第 38 卷 第 21 期
Vol.38 No.21

计 算 机 工 程
Computer Engineering

2012 年 11 月
November 2012

• 工程应用技术与实现 •　文章编号：1000—3428(2012)21—0253—04　文献标识码：A　中图分类号：TP391

一种 TLB 结构优化方法

何 军，张晓东，郭 勇

(上海高性能集成电路设计中心，上海 201204)

摘 要：针对国产处理器地址代换旁路缓冲(TLB)性能不足的问题，通过对现有的虚实地址代换流程进行分析，提出设置独立第三级页表基址虚实映射缓存，对数据 TLB 结构进行优化的方法，减少低级页表虚实映射关系对高级页表虚实映射关系的挤占淘汰。SPEC CPU2000 测试结果表明，近一半的课题能减少 60%以上数据 TLB 的 DM 次数，少数课题甚至能减少 90%以上，有效减少数据 TLB 缺失率。

关键词：地址代换旁路缓冲；缺失率；多级页表；页表；虚页号；物理页号

An Optimization Method of TLB Architecture

HE Jun, ZHANG Xiao-dong, GUO Yong

(Shanghai High Performance IC Design Centre, Shanghai 201204, China)

【Abstract】 Aiming at the problem of the inefficiency of the Translation Look-aside Buffer(TLB) of a homegrown microprocessor, based on the analysis of current virtual to real address mapping program, a method of TLB architecture optimization is put forward, which is to setup a separate virtual to real address mapping cache of the base address of third level page tables, decreasing the occurrence of replacement of higher level page table entries by lower level ones. After evaluation using SPEC CPU2000 benchmark, the Double Miss(DM) rate of the data TLB of almost half of the benchmarks is dropped down by 60% at least and some of the benchmarks are decreased by 90% above, such optimization can reduce data TLB miss rate effectively.

【Key words】 Translation Look-aside Buffer(TLB); miss rate; multilevel page table; page table; virtual page number; physical page number

DOI: 10.3969/j.issn.1000-3428.2012.21.067

1 概述

现代微处理器一般都采用分页式虚拟存储，并利用存储管理部件(MMU)实现虚地址到物理地址的映射，虚拟地址到物理地址的映射关系存储在页表中，页表管理一般由操作系统完成。为了减少访问页表的长存储访问延迟，MMU 将虚实地址映射信息保存在地址代换旁路缓冲(Translation Look-aside Buffer, TLB)中。现代多数处理器一般采用先查询 TLB 将虚地址代换为物理地址，然后访问一级 Cache，也就是说 TLB 处在处理器流水线的关键路径上。因此，TLB 对处理器的性能具有重要影响，一是其访问延迟影响处理器的周期时间，二是其命中率影响访存性能。相关的 TLB 研究表明[1-2]，一般 TLB 缺失(Miss)的开销约占系统运行时间的 5%~10%。另外还有一些研究表明[1,3-4]，对于一些访存空间比较大的应用，TLB 缺失开销占系统运行时间的比例可达到 40%~50%。对于采用软件管理 TLB 缺失处理的处理器[4]，软件缺失处理程序的执行时间占核心执行时间的比例可达 80%。

为了提高 TLB 的性能，可以从 3 个方面进行：减少访问延迟，减少缺失率，减少缺失处理开销(缺失延迟)。针对某国产处理器 TLB 性能的不足，除了采用常见两级 TLB 结构外，本文通过对现有的虚实地址代换流程进行分析，提出一种新的 TLB 结构优化方法，并进行 SPEC CPU2000 测试课题评估分析。

2 相关研究

提高 TLB 性能的主要方法包括 3 个方面：减少访问延迟，减少缺失率，减少缺失处理开销。减少 TLB 访问延迟是为了不影响处理器主频的提升，一般采用较小容量的 TLB，另一方面小容量的 TLB 又不利于减少缺失率，因此，一般采用两级 TLB 结构，即一级较小容量的 TLB 实现快速访问，二级较大容量的 TLB 作为后援，减少缺失率。一般一级 TLB 采用全联想结构，二级 TLB 采用多路组联想结构。例如 Intel Nehalem[5]设有 128 条目的一级指令 TLB，64 条目一级数据 TLB，512 条目的二级 TLB，由指令和数据共享。

在不增加 TLB 容量的前提下，为提高 TLB 命中率，大页(superpage)[2]是一种常用的技术。现代处理器和 OS 都支持多种页面粒度，例如 x86-64 支持多种页面粒度：4 KB，2 MB，4 MB[6]，Power5 处理器支持 4 KB、64 KB、16 MB 和 16 GB 等页面粒度[7]。

在缺失率难以减少的情况下，减少缺失处理开销就十

作者简介：何 军(1980－)，男，博士研究生、CCF 会员，主研方向：微处理器设计；张晓东、郭 勇，硕士

收稿日期：2011-12-29　**修回日期**：2012-02-29　**E-mail**：joyhejun@126.com

分重要。TLB 缺失处理采用硬件自动装填或者软件装填 2 种方式[8]，一般来说硬件自动装填开销较小。缺失处理开销还与页表的组织结构有关，有多级页表和哈希页表 2 种页表组织方式，其中，多级页表结构比较常见。上述页表组织结构和 TLB 装填方式各有优缺点，在现代处理器中均有被采用。

3 TLB 结构优化

3.1 虚实地址代换流程

该国产处理器支持 64 位虚地址，实际实现 43 位虚地址，高位为符号扩展。基本页面为 8 KB，页表条目为 64 位(8 Byte)。因此，对于 8 KB 页面来说，虚实地址代换为三级页表结构。为避免混淆，这里约定三级页表从高到低依次名为 L1、L2、L3，每一级页表的条目均称为 PTE，需要区别三级页表条目时，分别称为 L1PTE、L2PTE、L3PTE，PTE 由物理页号 PFN 和页保护信息等组成。L1 页表的基址保存在页表基址寄存器 PTBR 中，L3PFN 指向数据所在的物理页基址，再加上页面偏移 VA[12:0]可以访问具体的数据，如图 1 所示。

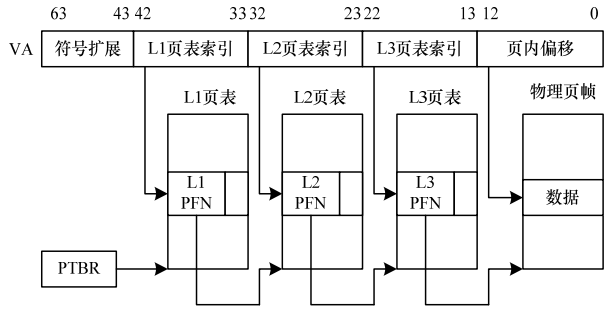

图 1 国产处理器三级页表代换示意图

该国产处理器采用哈佛结构，分别设置指令 TLB 和数据 TLB，本文的研究主要针对数据 TLB 进行。对于数据流虚地址 VA，以 8 KB 页面为例，总体访问流程如下：

(1)普通访存指令访问虚地址 VA 时，先查询 TLB，若命中，则取出物理页号 PFN 与页内偏移组合成 VA 的物理地址 PA，虚实地址代换结束。

(2)若不命中 TLB，则发生 TLB Single Miss(SM)自陷，由对应的自陷处理程序(SMR)进行处理，SMR 将使用特权指令 pri_ldvpte 按 L3PFN 的虚地址 VA3 读取 L3PTE 以获取 L3PFN。

(3)按 VA3 读取时，也要先查询 TLB，如果命中，则取出物理页号 PFN，与页内偏移{VA[22:13],000}组合得到 VA3 的物理地址 PA3，并按 PA3 读取 L3PTE 以获取 L3PFN(如果发生缺页，则转入缺页故障处理程序)，然后取出 L3PFN，最后将 VA 的虚页号 VPN(VA[42:13])与 PFN(L3PFN)的映射关系装填写入 TLB 中。TLB SM 自陷处理完成，并返回，转步骤(7)。

(4)按 VA3 读取时，如果不命中 TLB，则发生 TLB Double Miss(DM)自陷，处由对应的自陷处理程序(DMR)进行处理，DMR 将使用特权指令 pri_ldp 按照物理地址进行访问。根据页表基址寄存器 PTBR 依次访问 L1 和 L2 页表，取得虚地址 VA3 的 PFN(L2PFN)。在访问前两级页表的过程中也可能发生缺页，如果发生，则转入缺页故障处理程序。

(5)将 VA3 的 VPN 和 PFN(L2PFN)的映射关系装填写入 TLB 中，TLB DM 自陷处理完成，返回 TLB SM 自陷处理流程。

(6)重新执行特权指令 pri_ldvpte 按 VA3 读取，应该命中 TLB，处理与步骤(3)类似。

(7)从 TLB SM 自陷返回后，重新执行最初的普通访存指令按虚地址 VA 进行访问，应该命中 TLB，处理与步骤(1)类似。

3.2 虚实地址代换分析

通过对该虚实地址代换流程分析，目前的 TLB 中实际上缓存了 2 类地址代换信息：

(1)VA-L3PFN：普通数据流虚地址 VA 的虚实映射关系，即 VA 的虚页号 VPN 与其物理页号 PFN(L3PFN)的映射关系。

(2)VA3-L2PFN：L3PTE 的虚地址 VA3 的虚实映射关系，也就是 VA3 虚页号 VPN 与其物理页号 PFN(L2PFN)的映射关系。这实际上是 L3 页面基址的虚实映射关系。

可见，现有 TLB 中缓存了基本物理页和 L3 页面基址的虚实映射关系，但没有缓存 L1、L2 页面基址的虚实映射关系。

根据多级页表结构，各级页表页面数量如下：(对于 43 位虚地址和 8 KB 页面)基本物理页面(8 KB)：2^{30}；L3 级页表页面：2^{20}；L2 级页表页面：2^{10}；L1 级页表页面：1。如果将各级页表页面基址的虚实映射关系都混放在一个 TLB 中，则容易出现低级页面基址的映射关系装填引起高级页面基址的映射关系被淘汰的情况，而高级页面基址映射关系发生缺失，影响的 TLB 映射范围更大，处理开销更大(需要多次访存)。因此，如果各级页表页面基址的映射关系独立缓存在各自的 TLB 中，则能减少这种情况的发生，只有同级别的映射才能互相淘汰。

3.3 结构优化

为了避免不同级别虚实映射关系的互相淘汰情况，本文考虑将现有 TLB 中混放的虚实映射关系分别独立缓存。保持现有数据 TLB(DTLB)的两级结构不变，但仅缓存基本物理页面基地址的虚实映射关系，不再缓存 L3 页面基址的虚实映射关系，后者由独立的设置 L3PTB 缓存(32 条目)。具体配置如表 1 所示。

表 1 TLB 配置参数

TLB 名称	缓存内容	条目数
DTLB	缓存基本物理页面基址的虚实映射关系(VA-L3PFN)和页保护信息等	一级：64 二级：512
L3PTB	缓存 L3 页面基址的虚实映射关系(VA3-L2PFN)和页保护信息等	32

由于 2 种虚实映射关系的独立缓存，TLB 的访问方式有所不同。原来普通访存指令和特权访存指令都是查询和

装填 DTLB，现在只有普通的访存指令查询 DTLB，发生 SM 后，执行 pri_ldvpte 指令，然后将“VA-L3PFN”虚实映射关系装填 DTLB；只有特权访存指令 pri_ldvpte 才查 L3PTB，并在发生 DM 后，执行 pri_ldp 指令，然后将“VA3-L2PFN”虚实映射关系装填 L3PTB。替换策略都是按照先进先出的原则。

新的 TLB 访问方式如表 2 所示。

表 2 新 TLB 访问方式

TLB 名称	查询	装填
DTLB	普通访存指令，非 pri_ldvpte 指令	发生 SM 后，执行 pri_ldvpte 指令结果而进行的 TLB 装填，替换策略按照先进先出
L3PTB	pri_ldvpte 指令	发生 DM 后，执行 pri_ldp 指令结果而进行的 TLB 装填，替换策略按照先进先出

3.4 性能评估

3.4.1 评估方法

基于本文内部的功能模拟器 ISP，按照 3.3 节的方案对其 DTLB 结构进行优化，通过运行 SPEC CPU2000[1] 测试课题，比较 DTLB Miss 的次数情况，分析性能改进。ISP 采用采用 C 语言编写，能完成指令集的功能模拟，对 DTLB 结构进行了模拟，能够运行操作系统核心和用户课题，但不含时钟周期等时序信息。

通过运行 SPEC CPU2000 整数和浮点测试课题，分别统计改进前后 2 种 DTLB 结构下发生 DTLB SM 和 DM 次数，对比分析性能改进情况。

3.4.2 评估结果

实验评估结果如表 3、表 4 所示，分别统计了 SPEC CPU2000 整数和浮点课题中 2 种 DTLB 结构配置下，DTLB 发生 DM 的次数(“DTLB DM”列和“新 DTLB DM”列)、发生 SM 的次数(“DTLB SM”列和“新 DTLB SM”列)，以及改进前后 2 种结构的 DM 差值(“DM 差”列)、DM 改进百分比(“DM 改进比”列)、SM 差值(“SM 差”列)、SM 改进百分比(“SM 改进比”列)。

表 3 SPEC CPU2000 整数课题评估结果

课题名称	DTLBDM	新 DTLBDM	DM 差	DM 改进比/(%)	DTLBSM	新 DTLBSM	SM 差	SM 改进比/(%)
164.gzip	74	58	16	21.62	4 595	4 553	42	0.91
175.vpr	246	229	17	6.91	3 115	3 027	88	2.83
176.gcc	1 154	1 005	149	12.91	23 661	23 343	318	1.34
181.mcf	133	126	7	5.26	48 347	48 362	−15	−0.03
186.crafty	386	156	230	59.59	6 950	7 051	−101	−1.45
197.parser	716	393	323	45.11	44 413	43 484	929	2.09
252.eon	172	172	0	0.00	762	761	1	0.13
254.gap	734	515	219	29.84	90 723	89 804	919	1.01
255.vortex	10 915	1 203	9 712	88.98	1 243 386	1 210 973	32 413	2.61
256.bzip2	3 889	72	3 817	98.15	439 218	422 723	16 495	3.76
300.twolf	279	279	0	0.00	1 094	1 094	0	0.00

表 4 SPEC CPU2000 浮点课题评估结果

课题名称	DTLBDM	新 DTLBDM	DM 差	DM 改进比/(%)	DTLBSM	新 DTLBSM	SM 差	SM 改进比/(%)
168.wupwise	683	214	469	68.67	400 448	400 243	205	0.05
171.swim	810	238	572	70.62	275 727	275 724	3	0.00
172.mgrid	12 661	154	12 507	98.78	3 070 884	3 051 944	18 940	0.62
173.applu	951	219	732	76.97	45 897	44 552	1 345	2.93
177.mesa	319	246	73	22.88	17 845	17 735	110	0.62
178.galgel	3 068	980	2 088	68.06	103 724	102 333	1 391	1.34
179.art	541	48	493	91.13	5 376	3 760	1 616	30.06
183.equake	228	78	150	65.79	51 197	51 133	64	0.13
187.facerec	966	239	727	75.26	350 067	349 829	238	0.07
188.ammp	64 119	435	63 684	99.32	18 437 240	18 360 606	76 634	0.42
189.lucas	147	147	0	0.00	444	443	1	0.23
191.fma3d	375	375	0	0.00	1 680	1 672	8	0.48
200.sixtrack	1 925	1 263	662	34.39	47 136	46 339	797	1.69
301.apsi	431	425	6	1.39	53 105	53 124	−19	−0.04

评估结果表明，设置独立 L3PTB 的新 DTLB 结构能有效减少 DM 次数，对不少课题能减少 60%以上，甚至 90%以上，比如课题 256.bzip2、172.mgrid、179.art、188.ammp，对浮点测试课题效果更为明显。但对减少 SM 次数效果有限，普遍低于 4%，只有 179.art 课题能减少 30%，另外少部分课题 SM 次数还略有增加，原因主要在于实验评估时，L3PTB 淘汰的条目直接丢弃，而 L3PTB 条目只有 32 条目。可以对该结构作进一步优化，比如将 L3PTB 淘汰的条目写入二级 DTLB，以弥补 L3PTB 条目数量的不足。

4 结束语

TLB 是微处理器的重要部件，对处理器性能具有重要影响。针对国产处理器 TLB 性能的不足，除了采用两级 TLB 结构外，本文对该处理器的虚实地址代换流程进行了分析和研究，提出通过设置独立的第三级页表基址虚实映射缓存 L3PTB，以减少低级页表虚实映射关系对高级页表虚实映射关系的挤占淘汰，减少 DTLB DM 的次数，从而减少 DTLB 缺失率。实验评估结果表明，该结构能有效减少 DM 次数。这种结构优化方法对多级页表结构的 TLB 具有普遍意义，通过将不同级别的页表虚实映射关系独立缓冲，减少低级页表对高级页表的挤占淘汰，从而减少 TLB 缺失率。

参考文献

[1] Kandiraju G, Sivasubramaniam A. Characterizing the d-TLB Behavior of SPEC CPU2000 Benchmarks[C]//Proc. of 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. New York, USA: [s. n.], 2002.

[2] Talluri M, Hill M. Surpassing the TLB Performance of Superpages with Less Operating System Support[C]//Proc. of International Conference on Architectural Support for Programming Languages and Operating Systems. New York, USA: [s. n.], 1994.

[3] McCurdy C, Cox A L, Vetter J. Investigating the TLB Behavior of High-end Scientific Applications on Commodity Microprocessors[C]//Proc. of IEEE International Symposium on Performance Analysis of Systems and Software. Washington D. C., USA: [s. n.], 2008: 95-104.

[4] Rosenblum M, Bugnion E, Herrod S A, et al. The Impact of Architectural Trends on Operating System Performance[C]//Proc. of the 15th ACM Symposium on Operating Systems Principles. New York, USA: [s. n.], 1995.

[5] Ronak S. Inside Intel Next Generation Nehalem Microarchitecture[Z]. Intel Developers Forum, 2008.

[6] Hennessy J L, Patterson D A. 计算机系统结构：量化研究方法[M]. 白跃彬, 译. 北京: 电子工业出版社, 2007.

[7] Korn W, Chang M. SPEC CPU2006 Sensitivity to Memory Page Sizes[J]. ACM SIGARCH Computer Architecture News, 2007, 35(1): 97-101.

[8] Jacob B, Mudge T. Virtual Memory in Contemporary Microprocessors[J]. IEEE Micro, 1998, 18(4): 60-75.

编辑 索书志

(上接第 252 页)

参考文献

[1] Zhang Yuanfang, Gill C D, Lu Chenyang. Configurable Middleware for Distributed Real-time System with Aperiodic and Periodic Tasks[J]. IEEE Transactions on Parallel and Distributed System, 2010, 21(3): 393-404.

[2] Graham S, Baliga G, Kumar P R. Abstractions, Architecture, Mechanisms, and a Middleware for Networked Control[J]. IEEE Transactions on Automatic Control, 2009, 54(7): 1490-1503.

[3] AUTOSAR Alliance. Autosar Specification V4.0[Z]. 2010.

[4] Wu Ruyi, Li Hong, Yao Min, et al. A Hierarchical Modeling Method for AUTOSAR Software Components[C]//Proc. of the 2nd International Conference on Computer Engineering and Technology. Chengdu, China: [s. n.], 2010: 184-188.

[5] Voget S. AUTOSAR and the Automotive Tool Chain[C]//Proc. of DATE'10. Dresden, Germany: [s. n.], 2010: 259-262.

[6] Hofer W, Lohmann D, Schröder-Preikschat W. Concern Impact Analysis in Configurable System Software——The AUTOSAR OS Case[C]//Proc. of 2008 AOSD Workshop on Aspects, Components and Patterns for Infrastructure Software. Brussels, Belgium: [s. n.], 2008.

[7] Lee Joo Chul, Han Tae Man. ECU Configuration Framework based on AUTOSAR ECU Configuration Metamodel[C]//Proc. of International Conference on Convergence and Hybrid Information Technology. Daejeon, Korea: [s. n.], 2009: 260-263.

[8] Klobedanz K, Kuznik C, Thuy A. Timing Modeling and Analysis for AUTOSAR-based Software Development——A Case Study[C]//Proc. of DATE'10. Dresden, Germany: [s. n.], 2010: 642-645.

[9] Piao Shiquan, Jo H, Jin Sungho. Design and Implementation of RTE Generator for Automotive Embedded Software[C]//Proc. of the 7th ACIS International Conference on Software Engineering Research, Management and Applications. Haikou, China: [s. n.], 2009: 159-165.

[10] Juan P, Pimentel R. An Incremental Approach to Task and Message Scheduling for AUTOSAR Based Distributed Automotive Applications[C]//Proc. of the 4th International Workshop on Software Engineering for Automotive Systems. Minneapolis, USA: [s. n.], 2007: 1-7.

编辑 索书志