

코어 내부 구성요소와 L2 캐쉬의 배치 관계에 따른 멀티코어 프로세서의 온도 분석

손 동 오*, 김 종 면**, 김 철 흥*

Analysis on the Temperature of Multi-core Processors according to Placement of Functional Units and L2 Cache

Dong-Oh Son*, Jong-Myon Kim**, Cheol-Hong Kim*

요 약

멀티코어 프로세서는 여러 개의 코어가 하나의 칩에 배치됨에 따라 전력 밀도가 상승하여 높은 발열이 발생하는 등. 이러한 발열 문제를 해결하기 위해서 최근까지 다양한 연구가 진행되고 있다. 마이크로프로세서의 온도 감소를 위한 기법으로는 기계적 냉각 기법, 동적 온도 관리 기법 등이 있지만 이러한 기법들은 추가적인 냉각 비용이 발생하거나 성능의 저하가 발생한다. 플로어플랜 기법은 추가적인 냉각 비용이 발생하지 않으며, 성능 저하가 거의 발생하지 않는다는 장점을 지닌다. 본 논문에서는 멀티코어 프로세서의 특정 구성요소의 발열 문제를 해결하기 위해 코어 내부 구성요소와 L2 캐쉬의 다양한 플로어플랜을 활용하고자 한다. 실험 결과, 코어의 뜨거운 구성요소를 L2 캐쉬와 인접하게 배치할 경우 칩의 온도 감소에 매우 효과적임을 알 수 있다. 코어를 캐쉬 상단-가운데 배치하는 기본 플로어플랜과 비교하여, 코어를 중앙에 배치하고 뜨거운 구성요소를 L2 캐쉬와 인접하게 배치하는 플로어플랜의 경우에는 8.04°C, 코어를 외곽에 배치하고 뜨거운 구성요소를 L2 캐쉬와 인접하게 배치하는 플로어플랜의 경우에는 8.05°C의 최고온도 감소 효과를 보임을 알 수 있다.

▶ Keywords : 멀티코어 프로세서, 온도, 플로어플랜, 열심 현상

Abstract

As cores in multi-core processors are integrated in a single chip, power density increased considerably, resulting in high temperature. For this reason, many research groups have focused

•제1저자 : 손동오 •교신저자 : 김철흥

•투고일 : 2013. 11. 20, 심사일 : 2013. 12. 18, 게재확정일 : 2014. 2. 25.

* 전남대학교 전자컴퓨터공학부(School of Electronics and Computer Engineering, Chonnam National University)

** 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

※ 본 연구는 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단 기초연구사업의 지원(2012R1A1B4003492)과 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업의 연구결과로 수행되었음(NIPA-2014-H0301-14-1014)

on the techniques to solve thermal problems. In general, the approaches using mechanical cooling system or DTM(Dynamic Thermal Management) have been used to reduce the temperature in the microprocessors. However, existing approaches cannot solve thermal problems due to high cost and performance degradation. However, floorplan scheme does not require extra cooling cost and performance degradation. In this paper, we propose the diverse floorplan schemes in order to alleviate the thermal problem caused by the hottest unit in multi-core processors. Simulation results show that the peak temperature can be reduced efficiently when the hottest unit is located near to L2 cache. Compared to baseline floorplan, the peak temperature of core-central and core-edge are decreased by 8.04°C , 8.05°C on average, respectively.

▶ Keywords : Multi-core processor, Temperature, Floorplan, Hotspot

I. 서 론

마이크로프로세서는 공정 기술의 발달에 기인하여 동작 주파수의 증가를 이루었다. 동작 주파수의 증가는 마이크로프로세서의 성능을 향상시켰지만, 전력 소모량이 증가하였고 이에 따라 높은 발열 등의 문제점이 발생하였다. 이처럼 싱글코어 프로세서의 동작 주파수의 증가는 더 이상 마이크로프로세서의 성능향상에 이점이 되지 못하였다. 따라서 마이크로프로세서 설계자들은 하나의 코어만을 사용하는 기존의 싱글코어 프로세서에서 2개 이상의 다수의 코어를 사용하는 멀티코어 프로세서를 제안하였다. 멀티코어 프로세서는 다수의 코어를 활용하기 때문에 기존의 싱글코어 프로세서보다 낮은 주파수라도 높은 성능을 보이며 전력밀도의 분산으로 인해 칩의 발열 문제를 해결할 수 있다. 이처럼 멀티코어 프로세서의 개발은 마이크로프로세서의 성능 향상과 온도 문제를 해결하였지만, 동작 주파수의 지속적인 증가와 하나의 칩에 여러 개의 코어가 배치됨으로써 전력밀도는 크게 증가하였다. 전력 밀도의 증가는 마이크로프로세서의 온도 문제로 이어진다. 전력 밀도의 증가에 따라 온도가 증가하게 되면 마이크로프로세서의 작동 요리를 유발시키고 칩의 신뢰성과 수명에 부정적인 영향을 미친다(1). 기존연구(2)에 따르면 마이크로프로세서의 온도가 $10\sim 15^{\circ}\text{C}$ 상승하게 되면 칩의 수명이 절반으로 감소한다. 이와 같은 문제를 방지하기 위해서는 냉각시스템 구축이 필요하지만, 이것은 냉각 비용의 기하급수적인 증가를 초래한다. 이처럼 시스템의 성능과 수명, 그리고 신뢰성을 향상시키기

위해서는 온도 문제 해결이 필수적으로 이뤄져야 한다.

이러한 온도 문제에 대한 해결책으로는 기계적 냉각 기법(3)과 동적 온도 관리 기법(DTM: Dynamic Thermal Management)(4) 등이 있다. 기계적 냉각 기법은 방열판과 냉각팬을 사용하여 마이크로프로세서의 온도를 낮추는 기법이다. 기계적 냉각 기법은 마이크로프로세서의 성능에 영향을 미치지 않고 효과적으로 프로세서의 온도를 감소시키지만, 냉각을 위해 추가적인 비용이 소모된다. 이러한 기계적 냉각 기법 이외에도 냉각 비용이 발생하지 않고 시스템적으로 온도를 관리하는 방법으로는 동적 온도 관리 기법이 있다. 동적 온도 관리 기법은 동적 주파수 조절(DFS: Dynamic Frequency Scaling), 동적 전압 조절(DVS: Dynamic Voltage Scaling), 클럭 게이팅(Clock Gating) 등을 이용하여 프로세서 내부의 온도를 제어한다. 동적 온도 관리 기법을 사용하면 최악의 상황을 방지하여 시스템의 평균적인 온도 값을 유지하며 냉각 비용을 절감할 수 있다. 이러한 동적 온도 관리 기법은 마이크로프로세서의 전력과 주파수를 조절하거나 멈추는 방법으로 동작한다. 하지만 이러한 방법은 성능에 저하가 발생하기 때문에 쉽게 사용할 수 없다.

프로세서의 온도 감소와 함께 성능의 저하를 막기 위해 연구자들은 온도 인지 기반의 플로어플랜 기법을 활용하였다(5-10). 온도 인지 기반의 플로어플랜 기법은 프로세서를 구성하는 구성요소들을 배치할 때 각 구성요소들의 온도를 고려하여 온도가 높은 구성요소들이 밀집되지 않도록 배치하여 프로세서의 온도를 분산시키는 기법이다. 또한, 온도가 높은 구성요소와 온도가 낮은 구성요소를 인접하게 배치하여 온도의 상해 효과를 통한 온도를 감소시킬 수 있다. 기존연구(11)의

결과에 따르면 프로세서에서 높은 온도를 보이는 코어 사이에 차가운 구성요소인 L2 캐쉬를 배치하면 코어의 열을 흡수하는 냉각장치로 이용할 수 있으며 프로세서의 온도를 크게 감소시킬 수 있다. 이처럼 플로어플랜 기법은 구성요소의 배치를 활용하여 성능의 저하 없이 프로세서의 온도를 감소시키는 효과가 있다. 이러한 이유로 본 논문에서는 기계적 냉각 기법과 동적 온도 관리 기법이 아닌 프로세서의 성능에 적은 영향을 미치면서 최대 온도를 감소시키는 플로어플랜 기법에 초점을 맞췄다. 본 논문에서는 플로어플랜 기법을 활용하여 작업 처리에 영향이 없는 마이크로프로세서의 온도 감소 기법을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 섹션 2에서는 제안하는 플로어플랜에 대해서 설명한다. 섹션 3에서는 제안한 플로어플랜 기법의 세부적인 평가 결과를 보여준다. 마지막으로 섹션 4에서 결론짓는다.

II. 분석 대상 플로어플랜 방법

본 논문에서는 멀티코어 프로세서에서 플로어플랜의 구성에 따른 구성요소의 온도변화를 자세히 분석하기 위해서 4개의 레이어를 구성하여 실험하였다. 실험에서 사용된 코어의 구성요소들은 플로어플랜의 변화 없이 코어 단위로 구성하여 하나의 블록으로 구분하여 플로어플랜을 구성하였다. 기존의 연구[12]에서는 정수형 레지스터(그림 1에서 IntReg)의 위치를 상, 하, 좌, 우로 대칭 시키지 않고 단순히 코어의 위치만을 변경하여 실험하였다. IntReg의 위치를 변경하지 않을 경우 IntReg가 다른 코어에 인접하게 배치되기 때문에 온도감소에 효과가 떨어진다. 하지만, 본 논문에서는, 플로어플랜의 대상이 되는 IntReg의 위치변경을 위해서 구성요소의 플로어플랜을 그대로 유지하고 상, 하, 좌, 우로 대칭 시켜 IntReg가 다른 코어에 인접하게 배치되지 않도록 플로어플랜 하였다. 또한, 이전 실험[13]에서는 코어 단위로 구성된 블록의 평균 온도를 측정 한 뒤 플로어플랜을 변경하였지만, 본 논문에서는 각 구성요소의 세부온도를 측정하여 보다 상세하게 온도변화를 분석하고자 한다. 실험에서 사용된 플로어플랜은 다음과 같다.

1. 기본 구조

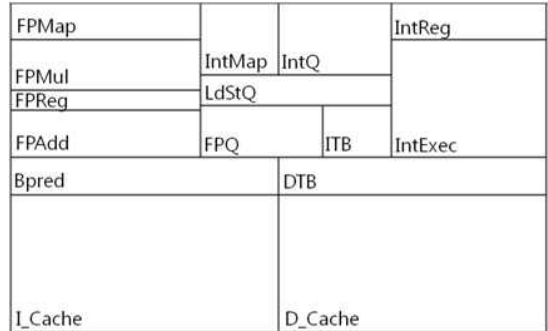


그림 1. Alpha 21264(EV6) 프로세서의 플로어플랜
Fig. 1. Floorplan of Alpha 21264(EV6) processor

그림 1은 실험에서 사용된 Alpha 21264(EV6)의 플로어플랜(L2 캐쉬 제외)이다[14]. 본 실험에서는 코어의 구성요소는 아래의 플로어플랜으로 고정하여 실험하며 코어에서 가장 뜨거운 구성요소인 IntReg의 위치를 변경하기 위해서 코어를 블록단위로 지정하고 상, 하, 좌, 우로 대칭 시켜 IntReg의 위치를 변경시킨다.

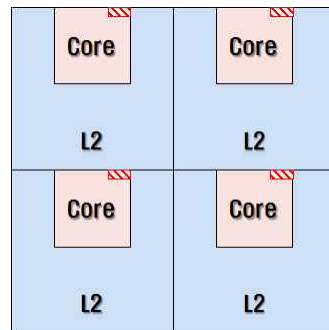


그림 2. 대상 프로세서 기본 플로어플랜
Fig. 2. Baseline floorplan

그림 2는 실험에서 사용된 기본 플로어플랜 구조이다. 기본 플로어플랜은 4개의 코어(L2 캐쉬 포함)를 사각형으로 배치하였다. 이 구조는 Alpha 21264(EV6)를 플로어플랜의 변화 없이 배치하였다. 코어의 오른쪽 윗부분(대각선 줄)은 코어에서 가장 뜨거운 구성요소인 IntReg이다. 이처럼 코어를 사각형으로 배치할 경우 코어에서 가장 뜨거운 구성요소인 IntReg에서 높은 발열이 발생할 것으로 예상된다. 따라서 본 논문에서는 IntReg의 온도 감소를 위해 IntReg의 위치를 변경시켜 플로어플랜 하였다.

2. 저온도 플로어플랜

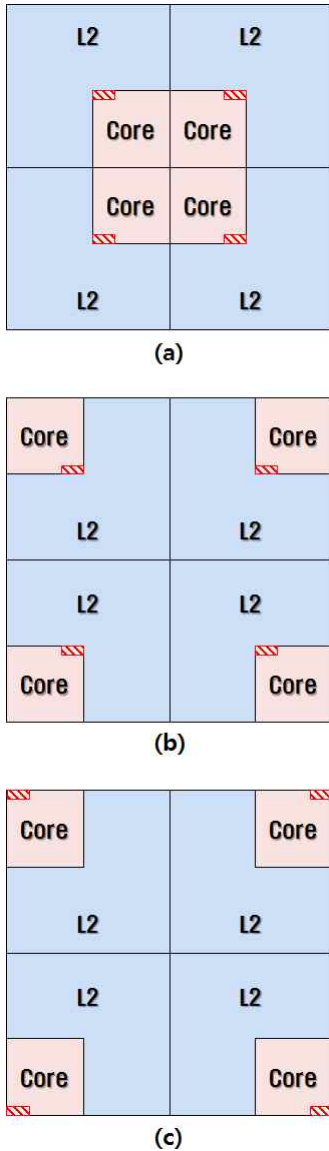


그림 3. 멀티코어 프로세서의 저온도를 위한 3가지 플로어플랜
 Fig. 3. Proposed three floorplan schemes for thermal-aware multi-core processor

본 논문에서 특정 구성요소의 온도감소를 위해 제안하는 플로어플랜은 위와 같다. 제안하는 3가지의 플로어플랜은 코어의 위치를 칩의 중앙에 배치하거나 칩의 외곽에 배치하여

특정 구성요소의 온도를 감소시킨다. 먼저 첫 번째 플로어플랜인 그림 3 (a)는 코어를 칩의 중앙에 배치한 후 코어에서 가장 뜨거운 구성요소인 IntReg의 방향을 바깥으로 향하게 배치한다. 이와 같은 방법은 IntReg의 높은 온도를 상대적으로 온도가 낮은 L2 캐쉬에서 상쇄시킬 것으로 예상된다. 두 번째 플로어플랜인 그림 3 (b)는 코어를 칩의 외곽에 배치한 후 IntReg의 방향을 칩 내부로 향하게 배치한다. 이와 같은 방법은 L2 캐쉬를 활용하여 IntReg의 온도를 감소시키는 그림 3 (a)의 원리와 비슷하지만 코어의 위치가 칩의 외곽에 있어 코어가 칩의 중앙에 모여있는 그림 3 (a)와는 다른 결과가 나올 수 있다. 마지막으로 세 번째 플로어플랜인 그림 3 (c)는 코어를 칩 외곽에 배치한 후 IntReg의 방향을 칩 외부로 향하게 배치한다. 이와 같은 방법은 IntReg를 L2 캐쉬를 통한 냉각 효과가 아닌 칩 외부의 공기를 통한 냉각 효과를 보기 위한 플로어플랜이다.

III. 모의 실험

1. 실험 환경

본 논문에서는 특정 구성요소의 온도감소를 위한 플로어플랜을 제안하였다. 제안된 플로어플랜에 따른 각 구성요소의 온도를 상세하게 측정하고 분석하기 위해 다음과 같은 실험환경을 구성하였다.

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
L1 I-Cache	32KB, 4-way, 32byte lines, 1 cycle latency
L1 D-Cache	32KB, 4-way, 32byte lines, 1 cycle latency
L2 Cache	2MB, 8-way, 128byte lines, 12 cycle latency

프로세서의 내부를 구성하는 코어와 L2 캐쉬의 특성을 상세하게 분석하기 위해 SimpleScalar[15] 시뮬레이터를 사용하였고, 프로세서에서 소모되는 소모 전력을 각 구성요소별로 측정하기 위해 Wattch[16]를 사용하였다. 실험에서 사용된 프로세서의 구성 변수는 표 1에서 보이는 바와 같다. 마지

막으로 각 플로어플랜에 따른 구성요소의 온도분포를 측정하기 위해서 온도 시뮬레이션이 가능한 Hotspot 5.0[17]을 사용하여 온도 분포를 측정하였다.

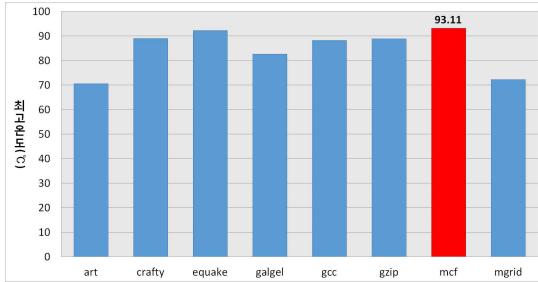


그림 4. 벤치마크별 IntReg 최고 온도
Fig. 4. IntReg peak temperature of benchmark

실험을 위한 벤치마크 프로그램으로는 SPEC CPU2000[18]의 mcf를 사용하였다. mcf는 동일한 조건으로 실험 하는 경우 다른 벤치마크 프로그램들보다 높은 온도를 보이기 때문에 플로어플랜에 따른 온도 분포를 보다 쉽게 파악할 수 있다. 이를 증명하기 위한 기초실험으로써 벤치마크별 구성요소의 최고온도를 조사하였다(실수형 벤치마크 4개, 정수형 벤치마크 4개). 그림 4는 벤치마크별 구성요소의 최고온도이다. 그림에서 알 수 있듯이 mcf에서 93.11°C로 가장 높은 온도가 발생함을 확인할 수 있다.

실험에서 사용된 각 플로어플랜에 따라 코어의 위치와 L2 캐쉬의 위치가 변경된다. 이러한 플로어플랜의 변화에 따라 내부 연결망의 길이가 변경되는데, 실험에서 사용된 SPEC CPU2000 벤치마크는 대부분 코어 내부의 L1 캐쉬에 접근하고 L2 캐쉬로의 접근은 2% 미만이 발생하기 때문에 L2 캐쉬 내부 연결망 길이 변경에 의한 성능은 고려하지 않는다 [11].

2. 실험 결과

4개의 코어를 사용하는 쿼드코어 프로세서에서 온도감사를 위한 3가지의 플로어플랜의 분석 결과는 다음과 같다.

그림 5는 각 플로어플랜별 구성요소의 최고온도이다. 왼쪽부터 순서대로 기본 플로어플랜, 제안하는 플로어플랜 (a), (b), (c)이다.

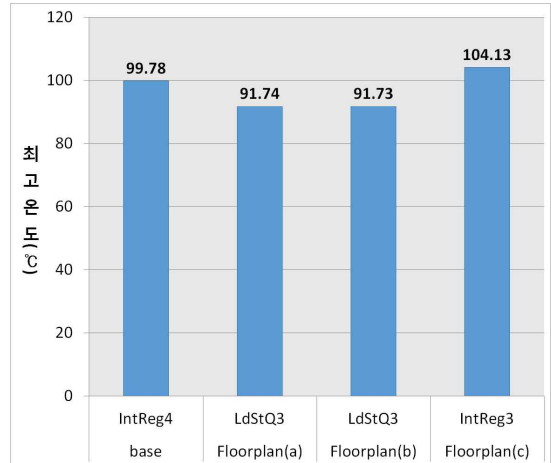


그림 5. 플로어플랜에 따른 구성요소의 최고온도
Fig. 5. Peak temperature according to floorplan

먼저, 기본 플로어플랜의 경우 99.78°C로 IntReg에서 가장 높은 온도를 보인다. 이처럼 특정 구성요소에서 매우 높은 온도가 발생하게 되면 칩의 신뢰성과 수명에 문제가 발생할 수 있다. 그래프를 보면 플로어플랜 (b)에서 91.73°C로 가장 낮은 온도를 보임을 알 수 있다. 또한, 플로어플랜 (a)도 91.74°C로 플로어플랜 (b)와 비슷한 온도를 보인다. 본 논문에서는 플로어플랜에 따른 각 구성요소의 온도를 자세히 분석하기 위해 각 플로어플랜의 온도그래프를 구하였다. 아래 그림은 기본 플로어플랜에 따른 각 구성요소의 온도분포이다.

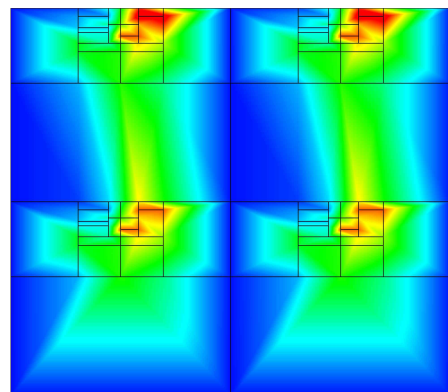


그림 6. 기본 플로어플랜의 온도 분포
Fig. 6. Temperature distribution of baseline floorplan

기본 플로어플랜의 경우 IntReg에서 가장 높은 온도를 보인다. IntReg1과 IntReg4의 경우 99.78°C로 칩에서 가장 높은 온도를 보이며 IntReg2와 IntReg3은 90.86°C로 다른

IntReg보다 낮은 온도를 보인다(오른쪽 위부터 시계방향으로 IntReg1, IntReg2, IntReg3, IntReg4). 이러한 이유는 아래쪽의 IntReg들의 경우 상대적으로 온도가 낮은 L2 캐쉬와 인접하여 배치되어 있기 때문에 L2 캐쉬에 의한 냉각 효과를 받지만 위쪽의 IntReg들의 경우 바로 공기와 접촉하기 때문에 냉각 효과를 받지 못한다. 이처럼 칩의 플로어플랜을 고려하지 않을 경우 특정 구성요소에서 매우 뜨거운 온도가 발생하여, 칩에 심각한 문제가 발생할 수 있음을 알 수 있다.

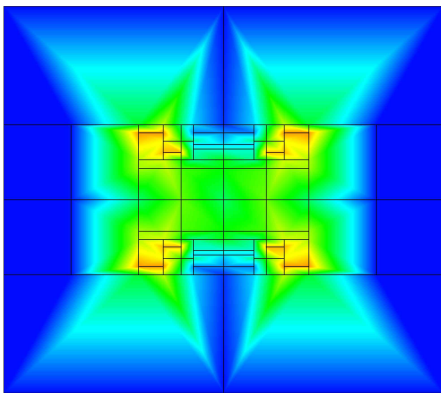


그림 7. 저온도 플로어플랜 (a)의 온도 분포
Fig. 7. Temperature distribution of floorplan (a)

위의 그림 7은 저온도 플로어플랜 (a)에 대한 온도 분포이다. 플로어플랜 (a)는 칩에서 가장 뜨거운 구성요소인 IntReg의 온도를 감소시키기 위해 코어의 위치를 중앙으로 배치하고 코어에서 가장 뜨거운 구성요소인 IntReg를 상대적으로 낮은 온도를 보이는 L2 캐쉬와 인접하게 배치하여 L2 캐쉬의 냉각 효과를 활용하도록 플로어플랜 하였다. 그림 7의

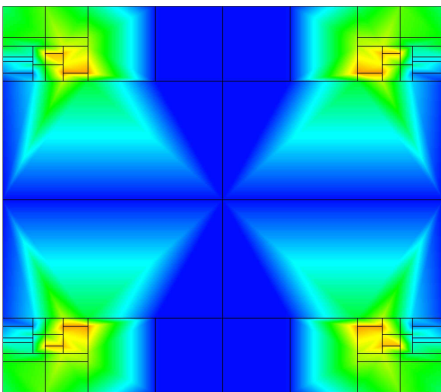


그림 8. 저온도 플로어플랜 (b)의 온도 분포
Fig. 8. Temperature distribution of floorplan (b)

경우 가장 뜨거운 구성요소는 LdStQ으로 91.74°C의 온도를 보인다. L2 캐쉬의 냉각 효과에 따라 기본 플로어플랜에서 가장 높은 온도를 보인 구성요소인 IntReg의 온도는 크게 감소하여 86.77°C로 LdStQ보다 낮은 온도를 보인다. 따라서 플로어플랜 (a)는 기존의 플로어플랜과 비교하여 칩의 온도를 크게 감소시키며 특정 구성요소의 온도 감소 효과가 있음을 확인할 수 있다.

그림 8은 저온도 플로어플랜 (b)에 대한 온도 분포이다. 플로어플랜 (b)는 플로어플랜 (a)와 비슷하게 코어에서 가장 뜨거운 구성요소인 IntReg를 L2 캐쉬와 인접하게 배치하였다. 단 코어를 중앙으로 배치한 플로어플랜 (a)와 달리 코어의 위치를 칩의 외각으로 배치하였다. 그림 8의 경우 가장 뜨거운 구성요소는 LdStQ로 91.73°C의 온도를 보인다. 플로어플랜 (a)와 마찬가지로 L2 캐쉬의 냉각 효과에 의해 IntReg의 온도가 크게 감소하여 86.76°C의 온도를 보인다. 하지만 코어의 위치를 중앙에 집중시키지 않고 칩의 외곽으로 배치하였기 때문에 각 코어에서 발생하는 발열이 분산되었기 때문에 플로어플랜 (a)보다 낮은 최고온도를 보인다.

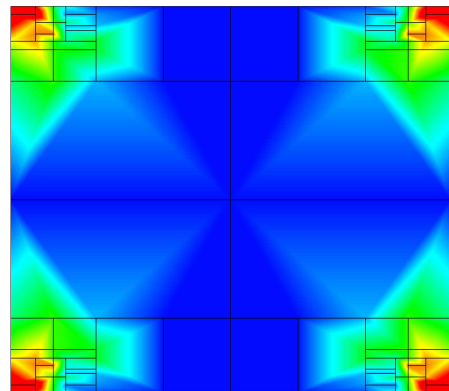


그림 9. 저온도 플로어플랜 (c)의 온도 분포
Fig. 9. Temperature distribution of floorplan (c)

그림 9는 저온도 플로어플랜 (c)에 대한 온도 분포이다. 플로어플랜 (c)는 가장 뜨거운 구성요소인 IntReg의 온도 감소를 위해 칩의 외곽에 배치하였고, 공기와 접촉하여 온도 감소 효과를 활용하고자 하였다. 하지만 실험 결과 IntReg에서 104.13°C의 온도를 보이며 기본 플로어플랜 구조보다 높은 온도가 발생하였다. 이러한 이유는 공기에 접촉시킨 구성요소는 온도 감소 효과가 미미하고 L2 캐쉬에 의한 냉각 효과를 전혀 받지 못하기 때문이다.

실험 결과 IntReg의 위치를 L2 캐쉬와 인접하게 배치하여 냉각 효과를 활용한 플로어플랜 (a), 플로어플랜 (b)에서

특정 구성요소의 온도 감소에 매우 효과적임을 알 수 있었다. 또한, 코어를 중앙으로 밀집시킨 플로어플랜 (a)보다 코어를 분산시킨 플로어플랜 (b)에서 보다 높은 온도 감소 효과를 확인하였다. 그리고 칩의 구성요소는 공기와 접촉하였을 경우 냉각 효과가 높지 않음을 확인하였다.

IV. 결론

본 논문에서는 멀티코어 프로세서에서 특정 구성요소의 온도 감소를 위해 다양한 플로어플랜을 적용하여 플로어플랜에 따른 구성요소의 온도 변화를 상세하게 분석하였다. 총 4가지의 플로어플랜을 실험한 결과 L2 캐쉬를 코어의 뜨거운 구성 요소에 인접하게 배치할 경우 온도 감소 효과가 매우 뛰어난 것을 알 수 있었고, 코어의 위치 또한 서로 집중시키는 것보단 분산시키는 것이 온도 감소에 효과적임을 알 수 있었다. 마지막으로 코어의 뜨거운 구성요소를 L2 캐쉬가 아닌 공기와 접촉시켰을 경우에는 냉각 효과가 미미함을 알 수 있었다. 기본 플로어플랜과 비교하여 플로어플랜 (a)는 최고온도가 8.04°C 감소하였고, 플로어플랜 (b)는 최고온도가 8.05°C 감소하였다. 하지만 플로어플랜 (c)는 최고온도가 4.35°C 증가하였다. 이는 L2 캐쉬가 칩의 냉각장치 역할을 함으로써 코어의 뜨거운 구성요소가 L2 캐쉬가 인접하게 배치되었는지에 따라 칩의 최고온도 차이가 발생하기 때문이다. 멀티코어 프로세서를 위한 저온도 플로어플랜은 멀티코어 프로세서의 문제점인 발열 문제를 해결하기 위한 하나의 대안이 될 수 있다. 본 논문에서 분석한 멀티코어 프로세서를 위한 저온도 플로어플랜의 결과를 잘 활용한다면 최신의 마이크로프로세서를 보다 효율적으로 설계할 수 있고, 발열 문제로 인한 냉각 비용을 크게 감소함으로써 제품 설계 시 경쟁력을 높일 것으로 기대된다.

참고문헌

[1] P. Dadvar, and K. Skadron, "Potential thermal security risks," In Proceedings of the IEEE/ASME Semiconductor Thermal Measurement, Modeling, and Management Symposium (SEMI-THERM), pp. 229-234, San Jose, USA, March. 2005.

[2] J. H. Choi, "Thermal Management for Multi-core Processor and Prototyping Thermal-aware Task

Scheduler," Journal of KIISE : Computer Systems and Theory, Vol.35, No.7•8, pp.354-360, Aug. 2008.

[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, No. 4, pp. 629-634, Dec. 2002.

[4] L. Benini, G. De Micheli, E. Macii, M. Poncino, and R. Scarsi, "Symbolic synthesis of clock-gating logic for power optimization of synchronous controllers," In Transactions on Design Automation of Electronic Systems (TODAES), Vol. 4, Issue. 4, pp. 351-375, Oct. 1999.

[5] P. Falkenstern, Y. Xie, Y. W. Chang, and Y. Wang, "Three-Dimensional Integrated Circuits (3D IC) Floorplan and Power/Ground Network Co-synthesis," In Proceeding of Design Automation Conference (ASP-DAC), pp.169-174, Taipei, Taiwan, Jan. 2010.

[6] A. Gupta, N. D. Dutt, F. J. Kurdahi, K. S. Khouri, and M. S. Abadir "STEFAL: A System Level Temperature- and Floorplan-Aware Leakage Power Estimator for SoCs," In Proceeding of VLSI Design held jointly with 6th International Conference on Embedded Systems, pp.559-564, Bangalore, India, Jan. 2007.

[7] P. N. Guo, C. K. Cheng, and T. Yoshimura, "An 0-Tree representation of non-slicing floorplan and its applications," In Proceeding of Design Automation Conference(DAC), pp.268-273, New Orleans, USA, June. 1999.

[8] C. H. Tsai, and S. M. Kang, "Cell-Level Placement for Improving Substrate Thermal Distribution," IEEE Tram. On Computer.-Aided Des, Vol. 19, No. 2, pp. 253-266, Feb. 2000.

[9] K. W. Lee, T. Nakamura, T. Ono, Y. Yamada, T. S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," In Proceedings of IEEE/ACM ICCAD, pp.484-491, San Jose, USA,

Nov. 1996.

[10] H. Murata, and E. S. Kuh, "Sequence Pair Based Placement Method for Hard/Soft/Pre-placed Modules," In Proceeding of International Symposium on Physical Design(ISPD), pp.167-172, Monterey, California, USA, April. 1998.

[11] Microarchitectural Floorplanning for Thermal Management: A Technical Report, available at <http://www.cs.virginia.edu/~techrep/CS-2005-08.pdf>

[12] D. O. Son, J. W. Ahn, J. H. Park, J. M. Kim, and C. H. Kim, "Analysis on the Temperature of 3D Multi-core Processors according to Vertical Placement of Core and L2 Cache," Journal of The Korea Society of Computer and Information, Vol. 16, No. 6, pp. 1-10, June. 2011.

[13] Coskun A.K, Kahng A.B, and Rosing T.S, "Temperature- and Cost-Aware Design of 3D Multiprocessor Architectures," In Proceedings of 12th Euromicro conference on Digital System Design, Architectures, Methods and Tools, pp.183-190, Patras, Greece, Aug. 2009.

[14] R. E. Kessler, E. J. McLellan, and D. A. Webb, "The Alpha 21264 Microprocessor Architecture," In Proceedings of the ICCD '98, pp.90-95, Austin, USA, Aug. 2002.

[15] D. C. Burger, and T. M. Austin, "The SimpleScalar tool set, version 2.0," ACM SIGARCH CAN, Vol. 25, No. 3, pp. 13-25, Jun. 1997.

[16] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in Proceedings of the 27th International Symposium on Computer Architecture, pp.83-94, Vancouver, Canada, Jun. 2000.

[17] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan and S. Ghosh, "HotSpot: A Compact Thermal Modeling Method for CMOS VLSI Systems," IEEE Transactions on VLSI Systems, Vol 14, No 5, pp. 501-513, May.

2006.

[18] SPEC CPU2000 Benchmarks, available at <http://www.specbench.org>

저 자 소 개



손 동 오
 2010: 전남대학교
 전자컴퓨터공학부 공학사
 2012: 전남대학교
 전자컴퓨터공학과 석사
 현 재: 전남대학교
 전자컴퓨터공학과 박사과정
 관심분야: 컴퓨터구조, 임베디드시스템
 Email : sdo1127@gmail.com



김 종 면
 1995: 명지대학교 전기공학사
 2000: University of Florida
 ECE 석사
 2005: Georgia Institute of
 Technology CE 박사
 2005-2007: 삼성종합기술원
 전임연구원
 현 재: 울산대학교
 전기공학부 교수
 관심분야: 임베디드 SoC, 컴퓨터구조,
 프로세서 설계, 병렬처리
 Email : jmkim07@ulsan.ac.kr



김 철 홍
 1998: 서울대학교 컴퓨터공학사
 2000: 서울대학교 컴퓨터공학부 석사
 2006: 서울대학교
 전기컴퓨터공학부 박사
 2005-2007: 삼성전자 반도체총괄
 SYS.LSI사업부
 책임 연구원
 현 재: 전남대학교
 전자컴퓨터공학부 교수
 관심분야: 임베디드시스템, 컴퓨터구조,
 SoC 설계, 저전력 설계
 Email : chkim22@chonnam.ac.kr