

# 一种高速嵌套 CRC 码的生成方法及其 FPGA 实现

段斌斌 孙嵩松 焦黎 周文利

(华中科技大学光学与电子信息学院 武汉 430074)

**摘要** 为了实现高速融合网络数据传输中的差错控制,针对现有循环冗余校验码(CRC)计算速度难以进一步提升的问题,提出了一种用嵌套 CRC 码实现高速数据差错控制的方法,并在 Xilinx 公司的 FPGA 芯片上进行了实现。该嵌套 CRC 码由多个通道的传统 CRC 码并行计算器同步计算得到,可大幅度提升差错控制码的生成速度,并通过不同计算通道的组合,支持多种流量的差错控制。最后分析了嵌套 CRC 码的计算性能以及差错控制能力,并提供了设定嵌套次数、通道数以及计算通道并行计算位数的依据。

**关键词** 高速数据传输,嵌套 CRC 码,并行计算器,多通道,多流量

**中图法分类号** TPN914 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.019

## High-speed Nested CRC Code Generation Method and Implementation

DUAN Bin-bin SUN Song-song JIAO Li ZHOU Wen-li

(School of Optical and Electronic Information, Huazhong University of Science & Technology, Wuhan 430074, China)

**Abstract** To achieve the data error control in high-speed converged network data transmission, a nested CRC code generation method was proposed to improve the situation that it is difficult to further enhance the computing speed through currently available cyclic redundancy check (CRC) code calculation technique. It was implemented on Xilinx Field Programmable Gate Array (FPGA) chip. This nested CRC code is obtained through calculating the traditional CRC code concurrently in multiple channels, and thus the speed of error control code generation is highly increased while multiple-types of data flows are processed by different kinds of calculating channels. At the end, the calculating performance and error control capability were analyzed and a guidance to set the nested level, the number of computing channel and parallel computing width of a single computing channel was given.

**Keywords** High-speed data transmission, Nested CRC code, Parallel calculator, Multiple channels, Multiple-types data flows

## 1 引言

随着通信网络速率的快速提升,10GB 以太网日益普及,40G 和 100G 以太网也呼之欲出<sup>[1]</sup>;同时,数据中心中多种数据传输网络的存在给数据中心带来了布线复杂、接口种类繁多、能源损耗大、管理成本高等缺点<sup>[2]</sup>。因此,网络的融合也成为通信领域发展的趋势,这要求通信设备能够高速处理多种数据流量。相应地,通信网络中传统的数据差错控制方法亟待改进,一方面需要提升数据差错控制的速度<sup>[3,4]</sup>,另一方面需要同时支持多种数据流量的差错控制。

目前,通信网络中通常采用循环冗余校验码来实现数据差错控制,CRC 码的生成可以用软件和硬件来实现。根据 CRC 的基本原理,直接利用计算公式按位进行计算<sup>[5]</sup>,即串行实现,一个时钟周期只能处理 1bit 的数据。CRC 码的生成可以通过并行处理来提高处理速度,并行 CRC 码生成一般采用查找表<sup>[6,7]</sup>和计算公式预推导<sup>[8-10]</sup>两种方法来实现,这能在一定程度上提升 CRC 码的计算速度,但仍然是一个 CRC 码计算器对一个数据进行计算,无法实现多个 CRC 计算器对一

个数据的各部分同时进行计算,而且随着并行的位数增加,计算需要的资源大大增加,硬件实现中串联的电路的结构也会限制最高计算频率的增加<sup>[11]</sup>。文献<sup>[12]</sup>中提出了一种多通道并行计算 CRC 码的方法,该方法可以实现多个通道同时对数据的不同部分进行多位并行计算,但是需要为各通道建立余数转换表,随着 CRC 码位数的增加,余数转换表所需的存储量将呈指数上升。

本文针对我们项目研究的光纤以太网(Fibre Channel Over Ethernet, FCoE)传输接口芯片中 10Gbps 的高速通信需求,提出了一种高速嵌套 CRC 码来实现数据的差错控制,并通过 FPGA 芯片实现了相应的嵌套 CRC 码计算器。

## 2 嵌套 CRC 码计算及差错控制原理

假定待计算数据的长度为  $L_F$ ,按照长为  $L_S$  将其分成若干段,如果  $L_F$  不是  $L_S$  的整数倍,则假定最后一段数据长度为  $L_F$ ,嵌套 CRC 码的嵌套次数为  $Q$ 。嵌套 CRC 码是指,在将长为  $L_F$  的源数据分成若干组后,利用多个 CRC 码计算通道对各个组分别计算 CRC 码,然后将计算得到的各个 CRC 码

到稿日期:2013-11-19 返修日期:2014-01-29 本文受国家高技术研究发展计划(863 计划)课题(2012AA012403)资助。

段斌斌(1988—),男,硕士,主要研究方向为存储网络通信;孙嵩松 男,硕士,主要研究方向为通信芯片设计和通信协议研究;焦黎 男,主要研究方向为通信协议和算法;周文利(1968—),女,教授,主要研究方向为 FPGA 及嵌入式系统,E-mail:wzlhust@hust.edu.cn。

重新组成一组新的数据,对其进行上述的嵌套 CRC 码计算过程,计算  $Q-1$  次后,将  $Q-1$  次计算得到的若干嵌套 CRC 码组成新的数据,最后由一个单独的 CRC 码计算器计算得到  $Q$  次嵌套的嵌套 CRC 码。图 1 为一次嵌套 CRC 码的计算过程示意图。

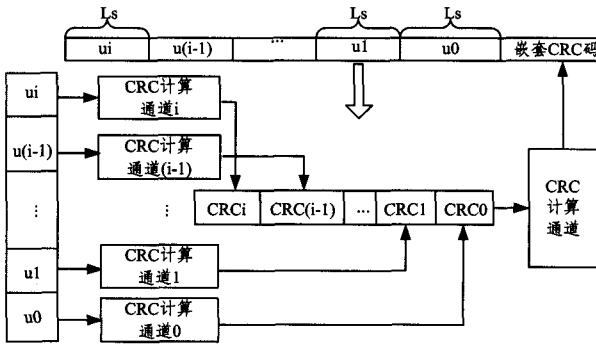


图 1 嵌套 CRC 码计算过程

设待计算 CRC 码的数据比特序列为  $\{m_{k-1}, m_{k-2}, m_{k-3}, \dots, m_i, \dots, m_1, m_0\}$ , 其多项式为:

$$m(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \dots + m_0x^0$$

其中  $m_i = 1$  或  $0$ ; 将待计算数据分成  $m-1$  个长度为  $L_s$  位的数据段  $(D_m, D_{m-1}, \dots, D_2)$  和一个长度为  $L_F$  位 ( $1 \leq L_F \leq L_s$ ) 的数据段  $D_1$ , 对  $D_1$  进行补零, 将其长度扩展为  $L_s$ , 得到扩展以后的  $m$  段数据的多项式为:

$$\begin{cases} D_m(x) = m_{k-1}x^{L_s-1} + m_{k-2}x^{L_s-2} + \dots + m_{k-L_s}x^0 \\ D_{m-1}(x) = m_{k-L_s-1}x^{L_s-1} + \dots + m_{k-2L_s}x^0 \\ \vdots \\ D_1(x) = m_{k-(m-1)L_s-1}x^{L_s-1} + \dots + m_{L_s-L_F}x^{L_s-L_F} \end{cases}$$

对于每一个数据段, 数据位数均为  $L_s$ , 传统 CRC 校验码的位数为  $R$ , 两者组成的码字一般记为  $(L_s + R, L_s)$ , 每个数据段采用独立的生成多项式, 根据 CRC 原理有:

$$\begin{cases} x^R D_m(x) = Q_m(x)g_m(x) + r_m(x) \\ x^R D_{m-1}(x) = Q_{m-1}(x)g_{m-1}(x) + r_{m-1}(x) \\ \vdots \\ x^R D_1(x) = Q_1(x)g_1(x) + r_1(x) \end{cases}$$

$Q(x)$  为  $x^R D(x)$  除以  $g(x)$  所得的商多项式,  $r(x)$  为余数多项式, 各数据段可采用同样的生成多项式。计算通道计算产生的  $m$  个余数多项式为  $(r_1(x), r_2(x), \dots, r_m(x))$ , 根据 CRC 原理, 并结合  $m$  个余数多项式, 分别对待计算数据的各数据块实现差错控制, 因此由  $(r_1(x), r_2(x), \dots, r_m(x))$  码字可以实现整个待计算数据的差错控制。

$(r_1(x), r_2(x), \dots, r_m(x))$  的位宽是原 CRC 码位数的  $m$  倍, 为了保持信道的编码率, 并不增加存储开销, 将  $(r_1(x), r_2(x), \dots, r_m(x))$  作为新的待计算数据, 进行 CRC 码嵌套计算。当嵌套数  $Q=1$  时, 由  $(r_1(x), r_2(x), \dots, r_m(x))$  组成数据的多项式为:

$$m(x)^1 = m_{mR-1}x^{mR-1} + m_{mR-2}x^{mR-2} + \dots + m_0x^0$$

新的数据长度为  $mR$ , 传统 CRC 校验码位数为  $R$ , 两者组成的码字为  $((m+1)R, mR)$ , 该组数据采用的生成多项式为  $g_{m+1}(x)$ , 根据 CRC 原理有:

$$x^R m(x)^1 = Q(x)^1 g_{m+1}(x) + r(x)^1$$

其中,  $Q(x)^1$  为  $x^R m(x)^1$  除以  $g_{m+1}(x)$  所得的商多项式,  $r(x)^1$  为余数多项式。

在数据的传送过程中, 可以由  $r(x)^1$  对数据  $(r_1(x), r_2(x), \dots, r_m(x))$  的实现进行差错控制, 而  $(r_1(x), r_2(x), \dots, r_m(x))$  可以对原待计算数据进行差错控制, 因此通过  $r(x)^1$  可以对原待计算数据实现差错控制。若嵌套计算后的 CRC 码字总长过长, 可以通过增加  $Q$  的值来增加计算速度。

CRC 码和传统的 CRC 码的长度一样, 在数据传递的过程中不会降低码率, 通过多个传统 CRC 码计算器同时对各数据段进行计算得到, 从而使计算速度成倍提升, 并且通过计算通道选择, 实现多种流量数据差错控制的支持; 通过嵌套计算省去文献[5]中的存储开销; 嵌套 CRC 码通过层次化的差错控制, 由上述计算过程可以看到, 一个 CRC 码完成数据差错控制的数据长度平均为  $L_s$ , 而传统 CRC 码数据差错控制的数据长度为  $L_F$ , 因此嵌套 CRC 码中, 一个 CRC 码需要进行差错控制的数据长度远小于传统 CRC 码, 因此差错控制能力得到提升。

### 3 嵌套 CRC 码计算器的实现

#### 3.1 嵌套 CRC 码计算流程

整个嵌套 CRC 码计算器的计算流程如图 2 所示。

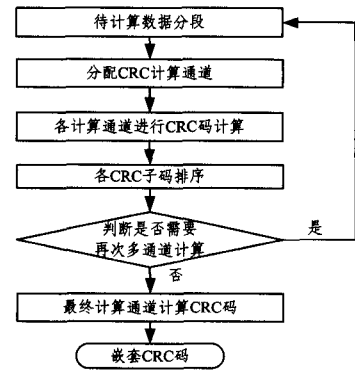


图 2 嵌套 CRC 码计算流程图

首先对待计算数据进行分段, 根据数据类型为各数据段分配所需 CRC 码计算通道, 各计算通道同时进行 CRC 码计算, 对计算得到的各 CRC 子码进行排序, 使 CRC 子码按原数据段排列顺序进行排列。判断 CRC 码嵌套的次数, 若嵌套完成, 则排序后的 CRC 子码可直接送入最终的 CRC 码计算通道, 若嵌套未完成, 则作为新的待计算数据, 多次重复上述 CRC 子码计算再送入最终的 CRC 码计算通道生成最终的嵌套 CRC 码。

#### 3.2 嵌套 CRC 码计算系统组成

整个嵌套 CRC 码计算系统主要包括以下几个部分: 数据分段模块、计算通道选择模块、多通道 CRC 码计算模块、反向计算通道选择模块、CRC 字码暂存模块、计数器、单通道 CRC 码计算模块组成, 系统的原理框图如图 3 所示。

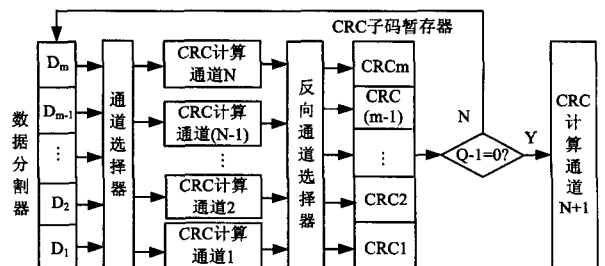


图 3 嵌套 CRC 码计算系统原理框图

### 3.2.1 数据分段模块

将待计算数据进行分段,其中对于最后一个数据段长度小于分段长度的情况,采取数据补零将数据长度扩展为分段长度。同时数据分割器需要记录每一次数据分段的段数,将该值传递给通道选择器来进行通道选择。

### 3.2.2 计算通道选择模块

为了增加计算系统的灵活性,使嵌套 CRC 码可以动态地根据流量的类型选择不同的计算方式,以实现不同生成多项式的 CRC 码计算通道。在待计算数据过程中,可以灵活根据待计算数据流量的类型,为其选择相应的 CRC 码计算通道来计算各数据段的 CRC 子码。通道选择器通过一个通道选择表来实现,根据待计算数据的流量以及待计算数据的分段数选择通道开关矢量,将待计算数据的各个段输入到相应的 CRC 码计算通道。

### 3.2.3 多通道 CRC 码计算模块

根据实际需要计算数据流量的类型,生成数量合适的 CRC 码计算通道,且各 CRC 码计算通道可以采用不同的生成多项式,由通道选择器来为待计算数据选择合适的计算通道。

### 3.2.4 反向计算通道选择模块

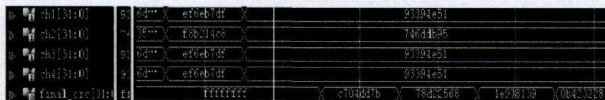
反向通道计算模块中的开关矢量和计算通道选择模块的选择矢量是相反的,因此如果在计算通道选择过程中导致 CRC 字码的乱序,通过反向通道选择可以将各通道的 CRC 字码按照数据段在源数据中的顺序进行排列组成新的数据。

## 3.3 嵌套 CRC 码生成系统的 FPGA 实现

本文采用 Xilinx 的 Spartan-3AN 系列芯片对嵌套 CRC 码的生成系统进行硬件实现,所实现的计算器的并行计算通道数为 4,每个通道采用 32bit 位宽的并行 CRC 计算器,通道的生成多项式均为 IEEE 802.3 CRC-32 所采用的生成多项式。计算器由 Verilog 语言完成电路描述,并通过 Xilinx 的 Isim 进行仿真。下面举例说明嵌套次数为 1 和 2 的 CRC 码生成过程及结果。



(a) 1 次嵌套计算



(b) 2 次嵌套计算

图 4 两层嵌套 CRC 计算器仿真结果图

输入 62 个 32 位的待计算数据,数据按照 16 个字的长度分成 4 段,最后一个通道只有 14 个字,通过补 0 达到 16 个字的长度,4 个通道的数据通过 4 个计算通道经过 16 个时钟计算得到 4 个 CRC 子码,如图 4(a)中 ch1-4 所示,4 个 CRC 子码组成新的待计算数据,由最后的 CRC 码计算通道 final\_crc 经过 4 个时钟周期后计算得到最后的嵌套 CRC 码,计算结果为 0x5501C303,如图 4(a)所示。整个计算过程需要 20 个时钟。

嵌套次数为 2 时,将第一次嵌套计算得到的 CRC 子码即图 4(a)中的 Ch1-4 组成 4 个长度的新的待计算数据再次进行嵌套计算,这次嵌套只占用一个计算通道(如图 4(b)中 Ch2)

进行计算,经 16 个时钟产生一个 32bit 的 CRC 子码,最后的 CRC 码计算通道(如图 4(b)中 final\_crc)需一个时钟,计算结果为 0xC704DD7B,嵌套次数为 2 的计算过程一共需要 33 个时钟周期。因此嵌套次数不同,计算结果不同,收发双方需要预先确定嵌套次数。上述情况下,两次嵌套比一次嵌套需要的时间长,两次嵌套没有必要,因此实际通信需要合理确定计算器的嵌套次数。下面我们综合考虑嵌套 CRC 码中通道数、嵌套次数以及通道并行处理位数对嵌套算法性能的影响。

## 4 算法的性能分析

设待计算数据的长度为  $L_F$ ,校验码长  $R$ ,若采用传统的  $W$  位并行 CRC 计算器计算 CRC 码,以计算时钟周期为单位,则需要  $\lceil L_F/W \rceil$  个时钟周期才能得到计算结果。在嵌套 CRC 码的计算过程中,若嵌套的次数为  $Q$ ,待计算数据按照长度为  $L_S$  分割成  $M$  段,一般取  $L_S = bR$ ,其中  $b$  为正整数,则计算通道数为  $M = \lceil L_F/L_S \rceil$ , $M$  段数据通过  $M$  个  $W$  位并行 CRC 码计算器同时进行计算,一般取  $W = aR$ ,其中  $a$  为正整数,第  $i$  次嵌套计算完成后产生的 CRC 字码个数为  $f(i)$ ,则  $f(i)$  的计算公式如下:

$$f(i) = \begin{cases} M, & i=1 \\ \lceil \frac{R \times f(i-1)}{L_S} \rceil, & i=2,3,4,\dots,Q \end{cases} \quad (1)$$

将  $L_S = bR$  代入式(1)得:

$$f(i) = \begin{cases} M, & i=1 \\ \lceil \frac{f(i-1)}{b} \rceil, & i=2,3,4,\dots,Q \end{cases} \quad (2)$$

通道完成一次计算需要的时钟数  $t_q = (L_S/aR)Q = (b/a)Q$ ,经过  $Q$  次计算后,剩余的  $R$  位 CRC 码数量为  $f(Q)$ , $f(Q)$  个  $R$  位 CRC 码经过最后的计算通道计算出嵌套 CRC 码需要的时钟数量为  $t_l = f(Q)/a$ ,整个计算过程需要的时钟数量为  $T = t_q + t_l = (b/a)Q + f(Q)/a$ 。在第  $i$  次嵌套计算完成后,剩余的  $R$  位 CRC 码数量为  $f(i)$ ,若进行第  $i+1$  次嵌套计算,则还需要  $b/a + \lceil f(i)/b \rceil/a$  个时钟,若不进行第  $i$  次嵌套计算,则需要  $f(i)/a$  个时钟,因此, $i$  次嵌套计算后, $i$  加 1,则所需的总时钟数量增加  $\Delta i = b/a + \lceil f(i)/b \rceil/a - f(i)/a$ 。为了使  $T$  取得最小值,则  $\Delta i \leq 0$ ,即:

$$b + \lceil \frac{f(i)}{b} \rceil \leq f(i) \quad (3)$$

因此,为了使计算器的性能最好,嵌套次数  $Q$  应该取不等式域中的最大值。

针对我们项目中 FCoE 传输接口要求,需要满足 10Gbps 的通信速度,若待计算数据的数据长度为 2kB,当取  $W = R = 32$  时,通道数量的变化对计算频率的影响如表 1 所列,当  $W = 2R$  时,计算频率如表 1 所列,表中的  $Q$  值由不等式(3)求得。

表 1 不同参数下嵌套 CRC 计算器工作频率

M	b	Q	Fclk(MHz) (W=R)	Fclk(MHz) (W=2R)
4	128	1	82.5	41.3
8	64	1	45	22.5
16	32	1	28.1	14.1
32	16	2	21.3	10.7
64	8	2	15	7.5

(下转第 109 页)

the 2008 International Conference on Advanced Computer Theory and Engineering. 2008;489-493

- [7] Wang Zhongyuan, Qi Qing-wen, Xu Li. Cluster analysis based on spatial feature selecting in spatial data mining [C]//Proceedings of the 2008 International Conference on Computer Science and Software Engineering. 2008;386-389
- [8] Yang Jiann-min, Wu Wen-chin, Liao Wei-cheng, et al. Trend analysis of machine learning—A text mining and document clustering methodology [C]//Proceedings of the 2009 International Conference on New Trends in Information and Service Science. 2009;481-486
- [9] 梁保松, 曹殿立. 模糊数学及其应用[M]. 北京: 科学出版社, 2007
- [10] 李文娟, 张启飞, 平玲娣, 等. 基于模糊聚类的云任务调度算法[J]. 通信学报, 2012, 33(3): 146-154
- [11] Sun Da-wei, Chang Gui-ran, Jin Li-zhong, et al. Optimizing grid resource allocation by combining fuzzy clustering with application preference [C]//Proceedings of the 2nd IEEE International Conference on Advanced Computer Control. 2010;22-27
- [12] 陈艳金. MapReduce 模型在 Hadoop 平台下实现作业调度算法的研究和改进[D]. 广州: 华南理工大学, 2011
- [13] 夏祎. Hadoop 平台下的作业调度算法研究与改进[D]. 广州: 华南理工大学, 2010
- [14] 杜晓丽, 蒋昌俊, 徐国荣. 一种基于模糊聚类的网格 DAG 任务图调度算法[J]. 软件学报, 2006, 17(11): 2277-2288
- [15] Luo Mei, Zhang Kai-long, Yao Long-hui, et al. Research on resources scheduling technology based on fuzzy clustering analysis

[C]//Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery. 2012;152-155

- [16] Li Wen-juan, Zhang Qi-fei, Wu Ji-yi, et al. Trust-based and QoS demand clustering analysis customizable cloud workflow scheduling strategies [C]//Proceedings of the 2012 IEEE International Conference on Cluster Computing Workshops. 2012;111-119
- [17] Li Fu-fang, Qi De-yu. Research on grid resource allocation algorithm based on fuzzy clustering [C]//Proceedings of the 2nd International Conference on Future Generation Communication and Networking. 2008;162-166
- [18] 刘伯成, 陈庆奎. 云计算中的集群资源模糊聚类划分模型[J]. 计算机科学, 2011, 38(10A): 157-160
- [19] 那丽春. 集群资源模糊聚类划分模型[J]. 计算机工程, 2012, 38(6): 34-36
- [20] Sejun K, Donald C. A GPU based parallel hierarchical fuzzy ART clustering [C]//Proceedings of the 2011 International Joint Conference on Neural Networks. 2011;2778-2782
- [21] Zhao Wei-zhong, Ma Hui-fang, He Qing. Parallel kmeans clustering based on mapreduce [C]//Proceedings of the 1st International CloudCom Conference. 2009;674-679
- [22] Aleksandar S, Sebastian M, Ralf S. RankReduce-processing k-nearest neighbor queries on top of mapreduce [C]//Proceedings of the 8th Workshop on Large-Scale Distributed Systems for Information Retrieval. 2010;13-18
- [23] 江小平, 李成华, 向文, 等. k-means 聚类算法的 MapReduce 并行化实现[J]. 华中科技大学学报: 自然科学版, 2011, 39: 120-124

(上接第 103 页)

根据 3.3 节的仿真结果, 由 Spartan-3AN 系列芯片实现并行 32 位 CRC 计算器, 生成多项式 IEEE 802.3 CRC-32, 一般需要 32 个寄存器和 238 个四输入的查找表实现, 而并行 64 位的 CRC 计算器则需要 32 个寄存器和 378 个四输入的查找表实现。考虑到查找表的利用率, 并行 64 位的 CRC 计算器所需要的资源约为 32 位计算器的 2 倍, 同样, 随着通道数量的增加, 所需的硬件资源成倍增加。这种情况下, 若直接采用单通道传统的计算方法, 为了满足 10Gbps 的速度, 32 位的并行计算器的工作频率为 312.5M, 64 位的计算器的工作频率为 156.25M。由表 1 可以看出,  $Q$  值取最佳值, 当通道数量较小时, 随着通道数  $M$  的成倍增加, 计算频率成倍减小, 当通道数较多时, 计算器工作频率的减小趋势减弱, 因此, 在实际使用过程中, 当硬件资源有限时, 可以采用较少的通道数量获得较小的计算频率来满足较大的计算带宽, 若硬件资源足够, 也可以设置足够的通道数量以及单通道并行计算的位数来实现所需的工作频率。因此嵌套 CRC 码可以很好地用于 10Gbps 通信网络, 通过取合适的  $M$ 、 $Q$ 、 $W$  的值, 同样可以将嵌套 CRC 码应用于 40Gbps 甚至 100Gbps 的通信网络。

**结束语** 本文提出了一种嵌套 CRC 码来实现数据的差错控制, 通过 Spartan-3AN FPGA 芯片实现了嵌套 CRC 码计算器。通过对嵌套 CRC 码的计算性能和差错控制能力的分析, 讨论了嵌套 CRC 码中通道数量、嵌套次数以及通道并行处理位数对计算性能的影响, 并给出了取值的依据。该方法有望为日益发展的聚合网络和数据中心提供高速数据差错控制途径。

## 参 考 文 献

- [1] 路渭华. 下一代以太网发展趋势[J]. 光通信技术, 2007, 12: 7-9
- [2] Gai S, DeSanti C. 思科数据中心 I/O 整合[M]. 陈柳, 译. 北京: 邮电出版社, 2013: 2-12
- [3] 彭建辉. 10G 以太网接口并行 CRC 校验的一种简化算法[J]. 微计算机信息, 2006, 20: 213-215
- [4] Renuka H K, Jayashree C N. Design and Computation of Cyclic Redundancy Code for Ethernet Application; an Implementation Using FPGA [J]. World Journal of Science and Technology, 2011, 1(8): 68-73
- [5] 杨梅娟, 尹德春. CRC 算法的研究[J]. 计算机与数字工程, 2005, 34: 30-32
- [6] Kounavis M E, Berry F L. Novel Table Lookup-Based Algorithms for High-Performance CRC Generation[J]. IEEE Transactions on Computer Society, 2008, 57(11): 1550-1560
- [7] 梁海华, 盘丽娜, 赵秀兰, 等. CRC 查询表及其并行矩阵生成方法[J]. 计算机科学, 2012, 39(B06): 154-158
- [8] 岳天天. 一种并行 CRC 校验算法的 IP 设计与实现[J]. 广东通信技术, 2013(3): 78-79
- [9] Sprachmann M. Automatic Generation of Parallel CRC Circuits [J]. IEEE Design & Test of Computers, 2001, 18(3)
- [10] 俞迅. 32 位 CRC 校验码的并行算法及硬件实现[J]. 信息技术, 2007(4): 71-74
- [11] 廖海红. 通信系统中的 CRC 算法的研究和工程实现[D]. 北京: 北京邮电大学, 2006: 60-65
- [12] 徐展琦, 裴昌幸, 董淮南. 一种通用多通道并行 CRC 计算及其实现[J]. 南京邮电大学学报: 自然科学版, 2008(2): 53-57