

기타의 음 합성을 위한 병렬 프로세서 구현

Implementation of Parallel Processor for Sound Synthesis of Guitar

최 자원*, 김 용 민*, 조 상 진**, 김 종 면*, 정 의 필*
(Ji-Won Choi*, Yong-Min Kim*, Sang-Jin Cho**, Jong-Myon Kim*, Ui-Pil Chong*)

*울산대학교 컴퓨터정보통신공학부, **울산대학교 전기전자정보시스템공학부

(접수일자: 2010년 2월 8일; 채택일자: 2010년 3월 23일)

물리적 모델링은 실제 악기음과 유사한 고음질의 음을 합성하는 방법이다. 그러나 물리적 모델링은 악기의 소리를 합성할 때 필요한 수많은 파라미터들을 동시에 계산해야 하기 때문에 동시 발음수가 높은 악기의 경우 실시간 처리에 문제가 발생할 수 있다. 이러한 문제를 해결하기 위해 본 논문에서는 기타의 음 합성 알고리즘을 실시간으로 처리 가능한 단일 명령어 다중 데이터 (Single Instruction Multiple Data, SIMD) 처리 방식의 병렬 프로세서를 제안한다. 대표적인 현악기인 기타의 6개 현을 제어하기 위해 6개의 프로세싱 엘리먼트 (Processing Element, PE)로 구성된 SIMD 기반 병렬 프로세서를 사용하였다. 각각의 프로세싱 엘리먼트는 해당되는 기타 현을 모델링하며, 각 현의 여기신호와 파라미터를 합성 병렬 알고리즘의 입력으로 받아 동시에 6개 현의 합성된 음을 실시간으로 생성할 수 있다. 표본화 비율을 44.1 kHz로 설정하고 16비트 양자화 데이터의 음을 합성한 모의 실험 결과, 제안한 SIMD 기반 병렬 프로세서를 이용한 합성음은 원음과 매우 유사하였으며, 상용 프로세서인 TI사의 TMS320C6416보다 실행 시간에서 8.9배, 에너지 효율에서 39.8배의 성능 향상을 보였다.

핵심용어: 음 합성, 물리적 모델링, 기타, SIMD 병렬 프로세서, 디지털 도파관

투고분야: 음향 신호처리 분야 (1,3)

Physical modeling is a synthesis method of high quality sound which is similar to real sound for musical instruments. However, since physical modeling requires a lot of parameters to synthesize sound of a musical instrument, it prevents real-time processing for the musical instrument which supports a large number of sounds simultaneously. To solve this problem, this paper proposes a single instruction multiple data (SIMD) parallel processor that supports real-time processing of sound synthesis of guitar, a representative plucked string musical instrument. To control six strings of guitar, we used a SIMD parallel processor which consists of six processing elements (PEs). Each PE supports modeling of the corresponding string. The proposed SIMD processor can generate synthesized sounds of six strings simultaneously when a parallel synthesis algorithm receives excitation signals and parameters of each string as an input. Experimental results using a sampling rate 44.1 kHz and 16 bits quantization indicate that synthesis sounds using the proposed parallel processor were very similar to original sound. In addition, the proposed parallel processor outperforms commercial TI's TMS320C6416 in terms of execution time (8.9x better) and energy efficiency (39.8x better).

Keywords: Sound synthesis, physical modeling, guitar, SIMD parallel processor, digital waveguide

ASK subject classification: Acoustic Signal Processing (1,3)

I. 서론

악기 음 합성을 위해 물리적 모델링 (physical modeling) 방식의 연구가 활발히 진행되어 오고 있다 [1]-[12].

물리적 모델링은 악기나 악기 내부의 파동을 분석하여 설계한 디지털 필터로 음을 합성하는 방식이다 [2]. 따라서 디지털 필터의 파라미터를 조합함으로써 음색을 조절하고 특정필터를 추가 및 제거함으로써 연주방식을 표현할 수 있다. 물리적 모델링은 대부분 디지털 도파관 (digital waveguide)을 이용한 방식을 사용하여 악기의 소리 발생 원리를 분석하고 그에 적합한 모델을 만들어

음을 합성한다 [2],[3].

A. Su는 중국의 구친 (Gu-Chin)이라는 현악기 모델링에 대해 연구하였고 [9], V. Valimaki는 칸텔레 (Kantele)라는 핀란드 전통악기를 모델링하였다 [4],[10]. 국내에서는 산조 가야금의 물리적 모델링에 대한 연구 [5],[8], 편성에 대한 진동모드 분석 [11] 및 물리적 모델링 합성법을 이용한 줄 없는 기타 구현 [7] 등이 있다. 그러나 손으로 현을 뜯거나 튕겨서 연주하는 현악기의 음 합성 알고리즘은 상당한 양의 연산과 입출력 처리를 요구하기 때문에 실시간으로 사용되는 애플리케이션에 적용하기에는 많은 어려움이 있다 [3],[4]. 특히 현악기 사운드 엔진 개발의 경우 기존 프로세서의 성능 제약으로 인하여 고음질의 악기 음 합성 알고리즘을 구현하기 어려운 실정이다 [12].

ASIC (Application-Specific Integrated Circuit)은 이러한 음 합성 알고리즘에서 요구되는 고 성능, 저전력을 충족시킬 수 있지만 다양한 애플리케이션에서 요구되는 프로그램 능력 (programmability)과 유연성 (flexibility)을 만족시키지 못한다. 반면에 범용 프로세서 (General-Purpose Processor, GPP)나 DSP (Digital Signal Processor)들은 다양한 애플리케이션에 대해 충분한 프로그램 능력과 유연성을 제공한다. 하지만, 멀티미디어 애플리케이션에서 요구되는 높은 레벨의 성능을 만족시키지 못한다. 왜냐하면 GPP나 DSP는 프로세서 구조의 특성상 멀티미디어에 내재한 고도 병렬성 (massive parallelism)을 활용하지 못하기 때문이다.

음 합성 애플리케이션의 성능을 충족시킬 수 있는 프로세서 모델 중에서 SIMD기반 병렬 프로세서 아키텍처가 대안으로 부각되고 있다. 명령어 레벨 (Instruction-level)이나 스레드 레벨 (thread-level) 프로세서들은 실리콘 변적을 멀티포트 레지스터 파일 (multiported register file), 캐시 (cache), 파이프라인 (deep pipelined) 기능 유닛 등으로 사용하는 반면, SIMD기반 병렬 프로세서는 수천 개의 저비용 프로세싱 엘리먼트 (processing element, PE)들을 이용하여 고성능을 추구하고 동시에 저장장소와 데이터 통신 요구를 최소화하기 위해 프로세싱 엘리먼트와 데이터 입출력을 동일위치에 배치함으로써 저 전력을 만족시킨다.

본 논문에서는 대표적 현악기인 기타의 물리적 모델링 합성법을 이용하여 실시간으로 음 합성을 하기 위해 SIMD기반 병렬 프로세서 아키텍처를 제안한다. 제안한 SIMD기반 병렬 프로세서는 6개의 프로세싱 엘리먼트 (processing element, PE)로 구성되어 있으며, 각각의 프

로세싱 엘리먼트는 6줄로 구성된 기타의 해당되는 현을 모델링한다. 따라서, 각 현의 여기신호와 파라미터를 음 합성 병렬 알고리즘의 입력으로 받아 동시에 6개 현의 합성된 음을 실시간으로 생성할 수 있다. 표본화 비율을 44.1 kHz로 설정하고 16비트 양자화 데이터의 음을 합성한 실험 결과, 제안한 SIMD기반 병렬 프로세서를 이용한 합성음은 원음과 매우 유사하였다. 또한, 현재 상용화되고 있는 고성능 DSP인 TI사의 TMS320C6416과 성능을 비교한 결과, 제안한 SIMD기반 병렬 프로세서는 C6416보다 8.9배의 성능 향상 및 39.8배의 에너지 효율 향상을 보였다. 이러한 결과는 현악기를 포함한 다양한 악기의 사운드 엔진 개발에 있어서 제안한 병렬 프로세서의 무한한 잠재 가능성을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 디지털 도파관을 이용한 현악기의 물리적 모델링을 소개하고, 3장에서는 제안한 SIMD기반 병렬 프로세서 모델과 음 합성 병렬 알고리즘을 소개한다. 그리고 4장에서는 SIMD기반 병렬 프로세서를 이용한 기타의 음 합성 결과를 확인하고, 상용 프로세서인 C6416과 성능 및 에너지 효율을 비교 및 분석한다. 끝으로 6장에서는 이 논문의 결론을 맺는다.

II. 디지털 도파관을 이용한 현악기의 물리적 모델링

현악기의 물리적 모델링에 대하여 많은 연구가 진행되어 왔다 [4]–[12]. J. O. Smith가 제안한 디지털 도파관 모델링은 악기의 파동을 모델링하는 방식으로, 현악기의 물리적 모델링에 가장 일반적으로 사용된다 [2]. 모든 악기는 공기의 파동 및 악기 자체의 파동을 가지게 된다. 예를 들어 관악기는 관 속에서 좌우로 움직이는 파동을 가지게 되고, 현악기에서는 현에서 좌우로 움직이는 파동을 가지게 된다. 이러한 파동은 특정방향 (x - y 축)만 따라 움직이게 되고, 이렇게 한 방향을 따라서 움직일 수 있도록 하는 악기의 물리적 구조를 디지털 도파관이라고 한다. 디지털 도파관 모델은 이러한 물리적 구조를 따라 움직이는 파동을 모델링한 것을 말한다.

디지털 도파관 합성에서 줄의 진동과 몸체의 방사에 대한 블록 다이어그램은 그림 1과 같다. 여기신호 (excitation signal) $x(n)$ 은 몸체의 방사 특성을 포함하고 있고, 웨이브 테이블 (wave table)에 저장된다. 지연 라인 (delay line) 길이는 표본화 비율 f_s 와 기본 주파수

f_0 의 비 ($L = f_s/f_0$)로 계산되며, $L = L_i + L_f$ 이다. L_f 는 지연 라인의 정수부, L_i 는 실수부이다. 그리고 지연라인의 길이는 파동의 속도에 따라 가변적이다. 미소지연필터 (fractional delay filter) $F(z)$ 는 미소지연 L_f 를 보상하기 위한 것으로, 본 논문에서는 식 (1)과 같은 1차 라그랑주 보간기를 사용하였다.

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

루프필터 (loop filter) $H_l(z)$ 는 현의 진동이 감쇄되는 현상을 표현하기 위한 것으로, 저역통과필터 (low-pass filter)의 특성을 가지며 식 (2)와 같이 나타낼 수 있다 [4].

$$H_l(z) = \frac{g(1 + a_1)}{1 + a_1 z^{-1}} \tag{2}$$

최종적으로 도파관 모델을 이용한 기타의 전달함수는 식 (3)과 같다.

$$S(z) = \frac{1}{1 - z^{L_i} F(z) H_l(z)} \tag{3}$$

III. SIMD 기반 병렬 프로세서와 기타 음 합성 병렬 알고리즘

현 악기는 여러 줄로 구성되어 있어 이를 위한 모델링을 구현하기 위해서는 상당한 계산량이 필요한데, 이를 해결하기 위한 방법으로 본 논문에서는 SIMD 기반 병렬 프로세서를 제안한다.

SIMD기반 병렬 프로세서 아키텍처의 블록 다이어그램은 그림 2와 같다. SIMD 병렬 프로세서는 여러 개의 프로세싱 엘리먼트와 이를 제어하는 Array Control Unit (ACU)으로 구성된다. 데이터가 각 프로세싱 엘리먼트들에 일정하게 분배되면 프로세싱 엘리먼트들은 매쉬 배열

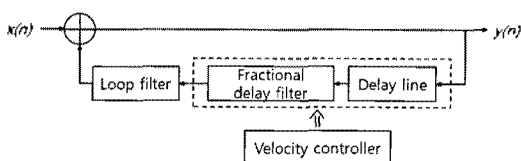


그림 1. 현악기 모델의 블록 다이어그램
Fig. 1. A block diagram of a plucked-string instrument model.

구조에서 명령어들을 수행한다. 본 논문에서는 기타의 6개 현을 모델링하기 위해 6개의 프로세싱 엘리먼트를 매쉬 구조로 연결하였다. 각 프로세싱 엘리먼트는 다음과 같은 특징을 가진다.

- 32비트 폭의 512개 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 (Barrel Shifter)
- 지역 정보를 이용해 각 PE들을 환성 및 비환성 시키는 Sleep 유닛
- 이웃하는 프로세싱 엘리먼트들과 데이터 통신을 위한 NEWS (north-east-west-south) 네트워크 및 serial I/O유닛

기타의 음 합성을 위한 병렬 알고리즘은 그림 3과 같다. 기타의 음 합성을 위해 먼저 기타의 연주 방법과 각 줄에 해당하는 여기 신호와 파라미터들을 입력으로 넣어 준다. ACU (Array control unit)는 입력 받은 데이터들을 각 줄에 맞게 6개의 프로세싱 엘리먼트에 분배한다. 프로세싱 엘리먼트는 각 연주 방법에 따른 파동의 상대적 이동속도 (velocity)를 계산하여 지연 라인의 길이를 결정하고, 세 개의 파라미터인 a_l (루프 필터 계수), g (루프 필터의 이득) 그리고 h_0, h_l (라그랑주 보간기의 필터 계수)를 이용하여 음을 합성한다. 음 합성 과정의 반복을 통해 각각의 프로세싱 엘리먼트는 해당되는 현의 음을 합성할 뿐만 아니라 6줄의 기타 음을 동시에 합성할 수 있다.

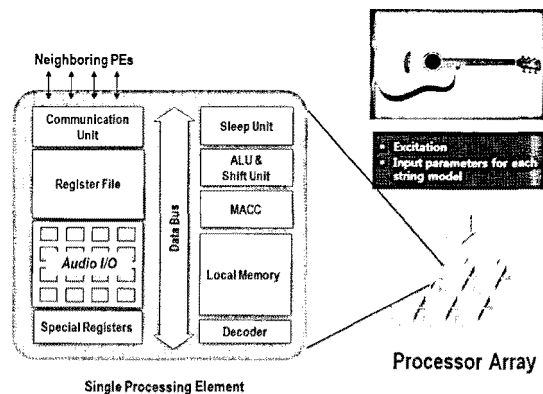


그림 2. SIMD기반 병렬 프로세서 아키텍처와 싱글 프로세싱 엘리먼트
Fig. 2. A block diagram of a SIMD processor array and Single processing element.

제한한 SIMD기반 병렬프로세서는 저비용, 저전력 및 고성능을 만족시키기 위해 정수 연산을 지원한다. 따라서 음 합성 알고리즘에서 사용되는 실수 연산 (floating point operations)은 Q-format을 사용하여 정수로 변환하여 처리하였다 [13]. 기타 음 합성 위한 알고리즘의 주요 연산 부분을 표 1에 나타내었다. SIMD기반 병렬프로세서에서 지원하는 정수 명령어를 사용하여 합성을 알고리즘에 사용되는 실수 연산을 수행하기 위해 음 합성에

사용한 여기 신호 및 필터 파라미터에 2^6 을 곱하여 (쉬프트 명령어 사용) 16비트 데이터 값으로 만들었다. 표 1의 C언어 소스의 $g*(1+a)*Dline[index]$ 의 결과가 48비트 데이터가 되기 때문에, 64비트 곱셈 및 누산기를 사용하여 누산기의 상위 32비트만 가져온다. 결과 값에서는 쉬프트 명령어를 사용하여 2^6 을 나누어 이를 보정하였다.

표 1에서 보여준 음 합성 부분을 SIMD기반 프로세서와 C6416을 사용하여 수행한 결과에 대한 성능 비교는 표 2와 같다. 동일한 연산 처리에서 제한한 SIMD기반 병렬 프로세서는 C6416보다 14배의 성능 향상을 보였다. C6416의 경우, 프로그램과 데이터를 internal RAM에 저장하고 컴파일 옵션을 -o3로 수행한 결과이다.

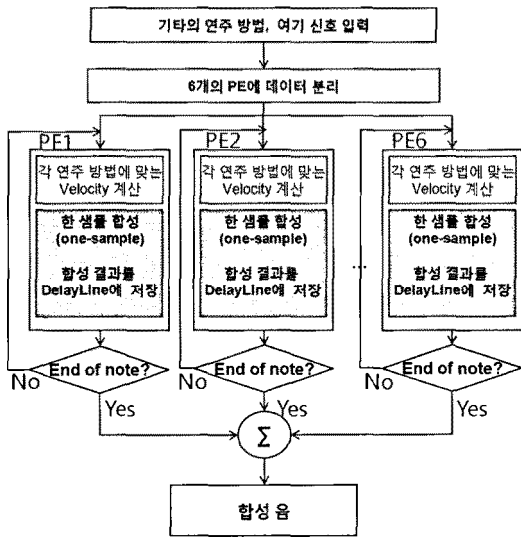


그림 3. 기타의 음 합성을 위한 병렬 알고리즘
Fig. 3. A parallel algorithm for synthesis of guitar.

IV. 모의실험 및 성능분석

4.1. 음 합성 결과

본 논문에서 사용된 음 합성 알고리즘은 기타의 몸통 특성을 여기 신호에 포함시켜 사용하는 대체 합성법 (commuted synthesis)을 기반으로 구현하였다 [7],[14]. SIMD기반 병렬프로세서를 이용하여 합성한 음과 원음을 청취한 결과 거의 동일하였으며, 그림 4에서 보는 바와 같이 스펙트럼 비교에서도 매우 유사함을 알 수 있다.

슬라이딩, 해머링 등과 같은 기타 솔로 연주법은 [7]에서 제안한 방법을 이용하였으며 합성 결과는 그림 5와

표 1. 음 합성 알고리즘의 주요 부분: C6416의 C언어 코드와 SIMD 병렬프로세서기반 어셈블리 언어 코드

Table 1. An example of the sound synthesis algorithm: C language code of C6416 and assembly language code of the SIMD parallel processor.

언어	소스 코드
C 언어 (C6416)	<pre>int a, g, Lbuf; int SynthesisOUT[], Excitation[], Dline[]; SynthesisOUT[n] = Excitation[m-1] + g*(1+a)*Dline[index] - a*Lbuf;</pre>
SIMD 병렬 프로세서 기반 어셈블리 언어	<pre>loadi a // -a값을 가져온다 ashi R12,R10,16 // R12 = Lbuf * 2^16 // 곱셈 및 누산기사용을 위해 2^16 을 쉬프트 한다. macc R15,R0,R12 // accumulator = - (a * 2^16) * (Lbuf * 2^16) // 곱셈 후 누산기에 저장한다. loadi b // {g * (1+a)} * 2^16 값을 가져온다. ashi R12,R11,16 // R12 = Dline[index] * 2^16 // 곱셈 및 누산기 사용을 위해 2^16을 쉬프트 한다 macc R15,R0,R12 // accumulator = b*Dline[index]* 2^32 + a*Lbuf * 2^32 // 곱셈 후 누산기의 데이터와 더한다. loadi Vd_start // Ro = 2 ^ 16 // 곱셈 및 누산기 사용을 위해 2^16을 쉬프트 한다. macc R15,R4,R0 // accumulator = b*Dline[index]* 2^32 + a*Lbuf * // 2^32 + Excitation * 2 ^ 32 // 곱셈 후 누산기의 데이터와 더한다 laoch R15 // R15 = SynthesisOUT(상위 32비트) // 누산기로부터 상위 32비트를 가져와서 // 한 샘플의 합성음을 r15로 저장한다.</pre>

같다. SIMD 기반 병렬 프로세서를 합성한 음의 스펙트럼은 [7]의 결과와 매우 유사하였는데, 이는 기타의 음 합성 병렬 알고리즘이 기타의 각 연주법을 잘 표현할 수 있음을 보여주는 것이다.

4.2. 실험 방법론 구조

세 가지 레벨 (애플리케이션, 아키텍처, 테크놀로지)로 구성되어 있는 SIMD 병렬 프로세서의 실험 방법론은 그림 6과 같다. 애플리케이션 레벨에서는 명령어 레벨의 SIMD 병렬 프로세서용 정밀 사이클 시뮬레이터를 이용하여, 기타의 음 합성 알고리즘에 사용되는 사이클 개수, 동적 명령어 빈도, 프로세싱 엘리먼트 이용률 (utilization) 등의 실험 데이터를 제공한다. 아키텍처 레벨에서는 보델링된 아키텍처의 디자인 변수들을 계산하기 위해 Chai가 제안한 SIMD 병렬 프로세서용 이중 아키텍처 모델링 툴을 사용하였다 [15]. 테크놀로지 레벨에서는 각 아키텍처 모델들의 테크놀로지 변수 (latency, power, clock frequency)를 계산하기 위해 Generic System Simulator (GENESYS)를 사용하였다 [16]. 마지막으로 세 레벨에서 구해진 데이터베이스를 조합하여 각 경우에 대한 실행시간, 처리량, 에너지 효율을 결정하였다.

4.3. 성능 분석 지표

본 논문에서는 기타의 음 합성 알고리즘을 구현하기 위해 SIMD 병렬 프로세서용 정밀 사이클 (cycle-accurate) 시뮬레이터를 이용하여 6개의 프로세싱 엘리먼트로 구성된 아키텍처를 구현하였으며, 피라미터는 표 3과 같다. C6416과 동일한 130 nm 테크놀로지와 720 MHz 클럭 주파수를 사용하였고, 기타의 음 합성에서 지연라인 길이 (최대 505)를 고려하여 각 프로세싱 엘리먼트는 32비트 워드 단위의 512개의 메모리를 사용하였다.

제안한 병렬 프로세서의 성능을 평가하기 위해 표 4와 같이 3가지 지표를 사용하였다 [17]. 실행 시간 (execution time)은 기타의 음 합성 알고리즘이 수행되는 시간을, 처리량 (sustained throughput)은 단위 시간당 처리되는 명령어 개수 (Giga-operations/second)를, 에너지 효율 (energy efficiency)은 단위 에너지당 소비된 명령어 개수 (Giga-operations/Joule)를 나타낸다.

4.4. 성능 평가 결과

본 논문에서 공정한 성능 평가를 위해 제안한 병렬 프로세서와 C6416을 동일한 130 nm 테크놀로지와 720 MHz 클럭 주파수를 사용하여 모의 실험하였다. 제안한

표 2. SIMD기반 병렬 프로세서와 C6416의 음 합성 알고리즘에서 주요 연산의 성능 비교
Table 2. Performance comparison of the sound synthesis algorithm using SIMD parallel processor and C6416.

	Units	SIMD130	C6416			
			internal RAM (프로그램, 데이터)		external SDRAM (프로그램, 데이터)	
			컴파일 옵션 -o3	컴파일옵션 -o0	컴파일 옵션 -o3	컴파일옵션 -o0
1 sample	[cycles]	9	125	270	848	4,936
1 second (44100 samples)	[cycles]	396,900	5,512,250	11,907,000	37,396,800	217,677,600

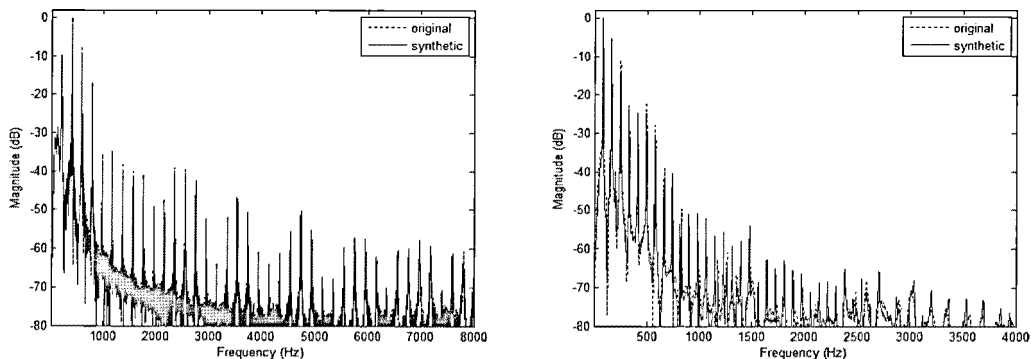


그림 4. 원음 (점선)과 SIMD기반 병렬프로세서 (실선)를 이용한 합성음에 대한 스펙트럼: (a) 3번 현, (b) 6번 현
Fig. 4. Spectra of the original guitar sound (dotted line) and synthesized guitar sound using the SIMD parallel processor (solid line): (a) 3rd string, (b) 6th string.

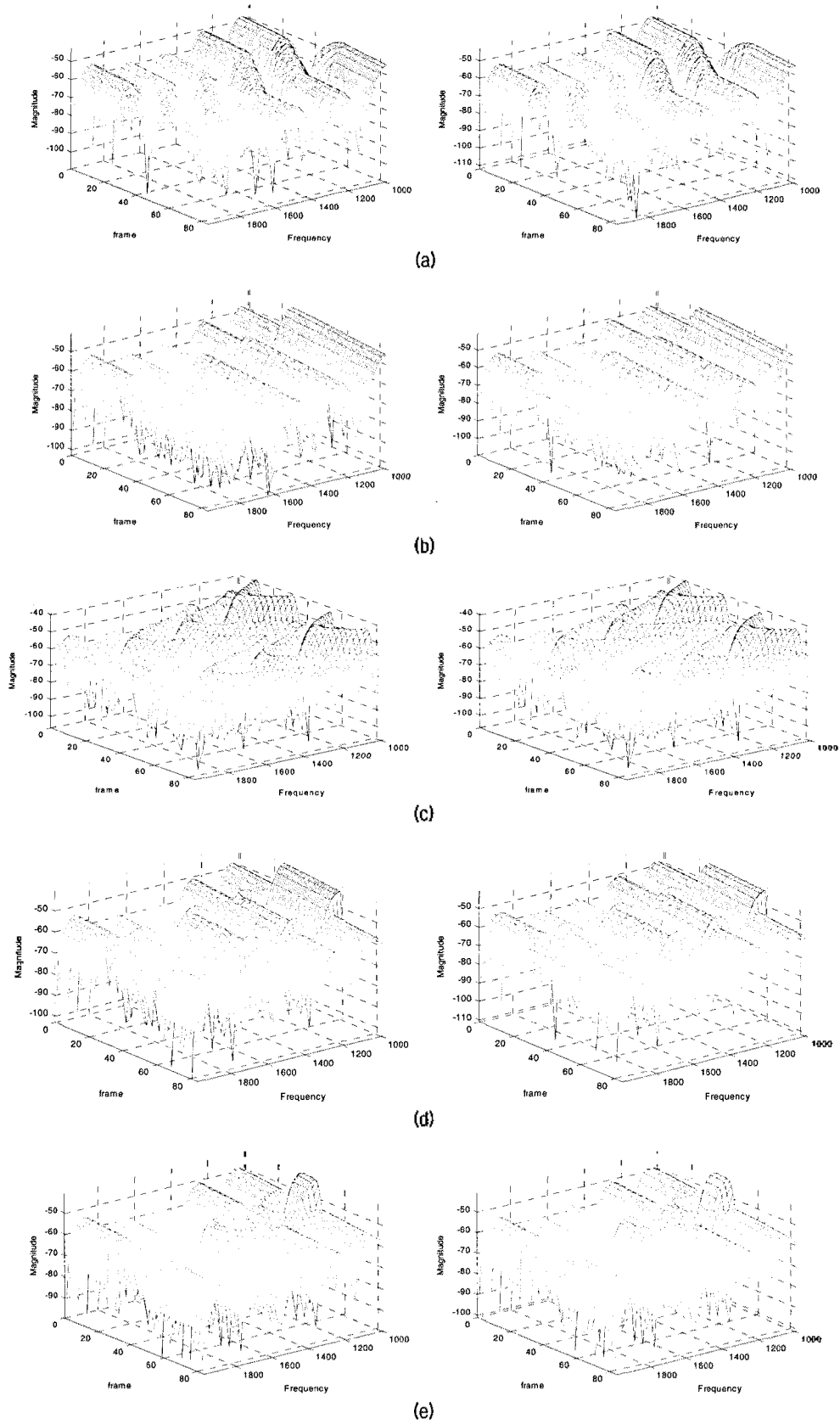


그림 5. [7]의 방법(좌)과 SIMD기반 병렬프로세서(우)를 이용한 5가지 연주법의 합성음에 대한 스펙트럼: (a) 슬라이딩, (b) 해머링 온, (c) 비브라토, (d) 해머링 온 풀링 오프, (e) 밴드 앤 릴리즈
 Fig. 5. Spectra of synthesized sounds by using method of [7] (left) and SIMD parallel processor (right): (a) sliding, (b) hammering on, (c) vibrato, (d) hammering-on and pulling-off, (e) bend and release.

프로세서는 6개의 프로세싱 엘리먼트 (PE)를 사용하여 데이터 레벨 병렬성 (data-level parallelism)을 추구하는 반면, C6416은 8-way VLIW 아키텍처로서 8개의 명령어를 동시에 처리할 수 있는 명령어 레벨 병렬성 (instruction-level parallelism)을 추구한다. 본 논문에는

서는 기존의 고성능 프로세서인 C6416과의 성능 비교를 통해 제안하는 병렬 프로세서의 잠재 가능성을 보여주고자 한다.

6개의 프로세싱 엘리먼트 시스템을 이용하여 각 연주법을 수행한 결과에 대한 전체 성능은 표 5와 같다. 기타의 연주법을 표현하기 위해 피치 변화 코드 부분을 포함시켰고, 이러한 이유로 피치가 변하는 연주법은 기본음보다 실행 시간에서 1.8배 증가 하였으며 처리량에서는 0.8배 감소하였다. 이는 기타의 연주법을 표현하기 위한 피치 변화가 성능에 영향을 미치고 있음을 보여주는 것이다.

기타의 음 합성 알고리즘에 대한 6개의 프로세싱 엘리먼트 시스템과 C6416의 성능 비교는 표 6과 같다. 성능 평가는 알고리즘 실행에서 가장 많은 연산량을 요구하는 기타 6개 현을 동시에 튕겼을때, 즉 동시 발음수가 6인 경우에서 시행하였다. 동일한 130 nm 테크놀로지와 720 MHz 클럭 주파수에서 실험한 결과, 제안한 병렬 프로세서는 C6416 (최대 성능을 위해 프로그램 코드와 데이터를 내부 메모리 영역에 적재하고, 컴파일러 옵션을 -o3으로 셋팅)보다 8.9배의 성능 향상을 보여주며, 실시간 음 합성이 가능함을 보여준다 (< 30 ms).

SIMD기반 병렬 프로세서와 C6416를 에너지 효율과 실행 시간 측면에서 비교는 그림 7과 같다. 제안한 병렬 프로세서는 C6416보다 에너지 효율에서 약 39.8배 이상의 향상을 보이준다. 에너지 효율의 증가는 시스템의 배터리 수명을 증가시키는 결과를 가져온다.

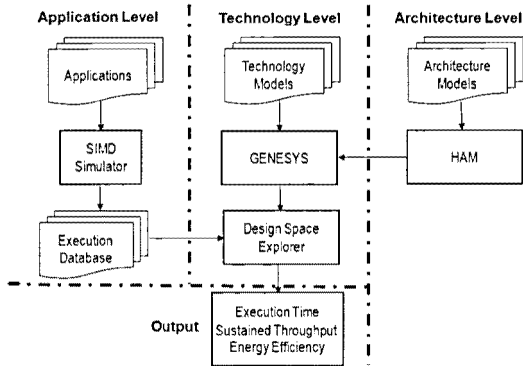


그림 6. SIMD기반 병렬 프로세서 시뮬레이션을 위한 실험 방법론 구조

Fig. 6. A methodology infrastructure for the SIMD parallel processor simulation.

표 3. 구현된 프로세싱 엘리먼트 시스템 파라미터
Table 3. Implemented PE system parameters.

parameter	value
Number of PEs	6
Memory/PE[word]	512[32-bit word]
VLSI Technology	130 nm
Clock Frequency	720 MHz
intALU/ intMUL/ Barrel Shifter/ intMACC	1 / 1 / 1 / 1

표 4. 평가 지표 요약

Table 4. Summary of evaluation metrics.

실행시간 (execution time)	처리량 (sustained throughput)	에너지 효율 (energy efficiency)
$t_{exc} = \frac{C}{f_{clk}}$	$Tb_{sust} = \frac{O_{exc} \cdot U \cdot N_{PE}}{t_{exc}}$	$\eta_E = \frac{O_{exc} \cdot U \cdot N_{PE} [Gops]}{Energy [Joule]}$

C : 사이클 개수, f_{clk} : 클럭 주파수, O_{exc} : 수행된 연산 개수
U : 프로세싱 엘리먼트 이용률, N_{PE} : 프로세싱 엘리먼트의 개수

표 5. 6개의 프로세싱 엘리먼트 시스템에서 기타의 각 연주법에 대한 성능 결과

Table 5. Performance results of each performance method of guitar on a 6 PE system.

연주방법 [6현]	Total Cycle [cycles]	Vector Instruction	Scalar Instruction	System Utilization [%]	Execution Time [ms]	Sustained Throughput [Gops/sec]
none	2,810,129	1,947,980	862,149	90.29	3.90	2.70
Sliding	3,603,929	2,521,280	1,082,649	86.90	5.01	2.63
Hammering on	3,515,729	2,256,680	1,259,049	90.45	4.49	2.51
Vibrato	3,793,229	2,036,180	1,757,049	90.72	5.27	2.10
Hammering on pulling off	4,750,529	3,094,580	1,655,949	82.92	6.60	2.33
Bend and release	4,926,929	3,138,680	1,788,249	82.17	6.84	2.26

표 6. SIMD와 C6416의 성능 비교

Table 6. Performance comparison between SIMD and C6416.

parameter	Units	SIMD130	TI C6416			
			internal RAM (프로그램, 데이터)		external SDRAM (프로그램, 데이터)	
			컴파일 옵션 -o3	컴파일 옵션 -o0	컴파일 옵션 -o3	컴파일 옵션 -o0
Technology	[nm]	130	130	130	130	130
Clock Frequency	[MHz]	720	720	720	720	720
Execution Time	[ms]	3.90	34.54	46.39	2,553.81	2,934.87
Average throughput	[Gops/sec]	2.71	1.21	0.96	0.016	0.015
Average Power	[mW]	1.85	950	950	950	950
Energy	[uJoule]	7.21	32,813.00	44,070.50	2,426,119.50	2,788,126.50
Energy Efficiency	[Gops/Joule]	1464.72	36.77	21.75	0.007	0.005

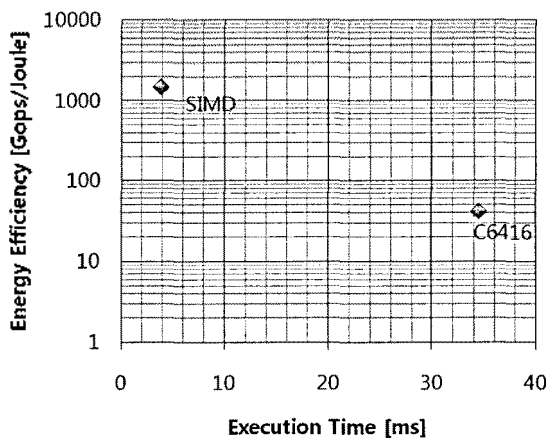


그림 7. 실행 시간과 에너지 효율 비교
Fig. 7. Execution time and energy efficiency comparison between SIMD and C6416.

V. 결론

본 논문에서는 기타의 음 합성 알고리즘을 실시간으로 처리하기 위해 SIMD기반 병렬 프로세서를 제안하였다. 제안한 병렬 프로세서는 6개의 프로세싱 엘리먼트의 메쉬 배열 구조로 구성하였으며, 각각의 프로세싱 엘리먼트는 해당되는 현을 모델링하여 최종적으로 6개의 현으로 구성되어 있는 기타 음을 실시간으로 합성할 수 있다. 제안한 SIMD기반 병렬프로세서를 이용한 합성음과 원음에 대한 청취 및 스펙트럼을 분석한 결과, 매우 유사함을 알 수 있었다. 또한 제안한 병렬 프로세서는 C6416과 비교 하여 실행 시간에서 약 8.9배, 에너지 효율에서 약 39.8배의 성능 향상을 보였다. 이러한 결과는 제안한 병렬 프로세서 구조가 실시간 음 합성에 있어서 무한한 잠재 가능성이 있음을 보여주는 것이다. 향후 제안한 기타의 음 합성을 위한 병렬 프로세서를 하드웨어로 구현하기

위한 연구와 이를 바탕으로 초보자도 손쉽게 연주할 수 있는 전자 악기 개발에 관한 연구가 필요할 것이다.

감사의 글

이 논문은 2009년도 정부 (교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. R01-2008-000-20493-0).

참고 문헌

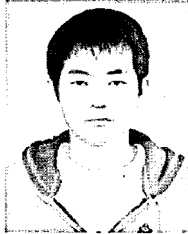
1. Curtis Roads, *The Computer Music Tutorial*, The MIT press, London, 1996.
2. J. O. Smith, "Physical modeling using digital waveguides," *Computer Music J.*, vol 16, no. 4, pp. 74-87, 1992.
3. H. G. Alles, "Music synthesis using real time digital techniques," *Proc. IEEE*, vol, 68, no. 4, pp. 436-449, 1980.
4. V. Valimaki, 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, 1996.
5. 조상진, 정의필, "산조 가야금의 물리적 모델링", *한국음향학회지*, 23권, 7호, 521-531쪽, 2004.
6. 강명수, 조상진, 정의필, "범용 디지털 신호처리를 이용한 국악기 사운드 엔진 개발", *한국음향학회지*, 28권, 3호, 229-238쪽, 2009.
7. 강명수, 조상진, 정의필, "물리적 모델링 합성법에 기반을 둔 줄 없는 기타 구현", *한국음향학회지*, 28권, 2호, 119-126쪽, 2009.
8. 조상진, 정의필, "개선된 산조 가야금의 물리적 모델링을 이용한 오른손 주법의 음 합성", *한국음향학회지*, 25권, 8호, 325-332쪽, 2006.
9. A. W. Y. Su, W.-C. Chang, and R.-W. Wang, "IIR synthesis method for plucked-string instruments with embedded portamento," *J. Audio Eng. Soc.*, vol. 50, no. 5, pp. 351-362, 2002.
10. 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, 1993.

11. 유준희. "편경의 진동모드 분석". *한국음향학회지*, 25권, 3호, 21-28쪽, 2006.
12. U. P. Chong and S. J. Cho, "Physical modeling of gayageum with application to sound engine in musical synthesizer," in *Proc. Int. Conf. High Performance Scientific Computing*, 2006.
13. *Q format*, <http://www.mcublog.co.kr/443>
14. J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proc. International Computer Music Conference*, pp.74-91, Sep., 1993.
15. 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, 2000.
16. V. Tiwari, S. Malik, and A. Wolfe, "Compilation techniques for low energy: An overview," in *Proc. IEEE Intl. Symp. on Low Power Electron.*, pp. 38-39, 1994.
17. Antonio Gentile, D. Scott Wills, "Portable Video Supercomputing," *IEEE Trans. Computers*, vol. 53, no. 8, pp. 960-973, 2004.

저자 약력

• 최 지원 (Ji-Won Choi)



2009년 8월 울산대학교 정보통신공학과 (학사)
 2009년 9월~현재: 울산대학교 컴퓨터정보통신공학과 석사과정
 ※ 주관심 분야: Embedded System, Application Specific SoC Design, Computer Architecture, Parallel Processing, Multimedia Signal Processing

• 김 용 민 (Yong-Min Kim)



2009년 8월 울산대학교 정보통신공학과 (학사)
 2009년 9월~현재: 울산대학교 컴퓨터정보통신공학과 석사과정
 ※ 주관심 분야: Embedded System, Application Specific SoC Design, Computer Architecture, Parallel Processing, Multimedia Signal Processing

• 조 상 진 (Sang-Jin Cho)

한국음향학회지 제 23권 7호 참조

• 김 중 면 (Jong-Myon Kim)

한국음향학회지 제 26권 6호 참조

• 정 의 필 (Ui-Pil Chong)

한국 음향학회지 제 23권 7호 참조