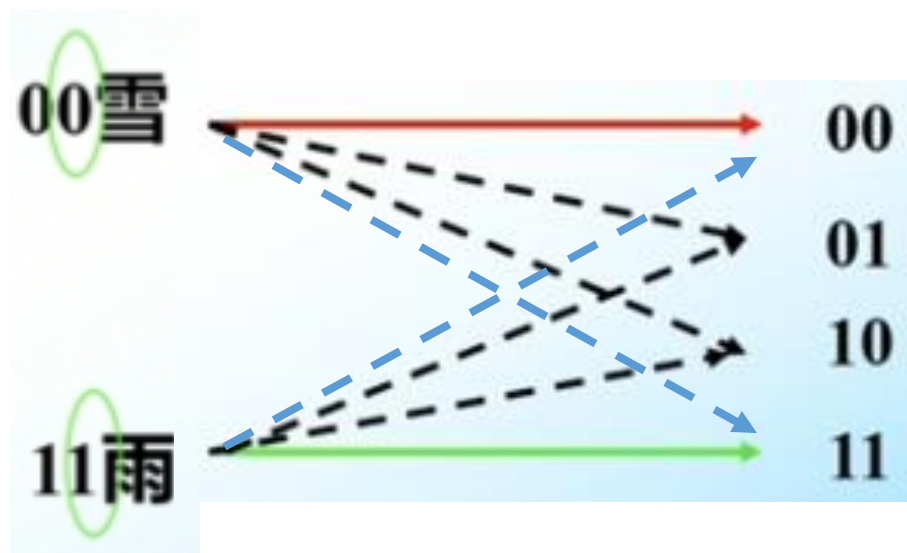
$$p_b = 0.01$$

记译码错误概率为  $p_E$



若  $1 \rightarrow 0, 0 \rightarrow 1$ , 收端无法发现错误

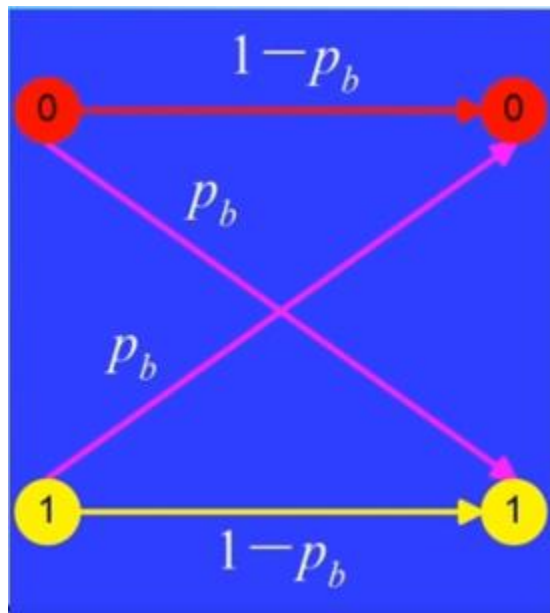
$$p_E = 0.01$$



} 禁用码组

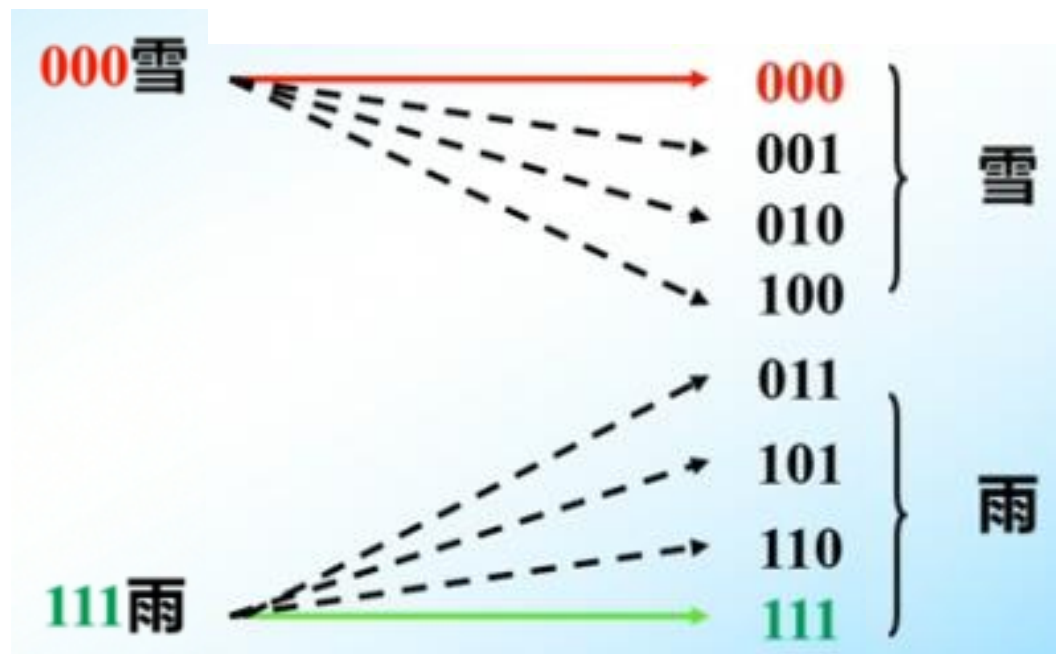
具有检出1位错码的能力,但不能予以纠正。

$$p_E = 10^{-4}$$



$$p_b = 0.01$$

记错误译码概率为  $p_E$



采用ARQ方式  $p_E = 10^{-6}$

采用FEC方式  $p_E \approx 10^{-4}$

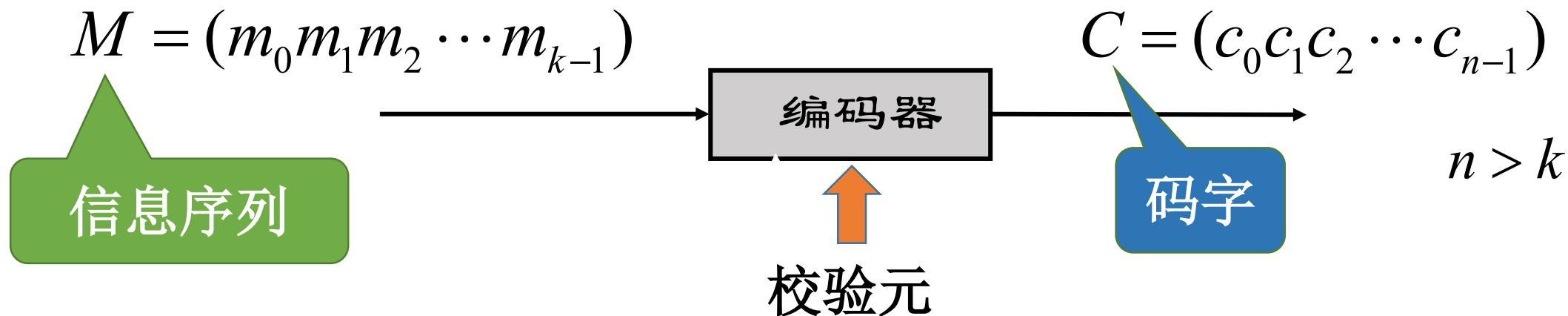
在只有1位错码的情况下,可以判决哪位是错码并予以纠正; 可以检出2位或2位以下的错码。

单个的字无法检错：扌→？

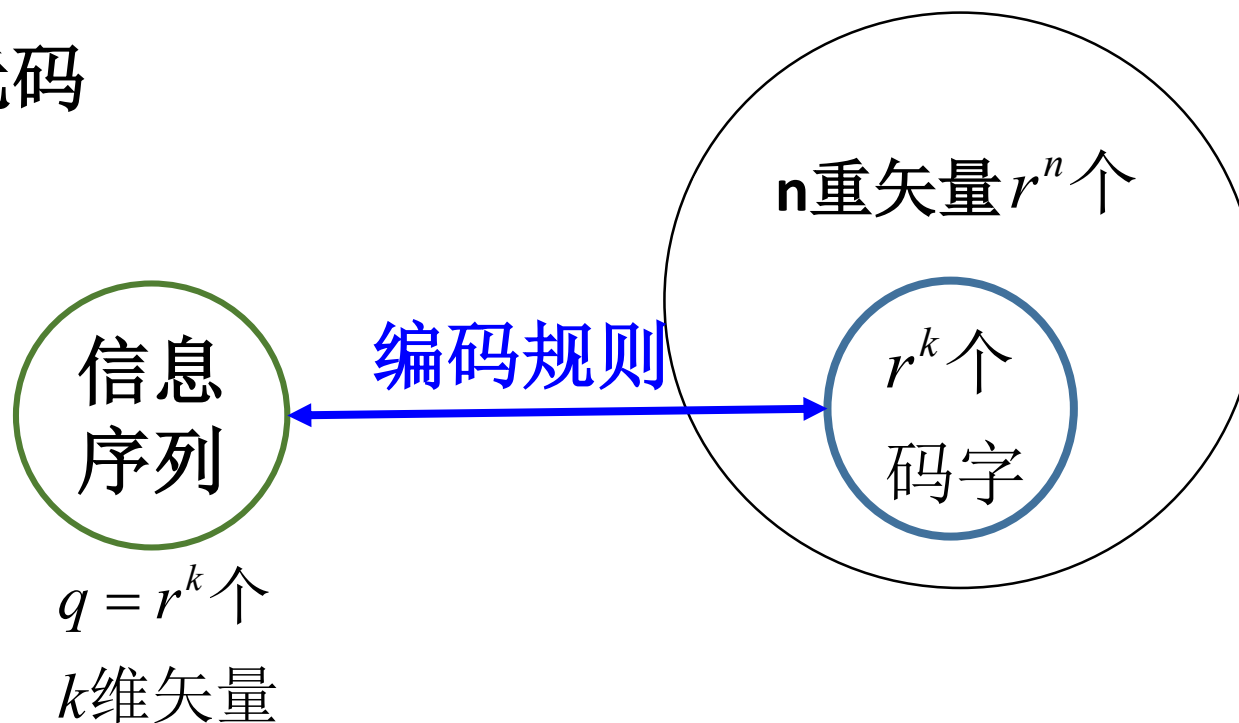
词汇能够检错：我们扌的→我扌的

词汇能够纠错：我们扌的→我们的，我等的，我辈的，我班的，...

结论：加入冗余后，根据词汇的概率分布稀疏性可以用来检错和纠错。



假设采用r元码





## 5.1.1 差错控制系统模型及分类

### 1 前向纠错方式



图5.1.1 FEC方式差错控制系统模型

### 2 反馈重传方式



图5.1.2 ARQ方式差错控制系统

### 3 混合纠错方式

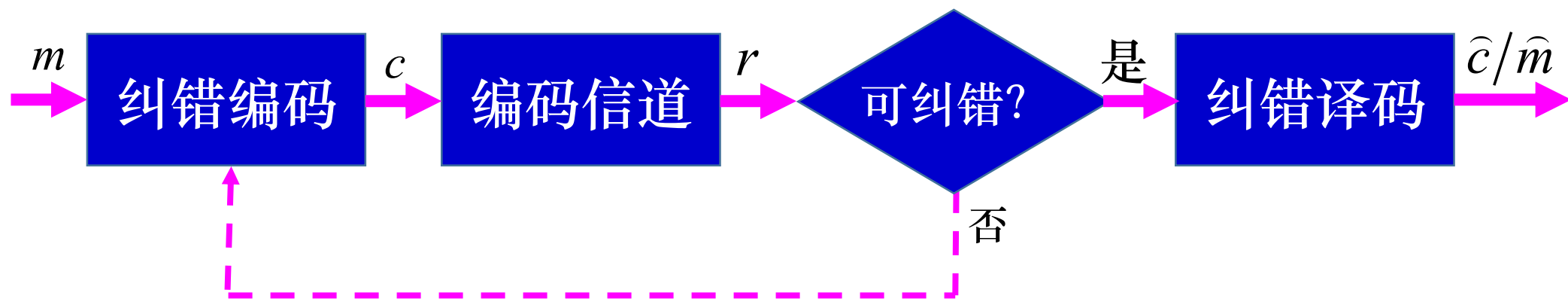


图5.1.3 HEC方式差错控制系统

## 5.1.2 纠错编码分类



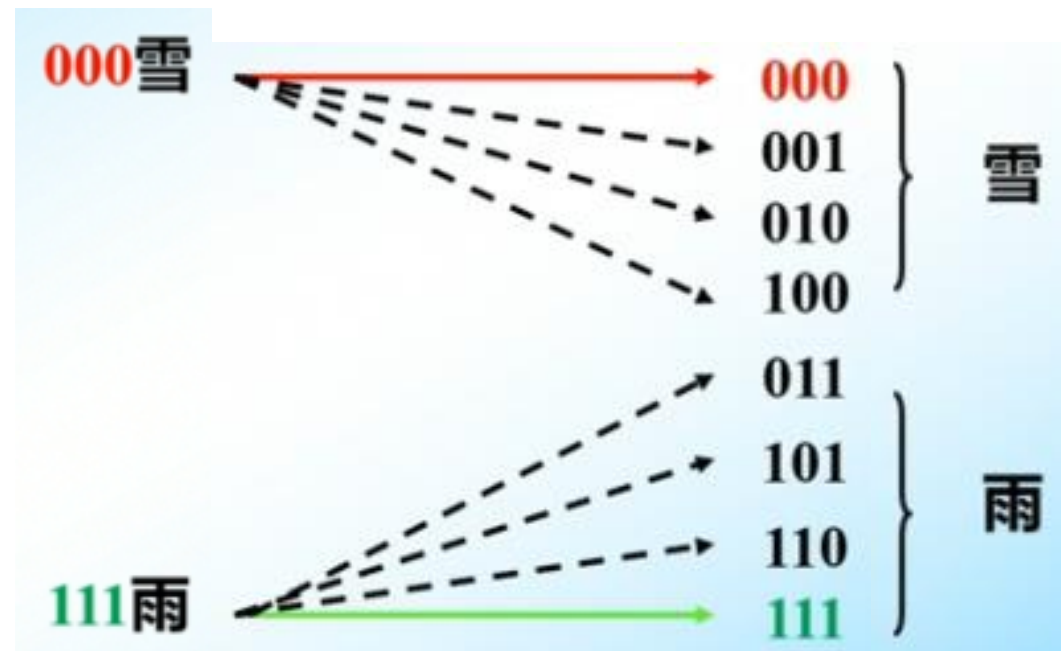
线性分组码 —— 群码

## 5.1.3 分组码的译码准则

**目的：**降低错误译码概率 $P_E$ 。

**对象：**信息序列（设码元间彼此无关且等概出现）。

**方法：**在传输的信息码之中按一定规律产生一些附加码元，经信道传输，在传输中若码字出现错误，收端能利用编码规律发现码的内在相关性受到破坏，从而按一定的译码规则自动纠正或发现错误，降低误码率。



译码准则就是分类准则，即当信道的输出为  $r$  时，  
将其译为哪个码字  $c$  最合理？

最小差错概率准则

$$\min p_E$$

最大后验概率准则

$$\max_{i=1,2,\dots,2^k} p(C_i / R) \quad p(C_i / R) = \frac{p(C_i)p(R / C_i)}{p(R)}$$

最大似然译码准则

$$\max_{i=1,2,\dots,2^k} p(R / C_i)$$

最小距离译码准则

# 最大似然译码准则

1. 它不要消息先验概率。
2. 在消息先验概率等概条件下，它等价于最大后验概率译码，因而也是最佳的。但若消息先验概率不确定时，采用最大似然译码不一定保证译码错误概率最小。
3. 实际系统中，信源发出的序列传送到信道之前都已进行信源编码，经过有效的信源编码，输出码元的概率分布会均匀化，所以信道的输入近似等概，因此工程应用中常常采用最大似然译码。

## 练习题

## 5.1.4 信道编码定理

(香农第二编码定理)

若一离散平稳无记忆信道，其容量为 $C$ ，输入序列长度为 $L$ ，只要待传送的信息率  $R < C$ ，总能找到一种编码。当 $L$ 足够长时，译码差错概率  $p_e < \varepsilon$ 。

$\varepsilon$  为任意正数。反之，当 $R > C$ ，任何编码的 $p_e$ 必大于0，当  $L \rightarrow \infty, p_e \rightarrow 1$

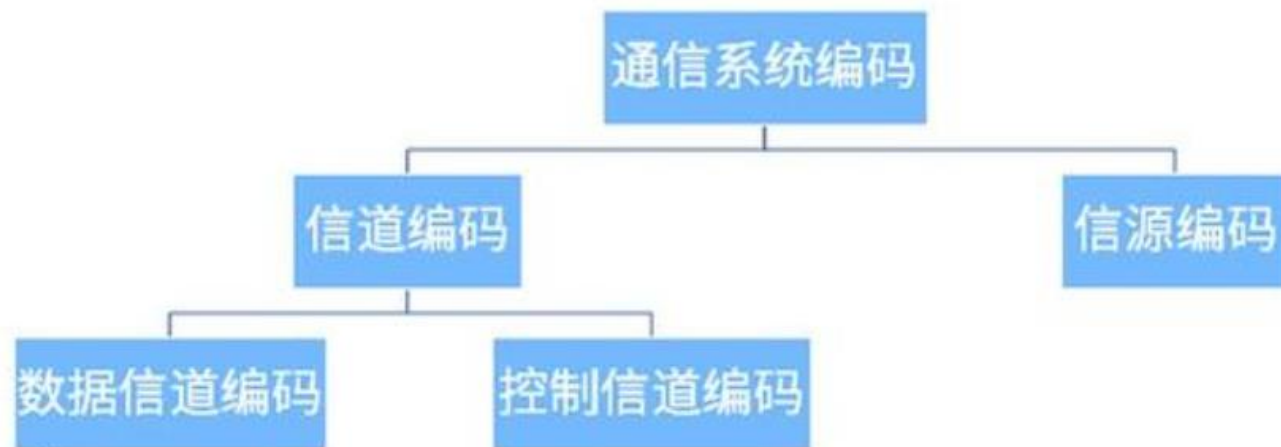
**结论：**信道容量 $C$ 是一个临界值。

对离散无记忆平稳信道，信息在有干扰信道中传输时，只要信息传输率不超过这个临界值，信道就可几乎无失真地把信息传送过去，否则就会产生失真。

1993年，日内瓦IEEE通信国际会议，法国电机工程师C.Berrou和A.Glavieux声称发明了接近香农极限的编码方法——**Turbo**码

1999年，人们重燃了对**LDPC**的兴趣。LDPC于1962年由Gallager提出，然后被人们遗忘了几十年。直到Turbo码被提出后，人们才发现它从某种角度上说也是一种LDPC码

2007年，土耳其比尔肯大学教授E.Arikan提出**Polar**码，该码字是迄今发现的唯一一类能够达到香农极限的编码方法





## 第5章

5.1

纠错编码的基本概念

5.2

线性分组码

5.3

循环码

5.4

卷积码

## 5.2.1 线性分组码的基本概念

### 信道编译码方法的最初范例

**基本思路：** 将码字分成两段



**信息位：** 含有信源信息的比特位(红色部分)，长度用 $k$ 表示。

**校验位：** 按一定规则添加的码位。用于监督码组内部码元的关联关系，可发现或纠正传输错误(蓝色部分)，亦称监督位。

**码 字：** 信息位和校验位组成的相对独立码组(红色和蓝色整体)，长度用 $n$ 表示。

**汉明重量：** 码字中非零码元的个数，亦称码重。

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

汉明重量(码重)= 5

**汉明距离：** 两个码字相应码元取不同数值的码元数，也称码距。

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

1	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---

汉明距离(码距)= 4

两个等长码字c和c'之间的汉明距离可用公式表示为

$$d(c, c') = \sum_{i=1}^n c_i \oplus c'_i, \quad c_i \neq c'_i$$

**码的最小距离：**任意两码字之间的最小汉明距离，简称最小码距。

$$d_{\min} = \min_{c \neq c'} d(c, c')$$



0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0

1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(4,3)偶校验码的最小码距=2

**分组码：** 将信息序列以 $k$ 为长度进行等长划分，然后按编码规则添加一定数目冗余位的编码方法，冗余位亦称校验位。

**线性分组码：** 信息位和校验位之间满足线性关系的分组码。

为评估编码效率，定义 **码率**： $R = \frac{k}{n}$

总结线性分组码的基本参数如下：

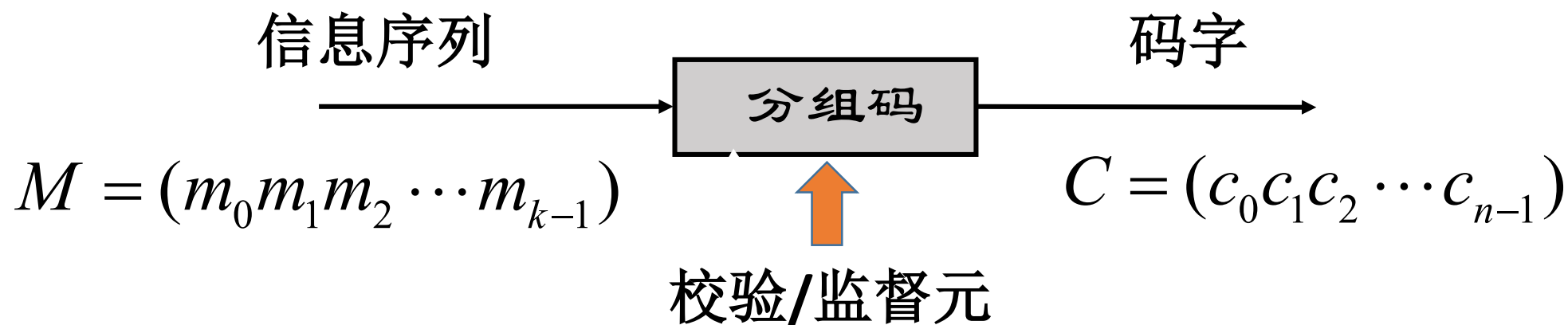
码 长： $n$

信息位长： $k$

码 字 数： $M$

监督位长： $r$

最小码距： $d_{min}$



$$\{M\} \xrightarrow{f(\cdot)} \{C\}$$

线性映射

$$f(\alpha M + \beta M') = \alpha f(M) + \beta f(M')$$

其中  $\alpha, \beta \in GF(2) = \{0, 1\}$ ,  $M, M' \in \{M\}$

信息	码字
00	11100
01	01011
10	10110
11	01101

$$\begin{cases} c_0 = m_0 \\ c_1 = m_1 \\ c_2 = m_0 \oplus m_1 \\ c_3 = \bar{m}_0 \\ c_4 = \bar{m}_1 \end{cases}$$

信息	码字
00	00011
01	01110
10	10101
11	11000

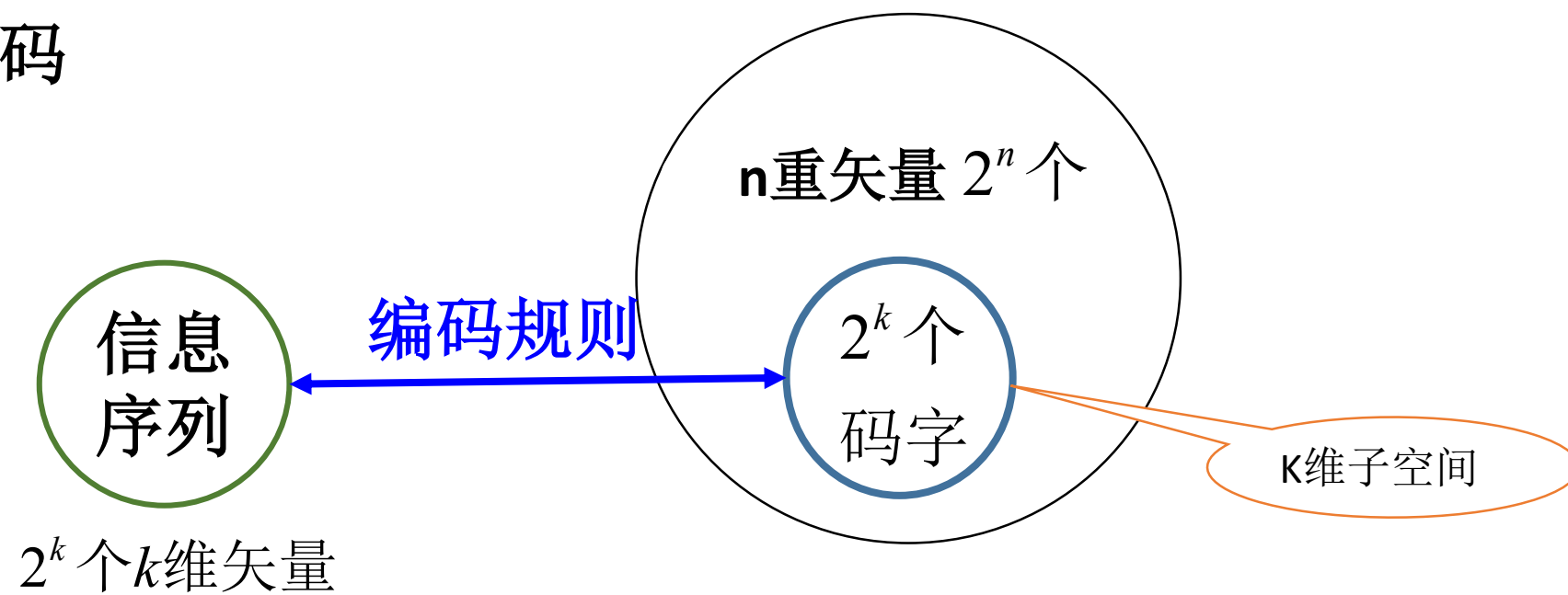
$$\begin{cases} c_0 = m_1 \\ c_1 = m_0 \oplus m_1 \\ c_2 = m_0 \\ c_3 = m_1 \\ c_4 = m_0 \oplus m_1 \end{cases}$$

信息	码字
00	00000
01	11011
10	01101
11	10110

$$\begin{cases} c_0 = m_0 \oplus m_1 \\ c_1 = m_0 \\ c_2 = m_1 \\ c_3 = m_0 \\ c_4 = m_1 \end{cases}$$

信息	码字
00	00000
01	10101
10	11010
11	01111

设采用二元码





# 矢量空间与基底

- 一组线性无关的矢量  $V_1, V_2, \dots, V_n$ ，它们的线性组合构成了一个**矢量空间** $\mathbf{V}$ ，这组矢量  $V_1, V_2, \dots, V_n$  就是这个矢量空间的**基底**。
- $n$ 维矢量空间应包含 $n$ 个基底
- **基底不是唯一的**，例：线性无关的两个矢量 $(1,0)$ 和 $(0,1)$ 以及 $(-1,0)$ 和 $(0,-1)$ 可张成同一个二维空间。

# 二元域GF(2)上三重矢量空间

- 以(100)为基底可张成一维三重子空间 $V_1$ ，含 $2^1=2$ 个元素，即

$$V_1 = \{(000), (100)\}$$

- 以(010)(001)为基底可张成二维三重子空间 $V_2$ ，含 $2^2=4$ 个元素，即

$$V_2 = \{(000), (001), (010), (011)\}$$

- 以(100)(010)(001)为基底可张成三维三重空间 $V$ ，含 $2^3=8$ 个元素， $V_1$ 和 $V_2$ 都是 $V$ 的子空间。

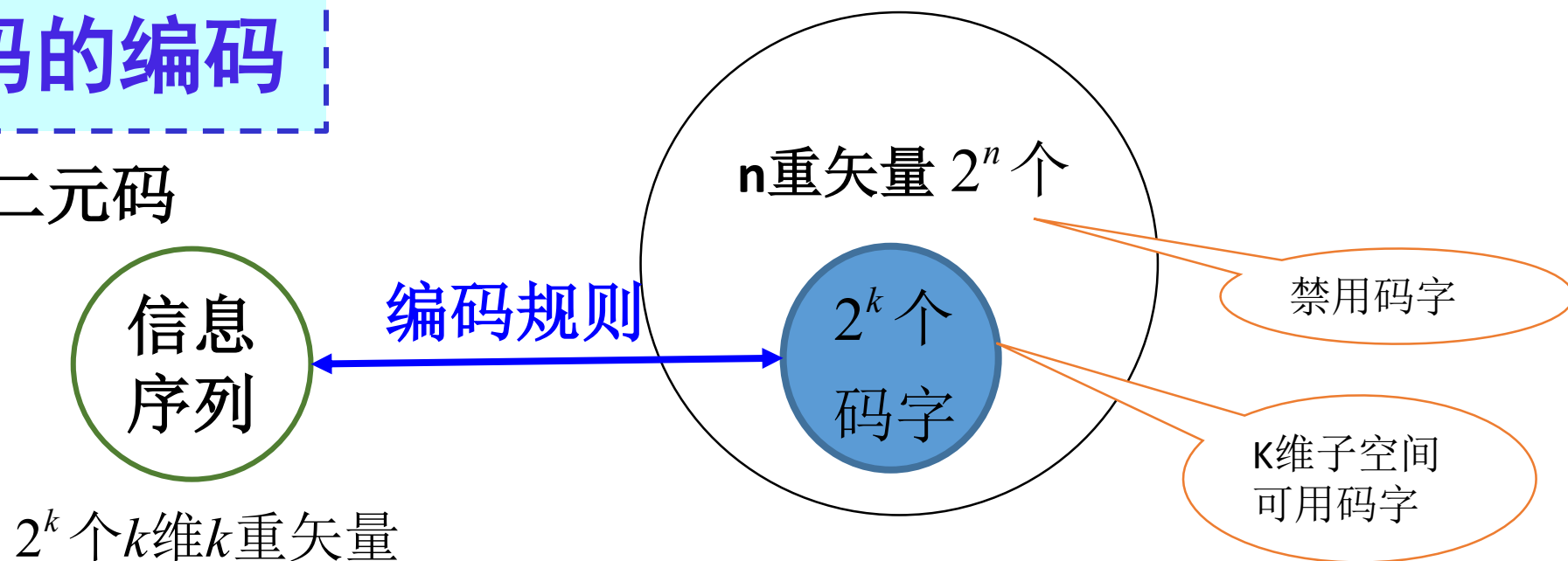
$$V_3 = \{(000), (001), \dots, (111)\}$$

# 矢量空间

- 每个矢量空间或子空间中必然包含零矢量
- 两个**矢量正交**:  $\mathbf{v}_1 \cdot \mathbf{v}_2 = 0$
- 两个**矢量空间正交**: 某矢量空间中的任意元素与另一矢量空间中的任意元素正交
- 正交的两个子空间 $\mathbf{v}_1$ 、 $\mathbf{v}_2$ 互为**对偶空间 (Dual Space)**, 其中一个空间是另一个空间的**零空间** (null space, 也称零化空间)。

## 5.2.2 线性分组码的编码

设采用二进制



线性分组编码的任务：

- 选择一个  $k$  维  $n$  重子空间作为码空间。
- 确定由  $k$  维  $k$  重信息空间到  $k$  维  $n$  重码空间的映射方法。

码空间的不同选择方法，以及信息组与码组的不同映射算法，就构成了不同的分组码。

$$\begin{cases} c_0 = m_1 \\ c_1 = m_0 \oplus m_1 \\ c_2 = m_0 \\ c_3 = m_1 \\ c_4 = m_0 \oplus m_1 \end{cases}$$

信息	码字
00	00000
01	<u>11011</u>
10	<u>01101</u>
11	10110

(5,2)线性分组码

$$\begin{cases} c_0 = m_0 \oplus m_1 \\ c_1 = m_0 \\ c_2 = m_1 \\ c_3 = m_0 \\ c_4 = m_1 \end{cases}$$

信息	码字
00	00000
01	<u>10101</u>
10	<u>11010</u>
11	01111

(5,2)线性分组码

$$(c_0 c_1 c_2 c_3 c_4)_{1 \times 5} = (m_0 \ m_1)_{1 \times 2} \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}_{2 \times 5}$$

生成矩阵

$$(c_0 c_1 c_2 c_3 c_4)_{1 \times 5} = (m_0 \ m_1)_{1 \times 2} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}_{2 \times 5}$$

生成矩阵

# 生成矩阵

设 $\mathbf{m}$ 为 $k$ 维信息矢量， $\mathbf{c}$ 为 $n$ 维码矢量，一般线性分组码可以 $(n,k)$ 两个参数表示。线性分组码的编码规则很简单：

$$\mathbf{c} = \mathbf{m}\mathbf{G}$$

式中 $\mathbf{G}$ 是 $k$ 行 $n$ 列( $n \geq k$ )的秩为 $k$ 的矩阵，称为生成矩阵。

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

对于二元编码， $\mathbf{c}$ 和 $\mathbf{m}$ 都是二元向量， $\mathbf{G}$ 为GF(2)上的  $k \times n$ 矩阵， $g_{i,j} \in \{0,1\}$ ，向量与矩阵之间，矩阵与矩阵之间均为模2运算。

## 生成矩阵 $G(k \times n)$ 的特点

- 想要保证  $(n,k)$  线性分组码能够构成  $k$  维  $n$  重子空间， $G$  的  $k$  个行矢量  $g_0, g_1, \dots, g_{k-1}$  必须是线性无关的。
- 由于基底不是唯一的，所以  $G$  也不是唯一的。
- 不同的基底有可能生成同一码集，但因编码涉及码集和映射两个因素，若码集一样而映射方法不同，则不能说是同样的码。

**例5.2.1** (4,3)偶校验码是一个(4,3)线性分组码，其生成矩阵

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \text{ 当 } \mathbf{m} = (m_0, m_1, m_2) = (101) \text{ 时, 求其码字。}$$

解:

$$(c_0 c_1 c_2 c_3) = (101) \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (1010)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \quad (c_0 c_1 c_2 c_3) = (m_0 m_1 m_2) \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = m_0 \mathbf{g}_0 \oplus m_1 \mathbf{g}_1 \oplus m_2 \mathbf{g}_2$$

码字 $c$ 是 $G$ 的行向量 $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}$ 的线性组合



### 例5.2.1

(4,3)偶校验码是一个(4,3)线性分组码，其生成矩阵

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \text{ 当 } \mathbf{m} = (m_0, m_1, m_2) = (101) \text{ 时, 求其码字。}$$

解：

$$(\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) = (101) \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (1010)$$

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

单位矩阵

系统码

### 例5.2.1

(4,3)偶校验码是一个(4,3)线性分组码，其生成矩阵

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \text{ 当 } \mathbf{m} = (m_0, m_1, m_2) = (101) \text{ 时, 求其码字。}$$

解:

$$(\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) = (101) \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (1010)$$

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

这两个生成矩阵  
张成的码空间是  
一样的

$$(\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) = (101) \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (0110)$$

但信息与码字的  
映射关系不一样

# 例5.2.1

(4,3)偶校验码是一个(4,3)线性分组码，其生成矩阵

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \text{ 当 } \mathbf{m} = (m_0, m_1, m_2) = (101) \text{ 时, 求其码字。}$$

解:

$$(\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) = (101) \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (1010)$$

验证  $\mathbf{GH}^T = \mathbf{0}$

$$(\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) = (\mathbf{m}_0 \mathbf{m}_1 \mathbf{m}_2) \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} = (\mathbf{m}_0 \quad \mathbf{m}_1 \quad \mathbf{m}_2 \quad \mathbf{m}_0 + \mathbf{m}_1 + \mathbf{m}_2)$$



$$\mathbf{H} = (1111) \leftarrow (\mathbf{c}_0 \mathbf{c}_1 \mathbf{c}_2 \mathbf{c}_3) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 0 \leftarrow \underbrace{c_0 + c_1 + c_2 + c_3 = 0}_{\text{一致校验方程}} \leftarrow c_3 = c_0 + c_1 + c_2$$

一致校验矩阵

一致校验方程

当 $G$ 给定时，线性分组码有如下**性质**：

- 1 零向量 $\theta = (0, 0, \dots, 0)$ 一定是一个码字，称为零码字；
- 2 任意两码字的和或差仍是一个码字； **封闭性**
- 3 任意码字 $c$ 是 $G$ 的行向量 $g_0, g_1, \dots, g_{k-1}$ 的线性组合；
- 4 线性分组码的最小距离等于最小非零码字重量，即

$$d_{\min} = \min_{c \neq \theta} w(c)$$

最小码距  $\longleftrightarrow$  最小码重

有何好处？

# 一致校验矩阵

对于GF(2)上的 $k \times n$ 矩阵 $G$ ，存在 $(n-k) \times n$ 矩阵 $H = \begin{bmatrix} h_{0,0} & \cdots & h_{0,n-1} \\ \vdots & \ddots & \vdots \\ h_{n-k-1,0} & \cdots & h_{n-k-1,n-1} \end{bmatrix}$ ,

使得

$$GH^T = [0]_{k \times (n-k)}$$

$H$ : 一致校验矩阵。 $H^T$ 为 $H$ 的转置。且有

$$cH^T = mGH^T = m[0]_{k \times (n-k)} = \theta$$

$\theta$ 为 $(n-k)$ 维零向量。

用来校验接收到的码字  
是否正确

如果生成矩阵 $G$ 具有形式

$$G_s = \begin{bmatrix} I_k & Q_{k \times (n-k)} \end{bmatrix}$$

则称该码为**系统码**，其中 $I_k$ 为 $k \times k$ 单位阵。

系统码的一致校验矩阵 $H_s$

$$H_s = \begin{bmatrix} Q^T & I_{(n-k)} \end{bmatrix}_{(n-k) \times n}$$

$G_s$ 与 $H_s$ 满足  $G_s H_s^T = [0]_{k \times (n-k)}$

对二元码来说， $H = H_s$

$G$ 和 $H$ 的角色可以互换，即 $H$ 作为生成矩阵， $G$ 为校验矩阵。由 $H$ 生成的码字称为 $c$ 的**对偶码**。

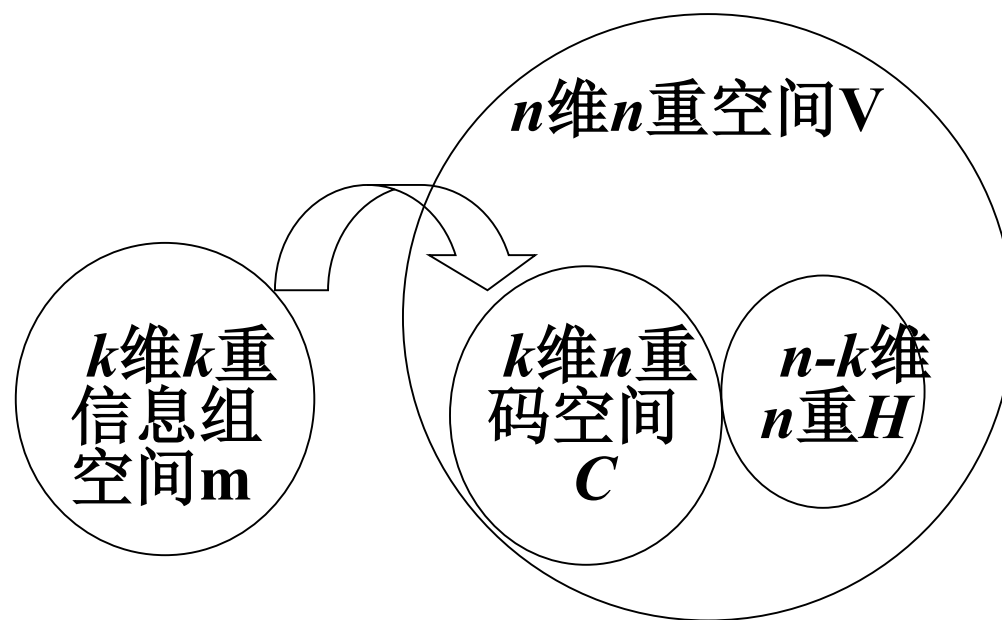
**系统码**的前 $k$ 位就是信息位。

**系统码**的一致校验矩阵好求。

# 深入理解G和H

- $n$ 维 $n$ 重空间有相互正交的 $n$ 个基底
- 选择 $k$ 个基底构成G，生成码空间C
- 选择另外的 $(n-k)$ 个基底构成H
- C和H是对偶的

$$CH^T=0, GH^T=0$$



例 设二元(7,4) 码的生成矩阵为

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

求其一致校验矩阵 $H$ 。



例 设二元(5,3) 码的生成矩阵为

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

求其一致校验矩阵 $\mathbf{H}$ 。

解：先求其系统码生成矩阵。

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

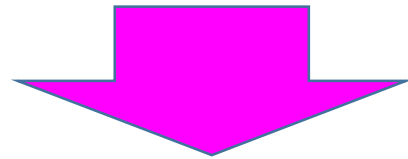


$$\mathbf{H}_s = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} = \mathbf{H} \quad \text{验证 } \mathbf{GH}^T$$

**例5.2.3** 一个(5,3)线性分组码的生成矩阵  $G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$ 。求对应的  $G_s$  和  $H_s$ 。

对  $G$  进行初等变换，用  $R_i$  表示  $G$  的第  $i$  行

$$R_3 \leftarrow R_2 \oplus R_3, R_1 \leftarrow R_1 \oplus R_3, R_1 \leftrightarrow R_3$$



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

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

验证：

$$G_s H_s^T = [0]_{3 \times 2}$$

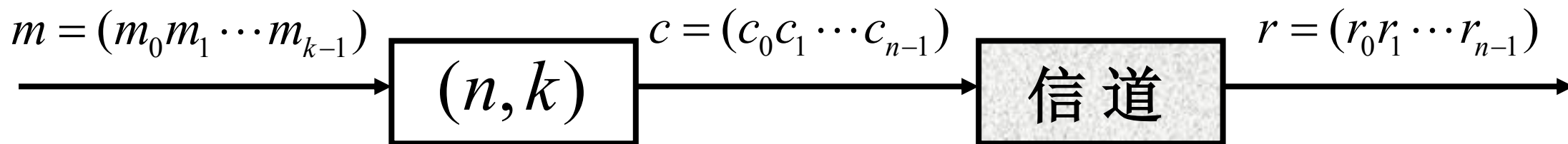
$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad G_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad H_s = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

### (5,3)线性分组码码字

消息 $m$	000	001	010	011	100	101	110	111
$G$ 生成 码字	00000	11010	01011	10001	10110	01100	11101	00111
$G_s$ 生成 码字	00000	00111	01011	01100	10001	10110	11010	11101
对偶码 码字			00000	11101	01110	10011		

- $G$ 与 $G_s$  两个生成矩阵生成的码字集合相同，消息与码字的对应关系不同。
- 由 $G_s$ 生成的码字中，前3位就是对应的消息。

## 5.2.3 线性分组码的译码



定义 **差错图案/图样** $e$  来定量描述出现的差错:

$$e = (e_0 e_1 \cdots e_{n-1}) = r - c = (r_0 - c_0, r_1 - c_1, \cdots, r_{n-1} - c_{n-1})$$

错误图案中某位为“1”表明该位出错。

二进制码中模2加与模2减等价，故一般记成

$$e = c \oplus r \text{ 或 } c = r \oplus e$$

译码器从接收矢量中确定错误图案 $e$ ，进而得到码字的估计值。若估计值正确则译码正确，否则译码错误。

## 译码准则：

最小差错概率准则  
最大后验概率准则  
最大似然译码准则  
最小距离译码准则

## 译码方法：

- 伴随式译码
- 标准阵列译码

# 伴随式译码

因为  $cH^T = 0$

所以  $rH^T = (c \oplus e)H^T = cH^T \oplus eH^T = eH^T$

如果收码无误：必有  $r=c$  则  $e=0$ ,

且  $rH^T = eH^T = 0$

如果收码有误：即  $e \neq 0$ ,

则  $rH^T = eH^T \neq 0$

在  $H^T$  固定的前提下， $rH^T$  仅仅与差错图案  $e$  有关，而与发送码  $c$  无关。定义伴随式  $s$

$$s = (s_0, s_1, \dots, s_{n-k-1}) = rH^T = cH^T \oplus eH^T = eH^T$$

码字通过信道传输可能出错，接收矢量可表示成码矢量和差错图案的线性叠加。

**接收矢量：**  $\mathbf{r} = (r_0, r_1, \dots, r_{n-1})$

**码矢量：**  $\mathbf{c} = (c_0, c_1, \dots, c_{n-1})$

**差错图案：**  $\mathbf{e} = (e_0, e_1, \dots, e_{n-1})$

**伴随式：**  $\mathbf{s} = (s_0, s_1, \dots, s_{n-k-1})$

$$\mathbf{s} = \mathbf{rH}^T = \mathbf{cH}^T \oplus \mathbf{eH}^T = \mathbf{eH}^T$$

若  $\mathbf{s} \neq \mathbf{0}$ ，则传输中一定有错误发生；若  $\mathbf{s} = \mathbf{0}$ ，传输中无差错或错误图案恰好为一个码字。

若能由伴随式  $\mathbf{s}$  得到差错图样  $\mathbf{e}$ ，则可得到码矢量的估值：

$$\hat{\mathbf{c}} = \mathbf{r} \oplus \mathbf{e} = (c_0 \oplus e_0, c_1 \oplus e_1, \dots, c_{n-1} \oplus e_{n-1})$$

# 差错图案的求解

可以通过解线性方程求解错误图案：

$$S = (s_0, s_1, \dots, s_{n-k-1}) = eH^T$$

$$= (e_0, e_1, \dots, e_{n-1})$$

$$\begin{bmatrix} h_{00} & \cdots & h_{10} & h_{(n-k-1)0} \\ \vdots & \ddots & \vdots & \vdots \\ h_{0i} & \cdots & h_{1i} & h_{(n-k-1)i} \\ h_{0(n-1)} & \cdots & h_{i(n-1)} & h_{(n-k-1)(n-1)} \end{bmatrix}_{n \times (n-k)}$$

得到的线性方程组：

有 $n$ 个未知数，有 $n-k$ 个方程  $\longrightarrow$  有多组解



- 在有理数或实数域中，少一个方程就可能导致无限多个解，而在二元域中，少一个方程导致两个解，少两个方程四个解，以此类推，少 $n - (n - k) = k$ 个方程导致每个未知数有 $2^k$ 个解。
- 因此，由上述方程组解出的 $e$ 可以有 $2^k$ 个解。到底取哪一个作为附加在收码 $r$ 上的差错图案 $e$ 的估值呢？
- **概率译码：**把所有 $2^k$ 个解的重量(差错图案 $e$ 中1的个数)作比较，选择其中最轻者作为 $e$ 的估值。

**依据：** 若BSC信道的差错概率是 $p$ ，则长度 $n$ 的码中错误概率：

0个错	1个错	2个错	...	$n$ 个错
$(1-p)^n$	$p(1-p)^{n-1}$	$p^2(1-p)^{n-2}$		$p^n$

由于 $p \ll 1$ ,       $\gg$                    $\gg$                    $\gg$       ...       $\gg$

出错越少的情况，发生概率越大， $e$ 的重量越轻，汉明距离越小，  
该译码方法实际上体现了最小距离译码准则。

## 伴随式纠错译码步骤:

- 1 根据最可能出现的差错图案计算相应的伴随式, 构造伴随式-差错图案表  $(s, e)$  ;
- 2 对接收向量计算伴随式:  $s = eH^T$  ;
- 3 查  $(s, e)$  表得差错图案  $e$  ;
- 4 纠错:  $\hat{c} = r \oplus e$  ;
- 5 译码, 得到信息矢量估值:  $\hat{m}$

### 例5.2.4

(6,3)线性分组码，系统生成矩阵  $G_s = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$ ,

接收矢量  $r=(111001)$ ，对其译码。

解：求系统校验矩阵。

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

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

1 根据最可能出现的差错图案，由  $s = eH_s^T$ ，构造  $(s,e)$  表。

差错图案 $e$	000000	100000	010000	001000	000100	000010	000001
伴随式 $s$	000	110	011	101	100	010	001

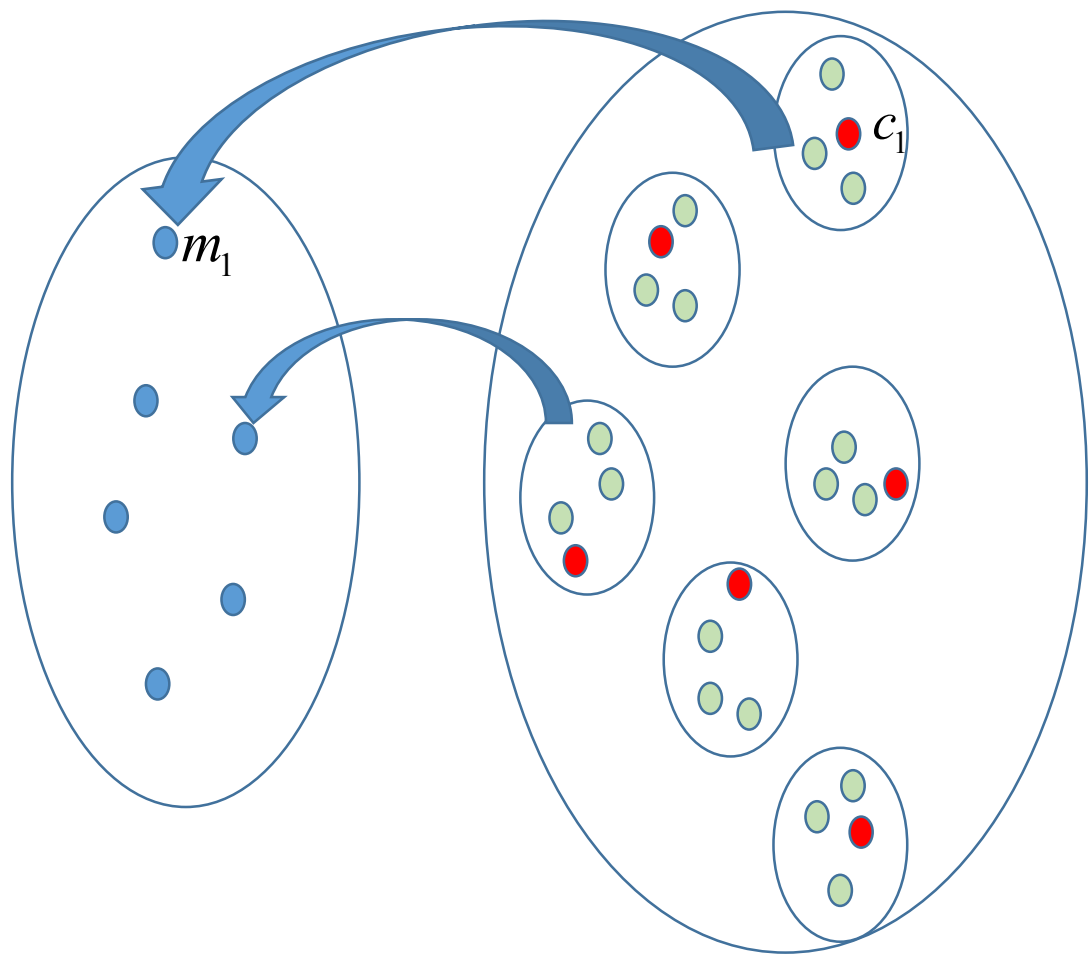
2 由式  $s = rH_s^T$  计算接收向量  $r=(111001)$  的伴随式, 得  $s=(001)$ 。

3 查  $(s,e)$  表可知, 对应的差错图案  $e=(000001)$ 。

4 纠错计算得码字估值

$$\hat{c} = r \oplus e = 111001 \oplus 000001 = 111000$$

5 译码。对于系统(6,3)码, 前三位即发送消息, 即  $m=(111)$ 。  
非系统码需要查消息-码字对照表恢复原始消息。



信息	$m_1$	$m_2$	...	$m_{2^k}$
许用码字	$c_1$	$c_2$		$c_{2^k}$
$2^n - 2^k$ 个 禁用码字				

译码表

# 标准阵列译码

- 从伴随式译码可以看出，伴随式 $s$ 的数目是有限的 $2^{n-k}$ 个。
- 如果 $n-k$ 不太大，我们可以预先把不同 $s$ 下的方程组解出来，把各种情况下的最大概率译码输出列成一个码表。
- 这样，在实时译码时只要象查字典那样查一下码表就可以了。
- 这样构造的表格叫做标准阵列译码表。

# 标准阵列译码表

伴随式



$$s_0 = \theta$$

$$s_1$$

$\vdots$

$$s_i$$

$\vdots$

$$s_{2^{n-k}-1}$$

全零码字



$e_0 + c_0 = 0$	$e_0 + c_1 = c_1$	$\dots$	$e_0 + c_j = c_j$	$\dots$	$e_0 + c_{2^k-1} = c_{2^k-1}$
$e_1 + c_0$	$e_1 + c_1$		$e_1 + c_j$		$e_1 + c_{2^k-1}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_i + c_0$	$e_i + c_1$	$\dots$	$e_i + c_j$	$\dots$	$e_i + c_{2^k-1}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}-1} + c_0$	$e_{2^{n-k}-1} + c_1$		$e_{2^{n-k}-1} + c_j$		$e_{2^{n-k}-1} + c_{2^k-1}$



许用码字



禁用码字

有多少个伴随式?

不同行元素有交集吗?

有多少个码字?

不同列元素有交集吗?



例

(4,2)线性分组码，编码规则如右所示。  
生成标准阵列译码表，分析其纠检错能力。

$$\begin{cases} c_0 = m_0 \\ c_1 = m_1 \\ c_2 = m_0 \oplus m_1 \\ c_3 = m_1 \end{cases}$$

解：

信息	码字
00	0000
01	0111
10	1010
11	1101

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

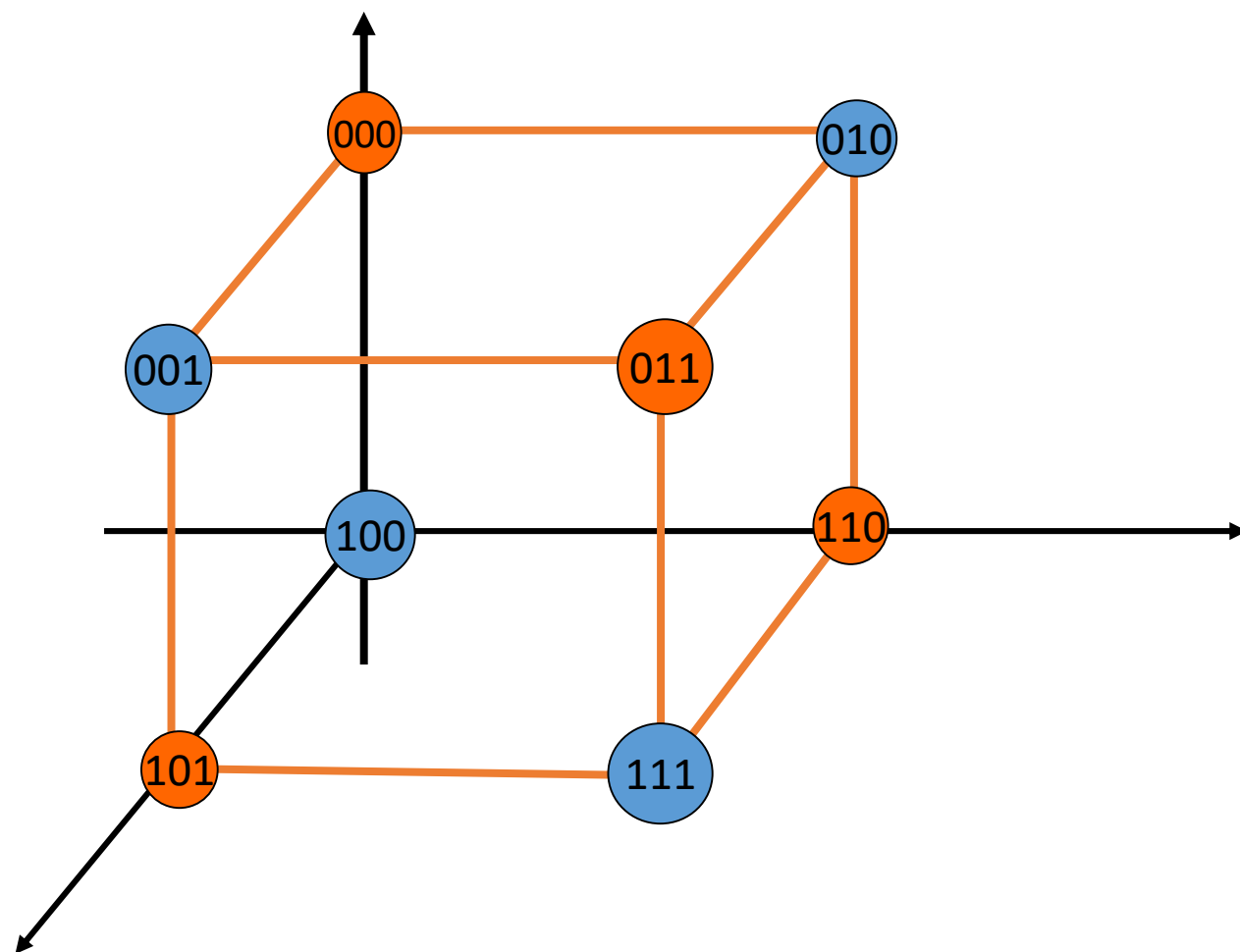
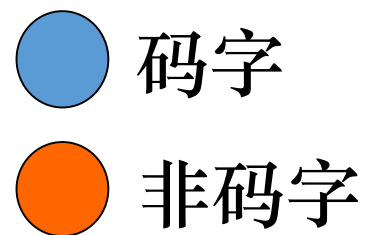
纠检错能力：

能检出所有1位差错，不能检出所有2位差错。  
能纠部分1位差错，但不是所有1位差错都能纠正。

# 纠错码的纠错能力

如果一种码的任一码字内出现了 $e$ 位或 $e$ 位以内的错误，能自动发现，则称该码的**检错能力**为 $e$ 。

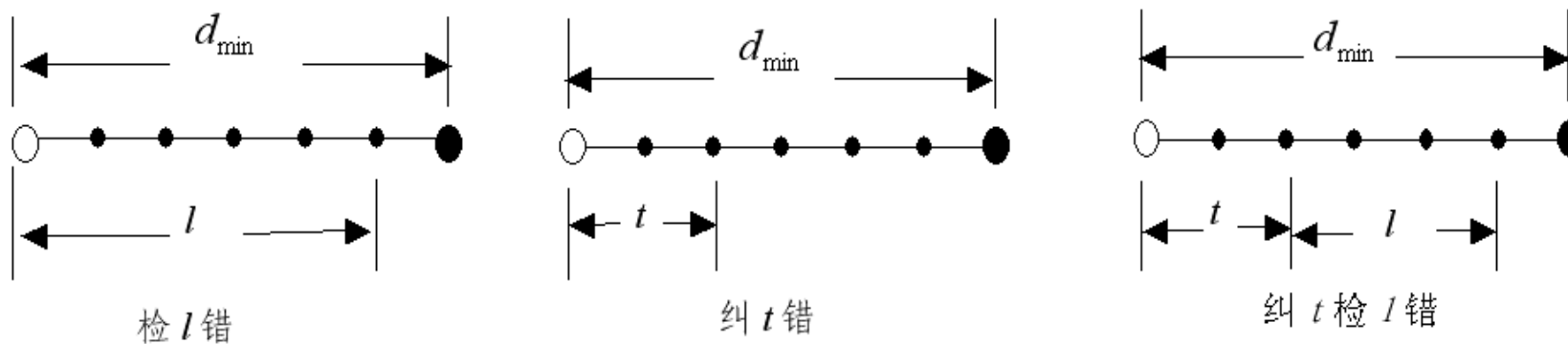
如果一种码的任一码字内出现了 $t$ 位或 $t$ 位以内的错误，能自动纠正，则称该码的**纠错能力**为 $t$ 。



检、纠错能力图示

**定理：** 若纠错码的最小距离为  $d_{\min}$ ，如下任何一个结论独立成立。

- ① 可以检测出任意小于等于  $l=d_{\min}-1$  个差错；  $(d_{\min} \geq l+1)$
- ② 可以纠正任意小于等于  $t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$  个差错；  $(d_{\min} \geq 2t+1)$
- ③ 可以检测出任意小于等于  $l$  同时纠正小于等于  $t$  个差错，其中  $l$  和  $t$  满足  $l+t \leq d_{\min}-1$ ,  $t < l$ 。  $(d_{\min} \geq t+l+1)$



最小码距与检错和纠错能力

**思考：** 分组码的最小码距为  $d_{\min}$  ， 则

$d_{\min} = 1$  :      无纠检错能力

$d_{\min} = 2$  :      检1位错

$d_{\min} = 3$  :      纠1位错，或检2位错

$d_{\min} = 4$  :      纠1位错，同时检2位错

若 $(n,k)$ 线性分组码的最小码距为 4，则该码一定不能纠正2个错误吗？

## 5.2.4 典型码例

### 多个校验位的汉明码

每个校验位是部分或全部信息位按模2和规则确定。汉明码满足下列条件

码 长:  $n=2^r-1$

信息位长:  $k=n-r=2^r-r-1$

码 字 数:  $M=2^k$

监督位长:  $r=n-k$

最小码距:  $d_{\min}=3$

纠错能力:  $t=1$

定理: 若线性分组码能纠正所有1位错误, 当且仅当校验矩阵没有全零列, 且任意两列都不相同。

## ■ 汉明界

任何一个二元 $(n,k)$ 线性分组码，有 $2^k$ 个码字， $2^{n-k}=2^r$ 个伴随式矢量。若要纠正所有小于等于 $t$ 个错误，伴随式的个数必须满足：

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

这个关系式称为**汉明界**。它是构造纠正 $t$ 位错的 $(n,k)$ 码的必要条件。

如果一个 $(n,k)$ 线性分组码使汉明界的等号成立，即伴随式的个数与所有可纠正的错误图样数正好相等，说明校验位得到了充分的利用，这种码称为完备码。

**例5.2.5** 二元(7,4)汉明码的系统码生成矩阵和校验矩阵分别为

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

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

等价编码方程

$$\begin{cases} c_i = m_i, i = 0, 1, 2, 3 \\ c_4 = m_0 \oplus m_1 \oplus m_2 \\ c_5 = m_1 \oplus m_2 \oplus m_3 \\ c_6 = m_0 \oplus m_1 \oplus m_3 \end{cases}$$

伴随式方程

$$\begin{cases} s_0 = r_0 \oplus r_1 \oplus r_2 \oplus r_4 \\ s_1 = r_1 \oplus r_2 \oplus r_3 \oplus r_5 \\ s_2 = r_0 \oplus r_1 \oplus r_3 \oplus r_6 \end{cases}$$

$$\mathbf{u} = (1101)$$

$$\mathbf{c} = \mathbf{u}\mathbf{G}_s = (1101001)$$



若传输无差错：

$$cH^T = [1101001] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = 0$$

译码得：

$$u = (1101)$$

对消息序列 $m=(m_0, m_1, m_2, m_3)=(1101)$ 编码，若传输过程中第4个比特出错，对接收矢量译码。

$m_0$	$m_1$	$m_2$	$m_3$	$c_4$	$c_5$	$c_6$
$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$

$(s,e)$ 表

$e$	$s$
0000000	000
1000000	101
0100000	111
0010000	110
0001000	011
0000100	100
0000010	010
0000001	001

- ①  $C=(1100001)$
- ② 计算 $(s,e)$ 表
- ③ 计算伴随式 $s=(011)$
- ④ 查 $(s,e)$ 表，得错误图案 $e=(0001000)$
- ⑤ 码估值， $\hat{c}=r\oplus e=(1101000)$
- ⑥ 恢复消息， $m=(1101)$

# 第5章 小 结

## 1. 掌握信道编码的基本概念

纠错编码的分类

差错控制方式

译码准则

## 2. 了解信道编码定理

临界值 $C$

### 3. 掌握线性分组码

基本概念：信息位、校验位、码字、汉明重量、汉明距离、码率

编码：生成矩阵 $G$  系统码生成矩阵 $G_s$

译码：译码准则、译码规则、错误译码概率

一致校验矩阵 $H$ ,  $H_s$ , 对偶码

码矢量, 接收矢量, 差错图案, 伴随式

伴随式纠错译码、标准阵列译码

最小汉明距离与纠、检错能力