

논문 2007-44CI-1-1

임베디드 프로세서를 위한 선인출 데이터캐시의 저전력화 방안

(Reducing Power Consumption of Data Caches for Embedded Processors)

문 현 주*, 지 승 현**

(Hyun Ju Moon and Sung Hyun Jee)

요 약

임베디드 프로세서는 총 에너지소모량 가운데 대략 40% 이상을 캐시에서 소모하고 있으므로 에너지-효율적 고성능 데이터 캐시 구조를 필요로 한다. 본 논문에서는 임베디드 프로세서를 위한 저전력 선인출 데이터캐시 구조를 제안하였다. 제안한 데이터캐시 구조는 선인출장치(prefetching unit)를 포함한 기존 데이터캐시 구조에 태그히스토리 테이블(tag history table)을 구비함으로써 요구인출 및 선인출시 발생하는 태그메모리 병렬탐색 횟수를 감소시켰다. 이와 같은 전략적인 캐시 구조는 적은 하드웨어 비용으로 병렬탐색을 위한 전력소모를 현저히 줄일 수 있다. 실험을 통하여 제안한 데이터캐시 구조가 기존 선인출 데이터캐시 구조와 동일한 성능을 유지하면서 낮은 전력을 요구함을 확인하였다.

Abstract

Since data caches used in modern embedded processors consume significant fraction of total processor power up to 40%, embedded processors need power-efficient high performance data caches. This paper proposes a prefetching data cache structure which pursuing low power consumption. We added tag history table on existing data cache structure which includes hardware unit for data prefetching so that reduce the number of parallel lookup on tag memory. This strategic cache structure remarkably reduces power consumption for parallel tag lookup. Experimental results show that the proposed cache architecture induce low power consumption while maintain the same cache performance.

Keywords : embedded processor, data cache, low-power, data prefetching, tag history table

I. 서 론

최근 임베디드 프로세서는 멀티미디어 및 유비쿼터스 분야를 지원하는 다양한 모바일 컴퓨팅 시스템의 휴대용 기기에서 주로 사용되고 있다. 모바일 분야에 사용되는 임베디드 프로세서는 배터리에 의존적이므로 데이터참조에 따른 실행지연이나 집중적인 전력소모 시에 전력한계 문제를 자주 발생시킨다.^[1] 임베디드 프로세서

는 사용하는 총 에너지소모량 가운데 많은 부분을 캐시에서 소모하고 있으며 예를 들어 StrongARM-1의 경우 총 에너지소모량 중 캐시참조로 인한 에너지소모량이 43%를 차지한다.^[2] 따라서 많은 연구자들이 임베디드 프로세서에 적합한 에너지-효율적 고성능 캐시구조를 설계하여 시스템의 성능향상과 에너지 소모량을 줄이는 저전력 문제를 해결하는 연구를 진행하고 있다.

인텔(Intel)을 포함한 임베디드 프로세서 설계자들은 임베디드 프로세서의 성능향상을 위하여 캐시설계 시에 데이터 선인출(data prefetch) 기법을 채용하고 있다.^[3-6] 선인출 기법은 캐시적중률을 높이고 CPU 대기시간을 단축시켜서 전체 성능을 향상시키는 반면, 선인출을 위한 별도의 에너지를 필요로 하므로 에너지효율 면에서는 성능향상에 저해가 된다.

임베디드 프로세서의 저전력을 구현하기 위한 캐시

* 정희원, 남서울대학교 컴퓨터학과
(Department of Computer Science, Namseoul University)

** 정희원, 한국고용정보원 정보화전략팀
(Information Strategy Team, Korea Employment Information Service)

※ 이 논문은 2006년 남서울대학교 학술연구비 지원에 의하여 연구되었음.

접수일자: 2006년7월6일, 수정완료일: 2007년1월4일

설계는 캐시접근(cache access) 시에 태그(tag)를 예측함으로써 태그메모리에서의 태그에 대한 병렬탐색(parallel lookup)을 최소화하는 방향으로 활발히 진행되고 있다. 이러한 연구는 캐시참조시에 지역성(locality)이 높다는 관측을 토대로 한다. 즉 앞으로 참조할 캐시라인 주소는 최근에 참조된 캐시라인 주소와 동일하거나 근접한 위치에 있다는 특성을 활용하는 것이다. 태그를 예측하는 다양한 연구들과 달리 영역결정장치(WDU: Way Determination Unit) 기법은 전에 사용한 태그(tag) 정보와 영역(way) 정보를 저장하는 태그히스토리 테이블(tag history table)을 이용하여 참조할 태그를 결정하는 방법이다. 즉, 태그히스토리 테이블에 저장된 태그정보로부터 태그메모리의 해당 영역(way)정보를 검색하고 자료를 곧장 읽을 수 있다. 이와 같이 영역결정장치 기법은 병렬탐색 없이 캐시를 참조하므로 병렬탐색에 필요한 전력을 현저하게 감소시킬 수 있다.^[7]

본 논문에서는 지금까지의 관측을 바탕으로 하여 임베디드 프로세서에 적합한 데이터캐시 구조를 제안하였다. 제안한 데이터캐시 구조는 기존 데이터캐시에 선인출(prefetch) 장치와 태그히스토리 테이블(tag history table)을 구비함으로써 캐시적중률을 높이며 캐시참조시 태그메모리에 대한 병렬탐색 횟수를 감소시켰다. 이러한 전략적인 데이터캐시 구조는 캐시적중률을 높여 시스템 성능을 향상시킬 뿐 아니라 태그메모리의 병렬탐색 횟수를 줄여서 전력소모를 현저히 줄일 수 있다.

본 논문의 제 II장에서는 임베디드 프로세서를 위한 기존 캐시연구를 비교하였다. 제 III장에서는 저전력 선인출 데이터캐시 구조를 설계하고, 제안하는 캐시구조에서의 데이터참조 과정과 동작 메커니즘을 설명하였다. 제 IV장에서는 다양한 벤치마크 프로그램들을 이용하여 저전력 선인출 데이터캐시 구조, 기존 선인출 데이터캐시 구조, 그리고 일반 데이터캐시 구조에 대한 실험을 수행하여 성능향상과 에너지소모율을 비교·분석하였다. 마지막으로 제 V장에서는 논문의 결론을 맺고 향후 계획을 기술하였다.

II. 본 론

1. 임베디드 프로세서용 데이터캐시 구조

인텔을 포함한 임베디드 프로세서 설계자들은 캐시를 설계할 때 캐시적중률을 높여서 시스템의 성능을 향상시키는 방안으로 데이터 선인출 기법을 채용하고 있다. 데이터 선인출 기법은 앞서 인출된 데이터를 처리

하는 동시에 미래에 사용될 데이터를 미리 인출함으로써 응용 프로그램의 수행시간에 대한 메모리참조시간을 줄이는 것이다.^[8] 이 기법은 선인출 명령을 발생시키는 시점을 기준으로 정적 선인출 기법과 동적 선인출 기법으로 구분되는데, 동적 선인출 기법은 단순한 참조 패턴에 대한 선인출에서 우수한 성능을 발휘하며 실행 사이클의 증가를 유발하지 않아 멀티미디어 응용프로그램과 같이 스트리밍 패턴으로 데이터를 참조하는 경우에 적용하기 적합하다. 동적 선인출 기법의 대표적인 예로는 블록참조(OBL: One Block Lookahead) 기법^[3], 스트림버퍼(Stream Buffer) 기법^[5], 동적참조예측기(RPT: Reference Prediction Table) 기법^[6]이다. 이들 기법 중 동적참조예측기 기법은 멀티미디어 응용 프로그램의 수행에 가장 우수한 성능을 나타낸다.^[6] 동적참조예측기 기법은 참조예측표를 포함하는 참조예측기를 메모리 시스템에 추가하고 반복해서 참조하는 메모리정보를 저장 및 갱신하면서 미래에 참조될 메모리주소를 예측한다. 선인출의 장점에도 불구하고, 캐시에서의 데이터 선인출은 참조할 메모리주소가 캐시 내에 있는 지를 미리 참조하고 후에 실제로 캐시를 참조하므로 기존 캐시참조 횟수의 두 배가 필요하다. 그러므로 데이터 선인출은 캐시적중률을 높여서 시스템의 성능을 향상시키는 반면, 별도의 전력을 필요로 하므로 저전력 측면에서는 저해요소가 될 수 있다.

임베디드 프로세서에서의 저전력을 구현하기 위하여, 캐시 설계자는 캐시접근 시에 태그메모리에서 병렬탐색을 최소화하는 연구를 진행하고 있다. 즉, 기존 캐시참조는 태그메모리내 한 집합(set) 내 모든 태그들을 병렬로 검색해야 하지만, 병렬탐색 최소화를 통한 캐시참조는 필요한 태그를 예측하여 해당 태그정보만을 검색한다. 태그예측 연구는 단계캐시(phased cache) 기법^[9], 영역예측(way-prediction scheme) 기법^[10] 등이 있다. 이 방법들은 태그를 예측하는 장점을 지나나 태그예측을 위한 하드웨어 비용과 소프트웨어 비용이 크고 잘못 예측할 경우 발생하는 패널티(penalty)도 고려해야 한다. 그에 비하여, 다른 태그예측 연구인 영역결정장치(WDU: Way Deterministic Unit) 기법^[7]은 태그히스토리 테이블(tag history table)을 구비함으로써 태그를 예측하지 않고 결정한다. 즉, 영역결정장치 기법은 태그히스토리 테이블에 저장된 태그정보를 이용하여 태그메모리의 해당 영역(way)을 참조하여 자료를 읽는다. 따라서 이 기법은 태그예측에서 발생할 수 있는 패널티도 고려할 필요가 없으며 타 기법과 비교할 때 최소한의

하드웨어비용으로도 전력소모를 줄일 수 있다.

지금까지 캐시관련 연구는 시스템의 성능향상 혹은 낮은 전력소모에 관한 한 분야의 연구에만 집중되었다.^[13-16] 본 논문에서는 위에서 제시한 두 가지 목표를 만족하는 임베디드 프로세서를 위한 저전력 선인출 데이터캐시 구조를 제안하였다. 제안한 저전력 선인출 데이터캐시 구조는 기존 데이터캐시의 선인출 기법과 태그메모리의 병렬탐색 최소화 기법을 적용하여 시스템의 전체 성능을 향상시킨다.

2. 저전력 선인출 데이터캐시 패러다임

기존 선인출 데이터캐시는 참조할 데이터가 캐시 내에 존재하는지를 알기 위하여 캐시참조마다 태그메모리에 대한 병렬탐색을 수행한다. 반면에 태그히스토리 정보를 활용한 선인출 데이터캐시는 태그히스토리 테이블에 저장된 최근 참조한 태그정보를 통하여 참조할 데이터가 캐시 내 어느 영역(way)에 존재하는지를 병렬탐색 없이 알 수 있다. 이상의 동작은 캐시참조시 태그메모리에 대한 병렬탐색 횟수를 최소화하여 낮은 전력소모율을 가지도록 한다. 이후부터 본 논문에서는 제안한 선인출 데이터캐시 구조를 WR-캐시(WDU-RPT Cache), 기존 선인출캐시 구조를 R-캐시(RPT-Cache), 기존 캐시구조를 T-캐시(Traditional Cache)라고 명한다.

가. WR-캐시 구조 설계

본 논문에서 제안한 WR-캐시 구조는 그림 1과 같이 동적참조예측기(RPT), 영역결정장치(WDU), 캐시컨트롤러(cache controller), 및 캐시메모리로 구성된다. 여기서 동적참조예측기는 데이터 선인출을 수행하기 위하여 이전에 참조한 주소 값으로부터 규칙성을 발견하여 다음에 참조할 메모리주소를 미리 계산한다. 영역결정장치는 최근에 참조된 태그정보와 영역(way)정보를 저장하는 태그히스토리 테이블을 구비하여 캐시참조시 테이블에 저장된 태그정보를 이용하여 태그메모리의 해당 영역(way)정보를 결정토록 한다. 이 때 데이터캐시의 높은 지역성으로 인하여, 태그히스토리 테이블은 적은 엔트리(entry) 개수로도 빠른 액세스와 낮은 에너지 오버헤드를 얻는다. 지금까지 영역결정장치에 관련한 연구에서도 엔트리수가 8~16개인 경우가 비용 대 에너지감소를 측면에서 가장 효과적인 에너지감소효과와 성능향상을 가진다.^[7] 캐시컨트롤러는 참조예측기로부터 전달된

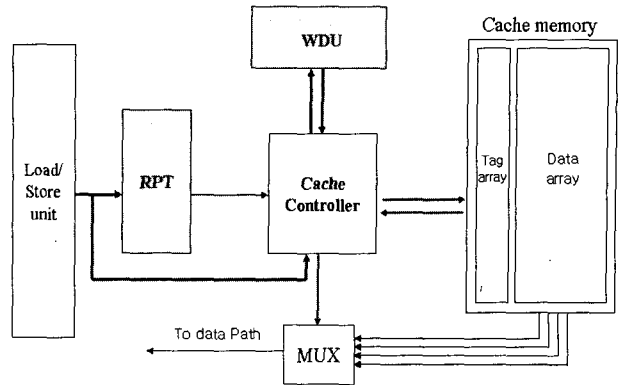


그림 1. WR-캐시(WDU-RPT Cache)구성도
Fig. 1. WR-cache architecture hierarchy.

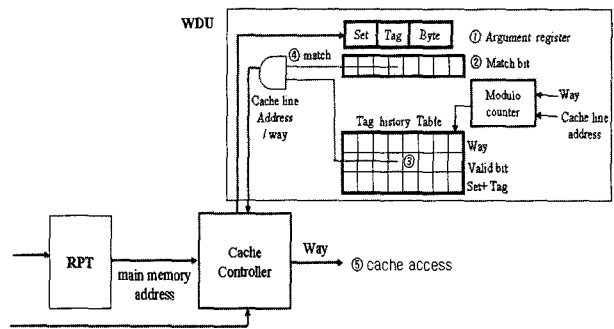


그림 2. 영역결정장치(WDU)의 태그탐색
Fig. 2. Tag lookup of way determination unit.

메모리주소를 캐시라인주소로 변환하여 주소전달 및 장치간 동작을 관리한다.

그림 2는 영역결정장치(WDU)에서 태그히스토리 테이블을 이용한 태그탐색 과정을 보여 준다. 태그히스토리 테이블은 집합(set)과 태그(tag), 영역(way) 및 유효비트(valid bit)로 이루어진 항목(entry)들로 구성된다. 각 항목은 최근 사용빈도를 나타내는 모듈러 카운터(modulo counter)를 가지고 있다. 만약 태그히스토리 테이블에서 사용할 수 있는 엔트리가 없다면 영역결정장치는 LRU(Least Recently Used) 기법으로 가장 오랫동안 사용하지 않은 항목을 선택하여 내용을 교체한다. 이때 태그히스토리 테이블은 완전연관캐시(fully associative memory)로 구성되어 빠른 태그탐색을 수행한다.

영역결정장치의 동작 메커니즘은 먼저 캐시컨트롤러는 동적참조예측기(선인출시) 또는 로드/스토어장치(요구인출시)로부터 전달받은 주기억장치 주소를 캐시라인 주소로 변환하여 인수레지스터(argument register)에 저장시킨다(①). 인수레지스터 내용(집합, 태그)과 태그히스토리 테이블의 엔트리가 서로 일치(적중)하고(②) 유효비트가 설정되면(③), 찾고자 하는 태그정보가 영역결정장치에 저장되어 있음을 의미하므로 캐시컨트롤러

에게 영역(way) 정보를 전달한다(④). 캐시컨트롤러는 영역(way)정보를 이용하여 태그메모리에 대한 병렬탐색 없이 정보를 읽어온다(⑤).

WR-캐시 구조는 태그히스토리 테이블에 유용한 태그정보가 많을수록, 캐시참조시 태그메모리에 대한 병렬탐색 횟수도 줄어든다. 만약 태그히스토리 테이블에 일치하는 태그정보가 없을 경우, 캐시컨트롤러는 기존 캐시참조와 동일하게 태그메모리에 대한 병렬탐색을 수행하여 원하는 자료를 읽어온다. 태그히스토리 테이블은 완전연관사상 캐시를 이용하여 용이하게 구현할 수 있으며, 태그히스토리 테이블의 항목수가 집합 연관 사상 캐시의 연관도보다 적은 경우에는 태그 병렬탐색에 비하여 적은 전력으로 캐시 참조를 수행할 수 있다. 뿐만 아니라, 태그히스토리 테이블을 CAM-tag 구조의 완전연관사상 캐시로 구현함으로써 소모 전력을 더욱 줄일 수 있다.^[7] 따라서 WR-캐시 구조는 태그히스토리 테이블을 이용하여 병렬탐색 횟수를 최소화할 수 있으므로 시스템의 전체 전력소모율은 낮추고 메모리참조 시간도 줄이는 장점을 지닌다. 그에 반해, T-캐시와 R-캐시는 캐시참조시마다 태그메모리에 대한 병렬탐색을 수행하므로 전력소모율이 높다.

나. WR-캐시 구조의 캐시참조 흐름

WR-캐시 구조는 그림 3과 같이 세 가지 경우(case1, case2, case3)중 한 경우를 통하여 원하는 정보를 읽을 수 있다. case1은 WDU탐색(WDU check)이다. 즉, 캐시컨트롤러로부터 캐시라인 주소의 태그정보와 태그히스토리 테이블의 태그정보가 일치하면, 캐시참조를 위한 태그정보를 결정한다((WDU hit). 그런 다음 캐시컨트롤러는 캐시참조시 태그정보를 이용하여 태그메모리에 대한 병렬탐색 없이 원하는 정보를 읽어 온다. 만약 영역결정장치에서 동일한 태그가 발견되지 않으면 캐시컨트롤러에게 그 사실을 알린다(WDU miss).

WDU미스가 발생한 경우, 캐시컨트롤러는 case2의 Cache탐색(Cache check)을 수행한다. 즉, 태그메모리를 병렬로 검색하여 해당 캐시라인 주소가 데이터캐시에 존재하는 지를 체크한다. 만약 원하는 정보가 캐시에 존재한다면, 해당 정보 및 영역(way)과 태그정보는 캐시컨트롤러에게 전달된다. 캐시컨트롤러는 그 정보를 영역결정장치에 전달하여 태그히스토리 테이블에 저장한다. case3는 MM탐색(Main Memory Check)으로 Cache탐색에서 캐시미스(cache miss)가 발생되어 원하는 데이터블록을 메모리로부터 데이터캐시로 가져온다.

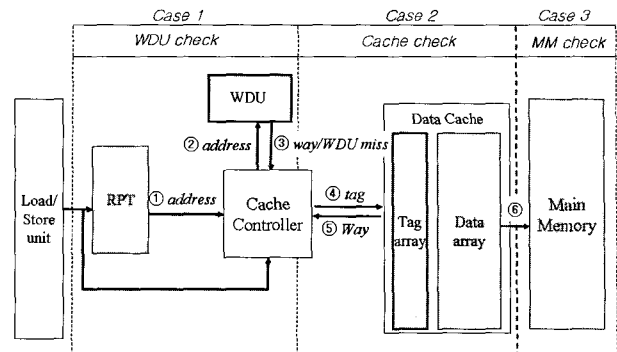


그림 3. WR-캐시 참조 과정

Fig. 3. Cache access flow in WR-cache.

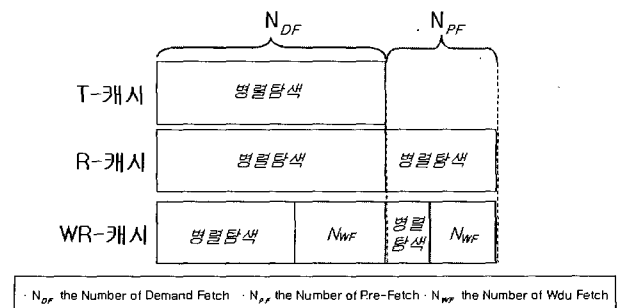


그림 4. 캐시구조별 태그 탐색 방법 구성

Fig. 4. Tag lookup methods according to cache type.

일반적으로 시스템에서 캐시참조는 요구인출과 선인출로 구성된다. 그림 4는 T-캐시, R-캐시 및 WR-캐시 구조의 캐시참조 분포를 보인다. T-캐시 구조는 캐시참조시 요구인출(N_{DF})로만 구성되며 이때의 캐시참조는 태그메모리에 대한 병렬탐색을 수행해야 한다. R-캐시 구조는 캐시참조시 요구인출(N_{DF})과 선인출(N_{PF})로 구성된다. 이때 선인출은 메모리참조 시간을 줄이는 효과는 크지만 태그메모리에 대한 병렬탐색 횟수를 증가시킨다. 따라서 R-캐시 구조는 T-캐시 구조에 비하여 메모리참조 시간은 단축시킬 수 있으나 별도의 에너지를 필요로 하므로 에너지소모율을 높이는 단점을 지닌다. WR-캐시 구조는 R-캐시와 마찬가지로 선인출을 수행하므로 캐시참조 횟수는 동일하다. 그러나 타 캐시구조와 비교할 때, WR-캐시 구조는 요구인출과 선인출시에 WDU탐색을 통하여 참조할 태그정보를 얻을 수 있으므로 병렬탐색을 회피할 수 있다. 즉, WR-캐시 구조는 WDU탐색을 통하여 WDU hit 횟수가 높을수록 태그메모리의 병렬탐색 횟수가 줄어들므로 낮은 에너지 소모율을 기대한다. 그에 반해, T-캐시나 R-캐시 구조는 WDU탐색이 존재하지 않으므로 병렬탐색 횟수는 변화 없다. 또한 WR-캐시 구조는 R-캐시 구조와 비교할 때 선인출로 인한 전체 메모리참

조 시간을 단축시키는 공통점이 있으나 WDU탐색을 통한 병렬검색 횟수를 최소화시키므로 에너지소모율도 낮출 수 있는 장점을 지닌다.

다. 캐시 구조별 성능 비교

이 절에서는 이미지처리 프로그램의 데이터참조 과정으로부터 선인출로 인한 소모전력의 증가 및 영역결정장치에 의한 전력 감소의 효과를 설명한다. 2차원 이미지를 처리한다고 가정하자. 그림 5(a)~(c)는 이미지를 픽셀 단위로 분할한 것으로 이미지는 행 우선순위에 의하여 메모리에 적재되고 참조순서가 적재순서와 일치한다고 가정한다. 각 그림에서 ①~⑧의 숫자는 픽셀의 참조순서 번호이며, 굵은 선으로 표시된 4 픽셀의 묶음은 캐시로 매핑되는 메모리블록을 구분한 것이다. 즉, 이미지 각 행은 2개의 메모리블록으로 구성된다.

그림 5(a)는 T-캐시 구조에서 이미지가 참조될 경우 병렬태그비교 결과를 나타낸다. 각 픽셀에 T로 표시된 것은 각 픽셀 데이터의 요구인출시 캐시에서 해당 데이터 검색을 위한 병렬 태그 비교가 수행됨을 나타낸다. 또한, 그림 5(a)에서 회색으로 표시된 영역(way)은 병렬 태그비교 결과 캐시미스가 발생한 것을 나타낸다. 프로그램이 이미지의 첫 행 참조 과정에서, 데이터 ①을 요구인출하기 위하여 캐시를 검색한다. 이 때, 캐시미스로 인해 해당 메모리 블록이 캐시로 적재된다. 데이터 ②~④의 요구인출은 해당 메모리블록이 캐시에 적재되었으므로 캐시적중이 발생한다. 단, 캐시적중은

메모리참조시간을 단축시키는 반면, 캐시적중/미스여부를 판단하는 태그비교는 모든 데이터의 요구인출 시에 수행된다.

그림 5(b)는 R-캐시 구조에서의 병렬 태그비교 결과이다. 데이터 ①~③의 요구인출 과정은 T-캐시와 동일하나, 각 요구인출마다 데이터의 메모리주소를 동적참조예측기로 전달하여 미래의 메모리 주소를 계산한다. 데이터 ①,② 및 데이터 ②,③간 메모리주소 간격이 동일하므로, 데이터 ③의 참조가 끝난 시점에서 데이터 ④의 메모리참조가 예측되고 선인출을 수행한다. 데이터 선인출 과정은 요구인출 과정과 동일하게 수행되므로 예측된 메모리주소를 캐시에서 탐색하기 위하여 병렬 태그비교를 수행한다. 그림 5(b)에서 T/ 로 표시된 것은 선인출시 캐시탐색을 위한 병렬 태그비교가 수행됨을 나타낸다. 그림 5(b)는 선인출하려는 데이터를 포함한 메모리블록이 이미 캐시에 적재되었으므로 병렬 태그비교로써 선인출 과정이 종료된다. 데이터 ③의 처리 종료 후, 데이터 ④의 요구인출은 병렬 태그비교를 통하여 캐시에서 적중된다. 그림에서 /T 표시는 데이터 요구인출시 캐시탐색을 위한 병렬 태그비교가 수행됨을 나타낸다. 그림 5(a)와 5(b)를 비교할 때, R-캐시가 T-캐시의 미스 횟수를 15회 감소시킨 반면 병렬 태그비교 횟수는 64회에서 125회로 급격히 증가함을 알 수 있다. 이는 데이터 선인출이 프로그램의 수행속도를 향상시킨 반면 전력소모를 크게 증가시킴을 나타낸다.

그림 5(c)는 WR-캐시 구조에서의 병렬 태그비교 결과를 나타낸다. 그림 5(c)에서 데이터 ①이 요구인출 될 때, 캐시의 병렬 태그비교 결과 캐시에서 미스 된다. 이 때, WR-캐시는 미스된 메모리블록을 캐시로 적재하고, 이 메모리블록의 영역(way) 정보를 WDU에 갱신하고, 요구인출된 메모리 주소를 동적참조예측기로 전달한다. 데이터 ②와 데이터 ③이 요구인출 될 경우, 이들을 포함하는 메모리 블록의 영역(way)정보가 앞서 WDU에 저장되었으므로 캐시의 병렬 태그비교 없이 캐시의 데이터 참조가 직접 수행된다. 또한 동적참조예측기는 데이터 ①~③의 참조정보로부터 데이터 ④를 선인출하며, 선인출 과정도 캐시의 병렬 태그비교 이전에 WDU의 영역(way) 정보를 검색한다. 단, 그림 5(c)의 데이터 ⑤의 선인출과 같이, 선인출 또는 요구인출 되는 데이터블록의 영역(way)정보가 WDU에 없는 경우에는 병렬 태그비교를 수행한다. 그림에서 W/ 및 /W 표시는 각각 선인출과 요구인출시 캐시 검색을 위한 WDU탐색이 수행됨을 나타내며 T/로 표시된 것은 병렬 태그비교

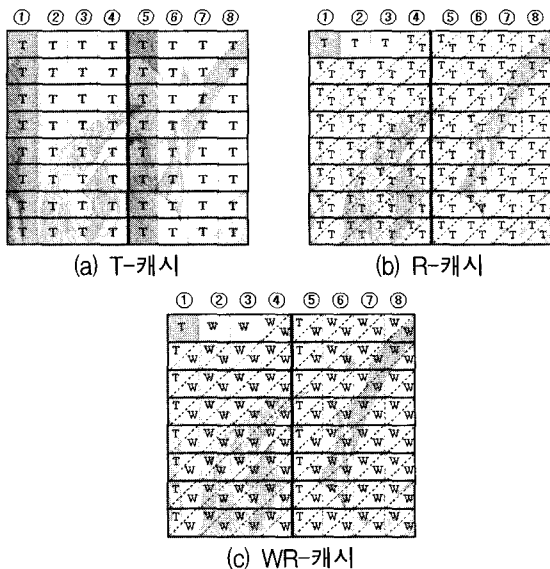


그림 5. 태그 탐색 방법에 따른 전력 소모 요소 비교
Fig. 5. Comparison of energy consumption factors according to tag lookup method.

색을 수행됨을 나타낸다. 이 때, T는 WDU탐색이 선행되었음을 내포한다. WR-캐시의 경우, T-캐시에 대한 캐시미스의 감소는 R-캐시와 동일하며 대다수 병렬 태그비교가 WDU검색으로 대체됨을 알 수 있다.

이상의 비교는 WR-캐시 구조가 데이터 선인출을 채용하여 캐시 적중률을 높이는 한편 선인출이 유발하는 태그 병렬탐색 횟수의 증가 및 이로 인한 전력소모 증가를 억제함으로써 저전력을 요구하는 고성능 임베디드 프로세서의 캐시 구조로 적합함을 나타낸다.

III. 실험

1. 실험환경과 벤치마크

본 논문에서는 WR-캐시 구조의 성능을 분석하기 위하여, T-캐시, R-캐시, WR-캐시 구조별 캐시 시뮬레이터를 구현하고 벤치마크 수행 결과를 분석하였다. 캐시 시뮬레이터는 위스콘신 대학에서 개발한 트레이스 구동형 캐시 시뮬레이터인 Dinero III^[11]를 기반으로 영역결정장치 및 선인출 메커니즘을 추가하여 구현하였다. 그림 6은 캐시 시뮬레이터 실험과정으로써, 실험은 벤치마크 프로그램의 명령어 트레이스 생성, 데이터 캐시 시뮬레이션, 소모전력 계산으로 구성된다.

벤치마크 프로그램의 명령어 트레이스는 Alpha CPU 용 목적코드 분석도구인 ATOM 시뮬레이터^[12]를 사용하여 생성하였다. 명령어 트레이스는 연산, 분기 및 메모리 참조 명령어가 혼합된 형태로 생성되는데, 메모리 참조 명령어는 캐시의 수행과정 모의실험으로 사용되며 연산/분기 명령어는 메모리참조 지연시간을 측정목적으로 사용된다. 데이터 캐시 시뮬레이션은 캐시크기, 블록 크기, 연관도, 교체정책 등을 매개변수로 입력 받아 벤치마크 수행 시 캐시, 동적참조예측기, 영역결정장치의 동작을 모의실행한 후 캐시적중률, 실행사이클, 선인출 적중률, 영역결정장치 적중률 등의 실험결과를 얻는다. 소모전력 계산은 캐시 시뮬레이션으로부터 도출된 실험 결과를 캐시전력분석모델에 적용하여 계산하였다. 캐시 전력분석모델은 Cacti 모델^[13]을 토대로 영역결정장치와 선인출 하드웨어 로직 변화를 반영하여 설계하였다.

실험에 사용된 벤치마크는 멀티미디어 처리의 대표적 프로그램인 MPEG과 JPEG으로써 각 프로그램은 압축모듈과 해제모듈로 구성되며 각 워크그룹이 제공하는 데이터를 대상으로 실험하였다. 실험은 캐시 크기가 8K, 16K, 32K이고 캐시 연관도가 2~32 인 경우에 대하여 수행하였으며 영역결정장치의 엔트리 수는 8개로

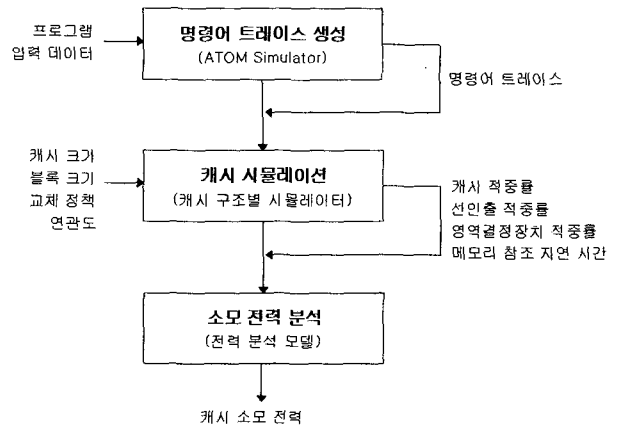


그림 6. 캐시 시뮬레이터를 통한 성능분석

Fig. 6. Performance analysis with cache simulator.

구현하여 실험하였다. 여기서 실험결과에 대한 분석은 T-캐시를 기준으로 다른 캐시에 대한 메모리참조시간 변화율(T), 소모전력 변화율(E)을 이용한 캐시성능 변화율(Performance: $P = \frac{1}{E \cdot T}$)로 나타낸다.

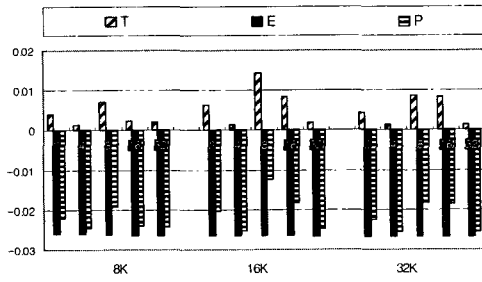
2. 성능평가 및 분석

가. 데이터 선인출에 대한 캐시 성능분석

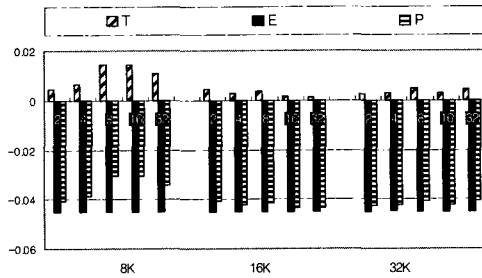
그림 7은 T-캐시 구조를 기준으로 R-캐시 구조에서 벤치마크별 메모리참조시간 변화율(T) 및 전력소모 변화율(E)을 나타낸 것이다. 실험결과로부터 R-캐시 구조는 데이터 선인출로 인하여 T를 단축시키는 반면 E를 크게 증가시키는 것을 알 수 있다. 즉, T-캐시에 대한 성능변화율로 볼 때, R-캐시는 E의 증가가 평균 2.53%로, T의 감소가 평균 1.15%에 비하여 큼을 알 수 있다. 따라서 캐시 성능(P)으로 나타낼 때, 데이터 선인출은 캐시 성능을 오히려 저하시킴을 알 수 있다.

나. 영역결정장치에 대한 캐시 성능분석

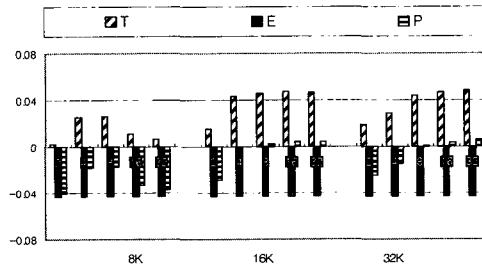
그림 8은 T-캐시 구조를 기준으로 WR-캐시 구조의 성능변화를 분석한 결과로 성능분석을 위하여 메모리참조시간 변화율(T) 및 전력소모 변화율(E)을 실험하였다. 실험결과를 통하여 cjpeg을 제외한 응용 프로그램의 경우, WR-캐시는 T를 단축시킬 뿐 아니라 E도 단축되는 것을 볼 수 있다. 따라서 캐시성능(P)은 두 가지 지표에 대한 변화율로서 더욱 크게 향상됨을 알 수 있다. 반면, cjpeg의 경우에는 WR-캐시의 소모전력이 나머지 응용 프로그램의 경우와는 달리 증가하였다. E의 증가율은 1.68%로 R-캐시에서의 증가율 2.65%에 비하여는 적으나 결과적으로 P의 감소가 나타났다. 이것은 cjpeg 벤치마크가 데이터참조시 이 낮은 지역성을 가지



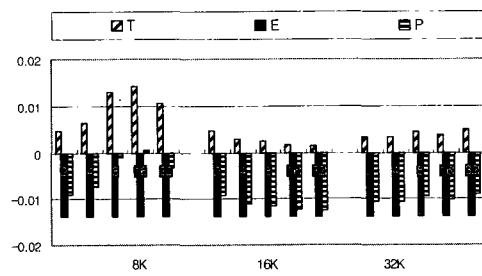
(a) cjpeg



(b) djpeg



(c) mpeg2encoder



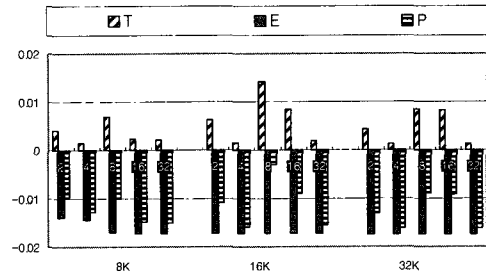
(d) mpeg2decoder

그림 7. T-캐시에 대한 R-캐시의 성능(P) 비교
Fig. 7. Comparison of cache performance.
($P = P_{R-cache} / P_{T-cache}$)

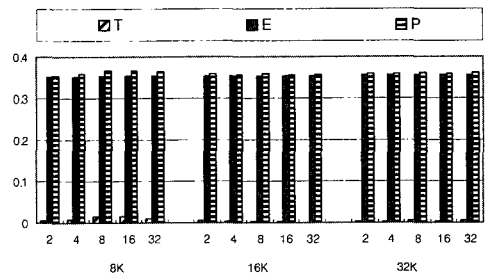
므로 영역결정장치에서의 태그 적중률이 낮아 영역결정 장치 사용에 의한 소모전력 감소가 낮다고 해석된다.

다. 캐시구조별 성능분석

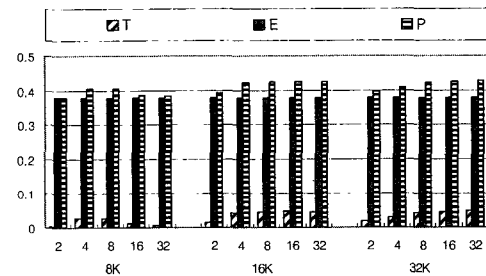
그림 9는 벤치마크의 수행에 대한 캐시구조별 성능



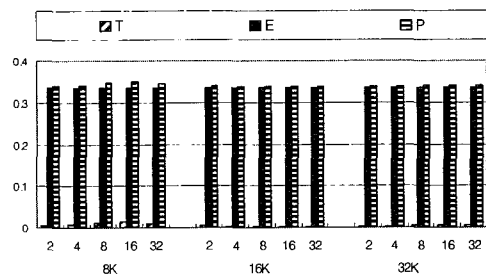
(a) cjpeg



(b) djpeg



(c) mpeg2encoder



(d) mpeg2decoder

그림 8. T-캐시에 대한 WR-캐시의 성능(P) 비교
Fig. 8. Comparison of cache performance.
($P = P_{WR-cache} / P_{T-cache}$)

비교를 나타낸다. 캐시 구조별 성능 비교에서 R-캐시 구조는 T-캐시 구조에 비하여 P가 저하되는 반면, WR-캐시 구조는 P가 평균 27.41%의 성능향상이 있음을 알 수 있다. 이와 같은 실험결과를 통하여, WR-캐시 구조가 기존의 다른 캐시 구조와 비교할 때 우수한

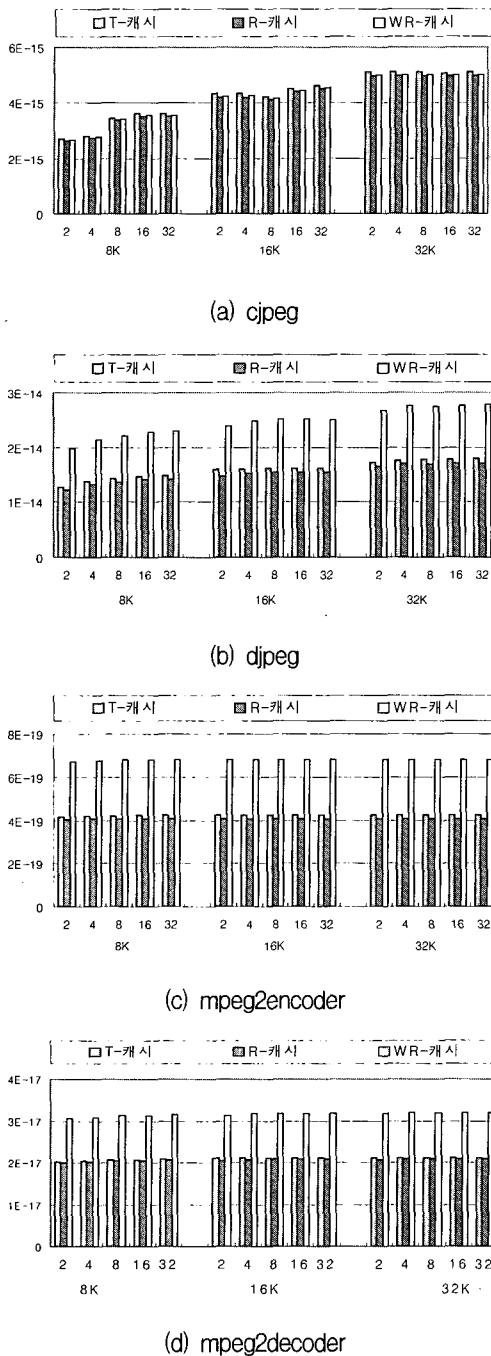


그림 9. 캐시구조별 성능비교
 Fig. 9. Comparison of cache performance.

캐시성능(P)을 가짐을 알 수 있다. 또한 이와 같은 WR-캐시 구조의 성능향상은 선인출로 인한 높은 캐시 적중률과 병렬탐색 최소화로 인한 낮은 전력소모율로 기인함을 알 수 있다.

IV. 결 론

모바일 시스템에 주로 사용되는 임베디드 프로세서

는 배터리에 매우 의존적이다. 일반적으로 임베디드 프로세서의 총 에너지소모량 가운데 40% 이상이 캐시에서 사용되고 있다. 따라서 많은 연구들이 임베디드 프로세서에 적합한 고성능 저전력 캐시를 설계하기 위하여, 데이터 선인출 기법과 캐시참조시에 병렬탐색을 최소화하는 기법을 연구하고 있다.

본 논문에서는 시스템의 성능향상과 에너지소모율의 최적화를 고려한 임베디드 프로세서에 적합한 데이터캐시(WR-캐시:WDRPT Data Cache) 구조를 제안하였다. 제안한 WR-캐시 구조는 기존 데이터캐시 구조에 선인출(prefetching) 장치와 태그히스토리 테이블(tag history table)을 구비하였다. WR-캐시는 선인출 장치를 구비함으로써 캐시적중률을 높이며 태그히스토리 테이블을 활용함으로써 캐시참조시 태그메모리에 대한 병렬탐색 횟수를 최소화하는 특징을 지닌다. 이와 같은 전략적인 WR-캐시 구조는 캐시적중률을 향상시켜서 시스템의 전체 성능을 향상시킬 뿐 아니라 태그메모리에 대한 병렬탐색의 횟수를 최소화하여서 시스템의 전체 전력소모를 현저히 줄어드는 장점을 지닌다.

본 논문에서는 WR-캐시 구조의 성능을 평가하기 위하여, 기존 데이터 캐시인 T-캐시, 선인출 캐시인 R-캐시 구조 및 제안한 캐시인 WR-캐시 구조별 캐시 시뮬레이터를 설계하여 각 캐시구조별 성능을 비교·분석하였다. 실험 결과를 통하여 메모리참조시간 측면에서는 WR-캐시구조가 타 캐시 구조에 비하여 평균 1.15% 성능을 향상시키며, 전력소모 측면에서는 WR-캐시구조가 T-캐시 구조에 비하여 40.97%, R-캐시 구조에 비하여 45.73%의 성능이 향상됨을 확인하였다.

참 고 문 헌

- [1] T. Mudge, "Power: A First-Class Architectural Design Constraint," *IEEE Computer*, Vol. 34, no. 4, pp. 52-58, Apr. 2001.
- [2] M. Baron, "Analog and CPU Wizards Reduce Digital Power: National Semiconductor and ARM Increase Battery Life," *Microprocessor Report*, Vol. 17, No. 1, pp. 10-14, Jan. 2003.
- [3] A. Smith, "Sequential Program Prefetching in Memory Hierarchies," *IEEE Computer*, Vol. 11, no. 2, pp.7-21, 1978.
- [4] B. Mathew and A. Davis, "An Energy Efficient High Performance Scratch-pad Memory System" *Proceedings of the Design Automation Conference (DAC)*, 2004.

[5] F. Dahlgren, M. Dubois and P. Stenstrom, "Fixed and Adaptive Sequential Prefetching in Shared-memory Multiprocessors," *Proceedings of the International Conference on Parallel Processing*, pp. 156-63, Aug. 1993.

[6] T. F. Chen and J. L. Baer, "Effective Hardware-Based Data Prefetching for High Performance Processors," *IEEE Transactions on Computers*, Vol. 44, no. 5, pp. 609-623, May. 1995.

[7] D. Nicolaescu, Al. Veidenbaum and A. Nicolau, "Reducing Power Consumption for High-Associativity Data Caches in Embedded Processors," *Proceedings of the conference on Design, Automation and Test in Europe*, pp. 11064-11069, Mar. 2003.

[8] K. Defendorff and P. K. Dubey, "How Multimedia Workloads Will Change Processor Design," *IEEE Computer*, Vol. 30, no. 9, pp. 43-45, Sep. 1997.

[9] A. Hasegawa, I. Kawasaki, K. Yamada, S. Yoshioka, S. Kawasaki and P. Biswas, "SH3: High Code Density, Low Power," *IEEE Micro*, Vol. 15, no. 6, pp. 11-19, Dec. 1995.

[10] Z. Zhu and X. Zhang, "Access-mode Predictions for Low-Power Cache Design," *IEEE Micro*, Vol. 22, no. 2, pp. 58-71, Mar.-Apr. 2002.

[11] M. D. Hill, Dinero III Cache Simulator, http://www.ece.cmu.edu/~ece_548/tools/dinero

[12] A. Srivastava and A. Eustace, "ATOM: A System for Building Customized Program Analysis Tools," *Proceedings of the ACM SIGPLAN 94*, pp. 196-205, 1994.

[13] P. Shivakumar and N. P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," *HP Western Research Labs, Tech. Rep.* 2001.

[14] M. Zhang and K. Asanovic, "Highly-Associative Caches for Low-Power Processors," *33rd International Symposium on Microarchitecture*, pp. 196-205, Dec. 2000.

[15] W. T. Shiu, "Memory Exploration for Low Power Embedded Systems", *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, pp. 250-253, May. 1999.

[16] 이정훈, 최진혁, 김신덕, "저전력 온칩 메모리에 관한 연구 동향 및 개발 방향," *정보과학회지*, 제20권, 제10호, 37-44쪽, 2002년 10월

저 자 소 개



문 현 주(정회원)
1995년 충북대학교 컴퓨터과학과 학사 졸업.
1997년 충북대학교 전자계산학과 석사 졸업.
2003년 충북대학교 전자계산학과 박사 졸업.

2003년~2005년 나사렛대학교 정보과학부 전임강사
2006년~현재 남서울대학교 컴퓨터학과 전임강사
<주관심분야 : 캐시구조, 임베디드 시스템, 유비쿼터스 컴퓨팅>



지 승 현(정회원)
1993년 충북대학교 전자계산학과 학사 졸업.
1995년 충북대학교 전자계산학과 석사 졸업.
2000년 충북대학교 전자계산학과 박사 졸업.

2000년~2003년 미국 미주리주립대 연구교수
1999년~2005년 백석대학 디지털정보학부 전임강사
2005년 9월~2006년 8월 한국산업기술대학교 교육학과(컴퓨터교육) 초빙교수
2006년 9월~현재 한국고용정보원 정보화전략팀 부연구원
<주관심분야 : 캐시구조, 임베디드 시스템, 유비쿼터스 컴퓨팅>