数据通信

差错检测与纠正

殷亚凤
yafeng@nju.edu.cn
http://cs.nju.edu.cn/yafeng/
Room 901, Building of CS

差错检测与纠正

1. **差错类型**
2. 差错检测
3. 奇偶校验
4. 因特网检验和
5. 循环冗余检验
6. 前向纠错

差错类型

不可靠通信信道、传输损伤、噪声等使得差错总是存在，导致被传输的数据块中有**一个或多个比特发生改变**。

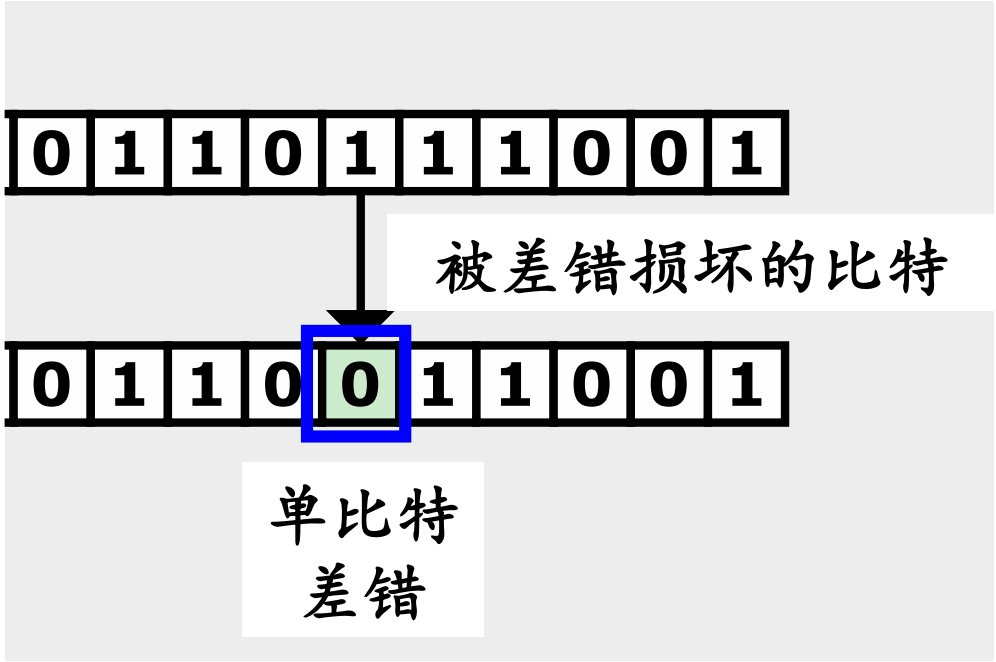
单比特错误

- 是一种孤立的差错状态
- 只改变**一个比特**，并不影响临近的其他比特

突发性错误

- 连续的B比特序列中的**第一个**和**最后一个比特**，以及**任意多个中间比特**都接收不正确

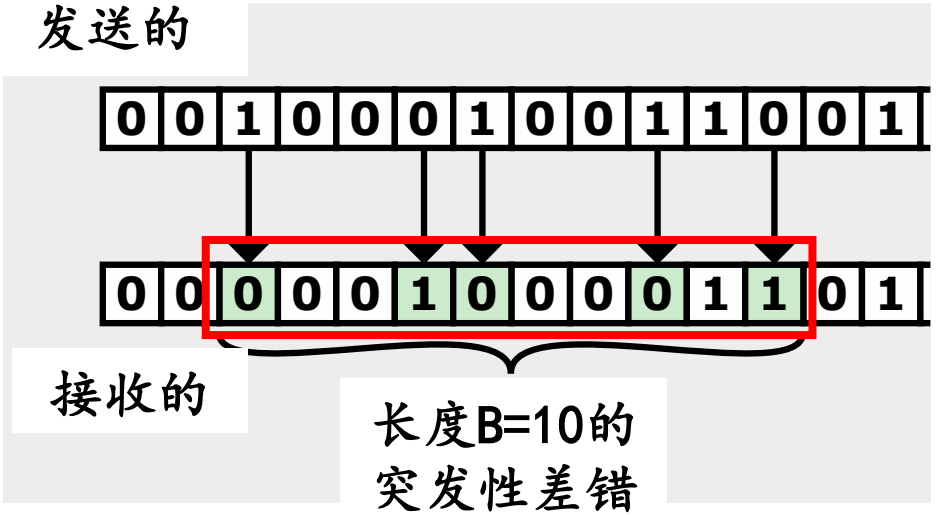
单比特差错



- **单比特差错**的出现可能是由于白噪声引起的，也就是说信噪比偶然出现程度不大的恶化，但这种恶化足以导致接收器在判断某一个比特的值时出现失误。

突发性差错

- **差错突发**：在一组比特中，任意两个**连续的差错比特之间**间隔的**正确比特数总是小于*x***，*x*为一个给定值。因此在差错突发时的最后一个差错比特与下一次突发的第一个差多比特之间会有*x*个以上的正确比特间隔。



- 突发性差错可能是由**冲激噪声**引起的，也可能是由移动无线环境中的**信号衰落**引起的。
- **数据率**越高，突发性差错影响越大

差错检测与纠正

1. 差错类型
2. **差错检测**
3. 奇偶校验
4. 因特网检验和
5. 循环冗余检验
6. 前向纠错

差错检测

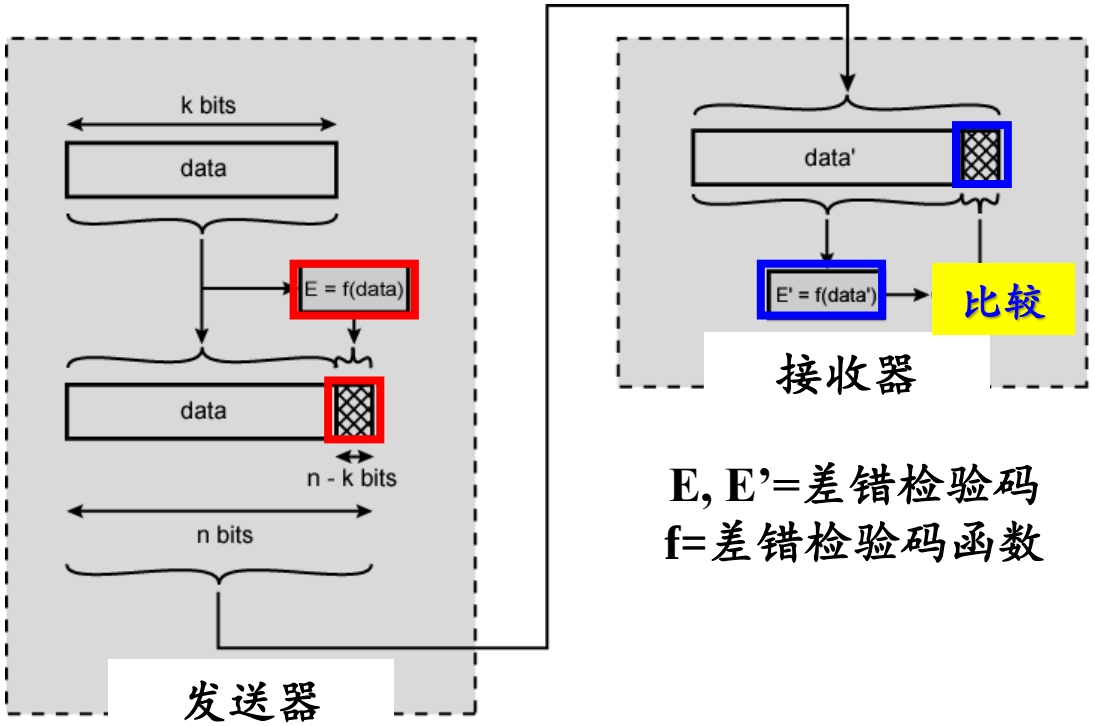
- 无论传输系统如何设计，差错总会存在，它会导致传输帧中一个或多个比特被改变。
- 数据以一个或多个连续的比特序列传送，称为**帧**。
- 与传输帧的差错有关的**概率值**：
  - P_b：接收到一个差错比特的概率，也成为比特差错率；
  - P_1：无比特差错的帧的到达概率；
  - P_2：在使用某种差错检测算法的情况下，含有一个或多个未检测到的比特差错的帧的到达概率；
  - P_3：在使用某种差错检测算法的情况下，含有一个或多个未检测到的比特差错，并且没有未检测到的比特差错的帧的到达概率。

差错检测

- 例6.2
- ISDN（综合业务数字网）连接规定要达到的目标是在64kbps的信道中，一分钟时间内至少有90%的时间里，比特差错率小于10^{-6}。
- 现在假设我们只需要在连续工作的64kbps的信道上，每天最多允许出现一个未检测到的比特差错的帧，并且假设帧的长度为1000比特。
  - 通过计算可知，一天能够**传输的帧**的数据为$5.529*10^6$ k帧bps，由此得出题目的所要求的**帧差错率为** $P_2=1/(5.529*10^6)=0.18*10^{-6}$。
  - 如果假设P_b**的值**为10^{-6}，那么$P_1=(0.999999)^{1000}=0.999$，并且因此得出$P_2=10^{-3}$。

差错检测

- 通过使用**差错检测码**来检测错误
- **发送端**添加到数据块后面一些额外的比特（**检验比特**）
  - k比特数据块，增加(n-k)比特检测码，(n-k)<k
  - **接收端**重新执行差错检测计算，并与接收到的**差错检测码比较**
  - 不同：检测到差错



E, E'=差错检验码
f=差错检验码函数

差错检测与纠正

1. 差错类型
2. 差错检测
3. **奇偶校验**
4. 因特网检验和
5. 循环冗余检验
6. 前向纠错

奇偶校验比特

- **奇偶校验**：使整个字符中1的个数为偶数（**偶校验**）或奇数（**奇校验**）

偶校验：偶数个1	**奇校验**：奇数个1
• 一般用于同步传输	• 一般用于异步传输

- 在数据块的末尾附加奇偶校验比特；
- 比如，字符传输，7比特字符1110001 → 附加1比特奇偶校验比特11110001（右边为字符的最低位，左边为最高位的奇偶校验比特）；
- 如果有一个（或任意奇数个）比特被错误地反转（如11100001），接收器将会检测出这个差错；
- 如果有两个（或任意偶数个）比特因错误而翻转，就会出现检测不到的差错

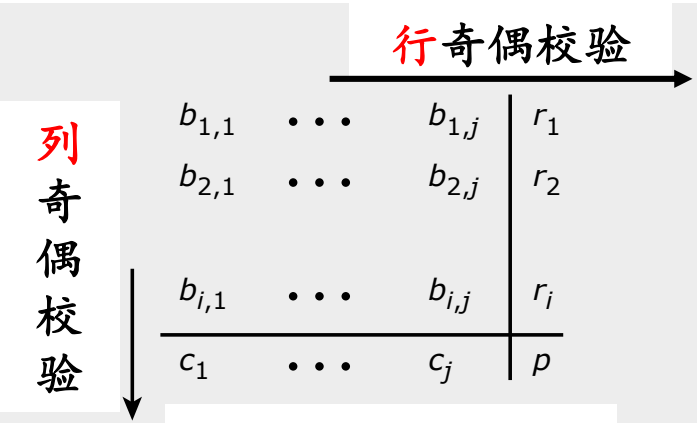
二维奇偶校验

- **二维奇偶校验**

1. **每一行**i上添加该行的偶校验比特r_i;
2. **每一列**j上添加该列的偶校验比特c_j;
3. **整体**添加一个奇偶校验比特p

→**i+j+1个**奇偶校验比特

- 每个比特都参与了两次奇偶校验;
- 更强的错误检测和一定的纠错能力;
- 矩形的四个错误无法检测。



(a) 奇偶校验的计算

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

(b) 无差错

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

行

列

(c) 可纠正单比特差错

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

(d) 不可纠正差错模式

差错检测与纠正

1. 差错类型
2. 差错检测
3. 奇偶校验
4. **因特网检验和**
5. 循环冗余检验
6. 前向纠错

因特网检验和

- **因特网检验和**是许多因特网协议所采用的差错检测码，包括IP，TCP，UDP
- 因特网检验和的计算利用了二进制**反码运算**以及**反码求和**
- **反码运算**：将数字1替换为数字0，将数字0替换为数字1
- **反码求和**：
  - 将两个数字视为无符号二进制整数，然后相加
  - 如果最左边有进位比特，则和再加1（循环进位）

```
  0011
+ 1100
------
  1111
```

```
  1101
+ 1011
------
 11000
+    1
------
  1001
```

因特网检验和

00 01 F2 03 F4 F5 F6 F7 00 00

发送方

局部和	0001 F203 F204
局部和	F204 F4F5 1E6F9
进位	E6F9 1 E6FA
局部和	E6FA F6F7 1DDF1
进位	DDF1 1 DDF2
对结果执行反码运算	220D

计算检验和

- 将检验和字段全部置为0;
- 将首部中所有的八位组执行反码求和;
- 将计算结果进行反码运算;
- 将结果存放在检验和字段中。

接收方

局部和	0001 F203 F204
局部和	F204 F4F5 1E6F9
进位	E6F9 1 E6FA
局部和	E6FA F6F7 1DDF1
进位	DDF1 1 DDF2
局部和	DDF2 220D FFFF

验证检验和

- 再次对所有八位组，包括检验和字段，进行反码求和;
- 如果结果为全1，则验证成功。

因特网检验和

- **相比于奇偶校验**，因特网校验和差错检测能力更强；
- 涉及简单的加法和比较运算，**开销小**；
- **和循环冗余检验相比**还是差很多；
- 链路层使用循环冗余校验（CRC）情况下，因特网校验和作为补充端到端的差错检测。

差错检测与纠正

1. 差错类型
2. 差错检测
3. 奇偶校验
4. 因特网检验和
5. **循环冗余检验**
6. 前向纠错

循环冗余检验

循环冗余检验(Cyclic Redundancy Check, CRC)

- 一种最常用也最有效的差错检验码
- 给定**k位的比特块**，发送器生成一个**n-k位**的比特序列，即**帧检验序列** (Frame Check Sequence, FCS)
- 含有**n比特的帧能被**一些预先设定好的数值**整除**
- 接收器用同样的数值对接收到的帧进行除法运算，若**没有余数**，则认为没有差错

模2运算

- **模2运算**是实现循环冗余检验的一种处理方法：

➢ 模2运算使用**无进位的二进制加法**，恰好是异或操作

$$\begin{array}{r} 1111 \\ +1010 \\ \hline 0101 \end{array} \qquad \begin{array}{r} 1111 \\ -0101 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 11001 \\ \times\ 11 \\ \hline 11001 \\ 11001 \\ \hline 101011 \end{array}$$

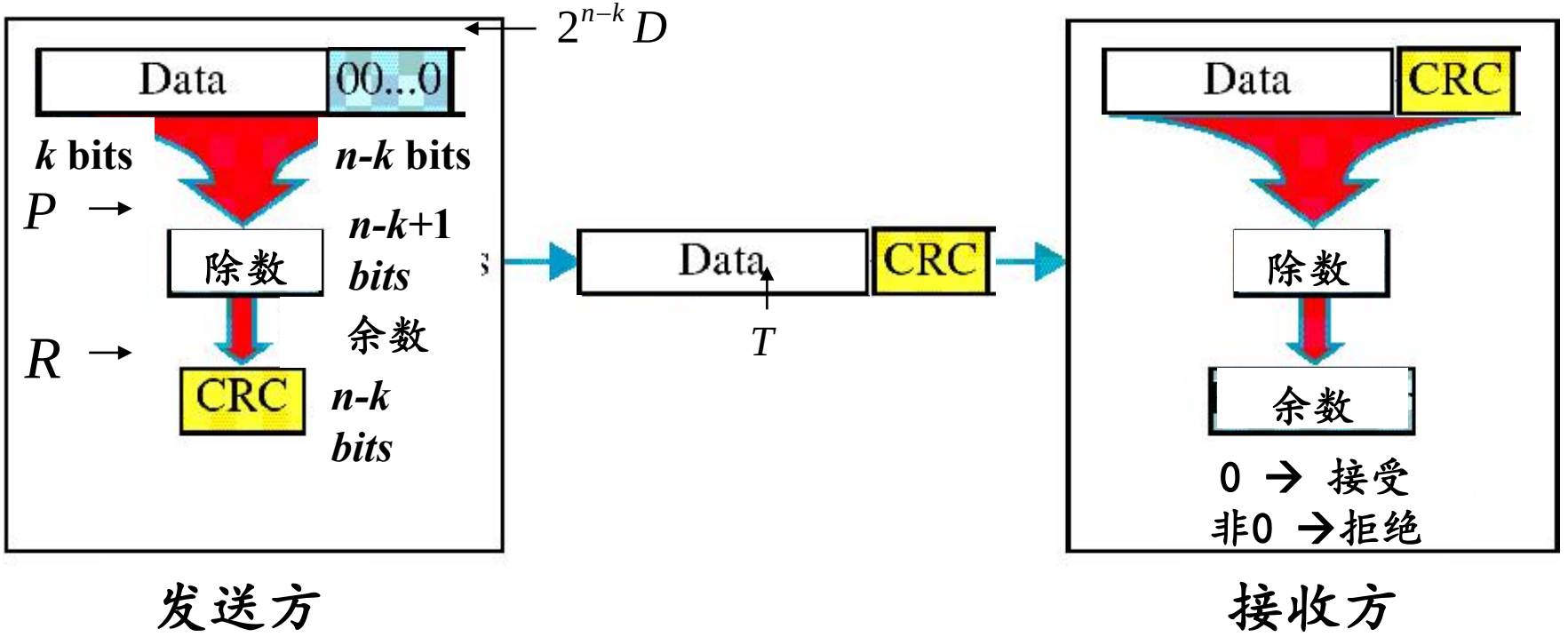
循环冗余检验

- T：要发送的n比特；
- D：k比特数据块；
- F：(n-k)比特检验序列；
- P：(n-k+1)比特的预定除数

$T = 2^{n-k}D + F$

问题：**如何选择F**，使得

$T\%P = 0$, i.e., $(2^{n-k}D + F)\%P = 0$



循环冗余检验

$$T = 2^{n-k}D + F$$

$$T\%P = 0$$

$$\frac{2^{n-k}D}{P} = Q + \frac{R}{P}$$

?

R

（摸2运算，R至少比P少1比特）

$$T = 2^{n-k}D + R$$

$$\frac{T}{P} = \frac{2^{n-k}D + R}{P} = Q + \frac{R+R}{P} = Q$$

用$2^{n-k}D$除以P，将其(n-k)比特的余数作为F，从而生成T。

循环冗余检验

1. 给定

报文 $D = 1010001101$ (10 bits)

预定比特序列 $P = 110101$ (6 bits)

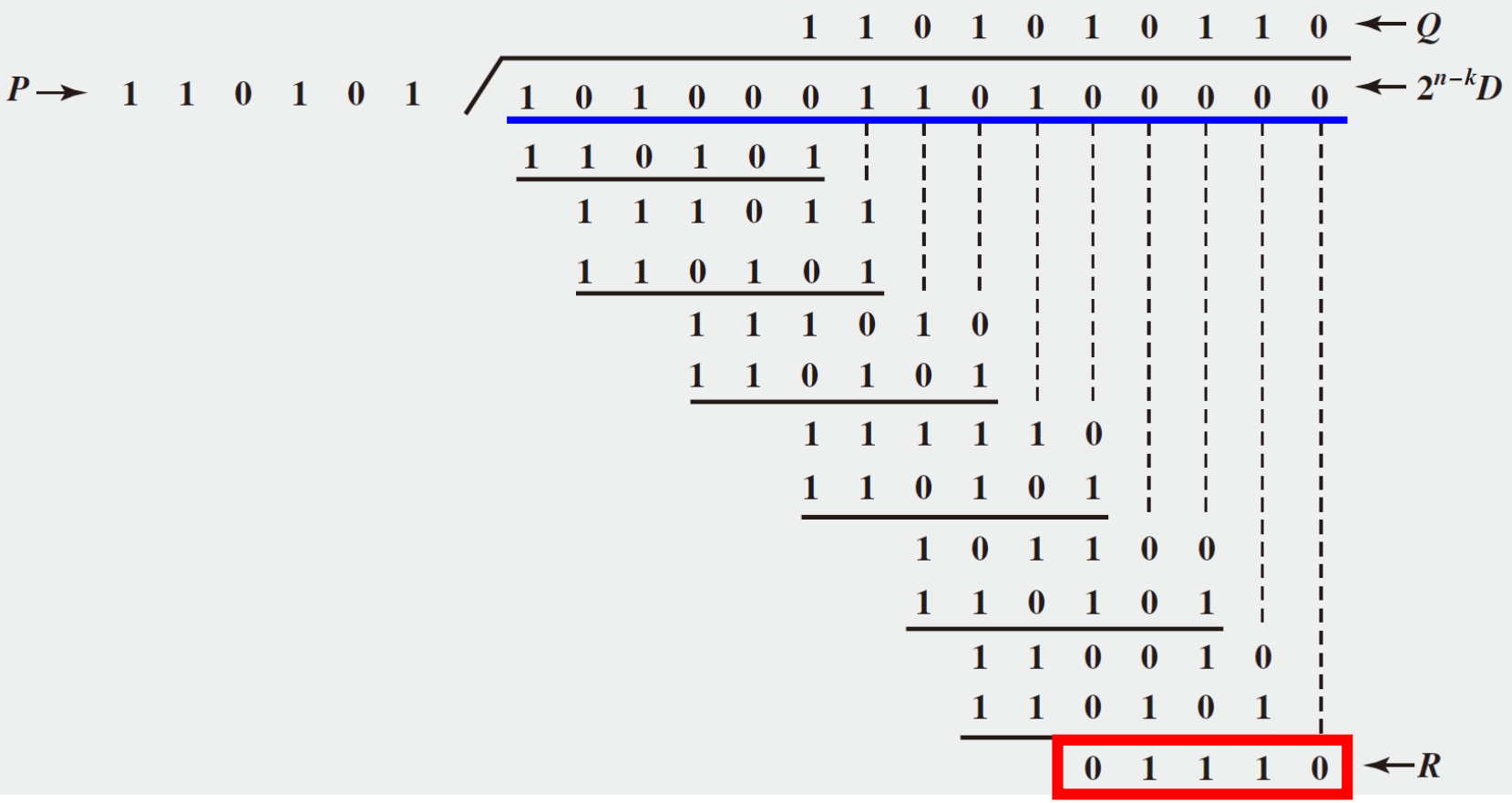
帧检验序列 $R =$ to be calculated (5 bits)

Thus, $n = 15$, $k = 10$, and $(n - k) = 5$.

2. 报文乘以 2^5, yielding 101000110100000.

循环冗余检验

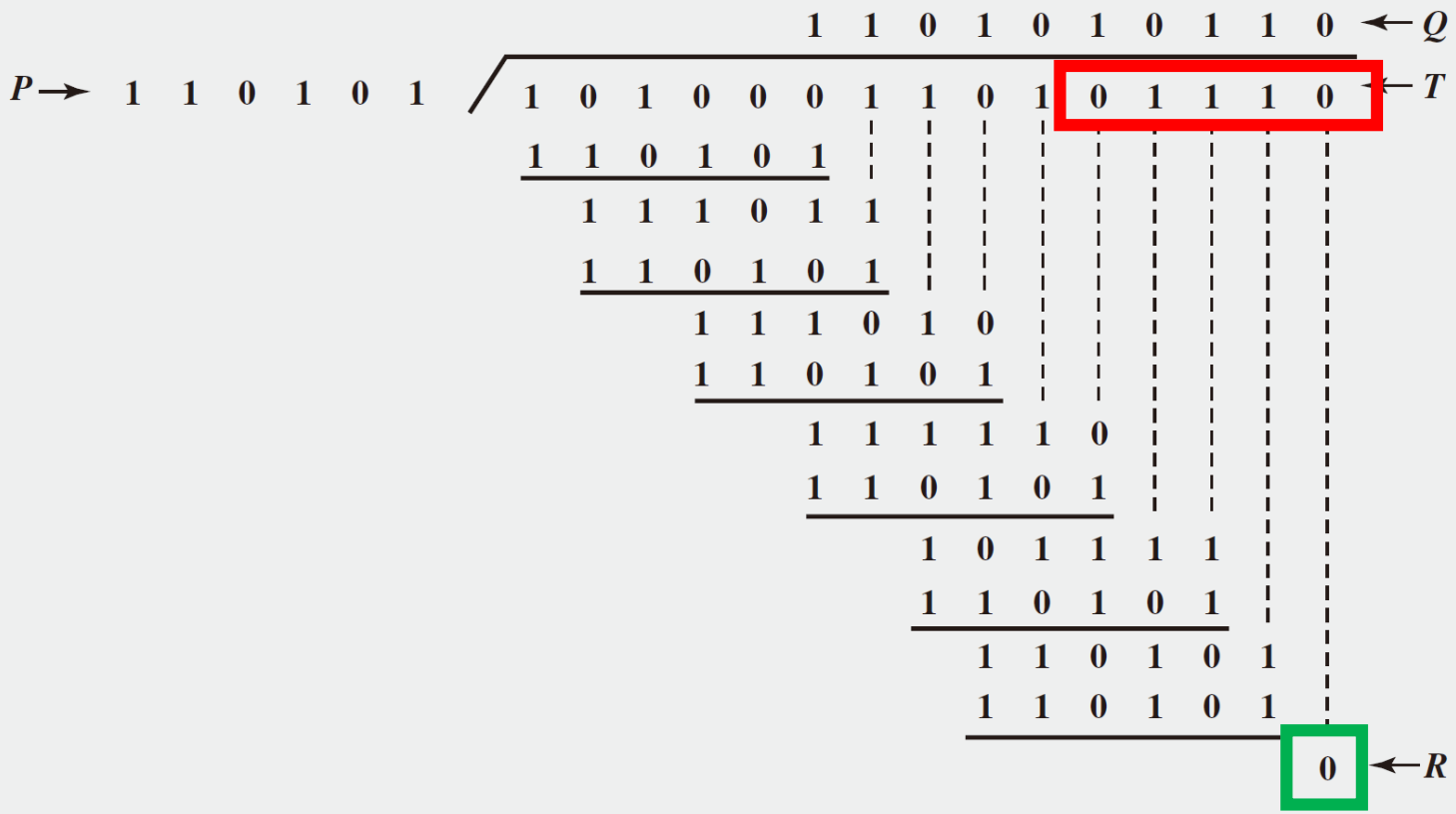
3. 得到的数值除以P：



循环冗余检验

4. 2^5D加上余数之后得到T=1010001101011110，T被发送。

5. 如果没有差错，则接收方的T原封未动，将其除以P：



由于**没有余数**，因此可以认为传输**没有差错**。

多项式

- **多项式表示：运算操作依然是模2运算**
  - D=110011　D(X)=X^5+X^4+X+1
  - P=11001　P(X)=X^4+X^3+1

$$\frac{X^3D(X)}{P(X)}=\frac{X^8+X^7+X^4+X^3}{X^4+X^3+1}$$

$$\frac{X^{n-k}D(X)}{P(X)}=Q(X)+\frac{R(X)}{P(X)}$$

$$T(X)=X^{n-k}D(X)+R(X)$$

D = 1010001101 → $D(X)=X^9+X^7+X^3+X^2+1$,

P = 110101, → $P(X)=X^5+X^4+X^2+1$.

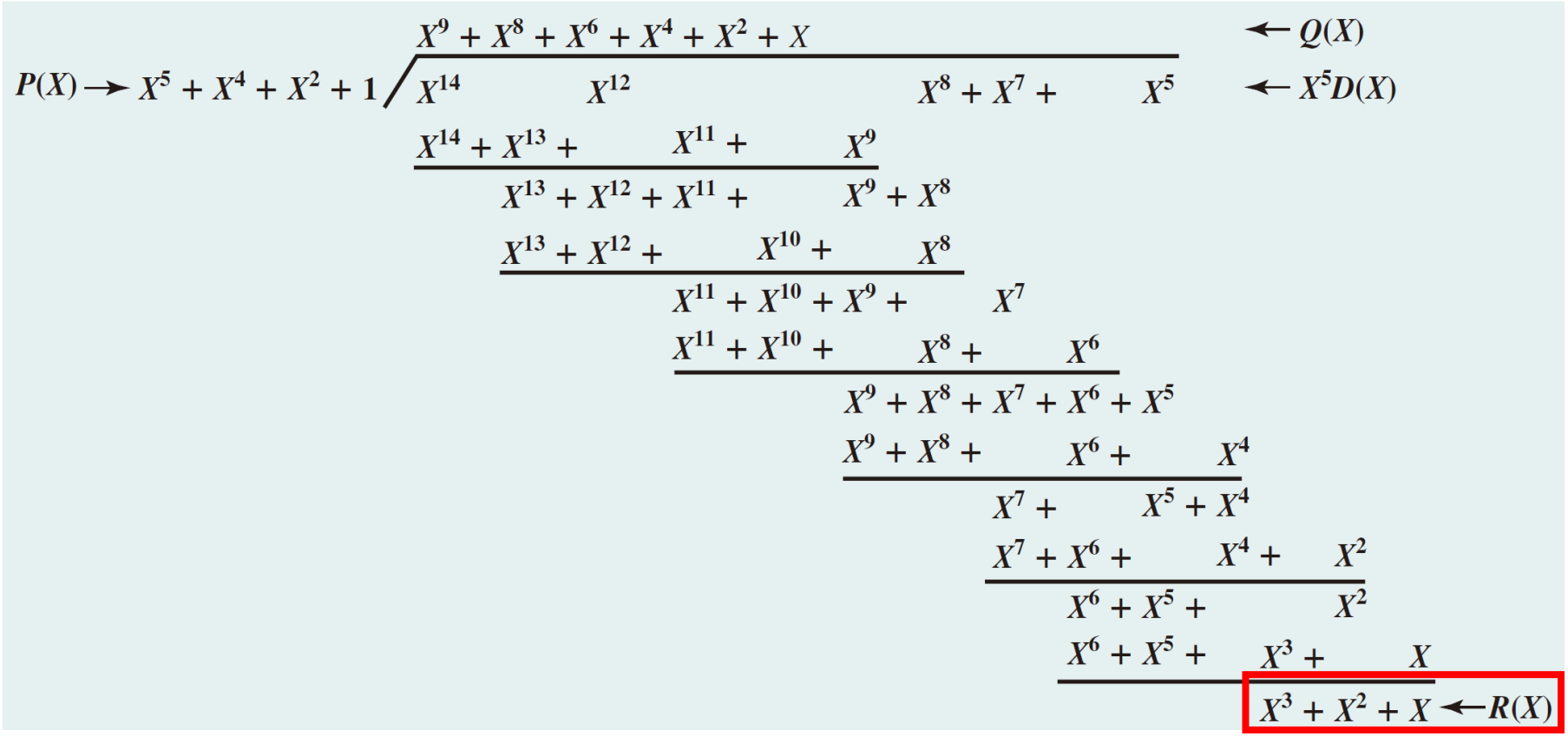
R = 01110, → $R(X)=X^3+X^2+X$.

多项式

D = 1010001101 → $D(X) = X^9 + X^7 + X^3 + X^2 + 1$,

P = 110101, → $P(X) = X^5 + X^4 + X^2 + 1$.

$X^{n-k}D(X) = X^5 D(X)$



（R = 01110）

多项式

- 检测差错等价于**选择P**使得**差错不能被P整除**。

- 广泛使用的P(X)序列：
  - CRC-12=$X^{12}+X^{11}+X^3+X^2+X+1=(X+1)(X^{11}+X^2+1)$
  - CRC-ANSI=$X^{16}+X^{15}+X^2+1=(X+1)(X^{15}+X+1)$
  - CRC-CCITT=$X^{16}+X^{12}+X^5+1=(X+1)(X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+1)$
  - IEEE-802= $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

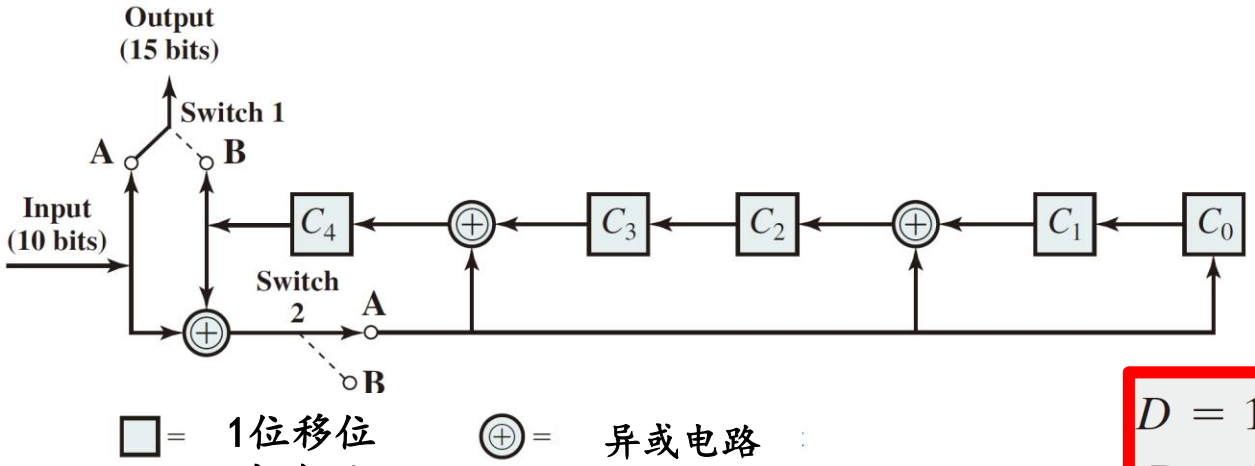
数字逻辑

- **CRC过程**可以由一些异或门和移位寄存器组成的除法电路实现：
  - 寄存器含有n-k比特，等于帧检验序列FCS长度；
  - 共有n-k个异或门；
  - 异或门是否存在，对应于多项式除数P(X)中的某一项是否存在，除了项1和X^{n-k}

数据 $D = 1010001101;\quad D(X) = X^9 + X^7 + X^3 + X^2 + 1$

除数 $P = 110101;\quad P(X) = X^5 + X^4 + X^2 + 1$

数字逻辑



(a) 移位寄存器的实现

$D = 1010001101;$
$P = 110101;$

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	I = input
Initial	0	0	0	0	0	1	1	1	1
Step 1	1	0	1	0	1	1	1	1	0
Step 2	1	1	1	1	1	1	1	0	1
Step 3	1	1	1	1	0	0	0	1	0
Step 4	0	1	0	0	1	1	0	0	0
Step 5	1	0	0	1	0	1	0	1	0
Step 6	1	0	0	0	1	0	0	0	1
Step 7	0	0	0	1	0	1	0	1	1
Step 8	1	0	0	0	1	1	1	1	0
Step 9	1	0	1	1	1	0	1	0	1
Step 10	0	1	1	1	0				

要发送的报文

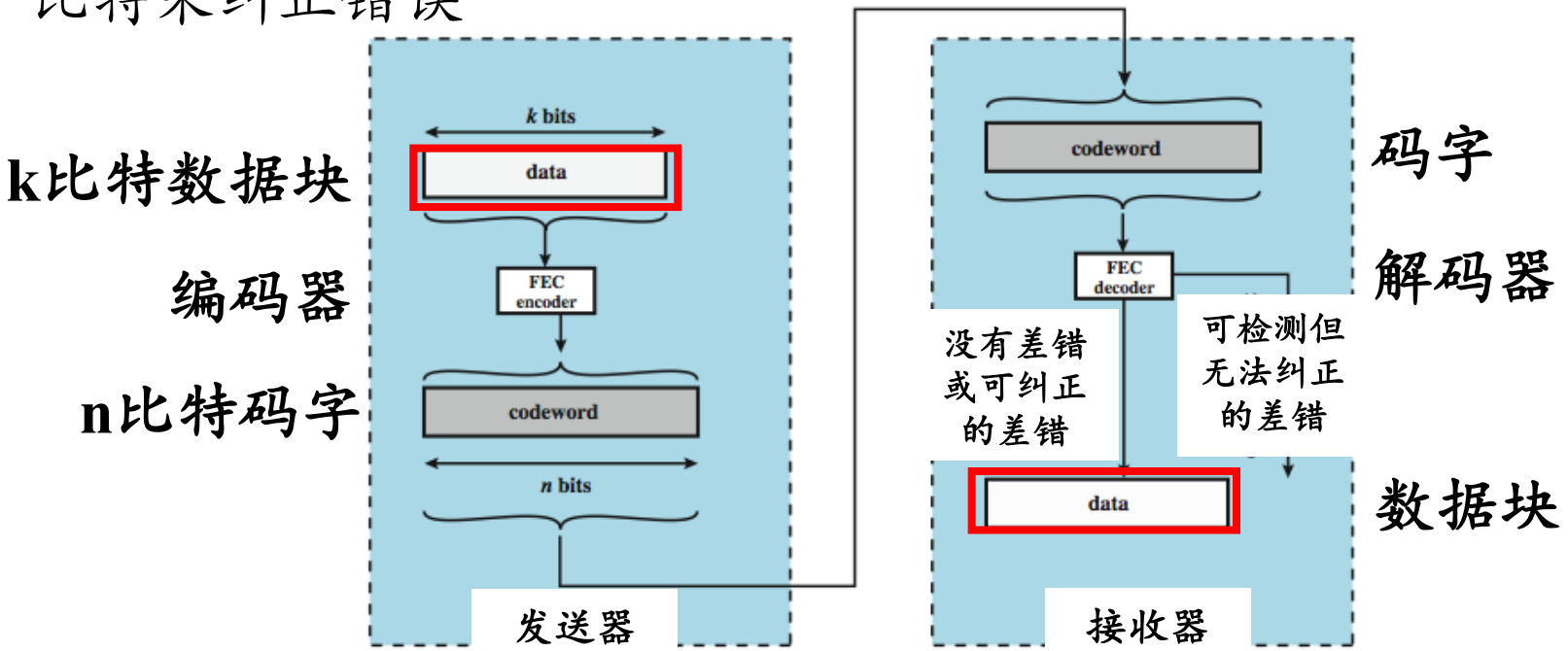
除以多项式$X^5+X^4+X^2+1$的移位寄存器电路

差错检测与纠正

1. 差错类型
2. 差错检测
3. 奇偶校验
4. 因特网检验和
5. 循环冗余检验
6. **前向纠错**

差错纠正

- **差错检测**需要数据块重传机制
- **差错重传**不适用与信道状态较差的链路
  - 无线链路：高比特差错率导致大量重传
  - 卫星链路：长传播时延导致效率低下
- **前向纠错（FEC）**：接收器能够在接收过程中根据传输的比特来纠正错误



前向纠错

FEC得到的四种可能输出：

- **无差错**：没有比特差错，FEC解码输入与原码字一致，解码器生成原数据块
- **可检测，可纠正差错**：即使收到的数据块与被传输的码字不同，FEC解码器也能通过映射产生原数据块
- **可检测，不可纠正之差错**：解码器能够检测到差错，但无法纠正
- **不可检测之差错**：解码器未能检测到已经出现的差错，产生与原数据块不一致的k比特数据块

差错纠正流程

- 差错纠正通过在传输报文上附加**冗余信息**完成：冗余信息使接收器能够推算出原报文

- **块纠错码**：一种广泛使用的纠错码

将k比特块映射成n比特的码字

每个码字都是唯一的

如果接收到无效的码字，就选择与它最接近的合法码字

块码原理

- **汉明距离**：d(v1，v2)是指v1和v2之间不同比特的个数。

$$v_1 = 011011, \quad v_2 = 110001$$

$$d(v_1, v_2) = 3$$

011**0**1
1100**0**1

- 将k比特数据块映射成**n比特码字**；

数据块	码字
00	00000
01	00111
10	11001
11	11110

块码原理

- 如果接收到一个非法码字，那么**选择与它最近（最短距离）的合法码字**。

00100

$d(00000, 00100) = 1;$

$d(11001, 00100) = 4;$

$d(00111, 00100) = 2;$

$d(11110, 00100) = 3$

01010

d(00000, 01010)=2;

d(11001, 01010)=3;

d(00111, 01010)=3;

d(11110, 01010)=2;

?

块码原理

- (n，k)块码：共有2^n个码字，其中合法码字为2^k个
- (n-k)/k：编码的**冗余度**
- k/n：**编码率**
- 假设一个编码由码字$w_1, w_2, \ldots w_s$组成，其中$s=2^n$，则编码的**最短距离**为为：

$$d_{\min} = \min_{i \neq j} [d(\mathbf{w}_i, \mathbf{w}_j)]$$

- 如果$d_{\min} \geq (2t+1)$，可纠正t个比特差错
- 如果$d_{\min} \geq 2t$，能检测t比特差错，纠正小于等于t-1个比特差错

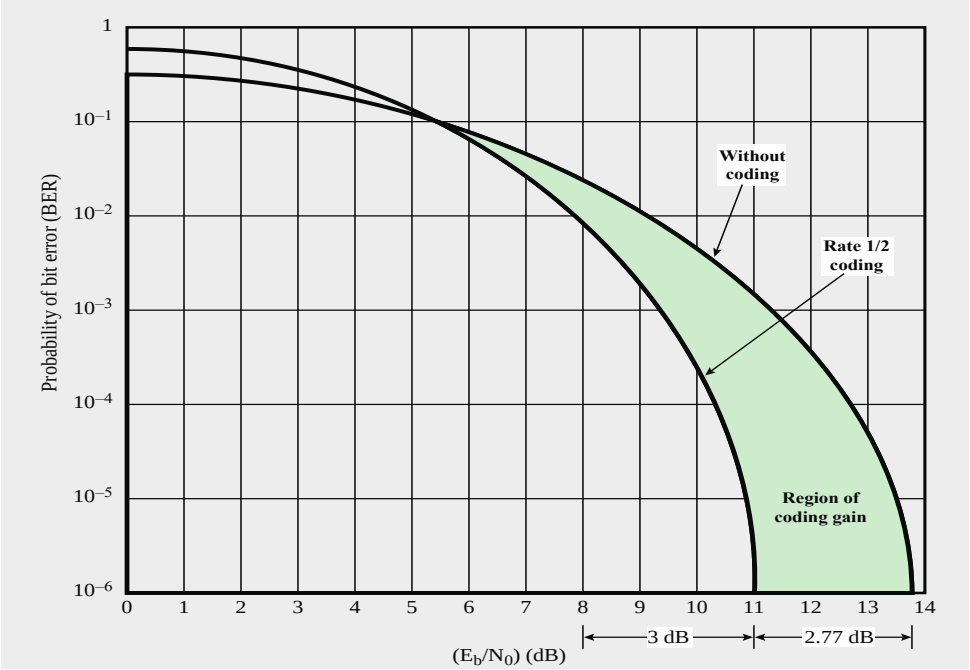
纠正： $t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$　　**检测：** $t = d_{\min} - 1$

纠错编码的基本原理

- 块码的设计要点
  - 对于给定的k和n，希望d_{min}尽可能大
  - 编码解码过程相对简单，内存开销和处理时间尽量小
  - 希望附加比特数$(n-k)$较少，减少带宽
  - 希望附加比特数$(n-k)$较多，减少差错率
    - 后两个目标相悖，需折衷考虑

纠错编码的基本原理

- 编码增益：如果使用相同的调制方法，达到相同的比特差错率时，有纠错码的系统与无纠错码的系统相比，所要求的E_b/N_0值减小。（E_b/N_0是每比特信号能量与每赫兹噪声功率密度的比值）



- 编码后误码率降低

课程习题（作业）

课本（截止日期：习题课前/4月21日晚23:55）：

6.6；　6.10(a)；　6.13(b)(c)；　6.17

提交方式：http://cslabcms.nju.edu.cn（本科教学支撑平台）

- 命名：学号+姓名+第*章。
- 若提交遇到问题请及时发邮件或在下一次上课时反馈。

课程习题（作业）

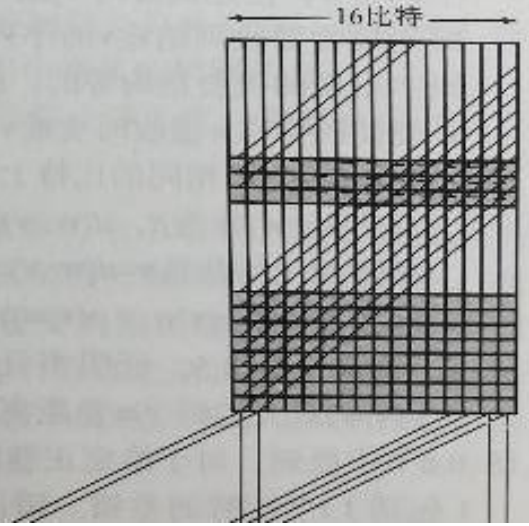
6.6 为数据块E3 4F 23 96 44 27 99 F3计算其因特网检验和。再进行验证计算。

6.7 因特网检验和的一个很好的特点是字节顺序无关性。小端字节序（little endian）的计算机在存储十六进制数时，最低有效字节在最后（如Intel处理器）。而大端字节序（big endian）的计算机将最低有效字节放在最前面（如IBM大型机）。请解释为什么检验和不需要考虑字节顺序？

6.8 高速运输协议（Xpress Transfer Protocol，XTP）使用了一个32比特的检验和函数，它被定义为两个16位函数的组合（concatenation）：XOR和RXOR，如图6.10所示。XOR函数计算列的奇偶校验。RXOR是一种对角奇偶校验，它先旋转数据块的每个连续16比特字中的1比特，然后再执行按位异或运算。

a. 这个检验和能检测出所有由奇数个差错比特引起的错误吗？请解释。

b. 这个检验和能检测出所有由偶数个差错比特引起的错误吗？如果不能，请给出将会导致检验失败的差错模式的特征。



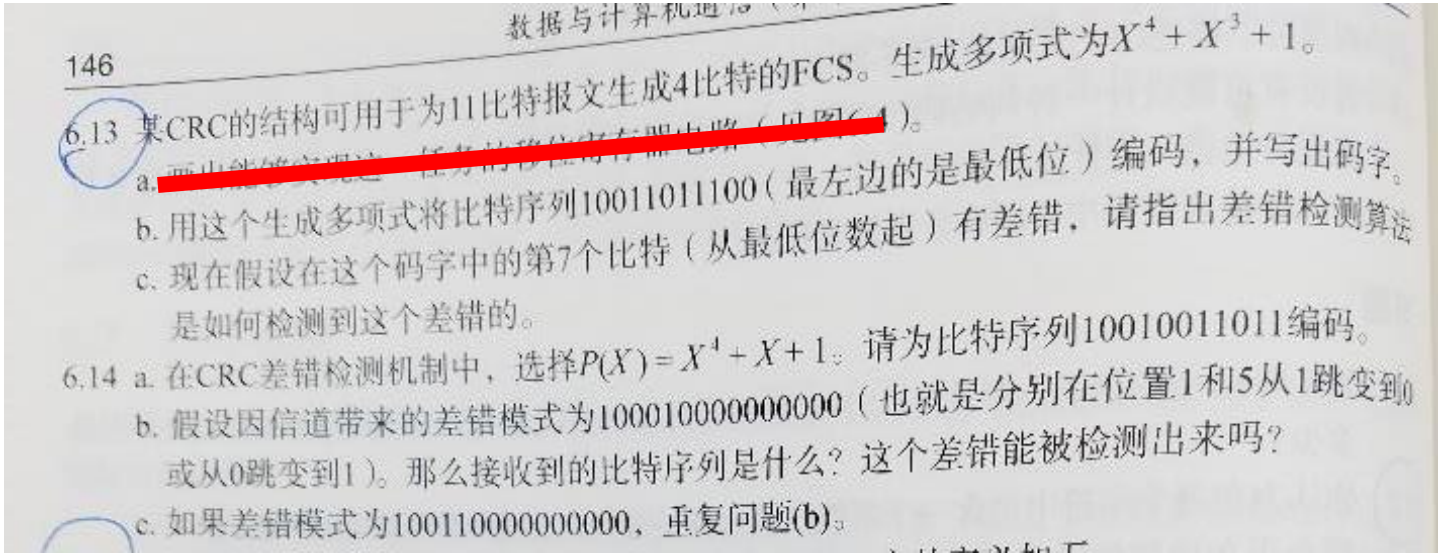
6.9 在计算FCS时，使用模2算法而不是二进制算法的意义是什么？

6.10 使用CRC-CCITT多项式，为一个1后面跟有15个0的报文计算生成的16比特CRC码。

a. 使用长除法。

b. 使用图6.9所示的移位寄存器电路。

课程习题（作业）



146 数据与计算机通信（

6.13 某CRC的结构可用于为11比特报文生成4比特的FCS。生成多项式为X^4+X^3+1。

a. [redacted]（见图6.4）。

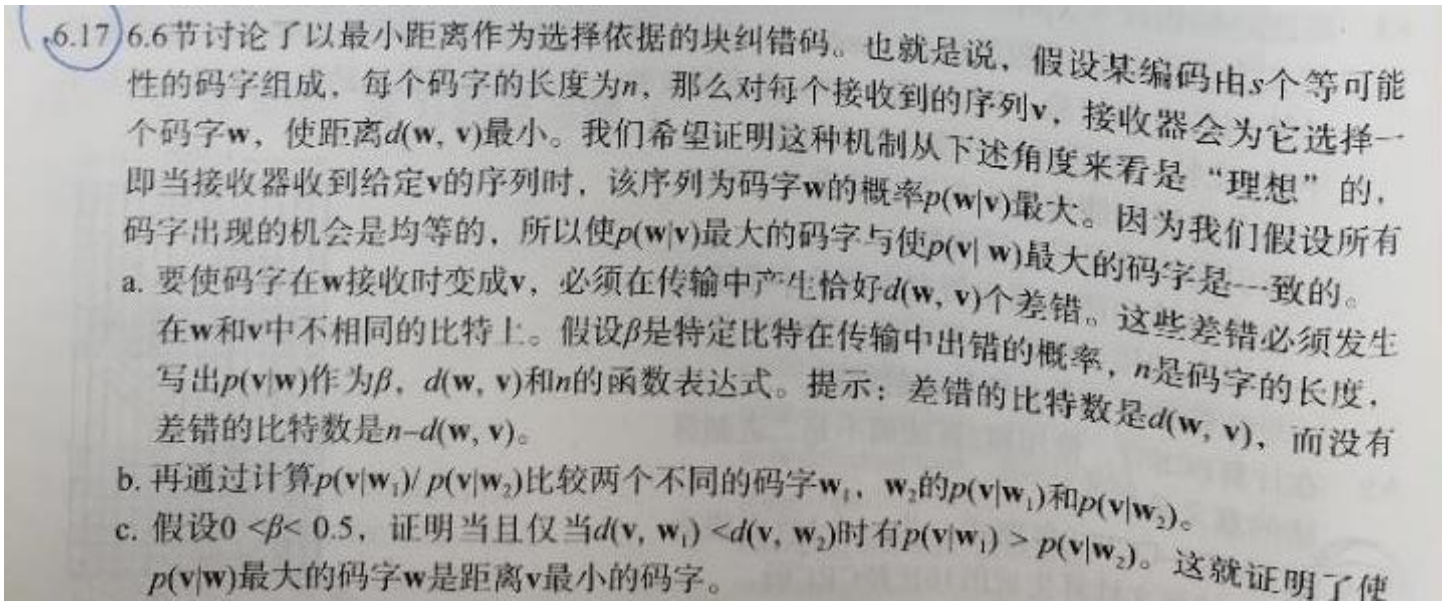
b. 用这个生成多项式将比特序列10011011100（最左边的是最低位）编码，并写出码字。

c. 现在假设在这个码字中的第7个比特（从最低位数起）有差错，请指出差错检测算法是如何检测到这个差错的。

6.14 a. 在CRC差错检测机制中，选择$P(X)=X^4+X+1$。请为比特序列10010011011编码。

b. 假设因信道带来的差错模式为1000100000000000（也就是分别在位置1和5从1跳变到0或从0跳变到1）。那么接收到的比特序列是什么？这个差错能被检测出来吗？

c. 如果差错模式为1001100000000000，重复问题(b)。



6.17 6.6节讨论了以最小距离作为选择依据的块纠错码。也就是说，假设某编码由s个等可能性的码字组成，每个码字的长度为n，那么对每个接收到的序列$\mathbf{v}$，接收器会为它选择一个码字$\mathbf{w}$，使距离$d(\mathbf{w}, \mathbf{v})$最小。我们希望证明这种机制从下述角度来看是“理想”的，即当接收器收到给定$\mathbf{v}$的序列时，该序列为码字$\mathbf{w}$的概率$p(\mathbf{w}|\mathbf{v})$最大。因为我们假设所有码字出现的机会是均等的，所以使$p(\mathbf{w}|\mathbf{v})$最大的码字与使$p(\mathbf{v}|\mathbf{w})$最大的码字是一致的。

a. 要使码字在$\mathbf{w}$接收时变成$\mathbf{v}$，必须在传输中产生恰好$d(\mathbf{w}, \mathbf{v})$个差错。这些差错必须发生在$\mathbf{w}$和$\mathbf{v}$中不相同的比特上。假设$\beta$是特定比特在传输中出错的概率，$n$是码字的长度，写出$p(\mathbf{v}|\mathbf{w})$作为$\beta$，$d(\mathbf{w}, \mathbf{v})$和$n$的函数表达式。提示：差错的比特数是$d(\mathbf{w}, \mathbf{v})$，而没有差错的比特数是$n-d(\mathbf{w}, \mathbf{v})$。

b. 再通过计算$p(\mathbf{v}|\mathbf{w}_1)/p(\mathbf{v}|\mathbf{w}_2)$比较两个不同的码字$\mathbf{w}_1$，$\mathbf{w}_2$的$p(\mathbf{v}|\mathbf{w}_1)$和$p(\mathbf{v}|\mathbf{w}_2)$。

c. 假设$0<\beta<0.5$，证明当且仅当$d(\mathbf{v}, \mathbf{w}_1)<d(\mathbf{v}, \mathbf{w}_2)$时有$p(\mathbf{v}|\mathbf{w}_1)>p(\mathbf{v}|\mathbf{w}_2)$。这就证明了使$p(\mathbf{v}|\mathbf{w})$最大的码字$\mathbf{w}$是距离$\mathbf{v}$最小的码字。

总结

问题？

殷亚凤
yafeng@nju.edu.cn
http://cs.nju.edu.cn/yafeng/
Room 901, Building of CS