

DOI:10.3969/j.issn.1003-5060.2026.06.006

基于 OpenLA500 的分支预测器分析与优化

贺军义, 袁东亮, 刘 帅, 张 敏, 王 磊, 李佳航

(河南理工大学 计算机科学与技术学院, 河南 焦作 454000)

摘 要:嵌入式微处理器为获得更高的处理器性能使用了许多并行技术,指令并行度的提高使得高精度的分支预测器变得越来越重要。文章以基于龙芯指令集 LoongArch 设计的 OpenLA500 处理器核心为平台,分别采用 Gshare 预测器和混合预测器 2 种方式扩大分支预测表项,提高分支历史信息的利用率,并对分支指令地址进行异或处理来降低分支别名的概率,从而提高原有处理器的分支预测能力。使用包括 CoreMark 在内的 10 项测试程序对优化前、后的处理器进行仿真验证,并在 Xilinx Artix-7 现场可编程门阵列(field-programmable gate array, FPGA)上完成原型验证。实验结果表明, Gshare 分支预测器和混合预测器平均预测准确率分别达到 91.87% 和 93.23%, 相较优化前分别提高了 3.22% 和 4.58%。

关键词: LoongArch 指令集; 分支预测器; 别名问题; 现场可编程门阵列(FPGA)

中图分类号: TP332.1

文献标志码: A

文章编号: 1003-5060(2026)06-0758-06

Analysis and optimization of branch predictor based on OpenLA500

HE Junyi, YUAN Dongliang, LIU Shuai, ZHANG Min, WANG Lei, LI Jiahang

(School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China)

Abstract: Embedded microprocessors use many parallel techniques to achieve higher processor performance, and the increase in instruction parallelism makes high-precision branch predictors increasingly important. This paper takes the OpenLA500 processor core based on the LoongArch instruction set as the platform and adopts two different methods for branch prediction: Gshare predictor and hybrid predictor. By expanding the branch prediction entries, improving the utilization of branch history information, and using XOR processing for branch instruction addresses to reduce the probability of branch aliasing, the branch prediction capability of the original processor is enhanced. The optimized processors were simulated and verified using ten test programs, including CoreMark, and the prototype validation was carried out on Xilinx Artix-7 field-programmable gate array(FPGA). The experimental results show that the average prediction accuracy of the Gshare branch predictor and the hybrid predictor reached 91.87% and 93.23%, respectively, representing improvements of 3.22% and 4.58% over the original branch predictor.

Key words: LoongArch instruction set; branch predictor; destructive aliasing; field-programmable gate array(FPGA)

0 引 言

在提高处理器性能的需求推动下,现代处理器设计越来越侧重于增强指令的并行性^[1]。在此

背景下,超标量和多级流水线技术的广泛应用提升了指令的并行度和吞吐量,但同时这些技术也带来了新问题,如分支延迟,从而增加了分支指令的执行成本^[1-2]。为解决分支延迟问题,引入了专

收稿日期:2024-04-12;修回日期:2024-06-11

基金项目:国家自然科学基金资助项目(61772159;61872126);河南省科技攻关资助项目(242102210185);河南省高校重点研究基金资助项目(20A520015);河南省高校基本科研业务费专项资金资助项目(NSFRF240614)和河南理工大学博士基金资助项目(60907023)

作者简介:贺军义(1982—),男,河南平舆人,博士,河南理工大学副教授,硕士生导师,通信作者,E-mail:hejunyi@hpu.edu.cn;
王 磊(1977—),男,辽宁北票人,博士,河南理工大学教授,硕士生导师。

门的硬件进行分支预测,以预测条件分支的方向和结果^[3],然而,错误的分支预测会造成清空流水线的延迟,浪费处理器资源,对处理器性能造成影响^[4]。因此,对于高性能处理器来说,具有高预测精度的分支预测器(branch predictor, BP)就显得尤为重要。

在不同的指令集体系结构和处理器架构下,分支预测的实现方式也有所不同^[5]。随着处理器的发展,根据不同的处理器架构,出现了众多可行的预测方案,但为了最大化预测性能,需要对这些预测方案进行自适应调整和优化。

本文研究对象 OpenLA500 是龙芯中科基于自主指令集 LoongArch 设计并开源的一款处理器核心^[6],其分支预测器采用 32 组,每组 4 个寄存器,用于缓存指令流中的分支指令 PC (program counter)、分支指令目的地址和该分支指令的分支状态。分支状态使用两位饱和计数器记录,对于每一个已跳转的分支指令 PC,对应的计数器值增加 1,反之,对应的计数器值被减去 1^[7]。但由于寄存器数量有限,导致在进行分支预测时不能很好地跟踪足够多的分支指令历史信息,同时也会产生一定的硬件开销,因此,在许多情况下分支预测精度并不理想。为改善这一问题,本文在 OpenLA500 的软硬件平台上,通过采用 Gshare 预测器和混合预测器扩充原先的分支预测表项,并优化取指 PC 以降低分支别名(Branch Aliasing)问题发生的概率,成功提高了 OpenLA500 分支预测器的准确率。

1 OpenLA500 及其分支预测单元

1.1 LoongArch32 指令集架构

LoongArch 是龙芯中科设计的一种基于精简指令集计算(reduced instruction set computing, RISC)原理的指令集架构,继承了精简指令集的设计传统,其指令长度固定,编码格式规整,并且大多数指令为三操作数,仅有访存指令可以访问内存^[8-9]。这种设计降低了指令集和硬件的复杂性,有利于流水线实现。OpenLA500 是龙芯中科针对教育领域设计并开源的一个处理器核心,其实现的指令集是基于 LoongArch32 的精简版(简称 LA32R)^[10]的基础整数指令。

1.1.1 条件跳转指令

OpenLA500 中实现的条件跳转指令共有 6 个,即 BEQ、BNE、BLT、BGE、BLTU、BGEU,它们的指令格式^[10]如图 1 所示。

	31	26	25		10	9	5	4	0
BEQ	010110	Offset[15:0]					rj		rd
BNE	010111	Offset[15:0]					rj		rd
BLT	011000	Offset[15:0]					rj		rd
BGE	011001	Offset[15:0]					rj		rd
BLTU	011010	Offset[15:0]					rj		rd
BGEU	011011	Offset[15:0]					rj		rd

图 1 LoongArch 条件跳转指令格式

条件跳转指令的跳转方向根据寄存器 rj 和 rd 中的数值大小关系确定。其中:BEQ 和 BNE 指令通过比较 2 个寄存器的值是否相等来判断是否跳转;BLT 和 BLTU 指令仅在 rj 的值小于 rd 的值时才会发生跳转;而 BGE 和 BGEU 指令仅在 rj 的值大于等于 rd 的值时才会发生跳转。条件跳转指令的目标地址由指令 PC 和偏移量共同决定,计算方式为 $Target = PC + SignExtend(\{offs16, 2'b0\}, GRLEN)$ 。这里的 SignExtend 表示符号扩展,而 GRLEN 表示通用寄存器(general-purpose register, GR)的位宽。

1.1.2 无条件跳转指令

在 LA32R 指令集中,无条件跳转指令共有 3 个,分别是 B、BL、JIRL,其指令编码格式^[10]如图 2 所示。

	31	26	25		10	9	5	4	0
B	010100	Offset[15:0]					Offset[25:16]		
BL	010101	Offset[15:0]					Offset[25:16]		
JIRL	010011	Offset[15:0]					rj		rd

图 2 无条件跳转指令格式

无条件跳转指令在执行时一定会发生跳转,但是不同的无条件跳转指令,其跳转目标地址的计算方式是不同的。对于 B 和 BL 指令,其目的地址计算方式为 $Target = PC + SignExtend(\{offs26, 2'b0\}, GRLEN)$,与 B 指令不同,BL 指令还需要将当前 PC 值加上 4 的结果写入到通用寄存器 r1 中。而对于 JIRL 指令,则需要将当前 PC 值加上 4 写入到通用寄存器 rd 中,其目的地址计算方式为 $Target = GR[rj] + SignExtend(\{offs16, 2'b0\}, GRLEN)$,通用寄存器 GR 的值记作 $GR[index]$ 。

1.2 OpenLA500 的分支预测器

OpenLA500 在经典的五级流水处理器的基础上增加了 Cache 模块以及分支预测模块,对取

2.1 Gshare 分支预测器的实现

Gshare 分支预测器使用一个 N 位的寄存器 GHR(global history register)来记录最近 N 次分支指令的执行结果,使用一组 K 位的饱和计数器组作为模式历史表(pattern history table, PHT),PHT 的每个表项存储了一种历史模式的预测结果,每个表项的序号由分支指令地址与 GHR 进行异或操作得到。全局分支预测利用分支指令之间的联系,使用全局分支历史,可以实现基于第 1 个分支的执行方向来预测第 2 个分支的功能^[11-13]。

受硬件资源限制,PHT 表项的个数是有限的。它使用有限的表项对所有的分支指令进行追踪,因此,不可避免地存在分支别名问题。本文实现的 Gshare 分支预测器并未直接使用分支指令的部分地址与 GHR 值进行异或,而是在缓存到 fetch_pc_r 后,将 fetch_pc_r 的低 24 位与高 24 位进行异或,然后拼接上 fetch_pc_r 的高 8 位作为新的 PC。这种方法可以在较大程度上降低分支别名问题,提高 Gshare 预测的精度。

本文实现的 Gshare 包含一个 10 位的 GHR 和 1 024 个两位饱和计数器的 PHT。分支指令在译码级时进行 PHT 的更新。另外,为了提供分支预测的跳转地址,本文实现了一个分支目标缓冲区(branch target buffer, BTB)^[14],包含 256 个成对的分支地址和目标地址。将 PC[11:2]与 GHR 进行异或后的结果作为 PHT 的索引,预测结果为 PHT 表项的高位。若为 1,则预测为跳转,否则不跳转。GHR 在译码级时进行更新,此时指令执行状态已知,可以及时记录最近的分支历史。本文实现的 Gshare 分支预测器如图 4 所示。

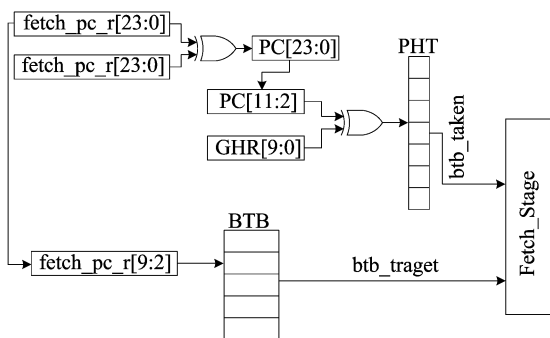


图4 Gshare 分支预测器

2.2 混合分支预测器的实现

本文中的混合预测器使用 2 个预测器,即基

于局部历史的分支预测器(Local History Predictor)和 Gshare 预测器。基于局部历史的分支预测器在偏向跳转的循环分支(Loop Branches)中有较好的预测精度;而 Gshare 预测器在依赖分支指令(Dependent Branch Instructions)中有较好的预测精度。通过在 Gshare 分支预测器中添加局部分支预测,可以提高处理器在 Gshare 预测器训练阶段的预测精度^[5,15-16],本文中实现的混合预测器结构如图 5 所示。

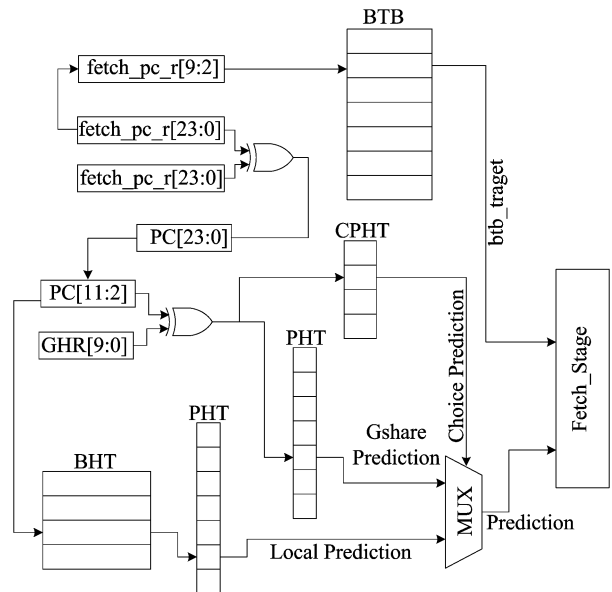


图5 混合分支预测器

混合预测器由互相独立的 Gshare 预测器和局部历史分支预测器组成,预测结果由一个 2 路选择器得到。局部历史分支预测器由 1 个分支历史表(branch history table, BHT)和 1 个 PHT 组成,BHT 共 256 个表项,其中每个表项有 10 bit,记录一个分支指令的历史执行情况,索引使用异或后 PC 的低 8 位。局部历史分支预测器的 PHT 是一个有 1 024 个表项的两位计数器,其索引地址由 PC 对应的 BHT 表项得到。

2 路选择器的控制信号由 CPHT(choice pattern history table)得到,CPHT 类似于 PHT,也是由两位饱和计数器组成,当其中一个分支预测方法 2 次预测失败,而另一种分支预测方法 2 次预测成功时,会使状态机转到使用另一个分支预测方法的状态。

对于这种混合分支预测,每条分支指令经过一段训练时间,都会根据 2 种分支预测方法的执行情况选择 2 种分支预测中的 1 种作为自己的分支预测结果。

3 实验结果

本文使用 OpenLA500 平台在 Xilinx 公司的 Artix-7 芯片上执行包括 CoreMark 和 Dhrystone 在内的一系列测试程序,通过记录测试程序执行过程中条件分支指令和指令预测错误的次数来表现分支预测器的性能。测试程序简介见表 2 所列。

表 2 测试程序描述

测试项	描述
bitcount	位计数函数
bubble-sort	使用冒泡排序算法对 1 000 个无序整数排序
coremark	嵌入式系统中 CPU 测试程序
crc32	循环冗余码校验测试程序
dhrystone	测试处理器的整数运算和逻辑运算的性能
quick-sort	使用快速排序算法对 1 000 个无序整数排序
select-sort	使用选择排序算法对 1 000 个无序整数排序
sha	sha 哈希计算一个字符串
stream-copy	复制 1 000 个整数到另一个数组
stringsearch	使用 BMH 算法进行字符串匹配

实验中使用 OpenLA500 原先的分支预测器、Gshare 分支预测器和混合预测器分别运行测试程序并进行仿真,结果如图 6 所示。

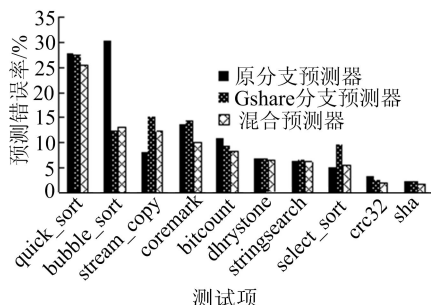


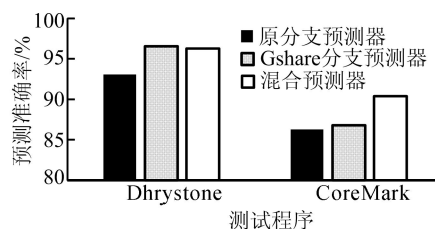
图 6 3 种分支预测器 1 轮测试的结果

根据图 6 显示的分支预测错误率,相较于原分支预测器,本文实现的 Gshare 分支预测器的平均预测错误率降低了 0.91%,混合预测器的平均预测错误率降低了 2.47%。在运行 stream-copy 测试时,原分支预测器的预测错误率低于本文 2 种预测器的预测错误率。经研究测试程序发现,在部分测试程序中,分支跳转指令数量较少,2 种分支预测器中的 PHT 表项状态未达到饱和状态,导致分支指令的预测结果不够稳定,从而造成较高的分支预测错误率。

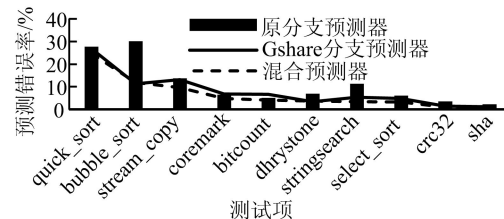
为了进一步评估本文所提出的 2 种分支预测器的有效性,增加了测试程序的执行轮数,以增加

分支指令数量,在 PHT 的各表项进入饱和状态后记录分支预测器的预测情况。

3 种分支预测器 10 轮测试的结果如图 7 所示。从图 7 可以看出,在 OpenLA500 平台上运行 10 轮测试程序后,原分支预测器的各项预测错误率波动不大,平均预测错误率相较于 1 轮测试降低了 0.23%。这是由于原分支预测器的分支预测表项非常有限,导致程序较长的时间内未能利用到分支的历史信息。本文实现的 Gshare 分支预测器和混合预测器在运行 Dhrystone 和 CoreMark 测试程序时,预测准确率取得了较大提升,分别达到 96.28% 和 90.39%。Gshare 分支预测器的平均预测错误率相较之前降低了 3.22%,而混合预测器的平均预测错误率相较之前降低了 4.58%。因此,本文实现的预测器相较于原先的准确度有大幅提高。



(a) Dhrystone和CoreMark的10轮测试结果



(b) 10轮测试的结果对比

图 7 3 种分支预测器 10 轮测试的结果

本文使用 Vivado 2019.2 工具基于 Artix-7 FPGA 芯片参数,对原分支预测器、Gshare 分支预测器和混合预测器分别进行综合,得到其功耗、时序和资源占用的相关数据,其中资源占用率 (Utilization Rate) 如图 8 所示。

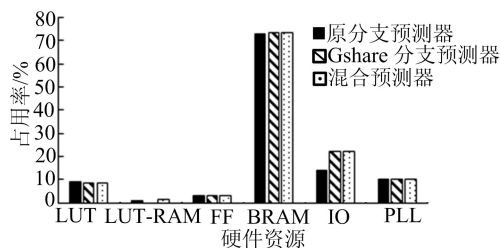


图 8 3 种分支预测器 FPGA 资源占用情况

从图8可以看出:Gshare分支预测器和混合预测器在资源占用方面表现相似,与原分支预测器相比,IO资源占用上提高了8.0%,其他资源基本保持不变;Gshare分支预测器和混合预测器的片上功耗比原分支预测器增加了1.2%,WNS(worst negative slack)比原分支预测器分别增加了0.071 ns和0.076 ns。整体来看,资源的消耗和时序的降低在可接受的范围。

4 结 论

本文研究了OpenLA500处理器的分支预测模块,着重探讨提升条件分支跳转方向的预测精度及分支预测精度对处理器性能的影响;借鉴Gshare预测器的理念,提出了新的Gshare分支预测器和混合预测器,以改进OpenLA500的分支预测模块。通过增加分支预测表项的数量,利用取指PC的部分位异或和拼接操作,以降低分支别名问题出现的概率,从而优化OpenLA500的分支预测模块。优化后的OpenLA500在动态仿真测试过程中,测试程序运行正常;在Artix-7 FPGA芯片上原型验证过程中,处理器表现良好。运行10轮测试程序,OpenLA500采用本文Gshare分支预测器,运行测试集的平均分支预测准确率提升到91.87%,较之前提高了3.22%;采用本文混合预测器,其平均分支预测准确率达到93.23%,较之前提高了4.58%。

[参 考 文 献]

- [1] MISRA S, ALFA A A, OLANIYI M O, et al. Exploratory study of techniques for exploiting instruction-level parallelism[C]//2014 Global Summit on Computer & Information Technology (GSCIT). [S. l.]: IEEE, 2014: 1-6.
- [2] SWEETY, CHAUDHARY P. An efficient branch predictor for improved accuracy of instruction level parallelism[J]. The Journal of Supercomputing, 2021, 77(10): 12098-12120.
- [3] MITTAL S. A survey of techniques for dynamic branch prediction[J]. Concurrency and Computation: Practice and Experience, 2019, 31(1): e4666.
- [4] EGAN C, STEVEN G, QUICK P, et al. Two-level branch prediction using neural networks[J]. Journal of Systems Architecture, 2003, 49(12): 557-570.
- [5] YAO C, MENG Z, GUO W, et al. Analysis and optimization of the branch prediction unit of SweRV EH1[C]//2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID). [S. l.]: IEEE, 2022: 5-9.
- [6] 龙芯中科技术股份有限公司. Snapshot[Z/OL]. [2023-10-25]. <https://gitee.com/loongson-edu/nscscc-openla500>.
- [7] LI B. Comparison of the efficiency and theoretical hardware consumption of four different dynamic and static branch predictors[J]. Journal of Physics: Conference Series, 2021, 2030(1): 012018.
- [8] 胡伟武, 汪文祥, 吴瑞阳, 等. 龙芯指令系统架构技术[J]. 计算机研究与发展, 2023, 60(1): 2-16.
- [9] 胡伟武, 高翔, 张戈. 龙芯指令系统架构及其软件生态建设[J]. 信息通信技术与政策, 2022(4): 43-48.
- [10] 龙芯中科技术股份有限公司. 龙芯架构文档[EB/OL]. (2023-10-24). <https://loongson.github.io/LoongArch-Dokumentation/README-CN.html>.
- [11] SUN M J, LI Y, CHEN S, et al. A low power branch prediction for deep learning on RISC-V processor[C]//2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP). [S. l.]: IEEE, 2021: 00037.
- [12] FAN D, YANG H, GAO G, et al. Evaluation and choice of various branch predictors for low-power embedded processor[J]. Journal of Computer Science and Technology, 2003, 18(6): 833-838.
- [13] CHANG P Y, EVERS M, PATT Y N. Improving branch prediction accuracy by reducing pattern history table interference[J]. International Journal of Parallel Programming, 1997, 25(5): 339-362.
- [14] 王强. 基于Gshare预测器的RISC-V处理器设计[D]. 合肥: 合肥工业大学, 2022.
- [15] WANG K, FRANKLIN M. Highly accurate data value prediction using hybrid predictors[C]//Proceedings of 30th Annual International Symposium on Microarchitecture. [S. l.]: IEEE, 1997: 645819.
- [16] 郭俊. 基于RISC-V处理器的分支预测研究与设计[D]. 无锡: 江南大学, 2022.

(责任编辑 胡亚敏)