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

2007－ARC－172 (45)
2007－HPC－109 (45)
2007／3／2

オンチップメモリプロセッサでの演算加速機構の検討

高橋睦史[†1] 佐藤三久[†1] 高橋大介[†1]
朴泰祐[†1] 宇川彰[†1] 中村宏[†2,†1]
青木秀貴[†3] 澤本英雄[†4] 助川直伸[†3]

電力性能の向上に有効であるオンチップメモリプロセッサアーキテクチャSCIMA に，演算あたりのハードウェアおよび電力コストに有利な演算加速機構を導入することとし，その演算加速機構の構成について検討した．ベクトル型および SIMD 型の演算加速機構を提案し，科学技術計算で頻出するベクトル処理を用いて予備評価を行った．ベクトル型に適したこのような処理において，SIMD 型ではプロセッサコアでのアドレス計算がボトルネックになることが分かり，演算加速機構にアドレス演算機構をいれるなど，いくつかの改善方法を提案した．電力などのいくつかの観点から検討を行ったが，さらに検討・評価を行う必要がある．

A study on arithmetic accelerators for on-chip memory processor

Chikafumi TAKAHASHI,[†1] Mitsuhisa SATO,[†1]
Daisuke TAKAHASHI,[†1] Taisuke BOKU,[†1] Akira UKAWA,[†1]
Hiroshi NAKAMURA,[†2,†1] Hidetaka AOKI,[†3]
Hideo SAWAMOTO[†4] and Naonobu SUKEGAWA[†3]

In this paper, we study a design of arithmetic accelerators to integrate the accelerator into on-chip memory processor, which is expected to be effective to improve power performance. We propose vector-type arithmetic accelerators and SIMD-type arithmetic accelerators. The results of preliminary evaluation using simple vector loops, which is suited to vector-type accelerators, indicates that address calculation by scalar processor is a bottleneck in SIMD-type accelerators. We present how to improve the performance of SIMD-type accelerator for such problems. We have also examined many aspects including the power performance for both accelerators, which still needs further investigation by simulation.

1. はじめに

マイクロプロセッサはデバイス技術やアーキテクチャの改良により，その動作周波数を向上させてきた．その結果，現在では 3GHz を上回る動作周波数のプロセッサが現れるに至り，その高い動作周波数により高い性能を達成してきた．しかし代償としてプロセッサの消費電力は高くなり，集積度や実装に関して問題があることが明らかになってきた．最近では，比較的周波数を抑えたコアを 1 チップ内に集積したマルチコアプロセッサが使われるようになり，高性能システムにおいても低電力プロセッサを集積した BlueGene/L[1] が注目されている．

現在，国内では次世代のスーパーコンピュータの開発プロジェクトが進められており，10PFLOPS を越える性能が期待されている．このような大規模超高性能システムを構築するに当たって，最も重要な課題は消費電力当たりの性能を向上させることである．この観点からは従来のベクトルスーパーコンピュータを集積する方法ではその限界は明らかであり，また現在主流となっている汎用プロセッサを用いた PC クラスタの延長でも難しい．

本稿では高性能な演算処理が必要とされる科学技術計算を対象に，オンチップメモリとそれを効率的に活用する演算機構により演算当たりの電力効率を高めるプロセッサアーキテクチャについて検討する．

オンチップメモリプロセッサはプロセッサコアと同一チップ上にメモリを持つプロセッサである．チップ外の主記憶にアクセスするには相対的に大きな電力が必要となるが，オンチップメモリを利用することでオンチップメモリ・主記憶間のデータ転送をソフトウェアで効率的に制御でき，電力性能を改善することができる．オンチップメモリを用いたプロセッサアーキテクチャは数多く提案されている．我々は SCIMA と呼ぶアーキテクチャを提案している[2),3)]．これはキャッシュと同じメモリ階層に，SRAM を用いたオンチップ RAM を実装するアーキテクチャであり，シミュレーションベースで電力性能を含めた有効性を確認している．また，SCIMA に近いメモリ構成にすることのできる SH4 プロセッサを利用し，実機の測定によりオンチップメモリが演算性能と電力性能の向上に有効であることを確認している[4)]．我々はこの SCIMA を基本的なオンチップメモリアーキテクチャとして考える．

†1 筑波大学 計算科学研究センター
Center for Computational Sciences, University of Tsukuba
†2 東京大学 先端科学技術研究センター
Research Center for Advanced Science and Technology, The University of Tokyo
†3 （株）日立製作所 中央研究所
Centeral Research Laboratory, Hitachi Ltd.
†4 （株）日立製作所 エンタープライズサーバ事業部
Enterprise Server Division, Hitachi Ltd.

他方，電力性能改善のアプローチとしてプロセッサあたりの演算器を増やすことが考えられる．プロセッサコアと比較して構成が単純な演算器を増やすことで単位時間当たりの演算量を増やし，消費電力の増加を最小限に抑えつつ性能を向上させることができる．そのためには効率的に演算器を利用するための機構が必要となる．

さらに，次世代のプロセッサで用いられると想定される 45nm プロセスではさらに多くのトランジスタを利用できるため，多数のコアを搭載するマルチコアとし，クロック周波数を押えて比較的低電圧で駆動し並列処理を行うことにより電力効率を高めることができる．

第一段階として，本稿ではオンチップメモリプロセッサに搭載する演算加速機構の構成を検討する．検討した構成についてはシミュレータを用いて予備評価を行った．この予備評価を考察することで検討した構成の問題点を抽出し，改善する方法を検討する．

2. オンチップメモリプロセッサと演算加速機構

本章では我々の前提とするオンチップメモリアーキテクチャSCIMA，考えられるいくつかの演算加速機構について述べる．

2.1 SCIMA:ソフトウエア可制御メモリ

SCIMA はハイパフォーマンスコンピューティング向けのオンチップメモリプロセッサアーキテクチャであり，オンチップメモリをメモリウェイごとに，すべてのデータ転送がソフトウェア制御可能な一時記憶か従来型のキャッシュとして使い分けることが可能である．オンチップメモリを使用することで以下のような利点がある．

(1) 演算に必要なデータを明示的に指定することで，従来のキャッシュで生じていたような意図しないキャッシュミスや固定されたキャッシュラインサイズでのデータ転送で発生する不要なデータ転送を抑制する．

(2) 演算に必要となるより前にデータをオンチップメモリに転送しておくことでデータプリフェッチが明示的に行える．これはキャッシュプリフェッチとほぼ同等の機能である．しかし，キャッシュプリフェッチとは異なり，プリフェッチしたデータが意図せずキャッシュから追い出されることはない．オンチップメモリが主記憶との DMA 転送をサポートする場合はキャッシュプリフェッチと同様にデータ転送時間の隠蔽が可能になる．

(3) (2) によりオンチップメモリにあらかじめデータを転送しておくことで，オンチップメモリへのアクセスに関してアクセスレイテンシが一定であることを保証できる．

(1), (2) の利点により性能が向上し，加えて主記憶へのアクセスを削減することによって電力性能の向上も見込まれる．また, (3) によりアプリケーションプログラムの実装時に，オンチップメモリへのアクセスレイテンシの隠蔽するための最適化が容易になる．以降，このオンチップメモリの利点を有効に活用することに念頭に置き，そのほかの検討を行う．

2.2 演算加速機構

物理的限界により制限された電力やハードウェアコスト下で高い演算性能を得るために，プロセッサに新たな演算器を付け加える方法がある．単純な演算器であれば消費電力やハードウェアコストは小さいことから，プロセッサコアよりも単純な制御機構のみを持つ演算機構を増やすことで，コストパフォーマンスを上げつつ性能を向上することができる．演算加速機構は，実現可能性を考えて既存のプロセッサコアに付加される形で実現するものとする．大きく分けて，以下の方式が考えられる:

- **ベクトル演算機構:** 従来のベクトルプロセッサに用いられている方式．ここではオンチップメモリに対してベクトル処理を行う．ベクトルレジスタに滞りなくデータを供給するために大きなバスバンド幅を用意する必要がある．大きなバスバンド幅を用意することは配線コストなどのハードウェアコストが大きく，また電力コストも大きい．
- **SIMD 演算機構:** 最近のマイクロプロセッサで採り入れられている方式．比較的小量のレジスタに対し，一括ロード／ストア，複数の演算器により同一の演算を行う命令を持つ演算機構．ベクトル演算機構に比べ，比較的細かい単位での演算を行うことができる．
- **その他の演算加速機構:** その他の演算器方式として，独立した命令ストリームで動作するシンプルな演算コアを持つ非対称型マルチプロセッサや，グラフィックアクセラレータや FPGA アクセラレータのように，プロセッサ外に付加される演算加速機構を利用する方法などが挙げられる[5),6)]．

グラフィックプロセッサや FPGA の演算加速機構では特定の処理に関しては低い消費電力および回路コストで高い演算性能が得られる半面，その接続方式によっては大きなメモリバンド幅を得ることは難しく用途が限られる．したがって，これらの演算器方式は本稿では扱わないこととする．

2.3 オンチップメモリと演算加速機構の接続方式・メモリ階層

オンチップメモリと演算加速機構の接続に関しては，以下の２つの方式が考えられる．

(1) オンチップメモリを複数階層とし，複数階層キャッシュと同じように小容量高速なオンチップメモリと大容量低速なオンチップメモリを組み合わせる．

(2) オンチップメモリを単層とし，演算加速機構と直接接続する

オンチップメモリプロセッサはソフトウェアによりオンチップメモリ・主記憶間のデータ転送を明示的に制御するという構造がさまざまな利点を生み出している．したがって，その間には別のメモリ階層を挟むことは，ハードウェアの複雑化に加え，複数階層のオンチップメモリを同時に制御しなくてはならず，プログラミングコストを著しく増大させる．ただし，SIMD 演算機構の場合，直接データをロードすることを仮定した場合，レイテンシを隠蔽するために多くのレジスタが必要となる．従来の SIMD 型プロセッサの場合，通常，キャッシュを用いて，プリフェッチを行う．この場合，プリフェッチを行ってもデータがキャッシュにあることは確実でなく，メモリアクセスのレイテンシが一定とはならない．演算加速機構で，ある程度の数のレジスタを持つならば，レジスタにプリロードするごとに L2 キャッシュレベルの数十サイクルのアクセスレイテンシを隠蔽することは可能である．したがって本稿では (2) の単層のオンチップメモリを実装し，

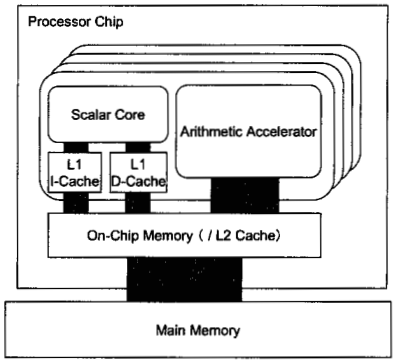


図 1 演算加速機構を持つオンチップメモリプロセッサの基本構成

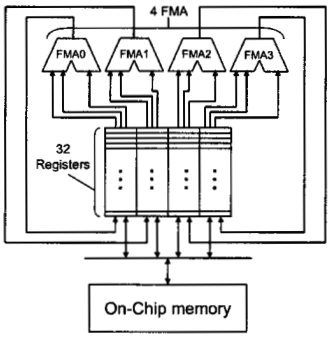


図 2 SIMD 型演算加速機構

オンチップメモリと演算加速機構，およびオンチップメモリと主記憶を直接接続するメモリ階層を取る方法を採用する．

また，通常の命令処理を行うスカラコアについては，演算加速機構が扱うデータ粒度より細かな処理を扱うことを想定すると，演算加速機構を使用しない場合の性能が著しく低下する恐れがある．従って，スカラコアには L1 キャッシュをオンチップメモリに直接接続する形で導入することとする．スカラコアと加速機構がコアを構成し，このコアを複数配置し，L2 およびオンチップメモリを共有する．この構成を**図 1** に示す．

3. オンチップメモリプロセッサの演算加速機構

本節では前節で検討した SIMD 型やベクトル型の演算加速機構についての詳細について述べる．なお，これらの演算加速機構は既存のプロセッサコアに付加される形で実現されるものとする．

3.1 SIMD 型

本稿で検討する SIMD 型演算加速機構を**図 2** に示す．この演算加速機構は 4 個の倍精度浮動小数点数積和演算器（FMA）を持ち，32 個の SIMD レジスタを持つ．1SIMD レジスタあたり 4 要素を持ち，演算時には 1 サイクルで 4 要素がそれぞれ対応する演算器へと送られる．

SIMD 命令としては演算命令は Fused Mutiply-Add を含む基本的な演算をサポートする．データ転送命令ではストライドアクセスをサポートしないが，レジスタ内要素を交換する命令をサポートする．SIMD 命令においても指定できるオペランドはレジスタ番号のみとし，オペランドでレジスタ内要素を指定して演算することはできない．SIMD レジスタへのロード命令のアドレスは，メインのプロセッサコアの汎用レジスタから指定するものとする．

SIMD 型演算加速機構をオンチップメモリプロセッサに導入した場合は，転送したデータは必ずオンチップメモリに保持するのでアクセスレイテンシが一定であることを保証できる．プリロード命令を用いて，演算とオーバーラップして効率的な命令スケジューリングが可能である．また，性能および消費電力はキャッシュを利用する場合よりも高くなることが予想される．

SIMD 型演算加速機構の場合，比較的小量の要素を SIMD レジスタにロードして処理を行う．再利用される値をレジスタに保持しておき，最適化を図るレジスタブロッキングやレジスタ上の交換命令を用いるなど細かい最適化が可能である．ただし，連続したベクトルを処理する場合，ベクトル型演算加速機構と比較して単位演算あたりの命令発行数が多くなるので，その点においてベクトル型よりも不利となる．

電力の観点からは，ベクトル型に比べ制御が簡単なため，消費電力を押えられる可能性がある．加速機構に L1 キャッシュを用いないことやレジスタの再利用によりオンチップメモリと演算器のトラフィックを削減することには電力削減の効果がある．

3.2 ベクトル型

本稿で検討するベクトル型演算加速機構を**図 3** に示す．この演算加速機構では 16 個の倍精度浮動小数点数積和演算器（FMA）を持ち，32 個のベクトルレジスタを持つ．1 ベクトルレジスタは通常のベクトルプロセッサよりも少ない 16 要素 ×8 要素 =128 要素で構成される．1 サイクルに 1 ベクトルレジスタあたり 16 要素ずつ演算器に送り込まれ，これが 8 サイクル繰り返される．チェイニング機構を持つものとし，後続のベクトル演算のオペランドとして用いられる場合には，バイパスして後続の演算に供給される．

Vector 命令としてはデータ転送命令，Fused Mutiply-Add を含む基本的な演算及びベクトルレジスタ内全要素の総和演算，スカラコアの浮動小数点レジスタ内容の全要素へのコピー，先頭要素のスカラコアへの取り出しなどを用意し，最大 3 オペランド入力 1 オペランド出力をサポートする．データ転送命令ではストライドアクセスをサポートするが，本稿では使用していない．1 オペランドで 32 のベクトルレジスタの 1 つを指定できることとし，レジスタ内の要素を指定して取り出すことや演算することはできない．なお，データ転送命令のアドレスはすべてスカラコア内で演算することとし，特別なインデックスレジスタは用意しない．

ベクトル型演算加速機構をオンチップメモリプロセッサに導入した場合は，データの時間的局所性を利用してデータを再利用することで，従来のベクトル型プロセッサと比較して主記憶へのアクセスを削減することができる．また，再利用性を十分に活用できた場合は主記憶へのバスバンド幅がある程度限られた状態でも効率的に演算できることが見込まれる．オンチップメモリから演算器への転送性能が十分にある場合，ほぼ理想的な性能を得る

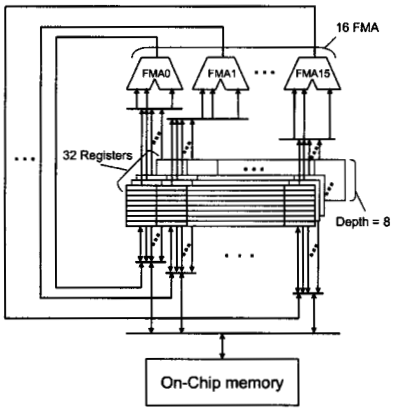


図 3 ベクトル型演算加速機構

ことが見込まれる．

本稿で想定する演算加速機構では L2 キャッシュ階層と同等のメモリ階層に置いたオンチップメモリにアクセスする．したがって従来のベクトル機とは異なりオンチップメモリへのデータ転送を行っておき，ここからロード命令を発行する．また，オンチップメモリ上でのデータの再利用も考えるため，ベクトル長が大きい場合にはオンチップメモリに保持できるデータセットの数が少なくなり再利用性を十分生かせない可能性がある．このため，オンチップメモリを組み合わせる場合にはベクトル長を大きくとることが難しい．この 2 点で本稿の演算加速機構で想定するベクトル型演算加速機構は通常のベクトルプロセッサとは異なる．

3.3 オンチップメモリに対するプログラム最適化

演算加速機構で演算を行う際には，いったんオンチップメモリに主記憶のデータを転送する必要があるが，そのデータ転送はソフトウェアにて制御される．この部分については基本的な仕組みが同じであるので既存研究[2),3)] の手法を襲用することができる．

オンチップメモリプロセッサ向けのプログラム最適化では基本的にキャッシュと同様に時間的局所性を利用する最適化を行う．オンチップメモリを利用したプログラムは

(1) 再利用性のあるデータを選択して主記憶からオンチップメモリにデータを転送
(2) オンチップメモリにあるデータを利用して演算
(3) 書き戻す必要のあるデータを主記憶に書き戻す

といった手順で実行される．このとき，データセットがオンチップメモリよりも大きい場合は何らかのブロッキングを行う．この手順についてはすでに検討したベクトル型および SIMD 型の演算加速機構で共通の動作となる．

なお，オンチップメモリの利用はデータのアクセスパターンが予測可能であることが前提となる．予測困難なアクセスパターンのデータは，プログラミングもしくはコンパイル時にデータ転送の制御を明示することが難しく，基本的にオンチップメモリを用いた最適化は行えない．この場合は予測可能なデータのみをオンチップメモリを用いて最適化を行い，それ以外のデータは従来どおりにキャッシュを利用することで解決する．

4. ベクトル処理に対する予備評価

第一段階の予備評価として，科学技術計算に多く用いられるベクトル処理に対し，SIMD 型演算加速機構とベクトル型演算加速機構のそれぞれについて性能の評価を行った．

SIMD 型演算加速機構とベクトル型演算加速機構の双方に対して，オンチップメモリ部分の動作は本評価で使用するベンチマークプログラムでは共通となるので，本評価では演算に必要なデータはすべてオンチップメモリに転送されているものと仮定して評価を行った．前章で述べた通り，オンチップと演算加速機構の間のデータ転送に対し，十分なメモリバンドが与えられる場合にはベクトル演算機構はほぼ理想的な性能が得られることが想定される．したがって，この評価はベクトル型演算加速機構と SIMD 型演算加速機構の性能を比較し，SIMD 型演算加速機構の問題点，さらにこれからの演算加速機構についての示唆を得ることを目的とする．

4.1 予備評価環境と評価用プログラム

演算加速機構とオンチップメモリを付加するプロセッサコアとして低消費電力プロセッサである SH-4A プロセッサを想定する．The GNU Project Debugger（GDB）5.0 付属の SH シミュレータを拡張し，シミュレータを開発した．クロックシミュレーション機能および演算加速機構を駆動するためのベクトル型および SIMD 型の命令セットを追加した．命令ストリームはすべて SH-4A コアで処理されることとし，ロードストア命令のアドレス演算も SH-4A コアで行う想定とする．

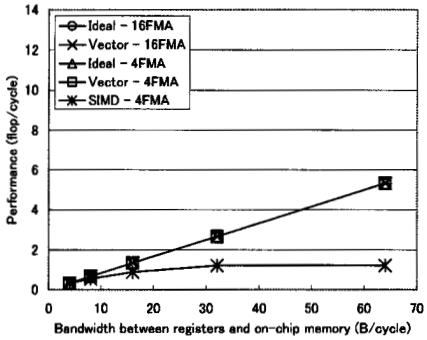
データの依存関係はベクトル命令，SIMD 命令同士のレジスタ依存関係のみをシミュレーションし，それ以外のスカラ命令は依存なく実行できるものと仮定する．また，命令分岐予測は 100%成立するとする．命令発行は SH-4A と同じく Dual Issue のスーパースケーラーとし，演算加速機構の演算命令とロードストア命令は同時に発行できることとする．

現在検討しているベクトル型と SIMD 型について，それぞれの倍精度浮動小数点演算器数を 16 個と 4 個としてシミュレーションを行う（以降，“Vector-16FMA” および “SIMD”）．またベクトル型と SIMD 型の演算器数以外の要素の比較を行うために 4 個の演算器を持つベクトル型のシミュレーションもあわせて行う（以降，“Vector-4FMA”）．その他使用する条件も含めたパラメータを**表 1** に示す．

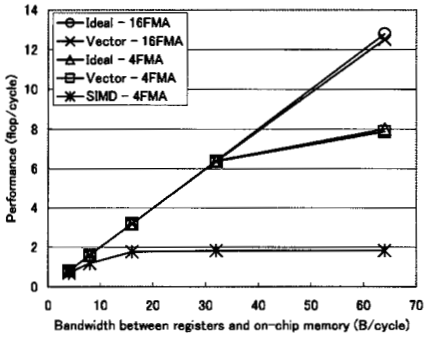
評価用プログラムには DAXPY ループおよび Livermore kernel 1,3,7[7)] を使用した．これらのプログラムは単純な演算のループで構成されており，本評価で各条件

表 1 条件別シミュレーションパラメータ

	Vector 16FMA	Vector 4FMA	SIMD
演算器数	16	4	4
加速機構用 L/S ユニット	1	1	1
レジスタ数	32	32	32
1 レジスタの要素数	128	128	4
オンチップメモリへのアクセスレイテンシ	20	20	20
FMA 命令発行間隔サイクル数	8	32	1
FMA 命令発行後レイテンシサイクル数	4	4	4



DAXPY



Livermore 7

図 4　予備評価結果

での基礎的な演算能力を測定するためのベンチマークである．各プログラムは N=64000 の倍精度浮動小数点数配列を使用する．それぞれのプログラムは C 言語により実装し，演算加速機構を利用する部分についてはベクトルもしくは SIMD 命令セットをインラインアセンブラでバイナリとして挿入した．

4.2 評価結果

評価の結果の内，DAXPY と Livermore Kernel 7 の結果を**図 4** に示す．X 軸がレジスタ=オンチップメモリ間のデータ転送量，Y 軸が性能（flop/cycle）を表す．また，各グラフ中に 16 個および 4 個の演算器があると仮定したとき，ロードストア命令と演算命令の実行クロック数から計算できる理想性能も合わせて示す．

演算器数が同じとなる Vector-4FMA と SIMD を比較した場合において SIMD の性能は不十分であり，最も性能を発揮している Livermore kernel 7 においても 2flop/cycle 以下の性能となっている．理論ピーク性能は 8flop/cycle であるので，理論ピーク性能比は 25%以下である．

SIMD 型の DAXPY プログラムのアセンブラについて調査したところ，1 積和演算命令あたりに 3 つのロード・ストア命令に加えて 6 個以上のアドレス計算のためのスカラ命令が発行されている．これは，アドレス計算をプロセッサコアで行い，ロードするアドレスを汎用レジスタで行っているためである．コアは 2 命令同時実行であるため，依存などによるストールがない場合でも 1 積和演算命令あたりに最低 5 サイクルを要する計算になる．このアドレス演算によるオーバヘッドが著しく大きいために加速機構の演算器を効率的に動作させることができていない．

なお，ベクトル用のプログラムについても，ベクトル型においてもほぼ同様の比率でスカラ命令が発行されているが，ベクトル型では Vector ロード・ストア命令の発行間隔が大きいため，ロード・ストア命令の実行の裏で演算命令とアドレス計算をすべて行えている．

また，最適化に際して，オンチップメモリへの 20cycle のアクセスレイテンシを隠蔽するには，ループアンローリングを行いあらかじめロード命令を発行しておくことが必要であるが，SIMD 型では命令発行間隔が短いためベクトル型と比較して多くの段数をアンローリングする必要がある．その結果 SIMD レジスタが不足するため，Livermore Kerenel 7 を除き隠蔽に十分なアンローリングができなかった．

Vector-4FMA と Vector-16FMA を比較した場合，DAXPY ではすべてのバンド幅において性能がほぼ同じである．加えて，両者ともほぼ理想性能に近い性能を出していることが分かる．DAXPY の場合，2flop の演算に 2 ロード 1 ストアで 24Byte のデータ転送が必要となるが，1 サイクルあたりのデータ転送量は最大で 64Byte であるので演算器が 4 個の想定条件でもデータ転送のためのバンド幅が逼迫していることがわかる．

5. 考察・今後の課題

予備評価では，科学技術計算において最も典型的な処理であるベクトル処理に限り性能評価を行った．ベクトル型演算加速機構にはほぼ理想的な処理であるが，SIMD 型演算加速機構ではベクトル型ではベクトル処理できないプログラムにも適用できる場合もあり，今後，他のプログラムについても評価を進めていく必要がある．ここでは SIMD 型演算加速機構の改善方法を検討する．また，本研究の目的としては限られた電力資源での性能を検討する必要があることから電力性能に関する考察も行う．

5.1 SIMD 型演算加速機構の改善

SIMD 型演算加速機構において性能が得られたかった原因は，アドレス計算を行うコアの性能が不足していたためである．この問題を解決する手段としては以下の方法が考えられる．

(1) スカラコアの処理能力を上げる．スーパースカラの並列度を上げるなどの方法が考えられる．

(2) 演算加速機構にアドレス演算機構を付加する．SIMD レジスタの他にアドレス計算のためのレジスタを導入し，インデックスレジスタによる相対アドレス指定やポストインクリメントなどの演算機構を導入する．

(3) SIMD レジスタにベクトルレジスタのような長さ（深さ）を持たせる．1SIMD 命令あたりの演算量を増やし，命令発行間隔を伸ばすことでアドレス計算命令のサイクル数を隠蔽する．

（1）による方法では，以前で述べたループアンローリング時におけるレジスタが不足する可能性がある．これから，以上の改善点をいれて性能評価を進めていく予定である．

5.2 電力性能面からの検討

ベクトル型と SIMD 型で FMA あたりの消費電力が等しいとして FMA のピーク性能発揮時での電力性能を考

えるならば，FMA が多いベクトル型の演算加速機構が1FMA あたりのオーバヘッドが少なくなる分有利である．しかし実際に比較すべき電力性能はピーク性能ではなく実効性能による値である．したがって電力性能はアプリケーションにより異なり，電力性能を求めるためにはシミュレーションによる評価が必要である．

また，消費電力についても SIMD 型とベクトル型では FMA あたりの消費電力も同じではない．本稿で評価したモデルにおいても，ベクトル型の総レジスタ内要素数は SIMD 型の 32 倍であり，ベクトルレジスタでは Read/Write ポート数に制限を設けて消費電力を抑えることができるとはいえベクトル型は FMA あたりの消費電力が大きい．また，オンチップメモリとレジスタ間の配線長や配線本数，演算加速機構の制御方式によっても FMA あたりの消費電力は異なる．これらをすべて考慮したうえで初めて電力あたりの性能を算出できる．

評価では使用したプログラムがベクトル型に有利であったことも考慮するとベクトル型の電力性能が優位であるとは断定できない．今後の演算加速機構では上記のパラメータに基づき，シミュレーションベースでの電力性能評価が必要であると考えられる．

5.3 マルチコアにおける課題

今回の評価では，1 つのコアと演算加速機構を前提としたものであったが，これからのプロセッサでは複数のコアと演算加速機構が搭載される．図 1 に示したとおり，オンチップメモリおよび L2 キャッシュはこれらにより共有されることになる．オフチップのメモリと異なり，オンチップメモリへのバンド幅についてはある程度の余裕はあるものの，コア数が増えた場合には相対的に制限されることになる．むやみにバンド幅を増やすことは，電力の増加につながる．

また，コアあたりの性能とコア数についても考慮しなくてはならない．コア数を増やすことにより，柔軟な並列処理ができるようになるが，汎用のプロセッサコアは消費電力が高く，コア数を増やすと電力が高くなる傾向がある．本稿で提案した演算加速機構を付加する場合，コア数とコアあたりの演算性能のトレードオフを考える必要がある．

5.4 演算加速機構アーキテクチャの課題

従来のベクトルプロセッサは，バンクメモリを用いた高いバンド幅を前提に実現されてきた．しかし，こういったメモリバンド幅増強手法は物量・電力の両面のコスト的限界により，特に大規模システムでの継続性が限界に近づいている．したがって，適用範囲に制限はあるものの，オンチップメモリの有効利用による電力削減は今後必須になると考えられる．一方，消費電力を抑えつつ CPU の数値演算性能を増強するには，演算器数のさらなる増強は必須で，このためには多数のレジスタを持ち，さらにこれらを効率的にアクセスできる必要がある．

これらの条件により，従来型の long vector アーキテクチャは不適当で，今後は水平方向のレーン数を増やし，相対的に各ベクトルレジスタの長さ（深さ）を短くする形で最適化がなされると考えられる．一方，SIMD アーキテクチャでは，演算器の水平展開度を上げ多数のレジスタを活用し，レジスタを要素毎ではなくある程度のグループ単位でアクセスする必要が生じてくると考えられる．これは，演算パイプラインとレジスタ空間を直交させることによるレジスタポート数の増加を抑制する効果も持つ．

これらの傾向が両者で進むと，最終的には多数のレジスタを時間（深さ）方向・空間（広さ）方向の両方に拡張し，さらにそのようなレジスタ群を多数用いることによって，(i)SIMD 的な演算器の水平展開，(ii) ベクトル的な時間方向連続レジスタアクセスの両特性を融合させた形に集約されていくものと思われる．このようなアーキテクチャの集約された姿を念頭に，ベクトル性と SIMD 性の両特性に対してレジスタ及び演算／メモリアクセスパイプライン構成のパラメータ空間を探索し，アーキテクチャ構成の指標を得ることが今後の課題である．

6. ま と め

本稿ではオンチップメモリプロセッサに多数の演算器を備えた演算加速機構を導入することを検討し，SIMD 型演算加速機構とベクトル型演算加速機構について提案した．これらについて，科学技術計算に多く用いられるベクトル処理について予備評価を行った．ベクトル型演算加速機構に適したこのような処理において，提案した SIMD 型演算加速機構では汎用コアでのアドレス演算がボトルネックになり，十分な性能が得られなかった．SIMD 型の演算加速機構の改善についての方法を示すとともに，電力などの課題について検討を行った．最適な演算加速機構を探るためには，ベクトル処理以外のプログラムでの性能，電力性能についてもさらに検討する必要がある．

今後，考察において検討した改良を行い，ハードウェアコストを試算した上で電力性能を評価するためのシミュレーションを行い，オンチップメモリプロセッサ向け演算加速機構について検討する予定である．

謝辞 本研究の一部は文部科学省「次世代ＩＴ基盤構築のための研究開発」プロジェクトの課題名「低電力高速デバイス・回路技術・理論方式の研究開発」による．

参 考 文 献

1) Adiga, N. et al.: An overview of the Blue Gene/L supercomputer, *Proc. SC2002*, pp. 1–22 (2002).
2) 中村宏ほか: ハイパフォーマンスコンピューティング向けアーキテクチャ *SCIMA*, 情報処理学会論文誌, Vol. 41, No. SIG 5(HPS 1), pp. 15–27 (2000).
3) Kondo, M. et al.: Software-controlled on-chip memory for high-performance and low-power computing, *ACM SIGARCH Computer Architecture News*, Vol. 30, pp. 7–8 (2002).
4) 高橋睦史ほか: オンチップ RAM 利用による電力性能の最適化と評価, 情報処理学会研究報告, 2005-ARC-164, pp. 43–47 (2005).
5) Owens, J. D. et al.: A Survey of General-Purpose Computation on Graphics Hardware, *Eurographics 2005, State of the Art Reports*, pp. 21–51 (2005).
6) Hofstee, H. P.: Power Efficient Processor Architecture and The Cell Processor, *Proc. HPCA'05*, pp. 258–262 (2005).
7) McMahon, F. H.: The Livermore Fortran Kernels: A Computer Test Of The Numerical Performance Range, Technical report, Lawrence Livermore National Laboratory (1986).