Low-Power Sequential MRU Cache Based on Valid-Bit Pre-Decision

Hsin-Chuan Chen
Department of Electronic Engineering, St. John's University
robin@mail.sju.edu.tw

Abstract-*The conventional sequential MRU cache has longer access time because the MRU information must be fetched from the MRU table before accessing the memory banks of cache, and incurs larger power consumption due to multiple accesses of memory banks. In this paper, focusing on the sequential MRU cache with sub-block placement, we propose an MRU cache scheme that separates the valid bits from data memory and uses these valid bits to pre-decide reducing the unnecessary access number of memory banks. By this approach, the probability of the front hits is thus increased, and it significantly helps in improving the average access time and average energy dissipation of the sequential MRU cache without valid-bit pre-decision search especially for large associativity and small sub-block size.*

Keywords: Sequential MRU cache, Low power, Sub-block placement, Valid-bit pre-decision.

1. Introduction

The cache memory has played an important role to reduce the speed gap between processor and main memory. Because processors access the cache memory very frequently, and several studies have shown the cache memory consumes about 25%~50% of the total power in many microprocessor systems [1]. For some applications, such as embedded systems, the requirement of low power consumption seems to be more important than the obtainment of high performance [4]. Therefore, to reduce the overall power consumption of computer systems, low-power caches have become important for modern computer architecture.

In the past, the predictive sequential associative cache (PSA-cache) [2], uses a steering bit table (SBT) to dynamically determine which block is probed first when the cache is accessed. By providing prediction information, the miss rate of PSA-cache can be as low as that of 2-way set-associative caches, and the cycle time is similar to that of the direct-mapped caches. However, this cache scheme is not suitable for the implementation of set-associative caches with higher associativity. Therefore, the MRU (Most Recently Used) cache [6][7] that is similar to PSA-cache but uses an MRU table to determine the first probed block location in a set of the cache, while a set is referred to, the probability to find the correct block location in this set at the first time is very high [10]. Therefore, the MRU cache can be considered to develop the set-associative caches with higher associativity.

In this paper, based on a sequential MRU cache scheme with sub-block placement, the corresponding valid bits of sub-blocks can be used to decide which sub-blocks need to be probed in advance. The proposed sequential MRU cache without much hardware cost thus can effectively improve the average access time and average energy dissipation due to reducing many unnecessary probes of the tag and data memories during the search period.

2. Sequential MRU Cache

For implementing a low-cost cache, Kessler's scheme [5] uses a sequential search to find the desired block in a set according to the content of the MRU table. In this sequential cache (SMRU cache), both tag memory and data memory are single bank, and only one comparator is required.

2.1 Basic Concept

The SMRU cache uses the MRU table to store the block bits that represent the most recently used block number for each set, and determine the search order which is from most-recently-used (MRU) to least-recently-used (LRU). For example: in a 4-way set-associative MRU cache, if the MRU block list for one set is "01001110", that means the search order of the locations is 1, 0, 3 and 2. The block bits indicating the present desired block location are used to associate the set bits of main memory address to form an effective address as accessing the tag bank and data bank [11]. Due to using a true LRU replacement policy, the MRU block list for each set can be maintained by the cache system. Because of sequential search, the sequential MRU cache has a longer average access time than that of other cache schemes [3]. However, the sequential MRU cache with high associativity can be used as a low-cost level 2 cache in a two-level multiprocessor cache architectures to reduce memory interconnection traffic [5].

2.2 Energy and Access Time Models

According to the sequential operation of the SMRU cache, the following equation expresses the average access time of an *n*-way SMRU cache [5]:

$$T_{AS(SMRU)} = H_1 \times 2 + \left[\sum_{i=2}^{n} H_i \times (i+1) \right] + M \times (n+1+P) \quad (1)$$

where H_i is the ith hit rate of the cache, M is the miss

rate of the cache, and P is the miss penalty cycles depending on the block size. Kessler stated that this scheme might not be suitable for first-level caches. However, this scheme reduces a lot of hardware cost due to only requiring one comparator.

The memory size of the tag bank or data bank equals n tag-subarrays or n data-subarrays for an n-way sequential MRU cache; therefore, there are n tag-subarrays and n data-subarrays concurrently enabled for each cache access, which means n times of E_{tag} (energy of one tag-subarray) and E_{data} (energy of one data-subarray) need to be dissipated [1]. Usually, E_{data} is lager than E_{tag} because the data memory has a larger size than that of the tag memory in a cache, where E_{tag} = [(tag bits)/(block size× 8)]× E_{data} [10].

From energy view, the sequential MRU cache consumes large power compared with the other parallel cache schemes, and its average energy dissipation is given by [8]:

$$E_{C(SMRU)} = \left[\sum_{i=1}^{n} H_i \times (i \cdot E_{mem} + E_{MRU})\right] + M \times \left[(n+1) \cdot E_{mem} + 2 \cdot E_{MRU}\right] \quad (2)$$

where E_{mem} is the energy dissipated in the tag memory and data memory banks that equals $n\times(E_{tag}+E_{data})$. E_{MRU}, the energy dissipated in accessing the MRU table, equals [$\log_2 n$/(block size× 8)]× E_{data}.

3. Proposed Low-Power MRU Cache

To improve the conventional sequential MRU cache on power consumption, achieving a large number of front hits is very important. When a sequential MRU cache employs the sub-block placement to reduce its miss penalty, fortunately, the valid bits can be used to pre-eliminate the unnecessary search times for each cache access, such that it can make the original rear hits become more front hits. Based on this idea, a sequential MRU cache with valid-bit pre-decision search (called SMRU-V cache) [9] is proposed to not only reduce the average access time but also further improve its average energy dissipation.

3.1 Sub-Block Placement

Increasing block size will reduce the tag memory size for an on-chip cache design [12]; however, the large miss penalty is incurred due to large block size. Usually, the sub-block placement [13], which only refills a part of the entire block into the cache when the miss occurs, is an appropriate approach to reduce the miss penalty. In this cache scheme, each data block is divided into several sub-blocks, and each sub-block has a corresponding valid bit to indicate if this sub-block exists in the cache. Therefore, for a set-associative cache with sub-block placement, when the cache is accessed, other than tag checking of all ways, the corresponding valid bits of all ways must be checked together.

3.2 Valid-Bit Pre-Decision Search

In the conventional sequential MRU cache, the search order always starts from the MRU block to the LRU block one by one. Even though the present probed block does not exist (i.e. the valid bit = "0"), it still must complete checking the present block before probing the next block, which means this search is redundant. In the proposed SMRU-V cache, the search order is the same as that of the SMRU cache. However, the valid bits of the sub-blocks for different ways in one set can be used to decide which sub-blocks need to be examined during the search process. For an n-way SMRU-V cache, the valid-bit pre-decision search algorithm is shown in Fig. 1. Consequently, for a cache with small sub-block size, such a search algorithm can achieve more front hits and reduce many unnecessary search times with the valid bits being "0" on a cache hit, and it also can help in reducing the miss search times even when a cache miss occurs.

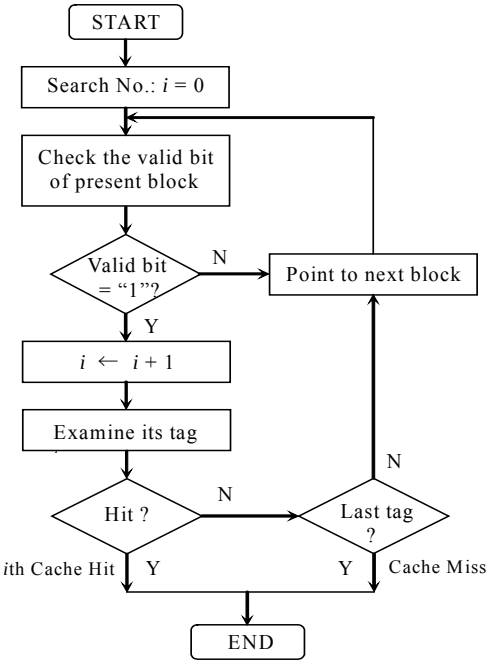

Figure 1. Valid-bit pre-decision search algorithm

3.3 Architecture and Operation

The architecture of the SMRU-V cache shown in Fig. 2 [9] modifies the data memory organization of the original sequential MRU cache, which the valid bits of all sub-blocks are also separated from the data memory bank, and they are organized as a single n-bit valid-bit bank and each bit represents one valid bit of the accessed sub-block for each way. The size of this single valid-bit bank is $(2^{s+w}/m)\times n$, where s denotes the set bits of the cache, w denotes the word offset bits of one block, and m is the sub-block size, and the bit order is from the MRU way (MSB bit) to the LRU way (LSB bit) for each set. When the cache is referred to, all memory banks including the MRU table and valid-bit bank are accessed concurrently, and almost no extra access time is needed. The content of the MRU table is the same as that of the SMRU cache, and the search order is from the MRU block location to the LRU block location. The valid bits stored in the valid-bit bank can be read to decide which block locations are needed to be probed when the MRU block

bits are taken from the MRU table. To implement the valid-bit pre-decision search, the control logic indeed requires the extra hardware components instead of the binary counter within the control logic of the conventional SMRU cache. Besides, the content of the MRU table and valid-bit bank can be maintained by the LRU replacement circuit after each cache access.

The operation of the SMRU-V cache is showed as follows [9]:

(1) While a set of the cache is referred to, the cache system fetches the MRU table and the valid-bit bank, and the control circuit takes the first MRU block bits that its corresponding valid bit is "1" from the MRU block list to form the address of the tag bank and data bank for the first MRU block.

(2) The cache system checks the tag of the first MRU block location selected by the first MRU block bits. Simultaneously, these bits are also used to speculatively select the data of the first MRU block location.

(3) If the first hit occurs, similar to the direct-mapped cache, the desired block data are directly read out from the data bank; however, two access cycles are required for the first probe.

(4) If the first hit does not occur, according to the valid bits with "1", the control circuit selects the next MRU block from the MRU block list in order, and checks the rest blocks in this set until all tags of this set are examined. If any hit is found again, the last selected MRU block bits are used to select the desired block data.

(5) When a miss occurs, the cache system will take more cycles to refill a new block from the lower-level memory to perform the replacement operation. Simultaneously, the status of the valid-bit bank will be maintained.

In our proposed cache architecture, due to the valid-bit pre-decision search and the support of LRU replacement algorithm, most of the increased front hits are first hits. Therefore, the first hit rate will be higher than that of the sequential MRU cache without valid-bit pre-decision search.

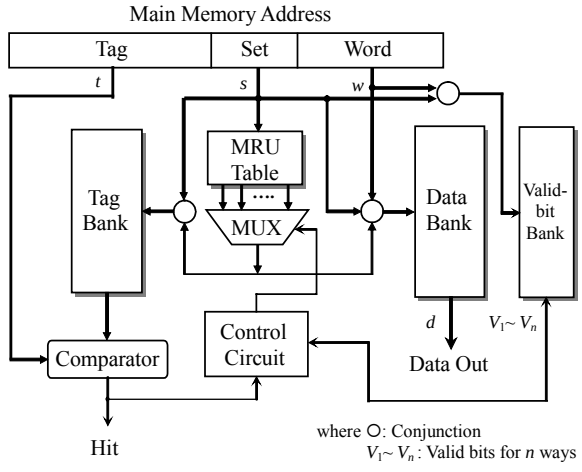

Figure 2. Architecture of SMRU-V cache

3.4 Energy and Performance Evaluation

In the sequential MRU caches, the average access time mainly depends on the first hit rate, the number distribution of hits in different times and hardware complexity. In the previous proposed SMRU-V cache [9], we know that the average access time ($T_{SMRU\text{-}V}$) is given by:

$$T_{AS(SMRU-V)} = H_1' \times 2 + \left[\sum_{i=2}^{n} H_i' \times (i+1)\right] + M \times (m+1+P) \quad (3)$$

In Eq. (3), the ith hit rate of the cache (H_i') and miss search times ($m \leqq n$) differ from the SMRU cache due to the valid-bit pre-decision search. Because the valid bits of all MRU blocks at the first hit are always equal to "1", and thus the first hits of the SMRU-V cache contains the original first hits and some new increased first hits that come from the original rear hits. Here, H_1' and H_i' are respectively given by:

$$H_1' = H_1 + \sum_{k=2}^{n} H_k \times V_{k1} \quad (4)$$

$$H_i' = H_i \times \prod_{j=1}^{i-1} (1 - V_{ij}) + \sum_{k=i+1}^{n} H_k \times V_{ki} \quad (5)$$

where V_{ij} (V_{ki}) denotes the probability to become the jth (ith) hit from the original ith (kth) hit. Consequently, the first hits and front hits are significantly increased by using valid-bit pre-decision search, and thus the SMRU-V cache can improve the SMRU cache on average access time.

In addition to reduction of access time, fortunately, due to eliminating the unnecessary search number by the valid-bit pre-decision, the proposed SMRU-V cache indeed effectively improves the SMRU cache on average energy dissipation. Therefore, the average energy dissipation of n-way SMRU-V cache can be expressed by:

$$E_{C(SMRU-V)} = \left[\sum_{i=1}^{n} H_i' \times (i \cdot E_{mem} + E_{MRU} + E_{VLD})\right] + M \times \left[(m+1) \cdot E_{mem} + 2 \cdot (E_{MRU} + E_{VLD})\right] \quad (6)$$

where E_{VLD} is the energy dissipated in accessing a the valid-bit bank, and it approaches $(n/sb)\times(E_{data}/8)$ for the sub-block size = sb. This overhead energy dissipation caused by the valid-bit bank almost can be neglected compared with the energy $E_{mem} = n\times(E_{tag}+E_{data})$. For example, a 4-way SMRU-V cache with 20 tag bits and sub-block size = 4 bytes, E_{VLD} only consumes $0.125E_{data}$ much less than $E_{men} = 4.3125E_{data}$.

In conclusion, when the associativity is high (i.e., n is large) and the sub-block size is small, the improvement in average access time and average energy dissipation of the SMRU-V cache is more significant than that of the SMRU cache.

4. Simulation Results

To evaluate and analyze the performance of the proposed cache architectures, we use a trace-driven cache simulator (Dinero) [14] to simulate the access behaviors of two sequential MRU caches including the SMRU cache and the SMRU-V cache. In our simulation, both cache architectures have the same cache size (= 32 KB),

block size (= 32 Bytes), and replacement policy (LRU). According to the operations of the different cache schemes, the average access time and average energy dissipation of the proposed cache can be evaluated by re-modeling Dinero to trace various trace programs [14] which some are belonged to SPEC benchmark suite such as SPICE, GCC, and XLISP.

4.1 Distribution of Hits

For the MRU caches using a sequential search, the number of hits in different times is also an important factor to influence the average access time and average energy dissipation. Table 1 shows the number of hits in different times for the SMRU cache and the SMRU-V by tracing various benchmarks, where both caches have the same associativity = 8 and sub-block size = 4 bytes. Because the SMRU-V cache using a valid-bit pre-decision search algorithm can reduce unnecessary search times, many original rear hits become the front hits, and we find that the first hits increase 5% on average. Consequently, its front hits are more than other sequential MRU caches. Therefore, the proposed SMRU-V cache can significantly improve the first hit rate of the SMRU caches especially for the cache with a large associativity and a small sub-block size.

Table 1. The number distribution of different hits

Programs		Number of different hits							
		1st	2nd	3rd	4th	5th	6th	7th	8th
SPICE	*SMRU*	398424	12590	3682	2215	2324	2136	1926	1452
	SMRU-V	414967	7229	1877	490	150	29	7	0
XLISP	*SMRU*	255783	22872	4486	1210	539	305	129	59
	SMRU-V	274806	9127	1107	268	63	11	1	0
FORA	*SMRU*	329782	18159	4859	2245	1359	972	766	564
	SMRU-V	349096	7827	1260	380	100	40	3	0
IVEX	*SMRU*	269420	14711	3686	1587	1053	658	486	335
	SMRU-V	283006	7520	886	327	149	40	8	0
GCC	*SMRU*	841767	43432	18792	11013	6799	4157	2628	1772
	SMRU-V	857172	40894	17393	8710	4010	1638	466	77
UE02	*SMRU*	271753	14110	4502	2049	1224	855	604	495
	SMRU-V	288451	6015	902	189	30	5	0	0

4.2 Improvement in Average Access Time

When the sub-block size starts decreasing, the average access time of the SMRU cache decreases at first due to the reduction of miss penalty, however; the miss rate increases and the first hit rate decreases as the sub-block size decreases, and thus the average access time increases again until the sub-block size decreases to 4 bytes. Contrary, the average access time of the proposed SMRU-V is always less than that of the SMRU cache as the sub-block size decreases. Fig. 3 indicates the improvement over the SMRU cache in average access time of the SMRU-V cache. Consequently, the improved rate in terms of the average access time IMR_{TAS} for the SMRU-V cache will increase as the associativity increases at the fixed sub-block size, or as the sub-block size decreases at the fixed associativity. When the associativity is larger than 4, the IMR_{TAS} has a significant increment. Especially for the associativity = 32 and the sub-block size = 4 bytes, the IMR_{TAS} can achieve up to about 40%.

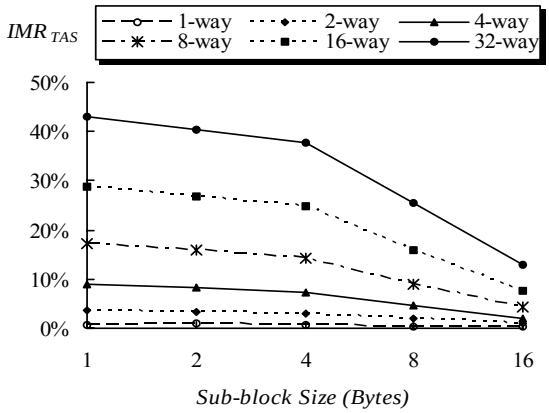

Figure 3. Improved rate in average access time

4.3 Improvement in Average Energy Dissipation

For the SMRU cache, because the miss rate increases and the first hit rate decreases as the sub-block size decreases, the energy dissipation of the SMRU cache will increase as the sub-block size decreases. However, when the sub-block size starts decreasing from 32 bytes, because valid-bit presence rates decrease, the average energy dissipation of the SMRU-V cache can be significantly reduced by eliminating many unnecessary search number compared with the conventional SMRU cache without valid-bit pre-decision, especially for the cache with a large associativity and a small sub-block size. From Fig. 4, we find that the SMRU-V cache has a significant energy improvement (about 40% at 32-way) over the SMRU cache when the sub-block size decreases down to 4 bytes, even at the often-used associativity = 4, the IMR_{EC} also can approach about 17%.

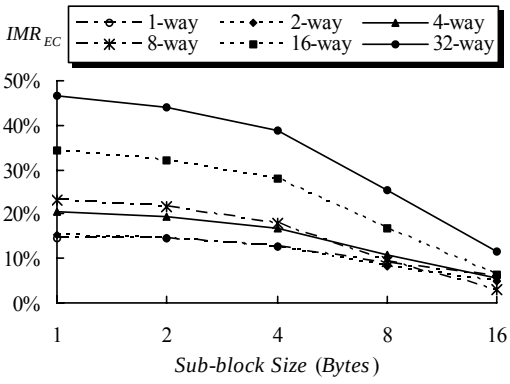

Figure 4. Improved rate in average energy dissipation

5. Conclusions

In this paper, using the valid-bit pre-decision search algorithm, the proposed SMRU-V cache can improve the average access time and average energy dissipation of the conventional sequential MRU cache with sub-block placement. Without adding much hardware in our proposed sequential MRU cache scheme, many

unnecessary search times are eliminated and more first hits are obtained by the valid-bit pre-decision. From simulation results, both improved rates in average access time and average energy dissipation can achieve about 40% on average at 32-way and the sub-block size = 4 bytes, even at the associativity = 4 and the sub-block size = 4 bytes, the proposed SMRU-V still has about 7% and 17% improvement for the average access time and average energy dissipation, respectively. Therefore, especially for large associativity and small sub-block size, the improvement in the average access time and average energy dissipation of the proposed SMRU-V cache will be more significant. Moreover, being a level two cache, the proposed SMRU-V cache not only achieves low power and high performance, but also it still maintains the benefit of low-cost implementation as that of the conventional sequential MRU cache.

References

[1] M. B Kamble and K. Ghose, "Analytical energy dissipation models for low power caches", *Proc. 1997 International Symposium on Low Power Electronics and Design*, pp. 143-148, Aug. 1997.

[2] B. Calder, D. Grunwald, and J. Emer, "Predictive sequential associative cache", *Proc. 2nd International Symposium on High Performance Computer Architecture*, pp. 244-253, Feb. 1996,.

[3] C. Zhang, X. Zhang, and Y. Yan, "Two fast and high-Associative cache schemes", *IEEE Micro*., vol. 17, pp. 40-49, 1997.

[4] C. Zhang, F. Vahid, and W. Najjar, "Energy benefits of a configurable line size cache for embedded systems," *Proc. International Symposium on VLSI Design*, pp. 136-146, Feb. 2003.

[5] R. Kessler, R. Jose, A. Lebeck and M. Hill, "Inexpensive implementations of set-associativity", *Proc. 16th Annual International Symposium on Computer Architecture*, pp. 131-139, May 1989.

[6] K. So and R. Rechtschaffen, "Cache operations by MRU change", *IEEE Transactions on Computers*, vol. 37, pp. 700-709, 1988.

[7] C. Wu, Y. Hsu, and Y. Liu, "A quantitative evaluation of cache types", *Proc. 26th Hawaii International Conference on System Sciences*, vol. 1, pp. 476-485, 1993.

[8] Z. Zhu and X. Zhang, "Access-mode predictions for low-power cache design", *IEEE Micro*, vol. 2, no. 2, pp. 58-71, Mar. 2002.

[9] H. C. Chen, "A high performance sequential MRU cache using valid-bit pre-decision search algorithm," *Proceedings of the International Computer Symposium 2006*, vol. 1, pp. 210-214, Dec. 2006.

[10] K. Inoue, T. Ishihara, and K. Murakami, "A high-performance and low-power cache architecture with speculative way-selection", *IEICE Transactions on Electron*, vol. E83-C, pp. 186-193, 2000.

[11] H. C. Chen, J. S. Chiang and Y. S. Lin, "A fast sequential MRU cache with competitive hardware cost", *2001 The 2nd International Conference on Parallel and Distributed Computing, Application and Technologies*, pp. 220-227, Jul. 2001.

[12] M. D. Hill and A. J. Smith, "Experimental evaluation of on-chip microprocessor cache memories", *Proc. 11th Annual International Symposium on Computer Architecture*, pp. 158-166, 1984.

[13] J. L. Hennessy and D. A. Patterson, "Computer architecture a quantitative approach", *Morgan Kaufman Publishers, Inc*., 2nd Edition, pp. 412-413, 1997.

[14] M. Hill, DINERO III Cache Simulator: Code and Documentation, *University of Wisconsin at Madison*, 1998.