

文章编号:1001-2486(2006)05-0084-06

一种低功耗预比较 TLB 结构*

侯进永,邢座程

(国防科技大学 计算机学院,湖南长沙 410073)

摘要:介绍了一种低功耗 TLB 结构。这种结构的思想是基于程序局部性原理,结合 Block Buffering^[1] 技术,并对 CAM 结构进行改造,提出一种预比较 TLB 结构,实现低功耗的 TLB。并且采用 SimpleScalar 3.0 模拟该 TLB 结构和几种传统的 TLB 结构的失效率。通过改进的 CACTI3^[2] 模拟结果显示:提出的 TLB 结构比 FA-TLB 平均功耗 × 延迟降低约 85%,比 Micro-TLB 降低 80%,比 Victim-TLB 降低 66%,比 Bank-TLB 降低 66% 以上。从而,所提出的 TLB 结构可以达到降低功耗的目的。

关键词:TLB;低功耗;CAM;block buffer

中图分类号:TP333.96 **文献标识码:**B

A Precomparison TLB Structure for Low Power

HOU Jin-yong, XING Zuo-cheng

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A structure of TLB for low power is introduced. The idea of the proposed TLB is based on the spatial locality, which is the result of combining with the block buffering technology and adjustment of the CAM structure. All of these make the TLB for low power. With Simple Scalar 3.0, a simulation of the proposed TLB and some traditional TLB structures were made to observe the miss ratio. The simulation results from the modified CACTI3 show that the proposed TLB structure can reduce power * delay about 85%, 80%, 66%, and 66%, compared with a FA-TLB, a micro-TLB, a victim-TLB, and a bank-TLB. Therefore the proposed TLB can achieve low power.

Key words: TLB; low power; CAM; block buffer

TLB(translation look-aside buffer)是微处理器中用来做虚地址到物理地址转换的快速 Buffer^[3]。目前为了提高 TLB 性能,普遍采用全并行比较结构,同时让 TLB 具有更多的入口数、增加页大小、以及支持多种页大小技术^[4],但这引起了额外延迟的增大和功耗的增加。比如在 StrongARM110 中 TLB 功耗占整体功耗的 17%之多^[5]。

目前有很多的方法用来降低 TLB 功耗,比如较早的 Filter-TLB^[5]、Banked TLB^[6]、victim-TLB^[7]等,这些方法的特点是通过一些简单的硬件控制来降低 TLB 比较时的入口数。近来出现的 Selectively Accessing TLB^[1]、Banked-Promotion TLB^[4]等等,也是通过硬件控制来降低访问 TLB 时参加比较的入口数,但是这些方法的缺点是:在这些结构中,对于一个 TLB 体的所有入口,全部参与比较,并且与所降低的功耗相比,所采用的硬件开销较大。还有一些采用软件的方法来降低 TLB 功耗,比如编译直接产生物理地址^[8]等,这些方法的缺点是需要操作系统的支持。

1 预比较 TLB

1.1 思想

为了降低功耗,本文所提出的 TLB 结构充分利用了程序空间局部性原理,由一个 Block Buffer 与采用预比较 CAM 的 TLB 体组成。Block Buffer 用来缓冲当前命中的比较项,这是基于程序局部性原理,顺

* 收稿日期:2006-07-06

基金项目:国家自然科学基金资助项目(90207011);国防科技大学预研基金资助项目(JC03-06-007)

作者简介:侯进永(1979—),男,硕士。

序访问的高可能性(在 SPEC95 一些测试程序组中,达 50% 以上^[1]),来实现 TLB 访问的低功耗。

本文的 CAM 结构与传统的 CAM 结构不同,分成两级比较,第一级比较称之为预比较,可与 Block Buffer 比较同步进行,也可在 Block Buffer 的比较之后进行。通过第一级的较少位参加的预比较,快速地筛掉不会命中的项,确定出 CAM 中可能会命中的项来参加第二级比较。这样通过第一级很小位宽的比较来大大降低全部比较时的项数,从而降低整个 TLB 的功耗。

1.2 Block Buffering 技术

Block Buffering 技术被系统开发者广泛采用。这种技术利用程序空间局部性支持低功耗系统^[1]。如果应用程序具有较高的空间局部性,Buffer 的命中率上升,存储器功耗就可以得到降低。

1.3 预比较 TLB 结构

本文的预比较 TLB 结构是由传统的 TLB 结构结合 Block Buffer 改进而来的。整体结构如图 1 所示。为实现功能增添了部分控制逻辑,CAM 单元的内部结构与传统 CAM 也有所不同,采用两级比较。

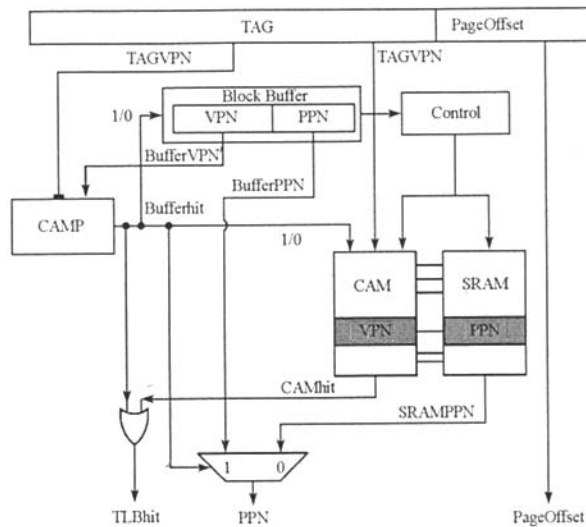


图 1 预比较 TLB 结构

Fig.1 Structure of Precomparison TLB

在该 TLB 结构中,对于 TLB 的访问(写入和地址转换),总是先访问 Block Buffer,这样能够更充分地利用程序空间局部性原理。因此高性能可以通过 Block Buffer 的快速访问得到,并且如果 Block Buffer 命中则由于不用访问 TLB 体使得 TLB 的功耗得到很大程度上的节约。同时当 Block Buffer 失效时,预比较 CAM 结构同样能够降低访问 TLB 体时的功耗。CAM 的第二级比较(如果需要)可以在同一个周期或下一个周期完成。

2 运算模型

2.1 写入替换运算

本文中 TLB 写入与传统的 TLB 略有不同,由图 1 中 Control 单元进行写入替换控制,替换策略是新载入的项替换掉 Block Buffer 中的项,并且将 Block Buffer 中原来的项保存入 CAM 中,因为该项是最近访问的或新写入的。CAM 中项的替换策略是 FIFO,也是由 Control 来控制替换,产生 CAM 和 SRAM 写入的相应地址。

2.2 比较运算

TLB 比较时,TAG 区的 VPN 与 Block Buffer 的 VPN 相比较,此时有两种情况,Buffer 命中和不命中。

2.2.1 Block Buffer 命中

此时, Bufferhit 信号为 1, Block Buffer 将缓存的 PPN 输出, 而 CAM 则不进行比较, 由此而减少了功耗, 整个 TLB 的命中信号 TLBhit 为高, 表示 TLB 命中, 并且 Bufferhit 驱动一个两路选择器, 输出物理页号 BufferPPN 和页内偏址 PageOffset 到 Cache。由于程序空间局部性原理, 并且顺序访问的高可能性, 使得这种结构对于节约功耗很有效。如果 Block Buffer 没有命中, 将进入 CAM 进行比较访问。

2.2.2 Block Buffer 不命中

此时根据 CAM 的结构有两种情况, 如果 CAM 的预比较与 Block Buffer 比较同步进行时(这种情况下延迟小), 则进行 CAM 的第二级比较, 确定出 CAM 命中情况; 如果 CAM 的预比较与 Block Buffer 比较没有同步进行, 此时由 Bufferhit(为 0)来对 CAM 进行预充电控制, 并且开始 TLB 的两级比较。当 CAM 二级命中时, 此时输出 SRAM PPN 就是转换成功的物理页号 PPN, 并将其和页内偏址 PageOffset 送到 Cache, 且 TLBhit 为高, 表示 TLB 命中。并且将该项与 Block Buffer 中的项置换, 因为该项下次的命中率在 50% 以上。当 Block Buffer 和 CAM 均失效时, 由 Control 单元控制进行失效写替换。

3 CAM 结构

CAM 结构是本 TLB 设计的一个关键, 降低 TLB 的功耗最重要的一个方面就是降低 CAM 的功耗。目前有很多方法降低 CAM 的功耗。近来, 一种 PB-CAM^[9] 结构的方法在降低 CAM 功耗方面效果明显。这种方法采用了参数预先比较的方法来降低 CAM 比较时的入口数, 并且采用静态伪 nmos 的字线结构来做预充电电路, 还有一个特点是没有采用时钟来控制预充电。这种结构的缺点是参数提取电路复杂, 并且延时大, 对性能影响较大。

本文采用较少的部分位先比较的方法来实现预比较, 从而确定出第二级比较的项。假设 CAM 的字长(VPN 的位宽长度)为 n , 而预比较的位宽长度为 m , 第二级比较的位宽长度为 $n - m$ 。

根据性能和功耗特性方面的不同, 本文提出了两种相应的 CAM 结构, 一种是预比较和 Block Buffer 比较同时进行的 CAM 结构(图 2), 这种结构延迟小, 特点是二级比较的预充电电路除了由预比较的结果控制外, 还由 Bufferhit 来控制。即当 Block Buffer 不命中(Bufferhit = 0)时, 二级比较电路才预充。预比较和 Block Buffer 比较同时进行, 预比较的预充电电路可以是时钟信号或 TLB 地址转换信号等的组合控制。由于当 Block Buffer 命中时, 预比较是没有意义的, 因此这种结构有部分功耗损失。但是当 Block Buffer 不命中时, CAM 可以立即进入二级比较, 使得延迟较小。

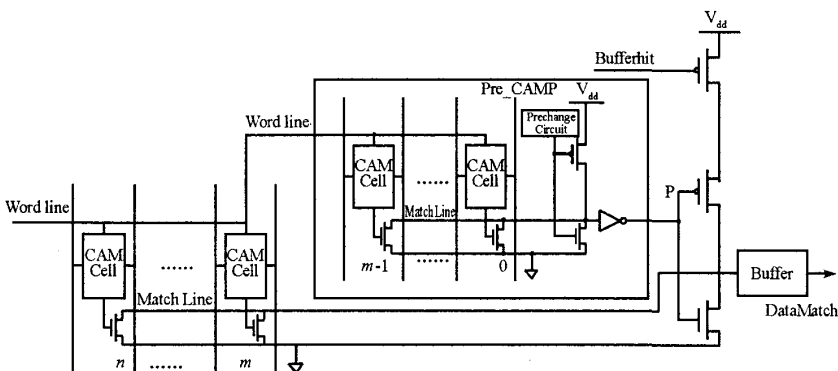


图 2 CAM 结构 1

Fig.2 First structure of CAM

另外一种预比较在 Block Buffer 比较之后进行的 CAM 结构(图 3), 这种结构比第一种结构功耗略小, 特点是预比较电路和二级比较电路都受 Bufferhit 控制。当 Block Buffer 命中时, 预比较和二级比较电路都不预充, 此时比第一种结构节约了预比较电路的功耗。当 Block Buffer 不命中时, 此时才由 Bufferhit (= 0)控制预比较电路和二级比较电路的预充, 延迟比第一种结构大。

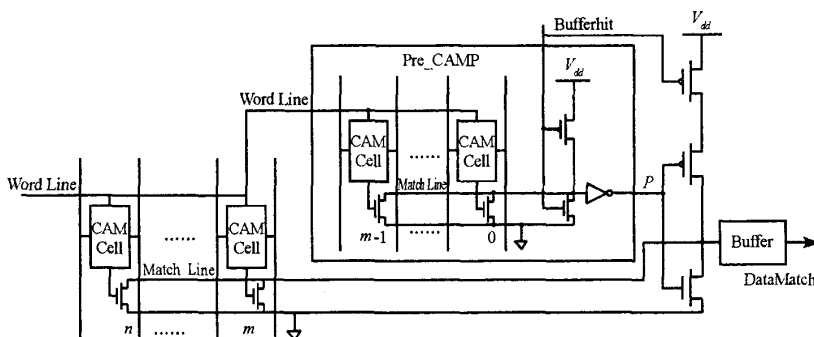


图3 CAM 结构 2

Fig.2 Second structure of CAM

这两种 CAM 结构中,每个入口的预比较和二级比较的 CAM 位单元组成了整个 CAM 的字结构,即两级比较的写入选通字线 Wordline 是相同的,连接在一起。在这两种结构中,采用 Buffer 整形^[9]预比较电路求经过一个反相器输出预比较的匹配情况,此时结果是与真正的结果相反的,即当 TAG 的低 m 位与 CAM 中的相应位匹配时,预比较结果是 0,这是为了驱动二级比较电路的预充 pmos 管,而二级比较电路接一个 Buffer 整形,得到 CAM 中整个字的命中情况。

4 分析模拟

4.1 参加比较的位单元数分析

假设所设计的 TLB 结构中包含一个 Block Buffer 和一个入口数为 y 的 CAM,并且 VPN 是 n 位,CAM 的预比较的位宽是 m 位。可以分析出各种情况下参加比较的位单元数。

首先考虑 m 的选取对于二级比较的影响。之所以选择低 m 位用来做预比较,也是根据程序的空间局部性原理,程序执行时的 VPN 在较短时间内是比较靠近的,即 VPN 的高位相同或相近,而低位的区别比较大,因此如果选择低位,就能够更好地筛掉 CAM 中不参加比较的项。理论上在平均情况下,当 m 增加 1,参加第二级比较的项就减少一半(这要在 $m \ll n$ 的情况下)。由于快速的 CAM 结构一般都采用并行的 NOR 结构,当比较的数据与参考数据不一致时,通过 nmos 管(如 CAM 结构图中最下面的并行 nmos 管)下拉 Match line,因此当在 CAM 比较的时候,提前筛掉那些肯定会下拉 Match line 的项,使得在第二级比较时下拉的 nmos 管数(也就是位单元数)会很大程度地减少,从而能够节约功耗。

其次讨论两种 CAM 的结构参加一次 TLB 地址转换访问而进行比较的位单元数。结果如表 1 所示。

表 1 CAM 参加比较位数分析

Tab.1 Analysis of comparison bits in CAM

Buffer 命中情况	CAM 结构 1	CAM 结构 1	CAM 结构 2	CAM 结构 2
	参加比较位数	节约位数	参加比较位数	节约位数
命中	$n + y * m$	$(y - 1) * n - y * m$	n	$(y - 1) * n$
	$n + y * m +$	$(y - 1) * n - y * m -$	$n + y * m +$	$(y - 1) * n - y * m -$
不命中	$\lceil \frac{y}{2^m} \rceil * (n - m)$	$\lceil \frac{y}{2^m} \rceil * (n - m)$	$\lceil \frac{y}{2^m} \rceil * (n - m)$	$\lceil \frac{y}{2^m} \rceil * (n - m)$

通过分析可以发现,在对于目前 TLB 主要的应用规模范围(入口数从 32 到 128、位宽从 30 到 128)内, m 取 5 时,节约的位数比例(节约位数/参加比较位数)最高,并且节约比例理论上随 n 和 y 成正比增长。

通过分析可以得到平均情况下比较时节约的比较位数和节约比例公式如下:

节约位数 = (buffer 命中时节约位数) * 命中率 + (buffer 不命中时节约位数) * 失效率

节约比例 = 节约位数 / (入口数 * 字位宽)

图4是128个入口、VPN为64位的该TLB结构取m从1~20时的平均情况下节约比例变化。可以看出,该结构在二级比较时节约的位数相当可观,达70%以上,最高可达90%。

由于这些位属于不匹配的项,在传统的全相联FA-TLB中,这些位对Match line下拉,消耗了大部分能量。而在本文的预比较TLB结构中,这些位不参加比较,从而节约了相当大的功耗。

4.2 模拟

采用4个参数来衡量TLB的性能:失效率、平均访存延迟、功耗和功耗×延迟。对采用两种CAM结构的TLB结构进行模拟测试,整个TLB访问是在一个周期完成。

通过SimpleScalar 3.0对相同主体规模入口数(32)的传统的TLB及本文预比较TLB结构的进行模拟,测试TLB的失效率,采用的程序组如图5所示。本TLB结构的整体失效率比与相同入口数的FA-TLB略小(图5)。这是因为本文TLB结构多一个buffer,相当于比相同入口数的FA-TLB多一个入口,所以失效率略小。

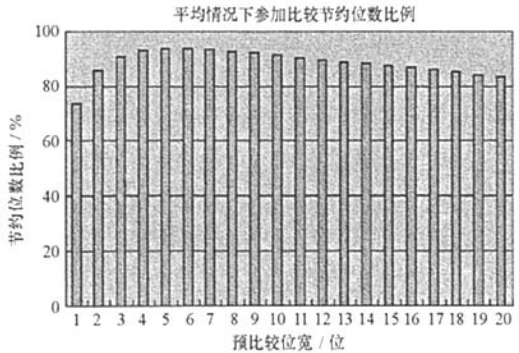


图4 节约比例
Fig.4 Economy ratio

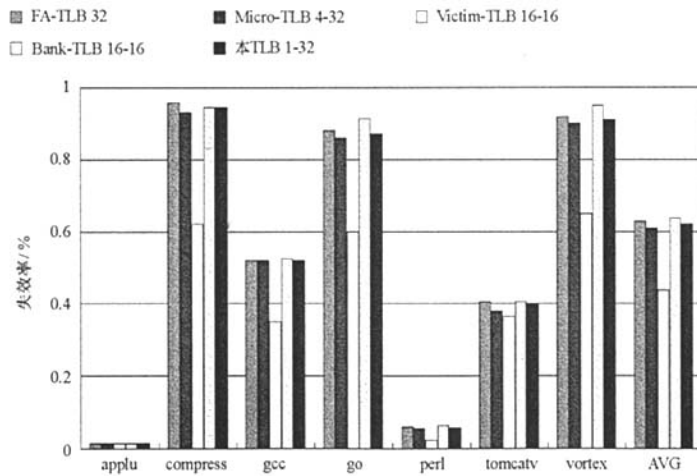


图5 传统的TLB与本文TLB失效率比较
Fig.5 Miss ratio comparison between traditional TLBs and precomparison TLB

模拟结果显示,Block buffer的平均命中率达52%以上,充分说明了程序局部性原理,这为采用本结构节约TLB功耗提供了保证。

采用改进的CACTI*模拟这些TLB结构,模拟得到0.8μm工艺下访问TLB访问命中功耗、命中延迟和失效功耗、失效延迟情况。本文取

$$TLB的功耗 = 命中率1 * 命中1功耗 + 命中率2 * 命中2功耗 + 失效率 * 失效功耗$$

其中,命中率1是buffer命中率,命中1功耗是此时的命中功耗,命中率2是buffer失效而TLB主体(除去buffer)命中率,命中2开销是此时命中功耗,失效率是整个TLB的失效率,失效功耗是整个TLB失效时的功耗开销。

* CACTI是HP公司WRL实验室开发的集成了访问延迟和功耗、面积模型的微体系结构及Cache模拟器,提供了详细的Cache结构、功耗、面积、访问时间等模型,并且考虑了工艺因素,是目前模拟Cache性能的主要模拟器之一。

图 6 列出了 4 种传统的 TLB 与本文 TLB 两种结构的性能比较,并考虑了命中率、失效率的平均加权功耗、平均加权延迟和功耗 \times 延迟。加权值是由改进的 CACTI3 模拟的结果与图 5 中的失效率加权乘得到的值。

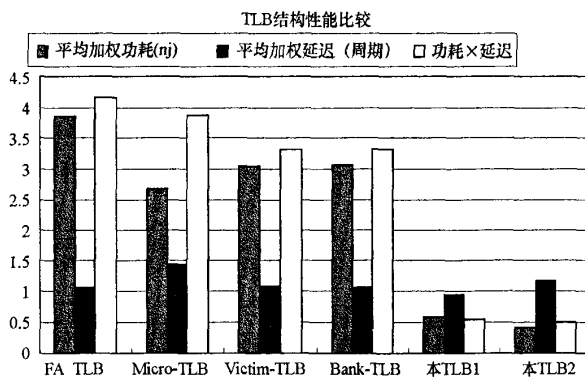


图 6 多种 TLB 结构性能比较

Fig.6 Performance comparison of manifold TLB

可以看出,本文的两种 TLB 结构比 FA-TLB 平均功耗 \times 延迟降低约 85%,比 Micro-TLB 降低 80%,比 Victim 降低 66%,比 Bank-TLB 降低 66% 以上。

5 结论

提出了一种低功耗高性能 TLB 结构。这种结构的思想是基于程序局部性原理,结合 Block Buffering 技术,并对 CAM 结构进行改造,提出一种预比较 TLB 结构,从而降低 TLB 功耗。采用 SimpleScalar 3.0 模拟该 TLB 结构和几种传统的 TLB 结构的失效率。通过改进的 CACTI3 模拟结果显示,本文介绍的 TLB 结构比 FA-TLB 平均功耗 \times 延迟降低约 85%,比 Micro-TLB 降低 80%,比 Victim-TLB 降低 66%,比 Bank-TLB 降低 66% 以上。所提出的 TLB 结构可以达到降低功耗的目的。

参考文献:

- [1] Min J H, Lee J H, Jeong S W, et al. A Selectively Accessing TLB for High Performance and Lower Power Consumption[A]. IEEE 2002, 2002.
- [2] Shivakumar P, Jouppi N P. CACTI 3.0: An Integrated Cache Timing, Power and Area Model[R]. Compaq WRL Research Report, 2001.
- [3] Austin T M, Sohi G S. High-bandwidth Address Translation for Multiple-issue Processors[A]. Proceedings of the 23rd ACM Int'l Symp. on Computer Architecture[C], 1996: 158 - 167.
- [4] Lee J H, Lee J S, Jeong S W, et al. A Banked-promotion TLB for High Performance and Low Power[A]. In: ICCD[C], 2001: 118 - 123.
- [5] Choi J H, Lee J H, Park G H, et al. An Advanced Filtering TLB for Low Power Consumption[A]. In: Proceedings of the 14th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD.02)[C], 2002.
- [6] Manne S, Klauser A, Grunwald D, et al. Low Power TLB Design for High Performance Microprocessors[R]. Univ. of Colorado Technical Report, 1997.
- [7] Jouppi N P. Improving Direct-mapped Cache Performance by the Addition of a Small Fully Associative Cache and Prefetch Buffers[A]. In: 17th ISCA [C], 1990: 364 - 373.
- [8] Kadayif I, Sivasubramaniam A, Kandemir M, et al. Generating Physical Addresses Directly for Saving Instruction TLB Energy[A]. Proceedings of the 35th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO - 35)[C], 2002.
- [9] Lin C S, Chang J C, Liu B D. A Low-power Precomputation-based Fully Parallel Content-addressable Memory[J]. IEEE Journal of Solid-state Circuits, 2003, 38(4).

