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

2003－ARC－155　(14)
2003／11／28

ソフトウェア制御オンチップメモリを用いた静的消費電力削減に関する検討

田中 慎一[†]　近藤 正章[††,†]
藤田 元信[†,††]　中村 宏[†]

今後，プロセスの微細化に伴いリーク電流が増大し，プロセッサ全体の消費電力に占める静的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静的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なわれている．

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．本稿では，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．

Leakage Power Reduction for Software Controlled On-Chip Memory

Shinichi TANAKA,[†] Masaaki KONDO,[††,†]
Motonobu FUJITA[†,††] and Hiroshi NAKAMURA[†]

As semiconductor technology scales down, the static power due to leakage current becomes dominant in the total power dissipation of microprocessors. 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由来する動的消費電力とリーク電流に由来する静的消費電力からなる．従来は動的消費電力がプロセッサにおける消費電力の大半を占めており，従来研究の多くは動的消費電力削減を目的としていた．しかし，今後は静的消費電力が消費電力の大きな割合を占めるようになると予測されている．これは半導体技術の微細化に伴い，トランジスタが導通し始める閾値電圧が低下し，リーク電流が増大するためである．

特に近年の高性能プロセッサにおいては，キャッシュメモリにおける静的消費電力が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多くの面積を占めており，それに比例して静的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いる．

その回路技術として，これまでに 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[6] や，電源電圧を制御してリーク電流の削減を狙う DVS[7] 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

† 東京大学 先端科学技術研究センター
Research Center for Advanced Science and Technology (RCAST), The University of Tokyo
†† 独立行政法人 科学技術振興機構 JST

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 (キャッシュヒットタイム) 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，性能への影響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多く提案されている．

我々は，従来のキャッシュに加えソフトウェア制御可能なオンチップメモリ (*SCM: Software Controlled Memory*) をチップ上に搭載する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[1]．SCIMA は SCM を用いて主記憶との間で柔軟なデータ転送を行なうことで，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できるため，SCM 上へ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上で指定することが可能となる．

この特徴から，SCIMA は Gated-Vdd など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．本稿では，主に Gated-Vdd を利用した SCIMA による静的消費電力削減について検討，評価を行なう．

2. SCIMA

2.1 SCIMA のアーキテクチャ

SCIMA の構成を図 1 に示す．従来のキャッシュに加えアドレス指定可能な Software Controlled Memory(SCM) をチップ上に追加する．

従来のキャッシュではデータのアロケーション・リプレースメントがハードウェアにより自動的に制御されるのに対し，SCM ではそれらをソフトウェアで明示的に制御可能である．SCIMA では SCM を利用することで従来のキャッシュで発生していた競合に起因するオフチップトラフィックの増加を抑えることができ，動的消費エネルギーの削減にも有効であることがわかっている[2]．

SCM 領域は page と呼ばれる単位☆に分割・管理されており，メモリアクセス順序保証もこれを単位として行われる．

また，キャッシュと SCM はハードウェアとしての SRAM は共有するが，その構成比をキャッシュのウェイ単位としてアプリケーション性質に応じて変更する手法も提案されている[1]．

2.2 SCM と主記憶のデータ転送

SCIMA では SCM とオフチップメモリ間のデータ転送を制御するための専用の page-load/page-store と呼ばれる命令を備える．本命令は，データ転送元の開始アドレス・データ転送先の開始アドレス・総転送サイズ・データ転送単位・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するほか，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もサポートしている．

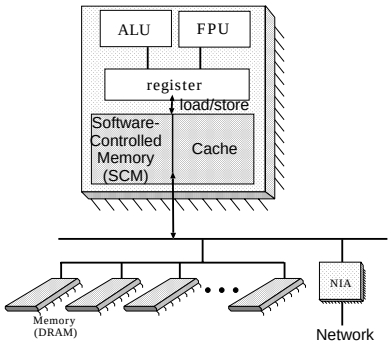

図 1 SCIMA の構成

また，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されている[3]．SCIMA 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 SCIMA 向けの最適化が可能である．

3. リーク電流削減の回路技術

これまで，キャッシュにおける静的消費電力削減の目的で SRAM のリーク電流を削減する回路技術がいくつか提案されている．SCIMA の SCM 自身はキャッシュと同様のハードウェアを用いるため，キャッシュのリーク電流削減の回路的技術がそのまま，あるいはわずかな拡張で SCM に利用できると考えられる．本節では代表的な回路技術をいくつか紹介し，その得失利害について述べる．

SRAM のリーク電流削減は，基本的に通常のデータ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，データが保持されないか，またはデータは保持されてもアクセスができないかわりにリーク電流が小さい *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いくつかの手法を述べる．

Gated-Vdd

Gated-Vdd[6] は，図 2 に示すように，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，Sleep モード時にはこれをオフにすることで電源供給を絶ち，リーク電流を削減している．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 モードの制御を行なう．

☆ 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．

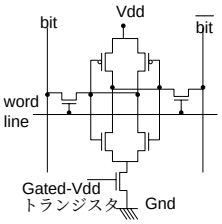

図 2 Gated-Vdd

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 時にはセル内の情報が失われ，キャッシュミスが増加するための性能のペナルティがあるという欠点を持つ．

DVS

DVS (Dynamic Voltage Scaling)[7] 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．アクセスを行なう際には High 電圧で行なう必要があり，Sleep モード時に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[8] 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削減を狙う．ABB-MTCOMS は DVS と同様にセル内の情報を保持できるが他の手法に比べ消費電力の削減率が小さく，Acitve/Sleep 切り替え時の遅延が生じ，またその際の電力的オーバーヘッドが大きいという欠点を持つ．

4. SCIMA による静的消費電力の削減

4.1 概　　要

本稿では，SCIMA の SCM の静的消費電力削減手法を提案する．SCM はプログラムによりデータのアロケーションやリプレースメントを行なう．そのため，必要なデータと不必要な (将来的にもう読み込む出すことがない) データが存在する SCM 領域を特定することが可能である．したがって，前節で述べた回路技術を用い，必要ない SCM 領域を Sleep モードにすることで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．

前述のように，SCM は回路的にはキャッシュと同様であり，前節で述べた回路技術のどれも適用することが可能である．ここで，SCIMA の場合，確実に使われない，すなわち Gated-Vdd により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．Gated-Vdd はリーク電流の削減率が最も良いこともあり，本稿では，SCIMA のリーク電流削減の回路技術として Gated-Vdd を採用し検討を行なう．

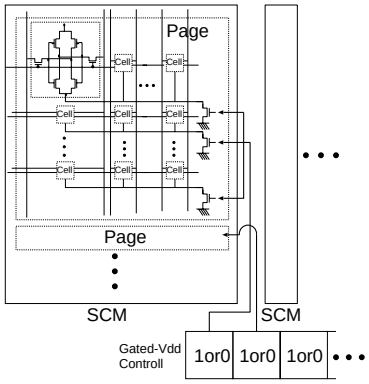

図 3 Active/Sleep 切り替えの制御

4.2 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 モード切り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なわれるため，Active/Sleep 切り替えも page 単位で行なうのが妥当であると考えられる．page サイズでのモード切り替えは図 3 のように各ラインの Gated-Vdd トランジスタに対して，page サイズ毎に制御ビットを設けることで行なう．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．

ソフトウェアからの各 page の Active/Sleep モードの切り替えは，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なう．

4.3 ソフトウェア制御によるリーク電流削減の戦略

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．ここで，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[3]」による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なうことが可能である．

図 4 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`!$scm begin` ディレクティブは，引数に配列名，配列の各次元毎のサイズ，ストライド幅をとる．プログラム実行時には，指定した配列用の SCM 領域が`!$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`

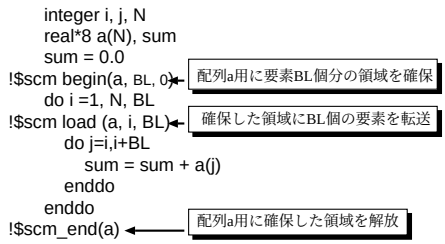

図 4 ソースコードにディレクティブを挿入

は引数で配列名，転送の起点となる要素，各次元毎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．したがって，`!$scm begin`によって確保された領域のみが必要なデータを保持しており，そのデータが必要であるのは対応する`!$scm end`までということになる．

そこで，プログラムの開始時にはpageを全て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の切り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．

このように，SCIMAディレクティブによってSCM領域のActive/Sleepの切り替えを制御することにより，性能に加え静的消費エネルギーを考慮してのSCMの利用最適化が可能となっている．

再利用性のある配列を扱うプログラムの場合，SCMを用いてブロッキングした要素の再利用性を最大限に活かすことができる．この時ブロッキングサイズを縮小し必要なSCM領域を節約することによってリーク電流削減効率を向上させることが可能であるが，ブロッキングされた要素の再利用性が減少し性能は低下する．反対にブロッキングサイズを拡大するとリーク電流の削減効率は減るが性能は向上する．

また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，SCMを用いて配列を大粒度で転送することにより主記憶アクセスのレイテンシを削減することができる．1つの配列につき転送粒度サイズのSCMが必要となるので，転送粒度を縮小することでリーク電流削減効率を向上させることができるが，レイテンシ削減の効果が減少し性能は低下する．反対に転送粒度を拡大するとリーク電流の削減効率は減るが性能は向上する．

表 1 メモリの仮定条件

キャッシュ/SCM	サイズ 64KB ラインサイズ 32B or 128B 連想度 4Way
Bus バンド幅	4Byte/cycle
主記憶アクセスレイテンシ	40cycle
page サイズ	4KB

このようにブロックサイズや転送粒度のサイズを変えることによって性能と静的消費エネルギーを変えることができる．しかし性能と静的消費エネルギーはトレードオフの関係にあるため，性能への影響を抑えつつ消費電力を削減できるブロックサイズや転送粒度のサイズを選ぶ指針を検討する必要がある．しかし，本稿ではその指針は未検討であるため，ブロックサイズや転送粒度を様々に変えて評価を行っている．

5. 性能・消費エネルギー評価

5.1 評価環境

本稿では，提案するリーク電流削減手法による性能への影響と，静的消費電力削減の効果を評価する．評価にはSCIMA評価用のシミュレータを用いる．静的消費エネルギーは，シミュレータより得られるActive/Sleepモードである領域のサイズやそのモードで動作していた時間などの情報から求める．

評価に用いるプログラムとしては，再利用性のある配列を扱うアプリケーションとして行列積，また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．

評価では，提案するリーク電流削減手法を適用したSCIMAと従来のキャッシュアーキテクチャ(Cacheと表記)を比較評価する．また，キャッシュにおけるリーク電流削減手法として，CacheDecay[5]手法を用いたキャッシュも比較対象とする．

5.2 評価の仮定条件

表1に，本稿を通して用いる共通のメモリの仮定条件を示す．

本評価では，提案するSCIMAのリーク電流削減手法や，CacheDecayの実装のための追加回路などの電力的オーバーヘッドは無視できるものとして評価を行なう．またGated-VddでActive/Sleepを切り替える際の遅延については，オフチップアクセスのレイテンシで隠蔽であり，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5，図8に示す．図中，SCIMAの評価結果のBLの値はブロックサイズ☆を

☆ 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.

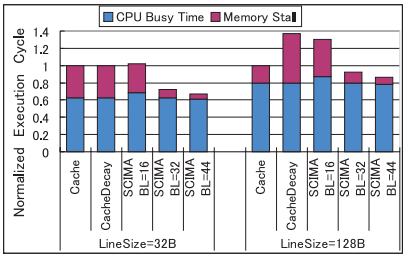

図 5 行列積の性能評価

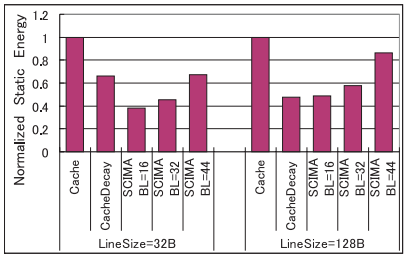

図 6 行列積の静的消費エネルギー評価

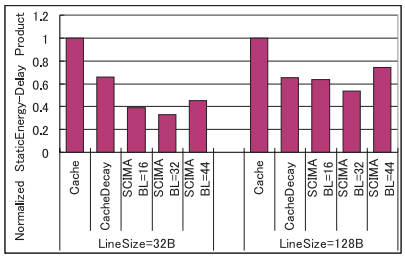

図 7 行列積の静的消費エネルギー・遅延積

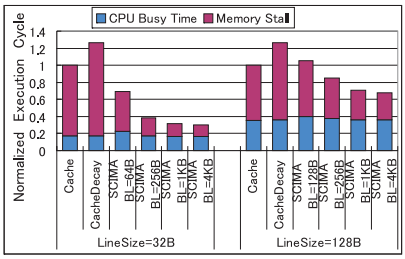

図 8 171.swim の性能評価

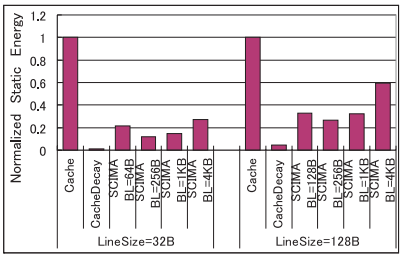

図 9 171.swim の静的消費エネルギー評価

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なっていた時間 (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最適なものを選択している．また CacheDecay において，どれだけの期間アクセスがなかったらラインを Sleep モードに移行するかの間隔 (Decay Inteval) は，性能低下が 1.5 倍以内に収まる範囲で最大限の消費電力削減効果が得られる点を選択した．

図より通常のキャッシュである Cache と CacheDecay 手法の性能を比較すると，CacheDecay では Memory Stall が増加し性能が大きく低下している．これは CacheDecay により将来的にアクセスされるはずのラインも，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 共に大きなブロックサイズの場合は Memory Stall の削減により高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，または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が原因である．ただしブロックサイズを小さくした場合は，行列積では再利用性が小さくなる，また 171.swim では大粒度転送の効果がなくなり，page-load/page-store 命令の増加によるオーバヘッドなどの理由から，Cache に比べてわずかに性能が低下している．

6.2 静的消費エネルギー

次に，静的消費エネルギーの評価結果を図 6，図 9 に示す．図は各ラインサイズの Cache の静的消費エネルギーを基準とした相対的な消費エネルギーを示している．

まず，Cache と CacheDecay を比較すると，CacheDecay では静的消費電力が大きく削減されている．実際に，将来的にアクセスされないと予測されるラインを Sleep モードにすることで，リーク電流削減の効果があるためである．

次に Cache と SCIMA を比較すると，SCIMA は従来のキャッシュに比べ，どのブロックサイズでも静的消費電力が削減されている．静的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者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 領域を節約できる効果が大きいために静的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．また行列積の BL=44 の

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．

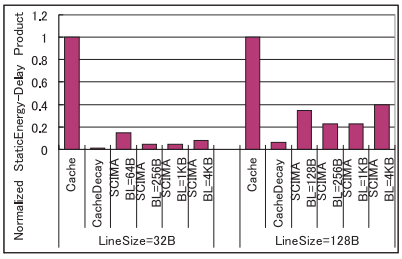

図 10 171.swim の静的消費エネルギー・遅延積

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静的消費電力が削減されている．

CacheDecay と SCIMA を比較すると，ほとんどの場合で CacheDecay が最も静的消費エネルギーが少ない結果となっている．CacheDecay では SCIMA と比べ Sleep となっているラインの割合が非常に多いため，実行時間増加の効果を打ち消して静的消費エネルギーの削減につながっている．

6.3 消費エネルギー・遅延積

図 7，図 10 に各ラインサイズの Cache の静的消費エネルギーと実行時間との積を基準とした場合の静的消費エネルギー・遅延積を示す．

図より SCIMA では Cache や CacheDecay に比べて静的消費エネルギー・遅延積をほぼ改善できている．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である．

したがって，提案する SCM の静的消費電力削減手法を用いることで，ブロックサイズによって性能の効果と SCM 領域を Sleep モードにすることの効果のどちらが大きいかは異なるが，SCIMA ではキャッシュに比べ静的消費電力を効率的に削減できると結論付けることができる．

7. 関 連 研 究

CacheDecay[5] 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[6] 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) を減らすことで静的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[7][4] 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一斉に 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．

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恐れがある．これに対し，提案手法はソフトウェアによる制御であり，確実に必要ない SCM 領域のみを Sleep モードにできるため，より効率的に静的消費電力を削減できる．

8. ま と め

本稿では，SCIMA のソフトウェア制御オンチップメモリ SCM による静的消費電力の削減手法について提案した．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ションにより評価した結果，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，静的消費エネルギーを削減できることがわかった．今後の課題としては，リーク電流削減のための SCIMA 用最適化の指針を検討し，また多くのアプリケーションにおいて性能と消費電力の評価を行なうことが挙げられる．

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) 近藤 正章ほか, “SCIMA における性能最適化手法の検討,” 情報学会論文誌, Vol 42, No. SIG 12(HPS4), pp.37-48, 2001 年 11 月.
2) 近藤 正章ほか, “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,” 情処研報, ARC-149, pp.1-6, 2002 年 8 月.
3) 藤田 元信ほか, “ソフトウェ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,” 情処研報, ARC-146, pp.31-36, 2001 年 2 月.
4) N. Kim, et al., “Drowsy instruction caches: leakage power reduction using dynamic voltage scaling and cache sub-bank prediction,” Proc. of 35th MICRO, Nov. 2002.
5) S. Kaxiras, et al., “Cache decay: Exploiting generational behavior to reduce cache leakage power,” Proc. of 28th ISCA, July 2001.
6) M. Powell, et al. “Gated-Vdd: A circuit technique to reduce leakage in deep-submicron cache memories,” Proc. of ISLPED’00, pp. 90-95, Aug. 2000.
7) K. Flautner, et al., “Drowsy Caches: Simple Techniques for Reducing Leakage Power,” Proc. of 29th ISCA, pp.148-157, June 2002.
8) K. Nii, et al., “A low power SRAM using auto-backgate-controlled MT-CMOS,” Proc. of ISLPED’98, pp.293-298, Aug. 1998.