

Energy-efficient Set-associative Cache Using Bi-mode Way-selector

Sungjae Lee[†] · Jinku Kang[†] · Juho Lee^{**} · Jiyong Youn^{**} · Inhwan Lee^{***}

ABSTRACT

The way-lookup cache and the way-tracking cache are considered to be the most energy-efficient when used for level 1 and level 2 caches, respectively. This paper proposes an energy-efficient set-associative cache using the bi-mode way-selector that combines the way selecting techniques of the way-tracking cache and the way-lookup cache. The simulation results using an Alpha 21264-based system show that the bi-mode way-selecting L1 instruction cache consumes 27.57% of the energy consumed by the conventional set-associative cache and that it is as energy-efficient as the way-lookup cache when used for L1 instruction cache. The bi-mode way-selecting L1 data cache consumes 28.42% of the energy consumed by the conventional set-associative cache, which means that it is more energy-efficient than the way-lookup cache by 15.54% when used for L1 data cache. The bi-mode way-selecting L2 cache consumes 15.41% of the energy consumed by the conventional set-associative cache, which means that it is more energy-efficient than the way-tracking cache by 16.16% when used for unified L2 cache. These results show that the proposed cache can provide the best level of energy-efficiency regardless of the cache level.

Keywords : Set-associative Cache, Energy-efficiency, Bi-mode Way-selector

1. 서 론

캐시는 빠른 프로세서와 상대적으로 느린 메모리 간의 지연 시간의 차이를 줄여주어 프로세서의 성능을 향상시키는 중요한 역할을 수행한다. 캐시의 적중률은 전체 시스템의 성능에 직접적인 영향을 미치기 때문에, 캐시의 적중률을 향상시키기 위하여 하드웨어 또는 에너지 측면에서 추가적인 비용을 거의 필요로 하지 않는 연관사상캐시(Set-associative cache)가 많이 사용되고 있다[1]. 고성능 시스템의 캐시에서 적중률 못지 않게 또한 중요한 것은 캐시의 에너지 효율이다. 실제 데이터에 의하면, Alpha 21264[2] 프로세서와 StrongARM-110[3] 프로세서의 전체 소비전력 중에서 각각 16% 및 43%가 캐시에서 소비된다. 따라서, 높은 에너지 효율을 갖는 연관사상캐시 구조를 개발하는 것이 매우 중요하다.

선택적웨이접근 연관사상캐시(Way-selective set-associative cache)[4-8]는 이러한 요구에 맞추어 제안되었다. 이 방식에서는, 매번 프로세서로부터 캐시 블록이 요청될 때, 연관사상캐시의 모든 웨이들 중에서 일부의 웨이만을 선택적으로 접근한다. 이때 선택된 웨이에 요청된 캐시 블록이 존재하면, 일반적인 연관사상캐시에 비하여, 나머지 선택되지 않은 웨이들로의 접근에 필요한 만큼의 에너지를 절약할 수 있다. 지금까지 제안된 여러 선택적 웨이접근캐시들 가운데 웨이룩업캐시(Way-lookup set-associative cache)[7]와 웨이

추적캐시(Way-tracking set-associative cache)[8]는 가장 우수한 수준의 에너지 효율을 제공한다. 하지만, 이 두 구조는 뚜렷이 대비되는 장단점을 가진다. 구체적으로, 웨이룩업캐시는 레벨1 캐시에서는 가장 우수한 수준의 에너지 효율을 제공하지만, 레벨2 캐시에서는 에너지 효율이 낮다. 반면, 웨이추적캐시는 레벨2 캐시에서 가장 우수한 에너지 효율을 제공하지만, 레벨1 캐시에서는 웨이룩업캐시에 비해 에너지 효율이 낮다. 본 논문에서 다루는 모든 캐시 구조들은 접근하는 웨이의 개수를 최소화함으로써 에너지 효율을 향상시키는 기술이며, 따라서 본 논문에서 언급하는 에너지 소모량은 모두 동적 에너지(Dynamic energy) 소모량을 의미한다.

본 논문에서는 상호보완적인 특성을 갖는 웨이추적캐시와 웨이룩업캐시에 대한 연구를 토대로, 각각의 장점을 효율적으로 결합하는 이중웨이선택형 연관사상캐시 (Bi-mode way-selecting set-associative cache)를 제안한다. 본 논문에서는 이중웨이선택형 연관사상캐시를 간단히 이중웨이선택캐시라 칭한다. 이중웨이선택캐시는 웨이추적캐시와 웨이룩업캐시의 웨이선택기법들을 함께 사용하면서, 매번 프로세서가 캐시 블록을 요청할 때 접근하는 웨이의 개수를 최소화하는 방향으로 동작함으로써, 어느 레벨의 캐시에 사용되더라도 가장 우수한 수준의 에너지 효율을 제공한다. 본 논문에서는 제안한 이중웨이선택캐시의 성능을 SPEC CPU2K 벤치마크 어플리케이션들을 이용하여 시뮬레이션을 통해 평가한다. 구체적으로, 이중웨이선택캐시의 성능을 캐시의 동적 에너지 소모량, 캐시의 접근지연시간 그리고 하드웨어 오버헤드 측면에서 웨이추적캐시 및 웨이룩업캐시와 함께 비교하여 평가한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 선택적 웨이접근캐시에 관한 연구에 대해 살펴보고, 3장에서는 본

[†] 준 회원: 한양대학교 전자컴퓨터통신공학과 박사과정

^{**} 준 회원: 한양대학교 전자컴퓨터통신공학과 석사과정

^{***} 정 회원: 한양대학교 컴퓨터공학부 교수

논문접수: 2012년 8월 9일

수정일: 1차 2012년 9월 10일

심사완료: 2012년 9월 24일

* Corresponding Author: Inhwan Lee(ihlee@hanyang.ac.kr)

논문에서 제안하는 이중웨이선택캐시의 구조와 동작을 다룬다. 4장에서는 제안한 캐시 구조의 성능을 평가하기 위한 시뮬레이션 환경에 대하여 설명하고, 5장에서는 성능 평가의 결과를 제시한다. 마지막으로 6장에서 본 논문의 결론을 내린다.

2. 관련 연구

연관사상캐시의 에너지 효율을 개선하기 위하여 선택적으로 웨이를 접근하는 캐시 구조에 대한 많은 연구가 진행되었고, 몇몇 대표적인 연구들은 다음과 같다. Inoue et al.[4]은 웨이예측 연관사상캐시(Way-predicting set-associative cache)를 제안하였다. 웨이예측캐시는 캐시의 각 세트(set)별로 가장 최근에 참조한(Most recently referenced; MRR) 캐시 블록의 웨이 정보를 기록한다. 프로세서가 메모리를 접근할 때마다, 웨이예측캐시는 일단 MRR 캐시 블록이 다시 사용된다고 예측(또는 가정)하고, 기록된 MRR 웨이 정보가 가리키는 웨이로 접근해서 태그 값을 비교하여 캐시 적중(Cache hit)이 되었는지의 여부를 판단한다. 만약 예측이 맞으면 여러 웨이들 중에서 하나의 웨이만을 접근하기 때문에 에너지 소비를 크게 줄일 수 있다. 하지만, 예측이 틀린 경우에는 추가로 나머지 웨이들에 접근하여 원하는 블록을 찾게 되어 에너지 측면에서 이득이 없을 뿐 아니라, 두 차례에 걸쳐 웨이를 접근하기 때문에 접근지연시간이 길어지는 단점이 발생한다. 따라서 웨이예측캐시의 에너지 효율은 웨이예측의 성공률에 의해 결정되고 이는 다시 메모리 접근에서의 지역성(Locality)에 의해 결정된다. 실제로 웨이예측캐시는 지역성이 높은 레벨1 캐시에서는 가장 우수한 에너지 효율을 제공하나, 지역성이 낮은 레벨2 캐시에서는 웨이예측캐시의 에너지 효율이 크게 떨어진다.

또한, Hasegawa et al.[5]은 단계접근캐시(Phased cache)를 제안하였다. 일반적인 연관사상캐시에서는 캐시 적중이 되었는지의 여부를 신속히 판단하기 위하여 각 웨이들의 태그 열과 데이터 열을 병렬로 접근한다. 그러나 단계접근캐시는 각 웨이들의 태그 열과 데이터 열을 순차적으로 접근한다. 구체적으로, 단계접근캐시는 모든 웨이의 태그 열들을 우선 접근하여 원하는 블록이 위치하는 하나의 웨이를 찾은 후, 해당 웨이의 데이터 열만을 접근하고 나머지 웨이들의 데이터 열을 접근하지 않음으로써 에너지의 소비를 줄인다. 단계접근캐시는 웨이예측캐시에서와 같은 지역성을 기반으로 한 예측에 의존하지 않기 때문에, 단계접근캐시의 에너지 효율은 상대적으로 지역성의 영향을 덜 받는다. 그 결과, 단계접근캐시는 레벨1 캐시에서는 높은 지역성을 잘 활용하는 웨이예측캐시에 비해 에너지 효율이 낮지만, 지역성이 낮은 레벨2 캐시에서는 웨이예측캐시에 비해 훨씬 우수한 에너지 효율을 제공한다. 하지만, 단계접근캐시는 태그 열과 데이터 열을 순차적으로 접근하기 때문에 접근지연시간이 길어지는 단점이 있다.

웨이룩업캐시는 웨이예측캐시에서 예측이 실패한 경우 접

근지연시간이 길어진다는 단점을 보완하기 위하여 제안되었다[7]. 웨이룩업캐시는 캐시의 각 세트에 대하여 MRR 캐시 블록의 웨이 정보뿐 아니라 태그 값을 함께 기록하는 웨이룩업버퍼(Way-lookup buffer; WLB)를 사용한다. 웨이룩업캐시는, 웨이예측캐시에 비해 태그 정보를 추가로 사용하기 때문에, WLB가 가리키는 웨이에 요청된 캐시 블록이 실제로 존재하는지의 여부를 WLB 내부에 있는 정보를 이용하여 빠르고 정확히 결정할 수 있다. 웨이룩업캐시는 요청된 캐시 블록이 WLB가 가리키는 웨이에 실제로 존재한다고 확인된 경우에만 WLB가 선택한 웨이를 접근하며, 그렇지 않은 경우에는 WLB가 선택한 웨이를 제외한 나머지 웨이들을 접근한다. 따라서, 웨이예측이 실패한 경우 두 차례에 걸쳐 순차적으로 웨이를 접근하는 웨이예측캐시와 달리, 웨이룩업캐시에서는 항상 한 차례만 웨이를 접근하기 때문에 웨이예측캐시의 단점인 접근지연시간이 길어지는 현상이 발생하지 않는다. 웨이룩업캐시는 웨이예측캐시와 같거나 또는 근소하게 우수한 수준의 에너지 효율을 제공한다.

마지막으로, 웨이추적캐시는 캐시의 각 블록에 대해 이 블록이 존재할 가능성이 있는 모든 웨이에 대한 정보를 웨이추적테이블(Way-tracking table; WTT)에 유지하고 관리한다[8]. 웨이추적캐시는 WTT를 이용하여 요청된 캐시 블록이 존재할 가능성이 있는 모든 웨이들을 한꺼번에 선택한다. 따라서 웨이추적캐시에서도 웨이룩업캐시에서와 마찬가지로 접근지연시간이 길어지는 현상이 발생하지 않는다. 실제 데이터에 의하면, 웨이추적캐시는 많은 경우 하나만의 웨이를 선택해 주며 이에 따라 에너지 소비를 크게 줄인다. 또한 웨이추적캐시는 WTT만을 이용하여 캐시 적중실패(Cache miss)를 정확히 판단할 수 있어, 이 경우의 불필요한 웨이 접근을 사전에 방지하여 에너지 소비를 줄인다. 웨이추적캐시는 MRR 정보에만 의존하지 않고 더 포괄적인 정보를 제공하는 WTT를 사용하기 때문에, 웨이추적캐시의 에너지 효율은 웨이룩업캐시의 에너지 효율에 비해 지역성의 영향을 덜 받는다. 구체적으로, 웨이추적캐시는 지역성이 높은 레벨1 캐시에서는 웨이룩업캐시에 비해 상대적으로 에너지 효율이 낮지만, 레벨2 캐시에서는 가장 우수한 에너지 효율을 제공한다.

요약하면, 웨이룩업캐시는 레벨1 캐시에서 가장 우수한 수준의 에너지 효율을 제공하고, 웨이추적캐시는 레벨2 캐시에서 가장 높은 에너지 효율을 제공한다. 또한 웨이룩업캐시와 웨이추적캐시에서는 다른 선택적 웨이접근캐시에서와 달리 접근지연시간이 길어지는 현상이 발생하지 않는다. 본 논문에서는, 웨이룩업캐시와 웨이추적캐시의 웨이선택기법들을 함께 사용함으로써, 레벨1 캐시 및 레벨2 캐시 중에서 어디에 사용되더라도 가장 우수한 수준의 에너지 효율을 제공하는 이중웨이선택캐시를 제안한다.

3. 이중웨이선택형 연관사상캐시

웨이룩업캐시는 WLB가 가리키는 웨이에 요청된 캐시 블

록이 존재한다는 것이 확인된 경우(이 경우를 WLB 적중이라 함)에는 하나의 웨이만을 접근하여 에너지의 소비를 크게 줄일 수 있으나, 그렇지 않은 경우에는 선택된 웨이를 제외한 나머지 모든 웨이들을 접근하므로 큰 이득을 얻을 수 없다. 다시 말해, 웨이록업캐시의 에너지 효율은 WLB의 적중률에 의해 결정된다. 따라서, 높은 지역성을 가지는 레벨1 캐시에서는 WLB의 적중률이 높아 웨이록업캐시가 가장 우수한 수준의 에너지 효율을 제공하지만, 낮은 지역성을 가지는 레벨2 캐시에서는 웨이록업캐시의 에너지 효율이 상당히 떨어진다. 반면, 웨이추적캐시는 WTT를 기반으로 요청된 캐시 블록이 존재할 가능성이 있는 모든 웨이들을 한꺼번에 선택한다. 따라서 웨이추적캐시의 에너지 효율은 WTT에서 선택되는 웨이의 개수가 적을수록 높아진다. 실제로 웨이추적캐시는 지역성의 영향을 상대적으로 덜 받아 캐시의 레벨과 무관하게 많은 경우에 한 개의 웨이를 선택하며 (5장 참조), 이에 따라 웨이추적캐시는 레벨2 캐시에서 가장 높은 에너지 효율을 보여준다. 여기서 보듯이 선택적웨이접근 연관사상캐시의 에너지 효율을 높이기 위한 가장 핵심적인 사항은 어떻게 하면 접근할 웨이의 개수를 최소화할 것인가 하는 것이다. 제안하는 이중웨이선택캐시는 웨이록업캐시와 웨이추적캐시의 웨이선택기법들을 결합하여, 즉 WLB와 WTT를 함께 사용함으로써, 매번 프로세서가 캐시 블록을 요청할 때 접근하여야 하는 웨이의 평균적인 개수를 보다 더 줄이고자 한다. 다시 말하면, 이중웨이선택캐시는 WLB 및 WTT를 이용한 웨이선택결과를 기반으로 접근할 웨이의 개수가 최소화되도록 결과를 취함으로써, 캐시의 레벨에 무관하게 가장 우수한 수준의 에너지 효율을 제공한다.

Fig. 1은 제안한 이중웨이선택캐시의 구조를 4-웨이 연관사상캐시의 예를 통해 보여준다. 이중웨이선택캐시는 크게 웨이록업캐시의 WLB와 웨이추적캐시의 WTT, 그리고 웨이활성유닛(Way enable unit)으로 구성된다. 여기서 WLB와 WTT가 관리하는 정보 및 이들의 개별적인 동작은 각각 웨이록업캐시와 웨이추적캐시에서의 그것들과 동일하다. 즉, WLB는 캐시의 각 세트 별로 MRR 캐시 블록의 태그 값과 웨이 정보 쌍을 관리하며[7], WTT는 각 블록에 대하여 이 블록이 존재할 가능성이 있는 모든 웨이에 대한 정보를 관리한다[8]. 그리고 웨이활성유닛은 WLB와 WTT에서 각각 제공하는 웨이선택정보 중에서 하나를 취하여 선택된 웨이들만을 접근하도록 한다. Fig. 1에서 보듯이 WTT는 여러 개의 작은 테이블들로 구성된다. 이들 테이블의 개수는 캐시에 있는 웨이의 개수와 같으며, 각각의 테이블은 태그의 하위 비트 정보를 이용하여 선택된다. 예를 들어, 4-웨이 웨이추적캐시의 WTT는 내부에 네 개의 테이블들을 가지고 있고, 태그의 하위 2비트 정보를 이용하여 어느 테이블에 있는 정보를 사용할 것인지를 결정한다. Fig. 1에서 각각의 테이블 내부에 있는 정보가 무엇을 의미하는지는 아래에서 예를 들어 간단히 설명한다.

프로세서가 캐시블록을 요청할 때, 이중웨이선택캐시는 우선 WLB와 WTT에 접근하여 각각으로부터 웨이선택정보를 얻는다. 그 다음의 동작은 다음의 세 가지 경우로 구분된다. 우선, WLB 적중이 발생하면, 이는 WLB가 요청된 캐시 블록이 위치하는 하나의 웨이만을 정확히 선택할 수 있음을 의미하므로, 웨이활성유닛은 WLB의 웨이선택정보를 취하여 하나의 웨이만을 접근함으로써 에너지의 소비를 최

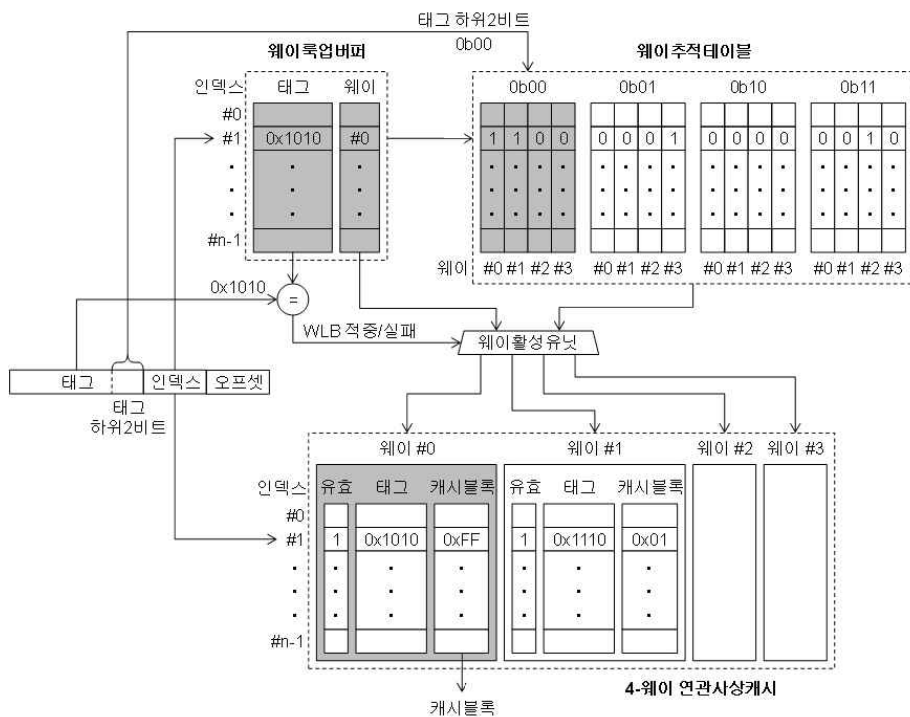


Fig. 1. Bi-mode way-selecting 4-way set-associative cache

소화한다. 둘째로, WLB 적중이 발생하지 않으면 웨이활성유닛은 WTT가 제공하는 웨이선택정보를 취하는데, 여기서 만일 요청된 캐시 블록이 캐시에 존재한다면 WTT는 요청된 캐시 블록이 위치하는 웨이를 포함하는 하나 이상의 웨이들을 선택하게 된다. 시뮬레이션 결과에 의하면, 이 때 실제로 WTT가 선택하는 웨이의 개수는 많은 경우 한 개이기 때문에, 이 경우 역시 소비되는 에너지를 상당히 줄일 수 있다. 세 번째의 경우로서, WLB 적중이 발생하지 않아 WTT의 웨이선택정보를 취하였으나 실제로 WTT가 어느 웨이도 선택하지 않았다면, 이는 캐시 적중실패를 의미한다. 이 때 이중웨이선택캐시는 불필요한 웨이 접근을 생략하고 바로 하위 레벨의 캐시 또는 메모리로 접근함으로써 에너지 소비를 줄인다. 여기서 중요한 것은 이중웨이선택캐시는 위의 세 가지 경우 모두에서 에너지 소비가 최소화되는 방향으로 동작한다는 것이다. 프로세서가 요청한 캐시 블록을 제공하기 위한 동작을 완료한 후, WLB와 WTT는 필요에 따라 업데이트된다. 구체적으로, WLB는 WLB 적중이 발생하지 않았을 경우에 새로운 MRR 캐시 블록에 대한 정보를 이용하여 WLB를 업데이트하며[7], WTT는 캐시의 적중실패로 인해 캐시에 새로운 캐시블록이 할당될 때 해당 블록이 어느 웨이에 들어왔는지에 대한 정보를 WTT에 반영한다[8].

Fig. 1의 예를 이용하여 위에서 설명한 이중웨이선택캐시의 동작을 보다 자세히 살펴 본다. 만일 요청된 캐시 블록의 태그 값이 "0x1010"이고 인덱스 값이 1이라면, Fig. 1에서 이 경우는 WLB 적중에 해당한다. 이 때 이중웨이선택캐시는 WLB가 가리키는 하나의 웨이, 즉 웨이 #0만을 접근하여 요청된 캐시 블록을 찾을 수 있어 에너지의 소비를 줄일 수 있다. 만약 요청된 캐시블록의 태그 값이 "0x1110" 이고 인덱스 값이 1이라면, WLB는 해당 블록의 정보를 갖고 있지 않고 따라서 WLB 적중이 발생하지 않는 경우이다. 이 때 웨이활성유닛은 WTT를 이용하여 접근할 웨이를 선택한다. 여기서 태그 값의 하위 2비트는 "0b00"이므로, WTT는 네 테이블 중에서 맨 왼쪽의 테이블을 이용하여 웨이를 선택한다. 인덱스 1에 해당하는 맨 왼쪽 테이블의 정보는 "1100"이다. 이 네 비트의 정보는 요청된 블록이 네 개의 웨이 중에서 어느 웨이에 있을 가능성이 있는지를 비트맵의 형태로 나타낸다. 여기서 "1"은 요청된 캐시 블록이 해당 웨이에 존재할 가능성이 있음을, 그리고 "0"은 그럴 가능성이 없음을 의미한다. 따라서 테이블의 정보 "1100"은 요청된 캐시 블록이 웨이 #0 또는 웨이 #1에 있음을 의미한다. 따라서 이중웨이선택캐시는 이 두 개의 웨이만 접근하게 되며, Fig. 1의 예에서는 태그 값을 이용하여 웨이 #1에서 요청된 캐시 블록을 찾을 수 있다. 마지막으로, 요청된 캐시 블록의 태그 값이 "0x1112"이고 인덱스 값이 1이라면, WLB 적중이 발생하지 않고 또한 태그의 하위 2비트 값 "0b10"에 해당하는 WTT의 왼쪽에서 세 번째 테이블의 웨이선택정보는 "0000"이 되어, 이는 어느 웨이도 요청된 캐시 블록을 가지고 있지 않음을 즉 캐시 적중실패를 의미한다. 이 때 이중웨이선택캐시는 어느 웨이도 접근하지 않고 곧바로 하위 레벨의 캐시 또는 메모리로 접근한다.

요약하면, 이중웨이선택캐시는 WLB 적중이 발생하면 WLB가 선택한 웨이, 즉 요청된 캐시 블록이 위치하는 하나의 웨이만을 접근한다. 또한 WLB 적중이 발생하지 않으면, WLB가 선택한 웨이를 제외한 나머지 모든 웨이들에 접근하는 것이 아니라, WTT가 선택한 웨이들을 접근함으로써, WLB만을 사용할 때보다 접근하는 웨이의 개수를 줄인다. 또한 WTT를 이용하여 캐시 적중실패를 빠르고 정확히 판단하여 이 경우에도 불필요한 웨이 접근을 피한다. 따라서, 이중웨이선택캐시에서는 어떠한 경우라도 웨이록업캐시 또는 웨이추적캐시에 비해 접근하는 웨이의 개수가 같거나 적으며 이에 따라 에너지 효율 측면에서 이득을 볼 수 있다. 이 밖에도, 이중웨이선택캐시는 WLB와 WTT를 병렬로 접근하므로, 이중웨이선택캐시의 전체적인 접근지연시간은 웨이록업캐시 또는 웨이추적캐시의 접근지연시간과 별 차이가 없다. 이중웨이선택캐시에서는 WLB와 WTT를 함께 사용하기 때문에 하드웨어 비용이 약간 증가하며, 이에 대해서는 5장에서 살펴 본다.

4. 실험 환경 및 절차

제한한 이중웨이선택캐시의 성능을 평가하기 위해 클락 사이클 단위로 시뮬레이션을 수행하는 실행 기반(Execution-driven cycle-accurate)방식의 시뮬레이터인 SimpleScalar[9]를 사용한다. 프로세서 모델로는 4-이슈의 비순차적 명령어 처리 방식(Out-of-order execution)의 슈퍼 스칼라 프로세서인 Alpha 21264 프로세서를 사용한다. 이 프로세서는 분리된 레벨1의 명령어캐시 및 데이터캐시를 가지고 있다. 레벨1의 명령어캐시 및 데이터캐시는 각각 전체 크기가 16KB인 4-웨이 연관사상캐시이고 캐시 블록의 크기는 64B이다. 통합된 레벨2 캐시(Unified L2 cache)는 전체의 크기가 512KB이고 캐시 블록 크기가 64B인 8-웨이 연관사상캐시를 사용한다.

본 연구에서는 웨이추적캐시, 웨이록업캐시 및 이중웨이선택캐시를 SimpleScalar에 직접 구현하여 다음과 같은 네 가지 측면에서 비교평가를 수행한다. 우선 각각의 캐시에서의 에너지 소모량을 측정하고, 다음에는 매번 프로세서가 캐시 블록을 요청할 때마다 접근하는 웨이의 평균적인 개수를 측정하며, 마지막으로 캐시의 접근지연시간 및 하드웨어 오버헤드를 측정하여, 비교평가를 수행한다. 성능 측정을 위한 벤치마크로는 SPEC CPU2K[10]의 어플리케이션들을 사용하였으며, 모든 어플리케이션의 입력 값으로 SPEC CPU2K에서 제공하는 "Reference" 값을 사용한다. 캐시의 에너지 소모량과 접근 지연 시간 등을 측정하기 위하여 CACTI 6.5[11]를 사용하며, 45nm의 CMOS 기술을 가정한다.

본 연구에서는 다음과 같은 절차로 에너지 소모량을 추정한다. 우선 SPEC CPU2K의 각 어플리케이션을 Simple Scalar 시뮬레이터를 이용하여 실행하면서 매번 캐시블록의 요청에 의해 캐시에 접근할 때마다 각 캐시에서 다음에 해당되는 데이터들을 모두 실시간으로 기록한다: 1) 전체 캐시 블록 요청 회수, 2) 매 캐시블록 요청 시 WLB의 적중 여부

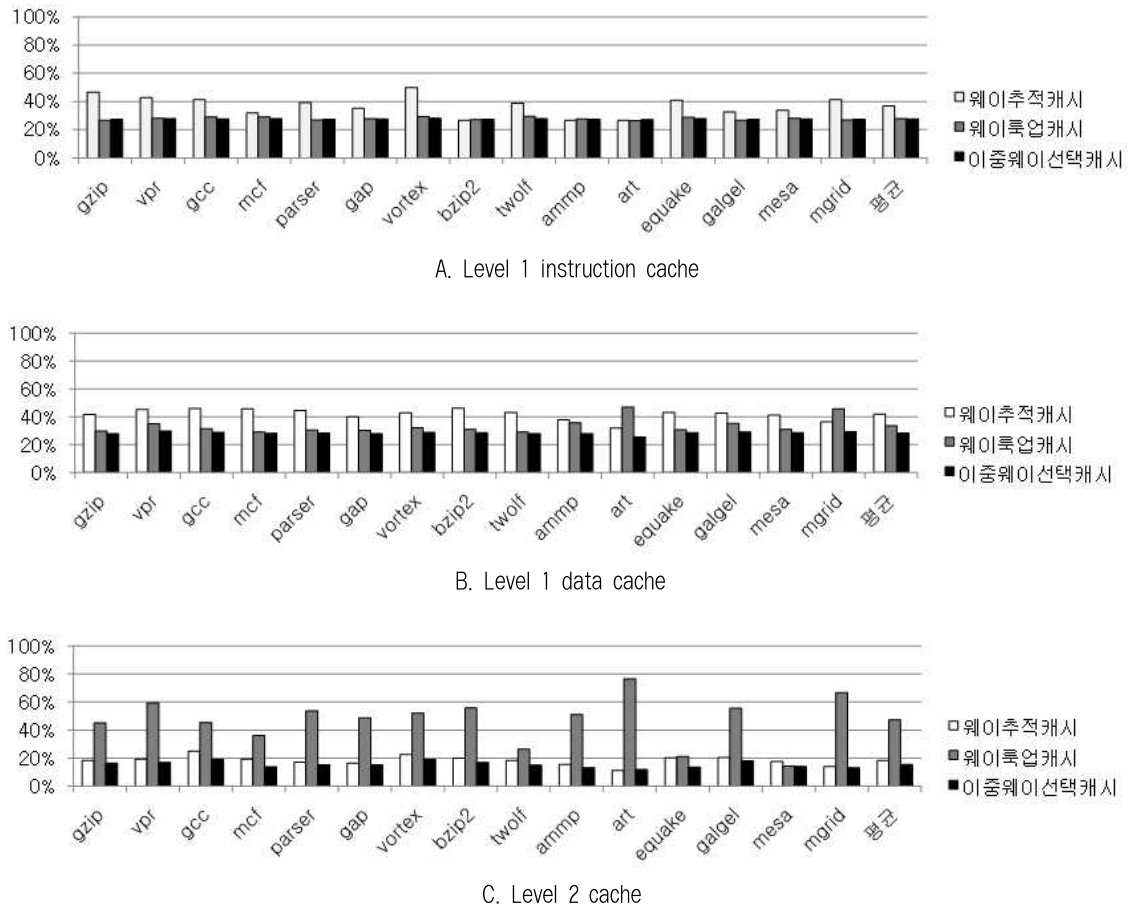


Fig. 2. Relative energy consumption

및 전체 적중 회수, 3) 매 캐시블록 요청 시 WTT에서 선택하는 웨이의 개수, 4) 매 캐시블록 요청 시 각 캐시 구조에서 실제로 접근하는 모든 웨이의 개수. 이 후, CACTI를 이용하여 WLB나 WTT를 사용하지 않은 일반 연관사상캐시의 동작에 필요한 동적 에너지를 추정한다. CACTI에서는 본 논문에서 선택한 캐시 구조에 근거하여 연관성 (Associativity) 즉, 웨이의 크기, 캐시블록의 크기, 태그의 길이 (물리적 메모리 크기) 및 사용할 CMOS 기술을 선택함으로써 묘사된 구조의 캐시를 접근할 때의 각 단위 동작에 소모되는 동적 에너지를 계산한다. 이러한 단위 동작에 소모되는 에너지 소모량의 결과들을 사용하여 캐시블록 요청에 의해 접근하는 웨이의 개수에 따른 동적 에너지 소모량을 계산한다. 다음으로 WLB와 WTT를 접근하는데 필요한 동적 에너지를 추정한다. WLB는 작은 직접사상캐시의 형태로써 앞서 설명한 것과 같은 방법으로 계산을 수행하며 WTT는 단순한 메모리테이블 형태이므로 CACTI가 지원하는 램 (RAM) 구조를 선택하고 그 크기와 CMOS 기술을 입력하여 계산한다. 최종적으로, 앞서 SimpleScalar를 이용한 시뮬레이션에서 얻은 여러 통계 정보와 CACTI로부터 얻은 동적 에너지 정보를 결합함으로써 각 캐시 구조에서 어플리케이션이 실행되는 동안 사용된 동적 에너지 소모량

을 추정한다. 본 논문에서 다루고 있는 모든 캐시 구조들은 접근하는 웨이의 개수를 최소화함으로써 에너지 효율을 향상시키는 기술이며, 따라서 본 논문에서 언급하는 에너지 소모량은 모두 동적 에너지 소모량을 의미한다. 참고로, 시뮬레이션에서 물리적 메모리의 크기를 가상 메모리의 크기와 같은 32비트의 주소 영역으로 가정하였으며, 따라서 블록 크기가 64B인 16KB 4-웨이의 레벨1 캐시에서는 20비트의 태그를 사용하고, 블록 크기가 64B인 512KB 8-웨이의 레벨2 캐시에서는 16비트의 태그를 사용한다.

5. 실험 결과

Fig. 2는 SPEC CPU2K의 어플리케이션들을 실행할 때의 웨이추적캐시, 웨이룩업캐시 및 이중웨이선택캐시에서 사용된 에너지 소모량을 보인다. 이 그림의 세로축은 에너지 효율을 높이기 위한 특별한 기술을 사용하지 않은 일반적인 연관사상캐시에 대비한 상대적인 에너지 소비를 나타낸다. 그리고 가로 축은 SPEC CPU2K의 각 어플리케이션을 나타내고, 맨 오른쪽 끝에는 모든 어플리케이션에 대한 평균값이 표시되어 있다. Fig. 2의 A, B, C는 각각 레벨1 명령어캐시, 레벨1 데이터캐시, 그리고 레벨2 캐시에서의 결과를 보

인다. 이 그림은 이중웨이선택캐시, 웨이룩업캐시, 그리고 웨이추적캐시 모두가 일반적인 연관사상캐시에 비해 에너지 효율이 상당히 우수함을 보인다. 특히 이중웨이선택캐시는 캐시의 레벨에 무관하게 가장 높은 수준의 에너지 효율을 제공한다. 실제 모든 어플리케이션에 대한 평균값을 보면, 일반적인 연관사상캐시에 대비하여, 이중웨이선택캐시는 레벨1 명령어캐시에서 27.57%의 에너지를 소비하고, 레벨1 데이터캐시에서 28.42%의 에너지를 소비하며, 레벨2 캐시에서 15.41%의 에너지를 소비한다.

구체적으로 살펴보면, Fig. 2A의 레벨1 명령어캐시의 경우, 잘 알려진 바와 같이 웨이룩업캐시는 높은 지역성에 힘입어 웨이추적캐시에 비하여 높은 에너지 효율을 제공한다. 제안한 이중웨이선택캐시는 레벨1 명령어캐시에서 기존에 가장 우수하다고 알려진 웨이룩업캐시와 유사한 에너지 효율을 보여준다. 실제 데이터를 보면 이중웨이선택캐시 및 웨이룩업캐시에서의 상대적인 에너지 소모량은 각각 27.57% 및 27.88% 이다. 어플리케이션에 따라서는 웨이룩업캐시가 이중웨이선택캐시보다 근소하나마 우수한 에너지 효율을 보이는 경우도 있다. 그 이유는 이들 어플리케이션에서는 WLB의 적중률이 100%에 가까워 이중웨이선택캐시와 웨이룩업캐시의 차이가 거의 없는데, 이중웨이선택캐시는 WTT를 추가적으로 사용하여 이에 따른 에너지를 소모하기 때문이다. 따라서, 제안한 이중웨이선택캐시의 평균적인 에너지 효율이 근소하게 높기는 하지만 큰 차이가 아니기 때문에, 하드웨어 오버헤드를 고려하여 레벨1 명령어캐시에서는 이중웨이선택캐시 대신에 웨이룩업캐시를 사용하는 것도 타당하다. 레벨1 명령어캐시에서 웨이추적캐시의 상대적인 에너지 소모량은 36.97%로서 다른 두 구조에 비해 에너지효율이 떨어진다.

레벨1 데이터캐시에서는 레벨1 명령어캐시에 비하여 지역성이 떨어진다. 따라서 웨이룩업캐시는 레벨1 명령어캐시에 비해 레벨1 데이터캐시에서 그 에너지 효율이 감소한다. 이와는 달리, 제안한 이중웨이선택캐시는 지역성의 영향을 적게 받기 때문에 레벨1 데이터캐시에서 기존에 가장 우수하다고 알려진 웨이룩업캐시보다 에너지 효율이 높다. 레벨1 데이터캐시에서 이중웨이선택캐시, 웨이룩업캐시 및 웨이추적캐시의 상대적인 에너지 소모량은 각각 28.42%, 33.65% 및 41.98%이다. 마지막으로, 레벨2 캐시에서는 지역성이 매우 낮아 웨이룩업캐시의 에너지 효율이 크게 떨어지며 상대적으로 웨이추적캐시가 높은 에너지 효율을 보인다. 제안한 이중웨이선택캐시는 레벨2 캐시에서 가장 우수하다고 알려진 웨이추적캐시보다 높은 에너지 효율을 보인다. 레벨2 캐시에서 이중웨이선택캐시, 웨이룩업캐시 및 웨이추적캐시의 상대적인 에너지 소모량은 각각 15.41%, 47.26% 및 18.38%이다. 이 결과를 종합하면, 이중웨이선택캐시는 모든 레벨의 캐시에서 가장 우수한 수준의 에너지 효율을 제공한다는 것을 알 수 있다.

캐시에서의 에너지 소모량에 대하여 더 자세히 이해하기 위하여 각 캐시 구조의 에너지 소모량은 크게 "(일반적인) 캐시 접근을 위해 소비되는 에너지"와 "웨이선택기법에서

사용되는 에너지"로 나눌 수 있다. 이에 따라 실제 데이터를 이용하여 에너지 소모량을 세분화하면, 먼저 레벨1 명령어캐시의 경우, 웨이추적캐시에서의 상대적 에너지 소모량 36.97% 중에서 36.18%는 캐시에서 나머지 0.79%는 WTT에 의해 소모되었고, 웨이룩업캐시에서의 상대적 에너지 소모량 27.88% 중에서 26.44%는 캐시에서 나머지 1.44%는 WLB에서 사용되었으며, 마지막으로 이중웨이선택캐시에서의 상대적 에너지 소모량 27.57% 중에서 25.34%는 캐시에서 나머지 2.23%는 이중웨이선택기에서 소모되었다. 다음으로 레벨1 데이터캐시의 경우, 웨이추적캐시에서의 상대적 에너지 소모량 41.98% 중에서 41.19%는 캐시에서 나머지 0.79%는 WTT에 의해 소모되었고, 웨이룩업캐시에서의 상대적 에너지 소모량 33.65% 중에서 32.21%는 캐시에서 나머지 1.44%는 WLB에서 사용되었으며, 마지막으로 이중웨이선택캐시에서의 상대적 에너지 소모량 28.42% 중에서 26.19%는 캐시에서 나머지 2.23%는 이중웨이선택기에서 소모되었다. 마지막으로 레벨2 캐시의 경우, 웨이추적캐시에서의 상대적 에너지 소모량 18.38% 중에서 17.74%는 캐시에서 나머지 0.64%는 WTT에 의해 소모되었고, 웨이룩업캐시에서의 상대적 에너지 소모량 47.26% 중에서 45.99%는 캐시에서 나머지 1.27%는 WLB에서 사용되었으며, 마지막으로 이중웨이선택캐시에서의 상대적 에너지 소모량 15.41% 중에서 13.51%는 캐시에서 나머지 1.9%는 이중웨이선택기에서 소모되었다.

이중웨이선택캐시의 에너지 효율이 왜 우수한지에 대해 보다 자세히 이해하기 위하여, 본 연구에서는 매번 프로세서가 캐시 블록을 요청할 때 평균적으로 몇 개의 웨이를 접근하는지를 분석한다. Table 1은 SPEC CPU2K의 각 어플리케이션을 실행할 때, 매 캐시 블록 요청에 대해 평균적으로 접근한 웨이의 개수를 각 캐시 레벨에서 세 캐시 구조에 대하여 보여준다. Table의 맨 아래쪽 열에는 모든 어플리케이션들에 대한 평균값도 표시되어 있다. Table에 표시된 바와 같이 레벨1 캐시에는 모두 4개의 웨이가 있고 레벨2 캐시에는 모두 8개의 웨이가 있다. 이때, Table 1은 캐시 적중시와 더불어 캐시 적중실패 시 접근해야 하는 모든 웨이의 개수를 고려한 평균을 비교한 것이다. 예를 들면, n-웨이의 일반 연관사상캐시 또는 웨이예측캐시에서는 캐시 적중실패의 경우에도 n개의 모든 웨이를 접근해야 한다. 다시 말하면, 이들 캐시는 n개의 웨이를 접근한 후에야 캐시 적중실패라는 것을 알 수 있다. 웨이룩업캐시에서는 캐시 적중실패의 경우 실제 n-1개의 웨이를 접근한다[7]. 반면 이중웨이선택캐시나 웨이추적캐시에서는 WTT에 있는 정보만으로도 캐시 적중실패를 미리 판단할 수 있고, 캐시 적중실패라고 판단된 경우 웨이를 접근하지 않고 바로 아래 단계의 메모리를 접근한다. 따라서 이들 캐시가 캐시 적중실패 시 접근하는 웨이의 개수는 0개이다[8]. 물론 접근하는 웨이의 개수가 적을수록 에너지 효율 면에서 우수한 구조이고, 이중웨이선택캐시나 웨이추적캐시에서 캐시 적중실패의 경우 접근하는 웨이의 개수가 0개라고 하는 것은 이들 구조가 캐시 적중실패의 경우에도 다른 캐시들에 비해 에너지를 절약한

Table 1. Average number of ways accessed per cache access

	레벨1 명령어 캐시(4-웨이)				레벨1 데이터 캐시(4-웨이)				레벨2 캐시(8-웨이)			
	적중률 (%) 캐시/WLB	웨이추적 캐시	웨이록업 캐시	이중웨이 선택캐시	적중률 (%) 캐시/WLB	웨이추적 캐시	웨이록업 캐시	이중웨이 선택캐시	적중률 (%) 캐시/WLB	웨이추적 캐시	웨이록업 캐시	이중웨이 선택캐시
gzip	100.0/99.5	1.82	1.01	1.00	97.5/93.2	1.61	1.14	1.03	98.2/58.0	1.41	3.52	1.17
vpr	100.0/96.9	1.66	1.06	1.02	95.8/82.8	1.76	1.34	1.10	77.4/38.8	1.47	4.67	1.20
gcc	97.6/94.9	1.60	1.10	1.01	98.2/90.1	1.79	1.20	1.07	98.2/57.4	1.93	3.56	1.38
mcf	99.9/94.6	1.21	1.11	1.02	99.9/95.0	1.78	1.10	1.04	76.2/69.8	1.48	2.81	0.96
parser	100.0/99.2	1.52	1.02	1.00	98.4/92.0	1.73	1.16	1.05	77.1/46.5	1.30	4.21	1.08
gap	99.5/97.0	1.35	1.06	1.02	96.0/92.1	1.55	1.16	1.02	60.7/53.3	1.25	3.80	1.07
vortex	98.0/94.5	1.95	1.11	1.04	97.9/88.5	1.66	1.23	1.06	98.6/48.7	1.74	4.08	1.37
bzip2	100.0/98.4	1.01	1.03	1.00	96.6/90.6	1.80	1.19	1.05	78.9/43.6	1.54	4.38	1.20
twolf	99.5/94.2	1.50	1.12	1.02	99.9/94.4	1.67	1.11	1.03	99.0/82.9	1.41	2.03	1.06
ammp	100.0/97.7	1.00	1.05	1.00	89.4/81.2	1.46	1.38	1.02	50.0/50.0	1.18	4.00	0.91
art	100.0/100.0	1.00	1.00	1.00	64.7/59.1	1.2	1.82	0.92	16.9/16.4	0.84	6.02	0.78
equake	99.7/95.3	1.58	1.09	1.03	99.9/91.2	1.67	1.18	1.06	91.3/89.9	1.55	1.61	0.96
galgel	100.0/99.8	1.25	1.00	1.00	92.3/82.5	1.66	1.35	1.07	91.5/44.1	1.57	4.36	1.29
mesa	99.8/96.5	1.29	1.07	1.01	100.0/91.0	1.60	1.18	1.05	99.1/98.9	1.34	1.06	0.99
mgrid	100.0/99.0	1.60	1.02	1.00	94.8/61.4	1.40	1.77	1.08	47.1/29.0	1.07	5.26	0.88
평균	100.0/99.5	1.42	1.06	1.01	97.5/93.2	1.62	1.29	1.04	98.2/58.0	1.41	3.69	1.09

다는 것을 의미하며, 이것은 이들 구조의 장점 중의 하나이다. Table 1은 이렇게 캐시 적중실패의 경우를 고려하여 어플리케이션이 실행되는 동안 매 캐시블록 요청 시 접근해야 하는 모든 웨이 개수의 평균을 보여준다.

Table 1에 의하면, 제안한 이중웨이선택캐시가 매 캐시블록 요청에 대해 접근한 웨이의 평균적인 개수는 레벨1 명령어캐시, 레벨1 데이터캐시 및 레벨2 캐시에서 각각 1.01개, 1.04개, 1.09개 이다. 에너지 효율 측면에서 가장 이상적인 것은 하나의 웨이만을 접근하는 것이므로, 이 결과는 이중웨이선택캐시의 특성이 모든 캐시 레벨에서 이상적인 수치에 상당히 근접함을 보인다. 기존에 레벨1 캐시에서 가장 우수하다고 알려진 웨이록업캐시가 매 캐시블록 요청에 대해 접근하는 웨이의 평균적인 개수는 레벨1 명령어캐시 및 데이터캐시에서 각각 1.06개, 1.29개로서, 이는 이중웨이선택캐시의 그것들에 비해 높은 수치이다. 또한 기존에 레벨2 캐시에서 가장 우수하다고 알려진 웨이추적캐시가 접근하는 웨이의 평균적인 개수는 1.41개로서, 이 역시 이중웨이선택캐시의 그것에 비해 높은 수치이다. 이러한 결과가 얻어지는 이유는 이중웨이선택캐시가 웨이추적캐시와 웨이록업캐시의 웨이선택기법들을 함께 사용하면서, 매번 프로세서가 캐시블록을 요청할 때 접근하는 웨이의 개수를 최소화하는

방향으로 동작하기 때문이다.

Table 1에서 만약 캐시 적중실패 시 접근하는 웨이의 개수를 고려하지 않는다면, 이상적인 에너지 효율을 갖는 캐시는 항상 1개의 웨이에 접근할 것이다. 그러나 Table 1에서는 캐시 적중실패 시 접근하는 웨이의 개수를 고려하였고, 이때 이중웨이선택캐시는 전체적으로 캐시 적중시의 에너지 효율이 이상적인 캐시에 가장 근접한데다가 캐시 적중실패에는 0개의 웨이를 접근하기 때문에, 실제 몇몇 어플리케이션에서는 접근하는 웨이의 평균 개수가 1.0보다 작은 경우가 발생하게 된다. 이때 매 캐시블록 요청 시 접근하는 웨이의 평균 개수가 1.0보다 작다는 것은 그만큼 이중웨이선택캐시가 에너지 효율성 측면에서 우수하다는 것을 보여준다. 마지막으로 Table 1은 매 캐시블록 요청 시 접근하는 웨이의 평균 개수와 함께 각 어플리케이션에서의 WLB의 적중률 및 캐시 적중률을 보여준다. WLB 적중률은 웨이록업캐시와 이중웨이선택캐시에만 해당되며, 레벨1 명령어캐시와 레벨1 데이터캐시 그리고 레벨2 캐시에서의 지역성을 반영한다. Table 1의 세 캐시 구조는 웨이선택의 방식이 다를 뿐, 각 캐시의 동작 및 캐시에 들어 있는 데이터의 내용은 일반 연관사상캐시와 동일하다. 따라서 Table 1의 세 캐시 구조 및 일반 연관사상캐시에서의 캐시 적중률은 같다.

Table 2. Access latency and hardware overhead

	레벨1 캐시		레벨2 캐시	
	접근지연시간 (ns)	하드웨어 오버헤드 (%)	접근지연시간 (ns)	하드웨어 오버헤드 (%)
일반 연관사상캐시	1.46		5.01	
웨이룩업캐시	1.69	1.03	5.43	0.45
웨이추적캐시	1.70	0.75	5.55	1.52
이중웨이선택캐시	1.70	1.79	5.55	1.96

Table 2는 일반적인 연관사상캐시와 웨이룩업캐시, 웨이추적캐시 및 이중웨이선택캐시의 접근지연시간과 일반적인 연관사상캐시에 대비한 하드웨어 오버헤드를 보여준다. 먼저 접근지연시간에 대하여 살펴보면, 일반 연관사상캐시의 접근은 구체적으로 태그 접근과 데이터 접근으로 구분되며 태그 접근과 데이터 접근은 병렬로 수행된다. 이때 태그 접근지연시간이 데이터 접근지연시간에 비해 상대적으로 짧기 때문에, Table 2의 일반 연관사상캐시의 접근지연시간은 데이터 접근지연시간에 의해 결정된다. 웨이룩업캐시는 WLB를 접근한 후에 실제 캐시의 웨이로의 데이터 접근이 일어나기 때문에 전체 접근지연시간이 늘어나며, Table 2에서 일반 연관사상캐시와 웨이룩업캐시의 접근지연시간의 차이는 WLB를 접근하기 위한 시간을 나타낸다. 마찬가지로 일반 연관사상캐시와 웨이추적캐시의 접근지연시간의 차이는 웨이추적캐시에서 WTT에 접근하기 위한 시간을 나타낸다. 마지막으로 이중웨이선택캐시에서는 WLB와 WTT를 병렬로 접근하며, 실제 WLB의 접근지연시간과 WTT의 접근지연시간이 유사하기 때문에, 이중웨이선택캐시의 접근지연시간은 웨이룩업캐시와 웨이추적캐시의 접근지연시간은 거의 같다.

또한 Table 2는 일반 연관사상캐시의 하드웨어 면적에 대비하여 각 캐시 구조에서 사용하는 웨이선택기법에 의해 추가되는 하드웨어 오버헤드를 보여준다. 예를 들어, 웨이룩업캐시는 일반 연관사상캐시에서 WLB를 추가적으로 사용하기 때문에, 웨이룩업캐시를 구현하기 위해서는 일반적인 연관사상캐시를 구현하는데 필요한 하드웨어 면적의 1.0103배가 필요하다. 먼저 웨이룩업캐시에서 WLB는 캐시의 각 인덱스 값에 대한 MRR 웨이 정보 및 태그 정보만을 기록하므로, WLB의 크기는 캐시의 연관성이 커질수록 상대적으로 작아진다. 따라서 Table 2에서 보듯이 웨이룩업캐시의 상대적 하드웨어 오버헤드는 4-웨이 캐시인 레벨 1캐시에서는 1.03%이지만, 8-웨이 캐시인 레벨 2캐시에서는 0.45%로서 레벨1 캐시에 비해 약 반으로 감소한다. 웨이추적캐시의 경우, 하나의 WTT는 캐시에 존재하는 모든 블록에 대한 Valid 비트의 집합으로써 그 크기가 캐시의 크기에 비례하지만, 실제 웨이추적캐시는 캐시의 연관성과 동일한 개수의 WTT를 사용하므로, 웨이추적캐시에서의 전체 하드웨어 오

버헤드는 연관성이 커질수록 상대적으로 증가한다. 따라서 Table 2에서 보듯이 웨이추적캐시의 경우 상대적 하드웨어 오버헤드가 4-웨이 캐시인 레벨 1캐시에서는 0.75%이지만, 8-웨이 캐시인 레벨 2캐시에서는 1.52%로서 레벨1 캐시에 비해 약 2배 증가한다. 마지막으로 이중웨이선택캐시의 하드웨어 오버헤드는 대략적으로 웨이룩업캐시와 웨이추적캐시 각각에서 발생하는 오버헤드의 합이 된다. 실제 이중웨이선택캐시의 하드웨어 오버헤드는 약 2% 정도로, 이 정도의 오버헤드는 실제 이중웨이선택캐시의 응용에 있어서 제한요소로 작용하지 않을 것으로 판단된다.

6. 결 론

웨이룩업캐시와 웨이추적캐시는 연관사상캐시에서의 에너지 효율을 높이기 위한 기술로써, 웨이룩업캐시는 레벨1 캐시에서 가장 우수한 수준의 에너지 효율을 제공하고, 웨이추적캐시는 레벨2 캐시에서 가장 우수한 에너지 효율을 제공한다. 본 논문에서는 웨이룩업캐시와 웨이추적캐시의 웨이선택기법을 함께 사용하여 두 기법의 장점을 결합함으로써, 모든 레벨의 캐시에서 가장 높은 수준의 에너지 효율을 제공하는 이중웨이선택캐시를 제안하였다. 제안한 이중웨이선택캐시의 에너지 효율은 SPEC CPU2K 어플리케이션을 이용하여 시뮬레이션을 통해 평가하였다.

Alpha 21264 프로세서의 예를 이용한 시뮬레이션 결과에 따르면, 이중웨이선택캐시는 레벨1 명령어캐시에서 에너지 효율을 높이기 위한 특별한 기술을 사용하지 않은 일반적인 연관사상캐시에 비해 27.57%의 에너지를 소비하여, 웨이룩업캐시와 같은 수준의 에너지 효율을 제공한다. 그리고 이중웨이선택캐시는 레벨1 데이터 캐시에서 일반적인 연관사상캐시에 비해 28.42%의 에너지를 소비하며, 이는 웨이룩업캐시에 비해 에너지 소비가 15.54% 감소한 것이다. 또한 이중웨이선택캐시는 레벨2 캐시에서 일반적인 연관사상캐시에 비해 15.41%의 에너지를 소비하며, 이는 웨이추적캐시에 비해 에너지 소비가 16.16% 감소한 것이다. 이 결과는 제안한 이중웨이선택캐시가 캐시의 레벨에 무관하게 가장 높은 수준의 에너지 효율을 제공한다는 것을 보인다. 이러한 결과가 얻어지는 이유는 이중웨이선택캐시가 웨이추적캐시와 웨

이룩업캐시의 웨이선택기법들을 함께 사용하면서, 매번 프로세서가 캐시 블록을 요청할 때 접근하는 웨이의 개수를 최소화하는 방향으로 동작하기 때문이다. 시뮬레이션 결과에 의하면, 제안한 이중웨이선택캐시가 매 캐시 블록 요청에 대해 접근한 웨이의 평균적인 개수는 레벨1 명령어캐시, 레벨1 데이터캐시 및 레벨2 캐시에서 각각 1.01개, 1.04개, 1.09개로서, 모든 캐시 레벨에서 이상적인 수치인 1에 상당히 근접함을 보인다. 마지막으로 이중웨이선택캐시는 접근 지연시간 측면에서 웨이추적캐시 및 웨이룩업캐시와 유사한 특성을 보이며, 이중웨이선택캐시의 구현을 위한 하드웨어 오버헤드는 실제 응용에 있어 제한요소가 되지 않는 정도로 판단된다.

참 고 문 헌

- [1] D. A. Patterson, John L. Hennessy, Computer architecture: a quantitative approach, 3rd Ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
- [2] D. Brooks, V. Tiwari, M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," Proc. of ISCA, Vancouver, British Columbia, Canada, pp.83-94, 2000.
- [3] J. Montanaro, et.al., "A 160 MHz, 32b 0.5W CMOS RISC microprocessor," IEEE Journal of Solid-State Circuits, Vol.31, No.11, pp.1703-1714, November, 1996.
- [4] K. Inoue, T. Ishihara, K. Murakami, "Way-predicting set-associative cache for high performance and low energy consumption," Proc. of ISLPED, San Diego, CA, USA, pp.273-275, 1999.
- [5] A. Hasegawa, I. Kawasakai, K. Yamada, S. Yoshioka, S. Kawasaki, P. Biswas, "SH3: high code density, low power," Proc. of MICRO, Ann Arbor, Michigan, USA, pp.11-19, 1999.
- [6] D. Nicolaescu, A. Veidenbaum, A. Nicolau, "Reducing power consumption for high-associativity data caches in embedded processors," Proc. of DATE, Munich, Germany, pp.1064-1068, 2003.
- [7] S. Lee, J. Kang, I. Lee, "Way-lookup buffer for low-power set-associative cache," IEICE Electronics Express, Vol.8, No.23, pp.1961-1966, December, 2011.
- [8] J. Kang, S. Lee, I. Lee, "Way-tracking set-associative caches," IEE Electronics Letters, Vol.46, No.22, pp.1497-1499, October, 2010.
- [9] D. Burger, T. Austin, "The simplescalar tool set version 2.0," Technical Report 1342, University of Wisconsin, Madison, June, 1997.
- [10] SPEC 2000 Benchmark. <http://www.spec.org>.
- [11] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "CACTI 6.0: A tool to model large caches," Technical Report, HPL-2009-85, HP Laboratories, April, 2009.



이 성 재

e-mail : sjlee@csl.hanyang.ac.kr

2003년 한양대학교 전자전기공학부(학사)

2008년 한양대학교 전자컴퓨터통신공학과

(공학석사)

2008년~현재 한양대학교 전자컴퓨터

통신공학과 박사과정

관심분야: 컴퓨터 아키텍처, 멀티프로세서, 저전력 시스템 등



강 진 구

e-mail : jinkukang@hanyang.ac.kr

2005년 순천대학교 전자공학과(학사)

2007년 한양대학교 전자컴퓨터통신공학과

(공학석사)

2012년~현재 한양대학교 전자컴퓨터

통신공학과 박사과정

관심분야: 컴퓨터 아키텍처, 멀티프로세서, 저전력 시스템 등



이 주 호

e-mail : racin@hanyang.ac.kr

2011년 한양대학교 전자통신컴퓨터공학부

(학사)

2012년~현재 한양대학교 전자컴퓨터

통신공학과 석사과정

관심분야: 프로그래밍 언어, 컴퓨터 교육 등



윤 지 용

e-mail : gojilyong@gmail.com

2012년 한양대학교 컴퓨터공학부(학사)

2012년~현재 한양대학교 전자컴퓨터

통신공학과 석사과정

관심분야: 프로그래밍 언어, 컴퓨터 교육 등



이 인 환

e-mail : ihlee@hanyang.ac.kr

1979년 서울대학교 전기공학과(학사)

1986년 서울대학교 전기공학과(석사)

1994년 University of Illinois at Urbana-

Champaign(박사)

1997년~현재 한양대학교

컴퓨터공학부 교수

관심분야: 컴퓨터 아키텍처, Fault-tolerant computing 등

에너지 효율이 높은 이중웨이선택형 연관사상캐시

이 성 재^{*} · 강 진 구^{*} · 이 주 호^{**} · 윤 지 용^{**} · 이 인 환^{***}

요 약

본 논문은 레벨1 캐시에서 가장 우수한 수준의 에너지 효율을 제공하는 웨이룩업캐시와 레벨2 캐시에서 가장 높은 에너지 효율을 제공하는 웨이추적캐시의 장점을 결합하여, 모든 레벨의 캐시에서 가장 높은 수준의 에너지 효율을 제공하는 이중웨이선택캐시를 제안한다. Alpha 21264 프로세서의 예를 이용한 시뮬레이션 결과에 따르면, 이중웨이선택캐시는 레벨1 명령어캐시에서 일반적인 연관사상캐시에 비해 27.57%의 에너지를 소비하여, 웨이룩업캐시와 같은 수준의 에너지 효율을 제공한다. 그리고 이중웨이선택캐시는 레벨1 데이터 캐시에서 일반적인 연관사상캐시에 비해 28.42%의 에너지를 소비하며, 이는 웨이룩업캐시에 비해 에너지 소비가 15.54% 감소한 것이다. 또한 이중웨이선택캐시는 레벨2 캐시에서 일반적인 연관사상캐시에 비해 15.41%의 에너지를 소비하며, 이는 웨이추적캐시에 비해 에너지 소비가 16.16% 감소한 것이다.

키워드 : 연관사상캐시, 에너지효율, 이중웨이선택캐시