

多核处理器中的超越函数协处理器设计

黄小康,杜慧敏,李 涛,周佳佳

(西安邮电大学 电子工程学院,陕西 西安 710121)

摘 要: SMT-PAAG 是一种专用于图形、图像及数字信号处理的多核处理器. 介绍了 SMT-PAAG 处理器中的专用于计算超越函数的协处理器设计,包括其原理、特点、实现和验证. 协处理器采用完全流水线结构,基于分段线性逼近的算法实现了一个统一了多种运算的四路算数通道,这些运算包括向量乘法、除法、平方根、点积、三角函数、幂指数及基于任意底的对数运算. 最后在 system verilog 仿真平台上仿真通过并统计了每种运算的误差.

关键词: 多核处理器;协处理器;分段线性逼近;四路算数通道

中图分类号: TN492

文献标识码: A

文章编号: 1000-7180(2016)05-0042-05

A Coprocessor Design for the Computation of Transcendental Functions in Multi-Core Processor

HUANG Xiao-kang, DU Hui-min, LI Tao, ZHOU Jia-jia

(School of Electronic Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710121, China)

Abstract: SMT-PAAG is a multi-core processor for graphics, image and digital signal processing. This article describes the design principle, features, implementation and verification of a coprocessor which uses for transcendental functions based on SMT-PAAG. The coprocessor implements a four-way count channel with a fully pipelined architecture, based on the piecewise linear approximation algorithm, which unified various arithmetic operation including vector multiplication, division, square root, dot product, trigonometric, exponential and logarithmic computation. Finally, we simulated correctly in the simulation platform of system verilog and got the statistical errors for each operation.

Key words: multi-core; coprocessor; piecewise linear approximation; four-way count channel

1 引言

浮点超越函数单元是 GPU 数据通路中的重要部件,其性能直接影响图形渲染效果^[1]. 从精度上来讲,单精度浮点可以满足图形需求,事实上,Nvidia 在 Fermi 之前的 GPU 只支持单精度浮点计算,支持双精度浮点的 Fermi 为 GPGPU,可用于科学计算.

目前在 FPGA 中计算超越函数常用的方法有级数近似法、查表法、CORDIC 算法和分段线性逼近法等.

其中,查表法虽然计算简单、易于实现,但是所

需存储单元随着计算精度的提高呈指数形式增加,资源消耗大^[2]. 分段线性逼近法^[3]将低阶多项式与较小的查找表相结合,编程简单、资源消耗少、计算速度快,被广泛应用于图像处理^[4]等领域,已成为在计算资源有限条件下超越函数计算的一种较为理想的选择.

多核处理器(SMT-PAAG)^[3,5]是西安邮电大学设计的可编程的阵列专用图形图像处理器,为了提高超越函数计算性能,减少面积,SMT-PAAG 采用一种基于分段线性逼近的算法^[4,6-7]定点实现了超越函数:除法、平方根、点积、三角函数、幂指数及基于任意底的对数这几种函数运算.

本文讨论 SMT-PAAG 中定点全流水线的超越函数协处理器设计与实现,实验结果表明,该协处理器的精度为 10^{-3} ,同迭代周期数较少的 CORDIC 相当^[6],满足数字信号处理及图形图像处理方面的应用。

2 算法原理

对数数字系统(Logarithmic number system, LNS)可以极大地简化算术运算,例如,乘法、除法和平方根运算都可以简化成加法、减法和右移运算.但是,LNS 并不能直接用于定点数格式数据的计算.因此,采用一种定点混合数字系统(fixed-point hybrid number system,FXP-HNS)^[8]的设计,并采用分段线性逼近的方法来实现这种数字系统。

2.1 对数转换

任何一个定点数 x 都可以用 $\log_2 X$ 进行逼近,这样就是将任意两个数的运算可转换成对数运算,降低了计算的复杂度。

任意一个 32 位二进制定点对数 x 可表示为

$$x = z_k z_{k-1} \cdots z_2 z_1 z_0, z_{-1} z_{-2} z_{-3} \cdots z_j,$$

即 $x = \sum_{i=j}^k 2^i z_i, z_i = \text{“0”或“1”}$ 。

最高位 z_k 如果为 1 的话, x 就可以表示成:

$$x = 2^k (1 + f), f = \sum_{i=j}^{k-1} 2^{i-k} z_i, 0 \leq f \leq 1.$$

它的对数就可以表示为

$$\log_2 x = k + \log_2 (1 + f).$$

k 是对数的整数部分, $\log_2 (1 + f)$ 是它的尾数部分,范围为 $[0, 1)$ 。

在分段线性逼近的方法中,非线性组 $\log_2 (1 + f)$ 可以被逼近表示为

$\log_2 (1 + f) \cong a_i * f + b_i$.

式中, a_i 和 b_i 是每个逼近区域 i 的逼近系数,这些系数都放在表 1 所示的查找表中. 根据面积和功耗的考虑,选择一个最优的系数,上面的公式可以写为

$a_i * f + b_i \cong f + [\text{inv}_i^0(f) \gg p_i] + [\text{inv}_i^1(f) \gg q_i] + [\text{inv}_i^2(f) \gg r_i] + b_i \triangleq \log_2 (1 + f)$.

式中, a_i 由移位结果的和计算出来, inv_i^l 是一个位翻转的函数, p_i, q_i 和 r_i 是每个逼近区域 i 所定义的移位数目. $(1 + f)$ 被划分成了 14 个逼近区域,每个区域的系数如表 1 所示. 因此, $\log_2 x$ 能够被表示为

$$\log_2 x = k + \log_2 (1 + f).$$

2.2 反对数转换

反对数转换将对数部分计算的结果转换成定点数原理如下:

x 是一个对数值, x 可以表示成:

表 1 修改前的对数转换器查找表系数 ^[6]				
f	$(\text{inv}_i^0(f), p_i)$	$(\text{inv}_i^1(f), q_i)$	$(\text{inv}_i^2(f), r_i)$	b_i
$[0, \frac{1}{2^6})$	$(f, 1)$	$(\sim f, 5)$	$(\sim f, 5)$	0
$[\frac{1}{2^6}, \frac{2}{2^6})$	$(f, 1)$	$(\sim f, 4)$	$(\sim f, 5)$	$\frac{7}{2^{14}}$
$[\frac{2}{2^6}, \frac{4}{2^6})$	$(f, 1)$	$(\sim f, 3)$	$(\sim f, 8)$	$\frac{27}{2^{14}}$
$[\frac{4}{2^6}, \frac{5}{2^6})$	$(f, 1)$	$(\sim f, 3)$	$(\sim f, 4)$	$\frac{47}{2^{13}}$
$[\frac{8}{2^6}, \frac{12}{2^6})$	$(f, 1)$	$(\sim f, 2)$	$(f, 8)$	$\frac{109}{2^{13}}$
$[\frac{12}{2^6}, \frac{16}{2^6})$	$(f, 1)$	$(\sim f, 2)$	$(\sim f, 4)$	$\frac{26}{2^{10}}$
$[\frac{16}{2^6}, \frac{24}{2^6})$	$(f, 2)$	$(\sim f, 3)$	$(\sim f, 6)$	$\frac{46}{2^{10}}$
$[\frac{24}{2^6}, \frac{32}{2^6})$	$(f, 3)$	$(\sim f, 3)$	$(\sim f, 8)$	$\frac{89}{2^{10}}$
$[\frac{32}{2^6}, \frac{40}{2^6})$	$(\sim f, 3)$	$(f, 4)$	$(\sim f, 7)$	$\frac{123}{2^{10}}$
$[\frac{40}{2^6}, \frac{48}{2^6})$	$(\sim f, 3)$	$(\sim f, 7)$	$(\sim f, 7)$	$\frac{1\ 338}{2^{10}}$
$[\frac{48}{2^6}, \frac{56}{2^6})$	$(\sim f, 2)$	$(f, 4)$	$(\sim f, 6)$	$\frac{6\ 881}{2^{15}}$
$[\frac{56}{2^6}, \frac{60}{2^6})$	$(\sim f, 2)$	$(f, 7)$	$(\sim f, 6)$	$\frac{999}{2^{12}}$
$[\frac{60}{2^6}, \frac{62}{2^6})$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 6)$	$\frac{4\ 459\ 550}{2^{24}}$
$[\frac{62}{2^6}, \frac{63}{2^6})$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 7)$	$\frac{4\ 587\ 840}{2^{24}}$
$[\frac{63}{2^6}, 1)$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 8)$	$\frac{4\ 653\ 070}{2^{24}}$

$x = k + f, 0 \leq f \leq 1$.

k 和 f 分别表示 x 的整数和尾数位. 因此, x 的反对数转换可以表示成:

$2^x = 2^k * 2^f$.

在上面的式子中,仅仅是一个移位操作,而非线性的需要被分段线性逼近为

$$2^f \cong a_i * f + b_i.$$

式中, a_i 和 b_i 都是每个逼近区域 i 里面定义的逼近系数, $0 \leq f \leq 1$. 因为反对数转换的误差基本横跨整个区域,所以我们将 f 划分成 8 个逼近区域. 上面的式子可以表示成:

$$a_i * f + b_i \cong f + [\text{inv}_i^0(f) \gg p_i] + [\text{inv}_i^1(f) \gg q_i] + b \triangleq 2^f.$$

a_i 同样由移位数的和来计算得到, inv_i^l 表示位翻转功能, p_i 和 q_i 是每个逼近区域 i 所对应的移位数目. f 被划分成了 8 个逼近区域,各个区域的系数如表 2 所示。

2.3 查找表修改

在设计的过程中,遇到部分定点数的对数转换结果误差过大,经过查错发现部分查找表系数存在问题;如对于 $(1 + \frac{56}{64})$ 的对数,由文献[4]中的查找

表计算出的结果为 0.893 3, C 语言计算出的对数转换结果为 0.906 9, 误差率为 1.498 45%, 这种对数转换误差是无法容忍的.

表 2 反对数转换器查找表系数

f	$(\text{inv}_i^0(f), p_i)$	$(\text{inv}_i^1(f), p_i)$	b_i
$[0, \frac{1}{8})$	$(\sim f, 2)$	$(\sim f, 5)$	$\frac{1\ 024}{2^{10}}$
$[\frac{1}{8}, \frac{2}{8})$	$(\sim f, 2)$	$(f, 5)$	$\frac{8\ 135}{2^{13}}$
$[\frac{2}{8}, \frac{3}{8})$	$(\sim f, 3)$	$(\sim f, 7)$	$\frac{995}{2^{10}}$
$[\frac{3}{8}, \frac{4}{8})$	$(\sim f, 4)$	$(f, 7)$	$\frac{964}{2^{10}}$
$[\frac{4}{8}, \frac{5}{8})$	$(f, 5)$	$(\sim f, 7)$	$\frac{3\ 695}{2^{12}}$
$[\frac{5}{8}, \frac{6}{8})$	$(f, 3)$	$(\sim f, 7)$	$\frac{3\ 455}{2^{12}}$
$[\frac{6}{8}, \frac{7}{8})$	$(f, 2)$	$(\sim f, 5)$	$\frac{1\ 571}{2^{11}}$
$[\frac{7}{8}, 1)$	$(f, 2)$	$(f, 4)$	$\frac{1\ 405}{2^{11}}$

因此本文对查找表部分内容进行了修改, 表 3 所为修改后的查找表系数, 与表 1 中的系数相比, 将 $[\frac{56}{2^6}, \frac{60}{2^6})$ 范围的系数删去并合并到了 $[\frac{60}{2^6}, \frac{62}{2^6})$ 中, $[\frac{63}{2^6}, 1)$ 的 b_i 从 4 653 070 修改到了 4 653 056. 修改后的对数转换结果在可容忍的范围之内, 全部低于 0.4%.

表 3 修改后的对数转换器查找表系数

f	$(\text{inv}_i^0(f), p_i)$	$(\text{inv}_i^1(f), q_i)$	$(\text{inv}_i^2(f), r_i)$	b_i
$[0, \frac{1}{2^6})$	$(f, 1)$	$(\sim f, 5)$	$(\sim f, 5)$	0
$[\frac{1}{2^6}, \frac{2}{2^6})$	$(f, 1)$	$(\sim f, 4)$	$(\sim f, 5)$	$\frac{7}{2^{14}}$
$[\frac{2}{2^6}, \frac{4}{2^6})$	$(f, 1)$	$(\sim f, 3)$	$(\sim f, 8)$	$\frac{27}{2^{14}}$
$[\frac{4}{2^6}, \frac{8}{2^6})$	$(f, 1)$	$(\sim f, 3)$	$(\sim f, 4)$	$\frac{47}{2^{13}}$
$[\frac{8}{2^6}, \frac{12}{2^6})$	$(f, 1)$	$(\sim f, 2)$	$(f, 8)$	$\frac{109}{2^{13}}$
$[\frac{12}{2^6}, \frac{16}{2^6})$	$(f, 1)$	$(\sim f, 2)$	$(\sim f, 4)$	$\frac{26}{2^{10}}$
$[\frac{16}{2^6}, \frac{24}{2^6})$	$(f, 2)$	$(\sim f, 3)$	$(\sim f, 6)$	$\frac{46}{2^{10}}$
$[\frac{24}{2^6}, \frac{32}{2^6})$	$(f, 3)$	$(\sim f, 3)$	$(\sim f, 8)$	$\frac{89}{2^{10}}$
$[\frac{32}{2^6}, \frac{40}{2^6})$	$(\sim f, 3)$	$(f, 4)$	$(\sim f, 7)$	$\frac{123}{2^{10}}$
$[\frac{40}{2^6}, \frac{48}{2^6})$	$(\sim f, 3)$	$(\sim f, 7)$	$(\sim f, 7)$	$\frac{1\ 338}{2^{10}}$
$[\frac{48}{2^6}, \frac{56}{2^6})$	$(\sim f, 2)$	$(f, 4)$	$(\sim f, 6)$	$\frac{6\ 881}{2^{15}}$
$[\frac{56}{2^6}, \frac{62}{2^6})$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 6)$	$\frac{4\ 459\ 550}{2^{24}}$
$[\frac{62}{2^6}, \frac{63}{2^6})$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 7)$	$\frac{4\ 587\ 840}{2^{24}}$
$[\frac{63}{2^6}, 1)$	$(\sim f, 2)$	$(\sim f, 5)$	$(f, 8)$	$\frac{4\ 653\ 056}{2^{24}}$

3 超越函数协处理器单元

现代可编程染色器中的标准算术平台通常采用四路算术通道结构来实现^[9].

计算机图形学中多采用四维矩阵进行运算, 因此采用四算术通道结构, 实现乘法、除法和平方根的向量运算操作.

本模块采用这种结构进行设计, 并在这种结构上实现了多种运算的结合, 包括乘法、除法、平方根、点积、三角函数、幂指数(x^y) 及基于任意底的对数运算($\log_b x$). 下面为各个运算在本结构中的实现原理.

3.1 向量运算

乘法、除法和平方根运算可以表示成一个单独的一般操作:

$$\text{VEC} = (x_i (*) y_i^p (+) z_i), i \in \{0, 1, 2, 3\},$$
$$(*) \in \{*, /\}, (+) \in \{+, -\}, p \in \{0, 5, 1\}.$$

表达式可以转换为

$$\text{VEC} = (2^{\log_2 x_i (+) ((\log_2 y_i \gg q)) (+) z_i}),$$
$$i \in \{0, 1, 2, 3\}, (+) \in \{+, -\}, q \in \{0, 1\}.$$

在这里, 由于这几种运算一次只需要一路通道, 所以可以同时实现四路向量运算.

3.2 三角函数运算

根据上面 VEC 的结构, 可以实现三角函数、双曲线以及它们对应的反三角函数和反双曲线等运算, 在这里只进行了正弦、余弦、正切三种函数的运算, 通过对三角函数进行泰勒级数展开来计算. 在这里定义了一种通用公式来将所有的泰勒级数展开结合起来.

$$\text{TRG} = c_0 x^{k_0} (+) c_1 x^{k_1} (k) c_2 x^{k_2},$$
$$(+) c_3 x^{k_3} (+) c_4 x^{k_4}, (+) \in \{+, -\},$$

表达式可以转换为

$$\text{VEC} = (2^{\log_2 x_i (+) ((\log_2 y_i \gg q)) (+) z_i}),$$
$$i \in \{0, 1, 2, 3\}, (+) \in \{+, -\}, q \in \{0, 1\}.$$

在这个式子里面, 第一项没有转换成对数的格式, 这是因为泰勒级数展开式里面第一项一般是一个常量或者一个和输入 x 近似的值. 另外的项需要在对数运算区域进行乘加计算.

3.3 幂指数及对数运算

幂运算也可以转换成对数区域的乘法运算:

$$\text{POW} = x^y = 2^{(y * \log_2 x)}.$$

基于任意底的对数可以通过下面的式子转换成乘法来计算:

$$\text{LOG} = \log_b x = k * \log_2 x, k = 1/\log_2 b.$$

可以看到,对于采用的结构,还可以扩展实现其他几种超越函数计算,如反三角函数、平方根倒数等,本文算法的电路结构具有较好的扩展性。

4 协处理器与 SMT-PAAG 的集成

SMT-PAAG 由 16 个处理单元(Processing Element, PE)互连构成一个 4×4 的二维阵列,还包括 1 个前端处理器、4 个协处理器、2 个调度器及 2 个存储管理。系统的整体结构如图 1 所示。

协处理器(ACCL)与周围处理单元(PE)的连接关系如图 1 所示。外部存在 4 个 PE 和协处理器相连接,每个 PE 都会发来一个操作码、操作有效信号和操作数 a 、 b 。在协处理器中存在一个预处理模块,将 4 个 PE 的运算进行先后顺序的调节,依次送入流水线进行计算。

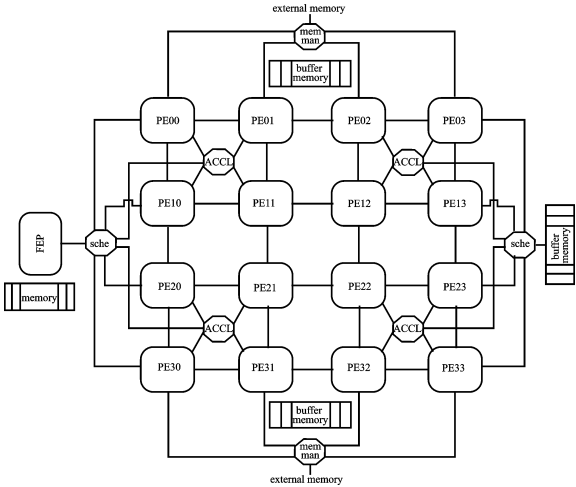


图 1 协处理器整体连接框图

4.1 流水线介绍

为了保证协处理器能够提供足够的计算吞吐量并使协处理器的延迟可预测,本文使用流水线实现整个协处理器设计。

在图 2 中,一共有 6 级流水线,包括预处理和浮点转定点(float switch fixed)模块、对数转换 LOGC 模块、对数运算 LNS 模块、反对数转换 ALOGC 模块、定点数 FXP^[3] (Fixed-point) 运算模块和定点转浮点(fixed switch float)模块。

协处理器首先将四个 PE 发来的不同运算请求进行排序,控制一路或多路数据同时送入流水线,并且对送入流水线的操作数进行浮点转定点处理;LOGC 模块再对操作数求对数,得到的结果;将转换完的结果在 LNS-domain 进行计算,运算的结果通过反对数转换器转换;再将转换完的结果在 FXP-

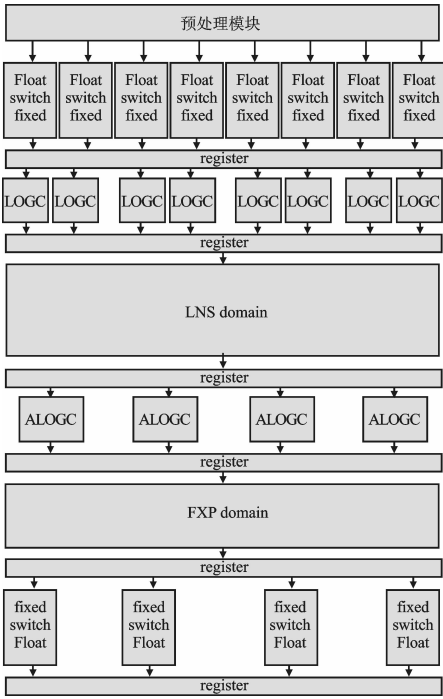


图 2 协处理器 6 级流水线框图

domain 进行计算,而最后的结果转换成浮点数输出。

其中,浮点运算结果 6 拍出,定点运算结果 4 拍出。流水线会根据不同的运算类型,最终输出一路或多路数据到不同的 PE。

4.2 预处理模块介绍

预处理模块如图 3 所示,分为 4 个状态,每个状态对应一个 PE。接下来分别介绍一下九种运算的不同处理情况。

对于乘法、除法、平方根这几种向量运算而言,每次可以进入流水线任意(1、2、3、4)组数据,因为每一条运算通路之间互不影响。点积运算则因为需要用到 4 组数据,所以必须四路数据同时有效才可以开始计算,4 个 PE 必须同时为有效的点积运算类型,才可以使用这 4 组数据进行运算。而对于剩下几种运算,每次只能送一组数据进入流水线,需要对各个 PE 的不同运算请求进行排序处理。

据此我们设计了一个状态机,每种状态对应相应的 PE。例如:如果当前状态为 01,则判断 PE01 的 vld 有效信号是否为 1;若有效,则将 PE01 的数据送入流水线进行计算,同时给 PE01 发送一个 ack 响应信号,表示上一组数据已经送入流水线,可以发送下一组数据了;否则不做操作并且状态机跳到下一状态。

状态机采用轮询的方法设计。在当前状态(如

00)时状态机会判断相邻下一状态(01)的 vld 有效信号;若有效,则跳转到该状态;若无效,则继续按照优先级的顺序判断接下来的状态(10)是否有效,有效则跳转到对应的状态,无效则继续按顺序进行判断.这个优先级的判断顺序是一个固定的优先级顺序,00→01→10→11→00.

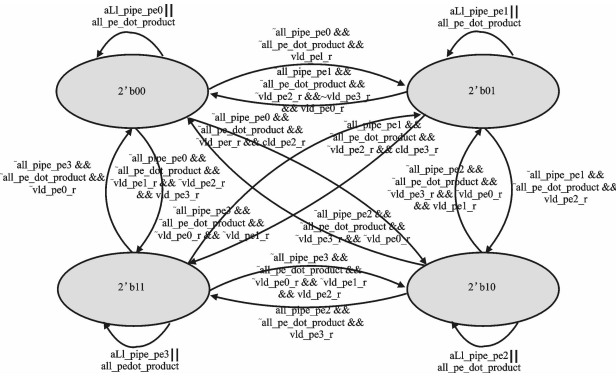


图 3 预处理状态转换图

5 仿真结果分析和比较

针对上面的 FXP 操作和 FXP-HNS 操作,设计了专门的 C 模型来实现预计的算法,用来测试误差结果.除过正切函数,每种运算都在[0,2)之间取了 10 000 个随机的输入向量,每组向量的取值范围都在[0,2)之间,具有普遍性.

表 4 给出了每种操作的最大误差以及两种误差率.在这里,从协处理器最后计算得到的结果会和 C 语言计算得到的单精度浮点结果相比来计算最大误差和误差率.

表 4 运算结果误差率统计

运算类型	最大误差率/%	平均误差率/%	最大误差
乘法	0.254 1	0.042 84	0.000 210
除法	0.233 9	0.051 01	0.000 438
平方根	0.106 2	0.038 80	0.000 751
正弦	0.318 0	0.024 76	0.000 069
余弦	4.434 9	0.205 10	0.000 167
正切	0.945 1	0.175 27	0.000 014
幂指数	0.256 5	0.042 10	0.000 144
lnx	1.607 2	0.126 02	0.000 494
lgx	1.688 1	0.151 81	0.000 149

其中,余弦在 $\frac{\pi}{2}$ 附近产生了一个较高的误差率,这是因为协处理器计算得到的超越函数结果是有一定精度的,因此结果如果在 0 附近,会产生较高的误

差率,但是实际的误差却很小.因此,这种函数计算的误差率并不影响实际的应用.

协处理器是用于图形处理器中 OpenGL 的超越函数运算,本文实现的超越函数计算精度达到小数点后第 3 位,最后的误差并不影响实际显示效果;和软件产生的计算结果相比,本方法产生的微小的区别可以忽略不计.

6 结束语

为了解决 SMT-PAAG 处理器在图形图像处理中遇到的超越函数运算的计算问题,本文描述了一个采用四路运算通道结构的超越函数协处理器设计,该设计具有较好的扩展性,可以实现多种超越函数运算,采用六级完全流水线结构,通过 system verilog 仿真平台计算得到的误差满足图形处理方面的应用.

参考文献:

[1] 焦继业,穆荣,郝跃,等.面向移动图形顶点处理器的高性能低功耗定点特殊函数运算单元[J].电子与信息学报,2011,33(11):2764-2770.

[2] 吴庆达,何书专,潘红兵,等.32 位浮点数正余弦函数 FPGA 实现方法[J].微电子学与计算机,2012,29(1):113-116.

[3] 王少军,张启荣,彭宇,等.超越函数 FPGA 计算的最佳等距分段线性逼近方法[J].仪器仪表学报,2014,35(6):1210-1216.

[4] 鲍宜鹏.一种 CORDIC 算法优化及 32 位浮点反正切函数 FPGA 实现[J].电子与封装,2015,15(3):22-25.

[5] 李涛,肖灵芝.面向图形和图像处理的轻核阵列机结构[J].西安邮电学院学报,2012,17(3):41-47.

[6] Florea C, Vertan C. Piecewise linear approximation of logarithmic image processing models for dynamic range enhancement[J]. IEEE Transactions on Power Systems,2011,26(4):2581-2583.

[7] 陈炳成.基于 FPGA 的一种改进型三角超越函数 CORDIC 实现方式[J].电子世界,2012,31(20):148-150.

[8] Byeong-Gyu Nam, Hyejung Kim, Hoi-Jun Yoo. A low-power unified arithmetic unit for programmable handheld 3-D graphics systems [J]. IEEE journal of solid-state circuits,2007,42(8):1767-1778.

[9] Microsoft Corp. Microsoft DirectX technology overview[EB/OL]. [2015-06-15]. <http://www.microsoft.com/windows/directx>, 2007.

5 结束语

本文提出了一种基于萤火虫算法的解决移动 sink 数据收集路径规划问题的方法,通过解空间的压缩提高了搜索的效率,通过设计离群萤火虫粒子变异和萤火虫个体的逐维移动机制提高了算法的求解精度和收敛速度.通过仿真及对比实验,验证了算法的有效性.下一步的工作将深入研究监控区域存在障碍物的情况,提高算法的适用性;考虑通过引入多移动 sink 节点,进一步缩短规划路径的长度,减少数据收集的延迟.

参考文献:

[1] Wu F J, Yu Chee Tseng. Energy conserving data gathering by mobile mules in a spatially separated wireless sensor network[J]. Wireless Communications and Mobile Computing, 2013,13(15):1369-1385.

[2] Mario Di Francesco, Sajal K Das, Giuseppe Anastasi. Data collection in wireless sensor networks with mile eements; a srvey [J]. ACM Trans on Sensor Networks, 2011,8(1):108-142.

[3] Ma Ming, Yang Yuanyuan, Zhao Miao. Tour planning for mobile data-gathering mechanisms in wireless sensor networks[J]. IEEE Trans on Vehicular Technology,2013,62(4):1472-1483.

[4] De Berg M, Gudmundsson J, Katz M. J, et al. TSP with neighborhoods of varying size[J]. Journal of Algorithms, 2005, 57(1):22-36.

[5] Sugihara R, Rajesh K Gupta. Path planning of data mules in sensor networks[J]. ACM Transactions on Sensor Networks,2011,8(1):1-27.

[6] He Liang, Pan Jianping, Xu Jingdong. A progressive approach to reducing data collection latency in wireless sensor ntworaks with mbile eements[J]. IEEE Trans

on Mobile Computing, 2013, 12(7):1308-1320.

[7] Yung Liang Lai, Jehn Ruey Jiang. A genetic algorithm for data mule path planning in wireless sensor networks[J]. Appl Math Inf Sci, 2013,7(1):413-419.

[8] Krishnanand K N, Ghose D. A glowworm swarm optimization based multi-robot sytem for signal source localization[J]. Design and Control of Intel. Robotic Systems,2009,177(7):49-68.

[9] 周永权,黄正新,刘洪霞. 求解 TSP 问题的离散型萤火虫群优化算法[J]. 电子学报,2012,40(6):1164-1170.

[10] Cook W. Concorde TSP solver [EB/OL]. (2014-06-02). [2015-07-19]. <http://www.math.uwaterloo.ca/tsp/concorde.html>.

[11] Yuan Bo, Orlowska M, Sadiq S. On the optimal robot routing problem in wireless censor networks[J]. IEEE Trans on Konowledge and Date Engineering,2007,19(9):1252-1261.

[12] 彭硕,欧阳艾嘉,乐光学,等. 求解高维函数的改进萤火虫群优化算法 [J]. 计算机应用, 2013, 33 (8): 2253-2256.

[13] 吴斌,崔志勇,倪卫红. 具有混合群智能行为的萤火虫群优化算法研究 [J]. 计算机科学, 2012, 39 (5): 198-200.

作者简介:

俸皓 男,(1978-),博士研究生. 研究方向为无线传感器网络、移动计算、嵌入式实时软件等.
E-mail:fengh@guet.edu.cn.

罗蕾 女,(1967-),教授,博士生导师. 研究方向为移动计算、嵌入式基础软件研究等.

王勇 男,(1964-),博士,教授. 研究方向为智能计算、泛在网研究及应用、计算机网络技术.

董荣胜 男,(1965-),教授,CCF 高级会员. 研究方向为无线传感器网络、网络可靠性、形式化技术、协议工程.

(上接第 46 页)

作者简介:

黄小康 男,(1992-),硕士研究生. 研究方向为电路与系统.
E-mail:hxx9122@gmail.com.

杜慧敏 女,(1966-),博士,教授. 研究方向为计算机体系结

构、计算机图形、大规模集成电路的研究.

李涛 男,(1954-),博士,特聘外籍教授. 研究方向为计算机体系结构、计算机图形学、大规模集成电路.

周佳佳 女,(1989-),硕士研究生. 研究方向为数字集成电路.