

## 動的再構成デバイス向け配線共有型マルチコンテキスト化手法の一検討

篠原 拓<sup>†</sup> 木幡 雅貴<sup>††</sup> 今井 茂毅<sup>††</sup> 飯田 全広<sup>†††</sup> 末吉 敏則<sup>†</sup>

<sup>†</sup> 熊本大学 工学部 数理情報システム工学科 〒860-8555 熊本市黒髪 2-39-1

<sup>††</sup> 熊本大学大学院自然科学研究科数理科学・情報システム専攻

<sup>†††</sup> 科学技術振興機構さきがけ 〒332-0012 埼玉県川口市本町 4-1-8

E-mail: †{shinohara,kobata,imai}@arch.cs.kumamoto-u.ac.jp, ††{iida,sueyoshi}@cs.kumamoto-u.ac.jp

あらまし 動的再構成デバイスを用いたマルチコンテキスト手法はハードウェアの仮想化、再構成速度の高速化などの利点があり、我々はこれを利用した低消費エネルギー化手法を提案している。しかし、従来のマルチコンテキスト手法は定められた領域でコンテキストを切り替えている。そのため、領域内のすべての構成データを変更する必要があり、切替対象ではない部分との混在は困難である。本研究ではコンテキストの配線を共有させ、論理ブロックのみの再構成によりコンテキストの切替えを実現する手法を提案する。さらに本稿では、提案手法の特徴を利用して、コンテキストの切替えに再構成を不要とするアーキテクチャモデルを提案し、その評価を行う。結果として、消費エネルギーを平均で約 19.7%削減し、構成データを平均で 45.6%削減できた。

キーワード マルチコンテキストデバイス、動的再構成機能、低消費電力化、配線共有

## A Study on Resource Sharing Technique for Multi-Context Logic Device.

Hiroshi SHINOHARA<sup>†</sup>, Masaki KOBATA<sup>††</sup>, Shigeki IMAI<sup>††</sup>,

Masahiro IIDA<sup>†††</sup>, and Toshinori SUEYOSHI<sup>†</sup>

<sup>†</sup> The faculty of engineering, Kumamoto University 2-39-1 Kurokami, kumamoto-shi, 860-8555 Japan

<sup>††</sup> Department of Mathematics and Computer Science,

Graduate School of Science and Technology, Kumamoto University

<sup>†††</sup> PRESTO, Japan Science and Technology Agency 4-1-8 Honcho, Kawaguchi-shi, Saitama, 332-0012 Japan

E-mail: †{shinohara,kobata,imai}@arch.cs.kumamoto-u.ac.jp, ††{iida,sueyoshi}@cs.kumamoto-u.ac.jp

**Abstract** Multi-context reconfigurable devices are convenient solution for runtime reconfiguration. However they suffer from reconfiguration unit area and reconfiguration overhead. A new technique is presented in this paper to reduce power consumption for multi-context reconfigurable devices. The main goal is to reduce power consumption by sharing datapath in each context. As a result of evaluation, the energy consumption was reduced by an average of about 19.7%.

**Key words** Multi-Context Device, Dynamic Reconfiguration, Low Power, Routing Sharing

### 1. はじめに

FPGA (Field Programmable Gate Array) は LUT (Look Up Table) に基づく基本論理ブロックを二次元アレイ上に配したデバイスで、回路を交換えられるという特性を持つ。この特性を積極的に利用した、RC (Reconfigurable Computing) と呼ばれる計算機アーキテクチャの研究が活発に行われている [1]。しかしながら、従来の FPGA では実行中の再構成ができないため回路を停止して再構成する必要があり、極端に性能

が落ちてしまう。これまで、再構成時間を隠蔽する試みとしてマルチコンテキストデバイスの研究・開発が盛んに行われてきた [2], [3]。これは、回路情報をデバイス内部のメモリに複数保持させ、必要な時に瞬時に回路構成を切替えることができるデバイスである。

我々は、これまでの研究においてマルチコンテキストデバイス向けに回路の低消費エネルギー化手法を提案し、その有効性を検証してきた [4]。しかし、その手法は再構成エネルギーが大きくなると低消費エネルギー化が実現できないことが確認され

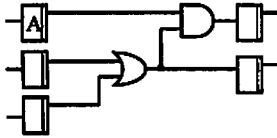


図1 オリジナル回路

ている。そこで本研究では、配線共有型マルチコンテキスト化手法を提案する。提案手法では、複数のコンテキスト間で共通の配線構造を用いて論理ブロックのみを切替えることにより、コンテキストの切替えを実現する。そのため、コンテキストの切替えの際に、配線部分の再構成エネルギーが不要となる。本稿では、提案している低消費エネルギー化手法を配線共有型マルチコンテキスト化手法で実現し、消費エネルギーと構成データ量の評価を行う。

まず、2章では我々が提案している低消費エネルギー化手法について説明する。次に、3章では今回提案するマルチコンテキスト手法について説明する。4章では本稿で評価の対象とするアーキテクチャモデルについて説明し、5章で評価手法について述べる。そして、6章で結果を述べるとともに考察を行い、最後に7章でまとめと今後の課題について述べる。

## 2. マルチコンテキストデバイス向け低消費エネルギー化手法

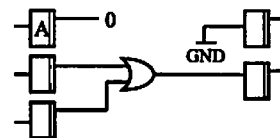
本章では、提案しているマルチコンテキストデバイス向け低消費エネルギー化手法について説明する。

### 2.1 低消費エネルギー化手法

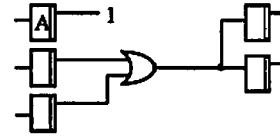
一般に、システム LSI に実装される回路は、ハードウェア化が必要なほどの演算能力と時間的制約を持つ。しかし、回路は動作中にすべての時間で処理データが供給され、演算が行われるわけではない。また、たとえデータが供給されているとしても、入力データによって処理内容が変化することが多い。つまり、動作中のある瞬間には、そのときの入力データや処理内容に依存した最適な回路が存在する。最適な回路はオリジナル回路と比較して、回路を構成する素子数が最適化されていることから、消費エネルギーの削減が期待できる。我々が提案している低消費エネルギー化手法は、まず、オリジナル回路から複数の最適な回路を設計する。そして、デバイスの持つ動的再構成機能により、実行中に最適な回路を割り当てて構成し、オリジナル回路と同じように動作させる。我々は、FIR フィルタなどで入力周波数別に最適な回路を設計し、オリジナル回路と消費エネルギーを比較することで、本手法の有効性を確認している [5]。

### 2.2 マルチコンテキスト化手法

任意のアプリケーションで最適な回路を設計する場合、設計にかかる期間の増加や誤りの発生が懸念される。また、回路の切替えタイミングの制御が問題となる。そこで、入力データに依存した回路、すなわち、論理ゲートの入力を固定させ、ゲートの削減を行ったデータ依存回路を設計し、マルチコンテキスト化する方法を提案している [4], [6]。この手法は必ずしも回路



(a) Data-dependent circuit set to 0



(b) Data-dependent circuit set to 1

図2 データ依存回路

の動作や特徴に則した最適な回路が生成されるとは限らないが、回路構造を意識することなく、短時間で簡単に設計を行うことができる。また、回路の切替えは、入力データに応じて自律的に行うことができ、本手法は再構成にかかるコストが低い。

図1にオリジナル回路の例を、図2にデータ依存回路の例を示す。データ依存回路を生成するために、回路中のフリップフロップの出力を固定する。オリジナル回路のAのフリップフロップを0に固定した回路が図2(a)、1に固定した回路が図2(b)である。それぞれ、ANDゲートが一つ削減された回路になっている。データ依存回路の切替えタイミングの制御は、固定したフリップフロップの出力信号により行う。また、再構成は組合せ回路部分のみに適用することにより、各コンテキスト間での演算結果は、共通のフリップフロップにより受け渡す。

このようなデータ依存回路をコンテキストとして用意し、デバイスの動的再構成機能により再構成させ、オリジナル回路と同様に動作させる。この時、データ依存回路は論理ゲートが削減されているため、オリジナル回路に対して消費エネルギーの削減が期待される。

## 3. 配線共有型マルチコンテキスト化手法

再構成可能デバイスは、再構成を行うときにエネルギーを必要とする。マルチコンテキストデバイスも例外ではない。したがって、提案している低消費エネルギー化手法を適用しても、回路の切替えの頻度が多くなると再構成エネルギーの総和が大きくなり、低消費エネルギー化が実現しない可能性がある。そこで我々は、再構成の前後で共通の配線構造を用い、配線部分の再構成エネルギーを必要としない配線共有型マルチコンテキスト化手法を提案する。

### 3.1 配線共有型マルチコンテキスト化手法の概要

配線共有型マルチコンテキスト化手法とは、複数のコンテキスト間で配線を共有し、論理ブロックのみの再構成でコンテキストの切替えを実現する手法である。図3に、配線共有型マルチコンテキスト化手法の例を示す。2つのコンテキストAとBから、統合コンテキストを設計する。Aの処理とBの処理は、各々統合コンテキストの配線構造上で実現可能なため、論理ブロックのみの再構成により、それぞれのコンテキストを排他的

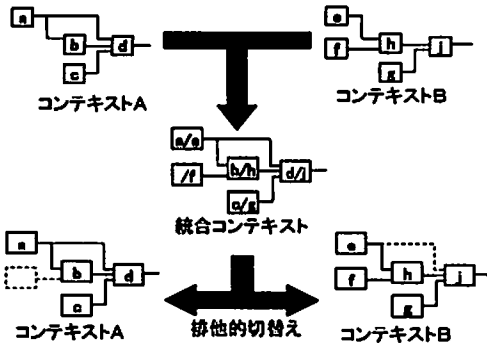


図3 配線共有型マルチコンテキスト化手法

に切替えて実行できる。それぞれのコンテキストを実行する際には、点線となっている論理ブロックや配線は使用されない。配線共有型マルチコンテキスト化手法は、コンテキスト切替の際に配線構造が変化しないことから、マルチコンテキスト部分の論理ブロックと、非マルチコンテキスト部分の論理ブロックの混在が容易となる利点がある。さらに、コンテキスト全体を再構成させる従来のマルチコンテキスト手法と比較して、構成データや再構成エネルギーの削減が望める。従来のマルチコンテキスト化手法とは、データ依存回路の切替え時にコンテキストを配線、論理ブロックの全てを再構成する手法である。

### 3.2 統合コンテキストの設計

本手法では、複数のコンテキスト間で、IPad (入力端子) や OPad (出力端子)、LUT などのリソースを互いにマッピングすることで統合コンテキストを設計する。マッピングされるリソースは、統合コンテキスト上で共有される。一方で、マッピングされないリソースは統合コンテキスト上で共有されず、あるコンテキストを実行している際には冗長となり、面積・消費電力・構成データ量の効率を悪化させる。よって、統合コンテキストの設計目標は、各コンテキスト間でリソースをより多くマッピングさせ、必要最小限のリソース量で統合コンテキストを設計することである。

本節では、統合コンテキストを設計する手法を述べる。まず、最小のリソース量で統合コンテキストを生成する手法を説明する。続いて、その手法の問題点を示し、さらに問題の改善方法を述べる。統合コンテキストの設計例を図4に示し、順に説明する。

#### 3.2.1 Universal Graph

設計手法では、コンテキストを有向グラフで表現する。有向グラフの節点は I/OPad や LUT を、弧は配線をそれぞれ表す。グラフ理論において、複数のコンテキストから統合コンテキストを設計することは、各コンテキストから得られる有向グラフの、Universal Graph を求めることに等しい。Universal Graph とは、複数のグラフを全て部分グラフとして内包するグラフである。以下、2つの有向グラフ  $G_A$ ,  $G_B$  の Universal Graph  $G_{univ}$  を最小の節点数、並びに最小の弧の数で設計する手法を述べる。

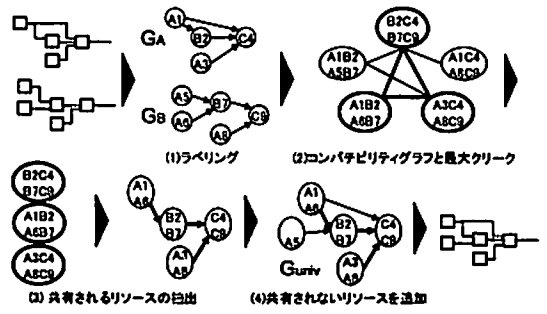


図4 統合コンテキストの設計

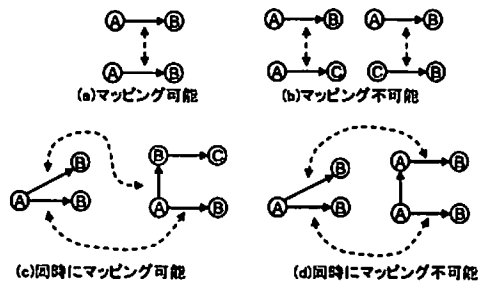


図5 マッピングの制約

#### 3.2.2 ラベリング

$G_{univ}$  の設計を行うために、 $G_A$ ,  $G_B$  の弧をマッピングする。しかし、 $G_A$  のある弧に対して、 $G_B$  のすべての弧がマッピング可能であるということではない。出力元あるいは入力先のリソースが異なる配線同士をマッピングすると、出力元と入力先のリソースもそれぞれマッピングされ、回路として矛盾が生じる。このような制約が弧のマッピングに存在することから、ある弧の組合せがマッピング可能かを、有向グラフ上で識別する必要がある。

そのため、リソースの種類 (I/OPad, LUT) に応じて、グラフの節点に異なるラベルをつける。そして、始点と終点のラベルの組が一致する弧同士のみをマッピング可能とする。図5(a)にマッピング可能な例を示す。2つの弧 AB は両端のラベルが一致するのでマッピング可能である。一方、図5(b)は、弧 AB が弧 AC, 弧 CB とは両端のラベルが不一致なので、マッピング不可能な例である。

#### 3.2.3 Compatibility Graph の生成

続いて、CG (Compatibility Graph) を生成する。図4(2)のように、CG の頂点は、 $G_A$  と  $G_B$  のマッピング可能な弧の組合せをそれぞれ頂点とする。この際、弧の組合せがマッピングできるかどうかは、前述のようにラベルに依存する。さらに、CG の2つの頂点が示す弧の組合せが同時にマッピング可能であるとき、その頂点同士は互いに接続される。図5(c)の2つの破線は、それぞれ CG の頂点となり、かつ、同時にマッピング可能なので頂点が接続される例である。一方、図5(d)の2つの破線は、それぞれ CG の頂点となるが、同時にマッピング不可能なので頂点は接続されない。

### 3.2.4 最大クリーク探索

CGの生成後、CGから最大クリークを探索する。クリークとは、全ての2点間に辺が存在する部分グラフであり、頂点数が最大となるクリークを最大クリークという。クリークのある頂点は、同じクリークの全ての頂点と隣接するので、CGの定義より、クリークは同時にマッピングが成立する弧の集合を意味する。よって、クリークの頂点数は共有される弧の数である。より多くの配線が統合コンテキスト上で共有されるためには、より多くの弧をマッピングさせる必要があるため、最大クリークを求めることが望ましい。

最大クリークの探索後、CGの最大クリークと、 $G_A$ 、 $G_B$ から統合コンテキストを表す有向グラフ $G_{uni}$ を生成する。まず、図4(3)のように、クリークの頂点で表現された $G_A$ 、 $G_B$ の弧が共有され、 $G_{uni}$ では1つの弧となる。続いて、図4(4)のように、 $G_A$ 、 $G_B$ でマッピングされなかった弧と頂点を追加することで、統合コンテキストを示す有向ユニバーサルグラフ $G_{uni}$ が完成する[7]。

### 3.3 設計手法の問題点

#### 3.3.1 最大クリーク探索時間

LUTベースの組合せ回路の場合、頂点がIPad、LUT、OPadの3種類しかないため、 $G_A$ 、 $G_B$ の辺数に対してCGの頂点数が大きく増加し、最大クリーク探索の計算時間が膨大となる。一般に、最大クリーク探索はNP困難であるため、多項式時間で解けるような効率の良い解法は望めない。関らの研究[8]によると、最大クリークの探索時間は、探索グラフの頂点数に対して指数関数的に増大することが報告されている。よって、実時間で最大クリークを探索するのは困難である。

そこで、CGの頂点数の増加を抑制するために、CGの節点生成の制約となるラベリングを拡張する。リソースの種類に応じたラベリングに加え、ラベルの種類を増やしつつ、最大クリークの頂点となるCGの頂点を減らさないようなラベリングを行えば、統合コンテキストの回路規模を最小に維持しながら、計算時間を抑えることができる。

#### 3.3.2 ラベリングの評価

ラベリングを拡張する際、どのようなラベリングを行うかが問題となる。ラベリングの絶対的な評価基準は、マッピングされるリソース量、すなわちCGの最大クリーク頂点数である。しかし、最大クリークは探索に膨大な時間を要するため、評価基準として実用的ではない。

我々はラベリングの評価指数の1つとして、配線共有数の上限値 $W_{shareMax}$ に着目した。 $W_{shareMax}$ とは、あるラベルを用いて統合コンテキストを生成する際、配線が最大で何本共有されるかを示す値である。統合コンテキストにおける配線共有数は、CGの最大クリーク頂点数と等しいので、 $W_{shareMax}$ は最大クリーク頂点数の上限値とも言える。 $W_{shareMax}$ は式(1)で定義される。

$$W_{shareMax} = \sum_i^{AllLabel} \sum_j^{AllLabel} \min(E(G_{Aij}), E(G_{Bij})) \quad (1)$$

ただし、 $E(G_{Aij})$ は始点のラベルが*i*、終点のラベルが*j*とな

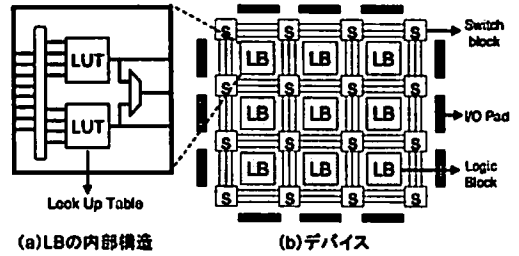


図6 アイランドスタイル再構成可能デバイス

るグラフAの弧の数を表す。

## 4. アーキテクチャモデル

提案している低消費エネルギー化手法を活かすために、図6に示すアーキテクチャモデルで実験を行った。モデルの基本構造は、プログラム可能な論理ブロックの二次元アレイとそれらを接続する配線エリアからなり、配線構造はスイッチブロックにより変更できる。配線共有型マルチコンテキスト化手法では、コンテキスト数が*k*個の場合、論理ブロックは切替え信号により、*k*通りの処理を行うことが要求される。マルチコンテキスト時には、それぞれのLUTに各コンテキストのLUTを実装し、MUXで入出力を切替えてコンテキストの切替えを実現する。これにより、再構成エネルギーが不要となる。また、非マルチコンテキスト時には、回路をクラスタリングして実装できる構成になっている。

## 5. 評価環境

本章では、まず評価方法を示し、評価に用いた回路について説明する。そして、オリジナル回路とデータ依存回路をマルチコンテキスト化した回路、および配線共有型マルチコンテキスト化手法を適用した回路について、それぞれの消費エネルギー、構成データ量の比較を行い、提案手法の有効性を検証する。なお、本稿ではコンテキスト数を2個に限定する。したがって、アーキテクチャモデルは図6(a)のように、論理ブロックは2つのLUTで構成する。

### 5.1 評価方法

回路の設計手順を説明する。まず入力回路から複数のデータ依存回路を設計し、それぞれ米国Altera社のQuartus II 5.0で論理合成する。続いて、バークレー大学で開発されたSIS[9]で4-LUTにマッピングして、BLIFネットリストを得る。そして、本稿で提案する手法で統合コンテキストを設計する。また、本実験では最大クリークの探索を1時間で打ち切っている。その場合は、探索時に得られた極大クリークを用いて統合コンテキストを設計する。

#### 5.1.1 消費エネルギーの見積り方法

本稿では、消費エネルギーの見積りにPowerModel[10]を用いる。PowerModelは、トロント大学において開発されたFPGA用配置配線ツールであるVPR (Versatile Place and Route) [11]の消費電力評価拡張版としてブリティッシュコロン

ピア大学において開発されたものである。PowerModel は、配線部分、論理ブロック部分それぞれの動作時に消費される動作消費電力、並びにリーク電力の評価を行うことができる。動作消費電力は、トランジスタの出力が切り替わるたびに消費されるエネルギーで、リーク電力は、トランジスタの動作に関係なく一定量消費されるエネルギーである。消費エネルギーの見積りは、これらの消費電力の特性を考慮して行う。さらに、マルチコンテキスト化した回路では、再構成回数を見積り、再構成エネルギーを加える。また、評価に使用するテクノロジーは 350nm とする。

### 5.1.2 マルチコンテキスト手法の消費エネルギー見積り

従来のマルチコンテキスト手法を適用する回路では、消費エネルギー  $P_{total}$  を式 2 のようにして求める。

$$P_{total} = P_{top} + P_A \times \alpha + P_B \times (1 - \alpha) + P_{conf} \quad (2)$$

$$P_{conf} = P_{conf}(A) \times N_A + P_{conf}(B) \times N_B$$

ここで、 $P_{top}$  はトップ回路の消費エネルギー、 $P_A$ 、 $P_B$  はそれぞれのデータ依存回路の消費エネルギーで、 $\alpha$  はデータ依存回路 A が使用される割合である。 $P_{conf}$  は再構成エネルギーの総和であり、 $N_A$ 、 $N_B$  はそれぞれ、データ依存回路 A、B に回路が切替わる回数である。 $P_{conf}(A)$ 、 $P_{conf}(B)$  は、それぞれデータ依存回路を切替える際の再構成エネルギーである。再構成エネルギーは、SRAM の記憶データが書き換えられる確率を 50% と想定し、トランジスタ数を見積って算出する [4]。

### 5.1.3 提案手法の消費エネルギー見積り

統合コンテキストでは、2 つの LUT で論理ブロックを構成するが動作消費電力は常に LUT1 つ分である。これは、アーキテクチャモデルの構造により、片方の LUT は入出力を遮断されるためである。そして、リーク電力は LUT2 つ分となる。よって、本実験では統合コンテキストの消費エネルギー  $P_{total}$  を式 (3) に示す方法で見積る。

$$P_{total} = P_{top} + MCP_{net} + MCP_{logic}$$

$$MCP_{logic} = DP_{logic}(A) \times \alpha + DP_{logic}(B) \times (1 - \alpha) + SP_{logic}(A) + SP_{logic}(B) \quad (3)$$

$\alpha$  は式 (2) と同様に、コンテキスト A が実行される割合である。 $MCP_{logic}$  を統合コンテキストの論理ブロックの消費エネルギー、 $MCP_{net}$  を統合コンテキストの配線の消費エネルギーとおき、 $DP_{logic}(X)$ 、 $SP_{logic}(X)$  をそれぞれ回路 X の動作消費電力、リーク電力とする。

### 5.1.4 構成データ量の見積り方法

構成データは、LUT に格納されるコンテキストと、配線上のコネクションブロックからなる。コンテキスト分の構成データ量は、LUT 使用数と LUT のビット数の積である。コネクションブロック分は、配線数、論理ブロック間の配線の平均接続点と、コネクションブロックのビット数の積として見積る。平均接続点数は、それぞれの回路を VPR で配置配線して得られる値を使用し、各コネクションブロックは 1 ビットの SRAM で構成されているとする。それぞれの回路の構成データ量 (CD) を

式 (4) に示す。 $N_{LUT}$  は LUT 数、 $K$  は LUT の入力数、 $N_{NT}$  は配線数、 $Av.Bends$  は平均接続点数である。

$$CD = N_{LUT} \times 2^K + N_{NT} \times Av.Bends \quad (4)$$

## 5.2 評価回路

評価回路には、JPEG エンコーダと RGB-YIQ 変換回路を用いた [12]。JPEG エンコーダは、DCT、ZIGZAG スキャン、量子化 (QNR, Quantization & Rounding)、符号化 (RLE, Run Length Encoder) などのモジュールに分割できる。DCT モジュールは、同様な 64 個の MAC (積和演算) モジュールが並列になっており、本稿では、2 つの MAC モジュール DCT04 と DCT34 に提案手法を適用する。データ依存回路を生成する際に固定するフリップフロップは、いずれも論理ゲートが最も削減されるフリップフロップを選択する。また、370 サイクルのテストベンチでシミュレーションし、再構成回数を見積り、マルチコンテキスト化手法の評価を行う。

## 6. 評価・考察

本章では、提案手法の評価結果を示し、考察を行う。

### 6.1 評価結果

表 1 に、評価結果を示す。3 つの回路 DCT04 と RGB-YIQ、DCT34 に対して、従来のマルチコンテキスト化手法、配線共有型マルチコンテキスト化手法をそれぞれ適用し、論理ブロック数、配線数、消費電力、構成データを見積った結果である。また、それぞれの結果はオリジナル回路の値で正規化している。構成データが整数でない理由は、平均接続点数  $Av.Bends$  が少数となるからである。

結果から、いずれの回路でもデータ依存回路を作成しマルチコンテキスト化すると、論理ブロック数と配線数が削減されることがわかる。配線共有型マルチコンテキスト化手法では、オリジナル回路と比較して論理ブロック数が平均で 58.1%、配線数が平均で 47.7% それぞれ削減された。また、消費電力は平均で 19.7%、構成データは平均で 45.6% それぞれ削減された。さらに、従来のマルチコンテキスト化手法と比較すると、論理ブロック数、配線数は増加するが、消費電力と構成データは削減される。論理ブロック数は平均で 1.6%、配線数は平均で 4.2% それぞれ増加し、消費電力は平均で 14.8%、構成データは平均で 1.1% それぞれ削減された。

### 6.2 考察

評価結果より、データ依存回路を作成する際に削減される LUT 数は大ききばらつきがあることがわかる。これには、回路を分割した後にテクノロジーマッピングを施したため、トップ回路やオリジナル回路で最適化の度合いが異なるという要因も考えられる。

また、配線共有型マルチコンテキスト化手法を従来のマルチコンテキスト化手法と比較すると、DCT 回路ではいずれも論理ブロックと配線数が増加する。これは、統合コンテキストを生成する際に冗長な回路が発生するためである。しかし、RGB-YIQ 変換回路では論理ブロック数、配線数が一致している。これは、一方のデータ依存回路の配線構造が、もう一方の

表1 評価結果

		オリジナル回路	%	MC化手法	%	配線共有型MC化手法	%
dct04	Logic Block 個数	212	100	171	80.7	180	84.9
	Routing 本数	251	100	225	89.6	234	93.2
	Power[w]	1.49E-02	100	2.24E-02	150.7	1.70E-02	114.5
	構成データ [bit]	4025.3	100	4464.0	110.9	4379.0	108.8
rgb-yiq	Logic Block 個数	4200	100	1082	25.8	1082	25.8
	Routing 本数	4227	100	1671	39.5	1671	39.5
	Power[w]	1.75E-01	100	1.15E-01	65.9	1.13E-01	64.4
	構成データ [bit]	71428.8	100	22756.2	31.9	22395.7	31.4
dct34	Logic Block 個数	1367	100	199	14.6	207	15.1
	Routing 本数	1383	100	328	23.7	336	24.3
	Power[w]	6.00E-02	100	2.32E-02	38.7	1.92E-02	32.0
	構成データ [bit]	23990.9	100	5716.3	23.8	5542.7	23.1

データ依存回路の配線構造を内包できたためである。

また、一般に配線共有型マルチコンテキスト化手法は従来のマルチコンテキスト化手法に対して論理ブロック数と配線数が多くなるにも関わらず、消費電力が削減されている。これは、配線構造部分の共有化により再構成エネルギーが不要なためである。

## 7. まとめと今後の課題

### 7.1 まとめ

本研究では、配線共有型マルチコンテキスト化手法の提案と、動的再構成デバイスにおける低消費エネルギー化手法の評価を行った。配線共有型マルチコンテキスト化手法の利点は、配線部分の再構成を行うことなくコンテキストの切替えが可能という点である。さらに、提案したアーキテクチャモデルならば、論理ブロック部分の再構成も不要である。この手法とアーキテクチャモデルをデータ依存回路を用いた低消費エネルギー化手法に適用した結果、オリジナル回路と比較して消費エネルギーを平均で19.7%削減し、さらに構成データを平均45.6%削減した。

### 7.2 今後の課題

今後の課題として、以下のことが挙げられる。

本論文では、データ依存回路の規模を、LUT数100以下の回路に限定した。これは、データ依存回路から統合コンテキストを生成する際、データ依存回路のLUT数に対してCGの頂点数が爆発的に増加し、最大クリーク探索時にメモリ不足が発生するためである。結果として、評価フローを通すことができる回路が限定されることになった。よって統合コンテキスト設計手法の改良が必要である。解決方法として、入力回路を複数の小さな回路に分割し、それぞれの統合コンテキストを求め、さらにそれらを統合するといった問題の分割化が考えられる。今後は、様々な回路を用いて本手法の有効性を詳細に評価したい。

また、回路の検証が必要であると考え、データ依存回路とトップ回路はシミュレーションの結果、正しく動作することが確認されているが、統合コンテキストについてはまだ動作検証を行っていない。よって、統合コンテキストが正しく動作するか、確認が必要である。

## 文 献

- [1] 末吉 敏則, 天野 英晴: リコンフィギャラブルシステム, オーム社, 2005
- [2] 飯田 全広, 末吉敏則: “リコンフィギャラブル・ロジック向き論理ブロックの提案と評価,” 情報処理学会論文誌, vol.43, No.5, pp.1181-1190, 2002.
- [3] Sato, T.: “Dynamically Reconfigurable Processor DAP/DNA-2 and Development DAP/DNA-FW,” IEEE CoolChips VII, Apr 2004
- [4] 今井 茂毅, 飯田 全広, 末吉 敏則: “RLDの動的再構成機能を利用した消費エネルギー削減手法,” 第12回FPGA/PLD Design Conference ユーザプレゼンテーション論文集 pp.57-64, Jan, 2005.
- [5] 今井茂毅, 飯田全広, 末吉敏則: “自律再構成による低消費エネルギー化手法” 信学技法 RECONF2005-44(2005-9), Vol.105, No.288, pp.10-24, Sep. 2005.
- [6] Shigeki IMAI, Masahiro HIDA and Toshinori SUEYOSHI.: “A low power design method using multi-context dynamic reconfiguration,” Proc. of The 20th Commemorative International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2005), Vol. 2, pp.563-564, July, 2005.
- [7] ZHINING HUANG and SHARAD MALIK, NAHRI MOREANO and GUIDO ARAUJO: “The Design of Dynamically Reconfigurable Datapath Coprocessors,” ACM Transactions on Embedded Computing System, vol.3, No.2, pp.361-384, May 2004.
- [8] 関 友和, 富田 悦次: “分枝限定法を用いた最大クリーク抽出アルゴリズムの効率化,” 信学技法 vol.101, no. 376, COMP 2001-50, pp101-108, 2001.
- [9] Sentovich, E. M., et al.: “SIS:A System for Sequential Circuit Synthesis,” Memorandum No. UCB/ERL M92/41 1992.
- [10] KARA K.W.POON, STEVEN J.E.WILTON and ANDY YAN: “A Detailed Power Model for Field Programmable Gate Arrays,” ACM Transactions on Design Automation of Electronic Systems, pp279-302, April 2005.
- [11] V. Betz and J. Rose: “VPR: A New Packing, Placement and Routing Tool for FPGA Research,” Proc. of Seventh International Workshop on Field-Programmable Logic and Applications, pp213-222,1997. .
- [12] “OPENCORES.ORG,” <http://www.opencores.org/>