

3次元マイクロ波シミュレータエンジンの基本設計に関する研究

正員 川口 秀樹* 非会員 廣瀬 公輝* 非会員 堀田 誠人*
非会員 松岡 俊佑* 正員 高原 健爾*

Design Study of Three-Dimensional Microwave Simulator Engine

Hideki Kawaguchi*, Member Koki Hirose*, Non-member Makoto Horita*, Non-member
Shunsuke Matsuoka*, Non-member Kenji Takahara*, Member

According to big demands for high performance computing of simulation of microwave phenomena in industries, development of specialized purpose digital circuit hardware is proposed instead of use of very large scale supercomputer. That is, by specializing simulation target, very high performance simulator is realized with much smaller hardware size than supercomputers. Authors already presented a design study of simulator engine for two dimensional microwave. However, three-dimensional em fields are much practical in microwave industry. From this viewpoint, design study of simulator engine for three dimensional microwave is discussed in this paper.

キーワード: FDTD, FIT, HPC, マイクロ波, 専用マシン

Keywords: FDTD, FIT, HPC, microwave, specialized purpose computer machine

1. まえがき

コンピュータ技術におけるの著しい性能向上、および FDTD 法⁽¹⁾、FIT 法⁽²⁾などの有力な計算スキームの確立により、実際の産業の場においてもマイクロ波の数値シミュレーションが実用的となりつつある。しかしながらその一方で、個々の電子部品や素子レベルから、さらに大きな系としてのパソコンなど電子製品全体をシミュレーションするとなると未だスーパーコンピュータなどの高性能マシンにおいてさえ容易には行えないのも事実であり、要求は強くてもこのような超大規模システムを取扱うには、年々コンピュータの性能向上があるとしても、まだまだ相当な年数を要すると予想される。計算機システムにおいて性能向上を行なう上でのもっとも大きな問題とされているのが、図1に示すようなノイマン型計算機アーキテクチャに起因するメモリとCPU間のデータ転送に関連したオーバーヘッド(ノイマンボトルネック)である。すなわち、この形態が採用される限りは、ベクトル方式や並列方式のCPUを採用しようとも、さらには将来的に量子コンピュータ、光コンピュータなど次世代CPUが登場しようとも、バスにおけるデータ転送の輻輳はかならず発生し、これが計算機全体と

しての性能向上を著しく妨げてしまう。もちろん、最近ではそのまま図1のアーキテクチャを採用しているケースは皆無であり、キャッシュなどによりバスでのオーバーヘッドを回避する方式がさまざまに講じられているが、超高性能計算の実現においては、基本的には上記CPU間の通信がボトルネックとなっていることには変わらない。その一方で、これに相対する典型的な方式が、専用ハードウェアである^{(3),(4)}。すなわち、図1のバス結合アーキテクチャを捨て去り、シミュレーションする問題の計算スキーム専用メモリとCPU間の結合を構成することにより、メモリとCPU間のデータ転送での複雑なバスプロトコルによるオーバーヘッドをなくし高速な計算を実現する方式である。このような方式は、図1のアーキテクチャがもっているソフト

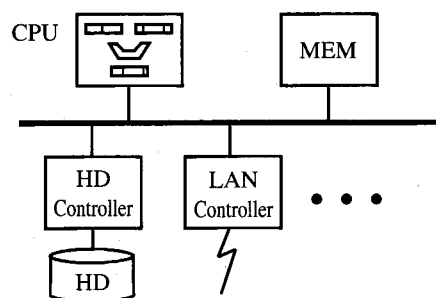


図1 バスアーキテクチャ型計算機構成
Fig.1. Standard computer bus architecture.

*室蘭工業大学 電気電子工学科
〒050-8585 室蘭市水元町 27-1
Dept of Electrical and Electronic Eng., Muroran Inst. of Tech.
27-1, Mizumoto-cho, Muroran 050-8585

ウェアの変更のみでどんな問題も取り扱える柔軟性という大きな利点があられる一方で、専用ハードウェアが取扱う問題に関しては、極めて高い性能を発揮し、さらに(複雑なOS管理がない)単純にメモリを増設するだけでほぼ無制限に取扱う問題を大規模化できるというもう一つの大きな利点を有することになる。

このような背景から、著者らはこれまで、マイクロ波の現象にターゲットをしばり、マイクロ波シミュレータエンジンの検討を行ってきた⁽⁴⁾。とりわけ、ここまでまずはじめの段階として2次元マイクロ波を対象としてFDTD/FIT法をハードウェア回路化すべく、その概念設計およびそのプロトタイプを試作してきた。しかしながら、実際にはマイクロ波では2次元系は極めて稀であり、実用的には3次元化が必須であることを鑑み、本稿では3次元マイクロ波シミュレータエンジンの概念設計を行なったので報告する。

2. 定式化

FDTD/FIT解析での大部分は真空であることを考え、ここではマクスウェル方程式としては、電磁場 **E**, 磁束密度 **B** を変数にとった、下記の形式をベースに考える。

$$\text{rot } \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \dots\dots\dots(1)$$

$$\text{rot } \mathbf{B} = \mu \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \dots\dots\dots(2)$$

また、これらの方程式の通常のFDTD/FIT定式化においては、新しい未知変数として、

$$\left. \begin{aligned} \mathbf{e} &= \frac{\mathbf{E}}{c} \\ \mathbf{b} &= \mathbf{B} \end{aligned} \right\} \dots\dots\dots(3)$$

を導入し、さらに安定条件をみたま時間差分間隔 Δt として、

$$\frac{c\Delta t}{\Delta l} = \frac{1}{2} \dots\dots\dots(4)$$

とここではとることとすると、(1)、(2)式は、それぞれ、

$$\begin{aligned} & b_{x_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}} - b_{x_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}}} \\ &= \frac{-1}{2} \left(e_{z_{i,j+1,k+\frac{1}{2}}^n} - e_{z_{i,j,k+\frac{1}{2}}^n} - e_{y_{i,j+\frac{1}{2},k+1}^n} + e_{y_{i,j+\frac{1}{2},k}^n} \right) \\ & b_{y_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}} - b_{y_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n-\frac{1}{2}}} \\ &= \frac{-1}{2} \left(e_{x_{i+\frac{1}{2},j,k+1}^n} - e_{x_{i+\frac{1}{2},j,k}^n} - e_{z_{i+1,j,k+\frac{1}{2}}^n} + e_{z_{i,j,k+\frac{1}{2}}^n} \right) \\ & b_{z_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}} - b_{z_{i+\frac{1}{2},j+\frac{1}{2},k}^{n-\frac{1}{2}}} \\ &= \frac{-1}{2} \left(e_{y_{i+1,j+\frac{1}{2},k}^n} - e_{y_{i,j+\frac{1}{2},k}^n} - e_{x_{i+\frac{1}{2},j+1,k}^n} + e_{x_{i+\frac{1}{2},j,k}^n} \right) \end{aligned} \dots\dots\dots(5)$$

$$\begin{aligned} & e_{x_{i+\frac{1}{2},j,k}^{n+1}} - e_{x_{i+\frac{1}{2},j,k}^n} \\ &= \frac{1}{2} \left(b_{z_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}} - b_{z_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}} - b_{y_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}} + b_{y_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}} \right) \\ & e_{y_{i,j+\frac{1}{2},k}^{n+1}} - e_{y_{i,j+\frac{1}{2},k}^n} \\ &= \frac{1}{2} \left(b_{x_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}} - b_{x_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}}} - b_{z_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}} + b_{z_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}} \right) \\ & e_{z_{i,j,k+\frac{1}{2}}^{n+1}} - e_{z_{i,j,k+\frac{1}{2}}^n} \\ &= \frac{1}{2} \left(b_{y_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}} - b_{y_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}} - b_{x_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}} + b_{x_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}} \right) \end{aligned} \dots\dots\dots(6)$$

となる。すなわち、未知数 **E**, 磁束密度 **B** に係る係数は、-1、0、1のみとなり、かつ1/2のファクターはデジタル表示では1ビット右シフトで実現されることに注意すると、(5)、(6)式にはもはや加減算のみの極めて単純な動

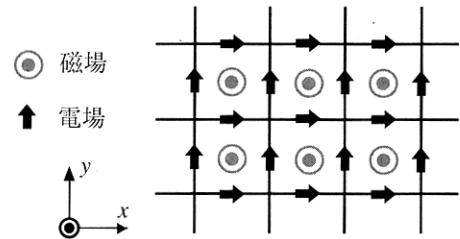


図2 2次元FDTD/FITグリッドと場の成分の配置
Fig.2. 2D FDTD/FIT grids and allocation of field components.

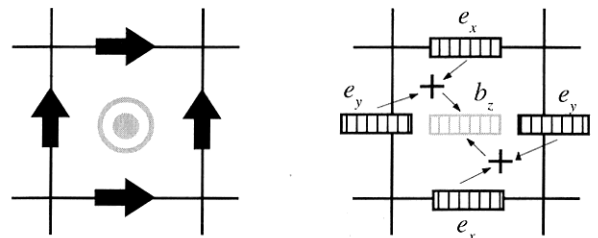


図3 2次元FDTD/FITグリッドと場の成分の配置
Fig.3. 2D FDTD/FIT grids and allocation of field components.

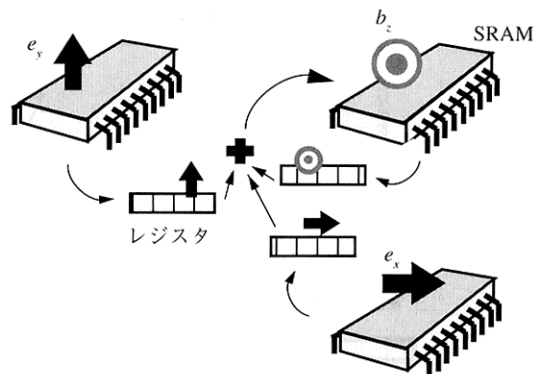


図4 エンジンの基本構成概念
Fig.4. Basic concept of simulator engine.

作で1ステップ分の場の計算が行え、ハードウェア化に最適な形式になることがわかる。

3. ハードウェア構成

前節で定式化したスキームをハードウェア化するには、さまざまな方法が考えられる。例えば、ひとつの方式としては、各格子状においてそれぞれ独立に未知数をレジスタに格納し、レジスタ間の結合を(5)、(6)式にしたがってデータが流れるよう構成する方法がある。(図2、3)この場合、ほぼ1クロックで一度に全ての格子状の場の値を計算することができるため超高性能の理想的なエンジンが実現できる。しかしながら、それは容易に想像できるようにハードウェア規模は膨大となり、さらに、一度ハードウェアをつくと取扱う問題のサイズに応じてハードウェアサイズや数値モデル形状を柔軟に変更するのが極めて難しくなるという困難がある。これに対し、例えば図2に示すようなFDTD/FITの2次元グリッド構造に対し、図4に示すように、それぞれ場の成分ごとに1つのRAMに格納し、アドレスを指定することにより所定の位置の場の値を読み出し、逐次計算しては再度メモリに格納する方式とすれば、前者の方式に比べて幾分性能の低下はあるものの上述のバストポロジーに起因するオーバーヘッドのない高速演算は実現でき、かつ実用的に問題に柔軟に対応できることになる。(但し、ここでは3次元問題を取扱っているので、例えばRAMは6個必要)またこのような方式を採用することにより、メモリは汎用のRAMを使うことができ、さらに同エンジンが通常のPCのようなOSによって動作していないこ

とを考えると、メモリ増設にあたっては単にアドレス空間を増やすだけで他のどんな制約なしにほぼ無制限にメモリを登載、すなわち計算規模を拡張できることになる。したがって、機能的に分類するなら前者図3の方式は超高速計算エンジン、後者図4の方式は超大規模計算エンジンといえる。本研究でも文献[4]同様、シミュレータエンジン開発の初期ステップとしてまず、柔軟性のある後者の超大規模計算エンジンの方式を採用することとする。

図4のスキームを具体的に実現するハードウェア回路を図5に示す。回路は、大別して、場の未知変数を格納するメモリモジュール、FDTD/FITのスキームを実現する演算モジュール、およびこれらの回路のデータフローを制御する信号を発生させるマスターコントローラからなる。メモリモジュールには、 $e_x, e_y, e_z, b_x, b_y, b_z$ がそれぞれ個別に格納されるためのRAM(×6)、およびメモリへの読み書き動作において格子空間における位置を指定するためのアドレスを生成しRAMに指定するロジックから構成されている。演算モジュールは、計算途中の値を一時的に格納するシフトレジスタ(×9)、加減算器(×3)、およびメモリーレジスタ間あるいはレジスタ-レジスタ間のデータフローを制御するスイッチ(ラッチ×12)から構成されている。全体の動作はシステムクロックに全て同期して進行し、動作のフェーズはクロックカウンタによって管理される。まず、マスターコントローラはクロックカウンタを参照し、各計算フェーズ(次節参照)に対応してメモリ/レジスタレジスタ間の結合を切り替える信号(図5の2重線矢印)、およびレジスタへのホールド信号(図5の1本線矢印)を演算

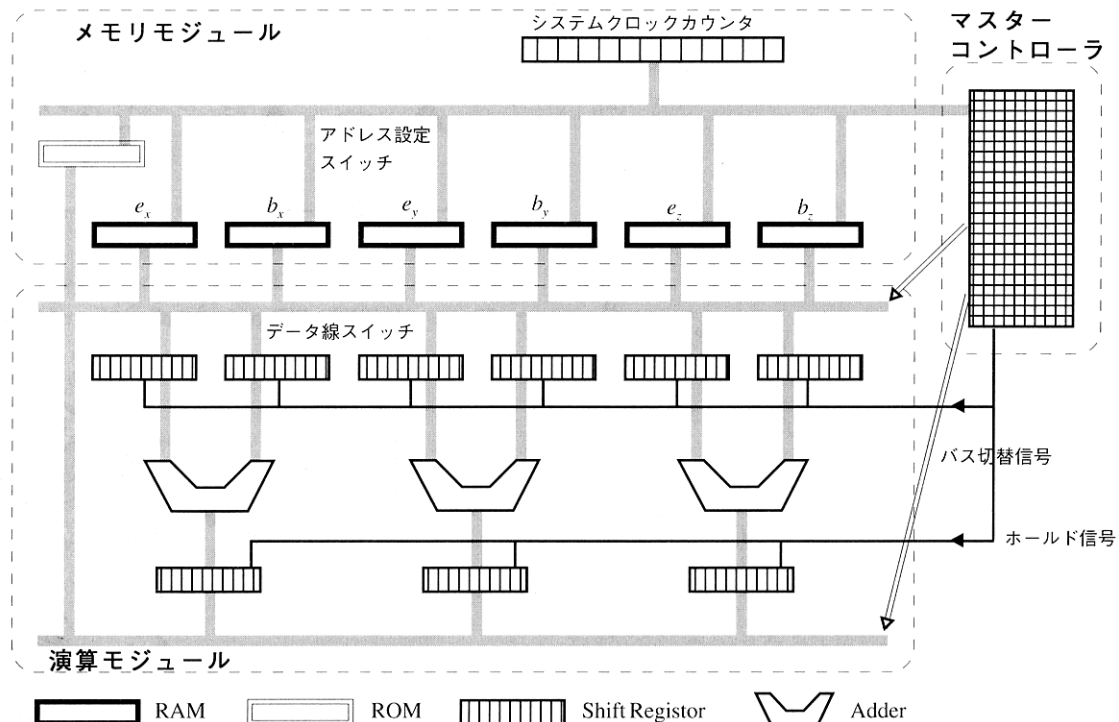


図5 FDTD/FIT エンジンハードウェア構成概要
Fig.5. Hardware structure FDTD/FIT simulator engine.

モジュールに出す。またそれと同時にアドレス設定スイッチを通じて各格子位置および計算フェーズに対応して使用される場の値が格納されているアドレスがRAMに指示される。この動作によりRAM内の場の値が時間ステップの進行にあわせて逐次更新され、FDTD/FITの計算が実施される。また、上記の基本的な計算動作に加え、実際のFDTD/FITの動作を実現するために、メモリモジュールに用意されたROMには、扱う問題の境界条件、または、系に加えるパワー入力信号など個々の問題に依存した情報が一括して格納される。

4. FDTD/FITスキームのデータフロー

エンジンの中心である演算モジュールで、(5)、(6)式のFDTD/FITスキームのデータフロー動作がどのように実現されるかを以下に具体的に示す。動作は大別して、データ線スイッチの切り替えとワークレジスタへのデータのロードの2つのアクションからなり、前者をクロックの立ち上がりのタイミングで行い、後者を立ち下がりで行うこととすると、1クロック毎にデータがメモリ/レジスタからレジスタに送られる動きとなる。したがって、1クロックを1つの動作フェーズとみなすと、例えば、格子点 $(i, j+1/2, k+1/2)$ での磁場 b_x の計算((5)式の最初の式)は次の10フェーズで行われる(図6参照)。

フェーズ1

$(i, j+1/2, k)$ の e_y , $(i, j+1, k+1/2)$ の e_z をレジスタへロード

フェーズ2

$((i, j+1/2, k)$ の e_y) + $((i, j+1, k+1/2)$ の e_z) を計算

フェーズ3

$(i, j+1/2, k+1)$ の e_y , $(i, j, k+1/2)$ の e_z をレジスタへロード

フェーズ4

(フェーズ2の結果) をレジスタへ再ロード、及び $((i, j+1/2, k+1)$ の e_y) + $((i, j, k+1/2)$ の e_z) を計算

フェーズ5

(フェーズ4の結果) をレジスタへ再ロード

フェーズ6

(フェーズ2の結果) - (フェーズ4の結果) を計算

フェーズ7

(フェーズ6の結果) を1ビット右シフト

フェーズ8

(フェーズ7の結果) をレジスタへ再ロード、及び $(i, j+1/2, k+1/2)$ の b_x をレジスタへロード

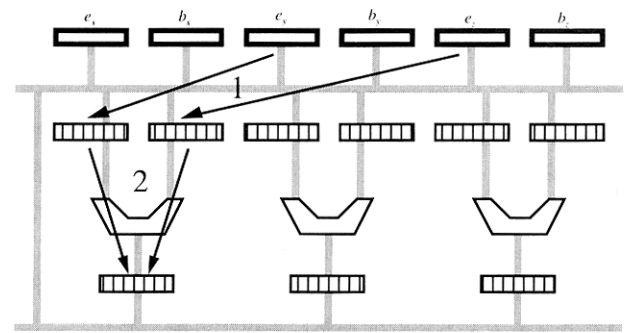
フェーズ9

$((i, j+1/2, k+1/2)$ の b_x) + (フェーズ8の結果) を計算

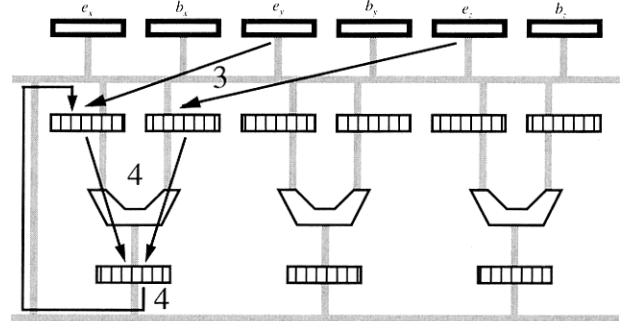
フェーズ10

新しい $(i, j+1/2, k+1/2)$ の磁場 b_x の値をメモリへ格納

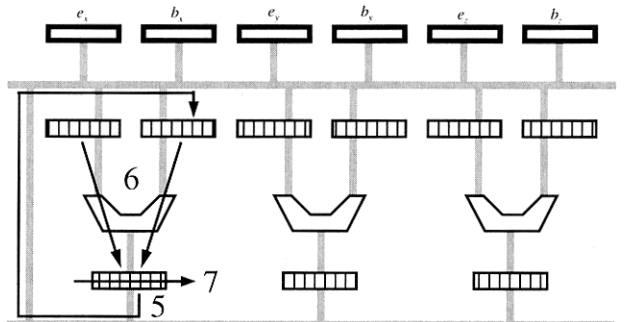
またまったく同様の要領で、磁場の他の成分 b_y , b_z の計算も行える。さらに、電場の計算も(6)式から容易に想像されるように、上記フェーズ9の部分を実算とすることでこの同じハードウェア上で行える。その際、電場と磁場の



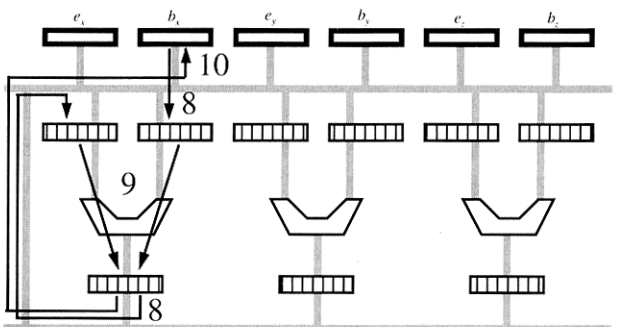
(a) フェーズ1および2



(b) フェーズ3および4



(c) フェーズ5、6および7



(d) フェーズ8、9および10

図6 磁場 b_x の計算

Fig.6. Calculation of magnetic field component b_x .

計算の違いは上述のようにフェーズ9の符号の違いのみであるので、マスターコントローラのロジックも大部分が電場と磁場で共通化できることもわかる。

5. 境界条件

上述のように FDTD/FIT のハードウェア化においてマクスウェル方程式の計算そのものは比較的スムーズに実現可能であることを見た。これに対し、もっとも煩雑な設計が予想されるのが境界条件の設定である。実用的に必要な境界条件としては、完全導体境界条件、誘電体境界条件、吸収境界条件などがある。このうち、完全導体境界条件はハードウェアでも比較的容易に実現が可能であり、ここではその具体的な方法を示す。

完全導体内部では場は全てゼロ、また、導体表面では電場の接線成分・磁場の法線成分がゼロとなるという事情は、図7の薄く塗りつぶされた格子の部分が完全導体であるとすると、それは、完全導体の格子上にある場の成分全てゼロとなるということに言い換えられる。したがって例えば、ハードウェアにおける完全導体の境界条件は次のようにすればよい。まず図8のように、取扱いたい形状を格子状に離散化し、真空部分を1、完全導体を0と分類してROMなどに格納しておく。そして、図6のフェーズ10において計算された最終的な場の値をメモリに格納する直前に、対応する格子の位置の0/1の情報をROMから読み出し、その計算値とANDをとった上でメモリに格納することとすれば、真空の部分はそのまま、完全導体の部分はゼロに設定され、結果としてどんな遅延もなしに自動的に形状に対応した境界条件が設定されることができ。すなわち、取扱う問題に応じてROMを取り替えることにより、ハードウェア本体の変更なしに、さまざまな問題を取扱うことが可能となる。

もう一点、これに関連して誘電体など真空以外の媒質を取扱う場合の対応方法について述べておく。媒質があることによる違いは、2節の定式化でいうと光速度 c が比誘電率あるいは比透磁率の平方根の逆数分 ($1/\sqrt{\epsilon_r \mu_r}$) だけ値を変えることである。この場合はもはや、(4)のように右シフトでは対処できない。このため例えば、各格子における上記媒質分の乗じるべき係数を図8と同様に各格子ごとにROMの当該アドレスに格納しておき、フェーズ8でフェーズ7の結果を上段のレジスタに移動する際にROMか

ら読み出し、フェーズ7の結果に乗じた上で、上段のレジスタに格納するよう回路を追加するなどして実現できる。

しかしながらこの場合も、乗算の分の回路は追加されるものの、クロック数の増加など処理の速度は変わらない。

6. FDTD/FITフローとシステムクロックカウンタ

以上これまで、マクスウェル方程式と完全導体境界条件のハードウェア化の方法について説明した。しかしながら、実際の計算を行なうには境界条件以外にもパワー入力などまだ多くの機能を実現する必要がある。シミュレータエンジンにおいて、これら全体の処理の流れを管理・制御するのが図5の中のシステムクロックカウンタであり、その詳細を図9に示す。(また、参考のため、標準的なFDTD/FITの処理フローチャートを図10に示す。)システムクロックを単純にカウントする同カウンタにおいて、LSB側の数ビットのフェーズカウンタ(4章で説明した各格子点での計算のフェーズを表す)、それにつづく上位のアドレスカウンタ(計算する格子点 i, j, k の位置を表す)、電場・磁場フラグ、モードフラグ、および時間ステップカウンタの順にMSB側にカウントアップが進むにつれて図10のフローチャートと同様の処理のスケジュール管理が行なわれる。そしてマスターコントローラによるスイッチ切替信号、RAMやROMへのアドレスはこのカウンタ値を参照して発生される。

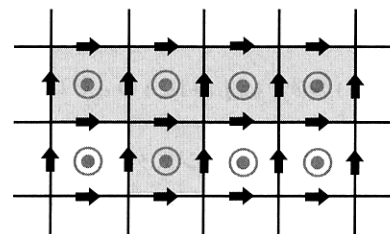


図7 完全導体の境界条件

Fig.7. Perfect conductor boundary condition.

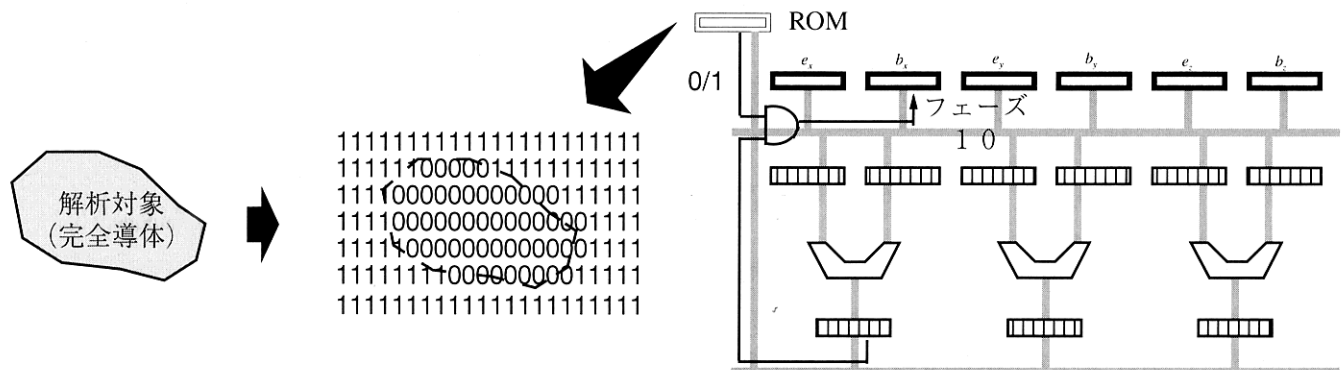


図8 ハードウェアエンジンにおける完全導体境界条件設定方法

Fig.8. Boundary condition setting for perfect conductor in hardware circuit.

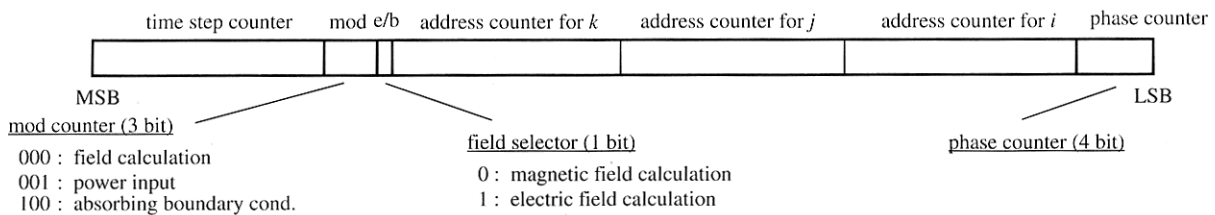


図9 ハードウェアエンジンクロックカウンタマップ

Fig.9. Map of system clock counter.



図10 FDTD/FITのフローチャート

Fig.10. Standard calculation flowchart of FDTD/FIT.

7. プロトタイプハードウェアの試作と今後の課題

ハードウェアは2次元用シミュレータエンジン⁽⁴⁾同様、図5の構成概略図同様、演算モジュール、メモリモジュール、マスターコントローラの3つにモジュール化することにより、ハードウェアの大部分を占める演算モジュールを共通化させて、メモリサイズあるいは部分的な処理の変更に対してはそれぞれメモリモジュールやマスターコントローラを取り替えるだけで対処できるような仕様とする。

このような方針に基づいてレイアウト設計を行ない、ソケットベースで作成した3次元シミュレータエンジンの演算モジュールを図11に示す。基盤のサイズはA3程度であり、約180個のTTLが搭載され、3次元化により2次元のものより約1.5倍程のサイズとなっている。

以上、3次元マイクロ波シミュレータエンジンの設計およびプロトタイプ作成について報告した。今後は、これらをベースに、まず、完全導体以外の境界条件、RAMへの場の値の格納方法、パワー入力の具体的ハード化方法の検討を行ない、さらに実用に近いハードウェアエンジンを開発していく予定である。また、この試作版ではハードウェアのデバクを目的としていたため全ての信号を観察できるようTTLベースで作製したが、今後実用に向けては、LSI化、特にFPGAにより作製することを検討していく予定である。

最後に、ハードウェアエンジンにおける予測計算性能に

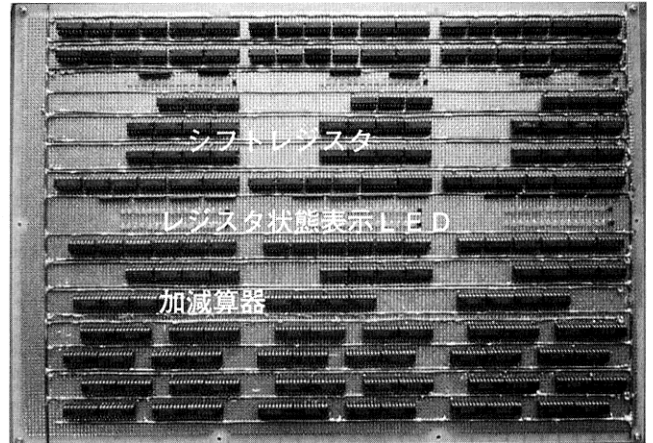


図11 プロトタイプの演算モジュール部のレイアウト

Fig.11. Layout of calculation module.

について議論しておく。本稿にて提案した方式の専用シミュレータエンジンは、このままの形態ではもちろん実用的な性能は望めない。すなわち、予測性能は次のようになる。(6)の各成分の計算はアセンブラレベルではそれぞれ5ステップであり、一方図6で説明した処理フローでは、電場、磁場とも x, y, z 成分を同時に並列処理できるため10クロックで15ステップ、すなわち、1クロックで1.5ステップ分のアセンブラレベルの処理ができることになる。たとえば仮に、手作業配線に起因するノイズの発生がなく100 MHzのクロックで同回路の動作が行えたとしても、それでも高々150 MFLOPS 相当の計算機性能しかなく、既存のスーパーコンピュータの数百ギガ~数テラ FLOPS には遠く及ばない。これは大きくは、3章の冒頭で議論した通り、ここではわれわれは高速演算タイプではなく、大容量メモリタイプの計算機を選択したこと起因している。しかしながら、実用機としてFPGAなどでLSI化するにあたっては、図11の回路は数千個程度一つのLSIに格納可能であり、本ハードウェアがノイマンボトルネックなどの問題がない方式であることを考えると、スケーラブルな並列処理ができ、したがって、上記予測性能の数千倍すなわちスーパーコンピュータに匹敵する計算性能の、かつ、メモリ容量に制限のないシミュレータエンジンを安価に構築することは十分可能であると考えられる。一方、すでにもう一方の方式である高速演算マシンの検討も行っており、近く報告の予定である。

(平成14年2月14日受付、平成14年6月10日再受付)

文 献

- (1) A. Taflov and S.C. Hagness : Computational Electromagnetics: the finite-difference time-domain method (2nd Ed.), Artech House (2000)
- (2) T. Weiland: "Time Domain Electromagnetic Field Computation with Finite Difference Methods," *Int. J. Numerical Modelling : Electronic Networks, Devices & Fields*, **9** pp.295-319 (1996)
- (3) J.R. Mark, M.A. Mehalic and A.J. Terzuoli : "A Dedicated VLSI Architecture for Finite-Difference Time Domain Calculations," Conf. Proc. of 8th Annual Review of Progress in Applied Comput. Electromagnetics, pp.546-553 (1992)
- (4) H. Kawaguchi, K. Takahara, and D. Yamauchi : "Design Study of Ultra-high Speed Microwave Simulator Engine," *IEEE Trans. Magn.*, **38**, No.2, pp.689-692 (2002)

川口 秀樹 (正員) 1962年12月3日生まれ。1988年3月北海道大学大学院工学研究科修士課程電気工学専攻修了。同年日本電信電話(株)NTT情報通信処理研究所入社。1991年4月北海道大学工学部電気工学科助手。1999年4月室蘭工業大学電気電子工学科助教授。この間、1995年12月～1996年9月文部省在外研究員(ゲルムシュタット工科大学、ドイツ)。博士(工学)。



広瀬 公輝 (非会員) 1979年4月26日生まれ。2002年3月室蘭工業大学電気電子工学科卒業。同年室蘭工業大学大学院工学研究科博士前期課程電気電子工学専攻入学。



堀田 誠人 (非会員) 1978年8月1日生まれ。2001年3月室蘭工業大学電気電子工学科卒業。同年室蘭工業大学大学院工学研究科博士前期課程電気電子工学専攻入学。



松岡 俊佑 (非会員) 1978年6月25日生まれ。2001年3月室蘭工業大学電気電子工学科卒業。同年室蘭工業大学大学院工学研究科博士前期課程電気電子工学専攻入学。



高原 健爾 (正員) 1966年1月11日生まれ。1997年3月東京医科歯科大学大学院医学研究科博士課程単位取得退学。同年4月より室蘭工業大学電気電子工学科助手。博士(工学)。

