# Design study of ultrahigh-speed microwave simulator engine

| | |
|---|---|
| メタデータ | 言語: English<br>出版者: IEEE<br>公開日: 2007-08-23<br>キーワード (Ja):<br>キーワード (En): finite difference methods, logic circuits, microwave propagation, numerical analysis<br>作成者: 川口, 秀樹, 高原, 健爾, YAMAUCHI, Daisuke<br>メールアドレス:<br>所属: |
| URL | http://hdl.handle.net/10258/219 |

# Design study of ultrahigh-speed microwave simulator engine

| | |
|---|---|
| 著者 | KAWAGUCHI Hideki, TAKAHARA Kenji, YAMAUCHI Daisuke |
| journal or publication title | IEEE transactions on magnetics |
| volume | 38 |
| number | 2 |
| page range | 689-692 |
| year | 2002-03 |
| URL | http://hdl.handle.net/10258/219 |

doi: info:doi/10.1109/20.996179

# Design Study of Ultrahigh-Speed Microwave Simulator Engine

Hideki Kawaguchi, *Member, IEEE*, Kenji Takahara, and Daisuke Yamauchi

***Abstract*—A design study of microwave simulator engine is presented in this paper. Taking note the simplicity and duality of data flow in finite-difference time-domain (FDTD) scheme, conceptual and hardware designs of the engine are shown for two-dimensional wave phenomena. To store field values in individual SRAMs, efficient use of digital hardware resources is achieved and the engine is constructed by very small size hardware. Based on the design study, a prototype hardware is made and basic operation is confirmed.**

***Index Terms*—Finite difference methods, logic circuits, microwave propagation, numerical analysis.**

## I. Introduction

According to rapid progress of microwave devices such as portable phones, demand of numerical microwave simulator is also increasing now for shortening the design term. The finite-difference time-domain (FDTD) method is then suitable for such simulations. Simplicity of the FDTD scheme is directly connected to high-performance calculation and small-size memory. Grid-based discretization in FDTD enables us to readily make up the numerical model of complicated three-dimensional (3-D) geometry. And the time-domain calculation scheme of FDTD gives us many possibilities to treat various kinds of materials and circumstances. For example, dispersive material also can be treated in FDTD by combining with equation of motion of materials [1].

On the other hand, although recent PCs show us remarkable progress in their performance, it is still not enough to simulate a whole system of microwave products in many cases. In general, there exist two ways to treat extremely big problems. One is enhancement of hardware performance as in supercomputer, and another is software methodological effort such as the subgrid technique [2]. Here, noticing more rapid progress of electronic parts technology such as RAM comparing with performance of a whole PC system, one thinks of an idea of hardware engine for the microwave simulator as the third possibility [3]. One big advantage of hardware engine is no time delay in memory access, which makes hardware simulations very efficient. The FDTD scheme is suitable for this hardware engine. The FDTD basic scheme consists of quite simple addition/subtraction operations and there exists clear duality between electric and magnetic fields equations. These factors make hardware small scale and simple structure. The cubic grid expression of numerical objects can be naturally implemented in 0/1 digital logic. Considering this situation, a design study of the FDTD microwave simulator engine for two-dimensional electromagnetic wave propagation phenomena is presented in this paper. The presented design shows us that the engine can be achieved by much smaller size hardware than the presented one [3].

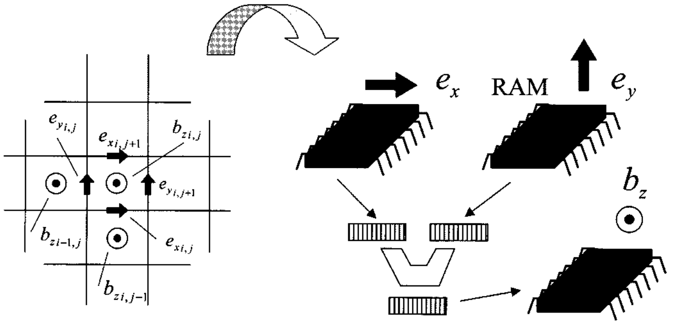


Fig. 1. 2–D FDTD grid and concept of hardware engine.

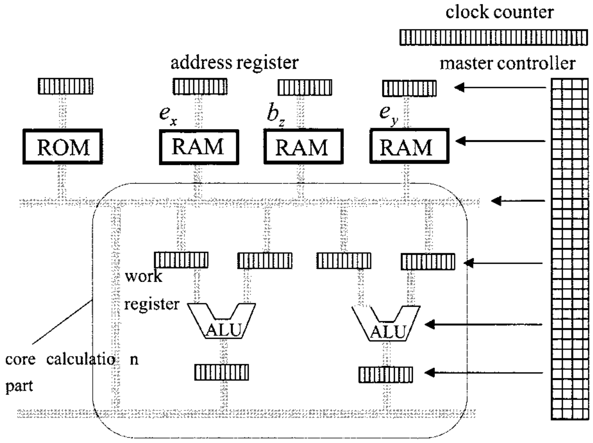


Fig. 2. Overview of hardware engine.

## II. Conceptual Design

Two-dimensional Maxwell's equations are divided into two independent modes of TE$_z$ and TM$_z$ if we assume that fields are uniform with respect to $z$-direction. For example, the TM$_z$ mode in vacuum consists of the following three equations:

$$\dot{E}_x = c^2 \frac{\partial B_z}{\partial y} \tag{1}$$

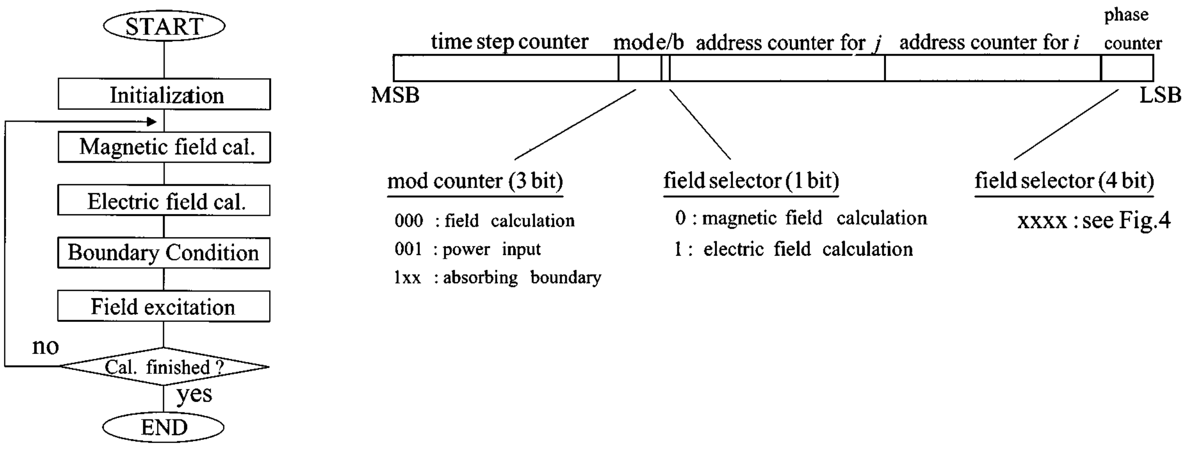
$$\dot{E}_y = -c^2 \frac{\partial B_z}{\partial x} \tag{2}$$

$$\dot{B}_z = -\left( \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right). \tag{3}$$

Manuscript received July 5, 2001; revised October 25, 2001.

The authors are with the Muroran Institute of Technology, Mizumoto-cho, Muroran 050-8585, Japan (e-mail: kawa@mmm.muroran-it.ac.jp; takahara@elec.muroran-it.ac.jp; s0824099@elec.muroran-it.ac.jp).

Publisher Item Identifier S 0018-9464(02)00962-7.



(a) (b)

Fig. 3. (a) FDTD scheme process flow. (b) System clock counter map.

Discretization of (1)–(3) according to the standard FDTD scheme yields the following finite difference equations (Fig. 1):

$$e_{xi,j}^{n+1} = e_{xi,j}^{n-1} + \tfrac{1}{2}\left(b_{zi,j}^{n} - b_{zi,j-1}^{n}\right) \tag{4}$$

$$e_{yi,j}^{n+1} = e_{yi,j}^{n-1} - \tfrac{1}{2}\left(b_{zi,j}^{n} - b_{zi-1,j}^{n}\right) \tag{5}$$

$$b_{zi,j}^{n+2} = b_{zi,j}^{n} - \tfrac{1}{2}\left(e_{yi+1,j}^{n+1} - e_{yi,j}^{n+1} - e_{xi,j+1}^{n+1} + e_{xi,j}^{n+1}\right) \tag{6}$$

where it is assumed that the grid size $\Delta l$ is uniform everywhere for both $x$ and $y$ directions. And the unknown values and stability condition are taken to be as follows:

$$e_x = \frac{E_x}{c}, \qquad e_y = \frac{E_y}{c}, \qquad b_z = B_z \tag{7}$$

$$\frac{c\Delta t}{\Delta l} = \frac{1}{2}. \tag{8}$$

Here, one can easily find that algebraic operations in (4)–(6) consist of just only addition, subtraction and 1-bit right shift (for factor 1/2). This simple calculation structure gives us a possibility of small size hardware (Fig. 1).

## III. Hardware Design

### A. Hardware Configuration

A hardware configuration of the digital logic circuit of the microwave simulator engine is shown in Fig. 2. The field values of $e_x$, $e_y$, $b_z$ are stored in individual SRAMs and the RAM address for the field values $e_{x\,i,j}$, $e_{y\,i,j}$, $b_{z\,i,j}$ at a grid $(i, j)$ are supplied by the address registers. Operations of (4)–(6) are performed at the core calculation part of the digital logic circuit which consists of the six work registers, two ALUs, and data flow control switches. Data flow of operations (4)–(6) in this circuit is managed by the master controller. All operations are executed to synchronize with the system clock counter. Information of boundary conditions and geometry of simulation objects are stored in the ROM, as well as the field excitation signal.

### B. System Clock Map

A standard FDTD simulation flow is shown in Fig. 3(a). After initialization process, electric and magnetic field calculations, boundary condition setting, and field excitation are repeated during specified time steps. This process flow is managed by a system clock counter in this hardware engine. Fig. 3(b) shows the system clock map. The system clock counter consists of the following six parts:

- time step counter (16 bit);
- mode counter (3 bit);
- field selector (1 bit);
- address counter for $i$ (16 bit);
- address counter for $j$ (16 bit);
- phase counter (4 bit).

These counters are connected as one 56-bit counter and counted up synchronizing with the system clock during the hardware operation. The address counters provide address to the address registers. And the mode counter, field selector, and phase counter are referred by the master controller to make control signals for the core calculation part.

### C. Data Flow Control in Core Calculation Part

The data flow of operation (6) consists of the following nine phases (see Fig. 4).

*Phase 1:* loading of $e_{xi,j}$ and $e_{yi,j}$ data from SRAM to work registers [see Fig. 4(a)].

*Phase 2:* loading of $e_{xi+1,j}$ and $e_{yi,j+1}$ [see Fig. 4(a)].

*Phase 3:* parallel calculation of $-e_{xi+1,j} + e_{xi,j}$ and $e_{yi,j+1} - e_{yi,j}$ [see Fig. 4(a)].

*Phase 4:* loading of $-e_{xi+1,j} + e_{xi,j}$ and $e_{yi,j+1} - e_{yi,j}$ to work registers [see Fig. 4(b)]

*Phase 5:* calculation of $e_{yi,j+1} - e_{yi,j} - e_{xi+1,j} + e_{xi,j}$ [see Fig. 4(b)].

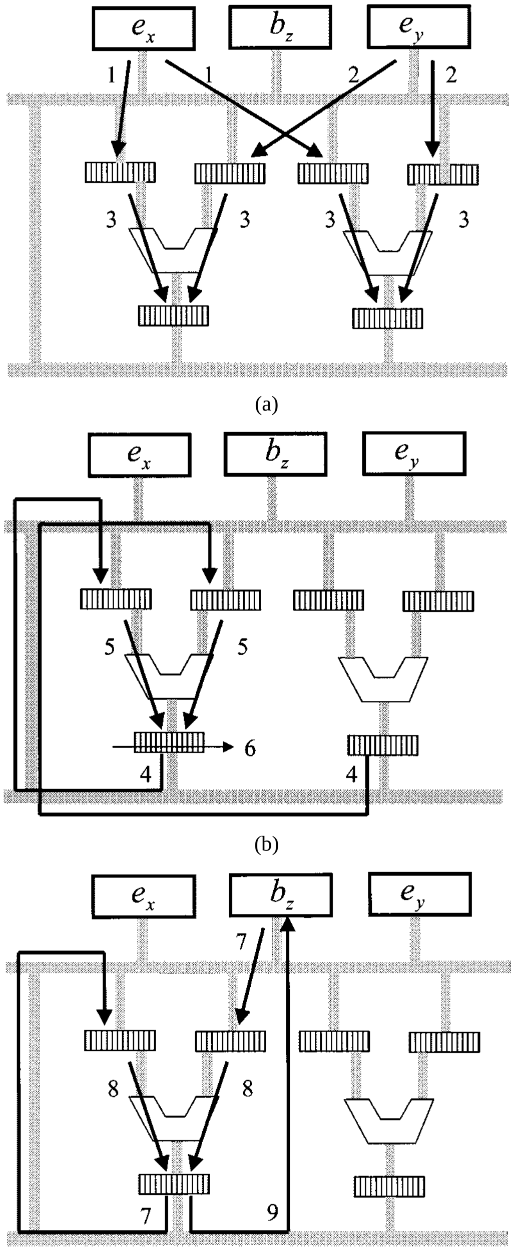
*Phase 6:* 1-bit right shift for the data $e_{yi,j+1} - e_{yi,j} - e_{xi+1,j} + e_{xi,j}$ [see Fig. 4(b)].

*Phase 7:* loading of $b_{zi,j}$ data from SRAM to work register and loading of the second term of (6) to work register [see Fig. 4(c)].

*Phase 8:* calculation of the right hand side of (6) [see Fig. 4(c)].

*Phase 9:* storing of the updated value of $b_{zi,j}$ to SRAM [see Fig. 4(c)].

*Phase A–F:* NOP.



(a)

(b)

(c)

Fig. 4. (a) Data flow in magnetic field calculation I. (b) Data flow in magnetic field calculation II. (c) Data flow in magnetic field calculation III.

Similarly, the data flow of operations of (4) and (5) is in the following eight clock steps.

*Phase 1:* loading of $b_{zi,j}$ data from SRAM to work registers.

*Phase 2:* loading of $b_{zi-1,j}$.

*Phase 3:* loading of $b_{zi,j-1}$.

*Phase 4:* parallel calculation of $b_{zi,j} - b_{zi-1,j}$ and $b_{zi,j} - b_{zi,j-1}$.

*Phase 5:* 1-bit right shift for the data $b_{zi,j} - b_{zi-1,j}$ and $b_{zi,j} - b_{zi,j-1}$.

*Phase 6:* loading of $e_{xi,j}$ and $e_{yi,j}$ from SRAM to work register and loading of $b_{zi,j} - b_{zi-1,j}$ and $b_{zi,j} - b_{zi,j-1}$ to work register.

*Phase 7:* parallel calculation of right hand side of (4) and (5).

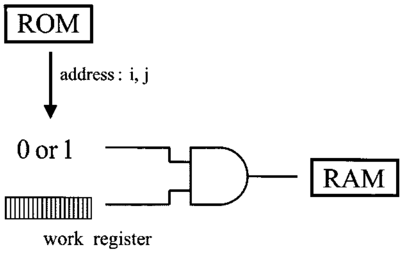


Fig. 5. Perfect boundary condition setting logic.

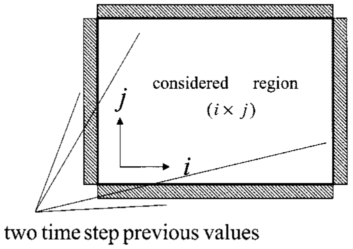


Fig. 6. RAM memory map for ABC data storing.

*Phase 8:* storing of updated value of $e_{xi,j}$ and $e_{yi,j}$ to SRAM.

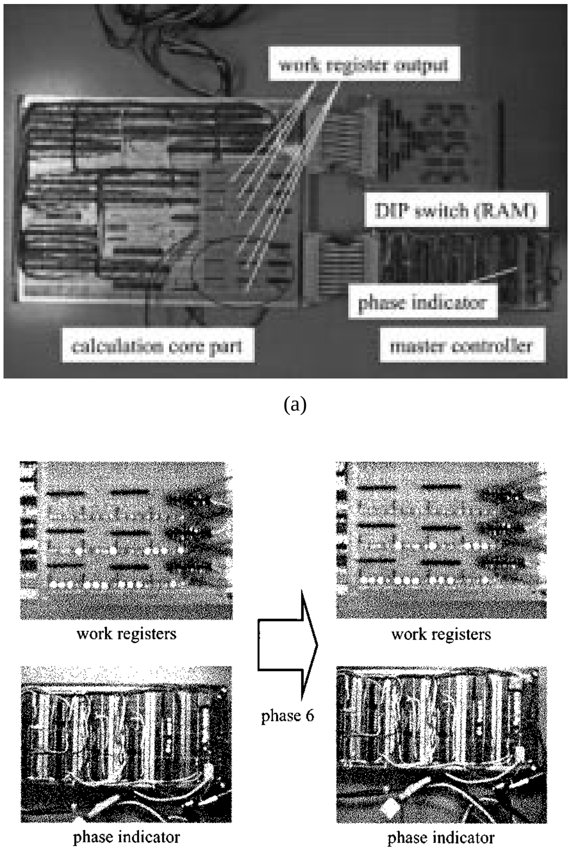
*Phase A–F:* NOP.

This data flow control is done by changing connection between RAM, work registers, and ALUs, and by switching the load/hold signal of the work registers. To carefully look the above process, it is found that the total process can be compressed into two or three steps less. For example, steps 1 and 2 can be done with the steps 8 and 9 simultaneously in Fig. 4. It is noticeable that all calculations of (4)–(6) can be done in one hardware. Accordingly 2-D FDTD calculation for each grid is performed by about 16 clock cycles. Roughly speaking, estimated performance of this simulator engine is about 6.25 $\times$ $10^6$ grid calculation/s if the system clock is 100 MHz. This performance is corresponding to 70 MFLOPS.

### *D. Boundary Value Setting*

*1) Perfect Conductor Boundary:* The perfect conductor boundary condition is readily set in the FDTD scheme. Owing to time domain calculation scheme, the boundary condition setting is achieved to set zero value on the all field value on the grid located in the perfect conductors. In the ROM, 1-bit data 0/1 is stored at the address $(i, j)$ which is corresponding to the grid position $(i, j)$. This 1-bit data is referred just before the final calculation value is stored into the RAM, and use as operand of AND operation with the register value. If the 1-bit data is 1, the register value is stored in RAM without any changing. On the other hand, the 1-bit data at address $(i, j)$ is zero, the register value is set to zero. Accordingly, the perfect conductor boundary condition setting can be done with almost no time delay (see Fig. 5).

*2) Absorbing Boundary:* One more important boundary condition in performing the FDTD simulation is the absorbing boundary condition (ABC). The difficulty of setting the ABC in hardware logic is in storing the previous field values in the



(a)

(b)

Fig. 7. (a) Overview of hardware simulation engine. (b) Register status after phase 6.

vicinity of the boundary. But this is possible if we adopt the Mur ABC

$$
\begin{aligned}
W|_{0,j}^{n+1} = &- W|_{0,j}^{n-1} + \tfrac{1}{3}\left(W|_{1,j}^{n+1} + W|_{0,j}^{n-1}\right) \\
&+ \tfrac{4}{3}\left(W|_{0,j}^{n} + W|_{1,j}^{n}\right) \\
&+ \tfrac{1}{12}\begin{bmatrix} W|_{0,j+1}^{n} - 2\,W|_{0,j}^{n} + W|_{0,j-1}^{n} + \\ W|_{1,j+1}^{n} - 2\,W|_{1,j}^{n} + W|_{1,j-1}^{n} \end{bmatrix}. \qquad (9)
\end{aligned}
$$

The Mur ABC requires to store only two time-step previous values on the boundary. This data storing is easily utilized to design the RAM memory map as shown in Fig. 6. The previous values are allocated outside of the considered region and placed parallel to boundaries. The data flow control of (9) in the hardware is achieved similarly to that of (4)–(6). The detailed description of it is omitted here.

### E. Field Excitation

Field excitation in FDTD is just addition of excitation signal value to the value on the excitation point. This is also possible when using the ROM. The excitation signal is then stored in the ROM and added to values at excitation grid point after the field calculation and boundary condition setting. Utilization of the excitation logic is very tedious but not so difficult. The detail of the logic is also omitted here.

## IV. Prototype Hardware

Photographs of the prototype hardware are shown in Fig. 7(a). The core calculation part of the engine is mainly implemented in the prototype. To be flexible for any modification, the master controller and RAMs are installed in the separated board. In the prototype, the field values are provided by the DIP switch instead of RAMs for simplicity. For example, Fig. 7(b) shows the status just after the phase 6 in Fig. 4. The value in work register is shifted by 1 bit. The prototype hardware is still being made and is in test phase. Basic operations of data load, addition/subtraction are confirmed, and complete operation will be confirmed in the near future.

## V. Summary

Conceptual design and the prototype hardware digital circuit for the microwave simulator engine have been presented in this paper. Implement of the boundary value setting and field excitation have not been completed and will appear in the near future. This engine has the following big advantages:

— easy extension of RAM capacity;

— easy enhancement of performance by parallel scheme.

These further improvements are also future works.

## References

[1] T. Kashiwa and I. Fukai, "A treatment by the FD-TD method of the dispersive characteristics associated with electronic polarization," *Microw. Opt. Technol. Lett.*, vol. 3, no. 6, pp. 203–205, 1990.

[2] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *Int. J. Numer. Model.; Electron. Network, Devices, Fields*, vol. 9, pp. 359–374, 1996.

[3] J. R. Marek, M. A. Mehalic, and A. J. Terzuoli, "A dedicated VLSI architecture for finite-difference time domain calculations," in *Conf. Proc. 8th Annu. Rev. Progress in Applied Computational. Electromagnetics*, 1992, pp. 546–553.