ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

PERFORMANCE ANALYSIS OF TAG BIT BASED LOW POWER CACHE MEMORY

Jeeva K[1], Kumar S[2]
[1]PG scholar, [2] Assistant professor, Department of Electronics and Communication Engineering
Bharathiyar Institute of Engineering for Women
jeeva.gtc@gmail.com, skumar_ece_tvl@yahoo.com

*Abstract:***In this paper, we propose a new cache scheme method, discussed to as early tag access (ETA) cache, to recover the energy effectiveness of data caches in embedded processors. The proposed methodimplements ETAs to define the purposeof memory instructions before the definite cache accesses. It, allows only the destination way to be retrieved if a hit ensues during the ETA. The proposed ETA cache can be designed under two operation modes to exploit the exchanges between energy efficiency and performance. It shown that our technique is very effective in reducing the number of ways accessed during cache accesses. This enables substantial energy reduction with negligible performance overheads.**

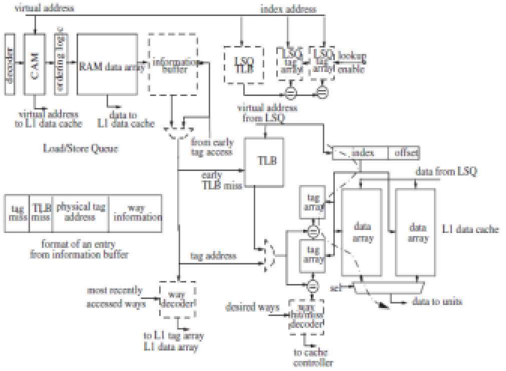
I. INTRODUCTION

Multi-level on-chip cache schemes have been widely accepted in high-performance microprocessors [1]–[3]. To retain data consistence through the memory grading, write-through and write-back policies are frequentlyengaged. Beneath the write-back policy, aimproved cache block is derivative back to its consistent lower level cache only when the block is almost to be interchanged. Butbeneath the write-through policy, all copies of a cache block are simplified immediately when the cache block is altered at the existing cache, even though the block might not be expelled. As a consequence, the write-through policy sustainssame data copies at all levels of the cache ladderthrough most of their life time of implementation. This article is significant as CMOS technology is climbed into the nanometer range, where soft errors have occurred as a mainconsistencyconcern in on-chip cache schemes. It has been described that single-event multi-bit distresses are receiving worse in on-chip memories [7]–[9]. Currently, this problem has been addressed at altered levels of the proposalconcept. At the architecture level, an effective solution is to keep data consistent among different levels of the memory hierarchy to inhibit the scheme from fall due to soft errors [10]–[12]. Promoted from instantaneous update, cache write-through policy is fundamentallylenient to soft errors since the data at all associated levels of the cache order are always kept constant. Due to this critique, many high-performance microprocessor schemes have executed the write-through policy [13]–[15]. While allowingenhancedlenience to soft errors, the write-through policy also acquiresenormous energy overhead. This is because below the write-through policy, caches at the lower level involvement more admittances during write operations. Study a two-level (i.e., Level-1 and Level-2) cache system for instance. If the L1 data cache outfits the write-back policy, a write hit in the L1 cache does not need to admittance the L2 cache. In dissimilarity, if the L1 cache is write-through, before both L1 and L2 caches essential to be retrieved for each write operation. Clearly, the write-through policy sustainsfurther write accesses in the L2 cache, which in chance increases the energy consumption of the cache scheme. Power dissipation is currentlymeasured as one of the critical problems in cache design. Studies have revealed that on-chip caches can consume around 50% of the entirecontrol in high-performance microprocessors [4]–[6].

In this paper, we suggestaninnovative cache method, denoted to as early tag access (ETA) cache, to recover the energy competence of L1 data caches. In a corporal tag and simulated index cache, a portion of the physical address is kept in the tag arrays though the alterationamong the virtual address and the physical address is achieved by the TLB. By retrieving tag arrays and TLB through the LSQ stage, the terminusmethods of maximum memory instructions can be resolutebeforeretrieving the L1 data cache. As a consequence, only single way in the L1 data cache desires to be retrieved for these instructions, thus reducing the energy consumption meaningfully. Note that the physical addresses created from the TLB at the LSQ stage can also be used for consequent cache admittances. Thus, for greatest memory instructions, the energy above of way purpose at the LSQ stage can be remunerated forby bouncing the TLB accesses through the cache access phase. For memory instructions whose terminus ways cannot be resolute at the LSQ stage, an improvedmethod of the ETA cache is planned to decrease the number of ways retrieved at the cache access period. Note that in several high-end processors, retrieving L2 tags is complete in parallel with the admittances to the L1 cache [2]. Our method is essentiallydissimilar as ETAs are achieved at the L1 cache.

II.PROPOSED ETA CACHE

In a conservative set-associative cache, all ways in the tag then data arrays are retrievedconcurrently. The entreated data, though, only exist in one way under a cache hit. The supplementary way entreessustainredundant energy consumption. In this unit, innovative cache architecture denoted to as ETA cache will be established. The ETA cache condenses the amount of redundant way admittances, thusdecreasing cache energy depletion. To put updissimilar energy and performance supplies in embedded processors, the ETA cache can be functionedbelow two dissimilar modes: the basic mode then the advanced mode.



LSQ Tag Arrays and LSQ TLB

To evade the data disputationby the L1 data cache, the LSQ tag arrays then LSQ TLB are executed as a facsimile ofthe tag arrays and TLB of the L1 data cache, individually. Here are two categories of operations in the LSQ tag arrays then LSQ TLB: lookup and update. Every time a memory address influences the LSQ, the LSQ tag arrays and LSQ TLB will be examined for the initial destination way. In instance of a hit, the early terminus way will be accessible; then, the instruction will reasonwhichever an initial tag miss forinform operations, the insides of LSQ tag arrays and LSQ TLB are efficient with the tag arrays and TLB of the L1 cache, consequently that they are same to avoid cache consistencydifficulties. The inform logic of LSQ tag arrays and LSQ TLB is the similarfor example that of the tag arrays then TLB of the L1 cache.

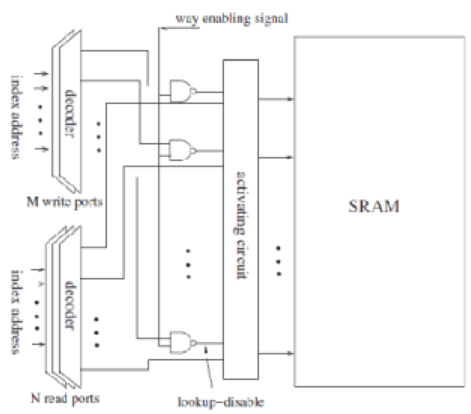
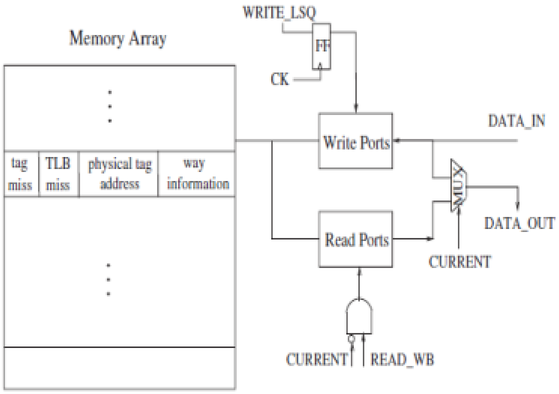


Fig.Architecture of LSQ Tag Array

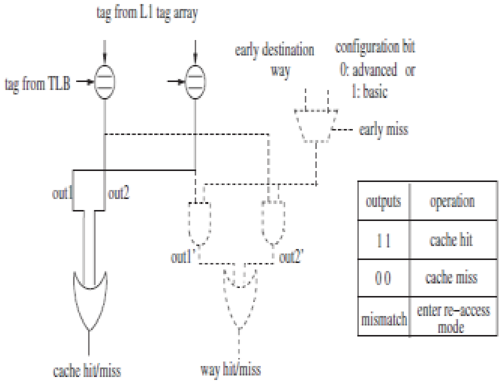
Fig. 11 shows the execution of the LSQ tag arrays;anywhere only one technique is shown as the other methods are the similar.Instructions can arrive the LSQ while the L1 data cache authorities'*M* substitutes to occur at the same time. Thus, there strengthis at most *N* lookup methods and *M* update proceduresup at the LSQ tag arrays and LSQ TLB at the same time. In mandate to execute these operations concurrently, the LSQ tag arrays and LSQ TLB require*N* read ports and *M* write ports. In the modelsin Section V, together*M* and *N* are selected to be two for the persistence of demonstration. Write/read clasheshappenwhilethe lookup and update operations aim the equal location of the LSQ tag arrays at the similar time. To address this concern, we restrict the lookup operation if an update operation is nowexecuted. This is accomplished by the control signal lookup-disable, which are created by the way enabling signalssince the cache controller for cache substitutes. Contemplate a two-way set-associative cache for instance. Accept that there is anauxiliary occurring by the way 1 of the L1 data cache. As a consequence, the way enabling signal is fixed to "1" and then directed to the NAND gates in way 1 of the LSQ tag selections. Uncertainty the write decoder outputs a "0," i.e., nope update operation on this access of the tag array, the lookup-disable sign will be set to "1" and the stimulating circuit will not chunk the lookup operation on this access. Then the lookup-disable signal will be "0," and the stimulating circuit will chunklikelylookup operations to evade write/read clashes.



Information Buffer

The information buffer consumesdistinct write and read ports to maintenance parallel write and read processes. The write processes of the information buffer constantlyjump one clock cycle advanced than the corresponding write processes in theLSQ. This is for the admittances to the LSQ, LSQ tag arrays, and LSQ TLB arisesconcurrently. Subsequently the wayinfo is accessiblenext the write processes in the LSQ, this info will be inscribed into the info buffer one clock cycle advanced than the equivalent write process in the LSQ.

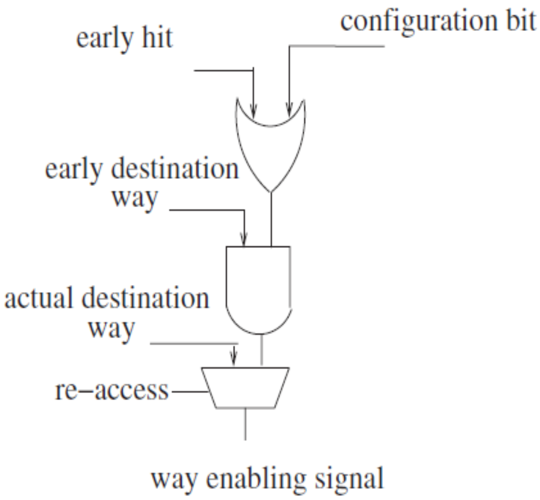
Way Hit/Miss Decoder



If a cache consistencyproblematic is perceived, an additional admittance to the L1 data cache is essential. Now, we announce a way hit/miss decoder to define whether the additional admittance is essential. Fig. 13

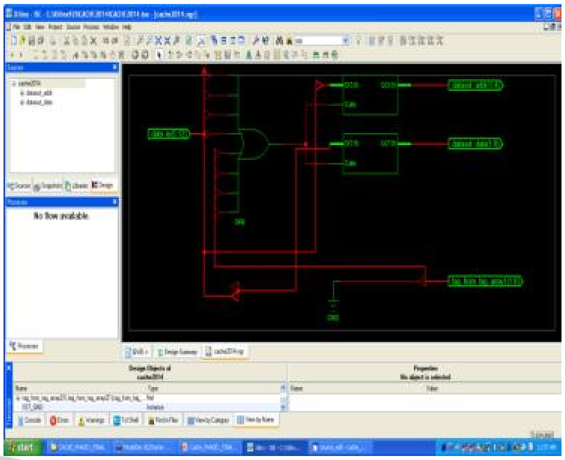
shows the execution of this decoder with dotted lines. A conservative cache hit/miss decoder is also exposed with solid lines. The configuration bit is recycled to fix the ETA cache for the simple mode or the innovative mode. As revealed in Fig. 13, if together the cache hit/miss and way hit/miss signals direct a hit (e.g., “1”), the cache admittance is reflected a hit.

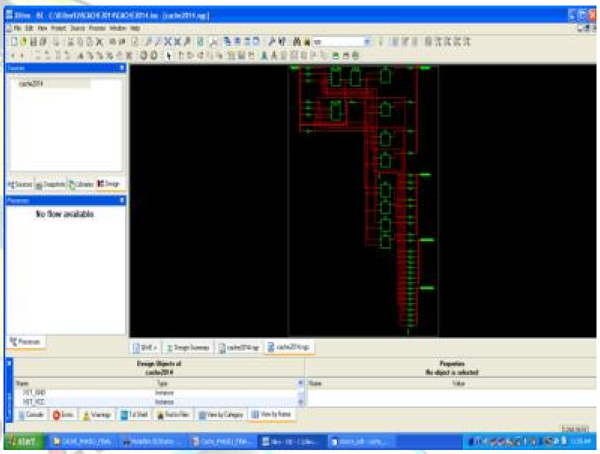
Way Decoder



In the future ETA cache, way enabling signals are desired to control the admittance to the ways in the data arrays. Fig. 14 illustrates the execution of the way decoder that creates these signals. When the instruction is related with an early hit (e.g., “1”), the data arrays essential to be accessed rendering to the initialterminus way. If the instruction capabilities an early tag miss or an initial TLB miss, the configuration bit shown in Fig. 14 defines which way in the data arrays 0 the L1 data cache desires to be retrieved. Specially, by set the configuration bit to “1,” the ETA cache will activatebelow the simple mode.

IV.Results and Discussion





Performance Evaluation Results:

Performance Parameters	Achieved Results
Power Consumption	203mW
Memory Usage	155996KB
Latency	5.077ns
Gate Counts	1728

Conclusion

This paper innovate a new energy-efficient cache method aimed at high-performance microprocessors retaining the write-through procedure. The future method assigns a tag to every method in the L2 collection. This method tag is directed to the way-tag arrangements in the L1 cache after the data is encumbered from the L2cache toward the L1 cache. Using the way labels kept in the way-tag collections, the L2 cache can be retrieved as a direct-map-ping cache during the following write triumphs, thus decreasing cache energy depletion. Model results establish knowingly decrease in cache energy depletion with insignificant area above and no concert ruin. Besides, the notion of method labeling can be functional to several present low-power reserve systems such as the phased admittance cache to further moderate cache energy depletion. Upcoming effort is being focused near lengthening this method to further positions of cache ladder and decreasing the energy depletion of other cache procedures.

References:

[1] *Intel XScale Microarchitecture*, Intel, Santa Clara, CA, USA, 2001.

[2] C. Zhang, F. Vahid, and W. Najjar“A highly-configurable cache architecture for embedded systems,” in *Proc. 30th Annu. Int. Symp. Comput.Archit.*, Jun. 2003, pp. 136–146.

[3] S. Segars, “Low power design techniques for microprocessors,” in *Proc. Int. Solid-State Circuits Conf. Tuts.*, Feb. 2001.

[4] S. Manne, A. Klauser, and D. Grunwald, “Pipline gating: Spculationconrol for energy reduction,” in *Proc. Int. Symp. Comput.Archit.*,
Jun.–Jul. 1998, pp. 132–141.

[5] M. Gowan, L. Biro, and D. Jackson, “Power considerations in the design of the alpha 21264 microprocessor,” in *Proc. Design Autom.Conf.*,
Jun. 1998, pp. 726–731.

[6] A. 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.

[7] T. Lyon, E. Delano, C. McNairy, and D. Mulla, “Data Cache Design Considerations for the Itanium Processor,” in *Proc. IEEE Int. Conf.*
Comput.Design, VLSI Comput.Process., 2002, pp. 356–362.

[8] D. Nicolaescu, A. Veidenbaum, and A. Nicolau, “Reducing power consumption for high-associativity data caches in embedded processors,”in *Proc. Design, Autom., Test Eur. Conf. Exhibit.*, Dec. 2003, pp. 1064–1068.

[9] C. Zhang, F. Vahid, Y. Jun, and W. Najjar, “A way-halting cache for low-energy high-performance systems,” in *Proc. Int. Symp.Low PowerElectron. Design*, Aug. 2004, pp. 126–131.

[10] J. Montanaro, R. T. Witek, K. Anne, A. J. Black, E. M. Cooper, D. W. Dobberpuhl, P. M. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, T. H. Lee, P. C. M. Lin, L. Madden, D. Murray, M. H. Pearce, S. Santhanam, K. J. Snyder, R. Stehpany, and S. C. Thierauf, “A 160-MHz 32-b 0.5- W CMOS RISC microprocessor,” *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, Nov. 1996.

[11] S. Santhanam, A. J. Baum, D. Bertucci, M. Braganza, K. Broch, T. Broch, J. Burnette, E. Chang, C. Kwong-Tak, D. Dobberpuhl,
P. Donahue, J. Grodstein, K. Insung, D. Murray, M. Pearce, A. Silveria, D. Souydalay, A. Spink, R. Stepanian, A. Varadharajan, V. R.
van Kaenel, and R. Wen, “A low-cost, 300-MHz, RISC CPU with attached media processor,” *IEEE J. Solid-State Circuits*, vol. 33, no. 11,pp. 1829–1838, Nov. 1998.

[12] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimizations,” in *Proc. Int. Symp. Comput. Archit.*, Jun. 2000, pp. 83–94.

[13] A. Hasegawa, I. Kawasaki, K. Yamada, S. Yoshioka, S. Kawasaki, and P. Biswas, “SH3: High code density, low power,” *IEEE Micro*, vol. 15, no. 6, pp. 11–19, Dec. 1995.

[14] J. Dai and L. Wang, “An energy-efficient L2 cache architecture using way tag information under write-through policy,” *IEEE Trans. Very*
Large Scale Integr. (VLSI) Syst., vol. 21, no. 1, pp. 102–112, Jan. 2013.