

## 필터 캐쉬의 저온도 유지를 위한 프로세서 설계 기법

최 흥 준\*, 양 나 라\*, 이 정 아\*\*, 김 중 먼\*\*\*, 김 철 흥\*\*\*\*

### Processor Design Technique for Low-Temperature Filter Cache

Hong-Jun Choi \*, Na-Ra Yang \*, Jeong-A Lee \*\*, Jong Myon Kim\*\*\*, Cheol-Hong Kim\*\*\*\*

#### 요 약

지난 수십 년 동안 프로세서의 성능은 크게 발전하여 왔다. 하지만, 공정 기술의 발달에 기인한 프로세서의 급속한 성능 향상은 최근 들어 몇 가지 문제점들에 직면하고 있다. 반도체 공정 기술이 크게 발전하면서 회로 집적도가 급속도로 높아짐에 따라서 단위 면적당 소모되는 전력량의 증가와 그에 따른 열섬 현상이 대표적인 문제점으로 인식되고 있다. 이와 같은 최근 상황에서, 최신의 프로세서를 설계할 때에는 전력 효율성 향상과 온도 제어 기술이 반드시 함께 고려되어야 한다. 본 논문에서는 프로세서에서 소비되는 전력의 상당 부분을 차지하고 있는 명령어 캐쉬의 전력 효율성을 향상시키기 위해 사용되는 대표적인 기법 중 하나인 필터 캐쉬 구조에서 발생하는 필터 캐쉬의 온도 상승 문제를 해결하기 위한 기법을 제안함으로써 저전력과 저온도 유지를 동시에 해결하고자 한다. 제안하는 변형 필터 캐쉬 구조는 세 가지로 분류된다. 프로세서가 명령어를 요청 시 필터 캐쉬와 메인 캐쉬를 선택적으로 접근하도록 하는 바이패스 필터 캐쉬 구조, 동일한 크기의 필터 캐쉬를 하나 더 추가하여 기존의 필터 캐쉬와 추가한 필터 캐쉬를 일정 시간 동안 번갈아 접근하도록 하는 중복 필터 캐쉬 구조, 그리고 기존의 필터 캐쉬를 두 개의 독립된 필터 캐쉬로 분할하여 요청 명령어에 따라 선택적으로 접근되도록 하는 분할 필터 캐쉬 구조이다. 본 논문에서는 제안된 변형 필터 캐쉬 기법들의 효율성을 정확하게 측정하기 위하여 Wattch 시뮬레이터와 Hotspot을 사용하여 모의실험을 수행한다. 모의실험결과, 본 논문에서 제안하는 세 가지 기법 중 분할 필터 캐쉬 구조가 저온도 필터 캐쉬 유지에 가장 적합한 구조임을 확인할 수 있다.

#### Abstract

Recently, processor performance has been improved dramatically. Unfortunately, as the process technology scales down, energy consumption in a processor increases significantly whereas the processor performance continues to improve. Moreover, peak temperature in the processor increases dramatically due to the increased power density, resulting in serious thermal problem. For this reason, performance, energy consumption and thermal problem should be considered together when designing up-to-date processors. This paper proposes three

• 제1저자 : 최흥준 교신저자 : 김철흥

• 투고일 : 2009. 09. 07, 심사일 : 2009. 09. 27, 게재확정일 : 2010. 01. 26.

\* 전남대학교 전자컴퓨터공학부 석사과정 \*\* 조선대학교 컴퓨터공학과 교수 \*\*\* 울산대학교 컴퓨터정보통신공학부 교수

\*\*\*\* 전남대학교 전자컴퓨터공학부 교수

※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(구 신진교수지원, KRF-2008-331-D00472)으로 연구되었음.

modified filter cache schemes to alleviate the thermal problem in the filter cache, which is one of the most energy-efficient design techniques in the hierarchical memory systems : Bypass Filter Cache (BFC), Duplicated Filter Cache (DFC) and Partitioned Filter Cache (PFC). BFC scheme enables the direct access to the L1 cache when the temperature on the filter cache exceeds the threshold, leading to reduced temperature on the filter cache. DFC scheme lowers temperature on the filter cache by appending an additional filter cache to the existing filter cache. The filter cache for PFC scheme is composed of two half-size filter caches to lower the temperature on the filter cache by reducing the access frequency. According to our simulations using Wattch and Hotspot, the proposed partitioned filter cache shows the lowest peak temperature on the filter cache, leading to higher reliability in the processor.

▶ Keyword : 임베디드 프로세서(Embedded Processor), 필터 캐쉬(Filter Cache), 변형 필터 캐쉬(Modified Filter Cache), 저전력 기법(Low-power Technique), 저온도 기법(Low-temperature Technique)

## 1. 서론

프로세서를 설계하는 데 있어서 칩의 용량은 더 이상 문제가 되지 않을 정도로 반도체 공정 기술은 지속적으로 발전되어 있다. 공정 기술의 발전에 기인하여 프로세서의 성능은 크게 향상되고 있는 반면에, 회로의 고집적화로 인하여 칩의 전력 밀도가 크게 높아지면서 칩 내부의 열섬(Hotspot) 현상이 새로운 문제점으로 부각되고 있다 [1]. 열섬 현상은 고온으로 인한 프로세서의 작동 오류 및 칩의 변형을 발생시키게 되고, 이는 칩의 신뢰성에 치명적인 영향을 미치게 된다 [2]. 열섬 현상을 해결하기 위한 방안은 크게 기계적인 냉각 방식과 구조적 설계를 통한 온도 제어 기법으로 나뉜다. 현재 널리 사용되는 기계적인 냉각 방식으로는 냉각 팬(Cooling Fan), 열 발산판(Heat Sink) 등을 들 수 있다 [3]. 하지만 기계적인 냉각 방식은 냉각 비용(Cooling Cost)의 급격한 증가로 이어져 점차 사용이 제한되고 있다 [4].

냉각 비용에 따른 문제점을 해결하기 위하여 사용되는 동적 온도 제어 기법(DTM: Dynamic Thermal Management)은 전압이나 주파수를 조절 하여 온도를 제어하는 가변 전압 주파수 조절(DVFS: Dynamic Voltage and Frequency Scaling)이나 명령어 인출을 지연시키거나 멈추는 기법을 사용해서 온도를 제어하는 인출 지연(Fetch Throttling) 기법을 통해 구현되고 있다 [5][6]. 동적 온도 제어 기법은 프로세서 실행 중에 발생하는 열섬 문제에 대해서 칩의 온도를 유동적으로 제어할 수 있기 때문에 냉각 비용을 줄이고 칩의 신뢰성을 확보할 수 있다. 하지만 동적 온도 제어 기법은 온도를 제어하기 위하여 성능을 희생시키기 때문에 성능 감소를 최소화 할 수 있는 저온도 프로세서 설계 기법에 대한 지속적인 연구가 필요하다.

저온도 유지문제가 대두되기 이전부터 성능과 함께 프로세서 설계 시 반드시 고려되던 사항으로는 전력 소모 감소를 들 수 있다. 프로세서의 성능 향상으로 인한 전력 소모의 급격한 증가는 프로세서 설계에 있어서 가장 큰 문제점으로 인식되고 있기 때문이다 [7]. 프로세서에서 소비되는 전력의 상당 부분을 차지하고 있는 명령어 캐쉬에 적용되는 대표적인 저전력 기법 중 하나로 필터 캐쉬(Filter Cache) 기법을 들 수 있다. 필터 캐쉬 기법은 크기가 작은 캐쉬 모듈을 프로세서 코어와 메인 캐쉬(Main Cache) 사이에 삽입하여 메인 캐쉬의 접근 횟수를 감소시킴으로써 전력 소모를 줄이는 기법이다 [8].

본 논문에서는 저전력을 위해 필터 캐쉬 구조를 적용한 프로세서에서 발생하는 필터 캐쉬의 온도 상승 문제를 해결하기 위한 구조적인 기법을 제안하고자 한다. 필터 캐쉬 구조에서는 크기가 작은 필터 캐쉬에 접근이 많이 일어남으로 인하여 단위 면적당 전력 소모량은 증가하게 되고 이로 인하여 높은 온도가 발생하게 된다. 필터 캐쉬에서 발생하는 높은 온도의 문제점을 해결하기 위해 세 가지의 변형 필터 캐쉬(Modified Filter Cache) 구조를 제안한다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 저전력 캐쉬 설계 및 저온도 유지 프로세서 설계에 대한 기존의 연구들에 대해서 알아본다. 3장에서는 기존의 필터 캐쉬 구조에 대한 설명과 함께 제안하는 세 가지 변형 필터 캐쉬 구조를 기술한다. 4장에서는 제안하는 기법의 효율성을 측정하기 위해 수행한 모의실험 환경과 모의실험 결과를 상세하게 비교 분석한다. 마지막으로 5장에서는 본 논문의 결론 및 향후 과제에 대하여 기술한다.

## II. 관련 연구

일반적으로 프로세서를 설계하는데 있어서 캐쉬의 크기가 작아지면 접근 실패율(Miss rate)이 증가하여 성능이 저하되므로 많은 전력 소모에도 불구하고 크기가 큰 캐쉬를 사용한다. 이와 같은 상황에서, 크기가 큰 캐쉬에서 소모되는 전력을 감소시키기 위해 다양한 기법들이 제안되었다. 프로세서의 성능 감소를 최소화하면서도 전력 소모를 감소시키기 위해 캐쉬에 대한 접근이 많지 않은 시간에는 연관 캐쉬(Associative Cache) 내의 일부 웨이(Way)들을 동적으로 비활성화시키고 캐쉬 접근이 많은 시간에는 모든 웨이들을 활성화시키는 기법을 통해 가변 크기를 갖는 캐쉬처럼 사용할 수 있는 기법이 제안되었다 [9]. 캐쉬 접근 시 요청되는 데이터가 있을 것으로 예상되는 특정 웨이만을 먼저 활성화시키고 예측이 틀린 경우에만 모든 웨이들을 활성화시켜 전력소모를 크게 줄인 기법 또한 제안되었다 [10]. 캐쉬에서의 동적 전력(Dynamic Power)을 줄이기 위해 사용되는 대표적인 기법 중 하나인 필터 캐쉬 기법은 앞서 기술된 바와 같이 한 번 접근 시 메인 캐쉬 접근에 비해 크기가 작은 필터 캐쉬에서 소모되는 전력이 적다는 사실을 이용하여 작은 크기의 필터 캐쉬를 코어(Core)와 메인 캐쉬 사이에 삽입하여 메인 캐쉬로의 접근 횟수를 줄임으로써 전력 소비를 감소시킨다 [8].

전력 소모 감소와 함께 최신 프로세서 설계 시 중요하게 고려되는 사항으로는 저온도 유지 문제를 들 수 있다. 칩 공정 기술의 발달로 인하여 성능은 크게 향상된 반면에 칩 내부의 전력 소모가 급속도로 증가함에 따라 칩 일부분의 온도가 크게 상승하는 열섬 현상이 발생하였다 [11]. 국제 반도체 기술 이정표(ITRS: International Technology Roadmap for Semiconductors)는 130nm 이하의 초미세 공정에서 칩의 최고 온도가 90°C 보다 낮아져야 한다고 발표한 바 있다 [12]. 칩의 신뢰성 향상을 위한 열섬 현상 해결 방안 중 대표적인 기법으로는 동적 온도 제어 기법(DTM: Dynamic Thermal Management)[13]을 들 수 있다. 동적 온도 제어 기법은 온도 센서를 통해 각 모듈의 온도 정보를 감시하면서 최고 온도가 일정 수준을 넘을 경우 가변 전압 주파수 조절 기법이나 인출 지연 기법을 사용하여 온도를 효과적으로 제어한다 [14]. 하지만 이 과정에서 온도를 제어하기 위하여 성능을 희생하기 때문에 프로세서의 성능 저하가 발생한다. 가변 주파수 조절 기법은 2~3%, 가변 전압 주파수 조절 기법은 6~9%, 인출 조절 기법은 8% 정도의 성능이 하락한다고 보고된 바 있다.

본 논문에서는 저전력을 위해 필터 캐쉬 구조를 사용하는 프로세서에서 발생 가능한 필터 캐쉬의 열섬 현상을 해결할

수 있는 구조적 기법을 제안하고자 한다. 필터 캐쉬의 열섬 현상을 구조적 기법을 통해 해결함으로써, 동적 온도 제어 기법으로 인한 프로세서의 성능저하 문제를 최소화하고자 한다.

## III. 제안하는 필터 캐쉬 구조

### 1. 기존의 필터 캐쉬 구조

캐쉬 메모리에서 소비되는 전력을 줄이기 위해 사용되는 대표적인 구조적 기법인 필터 캐쉬 구조는 그림 1에서 보이는 바와 같다.

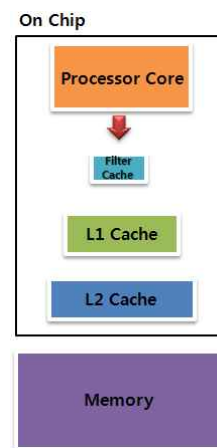


그림 1. 기존의 필터 캐쉬 구조  
Figure 1. Conventional Filter Cache Architecture

필터 캐쉬 구조는 크기가 작은 필터 캐쉬를 프로세서 코어와 메인 캐쉬 사이에 삽입함으로써 명령어 요청 시 메인 캐쉬에 접근하기에 앞서 필터 캐쉬를 먼저 접근하도록 하고, 필터 캐쉬에 요청된 명령어가 있는 경우에는 메인 캐쉬를 접근하지 않도록 함으로써 메인 캐쉬에 대한 접근 횟수를 감소시켜 프로세서 내에서 사용되는 동적 전력 소모를 줄인다. 필터 캐쉬에 요청된 명령어가 없는 경우에는 메인 캐쉬를 접근하게 되므로 캐쉬에 대한 접근 시간이 증가된다는 단점은 있지만, 저전력이 강조되는 프로세서에서는 사용될 수 있다.

기존의 필터 캐쉬 구조는 저전력을 위해서는 효율적인 기법이지만, 작은 크기의 필터 캐쉬에 접근이 빈번하게 일어나므로 인하여 단위 면적당 전력 소모가 증가하기 때문에 필터 캐쉬의 온도가 비정상적으로 상승하는 문제점이 발생할 수 있다. 기존의 필터 캐쉬 구조에서 발생하는 필터 캐쉬의 열섬 현상을 해결하기 위하여 본 논문에서는 세 가지의 변형된 필터 캐쉬 구조를 제안한다.

## 2. 변형 필터 캐쉬 구조

### 2.1 바이패스 필터 캐쉬 구조

그림 2에서 보이는 바이패스 필터 캐쉬 구조는 기존의 필터 캐쉬 구조와 유사하다. 이 구조에서는 프로세서에서 명령어를 요청할 때 일반적으로는 필터 캐쉬로 접근을 하다가 온도 센서를 통해 감지되는 필터 캐쉬의 온도가 일정 온도(72°C) 이상이 되면 필터 캐쉬에 접근하지 않고 바로 메인 캐쉬에 접근하도록 한다. 향후, 필터 캐쉬의 온도가 일정 온도 이하로 떨어지게 되면 다시 필터 캐쉬를 접근하도록 한다. 반면에, 외부 영향으로 인해 필터 캐쉬의 온도가 일정 온도 이하로 떨어지지 않는 경우에는 필터 캐쉬와 메인 캐쉬의 온도 차이가 일정 온도(5°C) 이하인 경우에 필터 캐쉬를 재접근하도록 한다.

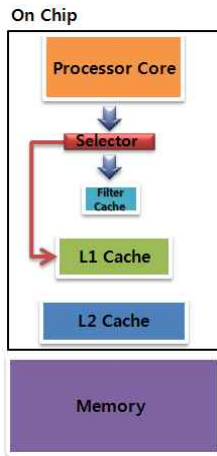


그림 2. 바이패스 필터 캐쉬 구조  
Figure 2. Bypass Filter Cache Architecture

바이패스 필터 캐쉬 구조는 온도가 상승한 필터 캐쉬의 접근을 낮은 온도의 메인 캐쉬로 이동하여 열섬 문제를 해결하고자 하는 기법이다. 이 기법은 기존의 필터 캐쉬 구조와 비교하여 단순한 선택기(Selector) 외에는 추가적인 하드웨어가 필요하지 않다. 바이패스 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 일정 온도 이상이 되는 경우 필터 캐쉬 대신에 메인 캐쉬로 접근이 이루어지는 구조이기 때문에 메인 캐쉬로의 접근이 매우 빈번하게 일어날 경우 동적 전력 소모량이 커질 수 있으므로 기존의 필터 캐쉬 구조와 비교하여 전력 소모량이 크게 증가할 위험이 있다. 그러므로 필터 캐쉬로의 접근을 막는 온도의 설정이나 메인 캐쉬에서 필터 캐쉬로 다시 접근하는 두 캐쉬 간의 온도 차이를 정하는 데 있어서 소모되는 전력량을 고려하여 설계해야 한다.

### 2.2 중복 필터 캐쉬 구조

중복 필터 캐쉬 구조는 그림 3에서 보이는 바와 같이 기존의 필터 캐쉬와 동일한 크기의 필터 캐쉬를 하나 추가하여 두 개의 필터 캐쉬 중 하나의 필터 캐쉬에만 접근하도록 하는 기법이다. 두 개의 필터 캐쉬 중에서 접근할 필터 캐쉬를 선택하는 방법은 두 필터 캐쉬의 온도에 따라 접근하는 방법(DFC-Temperature)과 일정한 횟수만큼 번갈아 가면서 접근하는 방법(DFC-Counting)을 생각할 수 있다. 온도에 따라 접근하는 방법은 초기에는 프로세서에서 명령어 요청이 오면 Filter A로 접근한 후 Filter A의 온도가 일정 온도 (70°C) 이상이 되면 Filter B를 접근하는 방식이다. 이후 두 필터 캐쉬의 온도를 비교하여 일정 온도(3°C) 이상 차이가 나는 경우 더 낮은 온도의 캐쉬로 접근을 하여 온도 상승을 분산시킨다. 접근 횟수에 따른 방법은 하나의 필터 캐쉬에 접근하는 횟수가 일정 값 (5,000,000) 이상이 되면 다른 필터 캐쉬를 접근하도록 하는 방식이다. 이 기법에서는 번갈아 가면서 일정 기간 동안 하나의 필터 캐쉬만을 접근하도록 함으로써 온도 상승을 분산시키고자 한다. 실험을 통해 성공률(Hit rate)과 온도의 하락이 적절한 반복적인 접근 횟수를 찾아내고 그 횟수에 맞춰서 두 개의 필터 캐쉬들을 선택적으로 접근하도록 한다.

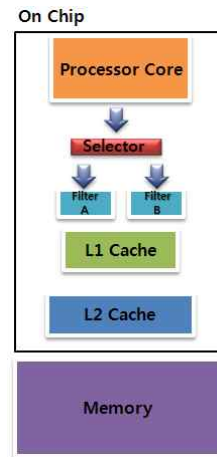


그림 3. 중복 필터 캐쉬 구조  
Fig 3. Duplicated Filter Cache Architecture

제한한 두 가지 중복 필터 캐쉬 기법 (DFC-Temperature, DFC-Counting)의 효율성을 비교하기 위해 사전 실험을 통해 필터 캐쉬의 최고 온도를 측정된 결과는 그림 4에서 보이는 바와 같다. 그래프에서 보이는 바와 같이 필터 캐쉬의 최고 온도를 비교한 결과, 접근 횟수에 따라 접근하는 방법(DFC-Co

unting)에 비하여 온도에 따라 접근하는 방법(DFC-Temperature)에서 필터 캐쉬의 최고 온도가 평균 2.8°C 감소하는 것을 확인할 수 있다. 이 결과 값을 토대로 본 논문에서는 중복 필터 캐쉬 구조의 구현 방법으로 온도에 따라 선택적으로 필터 캐쉬를 접근하는 방법을 채택한다. 여기서 사용한 평균 (avg)이란 좌측에 표현된 4가지 벤치마크 프로그램인 bzip, gcc, swim, 그리고 mgrid의 최고 온도의 평균 값으로써 전체적인 영향을 보다 쉽게 표현한다.

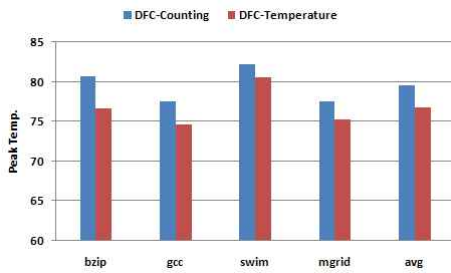


그림 4. 두 가지 중복 필터 캐쉬 구조에 따른 필터 캐쉬 최고 온도 비교

Figure 4. Peak temperature comparison for temperature method and counting method

### 2.3 분할 필터 캐쉬 구조

분할 필터 캐쉬 구조는 그림 5에서 보이는 바와 같이 중복 필터 캐쉬 구조와 차이가 없어 보이지만, 필터 캐쉬의 크기 및 접근 방식은 크게 다르다.

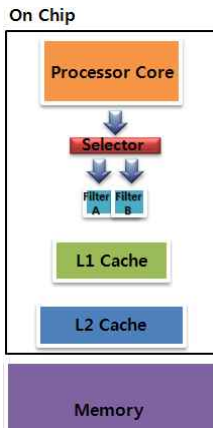


그림 5. 분할 필터 캐쉬 구조  
Figure 5. Partitioned Filter Cache Architecture

중복 필터 캐쉬 구조에서는 동일 크기의 필터 캐쉬가 하나

더 추가된 반면, 분할 필터 캐쉬 구조에서는 기존의 필터 캐쉬를 절반 크기의 두 개의 필터 캐쉬로 분할한다. 분할 필터 캐쉬 구조에서는 요청되는 명령어의 주소값에 따라 두 개의 캐쉬 중 하나만을 활성화시킨다. 일반적인 프로세서에서 명령어 워드 길이가 32비트이기 때문에 PC 값이 4씩 증가하게 되므로, 요청되는 명령어의 주소값을 읽어서 8로 Modular 연산을 수행한 후 결과값이 0이면 Filter A에 접근하고 그렇지 않은 경우 Filter B에 접근하도록 구현한다. 명령어 요청으로 인한 필터 캐쉬 접근을 캐쉬를 두 개로 분할하여 접근함으로써 실제 활성화 되는 캐쉬의 크기가 절반으로 줄어들게 한다. 이로 인해 소모되는 전력량이 감소될 것으로 보이며 결과적으로 전력밀도 또한 낮아지게 되어 특정 부분의 온도가 비정상적으로 상승하는 현상을 예방할 수 있을 것으로 기대된다. 반면에 위의 방법을 사용하는 경우 캐쉬의 스위칭에 드는 비용이나 전력 소모가 증가될 것으로 보이나 명령어 주소를 이용한 간단한 연산을 하는 모듈 하나만을 추가하면 되므로 이는 전체 프로세서에서 소모되는 비용이나 전력에 비교해 볼 때 무시될 수 있을 정도로 적을 것으로 기대된다.

## IV. 모의실험

본 논문에서는 다양한 필터 캐쉬 구조에 따른 전력소모와 온도변화를 측정하기 위하여 Watch[15] 시뮬레이터 Hotspot [16]을 사용하였다. Watch 시뮬레이터의 입력으로는 다양한 프로그램의 특성에 따라 구현된 SPEC2000 Benchmarks [17] 프로그램들을 사용하였다. 모의 실험 시 사용한 프로세서 구성 변수들은 표 1에서 보이는 바와 같다.

표 1. 프로세서 구성 변수  
Table 1. System Parameters

실험인자	값
Functional Units	4 integer ALUs, 4 FP ALUs, 1 integer multiplier/divider, 1 FP multiplier/divider
Conventional Filter Cache	1KB, fully-associative, 32bytes lines, 1 cycle latency
L1 I-Cache	32KB, 4-way, 32bytes lines, 1 cycle latency
L1 D-Cache	32KB, 4-way, 32bytes lines, 1 cycle latency
L2 Cache	512KB, 8-way, 32bytes lines, 12 Cycle latency

실험에서는 최신 프로세서에서 발생하는 온도 문제를 해결

하기 위해 사용되는 동적 온도 제어 기법을 적용하였다. 적용된 동적 온도 제어 기법은 2단계로 동작한다. 1단계는 구동단계(Trigger threshold)로 명령어의 인출 지연 방법으로 온도를 조절한다. 2단계는 긴급단계(Emergency threshold)로 가변전압 및 주파수 조절 방법으로 온도를 제어한다. 모의실험 시 구동단계 온도는 80℃로 설정하였고, 긴급단계온도는 85℃로 설정하였다. 유닛의 기본온도는 60℃로 설정하고, 실행되는 명령어 개수는 10억개로 설정하였다. 전체 프로세서의 온도 분포는 Origin Prof[18] 툴을 사용하여 표현한다.

1. 성능

그림 6,7은 각각 정수형과 실수형 프로그램을 수행할 때의 성능을 비교 분석한 결과를 나타내고 있다. 성능은 가장 일반적으로 사용되는 사이클당 수행 명령어 수를 나타내는 IPC (Instruction Per Cycle)를 사용하였다. 그림에서 사용한 평균(avg)이란 좌측에 표현된 4가지 벤치마크 프로그램의 평균 IPC를 나타내는 것으로 전체적인 영향을 보다 쉽게 표현한다. 그림 6에서의 평균은 bzip2, gcc, gzip, 그리고 mcf의 평균을 의미하며, 그림 7에서의 평균은 swim, mgrid, applu, 그리고 lucas의 평균을 의미한다. 그래프에서 CFC는 기존의 필터 캐쉬 구조를 의미하고, BFC는 바이패스 필터 캐쉬 구조를 의미하며, DFC는 중복 필터 캐쉬 구조를, 마지막으로 PFC는 분할 필터 캐쉬 구조를 나타낸다.

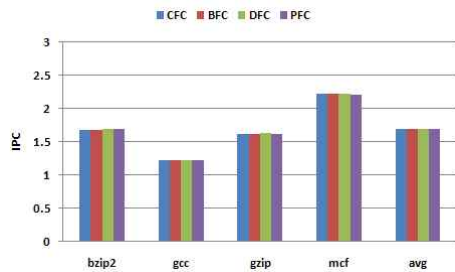


그림 6. 정수형 프로그램에서의 IPC  
Figure 6. IPC (CINT)

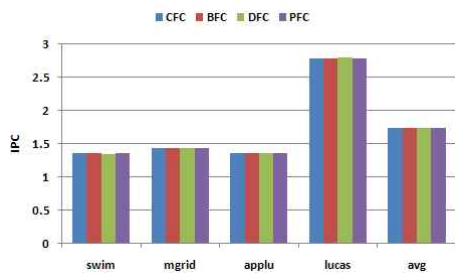


그림 7. 실수형 프로그램에서의 IPC  
Figure 7. IPC (CFP)

각 필터 캐쉬 구조에 성능을 비교, 분석한 결과는 그래프에서 보이는 바와 같이 별다른 차이를 보이지 않음을 알 수 있다. 성능 차이가 최대 0.5% 미만이므로, 각 필터 캐쉬 구조에 따라 성능 차이는 거의 발생하지 않음을 확인할 수 있다. 이는 변형 필터 캐쉬가 캐쉬의 적중률에 미치는 영향이 미미하다는 사실을 반영한다.

2. 프로세서 최고 온도

그림8-11은 gcc 프로그램을 수행할 때의 각 필터 캐쉬 구조에 따른 프로세서 전체의 최고 온도를 나타낸다. 프로세서의 최고 온도는 Origin Pro를 이용하여 아래와 같이 나타내었으며, 가로와 세로축의 값은 칩의 크기를 나타내는 것으로써 단위는 미터(meter)이고, 우측의 범례는 온도에 따른 색깔을 나타내고 있다.

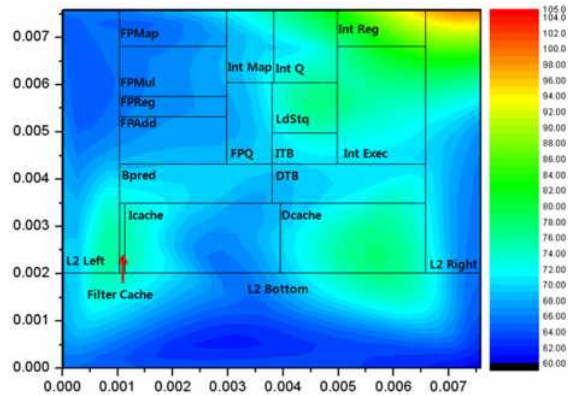


그림 8. gcc 수행 시 CFC 구조의 프로세서 온도  
Figure 8. Processor temperature of CFC (gcc)

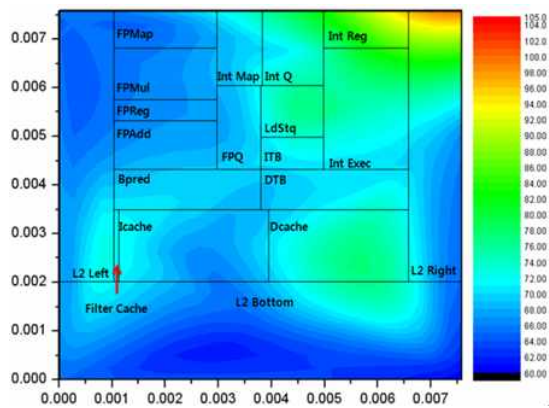


그림 9. gcc 수행 시 BFC 구조의 프로세서 온도  
Figure 9. Processor temperature of BFC (gcc)

프로세서 내에서 필터 캐쉬 부분은 화살표로 표시되어 있다. 필터 캐쉬의 온도를 분석한 결과 기존의 필터 캐쉬 구조와 비교하여 바이패스 필터 캐쉬 구조에서는 3℃, 중복 필터 캐쉬 구조에서는 0.5℃, 분할 필터 캐쉬 구조에서는 6℃의 온도가 각각 감소하는 것을 확인할 수 있다. 그림 8에서는 필터 캐쉬 부분이 녹색으로 표시되어 온도가 상승되어 있는데 반해, 바이패스 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 감소되어 청색으로 표시되어 있다 (그림 9).

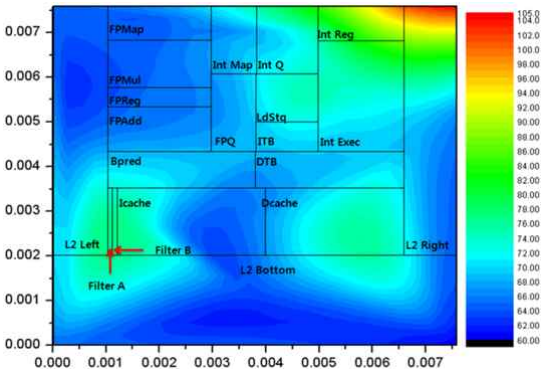


그림 10. gcc 수행 시 DFC 구조의 프로세서 온도  
Figure 10. Processor temperature of DFC (gcc)

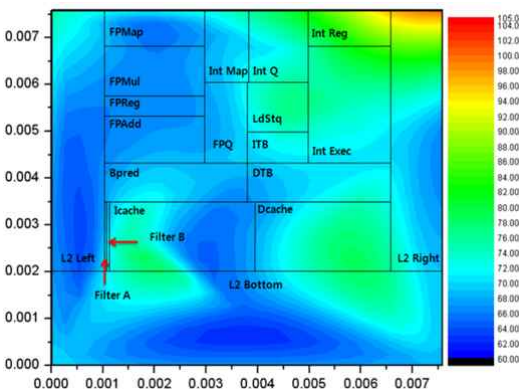


그림 11. gcc 수행 시 PFC 구조의 프로세서 온도  
Figure 11. Processor temperature of PFC (gcc)

중복 필터 캐쉬 구조에서의 필터 캐쉬는 그림 10에서 보이는 바와 같이 기존 필터 캐쉬 구조에서의 필터 캐쉬와 마찬가지로 녹색으로 표시되어 있다. 중복 필터 캐쉬의 구조의 경우에는 같은 크기의 모듈이 하나 더 추가됨으로써 온도가 감소될 것으로 예측되었지만, 실험 결과에서는 기존의 구조와 비교하여 온도 감소가 거의 없음을 알 수 있다. 이는 중복된 두

개의 필터 캐쉬(Filter A, Filter B)가 서로 인접하여 열을 다른 부분으로 발산시키지 못하고 오히려 서로에게 열을 전달하기 때문에 분석된다.

분할 필터 캐쉬 구조에서는 그림 11에서 보이는 바와 같이 질은 청색으로 표시되어 있어 제안한 세 가지 구조 중 온도 감소폭이 가장 큰 것을 확인할 수 있다. 이는 캐쉬 분할을 통해 활성화 되는 캐쉬 크기가 줄어들어 소모되는 전력량이 감소하고 접근이 분산되어 전력밀도 또한 낮아지게 됨으로써 온도 또한 낮아진 것으로 분석된다. gzip 프로그램을 실행할 때의 각 필터 캐쉬 구조에 따른 프로세서의 최고 온도를 비교 분석한 결과는 그림 12-15에서 보이는 바와 같다.

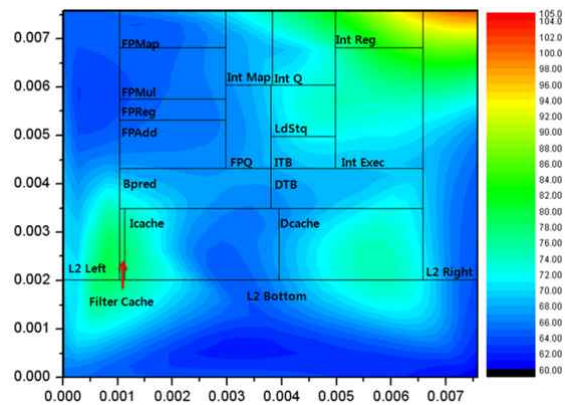


그림 12. gzip 수행 시 CFC 구조의 프로세서 온도  
Figure 12. Processor temperature of CFC (gzip)

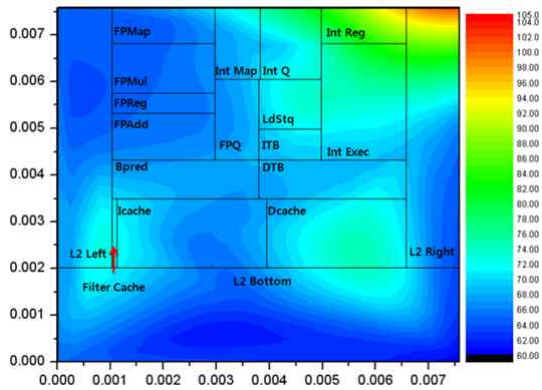


그림 13. gzip 수행 시 BFC 구조의 프로세서 온도  
Figure 13. Processor temperature of BFC (gzip)

기존의 필터 캐쉬 구조와 비교하여 바이패스 필터 캐쉬 구조에서는 5℃의 온도 감소 효과가 나타나고, 중복 필터 캐쉬 구조에서는 오히려 0.01℃의 온도 상승 현상이 나타나며, 분할

필터 캐쉬 구조에서는 최대인 8°C의 온도가 감소하는 것을 확인할 수 있다.

그림 12에서 나타내는 바와 같이 기존의 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 상승하여 녹색으로 표시되는 것을 확인할 수 있다. 제안하는 변형 필터 캐쉬 구조들을 살펴보면 바이패스 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 감소하여 청색으로 표시되고 있다 (그림 13).

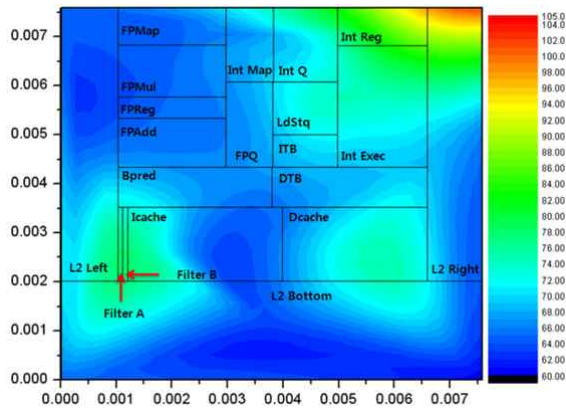


그림 14. gzip 수행 시 DFC 구조의 프로세서 온도  
Figure 14. Processor temperature of DFC (gzip)

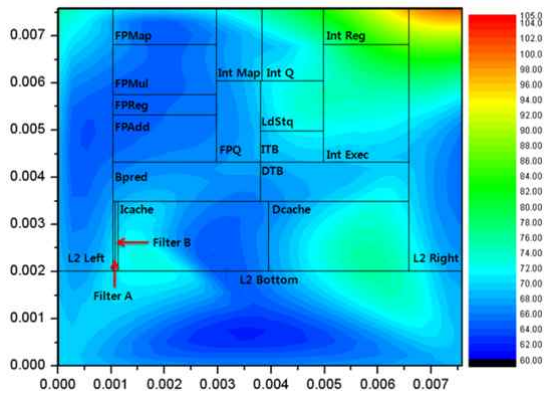


그림 15. gzip 수행 시 PFC 구조의 프로세서 온도  
Figure 15. Processor temperature of PFC (gzip)

중복 필터 캐쉬 구조에서는 기존의 필터 캐쉬 구조와 비교하여 오히려 필터 캐쉬의 온도가 약간 상승하는 현상을 확인할 수 있다 (그림 14). gcc 수행 시와 마찬가지로 필터 캐쉬에서 발생된 열을 제대로 분산시키지 못하기 때문인 것으로 분석된다. gcc 프로그램 수행 결과와 마찬가지로 gzip 프로그램을 수행한 결과에서도 분할 필터 캐쉬 구조는 타 구조들과 비

교하여 필터 캐쉬의 온도를 크게 감소시키고 있음을 확인할 수 있다 (그림 15).

실험 결과를 토대로 판단할 때, 정수형 프로그램을 실행할 때 분할 필터 캐쉬 구조가 필터 캐쉬로의 접근을 효율적으로 분산시킴으로써 온도 감소 효과가 가장 크게 나타나는 최적의 구조임을 알 수 있다.

실수형 프로그램인 swim을 수행할 때 필터 캐쉬 구조에 따른 프로세서의 최고 온도를 분석한 결과는 그림 16-19에서 보이는 바와 같다. swim 프로그램이 수행될 때, 기존의 필터 캐쉬 구조와 비교하면, 바이패스 필터 캐쉬 구조에서는 6°C의 온도가 감소하고, 중복 필터 캐쉬 구조에서는 0.01°C의 온도가 상승하며, 분할 필터 캐쉬 구조에서는 8°C의 온도가 감소하는 것을 확인할 수 있다.

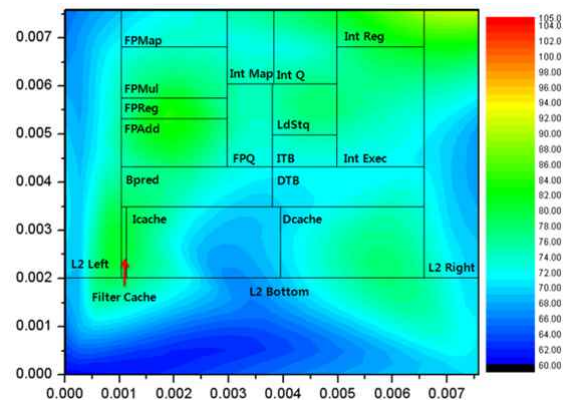


그림 16. swim 수행 시 CFC 구조의 프로세서 온도  
Figure 16. Processor temperature of CFC (swim)

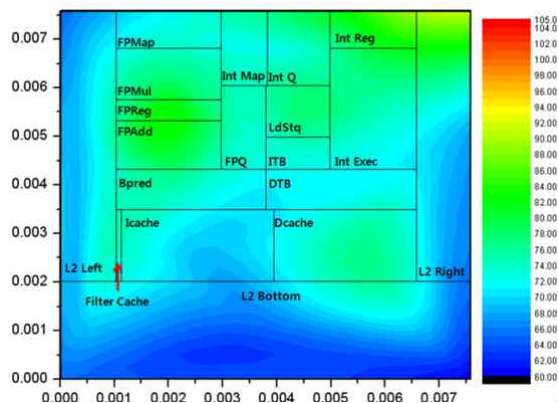


그림 17. swim 수행 시 BFC 구조의 프로세서 온도  
Figure 17. Processor temperature of BFC (swim)



그림 16에서 나타내는 바와 같이 기존의 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 상승하여 녹색으로 표시되는 것을 확인할 수 있다. 또한, 정수형을 수행한 필터 캐쉬 구조에 비해 프로세서 전체적인 온도가 높음을 확인할 수 있다. 제안하는 변형 필터 캐쉬 구조들을 살펴보면 바이패스 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 감소하여 옅은 청색으로 표시되고 있다 (그림 17). 바이패스 필터 캐쉬 구조 또한 정수형을 수행한 경우에 비해 전체적으로 온도가 낮음을 확인할 수 있다.

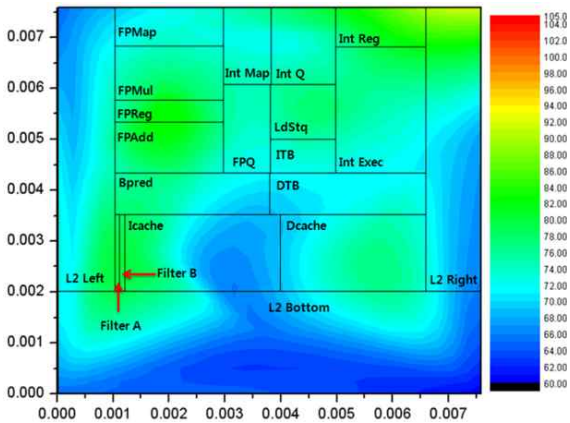


그림 18. swim 수행 시 DFC 구조의 프로세서 온도  
Figure 18. Processor temperature of DFC (swim)

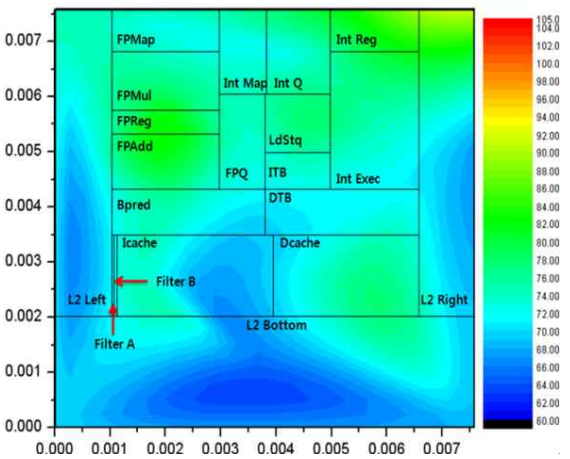


그림 19. swim 수행 시 PFC 구조의 프로세서 온도  
Figure 19. Processor temperature of PFC (swim)

중복 필터 캐쉬 구조에서는 정수형을 수행한 경우와 마찬가지로 필터 캐쉬의 온도가 약간 상승하는 현상을 확인할 수 있다 (그림 18). 실수형 프로그램을 수행하는 경우에도 분할

필터 캐쉬 구조에서 필터 캐쉬의 온도가 가장 크게 감소하고 있음을 확인할 수 있다 (그림 19). 전체적인 프로세서 온도는 정수형 프로그램을 수행한 경우에 비해 실수형 프로그램을 수행하는 경우가 더 높으나, 제안 기법을 통한 필터 캐쉬의 온도 감소 효과는 정수형과 실수형에서 유사함을 확인할 수 있다.

정수형 프로그램과 실수형 프로그램을 수행할 때의 프로세서의 전체적인 온도 분포를 비교하면 정수형 프로그램을 수행하는 경우에는 Integer Queue, Integer Register 등의 온도가 상승하는 것을 확인할 수 있고, 실수형 프로그램을 수행하는 경우에는 Floating-point Register, Floating-point Adder 등의 온도가 상승하는 것을 확인할 수 있다. 이는 프로그램의 종류에 따라 이와 관련된 모듈들의 활용 빈도가 각기 다르기 때문이다.

실수형 프로그램인 mgrid를 수행할 때의 필터 캐쉬 구조에 따른 프로세서의 최고 온도는 그림 20-23에서 보이는 바와 같다.

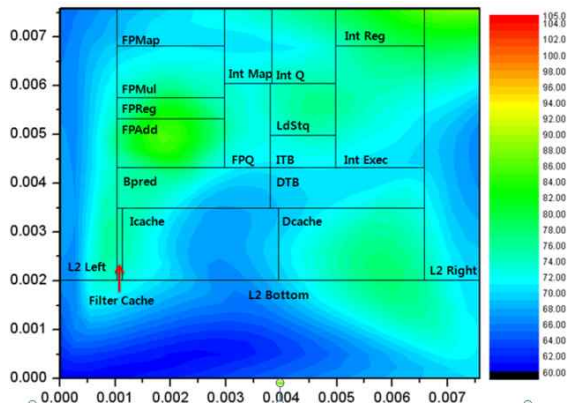


그림 20. mgrid 수행 시 CFC 구조의 프로세서 온도  
Figure 20. Processor temperature of CFC (mgrid)

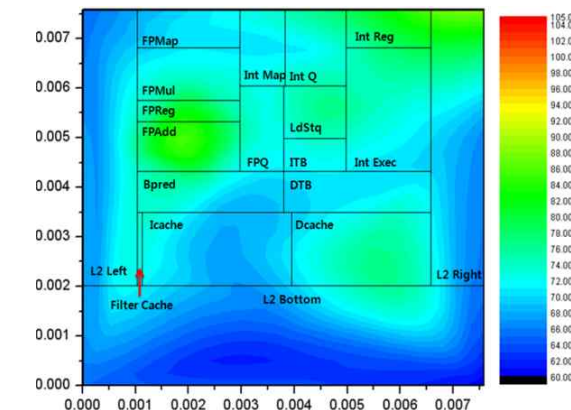


그림 21. mgrid 수행 시 BFC 구조의 프로세서 온도  
Figure 21. Processor temperature of BFC (mgrid)

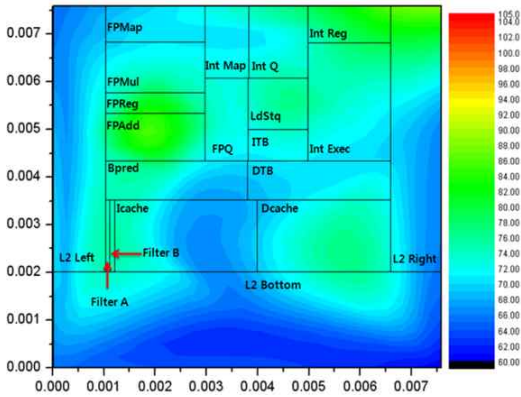


그림 22. mgrid 수행 시 DFC 구조의 프로세서 온도  
Figure 22. Processor temperature of DFC (mgrid)

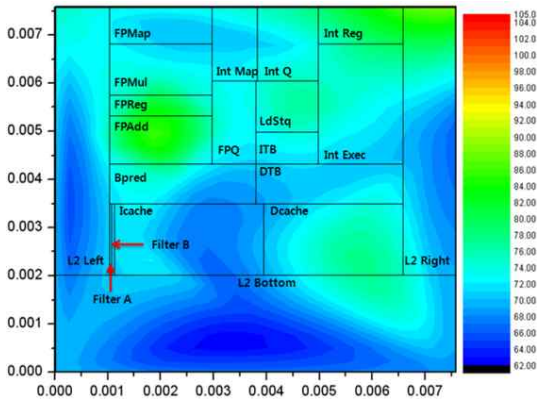


그림 23. mgrid 수행 시 PFC 구조의 프로세서 온도  
Figure 23. Processor temperature of PFC (mgrid)

기존의 필터 캐쉬 구조와 비교하여 바이패스 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 3°C 감소하고, 중복 필터 캐쉬 구조에서는 0.5°C의 온도가 상승함을 확인할 수 있다. 분할 필터 캐쉬 구조에서는 필터 캐쉬의 온도가 6°C 감소하여 다른 프로그램들을 수행할 때와 마찬가지로 필터 캐쉬의 온도 감소에 가장 효율적인 구조임을 확인할 수 있다. 실수형 프로그램인 swim과 mgrid를 비교하면 swim 프로그램을 수행할 때가 mgrid를 수행할 때와 비교하여 온도가 더 많이 감소하는 결과를 확인할 수 있다.

## V. 결론

본 논문에서는 저전력을 위해 사용되는 필터 캐쉬 구조에서 발생하는 필터 캐쉬의 온도 상승 문제를 해결하기 위하여 변형 필터 캐쉬 구조를 제안하였다. 제안한 변형 필터 캐쉬 구조는 바이패스 필터 캐쉬 구조, 중복 필터 캐쉬 구조, 분할 필터 캐쉬 구조의 3가지이다. 모의실험 결과, 필터 캐쉬의 온도에 따라 필터 캐쉬와 메인 캐쉬를 선택적으로 접근하는 바이패스 필터 캐쉬 구조에서는 기존의 필터 캐쉬 구조와 비교하여 필터 캐쉬의 온도가 최대 6°C 감소되는 것을 확인하였다. 같은 크기의 필터 캐쉬를 하나 추가하여 일정 기간 동안에는 하나의 필터 캐쉬만이 사용되도록 구성한 중복 필터 캐쉬 구조에서는 기존의 필터 캐쉬 구조와 비교하여 필터 캐쉬의 온도가 거의 차이가 없었으며, 일부 프로그램에서는 오히려 온도가 상승하는 결과를 얻었다. 예상과 달리 중복 필터 캐쉬 구조에서 필터 캐쉬의 온도가 오히려 상승하는 이유로는 같은 크기의 모듈이 서로 붙어 있어 상승하는 열을 서로 분산시키지 못하기 때문인 것으로 판단된다. 하나의 필터 캐쉬를 절반 크기의 두 개의 캐쉬로 분할하고 요청된 명령어에 따라 선택적으로 접근하도록 하는 분할 필터 캐쉬 구조에서는 필터 캐쉬의 온도를 제안 구조 중 최대인 8°C 감소시킴을 확인하였다. 실험 결과를 기반으로 판단할 때, 필터 캐쉬를 사용하는 프로세서의 저온도 유지에 가장 적합한 구조는 분할 필터 캐쉬 구조임을 알 수 있다.

향후에는 본 논문에서 제안한 변형 필터 캐쉬 기법들은 기반으로 프로세서의 저전력/저온도 유지를 위한 새로운 구조 연구 및 공간 배치(Layout)를 이용한 온도 감소 기법에 관한 연구를 수행하고자 한다.

## 참고문헌

- [1] 이병석, 김철홍, 이정아, “온도인지 마이크로 프로세서에서 동적 연산 이관을 위한 유닛 선택 기법에 관한 비교분석,” 한국컴퓨터종합학술대회 논문집 제 36권, 제 1호(A), 328-329 쪽, 2009년 6월.
- [2] L. Yeh and R. Chy, “Thermal Management of Microelectronic Equipment,” American Society of Mechanical Engineering, 2001.
- [3] Z. Zhijun, L. R. Hoover, and A. L. Phillips, “Advanced thermal architecture for cooling of high power electronics,”

Components and Packaging Technologies, IEEE Transactions on, vol. 25, pp. 629-634, 2002.

[4] R. Mahajan, "Thermal Management of CPUs: A Perspective on Trends, Needs, and Opportunities," In Proceedings of the 8th International Workshop on THERMAL INvestigations of ICs and Systems, 2002.

[5] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," Journal of Instruction-Level Parallelism, Vol. 8, pp. 1-16, 2005.

[6] 최진항, 이종성, 공준호, 정성우, "실시간 온도 감시를 위한 시뮬레이션 도구의 표현," 한국컴퓨터정보학회 논문지 제 14권 제 1호, 145-151쪽, 2009년 1월.

[7] 김중면, 정재욱, 김철홍, "분할 기법을 이용한 저전력 명령어 캐쉬 설계," 정보과학회논문지 : 컴퓨팅의 실제 및 레터 제 13권 제 5호, 241-251쪽, 2007년 10월.

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

[9] David H. Albonese, "Selective Cache Ways: On Demand Cache Resource Allocation," In Proceedings of International Symposium on Microarchitecture, pp. 70-75, 1999.

[10] M. Powell, A. Agarwal, T. N. Vijaykumar, B. Falsafi, and K. Roy, "Reducing Set-Associative Cache Energy via Way-Prediction and Selective Direct-Mapping," In Proceedings of International Symposium on Microarchitecture, pp. 54-65, 2001.

[11] 공준호, 최진항, 이종성, 정성우, "인텔 펜티엄 4와 코어2 듀오의 실행시간과 파워소모량 효율성 비교," 한국컴퓨터정보학회논문지. 제 13권, 제 7호, 165-172쪽, 2008년 12월.

[12] SIA, "The International Technology Roadmap for Semiconductors," 2005.

[13] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," In Proceedings of the 7th International Symposium on High-Performance Computer Architecture, 2001.

[14] 최진항, 공준호, 정의영, 정성우, "온도 인지 마이크로프로세서를 위한 듀얼 레지스터 파일구조," 정보과학회논문지 : 시스템 및 이론 제 35권, 제 11·12호, 510-551쪽, 2008년 12월.

[15] Wattch, <http://www.eecs.harvard.edu/~dbrooks/>

[16] Hotspot, <http://lava.cs.virginia.edu/HotSpot/>

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

[18] Origin Pro, <http://www.originlab.com/>

저 자 소개



**최 홍 준**  
 2009년 : 전남대학교 전자컴퓨터공학과 공학사  
 2009년 : 전남대학교 전자컴퓨터공학부 석사과정 입학  
 관심분야 : 저전력 설계, 컴퓨터 구조



**양 나 라**  
 2008년 : 호남대학교 컴퓨터공학과(학사)  
 2008년 : 전남대학교 전자컴퓨터공학부석사과정 입학  
 관심분야 : 컴퓨터구조, 임베디드 시스템



**이 정 아**  
 1982년 : 서울대학교 컴퓨터공학과 공학사  
 1985년 : 인디애나 주립대학교 컴퓨터학과 공학석사  
 1990년 : 캘리포니아 주립대학교(UCLA) 컴퓨터공학과 공학박사  
 1995년-현재 : 조선대학교 컴퓨터공학과 교수  
 관심분야 : 고속 디지털 연산기, 특수 용도의 VLSI 구조, 컴퓨터 구조



**김 중 면**  
 1995년 : 명지대학교 전기공학사  
 2000년 : University of Florida ECE 석사  
 2005년 : Georgia Institute of Technology ECE 박사  
 2005년 - 2007년 : 삼성종합기술원 전임연구원  
 2007년 - 현재 : 울산대학교 컴퓨터정보통신공학부교수  
 관심분야 : 임베디드 SoC, 컴퓨터구조, 프로세서 설계, 병렬처리



**김 철 흥**

1998년 : 서울대학교 컴퓨터공학사

2000년 : 서울대학교 대학원 컴퓨터 공학부 석사

2006년 : 서울대학교 대학원 전기컴퓨터공학부 박사

2005년 - 2007년 :

삼성전자 반도체총괄 SYS.LSI사업부 책임연구원

2007년 - 현재 :

전남대학교 전자컴퓨터공학부 교수

관심분야 : 임베디드시스템, 컴퓨터구조, SoC 설계, 저전력 설계