

耐劣化故障性を有する並列演算機構に関する研究

Achieving degradation tolerance in a hardware accelerator with parallel processing elements

米田 友洋
Tomohiro YONEDA

どんな研究？

半導体プロセス技術の進歩に伴い、大規模でディペンダブルなVLSIを実現する上で、今までにないタイプの故障が問題となりつつあります。この研究は、使用中のストレスに起因するチップ内の局所的な性能劣化に着目し、それに対する耐故障性を実現する一手法を提案するものです。

何が特徴？

非同期式回路は、ローカルなハンドシェイクにより演算制御を実行するため、演算器が劣化しても常に正しい出力が得られます。本研究ではさらに、劣化した演算器に流すデータ量を自律的に減少させ、正常な演算器により多くのデータを処理させることで、全体の性能を維持します。

基本的アイデア

- ◆ 微細化に伴うコア内の局所的動的性能劣化
 - 素子の物理的な劣化
 - 例) NBTI(Negative Bias Temperature Instability)
 - 電源電圧低下
 - 温度上昇
- ◆ 影響
 - 突然の誤動作
- ◆ 従来の対策
 - 十分なマージンを取っておく → 性能向上に悪影響
- ◆ 大きなマージンを取ることなく、また、劣化が生じて、自律的にそれに耐える機構が必要
- ◆ 前提
 - 同種の演算器が複数使われているような並列演算機構を対象
 - 劣化部分を同種の正常演算器で補うため
- ◆ アプローチ
 - 非同期式回路技術
 - クロックを用いていないので、劣化が生じても演算結果は正しい。ただ、遅れるだけ
 - 自律的な、劣化検出と演算割り当て変更

並列演算機構の例

- ◆ 線形1次方程式のソルバを反復法で実現
 - Gauss-Seidel法

$$\begin{bmatrix} a_{11} & \dots & a_{1i} & \dots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{i1} & \dots & a_{ii} & \dots & a_{in} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \dots & a_{ni} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=0}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right)$$

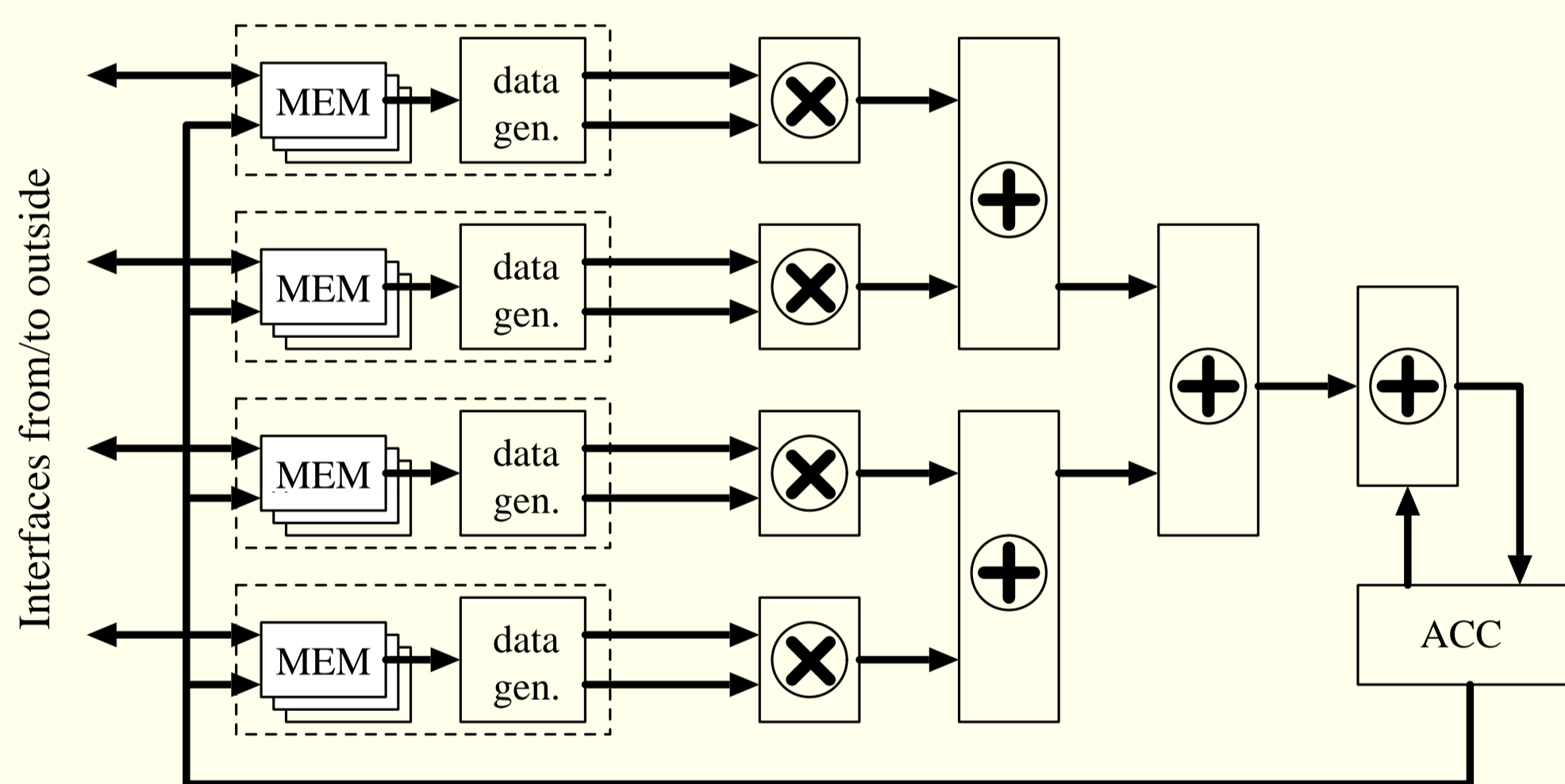
$A_{ii} = \frac{K}{a_{ii}}$ および $A_{ij} = -\frac{Ka_{ij}}{a_{ii}}$ を行列 A に求めておく

$$x_i^{(k+1)} = \frac{1}{K} \text{round}(A_{ii} b_i + \sum_{j \neq i} A_{ij} x_j)$$

変数: 符号付16ビット整数, 演算: 符号付31ビット整数
 $K=2^{15}$ であり、四捨五入して上位から16ビット取ることにより $x_i^{(k+1)}$ が求まる

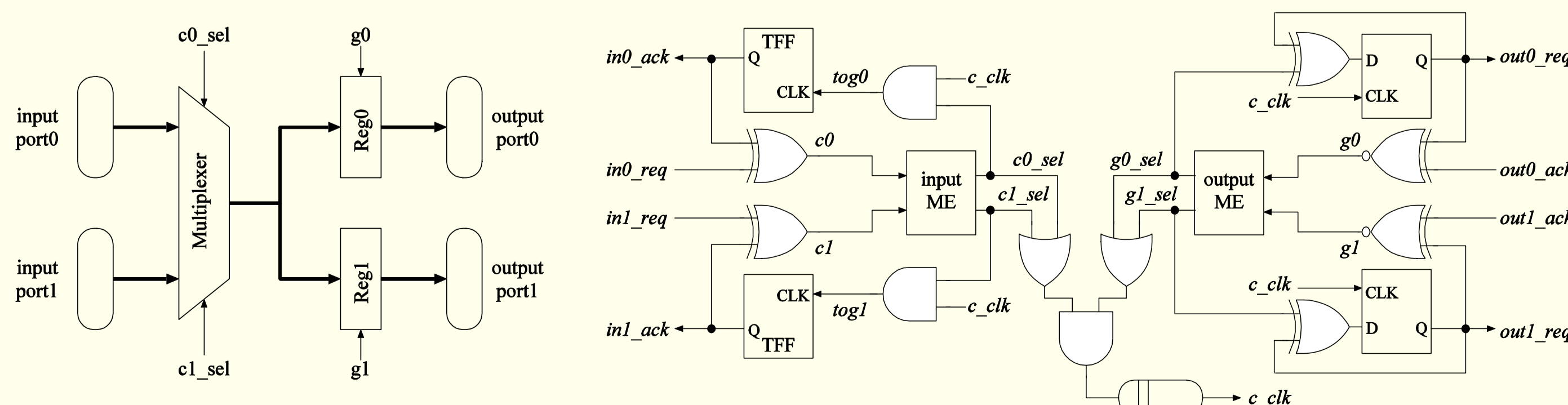
具体的な実現手法

オリジナルデザイン

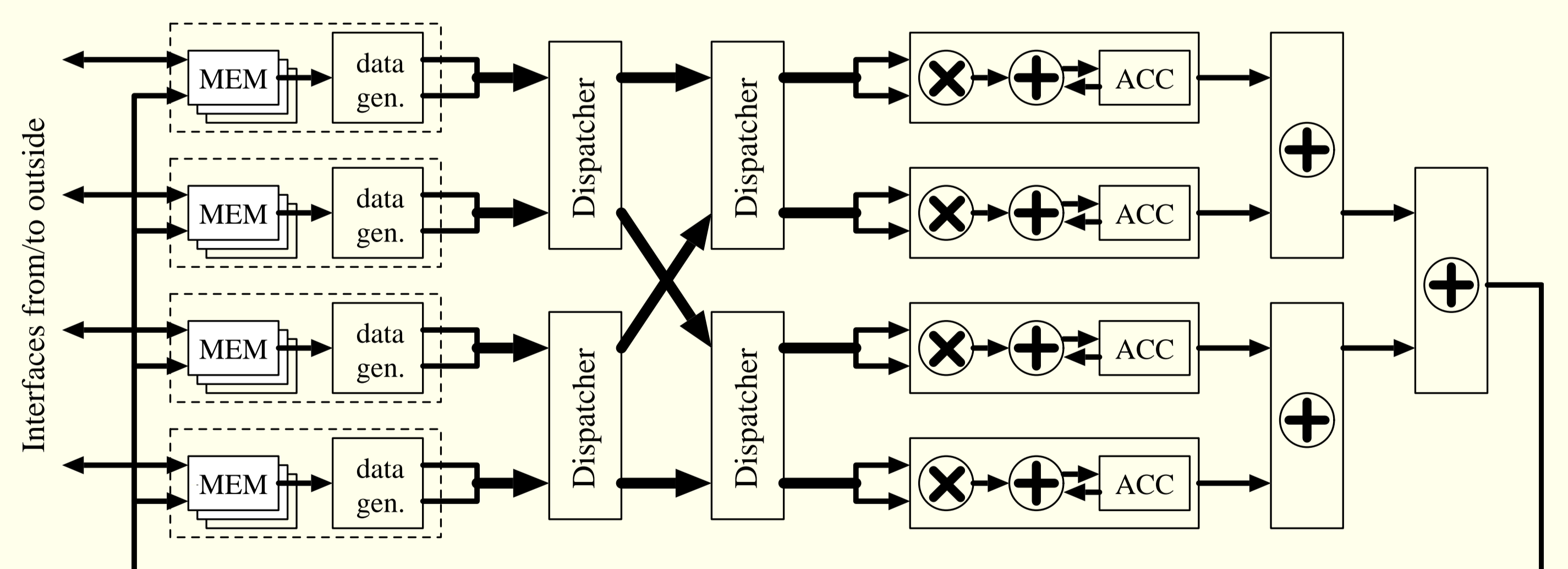


Dispatcher部

- ◆ 負荷の均等化
 - 出力バッファを各方向に持たせる
 - 2つの入力データは入力アービタで調停
 - Ackが帰ってきている側の出力バッファに、入力データを送る
 - 両方ともAckが帰ってれば、任意のバッファに入れる
 - 両方ともAckが帰ってきていなければ、どちらかが帰るまで待つ
 - アービタが必要(出力アービタ)
- 自律的に負荷の均等化が実現



非同期化+劣化検出・自律的割当て変更



演算器部

