

가야금의 실시간 음 합성을 위한 멀티코어 프로세서 구현

최 지원[†] · 조 상 진^{††} · 김 철 홍^{†††} · 김 종 면^{††††} · 정 의 필^{†††††}

요 약

물리적 모델링은 실제 악기음과 유사한 고음질의 음을 합성하는 방법으로 많은 연구가 진행되어 왔다. 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 전통 현악기인 가야금의 음 합성 알고리즘을 실시간으로 처리 가능한 단일 명령어 다중 데이터(Single Instruction Multiple Data, SIMD) 방식의 멀티코어 프로세서를 제안한다. 제안하는 SIMD기반 멀티코어 프로세서는 가야금의 12개 현을 제어하기 위해 12개의 프로세싱 엘리먼트(Processing Element, PE)로 구성되어 있다. 각각의 프로세싱 엘리먼트는 해당되는 가야금 현을 모델링하며, 각 현의 여기신호와 파라미터를 음 합성 병렬 알고리즘의 입력으로 받아 동시에 12개 현의 합성된 음을 실시간으로 생성할 수 있다. 표본화 비율을 44.1kHz로 설정하고 16비트 양자화 데이터의 음을 합성한 모의실험 결과, 제안한 SIMD기반 멀티코어 프로세서를 이용한 합성음은 원음과 매우 유사하였으며, 상용 프로세서(TI TMS320C6416, ARM926EJ-S, ARM1020E)보다 실행 시간에서 5.6~11.4배, 에너지 효율에서 553~1,424배의 향상을 보였다.

키워드 : SIMD기반 멀티코어 프로세서, Application-specific 프로세서, 음 합성, 물리적 모델링, 디지털 도파관, 가야금

Multi-Core Processor for Real-Time Sound Synthesis of Gayageum

Jiwon Choi[†] · Sang-Jin Cho^{††} · Cheol-Hong Kim^{†††} · Jong-Myon Kim^{††††} · Ui-Pil Chong^{†††††}

ABSTRACT

Physical modeling has been widely used for sound synthesis since it synthesizes high quality sound which is similar to real-sound for musical instruments. However, physical modeling requires a lot of parameters to synthesize a large number of sounds simultaneously for the musical instrument, preventing its real-time processing. To solve this problem, this paper proposes a single instruction, multiple data (SIMD) based multi-core processor that supports real-time processing of sound synthesis of gayageum which is a representative Korean traditional musical instrument. The proposed SIMD-base multi-core processor consists of 12 processing elements (PE) to control 12 strings of gayageum in which each PE supports modeling of the corresponding string. The proposed SIMD-based multi-core processor can generate synthesized sounds of 12 strings simultaneously after receiving excitation signals and parameters of each string as an input. Experimental results using a sampling rate 44.1 kHz and 16 bits quantization show that synthesis sound using the proposed multi-core processor was very similar to the original sound. In addition, the proposed multi-core processor outperforms commercial processors(TI's TMS320C6416, ARM926EJ-S, ARM1020E) in terms of execution time (5.6~11.4x better) and energy efficiency (about 553~1,424x better).

Keywords : SIMD-Based Multi-Core Processor, Application-Specific Processor, Sound Synthesis, Physical Modeling, Digital Waveguide, Gayageum

1. 서 론

악기 음을 사실적으로 재생하기 위해서는 악기의 특성에 맞는 모델링 방법을 선택해야 한다. 모델링을 이용한 악기 음 합성에는 물리적 모델링 (physical modeling)과 스펙트럼 모델링 (spectral modeling) 방식이 있다. 스펙트럼 모델링은 대상 악기 음의 스펙트럼을 정현파 성분과 음의 특성에 영향을 미치는 다른 성분들의 합으로 해석하고 이를 통해서

* 이 논문은 2010년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. R01-2008-000-20493-0).
† 준 회원: 울산대학교 컴퓨터공학과 석사과정
†† 정 회원: 울산대학교 BK21 e-Vehicle연구인력양성사업단 연구교수
††† 종신회원: 전남대학교 전자컴퓨터공학부 교수
†††† 정 회원: 울산대학교 컴퓨터정보통신공학부 교수(교신저자)
††††† 정 회원: 울산대학교 컴퓨터정보통신공학부 교수
논문접수: 2010년 7월 26일
수정일: 1차 2010년 10월 11일, 2차 2010년 11월 10일
심사완료: 2010년 11월 10일

음을 합성하는 기술이다. 스펙트럼 모델링에 대한 연구로 악기 음 합성 분야에는 해금 소리의 포먼트 합성[1]이 있고, 악기 음 합성 분야는 아니지만 스펙트럼 모델링을 사용한 수염고래의 소리 합성[2]이 있다. 물리적 모델링은 악기의 구조적, 음향적 특성을 분석하여 설계한 디지털 필터를 이용하여 실제 악기음과 유사한 고음질의 음을 합성하는 방식이다 [3]. 따라서 디지털 필터의 파라미터를 이용하여 음색을 조절하고 특정 필터를 추가 및 제거함으로써 연주 방법을 표현할 수 있다. 또한 물리적 모델링은 디지털 도파관(digital waveguide)을 이용한 방식을 사용하여 악기의 소리 발생 원리를 분석하고 그에 적합한 모델을 만들어 음을 합성한다[3,4].

악기 음 합성 분야에서 물리적 모델링의 연구가 활발히 진행되어 왔다. A. Su는 중국의 구친 (Gu-chin)이라는 현악기 모델링에 대하여 연구 하였고[5], V. Valimaki는 칸텔레 (Kantele)라는 핀란드 전통악기를 모델링하였다[6,7]. 국내에서는 산조 가야금의 물리적 모델링에 대한 연구[8,9], 편경에 대한 진동모드 분석[10] 및 물리적 모델링 합성법을 이용한 줄 없는 기타 구현[11] 등이 있다. 그러나 손으로 현을 뜯거나 튕겨서 연주하는 현악기의 음 합성 알고리즘은 상당한 양의 연산과 입출력 처리를 요구하기 때문에 실시간 처리에 많은 어려움이 있다[2,6]. 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 고성능, 저전력을 충족시킬 수 있지만 다양한 애플리케이션에서 요구되는 프로그램능력(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 멀티코어 프로세서 아키텍처가 대안으로 부각되고 있다. 명령어 레벨 (Instruct-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, SIMD기반 멀티코어 프로세서는 여러 개의 저비용 프로세싱 엘리먼트 (processing element, PE)들을 이용하여 고성능을 추구하고 동시에 저장장치와 데이터 통신 요구를 최소화하기 위해 프로세싱 엘리먼트와 데이터 입출력을 동일위치에 배치함으로써 저 전력을 만족시킨다[13].

본 논문에서는 대표적 국악기인 가야금의 물리적 모델링

합성법을 실시간으로 처리하기 위해 SIMD기반 멀티코어 프로세서 아키텍처를 제안한다. 제안한 SIMD기반 멀티코어 프로세서는 12개의 프로세싱 엘리먼트로 구성되어 있으며, 각각의 프로세싱 엘리먼트는 12개 현으로 구성된 가야금의 해당되는 현을 모델링한다. 따라서 각 현의 여기신호 (excitation signal)와 파라미터를 합성 병렬 알고리즘의 입력으로 받아 가야금의 다양한 연주 방법에 대한 12개 현의 합성된 음을 실시간으로 생성할 수 있다. 제안한 SIMD기반 멀티코어 프로세서를 이용한 합성음은 원음과 매우 유사하였으며, 상용 프로세서(TI TMS320C6416, ARM926EJ-S, ARM1020E)보다 실행 시간에서 5.6~11.4배, 에너지 효율에서 553~1,424배의 성능 향상을 보였다. 이러한 결과는 현악기를 포함한 다양한 악기의 사운드 엔진 개발에 있어서 제안하는 멀티코어 프로세서의 무한한 잠재 가능성을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 제안한 SIMD기반 멀티코어 프로세서의 관련 연구에 대해 소개하고, 3장에서는 디지털 도파관을 이용한 현악기의 물리적 모델링을 소개한다. 그리고 4장에서는 제안하는 SIMD기반 멀티코어 프로세서 모델을 소개한하고, 5장에서는 가야금의 음 합성 병렬 알고리즘을 설명하고, 6장에서는 SIMD기반 멀티코어 프로세서를 이용한 가야금의 음 합성 결과 및 상용프로세서와의 비교, 분석한다. 끝으로 7장에서 이 논문의 결론을 맺는다.

2. SIMD기반 멀티코어 프로세서의 관련 연구

멀티미디어 애플리케이션에 대한 데이터 레벨 병렬성 (data-level parallelism, DLP)에 관한 연구는 크게 두 개의 연구 그룹으로 나누어 진다: (1) 현재의 SIMD 명령어를 이용하여 성능을 향상시키는 그룹 [14],[15],[16]과 (2) SIMD기반 병렬 프로세서를 이용하여 성능을 향상시키는 그룹 [17],[18]. 많은 연구 그룹 혹은 개인들이 범용 마이크로프로세서에서 멀티미디어 애플리케이션에 대한 SIMD 명령어의 효율성에 대하여 분석하였다. [14]에서는 UltraSPARC 프로세서에서 이미지와 비디오 처리에 대한 VIS 명령어의 효율성을 기술하였다. 4-way out-of-order 프로세서는 single in-order 프로세서보다 2.3배~4.2배의 성능을 향상시켰고 더불어 VIS 명령어는 1.1배~4.2배의 성능을 더 향상시켰다. [15]에서는 DSP와 멀티미디어 애플리케이션에 대한 MMX 명령어의 성능 평가를 기술하였다. MMX 명령어는 81%의 다이내믹 명령어를 감소시켜 평균 5.5배의 성능 향상을 보였다. 이러한 결과에서 보는 바와 같이 SIMD 명령어는 적당한 수준의 성능을 향상시킨다. 하지만 멀티미디어 애플리케이션에 내재한 완전한 데이터 병렬성을 얻지 못하기 때문에 다양한 형태의 멀티미디어에서 요구되는 상당한 양의 성능 요구를 만족시키지 못할 것이다.

SIMD기반 병렬 프로세서는 공간적 병렬성(spatial parallelism)을 실현하기 위해 여러 개의 동기화된 프로세싱 유닛(processing unit)들을 사용한다. 이 유닛들은 하나의 제어 유닛으로부터 동시에 전송되는 동일한 연산 명령을 서로 다른 데이터에 대하여 수행한다. 따라서 데이터 병렬 모델을 이용하여 성능을 향상시킨다. 고도 데이터 병렬 어레이(massively data parallel array)들은 거의 30년 동안 이미지 처리에 사용되어 왔지만, 초기의 SIMD기반 병렬 프로세서(TMC Connection Machine 1[19])는 I/O 테크놀로지에 의해 제한되었다. 이후의 SIMD 병렬 프로세서인 TMC CM-2[20]와 MasPar MP-2[21]는 버퍼 이미지의 큰 병렬 디스크 어레이의 사용을 통해 이러한 제한을 극복하였지만 큰 비용과 휴대성에서 문제가 있다. Fine-grained 병렬 프로세서인 MGAP[22]와 ABACUS[23]는 이러한 휴대성 이슈를 해결하였지만, 그들의 성능은 I/O bandwidth와 latency에 의해 제한되었다.

악기의 음 합성 분야에서 가장 많이 사용되는 프로세서는 TI DSP이다. 그러나 동시 발음수가 높은 가야금의 경우 실시간 처리에 어려움이 있다. 실시간 처리를 위해 여러 개의 TI DSP를 사용하면 비용, 하드웨어 사이즈, 파워 등의 낭비가 심하게 된다. 이러한 문제를 해결하기 위해 본 논문에서는 SIMD기반 멀티코어 프로세서를 이용한 음 합성을 제안한다. 이미지 처리에 주로 사용되었던 SIMD기반 멀티코어 프로세서를 오디오 처리도 가능하게 변경 하였다.

3. 디지털 도파관을 이용한 현악기의 물리적 모델링

J. O. Smith가 제안한 디지털 도파관 모델링은 악기의 파동을 모델링 하는 방식으로[3], 파동의 이동은 지연 라인으로 묘사하고, 악기의 다른 부분은 디지털 필터로 구현함으로써 악기의 물리적 특성과 음 발생 원리를 바탕으로 한 합성 방식이다. 예를 들어 기타는 파동을 발생시키는 현을 지연라인으로, 파동을 증폭시키는 몸통과 파동을 전달시키는 브리지는 디지털 필터로 나누어 표현 할 수 있다. 이렇게 악기를 기능별로 나누어 구현할 경우 각각의 모델은 유사한 악기에서 공통으로 사용될 수 있는데, 가야금의 안죽은 거문고나 아쟁의 안죽 모델로 사용할 수 있고, 기타 현의 모델은 다른 뜯는 현악기 현의 모델로 사용할 수 있다.

디지털 도파관 합성에서 줄의 진동과 몸체의 방사에 대한 블록 다이어그램은 (그림 1)과 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, $L = L_I + L_f$ 와 같이 표현할 수 있다. L_I 는 지연 라인의 정수부로서 가야금 모델에서의 지연에 해당하고, L_f 는 미소 지연부로서 1보다 작은 양의 실수를 가진다.

그리고 지연라인의 길이는 파동의 속도에 따라 가변적이다. 미소지연 필터 (fractional delay filter) $F(z)$ 는 미소 지연 L_f 를 보상하기 위한 것으로, 본 논문에서는 식 (1)과 같은 1차 라그랑주 보간기를 사용하였다.

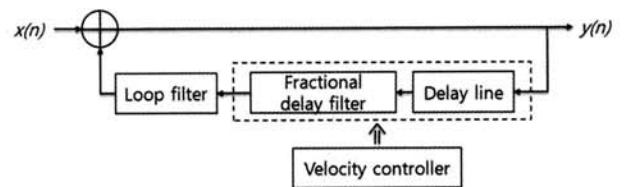
$$F(z) = h_0 + h_1 z^{-1} \tag{1}$$

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[1].

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

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

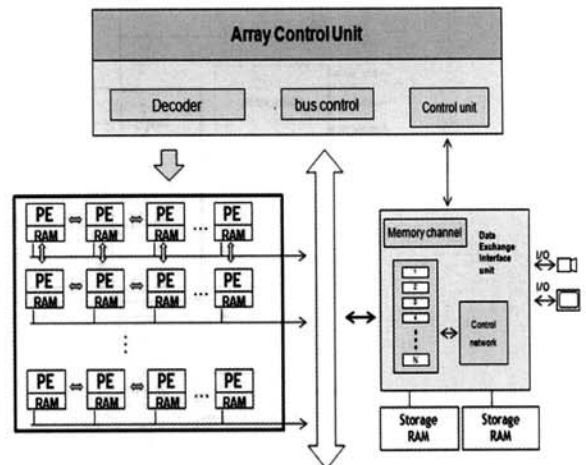
$$S(z) = \frac{1}{1 - z^{L_I} F(z) H_l(z)} \tag{3}$$



(그림 1) 현악기 모델의 블록 다이어그램

4. SIMD기반 멀티코어 프로세서 모델

4.1 멀티코어 프로세서 구조



(그림 2) 멀티코어 프로세서 전체구조

멀티코어 프로세서는 2차원으로 되어 있는 프로세싱 엘리먼트 어레이, 로컬 메모리 그리고 각 프로세싱 엘리먼트와 입출력 유닛을 제어 하는 ACU(array control unit)로 구성된다. (그림 2)는 본 논문에서 제안하는 멀티코어 프로세서 아키텍처와 상호 연결 네트워크를 보여준다. 프로세싱 엘리먼트는 매쉬 구조 (mesh network)로 구성되어 있으며, 컨트롤 유닛은 명령어 메모리에서 명령어를 패치 (fetch)하여 각 프로세싱 엘리먼트에 전송하는 역할을 한다. 그리고 데이터 전송 유닛 (data exchange interface, DEI)은 프로세싱 엘리먼트의 레지스터와 외부 I/O 사이의 데이터 전송을 담당한다.

4.2 프로세싱 엘리먼트 구조

각 프로세싱 엘리먼트는 RISC 아키텍처로 구성되어 있으며 연산 및 데이터 처리는 32비트 단위로 수행한다. 프로세싱 엘리먼트 내부의 구성은 (그림 3)과 같다. 512개 워드 폭을 가진 로컬 메모리와 데이터 입출력의 메모리 저장 장소를 가지고 있다. 레지스터 파일은 16개로써 3개의 포트를 이용하여 연산의 입력 및 출력이 이루어진다. 명령어 디코더는 ACU에서 받은 명령어를 해석하여 프로세싱 엘리먼트 내 모듈들의 동작을 결정하여 신호를 발생한다.

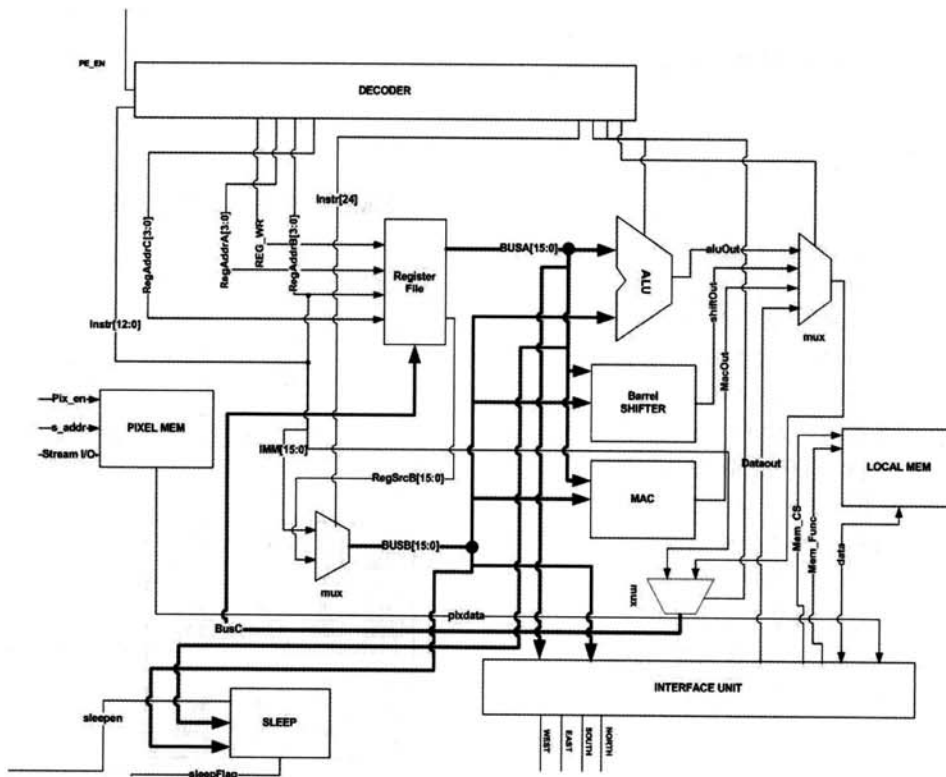
곱셈 및 누산기(multiply accumulator)는 64bit 크기의 누산기로 구성되어 있다. 곱셈 및 누산의 연산은 1cycle로 정

의되어 있으며 상위 32 비트와 하위 32비트를 레지스터 파일에 저장하는데 1cycle이 더 소요된다. 산술/논리 연산 장치(ALU)는 논리곱(AND), 논리합(OR), 배타적 논리합(XOR) 논리 모듈과 산술 모듈인 덧셈기와 뺄셈기로 구성되어 있다. 배럴 쉬프트(Barrel Shifter) 모듈은 왼쪽 또는 오른쪽으로 32비트까지 쉬프트 할 수 있다. 알고리즘의 효율성을 높이기 위해서는 쉬프트 모듈의 필요성이 증대되기 때문에 산술 논리 연산 모듈 이외에 추가적으로 이 모듈을 구성하였다. 프로세싱 엘리먼트의 제어 기능을 하는 활성 (Wakeup) 및 비활성(Sleep) 유닛은 상태 및 조건에 따라 프로세싱 엘리먼트를 활성 및 비활성 상태를 유지시킨다.

마지막으로 NWES(north-west-east-south) 명령어를 통해 프로세싱 엘리먼트 어레이 내부에 각 프로세싱 엘리먼트 간의 데이터 통신이 이루어지며, 1cycle 내에 전송과 레지스터에 적재 되도록 설계 하였다.

로컬 I/O 메모리는 데이터의 입력과 출력이 같은 저장장소를 가지는 구조로 되어 있으며 DEI를 통해 프로세싱 엘리먼트와의 데이터 교환이 이루어진다. 레지스터 파일에는 32비트 크기의 16개의 레지스터가 포함되어 있으며, 0번째 레지스터는 소스 레지스터 없이 상수 값이 직접적으로 쓰여 지도록 설계 되었다.

4.3 가야금을 위한 멀티코어 프로세서 구조

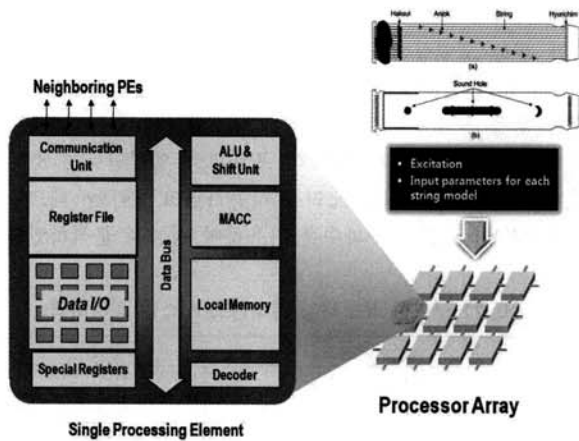


(그림 3) 프로세싱 엘리먼트 구조

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

SIMD기반 멀티코어 프로세서 아키텍처의 블록 다이어그램은 (그림 4)와 같다. SIMD기반 멀티코어 프로세서는 여러 개의 프로세싱 엘리먼트와 이를 제어하는 Array Control Unit (ACU)으로 구성된다. 데이터가 각 프로세싱 엘리먼트에 일정하게 분배되면 프로세싱 엘리먼트들은 메쉬 배열 구조에서 명령어들을 수행한다. 본 논문에서는 가야금의 12개 현을 모델링하기 위해 12개의 프로세싱 엘리먼트를 메쉬 구조로 연결하였으며, 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 활성화 및 비활성 시키는 Sleep 유닛
- 이웃하는 프로세싱 엘리먼트들과 데이터 통신을 위한 NEWS(north-east-west-south) 네트워크 및 serial I/O 유닛

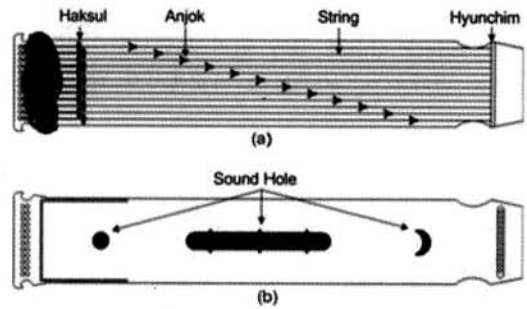


(그림 4) SIMD기반 멀티코어 프로세서 아키텍처와 싱글 프로세싱 엘리먼트

5. 가야금의 음 합성 병렬 알고리즘

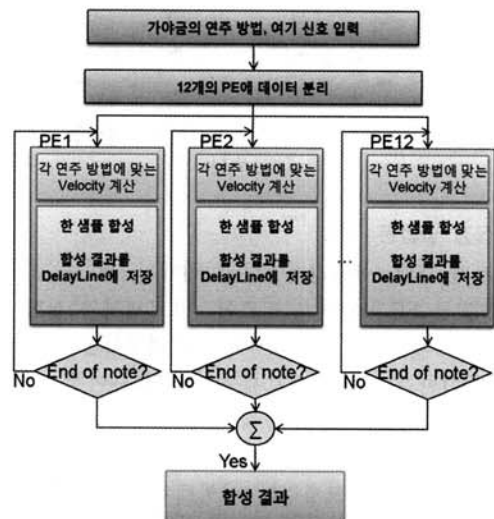
(그림 5)에서 보는바와 같이 가야금은 다른 현악기와 마찬가지로 현, 브리지, 몸통으로 이루어져 있으며, 손으로 뜯거나 튕겨서 연주를 한다. 가야금에는 안죽(雁足, Anjok)이라는 독특한 브리지가 있는데, 연주자는 이를 움직여 조율한다. 안죽은 몸통 전체에 걸쳐 있는 현을 받치고 있어 실제 진동하는 현의 길이를 조절할 뿐만 아니라, 현의 진동을

몸통으로 전달하는 기능도 한다. 이러한 안죽의 진동 전달 특성은 임펄스 응답으로부터 근사화된 디지털 필터로 구현할 수 있다. 몸통은 현의 진동을 안죽으로부터 전달받아 공명시키는 역할을 한다. 몸통의 공명 특성은 몸통의 임펄스 응답으로부터 추출하였으며, 이는 몸통의 공명 특성을 대체할 공명기 (resonator)의 파라미터로 사용된다.



(그림 5) 대표적인 전통 현악기 : 가야금. (a) 앞면, (b) 뒷면

가야금의 음 합성을 위한 병렬 알고리즘은 (그림 6)과 같다. 가야금의 12개 현을 모델링하기 위해 12개의 프로세싱 엘리먼트를 메쉬구조로 연결하였으며, 각각의 프로세싱 엘리먼트는 해당되는 현을 모델링한다. 가야금의 음 합성을 위해 가야금의 연주법과 각 줄에 해당하는 여기신호와 파라미터들을 입력 받고, ACU (Array control unit)는 입력 받은 데이터들을 각 줄에 맞게 12개의 프로세싱 엘리먼트에 분배한다. 프로세싱 엘리먼트는 각 연주방법에 따른 파동의 상대적인 속도 (velocity)를 계산하여 지연 라인의 길이를 결정하고, 세 개의 파라미터인 a1 (루프 필터 계수), g (루프 필터의 이익) 그리고 h0, h1 (라그랑제 보간기의 필터 계수)를 이용하여 한 샘플의 음을 합성한다. 음 합성 과정의 반복을 통해 각각의 프로세싱 엘리먼트는 해당되는 현의 음을 합성할 뿐만 아니라 12줄의 가야금 음을 동시에 합성할 수 있다.

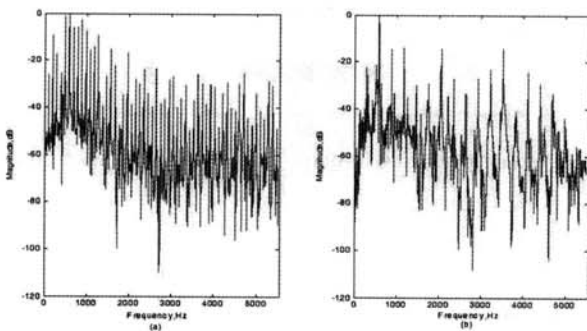


(그림 6) 가야금의 음 합성을 위한 병렬 알고리즘

6. 모의실험 및 성능 분석

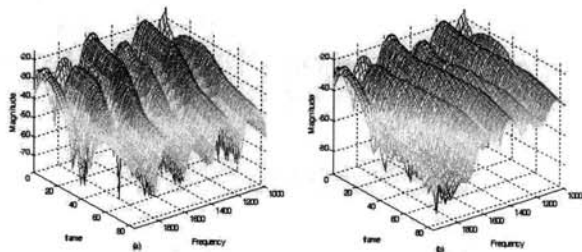
6.1 가야금의 음 합성 결과

본 논문에서는 가야금의 몸통 특성을 여기 신호에 포함시켜 사용하는 대체 합성법(commuted synthesis)을 기반으로 구현한 것이므로 합성된 신호는 가야금의 음색을 표현하는데 문제가 없다[24]. SIMD기반 멀티코어 프로세서를 이용한 합성음과 원음을 청취한 결과 거의 동일하였으며, (그림 7)에서 보는 바와 같이 스펙트럼 비교에서도 매우 유사함을 보였다.



(그림 7) 원음(실선)과 SIMD기반 멀티코어 프로세서(점선)를 이용한 합성음의 스펙트럼 : (a) 1번 현, (b) 7번 현

추성, 퇴성과 같은 가야금 연주법에 대한 스펙트럼은 (그림 8)과 같다. 이는 가야금의 음 합성 병렬 알고리즘이 피치가 변하는 가야금의 각 연주법을 잘 표현할 수 있음을 보여주는 것이다.

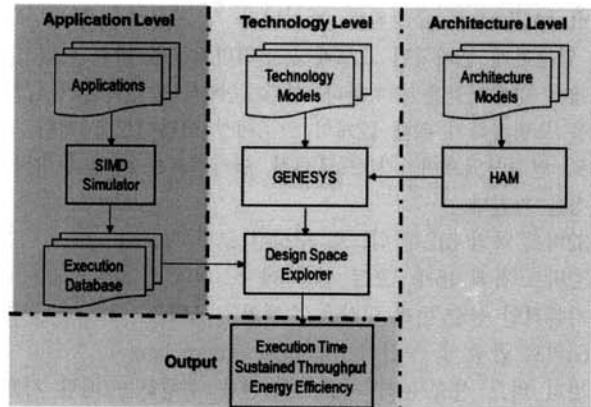


(그림 8) 가야금 연주법의 합성음에 대한 스펙트럼: (a) 추성, (b) 퇴성

6.2 실험 방법론 구조

세 가지 레벨(애플리케이션, 아키텍처, 테크놀로지)로 구성되어 있는 SIMD기반 멀티코어 프로세서의 실험 방법론은 (그림 9)와 같다. 애플리케이션 레벨에서는 명령어 레벨의 SIMD기반 멀티코어 프로세서용 정밀 사이클 시뮬레이터를 이용하여 사이클 개수, 동적 명령어 빈도, 프로세싱 엘리먼트 이용률(utilization) 등의 실험 데이터를 추출한다. 아키텍처 레벨에서는 모델링된 아키텍처의 디자인 변수들을 계산하기 위해 Chai[25]가 제안한 SIMD 기반 멀티코어 프

로세서용 이중 아키텍처 모델링 툴을 사용하였다. 테크놀로지 레벨에서는 각 아키텍처 모델들의 테크놀로지 변수(latency, power, clock frequency)를 계산하기 위해 Generic System Simulator (GENESYS)를 사용하였다[26]. 마지막으로 세 레벨에서 구해진 데이터베이스를 조합하여 각 경우에 대한 실행시간, 처리량, 에너지 효율을 결정하였다.



(그림 9) SIMD기반 멀티코어 프로세서 시뮬레이션을 위한 실험 방법론

6.3 멀티코어 프로세서를 위한 성능 분석 지표

가야금의 음 합성 알고리즘을 구현하기 위해 SIMD기반 멀티코어 프로세서용 정밀 사이클(cycle-accurate) 시뮬레이터를 사용하였으며, 파라미터는 <표 1>과 같다. 가야금의 12개 현을 모델링하기 위해 12개의 프로세싱 엘리먼트를 메쉬구조로 연결하였으며, 각각의 프로세싱 엘리먼트는 해당되는 현을 모델링한다. 가야금의 음 합성에서 지연라인 길이(최대 450)를 고려하여 각 프로세싱 엘리먼트는 32비트 워드 단위의 512개의 메모리를 사용하였고, 130nm 테크놀로지와 720MHz 클럭 주파수를 사용하여 성능을 분석하였다.

<표 1> 구현된 프로세싱 엘리먼트 시스템 파라미터

parameter	value
Number of PEs	12
Memory/PE [32bit - word]	512
VLSI Technology	130nm
Clock Frequency	720MHz
intALU/intMUL/Barrel Shifter/intMACC/comm	1 / 1 / 1 / 1 / 1

<표 2>는 제안한 멀티코어 프로세서의 성능을 평가하기 위한 3가지 지표를 보여준다[17]. 실행 시간(execution time)은 가야금의 음 합성 병렬 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.

〈표 2〉 멀티코어 프로세서를 위한 평가 지표 요약

실행시간 (execution time)	처리량 (sustained throughput)	에너지 효율 (energy efficiency)
$t_{exec} = \frac{C}{f_{ck}}$	$Th_{sust} = \frac{O_{exec} \cdot U \cdot N_{PE}}{t_{exec}} [\frac{Gops}{s}]$	$\eta_E = \frac{O_{exec} \cdot U \cdot N_{PE}}{Energy} [\frac{Gops}{Joule}]$
C : 사이클 개수, f_{ck} : 클럭 주파수, O_{exe} : 수행된 연산 개수 U : 프로세싱 엘리먼트 이용률, N_{PE} : 프로세싱 엘리먼트의 개수		

〈표 3〉 12개의 프로세싱 엘리먼트 시스템에서 가야금의 각 연주법에 대한 성능 결과

연주 방법 [12현]	total cycle [cycles]	vector instruction	scalar instruction	system utilization [%]	sustained throughput[Gops/sec]	execution time [ms]
기본음	4,369,107	3,089,069	1,280,038	80.00	4.88	6.07
추성	5,295,207	3,794,669	1,500,538	78.48	4.86	7.35
퇴성	5,471,607	3,794,669	1,676,938	78.48	4.70	7.60

본 논문에서 공정한 성능 평가를 위해 제안한 멀티코어 프로세서와 상용 프로세서 프로세서인 TI C6416, ARM 926EJ-S, ARM1020E를 동일한 130nm 테크놀로지를 사용하여 모의 실험하였다. 상용 프로세서를 위해 Code Composer Studio 3.1, ARM Developer Suite 1.2를 사용하여 시뮬레이션 하였다. 제안한 프로세서는 12개의 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.

6.4 성능 평가 결과

12개의 프로세싱 엘리먼트로 구성된 멀티코어 프로세서를 이용하여 가야금의 각 연주법을 수행한 결과는 <표 3>과 같다. 가야금의 연주법을 표현하기 위해 피치 변화 코드 부분을 포함시켰고, 이러한 이유로 피치가 변하는 연주법은 기본음 보다 실행 시간에서 1.2배 증가 하였다. 이는 가야금의 연주법을 표현하기 위한 피치 변화가 성능에 영향을 미치고 있음을 보여 준다.

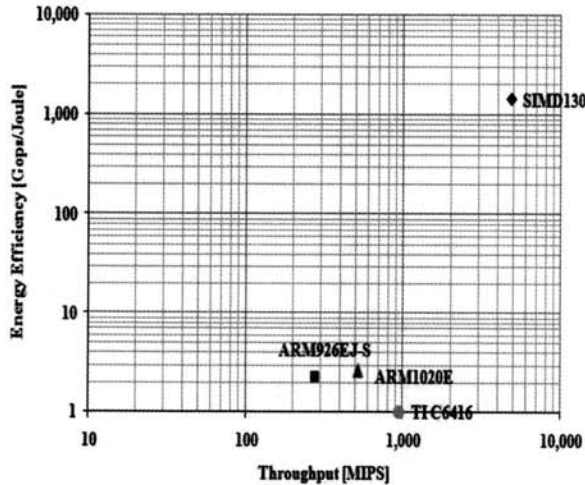
<표 4>는 가야금 음 합성 알고리즘에 대한 제안하는 멀티코어 프로세서와 상용 프로세서의 성능 비교를 보여준다. C6416에서 내부 메모리의 접근 속도가 외부 메모리의 접근 속도보다 빠른 장점을 이용하기 위해 내부 메모리의 사용을 최대화 하였고, 컴파일러 옵션을 -o3으로 셋팅하여 최적화 하였다. 모의 실험한 결과, 제안하는 멀티코어 프로세서는 상용 프로세서보다 5.6~11.4배의 성능 향상을 보였고, 실시간

음 합성이 가능함을 보여 주었다 (> 30ms). 또한 공정한 평가를 위해 각 프로세서에서 구해진 에너지 소모량은 음 합성에 사용된 명령어의 Functional Unit당 소모된 에너지를 계산하여 측정하였다.

〈표 4〉 SIMD기반 멀티코어와 상용 프로세서의 성능 비교

parameter	Units	SIMD130	TI C6416	ARM 926EJ-S	ARM 10E
Technology	[nm]	130	130	130	130
Clock Frequency	[MHz]	720	720	250	400
Average Power	[mW]	3.30	950	120	200
Average Throughput	[MIPS]	4,886	958	275	520
Execution Time[12현]	[ms]	6.07	46.38	69.35	35.63
Energy	[uJoule]	20.03	44,066.31	8322.52	7126.34
Energy Efficiency	[Gops/Joule]	1438.43	1.01	2.29	2.6

(그림 10)은 SIMD기반 멀티코어 프로세서와 상용 프로세서의 에너지 효율과 처리량의 비교를 보여준다. 제안한 멀티코어 프로세서는 상용 프로세서보다 에너지 효율에서 553~1,424배 이상의 향상을 보였다. 에너지 효율은 단위 에너지 (Joule)당 소비된 명령어 개수 (Giga-operations)로서, 에너지 효율의 증가는 시스템의 배터리 수명을 증가시키는 결과를 가져온다.



(그림 10) 에너지 효율과 처리량의 비교

6.5 합성 및 실험 결과

제안된 SIMD기반 멀티코어 프로세서 구조를 RTL (Register Transfer Level)레벨로 설계하고, Xilinx사의 Vertex-4 XC4VLX60 FPGA[27]를 이용하여 합성하고 테스트 하였다. 합성 결과는 <표 5>와 같다, 각 프로세싱 엘리먼트는 1,095개의 LUT와 195개의 레지스터가 사용되었으며, ACU는 1,147개의 LUT와 124개의 레지스터가 사용되었다. 12개의 프로세싱 엘리먼트로 구성된 멀티코어 프로세서는 14,827개의 LUT와 2,464개의 레지스터가 사용되고 전체 메모리 비트는 204,800비트이다.

<표 5> 제안된 멀티코어 프로세서 구현의 합성 결과

ACU (Array Control Unit)	LUTs	1,147
	Register	124
PE (Processing Element)	LUTs	1,095
	Register	195
Total Block Memory bits		204,800

7. 결 론

본 논문에서는 가야금의 음 합성 알고리즘을 실시간으로 처리하기 위해 SIMD기반 멀티코어 프로세서를 제안하였다. 제안한 멀티코어 프로세서는 12개의 프로세싱 엘리먼트로 구성되며, 각각의 프로세싱 엘리먼트는 해당되는 현을 모델링하기 때문에 12개의 현으로 구성되어 있는 가야금 음을 실시간으로 합성할 수 있다. 제안하는 SIMD기반 멀티코어 프로세서를 이용한 합성음은 원음에 대한 청취 및 스펙트럼 결과를 통해 매우 유사함을 보였다. 또한 동일한 공정 (130 nm Technology)에서 제안하는 멀티코어 프로세서는 상용

프로세서보다 실행 시간에서 5.6~11.4배, 에너지 효율에서 553~1,424배의 향상을 보였다. 이러한 결과는 제안하는 멀티코어 프로세서 구조가 실시간 가야금 음 합성에 있어서 무한한 잠재 가능성을 보여주며, 모바일 시스템에 적용할 경우 상당한 성능 향상 및 에너지 소비 감소가 기대된다. 향후 가야금 음 합성을 실시간으로 처리하기 위해 제안한 멀티코어 프로세서를 하드웨어로 구현할 것이며, 이를 바탕으로 초보자도 손쉽게 연주할 수 있는 전자 악기 개발에 관한 연구를 진행할 것이다.

참 고 문 헌

- [1] 홍연우, 조상진, 김종면, 정의필, "캡스트럼 포락선을 이용한 해금 소리의 포먼트 합성", 한국음향학회지, 제28권, 제6호, pp.526-533, 2009년 8월.
- [2] 전희성, Pranab K. Dhar, 김철홍, 김종면, "수정된 스펙트럴 모델링을 이용한 수염고래 소리 합성", 한국정보처리학회 논문지, 제17권, 제1호, 69-78페이지, 2010년 2월.
- [3] J. O. Smith, "Physical modeling using digital waveguides," Computer Music J., vol 16, no. 4, pp. 74-87, 1992.
- [4] H. G. Alles, "Music synthesis using real time digital techniques," Proc. IEEE, vol. 68, no. 4, pp. 436-449, 1980.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 조상진, 정의필, "산조 가야금의 물리적 모델링", 한국음향학회지, 제23권, 제7호, pp. 521-531, 2004.
- [9] 조상진, 정의필, "개선된 산조 가야금의 물리적 모델링을 이용한 오른손 주법의 음 합성", 한국음향학회지, 제25권, 제8호, pp. 325-332, 2006.
- [10] 유준희, "편경의 진동모드 분석", 한국음향학회지, 제25권, 제3호, pp. 21-28, 2006.
- [11] 강명수, 조상진, 정의필, "물리적 모델링 합성법에 기반을 둔 줄 없는 기타 구현", 한국음향학회지, 제28권, 제2호, pp. 119-126, 2009.
- [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] Luong Van Huynh, 김철홍, 김종면, "퍼지 백터 양자화를 위한

대규모 병렬 알고리즘“, 한국정보처리학회논문지 A, 제16-A권, 제6호, pp. 411-418, 2009년 12월

- [14] P. Ranganathan, S. Adve, and N. P. Jouppi, "Performance of image and video processing with general-purpose processors and media ISA extensions," in Proc. of the 26th Intl. Sym. on Computer Architecture, pp. 124-135, May. 1999.
- [15] R. Bhargava, L. John, B. Evans, and R. Radhakrishnan, "Evaluating MMX technology using DSP and multimedia applications," in Proc. of IEEE/ACM Sym. on Microarchitecture, pp. 37-46, 1998.
- [16] N. Slingerland and A. J. Smith, "Measuring the performance of multimedia instruction sets," IEEE Trans. on Computers, vol. 51, no. 11, pp. 1317-1332, Nov. 2002.
- [17] Antonio Gentile, D. Scott Wills, "Portable Video Supercomputing," IEEE Trans. Computers, vol. 53, no. 8, pp. 960-973, 2004.
- [18] A. Krikelis, I. P. Jalowiecki, D. Bean, R. Bishop, M. Facey, D. Boughton, S. Murphy, and M. Whitaker, "A programmable processor with 4096 processing units for media applications," in Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, vol. 2, pp. 937-940, May. 2001.
- [19] L. W. Tucker and G. G. Robertson, "Architecture and applications of the connection machine," IEEE Computer, vol. 21, no. 8, pp. 26-38, 1988.
- [20] "Connection machine model CM-2 technical summary," Thinking Machines Corp., version 51, May 1989.
- [21] MarPar (MP-2) System Data Sheet. MarPar Corporation, 1993.
- [22] M. J. Irwin, R. M. Owens, "A Two-Dimensional, Distributed Logic Processor," IEEE Trans. on Computers, vol. 40, no. 10, pp. 1094-1101, 1991.
- [23] M. Bolotski, R. Armithrajah, W. Chen, "ABACUS: A High Performance Architecture for Vision," in Proceedings of the International Conference on Pattern Recognition, 1994.
- [24] J. O. Smith, "Efficient synthesis of stringed musical instruments," Proc. International Computer Music Conference, pp.74-91. Sep., 1993.
- [25] 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.
- [26] V. Tiwari, S. Malik, and A. Wolfe, "Compilation techniques for Low Energy: An Overview," in Proc. IEEE Intl. Symp. on Low Power Electr., pp. 38-39, 1994.
- [27] Xilinx Vertex-4 FPGA XC4VLX60 data sheet, <http://www.alldatasheet.net/datasheet-pdf/pdf/152986/XILINX/XC4VLX60.html>



최 지 원

e-mail : regualr912@nate.com
 2009년 울산대학교 컴퓨터정보통신공학부 졸업(학사)
 2009년~현재 울산대학교 컴퓨터공학과 석사과정
 관심분야: 임베디드 시스템, 신호처리, 컴퓨터구조



조 상 진

e-mail : sjcho75@ulsan.ac.kr
 1997년 울산대학교 전자공학과 졸업(학사)
 2002년 울산대학교 대학원 전자공학과 (공학석사)
 2007년 울산대학교 대학원 컴퓨터정보통신공학과(공학박사)
 2007년~2008년 울산대학교 컴퓨터정보통신공학과 Post-Doc.
 2008년~현재 울산대학교 BK21 e-Vehicle연구인력양성사업단 연구교수
 관심분야: 음향신호처리, 멀티미디어응용, 워터마킹, 고장진단 등



김 철 홍

e-mail : cheolhong@gmail.com
 1998년 서울대학교 컴퓨터공학과 졸업(학사)
 2000년 서울대학교 컴퓨터공학부 졸업 (공학석사)
 2006년 서울대학교 전기컴퓨터공학부 졸업 (공학박사)
 2005년~2007년 삼성전자 반도체총괄 SYS.LSI사업부 책임연구원
 2007년~현재 전남대학교 전자컴퓨터공학부 교수
 관심분야: 임베디드시스템, 컴퓨터구조, SoC 설계, 저전력 설계 등



김 종 면

e-mail : jmkim07@ulsan.ac.kr
 1995년 명지대학교 전기공학과 졸업(학사)
 2000년 Electrical & Computer Engineering, University of Florida, USA(공학석사)
 2005년 Electrical & Computer Engineering, Georgia Institute of Technology, USA (공학박사)
 2005년~2007년 삼성종합기술원 전문연구원
 2007년~현재 울산대학교 컴퓨터정보통신공학부 교수
 관심분야: 임베디드시스템, 시스템-온-칩, 컴퓨터구조, 병렬처리, 신호처리 등



정의필

e-mail : upchong@ulsan.ac.kr

1978년 울산대학교 전기공학과 졸업(학사)

1980년 고려대학교 대학원 전기공학과
(공학석사)

1986년 미국 오레곤 주립 대학교 전기및
컴퓨터 공학과(공학석사)

1996년 미국 뉴욕 폴리테크닉 대학교 전자공학과(공학박사)

1997년~현재 울산대학교 컴퓨터정보통신공학부 교수

관심분야 : 음향신호처리, 멀티미디어응용, 워터마킹, 고장진단 등