

## CPU-FPGA 密結合アーキテクチャを用いた Full-PIC 法の高速化の検討

酒井 諒太郎† 宮島 敬明‡ 杉本 成† 藤田 直行‡ 天野 英晴†

†慶應義塾大学 理工学部 情報工学科 ‡宇宙航空研究開発機構 航空技術部門 数値解析技術研究ユニット

## あらまし

Full-PIC 法は、近年注目されている電気推進エンジンのひとつであるホールスラストの、高精度なシミュレーションとして用いられている。高精度な試験設備を整えるのが困難なホールスラストは、計算機によるシミュレーションが不可欠である。Full-PIC 法により、実機でのテストに比べて非常に低コストで試験を行えるものの、処理に膨大な時間がかかるという問題がある [1]。そこで、高速化のために、GPU によってオフロードを試みた先行研究が存在する [2]。今回我々は、CPU-FPGA 密結合アーキテクチャである Zynq 上の CPU に Full-PIC 法を実装し、そのコード内の高負荷な部分を Zynq 上の FPGA でオフロードすることにより高速化が可能であるか検討する。

## 1 Full-PIC 法

粒子法とは、連続体を粒子の集合として解析する手法であり、粒子の衝突の処理の仕方などによって様々な手法が存在する。ホールスラストは、ホール効果を利用した静電加速方式のエンジンであるため、そのシミュレーションはプラズマのシミュレーションの一種である。そこで、プラズマの生成、加速や衝突をシミュレーションするために粒子法ベースの手法が用いられており、ホールスラストの放電の支配方程式は、確率解法である Particle-In-Cell (PIC) 法を用いて解くことが可能であることが知られている [3]。

PIC 法では、電場や磁場の場の量の計算に空間格子を用いており、空間を自由に飛び回る各粒子を、格子の最小単位である 1 セル内に含めて計算している。今回取り扱う Full-PIC 法は、電子のみでなくイオンについても粒子として扱う PIC 法であるために、非常に計算コストが高いが高精度である。ゆえに、希薄なプラズマを取り扱うために流体力学的な近似をするのが難しいホールスラストのシミュレーションに用いられている。Full-PIC 法の流れを図 1 に示す。

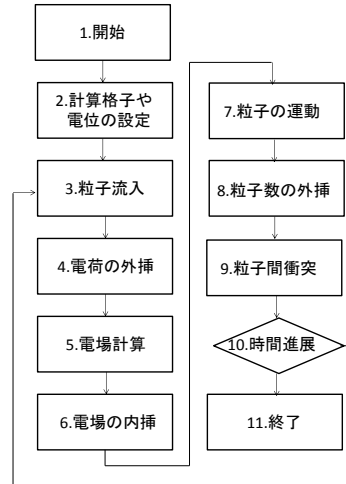


図 1: Full-PIC 法の流れ

## 2 実装と評価

## 2.1 実装

実装には Xilinx 社の Zed Board を用いた。Zynq の CPU 上で Linux を動作させているため、CPU-FPGA 間の通信や処理時間の計測などのあらゆる操作が Zed Board 単体で可能である。JAXA で現在研究開発中である CPU 用シミュレーションコード”NSRU-Full-PIC”を実装の対象とし、図 1 のステップ 4 に含まれるサブルーチン”field\_source”を FPGA へのオフロード対象とした。

”field\_source”は、ホールスラスト壁面の条件を再現するサブルーチンであり、電場・イオン相互作用を計算している。グリッドの 4 点ステンシル計算を行うためにそのコード内で比較的高負荷である。FPGA 実装には、Vivado HLS による高位合成を用い、Xilinx IP core を含む全てのモジュールは Vivado IP integrater で接続されている。なお、このシミュレーションコードは Fortran90 で記述されているため、対象のサブルーチンのコードを一旦 C 言語で書き直す手順を挟んだ。

実装の詳細を以下に示す。

- SoC: Zynq-7000 All Programmable SoC
  - CPU: Cortex-A9 (667MHz)
  - FPGA: Artix-7
- メモリ: DDR3 512MB

Acceleration of Full-PIC method with a tightly coupled CPU-FPGA architecture

†Ryotaro Sakai ‡Takaaki Miyajima †Naru Sugimoto

‡Naoyuki Fujita †Hideharu Amano

†Keio University

‡Japan Aerospace Exploration Agency

- OS: Linaro Ubuntu 14.04.3 LTS
- コンパイラ: GNU Fortran 4.8.4, gcc 4.8.4
- ツール: Vivado HLS 2014.2, Vivado 2014.2

また、実装した”field\_source”のブロック図を図2に示す。”field\_source”は3つのループから構成されており、各ループをそれぞれモジュール化した。全てのモジュールは100MHzで動作する。すべてのポートはAXI Streamでdouble型64bit幅のデータの転送をしている。”First loop”の22入力ポートと”Third loop”の2出力ポートは、22個のAXI DMA IP coreを通じてプロトコルの変換後にデータ転送を行っている。また、3つすべてのモジュールはAXI Liteを通してCPUから制御される。

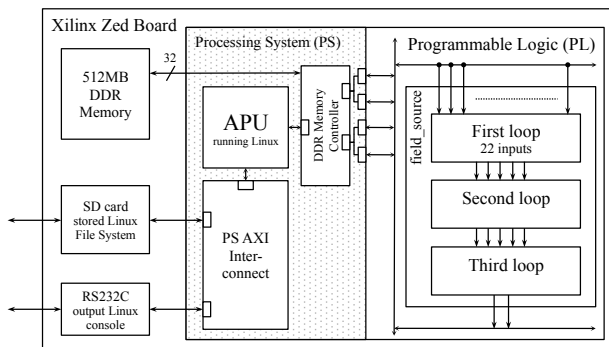


図2: field\_source のブロック図

## 2.2 評価

まず、表1に、図1のステップ4の”電荷の外挿”内の処理のCPUでの測定結果を示す。

表1: ”電荷の外挿”内の処理のCPUでの測定結果

処理内容	処理時間 [ $\mu$ s]
初期化	3805.26
電荷の外挿	140679.94
イオンの外挿	90325.72
電子の場の計算	6548.29
イオンの場の計算	9399.71
電場・イオンの相互作用の計算	42300.94
境界処理	2891.53
ポアソン方程式	7451.57
ポアソン方程式右辺の計算	8504.20

表1内の”電場・イオンの相互作用の計算”は、サブルーチン”field\_source”の処理である。次に、”field\_source”のFPGA実装の処理時間を示す。現状、Linux上から作成したモジュールを操作できていないために、見積もりである。多数の入力ポート故にボトルネックとなる”First loop”のみを考慮する。DDRのバンド幅を考

慮して、DDRからAXIバースト転送し続けることが可能なポート数を3とみなす。その3ポートでの入力を”First loop”の22入力に極力均等に分配するため、インターバルは8とする。また、今回モデルのグリッド数は $271 \times 311 = 84181$ であるので、

$$8 \times 84181 \times 10ns \times 10^{-3} = 6734.48\mu s$$

となり、”field\_source”はCPU実装に対しておよそ6.28倍の高速化が見込めた。最後に、モジュール”field\_source”のリソースの使用率を表2に示す。

表2: field\_source のリソースの使用率

リソース	使用率 [%]
FF	42.73
LUT	65.81
BRAM	42.86
DSP48	53.18

リソースの使用率は全体的に高く、現状モジュールは1つしか実装できないと思われる。

## 3 結論

本稿では、CPU-FPGA密結合アーキテクチャを用いたFull-PIC法の高速化を検討した。ホールスラスト用シミュレーションコード”NSRU-Full-PIC”をZynqのCPU上で動作させ、高速化のために一部のサブルーチンのFPGA実装を試みた。見積もりではあるがCPUに対しておよそ6.28倍の高速化が見込めた。今後は実装を進めるとともに、高いリソース使用率の改善のため設計を最適化し、オフロード可能な範囲を拡大したい。

## 参考文献

- [1] Sigeru Yokota, Kimiya Komurasaki, and Yoshihiro Arakawa. Plasma density fluctuation inside a hollow anode in an anode-layer hall thruster. Reston, Va, 2006. American Institute of Aeronautics and Astronautics.
- [2] Takaaki Miyajima, Shinatora Cho, and Naoyuki Fujita. A study of gpu acceleration of ”source” part in hall-thruster simulation. In *IEICE Tech. Rep.*, Vol. 115 of *CPSY2015-62*, pp. 7–12, Dec. 2015.
- [3] Shinatora Cho, Kimiya Komurasaki, and Yoshihiro Arakawa. Kinetic particle simulation of discharge and wall erosion of a hall thruster. *Physics of Plasmas*, Vol. 20, No. 6, Jun 2013.