ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Early Tag Access for Reducing L1 Data Cache Energy for Memory Access

S.Manoj Kumar [1], P.SaravanaKumar [2], A.Sathish Kumar [3], S.Saravanan. [4]

P.G. Student, Department of ECE, MahaBarathi Engineering College, Chinnasalem, Tamil Nadu , India[1]

Assistant Professor, Department of ECE, MahaBarathi Engineering College, Chinnasalem, Tamil Nadu, India[2]

Assistant Professor, Department of ECE, MahaBarathi Engineering College, Chinnasalem, Tamil Nadu, India[3]

Professor & Head, Department of EEE, Muthyammal Engineering college, Rasipuram, Tamil Nadu, India[4]

ABSTRACT: Modern Multicore processors are employing large last-level caches, for example Intel's E7-8800 processor uses 24MB L3 cache. Further with each CMOS technology generation, leakage energy has been dramatically increasing and hence, leakage energy is expected to become a major source of energy dissipation, especially in last-level caches (LLCs). The conventional schemes of cache energy saving either aim at saving dynamic energy or are based on properties specific to first-level caches, and thus these schemes have limited utility for last-level caches. Further, several other techniques require offline profiling or per-application tuning and hence are not suitable for product systems. This project proposes a novel cache leakage energy saving schemes for single-core and multicore systems; desktop, QoS, real-time and server systems. Proposed here is a software-controlled, hardware-assisted techniques which use early tag access (ETA) to configure the cache to the most energy efficient configuration while keeping the performance loss bounded. This technique is compared with other techniques and ETA outperforms them in their energy efficiency. This research has important applications in improving energy-efficiency of higher-end embedded, desktop, server processors and multitasking systems.

I. INTRODUCTION

Reducing power consumption in cache memory is a critical problem for embedded processors that target low power applications. It was reported that on-chip caches could consume as much as 40% of the total chip power. Furthermore, large power dissipation could cause other issues, such as thermal effects and reliability degradation. This problem is compounded by the fact that data caches are usually performance critical. Therefore it is of great importance to reduce cache energy consumption while minimizing the impact on processor performance. Many cache design techniques have been proposed at different levels of the design abstract to exploit the tradeoffs between energy and performance. As caches are typically set-associative, most micro architectural techniques aim at reducing the number of tag and data arrays activated. During an access so that cache power dissipation can be reduced. Phased caches access tag arrays and data arrays in two different phases.

Energy consumption can be reduced greatly because at most only one data array corresponding to the matched tag, if any, is accessed. Due to the increase in access cycles, phased caches are usually applied in the lower level memory, such as L2 caches, whose performance is relatively less critical. For L2 caches under the write through policy, a way-tagging technique sends the L2 tag information to the L1 cache when the data is loaded from the L2 cache. During the subsequent accesses, the L2 cache can be operated in an equivalent direct-mapping manner, thereby improving energy efficiency without incurring performance degradation. To reduce the energy consumption of L1 caches,way-predicting techniques make a prediction on the tag and data arrays that the desired data might be located.

If the prediction is correct, only one way is accessed to complete the operation; otherwise, all ways are accessed to search for the desired data. Mispredictions lead to cache re-accesses, which introduce a performance penalty. Another cache design technique, referred to as way-halting cache stores some lower order bits of each tag in tag arrays into a fully associative cache and compares them against the corresponding bits in the coming memory address in parallel with set index decoding.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

During the subsequent accesses, the L2 cache can be operated in an equivalent direct-mapping manner, thereby improving energy efficiency without incurring performance degradation. To reduce the energy consumption of L1 caches, way-predicting techniques make a prediction on the tag and data arrays that the desired data might be located. If the prediction is correct, only one way is accessed to complete the operation; otherwise, all ways are accessed to search for the desired data. Mispredictions lead to cache re-accesses, which introduce a performance penalty.

To Overcome the L2 Cache Memory Proposed new L1 Cache memory for bringing the Better performance and efficiency improvement in energy cache. It propose a new cache technique, referred to as early tag access (ETA) cache, to improve the energy efficiency of L1 data caches. In a physical tag and virtual index cache, a part of the physical address is stored in the tag arrays while the conversion between the virtual address and the physical address is performed by the TLB. By accessing tag arrays and TLB during the LSQ stage, the destination ways of most memory instructions can be determined before accessing the L1 data cache. As a result, only one way in the L1 data cache needs to be accessed for these instructions, thereby reducing the energy consumption significantly. Note that the physical addresses generated from the TLB at the LSQ stage can also be used for subsequent cache accesses.

II. RELATED WORK

Power Considerations in the Design of the Alpha 21264 Microprocessor Power dissipation is rapidly becoming a limiting factor in high performance microprocessor design due to ever increasing device counts and clock rates. The 21264 is a third generation Alpha microprocessor implementation, containing 15.2 million transistors and operating at 600 MHz. This paper describes some of the techniques the Alpha design team utilized to help manage power dissipation. In addition, the electrical design of the power, ground, and clock networks is presented.

Low Power Unified Cache Architecture Providing Power and Performance Flexibility The M-CORE M3 architecture was developed specifically for these embedded applications. To address the growing need for longer battery life and higher performance, an 8-Kbyte, 4-way set-associative, unified (instruction and data) cache with programmable features was added to the M3 core. These features allow the architecture to be optimized based on the application's requirements. It focus on the features of the M340 cache sub-system and illustrate the effect on power and performance through benchmark analysis and actual silicon measurements. The M-CORE M340 processor contains an 8-Kbyte, 4-way set-associative, unified L1 cache with 16-byte line size, a M3 processor core, and a Memory Management Unit (MMU). The M340 cache sub-system supports programmable modes of operation to the varying embedded application environments in order to improve overall cache efficiency. These modes are controlled via a Cache Control Register(CACR) which allow certain features of the cache to be enabled/disabled for power and performance tuning.

A low-cost, 300-MHz, RISC CPU with attached media processor describes the 300-MHz StrongARM 1500 microprocessor, which is capable of more than two billion 16-b operations per second. Starting with the original StrongARM 110 design, an attached media processor (AMP) has been integrated along with a synchronous DRAM memory controller and separate I/O bus. In addition, several enhancements have been made to the CPU and cache subsystem, and the chip has been shrunk from a 0.35-to a 0.28-µm technology. The chip includes 3.3 million transistors and measures 60 mm^2. It dissipates less than 3 W at 300 MHz at 2.0-V internal, 3.3-V I/O. The chip supports dynamic clock frequency switching for reduced operating power during low performance demands. There are 333 separately conditioned clocks on the chip, For battery-powered applications, V_{dd} can be reduced to achieve <0.5 W operation at 150 MHz. The chip is pseudostatic and can support clock stop and quiescent supply current testing. It implements the ARM V4 instruction set .

An Energy-Efficient L2 Cache Architecture Using Way Tag Information Under Write-Through Policy 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. Simulation results on the SPEC CPU2000 benchmarks demonstrate that the proposed technique achieves 65.4% energy savings in L2 caches on average with only 0.02% area overhead and no performance degradation. Similar results are also obtained under different L1 and L2

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

cache configurations .Simulation results demonstrate significantly minimum energy reduction in cache energy consumption with increasing hardware complexity.

Next Cache Line and Set Prediction Accurate instruction fetch and branch prediction is increasingly important on today's wide-issue architectures. Fetch prediction is the process of determining the next instruction to request from thememory subsystem. Branch prediction is the process of predicting the likely out-come of branch instructions. Several researchers have proposed very effective fetch and branch prediction mechanisms including branch target buffers (BTB) that store the target addresses of taken branches. An alternative approach fetches the instruction following a branch by using an index into the cache instead of a branch target address. It call such an index a next cache line and set (NLS) predictor. ANLS predictor is a pointer into the instruction cache, indicating the target instruction of a branch. It examine the use of NLS predictors for efficient and accurate fetch and branch prediction. Previous studies associated each NLS predictor with a cache line and provided only one-bit conditional branch predictors. Our study examines the use of NLS predictors with highly accurate two-level correlated conditional branch architectures. Examine the performance of decoupling the NLS predictors from the cache line and storing them in a separate tag-less memory buffer.

A Way Memoization Technique for Reducing Power Consumption of Caches in Application Specific Integrated Processors presents a technique for eliminating redundant cache-tag and cache-way accesses to reduce power consumption. The basic idea is to keep a small number of Most Recently Used (MRU) addresses in a Memory Address Buffer (MAB) and to omit redundant tag and way accesses when there is a MAB-hit. Since the approach keeps only tag and set-index values in the MAB, the energy and area overheads are relatively small even for a MAB with a large number of entries. The approach does not sacrifice the performance. In other words, neither the cycle time nor the number of executed cycles increases. The proposed technique has been applied to Fujitsu VLIW processor (FR-V) and its power saving has been estimated using NanoSim. Experiments for 32kB 2-way set associative caches show the power consumption of I-cache and D-cache can be reduced by 40% and 50%, respectively.

III. PROPOSED SYSTEM

To Overcome the L2 Cache Memory Proposed new L1 Cache memory for bringing the Better performance and efficiency improvement in energy cache. It propose a new cache technique, referred to as early tag access (ETA) cache, to improve the energy efficiency of L1 data caches. In a physical tag and virtual index cache a part of the physical address is stored in the tag arrays while the conversion between the virtual address and the physical address is performed by the TLB. By accessing tag arrays and TLB during the LSQ stage the destination ways of most memory instructions can be determined before accessing the L1 data cache. As a result, only one way in the L1 data cache needs to be accessed for these instructions, thereby reducing the energy consumption significantly.

The physical addresses generated from the TLB at the LSQ stage can also be used for subsequent cache accesses. Therefore, for most memory instructions, the energy overhead of way determination at the LSQ stage can be compensated for by skipping the TLB accesses during the cache access stage. For memory instructions whose destination ways cannot be determined at the LSQ stage, an enhanced mode of the ETA cache is proposed to reduce the number of ways accessed at the cache access stage. Note that in many high-end processors, accessing L2 tags is done in parallel with the accesses to the L1 cache. Our technique is fundamentally different as ETAs are performed at the L1 cache.

PROPOSED ETA CACHE:

In a conventional set-associative cache all ways in the tag and data arrays are accessed simultaneously. The requested data, however, only resides in one way under a cache hit. The extra way accesses incur unnecessary energy consumption. In this section, a new cache architecture referred to as ETA cache will be developed. The ETA cache reduces the number of unnecessary way accesses, thereby reducing cache energy consumption. To accommodate different energy and performance requirements in embedded processors, the ETA cache can be operated under two different modes: the basic mode and the advanced mode.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

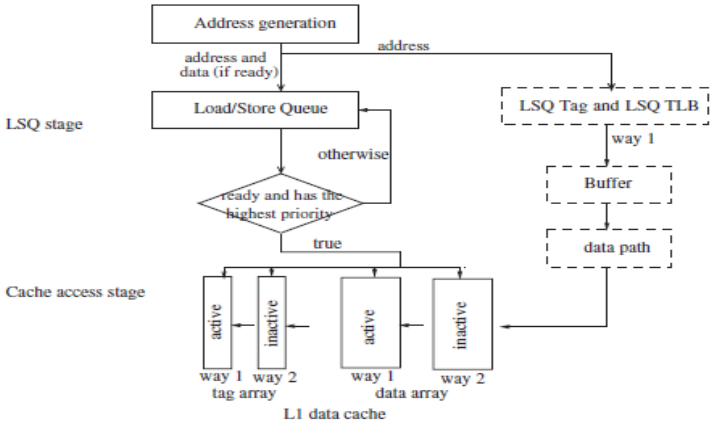


Fig. 3. Operation of a load/store instruction between LSQ and L1 data cache under the proposed ETA cache.

Basic Mode

It is possible to perform an access to the tag arrays at the LSQ stage due to the availability of memory addresses. In the basic mode of the ETA cache, each time a memory instruction is sent into the LSQ, an access to a new set of tag arrays and TLB (referred to as LSQ tag arrays and LSQ TLB, see Section IV-A) is performed. This new set of LSQ tag arrays and LSQ TLB are implemented as a copy of the tag arrays and TLB of the L1 data cache, respectively, to avoid the data contention with the L1 data cache. If there is a hit during the LSQ lookup operation, the matched way in the LSQ tag arrays will be used as the destination way of this instruction when it accesses the L1 data cache subsequently.

If this destination way is correct, only one way in the L1 data cache needs to be activated and thus enables energy savings. On the other hand, if a miss occurs during the lookup operation in the LSQ tag arrays or in the LSQ TLB, the L1 data cache will be accessed in a conventional manner, i.e., all ways in the tag arrays and data arrays of the L1 data cache will be activated. That the two sets of tag arrays and TLB are accessed at two different stages: LSQ stage and cache access stage

TABLE I
DEFINITIONS OF CACHE ACCESS IN THIS PAPER

	LSQ stage	Cache Access Stage
Tag array access	ETA	actual tag access
TLB access	early TLB access	actual TLB access
Tag hit/miss	early tag hit/miss	cache hit/miss
TLB hit/miss	early TLB hit/miss	TLB hit/miss
Destination way	early destination way	actual destination way

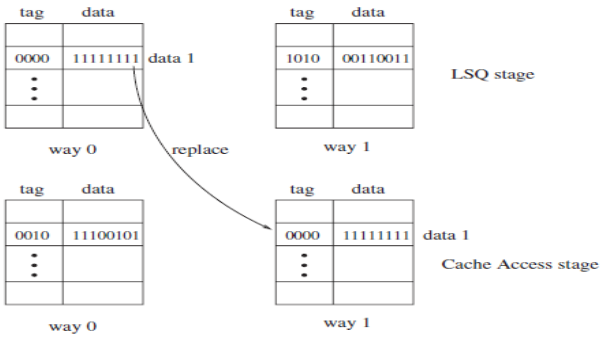


Fig. 4. Illustration of cache coherence problem related to early destination way information.

It is possible that the early destination way of a memory instruction determined at the LSQ stage is not the same as the actual one determined at the cache access stage due to the cache misses that happen in between. This causes a cache

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

coherence problem if the memory instruction simply follows its early destination way, if any, to access the L1 data cache.

Assume that an instruction Inst1 loads data1 into register *R*1. At the LSQ stage, data1 is stored in the way0 of the L1 data cache. Thus, Inst1 has an early hit at the LSQ stage and its early destination way is determined as way0. When Inst1 accesses the L1 data cache at the cache access stage (which may happen several clock cycles later if some instructions in the LSQ have a higher priority than Inst1), data1 actually resides in the way1 of the L1 data cache due to a cache miss that happens between the LSQ stage and cache access stage of the instruction Inst1. In this case, a cache coherence problem will occur if Inst1 simply uses its early destination way to access the L1 data cache. To avoid this problem, the ETA cache in the basic mode requires the memory instruction to access all the ways in the actual tag arrays during the cache access stage. During the same time, the data arrays are also accessed in parallel using the early destination way only (as the actual destination way is not available yet) to reduce energy consumption while maintaining performance.

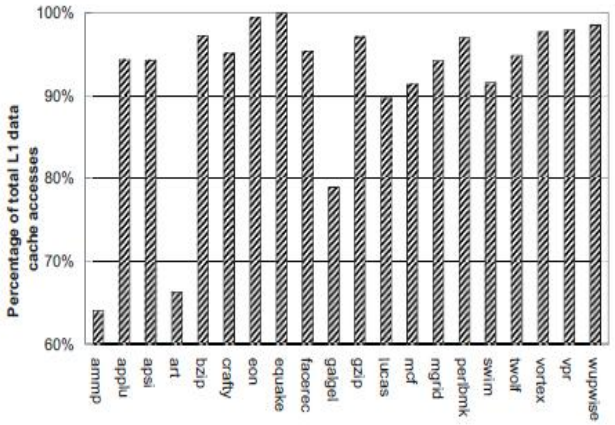


Fig. 5. Percentage of cache accesses whose destination ways can be determined through early tag accesses at the LSQ stage under SPEC2000 benchmarks.

The destination way obtained from the actual tag arrays is then compared with the early destination way to detect any cache coherence problem. If a cache coherence problem is detected, the accessed data from the data arrays is discarded and an additional access is performed. This occurs rarely, e.g., less than 1% of the total cache accesses. Therefore, only minor performance degradation is incurred, which is typically acceptable for embedded processors.

Note that another method to resolve the above cache coherence problem is to only access the predicted early destination way during the cache access stage. If the prediction is incorrect, simultaneous tag-data lookup will be performed to obtain the desired data from the data arrays. This solution saves energy in accessing tag arrays but consumes more energy during the data array access when the prediction is incorrect. Since data arrays consume much larger energy than tag arrays, However, if the re-access rate is extremely small, the second method may enable 1% additional energy savings due to the reduction in accessing the tag arrays.

Advanced Mode

There are some memory instructions whose early destination ways cannot be determined due to either early tag misses or early TLB misses. This could be significant for applications with a high miss rate (e.g., amp and art). As shown in Figs. 7 and 8, these early tag/TLB misses at the LSQ stage are usually associated with real misses at the cache access stage. In fact from observe that only a few memory instructions have early tag/TLB misses but real cache hits. This observation leads to the development of the advance mode of the ETA cache for early tag/TLB misses (i.e., early destination ways cannot be determined at the LSQ stage).

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

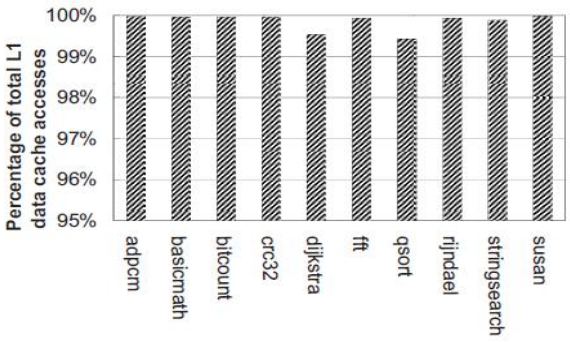


Fig. 6. Percentage of cache accesses whose destination ways can be determined through early tag accesses at the LSQ stage under MiBench benchmarks.

Specifically, when a memory instruction with either an early tag miss or an early TLB miss accesses the L1 data cache, the tag arrays of the L1 data cache are accessed first. In most cases this will be a real cache miss, and thus the L1 cache access is completed without the need to access the data arrays of the L1 data cache, thereby saving energy. In rare cases if a cache hit occurs, the actual destination way is obtained from the tag arrays and only this way is accessed in the data arrays at the next clock cycle. By applying this advanced mode, the energy consumption can be reduced because most of the memory instructions with early tag/TLB misses do not need to access the data arrays.

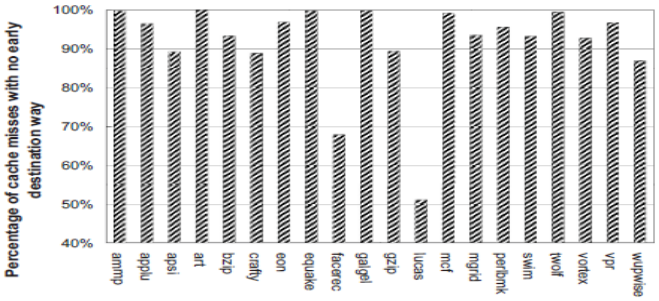


Fig. 7. Percentage of cache misses with no early destination ways available under SPEC2000 benchmarks.

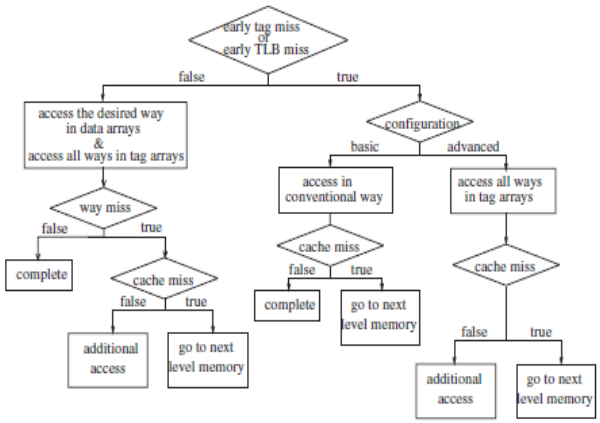


Fig. 9. Operation flow of the proposed ETA cache under two modes.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Unlike the phased cache where all data cache accesses are divided into two phases, the ETA cache only applies the advanced mode to memory instructions with early tag and/or early TLB misses. Some of these instructions have real cache hits, thereby causing an extra cycle in the cache access. Since the number of such cases (i.e., early tag/TLB misses but real cache hits) is very small, the performance overhead is negligible. Note that the advanced mode can be turned off if performance is truly critical.

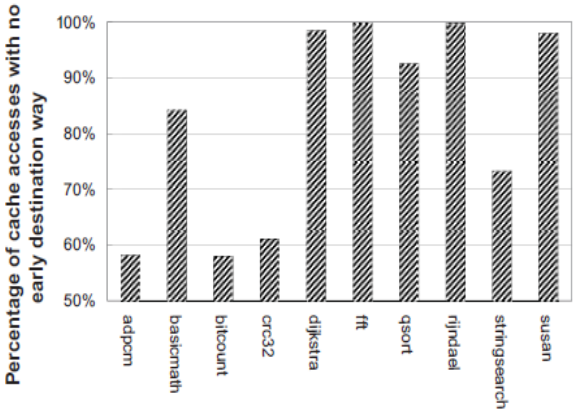


Fig. 8. Percentage of cache misses with no early destination ways available under MiBench benchmarks.

LSQ Tag Arrays and LSQ TLB

To avoid the data contention with the L1 data cache, the LSQ tag arrays and LSQ TLB are implemented as a copy of the tag arrays and TLB of the L1 data cache, respectively. There are two types of operations in the LSQ tag arrays and LSQ TLB: lookup and update. Each time a memory address reaches the LSQ, the LSQ tag arrays and LSQ TLB will be searched for the early destination way. In case of a hit, the early destination way will be available; otherwise, the instruction will cause either an early tag miss (if the access to the LSQ tag arrays encounters a miss) or an early TLB miss (if the address is not in the LSQ TLB).

Information Buffer

A load/store instruction might stay in the LSQ stage for more than one clock cycle. This requires an information buffer to hold the early destination way until the corresponding load/store instruction is issued to the L1 data cache. The implementation of the information buffer is shown. It has the same number of entries as the LSQ. In each entry, the tag miss and TLB miss bits ("1" for hit and "0" for miss) indicate the status of early tag and early TLB accesses, respectively. The early destination way is stored in the way information bits. The physical tag address is generated by searching the LSQ TLB. For a memory instruction with an early tag hit, the access to the actual TLB during the cache access stage can be avoided by sending this physical tag address to the data cache. Since the information buffer is small (32 entries with 20 bits per entry in a typical configuration), the induced energy overhead can be easily offset by the energy savings from the cache access stage.The information buffer has separate write and read ports to support parallel write and read operations.

Way Decoder and Way Hit/Miss Decoder

During a cache access, all ways in the cache tag arrays and one way in the data arrays (i.e., the early destination way) are accessed if the instruction has an early tag hit. As explained in Section III, activating all ways in the actual tag arrays is needed to detect the cache coherence problem. If a cache coherence problem is detected, an additional access to the L1 data cache is required. Here introduce a way hit/miss decoder to determine whether the additional access is

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

necessary. The implementation of this decoder with dotted lines. A conventional cache hit/miss decoder is also shown with solid lines.

The configuration bit is used to set the ETA cache for the basic mode or the advanced mode. if both the cache hit/miss and way hit/miss signals indicate a hit (e.g., "1"), the cache access is considered a hit. The access will be treated as a miss if both the cache hit/miss and way hit/miss signals indicate a miss. If the cache hit/miss signal indicates a hit while the way hit/miss signal indicates a miss, or if the early destination way does not match the actual destination way, a cache coherence problem is detected. The cache controller will trigger an additional access in the data arrays at the next cycle based on the actual destination way. This access is referred to as re-access.

Energy Model:

The following energy model is employed in our study:

$$E\text{tot} = N\text{tag\_hit} \times (E\text{tag\_hit,L1} + E\text{info}) + N\text{tag\_miss\_only} \times (E\text{tag\_miss\_only,L1} + E\text{info}) + N\text{tag\_TLB\_miss} \times (E\text{tag\_miss,L1} + E\text{TLB} + E\text{info}) + N\text{re−access} \times E\text{re−access,L1} + N\text{tot} \times (E\text{info} + E\text{LSQ\_tag} + E\text{LSQ\_TLB}) + E\text{others} \quad (1)$$

where *N*tag_hit, *N*tag_miss_only, and *N*tag_TLB_miss are the numbers of load/store instructions with early tag hit, early tag miss only, and both early tag and TLB misses, respectively, and *N*re−access is the number of cache re-accesses.

The energy consumptions per access related to these cases are denoted as *E*tag_hit, L1, *E*tag_miss_only,L1, *E*tag_TLB_miss,L1, and *E*re−access, respectively. *N*tot is the total number of load/store instructions issued to the LSQ. *E*info, *E*LSQ_tag, and *E*LSQ_TLB are the energy consumption per access of the information buffer, LSQ tag arrays, and LSQ TLB, respectively. Since the energy overheads from other components, such as the MUXes used in the ETA cache are very small, they are included in the *E*others.

The energy overhead due to the update of LSQ tag arrays and LSQ TLB is also included in the *E*others because update operations occur at a much lower rate than read operations (i.e., the miss rate is in general much lower than the hit rate). We employ CACTI to obtain the energy consumption per access under different operating modes using a 90-nm process. Since the proposed technique is technology-independent, these results are normalized by the energy consumption per access of the conventional L1 data cache for comparison. As shown in Table II, the "access" columns summarize the number of ways that needs to be activated in different units.

The energy consumptions of different components during different stages are also shown in Table II. For example, the LSQ tag array and LSQ TLB are only accessed during early tag access stage. Their total energy consumption per access is 21.4% of that of a conventional cache under the study.

Energy Efficiency

Since the proposed ETA cache does not affect the cache miss rate (i.e., no change in the replacement policy for the L1 data cache), the energy consumption related to cache misses,such as data replacement and off-chip memory accesses will be the same as that of the conventional cache. Therefore, we did not include these results in this paper. The Simple scalar toolset is employed to collect the statistics of different operations in these simulations. The results of replacements in the LSQ tag arrays and LSQ TLB are also collected. It was shown that the number of early tag and TLB accesses at the LSQ stage is more than that of the actual cache accesses. This is because some memory instructions can get the data directly from the LSQ by store forwarding, and some memory instructions might be flushed from the LSQ thus not issued to the L1 data cache. These results combined with the energy per access shown in Table II allow us to evaluate different cache configurations.

EVALUATION:

Evaluate the energy consumption, performance and design overhead of the proposed ETA cache. It experimental results for a processor with separate L1 instruction and data caches, both of which are dual ports 16 kB four-way set-

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

associative with the cache line size of 64 bytes. The TLB is four-way set-associative with 32 sets and page size of 4 kB, and the LSQ is a unified LSQ with 32 entries. It also evaluate the energy efficiency of the proposed technique under different cache configurations such as size and associativity. The Simple scalar toolset is employed.

TABLE II
OPERATION MODES OF THE ETA CACHE AND THE CORRESPONDING ENERGY CONSUMPTION (BASIC/ADVANCED) PER ACCESS NORMALIZED BY THE CONVENTIONAL L1 DATA CACHE

	ETA				Actual Cache Access									
	ETA		Conv.		Early Tag Hit		Early Tag Miss Only		Early TLB Miss		Re-Access		Conv.	
	Access	Energy	Access	Energy	Access	Energy	Access	Energy	Access	Energy	Access	Energy	Access	Energy
L1 tag array	0	0	0	0	full	0.01	full	0.01	full	0.01	0	0	full	0.01
L1 data array	0	0	0	0	1	0.257	full/0	0.99/0	full/0	0.99/0	1	0.257	full	0.99
TLB	0	0	0	0	0	0	0	0	full	0.18	0	0	full	0.18
LSQ TLB	full	0.2	0	0	0	0	0	0	0	0	0	0	0	0
LSQ tag array	full	0.012	0	0	0	0	0	0	0	0	0	0	0	0
Info buffer	1	0.002	0	0	1	0.002	1	0.002	1	0.002	0	0	0	0
Total energy		0.214		0		0.269		1.012/0.012		1.182/0.192		0.257		1.18

OPERATING MODES

ETA (The Basic Mode):

The energy reduction achieved by the proposed ETA cache under the basic mode over the conventional L1 data cache. It observe 34.1%–58.3% energy reduction across different SPEC CPU2000 benchmarks, with an average energy reduction of 52.8%.

ETA (The Advanced Mode):

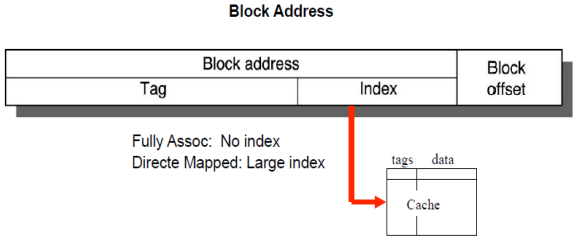
The energy reduction of the proposed ETA cache under the advanced mode for the SPEC CPU2000 benchmarks. The energy reduction ranges from 58.4% to 63.6% across different benchmarks with 59.6% on average, all higher than the corresponding measures in the basic mode. The advanced Mode is more energy efficient due to fewer ways accessed during the cache access stage. In particular, it is very effective for workloads whose memory instructions cannot find their early destination ways at the LSQ stage such as amp and art. These two benchmarks achieve 29.3% and 27.7% more energy reduction, respectively, in comparison with the basic mode of the ETA cache.

Cache Characteristics

- Cache organization
- Cache access
- Cache replacement
- Write policy

CACHE ACCESS:

- Tag on each block
  - No need to check index or block offset
- Increasing associativity shrinks index, expands tag



ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

VI. CONCLUSION

This paper presented a new energy-efficient cache design technique for low-power embedded processors. The proposed technique predicts the destination way of a memory instruction at the early LSQ stage. Thus, only one way needed to be accessed during the cache access stage if the prediction is correct, thereby reducing the energy consumption significantly. By applying the idea of phased access to the memory instructions whose early destination ways cannot be determined at the LSQ stage, the energy consumption can be further reduced with negligible performance degradation. Simulation results demonstrated the effectiveness of the proposed technique as well as the performance impact and design overhead. While our technique was demonstrated by a L1 data cache design, future work is being directed toward extending this technique to other levels of the cache hierarchy and to deal with multithreaded workloads.

VII. FUTURE WORK

In order to reduce acess latency new proposed **ELD3** Technique is added to reduce the energy level & Stall cycles in future.

Stall cycles- When the pipeline arichecture increases the execution time during tag & data acess.**ELD** and **ELA** techniques are used in **ELD3.**
ELA- Early load acess.
EDA- Early Data Acess.
ETA- Early Tag Acess.

Due to performance reasons, all ways in set-associative level-one (L1) data caches are accessed in parallel for load operations even though the requested data can only reside in one of the ways. Thus, a significant amount of energy is wasted when loads are performed. We propose a speculation technique that performs the tag comparison in parallel with the address calculation, leading to the access of only one way during the following cycle on successful speculations.
The technique incurs no execution time penalty, has an insignificant area overhead, and does not require any customized SRAM implementation. Assuming a 16kB 4-way set-associative L1 data cache implemented in a 65-nm process technology, our evaluation based on 20 different MiBench benchmarks shows that the proposed technique on average leads to a 24% data cache energy reduction.

REFERENCES

1. S. Mittal and Z. Zhang, "EnCache: Improving cacheenergyefficiency using a software-controlled profiling cache," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, May 2012, pp. 1–8.
2. X. Jiang, A. Mishra, L. Zhao, R. Iyer, Z. Fang, S. Srinivasan,S. Makineni, P. Brett, and C. R. Das, "ACCESS: Smart scheduling for asymmetric cache CMPs," in *Proc. IEEE 17th Int. Symp. HPCA*, Feb. 2011, pp. 527–538.
3. T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *Proc. Int. Conf. High Perform. Comput., Netw., SC*, 2011, pp. 1–12.
4. W. Wang, P. Mishra, and S. Ranka, "Dynamic cache reconfiguration and partitioning for energy optimization in real-time multi-core systems," in *Proc. DAC*, 2011, pp. 948–953.
5. R. Reddy and P. Petrov, "Cache partitioning for energy-efficient and interference-free embedded multitasking," *ACM Trans. Embedded Comput. Syst.*, vol. 9, no. 3, pp. 16:1–16:35, 2010.
6. N. A. Kurd, S. Bhamidipati, C. Mozak, J. L. Miller, T. M. Wilson, M. Nemani, and M. Chowdhury, "Westmere: A family of 32 nm IA processors," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2010.
7. D. K. Tam, R. Azimi, L. B. Soares, and M. Stumm, "RapidMRC: Approximating L2 miss rate curves on commodity systems for online optimizations," in *Proc. 14th Int. Conf. ASPLOS*, 2009, pp. 121–132.
8. S. Rodriguez and B. Jacob, "Energy/power breakdown of pipelined nanometer caches (90 nm/65 nm/45 nm/32 nm)," in *Proc. ISLPED*, 2006, pp. 25–30.
9. S. Borkar, "Design challenges of technology scaling," *IEEE Micro* vol. 19, no. 4, pp. 23–29, Jul. 1999.