システムＬＳＩ設計技術　103－5
（2001. 11. 29）

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

石原　亨†　浅田　邦博†

† 東京大学 大規模集積システム設計教育研究センター
〒113–8656 東京都文京区本郷 7–3–1

E-mail: †{ishihara,asada}@vdec.u-tokyo.ac.jp

あらまし　近年の集積回路の低電圧化にともない，閾値電圧が低下しサブスレッショルドリーク電流の増加が問題となっている．一方，プロセッサベースのシステムではCPUの一クロックサイクル内にキャッシュメモリからのデータを読み出すためにキャッシュメモリの高速化が重要な課題となっている．本稿では，キャッシュメモリの高速化とリーク電流の削減を目的としたアーキテクチャレベルの手法を提案する．キャッシュメモリのアレイ部分を幾つかのブロックに分割し，少数のブロックのみを低閾値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．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し，閾値を動的に変更させることにより，アクセス時間を増加させること無くキャッシュメモリのリーク電流を1/20に削減できることを確認した．

キーワード　キャッシュメモリ，低消費電力化設計，サブスレッショルドリーク

A Leak Energy Reduction Technique for Deep Submicron Cache Memories

Tohru ISHIHARA† and Kunihiro ASADA†

† VLSI Design and Education Center, University of Tokyo
7–3–1 Hongo, Bunkyo-ku, Tokyo, 113–8656, Japan

E-mail: †{ishihara,asada}@vdec.u-tokyo.ac.jp

Abstract　An architectural level technique for a high performance and low energy cache memory is proposed in this paper. The key idea of our approach is to divide a cache memory into several number of cache blocks and to activate a few parts of the cache blocks. The threshold voltage of each cache block is dynamically changed according to an utilization of each block. Frequently accessed cache blocks are woken up and others are put to sleep by controlling the threshold voltage. Since time overhead to change the threshold voltage can not be neglected, predicting a cache block which will be accessed in next cycle is important. History based prediction technique to predict cache blocks which should be woken up is also proposed. Experimental results demonstrated that the leakage energy dissipation in cache memories optimized by our approach can be less than 5% of energy dissipation in a cache memory which does not employ our approach.

Key words　Cache memory, Low power design, Subthreshold leak

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無くなってきている．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時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]のなかでBorker氏は，単位ゲート幅当りのリーク電流は世代毎に5倍に増加すると予想している．単位面積当りのダイナミックなエネルギーの消費電力がそれほど変らないとすると，加工寸法が$0.10\mu m$未満になる頃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 MT-CMOS (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よく決定する手法[6]が重要である．キャッシュメモリのリーク電流を削減する手法も既にいくつか提案されている[7]～[9]．これらの手法は，アクセス頻度が低いと予測されるキャッシュ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．これらの手法は"完全稼働状態"と"停止状態"の2種類の状態しか持たないが，閾値電圧を調節することによりさらに大幅なリーク電流の削減が期待できる．本稿で提案する手法の基本的なアイデアは，キャッシュメモリを幾つかのサブブロックに分割し，そのうち少数のサブ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である．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サブブロックを予測する機構についても提案する．

2章では，キャッシュメモリのリーク電流を削減する手法の基本的アイデアを述べ，3章では提案するSAC(Selectively activated cache)アーキテクチャとSACアーキテクチャを用いたリーク電流の削減手法について説明する．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)に示す通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[10]．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戊ロックのみを低いV_{th}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低下させること無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 可変閾値電圧 SRAM

閾値を動的に変更する方法としてNMOS，PMOS両方のバックゲートバイアスを動的に変更する手法と電源電圧とNウェルの電位を変更する方法がある．NMOS，PMOS両方のバックゲートバイアスを変更する場合は，3重ウェル構造が必要となるが，図1は比較的安価な2重ウェル構造で閾値を動的に変更する回路の例である[5]．トランジスタ Q1，Q2，Q3，Q4はリーク電流を遮断するためにあらかじめ高い閾値に設定されたMOSトランジスタである．メモリが稼働状態の時 Q1，Q2，Q3はオン状態となる．この時，メモリセルの電源線であるVVDDは，Q1を通してVdd1と同電位(1.7V)となる．PMOS側のバックゲートバイアスもQ3を通してVdd1 と同電位となる．メモリセルのグランド線であるVGNDはQ2を通してグランドレベルに短絡される．NMOS側のバックゲートバイアスは常にグランドレベルである．一方スリープ時にはVVDDとVGNDはダイオードにより逆バイアスされ閾値が引き上げられる．この時メモリセルのリーク電流は基板バイアス効果により大幅に削減される．SPICEを用いた回路シミュレーションによりスリープ時のリーク電流を稼働時の1/1000 に抑えられることを実証している．

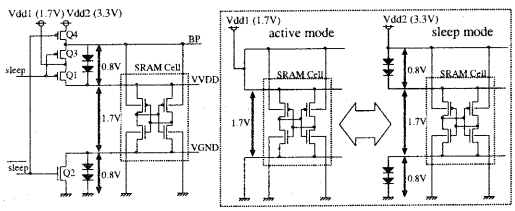

図1 動的可変閾値電圧SRAMの例

2.3 提案手法

メモリ参照の局所性は良く知られている．この性質はメモリのごく一部のデータのみが頻繁にアクセスされることを表している．時間的な局所性に注目すると，ある時間的局面においてはごく一部のメモリアドレスしかアクセスされていない．この性質を考慮して，アクセス頻度の低い大部分のメモリブロックの閾値を高く設定することにより，平均アクセス時間をほとんど増加させること無くリーク電流を大幅に削減することが可能である．

本稿では，少数のキャッシュラインのみを選択的に活性化する機能を持った電力制御キャッシュメモリアーキテクチャ(Selectively Activated Cache : SACアーキテクチャ)とSACアーキテクチャを用いたリーク電流の削減手法を提案する．提案手法の特徴はキャッシュアドレスを幾つかのサブブロックに分割し，少数のサブブロックのみを低い閾値で動作させることにより平均アクセス時間をほとんど増加させること無くキャッシュメモリのリーク電流を大幅に削減できることである．閾値電圧を動的に変更する方法として，VT-CMOS[4]や，図1に示したABC-MOS[5]を想定している．

3. SACアーキテクチャ

本章では，対象とするシステムについて説明し，SACアーキテクチャを用いて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を定義する．

3.1 対象とするシステム

本稿で提案するキャッシュメモリのリーク電流削減手法は下記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) 一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2に示すように，幾つかのサブブロックに分割されており，各々のサブ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) キャッシュの読み出しは稼働状態でのみ可能である．

(6) スリープ状態から稼働状態に移行する時間は1クロックサイクルと仮定する．

(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) スリープ状態のサブブロック(以下，スリープブロック)および稼働状態のサブブロック(以下，稼働ブロック)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．

スリープ状態から稼働状態への遷移時間の妥当性を評価するために回路シミュレーションを行った．実験結果を図3に示す．実験のために4ビット×128ワードのSRAMブロックを0.5μmCMOSテクノロジを使用して設計した．回路シミュレーションの結果，読み出

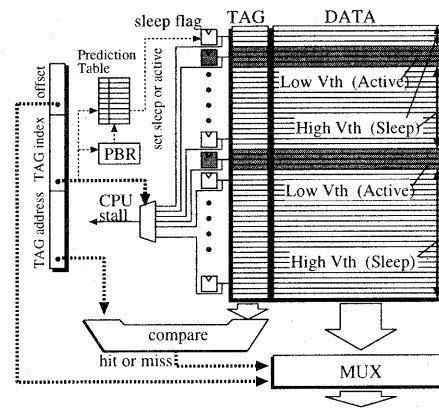

図2 SACアーキテクチャ

しサイクル時間 (1 クロックサイクル) 内にスリープ状態から稼働状態への遷移が可能であることが確認できた.

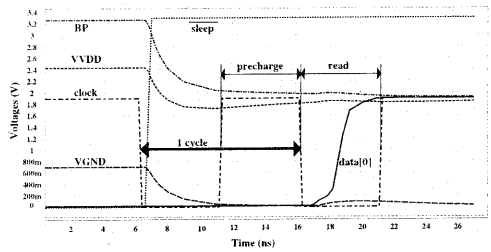

図 3 稼働状態への遷移時間

3.2 ブロック予測

スリープブロックへのアクセス回数が多くなるとキャッシュのパフォーマンスが著しく低下する可能性があるため, 稼働ブロックへのヒット率を向上させることが重要である. 稼働ブロックのヒット率を向上させるためにブロック予測テーブルを採用した. ブロック予測とは分割されたサブブロックのうちで次のアクセスで最もアクセスされる可能性が高いサブブロックを予測することを意味する. ブロック予測テーブルは, 過去の履歴情報をもとに, 次にアクセスされるサブブロックの候補を登録するためのメモリである. ブロック予測テーブルの例を図 4に示す. 分割されたサブブロックは, 各サブブロック毎に次にアクセスされる幾つかのサブブロックの候補を持っている. 図 4の例では, CPU が現在サブブロック 0 にアクセスしている時, サブブロック 32, 1, $\cdots$ 19 が稼働状態に遷移しはじめる. フロック予測テーブルのエントリーは下記の履歴情報をもとに決定する. サブブロック毎のエントリー数 (ne) が増加すると稼働ブロックのヒット率は向上するが, キャッシュメモリのリーク電流が増加する. また分割数 (nb) を大きくするとブロック予測テーブルのサイズが大きくなり, ブロック予測テーブル自体の面積とリーク電流が無視できなくなる. したがって, エントリー数 (ne) および分割数 (nb) はアプリケーションに応じて適切に決定する必要がある.

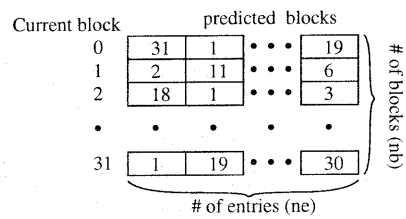

図 4 ブロック予測テーブル

図 5 はキャッシュの読み出しタイミングチャートである. レジスタ PBR は一サイクル前にアクセスされたサブブロックのアドレスを記憶する. それぞれのサブブロックは図 2に示す通りスリープフラグを持っており, 稼働状態の時はスリープフラグが 1 にセットされる. PBR の値と現在アクセスされているブロックのアドレスが異なる場合, スリープフラグの値を更新する. 同時に, スリープフラグが 1 にセットされたサブブロックは稼働状態に遷移する. 現在アクセスされているサブブロックがスリープ状態であった場合 1 クロックサイクルのペナルティーが発生する.

3.3 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義を行う. 問題定義の前に変数の定義を行う.

- ne_i : 命令キャッシュのブロック予測テーブルのエントリー数
- ne_d : データキャッシュのブロック予測テーブルのエントリー数
- nb_i : 命令キャッシュのサブブロック数
- nb_d : データキャッシュのサブブロック数
- $PM_i(ne_i, nb_i)$: 命令キャッシュのブロック予測ミス回数
- $PM_d(ne_d, nb_d)$: データキャッシュのブロック予測ミス回数
- $LS\ (= 256\ bits)$: キャッシュラインサイズ
- $IX\ (= 128)$: キャッシュインデックスサイズ
- $TW\ (= 20\ bit)$: タグメモリのビット幅

$$OBJ = \frac{(LS+TW)\cdot IX}{nb_i}\cdot(ne_i+1)+ne_i\cdot nb_i\cdot log(nb_i)+\frac{(LS+TW)\cdot IX}{nb_d}\cdot(ne_d+1)+ne_d\cdot nb_d\cdot log(nb_d) \quad (5)$$

$$PM_i(ne_i, nb_i)+PM_d(ne_d, nb_d)\leqq N_{const} \quad (6)$$

目的関数と制約条件はそれぞれ式 (5), 式 (6) である. 決定変数は nb_i, ne_i, nb_d, ne_d である. 目的関数は稼働状態になる総ビット数を表している. スリープブロックのリーク電流は無視できるほど小さいものとする. 従って, アレイ部分のリーク電流は稼働状態の SRAM セルの数にほぼ比例する. また, ブロック予測テーブルは常に低閾値動作させることを想定しているため, ブロック予測テーブルはそのサイズに比例したリーク電流を消費する. ただし, ブロック予測テーブルの読み出し頻度は非常に小さいため読み出し時のエネルギー消費は無視できるものと仮定した. 目的関数の 1 項目は命令キャッシュの稼働ビット数, 2 項目

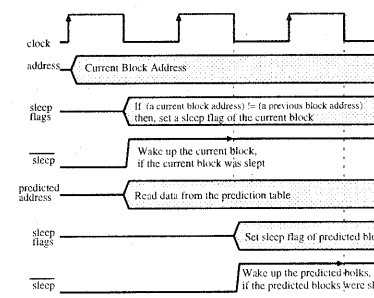

図 5 Timing diagram of a read access

表1 ベンチマークプログラムの仕様

ベンチマーク	プログラムサイズ (ワード)
Arithmetic Calculator	11,451
TV Remote Controller	15,360
Espresso	62,156

表2 実効データサイズ (バイト)

ベンチマーク	Data1	Data2	Data3
Arithmetic Calculator	14,816	15,240	15,164
TV Remote Controller	19,232	19,216	19,220
Espresso	29,260	71,192	151,072

は命令キャッシュのブロック予測テーブルの総ビット数，3項目，4項目はそれぞれデータキャッシュの稼働ビット数，ブロック予測テーブルの総ビット数を表している．式(6)は，ブロック予測のミス回数が制約条件 N_{const} 未満でなくてはならないことを表している．

SACアーキテクチャを用いてリーク電流を最小化する問題は，“制約条件 N_{const} の下で，OBJ を最小化する ne_i，nb_i，ne_d および nb_d，をアプリケーションプログラム毎に決定する問題”として定義できる．

4. 実験結果

実験には表1に示す3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[11]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[12]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．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．本実験はダイレクトマップ方式のキャッシュメモリを対象としている．セットアソシアティブキャッシュに対する実験は今後の課題である．

はじめに，命令キャッシュとデータキャッシュのサブブロック数 (nb_i と nb_d) を変更した時の，OBJ，$PM_i(nb_i, ne_i)$，および $PM_d(nb_d, ne_d)$ の値を評価した．この実験には $Data1$ のみを使用した．表3にリーク電流とプログラムの実行時間に関する実験結果を示す．本稿で仮定するモデルでは，リーク電流はアプリケーションプログラムと入力データには依存しないため，リーク電流に関する実験結果は一種類のみである．実行時間は命令レベルシミュレータにより計測した実行サイクル数により求めた．表3の値は提案手法を適用していないキャッシュメモリの値で正規化した値である．実験結果から，SACアーキテクチャを採用することにより，3%程度のパフォーマンス低下のみでリーク電流を1/15に削減できることが確認できる．
実験の結果からブロック予測の効果が非常に大きいことが確認できる．例えばブロック予測テーブルの候補数を0から1に増やすとパフォーマンスの劣化は半分に減少されている．

表3 サブブロック数を変更した時の結果

Leakage Energy				
nb_i	$ne_i = 0$	$ne_i = 1$	$ne_i = 2$	$ne_i = 3$
8	12.5%	25.07%	37.64%	50.20%
16	6.25%	12.68%	19.11%	25.54%
32	3.13%	6.70%	10.26%	13.86%

実行時間 (命令キャッシュ)				
Arithmetic Calculator				
nb_i	$ne_i = 0$	$ne_i = 1$	$ne_i = 2$	$ne_i = 3$
8	104.10%	102.12%	101.54%	101.17%
16	105.41%	102.80%	102.12%	101.59%
32	107.56%	103.07%	102.12%	101.58%
TV Remote Controller				
nb_i	$ne_i = 0$	$ne_i = 1$	$ne_i = 2$	$ne_i = 3$
8	102.21%	101.18%	100.74%	100.27%
16	103.13%	101.61%	101.05%	100.52%
32	104.73%	101.94%	101.12%	100.62%
Espresso				
nb_i	$ne_i = 0$	$ne_i = 1$	$ne_i = 2$	$ne_i = 3$
8	102.12%	101.13%	100.81%	100.53%
16	102.97%	101.39%	100.96%	100.70%
32	104.74%	101.92%	101.40%	100.92%

実行時間 (データキャッシュ)				
Arithmetic Calculator				
nb_d	$ne_d = 0$	$ne_d = 1$	$ne_d = 2$	$ne_d = 3$
8	117.60%	106.58%	101.08%	100.88%
16	117.76%	106.66%	101.45%	101.09%
32	118.17%	107.39%	102.16%	101.66%
TV Remote Controller				
nb_d	$ne_d = 0$	$ne_d = 1$	$ne_d = 2$	$ne_d = 3$
8	113.12%	105.95%	101.87%	101.09%
16	114.32%	107.99%	103.73%	102.35%
32	115.70%	108.58%	104.69%	102.78%
Espresso				
nb_d	$ne_d = 0$	$ne_d = 1$	$ne_d = 2$	$ne_d = 3$
8	112.89%	107.66%	104.59%	102.43%
16	114.30%	109.85%	106.89%	104.76%
32	116.43%	111.82%	109.24%	107.73%

次に，入力データを変更した際の実行時間の結果を表4に示す．表3に示す結果は，ブロック予測テーブルのエントリーを決定するために使用する入力データと実行時間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ーはSACアーキテクチャのパフォーマンスを決定する鍵となるため，入力データに依存せずにブロック予測テーブルのエントリーを決定できることが好ましい．表4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．キャッシュが頻繁にアクセスするアドレスは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．したがって，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キャッシュ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ーを決定することにより，任意のデータに対して実行時間を小さく抑えることが可能となる．

最後に，実行時間の制約下でリーク電流を最小化したときに実験結果を図6に示す．リーク電流の結果は命令キャッシュとデータキャッシュのリーク電流の合計を，従来のキャッシュメモリのリーク電流の値で正規化した値である．従来のキャッシュメモリのリー

表4　入力データを変更した時の実験結果

命令キャッシュ			
Arithmetic Calculator			
Predicted for input data	Data1	Data2	Data3
Data1	101.58%	101.77%	101.59%
Data2	101.92%	101.73%	101.81%
Data3	101.76%	101.76%	101.66%
TV Remote Controller			
Predicted for input data	Data1	Data2	Data3
Data1	100.62%	100.68%	100.68%
Data2	100.65%	100.72%	100.72%
Data3	100.64%	100.71%	100.71%
Espresso			
Predicted for input data	Data1	Data2	Data3
Data1	100.92%	101.07%	101.03%
Data2	101.33%	100.95%	101.41%
Data3	101.00%	101.29%	100.69%

データキャッシュ			
Arithmetic Calculator			
Predicted for input data	Data1	Data2	Data3
Data1	101.66%	101.70%	101.69%
Data2	102.18%	102.05%	102.04%
Data3	102.04%	101.91%	101.91%
TV Remote Controller			
Predicted for input data	Data1	Data2	Data3
Data1	102.78%	102.85%	102.78%
Data2	102.30%	102.33%	102.30%
Data3	102.53%	102.60%	102.53%
Espresso			
Predicted for input data	Data1	Data2	Data3
Data1	107.73%	108.91%	110.99%
Data2	112.62%	110.49%	114.61%
Data3	121.53%	120.41%	115.67%

ク電流の値とは，SACアーキテクチャを採用しないキャッシュメモリを時間制約を満たすように動作させたときのリーク電流を意味する．実行時間も同様の方法で計算した．図6に示す実験結果から，実行時間を5%増加させるだけでキャッシュメモリのリーク電流を1/20に削減できることが確認できた．

5. おわりに

トランジスタサイズの縮小に伴って，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SACアーキテクチャは，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共に，セットアソシアティブキャッシュへの応用を検討する予定である．

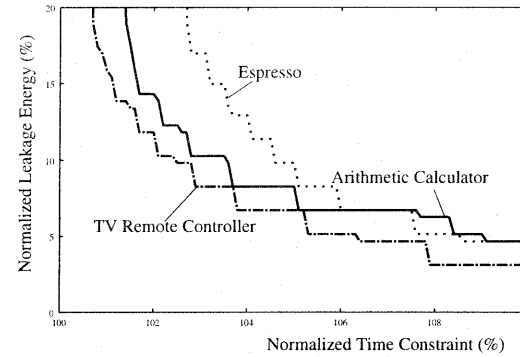

図6　時間制約下でのリーク電流の結果

文　献

[1] Nikolaos Bellas, and Ibrahim Hajj,. "Architectural and Compiler Support for Energy Reduction in the Memory Hierarchy of High Performance Microprocessors". In *Proc. of ISLPED'98*, pages 70–75, 1998.

[2] S. Borker. "Design Challenges of Technology Scaling". *IEEE Micro*, 19(4):23–29, July 1999.

[3] S. Mutoh S. Date, N. Shibata and J. Yamada. "1-V, 30-MHz Memory-Macrocell-Circuit Technology with a 0.5μm Multi-threshold CMOS". In *Proc. of IEEE Symposium on Low Power Electronics*, pages 90–91, 1994.

[4] T. Kuroda and T. Sakurai. "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 computeiog*, 1996.

[5] K. Nii, H. Makino, Y. Tujihashi, C. Morishima, and Y. Hayakawa. "A Low Power SRAM using Auto-Backgate-Controlled MT-CMOS". In *Proc. of Int'l Symposium on Low Power Electronics and Design*, pages 293–298, 1998.

[6] T. Ishihara and K. Asada. "A System Level Memory Power Optimization Technique Using Multiple Supply and Threshold Voltages". In *Proc. of ASPDAC'01*, pages 456–461, Jan. 2001.

[7] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T.N. Vijaykumar. "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories". In *Proc. of ISLPED'00*, pages 90–95, 2000.

[8] S.-H. Yang, M. Powell, B. Falsafi, K. Roy, and T.N. Vijaykumar. "An Integrated Circuit/Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches". In *Proc. of HPCA'01*, Jan. 2001.

[9] S. Kaxiras, Z. Hu and M. Martonosi. "Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power". In *Proc. of ISCA'01*, pages 240–251, 2001.

[10] T. Hiramoto and M. Takamiya. "Low Power and Low Voltage MOSFETs with Variable Threshold Voltage Controlled by Back-Bias". *IEICE Trans. on Electronics*, E83–C(2):161–169, February 2000.

[11] J. L. Hennessy and D. A. Patterson. *"Computer Architecture: A Quantitative Approach"*. Morgan Kaufmann Publishers, Inc., 2nd edition, 1996.

[12] http://www-mount.ece.umn.edu/~okeefe/mcerg/fast-dlx/.