

## スレッドレベル並列性を活かす科学技術計算用FPGAシステムの提案

吉見真聡<sup>†1</sup> 西川由理<sup>†1</sup> 長名保範<sup>†5</sup>  
舟橋啓<sup>†1</sup> 広井賀子<sup>†3</sup> 柴田裕一郎<sup>†2</sup>  
山田英樹<sup>†2</sup> 北野宏明<sup>†4</sup> 天野英晴<sup>†1</sup>

近年、FPGAを科学計算のアクセラレータとして用いる研究が盛んに行われている。ループレベル並列性の高いアルゴリズムでは、汎用マイクロプロセッサの数十倍の高速化を示すが、複雑に挙動が制御されるアルゴリズムが対象の場合、大きな性能向上は見込まれない。それにもかかわらず、FPGAの応用範囲を広げ、アプリケーションごとに専用ハードウェアを開発するコストを削減するニーズが高まるばかりである。我々は、応用範囲を広げる鍵はスレッドレベル並列性を引き出すことだと考え、それを実現するためにモジュラーな構造のハードウェアの組み合わせで構成された計算システムを提案する。本稿では、その構成の有効性を示すため、確率モデル生化学シミュレーションアルゴリズムを題材として、その構成の有効性について議論する。

### Proposal of FPGA-based calculation system exploiting thread level parallelism for scientific applications

MASATO YOSHIMI,<sup>†1</sup> YURI NISHIKAWA,<sup>†1</sup> YASUNORI OSANA,<sup>†5</sup>  
AKIRA FUNAHASHI,<sup>†1</sup> NORIKO HIROI,<sup>†3</sup> YUICHIRO SHIBATA,<sup>†2</sup>  
HIDEKI YAMADA,<sup>†2</sup> HIROAKI KITANO<sup>†4</sup> and HIDEHARU AMANO<sup>†1</sup>

There are many ongoing researches on acceleration of various scientific applications using FPGAs. Dozens of throughput of general-purpose processors can be easily achieved on an FPGA if a target algorithm has high degree of loop- or data-level parallelism. However, it becomes instantly difficult to improve performance with FPGAs at a slight complexity being introduced to the algorithm. Nevertheless, there is an increasing motivation to expand application range of FPGAs due to strong demands of cost reduction for developing application specific hardware. We think that the key for expanding application range is to exploit thread-level parallelism, and we are going to propose a computation system composed of various modular hardware. This paper takes up a recent stochastic biochemical simulation algorithm, which is a fairly complex Monte Carlo Method, and discusses the benefit of the proposed architecture.

#### 1. はじめに

近年の半導体の集積率の向上により、FPGAはプログラマブルブロック数が増大し、メモリやDSPブロックなどのハードマクロの充実も図られるようになった。このため、低コストで運用可能な科学計算に高速実行システムとして、FPGAを利用する研究が盛んに行わ

れている。

これらの研究から、他の専用ハードウェアと同様に、ループレベル並列性を十分に引き出せる場合にはFPGAを用いて汎用マイクロプロセッサの数十倍の高速化が可能であることが知られている。しかし、データの依存性があったり、演算発生頻度やタイミングが変動したりするなど、アルゴリズムが多少複雑になると、計算速度が大きく下落することが多いため、そのような場合には様々な工夫が必要になる。

本稿では、このような高速化しにくいアルゴリズムの例として、著者らが取り組んできた確率モデル生化学シミュレーションを紹介し、スレッドレベル並列性を引き出して高速化する手法を提案する。また、この際に発生する問題点について議論する。

<sup>†1</sup> 慶應義塾大学大学院理工学研究科  
Graduate School of Science and Technology, Keio University  
<sup>†2</sup> 長崎大学工学部情報システム工学科  
Dept. of Computer and Information Sciences, Nagasaki University  
<sup>†3</sup> EMBL-EBI, Wellcome Trust Genome Campus  
<sup>†4</sup> 科学技術振興機構 北野共生システムプロジェクト  
Kitano Symbiotic Systems Project, ERATO-SORST, Japan Science and Technology Agency  
<sup>†5</sup> 成蹊大学 理工学部  
Faculty of Science and Technology, Seikei University

## 2. 確率モデル生化学シミュレーション

著者らは FPGA を用いて確率モデル生化学シミュレーションアルゴリズム (Stochastic Biochemical Simulation Algorithm : 以下 SSA)<sup>1)</sup> のアクセラレータを実装し、その性能評価を行ってきた<sup>2)3)</sup>。

SSA は, Gillespie が提案したモデリング手法を基礎とし, 生化学モデルの分子数の時間変化を乱数を用いて計算するモンテカルロ法である。細胞内で発生する反応では数個だけ存在する反応物が重要な振る舞いを担うことがあるため, SSA はこのような状況のシミュレーションに有用であり, システムバイオロジーの研究において広く使用されている。しかし, 統計的に有意な解析を行うための十分なサンプル数を得るには, 異なる乱数系列を用いた数千回以上のシミュレーション実行を要するため, 膨大な計算時間に対応できる手法の研究が行われている。

高速実行のためのソフトウェア的なアプローチでは, チューニングされた様々なアルゴリズムが開発されている<sup>4)5)</sup>。また一方で, SSA の実行プラットフォームに関する研究も進められており, GPU<sup>6)</sup> や PC クラスタ<sup>7)</sup> などを用いた評価も報告されている。なかでも, FPGA を用いた高速な計算システムの開発は盛んであり, 著者らの他にも複数の研究グループが取り組んでいる<sup>3)8)9)</sup>。

## 3. FPGA を用いた高速化

### 3.1 First Reaction Method アクセラレータ

Gillespie が SSA と併せて提案している First Reaction Method(以下, FRM) は, SSA の中でもっともシンプルなプログラムコードで記述できるアルゴリズムである。浮動小数点数 (以下, FP) の対数の計算や除算など, FPGA に実装すると比較的面積が大きくなる算術演算が実行されるものの, ループ間にデータ依存性が無く, 高いループレベル並列性を持っている。

そのため, FP 演算モジュールを直列に接続し, そのパイプラインに間断なくデータを投入することで, ハードウェアの使用率を最大に利用する構造を取ることができる。また, シミュレーションの中間データや定数テーブルなどは Xilinx 社の FPGA が持つ組み込みメモリ (BlockRAM) に格納し, 反応数が多い生化学モデル (約  $10^3$  反応まで) のシミュレーションに対応する。実装と評価の結果, Xeon 2.80GHz を用いた FRM の実行と比較して, 約 90 倍程度のスループットが得られることを確認した<sup>2)</sup>。

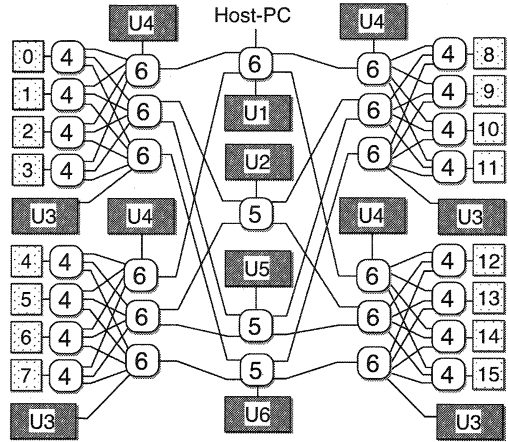
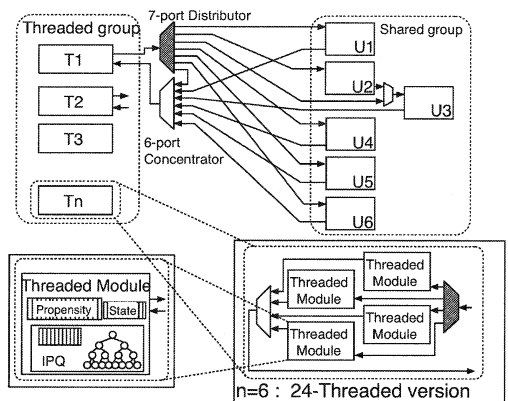


図1 NoC型 NRM 実行システムのブロック図: 16 スレッド並列実行。正方形はスレッドモジュールを示す (内部の数字はスレッドモジュールの番号), 網掛けの四角形は共有モジュール, 角丸の四角形は両者を接続するルータ (内部の数字はポート数)。関数 U3, U4 は頻繁に使用されるので, 各 4 つ配置した



U1 : Tables that store update vector for each reaction  
U2 : Tables that store dependency graph  
U3 : Tables that store reactants to calculate propensity  
U4 : Propensity calculator  
U5 : putative time calculator (calculates Eq.1)  
U6 : putative time modifier (calculates Eq.2)

図2 階層バス型 NRM 実行システムのブロック図: 24 スレッド並列実行例。U1 から U6 は算術演算を行う共有モジュール, 左側の 3 種のメモリを持つモジュールはスレッドモジュールを示す。それらは階層バス構造の接続網で接続される。

### 3.2 Next Reaction Method アクセラレータ

Gibson らが開発した Next Reaction Method(以下, NRM) は, 統計的な等価性を保持しながら FRM の計算量を  $O(n)$  から  $O(\log(n))$  に改善したアルゴリズムである<sup>5)</sup>。

NRM はデータ構造として 2 分木を持っており, 算術演算の回数やタイミングが変動するので, FPGA に

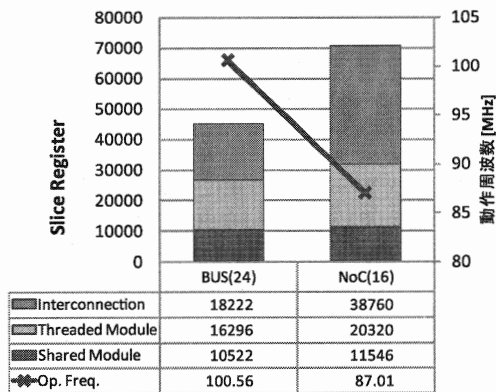


図3 NRM 実行システムのリソース使用量と動作周波数：パス型は Virtex5 XC5VLX110T, NoC 型は Virtex4 XC4VLX200 を対象に Verilog-HDL で実装, ISE-8.2i で合成, 配置配線を行った評価結果。

実装した場合、パイプラインを効率良く利用することは困難である。

著者らは NRM のアルゴリズムを、シミュレーション 1 スレッドの中間状態を保持するスレッドモジュールと、算術演算モジュール (共有モジュールと呼ぶ) とに分割し、複数のスレッドモジュールが、データ転送網を通して共有モジュールを使用する構造で、SSA のスレッドレベル並列性を活かすハードウェア構造を提案した<sup>10)</sup>。このような構造では、データ転送網が性能を決める要因となる<sup>11)</sup> ため、様々な構成が取れるよう、各モジュールを共通の入出力インターフェースを持つモジュラーな実装を行った。著者らは、NRM 実行システムとして図 1 および図 2 に概要を示す 2 種類のデータ転送網を実装して性能評価を行った<sup>11)3)</sup>。

これらのシステムの面積と動作周波数を図 3 に示す。ソフトウェアプログラムとの性能比較を図 4 に示す。図 4 の性能評価から、FPGA を用いた NRM 実行システムは、Core 2 Quad Q6600 上でのソフトウェア実行と比べ、約 4.33 倍から 5.38 倍のスループットが得られることが確認された。

#### 4. マルチスレッド並列計算システムの提案

前章の結果をもとに、より高性能を実現し、アプリケーションを拡大するための方法について検討する。

##### 4.1 スレッドモジュールのマルチスレッド化

FPGA を用いた NRM アクセラレータのスループットは、汎用プロセッサ上の 1 コア実行の約 5 倍と、性能向上としてのインパクトが大きいとは言いがたい。そのため、さらに性能を向上するための手法について検討する。

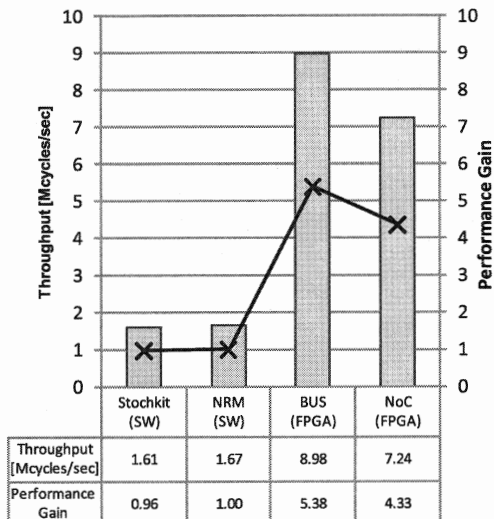


図4 計算性能とソフトウェア実行との性能比較：生化学モデル HSR(反応数 61) を対象に、実装したハードウェアのスループットを RTL シミュレーションで測定。ソフトウェアプログラムは C++ で NRM を実行するプログラムコード (シングルコア実行) を実装し、確率モデル生化学シミュレーションツールキット Stochkit と共に Core 2 Quad Q6600 2.40GHz, 4GB RAM, Linux 2.6.22-14, gcc-4.1.3 (-O3) の環境での実行結果を測定。

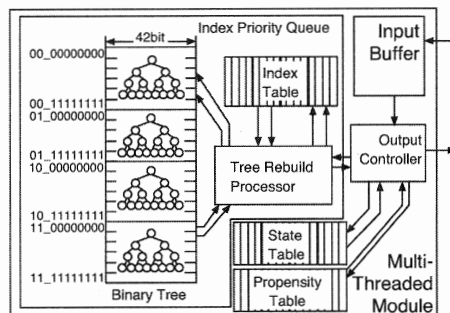


図5 マルチスレッドモジュールの概要：4 スレッドの例。BlockRAM のアドレスの先頭 2 ビットがスレッドを指定する ID として使用される。

NRM はアルゴリズム上の並列性が低いため、性能向上は複数シミュレーションスレッドの並列実行に頼るところが大きい。そこで、並列実行できるスレッド数を増やす方法として、スレッドモジュール内の BlockRAM を分割し、各区画にシミュレーションスレッドのデータを格納して並列実行するマルチスレッドモジュールを提案する。図 5 に、NRM におけるマルチスレッドモジュールの概要を示す。スレッドモジュールのコントローラは、共有モジュールからの計算結果を待つ時間や、2 分木の値を比較時など、BlockRAM へアクセスしない時間が多いことを利用し、1 つのモ



表 1 NRM で HSR の計算を実行した場合の関数コール回数

関数名	実行回数	比率
U1	2,000,000	1.00
U2	2,000,000	1.00
U3	14,870,015	7.44
U4	14,870,015	7.44
U5	3,392,000	1.70
U6	5,515,890	2.76

ジュールで複数のスレッドを実行し、回路の利用効率を高める。例えばこの方法を用いると、HSR(反応数61)ではスレッドモジュールに16スレッド格納できる。この方法により、スループットを向上できると考えられる。

また、利用頻度の高い共有モジュールの通信路の混雑が性能に大きく影響することがわかっている<sup>11)</sup>。HSRを $2 \times 10^6$ 反応サイクル実行した場合に、各関数の呼び出し回数を表1にまとめた。表1は、共有モジュールU3, U4の使用回数が他のモジュールよりも多いことを示している。このような場合、U3, U4の通信路における転送待ち時間を調べ、性能のボトルネックになっていれば、これらの共有モジュールを複数配置することで解消する。また、U3, U4の転送待ち時間が長くない場合は、それ以外のモジュールの通信路は帯域に余裕があるため、それらを多機能共有モジュールとして統合し、通信路のロジック資源を削減することができる。

#### 4.2 データ転送網の面積削減

図2の階層バス型のデータ通信網は、図1のNoC型の通信網から、不要な経路を取り除いた形状と見なすことができる。図3に示すように、階層バス型の実行システムは、NoC型に比べてより多くのスレッドモジュールを持ちながらも通信網部分のロジック資源使用量を約53%削減している。FPGA上にデータ通信網を構成する場合には、データの進行方向に注目し、必要な経路を限定して構成する必要がある。しかし、階層バス型のデータ通信網のロジック資源は全体の約40.46%と依然大きな割合を占めている。

より多くのスレッドモジュールをFPGA上に配置するためには、通信網部分のロジック資源を削減する必要がある。この場合には、機能共有モジュールを使用して共有モジュールの数を減らす方法が考えられる。例えば、U5, U6は使用するFP演算の種類が類似しているため、演算器を共有する方法が挙げられる。

#### 5. おわりに

本稿では、スレッドレベル並列性を持つ科学計算の

アクセラレータをFPGAを用いて実装する方法について、確率モデル生化学シミュレータの実装例を通して検討した。

FPGAを用いたアクセラレータは今後、様々なアプリケーションを対象としていくことが考えられる。複雑な挙動を示すアルゴリズムのFPGA実装では、モジュラーな構成のハードウェアを用いて、ロジック資源の有効利用や性能などについて、多面的な評価を行っていく必要があると言える。

#### 参考文献

- 1) Gillespie, D. T.: Exact Stochastic Simulation of Coupled Chemical Reactions, *J. Phys. Chem.*, Vol.81, No.25, pp.2340–2461 (1977).
- 2) 吉見真聡ほか：FPGAを用いた確率モデル生化学シミュレータ, 情報処理学会論文誌：コンピューティングシステム, Vol.48, No.SIG 3 (ACS 17), pp.45–58 (2007).
- 3) Yoshimi, M. et al.: Practical Implementation of a Network-based Stochastic Biochemical Simulation System on an FPGA, *proc. of the 18th Intl. Conf. on FPL, IEEE*, pp.663–666 (2008).
- 4) Gillespie, D. T.: Stochastic Simulation of Chemical Kinetics, *Annual Review of Physical Chemistry*, Vol.58, pp.35–55 (2007).
- 5) A.Gibson, M. and Bruck, J.: Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels, *J. Phys. Chem. A*, Vol.104, No.9, pp.1876–1889 (2000).
- 6) Li, H. et al.: Efficient Parallelization of Stochastic Simulation Algorithm for Chemically Reacting Systems on the Graphics Processing Unit, *13th SIAM Conference on Parallel Processing for Scientific Computing* (2008).
- 7) Schwehm, M.: Parallel Stochastic Simulation of Whole-Cell Models, *Proceedings of the Second Intl. Conf. on Systems Biology*, pp.333–341 (2001).
- 8) Lok, L.: The need for speed in stochastic simulation, *Nat. Biotechnol.*, Vol. 22, No. 8, pp.964–965 (2004).
- 9) Keane, J.F. et al.: A Compiled Accelerator for Biological Cell Signaling Simulations, *proc. of the 12th Intl. Symp. on FPGA*, pp.233–241 (2004).
- 10) Yoshimi, M. et al.: FPGA Implementation of a data-driven Stochastic Biochemical Simulator with the Next Reaction Method, *proc. of the 17th Intl. Conf. on FPL, IEEE*, pp.254–259 (2007).
- 11) Yoshimi, M. et al.: A Framework for Implementing a Network-Based Stochastic Biochemical Simulator on an FPGA, *proc. of the Intl. Conf. on FPT*, pp. 193–200 (2007).