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

2005－ARC－164（8）
2005／8／3

オンチップRAM利用による電力性能の最適化と評価

高橋睦史[†]　佐藤三久[†]　高橋大介[†]
朴泰祐[†]　中村宏[††]
近藤正章[††]　藤田元信[††]

近年のプロセッサは消費電力あたりの性能が重要となっている．我々はキャッシュの代替としてオンチップ RAM を用い，主記憶・プロセッサ間のデータ転送の最適化を行うことで電力性能を改善することを提案し，その有効性を確認するために実在のプロセッサ SH4 を利用して評価を行った．その結果，オンチップ RAM を適切に使用することでデータ転送を最適化でき，Energy Delay Product (EDP) を最大で約 15.2% 削減できることがわかった．また，オンチップ RAM・主記憶間で DMA 転送がサポートされる場合は EDP を約 26.3% 削減できることがわかった．

Optimization and Evaluation of Power Performance by Using On-Chip RAM

Chikafumi TAKAHASHI,[†] Mitsuhisa SATO,[†]
Daisuke TAKAHASHI,[†] Taisuke BOKU,[†]
Hiroshi NAKAMURA,[††] Masaaki KONDO[††]
and Motonobu FUJITA[††]

In recently years, power consumption per performance has become more important. We propose to improve power performance by optimizing data transfer with on-chip RAM which is a substitution of cache. To confirm effectivity of our proposal, we evaluate power consumption with SH4 commercial processor. The evaluation indicates that using on-chip RAM makes data transfer optimal, and the optimization reduces about 15.2% EDP (Energy Delay Product) at the maximum. If SH4's on-chip RAM supports DMA between the on-chip RAM and main memory, it is estimated that the optimization reduces about 26.3% EDP at the maximum.

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してきた．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得た．しかし代償としてプロセッサは高い消費電力によって猛烈に熱を発し，その発熱が制約となり性能向上が妨げられることとなった．したがって現在では単純に高い性能を求めるのではなく，消費電力あたりの性能（電力性能）を向上することが重要となってきている．また発熱に関する問題はシステム全体に関係することから，プロセッサだけではなくメモリやバスなど周辺も含めた電力性能も注目されている．

現在ほぼすべてのプロセッサではレジスタ・主記憶（メモリ）間のデータ転送にキャッシュが用いられている．キャッシュを用いることで主記憶へのアクセスを減らし，結果としてメモリアクセスに起因する消費電力を削減している．しかしキャッシュは実行時に暗黙的に選択されたデータを再利用するものであり，必要なデータがキャッシュから追い出されたり，必要のないデータが読み込まれるなどの問題が存在する．本来不要であるデータの転送や，データ転送量の増加に伴う実行時間の増加は無駄な電力消費を生じ，電力性能に悪影響を与えている．

そこで我々はオンチップ RAM の利用して電力性能を改善する手法を提案する．この手法はオンチップ RAM を用いてデータ転送をソフトウェアで明示的に制御し，データ転送にまつわる電力を削減することで電力性能を改善する．また，その電力性能を実際のプロセッサを用いて評価を行い，提案手法の有効性を評価する．

2. オンチップ RAM 利用による電力性能最適化

2.1 オンチップ RAM

オンチップ RAM とはプロセッサと同一チップ上に実装される RAM のことである．オンチップ RAM 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 と呼ばれるアーキテクチャを提案している[1),2)]．これはキャッシュと同じメモリ階層に，S-RAM を用いたオンチップ RAM を実装するアーキテクチャである．Patterson らは IRAM (Intelligent RAM) と呼ばれる，チップ上に D-RAM を実装したアーキテクチャを提案している[3)]．Kogge らは PIM (Processor-In-Memory) と呼ばれる，メモリモジュール内に演算器を有するアーキテクチャ

† 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
†† 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

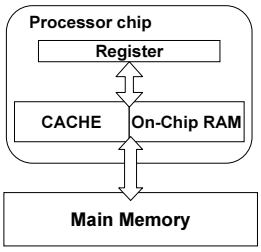

図 1 オンチップ RAM モデル

を提案している[4]．

商用プロセッサでは，プロセッサチップ上に一時的に計算結果などのデータを保管するための scratch pad RAM を搭載するプロセッサがある[5),6)]．BlueGene/L[7)] に使用されている PowerPC プロセッサでは L3 キャッシュを RAM として使用できる[8)]．また SH4 プロセッサではデータキャッシュの半分の容量をオンチップ RAM として使用できる[9)]．

本稿ではオンチップ RAM を利用した電力性能の改善を提案する．これは従来における一般的なプロセッサアーキテクチャを改良してキャッシュの一部をオンチップ RAM とし，オンチップ RAM・主記憶間のデータ転送をソフトウェアで効率的に制御することで電力性能の改善を図るものである．提案手法によるデータ転送のモデルを**図 1** に示す．図のようなモデルのオンチップ RAM を導入することで，次の利点が生じると考えられる．

(1) 演算に必要なデータを明示的に指定することで，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できる．

(2) 演算に必要となるより前にデータをオンチップ RAM に転送しておくことでデータプリフェッチが行える．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．オンチップ RAM 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．

なお，オンチップ RAM はソフトウェアでデータ転送を制御する都合上，キャッシュのみを用いるプログラムと比較して実行命令数は増加する．

2.2 電力性能

プロセッサおよびメモリなどで消費される電力は，リーク電流などにより常時消費されるスタティック電力とスイッチング動作により消費されるダイナミック電力に分けられる．クロックラインなど待機時にスイッチング動作により消費される電力は通常ダイナミック電力に分類されるが，本稿では議論を簡便にするために待機時の消費電力全体をスタティック電力，動作時の消費電力からスタティック電力を除いたものをダイナミック電力とする．

スタティック電力が一定であると仮定したとき，スタティック電力を P_{static}，ダイナミック電力量を $E_{dynamic}$，動作時間を t とすると，全体の消費電力量 E_{all} は以下の式で表される．

$$E_{all} = P_{static} * t + E_{dynamic}$$

前小節（1）で述べた，メモリアクセスを最適化することによるデータ転送量の削減はバスおよびメモリモジュールのスイッチング動作を削減する．したがって，メモリアクセスの最適化は直接的にダイナミック電力量 $E_{dynamic}$ の削減につながる．加えて（1），(2）による性能の向上，すなわち実行時間 t の短縮はスタティック電力 P_{static} による消費電力量を削減できる．したがってオンチップ RAM を用いることは消費電力量削減に大きな効果があると考えられる．

なお本稿で想定するオンチップ RAM の構成は我々の提案している SCIMA の構成に類似している．シミュレーションベースの評価において SCIMA の電力性能の評価は既に行われており，オンチップ RAM の有効性が確認されている[10)]．本稿では実際のプロセッサを取り上げて，実際の電力評価の結果に基づいてオンチップ RAM の有効性を検討する．

2.3 オンチップ RAM 向け最適化手法

本稿ではオンチップ RAM をキャッシュと同じメモリ階層に置くことを想定し，オンチップ RAM 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 RAM を利用した最適化は

(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 RAM にデータを転送

(2) オンチップ RAM 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

といった手順で行われる．このとき，データセットがオンチップ RAM よりも大きい場合は何らかのブロッキングを行う必要がある．

時間的局所性を利用する最適化のほかにはデータ転送時間の隠蔽が考えられる．オンチップ RAM がダイレクトメモリアクセス (Direct Memory Access: 以降，DMA) をサポートし，演算とデータ転送のオーバーラップが可能である場合は，データプリフェッチを行うことでデータ転送の完了待ちによる演算のストールを削減することが出来る．このような最適化を行う場合，データ転送のソフトウェアパイプライニングが効果的である．

なお，オンチップ RAM 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 RAM 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 RAM 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．

具体的な最適化手法については “3.4 オンチップ RAM 向け最適化” の項で示す．

3. 性能評価環境

3.1 評価用マシン

本稿では実在のプロセッサである SH4 シリーズより SH7751R を用いて電力性能の評価を行う．SH4 は（株）日立製作所（現：(株）ルネサス テクノロジ）によって作成された 32bit スーパースカラ型 RISC プロセッサである．SH7751R は 32KB のデータキャッシュを持つが，このキャッシュ領域の半分をオンチップ RAM として用いることができる．以降，オンチップ RAM を使用するモードを OCR モードと呼び，従来のキャッシュのみのモードを CACHE モードと呼ぶ．オンチップメモリに関する詳細を**表 1** に示す．OCR モードではオンチップメモリの半

分に対して特定のアドレス領域が付加され，move 命令などによるデータの書き込みと読み出しが可能となる．

評価用マザーボードには（株）日立超 LSI システム製 Solution Engine を用い，OS は Linux をインストールして使用した．諸元を**表 2** に示す．

3.2 電力測定環境

電力測定用機器として（株）シナジェテック製 CT-30000 を用いた．これはホール素子を用いた導線の磁場変動の観測により，導線に流れる電力を測定する電力測定装置である．このホール素子を Solution Engine 上の各 ATX 電源入力、およびマザーボード上に実装されたプロセッサコア用三端子レギュレータの入出力に設置し，各電力を測定する．本稿では以下の 2 種類の系統に分けて電力を測定する．

- CORE: 三端子レギュレータの出力（1.5V）を測定したもの．この出力は全量がプロセッサコアに供給される．プロセッサではこの電源を I/O 以外のプロセッサコアで消費することから，CORE はプロセッサコア中の I/O 部分を除いたレジスタ，キャッシュ，およびオンチップ RAM などの消費電力を示す．
- MEMORY: ATX3.3V 系の電力から三端子レギュレータへの入力電力を除いたもの．Solution Engine ではこの 3.3V 系がメモリモジュール，プロセッサコアの I/O 部分，およびプロセッサバスなどに供給されており，メモリ関連のデバイスにおける消費電力の変動が MEMORY の電力値に現れる．

なお，Solution Engine では ATX 5.0V 系も使用されているが，そのほとんどがプロセッサやメモリ以外で消費され，また本稿中の評価で使用されないデバイスに多くの電力を供給していることから，以後，この ATX 5.0V 系で消費される電力については扱わず，前述の CORE および MEMORY のみを扱うこととする．

3.3 メモリアクセス時の基本電力特性

予備評価として，簡単なデータスキャンプログラムの実行時の消費電力を測定した．測定結果を**図 2** に示す．これは，連続したメモリ領域に対してスキャン領域を段階的に増やしつつ反復的にアクセスするプログラムであり，データアクセスがキャッシュヒットしている状態から徐々にキャッシュミスしていく様子を観測することが出来る．

図 2 を見ると，測定開始から約 0.2 秒後にプログラムが起動し CORE が約 360mW に上昇していることがわかる．このとき，しばらくはキャッシュにヒットするために MEMORY は約 1.6W の値を保つが，1.7 秒以降は CORE

表 1 SH4（SH7751R）オンチップメモリ

モード	D-Cache	オンチップ RAM
CACHE モード	32 KB (2 way)	0 KB
OCR モード	16 KB (1 way)	16 KB

表 2 Solution Engine 諸元

名称	SuperH Solution Engine
型番	MS7751RSE01
CPU	SH7751R (SH-4) 240MHz
メモリ	64MB SDRAM 60MHz
I/O	CompactFlash, IDE, etc
OS	Linux 2.4.18
コンパイラ	gcc 3.2.3

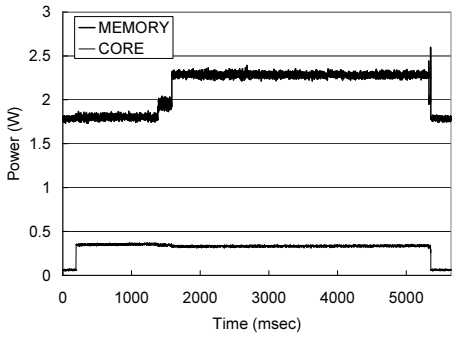

図 2 データスキャンプログラムの消費電力

が約 15mW 低下する代わりに MEMORY が約 500mW 上昇する．これはキャッシュミスの頻発によるメモリアクセスの増加で MEMORY の値が増加し，データ転送待ちによる命令実行のストールで CORE の値が減少していると考えられる．

また約 5.3 秒後にプログラムが終了し，その後はアイドル状態となっているが，このときの消費電力を見ると CORE が約 65mW と低い水準となっているのに対して MEMORY は約 2.3W と非常に大きいことがわかる．これは今回使用した Solution Engine が ATX サイズの開発用ボードであり，メモリ関連以外のデバイスにも常時，一定量の電力を供給しているためである．

3.4 オンチップ RAM 向け最適化

SH4 の制約上，主記憶・オンチップ RAM 間のデータ転送は通常の転送命令を用いて実行される．つまり，2 者間のデータ転送は move 命令等により実装され，転送データは必ずレジスタ及びキャッシュを経由することとなる．またキャッシュを経由することでバスを経由する際のデータの粒度は 32Byte で固定となり，32Byte 未満のデータを転送する場合は主記憶・キャッシュ間において本来不要であるデータの転送を生ずる．

また表 1 で示したように，SH4（SH7751R）のオンチップ RAM モードではチップ上に実装されているメモリの半分である 16KB のみをオンチップ RAM として利用する．そのため CACHE モードでのキャッシュと比較して OCR モードでのオンチップ RAM の容量は相対的に小さく，最適化の際はキャッシュも有効に活用することを考えなければならない．

以上のことを踏まえ，プログラムに対して時間的局所性を考慮した最適化を行う．以下に 3 種類のプログラムについて具体的な最適化方法を示す．

3.4.1 行列積演算

まず $C = A \times B$ で表される N 次の倍精度行列積演算 (以降，MM) について述べる．行列積演算を単純に記述すると 3 重ループ構造となるが，最内側ループ内でのデータセットがキャッシュ容量を上回った場合，時間的局所性を生かすことができない．そこで最適化手法の 1 つとしてキャッシュブロッキング[11]（タイリング）がしばしば用いられる．

しかし行列積演算にタイリングを用いた場合，アドレスが不連続になる方向のアクセスにおいてラインコンフリクトが生じる可能性がある．通常，ラインコンフリクトが発生した場合 n-way set assiciative キャッシュを用いた

り，データのアクセスパターンにソフトウェア的な変更を行うことでコンフリクトを回避する．行列積においては行列の転置を行うことによりコンフリクトが回避できる．しかし，配列間のラインコンフリクトは回避できない．

オンチップ RAM 向けアルゴリズムでは CACHE 向け最適化で行った，キャッシュブロッキングされた A, B, C の行列の部分領域のうち，A, B の部分領域についてオンチップ RAM に転送して演算を行う．これにより行列 A, B に関してキャッシュミスは生じなくなる．行列 C に関してはキャッシュを利用することになるが，単一の行列のみでキャッシュを利用するため，ラインコンフリクトはほとんど生じない．

今回の評価では行列サイズを 500×500 とし，ブロッキングサイズは予備実験で最も性能の高かったブロッキングサイズ（CACHE モード 36×36，OCR モード 32×32）を用いた．

3.4.2 NAS Parallel Benchmarks - CG

次に NAS Parallel Benchmarks[12] CG（以降，CG）について述べる．CG は正値対称な大規模疎行列の固有値を Conjugate Gradient 法によって求めるベンチマークである．このベンチマークのカーネルとなる行列・ベクトル積は $q = \sum_i (A[i] \times p[colidx[i]])$ という式で表現され，ベクトル q の 1 要素を求めるために，行列 A への連続アクセスと，ベクトル p の要素への間接アクセスが生じる．

式中で時間的局所性のあるデータはベクトル p のみであるが，$colidx$ を用いて間接参照されることでベクトル p 全域に対するランダムアクセスとなり局所性を生かすことが難しい．また，再利用性の無い A と $colidx$ および p が配列間でラインコンフリクトを発生することでキャッシュミスが多発する．

このとき，ベクトル p をブロック分割し，その分割したブロックにアクセスするよう A と $colidx$ をあらかじめ p のブロックにあわせて並べ替え，p の参照に空間的局所性を持たせることによって，p についてはキャッシュブロッキングによる性能改善が可能となり，キャッシュミスを軽減できる[13]．しかし 3 種類の配列間でのラインコンフリクトは抑制できず，p の再利用性を生かせない恐れがある．

ここでオンチップ RAM の利用を考えると，先ほどキャッシュブロッキングを行った p をオンチップ RAM に転送することで，p の再利用性を完全に利用することが出来る．このとき $A, colidx$ で配列間のラインコンフリクトが発生する恐れはあるが，この 2 種類の配列のアクセスパターンは順次アクセスであるのでコンフリクトを生じる可能性は少なく，また再利用性もないので影響は少ないと考えられる．

今回の評価では Class W を用いた．このときの A は 7000×7000 要素で非零率 1%の疎行列，p は 7000 要素のベクトルである（精度は倍精度浮動小数）．ブロッキングサイズは予備実験で最も性能の高かったブロッキングサイズ（CACHE モード・OCR モードともに 4 分割）を用いた．

3.4.3 FFT: Fast Fourier Transform

3 つ目に Fast Fourier Transform[14] (以降，FFT) について述べる．FFT は離散 Fourier 変換を高速に計算するアルゴリズムであり，大規模科学技術計算から信号処理まで幅広く用いられている．本稿では 3 次元倍精度複素 FFT プログラムを評価用プログラムとして利用する．

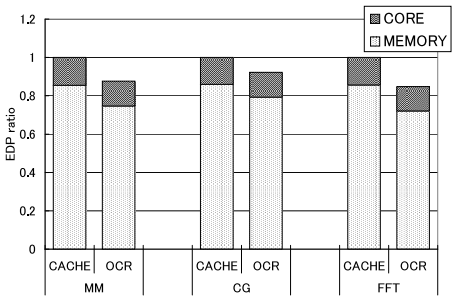

図 3 SH4 の EDP 比

3 次元 FFT では行列の各次元軸方向のデータ同士で演算を行うが，アドレスが連続とならない次元方向ではデータに対してストライドアクセスが発生する．このときアクセス単位がキャッシュラインサイズ未満である 16Byte 単位の不連続アクセスになるので，32Byte キャッシュラインの SH4 では本来不要であるデータも含めて転送されてしまい，容量の半分しか有効に利用できない．また，転送粒度が小さくなるためにループによるオーバヘッドも発生する．これを防ぐためにアドレスが連続となる方向でブロッキングを行い，キャッシュラインサイズ以上のブロック単位でデータを転送した上で演算を行う[15]．

ブロッキングを行った後のデータは複数の配列にコピーしながら処理されるため，アドレス確保時にパディングを用いることで配列間のキャッシュコンフリクトを防ぐことができる．しかし，これはコンフリクトを完全に解消するには至らない．ここでブロッキングデータをオンチップ RAM 内で閉じて使用することで，コンフリクトが発生する恐れは無くなる．ただし，SH4 のオンチップ RAM は CACHE モードのキャッシュ容量の半分であり，ブロッキングサイズは必然的に小さくなってしまう．

今回の評価ではデータサイズ $64 \times 64 \times 64$ の 3 次元 FFT を 2 回実行し評価する．ブロッキングサイズは予備実験で最も性能の良かった値（CACHE モード 16, OCR モード 8）を使用した．

4. SH4 による性能評価結果

本評価では電力性能の指標としてエネルギー遅延積 (Energy Delay Product: 以降，EDP) を使用する．EDP は消費電力量に時間を乗じたものであり，電力あたりの性能を表す指標として広く用いられている．**表 3** に SH4 で測定した各プログラムの測定結果を，**図 3** に CACHE モードでの EDP を 1 として正規化した EDP 比を示す．

図 3 を見ると MM，CG ともにオンチップ RAM を利用することにより EDP が削減されていることがわかる．CACHE モードと比較して MM では約 12.4%, CG では約 7.8%， FFT では 15.2% EDP が削減されている．

表 3 のダイナミック消費電力量 $E_{dynamic}$ に注目すると，どのプログラムの CORE および MEMORY でも OCR モードはダイナミック消費電力量が減少していることがわかる．MEMORY の消費電力量の減少はオンチップ RAM を用いたことでキャッシュコンフリクトが抑制され，データ転送量が削減されたためと考えられる．CORE の電力

表 3　SH4 の電力性能

		MM CACHE	MM OCR	CG CACHE	CG OCR	FFT CACHE	FFT OCR
EDP $(W*sec^2)$	CORE	35.5	31.4	761.0	699.8	2.892	2.561
	MEMORY	207.0	181.0	4615	4258	17.10	14.38
	Total	242.5	212.5	5376	4958	19.99	16.95
$E_{dynamic}$ $(W*sec)$	CORE	2.848	2.675	14.16	13.48	0.826	0.801
	MEMORY	1.876	1.470	22.54	20.89	0.766	0.669
	Total	4.723	4.145	36.70	34.36	1.592	1.471
E_{static} $(W*sec)$	CORE	0.621	0.585	2.781	2.681	0.177	0.164
	MEMORY	18.32	17.27	80.20	77.44	5.163	4.751
	Total	18.94	17.86	82.98	80.13	5.340	4.915
実行時間 (sec)		10.25	9.66	44.93	43.30	2.88	2.66

量の減少はキャッシュミスの減少により無駄な書き換えが減少したことが要因だと考えられる．

ダイナミック消費電力量 $E_{dynamic}$ とスタティック消費電力量 E_{static} の電力削減率を比較すると，どのプログラムにおいても E_{static} の削減率のほうが高い．これは 3.3 節でも述べたように，MEMORY のスタティック電力が大きいためである．SH4 を用いる通常のシステムであれば $E_{dynamic}$ の削減が電力性能改善に寄与する割合は，本評価より大きくなると考えられる．

5. 考　　察

5.1 SH4 オンチップ RAM による電力性能改善

前章より，オンチップ RAM を用いることにより SH4 の電力性能を向上させることがわかった．単純に命令実行数を考えると，CACHE モードでは必要ない転送命令が挿入されている OCR モードのほうが消費電力的に不利であり，大幅に実行命令数が増えた場合は CORE のダイナミック消費電力は増加するはずである．しかし，ダイナミック電力量をみると CORE については OCR モードのほうが消費電力量が少ない．MEMORY のダイナミック消費電力量も減少していることから考えると，これは実行命令数増加のデメリットを，メモリアクセス量削減によるキャッシュ書き換え回数の減少と実行時間短縮というメリットが上回っていると考えられる．

また，表 3 の CORE の欄を見ると，どちらのプログラムでも OCR モードのほうがダイナミック消費電力量が少なかった．これは前述のとおり CACHE モードではキャッシュミスによるデータ転送があり，その無駄な転送をオンチップ RAM により削減できたためと考えられる．しかし OCR モードではオンチップ RAM の転送にキャッシュを経由しているので，CORE 電力にはこのキャッシュを経由するために必要な電力量も含まれている．これは OCR モードでキャッシュを経由することによって増加する電力量よりも，CACHE モードで無駄なデータ転送に費やされる電力量のほうが多いことを示しており，オンチップ RAM がキャッシュと比較して良好な電力性能を持つことがわかる．

ところで，SH4 のオンチップ RAM は CACHE モード時のキャッシュサイズに比べて容量は半分となり，必然的にブロッキングサイズが小さくなる．MM の場合，CACHE モードでは最大 36×36 のブロッキングサイズを取れるが OCR モードでは 32×32 となり，CACHE モード比で約 79%のブロッキングサイズとなってしまう．FFT の場合では最適化の結果，CACHE モードの半分のブロッキングサイズとなった．ソフトウェアで制御するというオンチップ RAM による最適化の限界により，キャッシュとオンチップ RAM の同時利用は必須であり，オンチップ RAM 容量が従来におけるキャッシュの容量よりも小さくなることは避けられない．しかし，前章の結果より適切な最適化手法を行うことでオンチップ RAM を利用する効果は十分に得られることが分かった．

5.2 DMA 機能付き SH4 による電力性能

SH4 のオンチップ RAM・主記憶間のデータ転送はキャッシュ及びレジスタ経由となっている．これによりオンチップ RAM・主記憶間のデータ転送中は演算実行が阻害されるので，OCR モードでは CACHE モードよりもより多くの電力が消費されていると考えられる．この問題はオンチップ RAM への DMA 機構の付加により解消が見込まれる．

DMA が付加されることにより，オンチップ RAM・主記憶間のデータ転送を行っても演算が阻害されることは無い．加えて，アルゴリズム的に変更を加えれば演算とデータ転送をオーバーラップするソフトウェアパイプライニングが行える．また，DMA 転送では転送データはキャッシュを経由しないので，キャッシュラインサイズ単位に固定されていたデータ転送の粒度が自由に選択できることになり，演算に不要なデータの転送を抑制できる．これにより実行時間の短縮によるスタティック電力量の削減とデータ転送量の削減によるダイナミック電力量の削減が見込まれる．

そこで SH4 に DMA 機能が付加されたと仮定した場合の電力性能を FFT を用いて評価する．想定するプロセッサは SH4（SH7751R）のオンチップ RAM に DMA 転送を付加したものとし，主記憶と直接のデータ転送を演算のバックグラウンドで実行できるとする．また，DMA 転送にはブロックストライド転送を指定できるものとする．

予備実験の結果から，今回使用した FFT では演算時間はデータ転送時間と比較して十分長いことが確認されている．そこで FFT 中のデータ転送について演算とデータ転送をパイプライニング化できるようにアルゴリズムを変更し，オンチップ RAM を半分の容量で分割し転送と演算で交互に使用する．これに伴いブロッキングサイズはより小さくなり，値は “2” となる．演算時間とデータ転送時間がパイプライン化可能なデータ転送の比率は 3 次元複素行列 A の要素数を $NX*NY*NZ$，ブロッキングサイズを $NBLK$ とすると次の式で求められる．

$$\frac{\begin{array}{l}5*NX*\log_2 NX*(NY-NBLK*2)*NZ+\\5*NY*\log_2 NY*(NX-NBLK*2)*NZ+\\5*NZ*\log_2 NZ*(NX-NBLK*2)*NY\end{array}}{5*NX*NY*NZ*\log_2(NX*NY*NZ)}$$

今回の評価では上式より全データ転送中 93.75% のデータ転送が隠蔽可能であると求まる．この値と実際の SH4

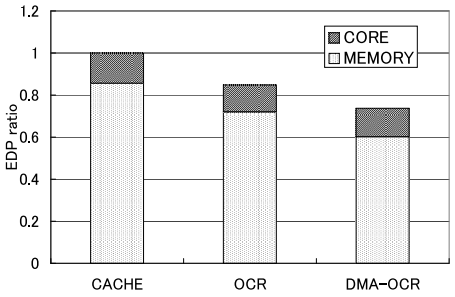

図 4 DMA 機能付き SH4 の EDP 比

での実行結果を利用して電力性能を求める．パイプライニング化を行ったプログラムを SH4 で実行し，そのときの消費電力のうちスタティック電力量を転送時間の隠蔽による実行時間短縮によって削減できると考える．SH4 でのデータ転送所要時間は，パイプライニング化を行ったプログラムと，そのプログラムのデータ転送部分を省略したプログラムの実行時間の差分をとって求める．**図 4** に算出された FFT の EDP 比を示す．

図 4 を見ると，SH4 のオンチップ RAM に DMA 機能を付加することによって EDP が 26.3% 削減されており，DMA を有しない OCR モードと比較しても 13.1% 削減された．この評価ではキャッシュを経由しないことによるメモリトラフィックの削減の影響は含まれてないので，実際では図 4 の結果よりも大きな EDP 削減が見込まれる．

ところで今回検討した DMA 構成は SCIMA とほぼ同等となるが，SCIMA における CG の実行時にはキャッシュだけの条件と比較して 32Byte キャッシュライン時に 70%近くの EDP を削減しており[10]，本稿の検討結果よりも大きな効果を得ている．これは SCIMA ではチップ上に実装された複数 Way によるメモリを，Way 単位でキャッシュ，もしくはオンチップ RAM に切り替えることでキャッシュ・オンチップ RAM 容量比を可変としていることが大きな要因であると考えられる．

なおオンチップ RAM を用いたソフトウェアパイプライニングによる最適化に関しては，キャッシュプリフェッチを用いればオンチップ RAM と同様に CACHE モードの EDP を削減させることは可能である．しかし，キャッシュプリフェッチを行うには正確なプリフェッチ命令の発行が必要であり，適切なタイミングでなければプリフェッチしたデータが追い出されることも考えれる．また，キャッシュラインサイズ単位のデータ転送しか行えず，オンチップ RAM を使用する場合と比較してバストラフィックは増加すると考えられる．

6. ま と め

本稿では，従来における一般的なプロセッサアーキテクチャを改良してキャッシュの一部をオンチップ RAM とするアーキテクチャを使用し，オンチップ RAM・主記憶間のデータ転送をソフトウェアで効率的に制御することで電力性能の改善を図ることの提案を行い，SH4 を用いた実験によりその有効性を評価した．

その結果，オンチップ RAM を適切に使用することでデータ転送を最適化でき，EDP を最大で約 15.2% を削減できることがわかった．また，オンチップ RAM・主記憶間で DMA 転送がサポートされる場合は EDP を約 26.3% 削減できることがわかった．

今後の課題としては，SH4 以外のプロセッサや他のプログラムでの評価の検討や電力シミュレーションとの差異の解析が挙げられる．また，キャッシュプリフェッチを行った CACHE モードの性能との比較も行いたい．

謝辞 本研究に関してご助言，ご討論いただきました CREST チームの各位に感謝いたします．なお，本研究は科学技術振興機構・戦略的創造研究「低消費電力化とモデリング技術によるメガスケールコンピューティング」による．

参 考 文 献

1) 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).
2) 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).
3) Patterson, D. et al.: A Case for Intelligent RAM: IRAM, *IEEE Micro*, Vol. 17, No. 2, pp. 34–44 (1997).
4) Sunaga, T. et al.: A Processor In Memory Chip for Massively Parallel Embedded Applications, *IEEE J. of Solid State Circuits*, pp. 1556–1559 (1996).
5) Diefendorff, K.: Sony's Emotionally Charged Chip, *Microprocessor Report*, Vol. 13, No. 5 (1999).
6) Turley, J.: StrongArm Speed to Triple, *Microprocessor Report*, Vol. 13, No. 6 (1999).
7) Almasi, G. et al.: Unlocking the Performance of the BlueGene/L Supercomputer, *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, p. 57 (2004).
8) Halfhill, T. R.: IBM Makes Designer Genes, *Microprocessor Report* (2004).
9) 日立製作所: SH-4 プログラミングマニュアル, 5 edition (1998).
10) 近藤正章ほか: ソフトウェア可制御オンチップメモリによるメモリシステムの低消費電力化, 情報処理学会研究報告, 2002-ARC-149, pp. 1–6 (2002).
11) Lam, M. et al.: The cache performance and optimizations of Blocked Algorithms, *Proc. ASPLOS-IV*, pp. 63–74 (1991).
12) Bailey, D. et al.: The NAS Parallel Benchmarks 2.0, *NASA Ames Research Center Report*, NAS-05-020 (1995).
13) 岩本貢ほか: NASPB CG，FT における SCIMA の性能評価, 情報処理学会研究報告, 2000-HPC-83, pp. 31–36 (2000).
14) Cooley, J. W. et al.: An Algorithm for the Machine Calculation of Complex Fourier Series, Math. Comput., Vol. 19, pp. 297–301 (1965).
15) Takahashi, D.: Efficient implementation of parallel three-dimensional FFT on clusters of PCs, *Computer Physics Communications*, Vol. 152, pp. 144–150 (2003).