

# 一种支持多种工作模式的可重构计算单元的设计

李浩洋, 宋宇鲲

(合肥工业大学 电子科学与应用物理学院, 安徽 合肥 230009)

**摘 要:** 针对高密度计算领域的的数据特点, 为目标多核计算系统设计了一种可重构计算单元, 其通过内部运算部件的运算路径重构, 可以实现多种乘法运算功能, 并对高密度计算领域中常见的矩阵乘法、快速傅里叶变换等算法进行了针对性的加速设计, 并能够支持多种工作模式. 该设计已经在 Xilinx Virtex-7 XC7V2000TFLG1925-1 FPGA 芯片上进行了原型验证, 测试结果表明, 对高密度计算领域的典型数据粒度, 该可重构计算单元的计算效率满足设计要求.

**关键词:** 可重构; 计算单元; 流运算; 存储运算; 硬件加速

**中图分类号:** TN 47

**文献标识码:** A

**文章编号:** 1000—7180(2017)10—0053—06

## Design of a Reconfigurable Computing Unit With Multiple Working Modes

LI Hao-yang, SONG Yu-kun

(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China)

**Abstract:** In this paper, a coarse-grained reconfigurable computing unit (RCU) is designed for the heterogeneous multi-core system (HMCS) which is applied to high density computing field. As a reconfigurable computing unit, RCU can achieve a variety of functions and work in different modes. The architecture of RCU is also optimized for some common complex algorithms, such as matrix calculation, FFT and etc. At last, the hardware prototype of RCU is implemented on the FPGA and the experimental results show that the performance has achieved the performance requirements.

**Key words:** reconfigurable; computing unit; stream computing; storage computing; hardware acceleration

### 1 引言

以数据密集和计算密集为典型特征的高密度计算应用, 对微处理器的处理能力和处理并行度的要求极高, 且需求仍呈现高速增长态势. 虽然随着微电子工艺的进步, 单块芯片上可以集成的晶体管数目不断增加, 在一定程度上缓解了这一矛盾, 但由于生产成本和功耗的急剧增长, 仍然难以满足市场对微处理器计算性能的需求. 此时, 可重构计算作为一种平衡计算能力与资源消耗的手段应运而生.

可重构计算的定义是: 使用集成了可编程硬件的系统进行计算, 该可编程硬件的功能可由一系列

定时变化的物理可控点来定义<sup>[1]</sup>. 当前, 可重构计算主要通过对特定可重构部件的配置来实现<sup>[2]</sup>, 而可重构部件的可重构性则体现在其内部执行单元的结构可变性和单元间互连线的结构可变性两个方面.

本文为目标多核计算系统设计了一种可重构计算单元, 针对目标系统所应用于的高密度计算领域的算法特点, 采用粗粒度动态可重构技术进行设计, 在保证针对目标领域的足够灵活性的基础上, 实现了资源复用, 提高了片上资源利用率, 实现了资源、效率与灵活性的折中.

随着微电子工艺和计算机技术的快速发展, 可

重构计算的概念被学界广泛接受,并正由学术研究向商业生产方向快速迈进.关于可重构计算,国外学术机构已有众多研究成果<sup>[3-4]</sup>.

国内可重构计算领域的研究也逐步从无到有,从跟随国外相关研究起步,与各个应用研究相结合,并逐步取得了许多研究成果<sup>[5-8]</sup>:

## 2 多核计算系统体系结构

HMCS 是一个面向高密度计算领域的异构多核计算系统,包括片上网络、主控制单元(MC)、计算单元(Cluster)、接口单元(ETH 和 IO)和存储器(MEM),其结构缩比示意图如图 1 所示.

HMCS 的数据交换使用一张二维 Mesh 结构的片上网络,采用包-电路交换协议,回退转向路由策略,能够以最短路径在任意两个路由节点间实现数据传输;主控制单元是 HMCS 的控制核,负责任务下发和算力分配;接口单元和存储器则分别完成计算系统与外界的数据交换和计算系统内数据的存储;计算单元负责计算系统中计算任务的执行,根据目标任务的不同,既可以集成多个相同的计算单元组成同构计算系统,也可以集成多种不同的计算单元组成异构计算系统.本文设计的可重构计算单元,就是针对高密度计算应用设计的一种粗粒度可重构计算单元.

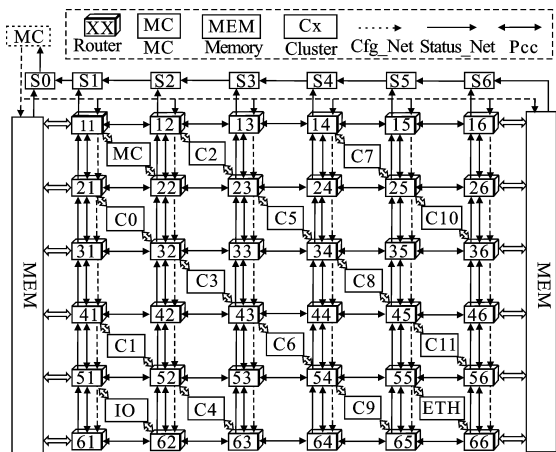


图 1 HMCS 结构缩比示意图

HMCS 系统结构简单、数据传输效率高、可拓展性强,能够实现极大的数据吞吐率和可观的计算核集成度,十分适合高密度计算应用中的大批量数据交换任务,符合未来众核发展趋势,相较于基于总线结构的多核系统,优势明显.

## 3 可重构计算单元的设计

目标计算系统所面向的高密度计算领域主要涉

及浮点数的处理,浮点乘法和浮点加法在该领域中的相关算法中尤为常见.针对这一特点,可重构计算单元内部集成了若干浮点乘法器和浮点加法器,能处理符合 IEEE-754 标准的单精度浮点数的各种乘加类运算,并能够通过总线/开关混合结构改变浮点运算器之间的连接路径,实现运算功能的粗粒度重构,达到资源、效率与灵活性的平衡与折中.特别是,由于高密度计算领域中数据量巨大且构成规则,面向向量和矩阵的处理需求较多,可重构计算单元被设计成粗粒度计算簇,即以数据块为单位而不是以单个数据为单位对数据进行处理.

### 3.1 工作模式介绍

可重构计算单元支持 3 种工作模式:存储计算工作模式、脉动计算工作模式和流计算工作模式,图 2 给出了 3 种工作模式的数据流示意图.

存储计算工作模式中,数据的处理被划分为相互分离的数据输入、计算执行和数据输出三个部分.考虑到可重构计算单元是一个粗粒度计算簇,且簇内数据缓存容量有限,在进行数据处理时,需要根据数据缓存容量的大小对数据进行分片操作,每次只处理一个“数据片”,以保证数据缓存不会溢出.工作中,可重构计算单元等待一个数据片被完全搬运到簇内缓存后才开始计算,并等待针对该数据片的所有计算完成后才将运算结果搬出计算簇.这种工作模式下,源数据被提前本地化,可以根据算法的需求应用多种地址跳变规则,以节省数据整序的时间,还可以通过地址循环实现数据复用,对于矩阵乘法等数据高复用率算法,能够大大减少源数据的请求次数,从而降低对多核计算系统中存储带宽和网络通信的压力.此外,本工作模式下,数据链路建立后不需要挂起链路等待数据,对片上网络的带宽利用率很高.

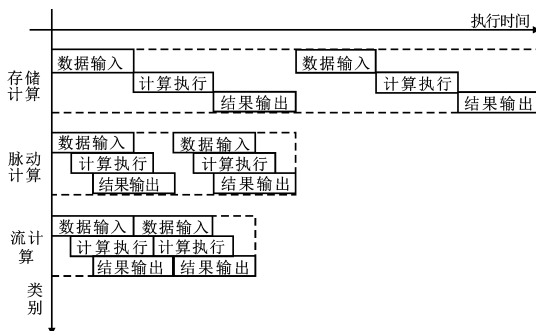


图 2 工作模式示意图

流计算工作模式中,数据以流水的方式通过可重构计算单元并完成相应计算,数据缓存只能顺序

提供数据进行计算,牺牲了一定的灵活性.但是,对于流计算应用,这种工作模式使得流水线的优势被充分发挥,数据的搬运时间与计算时间基本重合,从而达到了隐藏数据搬运时间、增加数据吞吐率的目标,成倍提升了对流计算应用的处理效率.流计算应用中不需要提前将所有数据实现本地化,当然也就不需要对数据进行分片,但为了保证数据安全,流模式下需要增加额外的传输控制信号来实现数据在传输中的流控功能,对片上网络的带宽利用率也不如存储计算工作模式高.

脉动计算工作模式结合了前两种工作模式的特点,其目的是实现灵活性和工作效率的折中.在这种工作模式下,可重构计算单元仍然以流水的方式顺序处理数据,以隐藏数据搬运时间,提升计算吞吐率,但和存储计算工作模式一样对数据进行分片操作,按顺序依次处理“数据片”.这样,脉动计算工作模式下与存储计算工作模式下可重构计算单元的“数据片”大小相当,可以实现对接.由于通过片上网络可以轻易实现簇间路径重构,那么利用存储计算工作模式和脉动计算工作模式数据吞吐率的差异,就可以使一个脉动计算工作模式的<sup>\*</sup>可重构计算单元服务于若干个存储计算工作模式的<sup>\*</sup>可重构计算单元,从而实现算力资源的时分复用,进一步提高算力利用效率.

3.2 可重构计算单元体系结构

如图 3 所示,可重构计算单元包括控制部分、计算部分和存储部分三大部分.下面对各个部分的结构与功能进行详细介绍.

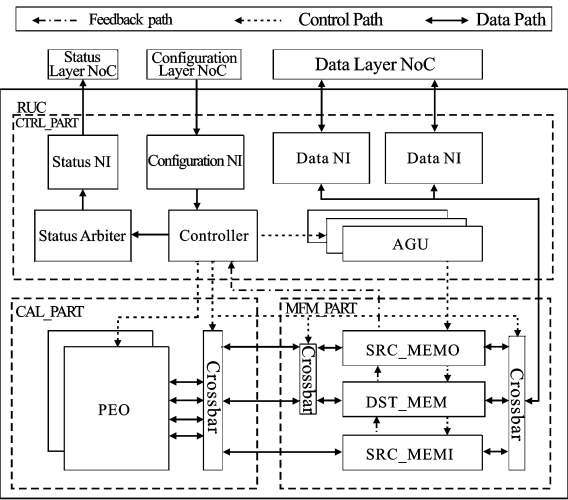


图 3 可重构计算单元结构

控制部分主要包括控制器模块、地址生成器模块和片上网络接口模块.其中,控制器模块是可重构

计算单元的核心控制器,可重构计算单元的数据输入、计算执行和数据输出等都是在该模块的直接或间接控制下实现的;地址生成器模块负责根据具体的计算类型和参数生成相应的存储地址,从而指导存储部分从指定的位置获取运算源数据或向指定的位置存储运算结果数据;片上网络接口模块包括数据层网络接口模块、状态层网络接口模块、配置层网络接口模块和接口仲裁器模块,是可重构计算单元的数据传输接口,用于实现与系统中其他功能单元的通信.

计算部分主要包含两个运算部件(Process Element, PE),每个运算部件中均含有 3 个浮点加法器和 2 个浮点乘法器,在收到配置信息后,浮点加法器和浮点乘法器之间的连接方式根据配置信息实现重构.举例来说,在进行累加运算时,运算部件被重构成浮点累加器,如图 4(a)所示;在执行矩阵乘法运算时,运算部件被重构成浮点乘累加器,如图 4(b)所示;而在执行快速傅里叶变换计算时,运算部件被重构成蝶乘运算单元,如图 4(c)所示,从而实现了不同的运算功能.

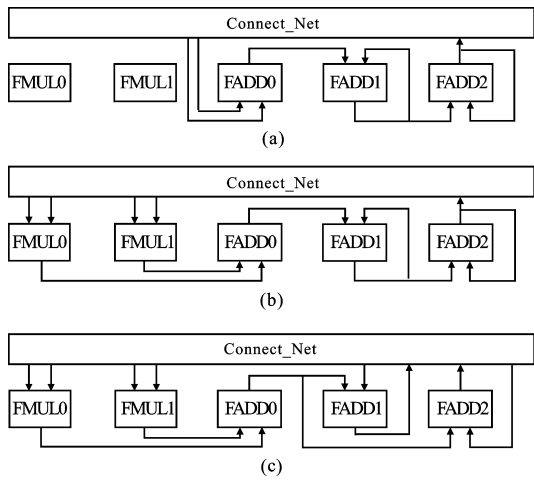


图 4 运算部件重构示例

一般地,在进行复数计算时,规定计算的实数部分和虚数部分分别在 PE0 和 PE1 中完成,而当进行实数计算时,则只有 PE0 接受配置并执行计算.特别的,可重构计算单元也可以实现对实数的 SIMD 计算功能,即 PE0 和 PE1 在一条配置信息的控制下,同时分别处理两路实数运算,从而使实数计算效率成倍获得提升.

存储部分是可重构计算单元的数据缓存,包括存放计算源数据的源数据缓存(SRC\_MEM)和存放运算结果的结果数据缓存(DST\_MEM).根据计算

任务的需要和系统的运行特点,存储部分可以实现 3 种重构,包括类型重构、容量重构和功能重构. 类型重构指存储部分可以根据计算任务的工作模式将缓存重构成 FIFO 或 RAM,即在流计算或脉动计算工作模式时,数据缓存工作在 FIFO 模式,主要用于缓解由于存储和网络带宽导致的数据延时;而在存储计算工作模式时,数据缓存工作在 RAM 模式,可以根据地址生成器提供的地址提供数据,并实现数据重用. 容量重构指在数据缓存总容量一定的情况下,存储部分可以根据所执行运算的目数将缓存重构成若干个不同容量的缓存,从而最大限度地提升对存储部分的利用率. 功能重构则是指在运算中,如果数据需要反复迭代运算,则可以通过动态重构将存储部分中的源数据缓存和结果数据缓存功能调换,从而避免了数据的搬运和额外功耗开销.

3.3 常见典型算法优化

在高密度计算领域,特别是图像处理等应用中,部分算法由于其算法自身特殊性和出现的高频率,经常成为算法映射时的算法瓶颈,拖慢整个系统的数据处理速度.

权衡算法优化带来的性能提升与付出的结构改造成本,可重构计算单元中对部分常见典型算法(如 FFT 算法、FIR 滤波、矩阵乘法运算)的处理过程进行了优化,从而发挥多核计算系统面向特定领域具有通用性的优势,提升总体计算效率. 本文以 FFT 算法为例,简要介绍可重构计算单元中的部分优化方法.

表 1 FFT 结果数据地址存储规则

| 存储器编号 |     |     |       |       | 地址规则  |       |       |       |       |
|-------|-----|-----|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0   | N/4 | 1     | N/4+1 | 2     | N/4+2 | ..... | N/4-1 | N/2-1 |
| 1     | N/4 | 0   | N/4+1 | 1     | N/4+2 | 2     | ..... | N/2-1 | N/4-1 |

4 实验与性能测试

本节对 HMCS 中集成的可重构计算单元进行计算性能测试. 实验将分为两部分:第一部分中,在 HMCS 中只集成一个可重构计算单元,进行较小数据批量的基础运算测试,以测试单个可重构计算单元的工作性能;第二部分中,在 HMCS 中集成多个可重构计算单元,进行计算加速比测试. HMCS 使用 Verilog HDL 语言编写,并在 Xilinx Virtex-7 XC7V2000TFLG1925-1 FPGA 芯片上实现,系统工作在 100 MHz 频率下,可重构计算单元的资源消耗情况如表 2 所示.

FFT 算法是图像处理中的一种常用经典算法,能够实现频域和时域的转换,计算方法如式(1)所示.

$$\begin{cases} PE1: & D_{r1} = A_r + (B_r * C_r - B_i * C_i) \\ & D_{r2} = A_r - (B_r * C_r - B_i * C_i) \\ PE2: & D_{i1} = A_i + (B_r * C_i - B_i * C_r) \\ & D_{i2} = A_i - (B_r * C_i - B_i * C_r) \end{cases} \quad (1)$$

可重构计算单元中,快速傅里叶算法采用倒序输入、顺序输出方式,并采用经典的单蝶乘单元基-2 算法实现. 首先,可重构计算单元中增加了数据倒序逻辑,源数据通过数据层网络接口存入源数据缓存时,即可同时完成数据的倒序过程;其次,运算部件由 4 个乘法器和 6 个加法器构成,恰好可以被重构成一个复数蝶乘单元;再次,快速傅里叶变换共包含 3 个复数源操作数和 2 个复数目的操作数,故存储部分被重构成 3 个源数据缓存和 2 个运算结果缓存;此外,考虑到快速傅里叶变换在计算执行时需要多次进行迭代,可重构计算单元的存储部分会在每一次计算完成后进行动态功能重构,使源数据缓存和结果数据缓存的功能实现调换,从而能够继续进行下一级迭代计算,避免了采用级联方式导致的过多资源消耗、片上网络占用和数据搬运延时;最后,为了使运算和数据存储规则化,在可重构计算单元的设计中还专门为快速傅里叶变换集成了无冲突运算结果地址规则,如表 1 所示,实现了各次迭代运算间的数据快速整序.

表 2 可重构计算单元资源消耗

| Resource      | FF    | LUT   | BRAM |
|---------------|-------|-------|------|
| Utilization   | 6 406 | 8 865 | 12   |
| Utilization % | 0.26  | 0.73  | 0.93 |

4.1 可重构计算单元测试

在本部分中,主要对可重构计算单元的基本运算能力进行测试. 首先,我们选择了基本的向量运算(向量加法、向量乘法和累加运算),进行了不同数据量下的基准计算测试,以分析可重构计算单元的计算性能;然后,以数据重用率较高的复数矩阵乘法运算为例,详细分析了可重构计算单元在不同工作模式下的性能区别;最后,以快速傅里叶变换算法为

例,测试了采用优化加速设计后,典型常见算法的计算性能.

图 5 中给出了系统空载时可重构计算单元在不同工作模式下对不同数据量的运算(加法、减法、乘法、累加)执行计算时消耗的时钟周期数的平均值.

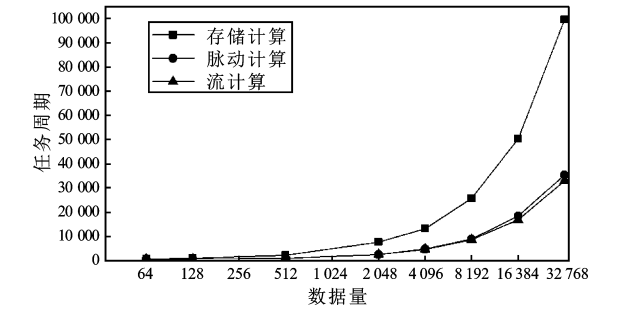


图 5 系统空载时各工作模式下数据量-运算周期数关系图

在数据量小于 1 k 时,三种工作模式消耗的时钟周期数相似,任务下发和存储与网络延时占主导地位,导致总时钟周期差异不大.当数据量超过 2 k 时,存储计算工作模式下消耗的时钟周期数量明显高于脉动计算工作模式和流计算工作模式,这是由于存储计算工作模式下,数据搬运和计算执行时间不重合,数据搬运时间无法隐藏,可重构计算单元在数据搬运上耗费大量时间,甚至超过了实际用于计算的时间.对比流计算工作模式和脉动计算工作模式,在数据量超过 8 k 并继续增加时,流计算工作模式和脉动计算工作模式的时钟周期数逐渐拉开,这些时钟周期差异主要是由脉动计算工作模式需要多次建立和撤销数据链路以及进入和退出流水线造成的,整体上对性能影响有限.

但是,当系统中的片上网络比较繁忙时,片上网络数据链路建立成功率大大降低时,平均包延时增加,会使额外开销急剧升高.图 6 给出了中等负载(平均包延时为 1.5 时钟周期)时,可重构计算单元在不同工作模式下对不同数据量的运算执行计算时消耗的时钟周期数的平均值.此时,脉动计算工作模式和流计算工作模式之间已经出现了比较明显的差距,但计算时间仍然明显低于存储计算工作模式.

综上,对于普通的顺序计算任务,在存储和网络带宽足够的情况下,应尽量避免存储计算工作模式的使用;当数据量极大或网络性能不佳时,优先考虑采用流计算工作模式.

表 3 给出了可重构计算单元分别采用流计算工作模式和存储计算工作模式执行矩阵乘法( $A \times B$ )计算时的情况.

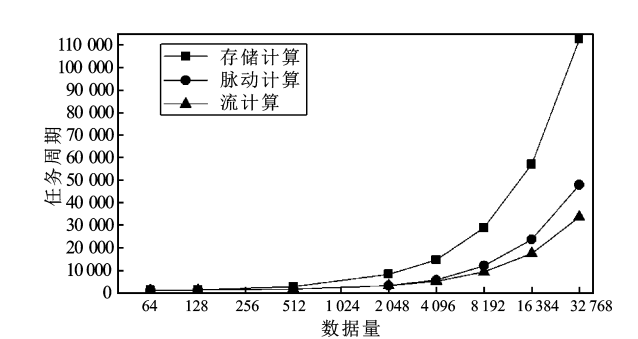


图 6 系统中等负载时各工作模式下数据量-运算周期数关系图

表 3 矩阵乘法运算性能表

| 矩阵规模                                     | 工作模式 | 计算周期数     | 任务周期数     | 计算周期比率  |
|------------------------------------------|------|-----------|-----------|---------|
| A: $8 \times 16$<br>B: $16 \times 8$     | 流    | 1 084     | 1 266     | 0.856 2 |
|                                          | 存储   | 1 084     | 1 560     | 0.694 9 |
| A: $32 \times 64$<br>B: $64 \times 32$   | 流    | 65 596    | 65 796    | 0.997 0 |
|                                          | 存储   | 65 596    | 68 245    | 0.961 2 |
| A: $32 \times 256$<br>B: $256 \times 32$ | 流    | 262 204   | 262 421   | 0.999 2 |
|                                          | 存储   | 263 104   | 296 674   | 0.886 8 |
| A: $32 \times 2k$<br>B: $2k \times 32$   | 流    | 2 097 212 | 2 097 458 | 0.999 9 |
|                                          | 存储   | 2 158 592 | 4 256 483 | 0.507 1 |

从表中数据可以看出,在执行矩阵乘法计算时,流计算工作模式在任务周期数和计算周期比率上仍然优于存储计算工作模式,但领先幅度相对于前一实验并不明显.分析其原因,是因为矩阵乘法计算中数据重用率较高,存储计算工作模式下,一次数据输入可以使用多次,从而降低了数据传输时间在整个任务时间中所占的比例,一定程度上缓解了性能下降.

在流计算工作模式下,任务下发时间、链路建立时间、存储器和网络延时等固定开销几乎不变,而随着矩阵规模增长,计算周期数迅速增加,从而使计算周期数在任务周期数中所占的比例升高,并在矩阵规模足够大时近似等于任务周期数,也就使得实验中计算周期比率随着矩阵规模的增长也迅速升高,并趋近于 1;存储计算工作模式下,计算周期比率随着矩阵规模的增长先升高后降低,其原因则是在矩阵规模达到一定程度后,可重构计算单元中的数据缓存容量不足以容纳足够多的矩阵向量,从而导致数据重用率下降,数据传输占用的时钟周期数再次成为了影响任务周期数的关键因素.

当然,有两个问题仍然值得关注,首先,流计算工作模式下,需要增加额外的数据整序步骤(如矩阵转置等),往往也需要花费可观的时钟周期数,而存

储计算工作模式下这一工作可以在可重构计算单元内部完成;其次,存储计算工作模式下,特别是矩阵规模较小时,可重构计算单元可以在计算簇内部实现完全数据重用,从而大大降低对计算系统中存储带宽和网络传输性能的压力,在计算系统处在高负载状态时更具有优势.

综上,在矩阵规模较小时,应该优先选择存储计算工作模式;而当矩阵规模极大,远超过可重构计算单元数据缓存容量时,流计算工作模式效率更高.

FFT 性能测试证明了本文设计的可重构计算单元.表 4 给出了可重构计算单元执行不同点数和不同数据量的 FFT 计算时的部分实验结果.由于 FFT 计算中不是顺序读取数据,只能采用存储计算工作模式.从表中可以看出,FFT 点数相同时,随着单次数据量的增加,计算周期比率逐渐上升,其原因是存储和网络延时相对固定,数据量增加时,分摊到单个数据的时钟周期数下降;数据量一定时,FFT 点数越大,计算周期比率越小,主要是因为 FFT 点数增加时迭代次数增加,从而在数据传输时间相似的条件下增加了计算周期数,使计算周期数占比上升.通过结构优化,单个可重构计算单元即可完成 2k 点以内的  $2^N$  点 FFT 的计算,且计算效率接近单蝶乘单元基-2 算法的理论值,大大提升了多核计算系统对 FFT 算法的计算效率.

表 4 FFT 计算性能表

| FFT 点数 | 数据量 | 计算<br>周期数 | 任务<br>周期数 | 计算周<br>期比率 |
|--------|-----|-----------|-----------|------------|
| 32     | 32  | 175       | 417       | 0.419 7    |
| 32     | 256 | 735       | 1 432     | 0.513 3    |
| 32     | 2k  | 5 215     | 9 494     | 0.549 3    |
| 256    | 256 | 1 176     | 2 143     | 0.548 8    |
| 256    | 2k  | 8 344     | 12 836    | 0.650 0    |
| 256    | 8k  | 33 376    | 44 324    | 0.753 0    |
| 2k     | 2k  | 11 473    | 16 781    | 0.683 7    |
| 2k     | 8k  | 45 892    | 57 613    | 0.796 6    |

4.2 多单元加速比测试

分别使用不同数量的可重构计算单元完成相同的计算任务,以测试系统中的并行加速比.

在流计算工作模式下,8 k 和 32 k 数据量的运算(加法、减法、乘法、累加)任务被分配给 1~8 个可重构计算单元执行,表 5 中列出了实验结果的平均值,并计算了加速比.可以看出,RCU 数量一定,数据量增加时,归一化加速比也升高;数据量一定,RCU 数量增加时,归一化加速比下降.其原因是数

据量越小、RCU 数量越多,不同 RCU 的计算时间越难以重合,从而导致任务周期拉长.此外,当同时使用的 RCU 数量超过 4 时,归一化加速比下降趋势极为明显,主要是由于当前计算系统中集成的 DDR 带宽大约只能为 4 个 RCU 提供数据,可以通过提升存储带宽消除这一瓶颈.当然,考虑到“存储墙”在一段时间内仍然难以消除,在 HMCS 中映射任务时,应尽量使计算簇之间实现级联,从而提高对存储带宽的利用率.

表 5 多可重构计算单元加速比实验

| 数据量 | RCU<br>数量 | 任务<br>周期数 | 加速比  | 归一化<br>加速比 |
|-----|-----------|-----------|------|------------|
| 8k  | 1         | 8 391     | 1    | 1          |
|     | 2         | 4 415     | 1.90 | 0.95       |
|     | 4         | 2 421     | 3.47 | 0.867 5    |
|     | 8         | 2 214     | 3.79 | 0.473 75   |
| 32k | 1         | 32 974    | 1    | 1          |
|     | 2         | 16 815    | 1.96 | 0.98       |
|     | 4         | 8 550     | 3.86 | 0.965      |
|     | 8         | 8 279     | 3.98 | 0.497 5    |

5 结束语

本文针对高密度计算领域的算法特点,为多核计算系统设计了一种可重构计算单元.文中首先介绍了可重构技术的发展情况,然后说明了目标计算系统 HMCS 的体系结构,接着对设计的可重构计算单元的结构与功能特点进行了详细介绍,最后,给出了可重构计算单元的性能测试结果,并根据对实验结果的分析,给出了在 HMCS 上进行算法映射时的注意事项.实验结果表明,本设计方案对高密度计算领域的算法有较高的加速效率,满足项目中的设计需求.

参考文献:

[1] Compton K, Hauck S. Reconfigurable computing: a survey of systems and software [J]. ACM Computing Surveys, 2002, 34(2): 171-210.

[2] 王志远,王建华,徐旸. 可重构计算综述[J]. 小型微型计算机系统,2009,30(6):1203-1207.

[3] Grasset A, Millet P, Bonnot P, et al. The MOR-PHEUS heterogeneous dynamically reconfigurable platform[J]. International Journal of Parallel Programming, 2011, 39(3): 328-356.

99.95%。这种测试方法可以有效评估 SpaceWire 总线传输系统误码率指标,具有较高的测试覆盖性和实用性,为航天型号系统测试保障提供了高效、可靠的技术支持,对于 SpaceWire 总线在航天型号中的应用具有重要的指导意义。

参考文献:

[1] SM. Parkes, SpaceWire-Links, nodes, routers and networks [S]. ESA, 2008.

[2] SM. Parkes, Serial point-to-point Links (ECSS-E-50-12) [Z]. 2000.

[3] Steve Parkes. SpaceWire User's Guide[M]. STAR-Dundee,2012.

[4] 张昊,张春熹. SpaceWire 总线 EMC 设计分析[J]. 电子测量技术,2014,38(37):41-44.

[5] Wolaver D H. Measure error rates quickly and accurately[J]. Electronic Design, 1995(5):89-93.

[6] Redd J. Calculating statictical confidence levels for error probability estimates [J]. Lightwave magazine,

2000(4), 110-114.

[7] 吴从中. 数字通信系统中误码率的估计方法[J]. 合肥工业大学学报,2002,47(25):113-116.

[8] 沈瀚涛,孙学聪,朱蕊蕻. 一种串行通信的误码率测试方法[J]. 现代防御技术,2013,41(41):110-115.

[9] National Semiconductor. LVDS owner's manual[M]. 3rd Edition. City of Santa Clara; National Semiconductor,2004.

作者简介:

**陈卫国** 男,(1983-),高级工程师. 研究方向为新型高速总线设计及测试.

**刘欢** 女,(1983-),高级工程师. 研究方向为新型高速总线设计及测试.

**王剑峰** 男,(1971-),研究员. 研究方向为 IC 设计、Soc 设计.

**吴龙胜(通讯作者)** 男,(1968-),研究员. 研究方向为 IC 设计、Soc 设计. E-mail:wls771@163. com.

(上接第 58 页)

[4] Wawrzynek J,Patterson D,Oskin M,et al. RAMP: research accelerator for multiple processors [J]. IEEE Micro, 2007,27(2):46-57.

[5] 李柏楠. 面向领域应用的可重构系统关键技术研究[D]. 郑州:解放军信息工程大学,2015.

[6] 陈鹏. 面向大数据应用的异构可重构平台关键技术研究[D]. 合肥:中国科学技术大学,2015.

[7] 谭龙生. 一种基于 FPGA 的低开销可重构容错系统设计[D]. 合肥:合肥工业大学,2015.

[8] 王侃文. 领域专用可重构计算结构研究[D]. 上海:复旦大学,2011.

作者简介:

**李浩洋** 男,(1991-),硕士研究生. 研究方向为 SoC 设计. E-mail:hy. li@mail. hfut. edu. cn.

**宋宇鲲** 男,(1975-),博士,副研究员. 研究方向为面向数据高密和计算密集应用的 SoC/MPSO 体系结构与实现.