

멀티미디어 응용을 위한 저전력 데이터 캐쉬 구조 및 마이크로 아키텍처 수준 관리기법

양 훈 모[†] · 김 정 길[†] · 박 기 호^{**} · 김 신 덕^{***}

요 약

최근 디지털 멀티미디어 응용기기는 휴대 편의성은 물론 하나의 기기에서 다양한 멀티미디어 데이터 처리를 가능하게 하는 기능적 집적이 이루어지고 있다. 이와 같은 추세는 기기가 처리해야 하는 데이터 양의 증가와 이를 수행하기 위하여 요구되는 온칩 메모리의 크기 및 연산 유닛의 고성능화를 요구하여 전력 소비량의 증가를 유발시킨다. 연산 엔진에서 사용되는 대표적인 온칩 메모리인 캐쉬는 전력 사용에 있어서 중요한 비율을 차지하는 구조로 저전력 설계를 위한 구조적 개선의 주요 대상이다. 본 논문에서는 멀티미디어 응용을 수행하는 연산 엔진의 데이터 캐쉬에서 소비되는 전력을 감소시키기 위하여 멀티미디어 응용의 데이터 사용 특성을 파악하여 이 특성을 전력소비를 감소시키는 목적으로 활용 가능한 분할된 캐쉬구조를 제안한다. 그리고 각각의 분할된 캐쉬에 대하여 특정 주소 영역의 데이터 참조를 고정시킴으로써 얻을 수 있는 전력 소비면의 성능 향상을 평가한다. 시뮬레이션 결과 제안하는 캐쉬 구조는 같은 크기의 직접사상 캐쉬, 2중연관 캐쉬, 4중연관 캐쉬에 대해 유사한 성능을 나타내면서, 각각의 기존 캐쉬 구조와 비교하였을 경우 33.2%, 53.3% 및 70.4%만큼 감소된 전력으로 동작 가능하다.

키워드 : 저전력, 마이크로 아키텍처, 메모리 시스템, 데이터 캐쉬, 캐쉬 분할, 멀티미디어 응용

Low-Power Data Cache Architecture and Microarchitecture-level Management Policy for Multimedia Application

Hoon-Mo Yang[†] · Cheong-Gil Kim[†] · Gi-Ho Park^{**} · Shin-Dug Kim^{***}

ABSTRACT

Today's portable electric consumer devices, which are operated by battery, tend to integrate more multimedia processing capabilities. In the multimedia processing devices, multimedia system-on-chips can handle specific algorithms which need intensive processing capabilities and significant power consumption. As a result, the power-efficiency of multimedia processing devices becomes important increasingly. In this paper, we propose a reconfigurable data caching architecture, in which data allocation is constrained by software support, and evaluate its performance and power efficiency. Comparing with conventional cache architectures, power consumption can be reduced significantly, while miss rate of the proposed architecture is very similar to that of the conventional caches. The reduction of power consumption for the reconfigurable data cache architecture shows 33.2%, 53.3%, and 70.4%, when compared with direct-mapped, 2-way, and 4-way caches respectively.

Key Words : Low-power, Microarchitecture, Memory System, Data Cache, Cache Partitioning, Multimedia Application

1. 서 론

디지털 멀티미디어 데이터의 처리 기능은 오늘날의 휴대용 기기에 있어서 중요한 기능적 요소이다. 대표적으로 휴대전화기를 예로 들 수 있는데, 정지화상의 촬영, 음성데이터

를 포함하는 동영상의 촬영 및 디지털 오디오 재생 등 디지털 영상/음성 신호를 복합적으로 처리하는 기능을 가지고 있다. 또한 하나의 기기에서 다양한 표준의 멀티미디어 데이터를 처리 가능하도록 하는 기능의 집적화가 이루어지고 있는 추세이다. 이와 같은 현상은 휴대용 기기에 내장되는 연산 엔진의 고성능화를 요구한다. 또한 멀티미디어 데이터의 특성상 연산 엔진이 처리하는 데이터의 크기가 증가하여 이를 처리하기 위해 저장 장소로 사용되는 온칩 메모리의 크기 또한 증가하여야 한다.

캐쉬는 연산 엔진에서 사용되는 대표적인 온칩 메모리 구

* 이 연구는 학술진흥재단(KRF-2004-041-D00545)의 지원에 의해 수행되었음.

† 준 회 원 : 연세대학교 공과대학 컴퓨터과학과 박사과정

** 정 회 원 : 삼성전자 SOC 연구소 Processor Architecture Lab. 책임연구원

*** 정 회 원 : 연세대학교 공과대학 컴퓨터과학과 교수

논문접수 : 2006년 2월 20일, 심사완료 : 2006년 4월 24일

조로서, 정적 메모리(SRAM)로 구성되고 참조가 빈번히 발생하는 특징을 가지므로 연산 엔진이 소비하는 전력중 큰 부분을 차지한다. 이러한 이유로 저전력 구조를 설계하는데 있어서 캐쉬는 주요한 개선 대상이 된다.

본 논문에서는 캐쉬를 다수의 독립된 작은 캐쉬로 분할하여 응용 프로그램의 요구에 따라 각각의 캐쉬를 특정 주소 영역의 데이터를 저장하는데 사용하도록 하는 캐쉬 구조를 제안하며 이러한 캐쉬 구조가 사용되었을 경우 전력 소비면의 효율성을 평가한다.

성능 평가 수행을 위하여 6개의 음성/영상 처리 프로그램과 mpeg2 비디오 인코딩/디코딩 프로그램을 사용하여 데이터 참조의 주소 영역의 편중 및 참조 패턴을 알아낸 후 이를 각각의 독립된 캐쉬에 할당하여 시뮬레이션을 수행하였다. 제안하는 캐쉬 구조는 기존의 직접 사상 캐쉬(direct-mapped cache), 2중 연관 캐쉬(2-way set associative cache), 4중 연관 캐쉬(4-way set associative cache)에 대하여 뒤떨어지지 않는 성능을 보이면서 전력 소비면에 있어서는 각각의 기존 캐쉬 구조에 대해 33.2%, 53.3% 및 70.4%만큼 감소된 전력으로 동작하는 것으로 나타났다.

본 논문의 구성은 다음과 같다. 2장에서는 저전력 및 고성능 멀티미디어 응용 프로그램을 위한 메모리 시스템 구조에 대하여 조사한 내용을 기록하였다. 3장에서는 멀티미디어 응용 프로그램의 특성 서술 및 본 논문에서 제안하는 구조를 제시하고 동작을 기술하였다. 4장에서는 시뮬레이션을 통하여 기존의 단일 캐쉬 구조와 제안하는 캐쉬 구조와의 성능을 비교 및 전력소비 측면의 효율성을 기술하였고, 5장에서는 결론을 맺는다.

2. 관련 연구

멀티미디어 응용 프로그램은 수행 흐름에 있어서 실행 시간동안 발생하는 동적인 변화가 상대적으로 단순하여 제어 명령어에 의한 비중이 적은 점 및 일련의 입력 데이터에 대한 연속적인 알고리즘들의 적용을 통하여 최종 결과값을 출력하는 형태로 데이터 의존적인 점을 다른 응용에 대한 차별성으로 들 수 있다.

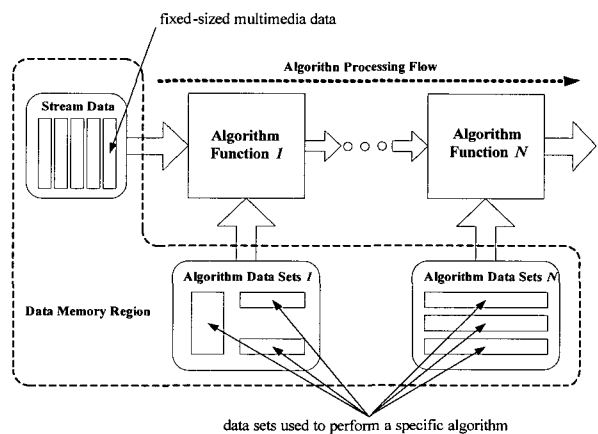
이러한 멀티미디어 응용을 수행하는 기기의 메모리 시스템 성능 향상 및 전력 소비를 감소시키기 위한 연구는 다양한 방법론에 의해 수행되었다. 우선, 기존의 일반적인 메모리 시스템에서 멀티미디어 응용 프로그램의 성능을 SPECint95와 비교 평가한 결과 기존의 메모리 시스템이 멀티미디어 응용 프로그램에 대해서 잘 적용되는 점 및 그 이유를 응용 프로그램의 분석을 통하여 찾아낸 연구가 존재하며[1], 저전력 설계 기술의 전반적인 분류 및 재구성 가능한 하드웨어를 기반으로 하는 방법론에 대한 조사를 수행한 연구[2]가 존재한다. 기존의 메모리 계층구조에 추가적인 구조적 개선을 통하여 저전력 메모리 시스템을 구현하고자 하는 것으로 L2캐쉬와 메인 메모리 사이에 Energy-Saver Buffers(ESB)를 사용하여 저전력 모드로 동작할 때 메인 메모리 모듈에

서 발생하는 재동기화(resynchronization) 오버헤드를 감소시키는 연구[3]가 있으며 데이터 버퍼를 사용하여 하위 계층 메모리로의 접근을 줄임으로써 저전력 모드의 유지시간을 증가시키는 방법[4]이 존재한다. 비디오 응용에 특화된 연구로는 범용 프로세서에서 선인출(prefetch) 기법을 사용하여 성능 증가를 목적으로 하는 연구[5]와 mpeg2 비디오 디코더의 데이터 사용 패턴의 특징을 분석한 연구[6]가 존재한다. 마지막으로 본 논문이 기반으로 하는 지역 기반 캐쉬(region-based caching)[7]는 데이터 캐쉬를 스택 캐쉬(stack cache)와 전역 캐쉬(global cache)로 분리하는 두개의 캐쉬를 두어 스택 데이터와 그 이외의 데이터를 각각의 캐쉬에 분산시켜 성능 향상 및 저전력 효과를 목적으로 하는 연구이다.

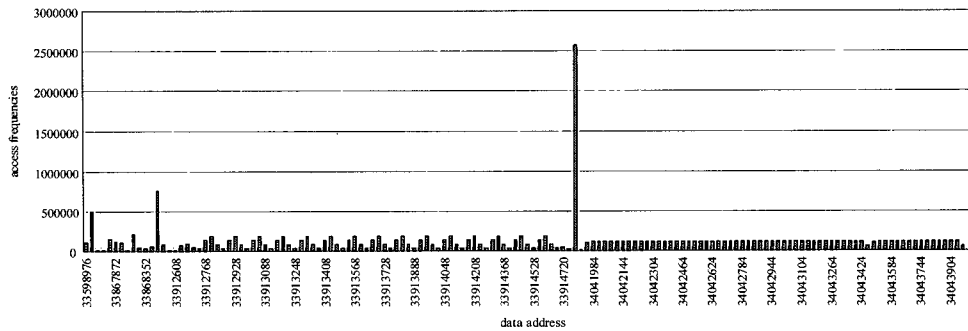
이와 같은 관련 연구들은 주로 고성능 또는 범용 메모리 시스템 구조에 기반하는 것으로, 구조 설계시 자원의 사용 및 전력 소비 요구량이 제한되는 내장형 시스템으로 적용하기에는 부적합한 면이 존재한다. 본 논문에서는 언급된 관련 연구 중 지역 기반 캐쉬와 같은 방법론을 데이터 영역에 대해 세분화 시켜 적용함으로써 보다 분산된 데이터 참조를 유도하여 데이터 메모리 참조의 전력소비를 감소시키는 구조를 제안한다.

3. 분할 데이터 캐쉬 구조

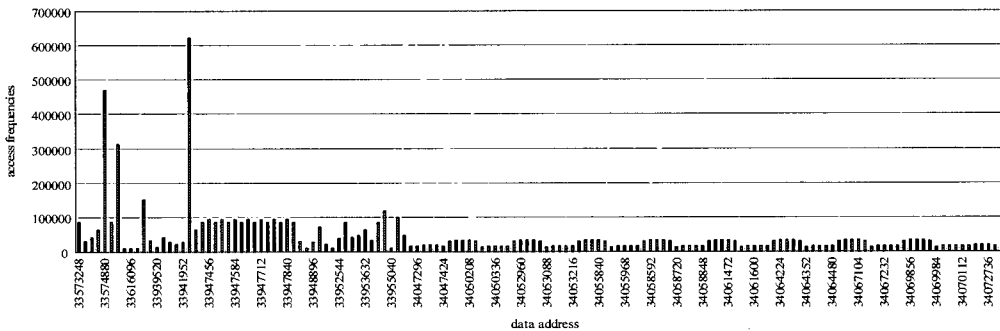
멀티미디어 응용 프로그램의 특징은 일정 크기 단위를 갖는 일련의 입력 데이터가 다수의 알고리즘 함수를 순서대로 적용 받으면서 변형되어 최종적인 출력 데이터를 생성하는 것이다. 이와 같은 어떤 하나의 알고리즘 함수 수행 과정을 하나의 기능 블록으로 가정하였을 경우, 해당 기능 블록에서 사용되는 데이터는 (그림 1)과 같이 연산 처리 대상인 멀티미디어 스트림 데이터(예: 샘플링되어 표준화된 오디오/비디오 데이터 등)와 알고리즘 수행과정에서 필요한 부가적인 데이터(예: 알고리즘에 따라 고정된 필터 계수 테이블 등) 크게 두가지로 나뉘는 것으로 생각할 수 있다.



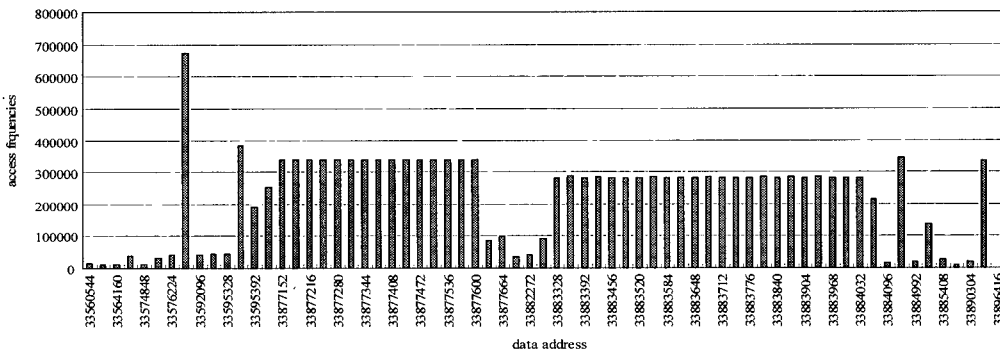
(그림 1) 멀티미디어 응용 프로그램의 일반적인 데이터 사용 특성



(그림 2) mad의 데이터 참조 패턴



(그림 3) mpeg2enc의 데이터 참조패턴



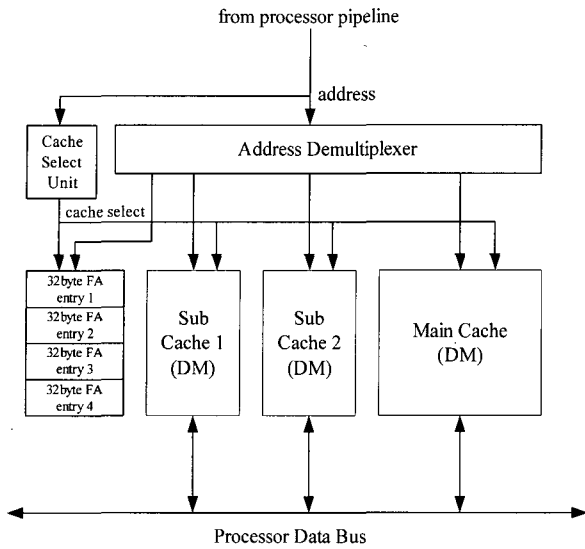
(그림 4) mpeg2dec의 데이터 참조패턴

이와같이 멀티미디어 응용 프로그램의 특징은 처리 대상 데이터 스트림이 연속적으로 일정한 순서의 알고리즘을 적용 받으면서 데이터 처리가 발생하는 것으로 요약할 수 있는데, 그 과정에서 적용되는 알고리즘의 순서 및 연산의 종류가 동적으로 큰 차이를 보이며 변화하는 경우가 발생할 가능성이 매우 높다. 이러한 데이터 사용 특성에 대한 가정을 확인하기 위하여 몇가지 멀티미디어 응용 프로그램의 벤치마크를 시뮬레이터에서 실행하면서 데이터 참조 패턴을 모니터링하였다. 그 결과 (그림 2)~(그림 4)와 같은 형태의 주소영역-참조 횟수의 상관 관계를 확인할 수 있다.

(그림 2)~(그림 4)에서 확인할 수 있듯이 각각의 벤치마크에 대해 특정 주소 영역의 데이터는 해당 영역에 대해 데이터 참조 횟수 및 그 반복성이 동일한 특성을 보인다. 또한 특정 주소값에 해당하는 데이터에 대한 참조 횟수는 다

른 데이터에 비해 그 참조 횟수가 월등히 많은 경우가 존재함을 확인할 수 있다. 이와 같이 동일한 패턴을 보이는 각 주소 영역 및 특정 주소의 데이터를 작은 크기의 독립적인 캐쉬에 할당하여 캐쉬 참조에 있어서의 전력 소비를 감소시키는 것이 본 연구의 목적이다.

이와 같은 방법으로 캐쉬를 사용하기 위하여 (그림 5)와 같은 분할된 형태의 캐쉬 구조를 제안한다. 제안하는 구조는 4개의 엔트리를 갖는 완전연관사상 버퍼(fully-associative buffer) 1개와 2개의 보조 직접사상 캐쉬(sub direct-mapped cache), 1개의 주 직접사상 캐쉬(main direct-mapped cache), 그리고 데이터 참조 주소를 기준으로 3개의 직접사상 캐쉬 및 1개의 완전연관사상 버퍼 중 하나를 선택하여 구동시키는 캐쉬 선택 유닛(cache select unit) 및 주소 분배기(address demultiplexer)로 구성된다.



(그림 5) 분할된 캐쉬 구조

4개의 엔트리를 갖는 완전연관사상 버퍼는 알고리즘 수행 중 가장 사용 빈도수가 높은 4개의 32바이트 데이터를 저장한다. 2개의 보조 직접사상 캐쉬는 일정한 사용 패턴을 보이는 특정 데이터 주소 영역에 대한 캐쉬로 할당되어 사용된다. 2개의 보조 직접사상 캐쉬에는 주로 데이터 주소가 순차적으로 증가하며 사용되는 데이터 주소 영역이 할당된다. 한개의 주 직접사상 캐쉬는 완전연관사상 버퍼와 보조 캐쉬에 할당된 주소 영역 이외의 데이터를 참조를 처리하는데 사용된다. 이와같이 서로 다른 주소 영역을 담당하도록 사용되는 캐쉬는 데이터 참조가 발생시 해당 데이터 주소를 캐쉬 구동 전에 미리 확인하여 현재 참조하는 데이터 주소를 담당하는 캐쉬를 구동시키는 기능이 필요한데, 이러한 기능은 캐쉬 선택 유닛에서 수행한다. 캐쉬 선택 유닛은 내부에 데이터 주소값을 저장할 수 있는 레지스터를 소유하여 데이터 참조 요청이 있을 경우 해당 주소를 레지스터와 비교하여 사용될 캐쉬를 선택하는 기능을 수행한다. 캐쉬 선택 유닛의 레지스터에 특정 주소값을 설정하는 작업은 응용 프로그램이 실행시 알고리즘 수행 전에 초기화 하도록 하며, 이러한 기능은 연산 엔진에서 지원하는 명령어에 의해 수행되는 것으로 가정한다.

제안하는 캐쉬 구조의 동작흐름은 다음과 같다.

- 1) 응용 프로그램 구동시 데이터 주소 영역 분할을 위한 초기화를 수행한다. 이 때 연산 엔진이 지원하는 캐쉬 할당 제어 명령어를 사용하여 캐쉬 선택 유닛의 레지스터에 주소값을 기록한다.
- 2) 초기화 이후 데이터 참조 주소는 각각 캐쉬 선택 유닛과 주소 분배기로 입력되며 캐쉬 선택 유닛은 초기화 시 설정된 주소값과 현재 참조 주소값을 비교하여 사용될 캐쉬에 대해 구동 신호를 보내게 된다.
- 3) 구동 신호를 받은 캐쉬에서만 참조가 발생하여 요청받은 데이터를 전송한다.

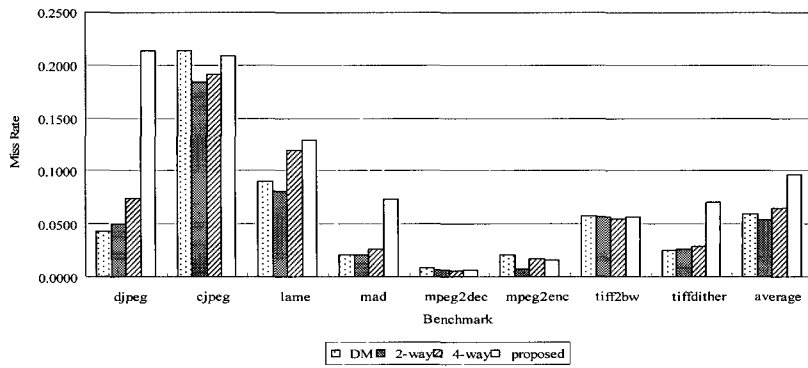
초기화시 각각의 캐쉬에 대한 데이터 영역 할당은 프로그램에서 지원 명령어를 실행하여 수행되며, 영역 할당의 판단 근거는 응용 프로그램 개발시 알고리즘 분석에 의한 데이터 셋의 분류를 통해 미리 결정되는 것으로 가정한다. 추가적으로, 분할되어 관리되는 데이터 주소 영역의 변경이 필요하여 캐쉬 선택 로직의 주소 레지스터 값을 변경할 경우 두개의 보조캐쉬 및 완전연관사상 버퍼는 각각 갱신된 데이터에 대하여 메인 메모리로의 후기록(write-back)을 수행 해야 한다. 이와 같은 후기록이 빈번히 수행될 경우 그 오버헤드가 성능 저하를 발생시킬 가능성을 가지지만 특정 응용 프로그램을 고정적으로 수행하는 멀티미디어 내장형 시스템의 경우 이러한 재할당 현상은 범용 시스템에 비하여 그 빈도가 매우 적을 것으로 예상되며, 성능평가 결과 그 가능성이 매우 낮았다. 따라서 초기화 작업은 프로그램 시작시 1회만 수행하는 것으로 가정한다.

4. 성능 평가

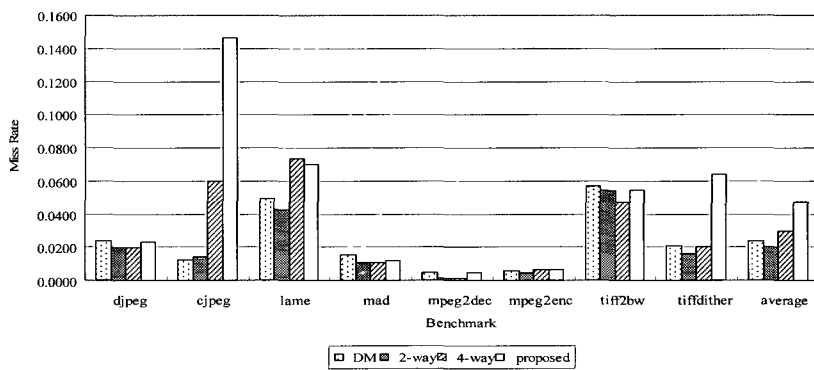
제안하는 캐쉬 구조의 성능을 평가하기 위하여 내장형 시스템을 평가하는데 사용되는 벤치마크인 mibench benchmark suite[8, 9]의 jpeg, cjpeg, lame, mad, tiff2bw 및 tiffdither 총 6개의 음성/영상 처리 프로그램과 mpeg2 인코딩/디코딩 프로그램인 mpeg2enc와 mpeg2dec[10]를 arm바 이너리 코드로 컴파일하여 사용하였다.

먼저 simplecalar-arm 0.2[11]프로세서 시뮬레이터를 사용하여 8개의 벤치마크를 각각 1억개의 명령어를 수행하는 동안의 데이터 참조 주소 및 횟수를 모니터링하였고, 데이터 캐쉬 참조 주소 트레이스를 캐쉬 시뮬레이터에 입력하여 제안하는 캐쉬 구조의 성능을 평가하였다. (그림 6)~(그림 9)는 기존의 직접사상 캐쉬, 2중연관 캐쉬, 4중연관 캐쉬에 대하여 제안하는 캐쉬 구조의 성능을 나타낸 것으로 각각 데이터 캐쉬의 크기가 4KB, 8KB, 16KB 및 32KB일 경우에 대한 캐쉬의 참조 실패율(miss rate)을 나타낸 것이다. 그 결과 캐쉬의 크기가 4KB, 8KB일 경우에는 각각의 보조 캐쉬 및 주 캐쉬의 상대적으로 작은 크기로 인하여 기존의 캐쉬에 비해서 비교적 높은 참조 실패율을 나타내지만(그림 6), (그림 7)), 캐쉬의 크기가 증가할수록 그 차이는 감소하여 크기가 16KB 이상이 될 경우 평균적으로 성능상의 차이가 거의 없음을 보여주었다.(그림 8)~(그림 10)

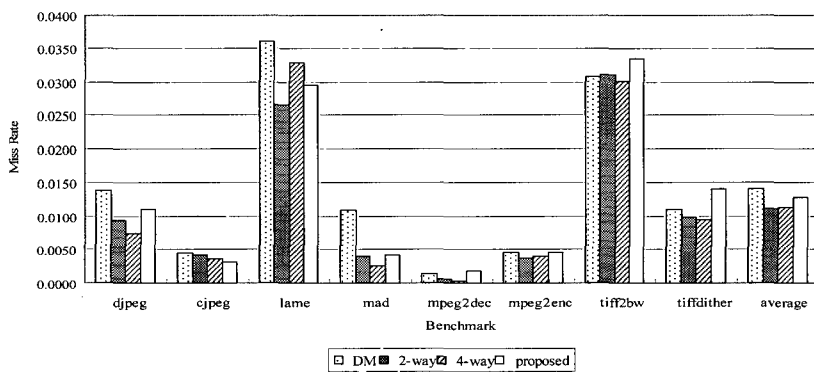
제안하는 캐쉬 구조의 참조 실패율을 기준으로 하는 성능 이외에 전력 소비 감소효과를 측정하기 위하여 캐쉬 구조에 대한 여러가지 인수들을 계산하여 주는 툴인 CACTI 3.2[12, 13]을 사용하여 캐쉬의 소비전력을 평가하기 위하여 사용되는 인수값을 획득하였다. 본 연구에서 사용된 CACTI의 공정값은 0.13μm 이다. CACTI로부터 얻어진 인수값을 사용하여 캐쉬 시뮬레이터를 실행하면서 버퍼와 각 캐쉬에 발생하는 동작 종류의 횟수를 사용하여 전력 소비값을 계산하였다. 각각의 벤치마크에 대하여 캐쉬의 크기별로 전력소비량을 비교한 결과(그림 11)~(그림 14)및 평균값을 (그림 15)확인할 수 있다.



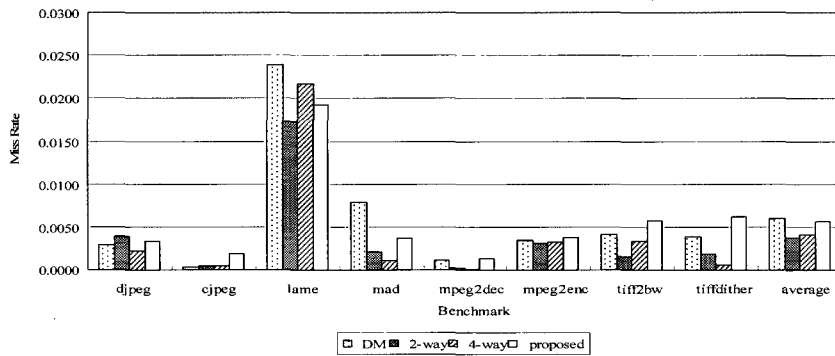
(그림 6) 캐쉬의 크기가 4KB일 경우의 성능 비교



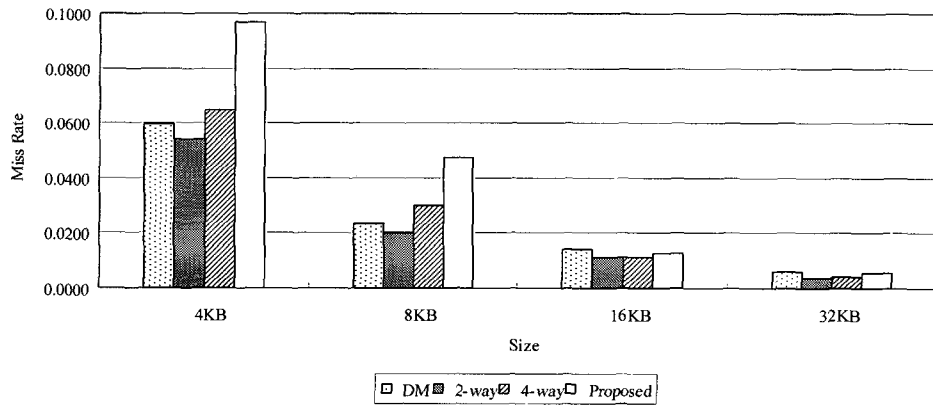
(그림 7) 캐쉬의 크기가 8KB일 경우의 성능 비교



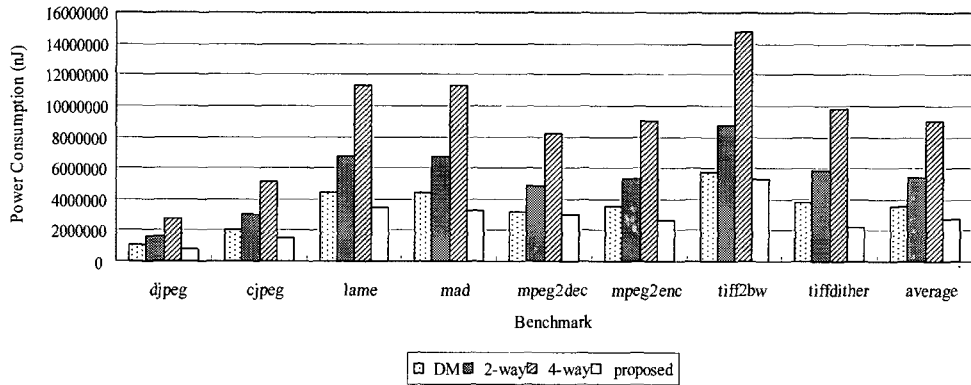
(그림 8) 캐쉬의 크기가 16KB일 경우의 성능 비교



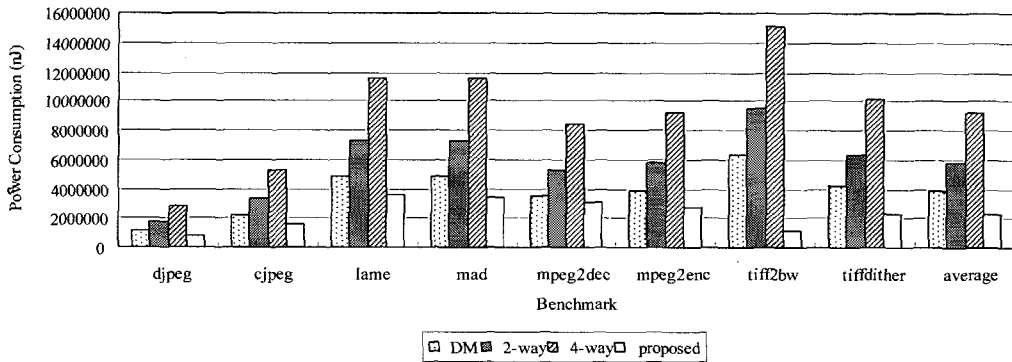
(그림 9) 캐쉬의 크기가 32KB일 경우의 성능 비교



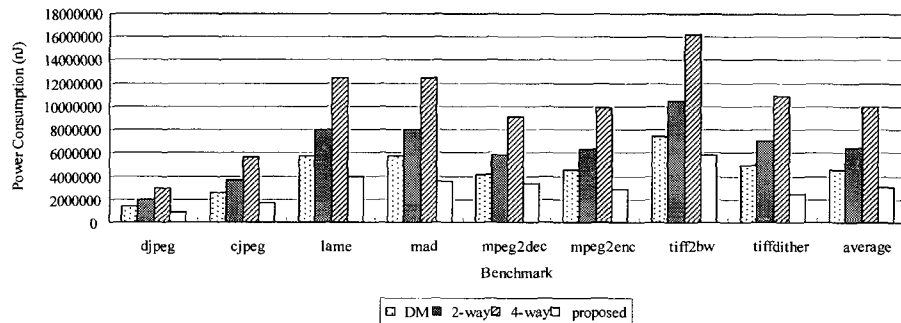
(그림 10) 기존의 캐쉬 구조와 제안하는 캐쉬 구조의 평균 성능 비교



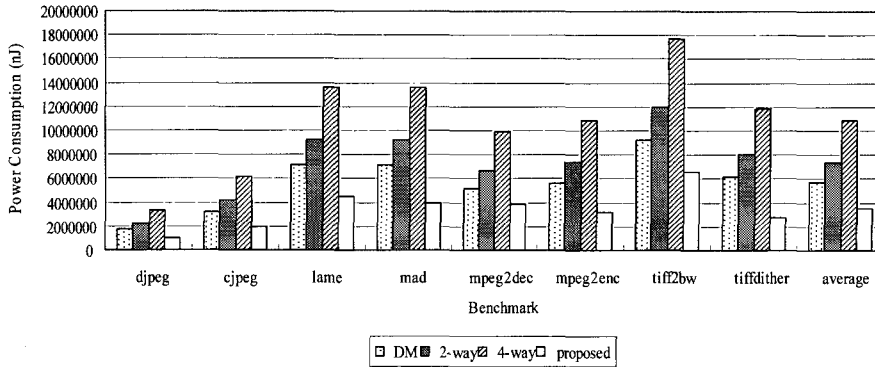
(그림 11) 캐쉬의 크기가 4KB일 경우의 전력소비 비교(nJ)



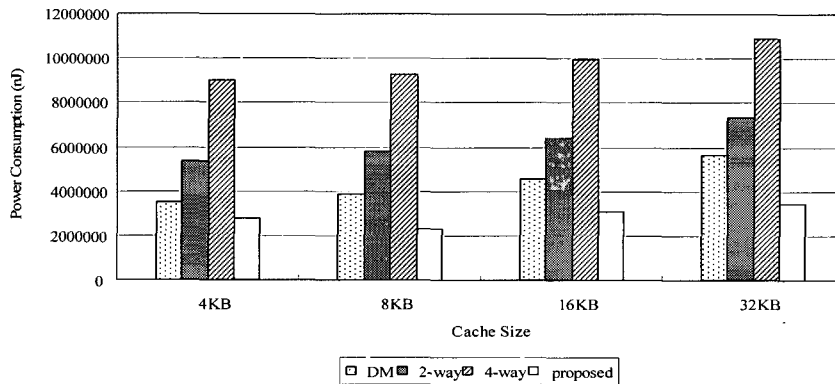
(그림 12) 캐쉬의 크기가 8KB일 경우의 전력소비 비교(nJ)



(그림 13) 캐쉬의 크기가 16KB일 경우의 전력소비 비교(nJ)



(그림 14) 캐쉬의 크기가 32KB일 경우의 전력소비 비교(nJ)



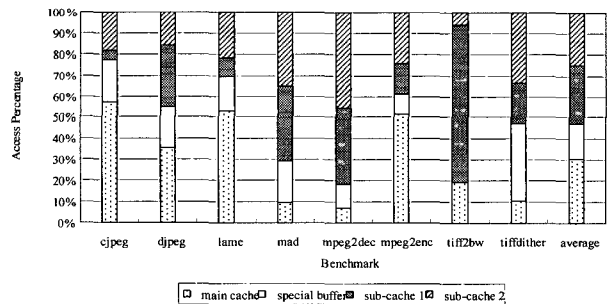
(그림 15) 기존 캐쉬구조에 대한 제안하는 캐쉬구조의 평균 전력소비 비교(nJ)

제안하는 캐쉬 구조의 평균 전력 소비량을 기존의 캐쉬 구조와 비교해 본 결과 같은 크기의 direct-mapped 캐쉬의 66.77%, 2-way set associative 캐쉬의 46.73%, 4-way set associative 캐쉬의 29.65%만큼의 전력 소비량을 보인다.<표 1>

<표 1> 기존의 캐쉬의 전력 소비량을 기준으로 두었을 경우의 제안하는 구조의 전력소비량

Cache size	Cache architectures		
	DM	2-way	4-way
4KB	78.37%	51.63%	30.59%
8KB	59.85%	39.99%	25.12%
16KB	67.66%	48.10%	31.00%
32KB	61.21%	47.20%	31.88%
average	66.77%	46.73%	29.65%

이와같은 전력소비 감소 효과는 버퍼와 각각의 캐쉬로 데이터 참조가 분산되어 발생하기 때문인데, 버퍼 및 각각의 캐쉬에서 발생하는 참조 비율은 평균적으로 완전연관사상 버퍼에서 16.7%, 보조캐쉬 1에서 27.6%, 보조캐쉬 2에서 25.2% 마지막으로 메인 캐쉬에서 30.3%의 참조가 발생하였다.(그림 16) 이는 약 70%의 데이터 참조를 메인 캐쉬 이외의 작은 캐쉬 및 버퍼들로 분산시킨 것을 의미한다.



(그림 16) 제안하는 캐쉬 구조 내부의 버퍼 및 각 캐쉬가 참조되는 비율

5. 결 론

본 논문에서는 멀티미디어 응용에 적합한 분할된 데이터 캐쉬 구조와 관리방법을 제안하였다. 멀티미디어 응용의 데이터 참조 특성을 분석하여 데이터 캐쉬를 한개의 완전연관사상 버퍼와 3개의 직접사상 캐쉬로 구성하여 각각의 메모리에 할당되는 데이터 주소 영역을 할당시켰다. 그리고 데이터 참조시 하나의 작은 버퍼 또는 캐쉬만을 참조하도록 운용하였을 경우의 전력 소비 감소효과를 측정하였다. 시뮬레이션 결과 기존의 직접사상 캐쉬, 2중연관 캐쉬, 4중연관 캐쉬 구조에 비교하였을 경우 각각 33.2%, 53.3% 및 70.4%의 전력소비가 감소하였음을 나타내었다. 또한 참조 실패율

을 기준으로 하는 캐쉬 성능은 캐쉬의 크기가 16KB이상일 경우 기존의 캐쉬 구조와 비슷한 수준임을 확인할 수 있다. 제안하는 분할된 캐쉬 구조 및 주소영역 분할에 의한 관리 방법은 범용 목적의 시스템 보다는 사용 목적이 특화되어 연산 형태 및 데이터 사용 패턴이 고정된 멀티미디어 응용의 시스템 온 칩(System-on-Chip)의 각 연산 가속 엔진의 내부 메모리 구조 및 관리 방법에 적용이 적합하다. 또한 사용되는 알고리즘 및 해당 알고리즘에서 요구하는 데이터의 사용 패턴 분석에 의하여 해당 응용에 최적화된 캐쉬의 분할 개수 및 크기의 선택에 응용할 경우 기존의 캐쉬 구조에 비해 보다 효율적인 온 칩 메모리의 운용과 전력 소비 감소 효과를 가져올 수 있다.

참 고 문 헌

[1] Sohum Sohoni, Zhiyong Xu, Rui Min and Yiming Hu, "A Study of Memory System Performance of Multimedia Applications," Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.206-215, 2001.

[2] Valentin Muresan, Noel O'Connor, Noel Murphy, Sean Marlow, Stephen McGrath, "Low Power Techniques for Video Compression," Irish Signals and Systems Conference, Cork, Ireland, 2002.

[3] Jayaprakash Pisharath, Alok N. Choudhary, "An Integrated Approach to Reducing Power Dissipation in Memory Hierarchies," Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, pp.88-97, 2002.

[4] Nathaniel Pettis, Le Cai, and Yung-Hsiang Lu, "Dynamic Power Management for Streaming Data," Proceedings of the International Symposium on Low Power Electronics and Design, pp.62-65, 2004.

[5] Daniel F. Zucker, Ruby B. Lee, Michael J. Flynn, "Hardware and Software Cache Prefetching Techniques for MPEG Benchmarks," IEEE Transactions of Circuits and Systems for Video Technology, Vol.10, No.5, pp782-796, 2000.

[6] Peter Soderquist, Miriam Leaser, "Optimizing the Data Cache Performance of a Software MPEG-2 Video Decoder," Proceedings of the 5th ACM International Conference on Multimedia, pp.291-301, 1997.

[7] Hsien-Hsin S. Lee, Gary S. Tyson, "Region-Based Caching: An Energy-Delay Efficient Memory Architecture for Embedded Processors," Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, pp.120-127, 2000.

[8] MiBench Version 1.0, <http://www.eecs.umich.edu/mibench>

[9] Matthew R. Guthaus, Jeffrey S. Ringenberg, Dan Ernst, Todd M. Austin, Trevor Mudge, Richard B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," IEEE 4th Annual Workshop on Workload Characterization, 2001.

[10] MPEG Software Simulation Group,

<http://www.mpeg.org/MPEG/MSSG>

[11] SimpleScalar LLC, <http://www.simplescalar.com>

[12] Western Research Laboratory, <http://research.compaq.com/wrl/people/jouppi/CACTI.html>

[13] Premkishore Shivakumar, Norman P. Jouppi, "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," WRL Research Report, 2001.



양 훈 모

e-mail : portent@parallel.yonsei.ac.kr
 2003년 연세대학교 기계전자공학부(학사)
 2005년 연세대학교 컴퓨터과학과(공학석사)
 2005년~현재 연세대학교 컴퓨터과학과 박사과정
 관심분야 : 프로세서 및 시스템 구조, 미디어 가속엔진 구조



김 정 길

e-mail : cgkim@parallel.yonsei.ac.kr
 2003년 연세대학교 컴퓨터과학과(공학석사)
 2003년~현재 연세대학교 컴퓨터과학과 박사과정
 관심분야 : 컴퓨터 구조, 멀티미디어 내장형 시스템



박 기 호

e-mail : giho.park@samsung.com
 1993년 연세대학교 이과대학 전산학과(학사)
 1995년 연세대학교 컴퓨터과학과(공학석사)
 2000년 연세대학교 컴퓨터과학과(공학박사)
 2000년~2001년 University of Utah Post-Doctoral Research Associate
 2002년~현재 삼성전자 SOC 연구소 Processor Architecture Lab. 책임연구원
 관심분야 : 컴퓨터 구조, 내장형 시스템



김 신 덕

e-mail : sdkim@yonsei.ac.kr
 1982년 연세대학교 공과대학 전자공학과(학사)
 1987년 University of Oklahoma 전기공학(공학석사)
 1991년 Purdue University 전기공학(공학박사)
 1993년~1995년 광운대학교 컴퓨터공학과 조교수
 1995년~현재 연세대학교 공과대학 컴퓨터과학과 교수
 관심분야 : 고성능 컴퓨터 구조, 지능형 캐쉬 메모리, 병렬처리 시스템, 그리드 컴퓨팅 등