

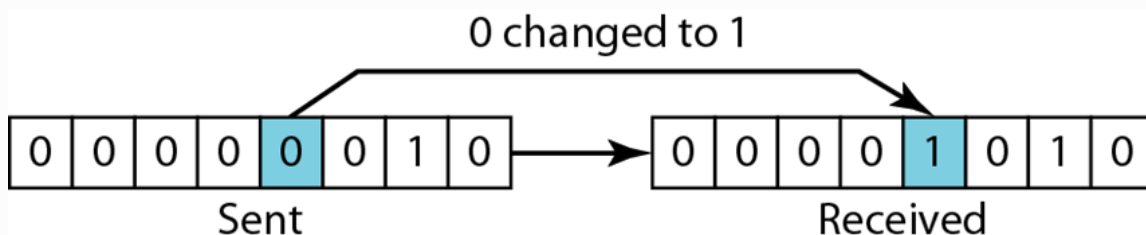
<!--

- @Title: 检错与纠错
 - @Description:
 - @Author: SoulCompiler/prinscarce
 - @Email: prinscarce@outlook.com
 - @Blog:
 - @LastEditors: prinscarce
- >

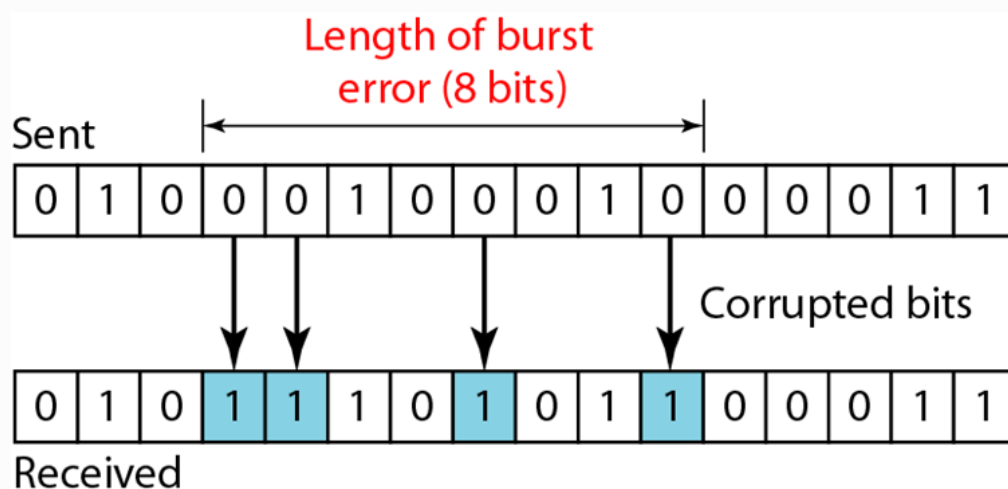
第 10 章 检错与纠错

一. 差错的类型

- 单个位差错
 - 在单比特差错中，数据单位中仅有一比特发生变化。



- 突发性差错
 - 一个突发差错意味着数据单元中两位或多位发生变化。

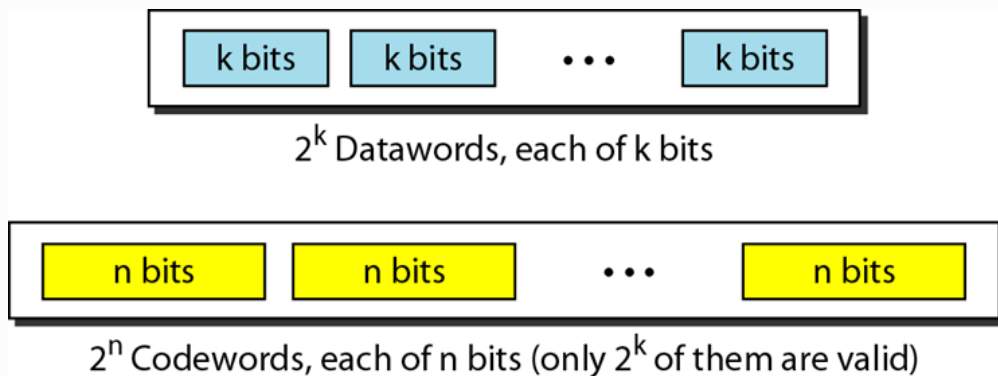


二. 块编码的概念

- 概念：

- 把报文划分成块，每个块有 k 位，称为**数据字**，并增加 r 个冗余位使其长度变为 $n = k + r$ ，形成 n 位的块称为**码字**。

- 块编码处理是一一对一，相同的数据字总是编码成相同的码字。



三. 纠错的概念与方法

汉明距离、最小汉明距离

汉明距离

- 概念：

- 两个（相同长度）字的汉明距离是对应位不同的数量。

- 最小汉明距离是一组字中所有可能对的最小汉明距离。

- 方法：

- 以 $d(x, y)$ 表示两个字之间的汉明距离。

- 对两个字进行异或操作并计算1的个数，就可以很容易地得出汉明距离。

- 汉明距离大于零。

最小汉明距离

- 概念：

- 在一组字中，最小汉明距离是有可能对中的最小汉明距离。

- 为了保证**检测**出最多 s 个错误，块编码中最小汉明距离一定是 $d_{\min} = s + 1$ 。

- 为了保证最多能**纠正** t 个差错，块码中最小汉明距离是 $d_{\min} = 2t + 1$ 。

- 方法：

- 以 d_{\min} 定义编码方案中的最小汉明距离。
- 导出所有字的汉明距离并选择最小值。

e.g. $d_{\min}=3$

| | | |
|-----------------------|-----------------------|-----------------------|
| $d(00000, 01011) = 3$ | $d(00000, 10101) = 3$ | $d(00000, 11110) = 4$ |
| $d(01011, 10101) = 4$ | $d(01011, 11110) = 3$ | $d(10101, 11110) = 3$ |

四. 线性块编码

线性块编码的距离、简单奇偶校验编码、两维奇偶校验编码、汉明码

线性块编码

- 非正式定义：
 - 线性块编码是一种由任何两个有效码字的异或（XOR）产生的另一个有效码字。
- 线性块编码的最小距离
 - 具有最小1的个数的非0有效码字中1的个数。（非零码字的1的个数）（

简单奇偶校验编码

- 概念：
 - 简单的奇偶校验码是 $n = k + 1$ ，且 $d_{\min} = 2$ 的单比特检错码。
 - n 位码字， k 位数据
- 简单奇偶校验编码能检测出奇数个差错。

两维奇偶校验编码

- 概念：
 - 数据字以表格形式（行和列）组织。
 - 对于每一行和每一列，计算出一个奇偶校验位，然后将整个表发送给接收方，接收方将分别得出每一行和每一列的校正子。
- 两维奇偶校验能检测出表中任何位置发生的最多三个差错（箭头指向生成的非零校正子位置）。4位的差错无法检测到。

汉明编码

。概念：

- 。汉明距离 m 与码字长 n 和数据字长 k 的关系为 $n = 2^m - 1$, $k = n - m$, 校验位个数 $r = m$
- 。 $d_{\min} = 3$
- 。 $n = 2^m - 1$
- 。 详见计组

5. 循环冗余编码 CRC

计算、检错能力分析（检测单个位差错、两个独立的位差错、奇数个差错、突发性差错）

循环冗余校验的计算

2. CRC 码的编码

设 n 位数据为 $D_{n-1} \sim D_0$, $k+1$ 位生成码为 $G_k \sim G_0$, 则构成的 CRC 码码长为 $n = n + k$, 其中包含 k 位校验位。

(1) 二进制数据用多项式表示。将 n 位数据用多项式 $M(x)$ 表示为

$$M(x) = D_{n-1}x^{n-1} + D_{n-2}x^{n-2} + \dots + D_1x^1 + D_0x^0 \quad (2.21)$$

式中, D_i 为 1 或 0。 $k+1$ 位生成码用多项式 $G(x)$ 表示为

$$G(x) = G_kx^k + G_{k-1}x^{k-1} + \dots + G_1x^1 + G_0x^0 \quad (2.22)$$

(2) 数据做左移 k 位操作。数据左移 k 位相当于多项式 $M(x)$ 做乘以 x^k 的操作, 即得 $M(x) \cdot x^k$, 其 $n+k$ 位的二进制编码如图 2.16 所示。

| | | | | | | | |
|-----------|-----------|---------|-------|-------|---------|---|---|
| $n+k-1$ | $n+k-2$ | | k | $k-1$ | | 1 | 0 |
| D_{n-1} | D_{n-2} | \dots | D_0 | 0 | \dots | 0 | 0 |

图 2.16 $M(x) \cdot x^k$ 的 $n+k$ 位二进制编码

其中, 低 k 位为 0, 将该编码作为待编信息码。

(3) 求余数。用 $M(x) \cdot x^k$ 对生成多项式 $G(x)$ 作模 2 除法, 求余数多项式 $R(x)$, 即

$$\frac{M(x) \cdot x^k}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \pmod{2} \quad (2.23)$$

其中, $Q(x)$ 为商的多项式, 余数 $R(x)$ 的二进制编码为 k 位。在此强调, 除法过程必须按照模 2 运算规则计算。

(4) 构成 CRC 码。将余数作为校验位, 用 (2) 中获得的待编信息码多项式 $M(x) \cdot x^k$ 与余数 $R(x)$ 作模 2 加, 构成 CRC 码多项式 $C(x)$, 即

$$C(x) = M(x) \cdot x^k + R(x) = Q(x) \cdot G(x) \pmod{2} \quad (2.24)$$

CRC 码的编码格式如图 2.17 所示。

| | | | | | | | |
|-----------|-----------|---------|-------|-----------|---------|-------|-------|
| $n+k-1$ | $n+k-2$ | | k | $k-1$ | | 1 | 0 |
| D_{n-1} | D_{n-2} | \dots | D_0 | R_{k-1} | \dots | R_0 | R_0 |

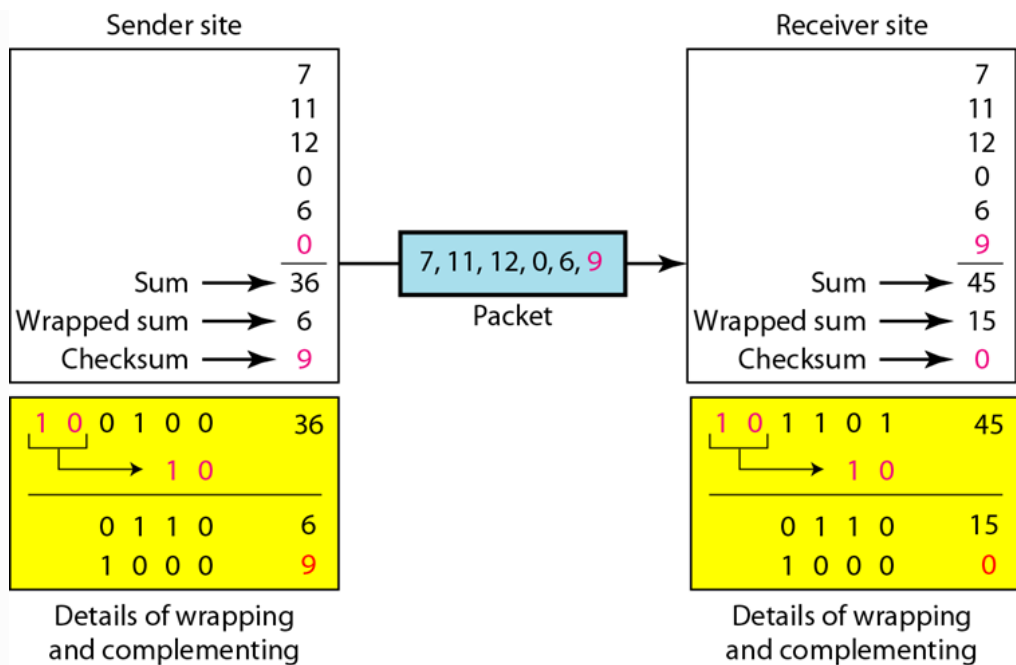
图 2.17 CRC 码的编码格式

CRC 的检错能力分析

- 单个位差错
 - 如果生成多项式至少有两项并且 x^0 的系数为1，那么所有单个位差错都可以被捕捉到。
- 两个独立的单个位差错
 - 如果生成多项式不能整除 $x^t + 1$ (t在0和n-1之间)，那么所有独立的双差错都能被检测到。
 - 若能整除 $x^t + 1$ ，则两个相隔t个位置的单个位差错不能检测到
- 奇数个差错
 - 包含因子 $x+1$ 的生成多项式能检测到所有奇数个差错。
- 突发性差错
 - 所有 $L \leq r$ 的突发性差错都会被检测到。
 - 所有 $L = r+1$ 的突发性差错有 $1 - (1/2)^{r-1}$ 的概率会被检测到。
 - 所有 $L > r+1$ 的突发性差错有 $1 - (1/2)^r$ 的概率会被检测到。
- 生成多项式检错能力总结：
 1. 至少有两项
 2. x^0 的系数应该是1
 3. 应该不能整除 $x^t + 1$ ($2 < t < n-1$)
 4. 应该有因子 $x + 1$ 。

六. 校验和及其计算方法

- 概念：
 - 假设要发送的目的数据是5个4位数。除了发送这些数，还发送它们的和。例如，如果数字组是 (7, 11, 12, 0, 6), 那么我们发送 (7, 11, 12, 0, 6, 36), 这里 36 是原来的数字的和。接收方将这5个数字求和并比较。如果相等，接收方认为没有差错；否则认为有差错。
 - 可发送校验和的补码
 - 约束和(wrapped sum): 多出的位数与和相加
 - 校验和: 求约束和的补码。可用 $2^n - 1$ 减去这个数
- p.s. 图左右下角都有错



• 计算方法：

。 发送方：

1. 报文被划分为16位字。
2. 校验和字的值设为0。
3. 所有字包括校验和使用反码运算相加。
4. 对这个和求反变成校验和。
5. 校验和随数据一起发送。

。 接收方：

1. 报文（包括校验和）被划分成16位字。
2. 用反码加法将所有字相加。
3. 对该和求反生成新的校验和。
4. 如果校验和的值是0,接收报文；否则就丢弃报文。

| 1 | 0 | 1 | 3 | Carries |
|---|---|---|---|--------------------|
| 4 | 6 | 6 | F | (Fo) |
| 7 | 2 | 6 | F | (ro) |
| 7 | 5 | 7 | A | (uz) |
| 6 | 1 | 6 | E | (an) |
| 0 | 0 | 0 | 0 | Checksum (initial) |
| 8 | F | C | 6 | Sum (partial) |
| 8 | F | C | 7 | Sum |
| 7 | 0 | 3 | 8 | Checksum (to send) |

a. Checksum at the sender site

| 1 | 0 | 1 | 3 | Carries |
|---|---|---|---|---------------------|
| 4 | 6 | 6 | F | (Fo) |
| 7 | 2 | 6 | F | (ro) |
| 7 | 5 | 7 | A | (uz) |
| 6 | 1 | 6 | E | (an) |
| 7 | 0 | 3 | 8 | Checksum (received) |
| F | F | F | E | Sum (partial) |
| 8 | F | C | 7 | Sum |
| 0 | 0 | 0 | 0 | Checksum (new) |

a. Checksum at the receiver site