关于循环冗余校验算法（CRC）总结

一、循环冗余校验介绍

1.1 校验算法原理

在**有效信息位**后面通过增加**冗余校验位**使其（合成的数据包）具有检错能力。

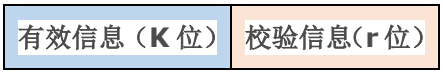
1.2 校验算法介绍

循环冗余校验（Cyclic Redundancy Check，CRC）具有检错纠错能力，但在实际中我们仅使用其检错能力。如接受端发现数据传输错误，发送端重新发送即可。

二、循环冗余校验原理

2.1 基本概念

1）增加冗余码（校验位）



$$N = k + r \leq 2^r - 1$$

校验位设置为 r 位，可表示2^r种状态，其中 1 种状态表示没有错误，$2^r - 1$种状态表示错误。

A_1~A_7	余数	出错位
1100010	000	无
1100011	001	7
1100000	010	6
1100110	100	5
1101010	011	4
1110010	110	3
1000010	111	2
0100010	101	1

2）生成多项式 G(x)

收发双方约定一个多项式 G(x)对应的二进制数位（r+1），发送方利用 G(x)对数据包（增加零占位冗余码的编码数据）做模 2 除运算，生成校验码。接收方利用 G(x)对收到的数据包（使用校验码替换零占位冗余码）做模 2 除运算检测差错及错误定位。

3）G(x)应满足的条件

- 最高位和最低位必须为 1；
- 当被传送信息（含 CRC 校验码）任何一位发生错误时，被生成多项式做模 2 除后余数不为 0；
- 传送信息不同位发生错误时，模 2 除运算后余数不同；
- 对不为 0 余数继续进行模 2 除运算后应使余数循环。

2.2 模 2 运算规则

1）加/减运算

异或运算，加不进位，运算法则：值相同结果为 0；值不同结果为 1。

0⊕0 = 0，1⊕1= 0，1⊕0 = 1，0⊕1 = 1

模 2 加减法运算类似于异或运算，加不进位，减不借位。

0±0 = 0，1±1= 0，1±0 = 1，0±1 = 1

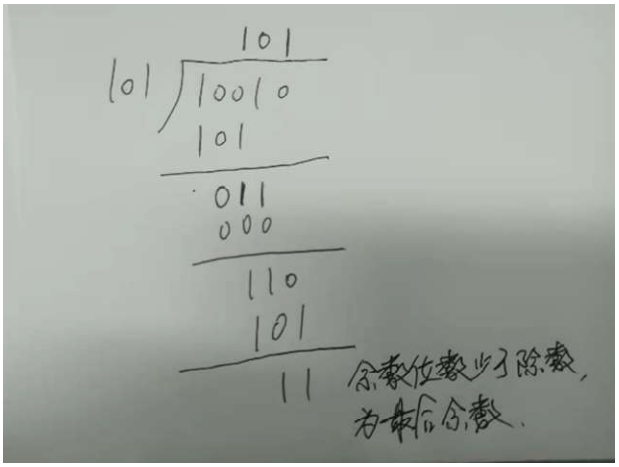
2）除法运算

“模 2 除法”与“算术除法”类似，区别是它既不借位，也不比较除数和被除数的相同位数值的大小，只要以相同位数进行模 2 减法运算即可。

3）上商原则

模 2 除法运算时商值确定原则：

- 部分余数首位为 1 时，商为 1，减除数，继续进行模 2 除法；
- 部分余数首位为 0 时，商为 0，减 0，继续进行模 2 除法，
- 当部分余数的位数小于除数的位数时，该余数即为最后余数。



三、循环冗余校验的检错与纠错

因为在发送端发送数据帧之前已添加 CRC 校验码，即做了“去余”处理（也就已经能整除了）。所以接收方利用 G(x)对收到的数据帧做模 2 除运算，结果应该是没有余数的。如果有余数，则表明该帧在传输过程中出现了差错。

3.1 检错

- 相同生成多项式 CRC 编码，不同 bit 位出错时对应的余数（校验码）不尽相同；
- 不同生成多项式 CRC 编码，不同 bit 位出错时对应的余数（校验码）不尽相同；

A_1~A_7	余数	出错位
1100010	000	无
1100011	001	7
1100000	010	6
1100110	100	5
1101010	011	4
1110010	110	3
1000010	111	2
0100010	101	1

G(X) = 1011

A_1~A_7	余数	出错位
1100101	000	无
1100100	001	7
1100111	010	6
1100001	100	5
1101101	101	4
1110101	111	3
1000101	011	2
0100101	110	1

G(X) = 1101

3.2 纠错

1bit 出错情况下余数的循环特性，bit（x+1）的余数补零求模 2 除法得到的余数是 bitx 出错情况下的余数。需要注意的是，该数据帧是接收方接收到的数据即数据中已包含 CRC 校验位。

利用出错情况下余数的循环特性进行纠错。以如下 A1~A7 数据为例。若余数不为 0，一边对余数补 0 继续做模 2 除，同时让被检测的编码（A1~A7）循环左移，当余数为 101 时，出错位也移到 A1 位置。通过异或运算纠正编码后继续循环左移和执行余数模 2 除法，直到修改后的出错位回原位。注意，其中提到的余数补 0 做模 2 除和编码循环左移时相互独立操作的。

A_1~A_7	余数	出错位
1100010	000	无
1100011	001	7
1100000	010	6
1100110	100	5
1101010	011	4
1110010	110	3
1000010	111	2
0100010	101	1

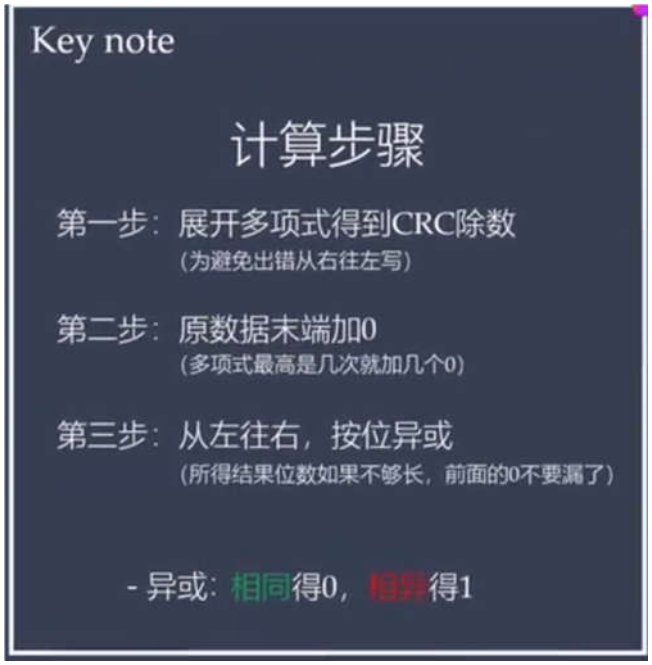
四、循环冗余校验码计算步骤

- 根据待校验信息的长度 k，按照 k+r≤2^r-1 确定校验位 r 的位数；
- 根据 r 和生成多项式的选择原则，选择位数为 r+1 的生成多项式；
- 待校验信息信息逻辑左移 r 位，得到增加零占位冗余码的编码数据；
- 对编码数据按模 2 运算法则除 G(x)，求 CRC 编码中的 r 位校验信息；

- 用得到的余数即 CRC 校验码替换编码数据中 r 位零占位冗余码；

余数的位数一定要是比生成多项式位数少一位。如果余数最前面位是 0，甚至是全为 0（刚好整除时）也都不能省略。

- 最终得到含有 CRC 校验信息的编码数据。



五、CRC 校验码计算示例

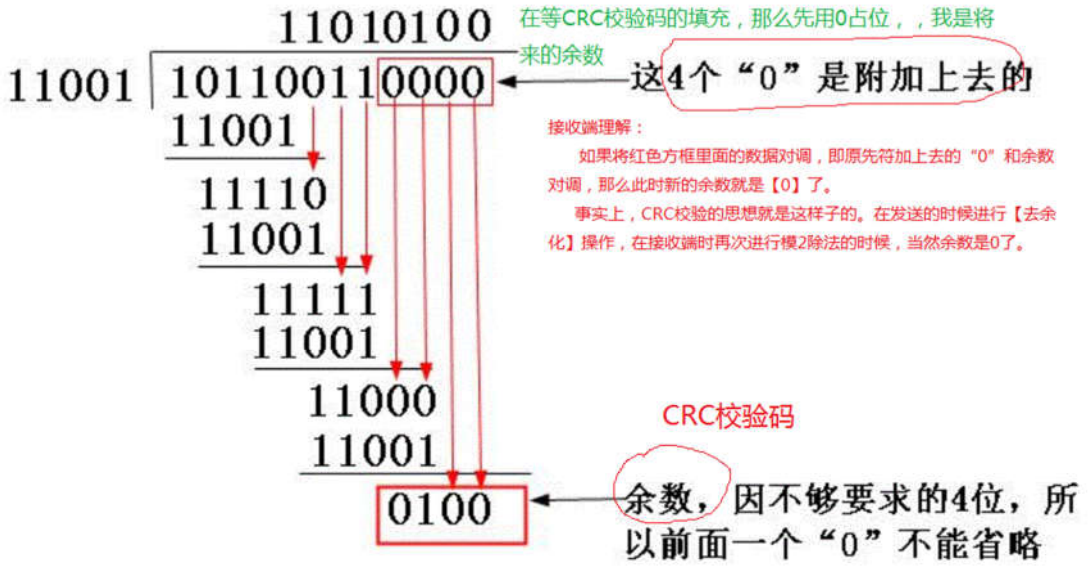
设 CRC 生成多项式为 G(x)=x^4+x^3+1，要求出二进制序列 10110011 的 CRC 校验码，计算过程如下：

（1）生成多项式转换成二进制数

由 G(x)=x^4+x^3+1 知，生成多项式是 5 位，然后根据多项式各项系数，就可得到其二进制比特串为 11001。

（2）计算 CRC 校验码

因为生成多项式的位数为 5，根据前面的介绍，得知 CRC 校验码的位数为 4（**校验码的位数比生成多项式的位数少 1**）。因为原数据帧 10110011，在它后面再加 4 个 0，得到 101100110000，然后把这个数以“模 2 除法”方式除以生成多项式，得到的余数，即 CRC 校验码为 0100，如下图所示。



（3）发送端发送含 CRC 校验码的数据帧

用计算得到的 CRC 校验码 0100 替换原始帧 101100110000 后面的四个“0”，得到新帧 101100110100，然后将其发送。

（4）接收端接收并校验数据帧

接收端使用相同的生成多项式 G(x)=xˆ4+xˆ3+1 对会数据帧进行“模 2 除法”，验证余数是否为 0。如果为 0，则证明该帧数据在传输过程中没有出现差错，否则出现了差错。（注意：此时运算的商和发送前计算 CRC 校验码时的商是相同的哦）

六、代码实现

略

七、常用 CRC 码标准

任意一个由二进制位串组成的代码都可以和一个系数仅为‘0’和‘1’取值的多项式一一对应。例如：代码 1010111 对应的多项式为 x6+x4+x2+x+1，而多项式为 x5+x3+x2+x+1 对应的代码 101111。

名称	生成多项式	简记式*	应用举例
CRC-4	x4+x+1	3	ITU G.704
CRC-8	x8+x5+x4+1	31	DS18B20
CRC-12	x12+x11+x3+x2+x+1	80F	
CRC-16	x16+x15+x2+1	8005	IBM SDLC
CRC-ITU** CRC-CCITT	x16+x12+x5+1	1021	ISO HDLC, ITU X.25, V.34/V.41/V.42, PPP-FCS,ZigBee
CRC-32	x32+x26+x23+...+x2+x+1	04C11DB7	ZIP, RAR, IEEE 802 LAN/FDDI, IEEE 1394, PPP-FCS
CRC-32c	x32+x28+x27+...+x8+x6+1	1EDC6F41	SCTP

* 生成多项式的最高幂次项系数是固定的1，故在简记式中，将最高的1统一去掉了，如04C11DB7实际上是104C11DB7。 ** 前称CRC-CCITT。ITU的前身是CCITT。

生成多项式的最高位固定是 1，故在简记中忽略最高位 1，比如:0x1021 实际是 0x11021.简记式本质式生成多项式的二进制位串对应的十六进制表示，因此我们可以根据简记式很轻松得出生成多项式。比如：1021。

1021 -→0x11021-→~~000~~1 0001 0000 0010 0001-→$G(x) = x^{16} + x^{12} + x^5 + 1$

八、注意事项

1) CRC 校验码的二进制位数比对应的 CRC 多项式的二进制位数少 1 位（比如：CRC-16 生成多项式 g（x）= x16 + x15 + x2 +1 对应的二进制是 17 位），所以直接通过 CRC 的标记，CRC16 那么校验码即为 16 位。

九、参考文献

9.1 网络资源

1）「CRC 校验」手算与直观演示

https://haokan.baidu.com/v?pd=wisenatural&vid=8173702608073301049

9.2 书籍