オンチップメモリの高速化と低スタンバイリークを実現する閾値電圧の静的スケジューリング手法

石原 亨† 浅田邦博†

メモリの閾値電圧を部分的にかつ動的に変更することによりメモリの平均アクセス時間をほとんど増加させることなくリーク電流を大幅に削減する手法を提案する．オンチップメモリのアレイ部分をいくつかのブロックに分割し，少数のブロックのみを低い閾値電圧で動作させることにより高速アクセスかつ低リーク電流を可能にする．本稿では，メモリのアクセス履歴情報から将来のメモリアクセスを予測し，各メモリブロックに対する閾値電圧のスケジュールを静的に決定する手法を提案する．閾値電圧のスケジュールを決定する際にはリーク電流による消費電力だけでなくバックゲートバイアスの変更にともなう消費電力および遅延時間も考慮する．いくつかのベンチマークプログラムを使用した実験により，各メモリブロックに対して閾値電圧を静的に割り当てる手法や，履歴情報を使用しない動的割り当て手法と比較し，提案手法がパフォーマンスおよび消費電力の点で優れていることを示す．

A Threshold Voltage Scheduling Technique for High Performance and Low Leakage On-chip Memory

Tohru Ishihara† and Kunihiro Asada†

A threshold voltage scheduling technique for a high performance and low leakage on-chip memory is proposed in this paper. The basic idea of our approach is to partition a memory into several blocks and to assign a low threshold voltage (V_{th}) to a small number of the blocks. Frequently accessed blocks are activated and others are put to sleep by controlling the back-gate bias of the memory cells. Since access time to the slept blocks is larger than that to the activated blocks, predicting a block which will be accessed in future cycles is important. A main contribution of this work is to propose an integer linear programming based optimization technique and algorithm to identify the blocks which should be activated. Experimental results demonstrated that the leakage energy dissipation in cache memories optimized by our approach is reduced by 90% with negligible performance degradation.

1. はじめに

半導体製造技術の進歩にともない大容量のメモリとCPUを混載できる時代となった．システム構成によってはメモリの占有面積がチップの大部分を占めるものもあり，メモリの消費電力が無視できなくなってきている．すでに商品化されているチップを例にとると，今日のマイクロプロセッサの多くは，キャッシュメモリを含むプログラムメモリが最も電力を消費する部分回路となっている．DEC社のアルファチップにおいてはオンチップキャッシュメモリがチップ全体の25%の電力を消費する．低電力に特化されたStrongARM SA-110プロセッサにおいてもオンチップ命令キャッシュメモリがチップ全体の27%の電力を消費する[1)]．メモリの消費電力を削減することは，今後のシステムLSI設計において非常に重要な課題である．CPUなどのCMOS論理回路においては，従来，回路中の容量性負荷を充放電するエネルギーが全消費エネルギーの大部分を占めていたため，多くのエネルギー削減手法は低電圧化にその大部分を頼ってきた．しかし，回路寸法が縮小され電源電圧も比例縮小すると，負荷容量を充放電するエネルギー（ダイナミックなエネルギー）の割合は急速に低下してきた．特にメモリなどの稼働率の小さい回路では，回路が動作していないときに消費するエネルギー（スタティックなエネルギー）の割合が急速に増大している．また，低電圧化によって低下したスイッチング速度を回復させるために低い閾値電圧を使用すると，スタティックなエネルギーの割合は指数関数的に増加する．文献2)の中でBorkerは，単

† 東京大学大規模集積システム設計教育研究センター
VLSI Design and Education Center, The University of Tokyo

位ゲート幅あたりのリーク電流は世代ごとに5倍に増加すると予想している．単位面積あたりのダイナミックなエネルギーがそれほど変らないとすると，加工寸法が 0.10 μm 未満になるころにはスタティックなエネルギーがダイナミックなエネルギーを上回る可能性が出てきた．スタティックなエネルギーの原因となっているリーク電流を削減するためにこれまでに多くの回路設計手法が提案されている．MT-CMOS を使用した手法[3]はシステムの停止時にトランジスタのリーク電流を遮断するが，カットオフ時にメモリに記憶されたデータが破壊されてしまうという問題が生じる．また，Variable threshold CMOS（VT-CMOS）を使用した手法[4]や，Auto backgate controlled MTCMOS（ABC-MOS）を用いた手法[5]は基板バイアス効果を利用してスタンバイ時のリーク電流を削減することができる．しかし，これらの手法をそのまま適用するとスイッチング速度の低下を招くため，スタンバイ状態にすべき回路ブロックを効率良く決定する手法が重要となる．スタティックメモリ（以下 SRAM）のリーク電流を削減する手法もすでにいくつか提案されている[6)~8)]．これらの手法は，アクセス頻度が低いことが予測されるメモリブロックの電源供給を遮断することによりキャッシュヒット率の低下を最小限に抑えてリーク電流を削減する．これらの手法では，メモリブロックの電源を遮断する前にデータをメインメモリに退避する必要があるためパフォーマンスの低下を招く可能性がある．本稿で提案する手法は，メモリをいくつかのブロックに分割し，そのうち少数のメモリブロックのみを低閾値動作させることによりリーク電流を大幅に削減することができる．閾値を変更するための時間的ペナルティを考慮すると，次にアクセスされるブロックを適切に予測する必要がある．本稿では，過去の履歴情報から次にアクセスされるブロックを予測する機構についても提案する．

2章では，SRAM のリーク電流を削減する手法の基本アイデアを述べ，3章ではアクセス時間の制約条件下でメモリのリーク電流を最小化する問題を整数線形計画問題として定式化する．4章では実験結果を述べ5章で本稿をまとめる．

2. 基本アイデア

2.1 電圧のスケーリングと電力/遅延モデル

CMOS 論理回路のダイナミックなエネルギー消費は，式 (2) に示すとおり電源電圧の二乗にほぼ比例するため，電源電圧の削減は CMOS 論理回路においてダイナミックなエネルギー消費を削減するための最も有効な手法である．

$$E_{total} = E_{active} + E_{stand-by} \quad (1)$$

$$E_{active} \propto V_{dd}^2 \quad (2)$$

ここで，E_{active} は容量性負荷を充放電する際に消費されるエネルギー，$E_{stand-by}$ はスタンバイ時のリーク電流によるスタティックなエネルギー消費，V_{dd} は電源電圧を表す[9]．低電圧化は電力削減に大きく貢献する反面，式 (3) に示すように電源電圧の削減は速度性能の低下を引き起こす原因となる．

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \quad (3)$$

t_{pd} は伝播遅延時間，V_{th} はデバイスの閾値電圧を表す．α はキャリアの速度飽和を表すパラメータで，近年の MOSFET ではおよそ $\alpha = 1.3$ である．

式 (3) から分かるように低い閾値電圧を使用することにより回路遅延（t_{pd}）が改善される．ところが閾値電圧の削減はスタンバイ電流を急激に増大させる原因となる．スタンバイ電流によるエネルギー消費は式 (4) によって表すことができる．

$$E_{stand-by} \propto 10^{-\frac{V_{th}}{S}} \cdot V_{dd} \quad (4)$$

S はサブスレッショルドファクタを表し，その下限は室温で 60 mV/dec. である．CMOS 論理回路におけるこれらのトレードオフを考慮すると，メモリアクセスの頻度に応じてアクセス頻度の大きいブロックのみを低い V_{th} で動作させることによってメモリへの平均アクセス時間をほとんど低下させることなくメモリの総エネルギー消費を削減できる可能性がある．

2.2 従来手法と提案手法

メモリのリーク電流を削減する手法はすでにいくつか提案されている[6)~8),10),11)]．回路レベルの代表的なリーク電流削減手法として Dynamic Leakage Cut-off（DLC）と呼ばれる手法が提案されている[10]．DLC は，選択されたメモリセルの N ウェルと P ウェルのバイアス電圧を動的に変更することによりアクセスされていないメモリセルのリーク電流を小さく抑えることができる．DLC はリーク電流の大幅な削減が見込まれるが，メモリにアクセスするたびにバイアス電圧を変更するためアクセス時間のオーバヘッドが無視できない．一方アーキテクチャレベルの代表的な手法として Dynamically Resizable Cache（DRI）[6]と呼ばれる手法や Drowsy Caches [12]，Cache Decay [8]などが提案されている．DRI 手法はキャッシュラインごとのアクセス頻度をプログラムの実行時に計測し，アクセス頻度が一定値より低くなるとキャッシュラインの電源供給を遮断する．DRI は非常に簡単なメカニズムで確実

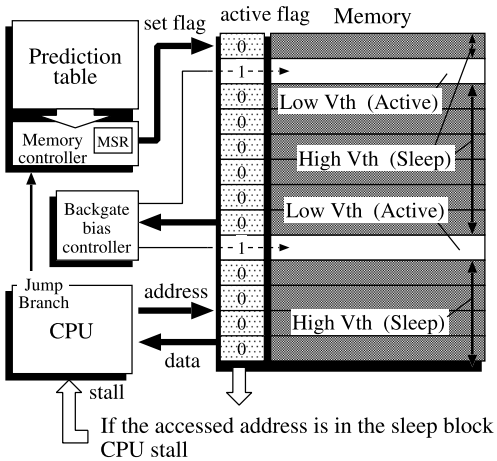

図 1 提案手法を実現するメモリの構成図
Fig. 1 An anatomy of proposed memory architecture.

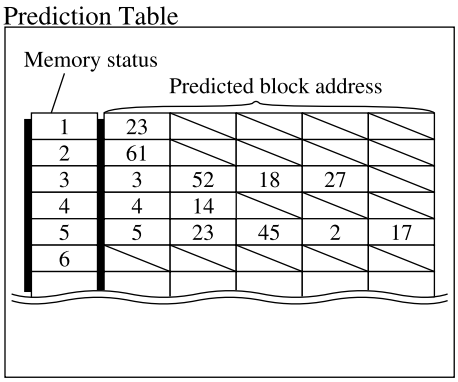

図 2 ブロック予測テーブル
Fig. 2 Block prediction table.

にリーク電流を削減することができるが，キャッシュメモリの実効的なサイズが小さくなるためキャッシュのヒット率が大幅に低下する可能性がある．Drowsy Caches や Cache Decay は，アクセス頻度の低いキャッシュラインを一定周期で高い閾値に設定するすることによりリーク電流を削減するが，閾値変更の周期が一定であるため，CPU が高い閾値に設定されたキャッシュラインをアクセスする頻度が高くなり，パフォーマンスが低下する可能性がある．筆者らはメモリのアレイ部分をいくつかのサブブロックに分割し（**図 1**），将来アクセスされるブロックをプロファイル情報から予測する手法を提案している[11),13),14)]．本稿では，文献 11) で提案した手法に加えて，低い閾値電圧を使用するメモリブロックの数をコンパイル時に決定することによりメモリのリーク電流をさらに削減する手法を提案する．本稿で提案する手法は組み込みシステムを対象としており，メモリブロックの閾値電圧はアプリケーションプログラムのコンパイル時にコンパイラによってスケジュールされることが特徴である．また，本稿で定式化する問題では閾値電圧の切替えに必要な遅延時間やエネルギー消費およびメモリブロックの予測に必要なエネルギー消費についても考慮する．

提案手法の特徴は，ブロック予測テーブルにより CPU が将来アクセスするメモリブロックを予測し，あらかじめ低い閾値電圧に設定することである．アクセスするブロックを予測することによりメモリのバックゲートバイアス変更に要する時間的オーバヘッドを隠蔽することができる．ブロック予測テーブルには将来アクセスされるメモリブロックの候補が格納されており，プログラムメモリの実行アドレスが基本ブロックをまたいだ直後，つまり CPU がジャンプ命令もしくは分岐命令を実行した後に，図 1 に示す専用のメモリコントローラが，ブロック予測テーブルからエントリを読み出す．エントリが読み出されるブロック予測テーブルの番地は CPU がアクセスしているメモリブロックの番号と同じ番号である．メモリコントローラはメモリステータスレジスタ（MSR）により，アクセスしたブロック予測テーブルのアドレスを記憶している．1 回前にアクセスしたブロック予測テーブルのアドレスと現在アクセスしているメモリブロックのアドレスが同じ場合にはブロック予測テーブルからのエントリの読み出しは行わない．このため閾値電圧変更の頻度はそれほど大きくないことが予想できる．コンパイラはプログラムのコンパイル時にメモリのブロックごとのアクセス回数をプロファイリングにより予測し，3 章で定式化する最適化問題を解くことによりブロック予測テーブルのエントリを決定する．**図 2** の例でプログラム実行時の動作を説明する．コンパイル時に図 2 に示すブロック予測テーブルのエントリを決定した場合，プログラム実行時に CPU がプログラムメモリのブロック 5 にアクセスしている間中は，ブロック 5，23，45，2，17 の閾値電圧は低く設定される．CPU が分岐命令またはジャンプ命令を実行しブロック 4 にアクセスした場合，ブロック 4 と 14 のメモリセルの閾値電圧が低く設定され，それ以外のメモリブロックは高い閾値電圧に設定される．

バックゲートバイアスの変更には**図 3** に示す MOS スイッチを使用する．active 信号を制御することによりメモリセルのバックゲート電圧を動的に変更し閾値電圧を変更する．バックゲートバイアスの変更にかかる時間は 5CPU サイクルを仮定している．バックゲートバイアスの変更にかかる時間および消費電力に関しては 4.2 節で詳しく議論する．

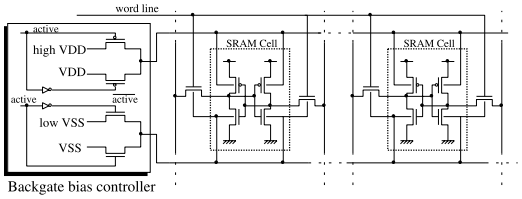

図 3 バックゲートバイアスコントローラの例
Fig. 3 A sample of the backgate bias controller.

3. 問題定義

本章では，平均アクセス時間の制約下でリーク電流を最小化する問題の定式化を行う．

3.1 用語の定義

問題定義の前に変数の定義を行う．

- N：メモリのブロック数．
- S：メモリのサイズ（bit）．
- メモリステータス i：ブロック予測テーブルから i 行が読み出され，読み出された番号のメモリブロックのみが低い閾値電圧に設定されている状態（$1 \leq i \leq N$）．
- A_{ij}：メモリステータス i のときに CPU がブロック j にアクセスする回数．
- X_i：メモリステータス i のときに実行される CPU のクロックサイクル数．
- a_{ij}：0-1 決定変数．ブロック予測テーブルのアドレス i（i 行目）にブロック j がエントリされているとき $a_{ij} = 1$，それ以外は $a_{ij} = 0$ をセットする．
- P：ブロック予測ミスペナルティ．CPU がスリープブロックへアクセスした際に，メモリセルの閾値を変更するために余分に必要となる CPU サイクル数．
- El：1CPU サイクルの間にメモリ 1 ビットで消費されるリークエネルギー．
- Er：ブロック予測テーブルの 1 ビット分を 1 回だけ読み出すのに必要なエネルギー．
- R_{ij}：ブロック予測テーブルのアドレス i が読み出された次にアドレス j が読み出される回数．
- Et：1 つのブロックの 1 回の閾値変更に必要なエネルギー．
- T_{cont}：プログラムの実行サイクル数に対する制約．

スリープブロックとは閾値が十分高く設定されリーク電流が無視できる状態にあるメモリブロックを意味し，アクティブブロックとは CPU が 1 サイクルでアクセスできるように十分に低い閾値が設定されているメモリブロックを意味する．したがって本稿では，アクティブブロックの読み出しアクセスサイクル数は 1 と仮定する．

3.2 問題の定式化

目的関数と制約条件はそれぞれ式 (6)，式 (7) である．決定変数は a_{ij} である．目的関数の第 1 項はメモリブロックで消費されるリークエネルギーの総和を表し，第 2 項はブロック予測テーブルで消費されるリークエネルギーの総和を表している．また，第 3 項と第 4 項はそれぞれブロック予測テーブルの読み出しエネルギーの総和，メモリブロックの閾値変更に必要なエネルギーの総和を表している．式 (7) は，キャッシュミスペナルティーを除く総実行サイクル数が制約サイクル数 T_{const} 未満でなくてはならないことを表している．A_{ij}，X_i，および R_{ij} はプログラムのコンパイル時にサンプルデータを用いたプロファイリングにより見積もることを想定している．

$$C_i = X_i + P \cdot \sum_{j=1}^{N} \{A_{ij} \cdot (1 - a_{ij})\} \tag{5}$$

$$\begin{aligned} Obj = {} & El \cdot \frac{S}{N} \cdot \sum_{i=1}^{N} \left(C_i \sum_{j=1}^{N} a_{ij} \right) \\ & + El \cdot \max_{\forall i} \left(\sum_{j=1}^{N} a_{ij} \right) \cdot N \cdot log(N) \sum_{i=1}^{N} C_i \\ & + Er \cdot \max_{\forall i} \left(\sum_{j=1}^{N} a_{ij} \right) \cdot log(N) \sum_{i=1}^{N} \sum_{j=1}^{N} R_{ij} \\ & + Et \sum_{i=1}^{N} \sum_{j=1}^{N} \left\{ R_{ij} \sum_{k=1}^{N} |(a_{ik} - a_{jk})| \right\} \end{aligned} \tag{6}$$

$$Sbj = \sum_{i=1}^{N} C_i \leq T_{const} \tag{7}$$

提案する手法を用いてリーク電流を最小化する最適化問題は，“制約条件 T_{const} の下で，Obj を最小化する a_{ij} をアプリケーションプログラムごとに決定する問題” として定義できる．

3.3 アルゴリズム

メモリブロックに対する閾値電圧の静的スケジューリングを決定する問題は，a_{ij} に対する 0-1 の最適な割当てを決定する問題である．N の値は 64 程度を想定しているため，最も単純な方法で組合せを探索すると計算量が膨大になる．本稿では，**図 4** に示すグリーディーアルゴリズムを用いて解を求めた．すべての a_{ij} に対して $a_{ij} = 0$ と設定した状態からアルゴリズムが開始する．つまり，初期状態ではブロック予測テーブルにエントリはない．この状態で制約条件が満たされない場合は，$a_{ij} = 0$ である a_{ij} の中で，その値を 1

Given: sets of X_i, A_{ij} and R_{ij}
Algorithm Memory optimization
  while (constraint is not satisfied)
    for each a_{ij}
      Select an a_{xy} which maximize $\frac{\partial obj(a_{xy})}{\partial sbj(a_{xy})}$.
    end for
    Selected a_{xy} is set to 1.
  end while
  Output a set of a_{ij} ;
end Algorithm

図 4 リークエネルギー削減アルゴリズム
Fig. 4 Leakage energy reduction algorithm.

に変更することによって $\frac{\mathbf{d}Obj}{\mathbf{d}Sbj}$ を最大にする a_{ij} を 1 つだけ選択し，その値を 1 に変更する．上記の手続きを制約条件を満たすまで繰り返す．制約条件が満たされた時点でアルゴリズムを終了し a_{ij} の集合を出力する．本アルゴリズムでは必ずしも最適な解は得られないが，計算オーダは N^2 と高速である．

4. 実験結果

4.1 実験環境

実験には**表 1** に示す 5 種類のベンチマークプログラムを使用した．ベンチマークプログラムは，DLX アーキテクチャ[15]を対象とする gcc-dlx コンパイラでコンパイルし，DLX アーキテクチャ用の命令レベルシミュレータ fast Ver. 0.97[16]を使って実行トレース情報を取得した．**表 2** は，それぞれのベンチマークプログラムの入力として使用した 3 種類のデータの実効サイズを示している．データの実効サイズとは，アクセスされたアドレス数を意味している．また，実験にはダイレクトマップ方式のキャッシュメモリを使用した．セットアソシアティブ方式のキャッシュメモリに対する実験は今後の課題である．ただし本手法は組み込みシステムを対象としており，キャッシュメモリに特化した手法ではない．

4.2 実験で対象とするシステム

本稿で提案するメモリのリーク電流削減手法は下記の項目で仮定するシステムを対象としている．

(1) 対象とするシステムはプロセッサとメインメモリで構成され，プロセッサは CPU と命令キャッシュ（i-cache）およびデータキャッシュ（d-cache）から成る．
(2) 1 回のメインメモリのアクセスサイクルは 1.3 と仮定する．
(3) キャッシュミスのペナルティーは 10 サイクルと仮定する．
(4) キャッシュのアドレスは図 1 に示すように，いくつかのサブブロックに分割されており，各々のブロックは独立に稼働状態とスリープ状態を選択できる．稼働状態とは CPU と同じ低い閾値で動作している状態を意味し，スリープ状態はリーク電流が無視できるほど高い閾値を使用している状態を表す．
(5) ブロック予測ミスペナルティーは 1 サイクルとする．
(6) メモリブロックの閾値変更に必要なサイクル数は 5 とする．
(7) キャッシュの読み出しサイクルは 1 クロックサイクルと仮定する（スリープ状態のブロックにアクセスした場合は 2 サイクル必要である）．
(8) スリープ状態のブロック（以下，スリープブロック）および稼働状態のブロック（以下，アクティブブロック）への書き込みサイクルは，いずれも 1 クロックサイクルと仮定する．書き込みアクセス時間は書き込み回路の動作速度に強く依存するためである．
(9) CPU のアドレスサイズ，キャッシュラインサイズ，キャッシュインデックスサイズおよびタグメモリのビット幅はそれぞれ，32 ビット，256 ビット，128，20 ビットである．したがって，メモリのサイズ S（ビット）は 35,328 ビットである．
(10) メモリブロック数 N は 64 とした．

スリープ状態から稼働状態への遷移時間の妥当性を確認するために 0.18 μm CMOS テクノロジの SPICE モデルを用いて回路シミュレーションを行った結果，**表 3** に示すとおり 5 クロックサイクル内の閾値電圧

表 1 ベンチマークプログラムの仕様
Table 1 Description of benchmark programs.

ベンチマーク	サイズ（ワード）
Arithmetic calculator	15,610
TV remote controller	15,360
Espresso (Boolean optimizer)	62,156
FFT	15,790
Compress	16,499

表 2 実効データサイズ（バイト）
Table 2 Active data size (byte).

ベンチマーク	Data1	Data2	Data3
Arithmetic calculator	15,560	15,996	15,916
TV remote controller	19,220	19,224	19,240
Espresso	29,260	71,188	151,068
FFT	17,016	16,940	16,872
Compress	66,472	64,112	61,140

表 3 予測ミスペナルティーと閾値変更の時間的オーバヘッド
Table 3 Prediction miss penalty and transition delay.

$P[cycle]$	$TransitionDelay[cycle]$
1	5

表 4 閾値変更に必要なエネルギーオーバヘッド
Table 4 Energy overhead for the transition of threshold voltage.

$Er[J]$	$El[J]$	$Et[J]$
1.23×10^{-12}	3.20×10^{-15}	2.52×10^{-12}

の変更は十分に可能であることが確認できた．

また，3 章で定義した Er，Et，および P に関しても同様の回路シミュレーションにより**表 4** の値を見積もった．閾値電圧を変更するためのスイッチはメモリセルに使用されるゲート幅の 12 倍のスイッチを使用した．リークエネルギー El に関しては，将来的には現在の値より大幅に増加すると想定して，キャッシュメモリ（ラインサイズ=256 ビット，インデックスサイズ=128）に対する 1 回の読み出しで消費されるエネルギーの 1/3 がキャッシュメモリ全ビットの 1 サイクルのリークエネルギーの総和に等しいと仮定した．

4.3 比較実験に用いた関連手法

提案手法の有効性を示すために，筆者らの提案する手法以外に下記の 2 種類の手法に対して同様の実験を行った．

Static approach メモリをいくつかのブロックに分割し，アクセス頻度の高い少数のメモリブロックに対して低い閾値を割り当てる．プログラムの実行時には閾値を変更しない．アクセス頻度の情報はサンプルデータを用いたプロファイリングにより取得した．パフォーマンスの制約条件に応じて低い閾値を使用するブロック数を変更する．本手法は筆者らが文献 17) で提案した手法と基本コンセプトは同様である．

Dynamic profiling メモリをいくつかのブロックに分割し，一定周期内にアクセスのなかったキャッシュラインを高い閾値に設定しアクセスのあったキャッシュラインを低い閾値に設定する．パフォーマンスの制約条件に応じてサンプリング周期を変更する．本手法は，境界条件が多少異なるが，文献 8)，12) で提案されている手法と同様である．

4.4 実 験 結 果

図 5，**6**，**7** に示すグラフは，パフォーマンスの制約条件を変更したときのエネルギー消費の値の推移を示している．横軸にパフォーマンスに対する制約条件を，縦軸にメモリのリークエネルギーを示した．横軸のパフォーマンスの値は，すべてのメモリブロックが低い閾値電圧に設定されているときのプログラムの実行時間を 100%として正規化した値である．縦軸のリークエネルギーの値も同様にすべてのブロックが低い閾値電圧に設定されているときのリークエネルギーを 100%として正規化した．ただし，図 7 の縦軸の値は式 (6) に示した Obj の値を示しているため，正確にはリークエネルギーだけでなく，ブロック予測テー

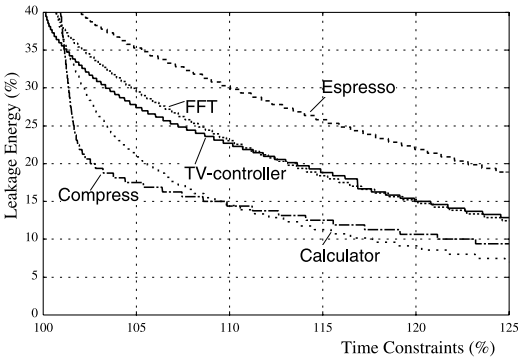

図 5 Static approach に対する実験結果
Fig. 5 Results for the static approach.

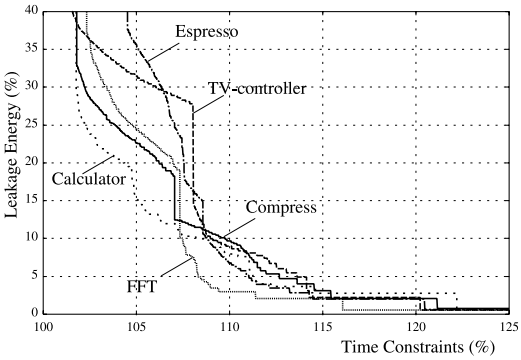

図 6 Dynamic profiling に対する実験結果
Fig. 6 Results for the dynamic profiling.

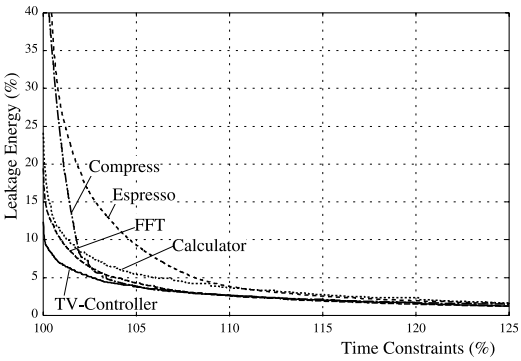

図 7 提案手法に対する実験結果
Fig. 7 Results for our method.

ブルの読み出しに必要なエネルギーや閾値の変更に必要なエネルギーも含まれている．本実験においては，スリープブロックの読み出しアクセスサイクル数を2，アクティブブロックのリードアクセスサイクル数を1とした．実験結果から提案手法によるリークエネルギー削減効果が非常に大きいことが分かる．キャッシュメモリではほとんどすべてのキャッシュラインのアクセス頻度が高いため Static approach のように閾値電圧を固定するとパフォーマンスが大幅に低下することが分かる．また，Dynamic profiling の効果がそれほど大きくない理由は，プログラムの動作状況と関連付けせずにプロファイリングを行っても，ある時点でのメモリアクセスのパターンが将来的に再現しないためであると思われる．提案手法ではプログラムの実行に合わせてブロック予測テーブルのエントリを読み出し，次にアクセスするメモリブロックを予測するため，メモリアクセスパターンがダイナミックに変更されても予測ヒット率を高く保つことができる．また，サブブロックの閾値の変更は，プログラムのアドレスが基本ブロックをまたいだとき，かつ，前回読み出したブロック予測テーブルのアドレスと異なる場合にのみに行われるため，本実験では約10サイクルに一度の頻度でしか閾値の変更が行われないことが確認できた．この結果から，メモリブロックの閾値を変更するためのエネルギーはメモリの読み出しに必要なエネルギーの約2〜3%程度であることが分かった．

次に，プロファイリングに使用するデータを変更した際の結果を**図8**と**図9**に示す．図7に示す結果は，ブロック予測テーブルのエントリを決定するために使用する入力データとリークエネルギーの評価に使用するデータに同じデータを使用した．ブロック予測テーブルのエントリはシステムのパフォーマンスを決定する鍵となるため，入力データにあまり依存せずにブロック予測テーブルのエントリを決定できることが好ましい．図8および図9に示す実験結果は，パフォーマンスの劣化が入力データにほとんど依存しないことを裏付けている．メモリへのデータ配置がアプリケーションプログラムの構造とコンパイラに強く依存しているためであると思われる．逆に，コンパイラが入力データになるべく依存しないデータ配置を行えば提案手法の効果が十分に得られると思われる．結果として，コンパイル時に無作為に選択したサンプルデータを用いてメモリアクセスの履歴情報を取得し，ブロック予測テーブルのエントリを決定することにより，任意のデータに対して高速かつ低リーク動作が可能となる．

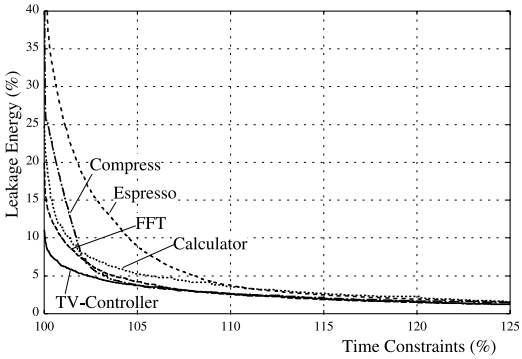

図8 プロファイルデータ2に対する実験結果
Fig. 8 Results for a different profile (Data2).

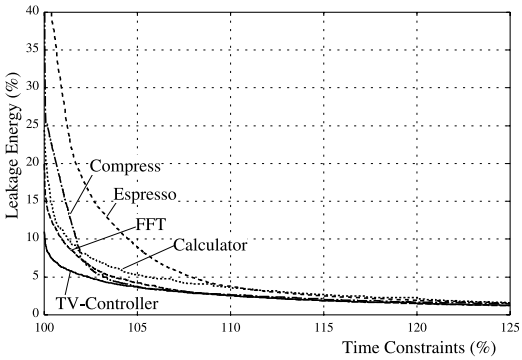

図9 プロファイルデータ3に対する実験結果
Fig. 9 Results for a different profile (Data3).

5. おわりに

トランジスタサイズの縮小にともなって，近い将来には電源電圧も1V以下に比例縮小される．そのような世代には，スイッチング速度を改善するために閾値電圧を0V以下に設定するトランジスタも登場する可能性がある．低電圧のメモリブロックは論理ブロックと比較して稼働率が低いためデータ保持のためだけにエネルギーを大量に消費してしまう．このため稼働率に応じてリーク電流を動的に制御するメカニズムが今後ますます重要になると考えられる．本稿で提案した手法は，アクセスされるブロックを適切に予測することによりわずかなパフォーマンス劣化だけでリーク電流を大幅に削減することが可能であることが確認できた．今後はブロック予測のアルゴリズムを改善するとともに，ロジックブロックへの応用を検討する予定である．

参考文献

1) Bellas, N. and Hajj, I.: Architectural and Compiler Support for Energy Reduction in the

Memory Hierarchy of High Performance Microprocessors, *Proc. Int'l Symposium on Low Power Electronics and Design* (*ISLPED'98*), pp.70–75 (1998).
2) Borker, S.: Design Challenges of Technology Scaling, *IEEE Micro*, Vol.19, No.4, pp.23–29 (1999).
3) Mutoh, S., Date, S., Shibata, N. and Yamada, J.: 1-V, 30-MHz Memory-Macrocell-Circuit Technology with a 0.5 μm Multi-threshold CMOS, *Proc. IEEE Symposium on Low Power Electronics*, pp.90–91 (1994).
4) Kuroda, T. and Sakurai, T.: Threshold-voltage control scheme through substrate-bias for low-power highspeed CMOS LSI design, *Kluwer J. of VLSI signal processing, special issues on technologies for wireless computing* (1996).
5) Nii, K., Makino, H., Tujihashi, Y., Morishima, C. and Hayakawa, Y.: A Low Power SRAM using Auto-Backgate-Controlled MT-CMOS, *Proc. Int'l Symposium on Low Power Electronics and Design* (*ISLPED'98*), pp.293–298 (1998).
6) Powell, M., Yang, S.-H., Falsafi, B., Roy, K. and Vijaykumar, T.N.: Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories, *Proc. Int'l Symposium on Low Power Electronics and Design* (*ISLPED'00*), pp.90–95 (2000).
7) Yang, S.-H., Powell, M., Falsafi, B., Roy, K. and Vijaykumar, T.N.: An Integrated Circuit/Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches, *Proc. Int'l Symposium on High Performance Computer Architecture* (*HPCA'01*) (Jan. 2001).
8) Kaxiras, S., Hu, Z. and Martonosi, M.: Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power, *Proc. Int'l Symposium on Computer Architecture* (*ISCA'01*), pp.240–251 (2001).
9) Hiramoto, T. and Takamiya, M.: Low Power and Low Voltage MOSFETs with Variable Threshold Voltage Controlled by Back-Bias, *IEICE Trans. Electronics*, E83–C(2), pp.161–169 (Feb. 2000).
10) Kawaguchi, H., Itaka, Y. and Sakurai, T.: Dynamic Leakage Cut-off Scheme for Low-Voltage SRAM's, *Proc. Simposium on VLSI Circuits*, pp.140–141 (1998).
11) Ishihara, T. and Asada, K.: An Architectural Level Energy Reduction Technique for Deep-Submicron Cache Memories, *Proc. VLSI Design'02 & ASP-DAC'02*, pp.282–287 (Jan. 2002).
12) Flautner, K., Kim, N.S., Martin, S., Blaauw, D. and Mudge, T.: Drowsy Caches: Techniques for Reducing Leakage Power, *Proc. Int'l Symposium on Computer Architecture* (*ISCA'02*) (May 2002).
13) 石原 亨，浅田邦博：ディープサブミクロン時代におけるキャッシュメモリのリーク電流削減手法，情報処理学会研究報告 CPSY2001-61, pp.1–6 (2001).
14) 石原 亨，浅田邦博：メモリの低消費電力化を目的とした閾値電圧の静的スケジューリング手法，DA シンポジウム論文集，pp.55–60 (2002).
15) Hennessy, J.L. and Patterson, D.A.: *Computer Architecture: A Quantitative Approach*, 2nd edition, Morgan Kaufmann Publishers, Inc. (1996).
16) http://www-mount.ece.umn.edu/~okeefe/mcerg/fast-dlx/
17) Ishihara, T. and Asada, K.: A System Level Memory Power Optimization Technique Using Multiple Supply and Threshold Voltages, *Proc. Asia and South Pacific Design Automation Conference* (*ASPDAC'01*), pp.456–461 (Jan. 2001).

(平成 14 年 10 月 16 日受付)
(平成 15 年 3 月 4 日採録)

石原 亨（正会員）

平成 7 年九州大学工学部情報工学科卒業．平成 12 年同大学大学院博士課程修了．プロセッサシステムの低消費電力化に関する研究に従事．現在，東京大学大規模集積システム設計教育研究センター（VDEC）助手．平成 10 年九州支部奨励賞，平成 14 年システム LSI 設計技術研究会優秀論文賞受賞．IEEE，電子情報通信学会各会員．

浅田 邦博

昭和 50 年東京大学工学部電子工学科卒業．昭和 55 年同大学大学院博士課程修了．平成 7 年同大学教授．現在，東京大学大規模集積システム設計教育研究研究センター（VDEC）センター長．IEEE，IEEJ，電子情報通信学会各会員．