

<http://dx.doi.org/10.7236/JIIBC.2013.13.4.171>

JIIBC 2013-4-23

# 멀티코어 디지털 신호처리 프로세서의 성능 연구

## Performance Study of Multicore Digital Signal Processor Architectures

이종복\*

Jongbok Lee

**요약** 최근에 이르러, 고속의 3차원 그래픽 렌더링, 비디오 파일 포맷의 변환, 압축, 암호화 및 암호해독 처리를 위한 디지털 신호처리 시스템의 성능이 고도화가 요구된다. 현재 범용 컴퓨터 시스템을 구축할 때 성능을 높이기 위하여 멀티코어 프로세서가 널리 이용되고 있으므로, 디지털 신호처리 프로세서 역시 멀티코어 프로세서 구조를 채택하여 디지털 신호처리 시스템에서 높은 성능을 얻을 수가 있다. 본 논문에서는 코어의 유형 및 개수가 멀티코어 디지털 신호처리 프로세서의 성능에 미치는 영향을 분석하기 위하여, 2 개에서 16 개로 구성되는 멀티코어 디지털 신호처리 프로세서에 대하여, UTDSP 벤치마크를 입력으로 하는 모의실험을 수행하였다. 이 때, 멀티코어 디지털 신호처리 프로세서를 구성하는 단위 코어로서, 단순한 RISC형부터 다양한 명령어 윈도우의 크기를 갖는 순차 및 비순차 실행 수 퍼스칼라 코어에 걸쳐 광범위한 모의실험을 수행하여 그 성능을 분석하였다.

**Abstract** Due to the demand for high speed 3D graphic rendering, video file format conversion, compression, encryption and decryption technologies, the importance of digital signal processor system is growing rapidly. In order to satisfy the real-time constraints, high performance digital signal processor is required. Therefore, as in general purpose computer systems, digital signal processor should be designed as multicore architecture as well. Using UTDSP benchmarks as input, the trace-driven simulation has been performed and analyzed for the 2 to 16-core digital signal processor architectures with the cores from simple RISC to in-order and out-of-order superscalar processors for the various window sizes, extensively.

**Key Words** : digital signal processor, multicore processor.

### 1. 서 론

디지털 신호처리 프로세서는 음성이나 영상 신호로부터 발생하는 일련의 데이터 샘플에 대하여 고속으로 반복적인 대량의 수치연산을 고성능, 저전력으로 수행하는

전용 마이크로 프로세서이다. 특히 최근에 이르러 그래픽 처리 분야에서 CUDA와 같은 GPGPU 형태를 이용하여 고속의 3 차원 그래픽의 렌더링, 비디오 파일 포맷의 변환, 압축, 암호화 및 암호해독 기능을 위하여 디지털 신호처리 프로세서 성능의 고도화가 요구되고 있다<sup>[1]</sup>.

\*정회원, 한성대학교 정보통신공학과  
접수일자 : 2013년 6월 27일, 수정완료 : 2013년 7월 26일  
게재확정일자 : 2013년 8월 16일

Received: 27 June, 2013 / Revised: 26 July, 2013 /

Accepted: 16 August, 2013

\*Corresponding Author: jblee@hansung.ac.kr

Dept. of ICs Engineering, Hansung University, Korea

역사적으로 범용 컴퓨터시스템에 탑재되는 중앙처리 장치 (CPU)는 성능의 증대를 위하여 CISC에서 RISC 및 슈퍼스칼라로 발전하여 오늘날 멀티코어 프로세서가 대세를 이루고 있다<sup>[2-5]</sup>. 따라서 디지털 신호처리 프로세서 시스템 역시 멀티코어 구조로 발전시켜서 디지털 신호처리 시스템의 성능을 높여야 한다.

본 논문에서는 단위 코어의 구조 및 개수가 디지털 신호처리 시스템의 성능에 미치는 영향을 분석하기 위하여, 멀티코어 디지털 신호처리 프로세서에 대한 모의실험을 수행하고 분석하였다. 이것을 위하여 2-코어에서 16-코어의 멀티코어 디지털 신호처리 프로세서에 대하여, UTDSP 벤치마크를 입력으로 모의실험을 수행하여 그 성능을 측정하였다<sup>[6]</sup>. 이 때, 디지털 신호처리 프로세서 단위 코어를 간단한 RISC 형태로부터, 다양한 크기의 명령어 윈도우 (instruction window)로 구성되는 순차 (in-order) 및 비순차 (out-of-order) 슈퍼스칼라 프로세서 형태를 채택하였다.

본 논문은 다음과 같이 구성된다. 2 장에서 멀티코어 디지털 신호처리 프로세서에 대하여 살펴보고, 3 장에서 모의실험기를 포함한 모의실험 환경에 대하여 고찰한다. 4 장에서 모의실험 결과를 보이며, 5 장에서 결론을 맺는다.

## II. 멀티코어 디지털 신호처리 프로세서 시스템

### 1. 멀티코어 디지털 신호처리 프로세서의 구조

그림 1에 N 개의 코어로 구성되는 멀티코어 디지털 신호처리 프로세서의 일반적인 구조를 나타내었다. 1 개부터 N 개의 디지털 신호처리 코어는 가장 간단한 RISC 형태, 일정한 크기의 명령어 윈도우를 갖는 순차 슈퍼스칼라 또는 비순차 슈퍼스칼라 프로세서로 구성될 수 있다.

한편, 각 디지털 신호처리 코어는 지역적으로 1 차 명령어 캐쉬와 1 차 데이터 캐쉬를 가지며, 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.

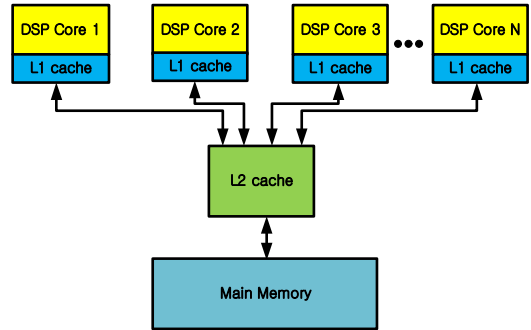


그림 1. 멀티코어 디지털 신호처리 프로세서의 구조  
Fig. 1. The multicore digital signal processor architecture

### 2. 전역제어부에 의한 태스크의 할당

전역제어부 (global control unit)는 멀티코어 디지털 신호처리 프로세서 시스템에서 각 코어에 동적으로 태스크 (task)를 할당하고 실행이 완료되었을 때는 할당을 취소하며, 다음 태스크를 예측하는 기능을 담당한다<sup>[7]</sup>. 태스크는 RISC 형일 때 명령어 1 개로, 슈퍼스칼라 형일 때는 한 개의 기본블럭에 해당하는 명령어 N 개로 구성된다. 전역제어부가 태스크를 코어에 할당할 때 하는 일은 적절한 프로그램 카운터의 값을 알려주는 것으로서, 명령어의 해독(decoding) 및 실행은 각 코어에서 처리된다.

전역제어부는 다음의 태스크를 동적으로 예측하기 위하여 2 단계 방법을 이용한다. 이것은 분기 예측 방법과 유사한데, 제 1 단계에서 최근 k 개의 태스크의 향방을 추적하고, 제 2 단계에서 카운터를 이용하여 각 태스크의 타겟이 선정된 회수를 기록해놓고, 특성에 맞추어 다음 분기 명령어의 예측에 이용하는 방법이다<sup>[8]</sup>.

### 3. 멀티코어 디지털 신호처리 프로세서를 구성하는 캐쉬

기존의 단일코어 RISC나 슈퍼스칼라 프로세서의 경우 명령어 캐쉬가 성능에 끼치는 영향력이 크므로, 전통적으로 2 차 이상의 연관도 (2-way set associative) 및 충분한 용량으로 구성하여 캐쉬 미스에 의한 성능의 손실을 최소화하였다. 멀티코어 프로세서 역시 2 차 이상의 연관도로 명령어 캐쉬를 구성해야 충분한 캐쉬 히트율을

확보할 수 있다.

한편, 전통적인 프로세서에서 데이터 캐시는 명령어 캐쉬와는 달리, 직접 캐쉬(Direct-Mapped Cache)로 구성해도 충분한 성능을 얻을 수가 있었다. 그러나, 멀티코어 프로세서 시스템에서는 여러 코어 간의 캐쉬 일관성(Cache Coherency)을 위한 MESI 프로토콜의 적용으로 인하여 캐쉬의 데이터를 무효화하는 경우가 빈번히 발생한다. 따라서, 직접 캐쉬로는 적절한 캐쉬 히트율을 확보할 수가 없어서 멀티코어 프로세서의 성능 손실이 크므로, 본 논문에서 데이터 캐쉬 역시 명령어 캐쉬와 마찬가지로 2 차 이상의 연관도로 구성하였다.

### III. 모의실험 환경

#### 1. 멀티코어 디지털 신호처리 프로세서 모의실험기

본 논문에서는 명령어 자취를 입력으로 하는 멀티코어 디지털 신호처리 아키텍처 시뮬레이터를 개발하여 모의실험에 이용하였다<sup>[9]</sup>. 멀티코어 디지털 신호처리 프로세서에서는 제 1 단계 명령어 자취의 발생, 제 2 단계 명령어 자취에 대한 멀티코어 디지털 신호처리 프로세서의 실행으로 나누어진다. 제 1 단계에서 디지털 신호처리 벤치마크 프로그램으로부터 임의의 차수의 멀티코어에 적합하도록 명령어 자취를 발생시켰다.

제 2 단계에서 각 코어는 간단한 RISC 프로세서, 순차 슈퍼스칼라 또는 비순차 슈퍼스칼라 프로세서로 구성되며, 이 멀티코어 프로세서 시스템에 제 1 단계에서 발생시킨 명령어 자취를 입력한다. 이 때, 쓰레드 수준 병렬성이 태스크로 매핑되어 각 코어에서 실행된다. 제 2 단계의 과정을 자세하게 기술하면 그림 2에 나타낸 것과 같다.

#### 가. 명령어 인출, 재명명 및 이슈

초기화 작업을 거친 후에, 각 코어는 매 사이클마다 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 핵심적인 방법이다. 레지스터 화일의 타임스탬프 값에 의하

여, 멀티코어 프로세서 명령어 간의 종속성이 정의되어 시스템의 성능을 구하는데 반영된다. 이와 같이 재명명을 거친 명령어는 각 코어의 명령어 윈도우에 삽입되며, 사이클이 증가함에 따라서 윈도우 내의 명령어는 자체의 타임스탬프 값이 현재 사이클보다 작거나 같을 때 삭제될 수 있다.

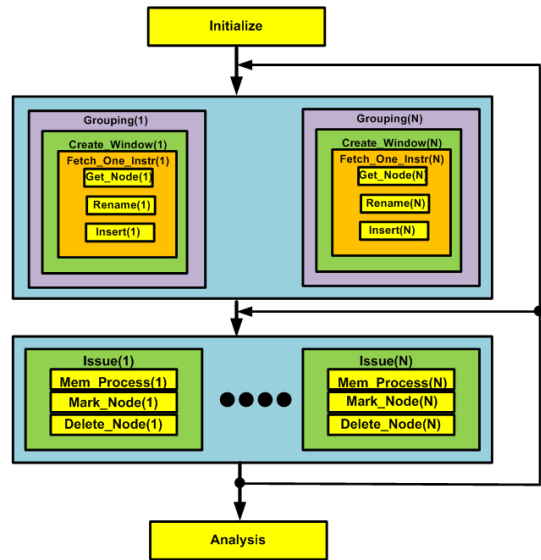


그림 2. 멀티코어 디지털 신호처리 프로세서 모의실험기의 흐름도

Fig. 2. The flow chart of multicore digital signal processor simulator

#### 나. 멀티코어 시뮬레이션

전역제어부에 의하여 N개의 멀티코어의 윈도우 공간에 한 개 또는 M 개의 명령어로 구성되는 태스크를 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, 이 과정은 입력으로 주어진 벤치마크 프로그램의 모든 명령어가 소진될 때까지 반복된다.

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 디지털 신호처리 프로세서 시스템의 성능의 척도인 IPC(Instruction Per Cycle)

를 계산할 수 있다.

## 2. 디지털 신호처리 벤치마크 및 멀티코어 디지털 신호처리 프로세서의 사양

표 1은 모의실험에 이용된 UTDSP 디지털 신호처리 벤치마크 프로그램이다. *FFT, FIR, IIR, lpc, multiplication, spectral estimation*은 실수형 덧셈, 실수형 곱셈 및 실수형 나눗셈 명령어로 구성되는 실수형 벤치마크 프로그램이며, *edge\_detect*와 *histogram*은 모두 정수형 명령어로 구성되는 정수형 벤치마크 프로그램이다.

표 1. UTDSP 벤치마크 프로그램  
Table 1. UTDSP benchmark programs

| 벤치마크                | 설 명   | 유 형 |
|---------------------|---|-----|
| edge_detect         | 256 그레이 수준 128x128 픽셀 이미지의 에지를 탐지           | 정수형 |
| FFT                 | 1024 개 점의 복소수 고속 푸리에 변환                     | 실수형 |
| FIR                 | 64 개의 점을 처리하는 256 탭 필터                      | 실수형 |
| histogram           | 히스토그램 균등화를 통한 256 그레이 수준 128x128 픽셀 이미지의 개선 | 정수형 |
| IIR                 | 64 개 점에 대한 4 개 직렬 IIR 바이쿼드 필터               | 실수형 |
| lpc                 | 선형 예측 코딩 부호화기의 구현                           | 실수형 |
| multiplication      | 두 개 10x10 행렬의 곱셈                            | 실수형 |
| spectral estimation | 피리어드그램 평균을 이용하여 음성의 입력 샘플에 대한 전력 스펙트럼 예측    | 실수형 |

위 벤치마크 프로그램에 대하여 SimpleScalar를 통하여 MIPS IV 10억 개의 명령어 자취를 임의의 차수의 멀티코어 디지털 신호처리 프로세서에 적합하도록 발생시켜서 모의실험기에 입력하였다<sup>[10]</sup>.

표 2에 모의실험에 이용된 멀티코어 프로세서 아키텍처의 사양을 나타내었다. 실험을 위한 디지털 신호처리 멀티코어 프로세서의 개수는 1 개, 2 개, 4 개, 8 개 및 16 개를 대상으로 하였다. 각 코어는 RISC 방식 또는 슈퍼스칼라 방식으로 운영되므로, 매 사이클마다 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 구성되었다.

표 2. 모의실험에 이용된 멀티코어 디지털 신호처리 프로세서 아키텍처 하드웨어의 사양

Table 2. The architecture specification of each digital signal processor core

| 항목                     | 값  |           |
|------------------------|--|-----------|
| 멀티코어 수                 | 1/2/4/8/16   |           |
| 구조                     | RISC   | 슈퍼스칼라     |
| 윈도우의 크기                | 1  | 4,8,16,32 |
| 인출율, 이슈율, 퇴거율          | 1  | 2,4,8,16  |
| 명령어 캐쉬 및 데이터 캐쉬의 공통 사항 | 64 KB, 2 차 연관, 16 B 미스 페널티 10 사이클  |           |
| 연산유닛 개수                | 산술논리(1,2,4,8), 분기(1), 로드(1,2), 스토어(1), 실수형 덧셈기(1,2,4), 실수형 곱셈기(1,2,4)    |           |
| 태스크 어드레스 캐쉬            | 2 K 엔트리  |           |
| 태스크 예측기                | 2 단계 14 비트 전역 히스토리 방식 미스 페널티 6 사이클                                       |           |
| 이슈 지연 사이클              | 산술논리(1), 분기(1), 로드(1), 스토어(1), 실수형 곱셈(1), 단정도 실수형 나눗셈(4), 배정도 실수형 나눗셈(7) |           |
| 결과 지연 사이클              | 산술논리(1), 분기(1), 로드(1), 스토어(1), 실수형 곱셈(3), 단정도 실수형 나눗셈(6), 배정도 실수형 나눗셈(9) |           |

명령어 캐쉬와 데이터 캐쉬는 각 코어마다 설치되는 데, 공통적으로 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.

## IV. 모의실험 결과

### 1. 멀티코어 RISC 디지털 신호처리 프로세서 모의실험 결과

그림 3에 RISC를 단위 코어로 활용하였을 때의 멀티코어 디지털 신호처리 프로세서의 모의실험 결과를 보였다. 각 코어에서 매 사이클마다 최대 한 개의 명령어를 인출하고 한 개의 명령어를 실행할 수 있다. 모의실험 결과에서 알 수 있듯이, 각 벤치마크 프로그램 별로 코어의 개수가 증가할수록, 성능이 증가함을 알 수 있다. 코어의 개수가 1 개에서 2 개로 증가하였을 때 평균 성능의 향상

률은 2.1 배이나, 8 개에서 16 개로 증가하였을 때는 1.5 배로 감소하였다. 코어의 개수가 증가함에 따라, 성능은 평균 1.7 배 증가하였지만 그 향상률은 둔화됨을 알 수 있다.

16-코어일 때 명령어 수준 병렬성에 의하여 순수하게 얻을 수 있는 성능에 대하여, *histogram*은 명령어 캐쉬 미스에 의하여 11.1 %의 최대 성능 손실을 가져왔으며, *FIR*은 데이터 캐쉬 미스에 의하여 9.4 %의 최대 성능 손실을 가져왔다. 전 벤치마크에 대하여 평균적으로 명령어 캐쉬 미스에 의하여 5.1 %, 데이터 캐쉬 미스에 의하여 4.3 %의 성능 손실을 가져왔으며, 태스크 예측 오류 및 태스크 캐쉬 미스에 의한 성능 손실은 각각 0.7 %와 1.3 %를 기록하였다.

1-코어일 때 평균 성능은 0.3 IPC를 기록하였으며, 16-코어일 때는 2.8 IPC를 나타냈다. 16-코어는 1-코어에 비하여 8.6 배의 성능 향상을 보였다.

## 2. 멀티코어 순차 디지털 신호처리 프로세서 모의실험 결과

그림 4(a)부터 그림 4(d)는 윈도우의 크기가 4부터 32 범위인 순차 디지털 신호처리 슈퍼스칼라 코어를 단위로 하는 멀티코어 디지털 신호처리 프로세서의 모의실험 결과를 나타낸 것이다. 코어의 개수가 2 배가 될수록 평균 1.7 배 성능이 향상하였으며, 이것은 RISC의 경우와 같은 수준이다. 한편, 슈퍼스칼라 프로세서 코어는 RISC 코어와 달리 다양한 크기의 윈도우를 가지는데, 윈도우의 크기가 2 배가 될 때마다 성능이 1.2 배 향상되었다.

각 벤치마크 프로그램 별로 성능의 손실을 분석하면, *histogram*에서 명령어 캐쉬에 의하여 27.0 %의 최대 손실을 기록하였으며, *FIR*에서 데이터 캐쉬에 의하여 최대 18.8 % 성능의 하락을 가져왔다. 평균적으로 명령어 캐쉬와 데이터 캐쉬에 의하여 각각 11.7 %와 9.3 %로 성능이 하락하였으며, 태스크 예측 오류 및 태스크 캐쉬 미스로 인한 성능의 하락은 1.7 %와 0.8 %로 낮았다. 이것은 슈퍼스칼라 코어를 채택하여 RISC 코어의 경우보다 모두 증가한 값들이다.

윈도우의 크기가 32일 때, 16-코어 디지털 신호처리 슈퍼스칼라 프로세서는 1-코어일 때 0.7 IPC, 16-코어일 때 6.4 IPC를 기록하였으며, 16-코어 RISC 디지털 신호처리 프로세서에 비하여 평균 2.3 배의 성능 향상을 가져왔다. 또한, 1-코어 순차 디지털 신호처리 프로세서와

16-코어 순차 디지털 신호처리 프로세서의 성능을 비교하면 8.9 배를 기록하였는데, 이것은 같은 조건에서의 RISC의 8.6 배를 능가하였다.

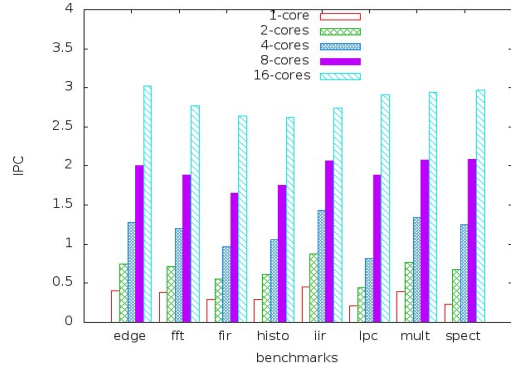


그림 3. 멀티코어 RISC 디지털 신호처리 프로세서의 모의실험 결과  
 Fig. 3. The simulation results of multicore RISC digital signal processor

## 3. 멀티코어 비순차 디지털 신호처리 프로세서 모의실험 결과

그림 5(a)부터 그림 5(d)에 윈도우의 크기가 4부터 32 인 비순차 디지털 신호처리 슈퍼스칼라 코어를 단위로 하는 멀티코어 디지털 신호처리 프로세서의 모의실험 결과를 나타냈다.

코어의 개수가 증가할수록 성능이 평균 2.0 배 향상하였으며, 윈도우의 크기가 2 배가 될 때마다 성능이 1.2 배 향상되었다. 이것은 순차 실행의 경우보다 모두 높은 값으로서, 명령어 비순차 실행으로 인하여 명령어 수준 병렬성이 증가하였기 때문이다.

한편, 비순차 실행에 의하여 평균적으로 명령어 캐쉬에 의하여 34.9 %, 데이터 캐쉬에 의하여 26.1 %의 성능 하락을 가져왔으며, 태스크 예측 오류 및 태스크 캐쉬 미스에 의한 성능은 각각 7.6 %와 6.3 % 하락하였다. 이 값들은 RISC 코어와 순차실행의 슈퍼스칼라 코어를 채택한 경우보다 모두 증가하였다.

윈도우의 크기가 32일 때, 평균적으로 1-코어는 0.9 IPC, 16-코어일 때 18.7 IPC를 기록하였으며, 1-코어에 대하여 16-코어는 19.5 배로 성능이 크게 증가하였다. 이것은 같은 조건에서 RISC 코어를 채택하였을 때의 8.6 배, 순차 슈퍼스칼라 코어를 채택하였을 때의 8.9 배보다 훨씬 높은 값이다.

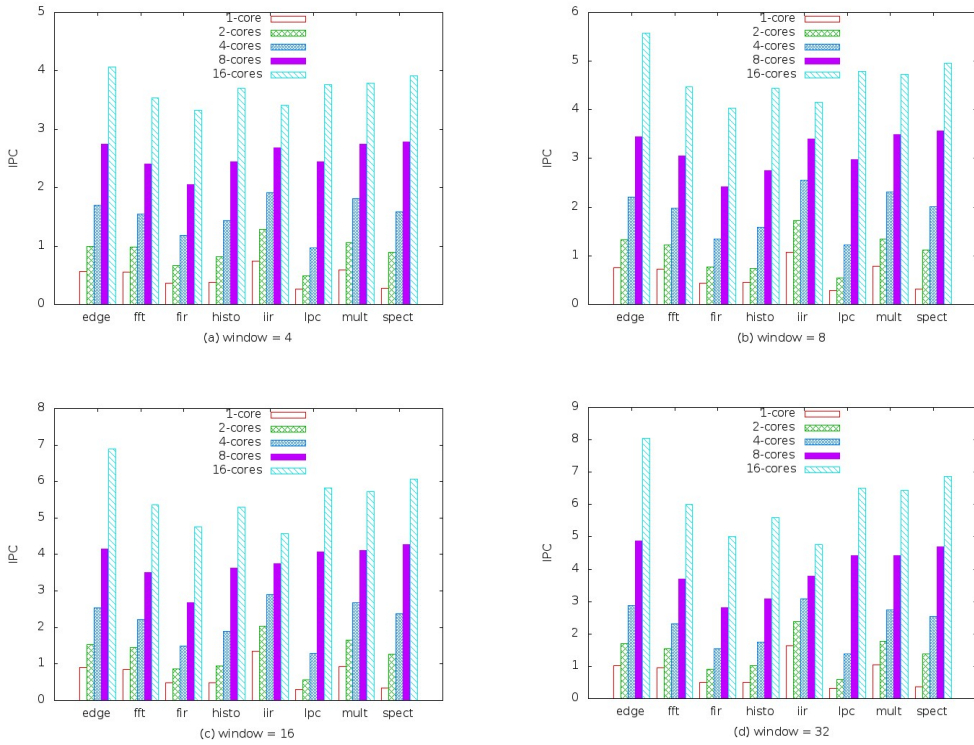


그림 4. 멀티코어 순차 디지털 신호처리 프로세서의 모의실험 결과  
 Fig. 4. The simulation results for the multicore in-order digital signal processor

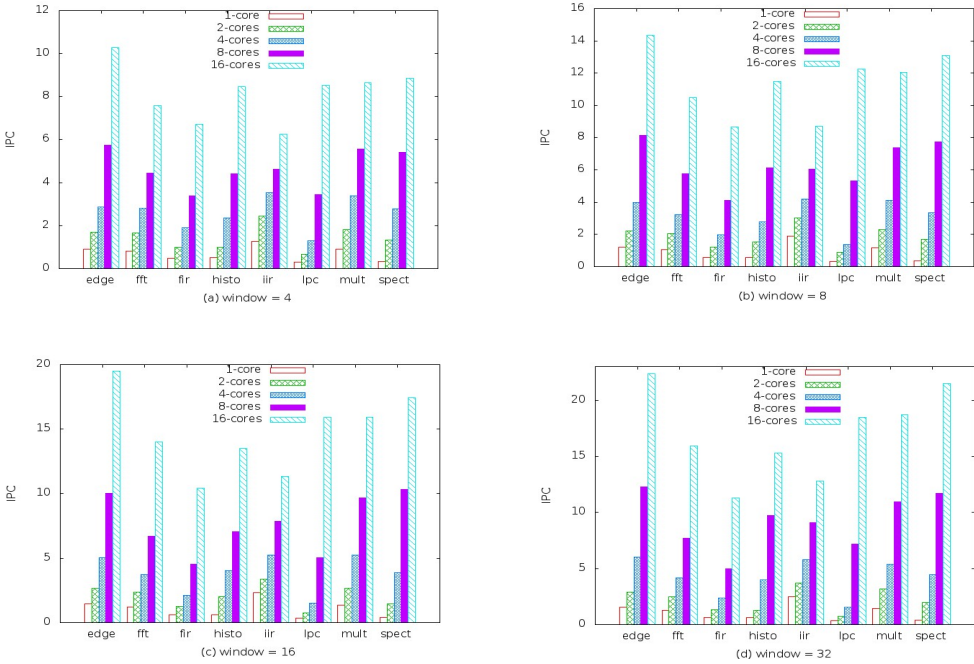


그림 5. 멀티코어 비순차 디지털 신호처리 프로세서의 모의실험 결과  
 Fig. 5. The simulation results for the multicore out-of-order digital signal processor

## V. 결론

본 논문에서는 RISC, 순차 및 비순차 수퍼스칼라로 구성되는 2 개부터 16 개까지의 멀티코어 디지털 신호처리 프로세서 아키텍처에 대하여, UTDSP 벤치마크를 입력으로 하여 성능을 측정하고 결과를 분석하였다. 그 결과, 1-코어와 16-코어 디지털 신호처리 프로세서의 성능을 비교하였을 때, 코어의 구조에 따라서 최저 8.6 배에서 최고 19.5 배 성능의 향상을 얻을 수 있었다.

추후로, 동질 코어(homogeneous core)가 아닌 비동질 코어(heterogeneous core)를 채택하는 비대칭 칩 멀티프로세서(asymmetric chip multiprocessor) 구조를 갖는 디지털 신호처리 프로세서에 대한 연구가 필요하다. 또한, 최신 경향인 코어 수 64개 이상의 매니코어 (many-core) 아키텍처로 구성된 디지털 신호처리 프로세서에 대한 연구 및 모의실험을 수행할 예정이다<sup>[11,12]</sup>.

## References

[1] L. J. Karam, I. AlKamal, A. Gatherer, G. A. Frantz, D. V. Anderson, B. L. Evans, "Trends in Multi-core DSP Platforms," IEEE Signal Processing Magazine, pp. 1- 10, Nov. 2009

[2] P. K. Dubey, G. B. Adams III, and M. J. Flynn, "Instruction Window Size Trade-Offs and Characterization of Program Parallelism," IEEE Transactions on Computers, vol. 43, pp. 431-442, Apr. 1994.

[3] 이종복, "멀티코어 프로세서의 명령어 자취형 모의실험에 대한 연구," 한국인터넷방송통신학회, 제12권, 제3호, pp. 9-13, 2012년 6월.

[4] S. W. Keckler, K. Olukotun, and H. P. Hofsee, "Multicore Processors and Systems," Springer. 2009.

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

[6] <http://www.eecg.toronto.edu/~corinna/DSP/infrastructure/UTDSP.html>

[7] 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.

[8] 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.

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

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

[11] I. Jeon, S. Kang, H. Yang, "Development of Security Quality Evaluate Basis and Measurement of Intrusion Prevention System," Journal of the Korea Academia-Industrial cooperation Society, v.11, no. 4, Apr. 2010.

[12] D. K. Lee, J. H. Kwon, "Social Search Algorithm considering Recent Interests of User", Journal of Korean Institute of Information Technology, vol. 9, issue 4, pp. 187-194, Apr. 2011.

## 저자 소개

### 이 종 복(정회원)



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

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

※ 본 연구는 한성대학교 교내연구장려금 지원과제임.