社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－SLDM－133（17）
2008／1／17

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

溝上 雄太† 中野 光臣† 飯田 全広†† 末吉 敏則††

†††熊本大学大学院 自然科学研究科 〒860-8555 熊本市黒髪2-39-1
E-mail: †{mizokami,nakano}@arch.cs.kumamoto-u.ac.jp, ††{iida,sueyoshi}@cs.kumamoto-u.ac.jp

あらまし 近年のデジタル機器の高機能化および多機能化に伴い，高性能と低消費電力を併せ持つ組込みシステムが求められている．我々はこれらの要求に対応するため，株式会社ルネサステクノロジ社が開発したMXコアに注目し研究を行っている．MXコアは，ビットシリアル演算器(PE：Processing Element)を複数搭載した超並列SIMD(Single Instruction Multiple Data)型プロセッサである．MXコアのPE間データ通信は，全てのPEが同距離，同方向でデータ移動を行うSIMD型制御である．そのため複雑なデータ移動が必要な際，非効率的な移動を繰り返すことになる．本稿では，各PEにおいて任意の距離と方向でデータ通信を実現する手法を提案する．PE間データ通信部を改良したMXコアにアプリケーションを実装し評価を行った結果，データ移動処理において従来のSIMD型データ通信と比べて5倍から10倍の性能向上が得られた．

キーワード MXコア，SIMD，プログラマブルデバイス，データ通信

Improvement in data communication between PEs for SIMD type processor MX CORE

Yuta MIZOKAMI†, Mitsutaka NAKANO†, Masahiro IIDA††, and Toshinori SUEYOSHI††

†††Department of Mathematics and Computer Science, Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Kumamoto-shi, 860-8555 Japan
E-mail: †{mizokami,nakano}@arch.cs.kumamoto-u.ac.jp, ††{iida,sueyoshi}@cs.kumamoto-u.ac.jp

Abstract We are researching about MX Core developed in Renesas Technology Corp.. MX Core is SIMD (Single Instruction Multiple Data) type architecture and has 1,024 fine grain degrees PEs. The data communication between PEs is regularly, so all data only move same direction and distance. This is a factor of the lower performance. In this paper, we proposed RECM (Reconfigurable Entry CoMmunicater) that gave flexibility to the data communication between PEs. Then, we applied RECM to MP3 decoder and evaluated it. The following results were obtained 5-10 times higher performace.

Key words MX core, SIMD, programmable device, data communication

1. はじめに

デジタルビデオやデジタルカメラなどのデジタル家電製品は，動画処理や画像処理，音声処理といったマルチメディアデータの取り扱いが必要である．従来，これらのアプリケーションや演算処理は，ASIC (Application Specific Integrated Circuit) やDSP (Digital Signal Processor) で対応していた．しかし，最近では一定の時間内に処理すべきデータの総量が著しく増大しており，DSPでは対応が難しくなっている．また，マルチメディアアプリケーションの進化は著しく，多種多様なマルチメディアデータ処理への対応が必要となる．これらをASICで対応すると，新しい規格が登場する度に新しいハードウェアを開発しなければならず開発コストが高くなるという問題がある．

このような背景から，株式会社ルネサステクノロジは新しい処理機構を持つハードウェアアクセラレータとして「マトリクス構造超並列プロセッサコア（以下MXコア）」を開発中である[1]．MXコアは，ビットシリアル演算器（以下PE）を複数搭載しており，この複数のPEを用いて1つの処理命令（single instruction）で並列にデータ（multiple data）を処理するSIMD (Single Instruction Multiple Data) プロセッサアーキテクチャである．MXコアは，主にマルチメディア分野を対象として開発されており，全体として1,024並列で処理が可能である．加えて，メモリ技術をベースにしていることにより，DSPや粗粒度MIMD型プロセッサと比べ小型かつ低消費

電力を実現している．

本稿では，MX コア内の PE 間のデータ通信に注目し，新しいデータ通信アーキテクチャの提案を行う．そして，提案手法を MP3 デコーダに適用し，その有効性の評価を行う．

以下，第 2 章では MX コアの特徴や構造，演算方法を紹介し，第 3 章では今回提案するデータ通信アーキテクチャについて述べる．第 4 章では，評価アプリケーションである MP3 デコーダの処理概要を述べる．第 5 章で評価環境と評価結果を示し，第 6 章でまとめと今後の課題について述べる．

2. MX コア

本章では，MX コアの構造および MX コアのシステムの構成について説明する．

2.1 MX コアの構成

MX コアはホスト CPU や DMA（Direct Memory Access）コントローラ，メインメモリである SDRAM（Synchronous DRAM）と 1 チップ上に実装され，様々なアプリケーションにおいてデータ処理アクセラレータとして機能する．システム全体で見ると，全体制御用のホストプロセッサが存在し，メディアデータ処理用に MX コアが存在するという構成になる．

MX コアの構造を図 1 に示す．MX コアは，ホスト CPU と通信するための I/O インタフェース，MX コアで行う処理を制御するコントローラ，及び SIMD 型演算モジュールから構成される．SIMD 型演算モジュールは，数多くの微小なプロセッサ（Processing Element：PE）とデータレジスタ（SRAM）によって構成される．PE の処理粒度は 2 ビットであり，データレジスタは PE の両翼に 512 ビットずつ配置される．1 つの PE と両翼の 512 ビットのデータレジスタの組を基本構成単位としており，これをエントリと呼ぶ．MX コアはこのエントリを一次元アレイ状に 1,024 個並べ，SIMD 型並列演算を行うことで高速に処理を行う．PE と両翼のデータレジスタ間は H チャネル (H-ch) で配線されており，水平方向でデータ通信を行う際は，それぞれのデータが相互に干渉することなく 1 サイクルで転送することができる．また PE 間（エントリ間）で垂直方向のデータ通信を行う際は，V チャネル (V-ch) を使用することで，一定の距離にある PE 間で一律にデータ通信を行うことができる．

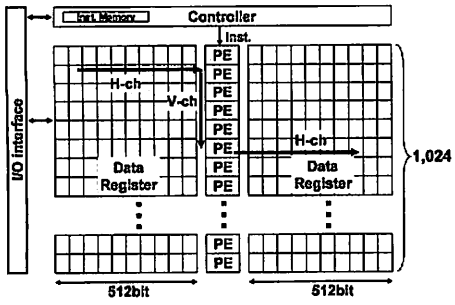

図 1　MX コアの構成

2.2 MX コアの演算方法の特徴

MX コアは，1 つの命令で全ての演算器が同時に同じ命令を実行する SIMD 型の構造であり，最大で 1,024 組のデータを並列に処理する．この超並列処理により，各 PE の演算粒度は 2 ビットと小さいが，全体として高速な演算を可能にしている．

汎用プロセッサと MX コアの演算方法の違いを図 2 に示す．汎用プロセッサの演算はメモリからのロード，演算，メモリへのストアの一連の動作を逐次的にデータの数だけ繰り返す必要がある．例えば，32 ビット汎用プロセッサでビット長 32 ビットのデータを 1,024 個処理する場合は，1,024 サイクルで処理が完了する．一方，MX コアはデータレジスタアレイに格納された 1,024 個のデータの最下位 2 ビットを同時に読出し，1,024 個の演算器（PE）で並列に演算を行い，メモリに書込む．これを最上位ビットまで繰り返すと，1,024 個の 32 ビットデータを処理したことになる．汎用プロセッサが，1,024 サイクル要したのに対し，MX コアはわずか 16 サイクルで処理できる．このように従来の汎用プロセッサの演算時間はデータの個数に比例して増大するのに対し，MX コアの演算時間はデータ長に比例する．

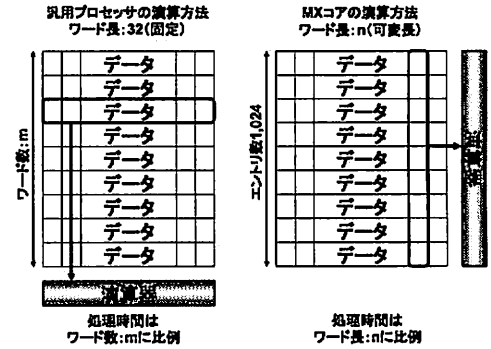

図 2　演算方法の特徴

2.3 MX コアの命令セット

MX コアは 1 ビットおよび 2 ビットの各種演算機能を備えている．それらの命令セットを以下に示す．

- ロード/ストア命令
- フラグ操作命令
- 算術論理演算命令
- PE 間データ通信命令
- レジスタ-レジスタ間の演算命令/移動命令
- レジスタ-メモリ間のロード/ストア命令

これらの命令の組み合わせにより，任意の処理を実現する．

2.4 PE の構成

図 3 に MX コアのエントリ（PE とデータレジスタ）の詳細を示す．A はデータレジスタを表す．PE は，1 ビットフラグ用レジスタ（S，D，F，C，V，N），1 ビットアキュムレータ（X，XH）および 1 ビット演算器（+）から成る．データレジスタ A は X，XH および演算器（+）と接続されている．X，XH は互いに独立しているので，1 ビットまたは 2 ビット両方の演算が可能となる．また，X，XH は PE 間通信回路を介し

て垂直方向のデータ通信用配線（V チャネル）とも接続されている．図 3 の場合，左翼のデータレジスタ A から X，XH へデータを読み出し，演算器（+）にて演算を行い，演算結果を右翼のデータレジスタ A へ書き込む構成となっている．フラグ用レジスタは直接操作することが可能であり，操作する際は一度 X レジスタを通して各レジスタへ値がセットされる．各 PE に設けられた Valid フラグ（V）を用いることで，H チャネル，V チャネルのデータ転送，あるいは PE の演算そのものを実行するか否かを選択することが可能となる．

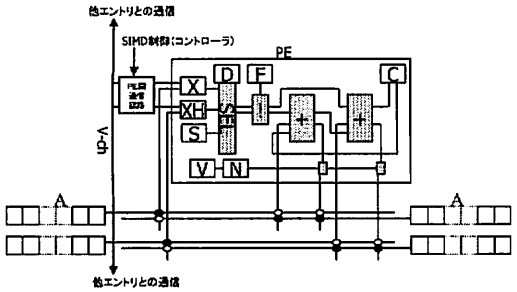

図 3　PE の構成

2.5 MX コアの PE 間データ通信

図 1 に示すように，MX コアは PE ーデータレジスタ間は H チャネル（H-ch）により，PE 相互間は V チャネル（V-ch）によりそれぞれ配線されている．図 4 に PE 相互間の配線状態を示す．現在の V チャネルは 4 のべき乗（± 1，4，16，64，256）の配線長となっている．

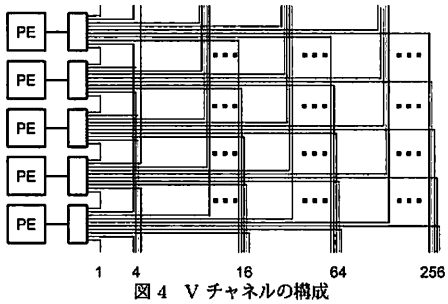

図 4　V チャネルの構成

PE 間のデータ通信を行う場合の処理手順は以下のようになる．

（1） データレジスタから PE（X，XH）へ H チャネルを介してデータを読出す（水平方向のデータ移動）

（2） コントローラからの移動制御（距離と方向）に従い，PE 間通信回路および V チャネルを介してデータを移動先の PE（X,XH）へ転送する（垂直方向のデータ移動）

（3） PE（X，XH）内のデータを H チャネルを介してメモリレジスタへ書込む（水平方向のデータ移動）

3. PE 間データ通信の高度化

本章では従来の PE 間通信の問題点を述べた後，提案する PE 間データ通信の説明を行う．

3.1 MX コアにおける PE 間データ通信の問題点

MX コアでは，全 PE が同距離，同方向の PE に対してデータ通信が行われる．これは全データの移動（距離と方向）の切替えを 1 つのコントローラで一律に制御しているためである．そのため，エントリ毎に移動させる距離が異なる場合は，距離の種類数だけ移動を繰り返す必要がある．このような逐次的な移動の繰り返しは，並列性の低下を招き，性能の低下の要因となる．

SIMD 制御によるデータ通信の例として，図 5 にデインタリーブ処理を実行した際のデータ移動の様子を示す．四角の中の数値はエントリ番号を示す．デインタリーブ処理とは，垂直方向に並んだデータ列に対し，偶数番目のデータをエントリ上部へ，奇数番目のデータをエントリ下部へ移動させる処理である．図 5 に示すように，偶数エントリのデータ移動と奇数エントリのデータ移動を別々に行い，最後に 1 列にまとめる。この場合の移動回数は 7 回となる．

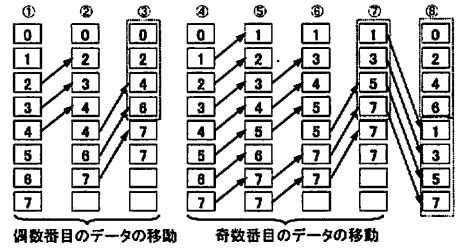

図 5　SIMD 制御によるデインターリーブ処理

3.2 提案手法

3.2.1 Reconfigurable Entry Communicator

本節では，PE 間データ通信の柔軟性を向上させることにより上記の問題点を解決する手法を示す．ここで言う柔軟性とは，それぞれの PE において異なる移動距離および移動方向の指定を実現させることである．この柔軟性を持たせた PE 間データ通信アーキテクチャを Reconfigurable Entry CoMmunicator (RECM) と呼ぶことにする．

2.5 節で述べたように，MX コアは V チャネルを用いることで ± 1，2，16，64，256 という 10 種の距離の垂直データ通信を行うことができる．これに移動しない（移動距離 0）場合を加えた 11 種の移動パタンを表現するためには 4 ビットのレジスタ（これを移動制御用レジスタと呼ぶ）が必要である．このレジスタを各 PE 内に設け，データ通信を行う際に（11 種の移動距離のうち）任意の移動距離及び方向を指定することで各 PE は任意の移動を実現する．

PE 間データ通信を行う際，算術演算や論理演算は実行されない．これより各 PE 内に設けられた 1 ビットフラグレジスタ S，D，F，C（図 3）は使用されない．このことに着目して，これら 4 つのレジスタを移動制御用レジスタとして利用する．表 1 に移動制御用レジスタに格納される移動制御用ビット（4 ビット）と通信距離および通信方向を一覧にして示す．ここで，UP とはエントリ番号が増加する方向への移動，DOWN とはエントリ番号が減少する方向への移動を意味する．

表 1　移動距離および方向と移動制御用ビットの対応

移動距離	移動方向	移動制御用レジスタ			
		S	F	D	C
0	-	0	0	0	0
1	UP (+)	0	0	0	1
4	UP (+)	0	0	1	0
16	UP (+)	0	0	1	1
64	UP (+)	0	1	0	0
256	UP (+)	0	1	0	1
256	DOWN (−)	1	0	1	0
64	DOWN (−)	1	0	1	1
16	DOWN (−)	1	1	0	0
4	DOWN (−)	1	1	0	1
1	DOWN (−)	1	1	1	0

移動制御用レジスタは，アキュムレータ X，XH を介して 1 ビットずつデータが格納される．4 ビット全てを格納するには 2 サイクル必要となる．RECM を用いたデータ通信の処理の流れを以下に示す．

（1） データレジスタから 2 ビット分の移動制御用ビットを PE（X，XH）へ読出す

（2） X，XH の値を移動制御用レジスタへ書込む

（3） 1，2 の動作をあと 1 回（計 2 回）繰り返し，PE 内の移動制御用ビットを設定する

（4） データレジスタから PE（X，XH）へ H チャネルを介してデータを読み出す

（5） 移動制御用ビットの制御（距離と方向）に従い，PE 間通信回路および V チャネルを介してデータを移動先の PE（X，XH）へ転送する

（6） PE（X，XH）内のデータを H チャネルを介してメモリレジスタへ書込む

この RECM を用いたデータ通信の例として，図 6 にデインタリーブ処理を実行した際のデータ移動の様子を示す．各エントリ（PE）のデータは，任意の方向および距離を選択可能なため，3.1 節と比べ，少ない移動回数でデインタリーブ処理を完了していることが分かる．

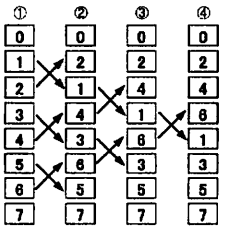

図 6　RECM を用いたデインターリーブ処理

3.2.2　RECM の構成

本節では，RECM 制御を実現するために PE 対して行ったアーキテクチャの変更点について述べる．

RECM 制御を用いたデータ通信を実現するための PE 構成を図 7 に示す．前節で述べたように，S，D，F，C レジスタを移動制御用ビットを格納するレジスタとして利用する．従来の SIMD 型アーキテクチでの演算時と，今回提案する RECM アーキテクチャでの演算時とで移動制御の経路を切替えるためにマルチプレクサ（MUX）を設ける．この MUX は，SIMD 制御によるデータ通信と RECM 制御によるデータ通信のどちらを行うかを指示するモード選択信号を制御信号とする．この制御信号に従い，コントローラからの制御信号あるいは S，D，F，C レジスタに格納された移動制御用ビットの一方を選択する．モード選択信号は，移動命令実行時にコントローラにより生成される．このように移動制御用ビットを格納するレジスタとして，移動命令実行時に使用されていない既存のレジスタを利用することで，追加リソースを極力抑えて RECM アーキテクチャを実現できる．

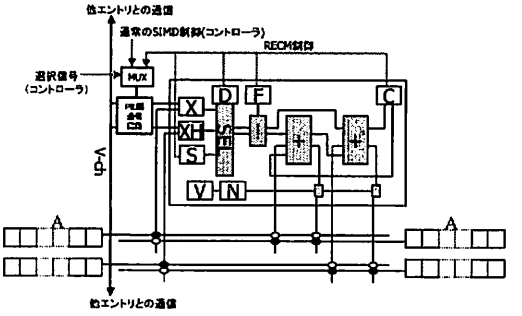

図 7　RECM 制御のための PE 構成

4.　評価アプリケーションと MX コアへの実装

本章では，評価アプリケーションとして用いた MP3 デコーダについて概説し，MX コアへ実装する際の方法について述べる [4]．

4.1　MP3 デコーダ

4.1.1　処理概要

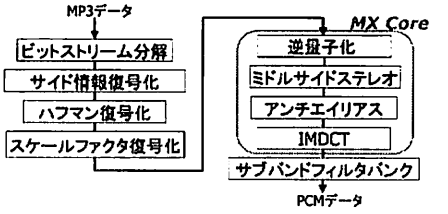

図 8　MP3 デコーダの処理の流れ

図 8 に MP3 デコーダの処理の流れを示す．ビットストリーム分解は，ファイルから MP3 データを読出すにあたり，各フレームデータを抽出する処理である．ハフマン復号化は，フレームデータ中の各チャネルデータに対して復号処理を行うものであり，量子化データへと変換される．スケールファクタ復号化は，サイド情報を基に復号化される．逆量子化は，各チャネルのデータに対して $\frac{4}{3}$ 乗演算を行う．この逆量子化の後のデータは音の周波数系列のデータ，すなわちスペクトルを表している．ステレオ処理は，二つのチャネル間においての加算と減算を行う．アンチエイリアスは，バタフライ演算による処理である．これにより，エンコード処理におけるハイブリッ

ドフィルタバンク処理でのエイリアスを削除することができる．IMDCT (Inverse Modified Discrete Cosine Transform) は，DCT (Discrete Cosine Transform) および窓関数処理である．IMDCT 処理の出力は 32 × 18 の周波数領域．時間領域のデータとなる．最後の逆サブバンドフィルタバンクは，IMDCT にて出力されたサブバンドデータを合成し PCM (Pulse Code Modulation) データの出力を行う．

4.1.2 MP3 データ構造

MP3 の圧縮対象となる音声データは，PCM 信号で構成され，モノラルであれば 1 つ，ステレオであれば左右 2 つの PCM 信号からなる．MP3 では，576 サンプリングデータを 1 グラニュール再生データと呼ぶ．また，再生可能な最小データ単位をフレームと呼び，2 グラニュール，すなわち 1,152 サンプリングデータを 1 フレームと定義している．1 フレームのデータ構造を図 9 に示す．グラニュール 0 はこのフレームに含まれる 2 つのグラニュールのうち時刻の早いほうを指す．

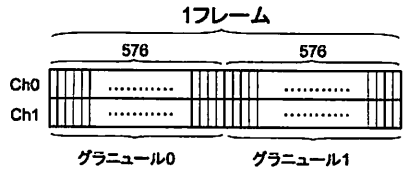

図 9 MP3 データ構造

4.2 MP3 デコーダの実装

本稿における MX コアへの MP3 デコーダの実装方法を以下に示す．なお，実装するにあたり MP3 デコーダのソースは Xing が提供しているフリーのデコードエンジン [2] を参考にし，同様の演算を行っている．

また，MP3 デコーダにおいてハフマン復号化，スケールファクタ復号化はテーブル参照を行う処理である．テーブルの値を全て MX コアのエントリに格納することは，MX コアのメモリの制限から難しい．そのため逆量子化以降の処理を MX コアに行わせる．

4.2.1 データ配置

第 3 章で説明したように，MP3 データは 1 グラニュールである 576 データを基本の単位としている．その 576 データを MX コアでの 576 エントリに配置し，最大で 576 個のデータを 1 度の演算で行う．また，演算に必要である各係数データや各エントリの演算のマスキングに必要なマスクビットもそれぞれのエントリに配置する．

4.2.2 演 算 方 法

- 逆量子化

逆量子化にて用いられる係数データは，前処理結果のスケールファクタを基に計算される．よって，この係数データについては毎回ホスト CPU にて計算を行った後に MX コア内に格納することにする．逆量子化ではこの係数データに対して $\frac{4}{3}$ 乗演算を行う．$\frac{4}{3}$ 乗演算に関しては浮動小数点演算の近似演算を用いて実現する [3]．

- ステレオ処理

本研究で対応するステレオ処理は，二つのチャネル間においての加算と減算処理を行う．これらは同一 PE に各チャネルデータを配置しているので浮動小数点演算の加減算のみで実現する．

- アンチエイリアス

バタフライ演算による処理であり，PE 間のデータ通信を利用して実現する．PE 間のデータ通信は規則的である．

- IMDCT

アンチエイリアス処理を行った後の MP3 データに対して，演算処理と窓処理を行う．次に，このデータを次のグラニュールでの演算の為に別の領域にコピーし，そのデータに対して窓処理を行う．窓処理を行ったあとにデータに対して，周波数領域から時間領域のデータ構造変換のためにエントリ間のデータの並べ替えを行う．IMDCT でのデータの並べ替えはアンチエイリアスと異なり，不規則な移動が発生する．エントリ間のデータ移動は SIMD という特徴から並列性は損なわれることが多く，処理時間の多くを占めてしまうのでできるかぎり最適化を行わなければならない．最後に，データに対して窓処理をもう一度行い，前のグラニュールの処理結果とオーバーラップ加算することで処理は終了する．

4.2.3 RECM の適用

前節より，MX コアに実装したデコード処理のうち，アンチエイリアス処理および IMDCT 処理においてエントリ間のデータ移動が発生することが分かる．特に，IMDCT 処理では不規則な移動が存在し，性能低下の原因になっている．そこで，このアンチエイリアス処理と IMDCT 処理の移動部に対して RECM を適用する．

データ移動は，2 つの一時格納領域を利用して異翼間で RECM を用いたコピー命令を繰り返し行うことにより実現する．異翼間で行う理由は 2 章で述べたとおり高速化が目的である．

5. 評　　価

本章では第 4 章で述べたアンチエイリアス処理および IMDCT で発生する PE 間データ通信処理に対して RECM を適用した際の性能評価および考察を行う．

5.1 評 価 環 境

本稿でデコード対象とした MP3 データは，2.1 秒 86 フレームのモノラル音声である．MP3 エンコード時に高速フーリエ変換を行う際のブロック長はロングブロックのみとする．また，MP3 デコード時の演算は，IEEE754 形式の 32 ビット単精度実数を用いて行う．評価は，ルネサステクノロジ社提供の MX シミュレータ（Version 0.05.01）を用いる．このシミュレータは，図 1 における入出力部および制御部の所要時間は考慮しない．以上の条件で，MP3 データの 1 グラニュール（576 サンプリングデータ）に対してデコード処理を行った際の評価を行う．

初めに，今回提案した RECM の効果を見るために，RECM をアンチエイリアス処理および IMDCT 処理に適用した際の移動回数，使用メモリおよび所要時間について評価を取る．移動回数および移動時間とは，各処理における移動処理を完了するために要した移動の回数および所要時間である．使用メモリとは，データ通信時に使用する SIMD 型移動制御におけるマスク

ビットおよびRECM移動制御における移動制御用ビットの1エントリあたりのメモリ使用量である．

続いて，MXコアで実装したMP3デコード処理，即ち逆量子化，ステレオ処理，アンチエイリアスおよびIMDCTの処理に必要な時間（サイクル数）の評価を行う．評価パラメータとして，各処理における移動の所要時間，浮動小数点演算の所要時間などをあげる．

5.2 評価結果

5.2.1 データ通信部の評価

MP3デコード処理のうち複雑なデータ移動が必要となるアンチエイリアス処理およびIMDCT処理のデータ通信処理部に対して今回提案したRECMを適用した．表2および表3に評価結果を示す．

表2 アンチエイリアスにおける提案手法の評価

	SIMD制御	RECM制御	効果
移動回数 (回)	11	4	63% 削減
使用メモリ (bit/エントリ)	11	16	50% 増加
所要時間 (cycle)	363	72	80% 削減

表3 IMDCTにおける提案手法の評価

	SIMD制御	RECM制御	効果
移動回数 (回)	91	11	87% 削減
使用メモリ (bit/エントリ)	91	44	51% 削減
所要時間 (cycle)	2,208	220	90% 削減

表2より，アンチエイリアス処理における移動回数は，SIMD制御の移動回数の63% 削減，データ移動の所要時間は約80% 削減されている．但し，使用メモリに関しては，50% 増加している．同様に，表3より，移動回数は約87% 削減，使用メモリは約51% 削減，データ移動の所要時間は約90% 削減されている．従来のSIMD制御と比べ，アンチエイリアスはおよそ5倍の性能向上，IMDCTはおよそ10倍の速度向上が得られた．これより，提案手法がデータ移動に対して効果的であることを示している．また，複雑な移動ほどRECM制御による移動の効果が大きいことが分かる．

5.2.2 MP3デコーダ全体の評価

図10にMP3デコーダにおける各処理の所要時間（サイクル数）を示す．但し，アンチエイリアス処理およびIMDCT処理の所要時間はRECM制御ではなく従来のSIMD制御を用いたときの値である．

MP3デコード処理全体において，IMDCTの処理時間の占める割合は大きいが，IMDCT処理のほとんどが浮動小数点演算であり，移動命令（コピー命令）は演算処理に次いで2番目の処理時間を占める．図10の移動（コピー命令）の所要時間と表3の所要時間が異なるが，これはRECMの適用対象とした移動処理は，あくまでIMDCTにおける移動処理の一部だからである．

今回のMP3デコーダは，32ビット浮動小数点数で処理を行ったが，デコードにそれほどの精度は必要なく，24ビットの固定小数点数程度の精度で十分であることが実証されている．従って，MP3デコーダを24ビット固定小数点数で実装することにより，浮動小数点演算の割合は大きく削減すると予想される．一方，データ移動の割合は不変であるため，演算精度を24ビット固定小数点数に落とすことにより，提案手法導入の効果はさらに大きくなり，デコード処理全体の性能改善に大きく寄与することが期待できる．

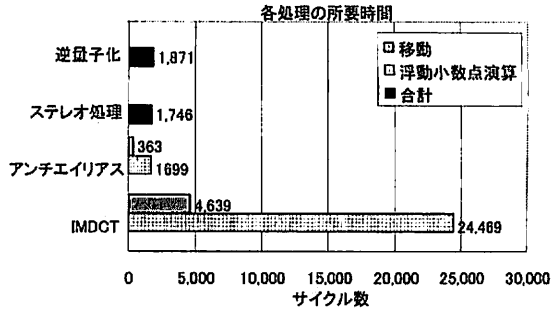

図10 各処理のサイクル数の比較

6. まとめと今後の課題

本稿では，新しい処理機構を持つデバイスとして，ルネサステクノロジ社のMXコアに注目した．MXコアにおけるPE間データ通信はSIMD制御であるため，全エントリのデータを同方向かつ同距離のエントリへ移動させるものである．そのため複雑なデータ移動を行う場合，データの移動距離の種類数だけ移動を繰り返す必要があり，性能の低下の要因となっている．そこで本稿では，既存リソースの増加を極力抑えつつ，各PEにおいて異なる移動距離および移動方向を指定可能なアーキテクチャRECM(Reconfigurable Entry CoMmunicator)を提案した．これにより，データ通信の制御を各PE（エントリ）レベルで指定可能とし，複雑なデータ移動にも柔軟に対応可能となる．

提案手法を評価として，MXコアにMP3デコーダを実装し，アンチエイリアス処理およびIMDCT処理内で発生するデータ移動処理部に対して提案手法を適用した．その結果，データ移動に関して5-10倍の速度向上が得られた．このことより，提案手法が複雑なデータ移動に対して有効であると言える．

今後の課題として，IMDCT処理後の逆サブバンドフィルタバンク処理への適用，またMP3デコーダ以外のアプリケーションに対しても提案手法を適用し，評価を行う必要がある．さらに，RECMにおけるデータ通信を行うための移動制御用ビットを自動生成するツールの作成も必要と考える．

文献

[1] M.Nakajima et al., *"A 40GOPS 250mW Massively Parallel Processor Based on Matrix Architecture"*, ISSCC Dig. Tech. Papers, pp. 410-411, Feb., 2006.

[2] "Xing Technology Corp." http://www.xingtech.com

[3] 山崎博之，飯田全広，水本勝也，山本治，末吉敏則：*"単純なSIMD演算の組み合せによる高速実数演算の実現"*. 信学技法 RECONF2005-23(2005-5), Vol.105, No.43, pp.49-54, May. 2005.

[4] 卜部公介，馬木渉，大野隆行，飯田全広，末吉敏則：*"MXコアにおけるメディアアプリケーションの実装と評価"*. 信学技法 CPSY2007-14(2007-8),, vol.107, no.175, pp.49-54, Aug. 2007.