

Self-Timed Cut-Off法の待ち時間動的最適化によるリーク電流削減手法

樋口 昭彦[†] 小林 和淑[†] 小野寺秀俊[†]

[†] 京都大学大学院情報学研究科 〒606-8501 京都市左京区吉田本町
E-mail: †{higuchi,kobayasi,onodera}@vlsi.kuee.kyoto-u.ac.jp

あらまし 動作時においてリーク電流を減らすために、予め決められた待ち時間だけ動いていないブロックの電源を切る Self-Timed Cut-Off 法が有効である。このとき最適な待ち時間を設定することが重要となる。単一用途の回路ブロックでは使用状況を予測できるが、汎用プロセッサの演算器においてはアプリケーション毎に使用状況が変化する。本稿では待ち時間を動的に最適化することを提案し、使用状況が刻々と変化する場合においてもリーク電流を削減可能であることを示す。

キーワード 消費電力, アクティブリーク電流削減, スイッチ

A Leakage Current Reduction Technique by Optimizing Waiting Time Dynamically in Self-Timed Cut-Off Scheme

Akihiko HIGUCHI[†], Kazutoshi KOBAYASHI[†], and Hidetoshi ONODERA[†]

[†] Graduate School of Informatics, Kyoto University
E-mail: †{higuchi,kobayasi,onodera}@vlsi.kuee.kyoto-u.ac.jp

Abstract Self-Timed Cut-Off scheme reduces leakage current effectively even if a chip is in an active mode. The circuit block is cut off if it is not used for the predetermined waiting time. It is important to determine the most appropriate waiting cycle in this scheme. The enable statistics of ASIC are expectable, but that of the computing unit in general-purpose processors varies by the executed applications. This paper proposes a dynamic optimization scheme of the waiting time in Self-Timed Cut-Off scheme. The proposed scheme reduce leakage current even if the enable statistics varies in certain intervals.

Key words Power consumption, Active leakage current reduction, Cut-off switch

1. はじめに

プロセスの微細化と共にリーク電流が増加し続けている。電力削減には電源電圧の低下が効果的である。パフォーマンスを維持しつつ電源電圧を下げる方法として、MOS のしきい値電圧も下げることが一般的であるが、サブスレシヨルドリーク (Subthreshold leakage) が指数的に増加する。ゲート酸化膜もスケーリング則に沿って薄くなり、20Å 程度になるとトンネル効果によるゲートリーク (Gate leakage) がリーク成分で支配的となる。70nm 以降ではゲートリークがサブスレシヨルドリークを上まわると予測されている [1]。また、50nm 以下になると接合リーク (Junction leakage) も問題になってくる。ISSCC2004 において Intel は 90nm プロセスで製造された Pentium4 のリーク電力が消費電力全体の約 30% にも及ぶと発表した [2]。

以上の背景からリーク電流削減技術の研究が盛んに行われている。これらは主にスタンバイ時のリーク電流削減を目的とし

ている。しかし、実際にはアクティブ時にも回路全体が使用されているわけではない。例えば、CPU ではメモリデータ転送命令が実行されているときはシフタや乗算器などは使用されていない。アクティブ時のリーク電流削減にスイッチによる電源遮断を用いる時、2つの大きな問題がある。電源切替から回路が安定するまでの時間と、その時の容量の充放電によって発生する動的電力消費である。そのためスイッチの切替が頻繁に起こると、ストールによる実行時間増加や動的電力消費で消費エネルギーがスイッチを使用しない場合に比べて増加してしまう。上記の問題を解決するアプローチとして [3] で Self-Timed Cut-Off 法を提案している。この技術は回路ブロックの使用状況がキャッシュメモリのように以前の履歴に依存することに基づいている。Self-Timed Sleep Timer をスイッチに接続し、待ち時間 (Waiting Time) だけ回路ブロックが使用されていなければ電源を切る。これにより無駄なスイッチ切替を減らすことができ、アクティブ時のリーク電流削減が実現する。

Self-Timed Cut-Off 法を適用する際に重要なのが最適な待ち

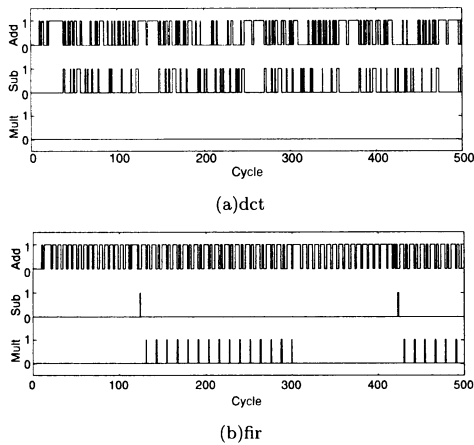


図3 各サンプルプログラムの Enable パターン

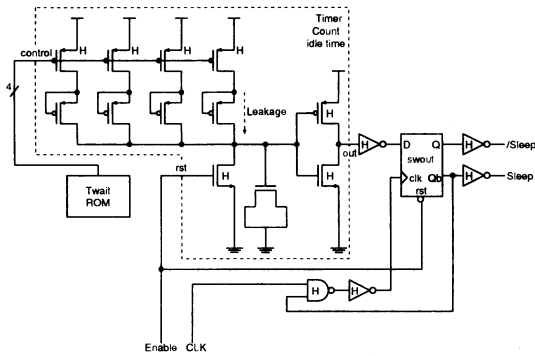


図4 Self-Timed Sleep Timer 回路

ある。収束前にスイッチがONとなった場合、消費エネルギーは微減または増加となる。

図2はスリープ時間とリークエネルギーの関係を示している。 E_{on} 、 E_{off} はそれぞれ充放電による動的エネルギー消費を表わす。 d は回路ブロックにスイッチを使わなかった場合のリークエネルギーである。 T_a だけスリープ状態が続いた後に回路が使用されて電源投入された a の場合、ストールによる実行時間増加や動的電力消費でリークエネルギーが d より増加してしまう。一方、スリープ時間が T_c である c の場合はリークエネルギーが小さくなっていることがわかる。 b は d のリークエネルギーと同じ場合であり、 T_b よりスリープ時間が長ければリークエネルギーが小さくなり、短ければ大きくなることを意味する。[3]ではこの T_b を Leakage equivalent time と定義している。

以上より、アクティブ時のリーク低減のためにはスリープ時間が Leakage equivalent time より大きくなるようなスイッチ切替が必要になる。図3はプロセッサに2つのサンプルプログラムを実行させた時の主な演算器の Enable パターンである。この図から回路ブロックの使用状況はキャッシュメモリのヒットミスのように、一度使用された回路ブロックは近いうちにまた使用される可能性が高く、Block enableをスイッチの入力にするのはいい方法ではない。しかし、この時間的局所性はある

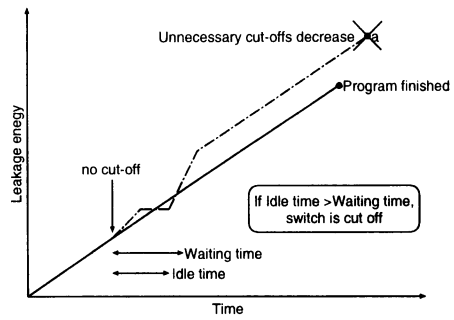


図5 Self-Timed Cut-Off 法による不要な電源切替の減少 (点線はタイマー無しで、実線は Self-Timed Cut-Off 法。)

程度使用されなければしばらく使用されないと言い替えることができる。したがって、待ち時間 (T_{wait} :Waiting time) だけ使用されていない状態が続けばスリープ状態に移行する図4の Self-Timed Sleep Timer 回路をスイッチに設けて、不要なスイッチの切替を減らしリークエネルギー削減を実現するのが Self-Timed Cut-Off 法である。タイマーは OFF 状態のトランジスタと容量の RC 回路で構成され、レジスタによって OFF 状態のトランジスタの数を決定する。例えば、図5では Block Enable がスイッチにつながっている場合は点線のように消費エネルギーが増加してしまうが、Self-Timed Cut-Off 法ではアイドル時間が待ち時間以下なので、実線のように電源切替が起らず消費エネルギー増加を防いでいる。使用状況に局所性がある場合はこの方法によりスリープ時間が Leakage Equivalent time 以下の電源切替を減少させることができる。実験によって、CPU 内にある加算器のリークエネルギーを約 75%削減したという結果が出ている [3]。

4. 待ち時間動的最適化によるリーク電流削減手法

Self-Timed Cut-Off 法を適用する際に重要であるのが、回路ブロックに最適な待ち時間 T_{wait} を設定することである。この値が大きすぎると、ほとんどスイッチの切替が起らず、大きな削減効果が望めない。逆に小さすぎると、スイッチを切った直後にスイッチを入れなければならない可能性が増し、電源の切替による動的エネルギー消費と実行時間増加が大きなオーバーヘッドとなる。[3]では Enable パターンから統計的に最適な T_{wait} を一意に決定する手法を取っている。ASIC などの特定用途のモジュールにおいては、使用状況が一定の傾向を持つことが予想されるため、最適な T_{wait} を一定値とすることができる。しかし、CPU 内のモジュールにおいては実行させるアプリケーションが様々なので、使用状況があるサイクル毎に大きく変化し、最適な T_{wait} も変動する。このような時に T_{wait} を一定値としてしまうと、予期しない高頻度の使用状況になった場合、消費エネルギー増加を招いてしまう。その解決策として、 T_{wait} 動的最適化によるリーク電流削減を提案する。3節で触れた Leakage equivalent time (T_{eqv}) を T_{wait} の動的最適化に利

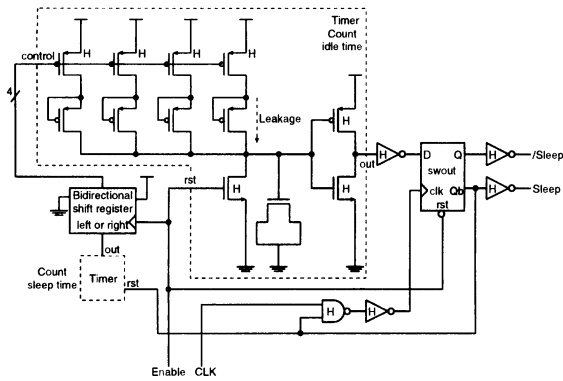


図 6 Cut-off timer with dynamic waiting time 回路 (シフトレジスタが 4 ビットの場合)

表 2 タイマーの主な仕様 (90nm プロセス, 5 ビットシフトレジスタ, active は 500MHz 動作時の消費電力)

タイマー	#Tr	leakage [μ W]	active [μ W]
図 4	200	0.056	0.304
図 6	262	0.073	0.326

用する。以下に制御方法を示す。

(1) スリープ時間が T_{eqv} 以下かつ T_{wait} が最大値未満であれば, T_{wait} を増加し, 電源切替を減らす。

(2) スリープ時間が T_{eqv} 以上かつ T_{wait} が最小値以上であれば, T_{wait} を減少し, 電源切替を増やす。

この制御方法によって, 様々な負荷においてもリーク電流削減の実現が可能になる。

4.1 Cut-off timer with dynamic waiting time 回路

図 6 は待ち時間 T_{wait} を動的に最適化するタイマー (Cut-off timer with dynamic waiting time 回路) である。OFF 状態のトランジスタの個数を決めるレジスタを双方向シフトレジスタとし, 一方からは 0 を, 他方からは 1 を入力し OFF 状態のトランジスタの個数を変える。スリープ時間をカウントするためのタイマーをもう 1 つ追加しており, T_{eqv} 以上で 1 を出力する。 T_{eqv} は回路によって一意に決まる値であるので, レジスタは不要である。スリープ信号のグリッチを防ぐための swout に対象回路ブロックが ON のときのみクロックが供給されるので, 動的電力は少ない。表 2 は 90nm プロセスにおける Cut-off timer with dynamic waiting time 回路 (図 6) と Self-Timed Sleep Timer 回路 (図 4) の主な仕様である。ただし, シフトレジスタ数のビット数は 5 である。カウント用である OFF 状態のトランジスタ以外は high- V_{th} のトランジスタを用いているのでリークは小さくなっている。タイマーで使用されるトランジスタの大部分がフリップフロップで占めており, この個数を変えることでタイマーの面積とリークがほぼ決まる。

5. 提案手法の評価

回路の使用状況が刻々と変化する場合でも, 提案手法がリーク削減可能であることを確かめるには, 大きなサイクル数のシ

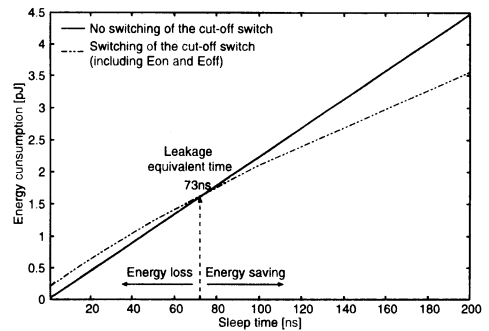


図 7 スリープ時間と消費エネルギーの関係 (dual- V_{th} 32bit 乗算器)

ミュレーションが必要になる。これを回路シミュレーションで行なうのでは, 膨大な時間を要する。そこで本節では C 言語プログラムを用いた以下の手順で提案手法の評価を行なう。

(1) 回路ブロックのリーク電力 P_{leak} と図 8 のようなスリープ時間と消費エネルギー比の関数 $R_e(t_{sleep})$ を回路シミュレーションによって求めておく。

(2) Enable パターンを入力とする C 言語プログラムで, スイッチが ON, OFF である時間 t_{on} , t_{off} と平均スリープ時間 t_{ave} を算出し, 式 (1) を計算して, リークエネルギー E_{leak} を求める。

$$E_{leak} = P_{leak} \times (t_{on} + R_e(t_{ave})t \times t_{off}) \quad (1)$$

リークエネルギーと実行時間も考慮した ED 積を評価指標として用いる。動作周波数は 500MHz とし, スイッチ切替のアプローチとして以下の 4 つを比較する。

No cut-off switch スイッチを使用しない。

Only cut-off switch イネーブル信号が直接スイッチに接続。Block enable が 0 のとき, 電源遮断。

Cut-off switch with constant twait timer 一定値の T_{wait} をもつタイマーを使用。 $T_{wait} = 4, 8, 16, 32, 64, 128$ ns で評価。

Cut-off switch with dynamic twait timer 動的に待ち時間 T_{wait} を変化させるタイマーを使用。 T_{wait} はスリープ時間が T_{eqv} 以上か以下かによって 4, 8, 16, 32, 64ns で変化する。

アクティブ時のリーク電流削減においては wake-up 時間が短いスイッチを使用するのが理想的である。ZSCCMOS は SC-CMOS や BGMOS に比べて約 10 倍高速であるが, 今後ゲートリークが大きくなると ZSCCMOS によるリーク電流削減効果が薄れてくる。したがって, スイッチとして SCCMOS や BGMOS を想定する。

5.1 対象回路

Self-Timed Cut-Off 法の有効性が発揮されるのはスイッチ切替が比較的発生する場合である。これは回路ブロックの負荷が極端に低い場合はタイマーが不要であるし, 逆に極端に高い場合はスイッチが不要であるからである。したがって, プロセッサの実行ユニットの演算器が主なターゲットとなり, 表 1 にある 64bit 加算器, 32bit 乗算器が対象候補となる。しかし, 回

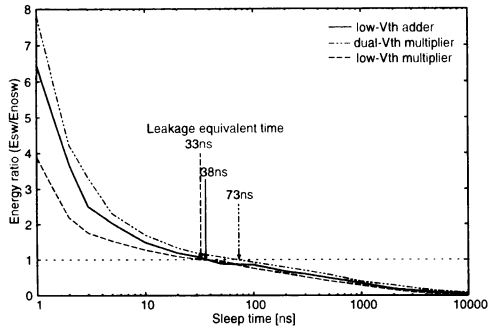


図 8 スリープ時間と消費エネルギー比の関係 (E_{sw} , E_{nosw} はそれぞれスイッチ切替をした、しなかった場合の消費エネルギー)

路が ON の場合に表 2 の active 時電力が消費され, dual- V_{th} 64bit 加算器はタイマーの消費エネルギーによるオーバーヘッドが大きくなるので, 残りの 3 つを対象とする。動作周期が 2ns なので wake-up 時間によるストールは 1 サイクルとする。

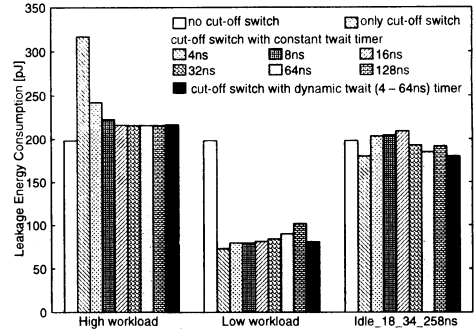
図 7 は dual- V_{th} 32bit 乗算器のスリープ時間と消費エネルギーの関係である。スイッチを切らなかった場合の消費エネルギー E_{nosw} とスイッチを OFF にしてスリープ時間後にスイッチを ON にした場合の消費エネルギー E_{sw} についてプロットしている。スリープ時間が 73ns 以上でスイッチを切り替えた方が消費エネルギーが小さくなるので, $T_{eqv} = 73ns$ となる。図 8 は各対象回路のスリープ時間と消費エネルギー比 (E_{sw}/E_{nosw}) の関係である。動的電力に比べてリーク電流が大きいものほど, T_{eqv} やスリープ時間が T_{eqv} 以下の時の消費エネルギー増加が小さくなっている。

5.2 Enable パターン

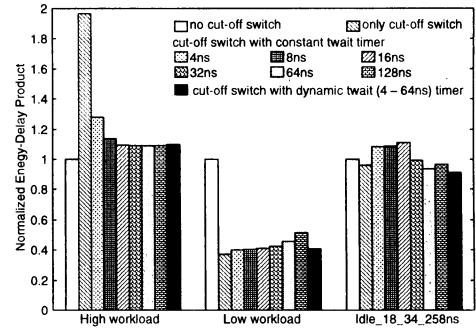
CPU 上で様々なアプリケーションが実行されたり, タスクが切り替わっている状況を表わすには, あるサイクル毎に傾向が変わる Enable パターンが必要になる。図 3 の Enable パターンを見ると, ループが単純な場合は Enable パターンは単純な周期性を持つし, ループがネスト構造で複雑になっている場合はランダムに近い性質をもつ。また, 多数のループが存在し, 特定のループ内のみで使われる演算器がある場合, その演算器の Enable パターンは局所性を持つ。したがって, Enable パターンを局所性モデル, ランダムモデル, 周期性モデルの 3 つの組合せで表現するのが適切であると考えられる。以上から, Enable パターンは 3 つの傾向を 5000 サイクル毎にランダムで切り替えるものとし, 高負荷, 低負荷の 2 つを 300000 サイクル実行させる。また, 負荷を明確に変動させるために, アイドル時間が 18ns, 34ns, 258ns の周期的パターンがそれぞれ 100000 サイクルある場合 (Idle_18_34_258ns と呼ぶ) も評価する。

6. 評価結果

図 9, 10, 11 はそれぞれ low- V_{th} 64bit 加算器, dual- V_{th} 32bit 乗算器, low- V_{th} 32bit 乗算器のリークエネルギーとスイッチを使用しない場合の ED 積を 1 として正規化したもので

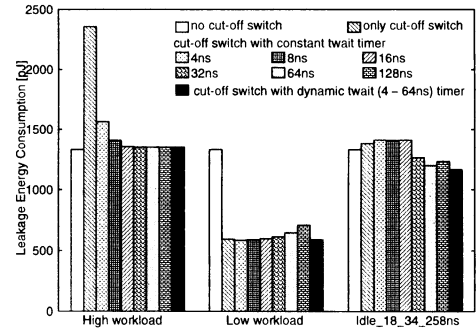


(a) 消費エネルギー

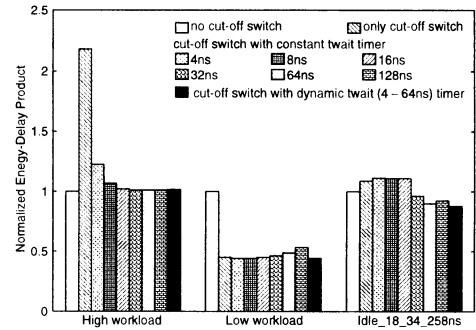


(b) 正規化された ED 積

図 9 low- V_{th} 64bit 加算器

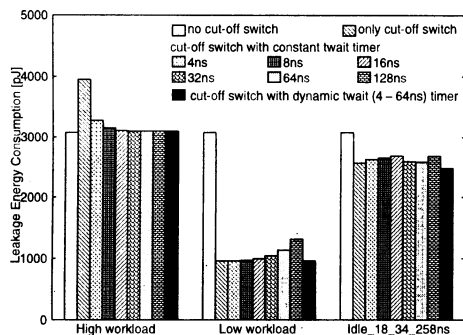


(a) 消費エネルギー

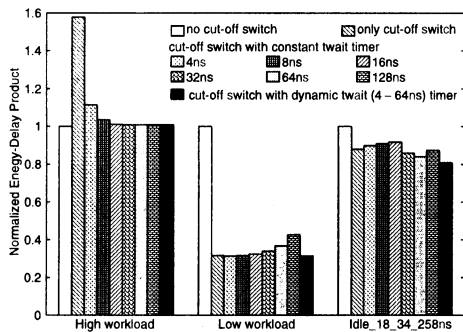


(b) 正規化された ED 積

図 10 dual- V_{th} 32bit 乗算器



(a) 消費エネルギー



(b) 正規化された ED 積

図 11 low- V_{th} 32bit 乗算器

ある。

まず、各スイッチ切替アプローチについて考察する。高負荷においてはスイッチだけの場合や T_{wait} が小さい場合でリークエネルギー、ED 積ともに大きくなってしまふ。 T_{wait} が大きい時でもスイッチを使わない時よりも大きくなっているが、これはタイマーのリークエネルギーのためである。逆に低負荷においては T_{wait} が大きい場合で、リークエネルギー削減効果が減少する。Idle_18_34_258ns では $T_{wait} = 8$, 16ns で大きくなっている。これはアイドル時間が 18ns の時にそれぞれスリープ時間が 10ns, 2ns と短くなってしまふからである。アイドル時間が 34ns の時には $T_{wait} = 32$ ns でスリープ時間が 2ns と短くなるが、アイドル時間が 18ns の時にスイッチ切替が起こらないので、前の 2 つよりはリークエネルギーが小さい。一方、提案手法のリークエネルギーは待ち時間を一定値にした場合の最適値に比べて最悪で 0.8% 増加、最良で 5.6% 減少となっている。

次に、対象回路によるリークエネルギー削減効果の変化について考察する。low- V_{th} 64bit 加算器では回路のリーク電力に比べて、タイマーの消費電力が無視できないので、低頻度や Idle_18_34_258ns におけるスイッチだけの結果や高頻度におけるスイッチ無しの結果が良くなっている。32bit 乗算器ではリーク電力の大きい low- V_{th} のもので各アプローチの差が減少している。これはスイッチ切替による動的消費エネルギーによるオーバーヘッドがリーク電力増加によって相対的に小さくなったからである。今後、動的電力がリーク電力に比べて無視でき

るような状況になった場合はスイッチだけを用いるのが最も効果的になると思われる。したがって Self-Timed Cut-Off 法が効果を発揮するのはリーク電力が大きい、動的電力も無視することができない場合である。

7. 結 論

本稿では Self-Timed Cut-Off 法において待ち時間を動的に最適化することを提案し、回路ブロックの使用状況が刻々と変化する状況においてもリーク電流削減を実現した。待ち時間を動的に最適化するタイマーを少ない面積、消費電力で作成し、対象回路はそれらが無視できるものとなる。64bit 加算器や 32bit 乗算器を用いた評価を行ない、待ち時間を一定値にした従来手法の最適値に比べて、リークエネルギーが最悪で 0.8% 増加、最良で 5.6% 減少となった。つまり、提案手法では統計的に最適な待ち時間を求める必要も無しに、常に従来手法の最適値と同等のリーク電流削減が実現できる。待ち時間を一定値にした場合には (待ち時間+1 サイクル) の周期を持つ Enable パターンでオーバーヘッドが大きくなるが、提案手法ではそれを回避することができ、ロバストなリーク電流削減が可能である。

また、Self-Timed Cut-Off 法はリーク電力が大きい、動的電力が無視できない状況で最も効果的である。ITRS ロードマップで考えると 90~65nm プロセス辺りである。今後は類似の使用状況を持った複数回路ブロックのタイマーを共有することで、小さい回路にも適用できるかどうかを検討したい。

文 献

- [1] T. Inukai et al. "Boosted Gate MOS (BGMOS): device/circuit cooperation scheme to achieve leakage-free gigascale integration". In *CICC 2000*, pp. 409-412, 2000.
- [2] J. Schutz and C. Webb. "A scalable X86 CPU design for 90nm process". In *ISSCC*, Vol. 47, 2004.
- [3] J. H. Choi and T. Sakurai. "Statistical Leakage Current Reduction by Self-Timed Cut-Off Scheme for High Leakage Environments". In *CICC*, 2003.
- [4] S. Mutoh et al. "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS". In *IEEE J. Solid State Circuits*, Vol. 30, pp. 847-854, August 1995.
- [5] H. Kawaguchi, K. Nose, and T. Sakurai. "A Super Cut-off CMOS (SCCMOS) Scheme for 0.5-V Supply Voltage with Picoampere Stand-By Current". In *IEEE J. Solid State Circuits*, Vol. 35, pp. 1498-1501, October 2000.
- [6] K.S. Min and T. Sakurai. "Zigzag Super Cut-off CMOS (ZSCCMOS) block activation with self-adaptive voltage level controller: An alternative to clock-gating scheme in leakage dominant era". In *ISSCC*, 2003.
- [7] M. Drazdziulis and P. Larsson-Edefors. "Evaluation of power cut-off techniques in the presence of gate leakage". In *ISCAS 2004*, pp. 745-748, 2004.
- [8] J. Tschanz, S. Narendra, Y. Ye, B. Bloechel, and S. Borkar. "Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors". In *IEEE J. Solid State Circuits*, Vol. 38, pp. 1838-1845, November 2003.