Vol.2013-SLDM-160 No.20
Vol.2013-EMB-28 No.20
2013/3/13

フロアプランを考慮したマルチクロックドメイン指向の低電力化高位合成手法

阿部 晋矢[†1,a)]　史 又華[†2]　柳澤 政生[†3]　戸川 望[†1,b)]

概要：本稿では，マルチクロックドメイン適用へ向け，HDR アーキテクチャを拡張した HDR-mcd を提案する．続いて HDR-mcd を対象にマルチクロックドメイン指向の低電力化高位合成を提案する．提案手法はフロアプラン情報をフィードバックし，反復改良する合成フローを取る．その際，1 クロック内の通信が保障されるハドルと呼ぶ区画を利用し，配線遅延の影響を予測，異なるクロック間の同期を考慮した高位合成を実現する．クロックはハドル毎に割り当て，資源制約と時間制約を満たす範囲で低い周波数のクロックを割り当てることで低電力化する．計算機実験により提案手法は従来の単一クロックのみを考慮したレジスタ分散型アーキテクチャと比較し 25%程度消費エネルギーを削減できることを確認した．

1. はじめに

近年の高集積，高機能化した LSI 設計では生産性向上のため,，より抽象度の高い設計が求められる．加えて，携帯機器に搭載される LSI の増加から，低消費電力でエネルギー効率の良い設計が求められる．動作記述からレジスタトランスファーレベル (RTL) の回路を自動合成する技術として高位合成がある．高位合成についても低消費電力技術への対応が求められる．

低消費電力技術としてマルチクロックドメインが提案された [5], [7], [8]．LSI 全体に対するクロック信号の消費エネルギーの割合は大きく，クロック周波数を削減するマルチクロックドメインはエネルギー削減に有効である．マルチクロックドメインを扱った高位合成として [7] があげられる．[7] はあらかじめ分割された各機能ブロックに対しより有効なクロック割り当てを実現した．しかし，高位合成中におけるクロックの周波数決定や，クロックドメイン分割を課題としている．加えて，LSI の微細化に伴い配線遅延が増加しており，各クロックドメイン間の通信は配線遅延の影響を受ける．我々の知る限り，クロックドメイン間の同期に対する配線遅延の影響を考慮した手法は存在しない．既存手法は，(1) 配線遅延の影響を考えた各クロック間での同期を考えていない．(2) 高位合成中の有効なクロック周波数選択とクロックドメイン分割が不可能である．更なる低電力化へ向け，既存手法の問題点を解決したマルチクロックドメインを考慮した高位合成が求められる．

配線遅延の影響を高位合成時に扱うプラットフォームとしてレジスタ分散型アーキテクチャが提案された [1], [2], [6], [9]．レジスタ分散型アーキテクチャは回路を配線遅延の影響がない範囲に分割，抽象化する．分割したブロックに対してフロアプランを実行する．フロアプランされたブロック間の配線遅延を予測することにより，配線遅延の影響を高位合成中に適切に扱う．低消費電力化を視野に入れたレジスタ分散型アーキテクチャとして，Huddle-based Distributed-Register アーキテクチャ (以下 HDR)[1] が提案された．HDR を対象に複数電源電圧 [1], [2]，クロックゲーティング [4]，パワーゲーティングを含んだ動的複数電源電圧 [3] が提案された．しかし，マルチクロックドメインを対象とした高位合成手法は提案されていない．

本稿ではまず，マルチクロックドメインを適用するため，HDR を拡張した HDR-mcd を提案する．HDR-mcd では 1 クロック内の通信が保障されるハドルと呼ぶ区画を利用する．ハドルをクロックドメインとみなし，クロックを割り当てる．各ハドルはマルチクロックドメインを考慮し同期する．次に，マルチクロックドメイン適用高位合成アルゴリズムを提案する．提案アルゴリズムは配置情報をフィードバックし，反復改良する合成フローを取る．配置情報か

†1 現在，早稲田大学大学院基幹理工学研究科情報理工学専攻
Presently with Dept. of Computer Science and Engineering, Waseda University
†2 現在，早稲田大学高等研究所
Presently with Waseda Institute for Advanced Study, Waseda University
†3 現在，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
Presently with Dept. of Electronic and Photonic Systems, Waseda University
a) shinya.abe@togawa.cs.waseda.ac.jp
b) togawa@togawa.cs.waseda.ac.jp

ら，レジスタ-レジスタ間通信を利用しクロックドメイン間の同期を考えたスケジューリングを実行する．その際，クリティカルパスでない演算を処理するハドルに低い周波数のクロックを割り当てる．

提案手法は，(1) ハドルとレジスタ-レジスタ間通信を考えた高位合成により，配線遅延を考慮したクロックドメイン間同期を実現する．また，(2) ハドルの配置情報をフィードバックする合成フローにより，クロックドメイン分割，クロック周波数選択を実現する．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25% 程度消費エネルギーを削減できることを確認した．

2. HDR-mcd

本章ではマルチクロックドメインと配線遅延を高位合成に統合するため，HDR[1] を拡張した HDR-mcd を提案する．HDR-mcd は HDR と同様にハドル (Huddle) という区画を導入し，各モジュールを抽象化する．割り当てるクロックドメインはハドル毎に決定する．ハドルはクロック周期制約により決定される範囲内において任意の矩形となる．その際のハドルの大きさの制約をハドルサイズ制約と呼ぶ．HDR では単一のクロックドメインを考えるため，ハドルサイズ制約は 1 通りである．一方，HDR-mcd は割り当てられたクロック周期に応じてハドルサイズ制約が変化する．

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．

HDR-mcd に対し，高位合成した結果を図 1 に示す．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 を利用することでデータ転送時間は無視できる．異なるハドル間の HFU 同士でデータ通信をする場合，HLR 間データ転送を行う．異なるクロックが割り当てられたハドル間の HLR 間データ転送では，配線遅延によるデータ転送時間に加えてレジスタの書き込みタイミングを考慮する．図 1(b) のエッジ e_1 と e_2 はそれぞれレジスタ-レジスタ間通信である．共にハドル A，B 間であるためデータ転送に必要な配線遅延は等しい．ハドル A，B 間の通信には 1 クロックの配線遅延が必要なため，e_1 はステップ 3 をかけてデータ転送が行われる．しかし，e_2 が送信されるハドル B はクロック周期が 2 ns であるため，ステップ 3 ではクロックが立ち上がらずデータ転送が終了しない．クロックが立ち上がるステップ 4 で書き込みが行われ，ステップ

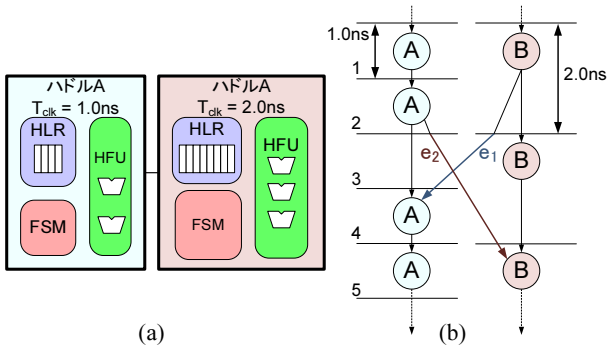

図 1 HDR-mcd を対象とした高位合成例．(a) HDR-mcd の構成．(b) DFG の例．

Fig. 1 A high-level synthesis example targeting HDR-mcd. (a) HDR-mcd configuration. (b) DFG.

5 から次の演算が実行される．ハドルによる抽象化で配線遅延，マルチクロックドメイン間の同期を考慮しハドルにクロックを割り当てることが可能である．

3. 問題の定式化

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．最小クロック周期制約として T_{clkmin}，ステップ制約 S_{max} を与える．S_{max} 個のコントロールステップ集合を $S = \{1, \cdots, S_{max}\}$ とし，あるコントロールステップを $S_t (1 \leq S_t \leq S_{max})$ で表す．

入力として p 個の演算器の集合 $F = \{f_1, \cdots, f_p\}$ を与え，演算器 f_i の遅延を $D_f(f_i)$ で表す．演算器 f_i で処理する演算数を $OP(f_i)$ で表す．

q 個 $(q \leq p)$ のハドル $H = \{h_1, \cdots, h_q\}$ 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．ハドル h_j のクロック因数を $CF(h_j)$ で表す．$T_{clk}(h_j)$ はハドル h_j に割り当てられたクロック周期を表し，最小クロック制約 T_{clkmin} より $T_{clk}(h_j) = T_{clkmin} \cdot CF(h_j)$ と計算される．ただし，$CF(h_j)$ は 2 のべき乗とし，$1 \leq CF(h_j) < S_{max}$ とする．$S_f(f_i)$ は $f_i \in F(hj)$ の最小クロック制約 T_{clkmin} を基にした必要ステップ数を表し，$S_f(f_i) = \lceil (D_f(f_i) + D_{reg}(h_j))/T_{clk}(h_j) \rceil \cdot CF(h_j)$ と計算される．

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．

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

$Slack(f_i)$ 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．

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

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．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．

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

$Slack(f_i)$ 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，最小クロック制約を基にした必要なステップ数 $DT(f_i, h_k)$ は以下の式で表される．

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

このとき，DT とは p 行 q 列のテーブルで，i 行 k 列の要素が $DT(f_i, h_k)$ となるものとしデータ転送テーブルとよぶ．コントロールステップ S_{start} からデータを転送し，S_{end} でデータ転送が終了する場合，ハドル h_k のクロック周期 $T_{clk}(h_k)$ を考え S_{end} は以下の式で求められる．

$$S_{end} = \left\lceil \frac{S_{start} + DT(f_i, h_k) - 1}{CF(h_k)} \right\rceil \cdot CF(h_k) \quad (5)$$

以上より，HDR-mcd を対象としたマルチクロックドメイン適用高位合成問題を次のように定義する．

定義 3.1 HDR-mcd を対象としたマルチクロックドメイン適用高位合成問題とは，CDFG，最小クロック周期制約，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，ハドルにクロック因数を割り当てることである．

4. マルチクロックドメイン適用低電力化高位合成手法

HDR-mcd はスケジューリング，バインディング結果でハドルの構成が決まり，ハドル間の通信の配線遅延が変化する．そのため，あらかじめスケジューリング，バインディング，フロアプランの回数を決定することは難しい．本稿では，HDR[1], [2], [4]，AVHDR[3] と同様に反復改良するアルゴリズムを採用する．

アルゴリズム考案に向けて問題となるのが，割り当てられたクロック周期により変化するハドルサイズ制約の扱いである．提案手法ではマルチクロックドメインに対応したスケジューリング/FU バインディングとハドル分割を提案する．HDR-mcd 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
- ハドル分割
- フロアプラン指向ハドル合成
- 仮想面積調整

以上のうち，初期ハドル合成，レジスタ/コントローラ合成については [2] と同様の処理を行う．フロアプラン，フロアプラン指向ハドル合成，仮想面積調整については [2] と同じアルゴリズムで処理するが，マルチクロックドメイン適用に伴いパラメータを変更した．スケジューリング/FU バインディングはフロアプラン結果から配線遅延を見積もり，マルチクロックドメイン，レジスタ-レジスタ間通信を考慮したスケジューリングおよび FU バインディングを行う．その際，クリティカルパスでない演算を処理するハドルにより低い周波数のクロックを割り当てる．ハドル分割では，ハドルサイズ制約を満たさないハドルを分割する．

提案アルゴリズムは仮想面積ベースの反復改良 [2] を採用する．仮想面積ベースの反復改良では，フロアプランの際に実面積と仮想面積という 2 つの面積を用意する．ハドル h_j の実面積を $A_{real}(h_j)$ とし，$A_{real}(h_j)$ はハドルに含まれる演算器やレジスタの面積を合計を表す．一方ハドル h_j の仮想面積を $A_{virtual}(h_j)$ とし，$A_{virtual}(h_j)$ は以下の通り算出する．

(1) 変更のあったハドル h_j の演算器，レジスタ，コントローラ，レベルコンバータの実面積 $A_{real}(h_j)$ を求める．
(2) $A_{virtual}(h_j) \geq A_{real}(h_j)$ の場合，ハドルの面積は変更しない．
(3) $A_{virtual}(h_j) < A_{real}(h_j)$ の場合，$A_{virtual}(h_j) = A_{real}(h_j)$ とし，元のアスペクト比を基に $H(h_j)$ と $W(h_j)$ を更新する．

提案手法は初期処理，反復処理，調整処理の 3 つの処理に分割される．初期処理，調整処理では実面積を基にフロアプランする．反復処理では仮想面積を基にフロアプランする．反復処理で解が収束した後は調整処理へ移り，実面積を基にフロアプランすることで仮想面積のオーバヘッドを削減する．最終的な高位合成アルゴリズムを図 2 に示す．

Vol.2013-SLDM-160 No.20
Vol.2013-EMB-28 No.20
2013/3/13

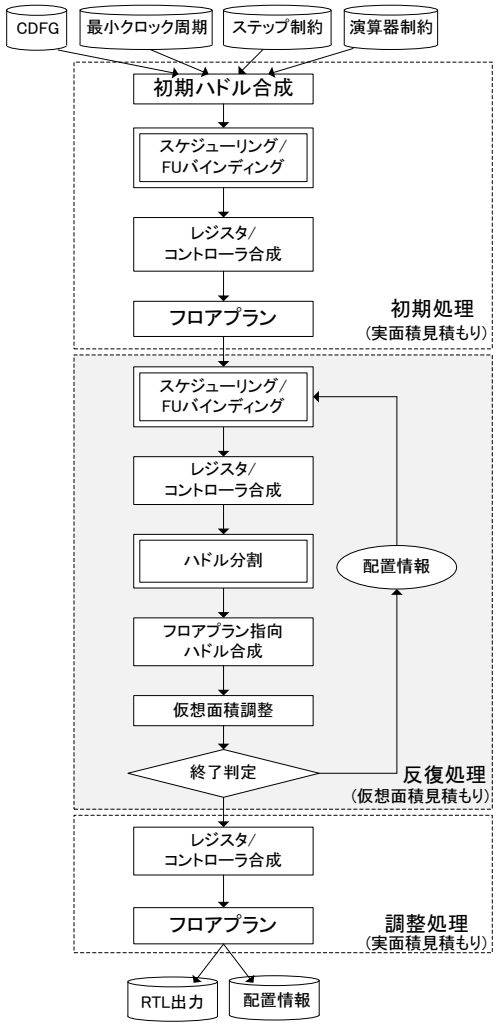

図 2 HDR-mcd を対象とした高位合成アルゴリズム.

Fig. 2 Proposed high-level synthesis algorithm targeting HDR-mcd.

4.1 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，最小クロック周期制約 T_{clk}，ステップ制約 S_{max}，CDFG $G(N, E)$，演算器数，ハドルの構成，配置情報である．出力は演算ノード $v \in N_o$ を実行するコントロールステップ，v を実行する演算器，各ハドルのクロック因数である．

スケジューリング/FU バインディングは (i) 初期フェーズ，(ii) クロック因数上昇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を変更せず，全てのハドルのクロック因数を 1 としてスケジューリング/FU バインディングを行う．第 1 回目のループはフロアプランの実行前のため，全てのハドルの間に配線遅延がないものとする．また，第 1 回目のループは初期処理のみを実行し，クロック因数上昇フェーズは行わない．初期フェーズの処理を Algorithm 1 に示す.

クロック因数上昇フェーズは S_{max} 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ごとクロック因数を増やす．クロック因数を変更するハドルは優先度により選択する．初期フェーズで各ハドルにバインディングされた演算数とする．ハドル h_j の優先度 $P_s(h_j)$ を以下の式で表す.

$$P_s(h_j) = \sum_{f_i \in F(h_j)} OP(f_i) \tag{6}$$

クロック因数上昇フェーズの処理を Algorithm 2 に示す.

Algorithm 1 (i) 初期フェーズ

```
1: 全てのハドル h_j に対し，CF(h_j) ← 1.
2: DT を作成
3: DT ベースのスケジューリング/FU バインディング [9].
```

Algorithm 2 (ii) クロック因数上昇フェーズ

```
1: P_s(h_j) を全てのハドルについて求める.
2: while do
3:    FLAG ← false
4:    for P_s(h_j) に対し降順な h_j do
5:       CF(h_j) ← 2 · CF(h_j)
6:       if CF(h_j) < S_max then
7:          クロックドメイン間のデータ転送終了時間を考え，DT
            ベースのスケジューリング/FU バインディング [9].
8:          if スケジューリング結果が S_max を超える then
9:             CF(h_j) ← CF(h_j)/2
10:         else
11:            FLAG ← true
12:         end if
13:      else
14:         CF(h_j) ← CF(h_j)/2
15:      end if
16:   end for
17:   if FLAG = false then
18:      クロック因数上昇フェーズ終了
19:   end if
20: end while
```

4.2 フロアプラン

フロアプランは HDR[2] と同じアルゴリズムを適用する．データ構造として Sequence-pair を用い，SA によりフロアプランすることとし，以下の 3 つの move を考える.

move 1 2 要素を選択し，Γ_+ で入れ替える.

move 2 2 要素を選択し，Γ_+, Γ_- で入れ替える.

move 3 1 要素を選択し，アスペクト比を変更する.

上記の 3 種類の move に加えて，SA のコスト関数 *cost* を以下に示す.

$$cost = \frac{A_{BB}}{A_{total}} + \alpha\frac{V}{T_{clock}} + \beta\frac{W}{W_{MAX}} + \gamma\frac{A_{CLK}}{A_{total}} \tag{7}$$

ただし，α，β，γ は，任意のパラメータである．A_{BB} は最小矩形の面積，A_{total} はモジュール面積の総計，T_{clock} はクロック周期，V はクロック周期制約違反の総合計，W は配線長，W_{MAX} は (最小矩形の縦+横)× 配線数で計算される配線長の最大値，A_{CLK} は各クロック周期における最小矩形の面積の総和である.

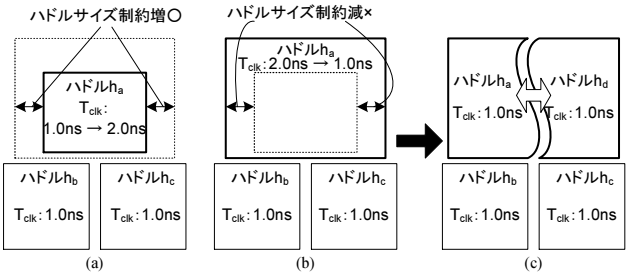

図 3 反復によるハドルサイズ制約の変化．(a) ハドルサイズ制約が増加する場合．(b) ハドルサイズ制約が減少する場合．(c) ハドルサイズ制約違反を解決するための分割．

Fig. 3 Huddle size constraint change. (a) Huddle size constraint of h_a become more permissive. (b) Huddle size constraint of h_a become more strict. (c) h_a is divided.

i 回目のイタレーションにおける最終的な配置結果を，$i+1$ 回目のイタレーションのフロアプランにおける SA の初期配置として用いる．イタレーションごとに SA の初期温度を下げていく．合成フローの i 回目のイタレーションの SA の初期温度を T_i としたとき $i+1$ 回目のイタレーションの初期温度を

$$T_{i+1} = T_i / K \tag{8}$$

とする．ただし $K > 1$ とする．

4.3 ハドル分割

HDR-mcd は割り当てられたクロック周期により，式 (2) で表すハドルサイズ制約が変化する．スケジューリング/FU バインディングで，図 3(a) のように前回の反復時よりも大きいクロック因数が割り当てられた場合，ハドルサイズ制約の違反は起こらない．一方，図 3(b) のように前回の反復時よりも小さなクロック因数が割り当てられた場合，ハドルサイズ制約を違反する場合がある．その場合，図 3(c) の通り，ハドルを分割する必要がある．

反復処理では仮想面積を用いるため，フロアプラン上にはいずれの演算器も所属していないハドルが存在する．いずれの演算器も所属していないハドルを空のハドルと呼ぶ．ハドル分割では空のハドルを利用し，図 3(c) の分割を実現する．ハドル分割の処理をアルゴリズム 3 に示す．

4.4 フロアプラン指向ハドル合成

フロアプラン指向ハドル合成は HDR[2] と同じアルゴリズムを適用する．4.2 節のフロアプランの 3 種類の move に加えて，ハドルの構成を変化させる操作として，以下の 4 つ目の move を追加する．

move 4 演算器 f_i を選択し，ハドル $h_j(= Hud(f_i))$ からハドル $h_k(\neq h_j)$ へ移動する．

SA のコスト関数 *cost* は式 (7) を用いる．SA の初期温度は式 (8) を用いる．

Algorithm 3 ハドル分割

```
for 全てのハドル h_j do
  if h_j がハドルサイズ制約を超える then
    for f_i ∈ F(h_j) do
      if 他に h_j に所属する演算器がある then
        ハドル h_vacant を用意.
        for 全てのハドル h_k do
          if h_k が空のハドル then
            h_vacant ← h_k，ステップ 11 へ.
          end if
        end for
        f_i を h_vacant へ移動.
        A_virtual(h_vacant) ← A_real(h_vacant).
      else
        A_virtual(h_j) ← A_real(h_j).
      end if
    end for
  end if
end for
```

表 1 演算器の情報．

Table 1 Component Information.

	Area [μm^2]	Delay [ns]	Dynamic energy [fJ]	Leak power [μW]
加算器	386	1.22	64.0	3.20
乗算器	2161	2.70	788	16.5
レジスタ	330	0.47	188	1.47
マルチプレクサ (1bit)	36	0.21	6.26	0.558

4.5 仮想面積調整

仮想面積調整は HDR[2] と同じアルゴリズムを適用する．仮想面積調整は以下の通り実行する．

(1) ハドル h_j の実面積 $A_{real}(h_j)$ と仮想面積 $A_{virtual}(h_j)$ の差，$A_{dif}(h_j) = A_{virtual}(h_j) - A_{real}(h_j)$ を求める．
(2) $A_{virtual}(h_j) = A_{real}(h_j) + \phi \cdot A_{dist}(h_j)$ と仮想面積を更新する．

ϕ は調整パラメータである．反復に応じて ϕ の値を削減するため，各反復 i において $\phi = 1 - 0.05i$ として実験した．

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 × 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 に示す．各演算器は 16 bit 幅と仮定し，最小クロック周期を 2.5 ns 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 乗に比例すると仮定し，250 μm あたり 1 ns とする [1]．クロックツリーの

表 2　計算機実験結果.
Table 2　Experimental results.

App. (S_{max})	FUs	Architecture and algorithm	Steps	Rectungular area [μm^2]	Dynamic [pJ]	Leak [pJ]	Wire [pJ]	Clock Tree [pJ]	All [pJ]	CPU time [ns]
DCT	+4	SR	13	60270	64.17	14.66	47.19	57.63	183.64	25.86
(12)	×4	RDR	12	129600	98.43	19.43	34.68	149.34	301.96	198.48
		HDR	12	59527	98.26	16.75	26.41	108.26	249.68	879.12
		HDR-mcd	12	86241	76.34	16.79	31.91	76.82	201.85	668.44
EWF3	+4	SR	80	69402	223.68	104.07	114.78	272.23	714.77	28.48
(54)	×4	RDR	59	176400	280.60	102.17	78.56	600.64	1061.97	206.96
		HDR	53	53985	260.73	67.25	59.31	364.58	751.87	746.38
		HDR-mcd	54	52716	233.71	73.66	54.83	293.39	655.59	615.43
FIR	+4	SR	31	58206	106.03	47.49	61.76	98.99	314.28	26.61
(31)	×4	RDR	29	86400	128.63	28.30	29.84	203.42	390.18	125.55
		HDR	31	31395	118.01	25.72	19.15	135.97	298.85	566.95
		HDR-mcd	30	33990	75.91	28.46	34.74	62.67	201.78	517.02

消費エネルギーについては [10] の式を用いた.

配線遅延を考えない共有レジスタによる高位合成 (表 2 の SR)[9], RDR[6], HDR[2] とマルチクロックドメイン拡張した HDR-mcd を比較する．実験結果を表 2 に示す．HDR-mcd は全消費エネルギーを最大 48.3%，平均 24.3% 削減した．クロックツリーのエネルギーに注目すると HDR-mcd は最大 69.2%，平均 29.7% 削減した．クロックツリーはクロック周波数ごとに用意するためエネルギーが増加する可能性がある．しかし，提案手法はクロックツリー数の増加のオーバヘッド以上にエネルギーを削減可能である．加えて，HDR-mcd は動的エネルギーを最大 41.0%，平均 17.0% 削減した．動的エネルギーの大部分はクロックによるレジスタの消費エネルギーである．HDR-mcd は静的エネルギーが平均 8.4% 削減したが，増加している場合もある．しかし，静的エネルギーのオーバヘッド以上にエネルギーが削減されるため，提案手法は有効である．

6. おわりに

本稿では，マルチクロックドメイン適用へ向け，HDR アーキテクチャを拡張した HDR-mcd を提案した．続いて HDR-mcd を対象にマルチクロックドメイン指向の低電力化高位合成を提案した．計算機実験により提案手法は従来の単一クロックのみを考慮したレジスタ分散型アーキテクチャと比較し 25%程度消費エネルギーを削減できることを確認した.

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] Abe, S., Yanagisawa, M. and Togawa, N.: Energy-efficient High-level Synthesis for HDR Architectures, *IPSJ Trans. on System LSI Design Methodology*, Vol. 5, pp. 106–117 (2012).

[2] Abe, S., Shi, Y., Yanagisawa, M. and Togawa, N.: MH^4 : 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).

[3] 阿部晋矢，史又華，宇佐美公良，柳澤政生，　戸川望：SAAV:AVHDR アーキテクチャを対象とした動的複数電源電圧指向の低電力化高位合成手法，信学技法，VLD2012–82，Vol. 112, No. 320, pp. 135–140 (2012).

[4] Akasaka, H., Yanagisawa, M. and Togawa, N.: Energy-efficient high-level synthesis for HDR architectures with clock gating, *Proc. of ISOCC '12*, pp. 135–138 (2012).

[5] Bomel, P., Martin, E. and Boutillon, E.: Synchronization processor synthesis for latency insensitive systems, *Proc. of DATE '05*, pp. 896–897 (2005).

[6] Cong, J., Fan, Y., Han, G., Yang, X. and Zhang, Z.: Architecture and synthesis for on-chip multicycle communication, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 23, No. 4, pp. 550–564 (2004).

[7] Lhairech-Lebreton, G., Coussy, P. and Martin, E.: Hierarchical and multiple-clock domain high-level synthesis for low-power design on fpga, *Proc. of FPL '10*, pp. 464–468 (2010).

[8] Nose, K., Shibayama, A., Kodama, H., Mizuno, M., Edahiro, M. and Nishi, N.: Deterministic inter-core synchronization with periodically all-in-phase clocking for low-power multi-core SoCs, *ISSCC Dig. Tech. Papers*, pp. 296–599 (2005).

[9] Ohchi, A., Kohara, S., Togawa, N., Yanagisawa, M. and Ohtsuki, T.: Floorplan-driven high-level synthesis for distributed/shared-register architectures, *IPSJ Trans. on System LSI Design Methodology*, Vol. 1, pp. 78–90 (2008).

[10] Vittal, A. and Marek-Sadowska, M.: Low-power buffered clock tree design, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 9, pp. 965–975 (1997).