Cache Capacity and its Effects on Power Consumption for Tiled Chip Multi-Processors

Shounak Chakraborty, Dipika Deb, Dhantu Buragohain, Hemangee K. Kapoor
Department of Computer Science and Engineering
IIT Guwahati,
Guwahati, India-781039
{c.shounak, d.dipika, dhantu.buragohain, hemangee}@iitg.ernet.in

Abstract—**Minimizing power consumption of Chip Multiprocessors has drawn attention of the researchers now-a-days. A single chip contains a number of processor cores and equally larger caches. According to recent research, it is seen that, on chip caches consume the maximum amount of total power consumed by the chip. Reducing on-chip cache size may be a solution for reducing on-chip power consumption, but it will degrade the performance. In this paper we present a study of reducing cache capacity and analyzing its effect on power and performance. We reduce the number of available cache banks and see its effect on reduction in dynamic and static energy. Experimental evaluation shows that for most of the benchmarks, we get significant reduction in static energy; which can result in controlling chip temperature. We use CACTI and full system simulator for our experiments.**

Keywords-Power optimisation; Cache; Chip multiprocessor; Dynamic power; Leakage power;

I. INTRODUCTION

In recent years, power consumption of Chip Multiprocessors(CMPs) has received attention of the researchers. With the rapid growth of IC technology, the number of on chip elements has been increased. Recently developed chips are having multiple processor cores with multilevel on chip caches to get better performance. The rapid increment of on chip components will increase the overall power consumption of the chip. The recent study[1] about the chip power consumption indicates that, the principal amount of chip power has been consumed by the on chip cache.

The power consumption of cache can be divided into two major parts-dynamic power and static power. Dynamic power is consumed when the cache is accessed and static power is generally referred as leakage power of the cache. To reduce cache power, focus should be given upon reduction of both the power components. The increased chip design complexity has increased power consumption of the chip, which will increase the chip running temperature. The rapid increment of chip temperature will increase the chip leakage power. Even, high working temperature of the chip can damage the internal circuits of the chip. To maintain a stable chip temperature, modern system suffers from high cooling cost. So, chip power minimization has opened a research avenue, under which cache power minimization for CMPs has received special attention now-a-days.

In modern CMPs, multilevel caches are organized in several ways. In our study, we will consider the chip having two levels of caches(i.e. L1 and L2). The L1 cache is considered as a private cache per core, whereas L2 will be shared by all the cores. According to the L2 organization, the L2 cache access pattern changes. When each core takes same amount of access time to access a particular data, the organization is called as Uniform Cache Access(UCA). When access time for a particular data differs from core to core, the cache organization is known as Non Uniform cache Access(NUCA). Generally, in case of on chip L2 NUCA cache, the L2 is divided into multiple banks. Cache lines and cache sets are organized inside the cache banks. The cache parameters need to be tuned to optimize cache power consumption according to [1] and references therein.

Cache power consumption is the major power component of chip power, out of which cache leakage is the principal one. So, for effective reduction in cache power consumption, selective bank shut down can be done. In this strategy, all cache banks will be allocated initially. Later, according to the change in Working Set Size(WSS) and cache bank usage, the number of required banks will be kept on, and remaining will be powered off. So, unnecessary power consumption will be reduced by this. In the latter time periods, if the program needs more caching of data, the banks will be turned on as needed. But frequent shut down and wake up of cache bank may be a system overhead with respect to power and performance. So, shut down and wake up decisions should be taken in a way which will not increase the system overhead.

In this work, we will study the cache power consumption while all the cache banks are on throughout the process execution. Later we will reduce the number of cache banks according to the requirement of the process. The bank shut down decision will be taken by running some benchmark programs in unchanged cache configuration and in the changed environment. With the improvement in IC technology, number of on-chip components has been increased. So, constant power supply to all of these components will increase chip temperature. So by putting some on chip components into power off state, effective chip

temperature will also be reduced which will reduce leakage power consumption. We will observe the power consumption for both cases by running the CACTI 6.0[2] tool with the proper configurations. The configuration requirement will be decided by running benchmark programs in Simics environment.

The paper is organized in the following ways. Section II reviews the related works. Section III presents our power optimization strategy and section IV discusses the experimental setup with results and analysis. Finally, section V will conclude the paper.

II. RELATED WORK

On chip cache power has become an important constraint for CMP Design. Some recent works has studied how cache power can be optimized by considering performance constraint. In [1], authors present a survey on cache tuning from power perspective. The survey presents a state-of-the- art offline static and online dynamic cache tuning techniques and summarizes the pros and cons of the techniques which open future research avenues.

Reduction of power consumption by Last Level Cache(LLC) can be a key factor for limiting peak power consumption of CMP chip. To reduce LLC power, dynamically some cache banks can be selected and put into low power mode. But this dynamic cache resizing can increase cache access latency which will increase the number of CPU stall cycles. To address these issues, Wang et. al. has proposed a novel cache management strategy to limit peak power consumption of LLC in CMPs[3]. Their work can be summarized in three steps-1) a novel L2 cache management strategy has been proposed which provides fair or differentiated cache sharing for threads running on a CMP, whose power consumption has been constrained. 2) a two-tier feedback control architecture has been designed to simultaneously limit peak cache power consumption to achieve the desired one. 3) an advanced feedback control theory has been used to incorporate stability in the system.

In another work, Brooks et. al. studied the validation and design strategy for power-performance simulators[4]. This study analyzed the accuracy of the simulators. In this work, authors break down accuracy into two sub-types: relative and absolute accuracy. They have also analyzed power-performance errors with their effects on the design choices used in the simulators.

To reduce the on chip cache power with considering performance constraint, cache configuration plays an important role. In [5], authors proposed an on chip cache management policy, named as Dynamic Cache Clustering(DCC), which dynamically forms a few clusters among the cache banks to provide a flexible and efficient cache organization for CMPs. A mapping and location strategy has been proposed to manage dynamically resizable cache configuration, especially on tiled CMPs.

In [6], Powell et. al. proposed a physical level power reduction technique to reduce leakage power of caches. The approach is known as Gated-Vdd, which gates the supply voltage and reduce leakage in unused SRAM cells. This technique together with the resizable cache architecture reduces energy-delay with less impact on performance. Apart from this physical level on chip cache power reduction, Aparna M. et. al. proposed an adaptive power optimization of on chip SNUCA cache on tiled CMP architectures[7]. In this work, authors proposed a tagged bloom filter, where (dynamic) L2 cache allocation will be done based upon the estimated WSS. In addition, a remap policy has been proposed here to prevent data loss in L2 cache during dynamic shut down of cache lines.

Adaptive Mode Control (AMC) technique has been proposed to reduce cache leakage power[8], where each individual cache line will be either in Active mode or Sleep mode. Sleep mode consumes comparatively low power among the two, by putting data store into low power mode. A new kind of cache miss, called sleep miss, will occur when data store is to be accessed during sleep mode. Activation from sleep mode needs few stall cycles without impacting any significant degradation in the performance. However, authors claimed a significant reduction in cache leakage with respect to prior works by implementing sleep strategy.

Drowsy cache is another way to cache power reduction. In [9], authors proposed a phase adaptive cache design method, through which both dynamic and static energy has been reduced. A small performance degradation has been noticed in this work. The whole cache has been partitioned into two parts, where one is faster and other is slower due to drowsy mode. Drowsy mode is a low power mode, which needs a few extra cycles to active and work normally. According to MRU policy, mostly used data will be kept in the fast accessed location, and remaining will be in the slower region, and they will be swapped as needed.

A workload independent cache energy reduction strategy has been proposed in [10]. In this work, authors proposed a power reduction technique for D-NUCA caches, which adapts the powered-on cache area to the needs of the running workload, but it does not rely on application-dependent parameters. Data of the farthest cache way will be brought at the possible nearest cache way of the core, which currently accesses the data. Turning of the cache line saves leakage power a lot. This strategy saves 49% of total cache energy consumption in single core environment and saves 44% in the CMP environment. In an analytical based work[11], A. Bardine studied the static and dynamic energy consumption of NUCA caches. They presented a comparison based energy consumption study on the conventional UCA caches with the SNUCA and DNUCA caches. The results show that, NUCA caches are the most energy saving architectures and give better performance with respect to conventional UCA caches. According to the results obtained in this work, it is proven that, DNUCA caches have highest number of bank accesses and also have highest amount of data migration in it. So it consumes more dynamic energy than other configuration. But, still result shows, static energy dominates the dynamic energy. This promotes a strong motivation to the future researchers for concentrating upon the leakage energy consumption.

III. STUDY METHODOLOGY

We have used a Tiled CMP architecture as a baseline of our work. The CMP consists of 16 tiles, with each tile has a core, a private L1 cache, and a shared L2 cache. The design of 16-tiled based CMP is shown in figure 1. The number written inside the rectangle represents the corresponding Tile-id. The energy model used in our experiment is similar with the model used in [11]. To convert the output in terms of power, we have modeified the Energy dissipation formula of [11] as needed. The execution time is taken in terms of seconds. Total power consumption will be computed as:

$$P_{total} = P_{dynamic} + P_{static} + P_{off-cache} \quad (1)$$

And total energy consumption will be computed as:

$$E_{total} = E_{dynamic} + E_{static} + E_{off-cache} \quad (2)$$

where $P_{dynamic}$ ($E_{dynamic}$) indicates dynamic power(energy) consumed by banks of the cache including network elements, if they present. The dynamic power and energy can be broken down as follows-

$$P_{dynamic} = (\text{no. of bank accesses} \times P_{\text{bank access}} + \text{no. of flit transmissions} \times P_{\text{flit transmission}} + \text{no. of flit traversals} \times P_{\text{flit traversal}}) \quad (3)$$

The dynamic energy for the same will be calculated as-

$$E_{dynamic} = P_{dynamic} \times \text{execution time}$$

And the static power dissipation by the cache bank and network switches will be calculated as follows-

$$P_{static} = (\text{no. of banks} \times P_{\text{bank static}} + \text{no. of switches} \times P_{\text{switch static}}) \quad (4)$$

The static energy for the same will be calculated as-

$$E_{static} = P_{static} \times \text{execution time}$$

All the terms of the equations 3 and 4 have been described in [11] with a simple change-the term "Energy(E)" has been replaced by "Power(P)".

A. CACTI

According to our study, we are going to analyze the cache power consumption for different cache configurations. The equations mentioned above will be used for the total energy dissipation of the cache. In this study, we will focus on the energy dissipation by L2 cache. We use CACTI 6.5 to derive the energy parameters for the L2 cache memory banks. This latest version of CACTI combines the enhancements made in CACTI 5.0 and CACTI 6.0.

We modify the configuration file of CACTI 6.5 as required. The modification details are given in table I. We will have a fixed L2 bank size, and for different cache sizes we will increase or decrease the number of banks in L2.

B. Evaluation Topologies

For our evaluation purpose, we have taken a 16-core Tiled CMP system like figure 1, initially. The tiles are connected to each other through a 2D mesh network, called network-on-chip. After running some benchmarks(from PARSEC[12]) in this unchanged configuration, we collect all the data required for our analysis. Later, we have reduced the number of L2 cache banks to get the changes in power values. We have assumed that, in our changed configurations, we will simply switch-off the L2 banks in the tiles. For concrete analysis, we will take different configurations and for each one, we will extract the power values.

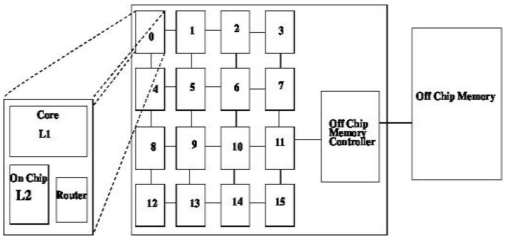


Figure 1. Tiled CMP architecture

For reduction in cache power, we have reduced the number of cache banks. But random shut down of cache banks may increase the data miss rate which will degrade the overall system performance. To overcome this performance degradation, we redirect the addresses of shut down banks to the powered-on banks. The address redirection strategy is shown in the figure 2. Here, in this figure, we have powered-off four banks(bank numbers, from 12 to 15), and the corresponding addresses are mapped into the banks, numbered 8 to 11. So, in the next, when a new request will arrive for the bank number 13, the redirection mechanism will send the request to new location at bank number 9.

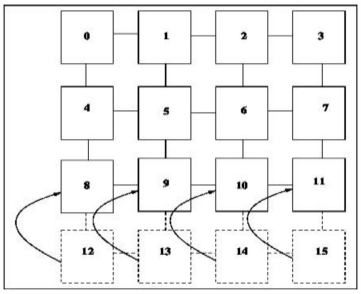


Figure 2. Address Redirection strategy when four banks are powered off, and their addresses will be redirected to the powered-on banks shown in the figure by arrow. The powered off banks are shown by dotted lines, along with the links they use.

For the evaluation purpose, we compare the following configurations as given below-

1) 16 banks vs. 12 banks
2) 16 banks vs. 8 banks

Bank shut-down reduces the cache power, but on the other hand, it will reduce the cache size. The reduction of cache size will increase the number of capacity and conflict misses which will degrade the system performance. To address these issues,

our analysis will give a clear idea about proper cache- size. Hence we use total cache initially, and later gradually we reduce the cache size by shutting down cache banks. By running certain benchmark programs, we will give the detailed analysis on system behaviors in terms of power consumption and performance for different cache configurations.

TABLE I. CACTI CONFIGURATIONS

Cache Parameteres	Values
Cache Level	L2
Size of a L2 Bank	256 KB
Block Size	64 Bytes
Technology used	32nm
Associativity	8
Cache Model	NUCA
Operating Temperature	340 K
Actual Cache Size	4 MB

TABLE II. SYSTEM PARAMETERS

Components	Parameters
No. of Tiles	16
Processor	UltraSPARCIII+
L1 I/D Cache	64KB, 4-way
L2 Cache bank	256KB, 4-way/8-way
Memory bank	1GB, 4KB/page
CMP-VR/RCMP-VR: reserveways per set(R)	50%

TABLE III. NETWORK PARAMETERS

Network Configurations	Parameters
Flit Size	16 bytes
Buffer Size	4
Pipeline Stage	5-stage
VCs per Virtual Network	4
Number of Virtual Networks	5

IV. EXPERIMETAL EVALUATION

A. Experimental setup

For evaluation purpose, simulations are performed by running benchmarks on a multi-core simulator GEMS[13] with the help of SIMICS[14], a full-system functional simulator. GEMS has a timing simulator of multiprocessor memory system, named Ruby. The detailed configuration about the processor, cache memory and main memory used for the experiment is given in Table II. Following multi-threaded benchmark suites from PARSEC[12] have been used for simulations- vips(vips), blacksholes(black), fluidanimate(fluid) and bodytrack(body). The tiled CMP used in the simulations, has 16 tiles in it as shown in figure 1. The L2 cache size used here is of 4MB. The L2 is divided into 16 banks where each bank is located in each tile. So, the size of each L2 bank will be 4MB/16=256KB. Remaining detailed configurations required for our experiment are given in table II. Apart from the system parameters, we need a set of network parameters, which are given in table III.

For the power calculation, we use CACTI 6.5 in the next. The dynamic energy and static energy for the L2 cache configurations will be computed as we described in the previous section.

B. Result

According to the above experimental setup we have run our simulation in Simics and CACTI 6.5. The obtained results are normalized according to the base line design. These normalized results are shown in the figures 3-5. Figure 3(a) and (b) show that three benchmark programs give better result in case of dynamic as well as static energy consumption, for 12 L2 cache banks, than our base line(i.e. 16 L2 cache banks). Among the four benchmark programs, black does not show any improvement. Reduction in the number of cache banks will decrease the on-chip active area, which will reduce the cache leakage power consumption. This is evident from the savings in static power in range of 16-34% with average of 24% for benchmarks that benefit from our proposal. The benchmark black gives the reverse results as per our prediction. For this two case, the number of on-chip cache accesses has increased with the reduction in cache size. This increased on-chip cache accesses will increase the chip temperature which will increase the static power consumption of the cache. Using this result we conclude that the working set size needed for this benchmark is much larger and hence we should not reduce the cache capacity.

In the next, we further decrease the cache size to 8 banks. Figure 4(b) shows the total energy consumption. Apart from black, all remaining benchmarks do show significant energy savings. In case of black, huge amount of data movement is the basic reason for increment in power consumption.

TABLE IV. PERCENTAGE REDUCTION IN POWER INCREASE IN CPI.

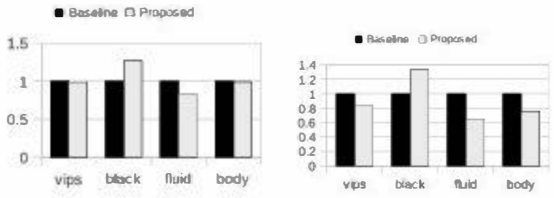
Benchmark	Comparison between 16 L2 banks and 12 L2 banks			
	Dynamic Energy	*Static Energy*	*Total Energy*	*CPI*
vips	1.5	16.48	1.5	7.13
black	−26.78	−32.88	−26.78	−1.43
fluid	16	34.91	16.72	0.29
body	1.27	24.51	1.27	5.63
Average improvement	3.17	24.16	3.17	2.23

Effect of energy savings on performance:

With the reduction in cache size, number of conflict and capacity misses may increase. This increased misses will increase the memory stall cycle, which degrades the system performance. According to figure 5, we can conclude that, reducing number of L2 cache banks from 16 to 12 will result in slight reduction in performance. The figure shows the graph of cycles-per-instruction for each benchmark. As is evident, the benchmarks that benefited from energy savings had to compromise on the performance. However, the degradation is not large. On average (cf TableIV) 2.23% increase occurs in CPI, except for program: black.

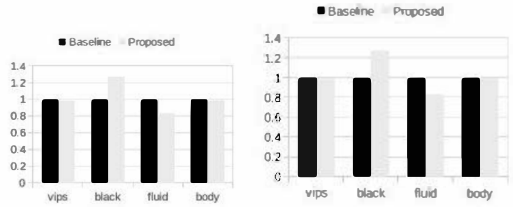
At the end, our conclusion on power consumption and system performance are given in table IV. According to the results available, we have made an average which can give us a concrete conclusion. The average values show that, reducing

L2 cache banks to 12 is beneficial than 8 for all the power values and performance. As excessive reduction in cache size will increase the capacity and conflict misses, which results a huge data movement to and from the cache. This huge data movement will increase the dynamic cache power along with the memory stall cycles which affects the system performance. The huge data movement in on-chip L2 cache will increase the chip temperature which will increase the on-chip L2 cache static power consumption. Using 12 L2 cache banks gives significant improvement in case of power consumption than our baseline, with negligible degradation in performance.



(a) (b)

Figure 3(a). Dynamic Energy savings by reducing cache banks from 16 to 12. (b). Static Energy savings by reducing cache banks from 16 to 12.



(a) (b)

Figure 4(a). Total Energy savings by reducing cache banks from 16 to 12. (b). Total Energy savings by reducing cache banks from 16 to 8.

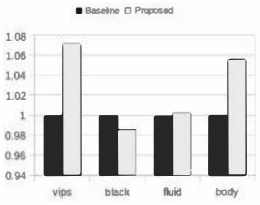


Figure 5. Performance degradation: by reducing cache banks from 16 to 12.

V. CONCLUSION

Due to increment in on-chip cache size, the cache leakage power consumption becomes the major on-chip power component. In this work, we have studied the effect of cache size reduction on cache power consumption and performance. Our base case has 16 banks on-chip L2 cache, later it will be reduced to 12 and 8 respectively. For certain applications reduction in cache size is not beneficial. However, for certain larger set the proposed approach gave significant savings. In particular we got an average reduction of 3.17% and 24.16% in dynamic and static energy respectively. This significant reduction in static energy will also help in maintaining lower chip temperatures, thus helping further reduction in leakage power. However, we also noted that too much reduction in cache size will degrade the performance by increasing cache misses. Hence, in the future, we plan to explore a dynamic L2 cache-size tuning strategy which will tune the cache size according to the cache demand of the running process on a tiled CMP architecture.

ACKNOWLEDGMENT

We wish to acknowledge Department of Electronics & Information Technology(DeitY), Ministry of Communications & IT, Government of India, for the financial assistance provided for this work.

REFERENCES

[1] W. Zang and A. Gordon-Ross, "A survey on cache tuning from a power/energy perspective," *ACM Computing Surveys,* vol. 45, no. 3, June 2013.

[2] N. Muralimanohar, R. BalaSubramonian, and N. P. Jouppi, "Cacti 6.0: A tool to understand large caches," March 2008.

[3] X. Wang, K. Ma, and Y. Wang, "Cache latency control for application fairness or differentiation in power-constrained chip multiprocessors," *IEEE Transactions on Computers*, vol. 61, no. 10, pp. 1371–1385, October 2012.

[4] D. Brooks, P. Bose, and M. Martonosi, "Power-performance simulation: Design and validation strategies," *ACM SIGMETRICS*, vol. 31, no. 4, pp. 13–18, March 2004.

[5] M. Hammoud, S. Cho, and R. Melhem, "Dynamic cache clustering for chip multiprocessors," *ACM ICS,* pp. 56–67, June 2009.

[6] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-vdd: A circuit technique to reduce leakage in deep-submicron cache memories," *ACM ISLPED*, pp. 90–95, 2000.

[7] A. M. Dani, B. Amrutur, and Y. N. Srikant, "Adaptive power optimization of on-chip snuca cache on tiled chip multicore architecture using remap policy," *Second Workshop on Architecture and Multi-Core Applications,* pp. 12–17, 2011.

[8] H. Zhou, M. C. Toburen, E. Rotenberg, and T. M. Conte, "Adaptive mode control: A static-power- efficient cache design," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 347-372, August 2003.

[9] B. Fitzgerald, S. Lopez, and J. Sahuquillo, "Drowsy cache partitioning for reduced static and dynamic energy in the cache hierarchy," *IGCC*, pp. 1–6, 2013.

[10] P. Foglia and M. Comparetti, "A workload independent energy reduction strategy for d-nuca caches," *The Journal of Supercomputing*, October 2013.

[11] A. Bardine, P. Foglia, G. Gabrielli, and C. A. Prete, "Analysis of static and dynamic energy consumption in nuca caches: Initial results," *ACM MEDEA07*, pp. 105–112, September 2007.

[12] "Parsec benchmark programs-," http://parsec.cs.princeton.edu/.

[13] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood, "Multifacets

general execution-driven multiprocessor simulator (gems) toolset," *Computer Architecture News (CAN)*, pp. 1–8, September 2005.

[14] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," *IEEE Transactions on Computers*, pp. 50–58, February 2002.