

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 †北野共生プロジェクト ††長崎大学

背景

- 計算機を用いた生物学の進展
 - 実験データの分析
 - 遺伝子配列の相同性検索など
 - すでに方法が確立された分野
 - PCクラスタなどが盛んに利用されている
 - 計算機シミュレーション
 - 化学反応などの数理モデル化とシミュレーション
 - 化学反応や力学系のモデル化
 - まだまだ発展途上の分野
 - 計算量はとても多い

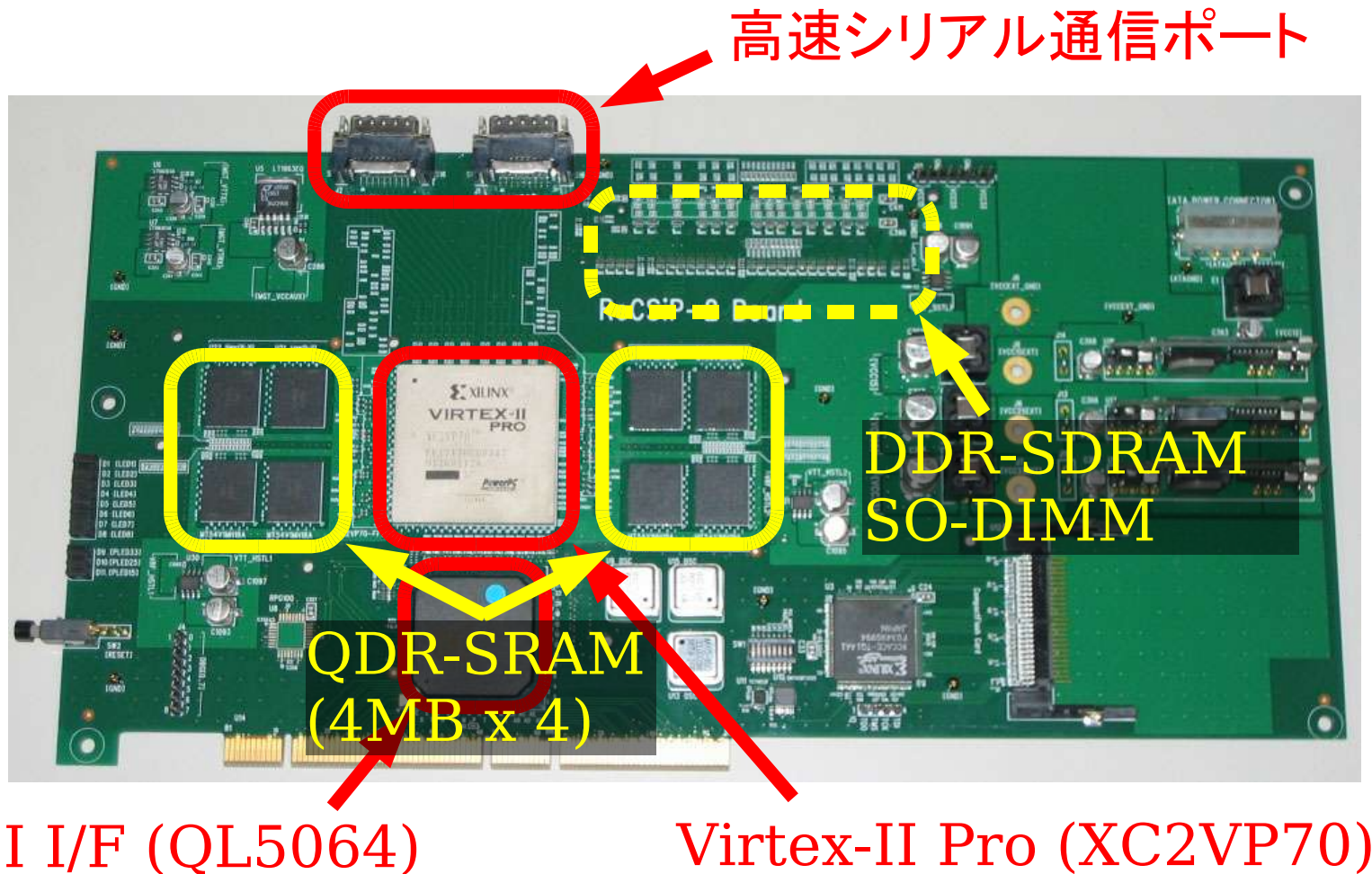
ReCSiP: a ReConfigurable Cell Simulation Platform

- FPGA を用いた生化学シミュレーション環境
 - 研究者が個人で使える環境を目指す
 - ボード1枚なので、PCクラスタよりも小型
 - 消費電力も小さい
 - FPGAに複数の処理ユニットを並べて並列処理
 - ハードウェア的な方式はひとまず決定
 - シミュレーションを走らせることはできるが、
 - シミュレーションモデルを書くのが困難



標準化されたモデル記述言語SBMLを利用

ReCSiP-2 Board



生化学シミュレーション

- 生命活動は化学反応の連鎖
 - 酵素などの modifier が介在する
 - 反応機構モデルから反応速度式が導出可能
 - たとえば、Irreversible Michaelis-Menten モデル:



- 一連の反応を常微分方程式のセットとして記述し、時系列での濃度変化をシミュレーションする

SBML: Systems Biology Markup Language

- 生物のシステムレベルモデル化の標準言語
 - シグナル伝達系・生化学反応・遺伝子発現などをXMLで記述
 - <http://www.sbml.org/>
 - 生化学反応モデルならば...
 - 反応系に含まれる物質のリスト
 - 反応系を構成する反応のリストと、それぞれの反応機構
 - 物質の初期濃度や反応速度定数などのパラメータなど

SBML 記述の概要

```
<?xml version="1.0" encoding="UTF-8"?>  
<sbml xmlns = "http://www.sbml.org/sbml/level1"  
  level = "1" version = "1">
```

```
  <model name = "Title">
```

```
    <listOfCompartments>
```

```
  </listOfCompartments>
```

反応の起こる「区画」のリスト

```
    <listOfSpecies>
```

```
  </listOfSpecies>
```

反応に関わる分子のリスト

```
    <listOfReactions>
```

```
  </listOfReactions>
```

反応のリスト

```
  </model>
```

モデル

```
</sbml>
```

反応記述の例

```
<listOfReactions>
```

```
<reaction id="R01">
```

```
<listOfReactants>
```

```
<speciesReference species="X0" stoichiometry="1"/>
```

```
</listOfReactants>
```

反応物

```
<listOfProducts>
```

```
<speciesReference species="S1" stoichiometry="1"/>
```

```
</listOfProducts>
```

生成物

```
<kineticLaw formula="mass(X0,k1)">
```

```
<listOfParameters>
```

```
<parameter id="k1" value="0.0033"/>
```

```
</listOfParameters>
```

```
</kineticLaw>
```

反応モデルとパラメータ

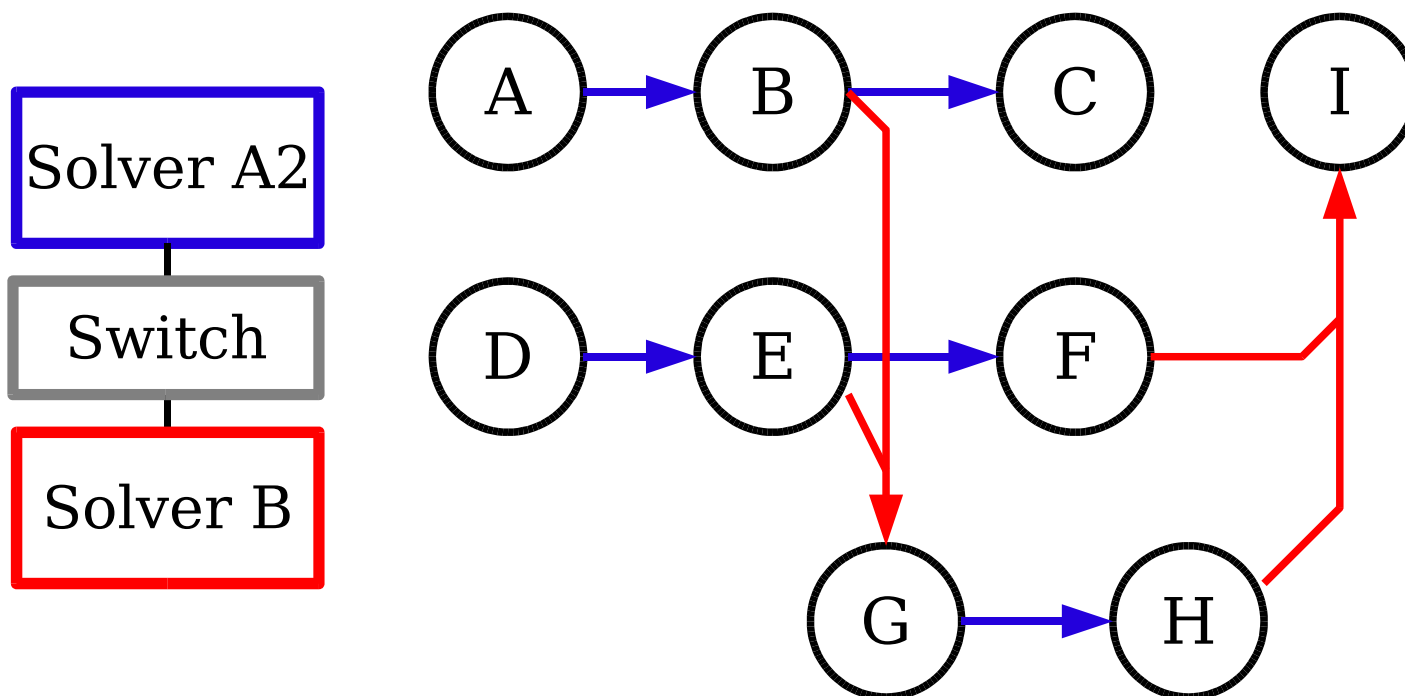
```
</reaction>
```

個々の反応を定義

```
</listOfReactions>
```


FPGA によるシミュレーション

- 反応速度式を解くモジュール: solver
 - 必要な solver をスイッチで結合して使用
 - スイッチで solver 間のデータを共有



Solver の構成 (1)

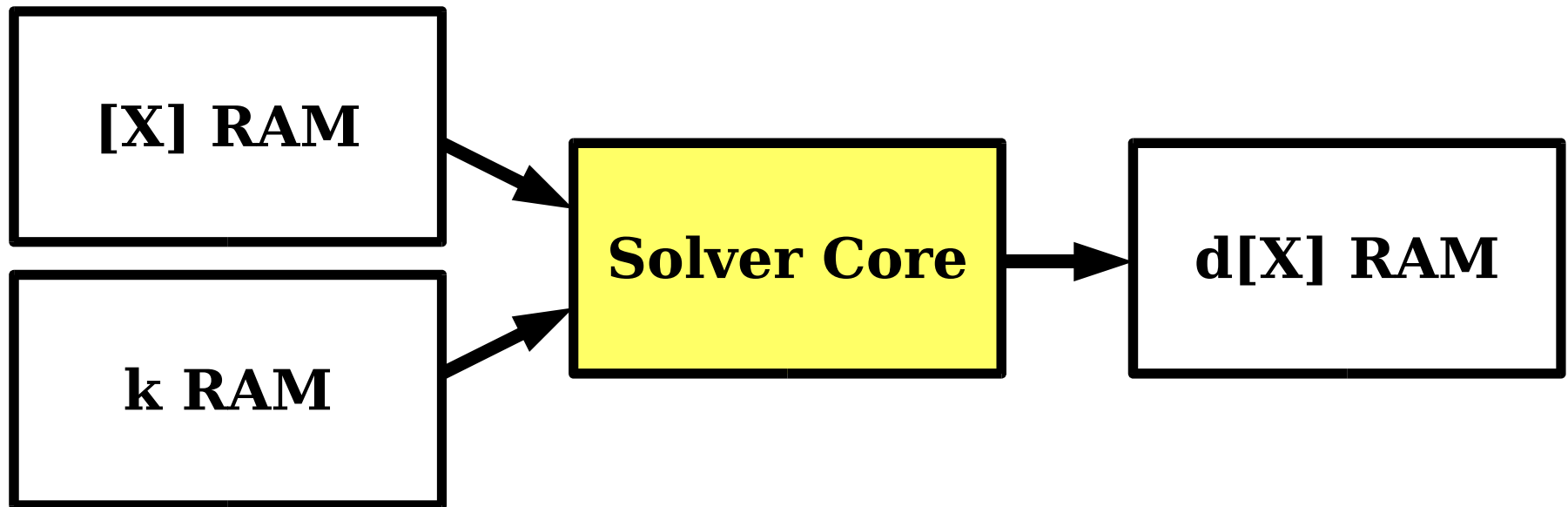
- Solver core: 反応速度式を解くモジュール
 - いくつかの FP 演算器で構成
 - 入力: 反応速度定数 k + 反応に関わる物質の濃度 $[X]$
 - 出力: 反応速度 v



- Solver をライブラリ化
 - SBML Level 1 の「Predefined Function Set」

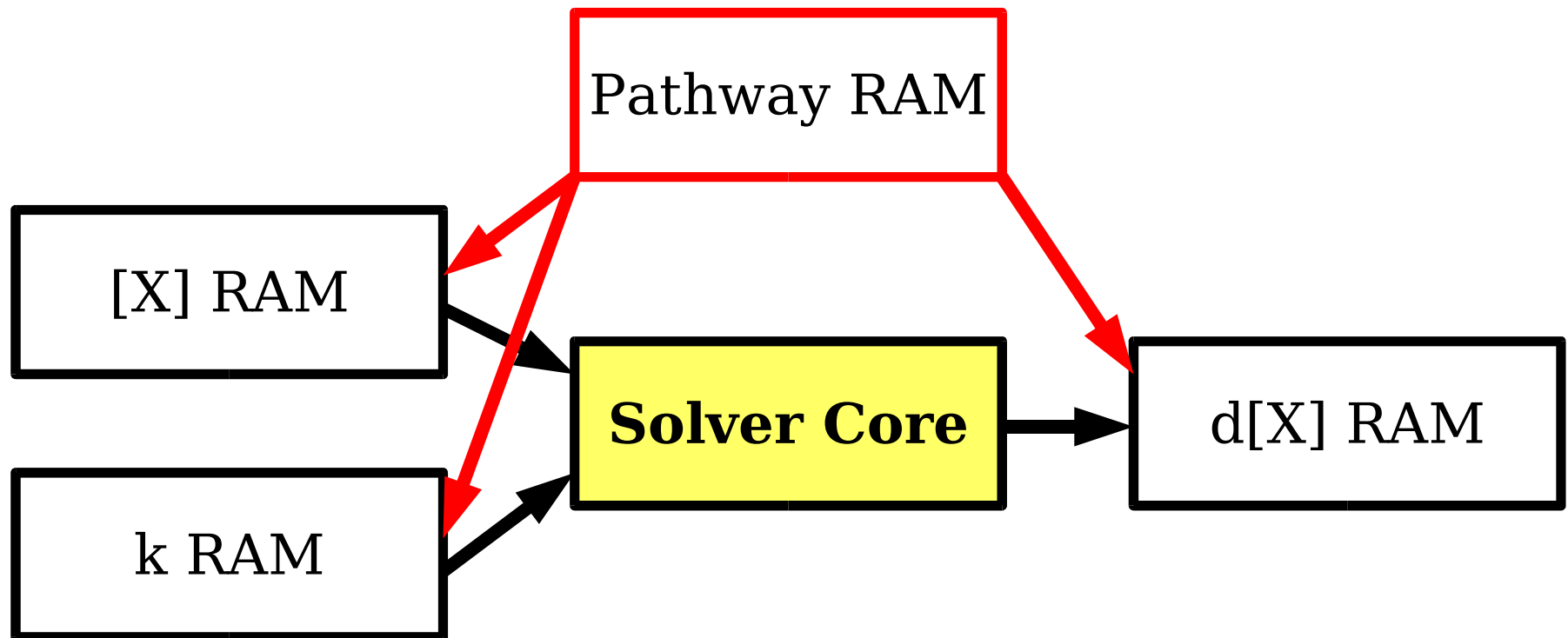
Solver の構成 (2)

- $\{[X], k, d[X]\}$ RAM
 - 入出力データを保持するためのメモリ



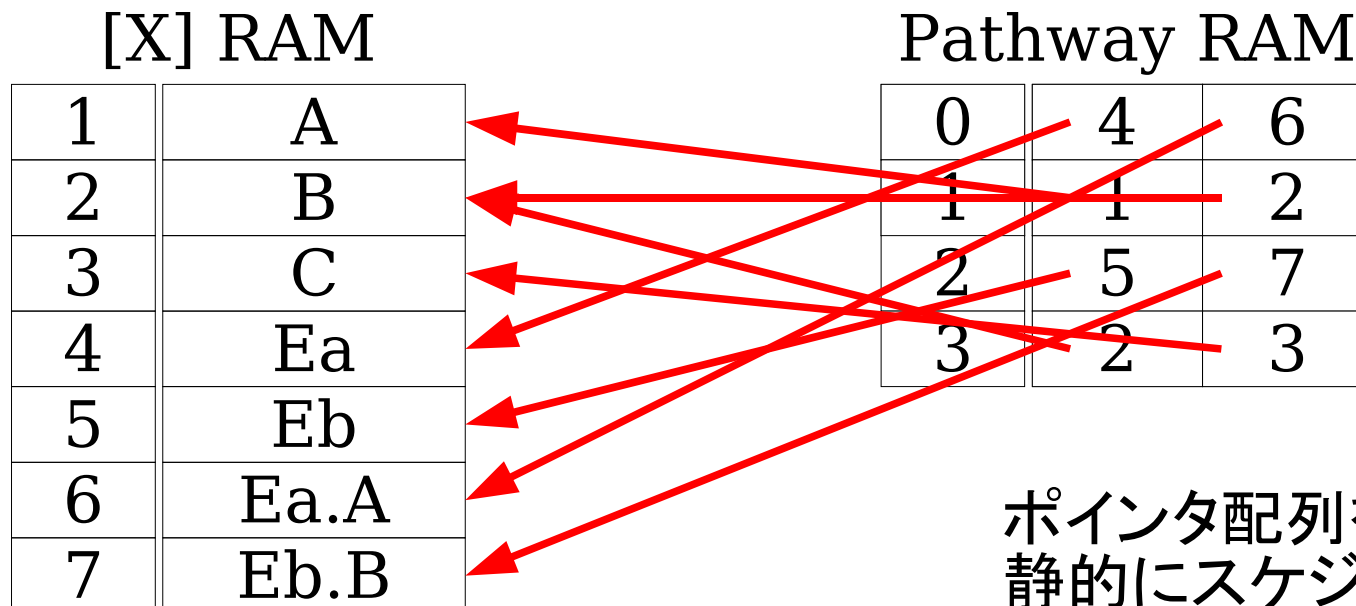
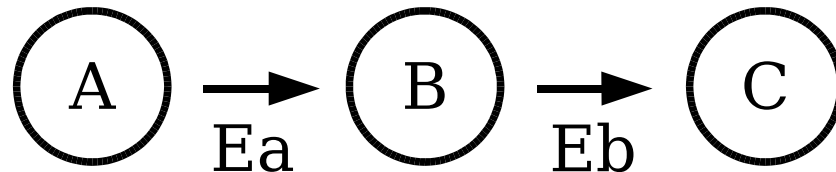
Solver の構成 (3)

- Pathway RAM: 各メモリのR/W制御用



Solver の構成 (4)

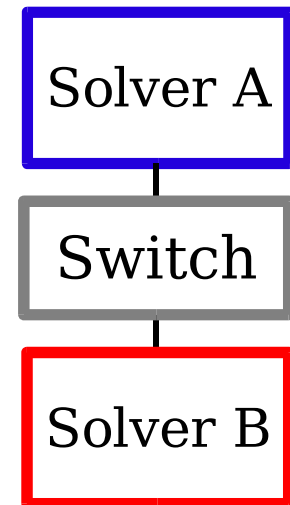
- Pathway RAM による反応経路表現 (イメージ)



ポインタ配列を使って、
静的にスケジューリング

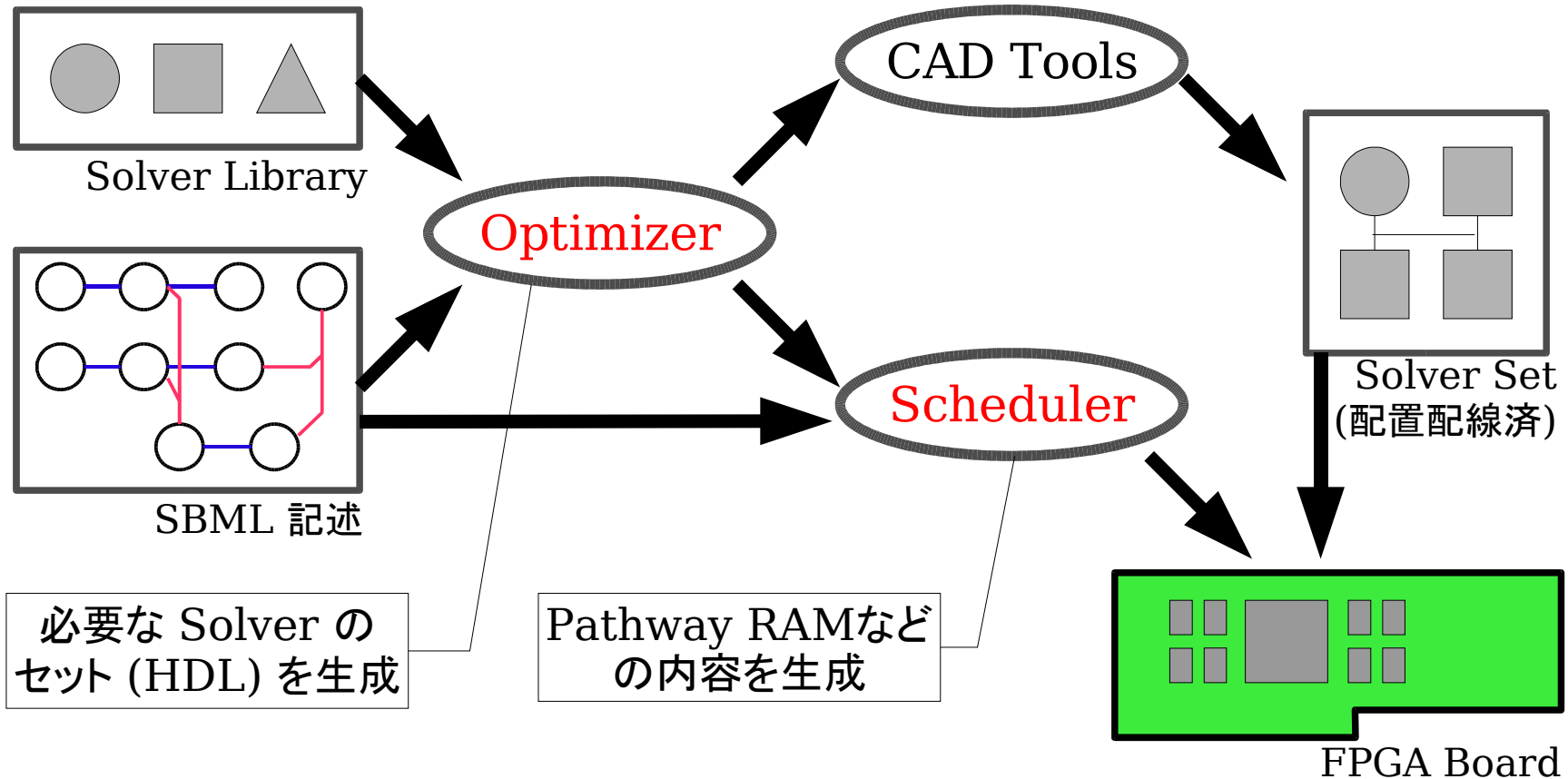
マルチモデルシミュレーション

- 複数の solver をスイッチで結合
 - 各 solver 間で $[X]$, $d[X]$ RAM のデータを交換
 - スイッチは crossbar
 - 転送制御用 RAM を持つ
 - 転送元、転送先を指定
 - 1ワードで1回分の転送を制御する
 - Pathway RAM に似た仕掛け
 - 静的なスケジューリング



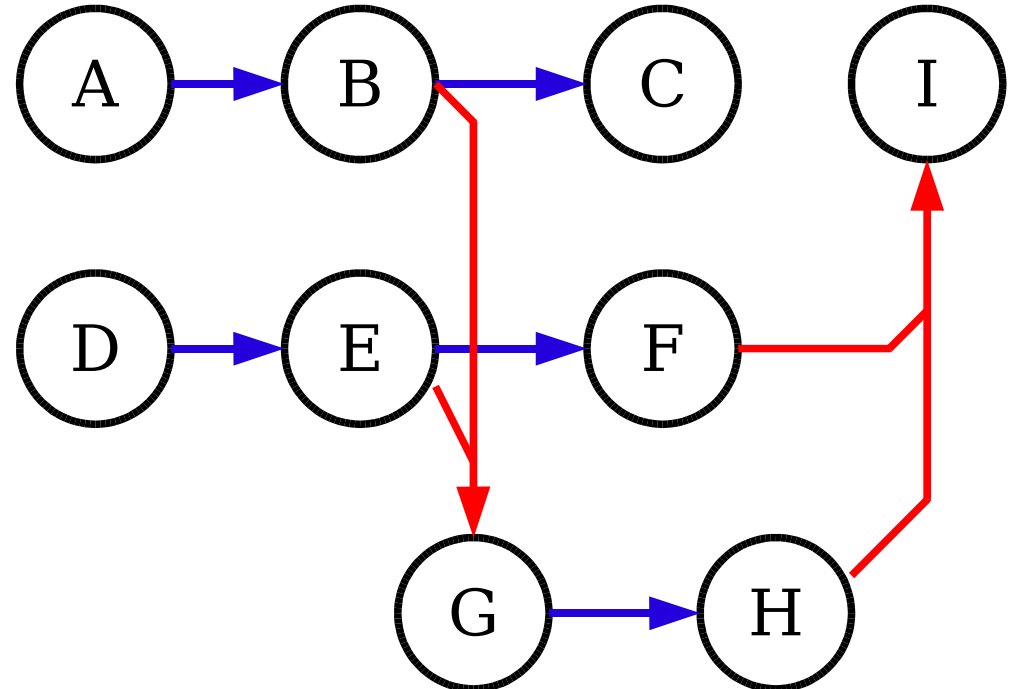
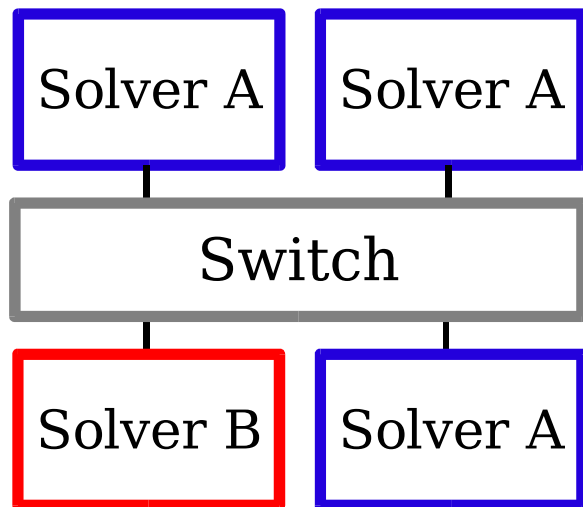
システムの構成

- Optimizer と Scheduler



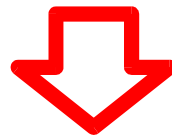
Optimizer: solver set の生成

- FPGA の面積の許す限り並列化
 - 数の多い反応に割り当てる solver を増やす
 - Solver のスループットと反応数を考慮



Optimizer: solver set の再利用

- Solver set の新規構成には CAD が必要
 - 論理合成・配置配線には時間がかかる
 - 2、3時間を要するのが普通
 - 毎回やっていたのでは、高速化にならない!
 - CAD の動く環境は限られている
 - 限られたアーキテクチャと大きなメモリに加えて
 - ライセンスが必要
- 一度使った FPGA の構成情報をリポジトリ化
 - 必ずしも最速ではないが、効率を改善できる



Scheduler: コード生成

- Solver set のメモリに書き込む情報を生成
 - SBML の記述と、solver set の構成がソース
 - $[X]$, k RAM の内容
 - Pathway RAM
 - スイッチの転送制御メモリ (Code RAM)
 - 各物質はいずれかひとつの solver に「所属」
 - タイムステップごとに、利用する他の solver に転送
 - データの効率のよい配置と転送スケジューリングが鍵

Scheduler のストラテジ

- 反応とデータの solver への配置
 - 反応の割り当て
 - 同種の反応をサポートする solver が複数ある場合、
 - 同じ分子を使う反応をなるべく同じ solver に割り当て
 - 分子の割り当て
 - 各分子はいずれかの solver に所属し、
 - その分子をもっとも多くの反応で利用する solver がその所属先 (home) となる
 - 通信のスケジューリング
 - 多くの solver で使われるものから先に転送
 - 転送はマルチキャストが可能

現状

- 各コンポーネントの実装状況
 - Solver core
 - ライブラリ整備中 (MM, Bi-Bi などが実装完了)
 - 仕様を記述する言語 (SCML: Solver Configuration Markup Language) の仕様策定作業中
 - Controller
 - Euler 法を実装完了/Runge-Kutta 法を実装中
 - Stiff な微分方程式も解けるようにしたい
 - Optimizer
 - 未実装
 - Scheduler
 - アルゴリズムの実装完了 (Solver の充実と SCML策定待ち)

性能評価 (1: 演算能力)

- Irreversible MM の実装
 - 旧バージョンの実装
 - パイプラインピッチ = 4 、 solver 数 = 10

システム	クロック (Hz)	速度 (Reactions/sec.)	
		10kdataset	1Mdataset
Xeon	3.06G	38.08	2.97
Pentium4	2.40G	29.55	1.64
Pentium3	1.13G	8.72	1.23
ReCSiP	93M	232.5	

- μP はキャッシュのサイズが性能に大きく影響

性能評価 (2: スケジューラ)

- パイプラインの稼働率
 - 概ね7割程度

	反応の種類	反応数	Solver数	分子の種類	稼働率(%)
A	1	1x900	1x9	100	70
B	3	3x300	3x3	100	75
C	9	9x100	9x1	100	77

まとめと今後

- FPGA による生化学シミュレーション
 - 演算資源の無駄を省くことで高い性能を達成
 - 対象の系に合わせて回路を構成して処理
 - すべてをスタティックにスケジュールする
 - デザインエントリに SBML を利用
 - 他のソフトウェアベースのシミュレータと同等の可用性
 - 今後の課題
 - 演算モジュールの整備と仕様記述言語の策定
 - FPGA 外のメモリを活用できるような手法の開発
 - 複数のFPGAを使う場合のデータ管理、スケジューリング手法の開発