

데이터 캐시의 선택적 프리차지를 통한 에너지 절감

최 병 창[†] · 서 효 중^{††}

요 약

최근 디지털 시스템 응용의 복잡성이 증가하면서, 고성능화와 전력 소모 절감은 중요한 문제로 대두되고 있다. 전력 소모를 줄이기 위한 방법으로, 프로세서의 저 전력화 또한 미세 공정의 개발과 함께 다양하게 접근되고 있으며, 이러한 결과 모바일 시스템의 시장 확대가 이루어지고 있다. 본 논문은 이러한 프로세서에서 소모되는 전력 중 데이터 캐시의 전력 손실을 줄이기 위하여 데이터 캐시의 웨이 중 필요로 하는 부분을 예측하여 선택적으로 프리차지 하도록 하는 전력 소모 개선 방법을 제안하였으며, 접근하는 웨이에 대한 예측이 성공했을 경우에는 에너지를 절감하면서 지연이 발생하지 않는 빠른 접근이 이루어지게 하고, 실패하였을 경우에도 최소한의 시간 지연으로 해당되는 웨이의 접근이 이루어지게 하였다. 시뮬레이션을 통한 검증 결과 웨이 예측 기법만을 사용한 경우에 대비하여 평균 10.2%의 전력을 절감할 수 있었으며, 일반적 캐시에 비하여는 평균 56.4%의 전력을 절감할 수 있었다.

키워드: 데이터 캐시, 저 전력, 캐시 웨이 선택, 프리차지

Low-power Data Cache using Selective Way Precharge

Byeong-Chang Choi[†] · Hyo-Joong Suh^{††}

ABSTRACT

Recently, power saving with high performance is one of the hot issues in the mobile systems. Various technologies are introduced to achieve low-power processors, which include sub-micron semiconductor fabrication, voltage scaling, speed scaling and etc. In this paper, we introduce a new method that reduces of energy loss at the data cache. Our methods take the benefits in terms of speed and energy loss using selective way precharging of way prediction with concurrent way selecting. By the simulation results, our method achieves 10.2% energy saving compared to the way prediction method, and 56.4% energy saving compared to the common data cache structure.

Keywords: Data Cache, Low-Power, Cache Way Select, Precharge

1. 서 론

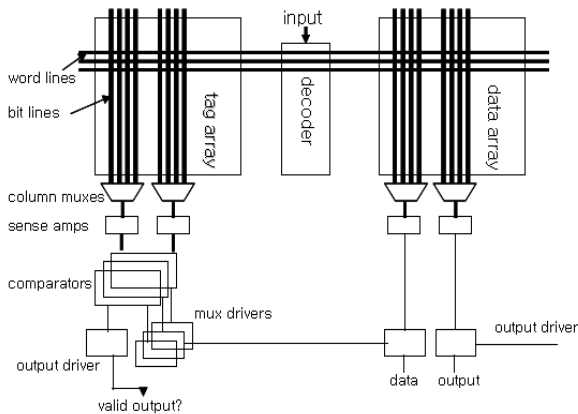
정보 통신 기술의 발달에 따라 유비쿼터스 환경이 확대되어 모바일 기기는 보다 더 고성능화 되어가고 있다. 이에 따라 모바일 기기도 또한 확대일로에 있으며 따라서 저 전력을 위한 연구가 다양하게 진행되고 있다. 모바일 기기에 내장된 마이크로프로세서의 내부 장치에서, 전력소모의 부분은 크게 프로세서 코어 부분과 캐시 부분으로 나누어 볼 수 있으며, 캐시가 소모하는 전력이 크게는 전체의 50% 정도를 차지하는 경우[1]도 있어, 캐시 메모리의 성능과 에너지 소모는 모바일 기기의 성능 향상에 큰 관심사 중 하나로 자리 잡아 왔다[2]. 이러한 캐시 메모리가 소모하는 에너

지 비중이 상대적으로 높기 때문에, 프로세서의 에너지 소모를 줄이는 방법의 하나로서 캐시 메모리가 소모하는 에너지를 줄이는 것은 매우 효율적인 방법이며 이에 따라 캐시 메모리의 에너지 소모를 줄이기 위한 많은 연구가 진행되고 있다.

(그림 1)은 전통적인 캐시 구조이다[3]. 그림과 같이 캐시는 태그 어레이와 데이터 어레이로 나뉘어 구성되며, 캐시 메모리는 일반적으로 SRAM(Static RAM) 구조로 캐시에서 소모하는 에너지는 데이터를 유지하는 정적 상태에서 기본적으로 소모하는 정적 에너지와 캐시에 대한 액세스가 발생할 경우 액세스를 위하여 소모하는 동적 에너지 소모로 분리해 볼 수 있다. 캐시에서 데이터를 읽어 오기 위해서는 우선 태그 어레이와 데이터 어레이 모두 비트 라인을 통해 프리차지(precharge)시켜 데이터를 빠르게 읽어 올 수 있는 상태로 만들어야 하며, 이러한 작업은 동적 에너지 소모에 포함된다. 그리고 주소 값의 인덱스 값을 디코딩하여 워드

[†] 준 회원: 가톨릭대학교 컴퓨터공학과 석사과정

^{††} 종신회원: 가톨릭대학교 컴퓨터정보공학부 조교수(교신저자)
논문접수: 2007년 8월 21일
수정일: 1차 2008년 5월 29일, 2차 2008년 12월 4일
심사완료: 2008년 12월 15일



(그림 1) 전통적인 캐시 구현

라인을 통해 해당 위치의 데이터 액세스가 이루어진다. 태그 어레이의 값은 주소 값의 태그 부분과 비교기(comparator)를 통해 비교를 하고 그 결과 값에 따라 데이터 중 유효한 데이터를 출력하게 된다. 이 때 비트라인이 프리차지된 데이터 어레이의 웨이 중 실제로 액세스가 이루어질 하나만 사용하게 되어 나머지 부분에 대한 프리차지한 만큼 에너지의 심각한 낭비가 발생하게 된다.

따라서 이와 같이 에너지 손실이 발생하는 단점을 보완하기 위해 필요로 하는 부분에서만 에너지를 공급하기 위하여 여러 가지 방법이 제시되었다. 대표적으로 순차 접근 캐시 방법[4]이 그 한가지로, 순차 접근 캐시는 캐시 메모리에서 태그 부분과 데이터 부분을 분리해서 순차적으로 접근하는 방법으로 먼저 태그 값을 비교하고 그 결과 값으로 해당 데이터가 있는 웨이를 알아내서, 해당 웨이만 프리차지하여 불필요한 에너지 소비를 줄이는 기법이다. 그러나 태그 비교와 데이터 읽기를 순차적으로 진행하기 때문에 추가적인 사이클이 소모되는 단점이 있다. 또 다른 방법으로 캐시 웨이 예측 기법[4]이 있으며, 이 방법은 명령어 프로그램 카운터 값 또는 로드 소스 레지스터와 명령어의 오프셋을 배타적 논리합(XOR) 시킨 결과를 이용하여 예측 테이블을 작성하는 방식이 있다. 예측이 성공할 경우 추가 지연이 발생하지 않으며 전력 소모를 절감할 수 있으나, 예측이 틀린 경우에는 모든 웨이에 에너지를 공급하게 되어, 데이터를 읽어오기 위한 추가적인 지연과 에너지가 손실되는 단점이 있다.

본 논문은 이와 같은 각 단점을 보완하는 새로운 저 전력 데이터 캐시 메모리를 설계하는 방법을 제안한다. 논문에서 제안하는 방법은 선택적 웨이만 프리차지하는 방법을 취하고, 웨이 선택에 대하여 예측 방법을 통합함으로써 예측한 웨이가 적중할 경우 지연이 발생하지 않으며 필요한 웨이만 프리차지 하도록 동작시키고, 예측이 실패할 경우 순차 접근 캐시와 동일한 지연시간 내에서 필요한 웨이에만 프리차지 함으로써 캐시 적중 시간의 단축과 손실 에너지를 제거한 방법이다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구에 대해서 서술하며, 3장에서는 본 논문에서 제안하는 기법을

이용한 캐시 메모리 구조와 정책을 기술하고, 4장에서는 제시된 기법에 대한 실험과 결과에 대해서 논하며, 끝으로 5장에서는 결론을 제시한다.

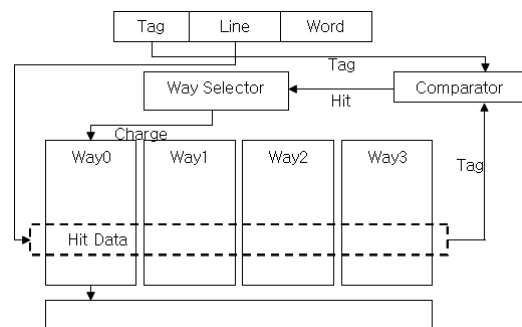
2. 관련 연구

본 장에서는 제안하는 캐시 구조의 기본이 되는 집합 연관 캐시 구조에서 에너지를 절감할 수 있는 기법들을 소개한다. 이후 해당 기법들의 장단점 분석을 통해 본 논문에서 제안한 캐시 구조를 제안하게 된 동기를 설명한다.

E. Witchel은 소프트웨어적인 방법을 이용해서 캐시에 접근할 때 태그를 비교하는 작업을 줄이는 기법을 제안하였다 [5]. 이 방법은 프로그램을 컴파일 할 때 캐시 메모리 상에서 데이터가 위치할 장소를 미리 선택하고, 명령어 중 특정 비트들을 사용하여 태그 비교 없이 캐시 메모리에서 데이터를 읽어 올 수 있게 했다. 이 방식은 태그를 비교할 때 소모되는 에너지를 줄이고 좀 더 빠른 속도로 캐시에 접근할 수 있도록 하는 장점이 있으나, 프로그램과 시스템 구성의 복잡도가 증가하고 태그 비교를 줄여서 얻을 수 있는 에너지 소모의 절감 효과가 상대적으로 작다는 단점이 있다.

C. L. Su와 A. M. Despain은 캐시 메모리의 동적 에너지 소모를 줄이기 위한 기법인 블록 버퍼링과 서브 बैं킹을 일반적인 캐시에서의 에너지 소모와 비교하는 실험을 하였다 [6]. 블록 버퍼링은 데이터 어레이에서 데이터를 읽을 때는 블록 단위로 데이터를 읽는데, 이때 읽은 블록을 블록 버퍼에 기록해 두었다가 다음 접근 때 빠르게 접근하는 것이 가능하도록 만든 기법이다. 따라서 에너지 소모 절감 효과와 빠른 접근이 가능한 방법이다. 이 기법은 데이터의 지역성과 연계하여 블록 내에 연속된 데이터를 연속적으로 읽어들이는 경우에 매우 효과적인 방법이다.

순차 접근 캐시는 캐시의 모든 웨이에 공급한 전력 중 실제 접근이 일어난 부분을 제외한 나머지 부분에 대한 손실을 줄이기 위하여 사용된 방법으로 (그림 2)와 같이 웨이에 전력을 공급하기 전에 먼저 태그 값을 비교하여 해당 데이터가 존재하는 웨이를 알아내고, 해당 웨이에 전력을 공급하는 방법이다. 그러나 태그 비교와 웨이에 전력을 공급하는 과정을 순차적으로 진행하게 되므로 태그의 비교와 모든



(그림 2) 순차 접근 캐시

웨이에 대한 전력 공급을 동시에 진행하는 일반적인 캐시에 비해 추가적으로 한 사이클을 더 소모하게 되는 단점이 있다[4].

선택적 직접 사상 캐시는 충돌에 의한 캐시 실패가 발생하지 않은 데이터에 대한 정보를 가지는 테이블을 별도로 유지하여 해당 데이터를 직접 사상 캐시에 위치하도록 하고, 특정 데이터에 대해서만 선택적으로 직접 사상 캐시를 사용하게 한 것이다[4].

또한 캐시 셀에 대하여 값을 유지할 수 있는 최소한의 에너지만을 공급하여 대기 전력을 줄인 방법으로 드라우지(drowsy) 상태를 이용한 캐시도 연구된 바 있으며[7] 드라우지 캐시 방법과 캐시 웨이 예측 기법을 하나로 통합하여 대기 전력과 동작 전력을 각각 줄인 방법도 연구된 바 있다[8]. 이 방법은 드라우지 캐시에서 캐시를 활성화 시키는 깨움(awake) 시간을 으로 발생하는 추가적 지연 사이클을 줄이기 위하여 웨이 예측을 통하여 깨움 작업을 수행하게 한 것이다. 예측이 성공하였을 경우 한 사이클이 소모되며, 실패하였을 경우 추가적으로 한 사이클이 더 소모되게 된다.

최근에는 위와 같은 방법과 더불어 동적 전압제어(Dynamic Voltage Scaling)[9]과 가변 클럭을 병행하여 캐시의 전력 소모를 줄이기 위한 방법이 제안되었으며, 이 방법은 프로세서 내의 블록과 일차 및 이차 캐시 블록을 별개의 클럭 영역으로 분할하여, 프로세서 코어와 캐시에 대해 요구 하는 성능만큼으로 동작 속도를 가변 제어하면서 동시에 동작 속도를 늦추어 전력 소모를 줄이는 방법 또한 제시되었다[10].

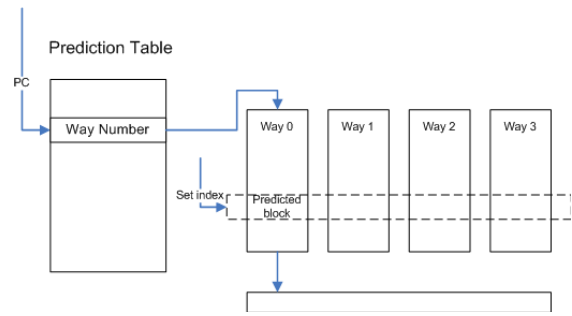
3. 저 전력 데이터 캐시 설계

3.1 저 전력 데이터 캐시 구조

3.1.1 웨이 예측기의 구조

캐시 웨이 예측 기법에는 (그림 3)과 같이 명령어 프로그램 카운터(PC) 값을 이용하는 방식과 로드 소스 레지스터와 명령어의 오프셋을 배타적 OR(XOR) 시킨 결과를 이용하여 예측 테이블을 작성하는 방식이 있다[4]. 프로그램 카운터를 이용하여 예측하였을 경우, 명령어를 읽어 들이는 단계에서 프로그램 카운터를 알 수 있으므로, 이를 위한 예측 결과가 빠르게 도출되나, 상대적으로 정확성이 조금 떨어지게 되고, 로드 소스 레지스터를 이용하는 방법의 경우 정확성은 높지만 명령어의 해독 이후에 레지스터를 액세스한 후에 알 수 있으므로, 해당 웨이를 구하는 타이밍이 느린 단점이 있다.

다음 (그림 3)은 프로그램 카운터를 이용한 웨이 예측기를 도시한 것이다. 프로그램 카운터에 따라 예측 테이블(prediction table)에서 웨이의 번호가 선택되며, 이 번호에 해당하는 웨이 에 미리 전력을 공급하게 된다. 예측이 성공한 경우는 바로 데이터를 읽을 수 있지만, 예측이 실패한 경우에는 미리 전력을 공급한 웨이를 제외한 나머지 웨이에 전력을 공급하고 캐시를 비교하는 등의 추가적인 작업으로 인한 지연이 생기게 된다.



(그림 3) 명령어 프로그램 카운터 기반의 예측 기법

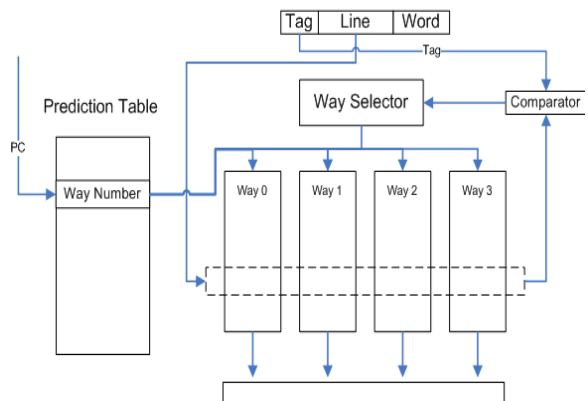
3.1.2 제안하는 캐시 구조

(그림 4)는 본 논문에서 제안하는 캐시 웨이 예측 기법과 순차 접근 캐시의 장점을 취하여 통합 설계한 구조이다. 일반적인 집합 연관 캐시와는 달리 웨이 선택기(Way selector)가 부가되며, 예측기에서 보내온 웨이 번호나 태그 비교를 통해서 찾아낸 해당 웨이에 전력을 공급하는 역할을 수행한다. 우선 이 웨이 선택기는 예측기를 사용하여 해당 웨이에만 미리 전력을 공급하여 캐시에 대한 빠른 접근을 가능하도록 하며, 예측이 적중한 경우에 지연 없이 바로 해당 웨이가 액세스 된다. 예측기에 의한 선택이 실패한 경우, 액세스 되는 주소의 태그 영역과 해당되는 태그가 비교되며, 웨이 선택기에 의해 해당 웨이만 활성화되어 액세스 된다.

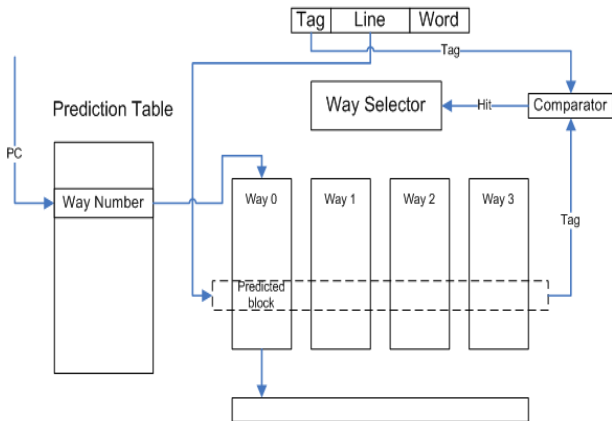
이러하게 예측기와 선택기를 모두 이용한 통합을 함으로써 예측기의 실패의 경우 낭비되는 전력 소모를 절감하고, 웨이 선택기에 의해 유발되는 한 사이클의 지연을 예측기를 통하여 최소화 한 것이다.

3.2 저 전력 데이터 캐시의 동작 과정

본 절에서는 예측기를 중심으로 적중 여부에 따른 저 전력 데이터 캐시의 동작 과정에 대해서 살펴본다. 본 논문에서 제안하는 저 전력 데이터 캐시는 캐시에 접근하기 전에 명령어 프로그램 카운터 값을 이용하여 예측 테이블에서 해당 웨이 값을 구하는 작업이 우선적으로 수행되고, 캐시에 접근 하였을 때 바로 읽을 수 있는 상태로 만든다. 이러한



(그림 4) 통합 캐시 구조

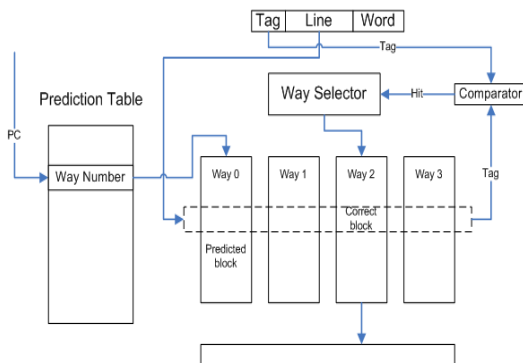


(그림 5) 예측이 적중한 경우

작업과 동시에 데이터 캐시에 접근하는 순간 데이터 캐시에서는 태그 값들을 읽어서 원하는 데이터가 어떤 웨이에 존재하는지 파악하는 작업을 수행한다. (그림 5)는 명령어 프로그램 카운터의 값을 이용한 예측이 적중한 경우를 나타내고 있다. 예측기를 통해 예측된 웨이에 전력을 공급하고, 캐시에 접근했을 때 한 사이클 만에 원하는 데이터를 읽어올 수 있게 된다. 따라서 순차 접근 캐시를 사용했을 경우 두 사이클이 걸리는 것과 비교하여 한 사이클을 절약하는 효과를 얻을 수 있다.

(그림 6)은 예측기의 예측이 틀린 경우를 나타내고 있다. 예측이 틀린 경우는 앞서 서술한 바와 같이 태그의 값이 일치하지 않은 경우 발생하게 된다. 이러한 경우 태그의 비교 결과에 의하여 해당되는 웨이를 알아내게 되며, 해당되는 웨이에 전력을 공급하여 활성화 하는데 한 사이클의 지연이 추가되게 된다. 이 경우 예측기의 결과와 관계없이 해당 웨이에만 전력을 공급하게 된다. 앞서 제안된 예측 기법의 적용 경우는 예측기가 실패했을 때 예측한 웨이를 제외한 나머지 모든 웨이에 전력을 공급하게 되나 제안된 캐시 구조는 이러한 전력 손실이 일어나지 않게 된다.

<표 1>은 n개의 웨이를 가지는 집합 연관 캐시의 경우와 비교하여 본 논문에서 제안하는 방법의 장점을 나타낸 것이다. 예측기가 적중하였을 때는 예측 기법만 사용한 경



(그림 6) 예측이 틀린 경우

<표 1> 각 기법과 제안하는 방법의 사이클과 활성화 웨이 수

	예측 기법		순차 접근 캐시		제안하는 데이터 캐시	
	사이클	활성화 웨이	사이클	활성화 웨이	사이클	활성화 웨이
적중	1	1	2	1	1	1
실패	2	n		1	2	2

우와 동일한 성능을 나타내게 되어 순차 접근에 비하여 사이클의 이득이 있고, 예측 기법이 실패한 경우는 순차 접근에 비하여 1 개의 웨이를 더 활성화 하게 되나 예측 기법은 n개의 웨이를 모두 활성화 하게 된다. 결국 최악의 경우 순차 접근 캐시 보다는 많은 전력을 소모하지만 예측 기법만 사용한 경우 보다는 에너지가 절약할 수 있으며, 예측의 적중 정도에 따라서 사이클 손실을 줄일 수 있다.

제안하는 데이터 캐시를 예측 기법만이 사용된 경우와 비교할 때, 일반적으로 캐시의 웨이 수에 대한 전력소모 경향은 웨이의 증가에 따라 예측 기법만을 사용한 경우 예측의 성공률이 저하될 수 있으나, 캐시의 적중률이 상승하는 경향을 보이게 될 것이다. 반면 예측 실패로 인한 에너지 손실은 더 커지게 된다. 따라서 예측 기법만을 사용한 경우의 적절한 웨이의 수는 프로그램의 참조 패턴에 따라 달라진다. 그러나 제안하는 데이터 캐시는 웨이의 증가에 따라 예측 실패에 따르는 에너지 손실이 증가하지 않고 일정하므로 상대적 장점을 가지게 된다.

순차 접근 캐시에 대비하여 제안하는 데이터 캐시는 예측기의 적중시 한 사이클의 시간 이득이 있는 반면, 예측기를 동작시키기 위한 만큼의 전력소모가 부가된다. 그러나 예측기는 프로그램 카운터의 값 또는 참조 레지스터의 값에 따라서 데이터 값을 출력하는 선형 메모리이므로, 이러한 전력 소모와 순차 접근 캐시의 사이클 지연에 따르는 프로세서 전체적인 정적 에너지 손실분과의 비교는 구현에 따라 고려해 보아야 한다.

캐시의 웨이 수와 관련된 에너지 소비 등 관련된 여러 사항은 다음 장에서 보다 자세히 살펴본다.

4. 성능 평가

4.1 실험 환경

실험은 Suse Linux 9.3이 설치된 컴퓨터를 사용하여 진행하였고, 제안한 시스템을 구현하기 위해 모의실험 도구인 SimpleScalar[11] 시뮬레이터에 기존의 구조인 예측기, 순차 접근 캐시 및 제안하는 데이터 캐시를 구현하여 사용하였다. 또한 캐시에서 소모되는 전력을 좀 더 정확하게 측정하기 위해 CACTI4.0[12]을 사용하고 0.13um를 기준으로 하여 전력 캐시의 각 소자별 전력 소모량을 구하였다. 모의실험

<표 2> 모의실험 인자

Parameter	Value
L1 I-cache	32KB, 4-way, 32 byte block, 1 cycle latency
L1 D-cache	32KB, 4, 8, 16 way, 32 byte block, 1 cycle latency
L2 cache	Unified cache, 4-way, 256KB, 64byte block, 8 cycle latency
Memory	64 cycle latency
Prediction Table	64, 128, 256, 512, 1024 entry

<표 3> 벤치마크 프로그램

SPEC2000			
SPEC INT 2000		SPEC FP 2000	
164.gzip	Compression	177.mesa	3-D Graphics
175.vpr	FPGA Place Route	179.art	Image Recognition
176.gcc	C Compiler	181.mcf	Combinatorial Optimization
197.parser	Word Processing	183.equake	Seismic Wave Propagation
255.vortex	Database	188.ammp	Chemistry

에 사용된 캐시는 명령어 캐시의 경우 4개의 웨이를 설정하였고, 본 논문의 대상인 데이터 캐시의 경우 웨이 수를 4, 8, 16개인 경우를 각각 실험하여 기존의 방법과 제안하는 방법을 측정하였다. 캐시 실패의 경우, 메인 메모리에 접근하는 시간은 64사이클이 소모되는 것으로 설정하였으며 기타 자세한 모의실험 인자는 <표 2>와 같다.

실험에 사용된 벤치마크 프로그램으로는 SPEC2000[13]에서 SpecINT2000의 벤치마크 프로그램 중 5개와 SpecFP2000의 5개를 선택하여 사용하였으며, 이러한 벤치마크 프로그램은 <표 3>과 같다. 프로그램은 gcc 컴파일러의 최적화 O2 레벨로 컴파일 하여 실험을 수행하였다.

4.2 실험 결과 및 분석

본 장에서는 제안한 저 전력 데이터 캐시와 기존 캐시 구조의 성능을 비교하여, 제안하는 캐시 구조가 어느 정도의 소모 에너지 절감 효과를 가지는지 기술한다.

<표 4>는 웨이 수에 따른 정적 전력 소모량을 나타내고

<표 4> 웨이 수에 따른 정적 전력 소모

웨이 수	정적 전력 소모(mW)
4	69.90
8	72.54
16	89.25

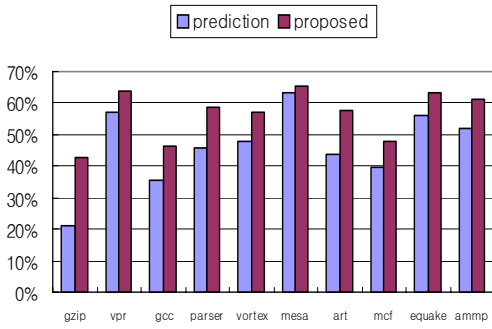
있다. 4웨이에서 8웨이로 증가할 때는 그 차이가 그리 크지 않지만 16웨이로 증가하면서 정적 전력 소모량이 확연히 증가하고 있다. 그 이유는 멀티플렉서 드라이버에 있다. 멀티플렉서 드라이버가 4웨이일 때는 0.035mW의 전력을 소모하고 8웨이일 때는 0.038mW의 전력을 소모하는데 반해 16웨이일 때는 9.928mW의 정적 전력이 소모된다. 웨이 수가 증가할수록 캐시 적중률은 증가할 것이나, 예측기의 적중률이 낮아지게 되고 정적 전력 소모가 상대적으로 커지게 되므로, 이후의 결과에서는 적절한 정적 전력 소모율을 나타내는 8웨이일 때를 기준으로 결과를 비교하였다. 캐시의 웨이 및 용량에 따라서 정적 전력 소모는 급격히 증가한다[14].

4.2.1 제안하는 구조의 에너지 절감

바람직한 시스템 모델의 조건은, 요구되는 최소한의 기능 및 처리 속도를 보장하면서 동시에 최소한의 에너지 소비를 통해 보유하고 있는 제한된 에너지 량으로 시스템이 가능한 장시간 운용될 수 있도록 하는 것이라 할 수 있다. 따라서 제안하는 저 전력 데이터 캐시 구조는 에너지 소모량과 처리 속도를 성능 평가의 기준으로 한다.

캐시 기법 수준의 추상화 수준에서 바라볼 때, 제안하는 데이터 캐시 구조는 프로그램 카운터를 이용한 웨이 예측기와 순차 접근 캐시를 변형하여 복합적으로 적용한 형태로 정의 할 수 있다. 따라서 각각의 저 전력 캐시 구조를 단일하게 적용했을 때와 제안하는 캐시 구조를 적용했을 때를, 바람직한 내장형 시스템 모델의 두 가지 요소에 따라 비교하면 적절한 측면에서 제안하는 저 전력 데이터 캐시 구조의 성능 파악이 가능할 것이다.

(그림 7)은 일반적인 데이터 캐시에 대비하여, 웨이 예측 기법만을 사용한 경우와 본 논문에서 제안하는 데이터 캐시 구조를 사용했을 경우의 에너지 소모 절감율을 비교한 것이다. 웨이 예측 기법을 사용한 경우 예측기의 부가에 따르는 전력 또한 포함하였다. 예측기의 엔트리 수는 512개로 하였고 데이터 캐시는 8웨이로 구성하여 실험을 진행하였다. ‘prediction’은 웨이 예측기만을 사용한 경우이고, ‘proposed’는 제안하는 데이터 캐시 구조를 사용한 경우이다. 제안하는 구조는 예측기가 실패하는 경우 활성화 웨이의 개수에서 큰 차이가 나타나게 되므로, 전체적으로 큰 에너지 절감을 나타내었다. 최대 격차를 보인 gzip에서는 약 22% 차이를 보였고, 최소 격차를 보인 mesa에서는 약 4%의 차이를 나타냈다. 예측 기법과 제안하는 방법 사이의 차이의 전체 평균은 약 10.2%로 제안하는 구조가 보다 뛰어난 에너지 절감을 나타냈음을 알 수 있다.



(그림 7) 웨이 예측기와 제안하는 구조의 에너지 절감율

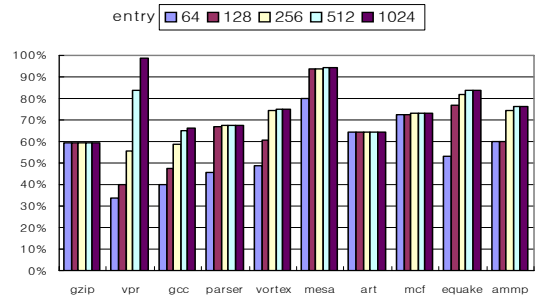
순차 접근방식은 본 논문에서 제안한 방식과 대비하여 한 사이클의 지연을 추가로 필요로 하나, 에너지 소모에 있어서는 논문에서 제안한 구조와 예측기의 실패 경우에 있어서 웨이의 부가적 구동의 차이가 되어 제안한 구조와 유의한 에너지 소모 차이가 나타나지 않았다.

제안하는 데이터 캐시 구조는 프로그램 카운터를 이용한 웨이 예측기를 통해 해당 웨이를 먼저 예측하고 웨이 예측기 작동과 동시에 태그 비교를 수행하여 작업 요청을 처리하게 된다. 따라서 제안하는 데이터 캐시 구조는 웨이 예측기를 가동하기 위해 소모된 에너지와 순차 접근 캐시 구조에서 소모되는 에너지의 합만큼 에너지를 소모 하게 된다. 예측기를 통한 예측이 적중했을 경우에는 예측기만 사용한 캐시 구조에서 웨이 선택기를 구동하기 위한 약간의 에너지 소모가 더해지게 된다.

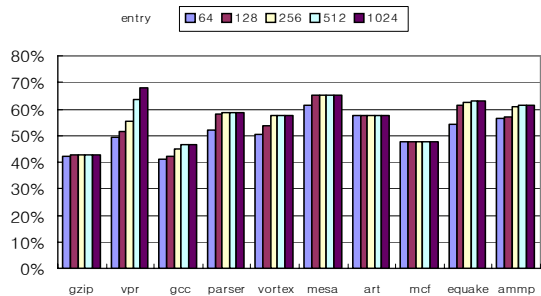
4.2.2 예측기의 적중률과 데이터 캐시에서의 에너지 소모

SPEC2000중 10개의 벤치마크 프로그램에서 데이터 캐시는 8웨이로 구성하고 예측기의 엔트리 수는 64, 128, 256, 512, 1024개 일 때의 각각의 예측기의 적중률을 알아보았다. 결과는 (그림 8)과 같다. gzip, art, mcf의 경우 예측기 엔트리 수의 변화가 적중률에 크게 영향을 주지 않았고, vpr과 gcc의 경우는 엔트리 수가 증가하는 것에 비례하여 적중률이 상승하였다. 그 외에는 예측기의 엔트리 수가 증가할수록 적중률이 소폭으로 상승하였다. 예측기의 엔트리 크기가 512일 때 평균 적중률이 74.2%인 것으로 나타났다.

다음 (그림 9)는 일반적 데이터 캐시 메모리에 대하여 제안한 구조의 엔트리 크기 변화에 따른 에너지 소모 절감율 나타내고 있다. vpr과 gcc의 경우 예측기의 엔트리 크기에 따라 에너지 소모 절감 정도가 민감하게 반응하고 있고, gzip, art, mcf 등은 엔트리 크기에 크게 영향을 받지 않는 것을 알 수 있다. 이 같은 결과는 앞의 (그림 8)의 예측기 적중률 그래프와 유사한 결과를 나타내고 있다. 이 실험을 통해 예측기의 적중률이 데이터 캐시 메모리에서의 에너지 소모 절감 정도에 큰 영향을 주는 것을 알 수 있었다. 제안한 방법을 사용했을 경우 데이터 캐시에서 평균 56.4% 정도의 에너지 소모를 절감할 수 있었다.



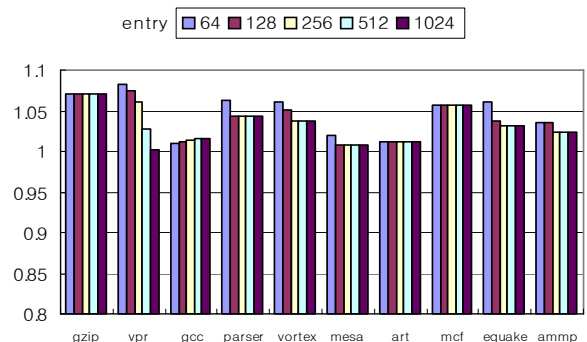
(그림 8) 예측기의 적중률



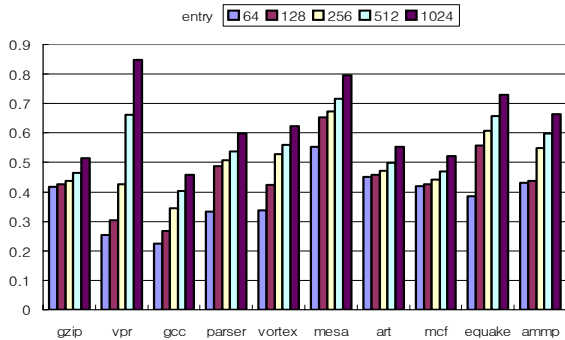
(그림 9) 데이터 캐시 메모리에서의 에너지 소모 절감

4.2.3 수행 시간 비교

본 논문에서 제안하는 데이터 캐시 메모리의 구조는 예측기의 예측이 성공했을 경우에는 수행 시간의 손실이 없으나, 예측이 틀렸을 경우, 추가적인 사이클이 소모되어 성능이 감소한다. (그림 10)은 일반적인 데이터 캐시에 대비하여 예측기의 엔트리 크기에 따른 프로그램의 실행시간 비율을 나타낸다. vpr과 gcc는 적중률에서 나타난 바와 마찬가지로 성능 감소 폭이 예측기의 엔트리 크기에 민감하게 반응하고 있으며, gzip과 art, mcf 등은 (그림 9)와 일치하는 결과를 나타냈다. 나머지 프로그램 들은 예측기의 엔트리 크기가 커질수록 성능 차이가 줄어들어 가는 것을 알 수 있다. 이 같은 패턴은 예측기의 실패에 따른 사이클 지연이 성능감소에 영향을 주고 있는 것을 알 수 있다. 예측기의 엔트리 크기가 512일 때 평균 성능이 1.032배 로 낮아졌음을 알 수 있다.



(그림 10) 실행 시간 비율



(그림 11) 일반적 캐시 대비 Energy-Delay Product의 비율

4.2.4 Energy-Delay Product

제안한 방법은 예측기의 적중률이 높아짐에 따라 캐시 메모리에 의하여 발생하는 지연 손실이 줄어들게 되어 에너지 소모의 이득이 발생할 수 있으나 예측기와 선택기로 인한 에너지 소모가 추가적으로 발생한다. 반면 일반적인 캐시에 대비하여 접근이 필요한 웨이에 대해서만 프리차지 함으로써 캐시를 구동하는데 필요로 하는 에너지를 줄이게 된다. 다음 (그림 11)은 일반 캐시에 대비하여 제안한 구조의 Energy Delay Product(EDP)의 비를 제시한 것이다.

캐시의 에너지 소모적 측면으로 볼 때, 예측기 entry의 증가는 수행시간의 이득으로 인한 개선과 예측기 자체에서 소모하는 정도의 손실이 상반되게 나타날 수 있다. 선택적 precharge에 의하여 전체적으로 개선을 나타내기는 하였으나 일정 크기 이상의 예측기는 효율이 떨어질 수 있음을 나타내고 있다. 이는 제안한 예측기와 선택기를 모두 갖춘 구조에서도 에너지 효율을 최대한으로 끌어내기 위해서는 적절한 크기의 예측기 크기가 고려되어야 함을 의미한다.

5. 결 론

모바일 시스템이 확대되는 것과 발맞추어 프로세서의 에너지 소모를 줄이는 것은 매우 중요한 핵심 기술이다. 미세공정의 발전과 고성능 프로세서의 개발은 자연스럽게 보다 더 큰 캐시 메모리로 나타나게 되었으며, 캐시 메모리의 크기가 커질수록 캐시 메모리의 전력 소모 절감은 필연적이다.

데이터 캐시의 전력 소모를 절감하기 위한 캐시 웨이 예측 기법은 부가적인 지연 없이 웨이 예측기를 이용해 예측된 웨이만을 활성화 시켜서 불필요한 에너지를 줄여줄 수 있으나, 예측이 틀렸을 경우에는 추가적인 사이클과 많은 에너지가 손실되는 단점이 있으며, 순차 접근 방식은 적합한 웨이를 알아낸 후 해당 웨이를 활성화하여 에너지 소모를 줄일 수 있으나, 항상 추가적인 사이클이 소모되었다.

본 논문에서 제안한 방법은 캐시 웨이 예측 기법과 순차 접근 캐시의 장점만을 취하고, 각 방법의 단점을 최소한의 부가장치인 웨이 선택기를 부가하여 전력 소모를 줄이고 이에 따르는 지연을 최소화하는 방법을 제안하였다. 10개의

벤치마크 프로그램을 사용하여 실험을 진행한 결과 웨이 예측 기법만 사용한 경우보다 통합 설계 방식을 적용한 경우가 평균 10.2% 정도의 에너지 소모를 더 절감할 수 있었다. 또한 일반적인 데이터 캐시 구조를 사용한 경우에 비해 평균 1.032배 정도의 성능이 감소하였으나, 데이터 캐시 메모리에서 평균 56.4%의 에너지 소모를 절감할 수 있어, 제한된 전력을 사용하는 모바일 시스템에 적용하기에 적합한 결과를 얻었다. 차후 연구로 본 논문에서 제안한 방법에 동적 전압 방법(DVS)과 가변 클럭을 적용한 기법을 병행하는 방법을 연구 중이다.

참 고 문 헌

- [1] Kin, J., M. Gupta, M.-Smith, W.H., "The filter cache: an energy efficient memory structure", In Proc. Microarchitecture(MICRO-97), pp.184-193, 1997.
- [2] C. Zhang, F. Vahid, W. Najjar, "A highly configurable cache architecture for embedded systems", Int'l Symp. Computer Architecture, pp.136-146, Jun., 2003.
- [3] D. A. Patterson, J. L. Hennessy, Computer Architecture: A Quantitative Approach, 3ed, Morgan Kaufmann.
- [4] M. Powell, A. Agarwal, T. N. Vijaykumar, B. Falsafi, K. Roy, "Reducing set-associative cache energy via way-prediction and selective direct-mapping", Int'l Symp. Microarchitecture, pp.54-65, Dec., 2001.
- [5] E. Witchel, S. Larsen, C. S. Ananian, K. Asanovic, "Direct addressed caches for reduced power consumption", Proc. 34th ACM/IEEE Int'l Symp. Microarchitecture, pp.124-135, Dec., 2001.
- [6] Ching-Long Su, A. M. Despain, "Cache designs for energy efficiency", Int'l Conf. System Sciences Architecture, Vol.1, pp.306-315, Jan., 1995.
- [7] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, T. Mudge., "Drowsy caches: simple techniques for reducing leakage power", Int'l Symp. Computer Architecture, pp.148-157, Jul., 2002.
- [8] 김성훈, 김철홍, 장성태, 전주식, "고성능 내장형 프로세서의 에너지 소비 감소를 위한 데이터 캐시 통합 설계 방법", 정보과학회논문지, 33권, 3호, pp.166-177, 2006년 4월.
- [9] G. Magklis, G. Semeraro, D.H. Albonese, S.G. Dropsho, S. Dwarkadas and M.L. Scott, "Dynamic Frequency and Voltage Scaling for a Multiple-Clock-Domain Microprocessor", IEEE Micro, Vol.23, No.6, pp.62 - 68, 2003.
- [10] N. AbouGhazaleh, B. Childers, D. Mosse, R. Melhem, "Integrated CPU Cache Power Management in Multiple Clock Domain Processors", Lecture Notes in Computer Science, Vol.4917/2008, pp.209-223, 2008.
- [11] T. Austin, E. Larson, D. Ernst, "SimpleScalar: an infrastructure for computer system modeling", IEEE

Computer, pp.59-67, Feb., 2002.

[12] D. Tarjan, S. Thoziyoor, N. P. Jouppi, CACTI 4.0, HP Laboratories, Technical Report HPL-2006-86, Jun., 2006.

[13] SPEC CPU2000 Benchmarks, <http://www.specbench.org>.

[14] D. Kudithipudi, E. John, "Parametrical characterization of leakage power in embedded system caches using gated-VSS", Proc. IASTED Int'l Conf. Circuits, Signals, and Systems, pp.308-312, 2005.



최 병 창

e-mail : hellocbc@catholic.ac.kr

2005년 가톨릭대학교 컴퓨터정보공학과
(학사)

2005년~현 재 가톨릭대학교 컴퓨터공학과
석사과정

관심분야 : 내장형시스템, 컴퓨터 구조



서 효 중

e-mail : hjsuh@catholic.ac.kr

1992년 서울대학교(학사)

1994년 서울대학교 컴퓨터공학과
(공학석사)

2000년 서울대학교 컴퓨터공학과
(공학박사)

2002년 (주)지씨티 리서치 책임연구원

2003년~현 재 가톨릭대학교 컴퓨터정보공학부 조교수

관심분야 : 컴퓨터 구조, 컴퓨터 시스템, 내장형시스템