International Journal of Wireless Networks and Communications.
ISSN 0975-6507 Volume 7, Number 1 (2015), pp. 17-24

http://www.irphouse.com

# Low Power Cache Memory Architecture Using Bandwidth Scalable Controller

**A.Muthumanicckam[1], R.Sornalatha[2], L.Vijayprabakaran[3]**

*Assistant Professor [1,2], Dept. of ECE, Shanmuganathan Engineering College,*
*UG Scholar [3,] Dept. of ECE, Shanmuganathan Engineering College,Pudukkottai.*
*Email:muthumanickama@yahoo.com[1], rslece@gmail.com[2]*

**ABSTRACT**:

A new cache design technique referred to as Early Tag Access (ETA) cache, to improve the energy efficiency of data caches in embedded processors. The proposed technique performs ETAs to determine the destination ways of memory instructions before the actual cache accesses. Thus, it enables only the destination way to be accessed if a hit occurs during the ETA. The proposed ETA cache can be configured under two operation modes to exploit the tradeoffs between energy efficiency and performance. It is shown that this technique is very effective in reducing the number of ways accessed during cache accesses. This enables significant energy reduction with negligible performance overheads. Simulation results show that the proposed ETA cache consumes a power of 203mW on average in the L1 data cache and translation look aside buffer. Compared with the existing cache design techniques, the ETA cache is more effective in energy reduction while maintaining better performance.

Bandwidth Scalable Controller (BSC) is another technique used in this work to reduce power consumption. BSC is defined as a concept that allocates the Bandwidth depending on the input data size. Since it changes the bandwidth accordingly the power is reduced.

## I. INTRODUCTION

Multi-level on-chip cache systems are wide adopted in superior microprocessors.To keep information consistence throughout the memory hierarchy, write-through and write-back policies ar unremarkably used.Beneath the write-back policy, a changed cache block is derived back to its corresponding lower level cache block are updated at once when the cache block is changed at the present cache, even if the block may

not be evicted. As a result, the write-through policy maintains identical information copies in the slightest degree levels of the cache hierarchy throughout most of their life time of execution. This feature is vital as CMOS technology is scaled into the nanometer vary, wherever soft errors have emerged as a serious responsibility issue in on-chip cache systems. it's been according that single-event multi-bit upsets have gotten worse in on-chip reminiscences.

Currently, this drawback has been self-addressed at totally different levels of the planning abstraction. At the design level, an efficient answer is to stay knowledge consistent among totally different levels of the memory hierarchy to stop the system from collapse thanks to soft errors. Benefited from immediate update, cache write-through policy is inherently tolerant to soft errors as a result of the information the least bit connected levels of the cache hierarchy are perpetually unbroken consistent. Thanks to this feature, several superior micro chip styles have adopted the write-through policy. Whereas facultative higher tolerance to soft errors, the write-through policy additionally incurs giant energy overhead. This can be as a result of below the write-through policy, caches at the lower level expertise a lot of accesses throughout write operations. Take into account a two-level (i.e., Level-1 and Level-2) cache system as an example. If the L1 knowledge cache implements the write-back policy, a write hit within the L1 cache doesn't have to be compelled to access the L2 cache. In distinction, if the L1 cache is write-through, then each L1 and L2 caches have to be compelled to be accessed for each write operation. Obviously, the write-through policy incurs a lot of write accesses within the L2 cache, that successively will increase the energy consumption of the cache system. Power dissipation is currently thought-about in concert of the essential problems in cache style. Studies have shown that on-chip caches will consume concerning five hundredth of the entire power in superior microprocessors.

In this paper, we have a tendency to propose a replacement cache technique, cited as early tag access (ETA) cache, to boost the energy potency of L1 information caches. In an exceedingly physical tag and virtual index cache, a vicinity of the physical address is keep within the tag arrays whereas the conversion between the virtual address and therefore the physical address is performed by the TLB. By accessing tag arrays and TLB throughout the LSQ stage, the destination ways that of most memory directions will be determined before accessing the L1 information cache. As a result, just one method within the L1 information cache has to be accessed for these directions, thereby reducing the energy consumption considerably. Note that the physical addresses generated from the TLB at the LSQ stage may be used for future cache accesses. Therefore, for many memory directions, the energy overhead of method determination at the LSQ stage will be remunerated for by skipping the TLB accesses throughout the cache access stage. For memory directions whose destination ways that cannot be determined at the LSQ stage, Associate in Nursing increased mode of the ETA cache is projected to cut back the quantity of the way accessed at the cache access stage. Note that in several high-end processors, accessing L2 tags is completed in parallel with the accesses to the L1 cache. Our technique is essentially completely different as ETAs are unit performed at the L1 cache.

## II. PROPOSED ETA CACHE

In a typical set-associative cache, all ways that within the tag and knowledge arrays are accessed at the same time. The requested knowledge, however, solely resides in a way beneath a cache hit.The additional means accesses incur spare energy consumption. during this section, a replacement cache design cited as ETA cache are developed.The ETA cache reduces the amount of spare means accesses, thereby reducing cache energy consumption. To accommodate totally different energy and performance needs in embedded processors, the ETA cache is operated beneath 2 totally different modes: the essential mode and also the advanced mode.

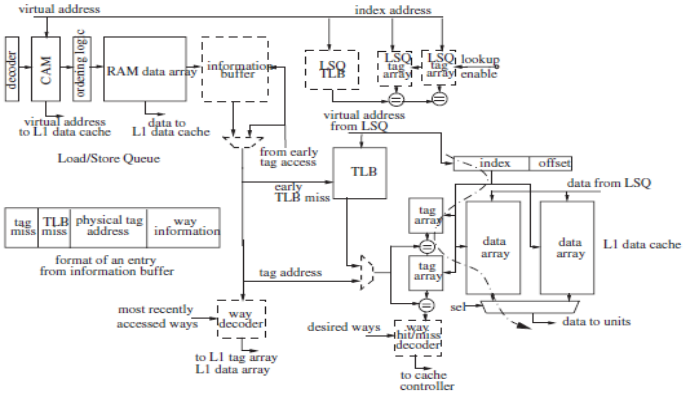



**FIG.1.Proposed ETA cache**

## III. LSQ TAG ARRAYS AND LSQ TLB

To avoid knowledge rivalry with the L1 data cache, the LSQ tag arrays and LSQ TLB are unit enforced as a duplicate of the tag arrays and TLB of the L1 knowledge cache, severally. There are two varieties of operations within the LSQ tag arrays and LSQ TLB: operation and update. Whenever a memory address reaches the LSQ, the LSQ tag arrays and LSQ TLB are explore for the first destination means. Just in case of a success, the first destination means are available; otherwise, the instruction can cause either associate degree early tag miss.For update operations, the contents of LSQ tag arrays and LSQ TLB are unit updated with the tag arrays and TLB of the L1 cache, in order that they avoid cache coherence issues. The update logic of LSQ tag arrays and LSQ TLB is that the same as that of the tag arrays and TLB of the L1 cache.

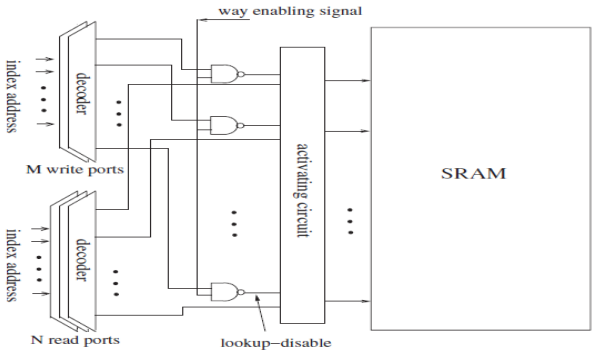



**FIG.2. Architecture of LSQ Tag Array**

The implementation of the LSQ tag arrays, wherever just one manner is shown because the different ways in which a similar think about that usually at the most N directions will enter the LSQ whereas the L1 knowledge cache permits M replacements to occur at a similar time. Therefore, there may well be at the most N lookup operations and M update operations occurring at the LSQ tag arrays and LSQ TLB at a similar time. So as to perform these operations at the same time, the LSQ tag arrays and LSQ TLB have N read ports and M write ports. Within the simulations each M and N are chosen to be two for the aim of demonstration. Write/read conflicts occur once the lookup and update operations target a similar location of the LSQ tag arrays at a similar time. To handle this issue, we have a tendency to disable the lookup operation if associate update operation is presently performed. This is often achieved by the management signal lookup-disable, that is generated by the manner sanctioning signals from the cache controller for cache replacements.Think about a two-way set-associative cache for instance. Assume that there's a replacement occurring at the manner one of the L1 knowledge cache. As a result, the manner sanctioning signal is ready to "1" then sent to the NAND gates within the manner one of the LSQ tag arrays. If the write decoder outputs a "0," i.e., no update operation on this entry of the tag array, the operation-disable signal are going to be set to "1" and also the activating circuit won't block the lookup operation on this entry. Otherwise the lookup-disable signal are going to be "0," and also the activating circuit can block potential lookup operations to avoid write/read conflicts.

## IV. INFORMATION BUFFER

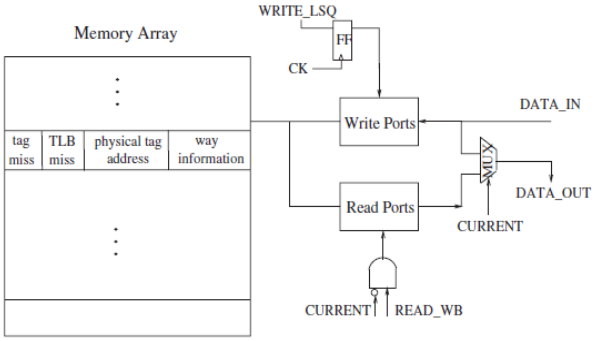



**FIG.3.Information Buffer**

The information buffer has separate write and read ports to support parallel write and read operations. The write operations of the information buffer always start one clock cycle later than the corresponding write operations in the LSQ. This is because the accesses to the LSQ, LSQ tag arrays, and LSQ TLB occur simultaneously. Since the way information is available after the write operations in the LSQ, this data will be written into the information buffer one clock cycle later than the corresponding write operation in the LSQ.

## V. WAY HIT/MISS DECODER

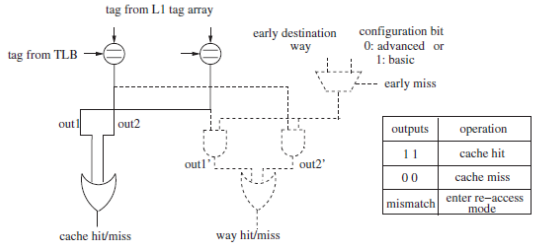



**FIG.4.Way Hit/Miss Decoder**

If a cache coherence drawback is detected, an extra access to the L1 knowledge cache is needed. Here, we tend to introduce the simplest way hit/miss decoder to work out whether or not the extra access is important.The implementation of this decoder is with dotted lines. A standard cache hit/miss decoder is additionally shown with solid lines. The configuration bit is employed to line the ETA cache for the fundamental mode or the advanced mode. As in Figure, if each the cache hit/miss and manner hit/miss signals indicate successful (e.g., "1"), the cache access is taken into account successful.

## VI. WAY DECODER

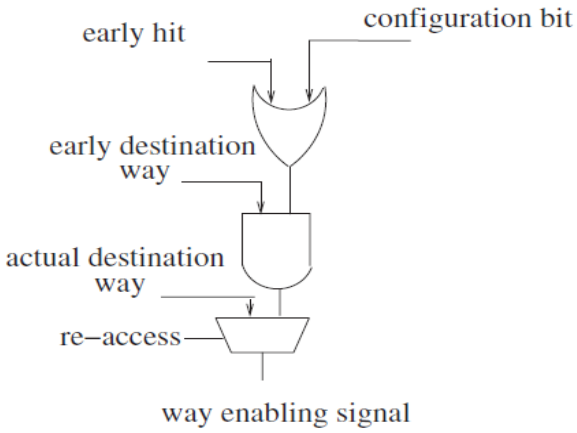
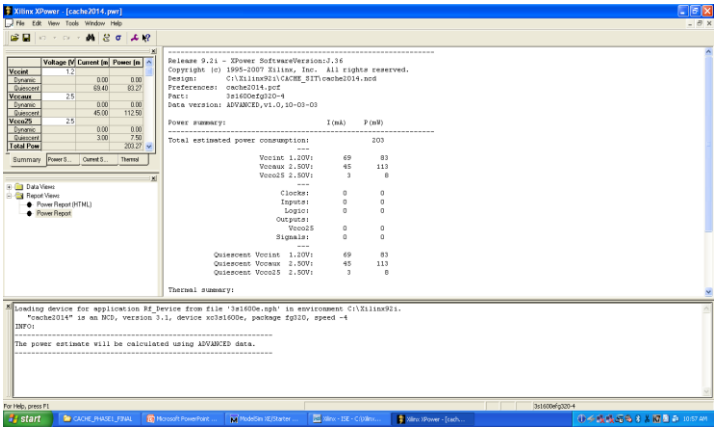



**FIG.5.Way Decoder**

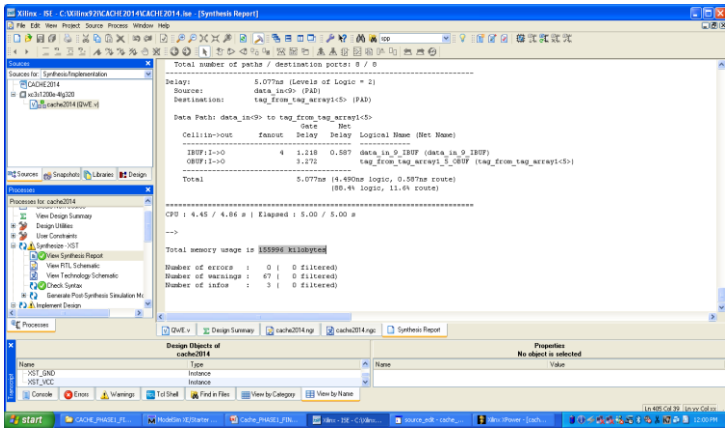
In the projected ETA cache, method enabling signals required to manage the access to the ways that within the information arrays. The implementation of the method decoder that generates these signals. Once the instruction is related to associate degree early hit (e.g., "1"), the information arrays have to be compelled to be accessed in line with the first destination method. If the instruction experiences associate degree early tag miss or associate degree early TLB miss, the configuration bit in figure determines that method within the information arrays the L1 information cache must be accessed. Specifically, by setting the configuration bit to "1," the ETA cache can operate below the fundamental mode.

## VII. RESULTS

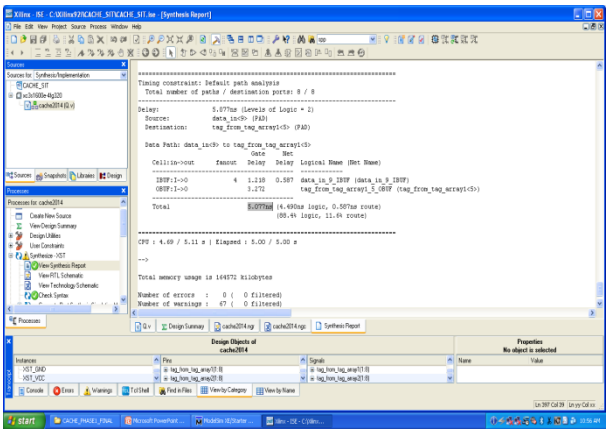
### POWER ANALYSIS REPORT



### MEMORY ANALYSIS REPORT



### LATENCY REPORT



**Performance Evaluation Results:**

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

## V. CONCLUSION

This paper presents a brand new energy-efficient cache technique for superior microprocessors using the write-through policy. The planned technique attaches a tag to every means within the L2 cache. this fashion tag is shipped to the way-tag arrays within the L1 cache once the information is loaded from the L2cache to the L1 cache. Utilizing the means tags hold on within the way-tag arrays, the L2 cache are often accessed as a direct-mapping cache throughout the next write hits, thereby reducing cache energy consumption. Simulation results demonstrate considerably reduction in cache energy consumption with stripped space overhead and no performance degradation. The performance parameters power, memory AND circuit counts square measure reduced to a extent.Furthermore, the concept of means tagging are often applied to several existing low-power cache techniques like the phased access cache to more scale back cache energy consumption. The power consumption is 203 mW and therefore the memory usage is 155996KB and therefore the gate counts great gross finally the latency is 5.077ns.Future work is being directed towards extending this method to different levels of cache hierarchy and reducing the energy consumption of different cache operations.

## 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: Spculation conrol 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 Power Electron. 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.