

## 기타 음 합성을 위한 최적의 SIMD기반 매니코어 프로세서 구현

최지원\*, 강명수\*\*, 김종면\*\*\*

### Implementation of an Optimal SIMD-based Many-core Processor for Sound Synthesis of Guitar

Ji-Won Choi\*, Myeongsu Kang\*\*, Jong-Myon Kim\*\*\*

#### 요약

프로세서는 더 이상 동작 주파수를 높이는 방법이 아닌 다수의 프로세서를 집적하는 멀티프로세서로 기술 발전이 이루어지고 있다. 최근 2, 4, 8개의 프로세서 코어를 넘어 64, 128개 이상의 프로세서를 집적한 대규모 데이터 처리용 고성능 프로세서들이 개발되고 있다. 본 논문에서는 기타의 음 합성을 위한 최적의 매니코어 프로세서 구조를 제안한다. 기존의 연구에서는 하나의 기타 현에 하나의 프로세싱 엘리먼트(processing element, PE)를 할당하여 음을 합성하였으나, 본 논문은 하나의 기타 현에 여러 개의 PE를 할당하고 각각의 경우에 대해 시스템 성능, 시스템 면적 효율 및 에너지 효율을 평가하였다. 샘플링율이 44.1kHz, 양자화 비트 16인 기타 음을 사용하여 모의 실험한 결과, 시스템 면적 효율은 PE 수가 24개, 에너지 효율은 PE 수가 96개일 때 각각 최적의 효율을 보였다. 또한, 최적의 매니코어 프로세서를 이용하여 합성한 결과 합성음은 원음과 스펙트럼에서 매우 유사하였다. 더불어, 음 합성에 가장 많이 사용되는 TI TMS320C6416보다 시스템 면적에서 1,235배, 에너지 효율에서 22배의 향상을 보였다.

▶ Keyword : 기타, 음 합성, 매니코어 프로세서, 시스템 면적 효율, 에너지 효율

#### Abstract

Improving operating frequency of processors is no longer today's issues; a multiprocessor technique which integrates many processors has received increasing attention. Currently, high-performance processors that integrate 64 or 128 cores are developing for large data processing over 2, 4, or 8 processor cores. This paper proposes an optimal many-core processor for

• 제1저자 : 최지원 • 교신저자 : 김종면

• 투고일 : 2011. 03. 14, 심사일 : 2011. 03. 26, 게재확정일 : 2011. 05. 11.

\* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 석사과정

\*\* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 박사과정

\*\*\* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 교수

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2010-0027574).

synthesizing guitar sounds. Unlike the previous research in which a processing element (PE) was assigned to support one of guitar strings, this paper evaluates the impacts of mapping different numbers of PEs to one guitar string in terms of performance and both area and energy efficiencies using architectural and workload simulations. Experimental results show that the maximum area energy efficiencies were achieved at PEs=24 and 96, respectively, for synthesizing guitar sounds with sampling rate of 44.1kHz and 16-bit quantization. The synthesized sounds were very similar to original guitar sounds in their spectra. In addition, the proposed many-core processor was 1,235 and 22 times better than TI TMS320C6416 in area and energy efficiencies, respectively.

▶ Keyword : guitar, sound synthesis, many-core processor, area efficiency, energy efficiency

## I. 서론

최근 모바일 멀티미디어 기기들의 사용이 증가하면서 낮은 소비 전력으로 실시간 고성능의 멀티미디어 애플리케이션을 구현하는 고성능 멀티미디어 프로세서에 대한 요구가 높아지고 있는 추세이다[1,2]. 악기음 합성을 위해 많은 연구가 진행되어 왔는데, 중국 전통 현악기 구친(gu-chin), 핀란드 전통 현악기 칸텔레(kantele), 베트남 전통 현악기 단트란(dan tranh)이 대표적인 예이다[3-6]. 국내에서는 산조 가야금의 물리적 모델링에 대한 연구[7,8], 편경에 대한 진동모드 분석[9] 및 물리적 모델링 합성법을 이용한 줄 없는 기타 구현[10] 등이 있다. 그러나 손으로 현을 뜯거나 튕겨서 연주하는 현악기의 음 합성 알고리즘은 상당한 양의 연산과 입·출력 처리를 요구하기 때문에 실시간으로 사용하는 애플리케이션에 적용하기에는 많은 어려움이 있다[4,11]. 특히 현악기 사운드 엔진 개발은 기존 프로세서의 성능 제약으로 인하여 고품질의 악기 음 합성 알고리즘을 구현하기 어려운 실정이다[12,13].

ASIC(Application-Specific Integrated Circuit)은 이러한 음 합성 알고리즘에서 요구되는 고성능, 저전력을 충족시킬 수 있지만 다양한 애플리케이션에서 요구되는 프로그램 능력(programmability)과 유연성(flexibility)을 만족시키지 못한다[14]. 반면 범용 프로세서(general-purpose processor, GPP)나 디지털 신호처리기(digital signal processor, DSP)는 다양한 애플리케이션에 대해 충분한 프로그램 능력과 유연성을 제공한다. 하지만, 멀티미디어 애플리케이션에서 요구하는 높은 레벨의 성능을 만족시키지 못하는데, 이는 GPP나 DSP 프로세서 구조 특성상 멀티미디어 애플리케이션에 내재한 고도의 병렬성(massive paral-

lelism)을 활용하지 못하기 때문이다.

음 합성 애플리케이션과 같이 고성능을 충족시킬 수 있는 프로세서 모델 중에서 SIMD 기반 매니코어 프로세서 아키텍처가 대안으로 부각되고 있다. 명령어 레벨(instruction-level)이나 스레드 레벨(thread-level) 프로세서들은 실리콘 면적을 멀티포트 레지스터 파일(multiported register file), 캐쉬(cache), 파이프라인(deep pipelined) 기능 유닛 등으로 사용하는 반면, SIMD 기반 매니코어 프로세서는 수백 개의 저비용 프로세싱 엘리먼트(processing element, PE)를 이용하여 고성능을 추구하고 동시에 저장장소와 데이터 통신 요구를 최소화하기 위해 PE와 데이터 입·출력을 동일위치에 배치함으로써 저전력을 만족시킨다[15].

프로세서는 더 이상 동작 주파수를 높이는 방법이 아닌 다수의 프로세서를 집적하는 매니코어 프로세서로 기술 발전이 이루어지고 있다. 최근 2, 4, 8개의 프로세서 코어를 넘어 64, 128개 이상의 프로세서를 집적한 대규모 데이터 처리 및 과학 연산용 고성능 프로세서들이 개발되고 있다. 그러나 최적의 효율을 갖는 매니코어 프로세서에 관한 연구는 희박하다. 영상데이터 처리를 위해 프로세싱 엘리먼트당 픽셀 수(pixel-per-processing element)를 변화시키면서 최적의 효율을 보이는 아키텍처 구조를 찾는 연구는 진행되었으나[16,17] 음향 분야에서 이러한 연구는 전무한 실정이다.

본 논문에서는 물리적 모델링 기반 기타 음 합성을 위한 최적의 SIMD 기반 매니코어 프로세서를 찾기 위해 기타 현 하나에 할당하는 PE 수를 변화시키는 실험을 하였다. 각 실험에서 시스템의 성능, 시스템 면적 효율 및 에너지 효율을 측정하고 이를 바탕으로 최적의 매니코어 프로세서를 선택하였다. 본 논문에서는 일곱 가지의 PE 구조(PEs=6, 12, 24, 48, 96, 192, 384)에 대해 샘플링율은 44.1kHz, 양자화 비트는 16인 기타 음을 합성하였다. 130nm 테크놀로지와 720MHz 클럭 주파수에서 메쉬 구조로 연결하여 실험

한 결과, 시스템 면적 효율은 PE 수가 24개, 에너지 효율은 PE 수가 96개일 때 가장 높은 효율을 보였다. 또한 음 합성에서 가장 많이 사용되는 TI TMS320C6416보다 시스템 면적 효율에서 1,235배, 에너지 효율에서 22배의 향상을 보였다. 이러한 결과를 통해 현악기 음 합성을 위한 매니코어 시스템을 효율적으로 디자인 할 수 있는 해결책을 제시할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 매니코어 프로세서의 구조, 최적의 구조를 찾기 위한 방법론 및 음 합성 알고리즘의 병렬 구현에 대해 소개하고, 3장에서는 매니코어 프로세서를 이용한 기타 음 합성 결과를 분석하고, 기타 음 합성을 위한 최적의 시스템 면적 효율 및 에너지 효율을 보이는 구조를 선택한다. 끝으로 4장에서는 본 논문의 결론을 맺는다.

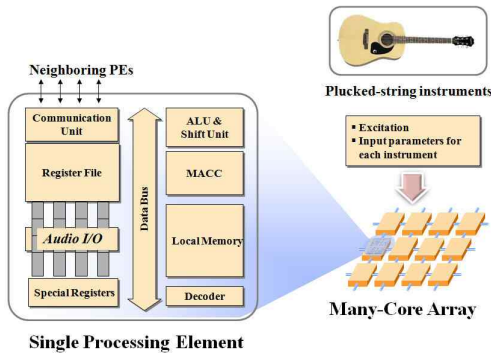


그림 1. SIMD 기반 매니코어 프로세서와 프로세싱 엘리먼트  
Fig. 1. A block diagram of a reference SIMD-based many-core processor and single processing element

## II. 시뮬레이션 환경

### 1. SIMD기반 매니코어 프로세서 모델

#### 1.1 기타 음 합성을 위한 매니코어 프로세서 구조

본 논문에서 사용한 SIMD 기반 매니코어 프로세서의 블록도는 그림 1과 같다. SIMD 기반 매니코어 프로세서는 여러 개의 PE와 이를 제어하는 ACU로 구성된다. 음 합성을 위한 파라미터 및 데이터가 각 PE에 일정하게 분배되면 PE는 매쉬 배열 구조에서 명령어들을 수행하며, 각 PE는 다음과 같은 특징을 갖는다.

- 32비트 폭의 512개 워드로 구성된 내부 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 배럴시프트,
- 지역 정보를 이용해 각 PE의 활성/비활성을 위한 SLEEP 유닛,
- 이웃하는 PE간 데이터 통신을 위한 NEWS(north-east-west-south) 네트워크 및 직렬 I/O유닛.

#### 1.2 매니코어 프로세서의 파이프라이닝

그림 2와 같이 SIMD 기반 매니코어 프로세서는 명령어 패치(fetch), 명령어 해석(decode), 명령어 실행(execution)의 3단계 파이프라인 구조로 설계되었다. 1단계에서는 ACU가 명령어 메모리로부터 명령어를 가져온다. 2단계에서는 ACU의 디코더 유닛이 ACU에서 수행되는 스칼라(scalar)명령어인지 PE에서 수행되는 벡터(vector)명령어인지를 구분하여 BusA, BusB, BusC의 각 포트에 해당되는 레지스터 주소 및 immediate 값을 할당한다. 마지막 3단계에서는 명령어가 각 유닛들의 컨트롤 신호에 의해 실행된다.

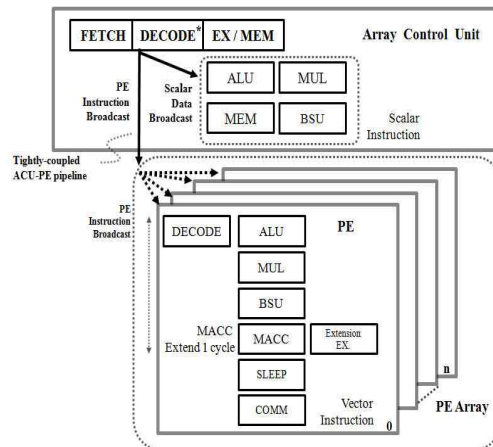


그림 2 매니코어 프로세서의 파이프라인 예  
Fig. 2. An example of pipelining of the many-core processor

### 2. 실험 방법론

세 가지 레벨(애플리케이션, 아키텍처, 테크놀로지)로 구성된 SIMD기반 매니코어 프로세서의 실험 방법론은 그림 3과 같다. 애플리케이션 레벨에서는 명령어 레벨의 SIMD 기반 매니코어 프로세서용 정밀 사이클 시뮬레이터를 이용하여 사이클 수, 동적 명령어 빈도, PE 시스템 이용률

(system utilization) 등의 실행 데이터를 추출한다. 아키텍처 레벨에서는 모델링된 아키텍처의 디자인 변수들을 계산하기 위해 Chai가 제안한 SIMD 기반 매니코어 프로세서용 아키텍처 모델링 툴을 사용하였다[18]. 테크놀로지 레벨에서는 각 아키텍처 모델들의 테크놀로지 변수(지연(latency), 전력(power), 클럭 주파수(clock frequency))를 계산하기 위해 Generic System Simulator(GENESYS)를 사용하였다[19]. 마지막으로 각 레벨에서 구해진 데이터베이스를 조합하여 실행 시간, 시스템 면적 효율 에너지 효율을 측정한다.

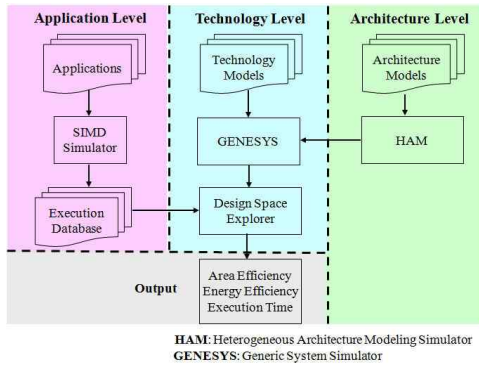


그림 3. 매니코어 프로세서를 위한 실행 방법론  
Fig. 3. Simulation methodology for the many-core processor

### 3. 기타의 음 합성 알고리즘

물리적 모델링 기반 현악기 음 합성을 위한 현 모델의 블록도는 그림 4와 같다.

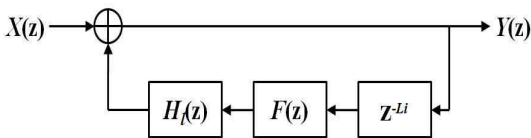


그림 4. 현 모델 블록도  
Fig. 4. A block diagram of the string model

여기 신호(excitation signal)는 몸체의 방사 특성을 포함하고 있고, 현 모델의 입력으로 사용된다. 지연 길이(delay length)는 샘플링율과 합성음의 기본 주파수의 비인 식(1)로 계산된다.

$$L = \frac{f_s}{f_0}, \tag{1}$$

여기서  $f_s$ 는 샘플링율,  $f_0$ 는 합성음의 기본 주파수이다. 이때 지연 길이  $L$ 은 정수부와 미소 지연부(소수부)의 합으로 표현되는데, 미소 지연부를 처리하기 위해 1차 라그랑즈 보간기(Lagrange interpolator)를 사용하였으며, 그 전달 함수는 식 (2)와 같다.

$$F(z) = h_0 + h_1 z^{-1}, \tag{2}$$

여기서  $h_0, h_1$ 은 라그랑즈 보간기의 계수이다.

루프 필터(loop filter)  $H_l(z)$ 는 현의 진동이 감쇄되는 현상을 표현하기 위한 것으로, 저역 통과 필터(low-pass filter)의 특성을 가지며 식 (3)과 같이 나타낼 수 있다[4].

$$H_l(z) = g \frac{1 + a_1}{1 + a_1 z^{-1}}, \tag{3}$$

여기서  $a_1$ 는 루프 필터의 차단 주파수를 결정하기 위한 계수이고,  $g$ 는 루프 필터의 이득이다.

최종적으로 도파관 모델을 이용한 기타 음 합성을 위한 전달 함수는 식 (4)와 같다.

$$Y(z) = \frac{X(z)}{1 - z^{-L} F(z) H_l(z)}. \tag{4}$$

### 4. 모델링된 매니코어 아키텍처

표 1은 기타의 음 합성을 위한 최적의 매니코어 아키텍처를 찾기 위해 모델링된 일곱 가지 매니코어 아키텍처의 파라미터를 보여준다. 이렇게 모델링된 PE 구성을 이용하여 샘플링율이 44.1kHz, 양자화 비트가 16인 기타 음을 합성하였다. 모델링된 모든 PE 아키텍처는 동일한 130nm 테크놀로지와 720MHz 클럭 주파수를 사용하여 시뮬레이션 하였다. 동시 발음 수가 6인 기타 음 합성을 위해 필요한 PE 수는  $NPE = 6 \times 2^i$  ( $i = 0, \dots, 6$ )으로 정의된다. 그림 5는 서로 다른 PE 구성에 따른 음 합성 예를 보이는데, PE가 6개 일 때는 하나의 PE가 하나의 기타 현을 지원하는 것과 동일하며, PE가 12일 때는 두 개의 PE가 하나의 기타 현을 지원하는 것과 동일하다. 이러한 서로 다른 PE 구성에 대해 기타 음 합성에 필요한 PE 내부 메모리는 식(5)와 같다.

$$MEM_{PE} = \frac{Maximum\ DL}{PE_{STR}} + 5 [words], \tag{5}$$

여기서 'Maximum DL'은 최대 지연 길이를,  $PE_{STR}$ 는

하나의 현에 할당된 PE의 수를 나타낸다. 본 논문에서 'Maximum DL'은 535로 설정되는데, 식 (1)에서 고정된 샘플링율일 경우 가장 낮은 기본 주파수에서 최대 지연 길이를 갖는다. 기타는 'E2(82.41Hz) - A2(110.00Hz) - D3(146.83Hz) - G3(196.00Hz) - B3(246.94Hz) - E4(329.63Hz)'의 음역을 가지며 가장 낮은 기본 주파수는 82.41Hz로 음 합성을 위한 가장 긴 지연 길이는 535이다.

또한 음 합성에 필요한 파라미터인 루프 필터 계수(a1), 루프 필터 이득(g), 미소 지연 필터 계수(h0, h1)를 각 PE의 내부 메모리에 저장해야하기 때문에 추가적으로 5워드의 메모리 공간이 필요하다. 이와 더불어 시스템 전체 메모리는 MEMSYS = MEMPE×NUMPE로 정의된다. 여기서 NUMPE는 PE의 수를 의미한다.

표 1. 모델링 된 매니코어 프로세서 시스템 파라미터  
Table 1. Modeled many-core processor system parameters

Parameters	Values						
	6	12	24	48	96	192	384
Number of PEs	6	12	24	48	96	192	384
Memory / PE[words]	540	273	139	72	39	22	14
System Memory[KB]	12.96	13.11	13.35	13.83	14.98	16.90	21.51
VLSI Technology	130nm						
Clock Frequency	720MHz						
Interconnection Network	Mesh						
intALU/intMUL/Barrel Shifter/intMACC/Comm	1/1/1/1/1						

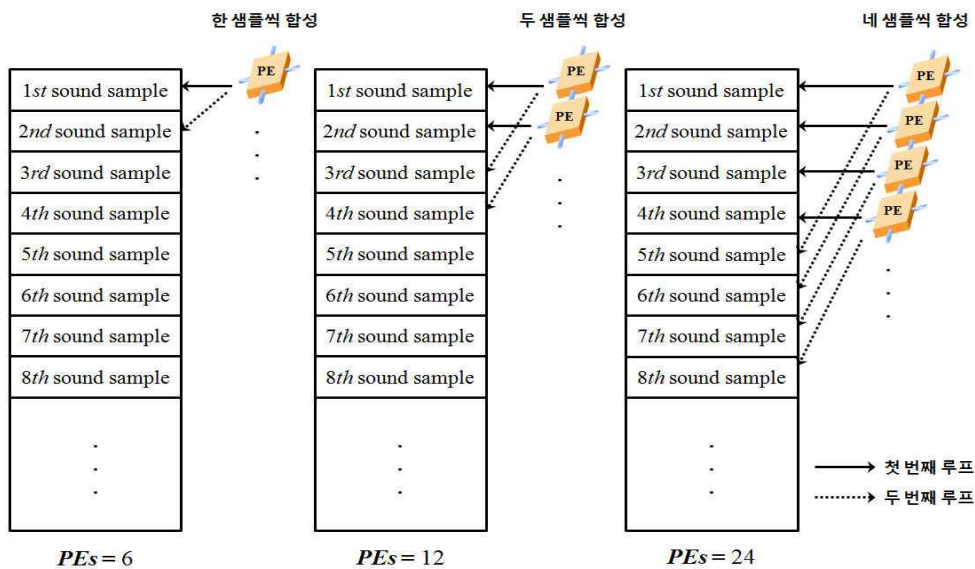


그림 5. 서로 다른 PE 구성에 따른 음 합성 예  
Fig. 5. An example of sound synthesis for various PE configurations

### 5. 음 합성 알고리즘의 병렬 구현

그림 6은 PE가 24개인 배열에서 음 합성을 위한 흐름도를 보여주며, 각 열은 하나의 기타 현을 나타낸다.

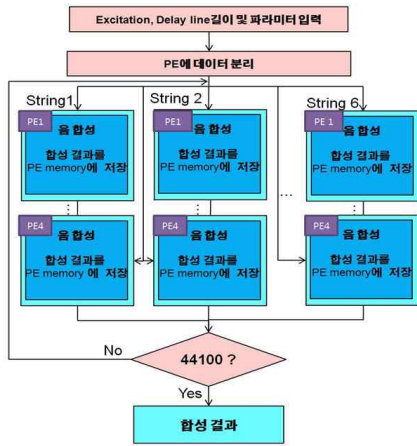


그림 6. 기타의 음 합성을 위한 병렬 구현  
Fig. 6. Parallel implementation for sound synthesis of guitar

그림 6은 하나의 기타 현에 네 개의 PE가 할당되었으며, 동시에 발음 수가 6인 기타 음 합성을 위해 각 현당 네 개씩 총 24개의 PE가 할당되었다. 음 합성을 위해 각 현에 해당하는 여기 신호와 지연 길이 및 합성에 필요한 파라미터가 각 PE에 분배된다. 각 PE는 지연 길이, 루프 필터 계수, 루프 필터 이득, 라그랑즈 보간기 계수를 이용하여 음을 합성한다. 이때, 각 PE에서 하나의 샘플을 합성하기 때문에 PE 수가 24일 때는 하나의 현에 대해 네 개의 샘플이 동시에 합성된다. 이와 같은 과정을 반복하면 해당 현의 음을 합성할 뿐만 아니라 여섯 현에 대한 기타 음을 동시에 합성할 수 있다. 그림 7은 기타 6번 현(E2, 82.41Hz)에 네 개의 PE가 할당된 경우의 음 합성 예를 보이는데, 134번째 단계에서 마지막 하나의 PE가 유휴 상태(sleep)에 놓이는 것을 확인할 수 있다. 올바른 음 합성을 위해 이웃한 PE로부터 이전에 합성된 음 샘플을 가져와야 하는데 이를 위해 PE간 통신을 하는 것보다는 해당 PE를 유휴 상태로 두는 것이 더 효율적이므로 본 논문에서는 이러한 경우가 발생할 경우 해당 PE를 유휴 상태로 둔다. 즉, 첫 번째 합성된 샘플이 PE1의 내부 메모리에 있기 때문에 PE4를 비활성화 시키고 135번째 단계에서 합성하도록 처리하였다. 이와 같이, 올바른 음 합성을 위해서는 PE를 비활성화시키는 “SLEEP” 명령어 사용이 불가피한데 이는 각 현에 할당된 PE의 수가 증가할수록 불규칙한 지연 길이(126, 168, 212, 283, 378, 505)

로 인해 더 많은 SLEEP 명령어가 사용되고, 결과적으로 시스템 이용률은 떨어지게 결과를 초래한다.

	Step 1	Step 2	Step 134	Step 135	Step 136	Step 268	Step 269	Step 270
PE 1	1	5	533	1'	5'	533'	1''	5''
PE 2	2	6	534	2'	6'	534'	2''	6''
PE 3	3	7	535	3'	7'	535'	3''	7''
PE 4	4	8		4'	8'		4''	8''

그림 7. 6번 현의 음 합성 과정(지연 길이 : 535)  
Fig. 7. A process of sound synthesis: sixth string (delay length : 535)

### III. 실험 결과

#### 1. 음 합성 결과

본 논문에서 사용된 음 합성 알고리즘은 기타의 몸통 특성을 여기 신호에 포함시켜 사용하는 물리적 모델링 합성 방법의 하나인 대체 도파관 합성(commuted waveguide synthesis)이다[5,22]. SIMD 기반 매니코어 프로세서를 이용하여 합성한 음과 원음을 청취한 결과 거의 동일하였으며, 그림 8에서 보는 바와 같이 스펙트럼 비교에서도 매우 유사함을 알 수 있다. 합성음 데모, 청취 테스트 결과 및 나머지 합성음에 대한 스펙트럼은 아래 웹페이지에서 확인할 수 있다.

[http://eucs.ulsan.ac.kr/KSCL\\_manycore](http://eucs.ulsan.ac.kr/KSCL_manycore).

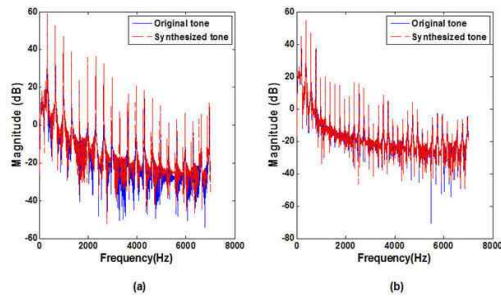


그림 8. 원음(실선)과 SIMD 기반 매니코어 프로세서(점선)를 이용한 합성음에 대한 스펙트럼: (a) 1번 현, (b) 3번 현  
Fig. 8. Spectra of the original guitar sound(solid line) and synthesized sound using the many-core processor(dotted line) : (a) 1st string, (b) 3th string

### 2. 성능 분석 지표

표 2는 모델링한 매니코어 프로세서의 성능을 평가하기 위한 세 가지 성능평가 지표를 보여준다[20]. 실행 시간(execution time)은 기타의 음 합성 병렬 알고리즘이 수행되는 시간을 나타내며, 시스템 면적 효율(area efficiency)은 단위 면적당 처리량을, 에너지 효율(energy efficiency)은 소비된 에너지당 처리량을 나타낸다.

표 2 모델링된 매니코어 프로세서의 성능평가 지표  
Table 2. Evaluation metrics for the modeled many-core processors

실행시간	$t_{exec} = C/f_{clk}$
시스템 면적 효율	$\eta_A = \frac{1}{t_{exec} \cdot Area} \left[ \frac{1}{s \cdot mm^2} \right]$
에너지 효율	$\eta_E = \frac{1}{t_{exec} \cdot Energy} \left[ \frac{1}{s \cdot Joules} \right]$

여기서 C는 사이클 수, fclk는 클럭 주파수, Area는 시스템의 단위 면적(mm<sup>2</sup>), Energy는 130nm 공정에서 소비된 에너지(Joules)를 나타낸다.

### 3. 실행시간

그림 9는 서로 다른 PE 구성에 따른 실행 시간을 보인다. 예상한 바와 같이, 하나의 기타 현에 할당된 PE의 수가 증가할수록 실행 시간이 감소함을 알 수 있다. PE가 96개 이하에서는 PE의 수가 2배로 증가할 때 실행 시간이 반으로 줄어들고 있으나, 96개 이상에서는 실행 시간의 감소가 줄어든다. 이는 PE의 수가 늘어날수록 PE간의 데이터 전송 명령어와 PE를 비활성시키는 SLEEP 명령어가 증가하기 때문이다.

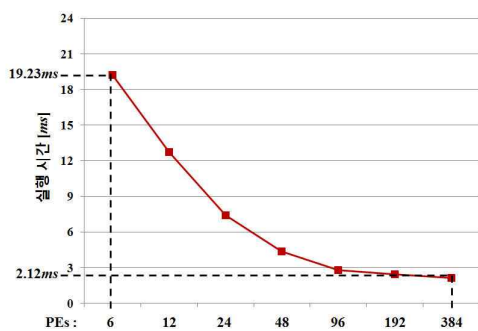


그림 9. 서로 다른 PE 구성에 따른 실행 시간  
Fig. 9. Execution time for variable PE configurations

또한 실행 시간은 합성음의 샘플링율과 밀접한 관계가 있다. CD 음질을 만족시키기 위해서는 44.1kHz의 샘플링율을 만족하여야 하는데, 이는 하나의 샘플이 0.02ms(1/44,100Hz)내에 합성되어야 함을 의미한다[21]. 그림 9에서 동시 발음 수 6인 44,100개의 샘플을 합성하는데 PE가 6개, 384개인 경우 각 19.23ms, 2.12ms가 소요되었다. 이는 하나의 샘플이 합성되는데 각 0.43μs, 0.048μs가 소요됨을 의미하므로 44.1kHz의 샘플링율을 충분히 보장할 수 있다.

### 4. PE 이용률

그림 10은 서로 다른 PE 구성에 따른 PE 이용률(utilization)을 보여준다. PE 개수가 24개 이하일 때 시스템 이용률은 계속적으로 감소하고, 24개 이상에서는 시스템 이용률이 약 88%로 유지되는 것을 알 수 있다. 이는 2.5절에서 설명한 것과 같이 불규칙적인 지연 길이를 갖는 음 합성 알고리즘의 특성으로 인해 PE를 비활성화시키는 SLEEP 명령어의 사용 증가로 인해 PE 이용률은 감소된다.

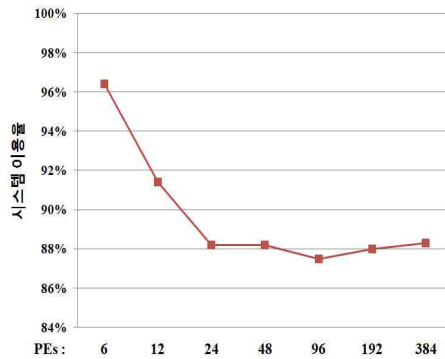


그림 10. 서로 다른 PE 구성에 따른 PE 이용률  
Fig. 10. PE utilization for variable PE configurations

### 5. 시스템 면적 효율

시스템 면적 효율은 단위 면적당 처리량이며, 그림 11은 서로 다른 PE 구성에 따른 시스템 면적 효율을 보여준다. 여기서 세로축은 시스템 면적 효율의 평균값으로 정규화한 값이므로 의미가 없으며 그래프가 나타내는 모양이 중요하다. PE 수가 24개일 때 최대의 시스템 면적 효율을 가지며, PE 수가 24개 이상일 경우는 시스템 면적 증가에 비해 실행 시간의 감소가 줄어들어서 시스템 면적 효율이 줄어든다. 실행 시간의 감소가 줄어드는 이유는 그림 10에서와 같이 PE=24 이상일 때 시스템 이용률이 줄어들기 때문이다.



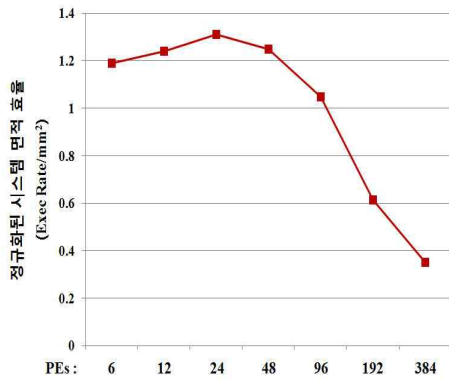


그림 11. 서로 다른 PE 구성에 따른 정규화된 시스템 면적 효율  
 Fig. 11. Normalized area efficiency for variable PE configurations

여기서 'Exec Rate'는  $1/t_{exec}$ 이다.

### 6. 에너지 효율

에너지 효율은 소비된 단위 에너지당 처리량이며, 그림 12는 서로 다른 PE 구성에 따른 에너지 효율을 보여준다. 에너지 효율은 PE 수가 96개일 때 가장 좋은 효율을 보인다. PE 수가 96개 이하에서는 PE의 수가 2배 증가 할 때 마다 실행 시간이 반으로 줄어드는 반면 거의 동일한 에너지를 소비하여 에너지 효율이 증가한다. 그러나, PE 수가 96개 이상일 때는 실행 시간 감소가 줄어드는 반면, 그에 따른 에너지 소비가 증가하여 에너지 효율이 감소된다.

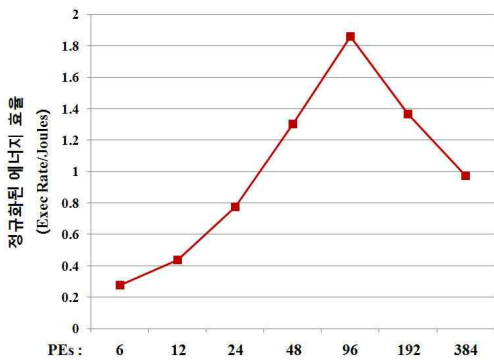


그림 12. 서로 다른 PE 구성에 따른 정규화된 에너지 효율  
 Fig. 12. Normalized energy efficiency for variable PE configurations

### 7. 상용 프로세서와의 비교

본 절에서는 최적의 매니코어 프로세서와 상용 프로세서(TI TMS320C6416, ARM 926EJ-S, ARMI0E)와의 효율을 비교한다. 상용 프로세서와의 정확한 비교는 적정하지 않지만 본 절의 목적은 상용 프로세서와의 비교를 통해 선택한 최적의 매니코어 프로세서의 효율이 어느 정도인지를 살펴보기 위한 것이다. 그림 13은 시스템 면적 효율과 에너지 효율 측면에서 상용 프로세서와의 비교 결과를 보여준다. 효율 비교 결과, 최적의 매니코어 프로세서는 음 합성에서 많이 사용되는 TI TMS320C6414보다 시스템 면적 효율에서 1,235배, 에너지 효율에서 22배의 향상을 보였다.

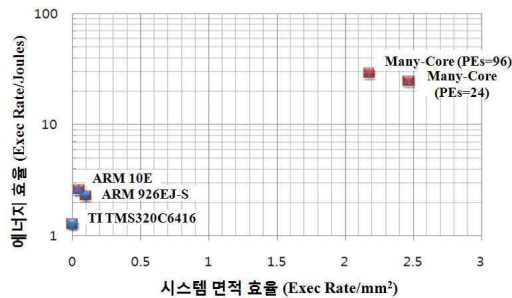


그림 13. 최적의 매니코어 프로세서와 상용 프로세서와의 효율 비교  
 Fig. 13. Efficiency comparison between the optimal many-core processors and other commercial processors

### IV. 결론

모바일 프로세서의 중요한 선택 기준은 고성능, 저전력 그리고 시스템 면적이다. 본 논문에서는 기타의 음 합성 알고리즘을 위한 최적의 SIMD 기반 매니코어 프로세서를 찾는 방법론을 제안하였다. 기타의 음 합성을 위한 최적의 매니코어 아키텍처를 찾기 위해 일곱 가지 서로 다른 PE 구성을 모델링하였고, 이러한 PE 모델들을 사용하여 샘플링율이 44.1kHz, 양자화 비트가 16인 기타 음을 합성하였다. 130nm 테크놀로지와 720MHz 클럭 주파수의 매니코어 프로세서 아키텍처를 이용하여 모의 실험한 결과, 시스템 면적 효율은 PE 수가 24개일 때, 에너지 효율은 PE 수가 96개일 때 최적의 효율을 보였다. 또한 음 합성에서 많이 사용되는 TI TMS320C6416보다 시스템 면적 효율에서 1,235배, 에너지 효율에서 22배의 향상을 보였다. 이러한 결과는 음 합성을 위한 최적의 매니코어 구조를 찾는 방법론에 대한 해결책으로 사용될 수 있다.



## 참고문헌

- [1] T. Agerwala, and S. Chatterjee, "Computer architecture: challenges and opportunities for the next decade," *IEEE Micro*, pp. 58-69, May-June 2005.
- [2] S.-H. Kim, S.-B. Nam, H.-J. Lim, "An improved area edge detection for real-time image processing," *Journal of the Korea Society of Computer and Information*, Vol. 14, No. 1, pp. 99-106, Jan. 2009.
- [3] A. W. Y. Su, W.-C. Chang, and R.-W. Wang, "An IIR synthesis method for plucked-string instruments with embedded portamento," *J. Audio Eng. Soc.*, Vol. 50, No. 5, pp. 351-362, May. 2002.
- [4] V. Valimäki, J. Huopaniemi, M. Karjalainen, and Z. Janosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc.*, Vol. 44, No. 5, pp. 331-353, Mar. 1996.
- [5] M. Karjalainen, J. Backman, and J. Polkki, "Analysis, modeling, and real-time sound synthesis of the kantele, a traditional finnish string instrument," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 229-232, Apr. 1993.
- [6] Sang-Jin Cho, Ui-Pil Chong, and Sang-Bock Cho, "Synthesis of the Dan Tranh Based on a Parameter Extraction System," *J. Audio Eng. Soc.*, Vol. 58, No. 6, pp. 498-507, June 2010.
- [7] Sang-Jin Cho, and Ui-Pil Chong, "Physical Modeling of a Sanjo Gayageum," *J. Acoustical Soc. Korea*, Vol. 23, No. 7, pp. 521-531, Oct. 2004.
- [8] Sangjin Cho, and Uipil Chong, "Sound Synthesis of Right-Hand Playing Styles Using Improved Physical Modeling of Sanjo Gayageum," *J. Acoustical Soc. Korea*, Vol. 25, No. 8, pp. 325-332, Nov. 2006.
- [9] Junehee Yoo, "Vibrational Modes of Pyeongeong," *J. Acoustical Soc. Korea*, Vol. 25, No. 3, pp. 21-28, Apr. 2006.
- [10] Myeongsu Kang, Sangjin Cho, and Uipil Chong, "Implementation of Non-Stringed Guitar Based on Physical Modeling Synthesis," *J. Acoustical Soc. Korea*, Vol. 28, No. 2, pp. 119-126, Feb. 2009.
- [11] H. G. Alles, "Music synthesis using real time digital techniques," *Proc. IEEE*, Vol. 68, No. 4, pp. 436-449, Apr. 1980.
- [12] Myeongsu Kang, Sangjin Cho, Sundeok Kwon, and Uipil Chong, "Sound Engine for Korean Traditional Instruments Using General Purpose Digital Signal Processor," *J. Acoustical Soc. Korea*, Vol. 28, No. 3, pp. 229-238, Apr. 2009.
- [13] Ui-Pil Chong and Sang-Jin Cho, "Physical modeling of gayageum with application to sound engine in musical synthesizer," in *Proc. Int. Conf. High Performance Scientific Computing*, Nov. 2006.
- [14] S.-H. Lee, "The Design and implementation of parallel processing system using the Nios(R)II embedded processor," *Journal of the Korea Society of Computer and Information*, Vol. 14, No. 11, pp. 97-103, Nov. 2009.
- [15] L. V. Huynh, C.-H. Kim, J.-M. Kim, "A massively parallel algorithm for fuzzy vector quantization," *Journal of Korea Information Processing Society*, Vol. 16-A, No. 6, pp. 411-418, Dec. 2009.
- [16] Jongmyon Kim, Yong-Min Kim, Cheol-Hong Kim, "Performance Evaluation of Multimedia Extensions on Variable Many-Core Processors," in the *International Conference on Computer Design (CDES'10)*, pp. 98-104, Las Vegas, USA, 12-15, July 2010.
- [17] A. Gentile, D. S. Wills, "Impact of pixel per processor ratio on embedded SIMD architectures," *Image Analysis and Processing*, pp. 204-208, Sept. 2001.
- [18] S. M. Chai, T. Taha, D. S. Wills, J. D. Meindl, "Heterogeneous Architecture Models for Interconnect-Motivated System Design," *IEEE Trans. on VLSI Systems*, Vol. 8, No. 6, pp. 660-670, Dec. 2000.
- [19] S. Nugent, D. S. Wills, and J. D. Meindl, "A Hierarchical Block-based Modeling Methodology for SoC in GENESYS," *15th Annual IEEE Int. AISC/SOC Conf.*, pp. 239-243, Sept. 2002.
- [20] A. Gentile, S. Sander, L. Wills, and D. S. Wills, "The Impact of Grain Size of the Efficiency of

Embedded SIMD Image Processing Architectures,"  
Journal of Parallel Distributed Computing, Vol.  
64, pp. 1318-1327, Nov. 2004.

[21] Sampling rate, available at  
[http://en.wikipedia.org/wiki/Sampling\\_rate](http://en.wikipedia.org/wiki/Sampling_rate)

## 저 자 소 개



### 최 지 원

2009 : 울산대학교 정보통신공학사  
2010 : 울산대학교 컴퓨터정보통신  
공학과 석사  
관심분야 : 임베디드시스템, 음향신  
호처리, 병렬처리  
Email: regular912@nate.com



### 강 명 수

2008 : 울산대학교 컴퓨터공학과  
학사  
2010 : 울산대학교 컴퓨터공학과  
공학석사  
2010~현재 : 울산대학교 컴퓨터공  
학과 박사과정  
관심분야 : 임베디드 시스템, 음향  
신호처리, 고장진단  
Email: ilmareboy@ulsan.ac.kr



### 김 종 먼

1995 : 명지대학교 전기공학사  
2000 : University of Florida  
ECE 석사  
2005 : Georgia Institute of  
Technology ECE 박사  
2005 - 2007 : 삼성종합기술원 전  
문연구원  
2007 - 현재 : 울산대학교 컴퓨터  
정보통신공학부 교수  
관심분야 : 프로세서 설계, 임베디  
드 SoC, 컴퓨터구조,  
병렬처리  
Email: jongmyon.kim@gmail.com