저전력 캐쉬 대체 정책

이문상 이원진 이준원 맹승렬
한국과학기술원 전자전산학과
{mslee, wjlee, joon, maeng}@camars.kaist.ac.kr

Cache Replacement Policies for Energy Efficiency

Moon-Sang Lee Won-Jin Lee Joon-Won Lee Seung-Ryoul Maeng
Dept. of EECS., KAIST

요 약

소형의 이동 컴퓨터 시스템이 발전하면서 프로세서의 전력 소모(energy dissipation)가 중요한 이슈가 되고있다. 현재 대부분의 프로세서들은 성능 향상을 위해 캐쉬를 사용하고 있고 이것은 프로세서내의 많은 비율의 전력을 소모한다. 따라서 저 전력 프로세서를 설계하기 위해서는 내장 캐쉬(on-chip cache)의 전력 소모를 줄이는 것이 중요하다. 본 논문은 캐쉬 대체 전략으로 현재 많이 사용되는 LRU(Least Recently Used) 방식을 LFU(Least Frequently Used), LFUT(LFU with Threshold), FIFO(First In First Out) 방식과 전력 효율적 측면에서 비교 분석한다.

1. 서론

최근 랩탑(lap-top), 노트북(notebook), PDA(Personal Digital Assistance) 등의 이동형 컴퓨터의 보급이 늘어나고 있고, 가까운 미래에는 소형의 고성능 이동 시스템이 보편화될 전망이다. 또, 이동 통신 분야에서도 앞으로는 개인용 휴대 단말기가 웹 브라우징 (web browsing), 영상 전송 등 복잡하고 다양한 기능을 제공하게 될 것이다. 이러한 추세에 따라 컴퓨터 시스템에서 전력 효율 문제가 점점 중요해지고 있으며 시스템의 각 부품별로 전력 소모를 낮추려는 노력이 많이 시도되고 있다. 프로세서는 시스템 전체의 40%이상의 전력을 소모하며, 프로세서의 25% - 50% 정도가 내장 캐쉬에서 소모된다[1]. 캐쉬가 많은 전력을 소모하는 이유는 캐쉬가 일반적으로 SRAM으로 구현되어 있으므로 트랜지스터 집적도가 높고, 프로세서 칩의 다이(die) 면적 중 50-90%의 많은 부분을 차지하고 있기 때문이다[2]. 대표적인 예로 HP PA 8500 프로세서는 다이 면적 중 70% 정도를 캐쉬가 차지하고 있고, Alpha 21164 프로세서는 프로세서의 전체 소모 전력 중 25% 정도를 내장 캐쉬가 소모하며 DEC SA-110 프로세서에서는 43%의 전력을 내장 캐쉬가 소모한다[1]. 이러한 이유로 현재 저 전력 마이크로프로세서 연구에서 캐쉬는 가장 중요한 부분 중 하나로 간주되고 있으며, 저 전력 캐쉬의 개발은 저 전력 프로세서의 개발을 의미하게 된다.

캐쉬에서의 전력 효율을 향상시키고자 하는 기존의 접근들은 반도체 자체의 정전용량(capacitance)을 낮추는 방법, 공급 전압을 낮추는 방법, 그리고 전력소모가 적은 캐쉬 구조를 설계하는 방법 등이 있다. 이 중 정전용량을 낮추는 방법은 반도체 공정기술을 향상시켜 전력 효율적인 하드웨어 회로소자를 구현하는 방법이고, 공급 전압을 낮추는 방법은 전압의 크기를 줄여 캐쉬의 클럭 속도(clock speed)를 떨어뜨림으로서 전력을 줄인다. 캐쉬의 소모 전력은 캐쉬의 각 비트 셀(bit cell)에 저장 되어있는 비트들이 전환(transition)하는 회수에 의해 결정되는데, 저 전력 캐쉬 구조를 설계하는 방법은 바로 비트 전환의 회수를 감소시켜 소모 전력을 줄이려는 방법이다.

본 논문에서는 캐쉬의 전력 효율을 비교, 분석하기 위해 블록 대체 전략에 중점을 둔다. 현재 대부분의 집합 연관 캐쉬(set associative cache)가 사용하고 있는 LRU(Least Recently Used) 방식은 캐쉬의 블록들에 대해 가장 오랫동안 사용되지 않은 블록을 대체될 블록으로 선정하기 때문에 각 블록 상태 비트(status bit)를 두어 이에 대해 순서 매김(ordering)이 필요하고, 매번 캐쉬 접근(access)이 일어날 때마다 상태 비트들이 갱신되어야 하기 때문에 전력 효율이라는 측면에서는 적합하지 않다. 따라서 본 논문에서는 다양한 캐쉬 설정들에 대해 직접사상 캐쉬(direct mapped cache)와 LRU, LFU, LFUT, FIFO 캐쉬 대체 정책을 사용하는 집합 연관 캐쉬의 전력 소모량을 측정하고, 측정된 결과를 분석함으로써 전력 효율적인 캐쉬 설계의 방법론을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 캐쉬에서의 소모 전력을 측정하기 위한 분석적 모델과 저전력 캐쉬 디자인에 관련된 기존의 연구를 살펴본다. 3장에서는 저 전력 캐쉬 대체 정책에 대해 살펴보고 4장에서는 실험 결과를 분석한다. 마지막으로 5장에서는 결론 및 향후 과제에 대해 설명한다.

2. 관련연구

2.1 전력 모델(Power model)

CMOS 회로에서 소모하는 전력은 다음과 같이 구할 수 있다[1].

$$P = \frac{1}{2} \times C \times V^2 \times bs \qquad (1)$$

여기서 C 는 회로의 정전용량(capacitance), V 는 공급 전압, bs 는 비트 전환(bit transition) 회수를 나타내며 전력 P의 단위는 와트(Watt)이다. 위의 식(1)에 의하면 칩 구현에 사용될 특정 CMOS 기술이 결정되면 소모 전력은 비트 전환 회수에 의해 결정됨을 알 수 있다.

일반적인 m 방향 집합 연관 캐쉬(m-way set-associative cache)는 주소 디코딩 단계(address decoding path), 셀 배열 접근(cell-array access) 단계, 입출력 단계(I/O path)의 주요 세 부분으로 구분할 수 있고, 이 세 부분은 캐쉬의 가장 큰 전력 소모원들이다. 각 부분에서 전력 모델식[3]을 살펴보면 셀 배열과 주소 버스(address bus)등에서의 비트 전환 회수가 전력 소모량을 결정하는 가장 중요한 요소임을 알수 있다.

2.2 저전력 캐쉬 구조

캐쉬 설정 최적화(optimized cache configuration)

캐쉬의 크기가 정해졌을 경우 캐쉬 설정(configuration)을 어떻게 정하느냐에 따라 다양한 성능과 전력 소모량을 나타낸다. 캐쉬 설정을 구성하는 인자(parameter)들은 집합 연관성(set-associativity), 캐쉬 라인(cache line) 크기, 캐쉬의 크기 등에 따라 다양한 성능과 전력 사용량을 나타낸다[1,3,4].

블록 버퍼링(block buffering)

공간 지역성(spacial locality)과 시간 지역성(temporal locality)을 이용하여 매번 캐쉬 접근이 발생할 때마다 셀 배열에서 출력 래치(output latch)로 비트 이동이 발생하는 것을 제거하는 방법이다. 가장 최근에 접근이 발생한 집합(set)을 블록 버퍼 래치에 기록하고 현재 접근되는 캐쉬 라인의 집합과 비교하여 동일한 집합 번호면 셀 배열로부터 출력 래치로 비트를 이동시키지 않고 출력 래치에 남아있던 데이터를 사용한다. 이렇게 함으로써 셀 선충전(precharging)의 회수를 줄이고, 출력 래치로 이동하는 비트수를 줄일 수 있다[1].

캐쉬 서브뱅킹(cache subbanking)

하나의 캐쉬 라인은 다수개의 워드(word)들로 구성되는데 실제로 프로세서가 캐쉬에 읽고 쓰는 동작을 할 때는 워드 단위로 수행한다. 따라서 매번 캐쉬 라인을 접근할 때마다 캐쉬 라인 전체를 활성화시키는 것은 전력 소모를 가져오므로 캐쉬 라인을 워드 단위의 서브뱅크(subbank)로 나누고 필요한 서브뱅크만 활성화시킴으로써 전력 소모를 줄일 수 있다[1].

그레이 코드 주소(Gray code addressing)

전형적인 프로세서에서는 주기억장치의 주소(address)를 2의 보수(2's complement)로 표현해 사용해왔다. 그러나 이는 순차적인 메모리 접근에 대해 많은 비트 전환을 발생시키므로 메모리 주소를 그레이 코드로 표현하여 순차적 메모리 접근이 발생할 경우 비트 전환을 줄일 수 있다. 예를 들어 메모리 주소 0번지부터 15번지까지 순차적 메모리 접근이 발생할 경우 2의 보수 방식은 31번, Grey 코드 방식은 16번의 비트 전환이 발생한다. 이처럼 공간 지역성이 높은 응용프로그램을 수행할 경우 주소 버스의 비트 변환을 감소시킴으로써 전력 사용을 줄일 수 있다[3].

3. 전력 효율을 고려한 캐쉬 대체 정책

캐쉬는 메모리 접근 지연시간(memory access latency)를 줄이기 위한 작고 빠른 속도의 SRAM이다. 내장 캐쉬는 프로세스의 모든 명령어와 데이타를 포함할 수 없기 때문에 공간 지역성과 시간 지역성을 활용하여 가장 최근에 사용된 명령어와 데이터를 저장한다. 캐쉬 실패(cache miss)가 발생할 경우, 캐쉬 대체 정책에 의해 선택된 캐쉬 라인을 메모리에 저장하고, 접근에 실패한 메모리 라인을 캐쉬로 불러온다. m 방향 집합 연관 캐쉬일 경우 메모리로부터 불러온 데이터를 m개의 위치 중 어느 곳에 저장할지를 결정하는 캐쉬 대체 정책은 <그림 1>의 빗금 친 영역에 해당하는 상태 비트들과 대체 로직(replacement logic)으로 구성된다. 기존의 집합 연관 캐쉬들은 주로 LRU 방식을 사용해 왔다. LRU 방식을 구현하기 위해서는 하나의 집합에 속하는 m개의 캐쉬 라인들을 순서대로 정리해야하기 때문에 $mlog_2m$ 비트가 요구된다. 또한 매번 캐쉬 접근이 발생할 때마다 접근 순서에 근거한 순서 매김이 필요하므로 상태 비트의 전이가 많이 발생한다. LRU 방식은 하드웨어로 구현하기 쉽고, 캐쉬 적중률(cache hit ratio)이 높기 때문에 전통적으로 많이 사용되어 왔지만 전력 낭비가 존재한다는 단점이 있다. 90% 이상의 높은 캐쉬 적중률로 인해 캐쉬 대체는 빈번하게 발생하지 않는 반면 순서 매김에 사용되는 캐쉬 상태 비트들은 매번 캐쉬 접근이 발생할 때마다 갱신되어야 한다. 따라서 4 방향 집합 연관 캐쉬에서 상태 비트의 전환을 나타내는 <그림 2>의 예와 같이 캐쉬에 존재하는 데이터의 비트 전환보다 상태 비트의 비트 전환이 전력 소모에 더 큰 영향을 미칠 수 있다. 특히, 명령어 캐쉬와 같이 순차적인 읽기 접근만 발생하는 경우 상태 비트의 비트 전환 회수는 캐쉬 전력 소모에 큰 영향을 미친다. 저 전력 캐쉬에 관한 기존의 연구들은 캐쉬 대체 정책에 사용되는 상태 비트의 비트 전환 회수를 고려하지 않고, 단순히 캐쉬 라인의 비트 전환 회수만을 고려하고 있기 때문에 [1,2,3,4] 캐쉬 대체 정책에 사용되는 상태 비트들의 비트 전환을 포함하는 종합적인 전력 소모량 측정 및 분석이 필요하다.

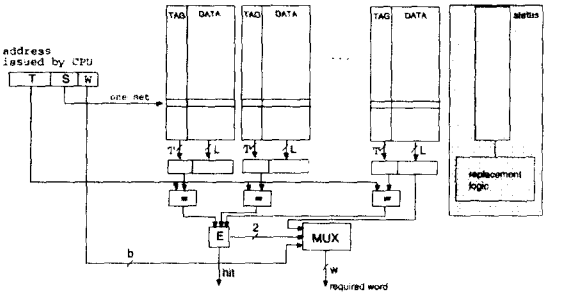


그림 1 m-way set associative cache

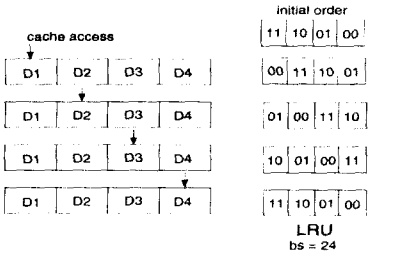


그림 2 LRU 상태 비트 전환

4. 성능 측정

4.1 실험 환경

본 논문에서는 캐쉬의 전력 소모 실험을 위해 SimpleScalar V2.0[5]을 사용하여 SPEC95 벤치마크를 수행하였다. 직접사상 캐쉬와 LRU, LFU, LFUT, FIFO 캐쉬 대체 정책을 채택한 집합 연관 명령어 캐쉬에 대해 비트 전환 회수를 실험하고, 캐쉬 전력 소모에 미칠 수 있는 영향을 분석하였다. 실험에 사용된 캐쉬 설정은 <표 1>과 같다.

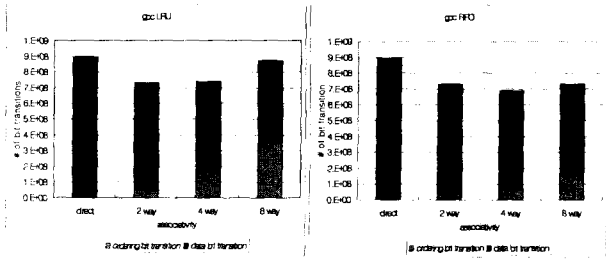
항목	설정값
cache size	16KB, 32KB
block size	32 bytes
associativity	direct mapped, 2 way, 4 way, 8 way
replacement policy	LRU, LFU, LFUT, FIFO

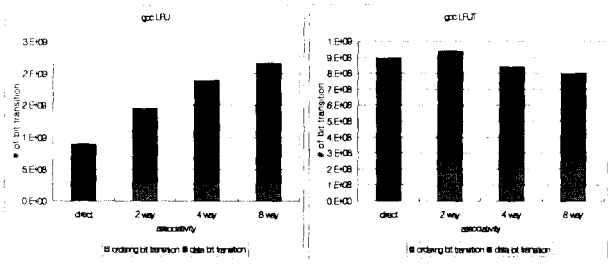
표 1 캐쉬 실정 항목

4.2 성능 분석

아래 4개의 그래프는 SPEC95 gcc를 수행했을 경우 캐쉬 대체 정책에 따른 비트 전환 회수를 나타낸다. 집합 연관 캐쉬의 캐쉬 적중률이 직접사상 캐쉬에 비해 높기 때문에 직접사상 캐쉬에 비해 집합 연관 캐

쉬의 총 비트 전환 회수가 전체적으로 작게 나타난다. 캐쉬 적중률이 낮을수록 캐쉬 대체가 많이 발생하므로 캐쉬 라인의 데이터 비트 전환 회수가 많아지기 때문이다. 뿐만 아니라 캐쉬 실패로 인한 메모리 접근은 수행 속도를 감소시키고, 정전용량이 더 큰 프로세서 칩 외부로의 접근이 발생하여 많은 전력을 소모한다. 연관성(associativity)이 큰 캐쉬 설정일수록 순서 매김 비트의 전환 회수가 증가한다. 따라서 캐쉬 적중률의 증가가 크지 않은 4 방향 이상의 집합 연관 캐쉬는 전력 효율성 측면에서 사용이 배제되어야 한다. 그레이 코드로 구현된 카운터를 사용하는 LFU와 LFUT는 LRU와 FIFO에 비해 순서 매김 비트의 전환이 많이 발생한다. 다수개의 워드로 구성되는 캐쉬 블록의 순차적인 접근은 LRU나 FIFO의 순서 매김 비트가 전환하지 않는 경우에도 $size_{word}/size_{block}$번 전환되기 때문에, 비록 LFUT의 캐쉬 적중률이 LRU나 FIFO의 캐쉬 적중률에 근접하게 나타나지만 전력 효율적 캐쉬 대체 정책으로 적합하지 않다.





캐쉬 적중률이 비슷한 LRU와 FIFO의 경우, LRU가 더 많은 비트 전환을 발생시킨다. 비록 데이터 비트 전환 회수는 비슷하지만 LRU의 순서 매김 비트 전환이 FIFO의 순서 매김 비트 전환보다 많이 발생한다. 이는 매번 캐쉬 접근이 발생할 때마다 순서 매김 비트를 갱신해야하는 LRU보다 캐쉬 대체가 발생할 경우에만 순서 매김 비트를 갱신하는 FIFO가 저 전력 캐쉬에 적합한 캐쉬 대체 정책임을 의미한다. FIFO를 사용하는 집합 연관 캐쉬는 LRU와 근접한 높은 캐쉬 적중률을 보이고, 순서 매김 비트 전환이 적어 실험에 사용된 캐쉬 대체 정책들 중 최소의 비트 전환을 나타낸다.

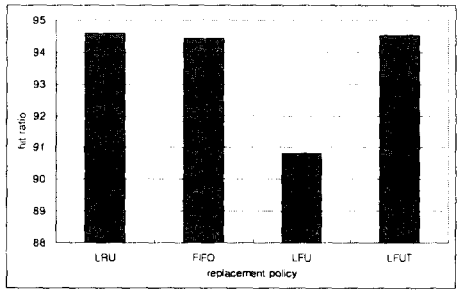
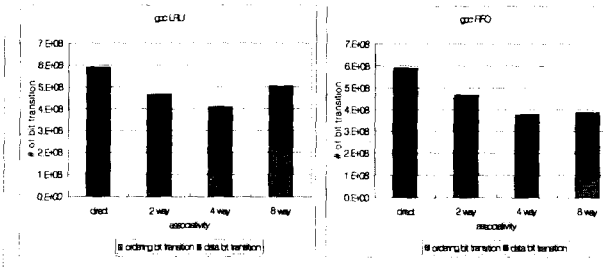


그림 3 2방향 집합 연관 캐쉬의 적중률

아래 2개의 그래프는 LRU와 FIFO를 사용하는 32KB 크기의 명령어 캐쉬에 대한 비트 전환 회수를 나타낸다. 캐쉬 크기가 커질수록 캐쉬 적중률이 증가하여 전체적인 비트 전환 회수가 감소하였다. 32KB 캐쉬에서도 16KB 캐쉬와 동일하게 LRU보다 FIFO의 상태 비트 전환이 작게 나타났다. 4 방향 집합 연관 캐쉬가 최소의 비트 전환 회수를 보이는 것은 증가한 캐쉬 크기로 인해 캐쉬 적중률이 증가했기 때문이다. 캐쉬 적중률 증가로 인한 데이터 비트 전환 회수 감소가 2 방향에서 4 방향으로 증가한 연관성에 의한 상태 비트 전환 회수보다 크기 때문에 전체적인 비트 전환 회수가 감소하였다. 하지만 4 방향에서 8 방향으로 연관성이 증가하면 캐쉬 적중률의 증가로 인한 데이터 비트 전환 회수의 감소보다 상태 비트 전환 회수 증가가 더 크기 때문에 전체적인 비트 전환 회수는 오히려 증가한다. 따라서 전력 효율적인 캐쉬 설정을 위해서는 캐쉬 크기, 연관성, 캐쉬 대체 정책이 종합적으로 고려되어야 한다.



5. 결론

본 논문에서는 저 전력 캐쉬를 위한 캐쉬 대체 정책에 대해 살펴보고, 상태 비트 전환을 포함하는 시뮬레이션을 통해 캐쉬 대체 정책이 캐쉬의 전력 소모에 미칠 수 있는 영향을 살펴보았다. 캐쉬내의 비트 전환 회수에 비례하는 전력 소모를 줄이기 위해서는 높은 캐쉬 적중률을 갖는 캐쉬 설정과 상태 비트 전환이 적은 캐쉬 대체 정책을 적용해야 한다. 앞으로의 연구는 좀 더 다양한 캐쉬 설정에 대한 정확한 전력 소모량 측정과 캐쉬 접근 지연 시간의 변화를 포함할 예정이다. 프로세서 내장 캐쉬의 전력 소모량과 데이타 접근 지연 시간의 타협(trade-off)에 관한 연구는 적정 성능을 요구하는 저 전력 시스템의 설계 및 평가 도구로 활용될 수 있다.

6. 참고 문헌

[1] Kanad Ghose and Milind B. Kamble, "Energy Efficient Cache Organizations for Superscalar Processors", Power-Driven Microarchitecture Workshop, June, 1998.

[2] Christoforos E. Kozyrakis and David A. Patterson, "A New Direction for Computer Architecture Research", IEEE Computer, pp. 24-32, November, 1998.

[3] Ching-Long Su and Alvin M. Despain, "Cache Design Trade-offs for Power Performance Optimization : A Case Study", IEEE ISLPE, pp.63-68, 1995.

[4] Patric Hicks, Matthew Walnock, and Robert Michael Owens, "Analysis of Power Consumption in Memory Hierarchies", Proceedings of the 1997 international symposium on Low power electronics and design, pp. 239-242, August, 1997.

[5] D.Burger and T.M.Austin, "The SimpleScalar Tool Set Version 2.0", Technical Report TR#1342, University of Wisconsin, 1997.

[6] Johnson Kin, Munish Gupta and William H. Mangione-Smith, "The Filter Cache : An Energy Efficient Memory Structure", Proceedings of the thirtieth annual IEEE/ACM international symposium on Micro-architecture, pp. 184-193, December, 1997.