文章编号:1001 - 2486(2004)04 - 0010 - 05

高性能微处理器 TLB 的优化设计*

陈海燕,邓让钰,邢座程
(国防科技大学计算机学院,湖南 长沙 410073)

摘 要:虚拟存储是现代微处理器系统必不可少的存储模式。在虚存模式下,虚拟地址到物理地址的变换是流水线中最频繁的核心服务,容易处于决定处理器时钟周期的关键路径上。为加快虚存的访问,现代高性能微处理器实现了一种硬件地址映射结构 转换后援缓冲器(简称 TLB)。在分析 TLB 传统的地址映射机制的基础上,提出了基于虚区域和 Cache 块标记的预验证技术,结果表明该技术优化了 TLB 的设计,避免了 TLB 访问时延成为访存的瓶颈。

关键词:虚拟存储;TLB;地址变换;预验证;Cache 块标记
中图分类号:TP333 文献标识码:A

The Optimization Design of TLB of High Performance Processor

CHEN Hai-yan, DENG Rang-yu, XING Zuo-cheng
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: The virtual memory is a staple in modern processor system. In virtual addressing scheme, the translation from virtual address to physical address is one of the highest frequency core service in the pipe line, and tends to be on the critical path determining the clock cycle of the processor. In order to speed up the address translation, the most modern processors have designed a hardware unit called translation look-aside buffer(TLB). Based on analyzing the traditional address mapping mechanism of TLB, this paper has put forward the regions and Cache line tag pre-validation to optimize the translation, and removed the TLB delay bottleneck in the whole memory access.

Key words: virtual memory; TLB; addressing translation; pre-validated; Cache address line tag

虚拟存储是现代高性能微处理器必须支持的存储模式,它自动管理由主存和辅存构成的二级存储层次,为支持多用户、多进程的现代计算机系统提供硬件支撑;简化了程序的装载:只要改变物理存储器与虚拟存储器间的映射关系,同一程序可运行在物理存储器的任何位置;它以透明的方式,给用户提供了一个比实际物理主存空间大得多的虚拟地址空间[1]。

计算机应用技术要求越来越大的地址空间,目前最新的 64 位微处理器的寻址空间可达 2^{64} 字节。在虚存模式中,64 位地址空间被分为多个区域,每个区域内包含一个 18 位以上的标志符,不仅能为多个进程提供单独的地址空间,而且将有限的物理地址空间映射到大得多的虚拟地址空间,从而实现资源虚拟化、信息共享与保护以及各进程的故障隔离。

在虚存模式下,每次取指、数据访问都要触发虚实地址转换。为加快虚实地址变换,根据程序访问的局部性原理,现代微处理器内设计了转换后援缓冲器(TLB),存放近期用到的虚实地址映射,处理器通过查询 TLB 中是否存在匹配的页表项,获得本次访存的物理地址。由于高性能处理器不断开发更高指令级并行性(ILP)的要求,TLB 出现了多端口结构;TLB 的页表项的位宽也随着虚拟和物理寻址空间的增大而迅速增大,地址变换的比较时间和内容读出时间变长,使 TLB 的访问处于访存的关键路径。同时伴随着处理器主频的不断提高,要在一个时钟周期内完成访存,即在 TLB 和 Cache 同时命中的情况下,一拍内既要完成 TLB 地址变换,又要完成 Cache 的访问,变得越来越困难。因此,高性能处理器必须优化 TLB 的地址映射过程,才能使地址变换的延迟满足系统高速访存的要求。

* 收稿日期:2004 - 03 - 10
基金项目:国家自然科学基金资助项目(90207011)
作者简介:陈海燕(1967—),女,副研究员,硕士。

针对这一问题,本文在分析传统的地址变换模式的基础上,提出了一种 TLB 的预验证方法,能有效缩短 TLB 的访问时延。

1 传统的地址变换机制

1.1 虚拟地址

虚存模式下,编译器生成的程序目标代码的逻辑地址空间被称为虚拟地址空间,主存的存储空间被称为物理地址空间;目标代码只有装入主存,被处理器读取,程序才能执行。为了更有效利用主存资源,动态地完成逻辑地址到物理地址转换,虚存系统采用了页式管理策略:即把物理地址空间等分为大小相同、位置固定的块,称之为页帧(物理地址),虚拟地址空间也等分为与其大小相同的块,称为页面(虚拟地址),每个虚拟地址映射一个物理地址,物理地址可以不连续。虚拟地址由虚页号和页内偏移表示,物理地址由物理页号和页内偏移表示,因两者页内偏移相同,因此访存时只需将虚拟页号转换成物理页号。为此,操作系统为处理器提供一个反映虚实页号映象关系的页表机构。对于 64 位处理器平台,页表通常太大,需要放到主存或虚存中。TLB 是页表的子集,存放处理器最近使用的页面地址映象,由处理器硬件实现。根据程序局部性原理,TLB 命中是大概率事件,与查询页表相比,TLB 无疑加速了地址变换,为一拍完成虚地址访存提供了可能。

1.2 传统的 TLB 地址变换机制

为支持 MAS(多地址空间)操作系统,高性能微处理器将虚地址寻址的概念扩大到支持虚区域:将按字节寻址的虚地址空间分为若干虚区域,由虚地址的最高几位作为虚区域地址槽号译码选择。每个虚区域对应一个虚区域寄存器,每个区域寄存器内定义一个 18 位以上的区域标志符 RID[3,5],惟一标识某给定进程的虚地址空间。

虚地址由虚区域号、虚页号和页内偏移组成,由 TLB 将其转换成物理地址。TLB 分标记和数据体两部分,由 CAM(内容相联存储器)实现。标记存有虚页号、区域标志符、有效位等内容;数据体包含对应的物理页号、权限、保护域、存储属性等内容。传统的访存步骤如图 1 所示,即首先将去掉页偏移的虚地址与 TLB 的标记体进行比较,存在匹配,则读出对应数据体中的物理页号,获得本次访问的物理地址,和由地址偏移索引得到的 Cache 块地址标记进行比较,最后获得访问数据。

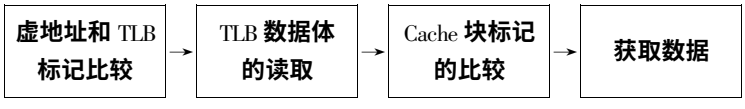

图 1 传统的虚存访问流程

Fig.1 Traditional virtual memory access flow

TLB 地址转换的具体过程如图 2 所示,即先用访存的虚地址高位索引区域寄存器,得到区域标志符 RID;RID 再和虚拟页号一起形成访问 TLB 的比较标记,与 TLB 的有效标记进行匹配,然后读出命中项对应的数据体;若读出的该项的保护和访问权限属性均表示此页可以访问,则取出相应的物理页号,并与页内偏移一起生成物理地址。TLB 缺失时,则需访问主存中的页表,更新 TLB 表项。

从图 2 可见,每次地址变换都需要完成对区域寄存器额外的查询步骤,再形成访问 TLB 的比较标记,该步骤是 TLB 访问过程中的关键路径,给地址变换带来了巨大的压力。

为满足更高的指令并行性(ILP)的要求,一些高性能处理器采用了多端口的 TLB 设计。同样条件下,多端口的 TLB 具有更大的时延压力[2]。若采用传统的地址映射流程,标记比较和数据体的读出是串行过程;对于多端口 TLB,要求在一拍内获取物理地址和访存数据,是非常困难的。下面我们在 TLB 的设计中引入预验证技术,能有效缩短地址变换的时延,满足高速访存的要求。

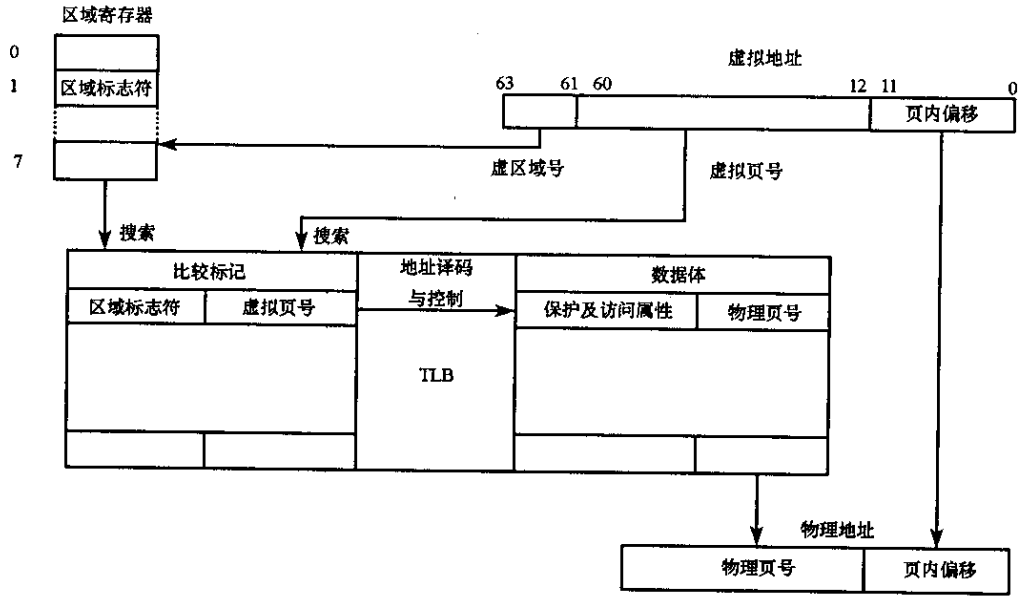

图2 传统的TLB地址变换过程

Fig.2 Traditional TLB address translation process

2 采用预验证技术的TLB优化设计

TLB是处理器中一种特殊的Cache,其平均访问时间也可依据Cache的访问时间公式,即:TLB平均访问时间=命中时间(命中率×命中开销)+缺失率×缺失开销[1],根据这个公式,TLB的优化策略可归结为以下3类:降低TLB的缺失率、减少缺失开销和减少TLB的命中时间。现在许多TLB的研究都集中在如何降低缺失率、减少缺失开销方面,如采用多级层次结构、实现硬件页搜索器,或采用TLB表项的提前预取技术,隐藏部分缺失开销等[2]。下面提出的预验证技术则是从如何减少TLB命中时间方面考虑的。

2.1 虚区域预验证技术

虚区域预验证技术,就是通过在TLB项中保存虚区域号VRN和区域标志符RID,设置预验证位,使区域被提前验证,因此访存时,能将区域寄存器从TLB匹配的关键路径上消除,加快地址匹配的速度。

根据虚区域预验证技术要求,TLB域中除了传统设计中的有效位、虚页号位、区域标志符RID、物理地址、保护和访问属性等域外,增加了区域预验证有效位PRV和虚区域号(VRN),PRV位表示当前VRN与对应的RID是否存在有效映射关系。

具体实现过程如图3所示。当为区域寄存器中的RID对应区域中某页建立虚实映射的TLB页表项时,将其RID和VRN都保存在TLB页表项的对应域中,并置有效位和预验证位PRV有效,表示该TLB项包含一个有效VRN——RID映射,提前验证了该区域。当处理器进行虚实转换时,从虚地址中选出VRN和虚页号形成访问TLB的比较标记,与有效位、预验证位均有效的所有TLB的标记进行匹配,若存在匹配项,则TLB命中。因为RID已经在TLB项中提前验证,不需再用虚区域号VRN去索引区域寄存器以获取RID。这样不仅将区域寄存器从地址转换的关键路径中消除,而且形成的比较标记位宽也比原来少。若VRN为3位,索引8个区域寄存器,对于24位的RID,采用区域预验证技术后,比较标记位宽比以前少了20位,有效减少了地址匹配逻辑级数和实现代价。

使用区域预验证技术,在改写区域寄存器进行进程切换时,产生了一个小小的性能代价。没有使用预验证技术时,写区域寄存器操作只需将RID等内容写入指定的区域寄存器,因为每次地址转换路径中均要通过访问区域寄存器获得RID,已被替换的RID不会出现。若TLB项中RID与获取的RID不一致,该TLB项就不会命中。如果以后要将被替换的RID在该区域寄存器中恢复,则再将其重新写入该RR,则原来的TLB项将又被重新访问。使用TLB预验证技术后,地址转换时不再访问区域寄存器。当需要

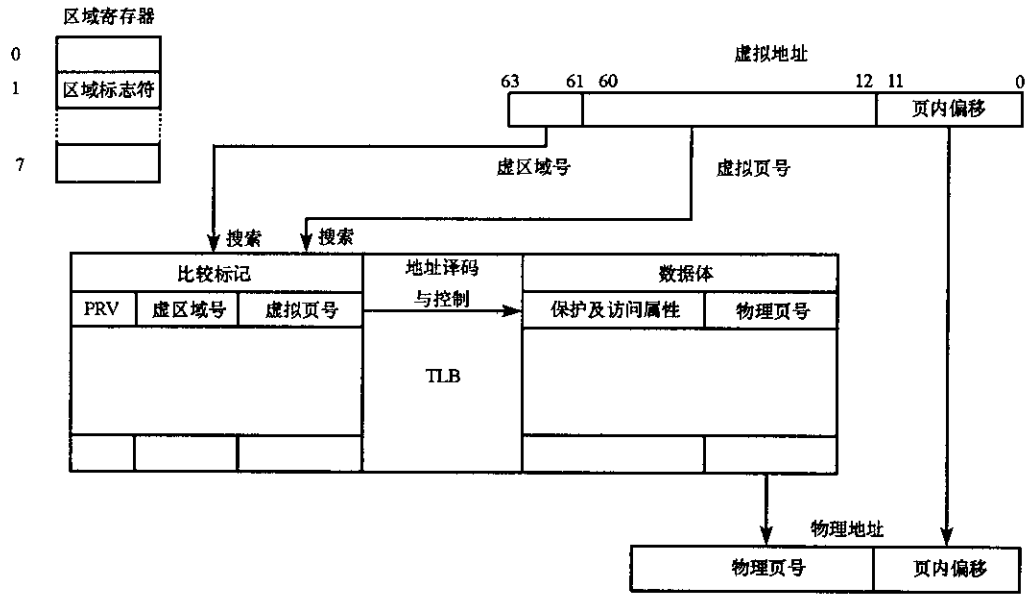

图 3 区域预验证的 TLB 地址变换过程

Fig.3 TLB address translation process based on region pre-validation

向某个区域寄存器写入新的 RID 时,必须搜索所有有效 TLB 项中的 VRN 域。如果与欲接受新 RID 的区域寄存器地址槽号一致,则需检查该项的 RID 是否与新写入的 RID 一致。若一致,则将该项的区域预验证位 PRV 置有效,否则将其置无效。当改变活动的区域时,该操作比以前多了检查预验证位是否有效的操作。由于虚实地址变换发生的频率远远高于改写区域寄存器的操作,根据局部性原理,预验证技术加快了大概率事件的执行速度,系统的总性能得到了提高。

2.2 Cache 地址标记的预验证技术

2.2.1 原理

为减少访存的缺失率,减少缺失开销,高性能处理器开始采用多级 TLB 和多级片内 Cache 结构。为满足高速访存的要求,一级 TLB 表项数和一级 Cache 容量较小。一级 Cache 一般被组织为两路或四路组相联映射方式,每一个 Cache 块带一个地址标记 tag。根据 Cache 的容量、组相联形式和块大小,其物理地址由高到低被分为以下三部分:

tag	Cache 块号	块内偏移

由于虚拟地址与物理地址的页内偏移相同,如果 Cache 每一路的容量不大于最小页面大小,利用虚地址的页内偏移地址可索引 Cache 块号,访问一级 Cache 可以和访问 TLB 同时进行。传统访存时,将 TLB 命中后从数据体中读出的物理页号(即物理地址高位)作为访问 Cache 块的比较标记,与按页内偏移索引得到的各路 Cache 块的地址标记进行比较,匹配后则获得访存的数据。

采用 Cache 块地址标记的预验证技术后,Cache 块的地址标记(tag)不再保存物理页号,而保存该 Cache 块对应的页表项在一级 TLB 的匹配信息编码。TLB 访问时,先通过地址比较得到所有 TLB 表项的匹配位,存在命中时,该匹配编码直接作为 Cache 块的比较标记,去访问 Cache 获取访存数据。若根据处理器可实现的物理地址空间的大小,适度控制一级 TLB 的项数,可使该标记宽度小于传统访存中 Cache 块所存的物理页号。这样不但完全消除了 TLB 地址匹配后再从数据体读取物理地址的过程,还减少了 TLB 命中后标记的比较位数,加快了访存速度。

在访存中获得这一好处是以增加处理器存储系统的管理复杂性为代价的。采用标记的预验证技术后,一级 Cache 与一级 TLB 之间需紧密耦合。对 TLB 的填充、替换及清洗操作时需要对一级 Cache 中的对应项进行相应的操作,以维护 TLB 表项的匹配编码与其对应 Cache 块标记的一致性。某项 TLB 被替换或无效,需要将 L1 Cache 中的所有对应块无效,因此 TLB 缺失必然造成 L1 Cache 缺失。根据程序局

部性原理,一级 TLB 和 Cache 缺失率很低,标记的预验证技术符合大概率事件优先原则。

2.2.2 设计与实现

假设一个 64 位的处理器平台,实现 50 位物理地址访存,最小页面为 4KB,具有两级片内 Cache 和两级 TLB 结构。L1 TLB 为 CAM 实现的 32 项全相联结构,L1Cache 容量为 16KB,4 路组相联,每一 Cache 块为 32 字节。因此,每一路 Cache 为 4KB,共 128 块。因为虚实地址页内偏移一样,物理地址的低 12 位就是虚地址的低 12 位,这 12 位的高 7 位用来索引 4 路组的 Cache 块,传统方式中 Cache 块的地址标记存的就是物理页号,即高 38 位的物理地址,若 TLB 命中,则用 TLB 获取的物理页号与由地址偏移索引得到的 4 个 Cache 块标记进行比较,存在匹配,则 Cache 命中,获得访存的数据。

采用 Cache 块标记的预验证设计,L1 TLB 可只保留虚区域号、虚页号、有效位等构成的标记体部分,是 L2 TLB 的严格子集;L2 TLB 由标记体和数据体两部分组成,具有物理页号在内的页表项的完整信息。访存时 L1 TLB 和 L1 Cache 可同时访问。存储管理系统需保证同级 TLB 的惟一性,即同级 TLB 内无地址重叠或相同的表项。因此 L1 TLB 中,访存的虚地址与 32 项表项同时比较,最多只有一个命中,共存在 33 个 32 位的匹配结果:0、2^0、2^1、2^2、2^3、…、2^{31}。

系统初始化时,访存产生缺失中断后(匹配编码为 32'h00000000),某一具有完整信息的页表项命中,假设该表项被填入到 L1 TLB 的第一项的位置,则命中该 TLB 项的匹配编码为(32'h00000001)。L1 Cache 块填充时,TLB 的位置信息码作为该 Cache 块的地址标记。由于 TLB 表项的惟一性,L1 Cache 中的每一 Cache 块的标记都是 L1 TLB 中某一 32 位的匹配编码,L1 TLB 的某一匹配编码与 L1 Cache 块标记存在一对多的映射关系,因此某一 TLB 项被替换或无效时,必须使 L1 Cache 中所有对应的 Cache 块无效。

从这个例子看到,Cache 块地址标记的预验证技术可使 TLB 省掉存储物理页号的数据体资源,L1 TLB 至少可减少 38(物理地址) × 32 = 1216 位,并消除了读取数据体的关键路径;而 32 位的匹配编码又比 38 位的物理页号少了 6 位,使整个 L1 Cache 的块标记资源节省 128(行) × 4(路) × 6(位) = 3072 位。TLB 的比较结果直接作为选通 L1 Cache 的标记,当两者都命中的情况下,可实现一拍的访存速度要求。

3 结 论

根据上面的分析,设计了 32 项全相联结构 TLB,并在 Sun 工作站进行模拟、综合,结果表明,采用 TLB 预验证技术后,TLB 访问时延仅为传统 TLB 的一半。因此,采用虚区域和 Cache 块标记预验证的优化设计充分利用了程序局部性原理,优化了 TLB 访问的关键路径,避免了 TLB 访问时延成为高性能处理器访存的瓶颈。

参 考 文 献:

[1] Hennessy J L, Patterson D A. 计算机体系结构——一种定量的方法[M]. 郑纬民,等译. 北京:清华大学出版社,2001:281 – 355.

[2] Kandiraju G B, Anand Sivasubramaniam, Characterizing the d-TLB Behavior of SPEC CPU2000 Benchmarks[A]. Proceedings of 29th International Symposium on Computer Architecture, June,2002.

[3] Stallings W. 计算机组织与结构——性能设计(第五版)[M]. 张昆藏,等译. 北京:电子工业出版社,2001:189 – 195.

[4] McNairy C, Soltis D. Itanium 2 Processor Microarchitecture[J]. IEEE Micro, March-April,2003:44 – 55.

[5] Intel(R) Itanium(R) 2 Processor Reference Manual for Software Development and Optimization[R]. Intel Corp. 2002:29 – 47.

[6] Itanium(R) Architecture Software Develop'Manual V2:System Architecture[R]. Intel Corp.2001:37 – 59,425 – 440.