

明德扬科技教育有限公司

CRC 原理和串行实现

官 网: www.mdy-edu.com

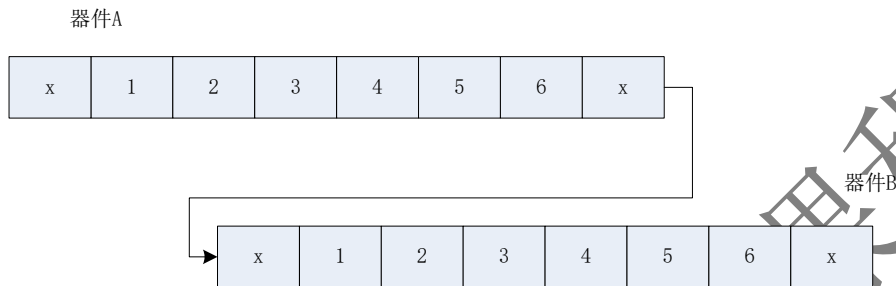
淘 宝: mdy-edu.taobao.com

QQ 群: 97925396

QQ 咨询: 158063679

CRC 原理和串行实现

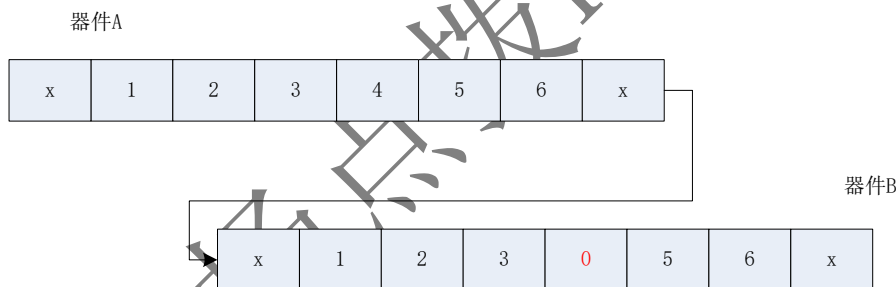
场景 1：器件 A 送数据给器件 B



A 发送数据：1 2 3 4 5 6

B 接收数据：1 2 3 4 5 6

场景 2：器件 B 接收数据有错误，B 如何知道数据有错误呢？

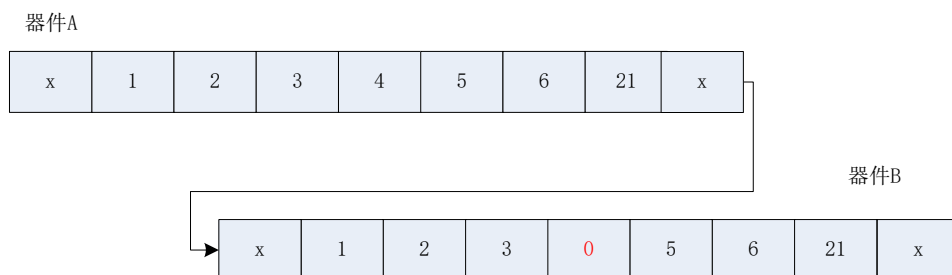


A 发送数据：1 2 3 4 5 6

B 接收数据：1 2 3 0 5 6。但 B 不知道数据有错。

场景 3：器件 A 和 B 双方约定一个算法，B 根据算法计算得知数据是否有错。

例如：AB 双方算法为数据和。

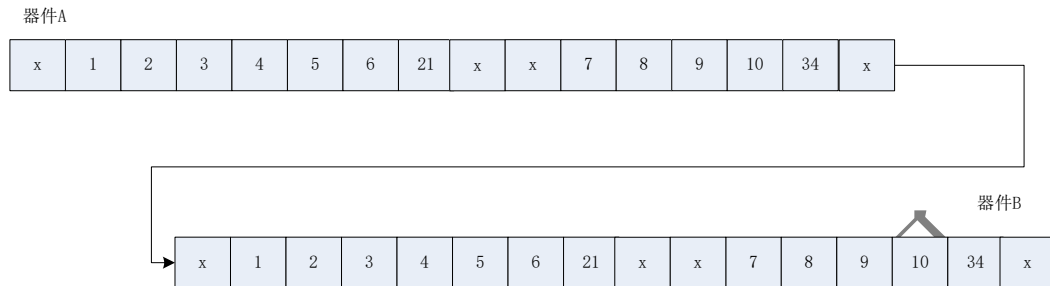


A 发送数据：1 2 3 4 5 6 和 21。21 获取方法： $1+2+3+4+5+6=21$ 。

B 接收数据：1 2 3 0 5 6 和 21。B 自己计算： $1+2+3+0+5+6=17$ ，与接收到的 21 不同。

B 判断出错。

场景 4：发送多个包文数据



A 发送数据：1 2 3 4 5 6 和 21；7 8 9 10 和 34。

B 接收数据：1 2 3 4 5 6 和 21。B 计算 $1+2+3+4+5+6=21$ ，与接收的 21 相同，因此判断接收正确。

B 接收数据：7 8 9 10 和 34。B 计算 $7+8+9+10=34$ 与接收的 34 相同，因此判断正确。

注意：算法以包文为基础，各个包文要独立运算。因此计算完一个包文后，CRC 运算要清零。

场景 5：CRC 算法运算。

简单求和的算法太简单，检测能力弱。为了尽可能多地发现错误，所以采用 CRC 算法。（提高检测能力的原因，请自行百度）。

CRC 算法是模 2 运算、也是做异或运算，其包括如下内容：

1. 生成多项式

生成多项式类似于 $g(x)=x^4+x^3+1$ 形式，其代表的值为 11001。 $g(x)=X^3+X+1$ ，其值为 1011。AB 双方都要约定一个生成多项式。

2. 检验位宽。

即有多少位是检验位。这一般是多项式最高系数。如 $g(x)=x^4+x^3+1$ ，其检验位宽是 4 位，称之为 CRC4。 $g(x)=X^3+X+1$ 的位宽是 3，称之为 CRC3。

多项式不是随便定义的，每个领域都有自己的标准，如果以太网是使用 CRC32，多项式有些为 $X^{32}+X^2+X+1$ ，广播领域也不相同。

3. 编码过程

假充双方约定的生成多项式为 11001，要发送的数据为 1011001，则生成检验码的过程如下：

a. 数据位左移(检验位宽)

左移后数据为：10110010000

b. 执行异或运算。

10110010000

```

^11001
-----
01111010000
1111010000
^11001
-----
0011110000
11110000
^11001
-----
00111000
111000
^11001
-----
001010

```

则最后的校验码是 1010。器件 A 发送给器件 B 的数据则为数据+校验码。即 10110011010。

4. 译码过程

假设器件 b 接收到正确的数据，即 10110011010，其也要做 CRC 运算。

a. 数据位左移(检验位宽)

```
101100110100000
```

b. 执行异或运算

```
101100110100000
```

```
^11001
```

```
-----
```

```
011110110100000
```

```
11110110100000
```

```
^11001
```

```
-----
```

```
0011110100000
```

```
111110100000
```

```
^11001
```

```
-----
```

```
001100100000
```

```
1100100000
```

```
^11001
```

```
-----
```

```
0000000000
```

最后器件 b 算出来的检验位是 0，则说明接收正确。如果不为 0，则说明接收错误。

总结：器件 a 和器件 b 的运算方法是相同的，不同的是器件 b 会将检验码当作数据一起运算，并检查最后结果是否为 0。所以我们只需要设计一个模块，可用于编码和解码。

在举例，双方约定的生成多项式为 $g(x)=x^8+x^2+x+1$ ，即 9'b1_0000_0111。器件 A 要发

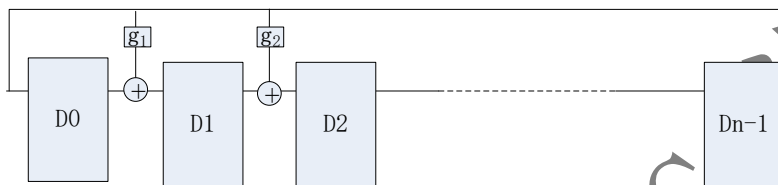
送数据 8'h1, 8'h2, 8'h3 给器件 B。则数据是 24'b0000_0001_0000_0010_0000_0011, 其运算是用 32'b0000_0001_0000_0010_0000_0011_0000_0000 对 9'b1_0000_0111 求模, 结果为 8'b0100_1000=8'h48。

器件 A 总共发送的数据是 8'h1, 8'h2, 8'h3, 8'h48。器件接收到数据 8'h1, 8'h2, 8'h3, 8'h48, 将这四个数据一起运算, 运算结果如果为 0, 则说明接收正确。

场景 6: 串行 CRC 的实现

场景 5 用 FPGA 实现起来比较麻烦, 需要经过多次移位和异或操作。

串行 CRC 常用 LFSR (线性反馈寄存器) 来实现, 非常简单。串行是指每次只输入 1bit 的数据, 直到把所有数据输入完后, 就可以得到最终的 CRC 检验码。下面是实现电路结构图。



从以上电路中, 可以有如下规律:

1. **i 是 1 比特输入的数据; n 个 D 寄存器的值就是我们要的检验码。**当输入 1 比特数据, 所有 D 寄存器就变化一次, 即得到输入 1 位后的检验码。再输入 1 比特数据, D 寄存器又更新, 得到输入 2 位数后的检验码。如果全部数据输入完毕, 就得到最终的检验码。
2. 此处一共有 n 个 D 寄存器, CRC32 就有 32 个 D 寄存器。自然 CRC8 自然是有 8 个寄存器。
3. 图中 $g_1 \sim g_{n-1}$ 对应多项式的系数, 只能是 0 或 1。如果为 0 线就断开, 为 1 线是连接的。例如 x^8+x^2+x+1 , 那么 g_2 、 g_1 是连接的, 其他都是断开的。
4. 如果 g_x 是连接的, 那么异或门有效, 如果是断开的, 则相当于没有异或门。如多项式是 x^3+x^2+x+1 , 则 D1、D2 前面有异或门, 并且 D31 后面有一个异或门。同样道理, 假如多项式是 $x^8+x^7+x^3+1$, 那么就是 D7/D3 前面都有一个异或门, D7 后面也有一个异或门。

下图是 CRC4, 多项式为 x^4+x+1 的电路图。

