

논문 2013-08-03

# 멀티미디어 전용 명령어를 내장한 멀티코어 프로세서 구현 및 검증

(Implementation and Verification of a Multi-Core Processor including Multimedia Specific Instructions)

서준상, 김종면\*

(Jun-Sang Seo, Jong-Myon Kim)

**Abstract :** In this paper, we present a multi-core processor including multimedia specific instructions to process multimedia data efficiently in the mobile environment. Multimedia specific instructions exploit subword level parallelism (SLP), while the multi-core processor exploits data level parallelism (DLP). These combined parallelisms improve the performance of multimedia processing applications. The proposed multi-core processor including multimedia specific instructions is implemented and tested using a Xilinx ISE 10.1 tool and SoCMaster3 testbed system including Vertex 4 FPGA. Experimental results using a fire detection algorithm show that multimedia specific instructions outperform baseline instructions in the same multi-core architecture in terms of performance (1.2x better), energy efficiency (1.37x better), and area efficiency (1.23x better).

**Keywords :** Multimedia specific instructions, Multi-core processors, Area efficiency, Energy efficiency.

## 1. 서 론

최근 모바일 환경에서 고해상도의 동영상 및 정지영상 등 다양한 형태의 멀티미디어에 대한 실시간 서비스 요구가 증가함에 따라 멀티미디어 데이터를 신속하게 처리할 수 있는 기술 개발의 필요성이 증대되고 있다. 또한 모바일 환경에서는 성능

\* Corresponding Author (jmkim07@ulsan.ac.kr)

Received: 16 Oct. 2012, Revised: 13 Nov. 2012,

Accepted: 14 Nov. 2012.

J.S. Seo, J.M. Kim: University of Ulsan

※ 본 연구는 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0031322)의 연구 수행으로 인한 결과물이며, 또한 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No. 2012-0004962)이며, 지식경제부의 광역경제권 선도산업 육성사업의 일환인 “동남광역경제권 선도산업 지원단”의 2012년 연구비 지원으로 수행되었음

만큼이나 저전력 기술에 대한 요구가 대두되고 있다.

기존의 ASIC(Application-Specific Integrated Circuit)은 멀티미디어 처리에서 요구되는 고성능을 만족시킬 수 있지만 다양한 형태의 멀티미디어 애플리케이션에서 요구되는 범용성을 만족시키지 못한다[1, 2]. 반면에 범용 마이크로프로세서 (GPP, General-Purpose Processor)나 DSP (Digital Signal Processor)들은 다양한 애플리케이션에 대해 충분한 범용성을 제공하지만 멀티미디어 애플리케이션에서 요구되는 높은 레벨의 성능을 만족시키지 못한다. 이유는 GPP나 DSP는 프로세서 구조의 특성상 멀티미디어에 내재한 고도 병렬성 (massive parallelism)을 추구하지 못하기 때문이다.

고성능 멀티미디어 처리를 위한 대안 중에 하나로 SIMD (Single Instruction Multiple Data) 기반 멀티코어 프로세서가 유망하다 [3]. 명령어 레벨 (instruction-level)이나 스레드 레벨 (thread-level) 프로세서들은 실리콘 면적을 멀티포트 레지스터 파일 (multiported register file), 캐쉬 (cache), 파

표 1. 범용 마이크로프로세서용 멀티미디어 명령어

Table 1. Multimedia instructions for general-purpose microprocessors

Processor	Extension	Product	Instructions	Register File
HP	MAX-1	1994	9	Integer (31x64b)
Sun	VIS	1995	121	FP (32x64b)
HP	MAX-2	1995	8	Integer (32x64b)
MIPS	MIPS-V	(-)	29	FP (32x64b)
MIPS	MDMX	(-)	74	FP (32x64b), Acc. (1x92b)
Intel	MMX	1997	57	FP (8x64b)
DEC	MVI	1997	13	Integer (31x64b)
Cyrix	Extended MMX	1997	12	FP (8x64b)
AMD	3D Now!	1998	21	FP (8x64b)
Intel	SSE	1999	70	8x128b
Motorola	AltiVec	1999	162	32x128b
MIPS	MIPS-3D	(-)	23	FP (32x64b)
AMD	Enhanced 3D Now!	1999	24	FP (8x64b)
Intel	SSE2	(-)	144	8x128b

이프라인 (deep pipelined) 기능 유닛 등으로 사용하는 반면, SIMD기반 멀티코어 프로세서는 수십, 수백 개의 저비용 프로세싱 엘리먼트 (Processing Element, PE)들을 이용하여 고성능을 추구하고 동시에 저장장소와 데이터 통신 요구를 최소화하기 위해 프로세싱 엘리먼트와 데이터 입출력을 동일위치에 배치함으로써 저전력을 만족시킨다 [4-6]. 특히, SIMD기반 멀티코어 프로세서는 지역성 (locality)이나 규칙성(regularity)이 있는 2차원 패턴의 이미지나 비디오 처리에 있어서 최적의 프로세서 구조이다. 현재 각광받고 있는 GPU (Graphics Processing Unit)도 이러한 SIMD기반의 멀티코어 프로세서의 한 예이다. 또한 멀티미디어 확장 명령어는 여러 개의 서버워드 데이터를 하나의 긴 레지스터에 패키징하여 처리하는 서버워드 병렬성 (subword parallelism)을 이용함으로써 성능을 향상시킨다 [7].

본 논문에서는 기존의 멀티코어 프로세서에 멀티미디어 전용 명령어를 추가하여 데이터 레벨 병렬성 뿐만 아니라 서버워드 병렬성을 동시에 추구함으로써 멀티미디어 애플리케이션에서 요구되는 성능을 만족시키고자 한다. 이를 검증하기 위해 Xilinx ISE 10.1 툴과 Vertex 4 FPGA를 포함한 SocMaster 3 테스트 베드에 구현하였고, 화제감지 알고리즘을 이용하여 모의 실험한 결과, 멀티미디어 전용 명령어는 베이스라인 멀티코어에 비해 면적은 8% 증가시키는 반면 성능은 1.2배, 에너지 효율은 1.37배, 면적 효율은 1.23배 향상시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로써 범용 프로세서용 멀티미디어 확장 명령어에 대해 기술하고, 3장에서는 시뮬레이션 환경으로

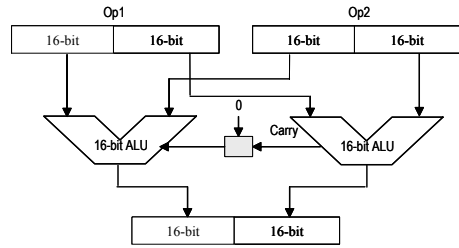


그림 1. 멀티미디어 확장 명령어의 예

Fig. 1 An example of multimedia extension instruction

써 SIMD기반의 멀티코어 프로세서와 이를 위한 실험 방법론을 설명한다. 4장에서는 멀티미디어 전용 명령어와 이를 이용한 화제감지 알고리즘을 기술하며, 5장에서는 이에 대한 성능을 분석한다. 마지막으로 6장에서는 본 논문의 결론을 맺는다.

## II. 관련 연구

### 1. 범용 프로세서용 멀티미디어 확장 명령어

범용 마이크로프로세서 제조회사는 멀티미디어 애플리케이션의 성능을 향상시키기 위해 멀티미디어 명령어를 그들의 instruction set architecture (ISA)에 첨가하였다. 표 1은 현재 모든 마이크로프로세서 제조회사에서 발표한 멀티미디어 확장 명령어의 리스트를 보여준다. 이러한 멀티미디어 확장 명령어의 주요 장점은 하나의 넓은 레지스터 (64 비트 혹은 128비트 예)에 여러 개의 작은 데이터를 저장하고 동시에 처리함으로써 성능을 향상시킨다.

제조회사의 타깃 애플리케이션에 따라 이러한 멀티미디어 명령어는 다양하다. Motorola AltiVec은 가장 많은 수의 SIMD 명령어 (162개)를 가지고 있는 반면, HP MAX-1은 단지 8개의 SIMD 명령어를 가지고 있다. 대부분의 멀티미디어 명령어 (AMD 3DNow!, DEC MVI, Intel MMX, Sun VIS)는 64 비트 레지스터를 사용하는 반면, Motorola AltiVec과 Intel SSE는 128 비트 레지스터를 이용한다. 한가지 주목할 만한 예외는 MIPS MDMX인데 이 명령어는 여러 번의 연산에 의한 결과를 축적하기 위해 하나의 넓은 누산기를 가지고 있다. 이러한 명령어의 유사성에도 불구하고 각각의 명령어는 독특한 특징을 가지고 있다. 예를 들어, MAX-2는 실행 유닛과 정수 레지스터를 재사용하여 추가적인 하드웨어를 필요로 하지 않는 반면, AltiVec은

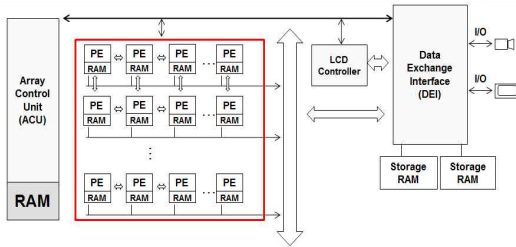


그림 2. SIMD기반 멀티코어 프로세서  
Fig. 2 SIMD based multi-core processor

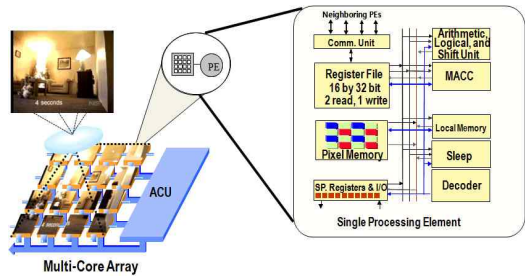


그림 3. 단일 PE 구조  
Fig. 3 Single PE architecture

전적으로 새로운 실행 유닛을 요구한다.

그림 1은 멀티미디어 확장 명령어의 한 예이다. 멀티미디어 확장 명령어는 하나의 넓은 레지스터에 여러 개의 작은 데이터를 저장하고 단일 명령 다중 데이터(SIMD, Single-Instruction Multiple-Data)를 기반으로 하여 서버워드 병렬처리를 수행함으로써 성능을 향상시킨다.

### III. 시뮬레이션 환경

#### 1. SIMD 기반 멀티코어 프로세서

그림 2는 본 논문의 베이스 아키텍처인 SIMD기반 멀티코어 프로세서의 블록 다이어그램을 보여준다. 2중 격자구조의 프로세싱 엘리먼트(Processing Element, PE)와 이를 제어하는 Array Control Unit (ACU), 외부 I/O와 PE 로컬 레지스터 간의 데이터 전송하는 DEI(data exchange interface)로 구성되어 있다. ACU는 명령어를 인출(fetch)하여 전체 PE에 동시에 전송(broadcasting)하며, 또한 특정 PE를 활성화(active) 또는 비활성화(sleep) 시키는 역할을 한다. ACU에 의해 데이터가 각 PE에 균등하게 분배되면 PE들은 명령어를 SIMD기반에서 동시에 수행한다.

그림 3은 단일 프로세싱 엘리먼트(PE)의 구조를 보여주며, 각각의 PE는 다음과 같은 특징을 가진다.

- 32비트 16384개 워드로 구성된 로컬 메모리
- 32비트 폭의 16개 3포트 범용 레지스터
- 기본적인 산술/논리 연산을 수행하는 ALU
- 64비트 곱셈 및 누산기 (multiply accumulator)
- 멀티 비트 산술/논리 시프트 연산을 수행하는 배럴 시프트
- 지역 정보를 이용해 각 PE들을 활성 및 비활성 시키는 Sleep 유닛
- 이웃하는 PE들과 데이터 통신을 위한 NEWS (north-east-west-south) 네트워크 및 serial I/O유닛

#### 2. 실험 방법론

그림 4는 SIMD기반 멀티코어 프로세서의 성능, 에너지효율 및 면적효율을 평가하기 위한 실험 방법론을 보여준다. 실험 방법론은 세 가지 레벨인 애플리케이션, 아키텍처 및 테크놀로지 레벨로 구성되어 있다. 애플리케이션 레벨에서는 매크로 프로세서용 정밀 사이클 시뮬레이터를 이용하여 사이클 개수, 동적 명령어 빈도, 시스템 이용률(system utilization) 등의 실행 데이터를 추출한다. 아키텍처 레벨에서는 Chai가 제안한 이중 병렬 프로세서용 아키텍처 모델링 툴을 사용하여 모델링된 아키텍처의 디자인 변수들을 계산한다 [8]. 테크놀로지 레벨에서는 아키텍처 레벨에서 구해진 디자인 변수들을 Generic System Simulator (GENESYS)의 입력으로 사용한다 [9]. GENESYS는 멀티코어 어레이 및 멀티 클러스터 등의 다양한 시스템 구조를 모델링하기 위한 테크놀로지 모델링 툴로서 각 코어들은 동기 ASIC 칩으로 표현되는 계층적 모델로 구성된다. 이러한 계층적 모델은 기본요소, 재료, 디바이스, 회로, 시스템 등 5-레벨로 이루어져 있다. 처음 세 가지 레벨은 재료특성과 스위칭 디바이스 특성의 물리적 효과를 모델링하며, 회로 레벨은 신호 지연, 동적 및 정적 에너지와 같은 게이트의 특징을 모델링하며, 마지막으로 시스템 레벨은 싱글 ASIC 칩을 묘사하게 위한 아키텍처, 연결구조, 패키징 상세들을 포함한다. GENESYS는 입력된 디자인 변수를 바탕으로 각 아키텍처 모델들의 사이클 시간(cycle time), 와이어 지연(latency), 전력(power), 클록 주파수(clock frequency)와 동적 및 정적 전력 등의 테크놀로지 변수를 예측한다. 마지막으로 세 레벨에서 구해진 데이터베이스를 조합하여 실행 시간, 에너지 효율, 시스템 면적 효율을 측정할 수 있다.

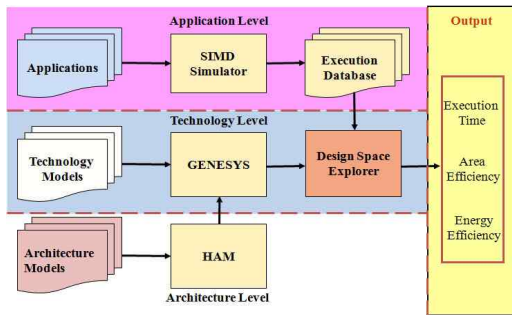


그림 4. 멀티코어 프로세서를 위한 실험 방법론  
Fig. 4 Simulation infrastructure for a multi-core processor

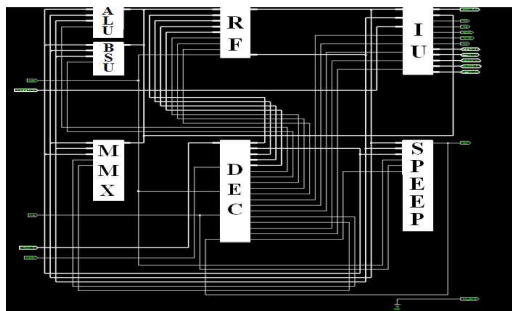


그림 5. 단일 PE에 멀티미디어 전용 명령어 모듈이 추가된 RTL 스키매틱  
Fig. 5 RTL schematic for a single PE including multimedia specific instructions

#### IV. 멀티미디어 전용 명령어를 적용한 화재감지 알고리즘

##### 1. 멀티미디어 전용 명령어 구현

그림 5는 기존 단일 프로세서에 멀티미디어 전용 명령어 모듈이 추가된 RTL (Register Transfer Level) 스키매틱을 보여준다. 멀티미디어 전용 명령어 모듈내의 연산 방식은 서브워드 단위로 연산이 가능하도록 설계되었으며, 설계된 명령어를 기능별로 구분하면 다음과 같다.

##### 1.1 Pack/Unpack 명령어

그림 6은 멀티미디어 전용 명령어 모듈의 내부에서 Pack/ Unpack 명령어의 구현 및 동작 상태를 보여준다. BusA와 BusB는 연산의 대상이 되는 두 개의 오퍼랜드가 입력되는 버스이고, BusC는 결과 데이터 버스이다. 제어 신호에 의해 결과 데이터가 선택되고 aluO 신호를 통해 BusC로 결과데이터가 출력된다.

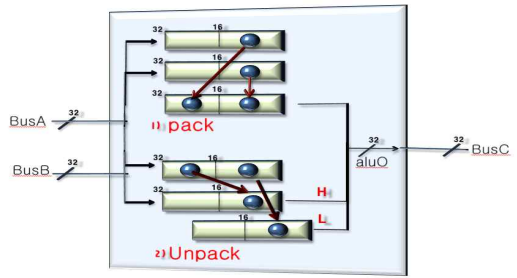


그림 6. Pack/Unpack 연산 예  
Fig. 6 An example of Pack/Unpack operations

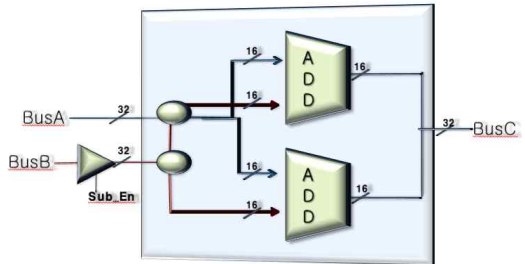


그림 7. 병렬 산술 연산 명령어 예  
Fig. 7 An example of parallel arithmetic operations

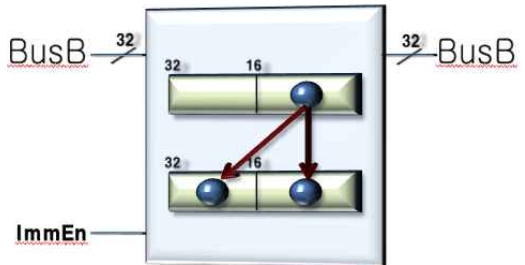


그림 8. Immediate packing 명령어의 예  
Fig. 8 An example of immediate packing instructions

##### 1.2 산술 및 논리연산 명령어

그림 7은 멀티미디어 전용 명령어 모듈 내에서 병렬 산술 연산 과정을 보여준다. 덧셈의 경우, 두 개의 16비트 데이터가 Pack된 32비트 소스 레지스터에서 상위 16비트와 하위 16비트로 분리시켜 각각 덧셈기를 통해 연산된 결과를 BusC를 통해 출력한다.

##### 1.3 Immediate Packing 명령어

멀티미디어 전용 명령어에서 상수를 연산과정

표 2. 멀티미디어 전용 명령어 요약

Table 2. Summary of multimedia specific instructions

	Mnemonic
Arithmetic	PADD ,PADDI
	PSUB ,PSUBI
	PMULHW ,PMULHWI
	PMULLW ,PMULLWI
	PMADDWD
Comparison	PCMPEQ
	PCMPGT
Conversion	PACKUSWB
	PUNPCKH
	PUNPCKL
Logical	PAND ,PANDI
	POR ,PORI
	PXOR ,PXORI
Shift	PSLL ,PSLLI
	PSRL ,PSRLI
	PSRA ,PSRAI

에 사용하는 경우가 빈번하게 존재한다. 그림8은 BusB의 하위 16비트 immediate 데이터 값을 상위 16비트에 복사하는 immediate packing을 수행하는 명령어를 보여준다.

1.4 그 외 명령어

표 2는 본 논문에서 구현한 멀티미디어 전용 명령어의 종류를 보여준다. 비교연산은 묶음처리가 된 2개의 레지스터를 비교하는 연산으로 결과가 참이면 0xffff, 거짓이면 0x0000으로 출력된다. 특히, PMADDWD 명령어는 벡터의 내적을 구하는 연산에 주로 사용되며 명령어로서 곱셈 연산 후 이전의 저장된 데이터와 더하기 연산을 수행한다.

2. 화재감지 알고리즘

그림 9는 본 논문에서 구현한 멀티미디어 전용 명령어의 성능 평가를 위해 사용된 화재감지 알고리즘 [10]의 흐름도를 보여준다. 총 4가지 단계로 구성되어 있는 화재감지 알고리즘은 움직임 영역 및 색상 분할로 화재영역을 추정하며, 조명과 화재 분할을 위해 웨이블릿 계수의 특징을 추출하고 이를 신경망(Neural Network)의 입력으로 사용하여 화재 유무를 판별한다.

그림 10은 화재감지 알고리즘의 각 단계별 결

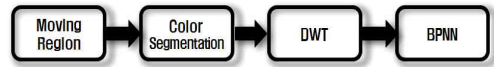


그림 9. 화재감지 알고리즘 흐름도

Fig. 9 Flow of a fire detection algorithm

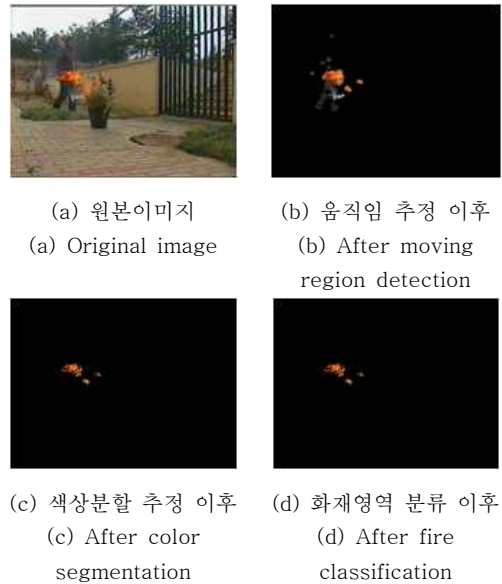


그림 10. 화재감지 알고리즘의 결과 이미지

Fig. 10 Resulting images of the fire detection algorithm

과 이미지를 보여주며, 각 단계별 처리과정은 다음과 같다.

2.1 움직임 추정

1단계인 움직임 추정 (moving region estimation)에서는 이전 프레임들과 현재 프레임의 감산 결과를 통하여 움직임 영역을 추정하여 배경과 이동하는 객체를 구분한다. 이를 통하여 배경 부분에 할당된 PE를 비활성화 시킴으로써 움직임 객체로 추정된 부분과 구분짓는다.

2.2 색상 분할 추정

색상 분할 추정 과정을 통해 화재 또는 화재와 유사한 색깔의 옷을 입은 사람과 같은 비화재 객체 등과 구별한다.

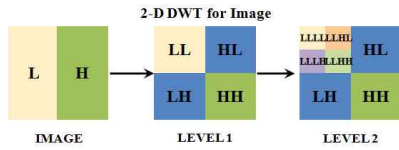


그림 11. 2단계 웨이블릿 처리 과정

Fig. 11 Flow of two-step wavelet processing

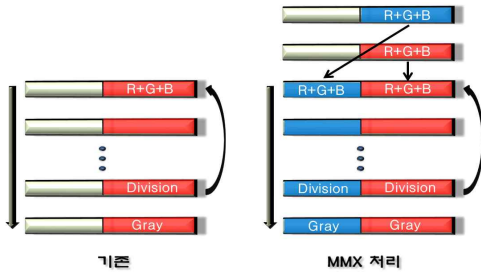


그림 12. 멀티미디어 명령어를 사용한 나눗셈 연산

Fig 12. Division process with and without multimedia specific instructions

2.3 DWT를 이용한 특징 파라미터 추출

본 논문에서는 전처리 과정인 2단계 과정 이후 웨이블릿 계수를 이용하여 불꽃의 특징을 추출한다. 그림 11은 2단계 웨이블릿 분해 과정을 보여주며, 이를 멀티코어 프로세서에 구현하였다.

2.4 화재영역 분류

본 논문에서는 9-14-1 토폴로지로 구성된 신경망을 멀티코어에 구현하였고 화재영역을 분류할 수 있었다.

2.5 멀티미디어 전용 명령어를 이용한 화재감지 알고리즘

화재감지 알고리즘의 수행 시간 중 약 78%는 색상분할단계에서 소비되며, 특히 색상분할단계의 그레이 스케일(Grayscale) 변환부분에서 많은 사이클이 소요된다. 그레이 스케일 변환부분의 약 50% 이상은 나눗셈 연산과정을 반복적으로 수행한다. 따라서 색상분할단계의 나눗셈 연산과정에 멀티미디어 전용 명령어를 적용하면 수행되는 사이클 수를 절약할 수 있다. 그림 12는 나눗셈 연산을 위해 멀티미디어 전용 명령어를 사용한 경우와 그렇지 않은 경우를 보여준다.

표 3. 시스템 파라미터  
Table 3. System parameter

Parameter	Value
Number of PEs	16
Pixels/PE	32 x 32 (1,024)
Memory/PE[word]	8,192[word, 32bit]
VLSI Technology	130nm
Clock Frequency	100MHz
Interconnection Network	Mesh

V. 성능평가

1. 성능 평가 지표

멀티미디어 전용 명령어 내장형 멀티코어 프로세서의 성능 평가를 위해 실행시간, 에너지 효율, 시스템 면적 효율 [10]을 측정한다. 실행 시간은 초음파 영상처리 알고리즘이 수행된 시간을 뜻하며 식 (1)과 같이 계산되어진다. 에너지 효율은 소비된 에너지 당 처리량으로써 식 (2)와 같이 계산되며, 시스템 면적효율은 단위 면적당 처리량을 나타내며 식 (3)과 같이 계산된다. 여기서  $C$ 는 사이클 수,  $f_{clk}$ 는 클럭 주파수,  $Energy$ 는 130nm 공정에서 소비된 에너지(Joule),  $Area$ 는 시스템의 단위 면적( $mm^2$ )을 나타낸다.

표 3은 멀티미디어 전용 명령어를 추가할 멀티코어 프로세서의 시스템 파라미터를 보여준다. 효율적인 영상처리를 위해 16개의 PE를 격자 구조로 연결하였으며, 각각의 PE는 균등하게 할당된 1024개의 픽셀 영상데이터를 처리한다. 16개의 프로세싱 엘리먼트는 각각 32비트 워드 단위의 8192개의 로컬 메모리를 가지고 있으며, 130nm 공정과 100MHz 클럭 주파수로 구현되었다.

$$t_{exec} = \frac{C}{f_{clk}} \tag{1}$$

$$\eta_E = \frac{1}{t_{exec} \cdot Energy} \left[ \frac{1}{s \cdot Joule} \right] \tag{2}$$

$$\eta_A = \frac{1}{t_{exec} \cdot Area} \left[ \frac{1}{s \cdot mm^2} \right] \tag{3}$$

2. 성능분석 및 결과

본 논문에서 제안한 멀티미디어 전용 명령어를 내장한 멀티코어 프로세서를 구현하고 성능 평가할

FPGA 프로토타입 시스템은 슬라이스 (Slice), 로직 셀과 같이 구성 가능한 로직 블록(Configurable Logic Block ,CLB)과 멀티플라이어와 같이 고정된 함수 로직 등으로 구성되어 있다. 가장 하위레벨에 있는 CLB는 슬라이스로 구성되어 있으며, 슬라이스는 LUT(Look Up Table)와 플립플롭으로 구성되어 있다.

표 4. 멀티미디어 전용 명령어에 사용된 LUT와 Slice 수

Table 4. Number of LUT and slice for multimedia specific instructions

	LUT	Slice	Cycle
멀티미디어 전용 명령어 미적용	29412	16209	330556
멀티미디어 전용 명령어 적용	31852	17606	264556

표 4는 FPGA 보드 상에서 멀티미디어 전용 명령어가 추가된 멀티코어 프로세서의 LUT, Slice 수 및 전체 사이클 수를 보여준다. 기존 멀티코어 프로세서에 멀티미디어 전용 명령어 모듈을 추가하였을 경우, 사용면적은 약 8% 증가하였으나, 사이클 수를 20% (264,556사이클)를 줄였다.

그림 13과 14는 멀티미디어 전용 명령어와 기존 명령어 사이의 실행시간, 에너지 효율 및 면적효율의 비교 결과를 보여준다. 멀티미디어 전용 명령어를 사용한 경우 약 20%의 실행시간 감소를 보였다. 또한 멀티미디어 전용 명령어는 에너지 효율 측면에서는 약 37%와 시스템 면적 효율에서는 약 23% 향상을 보였다. 에너지 효율은 소비된 단위 에너지 당 수행된 작업량을 의미하며, 시스템 면적 효율은 단위 면적당 처리량을 의미한다. 따라서, 멀티미디어 전용 명령어는 에너지 효율을 증가시킴으로써 배터리 전력을 더 오랜 시간동안 사용할 수 있으며, 시스템 면적 효율을 증가시킴으로써 시스템의 컴포넌트 이용률을 증가시키는 결과를 가져온다.

## VI. 결 론

본 논문에서는 대용량의 멀티미디어 데이터 처리를 향상시키기 위해 멀티미디어 전용 명령어 모듈을 기존의 SIMD 기반 멀티코어 프로세서에 추가하였고, FPGA 프로토타입 시스템에 구현 및 기능을 검증하였다. 멀티미디어 전용 명령어의 성능, 에

너지 효율 및 면적 효율에 대한 검증을 위해 화재 감지 알고리즘을 적용하여 분석하였다. 동일한 130nm 공정과 100MHz의 동작 주파수에서 모의 실험한 결과, 멀티미디어 전용 명령어는 동일한 멀티코어 프로세서에서 실행시간 측면에서는 20% 감소, 에너지 효율 측면에서는 37% 향상 및 시스템 면적 효율 측면에서는 23% 향상을 보였다. 이러한

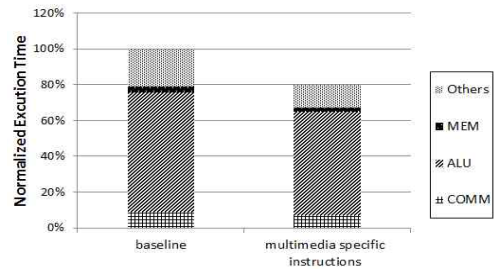


그림 13. 멀티미디어 전용 명령어의 실행시간  
Fig. 13 Execution time with and without multimedia specific instructions

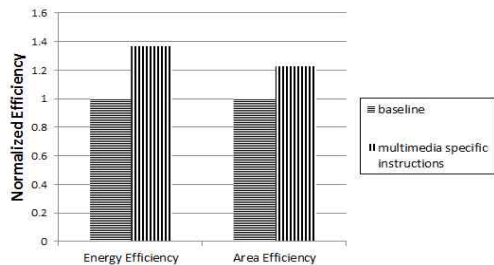


그림 14. 멀티미디어 전용 명령어의 에너지 및 시스템 면적 효율

Fig. 14 Energy and area efficiencies with and without multimedia specific instructions

결과에서 멀티미디어 전용 명령어는 모바일 멀티코어 프로세서의 성능 및 에너지 효율 향상을 위한 잠재성을 보여준다. 향후 현재 널리 이용되고 있는 상용 GPU (Graphics Processing Unit)와의 성능 및 효율 비교를 통해 제안하는 멀티미디어 명령어 내장형 멀티코어 프로세서의 우수성을 검증할 것이다.

## 참고문헌

[1] E.B. Bourennane, S. Bouchoux, J. Miteran, M. Paindavoine, S. Bouillant, "Cost Comparison of Image Rotation Implementations on Static and



- Dynamic Reconfigurable FPGAs,” Proceedings on the International Conference of Acoustics, Speech, and Signal Processing, Vol. 3, pp.3176-3179, 2002.
- [2] X.G. Jiang, J.Y. Zhou, J.H. Shi, H.H. Chen “FPGA Implementation of Image Rotation Using Modified Compensated CORDIC,” Proceedings on the International Conference of ASIC, Vol. 2, pp.752 - 756, 2005.
- [3] S.H. Lee, “The Design and Implementation of Parallel Processing System using the Nios<sup>(R)</sup> II Embedded Processor,” The Korea Society of Computer and Information, Vol. 14, No. 11, pp.97-103, 2009 (in Korean).
- [4] A. Gentile, D.S. Wills, “Portable Video Supercomputing,” IEEE Trans. on Computers, Vol. 53, No. 8, pp.960 - 973, 2004.
- [5] Y.H. Kim, J.M. Kim, “Design Space Exploration of Optimal many-Core Processors for Discrete Wavelet Transform,” Journal of Institute of Embedded Engineering of Korea, Vol. 7, No. 5, pp. 277-284, 2012 (in Korean).
- [6] Y.M. Kim, J.M. Kim, “Design and Verification of High-Performance Parallel Processor Hardware for JPEG Encoder,” Journal of Institute of Embedded Engineering of Korea, Vol. 6, No. 2, pp.100-107, 2011 (in Korean).
- [7] R.B. Lee, “Subword Parallelism with MAX-2,” IEEE Micro, Vol. 16, No. 4, pp.51-59, 1996.
- [8] S.M. Chai, T. Taha, D.S. Wills, J.D. Meindl, “Heterogeneous architecture models for interconnect-motivated system design,” IEEE Transactions on VLSI Systems, Vol. 8, No. 6, pp.660-670, 2000.
- [9] J.C. Eble, V.K. De, D.S. Wills, J.D. Meindl, “A Generic System Simulator (GENESYS) for ASIC Technology and Architecture beyond 2001,” Proceedings on the International Conference of ASIC, pp.193-196, 1996.
- [10] T.X. Truong, J.M. Kim, “An Effective Four-Stage Smoke-Detection Algorithm using Video Images for Early Fire-Alarm Systems,” Fire Safety Journal, Vol. 46, No. 5, pp.276-282, 2011.

## 저 자 소 개

### 서 중 상



2012년 현재, 울산대 전기공학부 석사과정.

관심분야: 병렬프로세서 구조, 임베디드시스템, 화재 감지 알고리즘

Email: stoking111@nate.com

### 김 종 면



1995년 명지대 전기공학과 학사.

2000년 University of Florida 전기컴퓨터공학과 석사.

2005년 Georgia Tech. 전기컴퓨터공학과 박사.

현재, 울산대 전기공학부 교수.

관심분야: 임베디드 SoC, 컴퓨터구조, 병렬처리.

Email: jmkim07@ulsan.ac.kr