基于 FPGA 技术的多通道 CRC 校验系统①

李洪进[1,2]，邓世昆[1]

[1](云南大学 信息学院，昆明 650091)
[2](遵义医学院 医学信息工程系，遵义 563000)

摘 要：CRC 编码由于其简单的编码规则的在网络及存储等诸多场合得到广泛应用，随着现代存储和传输技术的发展，软件编码校验已难以满足 Gbit 级高速传输的需要。基于 FPGA 技术设计了一个采用多通道高度并行技术实现的高速循环冗余校验(CRC)系统。系统采用五个 2Gbps 校验通道并行工作的方式来达到 10Gbps 的数据吞吐率，系统实现采用 VerilogHDL 硬件描述语言设计，在 QuartusII8.0 平台上进行综合与布线，并将该处理单元封装为独立的 IP 核，并以 Altera 公司的 EP2C20F484C6 芯片为下载目标进行实现验证。综合结果表明，本设计可满足高速数据完整性检查的速率要求。

关键词：循环冗余校验；FPGA；10Gbps 以太网；伽罗瓦域

Multi-Channel CRC Checking System Based on FPGA Technology

LI Hong-Jin[1,2], DENG Shi-Kun[1]

[1](Information College, Yunnan University, Kunming 650091, China)
[2](Zunyi Medical College, Zunyi 563000, China)

Abstract: CRC codes are widely used in networking and storage, and many other occasions due to its simple encoding rules. With the development of modern storage and transmission technologies, the check of software code has been difficult to meet the needs of high level transmission of Gbit. This paper has been achieved highly parallel cyclic redundancy check (CRC) system which based on FPGA technology to design a multi-channel high-speed technology. The design uses five parallel channels of 2Gbps check in order to achieve data throughput rate of 10Gbps. Each CRC channel compatible with 32-bit Ethernet standard. This design uses VerilogHDL for hardware description language, QuartusII8.0 for Integrated wiring, and packaging the processing unit into an independent IP core, then uses the Altera Corporation's EP2C20F484C6 chips as download target for verification. The results show that the design can meet the rate's requirements of high-speed data integrity checks.

Key words: CRC; FPGA; 10Gbps Ethernet; Galois Fields

1 前言

在大量数据交互的信息时代，我们都希望数据在网络或其它形式的传输中尽可能不要出错，可是由于传输介质材质瑕疵、接口干扰和其它可能导致数据出错或丢失的因素影响下，我们必须面对无法避免的数据传输出错问题。故而需要对接收的数据进行错误检测并尽可能地进行修正。在早期，数据传输速率并不快的情况下，这个问题可以用不同的方法实现，也随之诞生了许多种校验策略。这些校验策略各具特点，但几乎所有的策略都是基于对传输位的数学运算来实现的。

CRC 技术[1,2]是用来检测数据完整性的一种技术，被广泛应用于网络和数据存储领域。如果数据的发送者使用 CRC 技术来产生检查序列窗口（Frame Check Sequence）并将其加入到所传数据的结尾，那么接收者就可以通过对比 FCS 与所收到数据的关系来判断所

① 收稿时间:2011-05-18;收到修改稿时间:2011-06-11

传输数据的正确性。最简单的 CRC 编码器使用线性反馈移位寄存器来构成按位产生 CRC 序列的编码电路。由于使用串行处理方法,它只能在一个时钟周期内处理一位数据，这就无法满足高速数据传输时所要求的编码效率。因此，探索一种并行 CRC 校验的实现方式是满足这一要求的有效途径。

并行 CRC 算法一般来说分为硬件和软件两种实现途径。软件途径通常采用查表法（look-up method），而查表法运行速度相对较慢且需要占用较多的存储资源，随着 CRC 校验序列长度的增加，所需的存储器容量成指数级增长。现有的 CRC 硬件并行实现方式基本上都是基于 Z 变换[3]算法、基于状态转换方程（STE）的线性反馈移位寄存器（LFSR）[4]电路结构以及有限域方法[5]。目前，对 CRC 算法并行性的研究以 STE 中的 w-step 过渡方程为基础来实现。但这种方式常常限制了计算的并行度使得少于或等于 CRC 算法多项式的项数，而其它的实现方法同样无法更有效地满足高度并行的需求。随着当前 Gbits 级别的数据传输协议的涌现，我们急需研究开发更加高速的 CRC 校验设备满足传输需求。

2 核心算法简介

CRC 算法传统的实现方法都是将输入的信息以移位的方式输入到线性反馈移位寄存器的最高有效位（MSB）中。再由线性反馈移位寄存器（LFSR）在 Galois 多项式模式下进行按位的乘法来运算。该部分通过反馈、移位处理后所得到的结果就是所需的 CRC 值。但这种串行移位运算的实现方法并不适合高速应用。本设计充分发挥伽罗瓦域（Galois Fields）[6]的特性来实现 CRC 高速应用，在伽罗瓦多项式的选择上则基于标准 ATM 和以太网的原则，具体表达式如下：

$$G(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$$

也可以将多项式视为一个 32 位的向量 GF(232)：

B’10000010011000001000111011011 0111

这个多项式有很多优势，不可约且拥有伽罗瓦域的所有特性。它使用伽罗瓦域的主要特性[6]：$a^{i}\otimes a^{j}=a^{i+j \bmod (2^{32}-1)}$、按字移位（每字长 16bit）、甚至输入数据也是伽罗瓦域的运算元素，这样就可以将 CRC 的计算表达为：

$$CRC(N+1)=CRC(N)\otimes a^{16}\oplus Word(N+1)$$

其中，CRC（N）是上一次运算的结果，a16 是 GF(232)的一个向量，“Word”则是输入的 16bit 字。$\otimes$ 符号表示一个多项式乘法，而 $\oplus$ 符号表示一个伽罗瓦域的加法。

3 系统设计与实现

CRC 算法单个周期的电路结构图如图 1 所示。

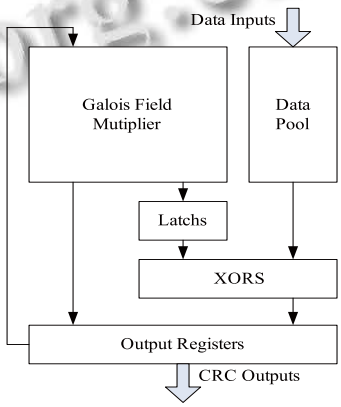

图 1 单个 CRC 单元结构图

其中最重要的部分是伽罗瓦域乘法（GFM）电路部分，其作用是用来实现 $CRC(N)\otimes a^{16}$ 运算。这个通常被称为“H 矩阵”的部分是伽罗瓦多项式乘法数学描述的核心，也是整个 CRC 单元的核心，如图 2 所示。这个矩阵是用来进行 a16 项的乘法的，利用 Galois 域的优秀特性可以简单地使用组合逻辑来实现这个矩阵乘法，所需的只是增加一些 XOR 型电路。从而我们将产生 CRC 序列的问题从计算除法的余数问题转化为一个使用了伽罗瓦域特性的组合逻辑电路来处理。这个矩阵的元素只有 1 和 0 两种，1 所在的位置标识了一个 XOR 型门电路，而 0 所在的位置则是直接用导线连通。如果我们把计算产生的结果表示为一个 32bit 的多元向量，那么这个乘法器的结构符合如下表达式：

$$CRC_{(T+1)}[31]=CRC_{(T)}[31]\oplus CRC_{(T)}[27]\oplus CRC_{(T)}[25]\oplus CRC_{(T)}[24]\oplus CRC_{(T)}[21]\oplus CRC_{(T)}[15]$$

我们可以很容易使用这个格式扩展整个矩阵，进而完成该部件的实现硬件。一旦乘法结束，算法就只剩下“”的部分了，该部分也可由一个 XOR 型门电路阵

列构成。最后的结果有一个输出寄存器保存用于完成反馈操作。为了符合以太网规范，该寄存器的初始值被设置为 32H ' 46AF6449。

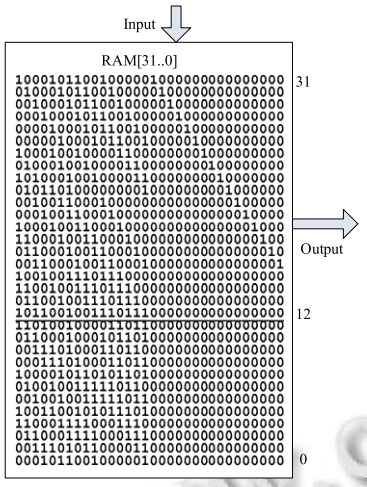

图 2　H 矩阵

3.1 流水线结构

等待方程计算完成将导致关键路径长度随乘法的复杂度而迅速增长，我们可以看到该部分逻辑的最大深度达到 10 个 XOR 门（$CRC_{(T+1)}[12]$）。这样的关键路径长度对于 10Gbps 的速率要求来说实在是太长了。解决的办法是使用一种高度流水线架构使得关键路径减少到 1 个 XOR 长度。GF 乘法流水线设计的方案如图 3 所示。

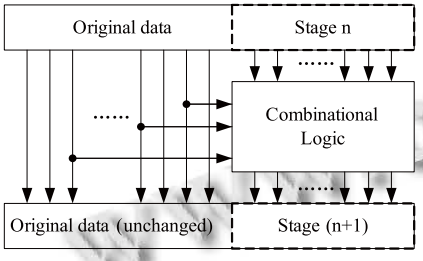

图 3　乘法部件的流水设计

3.2 时序约束

乘法部件流水线所需的寄存器组的读写将增加额外的延迟，但这些寄存器是保证流水计算正确性所必须的。我们引入同一时钟周期内的两相逻辑（φ_1 、φ_2）来优化流水线。两相逻辑使得轮次控制电路及其它与之相关的部分也必须精心设计，以避免数据延迟带来的偏相问题。整个设计的时序逻辑如图 4 所示。

其中，输出寄存器的数据输入和输出是该部分设计的重点。因为输入寄存器的输出块与伽罗瓦域乘法的输出块必须在时钟的同一个相位进行操作。在来自反馈的数据到达输入端时，同样的时延约束也存在于输出寄存器的输出端与输入寄存器的输入端之间。而空闲的寄存器将会被延迟数个时钟周期，这个周期数等同于空闲寄存器的个数，用来保持数据的同步。

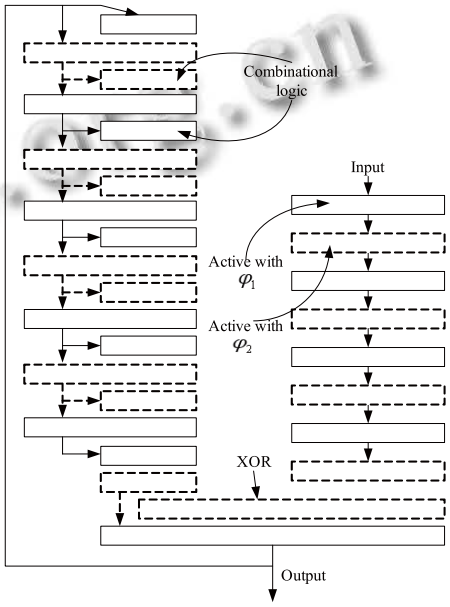

图 4　两相逻辑时序

3.3 优化设计

单纯流水线的解决方案有一定的缺点，如潜伏等待时间和“IDEL”状态周期。这些等待周期来自流水线的反馈：乘法需要 4 个时钟周期来进行，而结果要与输入数据进行 XOR 操作产生反馈数据，在超过 4 个时钟周期之后，新的结果才能为新一次与输入的 XOR 操作做好准备。 不过，该数据流的形成可以视为在一个私有的通道中完成，而不与邻近的通道产生数据的交换[7]。因此,我们可以使用多通道的方式来提高性能[8]。

为了达到 10Gbit 网络的传输要求，我们增加了 4 个通道。每个通道的输出寄存器的值必须等于第一个输入字，因为我们设计的 CRC 单元必须对“TDMA 标准 CRC 单元”保持透明。这个问题可以简单地通过加载乘法单元寄存器中非零值地方法来处理。

流水寄存器的输入端在重置时会被设置为默认，初始值设置方式如下：伽罗瓦域乘法单元的第一流水

线的首个寄存器的最高的 32 位值为 H’E3ED5B2A[9]。这个值对应着对输出寄存器的初始化值(H ' 46AF6449) ，如表 1 所示。

表 1 伽罗瓦域相关寄存器复位初始值[10]

Registers	Values[31..0](HEX)
GF – Register 2	E3ED5B2A
GF – Register 3	CEAD1918
GF – Register 4	90903DD8
GF – Register 5	74EBF27F
GF – Register 6	462A4987
GF – Register 7	46AFBDFF
GF – Register 8	46AF747D
GF – Register 9	46AF7449

在 100MHz 的仿真时钟下，其仿真波形图如图 5 所示。最终该含有所设计高速 CRC 校验单元的 IP 核在功能和时序上都保证了正确性(结果与软件方式计算的 Hash 和 CRC 运算结果一致[11])并达到了预期的性能期望。而在波形图中的延迟与毛刺也正确的反映了电路特征与传输延迟。

4 综合与仿真测试

本设计使用 VerilogHDL 硬件设计语言作为多通道高速 CRC 校验系统的设计语言对该功能单元进行了 RTL 级描述，并以 Altera 公司的 EP2C35F672C6 为目标芯片，在 Quartus II 8.0 平台上进行了综合，布局，布线，利用 ModelSim SE 6.0 进行了系统级的综合仿真，最后结合 SOPC 组件和 NIOS IDE 软件[7-8]环境在 DE2 实验板上进行了下载验证。

表 2 CRC 校验单元的综合结果

项目名称	综合结果	占用器件比
Total logic elements	4065	13%
Logic registers	3588	10%
Dedicated logic register	1540	4%
Total PLLs	0	0%
Total memory bits	0	0%

5 结论

本论文给出了一个高速 CRC 的改进算法，主要分析了其中与编码速率相关的部分。然后提出了一个基于 Galois 域的多通道并行 CRC 编码结构，并给出了相应的硬件设计。与其它结构进行的对比可见，该硬件实现方法有更佳的时序特征，最终基于 SOPC 技术生成的 CRC 校验系统的吞吐量可以满足 Gbits 级别的高速网络。

图 5 CRC 校验单元的仿真波形图

参考文献

1 Pretzel O. Error-Correcting Codes and Finete Fields.Oxford University Press,1992.

2 Pless V. Introduction to the Theory of Error-Correcting Codes. 3rd Ed. The Wiley-Interscience Press, 1998.

3 Albertengo G, Sisto R. Parallel CRC generation. IEEE Micro, 1990,10(5):63−71.

4 Derby JH. High-speed CRC computation using state-space transformations. Proc. of GLOBECOM’01,San Antonio, TX, USA, 2001. 166−170.

5 Glaise RJ, Jacquart X. Fast CRC calculation. Proc. of the 1993 IEEE International Conference on Computer Design: VLSI in Computers & Processors, Cambridge, MA, USA, 1993, 602−605.

6 Glaise RJ. A two-step computation of cyclic redundancy code CRC−32 for ATM networks. IBM Journal of Research and Development,1997,41(6):705−709.

7 陆建松,李明,李纪军,刘玲.EPON 中 CRC 校验码的并行算法实现.无线通信技术,2006,1.

8 Hobson RF, Cheung KL. A High Performance CMOS 32-bit Parallel CRC Engine. IEEE Journal of Solid-State circuits,1999,9(40).

9 彭建辉.10G 以太网接口并行 CRC 校验的一种简化算法.微计算机信息,2011,19.

10 毕占坤,张羿猛,黄芝平,王跃科.基于逻辑设计的高速 CRC 并行算法研究及其 FPGA 实现仪器仪表学报,2007,3(12).

11 Jiang H, Li ZY. FPGA design flow based on a variety of EDA tools, Micro Computer Information, 2007(23)11−2: 201−20.