

<https://doi.org/10.7236/IIBC.2017.17.2.251>

IIBC 2017-2-36

멀티코어 프로세서의 전력 소비에 대한 연구

A Study on Power Dissipation of The Multicore Processor

이종복*

Jongbok Lee*

요약 최근에 이르러, 범용 컴퓨터 뿐만이 아니라 임베디드 시스템 및 모바일 장치에서도 광범위하게 멀티코어 프로세서가 이용되어 그 성능이 증대되고 있다. 이러한 멀티코어 프로세서 시스템의 전력 소비량이 매우 중요하므로, 설계의 초기 단계에서 그 값을 정확하게 예측할 수 있어야 한다. 본 논문에서는 멀티코어 프로세서에 대하여 빠른 속도를 갖는 명령어 자취형 (trace-driven) 모의실험기 기반의 전력 분석기를 개발하였다. 이 때, 각 코어를 구성하는 하드웨어 유닛별 소비전력을 계산하여 합산하였다. 또한, SPEC 2000 벤치마크를 입력으로 모의실험을 수행하여 명령어 당 평균 전력 소비량을 측정하였다.

Abstract Recently, multicore processor system is widely adopted not only in general purpose computers but also in embedded systems and mobile devices in order to improve performance. Since the power dissipation issue of multicore processor system is very significant, it must be estimated accurately in the early design stage. In this paper, a fast power analysis tool for a high performance multicore processor based on the trace-driven simulator has been developed. To achieve it, the power dissipation of each hardware unit per core are added. Using SPEC 2000 benchmarks as input, the trace-driven simulation has been performed to estimate the average power dissipation per instruction.

Key Words : multicore processor, power dissipation, instruction trace

1. 서 론

현재 멀티코어 프로세서가 스마트폰, 태블릿 PC, 노트북, 데스크탑, 대형 서버 등과 같은 컴퓨터 시스템의 성능 향상을 높이기 위하여 널리 쓰이고 있다. 그동안 전력 소비량은 임베디드 컴퓨터 또는 모바일 컴퓨터 시스템에서 중요하게 다루어왔으나, 최근에 이르러 고성능 마이크로 프로세서나 멀티코어 프로세서에서의 설계에서도 주요 제약요소로 대두되고 있다. 반도체 다이의 크기가 작아 질수록 프로세서의 클럭의 주파수가 더욱 증가하므로 전력 소비량이 큰 비중으로 고려되어야 한다. 멀티코어 프

로세서의 성능과 전력을 측정하기 위한 모의실험에는 실행 위주(execution-driven) 방식과 명령어 자취(trace-driven) 방식이 널리 쓰이고 있다. SimpleScalar와 같은 실행 위주 방식은 정확하지만 시간이 많이 걸린다는 단점이 있다^[1].

본 논문에서는 멀티코어 프로세서의 전력 소비량을 빠르고 효율적으로 측정할 수 있는 명령어 자취를 기반으로 하는 전력 모델 및 모의실험기를 제안하였다. 또한, 본 모의실험기를 이용하여, SPEC 2000 벤치마크에 대하여 평균 전력의 소비량을 측정하고, 멀티코어 프로세서의 각 하드웨어 유닛에서 소비되는 전력을 분석하였다.

*정회원, 한성대학교 전자정보공학과
접수일자 : 2016년 12월 30일, 수정완료 : 2017년 3월 7일
게재확정일자 : 2017년 4월 7일

Received: 30 December, 2016 / Revised: 7 March, 2017 /
Accepted: 7 April, 2017

*Corresponding Author : jblee@hansung.ac.kr
Dept. of ICs Engineering, Hansung University, Korea

본 논문은 다음과 같이 구성된다. 2 장에서 멀티코어 프로세서의 전력 소비량을 측정할 수 있는 모델과 멀티코어 프로세서 모의실험기에 대하여 살펴보고, 3 장에서 모의실험 환경에 대하여 고찰한다. 4 장에서 모의실험 결과를 보이며, 5 장에서 결론을 맺는다.

II. 멀티코어 프로세서의 전력 모델 및 모의실험기

1. 멀티코어 프로세서의 구조 및 전력 모델

그림 1은 M 개의 코어로 구성되는 멀티코어 수퍼스칼라 프로세서 구조를 나타낸 것이다. 각 코어는 자체적으로 1 차 명령어 및 데이터 캐쉬를 가지며, 메인 메모리와 연결되는 공통의 2 차 캐쉬를 공유한다. 각 코어에 설치된 1 차 데이터 캐쉬의 일관성(cache-coherency)을 위하여 MESI 프로토콜을 이용한다.

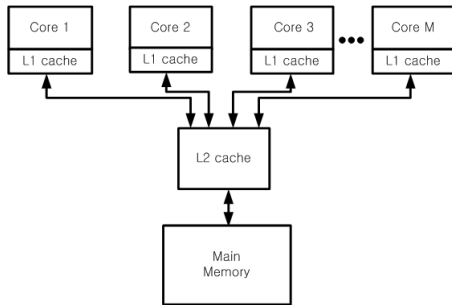


그림 1. 멀티코어 프로세서의 구조
Fig. 1. The multi-core processor architecture

CMOS 멀티코어 프로세서의 전력 소비량 중에서, 가장 큰 비중을 차지하는 동적인 전력 소비량을 수식 1과 같이 나타낼 수 있다.

$$P_d = \alpha CV_{dd}^2 f$$

수식 1. 멀티코어 프로세서의 전력 소비량
Eqn. 1. Power dissipation of microprocessors

이 때 α 는 하드웨어 유닛에 대한 클럭에 의한 평균 활성화율이며, 0과 1 사이의 값이다. C 는 정전용량, V_{dd} 는 공급되는 전원, f 는 동작 주파수이다. 전력 소비량은 프로세서의 설계 및 구현 방법과 프로세서를 구성하는 내부

정전용량에 의하여 크게 영향을 받는다. 지금까지 연구된 전력소비량을 계산하는 가장 일반적인 방법은 멀티코어 프로세서를 구성하는 코어를 여러 단계로 나누어 각 단계마다 RC 회로를 모델링하는 것이다. 이렇게 함으로써 각 단계의 지연시간을 측정할 수 있으나, 임계경로를 위주로 측정하기 때문에 전력 소비량을 측정하는데 한계가 있다^[3]. 이러한 단점을 극복하기 위하여, 본 논문에서는 각 단계에서의 정전용량을 고려하고 모든 경로에서의 전력 소비량을 분석하고 합산하는 방법을 택하였다. 이때, 멀티코어 프로세서를 구성하는 트랜지스터의 크기는 다양한 하드웨어 구조의 정전용량에 큰 영향을 끼친다^[2,3].

표 1. 단위 멀티코어 프로세서 하드웨어 유닛과 대응하는 전력 모델
Table 1. Unit multicore processor hardware units and associated power model

카테고리 유형	하드웨어 유닛
복합 조합회로	명령어 선택 논리장치, 레지스터 재명명 종속검사 장치, 정수형 연산 유닛, 실수형 연산 유닛
배열구조	명령어 캐쉬, 데이터 캐쉬, 레지스터 화일, 분기예측기
CAM 구조	명령어 기동 논리장치
배열 및 CAM 구조	명령어 윈도우, TLB, 로드/스토어 큐

본 논문에서 멀티코어 프로세서에 대한 전력 소비량의 측정은, 클럭을 제외하고 표 1에 나타낸 것과 같이 크게 네 가지 유형의 하드웨어 구조로 분류하였다. 첫 번째는 복합적 조합회로로서, 연산유닛, 명령어 윈도우 선택 논리회로, 종속 검사 논리회로, 결과버스 등을 예로 들 수 있다. 두 번째는 배열 구조로서, 명령어 캐쉬, 데이터 캐쉬, 캐쉬 태그 배열, 레지스터 화일, 분기 예측기, 로드 스토어 큐 등이 이에 해당된다. 세 번째는 완전 연관형 CAM (Content Addressable Memory) 구조로서, 명령어 기동 논리장치 (instruction wakeup logic)가 이에 속한다. 네 번째는 배열구조와 CAM 구조의 특징을 동시에 갖는 회로로서, 명령어 윈도우, TLB, 로드/스토어 큐 등이 여기에 속한다.

(1) 복합 조합회로의 전력 소비량

복합 조합회로는 이전 상태를 기억할 필요가 없으며,

규칙적이거나 반복적인 구조를 갖지 않고 임의의 논리를 구현하는데 쓰인다. 복합 조합회로의 대표적인 사례는 정수형 연산유닛, 실수형 연산유닛, 명령어 윈도우의 명령어 선택 논리장치와 레지스터 재명명 장치의 종속 검사 논리장치를 들 수가 있다^[24,56].

(2) 배열구조의 전력 소비량

배열구조는 규칙적이고 반복적인 구조를 갖는 하드웨어 유닛인 명령어 캐쉬, 데이터 캐쉬, 레지스터 화일 및 분기예측기의 전력을 측정하는데 쓰인다. 이 때, 행의 수, 열의 수, 읽기 및 쓰기 포트 수를 기준으로 하여 디코더의 크기와 개수, 워드라인 (word line) 및 비트라인 (bit line)의 개수가 결정된다. 따라서, 배열구조의 전력 소비는 디코더, 워드라인 구동기, 비트라인 방전으로 결정된다.

워드라인과 비트라인의 전력 소비량 측정은 각 라인의 총 정전용량을 계산하여 얻을 수 있다. 워드라인과 관련된 정전용량은 워드라인 구동기의 확산 정전용량 (diffusion capacitance), 셀 접근 트랜지스터의 게이트 정전용량 (gate capacitance), 워드라인 금속선의 정전용량 (metal wire capacitance)의 세 가지로 구성된다. 한편, 비트라인의 정전용량은 선충전 트랜지스터의 확산 정전용량, 셀 접근 트랜지스터의 확산 정전용량, 비트라인 금속선의 정전용량으로부터 계산에 의하여 구할 수 있다. 본 논문에서 전력을 계산하기 위한 캐쉬 구조의 물리적인 특성은 Cacti 도구를 참조하였다^[3].

(3) CAM 구조의 전력 소비량

CAM 구조는 배열구조와 유사하지만, 내용으로 메모리를 어드레싱할 수 있다는 점이 다르다. 이 구조에서는 배열구조의 비트라인과 워드라인이 각각 태그라인과 매치라인에 대응된다는 점만 제외하고 배열구조와 같다. CAM 구조에 속하는 하드웨어 장치는 명령어 기동 논리 장치가 대표적이다. 이 때, 명령어의 이슈 및 완료 폭은 CAM 내부의 각 코어셀의 매치 및 태그 라인 수를 결정하며, 윈도우의 크기는 물리적 레지스터의 크기를 결정한다.

마지막으로, 배열 및 CAM 구조는 규칙적인 구조를 가지면서 동시에 저장된 데이터의 내용으로 장치를 접근해야 하는 경우이다. 명령어 윈도우, TLB, 로드/스토어큐가 이에 해당된다. 이상, 본 논문의 전력모델은 위의 네 가지 카테고리에 대하여, Wattch 도구와 Cacti 도구를 참

조하여 C 언어로 완성되었다^[37,8].

2. 멀티코어 프로세서 모의실험기

본 논문에서는 명령어 트레이스 모의실험기를 개발하여 모의실험에 이용하였다^[11]. 멀티코어 프로세서는 제 1 단계 명령어 자취의 발생, 제 2 단계 명령어 자취에 대한 멀티코어 프로세서의 실행으로 나누어진다. 제 1 단계에서 명령어 자취는 SimpleScalar를 이용하여 표 2에 나타난 SPEC 벤치마크 프로그램으로부터 임의의 차수의 멀티코어에 적합하도록 발생되었다^[9,10].

제 2 단계에서 각 코어가 순차 수퍼스칼라로 동작하는 멀티코어 프로세서에 입력된다. 이 때, 쓰레드 수준 병렬성이 쓰레드로 매핑되어 각 코어에서 실행된다. 제 2 단계의 과정을 자세하게 기술하면 그림 2에 나타난 것과 같다.

(1) 명령어 인출, 재명명 및 이슈

초기화 작업을 거친 후에, 각 코어는 매 사이클마다 1 개 또는 N 개의 명령어를 인출받는다. 인출한 명령어는 재명명 (renaming) 작업을 거치면서 명령어 종속에 의한 타임스탬프(timestamp) 값을 설정받는다. 타임스탬프 방식은 명령어 자취를 이용하는 모의실험에서 데이터 종속성을 신속하고 효율적으로 부여할 수 있는 방법이다. 레지스터 화일의 타임스탬프 값에 의하여, 멀티코어 프로세서 명령어 간의 종속성이 유지되어 성능을 구하는데 반영된다.

한편, 재명명을 거친 명령어는 각 코어의 쓰레드에 삽입된다. 사이클이 증가함에 따라서 쓰레드 내의 명령어는 자체의 타임스탬프 값이 현재 사이클 보다 작거나 같을 때 삭제될 수 있다.

(2) 멀티코어 시뮬레이션

M 개의 멀티코어에 대하여 해당 코어의 윈도우 공간에 적절하게 1 개 또는 N 개의 명령어를 인출해서 채우고, 역시 M 개의 멀티코어에 대하여 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.

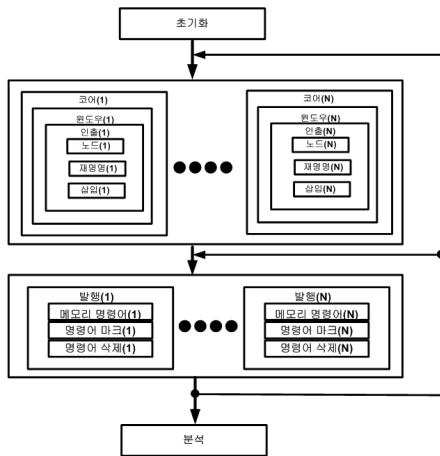


그림 2. 멀티코어 프로세서 모의실험기의 흐름도
 Fig. 2. The flow chart of digital signal multicore processor simulator

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

표 2. SPEC 2000 정수형 벤치마크 프로그램
 Table 2. SPEC 2000 integer benchmark programs

벤치마크	설 명
bzip2	압축
crafty	체스 경기 놀이
gap	그룹 이론 해석기
gcc	C 프로그래밍 언어 컴파일러
gzip	압축
mcf	조합 최적화
paser	워드 프로세서
twolf	배선 및 배치 모의실험기

(3) 전력 모델과의 인터페이스

전력 모델은 위에서 기술한 멀티코어 프로세서 모의실험기에서 매 사이클마다 각 코어에서 접근되는 각 하드웨어 유닛을 추적하여 실행을 위하여 소비되는 전력을 계산한다. 이것은, 명령어 캐쉬를 접근할 때의 전력, 명령어 윈도우에서 이슈 가능한 명령어를 명령어들을 선택할 때의 전력, 레지스터 화일에서 데이터를 읽을 때의 전력, 산술 논리연산 유닛에서 계산을 수행할 때의 전력을 계산하여 이루어진다. 따라서 각 하드웨어 유닛의 소비전력을 합산하여 총 전력을 구할 수 있으며, 이 값을 실행된 총 명령

어의 수로 나누면, 명령어 당 전력소비량을 구할 수 있다.

III. 모의실험 환경

표 3은 모의실험에 이용된 프로세서 아키텍처의 사양을 나타낸 것이다.

각 코어는 슈퍼스칼라 방식으로 운영되므로, 매 사이클 마다 4 개의 명령어를 인출하고 윈도우의 크기는 16이다. 각 코어의 연산유닛은 정수형 유닛, 로드 스토어 유닛, 그리고 분기명령어로 구성된다. 1 차 명령어 캐쉬와 데이터 캐쉬는 각 코어마다 설치되며, 코어의 개수가 증가할 수록, 그 용량이 증가하므로 성능에 유리하다. 따라서 공정한 성능 비교를 위하여, 단일코어 프로세서의 명령어 캐쉬와 데이터 캐쉬는 동일하게 64 KB의 용량을 갖도록 설정하였다. 1 차 명령어 캐쉬는 2 차 연관도(set associativity)를 가지나, 1 차 데이터 캐쉬는 직접 매핑을 통하여 접근된다. 2 차 캐쉬는 100 % 히트가 난다고 가정하였으며, 분기 명령어는 2 단계 적응형 분기 예측 방식을 적용하였다.

본 논문의 모의실험은 운영체제 Fedora 24에서 3.1GHz로 동작하는 Intel Core i5-2400에서 시행하였다. 멀티코어 프로세서는 600MHz에서 동작하는 0.35µm CMOS 기술을 가정하였다. 모의실험에 이용된 SPEC 2000 정수형 벤치마크 프로그램은 SimpleScalar를 통하여 MIPS IV 10억 개의 명령어 자취를 임의의 차수의에 적합하도록 발생시켜서 모의실험기에 입력하였다.

표 3. 멀티코어 프로세서 아키텍처 하드웨어의 사양
 Table 3. The architecture specification of multicore processor

항목	값
코어의 개수	2, 4
1 차 명령어 캐쉬	64KB
1 차 데이터 캐쉬	64KB
1차 명령어 캐쉬 공통 사항	2 차 연관, 16 B 미스 페널티 10 사이클
1차 데이터 캐쉬 공통 사항	직접, 32 B 미스 페널티 10 사이클
명령어 윈도우의 크기	16
인출율, 이슈율, 퇴거율	4
연산유닛 사양	ALU(4), 로드 및 스토어(2), 분기(1)
분기 어드레스 캐쉬	1 K 엔트리
분기 예측기	8 비트 전역 히스토리 방식 미스 페널티 6 사이클
이슈 지연 사이클	산술논리(1), 분기(1), 로드(1), 스토어(1),
결과 지연 사이클	산술논리(1), 분기(1), 로드(1), 스토어(1),

IV. 모의실험 및 결과

그림 3에 듀얼코어 프로세서와 쿼드코어 프로세서를 구성하는 각 연산유닛 및 캐쉬 장치에서 싸이클 당 전력 소비량을 나타냈다. 이 때 싸이클 당 전력 소비는 크게 재명명 장치, 분기예측 장치, 명령어 윈도우 장치, 로드 및 스토어큐 장치, 레지스터 파일, 명령어 캐쉬, 데이터 캐쉬, 산술연산장치, 결과 버스에서 이루어진다. 모의실험 결과에 의하면 재명명 장치, 분기예측 장치, 로드 및 스토어 장치에서의 전력 소비는 상대적으로 10 W 이하로 미미하고, 명령어 윈도우 장치, 레지스터 파일, 명령어 캐쉬, 데이터 캐쉬, 산술연산 장치에서 50 W 이상의 대부분의 전력이 소비된다는 것을 알 수 있다.

그림 4에 듀얼코어 프로세서와 쿼드코어 프로세서에서 8 개의 SPEC 벤치마크를 입력으로하여 모의실험을 수행하여 얻은 평균 전력 소비량의 결과를 보였다. 모의 실험 결과에서 알 수 있듯이, 각 벤치마크 프로그램 별로 최저 30.8 W에서 최고 50.9 W 범위의 평균 전력을 기록하였다.

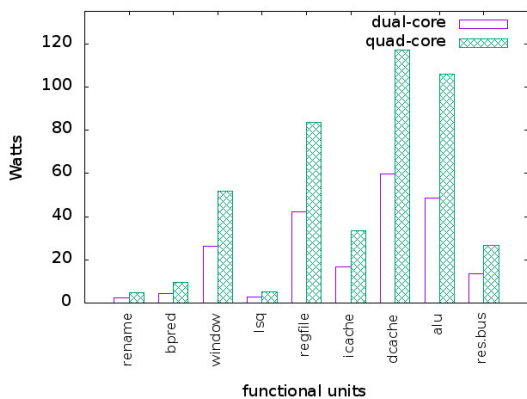


그림 3. 하드웨어 장치별 전력 소비량
 Fig. 3. Power dissipation of each hardware unit

듀얼코어 프로세서와 쿼드코어 프로세서에서 공통적으로 *bzip2*의 경우, 데이터 캐쉬에서의 전력 소비량이 많아서 가장 높은 전력 소비량을 기록하였으며, *twolf*의 경우 명령어 캐쉬와 데이터 캐쉬에서의 전력 소비량이 적어서 가장 낮은 전력을 소비하였다. 결국 멀티코어 프로세서의 전력을 가장 많이 소비하는 장치가 캐쉬임을 알 수 있다. 듀얼코어 프로세서의 명령어당 평균 소비전력

은 36.9 W를 기록하였으며, 쿼드코어 프로세서의 경우에는 이것보다 약간 증가한 37.7 W를 나타냈다.

위에서 전력을 측정하기 위한 모의실험 결과, 초당 260K 개의 명령어를 처리할 수 있었으며, 이것은 초당 80K 개의 명령어를 처리할 수 있는 Wattch 도구보다 월등한 결과이다. 따라서, 본 논문에서 제안하는 프로세서의 모의실험에 명령어 자취형 방식을 채택함으로써, 실행위주 방식보다 3.3 배 빠르게 전력 소비량에 대한 결과를 얻을 수 있다.

V. 결론

본 논문에서는 0.35 μm CMOS를 기반으로 600 MHz에서 동작하는 멀티코어 프로세서의 전력을 명령어 자취 모의실험기를 기반으로 하여 빠르게 측정할 수 있는 전력 측정기를 개발하였다. SPEC 벤치마크에 대하여 멀티코어 프로세서에 대한 전력을 모의실험을 통하여 측정할 결과, 명령어 당 37.2 W의 평균 전력을 기록하였다. 또한, 모의실험에 의하여 전력을 측정하는 속도를 Wattch 도구와 비교한 결과 3.3 배의 향상을 가져왔다.

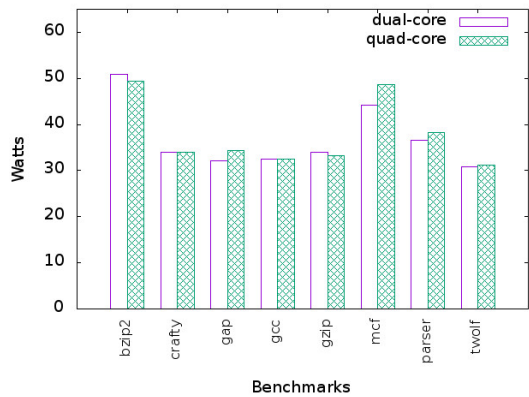


그림 4. SPEC 벤치마크의 평균 전력 소비량
 Fig. 4. The average power dissipation of SPEC benchmarks

추후로, 임베디드 멀티코어 프로세서 및 디지털 신호 처리 멀티코어 프로세서의 전력 소비량을 측정하는 연구가 필요하다. 또한, 이러한 유형의 각 멀티코어 프로세서에 대하여 동질 코어(homogeneous core)가 아닌 비동질 코어(heterogeneous core)를 채택하는 비대칭 칩 멀티코어

어 프로세서(asymmetric multicore processor) 구조에 대한 연구를 들 수 있다. 더 나아가, 최신 경향인 코어 수 64개 이상의 매니코어 (many-core) 아키텍처의 전력에 대하여 연구를 수행할 예정이다.

References

- [1] T. Austin, E. Larson, and D. Ernest, "SimpleScalar : An Infrastructure for Computer System Modeling," Computer, Vol. 35, No. 2, pp. 59-67, Feb. 2002.
- [2] S. Palacharla, N. Jouppi, and J. Smith, "Complexity-Effective Superscalar Processors", Proc. of the 24th International Symposium on Computer Architecture, 1997.
- [3] S. Wilton and N. Jouppi, "An Enhanced Access and Cycle Time Model for On-chip Caches," WRL Research Report 93/5, DEC Western Research Laboratory, 1994.
- [4] B. Bishop, T. Kelliher, and M. Irwin, "The Design of a Register Renaming Unit," Proc. of Great Lakes Symposium on VLSI, 1999.
- [5] R. Zimmermann and W. Fichtner, "Low-power logic styles : CMOS versus pass-transistor logic", IEEE Journal of Solid State Circuits, Vol. 32, No. 7, pp.1079-1090, 1997.
- [6] M. Borah, R. Owens, and M. Irwin, "Transistor sizing for low power CMOS circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems", Vol. 15, No. 6, pp. 665-671, 1996.
- [7] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," Proceedings of the 27th International Symposium on Computer Architecture, pp 83-94, Jun. 2000.
- [8] R. Joseph and M. Martonosi, "Run-time Power Estimation in High-Performance Microprocessors," The International Symposium on Low Power Electronics and Design, Aug. 2001.
- [9] J. Lee, "A Study of Trace-driven Simulation for Multi-core Processor Architectures," Journal of The Institute of Internet, Broadcasting and Communication, vol. 12, no. 3, pp. 9-13, Jun. 2012.
- [10] J. Lee, "Performance Study of Asymmetric Multicore Processor Architectures," Journal of The Institute of Internet, Broadcasting and Communication, vol. 14, no. 3, pp. 163-169, Jun. 2014.

저자 소개

이 종 복(정회원)



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

- Tel : 02-760-4497
- Fax : 02-760-4435
- E-mail : jblee@hansung.ac.kr
- <주관심분야 : 멀티코어 프로세서>

※ 본 연구는 한성대학교 교내학술연구비 지원과제임.