

DOI: 10.7652/xjtuxb201502004

面向 OpenCL 的 Mali GPU 仿真器构建研究

崔继岳, 梅魁志, 刘冬冬, 李博良

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对嵌入式 GPU 通用计算的仿真器构建需求,通过对通用图形处理单元仿真器(general purpose graphics processing unit-simulator, GPGPU-sim)的计算核心、存储结构与 Mali GPU 的异同进行比较分析,首先建立面向 OpenCL 的 Mali GPU 仿真器的流程与结构,并设计计算单元数、寄存器数、最小并行粒度等 GPU 微体系结构参数的获取方法,在对 GPGPU-sim 进行修改和配置后,实现了对特定 GPU 架构的仿真器构建。使用矩阵相乘、图像处理等 OpenCL 程序对仿真器的准确性进行测试,以程序在仿真器和硬件平台上的执行周期数差距作为评估依据。实验结果表明:对于测试程序集中优化前的 OpenCL 程序,其中 70% 的程序在两个平台上的运行周期数差距不超过 30%;对于优化后的 OpenCL 程序,其中 90% 的程序的运行周期数差距不超过 30%。由此证明,构建的 GPU 仿真器能够满足 OpenCL 程序的仿真与性能评估。

关键词: 图形处理器;OpenCL;微体系结构参数;仿真器

中图分类号: TP391 **文献标志码:** A **文章编号:** 0253-987X(2015)02-0020-05

Construction of Embedded Mali GPU Simulator for OpenCL

CUI Jiyue, MEI Kuizhi, LIU Dongdong, LI Boliang

(School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: The similarities and differences between GPGPU-sim and Mali GPU in computing cores and the storage structure are analyzed and compared, and simulating procedures and structures of Mali GPUs for OpenCL are built up to develop simulators for the general-purpose computing on embedded GPU. Methods to obtain the GPU microarchitecture parameters such as the computing unit number, the number of registers and the minimum parallel granularity are designed, and then the GPGPU-sim is configured and modified to construct specific GPU simulators. The accuracy of the simulator is tested through comparisons of running OpenCL programs, such as matrix multiplication and image processing on a real GPU and the simulator, and the difference between running cycles on the real GPU and the simulator is used as evaluation. Results show that the cycle differences are within 30% for about 70% OpenCL programs with simple implementation, and the cycle differences are within 30% for about 90% OpenCL programs with optimization. Therefore, it can be concluded that the constructed simulator meets the requirements of simulating and evaluating OpenCL programs on the embedded GPU.

Keywords: GPU; OpenCL; microarchitecture parameters; simulator

GPU 通用计算(general purpose computing on GPU, GPGPU)技术将 GPU 与 CPU 组成异构计

收稿日期: 2014-07-06。 作者简介: 崔继岳(1988—),男,硕士生;梅魁志(通信作者),男,副教授。 基金项目: 国家高技术
研究发展计划资助项目(2012AA010904);国家自然科学基金资助项目(61375023)。

网络出版时间: 2015-01-05 网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20150105.0859.007.html>

算平台,使得 GPU 不再局限于传统的图形计算,可以加快现有算法运行速率^[1]。嵌入式设备的 GPU 已经采用多核架构,使得利用嵌入式 GPU 进行通用计算成为新的研究方向和热点,例如 ARM 发布的 Mali T6 系列 GPU 目前已支持 OpenCL 1.1 规范,对嵌入式平台的 GPGPU 开发环境产生了需求,因此本文构建了 Mali 系列 GPU 仿真器,以满足 OpenCL 程序的运行性能预测与程序优化。

1 嵌入式 GPU 仿真器的构建设计

GPU 的体系结构决定了 OpenCL 程序的执行方式,与嵌入式 GPU 的体系结构保持一致是仿真器的准确性的前提^[2]。本文以通用图形处理单元仿真器 (general purpose graphics processing unit-simulator, GPGPU-sim) 作为仿真器设计基础,对其进行修改,使其体系结构与 Mali T-628 一致,以完成仿真器的构建。

1.1 GPGPU-sim 仿真结构

GPGPU-sim 仿真器面向 GPU 通用计算,着色器模型采用统一渲染架构,由 UBC 大学 Aamodt 等于 2009 年发布,以 NVIDIA 的 GPU 为模拟对象^[2]。GPGPU-sim 提供了详细的配置参数,能够对着色器结构以及内存缓存结构进行详细模拟,并提供了 OpenCL 接口,支持对 OpenCL 程序的仿真。

图 1 给出了 GPGPU-sim 的整体结构。计算核心部分模拟了 GPU 的并行核心架构,每个核心都是一个单指令多线程 (single instruction multiple thread, SIMT) 架构。SIMT 核心负责指令的执行,包括算术运算指令和数据存取指令。在 SIMT 核心中含有多个算术逻辑单元 (arithmetic logic unit, ALU),所有的 ALU 执行相同的操作以实现并行化。GPGPU-sim 的缓存体系模块包括位于核心内的一级缓存、本地内存和位于存储器的二级缓存、内存芯片 (DRAM) 4 部分。计算核心与外部的存储器通过内部互联网络总线相连。

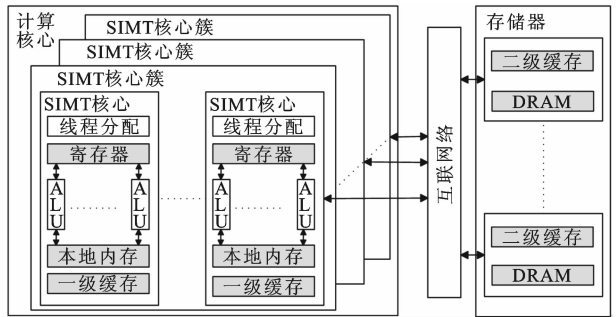


图 1 GPGPU-sim 的体系结构

1.2 Mali 系列 GPU 体系结构

Mali 系列 GPU 主要分为计算核心和存储结构两部分。

存储结构部分由寄存器、一级缓存、二级缓存和系统内存 4 部分组成。在系统内存中划分出 4 块内存空间——私有内存、常量内存、本地内存和全局内存,分别与 OpenCL 的内存模型相对应,其中私有内存和寄存器共同存储 OpenCL 的私有类型变量。

仿真对象 Mali T-628 的计算核心内部包含一个矢量处理器和一个标量处理器,矢量处理器为 128 位的单指令多数据 (single instruction multiple data, SIMD) 结构,标量处理器单周期内能够处理不大于 32 位的算术运算。图 2 给出了 Mali GPU 体系结构的抽象模型。

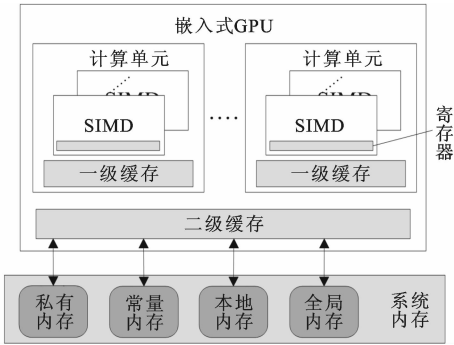


图 2 嵌入式 GPU 体系结构抽象模型

对比图 1 和图 2,可以看出 GPGPU-sim 的体系结构和嵌入式 GPU 体系架构相类似,GPGPU-sim 中片上互连网络模块支持 3 种模式,不同模式对性能影响较小,因此本文对 Mali T-628 中互连网络模块部分不做研究,仿真时将其设为蝶形模式。GPGPU-sim 与 Mali T-628 计算核心和存储系统体系结构的主要区别如下。

(1)GPGPU-sim 计算核心为多个 32 位标量处理器,Mali T-628 计算核心为 128 位矢量处理器和一个标量处理器,需要对 GPGPU-sim 仿真流程和计算核心部分进行修改。

(2)二级缓存的位置不同。GPGPU-sim 的二级缓存在外部内存,计算核心通过互连网络从二级缓存中读取数据;嵌入式 GPU 的二级缓存在 GPU 芯片内,二级缓存通过互连网络与外部存储器交换数据。因此,计算核心对外部存储器的数据传输在两种架构中具有一定的等价性。

(3)本地内存的存取速率不同。GPGPU-sim 中本地内存的存取速率和一级缓存一致,Mali T-628 本地内存存取速率和外部存储器一致。当 OpenCL 程

序中大量使用本地内存时,GPGPU-sim 中对本地内存中数据访问延迟小,而 Mali GPU 中本地内存中数据访问延迟大,仿真结果会有较大差别。

1.3 GPGPU-sim 仿真流程修改

在 NVIDIA 平台,OpenCL 的 kernel 函数首先编译成并行线程执行(parallel thread execution, PTX)语言,然后加载到 GPU 上执行。GPGPU-sim 实际上是以 PTX 语言作为输入,通过对其解析能够有效确定 GPU 的行为实现仿真。矢量核心的特性是指令并行度随处理数据的宽度变化,而 GPGPU-sim 中的标量核心指令并行度是固定的。为了使 GPGPU-sim 支持矢量处理器特性,本文加入了对指令的逻辑判断,对于不同的指令分配不同的并行宽度,并将其封装为指令检测器模块,修改后的仿真流程如图 3 所示。

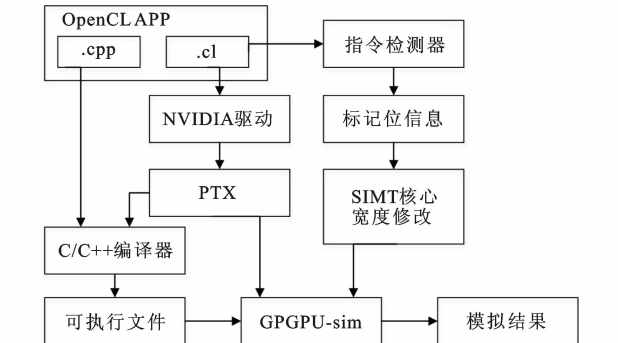


图 3 修改后的 GPGPU-sim 仿真流程

图 4 给出了指令检测器的内部流程。在 GPGPU-sim 中增加一个指令宽度标记位,每一个数据类型都有其对应的标记值。在仿真器运行前,首先对核函数进行数据类型检测,得到标记位信息,然后对核心并行宽度进行设置,直至当前核函数执行完毕,

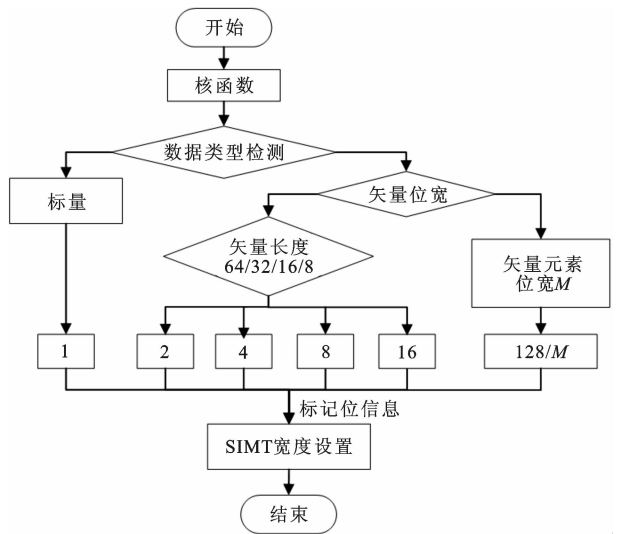


图 4 指令检测器的内部流程

在下一个核函数执行前再重复该过程。

2 GPU 微体系结构关键参数获取

GPGPU-sim 提供了详细的配置参数^[3],能够对计算核心模块、存储系统模块以及互连网络模块进行配置。仿真器各模块参数与 Mali GPU 参数一致性越高,仿真结果就越准确。由于从公开的技术文档无法得到 Mali T-628 的结构参数,本文参考 NVIDIA 平台^[4]以及 AMD 平台^[5]的体系结构来研究参数获取。

2.1 算术运算指令延迟

指令延迟主要研究 GPU 的算术逻辑单元对不同算术运算指令的执行时间。为了减少误差,需要在 GPU 中填充大量的指令,使得 GPU 在程序运行时间内主要进行数据运算,得到核函数的运行时间 T 后即可根据核函数内循环次数与 GPU 的时钟频率得到对应指令执行的时钟周期数。测试结果如表 1 所示(Mali GPU 不支持双精度浮点数据运算)。

表 1 Mali T-628 算术运算指令延迟

数据类型	延迟周期数			
	加	减	乘	除
Float	29	29	29	130
Int	2	2	3	14

2.2 计算单元数目

OpenCL 中工作项被划分为不同的工作组,一个工作组在 GPU 上的一个计算单元执行时,可以通过分配不同工作组的数目来测量计算单元的数目,当工作组的数目从 1 依次增加时,程序的执行时间会按照 $1 \sim S, S+1 \sim 2S, \dots$ 的规律呈现阶梯式增长,阶梯的长度 S 即计算单元的数目。测试程序在 Mali T-628 上的运行结果见图 5,可得出计算单元数为 4。

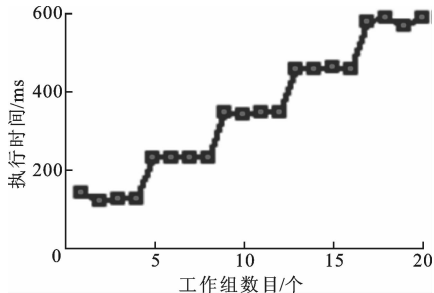


图 5 执行时间与工作组数的关系

2.3 寄存器数目

在 OpenCL 中,每个工作项的临时变量存放在 GPU 设备上的私有内存中,Mali GPU 的私有内存

包含两部分,一部分是寄存器,另一部分是在设备内存中划分的,其存取性能较寄存器有明显差距。

如图 6 所示,Mali GPU 的寄存器数目可以得到确定。当单个工作项占用的临时变量大于寄存器数目时,多余的临时变量就会存放到设备内存的私有内存部分,计算时间会快速增长,由此可确定每个工作项最大支持的寄存器数目。图 6a 中,最大寄存器数目为 38,此时该工作项执行时间为 0.955 s。保持每个工作项占用寄存器数目为最大值,不断扩大工作组内工作项的数目,当整个计算单元的寄存器全部占用时,执行时间增长幅度变大。图 6b 中,工作项数目增加到 73 时,执行时间为 8.833 ms,因此一个计算单元内的寄存器数目约为 73 与 38 的乘积,即 2 774。

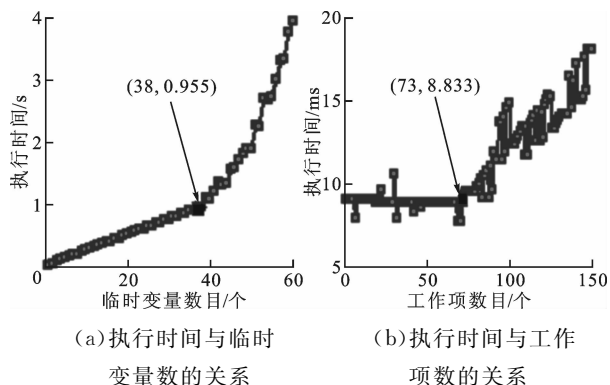


图 6 寄存器数目测试结果

2.4 最小并行粒度

假设 GPU 的最小线程并行数是 N ,那么该 GPU 将以 N 个工作项作为一组原子并发的线程组。如果遇到分支,那么一组 N 个工作项都将介入这个分支,对于不满足条件的工作项,则会等到当前分支处理之后再执行。利用最小并行粒度内的工作项并发执行这一特性,在基准测试设计中,假设工作组大小为 $2N$,在核函数中增加一个判断语句,当工作项标识符小于 N 时执行操作 A,否则执行操作 B,这时 $2N$ 个工作项在运行时,操作 B 在操作 A 之前执行。当工作组大小恰好为 N 时,操作 A 是在操作 B 之前执行的。通过标记位信息可以判断出操作 A 和操作 B 的执行顺序,从而判断工作组大小与最小并行粒度是否相等。当工作组足够大时,逐次递减判断,就可以找到最小并行粒度的值。经测试,Mali T-628 最小并行粒度为 1。

2.5 缓存结构

利用程序的空间局部性和时间局部性原理可以构造出一个存储器测试程序,通过数据的访问速率

分析出缓存结构。设计的特殊数组结构及其数值初始化方式为^[6]

$$X_i = (p + i) \% L \quad (1)$$

式中: X 为数组; i 为数组下标; L 为数组长度; p 为偏移量(4B,8B,12B,...)。

当数组中偏移量不断增加时,缓存中数据的更新次数也逐渐增加,程序的执行时间同比增长,当偏移量达到缓存中块大小时,缓存中数据的更新次数会暂时达到一个稳定值。偏移量继续增加到组大小时,缓存中数据的更新频率再次达到一个稳定值。依据此特性可以分析出缓存中的块大小和组大小。图 7 显示了利用该方法获得的 Mali T-628 中缓存结构。当偏移量小于 64 B 时,执行时间正比增加;当偏移量为 64 B 时,执行时间是 0.632 s;当偏移量在 64 B~1 KB 之间时,执行时间基本不变,可以确定块大小为 64 B;当偏移量在 1 KB~30 KB 时执行时间正比增加,可以确定组大小为 1 KB;当偏移量为 30 KB 时,执行时间是 1.030 s;当偏移量大于 30 KB 后执行时间基本不变,可以确定一个计算单元内缓存总大小为 30 KB 左右。

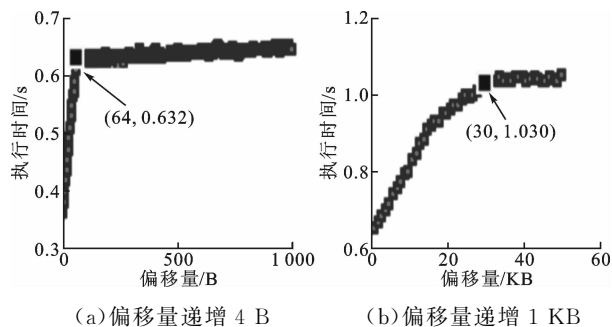


图 7 执行时间与数组偏移量关系

2.6 Mali T-628 体系结构参数设定

结合 OpenCL 中的函数 `clGetDeviceInfo()` 可以获得 Mali T-628 的基本信息,最终得到的仿真器中各模块的体系结构参数值与 GPGPU-sim 的默认参数值(用来模拟 NVIDIA 平台的 Quadro FX5800)的比较见表 2。片上互连网络类型(Topology)、二级缓存结构(gpgpu_cache: dl2)、一级指令缓存(gpgpu_cache: il1)、一级纹理缓存(gpgpu_cache: l1)等值采用 GPGPU-sim 中提供的默认值。

3 实验及结果分析

3.1 硬件平台

实验使用基于三星 Exynos 5420 SoC 的 InSignal Arndale Octa 5420 开发板作为平台,操作系统

为 Android 4.2,内核为 Linux 3.4.39。GPU 为 ARM 的 Mali T-628,核心频率为 533 MHz。

表 2 Mali T-628 体系结构参数

参数内容	Quadro FX5800	Mali T-628	获取 方法
gpgpu_n_clusters (SIMT 簇数目)	10	4	测试
gpgpu_n_cores_per_cluster (每簇 SIMT 核心数目)	3	2	基本 信息
gpgpu_shader_registers (寄存器数目)	16 384	2 774	测试
K (互联端口数目)	18	5	基本 信息
gpgpu_n_mem (内存片数目)	8	1	基本 信息

3.2 测试程序

OpenCL 程序的开发分为两种:第一种将算法用 OpenCL 实现;第二种依据硬件平台的体系结构对 OpenCL 程序进行优化,以提升程序的运行速度。因此,仿真器的准确性测试包括优化前和优化后的两种 OpenCL 程序测试。实验选用的矩阵相乘算法实现了粗粒度和细粒度版本,其中细粒度版本每个工作项计算目标矩阵中一个元素,粗粒度版本每个工作项计算目标矩阵中一行元素;方向梯度直方图(histogram of oriented gradient, HOG)特征提取程序的算法复杂程度较高,主要测试仿真器对于复杂算法仿真时的准确性。

3.3 实验结果

为验证仿真器的准确性,需要对比同样的 OpenCL 程序在真实硬件与仿真器上的运行状态。本文采用周期数作为衡量标准,仿真器运行结束后会返回执行的周期数,而真实硬件上只能返回程序运行时间,将运行时间与频率的乘积作为硬件平台上的运行周期。

表 3、表 4、表 5 列出了测试程序在仿真器与模拟对象 Mali T-628 上执行的周期数差距,对每一个测试程序均运行了多次。

测试结果表明,对于优化前的 OpenCL 程序,其中 70%的测试程序周期数差距在 30%以内;对优化后的 OpenCL 程序,其中 90%的测试程序周期数差距在 30%以内。Bakhoda 等在对 GPGPU-sim 进行测试时,周期数差距在 30%以内的测试程序占总数的 67%^[2]。

表 3 矩阵相乘周期数差距测试结果 %

矩阵相乘 OpenCL 程序	矩阵相乘周期数差距		
	40 阶	100 阶	200 阶
细粒度优化前	45	50	50
细粒度优化后	8	26	15
粗粒度优化前	23	19	123
粗粒度优化后	13	16	140

表 4 图像边缘检测周期数差距测试结果 %

图像边缘 检测 OpenCL 程序	图像边缘检测周期数差距			
	800× 400	1 280× 720	1 920× 1 080	2 560× 1 600
优化前	23	24	24	25
优化后	5	4	3	3

表 5 HOG 特征提取周期数差距测试结果 %

HOG 提取 OpenCL 程序	HOG 提取周期数差距		
	1 280×720	1 920×1 080	2 560×2 560
优化前	19	23	26

因为对存储系统中二级缓存和主存储器的建模利用了体系结构的等价性(采用 GPGPU-sim 中提供的默认值),与真实 GPU 之间存在差距,所以在矩阵计算的粗粒度版本中,当矩阵规模增加到 200 阶时,对外部数据访问频繁,仿真准确性因而变差。

4 结 论

本文通过对嵌入式 GPU 关键微体系结构的参数获取,在现有仿真器 GPGPU-sim 的基础上进行了改进,设计实现了 Mali T-628 GPU 仿真器,并使用多种 OpenCL 程序测试了仿真器的准确性和有效性。后续工作将进一步完善该仿真器存储系统模块参数的获取方法,以提供更高的仿真精度。

参考文献:

[1] NVIDIA. NVIDIA GeForce 8800 GPU architecture overview, TB-02787-001_V01[R]. Santa Clara, CA, USA: NVIDIA Corporation, 2006.

[2] BAKHODA A, YUAN G L, FUNG W W L, et al. Analyzing CUDA workloads using a detailed GPU simulator [C] // Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software. Piscataway, NJ, USA: IEEE, 2009: 163-174.

(下转第 68 页)