

# Cell プロセッサにおける時空間パイプラインによる FDTD 法の高速度化

遠藤 翔<sup>1</sup> 園田 潤<sup>2</sup>

仙台電波工業高等専門学校 専攻科電子システム工学専攻<sup>1</sup>

仙台電波工業高等専門学校 電子工学科<sup>2</sup>

FDTD 法は計算コストが膨大になるため、Cell プロセッサによる高速化が行われているが、領域分割を用いた並列計算ではメインメモリの帯域がボトルネックになる。一方、我々が提案した時空間パイプライン (TSP) は、メモリアクセスを削減し高速化できる手法であり、FDTD 法の FPGA 実装において並列計算と比べ 22 倍の高速化効果を示している。本報告では、Cell プロセッサ上に時空間パイプラインを用いた FDTD 法を実装し、高速化の検討を行っている。この結果、SPE7 基において、TSP を用いた FDTD 法は、大規模解析では並列計算の約 2 倍、小規模解析では並列計算の約 1.3 倍の効率で計算できることを示した。

## 1 まえがき

現在、電磁界解析の手法として FDTD (Finite - Difference Time-Domain) 法 [1] が広く用いられているが、位相誤差が蓄積するという問題がある。このことから、大規模な計算ではセルサイズを小さくすると同時にセル数を増やす必要があり、計算コストが増大する問題がある。一方、近年開発された Cell プロセッサは、マルチメディア処理 [2][3] および科学技術計算 [4] に多く用いられ、従来の CPU に比較して大幅な高速化を達成しており、FDTD 法にも適用できると考えられる。

そこで、Cell プロセッサを用いた FDTD 法の並列計算による高速化が行われている [5]–[7]。文献 [5][6] では、Cell プロセッサ (3.2 GHz) を搭載した IBM Blade QS20 上に二次元 FDTD 法を実装し、AMD Athlon プロセッサの 14.14 倍の計算速度を示している。文献 [5][6] では、メインメモリ上に解析領域を確保し、ローカルストアに逐次データを読み込み計算している。一方、文献 [7] では SPE のローカルストアを分散メモリとして用いて領域分割を行い、SPE 数にほぼ比例した速度向上を実現している。しかしながら、Cell プロセッサ上では計算時間に対してメインメモリと SPE 間の DMA 転送時間が占める割合が高く、メインメモリへの DMA 転送時間がボトルネックになることが示されている [5]。

一方、我々は時空間パイプライン (Time and Space Pipeline: TSP) という新しいハードウェアアルゴリズムを提案し、FDTD 法で FPGA (Xilinx XC3S1500) に実装した [8]。この結果、並列計算と比較して 22 倍、PC (Pentium 4 3 GHz) に比べ 9.2 倍高速に計算できることを示した。TSP は、共有メモリ計算機においてメモリア

クセスのボトルネックを解消し、高速化できるという特長がある。これを Cell プロセッサに適用することで、文献 [5] で示されている DMA 転送のボトルネックを解消でき、高速化が期待できる。

本報告では、Cell プロセッサ上に TSP を用いた FDTD 法を実装し、高速化の検討を行う。2. では TSP を用いた FDTD 法について述べる。3. では TSP を用いた FDTD 法の計算時間を従来の並列計算と比較し、TSP の有効性を示す。最後に 4. で本報告のまとめと今後の課題について述べる。

## 2 Cell プロセッサにおける時空間パイプラインを用いた FDTD 法

### 2.1 FDTD 法

FDTD 法は、マクスウェルの方程式を時間および空間で差分化して解く方法である。式 (1), (2) に一次元における電磁界の計算式を示す。ここで、 $\varepsilon$  は誘電率、 $\mu$  は透磁率、 $\Delta t$  は時間刻み、 $\Delta z$  はセルサイズである。 $F^n(k)$  は空間上の位置  $k$  ( $z = k\Delta z$ )、時間ステップ  $n$  ( $t = n\Delta t$ ) における値を示す。FDTD 法は解析領域のすべての点について、式 (1), (2) を交互に計算することで電磁界値を求める。

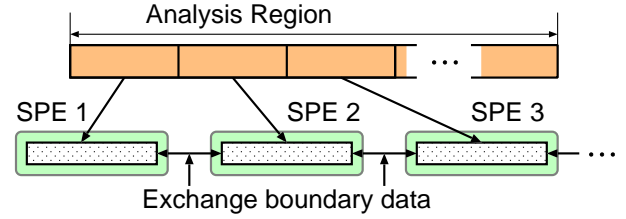
$$E_y^{n+1}(k) = E_y^n(k) + \frac{\Delta t}{\varepsilon \Delta z} \left\{ H_x^{n+\frac{1}{2}} \left( k + \frac{1}{2} \right) - H_x^{n+\frac{1}{2}} \left( k - \frac{1}{2} \right) \right\} \quad (1)$$

$$H_x^{n+\frac{1}{2}} \left( k + \frac{1}{2} \right) = H_x^{n-\frac{1}{2}} \left( k + \frac{1}{2} \right) + \frac{\Delta t}{\mu \Delta z} \left\{ E_y^n \left( k + \frac{1}{2} \right) - E_y^n(k) \right\} \quad (2)$$

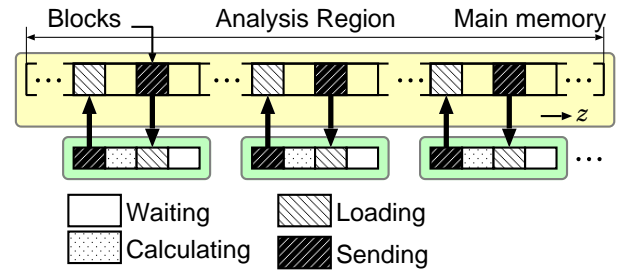
## 2.2 領域分割による FDTD 法の並列計算

図 1 に、既存研究で行われている領域分割による並列計算の方法を示す。領域分割による並列計算には、解析領域を SPE のローカルストア上に確保できる小規模解析と、確保できない大規模解析がある。図 1(a) に小規模解析における並列計算の計算方法を示す。小規模解析ではローカルストアを分散メモリとして使用できるため、解析領域を SPE の個数に分割し、各 SPE のローカルストアに割り当てる。各 SPE 間で割り当てられた領域の境界の値を交換しながら計算を行う。小規模解析では、メインメモリへのアクセスは不要である [7]。

図 1(b) に大規模解析における計算方法を示す。大規模解析ではローカルストアに解析領域全体を格納できないため、メインメモリに解析領域を確保し、計算を行う部分のみをローカルストアに読み込み計算する。各 SPE に割り当てられた領域をブロックと呼ばれる領域に分割し、DMA 転送によりメインメモリと SPE 間でブロックを順次入れ換えて計算を行う。すべての SPE のローカルストアがメインメモリと DMA 転送を行うため、SPE 数が増えると DMA 転送時間がボトルネックになり、速度向上が飽和する問題がある [5]。



(a) 小規模解析の並列計算



(b) 大規模解析の並列計算

図 1: 領域分割による並列計算

## 2.3 時空間パイプラインを用いた FDTD 法

大規模解析における DMA 転送のボトルネックを解決するため、本報告では複数の時間ステップにおける電磁界値の計算を同時に行う TSP[8] により高速化を行う。TSP はメモリアccessを増加させずに高速化できるため、メモリアccessのボトルネックを解消することができる。

図 2 に、TSP におけるデータの流れを示す。 $n_{\text{spe}}$  は SPE 数である。TSP では領域分割で大規模解析を行う場合同様、メインメモリに解析領域を格納する。並列計算との相異点は、各 SPE が並列に動作するのではなく、パイプライン状になっている点である。また、各 SPE には解析領域の一部を割り当てるのではなく、時間ステップを割り当てるため、各 SPE はそれぞれの時間ステップにおける電磁界値を全解析領域にわたり計算する。

図 2 において、1 段目の SPE は、時間ステップ  $n$  の電界および  $n + 1/2$  の磁界のデータが含まれるブロックをメインメモリから読み込み計算する。TSP では電界の計算を行った後、直ちに磁界の計算を行う。そのため、データが 1 個の SPE を通るとき時間ステップは 1 進む。あるブロックの計算が終了すると、SPE は次の段の SPE に計算完了信号を送信する。この通信にはシグナル通知レジ

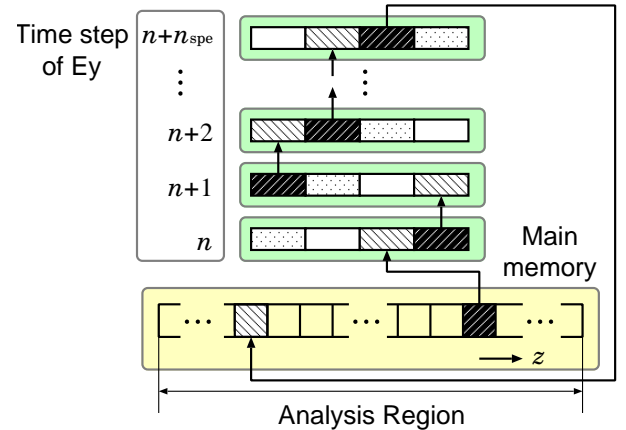


図 2: 時空間パイプラインにおけるデータの流れ

スタを用いる。計算完了信号を受信した SPE は、前の段の SPE から計算が完了したブロックを DMA 転送により取得する。最終段の SPE はメインメモリにブロックを書き込む。TSP においてメインメモリにアクセスする SPE は、最初と最後の段のみである。このため SPE 数を増加させてもメモリアccessは増加せず、ボトルネックを解消できる。

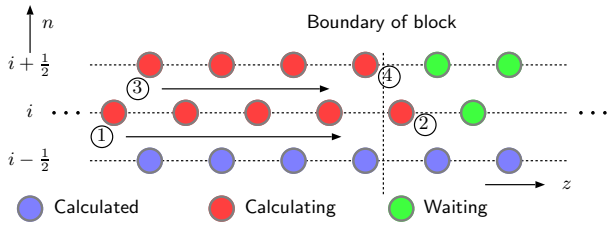


図 3: 各ブロックの計算方法

## 2.4 TSP における各ブロックの計算方法

TSP における各ブロックの計算方法を図 3 に示す．①-④にアルゴリズムを示す．

- ① 時間ステップ  $i$  における電界の計算を行う
- ② 次のブロックの境界の電界値を計算する
- ③ 時間ステップ  $i + 1/2$  における磁界の計算を行う
- ④ 現在のブロックの境界の磁界値を計算する

TSP では電界の計算の直後に磁界の計算を行うが，④の計算には次のブロックに属する値である②が必要になる．そこで，計算を行うブロックの次のブロックもあらかじめ読み込んでおき，この値を用いて②の値を計算する．

## 3 Cell プロセッサにおける時空間パイプラインの評価

### 3.1 解析条件

TSP を FDTD 法による一次元自由空間の伝搬問題に適用し，Cell プロセッサにおける TSP の評価を行う．解析領域は原点に電界  $E_y$  のハードソースを置き，観測点で電界  $E_y$  を観測する．解析条件は，セルサイズ  $\Delta z = \lambda/90$ ，時間刻み  $\Delta t = 1.010 \times 10^{-11}$  s とした．ただし， $\lambda$  は波源の波長である．大規模解析では，解析時間  $t = 204$  ns，波源と観測点の距離は  $200\lambda$  とした．解析領域のセル数は 20160，解析ステップ数は 20160 である．小規模解析では，解析領域を SPE のローカルストア内に確保できるよう，解析領域の長さを短くしている．解析条件は，解析時間  $t = 1020$  ns，波源と観測点の距離は  $88.9\lambda$  とした．解析領域のセル数は 10080，解析ステップ数は 108000 である．

評価に用いた計算機は Cell リファレンスセットであり，コンパイラは gcc 4.1.1(最適化 -O3) である．浮動小数点の変数には倍精度浮動小数点型を用いた．TSP，並列計算ともに，ダブルバッファリングによる DMA 転送時間

表 1: 各 SPE 数における DMA 転送サイズ [Bytes]

Number of SPEs	Parallelism	TSP
1	10080	13440
2	10080	10080
3	4480	13440
4	10080	13440
5	4032	8064
6	2280	6720
7	1152	5760

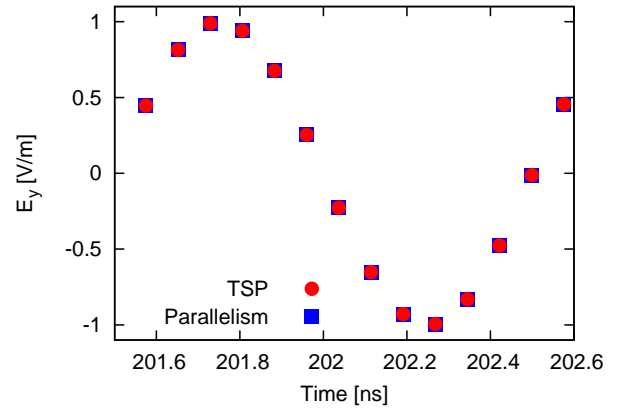


図 4: 観測点における  $E_y$  の値

の隠ぺい，SIMD 化およびループアンローリング [9][10]を行っている．

大規模解析の並列計算ではメインメモリと SPE 間で DMA 転送を行う．表 1 に各 SPE 数における DMA 転送サイズを示す．DMA 転送サイズは，DMA 転送命令 1 回で転送するバイト数である．この値は，解析領域を SPE 数 (並列計算の場合) およびブロック数で分割して整数のセル数になるようにしなければならない．表 1 は，TSP，並列計算ともにこの条件の中で最大の速度となる転送サイズである．

### 3.2 計算精度と速度向上の検討

図 4 に TSP の計算結果を示す．図 4 より，TSP と並列計算の結果は一致していることが分かる．TSP は計算順序を変えるだけなので，通常の FDTD 法および並列 FDTD 法の計算結果と一致する．

図 5 に TSP と並列計算の速度向上比を示す．まず，大規模解析について評価を行う．図 5 より，TSP においては SPE 数に比例して計算速度が増加していることが分か

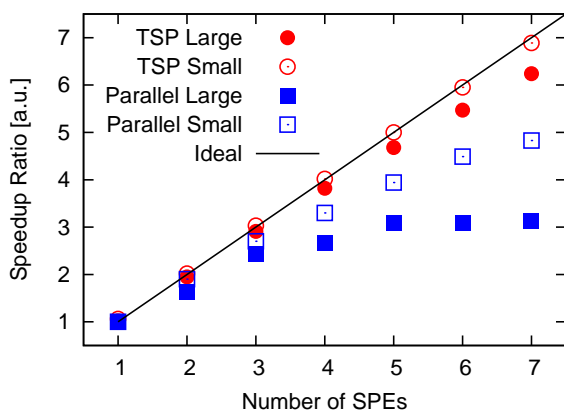


図 5: TSP と並列計算の速度向上比

る．一方，並列計算においてはSPE数を増加させていくと，SPE数が5のところで計算速度が飽和していることが分かる．大規模解析においてはSPE数の増加に伴いメインメモリへのDMA転送が増加し，転送の待ち時間が生じていると考えられる．SPE7基において，並列計算の効率は46%であるのに対し，TSPの効率は87%の効率が得られた．これより，SPE7基において，TSPは並列計算の約2倍の効率で計算できることが分かる．

次に，小規模解析について評価を行う．図5より，TSPにおいては大規模解析と同様SPE数に比例して計算速度が増加していることが分かる．一方，並列計算においてはSPE数を増加させていくと，計算速度はSPE数に比例せず，SPE数が多くなるほど速度の向上が小さくなることが分かる．並列計算では，DMA転送により隣のSPEと値を交換する．転送するデータ量は16バイトと小さいが，すべてのSPEがほぼ同時にDMA転送命令を発行するためDMA転送の終了待ちが起こり，速度向上が小さくなると考えられる．SPE7基において，TSPの効率は93%，並列計算の効率は69%であり，小規模解析においてもTSPは並列計算の1.3倍の効率で計算できることが分かる．

## 4 まとめ

電磁界解析の高速化を目的に，Cellプロセッサ上にTSPを用いたFDTD法を実装した．この結果，TSPを用いたFDTD法は，SPE数が増加しても速度向上は飽和せず，並列計算におけるメモリ帯域のボトルネックを解消できることを示した．SPE7基において，TSPは効率87%を達成し，並列計算の効率46%に対して約2倍の効率で計算できることを示した．小規模解析においてもDMA転

送の待ち時間を削減し，並列計算の約1.3倍の効率で計算できることを示した．

共有メモリのアーキテクチャを持つCPUは今後増加すると考えられるため，TSPは次世代のアーキテクチャにおいて有効な手法であるといえる．今後の課題としては，TSPの二次元，三次元解析への適用，実問題への応用が挙げられる．

## 参考文献

- [1] K.S. Yee. Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media. *IEEE Trans. Antennas and Propagat.*, Vol. 14, No. 2, pp.302–307, May 1966.
- [2] 中山ほか．Cellプロセッサによる計算機合成ホログラムの高速化．信学論(D)，vol.J91-D，no.8，pp.2035–2036，2008．
- [3] 岡田，近藤．Cell Broadband Engineを用いたマーカレスモーションキャプチャ．東芝レビュー Vol.62，No.6，2007.
- [4] 富田，山崎．Cell Broadband Engineに対する重力多体問題計算の最適化．Cell Speed Challenge 2008 自由課題部門，2008.
- [5] M. Xu, P. Thulasiraman. Finite-Difference Time-Domain on the Cell/B.E. Processor. *IEEE International Symposium on Parallel and Distributed Processing*, pp.1–8, 2008.
- [6] M. Xu, P. Thulasiraman. Parallel Algorithm Design and Performance Evaluation of FDTD on 3 Different Architectures: Cluster, Homogeneous Multi-core and Cell/B.E.. *10th IEEE International Conference on High Performance Computing and Communications*, 2008.
- [7] B. Li, H. Jin, Z. Shao. Two-Level Parallel Implementation of FDTD Algorithm on CBE. *IEEE International Conference on Networking, Sensing and Control*. pp.1812–1817, 2008.
- [8] 遠藤，園田，佐藤. FDTD法のFPGA実装における時空間パイプラインによる高速化. 信学論(B)，vol.J92-B，no.1，pp.243–249，2009．
- [9] IBM. Cell Broadband Engine Programming Handbook Version 1.11. 2008.
- [10] IBM. Cell Broadband Engine Programming Tutorial Version 3.0. 2007.