

부동소수점 응용을 위한 저온도 마이크로프로세서 설계

(Temperature-Aware Microprocessor Design for Floating-Point Applications)

이 병 석 [†] 김 철 흥 ^{**} 이 정 아 ^{***}
(Byeong Seok Lee) (Cheol Hong Kim) (Jeong-A Lee)

요 약 동적 온도 제어 기술은 마이크로프로세서 내부 특정 유닛의 온도가 크게 올라가는 열섬 문제를 해결하기 위해 널리 사용되는 기법으로 냉각 비용을 감소시키고 칩의 신뢰성을 높인다는 장점이 있지만, 기법 적용으로 인해 성능이 저하되는 단점이 있다. 본 논문에서는 부동소수점 응용 프로그램 수행 시 발열 문제를 해결하기 위해 적용되는 동적 온도 제어 기술로 인한 성능 저하를 최소화하기 위하여 듀얼 부동소수점 가산기 구조를 제안하고자 한다. 부동소수점 응용 프로그램을 수행할 때, 가장 많이 활성화되는 유닛 중 하나인 부동소수점 가산기를 두 개로 중복시켜서 접근을 분산시키는 기법을 통해 열섬 문제를 해결하고자 한다. 또한 상호 인접한 유닛 간의 열 전달로 인해 온도가 상승하는 문제를 해결하기 위하여, 열 전달 지연 공간을 마이크로프로세서 내에 배치시키는 방법을 제안한다. 제안 기법들의 적용 결과, 동적 온도 관리 기술을 사용하는 환경에서 마이크로프로세서의 최고 온도가 평균 5.3°C 최대 10.8°C 낮아지면 서 발열로 인한 칩의 안정성 저하 문제를 완화시킬 수 있다. 또한 동적 온도 관리 기술이 적용되는 시간 을 크게 줄임으로써 프로세서의 성능은 평균 1.41배(최대 1.90배) 향상된다.

키워드 : 마이크로프로세서 설계, 동적 온도 관리, 부동소수점 가산기, 저온도 설계기법

Abstract Dynamic Thermal Management (DTM) technique is generally used for reducing the peak temperature (hotspot) in the microprocessors. Despite the advantages of lower cooling cost and improved stability, the DTM technique inevitably suffers from performance loss. This paper proposes the DualFloating-Point Adders Architecture to minimize the performance loss due to thermal problem when the floating-point applications are executed. During running floating-point applications, only one of two floating-point adders is used selectively in the proposed architecture, leading to reduced peak temperature in the processor. We also propose a new floorplan technique, which creates Space for Heat Transfer Delay in the processor for solving the thermal problem due to heat transfer between adjacent hot units. As a result, the peak temperature drops by 5.3°C on the average (maximum 10.8°C for the processor where the DTM is adopted, consequently giving a solution to the thermal problem. Moreover, the processor performance is improved by 41% on the average by reducing the stall time due to the DTM.

Key words : Microprocessor design, Dynamic Thermal Management, Floating-Point Adder, Temperature-aware design

· 본 연구는 한국연구재단 핵심연구(구 한국과학재단 특정기초연구, R01-2007-000-20750-0)의 지원과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 (IITA-2009-(C1090-0903-0008)) 지원사업의 연구결과로 수행되었음

논문접수 : 2009년 3월 2일
심사완료 : 2009년 8월 12일

[†] 학생회원 : 조선대학교 컴퓨터학과
novrain@chosun.ac.kr
^{**} 종신회원 : 전남대학교 전자컴퓨터공학부 전일강사
chkim22@chonnam.ac.kr
^{***} 종신회원 : 조선대학교 컴퓨터공학부 교수
jalee@chosun.ac.kr

Copyright©2009 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지 : 시스템 및 이론 제36권 제6호(2009.12)

1. 서론

반도체 공정 기술의 발전으로 마이크로프로세서의 성능은 크게 향상되고 있다. 하지만 높은 주파수와 고밀도 집적화로 인해 전력 소모가 크게 증가하면서 칩이 뜨거워지는 발열 현상이 마이크로프로세서의 성능 향상을 제한시키는 문제점으로 새롭게 등장하였다. 칩 일부분의 온도가 급격하게 상승하는 열섬(hotspot) 현상은 마이크로프로세서가 오동작하거나 심할 경우에는 칩의 회로가 손상되어서 신뢰성을 저하시키는 원인이 되기도 한다. 칩의 냉각을 위해 일반적으로 사용되는 히트 스프레더(heat spreader), 방열 판(heat sink), 냉각 팬(cooling fan) 등을 이용한 물리적인 냉각 방법들은 물리적인 한계와 냉각 비용이 상승하는 문제점을 가지고 있다[1]. 최근에는 심각한 발열 현상을 보이는 고성능 프로세서 뿐만 아니라 비용과 성능에 민감한 임베디드 프로세서 또한 디자인 특성으로 인해 공간적인 제약 조건을 만족하면서 최대한 열섬을 줄여 열적 분포(thermal profile)를 균등하게 하는 온도 지향적 설계가 요구되고 있다 [2]. 따라서 최신 마이크로프로세서를 설계할 때에는 성능과 전력 소비 문제뿐만 아니라 발열 문제도 함께 고려하여야 한다.

마이크로프로세서의 발열 문제를 해결하기 위해 주로 사용되는 동적 온도 관리(DTM; Dynamic Thermal Management)[3] 기술은 냉각 비용을 감소시키고 동시에 칩의 신뢰성을 높이는 장점이 있지만, 마이크로프로세서의 성능이 저하된다는 단점이 있다. 그러므로 마이크로프로세서의 발열 문제를 해결하면서 동시에 성능 저하를 최소화할 수 있는 해결책이 요구된다. 지금까지 수행된 성능 저하를 최소화하면서 발열 문제를 해결하기 위한 기법들은 대부분 마이크로프로세서 내에서 가장 뜨거운 정수 레지스터 유닛을 대상으로 연구되었다. 하지만 정수 레지스터 유닛을 대상으로 하는 기존의 방법들은 DSP나 멀티미디어 시스템과 같이 부동소수점 연산량이 많은 응용 프로그램을 수행할 경우 별다른 효과가 없으며[4], 멀티미디어 재생 기술에서 서비스 품질(QoS)을 유지하기 위해서는 동적 온도 제어 기법의 수행을 최소화하는 기법이 요구되고 있다[5].

본 논문에서는 부동소수점 응용 프로그램을 수행할 때의 발열 문제와 이로 인한 성능 저하 문제를 해결하기 위하여 듀얼 부동소수점 가산기 구조를 제안한다. 부동소수점 응용 프로그램은 정수형 응용 프로그램과는 다르게 연산기의 발열이 심한 편이기 때문에, 부동소수점 연산 유닛을 중복 배치하여 접근을 분산시킴으로써 온도 상승을 지연시키는 기법을 제안하고자 한다. 또한 모의 실험을 통해 확인된 온도가 높은 유닛으로부터의

열 전달로 인해 인접한 유닛의 온도가 함께 상승하는 문제를 해결하기 위하여 열 전달 지연 공간을 배치하는 방법 또한 제안한다

본 논문의 구성은 다음과 같다. 2장에서는 기존의 온도 제어 기술과 부동소수점 응용 프로그램에서의 발열 특성을 기술한다. 3장에서는 제안하는 듀얼 부동소수점 가산기 구조를 통한 동적 온도 제어 방법과 열 전달 지연 공간 배치를 이용한 정적 온도 제어 방법을 기술한다. 4장에서는 모의 실험을 통해 일반적인 온도 제어 기술과 본 논문에서 제안하는 기술을 비교, 분석한다. 마지막으로 5장에서는 결론과 향후 연구 과제를 기술한다.

2. 관련 연구 및 연구 배경

2.1 마이크로프로세서 온도 제어 기법 연구

마이크로프로세서에 적용되는 온도 제어 기술은 크게 정적 기법과 동적 기법으로 구분된다. 정적 온도 제어 기법은 유닛(unit)을 저전력(low power)으로 설계하고, 발열에 최적화 하도록 유닛을 배치(floorplan)하는 방법이다[6]. 이 방법은 칩의 온도를 간접적으로 관리하므로 냉각에는 한계가 있다. 따라서 마이크로프로세서 내부의 온도를 효과적으로 관리하기 위해 동적 온도 관리(Dynamic Thermal Management)[3] 기법이 함께 사용된다. 동적 온도 관리 기법은 가변 주파수 조절(Dynamic Frequency Scaling), 가변 전압 조절(Dynamic Voltage Scaling), 명령어 인출 지연(Fetch Throttling) 기법 등을 적용하여 마이크로프로세서의 발열 문제를 직접적으로 제어하는 방법이다. 동적 온도 관리 기법은 칩에 내장된 온도 센서를 통해 내부 온도가 사전에 설정한 임계 값에 도달한 것으로 확인될 경우에 수행되는 방식으로, 온도를 낮추는 과정에서 평균적으로 가변 주파수 조절(DFS)은 2%, 가변 전압 주파수 조절(DVFS)은 6~9%, 명령어 인출 조절은 8%의 성능 저하가 발생한다. 마이크로프로세서 내부 유닛의 온도가 일정 수준 이상으로 계속 발열할 경우에는, 동적 온도 관리 수행으로 인해 해당 유닛의 비활성화가 지속되면서 마이크로프로세서의 성능이 크게 떨어지는 문제점이 있다.

동적 온도 관리를 최소한으로 수행하기 위해 제안된 기법들은 다음과 같다. 성능 계수기(on-chip performance counter)를 이용하여 발열을 예측한 후, 미리 온도 제어 기술을 사용하여 온도 상승을 억제함으로써 동적 온도 관리를 최소한으로 동작하게 하는 기법이 연구되었다[7]. 보조 유닛을 추가하여 주 유닛의 사용률을 낮춤으로써 발열을 감소시키는 연산 이관(computation migrating) 기법 또한 연구되었다[8]. 연산 이관 기법 중에서는 주 정수 레지스터의 값을 보조 정수 레지스터에 복사한 후, 주 정수 레지스터의 온도가 임계 온도 이상이

되면 주 정수 레지스터 대신에 보조 레지스터를 사용하여 정수 레지스터의 온도를 제어하는 기술이 대표적이다 [9]. 이 기술은 연산 이관을 실행할 때 파이프라인을 멈춘 후 데이터를 다른 레지스터에 복사해야 하므로 마이크로프로세서의 성능이 낮아지는 문제점이 있다. 다른 연산 이관 기법으로는 하나의 레지스터를 두 영역으로 분할한 후, 분할된 영역을 상호 번갈아 가면서 이용하는 연산 분할(computation division)이 있다[10]. 연산 분할 기술 역시 분할된 레지스터의 위치가 상호 인접하기 때문에 서로 열이 전달되어 레지스터에서 열섬이 발생하는 경우에는 마이크로프로세서의 실행을 중지해야 하는 문제점이 발생한다. 이와 같은 연산 이관 기법들의 단점을 해결하기 위한 기법으로는 서로 떨어진 두 개의 레지스터를 번갈아 접근하는 듀얼 레지스터 구조가 있다[4].

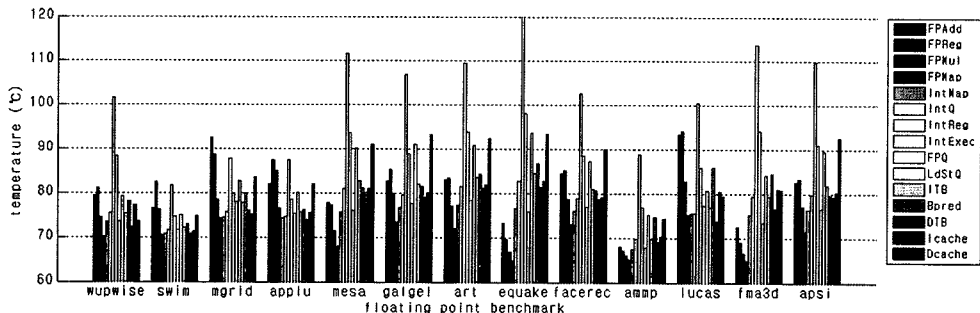
지금까지 수행된 대부분의 저온도 연구는 발열이 가장 높은 정수 레지스터에 집중되었다. 하지만 정수 레지스터의 온도를 효과적으로 제어하더라도 부동소수점 연산량이 많은 응용프로그램을 수행하는 도중에는 정수 레지스터가 아닌 다른 유닛의 발열로 인해 칩의 안정성과 성능에 문제가 발생할 수 있다[4]. 또한 대부분의 부동소수점 응용 프로그램은 고속 연산을 요구하기 때문에 동적 온도 제어로 인한 성능 감소 또한 최소화 되어야 한다. DSP와 멀티미디어와 같은 부동소수점 응용

프로그램을 주로 수행하는 프로세서에 대한 발열 연구는 거의 이루어지지 않다가 최근 MPEG-4, H.264/AVC 등 비디오 디코딩 분야에서 동적 온도 제어에 따라 동영상 프레임의 제어하는 기법에 대한 연구가 있지만[5], 동영상 재생이라는 제한된 분야에만 적용 가능한 기법이다. 따라서 본 논문에서는 이와 같은 상황을 바탕으로 부동소수점 응용 프로그램을 수행하는 경우 마이크로프로세서 내부 온도를 효과적으로 제어 하기 위한 기법을 제안하고자 한다.

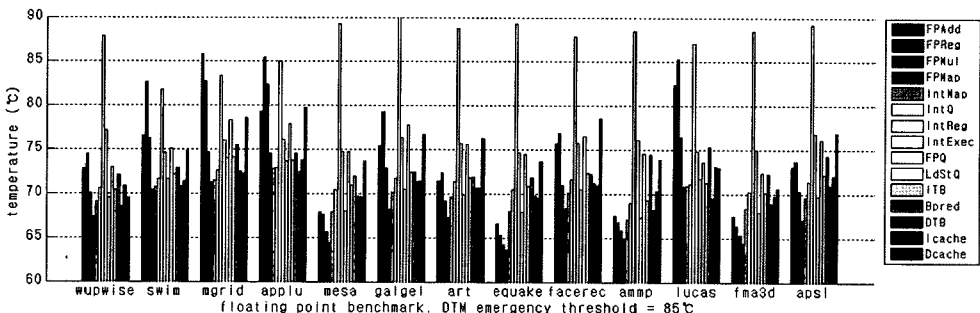
2.2 부동소수점 응용 프로그램 수행 시 열섬 현상

국제 반도체 기술 로드맵(ITRS; International Technology Roadmap for Semiconductors)에서는 반도체 공정이 0.13um 이하에서는 칩의 최고 적합 온도(maximum junction temperature)가 90°C 보다 낮아야 한다고 발표하였다[11]. 이 조건을 만족하기 위하여 표본 집단 중에서 최악의 경우에 해당하는 응용 프로그램(worst typical application) 이 수행되는 동안의 온도를 추적하여 동적 온도 제어를 수행하는 기법이 제안되었다[9].

그림 1의 데이터는 Alpha 21264 마이크로프로세서에서 SPEC CPU2000[12] 부동소수점 벤치마크 프로그램(CFP2000)를 실행할 때, 각 유닛의 최고 온도를 추적한 결과이다(모의 실험 방법에 관한 설명은 4장에서 자세히 설명한다).



(a) 동적 온도 관리를 적용하지 않을 때의 유닛별 최고온도



(b) 동적 온도 관리를 적용할 때의 유닛별 최고온도

그림 1 Alpha 21264 마이크로프로세서의 기능 유닛에 대한 최고 온도

그림 1의 (a)는 동적 온도 제어 기법이 사용되지 않는 상황에서 각 유닛에 대한 최고 온도를 나타낸다. 정수 레지스터(IntReg: 120.9°C 온도가 가장 높고, 정수 연산기(IntExec: 98.0°C, 부동소수점 레지스터(FPReg: 94.2°C, 적재/저장 큐(LdStQ: 93.7°C, 데이터 캐시(Dcache: 93.5°C, 부동소수점 가산기(FPAdd: 93.5°C 순서로 온도가 높다. 대부분의 부동소수점 벤치마크 프로그램에서도 정수 연산과 관련 있는 유닛의 온도가 가장 높으나, swim, mgrid, applu 벤치마크 프로그램에서는 부동소수점 연산과 관련 있는 유닛의 온도가 가장 높게 나타난다. 그림 1의 (b)는 긴급 단계(emergency threshold)의 온도를 85°C로 설정한 후 동적 온도 제어 기술을 적용할 때, 각 유닛에 대한 최고 온도를 나타낸다. 이 결과 역시 정수 레지스터(IntReg: 90.4°C의 온도가 가장 높지만, 그림 1의 (a)와는 다르게 부동소수점 가산기(FPAdd: 85.8°C, 부동소수점 레지스터(FPReg: 85.3°C 순서로 온도가 높은 것을 볼 수 있다. 또한 swim과 mgrid에서는 부동소수점 연산 유닛인 FPReg(82.6°C와 FPAdd(92.6°C가 정수 레지스터(IntReg)보다 온도가 더 높은 것을 확인할 수 있다. 이는 곧 부동소수점 연산량이 많은 경우에는 부동소수점 연산기에 대한 접근 비율이 높아서 해당 유닛에 대한 냉각 효과가 저조하다는 것을 의미한다. 따라서 정수형 응용 프로그램을 실행할 때에는 정수 레지스터의 온도만 제어하여도 충분한 냉각 효과를 볼 수 있지만, 부동소수점 응용 프로그램을 실행할 때에는 부동소수점 레지스터와 부동소수점 가산기 역시 집중적인 온도 제어가 필요하다는 사실을 확인할 수 있다.

부동소수점 응용 프로그램을 주로 수행하는 DSP나 멀티미디어 시스템은 시스템 품질(QoS)이 중요한 요소이다. 이를 만족하기 위해 부동소수점 응용 프로그램은 고속의 연산 장치를 통한 빠른 연산 시간을 요구한다. 따라서 동적 온도 제어 시간을 최소화하여 프로세서의 성능 저하 또한 최소화해야 할 필요가 있다. 즉, 부동소수점 응용 프로그램이 실행되는 도중에는 부동소수점 연산에 필요한 유닛의 발열뿐만 아니라, 동적 온도 제어로 인한 성능 저하 문제까지 함께 고려해야 한다. 여기서는 부동소수점 레지스터와 가산기 중에서 열섬 가능 인자(hotspot-possible factor)를 선택하는 문제가 발생한다. 부동소수점 레지스터를 열섬 가능 인자로 선택한 후 기존의 연산 이관 기법[9]을 적용하면, 마이크로프로세서의 성능이 저하되는 문제가 발생한다. 다른 방법인 듀얼 레지스터 구조[4]를 적용하면 부동소수점 레지스터로 인한 발열 문제는 해결할 수 있지만, 부동소수점 가산기로 인한 발열 문제까지는 해결할 수는 없다. 결국은 부동소수점 가산기의 발열 문제로 인하여 마이크로프로

세서의 안정성 및 성능 저하 문제가 발생할 수 있다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 부동소수점 가산기(FPAdd)를 열섬 가능 인자로 선택하고, 유닛의 발열에 대한 안정성 향상과 동적 온도 관리로 인한 성능 저하를 최소화 하는 기술을 제안하고자 한다.

3. 부동소수점 응용 프로그램을 위한 저온도 유지 기법

3.1 듀얼 부동소수점 가산기 구조

부동소수점 응용 프로그램은 부동소수점 연산량이 많고 성능 저하에 민감하다는 특징이 있다. 그러므로 부동소수점 응용 프로그램을 위한 저온도 마이크로프로세서를 설계할 때에는 부동소수점 연산량에 따른 부동소수점 관련 유닛의 온도 상승을 해결하고, 동시에 동적 온도 제어로 인한 성능 저하를 최소화 할 수 있도록 설계해야 한다.

유닛의 온도는 해당 유닛이 동작 중에 소비하는 전력량과 비례한다. 유닛이 동작하지 않도록 비활성화가 되면 소비 전력이 감소하면서 온도 역시 낮아지므로 이를 활용하여 유닛을 냉각시킬 수 있다. 하지만 유닛이 비활성화되는 시간만큼 마이크로프로세서의 성능이 저하되는 문제점이 발생한다. 특히 성능 저하에 민감한 부동소수점 응용 프로그램에서는 온도 저하를 위한 성능 감소 현상은 치명적이라고 할 수 있다. 본 논문에서는 마이크로프로세서의 추가적인 성능 저하 없이 부동소수점 가산기에서의 온도 상승을 최소화하기 위하여 부동소수점 가산기를 두 개로 분산하여 연산을 이관하는 듀얼 부동소수점 가산기 구조(dual floating-point adder unit architecture)를 제안하고자 한다.

제안하는 듀얼 부동소수점 가산기 구조는 보조 부동소수점 가산기를 추가 배치함으로써 부동소수점 가산기에 대한 접근을 분산하는 기술이다. 하나의 부동소수점 가산기가 활성화되어서 동작하는 동안에는 유닛의 온도가 올라가지만, 다른 비활성화된 부동소수점 가산기는 동작을 하지 않는 동안에 냉각이 되는 원리를 이용하여

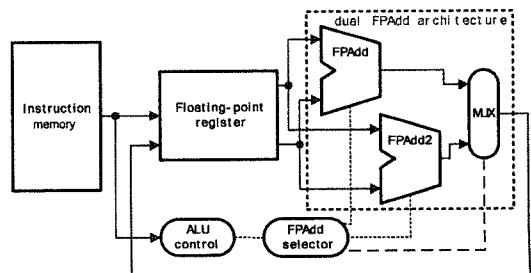


그림 2 듀얼 부동소수점 가산기 구조

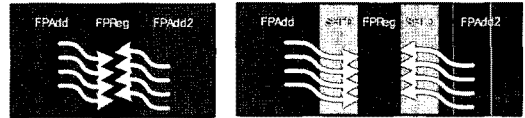
부동소수점 가산기의 온도를 제어한다. 그림 2에서 보이는 바와 같은 듀얼 부동소수점 가산기 구조는 다음과 같이 동작한다. 먼저, 부동소수점 가산 명령어가 명령어 메모리(instruction memory)에서 인출(fetch)되면, 명령어를 해석한 후 연산기 제어기(ALU control)에서 부동소수점 가산기 선택기(FPAdd selector)가 동작하도록 설정한다. 연산에 필요한 인자 값은 부동소수점 레지스터로부터 주 부동소수점 가산기(FPAdd)와 보조 부동소수점 가산기(FPAdd2) 모두에 전송된다. 두 개의 부동소수점 가산기는 연산에 필요한 인자 값을 받지만, 부동소수점 가산기 선택기(FPAdd selector)에 의해 선택되는 부동소수점 가산기만이 동작하면서 전력을 소비한다. 이와 같은 구조를 활용하여 부동소수점 연산 동작을 두 개의 부동소수점 가산기로 분산하면, 각 부동소수점 가산기의 소비 전력이 낮아지면서 발열 역시 낮아지게 될 것으로 기대된다. 마지막으로 부동소수점 가산기 선택기(FPAdd selector)는 실제 부동소수점 연산을 수행한 부동소수점 가산기의 출력 값을 MUX를 통하여 부동소수점 레지스터에 저장한다. 제안하는 구조에서는 활성화할 유닛을 선택하는 과정에서 발생하는 지연 현상이 나타나지 않으므로 성능 저하 현상이 발생하지 않는다는 장점이 있다.

제안하는 듀얼 부동소수점 가산기 구조는 다음과 같은 문제점을 가지고 있다. 첫 번째는 하나의 부동소수점 가산기가 추가되는 만큼 칩 면적이 증가한다는 것이다. 두 번째는 하나의 부동소수점 레지스터에서 두 개의 부동소수점 가산기로 데이터 전송 경로를 연결해야 하기 때문에 부동소수점 가산기와 부동소수점 레지스터 사이의 데이터 패스가 복잡해진다는 점이다. 만약 레지스터와 가산기 사이가 멀어지면 데이터 전송 경로는 더욱 복잡해지는 문제가 발생한다. 따라서 제안하는 구조에서 부동소수점 레지스터와 부동소수점 가산기는 상호 인접해야 한다.

3.2 열 전달 지연 공간을 사용한 유닛 배치

열은 온도가 높은 곳에서 낮은 곳으로 전달된다. 즉, 두 개의 유닛이 서로 인접해 있으면 온도가 높은 유닛에서 온도가 낮은 유닛으로 열이 전달(heat transfer)된다. 그러므로 온도가 낮은 유닛이라도 온도가 높은 유닛에 인접해 있으면, 활성화 빈도와는 관계 없이 온도가 상승하면서 발열 안정성과 마이크로프로세서의 성능에 큰 영향을 줄 수 있다. 본 논문에서는 열 전달로 인한 유닛의 온도 상승 문제를 해결하기 위하여, 인접한 두 유닛 사이에 열 전달 지연 공간(Space for Heat Transfer Delay)을 마이크로프로세서 코어(core) 내부에 추가로 배치하는 정적인 온도 제어 방법 또한 제안하고자 한다.

앞서 기술한 바와 같이 제안한 듀얼 부동소수점 가산기 구조는 데이터 패스로 인한 성능 저하를 최소화하기 위해서 그림 3의 (a)와 같이 배치가 이루어진다. 이러한 경우 두 개의 부동소수점 가산기(FPAdd와 FPAdd2)의 발열로 인하여 인접한 부동소수점 레지스터(FPReg)의 온도가 더불어 상승하는 현상이 발생할 수 있다. 마이크로프로세서의 성능 저하를 최소화 하기 위해서는 부동소수점 레지스터의 온도를 낮추는 방법 또한 필요하다. 그림 3의 (b)처럼 두 개의 부동소수점 가산기(FPAdd와 FPAdd2)와 부동소수점 레지스터(FPReg) 사이에 열 전달 지연 공간(SHTD)를 배치함으로써 부동소수점 가산기와 부동소수점 레지스터에서의 발열을 완화시킬 수 있다.



(a) 열 전달 지연 공간이 없는 유닛 배치 (b) 열 전달 지연 공간을 이용한 유닛 배치

그림 3 열 전달 지연 공간 배치 기법

4. 모의 실험 및 결과

본 장에서는 모의 실험 도구를 이용하여 동적 온도 제어 환경에서 듀얼 부동소수점 가산기 구조와 열 전달 지연 공간을 배치하는 경우에 대해, 칩의 온도 변화를 살펴봄으로써 발열에 대한 안정성과 온도 상승으로 인한 성능 저하 정도를 평가한다.

4.1 모의 실험 환경

제안된 부동소수점 응용 프로그램을 위한 저온도 마이크로프로세서 설계 기법의 효율성을 분석하기 위하여 본 논문에서는 마이크로프로세서 구조에 따른 성능, 전력 소비 및 온도 변화를 사이클 단위로 측정할 수 모의 실험 도구를 사용하였다. 모의 실험을 위한 목표 프로세서는 0.13um 공정의 3GHz 슈퍼스칼라 Alpha 21264 프로세서로 가정하였고, 특성은 [9]에서와 동일하게 설정하였다. 모의 실험은 Watch[13], HotSpot[14] 프로그램을 하나로 통합하여 수행하였다. Watch 실험 도구는 사이클 단위로 마이크로프로세서의 성능과 내부 유닛들의 전력 소비를 계산하고, HotSpot은 각 유닛의 전력 데이터를 이용하여 마이크로프로세서의 온도 분포를 예측한다. 유닛의 배치(floorplan)는 AlphaEV6(21264)의 코어를 참조하여서 유닛의 영역과 순서를 먼저 설정한 후, hotfloorplan[6]을 이용하여 설계하였다.

모의 실험에서는 마이크로프로세서의 온도가 일정 수준에 도달하면 동적 온도 관리가 수행되면서 온도를 제

어한다. 동적 온도 관리는 2단계로 수행된다. 첫 번째 단계인 구동 단계(trigger threshold)에서는 명령어 인출 지연 방법으로 온도를 조절한다. 두 번째 단계인 긴급 단계(emergency threshold)에서는 가변 전압 및 주파수 조정 방법으로 온도를 조절한다. 긴급 단계에서는 온도를 급격하게 낮춤으로써 칩의 온도가 최대 한계를 넘지 않도록 조절한다. 이러한 2단계 정책은 고성능 프로세서에서 성능 손실을 적게 하면서 온도를 조절하는데 도움이 된다. 본 연구에서는 [4]와 같이 동적 온도 제어의 구동 단계 온도를 긴급 단계 온도보다 2.0°C 낮게 설정하고 실험을 수행하였다.

온도 모델링 모의 실험에서 HotSpot의 설정은 [9]의 명세에 따른다. 정밀한 모의 실험 결과를 얻기 위하여 [4]와 같이 모든 유닛이 60.0°C 에서 시작하도록 온도를 설정하고, 3억 개의 명령어를 프로세서의 예열(warm-up) 기간으로 설정한다. 예열 기간이 끝나면, HotSpot이 출력하는 안정 상태 온도(steady state temperature)로 프로세서의 초기 온도를 설정한다. 캐시와 분기 예측의 준비 기간은 예열 기간보다 작은 2억 개로 설정한다. 모의 실험에는 많은 시간이 필요하기에 명령어의 실행 개수를 5억 개로 제한한다. 실행 프로그램으로는 SPEC CPU2000[12] 중에서 부동소수점 벤치마크(CFP2000)를 사용한다.

표 1은 AlphaEV6(21264)의 각 유닛에 대한 면적 명세를 보여준다. 표에서 Ev6는 AlphaEV6(21264)의 코어 구조를 의미하고, Ev6.fpadd2는 Ev6 코어 구조에 본 논문에서 제안한 듀얼 부동소수점 가산기를 적용한 코어 배치를 의미한다. Ev6.fpadd2s는 Ev6.fpadd2에 열 전달

지연 공간을 추가한 코어 배치를 나타낸다. 또한, SHTD는 열 전달 지연 공간을 의미한다. SHTD1과 SHTD2는 부동소수점 레지스터(FPReg)의 면적과 동일하게 설정하였고, SHTD3과 SHTD4는 정수 레지스터의 면적과 동일하게 설정하였다. 듀얼 부동소수점 가산기 구조(Ev6.fpadd2)를 사용하면 칩의 면적은 기존의 Ev6 코어보다 0.77%정도 증가하고, 부동소수점 레지스터와 정수 레지스터 주변에 열 전달 지연 공간을 추가로 배치(Ev6.fpadd2s)하면 2.4%정도로 기존 코어 대비 증가하게 되는데, 이와 같은 면적 증가는 칩 전체적으로 보았을 때 그리 크지 않은 수치라 판단된다.

그림 4는 표 1의 값으로 hotfloorplan을 이용하여 설계한 최적화된 코어 배치이다. 그림 4의 (a)는 Alpha 21264의 Ev6 코어 배치도를 보여준다. 그림 4(b)는 보조 부동소수점 가산기(FPAdd2)를 추가한 Ev6.fpadd2 코어 배치도를 나타낸다. 이 배치도에서 적색 영역이 보조 부동소수점 가산기(FPAdd2)의 위치이며, 데이터 패스를 고려하여 부동소수점 레지스터(FPReg)와 인접하게 배치한다. 그림 4의 (c)는 Ev6.fpadd2 코어 배치에 열 전달 지연 공간(SHTD1~4)를 추가한 Ev6.fpadd2s 코어 배치도이다. 이 배치도에서 청색 영역이 열 전달 지연 공간의 위치이며, 각각 부동소수점 레지스터(FPReg) 및 정수 레지스터(IntReg)와 인접한 기능 유닛 사이에 배치한다.

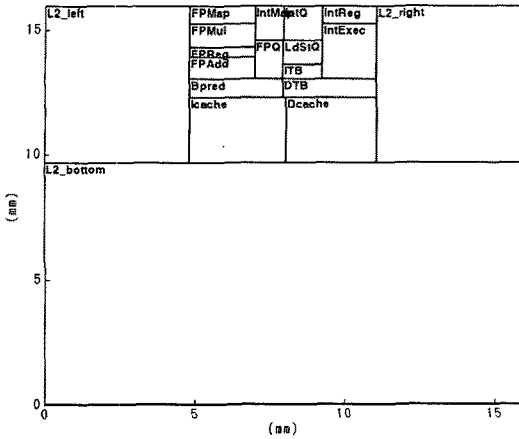
4.2 유닛의 배치에 따른 열섬 현상 분석

그림 5는 부동소수점 벤치마크인 mgrid를 수행하는 경우에 대해 각 기능 유닛의 최고 온도 분포를 보여준다. 제안하는 기법이 적용되지 않았을 경우의 결과인 그림 5의 (a)를 보면, 부동소수점 가산기(FPAdd)의 온도가 가장 높은 것을 볼 수 있다. 듀얼 부동소수점 가산기 구조를 적용한 그림 5의 (b)에서는 두 개의 부동소수점 가산기 온도가 그림 5의 (a)의 부동소수점 가산기 온도보다 낮아진 것을 확인할 수 있다. 하지만 두 개의 부동소수점 가산기에서의 발열로 인한 열 전달 현상으로 부동소수점 레지스터(FPReg)의 온도가 올라간 것을 볼 수 있다. 열 전달 지연 공간을 추가로 배치한 구조인 그림 5의 (c)를 살펴보면, 부동소수점 레지스터(FPReg)와 정수 레지스터(IntReg)에 대해 서로 인접한 기능 유닛 사이에 열 전달 지연 공간(SHTD)을 배치한 결과, 모든 레지스터의 온도가 낮아진 것을 확인할 수 있다. 특이한 사항으로 제안 기법을 적용한 결과 목표로 하였던 부동소수점 레지스터뿐만 아니라 정수 레지스터 역시 온도가 낮아진 것을 볼 수 있다. 대신 보조 부동소수점 가산기가 추가되면서 데이터 읽기 접근 비율이 증가하여, 데이터 캐시(Dcache)와 적재/저장 큐(LdStQ) 기능 유닛의 온도는 증가한다. 하지만 전체 기능 유닛의 최고 온

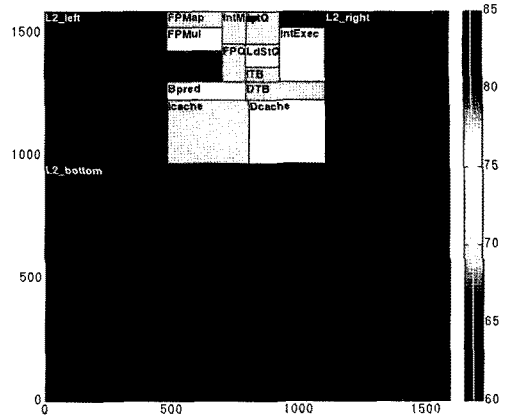
표 1 AlphaEV6(21264) 유닛 면적 명세($\times 10^{-6} \text{ m}^2$)

배치도	유닛	면적	유닛	면적
Ev6	FPAdd	1.940	FPQ	1.445
	FPReg	0.835	LdStQ	1.252
	FPMul	2.076	ITB	0.782
	FPMMap	1.445	Bpred	2.269
	IntMap	1.283	DTB	2.295
	IntQ	1.758	Icache	8.346
	IntReg	1.226	Dcache	7.824
	IntExec	3.985	L2	214.317
Ev6.fpadd2	FPAdd2 (추가)	1.940		
Ev6.fpadd2s	SHTD1/2 (추가)	1.670	SHTD3/4 (추가)	2.452
배치도	총 면적	증가 면적*	증가 비율 (%)*	
Ev6	253.078	-	-	
Ev6.fpadd2	255.018	1.940	0.77%	
Ev6.fpadd2s	259.140	6.062	2.40%	

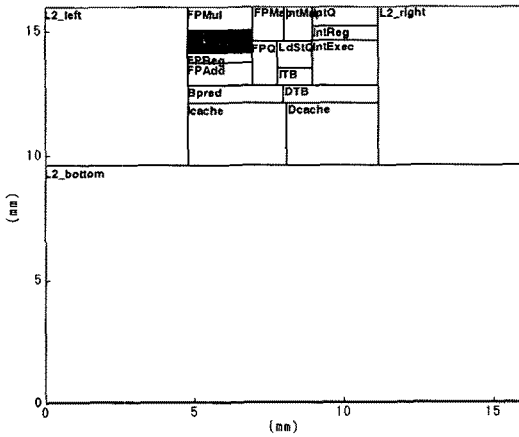
* 증가 면적과 증가 비율은 Ev6 코어 배치가 기준임.



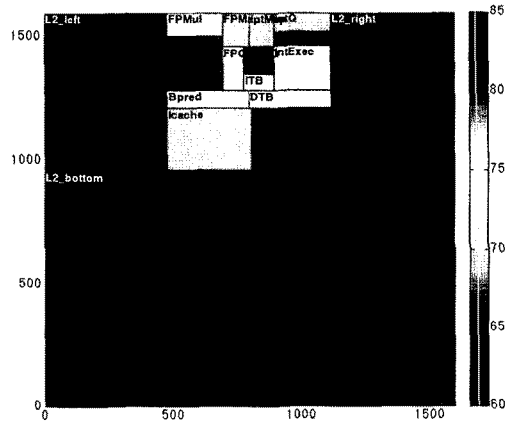
(a) Ev6 core floorplan



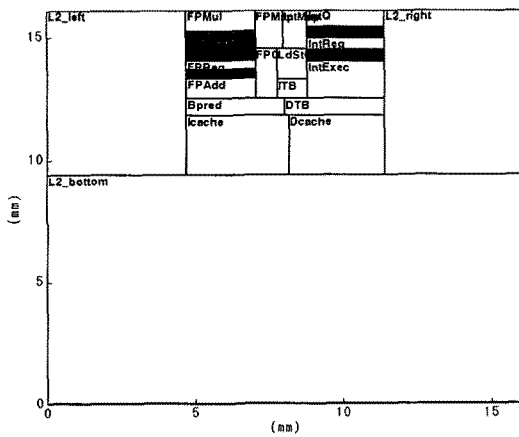
(a) Ev6 core floorplan



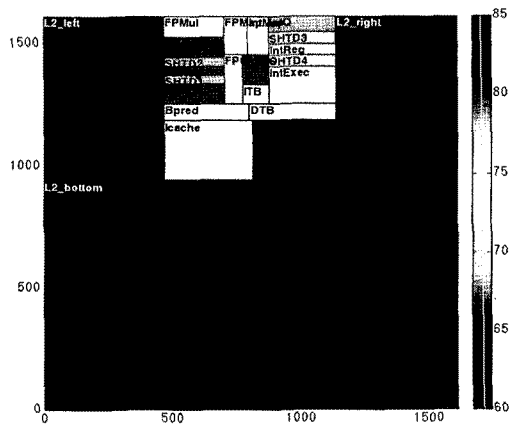
(b) Ev6.fpadd2 core floorplan



(b) Ev6.fpadd2 core floorplan



(c) Ev6.fpadd2s core floorplan



(c) Ev6.fpadd2s core floorplan

그림 4 모의 실험에 사용한 마이크로프로세서 유닛 배치도

그림 5 mgrid 벤치마크 모의 실험 결과 각 코어 배치에 따른 기능 유닛의 최고 온도 분포

도보다는 낮기 때문에 발열 현상으로 인한 칩의 안정성에는 큰 영향을 미치지 않는다. 따라서 열 전달 지연 공

간을 배치하는 경우 유닛의 최고 온도가 낮아지면서 칩의 안정성을 높일 수 있음을 알 수 있다.

4.3 최고 온도 분석을 통한 발열 안정성 평가

동적 온도 제어를 위한 긴급 단계의 온도는 85°C와 90°C로 각각 설정하는 경우에 대한 최고 온도 분포는 그림 6과 같다. 실험 결과, 본 논문에서 제안하는 기법인 Ev6.fpadd2와 Ev6.fpadd2s가 기존의 Ev6보다 유닛의 최고 온도가 하강한 것을 확인할 수 있다. 듀얼 부동소수점 가산기를 적용하는 경우 대부분의 벤치마크에서 유닛의 최고 온도가 하강하지만, 그림 6의 (a)에서 mgrid(Ev6.fpadd2)와 (b)에서 galgel(Ev6.fpadd2s)에서는 유닛의 최고 온도가 오히려 상승한다. 그림 6의 (a)에서 mgrid(Ev6.fpadd2) 벤치마크의 경우 부동소수점 가산기들의 최고 온도는 하강하지만, 열 전달 현상으로 인해 부동 소수점 레지스터의 온도가 상승하면서 유닛의 최고 온도 역시 상승하는 현상이 발생한다. 하지만 mgrid 또한 Ev6.fpadd2s 배치를 적용하면 유닛의 최고 온도가 5.2°C 하강한다

표 2는 그림 6에서 Ev6 배치를 기준으로 하여 Ev6.fpadd2와 Ev6.fpadd2s 배치를 대한 온도차이 결과를 정리해서 보여준다. 표에서 나타내는 바와 같이 긴급 단계 온도가 90°C로 설정할 때, Ev6.fpadd2 배치를 적용하면 최대 6.7°C평균 3.0°C 온도가 하강하고, Ev6. fpadd2s 배치를 적용하면 최대 10.8°C평균 5.3°C 온도가 하강한다.

모의 실험 결과를 보면 극히 일부 부동소수점 벤치마크를 제외하고는 듀얼 부동소수점 가산기 구조(Ev6.fpadd2)

표 2 긴급 단계 온도에 따른 유닛의 최고 온도 결과(°C)

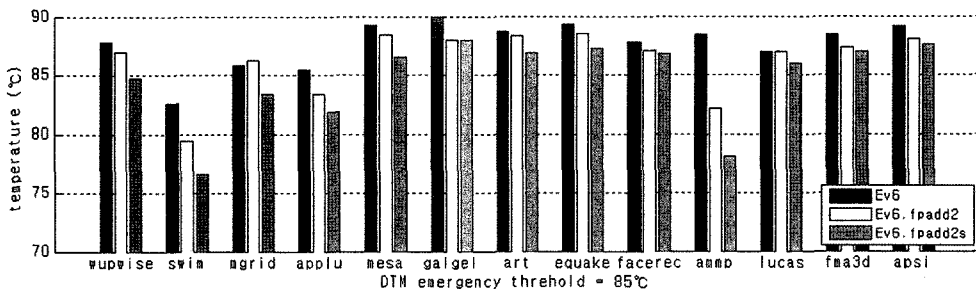
긴급 단계 온도	Ev6.fpadd2		Ev6.fpadd2s	
	최대 온도차	평균 온도차	최대 온도차	평균 온도차
85	- 6.2	- 1.5	- 10.4	- 3.0
90	- 6.7	- 3.0	- 10.8	- 5.3

* 유닛의 최고 온도는 Ev6 코어 배치가 기준임.

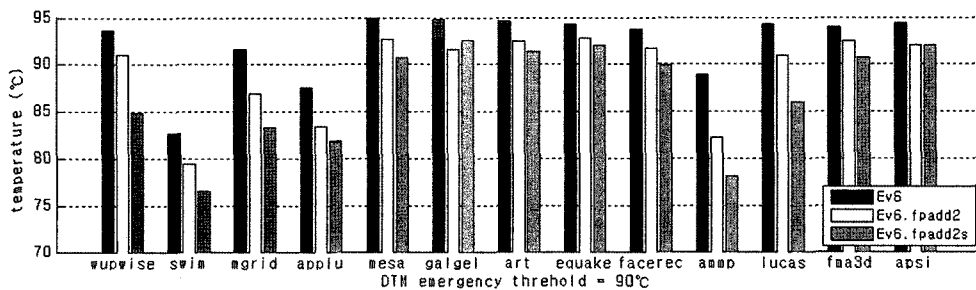
와 열 전달 지연 공간(Ev6.fpadd2s)를 함께 적용하면 부동소수점 응용 프로그램에서 마이크로프로세서 내부의 최고 온도가 가장 효과적으로 낮아진다는 것을 확인할 수 있다. 이를 통해 본 논문에서 제안한 동적 온도 제어 방법과 정적 온도 제어 방법을 함께 사용한다면, 칩의 발열 안정성에 영향을 미치는 최고 온도를 낮추는데 매우 효과적이라는 것을 확인할 수 있다.

4.4 동적 온도 제어에 따른 마이크로프로세서 성능 분석

부동소수점 응용 프로그램의 실행 시간 중에 동적 온도 관리의 수행 시간이 짧을수록 프로그램의 실행 시간 역시 짧아진다. 즉, 동적 온도 관리의 개입 시간과 마이크로프로세서의 성능은 반비례한다. 동적 온도 관리 기법의 적용이 성능에 미치는 영향을 상세하게 분석하기 위하여, 모의 실험을 통해 프로그램의 수행 시간을 비교 하였다. 그림 7은 긴급 단계 온도에 따른 부동소수점 벤치마크의 실행 시간을 보여준다. 그림 7에서 'C1'은 Ev6, 'C2'는 Ev6.fpadd2, 'C3'는 Ev6.fpadd2s 배치 구

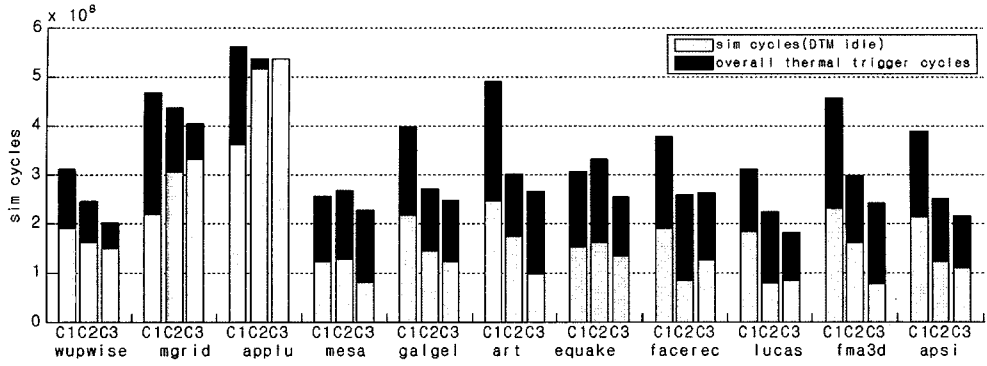


(a) 긴급 단계 온도가 85°C일 때 최고온도

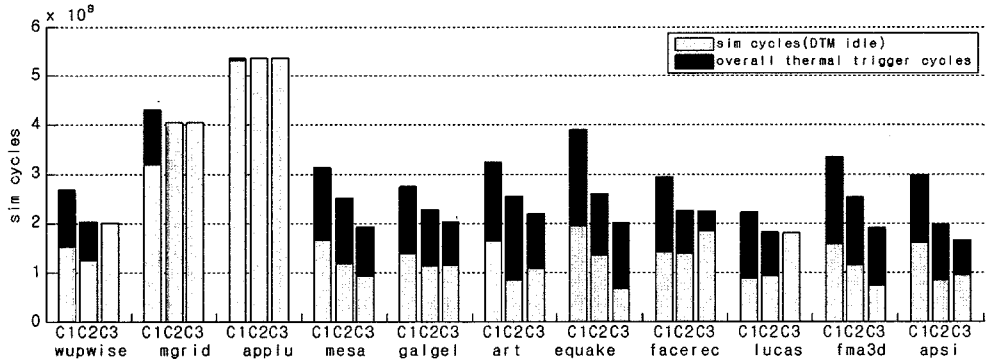


(b) 긴급 단계 온도가 90°C일 때 최고온도

그림 6 동적 온도 관리 기능의 설정에 따른 기능 유닛 중에서 최고 온도



(a) 긴급 단계 온도가 85°C일 때 실행 시간



(b) 긴급 단계 온도가 90°C일 때 실행 시간

그림 7 동적 온도 관리 기능의 설정에 따른 부동소수점 벤치마크 실행 시간

조를 의미한다. 그래프의 각 바는 프로그램 수행에 소요되는 사이클을 의미하고, 녹색은 동적 온도 관리 기능이 동작하지 않는 시간을, 갈색은 동적 온도 제어에 의한 구동 단계 시간을 의미한다.

그림 7에서는 전체 부동소수점 벤치마크 중에서 swim과 ammp는 제외한다. 이 두 개의 부동소수점 벤치마크 프로그램 실행 시에는 전체 기능 유닛의 최고 온도가 설정한 구동 단계의 온도보다 낮아서 동적 온도 관리 기능이 전혀 동작하지 않기 때문이다. 그림 7의 (a)를 보면 Ev6.fpadd2와 Ev6.fpadd2s에서는 동적 온도 제어의 수행 시간이 줄어든다는 것을 확인할 수 있다. 특히 applu 벤치마크 프로그램의 경우에는 Ev6.fpadd2s에서는 동적 온도 제어를 전혀 수행하지 않는다는 것을 볼 수 있다. 그림 7의 (b)를 보면 Ev6.fpadd2에서는 mgird, applu, Ev6.fpadd2s에서는 wupwise, lucas에 대해 추가로 동적 온도 제어를 사용하지 않는 것을 살펴볼 수 있다.

표 3은 Ev6 배치를 기준으로 Ev6.fpadd2와 Ev6.fpadd2s에 대한 성능 결과를 정리하여 나타낸다. 표 3에서 긴급 단계 온도가 85°C일 때, Ev6.fpadd2 배치를

표 3 긴급 단계 온도에 따른 실행 시간 비교

긴급 단계 온도 (°C)	Ev6.fpadd2		Ev6.fpadd2s	
	최대 성능	평균 성능	최대 성능	평균 성능
85	+ 1.62	+ 1.25	+ 1.90	+ 1.41
90	+ 1.51	+ 1.23	+ 1.96	+ 1.38

* 모의 실험 실행 시간은 Ev6 코어 배치가 기준임.

적용하면 성능이 최대 1.62배(평균 1.25배) 상승하고, Ev6.fpadd2s 배치를 적용하면 성능이 최대 1.90배(평균 1.41배) 향상된 것을 볼 수 있다. 다만, Ev6.fpadd2 배치일 때 mesa와 earthquake 벤치마크 프로그램에 대해서는 성능이 각각 1.04배와 1.08배 정도 저하되었다.

지금까지의 모의 실험 결과를 종합하면, 듀얼 부동소수점 가산기 구조(Ev6.fpadd2)를 적용한 경우 일부 벤치마크 프로그램을 수행 시 마이크로프로세서의 최고 온도가 상승하거나, 성능이 조금 저하된 경우가 발생하지만 그 폭은 미미하다고 할 수 있다. 그리고 최고 온도가 상승한 벤치마크 프로그램은 성능이 향상되고, 성능이 저하된 벤치마크 프로그램은 최고 온도가 하강하기 때문에 결코 나쁜 결과라고만 볼 수는 없다. 열 전달 지연 공간 배치 구조(Ev6.fpadd2s)를 사용하면 최고 온도

가 상승하거나 성능이 저하되는 문제가 전혀 발생하지 않는다. 따라서 부동소수점 가산기 구조와 열 전달 지연 공간 배치 구조가 부동소수점 응용 프로그램 수행 시 동적 온도 제어를 사용하는 경우에는 발열 안정성과 성능 모두를 향상시킬 수 있을 것으로 기대된다.

5. 결론 및 향후 과제

본 논문에서는 부동소수점 응용 프로그램 수행 시 마이크로프로세서의 성능 저하를 줄이면서 발열에 대한 안정성을 높이는 방법으로 듀얼 부동소수점 가산기 구조와 열 전달 지연 공간을 이용한 기능 유닛 배치 방법을 제안하였다. 기존에는 저온도 마이크로프로세서 설계를 위해 정수 레지스터의 열섬 현상을 해결하기 위한 연구가 대부분이었으나, 이러한 해결 방법은 부동소수점 응용 프로그램을 수행하는 경우에는 큰 효과를 볼 수 없었다. 본 논문에서는 듀얼 부동소수점 가산기 구조를 이용하여 부동소수점 응용 프로그램을 수행하는 경우 마이크로프로세서의 열섬 현상을 성능 저하 없이 완화하였으며, 동적 온도 관리에 의한 성능 감소를 최소화하였다. 또한 열 전달 지연 공간을 레지스터 주변에 추가로 배치하여 인접한 기능 유닛의 열 전달로 인해 발생하는 온도 상승 문제를 완화하였다. 그 결과 마이크로프로세서의 최고 온도는 더욱 낮아졌으며, 이를 통해 동적 온도 제어의 개입으로 인한 성능 저하를 크게 완화하였다. 따라서 제안하는 기법들을 적용하면 부동소수점 응용 프로그램을 주로 수행하는 고성능 DSP나 멀티미디어 시스템에서 동작하는 마이크로프로세서의 안정성과 성능 향상에 도움을 줄 수 있을 것으로 기대된다.

본 논문에서 제안한 방법은 비록 작은 수치이지만 칩의 면적이 0.77%에서 2.40%까지 증가하는 단점이 있다. 따라서 동적 온도 관리에 따른 성능 저하가 다소 증가하더라도 칩의 면적이 증가하지 않는 방법에 대한 연구가 추후 필요하다. 또한 열 전달 지연 공간의 배치는 칩의 면적을 증가시키는 커다란 원인이기에 가장 효율적인 공간 배치 방법에 대한 추가 연구가 필요하다.

참고 문헌

- [1] R. Mahajan, "Thermal management of CPUs: A perspective on trends, needs and opportunities," in *Proceeding of the 8th International Workshop on THERMal INvestigations of ICs and Systems*, 2002.
- [2] V. Narayanan and Y. Xie, "Reliability concerns in embedded system designs," *COMPUTER*, vol.39, pp.118-120, 2006.
- [3] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proceedings of the 27th International Symposium on Computer Architecture*, pp.83-94, 2000.
- [4] J.S. Choi, J.H. Kong, E.Y. Chung and S.W. Chung, "A Dual Integer Register File Structure for Temperature-Aware Microprocessor," *Journal of KIISE: Computer System and Theory*, vol.35, no.12, pp.540-551, 2008. (in Korean)
- [5] I. Yeo and E. J. Kim, "Hybrid dynamic thermal management based on statistical characteristics of multimedia applications," in *Proceeding of the thirteenth international symposium on Low power electronics and design (ISLPED)*, pp.321-326, 2008.
- [6] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A case for thermal-aware floor-planning at the microarchitectural level," *Journal of Instruction-Level Parallelism*, vol.7, pp.1-16, 2005.
- [7] S. Chung and K. Skadron, "Using On-Chip Event Counters For High-Resolution, Real-Time Temperature Measurement," *Thermal and Thermomechanical Phenomena in Electronics Systems*, pp. 114-120, 2006.
- [8] H. Seongmoo, K. Barr, and K. Asanovic, "Reducing power density through activity migration," in *Proceeding of the 2003 International Symposium on Low Power Electronics and Design*, pp.217-222, 2003.
- [9] K. Skadron, M. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol.1, pp.94-125, 2004.
- [10] K. Patel, W. Lee, and M. Pedram, "Active bank switching for temperature control of the register file in a microprocessor," in *Proceedings of the 17th great lakes symposium on Great lakes symposium on VLSI*, 2007, pp.231-234.
- [11] SIA, "Technology Roadmap for Semiconductors," 2005.
- [12] SPEC(Standard Performance Evaluation Corporation) CPU2000, <http://www.spec.org/cpu2000/>
- [13] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *Computer Architecture, 2000. Proceedings of the 27th International Symposium on 2000*, pp.83-94.
- [14] Hotspot Tool Set v3.1, <http://lava.cs.virginia.edu/HotSpot/>



이 병 석

1997년 조선대학교 전자계산학과 이학사
 1999년 조선대학교 대학원 전자계산학과
 이학석사. 2000년~현재 조선대학교 대
 학원 컴퓨터학과 박사과정. 관심분야는
 저온도 프로세서, 임베디드 시스템, SoC
 설계, 연산기 설계



김 철 홍

1998년 서울대학교 컴퓨터공학과 공학사
 2000년 서울대학교 대학원 컴퓨터공학부
 공학석사. 2006년 서울대학교 대학원 전
 기컴퓨터공학부 공학박사. 2005년~2007
 년 삼성전자 반도체총괄 SYSTEM.LSI
 사업부 책임연구원. 2007년~현재 전남
 대학교 전자컴퓨터공학부 교수. 관심분야는 임베디드 시스
 템, 컴퓨터구조, SoC 설계, 저전력 설계



이 정 아

1982년 서울대학교 컴퓨터공학과 공학사
 1985년 인디애나 주립대학교 컴퓨터학과
 석사. 1990년 캘리포니아 주립대학교
 (UCLA) 컴퓨터공학과 공학박사. 1990
 년~1995년 Assistant Professor, Uni-
 versity of Houston, U.S.A. 1995년~현
 재 조선대학교 컴퓨터공학과 교수. 관심분야는 임베디드 시
 스템, 컴퓨터 구조, SoC 설계, 고속 연산 알고리즘 및 연산
 기 구조