

令和元年6月19日現在

機関番号：18001

研究種目：基盤研究(C) (一般)

研究期間：2016～2018

課題番号：16K00078

研究課題名(和文) CPU-FPGA混在クラスタによるCFD向け高性能計算システムの研究

研究課題名(英文) Development of a high-performance CPU-FPGA hybrid computing system

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) 3,500,000円

研究成果の概要(和文)：計算機システムへのFPGAの導入は電力対性能比面を改善する手段として有望視されているものの、プログラミングがソフトウェアの実装ではなく回路設計となる点をはじめ、本来の性能を発揮するためにはいくつかの問題が未解決である。本研究プロジェクトではこの点を解決するためのプログラミング手法と基盤システムを開発した。これにより、C++やJavaによるアルゴリズム記述で、多数のCPUとFPGAから構成される大規模な並列システムの実現が可能である。

研究成果の学術的意義や社会的意義

高性能コンピュータの電力あたりの計算能力を改善するために、FPGAと呼ばれる論理回路を書き換え可能なLSIを導入することが世界中で検討されているが、プログラミングの困難さやツールの制約によってその潜在的な性能を十分に発揮するには至っていない。本研究プロジェクトではFPGAを用いた大規模計算システムを構築するための基本的なソフトウェア・ハードウェアの基盤と、その上でアプリケーションを開発するためのプログラミングツールを開発した。これによりFPGAによる大規模並列システムとその上で動作する高性能なアプリケーションの開発が容易になり、機械学習や大規模データ処理・科学技術計算などに広く応用可能である。

研究成果の概要(英文)：FPGAs are drawing attentions as a hopeful solution to make watt per performance of high-performance computing systems. However, there is still large barriers to use FPGAs as a general-purpose computing device: for example, require circuit design is required instead of software programming. To mitigate this situation, we have developed a software/hardware platform for FPGA cluster with HLS-based programming tools. With these resources, parallel CPU-FPGA cluster can be programmed by C++ or Java algorithm descriptions.

研究分野：計算機アーキテクチャ

キーワード：FPGA 高位合成 協調設計 科学技術計算

1. 研究開始当初の背景

データセンターやスーパーコンピュータなどの大規模計算システムにおいて、消費電力の壁が顕在化するにつれ、FPGA によるアクセラレーションが注目されはじめています。FPGA によるアプリケーションの高速化についてはすでにデータセンター向けや科学技術計算など、多岐にわたる分野で研究されており、GPU の台頭でいったんは下火となったものの、ふたたび注目を集めています。近年では従来からの HDL 記述と EDA ツールによる回路設計によらず、OpenCL などを用いてソフトウェア的アプローチでアクセラレーションを実現するための開発ツールも市販されはじめています。しかし、これらはいずれも単一の CPU ノードに接続される単一の FPGA をターゲットとしたもので、GPU と同じようなシステムモデルを想定したものである。

しかし、FPGA による計算の高速化は主に、問題を直接ハードウェアで解く深いパイプライン処理から得られるものであることが広く知られている。そこで、本研究グループをはじめとしてさまざまな研究グループが多数の FPGA を高速シリアルリンクで接続し、FPGA にまたがる大規模なパイプラインにより大幅な高速化を実現する可能性について研究を行ってきた。これらのマルチ FPGA システムは深いパイプライン処理によりメモリアクセスを削減し、高い計算スループットを出すことができるが、そのパイプラインを常時稼働させ続けるだけの十分なデータ供給を行うシステムの構成と、大規模なパイプラインの設計を容易化するための設計ツールの欠如が問題であった。

2. 研究の目的

前述のような背景から、本研究課題では図 1 のように多数の CPU と FPGA から構成されるヘテロジニアスな並列システムをターゲットとして、以下のような問題を解決することを研究の目的とした。

- A) CPU と FPGA を一対一で利用するのではなく、システム内の FPGA 上に構築された演算パイプラインを、複数の CPU ノードにまたがって並列化されたソフトウェアから同時に共有する仕組みをハードウェア・ソフトウェアの両面から実現する。
- B) 既存の高位合成ツールも活用しつつ、FPGA のパイプライン設計およびそれを利用するためのソフトウェアの開発を容易化するためのハードウェア・ソフトウェア協調開発環境を実現する。これには、OpenCL のようなアクセラレータ向けの言語を使わずに、一般的なソフトウェア開発用の言語とその文法の範囲内でのアクセラレータ開発も含まれる。
- C) マルチ FPGA でのアクセラレータ設計時に設計上の大きな負担となる回路の分割を全自動、あるいは半自動で実現する。

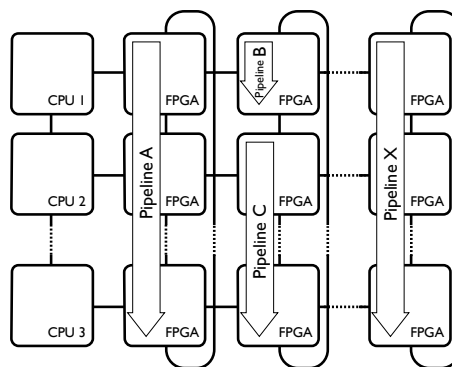


図 1: CPU-FPGA クラスタ

これにより実現されるシステムはその構成上、計算センターのような共有環境での運用にも適しており、近い将来の HPC を支える基盤技術としての FPGA アクセラレーションの実用化にも貢献することができる。

3. 研究の方法

まず、CPU-FPGA クラスタの基盤ソフトウェアとハードウェアを研究代表の長名(琉球大)らのグループで開発した。これは以前の科研費による研究課題から継続した研究であり、本研究課題では主に複数の CPU ノードで FPGA を共有するために必要となる制御機構の拡張や、さまざまな FPGA ボードに対応するための抽象化レイヤの強化などが主体である。これらを整備することで、研究グループ内で運用している概念検証用のデモシステム以外のハードウェアを用いた場合にも CPU-FPGA クラスタを容易に開発することが可能になり、研究成果を広く利用することが可能になる。

高位合成技術に基づくハードウェア・ソフトウェア協調設計の手法としてはまず、商用の C++ ベースの高位合成ツールをもとに、速やかに実用化レベルに達することを目指した開発ツールを長名らのグループが開発した。また、研究分担者である中條(東京農工大)らのグループでは Java スレッドによって並列性を明示的に強力に記述することが可能な高位合成ツールの開発や回路の自動分割手法についても研究を行った。

システムの性能を評価するためのアプリケーションとしては研究分担者である天野(慶應義塾大)・藤田(JAXA)らのグループで宇宙船むげ電気推進エンジン(ホールスラスタ)のシミュレータの高速化についての検討と実装を行った。また、より大規模に並列できる計算問題の事例として、長名・中條らのグループでは 2次元拡散方程式ソルバの実装を行った。

4. 研究成果

(1) ハードウェア・ソフトウェア基盤の開発

本研究で開発する一連のハードウェア・ソフトウェアの動作検証および性能評価を行うプラットフォームとして、図 2 に示すようなデモシステムを構築した。このシステムの構成を図示したものが図 3 であり、2つの CPU ノードにはそれぞれ PCI Express によって FPGA が接続されており、さらに 2つの独立した FPGA ノードが設置されている。FPGA ノード間は同軸ケーブルや SFP+ケーブルによってリング状のネットワークを構成している。

FPGA 内部の構成は図 4 のようになっており、ルータを中心に FPGA 間のシリアルリンクによるデータ転送を行う Aurora コアや PCI Express によるホストとの通信を行う RIFFA/Xillybus (引用文献①, ②) の DMA コアと、アプリケーションの高速化を行うストリーム計算パイプラインの PE が配置される。シリアルリンクや PCI Express から到着したデータストリームはルータによってストリーム PE に送られ、その出力はルータによってさらに別の FPGA やホストに転送される。DMA や Aurora とのインタフェースは、それぞれ専用のアダプタモジュールを介することで共通のポートインタフェースとなっており、これにより異なる世代の FPGA や、あるいは図 2・3 のシステムには含まれていない Intel 社製の FPGA でも同様に動作する。

DMA コアを変更した場合でもホストからは共通の API で FPGA を操作することのできるライブラリも開発し、これによりボードや DMA エンジンの違いを意識せずにホスト側のソフトウェアを記述することができるようになった(学会発表②)。また、FPGA 上のストリーム PE はシステムの稼働中に部分再構成によって動的に書き換えることが可能(学会発表⑦)であり、これも API ライブラリによってシステム内の任意の FPGA のストリーム PE を容易に書き換えることができる。

以上のように汎用的に使える CPU-FPGA クラスタあるいはマルチ FPGA システムのフレームワークは、いまのところ研究・商用ともに公開されておらず、今後これをさらに発展させ、より汎用的に使える形でオープンソースとして公開することも検討している。

(2) ハードウェア・ソフトウェア協調設計環境の開発

本研究課題では、新しい言語を修得することなく、通常の C++や Java の文法の範囲内で CPU と FPGA の双方の動作を記述し、アプリケーションを実行させることを目標とした。

高位合成ツールを用いることで、C++や Java によるアルゴリズム記述に基づくハードウェア設計を行うことはすでに可能であり広く用いられている。しかし、これらのツールを用いる場合には開発・デバッグ中に用いるテストベンチと、実際のアプリケーション実行時にホストで実行するプログラムは異なるのが普通で、後者には PCI Express などを經由して FPGA とやりとりするためのコードが含まれるためである。これらを同一のコードで実現すれば開発の労力を低減することが可能であるとともに、バグが混入するリスクも抑えることが可能になる。

そこで、Xilinx の高位合成ツールである Vivado HLS の C++ライブラリに用意されているストリーム通信のクラスである `hls::stream` を基に、ホストプログラムと FPGA 上のストリーム PE とのやりとりを行うための C++ API ライブラリを開発した (学会発表③。) `hls::stream` クラスは通常、Vivado HLS の C++デザイン内で FIFO やモジュール間のインタフェースとして用いられるが、これをホストとの通信にも使えるように拡張することによって、テストベンチとホストプログラムの差異を大幅に小さくすることが可能であり、さらにテストベンチを自動的にホストプログラムに変換する簡単なプリプロセッサも開発し、実際に性能評価で使用した。

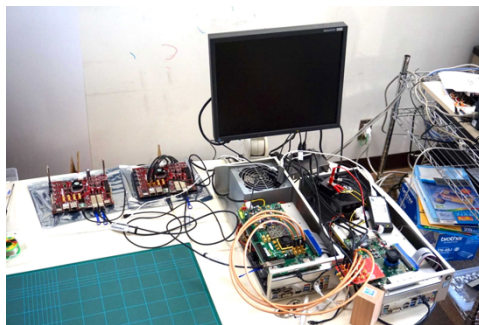


図 2: 検証用デモシステムの外観

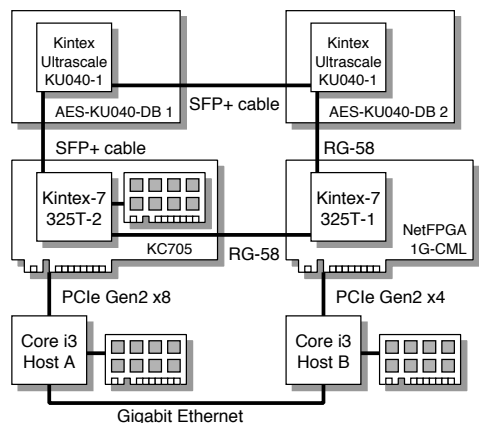


図 3: 検証用デモシステムの構成

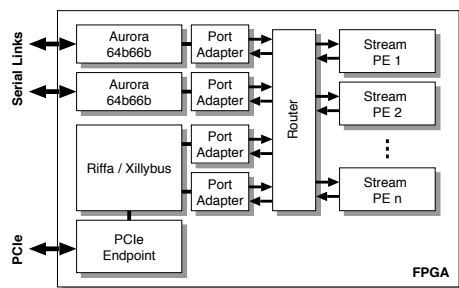


図 4: FPGA 内部の構成

```
int main(){
    hls::stream<uint64_t>
    a, x;

    a.write(...);
    ...

    fpgal(a, x);

    x.read(...);
    ...
}
```

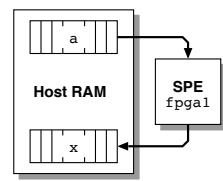


図 5: C++によるホストのコード例

またこれとは別に、Java からのデザインエントリを行う高位合成ツールである JavaRock Thrash の開発も行った。このツールは Java スレッドを用いて回路の並列性を明示的に記述することが可能な点が特徴であるとともに、回路の自動分割の機能（雑誌論文②）などマルチ FPGA システム向けの機能も備えている。

(3) 回路自動分割手法の開発

前述の JavaRock Thrash による回路分割とは別に、Vivado HLS を用いたデザインにおける自動分割についても検討・実装を行った(学会発表④, ⑧。)この手法では高位合成後に得られる HDL を対象にしているため、Vivado HLS に限らず他の高位合成ツールの出力や HDL を用いて記述されたデザインを対象にすることも可能である。

(4) 性能評価

以上のようなハードウェア・ソフトウェアが正しく動作することとその性能を検証するため、図 1 のデモシステムを用いて 2 次元拡散方程式を解くハードウェアを実装した(学会発表②, ⑥。)このモデルは、2 次元の正方形格子上で拡散方程式を解き、熱の移動や水中でのインクの拡散などを計算するものである。図 6 は計算の一例で、初期状態では四角い領域にだけインクが存在しており、時間の経過とともに拡散する様子が確認できる。

FPGA での実装は Vivado HLS によって行われ、メモリに格納されたある時刻 t における各セルの値(濃度または温度)を順次読み出して演算パイプライン(PE)に送り、次の時刻 $t+\Delta t$ における値を得るストリーム計算となっている。この演算パイプライン(PE)を直列に n 段接続すると、1 回のメモリアクセスで時刻 t から $t+n\Delta t$ までを一気に計算することができる。このときの計算スループットを測定した結果が図 7 で、ほぼ PE の数に比例して性能が向上していることが確認できる。4 つの FPGA の消費電力を合計すると 20W に満たないが、同様の計算を Intel Core i7-7700K で行った場合実効性能は 10GFLOPS 程度であり、この CPU の消費電力が 90W 程度であることを考えると、計算のスループットと消費電力の双方で大きく有利である。

また、同様の計算を Java で記述し、JavaRock Thrash で高位合成して性能評価も行った。JavaRock Thrash はいまのところストリーム計算には対応しておらず、1 反復ごとにメモリの読み書きを行う必要がある。また、同じ理由によって Vivado HLS による実装のようなパイプラインの直列接続による時間方向の並列化も困難である。しかし、Vivado HLS での実装と異なり Java スレッドによる空間方向の並列化が強力であり、Vivado HLS では 1PE で 0.87GFLOPS 程度であるのに対して 1.3GFLOPS 程度の性能が得られた。システム全体での性能はパイプラインの直列接続が可能な Vivado HLS 版には遠く及ばないが、今後 JavaRock Thrash のストリーム計算対応などを行うことで、高位合成によるより強力な PE の実装も可能である。

より複雑なアプリケーションの FPGA 実装として、宇宙船の電気推進に用いられるホールスラストのシミュレータの実装を試みた(学会発表①。)これは実際にスラストの研究に用いられているシミュレータをもとにした実装で、実装開始時点では上記の CPU-FPGA クラスタおよびそのための Vivado HLS や JavaRock Thrash を用いた開発環境の用意が間に合わなかったため、ARM プロセッサと FPGA を同一チップ上に搭載した Arria 10 SoC と Intel OpenCL SDK for FPGA を用いた。その結果、実行時間は Tesla K20c の 1/2 程度、Xeon E5-2680v2 (10 コア) の 5 倍程度、消費エネルギーは Xeon プロセッサの 90%弱となり、実行時間では劣るもののエネルギー効率に優れていることが実証できた。Arria 10 SoC ではメモリバンド幅が限られているため、この成果をより強力なメモリシステムを持つ FPGA ボードに移植すればさらなる性能向上も可能と考えられる。

〔引用文献〕

- ① Xillybus Ltd., “Xillybus IP core product brief.” http://xillybus.com/downloads/xillybus_product_brief.pdf, 2016.
- ② M. Jacobsen, D. Richmond, M. Hogains, and R. Kastner, “RIFFA 2.1: A reusable integration framework for FPGA accelerators,” ACM Transactions on Reconfigurable Technology and Systems (TRETs), vol.8, no.4, Sep. 2015.

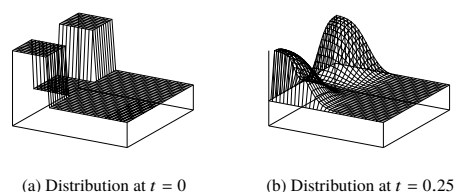


図 6: 2 次元拡散方程式の計算例

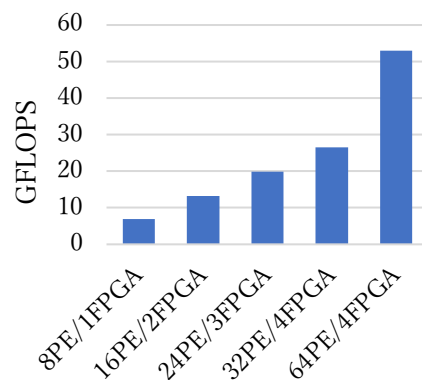


図 7: 拡散方程式ソルバの性能

5. 主な発表論文等

〔雑誌論文〕(計6件)

- ① 矢内奎太朗, 長名保範, 中條拓伯: "Java 言語ベース高位合成ツールによる高性能計算の実機による検証." 情報処理学会論文誌, Vol. 60 (採録決定), 2019年(査読あり)
- ② 松田和也, 三好健文, 竹本正志, 船田悟史, 中條拓伯: "回路分割機能付き Java 言語ベース高位合成ツールにおける回路検証機構." 情報処理学会論文誌, Vol. 57, pp.1680-1689, 2016. (査読あり)

〔学会発表〕(計33件)

- ① Hiroyuki Noda, Manfred Orsztynowicz, Kensuke Iizuka, Takaaki Miyajima, Naoyuki Fujita and Hideharu Amano: "An ARM-based heterogeneous FPGA accelerator for Hall thruster simulation." The 2019 International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies, 2019年6月
- ② Yasunori Osana: "A CPU-FPGA hybrid cluster platform for accelerated scientific computing." CANREXI workshop in CANDAR'18, 2018年11月
- ③ 長名保範, 高見昌志: "Vivado HLS と hls::stream によるマルチFPGAプログラミング手法の提案." 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, 2018年9月
- ④ 村田義雄, 中條拓伯: "スケーラブル・ハードウェア機構におけるストリーム計算のための回路分割手法とFPGAによる検証." 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, 2018年1月
- ⑤ 矢内奎太朗, 長名保範, 中條拓伯: "Java ベース高位合成におけるマルチスレッド機能によるステンシル計算のFPGA実装.", 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, 2018年1月
- ⑥ 長名保範, 坂本洋平: "CPU-FPGA クラスタ上でのストリーミング計算による拡散方程式の計算", 第31回数値流体力学シンポジウム, 2017年12月
- ⑦ 坂本洋平, 前原秋水, 長名保範: "CPU-FPGA クラスタのデータ転送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, 2017年11月
- ⑧ 村田義雄, 吉内大成, 中條拓伯: "スケーラブル・ハードウェア機構における信号圧縮による分割回路効率実行." 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, 2016年5月

6. 研究組織

(1) 研究分担者

研究分担者氏名：天野英晴
ローマ字氏名：AMANO, Hideharu
所属研究機関名：慶應義塾大学
部局名：理工学部
職名：教授
研究者番号(8桁)：60175932

研究分担者氏名：藤田直行
ローマ字氏名：FUJITA, Naoyuki
所属研究機関名：宇宙航空研究開発機構
部局名：航空本部
職名：主幹研究員
研究者番号(8桁)：70358480

研究分担者氏名：中條拓伯
ローマ字氏名：NAKAJO, Hironori
所属研究機関名：東京農工大学
部局名：大学院工学研究院
職名：准教授
研究者番号(8桁)：80217736

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。