Microprocessors and Microsystems 52 (2017) 221–235

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Performance linked dynamic cache tuning: A static energy reduction approach in tiled CMPs

Shounak Chakraborty*, Hemangee K. Kapoor

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Assam-781039, India

ARTICLE INFO

Article history:
Received 8 June 2016
Revised 3 April 2017
Accepted 14 June 2017
Available online 16 June 2017

Keywords:
Power optimization
Last level cache
Chip multiprocessor
Dynamic power
Leakage power
Performance
Cache power

ABSTRACT

Advancement in semiconductor technology increases power density in recent Chip Multi-Processors (CMPs) which significantly increases the leakage energy consumptions of on-chip Last Level Caches (LLCs). Performance linked dynamic tuning in LLC size is a promising option for reducing the cache leakage.

This paper reduces static power consumption by dynamically shutting down or turning on cache banks based upon system performance and cache bank usage statistics. Shutting down of a cache bank remaps its future requests to another active bank, called as target bank. The proposed method is evaluated on three different implementation policies, viz (1) The system can decide to shutdown or turn-on some cache banks periodically throughout the process execution. (2) The system allows to shutdown banks initially and once the bank restarting initiates, no more shutdown is permitted further. (3) This policy resizes cache like first policy with some predefined time slices, in which cache cannot be resized.

For a 4MB 4 way set associative L2 cache, experimental analysis shows 66% reduction in static energy with 29% gain in Energy Delay Product (EDP) for first strategy; for the second policy, static power is reduced by 59% with 27% savings in EDP. Finally, last policy saves 65% in static power and 30% in EDP with minimal performance penalty.

1. Introduction

Rapid progress in VLSI evolves more powerful designs for modern processors by incorporating more number of processor cores on a single chip: called Chip Multi-Processors (CMPs). To commensurate high data demand of these on-chip cores, larger on-chip caches are needed. These increased levels of core integration and transistor density increase the power consumption of the CMPs which leads to rise in the chip temperature. Due to the higher chip temperature, more expensive cooling and packaging techniques are required as higher chip temperature can damage the on-chip components permanently and also increases the on-chip leakage power. Therefore low power processor chip design is becoming essential with the rapid progress in chip design technology.

According to the survey given in [1], power consumption of on-chip memory subsystem shares a major portion of total power consumed by the chip. In modern CMPs, the on-chip caches are organised into multiple levels with the Last Level Cache (LLC) biggest in size. As LLC occupies large on-chip area, it consumes more leakage power that, at times exceeds dynamic power [2,3]. Table 1 gives a set of power consumption values of on-chip caches for a few microprocessors, which motivate one to attempt to reduce these numbers.

An effective way of reducing power consumption of the on-chip LLCs is by shrinking its size. Some recent works [4–6] have proposed power optimization approaches where on-chip LLC has been shrinked. Reduction in LLC size can degrade system performance if the application's cache demand is more or if some heavily used cache portion is powered off. Hence, tuning process of on-chip LLC size should consider the system performance and locality of reference as its constraints. As these constraints are only known during execution of the applications, hence, dynamic/runtime cache size tuning approaches will be more effective for reducing LLC power consumption.

Zang and Gordon-Ross have broadly classified the cache power reduction techniques into two categories [1]:

1. Power supply control, and
2. Resizing of the Cache Memory.

* Corresponding author.
E-mail addresses: c.shounak@iitg.ernet.in (S. Chakraborty), hemangee@iitg.ernet.in (H.K. Kapoor).

http://dx.doi.org/10.1016/j.micpro.2017.06.012
0141-9331/© 2017 Elsevier B.V. All rights reserved.

Table 1
Power consumed by on-chip caches [1,23,24].

Microprocessor	Power Consumed by on-chip Caches with respect to total power
ARM 920T	44%
Strong ARM SA-110	27%
21164 DEC Alpha	25–30%
Niagara	12%
Niagara2	21%
Alpha 21364	13%
Xeon (Tulsa)	13%

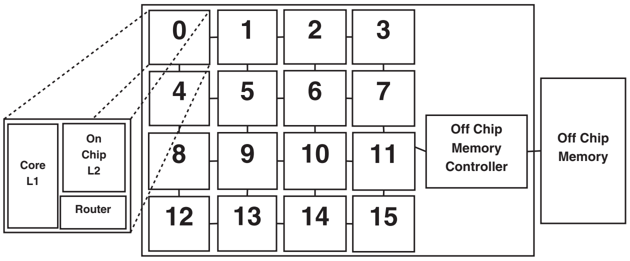

Fig. 1. Tiled CMP architecture.

The former one optimizes the power consumption by controlling the power supply at physical circuits whereas the latter one resizes the cache.

Power gating, a circuit level technique [7], controls the supply voltage of the least used cache ways, is heart of the drowsy cache [8] (we named this technique as Drowsy in this paper), which puts least utilized cache ways either in low power mode or turned off. For modern tiled CMPs, recent works [5,6] propose a set of utilization based cache resizing techniques, which power off least utilized cache portions and dynamically remaps subsequent future requests to other active parts. In this paper, a similar approach is taken for optimizing cache power consumption, by request remapping at L2-controller, unlike the prior works, where remapping is done at L1-controllers.

An earlier work [9] attempts to reduce cache-size by shutting down cache banks till an allowable degradation threshold in IPC. This technique is referred to as BSP. However, this policy cannot provide adequate cache space to the process in case it needs more cache space in future, during execution. The current work proposes a dynamic cache tuning technique which considers performance and locality of reference as its constraints for managing the cache size. The baseline architecture is elaborated in Fig. 1. In order to save leakage power, based on usage statistics, L2 cache banks are shutdown at runtime and its future accesses are remapped to other L2 cache banks. Additionally, this policy also takes care of the sudden increment in application's WSS (Working set Size) during execution by allowing dynamic restarting of the powered off cache banks. System performance is monitored periodically and accordingly L2 bank(s) will be restarted if performance degradation is more than a threshold value. During turning on process, all the remapped contents are brought back to this bank from its remapped location. The results are compared with BSP [9] and Drowsy [8], an existing policy. Specifically, the main contributions of this paper are as follows:

1. A performance linked dynamic cache tuning strategy resizes the L2 caches by turning off cache banks. However, if the application needs more cache space, L2 banks are turned on/restarted.
2. Frequent turning on and off of the L2 banks degrades performance. Hence, we experiment with different on-off patterns. Once the performance degradation reaches a threshold value, the system will not allow anymore shutdown of L2 bank(s). After this only turning on of L2 cache banks will be allowed.
3. The frequent resizing of L2 cache of first policy may degrade system performance. On the other hand, second policy does not allow the system to save power by turning-off the cache banks once the turn-off process is stopped, even when there is scope to do so in future. These two problems have been rectified in the third policy by putting some restrictions on cache resizing.

This paper is organized as follows. Section 2 reviews the related works. Section 3 presents the proposed energy saving policy and discusses criteria for cache bank shutdown, turn-on and remapping strategy. The implementation details and experimental setup will be discussed in Section 4. Results and analysis are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Recent survey [1] highlighted that on-chip caches are the most power consuming component in modern processor chips and so, optimizing cache power consumption is a good option for optimizing chip power. The survey presents state-of-the-art offline static and online dynamic cache tuning strategies from a power consumption perspective. The techniques, along with their applications are classified into several domains and sub-domains with their pros and cons.

In recent work [10], Wang et. al. proposed a stability controlled cache power management system which employs a two-tier feedback control architecture for limiting peak power consumption. The cache is dynamically resized with maintaining a balance between power and performance. In an earlier work [11], "folding" technique dynamically reduces cache size by powering off a number of cache sets. An address remapping policy has been attached with this to remap the data of these powered off sets to some other locations and each time the cache is resized, remap table updates itself. However, this technique increases conflict misses and due to which a performance degradation is noticed with savings in EDP.

Gated-Vdd technique [7] gates power supply to the unused SRAM cells by making changes at circuit level. This method optimizes cache power along with some novel cache tuning techniques at architectural level. Dynamic cache resizing can be done either by predicting WSS of the application before execution or by monitoring locality of reference of the application while running. In their recent work [5], Dani et. al. used Tagged Bloom Filter for estimating WSS of the application and allots cache space accordingly. To prevent data loss for the shutting down cache lines, a remap strategy has been clubbed with this. This remapping is done at L1 controller which increases extra burdens on L1 caches. But, in this paper, we dynamically tune the size of L2 cache and remapping is done at L2, so no extra burden for L1 controller.

By keeping a cache line either in active mode or sleep mode cache leakage power can also be reduced, where sleep mode consumes less power to retain the stored data in it [12]. A charging up is required before accessing the lines in sleep mode to bring them back into active mode. This charging up process takes a few clock cycles which have impact on performance but the paper claims it as negligible. In a recent work, Fitzgerald et. al. proposed a similar approach where a drowsy mode for cache sets is used [8]. The most recently used cache sets are kept in normal mode where the least used ones will be kept in drowsy mode. Drowsy mode is considered as slower mode as it requires a few extra clock cycles for waking up.

In another work, a workload independent cache power reduction strategy has been proposed for DNUCA caches [4,13], which exploits locality of reference to save power. The frequently used