한국컴퓨터정보학회 동계학술대회 논문집 제22권 제1호 (2014. 1)

휴대 장치용 기타 음 합성을 위한 매니코어 아키텍처의 디자인 공간 탐색

강명수°, 김종면*
°울산대학교 전기공학부
*울산대학교 전기공학부
e-mail: {ilmareboy°, jmkim07*}@ulsan.ac.kr

Design Space Exploration of Many-Core Architecture for Sound Synthesis of Guitar on Portable Device

Myeongsu Kang°, Jong-Myon Kim*
°Dept. of Electrical Engineering, University of Ulsan
*Dept. of Electrical Engineering, University of Ulsan

● 요 약 ●

Although physical modeling synthesis is becoming more and more efficient in rich and natural high-quality sound synthesis, its high computational complexity limits its use in portable devices. This constraint motivated research of single-instruction multiple-data many-core architectures that support the tremendous amount of computations by exploiting massive parallelism inherent in physical modeling synthesis. Since no general consensus has been reached which grain sizes of many-core processors and memories provide the most efficient operation for sound synthesis, design space exploration is conducted for seven processing element (PE) configurations. To find an optimal PE configuration, each PE configuration is evaluated in terms of execution time, area and energy efficiencies. Experimental results show that all PE configurations are satisfied with the system requirements to be implemented in portable devices.

키워드: design space exploration, many-core architecture, physical modeling synthesis, single-instruction multiple-data

I. Introduction

Mobile phones have been explored as new musical interfaces by controlling musical parameters for performances and *iPhone* ocarina is an example of this trend [1]. Physical modeling synthesis has received increasing attention to develop musical interfaces with mobile phones [2], since it is becoming more and more efficient in terms of generating rich and natural high-quality sounds that imitate natural instruments. However, its required computational complexity due to many numerical equations limits its use in portable devices.

Among many available computational models, single-instruction multiple-data (SIMD) architectures are typically appropriate for multimedia applications since they can efficiently achieve massive data parallelism inherent in those applications. While it is evident that the overall performance improves by increasing the number of processing elements (PEs), there is no general consensus which grain sizes of processors and memories on SIMD architectures provide both the highest performance and lowest power consumption for sound synthesis (i.e., physical modeling synthesis). Thus, it is important to explore the effect of the amount of sample data directly mapped to each PE. To identify the most efficient sample-per-processing element, seven different PE configurations were simulated for sound synthesis of the acoustic guitar with a sampling rate of 44.1 *kHz* and 16-bit quantization by implementing them in 130 *nm* CMOS technology.

II. The Reference SIMD Many-Core Architecture

Fig. 1 pictorially illustrates the reference SIMD many-core processors consist of a two-dimensional PE array, and PEs execute a set of instructions in lockstep fashion.

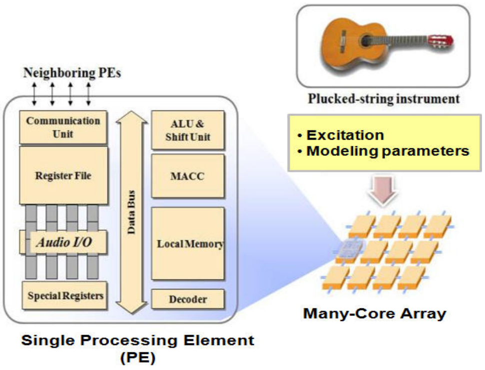

그림 1. SIMD 매니코어 아키텍처
Fig. 1. The Reference SIMD Many-Core Architecture

Each PE has a reduced instruction set computer (RISC) data path with specialized units for SIMD operation and mechanisms for the area I/O data streams. The key functional units are as follows [3].

- A small amount of 32-bit word local memory,
- 16 32-bit three-ported general purpose registers,
- Arithmetic logic unit (ALU) computes basic arithmetic and logic operations,
- Multiply-accumulate (MACC) unit multiplies 32-bit values and accumulates them into a 64-bit accumulator,
- Sleep unit activates/deactivates PEs based on local information,
- PEs communicate with their four nearest neighbors through a north-east-west-south (NEWS) network.

III. Parallel Implementation of Sound Synthesis on Many-Core Architecture

1. Commuted waveguide synthesis

For plucked-string instruments, the strings couple via a bridge to some resonating structure that is required for efficient transduction of string vibration to acoustic propagation, and the resonator imposes its own characteristic frequency response [4]. The key idea of commuted synthesis is to commute the string and resonator, providing a highly efficient way of producing high-quality plucked-string sounds. Since body response of a string instrument is complex and generally requires a high-order digital filter to simulate it, it is possible to record a body impulse response and use it as the string excitation by taking advantage of system commutativity [4]. Consequently, the excitation is convolved with the resonator impulse response, and single aggregate excitation tables can be utilized to generate plucked-string sounds as shown in Fig. 2.

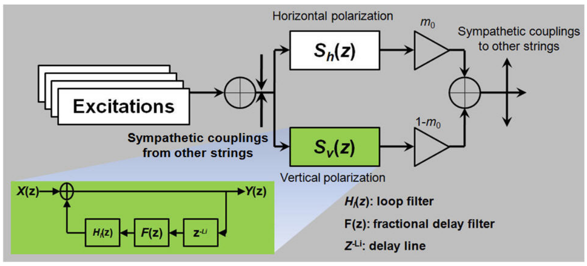

그림 2. 기타 음 합성을 위한 대체 도파관 모델
Fig. 2. Commuted Waveguide Model for Sound Synthesis of Acoustic Guitar

2. Parallel implementation

Fig. 3 shows a pictorial representation of full synthesis mechanism as the sample-per-processing element is 44,100. In this case, two PEs basically generate 44,100 samples for the same guitar string in order to represent dual polarization as depicted in Fig. 3 (i.g., one PE for vertical polarization and another PE for horizontal polarization).

Case I: Since the length of the aggregate excitation for each guitar string is varied, PEs synthesize acoustic guitar sound until their own index values are less than equal to the lengths of their own excitation, where index values indicate locations of PEs' local memories to store synthesized sound samples and load their own excitations. As mentioned just before, two PEs generate acoustic guitar sound samples with slightly different frequencies for the same string. To mix synthesized samples, the left-side PE transfers its synthesized sound sample to the right-side PE. In addition, several PEs should communicate to transfer sound sample owing to sympathetic couplings.

Case II: If PEs' index values are greater than the lengths of excitation, acoustic guitar sounds are synthesized by taking sound samples from their own delay lines, and then the delay lines are updated with synthesized sound samples. Similar to Case I, a number of inter-communications between PEs frequently occur due to dual polarizations and sympathetic couplings. This process repeats until 44,100 six-note polyphonic acoustic guitar sounds are synthesized.

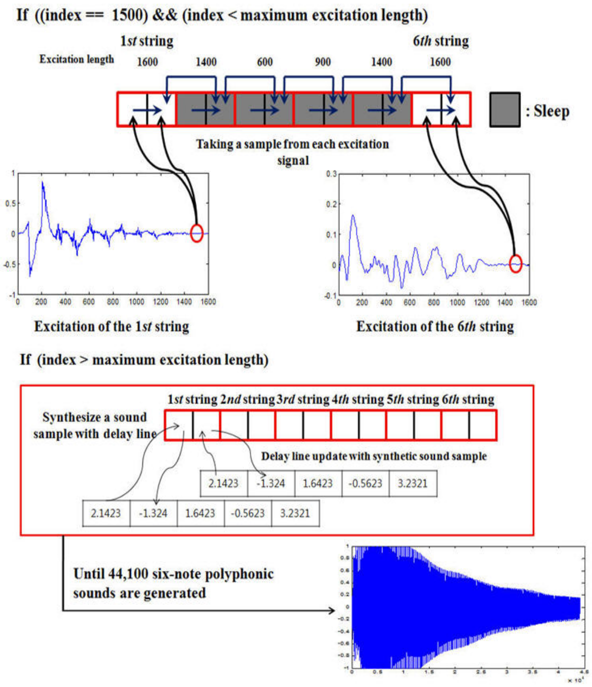

그림 3. 음 합성 알고리즘의 병렬 구현
Fig. 3. Parallel Implementation of Sound Synthesis

IV. Experimental Results

1. Evaluation metrics

This paper evaluates the performance of seven PE configurations (see Table 1) in terms of execution time, area efficiency, and energy efficiency. Execution time is the processing time of sound synthesis. As operations are the units of computations at application level, the throughput can be meaningfully measured in units of Gops/s (Giga-operations per second). Area efficiency is the number of operations per second generated per unit area. As silicon estate is unquestionably one big component of the system cost, maximizing area efficiency implies better component utilization for given system capabilities. Energy efficiency is the task throughput achieved per *Joule*. Increasing energy efficiency implies augmenting sustainable battery life; equivalently, minimizing power dissipation translates into minimizing energy per operation. Area and energy efficiencies are as significant as processing throughput to achieve low-cost design and low-power consumption.

표 1. 실험을 위한 PE 구성
Table 1. PE Configurations for Experiments

PEs/system	Memory/PE [Words]	System memory [KB]
12	1,220	29.28
24	616	29.58
48	314	30.16
96	162	31.04
192	86	33.04
384	46	35.34
768	30	46.08

2. Execution time

Execution times for seven PE configurations are depicted in Fig. 4. As expected, the execution time decreases as the number of PEs increases due to the massive parallelism inherent in physical modeling synthesis. However, the speedup is not linearly decreased since a huge number of inter-communications between PEs occurred as the amount of sample-per-processing element decreases (or the number of PEs increases). Moreover, frequent PE activation/deactivation occurred due to irregular delay line lengths and inter-communications between PEs.

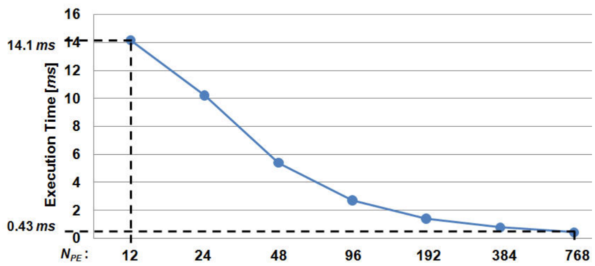

그림 4. PE 구조에 따른 음 합성 실행 시간
Fig. 4. Execution Time with Different PE Configurations

To guarantee CD-quality sound, a sound sample should be sampled at 44.1 *kHz*, which means a sound sample must be produced within 0.2 *ms*. As illustrated in Fig. 4, the longest execution time for all PE configurations was 14.1 *ms* to generate 44,100 six-note polyphonic acoustic guitar sounds. This can be interpreted that one six-note polyphonic sound sample was synthesized in 0.32 *us*. As a consequence, all PE configurations were fast to guarantee CD-quality sound.

2. Energy efficiency

Fig. 5 illustrates energy efficiencies for varying amount of sample-per-processing element. In Fig. 5, the horizontal axis was normalized to an average efficiency of all PE configurations and consequently the shape of the curve is significant. As shown in Fig. 5, the highest energy efficiency was achieved as the number of PEs was 192. At above PEs=384, power consumption of the sleep unit took the largest portion of total power consumption. This is because extremely large amount inter-communications between PEs occurred because of dual polarizations and sympathetic couplings. To communicate between far-apart PEs, it was necessary to activate/deactivate intermediate PEs.

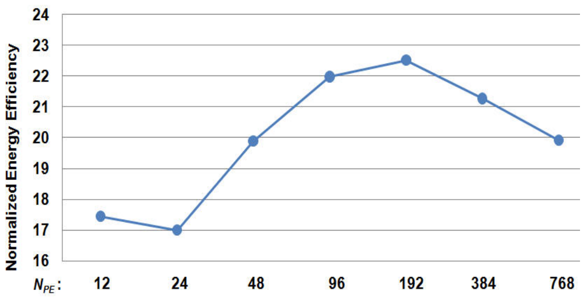

그림 5. PE 구조에 따른 에너지 효율
Fig. 5. Energy Efficiencies with Different PE Configurations

3. System area vs. power consumption

To implement the reference SIMD many-core processor arrays in portable devices, the following two constraints must be considered [5]: 1) the system area should be limited to 140 mm^2, and 2) the power should not exceed 3 watts. As shown in Fig. 6, all of configurations have power consumptions less than 3 watts and system area less than 140 mm^2.

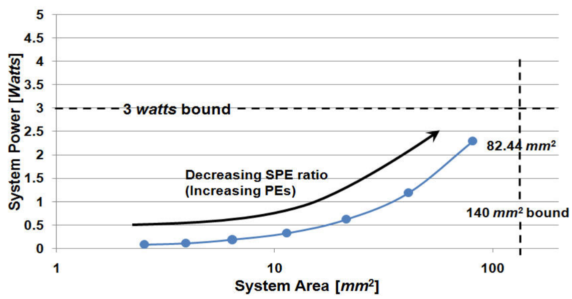

그림 6. PE 구조에 따른 면적 효율 대비 에너지 효율
Fig. 6. System Area vs. Power Consumption for all PE Configurations

V. Conclusions

To find an ideal many-core configuration, it was necessary to explore the impacts of varying amount of sample-per-processing element. Thus, design space exploration of seven PE configurations on performance and efficiency figures was conducted by implementing each PE configuration in 130 *nm* CMOS technology. Experimental results indicated that the most efficient operation was achieved as the number of PEs was 192 (or the amount of sample-per-processing element was 2,756) in order to synthesize 44,100 six-note polyphonic acoustic guitar sound sampled at 44.1 *kHz*. Likewise, all PE configurations met system requirements to be implemented in portable devices.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. NRF-2013R1A2A2A05004566).

References

[1] G. Wang, “Designing Smule’s Ocarina: the iPhone’s Magic Flute,” International Conference on New Interfaces for Musical Expression, pp. 303-309, 2009.
[2] S. Bilbao, “Robust Physical Modeling Sound Synthesis for Nonlinear Systems,” IEEE Signal Processing Magazine, Vol. 24, No. 2, pp. 32-41, 2007.
[3] A. Gentile and D. S. Wills, “Portable Video Supercomputing,” IEEE Trans. Computers, Vol. 53, No. 8, pp. 960-973, 2004.
[4] V. Valimaki and C. Erkut, “Comuuted Waveguide Synthesis of the Clavichord,” Computer Music Journal, Vol. 27, No. 1, pp. 71-82, 2006.
[5] International Technology Roadmap for Semiconductors 2009 Edition, available at http://www.Itrs.net/Links/2009ITRS/Home2009.htm