

引用格式:贾蕊,李涛,冯臻夫,等.高性能机器学习 SIMT 处理器的调度机制设计与实现[J].微电子学与计算机,2019,36(9):67-72. [JIA R, LI T, FENG Z F, et al. Design and implementation of scheduling mechanism for high performance machine learning SIMT processor[J]. Microelectronics & Computer, 2019, 36(9): 67-72.]

高性能机器学习 SIMT 处理器的调度机制设计与实现

贾蕊¹, 李涛^{1,2}, 冯臻夫¹, 张宏伟¹

(1 西安邮电大学 电子工程学院, 陕西 西安 710121;

2 西安邮电大学 计算机学院, 陕西 西安 710121)

摘 要: 针对面向机器学习的高性能单指令多线程(Single Instruction Multiple Threads, SIMT)处理器提出了结构简单且高效的调度机制, 支持 4 个区块、8 个 warp、64 个线程的并行运算, 并采用两种可配置调度模式相结合的动态调度方式. 该设计使用可综合的 Verilog HDL 语言实现其硬件电路, 并搭建基于 FPGA 的验证平台对整体电路进行功能验证, 结果表明, 本文设计的调度机制满足 SIMT 处理器需求, 且该调度机制使得处理器整体性能提升了 82.17%. 在 Xilinx 公司的 FPGA 芯片 xcvu440-flga-2892-2-e 上综合最大时钟频率可达到 181 MHz.

关键词: 机器学习; SIMT 处理器; SIMT 调度机制; 多线程并行处理; 动态调度

中图分类号: TN492

文献标识码: A

文章编号: 1000-7180(2019)09-0067-06

Design and implementation of scheduling mechanism for high performance machine learning SIMT processor

JIA Rui¹, LI Tao^{1,2}, FENG Zhen-fu¹, ZHANG Hong-wei¹

(1 School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

2 School of Computer, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

Abstract: Aiming at the high-performance Single Instruction Multiple Threads (SIMT) processor for machine learning, a simple and efficient scheduling mechanism is proposed, which supports parallel operation of 4 blocks, 8 warps and 64 threads. The dynamic scheduling method combines two configurable scheduling modes. The design uses the synthesizable Verilog HDL language to implement its hardware circuit, and builds an FPGA-based verification platform to verify the function of the whole circuit. The results show that the scheduling mechanism designed in this paper meets the requirements of SIMT processor, and the overall performance of the processor is increased by 82.17%. The integrated maximum clock frequency can reach 181MHz on Xilinx's FPGA chip xcvu440-flga-2892-2-e.

Key words: machine learning; SIMT processor; SIMT scheduling mechanism; multi-thread parallel processing; dynamic scheduling

1 引言

大数据时代的到来使得机器学习技术应用于更多场景, 并得到越来越多的关注. 日益增大的数据量

在促进机器学习的性能不断提高的同时, 衍生出一个棘手问题: 训练过程耗时太长导致大量数据得不到及时处理. 一个非常重要的原因就是通用处理器不适合于大规模计算, 其处理速度无法满足机器

学习^[1].

2017 年,ISCA 和 ISSCC 专门针对深度学习进行讨论;2018 年,ARM 发布第一代名为“Trillium”的面向 AI 和机器学习的处理器;NVIDIA 的最新版 GPU 使用其最新的 Tensor 核心架构,FP32 可以累积或增加以适应机器学习,由此可见,深度学习对专用处理器设计产生了巨大影响^[1].

大数据处理和深度学习算法中数据间的独立性,使此类应用适用于并行计算^[1],因此需要一种高效的并行结构^[2-4]来进行大规模的机器学习.在众多提高处理器性能的研究中^[2-4],单指令多线程^[5-6](SIMT)结构得到高度关注. SIMT 体系结构在第一个 Unified Computing GPU 中由 NVIDIA 公司生产的 GPU^[5-7]引入. SIMT 架构下指令分配给调度器所控制的多个计算核(PE)同时执行数据的计算和处理,每个 PE 为一个线程,即多线程同时进行数据的处理.如此高效执行的并行计算很大程度上依靠的是其高效的多线程调度机制^[6],SIMT 调度过程分为软件调度和硬件调度过程,其中硬件调度过程是该结构并行执行高效性的关键所在^[6].

2 整体硬件结构

自主架构处理器采用 SIMT + VLIW (Very Long Instruction Word,超长指令字)并行方式,形成异构计算加速模式. SIMT 组处理器含有 8 个流处理器(PE)、32 页程序存储、8 页数据存储、8K 字 cache 缓存、计算控制器(VPU)和存储控制器.每个 PE 对应一个线程,8 个 SIMT 线程构成一个 warp,每个 SIMT 组可以容纳最多 8 个活跃 warp,实现 8 个 warp、64 个线程之间的并行运算.其中 VPU 控制器实现整体处理器的调度机制. SIMT 组处理器整体硬件结构如图 1 所示.

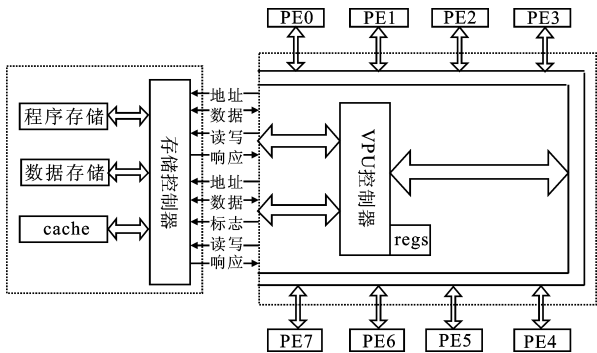


图 1 SIMT 处理器整体架构

3 调度机制功能描述

SIMT 处理器调度机制主要利用线程的数量来隐藏因存储器访问和其他耗时较长指令导致的线程执行流水线停滞所带来的延时^[6].以 warp 为单位的硬件调度主要包含取指阶段、译码阶段、发射阶段,以及 warp 切换调度.

取指阶段,取指单元根据分派的 warp 号及 pc 从存储单元取指,并将指令放入相应 warp 号的缓存队列,同时,根据当前正在执行的 warp 号,向译码单元发送指令.译码阶段,对所取的指令进行译码,并将译码后的信息根据当前 warp 号放入相应缓存队列,若译出为特殊指令,即需要切换的指令,同时向 warp 切换调度单元发出切换请求,并暂停当前 warp 的取指、译码,直到切换响应.发射阶段,根据当前 warp 号及调度状态控制指令发射,将相应缓存队列中的译码后的指令信息发送给 PE 单元执行计算. warp 切换调度阶段,作为程序执行的总调度单元,负责接收启动、配置信息并根据程序运行实时情况进行效率最高化优化.如果一个 warp 执行的指令需要较长时钟周期,则该 warp 可暂时被挂起,调度器会按优先级调度其他 warp 去执行,这样在该 warp 被挂起等候的时候,处理器并没有空闲,隐藏了线程执行的延时时间. SIMT 处理器调度机制模型如图 2 所示.

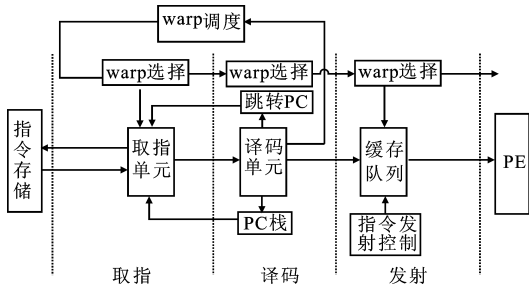


图 2 SIMT 处理器调度机制模型

具体切换调度流程为:接收译码单元的切换请求,根据系统运行情况进行相应应答.检查当前 warp 状态寄存器,根据当前 warp 队列的激活及有效状态返回响应信号,如有满足调度条件的 warp 则切换响应为 1,根据模式选择,结合排序算法及各 warp 状态进行调度后,给取指单元分派新的 warp 号,一轮调度结束;否则切换响应为 0,即当前无可调度 warp 需等待. warp 切换调度流程如图 3 所示.

该机制兼容两种调度模式,由外部控制信号进行选择.第一种,根据区块间优先,区块内轮询的规

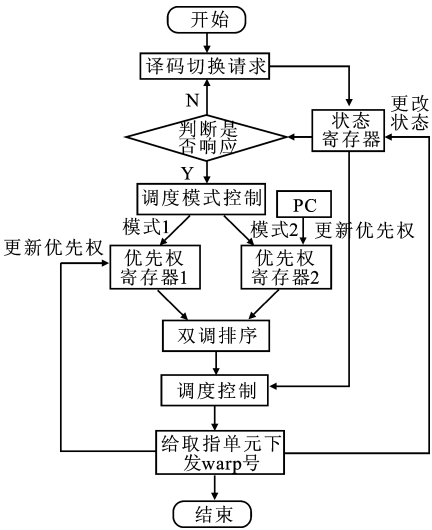


图 3 warp 切换调度流程图

则;第二种,根据每个 warp 已执行的指令数确定优先级,已执行指令数多则优先级低,反之则高. 对于单个 warp 的特殊处理:若程序初始配置只有一个有效的 warp,则无须启动切换调度单元,消除调度消耗的时间.

4 调度机制整体设计方案

计算控制器(VPU)为该 SIMT 处理器的中央调度部分,包括取指单元(FETCH)、译码单元(DECODE)、调度控制单元(WARP_CTRL)、8 线程数数据存取单元(LDST)、溢出缓存单元(PE_STA),VPU 整体结构如图 4 所示.

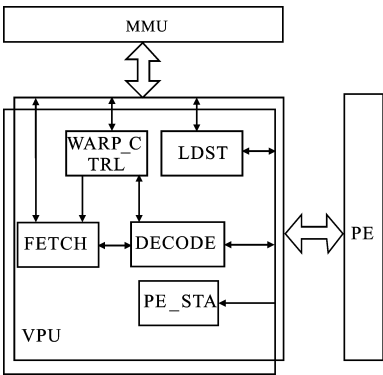


图 4 VPU 整体结构框图

其中调度机制的核心部分为调度控制单元(WARP_CTRL),设计包括动态优先级模块、warp 状态模块、排序模块、控制模块及 WC 状态模块,调度控制单元(WARP_CTRL)整体结构如图 5 所示.

动态优先级模块用于各 warp 调度优先权的确定,根据调度过程中 warp 的执行状态,实时更改每

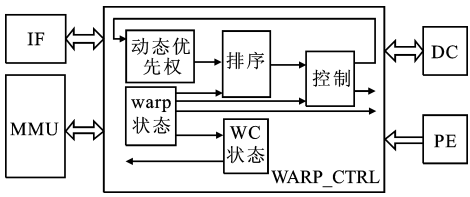


图 5 调度控制单元整体结构框图

个 warp 的优先级. warp 状态模块存储 warp 的执行状态,每个 warp 分配一个状态寄存器,记录该 warp 的有效状态、激活状态、完成状态以及同步等待状态. 排序模块主要功能为将各 warp 的动态优先级进行实时排序,用于后续控制模块的调度. 控制模块的主要功能为当有 warp 切换请求并响应时,综合排序后 warp 的优先级以及状态,调度得到可执行的 warp. WC 状态模块主要功能为记录整个程序调度执行的状态,并向上级反馈.

5 调度控制详细设计

调度控制单元(WARP_CTRL)详细设计如图 6 所示.

5.1 WARP_STA 状态模块

WARP_STA 为 warp 状态更新模块,设置 8 个 6 位状态寄存器分别对应 8 个 warp,每个状态寄存器组成如图 7 所示.

有效位:由初始配置确定,表示该 warp 是否有效,高电平有效. 激活位:由 warp 的执行状态确定,激活状态下才可被调度,激活状态为 1. 完成位:标记该 warp 本轮所有指令是否已全部执行,由译码单元译出 halt 指令后发送完成信号,将该位置 0,不再被调度,未完成状态下为 1. 每个 warp 结束时都有自己独立的 halt 指令,当所有 warp 均处于完成状态,准备接收新一轮的初始化及配置信息. 同步位:用于 SYNC 同步指令,若该 warp 到达 SYNC,则该位置 1,表示非激活,不可被调度;当该 warp 所属区块的所有 warp 的同步位均为 1,则重新激活本区块所有 warp,同步位置 0,此时可被调度. 区块号(#B):记录 warp 所属的区块号,主要用于同步指令的区块识别.

5.2 RANK 排序模块

排序模块采用双调排序(Bitonic Sort),对 8 个 warp 的优先级进行排序,进行 6 次排序后得到一组从小到大的顺序数组,数值越小优先级越高,数值越大优先级越低,排序有效后依次进行调度.

双调排序算法是一种比较顺序与数据无关的排

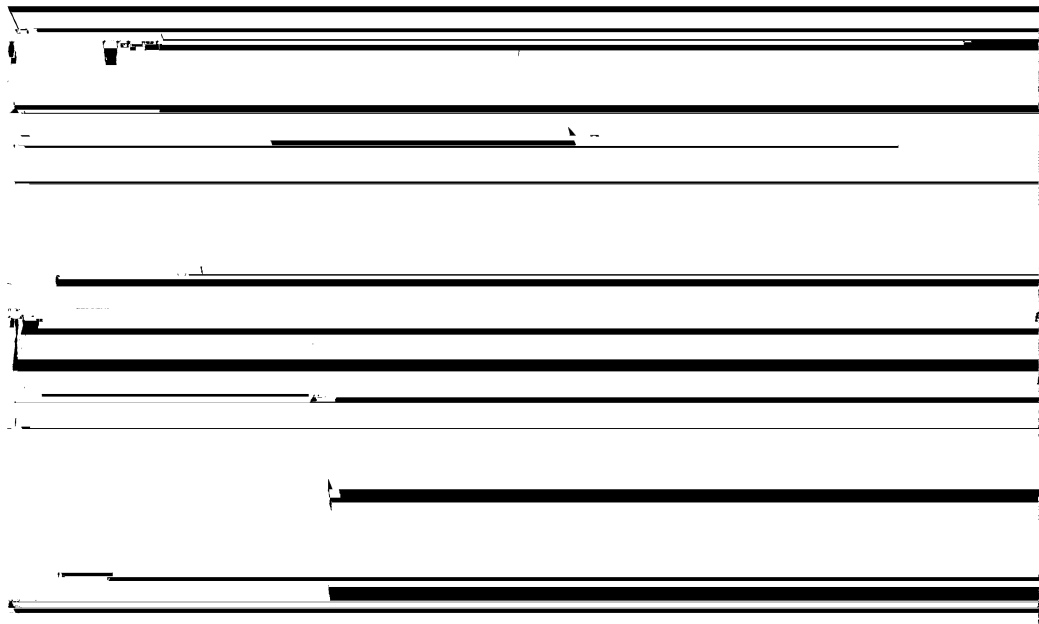


图 6 调度控制单元详细电路图

5	4	3	2	1	0
#B	有效	激活	完成	同步	

图 7 warp 状态寄存器

序方法,适合做并行计算. 双调序列是指一个先单调递增后单调递减(或者先单调递减后单调递增)的序列. 一个随机序列首先转化成一个双调序列,然后在再按算法进行排序. 本设计需要对 8 个 warp 的优先权数据进行双调序列转化并进行双调排序,最终转化成从小到大递增的序列,8 个数据所组成的序列排序具体步骤如图 8 所示.

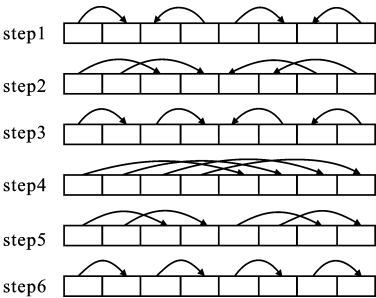


图 8 8 个数据的双调排序

5.3 动态优先权模块

该模块为 warp 优先权寄存器,prio_reg0 为第一种调度模式下的动态优先权更新模块,prio_reg1 为第二种调度模式下的动态优先权更新模块,可根据外部控制信号进行模式切换.

第一种调度模式动态优先权的更新以实现区块

间优先,区块内轮询原则进行规律变化. 根据轮询(Round robin)调度算法^[8]中提出的,对于在一个时间片内未完成要求的任务,调度程序暂时停止该进程的执行,将它排到就绪队列的末尾,等待下一次调度,调度程序会调度当前就绪队列中的下一个进程. 在本文设计中初始的优先权由 MMU 下发的初始配置信息确定,用 warp 的索引号中区块号、区块优先级以及区块内的 warp 号作为初始优先权.

动态优先权更新方式:由译码单元发来切换请求的 warp,应该将其执行顺序排至该 warp 所在区块队列末尾,同时,该区块的其他 warp 优先权都同时提高一级. 但由于译码单元发来的切换请求的 warp 即为上一次调度控制单元调度下发的,所以为了提高切换调度时的效率,将每一次调度成功的 warp 直接更改其优先权,从而节省下次切换调度前更新优先权的时间. 优先权具体更改规则为:每成功调度一个 warp,将当前 warp 的优先权中[2:0]改为 3'b111,排至所在区块队列末尾,该区块的其他所有 warp 优先权均减 1. 假设此次调度到的是 warp0,给取指单元下发 warp0 的同时,将 warp0 按优先权的排序排到它所在区块的末尾,该区块的其他 warp 同时提升,则 warp1 移为排序中的首位,下次调度时首先依次询问区块 B0 中的 warp1, warp2, warp3, warp0,进而再询问区块 B1, B2. warp 优先权的动态更新方式如图 9 所示.

第二种调度模式动态优先权的更新根据每个

表 2 矩阵乘性能对比表

	所需时钟周期数	性能提升/%
通用处理器	4 588	
本设计	818	82.17%

本设计采用 Xilinx 公司的 FPGA 芯片 xc-vu440-flga-2892-2-e 进行综合,最大时钟频率可达到 181MHz.

7 结束语

本文针对面向机器学习的高性能单指令多线程 (SIMT)处理器设计了结构简单且高效的调度机制,提供两种可配置的动态调度模式,可根据数据处理的需求选择合适的调度策略.通过搭建基于 FPGA 的验证平台,对整个电路的功能进行了全面的 功能验证,本文设计的调度机制满足 SIMT 处理器需求,且该调度机提高了线程的并发执行效率,实现了该结构的高效并行处理,使得处理器整体性能提升了 82.17%.本文设计的调度机制可扩展性强,可以扩展为 2n 个 warp、任意区块数的调度;可移植性强,可用于其他多核多线程处理器调度.接下来工作为进行多个 SIMT 处理器调度器的扩展,以更适用于机器学习大数据量的快速并行处理.

参考文献:

[1] 刘道福. 机器学习处理器研究[D]. 北京:中国科学院大学,2015.
[2] 吴佳骏. 多核多线程处理器上任务调度技术的研究

[D]. 北京:中国科学院计算技术研究所,2006.
[3] 李祖松,徐先超,胡伟武,等. 龙芯 2 号处理器的同时多线程设计[J]. 计算机学报, 2009,32(11):2266-2273.
[4] 李涛,杨婷,易学渊. 萤火虫 2: 一种多态并行机的硬件体系结构[J]. 计算机工程与科学, 2014, 12 (2): 191-200.
[5] KYUNG G, JUNG C, LEE K. An implementation of a SIMT architecture — based stream processor[C]// Tencon IEEE Region 10 Conference, Chengdu, China, IEEE, 2015.
[6] 徐元旭. SIMT 线程调度模型分析及优化[D]. 哈尔滨:哈尔滨工业大学, 2013.
[7] KIRK D. NVIDIA CUDA software and GPU parallel computing architecture[C]// International Symposium on Memory Management. [s. l.], 2007.
[8] 孙华锦,高德远,张盛兵. Round robin 调度算法在 FPGA 中的实现[J]. 电子与信息学报, 2003, 25(8): 1143-1147.
[9] 钱博文,李涛,韩俊刚,等. 多态并行处理器中的线程管理器设计[J]. 电子技术应用,2014,39(2):30-32.

作者简介:

贾蕊女,(1991-),硕士研究生.研究方向为数字集成电路. E-mail:545458001@qq.com.
李涛男,(1954-),博士,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.
冯臻夫男,(1982-),博士,讲师.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.
张宏伟男,(1992-),硕士研究生.研究方向为电路与系统.