The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 14, No. 3, pp.163-169, Jun. 30, 2014. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2014.14.3.163
JIIBC 2014-3-23

비대칭적 멀티코어 프로세서의 성능 연구

Performance Study of Asymmetric Multicore Processor Architectures

이종복*

Jongbok Lee*

요 약 현재 범용 컴퓨터 시스템을 구축할 때 성능을 높이기 위하여 멀티코어 프로세서가 널리 이용되고 있으며, 멀티코어 프로세서의 구조는 크게 대칭적 구조와 비대칭적 구조로 나뉜다. 비대칭적 멀티코어 프로세서는 크고 복잡한 고성능의 코어와, 작고 간단한 저성능의 프로세서들로 구성되며, 대칭적 멀티코어 프로세서에 비하여 더욱 성능과 효율이 높은 것으로 알려져 있다. 본 논문에서는 다양한 구성을 갖는 비대칭적 쿼드코어 및 옥타코어 프로세서에 대하여 SPEC 2000 벤치마크를 통하여 모의실험을 수행하여 그 성능을 측정하고, 대칭적 쿼드코어 및 옥타코어 프로세서와 그 성능을 비교하였다.

Abstract Recently, the importance of multicore processor system is growing rapidly. Multicore processors are classified either as symmetric or asymmetric. Asymmetric multicore processors consist of a high performance complex core and number of low performance simple cores, and are known to be more efficient than symmetric multicore processors. Therefore, performance impact on various configurations of asymmetric multi-core processor needs to be studied. Using SPEC 2000 benchmarks as input, the trace-driven simulation has been performed for different asymmetric quad-core and octa-core processors and compared to the corresponding symmetric ones.

Key Words : asymmetric multicore processor, quadcore, octacore.

Ⅰ. 서 론

현재 멀티코어 프로세서가 스마트폰, 태블릿 PC, 노트북, 데스크탑 등과 같은 컴퓨터 시스템의 성능 향상을 높이기 위하여 광범위하게 쓰이고 있다[1-5]. 그 중에서 비대칭적 (asymmetric) 멀티코어 프로세서는 동일한 명령어 집합을 가지면서 서로 다른 구조, 복잡도, 성능 및 전력소비를 갖는 이종 (heterogeneous)의 코어들로 구성된다[6,7]. 전형적인 비대칭적 멀티코어 프로세서는 크고 복잡한 고성능의 코어와, 작고 간단한 저성능의 코어들로 구성된다. 이러한 비대칭적 멀티코어 프로세서는 동종 (homogeneous)의 코어들로 구성되는 대칭적 멀티코어 프로세서에 비하여 더욱 성능과 효율이 높은 것으로 알려져 있다. 본 논문에서는 다양한 사양의 비대칭적 쿼드코어(quad-core) 및 옥타코어(octa-core) 프로세서에 대하여, SPEC 2000 벤치마크를 통하여 모의실험을 수행하여 그 성능을 측정하였으며, 하드웨어 비용이 비슷한 대칭적 멀티코어 프로세서와 그 성능을 비교하였다.

본 논문은 다음과 같이 구성된다. 2 장에서 비대칭적 멀티코어 신호처리 프로세서에 대하여 살펴보고, 3 장에서

*정회원, 한성대학교 정보통신공학과
접수일자 2014년 4월 30일, 수정완료 2014년 5월 28일
게재확정일자 2014년 6월 13일

Received: 30 April, 2014 / Revised: 28 May, 2014
Accepted: 13 June, 2014
*Corresponding Author: jblee@hansung.ac.kr
Dept. of ICs Engineering, Hansung University, Korea

모의실험기 및 모의실험 환경에 대하여 고찰한다. 4 장에서 모의실험 결과를 보이며, 5 장에서 결론을 맺는다.

II. 비대칭적 멀티코어 프로세서 시스템

1. 비대칭적 멀티코어 프로세서의 구조

그림 1은 N 개의 코어로 구성되는 비대칭적 멀티코어 프로세서의 일반적인 구조를 나타낸 것이다. 이 때, 한 개의 코어는 여러 개의 긴 쓰레드 (thread)를 실행시킬 수 있는 고성능 비순차 (out-of-order) 수퍼스칼라 프로세서의 형태를 가지며, 2부터 N까지 대칭적 멀티코어 프로세서로 구성되는 나머지 N-1 개의 코어들은, 단순한 구조의 RISC 또는 짧은 길이의 쓰레드를 실행시키는 순차 (in-order) 수퍼스칼라 프로세서들로 구성된다.

한편, 각 코어는 자체적으로 1 차 명령어 캐쉬와 1 차 데이터 캐쉬를 가지며, 또한 메인 메모리와 연결되는 공통의 2 차 통합 캐쉬를 공유한다. 각 코어에 설치된 1 차 데이터 캐쉬의 일관성(cache-coherency)을 위하여 MESI 프로토콜을 이용하여, 어느 코어에서 공유된 캐쉬 메모리에 쓰기 작업을 하였을 때, 나머지 코어에서는 해당 데이터를 무효화 (write-invalidate) 시킨다. 이하 본 논문에서 기술하는 명령어 캐쉬와 데이터 캐쉬는 모두 1 차 캐쉬를 의미한다.

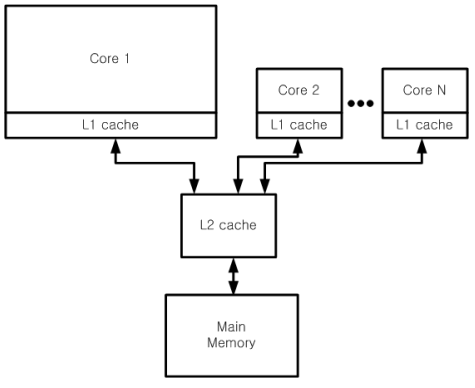

그림 1. 비대칭적 멀티코어 프로세서의 구조
Fig. 1. The asymmetric multicore processor architecture

2. 전역제어부에 의한 쓰레드의 할당

전역제어부 (global control unit)는 그림 2에 나타낸 것과 같이, 비대칭적 멀티코어 프로세서 시스템에서 각 코어에 동적으로 쓰레드를 예측하여 할당하고 실행이 완료되었을 때는 할당을 취소하고, 다음 쓰레드를 예측하는 기능을 담당한다 [8,9].

쓰레드는 RISC형 코어일 때 명령어 1 개로, 수퍼스칼라형 코어일 때는 명령어 N 개로 구성된다. 그리고, 코어당 쓰레드의 개수는 구조에 따라 한 개 또는 두 개 이상일 수도 있다. 전역제어부가 쓰레드를 코어에 할당할 때 하는 일은 프로그램 카운터의 값을 알려주는 것으로서, 명령어의 해독(decoding) 및 실행은 각 코어에서 처리된다.

전역제어부는 다음의 쓰레드를 동적으로 예측하기 위하여 2 단계 방법을 이용한다. 이것은 분기 예측 방법과 유사한데, 제 1 단계에서 최근 k 개의 쓰레드의 향방을 추적하고, 제 2 단계에서 카운터를 이용하여 각 쓰레드의 타겟이 선정된 회수를 기록하는 것이다[10].

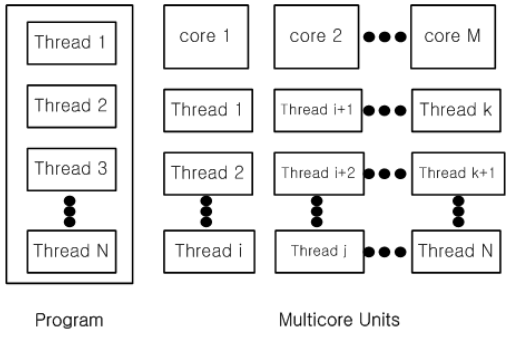

그림 2. 멀티코어 프로세서에서 전역제어부의 쓰레드 할당
Fig. 2. The thread allocation by global control unit in multicore processor

3. 비대칭적 멀티코어 프로세서를 구성하는 캐쉬

멀티코어 프로세서에서 명령어 캐쉬 미스에 의한 성능의 손실을 최소화하기 위하여, 캐쉬의 구조를 2 차 이상의 연관도 (2-way set associative) 및 충분한 용량으로 구성하여 충분한 캐쉬 히트율을 확보할 수 있다. 데이터 캐쉬의 경우, 여러 코어 간의 캐쉬 일관성(Cache Coherency)을 위한 MESI 프로토콜의 적용으로 인하여 캐쉬의 데이터를 무효화 (write-invalidate)하는 경우가 빈번히 발생한다. 따라서, 직접 캐쉬로는 적정한 캐쉬 히트율을 확보할 수가 없어서 멀티코어 프로세서의 성능 손실이 크므로, 데이터 캐쉬 역시 2 차 이상의 연관도로 구성하였다.

Ⅲ. 모의실험 환경

1. 비대칭적 멀티코어 프로세서 모의실험기

본 논문에서는 명령어 자취형 모의실험기를 개발하여 모의실험에 이용하였다 [11]. 비대칭적 멀티코어 프로세서는 제 1 단계 명령어 자취의 발생, 제 2 단계 명령어 자취에 대한 비대칭적 멀티코어 프로세서의 실행으로 나누어진다. 제 1 단계에서 명령어 자취는 Simplescalar를 이용하여 SPEC 벤치마크 프로그램으로부터 임의의 차수의 멀티코어에 적합하도록 발생되었다 [12].

제 2 단계에서 각 코어가 단순한 RISC 프로세서, 순차 수퍼스칼라 또는 비순차 수퍼스칼라 프로세서로 동작하는 비대칭적 멀티코어 프로세서에 입력된다. 이 때, 전역 제어부에 의하여 쓰레드 수준 병렬성이 쓰레드로 매핑되어 각 코어에서 실행된다. 제 2 단계의 과정을 자세하게 기술하면 그림 3에 나타낸 것과 같다.

(1) 명령어 인출, 재명명 및 이슈

초기화 작업을 거친 후에, 각 코어는 매 싸이클마다 1 개 또는 N 개의 명령어를 인출 받는다. 인출된 명령어는 재명명 (renaming) 작업을 거치면서 명령어 종속에 의한 타임스탬프(timestamp) 값을 설정 받는다. 타임스탬프 방식은 명령어 자취를 이용하는 모의실험에서 데이터 종속성을 신속하고 효율적으로 부여할 수 있는 방법이다.

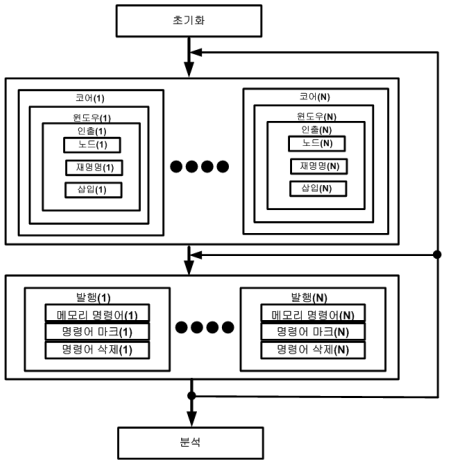

그림 3. 비대칭적 멀티코어 프로세서 모의실험기의 흐름도
Fig. 3. The flow chart of asymmetric multicore processor simulator

레지스터 화일의 타임스탬프 값에 의하여, 멀티코어 프로세서 명령어 간의 종속성이 유지되어 성능을 구하는데 반영된다.

한편, 재명명을 거친 명령어는 각 코어의 쓰레드에 삽입된다. 싸이클이 증가함에 따라서 쓰레드 내의 명령어는 자체의 타임스탬프 값이 현재 싸이클보다 작거나 같고, 해당 연산유닛이 활용 가능할 때 삭제될 수 있다.

(2) 멀티코어 시뮬레이션

N 개의 비대칭 멀티코어에 대하여 해당 코어의 윈도우 공간에 동종형 및 이종형 여부에 따라서 적절하게 한 개 또는 M 개의 명령어를 인출해서 채우고, 각 코어에 대하여 명령어를 실행하면서 종속성에 의하여 부여된 명령어의 타임스탬프가 충족되면 삭제한다. 이 과정은 코어 내부 및 코어 간의 레지스터 종속 및 메모리 종속 검사에 적용되며, 입력으로 주어진 벤치마크 프로그램의 모든 명령어가 소진될 때까지 반복된다.

위 과정이 한번 실행될 때 마다 싸이클이 증가하므로, 매 싸이클 당 명령어의 실행 및 삭제가 가장 오래 걸리는 코어가 해당 싸이클 수를 결정한다. 모의실험에 입력으로 쓰인 명령어의 총 개수를 처리하기 위하여 소요된 총 싸이클 수로 나누어, 비대칭적 멀티코어 프로세서 시스템의 성능의 척도인 IPC(Instruction Per Cycle)를 계산할 수 있다.

2. 비대칭적 멀티코어 프로세서 및 벤치마크의 사양

(1) 멀티코어 프로세서의 사양

표 1은 모의실험에 이용된 비대칭적 멀티코어 프로세서 아키텍처의 사양을 나타낸 것이다.

멀티코어의 개수는 쿼드코어에서 4 개, 옥타코어에서 8 개를 대상으로 하였다. 각 코어는 RISC 방식 또는 수퍼스칼라 방식으로 운영되므로, 동종형 및 이종형 여부에 따라 적절하게 매 싸이클마다 1 개에서 N 개의 명령어를 인출, 이슈, 실행 및 종료한다. 각 코어의 연산유닛은 정수형 유닛, 로드 스토어 유닛, 분기 유닛, 실수형 덧셈기 및 실수형 곱셈기로 구성된다. 명령어 캐쉬와 데이터 캐쉬는 각 코어마다 설치되는데, 64KB의 용량을 갖도록 설정하였으며, 2 차 연관도 (2-way set associativity) 방식을 통하여 접근된다. 그러나, 모든 코어에 의하여 공유되는 2 차 캐쉬는 충분한 용량으로 인하여 100 % 히트가 난다고 가정하였다.

표 1. 모의실험에 이용된 비대칭적 멀티코어 프로세서 아키텍쳐 하드웨어의 사양
Table 1. The architecture specification of asymmetric multicore processor hardware

항목	값	
멀티코어 수	4, 8	
구조	RISC	수퍼스칼라
쓰레드의 개수	1	1 또는 2
쓰레드의 크기	1	4/8/16
인출율, 이슈율, 퇴거율	1	2/4/8/16
명령어 캐쉬 및 데이터 캐쉬의 공통 사항	64 KB, 2 차 연관, 16 B 미스 페날티 10 싸이클	
연산유닛 개수	산술논리(1/2/4/8), 분기(1), 로드(1/2), 스토어(1),	
쓰레드 어드레스 캐쉬	2 K 엔트리	
쓰레드 예측기	2 단계 14 비트 전역 히스토리 방식 미스 페날티 6 싸이클	
이슈 지연 싸이클	산술논리(1), 분기(1), 로드(1), 스토어(1),	
결과 지연 싸이클	산술논리(1), 분기(1), 로드(1), 스토어(1),	

표 2는 본 모의실험에서 사용하는 대칭적 및 비대칭적 쿼드코어 및 옥타코어 프로세서의 구성을 나타낸 것이다. 본 논문에서 코어의 복잡도는 코어가 처리할 수 있는 쓰레드의 최대 개수, 쓰레드당 최대 명령어의 개수 및 명령어에 대한 순차 수행 또는 비순차 수행의 여부로 결정된다. 쿼드코어일 때 대칭적 멀티코어 프로세서는 쓰레드의 길이가 2, 4, 8, 16이고 쓰레드의 개수가 1이며 순차 실행되는 간단한 수퍼스칼라 코어 네 가지 유형으로 설정하였으며, 각각 I_2, I_4, I_8, I_{16}로 표기하였다.

표 2. 대칭적 멀티코어 프로세서와 대응하는 비대칭적 멀티코어 프로세서의 세부 사양
Table 2. The specific architectures of symmetric multicore processors and corresponding asymmetric ones

코어수	대칭적	비대칭적	
쿼드 코어	$4I_2$	O_4	3R
	$4I_4$	O_8	$3I_2$
	$4I_8$	O_{16}	$3I_4$
	$4I_{16}$	O_{32}	$3I_8$
옥타 코어	$8I_2$	O_8	7R
	$8I_4$	O_{16}	$7I_2$
	$8I_8$	O_{32}	$7I_4$
	$8I_{16}$	O_{64}	$7I_8$

한편, 비대칭적 멀티코어 프로세서에서 쓰레드의 개수가 2이고 처리할 수 있는 쓰레드의 길이가 4, 8, 16, 32, 64인 비순차 실행되는 수퍼스칼라 코어를 각각 O_4, O_8, O_{16}, O_{32}, O_{64}로 표기하였다. 또한, 이 표에서 R은 간단한 RISC 코어를 의미한다.

첫 번째 대칭적 쿼드코어 프로세서는 I_2 네 개의 프로세서로 구성되며, 대응하는 비대칭적 쿼드코어 프로세서의 사양은 O_4 한 개와 R 세 개의 조합이다. 두 번째 대칭적 쿼드코어 프로세서는 네 개의 I_4로 구성되며, O_8 한 개와 I_2 세 개의 비대칭적 쿼드코어 프로세서와 비교된다. 세 번째 대칭적 쿼드코어 프로세서는 네 개의 I_8로 구성되며, 대응되는 비대칭적 쿼드코어 프로세서는 한 개의 O_{16}과 세 개의 I_4이다. 마지막으로, 네 번째 대칭적 쿼드코어 프로세서는 I_{16} 네 개로 동작한다. 여기에 대응하는 비대칭적 쿼드코어 프로세서의 사양은 O_{32} 한 개와 I_8 세 개이다.

옥타코어 프로세서일 때 첫 번째 사양은 I_2 여덟 개로 구성되는 대칭적 구조이며, 이에 비교하는 비대칭적 옥타코어 프로세서는 O_8 한 개와 R 일곱 개이다. 두 번째로 여덟 개의 I_4로 구성되는 대칭적 옥타코어 프로세서는 한 개의 O_{16}과 일곱 개의 I_2로 구성되는 비대칭적 옥타코어 프로세서에 대응된다. 세 번째, 여덟 개의 I_8로 이루어진 대칭적 옥타코어 프로세서는 한 개의 O_{32}와 일곱 개의 I_4로 이루어진 비대칭적 옥타코어 프로세서에 대응된다. 마찬가지로, 네 번째 사양의 대칭적 옥타코어 프로세서는 I_{16} 여덟 개로 구성되며, O_{64} 한 개와 I_8 일곱 개로 이루어진 비대칭적 옥타코어 프로세서와 비교할 것이다. 모든 경우에 대하여, 공정한 비교를 위하여 비대칭적 멀티코어의 하드웨어 비용은 대칭적 멀티코어보다 작거나 같다.

(2) 벤치마크의 사양

표 3은 모의실험에 이용된 다섯 개의 SPEC 2000 벤치마크 프로그램이다. Simplescalar를 통하여 MIPS IV 10억 개의 명령어 자취를 임의의 차수의 비대칭적 멀티코어 프로세서에 적합하도록 발생시켜서 모의실험기에 입력하였다.

표 3. SPEC 2000 정수형 벤치마크 프로그램
Table 3. SPEC 2000 integer benchmark programs

벤치마크	설 명
bzip2	압축
crafty	체스 경기 놀이
gzip	압축
paser	워드 프로세서
twolf	배선 및 배치 모의실험기

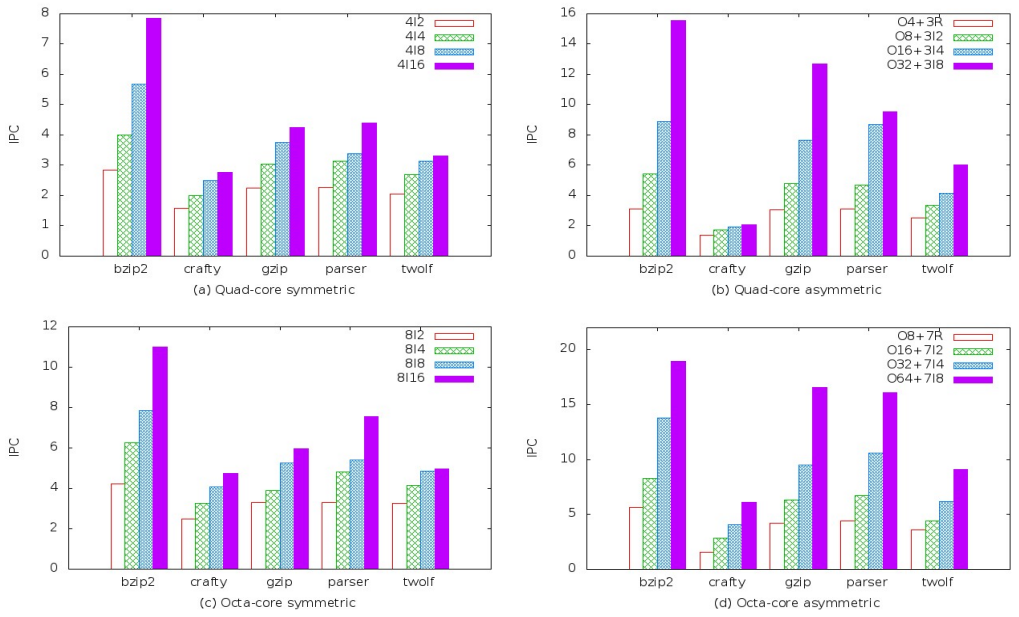

그림 4. 대칭적 멀티코어 프로세서와 비대칭적 멀티코어 프로세서의 성능 비교
Fig. 4. The performance comparison between symmetric and asymmetric multicore processors

Ⅳ. 모의실험 결과

그림 4에 쿼드코어와 옥타코어 프로세서일 때, 대칭적 멀티코어 프로세서와 이에 대응하는 비대칭적 멀티코어 프로세서의 성능을 각각 나타냈다. 모의실험 결과를 분석하면 다음과 같다.

1. 쿼드코어 프로세서의 모의실험 결과

그림 4(a)와 4(b)는 각각 대칭적 쿼드코어 프로세서와 비대칭적 쿼드코어 프로세서의 모의실험 결과이다. I_2 네 개의 대칭적 쿼드코어 프로세서는 평균 2.16 IPC를 나타냈으며, 대응하는 I_4 한 개와 R 세 개의 조합의 비대칭적 쿼드 코어 프로세서는 그보다 높은 2.53 IPC를 기록하여 성능이 17 % 개선되었다. 한편, I_4 네 개의 대칭적 쿼드코어 프로세서는 평균 2.89 IPC를 나타냈으며, O_8 한 개와 I_2 세 개의 조합으로 구성되는 비대칭적 쿼드코어 프로세서는 역시 그보다 높은 3.72 IPC를 기록하였다. 또한, I_8 네 개의 대칭적 쿼드코어 프로세서의 사양은 3.55 IPC를 얻은 반면에, 대응하는 O_{16} 한 개와 I_4 세 개로 구성되는 비대칭적 쿼드코어 프로세서는 5.52 IPC를 기록하였다. 마지막으로, I_{16} 네 개의 대칭적 쿼드코어 프로세서는 4.22 IPC 밖에 얻지 못한 반면에, O_{32} 한 개와 I_8 세 개의 비대칭적 쿼드코어 프로세서는 훨씬 높은 7.38 IPC를 기록하여, 성능이 74 % 개선되었다. 쿼드코어의 네 가지 사양을 평균하면, 비대칭 구조가 대칭 구조보다 44 % 높은 성능을 기록하였다.

2. 옥타코어 프로세서의 모의실험 결과

그림 4(c)와 4(d)는 옥타코어 프로세서일 때의 모의실험 결과를 비교하여 나타낸 것이다. I_2 여덟 개로 구성되는 대칭적 옥타코어 프로세서의 성능은 평균 3.13 IPC를 기록하였으나, O_8 한 개와 R 일곱 개의 조합인 비대칭적 옥타코어 프로세서는 보다 높은 평균 3.67 IPC를 기록하였다. 또한 여덟 개의 I_4로 구성된 대칭적 옥타코어 프로세서는 4.30 IPC에 그친 반면, 대응하는 O_{16} 한 개와 I_2 일곱 개로 구성되는 비대칭적 옥타코어 프로세서는 5.86 IPC를 획득하였다. 한편, 여덟 개의 I_8로 구성되는 대칭적 옥타코어 프로세서는 4.94 IPC를 나타냈고, 한 개의 O_{32}와 일곱 개의 I_4로 구성되는 비대칭 옥타코어 프로세서는 39 % 성능이 개선된 6.85 IPC를 기록하였다. 마지막으로,

여덟 개의 I_8로 구성되는 대칭적 옥타코어 프로세서는 평균 6.19 IPC를 얻은 반면에, 한 개의 O_{64}와 일곱 개의 I_8의 조합인 비대칭적 옥타코어 프로세서는 80 %의 성능이 개선된 11.12 IPC를 나타냈다. 비대칭적 옥타코어 프로세서의 대칭적 옥타코어 프로세서에 대한 평균 성능의 개선은 43 %를 기록하여, 쿼드코어의 경우와 거의 같은 결과를 나타냈다.

이상에서 볼 때, 같은 하드웨어 비용과 코어의 개수로 설계하는 경우, 동종의 대칭적 멀티코어 프로세서로 구축하는 것 보다는, 한 개의 크고 성능이 높은 코어와 작고 간단한 여러 개의 코어로 구성되는 이종의 비대칭적 멀티코어 프로세서로 구현하는 것이 더욱 높은 성능을 얻을 수 있다는 것을 알 수 있다.

V. 결 론

본 논문에서는 RISC, 순차 및 비순차 수퍼스칼라와 같은 이종형 프로세서의 다양한 조합으로 구성되는 비대칭적 쿼드코어 및 옥타코어 프로세서 아키텍처에 대하여 SPEC 벤치마크를 입력으로 하여 모의실험을 통하여 성능을 측정하고 그 결과를 분석하였다. 비슷한 하드웨어 비용의 동종형 프로세서로 구성되는 대칭적 멀티코어 프로세서와 그 성능을 비교한 결과, 비대칭적 멀티코어 프로세서가 평균 44 % 높은 성능을 나타냈다.

따라서, 옥타코어 프로세서 이하이면서 동일한 개수의 코어로 비슷한 하드웨어 비용을 투자하였을 때, 동종의 대칭적 멀티코어 프로세서로 구현하는 것보다는 한 개의 크고 성능이 높은 코어 한 개와 작고 간단한 여러 개의 코어로 구성되는 이종의 비대칭적 멀티코어 프로세서가 더욱 높은 성능을 얻을 수 있다는 것을 알 수 있다.

추후로, 통계적 모의실험을 기반으로 하는 멀티코어 프로세서에 대한 연구를 진행할 계획이다. 통계적 모의실험에서는, 입력으로 이용되는 벤치마크 프로그램의 명령어자취에 대한 사전 프로화일링을 통하여 특성을 잘 나타낼 수 있는 새로운 명령어 자취를 생성한다. 생성된 새로운 명령어 자취는 데이터 크기가 작고 모의실험 시간을 대폭 단축할 수 있으면서 원래의 명령어 자취의 특성을 압축적으로 잘 나타내기 때문에, 모의실험에 대한 기억장치의 용량과 소요시간을 대폭 줄일 수 있다는 장점이 있다.

References

[1] J. B. Lee, “A Study of Trace-driven Simulation for Multicore Processor Architectures,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 12, No. 3, pp. 9-13, Jun. 2012.

[2] J. B. Lee, “Performance Anaylsis of Multicore Processor Architectures Based On Cache Size Effects,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 12, No. 6, pp. 175-180, Dec. 2012.

[3] J. B. Lee, “Performance Study of Multi-core In-Order Superscalar Processor Architectures,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 12, No. 5, pp. 123-128, Oct. 2012.

[4] J. B. Lee, “A Performance Study of Embedded Multicore Processor Architectures,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 13, No. 1, pp. 163-169, Feb. 2013.

[5] J. B. Lee, “Performance Study of Multicore Digital Signal Processor Architectures ,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 13, No. 14, pp. 171-177, Aug. 2013.

[6] R. Kumar et al, “Single-ISA heterogeneous Multicore Architectures for Heterogeneous for Multithreaded Workload Performance,” Annual International Symposium on Computer Architecture, Mar. 2004.

[7] Hourd, Jon, et al. "Exploring Practical Benefits of Asymmetric Multicore Processors." Workshop on Parallel Execution of Sequential Programs on Multi-core Architectures, Apr. 2009.

[8] T. Ungerer, B. Robic, and J. Silk, "Multithreaded Processors," The Computer Journal, Vol. 45, No. 3, 2002

[9] G. S. Sohi, S. E. Breach, and T. N. Vijaykumar, "Multiscalar Processors," Proceedings of the 22nd

annual international symposium on Computer architecture, pp. 414-425, May 1995.

[10] T-Y. Yeh and Y. N. Patt, "Alternative Implementations of Two-Level Adaptive Branch Prediction," in Proceedings of the 19th International Symposium on Computer Architecture, pp.124-134, May. 1992.

[11] A. Rico, A. Duran. F. Cabarcas, Y. Etsion, A. Ramirex, and M. Valero, "Trace-driven Simulation of Multithreaded Applications," ISPASS, 2011.

[12] T. Austin, E. Larson, and D. Ernest, "SimpleScalar : An Infrastructure for Computer System Modeling," Computer, vol. 35, no. 2, pp. 59-67, Feb. 2002.

저자 소개

이 종 복(정회원)

- 1964년 8월 20일생.
- 1988년 : 서울대 컴퓨터공학과 졸업.
- 1998년 동 대학 전기공학부 졸업 (공박).
- 1998~2000 : LG반도체 선임연구원.
- 2000년~현재 : 한성대 정보통신공학과 교수

- Tel : 02-760-4497
- Fax : 02-760-4435
- E-mail : jblee@hansung.ac.kr

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.