IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. III (Sep - Oct 2018), PP 49-53
www.iosrjournals.org

Techniques for Cache Improvement

Dr. Mohammad Mahmood Otoom, Dr. Khalid Nazim Abdul Sattar
Department of Computer Science and Information, College of Science, Majmaah University, Saudi Arabia
Corresponding Author: Dr. Mohammad Mahmood Otoom

Abstract: *Personal data caches have been rendered ineffective in the process of reducing the median memory lag in multiprocessors, as is the case in single-processors, due to the distribution of data amongst processors, and further related to a cache issue. Therefore, a multitude of processors have been implemented with the purpose of sustaining cache improvement in the presence of large-scale shared data processes; thus it is difficult to perform a comparison of implementation implications and performance. This paper is penned with the purpose of aiding future computer architects comprehend the opportunity costs involved as well as contemporary cache improvement processes, and other design-related problems. Current design problems include: 1) the improvement strategy, which provides the means for cache data access to be detected. 2) An enforcement strategy, which dictates that updates and validations are run to ensure that outdated cache entries are not referenced by a processor 3) Block sharing information precision and how it can be altered to lessen the implementation costs and to increase the performance of the memory system.*

Date of Submission: 12-10-2018 Date of acceptance: 268-10-2018

I. Introduction

The modern movement in the sector of computing is towards a process that favors green designs and an efficient use of energy. Within the realm of IT solutions, this has become of increasing importance [1]. Moreover, the goal of recent chip design has been geared towards creating hardware that is capable of the highest performance whilst still maintaining maximum energy efficiency. This has become an important goal in the creation of a range of processors, such as mobile devices, supercomputers and desktops. While these goals may seem to be a binary option, as energy and efficiency do not always contribute to each other, there have been a variety of research that introduce varying architectural techniques concerning the various components of the processor: DRAM, processor core, caches, and more [2]. The cache is, for a multitude of reasons, a focal point of the modern design, as with each CMOS creation, energy is leaked at a dramatic rate. Thus, in accordance with the findings of International Technology Roadmap for Semiconductors, as technology continues to scale, the energy consumption has the potential to create an industry-wide crisis, one that may threaten the survival of CMOS and the technology surrounding it.

As the amount of processor cores available on a singular chip has increased over the years, the projection is that the amount of processor cores will continue to rise. Moreover, the speed of processors have advanced at a dramatic pace, translating into an industry that uses larger caches as a means of bridging the gap between processing speeds and memory. This can be seen with a simple look into the current technology: modern household processors (desktops) have around 8MB of caches, whereas those contained in servers have around 24 – 32MB caches [7]. Caches consume a substantial portion of the total energy. Chip caches consume approximately 16% of the available energy; Niagra processors see caches consume 24% of the total available power; StrongArm processors also suffer a 30% power uptake due to caches alone. As can be seen, caches are responsible for a large percentage of power usage and this must be met with the development of designs that are geared towards achieving increased energy efficiency in caches. The goal of this paper is to analyze a few of these techniques.

II. Background

The energy usage of CMOS is largely categorized into two segments: dynamic power and leakage power [3]. Following, the presentation for the modeling equations will be shown to provide greater clarity, both for dynamic power and leakage power in their non-complex form; this aids in the obtaining of valuable insight into how energy efficiency may be achieved and how exactly architectural techniques work. Dynamic power is denoted by "P(dynamic)" and t is dissipated in the presence of a transistors switch to alter the voltage in an isolated node, Leakage power is denoted by "P(leakage)" and it is dissipated when there is a leakage of current that are present even when the device is turned off or otherwise inactive. This is represented mathematically:
"P(dynamic) = α×Ceff ×V 2DD ×F
P(leakage) = VDD ×I(leak) ×N ×k(design)"

From the first equation, one can clearly notice, within the realm of CMOS tech and generation, dynamic power usage may be reduced significantly if the voltage were to be adjusted alongside the frequency of uptime, or simply the lessening of activity factors (this may be achieved through the reduction of the number of cache accesses or the amount of bits available per cache access, or something similar). From equation 2, one can see that CMOS tech and generation provides the opportunity for energy saving. This occurs through the redesign of circuits to one that uses lower power cells, thus reducing the overall number of resistors [5]. Alternatively, some of the parts of caches may be put into leakage mode. With the foundation of these core principles, the industry has witnessed a variety of proposed architectural designs and techniques, which shall be discussed in the following segments.

First-level caches and last-level caches both play an important role in effective cache operation. The former controls access latency with the purpose of minimizing it; the latter pertains to cache miss-rate as it minimizes both miss-rates and off-chip accesses. FLCS, having smaller associativity and being necessary only to employ parallel tag of arrays and data, are smaller in size – about 32KB or 16KB. LLCs are required higher associativity and they perform multiple phased lookups of data and tag arrays, thus they are larger in size – usually 4MB or 2MB. Logically, power efficiency would concern itself more with LLCs instead of FLCs. Due to the size and nature of the two, LLCs provides the larger scope for energy saving techniques; this is also because FLCs are more devoted to dynamic energy and LLCs spend most of their energy in the form of leakage energy [4].

III. Methodology

This paper is penned with the task of discussing techniques that contribute to solving the efficiency of caches. Thus, the following sections will introduce a brief background on the topic of CMOS power usage. Thereafter, there will be a shift to the discussion of cache design guidelines, which may serve to reduce the overall energy consumed and increase performance. The landscape of the methods are the first point of discussion prior to going into further detail. Moreover, the foundations and similarities and key differences are also analyzed [16]. Due to the leakage of energy being a crucial factor in the industry, this paper focuses on ways to stop power leakage. Practical applications will also be demonstrated as to allow for the implementation of the current literature and to provide for real-world examples. To this end, commercial chip designs are also mentioned, those that show energy saving techniques in the design of the cache [8].

It is important to state that it is not possible to provide a comprehensive analysis of the technique in a review of this size; thus the direction of the paper has been carefully managed. Performance-related techniques may aid in the process of solving the energy crisis, but this paper shall only deal with those techniques that are aimed at providing solutions to energy leakage – moreover those that have been proven to increase energy efficiency [6]. The increase of energy efficiency may also be achieved through the use of circuit-level designs, but for the sake of the interests of this paper only architecture-level designs will be the topic of discussion. These allow runtime cache energy efficiency. Finally, as there have been a number of different techniques which have been subject to evaluation by varying simulation infrastructure, qualitative improvement outcome shave been discarded. Instead, there is a focus on architectural techniques that are capable of beneficial insight [7].

Dynamic Energy Saving Techniques

According to Kaxiras, Hu, and Martonosi (2011) recent times have seen a multitude of designs that are geared toward saving dynamic energy [13]. However, in order to provide a sufficient analysis, one must first be attuned to the foundational similarities and differences. As such, they must be properly categorized. Certain designs involve a reduction in the overall volume of accesses to a designated level of the cache hierarchy, accomplished with the development of extra memory structures. These are most likely used as a means of predicting cache access results, or as a mechanism for the storage of data, or as preempt of access results [9].

Other techniques are aimed at reducing the amount of ways through which a cache can be accessed in each cache access. This is achieved with the use of software, hardware design, or compiler information. Some of these techniques allow the access of frequently used data with a reduction in the usage of energy [10]. Thus, the average energy consumption upon the access of these types of data is greatly reduced. Moreover, there are other techniques which sacrifice access time for the sake of gaining efficiency of energy. In these scenarios, the tasks performed are done in a sequential manner instead of all at once. Thus, if a cache has already undergone a hit/miss decision, then there is room for additional tasks to be avoided, and in this way dynamic energy is saved. Likewise, energy efficiency may also be obtained through the performing of matching tag-bits in a plurality of steps, or the reduction of the amount of tag-bits that are active or needed for a comparison. Designs for the reduction of data transferred per cache access have also been proposed [11]. These, whilst they may all be somewhat effective in the reduction of dynamic energy in multiprocessors, are divided amongst those techniques that are specifically created with multiprocessors in mind and those which are not.

Powell et al. provides the framework for a technique that predicts the cache way, on that is the most likely to have the data. Thus, only a single way is allowed access on a cache access. Therefore, a correct prediction sees the cache operate as if it was a direct mapped cache, reducing the dynamic energy. In the event of misprediction, all cache ways are forced into access. This may lead to an increase in energy due to higher access time. Furthermore, this technique also faces the potential for non-uniform hit latency, bot with correct and incorrect predictions, and a scenario that has been corrected with way-section designs. Zhu and Zhang also add to the technical literature with their proposal to combine way-prediction with phased access mechanisms [14]. In essence, this method dictates that a way-prediction system has the capability to handle a cache hit. The phase mode will then handle any cache miss. Simple predictors are implemented to find the results of any cache access. Upon a prediction, the way-prediction system is used to which way is the most efficient; only that way is used. Upon the prediction of a miss, the phase-access system provides access to all tags of the cache-set prior to the correct way being accessed [12].

Leakage Energy Saving Techniques

Earlier, this paper mentioned the findings of energy saving in the presence of leakage systems. This is accomplished by turning off a portion of the cache in order to minimize the overall energy usage of the cache [15]. Therefore, due to the nature of the turned off blocks, the leakage energy designs must be split into two main camps: state-destroying methods and state-preserving methods. State-preserving methods have the ability to switch off a block whilst maintaining its state; this translates into a reactivated block that does not require memory to be fetched from elsewhere in the system. In comparison, sate-destroying methods achieve power saving by not preserving the state of a block. However, they do save more energy when the system is in low power or inactive states. Certain measures have been employed in an attempt to marry both these methods. However, Li et al. have compared the efficiency of both state-preserving and state-destroying techniques, and they have found the latter to be less efficient. This is due to the high costs associated with fetching memory. State-preserving techniques also showcase higher performance. Moreover, it must be considered that each of these techniques work at varying granularities [17].

VI. Results

The contemporary processor is designed to produce increasing performance, a movement which consequents in the increased size of on-chip caches. Moreover, as the CMOS production technology increases, the industry has witnessed a dramatic incline in the leakage of energy and the rise of power consumption. Thus, the results of this study are vital as the industry is hungry for a reliable way through which cache power management can be controlled. This is also a focal point of research due to the future of the technology, as future processing speeds and performance will require caches which are far more efficient than the present version. This paper has seen the presentation of a multitude of techniques which showcase the feasibility of achieving better energy efficiency; each method is proposed with better performance and efficiency in mind. Thus, different architecture has been presented. Recent designs have fared better than those which cater to a less technologically advanced ecosystem; such has been the demands and advancements of the last 5 years that only those studies which showcase an understanding of the present scenario are considered in this section. Therefore, recent designs that are purposed with adding to the energy efficiency of caches have been considered. Moreover, due to the fact that these are only useful in practical application, several real-world designs have been analyzed – these pertain to commercial processing chips that are currently available on the market. Thus, the results are aimed at allowing engineers better understand how the problem may be solved and where the current literature is headed. Moreover, it should form a platform which other architects with the same vision can build on, allowing for the construction of new and innovative solutions to a real problem: the inefficiency of caches.

As a result, the reconfiguration of these various levels of granularities comes with a variance in pros and cons. Techniques such as selective-way methods, cache-coloring and selective-sets do not require the alteration of the coding or cache configuration; thus implementation is made simpler and easier [18]. There is a downside in that selective-way approaches may cause harm to the cache itself and its associativity. Moreover, such an approach provides a limited granularity, and it requires caches with high associativity; these have increasing potential for access times and energy consumption. Despite this, it has been well-documented that a reduction in the size of the cache will translate into an increased miss rate. In this vein, cache coloring sees better granularity and efficiency of configuration – more so than a selective-sets approach. Moreover, it has a higher overhead implementation. However, a hybrid of the two does exist in an attempt to obtain higher granularity than either in a singular form. Moreover, such an approach allows for a combination of benefits, but they suffer from a higher overhead and difficult implementation [19]. The studies also show that temperature is a variable in the process of analyzing the effectiveness and efficiency of cache function, as a rise in the temperature increases the amount of leakage energy, which may increase the temperature even more. Thus, temperature has to be controlled, and a variety of techniques have been proposed. These may be referred to as

"thermal-aware" or "thermal-sensitive" methods. Moreover, such techniques may be implemented with both single processors or multi-core processors, but these do have separately designed systems that allow for further enhancement through specialization of design.

Lastly, many techniques that are aimed at solving efficiency have been combined with a number of other methods with the hope of achieving certain synergies that contribute to overall efficiency. These include the likes of DVFS, prefetching and data compression. These provide an entire new set of permutations for the solution to a highly complex problem. However, each has its own drawbacks which need to be properly managed in order to solve the performance and efficiency of cache function. To this end, leakage saving techniques are still underutilized.

VI. Conclusion

CMOS fabrication has been a field of study in recent years, a fact that has yielded a growing number of innovations in this specific technological industry. Moreover, the last few years have also been the foundation for the mass use of multicore processors in conjunction with larger caches that are located on-chip. This is done to increase the performance. However, an opportunity cost has arisen in the form of energy efficiency. The total power usage of modern processors is fast reaching a plateau, as can be logically observed by limitations in cooling technology and the amount of power available. Therefore, if higher performance is to be achieved, energy efficiency must be carefully managed. Technology scaling is only viable in the presence of an effective power solution; it is now a necessity. Therefore, this paper has been the revision of a few techniques that are proposed at solving the current dilemma, in particular the management of leakage power and dynamic power in caches. There has also been room for the discussion of the commercial field, as the market remains hungry for increased performance, and companies seek to do so by finding ways to save cache power runtime. This paper, therefore, is a respectable platform that can aid researchers, developers and engineers during the process of discovering the latest trends in architectural techniques that are purposed with solving the energy crisis. This may also be a mechanism through which future trends in CMOS technology and processor deign may be enhanced and the challenges of that process met.

References

[1]. S. Murugesan, "Harnessing green IT: Principles and practices," IT professional, vol. 10, no. 1, pp. 24–33, 2008.

[2]. S. Borkar, "Design challenges of technology scaling," Micro, IEEE, vol. 19, no. 4, pp. 23 –29, jul. 1999.

[3]. G. Gammie, A. Wang, H. Mair, R. Lagerquist, M. Chau, P. Royannez, S. Gururajarao, and U. Ko, "Smartreflex power and performance management technologies for 90 nm, 65 nm, and 45 nm mobile application processors," Proceedings of the IEEE, vol. 98, no. 2, pp. 144–159, 2010.

[4]. "International technology roadmap for semiconductors (ITRS)," http:// www.itrs.net/Links/2011ITRS/2011Chapters/2011ExecSum.pdf, 2011.

[5]. S. Borkar, "Thousand core chips: a technology perspective," in 44th annual Design Automation Conference. ACM, 2007, pp. 746–749.

[6]. "First the Tick, Now the Tock: Next Generation Intel Microarchitecture (Nehalem)," Intel Whitepaper, Tech. Rep., 2008.

[7]. B. Stackhouse et al., "A 65 nm 2-billion transistor quad-core Itanium processor," IEEE Journal of Solid-State Circuits, vol. 44, no. 1, pp. 18–31, 2009.

[8]. R. Riedlinger, R. Bhatia, L. Biro, B. Bowhill, E. Fetzer, P. Gronowski, and T. Grutkowski, "A 32nm 3.1 billion transistor 12-wide-issue Itanium®processorformission-criticalservers,"inIEEEInternational Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011, pp. 84–86.

[9]. A. Vardhan and Y. Srikant, "Exploiting critical data regions to reduce datacacheenergyconsumption,"IndianInstituteofScience,Bangalore, Tech. Rep., 2013.

[10]. S. Li, J. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in 42nd IEEE/ACM International Symposium on Microarchitecture (MICRO), 2009, pp. 469–480.

[11]. B. Calder, D. Grunwald, and J. Emer, "Predictive sequential associative cache," in International Symposium on High-Performance Computer Architecture, 1996, pp. 244–253.

[12]. K. Inoue, T. Ishihara, and K. Murakami, "Way-predicting setassociative cache for high performance and low energy consumption," ininternationalsymposiumonLowpowerelectronicsanddesign,1999, pp. 273–275.

[13]. S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," in 28th international symposium on Computer architecture (ISCA), 2001, pp. 240–251. M. Powell, A. Agrawal, T. Vijaykumar, B. Falsafi, and K. Roy, "Reducingset-associativecacheenergyviaway-predictionandselectivedirectmapping," in 34th International Symposium on Microarchitecture, 2001, pp. 54–65.

[14]. P. Carazo Minguela, R. Apolloni, F. Castro, D. Chaver, L. Pinuel, and F. Tirado, "L1 Data Cache Power Reduction using a Forwarding Predictor," Integrated Circuit and System Design. Power and Timing Modeling, Optimization, and Simulation, pp. 116–125, 2011.

[15]. S. Kim and J. Lee, "Write buffer-oriented energy reduction in the L1 data cache of two-level caches for the embedded system," in 20th Great Lakes symposium on VLSI. ACM, 2010, pp. 257–262.

[16]. Z. Fang, L. Zhao, X. Jiang, S. Lu, R. Iyer, T. Li, and S. Lee, "Reducing L1 caches power by exploiting software semantics," in International Symposium on Low Power Electronics and Design (ISLPED), 2012.

[17]. D. Nicolaescu, B. Salamat, A. Veidenbaum, and M. Valero, "Fast speculative address generation and way caching for reducing L1 data cache energy," in International Conference on Computer Design (ICCD), 2006, pp. 101–107.

[18]. A. Bardine, M. Comparetti, P. Foglia, and C. A. Prete, "Evaluation of leakage reduction alternatives for deep submicron dynamic nonuniform cache architecture caches," in IEEE Transactions on VLSI, 2013.
[19]. A. Bardine, M. Comparetti, P. Foglia, G. Gabrielli, C. Prete, and P. Stenstr¨om, "Leveraging data promotion for low power D-NUCA caches," in 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools (DSD). IEEE, 2008, pp. 307–316.

Dr. Mohammad Mahmood Otoom. " Techniques for Cache Improvement." IOSR Journal of Computer Engineering (IOSR-JCE) 20.5 (2018): 49-53.