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

佐藤, 寿倫
九州大学システムLSI研究センター

出版情報：5th Symposium on Advanced Computing Systems and Infrastructures, SACSIS 2007 || || p227-234, pp.227-234, 2007-05-25. 先進的計算基盤システムシンポジウム
バージョン：
権利関係：

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

佐藤 寿倫 †

半導体技術の微細化が進展するにつれて，従来行われてきた最悪ケースを想定した LSI 設計は困難になると予測されている．何故なら，そのために必要な設計マージンを十分に確保出来ないからである．設計者が最悪ケースに煩わされること無く，典型的ケースに注力可能な設計手法が求められている．そのような典型的ケース指向設計を実現するために，我々はカナリア方式を検討している．加算器において約 30%の消費電力を削減できる可能性をゲートレベルシミュレーションにより確認しているが，マイクロプロセッサ全体での効果は未だ不明である．本稿で我々は，カナリア FF がマイクロプロセッサの省電力化に有効であるかどうかを評価する．シミュレーションにより平均で約 9%のエネルギー削減を確認している．

A Simulation Study of Energy-Efficient Microprocessors Using Canary FFs *

TOSHINORI SATO †

The deep submicron semiconductor technologies will make the worst-case design difficult, since they can not provide design margins that it requires. Research directions should go to typical-case design methodologies, where designers are focusing on typical cases rather than worrying about very rare worst cases. We are investigating canary logic, which we proposed as a promising technique that enables the typical-case design. While we estimated its potential power reduction of 30% in adders at gate-level simulations, its effectiveness in the total microprocessor is still unclear. In this paper, we evaluate how canary FFs reduce power consumed by the total microprocessor and find the potential energy reduction of 9%.

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，素子ばらつきを制御することが困難になりつつある[2, 9, 17]．省電力化に対する要求の高まりとともに電源電圧が降圧され，そのゆらぎが深刻になっている．クロック周波数の上昇に伴い，チップ内における温度ばらつきが増加している．これらの要因による素子特性のばらつきのために，ディープサブミクロン領域の半導体技術において，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．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．

ひとつの有望な解決策は，典型的ケースを指向する設計である．そこでは，最悪ケースではなく典型的ケースに最適な設計が実施される．近年，同様の思想に基づいて，様々な設計手法が研究されている．Razor[5,6]，approximation circuits 方式[12,13]，algorithmic noise tolerance 方式[16]，TEAtime[18]，そして建設的タイミング違反方式[22]などである．我々は Razor に着目し，その改良版であるカナリア方式を検討している[15]．加算器を用いて行った初期評価で 30%を越える消費電力削減を確認できている[15]が，プロセッサ全体での効果は不明であるため，本稿でその効果を評価する．

本稿は以下の構成をとる．2 節で典型的ケース指向設計を説明する．3 節で関連研究を紹介し，特に Razor を詳しく説明する．4 節でカナリア方式を提案する．5 節で評価環境を紹介した後に，6 節でシミュレーション結果を示す．7 節でまとめとする．

2. 典型的ケース指向設計

半導体技術の微細化が一層進展するにつれ，素子ばらつきの問題が深刻化している．ばらつきが大き

---

* 査読者に“canary flip-flop”という名前は既に[3]で使用されていることを教えていただいた．エラーの予報という概念は同じだが，目的と実現法が異なる．不要な混乱を避けるためには別の名前とすることが好ましいが，タイトルに使用されていることから，本稿では「カナリア FF」を使い続けることとした．

† 九州大学システム LSI 研究センター，System LSI Research Center, Kyushu University

くなるに伴い，設計マージンを確保することが困難になっている．マージンがなくなれば素子特性の最悪ケースを想定することは出来ず，従来の設計手法ではLSIの設計が困難な状況に陥ると危惧されている．このような背景に鑑み，設計容易さを向上するための手法が求められている．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[22]．この手法では，最悪ケースが現実に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無くなれば，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において，設計者はひとつの機能に対して同時に二つの回路を実現する．ひとつは性能（例えば動作速度や消費電力）を指向した設計であり（性能指向設計），その際には典型的ケースのみが考慮される．最悪ケースを考慮する必要が無いので，制約が緩和され設計が容易になる．もうひとつは機能を保障する設計である（機能保障設計）．ここでは設計者は，最悪ケースを考慮しなければならないが，性能への配慮からは開放される．機能の保障のみを達成すれば良いので，シンプルな設計が選択可能になり，その検証も容易になる．

LSI チップを構成する機能ブロックの中で，（例えば動作速度の点で）クリティカルなブロックのみが上述の二つの設計を施される．典型的ケース指向設計の概念を図 1 に示す．図において，設計対象は二つの回路として設計されている．それぞれをメイン部，チェック部と呼んでいる．それらの機能は同一であるが，役割と実装を異にしている．メイン部の設計において設計者は，性能を向上するような最適化を施す必要があるが，機能が常に正しいことを保障する必要は無い．つまり，メイン部はエラーを生じ得る．上述した性能指向設計に基づいている．チェック部は，信頼性に欠けるメイン部のセーフティネットとして用意される．メイン部で起こり得るあらゆるエラーを検出しなければならないので，設計対象のLSIチップにおける全ての設計制約を満足することが求められる．つまりチェック部の設計において設計者は最悪ケースを考慮しなければならないわけであるが，性能改善のための最適化を実施する必要は無い．上述の機能保障設計に基づいている．エラーが検出された際には，回路の内部状態を何らかの方法で正常な状態に回復させる．

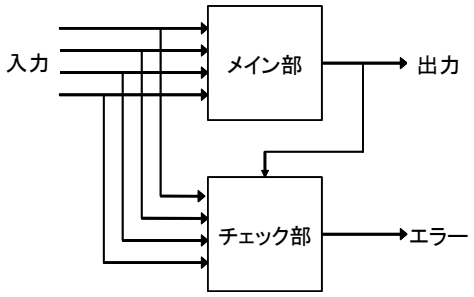

図 1:典型的ケース指向設計

3. 関連研究

典型的ケース指向設計と同様の思想に基づいて，近年様々な研究が行われている．Razor[5,6]，approximation circuits 方式[12,13]，algorithmic noise tolerance 方式[16]，TEAtime[18]，そして建設的タイミング違反方式[22]である．

Approximation circuits 方式[12,13]では，所望の機能を実現するための完全な回路ではなく，それを模倣する簡易版の回路（近似回路）を用意する．近似回路は，完全な回路に比べて高速動作が可能な上，たいていの場合に正しい結果を出力する．したがって，最悪ケースに配慮した設計に比べると，より高速に動作する設計が可能になる．

Algorithmic noise tolerance 方式[16]では，消費エネルギーと性能における限界を定めるために情報理論を応用する．回路レベルやアルゴリズムレベルでノイズに対する耐性を備えさせることで，上記の限界に近い領域での回路動作を実現している．

TEAtime[18]では最悪ケースを模倣する回路（模倣回路）を用意し，その回路が正常動作する範囲で電源電圧を降圧したり動作周波数を上昇させたりする．模倣回路には，クリティカルパスのビット幅を圧縮させたコピーを用いる．その入力には 0/1 のトグルを与え，0→1 と 1→0 のいずれの遷移においても正常動作することを保障する．

建設的タイミング違反方式[22]は，回路遅延の入力ばらつきを利用する．クリティカルパスは常には活性化しないことに着目し，クリティカルパス遅延に違反する周波数あるいは電源電圧で，回路を動作させる．正常動作を保障するために，機能は同等であるが低速に動作させる回路を併せて用意する．

3.1 Razor

Razor [5,6] では，エネルギー利用効率を改善す

る目的で，設計におけるタイミング制約に違反することが許容される．つまり，クリティカルパス遅延で決まる周波数よりも速い速度で，Razor を採用する LSI は動作する．タイミング制約違反（タイミングエラー）を動作時に検出するために，図 2 に示す Razor フリップフロップ(FF)が提案されている．タイミング制約においてクリティカルな FF（メイン FF）の全てにシャドウ FF を用意する．前段からシャドウ FF に至る回路のタイミング制約を満足できるように，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．シャドウ FF は常に正しい値を保持することが期待されているわけである．もしメイン FF とシャドウ FF に保持されている値が異なると，前段からメイン FF に至る回路のタイミングエラーが検出されたことになる．マイクロプロセッサのパイプラインでタイミングエラーが検出された際には，カウンターフローパイプラインに基づく回復機構を利用して，プロセッサ状態を正常に復帰させる[6].

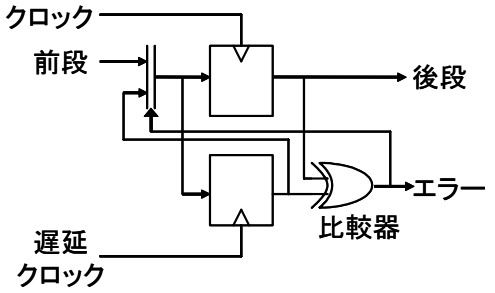

図 2：Razor FF

Razor では，消費電力を削減するために，電源電圧におけるマージンを取り除く．タイミングエラーの発生頻度に応じて電圧を制御する．Razor で用いられる動作時電圧制御システムを図 3 に示す．エラー発生頻度が小さい間は，電源電圧を下げる余地がある．逆に頻度が高いと，電源電圧を上げる必要がある．閾値となるエラー発生頻度(E_{ref})が予め設定され，制御システムは E_{ref} を維持することに努める．決まったインターバルで，エラー発生頻度(E_{sample})と閾値との差($E_{diff} = E_{ref} - E_{sample}$)が計算される．$E_{diff}$ が正の値をとるときには，電圧を降圧する余地がある．逆の場合には昇圧しなければならない．

Razor を採用した設計も依然として困難を伴う．シャドウ FF が正しい値を常に保持することを如何に保障するかという点においてである．前段の遅延が小さいと，1 サイクル前の値をラッチすべき遅延クロックのタイミングで，現在のサイクルの値をラッチしてしまう．所謂ショートパス問題[6]である．

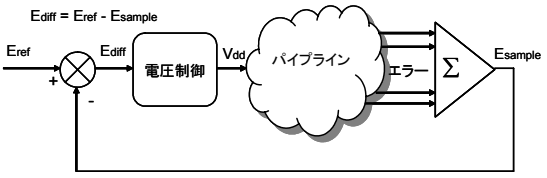

図 3：Razor における電圧制御システム

4. カナリア方式

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．本節では Razor を改善するカナリア方式を述べる．

4.1 カナリア FF

我々は改善方式のひとつとして，図 4 に示す回路を提案している[15]．この回路では，遅延クロックの代わりに遅延素子とカナリア FF 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．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．

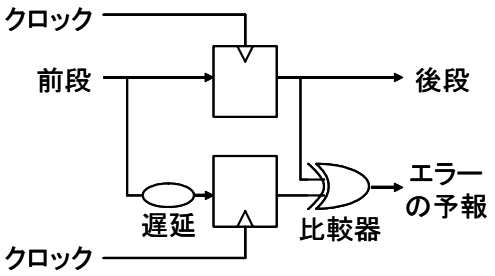

図 4:カナリア FF

注意しなければならない点は，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%は保証出来ないことである．この場合には二つの FF が同時にエラーとなるので，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．

4.2 カナリア FF を利用した省電力方式

図 5 は，動的電圧周波数可変技術(Dynamic Voltage Frequency Scaling : DVFS)がカナリア FF と協調動作する様子を説明している．横軸は時刻を表している．図の上半分に電源電圧の変化を示し，下半分に周波数の変化を示している．DVFS の動作を説明する．まず，二つの動作モードが存在する．それぞれを，過渡モード，定常モードと呼んでいる．

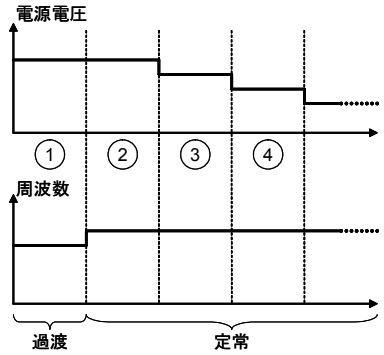

図 5:カナリア方式における DVFS

過渡モード（図中の①）では，遅延素子における信号遅延に配慮する．前段からカナリア FF に至る回路でタイミングエラーを生じないように，所望の周波数に比べて低速に動作させる（動的周波数可変：DFS）．クリティカルパスが活性化される最悪ケースは稀であるので周波数を容易に上昇できることに注意されたい．クリティカルパスは少数であり，残りの多くのパスはクリティカルパスの遅延よりも比較的小さな遅延で分布する[20]．例えば，約 80%のパスの遅延はクリティカルパスの遅延の半分である[19]．省電力設計時にはパス遅延の分布がクリティカルパス遅延付近に集中するように最適化を施すことが通常行われるが，それでもなおクリティカルパスは少数である[20]．これらのパスが活性化されてもタイミングエラーにはならないので，クリティカルパスにおけるタイミング制約が満足されなくても，タイミングエラーは稀にしか生じないことになる．このパス遅延の動作時ばらつきを利用すると，遅延素子の信号遅延を無視して所望の周波数まで向上でき，定常モードに移行する．

定常モード（図中の②）では，パス遅延の動作時ばらつきを更に積極的に利用して，図中の③，④，…のように電源電圧を降圧する（動的電圧可変：DVS）．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．

定常モードでは周波数が一定であることに注意されたい．クリティカルパス遅延で決定される電源電圧よりも低い状態になるため，大幅に消費電力を削減できる．カナリア方式は Razor と同等の電力削減[5,6]が期待できる．我々はすでに加算器を対してカナリア方式を採用し初期評価を行っている[15]．細粒度な電圧制御が可能で電圧変更に要するオーバヘッドを無視できるとする仮定の下で，30%を越える消費電力削減が確認できている．本稿の後続の節で，プロセッサ全体でカナリア方式を採用する場合の効果を評価する．

4.3 スキャン回路を利用したカナリア FF の実装

量産時テストのために用意されているスキャン回路を利用することで，極めて小さなハードウエアコストでカナリア FF を実装可能である．

図 6 はマイクロプロセッサで用いられているスキャン回路[14]である．システム FF とスキャン部か

ら構成されている. 図中の SI が別の FF の SO に接続され，プロセッサ中の全てのスキャン FF が接続されてシフトレジスタを構成する．テストモードでは，クロックとして SCA と SCB がそれぞれ LA と LB にテストパタンを入力する目的で供給される．続いてクロック UPDATE が与えられると, LB から PH1 に内容が移動し，テストパタンがシステム FF に書き込まれる．次にクロック CLK が与えられると，テストパタンを入力として得られた出力がラッチに保持される. 更に PH1 から LA に結果を移動する目的でクロック CAPTURE が供給される．最後に再び SCA と SCB が，今度はチップ外に結果を出力する目的で供給される．

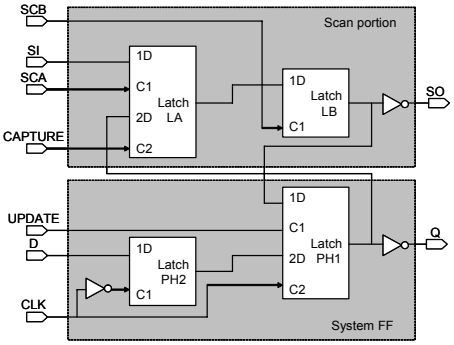

図 6:マイクロプロセッサのスキャン回路[14]

スキャン部は全ての FF に必要だが，テスト時のみに利用されるので，通常動作時には LA と LB は使用されていない．これらを通常動作時にも使用することで，極めて小さなハードウエアコストの追加で，カナリア FF を実装できる．スキャン回路を利用したカナリア FF を図 7 に示す.テスト時の動作は上述と同様である．通常動作時には，LA と LB がそれぞれ PH2 と PH1 のコピーを保持する．タイミングエラーが生じていなければ ERROR が 0 であるので，LA には D を遅延した値が保持される．一旦 ERROR が 1 になると LA には D の反転が保持され，エラーの検出が維持できる. これを受けて DVFS による昇圧を開始する．スキャン FF を利用出来るのはクロックが単一であるためであり，Razor FF に対するもうひとつの優位点である.

5. 評価環境

SimpleScalar ツールセット[1,4]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 1 億命令を対象とする．

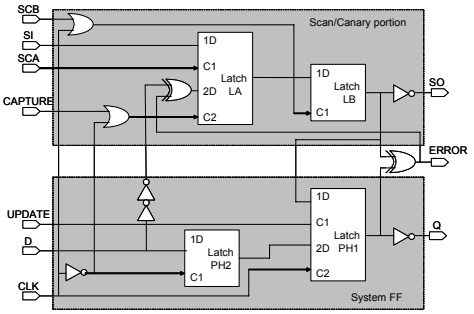

図 7:スキャン回路を利用したカナリア FF

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

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.

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おけるタイミングエラーで決まる[11]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，[15]で評価した桁上げ選択加算器（carry select adder：CSLA）をゲートレベルシミュレーションして得られた値[15]を用いる．ペンティアム M の電源電圧[8]を元に，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を求めている. 10 ステップの電圧を用意する．電源電圧とタイミングエラーの関係を図 8

に示す．ここで，遅延素子における遅延を CSLA におけるそれの 5%と仮定している．

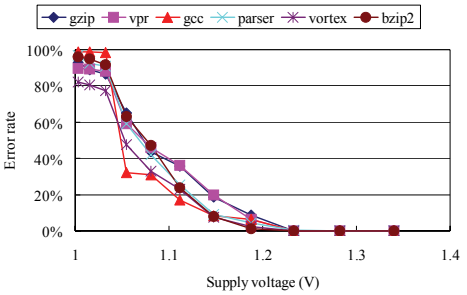

図 8:電源電圧とタイミングエラー率の関係

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．実際の入力に応じて CSLA の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．CSLA におけ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[10]ので初期評価としては問題無いと考えている．

6. 結果

選択された電源電圧，実行時間への影響，エネルギー削減効果，そしてエネルギー利用効率の改善について，順にシミュレーション結果を紹介する．

6.1 電源電圧

図 9 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．10 ステップの電圧から実際には 4 ステップしか選択されなかった．

まずリセット法とステップ法とを比較する．同じインターバルで両者を比較すると，ステップ法がより頻繁に低い電圧を選択している．リセット法はタイミングエラーを予報すると過渡モードに戻るので，これは容易に予想できる結果である．低い電圧をより頻繁に選択しているステップ法の方が，より大きな電力削減効果が得られると期待される．

続いてインターバルの影響を考察する．インターバルが短いほどより頻繁に低い電圧を選択している．インターバルが長いと次のステップに降圧されるまでの期間が長くなるので，これも容易に予想できる結果である．低い電圧を選択するほど，より大きな電力削減効果が得られると期待されるが，インターバルが短いと頻繁にステップ間を移動して電圧変更に要するオーバヘッドの影響が大きくなり，実行時間への悪影響が懸念される．

また，選択される電圧に振動が観察された．低い電圧が選択されるとタイミングエラーを発生しやすく直ちに高い電圧に変更されるが，その高い電圧ではエラーを生じないので再び低い電圧へ移行する，という現象である．この振動の抑制方法については[21]で報告する予定である．

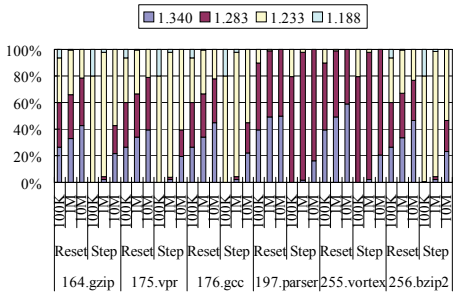

図 9:電源電圧の分布

6.2 実行時間

図 10 に実行時間に与える影響をまとめる．カナリア方式を採用しないプロセッサモデルでの実行時間に対する増加率で表している．

まずプログラム間の違いを観察する．容易にわかるように，プログラムの違いによる実行時間への影響の違いは見られない．図 8 よりプログラムの違いによるタイミングエラー発生率への影響は小さいことがわかるので，納得出来る結果である．

次に昇圧ポリシーの影響を考察する．プログラムやインターバルの違いに関係なく，ステップ法でリセット法よりも実行時間の増加割合が大きい．図 9 からステップ法の方が低い電圧を選択している割合が高いことがわかる．これはタイミングエラーに遭遇する確率も高いことを意味するので，ステップ法の方が頻繁に電圧を変更することになる．そのためのオーバヘッドが実行時間に大きな影響を与え，二方式で実行時間に有意差が現れている．前節で観察した結果からステップ法の電力削減効果が期待されたが，実行時間の増大を考慮すると，期待されるほどのエネルギー削減はないことが危惧される．

最後にインターバルの影響を考察する．昇圧ポリシーやプログラムの違いに関係なく，インターバルが小さくなるほど実行時間への影響は大きい．インターバルが小さくなればより頻繁に電源電圧を変

更し，そのオーバヘッドを大きく被るので，容易に予想される結果である．前節での危惧が確認できたことになる．したがってどの程度のインターバルが適切であるかは，エネルギー消費量を評価しないことには判断できない．

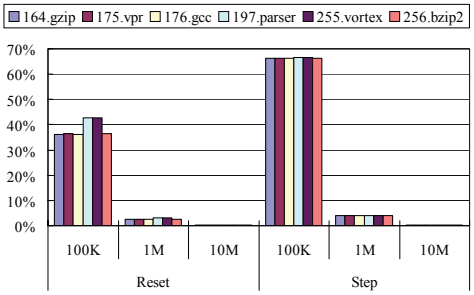

図 10:実行時間の増加割合

6.3 エネルギー削減率

図 11 に消費エネルギー削減率をまとめる．動作させるスキャン部と遅延回路で消費される電力は評価出来ていない．Razor でこれらに相当するシャドウ FF 等による影響は 3%の電力増である[5]．カナリア方式を採用しないプロセッサモデルの消費エネルギーを基準にしている．負の値は消費エネルギーが増大したことを表している．

まずプログラム間の比較であるが，絶対値は異なるものの，傾向には大差は見られない．

続いてインターバルの影響を考察する．前節での予想のとおり，インターバルが 100K サイクルの時には電圧変更のオーバヘッドの影響が深刻で，消費エネルギーが増大している．どの場合でも 20%以上増大しており，50%を越える場合も観察される．1M サイクルと 10M サイクルの時を比較すると，昇圧ポリシーの違いにより，優劣が異なる．

以上を受けて昇圧ポリシーの影響を考察する．リセット法の場合は 10M サイクルが好ましく，ステップ法の場合には逆に 1M サイクルが好ましい．前節で考察したように，リセット法の方がより稀に電圧を変更するためそのオーバヘッドの影響が比較的小さい．したがって，電圧変更の機会の多い 1M サイクルの方が，より低い電圧を選択出来る意味で好ましい結果となっている．ステップ法ではちょうど逆の理由から，電圧変更の機会の少ない 10M サイクルの方が良い結果となっている．

リセット法とステップ法とを比較すると，大きなインターバルの場合には後者が良好な結果となっている．図 10 よりわかるように，両者間での実行時間の違いは僅かであり，その影響よりも選択された電圧の影響の方が大きいと考えられる．その結果，低い電圧をより頻繁に選択しているステップ法が，大きなエネルギー削減を達成出来ている．

加算器での評価[15]と比較して，プロセッサ全体の評価では，エネルギー削減効果が小さい．前者では細粒度な電圧変更が可能であるという仮定を採用していたのに対して，後者ではより現実的な粗粒度な電圧変更のみが可能であると仮定したためである．細かく電圧を選択出来ないため，必ずしも最適な電圧で動作しているわけではないからである．

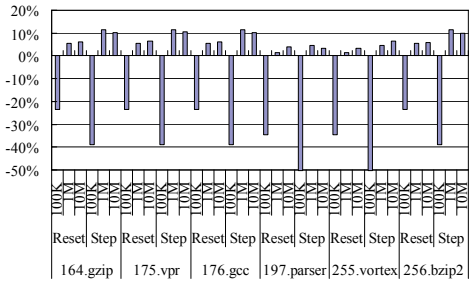

図 11:エネルギー削減率

6.4 エネルギー利用効率

図 12 にエネルギー，エネルギー遅延積(energy-delay product：EDP)，そしてエネルギー遅延自乗積(energy-delay-square product：ED²P)をまとめる．カナリア方式を採用しないモデルを基準にしており，値が小さい方が好ましい．前節で考察したエネルギーとは異なり，EDP と ED²P のどちらでも，インターバルは最も大きな 10M サイクルが良好な結果である．電圧変更ポリシーの影響は，どの指標でもステップ法が良好な結果である．

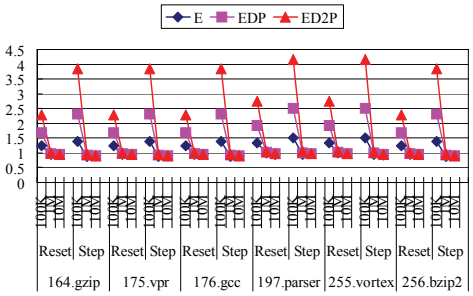

図 12:エネルギー利用効率

7. まとめ

ディープサブミクロン領域でのLSI設計を容易化するために，我々はカナリア方式を検討している．本稿ではマイクロプロセッサ全体での効果をシミュレーションにより評価した．電圧変更のポリシーとインターバルを適切に選択すれば，平均で約 9%のエネルギー削減を達成出来ることが確認された．

カナリア FF に関する第一の課題は，メイン FF でのタイミングエラーを回避あるいは検出する方法を検討することである．Razor との定量的な比較も興味深い．カナリア FF では電圧に応じて遅延量が変化するので，その決定と変化の影響などを検討・調査する必要がある．

将来の発展としては，まず，省電力効果の改善である．今回は入力依存の遅延ばらつきのみに着目した．1 節で述べたように，ばらつきには他にも電源のゆらぎや温度のばらつき等があり，それぞれに設計マージンが確保される．これらのマージンを取り除くことが出来れば，更なる省電力効果が期待できる．また，オーバークロック応用の検討も興味深い．

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．

参 考 文 献

[1] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: an Infrastructure for Computer System Modeling", IEEE Computer, Vol.35, No.2, 2002.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", 40th Design Automation Conference, 2003.

[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] D. Burger and T. M. Austin, "The SimpleScalar Tool Set, Version 2.0", ACM SIGARCH Computer Architecture News, Vol. 25, No. 3, 1997.

[5] S. Das, P. Sanjay, D. Roberts, L. Seokwoo 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.

[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] T. Karnik, S. Borkar, and V. De, "Sub-90nm Technologies: Challenges and Opportunities for CAD", International Conference on Computer Aided Design, 2002.

[10] 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.

[11] 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.

[12] T. Liu and S-L. Lu, "Performance Improvement with Circuit-level Speculation", 33rd International Symposium on Microarchitecture, 2000.

[13] S-L. Lu, "Speeding up Processing with Approximation Circuits", IEEE Computer, Vol.37, No.3, 2004.

[14] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust System Design with Built-In Soft-Error Resilience", IEEE Computer, Vol.38, No.2, 2005.

[15] T. Sato and Y. Kunitake, "A Simple Flip-Flop Circuit for Typical-Case Designs for DFM", 8th International Symposium on Quality Electronic Design, 2007.

[16] N. R. Shanbhag, "Reliable and Efficient System-on-chip Design", IEEE Computer, Vol.37, No.3, 2004.

[17] X. Tang, V. K. De, and J. D. Meindl, "Intrinsic MOSFET Parameter Fluctuations Due to Random Dopant Placement", IEEE Transactions on VLSI Systems, Vol.5, No.4, 1997.

[18] A. K. Uht, "Going beyond Worst-case Specs with TEAtime", IEEE Computer, Vol.37, No.3, 2004.

[19] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanazawa, M. Ichida, and K. Nogami, "Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor", IEEE Journal of Solid-State Circuits, Vol. 33, No. 3, 1998.

[20] 黒田, "低電力 CMOS 設計の秘訣とテクニック", IEEE SSCS Kansai Chapter Technical Seminar, 2003.

[21] 佐藤, 国武, "カナリア・フリップフロップを利用する DVS 方式の改良", 第 165 回計算機アーキテクチャ研究会, 発表予定, 2007.

[22] 山原, 美馬, 千代延, 佐藤, "タイミング違反を許容する省電力加算器における違反検出回路の高速化", 情報処理学会論文誌コンピューティングシステム, Vol.47, No.SIG18(ACS16), 2006.