

Implementation of L2 Way Tagged Cache Architecture

Aparna M S, Sharon Davis ,Sherin C George
M.Tech. Scholars, Embedded System
Sahrdaya College of Engineering and Technology ,Kodakara

Abstract- The advancements of semiconductor technology have boosted the rapid growth of very large scale integrated (VLSI) systems in our day-to-day life. Microprocessors and systems-on-chip (SOCs) are now extensively used in a variety of applications ranging from smart phones to handheld computers, from entertainment systems to sophisticated automotive controllers, and from gaming devices to life-saving medical equipment. The processing speed or performance of these systems is primarily limited by the power budget, which is determined by the battery life for mobile devices. Many high-performance microprocessors employ cache write-through policy for performance improvement and at the same time achieving good tolerance to soft errors in on-chip caches. However, write-through policy also incurs large energy overhead due to the increased accesses to caches at the lower level (e.g., L2 caches) during write operations. In this paper, we propose a new cache architecture referred to as *way-tagged cache* to improve the energy efficiency of write-through caches. By maintaining the way tags of L2 cache in the L1 cache during read operations, the proposed technique enables L2 cache to work in an equivalent direct-mapping manner during write hits, which account for the majority of L2 cache accesses. This leads to significant energy reduction without performance degradation. Furthermore, the idea of way tagging can be applied to existing low-power cache design techniques to further improve energy efficiency.

Keywords- Cache, low power, write-through policy, way tag.

I.INTRODUCTION

Microprocessors and systems-on-chip (SOCs) are now extensively used in a variety of applications ranging from smart phones to handheld computers, from entertainment systems to sophisticated automotive controllers, and from gaming devices to life-saving medical equipment. The advancements of semiconductor technology have boosted the rapid growth of very large scale integrated (VLSI) systems in our day-to-day life

In this paper, we propose a new cache architecture, referred to as way-tagged cache, to improve the energy efficiency of write-through cache systems with minimal area overhead and no performance degradation. Consider a two-level cache hierarchy, where the L1 data cache is write-through and the L2 cache is inclusive for high performance. It is observed that all the data residing in the L1 cache will have copies in the L2 cache. In addition, the locations of these copies in the L2 cache will not change until they are evicted from the L2 cache. Thus, we can attach a tag to each way in the L2 cache and send this tag information to the L1 cache when the data is loaded to the L1 cache. By doing so, for all the data in the L1 cache, we will know exactly the locations (i.e., ways) of their copies in the L2 cache. During the subsequent accesses when there is a write hit in the L1 cache (which also initiates a write access to the L2 cache under the write-through policy), We can access the L2

This results in an increase in the write accesses to the L2 cache and consequently more energy consumption. Virtual address is obtained from the CPU and the transition look aside buffer will speed up the

virtual to physical address translation. Data array store data from the CPU or lower memories.

Operations in the L1 cache				
	Read hit	Read miss	Write hit	Write miss
L2	No access	Set-associative	Direct-mapping	Set-associative

Table 1. Equivalent L2 access modes under different operations in the L1 cache

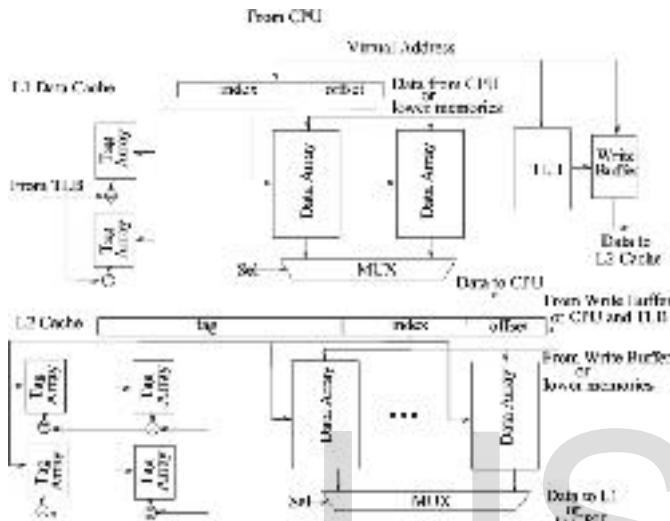


Fig. 2.1 Conventional two-level cache architecture.

Write buffers are commonly employed in write-through caches (and even in many write-back caches) to improve the performance. With a write buffer, the data to be written into the L1 cache is also sent to the write buffer.

B. Proposed way tagged cache

The L2 cache, the way tag of the data in the L2 cache is also sent to the L1 cache and stored in a new set of way-tag arrays[8]. These way tags provide the key information for the subsequent write accesses to the L2 cache. Figure 2.2 shows the system diagram of proposed way-tagged cache. We introduce several new components: way-tag arrays, way-tag buffer, way decoder, and way register, all shown

in the dotted line. The way tags of each cache line in the L2 cache are here in maintained in the way-tag arrays, located with the L1 data cache. Note that write buffers are commonly employed in write-through caches (and even in many write-back caches) to improve the performance. With a write buffer, the data to be written into the L1 cache is also sent to the write buffer. The operations stored in the write buffer are then sent to the L2 cache in sequence. This avoids write stalls when the processor waits for write operations to be completed in the L2 cache. In the proposed technique, we also need to send the way tags stored in the way-tag arrays to the L2 cache along with the operations in the write buffer. Thus, a small way-tag buffer is introduced to buffer the way tags read from the way-tag arrays. A way decoder is employed to decode way tags and generate the enable.

C. L2 access mode

In general, both write and read accesses in the L1 cache may need to access the L2 cache. These accesses lead to different operations in the proposed way-tagged cache, as summarized in Table 1. Under the write-through policy, all write operations of the L1 cache need to access the L2 cache. In the case of a write hit in the L1 cache, only one way in the L2 cache will be activated because the way tag information of the L2 cache is available, i.e., from the way-tag arrays we can obtain the L2 way of the accessed data. While for a write miss in the L1 cache, the requested data is not stored in the L1 cache.

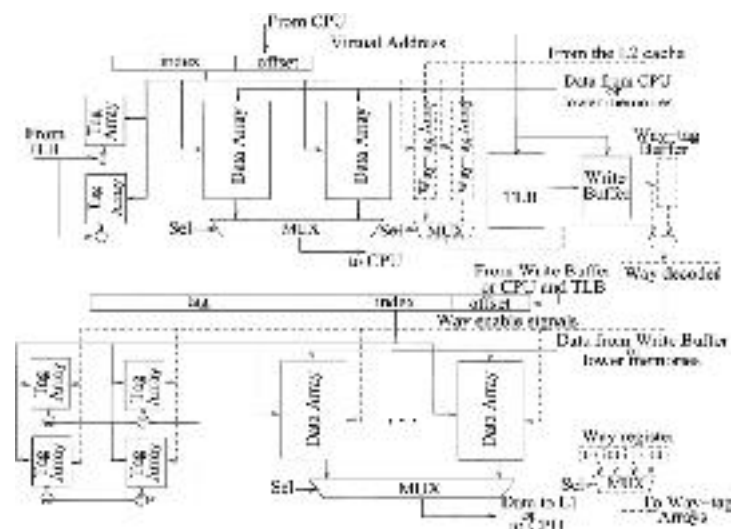


Fig. 2.2 Proposed way-tagged cache.

as a result, its corresponding L2 way information is not available in the way-tag arrays. Therefore, all ways in the L2 cache need to be activated simultaneously. Since write hit/miss is not known a priori. The way-tag arrays need to be accessed simultaneously with all L1 write operations in order to avoid performance degradation. Note that the way-tag arrays are very small and the involved energy overhead can be easily compensated. For L1 read operations, neither read hits nor misses need to access the way-tag arrays. This is because read hits do not need to access the L2 cache; while for read misses, the corresponding way tag information is not available in the way-tag arrays. As a result, all ways in the L2 cache are activated simultaneously under read misses.

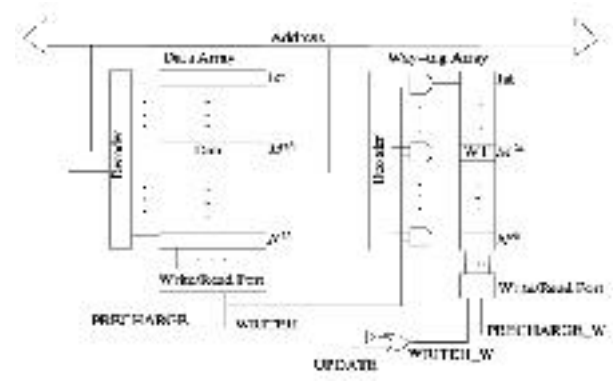


Fig 3.1 Way tag array

WRITEH	UPDATE	OPERATION
1	1	Write way-tagged arrays
1	0	Read way-tagged arrays
0	0	No access
0	1	No access

Table 2. Operation of way tag arrays

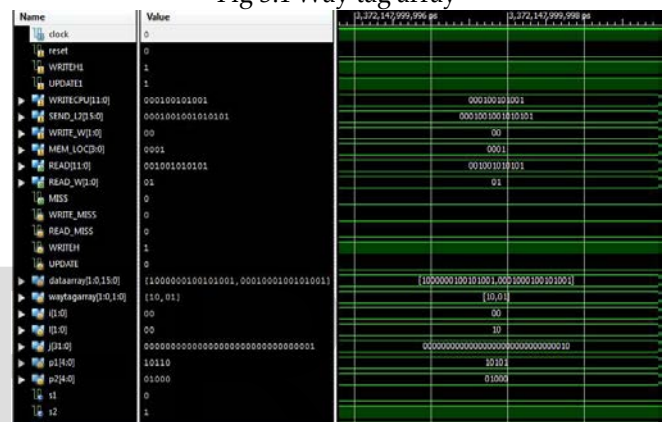


Fig 3.2 Simulation result of way tag array

III. IMPLEMENTATION OF WAY-TAGGED CACHE

The way-tag cache has been implemented using Verilog Language using software Xilinx ISE Design Suite 14.2 version and Isim simulator. Here, the implementation of additional components like way tag array, way decoder and way tag buffer has been carried out

A. Way tag arrays

Way tag array is implemented using table 2. Fig 3.2 shows the simulation results for way tag array. Clock, array of tag and data are given as input, corresponding data are return into L1 or else read the data from the L2 cache depending upon operation cache hit and cache miss occur in L1 cache.

B. Way tag buffer

Fig 3.4 shows the simulation results for way tag buffer. Clock, data in and empty are given as input, corresponding data are return into L1 or else read the data from the L2 cache depending upon operation cache hit and cache miss occur in L1 cache. If empty is zero, then data will be taken from memory array and give data out as output. Flip flop and Multiplexer is declared as component.

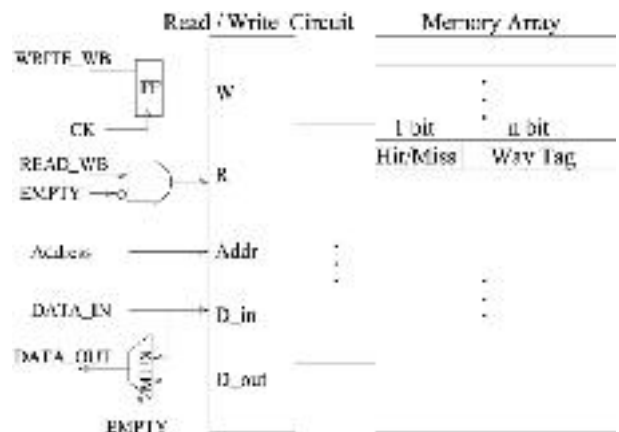


Fig. 3.3 Way-tag buffer.

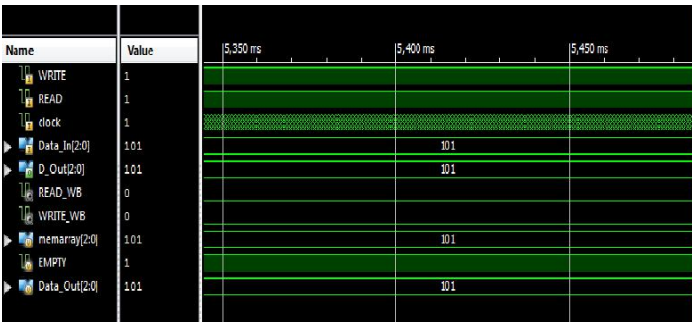


Fig 3.4 Simulation result of way tag buffer

C. Way decoder

Two signals, read and write miss, determine the operation mode of the way decoder. Signal read will be "1" when a read access is sent to the L2 cache. Signal write miss will be "1" if the write operation accessing the L2 cache is caused by a write miss in the L1 cache. If miss occurs in L2 cache, then it works as conventional set associative cache. Hence all ways are activated which increases the energy consumption. If hit occurs in L2 cache, then only the corresponding way will be activated. Hence it helps to increase the performance of the system. Read_miss, write_miss and waytag are given as input. 4 ways are taken as output.

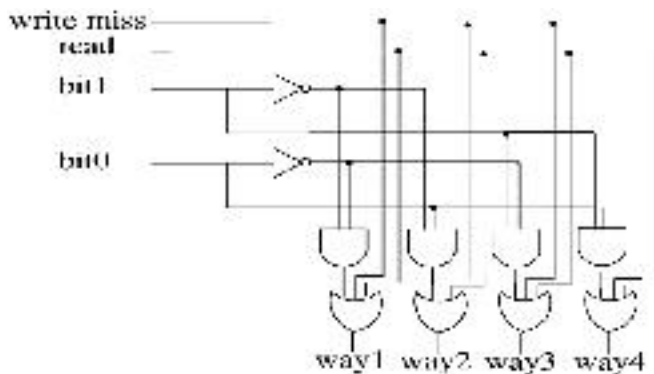


Fig 3.5 Implementation of the way decoder.

D. Way register

The way register provides way tags for the way-tag arrays. For a 4-way L2 cache, labels "00", "01", "10", and "11" are stored in the way register, each tagging one way in the L2 cache. When the L1 cache loads a data from the L2 cache, the corresponding way tag in the way register is sent to the way-tag arrays

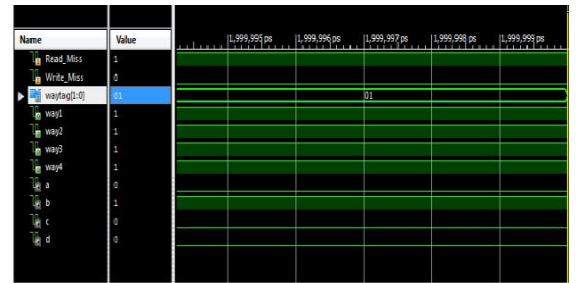


Fig 3.6 Simulation result of way decode during read miss "1"

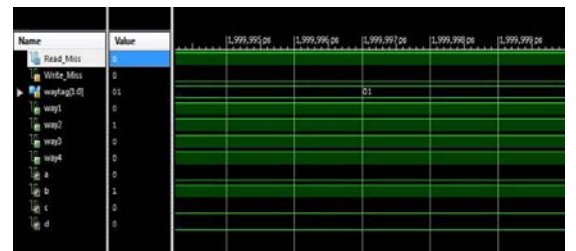


Fig 3.7 Simulation result of way decoder during read miss "0"

	Conventional L2 cache(nJ)	Way-tagged L2 cache(nJ)	Way-tag arrays(nJ)
Read access	35.50	35.50	0.002
Write access under write hit	35.53	4.58	0.002
Write access under write miss	35.53	35.53	0.002

Table 3. Energy consumption per read and write access of the conventional Set-associative L2 cache and the proposed L2 cache

VI. CONCLUSION

This report presents a new energy-efficient cache technique for high-performance microprocessors employing the write-through policy. The proposed technique attaches a tag to each way in the L2 cache. This way tag is sent to the way-tag arrays in the L1 cache when the data is loaded from the L2 cache to the L1 cache. Utilizing the way tags stored in the way-tag arrays, the L2

cache can be accessed as a direct-mapping cache during the subsequent write hits, thereby reducing cache energy consumption. Simulation results demonstrate significantly reduction in cache energy consumption with minimal area overhead and no performance degradation.

Furthermore, the idea of way tagging can be applied to many existing low-power cache techniques such as the phased access cache to further reduce cache energy consumption. Future work is being directed towards extending this technique to other levels of cache hierarchy and reducing the energy consumption of other cache operations.

REFERENCES

- [1] B. Malik, B. Moyer, and D. Cermak, "A low power unified cache architecture providing power and performance flexibility," in *Proc. Int. Symp. Low Power Electron. Design*, 2000, pp. 241–243.
- [2] K. Ghose and M. B.Kamble, "Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation," in *Proc. Int. Symp. Low Power Electron. Design*, 1999, pp. 70–75.
- [3] K. Inoue, T. Ishihara, and K. Murakami, "Way-predicting set-associative cache for high performance and low energy consumption," in *Proc. Int. Symp. Low Power Electron. Design*, 1999, pp. 273–275.
- [4] A.Ma, M. Zhang, and K.Asanovi, "Way memoization to reduce fetch energy in instruction caches," in *Proc. ISCA Workshop Complexity Effective Design*, 2001, pp. 1–9.
- [5] T. Ishihara and F. Fallah, "A way memoization technique for reducing power consumption of caches in application specific integrated processors," in *Proc. Design Autom. Test Euro. Conf.*, 2005, pp. 358–363.
- [6] R. Min, W. Jone, and Y. Hu, "Location cache: A low-power L2 cache system," in *Proc. Int. Symp. Low Power Electron. Design*, 2004, pp. 120–125.
- [7] T. N. Vijaykumar, "Reactive-associative caches," in *Proc. Int. Conf. Parallel Arch. Compiler Tech.*, 2011, p. 4961.
- [8] J. Dai and L. Wang, "Way-tagged cache: An energy efficient L2 cache architecture under write through policy," in *Proc. Int. Symp. Low Power Electron. Design*, 2009, pp. 159–164.
- [9] L. Hennessey and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 4th ed. New York: Elsevier Science & Technology Books, 2006.
- [10] B. Brock and M. Exerman, "Cache Latencies of the PowerPC MPC7451," Freescale Semiconductor, Austin, TX, 2006. [Online]. Available: cache.freescale.com
- [11] T. Lyon, E.Delano, C. McNairy, and D.Mulla, "Data cache design considerations for Itanium 2 processor," in *Proc. IEEE Int. Conf. Comput. Design*, 2002, pp. 356–362.
- [12] Standard Performance Evaluation Corporation, Gainesville, VA, "SPEC CPU2000," 2006. [Online]. Available: <http://www.spec.org/cpu>
- [13] "Pentium Pro Family Developer's Manual," Intel, Santa Clara, CA, 1996.
- [14] M. K. Qureshi, D. Thompson, and Y. N. Patt, "The V-way cache: Demand based associativity via global replacement," in *Proc. Int. Symp. Comput. Arch.*, 2005, pp. 544–555.
- [15] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [16] R.Min,W. Jone, and Y. Hu, "Phased tag cache: An efficient low power cache system", in *Proc. Int. Symp Circuits Syst.*, 2004, pp. 23–26