EXPLOITING EARLY TAG ACCESS FOR REDUCING L1 DATA CACHE ENERGY IN EMBEDDED PROCESSORS

Kishore Kumar T [1], Azhagar Swamy P [2], Kathiresan M [3]

[1,2,3] U.G Student, Department of Electronics and Communication Engineering,
Raja College of Engineering and Technology, Madurai, Tamilnadu, (India)

ABSTRACT

We propose a new cache design technique, referred to as early tag access (ETA) cache, to improve the energy efficiency of data caches in embedded processors. ETAs to determine the destination ways of memory instructions before the actual cache accesses two operation modes to exploit the tradeoffs between energy efficiency and performance. This enables significant energy reduction with negligible performance overheads.

Keywords: ETA, LSQ. Energy Model

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.

II. RELATED WORK

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.

III.PROPOSED SYSTEM

To Overcome the L2 Cache Memory, We proposed new L1 Cache memory for bringing the better performance and efficiency improvement in energy cache. In this paper, we 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. 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.

3.1 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.

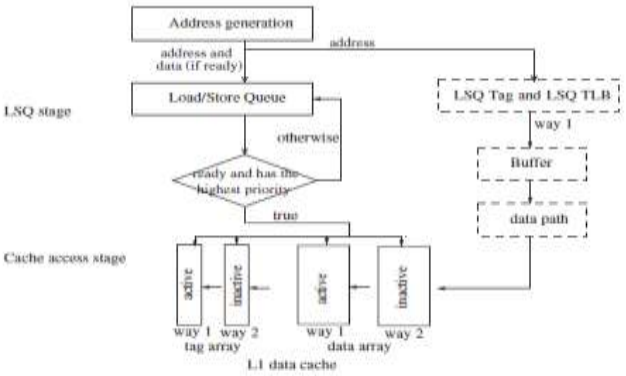

Fig. 3. Operation of a load/store instruction between LSQ and L1 data cache under the proposed ETA cache.

3.2 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 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. We can see that the two sets of tag arrays and TLB are accessed at two different stages: LSQ stage and cache access stage

3.3 Implementation

This section presents the VLSI implementation of the proposed ETA cache. Depicts a two-way set-associative L1 data cache for demonstration. The key components in the ETA cache, such as LSQ tag arrays, LSQ TLB, information buffer, way decoder, and way hit/miss decoder will be discussed in the following sections.

3.4 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). For update operations, the contents of LSQ tag arrays and LSQ TLB are updated with the tag arrays and TLB of the L1 cache, so that they are identical to avoid cache coherence problems. The update logic of LSQ tag arrays and LSQ TLB is the same as that of the tag arrays and TLB of the L1 cache.

Consider that generally at most N instructions can enter the LSQ while the L1 data cache allows M replacements to occur at the same time. Therefore, there might be at most N lookup operations and M update operations occurring at the LSQ tag arrays and LSQ TLB at the same time. In order to perform these operations simultaneously, the LSQ tag arrays and LSQ TLB have N read ports and M write ports. Write/read conflicts occur when the lookup and update operations target the same location of the LSQ tag arrays at the same time

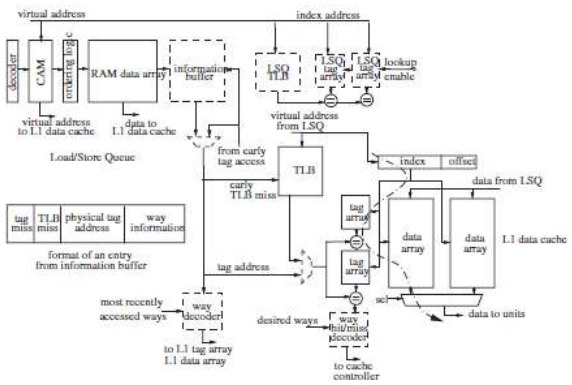

Fig. 10. Proposed ETA cache (blocks in dash line are new components and dotted line is the timing critical path).

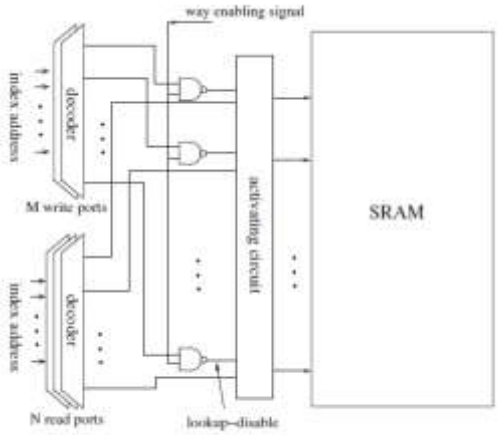

Fig. 11. Implementation of LSQ tag arrays (only one way is shown for simplicity).

Consider a two-way set-associative cache for example. Assume that there is a replacement occurring at the way 1 of the L1 data cache. As a result, the way enabling signal is set to "1" and then sent to the NAND gates in the way 1 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 lookup-disable signal will be set to "1" and the activating circuit will not block the lookup operation

on this entry. Otherwise the lookup-disable signal will be "0," and the activating circuit will block possible lookup operations to avoid write/read conflicts. The lookup operation, if any in this case, is considered as a miss. This miss might cause some performance degradations if it turns out to be a cache hit in the cache access stage. Fortunately, this rarely happens. We observed from simulations that less than 0.01% of the total LSQ accesses experienced this issue. Note that if the way enabling signal is "0," i.e., no update operation, the lookup operation will not be affected. This scheme is also used in the LSQ TLB. Since the activating circuit introduces no performance penalty, the enabling circuit increases the critical path of the LSQ tag arrays and LSQ TLB by the delay of an NAND gate.

3.5 Energy Model

The following energy model is employed in our study:

Etot = Ntag_hit × (Etag_hit,L1 + Einfo)+ Ntag_miss_only × (Etag_miss_only,L1 + Einfo)+ Ntag_TLB_miss × (Etag_miss,L1 + ETLB + Einfo) + Nre−access × Ere−access,L1 + Ntot ×(Einfo + ELSQ_tag + ELSQ_TLB) + Eothers (1)

where Ntag_hit, Ntag_miss_only, and Ntag_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 Nre−access is the number of cache re-accesses.

The energy consumptions per access related to these cases are denoted as Etag_hit, L1, Etag_miss_only,L1, Etag_TLB_miss,L1, and Ere−access, respectively. Ntot is the total number of load/store instructions issued to the LSQ. Einfo, ELSQ_tag, and ELSQ_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 Eothers.

The energy overhead due to the update of LSQ tag arrays and LSQ TLB is also included in the Eothers 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.

IV. OPERATING MODES

4.1 ETA (The Basic Mode)

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

The energy reduction for the MiBench benchmarks is shown in Fig. Due to the higher percentage of early tag determination in the MiBench benchmarks. The achieved energy reduction is higher than that of the SPEC CPU2000 benchmarks. Note that these results include the energy overheads of early tag and TLB accesses at the LSQ stage, such as the lookup and update operations.

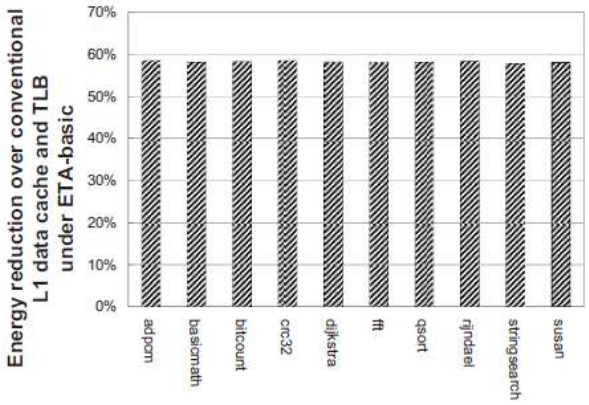

Fig. 16. Energy reduction of the ETA cache in the basic mode over the conventional L1 data cache and data TLB under MiBench benchmarks.

4.2 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.

IV. RESULTS AND DISCUSSION

The designs of the modules are coded using Verilog language and are compiled and simulated using modelsim software. The waveforms of the completed modules are described below.

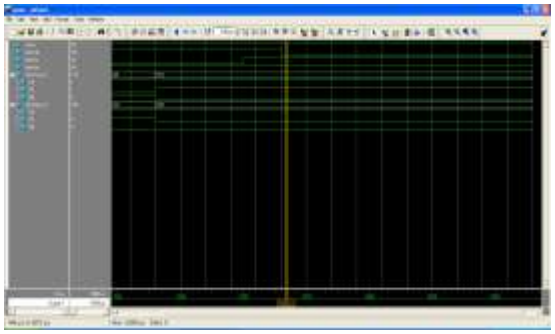

Figure 1. Simulation Result Of The Tag In

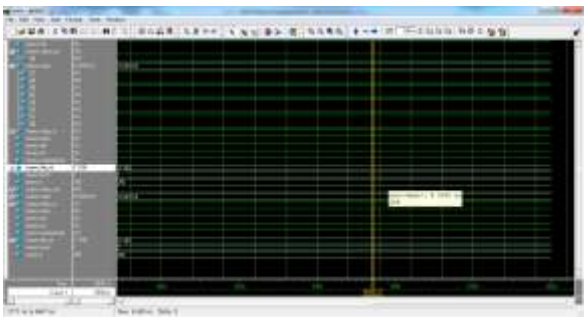

Figure2: Simulation For Tagged Out

The above figure explains about the output which is mentioned in data through policy for the given input data and clk and reset are enabled to form the MCD file. Multi clock Design it is to enable the data through access for the expression in the file ID and MCD in the given input.

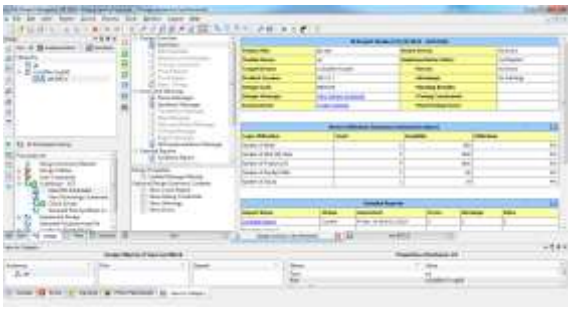

Figure3: Data Flow Diagram **Figure 4: Data Access Performance**

V. CONCLUSION AND FUTURE WORK

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.

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.

In order to reduce access 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.

REFERENCE

1. Afzal Malik, Bill Moyer, Dan CermakA(2008)- Low Power Unified Cache Architecture Providing Power and Performance Flexibility.
2. Jianwei Dai and Lei Wang (2013)-An Energy-Efficient L2 Cache Architecture Using Way Tag Information Under Write-Through Policy.
3. Santhanam.S (2008)-A low-cost, 300-MHz, RISC CPU with attached media processor
4. Ishihara, T ,- 2005 A Way Memoization Technique for Reducing Power Consumption of Caches in Application Specific Integrated Processors.
5. Calder, B, (1995)- Next Cache Line and Set Prediction.
6. Michael K. Gowan, Larry L. Biro, Daniel B. Jackson (2000)-Power Considerations in the Design of the Alpha 21264 Microprocessor.