

カナリア・フリップフロップを利用する DVS 方式の改良

佐藤 寿倫[†] 国武 勇次[‡]

[†]九州大学 システム LSI 研究センター 〒814-0001 福岡市早良区百道浜 3-8-33-3F

[‡]九州工業大学大学院 情報工学研究科 〒820-8502 飯塚市川津 680-4

E-mail: [†]toshinori.sato@computer.org, [‡]y.kunitake@klab.ai.kyutech.ac.jp

あらまし 微細化の進展により顕在化しつつあるパラメタばらつきを抑制するために、カナリア・フリップフロップを提案している。これまで、入力値のばらつきを利用して、電源電圧マージンの削減による消費電力削減方式を検討してきた。本稿では、カナリア FF を利用する DVS システムにおける課題を検討する。特に電源電圧の振動問題を扱い、シミュレーションの結果、性能への悪影響が取り除かれることを確認出来ている。

キーワード パラメタばらつき、省電力、カナリア FF

Improving Energy-efficiency of Canary-based DVS System

Toshinori SATO[†] and Yuji KUNITAKE[‡]

[†]System LSI Research Center, Kyushu University

[‡]Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology

E-mail: [†]toshinori.sato@computer.org, [‡]y.kunitake@klab.ai.kyutech.ac.jp

Abstract In deep submicron technologies, parameter variations have become a serious problem on LSI design. We proposed canary flip-flops to attack the problem, and have investigated its application on power reduction by exploiting input value variations. This paper considers some critical issues on the canary FFs and the DVS systems utilizing the canary FF. Especially, this paper focuses on the problem of power supply voltage and evaluate a solution. Detailed simulation results show that performance penalty is effectively eliminated.

Keyword Parameter variations, low power, canary flip-flop

1. はじめに

半導体製造プロセスの複雑度が増すにしたがって、素子特性のばらつきのために、従来行われてきた最悪ケースに配慮した LSI 設計は困難になりつつある[13, 15]。なぜなら、最悪ケースを考慮するための設計マージンの確保が困難なためである。ロバストな設計を実現するために、設計者には製造容易性を考慮した設計が求められている。有望な設計手法のひとつとして、我々は典型的ケース指向設計を検討している。この手法では、最悪ケースが現実には生じるのは極めて稀である、という観察結果を利用している。そこでは設計者は、最悪ケースに煩わされるのではなく、典型的ケースに集中することが出来る。最悪ケースに配慮する必要がなくなれば、設計制約は大きく緩和されるので、設計が容易になるとともにその期間も短縮されることが期待される。

典型的ケース指向設計を可能にするために、我々はカナリア・フリップフロップ (FF) を提案し[12]、設計マージンを取り除くことによって消費電力を削減する方法を評価してきた[12,15]。これまでのところ、加

算器への入力値のばらつきを利用することで、プロセス全体で概ね 10% の消費電力を削減できることが確認できている[12,15]。しかしカナリア FF の実装には、克服すべき課題が残っている。本稿ではそれらの課題を考察し、特に DVS システムにおける問題の解決策についてシミュレーションにより評価する。

本稿は以下の構成をとる。2 節で関連研究をまとめる。3 節でカナリア FF を説明し、4 節と 5 節でその実装における課題を検討する。6 節で評価環境を紹介した後、シミュレーション結果を示す。7 節でまとめとする。

2. 関連研究

Razor[4, 6]は電源電圧におけるマージンを取り除いて消費電力を削減する。タイミングエラーの発生頻度に応じて DVS 制御を実施する。エラー頻度が小さい間は電源電圧を下げる余地がある。逆に頻度が高いと電圧を上げる必要がある。閾値となるエラー頻度が予め設定され、DVS システムはそれを維持することに努める。タイミングエラーを動的に検出するために、図 1

に示す Razor フリップフロップ(FF)が提案されている。タイミング・クリティカルな FF (メイン FF) の全てにシャドウ FF を用意する。シャドウ FF にはメイン FF に比べて遅いクロック(遅延クロック)が供給される。メイン FF とシャドウ FF に保持されている値が異なると、タイミングエラーを検出したことになる。Razor の持つ難しさのひとつに、シャドウ FF が常に正しい値を保持することをどのようにして保証するかという問題がある。ホールドタイムの制約を満足するように遅延クロックを設計しなければならない。さもないと、所謂ショートパス問題を生じる[6]。遅延バッファを挿入すればこの問題を解決できる[4]が、セットアップタイムの制約を厳しくしてしまい、タイミングマージンを減じることになる。

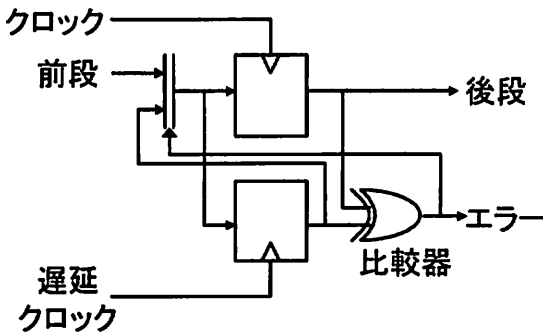


図1: Razor FF

iRoC Technologies[5]はシャドウ FF をソフトエラー検出の目的で利用する。二通りの実装法が検討されている。ひとつは Razor FF と非常に似ており、遅延クロックを必要とする。他方は遅延クロックを必要としないが、入力が遅延されてシャドウ FF に保持される。メイン FF とシャドウ FF に保持されている値が不一致となると、ソフトエラーが検出されたことになる。上述の遅延を調整することで、検出される過渡パルスの最大幅を変更可能である。

NEC[11]はシャドウ FF を経年劣化によるタイミング故障を予測する目的で利用する。全組合せ回路部は複製され、故障箇所はその複製に切り替えられる。故障を予測するために、故障予測 FF が提案されており、シャドウ FF を利用している。前段の組合せ回路部とシャドウ FF との間には遅延が挿入されており、メイン FF がタイミング制約違反に遭遇しない場合でも、シャドウ FF が違反に遭遇し得る。つまり、メイン FF とシャドウ FF に保持されている値を比較することで、経年劣化によるパス遅延の増大を検出可能である。

インテル[1]はシャドウ FF をパラメタばらつきに起因するタイミングエラーを検出する目的で利用する。

シャドウ FF はメイン FF への入力を共有するのではなく、メイン FF のマスターラッチを通過した値を保持する。したがって、メイン FF よりもシャドウ FF の方がタイミング制約が厳しい。両 FF のマスターラッチが閉じたときにそれらの値が一致しないと、タイミングエラーが検出されたことになる。

次節で説明するカナリア FF[12, 15]は、上記の三つの方法[1, 5, 11]とは独立に検討され、Razor FF[4, 6]の一改良として提案された。

Liら[10]は Razor FF のロバスト性を向上させた回路を提案し、スーパースカラプロセッサを設計して評価している。

Calhoun ら[3]は、同じ名前ではあるが次節のそれとは原理と目的が全く異なるカナリア FF を提案している。それは、他の FF と比較して高い電源電圧であってもエラーを生じ易いように設計され、スタンバイ時の電源電圧決定に利用される。カナリア FF がエラーを生じるまで電源電圧を低下させることでリーク電流を削減し、その結果消費電力も低減させる。

3. カナリア・フリップフロップ

Razor は 設計マージンを取り除くことが出来る非常に優れた方式であるが、その回路実装には改善の余地が残されている。我々は改善方式のひとつとして、図2に示す回路を提案している[12, 15]。

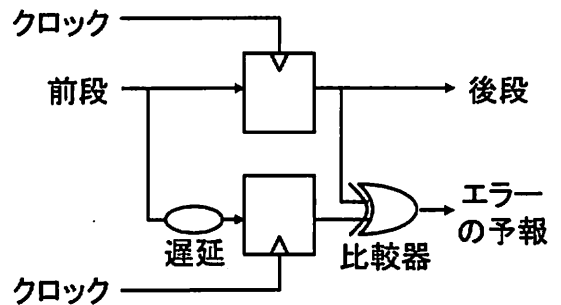


図2: カナリア FF

この回路では、遅延クロックの代わりに遅延素子が利用される。カナリア FF は炭鉱のカナリアとして働き、今まさに生じようとしているタイミングエラーを検出する目的で利用される。遅延素子が挿入されているので、カナリア FF はメイン FF よりもタイミング制約が厳しい。タイミング制約を徐々に厳しくしていくと、必ずカナリア FF が先に前段の回路におけるタイミングエラーに遭遇する。メイン FF とカナリア FF に保持されている値が異なることが、タイミングエラーの予報に相当する。エラーが予報されると、周波数を下げたり電源電圧を上げたり等の対処が施され、メイ

ン FF でのエラーを予防する[15]。カナリア FF の使用には以下のメリットがある。

遅延クロックの廃止：単一単相クロックの採用によって、クロック分配の設計が単純化される。加えて、Razor FF での設計を複雑化しているショートパス問題[6]を取り除くことができる。

タイミングエラーの抑制：上述したように Razor FF と異なり、カナリア FF はメイン FF でのタイミングエラーを予防する。このため、いかなる回復機構も不要である。加えて、メイン FF の入力にあるマルチプレクサも不要なため、Razor FF と比較してタイミング制約が緩和される。

素子ばらつきに対する柔軟性：遅延素子が挿入されているため、どのような素子ばらつきを生じてても、メイン FF よりもカナリア FF が先にタイミングエラーに遭遇する。つまり、大きな素子ばらつき下における動作が保障されている。

DVS システムは、入力値ばらつきによるバス遅延の動作時ばらつきを積極的に利用して電源電圧を降圧する。予め定められたインターバルの間にタイミングエラーが予報されなければ、ひとつ下のステップまで降圧する。タイミングエラーが予報された場合には、インターバルの途中であっても直ちに昇圧する。この昇圧ポリシーには、ひとつ上のステップへ移動するか、あるいは過渡モードへ移動するかの選択肢が存在する。前者を採用する場合をステップ法と呼び、後者を採用する場合をリセット法と呼ぶ。回路遅延は大幅に変動しないと予想されるのでステップ法で問題なく動作すると考えられるが、大幅な変動に対しても動作を保障するためにはリセット法の採用が必要である。

4. 回路上の問題

カナリア FF の実装においては解決すべき問題がある。本節ではこれまでに認識できている問題を示し、それらへの対応を検討する。

4.1. メタスタビリティの問題

シャドウ FF のタイミング制約は常に満足されるわけではないので、それはメタステーブルの状態に成り得る。メタスタビリティは、回路遅延を増大させるだけでなく、誤動作を引き起こしかねない。Razor では、メタスタビリティを解消して常に正しい値を必要は無く、メタステーブル状態を検出し、タイミングエラーと同様に扱えば十分である。カナリア FF でも事情は同じである。Razor で用いられているメタスタビリティ検出回路[6]が使用可能である。ところで、Razor ではメイン FF がメタステーブル状態と成り得るのに

対し、カナリア FF ではメイン FF は常に正しい値を保持することが期待されている。したがって、カナリア FF では時間を要するメタスタビリティ検出方法を選択することも可能である。

4.2. メイン FF でのタイミングエラーの問題

カナリア FF では、メイン FF が常に正しいことが期待されている。しかし、原理的にはメイン FF がタイミングエラーを生じないことを 100% は保証出来ない。クリティカルパス遅延が投機的に違反されると、二つの FF が同時にエラーとなる可能性があり、この場合には比較器の結果は一致しエラーを予報出来ない。しかしタイミング制約が急激に変化しないという条件化では、そのようなことは発生しないと期待できる。

4.3. 遅延素子における電圧依存性の問題

上述のようにカナリア FF において遅延素子は重要な役割を果たしている。何故なら、その遅延量がカナリア FF のロバスト性を規定するからである。問題となるのが、この遅延量が電源電圧に依存して変化することである。幸いなことに電源電圧が下がるにしたがって遅延量は増大するので、投機が積極的になればなるほどタイミングエラー予測は保守的になる。したがって、遅延量の電源電圧依存性には安全な方向の性質がある。

4.4. 消費電力の問題

カナリア FF はスキャン FF を流用して実現可能[15]なので回路規模のオーバーヘッドは非常に小さいが、シャドウ FF の消費する電力は無視出来ないかも知れない。幸いなことに Razor の報告では、シャドウ FF と遅延バッファが消費する電力はチップ全体の 3% に過ぎない[4]。カナリア FF でも同様の傾向となると期待できる。

5. DVS システムの問題

カナリア FF を利用する DVS システムにおいても解決すべき問題がある。本節ではこれまでに認識できている問題を示し、それらへの対応を検討する。

5.1. エラー予報の収集における問題

カナリア FF を利用する DVS システムでは、各カナリア FF の発したエラー予報を収集する必要がある。ひとつでもエラー予報が発せられると、プロセッサ全体として電源電圧あるいは動作周波数を変更する必要があるからである。そのためのネットワークはチップ面積を大いに消費する可能性がある。また、エラー収集に要するレイテンシがサイクルタイムに与える影響

も懸念される。前者に対しては、スキャン回路を利用して面積を削減する方法を検討中である。後者に対しては、4.1 節で述べたようなレイテンシの大きな方法も利用可能であると考えている。

5.2. 電源電圧における振動の問題

タイミングエラーを生じる電源電圧に近づくと、しばしばエラー予報が発せられることになる。タイミングエラーが予報されると電源電圧を昇圧するが、3 節で説明したインターバルの経過後に再び降圧される。その結果降圧と昇圧を繰り返すことになり、電源電圧に図 3 に示すような振動現象が観測される。電圧を変える度にオーバヘッドを被るので、この振動は性能の点でも消費電力の点でも好ましくない。

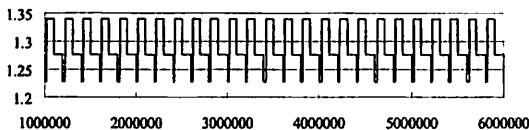


図 3: 電源電圧の振動 (164.gzip)

5.2.1. 振動抑制法

振動を抑制するためにカウンターを利用することを考える。まず、タイミングエラーが予報された電源電圧を記憶する。次回タイミングエラーが予報された電圧が、記憶されている前回の電圧と同じであれば、カウンターをインクリメントする。異なる場合にはカウンターをリセットする。カウンターの値が予め決められた閾値を越えた場合には振動していると見做し、別に予め定められた期間降圧を停止する。降圧を停止する期間の決定には、電源電圧の変更に要するオーバヘッド時間を考慮する。この期間の経過後は、DVS 動作を再開する。

6. 評価

6.1. 環境

SimpleScalar ツールセット[2]を用いたシミュレーションで評価する。命令セットには Alpha ISA を選択する。プロセッサ構成は表 1 の通りである。ベンチマークには SPEC2000 の整数系プログラムから選ばれた 6 つである。なお、最初の 10 億命令をスキップし、続く 20 億命令を対象とする。

電源電圧を変更するインターバルとして、100K、1M、10M サイクルを評価する。電源電圧の変更には 10usec を要する[7]と仮定した。定常モードでは周波数は 2GHz で一定とする。降圧を停止する閾値には、2 と 8 を評価する。降圧を停止する期間は、電圧変更によるオーバヘッドが 0.1%となるように決定した。

表 1: プロセッサ構成

Clock frequency	2 GHz
Fetch width	8 instructions
L1 instruction cache	16K, 2 way, 1 cycle
Branch predictor	gshare + bimodal
Gshare predictor	4K entries, 12 histories
Bimodal predictor	4K entries
Branch target buffer	1K sets, 4 way
Dispatch width	4 instructions
Instruction window size	128 entries
Issue width	4 instructions
Integer ALUs	4 units
Integer multipliers	2 units
Floating ALUs	1 unit
Floating multipliers	1 unit
L1 data cache ports	2 ports
L1 data cache	16K, 4 way, 2 cycle
Unified L2 cache	8M, 8 way, 10 cycles
Memory	Infinite, 100 cycles
Commit width	8 instructions

カナリア方式は整数加算器のみに採用する。パイプラインの歩留まりは主に実行ステージにおけるタイミングエラーで決まる[10]からである。電源電圧および動作周波数を変化させた時の加算器におけるタイミングエラー発生率として、桁上げ選択加算器をゲートレベルシミュレーションして得られた値を用いる。表 2 に示すベンチマーク M の電源電圧[8]を使用して、シミュレーション結果から電源電圧とタイミングエラーの関係の求められている。電源電圧とタイミングエラーの関係を図 4 に示す。

表 2: 周波数 - 電源電圧

F(GHz)	2.1	1.8	1.6	1.4
Vdd(V)	1.340	1.276	1.228	1.180
F(GHz)	1.2	1.0	0.8	0.6
Vdd(V)	1.132	1.084	1.036	0.988

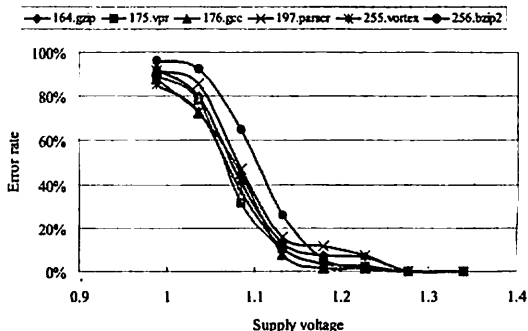


図 4: タイミングエラー率

上述のように、各電源電圧および動作周波数において、確率的にタイミングエラーが生じると仮定している。実際の入力値に応じたタイミングエラーを考慮し

ているわけではないことに注意されたい。ゲートレベルシミュレーションの結果をアーキテクチャレベルシミュレーションで利用する評価環境[9]を構築したが、ゲートレベルシミュレーションの結果を再現出来ない不具合が見つかったため、今回の方法で評価している。タイミングエラーのみを考慮して建設的タイミング違反方式を評価した場合には、入力を考慮する場合と確率的な場合とで、結果に有意差は無いことがわかっている[9]ので深刻な問題は無いと考えている。上述の不具合が発生しないCo-Simulatio環境[14]を構築して、将来入力値の影響を評価する予定である。

6.2. 結果

図5に選択された電源電圧と、実行時間に対するそれらの割合を示す。8ステップの電圧から実際には3ステップしか選択されず、1.276Vが大部分を占める結果となった。リセット法とステップ法とを比較すると、後者がより頻繁に低い電圧を選択している。リセット法が降圧停止状態に入るまでには多くの時間を要するので、容易に予想できる結果である。続いてインターバルの影響については、リセット法でのみインターバルが短いほどより頻繁に低い電圧を選択する現象が観察される。

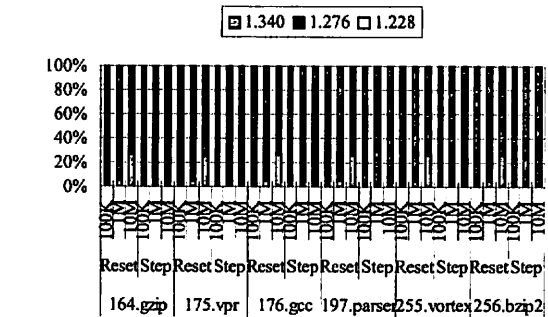


図 5: 電源電圧の分布 (閾値=2)

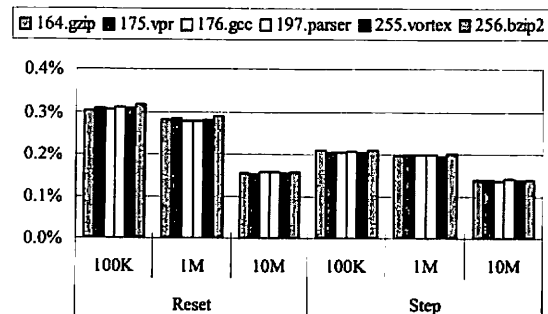


図 6: 実行サイクルの増加 (閾値=2)

図6に実行サイクルへの影響を示す。不要な降圧と昇圧が取り除かれた結果、想定されたオーバーヘッド(0.1%)よりは大きいですが、性能への悪影響は無視できる程度にまで改善できている。

図7に消費エネルギーの削減率を示す。リセット法では、選択された電源電圧の分布にしたがう様に、インターバルが小さいほどエネルギー削減効果が大きい。ステップ法ではインターバルの影響は観察されない。不要な降圧と昇圧を取り除いたにも関わらず、エネルギー削減効果は以前の結果[15]と大差が無い。これは、電源電圧の選択が保守的となり、より低い電圧を選択出来なくなったためと考えられる。

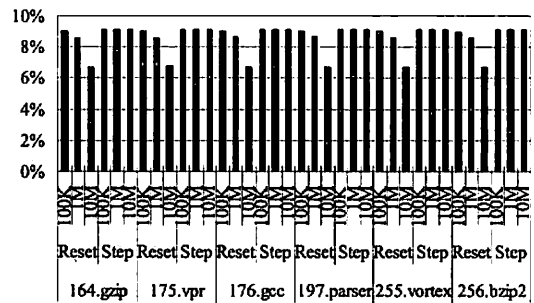


図 7: 消費エネルギーの削減 (閾値=2)

図8~10に閾値が8の場合の結果を示す。閾値に至るまでのサイクル数が大きいため、閾値が2の場合よりも振動が長期に続き性能や消費エネルギーには好ましくないと予想されたが、実際には大きな違いは見られなかった。閾値に至るまでの実行時間が全体に占める割合が小さいためであると予想される。

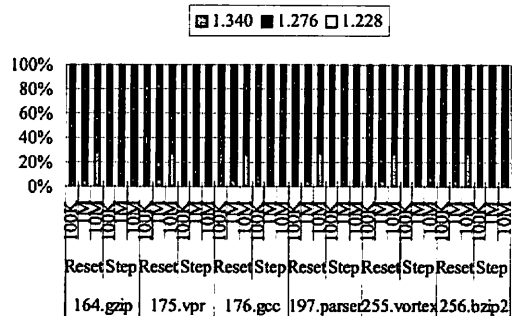


図 8: 電源電圧の分布 (閾値=8)

7. まとめ

カナリア FF を利用する DVS システムについて、実装に向けての課題を検討した。特に電源電圧の振動を抑制する方式を評価した。性能に対するオーバーヘッド

□ 164.gzip ■ 175.vpr □ 176.gcc □ 197.parser ■ 255.vortex □ 256.bzp2

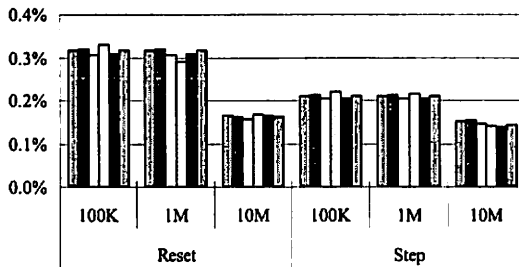


図 9: 実行サイクルの増加 (閾値=8)

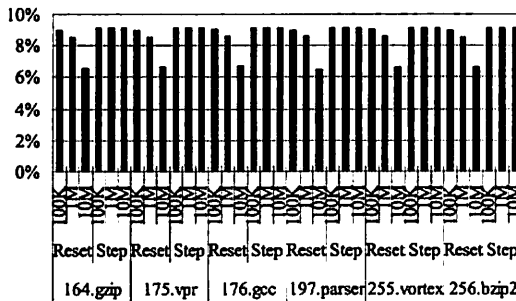


図 10: 消費エネルギーの削減 (閾値=8)

を大きく抑制することが出来たが、消費エネルギーの削減に関しては大きな改善は見られなかった。電源電圧の選択が保守的になることが理由と考えられるため、積極的な降圧が可能な方式の検討が必要である。

謝辞

本研究の一部は、科学技術振興機構・戦略的創造研究推進事業の CREST プロジェクト「情報システムの超低消費電力化を目指した技術革新と統合化技術」の支援によるものである。

文 献

- [1] M. Annavaram, E. Grochowski, and P. Reed, "Implications of Device Timing Variability on Full Chip Timing", 13th International Symposium on High-Performance Computer Architecture, 2007.
- [2] D. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0", ACM SIGARCH Computer Architecture News, Vol. 25, No. 3, 1997.
- [3] B. H. Calhoun and A. P. Chandrakasan, "Standby Power Reduction Using Dynamic Voltage Scaling and Canary Flip-Flop Structures", IEEE Journal of Solid-State Circuits, Vol. 39, No. 9, 2004.
- [4] S. Das, P. Sanjay, D. Roberts, L. S. Lee, D. Blaauw, T. Austin, T. Mudge, and K. Flautner, "A Self-Tuning DVS Processor Using Delay-Error Detection and

Correction", Symposium on VLSI Circuits, 2005.

- [5] E. Dupont, M. Nicolaidas, and P. Rohr, "Embedded Robustness IPs for Transient-Error-Free ICs", IEEE Design & Test of Computers, Vol. 19, No. 3, 2002.
- [6] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation", 36th International Symposium on Microarchitecture, 2003.
- [7] S. Gochman, R. Ronen, I. Anati, A. Berkovits, T. Kurts, A. Naveh, A. Saeed, Z. Sperber, and R. C. Valentine, "The Intel Pentium M Processor: Microarchitecture and Performance", Intel Technology Journal, Vol. 7, No. 2, 2003.
- [8] Intel Corporation, "Intel Pentium M Processor on 90nm Process with 2-MB L2 Cache", Datasheet, 2006.
- [9] Y. Kunitake, A. Chiyonobu, K. Tanaka, and T. Sato, "Challenges in Evaluations for a Typical-Case Design Methodology", 8th International Symposium on Quality Electronic Design, 2007.
- [10] H. Li, Y. Chen, K. Roy, and C.-K. Koh, "SAVS: A Self-Adaptive Variable Supply-Voltage Technique for Process-Tolerant and Power-Efficient Multi-Issue Superscalar Processor Design", 11th Asia and South Pacific Design Automation Conference, 2006.
- [11] T. Nakura, K. Nose, M. Mizuno, "Fine-Grain Redundant Logic Defect-Prediction Flip-Flops", International Solid-State Circuits Conference, 2007.
- [12] T. Sato and Y. Kunitake, "A Simple Flip-Flop Circuit for Typical-Case Designs for DFM", 8th International Symposium on Quality Electronic Design, 2007.
- [13] O. S. Unsal, J. W. Tschanz, K. Bowman, V. De, X. Vera, A. Gonzalez, O. Ergin, "Impact of Parameter Variations on Circuits and Microarchitecture", IEEE Micro, Vol. 26, No. 6, 2006.
- [14] 國武, 千代延, 田中, 佐藤, "タイミング制約違反を利用する設計手法とコ・シミュレーション環境による評価", 情報研報 ARC, 発表予定, 2007.
- [15] 佐藤, "カナリア・フリップフロップを利用する省電力マイクロプロセッサの評価", 第 5 回先進的計算基盤システムシンポジウム, 2007.