

임베디드 멀티프로세서 환경에서의 중복 데이터를 고려한 캐시 누전 에너지 관리 기법

(Replication-Aware Leakage Management Technique
in Embedded Multiprocessors with Private L2 Caches)

김 현 희, 김 지 흥*
서울대학교 컴퓨터 공학과
(Hyunhee Kim, Jihong Kim)

(School of Computer Science & Engineering , Seoul National University)

Abstract : In modern embedded multiprocessors, the leakage power dissipation has become a critical issue as the process technology advances below 90nm. In particular, the leakage power management of the L2 cache is important in designing low-power embedded multiprocessors because they generally employ a large L2 cache for the performance issue, which making the leakage power consumption of the L2 cache a major power source in the total chip.

In this paper, we propose the technique to reduce the leakage energy consumption in a private L2 cache of embedded multiprocessors by selectively turning off replicated data. Turning off the replicated data can reduce the leakage energy consumption without significant performance loss because an extra miss due to the turned off replication is smaller and the replications are often invalidated after being shared by other cache to meet the cache coherence. Experiments show that the proposed technique reduces the energy consumption by 15% on average without any significant performance loss.

Keywords : embedded multiprocessors, private L2 cache, replication, leakage power management

1. Introduction

최근 임베디드 시스템의 기능이 다양화됨에 따라 기존의 한 가지 기능만을 제공하던 시스템에서 점차 복잡한 기능을 제공하는 방향으로 발전하고 있다. 특히, 여러 가지 프로그램을 동시에 수행시키기 위해 하나의 칩 안에 다수의 프로세서를 사용하는 멀티프로세서 환경을 고성능 임베디드 시스템에 적용하는 사례가 증가하고 있다. 그러나, 이와 같은 임베디드 시스템은 대부분 배터리 기반으로 동작하기 때문에, 이들 시스템에서 전력을 줄이기 위한 노력은 꾸준히 중요한 이슈가 되어오고 있다. 또한, 프로세서 공정이 65nm이하로 발전하면서 누전 에너지 소모는 전체 칩 에너지 소모의 50%이상을 차지할 것이라고 예측되고 있으며 [1], 이는 누전 에너지 관리 기법이 더욱 중요해 지고 있음을 시사한다.

한편, 멀티프로세서 환경에서는 프로세서와 메모

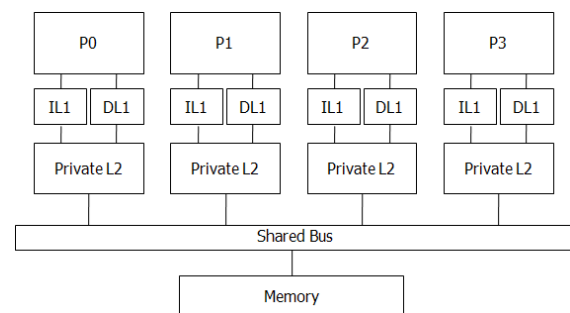


그림 1. 타겟 아키텍처

Fig. 1. Target architecture.

리 속도 차이로 인한 성능 저하를 해결하기 위해 칩 면적의 많은 부분을 L2 캐시에 할애한다. 이와 같은 사실은 L2 캐시가 전체 칩 전력 소모에서 많은 전력 소비를 차지하는 컴포넌트가 되고 있다는 의미이며, L2 캐시에서의 누전 에너지를 관리하는 기법이 중요해 짐을 의미한다.

이와 같은 캐시에서 소모되는 누전 에너지를 줄

```

1: if (이웃 캐시가 해당 블록의 복사본을 할당)
2:   if (캐시 블록이 턴 온 상태)
3:     캐시 블록을 턴 오프 상태로 변경:
4:   end if
5: else if (프로세서가 지역 캐시 블록을 읽음)
6:   if (캐시 블록이 S 상태이면서 턴 오프 상태)
7:     if (access_counter < 임계치)
8:       access_counter += 1;
9:     else
10:      캐시 블록을 턴 온 상태로 변경:
11:      access_counter = 0;
12:    end if
13:  end if
14: else if (프로세서가 지역 캐시 블록을 변경)
15:   if (캐시 블록이 S 상태이면서 턴 오프 상태)
16:     캐시 블록을 턴 온 상태로 변경:
17:     access_counter = 0;
18:   end if
19: end if

```

그림 2. 제안하는 기법의 알고리즘
 Fig. 2. Algorithm of the proposed technique

이기 위해 여러 관련 연구들이 있어 왔다. Cache Decay [2] 기법은 앞으로 사용되지 않을 것이라고 예측 되는 캐시 메모리의 전력 공급을 끊어 캐시 블록을 턴 오프(turn-off) 시킴으로써 캐시에서 소모되는 누전 에너지를 줄인다. 이 기법에서는 캐시 블록이 미리 정해진 타임아웃 임계치 시간동안 접근되지 않으면 다시 사용되지 않을 블록으로 예측한다. 하지만, 블록이 턴 오프 상태가 되면 데이터를 유지하지 않으므로, 다시 접근되는 경우 오프 칩(off-chip memory)를 다시 접근해야 하는 비용이 들게 된다. 이와 같은 비용을 줄이기 위해, Drowsy Cache [3] 논문에서는 데이터를 유지하면서 공급 전압을 줄여 누전 에너지 소모를 줄이는 기법을 제안하였다. 그러나, 이들 기법은 단일 프로세서 환경에서 캐시의 누전 에너지를 줄이기 위한 기법들을 제안하고 있으며, 멀티프로세서 환경에서의 캐시 블록들의 특성들은 고려하지 않고 있다.

본 논문에서는 멀티프로세서 환경에서의 캐시 블록들이 갖는 특성들을 고려함으로써, 기존 Cache Decay 기법을 사용하는 환경에서 추가적으로 누전 에너지를 줄이기 위한 기법을 제공한다. 그림 1은 논문에서 가정하는 타겟 아키텍처를 보여 준다. 타겟 아키텍처는 4개의 프로세서로 구성되고, 각 프로세서가 사유의 L1 데이터/명령 캐시 및 L2 캐시를 갖고 있다. 또한 L2 캐시들은 데이터의 일관성을 유지하기 위해 snooping 기반의 MESI (Modified, Exclusive Clean, Shared, Invalid) 캐시 일관성 프로토콜을 사용하며, 공유 버스를 사용하여 통신한다.

제안하는 기법은 이와 같은 환경에서 사유 L2 캐시에서 중복되는 데이터를 선택적으로 끄으로써

누전 에너지 소모를 줄인다. 중복 데이터들은 칩 내에 데이터의 복사본을 유지하고 있기 때문에, 일찍 데이터를 끄으로써 발생하는 추가적인 미스 시 비용을 크게 늘이지 않고 에너지 소모를 줄일 수 있다. 또한, 많은 공유 데이터의 경우, 공유 되고 난 후 데이터를 공유하고 있는 프로세서들 중 한 프로세서가 데이터를 업데이트 하면 무효화(Invalidate) 되어야 한다. 이 때, 캐시 블록이 무효화되기 전에 캐시 블록을 꺼도 성능 상의 손실이 없기 때문에, 해당 블록을 끄기 위해 기존 기법에서와 같이 타임아웃 임계치동안 기다릴 필요가 없어진다.

이와 같이 중복 데이터를 끄으로써, 제안하는 기법은 L2 캐시에서 소모되는 에너지를 기존 Cache Decay 기법에 비해 15% 정도 줄일 수 있었다. 본 논문의 구성은 다음과 같다. 다음 II절에서는 제안하는 기법을 설명하며, III절에서는 실험결과를 보여 준다. 그리고 마지막으로 IV절에서 결론을 맺는다.

II. 중복 데이터를 고려한 누전 에너지 관리 기법

본 논문에서 제안하는 기법은 사유 L2 캐시를 갖는 멀티프로세서 환경에서 각 L2 캐시에 할당되는 중복 데이터를 선택적으로 끄으로써 L2 캐시에서 소모되는 누전 에너지 소모를 줄인다. 맨 처음 L2 캐시에서 미스가 발생할 때, 데이터는 오프 칩 메모리로부터 읽혀진다. 이때, MESI 일관성 프로토콜에서는 캐시의 상태를 E로 유지한다. 반면, 다른 L2 캐시에서 동일 블록에 대해 미스가 발생한 경우 MESI 캐시 프로토콜은 메모리 혹은 이웃 캐시에서 데이터 블록을 읽어오기 위해 BusRead 트랜잭션을 발생시킨다. 이때, 해당 데이터 블록을 갖고 있는 캐시는 블록의 상태를 S로 바꾸면서 데이터를 요청하는 캐시로 전송하게 된다. 이때, 요청하는 캐시의 데이터 블록도 S로 셋팅되며, 이는 해당 데이터 블록이 중복 데이터임을 의미한다.

제안하는 기법에서는 이와 같이 BusRead를 받는 캐시가 해당 데이터를 요청하는 캐시로 전송할 때, 캐시 블록의 상태를 S로 바꾸면서 데이터 블록을 끄으로써 누전 에너지 소모를 추가적으로 줄일 수 있다. 턴 오프상태의 블록에 대해 읽기 미스가 다시 발생하는 경우, 블록은 다시 켜지지 않고 이웃 캐시로부터 읽어올 수 있다. 반면, 턴 오프상태의 블록에 대해 변경이 일어나면, 해당 블록을 턴 온(turn-on) 시킨다. 이는 공유 상태의 캐시 블록이

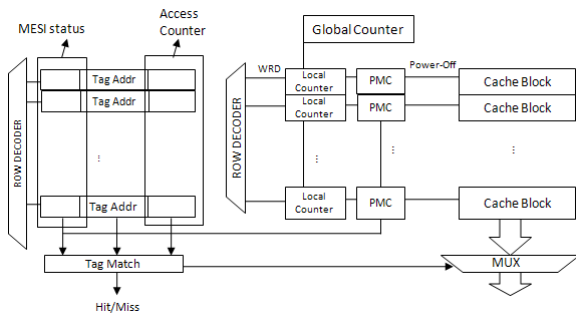


그림 3. 제안하는 기법의 블록 다이어그램
Fig. 3. Block diagram of the proposed technique.

변경되는 경우, 다른 이웃 캐시들의 해당 블록들은 무효화 (invalidate) 시키면서 변경되는 캐시의 블록이 새로운 값을 유지해야 하기 때문이다.

한편, 제안하는 기법에서 턴 오프상태의 공유 캐시 블록에 대해 읽기 미스가 발생했을 때, 캐시 블록은 매번 이웃 캐시에서 읽혀져야 한다. 이는 이웃 캐시 접근에 대한 비용이 메모리 접근 비용에 비해 매우 작기 때문에 작은 성능 저하를 유지하면서 누진 에너지 소모를 줄일 수 있지만, 매우 빈번히 접근되는 캐시 블록에 대해서는 성능 저하가 커진다. 이와 같은 경우 성능이 저하되는 것을 컨트롤하기 위해 제안하는 기법에서는 턴 오프 상태의 공유 블록이 접근되는 빈도를 추적하여, 접근 빈도가 미리 정해진 임계 값을 넘은 경우 해당 블록을 턴 온 상태로 변경하고, 다시 턴 오프하지 않는다.

그림 2는 제안하는 기법의 알고리즘을 보여준다. 위에서 설명한 것과 같이, 이웃 캐시가 해당 블록의 복사본을 할당할 때 캐시 블록을 턴 오프 상태로 만든다. 또한 프로세서가 턴 오프 상태의 공유 캐시 블록을 읽을 때, access_counter가 임계치 보다 작으면 access_counter 값을 하나 증가시키며, 임계치 보다 크면 이를 턴 온 상태로 바꾸면서 access_counter의 값을 0으로 리셋한다. 반면, 프로세서가 턴 오프상태의 공유 캐시 블록을 변경하면 캐시 블록을 턴 온 상태로 변경하면서 access_counter의 값을 0으로 리셋한다.

그림 3은 제안된 기법의 블록 다이어그램을 보여준다. 기존 연구인 Cache Decay기법에서와 같이 타임아웃 임계치에 이른 캐시 블록을 턴 오프하기 위해 2단계 카운터를 사용하여, 공유 데이터를 제외한 블록들에 대해서는 타임아웃 턴 오프 기법을 적용한다. 따라서, 각 캐시 블록은 작은 local counter를 갖고 global counter가 틱(tick)을 보낼

때 마다 증가한다. local counter가 타임아웃 임계치 값에 다다르면 해당 블록은 턴 오프상태로 바뀐다. 한편, 제안하는 기법에서는 캐시 일관성을 유지하기 위해 캐시의 태그(tag)는 턴 오프하지 않는다. PMC는 본 논문에서 제안하는 기법을 이용하여 캐시 블록을 턴 오프 하기 부분이며, local counter의 신호와 MESI 상태의 변화, access_counter 값을 이용하여 블록을 턴 오프 할지 결정한다.

III. 실험 결과

본 논문에서 제안하는 기법을 평가하기 위해 CATS 멀티프로세서 시뮬레이터를[4] 이용하였으며, 캐시 에너지 파라미터들은 CACTI 4.0[5]로부터 얻었다. 또한 벤치마크로는 SPLASH2의[6] VOLREND, FMM, CHOLESKY, ALPBench의[7]의 Decoder와 Encoder를 이용하였다.

그림 4에서는 각 기법의 정규화된 에너지 소모를 보여준다. BASELINE에서는 캐시 블록 턴 오프 기법을 사용하지 않은 기본적인 캐시 구조를 사용하였으며, DECAY에서는 Cache Decay기법에서 제안하는 타임아웃 기반의 캐시 턴 오프 기법을 이용하였다. 마지막으로, RALM은 제안하는 기법의 실험 결과를 보여준다.

그림에서 볼 수 있듯이, 제안하는 기법은 모든 벤치마크들에 대해 평균 적으로 15% 정도의 에너지 소모를 줄일 수 있었다. 특히, Decoder와 Encoder에서는 최대 30% 이상의 에너지 소모를 줄일 수 있었다. 이는 본 벤치마크들이 SPLASH2 벤치마크들에 비해 많은 비율의 공유 데이터를 가지고 있기 때문이다. 따라서, SPLASH2 벤치마크들에 대해서는 Decoder와 Encoder에 비해서는 에너지 감소가 적음을 확인할 수 있다.

그림 5는 각 기법의 정규화된 실행 시간을 보여준다. 그림에서 확인할 수 있듯이, 모든 벤치마크들에 대해 실행 시간이 DECAY기법에 대해 실행 시

표 1. 시뮬레이션 변수

Table 1. Architectural parameter.

Processor	4 Processors, in-order
L1 I/D cache	32KB, 1-way, 32B block, 1 cycle latency
L2 Private Cache	33.8 %
Shared Bus	4bytes bus width
Off-Chip Memory	300 cycle access latency

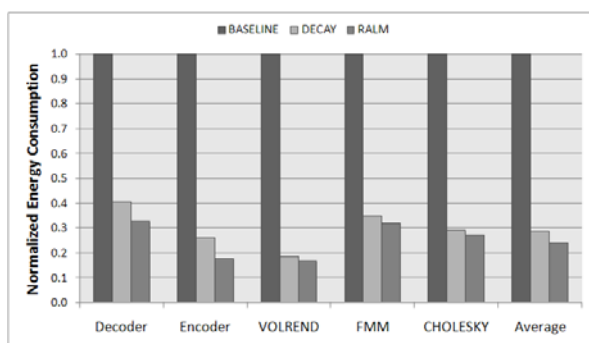


그림 4. 정규화된 에너지 소모

Fig. 4. Normalized energy consumption.

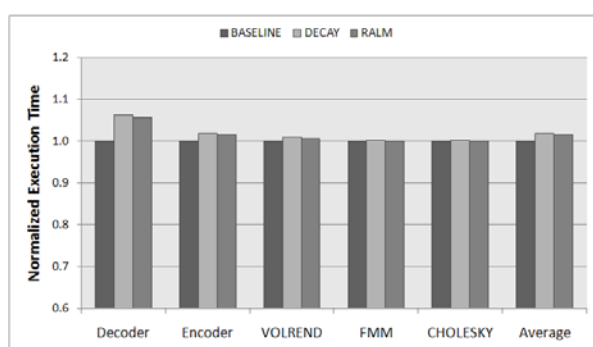


그림 5. 정규화된 실행 시간

Fig. 5. Normalized Execution Time.

간이 조금 감소한 것을 볼 수 있다. 이는 제안하는 기법에서 온 칩상에 오직 하나 남은 공유 데이터에서는 타임아웃 기법은 적용하지 않기 때문에, DECAY 기법에 비해 오프 칩 접근이 줄어들었기 때문이다.

IV. 결 론

본 논문에서는 멀티프로세서 환경의 L2 캐시에서 중복 데이터를 선택적으로 끄므로써 누전 에너지 소모를 감소시키기 위한 기법을 제안한다. 제안하는 기법은 사유 L2 캐시 블록에서 캐시 블록의 복사본이 만들어 질 때, 온 칩상에 중복된 데이터들 중 하나만을 유지시키고 나머지 중복된 데이터들에 대한 캐시 블록은 턴 오프 상태로 만듦으로써 누전 에너지를 줄인다. 이와 같은 중복된 데이터들을 끄는 것은, 데이터들이 다시 요구될 때, 이웃 캐시에서 읽어올 수 있기 때문에 성능 상의 비용이 적게 들면서, 공유 블록을 많이 갖는 프로그램들에 대해 에너지를 크게 줄일 수 있다는 장점이 있다. 실험 결과, 제안하는 기법이 성능 저하 없이 평균 15% 정도의 에너지 소모를 줄일 수 있었다.

참 고 문 헌

- [1] ITRS (Intl Technology Roadmap for Semiconductor), <http://public.itrs.net>.
- [2] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: exploiting generational behavior to reduce cache leakage power," Proc. of the International Symposium on Computer Architecture, pp. 240-251, June, 2001.
- [3] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," Proc. of the International Symposium on Computer Architecture, pp. 148-157, May, 2002.
- [4] D. Kim, S. Ha, and R. Gupta, "CATS: cycle accurate transaction-driven simulation with multiple processor simulators," Proc. of the Design, automation and test in Europe, pp. 749-754, April, 2007.
- [5] T. David, T. Shyamkumar, and J. Norman, "Cacti 4.0: An integrated cache timing, power and area model," <http://www.hpl.hp.com/research/cacti/>, 2006.
- [6] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh and A. Gupta, "The SPLASH-2 Programs: characterization and methodological considerations," Proc. of the International Symposium on Computer Architecture, pp. 24-36, June, 1995.
- [7] M. Li, R. Sasanka, S. V. Adve, Y. Chen, and E. Debes, "The ALPBench benchmark suite for complex multimedia applications," Proc. of the IEEE International Symposium on Workload Characterization," pp. 34-45, October, 2005.
- [8] D. E. Culler, J. P. Singh, A. Gupta, "Parallel Computer Architecture: A Hardware/Software Approach," Morgan Kaufman, 1999.