

基于 GPU 的 RS 译码处理技术研究

顾久祥^{1,2}, 杨仁忠¹, 韦宏卫¹

(1 中国科学院 对地观测与数字地球科学中心, 北京 100094; 2 中国科学院研究生院, 北京 100039)

摘 要: 在分析高速译码处理技术的基础上, 提出了基于 GPU 平台 RS 译码的解决方案, 并基于 CUDA 编程环境完成了 RS 译码处理的研究与实现. 经测试表明, 本文基于 GPU 的 RS 译码处理模块在理论纠错范围内的处理速度达 900Mbps 以上, 满足实际遥感信道的译码要求.

关键词: Reed-Solomon; 译码器; GPU; CUDA; 并行计算;

中图分类号: TN919

文献标识码: A

文章编号: 1000-7180(2013)04-0119-04

Research of RS Decoding Technology Based on GPU

GU Jiu-xiang^{1,2}, YANG Ren-zhong¹, WEI Hong-wei²

(1 Center for Earth Observation and Digital Earth, Chinese Academy of Science, Beijing 100094, China;

2 Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: In this paper, A RS decoding solution based on GPU platform is proposed, we investigated decoding implementation of RS codes on CUDA. Experimental results show that the processing speed is more than 600Mbps, which can achieve throughput requirement of the remote sensing channel decoding requirements.

Key words: Reed-Solomon; decoder; GPU; CUDA; Parallel computing

1 引言

RS 码具有很强的纠错能力, 特别适用于在衰落信道中用来克服突发性差错. 目前在深空通信、军事通信、多媒体数据通信等领域获得广泛应用^[1]. CCSDS 在 NASA 对 RS 码的研究成果的基础上, 将 RS 码确定为分包遥测信道编码的标准之一^[2]. 随着遥感卫星技术迅速发展, 卫星下行码速率也越来越高, 对地面接收系统提出了更高的要求, 高速 RS 译码处理已经成为人们关注的焦点. 在解决高速 RS 译码的技术方案中, 采用纯软件方式很难达到实时处理的要求, 目前主要是硬件实现 (DSP 或 FPGA).

随着集成工艺与集成度的提高, 图像处理器 (GPU) 的处理能力得到了飞速发展. 本文提出了一种基于 GPU 的 RS 译码处理方案. 该方案充分利用了 GPU 强大的并行计算能力, 在 CUDA 编程环境

下得到了很好的加速比, 可以满足高分辨率遥感卫星下行数据的译码处理要求. 同时, 该方案可以针对不同码长的 RS 码修改配置参数进行译码, 具有良好的适应性.

2 RS 码及其译码算法

2.1 RS 码介绍

Reed-Solomon 码 (RS) 是一类具有很强纠错能力的多进制 BCH 码, 也是一类典型的代数几何码. RS(n, k) 码的基本含义是: 前 k 个符号表示信息码, 后 n-k 个符号是用生成多项式产生的冗余监督码, 如图 1 所示.

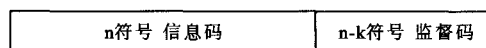


图 1 RS 码的组成结构

以 RS(255, 239) 为例说明, $m=8$, 表示每个符号的比特数; $n=255$, 是每个 RS 码字的符号数; $k=$

收稿日期: 2012-07-03; 修回日期: 2012-08-16

基金项目: 高分辨率对地观测系统 (2010-1481)

239, 表示 RS 码字中的信息码的符号数; $t=8$, 代表其纠错能力最大为 8 个符号。

RS 码作为一种非二进制循环码, 每个码元均取自伽罗华域 $GF(2^m)$, m 是大于 2 的任意正整数。RS 编译码过程中所有运算都是基于伽罗华域进行的, 所以它的运算特性已经不再具备二元域的某些特性。RS 译码过程中不仅要找到误码的位置, 而且还要计算出相应错误位置的错误值, 因此增加了译码的难度。目前, RS 译码已经有很多种成熟的算法, 主要有时域译码和频域译码。时域译码通常优于频域译码, 所以一般采用的较多。

2.2 RS 时域译码算法

RS 码作为分组码, 其传统译码过程主要分为 5 步: 伴随式计算、关键方程求解、钱氏搜索计算错误位置、Forney 算法计算错误值、纠错输出。

假设发送的码字为 $C(x) = \sum_{i=0}^{n-1} c_i x^i$, 接收的码字为 $R(x) = \sum_{i=0}^{n-1} r_i x^i$, 错误图样为 $E(x) = \sum_{i=0}^{n-1} e_i x^i$, 则有: $R(x) = C(x) + E(x)$, 所以 RS 译码的最重要的是求得错误位置和错误值。

对于 $RS(n, k, 2t)$ 的码型来说, 其纠错能力为 t , 每个码字的伴随式中包含了 $2t$ 个伴随式分量。计算伴随多项式的系数的公式为: $S_i = R(\alpha^i) = r_{n-1}(\alpha^i)^{n-1} + r_{n-2}(\alpha^i)^{n-2} + \dots + r_0 = \sum_{j=0}^{n-1} \alpha^{ij} r_j$ 。其中, α 是 $GF(2^m)$ 域上的本原元, $i = 0, 1, \dots, 2t-1$ 。

在得到 $2t$ 个伴随式后, 如果计算出的伴随式系数 S_i 不为零, 则认为接收码字中存在错误。当码字中统计的误码个数小于等于 t 时, 表明译码算法可对接收的码字进行正确译码, 然后用 BM 算法解关键方程得到错误位置多项式, 再用钱氏搜索得到错误位置。在确定误码位置后, 用 Forney 算法求对应错误位置上的错误值, 计算错误值并纠正相应错误。当码字中统计的误码个数大于 t 时, 表明超出译码能力, 无法进行纠错, 结束译码过程。译码基本处理流程如图 2 所示。

3 基于 GPU 的 RS 译码实现方案

3.1 CUDA 编程模型简介

CUDA 是 NVIDIA 公司为在 GPU 上实现编程而提供的基于 C 的编程环境, 它是一个完整的 GPGPU 解决方案, 提供了硬件的直接访问接口, 而不必像传统方式一样必须依赖图形 API 接口来实

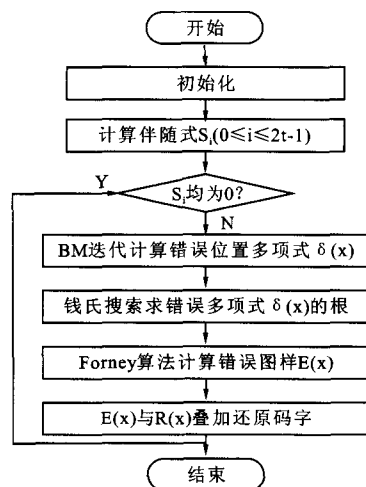


图 2 RS 译码基本流程图

现 GPU 的访问。在架构上采用了一种全新的计算体系结构来使用 GPU 提供的硬件资源, 从而给大规模的数据计算应用提供了一种比 CPU 更加强大的计算能力。图 3 是一个典型的 CUDA 编程模型, 主要分为两个部分: Host 端和 Device 端。在 CUDA 编程模型中, CPU 与 GPU 协同工作, 主程序部分还是在主机端进行执行, GPU 则专注于高度线程化的并行处理任务。每当遇到数据并行处理的部分时, CUDA 就会将程序编译成 GPU 能执行的程序传送到 GPU 上运行。运行在 GPU 上的 CUDA 并行计算函数称为内核函数 (kernel), 一个完整的 CUDA 程序是由一系列的主机端串行步骤和设备端并行步骤共同构成的。

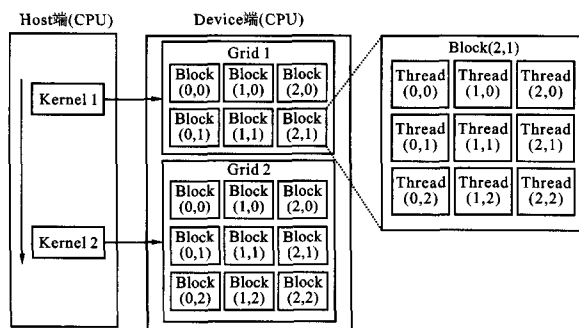


图 3 CUDA 编程模型的 Host+Device 结构

CUDA 将计算任务映射为大量的可以并行执行的线程, 并由硬件动态调度和执行这些线程, 属于同一个线程块的线程不仅可以并行执行, 而且可以通过共享内存和栅栏同步实现线程块内的线程间通信。这样, 同一 grid 内的不同线程块之间形成了不需要通信的粗粒度并行, 同一 block 内的线程之间又形成了允许通信的细粒度并行。这就是 CUDA 的关键特性: 线程按照两个层次进行组织、在较低的层

次通过共享存储器和栅栏同步实现通信^[3]。

3.2 RS译码处理的GPU实现

在编写CUDA程序时,需要进行主机端和设备端的数据交互,因此在对设备端显存访问时需要注意内存融合(Memory Coalescing)。SM对线程的调度是以一个Warp(32个线程)为单位进行,而访问显存是以Half warp(16个线程)为单位进行,当访问的起始地址符合16个线程共访问的数据量的整数倍,且它们在同一片连续的地址空间访问数据时,则整个16个线程对数据的访问可以作为一个Transaction进行,这样能最大限度的利用显存的带宽,即内存融合。

在RS译码过程中,加、减、乘和除都是在伽罗华域进行的,加减法运算可以通过模二运算来完成,除法也可以转换成乘法来实现。本文中乘法运算采用了二表法,二表法通过建立两个数据存储表——元素表和指数表来进行运算转换。元素表用于存储根据本原多项式得到的 α^i 所对应的元素值;指数表是根据 α^i 来查找其对应的指数。由于元素和指数是一一对应,而且指数和元素的分布是连续的,因此可以直接采用连续的元素和指数分别作为指数表和元素表的偏移地址,从而在查找时可以快速定位。在计算能力为1.0/1.1/1.2/1.3的GPU中,同一个block中shared memory可以被该block中所有的线程访问,且其访问速度几乎和访问寄存器一样。由于二表法只涉及1次算术加、1次模运算以及三次查表操作,所以在计算过程中将元素表和指数表存放在shared memory中,对译码处理速度有很大提高。

对RS(n,k)码来说,其最大纠错能力为t,而在实际的通信中码元错误概率 $p_e \leq 1/q$,很容易产生存在0个错误和1个错误的错误图样。假设码元错误概率为5%,若将数据按31个码元为一组发送,则接收码组中,约20%没有出错。约33%只有1个码元出错。图4是对不同码型发生0、1个错误情况下错误概率对比(每个码元误码率5%),可以看出这两种情况占的比例很大,所以要设法减少上述两种情况下的译码计算量。

由文献[4]可知,若RS码字中错误个数不超出该码型的最大纠错能力t时,只要 $S_1, S_2, \dots, S_t$ 全为0,就可以确定码字中没有错误。而当 $S_1, S_2, \dots, S_{t+1}$ 满足 $S_{t+1}/S_t = S_t/S_{t-1} \neq 0$,也可以确定码字中只有1个错误发生。这时根据 $S_2/S_1 = \alpha^j, S_1 = e_j \alpha^j$ 可计算出错误位置j和对应的错误值 e_j ,并实现纠

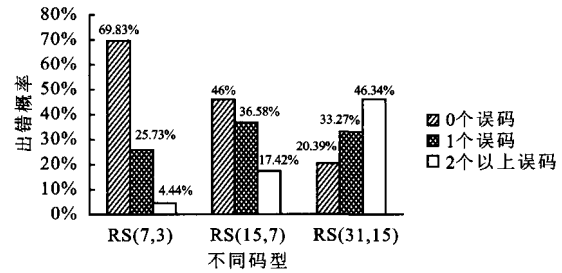


图4 不同码型每个码字发生0、1个错误的概率

错。否则,表明该码字的错误大于等于2个,这时需要计算剩下所有的伴随式分量,然后按照传统的RS译码算法求解错误位置多项式得到错误位置以及相应错误值,并对接收的码字进行纠错^[4]。从上图的数据可以看出,对出现一个错误的情况下单独处理,可以避免了解关键方程以及后续处理的复杂运算,大大降低了计算量。

根据以上提出的改进译码算法,在计算伴随式时采用了Horner算法。在码字中不出现错误时,传统算法该步骤需要计算 $2t$ 个伴随分量,其计算量为 $2t(n-1)$ 次乘法和加法,改进的方法计算量为 $t(n-1)$ 次乘法和加法。在码字中只有一个错误时,传统算法该步骤需要计算 $2t$ 个伴随式分量,计算量约为 $(2t+1) \cdot n$ 次乘法和加法,改进算法只计算 $t+1$ 个伴随式分量,其计算量约为 $(t+1)n$ 次乘法和 $(t+1)(n-1)$ 加法。本文考虑到在发生2个错误以上情况下需要另行计算剩余的伴随式分量,会带来较多的分支运算。而GPU的优势是无逻辑关系数据的并行计算,过多的分支会严重影响GPU的运算能力,所以在实现时没有根据误码情况来计算部分的伴随式分量,而是一次发起多个线程计算该码字全部 $2t$ 个伴随式分量,降低了GPU与CPU之间的交互操作。图5是RS译码的基本处理流程图。

4 结束语

本文提出了一种基于GPU的RS译码处理方案,并基于CUDA编程环境完成了RS译码处理模块的设计与实现。经过测试表明,在保证系统可靠译码与纠错处理的前提下,本文实现的RS译码模块的速度达900Mbps以上,满足高分辨率遥感卫星下行数据的译码需求。采用基于GPU的RS译码处理技术方案,与基于CPU处理器的软件处理方案相比,得到6倍以上的加速比,与DSP、FPGA实现方案相比,实现更容易更快捷。值得说明的是,采用本文技术方案需合理地选取数据块尺寸,有利于达到最佳RS译码处理性能。本方案仅需修改少量参数

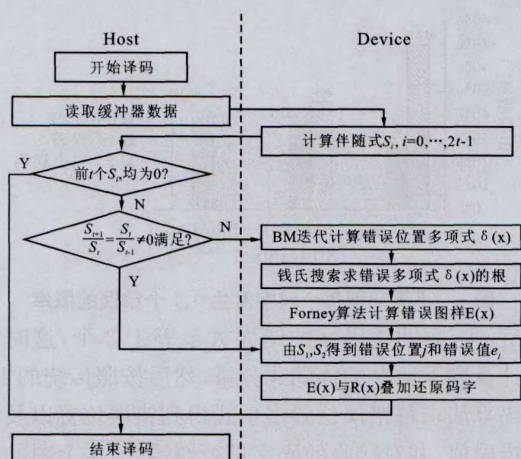


图5 RS译码 CUDA 处理流程

即可实现其它 RS 码的译码处理功能,因此,本文研究具有一定的技术推广意义及工程实用价值。

参考文献:

- [1] 王新梅, 肖国镇. 纠错码——原理与方法[M]. (修订版). 西安: 西安电子科技大学出版社, 2006.
- [2] CCSDS. Recommendation for space data system standards telemetry channel coding[S]. Blue Book, 2001.
- [3] 张舒, 褚艳利. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009.
- [4] 郝永杰, 蒋建国. 改进的 RS 时域译码算法[J]. 计算机工程, 2008, 34(14): 104-106.

作者简介:



顾久祥 男, (1988—), 硕士研究生. 研究方向为遥感卫星数据获取与处理.



杨仁忠 男, (1952—), 研究员. 研究方向为信号获取与处理技术、地面接收系统工程技术、实时数据处理.



韦宏卫 男, (1981—), 助理研究员. 研究方向为地面接收系统工程技术、高速数字系统设计.