

FPGA を用いた確率モデル生化学シミュレータ

吉見真聡[†] 長名保範[†] 岩岡 洋[†]
 西川由理[†] 小嶋利紀[†] 柴田裕一郎^{††}
 岩永直樹^{†††} 舟橋 啓^{††} 広井賀子^{††}
 北野宏明^{††} 天野英晴[†]

確率モデル生化学シミュレーションアルゴリズム (SSA) は、定義した生化学システムの確率的挙動を厳密に計算できるアルゴリズムとして知られている。しかし、SSA の実行には膨大な計算時間が必要であり、高速な実行環境が求められている。本論文では、高速実行の一手法として、Xilinx 社の FPGA (XC2VP70-5) を用いて、SSA (First Reaction Method) を実行する回路を実装、評価した結果について述べる。高速化は、パイプライン化した演算ユニットを使い、複数スレッドのシミュレーションを同時に実行することで実現する。シミュレータ回路は、中間データを BlockRAM に保持し対象の生化学システムごとの回路再構成を要しない、実用的な構造になっている。ベンチマーク的に定義できる生化学システム TIS, D4S で評価した結果、Xeon 2.80 GHz による実行と比較して、TIS では約 83 倍、D4S では約 95 倍のスループット向上が可能であることを確認した。

FPGA-based Stochastic Biochemical Simulator

MASATO YOSHIMI,[†] YASUNORI OSANA,[†] YOW IWAOKA,[†]
 YURI NISHIKAWA,[†] TOSHINORI KOJIMA,[†] YUICHIRO SHIBATA,^{†††}
 NAOKI IWANAGA,^{†††} AKIRA FUNAHASHI,^{††} NORIKO HIROI,^{††}
 HIROAKI KITANO^{††} and HIDEHARU AMANO[†]

This paper discusses an FPGA implementation and evaluation of a Stochastic Simulation Algorithm (SSA) called the First Reaction Method. SSAs are widely known as rigorous methods for simulating the stochastic behaviors of various biochemical systems, but also as CPU-hogging applications due to vast repetition of the algorithm. This work accelerates the execution by achieving high throughput with concurrent simulations of highly utilized pipelined arithmetic units. For upgrading practical utility, the design stores intermediate data in a BlockRAM so that reconfiguration is unnecessary for different target biochemical systems. Benchmark results on an FPGA (Xilinx XC2VP70-5) have shown that the circuit provides throughput of approximately 83 times and 95 times compared to software execution on Xeon 2.80 GHz when it was evaluated with biochemical models called TIS and D4S, respectively.

1. はじめに

1980 年代後半からの分子生物学の急速な進展により、種々の生物の遺伝子配列や、細胞内の代謝系などに関する定量的なデータが蓄積されつつある。これら

のデータをもとに計算機を用いて、様々な条件の化学反応系を分子レベルでシミュレーションする方法が研究されてきた。近年では、生命現象をシステムとして理解しようとするシステムバイオロジー¹⁾ と呼ばれる研究分野が成立し、モデリング、シミュレーション、解析など、生化学分野と計算機分野の両面からの研究がさかんに続けられている。

生化学シミュレータは The Virtual Cell²⁾ や E-Cell³⁾ などが開発されているが、化学反応系のシミュレーションに要する時間は実時間の数倍から数万倍に及び、計算時間に関する問題は深刻である。特に、確率モデル生化学シミュレーションアルゴリズム (Stochastic Simulation Algorithm, 以下 SSA)⁴⁾ には、膨大な計

[†] 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

^{††} 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

^{†††} 長崎大学工学部情報システム工学科

Department of Computer and Information Sciences, Nagasaki University

算時間が必要になることが広く知られている(2.4節を参照)。この問題に対しては、PC/WS クラスタなどの並列システムを用いるのが一般的である。しかし、このような環境の導入、維持には大きなコストがかかるため、小規模な研究室では運用が困難である。また、生化学シミュレーションは、新しいアルゴリズムが次々と提案され、それらが組み合わされて実行されるため、高速演算が可能な専用ハードウェアを開発する方法でも対応が難しい。そこで、アルゴリズムの変更に対する柔軟性を備えた低コストな高速化手法として、リコンフィギャラブルシステムを利用したアプローチが有効であると考えられる。

本論文では、生物学計算用途に開発したFPGA搭載PCIボードであるReCSiP2-board (Reconfigurable Cell Simulation Platform⁵⁾)を対象に、First Reaction Method (以下FRM⁴⁾)と呼ばれるSSAを実行する回路を実装し、性能評価を行った。

2. 確率モデル生化学シミュレーション

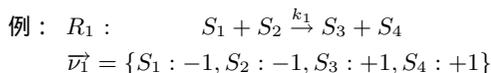
一般的に確率モデルによる生化学シミュレーションとは、Gillespieが開発した化学反応系のモデリング手法を用いたモンテカルロ法のことを指す⁴⁾。確率モデルにおいて、細胞などシミュレーション対象となる生化学システムは化学反応のリストで表現される。各反応は、発生確率と発生時に増減する分子で定義される。各反応のパラメータと、初期分子数が与えられたときに、分子数の時間変化を計算するアルゴリズムがSSAである。

2.1 確率モデル生化学システム

確率モデルにおける生化学システムとは、以下の3項目が定義された化学反応系のモデルである。時刻を t で表し、初期時刻を t_0 とする。

- M 種類の分子 S_1, \dots, S_M と、その初期分子数 $\vec{X}(t_0) = \{X_1(t_0), \dots, X_M(t_0)\}$ ($X_i(t)$: 整数)。
- N 種類の反応 R_1, \dots, R_N とその発生によって増減する分子を示す状態更新ベクトル $\vec{\nu}_j$ 。状態更新ベクトルは、反応のReactants (基質) とProducts (生成物) のリストからなる。
- 各反応の確率反応速度定数 $\vec{k} = \{k_1, \dots, k_N\}$ (k_j : 実数)。 k_j は生化学システムの体積、温度および R_j のReactantsとなる分子の物理的性質で与えられる。

以下に反応定義の例を示す。



反応 R_1 は、そのReactantsである S_1 と S_2 が反応して発生し、反応の発生頻度は k_1 で与えられる。反応 R_1 の発生により、 S_1, S_2 の数 X_1, X_2 がそれぞれ1減少し、 R_1 のProductsである S_3, S_4 の数 X_3, X_4 が1増加する。

2.2 First Reaction Method (FRM)

SSAは生化学システムから、次に発生する反応と、その発生時刻を1つずつ計算し、分子数の時間変化を求めるためのアルゴリズムである。発生する反応を決定する方法はいくつか提案されており、SSA開発初期に提案された方法がFRM⁴⁾である。FRMでは、現時刻の分子数から、反応の予測発生時間 τ を計算し、その中から最短の時間 τ_μ で発生する反応を、次に発生する反応 R_μ として決定する。

FRMは以下のステップで進展する。(2)~(4)の計算を1サイクルとし、計算時間の大部分はこの繰返して占められる。

- 初期値を設定する。
- R_j ($j = 1, \dots, N$) について、式(1)から予測発生時間 τ_j を計算する。

$$\tau_j = \ln(1/r) / a_j \quad (1)$$

- 式(2)から反応発生時間 τ_μ と、その反応のインデックス μ を求める。

$$\tau_\mu = \min(\vec{\tau}) \quad (2)$$

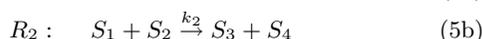
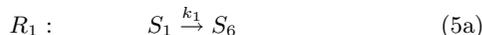
- μ と τ_μ により、分子数と現在時刻を更新する。

$$\vec{X}(t + \tau_\mu) \leftarrow \vec{X}(t) + \vec{\nu}_\mu \quad (3)$$

$$t \leftarrow t + \tau_\mu \quad (4)$$

- (2)に戻る。

r は $[0, 1]$ の範囲の一様乱数である。 a_j は propensity と呼ばれる反応の起こりやすさを表す値であり、サイクルごとに全反応について計算される。



式(5a)、式(5b)、式(5c)で表される反応の propensity は、各反応のReactantsである分子 S_1, S_2 の時刻 t における分子数 $X_1(t), X_2(t)$ と確率反応速度定数 k_j を用いて、それぞれ式(6a)、式(6b)、式(6c)で求められる。

$$a_1 = X_1(t) \times k_1 \quad (6a)$$

$$a_2 = X_1(t) \times X_2(t) \times k_2 \quad (6b)$$

$$a_3 = X_1(t) \times (X_1(t) - 1) / 2 \times k_3 \quad (6c)$$

反応の propensity は、Reactantsの組合せ数に k_j を乗算した値である。式(5a)のようなReactantが1つ

の反応を 1 次反応, 式 (5b), 式 (5c) のような 2 つの反応を 2 次反応と呼ぶ。

2.3 確率モデルの利点

現在の生化学シミュレーションでは計算時間の問題から, 分子濃度の時間変化を常微分方程式でモデル化する解析モデルが使用されることが多い。しかし, 濃度変動を決定論的に記述する解析モデルを用いた場合には, 解析不可能な課題も存在する。確率モデルの使用により解けるようになると予想される重要な問題として, シグナル伝達系や転写・翻訳系などにおいて存在するゆらぎの問題があげられる。また, 実数を用いる数値計算で問題になる誤差の蓄積や初期値感受性を回避し, 生化学システムの定義に厳密なシミュレーションが可能な点も, 確率モデルの大きな利点である。

2.4 高速化の必要性

以下の 3 つの理由から, 確率モデルシミュレーションには膨大な計算時間が必要になる。

- (1) 反応が 1 回ずつ計算されるため, 膨大な回数, 演算が繰り返される。
- (2) モンテカル口法であるため, 数千回以上の実行結果を統計的に解析しなければならない。
- (3) 生化学シミュレーションはパラメータサーベイで利用されるため, 多数のパラメータセットでパラメータ空間をスキャンしなければならない。

Kierzek の計算時間の例⁶⁾によると, 大腸菌の LacY, LacZ と呼ばれるタンパク質の発現に関わる酵素反応のモデルを, 確率モデル生化学シミュレータ STOCKS を用いて Pentium III 800 MHz の PC 上で実行した場合, 1 世代 2,100 秒間の結果を 100 回分得るために, 約 2.5 時間かかる。大腸菌 10 世代分のシミュレーションでは, 細胞分裂にともない計算しなければならない反応数が増大し, 約 90 時間かかる。これは, パラメータサーベイに利用できる実用的な計算時間ということではできない。

SSA にかかる計算時間の問題を改善しようとする研究は, 生物学分野および計算機分野から関心を集めている。実行環境の面からは, PC クラスタを用いて計算時間を評価する研究が行われている⁷⁾。確率モデルは並列処理により高速化しやすい性質を持つが, PC クラスタ環境は価格やサイズ, 消費電力の面から, 個人単位での導入, 利用には問題があり, より小型で高い性能を実現できるシステムが求められている。この場合, 生化学シミュレーション専用のハードウェアを開発して対応する方法が考えられるが, 以下の点から得策ではない。

- (1) 低価格で専用 LSI を提供できるほど多数のチップが利用されない。

- (2) 計算量の削減手法^{8),9)} や, 解析モデルと確率モデルの複手法^{10),11)} など, アルゴリズムの研究もさかんであるため, ハードウェアが固定されるとそれらの改良を反映することができない。

このような場合, 書き換え可能な FPGA を利用したりコンフィギャラブルシステムの利用が, 価格, 柔軟性, 高速性の点で有利である¹²⁾。SSA は, 単純な演算を膨大な回数繰り返すため, ループレベル, スレッドレベルの並列性を持つ。そのため, パイプライン化された演算ユニットにデータを連続投入して回路の利用率を上げること, 複数のシミュレーションを並列に実行することにより, 高スループットな演算の実現が期待できる。

そこで, 本論文ではバイオインフォマティクス用アクセラレータ ReCSiP 上に確率モデルシミュレータ回路を実装し, 高速化を行うシステムを開発する。

3. ReCSiP

3.1 ReCSiP の概要

ReCSiP とは, 生命科学向け高速演算環境の実現を目的として開発が続けられている, FPGA とメモリで構成される PCI ボードとホスト CPU の協調システムである。ReCSiP を用いた高速化が試みられているアプリケーションとして, 2 つの例があげられる。第 1 に, 解析モデル生化学シミュレーションの高速実行環境の開発が行われている¹³⁾。第 2 に, 顕微鏡から得られた線虫の画像を解析する細胞分裂パターン計測システムが, PC クラスタの代替として利用されている¹⁴⁾。

3.2 ReCSiP-2 board

ReCSiP の動作確認のためのリファレンス環境として, ReCSiP-2 board が開発された。図 1 に ReCSiP-2 board の概観を, 図 2 に ReCSiP-2 board の構成をそれぞれ示す。ReCSiP-2 board は, 演算を実行する FPGA として Xilinx 社の Virtex-II Pro (XC2VP70-

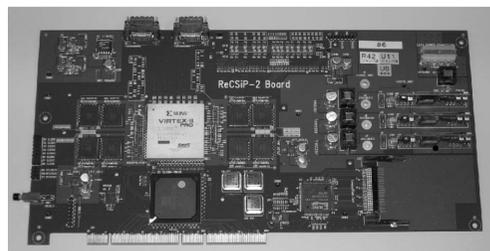


図 1 ReCSiP-2 board の写真

Fig. 1 A picture of ReCSiP-2 board.

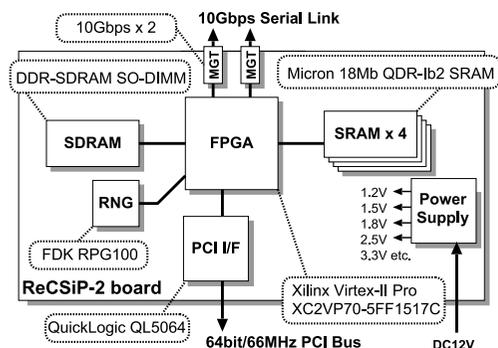


図 2 ReCSiP-2 board の構成図

Fig. 2 A structure of ReCSiP-2 board.

5) を搭載している。PCI バスを介したホスト PC との通信部分は QuickLogic 社の QL5064 が担い、64 bit/66 MHz の PCI バスに接続できる構成となっている。メモリとして、並列アクセス可能な 36 bit データ幅の QDR-SRAM (2 チップで 1 セット) が 4 セットと、SO-DIMM ソケットによる DDR-SDRAM が実装されている。また、物理乱数生成 IC である FDK 社の RPG100¹⁵⁾、ボード間の相互接続用に Virtex-II Pro の MGT (Multi Gigabit Transceiver) に接続された HSSDC コネクタを搭載している。ボードはホスト PC からメモリデバイスとして認識され、ボード上のメモリや FPGA 上の任意のレジスタをメモリ空間上に割り当てることが可能である。ホスト PC 上に割り当てられたメモリ領域へ値を書き込むことでハードウェアの制御が、メモリ領域から値を読み込むことでボードの状態や結果などを得ることができる。また、主記憶とボード上のメモリ間でデータ転送を行う際に、ボードがマスタとなって DMA 転送を行うホストインタフェースも開発中である。PCI バスとボード上のローカルバスのクロックは分離されており、ハードウェア設計者の負担も軽減されるように配慮されている。

4. リンコンフィギャラブルシステムによる高速化

4.1 関連研究とその問題点

FPGA を用いて SSA を高速化しようとする試みは、2004 年頃から本研究のほかにも複数進められている。Salwinski ら、Keane らは、生化学システムの定義を入力としてシミュレータ回路を生成する SSA 実行環境を開発している^{16),17)}。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、計算速度の面では大きな効果が得られている。しかし、以

下の 2 つの点から、実用的な利用ができていない。

第 1 に、アルゴリズムが近似、単純化されていることがあげられる。Salwinski ら、Keane らの手法では、SSA をそのまま FPGA 上で実行することはせず、近似アルゴリズムを提案し、各パラメータを実数型から整数型に変換して計算を行う回路を生成している。近似アルゴリズムは、あるタイムステップ Δt ごとに、各反応の発生の有無を判定し、まとめて分子数を更新するアルゴリズムである。このアルゴリズムでは、 Δt の決定方法が正確ではなく、 Δt が大きすぎればシミュレーションが不正確になり、小さすぎれば計算速度が遅くなってしまふ。また、現実の生化学システムは、マルチスケールなパラメータを含んでおり、この場合、反応の発生頻度に極端な差が生じていることが多い。実数型変数を整数型に置き換えることにより、定義された反応が発生しなくなってしまうこともある。そのため、生化学システムごとにシミュレーションの正確性に関する検証が必要になる。より厳密な高速 SSA 実行環境には、浮動小数点 (以下 FP) を用いた演算が必要である。

第 2 に、生化学システムの規模に回路面積が比例して増加する構成のため、大規模な生化学システムの計算回路が構成不可能であることがあげられる。Keane らの開発したシステムを例にあげると、反応を 120 種類程度定義した生化学システムで FPGA (XC2V8000) の回路面積を超え、シミュレータ回路の生成が不可能になる¹⁷⁾。筆者らの研究における過去の実装¹⁸⁾では、アルゴリズムの近似は行わず、FP 演算器を組み合わせた演算ユニットに、データを連続して投入するようスケジューリングすることで大きなスループットが得られることを確認したが、回路面積が生化学システムの規模に比例して大きくなる問題は解決できていなかった。数百種類の反応を含む生化学モデル¹⁹⁾が構築されつつある現在において、シミュレータ回路には反応が 1,000 種類以上定義された大規模なシステムをシミュレーション可能であることが求められる。また、前述の研究は、生化学システムごとに専用の回路を生成するため、モデルの修正にともなう回路の合成・配置配線および動作検証に要する時間も無視できない。そのため、生化学システムの形状に依存しない回路構成が望ましい。

4.2 問題点を改善する設計方針

4.2.1 浮動小数点実数を用いた演算

近似や型変換を行わず、FP 演算を含む FRM を実行する回路を設計する。FP 演算は一般的に短いクロックで計算結果を得ることはできないため、汎用プロセッ

サと比較して動作周波数に数十倍の差がある FPGA で高速な演算を実現することは困難である。しかし、FRM のアルゴリズム (2.2 節) の (2) において、各 τ_j の計算には依存関係がなく、並列実行が可能である。そのため、パイプライン化した propensity と式 (1) を計算する FP 演算ユニットに、分子数の値を連続して入力することで、1 回あたりの τ_j 計算時間を短縮し、高スループットな処理を実現する。

4.2.2 大規模生化学システムに対応した構造

シミュレータ回路は、生化学システムの規模と回路面積に相関が弱く、回路を生成することなく任意の形状の生化学システムをシミュレーション可能な構造とする。これは、propensity 計算のための分子番号を指定するポインタ、および分子数を格納するテーブルを、Virtex-II Pro の BlockRAM で構成することで実現する。

BlockRAM は様々な構成で使用することができるが、18 bit \times 1,024 words を 1 単位として持つため、1,024 words 単位でデータを保持する構成が効率が良い。分子数と反応速度定数、状態更新ベクトルを BlockRAM に格納することで、1,024 種類までの反応と分子が定義された生化学システムまで、回路面積が増大しないシミュレータ回路の構築が可能になる。また、反応の形状 (反応の次数と Reactants) も BlockRAM に保持する構成で、生化学システムごとに回路を生成する必要がなくなる。1,024 種類を超える反応と分子が定義されたより大規模な生化学システムをシミュレーションする回路も、BlockRAM のサイズの変更で対応することができる。

4.2.3 マルチスレッド実行

シミュレータ回路は、演算ユニットのパイプライン利用効率を向上させるために、複数のシミュレーションを並行して実行する構造を持つ。固定した構造の回路で計算を実行するため、反応 1 サイクルあたりの計算時間は、 τ_j を計算する回数 (式 (1)) である N に比例して長くなる。生化学システムに定義された最後の反応 R_N の Reactants の分子数が演算ユニットに入力されてから、反応が決定して分子数が更新されるまで、演算ユニットへの入力は空き状態となる。この空きを、他のシミュレーションスレッドを実行することで、回路の利用効率を向上させる。

また、FPGA の面積に余裕がある限り、シミュレータを複数ユニット構成し、並列に動作させることで、高スループットな実行環境を構築する。パラメータスキャン的な用途やモンテカルロ法である確率モデルの性質から、複数のシミュレーションスレッドを同時に

実行する方法が有利である。

5. 実装

5.1 確率モデル生化学シミュレータ回路の構成

本論文では、1,024 種類までの反応と分子が定義された生化学システムを計算対象とするシミュレータ回路を実装例として示す。このとき、生化学システムで定義される反応は 2 次反応または 1 次反応とする。これは、化学反応は分子の分解、あるいは 2 つの分子の衝突により発生するといえることから、2 次反応まで対応するシミュレータであれば、多くの生化学システムのシミュレーションが実行できるためである。

実装するシミュレータ回路の構成を図 3 に示す。FPGA 上に実装するシミュレータ回路全体を FRM-FPGA と呼び、FRM-FPGA 内で並列実行される各シミュレータユニットを FRM-UNIT と呼ぶ。FRM-FPGA は FRM-UNIT を 3 つ持っており、3 並列で動作する。FRM-UNIT は、パイプライン化された演算ユニット (FU) と、シミュレーション中のデータを保持するデータユニット (DU)、データの流れを制御するコントローラユニット (CU) の 3 つの部分からなる。DU はデータセット (DS_A , DS_B) を 2 つ持ち、1 つの FRM-UNIT は 2 並行のシミュレーションを実行する。このため FRM-FPGA では、6 つのシミュレーションが同時に実行されることになる。

図 4 に、反応の種類 $N = 5$ の生化学システムをシミュレーションする場合の実行例を示す。各 FPGA-UNIT の 1 サイクルの計算は、以下の手順で進行する。

- (1) CU が、反応 R_1 から順番に、Reactants の分子を読み出し、DU の DS_A へ出力する。 R_N の Reactants を読み出し終わったら、出力先データセットを DS_B に切り替え、 R_1 から順番に Reactants を読み出す (図 4: Phase 1)。
- (2) DU が、CU から受け取った Reactants の分子数を、指定されたデータセットから読み出し、

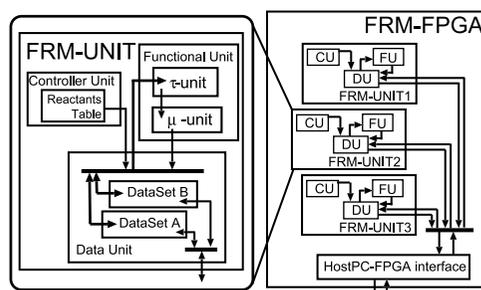


図 3 FRM-FPGA の構成

Fig. 3 A structure of an FRM-FPGA.

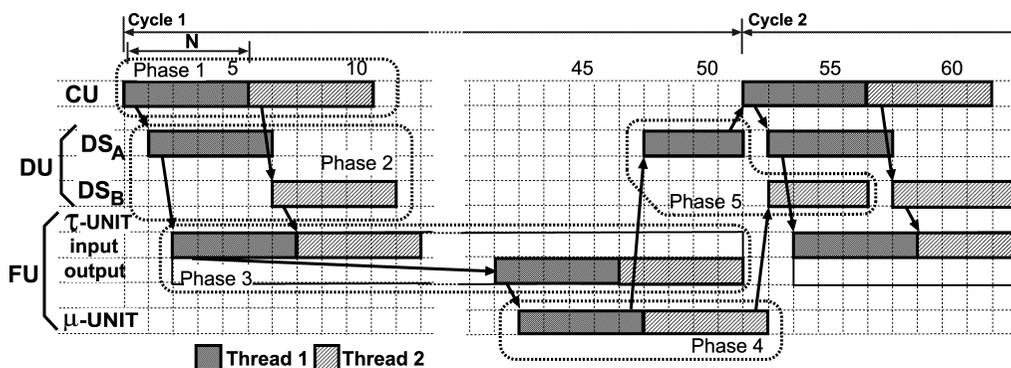


図 4 パイプライン動作の例

Fig. 4 An example of a pipeline operation.

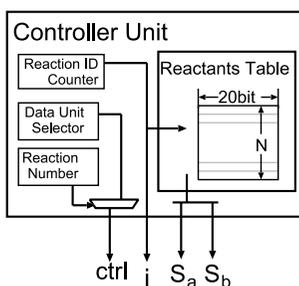


図 5 コントローラユニットの構成

Fig. 5 A structure of a controller unit.

FU へ出力する (図 4: Phase 2)。

- (3) 分子数を FU の τ -unit に入力し, τ_j を計算する。 τ -unit で求められた τ_j を μ -unit に入力する (図 4: Phase 3)。
- (4) τ_N が μ -unit に入力された次のクロックで, μ と τ_μ が決定され, τ_μ と反応 R_μ の状態更新ベクトル ν_μ を DU へ出力する (図 4: Phase 4)。
- (5) DU は ν_μ と τ_μ に従い, 時刻 t と分子数を更新する (図 4: Phase 5)。
- (6) DS_A の分子数更新と, (1) の DS_B への Reactants の出力の両方が完了した後, (1) に戻る。 1 サイクルの計算終了時に, ホスト PC へ結果を転送するかどうかを判定する。 現在の実装ではシミュレーション結果として, 一定のサイクルごとに時刻 t と全分子数をホスト PC 上のメモリへ DMA 転送することを仮定している。

FRM-FPGA を Verilog-HDL を用いて実装した。 以下に各ユニットの詳細を述べる。

5.2 コントローラユニット (CU)

図 5 に CU の構成を示す。 CU はサイクルの最初に, 反応 R_j の propensity を計算するために必要な

Reactants を指定するユニットである。 CU は, Reactants テーブルと, 定義された反応の数を示すレジスタ, 読み出し中の反応のインデックスを示すカウンタ, データセット指定フラグで構成される。 分子は最大 1,024 種類 (10 bit) であり, propensity の計算に必要な分子は最大で 2 つであるため, Reactants テーブルは BlockRAM を 20 bit 幅で使用する。

Reactants テーブルには各反応の Reactants が格納される。 反応サイクル開始とともに, 1 クロックに 1 つずつ, 反応 R_1 から順に, propensity 計算に必要な Reactants が Reactants テーブルから読み出される。 読み出された Reactants は, そのときの反応のインデックスとともに, DU へ出力される。 反応 R_N の Reactants が読み出された後, データセット指定フラグが切り替わり, また R_1 から Reactants を読み出し, DU へ出力される。 2 回分の Reactants の読み出しが終わった後に, DS_A の分子数更新が終わっていれば, 次のサイクルの読み出しを開始する。

Reactants は, DS_A 用, DS_B 用に N 種類の反応が計 2 回読み出されるため, 1 サイクルごとに, $2N$ クロックかかる。

5.3 データユニット (DU)

図 6 に DU の構成を示す。 DU は, 2 回分のシミュレーションデータ (分子数 \vec{X} と現在時刻 t) を保持するユニットである。 分子数を格納するテーブルは $32 \text{ bit} \times 1,024$ のサイズのデュアルポート BlockRAM で構成する。

DU には 2 つの機能がある。 まず, CU からデータセット指定フラグと Reactants を受け取り, 指定されたデータセットから分子数を読み出し, FU へ出力する機能である (図 4: Phase 2)。 次に, FU から状態更新ベクトル $\vec{\nu}_\mu$ と τ_μ を受け取り, 分子数と時刻を更新する機能である (図 4: Phase 5)。 現在時刻 t を

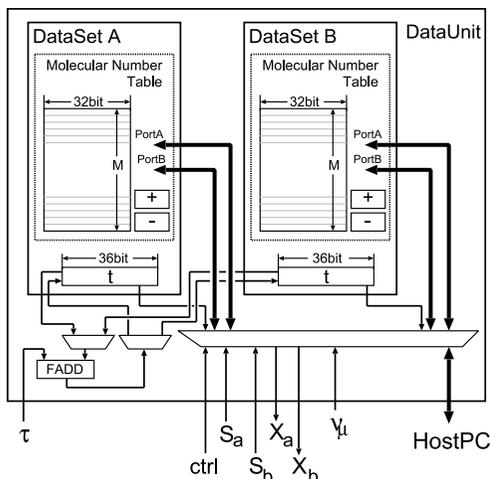


図 6 データユニットの構成
Fig. 6 A structure of a data unit.

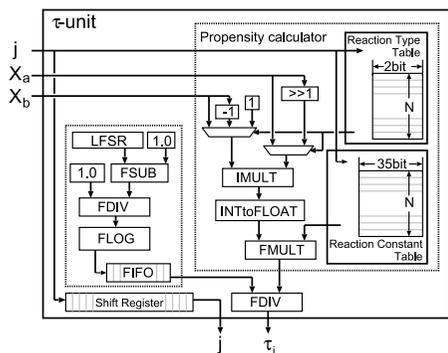


図 7 τ -unit の構成
Fig. 7 A structure of a τ -unit.

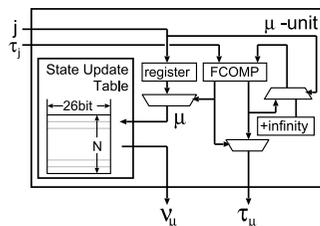


図 8 μ -unit の構成
Fig. 8 A structure of a μ -unit.

Constant テーブルの 2 種類のメモリを持つ．Reaction Type テーブルは， a_j を計算する際に，式 (6a)，式 (6b)，式 (6c) のうち，どの計算を適用するかを指定するデータが格納されている．Reaction Constant テーブルは確率反応速度定数 k_j が格納されるテーブルである．分子数は整数であるため，その組合せ数も必ず整数である．組合せ数を整数-FP 型変換モジュールを通して FP フォーマットに変換し， k_j との FP 乗算により， a_j が求められる．また， τ -unit は τ_j の出力と同時に反応インデックス j を出力するためのシフトレジスタを持つ． τ -unit において，Reactants の分子数が入力されてから τ_j が求められるまで，42 クロックかかる．

式 (1) における $\ln(1/r)$ は，propensity の計算とは独立である．そのため， $\ln(1/r)$ の値をバッファする FIFO を持ち，必要になるつど，FIFO からデータを 1 つ取り出し，乱数を 1 つ生成するよう指示することで， $\ln(1/r)$ の計算にかかる時間を短縮する．この方法によって，乱数周期を有効利用し，動作検証を容易にすることができる．

乱数生成器には，毎クロック 26 bit の乱数を生成する演算ユニットが使用できる．現在では，実装が容易であることから，167 bit の LFSR (Linear Feedback Shift Register) により M 系列の乱数を生成するモジュールを使用している．ReCSiP が搭載している RPG100 は，物理乱数の生成速度が最大 250 kbps と遅いため，使用する乱数すべてに物理乱数を使用することはできない．そのため，動作中の疑似乱数の一部を RPG100 の出力で置き換える方法が考えられるが，計算の再現性を重視して，現在の実装では使用していない．確率モデルで重要となる数値は，分子数の変動幅や一定時間後の生化学システムの状態であり，疑似乱数でも明らかな線形性が認められなければ十分であるとされ，乱数生成アルゴリズムの精度による重大な結果の違いはないと予測される．FPGA を用いた乱数生成器に関する研究^{20),21)} も複数進められているため，乱数生成モジュールを入れ換え，他の乱数生

更新するため，DU は FP 加算器を持っている．

μ -unit から v_μ を受け取ってから分子数を更新するために 4 クロックかかる．システム時刻 t は，次のサイクルの計算に用いられないため，次サイクルの τ_j 計算中に更新される．

5.4 演算ユニット (FU)

FU は，予測発生時間 τ を計算する τ -unit と， τ から最小値とその反応のインデックスを探索する μ -unit からなる．図 7 と図 8 にそれぞれ τ -unit と μ -unit の構成を示す．

5.4.1 τ -unit

τ -unit はパイプライン化された FP 演算器の組合せで構成されており，propensity の計算を含め，1 クロックに 1 つずつ τ_j を計算できる．反応インデックス j とその Reactants の分子数を入力とし，予測発生時間 τ_j を出力する．

τ -unit は，Reaction Type テーブルと，Reaction

成アルゴリズムも利用できる設計とした。

5.4.2 μ -unit

τ -unit で計算された τ_j はそのまま μ -unit の入力となり、 τ の計算が終わった後に、 μ -unit からは、そのサイクルでの発生反応 μ によって分子数が更新される分子のリスト \vec{v}_μ と発生時間 τ_μ が出力される。

μ -unit は、FP 比較器と反応インデックス j を保持するレジスタ、状態更新ベクトル v_j が格納される Molecular Update Table で構成される演算ユニットである。FCOMP は 2 つの FP データを入力として、小さい方の値と、1 クロック前と値が変わったかを示すフラグを出力する FP 比較器である。 μ -unit の FCOMP の入力は、一方は μ -unit 外部から入力され、他方は、反応インデックス j が 1 のときは $+\infty$ 、それ以外のときは FCOMP の出力が入力される。

ある生化学システムについて、 τ_1 から τ_N までの τ を順番に μ -unit に入力することで、 $\min(\tau)$ とその反応インデックス μ を求めることができる。 μ -unit は N クロックで τ_μ と μ を計算する。 μ -unit は τ_μ が求められたときに、 μ をアドレスとして、Molecular Update Table から更新する分子数を読み出し、 τ_μ とともに DU に出力する。

5.5 浮動小数点術演算器

一連の FU は、パイプライン化された単精度浮動小数点演算器²²⁾を組み合わせて実現する。表 1 に、Virtex-II Pro (XC2VP70-5) を対象デバイスとして、ISE8.1i を用いて合成、配置配線を行ったときの各演算器の面積と動作周波数を示す。FP 演算器は 1 つのデータを 35 bit で扱い、IEEE754 準拠の単精度浮動小数点フォーマットに丸め用の追加ビットを保持したまま、値を丸めずに演算を進める方法をとる。

対数器は、入力 x に対し、2 次の補間によって出力 $\ln(x)$ を計算する算術演算器である。 $[0, 1]$ の範囲の乱数は、26 bit の乱数を FP の仮数部として使い、正の符号と 127 を指数部として付加することで $[1, 2]$ の範囲の一樣乱数 r_p を作り、1.0 を減算して生成する。

表 1 FP 演算器の面積と動作周波数

Table 1 Evaluation of area and operation frequency of FP calculation units.

	Stage	Slices*	Mult.*	Block* RAM	Freq.* [MHz]
加減算器	6	663	-	-	134.6
乗算器	8	246	4	-	196.6
除算器	17	1,542	-	-	106.7
比較器	1	48	-	-	119.2
対数器	32	2,276	6	13	109.2

* XC2VP70-5 : 33,088 slices, 328 Multipliers and 328 BlockRAMs

5.6 FRM-UNIT における 1 サイクルの計算時間

FRM-UNIT の、1 サイクルの計算時間について述べる。5.1 節の計算において、(1)、(2)、(4) には $2N$ クロック、(3) には 42 クロック、(5) には 4 クロックかかる。(1)、(2)、(4) は、オーバラップして計算が進行する。 N が (3)+(5) の計算時間である $42+4=46$ 以下である場合、1 スレッド実行の時間で、2 スレッドのシミュレーションが実行できる。 $N \leq 46$ の場合、反応 1 サイクルの計算時間は $N+46$ クロックである。 N が 47 以上の生化学システムでは、 DS_A の分子数更新が終わっても、 DS_B への Reactants の出力が終わるまで、次のサイクルの計算を始めることができない。そのため、 $N \geq 47$ の場合は、反応 1 サイクルの計算に、ホスト PC へのデータ転送を行うかどうかを判定するための 2 クロックを含めて、 $2N+2$ クロックかかる。

N が大きい場合は、1 スレッド構成の実行が効果的であるが、Cao らが生化学システムのテストケースとして使用している *E.coli* の HSR モデルは $N=61$ であり⁹⁾、現在のところ、主に使用される生化学システムの規模は、 $N=50$ から $N=100$ 程度であると考えられる。この場合、空き時間と演算ユニットの占有時間の長さは同程度になるため、シミュレーションを 2 スレッド実行することにより、空き時間を埋める方法が有利である。一方、 N が小さいときは、3 スレッド以上のスレッド構成が考えられる。しかし、 N が小さい極小規模な生化学システムは、汎用プロセッサで高スループットな演算が可能であることも多く、マルチスレッド化による高速化は実用上大きな効果が得られないと考えられる。これらの理由から、実行スレッド数を 2 に固定した。主に計算対象となる生化学システムの規模が大きくなった場合は、スレッドが 1 の構成とし、シミュレーション 1 サイクルあたりの計算時間を短縮する方法が有効になると考えられる。

6. 評価

6.1 面積評価

実装した FRM-FPGA を、ISE8.1i で合成、配置配線した結果を表 2 に示す。対象 FPGA は ReCSiP-2 board に搭載の Virtex-II Pro (XC2VP70-5) を使用した。FRM-UNIT は演算ユニット (FU)、データユニット (DU)、コントローラユニット (CU) を各 1 つずつ持つ FRM 実行回路である。FRM-FPGA は FRM-UNIT を 3 つ持ち、計 6 回分のシミュレーションを同時に実行する。FRM-FPGA の面積は XC2VP70 の約 74% であり、PCI バスを介したホストとのイ

表 2 FRM-FPGA の面積評価
Table 2 Area evaluation of FRM-FPGA.

Module	Slices*	Mult.*	Block* RAM	Freq.* [MHz]
FU	7,360 (22.24%)	26	8	106.43
DU	992 (2.99%)	-	8	106.32
CU	60 (0.18%)	-	2	143.88
FRM-UNIT	8,166 (24.68%)	26	18	106.56
FRM-FPGA	24,496 (74.03%)	78	54	106.29

* XC2VP70-5 : 33,088 slices, 328 Multipliers and Block-RAMs

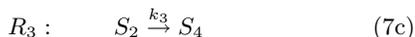
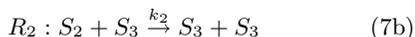
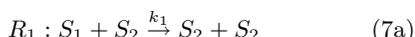
インタフェース部分を追加する余裕が残されている。

6.2 性能評価

6.2.1 ベンチマーク生化学システム

FRM-FPGA の計算性能を, RTL シミュレーションにより以下の 3 つの生化学システムで評価した。D は生化学システムの複雑さの指標となる値である⁹⁾。

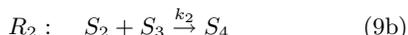
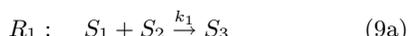
- (1) **Lotka System (LTS)**²³⁾: 4 種の分子により 4 種の反応が発生するシステム ($D = 3$)



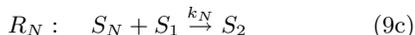
- (2) **Totally Independent System (TIS)**⁹⁾: 分子が壊変するシステム ($D = 1$)



- (3) **D = 4 System (D4S)**: 2 種類の分子から新しい分子が生成されるシステム ($D = 4$)



⋮



性能評価, 速度比較用に, FRM 実行プログラム FRM-SW を C++ を用いて実装した。FRM-SW は乱数生成アルゴリズムとして Mersenne-Twister²⁴⁾ を使用し, 実数の計算を単精度浮動小数点で実行する。

LTS は, 以下の初期値

$$\vec{k} = \{0.002, 0.01, 10.0, 10.0\}$$

$$\vec{X}(t_0) = \{100,000, 1,000, 1,000, 0\}$$

を設定し, X_1 を減少しない定数と定義すると, X_2, X_3 が振動する生化学システムである。このパラメータを設定した LTS で, 反応が 10 万回発生するまでの分子数の変化を, 図 9, 図 10, 図 11 に示す。図 9, 図 10

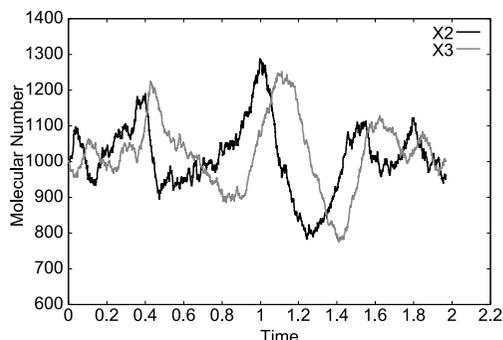


図 9 RTL シミュレーションによる LTS の出力 (DS_A)
Fig. 9 Output of LTS model (DS_A) based on RTL-simulation.

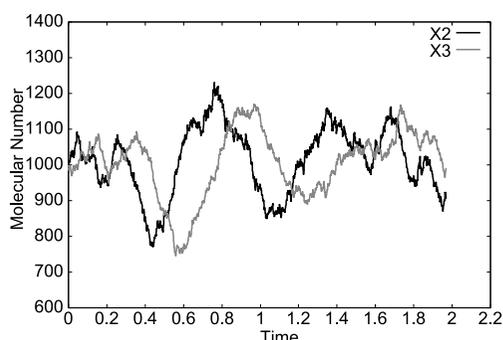


図 10 RTL シミュレーションによる LTS の出力 (DS_B)
Fig. 10 Output of LTS model (DS_B) based on RTL-simulation.

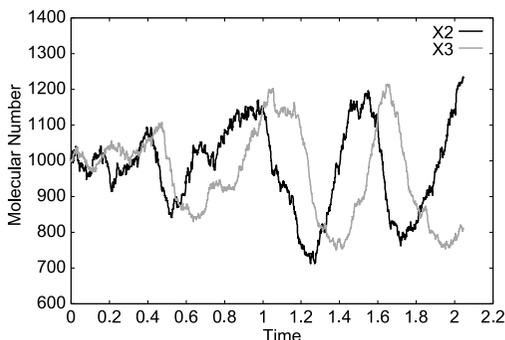


図 11 C++プログラムによる LTS の出力
Fig. 11 Output of LTS model (written in C++).

は, 1 つの FRM-UNIT から得られる 2 回の RTL シミュレーションの実行結果, 図 11 は FRM-SW による実行結果である。パラメータは相対的に記述されるため, LTS の時刻や反応定数には単位が存在しない。各 FRM-UNIT の DS_A と DS_B は, 共通の乱数生成モジュールから乱数を利用するが, 使用する乱数列が異なるため, 分子数の変化は異なる軌道を描く。これらは, Gillespie が示している LTS の実行結果²³⁾ と

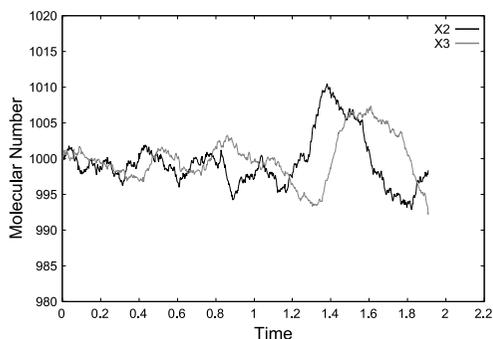


図 12 FRM-UNIT の RTL シミュレーション結果 1,000 回の平均分子数

Fig. 12 Average number of molecules after 1,000-times RTL-simulation of FRM-UNIT.

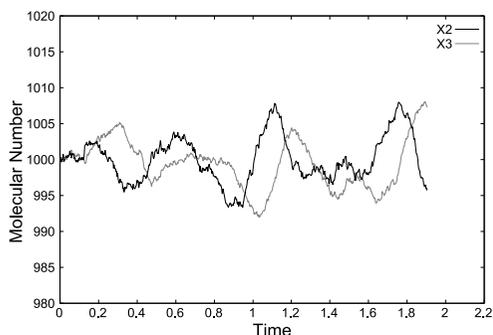


図 13 FRM-SW の実行結果 1,000 回の平均分子数

Fig. 13 Average number of molecules after 1,000-times simulation of FRM-SW.

同様に、初期値を中心として振動しており、妥当な結果が得られていると考えられる。確率モデルは 1 回のシミュレーション結果では一般的なシステムの挙動を得られないため、シミュレーションの精度を厳密に測定することは困難である。そのため、FRM-FPGA の RTL シミュレーションおよび FRM-SW をそれぞれ 1,000 回実行した結果について、各時刻の LTS の分子数の平均を求め、定常状態であることを確認した。その結果を図 12、図 13 に示す。これらの結果は、 $X_2 = 1,000$ 、 $X_3 = 1,000$ を中心に約 $\pm 1.5\%$ の範囲で定常状態を保っており、FRM-FPGA が妥当な結果を出力していることが分かる。LTS では、FRM-FPGA と FRM-SW の間に、大きな差異は認められなかった。

しかし、LTS は小規模かつ単純な生化学システムであるため、乱数生成アルゴリズムを含めて、異なるパラメータでの評価や、他のより大規模な生化学システムを用いた精度および結果の理論的検証が今後必要である。

6.2.2 性能比較

6.2.1 項で述べた FRM-SW と FRM-FPGA の計算

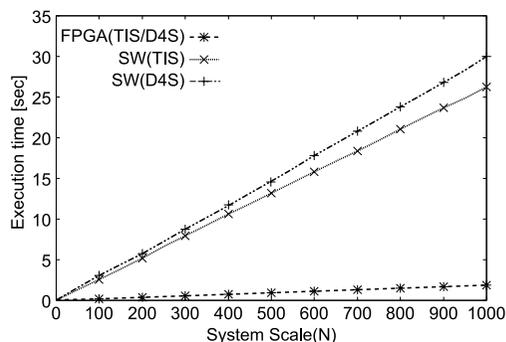


図 14 N と計算時間の比較

Fig. 14 N versus calculation time.

時間、スループットを評価した。FRM-FPGA は表 2 に示された動作周波数 (106.29 MHz) で動作すると仮定し、1 サイクルにかかるクロック数から、計算時間とスループットを計算した。FRM-SW は以下の環境で実行し、計算時間を計測した。

- Xeon 2.80 GHz 4.0 GB RAM
- Linux 2.4.31 gcc 3.3.5 (-O3)

TIS, D4S は、生化学システムで定義される反応の数 N の値を変えることで、システムの規模を変更することができる。図 14 に、 N を 1 から 1,000 まで変化させたときの、反応が 10 万回発生するまでの計算時間を示す。図 14 に示すように、FRM-FPGA、FRM-SW ともに、シミュレーションは N に比例した計算時間がかかっている。FRM-FPGA では、5.6 節で述べたように、シミュレーション 1 スレッドあたりの FU 占有時間が N に比例して増加するため、生化学システムの規模が大きくなるにつれて計算時間が長くなる。また、FRM-SW でも、生化学システムを構成する全反応の発生予測時間を毎サイクル計算する FRM の性質から、 N に比例した計算時間がかかる。評価に使用した生化学システムは FRM-SW の実行プロセッサ (Xeon) のキャッシュ内に収まる規模と考えられるが、図 14 に示すように、FRM-SW は FRM-FPGA と比較して N の増大にともなう計算時間の増加が大きい。この原因を明らかにするために、 $N = 10, 100, 1,000$ における TIS, D4S のシミュレーションを FRM-SW で実行したときのプロファイル結果を表 3 にまとめた。表 3 で示されたように、FRM-SW の実行時間においては対数の計算が占める割合が大きく、全体の実行時間も N に比例している。一方 FRM-FPGA は、5.4 節で述べた機構により、式 (1) の $\ln(1/r)$ の計算結果を 1 クロックサイクルごとに 1 つ得られること、各ユニット内の BlockRAM へ連続したデータの読み書きが実行されることから、 N が増加した際の

表 3 FRM-SW の計算時間の内訳

Table 3 Breakdown of calculation time for executing FRM-SW.

System	N	割合 [%]	時間 [sec]	処理
TIS*	10	42.18	0.1197	対数の計算 (log(x))
		19.24	0.0546	propensity 計算 (式 (5a))
		12.83	0.0364	τ の計算 (式 (1))
		25.76	0.0731	その他
	100	47.05	1.2519	対数の計算 (log(x))
		20.27	0.5392	propensity 計算 (式 (5a))
		11.63	0.3095	τ の計算 (式 (1))
		21.05	0.5601	その他
	1,000	49.37	12.9618	対数の計算 (log(x))
18.41		4.8324	propensity 計算 (式 (5a))	
10.36		2.7189	τ の計算 (式 (1))	
21.87		5.7413	その他	
D4S \diamond	10	47.73	0.1569	対数の計算 (log(x))
		22.85	0.0751	propensity 計算 (式 (5a))
		8.37	0.0275	τ の計算 (式 (1))
		21.05	0.0692	その他
	100	41.87	1.2735	対数の計算 (log(x))
		30.08	0.9149	propensity 計算 (式 (5a))
		8.66	0.2634	τ の計算 (式 (1))
		19.39	0.5897	その他
	1,000	41.12	12.3787	対数の計算 (log(x))
		32.28	9.7154	propensity 計算 (式 (5a))
		7.12	2.1432	τ の計算 (式 (1))
		19.48	5.8637	その他

表 4 計算時間の評価

Table 4 Evaluation of calculation time.

System	N	計算時間 [sec]		速度向上率 (SW/HW)
		SW	HW	
TIS*	10	0.2838	0.0527	5.3859
	100	2.6607	0.1900	14.0004
	1,000	26.2544	1.8835	13.9393
D4S \diamond	10	0.3287	0.0527	6.2395
	100	3.0415	0.1900	16.0044
	1,000	30.1010	1.8835	15.9816

* Totally Independent System $\diamond D = 4$ System

計算時間の増大が、FRM-SW に比べ小さく抑えられている。また、図 14 から、FRM-SW では生化学システムの形状も計算時間に影響を与えていることが分かる。TIS は確率モデルで定義可能な反応のうち最も単純な反応のみで定義された生化学システムであり、D4S は TIS に比べて propensity の計算に関わる乗算の回数が 1 回多い生化学システムである。そのため、表 3 に示されたように D4S では propensity の計算時間が長くなり、シミュレーション時間も長くなっている。一方、FRM-FPGA は、propensity を計算時間固定の FU を使用して求めるため、計算時間はシステム規模 N のみに比例し、各反応の形には依存しない。表 4 に、図 14 の $N = 10, 100, 1,000$ における

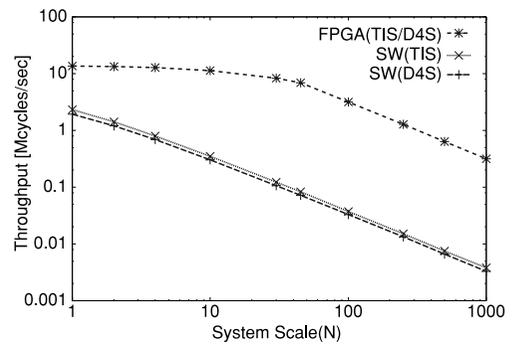


図 15 N とスループットの比較
Fig. 15 N versus throughput.

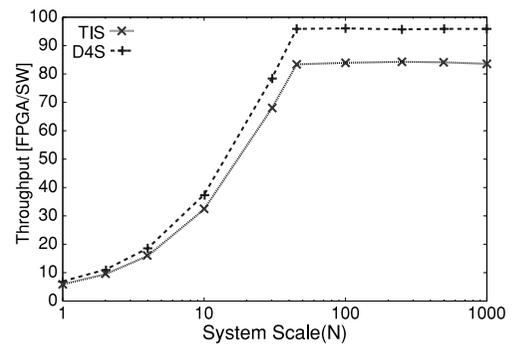


図 16 FRM-FPGA のスループット向上率
Fig. 16 Throughput improvement of FRM-FPGA.

FRM-FPGA と FRM-SW の計算時間の抜粋、および FRM-SW に対する FRM-FPGA の速度向上率を示す。表 4 から、 $N = 10$ のような小規模な生化学システムでは、1 サイクルの計算時間のうち 5.6 節で述べた τ の計算時間が支配的になるため FRM-FPGA の速度向上率は小さいが、 $N = 100$ 以上の生化学システムでは、FRM-SW と比べて TIS では約 14 倍、D4S では約 16 倍の計算速度向上が得られることが確認された。

FRM-UNIT のスレッド 1 つあたりのスループット T_h [Mcycles/sec] は、FPGA の動作周波数を F [MHz] とすると、式 (10) で計算できる。

$$T_h = \begin{cases} F/(N + 46) & (N \leq 46) \\ F/(2N + 2) & (N \geq 47) \end{cases} \quad (10)$$

FRM-FPGA は 1 回のシミュレーション実行時間で 6 スレッドの計算を同時に実行するため、FRM-FPGA のスループットは $6T_h$ [Mcycles/sec] となる。計算で求めた FRM-FPGA のスループットと FRM-SW のスループット計測結果について、システム規模との関係を図 15 に、FRM-SW と比較したときの FRM-FPGA のスループット向上率を図 16 に示す。図 16 に示されたように、 N が小さい小規模な生化学システムで

表 5 スループットの評価
Table 5 Throughput evaluation.

System	N	スループット [Mcycles/sec]		スループット 向上率 (HW/SW)
		SW	HW	
TIS*	10	0.3524	11.3885	32.3146
	100	0.0376	3.1572	83.9989
	1,000	0.0038	0.3186	83.6324
D4S ^o	10	0.3042	11.3885	37.4366
	100	0.0329	3.1572	96.0263
	1,000	0.0033	0.3186	95.8888

* Totally Independent System ^o D = 4 System

は演算ユニットの空きが大きく、並列実行による利得は小さいが、パイプラインがすべて埋まる $N \geq 47$ 以上の生化学システムにおいて、FRM-FPGA は FRM-SW と比べてスループットの向上を実現できる。また、図 15 および図 16 から $N = 10, 100, 1,000$ のときのスループットとスループット向上率の抜粋を表 5 に示す。表 5 から、FRM-FPGA は Xeon 2.80 GHz 上でのソフトウェア実行と比べ、TIS においては約 83 倍、D4S においては約 95 倍を超えるスループットが得られることが確認された。

6.2.3 ホスト PC との通信に関する議論

ReCSiP2-board は DMA 転送用のインタフェースが開発中であるため、転送時間に関する評価はなされていない。FRM-FPGA を実行する場合に生じるホスト PC と ReCSiP2-board のデータ転送には、ホスト PC から FRM-FPGA への初期値設定と実行開始信号の入力、および FRM-FPGA からホスト PC への結果の出力がある。このうち、入力方向の通信はシミュレーション開始時に 1 回だけ実行され転送量も少ないが、出力方向の通信はシミュレーションの進展とともに逐次発生するため、PCI バスの通信能力がシミュレーション速度に影響を与える場合が考えられる。5.1 節で述べたように、シミュレーション結果として生化学システムの時刻 t と全分子数をホスト PC に出力すると仮定すると、転送量 T_r [MB/sec] は、以下の式 (11) で表される。

$$T_r = \{T_h \times (4 \times M + 4)\} / d \times 6 \quad (11)$$

ここで、 M は生化学システムで定義された分子の数、 d は転送するサイクル間隔を表す。分子数は 32 bit (= 4 Byte) 整数で、システム時刻は 32 bit 単精度浮動小数点実数形式に丸めて転送される。 $M = 44$ 以上のシステムで、 $d = 1$ の場合の T_r は約 7,474 [MB/sec] になる。これは、64 bit/66 MHz の PCI バスの最大データ転送速度である 533 [MB/sec] を大きく超えるため、すべてのデータを転送することは現実的ではな

い。確率モデルの性質上、1 サイクルで数が増える分子はたかだか 4 種類であり変化量も小さいため、1 サイクルごと ($d = 1$) に分子数をすべてホスト PC に転送する必要はない。転送量は、必要な分子のみを転送する方法、または、大きい d を選択することで削減する必要がある。

7. まとめと今後の展開

First Reaction Method による確率モデル生化学シミュレータ回路を FPGA 上に実装し、評価を行った。固定された演算を膨大な回数繰り返すアルゴリズムの性質を利用して、パイプライン化された演算ユニットに間断なくデータを投入し回路の利用効率の向上を図っている。また FPGA を用いた他の実装では実行できない大規模な生化学システムまでシミュレーションが可能であり、回路規模に合わせて生化学システムの対応規模や並列実行数を増やせる構造を持っている。Xeon 2.80 GHz によるソフトウェア実行と比較した結果、ベンチマークとして使用した生化学システムである TIS において約 83 倍、D4S において約 95 倍の性能向上が可能であることを確認した。

複数のシミュレーションの並列動作により高いスループットが実現できることが明らかになったため、計算効率改良された SSA である Next Reaction Method⁸⁾ について、FPGA を用いた実行環境の構築を行う予定である。

謝辞 本研究は、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。また、本研究は東京大学大規模集積システム設計教育研究センターを通じ、ケイデンス株式会社の協力で行われた。

参考文献

- 1) Kitano, H.: Computational systems biology, *Nature*, Vol.420, pp.206–210 (2002).
- 2) Schaff, J., et al.: A General Computational Framework for Modeling Cellular Structure and Function, *Biophysical Journal*, Vol.73, pp.1135–1146 (1997).
- 3) Tomita, M., et al.: E-Cell: Software environment for whole-cell simulation, *Bioinformatics*, Vol.15, No.1, pp.72–84 (1999).
- 4) Gillespie, D.T.: A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions, *Journal of Computational Physics*, Vol.22, pp.403–434 (1976).
- 5) Osana, Y., et al.: Implementation of ReCSiP:

- A Reconfigurable Cell SIMulation Platform, *The 13th International Conference on Field Programmable Logic and Applications*, Lecture Notes in Computer Science, Vol.2778, pp.766–775, Springer (2003).
- 6) Kierzek, A.M.: STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm, *Bioinformatics*, Vol.18, No.3, pp.470–481 (2002).
 - 7) Schwehm, M.: Parallel Stochastic Simulation of Whole-Cell Models, *Proc. 2nd International Conference on Systems Biology*, pp.333–341 (2001).
 - 8) Gibson, M.A. and Bruck, J.: 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).
 - 9) Cao, Y., 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).
 - 10) Takahashi, K., et al.: A multi-algorithm, multi-timescale method for cell simulation, *Bioinformatics*, Vol.20, No.4, pp.538–546 (2004).
 - 11) Salis, H., et al.: Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions, *Journal of Chemical Physics*, Vol.122, pp.1–13 (2005).
 - 12) Lok, L.: The need for speed in stochastic simulation, *Nature Biotechnology*, Vol.22, No.8, pp.964–965 (2004).
 - 13) Osana, Y., et al.: An FPGA-Based Acceleration Method for Metabolic Simulation, *IEICE Trans. Inf. & Syst.*, Vol.E87-D, No.8, pp.2029–2037 (2004).
 - 14) 福島知紀ほか：細胞系譜構築システムのFPGA利用による高速化，第2回リコンフィギャラブルシステム研究会-14 (2003).
 - 15) FDK 株式会社：物理乱数生成器 RPG100/100B データシート Rev06.
 - 16) Salwinski, L., et al.: In silico simulation of biological network dynamics, *Nature Biotechnology*, Vol.22, No.8, pp.1017–1019 (2004).
 - 17) Keane, J.F., et al.: A Compiled Accelerator for Biological Cell Signaling Simulations, *The 12th Int. Symp. on Field-Programmable Gate Arrays (FPGA)*, pp.233–241 (2004).
 - 18) Yoshimi, M., et al.: Stochastic Simulation for Biochemical Reactions on FPGA, *The 14th International Conference on Field Programmable Logic and Applications*, Lecture Notes in Computer Science, Vol.3203, pp.105–114, Springer (2004).
 - 19) Oda, K., et al.: A comprehensive pathway map of epidermal growth factor receptor signaling, *Molecular Systems Biology*, Vol.1, pp.1–17 (2005).
 - 20) 黒川恭一，梶崎浩嗣：Mersenne Twister のFPGAによる実装，防衛大学校理工学研究報告，Vol.40, No.2, pp.15–21 (2003).
 - 21) Kohlbrenner, P., et al.: An Embedded True Random Number Generator for FPGAs, *The 12th Int. Symp. on Field-Programmable Gate Arrays (FPGA)*, pp.71–78 (2004).
 - 22) 吉見真聡ほか：FPGAを用いた科学技術計算向け浮動小数点算術演算器の設計，第26回バルテノン研究会，Vol.1, No.7, pp.49–55 (2005).
 - 23) Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions, *The Journal of Physical Chemistry*, Vol.81, No.25, pp.2340–2461 (1977).
 - 24) Matsumoto, M., et al.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, *ACM Trans. Modeling and Computer Simulation*, Vol.8, No.1, pp.3–30 (1998).

(平成18年7月21日受付)

(平成18年11月14日採録)



吉見 真聡

2006年慶應義塾大学大学院理工学研究科修士課程修了。現在同大学院博士課程在学中。2006年より日本学術振興会特別研究員。リコンフィギャラブルシステムの研究に従事。電子情報通信学会会員。



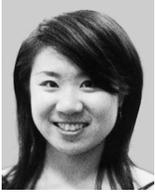
長名 保範

2006年慶應義塾大学大学院理工学研究科博士課程修了。工学博士。現在慶應義塾大学理工学部生命情報学科博士研究員。リコンフィギャラブルシステム、システムバイオロジー、比較ゲノム等の研究に従事。



岩岡 洋

2005年慶應義塾大学理工学部情報工学科卒業。現在同大学院理工学研究科修士課程在学中。リコンフィギャラブルシステムの研究に従事。



西川 由理 (学生会員)

2006年慶應義塾大学工学部情報工学科卒業。現在同大学大学院理工学研究科修士課程在学中。リコンフィギャラブルシステムの研究に従事。



小嶋 利紀

2006年慶應義塾大学工学部情報工学科卒業。現在同大学大学院理工学研究科修士課程在学中。リコンフィギャラブルシステムの研究に従事。



柴田裕一郎 (正会員)

2001年慶應義塾大学大学院理工学研究科後期博士課程修了。工学博士。現在長崎大学工学部情報システム工学科講師。リコンフィギャラブルシステムの研究に従事。電子情報通信学会, IEEE-CS 各会員。



岩永 直樹

2005年長崎大学工学部情報システム工学科卒業。現在同大学大学院生産科学研究科修士課程在学中。リコンフィギャラブルシステムの研究に従事。



舟橋 啓 (正会員)

2000年慶應義塾大学大学院理工学研究科後期博士課程修了。工学博士。現在科学技術振興機構北野共生システムプロジェクト研究員。システム生物学, 計算生物学の研究に従事。IEEE, IEEE-CS 各会員。



広井 賀子

2002年東京大学大学院医学系研究科博士課程修了。医学博士。現在科学技術振興機構北野共生システムプロジェクト研究員。システム生物学, 計算生物学, 分子生物学の研究に従事。



北野 宏明

1991年京都大学博士号(工学博士)取得。現在ソニーコンピュータサイエンス研究所取締役副所長。特定非営利活動法人システム・バイオロジー研究機構会長。慶應義塾大学大学院理工学研究科客員教授。システムバイオロジー・人工知能・ロボティクス等の研究に従事。



天野 英晴 (正会員)

1986年慶應義塾大学大学院理工学研究科博士課程修了。工学博士。現在同大学理工学部情報工学科教授。計算機アーキテクチャ, リコンフィギャラブルシステムの研究に従事。電子情報通信学会, IEEE 各会員。