

# 超並列プロセッサコアにおけるPE間データ転送効率の改善

中野 光臣<sup>†</sup>, 弘田 澄男<sup>†</sup>, 児玉 章宏<sup>†</sup>, 飯田 全広<sup>†</sup>, 末吉 敏則<sup>†</sup>

<sup>†</sup> 熊本大学大学院自然科学研究科  
〒 860-8555 熊本県熊本市黒髪 2-39-1

多数の PE (Processing Element) を持つ並列プロセッサにおいて, PE 間を接続する相互結合網は重要な構成要素であり, データ転送処理はアプリケーションの実装において性能に影響を与える. 本研究では, 株式会社ルネサステクノロジ社が開発した SIMD 型アクセラレータである MX コアにおいて, SIMD 型および MIMD 型の PE 間データ転送制御での配線構造の定式化を行う. また, データ転送問題を 2 つの部分問題に分け, ルーティング手法を提案し評価を行う. 規則的な転送処理および複雑なデータ転送処理を用いて評価を行った結果, 両制御において効果的なデータ転送制御が行えることを示す. また, 複雑なデータ転送処理において, MIMD 制御の転送回数は SIMD 制御時より 90%削減することができた.

## Improvement of data transfer efficiency between PEs in a massively parallel processor

Mitsutaka Nakano<sup>†</sup> Sumio Hirota<sup>†</sup> Akihiro Kodama<sup>†</sup> Masahiro Iida<sup>†</sup> Toshinori Sueyoshi<sup>†</sup>

<sup>†</sup> Graduate School of Science and Technology, Kumamoto University  
2-39-1 Kurokami, Kumamoto-shi, 860-8555 Japan

An interconnection network between PEs is an important component in parallel processors. The data transfer processing with it has a serious influence on the performance of applications. MX core developed by Renesas technology Corp., is a massively parallel SIMD type accelerator. We formulate the data transfer control of the MX core both of SIMD type and of MIMD type. The data transfer problem is divided into two problems, and we propose both solutions of them. Our proposals are able to perform effective data transfer control by either two types. As an evaluation result, the MIMD control was reduced the number of transfer counts by 90% from SIMD control.

### 1 はじめに

多数の PE (Processing Element) を持つ並列プロセッサにおいて, PE 間相互結合網は重要な構成要素であり, アプリケーションの実装において性能に重要な影響を与える<sup>1)</sup>. また, 多数の PE を持つ場合, 手動でのルーティングは困難である. そこで, PE 間データ転送における問題点の定式化と効率の良いルーティングアルゴリズムが必要である.

我々は, 株式会社ルネサステクノロジが開発した超並列 SIMD (Single Instruction/Multiple Data) 型プロセッサコアである MX コア<sup>2)</sup> に注目している. MX コアは 1,024 個の 2 ビット PE と制御回路等からなり, 各 PE には 512 ビットのデータレジスタを 2 セット持つ. MX コアは SIMD 型プロセッサコアであるため, 性能を出すには処理の並列性の抽出および PE 間のデータ転送の効率化が必要不可欠である. 本論文ではアプリケーション実装において重要となる PE 間配線構造, および

データ転送効率改善手法について議論する. 以下, 2. で MX コアの詳細を示す. 3. では PE 間データ転送の特性と問題点について述べ, 4. では PE 間相互配線の定式化および提案するデータ転送手法について述べる. 5. で評価方法を示し, 6. で評価結果および考察, 7. にてまとめる.

### 2 MX コア

図 1 に MX コアの構成を示す. MX コアは, PE とデータレジスタをあわせたエントリで構成される. 各 PE 間のデータ転送は, 垂直方向のデータ転送用配線 (V-ch) を使用する. V-ch は, 一定の距離にある PE 間のデータ転送を並列にすることが可能である. PE は, 1 ビットフラグ用レジスタ S, F, D, C, V, N, アキュムレータ X, XH および 2 ビット ALU から成る. また, X, XH は PE 間転送を制御している ECM (Entry CoMmunicator) を介して V-ch とも接続されている. C はキャリー, S, F および D は演算における各種処理レジスタ

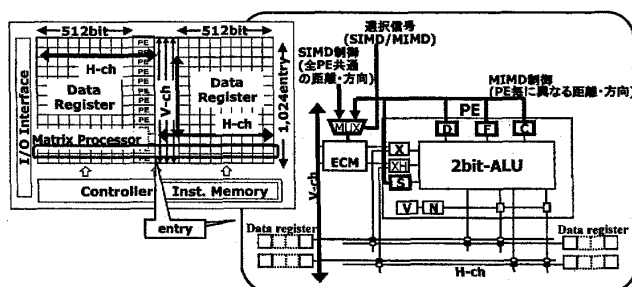


図1 MX コアの構成

として動作する。V と N は Valid フラグとして、データ転送や演算時のマスクの指定に使用する。

PE 間データ転送において、SIMD 制御では PE 毎に転送する距離が異なる場合、逐次的に転送を繰り返す必要があり、性能低下の要因となる。そこで、我々は各 PE が個々に制御可能な MIMD 制御手法を提案している<sup>3)</sup>。各 PE 内に設けられた S, F, D, C レジスタは、データ転送処理には使用されないため、これらのレジスタを通信先の距離と方向を制御する制御用レジスタとして利用する。太線部が MIMD 制御部である。このように、通信制御用ビットを格納するレジスタとして既存のものを利用することで、追加資源を極力抑えて MIMD 制御 PE 間データ通信を実現している。

### 3 PE 間データ転送の特性と問題点

図1のように、MX コアの PE-データレジスタ間は H-ch により、PE 間は V-ch によりそれぞれ配線されている。MX-SoC に搭載されている MX コアの V-ch の配線長は4のべき乗 1, 4, 16, 64, 256 であり、一回の PE 間転送処理でこれらの距離のデータ転送が可能である。任意の距離のデータ転送は、上述の配線の組合せで実現している。

#### 3.1 SIMD 制御におけるデータ衝突

図2にデータ衝突の具体例を示す。1回目の転送においてデータ A, B を転送した後、2回目の転送でデータ C, E に対して上方向に転送距離1の転送を行うとする。しかし、エントリ2の転送先に1回目に転送を行ったデータ A が存在しているため、データの上書きが起こる。上書きを回避するデータ転送処理を行う必要があるため、エントリ2, 4 を一度に転送できず転送回数が増加する。

#### 3.2 MIMD 制御におけるデータ衝突

MIMD 制御では各 PE が転送距離を選択でき、毎回全 PE が動作するため、SIMD 制御のようなデータ衝突は起こらない。しかし、複数の PE の転送先が同じ PE になる場合がある。MIMD 制御

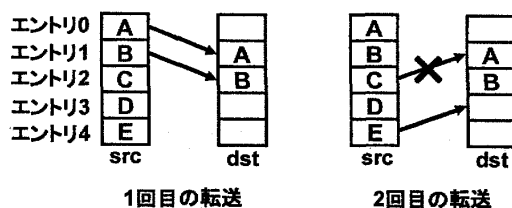


図2 データ衝突の例

では、このような異なるデータ衝突を回避する必要がある。動作する PE の数が多くなるほどこのデータ衝突が起こる確率が増し、回避することが困難となってくる。

## 4 提案手法

本章では、MX コアのデータ転送処理問題を定式化し、データ転送の効率化手法を提案する。我々はデータ転送問題を2つの問題に分割する。第一の問題は、全エントリが目標状態へデータを転送するために必要な転送処理要素を決定し、全体の転送回数を最小化する組合せを求める問題である。これを転送処理要素設定問題と呼ぶ。第二の問題は、転送処理要素設定問題によって得られた転送の組み合わせについて、データ衝突を回避し、全エントリのデータ転送が最小回数で終了するように転送処理要素の順序を決定するスケジューリング問題である。次節より、これらの問題の解法を SIMD 制御、MIMD 制御それぞれについて述べる。

### 4.1 SIMD 型転送処理要素設定問題の定式化

以下に転送処理要素設定問題を定式化する<sup>4)</sup>。

$$D_n = 1x_{n1} + 4x_{n2} + 16x_{n3} + 64x_{n4} + 256x_{n5} - 1x_{n6} - 4x_{n7} - 16x_{n8} - 64x_{n9} - 256x_{n10} \quad (n = 1, \dots, n) \quad (1)$$

式(1)の  $D_n$  は、 $n$  エントリのデータが目標状態へ達するまでの転送距離を表している。各項の係数は、データ転送を行う際に選択することができる転送距離を表しており、各項の変数  $x_{nm} \geq 0$  は各転送処理要素の個数を示す。よって、式(1)は各エントリの転送距離を転送処理要素の組合せに分解することを意味する。式(1)は配線構造の制約条件であり、SIMD 制御、MIMD 制御共通である。

$$\text{minimize } z = \sum_{m=1}^{10} y_m \quad (2)$$

$$y_m = \max\{x_{1m} \dots x_{nm}\} \quad (m = 1, \dots, 10) \quad (3)$$

式(1)、式(3)は転送処理要素設定問題の制約条件である。式(3)は、全エントリのデータが目標状態まで到達するために各転送処理要素を何回行うかを示す。式(2)は本問題の目的関数である。データ転送を行う全データが目標状態へ到達するまで

表 1 転送例

初期状態	目標状態	転送距離	転送処理要素
0	1	1	$+1(J_1)$
1	6	5	$+1(J_2), +4(J_3)$
2	0	-2	$-1(J_4), -1(J_5)$

に処理する各通信要素の個数  $y_m$  の総和を  $z$  とする。転送処理要素設定問題の目的は  $z$  を最小とする  $x_{nm}$  の値を割り当てることである。

#### 4.2 転送処理要素設定問題の最適解法

ここで、あるエントリが目的状態に到達するとき使用する転送処理要素の組合せを転送パスと呼ぶ。最小回数で到達できる転送パスをそのエントリにおける最短パスとする。転送を行うとき、SIMD 制御では同距離同方向のデータ転送のみ一律に動作することができる。したがって、全エントリで最短パスを用いても、その組合せが SIMD 制御にあわなければ逐次処理になり、全体の転送処理回数が増加する。よって、最短パス以外も考慮に入れ、全エントリが完了する最小転送回数を解とする。

#### 4.3 SIMD 型スケジューリング問題の最適解法

転送パスの処理順序決定には、分枝限定法を併用した深さ優先探索<sup>5)</sup>を用いる。表 1 の転送例を用いて、転送パスのスケジューリングの具体例を示す。表 1 において、 $J_1$  と  $J_2$  は並列に処理できるので、これをグループ化し、 $G_1$  とする。この最短転送パスを用いて、 $J_1, J_2, J_3, J_4, J_5, G_1$  の全探索を行い、全エントリのデータが目標状態に到達するまでの最小転送回数を求める。

#### 4.4 MIMD 型転送処理要素設定問題の定式化

転送処理要素設定問題を次のように定式化する。

$$\text{minimize } z = \max\{y_1, y_2, \dots, y_n\} \quad (4)$$

$$y_n = \sum_{m=1}^{10} x_{nm} \quad (n = 1, \dots, n) \quad (5)$$

SIMD 制御の場合と同様に式 (1)、式 (5) が制約条件である。MIMD 制御において異なる部分は、目的関数である式 (4) と制約条件の式 (5) となる。式 (5) は、 $n$  エントリが目標状態まで到達するために必要な各転送処理要素の回数の和を  $y_n$  で表している。 $y_n$  の中で最大の処理数を  $z$  とする。この問題は、 $z$  を最小とする  $x_{nm}$  を導出することである。

##### 4.4.1 転送処理要素設定問題の最適解法

各エントリの転送パスの組合せから、 $z$  を最小にする  $x_{nm}$  の組合せを探索する。SIMD 制御と異なる点は、各エントリが独立して転送距離を選択できるので、最短パスの組合せが処理要素設定問題

の最適解となることである。そのときの  $z$  は全エントリにおける最短パスの最大値である。

#### 4.5 MIMD 型スケジューリング問題の最適解法

MIMD 制御におけるスケジューリングでは、MIMD 制御におけるデータ衝突を回避し  $z$  を増加させないことが重要である。スケジューリングにおけるデータ衝突回避法として、最短パスが  $z$  未満のエントリに着目する。転送回数が  $z$  以下のエントリにおいて、転送回数が  $z$  までであれば自由に処理要素の組合せを変更することが可能である。したがって、転送回数が  $z$  以下のエントリにおいて、データ衝突の可能性があれば転送処理要素の組合せを増やすことで、異なるエントリへ迂回し回避する。データ衝突回避手法によって、転送回数が  $z$  以下のエントリは最短パス以外のパスについても考慮する必要がある。これは SIMD 制御と同様だが、何番目の短いパスまで考慮すればよいといったものはない。そのため、SIMD 制御よりも探索範囲が広がることになり、解を導出することが難しくなる。そこで、転送回数が  $z$  に近いものから優先して探索を行う<sup>6)</sup>。探索には SIMD 制御と同様に深さ優先探索を用いる。

### 5 評価方法

評価は、ルネサステクノロジ社提供の MX シミュレータ Ver.0.03.01 およびこのシミュレータに MIMD 制御手法を追加したものを用いる。評価方法として、実際のアプリケーションに現れるデータ転送処理を用いる。評価対象は RSA 暗号、および、MP3 デコーダ<sup>7)</sup> 内に含まれるアンチエイリアス処理、IMDCT(Inverse Modified Discrete Cosine Transform) 処理に必要なデータ転送処理の 3 種類である。また、単純で規則的な転送パターンである RSA およびアンチエイリアスでは、転送を行う各エントリに 20%, 50%, 70% の確率でランダムに転送先を変更したケースの評価を取る。このように規則的な転送から一部を確率的に変更することでランダムなデータ転送への耐性を評価する。

評価項目として、データ転送処理にかかる転送回数を比較する。また、ランダム評価では、各確率毎に 10 パターンずつエントリの目標状態を変更した転送パターンの平均値を取る。

##### 5.1 規則性のあるデータ転送処理

RSA のデータ転送は、複数のデータを同距離同方向に転送するのみである。アンチエイリアス処理

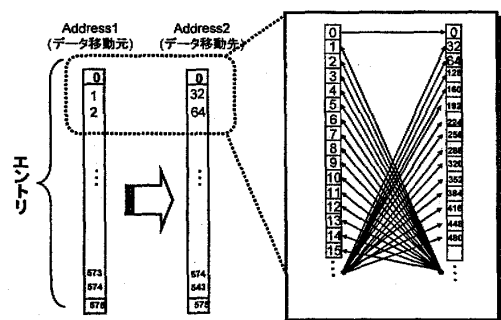


図3 IMDCTのデータ転送

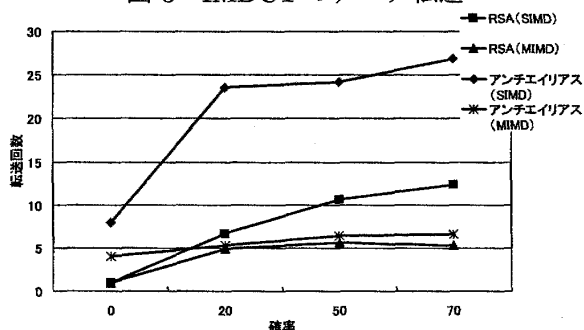


図4 RSAおよびアンチエイリアスのデータ転送は、MP3で使用される1グラニューールである576データに対して576エン트리使用して演算する。データ転送処理は、10から25までの16エントリで逆順の並び替えを行う。また、2エントリ挟んで16エントリの逆順を取る、これを576エントリまで行う処理となる。これら規則的なデータ転送において、ランダム性を付加することでSIMD制御およびMIMD制御における特性を評価する。

## 5.2 複雑なデータ転送処理

規則的だが複雑なデータ転送として、IMDCTのデータ転送を図3に示す。IMDCTにおけるデータ転送処理は、576個のデータを32個おきに上から集める処理である。前述したアンチエイリアス処理の転送は近辺のエントリで完結していたがこの転送は576個のエントリ全域に渡り転送が起こる。

## 6 評価結果および考察

図4に、RSAおよびアンチエイリアス処理中のデータ転送処理における転送回数を示す。図中の縦軸は転送回数、横軸はランダム転送処理の付加確率である。ランダム転送を付加していないものが0であり、以降、動作エントリに対し20%、50%、70%とランダム転送の割合を増やす。

全ての結果において、MIMD制御の方がより少ない回数で転送処理を終了している。これは、SIMD制御に合わない転送の組合せが出たためであり、ランダム転送の確率が多くなるに従いより顕著になる。

表2 IMDCTにおける転送回数とメモリ使用量

	転送回数	制御ビット (bit)
SIMD 制御	91	91
MIMD 制御	9	36

表2に、IMDCT処理中のデータ転送における転送回数と制御ビット数を示す。IMDCTにおけるデータ転送処理では、SIMD制御に対しMIMD制御は約90%の転送回数削減、および60%のメモリ使用量の削減を達成した。

これらの結果から、MIMD制御によるデータ転送処理は複雑なデータ転送が起こっても安定して良い性能が出ていることがわかる。規則性が高い転送になるとSIMD制御においても十分な性能が出ている。制御メモリは一回の転送につきMIMD制御ではSIMD制御の4倍必要とするので、メモリ使用量を考えると両制御を切り替えて利用することで効率的なデータ転送が実現すると考える。

## 7 まとめと今後の課題

本稿では、MXコアのデータ転送効率改善として、定式化とその問題解決手法について議論を行った。結果として、SIMD/MIMD制御それぞれにおいて効果的なデータ転送方法を導出できた。今後の課題として、スケジューリングの計算時間の短縮化およびデータ転送時間と配線リソースのトレードオフの調査を行う必要がある。また、アプリケーション実装におけるデータのメモリ配置問題も含めた評価を行う必要があると考える。

## 参考文献

- 1) 富田眞治, 並列コンピュータ工学, 昭晃堂 (1996).
- 2) K. Mizumoto, et al, "A multi matrix-processor core architecture for real-time image processing Soc," A-SSCC, no.6-2, pp.180-183 (2007).
- 3) 溝上雄太, 中野光臣, 飯田全広, 末吉敏則, "SIMD型プロセッサMXコアにおけるPE間データ通信の高度化," 信学技法 CPSY2007-64, Vol.107, No.415, pp.19-24 (2008).
- 4) 西原 清一, "制約充足問題の基礎と展望," 人工知能学会誌, Vol.12, No.3 pp. 351-358(1997).
- 5) 伊庭 斉志, "探索のアルゴリズムと技法," サイエンス社, (2002).
- 6) 兒玉章宏, 溝上雄太, 中野光臣, 飯田全広, 末吉敏則, "MXコアのMIMD型PE間データ通信における経路決定手法の提案," 信学技報 RECONF2008-6, vol.108, no.48, pp.31-36, (2008).
- 7) 浦田敏道: 詳細MP3マニュアル, エム研 (1999).