

基板バイアス制御による遅延ばらつき補償および配線遅延を考慮した低エネルギーオーバーヘッド指向の高位合成手法

井川 昂輝¹ 史 又華² 柳澤 政生¹ 戸川 望¹

概要: 製造プロセスの微細化により、配線遅延の相対的な拡大や、製造ばらつきの増大が問題となっている。製造ばらつきに対する有力な解決策として、基板バイアス制御がある。順方向基板バイアス電圧の印加により、低速な LSI は高速化するため、歩留りが向上する。しかしリークエネルギーの増大が問題となるため、増大量を最小化する回路構成を設計段階で考慮する必要がある。本稿では高位合成段階で適切に配線遅延への対応が可能な分散レジスタアーキテクチャを対象に、基板バイアス制御時のリークエネルギーを最小化する高位合成手法を提案する。提案手法ではワーストケース遅延の下でもタイミング違反が起こらないように、回路ブロックごとに最小のワースト印加基板バイアス電圧を見積もる。さらにレイテンシが増大しない範囲内で、ワースト印加基板バイアス電圧を最小化することで、リークエネルギーを最小化する。計算機実験により、遅延ばらつき補償時のリークエネルギーの平均値を、従来手法と比較し最大 53.9%削減できることを確認した。

キーワード: 高位合成, 製造ばらつき, 基板バイアス制御, 配線遅延, 分散レジスタアーキテクチャ

A Low-Energy-Overhead and Interconnection-Delay Aware High-level Synthesis Algorithm for Delay Variation Compensation with Body Biasing

KOKI IGAWA¹ YOUHUA SHI² MASAO YANAGISAWA¹ NOZOMU TOGAWA¹

Abstract: In this paper, we propose a high-level synthesis algorithm to minimize the expected overhead of leak energy by body biasing for distributed-register architectures. To use distributed-register architectures, we can estimate interconnection delays efficiently in high-level synthesis. We estimate worst body-bias voltages for each circuit block so that timing errors cannot occur. Moreover, we minimize the expected leak energy under the condition that latency does not increase. Experimental results show that our algorithm reduces the expected leak energy when we use body biasing by up to 53.9% compared with several existing methods.

Keywords: high-level synthesis, process variation, body biasing, interconnection delay, distributed-register architectures

1. はじめに

製造プロセスの微細化が進み、製造ばらつきの増大によるタイミング歩留りの低下や、配線遅延のゲート遅延に対する相対的な拡大が問題となっている。このような問題に対し、従来、ワーストケース遅延や大きな配線遅延の下でも動作できるように、十分なマージンを設けた設計により対応してきた。しかし過大なマージン設計による性能低下が問題となる。一方、大規模化する LSI 設計の生産性を向

上させる設計技術として、高位合成の重要性が増大している。本稿では、製造ばらつきや配線遅延を考慮し、性能低下が小さい高位合成手法に注目する。

製造ばらつきに対する有力な解決策として、基板バイアス制御による回路性能の調整がある。基板バイアス制御は MOS トランジスタの基板バイアス効果を利用した技術である。NMOS トランジスタのしきい値電圧 V_{th} は、次式で表せる [1]。

$$V_{th} = V_{th0} + \gamma(\sqrt{2\phi_f - V_{bs}} - \sqrt{2\phi_f}) \quad (1)$$

ここで V_{bs} は基板-ソース間の電圧 (基板バイアス電圧)、 V_{th0} は V_{bs} が 0V のときのしきい値電圧、 γ は基板係数、 ϕ_f はフェルミポテンシャルである。基板バイアス制御では、

¹ 早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻
Department of Computer Science and Communications Engineering, Waseda University

² 早稲田大学高等研究所
Waseda Institute for Advanced Study, Waseda University

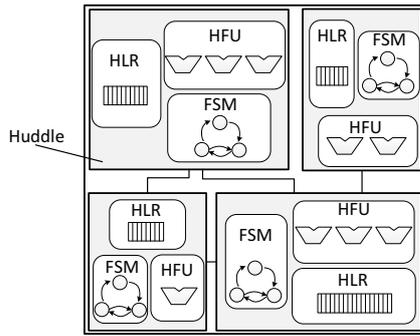


図 1: HDR アーキテクチャ [4].

低速な LSI に対し、順方向に基板バイアス電圧を印加する ($V_{bs} > 0$) ことにより、しきい値電圧を下げる。回路動作が高速化するため、タイミング歩留まりの向上が可能である。しかしリークエネルギーの増大が最大の問題となる。

基板バイアス制御を考慮した高位合成手法として、[2, 3] がある。[2] は、基板バイアス制御を考慮したモジュール選択アルゴリズムを提案している。しかし配線遅延を考慮していない。[3] は、基板バイアス電圧と電源電圧の同時調整を考慮した高位合成手法を提案している。遅延と消費電力のばらつきを改善した結果が示されている。

一方、高位合成段階で配線遅延の適切な扱いが可能なアーキテクチャモデルとして、分散レジスタアーキテクチャが提案されている。分散レジスタアーキテクチャでは、各演算器の近くにレジスタを分散させることで、演算器とレジスタ間の配線遅延を小さくできる。本稿では、分散レジスタアーキテクチャの 1 つである Huddle-based Distributed Register (HDR) アーキテクチャ [4] に注目する。

本稿では HDR アーキテクチャを対象に、基板バイアス制御時のリークエネルギーを最小化する高位合成手法を提案する。提案手法ではワーストケース遅延の下でもタイミング違反が起らないように、回路ブロックごとに最小のワースト印加基板バイアス電圧を見積もる。さらにレイテンシが増大しない範囲内で、ワースト印加基板バイアス電圧の最小化することで、リークエネルギーを最小化する。計算機実験により、基板バイアス電圧印加時のリークエネルギーの平均値を、従来手法と比較し最大 53.9%削減できることを確認した。

2. 問題定義

2.1 分散レジスタアーキテクチャ

本稿では、分散レジスタアーキテクチャの 1 つである HDR アーキテクチャ [4] を対象とする。HDR アーキテクチャの概要図を図 1 に示す。HDR アーキテクチャでは、チップ全体をハドルと呼ばれる配線遅延の影響を無視できる区画に分割する。ハドル内の演算器は、同一ハドル内のレジスタやコントローラを共有する。ハドルの構成要素を次に示す。

Huddled Functional Unit (HFU): 演算器およびマルチプレクサの集合。主に同一ハドルのローカルレジスタにアクセスする。

Huddled Local Register (HLR): ローカルレジスタおよびマルチプレクサの集合。

Finite State Machine (FSM): 同一ハドル内の HFU および HLR を制御するコントローラ。

ハドル内では配線遅延の影響が十分小さいとみなせるため、モジュール配置の抽象化が可能である。ハドル間の配線遅延を予測し、ハドル単位でフロアプランを実行することで、高位合成段階で適切に配線遅延を扱える。

2.2 基板バイアス制御とリークエネルギーの最小化

入力として、CDFG $G = (V, E)$ 、演算器集合 $F = \{f_1, \dots, f_p\}$ 、クロック周期制約 T_{clk}^{*1} 、基板バイアス電圧集合 $V = \{0V, 0.05V, \dots, 0.4V\}$ を与える。 p 個の演算器を q 個 ($q \leq p$) のハドル集合 $H = \{h_1, \dots, h_q\}$ に割り当てる。演算器 f_i を割り当てるハドルを $Hud(f_i)$ と表す。ハドル h_j に割り当てられた演算器の集合を $FH(h_j)$ と表す。以下、ハドル h_j を演算器 f_i が属するハドルとする ($h_j = Hud(f_i)$)。

演算器 f_i に基板バイアス電圧 v が印加された場合の遅延を $d_{f_i}(v)$ と表す。製造ばらつきによりトランジスタのパラメータはばらつきを持つため、 $d_{f_i}(0)$ もばらつきを持つ。典型的ケース、ワーストケースの場合の $d_{f_i}(v)$ を特に $d_{f_i}^{typ}(v)$ 、 $d_{f_i}^{wst}(v)$ と表す。同様に、レジスタについて $d_r(v)$ 、 $d_r^{typ}(v)$ 、 $d_r^{wst}(v)$ を定義できる。チップ製造後、ハドル h_j に対し、タイミング違反を補償できる最小の基板バイアス電圧を $HV(h_j)$ と表す。ワーストケースの場合を特に $HV^{wst}(h_j)$ と表し、ワースト印加基板バイアス電圧と呼ぶ。

HDR アーキテクチャをターゲットアーキテクチャとするため、基板バイアス電圧の制御粒度はハドル単位とする。したがって、タイミング違反が起きた LSI に対し、遅延補償ができるように最適な基板バイアス電圧をハドルごとに印加する。しかし、製造された LSI の遅延はばらつきを持つため、最適な基板バイアス電圧を設計時に一意に定めることはできない。[2] の遅延モデルより、 $d_{f_i}(0V) \leq d_{f_i}^{wst}(0V)$ を満たすならば、 $HV(h_j) \leq HV^{wst}(h_j)$ 、 $d_{f_i}(HV(h_j)) \approx d_{f_i}(HV^{wst}(h_j))$ が成立するものとする。したがって、ワーストケース遅延を想定し、基板バイアス制御によるリークエネルギーが最小となるように生成した高位合成結果から生成された回路に対して、製造後に基板バイアス制御による遅延補償を行う場合、ワーストケース遅延以下であるならばリークエネルギーが最小となることが期待される。本稿では、高位合成段階で補償すべき遅延としてワーストケース遅延を想定する。

2.3 データ転送サイクル行列

$HS(h_j)$ は、高位合成段階でハドル h_j のモジュールに想定する遅延特性を表す。 $HS(h_j) = typ$ ならばハドル h_j のモジュールは典型的ケース遅延が想定され、 $HS(h_j) = wst$ ならばワーストケース遅延が想定される。これを H 行に並べ、想定モジュール特性ベクトル HS を得る。高位合成時における、演算器 f_i およびハドル h_j のレジスタの遅延を $D_f(f_i)$ 、 $D_r(h_j)$ と表す。 $D_f(f_i)$ 、 $D_r(h_j)$ の値は、 $HS(h_j) = typ$ の場合、 $d_{f_i}^{typ}(0V)$ 、 $d_r^{typ}(0V)$ 、 $HS(h_j) = wst$ の場合、 $d_{f_i}^{wst}(HV^{wst}(h_j))$ 、 $d_r^{wst}(HV^{wst}(h_j))$ とする。演算器 f_i からハドル h_k のレジスタに演算結果を転送するま

*1 製造後に、ばらつきに応じて動作クロック周波数を変更する手法 [5] も広く用いられている。しかし周辺回路との兼ね合いでクロック周波数の変更が許されない場合も存在する。本稿では設計時に与えたクロック周期は製造後でも変更せずに製造ばらつきに対応する。

でのスラック時間は、以下の式で表される。

$$Slack(f_i, h_k) = S_f(f_i) \times T_{clk} - D_f(f_i) - D_r(h_j) - D_{tr}(h_j, h_k) \quad (2)$$

$D_{tr}(h_j, h_k) = D_w(Dist(h_j, h_k)) + \epsilon_{jk}$ である。 $Dist(h_j, h_k)$ はハドル h_j とハドル h_k のマンハッタン距離を表す。 $D_w(x)$ は距離 x における配線遅延を表す。 ϵ_{jk} はハドル h_j からハドル h_k へのデータ転送におけるマージンである。 $S_f(f_i)$ は演算器 f_i の実行サイクル数を表す。 $S_f(f_i)$ は次式で算出する。

$$S_f(f_i) = \left\lceil \frac{D_f(f_i) + D_r(h_j)}{T_{clk}} \right\rceil \quad (3)$$

以上より、データ転送サイクル行列 $DC = \{dc_{ik}\}$ を次式で定義する。

$$dc_{ik} = \begin{cases} 0 & \text{if } Slack(f_i, h_k) > 0 \\ \left\lceil \frac{D_r(h_j) + D_{tr}(h_j, h_k)}{T_{clk}} \right\rceil & \text{otherwise} \end{cases} \quad (4)$$

dc_{ik} は演算器 f_i からハドル h_k のローカルレジスタへのデータ転送に必要なステップ数を表す。 $dc_{ik} = 0$ の場合、演算器 f_i の演算結果を直接ハドル h_k のローカルレジスタに $S_f(f_i)$ ステップで転送する。 $dc_{ik} \geq 1$ の場合、最初に演算器 f_i から自身のハドル h_j のローカルレジスタに $S_f(f_i)$ ステップでデータを転送する。次のステップで、ハドル h_j のローカルレジスタからハドル h_k のローカルレジスタに $\left\lceil \frac{D_r(h_j) + D_{tr}(h_j, h_k)}{T_{clk}} \right\rceil$ ステップかけて転送する。

2.4 HDR アーキテクチャを対象とした基板バイアス制御時のリークエネルギーの最小化問題

以上より、本稿で対象とする高位合成問題を次のように定義する。

定義 1. 本稿で対象とする高位合成問題とは、CDFG, 演算器集合, クロック周期制約, 基板バイアス電圧集合が与えられたとき、ティピカルケース高位合成のレイテンシを超えない範囲内で、基板バイアス制御時のリークエネルギーを最小化するように CDFG をスケジューリングおよびバインディングし、各ハドルのワースト印加基板バイアス電圧を決定し、各演算器をハドルに割り当て、RTL 回路およびハドル配置を出力することである。

3. アルゴリズム

フロアプランを考慮した高位合成の主要な工程には、スケジューリング、バインディング、コントローラ合成、フロアプランがある。スケジューリング、バインディング、コントローラ合成およびフロアプランは互いの結果に影響を及ぼし合うため、これらの工程の最適な順序関係は合成前に確定できない。したがって、[4, 6, 7] と同様に、スケジューリング、バインディング、コントローラ合成、フロアプランについて反復フローを採用する。

本稿で解決すべき問題は、基板バイアス制御時のリークエネルギーが最小となるように、ワースト印加基板バイアス電圧を各ハドルに対して定めることである。演算の実行タイミングはスケジューリングで決定されるため、ワースト印加基板バイアス電圧もスケジューリング段階で決定することが適する。

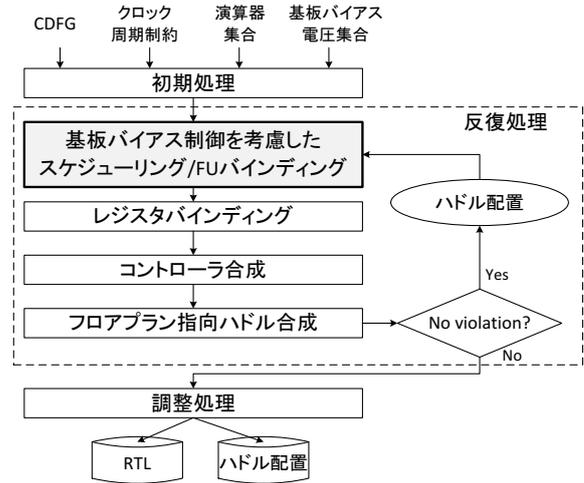


図 2: 合成フロー。

以上の議論より、合成フローを図 2 に示す。初期処理では、各ハドルに対し 1 つの演算器を割り当て、配置情報の初期解を生成する。各ハドルは重なっており、すべてのハドル間の配線長は 0 とする。反復処理では、タイミングを満たすようなスケジューリング・バインディング結果とハドルの構成・配置を決定する。調整処理では、反復処理で得られた結果から、正確な面積見積もりを決定する。初期処理、調整処理、レジスタバインディング、コントローラ合成、フロアプラン指向ハドル合成は [4] と同様の処理を実行する。以下、基板バイアス制御を考慮したスケジューリング/FU バインディングを提案する。

3.1 基板バイアス制御を考慮したスケジューリング/FU バインディング

基板バイアス制御を考慮したスケジューリング/FU バインディングでは、CDFG, クロック周期制約, ハドルの構成, ハドルの配置情報, 基板バイアス電圧集合を入力とし、各演算ノードにコントロールステップおよび演算器を割り当て、各ハドルにワースト印加基板バイアス電圧を定める。[6] に基づき、データ転送サイクル行列に基づいたリストスケジューリングを実行する。もし製造ばらつきにより設計時に想定した遅延よりも遅延が増大した場合、データ転送サイクル行列通りにデータ通信が行われずタイミング違反が生じる可能性がある。このような場合、適切な基板バイアス電圧を印加しタイミング違反を補償する必要がある。

3.2 ワーストケース遅延におけるタイミング違反補償

データ転送サイクル行列 DC に基づいたスケジューリング/FU バインディング結果に対し、タイミング違反が起きない条件は、 $\forall f_i \in F, \forall h_k \in H$ に対し、演算器 f_i からハドル h_k への通信が存在する場合に、式 (5), (6) を満たすことである。

(1) $dc_{ik} = 0$ の場合

$$S_f(f_i) \times T_{clk} - D_f(f_i) - D_r(h_j) - D_{tr}(h_j, h_k) \geq 0 \quad (5)$$

(2) $dc_{ik} \geq 1$ の場合

$$S_f(f_i) \times T_{clk} - D_f(f_i) - D_r(h_j) \geq 0 \quad (6)$$

$$dc_{ik} \times T_{clk} - D_r(h_j) - D_{tr}(h_j, h_k) \geq 0$$

Step 1: 入力スケジューリング/FU バインディング結果に対し、すべてのモジュールの遅延がワーストケースとなる場合を想定する。各ハドル h_j に対し、 $HV^{wst}(h_j) \leftarrow 0V$ とする。

Step 2: 式 (5) を満たすように、各演算器 $f_i \in F$ に対し以下の処理を実行。

- (1) 演算器 f_i からの通信が存在するハドル $h_k \in H$ に対し、以下の処理を実行。ハドル $h_j = Hud(f_i)$ とする。
- (2) ハドル h_j に対し、式 (5) を満たす最小のワースト印加基板バイアス電圧 $v_{min} \in V$ を選択。
- (3) 現在設定されている $HV^{wst}(h_j)$ よりも v_{min} が大きい場合、 $HV^{wst}(h_j) \leftarrow v_{min}$ とする。(2) で式 (5) を満たす v_{min} が存在しなかった場合、 $HV^{wst}(h_j) \leftarrow \max V$ として (4) を実行。
- (4) ハドル h_k に対して (2), (3) と同様に $HV^{wst}(h_k)$ を定める。

Step 3: Step 2 と同様の処理を式 (6) を満たすように実行。

図 3: ワースト印加基板バイアス電圧決定処理。

あるスケジューリング/FU バインディング結果が与えられたとき、ワーストケース遅延の下でも式 (5), (6) を満たし、リーク電力が最小となるようなワースト印加基板バイアス電圧を、各ハドルに対して定める必要がある。

ワースト印加基板バイアス電圧決定処理を図 3 に示す。入力スケジューリング/FU バインディング結果に対し、すべてのモジュールの遅延がワーストケースである場合を想定し、 $HV^{wst}(h_j) \leftarrow 0V$ と初期化する (Step 1)。最初に、演算器-レジスタ間のデータ転送のタイミング違反を補償する (Step 2)。 $S_f(f_i) \times T_{clk}$ 以内で、演算器 f_i の演算を終了し結果をハドル h_j のレジスタに転送できるように、最小の基板バイアス電圧を V から選択する (Step 2-(1), (2))。現在設定されている基板バイアス電圧よりも選択した基板バイアス電圧が大きい場合、更新する (Step 2-(3))。 V に存在する最大の基板バイアス電圧をハドル h_j に設定しても条件が満たされない場合、受信側のハドル h_k に対しても同様の操作を実行する (Step 2-(4))。次に、レジスタ-レジスタ間データ転送のタイミング違反を補償する (Step 3)。演算器 f_i から自身のハドル h_j のレジスタへのデータ転送のタイミング違反の補償は Step 2 で完了している。 $dc_{ik} \times T_{clk}$ 以内で、ハドル h_j のレジスタからハドル h_k のレジスタにデータを転送できるように、Step 2 と同様の手順を実行する。

3.3 ワースト印加基板バイアス電圧の最小化

図 3 のアルゴリズムにより、あるスケジューリング/FU バインディング結果に対し、ワーストケース遅延の下でもタイミングを満たす最小の基板バイアス電圧は決定される。スケジューリング/FU バインディング結果自体を、増大し得るリークエネルギーが削減されるように生成することで、さらなるエネルギーオーバーヘッドの削減が期待できる。

演算器 f_i からハドル h_k へのデータ転送において、タイミング違反が生じない確率 $P_{trans}(f_i, h_k)$ を算出する。

(1) $dc_{ik} = 0$ の場合

演算の実行ステップ内で、演算器 f_i からハドル h_k のローカルレジスタに対して演算結果を転送する。演算器 f_i の遅延とレジスタの遅延の合計値が、 $d_{border} = S_f(f_i) \times T_{clk} - D_{tr}(h_j, h_k)$ を超えた場合に式 (5) を満たさなくなる。したがって、 $P_{trans}(f_i, h_k)$ を次式で算出する。

$$P_{trans}(f_i, h_k) = \int_0^{d_{border}} G_{f_i r}(t) dt \quad (7)$$

$G_{f_i r}(t)$ は演算器 f_i とレジスタ r の遅延の和の確率密度関数を表す。

(2) $dc_{ik} \geq 1$ の場合

演算の実行ステップ内で、演算器 f_i から自身のハドル h_j のローカルレジスタに対して演算結果を転送する必要がある。したがって、式 (7) を用いてタイミング違反が起きない確率 $P_{trans}(f_i, h_j)$ を算出できる。次のステップで、レジスタ間データ転送を実行する。ハドル h_j のローカルレジスタからデータを読み出し、ハドル h_k のローカルレジスタに転送するのに要する時間は、 $dc_{ik} \times T_{clk}$ である。したがって、レジスタの遅延値が、 $d_{border} = dc_{ik} \times T_{clk} - D_{tr}(h_j, h_k)$ を超えた場合に、式 (6) を満たさなくなる。したがって、 $P_{trans}(f_i, h_k)$ を次式で算出する。

$$P_{trans}(f_i, h_k) = P_{trans}(f_i, h_j) \times \int_0^{d_{border}} G_r(t) dt \quad (8)$$

$G_r(t)$ はレジスタの遅延の確率密度関数を表す。

以上の議論より、ハドル h_j に基板バイアス電圧を印加しなくてよい確率 $P(h_j)$ を次式で定義する。

$$P(h_j) = \min_{f_i \in FH(h_j)} \left\{ \min_{h_k \in H} P_{trans}(f_i, h_k) \right\} \quad (9)$$

$1 - P(h_j)$ は、ハドル h_j に基板バイアス電圧を印加しなければならない確率を表す。

本稿では、演算器 f_i のリーク電力の見積もり値として、ワーストケース遅延時に基板バイアス電圧を印加した場合のリーク電力を用いる。演算器 f_i に基板バイアス電圧 v を印加した時のリーク電力を $L_{f_i}(v)$ と表す。ハドル h_j において、基板バイアス電圧 $v_j = HV^{wst}(h_j)$ を印加した時のリーク電力の増大量の見積もり値 $\Delta L_{est}(h_j)$ を、次式で定義する。

$$\Delta L_{est}(h_j) = \sum_{f_i \in FH(h_j)} (L_{f_i}(v_j) - L_{f_i}(0V)) \quad (10)$$

ハドル h_j に対する基板バイアス電圧の印加コスト $Cost(h_j)$ を次式で定義する。

$$Cost(h_j) = (1 - P(h_j)) \times \Delta L_{est}(h_j) \quad (11)$$

$Cost(h_j)$ は、ハドル h_j におけるリーク電力の増大量の見積もり期待値を表す。したがって、 $Cost(h_j)$ が大きいハドルほど、可能な限り $HV^{wst}(h_j)$ を小さくする必要がある。以上より、全体の基板バイアス電圧の印加コスト $Cost_{all}$ を次式で定義する。

$$Cost_{all} = \sum_{h_j \in H} Cost(h_j) \quad (12)$$

3.4 アルゴリズム

提案アルゴリズムを図 4 に示す。最初に、各ハドル $h_j \in H$ に対し $HS(h_j)$ の値を typ とする (Step 1)。入力 CDFG に対し、 HS に基づいた遅延を用いて、スケジューリング/FU バインディング [6] を実行する (Step 2)。これにより、全ての演算器の遅延にティピカルケースの遅延を想定したスケジューリング/FU バインディング結果を得る。結果の総コントロールステップをレイテンシ制約 (cs_{max}) として、以降の探索を実行する。このスケジューリング/FU バインディング結果に対して図 3 を実行することで、ワースト印加

- Step 1:** 各ハドル $h_j \in H$ に対し, $HS(h_j) \leftarrow typ$ とする.
Step 2: HS に基づきスケジューリング/FU バインディング [6] を実行. 結果の CS 数を cs_{max} とする.
Step 3: ワorst印加基板バイアス電圧決定処理を実行.
Step 4: 各ハドル $h_j \in H$ に対し, $Cost(h_j)$ を計算.
Step 5: すべてのハドル $h_j \in H$ が選択済みでないならば, 以下の処理を実行. そうでないならばアルゴリズムを終了.
 (1) 未選択のハドルの中で, $Cost(h_j)$ が最大のハドル h_j を選択.
 (2) $HS(h_j) \leftarrow wst$, $v_{max} \leftarrow HV^{wst}(h_j)$, $v \leftarrow 0V$.
 (3) $v < v_{max}$ であるならば以下の処理を実行.
 (a) $HV^{wst}(h_j) \leftarrow v$ とし, DC を更新.
 (b) HS と HV^{wst} に基づきスケジューリング/FU バインディング [6] を実行. 結果の CS 数を cs とする..
 (c) ワorst印加基板バイアス電圧決定処理を実行.
 (d) $cs < cs_{max}$ かつ $Cost_{all}$ が減少したならば, h_j 以外のハドルに対してもコストを更新し, $HV^{wst}(h_j) \leftarrow v$ とする. $HS(h_j)$ の値を wst に固定し, Step 5 に戻る. そうでなければ結果と変更した変数を元に戻し, v に V から次に大きい値を代入する.
 (4) $v = v_{max}$ ならば $HS(h_j) \leftarrow typ$ とする.

図 4: 提案アルゴリズム.

基板バイアス電圧を各ハドルに定める (Step 3). 次に, 各ハドルに対して $Cost(h_j)$ を計算する (Step 4). $Cost(h_j)$ が最大のハドル h_j を選択する (Step 5-(1)). ハドル h_j のモジュールの遅延がワorstケースであると想定し, 印加する基板バイアス電圧 v を $0V$ とする (Step 5-(2)). ハドル h_j に基板バイアス電圧 v を印加し, DC を更新する (Step 5-(3)-(a)). HS に基づいた遅延を用いて, スケジューリング/FU バインディング [6] を実行し, 再度ワorst印加基板バイアス電圧を設定する (Step 5-(3)-(b), (c)). 結果の総 CS が cs_{max} 以下で, $Cost_{all}$ が減少したならばこの結果を採用する. そうでない場合, V の中から次に大きい電圧を選択して v の値とし (Step 5-(3)-(d)), 同様な操作を実行する. $v = v_{max}$ となった場合, $HS(h_j) \leftarrow typ$ (Step 5-(4)) として, 次にコストが大きいハドル h_k を選択し, 同様な操作を実行する.

4. 計算機実験結果

提案手法を C++ 言語を用いて計算機上に実装した. 評価対象のアプリケーションとして, DCT (ノード数 48), 7 次 FIR フィルタ (ノード数 75), EWF3 (ノード数 102) [4] を使用した. クロック周期制約は 2.9ns とした. RTL モジュール情報を表 2 に示す. RTL モジュールのビット幅は 16bit, 動作電圧は 1.0V, 動作温度は 25°C とした. 各 RTL モジュールの遅延分布はガウス分布を仮定し, ティピカルケース, ワorstケースの遅延を, μ , $\mu + 3\sigma$ とする [8]. 表 2 の σ は, しきい値電圧および移動度にガウス分布を想定したグローバルばらつきを変動係数を 0.1 として与え, モンテカルロ解析により得た. [9] を参考に, 各モジュールに対し, 高位合成段階で使用される基板バイアス電圧印加-遅延・リーク電力特性を測定した. 加算器の場合を図 5 に示す. ただし $V_{bs}=0V$ の時の遅延・リーク電力で正規化している. 以上のデータは, Synopsys 社の Design Compiler および HSPICE による測定結果より得た *2. 図 2 のコン

*2 本稿で提案する高位合成手法は, ターゲットプロセスとして SOI プロセスを想定している. したがって, RTL モジュール単位で PMOS, NMOS の両方に同一割合で基板バイアス電圧を印加するが, バルクプロセスに必要とされるようなトリプルウェルプロセスによる面積オーバーヘッドを考慮していない. しかし SOI プロセスのライブラリを所持していないため, 本実験では 90nm のバルクプロセスのライブラリを使用した. SOI プロセスに対し提案手法を適用した場合, 同様の改善率が得られると仮定する.

表 2: 演算器情報

	Area [μm^2]	Delay		Dynamic energy [fJ]	Leak power [μW]
		Mean (μ) [ns]	Std. dev. (σ) [ns]		
Adder	1028	2.09	0.156	269.643	4.781
Multiplier	6831	4.39	0.336	3183.285	29.351
Register	725	0.62	0.0554	57.054	8.982
MUX	320	0.46	0.0300	90.272	0.928
Memory	-	4.39	-	-	-

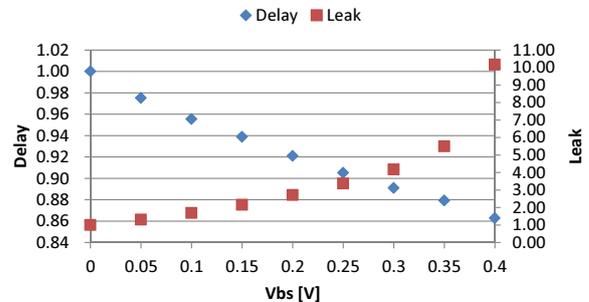


図 5: 高位合成段階で使用される加算器の基板バイアス電圧-遅延・リーク電力特性.

トローラ合成において, コントローラの面積は Synopsys 社の Design Compiler により論理合成して算出した. 配線遅延は配線長の 2 乗に比例すると仮定し, 250 μm あたり 1ns と仮定する [4].

比較手法は次の通りである.

Typical [4]: モジュールの遅延にティピカルケースを想定し, HDR アーキテクチャを対象とした高位合成手法 [4] を実行する. タイミング違反が起きた場合, 基板バイアス制御による遅延補償を実行する.

Worst [4]: モジュールの遅延にワorstケースを想定し, HDR アーキテクチャを対象とした高位合成手法 [4] を実行する. タイミング違反が起きた場合でも, 基板バイアス制御による遅延補償は実行しない.

Ours: 提案手法を実行する. タイミング違反が起きた場合, 基板バイアス制御による遅延補償を実行する.

最初に基板バイアス制御自体の有用性を, Worst [4] とレイテンシおよびタイミング歩留りを比較することで確認する. 次に, 提案手法の基板バイアス制御時のエネルギーオーバーヘッドが Typical [4] と比較し, 小さいことを示す.

実験結果を表 1 に示す. “Res. of HLS” は, 高位合成自体の結果である. “CS” は合成結果の総コントロールステップを表す. 提案手法および Typical [4] は, Worst [4] と比較し低いレイテンシを実現できる.

次に各手法に対し, 図 6 を実行する. 図 6 のモンテカルロ解析では, しきい値電圧および移動度にガウス分布を想定したグローバルばらつきを変動係数を 0.1 として与えた. N_{max} は 1000 とした. ここで, “チップが受け入れられた” とは, そのチップでタイミング違反が起きなかったことを意味する. 表 1 の “Timing yield” は, 全体のチップ数 (N_{max}) に対し, 受け入れられたチップの割合である. “Before BB” は, 最初の段階で受け入れられたチップ数 (N_{or}) の割合である. “After BB” は, 基板バイアス制御による遅延補償後に受け入れられたチップ数 (N_{bb}) と N_{or} の合計の割合である. 提案手法および Typical [4] は, 遅延補償前では Worst [4] よりも歩留まりが低い. しかし遅延補償後には Worst [4] と同じ程度の高い歩留まりを実

表 1: 計算機実験結果

Apps.	FUs	Algo.	Res. of HLS			Timing yield		Mean of energy		
			#CSs	ite.	CPU times [sec]	Before BB N_{or}/N_{max}	After BB $(N_{or} + N_{bb})/N_{max}$	Dynamic [pJ]	Leak [pJ]	All [pJ]
DCT	Add×3	Typical [4]	16	2	173.33	0.717	0.935	101.93	29.56	131.49
	Mult×3	Worst [4]	25	7	508.32	0.993	-	100.35	19.68	120.03
		Ours	16	9	600.64	0.741	0.954	101.39	18.31	119.70
EWF3	Add×3	Typical [4]	89	3	357.19	0.729	0.959	217.33	188.82	406.15
	Mult×2	Worst [4]	123	6	677.93	1.000	-	200.84	72.24	269.24
		Ours	89	2	231.8	0.778	0.992	218.34	87.09	305.43
FIR	Add×3	Typical [4]	36	2	228.55	0.777	0.973	155.12	39.63	194.75
	Mult×3	Worst [4]	47	6	580.43	0.996	-	156.65	36.21	192.86
	Mem×1	Ours	36	3	245.3	0.746	0.947	155.45	33.35	188.80

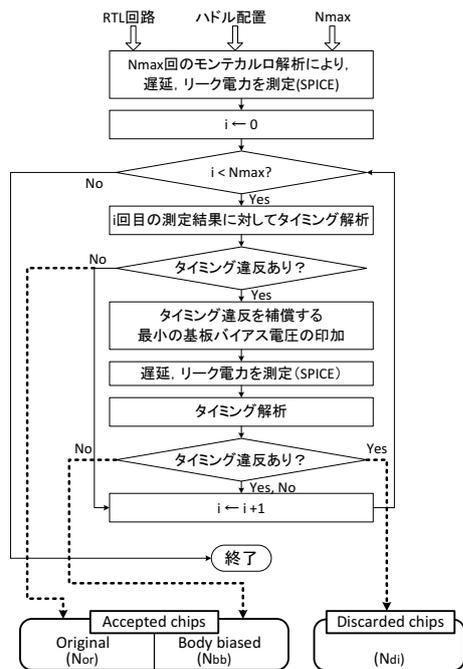


図 6: 評価フロー.

現している。したがって、基板バイアス制御による遅延補償はワーストケース設計よりも高い性能を引きだし、かつ高い歩留まりを実現できる。

次に、提案手法によるリークエネルギーの削減効果を検証する。同一のばらつきデータに対する比較を行うため、 i 回目の結果に対し、提案手法と Typical [4] とともに基板バイアス制御による性能補償が実行された場合のエネルギーに注目する。 $0 \leq i < N_{max}$ に対し、2つの手法とともに図6の Accepted chips の Body biased に入ったチップのエネルギーの平均値を算出した結果が、表1の“Mean of energy”である。提案手法は Typical [4] と比較し、最大53.9%、平均35.9%のリークエネルギーの平均値の削減を実現した。

以上より、提案手法は製造ばらつき環境下でも性能低下が小さく、高い歩留まりでエネルギーオーバーヘッドの小さい手法であると言える。

5. おわりに

本稿では、基板バイアス制御による遅延ばらつき補償および配線遅延を考慮した低エネルギーオーバーヘッド指向の高位合成手法を提案した。提案手法はティビカルケース高位合成と比較し、最大53.9%リークエネルギーの削減を

確認した。今後の課題として、より詳細な問題設定や、消費電力のばらつきへの対応が挙げられる。

謝辞

本研究は、独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) の先導的産業技術創出事業の支援を受けて行われた。

参考文献

- [1] M. Mani, A. K. Sing, and M. Orshansky, “Joint design-time and post-silicon minimization of parametric yield loss using adjustable robust optimization,” in *Proc. of IEEE/ACM international conference on Computer-aided design 2006*, 2006, pp. 19–26.
- [2] F. Wang, X. Wu, and Y. Xie, “Variability-driven module selection with joint design time optimization and post-silicon tuning,” in *Proc. of ASP-DAC 2008*, 2008, pp. 2–9.
- [3] D. Helms, O. Meyer, M. Hoyer, and W. Nebel, “Voltage- and abb-island optimization in high level synthesis,” in *Proc. of ACM/IEEE International Symposium on Low Power Electronics and Design 2007*, 2007, pp. 153–158.
- [4] S. Abe, Y. Shi, M. Yanagisawa, and N. Togawa, “Mh⁴: multiple-supply-voltages aware high-level synthesis for high-integrated and high-frequency circuits for hdr architectures,” *IEICE Electronics Express*, vol. 9, no. 17, pp. 1414–1422, 2012.
- [5] M. Zhao, A. Orailoglu, and C. J. Xue, “Profit maximization through process variation aware high level synthesis with speed binning,” in *Proc. of DATE 2013*, 2013, pp. 176–181.
- [6] A. Ohchi, S. Kohara, N. Togawa, M. Yanagisawa, and T. Ohtsuki, “Floorplan-driven high-level synthesis for distributed/shared-register architectures,” *Information and Media Technologies*, vol. 3, no. 4, pp. 691–703, 2008.
- [7] S. Abe, M. Yanagisawa, and N. Togawa, “Energy-efficient high-level synthesis for hdr architectures,” *IPSSJ Trans. on System LSI Design Methodology*, vol. 5, pp. 106–117, 2012.
- [8] G. Lucas and D. Chen, “Variation-aware layout-driven scheduling for performance yield optimization,” in *Proc. of ICCAD 2010*, 2010, pp. 17–24.
- [9] S. H. Kulkarni, D. M. Sylvester, and D. T. Blaauw, “Design-time optimization of post-silicon tuned circuits using adaptive body bias,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 481–494, 2008.