

データ転送網を用いた確率モデル生化学シミュレータの FPGA への実装の検討

吉見 真聡[†] 長名 保範[†] 岩岡 洋[†] 西川 由理^{††} 小嶋 利紀^{††}
舟橋 啓^{†††} 広井 賀子^{†††} 柴田裕一郎^{††††} 岩永 直樹^{††††} 北野 宏明^{††††}
天野 英晴[†]

[†] 慶應義塾大学大学院理工学研究科
^{††} 慶應義塾大学理工学部 〒 223-8522 横浜市港北区日吉 3-14-1
^{†††} 科学技術振興機構 北野共生システムプロジェクト
〒 150-0001 東京都渋谷区神宮前 6-31-15 マンション 31, 6A
^{††††} 長崎大学工学部情報システム工学科 〒 852-8521 長崎市文教町 1-14
E-mail: †bio@am.ics.keio.ac.jp

あらまし 確率モデル生化学シミュレーションアルゴリズム (SSA) は、生化学システムの厳密なシミュレーションが可能である一方、膨大な計算時間を要求することが知られている。本研究報告では、Next Reaction Method (SSA-NRM) を FPGA 上で高スループットに実行することを目的として、シミュレーションデータ部分 (DU) と計算部分のデータ転送をルータを介して接続する手法に関する予備評価を行った。その結果、単一のシミュレーション速度は汎用プロセッサに劣るものの、中規模な FPGA に 15 個以上の DU を搭載可能であるため、汎用プロセッサの演算能力を超える性能が期待できることがわかった。

キーワード データ接続網, FPGA, 確率モデル生化学シミュレーション

A Stochastic Biochemical Simulator with a Data-transfer Network on an FPGA

Masato YOSHIMI[†], Yasunori OSANA[†], Yow IWAOKA[†], Yuri NISHIKAWA^{††},
Toshinori KOJIMA^{††}, Akira FUNAHASHI^{†††}, Noriko HIROI^{†††}, Yuichiro SHIBATA^{††††},
Naoki IWANAGA^{††††}, Hiroaki KITANO^{†††}, and Hideharu AMANO[†]

[†] Graduate School of Science and Technology, Keio University
^{††} Faculty of Science and Technology, Keio University
Hiyoshi 3-14-1, Yokohama-shi, Kanagawa, 223-8522 JAPAN
^{†††} Kitano Symbiotic Systems Project, ERATO-SORST, Japan Science and Technology Agency
Suite 6A, M31, 6-31-15 Jingumae, Shibuya-ku, Tokyo, 150-0001 JAPAN
^{††††} Dept. of Computer and Information Sciences, Nagasaki University
1-14 Bunkyo-machi, Nagasaki, 852-8521 JAPAN
E-mail: †bio@am.ics.keio.ac.jp

Abstract A high-performance biochemical simulator using a reconfigurable system draws attention as a low-cost solution from biologists. Especially, stochastic simulation algorithms (SSAs) are widely known to be rigorous, but also to be time-consuming for advanced microprocessors due to its algorithm characteristics. In this research report, a prototype of data-transfer network based on SSA-NRM (Next Reaction Method) simulator is designed and evaluated with the most basic configuration. The evaluation result indicates high-throughput capability for large-scale biochemical systems.

Key words Data-transfer network, FPGA, Stochastic biochemical simulation

1. はじめに

確率モデル生化学シミュレーションアルゴリズム (SSA : Stochastic Simulation Algorithm) は、定義された生化学システムの挙動を厳密に計算できる。しかし、モンテカルロ法の一つであるため、1回の実行結果からはシステムの挙動の一般性を得ることはできず、同一のパラメータを用いて数千回以上のシミュレーションを実行し、それらの結果から統計的に挙動を解析しなければならない。また SSA の性質上、反応1回が計算サイクルの単位になるため、1回のシミュレーションにも、膨大な回数演算を繰り返す必要がある。そのため、SSA は一般的に非常に長い計算時間が要求されることが知られており、高速化に対する要求は大きい。

計算時間の問題に対応するため、計算効率が向上した改良型 SSA [1] や近似アルゴリズム [2] が開発中であり、PC/WS クラスタを用いて実行されているが、小規模な研究室でも利用できる安価で高速、構成変更への柔軟な対応が可能なりコンフィギュラブルシステムを利用した確率モデル生化学シミュレータも開発されている [3] [4]。これらはいずれも、高い動作周波数で動作する汎用プロセッサの数倍から数百倍の高速化を実現しており、高い効果が得られている。しかし、アルゴリズムの近似や生化学システムの規模の制約など、多くの課題も残っている。

大規模システムに対応する最も高効率で、近似を行わない SSA として、Next Reaction Method (NRM) が知られている [1]。また、FPGA を用いて NRM を簡略化せずに実行する回路に関する評価も行われている [5]。実行中の状態を保持するデータユニット (DU) を複数持ち、パイプライン化された演算ユニット (FU) に順次アクセスする方法により、高スループットな演算の実現が期待できたが、DU 数が増えたときの回路面積増加率が大きいこと、生化学システムの規模に比例して計算時間が下落することなど、いくつかの問題が存在していた。

本研究報告では、これらの問題に対し適切な解決策を明らかにするために、NRM の挙動を解析して DU-FU の分割を完全に行い、インタフェースを共通にした上で、ユニット間の接続にルータを使用する方法について検討を行った。

2. 確率モデル生化学シミュレーション

2.1 確率モデル生化学システム

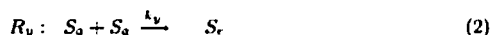
生化学システムは、以下の3項目について定義されたものである。システムの時間を t 、初期時間を t_0 とする。

(1) 分子の種類と初期状態 :

m 種類の分子 S_1, \dots, S_m の初期状態が $\overline{X}(t_0) = \{X_1(t_0), \dots, X_m(t_0)\}$ ($X_i(t)$: 整数) で表される。

(2) 反応の種類と形式 :

n 種類の反応 R_1, \dots, R_n が以下のように表される。各反応ごとに反応速度定数 k_j が設定される。



(3) 状態更新ベクトル :

状態更新ベクトル \vec{a} は、 R_j の要因分子と生成分子のリストで表される。

要因分子 (Reactants) :

反応発生で分子数が減少 (式 1-3 の左辺の分子)

生成分子 (Products) :

反応発生で分子数が増加 (式 1-3 の右辺の分子)

2.2 Propensity Function

Propensity function a_j は、各反応について、分子数と反応速度定数から計算される値である。 R_x (式 1) の a_x は、式 4、 a_y (式 2) は式 5、 a_z (式 3) は式 6 から計算される。

$$a_x = X_a \times X_b \times c_x \quad (4)$$

$$a_y = \frac{X_a \times (X_a - 1)}{2} \times c_y \quad (5)$$

$$a_z = X_a \times c_z \quad (6)$$

各反応のパラメータ c_j は、生化学システムの温度や体積および反応速度定数から設定される値である。 a_j は反応 R_j の Reactants の組合せ (combination) に c_j を乗算した値である。Reactants が 1 個の反応 (R_x) を 1 次反応、2 個の反応 (R_x, R_y) を 2 次反応と呼ぶ。

2.3 確率モデルシミュレーションの概要

SSA は、定義を淡く生化学システムが与えられたときに、以下の手順で分子数の時間変化を求めるアルゴリズムである。

STEP1 乱数と分子状態を用いて以下の2つの値を計算する

- 次に起こる反応の番号 μ
- 反応発生の時間 τ

STEP2 R_μ に従い、分子数を更新する

STEP3 システム内の時間を τ に従って更新する

STEP4 STEP1 に戻る

R_j の発生予測時間 τ_j は、 $[0, 1]$ の範囲の1様乱数 r と、 a_j を用いて、式 7 で求められる。

$$\tau_j = \frac{1}{a_j} \ln \left(\frac{1}{r} \right) \quad (7)$$

複数提案されている SSA は、 μ の決定方法で分類できる。

2.4 Next Reaction Method

2.4.1 NRM の概要

Gillespie が開発した2種類のオリジナル SSA [6] [7] の計算時間は、対象システム内で定義された反応数 N に線形に比例する (計算量 $O(N)$)。Next Reaction Method (NRM) は、Gibson と Bruck による大規模な生化学システムに対応する SSA である [1]。NRM は、Gillespie の First Reaction Method (FRM) [6] に Indexed Priority Queue (IPQ)、Dependency Graph (DG) と呼ばれる2種類のデータ構造を導入し、FRM と統計的に等価でありながら、計算量を $O(\log(N))$ に向上したアルゴリズムである。

NRM は、すべての反応の発生予測時間のうち最小値である反応 (最も直前に起こる反応) を、発生反応 R_μ として選択する。最小値の探索を IPQ を使用して効率化し、 τ_j の修正回数を DG によって最小限に抑えることで、高速な計算が可能になっている。また、計算量が $O(\log(N))$ であることから、大規

模な生化学システムのシミュレーションが可能であると考えられている。

2.4.2 Indexed Priority Queue

IPQは、2分木を構成するメモリと1次元配列のメモリの組で作られるデータ構造である(図1)。2分木は、反応番号とその発生予測時間の組 (j, τ_j) をノードとし、親ノードの τ_j は子ノードの τ_j よりも常に小さいという性質を満たす。1次元配列のメモリは、ノード R_j の2分木上の位置を示すポインタテーブルである。

2分木がその性質を満たす状態であれば、根ノードには全ノード中で最小の τ_j を持つノードが格納されている。また、根ノードの反応番号 j から、発生した反応を決定することができる。

また、特定の反応の発生予測時間は、ポインタテーブルから目的の反応の位置を調べ、2分木から読み出すことで知ることができる。

IPQが持つべき機能は、以下の3つである。

ADD 2分木の末端にノードを追加する。木の親方向のノードと τ_j を比較、交換を行い、木の性質を保つ

UPDATE 2分木の任意の位置のノードの τ_j を更新し、親方向あるいは子方向のノードと τ_j を比較、交換を行い、木の性質を保つ

READ 2分木の任意の位置のノードを読み出す
ノードの交換の際、同時にポインタテーブルの値も交換する。2分木のノード数は生化学システムの反応数 N と等しく、大規模システムではノードの交換時間がシミュレーション時間の中で支配的になるため、計算量が $O(\log(N))$ となる。

2.4.3 Dependency Graph

ある反応が発生して分子数が変化した場合、他の反応の発生予測時間が変化する場合がある。DGは、 R_j が起こった時に、 τ_j が修正される反応を列挙したリストである。例として図2のような反応 $R_1 - R_4$ が発生する生化学システムを定義する。反応 R_2 が発生した場合、修正が必要な反応は、 R_2 で増減する分子が Reactants に含まれる反応である。 R_2 発生により、分子 B, C が減少し、 D が増加する。分子 B, C, D を Reactants に持つ反応は R_1, R_3 なので、 τ_1, τ_3 を修正しなければならない。そのため、DGの R_2 のエントリには R_1, R_3 が記述される。 τ_j の修正は、式8に従う。

$$\tau_{j,\text{new}} = \frac{\alpha_{j,\text{old}}}{\alpha_{j,\text{new}}} (\tau_{j,\text{old}} - \tau_\mu) + \tau_\mu \quad (8)$$

Gibsonらによると、51種類の分子と71種類の反応で構成される λ フェージモデルの場合、1回の反応発生による τ_j の平均修正回数は $D = 4.2$ 回である[1]。Caoらは、反応が数百種類程度の生化学システムでは、反応数 N だけでなく、平均 τ_j 修正回数 D も計算時間に影響を与えることを明らかにしている[8]。

2.4.4 アルゴリズム

NRMは、分子数の時間変化を以下の手順で計算する。

STEP1 初期値を設定する

- 初期分子状態ベクトル $\bar{X}(t_0)$
- DGの生成

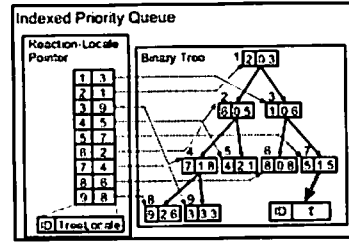


図1 IPQの構成

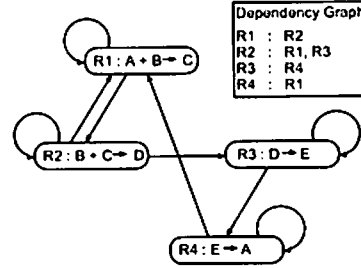


図2 DGの構成

- システム時間 $t = 0$

STEP2 全反応について、式7から τ_j を計算し、IPQに格納(ADD)する

STEP3 2分木の根ノードを、発生反応 R_μ として選択する

- 分子状態を更新する $\bar{X}(\tau_\mu) = \bar{X}(t) + \bar{\nu}_\mu$
- システム時間を根ノードの値で更新する $(t = \tau_\mu)$

STEP4 R_μ の次の発生予測時間 τ_μ を式7の結果と現在時間 t を加算して求め、ノードを更新(UPDATE)する

STEP5 DGから R_μ 発生時に修正する τ_j のリストを得る。リストにある各反応の τ_j を2分木から読み出し、式8によって τ_j を更新する

STEP6 STEP3に戻る

2.5 関連研究

FPGAを利用してSSAを高速化しようとする研究は2004年頃から複数進められている[3][9][4]。これらの研究は、高い周波数で動作する汎用プロセッサの約20倍以上と、大きな速度向上率を示している。しかし、NRMをそのままFPGAで実行することはせず、アルゴリズムの近似と単純化を行っており、誤差範囲が生化学システムごとに変わるため、シミュレーションの精度に関する検証がそれぞれ必要になる。DM-SSA[4]はアルゴリズムの近似は行わず、浮動小数点演算器を組み合わせた機能ユニットに、データを間断なく投入することで大きな速度向上を実現した。これらの研究は、すべて生化学システムごとに専用の回路を生成するため、検証および合成・配置記録に無視できないオーバーヘッドが発生する場合がある。

本研究では、これらの問題に対応しながら、NRMを近似せずにFPGAで実行するアプローチを保持し、生化学システムに依存しない回路の設計を目標とする。同時に、FPGAの規模、対象システムに合わせた柔軟な回路生成が可能な構成を目指す。

3. 設 計

3.1 NRM の解析

NRM は演算とデータの観点から、以下の 3 つの部分に分類できる。

(1) 浮動小数点 (FP) 演算

- k_j のテーブルを含む propensity の計算 (式 4-6)
- 式 7, 8 の計算

(2) 生化学システムごとに設定されるが、実行中に値が不変の参照用データ

- 状態更新ベクトル \mathbf{r}
- Dependency Graph
- Reactants テーブル

(3) 生化学システムの状態を保持するデータテーブル

- Indexed Priority Queue
- 分子数テーブル

各分類について、詳細に検討する。第 1 項、第 2 項の FP 演算器やメモリ (BlockRAM) を組み合わせて実装されたハードウェアモジュールを演算ユニット (FU) と呼ぶ。第 1 項の FU は、propensity 演算、式 7 の演算、式 8 の演算を行う 3 種類であり、それぞれ PF-FU、TC-FU、TR-FU と呼ぶ。第 2 項の FU は、生化学システムごとに設定され、シミュレーション実行中は値が変化しない定数テーブルが 3 種類あり、状態更新ベクトルテーブル (UM-FU)、Dependency Graph (DG-FU)、Reactants テーブル (RT-FU) である。第 3 項のシミュレーション状態を保持するデータテーブルをデータユニット (DU) と呼ぶ。DU は最小構成で IPQ (2 分木とポインタテーブル)、分子数テーブルが必要であり、計算時間とのトレードオフにより、propensity テーブル等が追加される。

動作周波数が汎用 PC のプロセッサと比べて数十分の一である FPGA を用いて、アルゴリズムを近似せずに単一のシミュレーションを高速に実行することは困難である。現在実装されている FPGA 用の FP 演算器は、全てパイプライン化されているため、1 クロックに 1 回の演算を実行することができる [10]。そのため、多数の DU と少数の FU を適切に接続した回路を構成し、FU のパイプラインを埋める通信を行うことで、複数回分のシミュレーションの並列実行による高スループットな演算の実現を目指す。このような高速化の方法は、生化学シミュレーションの主な用途が反応パラメータの決定を目的とするパラメータサーベイであること、さらに SSA はモンテカルロ法であり、同一パラメータによる多数回の実行結果が必要であることから、十分に合理的であると言える。

3.2 データ転送網によるユニット間接続

現在までに FU と 20 個の DU をクロスバで接続した NRM シミュレータを実装し、評価を行っている [11]。このシミュレータは、各 DU に割り振られる各 FU への通信権が時分割であり、木の再構成の待ち時間をワーストケースで動作する。対象の生化学システムが大きい場合に、計算速度の下落が大きいこと、20 個の DU でも FU のパイプラインの使用率が低く、DU 数が増えた場合にクロスバの面積増加が深刻であることなど、

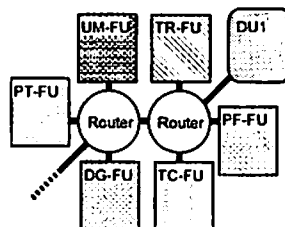


図 3 データ転送網を用いた NRM 回路の例

いくつかの問題が存在していた。

そこで、DU-FU 間が少数のポートを持つ複数のバスやルータを介して接続された NRM 実行回路を提案する。図 3 に 5-ポートのルータを用いた DU を 1 個だけ持つ NRM 実行回路の例を示す。ルータを介して DU を複数接続することで、同時に動作する DU の数を増やすことができる。全体をクロスバで接続する方法と比べ、DU 数が増えた場合に、ユニット間接続に必要な面積の増大が小さいため、より効率的な面積利用が可能になると考えられる。また、実行する FPGA の回路規模が異なる場合にも柔軟に対応できるため、回路面積の増大に比例したスループットの向上も期待できる。

NRM は必ず、DU から何らかのデータを FU に送り、演算あるいはテーブル引きの結果が、FU から送信元 DU に返送される手順を繰り返して計算が進展していく。単一 DU の、初期化処理を除いた反応 1 サイクルの演算は、以下の手順で行われる。

- (1) 2 分木の根ノードの反応番号 μ を UM-FU に送り、増減する分子番号を得、分子の数を更新する
- (2) μ を RT-FU に送り、 R_μ の Reactants を得る
- (3) 得られた Reactants を PF-FU に送り $a_{\mu,new}$ を得る
- (4) $a_{\mu,new}$ を TC-FU に送り、 $\tau_{\mu,new}$ を得る
- (5) μ を DG-FU に送り修正が必要な反応番号 x を得る
- (6) τ_x を TR-FU に送り $\tau_{x,new}$ を受け取る
- (7) 他に修正すべき反応があれば、(5) に戻る
- (8) (1) に戻る

3.3 Data Unit

シミュレーション状態を保持するために必要なデータは以下の 3 種類である。

- (1) IPQ (2 分木、ポインタテーブル)
- (2) 分子状態テーブル
- (3) Propensity Table

これらのメモリは、Virtex-II-Pro の BlockRAM を使用している。BlockRAM は 1 つあたり 18bits \times 1024 エントリであることから、各メモリのエントリは 1024 を使用している。2 分木はアドレス 0 番地を使用しないため、この構成では、最大で反応 1023 種類、分子 1024 種類で構成される生化学システムのシミュレーションを実行することができる。2 分木は、BlockRAM 外部に木のノードの比較、交換を行うコントローラ (Tree Controller) を実装して実現している。より上位のモジュールからは、命令入力 (4bit)、ノード入力 (46bit)、根ノード

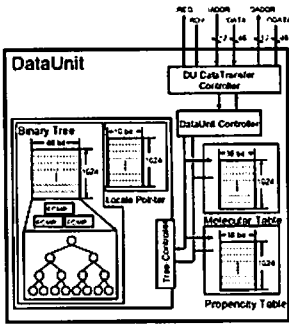


図4 Data Unit の構成

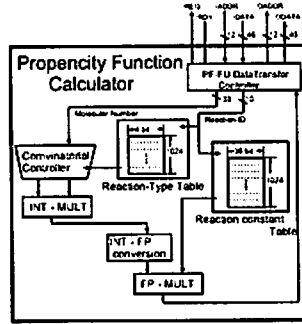


図5 PF-FU の構成

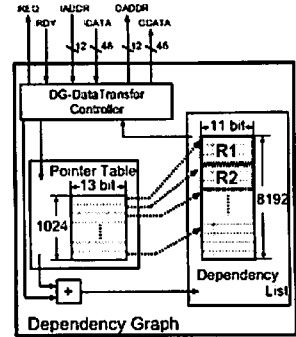


図6 DG-FU の構成

ド値 (46bit), 出力ノード値 (46bit) の 4 種類の入出力でコントローラが制御される。2 分木は参考文献 [11] で実装された 2 分木から、先読み機能を削除したものである。これは、先読みのための回路のオーバヘッドが大きいことによる。ノードの交換には 3 クロック掛かり、交換と同時にポインタテーブルの値も交換される。 a_j は必要になるごとに再計算することが可能だが、実行時間の短縮のため、Propensity Table に保持している。DU 外部との通信は、専用のコントローラが行う。

3.4 Functional Unit

3.4.1 FP 演算を行う FU

FU は 2 分類計 6 種類必要になる。第 1 分類の FU はパイプライン化された FP 演算器の組み合わせで構成され、PF-FU, TC-FU, TR-FU の 3 種類ある。第 1 分類の FU について、PF-FU を例に説明する。PF-FU は、反応番号 j と何種類かの分子の数を入力とし、 a_j を計算結果として出力する FU である。PF-FU は内部に k_j テーブル、反応型テーブルを持っており、入力された反応番号 j をアドレスとしてテーブルを引くことでその反応の k_j および反応型を得ることができる。現在の PF-FU は 2 次反応までの a_j を計算できるが、同じインタフェースで 3 次以上の反応にも対応することが可能である。2 次反応までの a_j の計算は式 4-6 の 3 種類であり、反応型は式 4-6 を選択するためのラベルである。分子数は整数のため、Reactants の組合せ数は、入力のビットシフトと整数の乗算で実現できる。図 5 に PF-FU の構成を示す。PF-FU の入出力部には DU と同じ信号線があり、外部と通信が行われる。

入力から計算結果の出力までに掛かる数十段分のパイプラインの占有率が、高スループットを実現するための評価基準のひとつになる。

3.4.2 定数テーブルを持つ FU

FU の第 2 分類は、シミュレーション中に利用される DG や Reactants に関するテーブルを持つユニットであり、DG-FU, UM-FU, RT-FU の 3 種類存在する。これらのテーブルの値は、シミュレーション対象である生化学システムごとに設定され、実行中は値が変化しない。

第 2 分類の FU について、DG-FU を例に説明する。DG-FU

は反応番号 j を入力とし、 R_j に関する DG のリストを出力する FU である。DG のリストの数は反応ごとに異なるため、十分に大きな DG 本体メモリと、各反応のリストが本体メモリのどの位置から始まるかが格納されているポインタテーブルで構成される (図 6)。現在の実装では、DG-FU からは反応を 1 種類ずつ返送する。入力として、反応番号 j とオフセット x を受け取り、 $DG(Pointer(j) + x)$ を返送する。

3.5 Router (Switching Unit)

DU1 個の試験評価を取るために、以下のような機能を持つ簡易なルータの実装を行い、ユニット間の接続に使用している。

- 5 ポートの簡易なルータ
- 転送は 1 クロック
- 衝突検出機能無し
- アドレス併走方式 (アドレス 12bit, データ 46bit)
- ソースルーティング方式
- 1bit 毎に転送を行い、バスの占有などは行わない

4. 評価

4.1 面積評価

Verilog-HDL を用いて実装を行い、Xilinx ISE7.1i を用いて Virtex-II PRO の BlockRAM, 組み込み乗算器などを生成して使用している。また、Xilinx ISE7.1i を用いて合成、配置配線を行った。ターゲットデバイスとして Virtex-II PRO XC2VP70 FF1517-5 を選択している。

データ転送網で接続された FPGA による SSA-NRM シミュレータの予備評価として、図 3 のような DU を 1 つだけ持つシミュレータ (NRM-DU1) を実装した。その面積および動作周波数に関する評価を表 1 に示す。評価に用いられた FU は、データが衝突しない前提で実装された簡単なコントローラのみを保持している。

DU は約 1180 slice, ルータは約 159 slice であるため、DU を 15 個程度搭載した SSA-NRM シミュレータが構築できると考えられる。各 FU のパイプラインの占有率は 2% に満たないため、FU は各 1 個あれば十分に対応できる。

4.2 性能評価

アルゴリズムの挙動を詳細に解析するために、NRM を実行

表1 面積・動作周波数の評価

Modules	Slices ^a [%]	18x18 Mult. ^a	Block RAM ^a	Freq. [MHz]
DU	1180(3.57)	-	8	81.65
TC-FU	7055(21.32)	13	6	100.22
TR-FU	3700(11.18)	4	-	102.84
PF-FU	1274(3.85)	13	2	134.43
UM-FU	58(0.18)	-	7	102.28
DG-FU	44(0.13)	-	6	144.20
RT-FU	7(0.02)	-	2	190.15
SW-5port	159(0.48)	-	-	166.53
NRM-DU1	12086(36.53)	31	30	81.65

^a Capacity(XC2VP70) : 33068 slices, 728 BlockRAM and 328 Multipliers

表2 NRM-DU1のスループットの評価

System	N	D	SW [*]	HW [▽]	HW/SW
LTS	4	2	1.01	0.29	0.29
TIS	4	0	1.94	0.95	0.49
	10	0	1.64	0.93	0.57
	100	0	1.10	0.88	0.80
	1000	0	0.74	0.84	1.13
LCS	4	1	1.11	0.47	0.42
	10	1	0.89	0.46	0.51
	100	1	0.77	0.43	0.56
	1000	1	0.60	0.41	0.69

^{*} Linux2.4.31 gcc3.3.5(-O3) on Xeon2.80GHz 4.0GB RAM

[▽] Operating at 81.65MHz

^{*} 単位は [Mcycles/sec]

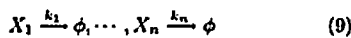
するソフトウェアプログラムをC++で実装し、以下の3つのベンチマークを用いて速度の評価を行った。

Lotka System(LTS)

4種類の分子が4種類の反応を発生させるシステム [7]($D=2$)

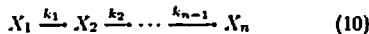
Totally Independent System(TIS)

各分子が壊変するシステム [8]($D=0$)



Linear Chain System(LCS)

分子Aが分子Bになる反応が、連鎖するシステム [8]($D=1$)



LTSおよびNの値を変えたときのTIS, LCSを対象の生化学システムとして、汎用プロセッサ, NRM-DU1のスループットに関する評価を表2に示す。Nは生化学システムで定義された反応数、Dは1サイクルあたりの τ_j の平均修正回数である。

LTSのような小規模な生化学システムの場合、FPGAによる高速実行の効果が得られていない。TISとLCSの結果から、Dの値が大きいほど、NRM-DU1の演算性能が低くなることからわかる。これは、サイクルごとに τ_j の修正(3.2のSTEP5-7)が平均D回実行されるためである。 τ_j の修正は依存関係が木の更新のみのため、DGからのデータの受信やTR-FUへの計算要求はD回連続して実行する方法にすることで、スループットの向上が可能である。

また、生化学システムが大規模になるほど、ソフトウェアのスループットの下落が大きいことがわかる。この理由として、木の再構成時間が汎用プロセッサに比べて高速であることが挙げられる。TISのように各反応が他の反応に全く影響を及ぼさない特殊なシステムを除いて、単一のDUの計算速度は汎用プロセッサに劣るが、複数のDUを配置する回路面積の余裕があることから、10個-15個程度のDUの並列実行により、大きな演算能力を得られると期待できる。

5. 今後の展開

複数DUを持つNRMシミュレータの実現のために、効率的なルータの構成の検討および評価を行う予定である。また、アドレス併走方式のデータ転送は効率が悪い。そのため、データ網の混雑や、ルーティング用のデータ網と面積の関係を調査し、効率の良い通信方法の検討を行う。

謝 辞

この研究は文部科学省の平成17年度科学技術振興調整費による「システム生物学者育成プログラム」の一環として行われた。

文 献

- [1] Michael A.Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, Vol. 104, No. 9, pp. 1876-1889, 2000.
- [2] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, Vol. 115, No. 4, pp. 1717-1733, Jul. 2001.
- [3] Lukasz Salwinski, et al. In silico simulation of biological network dynamics. *Nature Biotechnology*, Vol. 22, No. 8, pp. 1017-1019, Aug. 2004.
- [4] Masato Yoshimi, et al. Stochastic simulation for biochemical reactions on FPGA. In *The 14th International Conference on Field Programmable Logic and Applications*, Vol. 3203 of *Lecture Notes in Computer Science*, pp. 105-114. Springer, Aug. 2004.
- [5] Masato Yoshimi, et al. The design of scalable stochastic biochemical simulator on FPGA. In *2005 IEEE International Conference on Field Programmable Technology*, pp. 339-340. IEEE, Dec. 2005.
- [6] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, Vol. 22, pp. 403-434, 1976.
- [7] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, Vol. 81, No. 25, pp. 2340-2461, Dec. 1977.
- [8] Yang Cao, et al. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *Journal of Chemical Physics*, Vol. 121, No. 9, pp. 4059-4067, 2004.
- [9] John F.Keane, et al. A compiled accelerator for biological cell signaling simulations. In *The 12th Int. Symp. on Field-Programmable Gate Arrays(FPGA)*, pp. 233-241, Feb. 2004.
- [10] 古見真聡ほか. FPGAを用いた科学技術計算向け浮動小数点演算器の設計. 第26回研究会1-7, パルテノン研究会, May. 2005.
- [11] 古見真聡ほか. ヒープ木を用いた効率モデル生化学シミュレータのFPGAへの実装. RECONF2005 9, 電子情報通信学会, May. 2005.