社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＳＬＤＭ－123（19）
2006／1／18

FPGAを用いた生化学シミュレータReCSiPにおける数値積分機構の性能改善手法

西川 由理† 長名 保範†† 吉見 真聡†† 岩岡 洋†† 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††

† 慶應義塾大学理工学部
†† 慶應義塾大学大学院理工学研究科

〒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 近年のバイオインフォマティクスにおいて、計算機上での生化学シミュレーションは、細胞内の生化学反応を解析する非常に有効な方法である。しかしシミュレーション対象となる系の大規模化が進むにつれ、高スループットのシミュレータの開発が急務となる。ReCSiPは、FPGAを用いて高速なシミュレーションを実現するシステムとして開発されており、本研究ではその反応速度式を解くモジュールと数値積分を行うモジュールを並列動作させることによってパイプライン稼働率の向上を図った。本研究報告ではその方法論の提案、ReCSiPへの実装および回路面積-速度効率の評価を示し、その結果について検討する。従来方式と比べ1.29倍の面積で、最高1.96倍のスループットを実現した。

キーワード 生化学シミュレータ, 数値積分, Euler法, パイプライン稼働率, FPGA

A Performance Improvement Strategy for Numerical Integration on an FPGA-Based Biochemical Simulator ReCSiP

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

† Faculty of Science and Technology, Keio University
†† Graduate School 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 Computational simulation is a highly effective solution for cellular analyses in recent bioinformatics. As the scale of analysis objects are rapidly expanding, however, it is a prime task to develop a simulator with high throughput. ReCSiP, an FPGA-based biochemical simulator, is being exploited for this necessity. Further speed-up of the simulator is enhanced with an improvement of its pipelining efficiency, by a parallel operation of a rate-law solver module and a numerical integration module. This research report discusses the methodology of enhancing the performance of ReCSiP, its implementation and the evaluation result. Speed-up of 1.96x in maximum is achieved with an area increase of 29% compared to the conventional implementation.

Key words Biochemical simulator, Numerical integration, Euler's Method, Pipelining efficiency, FPGA

1. はじめに

近年、細胞内の代謝システムを数理的にモデリングする試みが盛んになっている。最近では E-Cell [1] や Virtual Cell [2] など、細胞全体のシミュレーションが可能なシステムも開発され、シミュレーションの対象となる系は大規模化、複雑化の一途を辿っている。

しかし系が拡大するに従い、長大な計算時間を要することが問題となる。代謝系シミュレーションにおいては個々の生化学反応の独立性が高いため、PC クラスタやグリッド計算機を用いて並列的に演算を実行するのが一般的である。しかし時間変化にともない物質の濃度も変化するため、反応間に依存関係が生じ、並列動作するプロセッサ間での通信や同期が必要となる。これは PC クラスタやグリッド計算機の性能にとって致命的である。

このような問題を解決するため、本研究グループでは ReCSiP という FPGA を用いた汎用生化学シミュレータの構築を目指している [3] [4]。システムは FPGA を搭載した PCI カードで構成され、汎用の PC に設置することでマイクロプロセッサ数十台に匹敵する計算速度が確認されている。

従来は、Euler 法による数値積分ソルバが実装されていたが、その中心的な高速演算処理部である Solver Core の性能を十分に引き出せないことが問題となっていた [5]。したがって今回はソルバの Solver Core の稼働率を向上させ、高スループットを得る実装を試みた。本稿ではその手法について述べるとともに、細胞の実モデルとして Goldbeter らによる Minimal Mitotic Oscillator [6] と呼ばれるモデルを拡張後のソルバに実装、評価した結果の考察を行う。

2. 微分方程式ソルバの構成

生化学反応に関わる物質の濃度は微分方程式で記述され、その濃度変化を追うことで反応の解析を行う。そのため ReCSiP には数値積分を行うモジュール (以後 Solver) が構成、実装されている。

ReCSiP は、SBML [7] で記述された生化学モデルから反応とパラメータのリストを抽出すると、リストに応じて系に合わせた回路 (Solver の組合せ) を生成し、パイプラインを静的にスケジューリングする [8]。本稿ではこの Solver の性能向上に着目する。

2.1 Solver の構成

図 1 に Euler 法で積分を行う場合の Solver の構成を示す。反応速度式を解く Solver Core と積分を行うロジック (以後 積分モジュール) で構成され、これらを切り離すことで柔軟性と拡張性を確保している。Solver Core は表 1 に示すような反応速度関数ライブラリを持ち、系に最適な関数がロードされて反応速度を計算する。また積分モジュールには表 2 に示す [X] RAM, k RAM, S RAM, d[X] RAM というデータ用のメモリブロックと、反応ごとに各メモリのどのメモリ番地を参照するかを記述した Pathway RAM が配置されている。

表 2 各モジュールの役割

モジュール	役割
Solver Core	反応速度の計算
Pathway RAM	反応経路を記載したポインタを保持
[X] RAM	各分子の濃度を保持
d[X] RAM	濃度の時間変化を保持
k RAM	反応速度定数を保持
S RAM	各反応で生成または消費される分子数を保持

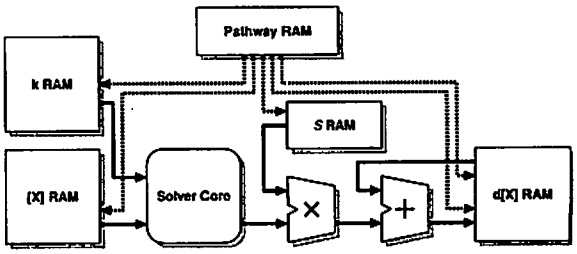

図 1 Solver の構成

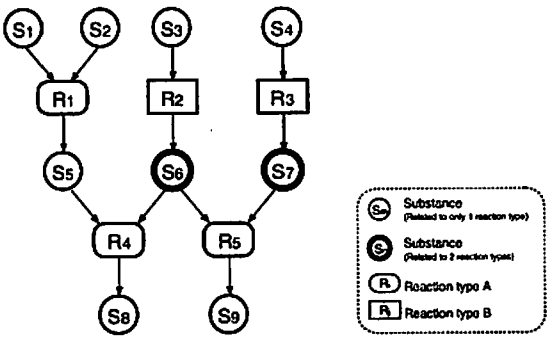

図 2 反応系の例

反応速度式は、関与する物質の濃度の関数として表される。濃度は積分モジュールの [X] RAM に、反応速度定数は k RAM というメモリブロックにそれぞれ格納されており、Solver Core は 1 サイクル毎にこれらのブロックから値を読み出して反応速度を求める。

アルゴリズムの詳細は次章で述べるが、Euler 法では反応速度が得られると次時刻の濃度が反復計算できる。図 1 に示すように、一回の反応で生成または消費される分子の数 (stoichiometry) を S RAM から読み出して濃度の時間変化を d[X] RAM に書きこむと、それが [X] RAM に加算されて次時刻の濃度となる。

2.2 Solver Set の構成

一つの系には複数の反応が存在するため、複数の Solver を組み合わせた Solver Set を構成して並列処理を行う必要がある。例えば図 2 のような反応経路を考える。これは物質 S_1〜S_9 と、反応 R_1〜R_5 で構成されているが、R_1, R_4, R_5 と $R_2, R3$ はそれぞれ同様の反応機構となっている。この場合、ReCSiP は 2 種類の Solver を FPGA 上に構成して、それぞれの反応速度計算および積分が並列に行われる。

なお、図 2 の物質 S_6 のように 複数の反応経路で同一物質の

表 1　Solver Core に実装されている反応速度関数の例

Name (abbr.)	Formula
Irreversible Simple Michaelis-Menten (UUI)	$v = \frac{V_m S}{K_m + S}$
Competitive Inhibition (Irreversible) (UCII)	$v = \frac{VS/K_m}{1 + S/K_m + I/K_i}$
Competitive Inhibition (Reversible) (UCIR)	$v = \frac{V_f S/K_{mS} - V_r P/K_{mP}}{1 + S/K_{mS} + P/K_{mP} + I/K_i}$

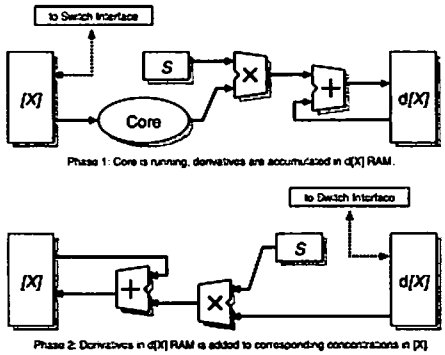

図 3　Euler 法による積分モジュールの動作

濃度データを共有する場合も存在する。そのため、Solver 間で分子の濃度や濃度変化量などを双方向に転送する通信機構 (スイッチ) を設けている。

2.3　Solver の動作

Euler 法とはテイラー展開の一次近似式で、常微分方程式の解法アルゴリズムのうち最も簡単なものである。時刻 t における濃度を $X(t)$ としたとき、その一次導関数

$$v = f(X(t))$$

および初期値 $X(t_0)$ が既知ならば、次の時刻の濃度を

$$X(t + \Delta t) = X(t) + f(X(t))\Delta t$$

と表すことができ、反復的に濃度の時間変化が求まる。

この解法で必要な数値演算は、各タイムステップにおける反応速度の計算、およびそれと時刻の刻み幅との積算である。Euler 法による積分を行う場合のデータパスを図 3 に示す。Solver は、タイムステップごとに 2 つのフェーズの処理を繰り返す。

最初のフェーズでは Solver Core を用いて時刻 t の反応速度を得ると、それと時刻の刻み幅と stoichiometry を乗算してタイムステップ間の濃度変化量が求められ、d[X] RAM に格納される。2 番目のフェーズでは、d[X] RAM の内容を [X] RAM に加算し、次の時刻の濃度を求める。以上の繰り返しで時刻を進め、[X] RAM 内のデータを更新する。これらの値は Solver 間の通信機構を用いて、必要に応じて取り出すことができ、時系列でのモデルの挙動を見ることができる。

3.　Solver の性能向上手法

ReCSiP の中心的な計算モジュールである Solver Core の持つ理論ピーク性能は、汎用マイクロプロセッサの数十台分に匹敵する [9]。しかし従来方式の実装による実効性能は、その 1/3〜1/2 程度に留まると見込まれていた。その理由として以下が挙げられている [5]。

（1）　前章の図 3 における後半フェーズで [X] RAM から d[X] RAM の値の加算時に、パイプラインの深さ分の間隔を空ける必要が生じ、反応数もしくは分子種が少ない場合にはデータの投入より待ち時間の方が長くなり、十分な性能が発揮されない

（2）　後半フェーズの処理を行っている間は Solver Core がまったく稼働しない

本研究では、後者の問題点を解決するための実装を行った。方針として、ハードウェアを追加することにより、前半フェーズと後半フェーズのマルチスレッド化を図った。以下に詳細を述べる。

図 3 より、Euler 法による積分では乗算器と加算器がどちらのフェーズでも 1 つずつ利用されている。したがってこれらの演算器を、前半と後半フェーズそれぞれに専用のものとして与えることで、両方の処理が同時に行うことが可能になる。これにより、Solver Core の稼働率をほぼ 100% に引き上げることができ、従来の約 2 倍の性能向上を見込むことができる。

実装した Solver Set の構成を図 4 に示す。反応を記述するポインタ配列 Pathway RAM はコントローラを除く全てのモジュールに接続され、それぞれの動作を管理する。今回の実装では dual port に拡張され、前半フェーズと後半フェーズはそれぞれ別の経路もしくは命令を読み出せる。

BRAM1、2 は同一のモジュールで、データを保持する [X]

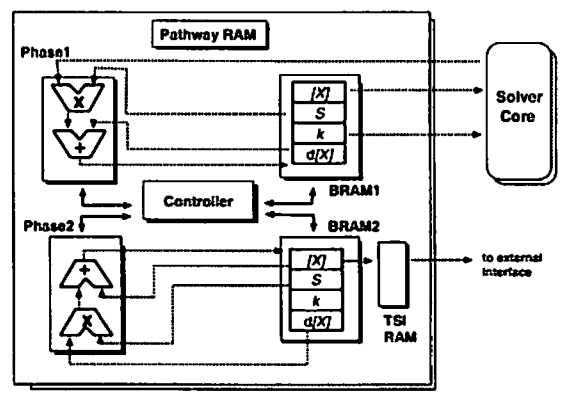

図 4　2 フェーズ並列動作する Solver

RAM, d[X] RAM, k RAM, および S RAM により成る。そして一方の BRAM が Solver Core にアクセスして反応速度を求める間、もう一方は d[X] と X の加算を行い、次の時刻の濃度を求める。図 4 に示す例では、BRAM1 が前者、BRAM2 が後者の処理を行っている。新たな濃度が求まると TSI RAM というメモリに格納され、外部インタフェースはここから値を読み出せる。

中心に位置するコントローラは、前半フェーズと後半フェーズの切替えを行い、これに応じて 2 つの BRAM の動作が入れ替わる。切替えの判断を与えるのが Phase1、2 モジュールであり、これらはそれぞれのフェーズごとに、Pathway RAM にて静的にスケジュールされたパイプライン動作を監視するとともに、内蔵された乗算器と加算器で演算処理を行う。コントローラは両者の動作終了を確認して信号を切替える。以上のように構成すると、前半フェーズの処理に m クロック、後半フェーズに n クロック要する場合、従来方式に比べ

$$2 \times \frac{m+n}{m+n+|m-n|} \qquad (1)$$

倍のスループットを得ることができる。

4. 細胞モデルシミュレーションの例

4.1 Minimal Mitotic Oscillator モデル

拡張後の積分モジュールの動作確認のため、細胞の実モデルのシミュレーションを行った。対象として、両生類の受精卵細胞の有糸分裂周期に寄与する蛋白質の周期的な挙動に関する (図 5)、1991 年に Goldbeter らにより提案された Minimal Mitotic Oscillator (以下 Cellcycle モデル) を用いた。

図 5 に示す Cyclin とは有糸分裂を誘発するシグナル伝達物質であり、この濃度の周期的変化に伴い細胞分裂が生じる。このモデルは図に示す 7 つの反応で記述することができる。Cyclin が生成されると CDC-2 キナーゼ (図中 M) を活性化する物質を生成する。さらにそれが Cyclin 分解酵素 (図中 X) が活性化される。したがって Cyclin 濃度が増すと負のフィードバックがかかるため、各物質の濃度は周期的に変化する。

4.2 Cellcycle モデルの実装と動作

図 6 に構成を示す。積分モジュールは前章で述べたものを用いた。また Cellcycle モデルの 7 つの反応機構は 3 種類に分類されるため、3 つの Solver より成る Solver Set を構成した。さらに Solver 間で共有する分子濃度の BRAM1、BRAM2 同士での転送を実現するため、スイッチを 2 つ設けた。

各 Solver の Pathway RAM にそれぞれ反応経路を記述し、BRAM1 と BRAM2 には同じ初期値を与えてシミュレーションを行った。図 7 に示す動作結果より、Cyclin 濃度の変化に従って、CDC-2 キナーゼおよび Cyclin 分解酵素の濃度がある時間差を持って周期的に変化する様子がわかる。また図 7 (a)、(b) より、拡張後の Solver に BRAM1 と BRAM2 モジュールがそれぞれ同一の動作をしていることが確認できる。

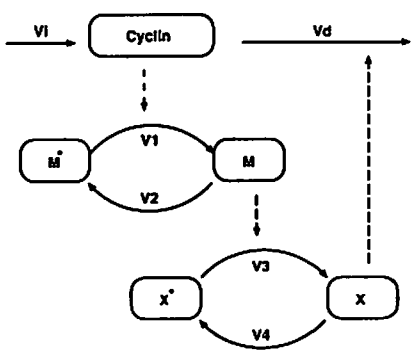

図 5 Mitotic Cellcycle モデル

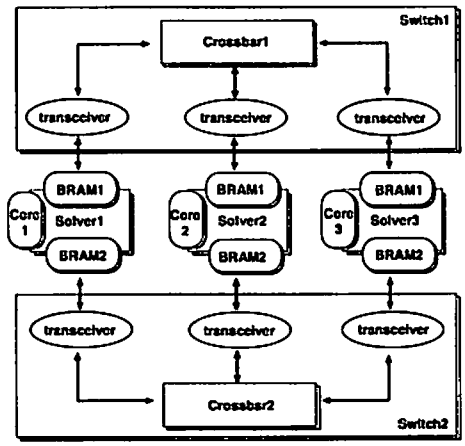

図 6 Cellcycle モデルの構成

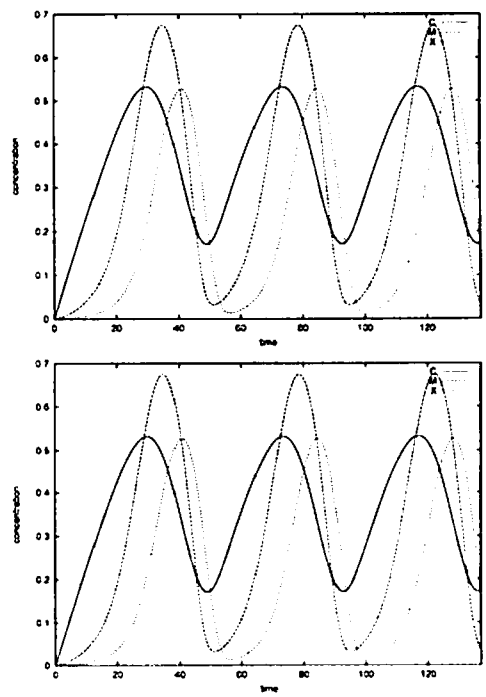

図 7 Cellcycle の実行結果
上段: (a) BRAM1 の出力、下段: (b) BRAM2 の出力

5. 実装と評価

5.1 Solver の単独評価

パイプライン稼働率を向上させたシステムを Verilog-HDL で記述し、Xilinx XC2VP70 FF1517-5 をターゲットデバイスとして、ISE 7.1i を用い論理合成と配置配線を行った。また性能比較のため、同ターゲットデバイスに対し従来方式による実装の論理合成と配置配線を行った。その面積に関する評価結果を表 3 に、処理性能に関するものを表 5 に示す。また表 4 には積分モジュールの各コンポーネントの面積を示す。なおここに示す Solver Core の面積は、現段階で実装が完了しているものの平均スライス数である [9]。

今回の拡張により、積分モジュール単独の面積は従来方式の 2.19 倍となった。しかし、このうち追加したハードウェア (浮動小数点加算器、乗算器および 4 つの Block RAM) による面積増がおよそ 1200〜1300 スライス程度と見込まれている。また Solver Core の面積を考慮すると、Solver 全体としては約 1.29 倍の面積増となる。

速度面では、表 5 の結果より従来方式の 97% の動作周波数を維持することができた。この上で、Solver の性能検証のためシンプルな単振動モデルを Pathway RAM にマッピングし、実効スループットを比較した。その結果、最大 1.96 倍のスループットが実現可能なことが確認され、良好な面積対性能比を実現できた。

表 3 Solver の面積比較

	Slices (Used %)		
	積分モジュール	Solver Core	Solver
Euler (拡張前)	1363 (4.1%)	4185 (12.6%)	5548 (16.3%)
Euler (拡張後)	2993 (9.0%)	〃	7178 (21.7%)

表 4 積分モジュールの各コンポーネントの面積

	Slices (Used %)	Numbers
Phase1	874 (2.6%)	1
Phase2	939 (2.8%)	1
Block RAM	406 (1.2%)	2
Pathway RAM	239 (0.7%)	1
TSI RAM	167 (0.5%)	1
Controller	4 (0.0%)	1

Numbers = 積分モジュールに用いる個数

表 5 Solver の動作周波数とスループット

システム	Euler (oscillator model)	
	MHz	MOps
Pentium4 (g++ -O3)	2600	3.21
1 Euler Solver (拡張前)	87.4	41.9
1 Euler Solver (拡張後)	85.7	82.0
4 Euler Solvers (拡張後)	79.3	303

MHz = システム動作周波数
MOps = 1 秒あたりの反応数

またマイクロプロセッサとの性能比較を行った。対象として、入力ファイルから反応経路を読み込み Euler 法で積分を実行する C++プログラムを用い、単振動モデルの計算時間を計測した。表 5 より、Solver 1 個はマイクロプロセッサ比にして単位時間あたり約 25.5 倍の反応数を計算できることが判明した。さらに表 3 の結果を勘案すると、ReCSiP ボードには最大 4 個の Solver を搭載できる。その場合、マイクロプロセッサと比べ最大 94.4 倍のスループットを得られる計算となる。

5.2 Cellcycle モデルの性能評価

複数個の Solver を利用した細胞シミュレーション例として、図 6 に示した構成で Cellcycle モデルを実装した。さらに性能比較のため、従来方式の積分モジュールによる実装の評価を行った。それぞれの面積と速度の評価結果を表 6 に示す。

今回の拡張で面積が約 1.6 倍となったが、動作周波数は 99%を維持することができた。これにより単位時間あたり処理できる反応数が 1.76 倍となり、良好な結果を得た。

しかし Cellcycle モデルを前述の C++プログラムにロードし計算時間を測定したところ、今回実装した Solver の単位時間あたり処理できる反応数はマイクロプロセッサの約 1.97 倍に留まった。

この結果より、Cellcycle モデルのように小規模な反応系では Solver Core の性能を十分に反映できないことが分かる。表 7 にモデルをスケジューリングした例を示すが、現実装ではデータハザードの回避のため、浮動小数点加算器を使用する際にパイプラインピッチ分だけ NOP を挟む必要が生じてしまい、パイプラインにデータが投入されない時間が長くなることが主要な原因である。

Cellcycle モデルは Solver のメモリ資源をわずか 2.4% しか使

表 6 Cellcycle モデルの性能

	Slices	MHz	Mreactions/s
Pentium4 (g++ -O3)	-	2600	3.88
Euler (拡張前)	8832	84.3	4.34
Euler (拡張後)	14142	83.5	7.64

表 7 Pathway RAM の内容

	Solver #0	Solver #1	Solver #2
0	R1	$0.2M \rightarrow k$	NOP4
1	R2	$0.1X \rightarrow k$	R4(1/4)
2	END	NOP2	R4(2/4)
3	$C+ = C$ (R1)	R5	R4(3/4)
4	NOP18	R6	R4(4/4)
5	$C+ = C$ (R2)	R7	END
6	NOP18	R3	NOP39
7	$C+ = C$ (R3)	END	END
8	END	$M+ = M$ (R6)	
9		$X+ = X$ (R5)	
10		NOP17	
11		$M+ = M$ (R4)	
12		$X+ = X$ (R7)	
13		NOP18	
14		END	

用せず、同モデルを 30〜40 個ほど複製して高速なシミュレーションを行う余地が残されている。メモリ容量の許す限り反応を多数記述もしくは複製したシミュレーションを行えば、パイプラインの空き時間が相対的に小さくなり、現実装のピーク性能を最大限に引き出すことができる。このような反応系の複製により、ReCSiP の特性を利用して多くのソフトウェアによるシミュレータが苦手とする広いパラメータ空間スキャンの高速化が可能になる [5]。

しかし今回、小さなモデルに対しても可能な限りの高スループットを目指すにあたっては、新たなハードウェアの構成も検討する必要性が示唆された。本研究報告の最後にて、Solver の新たな構成案を述べる。

6. まとめと今後の課題

本稿では、ReCSiP の高速演算処理部である Solver Core を常時稼働させる方法について述べた。実装および評価の結果、拡張前と比べ最大約 1.96 倍、マイクロプロセッサ比では最大約 25.5 倍のスループットが得られた。また実際の細胞モデルを用いた動作検証と性能測定を行い、拡張後も Solver が正確に動作し、従来の 1.76 倍のスループットが引き出されることを確認した。

しかし本稿に示した実装では、小さなモデルのシミュレーション時に、浮動小数点加算器のパイプラインピッチがボトルネックになることが判明した。したがってこの処理時間を考慮したスケジューリングを行うことが Solver Core の持つ性能を最大に引き出す方法だと考えられる。ところが、現在は積分モジュール内のメモリが Solver Core に直結する構造となっており、その間の通信路が固定されているため、反応系の最適化が困難な状態である。今後はこれを分離してスイッチで結合することにより、メモリアクセスの自由度を高め、メモリの入出力バンド幅を有効に利用できる構造に移行する予定である。同時に、Euler 法による実装を拡張し、解の精度がより優れた Runge-Kutta 法による積分処理ユニットの構築を目指す。

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ものです。また、本研究のハードウェア設計は、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し、ケイデンス株式会社の協力で行われています。

文　献

[1] Masaru Tomita, *et al.* E-Cell: software environment for whole-cell simulation. *Bioinformatics*, Vol. 15, No. 1, pp. 72–84, Jan. 1999.

[2] J. Schaff, *et al.* A general computational framework for modeling cellular struc ture and function. *Biophysical Journal*, Vol. 73, pp. 1135–1146, Sep. 1997.

[3] Y.Osana, T.Fukushima, M.Yoshimi, and H.Amano. An FPGA-based acceleration method for metabolic simulation. *IEICE Trans. on Information and Systems*, Vol. E87-D, No. 8, pp. 2029–2037, Aug. 2004.

[4] ReCSiP Project Team: "ReCSiP project frontpage". http://recsip.org.

[5] 長名保範, 吉見真聡, 岩岡洋, 小鴇利紀, 西川由理, 舟橋啓, 広井賀子, 柴田裕一郎, 岩永直樹, 北野宏明, 天野英晴. FPGA を用いた生化学シミュレータ ReCSiP のシミュレーション制御機構. RECONF2005 39, 電子情報通信学会, Sep. 2005.

[6] Albert Goldbeter. A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. In *Proceedings of the National Academy of Sciences*, Vol. 88, pp. 9107–9111, Oct. 1991.

[7] M. Hucka, *et al.* The systems biology markup language(SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, Vol. 19, No. 4, pp. 524–531, Mar. 2003.

[8] 岩岡洋, 長名保範, 吉見真聡, 小鴇利紀, 西川由理, 舟橋啓, 広井賀子, 柴田裕一郎, 岩永直樹, 北野宏明, 天野英晴. FPGA を用いた生化学シミュレータ向け SBML 処理系の構築. RECONF2005 39, 電子情報通信学会, Sep. 2005.

[9] N.Iwanaga, Y.Shibata, M. Yoshimi, Y. Osana, Y. Iwaoka, T. Fukushima, H. Amano, A. Funahashi, N. Hiroi, H. Kitano, and K. Oguri. Efficient scheduling of rate law functions for ODE-based multimodel biochemical simulation on an FPGA. In *Proceedings of the 15th Field Programmable Logic and Applications (FPL) 2005*, pp. 666–669, Tampere, Finland, Aug. 2005. IEEE.