2007－ARC－174 (11)
2007／8／1

大規模科学技術計算向けSIMD拡張スカラプロセッサの提案とその評価

山村 周史† 青木 孝† 安藤 寿茂†

我々は，ペタスケールシステム向けのプロセッサアーキテクチャの検討を行っている．ペタスケール規模の科学技術計算アプリケーションを高速に実行するためには，大量の浮動小数点演算を高効率で処理できなければならない．これを実現するために，我々は，既存のスカラプロセッサに対して，SIMD演算ユニットを拡張装備するアーキテクチャを提案する．HPLおよびPHASEの主要計算ルーチンを対象として，シミュレーションにより本アーキテクチャの性能評価を行い，その有効性について述べる．

Proposal and Evaluation of SIMD Extended Scalar Processor for Large-scale Scientific Applications

SHUJI YAMAMURA† TAKASHI AOKI† HISASHIGE ANDO†

A processor for a peta-scale supercomputer requires achieving high floating point performance with high energy efficiency. To meet these requirements, we propose an architecture with the combination of a high performance superscalar processor core and wide SIMD processing elements. In this paper, we evaluate its performance and effectiveness with an architecture simulator using math kernels of HPL and PHASE.

1. はじめに

近年，大規模シミュレーションをはじめとして，HPC分野における計算能力への要求は高まる一方であり，ペタスケール規模の大規模科学技術計算機システムの開発が望まれている．多数のプロセッサを相互に接続してペタスケール規模の大規模計算機システムを構成する場合，そのネットワーク構成やシステム全体の電力，運用管理などの観点から，計算ノードプロセッサ単体は従来以上に高性能かつ電力（面積）あたりで高効率であることが望まれる．

これを実現するために我々は，既存スカラ型のコアプロセッサを基本として，浮動小数点演算ユニットを多数個搭載するSIMD拡張スカラプロセッサを提案する．本研究では，本アーキテクチャの性能を確認するとともに，その特性を分析することを目的として，SIMD化命令列を生成可能なコンパイラおよびプロセッサシミュレータを含めた性能評価システムの開発を行った．これを用いて，大規模HPCシステムにおいて利用が想定される主要計算処理であるDGEMM, ZGEMM, FFTを対象として本プロセッサの性能分析を行い，その有用性について検討を行う．加えて，従来型のプロセッサアーキテクチャとしてSIMD拡張しないCMP構成を採るプロセッサを取り上げ，SIMD拡張プロセッサとの間で，チップ上の回路規模面積あたりの性能比較，検討を行う．

以降，第2章において本論文で提案するSIMD拡張スカラプロセッサのアーキテクチャ構成について詳述する．第3章で，性能評価結果について詳述し，本アーキテクチャの有用性について考察する．最後に，第4章でまとめる．

2. SIMD拡張スカラプロセッサのアーキテクチャ

2.1 設計のねらい

我々は，高い浮動小数点（以降，FPと略記）演算性能を高いエネルギー効率で実現するために，次の2つの効率化アプローチを採った．

1) 多数のFP演算器でのSIMD計算による演算処理の効率化
2) メモリアクセス処理の効率化

まず，ピーク演算処理性能を上げるためにFP演算器数を増加させる．従来のスカラコアを複数個単純に並べるアーキテクチャもあり得るが，制御部のハードウェアコストを抑えるために，命令制御部は単一のままで，多数のFP演算器を接続するSIMD処理方式を採用した．アプリケーション実行において，SIMD処理により並列計算部を高速化すると逐次計算部の処理時間がボトルネックとなる．そこで，逐次計算に対しても高い性能を持つ単一スカラコアを搭載することとした．このような構成により，スカラコアを単純に並べるよりも高い電力効率とピーク性能が実現できると考えた．

次に，メモリアクセスの効率化を図るために，SIMD部に搭載したオンチップメモリ（ローカルメモリ）とス

†富士通株式会社 Fujitsu, Ltd.

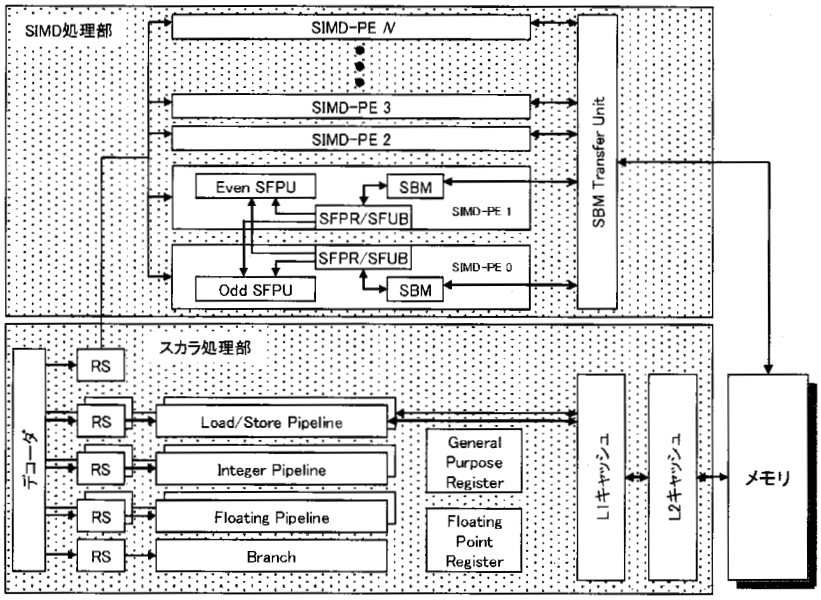


図 1　SIMD 拡張型プロセッサの構成

Figure 1 Structure of SIMD Extended Processor

カラコアのキャッシュを有効利用するというアプローチを採る．SIMD 処理用の大量のデータは，メモリからローカルメモリに直接，高速転送し，行列の転置処理のようなアドレスが不連続なデータはスカラコアでキャッシュに取り込み，SIMD 処理部に転送する．ベクトル型プロセッサなどのストライドアクセスではメモリからのデータの一部しか使用することができないが，キャッシュをバッファとして使うことにより，メインメモリバンド幅を有効に利用することができる．

以上のように，演算およびメモリアクセスの効率化 2 つの観点からアーキテクチャの設計を行った．

2.2　SIMD 拡張スカラプロセッサの構成

図 1 に SIMD 拡張スカラプロセッサの構成を示す．プロセッサコアは，主として「スカラコア部」と「SIMD 演算部」で構成される．スカラコア部は，SPARC64 V[1] をベースとしており，SPARC-V9 アーキテクチャで定義された命令を実行する部分である．SIMD 演算部は，SPARC-V9 アーキテクチャで定義されている FSQRT 命令および FDIV 命令を除く全ての FP 演算命令と後述する新規 SIMD 命令を並列実行する部分である．

SIMD 処理部は，最大 32 個の「SIMD-PE (SIMD Processing Element)」と呼ぶ演算ユニットで構成されている．SIMD-PE は，1 個の FP 演算器（SIMD Floating Point Unit，以下「SFPU」と略記），SIMD FP レジスタ（以下「SFPR」と略記）および後述する SIMD Buffer Memory（以下「SBM」と略記）と呼ぶオンチップの小容量メモリで構成されている．各 SIMD-PE は SBM に格納されているデータを SFPR に転送し，それをオペランドとして演算を行う．本 SIMD アーキテクチャの SIMD-PE は 2 つの SFPU を「Even SFPU」と「Odd SFPU」としてペアで用いて，それぞれの SFPR に格納されている値を相互に参照（クロス参照）して演算することができる．これについては，2.3 節で詳述する．

SIMD 処理部で実行する命令の制御（発行やコミット制御等）は，スカラ処理部によって行われる．SIMD-PE は，スカラ部のデコーダと RS(Reservation Station)に直結されており，プロセッサコアの SIMD 状態ビットをデコーダがチェックする．このビットがセットされた状態で FP 演算命令が現れた場合は，SIMD 命令と解釈して全ての SIMD-PE に同一の命令を発行する．一方，SIMD 状態ビットがリセットされた状態では，FP 演算命令はスカラ処理部の FP 演算器で処理され，SIMD 処理部は動作しない．

SBM は，スカラ処理部の L1 キャッシュに相当するバッファメモリ領域である．キャッシュメモリとは異なり，別アドレス空間のローカルメモリとしてプログラムによってアドレス指定してアクセスできる．ローカルメモリとしたことにより，格納するデータをプログラムで完全にコントロールすることが可能であり，効率的な SIMD 演算を実現するようにデータを配置す

表 1　SIMD 拡張スカラプロセッサの主要新規追加命令

Table 1 New instructions of SIMD Extended Scalar Processor

※ (e)が付いたレジスタは偶数 PE 側，(o)がついたレジスタは奇数 PE 側のレジスタを指す

	命令	ニーモニック	動作
(1)	FXMADDD	fxmddd rs1, rs2, rs3, rd	偶数 PE rd(e) = rs3(e) + rs1(e) * rs2(e) 奇数 PE rd(o) = rs3(o) + rs1(o) * rs2(e)
	FXMSUBD	fxmsubd rs1, rs2, rs3, rd	偶数 PE rd(e) = rs3(e) - rs1(e) * rs2(e) 奇数 PE rd(o) = rs3(o) - rs1(o) * rs2(e)
	FXMSBADD	fxmsbadd rs1, rs2, rs3, rd	偶数 PE rd(e) = rs3(e) - rs1(o) * rs2(o) 奇数 PE rd(o) = rs3(o) + rs1(e) * rs2(o)
	FXMADSBD	fxmadsbd rs1, rs2, rs3, rd	偶数 PE rd(e) = rs3(e) + rs1(o) * rs2(o) 奇数 PE rd(o) = rs3(o) - rs1(e) * rs2(o)
(2)	SBMLD	sbmld [addr], [rd]	メインメモリ上のアドレス addr のデータをレジスタ rd の値をアドレスとする SBM へロードする.
	SBMST	sbmst [rs1], [addr]	レジスタ rs1 の値をアドレスとする SBM のデータをメインメモリ上のアドレス addr へストアする.
(3)	FSBCAST	fsbcast rs1, rd	スカラコアのレジスタ rs1 の値を全 SIMD-PE のレジスタ rd へ転送する.
	SFMOV	sfmov [rs1], rs2, rd	rs1 で指定した SIMD-PE のレジスタ rs2 の値をスカラコアのレジスタ rd へ転送する.

ることが可能となる.

2.3　SIMD 拡張命令とクロス演算

SIMD 拡張プロセッサでは，新規に SIMD 処理部で実行するための命令を追加している．主な新規追加命令を表 1 にまとめる．これらの命令は大きく以下の 3 つのグループに分けられる.

(1)　演算命令

(2)　SBM データ転送命令

(3)　スカラ・SIMD 間データ転送命令

(1)の演算命令では，ペアとなった 2 つの SFPU を用いてそれぞれの SFPR に格納された値を相互に参照して演算を行う「クロス演算」を行うことができる．図 2 に演算例（FXMADDD 命令）を示す．この命令では，奇数 SIMD-PE が他方の rs2（第 2 ソースオペランド）のレジスタ値を入力として演算を行い，自身の SFPR に演算結果を格納する．この機能は，特に，SIMD 処理部において複素数演算を実行する場合に有効利用できる.

SIMD 拡張スカラプロセッサでは，SBM を設けることにより，スカラ部と SIMD 部とで図 3 に示すように異なるメモリ階層を有することになる．スカラ処理部のメモリ階層は，FP レジスタ（FPR），L1/L2 キャッシュ，メインメモリで構成される．一方の SIMD 処理部のメモリ階層は，SFPR，SBM，メインメモリで構成される．本アーキテクチャでは，SBM とメインメモリ間のデータ転送を行うために，(2)の SBM 操作命令である SBMLD(SBM Load) 命令および SBMST(SBM Store) 命令を追加している．通常の Load/Store 命令は，前述の SIMD 状態ビットによりデコーダが発行先をスカラコア側か SIMD 側かに切り替えて実行する.

(3)のスカラ・SIMD 間データ転送命令は，スカラコアにおける FPR のレジスタ値と SIMD-PE における SFPR の値を互いに転送する命令である．キャッシュをバッファとして不連続なメモリアクセスを行う場合や，すべての SIMD-PE にオペランド値を配布する場合などに使用する.

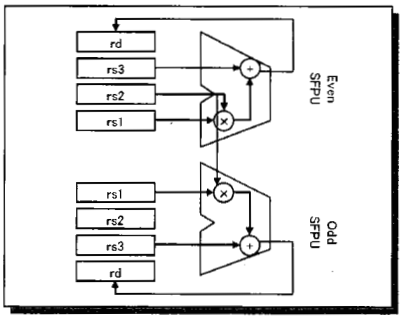


図 2　SIMD-PE ペアの演算

Figure 2 Example of Execution with SIMD-PE Pair

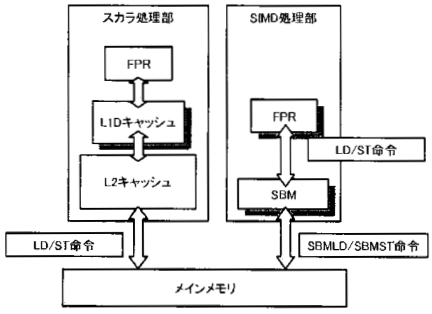


図 3　スカラコアと SIMD-PE のメモリ階層の違い

Figure 3 Memory Hierarchy Difference between Scalar-core and SIMD-PE

2.4 SIMD-PE 実行パイプラインの構成

SIMD-PE は，以下の 3 つのパイプラインを持つ．

1) SIMD 演算パイプライン　　1 本
2) SIMD ロードパイプライン　　2 本
3) SIMD ストアパイプライン　　1 本

2)および 3)のパイプラインは，それぞれ SBM へのデータのロード/ストア処理を行うために使用する．

これらのデコードステージ以降のパイプライン構成を図 4 に示す．基本的に SPARC64 V の実行パイプラインと同様の動作を行うが，スカラコア部から SIMD-PE への命令送信やコミット指令の送信が長距離配線となるため，下記の 2 ステージを追加した．

- c ステージ（Communication）ステージ　スカラコアからの SIMD-PE へデータを転送するサイクル．
- W1 ステージ（Write back 1）ステージ　コミット処理のための指示を SIMD 処理部からスカラ処理部に転送するサイクル．

SIMD-PE での命令はアウトオブオーダで実行が行われ，命令のコミットの制御はスカラコア部で行う．

2.5 チップ面積の試算

90nm プロセスルールで作られた SPARC64 V プロセサ[1]の面積から推定した，SIMD プロセサの面積の推定を表 2 に示す．この表から分かるように，プロセサコア部分の比較では，16 個の SIMD-PE の SIMD 機構の追加によるチップ面積増は 50%程度であると見積もられる．

しかし，メモリコントローラとインタフェースに必要な面積は無視できず，SIMD 機構の追加に伴いメモリインタフェースの増加が必要になる場合は，さらにチップ面積が増加する．

表 2　90nm SIMD プロセサチップの面積推定
Table 2 Chip Area Estimation with 90nm Design Rule

	SPARC64 V	16xSIMD-PE
スカラコア	110 mm²	
SIMD PE	2FPU (6.25 mm²)	50 mm²
SBM 64KB / PE		50 mm²
SIMD ロジック		15 mm²
4MB L2 キャッシュ	120 mm²	
コア+キャッシュ	230 mm²	115 mm²
メモリコントローラ +インタフェース	~10mm²/XDR (CELL [2]) ~20mm²/FB-DIMM (Niagara II) [3]	

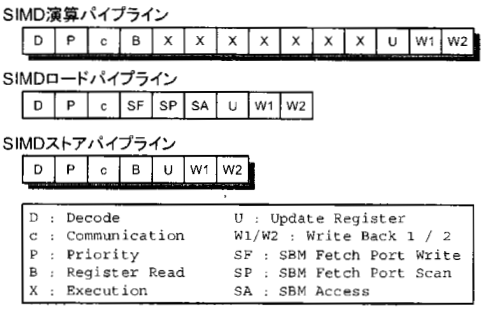


図 4　SIMD 実行パイプラインの構成
Figure 4 Structure of SIMD Execution Pipeline

3. 性能評価

3.1 評価環境

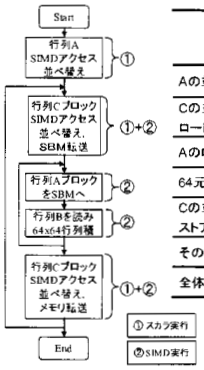
プロセッサ内部の各種機能ユニットやその動作を 1 クロック単位でシミュレーションし，性能評価情報が取得できるように，トレースベースのプロセッサシミュレータ PSI-PSIM を開発した．開発にあたっては，既存 SPARC64 V 向けプロセッサシミュレータ[1]を拡張実装した．このシミュレータは，アーキテクチャを構成する各種資源等をシミュレータへの入力パラメータとして柔軟に変更できる．また，新規追加した SIMD 拡張命令を出力する FORTRAN コンパイラも合わせて開発し，これらを用いて行列積計算である DGEMM（倍精度実数）と ZGEMM（倍精度複素数）および FFT の評価を行った．

3.2 行列積の計算アルゴリズム

行列積は，以下の式で求められる．

$$C(i,j) = \sum_{k=1,K} A(i,k) \times B(k,j)$$

C(1,j)の計算を SIMD-PE0，C(2,j)の計算を SIMD-PE 1 というように各 SIMD-PE にデータ分割して計算を行う．これを行うために，A(1,k)を各 SIMD-PE に格納し，B(k,1)を k の順にスカラコア部でメモリから読み込み全 PE にデータ転送した後，積和演算を繰り返す．性能評価に使用したプログラムでは 32 個の浮動小数点レジスタを使って 4x2 のレジスタブロッキングを行っている．FORTRAN プログラムの場合第 1 添え字の順にメモリに格納されているので，ブロック単位で PE に割り当てるために行列 A および C のデータ並べ替えを必要とする．また，メモリアクセスを減少させるため，SBM において 64x64 要素のブロックキングを行っている．結果として，行列積の計算のフローは図 5 のようになる．



	コンパイラ [サイクル]	アセンブラチューニング [サイクル]	比例係数
Aの並べ替え	21,486	21,486	M*K
Cの並べ替えロード	29,623	16,629	M*N
Aのロード	482	482	M*N*K
64元行列積	30,298	23,911	M*N*K
Cの並べ替えストア	27,565	18,869	M*N
その他	509	507	1
全体	109,963	81,884	

図 5　行列積の処理フローと実行サイクル数

Figure 5　Matrix Multiply flow-diagram and Execution Cycles

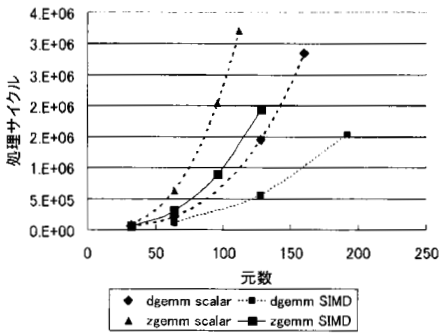


図 6　スカラコアと SIMD の行列積実行サイクル比較

Figure 6　Matrix Multiply Execution Cycles: Scalar vs. SIMD

行列 A は複数回使用されるので，並べ替えを行った形式のデータはメインメモリに置き，再利用を行う．一方，行列 C は 1 回しか使用されないので，スカラコア部でメモリから読み込み，並べ替えを行って SFPR を経由して SBM に書き込む．また，計算結果は，この逆のパスで格納する．

3.3　行列積の実行性能

SIMD コンパイラおよび PSI-PSIM を用いて DGEMM 計算ルーチンのシミュレーションを行った．シミュレーション時の各種パラメータを表 3 に示す．なお，プリフェッチが有効に働くものとして，スカラコア部の L2 キャッシュは完全ヒットの状態でシミュレーションした．

レジスタブロッキングを行った行列演算処理（4x4 の C の部分行列 += 4x2 の A の部分行列 × 2x4 の B の部分行列）を行う最内ループの命令数は，アセンブラチューニングを行ったコードでは 68 命令である．これらは，以下のような命令で構成される．

(1)　浮動小数点積和演算命令：32 命令，
(2)　スカラコア部でのロード命令：8 命令
(3)　スカラコア部から SIMD-PE の SIMD レジスタ SFPR へのデータ転送命令：8 命令
(4)　SIMD レジスタからのロード命令：8 命令
(5)　整数演算，条件分岐などの命令：12 命令

スカラコアのロード命令は最大 2 命令/サイクルで実行可能であるが，ロード値を SIMD レジスタへ転送する命令は，SIMD-PE で 1 サイクルに 1 命令しか実行できない．そのため，(2)で 4 サイクル，(3)で 8 サイクルを必要とする．(1)の浮動小数点積和演算命令は，1 サイクルに 1 命令実行であり，32 サイクルを要する．従って，ループ 1 回あたりの実行サイクル数は最小 44 サイクルであり，シミュレーションしたところ 46 サイクルを要した．

上記以外の 20 命令による実行サイクルオーバヘッドが 2 サイクルと小さいのは，制御を行うスカラコア部が 4 命令発行のアウトオブオーダ制御を行っており，サイクルを決める主要命令と並列にこれらの命令を実行できているためである．

スカラコアだけを使う富士通製 BLAS ライブラリに含まれる DGEMM/ZGEMM を用いた行列積と SIMD プロセサの実行サイクルの比較を図 6 に示す．

図 6 から求めたスカラコア DGEMM の行列積コアループの実行サイクルは，64 元の場合で約 200K サイクルである．一方，アセンブラチューニングしたコードを SIMD プロセサで実行した場合は，23.4K サイクルであり，十分大きな行列の場合は，8 倍の性能が得られる．また，外挿した 1,024 元のケースでは，SIMD プロセサは 18.8Flop/Cycle となり，スカラコア単体の 6.98 倍の性能

表 3　性能評価におけるシミュレーションパラメータ

Table 3 Simulation Parameter of Performance Evaluation

命令発行幅	4 命令 / サイクル
演算パイプライン数	INT : 2, FP : 2, LD/ST : 2
INT/FP リザベーションステーション	INT : 8 エントリ x 2 FP : 8 エントリ x 2
リオーダバッファ	64 エントリ
L1 キャッシュ	命令 128KB (2-way SA) / データ 128KB (2-way SA)
L2 キャッシュ	共有 6MB (12 way SA)
SIMD-PE 数	16
SIMD Reservation Station	10 エントリ
SBM	64KB × 16
SBM からメモリへの outstanding リクエスト数	16 エントリ
メモリアクセスレイテンシ	120 サイクル

となる．また，SIMD 拡張スカラプロセッサでの 1024 元 ZGEMM 実行性能は，20.2Flop/Cycle であり，スカラコア単体の 5.4 倍の性能となる．

表 4 に本研究で想定した 4 命令発行のスカラコアを用いた場合と，命令デコード数や実行ユニット数を半減した 2 命令発行スカラコアを用いた場合の行列積の実行サイクル数を示す．この結果に見られるように，4 命令発行コアを用いることにより，スカラ処理部では 1.66 倍の性能が得られている．また，命令のデコードやコミットの自由度が増加することから，SIMD 処理部でも 10%強の性能向上が見られており，高性能スカラコアを制御部として採用することの有効性を示している．

表 4　スカラコアの行列積実行サイクル数への影響

Table 4　Effect of Scalar Core structure on Matrix Multiply Execution Cycles

	4 命令発行コア [サイクル]	2 命令発行コア [サイクル]	2 命令発行コアを 1 とした性能比
スカラ処理部	78,764	130,609	1.660
SIMD 処理部	30,780	34,014	1.105

3.4　複素 FFT 実行性能

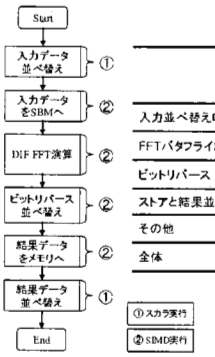
SIMD 処理部は，2 個の PE をペアとして，複素数の積和計算を高速化するクロス演算命令を追加しており，8 個の SIMD-PE ペアで 8 個の複素 FFT 計算を並列実行する．

一般に，メモリ上ではそれぞれの FFT データが連続したメモリ領域に格納されているので，8 並列に FFT を実行するためには行列の転置と同様なデータの並べ替えを必要とする．また，FFT を行った結果についても逆の並べ替えを行って，それぞれの FFT の結果配列に格納する必要がある．

前節と同じ条件でシミュレーションを行った FFT の処理フローと実行サイクル数を，図 7 に示す．富士通製 SSL2（数値計算ライブラリ）[4]の FFT ライブラリを用いてスカラコアのみで FFT を実行した場合は，44K サイクルを要した．これに対して SIMD プロセサの実行サイクルは 2.73 倍の 120K サイクルであった．同時に 8 個の FFT を並行して実行しているので，スループットとしては 3.4Flop/cycle であり，スカラコア単体と比較すると 2.93 倍の性能である．

4.　おわりに

本論文では，4 命令同時発行可能なスカラコア部に最大 32 個の SIMD 演算ユニットを接続するアーキテクチャを提案し，その性能評価を行った．その結果，16SIMD の場合スカラコア単体と比較して，1024 元の DGEMM では約 7 倍，ZGEMM では 5.4 倍の性能を実現可能である結果を得た．また，1024 点の複素 FFT の場合は，約 3 倍のスループットが得られることを示した．16PE の追加によるチップ面積の増加は，スカラコア＋キャッシュの 50%程度と見積もられ，スカラコア単体と比較して，チップ面積あたりの実行性能で評価すると，DGEMM の場合は 4.7 倍，ZGEMM では 3.6 倍，FFT では約 2 倍に改善される．



	コンパイラ [サイクル]	アセンブラチューニング [サイクル]	比例係数
入力並べ替えロード	33,861		N
FFTバタフライ計算	65,985	46,752	Nxlog2N
ビットリバース	6,936		N
ストアと結果並べ替え	31,803		N
その他	1,058		1
全体	139,643	120,410	

図 7　1024 点 FFT の処理フローと実行サイクル数

Figure 7　FFT flow-diagram and Execution Cycles

今後は，SIMD コンパイラの改善や電力評価，SIMD-PE 数を増減した場合の性能評価を継続して行い，電力あたり性能の優れたトレードオフポイントについて研究する予定である．

謝辞　本研究は，文部科学省から委託された「ペタスケール・システムインターコネクト技術の開発」の一部として実施したものである．また，本研究を行うにあたり多くの助言を頂いた富士通の坂本真理子氏，富士通研究所の木村康則氏，PSI 関連の九州大学研究員の方々に感謝します．

参考文献

[1] UNIX サーバ用プロセッサ「SPARC64 V」 http://primeserver.fujitsu.com/primepower/catalog/data/pdf/sparc64_v_j.pdf, Aug, 2004.

[2] O.Takahashi, et.al., "The Circuits and Physical Design of the Synergistic Processor Element of a CELL Processor", Symposium on VLSI Circuits, 2-3, June, 2005.

[3] U.Gajanan, et. al., "An 8-Core 64-Thread 64b Power-Efficient SPARC SoC", Tech. dig. ISSCC 2007, Feb., 2007, pp.108-109.

[4] Fortran & C Package Family, http://jp.fujitsu.com/group/fqs/services/dev-tech/fortran/function/