Communication Systems

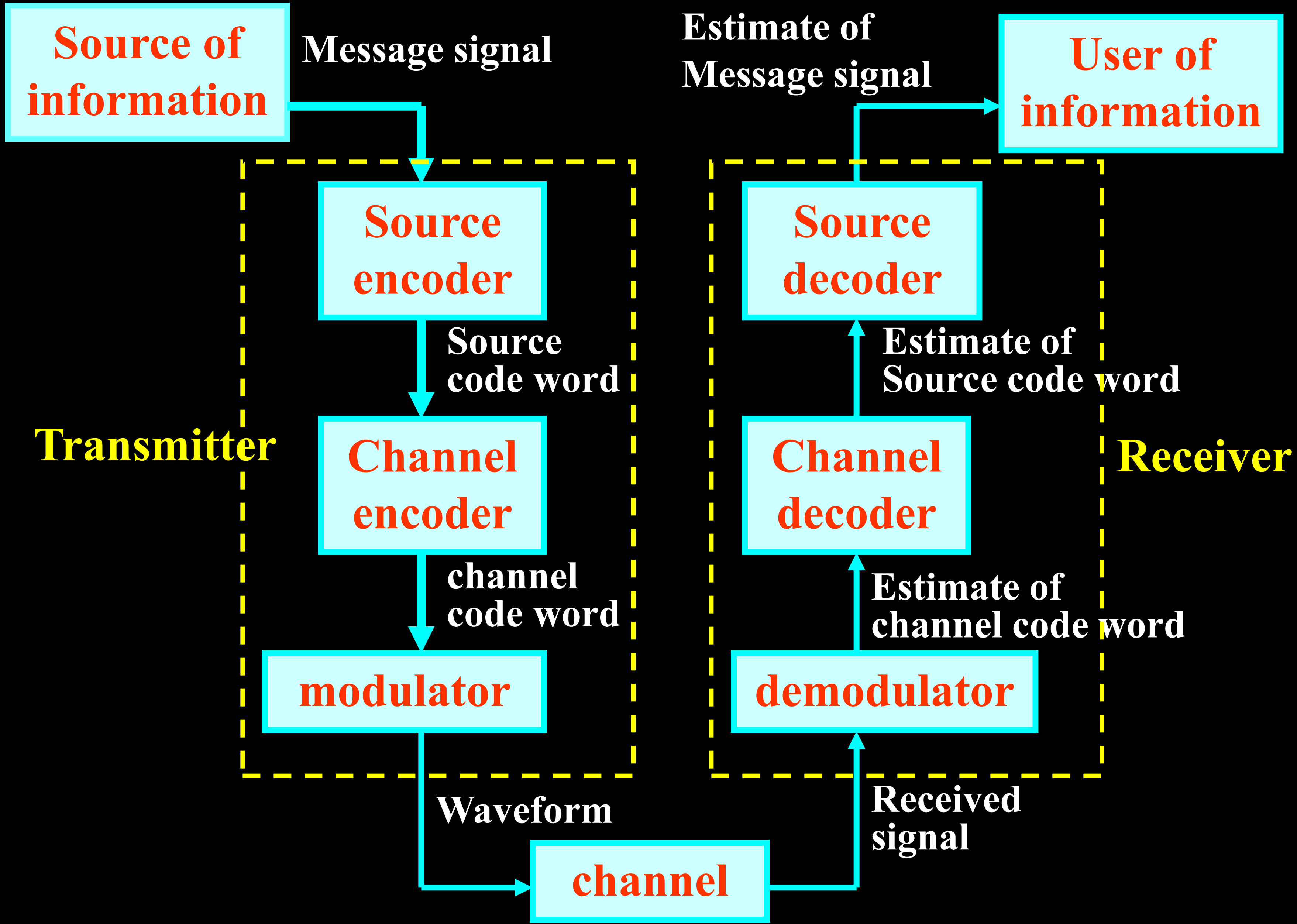
Xiong Shu Hua
Associate Professor

College of Electronics and information Science
Sichuan University

E-mail：xiongsh@scu.edu.cn

Chapter 10

Error-Control Coding



Source of information

Message signal

Source encoder

Source code word

Channel encoder

channel code word

modulator

Transmitter

Waveform

channel

Received signal

demodulator

Estimate of channel code word

Channel decoder

Estimate of Source code word

Source decoder

Estimate of Message signal

User of information

Receiver

Figure 9 Block diagram of digital communication system.

contents

Important classes of error-control coding:

- **Linear block codes 线性分组码**
- **Cyclic codes 循环码**
- **Convolutional codes 卷积码**
- **Turbo codes**
- **Low-density parity-check codes 低密度奇偶校验码**

10.1 Introduction

Purposes of encoding

- Source coding reduces redundancy to improve efficiency.
- Channel coding introduces controlled redundancy to improve reliability.

目的：

信源编码以提高通信的有效性为目的。

信道编码以提高通信的可靠性为目的。

Principle of Error-control Coding

信道编码的主要任务是差错控制编码，亦称检错纠错编码，其目的就是为了使接收端能对接收到的码元序列进行检错和纠错，以降低错误率，提高通信的可靠性。

在发送端利用信道编码器，按照一定的规则在信息码字中增加一些监督码元，接收端的信道译码器利用监督码元和信息码元之间的监督关系来检验接收到的码字，以发现错误或纠正错误。

差错控制编码提高通信的可靠性是以降低通信的有效性为代价的。

Classification of Error-Control Coding:

- Systematic Code 1010 → 1010001
- Nonsystematic Code 1010 → 1000110

根据码字中的信息位是否与原始数字信息一致，可将差错控制编码分为系统码和非系统码；

- Linear code
- Nonlinear code

根据信息码元与监督码元之间的关系是否存在线性特性，可将它分为线性码和非线性码;

$$\begin{cases}\text{Block code}\\ \text{Convolutional code}\end{cases}$$

根据信息码元与监督码元之间的关系是否局限在一个码字内，可分为分组码和卷积码;

Rule: the message sequence is subdivided into sequential blocks each *k* bits long, and each k-bit block is mapped into an *n*-bit block, where *n*>*k*. the number of redundant bits added by the channel encoder to each transmitted block is n-k bits.

Code Rate 码率，编码效率 .

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

$$r \le 1$$

Random error-correcting code
Burst error-correcting code

根据被纠错误的性质可将差错控制编码分为纠随机错误码和纠突发错误码。

Error-detecting code
Error-correcting code

根据译码后是能够检错还是能够纠错，可将差错控制编码分为检错码和纠错码

A. 只能够发现错误的码称为**检错码**，所谓发现错误是指译码后只知道是否有错但不知道错在什么码位上；

B. 能够纠正错误的码称为**纠错码**，所谓纠错是指不但知道是否有错而且还知道错在哪些码位上，因而能够将错误纠正过来。

Methods of Error-Control差错控制的方式:

① ARQ: automatic-repeat request 检错重发：发送端发送的是具有一定检错能力的检错码，接收端在接收的码字中一旦检测出错误，就通过反馈信道通知发送端重发该码字，直到正确接收为止。

② FEC: feed-forward error correction 前向纠错：又称自动纠错。发送端发送的是具有一定纠错能力的纠错码，接收端对接收码字中不超过纠错能力范围的差错自动进行纠正。其优点是不需要反馈信道，但如果要纠正大量错误，必然要求编码时插入较多的监督码元，因此编码效率低，译码电路复杂。

③ 混合纠错：检错重发与前向纠错相结合。

Outline for this class

- Hamming Distance and Hamming Weight
- Relation between minimum distance and capability of error-detecting and error-correcting
- Definition of systematic linear block codes
- How to encode and decode linear block code?
- How to detect and correct error in linear block code?

Code and Code Word

- Code码集，码
- Code word, code vector码字，码矢量

A code consists of a number of code words.

例如：

- 码集 {000，001，010，011，100，101，110，111}有8个码字。
- 码集 {000，010，100，110}有4个码字, 许用码字。

Hamming Distance and Hamming Weight
汉明距离与汉明重量

- **Hamming weight *w(c)*: defined as the number of nonzero elements in the code vector *c*.**

1 0 1 1 0 0 1 1 → w = 5

- **Hamming Distance $d(c_1,c_2)$: defined as the number of locations in which their respective elements differ between two code words c_1 and c_2.**

1 0 1 1 0 0 1 1

1 0 0 1 1 0 1 0

→ d = 3

Minimum Distance and Minimum Weight
最小码距与最小码重 Page 637

- The minimum weight w_{min} is defined as the smallest weight of all nonzero code vectors 非全零码矢量 in the code.

- The minimum distance d_{min} is defined as the smallest Hamming distance between any pair of code vectors in the code.

In linear block code $\boxed{d_{\min} = w_{\min}}$

What determines the error-detecting and error-correcting capability of a linear block code?

Minimum Distance !

最小码距

Relation between minimum distance and capability of error-detecting and error-correcting 最小码距与检错纠错能力的关系

Example 1

码集:

{000， 001， 010， 011， 100， 101， 110， 111}

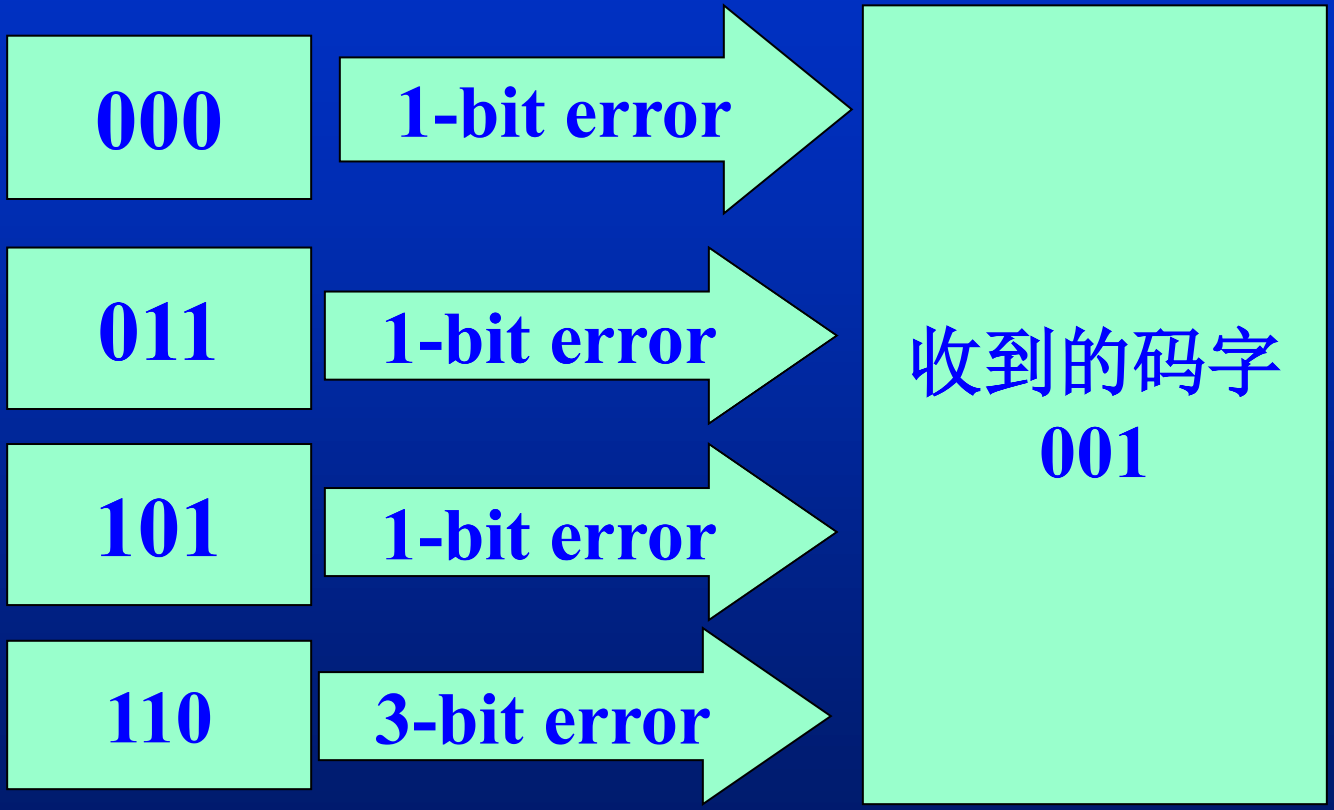
最小码重$W_{\min}$ =1， 最小码距$d_{\min}$ =1

既不能发现错误也不能纠正错误.

Example 2

码集 {000， 011， 101， 110}

$w_{\min}=2$， $d_{\min}=2$.



000

1-bit error

011

1-bit error

101

1-bit error

110

3-bit error

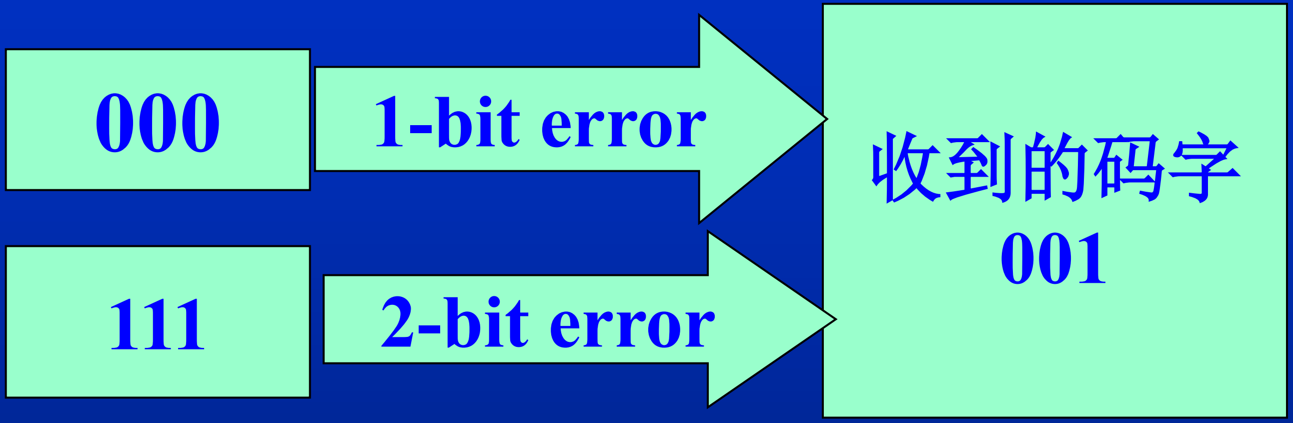
收到的码字
001

用这种简单的校验关系可以发现一位或三位误码， 但不能纠正错误。

Example 3

码集 {000，111}

$W_{\min}=3$，$d_{\min}=3$。



- 能发现所有两位以下错误，但不能检测出三位误码。
- 由于一个码字中同时发生两位误码的可能性比发生一位误码的可能性小得多，可以认为是发送码字发生一位误码造成的，因此可以判定原来发送的正确码字是000，即是说这种码集可以纠正一位错误。

最小码距与检错纠错能力的关系

① 在一个码集中检测 e 个错误，要求最小码距满足：

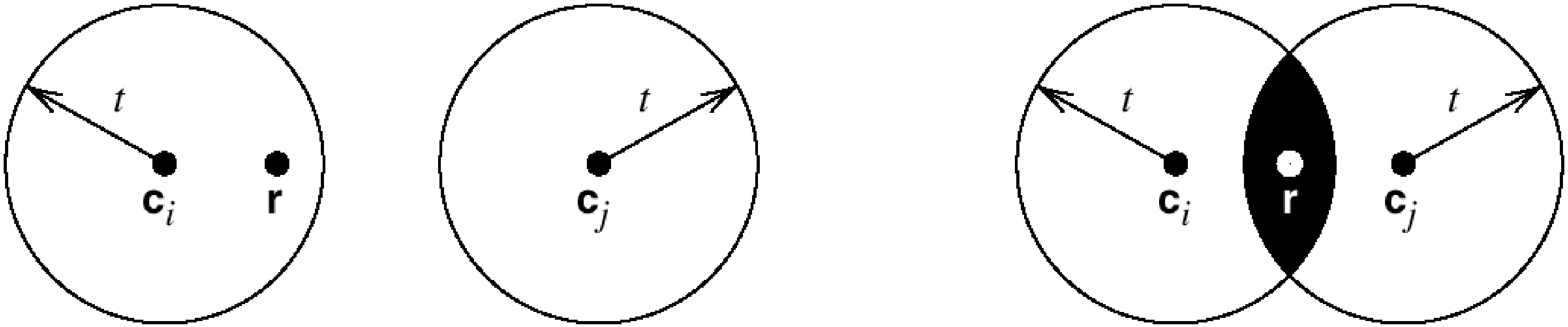
$$d_{\min} \geq e+1$$

② 在一个码集中纠正 t 个错误，要求最小码距满足：

$$d_{\min} \geq 2t+1 \Rightarrow \boxed{t \leq \left\lfloor \frac{1}{2}(d_{\min}-1) \right\rfloor \qquad (10.25)}$$

③ 在一种码集中纠正 t 个错误同时检测 e 个错误，要求最小码距满足：

$$d_{\min} \geq t+e+1$$



(*a*) $d(ci, cj) \geq 2t + 1$.

(*b*) $d(ci, cj) < 2t$

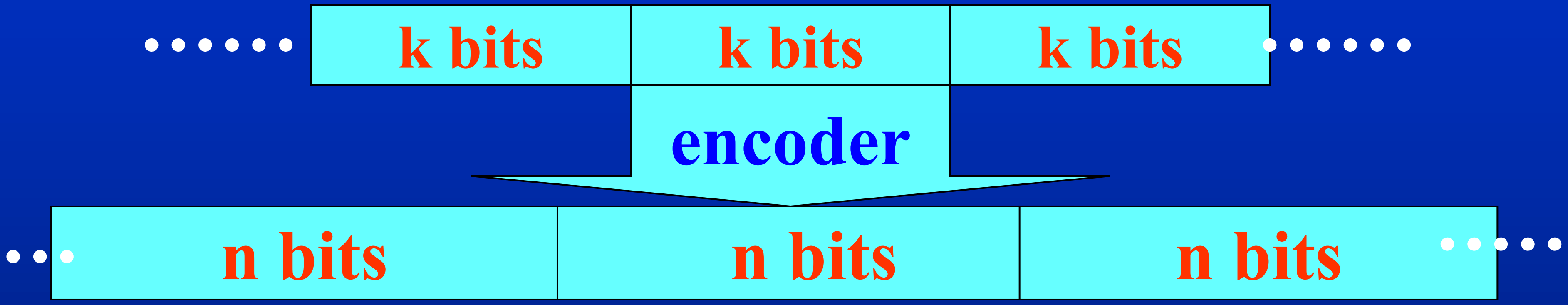
The received vector is denoted by r.

Fig. 10.6 码距与检错纠错能力关系示意图.

10.3 Linear Block Codes 线性分组码

What is block code?

Message sequence



What is linear block code?

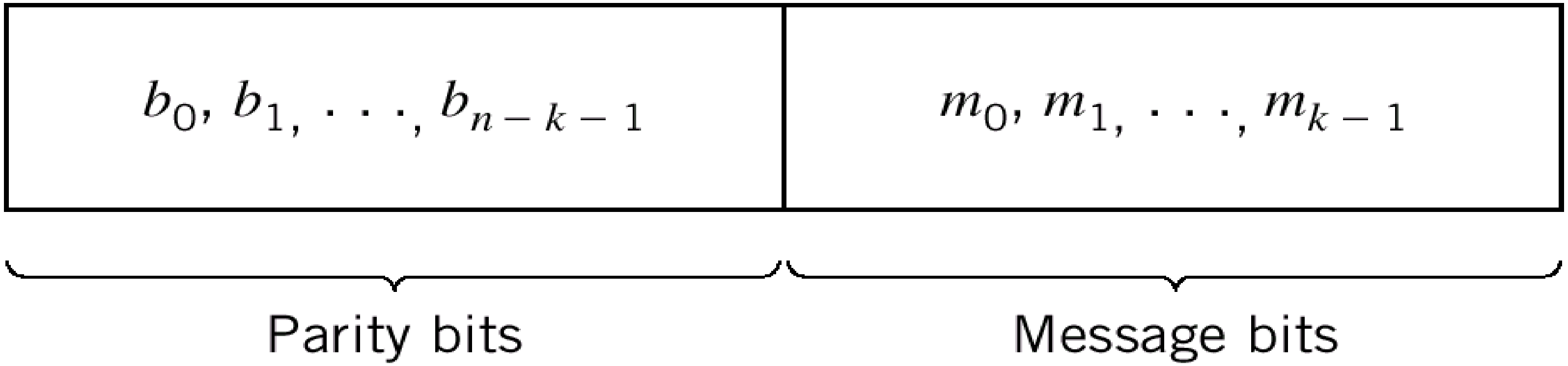
线性码是指信息位与监督位满足一组线性代数方程式的码.

(n, k) 线性码 编码效率: $\frac{k}{n}$

What is Systematic linear block codes?

The message bits are transmitted in unaltered form.

Figure 10.4 Structure of systematic code word



$b_0, b_1, \ldots, b_{n-k-1}$ | $m_0, m_1, \ldots, m_{k-1}$

Parity bits | Message bits

K bits: 1 1 0 1

N bits: 0 0 1 1 0 1

or 1 1 0 1 0 0 --→ systematic code

N bits: 1 0 1 0 0 1--→ non-systematic code k-bit message $m_0, m_1, \ldots, m_{k-1}$

n-bit channel code word

$$c_0, c_1, \ldots c_{n-1} = \boxed{b_0, b_1, \ldots b_{n-k-1}}, m_0, m_1, \ldots, m_{k-1}$$

parity check bits
校验比特, 校验位
监督码元, 监督位

message bits
信息比特 信息位

The n-k bits are computed from message bits in accordance with a given encoding rule 编码规则.

Relation between parity bits and message bits

The (n-k) parity bits are linear sums of the k message bits, as shown by generalized relation 监督关系式

$$b_i = p_{0i}m_0 + p_{1i}m_1 + \cdots + p_{k-1,i}m_{k-1} \qquad (10.2)$$

Where, the coefficients are defined as follows:

$$p_{ij} = \begin{cases} 1 & \textit{if} \quad b_i \quad \textit{depends} \quad \textit{on} \quad m_j \\ 0 & \qquad\qquad \textit{otherwise} \end{cases}$$

系数 p_{ij} 的选择要使生成矩阵的各行线性独立，且检验方程唯一。

Matrix Notation 矩阵表示

1-by-k message vector $m=[m_0,m_1,\cdots,m_{k-1}]$

1-by-(n-k) parity vector $b=[b_0,b_1,\cdots,b_{n-k-1}]$

1-by-n code vector $c=[c_0,c_1,\cdots,c_{n-1}]$

All of them are row vectors 行矢量.

It is clear that *c* can be expressed in terms of *m* and *b* as follows 分块矢量

$$c=[b\vdots m]$$

监督方程：
$$b_i = p_{0i}m_0 + p_{1i}m_1 + \cdots + p_{k-1,i}m_{k-1} \qquad (10.2)$$

$$\begin{cases} b_0 = p_{00}m_0 + p_{10}m_1 + \cdots + p_{k-1,0}m_{k-1} \\ b_1 = p_{01}m_0 + p_{11}m_1 + \cdots + p_{k-1,1}m_{k-1} \\ \vdots \\ b_{n-k-1} = p_{0,n-k-1}m_0 + p_{1,n-k-1}m_1 + \cdots + p_{k-1,n-k-1}m_{k-1} \end{cases}$$

k-by-(n-k)
系数矩阵

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

$$\boxed{\boldsymbol{b} = \boldsymbol{mP}}$$

Where p_{ij} is 0 or 1.

Because $b = mP$ $c = [b \vdots m]$

therefore $c = [mP \vdots m] = m[P \vdots I_k]$

$$I_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

I_k : ***k-by-k*** **identity matrix**
k阶单位矩阵

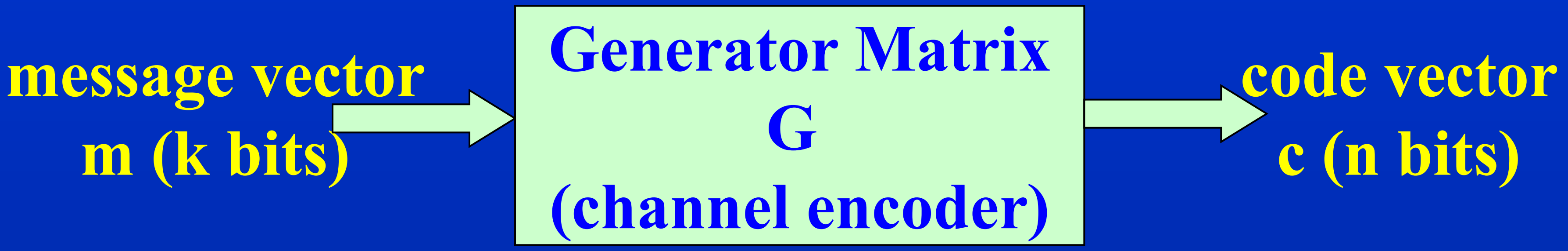
k-by-n **generator matrix**
生成矩阵

$$G = [P \vdots I_k] \qquad (10.12)$$

Hence , we get generation equation 生成方程

$$c = mG \qquad (10.13)$$

Why is G called generator matrix? $c = mG$



How to get the full set of code words?

The full set of code words, referred to simply as code, is generated in accordance with *c=mG* by setting the message vector m range through the set of all 2^k binary k-tuples (1-by-k vectors) 2^k个不同的二进制组合.

$$b_i = p_{0i}m_0 + p_{1i}m_1 + \cdots + p_{k-1,i}m_{k-1} \qquad (10.2)$$

$$\begin{cases} b_0 = p_{00}m_0 + p_{10}m_1 + \cdots + p_{k-1,0}m_{k-1} \\ b_1 = p_{01}m_0 + p_{11}m_1 + \cdots + p_{k-1,1}m_{k-1} \\ \vdots \\ b_{n-k-1} = p_{0,n-k-1}m_0 + p_{1,n-k-1}m_1 + \cdots + p_{k-1,n-k-1}m_{k-1} \end{cases}$$

变形为齐次方程

$$\begin{cases} b_0 + p_{00}m_0 + p_{10}m_1 + \cdots + p_{k-1,0}m_{k-1} = 0 \\ b_1 + p_{01}m_0 + p_{11}m_1 + \cdots + p_{k-1,1}m_{k-1} = 0 \\ \vdots \\ b_{n-k-1} + p_{0,n-k-1}m_0 + p_{1,n-k-1}m_1 + \cdots + p_{k-1,n-k-1}m_{k-1} = 0 \end{cases}$$

Parity-Check Matrix
奇偶校验矩阵或监督矩阵 H

Let H denote an *(n-k)-by-n* matrix, defined as

$$H = [I_{n-k} \vdots P^T]$$

***P^T* ： *(n-k)-by-k* matrix, the transpose（转置）of P**

since $$G = [P \vdots I_k]$$

Hence $$HG^T = [I_{n-k} \vdots P^T]\begin{bmatrix} P^T \\ \cdots \\ I_k \end{bmatrix} = P^T + P^T = 0$$

Modulo-2 arithmetic

That is $$HG^T = 0$$

We have known that

$$\boxed{c = mG}$$

Postmultiplying (后乘, 右乘) both sides by *H*^{T}, we get

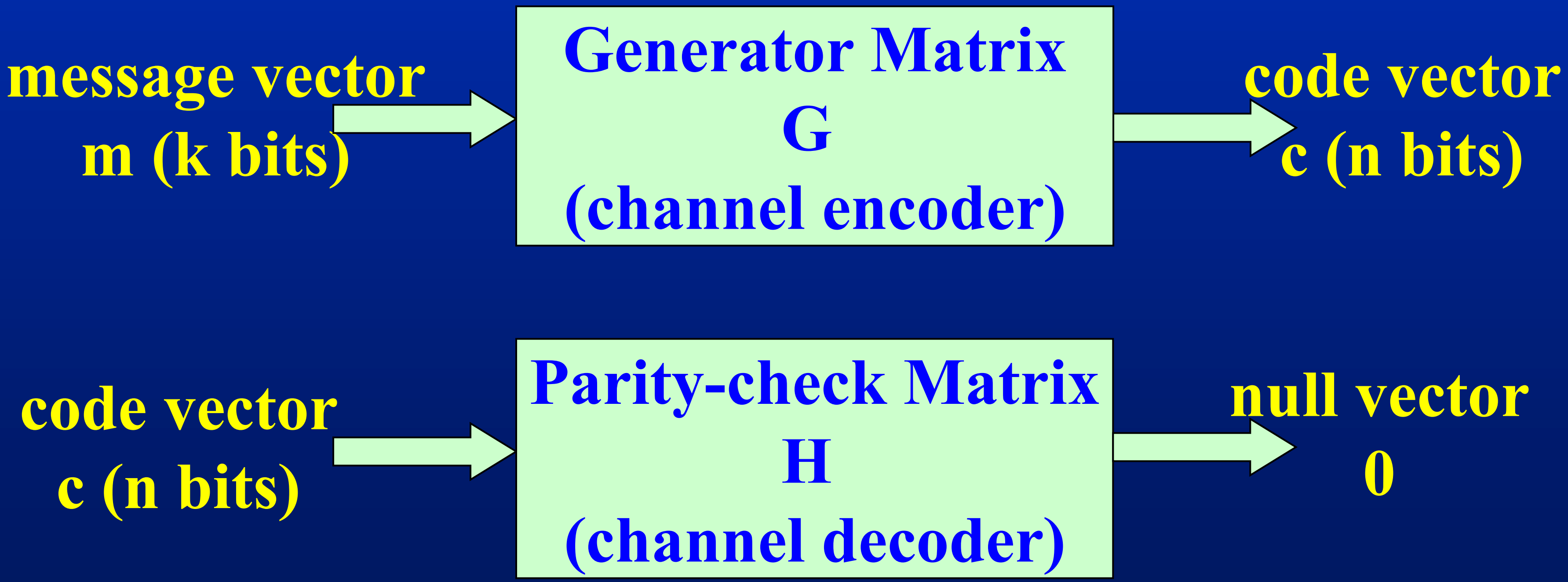
$$\boxed{cH^T = mGH^T = 0 \qquad (10.16)}$$

Equation (10.16) is called parity-check equation 奇偶校验方程.

Generator equation $$c = mG \qquad (10.13)$$

Parity-check equation $$cH^T = 0 \qquad (10.16)$$

They are basic to the description and operation of a linear block code.



message vector m (k bits)

Generator Matrix G (channel encoder)

code vector c (n bits)

code vector c (n bits)

Parity-check Matrix H (channel decoder)

null vector 0

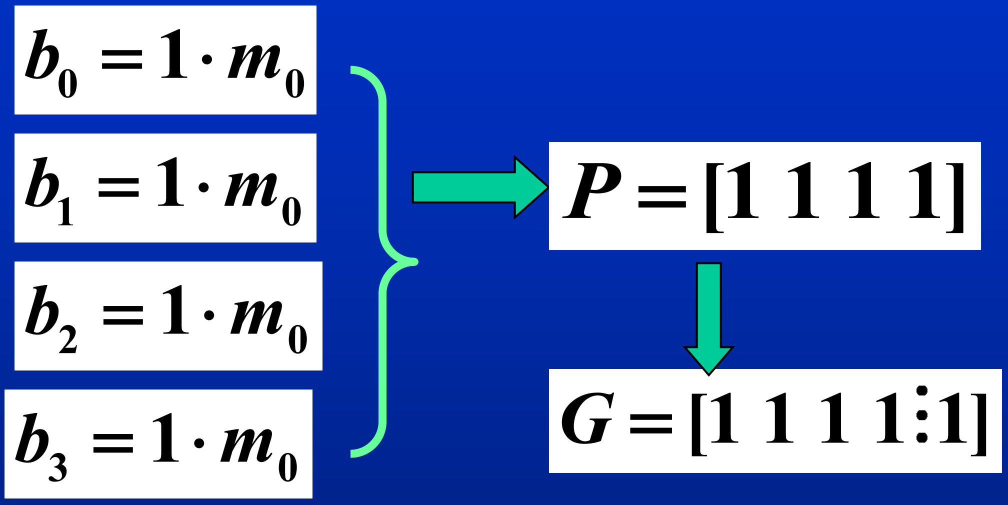
Example 10.1 Repetition Codes 重复码

- It is the simplest type of linear block codes.
- (n,1) block code: only 1-bit message, (n-1) parity-check bits, which are the repetition of the message bit.

例：(5,1) 重复码

$k=1$ $n=5$ $I_k=[1]$

$$c_0,c_1,\ldots c_4=b_0,b_1,b_2,b_3,m_0$$



$b_0=1\cdot m_0$

$b_1=1\cdot m_0$

$b_2=1\cdot m_0$

$b_3=1\cdot m_0$

$P=[1\ 1\ 1\ 1]$

$G=[1\ 1\ 1\ 1\ \vdots 1]$

$$H=[I_{n-k}\vdots P^T]=\begin{bmatrix}1&0&0&0\vdots&1\\0&1&0&0\vdots&1\\0&0&1&0\vdots&1\\0&0&0&1\vdots&1\end{bmatrix}$$

Because: m=0 or 1

Thus, c=mG=00000 or 11111

码集中只有两个码字：00000 and 11111

- 生成矩阵G用于编码。 $c=mG$
- 校验矩阵H用于译码。 $cH^T=0$
- 校正子用于检错或纠错。$s=rH^T$

What is used to detect and correct errors in the decoder?

Syndrome
校正子,伴随式

Error Pattern 错误图样

transmitted code(发送码): *c* (1×n vector)

received code(接收码): *r* (1×n vector)

error pattern(错误图样): *e* (1×n error vector)

$$r = c + e$$

$$e = \{e_1, e_2, \ldots e_i, \ldots e_n\}$$

$$e_i = \begin{cases} 1, & \text{if an error has occurred in the i}^{\text{th}} \text{ location} \\ 0, & \text{otherwise} \end{cases}$$

If c= [0 0 1 1 0 1 0]
r= [0 1 1 1 0 0 0]

Then e= [0 1 0 0 0 1 0]

Our task now is to decode the code vector *c* from the received vector *r*. How to do it?

Definition and properties of Syndrome 校正子(伴随式)定义与性质

Definition:

$$s = rH^T \qquad (10.19)$$

H^T is a $n \times (n\text{-}k)$ vector, so, s is a 1×(n-k) vector.

Property 1: it depends only on the error pattern, and not on the transmitted code word.

because:

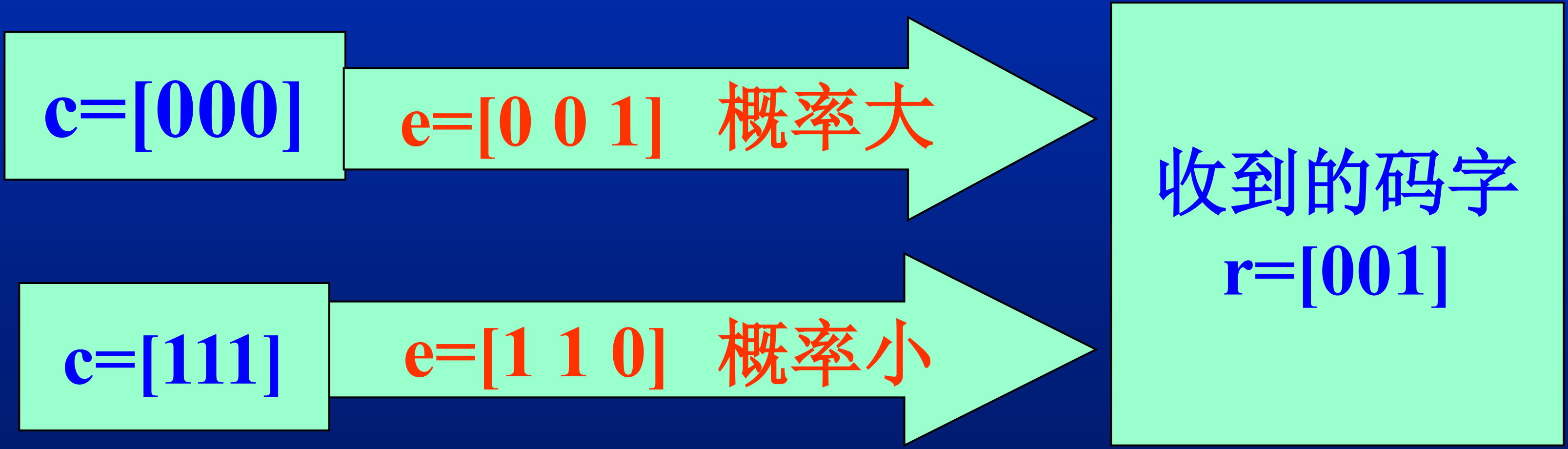
$$s = rH^T = (c+e)H^T = cH^T + eH^T = eH^T$$

如果在传输过程中没有发生错误，即 E=0，则有伴随式 S=0。

Property 2: all error patterns that differ by a code word have the same syndrome.

不同码字的不同错误图样可能对应相同的校正子,因此不能根据校正子准确确定原始的发送码字.

Example 3 码集 {000，111}


c=[000]

e=[0 0 1] 概率大

c=[111]

e=[1 1 0] 概率小

收到的码字
r=[001]

因此，根据概率论或者在某种假设前提下，可以利用伴随式进行纠错.

Syndrome Decoding 校正子译码

The decoding procedure for a linear block code

① For the received vector *r*, compute the syndrome $s=rH^T$

② Identify the error pattern e_{max} with the largest probability of occurrence.

③ Compute the code vector $c = r + e_{max}$ as the decoded version of the received vector *r*.

最大似然译码：

运用矩阵的分块，可将监督矩阵按列表示成：

$$H=\left[H_1 \quad H_2 \quad \cdots \quad H_i \quad \cdots \quad H_n\right]$$

$$e=\{e_1,e_2,\ldots e_i,\ldots e_n\}$$

伴随式的计算变成

$$s=rH^T=(c+e)H^T=cH^T+eH^T=eH^T$$

$$s=eH^T=e_1H_1^T+e_2H_2^T+\cdots e_iH_i^T+\cdots+e_nH_n^T$$

$$s = eH^T = e_1 H_1^{\ T} + e_2 H_2^{\ T} + \cdots e_i H_i^{\ T} + \cdots + e_n H_n^{\ T}$$

- 如果在传输过程中没有发生错误，即 E=0，则有伴随式 S=0。

- 如果一个码字在传输过程中只有第 i 位发生错误，即e_i=1，而其它码元均无错，这时应有：

$$\boxed{s = H_i^{\ T}} \quad \text{or} \quad \boxed{s^T = H_i}$$

Conclusion: 当接收码字中有且只有第 i 位发生误码时，接收端伴随式电路的计算结果正好和监督矩阵的第 i 列相同，跟据这一结果，译码电路可判断出错误的具体位置并自动将第 i 位纠正过来。

Example 10.2 Hamming Codes

汉明码是能纠正一位错误的线性分组码。

It is a special (n, k) linear block code that can correct 1-bit error. Speciality (特殊性）

Block length: $n = 2^m - 1$

Number of message bits: $k = 2^m - m - 1$

Number of parity bits: $n - k = m$

Such as (7, 4) Hamming code.

(7, 4) Hamming Code

Generator matrix

$$G=\left[\underbrace{\begin{matrix}1&1&0\\0&1&1\\1&1&1\\1&0&1\end{matrix}}_{p}\begin{matrix}\vdots\\\vdots\\\vdots\\\vdots\end{matrix}\underbrace{\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{matrix}}_{I_k}\right]$$

Parity-check matrix

$$H=\left[\underbrace{\begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix}}_{I_{n-k}}\begin{matrix}\vdots\\\vdots\\\vdots\end{matrix}\underbrace{\begin{matrix}1&0&1&1\\1&1&1&0\\0&1&1&1\end{matrix}}_{p^T}\right]$$

Suppose： received vector is [1100010]

Determine: is it correct? If it is wrong, under the condition that only one bit may be wrong, which bit is wrong? What is the correct code.

Because

$$s=[1100010]\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&1&0\\0&1&1\\1&1&1\\1&0&1\end{bmatrix}=[0\quad 0\quad 1]$$

$$s\neq 0$$

So, the received code vector is wrong.

$$s = [0 \;\; 0 \;\; 1]$$

$$H = \left[\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{I_{n-k}} \begin{matrix} \vdots \\ \vdots \\ \vdots \end{matrix} \underbrace{\begin{matrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{matrix}}_{p^T} \right]$$

It is the same as the third column of H, so the third bit is wrong.

the error pattern with the highest probability of occurrence is e=[0 0 **1** 0 0 0 0]

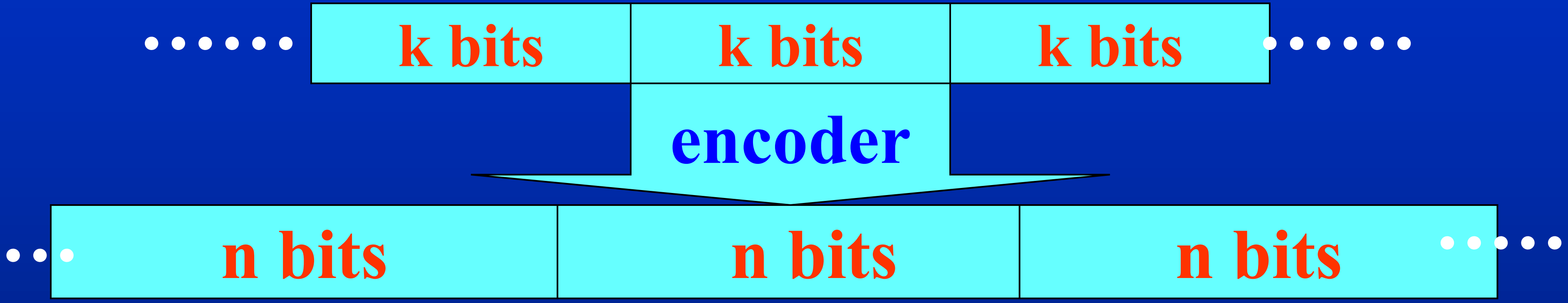
Therefore, the decoded CORRECT word is

r = [1 1 **0** 0 0 1 0] 纠错后 → [1 1 **1** 0 0 1 0]

Review about systematic linear block code

block code:

Message sequence



linear block code:

线性码是指信息位与监督位满足一组线性代数方程式的码.

(n, k) 线性码 编码效率: $\dfrac{k}{n}$

Two matrixes, two equations:

Generator matrix 生成矩阵

Parity-check matrix 奇偶校验矩阵，监督矩阵

$$G=[P\vdots I_k]$$

$$H=[I_{n-k}\vdots P^T]$$

Generator equation

$$c=mG \qquad (10.13)$$

Parity-check equation

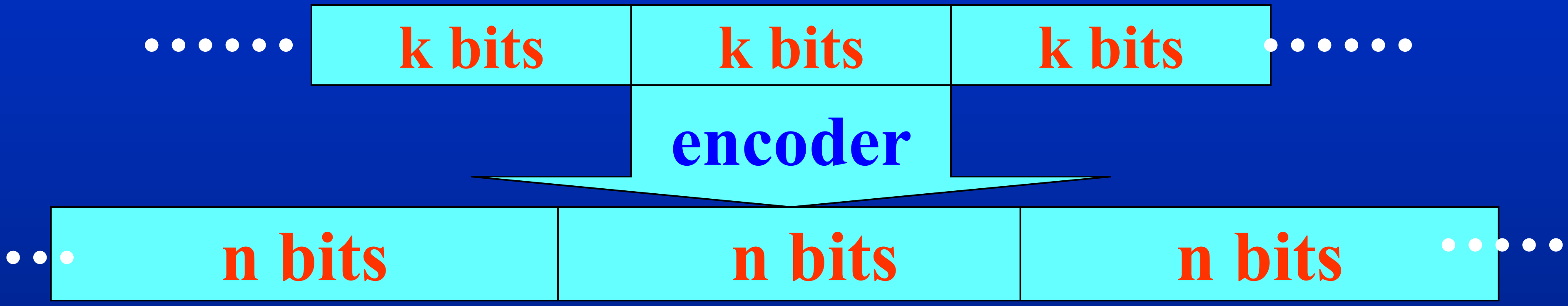
$$cH^T=0 \qquad (10.16)$$

- **生成矩阵G用于编码。** ***$c=mG$***
- **校验矩阵H用于译码。** ***$cH^T=0$***
- **校正子用于检错或纠错。*$s=rH^T$***

Review about systematic linear block code

block code:

Message sequence



linear block code:

线性码是指信息位与监督位满足一组线性代数方程式的码.

(n, k) 线性码 编码效率: $\frac{k}{n}$

Two matrixes, two equations:

Generator matrix 生成矩阵

Parity-check matrix 奇偶校验矩阵，监督矩阵

$$G = [P \vdots I_k]$$

$$H = [I_{n-k} \vdots P^T]$$

Generator equation

$$c = mG \qquad (10.13)$$

Parity-check equation

$$cH^T = 0 \qquad (10.16)$$

- 生成矩阵G用于编码。 $c=mG$
- 校验矩阵H用于译码。 $cH^T=0$
- 校正子用于检错或纠错。$s=rH^T$

10.4 Cyclic Codes 循环码

- Cyclic code is a kind of linear block codes.
- Any cyclic shift (循环移位) of a code word in the code is also a code word.
- Cyclic codes are easy to encode.
- Cyclic codes possess a well-define mathematical structure, which has led to the development of very efficient decoding schemes for them.

Cyclic Property 循环特性

Assume that $(c_0, c_1, \cdots c_{n-1})$ is a code word of an (n,k) linear block code.

码字： $(c_0, c_1, \cdots c_{n-1})$

循环移位一位后的码字： $(c_{n-1}, c_0, c_1, \cdots c_{n-2})$

$\vdots$

循环移位 *i* 位后的码字： $(c_{n-i}, \cdots c_{n-1}, c_0, c_1 \cdots c_{n-i-1})$

$\vdots$

循环移位 *n*-1 位后的码字： $(c_1, c_2, \cdots, c_{n-1}, c_0)$

All of them are also code words in the code.

Code Polynomial 码多项式

Code word: $(c_0, c_1, \cdots c_{n-1})$

Code polynomial:

$$c(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}$$

循环移位*i*次后的码字

$(c_{n-i}, \cdots c_{n-1}, c_0, c_1 \cdots c_{n-i-1})$

?

对应的码多项式：

$$c^{(i)}(X) = c_{n-i} + \cdots + c_{n-1} X^{i-1} + c_0 X^i + c_1 X^{i+1} + \cdots + c_{n-i-1} X^{n-1} \quad (10.30)$$

对 *c(X)* 作二元域上的代数运算：

$$X\ c(X) = c_0 X + c_1 X^2 + c_2 X^3 + \cdots + c_{n-1} X^n$$

$$\begin{aligned} & X^i c(X) \\ & = X^i (c_0 + c_1 X + \cdots + c_{n\text{-}i\text{-}1} X^{n\text{-}i\text{-}1} + c_{n\text{-}i} X^{n\text{-}i} + \cdots + c_{n\text{-}1} X^{n\text{-}1}) \\ & = c_0 X^i + c_1 X^{i+1} + \cdots + c_{n\text{-}i\text{-}1} X^{n\text{-}1} + c_{n\text{-}i} X^n + \cdots + c_{n\text{-}1} X^{n+i\text{-}1} \end{aligned}$$

Because in modulo-2 addition, $c_{n-i} + c_{n-i} = 0$

$$\begin{aligned} X^i c(X) = & \; c_{n\text{-}i} + \cdots + c_{n\text{-}1} X^{i\text{-}1} + c_0 X^i + c_1 X^{i+1} + \cdots + c_{n\text{-}i\text{-}1} X^{n\text{-}1} \\ & + c_{n\text{-}i}(X^n + 1) + \cdots + c_{n\text{-}1} X^{i\text{-}1}(X^n + 1) \end{aligned}$$

$$c^{(i)}(X) = c_{n\text{-}i} + \cdots + c_{n\text{-}1} X^{i\text{-}1} + c_0 X^i + c_1 X^{i+1} + \cdots + c_{n\text{-}i\text{-}1} X^{n\text{-}1} \quad (10.30)$$

Relation between $X^i c(X)$ and $c^{(i)}(X)$

$$\frac{X^i c(X)}{X^n+1} = q(X) + \frac{c^{(i)}(X)}{X^n+1}$$

remainder
余式

Quotient 商式 $q(X) = c_{n-i} + c_{n-i+1}X + \cdots + c_{n-1}X^{i-1}$

$$X^i c(X) = q(X)(X^n+1) + c^{(i)}(X)$$

$$c^{(i)}(X) = X^i c(X) \bmod (X^n+1) \qquad (10.33)$$

如果 $c(X)$ 是循环码集中的一个码多项式，则 $X^i c(X)$ 除以 X^n+1 所得的余式就是该码循环移位 i 位后的码多项式。

Generator Polynomial 生成多项式

Let *g(X)* be a polynomial of degree (阶） n-k that is a factor （因式） of X^n+1. It may be expanded as:

$$g(x) = 1 + \sum_{i=1}^{n-k-1} g_i X^i + X^{n-k}, \quad g_i = 0 \text{ or } 1$$

Generator polynomial

- A cyclic code is uniquely determined by the generator polynomial; 循环码由生成多项式唯一确定。
- g(x) is the polynomial of least degree in the code;
- Each code in the cyclic code can be expressed as

$$c(X) = a(X)g(X) \qquad (10.35)$$

Encoding Procedure

Suppose: we are given g(X) and message sequence $(m_0,m_1\ldots,m_{k-1})$

***R*equire: obtain the (n,k) systematic cyclic code.**

Message polynomial $m(X)=m_0+m_1X+\cdots+m_{k-1}X^{k-1}$

$$c_0,c_1,\ldots c_{n-1}=\underbrace{b_0,b_1,\ldots b_{n-k-1}}_{\text{n-k parity check bits}},\underbrace{m_0,m_1,\ldots,m_{k-1}}_{\text{k message bits}}$$

$$b(X)=b_0+b_1X+\cdots+b_{n-k-1}X^{n-k-1}$$

Therefore, code polynomial ?

$$c(X)=b(X)+X^{n-k}m(X) \qquad (10.38)$$

How to determine b(X)?

Because

$$c(X) = a(X)g(X) \qquad (10.35)$$

$$c(X) = b(X) + X^{n-k}m(X) \qquad (10.38)$$

Thus,

$$a(X)g(X) = b(X) + X^{n-k}m(X)$$

$$\Downarrow$$

$$\frac{X^{n-k}m(X)}{g(X)} = a(X) + \frac{b(X)}{g(X)} \qquad (10.39)$$

It states that the polynomial *b(X)* is the remainder left over after dividing x^{n-k} *m(X)* by *g(X)*.

Steps for encoding an (n,k) systematic cyclic code P644

1. Multiply the message polynomial m(X) by X^{n-k};
2. Divide $X^{n-k}\, m(X)$ by the generator polynomial g(X), obtaining the remainder b(x);
3. Add b(x) to $X^{n-k}\, m(X)$, obtaining the code polynomial c(X).

How to select generator polynomial?

Rule: Any factor of X^n+1 can be used as a generator polynomial, the degree of the factor determines the number of parity bits.

X^n+1的因式，n就是整个码的长度，因式的阶数就是校验位的个数。

It is difficult to select a polynomial factor to construct a **GOOD** cyclic code.

Example: (7, k) 循环码

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

if $g(X)=1+X$ → (7,6) cyclic code

if $g(X)=1+X^2+X^3$ → (7,4) cyclic code

if $g(X)=1+X+X^3$ → (7,4) cyclic code

if $g(X)=(1+X)(1+X+X^3)$ → (7,3) cyclic code

if $g(X)=(1+X)(1+X^2+X^3)$ → (7,3) cyclic code

if $g(X)=(1+X+X^3)(1+X^2+X^3)$ →

(7,1) cyclic code

Generator and Parity-Check Matrices
生成矩阵G 与校验矩阵H

Generator matrix polynomial 生成矩阵多项式

$$G(X)=\begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{k-1}g(X) \end{bmatrix} \xrightarrow{\text{系数}} \mathbf{G} \longrightarrow \mathbf{H}$$

Note: 这样得到的生成矩阵不是系统形式的，需要经过变换才可以得到系统形式的G。

$$\boldsymbol{G}=[\boldsymbol{P} \vdots \boldsymbol{I}_k]$$

Encoder for Cyclic Codes P645

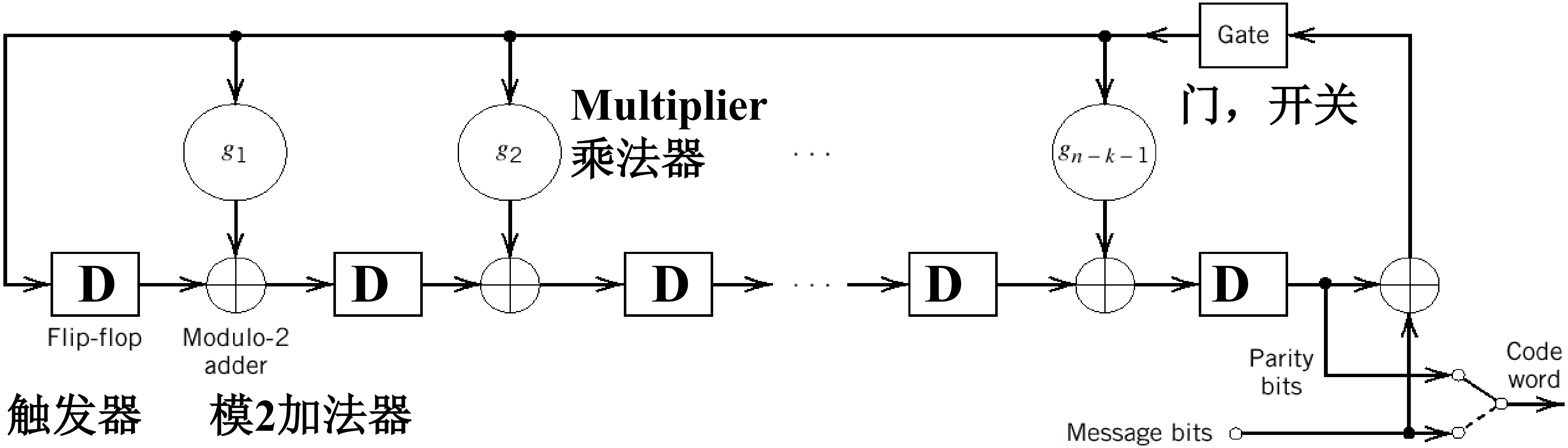
系统形式循环码的编码步骤:

1. 信息多项式 m(X) 乘以 X^{n-k};
2. $X^{n-k}\,m(X)$ 除以生成多项式 g(X), 得到余式 b(x);
3. 码多项式 c(X)= b(x)+ $X^{n-k}\,m(X)$.

How to implement these steps with circuits?
How to design the encoder?

These encoding procedure can be implemented by means of encoder consisting of a linear feedback shift register with (n-k) stages. n-k 级线性反馈移位寄存器构成。

Fig.10.8 Encoder for an (*n*, *k*) cyclic code



$$g(x) = 1 + \sum_{i=1}^{n-k-1} g_i X^i + X^{n-k}, \quad g_i = 1 \text{ or } 0$$

Filp-flop: 触发器, unit-delay element: 单位延时元件, 一共有(n-k)级.

$$m = [m_0, m_1, \cdots, m_{k-1}]$$

$$c_0, c_1, \ldots c_{n-1} = \boxed{b_0, b_1, \ldots b_{n-k-1}}, m_0, m_1, \ldots, m_{k-1}$$

Operation of the encoder

1. The gate is switched on. The k message bits are shifted into the channel, and enter the shift registers.

2. The gate is switched off, thereby breaking the feedback connections.

3. The contents of the shift register are read out into the channel.

Calculation of the Syndrome 校正子计算

$$s = rH^T$$

- If the syndrome is zero, there are no transmission errors in the received word.
- If the syndrome is nonzero, the received word contains transmission errors that require correction.

In the case of a cyclic code in systematic form, the syndrome can be calculated easily.

Syndrome polynomial 校正子多项式

Suppose: Code word $(c_0, c_1, \cdots c_{n-1})$

Received code $(r_0, r_1, \cdots r_{n-1})$

Then, the received code polynomial

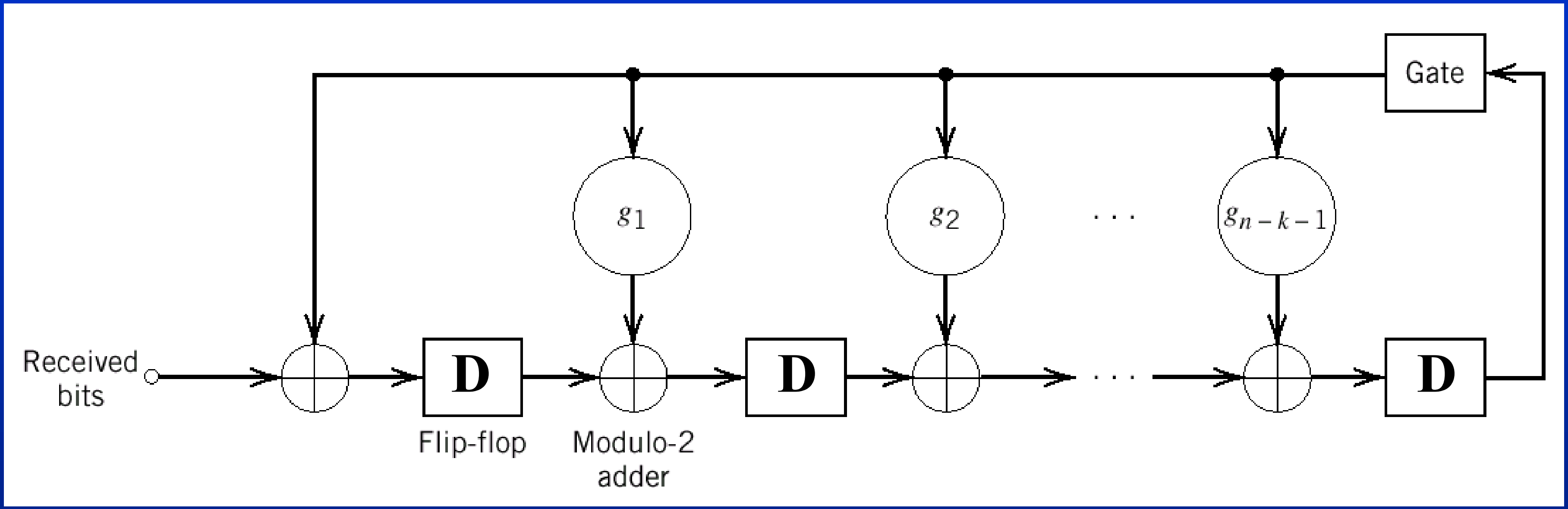
$$r(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$$

$\dfrac{r(X)}{g(X)}$ → Quotient 商 q(X)
Remainder 余式 s(X)

$$r(X) = q(X)g(X) + s(X) \qquad (10.47)$$

余式s(X) 就是校正子（伴随式）多项式。其系数就是校正子s。

Fig.10.9 正子计算器
Syndrome Calculator for (n,k) Cyclic Code



- Syndrome calculator is identical to the encoder.
- As soon as all the received bits have been shifted into the shift register, its contents are the syndrome S.

Example 10.3 Hamming Cyclic Codes

汉明码和循环码是两个不同的概念。根据定义，应该可以找到一类码，既是汉明码又是循环码。如（7，4）汉明循环码。

汉明码的定义条件

$n = 2^m - 1$ $k = 2^m - m - 1$ $n - k = m$

现在的关键问题是找到生成多项式*g(X)*，构成循环码。

Factorizing (因式分解）x^7+1 into three irreducible polynomials（不可约多项式）：

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

Primitive polynomial 本原多项式

一个(n, k)线性分组码，如果n=2^m-1, 对 X^n+1进行因式分解，得到若干个不可约多项式，其中阶次正好为*m*的多项式称为本原多项式。用本原多项式作为生成多项式构成的循环码就是汉明循环码。

In this example, we have two primitive polynomials (两个本原多项式):

$$\boxed{1+X^2+X^3} \qquad \boxed{1+X+X^3}$$

因此，我们可以构造两种（7，4）汉明循环码。

Here, we take $\boxed{g(X)=(1+X+X^3)}$

Hence, parity-check polynomial

$$h(X)=\frac{X^7+1}{g(x)}=(1+X)(1+X^2+X^3)$$
$$=1+X+X^2+X^4$$

?

Suppose: message sequence 1001

Determine: the whole code word xxx1001

Message polynomial

$$m(X)=1+X^3 \Longrightarrow X^{n-k}m(X)=X^3+X^6$$

$$\Downarrow$$

$$\frac{X^3+X^6}{1+X+X^3}=X+X^3+\frac{X+X^2}{1+X+X^3}$$

Quotient $a(X)=X+X^3$ **remainder** $b(X)=X+X^2$

Code polynomial

$$c(X)=b(X)+X^{n-k}m(X)=X+X^2+X^3+X^6$$

Code word **0111001**

?

Suppose: g(X)=1+X+X^3 is given
Determine: G and H

$$G(X)=\begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{k-1}g(X) \end{bmatrix}$$

$$g(X)=1+X+X^3$$

$$Xg(X)=X+X^2+X^4$$

$$X^2g(X)=X^2+X^3+X^5$$

$$X^3g(X)=X^3+X^4+X^6$$

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

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

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

Non-systematic form **Systematic form**

If $m=(1001) \longrightarrow C=0111001$

$$G = [P \vdots I_k] \qquad (10.12)$$

$$H = [I_{n-k} \vdots P^T] \quad (10.14)$$

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

?

Suppose: $g(X)=1+X+X^3$ is given

Determine: encoder and syndrome calculator

$$g(x)=1+\sum_{i=1}^{n-k-1} g_i X^i + X^{n-k}$$

$g(X)=1+X+X^3$ ⟶ $g_1=1, \quad g_2=0$

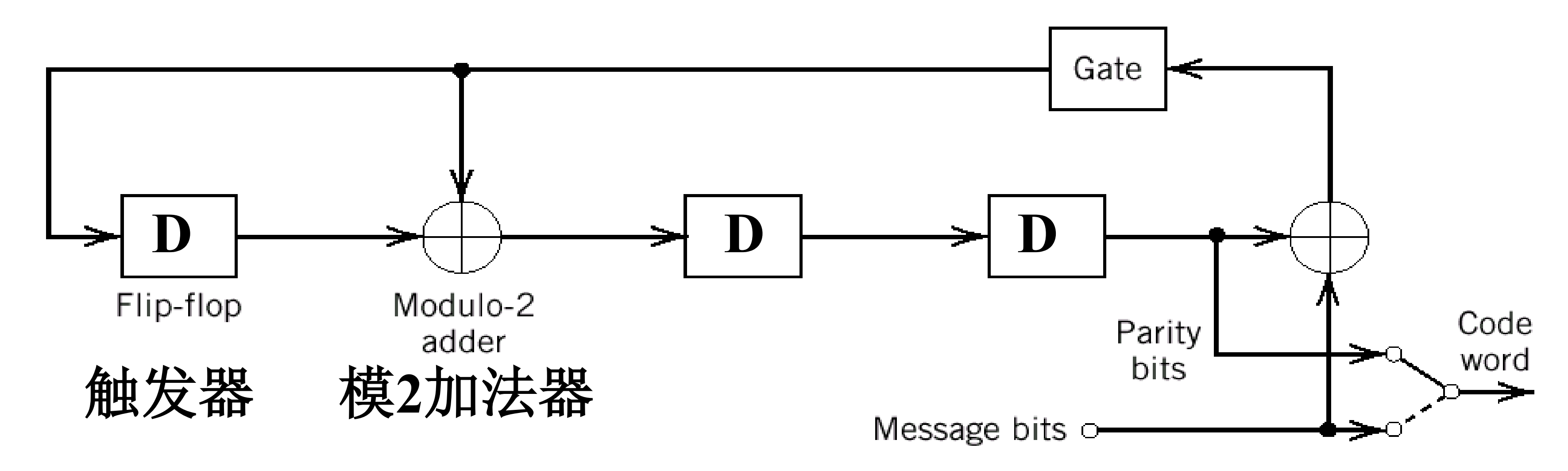
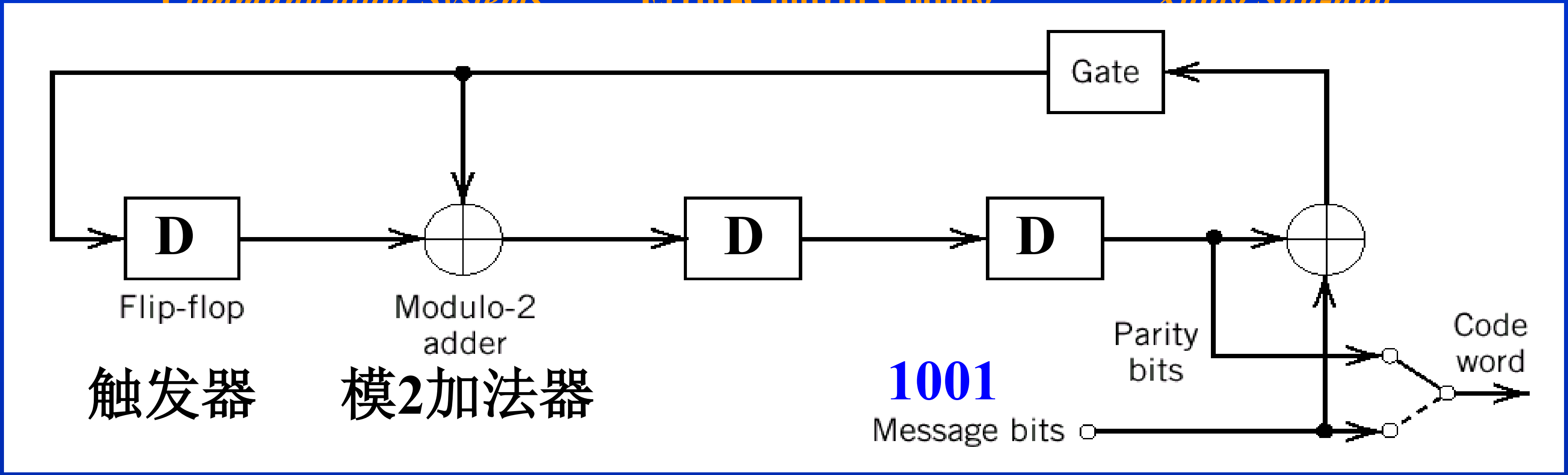


Figure 10.10 Encoder for the (7, 4) cyclic code generated by $g(X)=1+X+X^3$.



shift	input	Register contents
		0 0 0 initial state
1	1	1 1 0
2	0	0 1 1
3	0	1 1 1
4	1	0 1 1 parity bits

1001

↓

0111001

?

Suppose: received code is 0110001

Determine: is it correct? If it is in error, what is the correct code word?

$$s = rH^T = \begin{bmatrix}0110001\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 1 & 0\\ 0 & 1 & 1\\ 1 & 1 & 1\\ 1 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 1 & 0\end{bmatrix}$$

$$s \neq 0$$

So, the received code vector is wrong.

Syndrome calculator

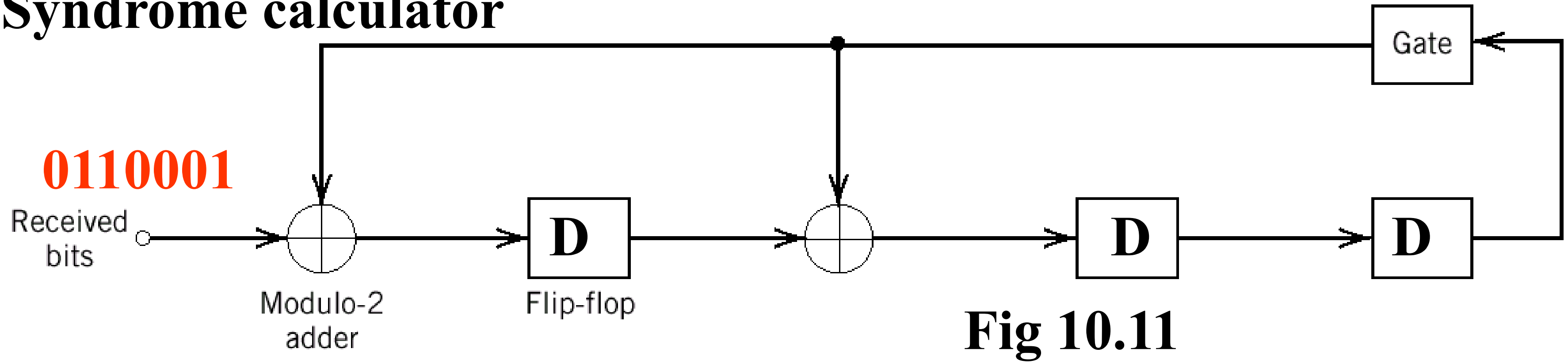


Fig 10.11

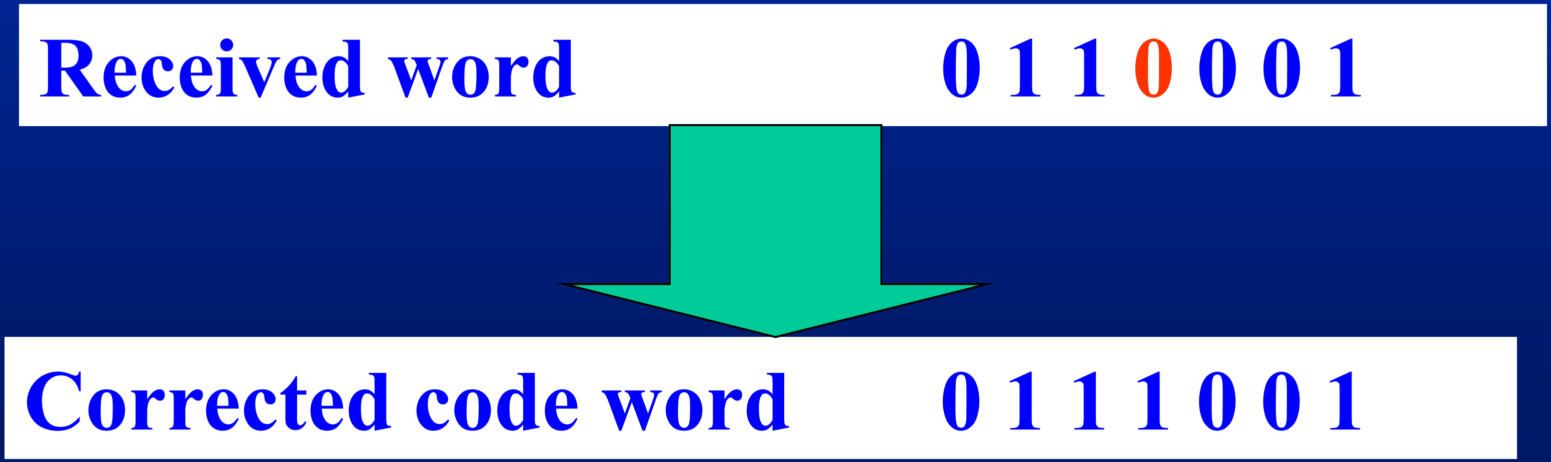
Shift	Input	Register contents
		0 0 0 (initial state)
1	1	1 0 0
2	0	0 1 0
3	0	0 0 1
4	0	1 1 0
5	1	1 1 1
6	1	0 0 1
7	0	1 1 0 syndrome

$$s \neq 0$$

So, the received code vector is wrong.

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

- **The syndrome is the same as the 4th column of parity-check matrix H, so the 4th bit is wrong.**



Received word 0 1 1 0 0 0 1

Corrected code word 0 1 1 1 0 0 1

Other cyclic codes

- Cyclic redundancy check codes CRC 循环冗余校验码
- Bose-Chaudhuri-Hocquenghem BCH 码
- Reed-Solomon Codes RS 里德-索罗蒙码

Other error-control codes

- Convolutional codes 卷积码
- Turbo codes
- Low-density parity-check codes 低密度奇偶校验码

Summary for this chapter

- Principle of error-control coding 差错控制编码原理
- Linear block codes 线性分组码
- Cyclic codes 循环码

Homework

10.4

10.10

10.12