一种选择性冲突预测高缓方案

李晓明[1],鲍东星[2],刘晓为[1]
(1. 哈尔滨工业大学微电子科学与技术系,黑龙江哈尔滨 150001;
2. 黑龙江大学电子工程学院,黑龙江哈尔滨 150080)

摘 要: NTS 高缓为直接映象高缓扩充了一个小的全相联高缓,其中保存被预测为具有非时间局部性的数据块. 与牺牲高缓的区别在于 NTS 高缓在两个高缓之间没有直接的数据通路,因此结构设计简单,功耗低. 本文提出了一个 NTS 高缓的改进方案,称为选择性冲突预测高缓(SCP 高缓)设计方案. SCP 高缓利用冲突预测算法监测高缓中数据块局部性,并有选择性地将数据块填充到直接映象高缓或全相联高缓中. 仿真结果显示,容量相当的 SCP 高缓性能优于 NTS 高缓.

关键词: 高缓性能;冲突预测;缺失率
中图分类号: TP302.2 **文献标识码:** A **文章编号:** 0372-2112 (2018)02-0473-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.02.029

A Cache Scheme by Selective Conflict Prediction

LI Xiao-ming[1], BAO Dong-xing[2], LIU Xiao-wei[1]
(1. *Dept. of Microelectronic Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang* 150001, *China*;
2. *School of Electronic Engineering, Heilongjiang University, Harbin, Heilongjiang* 150080, *China*)

Abstract: NTS cache augments the direct-mapped main cache with a small fully-associative cache that holds those blocks predicted as holding non-temporal locality characteristics. The most outstanding difference of the NTS cache from the victim cache lies in the NTS cache doesn't have direct data path between the two caches, so its advantages are lower power and easier structure design. In this paper, an improvement of the NTS cache scheme, called Selective Conflict Prediction cache (SCP cache), is proposed. In this scheme, incoming blocks into the cache are placed selectively in the direct-mapped cache or the fully-associative cache by the use of a conflict prediction algorithm which detects the locality of data blocks in the cache. Simulation results show that the performance of SCP cache is always better than that of NTS cache with similar area.

Key words: cache performance; conflict prediction; miss ratio

1 引言

访存延迟一直是微处理器的性能瓶颈. 尤其是在进行高性能超标量以及超长指令字处理器设计时,访存问题更加突出. 处理器与存储器性能的差距不断扩大,这样一来使得高缓缺失(cache miss)导致的性能损失代价也就越发高昂. 因此,现代微处理器体系结构中,高缓在降低访存代价和功耗方面越来越重要. 同时,随着半导体工艺水平的不断提高,使得单个芯片上可集成更多的功能部件而同时不会增加开销,因此采用片内高缓智能管理机制成为可能,该机制按照高缓动态访问模式确定缓存策略,从而改善片内高缓的性能.

高缓的性能是由命中率和访问时间共同决定的. 通常容量大、相联度大的高缓具有较高的命中率,但是从处理器设计角度来看,大容量的多路组相联高缓并不合适,因为大量使用价格昂贵的高速 SRAM 会大大增加系统开销,而且随着高缓相联度的增加,相应的控制逻辑复杂度也增大,从而增加了高缓访问时间. 因此目前一些处理器厂商宁愿采用直接映象高缓(Direct Mapped Cache,简称 DM 高缓)做为其高缓组织形式,这是因为在容量相同的条件下,DM 高缓比其他映象形式的高缓结构的访问延迟小,而且结构实现简单,数据访

收稿日期:2017-04-10;修回日期:2017-07-05;责任编辑:郭游

问与标签比较可以同时进行,且每次高缓访问只需进行一次标签比较,大量节省了复杂的全同比较线路. 但其缺点也很明显:命中率较其它组织形式高缓低,这主要是由大量冲突缺失而引起的. 文献[1]通过对应用程序代码的特征分析指出很少的 load/ store 指令引发大量的高缓缺失,因此高缓的选择性缓存是高缓管理重要方向.

为了减少 DM 高缓命中率低的问题,研究人员提出了很多高缓管理策略和方法,如非时间局部性分流高缓(简称 NTS 高缓)[2,3],牺牲高缓[4],高缓预取[5],踪迹高缓[6],以及其他新的高缓方案[7~9],其中牺牲高缓无疑是最著名的. 牺牲高缓在 DM 高缓的基础上扩充了一个小的全相联高缓,两个高缓可并行访问,同时由于全相联高缓的容量很小,所以总的访存时间与 DM 高缓相当,这种高缓结构称为扩充高缓. 但是,由于两个高缓之间大量的数据传输会使得控制机制复杂,增加了大量功耗,在某些系统中甚至会导致更多的延迟命中情况. 一些研究者据此把研究重点投向两个高缓之间彼此不发生数据交换的高缓策略上,即无通路扩充高缓,具有代表性的是 NTS 高缓和程序指针选择性高缓[2,3,10](简称 PCS 高缓).

NTS 高缓设计方案是基于如下前提而提出的:程序中的数据访问行为是不同的,因此将数据流分为不同的种类,由此采用选择性缓存可能会改进高缓性能. 尤其是 NTS 高缓将数据流分成具有时间局部性块(T 型块)和非时间局部性块(NT 型块),T 型块在其高缓生命期中表现出较高的复用的可能性,以此被赋予更高的优先权,以尽可能长时间停留在高缓中. PCS 高缓的设计原理与 NTS 高缓类似,不过其数据分配策略是基于引发数据缺失的访存指令的 PC 指针来进行的,而 NTS 高缓则是基于 NTS 高缓中块的有效地址. 从文献[1]的分析可知,少数的数据装载指令对数据块的访问引发了绝大多数的数据缺失,因此将数据访问指令的程序指针做为数据块分配的判断标准可能会改善数据高缓的命中率. 文献[11]提出了一个基于冲突预测表的无通路扩充高缓方案-冲突预测高缓(简称 CP 高缓),在 NTS 高缓的基础上进一步提升了高缓的性能.

本文提出了一个基于 NTS 高缓的选择性冲突预测高缓管理策略,并据此设计了相应的高缓结构,称作选择性冲突预测高缓.

2 NTS 高缓方案

如图 1 所示,NTS 高缓将片上高缓存储器分为并行的两个部分:一个为大容量 DM 高缓(cache A),另一个为容量相对较小的全相联高缓(cache B). 为了有效分析冲突型缺失产生的原因,文献[3]将其分为 T 型冲突和 NT 型冲突,其定义如下:

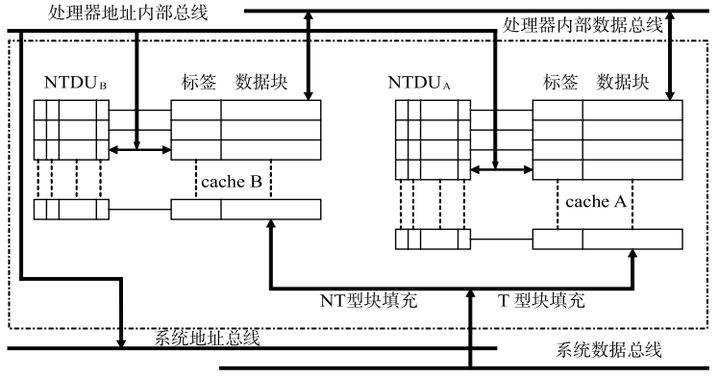

图1 NTS 高缓结构示意图

定义 1:当多于一个 T 型块映射到高缓同一组,并在一定时段内被反复访问所造成的缺失称为 T 型冲突;

定义 2:映射到高缓同一组的数据块中至少有一个为 NT 型块,由此块而来的访问冲突称为 NT 型冲突.

NTS 高缓的管理思想就是希望将 T 型块和 NT 型块分配到不同的高缓存储体中:将 T 型块尽量保存在 cache A 中,而将那些可能引发 NT 型冲突的 NT 型数据块都存储在 cache B 中,以避免此类冲突对命中率的影响.

NTS 高缓的控制逻辑需用的附加硬件包括非时间局部性数据监测单元(NTDU)以及数据块局部性监测单元(DU). NTS 高缓通过 NTDU 来监测每个高缓数据块的局部性特征. 片上高缓中的每个块都有其相应的 NTDU 项,块中每个字在其 NTDU 项中映射为 1 位,当相应字被访问过那么其映射位更改为 1. 当块中有两个以上的字被访问,则说明该块具备空间局部性. 此外还有 1 位称作 T/NT 位,用来表明数据块的时间局部性,即当高缓数据块中某个字被访问 1 次以上,则将相应 NTDU 中的 T/NT 位设置为 1.

为了将 T 型块和 NT 型块的分配区分开来,NTS 高

缓应用一个数据块局部性监测单元 DU 来为分配机构提供信息. 它是一个历史记录表,记录最近被替换掉的数据块的过去一个高缓生命期中表现的局部性信息. 每一记录项由两部分组成:被替换的数据块标签(用于查找)及其 T/NT 位(来自 NTDU).

NTS 高缓操作机理如下:

当处理器进行数据访问且相应数据块存在 cache A 或 cache B 中,数据及时送入处理器,同时相应的 NTDU 项被更改,即把被访问字的 NTDU 映射位设置为 1,或在该位已经为 1 时设置 T/NT 位为 1. 而当被访数据不存在二者中,即缺失时,待取数据块的标签作为索引查询 DU. 如果相应记录项存在,则根据其 T/NT 位来决定块分配方向:当 T/NT = 1,该块将被放入 cache A 中,否则被放入 cache B 中. 如果相应 DU 记录项不存在,被取来的块假定为 T 型块,放入 cache A 中. 同时在 DU 中生成从 cache A 或 cache B 中被替换的块的相应记录项.

3　选择性冲突预测高缓方案

基于 NTS 高缓结构的设计思想,本文提出一个新的高缓设计方案 - 选择性冲突预测高缓(SCP 高缓). 它包括一个容量大的 DM 高缓,另一个为容量相对很小的全相联高缓;与每一数据块联接的局部性监测单元(NTDU),以及一个冲突预测表(CPT) - 由两部分组成:CPT_{NT}和 CPT_T. 与 NTS 高缓不同在于本文提出的填充控制机构 CPT 将部分 T 型冲突块动态地存入全相联高缓中,从而减少 T 型冲突发生的几率. 此时,全相联高缓不仅做为部分 NT 型数据块的缓冲器,也是部分 T 型数据块的缓冲器.

SCP 高缓的操作过程如下:

当处理器进行数据访问且相应数据块存在高缓中,数据及时送入处理器处理. 同时更新相应 NTDU 项,即如果访问字对应 NTDU 位为 1,则将 T/NT 位置 1;否则,只将其 NTDU 位置为 1,而 T/NT 位保留原值. 当被访数据不存在于高缓中,即缺失时,将待取数据块的地址作为索引查询 CPT. 如果相应记录项不存在,则待取数据块预设为 T 型并将替换 DM 高缓中同组数据块;如果相应记录项存在,则待取数据块设为 NT 型并将填充全相联高缓. 当缺失数据取回,则按 CPT 查询命中与否进行分配. 即当 CPT 命中时,该块将被放入全相联高缓中,否则被放入 DM 高缓中. 同时 NTDU 相应项被更新,即 T/NT 位及所有映射位都清零,以备重新估计该块的局部性.

本文提出的方案与 NTS 高缓区别在于:在 NTS 高缓中高缓块或被预测为 T 型或是 NT 型,基于其上一次在片上一级高缓(L1 高缓)中表现的复用行为,然后 T 型块将分配到 DM 高缓,而 NT 型块分配到全相联高缓中以避免产生可能的 NT 冲突. 本文的方案中,全相联高缓中不仅分配 NT 型块,也将分配一定数量的 T 型块. 所有分配到全相联高缓中的数据块在 CPT 中都存有其条目,而 T 型块必须取得许可才可被记录到 CPT 中,也就是说,只有那些刚刚被驱逐出 DM 高缓的才可以被记录. NT 型块没有上述限制.

NTS 高缓中的 DU 只记录最近从 L1 高缓中被驱逐出的块,通过观察可知事实上只有 NT 型块需要记录,此外当冲突发生在 T 型块之间时,NTS 高缓提高命中率的幅度有限. 因此,SCP 高缓采用两个表来记录驱逐块,一个表记录从 L1 高缓中驱逐出的 NT 型块,另一个只记录从 DM 高缓中驱逐出的 T 型块. 两个表共同构成 CPT. 这种结构将有效降低 T 型块之间的冲突引发的缺失率.

在仿真试验我们发现,如果想要达到理想的结果,即较高的命中率,还要考虑以下两方面问题:CPT 内部子块的比例,以及 CPT_T 中 T 型标签的来源.

CPT_{NT}从本质上来看,就是 NTS 高缓 DU 的改进版,其执行的功能与 DU 是一样的. 充分保留 NTS 高缓减少 NT 型冲突的有效机制,是本文设计 CPT 的基础,因此原则上希望 CPT_{NT}尽量大一些. 由于在 NTS 高缓中,无论是 cache A 还是 cache B,都有相应的 NTDU,所以都能够产生 T 型标签,从而当 cache 驱逐发生时能够填充 CPT_T. 但是,无论某一数据块在程序运行过程中表现为 T 型还是 NT 型,它第一次被填入高缓中时都假设为 T 型(为了防止高缓启动填充时,充分利用 cache A 容量大的优势,从而减少单一的启动填充策略可能带来的大量不必要的缺失),并存入 cache A 中,并在其 cache A 生命期中判断其当次的局部性特征. 但是程序执行过程大部分 T 型块本身也只是在一段时间内被使用,过了这一段时间将不再使用,因此如何既将 T 型块在其使用期保留在高缓,而又能在必要时将其驱逐出去,是一个难以解决的问题. T 型块的使用期长短不一,而且也没有规律可循,因此只能采取经验估计.

经过分析可知,当某 T 型块存入 cache B 中时,如果继续表现为 T 型,那么当它再次被驱逐出高缓时,其使用期基本过去,因此对于以解决高缓冲突为目的的 CPT 而言,不应将其数据块标签填入 CPT_T 中. 这样也可以减少对其他正处于其使用期的 T 型块的影响. 因此 CPT_T 中 T 型标签应仅仅来自于 cache A 的驱逐块. 从仿真试验结果来看也证明了这一推断,事实上如果 CPT_T 也接纳 cache B 的 T 型标签,在一些基准测试程序中会大大增加高缓的缺失率.

4　实验仿真

4.1　仿真环境

本文采用了 SimpleScalar 工具集[12] 中的 sim-out-

order 仿真器做为仿真平台,将 mlcache 仿真器[13]取代 sim-outorder 中相关的高缓. 由于本实验只检验 L1 数据高缓的性能,因此指令高缓为理想化的. L2 及总线假设为理想结构. 仿真器和存储器系统的参数见表 1. 本文采用 SPEC95 标准测试程序集中 8 个典型的程序来进行仿真,其中除了 compress 程序采用训练数据集(train data set)做为仿真输入外,其他程序的输入文件均来自测试数据集(test data set). 所有的仿真程序都运行到结束.

表 1 处理器及存储器参数表

取指机构	每周期按程序顺序可取多达 4 条指令
转移预测	2084 – entry Bimodal 预测
发射机构	每周期无序发射达 4 条操作,16 – entry 重排序缓冲器,8 – entry 装载/存储队列
功能部件	4 个整数 ALU,4 个浮点 ALU,1 个整数乘/除部件,1 个浮点乘/除部件
数据高缓	回写式,写分配,每块为 32 个字节,4 读/写端口,无阻塞式访问
L2 高缓	L1-L2 256 位总线,数据带宽 32 字节/周期,L2 高缓设为无穷大,访问延迟为 18 周期,L1 到 L2 的访问为充分流水线式访问

4.2 性能标准

对于给定结构的高缓,评价其性能的首选标准为其存储器等效访问周期或存储器访问周期总数. 我们定义存储器等效访问周期为

命中率 × 命中周期数 + (1 – 命中率) × 等效缺失损耗周期数 (1)

这里我们假设直接映象高缓和全相联高缓命中周期同为 1 个时钟周期,而等效缺失损耗为 18 个时钟周期. 对于本实验,缺失率为

L1 高缓缺失数 / L1 高缓访问总数 (2)

为了便于比较,我们定义了加速比参量

目标结构的存储器等效访问周期/基准结构的存储器等效访问周期 (3)

总线交通量(bus traffic)— 即 L1 高缓与下一级存储器通过总线交换的数据量—是另一个重要的性能参数,对于整个处理器系统而言,总线交通量越小,系统的速度也会越快. 本文将由数据高缓申请的通过总线的数据总量(以百万字为单位)做为总线交通量的衡量标准. 为便于比较,定义相对总线交通量为

基准结构的总线交通量 – 目标结构的总线交通量 (4)

4.3 仿真结果与分析

本文以 L1 数据高缓为例进行仿真实验. 用于参考比较的高缓结构包括:DM 高缓、2 路组相联高缓、NTS 高缓、PCS 高缓和 CP 高缓. 为了比较目标高缓的性能,限定所有参与比较的扩充高缓容量固定为(8 + 1)KB,其中 DM 高缓容量为 8KB,全相联高缓为 1KB. 做为基准结构的 DM 高缓则分别为 8KB 和 16KB,此外还对 8KB 2 路组相联高缓的性能进行了仿真以便于对比. 所有仿真的高缓块长均为 32 字节. 所有的全相联结构的硬件组织都应用 LRU 替换算法.

4.3.1 缺失率 为了找到 CPT 内部 CPT_{NT} 与 CPT_T 容量大小的最佳比例,我们分别采用 $CPT_{NT}:CPT_T = 4:1$、$2:1$、$1:1$ 等进行对比试验,CPT 总容量不大于 NTS 高缓和 PCS 高缓的 DU 容量(设 DU 最大容量为 16 条). 与此同时 NTS 高缓及 PCS 高缓的 DU 大小,以及 CP 高缓的 CPT 也进行调整. 表 2 给出用于实验的不同高缓结构在运行 SPEC95 程序后的缺失率. 其中 SCP 高缓的 CPT 大小分别表示为 (16 + 16)/(16 + 8)/ (16 + 4)/(8 + 4)/(8 + 2),在括号里加号前为 CPT_{NT} 的条数,后面为 CPT_T 的条数. NTS 及 PCS 高缓后数字表示 DU 的条数. 而 CP 高缓后数字表示 CPT 的条数. DM 高缓后面标明容量大小. 8K2W 表示 8KB 2 路组相联式高缓结构.

实验结果显示在预测表容量相近的条件下,SCP 高缓的 $CPT_{NT}:CPT_T = 2:1$ 时性能最优,同时 NTS 高缓和 PCS 高缓中 DU 为 16 条时平均性能最佳,而 CP 高缓中 CPT 为 16 条时平均性能亦最佳. 其中 SCP:8 + 4 的平均缺失率为 4.99%,NTS:16 为 5.35%,PCS:16 为 4.99%,CP:16 为 4.97%. 同时我们可以看出在每个程序中 SCP 高缓缺失率均低于 NTS 高缓. 在 swim 中 SCP:8 + 4 比 NTS:16 降低 25%,8 个测试程序后缺失率比 NTS 高缓平均降低约 6.7%. 除了在 swim 中 SCP:8 + 4 缺失率高于 PCS:16 以外,其他均有所降低,8 个测试程序后缺失率与 PCS:16 相当. SCP:8 + 4 的缺失率在 5 个测试程序过后低于 CP:16,8 个测试程序后的平均缺失率略高于 CP:16 的平均缺失率,但在预测表的硬件面积开销上有所降低.

我们也能够看出 SCP:(8 + 4)的缺失率低于 16KB DM 高缓和 8KB 2 路组相联高缓,分别平均降低 19% 和 47%.

4.3.2 加速比

图 2 给出 SCP:(8 + 4)、NTS:16、PCS:16、16KB DM 高缓以及 8KB 2 路组相联高缓相对于 8KB DM 高缓的加速比. 值越大,相应高缓系统性能越好. SCP 高缓除了在 swim 中加速比小于 PCS 高缓外,在其他程序中均优于相比较的其他高缓结构.

4.3.3 总线交通量

表 3 给出以 16KB DM 高缓为基准结构,一些典型用于比较的目标高缓结构的相对总线交通量,单位为

百万字. 用于比较的高缓结构分别为同样结构中平均缺失率最低的(8 + 1) KB SCP:(8 + 4),(8 + 1) KB NTS:16,(8 + 1) KB PCS:16,以及 8KB 2 路组相联高缓. 如式(4)所示,表中数值是由 16KB DM 高缓总线交通量减去相应目标高缓的总线交通量所得,因此当此值为正数时,说明目标结构的总线交通量要低于基准结构,否则当相应值为负数则说明目标结构的总线交通量高于基准结构. 正值越大,目标结构的总线交通量越低. 仿真结果显示 SCP 高缓在多数基准测试程序运行结束后,总线交通量是最低的. 在 hydro2d 中,SCP 高缓的总线交通量要高于 16KB DM 高缓,但是其他高缓结构也很高,其中 SCP 高缓相对较低.

表 2　运行 8 个 SPEC95 基准测试程序后 14 个高缓方案的缺失率

	compress	gcc	li	ijpeg	perl	hydro2d	su2cor	swim
SCP:16 + 16	0. 0554	0. 0291	0. 0147	0. 0123	0. 0280	0. 1114	0. 0840	0. 0752
SCP:16 + 8	0. 0551	0. 0285	0. 0145	0. 0123	0. 0282	0. 1106	0. 0812	0. 0752
SCP:16 + 4	0. 0550	0. 0284	0. 0145	0. 0122	0. 0285	0. 1101	0. 0808	0. 0751
SCP:8 + 4	0. 0549	0. 0283	0. 0143	0. 0122	0. 0285	0. 1097	0. 0745	0. 0766
SCP:8 + 2	0. 0549	0. 0284	0. 0144	0. 0124	0. 0290	0. 1096	0. 0767	0. 0766
NTS:16	0. 0564	0. 0300	0. 0150	0. 0125	0. 0286	0. 1099	0. 0749	0. 1005
NTS:8	0. 0564	0. 0306	0. 0151	0. 0126	0. 0285	0. 1100	0. 0754	0. 1033
PCS:16	0. 0592	0. 0320	0. 0161	0. 0131	0. 0315	0. 1138	0. 0730	0. 0607
PCS:8	0. 0590	0. 0327	0. 0164	0. 0131	0. 0307	0. 1143	0. 0749	0. 1113
CP:16	0. 0550	0. 0284	0. 0143	0. 0123	0. 0280	0. 1099	0. 0736	0. 0752
CP:8	0. 0549	0. 0285	0. 0144	0. 0123	0. 0282	0. 1094	0. 0729	0. 0766
DM:8K	0. 0673	0. 0462	0. 0231	0. 0564	0. 0597	0. 1195	0. 0891	0. 4068
DM:16K	0. 0557	0. 0288	0. 0178	0. 0221	0. 0373	0. 1063	0. 0796	0. 1424
8K2W	0. 0563	0. 0300	0. 0147	0. 0448	0. 0288	0. 1112	0. 0768	0. 3904

表 3　4 个高缓方案的以百万字为单位的相对总线交通量

	compress	gcc	li	ijpeg	perl	hydro2d	su2cor	swim
SCP:8 + 4	0. 3	1. 4	16. 6	0. 6	2. 5	- 6. 3	18. 9	166. 8
8K2W	0. 1	- 0. 4	16. 4	- 2. 1	2. 4	- 8. 8	11. 4	- 144. 0
NTS:16	- 0. 007	- 0. 7	13. 3	0. 6	2. 4	- 6. 7	16. 5	104. 1
PCS:16	- 0. 3	- 3. 8	7. 6	0. 6	1. 7	- 15. 4	22. 8	187. 5

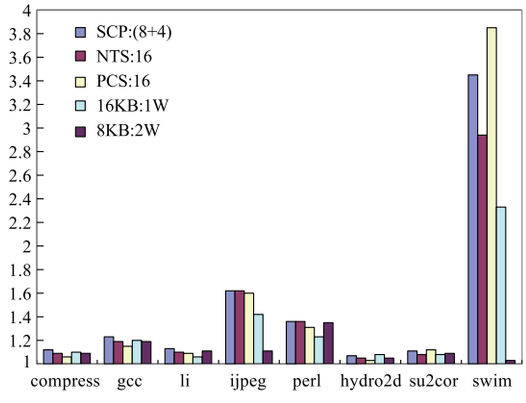

图2　五个高缓方案相对于8KB DM 高缓的加速比

5　结论

本文提出一种基于 NTS 高缓的改进方案 - SCP 高缓. SCP 高缓利用一个冲突预测表充分监测高缓中数据块局部性,有选择性地进行数据块的分配及替换,一方面通过 NT 型数据块的选择性填充以减少数据块 NT 型冲突,另一方面通过使 T 型数据延长高缓生命期,在一定程度上减少 T 型冲突量的方法,减小了直接映象高缓缺失率中占主要比例的冲突缺失,并且硬件开销与类似结构的 NTS 高缓方案相比有所降低. 从 SimpleScalar + mlcache 性能仿真平台的仿真结果可知,与相同容量的 NTS 高缓方案相比,平均缺失率降低约 6. 7%. 与相似结构的 CP 高缓相比,在平均缺失率相当的情况下,预测表硬件开销有所降低. 同时也能够看出(8 + 1) KB SCP 高缓的缺失率低于 16KB DM 高缓和 8KB 2 路组相联高缓,分别平均降低 19% 和 47%. 并且其总线交通量在同比高缓结构中也是最低的.

参考文献

[1] Tyson G, Farrens M, et al. A modified approach to data cache management [A]. Proceedings of the 28th Annual IEEE/ACM International Symposium on Microarchitecture

(MICRO-28)[C]. Ann Arbor, USA, 1995. 93 – 103.
[2] Rivers J A, Tam E S, Davidson E S. On effective data supply for multi-issue processors[A]. Proceedings of the 1997 International Conference on Computer Design (ICCD'97)[C]. Austin, USA, 1997. 519 – 528.
[3] Rivers J A, Tam E S, et al. Evaluating the performance of active cache management schemes[A]. Proceedings of the 1998 International Conference on Computer Design (ICCD'98)[C]. Austin, USA, 1998. 368 – 375.
[4] Jouppi N P. Improving direct mapping cache performance by the addition of a small full associative cache and prefetch buffers[A]. Proceedings of 17th Annual International Symposium on Computer Architecture (ISCA 1990)[C]. Seattle, USA, 1990. 364 – 373.
[5] Tse J, Smith A J. CPU prefetching: timing evaluation of hardware implementations[J]. IEEE Transaction on Computers, 1998, 47(5): 509 – 526.
[6] Rotenberg E, Bennett S, Smith J E. A trace cache micro-architecture and evaluation[J]. IEEE Transaction on Computers, 1999, 48(2): 111 – 120.
[7] Kharbutli M, Solihin Y. Counter-based cache replacement and bypassing algorithms[J]. IEEE Transaction on Computers, 2008, 57(4): 433 – 447.
[8] Davanam N, Lee B K. Towards smaller-sized cache for mobile processors using shared set-associativity[A]. Proceedings of the 7th International Conference on Information Technology[C]. Las Vegas, USA, 2010. 1 – 6.
[9] Qureshi M K, Thompson D, Patt Y N. The VWay cache: demand based associativity via global replacement[A]. Proceedings of 32nd Annual International Symposium on Computer Architecture (ISCA'05)[C]. Madison, USA, 2005. 544 – 555.
[10] Tam E S, Rivers J A, et al. Active management of data caches by exploiting reuse information[J]. IEEE Transaction on Computers, 1999, 48(11): 1244 – 1258.
[11] 李晓明,鲍东星,喻明艳,叶以正. 利用冲突预测方法的高缓组织方案[J]. 电子学报,2003,31(5):724 – 727.
Li Xiao-ming, Bao Dong-xing, Yu Ming-yan, Ye Yi-zheng. Cache management by using conflict prediction method[J]. Acta Electronica Sinica, 2003, 31(5): 724 – 727. (in Chinese)
[12] Burger D, Austin T M. Evaluating future processors: the SimpleScalar tool set. Technical Report #1342[R]. University of Wisconsin, 1997.
[13] Tam E S, Rivers J A, et al. Mlcache: a flexible multilateral cache simulater[A]. Proceedings of the 6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'98)[C]. Montreal, Canada, 1998. 19 – 26.

作者简介

李晓明 男,1969 年 2 月出生,黑龙江省哈尔滨人. 博士. 目前主要从事计算机体系结构、数字图像处理、无线传感器网络等研究工作.
E-mail: lixiaoming@hit.edu.cn

鲍东星 女,1969 年 3 月出生,黑龙江省哈尔滨人. 副教授. 主要从事计算机体系结构、传感器技术、人机交互技术以及数字图像处理等研究.
E-mail: eastarbox@163.com

刘晓为(通信作者) 男,1955 年出生,黑龙江省哈尔滨人. 教授,博士生导师,哈工大国家人才培养基地主任,哈工大微电子科学与技术系主任,微系统与微结构制造教育部重点实验室副主任, MEMS 中心主任,城市水资源开发利用(北方)国家工程研究中心环保工程无线网络测控技术研究所所长,国防 511 人才工程学术带头人,微纳米技术学会常务理事,总装微纳米技术专家组成员,《传感技术学报》与《传感器技术与微系统》编委副主任,《测控技术学报》编委,曾承担国防 973、总装重点基金、国家 863、国家自然科学基金等项目 20 余项,发表学术论文 200 余篇,100 余篇被 SCI/EI 检索,获得国家发明专利 23 项. 获省部级科学技术奖 4 项. 主要研究领域为:集成传感器技术、MEMS 与微系统、MEMS 微能源、纳米膜与纳机电技术、物联网技术.
E-mail: lxw@hit.edu.cn