

HDR アーキテクチャを対象とした 複数電源電圧指向の低電力化高位合成手法

阿部 晋 矢^{†1} 柳澤 政 生^{†2} 戸川 望^{†1}

携帯機器の駆動時間や発熱が問題となる現代、低電力化を意識した LSI 設計が必要である。半導体の微細化技術の向上のため、ゲート遅延に対する配線遅延の割合が増加し、配線遅延を考慮した設計も必要である。システム LSI の設計手法として高位合成があるが、低電力化と配線遅延の双方を意識した高位合成として HDR アーキテクチャを対象とした複数電源電圧指向の高位合成がある。しかし、これはスケジューリング/FU バインディングの際、直接的に消費エネルギーを最小化するのではなく、実行時間の最小化を目的とすることで 2 次的に消費エネルギーを削減している。本稿では、HDR アーキテクチャを対象とした、複数電源電圧を考慮した消費エネルギーの最小化を目的とするスケジューリング/FU バインディングを提案する。計算機実験により提案手法は、従来のレジスタ分散型アーキテクチャと比較して最大 45.1% 程度消費エネルギーを削減でき、従来の HDR アーキテクチャを対象とした手法と比較して最大 15.9% 程度消費エネルギーを削減できることを確認した。

Multiple Supply Voltages aware High-level Synthesis for HDR architecture

SHIN-YA ABE,^{†1} MASAO YANAGISAWA^{†2}
and NOZOMU TOGAWA^{†1}

As battery runtime and overheating problems for portable devices become unignorable, energy-aware LSI design is strongly required. Moreover, an interconnect delay should be explicitly considered there because it exceeds a gate delay as the semiconductor devices are downsized. We must take account of energy efficiency and interconnect delay even in high-level synthesis. Recently, a *huddle-based distributed-register architecture (HDR architecture)*, which is a kind of island-based distributed-register architecture for multi-cycle interconnect communications, and its associated synthesis algorithm have been proposed. The algorithm is composed of scheduling/FU binding, huddling, un-huddling, and floorplanning. However, the original scheduling/FU binding does not minimize energy consumption directly but minimizes execution time. In

this paper we propose a new scheduling/FU binding algorithm whose purpose is the minimization of energy consumption considering multiple supply voltages for HDR architectures. Experimental results show that our algorithm achieves 45.1 % energy-saving compared with the conventional distributed-register architectures and conventional algorithms, and 15.9 % energy-saving compared with the conventional algorithm for HDR architecture.

1. はじめに

携帯機器に搭載される LSI では駆動時間を伸ばすことが問題となり、据え置き器に搭載される LSI では動作時の発熱が問題となっている。双方を解決するため低電力化を意識した LSI 設計が求められる。その他の LSI 設計時の問題として、配線遅延がある。配線遅延はモジュール間の通信に必要な遅延である。LSI 全体の遅延を考えた場合、半導体の微細化技術によりゲート遅延が減少する一方、配線遅延の占める割合は増加傾向にある。今後もこの傾向が継続すると予想されるため、配線遅延を考慮した LSI 設計が必要となる。LSI の設計手法として高位合成が目されるが、低電力化と配線遅延は高位合成においても意識すべきである。本稿では、低電力化および配線遅延を考慮した高位合成手法に着目する。

低電力化を意識した高位合成手法として、複数電源電圧を考慮する手法^{4),9)}がある。電圧を低くできれば消費電力を抑えられるが、回路の遅延が増加する。複数電源電圧では回路のクリティカルパスに高電圧を用い、速度を求められない部分に低電圧を割り当てることで低電力化を図る。しかし、これらの手法は電力の最適化のみを目的とし、配線遅延は考慮されておらず配線遅延の割合が増加する現状に則さない。また、これらの手法は演算器の電圧も制約として与えており、消費エネルギーを最小化する演算器の電圧を求めることはできない。演算器の電圧を決定する LSI 設計の方法としては、高位合成より後の工程であるフロアプランで決定する方法^{3),10)}がある。しかし、フロアプラン時には演算の実行タイミングが決定しており、タイミング制約を満たした上で電圧を大幅に下げることが難しい。そのため、高位合成時に複数電源電圧を適用した場合と比較し、消費エネルギーの削減量が少なくなっている。

^{†1} 早稲田大学大学院基幹理工学研究科情報理工学専攻

Dept. of Computer Science and Engineering, Waseda University.

^{†2} 早稲田大学大学院基幹理工学研究科電子光システム学専攻

Dept. of Electronic and Photonic Systems, Waseda University.

配線遅延を考慮した高位合成手法として、フロアプランを意識する高位合成手法がある^{1),2),5),6)}。これらは従来のレジスタ集中型アーキテクチャではなく、レジスタを演算器に分散させレジスタ-レジスタ間の通信にクロックサイクルを割り当てる、レジスタ分散型アーキテクチャを基にしている。6)では Generalized Distributed-Register アーキテクチャ(以下 GDR)を提案している。GDR はモジュールを個別にフロアプランすることで高速かつ小面積な解を得ることができる。しかし、モジュールを個別に扱う GDR に新たなモジュールを追加することは難しく、複数電源電圧と GDR を同時に扱うことは難しい。2)では Regular Distributed-Register アーキテクチャ(以下 RDR)を提案している。RDR はチップを均一の区画に分割することで、高位合成中の配線遅延の予測を容易にしている。また、区画内のモジュールの配置は抽象化され、モジュールの変化にも対応しやすい。区画によるモジュールの抽象化で、RDR には多くの派生アーキテクチャ^{2),7)}がある。しかし、RDR はチップを一定の大きさに分割するため、面積のオーバーヘッドが大きく結果として消費電力も大きくなる。

配線遅延と低電力化を同時に考慮した高位合成手法として、HDR アーキテクチャを対象とした複数電源電圧指向の高位合成¹⁾を提案した。1)では低電力化手法を適用しやすいレジスタ分散型アーキテクチャとして HDR アーキテクチャを提案している。HDR アーキテクチャはハドル (Huddle) と呼ぶ可変の区画により、各モジュールを抽象化することでレベルコンバータの様な新たなモジュールの追加を容易にする。また、ハドルは可変の区画のため、RDR と比較し面積のオーバーヘッドが少なく消費エネルギーの少ないアーキテクチャとなっている。1)では HDR アーキテクチャを対象とした複数電源電圧による低電力化高位合成手法が提案されている。1)では複数電源電圧による消費エネルギーの削減をスケジューリング/FU バインディングで考慮している。1)のスケジューリング/FU バインディングでは、直接消費エネルギーの削減を目的とするのではなく、実行時間の最小化を目的とすることで2次的に消費エネルギーを削減している。そのため、1)は消費エネルギーの最小化が不十分である。

以上の背景から、本稿では HDR アーキテクチャを対象とした複数電源電圧指向の高位合成のうち、消費エネルギー最小化を目的としたスケジューリング/FU バインディングを提案する。提案手法は演算器集合とハドルの配置から、配線遅延を考慮したスケジューリング/FU バインディングを実行する。その際、時間制約の中で動的消費エネルギーを最小化するように演算に電圧を割り当てる。それにより、従来手法¹⁾と比較し最大 10% 消費エネルギーを削減できた。

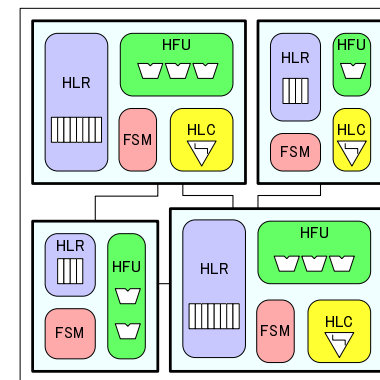


図 1 HDR アーキテクチャの構成.
Fig.1 HDR architecture.

2. HDR アーキテクチャ

GDR は高速かつ小面積という利点があり、RDR はモジュールの追加がしやすいという利点がある。しかし、GDR はモジュールの追加が難しく、RDR は面積が大きいという欠点がある。それぞれの欠点から、既存のレジスタ分散型アーキテクチャは低電力化に向かない。そこでモジュールを追加しやすく、かつ小面積で消費電力の少ないレジスタ分散型アーキテクチャ、Huddle-based Distributed Register アーキテクチャ(以下 HDR)が提案された¹⁾。HDR は GDR にハドル (Huddle) という区画を導入し、各モジュールを抽象化したアーキテクチャである。ハドル導入によるモジュールの抽象化で、RDR と同様にモジュールを容易に追加できる。ハドルはクロック周期により決定される範囲内において任意の矩形を取り、演算器やレジスタ、コントローラ、レベルコンバータを共有する。任意の矩形を取るハドルに対しフロアプランするため、GDR と同様に小面積で消費電力の少ないアーキテクチャとなる。

ハドルは以下の要素からなる。

- Huddled Local Register (HLR)
各ハドル専用のローカルレジスタとマルチプレクサの集合。
- Huddled Functional Unit (HFU)
ハドルに集められた演算器の集合。ハドル内で処理する演算に必要な演算器を必要数持

ち、同一のハドル内の HLR のみにアクセスできる。

- Finite State Machine (FSM)

各ハドル専用のコントローラ。同一ハドルの HFU と HLR を制御する。

- Huddled Level Converter (HLC)

ハドルに集められたレベルコンバータの集合。電圧の異なるハドルとデータ転送を行う際、HLC を用いる。

ハドルは図 1 の様に配置する。同一ハドル内の HFU でデータを処理する場合、ハドル内の HLR を使うことでデータ転送時間は無視できる。異なるハドルの HFU 同士でデータ通信する場合、HLR 間データ転送を行う。HLR 間データ転送をする際、各ハドルの電圧が異なる場合 HLC を用いる。

3. 問題の定式化

提案手法では高位合成の対象として HDR アーキテクチャ¹⁾を対象とする。HDR アーキテクチャの構成を図 1 に示す。

入力としてコントロールデータフローグラフ (以下 CDFG) を与える。CDFG $G(N, E)$ は有向グラフで表現される。 N は演算ノード N_o と制御ノード N_c からなる。 E はデータフローエッジ E_d とコントロールフローエッジ E_c からなる。クロック周期制約として T_{clk} 、ステップ制約 S_{max} を与える。

入力として p 個の演算器の集合 $F = \{f_1, \dots, f_p\}$ を与え、演算器 f_i の遅延を $D_f(f_i)$ で表す。 $S_f(f_i)$ は f_i の必要ステップ数を表し、クロック周期 T_{clk} より $S_f(f_i) = \lceil D_f(f_i)/T_{clk} \rceil$ と計算される。演算器 f_i が 1 回の処理につき消費するエネルギーを $E(f_i)$ で表す。

q 個 ($q \leq p$) のハドル $H = \{h_1, \dots, h_q\}$ に各演算器を割り当てる。演算器 f_i を割り当てるハドルを $Hud(f_i)$ で表す。一方、ハドル h_j に割り当てられた演算器の集合を $F(h_j)$ で表す。ハドル h_j における HLR の遅延を $D_{reg}(h_j)$ で表す。

各ハドルに割り当てる電圧は $v_l, v_m, v_h (v_l < v_m < v_h)$ の 3 種類とする。ハドル h_j の電圧を $V(h_j)$ で表す。 v_l から v_m へ電圧を変更する際のレベルコンバータの遅延を $D_{lc}(v_l, v_m)$ で表す。 v_l から v_h など他の電圧についても同様である。

演算器 f_i における、クロック周期から演算処理に必要な時間を除いた時間を以下の式で表す。

$$Slack(f_i) = T_{clk} \cdot S_f(f_i) - D_f(f_i) \quad (1)$$

RDR のアイランド²⁾を参考に $Slack(f_i)$ から矩形であるハドル $h_j = Hud(f_i)$ の幅、高さ

を以下の式で求める。

$$2 \cdot D_w(W(h_j) + H(h_j)) + D_{reg}(h_j) \leq \min_{f_i \in F(h_j)} \{Slack(f_i)\} \quad (2)$$

式 (2) において $W(h_j)$ はハドル h_j の幅、 $H(h_j)$ はハドル h_j の高さを表す。 $D_w(x)$ は距離 x における配線遅延を表す。配線遅延は距離の 2 乗に比例するとし、配線遅延係数 C_d を用いて $D_w(x) = C_d x^2$ で計算される。式 (2) をハドルサイズ制約と呼ぶ。

演算器 f_i からハドル h_k へデータ転送する場合を考える。 $Hud(f_i) = h_j$ とする時、 h_j と h_k の中心間のマンハッタン距離を $Dist(h_j, h_k)$ で表すと、 h_j, h_k 間の配線遅延は $D_w(Dist(h_j, h_k))$ となる。 f_i からデータを転送し、 h_k の HLR へ書き込むまでの時間は以下の式で表される。

$$Tr(f_i, h_k) = D_w(Dist(h_j, h_k)) + D_{lc}(V(h_j), V(h_k)) + D_{reg}(h_k) \quad (3)$$

$Slack(f_i)$ および $Tr(f_i, h_k)$ より f_i から h_k へデータ転送する際に必要なクロック数 $DT(f_i, h_k)$ は以下の式で表される。

$$DT(f_i, h_k) = \begin{cases} 0 & (Slack(f_i) \geq Tr(f_i, h_k)) \\ \lceil Tr(f_i, h_k)/T_{clk} \rceil & (Slack(f_i) < Tr(f_i, h_k)) \end{cases} \quad (4)$$

このとき、 DT とは p 行 q 列の行列で、 i 行 k 列の要素が $DT(f_i, h_k)$ となるものとする。

以上より、HDR アーキテクチャを対象とした複数電源電圧高位合成問題を次のように定義する。

定義 1. HDR アーキテクチャを対象とした複数電源電圧高位合成問題とは、CDFG、クロック周期制約、ステップ制約、演算器の集合が与えられた時、消費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、各演算器をハドルに割り当て、ハドルに電圧を割り当てることである。 □

4. HDR アーキテクチャを対象とした低電力化高位合成アルゴリズム

4.1 高位合成アルゴリズム全体の構成

HDR アーキテクチャを対象とした複数電源電圧を考慮した高位合成として 1) がある。1) では複数電源電圧による消費エネルギーの削減をスケジューリング/FU バインディングで考慮している。1) のスケジューリング/FU バインディングでは、直接消費エネルギーの削減を目的とするのではなく、実行時間の最小化を目的とすることで 2 次的に消費エネルギー

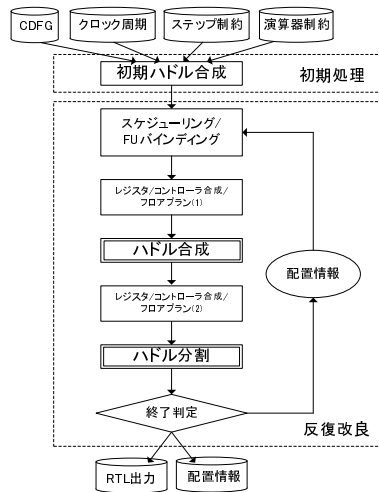


図 2 HDR 合成アルゴリズム.

Fig. 2 Synthesis algorithm for HDR architecture.

を削減している. そのため, 1) は消費エネルギーの最小化が不十分である. そこで本稿では, 演算の動的エネルギー最小化することで, 全体の消費エネルギー最小化を目指すスケジューリング/FU バインディングを提案する.

HDR を対象とする高位合成は以下の要素から構成される.

- 初期ハドル合成
- スケジューリング/FU バインディング
- レジスタ/コントローラ合成/フロアプラン
- ハドル合成
- ハドル分割

スケジューリング/FU バインディングについては本稿で提案するアルゴリズムを適用し, 他の要素については 1) を適用する.

最終的な高位合成アルゴリズムを図 2 に示す. なお, 説明を簡単にするため本章では入力力は DFG のみを考えるが, 入力を CDFG としても同様である.

4.2 スケジューリング/FU バインディング

スケジューリング/FU バインディングの入力は, クロック周期制約 T_{clk} , ステップ制約

S_{max} , CDFG $G(N, E)$, 演算器数, ハドルの構成, ハドルの電圧, 配置情報である. 出力は演算ノード集合 N を実行するコントロールステップ, 実行する演算器, ハドルの電圧である. ただし, 第 1 回目の反復はフロアプランの実行前のため, 全てのハドル間の配線遅延を 0 とする.

HDR アーキテクチャを対象とした複数電源電圧を考慮した高位合成として 1) が提案された. 1) では, スケジューリング/FU バインディングで演算に電圧を割り当て, スケジューリング/FU バインディングで決定した電圧をもとに他の要素を実行する. 1) のスケジューリング/FU バインディングでは, 直接消費エネルギーの削減を目的とするのではなく, 実行時間の最小化を目的とすることで 2 次的に消費エネルギーを削減している. そのため, 1) は消費エネルギーの最小化が不十分である. 本節では, スケジューリング/FU バインディング時に演算処理の動的エネルギーを最小化することで, 回路全体の消費エネルギーの最小化を目指したスケジューリング/FU バインディングを提案する.

提案するスケジューリング/FU バインディングでは, 各ハドルへ電圧を割り当てることで, ハドルに含まれる演算器に電圧を割り当てる. 電圧を割り当てられた演算器に演算をバインディングすることで, 演算に電圧を割り当てる.

スケジューリング/FU バインディングは初期フェーズ, 電圧上昇フェーズ, 電圧下降フェーズの 3 つのフェーズで構成される. 初期フェーズは前回の反復時の配置, 電圧を変更せずにスケジューリング/FU バインディングを行う. 電圧上昇フェーズは初期フェーズの結果がステップ制約を満たさない場合実行され, ステップ制約を満たすようハドルの電圧を上げる. 電圧下降フェーズはステップ制約を満たす範囲で消費エネルギーが最小となるようハドルの電圧を下げる. 各フェーズを通し, 動的エネルギーが最小となる結果を採用する.

電圧上昇フェーズ, 電圧下降フェーズにおいて電圧を変更するハドルは優先度により選択する. 優先度はハドルに所属する演算器が 1 回の処理に消費するエネルギーの合計とし, ハドル h_j の優先度 $P_s(h_j)$ を以下の式で表す.

$$P_s(h_j) = \sum_{f_i \in F(h_j)} E(f_i) \quad (5)$$

初期フェーズ

初期フェーズはスケジューリング/FU バインディングで最初に行われ, 前回の反復時のハドルの配置, 電圧を変更せずにスケジューリング/FU バインディングを行う. 初期フェーズ, 電圧上昇フェーズ, 電圧下降フェーズ全てのフェーズで共通する処理を図 3 に示す. 初期フェーズでは共通処理のみを実行し, 初期フェーズの結果がステップ制約を満たさない場

共通処理

- (1) データ転送の際に必要なクロック数を表す行列 DT を計算.
- (2) DT ベースのスケジューリング/FU バインディング⁵⁾.
- (3) ステップ制約を満たす場合、以下の操作で動的エネルギーを最小化.
 - (a) 電圧 v_l, v_m の乗算器がある場合以下を実行.
 - (i) v_m で実行する乗算を 1 つ選択.
 - (ii) 選択した乗算を v_l で実行することとし、他の演算の電圧は変更せず、 DT ベースのスケジューリング/FU バインディング⁵⁾ を実行.
 - (iii) ステップ制約を満たす場合採用し、ステップ制約を満たさない場合電圧を戻す.
 - (iv) v_m で実行する乗算をすべて試行したら (a) を終了し、そうでない場合 i へ戻る.
 - (b) 電圧 v_l, v_h の乗算器がある場合、 v_h の乗算に対し (a) と同じ処理を実行.
 - (c) 電圧 v_m, v_h の乗算器がある場合、 v_h の乗算に対し (a) と同じ処理を実行.
 - (d) 電圧 v_l, v_m の加算器がある場合、 v_m の加算に対し (a) と同じ処理を実行.
 - (e) 電圧 v_l, v_h の加算器がある場合、 v_h の加算に対し (a) と同じ処理を実行.
 - (f) 電圧 v_m, v_h の加算器がある場合、 v_h の加算に対し (a) と同じ処理を実行.

図 3 共通処理.

Fig. 3 Common processing algorithm.

電圧上昇フェーズ

- (1) v_l のハドルのうち、最小の優先度 P_s を持つハドルを選択、電圧を v_m へ変更
- (2) 共通処理
- (3) ステップ制約を満たす場合、終了
ステップ制約を満たさない場合、 v_l のハドルがあれば 1 へ、なければ 4 へ
- (4) v_m のハドルのうち、最小の優先度 P_s を持つハドルを選択、電圧を v_h へ変更
- (5) 共通処理
- (6) ステップ制約を満たす場合、もしくは v_m のハドルがない場合、終了
ステップ制約を満たさない場合、 v_m のハドルがあれば 4 へ

図 4 電圧上昇フェーズの処理.

Fig. 4 Voltage-increasing phase algorithm.

合電圧上昇フェーズへ移行し、ステップ制約を満たす場合は電圧下降フェーズへ移行する.

電圧上昇フェーズ

電圧上昇フェーズは初期フェーズでステップ制約を満たさない場合に実行される. 電圧上昇フェーズの目的は初期フェーズ結果がステップ制約を満たさない場合、電圧を上昇することでステップ制約を満たすハドルの電圧を発見することである. 電圧上昇フェーズの処理を図 4 に示す.

電圧下降フェーズ

- (1) v_h のハドルのうち、最大の優先度 P_s を持つハドルを選択 (ただし、一度選んだハドルは選ばない)、電圧を v_m へ変更
- (2) 共通処理
- (3) ステップ制約を満たさない場合、電圧を v_h へ戻す
- (4) 選択したことのない v_h のハドルがある場合、1 へ
- (5) v_m のハドルのうち、最大の優先度 P_s を持つハドルを選択 (ただし、一度選んだハドルは選ばない)、電圧を v_l へ変更
- (6) 共通処理
- (7) ステップ制約を満たさない場合、電圧を v_m へ戻す
- (8) 選択したことのない v_m のハドルがある場合、5 へ
全ての v_m のハドルを選択した場合、終了

図 5 電圧下降フェーズの処理.

Fig. 5 Voltage-decreasing phase algorithm.

表 1 演算器の情報.

Table 1 FU component information.

演算器 (電圧)	面積 [μm^2]	遅延 [ns]	エネルギー [fJ]	リーク電力 [μW]	$V_{in} - V_{out}$	面積 [μm^2]	遅延 [ns]	エネルギー [fJ]	リーク電力 [μW]
加算器 (1.2 V)	386	0.75	92	3.9	1.2 V - 1.0 V	113	0.17	83	49.1
加算器 (1.0 V)	386	1.22	64	3.2	1.2 V - 0.8 V	113	0.22	71	32.3
加算器 (0.8 V)	386	2.71	41	2.6	1.0 V - 1.2 V	113	0.17	76	45.0
乗算器 (1.2 V)	2161	1.65	1135	19.8	1.0 V - 0.8 V	113	0.30	55	18.3
乗算器 (1.0 V)	2161	2.7	788	16.5	0.8 V - 1.2 V	113	0.22	86	39.1
乗算器 (0.8 V)	2161	6.0	504	13.2	0.8 V - 1.0 V	113	0.30	55	18.3

表 2 レベルコンバータの情報.

Table 2 Level converter information.

電圧下降フェーズ

電圧下降フェーズはステップ制約を満たす中で、動的エネルギーを最小とする電圧を探すフェーズである. 電圧下降フェーズの処理を図 5 に示す.

5. 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. 計算機実験環境は、CPU が AMD Quad-Core Opteron 2360 SE 2.5 GHz \times 2、メモリ容量が 16 GB である. 対象アプリケーションとして DCT (ノード数 48)、EWF3 (ノード数 102)、7 次 FIR フィルタ (ノード数 75) を用いた. 実験で用いた演算器情報を表 1 に示し、レベルコンバータの情報⁸⁾を表 2 に示す. 各演算器は 16 bit 幅と仮定し、クロック周期を 2.5 ns とする. 電圧は $v_l = 0.8\text{V}, v_m = 1.0\text{V}, v_h = 1.2\text{V}$ とする. コントローラの面積は Synopsys 社の Design Compiler により実際に論理合成して求めた. 配線遅延は配線長の 2 乗に比例すると仮

表 3 計算機実験結果.
Table 3 Experimental result.

App.	FUs	CS	Architecture	Area [μm^2]	Rectangular Area [μm^2]	Dynamic Energy [pJ]	Leak Energy [pJ]	All Energy [pJ]
ewf3	+3*2	50	GDR	47792	55250	414.34	57.64	471.98
			RDR	69530	69530	362.35	85.06	447.41
			HDR	52506	62252	331.95	77.96	409.91
			HDR +10)	53405	66196	267.02	99.47	366.49
			1)	43074	47674	231.62	89.15	320.77
MHDR	47002	49400	241.11	87.53	328.64			
fir	+3*3	35	GDR	36840	48278	223.34	38.35	261.69
			RDR	81920	81920	191.34	105.18	296.52
			HDR	30082	35910	191.65	31.61	223.26
			HDR +10)	30082	35910	191.65	31.61	223.26
			1)	35028	40400	138.60	60.08	198.69
MHDR	52051	61500	122.22	60.04	182.26			
fir	+4*4	30	GDR	34540	36000	211.94	44.00	255.94
			RDR	82816	82816	188.83	123.57	312.40
			HDR	32047	35624	207.59	33.05	240.64
			HDR +10)	32047	35624	207.59	33.05	240.64
			1)	25836	29400	145.32	36.79	182.11
MHDR	44554	48174	128.59	43.60	172.19			
dct	+3*3	15	GDR	48551	58843	181.96	13.56	195.52
			RDR	59544	59544	144.15	15.99	160.13
			HDR	47321	63096	127.58	17.42	145.00
			HDR +10)	48076	55770	112.25	25.10	137.35
			1)	44850	68392	94.50	33.11	127.61
MHDR	51269	55221	86.17	21.17	107.34			
dct	+4*4	10	GDR	53864	58378	152.01	11.00	163.01
			RDR	81476	81476	127.87	13.69	141.57
			HDR	58184	71520	128.71	13.69	145.22
			HDR +10)	58184	71520	128.71	13.69	145.22
			1)	49503	54758	97.21	27.02	124.23
MHDR	58272	68255	89.11	25.37	114.48			

定し、 $250 \mu\text{m}^2$ あたり 1 ns とする。

GDR⁶⁾, RDR²⁾, 1 電源の HDR, 1 電源の HDR として作成した後 10) を適用し複数電源電圧とした HDR, 複数電源電圧を考慮した HDR(MHDR : MultiV_{dd} HDR)¹⁾, 提案手法を適用した複数電源電圧の MHDR を比較する。

実験結果を表 3 に示す。MHDR は従来手法と比較し、平均 23.2 % , 最大 45.1 % 全消費エネルギーを削減できた。ダイナミック電力による消費エネルギーに関しては、MHDR は従来手法と比較し平均 29.3 % , 最大 52.6 % 削減されている。一方リーク電力による消費エネルギーに関しては、MHDR は従来手法より平均 26.8 % 消費エネルギーが増大している。複数電源電圧の手法全てでリーク電力による消費エネルギーが増大していることより、エネルギー増大の原因はレベルコンバータによるものと考えられる。複数電源電圧を適用した場合同士で比較しても、平均 13.5 % , 最大 28.4 % 全消費エネルギーを削減できた。1) と比較しても、平均 7.0 % , 最大 15.9 % 全消費エネルギーを削減できた。

6. おわりに

本稿では低電力化を考慮したレジスタ分散型アーキテクチャHuddle-based Distributed Register(以下 HDR) アーキテクチャをを対象とした複数電源電圧による低電力化高位合成手法を提案した。提案手法は配線遅延を考慮する HDR アーキテクチャを対象に低電力化することで、配線遅延と低電力化を同時に考慮する高位合成となっている。提案手法は従来手法と比較し、全消費エネルギーにおいて最大 45.1 % , 平均で 23.2 % のエネルギー削減を実現できた。

謝 辞

本研究は NEDO の助成による。

参 考 文 献

- 1) 阿部晋矢, 柳澤政生, 戸川望, “複数電源電圧および複数サイクルレジスタ間通信指向の低電力化高位合成手法,” 情報処理学会 DA シンポジウム 2011 論文集, 2011.
- 2) J.Cong, Y.Fan, G.Han, X.Yang, and Z.Zhang, “Architecture and synthesis for on-chip multi-cycle communication,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 550–564, Apr. 2004.
- 3) W.P.Lee and Y.W.Chang, “Voltage island aware floorplanning for power and timing optimization,” in *Proc. of ICCAD '06*, pp. 389–394, 2006.
- 4) A.Manzak and C.Chakrabarti, “A low-power scheduling scheme with resources operating at multiple voltages,” *IEEE Trans. on VLSI Systems*, vol. 10, no. 1, pp. 6–14, 2002.
- 5) A.Ohchi, S.Kohara, N.Togawa, M.Yanagisawa, and T.Ohtsuki, “Floorplan-driven high-level synthesis for distributed/shared-register architectures,” *IPJS Trans. on System LSI Design Methodology*, vol. 1, pp. 78–90, 2008.
- 6) A.Ohchi, N.Togawa, M.Yanagisawa, and T.Ohtsuki, “Floorplan-aware high-level synthesis for generalized distributed-register architectures,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 92, no. 12, pp. 3169–3179, 2009.
- 7) S.Tanaka, M.Yanagisawa, T.Ohtsuki, and N.Togawa, “A fault-secure high-level synthesis algorithm for RDR architectures,” *IPJS Trans. on System LSI Design Methodology*, vol.4, 2011.
- 8) C.Q.Tran, H.Kawaguchi, and T.Sakurai, “Low-power high-speed level shifter design for block-level dynamic voltage scaling environment,” in *Proc. of ICICDT 2005*, pp. 229–232, May 2005.
- 9) H.-C.Yang and L.-R.Dung, “On multiple-voltage high-level synthesis using algorithmic transformations,” in *Proc. of ASP-DAC '05*, pp. 872–876, Jan 2005.
- 10) B.Yu, S.Dong, S.Goto, and S.Chen, “Voltage-island driven floorplanning considering level-shifter positions,” in *Proc. of GLSVLSI '09*, pp. 51–56, 2009.