EURASIP Journal on Embedded Systems
a SpringerOpen Journal

RESEARCH Open Access

Dual-access way-prediction cache for embedded systems

Yul Chu[1*] and Jin Hwan Park[2]

Abstract

Way-prediction (WP) caches have advantages of reducing power consumption and latency for highly associative data caches and thus are favorable for embedded systems. In this paper, we propose an enhanced way-prediction cache, dual-access way-prediction (DAWP) cache, to cope with the weakness of the WP cache. The prediction logic designed for the DAWP cache contains a scaled index table, a global history register, and a fully associative cache to achieve higher prediction accuracy, which eventually yields less energy consumption and latency. In our practice, performance measurement is done with a simulation model, which is implemented with SimpleScalar and CACTI, and nine SPEC2000 benchmark programs. Our experimental results show that the proposed DAWP cache is highly efficient in power and latency for highly associative cache structures. The efficiency is increased with the increasing associativity, and the testing results with 64 KB cache show that the DAWP cache achieves 16.45% ~ 75.85% power gain and 4.91% ~ 26.96% latency gain for 2-way ~ 32-way structures, respectively. It is also observed that the random replacement policy yields better efficiency in power and latency than the LRU (least recently used) policy with the DAWP cache.

1. Introduction

Since last decade, low-power system design has been a hot issue for embedded systems, especially for hand-held devices, which are dependent on the limited battery power. With the most of embedded systems, the major source of energy consumption has been known as microprocessor and cache. In fact, it is reported that cache memories occupy more than 60% of the microprocessors' die area and consume more than 40% of the total system power [1-5]. To reduce the power dissipation in an embedded system, it is highly desired to design and use an energy-efficient cache system.

In general, on-chip caches used in mobile devices are highly associative with more than 16-way sets to provide better performance by reducing the costly access to the memory. The highly associative caches proposed in [6-8] are specifically designed for embedded systems to provide better performance by reducing the conflict misses, which are due to imperfect allocations of entries in the cache. However, they significantly increase the system power consumption due to the simultaneous accesses to all the banks in the cache, e.g., n-way set-associative cache has n banks to be accessed simultaneously. A basic method of reducing power consumption in a cache system is reducing the number of bank accesses to charge the bit-lines of the cache memory [4,9]. In this paper, we aim to reduce the power consumption of the highly associative caches used in embedded systems by accurately predicting the target bank, i.e., accessing only one bank from all the banks in the cache, to access the referenced data. The resulting cache system is named dual-access way-prediction (DAWP) cache and it not only saves the power but also reduces the latency since the cache is accessed as a direct-mapped cache when the prediction is hit.

The rest of this paper is organized as follows. A brief review of some related work is provided in Section 2. In Section 3, the proposed dual-access way-prediction cache is described. In Section 4, simulation model and performance measurement metrics used in our practice are described. In Section 5, experimental results and discussions are provided, and finally, Section 6 concludes the paper.

2. Related work

Researchers have proposed various methods to reduce the power consumption of highly associative cache memories.

* Correspondence: chuy@utpa.edu
[1]University of Texas Pan American, Edinburg, TX 78539, USA
Full list of author information is available at the end of the article

One popular approach is using a phased cache in which the cache is divided into two parts, i.e., tag part and data part [10,11]. In the phased cache, tag bits for all tag banks are enabled (powered) and checked with the memory reference. On a hit in a bank, the bank is enabled and accessed during the next cycle. Although the phased cache can reduce the power consumption in a certain amount, it has a disadvantage caused by using more clock cycles to fetch the desired data, compared to other conventional set-associative caches.

Another popular approach is using a way-prediction (WP) cache [10,12] in which a prediction logic is added to a conventional set-associative cache structure, such as 2-way set, 4-way set, etc. The prediction logic predicts a way, which is the target bank to be accessed. On a prediction miss, all remaining banks in the cache must be accessed. It is known that the way-prediction cache reduces energy consumption more effectively than the phased cache [10]. Although the prediction logic itself consumes some amount of extra energy, the cache system consumes less total energy than the conventional cache. One additional gain from using such prediction is reduced latency since the cache behaves like a direct-mapped cache when a prediction hits [10]. An approach proposed in [13] shows better power reduction compared to a way-prediction cache, but it employs complicated cache structures, such as two cache structures (way-prediction and phased structures) and multicolumn-based way-prediction mechanism, by using 2-way branch prediction techniques.

The way-prediction logic implemented in [10] is a simple *most recently accessed* logic, which uses the same number of index table entries as the number of sets in the cache, and the energy efficiency of the cache system is entirely dependent on the prediction hit rate of the predictor. A weak point of this mechanism is that the number of entries in the index table, which is a prediction buffer, is reduced by the increased associativity of the cache (assuming the cache size is fixed), and thus the prediction accuracy is reduced accordingly. For example, changing from 2-way to 4-way makes the number of index table entries reduced by half and, thus, yields relatively reduced prediction accuracy. Figure 1 (based on [10,14]) shows the power consumption of the WP cache for different associativity levels. The values shown are normalized to the power consumption value of the conventional 4-way set-associative cache. As shown in the figure, the average power consumption of the WP cache outgrows as the associativity increases. Although it is not shown in the figure, the power consumption of the conventional cache increases almost linearly as the associativity increases and, thus, the power gain rate of the WP cache significantly decreases as the associativity increases (e.g., 16-way and 32-way in Figure 1). This becomes the motivation of our research and we focus on achieving the higher power efficiency for highly associative cache structures.

3. Dual-access way-prediction cache

In this section, we describe our proposed low-power/latency cache named dual-access way-prediction or DAWP cache, which is an enhanced way-prediction cache and is suitable for building energy-efficient embedded systems for highly associative cache structures.

Figure 2 shows a typical WP cache in which the index table (a prediction buffer) keeps previously accessed

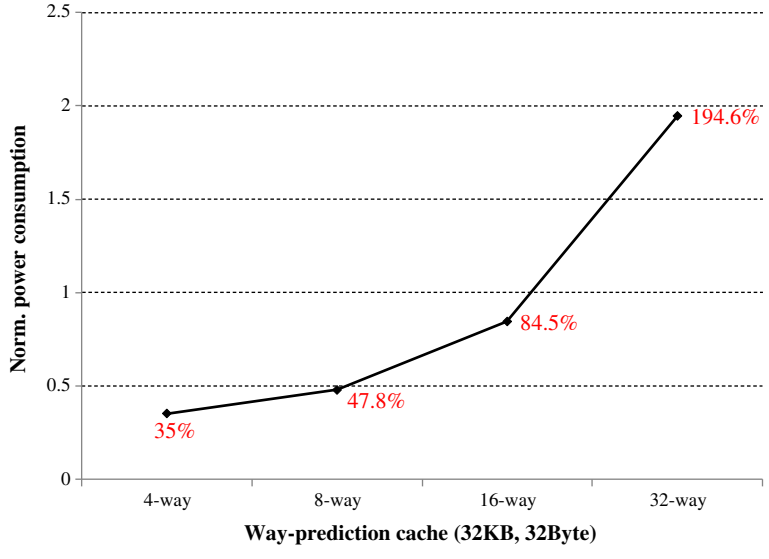

Figure 1 Power consumption of WP cache.

bank (way) numbers and the prediction logic speculatively chooses one bank from the entries of the index table and then accesses the predicted bank. In the case that the prediction is a hit, the cache access is completed with the predicted bank, otherwise, all other banks in the cache are accessed using the normal cache access process. As described in Section 2, the dedicated index table size degrades the gain from using the prediction logic when associativity increases.

Figure 3 shows the proposed DAWP cache in which the predictor consists of two major parts. One part is the index table, which stores previously accessed bank (way) numbers. Unlike the WP cache, the size of this index table is stable, i.e., the number of table entries does not change depending on the change of the set associativity in the cache. To cope with the prediction power degradation in the WP cache, which we explained in Section 2, we scale up the number of the index table entries as the associativity increases to keep the same number of entries as the direct-mapped cache. The referenced entry in the index table (a candidate predicted bank) is sent to the 2×1 multiplexer to compete to the other candidate from the other part. The other major part of the predictor consists of a global history register and a small fully associative cache. The global history register stores a group of recently accessed six bank numbers, e.g., 0001 for bank_1, 0010 for bank_2, etc. The fully associative cache contains a fixed number of lines (2 to 4 lines, negligible size), each of which holds a selected global history and the corresponding predicted bank. As shown in Figure 3, a selected entry from the fully associative cache also is sent to the 2×1 multiplexor to compete to the counterpart candidate, which we described earlier. The global history and the fully associative cache are rarely enabled (valid), i.e., the valid signal in Figure 3 will be '1' only when there are constant trashing of data in the index table. If the global history is matched with an entry of the fully associative cache, the corresponding predicted bank (way) is accessed and checked for the presence of the referenced data. In the case that the data is not present, all other banks in the cache are checked for the data. On a cache miss, the cache is updated with the retrieved data from the lower-level memory, i.e., level-2 cache or main memory, and the bank information is updated in the index table and the fully associative cache. On a way-prediction miss but a cache hit, i.e., the referenced data is found in one of unpredicted banks, the index table entry is updated with the referenced bank.

The proposed prediction scheme reflects the spatial locality of data. In other words, a sequence of related data (spatial locality) tends to be located and accessed in the same bank (way). As we described earlier, the bank information for the prediction is updated via memory address (index part) and global history. To reduce the power consumption in the DAWP cache, it is desired to reduce conflicts in the index table and the fully associative cache. In fact in our design, the highly biased accesses are filtered using the global history, and the hit time is reduced since the system accesses only one bank instead of accessing full *n*-way banks on prediction hits.

The dual-access prediction mechanism used in the DAWP cache yields high accuracy since the scaled-up index table and the fully associative cache are used. Unlike the WP cache, the performance gain of the DAWP cache (against the conventional cache) increases with increasing set associativity of the cache.

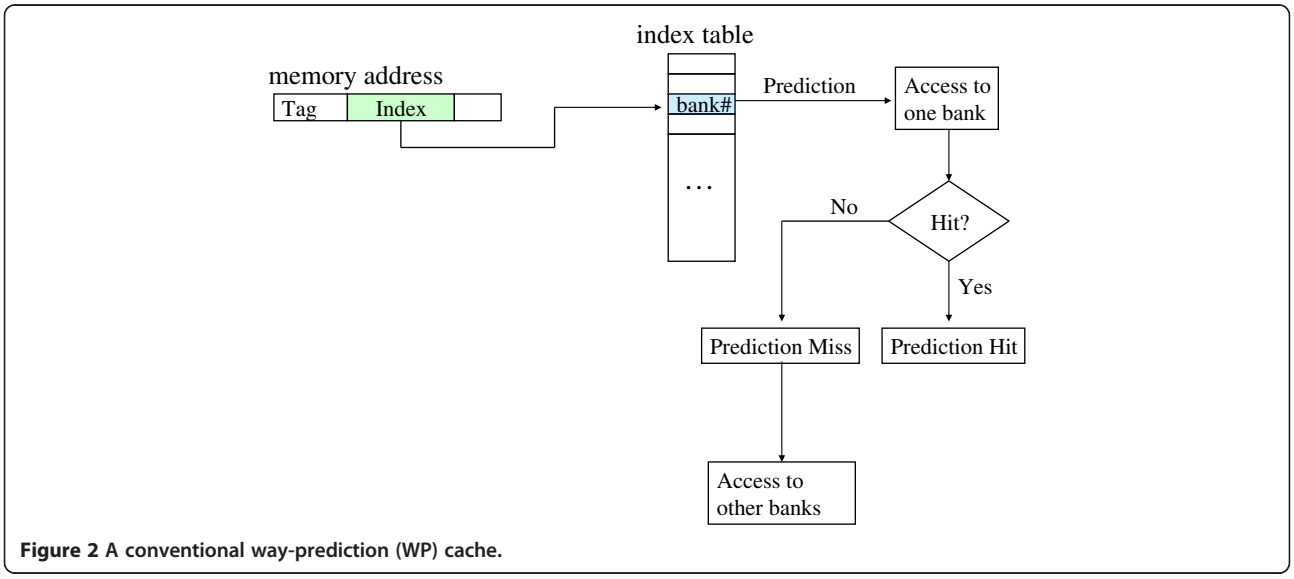

Figure 2 A conventional way-prediction (WP) cache.

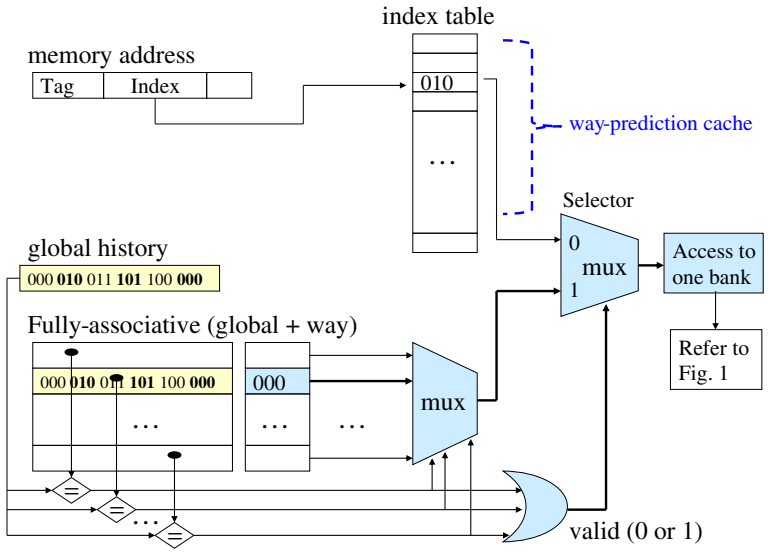

Figure 3 Dual-access way-prediction (DAWP) cache.

4. Simulation model and performance measurement metrics

In our practice, we built a simulator to measure the performance of the proposed DAWP cache. The simulation is done with 16, 32, and 64 KB cache sizes and six different cache structures, i.e., direct-mapped, 2-way, 4-way, 8-way, 16-way, and 32-way set-associative caches. It is assumed in the simulation that the cache uses WB (write back) and write-allocation mechanisms on write-hit and write-miss, respectively. Our simulation model is based on SimpleScalar [15] and CACTI [16], and WP and DAWP cache modules are implemented and ported into the simulator. For performance measurement, we use nine SPEC2000 programs, which are art, ammp, equake, mesa, mcf, vpr, vortex, gcc, and gzip.

Figure 4 shows our simulation model in detail. As shown in the figure, the executables of the SPEC2000 benchmark programs are processed in the simulator to collect the following data for the tested caches: prediction hit rate, power consumption amount, cache miss rate, and latency.

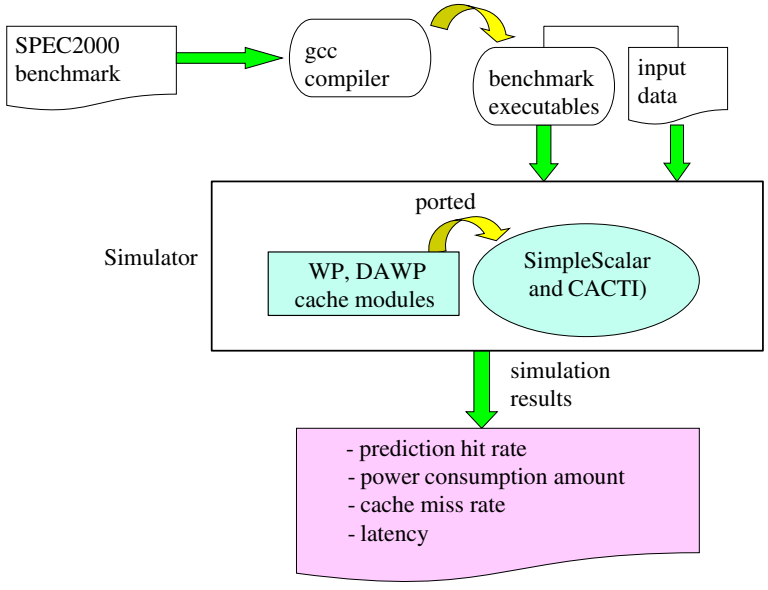

Figure 4 Simulation model.

In the simulator, CACTI [16] is used to calculate the memory access latency and power consumption amount, and we followed the estimation methodologies used in [17]. Other data, which are prediction hit/miss rate and cache hit/miss rate, are obtained from the SimpleScalar part of the simulator. In order to calculate the latency and power consumption amount, cache accesses are categorized into three cases: (1) The first case is a correct way-prediction. In this case, the latency and power dissipation are based on the accesses to the index table and the predicted bank, i.e., analogous to the operation on the direct-mapped cache. (2) The second case is a wrong way-prediction but is a cache hit. In this case, the way-predictor fails to select the correct bank, but the referenced data is found in a different bank. This requires the index table to be updated with the correct bank information after the data access. The latency and power dissipation are based on an access to the index table, an access to one bank, accesses to all other banks in the cache, e.g., *n-1* banks for *n*-way set, and an access to update the index table. (3) The last case is a cache miss, which is the worst case and consumes time and energy for all the accesses listed in the second case plus a costly access to the lower-level memory to fetch the referenced data.

5. Experimental results

The major goal of our proposed approach is raising up the way-prediction accuracy to achieve the power efficiency and lower latency. One factor that affects the prediction rate is the size of the index table, which is a prediction buffer. Figure 5 shows way-prediction miss rates for various index table sizes in a 32-way set-associative cache for the benchmark programs (averaged). In the figure, the index table size is normalized to the size used in the WP cache [10] in which the number of entries in the index table is the same as the number of sets in the cache. For example, with a 32-way set-associative cache having total *m* lines, the number of table entries is *m/32*. We let the base size, i.e., the index table size used in the WP cache, as 1X and other sizes (2X … 32X) are scaled-up sizes tested in the DAWP cache. As the associativity of the cache increases, the number of sets in the cache is reduced drastically, and so the number of the index table entries in the WP cache also is reduced. As we described in Section 2, this brings about the degraded prediction rate due to the conflicts in the index table. To cope with the degraded prediction rate, the size of the index table should be scaled up by multiplication factors of 2X, 4X, 8X, etc. As shown in Figure 5, we tested all possible scaling factors (2X … 32X) with the DAWP cache. For example, with the 32-way cache having total *m* lines, 2X, 4X, 8X, 16X, and 32X represent the number of table entries *m/16*, *m/8*, *m/4*, *m/2*, and *m*, respectively. Since the size of each entry in the index table ranges from 1 to 5 b only, the scaling up of the index table does not affect the total power consumption and latency significantly. As shown in Figure 5, the way-prediction miss rate decreases according to the increased scaling factor. In our practice with the given settings, prediction miss rate is saturated to approximately 5% with 16X and 32X scaling factors.

Another factor that affects performance (i.e., prediction rate and power consumption) is the replacement policy. In our practice, we evaluated the DAWP cache with two dominant replacement policies, least recently used (LRU) and random. In this test, we used the scaling factor corresponding to the ways used in the cache, e.g., *n*X for *n*-way set-associative cache. In general, LRU is

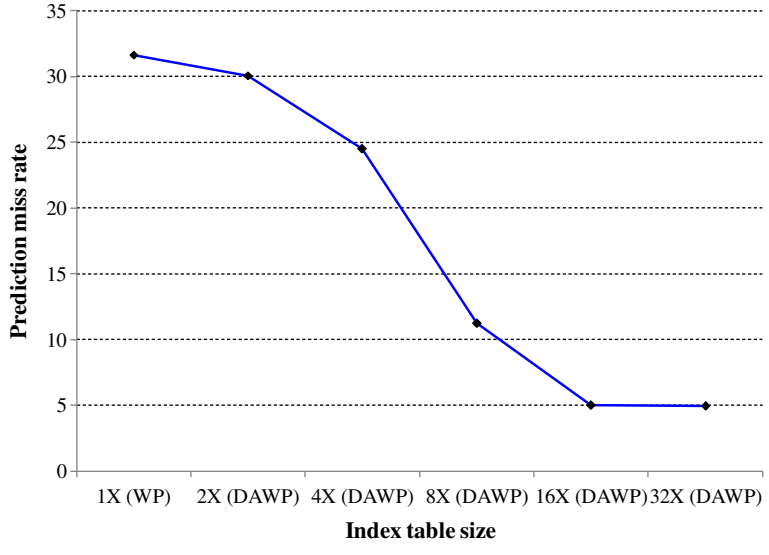

Figure 5 Prediction miss rate vs. index table size.

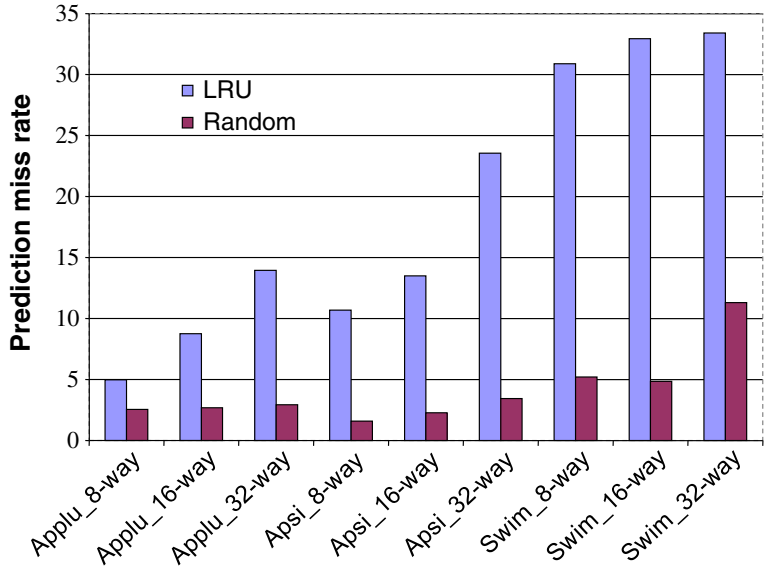

Figure 6 Prediction miss rate for LRU vs. random replacement policies.

known as providing lower miss rate with the cost of more complex hardware than the random policy. Figure 6 shows the prediction-miss rate comparison between the two replacement policies using SPEC2000 benchmark programs, which are applu, apsi, and swim. As shown in the figure, our experimental results reveal that the random policy yields the better performance than the LRU policy with DAWP cache since, for highly associative caches, the random policy can use all locations evenly compared to the LRU policy, which has a certain number of unused banks for long time. Figure 7 shows the total power dissipation comparison (in the arbitrary unit) between the two replacement policies using the identical benchmark programs used for the prediction-miss rate comparison. Again, it is revealed that the random policy yields the better power efficiency than the LRU policy. Therefore, we select and use the random replacement policy together with the maximum index table size scaling factor, i.e., nX for n-way set-associative cache, in the remaining experiments in this paper.

Figures 8 and 9 show the performance of DAWP cache compared to the conventional cache, which does not have the way-prediction logic, with the nine SPEC2000 benchmark programs. The values shown are averaged (harmonic mean) from the nine benchmark programs used and normalized to the value of the conventional

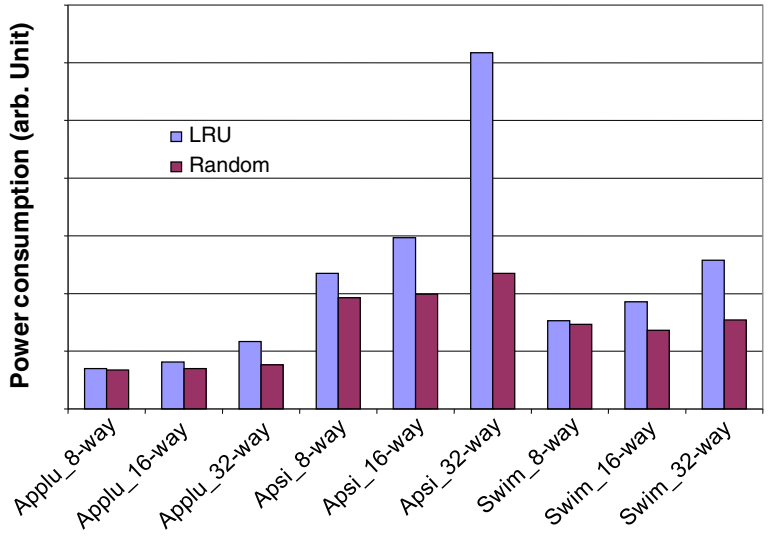

Figure 7 Power dissipation for LRU vs. random replacement policies.

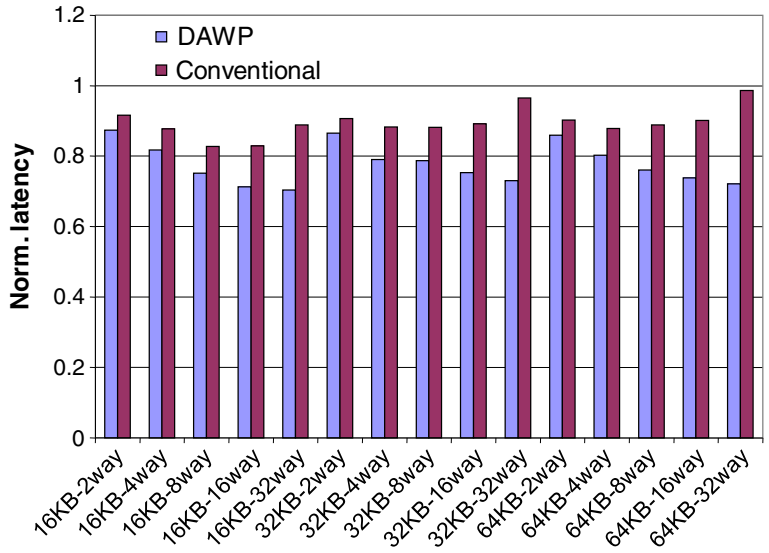

Figure 8 Harmonic mean of the latencies for SPEC2000 (normalized to the conventional direct-mapped cache).

direct-mapped cache, i.e., value 1 for the conventional direct-mapped cache. As shown in Figure 8, the DAWP cache structures yield lower latencies than the counterpart conventional cache structures in all the cases tested. In fact, the latency gain (DAWP cache against conventional cache) is amplified with the increasing associativity, e.g., latency gain of 32-way is greater than that of 16-way, and so on. This is due to the behavior of the DAWP cache in which highly accurate prediction hits cause the access operations analogous to the direct-mapped cache and thus, the higher associativity the more gain. For instance with 64 KB cache size, the DAWP cache achieves 4.91% ~ 26.96% latency gain for 2-way ~ 32-way structures, respectively. Figure 9 shows the total power dissipation from running the nine SPEC2000 benchmark programs (harmonic mean values) normalized to the corresponding conventional direct-mapped cache, i.e., value 1 for the conventional direct-mapped cache. For the power consumption measurement, CACTI [16] is used in our simulator and all the prediction components used in the DAWP cache including the index table and the small fully associative cache are counted. As shown in Figures 8 and 9, the gain of power efficiency is more rapidly amplified than the latency case

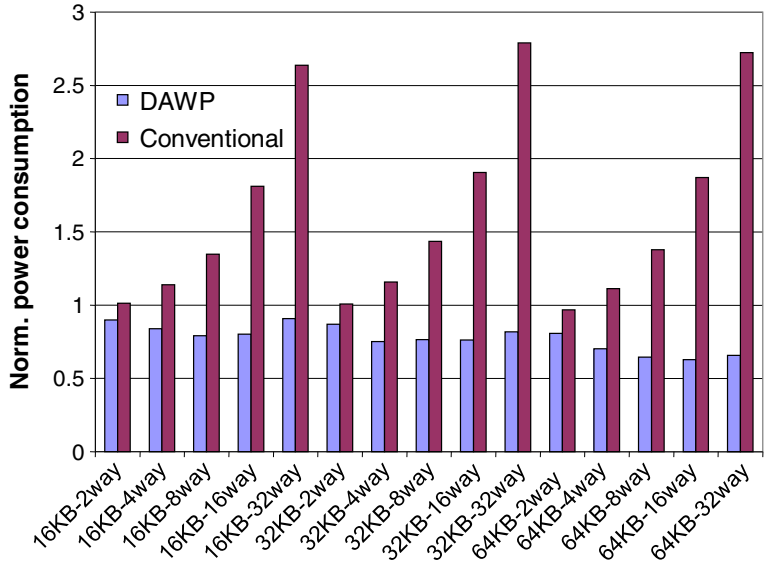

Figure 9 Harmonic mean of the power dissipations for SPEC2000 (normalized to the conventional direct-mapped cache).

(shown in Figure 2) as the associativity increases. For instance with 64 KB cache size, the DAWP cache achieves 16.45% ~ 75.85% power gain for 2-way ~ 32-way structures, respectively.

From our experiments, we observed that the proposed DAWP cache yields higher way-prediction accuracy than the WP cache for highly associative caches and yields considerably higher power/latency efficiency than the conventional cache with the benchmark programs that we used. We also observed that the power and latency gains are more significant according to the increasing set associativity.

6. Conclusions

Dual-access way-prediction (DAWP) cache, which is an enhanced way-prediction cache, is proposed and the performance measurement is done based on the simulation model built from the SimpleScalar and CACTI. In our practice, nine SPEC2000 benchmark programs are used to measure the power and latency efficiencies of the DAWP cache.

From our experiments, we observed that the proposed DAWP cache yields higher way-prediction accuracy than the WP cache for highly associative caches and yields considerably higher power/latency efficiency than the conventional cache with the benchmark programs that we used. We also observed that the power and latency gains from using the DAWP cache are more significant according to the increasing set associativity; in fact, it is more significant for the power efficiency gain than the latency gain. This demonstrates that the proposed DAWP cache is relatively more efficient with highly associative caches. One additional observation from our experiment is that the random replacement policy yields better performance in both latency and power dissipation than the LRU replacement policy with the DAWP cache.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]University of Texas Pan American, Edinburg, TX 78539, USA. [2]California State University, Fresno, CA 93740, USA.

Received: 17 July 2013 Accepted: 26 April 2014
Published: 20 May 2014

References

1. J Hennessy, DA Patterson, *Computer Architecture – a Quantitative Approach* (Morgan Kaufmann, Elsevier, Waltham, 2012)
2. J Montanaro, RT Witek, K Anne, AJ Black, EM Cooper, DW Dobberpuhl, PM Donahue, J Eno, GW Hoeppner, D Kruckemyer, TH Lee, PCM Lin, L Madden, D Murray, MH Pearce, S Santhanam, KJ Snyder, R Stehpany, SC Thierauf, A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor. IEEE J. Solid State Circuits **31**(11), 1703–1714 (1996)
3. MJ Flynn, P Hung, Microprocessor design issues: thoughts on the road ahead. IEEE Micro. **25**(3), 16–31 (2005)
4. C Zhang, A low power highly associative cache for embedded systems, in *Proceedings of the IEEE International Conference on Computer Design (ICCD), San Jose, CA, 1–4 October 2007*, pp. 31–36
5. M Alipour, K Moshari, M Bagheri, Performance per power optimum cache architecture for embedded applications, a design space exploration, in *Proceedings of the 2nd IEEE International Conference Networked Embedded Systems for Enterprise Applications (NESEA), Fremantle, WA, 8–9 December 2011*, pp. 1–6
6. SB Furber, ARP Thomas, HE Oldham, DW Howaid, ARM3 - 32b RISC processor with 4kbyte on-chip cache, in *VLSI: Proceedings of the IFIP TC 10/WG 10.5 International Conference on Very Large Scale Integration, Munich, 16–18 August 1989*, pp. 35–44
7. S Santhanam, AJ Baum, D Bertucci, M Braganza, K Broch, T Broch, J Burnette, E Chang, KT Chui, D Dobberpuhl, P Donahue, J Grodstein, I Kim, D Murray, M Pearce, A Silveria, D Soudalay, A Spink, R Stepanian, A Varadharajan, R Wen, A low-cost, 300-MHz, RISC CPU with attached media processor. IEEE J. Solid State Circuits **33**(11), 1829–1839 (1998)
8. Intel, *3rd Generation Intel Xscale® Microarchitecture–Developer's Manual*, 2007
9. MD Powell, A Agarwal, TN Vijaykumar, B Falsafi, K Roy, Reducing set-associative cache energy via way-prediction and selective direct-mapping, in *Proceedings of the 34th Annual ACM/IEEE International Symposium on Microarchitecture (MICRO), Austin, 1–5 December 2001*, pp. 54–65
10. K Inoue, T Ishihara, K Murakami, Way-predicting set-associative cache for high performance and low energy consumption, in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design, San Diego, 16–17 August 1999*, pp. 273–275
11. RK Megalingam, KB Deepu, IP Joseph, V Vikram, Phased set associative cache design for reduced power consumption, in *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009), Beijing, 8–11 August 2009*, pp. 551–556
12. B Batson, TN Vijaykumar, Reactive-associative caches, in *Proceedings of the 2001 International Conference on Parallel Architectures and Compilation Techniques, Barcelona, 8–12 September 2001*, pp. 49–60
13. Z Zhu, X Zhang, Access-mode predictions for low-power cache design. IEEE Micro. **22**(2), 58–71 (2002)
14. H Chen, J Chiang, Low-power way-predicting cache using valid-bit pre-decision for parallel architectures, in *Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA 2005), Taipei, 28–30 March 2005* pp. 203–206
15. DC Burger, TM Austin, The SimpleScalar tool set, version 2.0. Comput. Arch. News **25**(3), 13–25 (1997)
16. P Shivakumar, N Jouppi, *CACTI 3.0: An integrated cache timing, power, and area model, WRL Research Report 2001/2* (Compaq, Palo Alto, 2001)
17. G Contreras, M Martonosi, Power prediction for Intel XScale® processors using performance monitoring unit events, in *Proceedings of the 2005 International Symposium on Low Power Electronics and Design, San Diego, 8–10 August 2005*, pp. 221–226

doi:10.1186/1687-3963-2014-16
Cite this article as: Chu and Park: **Dual-access way-prediction cache for embedded systems.** *EURASIP Journal on Embedded Systems* 2014 **2014**:16.