一种 CRC 并行计算原理及实现方法*

朱荣华
(电子科技大学光纤通信国家重点实验室,成都 610054)

【提要】 本文提出一种通用的 CRC 并行计算原理及实现方法,适于不同的 CRC 生成多项式和不同并行度(如 8 位、16 位、及 32 位等),与目前已采用的查表法比较,不需要存放余数表的高速存储器,减少了时延,且可通过增加并行度来降低高速数传系统的 CRC 运算时钟频率.

关键词: 循环冗余码的并行计算, CRC 余数, 高速数传系统

The Principle and Implementation of a Parallel CRC Computing

Zhu Ronghua
(*Optical Fiber Communication National Key Lab of UESTC, Chengdu* 610054)

Abstract: The principle and implementation of a general parallel Cyclic Redundancy Code, or CRC computing are described in the paper. It is suitable for any generator polynomial and any parallel degree of generator polynomial between 1 and 32. Compare with Table Lookup Algorithm, it need not the high speed RAM which was used to store the remainder table, and decrease the delay. Thus, we can increase properly parallel degree to decrease the clock frequency of CRC computing in high-speed digital systems.

Key words: CRC parallel computing, CRC remainder, High-speed digital system

一、引 言

循环冗余校验码简称为循环冗余码或 CRC 码(Cyclic Redundancy Check),是一种检出概率高、且易于用硬件实现的检错码. CRC 码由一个生成多项式(最高次幂为 k)产生, k 次幂的生成多项式可产生 $k-1$ 位的冗余码. 适当选取生成多项式可以使 CRC 码能检出所有奇数位的随机误码,以及突发长度小于等于 $k-1$ 的突发误码[1,3].

CRC 码的编码过程如下:

设待校验的信息码有 n 位, $M=(m_{n-1}, m_{n-2}, \cdots, m_1, m_0)$,用多项式 $M(x)$ 表示:

$$M(x)=m_{n-1}X^{n-1}+\cdots+m_1X^1+m_0 \tag{1}$$

如果所采用的生成多项式 $g(x)$ 的最高次幂为 k,则先在式(1)的两端乘以 X^k,变成:

$$X^kM(x)=m_{n-1}X^{k+n-1}+m_{n-2}X^{k+n-2}+\cdots+m_1X^{k+1}+m_0X^k \tag{2}$$

$X^kM(x)$ 模-2 除以 $g(x)$,得到商多项式为 $q(x)$,余数多项式为 $R(x)$,即:

$$X^kM(x)+R(x)=q(x)g(x) \tag{3}$$

余数多项式 $R(x)$ 可表示为:

$$R(x)=r_{k-1}x^{k-1}+r_{k-2}x^{k-2}+\cdots+r_1X^1+r_0 \tag{4}$$

将式(2)和式(4)代入式(3),得:

$$q(x)g(x)=X^kM(x)+R(x)=m_{n-1}X^{k+n-1}+\cdots+m_0X^k+r_{r-1}X^{k-1}+\cdots+r_1X^1+r_0 \tag{5}$$

式(5)所对应的码组为 $n+k$ 位,即:

$$M'=(m_{n-1}, m_{n-2}, \cdots, m_1, m_0, r_{k-1}, r_{k-2}, \cdots, r_1, r_0) \tag{6}$$

从 M 到 M',就是 CRC 的编码过程, $r_{k-1}, r_{k-2}, \cdots, r_1, r_0$ 即为校验位. 在接收端,将接收到的 $n+k$ 位码除以相同的生成多项式 $g(x)$,根据式(3),所产生余数为零,就认为接收到的信息正确无误;否则就认为在传输过程中发生了误码.

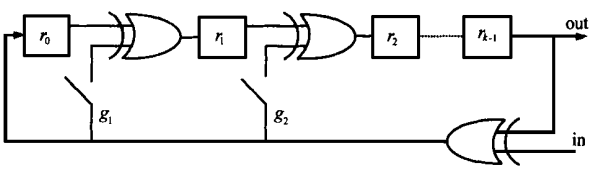

图 1 通用 CRC 串行电路

式(1)～(6)表示,CRC 编码必须进行模-2 除运算,CRC 码的校验位就是模-2 除的余数. 如果余数用寄存器的存数表示,模-2 运算用异或门表示,那么一个通用的 CRC 串行电路就可以用图 1 所示的电路来表示.

在图 1 中, $r_0, r_1, \cdots, r_{k-1}$ 表示 k 个二进制移位寄存器的存数,它由外部同步时钟启动移位. $g_i(i=1,2,\cdots,k-1)$ 表示生成多项式 $g(x)$ 的系数,当 g_i 为"1"时,开关相当于闭合;当 g_i 为"0"时,开关相当于断开. 所以,当给定一生成多项式 $g(x)$,图 1 可得到简化. 如 CRC-16 的生成多项式 $g(x)=X^{16}+X^{15}+X^2+1$,相应的串行电路可简化为(图 2).

* 1997 年 11 月收到,1998 年 4 月修改定稿

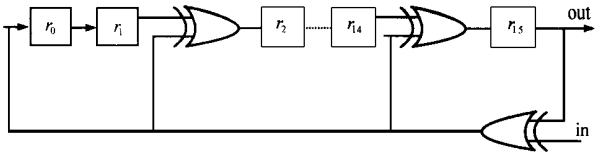

图 2 CRC-16 的串行电路

在串行电路中,只用移位寄存器和异或门,是基本的 CRC 运算电路.但在高速数字系统中,若采用串行运算,需要高速串行移位时钟及相应的高速器件,极大地增加了实现的难度,因而通常采用并行处理方法.例如,在 155Mpbs 的 ATM 网络或 1000Mbps 高速以太网等高速传输系统中,往往采用 8 位或 32 位的并行处理电路以降低时钟.因此需要研究一种并行 CRC 处理电路.

二、并行计算及实现

目前已采用的 CRC 并行算法是查表法及基于查表法而导出的一些方法[2,4].这些方法均需要存储长度较大的 CRC 余数表,随着并行度的增加,余数表的长度大大增加(按指数增加),其现实性亦随之大大降低.

笔者在设计 CRC 电路的实践过程中提出了一种并行计算法及实现方案,如下所述.

如图 1 所示,各移位寄存器的状态值即为 CRC 余数值,当进行串行运算时,当前的 CRC 余数值只与信息码的当前一位的输入值和前一状态的余数值有关.若进行并行运算,如 8 位并行运算,即 8 位信息码同时输入并行运算电路所产生的 CRC 余数与串行运算时连续 8 位信息码相继输入串行运算电路所产生的 CRC 余数相同,则称这两种电路是等效的.由此产生出 CRC 并行计算方法.其运算过程如下:

设 r_j^i 为图 1 中移位寄存器状态值,m_i 为输入信息码序列,$i=1,2,\cdots$,为输入信息码序号,$j=0,1,\cdots,k-1$,为移位寄存器编码,则

$$r_j^i = r_{j-1}^{i-1} \oplus g_j \cdot (r_{k-1}^{i-1} \oplus m_j),\ \text{当 } j=0 \text{ 时},\ r_{j-1}^{i-1}=0 \qquad (7)$$

下面以 8 位并行输入为例,推导 8 位并行计算的 CRC-16 (其中生成多项式为 $g(x)=X^{16}+X^{15}+X^2+1$,即 $K=16$)的逻辑关系式.利用式(7),可递推出 $r_0^8,\cdots,r_{15}^8$:

$$r_0^8 = r_{15}^7 \oplus m_8 = r_{15}^6 \oplus r_{14}^6 \oplus m_8 \oplus m_7 = r_{15}^5 \oplus r_{14}^5 \oplus r_{13}^5 \oplus m_8 \oplus m_7 \oplus m_6 = r_{15}^4 \oplus r_{14}^4 \oplus r_{13}^4 \oplus r_{12}^4 \oplus m_8 \oplus m_7 \oplus m_6 \oplus m_5 = r_{15}^3 \oplus r_{14}^3 \oplus r_{13}^3 \oplus r_{12}^3 \oplus r_{11}^3 \oplus m_8 \oplus m_7 \oplus m_6 \oplus m_5 \oplus m_4$$
$$= r_{15}^2 \oplus r_{14}^2 \oplus r_{13}^2 \oplus r_{12}^2 \oplus r_{11}^2 \oplus r_{10}^2 \oplus m_8 \oplus m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3$$
$$= r_{15}^1 \oplus r_{14}^1 \oplus r_{13}^1 \oplus r_{12}^1 \oplus r_{11}^1 \oplus r_{10}^1 \oplus r_9^1 \oplus m_8 \oplus m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3 \oplus m_2$$
$$= r_{15}^0 \oplus r_{14}^0 \oplus r_{13}^0 \oplus r_{12}^0 \oplus r_{11}^0 \oplus r_{10}^0 \oplus r_9^0 \oplus r_8^0 \oplus m_8 \oplus m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3 \oplus m_2 \oplus m_1 \qquad (8)$$

如式(8)所示,类似地可推导出 $r_1^8,\cdots,r_{15}^8$.并定义:

$$r_j^{n-1} = r_j^0,\ r_j^n = r_j^8,\ j=0,1,\cdots,15$$
$$s_i = r_{k-1}^{n-1} \oplus m_i,\ i=1,2,\cdots,8 \qquad (9)$$

则可得表 1.

表 1 8 位并行计算的 CRC-16 逻辑关系式

$r_0^n = s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5 \oplus s_6 \oplus s_7 \oplus s_8$	$r_8^n = r_0^{n-1} \oplus s_1 \oplus s_2$
$r_1^n = s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5 \oplus s_6 \oplus s_7$	$r_9^n = r_1^{n-1} \oplus s_1$
$r_2^n = s_7 \oplus s_8$	$r_{10}^n = r_2^{n-1}$
$r_3^n = s_6 \oplus s_7$	$r_{11}^n = r_3^{n-1}$
$r_4^n = s_5 \oplus s_6$	$r_{12}^n = r_4^{n-1}$
$r_5^n = s_4 \oplus s_5$	$r_{13}^n = r_5^{n-1}$
$r_6^n = s_3 \oplus s_4$	$r_{14}^n = r_6^{n-1}$
$r_7^n = s_2 \oplus s_3$	$r_{15}^n = r_7^{n-1} \oplus s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5 \oplus s_6 \oplus s_7 \oplus s_8$

根据表 1 的逻辑关系式容易直接实现 8 位并行的 CRC-16 的硬件运算电路,图 3 为其硬件示意图.

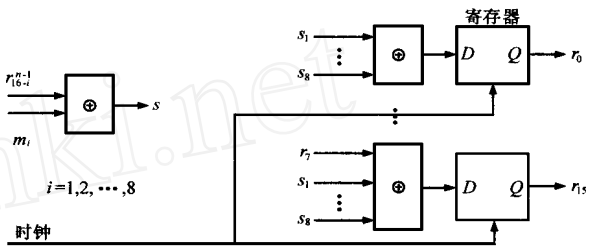

图 3 8 位并行的 CRC-16 硬件实现示意图

上述方法的推导过程虽然是针对 8 位并行 CRC-16 运算为例进行的,但这种方法具有通用性,即用这种方法可推导各种 CRC 生成多项式的各种并行度的并行计算逻辑关系式.

三、性能比较

将本文提出的方法与查表法作比较.

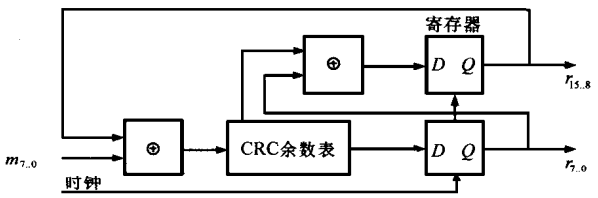

图 4 8 位并行 CRC 查表法硬件实现示意图

查表法可以用软件实现[2,3],也可以用硬件实现(图 4).如图 4 所示,要进行一个 CRC 生成多项式的运算,需要在余数表中存放相应的余数.

将图 3 与图 4 加以比较可得到如下结论:

(1) 本文提出的 CRC 实现法消除了查表法所必须的 CRC 余数表.降低了成本,提高了计算速度.以 Xilinx FPGA XC4003E-3 实现 CRC-16 为例,按本文提出的直接硬件实现法,如图 3 所示,可以全部用 FPGA 的内部资源实现,总的时延为两级异或门时延和寄存器的锁存时延之和,约为 15ns;如图 4 所示.查表法不仅需要 FPGA 内部资源,还必须使用 FPGA 的输入/出引脚和外部高速存储器,总的时延为两级异或门时延、寄存器的锁存时延、输入/出引脚时延和外部存储器的读时延之和,大于 100ns.在查表法中,由于需要输入输出引脚和外部存储器,所以时延较大,限制了使用更高的处理时钟.

(2)本文所提出的 CRC 实现法可提高并行计算的并行度.在查表法中,不管是软件实现还是硬件实现,并行度一般不超过 8 位,因为随着并行度的增大,CRC 余数表将急剧增大.例如,并行度为 16 位时,CRC 余数表的长度将达到 65536(2^{16})项,而并行度为 32 位时,CRC 余数表的长度将达到 4,294,967,296(2^{32})项,这是不现实的.在实现高速 CRC 计算时,查表法因其大并行度的实现受限,而须较高的处理时钟.往往须采用 ECL 之类昂贵的器件来实现,如 Jesper[4]等人,为了实现一个 800Mbps 的 CRC 计算,就采用 ECL 电路.而本文所提出的 CRC 实现法可灵活地实现各种并行度的 CRC 计算.由于可以采用更大的并行度(如 32 位并行计算,甚至 64 位的并行计算),降低了处理时钟,用普通的 CMOS 器件就可达到很高速率的 CRC 计算.如用 Xilinx FPGA XC4000E 就可实现超过 2000Mbps 的 CRC 计算.

四、结束语

本文在分析了 CRC 计算原理的基础上,提出了一种推导和实现 CRC 并行计算的通用方法.用这种方法,可以灵活实现各种 CRC 生成多项式在不同并行度下的计算以及硬件实现(采用可编程器件或ASIC专用芯片的设计),尤其对需要提高并行度(大于 8)的高速系统的 CRC 计算,较之查表法具有现实性及优越性.

参 考 文 献

1 邵军力,杨心强,钱水春.数据通信技术基础.成都电讯工程学院出版社,1988,10

2 A. Perez. Byte-wise CRC calculations. IEEE Micro,1983,3(3):40～50

3 T. V. Ramabadran et al. A tutorial on CRC computations. IEEE Micro,1988,8(4):62～75

4 Jesper Birch et al. A programmable 800 Mbit/s CRC check/generator unit for LANs and MANs Computer Networks and ISDN Systems. 1992,24,109～118

朱荣华 讲师,1989 年毕业于电子科技大学,1993 年获硕士学位,曾参加光纤通信,电视图像传输等多项重点课题的研究工作.现主要研究 SDH 和 ATM 网管.

1999 年第 4 期 Acta Electronica Sinica No.4 1999

电子学报

(总期 180 期) (Monthly) (Series No.180)

主办单位	中国电子学会	Published by the Chinese Institute of Electronis,Beijing
协办单位	中国计算机报社	China Infoworld
编辑	《电子学报》编辑委员会	Edited by Editorial Board of Acta Electronica Sinica
主编	王守觉	Chief Editor:Wang Shoujue
总编辑	刘力	Director:Liu Li
通信处	北京 165 信箱 (邮政编码 100036)	Editorial Office of Acta Electronica Sinica (P.O.Box 165, Beijing 100036,China)
电话	(010)68285082	Tel (010)68285082
传真	(010)68173796	Fax (010)68173796
排版印刷	中国纺织印刷厂	Printed by Textile Printinghouse,China
国内总发行	北京市报刊发行局	Distributed by Domestic:Beijing Baokan Faxingju,China
国内订购处	全国各邮电局	Subscription Office ——All Local Post Offices in China
国外总发行	北京国际书店 (北京 399 信箱)	Foreign:Guoji Shudian (P.O.Box 399,Beijing,China)

国内统一刊号:CN11-2087/TN 邮发代号(国内/国外):2-891/M436 国内定价 20.00