

引用格式:张宏伟,李涛,冯臻夫,等.机器学习高性能 SIMT 处理器的设计与实现[J].微电子学与计算机,2019,36(9):79-83.
[ZHANG H W,LI T,FENG Z F, et al. Design and implementation of high performance SIMT processor for machine learning
[J]. Microelectronics & Computer,2019,36(9):79-83.]

机器学习高性能 SIMT 处理器的设计与实现

张宏伟¹,李 涛^{1,2},冯臻夫¹,贾 蕊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:针对机器学习中出现的大数据量运算的问题,自主研发了一款高性能 SIMT(Single Instruction Multiple Threads)架构处理器.采用特殊的四级流水线结构,通过可综合的 Verilog HDL 语言对电路进行描述,完成了数据的多线程并行运算.在 Xilinx 公司 VirtexUltraSacle 系列的 xcvu440-flga2892-2-e FPGA 上搭建仿真验证平台对整体电路进行功能验证,结果表明,本设计电路满足多线程并行处理机制.采用 SYNOPSYS 公司 Design-Compile 在 SMIC 65nm CMOS 工艺标准单元库进行综合验证,系统时钟最高工作频率为 370 MHz,系统最大功耗为 4.251 mw.

关键词: SIMT;流水线;多线程;并行运算;FPGA

中图分类号: TP338.6

文献标识码: A

文章编号: 1000—7180(2019)09—0079—05

Design and implementation of high performance SIMT processor for machine learning

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

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

2 School of Computing, Xi'an University of Posts & Telecommunications, Xi'an 710121,China)

Abstract: Aiming at the problem of large data volume computing in machine learning, a high-performance SIMT (Single Instruction Multiple Threads) architecture processor was developed. Using a special four-stage pipeline structure, the circuit is described in a synthesizable Verilog HDL language, and multi-thread parallel computing of data is completed. The simulation verification platform was built on the xcvu440-flga2892-2-e FPGA of XilinxVirtexUltraSacle series to verify the function of the whole circuit. The results show that the design circuit satisfies the multi-thread parallel processing mechanism. The SYNOPSYS Design-Compile is used for comprehensive verification in the SMIC 65 nm CMOS process standard cell library. The maximum operating frequency of the system clock is 370 MHz, and the maximum power consumption of the system is 4.251 mw.

Key words: SIMT; pipeline; multi-threading; parallel computing; FPGA

1 引言

随着科学技术的发展,人工智能^[1]正在应用到生活的各个方面.机器学习作为人工智能的技术基础,通过对大数据的快速训练处理,得到相应算法实

现对数据的分析、预测以及分类,以此来发掘人们感兴趣的领域^[2].由于数据间的独立性,数据的训练过程存在大量并行性运算问题^[3-4],提高这部分运算的效率,将会大幅度提升数据处理的整体速度.

并行运算架构的开发主要包括单指令多数据的 SIMD^[5] (Single Instruction Multiple Data) 架构和

单指令多线程的 SIMT^[6-7]架构两种. SIMD 架构采用单个控制逻辑控制多个处理器来完成单指令多数据的矢量运算,优势在于各处理器之间的数据可以自由通信,但处理过程中需要将线性数据整合成相匹配的向量数据,且无法并行处理分支跳转函数. SIMT 架构由于多线程的处理,对于分支跳转函数可以并行实现,而且不需要对数据进行矢量处理,但是由于线程间的独立性,无法进行数据共享,需要借助共享存储实现.

针对上述问题,本文根据自主研发的指令集以及特殊的 SIMT 架构,设计了一款高性能 SIMT 架构的处理器.通过特殊的四级流水线设计,实现相同指令不同数据的并行化运算;通过运算资源控制器,实现多线程的并行化机制;通过线程状态控制器,实现分支跳转函数的并行化处理.本设计已经在 FP-GA 平台上完成验证.

2 系统整体架构

系统整体架构设计主要包括 1 个簇控制单元 CCU(Cluster Controller Unit,CCU)、4 个线程并行簇(Thread Parallelism Cluster,TPC)、外部存储器 DDR(Double Data Rate SDRAM,DDR)、片上互联以及片上缓存等五部分.其中,每个线程并行簇包括 4 个 SM 组处理器,SM 组处理器主要由 MMU(Memory Management Unite,MMU)存储管理单元、GCU(Group Control Unite,GCU)组控制单元两部分构成. MMU 负责指令和数据的存取;GCU 负责处理器部分设计.系统整体架构如图 1 所示.

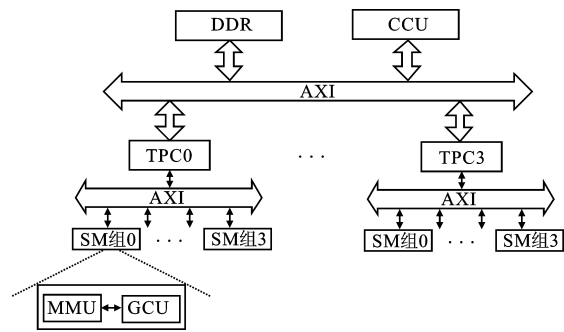


图 1 系统整体架构图

本文主要实现单个 SM 组中 GCU 部分设计.设计过程中将 8 个线程划分为一个 warp^[8],每一个 warp 共享一个指令单元,在外部调度机制控制下,实现不同 warp 切换后,内部多个线程之间的切换支持.在遇到访存时,当前 warp 被切换,对应线程挂起;切换到其它 warp 后,与之对应的线程正常执行,这样做的目的主要是为了隐藏数据访存的延时.通过 8 个 warp 之间的切换,实现 8 个 PE、64 个线程之间的并

行处理,极大提高了处理器数据处理效率.

3 组控制单元设计

组控制单元设计包括 VPU(Vector Processing Unit,VPU)向量处理单元、PE(Processing Element,PE)运算单元、线程状态控制器、运算资源控制器四部分组成. VPU 实现指令的取指、译码以及数据的交互;PE 以及运算资源控制器共同实现不同运算控制模式;线程状态控制器完成多线程的并行化处理机制.设计区别于普通的处理器,主要以实现多线程、并行化的机制为目标.组控制单元整体结构如图 2 所示.

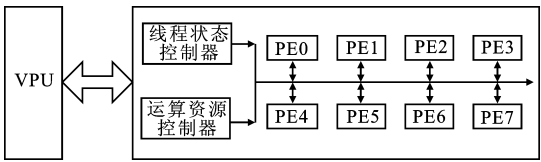


图 2 组控制单元结构图

3.1 指令集

采用自主设计的指令格式,指令长度 24 位,最高 6 位表示操作码,17:12 位表示目的寄存器,11:6 位表示源操作数 1,5:0 位表示源操作数 2 或者访存的偏移地址;对于特殊的跳转类指令,第 17 位表示绝对地址与相对地址的标志;16:0 位表示地址偏移量.指令格式如图 3 所示.

23:18	17:12	11:6	5:0
opcode	RD	RA	RB/imm

图 3 指令格式图

处理器内部包含 42 条相互独立的指令,对于扩展类指令如数据移动中的 LDM、STM 以及超越函数预留有指令格式.具体的指令如表 1 所示.

3.2 流水线结构

处理器采用特殊的四级流水线设计,分别包括取指、译码、执行、回写四个阶段.考虑到本设计中 SIMT 架构的特殊性,内部包含 8 个运算单元,因此取指和译码两级是 8 个 PE 运算单元共享,执行与回写两级在 8 个运算单元之间分别处理.

表 1 指令集

指令类型	操作	数量
定点指令	+、-、*、/、比较	10
浮点指令	+、-、*、/、比较	10
逻辑移位	&、 、~、^、<<、>>、>>>	7
数据移动	MOV、SWAP、SETB、NEG、LDL、STL	6
控制指令	CALL、RETN、JUMP SETM、REVM、ENDM	6
同步	SYNC	1
其他指令	NOOP、HALT	2

取指阶段:在外部调度机制的调度下,选择不同 warp 所提供的 PC 值,进行正常 PC 自加 4 取指令.在内部设计过程中,因为需要支持 8 个 warp 的程序调度机制,所以 8 个对应的初始 PC 值进行分别存放,保证不同 warp 切换过程中 PC 值选择的正确性.

译码阶段:对指令进行分析,将其划分为不同类型,同时将指令对应的 opcode 以及寄存器信息下发到每一个 PE 运算单元.对于不同 warp 的切换,通过内部 FIFO 来保存当前译码的信息.

执行阶段:本设计的特殊之处在于执行级采用 8 个 PE 运算单元,实现多线程的并行化运算处理.通过译码部分的 opcode 以及寄存器号,分别实现不同类型的运算机制.通过不同的线程切换,选取线程对应的寄存器单元,从而取出对应的操作数,进行相应运算.

单个运算单元内部详细电路结构如图 4 所示.根据指令集的特点以及执行周期的多少,将指令进行划分,设计了 7 条算数逻辑流水线:

- 一个定点加减/比较/移位/逻辑运算器;
- 一个定点乘法器;
- 一个定点除法器;
- 一个 32 位浮点加减/比较运算器;
- 一个 32 位浮点乘法器;
- 一个 32 位浮点除法器;
- 一条数据移动转换流水线.

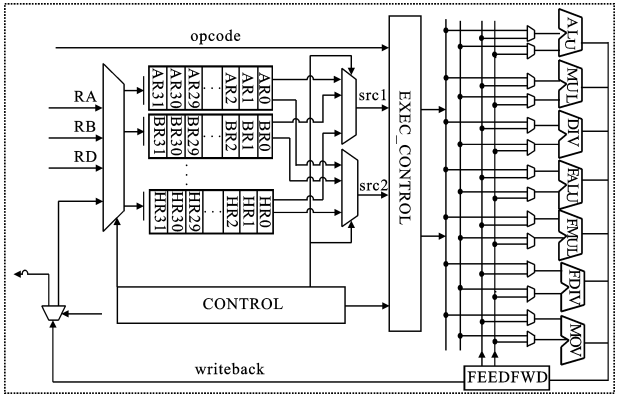


图 4 单个运算单元结构图

通过译码部分的 opcode 以及寄存器号,分别实现不同类型的运算机制.通过不同的线程切换,选取线程对应的寄存器单元,从而取出对应的操作数,进行相应运算.这样做的目的是为了提高电路的利用率,并且对后续电路的可扩展提供支持.本设计的 SIMT 处理器,最大特点是各线程的寄存器是私有的,线程与线程之间无法直接访问数据,必须通过共享存储来完成.本设计最大支持 64 个线程,因此需

要 8 个寄存器文档,分别存放于 8 个 PE 内部,每个寄存器文档包含有 128 个寄存器,依据线程的多少进行动态分配.

为了提高运算效率,对 PE 运算的结果,通过添加前馈机制保证寄存器读写数据的高效性.

回写阶段:包括回写寄存器与回写存储,对执行阶段的运算结果,按照不同线程回写到 PE 所对应的私有寄存器或者共享存储中.

3.3 控制电路设计

控制电路的设计主要是对 8 个 PE 运算单元的控制与管理,保证 8 个 PE 运算单元正常运算.控制电路包括线程状态控制器和运算资源控制器两部分.线程状态控制器是对不同线程之间的控制,运算资源控制器是对不同 PE 资源的使用进行控制.

3.3.1 线程状态控制器

SIMT 架构的优势在于,允许每个线程可以有不同的分支,执行不同的操作,实现并行的条件跳转指令.因此添加线程状态控制器,通过比较指令的比较结果,分别对 8 个 PE 运算单元的执行状态进行控制.

线程状态控制器主要功能是对任务执行中各个线程分支进行管理,在处理应用程序中比较指令时,根据寄存器文件堆下发的相关信号对线程分支的真假状态进行标记或修改,下发 8 个线程的真假信号.

线程状态控制结构如图 5 所示,在相关指令译码后,分别从 8 个线程对应的寄存器中取出 8 个线程的比较结果(ch_thrd_ste0,...ch_thrd_ste7),通过当前比较结果的逻辑值 0 或 1 来表示当前线程的真假;同时,需要对线程的上一个状态 state 和寄存器读取的比较结果 cmp_rslt 进行拼接堆栈;堆栈的同时,对栈的指针 stack_top 进行自加 1 操作.在跳出当前的分支时,需要将上一个状态进行弹栈,同时对栈指针 stack_top 进行自减 1 操作,以保证线程状态和程序运行的正确性.考虑到 8 个线程的同时运算,所以需要使用 8 个栈分别进行存放.在嵌套过程中,对于每一次的弹栈操作,需要考虑到多重分支条件下,线程状态的真假情况.因此,在弹栈过程中需要采用前馈机制,在每一次弹栈过程中考虑到上一次的线程状态,依次类推,直到分支嵌套的顶层,从而实现多线程的并行分支运算.本设计中,对于分支跳转类指令,最多支持 16 层嵌套.采用深度为 16、位宽为 2 的一个寄存器堆作为栈,用于指令嵌套时的保存和恢复.栈中主要保存线程的真假和比较结果,压栈和弹栈时会同时被保存或恢复.

3.3.2 运算资源控制器

运算资源控制器是对 8 个运算单元使用选择、

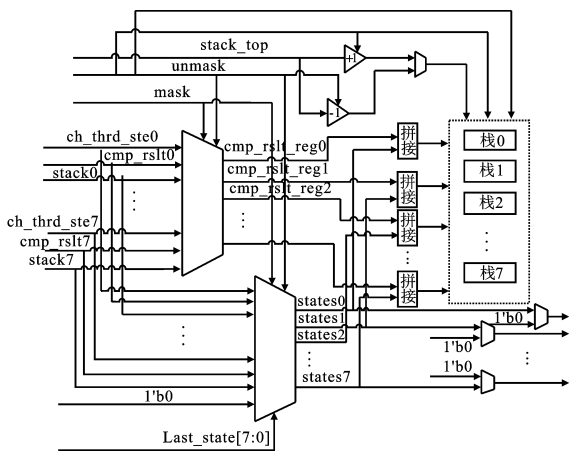


图 5 线程状态控制结构图

数据访存以及异常信息进行处理。根据设计要求,在选择当前 warp 后,需要对当前 warp 所需的 PE 资源进行配置,从而主动选择不同的 PE 资源。在程序执行过程中,通过不同的 PE 资源控制,实现线程的动态分配,最大化的使用 PE 资源。

运算资源控制结构如图 6 所示,由于设计是 8 个运算单元的并行化处理,所以访存需要 8 个通道,分别包括(wr_data、wr_addr、reg_data);根据当前线程的真假状态进行传递。出现运算异常后,需要对异常信息上报,考虑到单指令多线程,当前指令的 PC 相同,所以根据线程的真假选取 PC 值,并对 8 个线程的异常信号 flag_en 进行拼接。在处理浮点除法运算时,考虑到数据的不同导致运算时间不同,无法进行流水线操作,所以需要中断流水线操作;等待 8 个线程全部运算结束后,发送 free 信号结束中断流水线信号,进行正常的流水线操作。

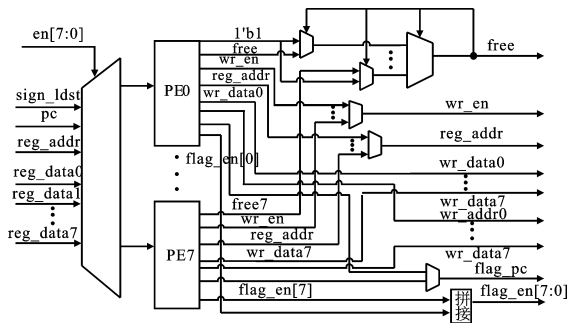


图 6 运算资源控制结构图

定点指令与浮点指令的混合使用过程中,当出现数据相关性时,需要在汇编代码中插入 NOOP 操作,此时如果产生了程序的 warp 切换, PE 运算单元继续运算当前的操作,并且将当前执行的现场信息保存,保证运算结束后可以将结果回写到对应的寄存器位置。对于 warp 的切换, PE

同时切换到不同的线程进行运算,保证数据处理的高效性。

4 验证结果与分析

4.1 FPGA 验证

将组控制单元嵌入整体设计中,包括外围 MMU 单元、AXI 总线以及 DDR,在 Xilinx 公司 VirtexUltraSacle 系列的 xcvu440-flga2892-2-e FPGA 开发板上,搭建仿真验证平台进行功能验证。

验证平台结构如图 7 所示,其中,DW_2x2 为异步转换核,用于频率的匹配;各模块互连采用 AXI4 (2V2)总线;SM 组处理器包括 MMU 存储管理单元和 GCU 组控制单元两部分;程序和数据通过上位机写入 DDR 中。

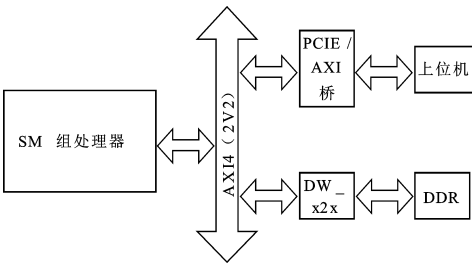


图 7 仿真验证平台结构图

通过编写卷积、矩阵点积、bresenham 画线以及顶点染色等十几个测试程序的汇编代码,对整体设计进行验证。将指令和数据通过上位机写入 FPGA 开发板的 DDR 中,然后处理器从 DDR 中读取指令和数据进行运算,最后将运算结果写回 DDR 中。通过采用 setup debug 抓取相关信号,截取部分仿真信号波形,如图 8 所示。

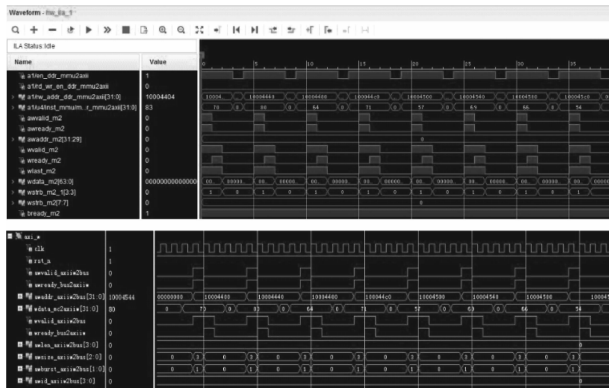


图 8 处理器仿真波形图

仿真结果显示:对于所有的测试用例仿真结果全部正确。

对比本文处理器在不同矩阵阶数下,通过 PE 资源以及线程的选择,统计完成矩阵运算所需要的

时钟数,并计算多线程与单线程运算的性能参数. 处理器运算性能如表 2 所示.

表 2 处理器运算性能表

阶数	PE 资源/线程	时钟数	性能提升
4	1/1	5 483	84.06%
	4/16	874	
8	1/1	40 764	91.49%
	8/64	3 407	
16	1/1	316 779	93.88%
	8/64	19 394	
32	1/1	2 508 480	96.59%
	8/64	78 533	

从表 2 中可以看出,对于不同阶数的矩阵运算,多线程的运算性能提升相对于单线程,性能提升超过 80%,而且随着矩阵阶数的增加,性能提升也在增加.因此,在数据量增大的过程中,通过多线程的并行化处理,更能体现本设计高性能处理的优势.

4.2 综合性能分析

采用 SYNOPSYS 公司 Design-Compile 在 SMIC 65nm CMOS 工艺标准单元库对处理器部分进行综合验证,结果如表 3 所示.

表 3 综合性能表

性能参数	大小
时钟频率	370 MHz
功耗	4.251 mw
面积	0.662 8 mm ²

处理器部分设计占用 FPGA 的资源如表 4 所示.

表 4 处理器 FPGA 资源占用表

模块	占用资源	利用率/%
LUT	61 574/2 532 960	2.43
LUTRAM	720/459 360	0.16
FF	57 704/5 065 920	1.14
DSP	32/2 880	1.11
BUFG	1/1 440	0.07

5 结束语

本文通过整体电路流水线、线程状态控制器以及 PE 资源控制器的设计,实现了处理器对大量数据的并行化、高性能处理;通过搭建 FPGA 仿真验证平台,完成了整体电路的多种程序测试.对比不同阶数矩阵的运算性能情况,结果表明,本设计的多线程处理相对于单线程处理,性能提升超过 80%.因此,对机器学习方面的大数据量处理,本设计处理速度将有很大的提高.本设计电路的可扩展性较强,通过增加超越函数指令和电路,可应用于 DSP 中 FFT

的运算、GPU 中图形的处理以及向量阵列运算等方面.

参考文献:

[1] ABDULHAFIS ABDULAZEEZ OSUWA, ESOSA BLESSING EKHORAGBON, LAI TIAN FAT. Application of artificial intelligence in Internet of Things[C]//2017 9th International Conference on Computational Intelligence and Communication Networks (CICN). Berlin, 2017:169-173.

[2] JINWOOK OH, GYEONGHOON KIM, JUNYOUNG PARK, et al. A simultaneous multi-threading heterogeneous object recognition processor with machine learning based dynamic resource management[J]. IEEE COOL Chips XV, 2012: 1-3. DOI: 10.1109/COOLChips.2012.6216579.

[3] 李涛,杨婷,易学渊,等. 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,2014,36(2): 191-200.

[4] ANKIT BANSAL, PRERNA SAINI. Design and development of low cost multicore processor for parallel computation[C]//2016 2nd International Conference on Communication Control and Intelligent Systems. Mathura, India, IEEE Micro, 2016:18-20.

[5] GUOJUN LI, FAN ZHANG, LIXIANG MA, et al. Accelerating SAR imaging using vector extension on multi-core SIMD CPU[C]//2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Milan, Italy,2015:537-540.

[6] GYUTA EK KYUNG, CHANGMIN JUNG, KWAN-GYEOB LEEG. An implementation of a SIMT architecture-based stream processor[C]//TENCON 2014, 2014 IEEE Region 10 Conference. Bangkok, Thailand,2014:1-5.

[7] NVIDIA. NVIDIA's next generation CUDA compute architecture: fermi-TM, v1.1 [EB/OL]. (2009-09-11). http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf.

[8] 徐元旭,张超,杨兵,等. 一种易实现的 SIMT 调度模型分析[J]. 微电子学与计算机,2014,31(1):25-28.

作者简介:

张宏伟 男,(1992-),硕士研究生.研究方向为电路与系统. E-mail:zhanghongwei9464@163.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.

贾蕊 女,(1991-),硕士研究生.研究方向为电子与通信工程.