韓國컴퓨터情報學會 論文誌
第14卷 第12號, 2009. 12.

2009-14-12-1-2

멀티 코어 프로세서를 위한 저전력 필터 캐쉬 설계 기법

박 영 진*, 김 종 면**, 김 철 홍***

Low-power Filter Cache Design Technique for Multicore Processors

Young-Jin Park*, Jong-Myon Kim**, Cheol-Hong Kim***

요 약

최신의 멀티코어 프로세서를 설계할 때에는 성능과 함께 전력 효율성이 반드시 고려되어야 한다. 본 논문에서는 싱글 코어 프로세서의 명령어 캐쉬에서 소비되는 전력을 줄이기 위해 사용되는 대표적 기법중 하나인 필터 캐쉬 구조를 멀티 코어 프로세서에 적용하기 위한 새로운 방안을 제시하고자 한다. 명령어 캐쉬는 프로세서 전체에서 소비되는 전력의 상당 부분을 차지하고 있기 때문에, 변형 필터 캐쉬 구조를 이용한 저전력 명령어 캐쉬 설계는 멀티 코어 프로세서의 전력 소비를 줄이는데 있어서 중요한 역할을 담당할 수 있다. 제안하는 변형 필터 캐쉬 구조는 멀티 코어 프로세서에서 필터 캐쉬에 대한 희생 캐쉬를 추가함으로써 1차 명령어 캐쉬에 대한 접근 횟수를 감소시키는 방법을 이용하여 명령어 캐쉬에서 소비되는 총 전력을 줄일 수 있다. 제안하는 명령어 캐쉬 구조의 효율성을 분석하기 위한 모의 실험 도구로 SimpleScalar 시뮬레이터와 CACTI를 사용한다. 모의실험 결과, 제안하는 기술은 멀티 코어 프로세서의 명령어 캐쉬에서 소비되는 전력을 기존의 필터 캐쉬 구조와 비교하여 최대 3.4% 감소시킬 수 있음을 확인할 수 있다. 더욱이 제안하는 구조는 기존의 필터 캐쉬 구조에 비해 보다 우수한 성능을 보여준다.

Abstract

Energy consumption as well as performance should be considered when designing up-to-date multicore processors. In this paper, we propose new design technique to reduce the energy consumption in the instruction cache for multicore processors by using modified filter cache. The filter cache has been recognized as one of the most energy-efficient design techniques for singlecore processors. The energy consumed in the instruction cache accounts for a significant portion of total processor energy consumption. Therefore, energy-aware instruction cache design techniques are essential to reduce the energy consumption in a multicore processor. The proposed technique reduces the energy consumption in the instruction cache for multicore processors by reducing the number of accesses to the level-1 instruction cache. We evaluate the proposed design using a simulation infrastructure based on SimpleScalar and CACTI. Simulation results show that the proposed architecture reduces the energy consumption in the instruction cache for multicore

• 제1저자 : 박영진 교신저자 : 김철홍
• 투고일 : 2009. 09. 04, 심사일 : 2009. 10. 27, 게재확정일 : 2009. 12. 24.
* 전남대학교 전자컴퓨터공학부 석사과정 **울산대학교 컴퓨터정보통신공학부 교수 ***전남대학교 전자컴퓨터공학부 교수
※ 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(구 기초연구지원기초과학 단독연구, KRF-2007-313-D00642)으로 연구되었음.

processors by up to 3.4% compared to the conventional filter cache architecture. Moreover, the proposed architecture shows better performance over the conventional filter cache architecture.

▸ Keyword : 멀티 코어 프로세서(Multicore Processor), 명령어 캐쉬(Instruction Cache), 희생 캐쉬(Victim Cache), 전력 소모량(Energy Consumption)

Ⅰ. 서 론

반도체 공정 기술의 지속적인 발전을 통해 프로세서 성능은 최근 몇 년 동안 급속도로 향상되어 왔다. 하지만 프로세서의 성능 향상은 전력 소모의 증가라는 부정적인 결과를 초래하고 있다. 프로세서에서 소모되는 전력은 클럭 속도에 비례하여 증가하기 때문이다. 이와 같은 이유로 인해 프로세서 설계 시 전력 소모량은 성능과 함께 중요한 고려 요소가 되고 있다 [1][2].

프로세서에서 소모되는 전력의 상당 부분은 캐쉬 메모리에서 소모되므로 캐쉬에서의 전력 소모를 감소시키기 위해 많은 연구들이 제안된 바 있다 [3]. 필터 캐쉬 (Filter Cache) 기법은 크기가 작은 캐쉬를 프로세서 코어와 1차 캐쉬 사이에 삽입하여 1차 캐쉬에 대한 접근 횟수를 감소시킴으로써 성능을 조금 줄이는 대신 전력 소모량을 감소시킬 수 있다 [4]. 필터 캐쉬 적용 시 발생하는 시스템의 성능 저하를 최소화하기 위하여 모드 선택 비트를 사용하는 예측기 (Predictor)를 활용하여 필터 캐쉬의 참조 실패를 야기하는 문제점을 개선할 수 있다 [5]. 선택적-웨이 캐쉬 (Selective-way Cache)는 전력 소모를 줄이기 위해서 동작 중에 연관 캐쉬 (Associative Cache)에서의 일부 웨이들을 비활성화시키는 방법을 이용한다 [6]. 웨이 예측 연관 캐쉬 (Way Predicting Set-associative Cache)는 캐쉬 접근 시 초기에는 예측을 통해 하나의 태그와 데이터 배열에 접근을 한 후, 예측이 정확하지 않은 경우에만 다른 배열들에 대한 접근을 수행한다. 이를 통해 접근 시간을 약간 증가시키지만 전력 소모량을 감소시킬 수 있다 [7][8]. 분할 명령어 캐쉬 (Partioned Instruction Cache)는 구조적인 캐쉬 분할을 통해 접근 시 활성화 되는 메모리 크기를 줄임으로써 전력 소모량을 감소시킨다 [9].

반도체 공정 기술의 지속적인 발전과 싱글 코어 프로세서의 여러 한계점들로 인해 최근에는 멀티 코어 프로세서가 프로세서 시장에서 상당히 큰 관심을 차지하고 있다 [10]. 앞서 기술한 바와 같이 싱글 코어 프로세서에서는 다양한 저전력 기법들이 제안되었지만, 향후 시장 상황을 예측하였을 때에는 멀티 코어 프로세서에서의 전력 소모를 감소시킬 수 있는 기법들이 필요하다. 본 논문에서는 이와 같은 상황에서 계층적 메모리 (Hierarchical Memory)를 사용하는 싱글 코어 프로세서의 전력 소모를 감소시키기 위해 사용되던 대표적인 방안 중 하나인 필터 캐쉬 구조 [4]를 멀티 코어 프로세서에 적용하여 성능 및 전력 효율성을 살펴보고, 이를 기반으로 멀티 코어 프로세서의 전력 소모를 감소시킬 수 있는 새로운 기법을 제시하고자 한다. 제안하는 기법은 멀티 코어 프로세서에서 1차 명령어 캐쉬에 대한 접근 횟수를 줄이기 위하여 필터 캐쉬에 수정된 희생 캐쉬를 추가하는 방식으로 새로운 명령어 캐쉬 구조를 구성한다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 기존의 필터 캐쉬 구조와 희생 캐쉬 구조에 대해 기술하고, 3장에서는 성능과 전력 소모 관점에서 멀티코어 프로세서에서 필터 캐쉬 구조의 효율성을 분석한다. 4장에서는 필터 캐쉬와 수정된 희생 캐쉬를 사용한 새로운 저전력 명령어 캐쉬 구조를 제안하고, 5장에서는 모의실험을 수행하는 방법과 상세한 실험 결과를 보여준다. 마지막으로 6장에서 결론을 맺는다.

Ⅱ. 관련 연구

2.1 기존의 필터 캐쉬 구조

전력 소모량을 줄이기 위해서 필터 캐쉬를 추가한 일반적인 계층적 메모리 구조는 그림 1에서 보이는 바와 같다. 필터 캐쉬는 프로세서 코어와 1차 캐쉬 사이에 삽입되는 매우 작은 크기의 캐쉬로, 1차 캐쉬의 접근 횟수를 줄임으로써 캐쉬 접근 시 소모되는 전력을 감소시킨다 [4]. 필터 캐쉬 구조는 필터 캐쉬를 사용하지 않는 기존의 계층적 메모리 구조와 비교하여 적중률 (Hit Rates)은 감소하는 대신에 전력 소모량을 감소시킬 수 있다.

2.2 희생 캐쉬 구조

희생 캐쉬 (Victim Cache)를 사용하는 계층적 메모리 구조는 그림 2에서 보이는 바와 같다. 희생 캐쉬는 1차 캐쉬에서 충돌 (Conflict)로 인해 방출되는 블록들로 채워지는 크기가 작은 캐쉬 메모리이다 [11]. 1차 캐쉬와 희생 캐쉬는 프로세서 코어에서 요청이 들어올 때 동시에 접근된다. 1차 캐쉬에 요청된 데이터가 있는 경우에 데이터는 프로세서 코어로 이동하게 되고, 1차 캐쉬와 희생 캐쉬 사이에 데이터 교체는 발생하지 않는다. 1차 캐쉬에서 요청된 데이터를 찾지 못하

고, 희생 캐쉬에 요청된 데이터가 존재하는 경우에 희생 캐쉬 내의 데이터는 1차 캐쉬로 보내지고, 1차 캐쉬에서 방출되는 블록은 희생 캐쉬로 보내지게 된다. 프로세서 코어가 요청한 데이터를 1차 캐쉬와 희생 캐쉬에서 모두 찾지 못하는 경우에는 하위 캐쉬 메모리에 접근하게 되고, 하위 메모리에서 가져온 블록은 1차 캐쉬에 위치하게 된다. 이와 동시에 1차 캐쉬에서 방출되는 블록은 희생 캐쉬에 위치하게 된다. 위와 같은 원리로 동작하는 희생 캐쉬 구조는 충돌 미스 (Conflict Miss)로 인한 캐쉬의 적중률 저하를 줄이기 위해 사용되는 가장 널리 알려진 캐쉬 설계 방법 중 하나이다.

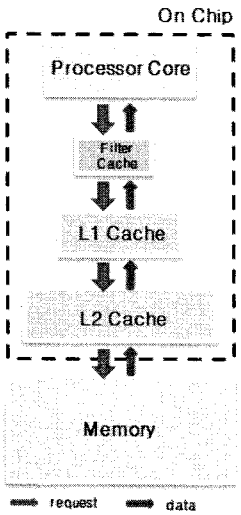

그림 1. 필터 캐쉬를 사용하는 계층적 메모리 구조
Figure 1. Hierarchical memory organization using filter cache

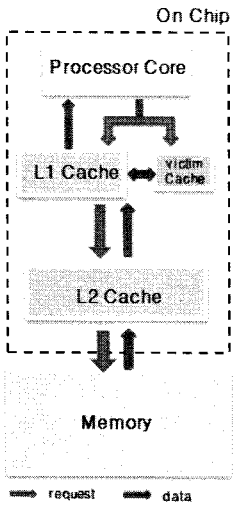

그림 2. 희생 캐쉬를 사용하는 계층적 메모리 구조
Figure 2. Hierarchical memory organization using victim cache

III. 멀티 코어 프로세서에서 필터 캐쉬 구조의 효율성 분석

이번 장에서는 성능과 전력 소모의 관점에서 싱글 코어 프로세서와의 비교를 통해 멀티 코어 프로세서에서의 필터 캐쉬 구조의 효율성을 분석한다. 분석에 사용된 자세한 실험 방법은 5장에서 설명한다.

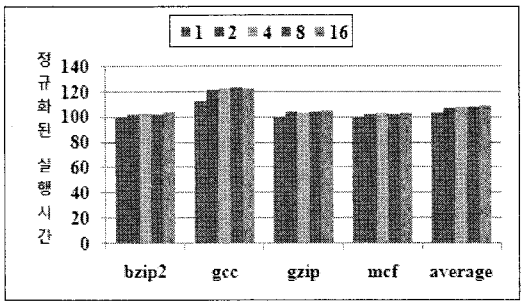

그림 3. 멀티 코어 프로세서에서 코어 개수에 따른 필터 캐쉬 구조의 정규화 한 실행 시간 (CINT)
Figure 3. Normalized execution time of the filter cache architecture for multicore processors (CINT)

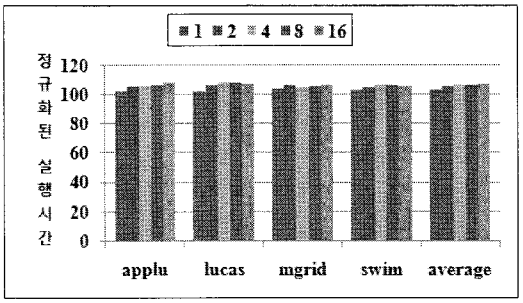

그림 4. 멀티 코어 프로세서에서 코어 개수에 따른 필터 캐쉬 구조의 정규화 한 실행 시간 (CFP)
Figure 4. Normalized execution time of the filter cache architecture for multicore processors (CFP)

3.1 성능

그림 3과 그림 4는 각각 정수형 프로그램과 부동 소수점 프로그램 실행 시, 멀티코어 프로세서의 코어 개수에 따른 필터 캐쉬 구조의 정규화 된 실행 시간을 나타낸다. 코어 개수가 1인 경우는 싱글 코어 프로세서를 의미한다. 그래프에서 각각의 막대는 동일한 개수의 코어에서 수행한 필터 캐쉬가 없는 기존의 캐쉬 구조에서의 실행 시간에 정규화한 필터 캐

쉬 구조의 실행 시간을 세로축으로 표시한다.

그래프에서 보이는 바와 같이 멀티 코어 프로세서에서의 필터 캐쉬 구조는 싱글 코어 프로세서에서의 필터 캐쉬 구조에 비해 낮은 성능을 보임을 확인할 수 있다. 필터 캐쉬 구조는 정수형 프로그램에서 평균적으로 실행시간을 코어가 1개일 때는 2.7%, 코어가 2개일 때는 6.9%, 코어가 4개일 때는 7.4%, 코어가 8개일 때는 7.6%, 코어가 16개일 때는 7.9% 증가시킨다. 부동 소수점 프로그램에서는 평균적으로 실행시간을 코어가 1개일 때는 2.3%, 코어가 2개일 때는 5.4%, 코어가 4개일 때는 5.7%, 코어가 8개일 때는 6.3%, 코어가 16개일 때는 6.5% 증가시킨다.

필터 캐쉬 구조가 캐쉬에서 소모되는 전력을 감소시키기 위해 캐쉬의 적중률을 희생하기 때문에 프로세서의 성능이 감소된다는 것은 익히 알려진 사실이지만, 본 분석을 통해 멀티 코어 프로세서에서는 성능 감소 현상이 더욱 크게 나타난다는 사실을 확인할 수 있다.

3.2 전력 소모량

그림 5와 그림 6은 각각 정수형 프로그램과 부동 소수점 프로그램 실행 시, 멀티코어 프로세서에서 코어 개수에 따른 필터 캐쉬 구조의 정규화 된 전력 소모량을 세로축으로 나타낸다. 각각의 막대는 동일한 수의 코어를 가지는 기존의 캐쉬 구조의 전력 소모량에 정규화된 필터 캐쉬 구조의 전력 소모량을 나타낸다. 그래프에 나타내는 전력 소모량은 캐쉬 메모리에서 소비되는 전력의 총합을 나타내고, 그래프에서 코어 개수가 1인 경우는 싱글 코어 프로세서를 의미한다.

그래프에서 보이는 바와 같이 멀티 코어 프로세서에서의 필터 캐쉬 구조는 싱글 코어 프로세서에서의 필터 캐쉬 구조에 비해 낮은 에너지 효율을 보인다. 필터 캐쉬 구조는 정수형 프로그램에서 캐쉬에서 소모되는 전력을 평균적으로 코어가 1개일 때는 12.9%, 코어가 2개일 때는 9.9%, 코어가 4개일 때는 9.2%, 코어가 8개일 때는 9.0%, 코어가 16개일 때는 8.9% 감소시킨다. 부동 소수점 프로그램에서는 캐쉬에서 소모되는 전력을 평균적으로 코어가 1개일 때는 7.9%, 코어가 2개일 때는 6.2%, 코어가 4개일 때는 5.7%, 코어가 8개일 때는 5.5%, 코어가 16개일 때는 5.8% 감소시킨다.

실험 결과를 통해, 싱글 코어 프로세서와 비교하여 멀티 코어 프로세서에서 필터 캐쉬 구조의 전력 감소 효과가 줄어든다는 사실을 확인할 수 있다.

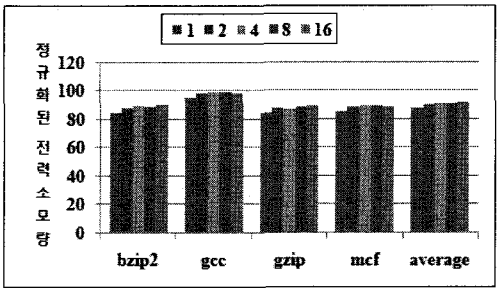

그림 5. 멀티 코어 프로세서에서 코어 개수에 따른 필터 캐쉬 구조의 전력 소모량 (CINT)
Figure 5. Normalized energy consumption of the filter cache architecture for multicore processors (CINT)

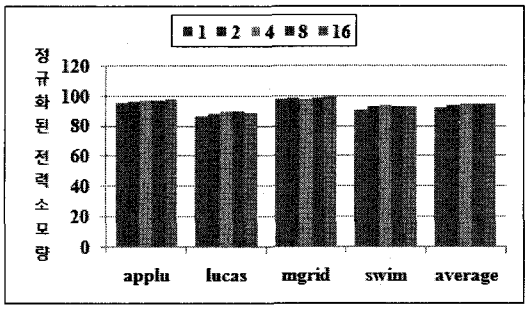

그림 6. 멀티 코어 프로세서에서 코어 개수에 따른 필터 캐쉬 구조의 전력 소모량 (CFP)
Figure 6. Normalized energy consumption of the filter cache architecture for multicore processors (CFP)

IV. 제안하는 저전력 명령어 캐쉬 구조

멀티 코어 프로세서에서 필터 캐쉬 구조의 에너지 효율성을 높이기 위해 본 논문에서는 두 개의 변형 필터 캐쉬 구조를 비교, 분석한다.

4.1 필터 캐쉬의 크기를 두 배로 하는 방안 (Double-size Filter Cache)

그림 7에서 보이는 필터 캐쉬의 크기를 두 배로 하는 방안 (DFC: Double-size Filter Cache)은 기존의 필터 캐쉬 구조와 (그림 1) 비교 시 필터 캐쉬의 크기가 두 배인 것을 제외하고는 동일하다. 제안하는 DFC 구조는 싱글 코어 프로세서의 클럭 속도와 크기를 줄이는 기법을 통해 저전력을 구현한 인텔의 아톰 프로세서와 달리 멀티 코어 프로세서의 캐쉬 구조 변화를 통해서 저전력을 구현하고자 한다. 이 구조를 프로세서에 적용하여 필터 캐쉬의 크기를 두 배로 늘려 적중률을 높임으로써 성능과 에너지 효율을 향상시키고자 한다.

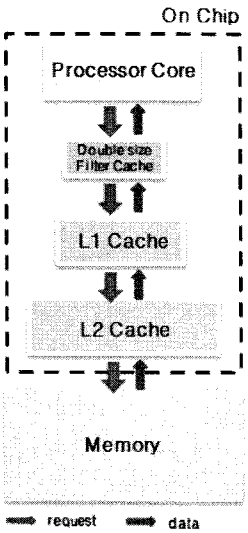

그림7. 두 배 크기의 필터 캐쉬를 사용하는 계층적 메모리 구조
Figure 7. Hierarchical memory organization using double-size filter cache

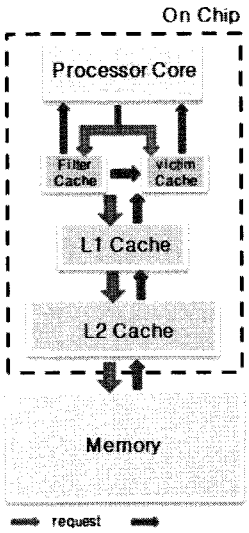

그림 8. 필터 캐쉬에 대한 희생 캐쉬를 사용하는 계층적 메모리 구조
Figure 8. Hierarchical memory organization using victim cache for filter cache

4.2 필터 캐쉬에 대한 희생 캐쉬 (Victim Cache for Filter Cache)

필터 캐쉬의 크기를 두 배로 하는 방안과 달리, 필터 캐쉬와 동일한 크기의 희생 캐쉬를 추가한 구조가 필터 캐쉬에 대한 희생 캐쉬 (VFC: Victim Cache for Filter Cache) 기법이다. 그림 8은 VFC를 적용한 계층적 메모리 구조를 나타낸다. VFC는 필터 캐쉬에서 방출되는 데이터를 저장하기 위한 캐쉬 메모리이다. 기존의 희생 캐쉬의 동작 원리와 제안하는 VFC의 차이점은 VFC에서 적중이 (Hit) 발생하는 경우에 필터 캐쉬와 VFC에서 데이터의 교환이 이루어지지 않는다는 것이다.

기존의 희생 캐쉬 구조에서는 희생 캐쉬에서 적중이 발생하는 경우에 메인 캐쉬와 희생 캐쉬 사이에 데이터 교환이 발생한다. 기존의 희생 캐쉬 구조에서의 데이터 교환은 복잡도 증가를 가져올 수 있다. 제안하는 구조에서는 필터 캐쉬와 희생 캐쉬 사이에 발생하는 데이터 교환으로 인한 회로의 복잡도 증가를 방지하기 위해 필터 캐쉬와 VFC 사이에서는 데이터 교환을 하지 않는다. 제안하는 구조를 통해 필터 캐쉬에서 방출되는 데이터에 대한 재접근 요청 시 하위 캐쉬를 접근하는 대신에 VFC에 대한 접근을 통해 프로세서 코어의 요청을 해결함으로써 메인 캐쉬에 대한 접근을 더욱 감소시켜 전력 소모를 줄이고자 한다.

V. 모의실험

성능과 전력 효율성 측면에서 제안하는 캐쉬 구조의 특성을 측정하기 위해 본 논문에서는 SimpleScalar [12] 시뮬레이터와 CACTI [13] 툴을 사용한다. 시뮬레이터의 입력 프로그램으로는 SPEC CPU2000 [14] 벤치마크 프로그램을 사용한다.

표 1. 프로세서 실험 인자
Table 1. Processor Configurations

실험인자	값
Number of cores	2, 4, 8, 16
프로세서 코어 특성	
Functional Units	4 integer ALUs, 4 FP Alus, 1 integer multiplier/divider, 1 FP multiplier/divider
Filter Cache	512B, fully-associative, 1 cycle latency
Double-size Filter Cache	1KB, fully-associative, 1 cycle latency
VFC	512B, fully-associative, 1 cycle latency
Level-1 Inst. Cache	32KB, 4-way, 32bytes lines, 1 cycle latency
Level-1 Data Cache	32KB, 4-way, 32bytes lines, 1 cycle latency

5.1 성능

그림 9와 그림 10은 각각 정수형 프로그램과 부동 소수점 프로그램에 대해서 코어의 개수에 따른 실행 시간을 보여준다. 그래프에서 CFC (Conventional Filter Cache)는 기존의 필터 캐쉬 구조를 의미한다. 필터 캐쉬의 크기를 두 배로 하는 방안은 DFC (Double-size Filter Cache), 필터 캐쉬에 대한 희생 캐쉬 방안은 VFC (Victim cache for Filter Cache)로 표시된다. 그래프에서 각 구조의 실행 시간은 CFC의 실행 시간에 정규화되어 세로축으로 표시한다.

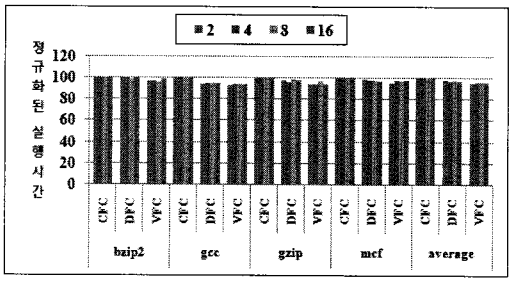

그림 9. 정규화 한 실행시간 (CINT)
Figure 9. Normalized execution time (CINT)

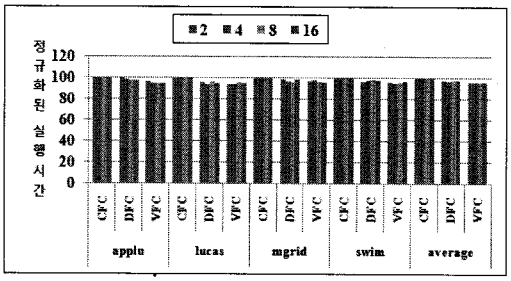

그림 10. 정규화 한 실행시간 (CFP)
Figure 10. Normalized execution time (CFP)

그래프에서 보이는 바와 같이 새롭게 적용한 DFC와 VFC 모두 멀티 코어 프로세서에서 CFC에 비해 우수한 성능을 보임을 알 수 있다. 정수형 프로그램에서 DFC는 CFC와 비교하여 평균적으로 코어가 2개일 때는 2.7%, 코어가 4개일 때는 3.3%, 코어가 8개일 때는 2.5%, 코어가 16개일 때는 2.9%의 실행 시간을 감소시킨다. 부동 소수점 프로그램에서 DFC는 평균적으로 코어가 2개일 때는 2.9%, 코어가 4개일 때는 3.7%, 코어가 8개일 때는 3.1%, 코어가 16개일 때는 3.1%의 실행 시간을 CFC에 비교하여 향상시킨다. 필터 캐쉬에 대한 희생 캐쉬를 사용하는 VFC의 경우에는 비교하는 구조들 중에서 가장 우수한 성능을 보여준다. 정수형 프로그램에서 VFC는 평균적으로 코어가 2개일 때는 5.4%, 코어가 4개일 때는 4.5%, 코어가 8개일 때는 4.3%, 코어가 16개일 때는 4.4%의 실행시간을 감소시킨다. 부동 소수점 프로그램에서는 평균적으로 CFC와 비교하여 코어가 2개일 때는 4.6%, 코어가 4개일 때는 4.9%, 코어가 8개일 때는 4.8%, 코어가 16개일 때는 4.9%의 실행 시간의 감소를 보인다.

5.2 전력 소모량

그림11, 12는 정수형 프로그램과 부동 소수점 프로그램 수행 시 각 구조에 따른 전력 소모량을 보여준다. 그래프에서 나타내는 전력 소모량은 캐쉬 메모리에서 소모되는 전력의 총 합을 나타낸다. 각각의 막대는 CFC 구조에서 소모되는 전력에 정규화하여 계산된 값을 세로축으로 표시한다.

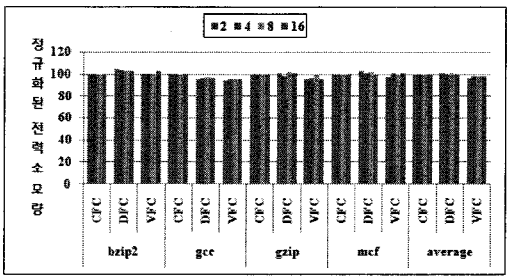

그림 11. 정규화 한 전력 소모량 (CINT)
Figure 11. Normalized energy consumption (CINT)

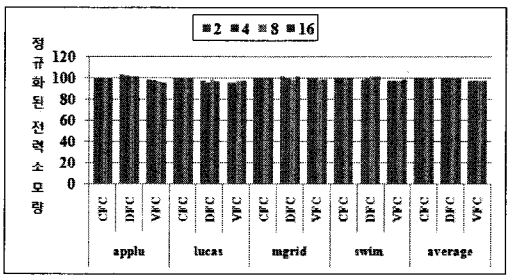

그림 12. 정규화 한 전력 소모량 (CFP)
Figure 12. Normalized energy consumption (CFP)

그래프에서 보이는 바와 같이 CFC와 DFC는 에너지 효율에 있어서 큰 차이를 보이지 않는다. VFC구조는 비교하는 구조들 중에서 가장 좋은 에너지 효율을 보여준다. 정수형 프로그램에서 VFC는 CFC 구조와 비교하여 코어가 2개일 때는 3.4%, 코어가 4개일 때는 2.1%, 코어가 8개일 때는 1.7%, 코어가 16개일 때는 1.9%의 전력 소모를 줄인다. 부동 소수점 프로그램에서는 평균적으로 코어가 2개일 때는 2.3%, 코어가 4개일 때는 2.6%, 코어가 8개일 때는 2.6%, 코어가 16개인 경우에는 2.7%의 전력 소모를 감소시킨다.

프로그램의 지역성(Locality)으로 인해 필터 캐쉬에서 방출되는 데이터가 프로세서 코어로부터 재요청될 확률이 상당히 높기 때문에 이러한 데이터들을 희생 캐쉬에 저장함으로써 메인 캐쉬에 대한 접근 횟수를 가장 줄여서 전력 소모 측면에서 우수한 결과를 보임을 알 수 있다.

VI. 결론

본 논문에서는 멀티 코어 프로세서의 명령어 캐쉬에서 소모되는 전력을 감소시키기 위하여 새로운 저전력 필터 캐쉬 구조를 제안하였다. 제안된 구조는 필터 캐쉬에 대한 희생 캐쉬를 추가함으로써 필터 캐쉬에서 방출되는 데이터가 프로세서 코어로부터 재요청되는 경우에 메인 캐쉬가 접근되는 것을 방지함으로써 기존 필터 캐쉬 구조와 비교하여 전력 소모를 감소시킬 수 있었다. 실험결과에 따르면 제안된 필터 캐쉬에 대한 희생 캐쉬 구조는(VFC) 기존의 필터 캐쉬 구조와 비교하였을 때, 명령 어 캐쉬에서의 전력 소모를 최대 3.4% 줄였다. 또한, 제안한 구조는 기존의 필터 캐쉬 구조와 비교하여 멀티코어 프로세서의 성능을 최대 5.4% 향상시켰다. 그러므로 제안한 VFC 구조는 멀티코어 프로세서의 명령어 캐쉬에서 소모되는 전력을 감소시키기 위한 우수한 방법 중 하나로 생각할 수 있다.

참고문헌

[1] 공준호, 최진항, 이종성, 정성우, "인텔 펜티엄 4와 코어 2 듀오의 실행시간과 파워소모량 효율성 비교," 한국컴퓨터정보학회 논문지, 제 13권, 제 7호, 165-172쪽, 2008년 12월.

[2] 양나라, 김종면, 김철홍, "임베디드 시스템에서 후방 분기 명령어 정보를 이용한 저전력 명령어 캐쉬 설계 기법," 한국컴퓨터정보학회 논문지, 제 13권, 제 6호, 33-39쪽, 2008년 11월.

[3] S. Segars, "Low Power Design Techniques for Microprocessors," Proceedings of International Solid-State Circuits Conference, 2001.

[4] J. Kin, M. Gupta, and W. Mangione-Smith, "The Filter Cache: An Energy Efficient Memory Structure," Proceedings of the International Symposium on Microarchitecture, pp. 184-193, 1997.

[5] 곽종욱, "모드 선택 비트를 사용한 필터 캐시 예측기," 전자공학회 논문지, 제 46권, 제 5호, 539-551쪽, 2009년 9월.

[6] D. H. Albonesi, "Selective Cache Ways: On-demand Cache Resource Allocation," Proceedings of the International Symposium on Microarchitecture, pp. 70-75, 1999.

[7] K. Inonue, T. Ishihara, and K. Murakami, "Way-predicting Set-associative Cache for High Performance and Low Energy Consumption," Proceedings of the International Symposium on Low Power Electronics and Design, pp. 273-275, 1999.

[8] M. Powell, A. Agarwal, T. N. Vijaykumar, B. Falsafi, and K. Roy, "Reducing Set-associative Cache Energy via Way-prediction and Selective Direct-mapping," Proceedings of the International Symposium on Microarchitecture, pp. 54-65, 2001.

[9] C. H. Kim, S. W. Chung, and C. S. Jhon, "PP-cache: A Partitioned Power-aware Instruction Cache Architecture," Microprocessors and Microsystems, Vol. 30, pp. 268-279, 2006.

[10] 이광용, 박호준, 김동환, 강동욱, 김재명, 박승민, "멀티코어 기술 및 산업 동향," 정보통신연구진흥원 학술정보, 주간기술동향 1295호, 2007년.

[11] N. P. Jouppi, "Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers," Proceedings of 17th Annual International Symposiumon Computer Architecture, pp. 364-373,1990.

[12] D. Burger, T. M. Austin, and S. Bennett, "Evaluating Future Micro-Processors: the SimpleScalar tool set," Tech. Report TR-138, Univ. of Wisconsin-Madison Computer Sciences Dept., 1997.

[13] P. Shivakumar and N. P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," TR-WRL-2001-2, 2001.

[14] SPEC CPU 2000 Benchmarks, *http://www.specbench.org*

저 자 소 개

박 영 진

2009: 전남대학교 전자컴퓨터공학부 학사

2009: 전남대학교 전자컴퓨터공학부 석사과정 입학

관심분야: : 저전력 설계, 저온도 설계, 컴퓨터 구조

김 종 면

1995: 명지대학교 전기공학사

2000: University of Florida ECE 석사

2005: Georgia Institute of Technology ECE 박사한국대학교 공학박사

2005 - 2007:

삼성전자 반도체 총괄 책임 연구원

2007 - 현재:

울산대학교 컴퓨터정보통신공학부 교수

관심분야: 프로세서 설계, 임베디드 시스템, SoC, 컴퓨터 구조, 병렬처리

김 철 홍

1998: 서울대학교 컴퓨터공학사

2000: 서울대학교 대학원 컴퓨터 공학부 석사

2006: 서울대학교 대학원 전기컴퓨터 공학부 박사

2005 - 2007:

삼성전자 반도체 총괄 책임 연구원

2007 - 현재:

전남대학교 전자컴퓨터공학부 교수

관심분야: 임베디드 시스템, 컴퓨터 구조, SoC 설계, 저전력 설계