

GPU・FPGA アクセラレータを有するヘテロジニアス計算プラットフォーム上での 2次元 FDTD 計算の高速化

Acceleration of 2D-FDTD Computation on a Heterogeneous Platform with GPU and FPGA accelerators

○張山昌論*, ハシタ ムトウマラ ウィシディスーリヤ*, 武井 康浩*, 亀山充隆*

○Masanori Hariyama*, Hasitha Muthumala Waidyasooriya*, Yasuhiro Takei*,
Michitaka Kameyama*

*東北大学大学院情報科学研究科

*Graduate School of Information Sciences, Tohoku University

キーワード : ヘテロジニアスアーキテクチャ(Heterogeneous architecture), 高性能計算 (high-performance computing), スーパーコンピューティング (supercomputing), 時間領域差分法 (FDTD), FPGA(FPGA)

連絡先 : 〒 980-8579 仙台市青葉区荒巻字青葉 6-6-05
東北大学大学院情報科学研究科

張山昌論, Tel.: (022)795-7153, Fax.: (022)263-9167, E-mail: hariyama@ecei.tohoku.ac.jp

一般にアプリケーションにはデータインテンシブとコントロールインテンシブのように異なる種類のタスクが含まれている。ヘテロジニアスな演算処理はこのように異なるタイプのタスクを電力効率よく処理するために提案されている。ヘテロジニアスなプロセッシングには、図 1 のように CPU とアクセラレータのように異なる種類のプロセッサが用いられている。アプリケーションのタスクを最適なプロセッサに割り当てることにより、全体の処理性能を大きく向上させることができる。ヘテロジニアスな高性能計算 (HPC) 向けプラットフォームの例としては、Intel の CPU と NVIDIA の GPU (graphic processing unit) で構成される “Tianhe-1A”¹⁾ などが上げられる。GPU は高並列で規則的な浮動小数点演算に適しているが、それ以外の種類のタスクを高速化するのは GPU だけでは難し

い。この問題の解決法としては、FPGA(Field-Programmable Gate Arrays)を使用することがあげられる。FPGA はプログラム可能なロジックブロック、スイッチブロック、メモリモジュール、DSP モジュールを用いて演算部、メモリ、接続部などを任意の規模、構成でデータパスを合成することができる。そのため、GPU と FPGA を組み合わせることにより、さまざまなタイプの演算を高速化することができる。近年の FPGA では 30 万以上のロジックセルや 5MB 以上のメモリ、高速な通信インターフェースが利用できるため、1つの FPGA に数百のプロセッシングエレメント (PE) を持たせることができる。しかし、FPGA は多くのリソースを有し、かつ非常に低消費電力であるにもかかわらず、HPC 向けプラットフォームに用いられていることは少ない。主な原因としては、FPGA 向けプログ

ラミングが複雑になること、GPU と比べて浮動小数点演算の性能が低いことなどがあげられる。本論文では、さまざまな種類の演算を高速化できる GPU と FPGA アクセラレータを有するヘテロジニアスな演算プラットフォームを提案し、計算量を削減した精度ハイブリッドな 2-D FDTD 演算に適用した場合について述べる。

1. 2-D FDTD 法

1.1 概要

FDTD²⁾ は単純で非常に効率的であることから、最もよく使用されている電磁界シミュレーション手法の1つである。FDTD はマルチコア CPU や⁷⁾, ⁸⁾ のような GPU による高速化手法が多く提案されている。図 2 に FDTD の主なタスクを示す。はじめに、電界と磁界の初期値から最初のタイムステップでの電界の情報が得られる。その後、周期的な境界条件が適用される。それから、磁界の情報の取得と境界条件の適用を行う。この処理を与えられたタイムステップ数だけ続行する。式 (1) に電界の計算式を、式 (2)(3) に磁界の計算式を示す。 x, y, z 方向の電界、磁界はそれぞれ E と H 、タイムステップは n 、2次元の座標は i と j で表わされる。電界と磁界での境界における計算は異なることに注意する。FDTD の詳しいアルゴリズムは²⁾ によって与えられている。

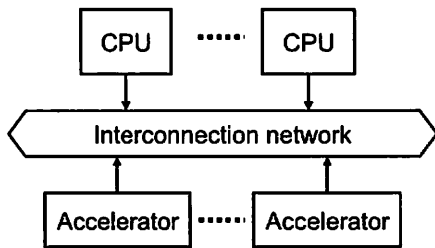


Fig. 1 ヘテロジニアスアーキテクチャ

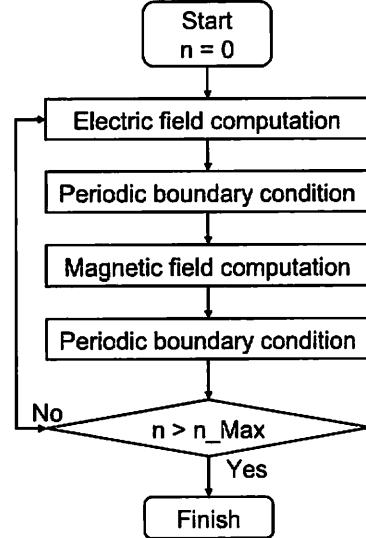


Fig. 2 FDTD のフローチャート

$$E_z^{n+1}(i, j) = E_z^n(i, j) - P_y(i, j) \left\{ H_x^{n+\frac{1}{2}}(i, j + 1/2) - H_x^{n+\frac{1}{2}}(i, j - 1/2) \right\} + P_x(i, j) \left\{ H_y^{n+\frac{1}{2}}(i + 1/2, j) - H_y^{n+\frac{1}{2}}(i - 1/2, j) \right\} \quad (1)$$

$$H_x^{n+\frac{1}{2}}(i, j + 1/2) = H_x^{n-\frac{1}{2}}(i, j + 1/2) - Q_y(i, j) \{ E_z^n(i, j + 1) - E_z^n(i, j) \} \quad (2)$$

$$H_y^{n+\frac{1}{2}}(i + 1/2, j) = H_y^{n-\frac{1}{2}}(i + 1/2, j) - Q_x(i, j) \{ E_z^n(i + 1, j) - E_z^n(i, j) \} \quad (3)$$

更なる高速化のために、光共振器のアプリケーション³⁾ の特徴について観察する。³⁾ によると、光共振器の外側の空洞の電磁界は内側に比べて弱くなっている。図 3 に FDTD による各格子点での計算結果を示す。これより、原点から遠い格子点での電界のダイナミックレンジは小さくなっていることがわかる。同様に、磁界においても似たような特徴が観察されている。これらの観察結果より、ダイナミックレンジが狭い領域では計算精度を減らすことができると考えられる。

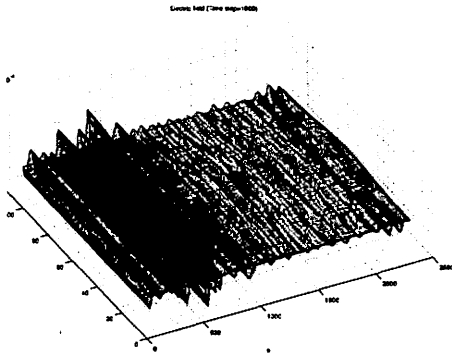


Fig. 3 電界のダイナミックレンジ

1.2 倍精度と単精度浮動小数点を組み合わせた GPU による高速化⁴⁾

ダイナミックレンジが大きい領域では倍精度浮動小数点を、ダイナミックレンジが小さい領域では単精度浮動小数点を用いている。図4のように、全体の境界と area 1 においては倍精度浮動小数点演算を、一方で area 2 においては単精度浮動小数点演算を行っている。この手法の問題点の1つとして float と double の変換によるオーバーヘッドがあげられる。式(1)のように、座標 (i, k) における電界の計算において左右の座標における磁界の値が要求される。同様に、座標 $(i, k+1/2)$ における磁界の計算においても左右の座標における電界の値が要求される。図4において分割された境界における電磁界計算のために、area1 と area2 にそれぞれ単精度と倍精度の値が必要となる。結局、float と double の変換が必要となり、オーバーヘッドが発生する。図5に単精度/倍精度ハイブリッドな浮動小数点演算のフローチャートを示す。

1.3 倍精度浮動小数点と固定小数点を組み合わせた FPGA による高速化⁵⁾

表1に浮動小数点と固定小数点による FDTD の演算精度を示す。倍精度浮動小数点を用いた手法1について、このデータを要求される結果として仮定する。手法3と4のデータより、固定小数点を用いた結果における精度の損失は非常

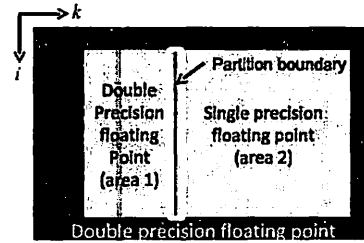


Fig. 4 単精度/倍精度演算への分割

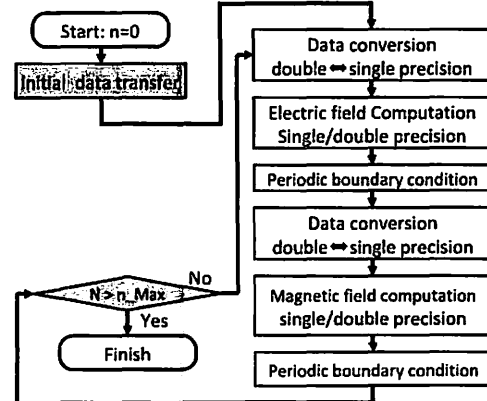


Fig. 5 単精度/倍精度ハイブリッド演算に小さいことがわかる。加えて、単精度浮動小数点を用いた手法2よりも精度が高くなっている。FDTD のアルゴリズムにおいて、電磁界の境界付近のデータのダイナミックレンジは非常に大きくなっているため、このエリアでは倍精度浮動小数点が必要される。これと比べて、中間領域のダイナミックレンジは非常に小さいため、32bit 固定小数点の効果的となる。これらの結果より、全体の演算の95%について、FPGA 上の 32bit 固定小数点 ALU を用いることにより、消費電力を 20W 以下と非常に小さくすることができる。

2. GPU・FPGA アクセラレータを有するヘテロジニアスプラットフォームフォーム

2.1 概要

本研究ではマルチコア CPU, GPU, FPGA からなるヘテロジニアスな環境を使用した。図6にヘテロジニアスプラットフォームの演算ノードを示す。FPGA と GPU はそれぞれ PCI Express

Table 1 演算精度とエラーの関係

Method		Maximum absolute error	
		Electric field	Magnetic field
1	Double-precision floating point (DPFP)	-	-
2	Single-precision floating point	1.16×10^{-5}	9.20×10^{-6}
3	5% DPFP 95% 32bit fixed-point	1.50×10^{-7}	1.92×10^{-7}
4	25% DPFP 75% 32bit fixed-point	6.80×10^{-8}	6.39×10^{-8}

で接続されている。このプラットフォームの仕様について表 2 に示す。

2.2 GPU による高速化⁴⁾

本研究では CPU(Intel core i7 960) と GPU (GeForce GTX 590) を使用した。図 7 のように、提案手法では従来の倍精度による GPU での演算と比べて 1.79 倍の高速化を達成した。倍精度浮動小数点演算には 2 つの CUDA コアが必要になるが、単精度では 1 つで済む。そのため、

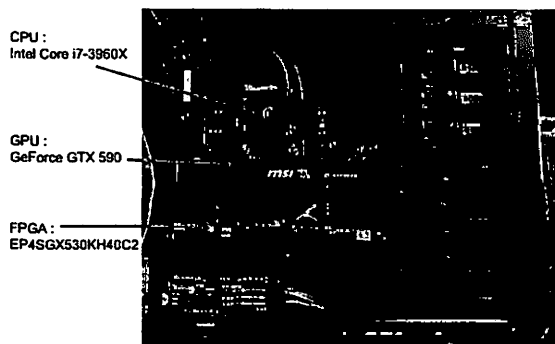


Fig. 6 ヘテロジニアスプラットフォームの演算ノード

Table 2 ヘテロジニアスプラットフォームの仕様

Processor	Number of cores	Maximum power
CPU: Intel core i7 3960X	6	130W
GPU: GeFoce GTX5900	1024	360W
FPGA: DE4 board ⁶⁾ Stratix IV GX EP4SGX530	191 PEs	20W

提案手法では演算を高速化することができる。

図 8 に演算時間と倍精度演算の関係を示す。“double precision computation area” は倍精度演算が実行された格子点の割合を指すことに注意する。演算時間は倍精度演算の割合が増えるほど増加しているが、倍精度演算が 95% の場合は 100% の場合と比べて演算時間が大きくなった。これは、単精度/倍精度ハイブリット演算において、float から double に数値を変換するときのオーバーヘッドによるものである。

2.3 FPGA による高速化⁵⁾

CPU と FPGA による低消費電力ヘテロジニアスプラットフォームの主な問題点の 1 つに、FPGA へのプログラミングの複雑さがあげられる。よりプログラミングを単純なものにするため、CUDA-like なアーキテクチャを提案する。図 Figure 9 に提案する FPGA アーキテクチャを示す。このアーキテクチャは Nios II CPU コア、オンチップメモリ、SIMD アクセラレータコアによって構成されている。外部から DDR2 SDRAM が Nios II CPU コアを通じて FPGA ボードに接続されている。Nios II CPU コアは主にアクセラレータの制御ユニットとして使用される。提案する SIMD アクセラレータは GPU と似たデザインをしているため、同じ CUDA のコードを使用することができる。SIMD アクセ

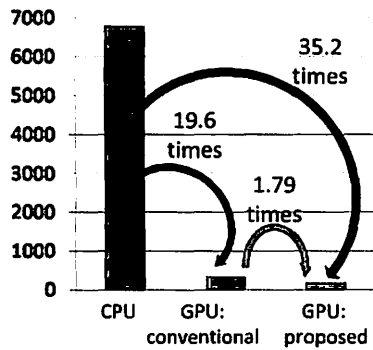


Fig. 7 演算時間の比較

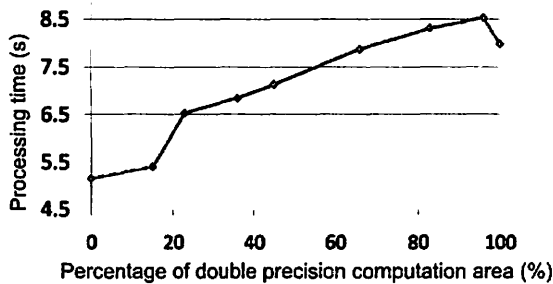


Fig. 8 ハイブリッド演算の精度
 ラレータの基本的なアイデアやプログラミング環境については、先行研究⁹⁾において議論されている。⁹⁾と本論文におけるアーキテクチャにおいて、PEの構造に大きな違いがある。

図10に32bit固定小数点PEのアーキテクチャを示す。このPEは5つの加算器と3つの乗算器から構成されている。データパスは完全にパイプライン化されているため、パイプラインが満たされたときには毎サイクルごとに演算結果が出力される。このPEはFPGA(Stratix IV GX EP4SGX530)に191個搭載することができる。さらに高性能なFPGAを用いた場合、より多くのPEを搭載することができる。

提案するプラットフォームにおいては、境界に近い電磁界のデータはCPU (Intel core i7-3960X)で、残りのデータはFPGAで計算される。境界データが演算された後にFPGAに転送されるが、CPUにおける境界データの演算とFPGAにおける電磁界の計算をオーバーラップさせることによってCPUの計算オーバーヘッドを隠すことができる。

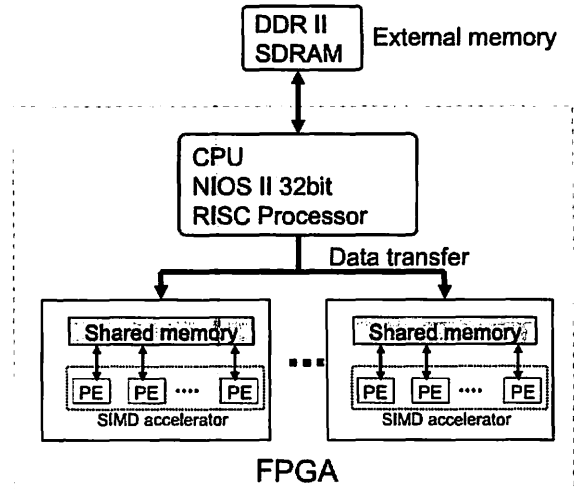


Fig. 9 FPGAプラットフォーム全体のアーキテクチャ

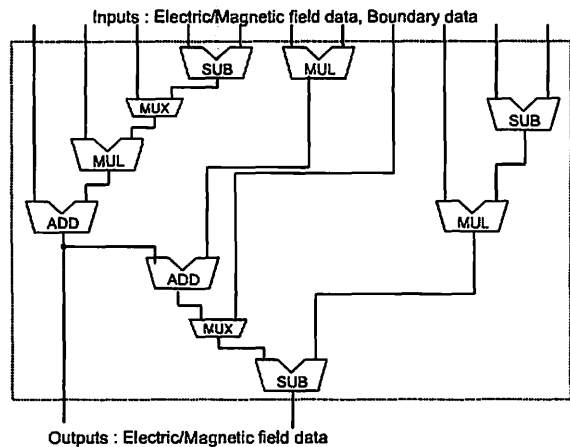


Fig. 10 PEのアーキテクチャ

提案するアーキテクチャをAltera DE4⁶⁾ボード上のStratix IV GX EP4SGX530に実装した場合について、リソース使用量、消費電力、周波数の見積もりを立てた。表3の見積もり結果から、動作周波数が50MHzで191個以上のPEを実装できることがわかる。また、消費電力は20W以下になる。

表4にCPU/GPUヘテロジニアスプラットフォームとの比較を示す。CPU/GPU実装での演算時間評価はVisual studio 2008でのCUDA timerを使用した。CPU/FPGAにおける演算時間は表1の手法3を用いて動作周波数50MHzで演算させた場合における数値である。CPU/FPGA

Table 3 FPGA 実装時のリソース使用量の見積もり

Resources (191 PEs)	394503 Logic blocks (92.8%) 417140 Registers (98.1 %) 1020 DSP units (99.6%) 20M bits (98.87%)
Frequency	50 MHz
Power (estimated)	Less than 20W

Table 4 演算時間

Method	Processing time (s)
CPU + GPU	10.08 (measured)
CPU + FPGA	11.26 (estimated)

の実装では、CPU/GPUの実装結果と比べてわずかに計算速度が遅くなっているが、消費電力は10分の1以下に抑えることができる。演算時間の90%以上が境界データのグローバルメモリからローカルメモリへの通信時間であることに注意すると、データ圧縮などの通信データ削減手法を適用する必要があると考えられる。

3. むすび

演算精度をできるだけ落とさずに高速化を行うために、高い演算精度が必要な領域を倍精度で、あまり演算精度が必要とされない領域を単精度または固定小数点で計算する精度ハイブリッドアプローチを提案した。さらに、精度ハイブリッドアプローチをGPU、FPGAを用いて実装し、高速化に有用であることを実証した。

謝辞

本研究はMEXT 科研費 12020735 の助成を受けたものである。

参考文献

- 1) <http://www.top500.org/system/10587>
- 2) H. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media", IEEE Transactions on Antennas and Propagation, Vol.14, No.3, pp.302-307, 1966.
- 3) Y. Ohtera, S. Iijima and H. Yamada, "Cylindrical Resonator Utilizing a Curved Resonant Grating as a Cavity Wall", micromachines, pp.101-113, 2012.
- 4) Hasitha Muthumala Waidyasoorya, Yasuhiro Takei, Masanori Hariyama and Michitaka Kameyama "Hybrid Single/Double Precision Floating-Point Computation on GPU Accelerators for 2-D FDTD", International Conference on Parallel and Distributed Processing Techniques and Applications(PDPTA) (to be appeared).
- 5) Hasitha Muthumala Waidyasoorya, Yasuhiro Takei, Masanori Hariyama and Michitaka Kameyama, "Low-Power Heterogeneous Platform for High Performance Computing and Its Application to 2D-FDTD Computation", International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)(to be appeared).
- 6) <http://www.altera.com/education/univ/materials/boards/de4/unv-de4-board.html>
- 7) Z. Bo, X. Zheng-hui, R. Wu, L. Wei-ming, S. Xin-qing, "Accelerating FDTD algorithm using GPU computing", International Conference on Microwave Technology & Computational Electromagnetics (ICMTCE), pp.410-413, 2011.
- 8) T. Nagaoka and S. Watanabe, "A GPU-based calculation using the three-dimensional FDTD method for electromagnetic field analysis", International Conference on Engineering in Medicine and Biology Society (EMBC), pp.327-330, 2010.
- 9) H. M. Waidyasoorya, M. Hariyama and M. Kameyama, "Architecture of an FPGA-Oriented Heterogeneous Multi-core Processor with SIMD-Accelerator Cores", International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA), pp.179-186, 2010.
- 10) A. Taflove and M. E. Brodwin, "Numerical Solution of Steady-State Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations", IEEE Transactions on Microwave Theory and Techniques, Vol23. Issue 8, pp.623-630, 1975.