

**科学研究費助成事業 研究成果報告書**

平成 28 年 10 月 21 日現在

機関番号：18001

研究種目：基盤研究(C) (一般)

研究期間：2013～2015

課題番号：25330067

研究課題名(和文)再構成型LSIによるCFD力学向け計算機システムの構成技術に関する研究

研究課題名(英文)Development of elemental technology of reconfigurable LSI based computer systems for CFD acceleration

研究代表者

長名 保範(Osana, Yasunori)

琉球大学・工学部・助教

研究者番号：00532657

交付決定額(研究期間全体)：(直接経費) 3,800,000円

研究成果の概要(和文)：FPGAによる数値計算のアクセラレーション技術は消費電力の面で有利であることから、GPUの次のアクセラレータとして注目を集めているが、アプリケーション毎にFPGA上の回路設計を行う必要があり、普及のネックとなっている。本研究課題では、計算流体力学を主なターゲットとして、大規模な回路を自動分割してシステムを構成する手法、商用の高位合成ツールを用いて元のソースコードをなるべく修正せずにシステム全体を高速化する手法、および科学技術計算でひろく用いられているFortranからの高位合成を行うツールを開発した。この成果は流体力学に限らずひろく応用が期待できるものである。

研究成果の概要(英文)：FPGAs are attractive candidate of accelerator device of next generation, because of its high power efficiency. However, FPGA based accelerator requires custom circuit design for each applications. This project choose CFD as the main target, to establish basic technics and methods to: automatically divide a large problem onto multiple FPGAs, and to apply FPGA acceleration on an existing source code by exploiting power of high-level synthesis (HLS) tools with minimal modifications on the source code. Because most HLS tools don't support Fortran (while Fortran is still popular in scientific computing), a Fortran HLS tool had developed in this project, too. The methods and tools are not limited to CFD, but also widely applicable to various area of scientific computation.

研究分野：計算機システム

キーワード：FPGA 数値計算 アクセラレーション技術 高位合成

## 1. 研究開始当初の背景

国内では理化学研究所の「京」コンピュータの運用が開始され、10PFLOPSを超えるようなスーパーコンピュータが国内外で稼働するようになった。多くのスーパーコンピュータはスーパースカラ型のプロセッサの大規模な並列システムであり、システムの大規模化に伴う消費電力の増大という問題に直面している。

そこで電力あたりの計算能力を高めるために、FPGA や GPGPU といったアクセラレータデバイスを導入する動きが盛んであるが、GPGPU は C 言語ベースの開発ツールや、マルチユーザ環境で稼働させるための環境が整っており、比較的計算センタ等の大規模並列システムで導入しやすい状況にあるのに対して、FPGA は計算エンジンとして有望視されてはいるものの、計算センタで運用するには、設計ツールやマルチユーザ環境での使用といったユーザビリティにおいて後れをとっている状況であった。

FPGA はそのユーザビリティにおいて GPGPU に後れをとっているものの、任意の論理回路をハードウェアとして構成可能であるため、対象の問題をハードウェアで直接解くことが可能であり、設計フローや運用方式を改善できれば、電力対性能比に優れた計算エンジンとして期待できると考え、本研究に着手した。

## 2. 研究の目的

FPGA による科学技術計算のアクセラレーションに関する研究は、2000 年代に入って急速に進んだものの、GPU コンピューティングの発展によって国際的にも停滞傾向であった。しかし、FPGA の低消費電力性能などがふたたび見直されており、FPGA による科学技術計算のアクセラレーションを容易に行えるようにする手法の確立が必要となっている。

そこで、本研究課題では、FPGA を用いた科学技術計算の高速化に不可欠な、不規則メモリアクセスや不定形処理への対応、容易なプログラミング手法、大規模な計算に対応するための多数の FPGA から構成されるマルチ FPGA システムに対応した設計、計算センタなどの共有環境での運用などについての基礎的な手法の確立が必要であると考えた。

これを達成するために、

- 不定形なメモリアクセスパターンを含むアプリケーションを FPGA に実装する際のチューニング手法
- 大規模な演算パイプラインを自動的に分割して、マルチ FPGA システム上に配置するための設計ツール
- システム全体を Fortran や C 言語などの高級言語で設計する手法
- マルチユーザ環境で扱えるような標準的なホストインタフェースへの対応と、共有環境への適応

などを開発・確立することにより、GPU のように実用的なアクセラレータとして FPGA を利用可能にすることを研究目標とした。

## 3. 研究の方法

実アプリケーションをもとにした研究を行うため、共通のターゲットとして JAXA で開発されている CFD ソフトウェア FaSTAR を用い、ソースコードのうち適切な部分、あるいは全部を用いて、研究目的の項に述べたような内容の研究を実施した。ただし、研究の内容によってはこれを C 言語に書き換えて用いたり、Fortran コードに直接適用できるレベルに達していない場合には別のコードを用いて評価を行っている場合もある。

FaSTAR のコード解析やテストデータの作成を JAXA が行い、不定形アクセスやマルチ FPGA システムへの対応を琉球大が、システム全体を Fortran や C 言語で設計する手法やホストインタフェースについては慶應義塾大と東京農工大が研究・開発を行った。

## 4. 研究成果

### (1) 不定形メモリアクセスへの対応

科学技術計算で使用されるようなマイクロプロセッサのメモリシステムには強力なキャッシュや、複数のバンクにインターリーブされた高バンド幅のメモリが付随している。したがって通常、プログラマはこの詳細についてあまり意識しなくても、アクセスの局所性など基本的な事項にだけ注意していれば、高い実効メモリバンド幅を得ることができる。

それに対して、FPGA を用いた計算機では一般にキャッシュメモリは用意されず、FPGA のユーザ I/O ピンに直結された SDRAM をどのように動かすかも完全にユーザの責任である。この場合、FPGA 上の回路で計算処理を行ったときの実効バンド幅は SDRAM にどのようにデータを配置したかで大きく左右されることになるので、メモリ配置の最適化は重要な課題であるものの、ハードウェアの設計前に性能予測と最適化を行うことは困難である。また、メモリ上のデータの再配置を行う場合には多少なりともハードウェアの再設計が必要であり、検証も含めると設計上の大きな負担となってしまうことが問題である。

そこで、多くの場合には演算ハードウェアの設計の前にアルゴリズムを検証するためのソフトウェア実装を行うことに注目し、このコードを用いてメモリアクセスのチューニングを行うことを可能にするためのツール(引用文献①)を作成した。

このツールは、C++ で記述された SDRAM モデルと、それを配列のように扱うためのテンプレートクラスから構成されており、メモリのデータシートに記載されたパラメータを

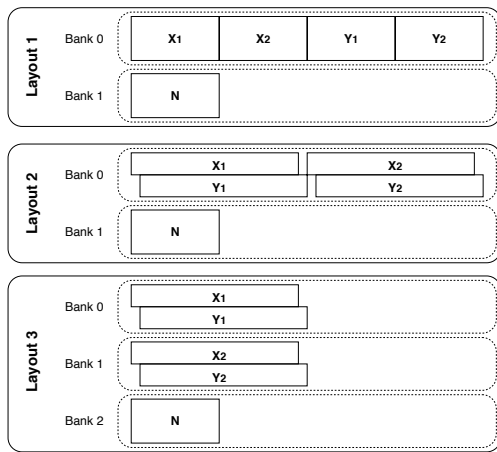


図1 メモリ上へ3つの配列を配置する例

SDRAM モデルに設定することで各種の SDRAM の挙動をクロックサイクルレベルでエミュレートすることができる。

配列として扱うためのテンプレートクラスを用いれば、ソースコードの変更を大きく変更することなく、コード中の配列をどのように SDRAM 上に配置するかを設定することができる。たとえば図 1 のようにそれぞれの配列をどのバンクのどこから配置するかを指定したり、あるいは複数の配列をインターリーブして配置する、といった具合である。

これにより、ソフトウェアで記述されたアルゴリズムを実行するのに必要なメモリアクセスに所要するクロックサイクル数を容易に計算し、各種の配置を試行することが可能になる。これにより、不定形メモリアクセスを含む数値計算アプリケーションの、ハードウェア化を全体としたメモリアクセスの性能改善を効率よく行うことができるようになった。

## (2) マルチ FPGA システムへの対応

FPGA を用いて数値計算アプリケーションを実装することのメリットのひとつに、計算式を完全にパイプライン化したハードウェアとして実装することで高いスループットが得られることがある。しかし、計算式を完全にパイプライン化するとハードウェアの規模が大きくなり、アプリケーション全体を単一の FPGA で実装することが困難なことは少なくない。このような場合にはハードウェアを分割して複数の FPGA に配置する必要があるが、回路の分割は通常の FPGA 設計ツールのデザインフローではサポートされていない。

一般的には、LSI のプロトタイピングなどを目的として大規模な回路を複数の FPGA に分割する場合、hMETIS(引用文献②)などのツールを用いてハイパーグラフとして回路を分割するなどの手法が採られている。このような手法を用いれば任意の回路を分割することが可能であるが、回路分割に時間がかかるなどの問題がある。しかし、本研究課題のような数値計算では演算パイプラインの分割配置だ

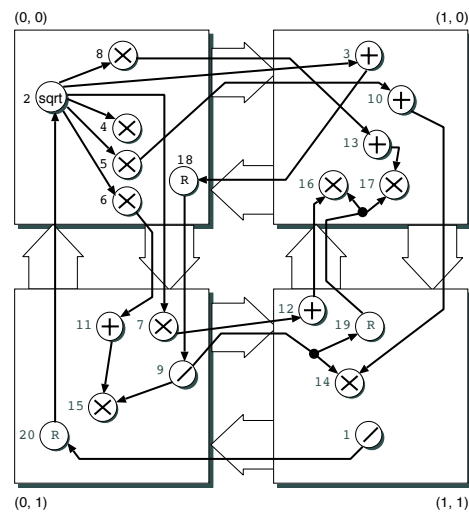


図2 データフローグラフの分割配置

けに単純化できるため、浮動小数点演算器やユーザ定義のモジュールを最小単位として回路を分割配置する手法を開発(引用文献③)した。この方法では、入力された数式をデータフローグラフに変換し、FPGA の面積や演算器の面積、FPGA 間の通信路の混雑度などを考慮しながら最適化し、図 2 のようにデータフローグラフを自動分割配置し、回路を生成する。

これにより、Fortran や C のような文法で記述した数式を入力として、演算のためのデータフローを生成し、演算器や FPGA 間の通信のための制御回路をふくめて一括で自動生成できるようになった。ハードウェアに依存する部分は適切に抽象化されており、新しい FPGA や、異なる FPGA 間通信ハードウェアにも容易に対応することができるため、大規模な数値計算パイプラインのマルチ FPGA 実装に広く応用することができる。

## (3) 高級言語での設計手法

(2)で述べたような手法とツールの開発によって、主要なデータパスの自動設計は可能になったものの、アルゴリズム全体の実装には、周辺の回路をハードウェア記述言語(HDL)などによって設計する必要があり、アルゴリズムが複雑な場合には HDL での設計と検証に多くの時間が必要である。

また、実際にはアプリケーション全体を FPGA に実装するのではなく、処理に時間のかかる計算アルゴリズムのコアのみをハードウェア化し、データの前処理・後処理や可視化などはソフトウェアで実装したほうが、自由度の高い使い方が可能である。このためには、元のソフトウェアの一部分だけを抜き出して、容易にアクセラレータへオフロードすることのできる設計・開発手法の確立が必要である。

そこで、ARM CPU コアと FPGA がひとつのチップに集積された Xilinx 社の Zynq-7000 SoC を用いて、ARM コアで動作する Linux 上のソフトウェアからアクセラレータハードウェア

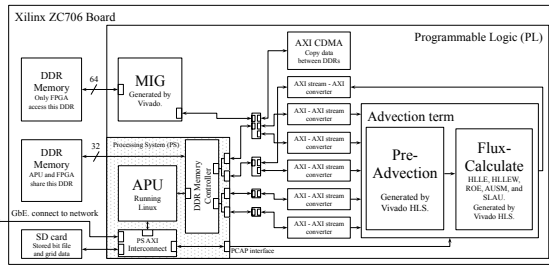


図3 Zynq SoC を用いたシステム構成

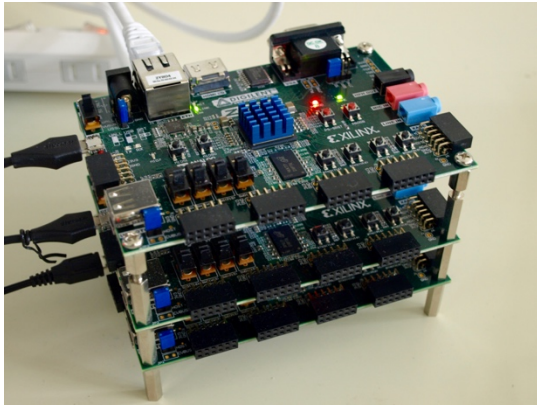


図4 Zynq クラスターのプロトタイプ

に容易にアクセスできる環境を構築した。さらに、同じく Xilinx 社から提供されている高位合成ツール Vivado HLS を用いて FPGA 上のアクセラレータ回路を実装することにした(引用文献④)。

この環境で、JAXA により開発された計算流体力学の Fortran コードの一部を C 言語に書き直してハードウェア化して FPGA にオフロードし、残りの部分はオリジナルの Fortran コードのままコンパイルして ARM プロセッサ上で実行するための開発フローを確立した。

この開発フローでは、基本的にオリジナルの Fortran ソースコードをそのまま用いることができる。変更が必要なのは、Vivado HLS は Fortran をサポートしていないため、オフロードの対象とするサブルーチンを C 言語に書き直す必要がある点と、そのサブルーチンの呼び出しの前後に FPGA との間のデータ転送のためのコードを挿入する点である。図 3 に示すように、アクセラレータの設計にあたっては、FPGA 側のメモリのアドレスマッピングなどを決定する必要はあるものの、基本的に C 言語記述と IP コアブロックの接続だけで完結しており、HDL を記述する必要がなく、従来の一般的な手法と比べると大幅に生産性が向上した。

使用した Zynq SoC の FPGA 部の面積が小さく、あまり大きな回路をオフロードできなかったことと、ARM プロセッサの性能がそれほど高くないことから、全体では Intel の Xeon プロセッサに比べると性能が大きく劣るものの、FPGA に実装した部分の性能は Xeon プロセッサを上回っており、システム全体でも ARM 単体での実行に比べれば高速化が実現できている。また、エネルギー効率的にも優れている。

るため、今後より大きなデバイスに展開したり、あるいは図 4 のようなクラスタシステムを構築することで、パラメータサーベイのような用途への応用が期待できる。

#### (4) Fortran からのシステム設計

(3)で述べたように、高位合成技術によるアクセラレータのハードウェア開発はある程度実用的な手法となりつつあるが、高位合成ツールの多くは C や Java からのデザインエントリを前提としており、膨大な HPC 資産のある Fortran は取り残されたような状態になっている。そこで、Fortran からのデザインエントリを可能にするためにまず、Java ベースの高位合成ツール JavaRock-Thrash のソースとなるように、Fortran のプログラムを Java に変換するシステム F2JRT (Fortran to JavaRock-Thrash)を開発し、評価を行った(引用文献⑤)。しかし、Java がサポートしていない goto 文などの変換が困難であったり、回路の性能が JavaRock-Thrash に依存しているなどの問題が生じた。C 言語ベースの設計ツールでも同様に Fortran から C へのトランスレータである f2c を使用することを試みたが、f2c は Fortran95 などの新しい文法に対応しておらず、やはり問題が生じた。

これらの問題点を解消するために筆者は、本研究の提案手法である FortRock の開発を行った(引用文献⑤)。FortRock では、F2JRT の開発時に生じた問題点を解決するために、Fortran で書かれ得たプログラムを入力として、直接 Verilog HDL を出力する。FortRock では、Fortran で記述されたソースコードを入力とし、一旦 LLVM の Fortran フロントエンドである DragonEgg を用いて、LLVM の中間表現である LLVM IR に変換する。その後、LLVM IR レベルで重複命令の削除やレジスタ名の指定などの前処理を行い、次に FortRock のコア部分である FortRock Core によって最終的な Verilog HDL の出力を得る。

これによって、オリジナルの Fortran ソースコードを他の言語に書き直すことなくハードウェア化することができるようになり、FPGA をアクセラレータとして利用する際の設計時間を短縮することができた。

#### <引用文献>

- ① 新垣誠, 喜屋武克樹, 平井裕介, 仲宗根宏貴, 長名保範: “カスタム計算機におけるメモリレイアウト最適化のためのチューニングツールの開発”, 第 14 回情報科学技術フォーラム (FIT2015). Sep. 2015.
- ② George Karypis, Rajat Aggarwal, Vipin Kumar and Shashi Shekhar: “Multilevel Hypergraph Partitioning: Applications in VLSI Domain.” IEEE Transactions on VLSI Systems, Vol. 7, No. 1, pp. 69-

79, 1999.

- ③ 平井裕介, 喜屋武克樹, 新垣誠, 天野英晴, 藤田直行, 長名保範: “マルチFPGAシステムにおける演算パイプラインの自動設計の実現に向けた研究.” 電子情報通信学会. RECONF2014-34. Nov. 2014.
- ④ Naru Sugimoto, Takaaki Miyajima, Ryotaro Sakai, Yasunori Osana, Naoyuki Fujita and Hideharu Amano: “Zynq Cluster for CFD Parametric Survey.” in Proceedings of 2016 International Symposium on Applied Reconfigurable Computing, pp. 287-299. Mar. 2016.
- ⑤ 山下 貴大, 五十嵐 雄太, 中條 拓伯: “Fortranによる高性能計算のハードウェア化と高位合成ツール FortRock.” 組み込みシステムシンポジウム 2014 (ESS2014) 論文集, pp. 90-95. Dec. 2014.

#### 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 7 件)

- ① 松田和也, 三好健文, 船田悟史, 中條拓伯: “Java 言語ベース高位合成ツールを用いた回路分割方式の実装と評価.” 情報処理学会論文誌, Vol. 56(8), pp. 1582-1592. 2015年8月. 査読有.
- ② 小池恵介, 三好健文, 五十嵐雄太, 船田悟史, 中條拓伯: “Java 言語ベース高位合成ツールによるアクセラレータ開発環境.” 電子情報通信学会論文誌, Vol. J98-D(3), pp. 373-383. 2015年3月. 査読有.
- ③ Takayuki Akamine, Mohamed Sofian Abu Talip, Yasunori Osana, Naoyuki Fujita and Hideharu Amano: “Reconfigurable Out-of-Order System for Fluid Dynamics Computation Using Unstructured Mesh.” IEICE Transaction on Information and Systems, Vol. E97-D(5), pp. 1225-1234. 2014年5月. 査読有. doi 10.1587/transinf.E97.D.1225
- ④ Mohamed Sofian Abu Talip, Takayuki Akamine, Mao Hatto, Yasunori Osana, Naoyuki Fujita and Hideharu Amano: “Adaptive Flux Calculation Scheme in Advection Term Computation Using Partial Reconfiguration.” International Journal of Networking and Computing, Vol. 3(2), pp. 289-306. 2013年7月. 査読有.  
<http://www.ijnc.org/index.php/ijnc/article/view/68/67>

[学会発表] (計 27 件)

- ① Naru Sugimoto: “Zynq Cluster for CFD Parametric Survey.” 2016

International Symposium on Applied Reconfigurable Computing, 2016年3月23日. Rio de Janeiro, Brazil.

- ② 酒井諒太郎: “CPU-FPGA 密結合アーキテクチャを用いた Full-PIC 法の高速度の検討.” 情報処理学会全国大会, 2016年3月11日. 神奈川県横浜市.
- ③ 喜屋武克樹: “FLOPS-2D の低レベル通信制御機構の実装と性能評価.” 電子情報通信学会リコンフィギャラブルシステム研究会, 2015年1月30日. 神奈川県横浜市.
- ④ Dipikarini Mishra: “FPGA implementation of Viscous Function in a package for Computational Fluid Dynamics.” The 2<sup>nd</sup> International Symposium on Computing and Networking, 2014年12月11日. 静岡県静岡市.
- ⑤ 平井裕介: “マルチFPGAシステムにおける演算パイプラインの自動設計の実現に向けた研究.” 電子情報通信学会リコンフィギャラブルシステム研究会, 2014年11月26日. 大分県別府市.
- ⑥ 山下貴大: “Fortranによる高性能計算のハードウェア化と高位合成ツール FortRock.” 組み込みシステムシンポジウム 2014, 2014年10月24日. 東京都渋谷区.
- ⑦ Mohamed Sofian Abu Talip: “Partially Reconfigurable Flux Calculation Scheme in Advection Term Computation.” The 2013 International Conference on Field-Programmable Technology, 2013年12月10日. 京都市京都市.

#### 6. 研究組織

##### (1) 研究代表者

長名保範 (OSANA, Yasunori)  
琉球大学・工学部・助教  
研究者番号: 00532657

##### (2) 研究分担者

天野英晴 (AMANO, Hideharu)  
慶應義塾大学・理工学部・教授  
研究者番号: 60175932

藤田直行 (FUJITA, Naoyuki)  
宇宙航空研究開発機構・航空本部・  
主幹研究員  
研究者番号: 70358480

中條拓伯 (NAKAJO, Hironori)  
東京農工大学大学院・工学研究院・准教授  
研究者番号: 80217736