

DMR: 兼容 RISC-V 架构的乱序超标量通用处理器核

孙彩霞 郑 重 邓 全 隋兵才 王永文 倪晓强
(国防科技大学计算机学院 长沙 410073)
(cxsun@nudt.edu.cn)

DMR: An Out-of-Order Superscalar General-Purpose CPU Core Based on RISC-V

Sun Caixia, Zheng Zhong, Deng Quan, Sui Bingcai, Wang Yongwen, and Ni Xiaoqiang
(College of Computer Science and Technology, National University of Defense Technology, Changsha 410073)

Abstract DMR is a RISC-V based out-of-order superscalar general-purpose CPU core from the College of Computer Science and Technology, National University of Defense Technology. Three privilege levels, user-mode, supervisor-mode and machine-mode, are all supported, and the standard RISC-V RV64G instruction set is implemented. In addition, custom vector instructions are extended in DMR. Sv39 and Sv48 are supported for the virtual-memory system, and the size of physical address is 44-bit. The pipeline for single-cycle integer instructions is 12-stage in all. All instructions are executed out of program order and committed in program order. More than four instructions can be issued per cycle. Distributed schedule queues are used and at most 9 instructions can be out-of-order scheduled for executions in one cycle. Multi-layer, multi-platform functional verification method driven by functional coverage is used, and Linux OS is already booted on FPGA prototype system. DMR reaches 5.12 CoreMark/MHz and targets 2 GHz clock speed in 14 nm technology.

Key words RISC-V; out-of-order; superscalar; CPU core; general-purpose CPU

摘 要 DMR 是由国防科技大学计算机学院自研的一款兼容 RISC-V 架构的乱序超标量通用处理器核,支持用户态(user-mode)、特权态(supervisor-mode)和机器态(machine-mode)三种特权级模式,兼容 RV64G 指令集规范,并进行了自定义向量扩展,虚存系统支持 Sv39 和 Sv48,物理地址为 44 b.DMR 的单周期整数流水线为 12 级,指令乱序发射、顺序提交,指令发射宽度为 4,实现了多个分布式调度队列,每拍最多可乱序调度 9 条指令执行.DMR 采用覆盖率驱动的多层次、多平台的功能验证方法,已经在 FPGA 原型系统下成功启动 Linux OS,CoreMark 分数为 5.12 MHz,在 14 nm 工艺下主频可达到 2 GHz.

关键词 RISC-V; 乱序; 超标量; 处理器核; 通用 CPU

中图法分类号 TP303

DMR 是由国防科技大学计算机学院自研的一款兼容 RISC-V 架构的乱序超标量通用处理器核,主要面向高性能计算领域。DMR 支持用户态(user-mode)、特权态(sup-

ervisor-mode)和机器态(machine-mode)三种特权级模式,兼容 RV64G 指令集规范^[1],并进行了自定义向量扩展,虚存系统支持 Sv39 和 Sv48^[2],物理地址为 44 b.

1 微架构

DMR 的微架构如图 1 所示.取指宽度为 256 b,即 8 条 32 b 指令;译码宽度、寄存器重命名宽度和指令分派宽度都是 4;寄存器重命名采用统一的物理寄存器文件方式,如果没有足够的空闲物理寄存器,重

命名过程将会发生阻塞;分派后的指令根据指令类型进入相应的指令调度队列,同时也会进入重定序缓冲(reorder buffer, ROB);采用分布式调度队列,根据指令类型设置了整数调度队列、访存调度队列和浮点调度队列;调度队列中的指令就绪后,就会被乱序调度执行,每拍最多可以调度 9 条指令,其中 3 条整数指令、1 条分支指令、2 条 load 指令、1 条 store 指令和 2 条浮点指令;指令被调度执行时读取寄存器文件获取源操作数,源操作数也可能来自旁路的数据;指令提交按序进行;指令 Cache 和数据 Cache 均为 64 KB,4 路组相联,Cache 行大小均为 64 B.

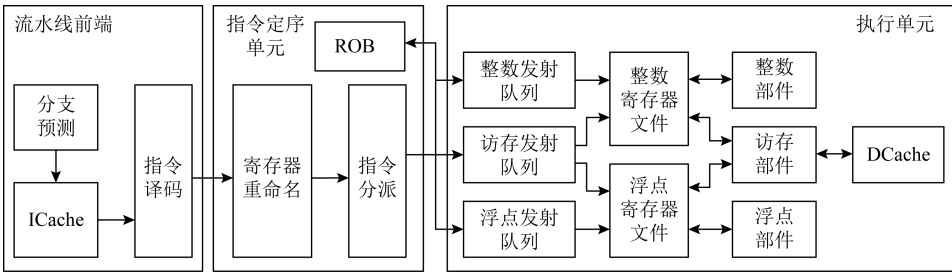


Fig. 1 DMR microarchitecture
图 1 DMR 的微体系结构

2 流水线

DMR 的流水线如图 2 所示.单周期整数流水线共有 12 级,取指 4 拍,然后是 2 拍的译码.第 1 拍进行预译码并处理指令拆分和指令融合,第 2 拍译码

出指令中的操作数信息,以供寄存器重命名时使用;REN 为重命名站,DS 为指令分派站,根据指令类型,将指令顺序分派到相应的指令调度队列;ISS 为指令发射站,发射后进入 RF 站,读取寄存器文件,读出数据和旁路数据进行选择后送到执行站 EX,执行结果在 WB 站被写回到寄存器文件.

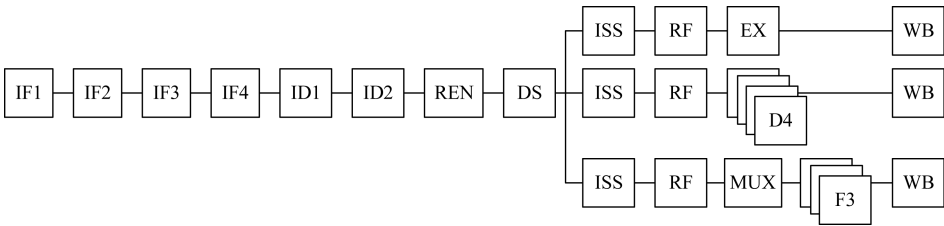


Fig. 2 DMR pipeline
图 2 DMR 的流水线

数据 Cache 命中时的 Load-to-use 延时是 4 拍;为了实现较高主频,浮点流水线在读寄存器文件之后增加了一拍 MUX 用于数据选择.

3 分支预测

DMR 采用 TAGE (tagged geometric history

length)^[3-4]算法预测分支方向,所实现的 TAGE 结构包含 5 个组件(component),即除了基本预测器组件外,还包含 4 个具有标记的用不同历史长度生成索引的预测组件;2K 项的 BTB(branch target buffer)、48 项的 RSB(return stack buffer)和 512 项的 IPB(indirect prediction buffer)分别被用于预测不同类型的分支的目标地址.

4 指令拆分和指令融合

DMR 在译码阶段将整数和浮点之间的转换指令拆分成 2 个内部操作,转换操作在浮点单元完成,而读取整数寄存器文件或写入整数寄存器文件的操作在访存部件完成,使得浮点执行单元不需要读写整数寄存器文件,从而可以减少整数寄存器文件的读写端口数目、简化整数数据旁路网络的设计,同时也有利于物理实现.

DMR 在指令译码阶段将某些指令组合融合成一条指令,然后进行重命名、发射和执行,从而提高指令实际发射宽度,并且可以减少指令占用的乱序执行资源数目和降低调度开销^[5].

根据 RISC-V 架构手册的描述^[1],结合 DMR 微架构的特点,DMR 实现了指令组合的融合,如表 1 所示.auipc 指令和 load 指令融合,可以实现 PC (program counter) 相对的 32 b 偏移寻址的数据加载,auipc 指令和 jalr 指令融合,可以实现 PC 相对的 32 b 偏移的分支跳转.

Table 1 Instruction Fusion in DMR
表 1 DMR 中的指令融合组合

组合	指令融合
组合 1	auipc rd, imm20; addi rd, rd, imm12
组合 2	auipc rd, imm20; l{b h w d} rd, imm12(rd)
组合 3	auipc rd, imm20; jalr rd, imm12(rd)
组合 4	lui rd, imm20; addi rd, rd, imm12
组合 5	lui rd, imm20; l{b h w d} rd, imm12(rd)
组合 6	lui rd, imm20; jalr rd, imm12(rd)

5 处理器状态的恢复

当发生异常或分支误预测、需要清除前瞻执行的指令时,被清除指令对处理器有关状态的影响同样需要被清除,以恢复到前瞻指令未执行之前的状态.前瞻寄存器重命名映射表就是需要被恢复的处理器状态之一.

DMR 维护了 2 个寄存器重命名映射表:前瞻映射表和体系结构映射表.指令重命名时更新前瞻映射表,指令提交时更新体系结构映射表.DMR 在指令提交时才报告该指令触发的异常,所以发生异常时,异常指令之前的所有指令都已经完成对体系结构映射表的修改,直接使用体系结构映射表恢复

前瞻映射表即可.而分支误预测一旦发生,需要立即清除后续前瞻执行的指令,这时前瞻映射表一般通过重建的方式进行恢复,重建完成前不能进行寄存器的重命名,从而可能造成流水线停顿.DMR 在分支指令进行重命名时,会对当前的前瞻映射表进行备份,分支误预测发生时使用该分支对应的备份数据对前瞻映射表进行快速恢复,避免重命名映射表重建导致的流水线停顿.

6 自定义向量扩展

DMR 面向高性能计算进行了浮点向量的自定义扩展,该扩展中浮点向量和浮点标量共用一套体系结构寄存器,标量占用寄存器的低位部分.如图 3 所示,其中 D 表示双精度浮点,S 表示单精度浮点.

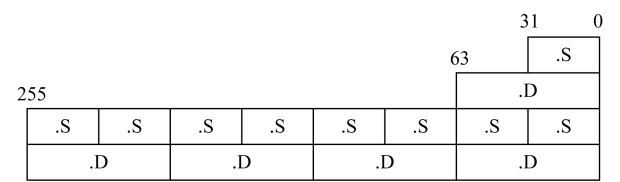


Fig. 3 Vector formats
图 3 向量格式

除了基本的单、双精度浮点计算操作和访存操作外,自定义扩展指令集还支持单、双精度浮点向量乘加操作以及 gather load/scatter store 操作.

向量长度支持动态配置为 128 b 或 256 b.DMR 的浮点执行单元实现了 2 条 256 b 流水线,双精度浮点峰值运算性能可达到每个时钟周期 16 个操作.

7 功能验证和性能结果

DMR 采用覆盖率驱动的多层次、多平台功能验证方法,如图 4 所示.

验证分为 3 个层次:单元级、核级和系统级.在单元级,针对不同的功能单元搭建了基于 UVM (universal verification methodology) 的软模拟平台和形式化验证平台,在核级,搭建了基于 trace 实时对比的软模拟平台,在系统级搭建了硬件仿真平台.对单元级的软模拟平台和形式化验证平台收集的覆盖率以及核级软模拟平台收集的覆盖率进行合并,统一管理.

DMR 已经在 FPGA 原型系统下成功启动 Linux OS,CoreMark 分数为 5.12 MHz,在 14 nm 工艺下主频可达到 2 GHz.

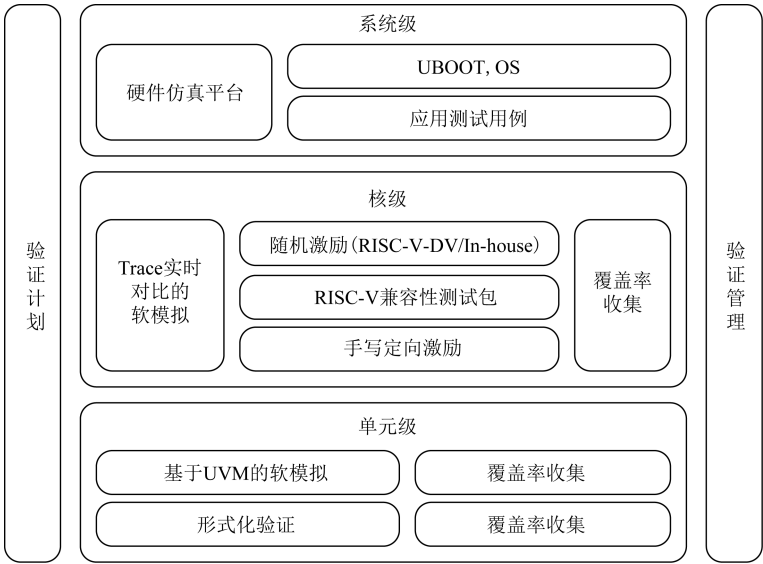


Fig. 4 Functional verification
图 4 功能验证方法

参 考 文 献

[1] RISC-V International. The RISC-V Instruction Set Manual Volume I, Unprivileged ISA, v20191214 [OL]. [2021-02-05]. <http://riscv.org>

[2] RISC-V International. The RISC-V Instruction Set Manual Volume II: Privileged Architecture, v1.12 [OL]. [2021-02-05]. <http://riscv.org>

[3] Seznec A. A new case for the TAGE branch predictor [C] // Proc of the 44th Annual IEEE/ACM Int Symp on Micro-architecture. Piscataway, NJ: IEEE, 2011: 117-227

[4] The Journal of Instruction-Level Parallelism. TAGE-SC-L Branch Predictor [OL]. [2020-01-05]. <http://www.jilp.org/cbp2014>

[5] AnandTech. Intel Core versus AMD’s K8 architecture [OL]. [2021-02-27]. <http://www.anandtech.com>



Sun Caixia, born in 1979. PhD, associate professor. Her main research interests include processor architecture and micro-processor design.

孙彩霞, 1979 年生, 博士, 副研究员, 主要研究方向为处理器体系结构和微处理器设计。



Zheng Zhong, born in 1986. PhD, assistant professor. His main research interests include processor microarchitecture and high performance computing.

郑 重, 1986 年生, 博士, 助理研究员, 主要研究方向为处理器微体系结构和高性能计算。



Deng Quan, born in 1989. PhD, assistant professor. His main research interests include processing in memory and non-volatile memory.

邓 全, 1989 年生, 博士, 助理研究员, 主要研究方向为存内计算和非易失性存储器。



Sui Bingcai, born in 1981. PhD, associate professor. His main research interests include micro-processor architecture, design and optimization.

隋兵才, 1981 年生, 博士, 副研究员, 主要研究方向为微处理器体系结构及其设计与优化。



Wang Yongwen, born in 1977. PhD, professor. Senior member of CCF. His main research interests include processor microarchitecture and high performance computing.

王永文, 1977 年生, 博士, 研究员, CCF 高级会员, 主要研究方向为处理器微体系结构和高性能计算。



Ni Xiaoqiang, born in 1974. PhD, associate professor. Member of CCF. His main research interests include processor architecture, design and verification.

倪晓强, 1974 年生, 博士, 副研究员, CCF 会员, 主要研究方向为处理器体系结构及其设计与验证。