低功耗高速缓存无效缓存路访问混合过滤机制研究*

范灵俊[1,2+], 徐远超[1,5], 施巍松[3], 范东睿[1], 娄杰[4]

[1](中国科学院计算技术研究所 计算机体系结构国家重点实验室,北京 100190)

[2](中国科学院研究生院,北京 100049)

[3](韦恩州立大学 计算机科学技术系,美国)

[4](英特尔公司,北京, 100190)

[5](首都师范大学,信息工程学院,北京,100048)

Low Power Cache Architectures with Hybrid Approach of Filtering Unnecessary Way Accesses*

FAN Ling-Jun[1,2+], XU Yuan-Chao[1,5], SHI Wei-Song[3], FAN Dong-Rui[1], LOU Jie[4]

[1](State Key Laboratory of Computer Architecture, ICT, CAS, Beijing 100190, China)

[2](Graduate University of Chinese Academy of Sciences, Beijing, 100049, China)

[3](Department of Computer Science, Wayne State University, Detroit MI-48202, USA)

[4](Intel Corporation, Beijing, 100190, China)

[5](College of Information Engineering, Capital Normal University, Beijing 100048, China)

+ Corresponding author: Phn: +86-10-62600611, Fax: +86-10-62600600, E-mail: fanlingjun@ict.ac.cn

Received 2012-07-04; Accepted 2012-09-16

Abstract: Power has been a big issue in processor design for several years. As caches account for more and more CPU die area and power, this paper presents using filtering unnecessary way accesses to reduce dynamic power consumption of caches shared by instruction and data. Our methods include using Invalid Filter, which could eliminate accesses to cache ways contained invalid blocks, and I/D Filter, which could eliminate accesses to cache ways contained instruction/data access type mismatch blocks, and Tag-2 Filter, which could eliminate accesses to cache ways contained tag lowest 2 bits mismatch blocks. Since the methods reducing the activities happened in cache architecture, dynamical CPU power could be significantly decreased. In the paper, we also propose combing the above methods together, which is called Invalid+I/D+Tag-2 Filter, in an attempt to achieve better power saving results. Our evaluations show that, we could obtain 19.6%-47.8% (which is on average 34.3%)improvement on a 64K-4way set-associative cache and 19.6%-55.2%(which is on average 39.2%) improvement on a 128k-8way set-associative cache comparing to Invalid+I/D Filter, and 6.1%-27.7%(which is on average 16.6%)improvement on a 64K-4way set-associative cache and 6.9%-44.4%(which is on average 25.0%) improvement on a 128k-8way set-associative cache comparing to Invalid+Tag-2 Filter, respectively.

* 本文得到国家“973”重点基础研究发展规划项目基金(2011CB302501), 国家杰出青年科学基金(60925009), 国家自然科学基金创新研究群体科学基金(60921002), 国家自然科学基金青年基金项目(61100013), 国家自然科学基金青年基金项目(61202059), 北京市科技新星计划(2010B058), 华为资助课题(YBCB2011030)资助

作者简介: **范灵俊**(1983－),男,湖北老河口人,博士研究生, CCF 学生会员,主要研究方向为低功耗高速缓存设计; **徐远超**(1975－),男,博士,讲师,主要研究领域为计算机系统结构; **施巍松**(1974－),男,博士生导师,副教授,主要研究领域为绿色计算; **范东睿**

Key words: Set-associative Cache; Dynamical Power; Invalid Filter; I/D Filter; Tag-2 Filter

摘 要: 近年来，功耗是处理器设计领域的关键问题之一。由于片上缓存占有了越来越多的 CPU 芯片面积和功耗，本文提出了通过过滤不必要的缓存路访问来降低缓存动态功耗的方法。我们的方法包括采用无效访问过滤器（Invalid Filter）来消除对含无效数据块的缓存路的访问；采用指令数据访问过滤器（I/D Filter）来消除对与访问类型（指令或数据）不匹配的数据块所在的缓存路的访问；以及采用 tag 低位过滤器（Tag-2 Filter）来消除对 tag 低位不匹配的数据块所在的缓存路的访问。本文提出将以上三种方法合并，称为 Invalid+I/D+Tag-2 Filter，以期取得更好的效果。实验表明，与 Invalid+I/D Filter 相比，Invalid+I/D+Tag-2 Filter 在 64KB 4 路组相联缓存上可以取得 19.6%-47.8% (平均 34.3%)的效果提升，在 128KB 8 路组相联缓存上可以取得 19.6%-55.2% (平均 39.2%)的效果提升；与 Invalid+Tag-2 Filter 相比，Invalid+I/D+Tag-2 Filter 在 64KB 4 路组相联缓存上可以取得 16.1%-27.7% (平均 16.6%)的效果提升，在 128KB 8 路组相联缓存上可以取得 6.9%-44.4% (平均 25.0%)的效果提升。

关键词: 组相联缓存; 动态功耗; 无效访问过滤器; 访问类型过滤器; Tag 低位过滤器
中图法分类号: TP301 文献标识码: A

1 引言

近年来，功耗是处理器设计领域被关注的热点问题之一。DVFS（Dynamic voltage and frequency scaling）技术是最常被采用的节省 CPU 功耗的技术，但通过 DVFS 降低功耗目前正遭遇了收益递减律[1]。此外，文献[5]的研究也表明，多核处理器核数的不断上升，将带来芯片的功耗受限，很快会遇到 dark silicon 问题。即，所有处理器核都工作起来将导致芯片的功耗不可忍受。因此，我们需要找到一种新的有效并且通用的降低 CPU 功耗的方法。

得益于程序访存的局部性，从 1980 年代开始，缓存在填补处理器速度和内存速度之间的鸿沟中扮演了重要的角色，并带来了处理器性能的不断提升。因为大多数应用程序都表现出了良好的数据访问的空间/时间局部性，对于一个特定的处理器结构，越大的缓存往往预示着更好的程序性能。在现代处理器设计中，多种多样的缓存结构和缓存层次被设计和采用，来应对“存储墙[7]”问题。组相联缓存是最基本的结构之一，由于其能很好地降低冲突失效，从而得到了最为广泛的应用。

此外，在多核/众核处理器领域，集成更多的处理器核必然需要更多的片上缓存来供数，以保证可以带来不断的性能提高。因此，片上缓存占用了越来越多的芯片面积[3][4]。然而，更大的缓存也带来了更多的功耗，有的甚至达到了处理器总功耗的 40%-50%[2]。

本文，为了降低 CPU 的动态功耗，我们提出了不必要缓存路访问的混合过滤方法，包括无效访问过滤器（Invalid Filter），指令数据访问类型匹配过滤器（Instruction/Data Filter, I/D Filter）和 Tag 低位匹配过滤器（Tag-2 Filter）。它们都基于组相联缓存结构，混合后的方法可以在指令数据共享的缓存中得到应用。针对每次访问，Invalid Filter 可以提前检查含有无效数据块的缓存路，I/D Filter 可以提前检查与本次指令或数据访问类型不匹配的缓存路，Tag-2 Filter 可以提前检查 Tag 最低 2 位不匹配的缓存路。符合以上三种情况，被以上过滤器提前检测出的缓存路，在接下来的操作中将不被使能（disabled）。以上方法由于消除了 cache 结构中不必要的活跃行为（activities），从而将有效降低 CPU 的动态功耗。

实验表明，对于不同的应用程序，采用 Invalid+I/D Filter，在 64K-4way 组相联 Cache 结构上 36.65%-65.88%的无效路访问可以被消除，在 128K-8way 组相联 Cache 结构上上述结果达到了 46.49%-82.94%；而采用 Invalid+Tag-2 Filter，在 64K-4way 组相联 Cache 结构上 59.65%-69.32%的无效路访问可以被消除，上述结果在 128K-8way 组相联 Cache 结构上测试时也提高到了 67.86%-84.66%。此外，我们提出的合并方法（Invalid+I/D

(1979－),男,博士生导师,副研究员,主要研究领域为众核处理器设计;**娄杰**,男,博士,主要研究领域为计算机系统结构

Filter+Tag-2 Filter）的实验效果比以上两者都好，文章第四部分将详细介绍。同时，采用 Cacti6.0 模拟器进行的测试表明，实际应用程序运行在以上 Cache 结构上也获得了同等的能耗降低。

本文接下来组织如下：第二部分讨论相关工作；第三部分介绍我们的方法及无效缓存路访问过滤机制的逻辑结构；第四部分给出具体的实验结果；第五部分对全文进行总结。

2 相关工作

作为处理器设计领域的研究热点之一，降低缓存的功耗近年来得到了学术界持续的关注[8-18]。以下，我们将讨论和比较一些常见的通过减少缓存中不必要行为来降低缓存功耗的方法。

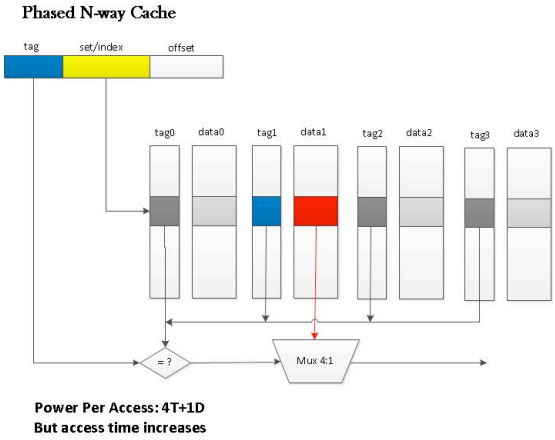

Fig.1 Phased cache[17].

图 1. Phased cache[17].

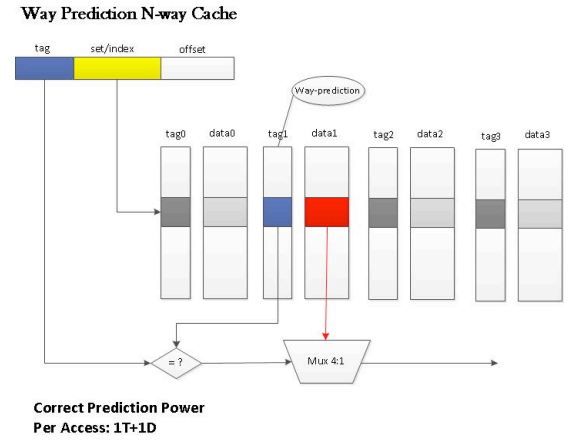

Fig.2 Way-prediction cache[9].

图 2. Way-prediction cache[9].

分段访问缓存（Phased caches）[7]，如图 1 所示，先访问 tag，再访问数据。访问数据时，只访问命中的那一路。此方法的缺点是，每一次缓存访问都多了一拍，增加了访存延迟。以 4 路组相联缓存为例，命中的情况下，每次访问的动态功耗为 4T+1D（4 路 tag 访问和 1 路 data 访问）。

如图 2 所示，路预测缓存（way-prediction cache）[9]提前预测可能匹配的一路，然后只访问这一路。每次访问的动态功耗为 1T+1D，即，1 路 tag 访问和 1 路数据访问。但是，当预测错误时，访问要再发生一次。

文献[14]提出了采用指令/数据过滤器来减少缓存中缓存路的访问操作从而降低功耗的方法。采用指令/数据过滤器，每次访问的功耗为 hT+hD，其中 h 是与本次访问类型（指令或数据）符合的缓存块数量。此方法单独作用效果有限，因此，本文我们提出的混合过滤器将此方法作为其中的一部分。

在文献[16]中，Keramidas 等利用 cache decay 来降低组相联缓存的动态功耗，称作 way-selection cache。如图 3 所示。Cache decay 技术最初由文献[11]提出，是用来降低静态功耗的。方法是，当一个缓存块里的数据不再被用到时，将其所在的缓存路关掉（power off）。此缓存结构，或者利用 decay bits，或者利用 decay bloom filters 来追踪和判断一个缓存块是处于 live 状态，还是处于 dead 状态，然后当访问发生时，只访问处于 live 状态的缓存块。然而，用 decay bits 的方法会带来低的准确度，用 decay bloom filters 会引入过多的硬件开销。在 way-selection cache 中，每次访问的动态功耗为 yT+yD，其中，y 是所访问的缓存组中 live 的缓存块数量。

Zhang 等在文献[10]中提出了 way-halting cache，一种 4 路组相联缓存，将每一路 tag 位的最低四位存在一个全相联的缓存中，称为 halt tag 阵列。halt tag 阵列用最低 4 位的提前比较来预先检查哪些路是不会被匹配的，在随后的访问中，这些路就被屏蔽掉，进而节省了访问时产生的功耗。在 way-halting cache 中，每次访问的动态功耗是 zT+zD，其中 z 取决于可能匹配的缓存路的数量。

表 1 中，我们将现有的一些低功耗缓存结构进行了总结和对比。对于本文提出的采用 Invalid+I/D+Tag-2 Filter 方法的 way-filtering cache，每次访问的动态功耗为 xT+xD，其中 x 为数据有效、访问类型匹配并且 tag

低 2 位匹配的缓存块数量。

Table.1 Comparison of different low power cache architectures

表 1. 不同低功耗缓存结构的对比.

缓存结构	只访问	每次访问功耗	硬件开销	性能损失
Way-prediction	predicted line	1T+1D	more	wrong prediction
Way-selection	live lines	yT+yD	more	wrong determination
Phased Cache	hit line	4T+1D	little	one more cycle
Way-halting	may match lines	zT+zD	more	one more cycle
I/D Filtering	I/D match lines	hT+hD	little	one more cycle
Way-filtering	Valid, I/D type and tag-2 match lines	xT+xD	little	one more cycle

3 我们的方法

得益于程序访存的局部性原理，片上缓存在现代处理器性能的提高中扮演了重要的角色。组相联缓存结构由于可以有效减少冲突缺失从而得到了最为广泛的使用。在对组相联缓存的每一次访问中，虽然访问一个缓存组中的所有缓存块，但最多只有一路缓存块是命中的。所以，如果我们能够事先消除不必要的缓存路访问，则由读操作产生的动态功耗必将被有效降低。

3.1 不必要的缓存路访问

对组相联缓存的每次访问产生的能耗 (E_cache) 可以由文献[9]给出的公式进行预估。如公式（1）和公式（2）所示。其中，E_decode 表示驱动地址总线和解析访存地址的能耗；E_memory 表示访问 tag 阵列和 data 阵列的能耗；E_I/O 表示当替换发生时驱动外部 I/O 管脚的能耗；N_tag 和 N_data 表示访问 tag 阵列和 data 阵列的个数；E_tag 和 E_data 表示访问单个 tag 阵列和 data 阵列的能耗。鉴于 E_memory 是整个能耗 E_cache 的主要来源，我们的方法主要是通过减少 N_tag 和 N_data 来降低 E_memory。

$$E_{cache} = E_{decode} + E_{memory} + E_{I/O} \quad (1)$$

$$E_{memory} = N_{tag} * E_{tag} + N_{data} * E_{data} \quad (2)$$

在传统的组相联缓存中，访存地址通常包含 tag 段，index 段和 offset 段。当读操作发生时，在访存地址生成以后，index 段被用来索引到相应的缓存组，进而读出此缓存组中的所有 tag 和数据，然后将读出的 tag 和 tag 段进行比较，看是否命中。如果命中，并且缓存块中的数据有效，offset 段用来选择需要的数据写回。否则，发出缺失消息，并等待数据的回填。

因为程序访存的空间局部性，组相联缓存同一个缓存组中，tag 比对导致不匹配发生的情况大多是由于 tag 低位不匹配造成的。所以，我们可以提前检查 tag 最低两位的匹配情况，进而在进一步的访问中过滤掉对最低两位不匹配缓存路的访问。同时，每次对某个缓存组的访问，并非任何时候每个要访问的缓存组含有的数据块都是有效的，比如，当某个缓存组中的块没有被填满，或者某数据块被无效消息无效掉时。在这种情况下，我们可以先通过检查要访问缓存组的 valid/invalid 标志位，然后在进一步的操作中，消除对含无效数据块的缓存路的访问。此外，在被指令和数据共享的缓存中，一个缓存组可能存有数据块，也可能存有指令块。而每次访问，要么访问数据，要么访问指令。所以，我们可以对每个数据块设置一个标志位（I/D bit），来指示里面存放的是指令还是数据，然后访问前检查此标志位，从而在进一步的操作中，消除对访问类型不匹配的数据块的访问。

以上过滤方法都对缓存的命中率没有产生影响，但是，由于我们相当于在缓存访问前放置了一个过滤器（Filter），所以从硬件实现的角度，需要多一拍的延迟来检测不符合进一步访问条件的缓存块。因此，建议

此过滤方法最好运用在延迟非敏感的缓存结构中，如 unified L2 cache 或共享 LLC（Last-Level-Cache）。

3.2 硬件结构

下面我们介绍以上三种不必要缓存路访问过滤机制的逻辑结构，用于检查无效数据块的 Invalid Filter，用于检查指令/数据访问类型不匹配数据块的 I/D Filter，以及用于检查 tag 最低 2 位不匹配数据块的 Tag-2 Filter，它们可以基于传统的组相联缓存而实现。

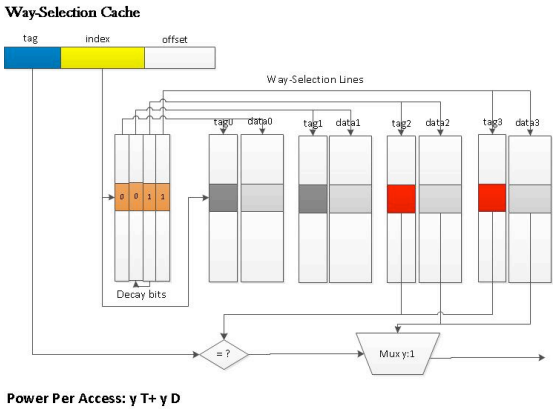

Fig.3 Way-selection cache[16].

图 3. Way-selection cache[16].

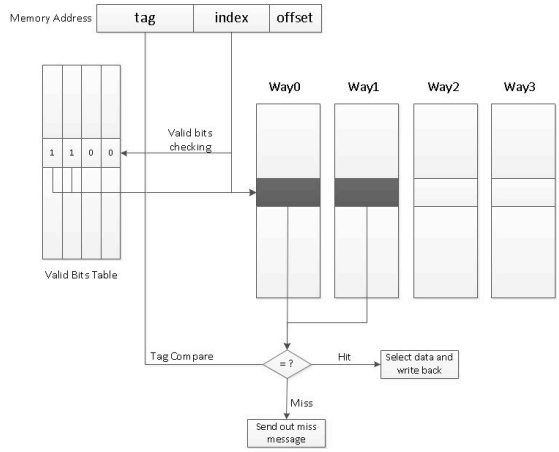

Fig.4 The architecture of Invalid Filter

图 4. Invalid Filter 的逻辑结构.

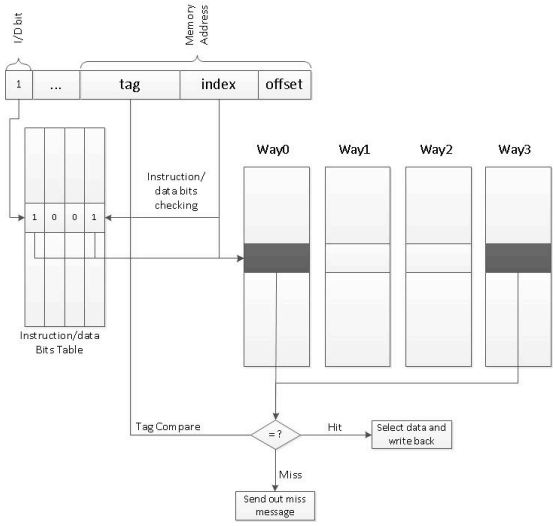

Fig.5 The architecture of I/D Filter

图 5. I/D Filter 的逻辑结构.

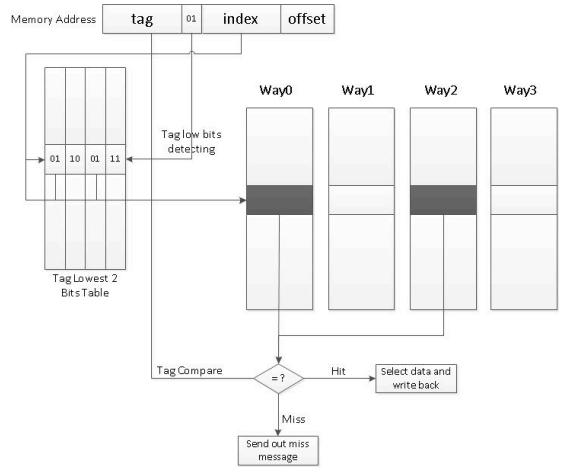

Fig.6 The architecture of Tag-2 Filter

图 6. Tag-2 Filter 的逻辑结构.

图 4 以 4 路组相联缓存为例，对 Invalid Filter 的结构进行了描述。由于传统的组相联缓存中，已存在 Valid/Invalid 标志位，图中为 Valid Bits Table。因此，我们只需将标志位的检查逻辑从读出数据和 tag 之后，移到读出数据和 tag 之前，并将此逻辑的输出与各个缓存路使能信号的生成逻辑合并即可。无效标志位所在的缓存路将不被使能。

图 5 对 I/D Filter 的结构进行了描述，也以 4 路组相联缓存为例。为了区分指令块和数据块，我们为每个缓存块添加了一位标志位，置为 1 表示数据块，为 0 表示指令块。如图 5 所示，还需要在访问缓存的请求消

息中设置一位来指示访问类型（数据或指令），从数据缓存发出置为 1，从指令缓存发出置为 0。数据块的标志位在发生回填时根据 miss 类型被置上。

Tag-2 Filter 的结构由图 6 给出，我们将所有 tag 的最低 2 位单独存储，在图 6 中我们称为 tag 最低两位表。在访问缓存组前，首先检查相应的 tag 最低两位表，并将结果输出与各个缓存路的使能信号产生逻辑合并。以缓存块大小为 32bytes 为例，加上 tag 位，每个缓存块的大小将超过 256bits，所以使每个缓存块多 2bits 的 tag 最低两位表的硬件开销不到百分之一，可忽略不计。

值得注意的是，以上三种方法是可以并行工作的，将以上方法混合后可以达到更好的过滤和节能效果，下一节我们将通过具体的实验进行验证。

4 实验结果

本节我们通过实验来评估混合过滤方法(Invalid+I/D+Tag-2 Filter)的效果。我们首先将实际程序的访存 Trace 在实现了混合过滤器的缓存上进行测试，以得到能够通过混合过滤器而消除的无效缓存路访问占整个访问的比率。然后，我们采用 CACTI6.0 [6]来评估采用过滤方法后可以被减少的动态功耗。同时，我们还将混合过滤器与只采用 Invalid+I/D Filter 的方法和只采用 Invalid+Tag-2 的方法取得的效果进行了对比。

4.1 可被消除的缓存路访问比率

实验中，2 个应用程序(spice, tex)和 6 个从 SPEC CPU2000 测试程序集中随机挑选的测试程序，其中 3 个来自 SPEC CPU2000 Integer (gcc, mcf 和 eon)， 另外 3 个来自 SPEC CPU2000 floating-point (mesa, ammp 和 art)，被用来验证通过各种过滤器可以消除的缓存路访问比率。其中，每个测试 benchmark 的访存 trace 都超过百万条。

针对三种不同的组合过滤器(Invalid+I/D Filter，Invalid+Tag-2 Filter，Invalid+I/D+Tag-2 Filter)，我们分别在 64K-4way 和 128K-8way 组相联缓存上进行了测试，这些缓存可以被指令和数据存储共享。实验结果分别如图 7 和图 8 所示。为了集中于我们所关注的问题，缓存块大小固定为 32bytes，替换策略采用随机替换。

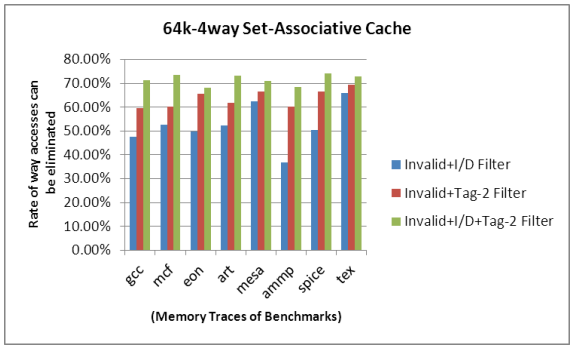

Fig.7 Eliminate rate on 64K-4way cache

图 7. 64K-4way 缓存上可被消除的路访问比率.

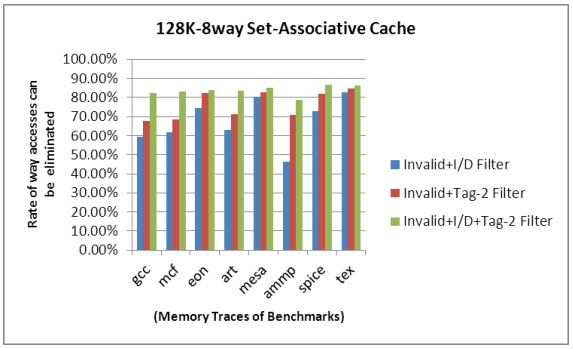

Fig.8 Eliminate rate on 128K-8way cache

图 8. 128K-8way 缓存上可被消除的路访问比率.

图 7 的结果显示，在 64k-4way 带有 Invalid+I/D Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc47.63%，mcf52.73%，eon49.60%，art52.41%，mesa62.50%，ammp36.65%，spice50.49%，tex65.88%；在 64k-4way 带有 Invalid+Tag-2 Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc59.65%，mcf60.07%，eon65.53，art61.64%，mesa66.38%，ammp60.25%，spice66.67%，tex69.32%；在 64k-4way 带有 Invalid+I/D+Tag-2 Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc71.42%，mcf73.58%，eon68.24%，art73.19%，mesa71.08%，ammp68.50%，spice74.16%，tex72.97%。

以上结果还表明，与 Invalid+I/D Filter 相比，Invalid+I/D+Tag-2 Filter 可以获得平均 19.41%的效果提高；与 Invalid+Tag-2 Filter 相比，Invalid+I/D+Tag-2 Filter 可以获得平均 7.95%的效果提高。

图 8 的结果显示，在 128k-8way 带有 Invalid+I/D Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc59.41%，mcf61.95%，eon74.54%，art63.12%，mesa80.39%，ammp46.49%，spice72.81%， tex82.94%；在 128k-8way 带有 Invalid+Tag-2 Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc67.86%，mcf68.52%， eon82.52%，art71.29%，mesa82.80%，ammp71.09%， spice82.10%，tex84.66%；在 128k-8way 带有 Invalid+I/D+Tag-2 Filter 的缓存结构上，能够被消除的缓存路访问比率分别为：gcc82.46%，mcf83.23%，eon84.07%，art83.42%，mesa85.31%，ammp78.91%，spice86.78%，tex86.48%。

以上结果还表明，与 Invalid+I/D Filter 相比，Invalid+I/D+Tag-2 Filter 可以获得平均 16.13%的效果提高；与 Invalid+Tag-2 Filter 相比，Invalid+I/D+Tag-2 Filter 可以获得平均 7.48%的效果提高。

4.2 能耗节省评估及对比

由于大量不必要的缓存路访问可以通过我们提出的过滤方法而预先消除，由此产生的动态能耗也必将大大被降低。接下来，我们采用 CACTI6.0 来评估和比较通过不同过滤器所能节省的能耗。实验中，我们只考虑由于缓存的访问而产生的动态能耗。实验结果表明，采用混合过滤机制，Invalid+I/D+Tag-2 Filter 的效果比 Invalid+I/D Filter 和 Invalid+Tag-2 Filter 的效果都好，实验结果的比较如图 9 和图 10 所示。

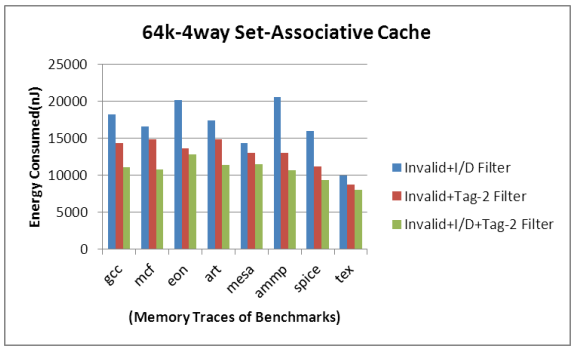

Fig.9 Comparison of power saving on 64K-4way cache

图 9. 64K-4way 缓存上的能耗节省比较.

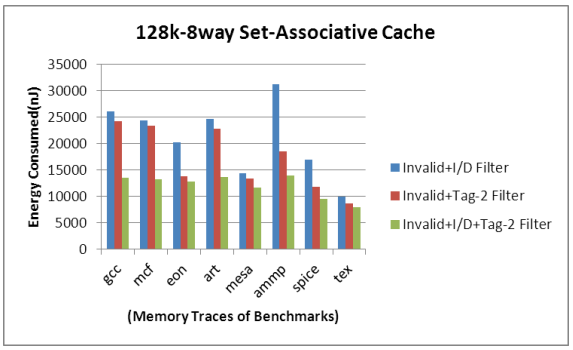

Fig.10 Comparison of power saving on 128K-8way cache

图 10. 128K-8way 缓存上的能耗节省比较.

图 9 的结果表明，在 64k-4way 缓存上，与采用 Invalid+I/D Filter 相比，Invalid+I/D+Tag-2 Filter 能得到的能耗降低效果提升为：gcc39.0%，mcf35.3%，eon36.6%，art34.7%，mesa20.1%，ammp47.8%，spice41.5%，tex19.6%，平均提高 34.3%；与采用 Invalid+Tag-2 Filter 相比，Invalid+I/D+Tag-2 Filter 能得到的能耗降低效果提升为：gcc22.5%，mcf27.7%，eon6.2%，art23.2%，mesa11.5%，ammp17.7%，spice16.2%，tex8.0%，平均提高 16.6%。

图 10 的结果表明，在 128k-8way 缓存上，与采用 Invalid+I/D Filter 相比，Invalid+I/D+Tag-2 Filter 能得到的能耗降低效果提升为：gcc48.3%，mcf45.7%，eon36.6%，art44.7%，mesa19.0%，ammp55.2%，spice44.1%，tex19.6%，平均提高 39.2%；与采用 Invalid+Tag-2 Filter 相比，Invalid+I/D+Tag-2 Filter 能得到的能耗降低效果提升为：gcc44.4%，mcf43.3%，eon7.0%，art40.0%，mesa12.4%，ammp24.7%，spice20.0%，tex8.0%，平均提高 25.0%。

5 结束语

一直以来，降低高速缓存的功耗，作为处理器领域的研究热点之一，无论在单核处理器还是在多核处理器中，都得到了国内外同行的广泛关注。本文，我们提出了在组相联缓存中采用过滤机制通过消除不必要缓存路访问来降低 CPU 动态功耗的方法。对于每次访问，利用 Invalid Filter 可以预先检查无效数据块，I/D Filter 可以提前查出指令/数据访问类型不匹配的数据块，Tag-2 Filter 可以提前查出 tag 最低 2 位不匹配数据块，被查出的数据块所在的缓存路的访问在随后得到消除。实验表明，以上方法可以有效降低缓存的能耗，并且当

三种方法混合使用时，效果最好。以上提出的方法也可以在多核/众核处理器的共享 LLC（Last-Level-Cache）中得以应用，这将是我们下一步的工作。

References:

[1] Le Sueur, Etienne and Gernot Heiser. “Dynamic voltage and frequency scaling: The laws of diminishing returns”. In Proceedings of the 2010 Workshop on Power Aware Computing and Systems, Vancouver, Canada, October 2010.

[2] Shekhar Borkar and Andrew A.Chien. “The future of microprocessors”. Communications of the ACM, Volume 54, Issue 5, pp.67-77. May 2011.

[3] Wendel D, Kalla R, Cargoni R, Clables J, Friedrich J, Frech R, Kahle J, Sinharoy B, Starke W, Taylor S, Weitzel S, Chu S.G, Islam S, and Zyuban V, “The implementation of POWER7TM: A highly parallel and scalable multi-core high-end server processor”, IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC'10). pp.102-103. 7-11 Feb. 2010.

[4] Kurd N.A, Bhamidipati S, Mozak C, Miller J.L, Wilson T.M, Nemani M, and Chowdhury M, “Westmere: A family of 32nm IA processors”, IEEE International Solid-State Circuits Conference Digest of Technical Papers(ISSCC'10). pp.96-97. 7-11 Feb. 2010.

[5] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. “Dark silicon and the end of multicore scaling”. In Proceedings of the 38th annual international symposium on Computer architecture (ISCA'11). New York, USA, pp.365-376. 2011.

[6] Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P.Jouppi. “Cacti 6.0: A tool to understand large caches”. In HP Research Report, 2007.

[7] W.A.Wulf and S.A.McKee. “Hitting the Memory Wall: Implications of the Obvious”. Computer Architecture News, Vol.23, No.1, Mar.1995, pp.20-24.

[8] C.Su and A.Despain. “Cache design tradeoffs for power and performance optimization: A case study”. In International Symposium on Low Power Electronics and Design, pp.63-68, 1997.

[9] Inoue K, Ishihara T, and Murakami K. “Way-predicting set-associative cache for high performance and low energy consumption”. In Proceedings of International Symposium on Low Power Electronics and Design (ISLPED'99). pp.273-275. 1999.

[10] D.H.Albonesi. “Selective cache ways: on-demand cache resource allocation”. In Proceedings of the 32nd Annual ACM/IEEE International Symposium on Microarchitecture. pp. 248-259, 1999.

[11] S.Kaxiras, Z.Hu, and M.Martonosi, “Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power”, in ISCA'01, Goteborg, Sweden, June 2001.

[12] C.Zhang, E.Vahlid and W.Naja. “A highly configurable cache architecture for embedded systems”. In The Prod. of the 30th Annual International Symposium on Computer Architecture (ISCA'03), pp.125-136, 2003.

[13] Chuanjun Zhang, Frank Vahid, Jun Yang and Walid Najjar.”A way-halting cache for low-energy high performance systems”. In International Symposium on Low Power Electronics and Design, pp.34-54. 2004.

[14] Ma Zhiqiang, Ji Zhenzhou, Hu Mingzeng, Ji Yi, Cao Jiannong and Xu Ming. “Energy Efficient United L2 Cache Design with Instruction/Data Filter Scheme”. Lecture Notes in Computer Science, Advanced Parallel Processing Technologies 2005.

[15] Chen Liming, Zou Xuecheng, Lei Jianming,and Liu Zhengli. “Dynamic Cache Resource Allocation for Energy Efficiency”. The Journal of China Universities of Posts and Telecommunications. Volume 16, Issue 1, February 2009, pp.121-126.

[16] Georgios Keramidas, Polychronis Xekalakis, and Stefanos Kaxiras. “Applying decay to reduce dynamic power in set-associative caches”. In Proceedings of the 2nd international conference on high performance embedded architectures and compilers (HiPEAC'07).

[17] Rajesh Kannan Megalingam, K.B Deepu, Iype P. Joseph, and Vandana Vikram. “Phased Set Associative Cache Design for Reduced Power Consumption”. In The Prod. of the 2nd IEEE International Conference on Computer Science and Information Technology(ICCSIT'09),pp.551-556,2009.

[18] Kamil Kedzierski, Francisco J.Cazorla, Roberto Gioiosa, Alper Buyuktosunoglu and Mateo Valero. “Power and performance aware reconfigurable cache for CMPs”. In Proceedings of the Second International Forum on Next-Generation Multicore/Manycore Technologies New York, NY, USA,2010.