FPGAによる
細胞内代謝系の
マルチモデルシミュレーション

慶應義塾大学 大学院理工学研究科
長名保範 福島知紀 吉見真聡 天野英晴
bio@am.ics.keio.ac.jp

背景

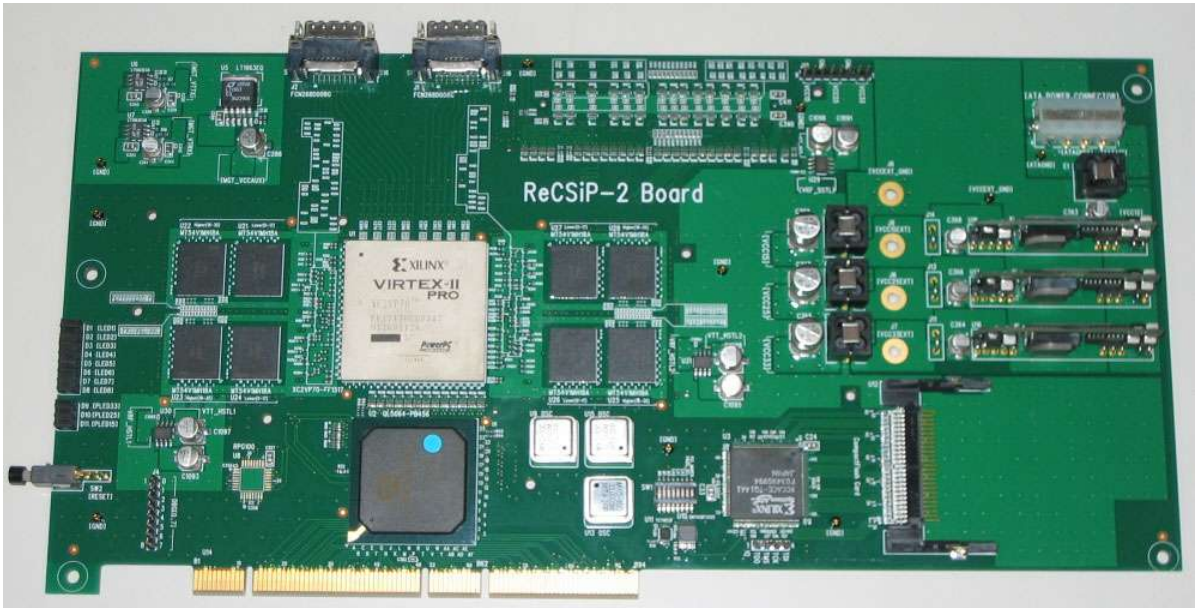
- 背景
  - 計算機を用いた生物学が盛ん
    - 遺伝子配列の解析
    - 細胞のシミュレーション
  - どちらも計算が大変
    - 前者はPCクラスタの利用が主流。確立された技術
    - 後者は生物学・計算機科学ともに、これからの研究分野

目的

- PCクラスタの代替としてのFPGA利用
  - 代謝系シミュレーション+PCクラスタ?
    - 計算の対象がきれいな形でない
    - したがって、効率のよい並列プログラム化は困難?
  - 代謝系シミュレーション＋専用計算機?
    - いろいろな反応モデルが出てくる
    - シミュレーション対象毎に違ってくるかも…
  - 代謝系シミュレーション+FPGA?
    - FPGA内での並列処理による通信ボトルネックの解決
    - FPGAの柔軟性によってさまざまな対象に対応

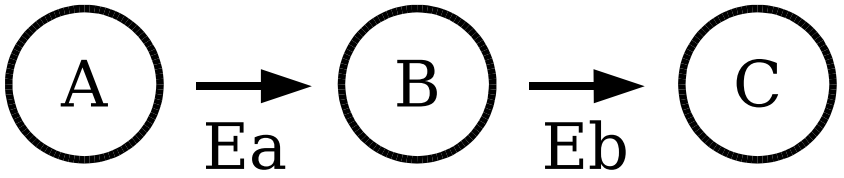
ReCSiP: a Reconfigurable Cell SImulation Platform

- ReCSiP2 ボード
  - 64bit/66MHz PCI with XC2VP70
    - 18Mb QDR-SRAM x 8
    - DDR SDRAM SO-DIMM

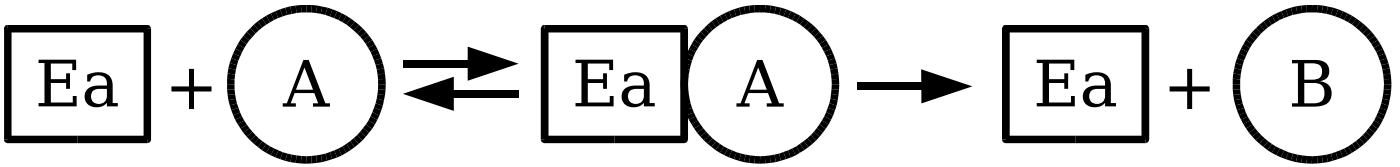


代謝系シミュレーションとは?

- 代謝系は一連の化学反応で構成
  - たとえば、



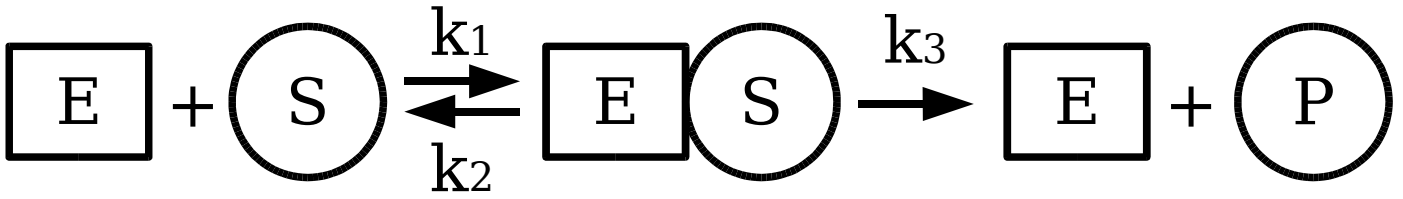
  - AからBに至る反応をモデル化すると



  - BからCの反応も同じ形
    - この形の反応の反応速度を定式化して数値的に解く

反応速度式

- さきほどの反応モデルを一般化
  - Michaelis-Mentenモデル



$$d[S] = -k_1[E][S] + k_2[ES]$$

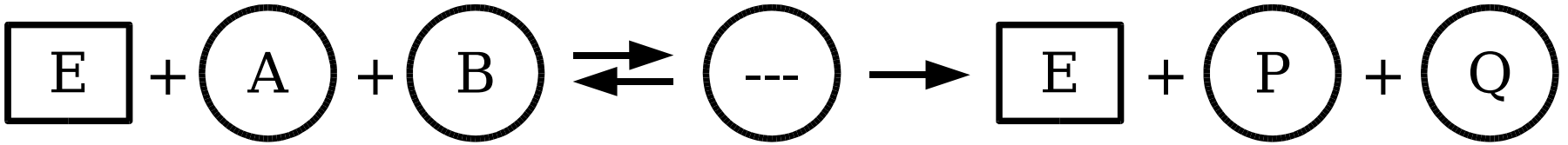
$$d[E] = -k_1[E][S] + (k_2 + k_3)[ES]$$

$$d[ES] = -d[E]$$

$$d[P] = k_3[ES]$$

その他の反応モデルの例

- Bi-Bi Michaelis-Menten
  - 2基質、2生成物
    - 別の反応速度式をたてて解く

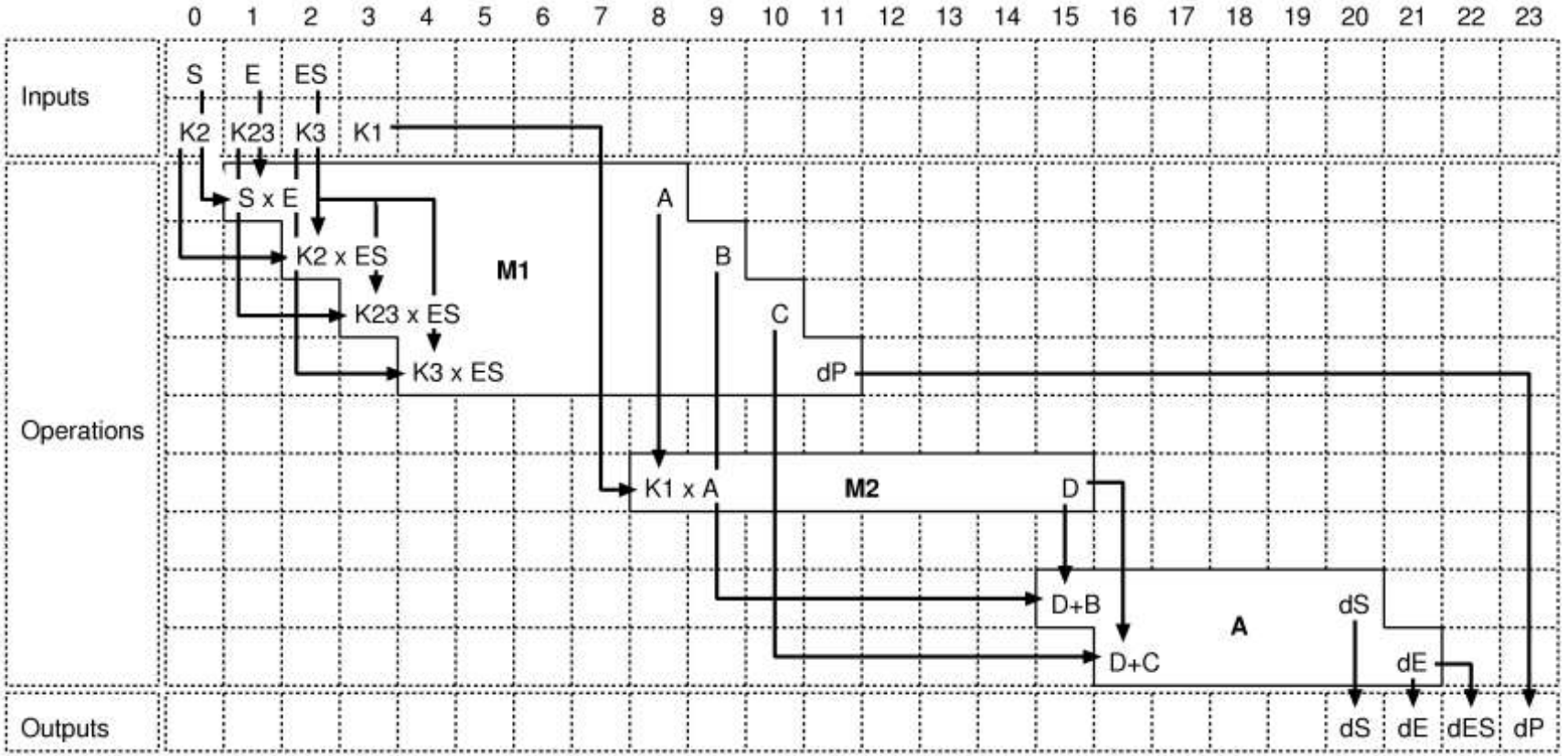


これまでの実装

- 計算モジュール (Solver) の実装
  - 反応速度式を解く Solver Core
    - 反応モデル固定 (Michaelis-Menten型)
      - [E], [S], [ES], [P] から d[E], d[S], d[ES], d[P] を計算
      - d[X] を [X] に足す処理は別
    - 微分方程式の数値解アルゴリズム固定 (Euler法)
  - Solver Core にデータを供給する制御機構
    - 反応経路をポインタ配列によって表現
      - FPGA の自律動作を可能にする
    - PCIバスに接続した場合の通信ボトルネックを回避

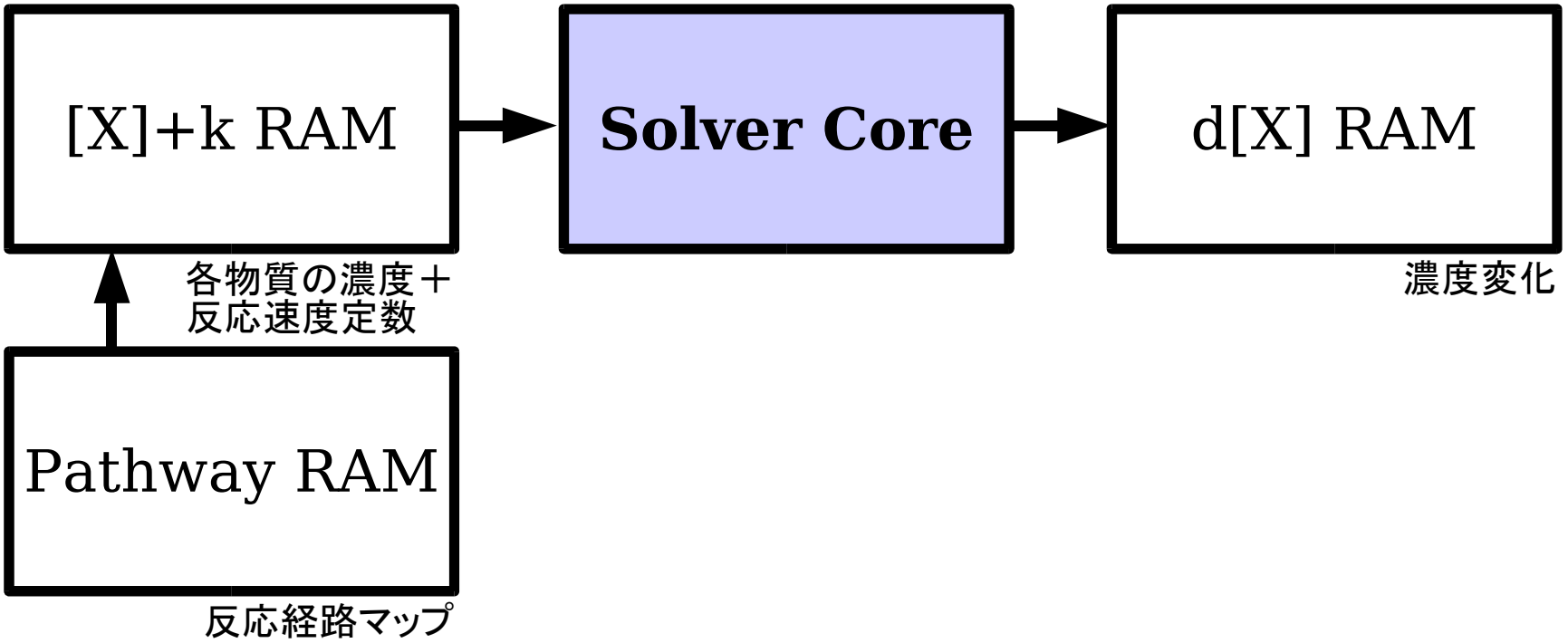
計算モジュールの構成

- 乗算と加算で構成 (単精度浮動小数点)



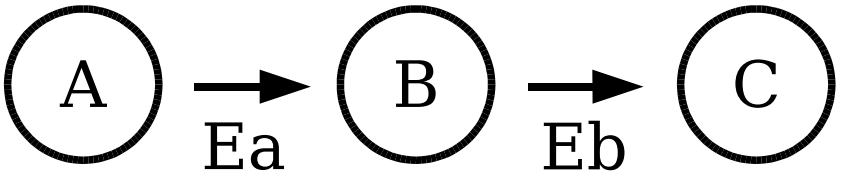
制御機構 (1): 概要

- 3つのメモリを使用
  - BlockRAM を使用



制御機構 (2): 方式

- 反応経路の表現方式 (実装とは異なります)



[X] RAM

1	A
2	B
3	C
4	Ea
5	Eb
6	Ea.A
7	Eb.B

Pathway RAM

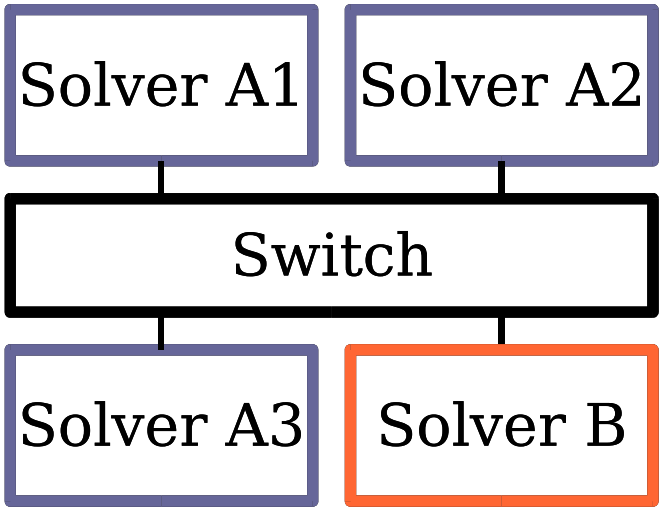
0	4	6
1	1	2
2	5	7
3	2	3

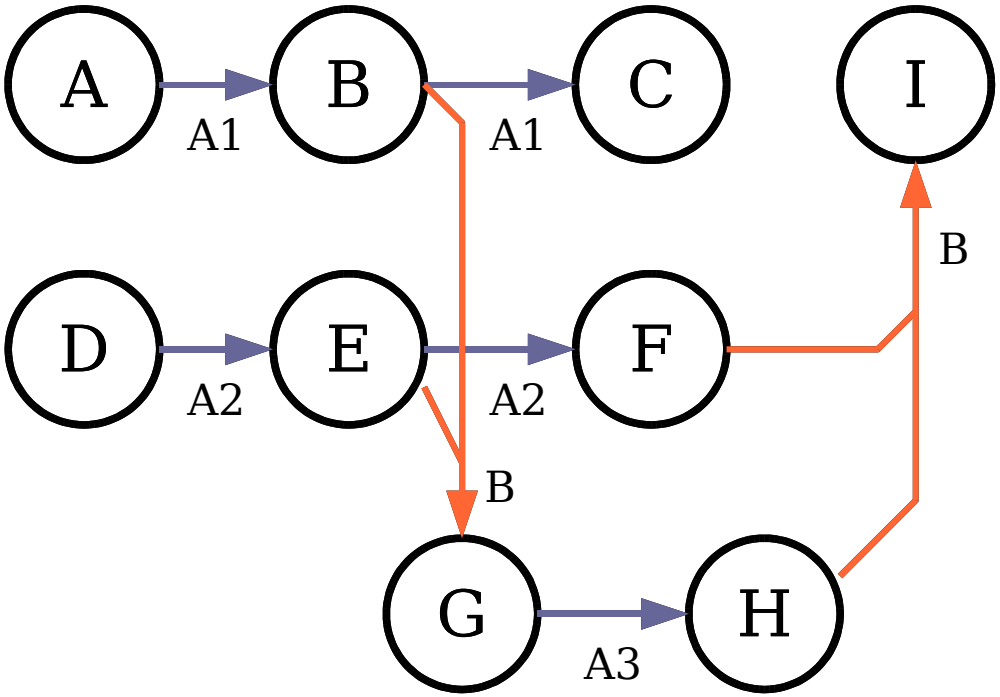
現在の実装

- Solver間の通信機構を実装
  - なぜ必要?
    - 並列化して高速化
      - FPGAの面積を有効活用
    - 複数の反応モデルのsolverを相互結合
      - 柔軟なモデル構築に必要
  - 評価
    - 実際のシミュレーションモデルを載せるのはまだ困難
      - モデル構築に膨大な手間が必要
    - 通信機構の追加による性能へのダメージを調べる

全体像 (マルチモデルシミュレーション)

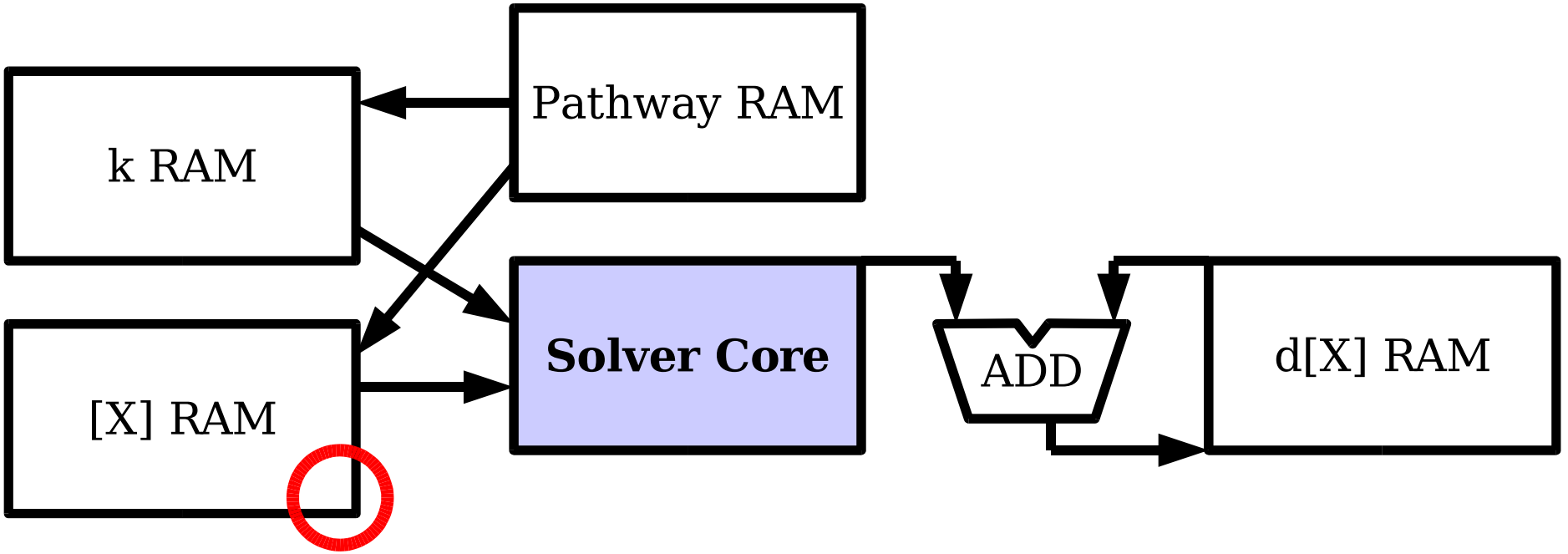
- スイッチで複数の solver を結合
  - 反応系をいくつかに分割して割付け
    - 反応の種類と反応系のサイズで切る





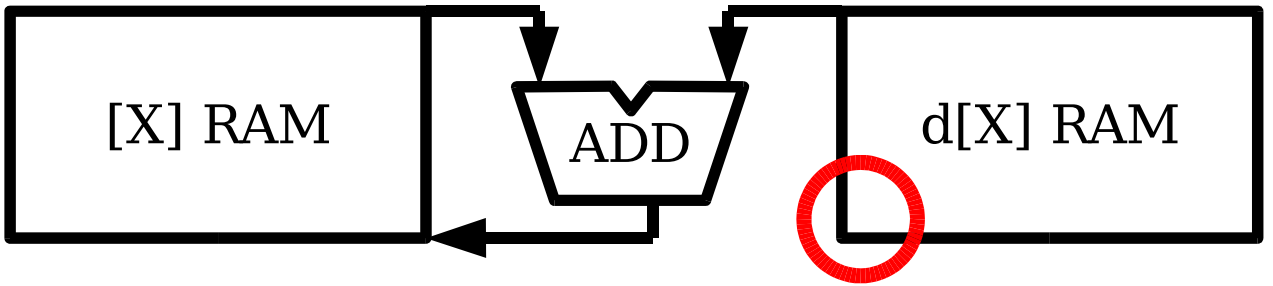
Solver の動作 (フェーズ1)

- 各反応に寄与する物質の濃度変化を計算
  - d[X] RAMでのアドレスは [X] RAM と同じ
    - [X] RAM 間でのデータ転送が必要



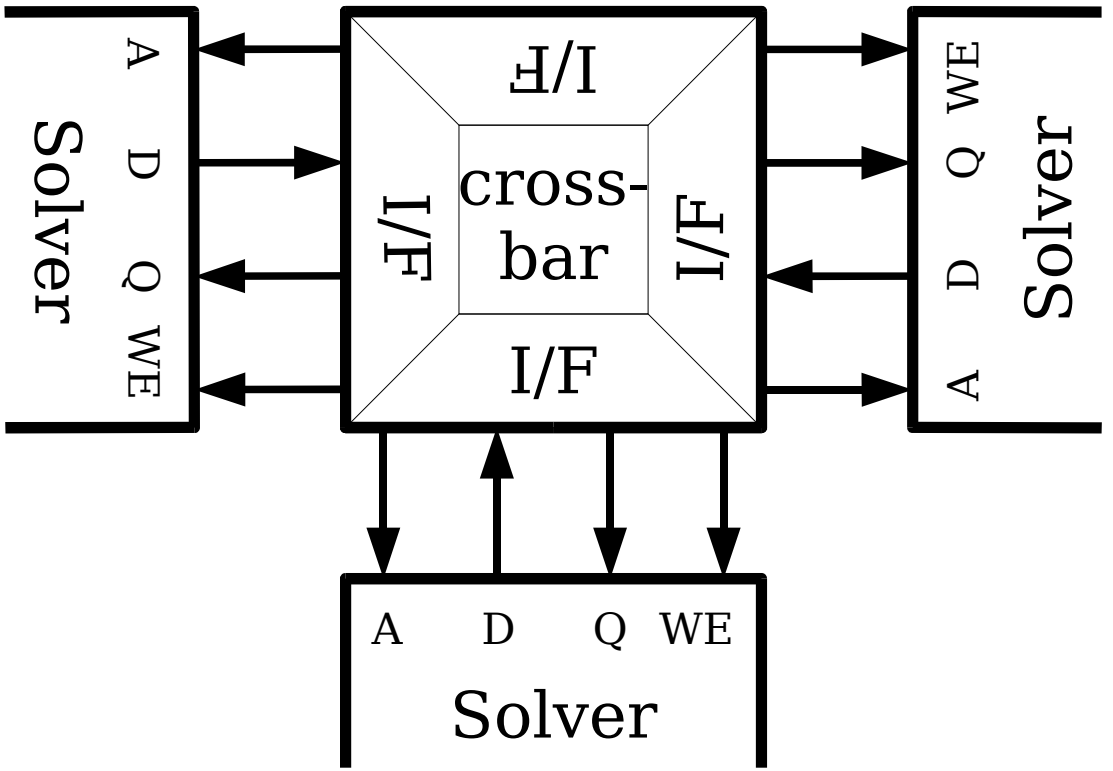
Solver の動作 (フェーズ2)

- d[X] RAM から [X] RAM へ
  - 濃度変化を [X] RAM へ反映
    - FPADD を用いて [X] RAM に加算する
    - [X] と d[X] が同じアドレスなので処理は単純
    - d[X] RAM 間でのデータ転送が必要



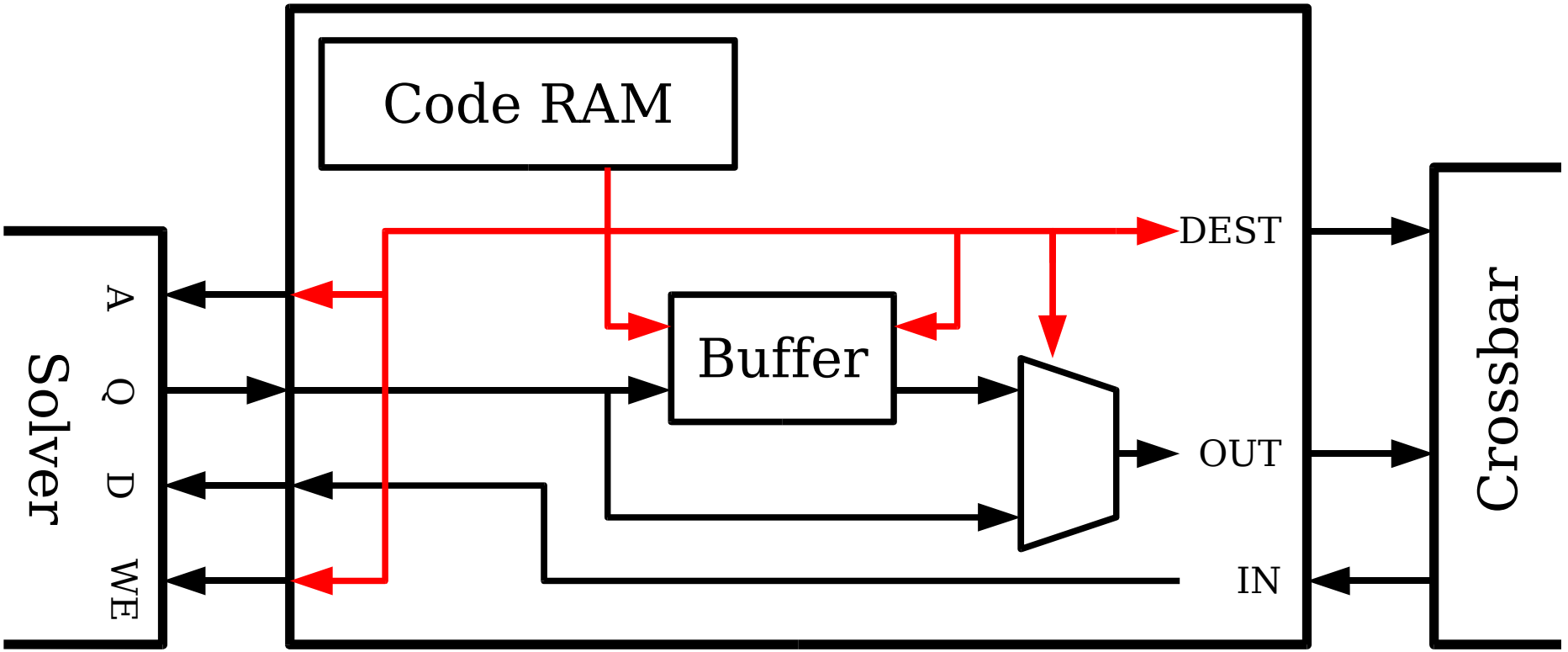
通信機構の概略

- メモリの空きポートを使って接続



インターフェイス部の構成

- 構成

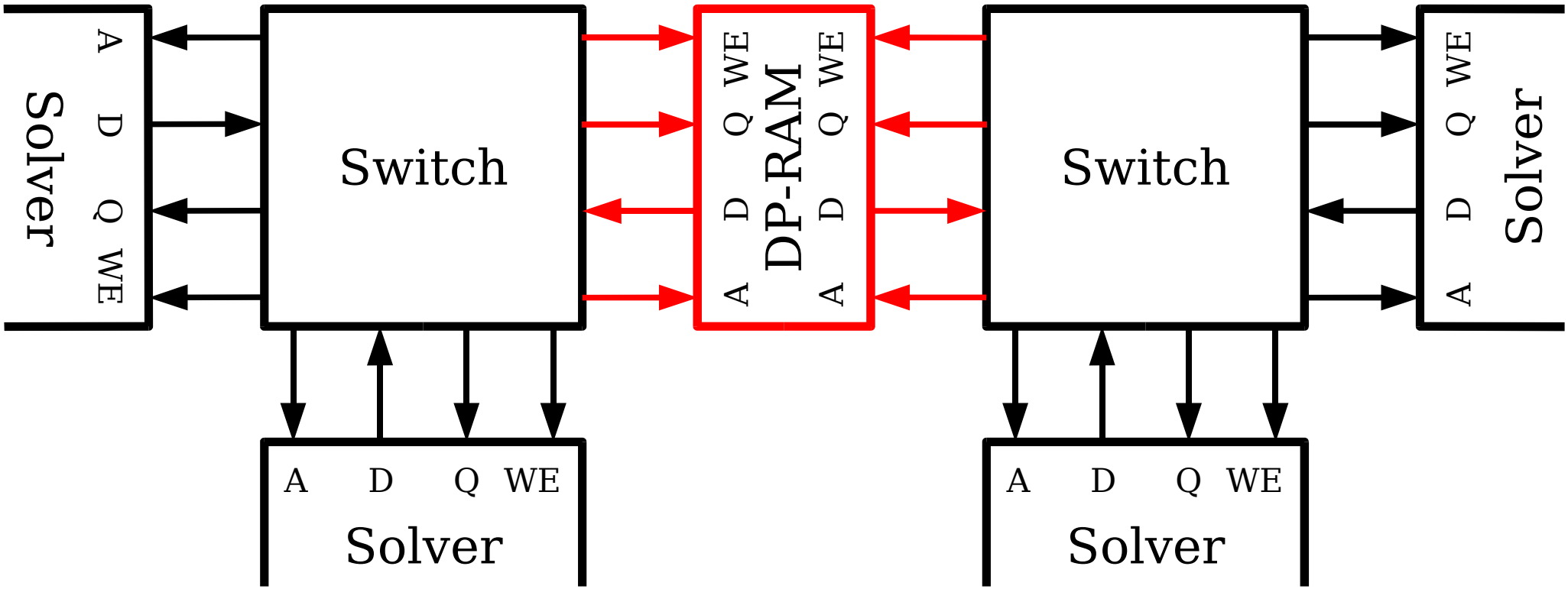


Code RAM の構成

- データ転送を制御するメモリ
  - 0 番地から順番に読み出される
    - Solver に渡すアドレス (11bit)、Write Enable (1bit)
    - バッファのSolver側アドレス (4bit)と Write Enable (1bit)
    - バッファのクロスバ側アドレス (4bit)
    - クロスバに渡す転送先(9bit: 1ノードにつき1ビットでマルチキャスト可)
    - 終了通知ビット (1bit)
      - 合計 31bit

複数のスイッチの使用

- 間に DP-RAM をはさんでスイッチ間接続

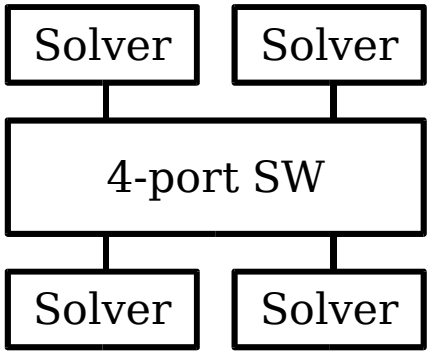


評価 (単体)

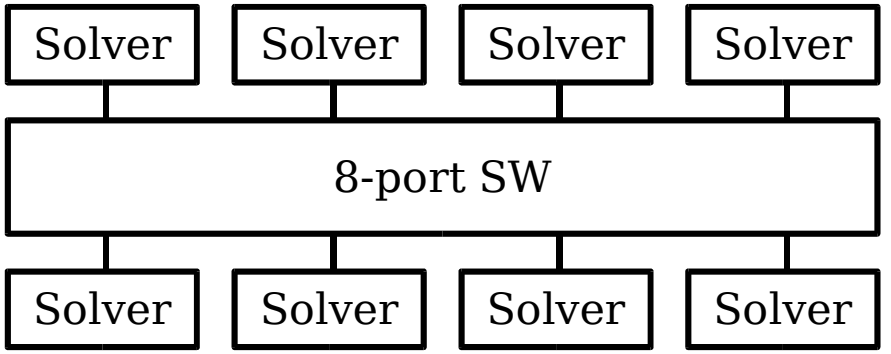
- Solver 単体での性能評価
  - ソフトウェア (Pentium3) との比較
  - ターゲットデバイス = XC2VP70-5FF1517
  - 結果
    - Software: 10.15Mops @ 1.13GHz
    - Hardware: 28.14Mops @ 112.56MHz

評価 (スイッチ付き: 1)

- スイッチを使った場合
  - 4 solvers + 4-port SW
    - 99.95MHz
    - 99.95Mops

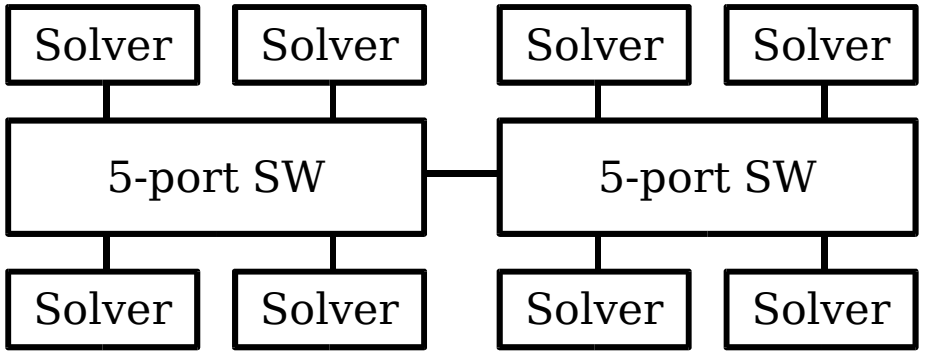


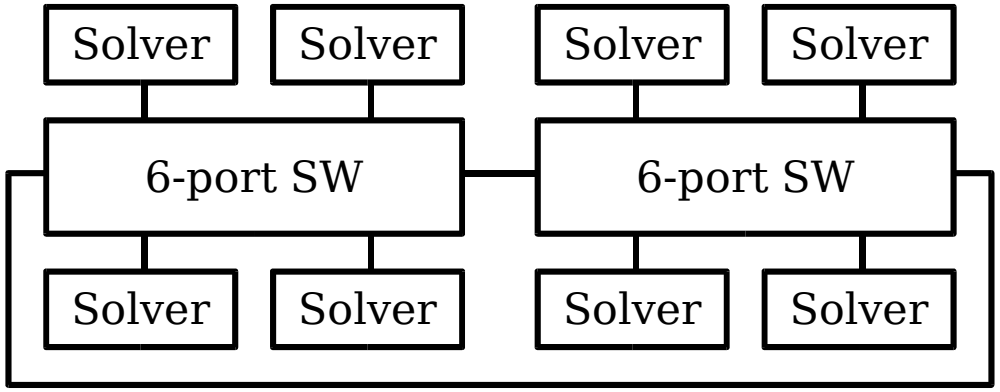
  - 8 solvers + 8-port SW
    - 91.24MHz
    - 182.48Mops



評価 (スイッチ付き: 2)

- スイッチを使った場合
  - 2 x (4+5port) (横並び)
    - 84.81MHz
    - 169.62Mops
  - 2 x (4+6port) (リング)
    - 87.53MHz
    - 175.06Mops





評価 (まとめ)

構成	[MHz]	[Mops]	Slices	MULT / BRAM
Software	10.15	10.15	N/A	N/A
1	112.56	56.28	5%	7%
4	99.95	99.95	21%	33%
8	91.24	182.48	43%	67%
4+4 (横並び)	84.81	169.62	49%	71%
4+4 (リング)	87.53	175.06	50%	75%

- 8ポートスイッチは slice 3%, BRAM 11% を消費
  - スイッチの導入によりクロックレートは約 19% 低下
- 性能的には 4+4 よりも 8 ポート構成のほうが有利

まとめ

- シミュレーションを行うための方式の提案と実装
  - Solver 間を接続するスイッチの実装
    - マルチモデルなシミュレーションが可能に
    - Michaelis-Menten モデルの solver 実装例
  - 動作周波数とスループット
    - 8 個程度では、単一のスイッチで接続したほうがよい

今後の予定

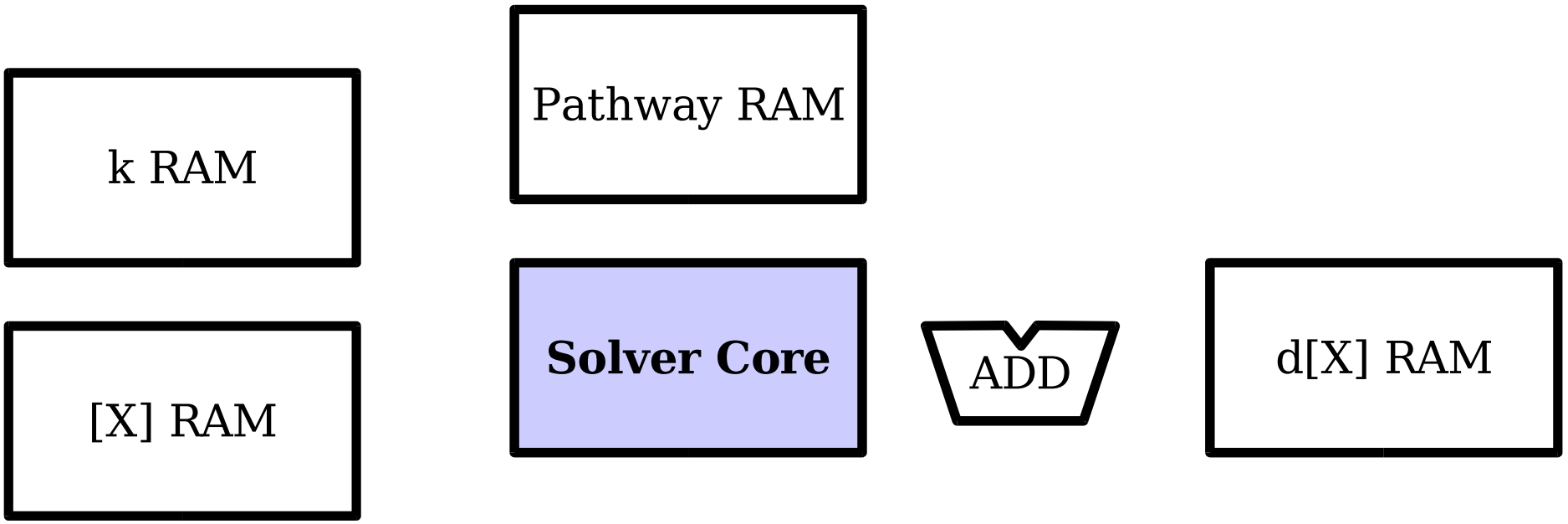
- 短期的予定
  - その他のsolverの実装
    - 他の反応モデル
    - Runge-Kutta 法などを使う
  - モデリングソフトウェアの開発
    - CodeRAM の中身を手で書くのは大変
- 長期的予定
  - 複数のFPGAを用いたシステムへの拡張
  - 実モデルを用いた検証
    - ヒト赤血球など

おしまい

ありがとうございました。

全体の構成 (予備)

- Solver core 周辺の回路を再構成



反応経路の分割

- 反応モデルが違う場合
- 計算時間を短縮したい場合
  - 同じ反応モデルでも複数の反応を別々の solver に割り当てることで性能向上が可能

問題点

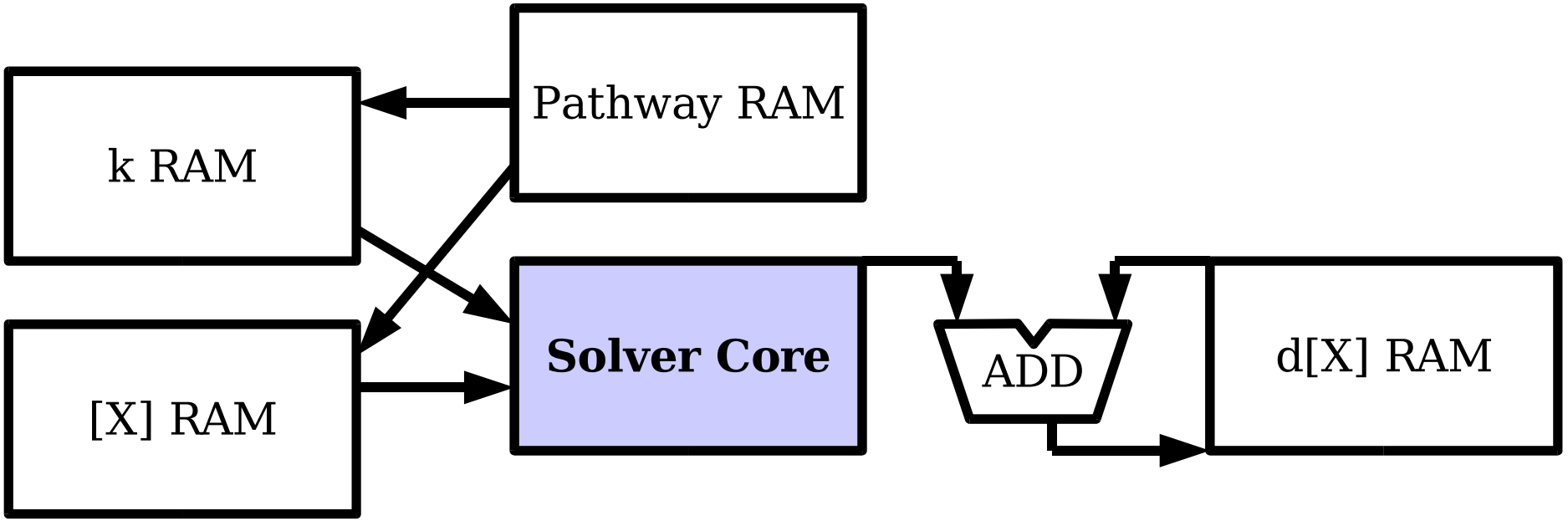
- PCIバスがボトルネック
  - CPU単体でシミュレーションをした場合
    - PentiumIII 1.13GHz で 10.15Mreactions/sec.
  - CPU主導でシミュレーションをした場合
    - FPGA と CPU は PCI を介して通信
      - PCI バスの通信速度は 64bit x 66MHz
      - 4物質の濃度と3つの反応速度定数を相互に交換
        - 1反応あたり 32bit x (4+4+3)
    - PCI の最大バンド幅で 9.42Mreactions/sec.

これまでの経過 (ステップ2)

- 簡単なシミュレータのFPGA実装 (長名,2003)
  - 単一の処理ユニット
    - ひとつでPentiumIII の 2〜4倍程度の性能
    - 面積的には、Xilinx XC2V6000に8つ載せられる
    - だが、8つは独立にしか動作しない
  - 演算回路の独立動作のための機構
    -

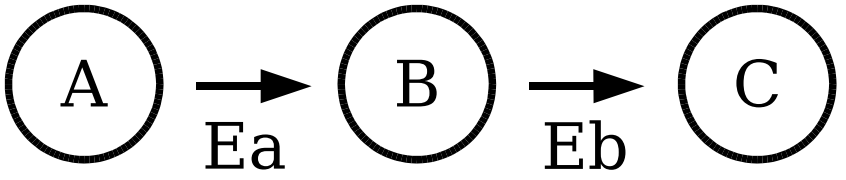
新しい solver の構成

- Solver core 周辺の回路を再構成
  - [X] RAM と k RAM を分離してバンド幅を確保
  - 専用の FPADD を追加して積分操作などに利用



Solver の動作 (フェーズ1)

- FPADD の役割
  - 複数の反応に寄与する物質(たとえばB)の濃度変化をまとめて d[X] に格納する

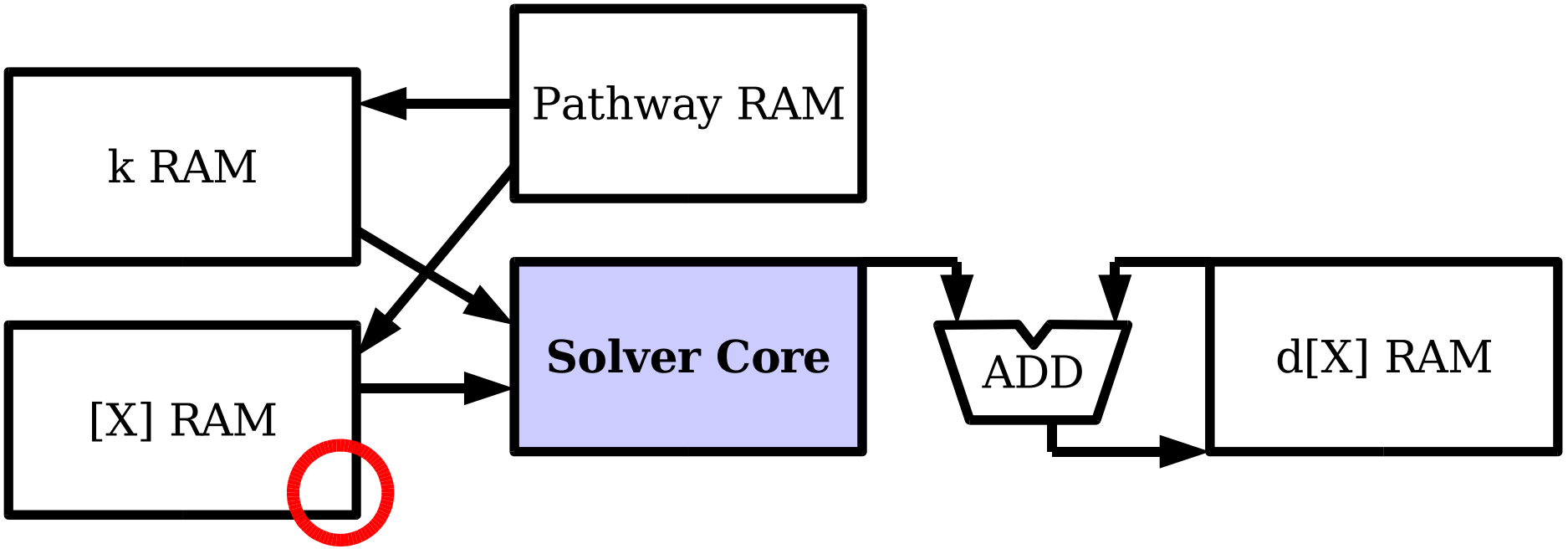


Solver 間通信機構

- 通信機構とは?
  - 求められるもの
    - Solver 間での物質の共有を可能にする機構
    - Solver の中身(アルゴリズム)に依存しない
      - 複数のモデルのsolverを組み合わせて利用可能に
  - 具体的な機能
    - 動作フェーズ 1 で、[X] RAM 間の任意データのコピー
    - 動作フェーズ 2 で、d[X] RAM 間の任意データのコピー

通信の方式 (1)

- Phase 1
  - [X] RAM が1ポート空き



通信の方式 (2)

- Phase 2
  - d[X] RAM が1ポート空き