CRC 码串并结合算法的研究与实现

王月琴,杨恒新
(南京邮电大学 电子科学与工程学院,江苏 南京 210003)

摘 要:CRC 码以其算法简单、检错能力强、抗干扰性能优异等特点,广泛应用于各种通信协议中。这里在分析 CRC 串行算法和并行算法的基础上,提出串并结合的算法。CRC 循环冗余串并结合算法相比 CRC 串行编码,大大提高了计算速率;相比 CRC 并行编码,克服了通信中数据位非 8 的整数倍的问题。以 CRC-ITU 生成多项式为例,通过仿真,验证了该算法的正确性和可行性。

关键词:循环冗余校验;串行算法;并行算法;VHDL
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2014)06-0103-04
doi:10.3969/j.issn.1673-629X.2014.06.026

Research and Implementation of CRC Code Serial Parallel Combining Algorithm

WANG Yue-qin, YANG Heng-xin
(College of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Owing to its simple algorithm, extraordinary error code checking and anti-jamming abilities, Cyclic Redundancy Check (CRC) finds the widest applications of all communication protocols. Based on analyzing the serial algorithm and parallel algorithm, propose the serial parallel combining algorithm. Compared with the serial algorithm, the new algorithm has greatly improved its computing rate, and with the parallel algorithm it has overcome the non-integer multiple of 8 issue. And simulation results show that the algorism is valid and feasible.

Key words: cyclic redundancy check; serial algorithm; parallel algorithm; VHDL

0 引 言

数据在传递过程中,可能因为各种各样原因而使传输或接收的数据发生错误[1]。为判断接收端数据的正确性,使用校验码是一种常用的方法[2]。循环冗余校验码(Cyclic Redundancy Check, CRC)就是其中的一种,它以检错概率高并且易于用硬件实现的优点,广泛应用在移动通信、计算机通信、USB 接口、测控等领域[3]。

CRC 计算有多种实现方法,可采用软件计算方法,也可采用硬件计算方法;可按位串行进行计算(可以适用于任意长度的数据,但计算速率低),也可多位并行进行计算(计算速率高,适用于长度有固定规律的数据,如长度为 8 的整数倍)[4-6]。文中通过对 CRC 原理和串行、并行算法的分析,提出串、并结合的算法,可适用于任意长度的数据,同时提高了算法的计算速率。

1 CRC 校验原理

CRC 的基本原理就是在一个 n 位二进制数据序列之后附加一个 r 位二进制检验码序列,从而构成一个总长为 $p=n+r$ 位的二进制序列。这里附加在数据序列之后的 CRC 码与数据序列的内容之间存在某种特定的关系。

如果在数据传输过程中,由于噪声或传输特性不理想而使数据序列中的某一位或某些位发生错误,这种特定关系就会被破坏。可见在数据的接收端通过检

收稿日期:2013-08-10 **修回日期**:2013-11-19 **网络出版时间**:2014-02-24
基金项目:江苏省自然科学基金青年基金(SJ212013)
作者简介:王月琴(1988-),女,硕士,研究方向为超高频 RFID 阅读器及标签的基带设计;杨恒新,副教授,研究方向为 UHF RFID 系统的设计理论、设计方法及其应用。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140224.0902.021.html

查这种特定关系,可以很容易地实现对数据传输正确性的检验[7]。

在 CRC 中,校验码 R 是通过对数据序列 M 进行二进制除法取余式运算得到的,M 被一个称为生成多项式的(r+1)位的二进制序列 G 除,具体的多项式除法形式如下所示[8]:

$$\frac{x^r M(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \tag{1}$$

其中,$x^r M(x)$ 表示将数据序列 M 左移 r 位,即在 M 的末尾增加 r 个 0;$Q(x)$ 代表这一除法所得的商;$R(x)$ 就是所需的余式。

目前,生成多项式具有以下一些通用标准,其中 CRC-12、CRC-16、CRC-ITU 和 CRC-32 是国际标准。

CRC-4	$x^4 + x + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x + 1$
CRC-16	$x^{16} + x^{12} + x^2 + 1$
CRC-ITU	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + \cdots + x^2 + x + 1$
CRC-32c	$x^{32} + x^{28} + x^{27} + \cdots + x^8 + x^6 + 1$

2 CRC 串并结合算法

2.1 串行算法原理与实现

设数据序列 M 有 n 位,即 $M = (m_{n-1}, m_{n-2}, \cdots, m_1, m_0)$,按位权展开,可表示为:

$$M = m_{n-1} \times 2^{n-1} + m_{n-2} \times 2^{n-2} + \cdots + m_1 \times 2^1 + m_0 \tag{1}$$

除数 G 的最高次幂为 k,$G = g_k \times 2^k + g_{k-1} \times 2^{k-1} + \cdots + g_1 \times 2^1 + g_0$。

根据 EPC Global UHF C1G2 协议,生成多项式采用 CRC-ITU[8]:$x^{16} + x^{12} + x^5 + 1$。

式(1)可重写为式(2),其中 R 为 16 位 CRC。

$$\frac{M \times 2^{16}}{G} = Q + \frac{R}{G} \tag{2}$$

实际计算时,CRC 寄存器并不是先并行置入 M 的前 16 位(不足 16 位时,后面补 0,M 后有 16 个 0),再进行 CRC 计算,而是赋予 CRC 寄存器一个初始值 $R' = r_{15}^0 \times 2^{15} + r_{14}^0 \times 2^{14} + \cdots + r_1^0 \times 2^1 + r_0^0$(EPC Global UHF C1G2 协议规定 16 位寄存器的初始值为 0xFFFF),该初始值可假设为信息码 $M' = (m'_{t-1}, m'_{t-2}, \cdots, m'_1, m'_0)$ 经公式(2)计算得到,即 $\frac{M' \times 2^{16}}{G} = Q' + \frac{R'}{G}$。

求 M 的 CRC 码转化为:已知 M' 的 CRC 码,求 $M'M$ 的 CRC 码(二进制数 M 置于 M' 后面)。设 $M^* = m'_{t-1} \times 2^t + m'_{t-2} \times 2^{t-1} + \cdots + m'_1 \times 2^2 + m'_0 \times 2^1 + m_{n-1}$,可以写成 $M^* = 2 \times (m'_{t-1} \times 2^{t-1} + m'_{t-2} \times 2^{t-2} + \cdots + m'_1 \times 2^1 + m'_0 \times 2^0) + m_{n-1}$。

则:

$$\frac{M^* \times 2^{16}}{G} = \frac{2 \times M' \times 2^{16}}{G} + \frac{m_{n-1} \times 2^{16}}{G} \tag{3}$$

式中,$2 \times M' \times 2^{16} = m'_{t-1} m'_{t-2} \cdots m'_1 m'_0 00 \cdots 0$($m'_0$ 后有 17 个 0),右边第一项相当于在计算 $m'_{t-1} m'_{t-2} \cdots m'_1 m'_0$ 的 CRC 码(m'_0 后接续 16 个 0 除 G)之后再运行一次除 G 计算,如图 1 所示。

$$\begin{array}{l} G\overline{)\,r_{15}^0 \quad r_{14}^0 \quad \cdots \quad r_1^0 \quad r_0^0 \quad 0} \\ \quad\; r_{15}^0 \quad g_{15}r_{15}^0 \quad \cdots \quad g_2 r_{15}^0 \quad g_1 r_{15}^0 \quad g_0 r_{15}^0 \\ \hline 0 \quad r_{14}^0 + g_{15}r_{15}^0 \quad \cdots \quad 0 + g_0 r_{15}^0 \end{array}$$

图 1 式(3)第一项运算示意图

故计算结果为:

$$(r_{14}^0 + g_{15}r_{15}^0) \times 2^{15} + \cdots + (r_0^0 + g_1 r_{15}^0) \times 2^1 + 0 + g_0 r_{15}^0 \tag{4}$$

式(3)右边第二项实际上是计算 m_{n-1} 的 CRC 码,如图 2 所示。

$$\begin{array}{l} G\overline{)\,m_{n-1} \quad 0 \quad \cdots \quad 0} \\ \quad\; m_{n-1} \quad g_{15}m_{n-1} \quad \cdots \quad g_0 m_{n-1} \\ \hline 0 \quad 0 + g_{15}m_{n-1} \quad \cdots \quad 0 + g_0 m_{n-1} \end{array}$$

图 2 式(3)第二项运算示意图

故计算结果为:

$$g_{15}m_{n-1} \times 2^{15} + \cdots + g_1 m_{n-1} \times 2^1 + g_0 m_{n-1} \tag{5}$$

由式(4)和式(5)可得式(3)的 CRC 码(接收第 1 位信息码后的 CRC 码)为:

$$(r_{14}^0 + g_{15}(r_{15}^0 + m_{n-1})) \times 2^{15} + \cdots + (r_0^0 + g_1(r_{15}^0 + m_{n-1})) \times 2^1 + g_0(r_{15}^0 + m_{n-1}) = \sum_{j=1}^{15} (r_{j-1}^0 + g_j(r_{15}^0 + m_{n-1})) \times 2^j + g_0(r_{15}^0 + m_{n-1}) \tag{6}$$

若用 r_j^1 表示第 j 级触发器接收第 1 位信息码后的状态值,其中 $j = 1, \cdots, 15$,为触发器编号。则根据式(6)可得

$$\begin{aligned} r_0^1 &= g_0(r_{15}^0 + m_{n-1}) \\ r_j^1 &= r_{j-1}^0 + g_j(r_{15}^0 + m_{n-1}) \end{aligned} \tag{7}$$

式(7)给出了当读入第一位数据 m_{n-1} 后,新的 CRC 码与原来 CRC 码之间的关系。其实现电路如图 3 所示。

基于生成多项式 CRC-ITU:$x^{16} + x^{12} + x^5 + 1$ 的 CRC 串行算法实现电路可简化为图 4。

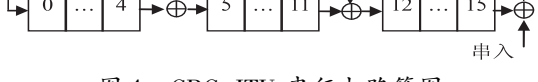

图 4 CRC-ITU 串行电路简图

n 位数据序列 M 从高位到低位依次从串行输入端输入。M 全部输入后,寄存器中的值就是所求的 CRC 校验码。这样经过 n 个时钟周期即可以算出 CRC 校验码。

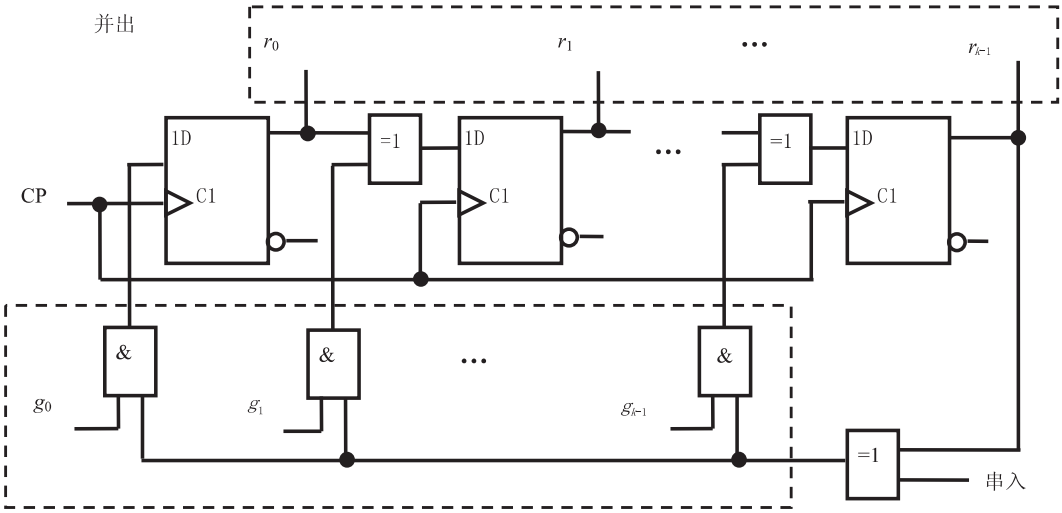

图 3　线性反馈移位寄存器电路实现 CRC

2.2　串并结合算法原理

基于比特的串行算法具有代码编写简单、修改灵活、可移植性好等优点,但是逐位处理效率太低,尤其在高速通信的场合,大量的计算会占用很多处理器的时间,甚至会因超时而导致通信失败[9]。目前,也有使用基于字节的并行算法以提高效率[10-11],但是在 EPC C1G2 协议中,涉及的信息长度不全是 8 的整数倍,因此文中提出将串行、并行两种算法相结合的新算法,以满足需求。

当进行串行运算时,当前的 CRC 值只与信息码的当前输入值 m_{n-i} 和前一 CRC 值有关。若进行并行运算,如 8 位并行运算,则 8 位信息码同时输入并行运算电路所产生的 CRC 值与串行运算时连续 8 位信息码相继输入串行运算电路所产生的 CRC 值应相同,此时,称这两种电路等效。基于这一点,可推导出 CRC 的并行计算方法[12]。

图 4 中,设 $M=(m_{n-1},m_{n-2},\cdots,m_1,m_0)$ 为输入信息码序列,其中的第 i 位用 m_{n-i} 表示,$i=1,2,\cdots,n$,为输入信息码序号;r_j^i 为第 j 级触发器接收第 i 位信息码后的状态值,r_{j-1}^{i-1} 为第 $j-1$ 级触发器接收第 $i-1$ 位信息码后的状态值,其中 $j=0,1,\cdots,15$,为触发器编号。由于第 0 级触发器无前一级触发器,可设 $r_{-1}^{i-1}=0$,则式(7)可统一用式(8)表示。

$$r_j^i=r_{j-1}^{i-1}+(g_j(r_{15}^{i-1}+m_{n-i})) \tag{8}$$

根据式(8)可递推出

$$r_0^1=r_{15}^0+m_{n-1}$$

$$r_0^2=r_{15}^1+m_{n-2}=r_{14}^0+m_{n-2}$$

$$r_0^3=r_{15}^2+m_{n-3}=r_{13}^0+m_{n-3}$$

$$r_0^4=r_{15}^3+m_{n-4}=r_{12}^0+m_{n-4}$$

$$r_0^5=r_{15}^4+m_{n-5}=r_{12}^1+m_{n-5}=r_{11}^0+r_{15}^0+m_{n-1}$$

$$r_0^6=r_{15}^5+m_{n-6}=r_{12}^2+m_{n-6}=r_{11}^1+r_{15}^1+m_{n-2}+m_{n-6}=r_{10}^0+r_{14}^0+m_{n-2}+m_{n-6}$$

$$r_0^7=r_{15}^6+m_{n-7}=r_{12}^3+m_{n-7}=r_9^0+r_{13}^0+m_{n-3}+m_{n-7}$$

$$r_0^8=r_{15}^7+m_{n-8}=r_{12}^4+m_{n-8}=r_8^0+r_{12}^0+m_{n-4}+m_{n-8}$$

同理,可推导出 $r_1^8\sim r_{15}^8$,如图 5 所示。

$r_0^8=r_8^0+r_{12}^0+m_{n-4}+m_{n-8}$
$r_1^8=r_9^0+r_{13}^0+m_{n-3}+m_{n-7}$
$r_2^8=r_{10}^0+r_{14}^0+m_{n-2}+m_{n-6}$
$r_3^8=r_{11}^0+r_{15}^0+m_{n-1}$
$r_4^8=r_{12}^0+m_{n-4}$
$r_5^8=r_8^0+r_{12}^0+r_{13}^0+m_{n-3}+m_{n-4}+m_{n-8}$
$r_6^8=r_9^0+r_{13}^0+r_{14}^0+m_{n-2}+m_{n-3}+m_{n-7}$
$r_7^8=r_{10}^0+r_{14}^0+r_{15}^0+m_{n-1}+m_{n-2}+m_{n-6}$
$r_8^8=r_0^0+r_{11}^0+r_{15}^0+m_{n-1}+m_{n-5}$
$r_9^8=r_1^0+r_{12}^0+m_{n-4}$
$r_{10}^8=r_2^0+r_{13}^0+m_{n-3}$
$r_{11}^8=r_3^0+r_{14}^0+m_{n-2}$
$r_{12}^8=r_4^0+r_8^0+r_{12}^0+r_{15}^0+m_{n-1}+m_{n-4}+m_{n-8}$
$r_{13}^8=r_5^0+r_9^0+r_{13}^0+m_{n-3}+m_{n-7}$
$r_{14}^8=r_6^0+r_{10}^0+r_{14}^0+m_{n-2}+m_{n-6}$
$r_{15}^8=r_7^0+r_{11}^0+r_{15}^0+m_{n-1}+m_{n-5}$

图 5　8 位并行计算的 CRC-ITU 逻辑关系式

串并行结合算法的实现步骤如下:

(1)判断信息序列 $M=(m_{n-1},m_{n-2},\cdots,m_1,m_0)$ 的位数 n 能否被 8 整除。设商为 q,余数为 r,若不能整除,转至(2),否则转至(3)。设循环变量 i 的初值为 q;

(2)寄存器右移一位,如图 4 所示,进行串行算法操作,循环 r 次,转至(3);

(3)对后续信息码按图 5 进行并行算法操作,$i=i-1$;

(4)判断 i 是否为 0,即信息码是否全部处理,若是,则结束,否则转至(3)。

运行结束后,寄存器中的值就是所要求的 CRC 码。

现假设输入数据 11100000 101,观察图 6,对图 6 的变量含义进行解释。CRC 校验结果为 1100100100110010(C932H),计算结果正确。由图 6 可知,采用 4 个 clk 周期即可完成 CRC 的计算。与串行算法(需 11 个 clk 周期)相比,运算速率提升了 2.75 倍,而且数据位数越多,运算速率提升越明显。

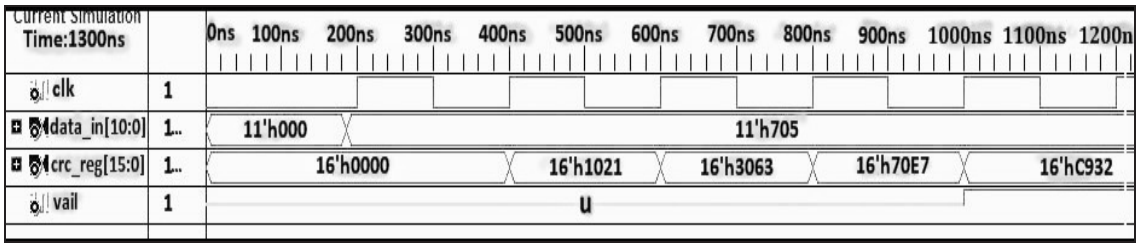

图 6 CRC 仿真图

3 结束语

串行算法电路简单,但存在校验时间长的缺点,并行算法校验时间短,但通常要求数据序列的长度为 8 的整数倍。文中提出了串并结合的算法,既提高了编码速率,又解决了数据序列长度非 8 的整数位的问题。文中推导出的算法具有通用性,可以根据不同的生成多项式推导出相似的实现算法,以运用在各种通信协议中。

参考文献:

[1] Nair R, Ryan G, Farzaneh F. A symbol based algorithm for hardware implementation of cyclic redundancy check (CRC)[C]//Proc of 1997 VHDL international user's forum. Washington, DC, USA: IEEE Computer Society, 1997.

[2] 宋富新,朱晓明,马小社. CRC 编码的并行算法与软件实现[J]. 电子科技,2007(11):62-65.

[3] 张德云,尹勇生,刘志文,等. 面向 USB 应用的 CRC 编解码电路的设计与实现[J]. 合肥工业大学学报(自然科学版),2005,28(3):292-295.

[4] 刘 峰. 超高频 RFID 读写器的研究与实现[D]. 天津:南开大学,2009.

[5] 蒋安平. 循环冗余校验码(CRC)的硬件并行实现[J]. 微电子学与计算机,2007,24(2):107-109.

[6] 朱荣华. 一种 CRC 并行计算原理及实现方法[J]. 电子学报,1999,27(4):143-145.

[7] 赵玉红. 循环冗余校验的实现方法[J]. 雷达与对抗,2006(4):25-27.

[8] 姚 威. 循环冗余校验码并行算法的研究与实现[J]. 计算机与数字工程,2006,34(9):112-114.

[9] 860 MHz-930 MHz class 1 radio frequency identification tag radio frequency & logical communication interface specification candidate recommendation, version 1.0.1[S]. [s.l.]: Auto-ID Center, 2002.

[10] EPC™ radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz version 1.1.0[S]. [s.l.]: EPC Global, 2005.

[11] Information technology automatic identification and data capture techniques-radio frequency identification for item management air interface-part 6: parameters for air interface communications at 860-960 MHz[S]. 2003.

[12] 樊昌信,张甫翊. 通信原理[M]. 北京:国防工业出版社,2005.

(上接第 102 页)

复杂系统与复杂性科学,2008,5(3):19-42.

[4] 张 聪,沈惠璋. 复杂网络中社团发现的快速划分算法[J]. 系统工程,2011,29(4):93-98.

[5] 山玉段,徐 勇,安利平. 一种复杂网络中社团划分的新算法[J]. 系统工程,2012,30(2):120-123.

[6] 程学旗,沈华伟. 复杂网络的社区结构[J]. 复杂系统与复杂性科学,2011,8(1):57-70.

[7] 骆志刚,丁 凡,蒋晓舟,等. 复杂网络社团发现算法研究新进展[J]. 国防科技大学学报,2011,33(1):47-52.

[8] 杨 博,刘大有,LIU Jiming,等. 复杂网络聚类方法[J]. 软件学报,2009,20(1):54-66.

[9] Zhang Ning, Tian Yuanyuan, Patel J M. Discovery-driven graph summarization[C]//Proc of ICDE. [s.l.]: IEEE, 2010:880-891.

[10] Zhou Yang, Cheng Hong, Yu J X. Graph clustering based on structural/attribute similarities[J]. Proceedings of the VLDB Endowment, 2009, 2(1):718-729.

[11] Elmacioglu E, Lee D. On six degrees of separation in DBLP-DB and more[J]. ACM SIGMOD Record, 2005, 34(2):33-40.

[12] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Phys Rev E, 2004, 69(2):026113.

[13] Tian Yuanyuan, Hankins R A, Patel J M. Efficient aggregation for graph summarization[C]//Proc of SIGMOD'08. New York, NY, USA: ACM, 2008:567-580.

[14] 解 冶,汪小帆. 复杂网络中的社团结构分析算法研究综述[J]. 复杂系统与复杂性科学,2005,2(3):1-12.

CRC码串并结合算法的研究与实现

作者： 王月琴， 杨恒新， WANG Yue-qin， YANG Heng-xin
作者单位： 南京邮电大学 电子科学与工程学院,江苏 南京,210003
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(6)