

FPGA を用いた生化学シミュレータ用の SBML 処理系の構築

長名 保範†, 福島 知紀†, 吉見 真聡†, 岩岡 洋†, 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 開放環境科学専攻 コンピュータ科学専修
〒 223-8522 横浜市港北区日吉 3-14-1 / E-Mail: bio@am.ics.keio.ac.jp

†† 科学技術振興機構 北野共生システムプロジェクト
〒 150-0001 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

概要

生物学実験の分野における定量的でスループットの高い実験手法が多く確立され、実験から得られるデータを元にした数値シミュレーションがさまざまに行われるようになってきている。生体内の反応機構のモデル化・シミュレーションと実験を繰り返すことで、対象の系に関するより詳細な理解が得られると期待されているが、生化学シミュレーションには膨大な計算時間が必要となる。これを解決するために、現在は PC クラスタを用いるのが一般的であるが、PC クラスタは価格やサイズ、消費電力などの面から研究者個人単位での計算資源としては適切でない。本研究グループではこれを解決するために、FPGA を用いた生化学シミュレーションアクセラレータ ReCSiP を構築中であり、本研究報告では、これを用いて一般的な生化学反応系の記述言語である SBML によって記述された系のシミュレーションを行うための方式を提案する。

Keywords : 生化学シミュレーション, FPGA, ReCSiP, SBML

Design of the SBML Processor for an FPGA-based Biochemical Simulator

Yasunori Osana†, Tomonori Fukushima†, Masato Yoshimi†, Yow Iwaoka†, Akira Funahashi††, Noriko Hiroi††, Yuichiro Shibata†††, Naoki Iwanaga†††, Hiroaki Kitano†† and Hideharu Amano†

† Department of Computer Science, Graduate School of Keio University
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama #223-8522, JAPAN

†† Kitano Symbiotic Project, ERATO-SORST, Japan Science and Technology Agency
6-31-15 Jungumae, M-31 Suite 6A Shibuya-ku #150-0001, JAPAN

††† Department of Computer and Information Sciences, Nagasaki University
1-14 Bunkyo-machi, Nagasaki #852-8521 JAPAN

Abstract

Recent high-throughput methods of biological experiments enabled raw-data based numerical simulation of biochemical systems. Biological systems will be understood by the repetition of simulations and experiments. To address the long computation time of biochemical simulators, PC clusters are used in many institutes. Although PC clusters are relatively economical solution in high-performance computing, it's not suitable as a kind of personal computing resources because of its cost, size and energy consumption. To address this problem, an FPGA-based biochemical simulation accelerator ReCSiP is proposed. In this research report, a method to simulate biochemical models described in SBML, the standard markup language for biochemical networks.

Keywords : Biochemical Simulation, FPGA, ReCSiP, SBML

1 はじめに

近年、生物学の分野では種々の高スループットの定量的計測技術の進歩により、精度の高い情報を大量に得ることができるようになったことから、E-Cell[1] や Virtual Cell[2] のように、細胞のシステム全体を計算機上に再構成しようとする意欲的な取り組みが行われている。

しかし、これらの大規模なシミュレータは膨大な計算時間を必要とし、実行には高性能な計算機を必要とする。研究者各個人がこれを利用できれば理想であるが、実際にはPC クラスタのような高性能な計算機は共有資源である。

そこで、本研究グループではFPGAを用いてシミュレーションを行うことで、さまざまな反応モデルや新しいアルゴリズムに柔軟に対応しつつ、ハードウェアで直接処理を行うことによる高速性を活かす可能性を模索している。現在、実機動作を視野に入れたシステムとして、ReCSiP[3](ReConfigurable Cell Simulation Platform)を開発中であり、これを用いてシミュレーションを効率よく行うための手法[4]を模索している。

本研究報告では、FPGAを用いてシミュレーションを行うための基本的な手法と、標準のモデリング言語であるSBML[5](Systems Biology Markup Language)で記述されたモデルを解析してシミュレーションを実行するためのソフトウェア基盤の構想と基本的な手法について述べる。

2 ReCSiPのハードウェア

ReCSiPは、

- FPGAを搭載したPCIカードとドライバ
- FPGA上に構成されるsimulation hardware
- シミュレータに与えるデータを生成するソフトウェア
- シミュレーションを走らせるためのユーザーインタフェース

といったコンポーネントから構成される。ボードは、Xilinx社のFPGA(XC2VP70)と、4セットのQDR-SRAMを64bit/66MHzのPCIバスに接続するもの(図1)であり、実機が稼働を開始している。ドライバはLinux-2.4.x用が現在動作しており、FreeBSD、Windows(NT系)、MacOS Xなどのドライバも実装を行う予定である。

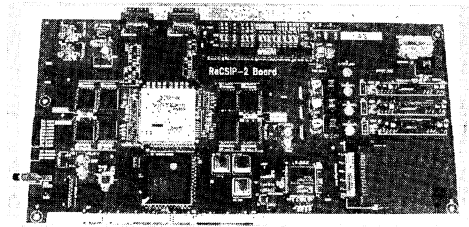


図1: ReCSiP-2 ボード

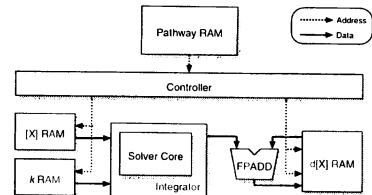


図2: Integrator + Solver Controller の構成

この節の以下の部分では、既に実装が開始されている、ReCSiPのハードウェア側の構成要素について述べる。

2.1 Solver Core

Solver coreは、反応速度式を解いてタイムステップあたりの濃度変化を計算するモジュールであり、FPGAでシミュレーションを行う上で基本となるモジュールである。反応速度式は微分方程式で記述され、反応モデル毎に異なる。Solver coreは一般に、いくつかの浮動小数点演算コンポーネントを接続して構成されており、それぞれ一種類、あるいは数種類の類似した形の反応速度式を解くことができる。

現在、SBML Level 1[6]で、predefined functionとして定められている各種の反応速度方程式を解くためのモジュール群を整備中である。

2.2 Integrator + Solver Controller

Integratorはsolver coreの出力を積分するための機構、solver controllerは、integratorを含めたsolver coreにデータを供給するための機構であり、図2にその構成を示す。現在高次のintegratorの実装は

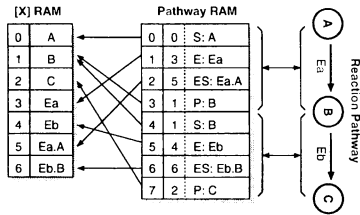


図 3: Pathway RAM による反応経路記述

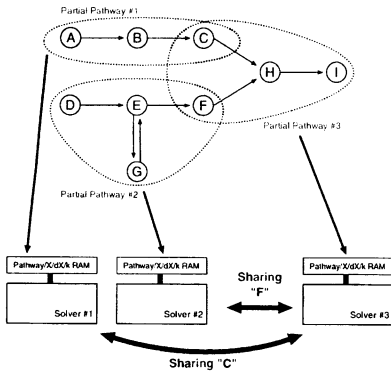


図 4: 反応経路の分割

ないため、1 次の Euler 法でのみ積分が可能であるが、今後 Runge-Kutta 法などが実装される予定である。

Solver controller の周辺には、反応経路を記述するための Pathway RAM と、時刻 t における各物質の濃度が保存されている [X] RAM、および時刻 t からのタイムステップ Δt における濃度変化を保存する d[X] RAM が配置されている。Pathway RAM と solver controller の動作を図示したのが図 3 であり、solver controller は、pathway RAM に書かれたポイントの配列の順番に [X] RAM をアクセスし、その出力を solver core に与えることで一連の反応を順次処理していく仕組みになっている。

濃度変化は d[X] RAM に一時格納されたのち、[X] RAM に一括して加算することで時刻を進めていく。

2.3 Solver 間スイッチ

ここまでで、一連の反応速度式を解く方式について述べたが、これだけでは図 4 のように複数の

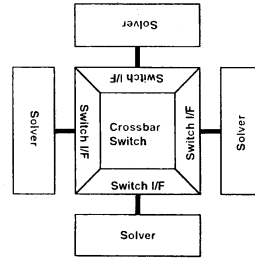


図 5: スイッチと Solver

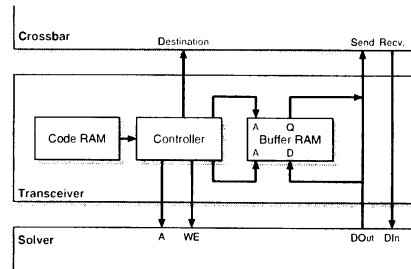


図 6: スイッチと Solver のインタフェース部

solver を用いて並列性を活かした高速化を行ったり、異なる種類の solver を併用して複雑な反応系を処理したりすることができない。これらを実現するためには、solver 間でデータを共有するための機構が必須である。

Solver 間でデータを交換するには、通信機構が各 solver のもつ [X] RAM や d[X] RAM に直接アクセスするのが簡単かつ効率のよい方法であり、これらのメモリは FPGA 上の dual-port メモリを用いているため、実現も簡単である。通信機構は図 5 のように構成されており、中心のクロスバスイッチと周辺部の各 solver のメモリをアクセスするためのインタフェース (図 6) から構成される。インタフェースには、Pathway RAM に似た方式で制御に用いるためのメモリ (Code RAM) が配置されており、Solver のメモリへのアクセスに用いるアドレスや、クロスバヘデータを送信する際の宛先などがこれによって制御される。

なお、中央のクロスバスイッチは、複数の宛先ビットを用いることで、一対多の転送を行うことが可能である。

3 ReCSiPのソフトウェア

前節で述べたように、シミュレーションの対象となる系に含まれる反応の反応速度式を処理することのできる solver を switch で接続し、solver に必要なデータと反応経路マップを与え、switch をプログラムしてやることでシミュレーションが可能になる。しかし、このデータを手作業によって作成することは簡単な作業ではない。

そこで、反応系の記述向けに標準化された言語である SBML から反応の種類、関係する分子種、初期濃度など(図7)を読み取って、solver や switch に与えるための各種要素の構成は以下ようになる。

Solver Library HDL で記述された solver と、その仕様を記述する XML ベースの言語 (SCML: Solver Configuration Markup Language) のライブラリ

Predefined Solver-set Library 論理合成・配置配線済みの、いくつかの solver とそれらを接続する switch を含む FPGA の構成情報

Optimizer SBML で記述されたモデルから、必要な solver のセットを抽出し、回路生成を行うソフトウェア

Scheduler Optimizer が生成した回路中の solver にデータセット ([X] RAM) や、反応経路 (Pathway RAM) を与え、switch の configuration を生成して solver 間の通信を決定するソフトウェア

Interface software SBW[7] (Systems Biology Workbench) に準拠し、モデリングツールや、

```

<model name="sample_model">
  <listOfCompartments>
    <compartment name="cell"/>
  </listOfCompartments>
  <listOfSpecies>
    <species name="S" initialAmount="0.3" compartments="cell"/>
    <species name="P" initialAmount="0" compartments="cell"/>
  </listOfSpecies>
  <listOfReactions>
    <reaction name="Reaction1" reversible="false">
      <listOfReactants>
        <speciesReference species="S" />
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="P" />
      </listOfProducts>
      <kineticLaw formula="uui(S,P,km)">
        <listOfParameters>
          <parameter name="km" value="0.1" />
        </listOfParameters>
      </kineticLaw>
    </reaction>
  </listOfReactions>
</model>

```

図7: SBMLの基本的な記述

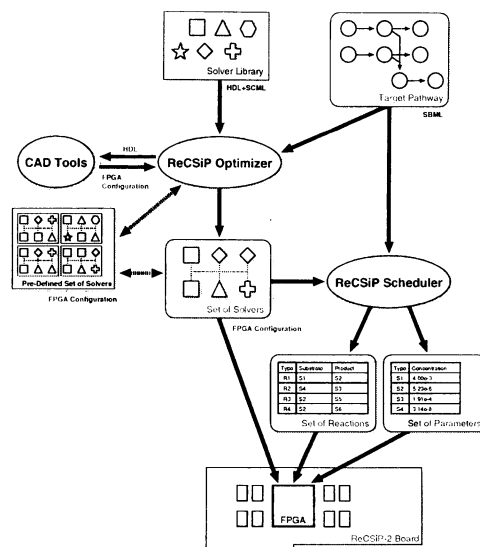


図8: Optimizer/Schedulerの役割

シミュレーション結果の解析ツールと ReCSiP の連携を行うソフトウェア

ソフトウェアのうち、最も重要なのは optimizer と scheduler であり、これらの働きを図8に示す。以下では、鍵となるコンポーネントの役割と基本的な方式について述べる。

3.1 Solver Library

Solver library には、SBML Level 1 の定義済み関数として規定されているさまざまな反応速度式の実装が含まれるが、ここでは例として、Michaelis-Menten 型の反応速度式ファミリーを挙げる。これには、以下の3つの反応速度式が含まれる。

- (1) Irreversible Simple Michaelis-Menten

$$v = \frac{V_m S}{K_m + S}$$

- (2) Uni-Uni Reversible Simple Michaelis-Menten

$$v = \frac{V_f S / K_m S - V_r P / K_m P}{1 + S / K_m S + P / K_m P}$$

- (3) Uni-Uni Reversible Simple Michaelis-Menten with Haldane adjustment

$$v = \frac{(V_f / K_{m1})(S - P / K_{eq})}{1 + S / K_{m1} + P / K_{m2}}$$

表 1: Michaelis-Menten ファミリの Solver Core

反応式	使用 FP ユニット数			遅延 (clk)	面積 (slice)
	Add	Mult	Div		
(1)+(2)+(3)	2	1	2	50	4699
(1)+(2)+(3) (*)	2	2	1	50	3177
(1)+(2) (*)	2	2	1	41	3177
(1)	1	1	1	26	2433

(*) = 定数除算を乗算で代替

これら 3 つの反応速度式を解く solver core は、現在表 1 に示す 4 種類が実装予定である。一番面積の大きなものは、3 つの反応速度式をそのまま解くことができるもので、対応する反応速度式を限定したり、あるいは定数除算を乗算器で行うことによって、占有面積や遅延クロック数を削減している。

これらの solver は HDL で記述されているが、シミュレーションを走らせるための solver set を構成する際に、FPGA の面積を考慮しながら、もっとも処理効率のよい組合せを導出する必要がある。したがって、HDL で書かれた solver のそれぞれについて、

- 占有する面積
- 計算に所要するクロック数
- 対応している反応速度式のリスト
- 各反応速度式について、変数の入出力の規定などを、機械可読なフォーマットで記述しておく必要がある。書式はまだ確定していないが、将来の拡張が容易な XML ベースで、SCML (Solver Configuration Markup Language) として文法を規定していく予定である。

3.2 Predefined Solver-set Library

シミュレーションを走らせるためには、HDL で記述された solver のいくつかを、必要なポート数の switch と組み合わせて論理合成・配置配線を行って、FPGA 上に構成できるようにする必要がある。

しかし、論理合成や配置配線には、FPGA ベンダの提供する CAD が必要であり、これには有料のライセンスが必要であるとともに、限られたプラットフォームでしか動作しないという問題がある。さらに、これらのプロセスには長い場合に数時間

要するため、シミュレーションの度に CAD を走らせるのは明らかに非効率的である。

これらの問題を解決するために、いくつかの solver の組合せの FPGA 構成情報をあらかじめ配置配線してライブラリとして提供するとともに、一旦配置配線された FPGA の構成情報はライブラリに登録できるようにしておくことが望ましい。

また、solver library や、solver-set library は、web を経由して、自動的に最新のものを入手する機構が必要不可欠であると考えられる。

3.3 Optimizer

Optimizer はまず、シミュレーション対象の SBML 記述から、対象の系に含まれる反応のリストを抽出し、必要な solver set を構成する。続いて、これを元にして必要な solver と switch を接続した HDL を生成し、CAD ツールを呼び出して論理合成・配置配線を行うか、あるいはシミュレーションに利用可能な構成情報を predefined solver-set library から探して取得する。

新しく、solver set を構成する際には、FPGA の面積の許す限り並列性を活かすことができるように、利用率の高い種類の solver から順に多く set に組み入れるのが基本的な方針である。

さらに、高次の解法では、時間刻み幅をいくつかに分割して、中間の点でそれぞれ計算を行うため、反復して solver core を利用する必要がある。この場合には複数の solver core を含む高次解法向けの solver が性能向上に有効であり、ひとつの反応に割り当てられ得る solver の種類 (該当する solver core の種類と、並列度のバリエーションの積) は、1 次の解法の場合 (該当する solver core の種類だけしかない) に比べて多くなり、solver の選択が実行性能に大きく影響することになる。

3.4 Scheduler

Scheduler は、optimizer が生成、あるいは選択した solver set の構成に合わせて、SBML で記述されている各反応の solver への割当てと実行スケジューリングを行うと同時に、solver 間で共有するデータの、switch を用いた交換のスケジューリングも同時に行う。

現状ではこれらのスケジューリングは、scheduler によって静的に行われ、実行時に変更されることはないが、将来的には特定の条件でパラメータや反応経路が変化したりするような記述をサポートする予定である。これには、scheduler の拡張とともに、pathway RAM を解釈する solver controller の仕様拡張が必要になる。

また、現在は単一の FPGA で構成されるシステムのみを対象としているため、optimizer や scheduler は比較的単純なストラテジで処理を行うことができるが、ReCSiP-2 ボードはボード間の相互接続を行うための multi-gigabit transceiver を装備しており、将来はこれを用いて多数の FPGA を用いたシステムを構築することが考えられる。FPGA 間の通信コストは FPGA 内の通信コストより格段に高いため、この場合には optimizer と scheduler の処理は複雑化するとともに、optimizer と scheduler を融合させて、スケジューリングまで考えた、複数の FPGA にまたがる最適な solver の配置を計算する必要がある。

3.5 Interface Software

Interface software は、ReCSiP のインターフェイス部となるソフトウェアで、SBW の API 規格に準拠したものとなる予定である。SBW に準拠することで、ユーザは他のソフトウェアベースのシミュレータと同様に、ReCSiP によるシミュレーションを任意の SBW 準拠なモデリングツールや解析ツールと併用することが可能になる。

Optimizer, scheduler といったツール群はこの interface software から呼び出され、ボード上の FPGA の configuration、ボードへのデータ転送およびシミュレーションの実行は interface software 自体によって実行される。

4 まとめと現状・今後

本研究報告では ReCSiP における、FPGA を用いた生化学シミュレーションの基本的な方式を示し、ソフトウェアベースのシミュレータと同等のインタフェイスをユーザに提供するための、SBML 処理系を中心としたソフトウェアツール群の構想とその基本的な構想について述べた。

スケジューラは既にプロトタイプを実装中であり、SCML の規格策定、solver core のライブラリ整備などと併せて、実際のモデルを用いた性能検証等を進めて行く予定である。将来的には、確率モデルを用いたシミュレータ [8] との統合も計画されている。

謝辞

この研究は文部科学省の平成 16 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。

参考文献

- [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 Structure and Function. *Bio-physical Journal*, Vol. 73, pp. 1135–1146, Sep. 1997.
- [3] Yasunori Osana, Tomonori Fukushima, Masato Yoshimi, and Hideharu 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] 長名保範, 福島知紀, 吉見真聡, 天野英晴. FPGA を用いた細胞内代謝系のマルチモデルシミュレーション. リコンフィギャラブルシステム研究会 4-8. 電子情報通信学会, Sep. 2004.
- [5] M. Hucka, A. Finney, B.J. Bornstein, S.M. Keating, B.E. Shapiro, J. Matthews, B.L. Kovitz, M.J. Schilstra, A. Funahashi, J.C. Doyle, and H. Kitano. Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project. *IEE Systems Biology*, Vol. 1, No. 1, pp. 41–53, 2004.
- [6] Michael Hucka, Andrew Finney, Herbert Sauro, and Hamid Bolouri. *Systems Biology Markup Language (SBML) Level 1: Structures and Facilities for Basic Model Definitions*. Systems Biology Workbench Development Group, ERATO Kitano Symbiotic Systems Projec, MC 107-81 California Institute of Technology, Pasadena, CA 91125, USA, version 2 edition, Aug. 2003.
- [7] H. Sauro, M. Hucka, A. Finny, C. Wellock, H. Bolouri, J. Doyle, and H. Kitano. Next Generation Simulation Tools: The Systems Biology Workbench and BioSPICE Integration. *Omics*, Vol. 7, No. 4, pp. 355–572, Dec. 2003.
- [8] 吉見真聡, 長名保範, 福島知紀, 天野英晴. 確率モデルを用いた化学反応シミュレーションの FPGA による高速化. リコンフィギャラブルシステム研究会 4-33. 電子情報通信学会, Sep. 2004.