

计算机网络编程

第4章 Ethernet帧的CRC校验

信息工程学院 方徽星
fanghuixing@hotmail.com

大纲

- 设计目的
- 相关知识
- 例题分析

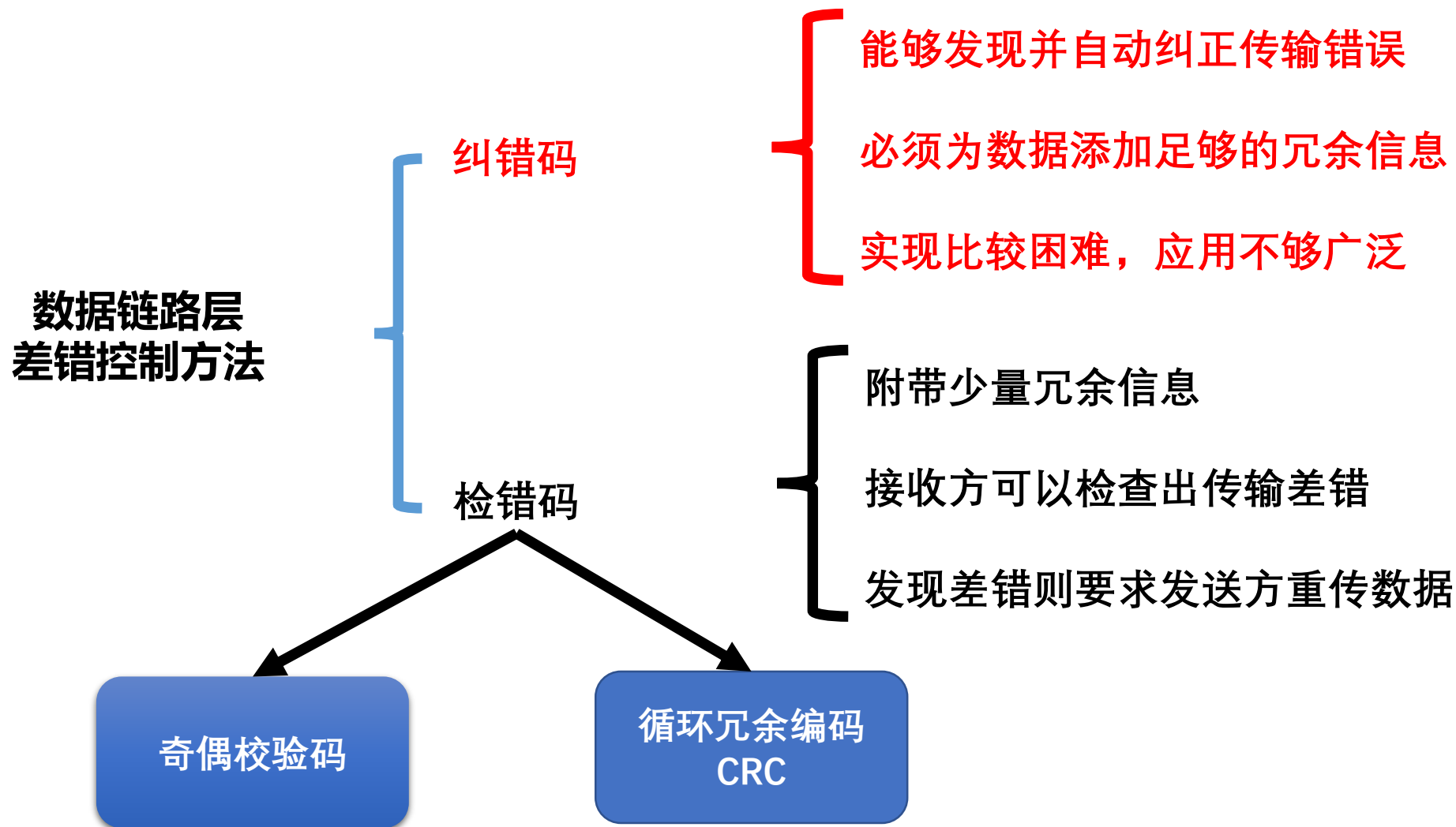
1. 设计目的

- 网络中的数据传输最终要**通过物理线路**完成，**无法保证数据传输不出错**
- 物理层上设计**数据链路层**：采用差错控制、流量控制等方法，将有差错的物理线路**改造成无差错的数据链路**
- 从而使数据链路层**向上面的网络层提供高质量的传输服务**

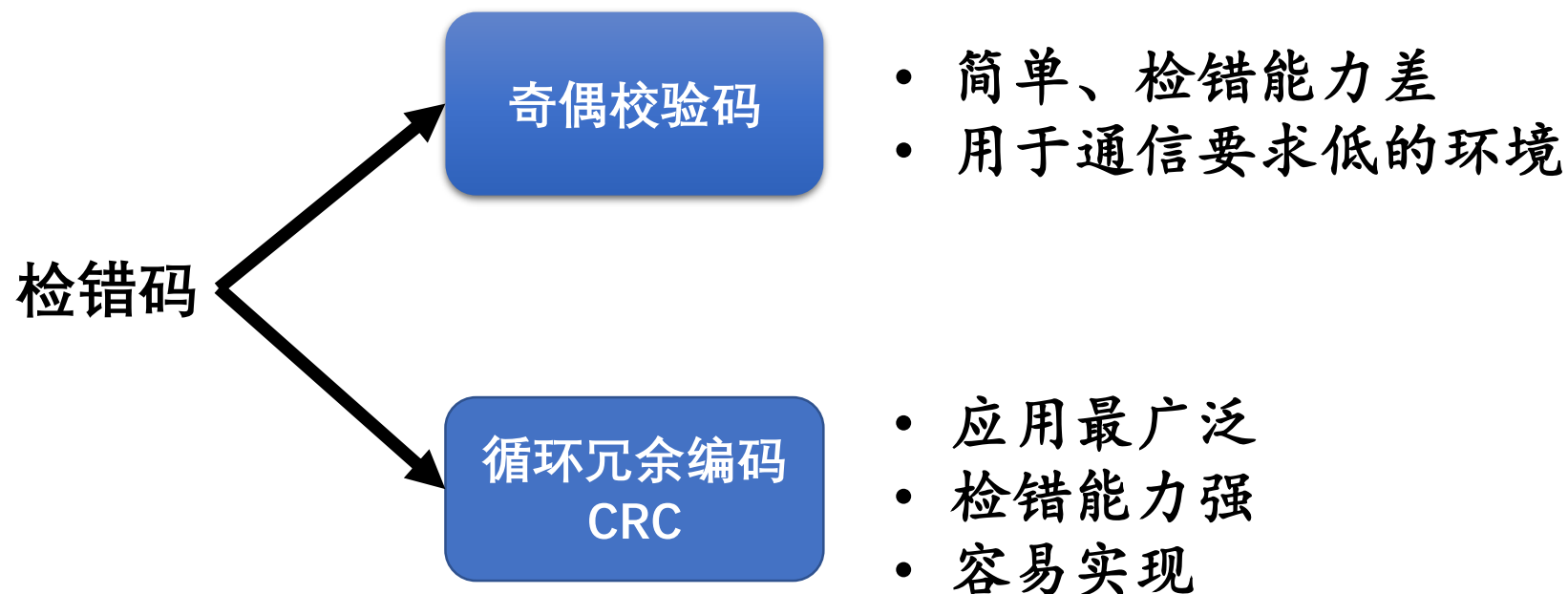
1. 设计目的

- 本章练习目的
 - 通过学习和编程实现Ethernet帧的CRC校验
 - 了解网络协议中校验和的计算过程与作用

2. 相关知识—CRC校验的概念

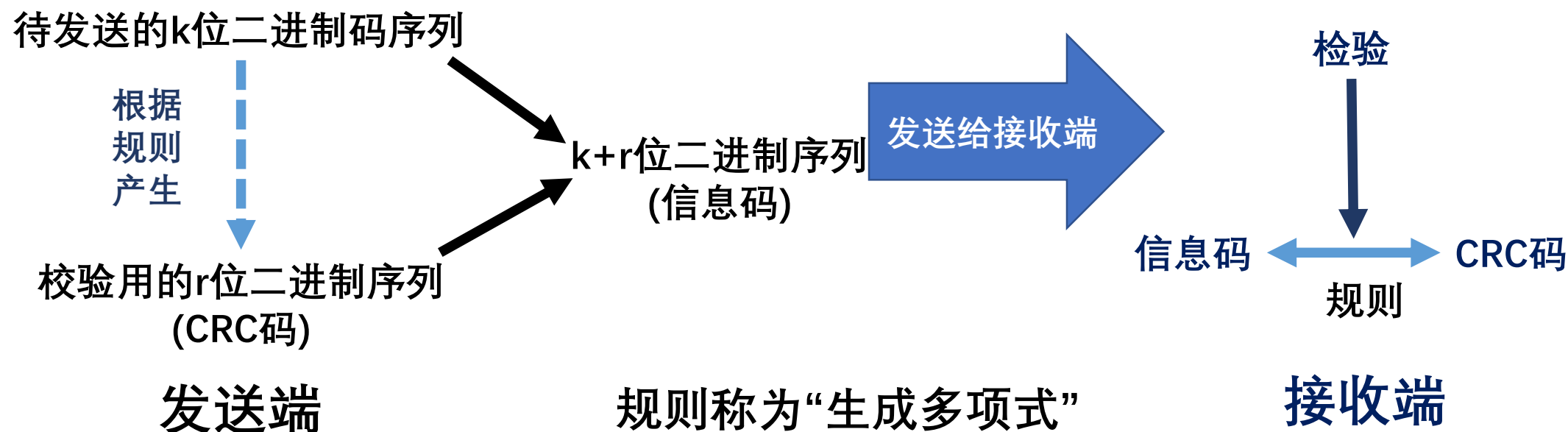


2. 相关知识—CRC校验的概念

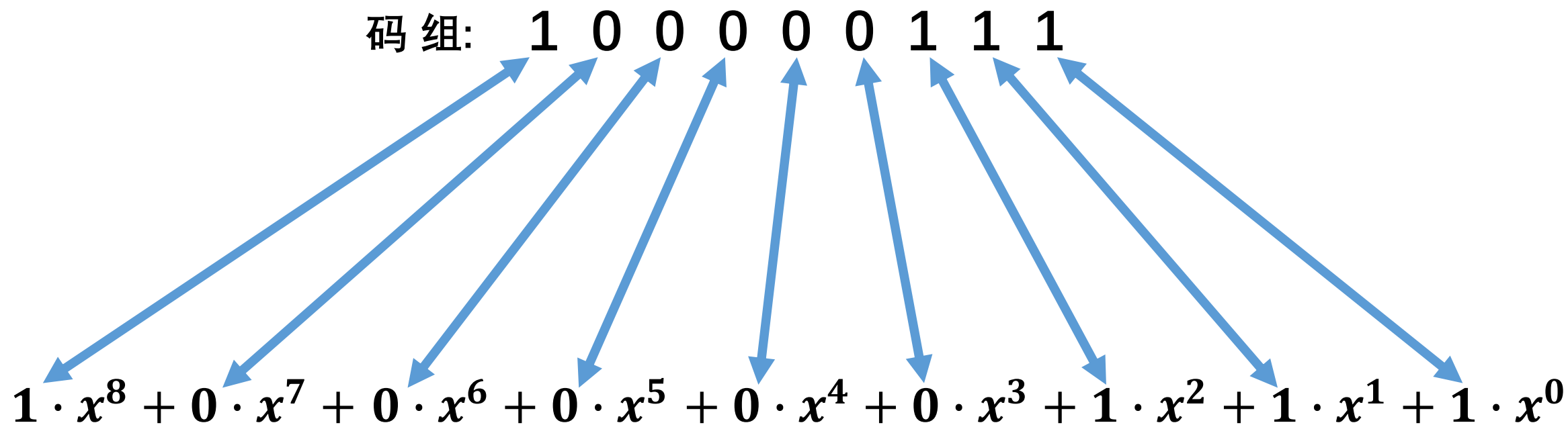


2. 相关知识—CRC校验的概念

CRC校验过程



2. 相关知识—CRC校验的概念



多项式: $x^8 + x^2 + x^1 + 1$

在代数编码理论中，可将一个码组表示为一个多项式，码组中的各个码元作为多项式的系数

2. 相关知识—CRC校验的概念

CRC生成多项式G(x)由不同的协议来定义，其结构和检错效果经过严格的数学分析与实践验证

| 名称 | 生成多项式 | 应用实例 |
|---------|---|--|
| CRC-4 | $x^4 + x + 1$ | ITU G.704 |
| CRC-8 | $x^8 + x^2 + x + 1$ | |
| CRC-12 | $x^{12} + x^{11} + x^3 + x + 1$ | |
| CRC-16 | $x^{16} + x^{12} + x^2 + 1$ | IBM SDLC（同步数据链路控制协议） |
| CRC-ITU | $x^{16} + x^{12} + x^5 + 1$ | ISO HDLC、ITU X.25、V.34/V.41/V.42 |
| CRC-32 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ | ZIP、RAR、IEEE 802.3、IEEE 802.6、IEEE 1394、PPP-FCS等 |

https://en.wikipedia.org/wiki/Cyclic_redundancy_check

2. 相关知识—CRC校验的概念

CRC校验的工作过程

(1) 发送端的发送数据多项式

$$\underline{F(x) \cdot x^k}$$

生成多项式
最高幂的值

意义：将发送数据比特序列
左移k位，以便放入余数

如：CRC-8的最高幂是8，则发送 $F(x) \cdot x^8$

2. 相关知识—CRC校验的概念

CRC校验的工作过程

(2) 将 $F(x) \cdot x^k$ 除以生成多项式 $G(x)$

$$\frac{F(x) \cdot x^k}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \quad \longleftarrow \text{余数多项式}$$

(3) $F(x) \cdot x^k + R(x)$ 将作为整体，从发送端通过通信信道发送到接收端

2. 相关知识—CRC校验的概念

CRC校验的工作过程

(4) 接收端对接收数据多项式将 $F'(x)$ 采用同样的运算

$$\frac{F'(x) \cdot x^k}{G(x)} = Q(x) + \frac{R'(x)}{G(x)} \quad \longleftarrow \text{(接收)余数多项式}$$

(5) 将 $R'(x)$ 与 $R(x)$ 比较, 判断是否存在传输出错

2. 相关知识—CRC校验的例子

- 实际的CRC校验采用二进制模2算法

- 减法不借位
- 加法不进位
- 是一种异或操作

模2加法和减法=XOR

| A | B | A xor B |
|---|---|---------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$$\begin{array}{r} 10011011 \\ + 11001010 \\ \hline 01010001 \end{array}$$

$$\begin{array}{r} 00110011 \\ + 11001101 \\ \hline 11111110 \end{array}$$

$$\begin{array}{r} 11110000 \\ - 10100110 \\ \hline 01010110 \end{array}$$

$$\begin{array}{r} 01010101 \\ - 10101111 \\ \hline 11111010 \end{array}$$

2. 相关知识—CRC校验的例子

- 实际的CRC校验采用二进制模2算法

- 减法不借位
- 加法不进位
- 是一种异或操作

模2乘法=AND

| A | B | A * B |
|---|---|-------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

2. 相关知识—CRC校验的例子

- 实际的CRC校验采用二进制模2算法

- 减法不借位
- 加法不进位
- 是一种异或操作

模2除法=“AND”

| A | B | A / B |
|---|---|-------|
| 0 | 1 | 0 |
| 1 | 1 | 1 |

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

1. 发送数据 $F(x)$ 为10010010 (8位)
2. 生成多项式 $G(x)$ 为100000111 (9位, $k = 8$)
3. 将发送数据 $F(x)$ 乘以 2^8 , 产生的乘积为1001001000000000
4. 将该乘积按模2除法除以生成多项式

$$G(x) \longrightarrow 100000111 \overline{1001001000000000} \longleftarrow F(x) \text{ 乘以 } 2^8$$

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

$$G(x) \longrightarrow 100000111 \overline{1001001000000000} \longleftarrow F(x) \text{乘以} 2^8$$

1

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

$$\begin{array}{r} G(x) \longrightarrow 100000111 \sqrt{\overset{1}{1001001000000000}} \longleftarrow F(x) \text{乘以} 2^8 \\ \underline{100000111} \\ 100011000 \end{array}$$

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

$$\begin{array}{r} \text{1001} \\ G(x) \longrightarrow 100000111 \sqrt{1001001000000000} \longleftarrow F(x) \text{乘以} 2^8 \\ \underline{100000111} \\ 100011000 \\ \underline{100000111} \\ 111110000 \end{array}$$

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

$$\begin{array}{r} \text{G}(x) \longrightarrow 100000111 \sqrt{\begin{array}{r} 1001001000000000 \\ 100000111 \\ \hline 100011000 \\ 100000111 \\ \hline 111110000 \\ 100000111 \\ \hline 11110111 \end{array}} \end{array}$$

$10010001 \longleftarrow Q(x)$

$\longleftarrow F(x) \text{ 乘以 } 2^8$

$\longleftarrow R(x)$

CRC校验码

少于9位 \longrightarrow

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

5. 将余数多项式 $R(x)$ 加到乘积中

1001001000000000

发送数据



(模2)减去

11110111

CRC校验码



1001001011110111

带CRC校验码的发送数据

2. 相关知识—CRC校验的例子

- 例子 (CRC-8)

$$G(x) \longrightarrow 100000111 \sqrt{1001001011110111} \begin{matrix} 10010001 \longleftarrow Q(x) \\ \longleftarrow F'(x) \end{matrix}$$

6. 如果数据在传输过程中没有出错，接收端收到的带CRC码的数据一定能被相同的生成多项式 $G(x)$ 整除

$$\begin{array}{r} 100000111 \\ \hline 100010111 \\ 100000111 \\ \hline 100000111 \\ 100000111 \\ \hline 0 \end{array}$$

2. 相关知识—CRC校验的主要特点

- CRC校验码可以检测出
 - 所有离散的1位错
 - 所有离散的2位错
 - 所有奇数位错
 - 所有小于或等于校验位长度的突发错(指几乎是连续发生的一串错)
 - 长度为 $(k + 1)$ 位突发错的概率为 $[1 - 0.5^{k-1}]$

原因：分析多项式除法;参考ANDREW S. TANENBAUM, DAVID J. WETHERALL: COMPUTER NETWORKS (5th), sec3.2, p213

3. 例题分析—设计要求

- 根据IEEE 802.3标准的Ethernet帧结构，编写程序来封装一个帧且进行CRC校验(求出余数)，并将帧的各个字段值写入文件
- 采用CRC-8校验
- 原始数据内容：Hello world!

3. 例题分析—设计要求

- 程序设计的具体要求

- 程序为命令行程序，如：可执行文件名为CrcEncode.exe，命令行格式为
CrcEncode output_file
- 将部分字段内容显示在控制台上，具体格式为

长度字段: xx (十六进制格式输出)

数据字段: ... (字符串格式输出)

帧校验字段: xx (十六进制格式输出)

3. 例题分析—设计要求

- 程序设计的具体要求
 - 有良好的编程规范与注释。编程所使用的操作系统、语言和编译环境不限，但是在提交的说明文档中需要加以注明
 - 撰写说明文档，包括程序的开发思路、工作流程、关键问题、解决思路以及进一步的改进等内容

3. 例题分析—关键问题

- CRC校验的过程

$$\begin{array}{r}
 10001001 \\
 100000111 \sqrt{1000101000000000} \\
 \underline{100000111} \\
 100110000 \\
 \underline{100000111} \\
 110111000 \\
 \underline{100000111} \\
 10111111
 \end{array}$$

源节点的CRC生成

假设数据为10001010

$$G(x) = x^8 + x^2 + x^1 + 1$$

3. 例题分析—关键问题

- CRC校验的过程

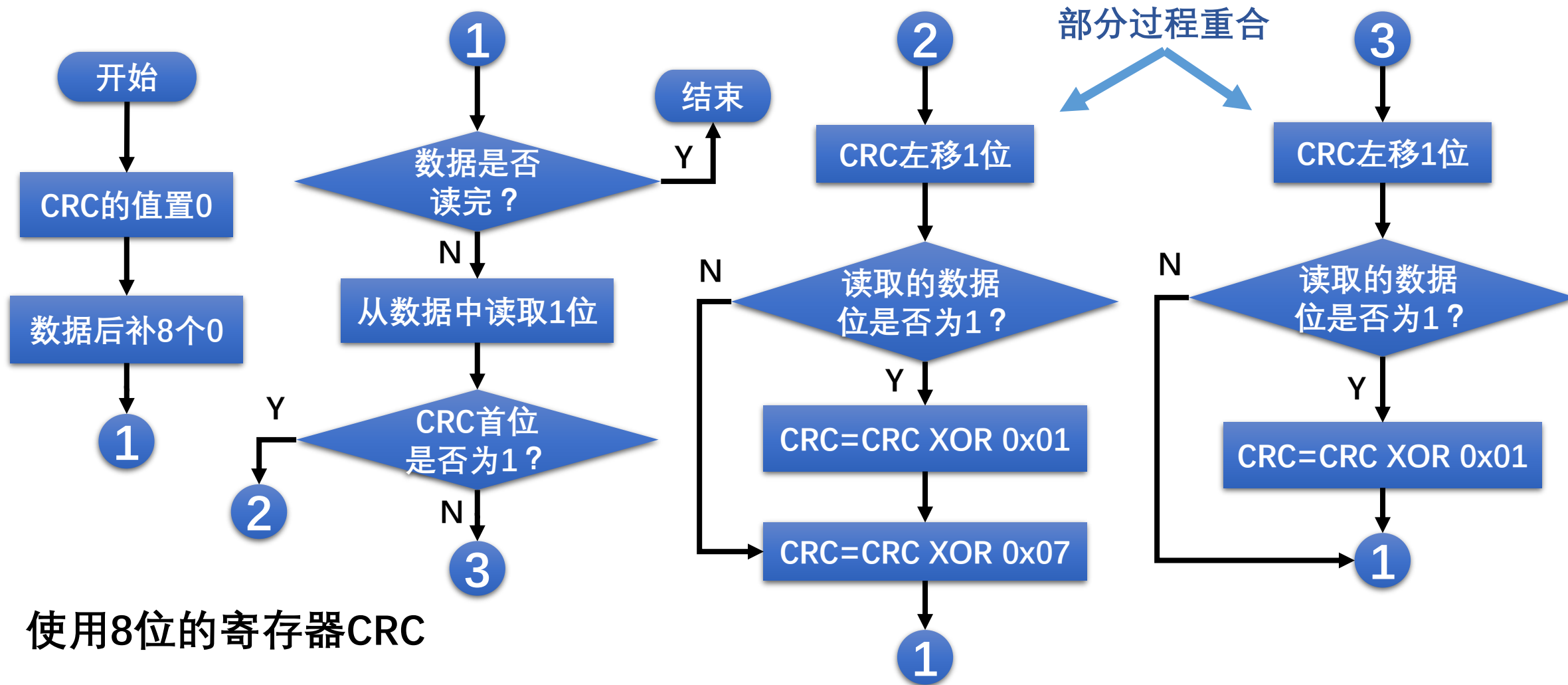
$$\begin{array}{r} \text{10001001} \\ \text{100000111} \sqrt{\text{1000101010111111}} \\ \underline{\text{100000111}} \\ \text{100100111} \\ \underline{\text{100000111}} \\ \text{100000111} \\ \underline{\text{100000111}} \\ \text{0} \end{array}$$

目的节点的CRC校验

10001010后面
添加余数10111111

余数0，说明CRC校验正确

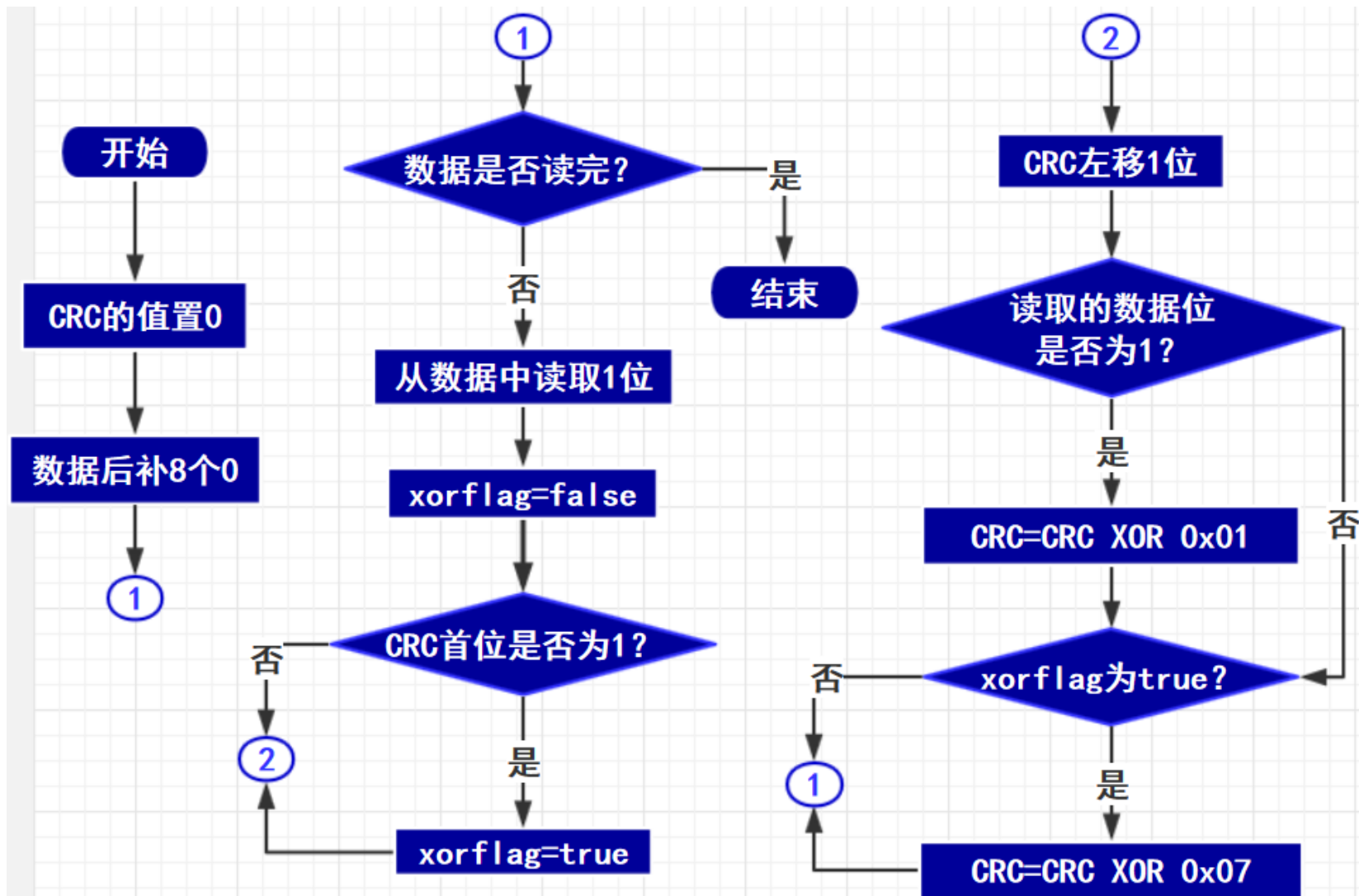
3. 例题分析—关键问题



3. 例题分析—关键问题

- CRC校验的过程

```
//CRC是一个8位的寄存器  
将CRC的值置为0  
在原始数据后面添加8个0  
while (数据未处理完)  
{  
    从数据中读取1位  
    xorflag = false  
    if(CRC首位为1)  
        xorflag = true  
    CRC左移1位  
    if(读取的数据位是1)  
        CRC=CRC XOR 0x01  
    if(xorflag)  
        CRC=CRC XOR 0x07  
}
```



3. 例题分析—关键问题

- CRC校验的范围

- 目的地址
- 源地址
- 长度
- 数据字段

参与余数的计算

pos_target_addr = 目的地址位置

写入文件(校验范围内所有内容)

pos_crc = 帧校验字段位置

total = 带读取数据长度

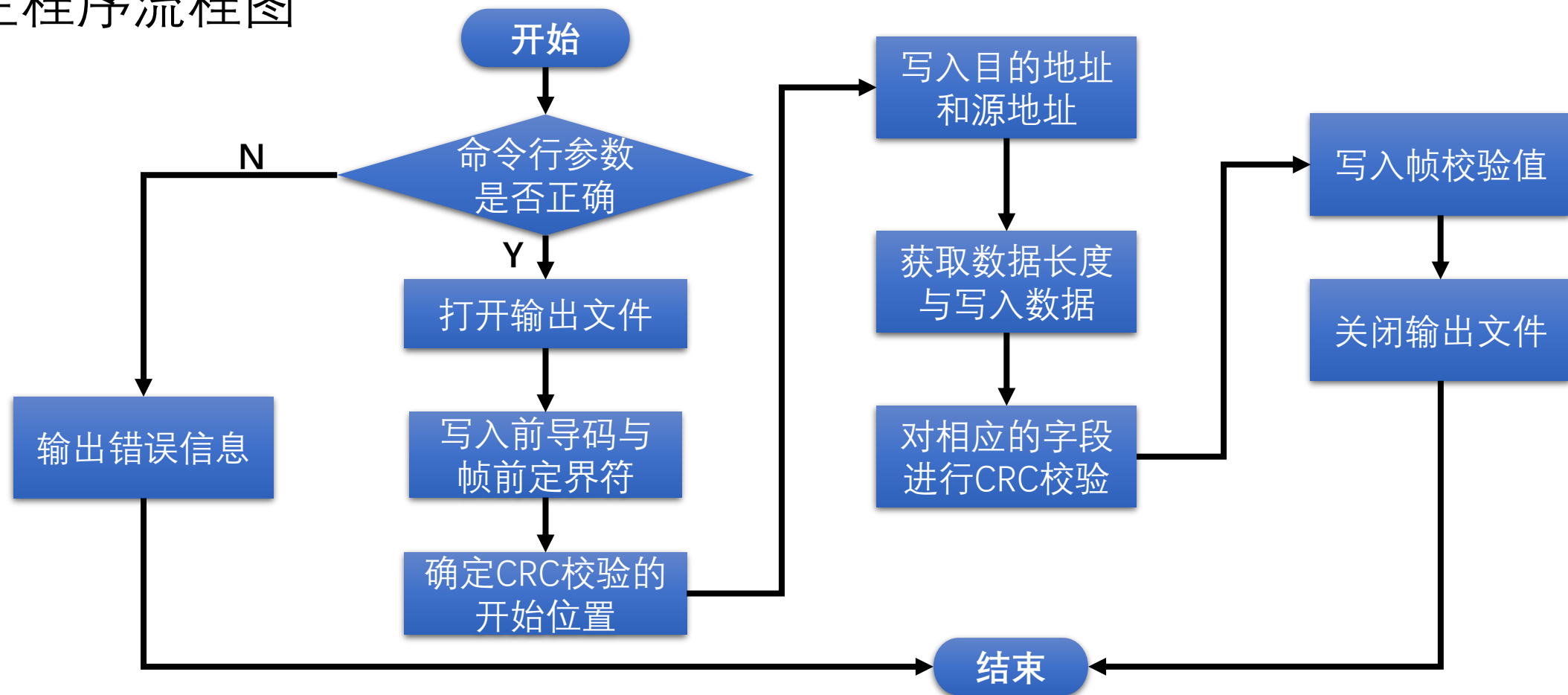
while(total--)

CRC校验过程 (以字节为单位)

将CRC值写入帧校验字段

3. 例题分析—关键问题

- 主程序流程图



3. 例题分析—演示



The image shows a screenshot of the Microsoft Visual Studio 调试控制台 (Debug Console) window. The window has a title bar with the text "Microsoft Visual Studio 调试控制台" and standard window controls (minimize, maximize, close). The console output is as follows:

```
长度字段: c(12)  
数据字段: Hello world!  
  
帧校验字段: 4f(79)  
  
帧封装与CRC校验完成
```

本章小结

- 设计目的
 - 通过编程实现CRC校验过程，了解网络协议中校验和的计算过程与作用
- 相关知识
 - CRC校验概论、过程、计算实例
- 例题分析
 - 编程实现帧的封装、以及CRC校验范围、计算余数