

时钟共享多线程处理器存储结构的设计与实现

李卯良¹, 李 涛¹, 刘 欢², 杨 铮², 郭佳乐¹, 李 明¹, 宋晨阳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: 针对时钟共享多线程处理器(SMT_PAAG)中指令和数据存取速率的需求, 提出了一种时钟共享多线程处理器交叉存储结构. 该结构分为数据交叉存储和指令交叉存储, 其中数据交叉存储实现了前向处理单元和多线程处理单元(Processing Element, PE)的数据交互、线程间数据交互以及线程内部数据存取; 指令交叉存储实现了 PE 内部线程的指令存取. 在 FPGA 上实现结果表明, 该交叉存储结构能够满足多线程处理器对于指令和数据存取速率的需求, 最高工作频率可达 518.309 MHz.

关键词: 时钟共享多线程; 处理单元; 交叉存储

中图分类号: TP302

文献标识码: A

文章编号: 1000-7180(2017)01-0110-04

Design and Implementation of Clock Shared a Multithreaded Processor Storage Structure

LI Mao-liang¹, LI Tao¹, LIU Huan², YANG Zheng², GUO Jia-le¹,
LI Ming¹, SONG Chen-yang¹

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

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

Abstract: For clock shared a multithreaded processor (SMT_PAAG) of instructions and data transfer rate indemand, proposes a shared clock multithreaded processor cross storage structure. The structure is composed of data storage and instruction cross storage. The data storage exchange between the front processing unit and the multi thread processing unit (Element Processing, PE) is realized, data interactions between threads, and thread the internal data access. Instruction cross storage to achieve the PE internal thread instruction access. Implemented on the FPGA results show that the cross-storage structures can meet a multithreaded processor for instructions and data transfer rate of the demand, the highest frequency can reach 518.309 MHz.

Key words: clock shared multithreading; processing element ; interleaved

1 引言

随着半导体工艺的进步, 处理器频率越来越高, 同时多核多线程处理器也快速发展. 对于多核多线程处理器来说, 不仅需要解决不同处理器之间的数据交互, 同是还要解决同一个处理器内部不同线程间的数据交互. 如果处理器内部相应的存储结构无法达到相应的速率或是无法在同一周期内进行多条

指令和数据的操作, 那么这必将会影响到处理器进行有效处理的时钟数, 从而降低吞吐量. 存储系统的速率问题已经成为提高处理器性能的主要瓶颈, 学术界与工业界都对目前的存储系统进行了多样的研究^[1], 比如 SUN 研究院的 SPARC 处理器, 每个内核部都是四路线程并发执行, 内部共有 8 个内核, 每个内核都有自己的一级 cache^[2]. Standford 研发的 Hyper-thread 处理器间采用的是私有一级

收稿日期: 2016-04-11 **修回日期:** 2016-06-02

基金项目: 国家自然科学基金(61136002); 陕西省科学技术研究发展计划(2011K06-47); 教育部科学研究计划重点资助项目(2111180)

cache,之间共享 2 级 cache^[3].

本文在多核处理器^[4]的基础上提出了一种与时钟共享多线程处理器相适应的交叉存储结构,与 SPARC 处理器的存储结构相比,本文的存储结构需要完成处理器内部 16 个 PE 之间数据交互,同一 PE 核内部四个线程的同时取指操作和八个线程的数据回写操作. 主要特点是将指令交叉存储结构分为 4 个 bank,能够同时满足四路取指单元对于指令的获取;将数据交叉存储分为 8 个 bank,从而能够有效解决不同多线程处理单元(Processing Element,PE)之间、同一 PE 不同线程间、单个线程与数据存储的交互.

2 交叉存储总体结构

本文采用的时钟共享多线程处理器^[5-6]是由 16 个处理单元(PE)互联而成的一个 4×4 的二维阵列,并且包括一个前向处理器(FEP),两个存储管理器(mem_man),两个调度器(sche)和四个协处理器(ACCL). 该处理器具有 MIMD(Multiple Instruction Stream Multiple Data Stream,MIMD)模式,分布式指令并行模式,和流处理运行模式三种工作模式.

交叉存储设计的整体结构如图 1 所示,分别完成与前向处理单元(Pre_process)、取指译码单元、地址回写单元的数据交互. 其中前向处理单元会进行包头解析后,如果正确,会将多个指令包或数据包发送到相应的地址中或读取相应地址中的数据和指

令,交叉存储需要根据相应的写使能和写地址完成相应的数据和指令的存储与返回. 同时交叉存储还需要根据取指译码单元发送来的读操作数 a 和读操作数 b 的使能和地址,将正确的操作数 a 和操作数 b 发送给取指译码单元,其次还需要根据地址回写单元发送来的写地址和写数据将 PE 内部 8 个不同线程的回写数据进行保存. 因为时钟共享多线程处理器(SMT_PAAG)中每个 PE 内部有 4 个取指单元、4 个译码单元、8 个线程,最多可以同时进行 4 个线程的取指、译码、执行并且可以在不同的线程间进行切换. 所以本设计中将数据存储分为 8 个 bank,指令存储分为 4 个 bank. 交叉存储的整体电路设计如图 2 所示.

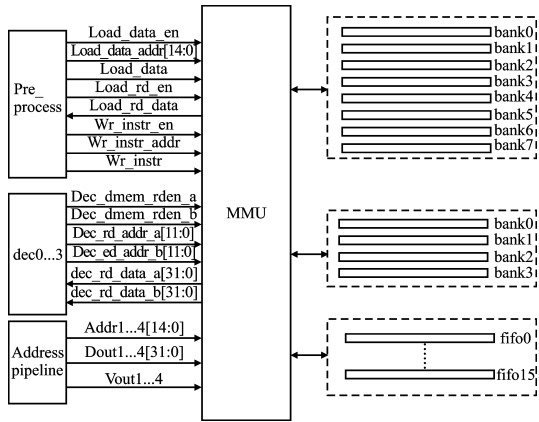


图 1 交叉存储设计的整体结构

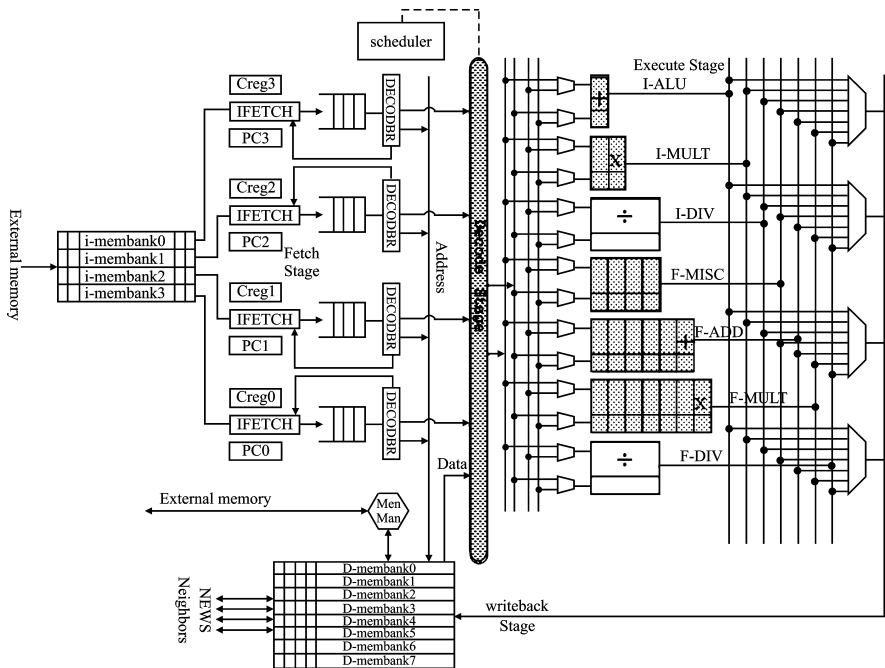


图 2 交叉存储的整体电路设计

3 指令交叉存储的设计与实现

3.1 指令交叉存储的设计

指令交叉存储主要功能:完成前向处理单元对指令的加载和取指单元对于指令的获取. 交叉存储的结构如图 3 所示.

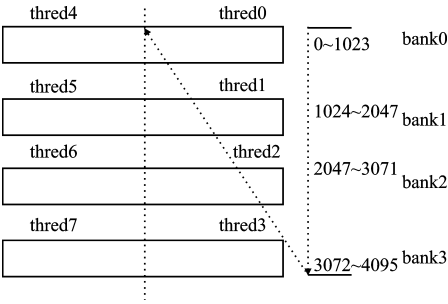


图 3 指令交叉存储的结构

指令交叉存储的地址映射关系是:指令存储总共分为 4 个 bank,每个 bank 的大小为 2k 个 48 bit 存储空间,整个指令存储有 8k 个 48 bit 存储空间,地址采用统一编址的方式,每个 bank 又被划分为高低两部分,分别对应不同线程. 其中 bank0 的低半部分对应线程 0,高半部分对应线程 4;bank1 的低半部分对应线程 1,高半部分对应线程 5;bank2 的低半部分对应线程 2,高半部分对应线程 6;bank3 的低半部分对应线程 3,高半部分对应线程 7. 地址映射方式按照从低线程到高线程逐次映射方式,低线程地址空间填充到 1 k 地址后开始填充下一个线程,直到所有的 bank 填充完毕. 对于取指单元来说,四个 bank 分别对应 PE 的四个取指单元, bank0 对应 ifetch0, bank1 对应 ifetch1, bank2 对应 ifetch2, bank3 对应 ifetch3. 每个 PE 内部有 8 个线程,能够进行最多 4 个线程的同时运行,每一个取指单元对应一个 bank,每个地址的低半部分对应低线程的指令存储空间,高半部分对应高线程的指令存储空间.

3.2 指令交叉存储的实现

指令交叉存储主要完成与前向处理单元和 PE 内部 4 个取指单元的交互,根据前向处理单元的读写使能和地址,将指令存储到相应的地址中,其次根据取指单元的读使能、读地址和相应的地址映射关系,将相应地址对应的指令发送到取指单元,指令交叉存储的实现模块如图 4 所示.

4 数据交叉存储的设计与实现

4.1 数据交叉存储的设计

数据交叉存储实现了前向处理单元与处理单元

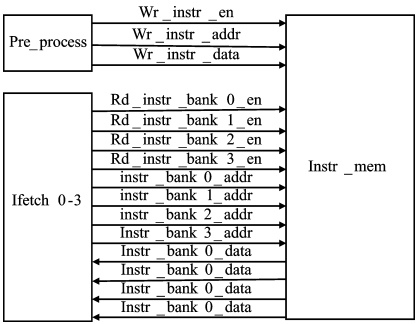


图 4 指令交叉存储的结构

的存储数据交互,线程间数据交互和线程内部数据存储. 数据交叉存储总共分为 8 个 bank,每个 bank 为 4k 个 32bit 存储空间,映射关系如图 5 所示.

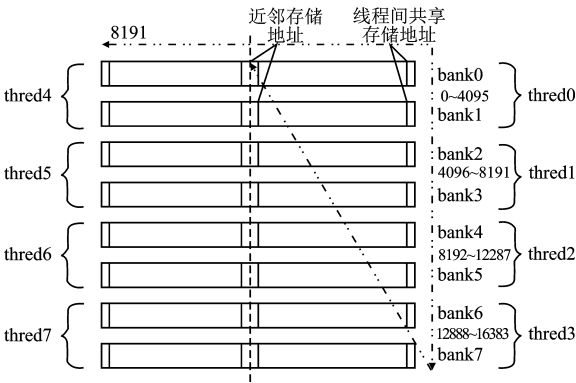


图 5 数据交叉存储的结构

地址映射包括两个部分,一个是写数据部分,地址采用统一编址方式,地址流水线地址映射如图中垂直方向的虚线所示,首先从右边的上方朝下映射到地址 16383 然后转到左边也按照从上到下的方式继续映射;一个是读数据部分,两个 bank 对应一个译码单元,读编址为两个 bank 统一编址,译码单元的读地址映射如图中最上边的虚线所示,从 0 到 8191. 每一个线程对应两个 bank 的低半部分,thread0 对应 bank0 的低半部分和 bank1 的低半部分,thread4 对应 bank0 的高半部分和 bank1 的高半部分,thread1 对应 bank2 的低半部分和 bank3 的低半部分,thread5 对应 bank2 的高半部分和 bank3 的高半部分,thread2 对应 bank4 的低半部分和 bank5 的低半部分,thread6 对应 bank4 的高半部分和 bank5 的高半部分,thread3 对应 bank6 的低半部分和 bank7 的低半部分,thread7 对应 bank6 的高半部分和 bank7 的高半部分. 其中一个线程的低 8 个地址为线程间的共享存储地址,高 4 个地址分别为东西南北四个方向 PE 的近邻存储地址.

4.2 数据交叉存储的实现

如图 6 所示,数据交叉存储具体实现主要包含以下几个模块:rd_ctrl 模块,wr_ctrl 模块,output_sel 模块,以及 rd_wr_blk_reg 模块。

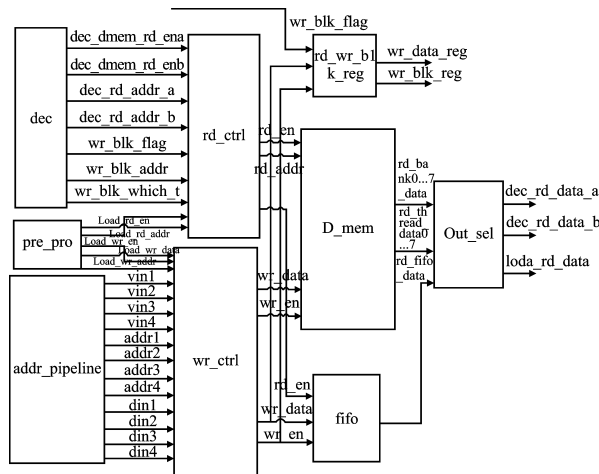


图 6 数据交叉存储的实现模块

主要工作流程:rd_ctrl 模块接收译码的读操作相关信号,并将读控制信号送入 out_sel 模块用来选择译码读取的操作数;wr_ctrl 模块接收地址流水的写控制信号,并将数据写入数据存储。

Rd_ctrl 模块接受分别来自译码单元和前向处理单元的读使能信号和读地址信号,其中来自译码单元的读使能信号和读地址信号包括读操作数 a 使能、读操作数 b 使能、读操作数 a 地址、读操作数 b 地址。读控制模块根据地址映射关系解析出读地址具体对应的存储部分,如果是直接读取线程内部的数据,则可以直接进行读取;如果是读取近邻 PE 间的存储,就要首先判断近邻 PE 的 fifo 是否为空, fifo 不空就可以读取数据,否则阻塞无法读取;如果是读取线程间的数据,首先要进行译码判断是否是读阻塞,如果没有发生阻塞就可以直接读取,否则阻塞无法读取。

Wr_ctrl 模块:该模块接受来自地址流水的写使能信号,和前向处理单元的写使能信号,根据分别解析前向处理单元和地址流水的写地址信号并与相应的写使能控制信号来产生写数据信号。其中来自地址流水线的写使能信号和写地址信号总共有四路。该模块的工作流程是:首先根据相应的数据存储的地址映射,解析相应的写地址是否是写线程内部信号,如果是则直接写如到线程内相应的地址中,否则不写。然后判断是否是写如 PE 间的近邻的存储地址,如果是则判断 fifo 是否为满,如果 fifo 不满,

则将数据直接写入,否则发生阻塞。最后判断是否是写入线程间的存储空间,其中通过地址低位来判断是给哪一个线程间存储空间写数据,高三位为对应的地址号。

rd_wr_blk_reg 模块:主要负责完成线程间数据共享存储是否阻塞的判断。其中寄存器的第[0]位表示是否发生写阻塞,是由译码通过解析指令产生;当从地址流水写线程间的数据存储时,其对应的寄存器的第[1]位为 1,表示写数据有效;当译码读取该地址的数据时,如果寄存器状态为 2b11 时,则读取数据,并将对应寄存器清零,否则为其他状态时,表示读阻塞,必须等待数据写入后开始进行读操作;当为 2b01 时,且写使能有效,则可写数据,否则,写阻塞,必须等待其数据读走后来进行写操作。

output_sel 模块:主要负责完成操作数 a 和操作数 b 的判断和选取。将相关信号送入 output_sel 模块来选择从线程内部存储、线程间共享存储和近邻共享存储空间读取的数据进行判断是输出给译码的读操作数 a 的数据还是操作数 b 的数据。

5 测试结果与分析

本设计通过采用 Mentor 公司的 ModelSim 10.1a 工具完成了功能仿真测试,并使用 xilinx 公司的 vivado15.4 工具对相应的硬件设计电路进行了综合,采用 xilinx 公司的 XC7V2000t-2fhl761 型号的 FPGA 开发板进行了实现。

测试方案:功能仿真主要通过验证交叉存储的地址映射是否正确及能否正确的写入和读取相应地址指令和数据的方式来检验该交叉存储的正确性。

测试结果:同过分析仿真波形,指令交叉存储在写完第一个 bank 的 1k 空间地址后,就开始第二个 bank 的 1k 空间的数据写入;依次写完四个 bank 的低 1k 空间,之后去写其高 1k bank。数据存储同样能够正确的完成相应的地址映射,从而保证地址指令交叉存储的地址映射的正确性。同时对相应的 bank 进行读取操作的结果进行了对比,读取数据与写入数据完全相同,从而保证数据读写功能的正确性。

同时对本设计的资源占用率进行了简单的分析,由于交叉存储的对外接口较多,所以单独综合时会超出开发板的 I/O 资源,但是整个时钟共享多线程处理器的对外接口能够满足资源要求,所以这并不会影响整个系统的实现。交叉存储的资源利用率情况如表 1 所示。

(下转第 118 页)

名称	类型	空	默认值	属性	备注
索引 (3)					
主索引	id			unique	
Index_JSand_User	groupid,code,name,pyName,sex				
Unique_JSand_User	uid,account			unique	
字段 (20)					
id	bigint(20)	否	<auto_increment>		编号
uuid	varchar(32)	否			全局编号
ctime	bigint(20)	否			创建时间
groupid	bigint(20)	否			所属分组
code	varchar(200)	否			分组编码
name	varchar(50)	否			用户名称
pyName	varchar(50)	否			拼音索引
account	varchar(50)	否			登录账户
password	varchar(50)	否			登录密码
sex	tinyint(4)	否			用户性别
tel	varchar(20)	是	<空>		联系电话
imei	varchar(15)	是	<空>		手机号码
qq	varchar(15)	是	<空>		用户QQ
comment	varchar(512)	是	<空>		用户描述
rights	text	是			权限编码
attributes	text	是			扩展属性
loginIP	varchar(20)	否	0.0.0.0		登录IP
loginTime	bigint(20)	否			登录日期
mtime	bigint(20)	否			修改日期
editor	varchar(50)	否			修改用户

图 1 测试图

Annotation 技术实现对数据库物理结构的自动生成,避免了人工对大批的数据表依次重复进行创建的无用工作,使相关数据库管理系统开发人员可以非常便捷、有效、快速地建立数据库的物理结构。

参考文献:

[1] 刘念祖. 数据库设计中数据结构的自动生成[J]. 上海第二工业大学学报, 1992,9(2): 50-54.

[2] 陈艳, 张冬姣. 数据库设计[J]. 福建电脑, 2012, 28(2): 109-110.

[3] 陈金水, 钱昆. 数据库物理结构自动生成技术的研究[J]. 电脑开发与应用, 2005,18(4):42-43.

[4] 刘杰. 数据库结构自动生成工具的开发[J]. 计算机时代, 2007,25(1): 52-54.

[5] 檀明, 许强, 李丛. 基于标注和反射机制的 JavaBean 持久化工具类[J]. 微电子学与计算机, 2010, 27(4): 29-32.

[6] 张文胜. 基于 Annotation 的 Spring AOP 日志处理的设计与实现[J]. 价值工程, 2012, 31(32): 207-209.

[7] 杨宇仙, 杨志勇. 用 Annotation 简化 Java 程序的开发[J]. 计算机与现代化, 2008,24(6): 121-122.

作者简介:

王雪梅 女,(1992-),硕士研究生. 研究方向为计算机技术. E-mail:18794842794@163.com.

李彩虹 女,(1973-),博士,高级实验师. 研究方向为计算机教学模式研究、软件设计.

慕富强 男,(1974-),硕士,教授. 研究方向为云计算、无线传感器、人工智能、智能家居.

李 廉 男,(1951-),博士,教授. 研究方向为网格计算、高性能计算及其在 e-Science 系统中的应用、无线传感器网络,以及行为与情感分析、健康关怀等.

(上接第 113 页)

表 1 FPGA 芯片资源使用报告表

逻辑器件	资源利用率
Slice Registers	25 700/2443 200(1%)
Slice LUTs	42 341/122 1600(3%)
LUT-FF pairs	11 185/56 856(19%)

6 结束语

本文通过对时钟共享多线程处理器的研究,采用了一种与其相适应的存储结构,通过对整个处理器的运行分析发现,将数据存储分为 8 个 bank,指令存储分为 4 个 bank,能够有效地解决 PE 内部四路线程取指和八路线程回写以及不同 PE 间数据交互问题,最终的综合结果表明交叉存储的最高频率是 518.309 MHz. 所以该结构能够保证时钟共享多线程处理器对指令和数据存储访存速率的需求,并有效提高了处理器的速率。

参考文献:

[1] 屈文新, 樊晓桢, 张盛兵. 多核多线程处理器存储技术研究进展[J]. 计算机科学, 2007, 34(4):13-16.

[2] Spracklen L, Abraham S G. Chip multithreading: op-

portunities and challenges[C]// International Symposium on High-performance Computer Architecture, Austin, 2005:248-252.

[3] Hammond L, Hubbert B A, Siu M, et al. The Stanford Hydra CMP[J]. Micro IEEE, 2000, 20(2): 71-84.

[4] Li T, Xiao L, Huang H, et al. PAAG: A polymorphic aArray architecture forgraphics and image processing[C]// International Symposium on Parallel Architectures, IEEE, 2012:242-249.

[5] 刘应天. 时钟共享多线程处理单元的设计与实现[D]. 西安:西安邮电大学, 2015.

[6] 雷晓锋, 李涛. 时钟共享多线程处理器通信机制的设计与实现[J]. 电子技术应用, 2016, 42(3):42-46.

作者简介:

李卯良 男,(1991-),硕士研究生. 研究方向为电路与系统. E-mail:limaoliang_lml@163.com.

李 涛 男,(1954-),教授. 研究方向为计算机体系结构、计算机图形学.

刘 欢 男,(1991-),硕士研究生. 研究方向为电路与系统.

杨 铮 男,(1991-),硕士研究生. 研究方向为电路与系统.

郭佳乐 女,(1992-),硕士研究生. 研究方向为电路与系统.