ソフトウェア制御オンチップメモリにおけるスタティック消費電力削減手法

藤田元信[†,††] 田中慎一[†,☆]
近藤正章[†] 中村宏[†]

今後，プロセスの微細化にともないリーク電流が増大し，プロセッサ全体の消費電力おけるリーク電流に由来するスタティック消費電力の割合が大きくなると予測されている．特にキャッシュはプロセッサのスタティック消費電力の多くを占めており，キャッシュにおけるリーク電流の削減は重要な課題である．そのため，近年ではキャッシュのリーク電流削減を目的とした回路的技術やマイクロアーキテクチャの研究が活発に行われている．一方，我々はキャッシュに加えソフトウェア制御可能なメモリをチップ上に搭載し，これを利用した高性能化を図るアーキテクチャSCIMA を提案している．ソフトウェアにより制御されるメモリ上では，将来的にアクセスされる領域とアクセスされない領域を完全に特定できることから，アクセスされない領域を静的消費電力が少ないモードに切り替えておくことで，従来のキャッシュよりも効率的にリーク電流を削減できると考えられる．本論文では，SCIMA のこの特徴を利用したリーク電流削減手法について検討し評価を行った．

A Static Power Reduction Method for Software Controlled On-chip Memory

Motonobu Fujita,[†,††] Shinichi Tanaka,[†,☆] Masaaki Kondo[†] and Hiroshi Nakamura[†]

As semiconductor technology scales down, the static power due to leakage current becomes dominant in the total power dissipation of a microprocessor. Especially, cache memories dissipate a large portion of static power in the processor chip. Therefore, reducing leakage current in cache memories is very important. We have proposed a new processor architecture called SCIMA, which has integrated Software-Controllable Memory (SCM) on the processor chip in addition to the ordinary cache. Because unused part of the SCM can be identified by software, we can effectively reduce a large portion of static power by putting unused SCM part into low leakage mode. In this paper, we propose a leakage current reduction method using SCM, and show preliminary evaluation results.

1. はじめに

近年，プロセッサの消費電力が増大し，モバイルコンピュータのバッテリ駆動時間や高性能プロセッサの放熱の観点から消費電力の削減は重要な課題となっている．プロセッサの消費電力は，主にトランジスタのスイッチングに起因するダイナミック消費電力とリーク電流に由来するスタティック消費電力からなる．従来はダイナミック消費電力がプロセッサの消費電力の大半を占めており，従来研究の多くはダイナミック消費電力の削減を目的としていた[1)~3)]．しかし，今後はプロセッサの消費電力におけるスタティック消費電力の寄与が大きくなると予測されている[4)~7)]．これは半導体技術の微細化にともない，トランジスタが導通し始める閾値電圧が低下し，リーク電流が増加するためである．

近年の高性能プロセッサにおいては，特にチップ上のキャッシュメモリにおけるスタティック消費電力がプロセッサの消費電力のうち大きな割合を占めている．キャッシュはプロセッサチップ上のトランジスタ数の多くの割合を占めており，それに比例してスタティック消費電力が増大するためである．したがって，最近ではキャッシュのリーク電流を抑えるための回路技術やマイクロアーキテクチャの研究が重要となってきて

† 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
†† 科学技術振興機構
Japan Science and Technology Agency (JST)
☆ 現在，株式会社 NTT データ
Presently with NTT Data Corporation

いる．その回路技術として，これまで SRAM のセルに対する電源の供給をオフにすることによりリーク電流を大幅に削減する Gated-Vdd[8] や，電源電圧を制御してリーク電流の削減を狙う DVS[9] などが提案されている．しかし，Gated-Vdd ではいったん供給電源をオフにすると回路が保持している情報が失われるために，結果としてキャッシュミスが増加し，また DVS ではキャッシュアクセスの時間が増加するなど，性能に対するペナルティがある．そこで，将来的にアクセスがないラインを予測し，そのラインの電源を Gated-Vdd によりオフにするなど，性能の低下を抑えつつリーク電流を削減するマイクロアーキテクチャ手法も提案されている[10),11)]．

我々はこれまでに，従来のキャッシュに加えてソフトウエアによりアドレス指定可能なオンチップメモリを持つプロセッサアーキテクチャ*SCIMA*（*Software Controlled Integrated Memory Architecture*）を提案している[12)]．SCIMA は，ソフトウェア制御のオンチップメモリ *SCM*（*Software Controlled on-chip Memory*）を用い，キャッシュに比べて柔軟にデータ転送を行うことで，高性能化を目指すアーキテクチャである．ここで，SCM におけるデータのアロケーション・リプレースメントはソフトウェアから明示的に制御されるため，SCM 上にいつデータを配置し，そのデータがいつ不要になるかをプログラムから判断することが可能になる．

この特徴から，SCIMA は Gated-Vdd などの従来のキャッシュ向けに提案されたリーク電流削減のための回路技術と組み合わせることで，キャッシュよりも効率的にリーク電流を削減することができると考えられる．本論文では，代表的な回路手法のうち，Gated-Vdd を利用した SCIMA によるスタティック消費電力削減手法を提案する．

関 連 研 究

CacheDecay[11)] は，キャッシュアクセスの時間的局所性から，ある一定期間アクセスのないキャッシュラインを Gated-Vdd を用いた Sleep モードとし，リーク電流の削減を図る．

DRI-Cache[8)] は命令キャッシュを対象としている．一定期間キャッシュミス回数をカウントし，ミス回数がある閾値を超えない場合は Gated-Vdd によりキャッシュサイズ（Active 領域）を減らすことでスタティック消費電力を削減している．

DVS を回路技術に採用しているアーキテクチャ的手法としては Drowsy Caches[9),10)] がある．DVS では Sleep モードでも情報を維持できるという利点を活かし，一定サイクルごとにキャッシュ上の全ラインをいっせいに Sleep モードとする．アクセスのあった Sleep ラインは Active モードに戻される．

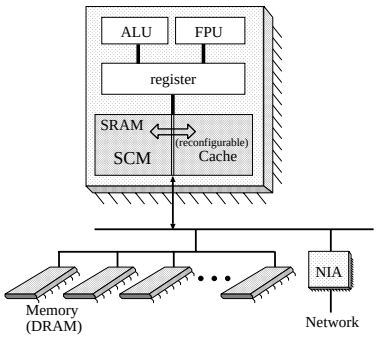

図 1 SCIMA の構成
Fig. 1 Schematic view of SCIMA.

これらの手法は，ハードウェアの予測に基づいた手法であるため，予測がうまくいかなかった場合に性能や電力的なペナルティが大きくなる．これに対し，提案手法はソフトウェアによる制御であり，確実に必要のない SCM 領域のみを Sleep モードにできるため，より効率的にスタティック消費電力を削減できる．

2. SCIMA

2.1 SCIMA のアーキテクチャ

SCIMA は，チップ上に従来のキャッシュに加えソフトウエア制御可能なメモリ *SCM*（*Software Controlled on-chip Memory*）を搭載する（**図 1**）．SCM は論理アドレス空間の一部の連続した領域を占めており（**図 2**），キャッシュと SCM の間にアドレス空間の包含関係はない．SCM 領域は *page* と呼ばれる単位☆に分割・管理され，メモリアクセス順序保証もこれを単位として行われる．キャッシュと SCM はハードウェアとしての SRAM 自体は共有し，アプリケーションの性質に応じてその容量比を動的に変更させることも可能である[12)]．

従来のキャッシュはハードウェア制御により自動的にデータ配置，置き換えが行われるのに対し，SCM では，ソフトウエアから明示的にデータ配置，置き換えを行う．SCM を利用することで，従来キャッシュで発生していた競合によるオフチップトラフィックの増加を抑えることができ，ダイナミック消費電力の削減にも有効であることが分かっている[13)]．

SCIMA では，SCM と主記憶間のデータ転送を行

☆ アーキテクチャによって決まる固定値であり，数 KB 程度のサイズを想定している．

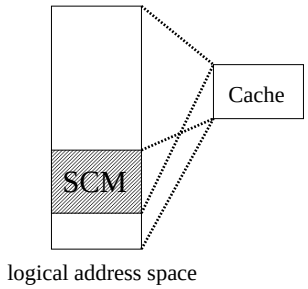

図 2 SCIMA におけるアドレス空間
Fig. 2 Address space.

```
integer i, j, N
real*8 a(N), sum
sum = 0.0
!$scm begin(a, BL, 0)   ← 配列a用に要素BL個分の領域を確保
do i =1, N, BL
!$scm load (a, i, BL)   ← 確保した領域にBL個の要素を転送
   do j=i,i+BL
      sum = sum + a(j)
   enddo
enddo
!$scm_end(a)            ← 配列a用に確保した領域を解放
```

図 3 SCIMA ディレクティブの挿入例
Fig. 3 Example of optimization using SCIMA directives.

う *page-load/page-store* 命令を備える．

本命令は，データ転送元の開始番地，データ転送先の開始番地，転送サイズ，ブロック幅，ストライド幅をオペランドにとる．本命令は *page* を最大サイズとした大粒度転送をサポートし，さらに一定間隔に存在するデータを SCM 上にパッキングするストライド転送機能もサポートしている．

2.2 SCIMA ディレクティブベースコンパイラ

SCM を利用したプログラミングを容易にする目的で，ディレクティブベースコンパイラも提案している[14]．ディレクティブを用いることでソースコードのセマンティクスに影響を与えることなく SCM と主記憶間のデータ転送が可能となる．

図 3 に SCIMA 用ディレクティブを用いたプログラムの例を示す．

SCM 領域を確保するための`!$scm begin` ディレクティブは，引数に配列名，配列の各次元ごとのサイズ，ストライド幅をとる．プログラム実行時には，指定した配列用の SCM 領域が`!$scm begin` の挿入位置で，各次元のサイズの積で表されたサイズ分確保される．SCM 領域の確保を済ませた配列は`!$scm load` ディレクティブにより SCM に転送される．`!$scm load` は引数で配列名，転送の起点となる要素，各次元ごとの転送サイズを指定する．また`!$scm begin` で確保された領域は，対応する`!$scm end` ディレクティブで解放される．

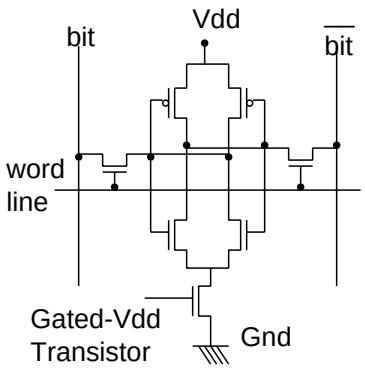

図 4 Gated-Vdd
Fig. 4 Gated-Vdd.

3. リーク電流を削減する回路技術

これまで，キャッシュにおけるスタティック消費電力削減の目的で，SRAM のリーク電流を削減するための回路的な手法が数多く提案されている．SCIMA の SCM 自体はキャッシュとハードウェアを共有していることもあり，キャッシュのリーク電流削減の回路技術がそのまま，あるいはわずかな拡張で SCM にも利用できると考えられる．

SRAM のリーク電流削減は基本的に，通常のデータの保持やデータアクセスが可能であるがリーク電流が大きい Active モードと，リーク電流は小さいがデータが保持されない，あるいはデータは保持されてもアクセスができない Sleep モードの 2 種類のモードを使い分けることで実現される．以下に，Sleep モードの実現の仕方が異なる代表的な回路技術をいくつかを紹介し，その得失利害について述べる．

Gated-Vdd

Gated-Vdd[8] は，SRAM セルの内部回路と Gnd の間に閾値の高い Gated-Vdd トランジスタを設け（**図 4**），Sleep モード時にはこれをオフにし電源供給を絶つことでリーク電流の削減を狙う．この Gated-Vdd トランジスタは同一ライン上のセルで共有され，ライン単位で Active/Sleep の制御を行う．Gated-Vdd は電源供給がオフになるためリーク電流は大きく削減できるが，Sleep モードに移行したセル内の情報は失われてしまう．

DVS

DVS（Dynamic Voltage Scaling）[9] は SRAM のセルに対して高い電圧（High）と低い電圧（Low）の 2 つの電源供給ソースを用意し，Sleep モード時に Low

電圧を供給することでリーク電流の削減を狙う．High は従来の供給電圧であるのに対し，Low はセル内の情報が維持できる最低限の供給電圧である．アクセスを行う際には電圧は High でなければならないため，Sleep モードでアクセスがあった場合は，Active モードに切り替える必要がある．したがって，DVS では Sleep モードになっていてもセル内の情報を維持できる反面，Sleep から Active に戻す際の遅延が発生する．

ABB-MTCMOS

ABB-MTCMOS[15] は，Sleep モード時にトランジスタの閾値を増加させることでリーク電流の削減を狙う．ABB-MTCMOS は DVS と同様にセル内の情報を保持できるが，回路への供給電圧が増すことによるリーク電流の増分も無視できず，他の 2 つの回路技術に比べ消費電力の削減率は小さい．また，Active/Sleep 切替え時の遅延が生じ，その際の電力的オーバヘッドが大きいという欠点を持つ．

ここにあげた代表的な回路技術のうち，最もリーク電流の削減率が高いのは Gated-Vdd である．しかし，キャッシュへの適用を考えた場合，Gated-Vdd では Sleep モードになったセルは情報を保持できないため，Sleep モードの箇所へアクセスするとキャッシュミスとなり，性能低下は免れない．一方，SCIMA では SCM 上の確実に使われない領域，すなわちセル内のデータが失われても性能的なペナルティがない領域を特定することができる．したがって，前節で紹介した回路技術のうち，リーク電流の削減率が最も良い Gated-Vdd を採用したとしても性能に影響がないと考えられる．次章以降では，SCIMA のリーク電流削減の回路技術として Gated-Vdd を採用し検討を行う．

4. SCIMA におけるスタティック消費電力削減

本章では，回路技術に Gated-Vdd を採用した SCM のリーク電流削減機構について検討し，これをソフトウェアから制御してスタティック消費電力を削減する戦略について述べる．

4.1 SCM のリーク電流削減機構

Gated-Vdd を用いて SCM におけるリーク電流を削減するために，Active/Sleep モード切替えの単位を決定する必要がある．SCM は page 単位でデータ転送の管理が行われるため，Active/Sleep 切替えも page 単位で行うのが妥当であると考えられる．page サイズでのモード切替えは**図 5** のように各ラインの Gated-Vdd トランジスタに対して，page サイズごとに制御ビットを設けることで行う．これは，もとの Gated-Vdd に対する簡単な拡張で実現可能であると考えられる．

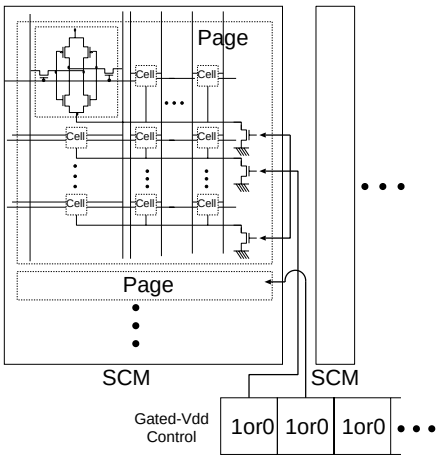

図 5 Active/Sleep 切替えの制御
Fig. 5 Active and Sleep mode control.

ソフトウェアからの各 page の Active/Sleep モードの切替えは，上述の制御ビット（制御レジスタ）に対して，対応する page のビットをセット/リセットすることで行う．これを実現するため，命令セットアーキテクチャ上に以下の命令を追加する．

activate_scm（page 番号） 引数として与えられたページ番号が指す page を Active に移行する．

deactivate_scm（page 番号） 引数として与えられたページ番号が指す page を Sleep に移行する．

4.2 ソフトウェア制御のリーク電流削減手法

SCM において page 単位で Active/Sleep を切り替えるために，page に保持されているデータが必要であるか不必要であるかを判断し，その情報をプログラム中で与えなければならない．SCM はソフトウェアにより制御されるため，この判断をソースコードレベルで行うことが可能である．

SCM と主記憶間のデータ転送のソースコード上での表現にはいくつかの方法があるが，ここでは SCIMA 用のプログラミングインタフェースとして開発している「SCIMA 用ディレクティブ[14]」を例として，これを利用した Active/Sleep 切替え戦略について検討を行う．

SCIMA ディレクティブを用いたプログラミングでは，ある配列用の SCM 領域の確保と解放のために明示的にディレクティブを挿入するため，SCM 上の領域のデータが必要か不必要かの判断はこのディレクティブに基づいて行うことが可能である．

そこで，プログラムの開始時には page をすべて

Sleep モードとし，`!$scm begin` によって SCM 領域が確保される際に，確保領域に含まれる page を Active モードとする．一方，`!$scm end` によって領域が解放される際に，その領域に含まれる page を Sleep モードとする（page 内に他の配列用に確保されている領域がある場合は Active モードを維持する）．これによって SCM 上の必要な page のみ Active モードにし，残りの領域を Sleep モードにできるためリーク電流を削減することができる．したがって，ユーザが新たに特別な命令を挿入する必要はなく，SCIMA 用ディレクティブベースコンパイラの簡単な拡張で Active/Sleep の切替えが実現可能である．なお，キャッシュ領域については常時 Active モードとする．

なお，SCIMA 向けに最適化されたプログラムでは，キャッシュ領域についてはほとんど使われない場合が多い．現在は way 単位でしかキャッシュと SCM の切替えができないため，キャッシュ領域におけるリーク電流も無視できない．なお，前述のように SCM は回路的にはキャッシュと同様であるため，SCM とキャッシュで共通の回路技術を使用することができると考えられる．したがって，SCM 領域のリーク電流削減手法と共通の回路技術に基づくキャッシュ向けの低消費電力化手法を，残されたキャッシュ領域に適用することもあわせて検討する．

性能とリーク電流のトレードオフ

SCM におけるスタティック消費エネルギー（E_{static}）は，実行時間 T と SCM 上のサイクルあたりのスタティック消費電力 P_{static} を用いて，

$$E_{static} = T \times P_{static}$$

と書き表すことができる．ここで，P_{static} は Active 領域の面積に比例するため，Active 領域を小さくすればそれにともない減少する．一方，実行時間 T は Active 領域を小さくすると増加するため，性能とリーク電流の間にはトレードオフの関係が存在する．

たとえば，再利用性のある配列を扱うプログラムの場合を考える．このようなプログラムでは，再利用性のある配列をブロッキングして SCM に載せることでその再利用性を最大限に活かすことができる．このとき，ブロッキングサイズを小さく設定し必要な SCM 領域を節約すれば P_{static} を減少させることが可能であるが，主記憶と SCM 間のデータ転送粒度が小さくなり，さらに SCM 上の要素が再利用される回数が減少するため実行時間 T は増加する．逆に，ブロッキングサイズを大きく設定すると実行時間 T の減少につながるが，SCM 上の Active 領域の面積は増えるため P_{static} は増加する．一方，再利用性はなく連続的にアクセスされる配列を扱うプログラムの場合は，SCM を用いて配列に対しストリームアクセスを行うことで主記憶アクセスのレイテンシに起因するストール時間を削減することができる．このとき，転送粒度を小さく設定すれば必要な SCM 領域も小さくなり，P_{static} は減少するが，転送回数は増加するためレイテンシ削減の効果が減少し実行時間 T は増加する．

このように，主記憶とのデータ転送の粒度は性能とスタティック消費エネルギーのトレードオフに大きく影響すると考えられる．このトレードオフを調べるために，以降ではブロックサイズや転送粒度をパラメータとして変化させて評価する．

5. 評価環境

5.1 評価対象

提案手法の有効性を検討するため，性能評価ならびにスタティック消費エネルギーの評価を行う．評価対象として，再利用性のある配列を扱うプログラムの例として行列積（倍精度 256 × 256），再利用性がなく連続的にアクセスされる配列を扱うアプリケーションとして SPEC2000 ベンチマークの 171.swim を選択した．

性能評価にはサイクルレベルシミュレータを用いた．また，スタティック消費エネルギーは，キャッシュ領域，SCM 領域のそれぞれについて，それぞれが Active または Sleep であったサイクル数に，単位サイクルあたりの消費エネルギーを積算して求めた．

評価では，以下の 3 つの比較を行った．

- 従来のキャッシュアーキテクチャ向けに最適化したコードを，従来構成のキャッシュのもとで実行した場合（Cache）
- SCIMA 向けに最適化を行ったコードを，提案手法のもとで実行した場合（SCIMA）
- Cache と共通のコードを，提案手法と共通の回路技術 Gated-Vdd を用いたキャッシュ向け低消費電力手法 CacheDecay [11] のもとで実行した場合（CacheDecay）

比較手法として，従来のキャッシュの消費電力削減手法として提案されている CacheDecay を取り上げる．提案手法と CacheDecay は回路手法 Gated-Vdd を利用するという点は共通であるが，それぞれの手法には以下の得失利害がある．

まず，提案手法はソフトウェアにより明示的に電源の制御を行うので，確実に必要な領域のみを Active にすることができる．また，再利用性の異なる配列が混在している場合も，個々の配列に対して最適化が可能

である．しかし，電源制御のタイミングがSCM領域の確保・解放に連動しており，電源制御の粒度がpage単位でしか行えないという欠点がある．一方，CacheDecayはある一定間隔（以降Decay Intervalと呼ぶ）ごとにアクセスのなかったキャッシュラインを自動的にSleepにするため命令の挿入を必要とせず，静的に挙動が解析しにくいプログラムにおいても効果を発揮する．また，電源制御の粒度はキャッシュラインであり，提案手法よりも細粒度での電源制御が可能である．しかし，CacheDecayではキャッシュの振舞いを完全に予測することができず，Decay Intervalをいかに選択してもつねに最適な性能/消費電力を得られるとは限らない．また，再利用性の異なる配列が混在している場合にも1つのDecay Intervalで対処しなければならないため，それぞれの性質に応じた最適化を行うことはできない．

また，SCIMAにおけるさらなるスタティック消費電力削減の可能性について検討するため，スカラ変数などのために残されているキャッシュ領域にCacheDecayを適用した場合（以降SCIMADecayと呼ぶ）についても評価を行った．CacheDecayにおいてもブロッキングにより実行時間の短縮を図ることは有用であると考えられるため，Cacheと共通のコードを用いて評価を行った．

5.2 評価条件

評価に共通のパラメータを**表1**に示す．それぞれに対するキャッシュとSCMの構成を**表2**に示す．キャッシュ領域は，つねに全体がActiveであるとする．ただし，CacheDecayおよびSCIMADecayのキャッシュ領域については一定サイクル（Decay Interval）アクセスがないラインは，Gated-Vddにより電源供給を絶たれSleepになる．また，Decay Intervalについてはスタティック消費エネルギー・遅延積が最小となる点を採用する．

本論文では，プログラムの実行時間をCPU-busy time（T_b），Memory-stall（T_m）の2つに分類する．CPU-busy timeはプロセッサが計算処理を行っている時間であり，Memory-stallは主記憶からのデータ転送待ちに起因するストール時間を表す．

ここで，プログラムの総実行時間をT，オフチップメモリスループットが無限大かつオフチップメモリレイテンシが0とした場合の実行時間T_pとする．このT，T_pを用い，本論文ではT_b，T_mを以下のように定義する．

$$T_b = T_p$$
$$T_m = T - T_p$$

表1 評価に用いたパラメータ
Table 1 Evaluation parameters.

キャッシュラインサイズ	32 B, 128 B
オフチップメモリスループット	4 B/cycle
オフチップメモリレイテンシ	40 cycle
page サイズ	4 KB

表2 キャッシュとSCMの構成
Table 2 On-chip memory configurations.

	キャッシュサイズ	SCM サイズ
Cache	64 KB (4-way)	0
CacheDecay	64 KB (4-way)	0
SCIMA	16 KB (1-way)	48 KB
SCIMADecay	16 KB (1-way)	48 KB

なお，本論文では，提案手法や，CacheDecayの実装のための追加回路に起因する電力的オーバヘッドは無視できるものとして評価を行う．またGated-VddでActive/Sleepを切り替える際の遅延は，主記憶からの転送待ちの時間に完全に隠蔽され，性能への影響はないものとする．

6. 評価結果

6.1 性能

性能に関する評価結果を**図7**，**図10**に示す．図中，SCIMAの評価結果のBLの値はブロックサイズ☆を表す．

図のグラフは各ラインサイズのCacheの値を基準とした相対的な実行時間を示しており，プロセッサが実行を行っていた時間（CPU busy time）と主記憶からの転送待ちによりストールした時間（Memory Stall）の内訳を示している．CacheとCacheDecayにおけるブロッキングサイズは，いくつかの実験から実行時間が最も短くなるもの（BL=32）を選択している．

一方，比較手法であるCacheDecayでは，性能と消費電力はトレードオフの関係にあり，性能を大幅な低下を許容すれば高い消費電力削減効果を引き出すことが可能である．しかし，性能の大幅な低下を許容しスタティック消費電力の削減を達成することは本論文の目的とは異なるため，Decay Intervalはいくつかの候補からスタティック消費エネルギー・遅延積が最小となる点を選択した（**図6**）．

図7，図10よりCacheDecayでは，Cacheと比較してMemory Stallが増加し性能が大きく低下していることが分かる．これはCacheDecayにより将来的に

☆ 行列積ではブロックキングにおける一辺の要素数を表す．171.swimでは，配列を大粒度転送する際に1つの配列に確保されるSCM容量を表している．

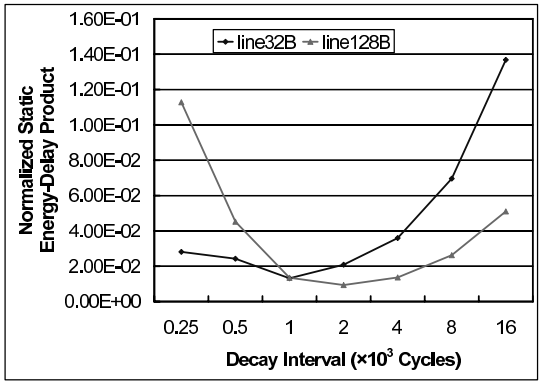

図 6 Decay Interval の選択（171.swim）
Fig. 6 Decay Interval exploration (171.swim).

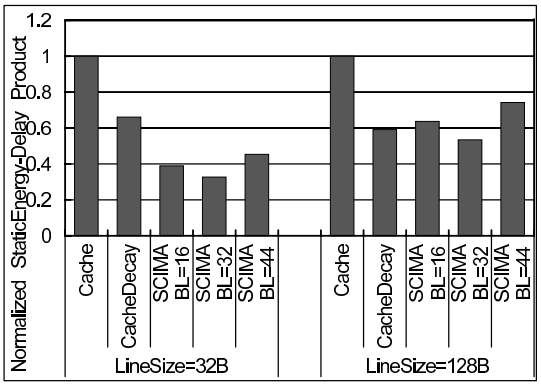

図 9 行列積のスタティック消費エネルギー・遅延積
Fig. 9 Normalized static energy-delay product (Matrix Multiplication).

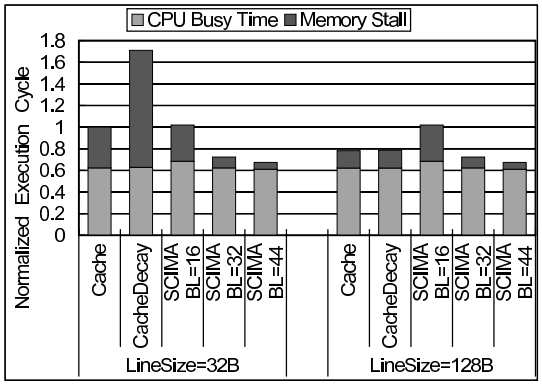

図 7 行列積の性能評価
Fig. 7 Normalized execution cycles (Matrix Multiplication).

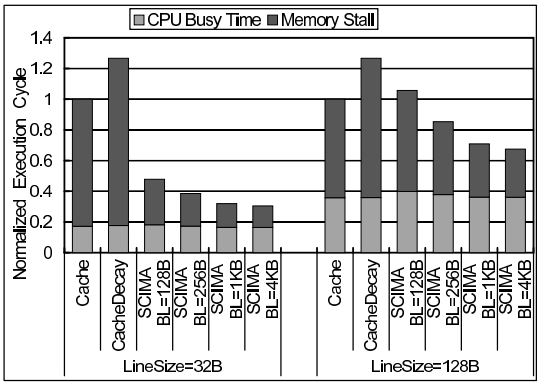

図 10 171.swim の性能評価
Fig. 10 Normalized execution cycles (171.swim).

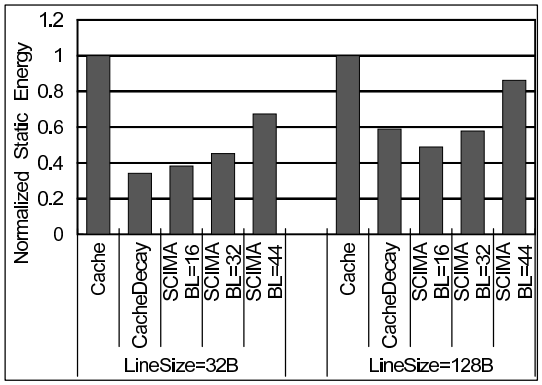

図 8 行列積のスタティック消費エネルギー評価
Fig. 8 Normalized static energy (Matrix Multiplication).

アクセスされるはずのラインが Sleep モードに移行されてしまうことで，必要な情報が失われ，キャッシュミスが増大してしまったためである．

一方，SCIMA では Cache に比べて，行列積と 171.swim ともに大きなブロックサイズの場合は Memory Stall の削減により高い性能が得られている．これは SCM を用いることによりデータの再利用性が最大限に活用できたこと，大粒度転送により主記憶アクセスのレイテンシを削減できたことに起因する．両プログラムともブロックサイズに比例して性能は向上しているが，性能向上率はブロックサイズを大きくしていくにつれて減少している．

6.2 スタティック消費エネルギー

次に，スタティック消費エネルギーの評価結果を**図 8**，**図 11** に示す．各々は各ラインサイズにおける Cache のスタティック消費エネルギーに対する相対値を示している．まず，Cache と CacheDecay を比較すると，CacheDecay では Cache に比べてスタティック消費電力が大きく削減されていることが分かる．図 8 と図 11 を比較すると，特に後者において CacheDecay による効果が顕著であるが，これは 171.swim ではアクセスされる配列の多くは再利用性がなく，キャッシュ上か

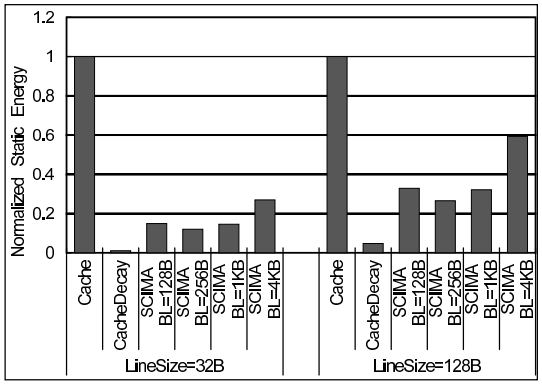

図 11 171.swim のスタティック消費エネルギー評価
Fig. 11 Normalized static energy (171.swim).

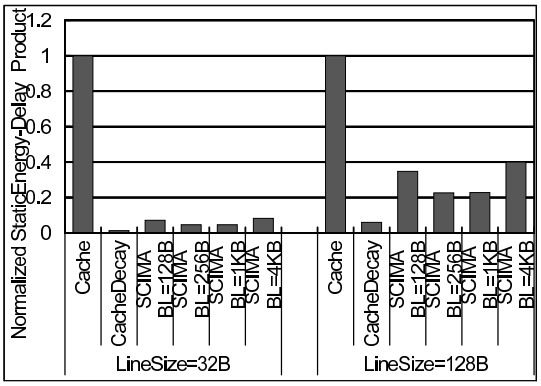

図 12 171.swim のスタティック消費エネルギー・遅延積
Fig. 12 Normalized static energy-delay product (171.swim).

ら消去されてもほとんど性能に影響がないためと考えられる．

次に，SCIMA と Cache を比較すると，SCIMA ではいずれのブロックサイズでもスタティック消費電力が削減されている．スタティック消費電力の削減には，実行時間が短縮されたことによる効果と，使用されない SCM 領域を Sleep モードにしたことの両方の効果が含まれている．ブロックサイズを小さくとると実行時間は増加するが，それ以上に Active な SCM 領域を縮小できる効果が大きいためにスタティック消費電力が削減されている．特に行列積ではブロックサイズが小さいほど削減率は高くなっている．一方，行列積の BL=44 の場合はすべての SCM 領域が Active であるため，実行時間短縮の効果のみでスタティック消費電力が削減されている．

CacheDecay と SCIMA を比較すると，行列積では CacheDecay に比べて SCIMA の方がスタティック消費電力が削減されている場合もある（図 8）のに対し，171.swim ではいずれのブロックサイズと比較しても CacheDecay が最もスタティック消費エネルギーが少ない結果となっている（図 11）．これは，後者に多く見られる再利用性のない配列へのアクセスでは，CacheDecay で短い DecayInterval を設定し，アクセスのない領域をライン単位で電源制御を行うほうが，提案手法のように SCM 領域確保・解放と同時に数 KB 単位で領域の Active/Sleep を切り替えるよりもより効率的であったためと考えられる．

6.3 消費エネルギー・遅延積

図 9，**図 12** に各ラインサイズの Cache のスタティック消費エネルギーと実行時間との積を基準とした場合のスタティック消費エネルギー・遅延積を示す．

まず，Cache と SCIMA を比較すると，SCIMA ではブロックサイズによらずスタティック消費エネルギー・遅延積が大幅に改善されている．また，図 9 において SCIMA の各ブロックサイズにおける値を比較すると，ブロックサイズ 32 で最小値をとり，以降増加している．同様に図 12 においても各ブロックサイズにおける値を比較すると，ブロックサイズ 256B で最小値をとり，以降増加している．SCM 領域におけるスタティック消費エネルギーは，Active 領域のサイズと Active 状態であった時間の積で求められるが，ブロックサイズが大きい場合は主に性能改善の効果が，小さい場合は主に SCM 領域を Sleep モードにすることの効果が大きいためと考えられる．

次に，SCIMA と CacheDecay を比較すると，171.swim では CacheDecay は SCIMA よりもスタティック消費エネルギー・遅延積において優れた結果となっている．しかし，図 7，図 10 に見られるとおり，CacheDecay では実行時間が長くなっている．CacheDecay により削減されるのはプロセッサ全体の消費電力のうち，キャッシュ部分のみであるため，キャッシュ以外の部分における消費電力の割合によっては SCIMA の方が優れた結果になると考えられる．

以上より，ブロックサイズによって実行時間とスタティック消費電力のトレードオフはあるものの，提案手法を用いることでソフトウェア制御オンチップメモリのスタティック消費電力を効率的に削減できると考えられる．

6.4 SCIMA におけるキャッシュ領域の低消費電力化

SCIMA 向けに最適化されたコードでは，配列に対して SCM を割り当てるためキャッシュはほとんど使われていない．**図 13** は，171.swim において，SCIMA と SCIMA のキャッシュ領域にキャッシュ向けの低消費電

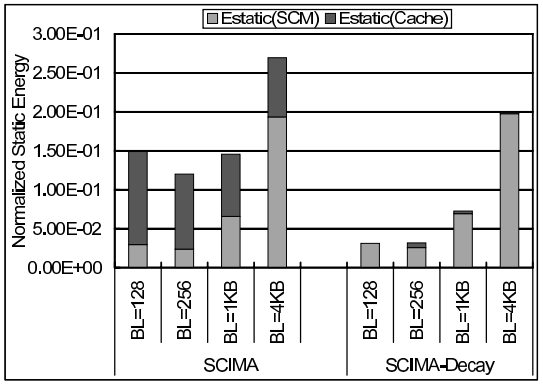

図 13 SCIMA におけるキャッシュ部分の低消費電力化
Fig. 13 Energy reduction for cache part in SCIMA.

力化手法である CacheDecay を適用した SCIMADecay について，SCM 領域におけるスタティック消費電力（$E_{static}(SCM)$），キャッシュ領域におけるスタティック消費電力（$E_{static}(Cache)$）の割合を示したものである．各値は Cache に対する相対値である．SCIMA においても，キャッシュ領域に対し既存の低消費電力化手法を適用することでさらなるリーク電流の削減を図ることは有用であると考えられる．

7. ま と め

本論文では，SCIMA のソフトウェア制御オンチップメモリ SCM によるスタティック消費電力の削減手法を提案した．ソフトウェアによりデータが制御される SCM では将来アクセスされる SCM 領域と，アクセスされない SCM 領域をプログラムの情報から判別することができる．この特徴を利用して，アクセスされない領域を Gated-Vdd により Sleep モードとすることで，効率的なリーク電流削減を狙う．

提案手法をサイクルレベルシミュレータを用いて評価した結果，SCIMA では利用する SCM のサイズによって性能と消費電力のトレードオフはあるものの，従来のキャッシュに比べて，スタティック消費エネルギーを削減できることが分かった．

今後の課題としては，このトレードオフを考慮しつつ，ダイナミック消費電力を含めた電流削減のための SCIMA 用最適化の指針を検討することがあげられる．また，今回評価に用いたプログラムは，再利用性がある配列またはない配列のみのどちらか片方の性質を示す配列のみを含んでいる．このようなプログラムでは，Decay Interval の選択次第で CacheDecay でも対応可能であるが，異なる再利用性を示す配列が混在しているプログラムに対しては，単一の Decay Interval で両方の配列に対応しなければならず，最適化が困難であると考えられる．一方，提案手法では各々の配列に対して使用する SCM サイズや転送ブロックサイズを明示的に指定することで，それぞれの配列を再利用性に応じて個別に最適化が可能である．この利点を明らかにするためにも，今後より多くのアプリケーションを用いて評価を行う予定である．

謝辞 本研究の一部は，文部科学省科学研究費補助金（基盤研究（B）No.14380136）によるものである．

参 考 文 献

1) Brooks, D. and Martonosi, M.: Value-based clock gating and operation packing: dynamic strategies for improving processor power and performance, *ACM Trans. Comput. Syst.* (*TOCS*), Vol.18, No.Issue 2 (2000).
2) Li, H., Bhunia, S., Chen, Y., Vijaykumar, T. and Roy, K.: Determistic Clock Gating for Microprocessor Power Reduction, *Proc. 9th HPCA*, pp.113–123 (2003).
3) Villa, L., Zhang, M. and Asanovic, K.: Dynamic zero compression for cache energy reducion, *Proc. 33rd MICRO*, pp.214–220 (2000).
4) Thompson, S., Packan, P. and Bohr, M.: MOS Scaling: Transistor Challenges for the 21st Century, *Intel Technology Journal* (Q3 1998).
5) Butts, J. and Sohi, G.: A static power model for architects, *Proc. 33rd annual ACM/IEEE International symposium on Microarchitecture*, pp.191–201 (2000).
6) De, V. and Borkar, S.: Technology and Design Challenges for Low Power and High Performance, *Proc. IELPED'99*, pp.163–169 (1999).
7) Keshavarzi, A., Roy, K. and Hawkins, C.: Intrinsic Leakage in Low Power Deep Submicron CMOS ICs, *Proc. ITC'97*, pp.146–155 (1997).
8) Powell, M., Yang, S., Falsafi, B., Roy, K. and Vijaykumar, T.: Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories, *Proc. ISLPED'00*, pp.90–95 (2000).
9) Flautner, K., Kim, N., Martin, S., Blaauw, D. and Mudge, T.: Drowsy Caches: Simple Techniques for Reducing Leakage Power, *Proc. 29th ISCA*, pp.148–157 (2002).
10) Kim, N., Flautner, K., Blaauw, D. and Mudge, T.: Drowsy instruction caches: leakage power reduction using dynamic voltage scaling and cache sub-bank prediction, *Proc. 35th MICRO* (2002).
11) Kaxiras, S., Hu, Z. and Martonosi, M.: Cache decay: Exploiting generational behavior to re-

duce cache leakage power, *Proc. 28th ISCA* (2001).
12) 近藤正章，中村　宏，朴　泰祐：SCIMA における性能最適化手法の検討，情報処理学会論文誌，Vol.42, No.SIG12(HPS4), pp.37–48 (2001).
13) 近藤正章，田中慎一，中村　宏：ソフトウェア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，No.ARC-149, pp.1–6 (2002).
14) 藤田元信，近藤正章，中村　宏，千葉　滋，佐藤三久：ソフトウエア制御オンチップメモリのための最適化コンパイラの構想，情報処理学会研究報告，No.ARC-146, pp.31–36 (2002).
15) Nii, K., Makino, H., Tujihashi, Y., Morishima, C., Hayakawa, Y., Nunogami, H., Arakawa, T. and Hamano, H.: A low power SRAM using auto-backgate-controlled MT-CMOS, *Proc. ISLPED'98*, pp.293–298 (1998).

(平成 16 年 1 月 31 日受付)
(平成 16 年 5 月 9 日採録)

藤田　元信（学生会員）

平成 13 年東京大学工学部計数工学科卒業．平成 15 年同大学大学院情報理工学系研究科修士課程修了．現在，同博士課程に在籍中．高速・低消費電力プロセッサアーキテクチャ向け最適化コンパイラの研究に従事．

田中　慎一

平成 14 年東京大学工学部計数工学科卒業．平成 16 年同大学大学院情報理工学系研究科修士課程修了．現在 NTT データ（株）に勤務．

近藤　正章（正会員）

平成 10 年筑波大学第三学群情報学類卒業．平成 12 年同大学大学院工学研究科博士前期課程修了．平成 15 年東京大学大学院工学系研究科先端学際工学専攻修了．工学博士．独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST 研究員を経て，現在東京大学先端科学技術研究センター特任助手．計算機アーキテクチャ，ハイパフォーマンスコンピューティング，ディペンダブルコンピューティングの研究に従事．電子情報通信学会，IEEE 各会員．

中村　　宏（正会員）

昭 60 年東京大学工学部電子工学科卒業．平成 2 年同大学大学院工学系研究科電気工学専攻博士課程修了．工学博士．同年筑波大学電子・情報工学系助手．同講師，同助教授を経て，平成 8 年より東京大学先端科学技術研究センター助教授．この間，平成 8 年〜9 年カリフォルニア大学アーバイン校客員助教授．高性能・低消費電力プロセッサのアーキテクチャ，ハイパフォーマンスコンピューティング，ディペンダブルコンピューティング，ディジタルシステムの設計支援の研究に従事．情報処理学会より論文賞（平成 5 年度），山下記念研究賞（平成 6 年度），坂井記念特別賞（平成 13 年度），各受賞．IEICE，IEEE，ACM 各会員．