

DHM AND FDTD BASED HARDWARE SOUND FIELD SIMULATION ACCELERATION

Yasushi Inoguchi, Tan Yiyu, Yukinori Sato

Center for Information Science, Japan
Advanced Institute of Science & Technology

Ishikawa, Japan

`inoguchi@jaist.ac.jp`

`yiyu-t@jaist.ac.jp`

`yukinori@jaist.ac.jp`

Yukio Iwaya, Hiroshi Matsuoka,

Research Institute of Electrical
Communication, Tohoku University

Sendai, Japan

`iwaya@riec.tohoku.ac.jp`

`matsuoka@riec.tohoku.ac.jp`

Makoto Otani

Faculty of Engineering, Shinshu University

Nagano, Japan

`otani@cs.shinshu-u.ac.jp`

Takao Tsuchiya

Faculty of Science and Engineering, Doshisha
University

Kyoto, Japan

`ttsuchiya@mail.doshisha.ac.jp`

ABSTRACT

Sound field simulation is widely used for acoustic design; however, this simulation needs many computational resources. On the other hand, FPGA becomes major for acceleration. To take advantage of hardware acceleration by FPGA, hardware oriented algorithm which consumes small number of gates and memory is necessary. This paper addresses hardware acceleration of sound field simulation using FPGA. Improved Digital Huygens Model (DHM) for hardware is implemented and speed up ratio is examined. For 2D simulation, the implemented accelerator is 1,170 times faster than software simulation. For 3D simulation, it is shown that FDTD based method is suitable for hardware implementation and required hardware resource are estimated.

1. INTRODUCTION

Simulation of sound field is useful for acoustic design. Recently, performance of computer system is much improved and many researches to simulate sound field numerically are proposed. Wave based numerical analyses such as finite element^[1], finite boundary element^[2], finite difference time domain (FDTD) method^[3-4] are major ways for simulation. Especially FDTD method is one of the best way to analyze sound field because this method calculates sound pressure at any place in the simulation field directly and it gives good accuracy. Recently, some of physical equivalent methods are also proposed^[5-7] to investigate acoustical behavior. These methods are derived from the wave equations of sound propagation, and Digital Huygen's Model (DHM) is one of these methods.

Concept of DHM have been initially proposed by S.A. Van Duyne and J.O. Smith^[6-7] and expanded by T. Tsuchiya et al.^[8-9] DHM is modification of FDTD for hardware implementation. It assumes that sound pressure pulse propagates between acoustic tubes to neighboring nodes. When a pulse is given to a node, scatters are sent to four adjacent nodes. Each node takes scatters from adjacent nodes and calculates its pressure. Iterating this procedure independently at each node, we can simulate sound pressures at all nodes. However, DHM also requires huge computational resources and it is difficult to simulate in real time.

On the other hand, recently Field Programmable Gate Array (FPGA) becomes popular to make a special purpose machines. Internal circuits of FPGA chip can be programmed by users, and hundreds percent of logical resources on a chip can be used for users' purpose. Several implementations using FPGA have been proposed^[10-12,15-17]. Chuan et al.^[10] reported 1.5 to 4 times acceleration and Motuk et al.^[11] reported 10 to 35 times acceleration.

Accelerating by GPGPU is another solution and several researches are reported^[13-14] but we don't treat GPGPU in this paper due to the limitation of pages.

In this research we introduce new implementation of real-time sound field simulation system using FPGAs. Hardware oriented two-dimensional DHM implementation and three-dimensional FDTD implementation are shown. Simulation by FPGA gives very good performance because all calculation is executed in a chip and all computing elements perform calculation simultaneously. Techniques to reduce gate consumption are also developed. Our system for 2D simulation is over than a thousand times faster than simulation by software.

Following is structure of this paper. Section 2 shows hardware implementation of a two-dimensional DHM algorithm, including algorithm optimization to reduce operations, and evaluation of calculation precision and speed. In section 3, three-dimensional FDTD implementation is discussed. Techniques to reduce circuit are introduced and number of chips to simulate a small chamber is shown. Chapter 4 is conclusion of this paper.

2. 2D DHM FOR HARDWARE IMPLEMENTATION

2.1. Basic of 2D DHM

The DHM is sound field simulation method for hardware. In the DHM, sound field is mapped on a grid, and each node of grid keeps its pressure. Figure 1 shows idea of the two-dimensional DHM. Figure 1 (a) shows that each node adjoins four neighbors by acoustic tubes which length is Δl . Assume that an incident is injected to a node. Then, sound pressure at the node is scattered through acoustic tubes as shown in figure 1(b). To make discussion easy, four adjacent tubes called as E, W, N, S, shown in figure 1 (c).

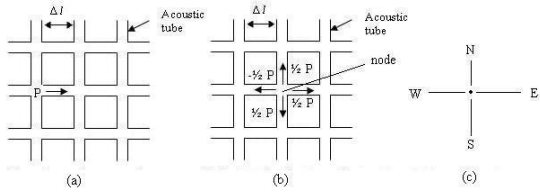


Figure 1: Concept of 2D Digital Huygen's Model. (a) shows mapping on a grid and unit length of an acoustic tube, (b) shows scattering at a node, and (c) shows direction definition.

A pressure at node (i,j) at time step n is denoted as $\phi^n(i,j)$, the incident will be transmitted to four neighboring nodes. Scattered pulse $S_m^n(i,j)$ from line $m \in \{E,W,N,S\}$ at the time step n at location (i,j) and sound pressure $\phi^n(i,j)$ are calculated by the following equations.

$$\begin{bmatrix} S_E^n(i,j) \\ S_W^n(i,j) \\ S_N^n(i,j) \\ S_S^n(i,j) \end{bmatrix} = \phi^n(i,j) - \begin{bmatrix} P_E^n(i,j) \\ P_W^n(i,j) \\ P_N^n(i,j) \\ P_S^n(i,j) \end{bmatrix} \quad (1)$$

$$\phi^n(i,j) = \frac{1}{2}(P_E^n(i,j) + P_W^n(i,j) + P_N^n(i,j) + P_S^n(i,j)) \quad (2)$$

where $P_m^n(i,j)$ are incident pulses. A scatter on each line becomes an incidence to the neighboring nodes. Thus, incidents at the next time step $n+1$ are calculated by following equation.

$$\begin{bmatrix} P_E^{n+1}(i,j) \\ P_W^{n+1}(i,j) \\ P_N^{n+1}(i,j) \\ P_S^{n+1}(i,j) \end{bmatrix} = \begin{bmatrix} S_E^n(i-1,j) \\ S_W^n(i+1,j) \\ S_N^n(i,j-1) \\ S_S^n(i,j+1) \end{bmatrix} \quad (3)$$

Through equations (1) to (3), scatters and pressure of all nodes can be calculated at every time step whose interval is $\Delta l/c$ where c is sound speed.

2.2. Hardware Resources to Implement 2D DHM

From point of view of implementation of this algorithm on a hardware circuit, you can see that only calculation of equations (1) and (2) are necessary. Equation (3) is implemented by variable renaming or wire connection and no operation is required. Four subtractors are needed to calculate equation (1), three adders and one right shifter are needed to calculate equation (2), and the right shifter to calculate $1/2$ in equation (2).

Each node must keep four incidences, four scatters and one sound pressure. However, either incidences or scatters are necessary because incidences are calculated by scatters of the neighboring nodes. Thus five temporary values must be kept at each node.

2.3. Hardware Oriented 2D DHM

Equations (1) and (2) are given by the original DHM algorithm. However, we can reduce number of operations of this algorithm. We eliminate incident $P_m^n(i,j)$ by inserting equations (1) and (3) into equation (2) and obtain a following equation.

$$\begin{aligned} \phi^n(i,j) = \frac{1}{2} & \left(\phi^{n-1}(i-1,j) + \phi^{n-1}(i+1,j) \right. \\ & \left. + \phi^{n-1}(i,j-1) + \phi^{n-1}(i,j+1) \right) \\ & + \sum_m S_m^{n-2}(i,j) \end{aligned} \quad (4)$$



Figure 2: Photograph of the hardware accelerator using FPGAs.

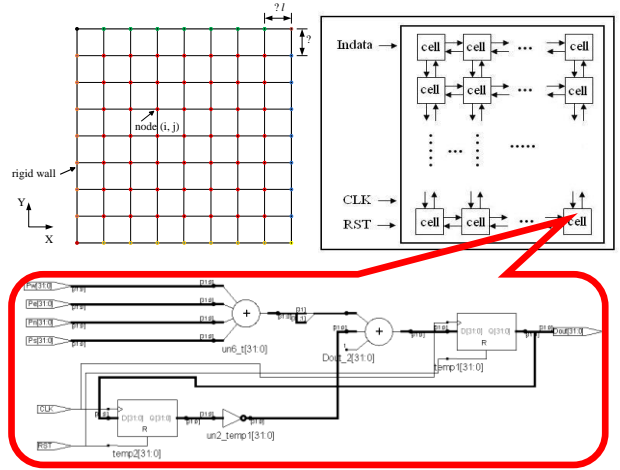


Figure 3: Structure of hardware accelerator. Sound pressure at each acoustic node is calculated by a computing cell synthesized on a FPGA chip.

Since scattering matrix is symmetrical, we can assume:

$$\sum_m S_m^n(i,j) = \sum_m P_m^n(i,j) \quad (5)$$

Using this relation, equation (4) can be written as

$$\begin{aligned} \phi^n(i,j) = \frac{1}{2} & \left(\phi^{n-1}(i-1,j) + \phi^{n-1}(i+1,j) \right. \\ & \left. + \phi^{n-1}(i,j-1) + \phi^{n-1}(i,j+1) \right) - \phi^{n-2}(i,j) \end{aligned} \quad (6)$$

Equation (6) shows that we need only three adders, one subtractor and one right-shifter. From point of view of memory, each node must keep two sound pressure (32bit) at each time steps.

2.4. Hardware Implementation of 2D DHM

Figure 2 shows the photograph of the hardware accelerator based on the 2D DHM. The system is SPP3000 provided by Tokyo Electron Device Ltd. and consists of several FPGAs and one CPU board to control system and program FPGAs. Each FPGA chip keeps a lot of calculation cells based on equation (6). In the figure, two inserted boards on the left side are FPGA boards and each board has two Xilinx XC5LVX330T-FF1738 FPGA chips and 512Mbit SRAM. This FPGA chip contains 51,840 Configu-

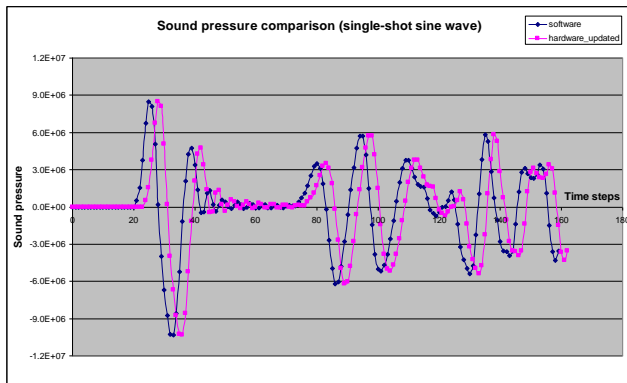


Figure 4: Sample of simulation result. Hardware simulated result is three-cycles delayed but both result are completely equal if software result puts off three cycles.

rable Logic Blocks (CLBs) and 11,664Kbit Block RAM. A board on the right is a CPU board which has single 1.4GHz Intel Pentium-M processor with 504MB memory. FPGA boards and host PC are connected by the compact PCI bus. Calculation for simulation is operated on the FPGA boards and host PC is used only for program and input/output data.

Figure 3 shows outline of our implementation. As shown in the left half of the figure, a sound field is mapped to a grid and each node on the grid calculates equation (6). Each node has actually implemented as calculation cell shown in right half of the figure. Calculation cell has a set of adders and registers. As shown in the schematic, one 4-inputs adder, one 3-inputs adder, and two 32-bit registers are used. A 4-inputs adder is actually implemented as three 2-inputs adders.

2.5. Precision Evaluation

To examine precision of the hardware simulation, we compared simulation result of hardware with software. Improved 2D DHM is programmed by C++ and executed by a software using an AMD Phenom 9500 1.8GHz Quad-core processor with 4GB memory. The simulation space is 32x32 and sampling rate is 16. Same algorithm is implemented on a FPGA system.

Incident point of signal is node (0,0) and simulation result is taken from node (6,15). In this evaluation, a single-shot sine wave and Gaussian wave are used and magnitude of them is 10^8 Pa. Both results of hardware and result of software are completely same. Figure 4 shows a sample of simulation result. Hardware result is delayed three clocks due to circuit initialization.

2.6. Evaluation of Simulation Speed

Table 1 shows comparison of simulation speed while 20,000 time steps. We examined three software simulation and a hardware simulation. In the table, “original DHM” shows time of simulation of DHM by software using equations (1) and (2). “Updated DHM” uses new DHM algorithm based on the equation (6) and its simulation time is almost half of original one. FDTD indicates the time of simulation using FDTD method. Hardware solution shows time of simulation based on equation (6) using FPGA. Hardware simulate system consist of two stages. One is calculation stage and another one is input/output (I/O) stage. This shows that calculation stage takes only 0.2ms and this speed is

Table 1: Comparison of simulation time for 2D simulation.

Nodes	Time steps	software solution			hardware solution
		original DHM	updated DHM	FDTD	updated DHM
10 x 10	20000	31ms	30ms	109ms	0.2ms (50ms)
25 x 25	20000	234ms	124ms	266ms	0.2ms (50ms)
32 x 32	20000	328ms	234ms	406ms	0.2ms (50ms)

almost 1,170 times faster than software simulation using same method. I/O stage takes much longer and time for simulation with I/O takes almost 50ms. Since I/O is only needed at initialization and finalization of a simulation, time for I/O can be ignored if the simulated time steps are much longer.

3. 3D FDTD FOR HARDWARE IMPLEMENTATION

3.1. 3D Hardware Acceleration of Sound Field Simulation

Although hardware implemented 2D-DHM is over a thousand times faster than software implementation, this scheme doesn't efficient at 3D. Division by 2 in equation (2) can be implemented as a 1-bit right-shifter and it is very simple circuit, however, division by three is needed for 3D and there is no simple way to implement this circuit. Thus, we use different scheme based on FDTD for 3D to eliminate division by three.

3.2. Basic Scheme of FDTD

Finite-difference time-domain (FDTD)^[3-4] method is another basic way to simulate sound field. In this section we introduce efficient architecture to implement FDTD on hardware system. Governing equations for linear sound propagation are

$$\frac{\partial \phi}{\partial t} = -\rho c^2 \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \right) \quad (7)$$

$$\frac{\partial u_x}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial x}, \quad \frac{\partial u_y}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial y}, \quad \frac{\partial u_z}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial z} \quad (8)$$

where ϕ is sound pressure, t is time, ρ is medium density and c is sound speed. Since discretized FDTD is well known, we introduce a hardware oriented FDTD as follows. Sound pressure ϕ at (i, j, k) at time step n is given by:

$$\begin{aligned} \phi^n(i, j, k) = & \phi^{n-1}(i, j, k) \\ & -\chi^2 \left\{ \bar{u}_x^{n-\frac{1}{2}} \left(i + \frac{1}{2}, j, k \right) - \bar{u}_x^{n-\frac{1}{2}} \left(i - \frac{1}{2}, j, k \right) \right. \\ & + \bar{u}_y^{n-\frac{1}{2}} \left(i, j + \frac{1}{2}, k \right) - \bar{u}_y^{n-\frac{1}{2}} \left(i, j - \frac{1}{2}, k \right) \\ & \left. + \bar{u}_z^{n-\frac{1}{2}} \left(i, j, k + \frac{1}{2} \right) - \bar{u}_z^{n-\frac{1}{2}} \left(i, j, k - \frac{1}{2} \right) \right\} \quad (9) \end{aligned}$$

$$\bar{u} = \frac{\rho c}{\chi} u \quad (10)$$

$$\Delta x = \Delta y = \Delta z = \Delta l, \quad \chi = \frac{c \Delta t}{\Delta l} = \frac{c}{\Delta l f} \leq \frac{1}{\sqrt{3}} \quad (11)$$

where Δt is the unit time, $\Delta x, \Delta y, \Delta z$ are the unit length.

$$\begin{aligned} \bar{u}_x^{n+\frac{1}{2}} \left(i + \frac{1}{2}, j, k \right) = & \bar{u}_x^{n-\frac{1}{2}} \left(i + \frac{1}{2}, j, k \right) \\ & - \{ \phi^n(i+1, j, k) - \phi^n(i, j, k) \} \quad (12) \end{aligned}$$

$\bar{u}_y^{n+\frac{1}{2}}$ and $\bar{u}_z^{n+\frac{1}{2}}$ are also calculated as well as equation (12).

3.3. Assumption for Hardware Implementation

When we assume $\chi = \frac{1}{\sqrt{3}}$, this is usually the most general assumption, solving equation (9) requires division by 3. However, from point of view of hardware implementation, a divider takes many clock cycles and large amount of gates, and spoils efficiency of acceleration.

Assume χ is 1/2 instead of $\frac{1}{\sqrt{3}}$, and this assumption satisfies condition (11). The coefficient in equation (9) is changed to 1/4. On hardware system, a divider by four is easily implemented by a 2-bit right-shifter.

3.4. Estimation of Required Hardware resources

We estimate exact value of $\chi, \Delta l$, and Δt . Sound speed c is 340m/s. In this implementation, maximum frequency to be simulated is 3KHz and we use 5 times over sampling to avoid error. Thus, Δt is $\frac{1}{3\text{KHz} \times 5} = 66.7\mu\text{s}$. In case of $\chi = \frac{1}{\sqrt{3}}$, Δl is 39.3mm and Δl is 45.3mm when $\chi = \frac{1}{2}$.

Now, we estimate the number of sound nodes to simulate a chamber which size is $2m \times 2m \times 4m$. 264k nodes are required when $\Delta l = 39.3\text{mm}$, and 172k nodes are required when $\Delta l = 45.3\text{mm}$, respectively. Our proposed assumption also reduces the number of sound nodes.

From viewpoint of memory consumption, each node keeps four variables: $p^n(i, j, k)$, $\bar{u}_x^{n+\frac{1}{2}}$, $\bar{u}_y^{n+\frac{1}{2}}$, and $\bar{u}_z^{n+\frac{1}{2}}$. Since each variable takes 32 bits, totally 256bits is needed for a node. Since XC5VLX330T has 11,664Kbit block RAM, one FPGA chip can accommodate 45.6K nodes in a chip. To simulate a chamber described above, 6 FPGA chips are needed if $\chi = \frac{1}{\sqrt{3}}$, and 4 chips are needed if $\chi = \frac{1}{2}$.

4. CONCLUSIONS

In this paper we proposed an implementation of hardware accelerated sound field simulator based on Digital Huygen's Model (DHM) for two-dimensional sound spaces. Improved DHM scheme requires very simple circuit and needs only four integer adders and a shifter. The proposed algorithm was implemented on a FPGA system and it was shown that hardware accelerated system is one thousand times faster than software simulation excluding I/O.

To simulate 3D, FDTD based implementation is better than DHM because FDTD based scheme can avoid synthesis of divider. From point of view of memory, it was shown that 45.6K nodes are accommodated in one FPGA chip.

To implement 3D simulator based on FDTD is our future work.

5. REFERENCES

- [1] J. N. Reddy, An Introduction to the Finite Element Method, 3rd ed., McGraw-Hill, USA, 2006.
- [2] P. K. Banerjee, and R. Butterfield, The Boundary Element Methods in Engineering Science, McGraw-Hill, USA, 1994.
- [3] D. Botteldooren, "Acoustical finite-difference time-domain simulation in a quasi-cartesian grid", Journal of the Acoustical Society of America, vol. 95, no. 5, pp. 2313–2319, 1994.
- [4] D. Botteldooren, "Finite-difference time-domain simulation of lowfrequency room acoustic problems", Journal of the

Acoustical Society of America, vol. 98, no. 6, pp. 3302–3308, 1995.

- [5] G. R. Campos and D. M. Howard, "On the computational efficiency of different waveguide mesh topologies for room acoustic simulation", IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5 pp. 1063-1072, 2005.
- [6] S. A. Van Duyne, and J. O. Smith III, "The 2D digital waveguide mesh", In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 1993, pp. 177–180.
- [7] S. A. van Duyne, J. R. Pierce, and J. O. Smith III. "Traveling Wave Implementation of a Lossless Mode-Coupling Filter and the Wave Digital Hammer". In Proc. Int. Computer Music Conf. (ICMC'94), pp. 411–418, 1994
- [8] Y. Kagawa, T. Tsuchiya, B. Fujii, and K. Fujika, "Discrete Huygens' model approach to sound wave propagation", Journal of Sound and Vibration, vol. 218, no. 3, pp. 419–444, 1998.
- [9] T. Tsuchiya, "Numerical simulation of sound wave propagation with sound absorption using digital Huygens' model", Japanese Journal of Applied Physics, vol. 46, no. 7B, pp.4809-4812, 2007.
- [10] Chuan He, Wei Zhao, and Mi Lu, "Time Domain Numerical Simulation for Transient Waves on Reconfigurable Coprocessor Platform", Proceedings of the 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'05)
- [11] E. Motuk, R. Woods, S. Bilbao and J. McAllister. "Design Methodology for Real-Time FPGA-Based Sound Synthesis", IEEE Trans. on Signal Processing, Vol. 15, No. 12, pp. 5833–5845, 2007
- [12] Campos, Guilherme; Barros, Sara, "The Meshotron: A Network of Specialized Hardware Units for 3-D Digital Waveguide Mesh Acoustic Model Parallelization", In Proc of 128th AES Convention, pp. 8054, 2010.
- [13] L. Savioja, "Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics", In Proc. Int. Conf. on Digital Audio Effects, 2010.
- [14] Craig J. Webb, and S. Bilbao, "Virtual Room Acoustics : A Comparison of techniques for computing 3D-FDTD schemes using CUDA", In Proc of the 130th Convention, 7 pages, 2011.
- [15] T. Tsuchiya, E. Sugawara and Y. Inoguchi, "On the numerical simulation of sound wave propagation using field programmable gate array device", Journal of the Acoustical Society of America, vol. 120, no. 5, pp. 3218-3218, 2006.
- [16] T. Yiyu, Y. Inoguchi, E. Sugawara, Y.i Sato, M. Ohya, Y. Iwaya, H. Matsuoka and T. Tsuchiya, "A FPGA Implementation of the Two-Dimensional Digital Huygens' Model", International Conference on Field-Programmable Technology (FPT'10), pp.304-307, Beijing, China, Dec. 8-10, 2010
- [17] T. Yiyu, Y. Sato, E. Sugawara, Y. Inoguchi, M. Ohya, Y. Iwaya, H. Matsuoka, T. Tsuchiy, "A real-time sound field renderer based on digital Huygens' model", submitted to Journal of Sound and Vibration.

6. ACKNOWLEDGEMENT

This research is supported by National Institute of Information and Communications Technology.