DESIGN STUDY OF DEDICATED COMPUTER SYSTEM FOR WAKE FIELD ANALYSIS WITH TIME DOMAIN BOUNDARY ELEMENT METHOD*

Kazuhiro Fujita[†#], Hideki Kawaguchi[††], and Takeaki Enoto[†]
[†]Hokkaido University, Sapporo, 060-8628, Japan
[††]Muroran Institute of Technology, Muroran, 050-8585, Japan

Abstract

Time domain boundary element method (TDBEM) has advantages of dispersion free calculations and modelling of curved beam trajectories in wake field analysis compared to conventional methods. These advantages give us useful possibilities for analysis of beam dynamics due to CSR in bunch compressors of next-generation accelerators. On the other hand, the TDBEM also has a serious difficulty of large computational costs. In this paper, a dedicated computer system for wake field analysis with the TDBEM is proposed as one of solutions for high performance computing (HPC) technologies. A system design and the VHDL logic simulations of the TDBEM dedicated hardware are demonstrated.

INTRODUCTION

A Time Domain Boundary Element Method (TDBEM) for wake field analysis has been successfully developed by being free from numerical instability in the last several years [1]. The TDBEM has advantages of dispersion free calculations and modelling of curved beam trajectories in wake field analysis compared to conventional methods. These advantages give us useful possibilities [2] for quantitative prediction of curved beam dynamics such as CSR in bunch compressors of next-generation accelerators. On the other hand, the TDBEM scheme requires very large memory and costly computational efforts, especially for full 3D simulations, and therefore the wake field TDBEM analysis for many practical cases must be done on high performance computing (HPC) environments such as Supercomputers. Then superscalar parallel architecture computers or grid computer systems are not suitable for implementation of the TDBEM scheme since performance gains saturate with increasing numbers of processors in the CPUs due to the interprocessor communication overhead. Although supercomputers with vector computation architecture are able to effectively execute the TDBEM calculations, there exist situations of prohibitive access costs and limited availability for computer users. Then one more possibility of HPC technologies is dedicated computer architectures. The dedicated computers achieve ultra-high speed calculations by constructing a dedicated hardware for a numerical scheme. In addition, recent remarkable progress of LSI technologies, especially rewritable LSI hardware design environments such as the HDL compiler tools and large scale FPGA, enable us to develop computer hardware systems at very low cost in short development period. In fact, the hardware-based implementation of electromagnetic field simulation has been actively discussed in recent several years [3, 4].

From this background, we propose a design study of a dedicated computer system of the TDBEM scheme for large-scale 3D wake field analysis.

*Work supported by JSPS Grants-in-Aid for Scientific Research (No.17540225)
[#]fujita@eng.hokudai.ac.jp

DEDICATED COMPUTER SYSTEM FOR TDBEM SCHEME

TDBEM scheme

The TDBEM scheme is based on the following Kirchhoff's boundary integral equation [1]:

$$\mathbf{B}(t,\mathbf{r}) = \mathbf{B}_{self}(t,\mathbf{r}) - \frac{1}{4\pi}\int_S \left[\frac{(\mathbf{r}-\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|^3} + \frac{(\mathbf{r}-\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|^2} \times \frac{\partial}{c\partial t} \right] (\mathbf{B}(t',\mathbf{r}') \times \mathbf{n}')dS', \quad (1)$$

where $\mathbf{r}$ is the observation point in the bounded analytical region, $\mathbf{r}'$ is the point on the boundary surface S, t' is the retarded time defined by $t' = t - |\mathbf{r} - \mathbf{r}'|/c$, $\mathbf{n}'$ is the unit normal vector on the boundary surface. In this equation, the total magnetic fields $\mathbf{B}$ are explicitly split into the self-fields of the charged particle beams $\mathbf{B}_{self}$ and the boundary integral term. The integrand $\mathbf{B}\times\mathbf{n}$ corresponds to the induced surface current density, and the wake fields are described by the boundary integral term. Accordingly the TDBEM can easily treat wake fields produced by curved trajectory charged particles.

Then discretization of (1) results in the following matrix equation:

$$[G_0][B^n] = [B_{self}{}^n] - \sum_{l=1}^{L}[G_l][B^{n-l}] \quad (2)$$

where $[B^l]$ denotes an unknown vector which consists of tangential components of magnetic fields on the boundary elements at time $t = l\Delta t$ (l = 0, ···, L), $[G_l]$ denotes a coefficient matrix determined by the boundary integral of (1) on S, L is the total number of the matrices of $[G_l]$, $[B_{self}{}^n]$ is a given vector calculated from the inner products of the tangential unit vectors and the self-fields.

The figure 1 shows one time step calculation process of the TDBEM system matrix equation (2) which consists of the following four procedures:

I. Multiplications of matrix $[G_l]$ and vector $[B^{n-l}]$
II. Summation of the result vectors $[G_l]$ $[B^{n-l}]$
III. Matrix inversion with respect to $[G_0]$
IV Shifting the present solution by one time step past

The boundary values over all time steps can be obtained by iteratively solving (2) according to these procedure.

Once the boundary values over all time steps are calculated, the wake fields at any position inside the domain surrounded by *S* can be obtained from (1).

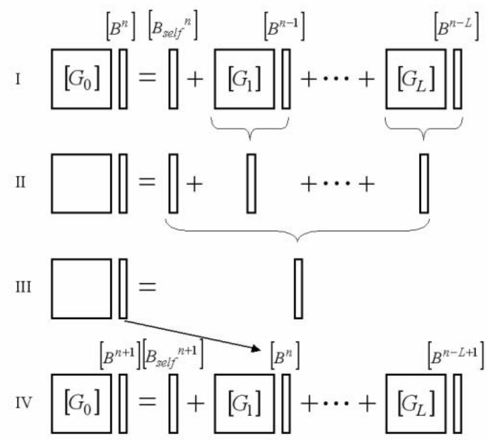


Fig. 1: Calculation process of TDBEM system matrix equation: I. Matrix-vector multiplication, II. Vector data addition, III. Matrix inversion, and IV. Boundary value shift by one time step past.

Overall System of TDBEM Machine

The matrix-vector multiplications in the right-hand side of (2) are the most costly calculation process. However, since the calculation itself is simple operation, it can be readily implemented on hardware. For this reason, we design the whole architecture of the TDBEM dedicated computer system as shown in Fig. 2. It consists of two parts: a PC for pre-processing and post-processing which are complicated but light calculations, and a dedicated hardware for matrix calculation processing which are simple but heavy calculations as in Fig. 1. First, setting up simulation parameters (e.g., bunch size and geometry data), mesh generation of a numerical model and matrix equation construction are performed in the PC. Next, the calculated matrix data is transferred into the dedicated hardware from the PC, and then boundary value vectors over all time steps are computed there. Finally, their boundary value vectors are uploaded to the PC, and the interior field calculation and pre-processing are performed.

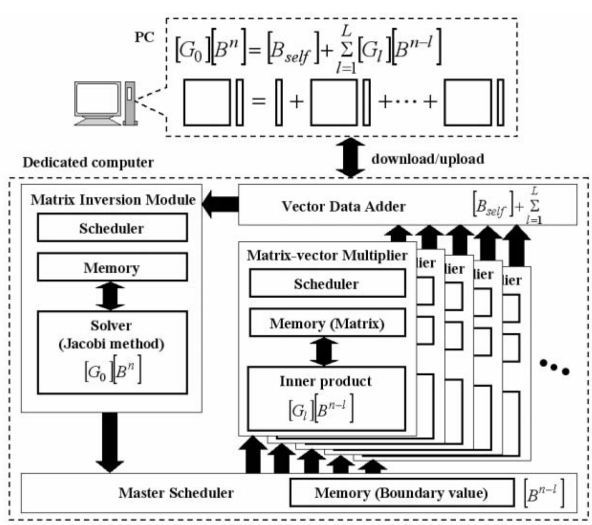


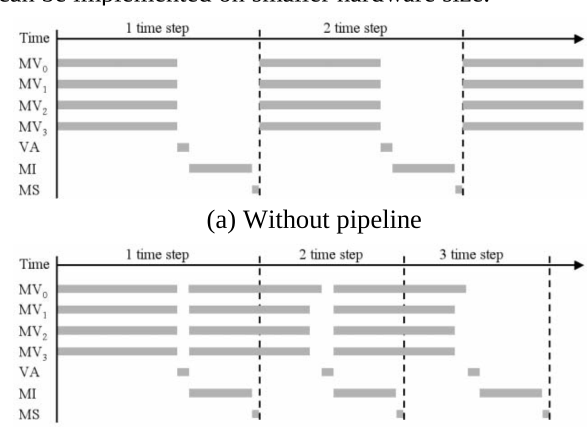
Fig. 2: Overview of dedicated computer system for TDBEM scheme.

Dedicated Hardware Architecture

A dedicated hardware for (2) is designed based on the four steps shown in Fig. 1. Therefore, it consists of the following four modules (see Fig.2); The first module is a matrix-vector multiplier (MV) module which calculates $[G_l][B^{n-l}]$ in (2). The second one is a vector data adder (VA) summing up the self-field vector $[B_{self}{}^n]$ and the resulting vectors $[G_l][B^{n-l}]$ from the MV module. The third is a matrix inversion (MI) module which performs matrix inversion with respect to $[G_0]$ to use the inhomogeneous term vector from the VA module. The final module is a master scheduler (MS) that controls overall operations and manage memory allocation of all boundary values at each time step.

Then, to construct the individual hardware modules for each process I - IV shown in Fig.2, parallelization and pipelining of the calculation process of (2) are also possible at module level. Since processes of MVs in (2) are independent each other, the parallelization of MVs is possible by preparing a numbers of multipliers as in Fig. 2. An example of parallelized processing with four MVs is shown in Fig. 3(a), in which we can find that all MV modules are operated in parallel. In addition to this parallelization of MV, in the calculation for a time step *n*, pipelining calculation between the MVs and the MI module is possible except for the multiplication of the matrix $[G_1]$ and the vector $[B^n]$ at the next time step n+1 as in Fig. 3(b).

In this case, since the total computational time for a time domain simulation mainly depends on the calculation at MVs and this processing term overlaps to that of the MI process, therefore any first matrix inversion algorithm (ex. ICCG) is not necessary. From this reason, we can adopt simple matrix inversion algorithm which can be implemented on smaller hardware size.



(a) Without pipeline

(b) With pipeline

Fig. 3: Timing chart of the dedicated hardware modules. MV0, MV1, MV2, and MV3: matrix-vector multipliers, VA: vector data adder, MI: matrix inversion module, MS: master scheduler.

VHDL SIMULATION

As the first step of development of the TDBEM dedicated computer, the logic circuit of the TDBEM dedicated hardware is designed with the hardware description language VHDL. As a test example, we design the pipelined TDBEM dedicated hardware including four MVs, VA, MI and MS modules. Then the MI module is constructed based on the Jacobi method. For checking the hardware operation we simulate a test example shown in Fig. 4, and demonstrates its logic circuit simulation result at the forth time step in Fig. 5. We can find good agreements between the VHDL logic circuit simulation result and the software one with the Mathematica. It is confirmed that the designed TDBEM dedicated hardware is correctly designed.

CONCLUSION

To aim for future large scale 3D CSR simulations, a dedicated computer system of the TDBEM wake field analysis has been proposed. The dedicated computer system consists of a PC for pre-processing and post-processing and a dedicated hardware for matrix computations of the TDBEM scheme. The TDBEM dedicated hardware can be parallelized and pipelined at module level. As the first step in the development of the dedicated computer system, the TDBEM dedicated hardware was designed in the VHDL. The hardware operation is checked by the logic simulation.

REFERENCES

[1] K. Fujita, H. Kawaguchi, I. Zagorodnov, and T. Weiland, IEEE Trans. Nucl. Sci., vol. 53, no. 2, pp. 431-439, 2006.

[2] K. Fujita, H. Kawaguchi, to be presented at EPAC 2006, WEPCH131.

[3] S. Matsuoka, K. Ohmi, and H. Kawaguchi, IEICE Trans. Electron., vol. E86-C, no. 11, 2003.

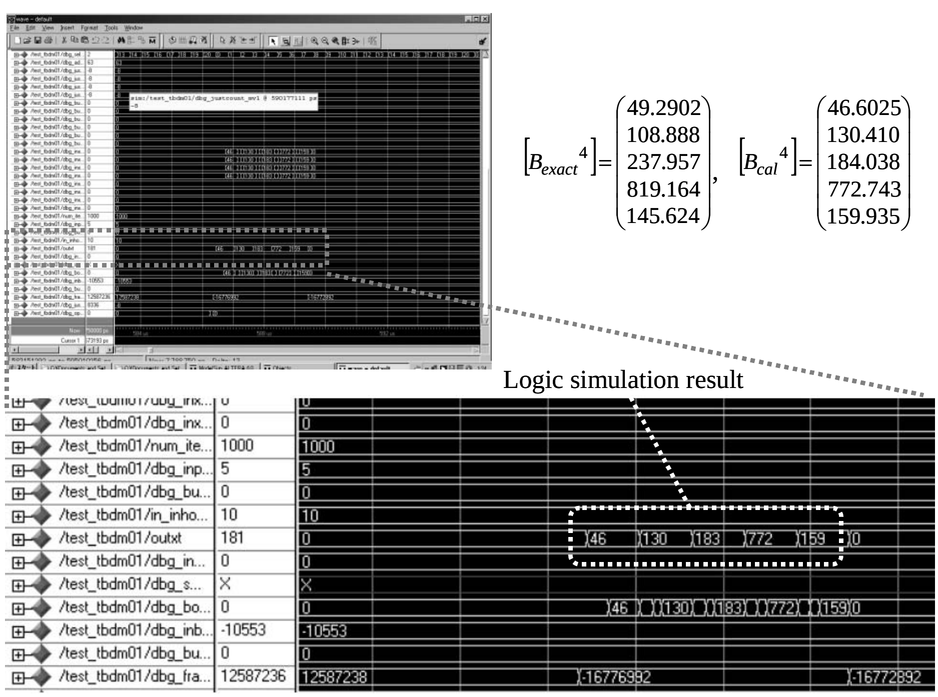
[4] P. Placidi, L. Verducci, G. Matrella, and L. Rosselli, IEICE Trans. Electron., vol. E85-C, March, 2002.

$$[G_0]=\begin{pmatrix} 4 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 4 \end{pmatrix}, \quad [G_1]=\begin{pmatrix} 0 & -13 & 3 & 0 & 0 \\ 15 & 0 & -22 & 10 & 0 \\ -6 & 41 & 0 & -41 & 6 \\ 0 & -10 & 22 & 0 & -15 \\ 0 & 0 & -3 & 13 & 0 \end{pmatrix}$$

$$[G_2]=\begin{pmatrix} 0 & 0 & 10 & 7 & 0 \\ 0 & 0 & 0 & 20 & -3 \\ 10 & 0 & 0 & 0 & 10 \\ -3 & -20 & 0 & 0 & 0 \\ 0 & 7 & 10 & 0 & 0 \end{pmatrix}, [G_3]=\begin{pmatrix} 0 & 0 & 0 & -5 & 3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 3 & -5 & 0 & 0 & 0 \end{pmatrix}, [G_4]=\begin{pmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$[B_{self}{}^1]=\begin{pmatrix} 100 \\ 350 \\ 132 \\ -581 \\ 690 \end{pmatrix}, [B_{self}{}^2]=\begin{pmatrix} 1816 \\ 120 \\ -9237 \\ 3028 \\ 1999 \end{pmatrix}, [B_{self}{}^3]=\begin{pmatrix} -2259 \\ -1486 \\ 12244 \\ 3115 \\ -3056 \end{pmatrix}, [B_{self}{}^4]=\begin{pmatrix} -3102 \\ -3540 \\ -6501 \\ 1403 \\ 4567 \end{pmatrix}$$

Fig. 4: Test example of matrix equation.



$$[B_{exact}{}^4]=\begin{pmatrix} 49.2902 \\ 108.888 \\ 237.957 \\ 819.164 \\ 145.624 \end{pmatrix}, \quad [B_{cal}{}^4]=\begin{pmatrix} 46.6025 \\ 130.410 \\ 184.038 \\ 772.743 \\ 159.935 \end{pmatrix}$$

Fig. 5: Logic simulation of a designed TDBEM dedicated hardware at the forth time step. The logic simulation result $[B_{logic}{}^4] = {}^t(46, 130, 183, 772, 159)$. $[B_{exact}{}^4]$ and $[B_{cal}{}^4]$ show exact solution and calculation results by the Mathematica. The difference between the logic simulation result and $[B_{exact}{}^4]$ is due to numerical truncation caused by the integer operation in the designed TDBEM dedicated hardware.