

Efficient Electromagnetic Simulation Including Thin Structures by Using Multi-GPU HIE-FDTD Method

Yuta Inoue and Hideki Asai

Research Institute of Electronics
Shizuoka University, Hamamatsu, 432-8561, Japan
inoue.yuta@shizuoka.ac.jp, asai.hideki@shizuoka.ac.jp

Abstract — This paper describes an efficient simulation method to solve the large scale electromagnetic problems with thin unit cells in the finite-difference time-domain (FDTD) simulation. The proposed method is based on the hybrid implicit-explicit and multi-GPU techniques, which can choose a larger time step size than that in the conventional one, and using the multiple graphic processing units (GPUs). In the proposed method, the computational time is significantly reduced.

Index Terms — Electromagnetic simulation, FDTD method, multi-GPU, time domain analysis.

I. INTRODUCTION

The finite-difference time-domain (FDTD) [1] method is one of the numerical simulation techniques for solving the electromagnetic problems. The FDTD method is a conditionally stable method. Therefore, the maximum time step size is constrained by the Courant-Friedrichs-Lewy (CFL) condition. If the time step size is not satisfied with the CFL condition, the FDTD method becomes unstable. For the analysis of the large scale electromagnetic problems with thin structures such as printed circuit boards, the time step size must be small and it can make the FDTD simulation a huge time consuming task. Thus, the efficient electromagnetic simulation technique is strongly demanded for the efficient designs. In order to overcome the CFL condition problem, the several unconditionally stable methods have been proposed for an arbitrary time step size [1], [2]. However, these methods are unsuitable for the parallel implementation because several overheads degrade the efficiency of parallel computing.

In order to alleviate the CFL condition problem, the hybrid implicit-explicit (HIE)-FDTD method has been proposed and studied for the fast electromagnetic simulation with thin unit cells in the FDTD simulation [3]-[7]. The HIE-FDTD method can adopt a larger time step size than that for the conventional FDTD method. The implicit technique is employed partially, and the computational domain can be easily divided for the

parallel computing. Therefore, the message passing interface (MPI)-based parallel-distributed HIE-FDTD method [6] and the general purpose computing on graphic processing unit (GPGPU) based massively parallel HIE-FDTD method [7] have been proposed for the efficient simulation. However, the parallel distributed HIE-FDTD method is required to be faster since CPU is slower than the graphic processing unit (GPU). On the other hand, the memory size of the GPU boards is not sufficient for the large scale problems. Hence, the combination of the MPI based method and the GPGPU based method is demanded for solving the large scale problems. As a result, the proposed method can solve the large scale problems and can reduce the elapsed time drastically from MPI-based and single GPU-based method.

In this paper, the multi-GPU based HIE-FDTD method with MPI and CUDA is proposed for the efficient electromagnetic simulation of the object with thin structures. First, the HIE-FDTD method is reviewed briefly. Next, the proposed method is described. Finally, the efficiency of the proposed method is evaluated through several FDTD simulations.

II. HIE-FDTD METHOD

The HIE-FDTD method [4] has been proposed for the efficient 3-D electromagnetic simulation of the given object with thin unit cells in the FDTD-based computational domain. Here, it is assumed that the given object has the fine scale dimension in the z direction such as printed circuit boards. In such a case, the updating formulas of the HIE-FDTD method consist of the two explicit equations which do not contain the derivatives with respect to z direction and four implicit equations which include the derivatives with respect to z direction. The updating formulas are described in [6].

The updating procedures of the HIE-FDTD method are partially different from the conventional FDTD method. First, E_z and H_z are explicitly updated. Next, E_x and E_y are updated by numerical solution method of simultaneous linear equations such as LU decomposition

method. After updating E_x and E_y , H_x and H_y are explicitly updated.

The CFL condition of the HIE-FDTD method is alleviated than that of the conventional FDTD method, which is given by:

$$\Delta t_{FDTD} \leq \frac{1}{c\sqrt{\Delta x_{\min}^{-2} + \Delta y_{\min}^{-2} + \Delta z_{\min}^{-2}}}, \quad (1)$$

where Δt_{FDTD} is maximum time step size for the conventional FDTD method, c is the speed of light, Δx_{\min} , Δy_{\min} and Δz_{\min} are minimum cell sizes along the x , y , and z direction in the computational domain. The CFL condition of the HIE-FDTD method is determined as follows:

$$\Delta t_{HIE} \leq \frac{1}{c\sqrt{\Delta x_{\min}^{-2} + \Delta y_{\min}^{-2}}}, \quad (2)$$

where Δt_{HIE} is maximum time step size for the HIE-FDTD method. From (2), the Δz is removed from the CFL condition of the HIE-FDTD method. Therefore, the time step size can choose the larger time step size than that for the conventional FDTD method. In the case of thin cell along with z direction, the HIE-FDTD method can efficiently simulate.

III. MULTI-GPU HIE-FDTD METHOD

The multi-GPU HIE-FDTD method is combination of the parallel distributed HIE-FDTD method and the GPGPU HIE-FDTD method. In the proposed method, the arithmetic operations are performed by a GPU instead of a CPU. In order to employ multi GPUs, the proposed method uses the three types of domain decomposition techniques. One is the domain decomposition technique for allocating a GPU to a subdomain. The others are the domain decomposition techniques for the GPU computing. In this section, the domain decomposition techniques and updating procedure are described. Here, the MPI library is employed for network communication between the neighboring subdomains and the CUDA is done for the GPU computing.

A. Domain decomposition

First, the domain decomposition technique for allocating a GPU to a subdomain is described. In the proposed method, the whole 3D spatial domain is divided into the several subdomains along the x and y directions. The number of subdomains is same as the number of GPUs. Note that the boundary cells of subdomain are overlapped with the neighboring subdomains. The overlapping boundary cells are employed to communicate magnetic components between the neighboring subdomains. Furthermore, the dummy cells are added to each subdomain in order to correct the total number of cells of x - y plane to a multiple of 64. The dummy cells are employed for the GPU

computing. In the updating process, the electromagnetic components at the dummy cells are not updated.

Next, the domain decomposition techniques for GPU computing are shown. The partitioned subdomains for GPU computing are illustrated in Fig. 1. Here, NX , NY , and NZ are the numbers of cells for the x , y , and z directions, respectively. In the proposed method, the domain decomposition techniques are different in the explicit updating procedure and the implicit updating procedure. Figure 1 (a) shows the divided subdomain for explicit updating procedure. The subdomain is partitioned by the blocks which are composed of the 64 threads. A cell is allocated to a thread and is updated by the thread. The thread is smallest element of the process. Therefore, in the explicit updating procedure, the subdomain is divided into $(NX \times NY \times NZ)/64$ blocks. On the other hand, the domain decomposition technique for implicit updating procedure is illustrated in Fig. 1 (b). In the implicit updating procedure, x - y plane of the sub-domain is partitioned by the blocks. Thus, each block is assigned to $64 \times NZ$ cells. Each thread in the block is allocated to the NZ cells. These cells are updated by using the LU decomposition method [8].

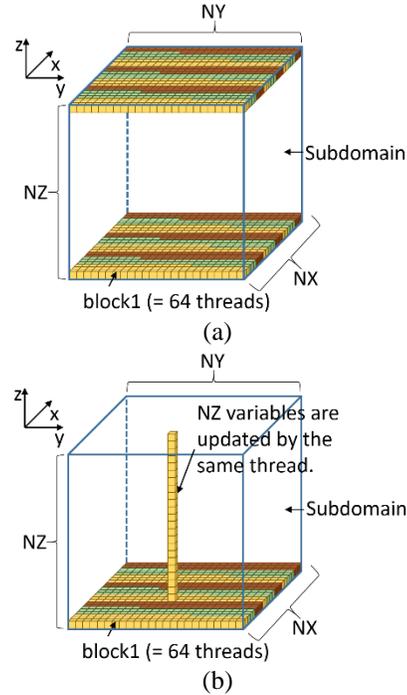


Fig. 1. The domain decomposition and block structure for GPU computing. (a) Domain decomposition for explicit updating procedure, and (b) domain decomposition for implicit updating procedure.

B. Updating procedure

In the proposed method, two types of updating procedures are employed. One is the updating procedure which invokes the blocking data communication

function. The other is the updating procedure which calls the nonblocking data communication function. In the case of invoking the blocking data communication function, each magnetic component at boundary cells is communicated after updating each magnetic component. On the other hand, in the case of calling the nonblocking data communication function, the updating process is divided into two parts. One is the boundary part. The other is except the boundary part. Figure 2 shows the pseudo-code of updating procedure for calling the nonblocking data communication function. From Fig. 2, the data communication and the computation is overlapped by the nonblocking data communication function. Therefore, it is efficiently performed than invoking the blocking data communication function.

```

Transient()
{
  COMMUNICATE Hx and Hy of boundary part
  between neighboring subdomains
  WHILE Current Time < Ending Time
    COMPUTE Hz of the boundary part
    COMMUNICATE Hz of boundary part between
    neighboring subdomains
    COMPUTE Hz except boundary part
    COMPUTE Ez except boundary part
    WAIT for completion of Hx and Hy of boundary
    part communicate
    COMPUTE Ez of boundary part
    COMPUTE Ex and Ey except boundary part
    WAIT for completion of Hz of boundary part
    communicate
    COMPUTE Ex and Ey of boundary part
    COMPUTE Hx and Hy of boundary part
    COMMUNICATE Hx and Hy of boundary part
    between neighboring subdomains
  ENDWHILE
}

```

Fig. 2. Pseudo-code of the proposed method.

IV. NUMERICAL RESULTS

First, in order to estimate the accuracy of the proposed method, the simulation has been performed for multi conductor transmission lines illustrated in Fig. 3. Each transmission line is terminated with the resistor ($100\ \Omega$). The voltage source is appended to the near end of the trace2. A pulse excitation with 0.5×10^{-9} sec rise/fall time, a width 4×10^{-9} sec, a period 1×10^{-8} sec, and an amplitude 3.3V was used. Mur's first order absorbing boundary condition is used for the absorbing boundary condition. The computational domain consists of $46 \times 40 \times 50$ cells and discretized by nonuniform meshes. The minimum cell sizes are $\Delta x = 0.2\text{mm}$, $\Delta y = 1\text{mm}$, $\Delta z = 0.01\text{mm}$, respectively. The time step size is 3.33×10^{-14} sec in the FDTD method and 6.53×10^{-13} sec

in the HIE-FDTD method. All of the simulations are performed by PC cluster, which is composed of two PCs. These PCs are connected by the gigabit Ethernet. Each PC has an Intel Xeon E5-2650 2GHz and four GPU boards, which are Tesla C2075. Tesla C2075 is one of the GPU boards for the high performance computing. In this simulation, the Intel Xeon E5-2650 was used for the FDTD method and the HIE-FDTD method and eight GPU boards were used for the proposed method. Open MPI is employed for MPI library. Figure 4 shows waveform results of the far end of the trace2 and the trace3. From Fig. 4, the waveform results show good agreement between these methods.

Next, in order to verify the efficiency of the proposed method, the large scale problem has been performed. The number of cells is $1270 \times 1270 \times 102$ cells. The minimum cell sizes are $\Delta x = \Delta y = 1\text{mm}$, $\Delta z = 0.01\text{mm}$. Mur's first order absorbing boundary condition is adopted. The time step size for the FDTD method is 3.33×10^{-14} sec. That for the HIE-FDTD method is 1.89×10^{-12} sec. Table 1 shows the simulation results, which are the elapsed time and the speed up ratio, by the FDTD method, the multi-GPU FDTD method, the HIE-FDTD method and the proposed method. The proposed method is performed with single precision floating point and double precision floating point. The peak performance of the Tesla C2075 by using the single precision floating point is two times faster than the peak performance by using the double precision floating point. In the bracket, the communication time is described. From Table 1, the proposed method is about 4000 times faster than the FDTD method by using the 8 GPUs with single precision floating point.

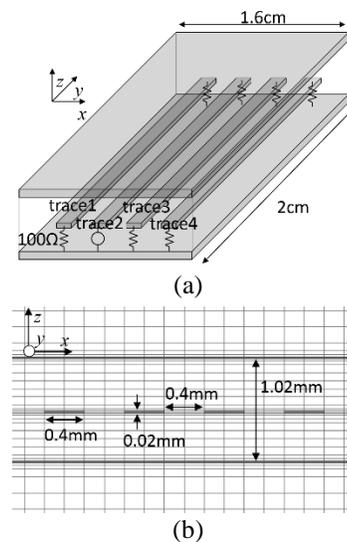


Fig. 3. Example printed circuit board: (a) overhead view of the example circuit, and (b) cross section view of the example circuit.

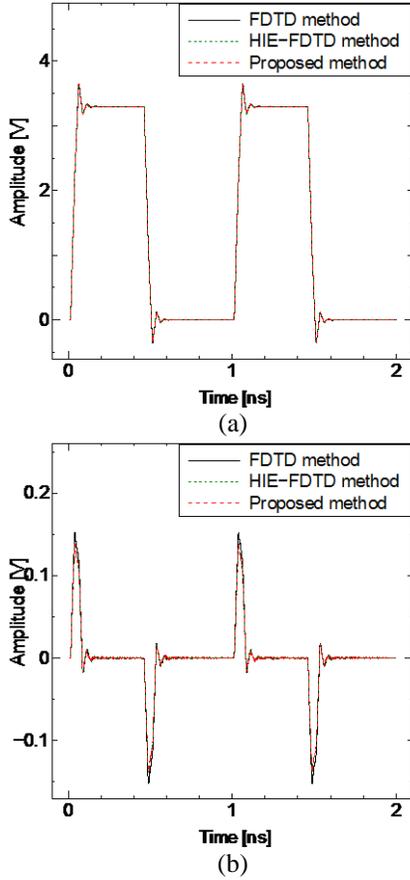


Fig. 4. Waveform results: (a) far end of the trace2, and (b) far end of the trace3.

Table 1: Elapsed time and speed up ratio

	Elapsed Time (sec)	Speed up Ratio (vs FDTD Method)
FDTD method	499395.0	1.00
Multi-GPU FDTD method	6189.88	80.68
HIE-FDTD method	15110.43	33.05
Proposed method (double precision and blocking)	287.39 [221.23]	1737.69
Proposed method (single precision and blocking)	145.62 [117.33]	3429.44
Proposed method (double precision and nonblocking)	244.86	2039.51
Proposed method (single precision and nonblocking)	126.64	3943.42

V. CONCLUSION

The multi-GPU HIE-FDTD method proposed for the efficient simulation of the large-scale electromagnetic problems including thin structures. In the case of suitable given objects for the HIE-FDTD method, it has been verified that the proposed method is more than about 4000 times faster than the conventional FDTD method in the case of using 8 GPUs with single precision floating point.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 24300018.

REFERENCES

- [1] A. Taflov and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, Inc., Norwood, 2005.
- [2] Y. Yang, R. S. Chen, and E. K. N. Yung, "The unconditionally stable Crank-Nicolson FDTD method for three-dimensional Maxwell's equations," *Microw. Opt. Tech. Lett.*, vol. 48, no. 8, pp. 1619-1622, May 2006.
- [3] J. Chen and J. Wang, "A three-dimensional semi-implicit FDTD scheme for calculation of shielding effectiveness of enclosure with thin slots," *IEEE Trans. Electromagn. Compat.*, vol. 49, no. 2, pp. 354-360, May 2007.
- [4] M. Unno and H. Asai, "HIE-FDTD method for hybrid system with lumped elements and conductive media," *IEEE Microw. Wireless Compon. Lett.*, vol. 21, no. 9, pp. 453-455, Sep. 2011.
- [5] H. Muraoka, Y. Inoue, T. Sekine, and H. Asai, "A hybrid implicit-explicit and conformal (HIE/C) FDTD method for efficient electromagnetic simulation of nonorthogonally aligned thin structures," *IEEE Trans. Electromagn. Compat.*, vol. 57, no. 3, pp. 505-512, June 2015.
- [6] Y. Inoue and H. Asai, "Fast fullwave simulation based on parallel distributed HIE-FDTD method," *IEEE APMC 2012*, Kaohsiung, Taiwan, pp. 1253-1255, Dec. 2012.
- [7] M. Unno, S. Aono, and H. Asai, "GPU-based massively parallel 3-D HIE-FDTD method for high-speed electromagnetic field simulation," *IEEE Trans. Electromagn. Compat.*, vol. 54, no. 4, pp. 912-921, Aug. 2012.