

入力依存の遅延ばらつきを利用するDVSシステムにおける性能およびエリアオーバーヘッドの改善検討

国武, 勇次
九州大学大学院システム情報科学府

佐藤, 寿倫
福岡大学工学部電子情報工学科 | 独立行政法人科学技術振興機構, CREST

安浦, 寛人
九州大学大学院システム情報科学研究院

<https://hdl.handle.net/2324/10201>

出版情報：情報処理学会研究報告, 2007-ARC-178. 2008 (5), pp.93-98, 2008-05. 情報処理学会
バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

入力依存の遅延ばらつきを利用する DVS システムにおける 性能およびエリアオーバーヘッドの改善検討

国武勇次[†] 佐藤寿倫^{††*} 安浦寛人^{†††*}

[†] 九州大学大学院システム情報科学府

^{††} 福岡大学工学部電子情報工学科

^{†††} 九州大学大学院システム情報科学研究所

* 独立行政法人科学技術振興機構, CREST

あらまし 半導微細化技術の進展に伴って素子特性のばらつきが増大し, これまで行われてきた最悪ケースを考慮した設計が困難となっている. なぜなら, 素子特性のばらつきの増大によって過剰な設計マージンが取られ, 性能や消費電力に大きな影響を与えているからである. この過剰に見積もられた設計マージンを取り除く手法として, 我々はカナリア方式を検討している. これまで検討してきた方式では, 性能と面積のオーバーヘッドにおいて改善の余地がある. 本論文ではこれらの問題に対する解決策について検討する.

Considering Performance and Area Overhead in DVS System Utilizing Input Variations

Kunitake YUJI[†], Sato TOSHINORI^{††*}, and Yasuura HIROTO^{†††*}

[†] Graduate School of Information Science and Electrical Engineering, Kyushu University

^{††} Department of Electronics Engineering and Computer Science, Fukuoka University

^{†††} Faculty of Information Science and Electrical Engineering, Kyushu University

* Japan Science and Technology Agency, CREST

Abstract The deep submicron semiconductor technologies increase parameter variations and thus the processor design becomes more difficult. The increase in parameter variations requires excessive design margin that has serious impact on performance and power consumption. In order to eliminate the excessive design margin, we are investigating canary logic. Current canary logic causes severe performance loss and area overhead. In this paper, we show solutions for these problems.

1. ま え が き

半導体微細化技術の進展に伴って素子特性のばらつきが増加し, 従来行われてきた最悪ケースを思考した設計が困難となりつつある [1]. 素子特性のばらつきは, 製造時に生じるプロセスばらつき, 集積度の増加による消費電力増加に伴う電源電圧のゆらぎ, 高クロック化に伴うチップ内の温度のばらつきによって起こる. 素子特性のばらつきの増加は, 最悪ケースを考慮するための設計マージンを増加させ, 性能や消費電力に大きな影響を与えている. そこで, 設計者には最悪ケースに煩わされることなく, 典型的ケースを指向する設計が求められている.

我々は典型的ケースを指向する設計手法としてカナリアフリップフロップ (FF) を利用することを提案している [7]. これまでの研究において, 我々はカナリア方式を動的電圧制御

(Dynamic Voltage Scaling :DVS) に利用することを検討してきた. その成果として, 過剰に見積もられた設計マージンを取り除くことができることを実験により確認した. 一方, いくつかの問題点が明らかとなった. 本論文では, そのうちの電圧変更に必要な性能低下とカナリア FF の付加による面積のオーバーヘッドの問題について改善策を検討する.

以下に本稿の構成について説明する. 2 節でカナリア方式とそれを利用する DVS システムを紹介する. 3 節でカナリア方式を利用する DVS システムの評価結果を示し, 性能低下の問題について説明する. 4 節で性能低下を抑制する手法の提案を行う. 5 節で構築したシミュレーション環境について述べ, 6 節で提案手法の評価を行う. 7 節で面積のオーバーヘッドの問題について対策を検討する. 8 節で関連研究を紹介する. 最後に 9 節で本論文をまとめる.

2. カナリア方式

典型的ケース指向設計のひとつとして、我々は図 1 に示すカナリア方式を検討している [7]。カナリア方式では、通常のフリップフロップ (メイン FF) に対して、カナリア FF と呼ぶフリップフロップを用意する。カナリア FF の直前には遅延素子が挿入されるため、カナリア FF の方がメイン FF よりもタイミング制約が厳しくなる。従って、カナリア FF の方がメイン FF よりも先にタイミングエラーに遭遇する。メイン FF とカナリア FF に取り込まれた値を比較し、一致しなかった場合にメイン FF で発生しようとしているタイミングエラーが予報される。

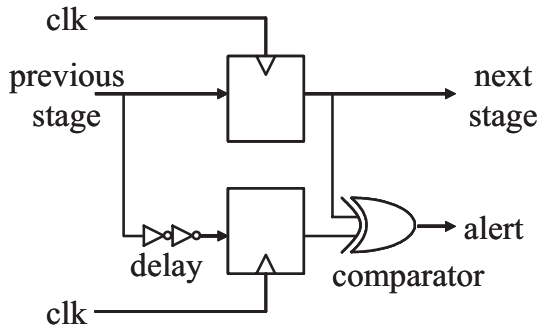


図 1 カナリア方式

このカナリア方式の特徴を利用して DVS を行うことを検討している。図 2 にカナリアを利用する DVS システムの電圧制御を示す。縦軸が電源電圧、横軸が時間をそれぞれ示している。図の点線で示されるのは予め決められた一定のインターバルである。決められたインターバル中にタイミングエラーが予報されなかった場合、メイン FF のマージンに余裕があると判断し降圧する。一方、インターバル中にタイミングエラーが予報されると、メイン FF のマージンに余裕がないと判断し直ちに昇圧する。このように電圧を制御することで、過剰に見積もられた設計マージンを取り除くことができる。

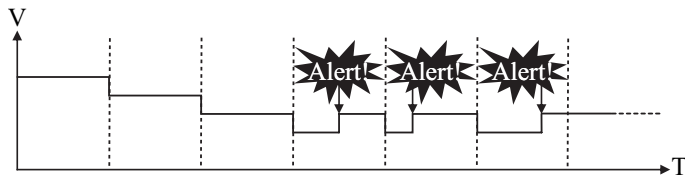


図 2 カナリア方式を利用する DVS システム

3. エネルギー利用効率への影響

カナリア方式を整数加算器に適用した評価結果から明らかとなった問題について言及する。カナリア FF の直前に挿入される遅延素子の遅延量は、整数加算器のクリティカルパスの遅延の 10% として評価を行った。シミュレーション環境については、5 節で詳しく述べる。図 3 にエネルギー利用効率を示す。エネルギー利用効率の評価指標にはエネルギー遅延積 (Energy Delay

Product: EDP) を用いる。各グラフはカナリア方式を適用しないモデルで正規化されている。margin50% と margin100% はそれぞれクリティカルパス遅延に対して 50% と 100% 設計マージンを持たせた場合を示している。つまり、margin50% と margin100% はそれぞれクロック周期をクリティカルパス遅延の 1.5 倍と 2 倍とした場合である。図からわかるように、マージンが 50% の場合 EDP はおよそ 10% 悪化している。EDP は消費エネルギーと遅延の積で求められる。そこで消費エネルギーと性能について調査を行い EDP の悪化の原因を考察する。

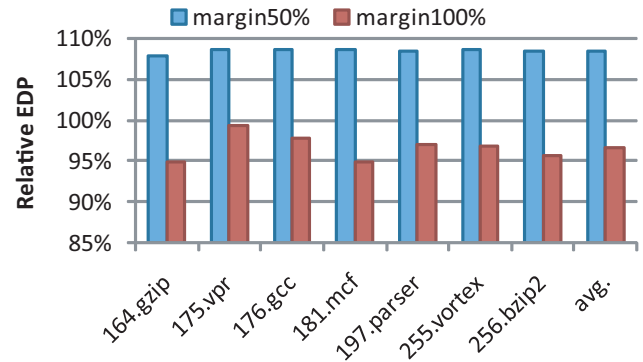


図 3 エネルギー利用効率

図 4 に消費エネルギーを示す。各グラフはカナリア方式を適用しないモデルで正規化されている。margin50% のとき平均でおよそ 25%、margin100% のとき平均でおよそ 30% 削減されていることが確認できる。これは、DVS 制御によって過剰な設計マージンが取り除かれた結果である。この結果から、消費エネルギーが EDP を悪化させる原因ではないと言える。

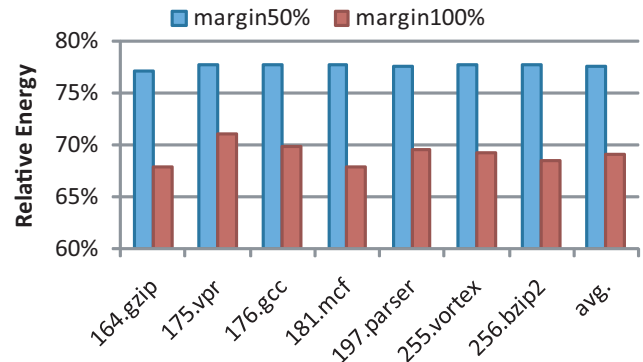


図 4 消費エネルギー

図 5 に実行時間を示す。グラフはカナリア方式を適用しないモデルの実行サイクル数で正規化されている。margin50% と margin100% は両方ともおよそ実行サイクル数が 40% 増加していることが確認できる。この結果から、EDP が悪化している原因は性能の低下であると言える。カナリア方式を利用する DVS システムで性能低下を引き起こす原因は、電圧変更するときそれが安定するまでに要する時間のみである。この時間のオーバーヘッドは $10\mu\text{s}$ [3] であると想定している。提案する

DVS システムで徐々に電源電圧を低下させると、しばしばタイミングエラーが予報されるようになる。タイミングエラーが予報されるとメイン FF のタイミングエラーを避けるために昇圧が行われる。これによってメイン FF のマージンに余裕が出来るため、次のインターバルで降圧される。これを繰り返すことによって電源電圧に振動が発生し性能低下を引き起こしてしまう。

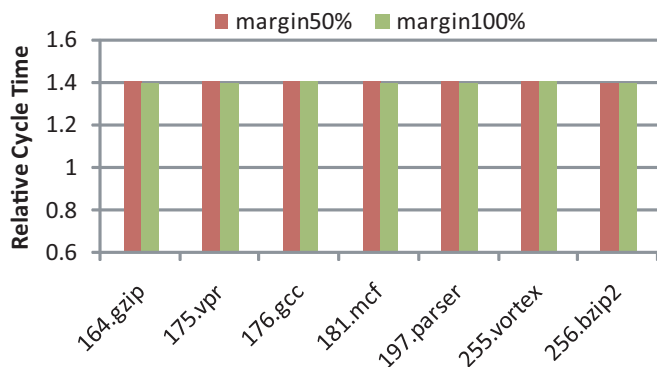


図 5 実行サイクル数

4. 二段階カナリア

前節で説明した電源電圧の無駄な振動を抑制するために、カナリア方式を改良した二段階カナリアを提案する。この二段階カナリアを用いて電圧制御方法を改良することで無駄な振動を取り除く。

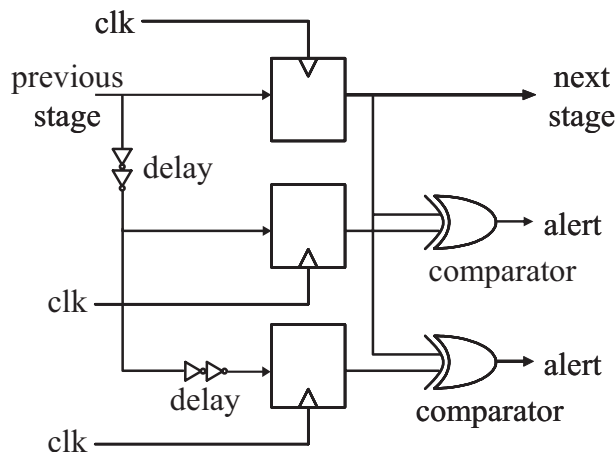


図 6 二段階カナリア

図 6 に二段階カナリアを示す。図に示すようにカナリア FF を二つ用意する。この二つのカナリア FF は遅延素子の遅延量が異なる。以後、遅延量の大きいカナリア FF をカナリア L とし、遅延量の小さいカナリア FF をカナリア S とする。従ってカナリア L の方がカナリア S より先にタイミングエラーに遭遇する。この二つのカナリア FF はそれぞれ異なる目的で動作する。カナリア L は振動を防ぐ目的で利用され、カナリア S はこれまでと同様メイン FF で発生しようとするタイミングエラー

を予報する目的で利用される。表 1 に電源電圧の制御方法をまとめる。カナリア L でエラーが予報され、カナリア S でエラーが予報されなかった場合、これ以上電源電圧を下げると振動が発生する可能性がある。従って次のインターバルで現在の電圧を維持する。カナリア L とカナリア S の両方でエラーが予報された場合は、メイン FF の設計マージンに余裕が無くなっているので直ちに昇圧するように制御を行う。

表 1 電源電圧の制御方法

カナリア S	カナリア L	電源電圧
Not predict	Not predict	降圧
Not predict	Predict	維持
Predict	Predict	昇圧
Predict	Not Predict	-

図 7 に二段階カナリアを用いた場合の電源電圧の遷移の例を示す。縦軸が電源電圧、横軸が時間を表している。また、 I_x はインターバルを表している。設計マージンに余裕がある間はインターバルの経過と共に電源電圧を下げていく。時刻 t_1 においてカナリア L のみタイミングエラーが予報されると、次の I_3 で振動を避けるために電圧を維持する。時刻 t_2 において両方のカナリア FF でタイミングエラーが予報される。この場合、メイン FF のタイミングエラーを避けるため電圧を昇圧する。時刻 t_3 でカナリア L のみタイミングエラーが予報され、次の I_4 では電圧を維持している。このように制御を行うことで、無駄な電圧の変更が抑制できると期待できる。

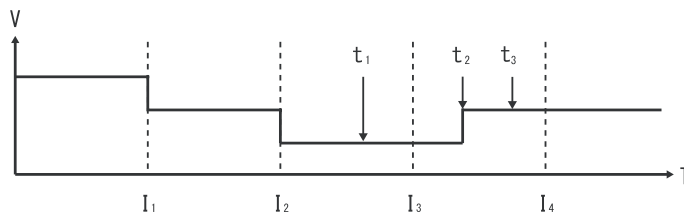


図 7 二段階カナリアを利用する DVS システム

5. コ・シミュレーションによる評価

回路遅延は回路中のバスを活性化する入力値に依存する。従って、カナリアを利用する DVS システムの振舞をシミュレーションするには回路遅延を考慮出来るゲートレベル・シミュレーションを行う必要がある。しかしながら、ゲートレベル・シミュレーションは評価する回路規模が大きいと非常に時間を要するため、アーキテクチャの評価には向いていない。そこで、回路遅延を考慮できるゲートレベル・シミュレーションとアーキテクチャの評価に適したアーキテクチャレベル・シミュレーションを協調させたコ・シミュレーション環境を構築する。

図 8 に構築したコ・シミュレーション環境の構成を示す。プロセッサ全体をアーキテクチャレベル・シミュレータでシミュレーションし、カナリア方式の適用回路のみゲートレベル・シミュレータでシミュレーションする。これによりゲートレベルの回路規模を抑える。左側がアーキテクチャレベル・シミュ

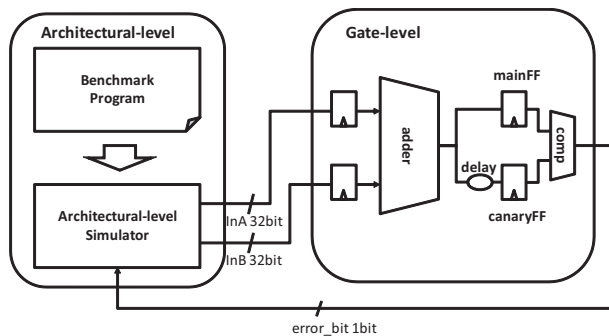


図 8 コ・シミュレーションの構成

レーションを、右側がゲートレベル・シミュレーションを表している。ゲートレベルとアーキテクチャレベルの通信にはプロセス間通信と SystemVerilog の機能である Direct Programming Interface(DPI) を用いる。DPI とは HDL と C 言語の通信を行うためのインターフェースである [8]。

コ・シミュレーションの流れを示す。本評価では整数加算器のみにカナリア方式を適用する。まずアーキテクチャレベル・シミュレータによってベンチマークプログラムを実行する。その実行中に整数加算器を利用する命令、つまり ADD 命令もしくは SUB 命令が発行されると、そのオペランドと現在選択されている電圧の情報をゲートレベル・シミュレータに送り制御をゲートレベル・シミュレータに切替える。ゲートレベル・シミュレータでは受け取った電圧の情報をもとにタイミング制約を変更し整数加算器で計算を行う。次に各 FF に取り込まれた値を比較し、その結果をアーキテクチャレベル・シミュレータに送信して制御を再びアーキテクチャレベル・シミュレータに戻す。アーキテクチャレベル・シミュレータでは、受け取った情報に応じて実行を続ける。

整数加算器には 32 ビット桁上げ選択加算器 (Carry Select Adder: CSLA) を選択した。論理合成には Synopsys 社の DesignCompiler を使用し、シミュレーションには Cadence 社の NC-Verilog を使用した。またライブラリには日立 0.18 μm スタandardセルライブラリを使用した。アーキテクチャレベル・シミュレータには MASE シミュレータ [5] を利用した。命令セットには PISA を使用する。プロセッサ構成は表 2 の通りである。ベンチマークプログラムには SPEC2000 の整数系プログラムから 7 つを選択した。実行命令は最初の 10 億命令をスキップし、続く 10 億命令を対象としている。

電源電圧を変更するインターバルは 100K サイクルで評価する。電源電圧の変更には $10\mu\text{s}$ を要する [3] と仮定した。この電源電圧を変更している間はプロセッサは何も実行していないアイドル状態である。ゲートレベル・シミュレーションで CSLA に与える設計マージンはクリティカル遅延の 50%と 100%とする。また、各カナリア FF の直前に挿入される遅延素子は、オリジナルのカナリア FF をクリティカルパス遅延の 10%としたため、カナリア S を 5%、カナリア L を 15%とする。DVS に使用する電源電圧は表 3 に示す PentiumM の電源電圧 [4] を利用する。

表 2 プロセッサの構成

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

表 3 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

6. 二段階カナリアの評価

マージンが 50%と 100%としたときのオリジナルのカナリア方式と二段階カナリアを利用した場合で評価を行い、その結果を比較する。以降のグラフに示す 1st_m50, 1st_m100, 2st_m50 そして 2st_m100 は、それぞれオリジナルのカナリア方式を利用した場合のマージン 50%、マージン 100%、二段階カナリアを利用した場合のマージン 50%、マージン 100%の結果を示している。

図 9 にオリジナルのカナリア方式と二段階カナリアを利用した場合の実行時間を示す。各グラフはカナリア方式を適用しないモデルで正規化されている。グラフからわかるように、ほとんどのベンチマークプログラムにおいて実行時間の増加が 10%未満に抑えられている。3 節で述べたように、カナリア方式を利用する DVS システムで、性能に影響を与えるのは電圧変更時にそれが安定するまでの時間のみである。このことから、二段階カナリアを利用することで電源電圧の振動を取り除くことが出来たと確認できる。

消費エネルギーを図 10 に示す。各グラフはカナリア方式を適用しないモデルで正規化されている。マージンが 50%の場合は同程度、マージンが 100%の場合はより大きな消費エネルギー削減率を示している。マージンが 100%の場合により多く消費エネルギーが削減された理由としては、二段階カナリアを利用することで、ある二つの電圧間で発生していた振動が、低い電圧で安定したためだと考えられる。マージンが 50%の場合でこのようにならなかった理由は、降圧されたら直ちにエラーが予報され昇圧されるような振動が発生していたためだと考え

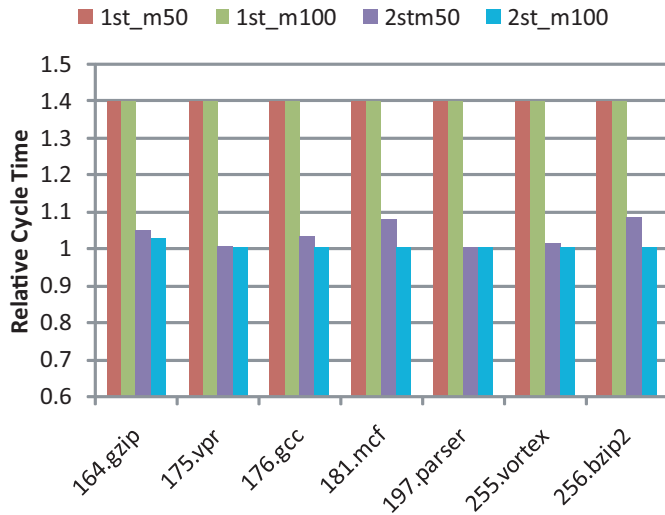


図 9 実行サイクル数の比較

られる。このような振動を取り除いた場合、振動が発生している二つの電圧のうち高い電圧で安定しよう。ただし、振動が発生していた際に低い方の電圧は瞬間的にしか選択されなかったため、消費エネルギーの削減率に影響が無かったと考えられる。

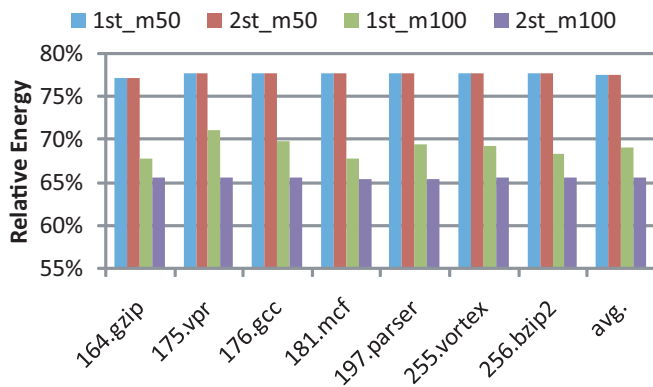


図 10 消費エネルギーの比較

EDP を図 11 に示す。各グラフはカナリア方式を適用しないモデルで正規化されている。二段階カナリアを用いることで性能低下を抑えた結果、カナリア方式を適用しないモデルと比べてマージンが 50% のときでおよそ 20%，マージンが 100% のときでおよそ 35% EDP が改善された。

7. エリアオーバーヘッド抑制手法

これまで、カナリア方式や二段階カナリアを利用することで過剰に見積もられた設計マージンを取り除くことができることを明らかにした。しかし、カナリア方式では、通常の FF に加えてカナリア FF、XOR ゲートと遅延素子が追加されるため、全ての FF をカナリア方式に置き換えると面積が増加してしまう。ましてや、カナリア FF を 2 つ要する二段階カナリアではその影響は大きいと考えられる。

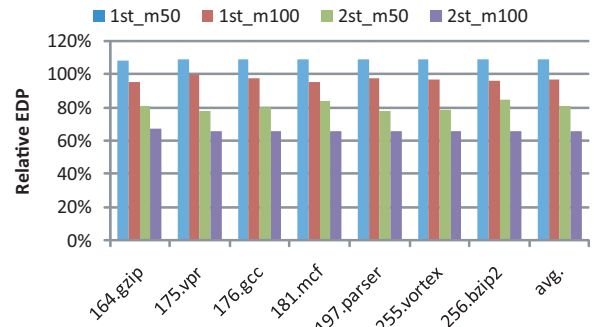


図 11 エネルギー遅延積の比較

しかし、全てのフリップフロップをカナリア FF に置き換える必要はない。回路のタイミング制約を決定するような、遅延の長いパスの出力側にあるフリップフロップのみ置き換えればよい。なぜならば、徐々に電源電圧を下げていったときに遅延の長いパスと短いパスでは、遅延の長いパスの方が先にタイミングエラーに遭遇するためである。ただし、遅延の長いパスが活性化されないような状況では、遅延の短いパスでタイミングエラーが発生してしまう可能性がある。従って、カナリア FF の挿入位置を限定するためには慎重なパス解析が必要となる。以下に、検討中のカナリア FF の挿入位置の限定方法を示す。

統計的遅延解析 (Statistical Static Timing Analysis : SSTA) のパス解析の高速化手法が応用できる [9]。SSTA とは、回路遅延のばらつきを統計的に考慮することにより、回路遅延をより正確に見積もる技術である。パス解析とは、信号が伝播するパスの遅延を一本ずつ独立に解析する方法で、パスの遅延を高い精度で解析できる。しかし、パスの本数は回路規模によって指数関数的に増加するため解析に時間がかかる。応用するのはこの問題を解決するための高速化手法である。

本間ら [9] により、チップの総パス数に関係無く、クリティカルパス遅延分布の -4σ 値以上の遅延をもつパスに対して SSTA を行えば全てのパスを解析した場合と同じ精度の統計的遅延値が得られることが明らかとなっている。そこで、パス解析の高速化手法を利用した挿入位置の限定方法のアルゴリズムを以下にまとめる。

1. 適応回路に対して STA を行い、パスの遅延の分布とクリティカルパスを求める。
2. 1 で求めたクリティカルパスに対して SSTA を行う。
3. 2 で求めたクリティカルパスの遅延分布とパスの遅延の分布を比較し、クリティカルパスの -4σ 以上の遅延をもつパスを求める。
4. 3 で求めたパスに対して SSTA を行い統計的遅延値を求める。
5. 4 で求めた統計的遅延値が最小となるものよりも長い遅延をもつパスに接続されるフリップフロップをカナリア FF と置き換える。

ただし、この方法ではカナリア FF を挿入したパスが活性化せず、カナリア FF が挿入されていないパスが活性化した場合、タイミングエラーを予報出来ない可能性がある。

そこで、カナリア FF を挿入されなかったフリップフロップの動作を保証する挿入位置の限定方法を検討する。

1. DVS で用意する電源電圧値の最も低い値を選択したときの回路のパスの遅延分布を STA によって求める。
2. パスの遅延とパスとフリップフロップの接続情報を持ったパス情報のリストを用意する。このリストを以降、未解析パスのリストと呼ぶ。
3. 未解析パスのリストから最も長い遅延をもつパスを抜き出し、SSTA を行いパスの遅延分布を求める。
4. 求めたパスの遅延分布がタイミング制約を少しでも満たしていなかった場合、このパスに接続されるフリップフロップにはカナリア FF が挿入される。このとき、解析したパスと同じフリップフロップに接続されるパスは新たに解析する必要が無い。従って、未解析パスのリストから取り除く。
5. 1~4 を 3 で求めたパスの遅延分布が完全に目標性能を満たす遅延内に収まるまで行う。

解析したパスの遅延分布がタイミング制約を満たす範囲内に収まったとき、そのパスよりも遅延の小さい未解析パスに接続されるフリップフロップはカナリア方式に置き換える必要が無い。なぜならば、それらのパス群は DVS で用意される最も低い電源電圧が選択された場合でもタイミングエラーにはならないためである。

8. 関連研究

設計マージンを取り除く有効な手法として Razor [2] がある。Razor は、性能が保証される動作周波数よりも高い周波数で動作する。このとき発生するタイミングエラーに対して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回復させる。このとき発生するタイミングエラーを検出する機構として RazorFF が提案されている。RazorFF はメイン FF に対して遅延クロックを供給するシャドウ FF を付加している。従って、シャドウ FF は常に正しい値が期待される。この二つのフリップフロップに取り込まれる値を比較して一致しなかった場合にタイミングエラーが検出される。

Samanta [6] らは RazorFF を改良した Modified RazorFF を提案している。Modified RazorFF はタイミングエラーが発生した場合、同一クロックサイクル内にシャドウ FF の保持する正しい値に信号を安定させることが出来る。従ってタイミングエラーが発生した場合、次のステージのタイミング制約が厳しくなってしまう。このため回路の動作を加速させる手法を提案している。また、これらのエリアオーバーヘッドを抑制するために、エリアオーバーヘッドの問題を混合整数線形計画問題 (MILP) に当てはめ最適なエリアオーバーヘッドのコストを求めている。

9. まとめ

我々は、過剰に見積もられた設計マージンを取り除くための手法としてカナリア方式を利用する DVS システムを検討して

いる。本稿では、本システムの無駄な電源電圧の振動による性能低下の問題について、カナリア方式を改良した二段階カナリア方式を利用することを提案し、その効果を明らかにした。また、カナリア方式を利用することによるエリアオーバーヘッドの問題に対して、カナリア方式の挿入位置を限定する方法を検討した。今後、カナリア方式の挿入位置を限定する方法の効果と、挿入位置の決定方法の妥当性を調査する。

謝辞

本研究の一部は、科学研究費補助金・基盤 A(No.19200004)、および科学技術振興機構 CREST プロジェクトの支援によるものである。なお、東京大学 VDEC を通じて提供された日立製作所の $0.18 \mu\text{m}$ ライブラリを使用している。

文 献

- [1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture", Annual ACM IEEE Design Automation Conference, pp.338-342, 2003.
- [2] 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, pp.7-18, December 2003.
- [3] 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.
- [4] Intel Corporation, "Intel Pentium M Processor on 90nm Process with 2-MB L2 Cache", Datasheet, 2006.
- [5] E. Larson, S. Chatterjee, and T. Austin, "MASE: A Novel Infrastructure for Detailed Microarchitectural Modeling", International Symposium on Performance Analysis of Systems and Software, pp.1-9, 2001.
- [6] R. Samanta, G. Venkataraman, N. Shah, and J. Hu, "Elastic Timing Scheme for Energy-Efficient and Robust Performance", 9th International Symposium on Quality Electronic Design, pp.537-542, 2008.
- [7] T. Sato, and Y. Kunitake, "Critical Issues Regarding A Variation Resilient Flip-Flop", 14th Workshop on Synthesis and System Integration of Mixed Information Technologies, pp.280-286, 2007.
- [8] S. Surtherland, S. Davidmann, and P. Flake, "System Verilog for Design", Kluwer Academic Publishers, 2003.
- [9] 本間克己, 澁谷利行, 新田泉, 松岡英俊, "パスベース統計的遅延解析における解析パス数と精度に関する考察", 情報処理学会研究報告 SLDM, Vol.2006, No.28, pp.61-66, 2006.