RIKEN Center for Computational Science

Supported by I-O DATA 財団

Sound Rendering and Its Acceleration Using FPGA

Yiyu Tan and Toshiyuki Imamura
RIKEN Center for Computational Science, 7-1-26 Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo, Japan
Email: tan.yiyu@riken.jp; imamura.toshiyuki@riken.jp

(i) Introduction

➢ Sound rendering based on FDTD schemes is computation-intensive and memory-intensive. The solutions include FPGA-based direct hardware implementation and software simulations on general-purpose processors and GPU.

■ **General-purpose Processors**

- Wave equations are implemented using programming language, such as C++.
- Computation efficiency is low due to intensive data requirements, lots of data misses in caches, and limited memory bandwidth.

■ **GPU**

- Wave equations are implemented using programming language, and computations are carried out by streaming multiprocessors in parallel.
- Data are exchanged through the on-chip shared memory.

■ **FPGAs**

- Wave equations are directly implemented by reconfigurable logic cells.
- Data are stored by on-chip D flip-flops or block memories.

(ii) Rendering Algorithm

The explicit compact FDTD rendering algorithm is applied, and 7-point stencil scheme is adopted. The updated equation and parameters are shown as follows.

$$P_{i,j,k}^{n+1} = D1 * [P_{i-1,j,k}^{n} + P_{i+1,j,k}^{n} + P_{i,j-1,k}^{n} + P_{i,j+1,k}^{n} + P_{i,j,k-1}^{n} + P_{i,j,k+1}^{n} + 2P_{i,j,k}^{n}] - D2 * P_{i,j,k}^{n-1}$$

- Six additions, one subtraction, and three multiplications are required to compute the sound pressure of a grid.
- The parameters are selected according to the position of each grid.

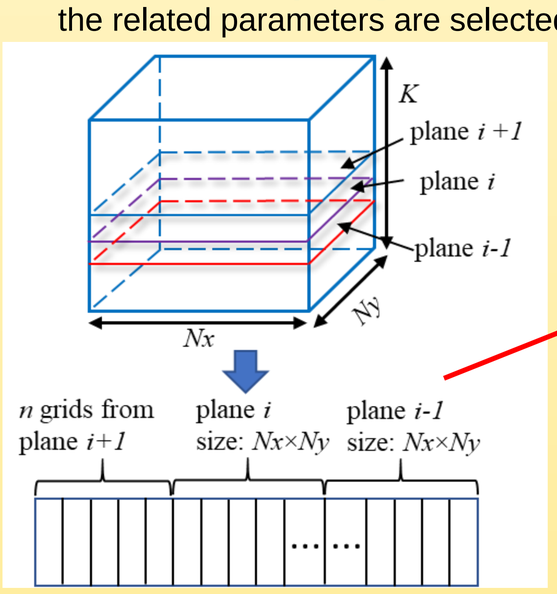
Table 1: *Parameters*

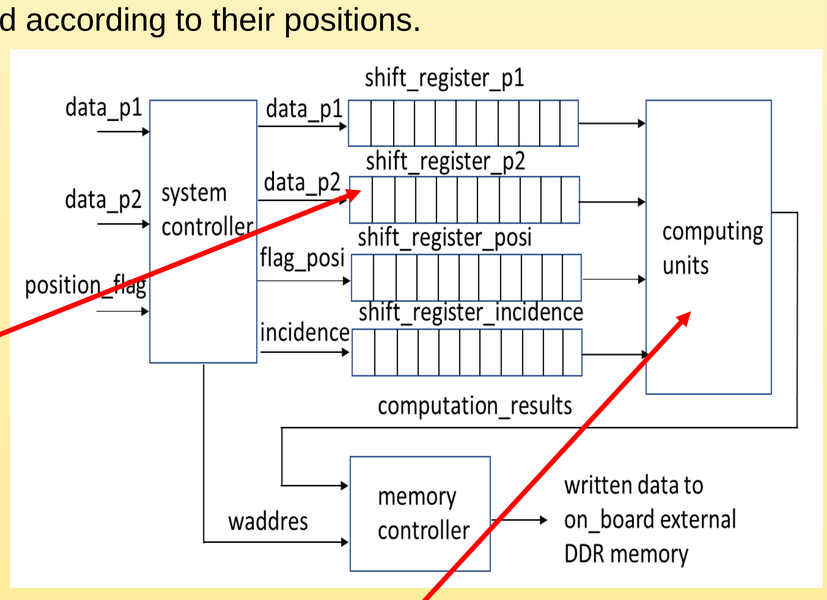
Grid Position	D1	D2
General	1/4	1
Interior	$\frac{R+1}{2\ (R+3)}$	$\frac{3R+1}{R+3}$
Edge	$\frac{R+1}{8}$	R
Corner	$\frac{R+1}{2\ (5-R)}$	$\frac{5R-1}{5-R}$

(iii) System Design and Implementation

The system is designed using OpenCL and implemented using the FPGA board DE5a-NET.

- The external large DDR memory is adopted to extend the simulated area, and **High-speed and high-bandwidth on-chip memories** is employed to implement **a sliding-window-based data buffering** to reduce the required memory bandwidth and data access overhead between the rendering engine and on-board external memory.
- **A uniform computing unit** is developed to compute sound pressures of all nodes, and the related parameters are selected according to their positions.





- Multiple nodes are computed in parallel.
- The sound pressures of nodes are alternatively written into two on-board independent memory banks, and read into two shift registers.
- Table 1 shows the hardware resource utilization in the case of 16 nodes computed in parallel.

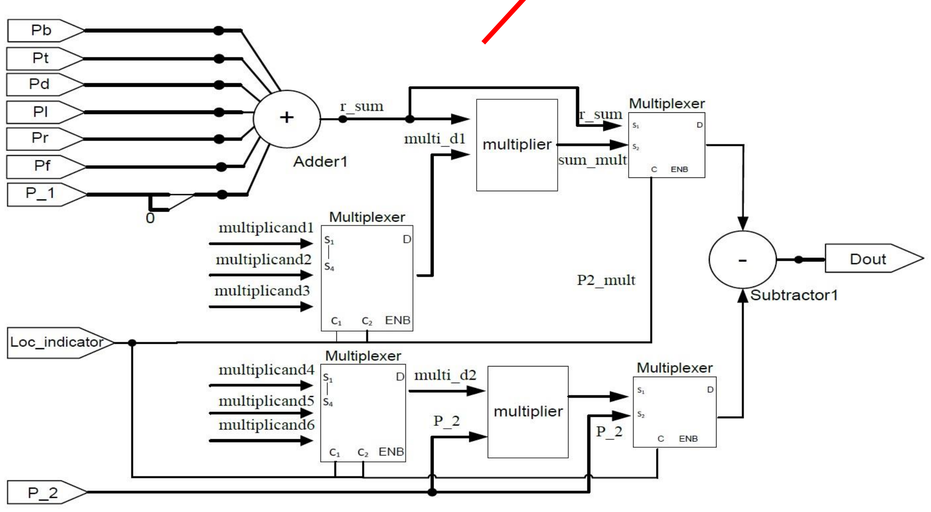


Figure 3. Computing unit.

Table 1: Hardware resource utilization

Logic utilization	DSP blocks	RAM blocks	Clock frequency
70701 (17%)	152 (10%)	891 (33%)	267 MHz

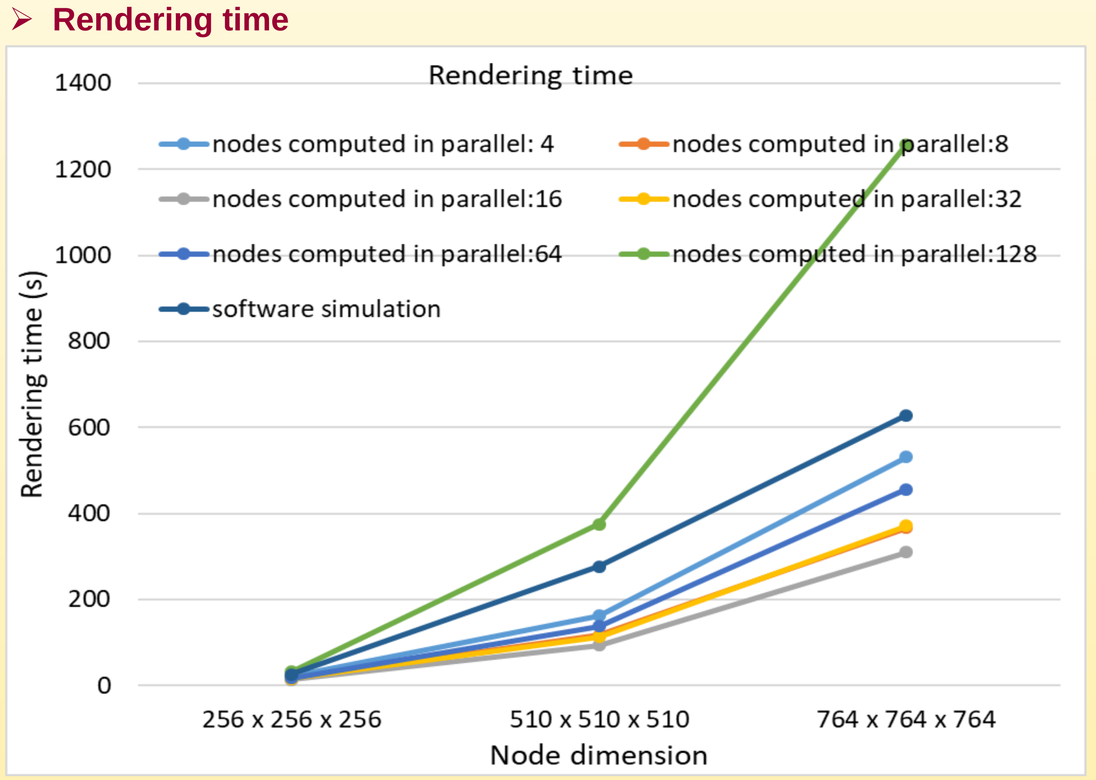
(iv) Performance Evaluation

- The performance of the proposed sound field rendering system is evaluated and compared with the software simulation on a desktop machine with 128 GB DDR4 memory and an Intel i7-7820X processor running at 3.6 GHz. The C++ codes in the software simulation are compiled by the gcc compiler with the option **-O3** and **-fopenmp** to use all eight cores in the processor.
- The reflection coefficient of boundaries is 0.95, the computed time steps are 1000. when the number of nodes computed in parallel is 16, the FPGA-based rendering system takes almost **half** of the rendering time, and **doubles** the computation throughput of the software simulations performed on the desktop.

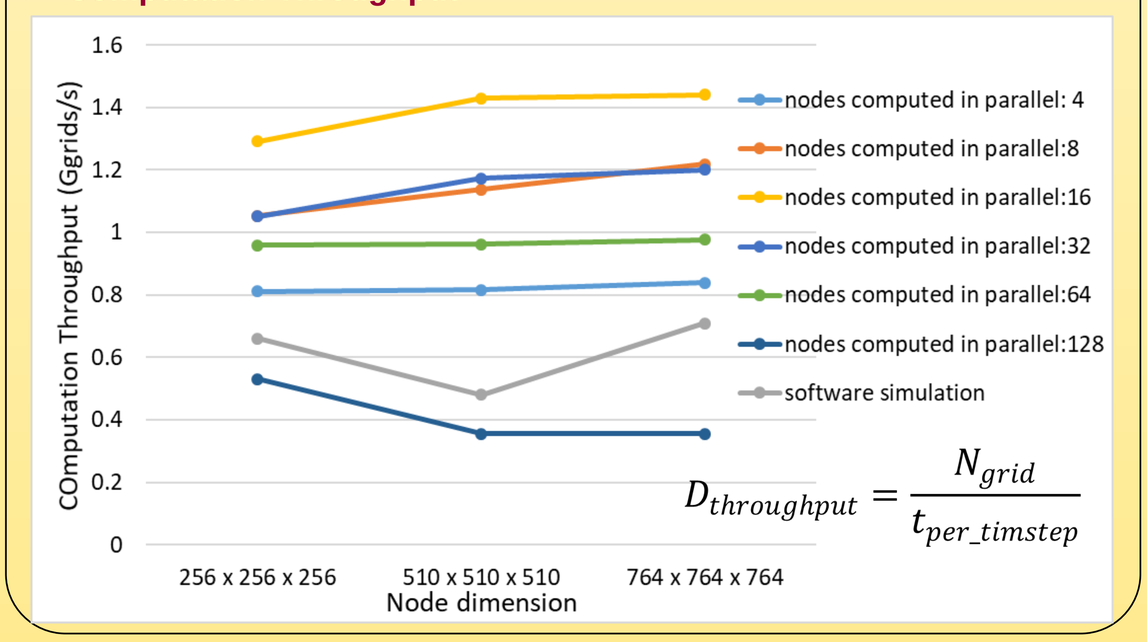
➢ **Technology Specification of Evaluation Environment**

	FPGA	CPU
Model	Arria 10 GX (10AX115N2F45E1SG)	Intel i7-7820X
Cores	1518 DSP blocks	8 cores
Clock frequency	267 MHz	3.6 GHz
On-chip memory	6.25 MB block RAMs	L1 cache: 256 KB L2 cache: 8 MB L3 cache: 11 MB
External RAMs	8 GB	128 GB
OS	CentOS 7.2	CentOS 7.2
Compiler	Intel SDK for FPGA 17.1	gcc 4.8.5
Program. lang.	OpenCL	C++

➢ **Rendering time**



➢ **Computation Throughput**



$$D_{throughput} = \frac{N_{grid}}{t_{per\_timstep}}$$

(v) Conclusions

A FPGA-based accelerator for sound field rendering is developed using high-level synthesis approach, in which the sliding-window-based data buffering scheme is applied to reduce the demand of memory bandwidth. Although the FPGA-based accelerator runs at 1/13 (0.267/3.6) of clock frequency and the memory size is about 1/16 (8/128) of those on the desktop machine, the FPGA-based accelerator doubles performance of the software simulations in the case of 16 nodes computed concurrently.

Acknowledgements

Thanks for Intel's donation of the FPGA board DE5a-NET and the related software tools through University Program. This work was supported by the I-O DATA Foundation and JSPS KAKENHI Grant Number JP19K12092.