

# An FPGA-Based, Multi-model Simulation Method for Biochemical Systems

Yasunori Osana, Tomonori Fukushima,  
Masato Yoshimi, Yow Iwaoka  
Keio University  
Yokohama, Japan

Akira Funahashi, Noriko Hiroi  
Kitano Symbiotic Systems Project,  
ERATO-SORST, JST  
Tokyo, Japan

Yuichiro Shibata  
Nagasaki University  
Nagasaki, Japan

Hiroaki Kitano  
Kitano Symbiotic Systems Project,  
ERATO-SORST, JST  
Tokyo, Japan

Hideharu Amano  
Keio University  
Yokohama, Japan

## Abstract

*Modeling and simulation of a cellular system on computers are now becoming an essential process in biological researches. However, modern PCs can't provide enough performance to simulate large-scale biochemical networks. ReCSiP is the alternative FPGA-based solution for biochemical simulations. In this paper, the novel method of biochemical simulation with multiple reaction models on an FPGA is proposed. The method generates optimal circuit and its optimal schedule for each simulation models written in SBML, the standard markup language in systems biology. ReCSiP has a Xilinx's XC2VP70 and achieved over 20-fold speedup compared to Intel's PentiumIII 1.13GHz.*

**Keywords:** Biochemical simulation, FPGA, SBML, Multi-model, ODE

## 1. Introduction

Biochemical simulation, or the simulation of a cellular system is one of the major applications in bioinformatics. Various biochemical simulators were developed since KINSIM[1] was developed in 1983. Some of recent simulators like E-Cell[9] and Virtual Cell[4] are called "whole-cell simulator" because their targets are large-scale networks such as whole cell models.

The whole-cell simulation is a major challenge in both biology and computer science because it requires large computational resources and detailed description of the target system. Computers are getting faster and faster, however, it's not sufficient to analyze the behavior of biochemical models in detail. For example, 2 seconds simulation of a nerve cell require 2 days on Virtual Cell using a workstation with MIPS R8000[8].

ReCSiP[5] is an FPGA-based, compact, high-throughput computing platform to address the problem about computational time.

### 1.1. Biochemical Simulation

The purpose of biochemical simulation is to know the behavior of the target reaction pathway. The behavior is expressed as the change of concentration of substances in time series.

Before starting the simulation, the following things have to be known:

- the reaction pathway map of the target system, as the example in Fig.1. This includes the list of substances and reactions in the system,
- initial concentration of all substances in the target system, and
- kinetic parameters of all reactions in the target system.

It's a kind of initial-value problem, which solves a set of kinetic formulas (usually expressed as ODEs: an example is

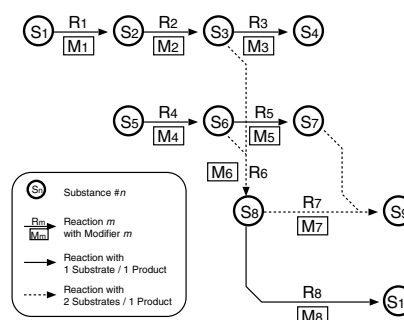


Figure 1. An Example of Reaction Pathway

described in 3.2.2) which correspond to the set of reaction models. By the repetition of numerical integration at every timestep, the concentration curve in the time series can be obtained.

## 1.2. SBML/SBW

Software infrastructure to enable integration of modeling, simulation and analysis is essential in modern biology. As the solution of this problem, SBML (Systems Biology Markup Language)[2] and SBW (Systems Biology Workbench)[7] have been developed.

SBML is an open, XML-based language for representing biochemical reaction networks. It's now supported by more than 65 software tools<sup>1</sup> such as modeling tools, simulators, analyzers and databases. SBML-compliant tools can share the models with each other.

SBW is a modular, broker-based, message-passing framework for simplified intercommunication between applications. This enables communication among software tools. For example, simulators can be invoked from modeling tools. There are many SBW-enabled modules<sup>2</sup>, including ODE-based simulator, stochastic simulator, Matlab translator, bifurcation analysis tool and optimization module.

## 1.3. ReCSiP

ReCSiP is an FPGA-based high-throughput biochemical simulation platform. Some prototype modules for biochemical simulations are already implemented on FPGAs[6][10], and they achieved about 10 to 200-fold speedup compared to modern microprocessors.

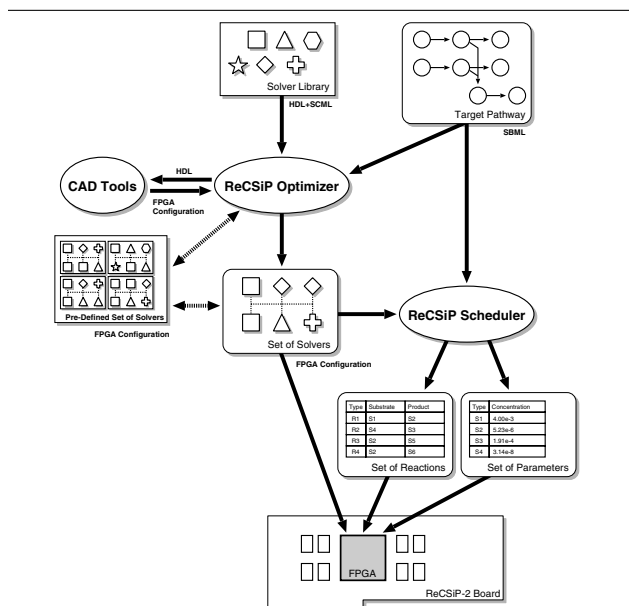
It runs the modules called "solvers" which calculate the velocity of biochemical reactions. By connecting the solvers with communication switches, it's possible to simulate biochemical systems with some different types of reaction. ReCSiP achieves the best performance by generating optimal set of solvers for each target system.

## 2. Software Organization

Although ReCSiP is an FPGA-based system, the software components, especially the optimizer and scheduler illustrated in Fig.2 take major roles. They import SBML description, then generate the circuit on FPGA and dataset to be loaded in the circuit. These components are called from SBW interface of ReCSiP.

<sup>1</sup> SBML-compliant tools are listed on <http://www.sbml.org/>.

<sup>2</sup> SBW-compliant tools are listed on <http://www.sbw-sbml.org/>.



**Figure 2. Overview of the FPGA-based Simulator**

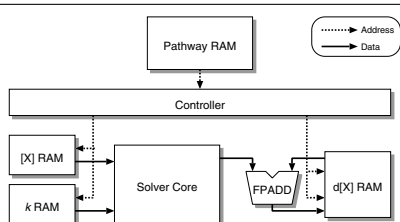
### 2.1. Optimizer and Scheduler

As illustrated in Fig.2, the optimizer generates a set of solvers which is capable of processing all kinetic formulas in the target system. The optimizer has 2 approaches to do this: to generate the "Optimal Set" of solvers, or select the "Best One" from pre-defined sets of solvers.

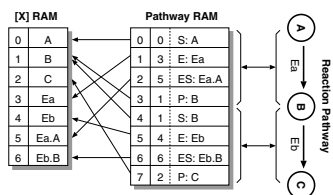
Optimal set consists of only the necessary solver modules. Some of them are placed two or more in the set to improve the throughput by parallelization. In this case, the optimizer passes the RTL description of solver set to CAD tools. However, not all biologists have CAD tools for FPGA on their computers. Moreover, CAD tools consume too long time for synthesis, placement and routing. To avoid these problems, the optimizer can select a solver set from the solver set repository. The solver set repository is a pool of pre-defined, pre-placed-and-routed set of solvers on the web.

The scheduler generates the list of reactions and parameter set for the set of solvers given by the optimizer. It also generates the control code for the communication switch in the set of solver.

The selected solver set by the optimizer is loaded as the configuration data for the FPGA, then the dataset generated by the scheduler is loaded on the memory blocks in the solver set on the FPGA before running simulation.



**Figure 3. Overview of Control Mechanism in Solver**



**Figure 4. Concept of Pathway RAM**

### 3. Hardware Organization

#### 3.1. ReCSiP-2 Board

The hardware part of ReCSiP, or the ReCSiP-2 board is a PCI board which has a Xilinx's XC2VP70 as its core, and 8 chips of 18Mbit QDR-I SRAM. Two multi-gigabit serial interfaces, a DDR-SDRAM SO-DIMM socket and a physical random number generator are on the board as well.

#### 3.2. Structure of Solvers

As illustrated in Fig.3, each solver consists of a solver core, a controller, a set of memory blocks and an adder. The solver core is the reaction-specific circuit to solve the kinetic formula, and the other components, including memory set, are the controlling facilities to run a simulation.

**3.2.1. Solver Controller** The role of Solver Controller is managing the input/output of the solver core.

There are 4 sets of memory blocks in a solver and are implemented on the BlockRAM in Xilinx's FPGAs. The details are as bellow:

- [X] RAM, to store the concentration of reactants,
- $k$  RAM, which has the rate constants,
- $d[X]$  RAM, to store the derivatives of [X], and
- Pathway RAM, which has the list of reactions.

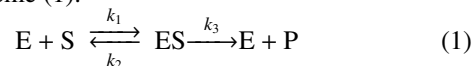
The functionality of the controller is realized by "Pathway RAM" which is an array of pointers for the list of reactions, as illustrated in Fig.4. [X] RAM is read by the order

described in Pathway RAM, and the data is sent to Solver Core. By this simple mechanism, a set of reactions can be solved<sup>3</sup>.

Solver Controller also manages the output of its core, and its process is divided into 2 phases. In phase 1, the result from core is accumulated in  $d[X]$  RAM. In phase 2 the derivatives in  $d[X]$  RAM are added to [X] RAM, to go to the next timestep.

**3.2.2. Solver Core** Solver cores are the modules to calculate the reaction rates from concentrations of reactants. Usually they're consisting of some IEEE-754 compliant single-precision pipelined FP arithmetic units.

For example, MM-Euler is a solver core that is based on irreversible Michaelis-Menten reaction model and Euler's method of numerical integration. This is a very simple model of enzyme kinetics as used in Fig.4, and it can be written as Scheme (1).



From this scheme, the velocity of concentration change can be expressed as the following ODEs.

$$\frac{d[S]}{dt} = -k_1[S][E] + k_2[ES] \quad (2)$$

$$\frac{d[P]}{dt} = k_3[ES] \quad (3)$$

$$\frac{d[E]}{dt} = -k_1[S][E] + (k_2 + k_3)[ES] \quad (4)$$

$$\frac{d[ES]}{dt} = k_1[S][E] - (k_2 + k_3)[ES] \quad (5)$$

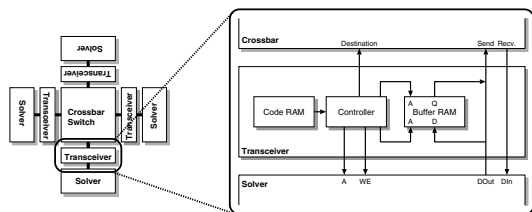
MM-Euler has 2 FP multipliers and 1 FP adder to solve these kinetic formulas. Implementation of other solver cores for reaction models which are defined as "predefined functions" in SBML level 1[3] are now going on.

**3.2.3. Evaluation** MM-Euler occupies about 5% of the total slices and 8% of the dedicated multipliers. Its maximum operating frequency was 112.56[MHz], without any additional circuits. The solver requires 4 clock cycles for each reaction, so the throughput is 28.14[Mreactions/sec.], which is about 2.8 times faster than Intel PentiumIII running at 1.13GHz.

#### 3.3. Solver-to-Solver Communication

When the target pathway has some different kind of reactions, or the number of reactions is too large, the pathway have to be divided onto some solvers. The solvers have to communicate with each other to keep the consistency of the contents in [X] RAM, since some reactants are shared among some solvers (S3, S6, S7, S8 in Fig.1) in such cases. This communication facility is the key of the multi-model simulation on an FPGA.

<sup>3</sup> Note that this figure is simplified for description.



**Figure 5. Structure of the Switch**

# of Solvers	Total (%)	SW ratio (%)
3	17.89	3.60
4	24.21	5.09
5	30.24	4.85
6	36.35	6.78
7	42.74	6.61
8	48.94	6.35
9	54.71	6.58
10	62.19	6.69

**Table 1. Resource Utilization**

**3.3.1. Implementation** The solvers are connected to the crossbar switch with transceiver modules, as shown in Fig.5. Transceivers send and receive data between the solver and crossbar. It can access  $[X]$  RAM or  $d[X]$  RAM without any interruption on solver's process itself, because the memory blocks are dual-ported. The crossbar is a bi-directional, multicast-capable switch.

Each transceiver has Code RAM, which programs the transaction between solver and crossbar, based on the same idea to Pathway RAM.

**3.3.2. Evaluation** The communication facility was also written in Verilog-HDL, and was synthesized, placed and routed with several MM-Euler based solvers. "Total" in table 1 is the area ratio of total design (including the solvers and the switch) in the FPGA, Xilinx's XC2VP70. "SW ratio" in the table is the area percentage of the switch in the design. It's about 4 to 6% of overall circuit size, and quite reasonable in size.

Maximum operating frequency of these circuits were around 90 to 93MHz, and the system with 10 solvers achieved 230[Mreaction/sec.]. It's about 22 times faster than Intel Pentium III running at 1.13GHz.

## 4. Summary and Future Works

The overview and preliminary evaluation of the FPGA-based biochemical simulator, ReCSiP was described. It achieves more than 20 times as fast as software-based simulators by generating optimal circuit and optimal schedule of solvers for each target system models.

The front-end part of ReCSiP, to support SBML/SBW is now under development, and will be available April 2005. The back-end part of ReCSiP, the solvers and solver-to-solver communication facility is already available, and the solver library is continuing to grow up.

## Acknowledgment

This work was supported by the Ministry of Education, Culture, Sports, Science and Technology, of the Japanese Government (Special Coordination Funds for Promoting Science and Technology).

## References

- [1] B. A. Barshop et al. Analysis of numerical methods for computer simulation of kinetic processes: Development of kin-sim — a flexible, portable system. *Analytical Biochemistry*, 130:134–145, 1983.
- [2] M. Hucka, A. Finney, B. Bornstein, S. Keating, B. Shapiro, J. Matthews, B. Kovitz, M. Schilstra, A. Funahashi, J. 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*, 1(1):41–53, 2004.
- [3] M. Hucka, A. Finney, H. Sauro, and H. Bolouri. *Systems Biology Markup Language (SBML) Level 1: Structures and Facilities for Basic Model Definitions*. Systems Biology Workbench Development Group, ERATO Kitano Symbiotic Systems Project, MC 107-81 California Institute of Technology, Pasadena, CA 91125, USA, version 2 edition, Aug. 2003.
- [4] I. Moraru, S. JC, S. BM, and L. LM. The virtual cell: an integrated modeling environment for experimental and computational cell biology. *ann n y acad sci* 971:595-6. *Annals of the New York Academy of Sciences*, 971:595–596, 2002.
- [5] Y. Osana, T. Fukushima, and H. Amano. Recsip: a reconfigurable cell simulation platform - accelerating biological applications with fpga -. In *Proc. of ASP-DAC '04*, pages 731–733, Jan. 2004.
- [6] Y. Osana, T. Fukushima, M. Yoshimi, and H. Amano. An fpga-based acceleration method for metabolic simulation. *IEICE Trans. on Information and Systems*, E87-D(8):2029–2037, Aug. 2004.
- [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. *Omic*, 7(4):355–572, Dec. 2003.
- [8] J. Schaff et al. A general computational framework for modeling cellular structure and function. *Biophysical Journal*, 73:1135–1146, Sep. 1997.
- [9] M. Tomita et al. E-cell: software environment for whole-cell simulation. *Bioinformatics*, 15(1):72–84, Jan. 1999.
- [10] M. Yoshimi et al. Stochastic simulation for biochemical reactions on fpga. In *The 14th International Conference on Field Programmable Logic and Applications*, volume 3203 of *Lecture Notes in Computer Science*, pages 105–114. Springer, Aug. 2004.