様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

# 科学研究費助成事業　　研究成果報告書

平成　３０　年　　６　月　１８　日現在

機関番号：　１５４０１
研究種目：　挑戦的萌芽研究
研究期間：　2016～2017
課題番号：　１６Ｋ１２４１３

研究課題名（和文）ヘテロ並列計算システムによる低消費電力・超線形加速計算とそのための開発環境

研究課題名（英文）Low-power and super-linear speed-up computation by heterogeneous parallel computing systems with development environment

研究代表者

中野　浩嗣（Nakano, Koji）

広島大学・工学研究科・教授

研究者番号：３０２８１０７５

交付決定額（研究期間全体）：（直接経費）　　2,600,000円

研究成果の概要（和文）：FDFM(Few DSP block Few Memory block)アプローチにより，さまざまな処理がFPGAで効率よく処理できることを示した．具体的には，多数のビット数の多い整数にペアに対して，ユークリッド互除法により，高速に最大公約数を求めることができ，RSA暗号のハッキングに適用できることを示した．また，FDFMアプローチにより多倍長演算をサポートするツールを作成した．さらには，C言語プログラムを自動並列化しGPU上で動作するプログラムに変換するツールを作成した．また，Photomosaicを高速に生成するGPUとCPUのハイブリッドアルゴリズムを考案し，実装・評価実験を行った．

研究成果の概要（英文）：We have shown that various computation can be processed efficiently by the FDFM(Few DSP block Few Memory block) approach on the FPGA. More specifically, we have shown that, for a lot of pairs of large integers, the GCD of them can be computed very efficiently on the FPGA, and the RSA encryption can be hacked. Also, multiple-length-integer supporting tools based on the FDFM approach have been developed. Further, we have developed a tool that we call C2CU, which automatically parallelizes a C language program and generates a CUDA C parallel program running on the GPU. We have also presented a CPU-GPU hybrid algorithm for fast photomosaic generation.

研究分野：　情報工学

キーワード：　FPGA　GPU　並列処理

１．研究開始当初の背景

並列処理による高速計算を行うには、メニーコア CPU（例えば、インテル社の Xeon）GPU(例えば､Nvidia 社の GeForce や Tesla)、FPGA（例えば、Xilinx 社の Virtex-7）などのデバイスを用いる方法がある．並列処理の研究の歴史は１９８０年代に始まり、並列アルゴリズム理論の研究や nCUBE やコネクションマシンなどの超並列計算機の開発が盛んに行われていた．研究代表者は大学院生であった１９８０年代後半から並列アルゴリズムの研究を一貫して行ってきた．１９９０年代半ばに GPU や FPGA が開発・販売されるようになり、それらを利用して高速計算を行うための研究が注目されはじめ、現在では多くの研究者がさまざまな観点から研究を行っている．研究代表者も２０００年から FPGA を用いた汎用高速計算の研究を始めた．また、GPU のための開発環境が提供されるようになった２０００年代後半から GPU を用いた汎用計算技術 GPGPU(General-purpose computing on GPUs）の研究を開始し、国内外の研究者と研究成果を競っている．さらに FPGA については、FPGA を用いた汎用計算技術を確立した．具体的には、我々が提案した FDFM(Few DSP blocks Few Memory blocks)アプローチを用いることにより、計算処理によってはメニーコア CPU や GPU を遥かに凌駕する性能を FPGA を用いて低い消費電力で得られることもあることがわかった．

２．研究の目的

比較的安価に並列処理による高速計算を行うには、メニーコア CPU，GPU(グラフィクス処理用のプロセッサ)，FPGA(書き換え可能な集積回路)を用いる方法がある．それぞれが得意な処理、不得意な処理があり、さまざまなタイプの処理を行わなければならない複雑な計算では、単一種類だけを用いた並列処理では十分な性能が得られないことが多い．本研究の目的は、さまざまなデバイスを最適に利用することにより、単一種デバイスによる並列処理の理論限界を超える性能を可能とする並列計算手法を提案する．特に，FPGA の FDFM アプローチを用いた手法について，GPU やマルチコアプロセッサを用いた場合よりも格段に優れた性能を得られることを実証する．

３．研究の方法

OpenCL（Open Computing Language)は、メニーコア CPU、GPU、FPGA などの計算資源を利用して、並列コンピューティングを行うためのフレームワークであり、オープン標準である．C 言語をベースにしたプログラミング言語をサポートしているが、基本的に抽象的な記述しかできないため、特に FPGA では高い性能を得ることができない．この OpenCL のフレームワークをベース利用して、３つのデバイス間の協調・並列計算が効率よく行える開発環境のプロトタイプを試作する．特に FPGA については、我々が提案した FDFM（Few DSP blocks Few Memory blocks)アプローチを容易に実装できるようにする．最新の FPGA は、DSP ブロック(プログラマブル演算器)とメモリブロック（小容量の高速デュアルポートメモリ）を 2000 個以上搭載している．FDFM アプローチは、少数の DSP ブロックとメモリブロックにより特定用途にカスタマイズしたプロセッサコアを構築するものである．この FDFM プロセッサコアを大量に並べ、並列に動作させることにより、メニーコア CPU や GPU を上回る性能を実現することができる．

４．研究成果

(1) FDFM(Few DSP blocks Few Memory blocks)とは，FPGA が内蔵する DSP ブロックとメモリブロックを用いて小さなプロセッサコアを作成し，これを大量にならべて究極の高速化を低消費電力で実現する方法である．DSP は乗算器や加算器を備えた ALU（Arithmetic Logic Unit）であり，その機能をユーザーが切り替えることができる．また内部にパイプラインレジスタを備えており，動作周波数を上げるためのパイプライン化に用いることができる．メモリブロックは，36kbit の小容量のデュアルポートメモリである．独立に読み書き可能な２つのポートをもち，また２個の 18kbit のメモリとして分割使用することもできる．

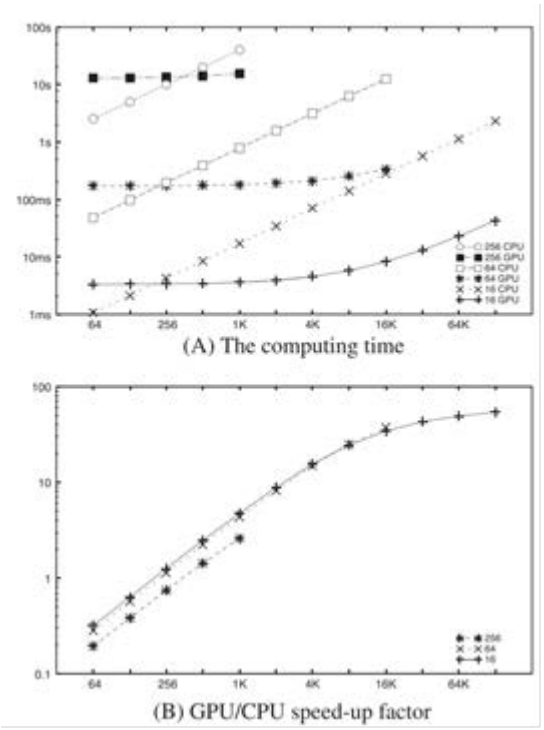
幅広く用いられている暗号方式に RSA 暗号がある．大きな２つの素数 $p$ と $q$ の積 $n=pq$ が公開の暗号化キーとして用いられるが，その $n$ を素因数分解してもとの $p$ と $q$ が求められると，復号化キーを生成することができ，暗号文を復号することができてしまう．暗号化キー $n$ が与えられた時に，それを $p$ と $q$ に素因数分解する処理に膨大な計算時間が必要となり極めて困難であることが RSA 暗号の解読の困難さの根拠になっている．しかし，暗号化キーが不適切に生成された場合，具体的には，２つの暗号化キー $n=pq$ と $n'=p'q'$ が素数を共有し，$p=p'$ が成り立つと，$n$ と $n'$ の最大公約数を求めることにより，素因数分解ができてしまう．最大公約数はユークリッドの互除法で桁数が大きくても高速に求めることができる．そこで，インターネットなどに公開されている大量の暗号化キーを収集し，そのすべてのペアについて最大公約数を求めることにより，不適切に生成された暗号化キーを探し出すことができ，そのような暗号化キーに対する復号化キーを求めることができ，暗号を破ることが可能となる．ブロックメモリに格納された複数の暗号化キーに対してユークリッドの互除法により最小公倍数を求める FDFM アプローチによるプロセッサコア（GCD プロセッサと呼ぶ）を設計し FPGA に実装した．具体的にはユークリッドの互除法をハードウェア実行しやす

いように修正したハードウェア２値ユークリッドアルゴリズムを提案し，それを実行するGCDプロセッサを１個のDSPブロックと１個の 18kbit ブロックメモリを用いて設計した．そのような GCD プロセッサは１つの FPGA（ザイリンクス社 XC7VX485T-2）に 1280 個実装することができ，1280 並列で動作させた結果, 1024 ビットの最大公約数計算が 1 つあたり 0.0904μs で行うことができた．これは既知で最速の GPU を用いた最大公約数計算に比べて 3.8 倍高速であり，また，CPU を用いたものに比べて 316 倍高速であった．

(2) FDFM アプローチを用いて，多倍長演算をサポートするプロセッサコアを FPGA に実装するツールを開発した．多倍長演算（例えば，1024 ビットの乗算など）は暗号化処理などで頻繁に用いられる重要な演算であるが，従来の CPU では直接サポートされておらず，ユーザーの責任で実装する必要があった．高速化のためには多倍長演算をハードウェア化し FPGA で実行する必要があるが，かなりの回路設計の知識と経験が必要で，かつ設計に要する手間も膨大である．そこで多倍長演算をサポートするプロセッサコアを FDFM アプローチにもとづいて自動生成するツールを開発した．このプロセッサコアは，多倍長の加算, 減算, 乗算, インクリメント，デクリメント，比較，左シフト，右シフトなどの演算と，分岐命令，ストア命令などの基本命令をサポートしている．ユーザーがこれらの命令を用いた多倍長演算を行うプログラムを記述すると，必要な演算・命令だけをサポートする最小のプロセッサコアが自動生成される．そしてそのプロセッサコアを大量に並べ，並列動作させることにより，演算スループットを最大化することができる．多倍長演算では乗算がもっともコストがかかるが, Comba 法[1]を巧妙に DSP ブロックに実装することにより，たった 2 個の DSP ブロックと 1 個のメモリブロックでプロセッサコアを実装することができた．

性能評価のため, RSA 暗号の暗号化処理を実装した．その結果，１個の FPGA（ザイリンクス社の XC6VLX249T-1）に 306 個のコアを実装することに成功した．1 個のプロセッサコアで 2048 ビット暗号化処理が 792ms で行えるので，306 個では，１回の 2048 ビット暗号化処理が 2.59ms で行えるスループットが実現できたことになる．これは，CPU による同じ処理より 2.19 倍高速である．

(3) 逐次処理を行う C 言語プログラムが与えられた時に，大量の入力に対して，その C 言語プログラムを同時実行する CUDA C プログラムを自動生成するツールを開発した．CUDA C は，NVIDIA 社が開発した並列言語・開発環境で，同社の GPU で動作させることができる．我々が開発したツールは１つの入力に対する C 言語プログラムをユーザーが記述すれば，そのユーザーが CUDA C の知識が全くなくても，自動的に $n$ 個の入力に対して同時処理を行う CUDA C プログラムに変換するものである．バイトニックソート，Floyd-Warshall アルゴリズム，モンゴメリ乗算のアルゴリズムを題材に性能評価を行った．ここで Floyd-Warshall アルゴリズムは，重み付き有向グラフの全ペアの最短経路問題を解くアルゴリズムである[2]．Floyd-Warshall アルゴリズムの C 言語プログラムを開発したツールを使って CUDA C プログラムに変換し，GPU を使って性能評価を行った．下のグラフはその実行時間と CPU(Intel 社の Xeon)と比べた GPU(NVIDIA 社の GeForce GTX TITAN)の加速率を表している．グラフのノード数が 16, 64, 256 の場合の性能評価を行った．同時計算するグラフの個数は 64 から 128K まで変化させた．その結果 128K のグラフ数の場合に，変換後の CUDA C プログラムは 54 倍の加速を達成することができた．




(4)最適な Photomosaic を生成する処理を CPU と GPU を併用することにより高速化した．２枚同じ大きさのグレイスケール画像 A と B が与えられた時に,画像 A をブロックに分割し，それを並び替えることにより画像 B に近い画像を生成するのが Photomosaic である．もっとも画像 B に近くなるような画像 A のブロックの並び替えを求めるのは容易でない．我々は

Step 1: 画像 A のブロックと画像 B のブロック間のすべての組み合わせについて類似度を求める．

Step 2:Step 1 の最適なブロック間の１対１

対応を求める．つまり，ブロックをノード，辺を類似度としたときに２部グラフの最小マッチングを求める．
という２つのステップで最適 Photomosaic を得られることを示した．Step 1 は並列化可能で GPU で高速に計算できるが，Step 2 の並列化は困難であり，GPU で高速に求めることができない．一方，CPU は Step 2 を高速に求めることができるが，Step 1 の処理は GPU よりかなり遅い．そこで，Step 1 を GPU で処理し，Step 2 を CPU で処理する方法を考案した．さらには，Step 2 を近似最小マッチングを求める並列手法を考案し GPU に実装した．その結果，2048×2048 ピクセルの画像に対して，ブロックサイズが 64×64 のときに，Step 1 と Step 2 ともに CPU（Core i7-3770）を用いて最適解を求めた場合は 36.8 秒，Step 1 に GPU（NVIDIA 社 Tesla K40），Step 2 に CPU を用いた場合は，16.2 秒となり，約 2.3 倍の高速化を達成できた．さらに近似解の場合，Step 1 と Step 2 の両方に CPU を用いた場合は，21.1 秒，両方に GPU を用いた場合は 0.386 秒で，54.7 倍の高速化が達成できた．目視において，最適解と近似解の Photomosaic の差は見られず，GPU を用いて近似解を求めるのが実用的であると言える．

参考文献

[1] P.G.Comba, Exponentiation cryptosystems on the IBM PC, IBM Systems Journal, Vol. 29, No. 4, pp. 526 - 538, 1990.
[2] Robert W. Floyd. Algorithm 97: shortest path. Communications of the ACM. 1962;5(6):345.

５．主な発表論文等

〔雑誌論文〕（計 5 件）

① Toru Fujita, Koji Nakano, Yasuaki Ito, Daisuke Takafuji, An Efficient GPU Implementation of CKY Parsing Using the Bitwise Parallel Bulk Computation Technique, IEICE Transactions on Information and Systems, Vol. E100-D, pp. 2857-2865, 2017, 査読あり
DOI: 10.1587/transinf.2017PAP0018

② Daisuke Takafuji, Koji Nakano, Yasuaki Ito, Jacir Luiz Bordim, C2CU: a CUDA C program generator for bulk execution of a sequential algorithm, Vol. 29, Concurrency and Computation: Practice and Experience, e4022, 2017, 査読あり
DOI: 10.1002/cpe.4022

③ Tatsuya Kawamoto, Xin Zhou, Jacir L. Bordim, Yasuaki Ito, and Koji Nakano, An FPGA implementation for a flexible-length-arithmetic processor employing the FDFM processor core approach, IEICE Transactions on Information and Systems, Vol. E99-D, pp. 2901-2910, 2016, 査読あり
DOI: 10.1587/transinf.2016PAP0029

④ Xin Zhou, Koji Nakano, Yasuaki Ito, Efficient Implementation of FDFM Approach for Euclidean Algorithms on the FPGA, International Journal of Networking and Computing, Vol. 6, pp. 420-435, 2016, 査読あり
DOI: 10.15803/ijnc.6.2_420

⑤ Yi Yang, Yasuaki Ito, Koji Nakano, Photomosaic Generation by Rearranging Divided Images, Bulletin of Networking, Computing, Systems, and Software, Vol. 6, pp. 22-27, 2016, 査読なし

〔学会発表〕（計 2 件）

①Takuma Wada, Shunji Funasaka, Koji Nakano, and Yasuaki Ito, A Hybrid Architecture for the Approximate String Matching on an FPGA, International Symposium on Computing and Networking, 2017

② Yi Yang, Yasuaki Ito, Koji Nakano, Photomosaic Generation by Rearranging Subimages, with GPU Acceleration, International Symposium on Parallel and Distributed Processing Systems Workshops (IPDPS-APDCM), 2017

〔図書〕（計 0 件）
なし

〔産業財産権〕
なし

〔その他〕
なし

６．研究組織

(1)研究代表者
中野　浩嗣（Koji Nakano）
広島大学・工学研究科・教授
研究者番号：30281075

(2)研究分担者
伊藤　靖朗（Yasuaki Ito）
広島大学・工学研究科・准教授
研究者番号： 40397964

(3)研究分担者
高藤　大介（Daisuke Takafuji）
広島大学・工学研究科・助教
研究者番号：00314732