Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 425 – 433

doi:10.3906/elk-1201-90

Early wakeup: improving the drowsy cache performance

Sunghoon SHIM[1], **Sungwoo CHUNG**[2], **Hongjun CHOI**[3], **Cheol Hong KIM**[3,*]
[1]Computing Platform Team, Semiconductor Division, Samsung Electronics, Yongin, Korea
[2]Division of Computer and Communication Engineering, Korea University, Seoul, Korea
[3]School of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea

Received: 25.01.2012 • **Accepted:** 22.11.2012 • **Published Online:** 17.01.2014 • **Printed:** 14.02.2014

Abstract: As process technology scales down, leakage power consumption becomes comparable to dynamic power consumption. The drowsy cache technique is known as one of the most popular techniques for reducing the leakage power consumption in the data cache. However, the drowsy cache is reported to degrade the processor performance significantly. In this paper, to maintain the performance of the processor with the drowsy cache technique, we propose an early wakeup technique, which predicts the next cache line to be requested by utilizing the way-prediction information. The proposed technique efficiently reduces the number of accesses to the cache lines in drowsy mode. Our simulation results show that the proposed technique reduces the extra delay due to the drowsy cache scheme by 29.6%, on average.

Key words: Processor architecture, low-power design, leakage power, drowsy cache, early wakeup

1. Introduction

As more computing power of a processor is needed, the power dissipation in the processor inevitably increases. For this reason, reducing power dissipation in the processor becomes one of the most important design considerations. Among the computer components, the cache is reported to take account of a significant fraction of the total processor power consumption. Thus, the power efficiency of the cache should be carefully considered. In the past, dynamic power consumption was larger than leakage power consumption. However, the leakage power consumption becomes comparable to the dynamic power consumption as the number of transistors employed in a processor increases. To reduce the leakage power consumption of the caches, various techniques have been proposed [1–3]. Powell et al. proposed the gated-V_{dd} [1], a circuit-level technique to gate the supply voltage, resulting in reduced leakage power in unused memory cells. Cache decay [2] reduces the leakage power consumption by invalidating and turning off the cache lines when they hold data that are not likely to be reused, based on the gated-V_{dd} technique. As described in [2], after the cache line is turned off, the data stored in that cache line cannot be reused. Therefore, when the processor needs the cache line that has been turned off, it has to be fetched from the lower-level cache or the memory, resulting in significant performance degradation. The drowsy cache scheme [3], which reduces the leakage power consumption with a multilevel supply voltage, is reported to be one of the most efficient leakage power reduction techniques. Each cache line has 2 modes in the drowsy cache scheme: normal mode and drowsy mode. The supply voltage for drowsy mode is lower than that for normal mode to save the leakage power consumption. The drowsy cache technique changes the mode of the cache line that has not been used frequently into low power consumption mode (drowsy mode) instead of

*Correspondence: chkim22@jnu.ac.kr

turning off the cache line. Contrary to the cache decay technique, when the data stored in the cache line in the drowsy mode is accessed, it just requires extra cycle(s) to wake up the cache line. The wakeup means that the cache line is changed to normal mode from drowsy mode. The drowsy technique shows better performance than the decay technique, because there is no need to fetch the data from the lower-level memory when the data in the cache line in the drowsy mode are required. However, the extra cycle still degrades the performance.

Several research groups have focused on the drowsy cache [4–7]. Research dealing with the drowsy cache has mainly focused on the energy reduction, while this work focuses on the performance improvement of the drowsy cache by reducing the extra cycles to wake up the drowsy cache lines. For instruction caches, a wakeup prediction technique based on branch prediction was proposed [6], which is not applicable to data caches. The improved drowsy (ID) technique, which determines the states of cache lines based on the locality, was proposed to improve the efficiency of the drowsy instruction cache [7]. The ID technique also cannot be applied to the drowsy data cache because the locality of the data cache is worse than that of the instruction cache. For data caches, our previous work [8] was the first technique for wakeup predictions. In this work, we elaborate our previous scheme more clearly to apply it to the up-to-date hierarchical memory system and show more accurate simulation results.

In this paper, we present a next line prediction technique, called early wakeup, to alleviate the performance degradation in the drowsy cache scheme. By applying the proposed early wakeup technique, the next expected cache line in the drowsy mode is woken up before the cache line is accessed, leading to a performance improvement compared to the conventional drowsy cache technique.

2. Early wakeup technique

In the conventional drowsy cache scheme [3], each cache line has 2 modes: one is a low power mode, also called the drowsy mode, and the other is a normal mode. Leakage power consumption can be reduced significantly by using the drowsy mode. In other words, when the cache line is not expected to be used in the near future, the supply voltage of the cache line is reduced to a lower value, typically close to the threshold voltage, leading to lower leakage power. In normal mode with nominal voltage, the cache line operates the same as in a conventional cache. In drowsy mode, however, the cache line cannot be accessed even though data are retained. After being awakened, the cache line can be accessed. Therefore, when the cache line in the drowsy mode is requested by the processor core, the drowsy cache scheme requires extra cycle(s) for waking up the cache line, resulting in performance degradation. It also causes an increase of power consumption in the processor due to increased execution time. If the cache line in the drowsy mode can be woken up prior to the access from the processor core, the extra cycle to wake up the cache line can be removed. For this reason, we propose an early wakeup technique that can wake up the cache line in the drowsy mode before the access to that line.

The proposed early wakeup technique is based on a program counter (PC)-based way-prediction technique [9] that was originally proposed to reduce the dynamic power dissipation in the set-associative cache architecture. Way-prediction techniques can be categorized into 2 groups: PC-based way-prediction schemes and XOR-based way-prediction schemes. PC-based schemes look up a prediction table using a PC. The prediction table contains the predicted way information, indexed by the PC, and the information is updated based on the execution history [9]. Unlike the PC-based scheme, the XOR-based way-prediction scheme accesses the prediction table using the XOR value of the source register and offset [9]. Figure 1 shows an overview of the PC-based and XOR-based way-prediction schemes, and, as depicted, the PC-based scheme can provide the prediction information earlier than the XOR-based scheme. Moreover, the PC-based scheme is simpler than the XOR-based scheme from

the perspective of hardware complexity. Therefore, the proposed early wakeup technique utilizes the PC-based way-prediction scheme to determine the cache line to wake up before the actual cache access.

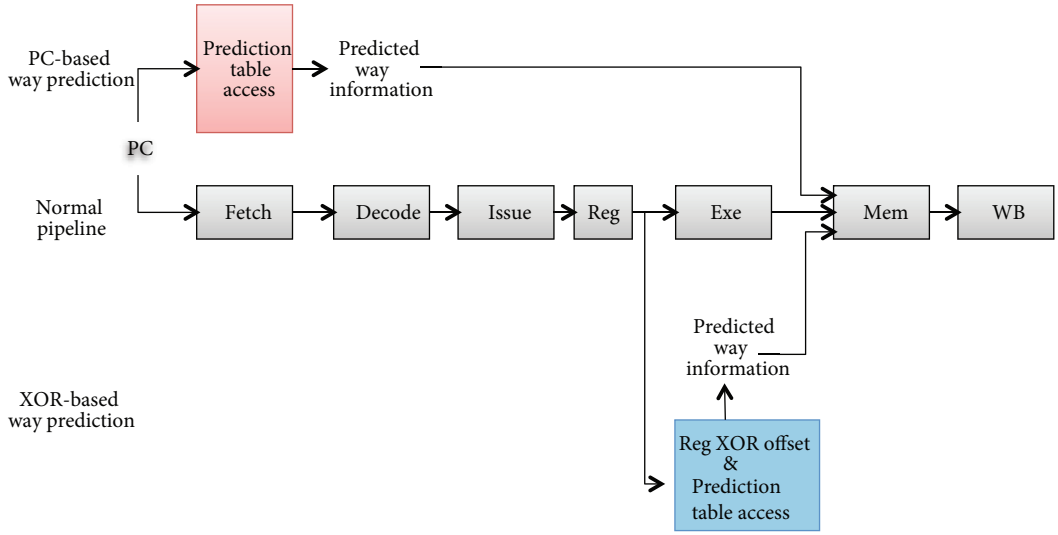

Figure 1. An overview of the way prediction schemes.

Each entry of the prediction table in the proposed early wakeup method is composed of a set index field and way-number field, as shown in Figure 2. The length of a set index field is the same as the bit length of the L1 data cache index field. The length of a way-number field corresponds to the logarithm value of the number of ways. When a cache block is accessed in the L1 data cache, the corresponding entry (set index and way-number) in the prediction table is updated. Thus, the prediction table includes the L1 cache access information based on the PC. Our scheme mainly relies on the fact that the same data block is accessed repeatedly when the applications are executed. Previous studies have shown that such data locality is prevalent due to common code patterns [10].

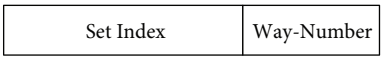

Figure 2. An entry of the prediction table.

Figure 3 shows the pipeline structure, including the proposed early wakeup scheme. When an instruction is fetched in the fetch stage, the prediction table is also accessed with the same PC. The corresponding entry (set index and way-number) in the prediction table is known in the decode stage, and changing the state of the cache line from drowsy mode to normal mode usually takes 1–2 cycles [3]. Therefore, if the fetched instruction is a load/store instruction requiring the data cache access and the access to the prediction table turns out to be a hit, the data cache line related to the selected table entry can be activated after 2–3 cycles from the end of the fetch stage. In a case where there is just one stage between fetch and mem, the proposed technique has no performance gain compared to the conventional drowsy cache. The proposed technique can provide better performance compared to the conventional drowsy cache if the number of stages between fetch and mem is larger than one, even though it cannot provide a full performance gain. The proposed technique can have a full performance gain if the number of cycles required to change the state of the cache line from drowsy mode to normal mode is n and the number of stages between fetch and mem is n + 1 or more. For example, if there

are only 2 stages between fetch and mem, the proposed technique cannot have a full performance gain but can have better performance than the conventional drowsy cache in a case where the number of cycles required to change the state of the cache line from drowsy mode to normal mode is 2, and it can have a full performance gain if the number of cycles to change the state is 1. In recent commercial processors from Intel, AMD, or even ARM, the number of cycles between the fetch stage and mem stage is larger than 3 cycles [11–14]. Therefore, the proposed early wakeup technique enables waking up the cache line in the drowsy mode prior to the actual cache access (mem stage), leading to reduced extra delay in the drowsy cache scheme.

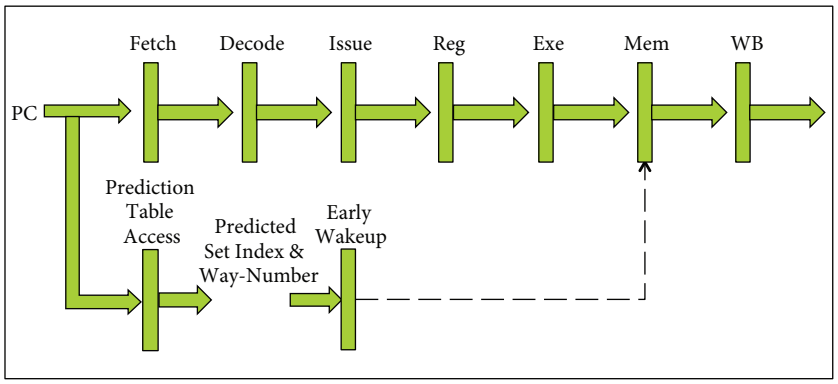

Figure 3. Pipeline structure for the early wakeup.

The prediction accuracy of the prediction table is an important factor for the efficiency of the proposed technique, because wakeup misprediction of the cache line causes power overhead. The prediction accuracy is dependent on the data locality and the number of entries in the prediction table. We expect that the data locality is high, since the data are likely to be accessed repeatedly. The optimal number of entries in the prediction table can be found based on the detailed simulations.

As shown in the advanced circuit technique for the drowsy cache (Figure 4), the wakeup can be done by increasing the supply voltage [3,15]. Therefore, the proposed wakeup does not require the activation of bitline or wordline in the memory, resulting in an acceptable overhead. In the proposed architecture, the access for the wakeup and the access for the read/write can be processed in parallel by appending a 1-bit wakeup signal, 1-address decoder inside of the data cache, and 1-address port for the wakeup request.

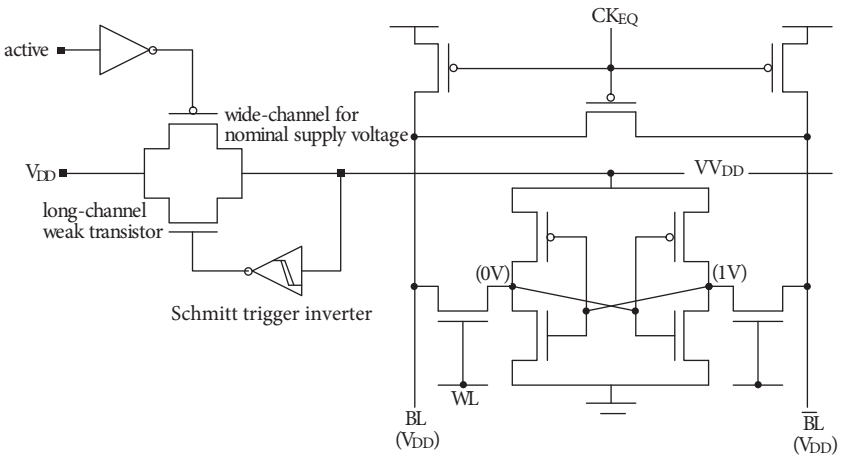

Figure 4. Implementation of the super drowsy cache line.

The mechanism of the proposed early wakeup technique can be summarized as follows: when an instruction is fetched, the prediction table is accessed using the PC. In a case where the fetched instruction is not a load/store instruction or the access to the prediction table turns out to be a miss, there is no operational difference between the proposed architecture and the conventional drowsy cache architecture. Otherwise (the fetched instruction is a load/store instruction and the access to the prediction table turns out to be a hit), the wakeup can be processed using the information (predicted set index and predicted way) from the corresponding entry of the prediction table, as shown in Figure 5, resulting in reduced performance degradation in the drowsy cache architecture.

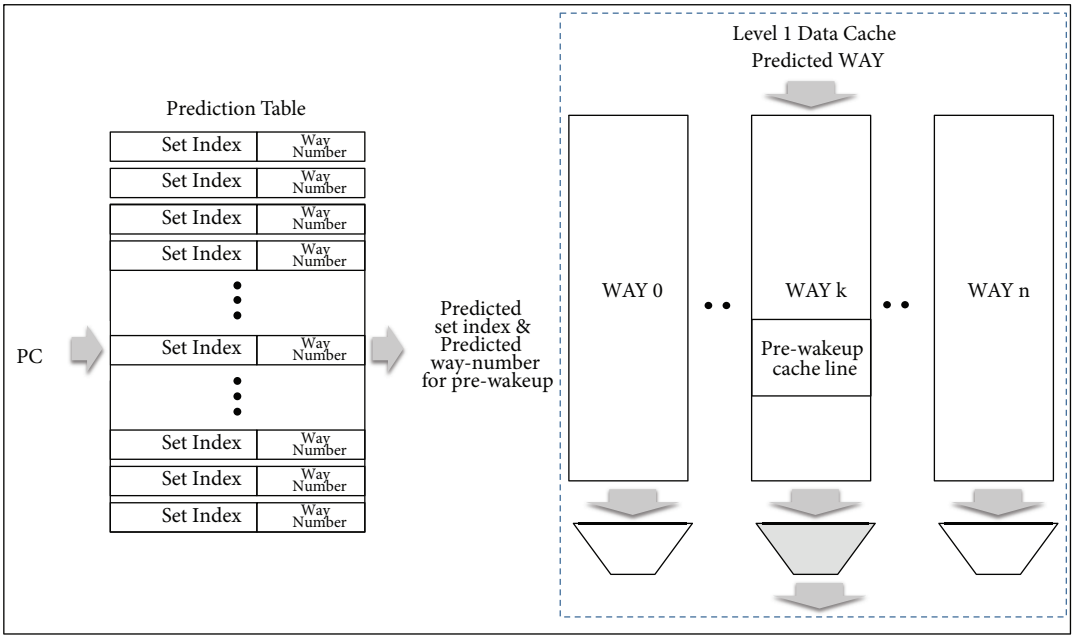

Figure 5. An overview of the proposed early wakeup technique.

3. Experiments

In order to evaluate the efficiency of the proposed early wakeup technique, we use the modified Wattch simulator [16]. The processor and memory configuration parameters used in this simulation are shown in the Table. We choose 20 applications from the SPEC CPU2000. The number of prediction table entries applied to the proposed scheme is varied to find the optimal number of entries.

Table. System configuration parameters.

Parameter	Value
CPU	2 issues per cycle
L1 I-cache	32 KB, 2-way, 32-byte block, 1-cycle latency
L1 D-cache	32 KB, 2-way, 32-byte block, 1-cycle latency
L2 cache	Unified, 4-way, 256 KB, 64-byte block, 8-cycle latency
Memory	64-cycle latency
Instruction translation lookaside buffer (TLB)	16 entries
Data TLB	32 entries
Prediction table	1024, 512, 256, 128, and 64 entries

Figure 6 shows the normalized number of accesses to the cache lines in the drowsy mode. In the graph, drowsy represents the conventional drowsy cache scheme, where 1024, 512, 256, 128, and 64 denote the proposed early wakeup cache scheme with 1024, 512, 256, 128, and 64 prediction table entries, respectively. As shown in the graph, the proposed early wakeup technique reduces the number of accesses to the cache lines in the drowsy mode compared to the conventional drowsy cache scheme by waking up the cache lines in the drowsy mode prior to the actual cache accesses, resulting in reduced extra cycles due to the drowsy cache scheme. On average, it reduces the number of accesses to the cache lines in the drowsy mode by 37.1% (1024 entries), 34.5% (512 entries), 30.6% (256 entries), 25.5% (128 entries), and 20.4% (64 entries). As the number of accesses to the cache lines in the drowsy mode is reduced, the performance is expected to be improved due to reduced extra cycles.

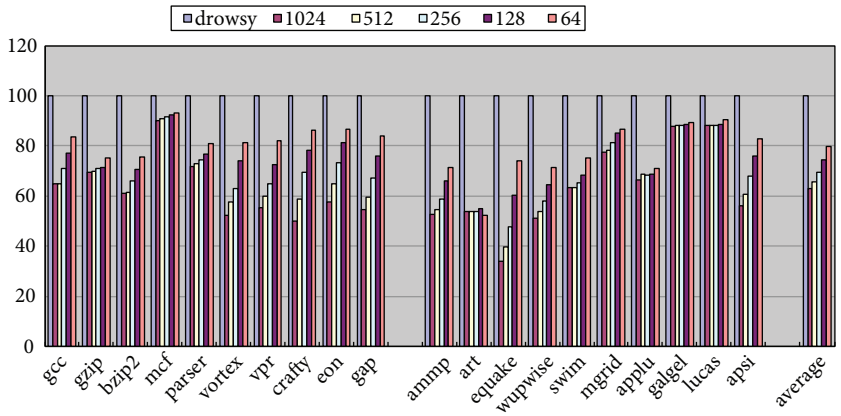

Figure 6. Normalized number of accesses to the cache lines in the drowsy mode (%).

In most benchmark applications, the number of prediction table entries is strongly correlated to the reduced number of accesses to the cache lines in the drowsy mode. As the number of prediction table entries increases, the number of accesses to the cache lines in the drowsy mode is reduced. However, in some applications, such as galgel, lucas, gzip, mcf, and art, where the data locality is high, the number of accesses to the cache lines in the drowsy mode is saturated.

Figure 7 shows the relative performance degradation compared to the processor without the drowsy cache technique. In all simulated applications, the proposed architecture with 1024 prediction table entries shows less performance degradation (0.07%–2.39%) than the conventional drowsy cache architecture (0.11%–4.59%). Moreover, the proposed technique shows little performance degradation (less than 1.5%) in most of the applications, while it reduces the leakage energy consumption significantly, leading to improved energy-delay efficiency.

Figure 8 shows the cache line rates in the drowsy mode. The average cache line rates in the drowsy mode of the proposed early wakeup cache scheme with 64 entries, 128 entries, 256 entries, 512 entries, and 1024 entries are 88.53%, 87.92%, 87.36%, 87.11%, and 87.14%, respectively. The average rate of the cache lines in the drowsy mode of the conventional drowsy cache scheme is 89.80%. As the number of prediction table entries increases, the cache line rates in the drowsy mode decrease, because a larger number of prediction table entries leads to more early wakeups. In some applications, the cache line rate in the drowsy mode with a small-size prediction table is higher than that with a large-size prediction table. This is caused by mispredictions in the prediction table. In the proposed scheme, on average, more than 87% of all of the cache lines are not activated during

the cache access since they are in the drowsy mode. Therefore, we can notice that the proposed early wakeup scheme reduces the leakage power consumption in the cache as much as the conventional drowsy cache scheme. Moreover, the proposed early wakeup scheme can improve the performance compared to the conventional drowsy cache scheme by reducing the number of accesses to the cache lines in the drowsy mode.

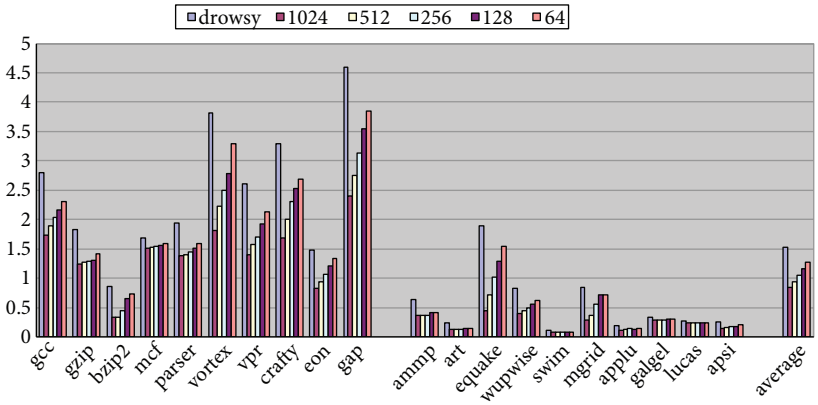

Figure 7. Performance degradation due to the wakeup delay (%).

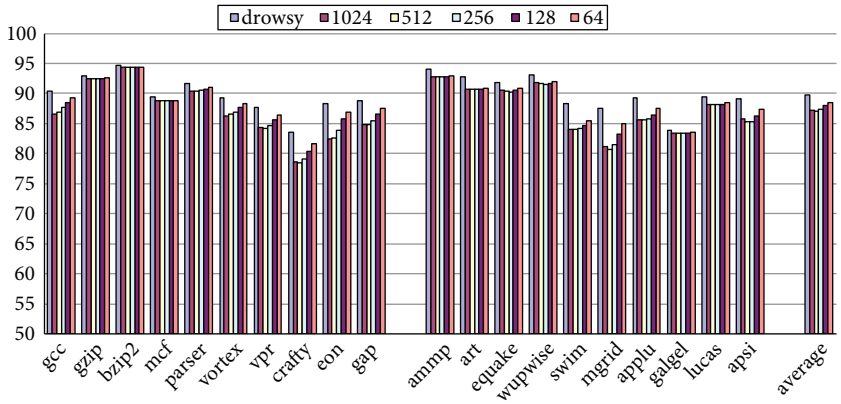

Figure 8. Cache line rates in the drowsy mode (%).

The area overhead for the proposed technique is the prediction table and the logic for the wakeup access, such as the 1-bit wakeup signal, 1-address decoder inside of the data cache, and 1-address port for the wakeup request. The energy overhead due to the proposed technique is depicted in Figure 9. In the graph, we assume that the L1 data cache energy consumption can amount to 15% of the total processor energy consumption [17]. In the graph, the vertical axis denotes the increased energy consumption in the processor due to the proposed technique compared to the processor with the conventional drowsy data cache. As the size of the prediction table increases, the dynamic energy consumption in the prediction table increases, resulting in increased energy consumption in the processor (0.1% with a 64-entry prediction table, 0.32% with a 1024-entry prediction table). We believe that the area/energy overhead caused by the proposed technique is acceptable considering the reduced extra delay (29.6%) in the drowsy data cache.

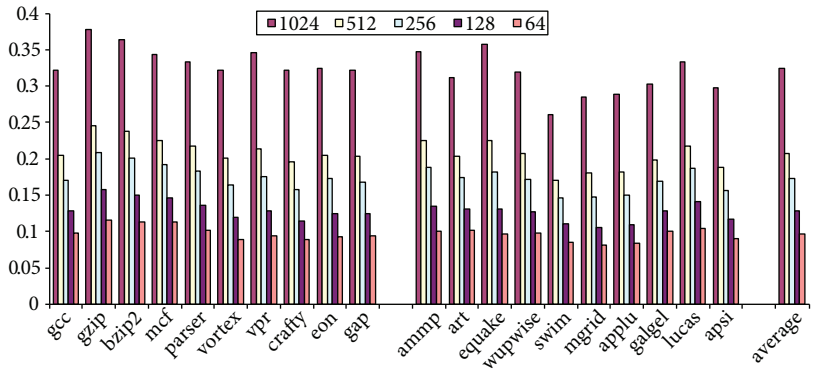

Figure 9. Energy overhead of the proposed technique (%) (increased energy consumption rate in the processor).

4. Conclusions

To reduce the performance degradation in the conventional drowsy cache scheme, we proposed an early wakeup technique to reduce the extra delay due to the accesses to the cache lines in the drowsy mode. By waking up the cache lines in the drowsy mode prior to the actual access based on the prediction mechanism, the extra delay due to the drowsy cache scheme can be reduced significantly. Simulation results show that the proposed early wakeup technique reduces 20.4% (64 entries) to 37.1% (1024 entries) of the extra cycles caused by waking up the cache lines in the drowsy mode compared to the conventional drowsy cache scheme. Moreover, in the proposed early wakeup scheme, the cache line rate in the drowsy mode is 87.14% (1024 entries) to 88.53% (64 entries), which is comparable to the conventional drowsy cache scheme rate (89.80%). In other words, the proposed cache scheme reduces the leakage power consumption as much as the conventional drowsy cache scheme.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education, Science, and Technology (2012R1A1B4003492) and the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the National IT Industry Promotion Agency (NIPA-2012-H0301-12-3005).

References

[1] M. Powell, S.H. Yang, B. Falsafi, K. Roy, T.N. Vijaykumar, "Gated-V_{dd}: a circuit technique to reduce leakage in deep-submicron cache memories", Proceedings of the International Symposium on Low Power Electronics and Design, pp. 90–95, 2000.

[2] S. Kaxiras, Z. Hu, M. Martonosi, "Cache decay: exploiting generational behavior to reduce leakage power", Proceedings of the International Symposium on Computer Architecture, pp. 240–251, 2001.

[3] K. Flautner, N.S. Kim, S. Martin, D. Blaauw, T. Mudge, "Drowsy caches: simple techniques for reducing leakage power", Proceedings of the International Symposium on Computer Architecture, pp. 148–157, 2002.

[4] R. Giorgi, P. Bennati, "Filtering drowsy instruction cache to achieve better efficiency", Proceedings of the ACM Symposium on Applied Computing, pp. 1554–1555, 2008.

[5] S. Petit, J. Sahuquillo, J.M. Such, D. Kaeli, "Exploiting temporal locality in drowsy cache policies", Proceedings of the 2nd Conference on Computing Frontiers, pp. 371–377, 2005.

[6] S.W. Chung, K. Skadron, "On-demand solution to minimize I-cache leakage energy with maintaining performance", IEEE Transactions on Computers, Vol. 57, pp. 7–24, 2008.

[7] M.B.C. Alioto, P. Bennati, R. Giorgi, "Exploiting locality to improve leakage reduction in embedded drowsy I-caches at same area/speed", Proceedings of the International Symposium on Circuits and Systems, pp. 37–40, 2010.

[8] S.H. Shim, C.H. Kim, J.W. Kwak, C.S. Jhon, "Hybrid technique for reducing energy consumption in high performance embedded processor", Proceedings of the International Conference on Embedded and Ubiquitous Computing, Vol. 3207, pp. 74–84, 2004.

[9] M.D. Powell, A. Agarwal, T.N. Vijaykumar, B. Falsafi, K. Roy, "Reducing set-associative cache power via way-prediction and selective direct-mapping", Proceedings of the International Symposium on Microarchitecture, pp. 54–65, 2001.

[10] B. Batson, T.N. Vijaykumar, "Reactive associative caches", Proceedings of the International Conference on Parallel Architectures and Compilation, 2001.

[11] ARM Co., ARM Technical Reference Manual, available at http://infocenter.arm.com/help/index.jsp.

[12] T.R. Halfhill, "Intel's tiny ATOM: new low-power microarchitecture rejuvenates the embedded x86", Microprocessor Report, Vol. 22, pp. 1–16, 2008.

[13] M.E. Thomadakis, "The architecture of the Nehalem processor and Nehalem-EP SMP platforms", Research Report, Texas A&M University, 2011.

[14] K. Diefendorff, "K7 challenges Intel: new AMD processor could beat Intel's Katmai", Microprocessor Report, Vol. 12, pp. 1–7, 1998.

[15] N.S. Kim, K. Flautner, D. Blaauw, T. Mudge, "Single-V_{dd} and single-V_T super-drowsy techniques for low-leakage high-performance instruction caches", Proceedings of the International Symposium on Low Power Electronics and Design, pp. 54–57, 2004.

[16] D. Brooks, V. Tiwari, M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations", Proceedings of the 27th Annual International Symposium on Computer Architecture, pp. 83–94, 2000.

[17] D. Nicolaescu, A. Veidenbaum, A. Nicolau, "Reducing data cache energy consumption via cached load/store queue", Proceedings of the International Symposium on Low Power Electronics and Design, pp. 252–257, 2003.