

## LUT 段数最小かつ個数極小な LUT 型 FPGA 向けテクノロジ・マッピング

高田 大河<sup>†</sup> 松永 裕介<sup>††</sup>

<sup>†</sup>九州大学 大学院システム情報科学府 情報工学専攻

〒819-0395 福岡県福岡市西区元岡 744

<sup>††</sup>九州大学 大学院システム情報科学府 情報工学部門

〒819-0395 福岡県福岡市西区元岡 744

E-mail: †{taiga,matsunaga}@c.csce.kyushu-u.ac.jp

あらまし 本稿では、出力における LUT 段数が最小かつ LUT 個数が最小なネットワークの生成を目的とした LUT 型 FPGA 向けテクノロジ・マッピングのアルゴリズムを提案する。従来は、LUT 段数に関してのみ最小であり、LUT 個数に関しては最小である保証がないヒューリスティックな手法が提案されてきた。提案手法は、ネットワーク各部の LUT 段数が最小である制約のもとで LUT 個数が最小なネットワークを得る問題を厳密に解くものである。提案手法と既存手法の比較実験を行い、提案手法の LUT 個数における優位性を示した。

キーワード EDA, FPGA, 論理合成, テクノロジ・マッピング

## Depth-Optimum and Area-Optimal Technology Mapping for LUT-based FPGAs

Taiga TAKATA<sup>†</sup> and Yusuke MATSUNAGA<sup>††</sup>

<sup>†</sup> Graduate School of Information Science and Electrical Engineering, Kyushu University

744 Motoooka, Nishiku, Fukuoka 819-0395 JAPAN

<sup>††</sup> Faculty of Information Science and Electrical Engineering, Kyushu University

744 Motoooka, Nishiku, Fukuoka 819-0395 JAPAN

E-mail: †{taiga,matsunaga}@c.csce.kyushu-u.ac.jp

**Abstract** This paper describes the technology mapping algorithm for LUT based FPGA to minimize the total number of LUTs, under the optimum depth of the mapped LUT network. Some heuristic algorithms have been proposed which derive an LUT network with optimum depth. However those algorithms do not guarantee that the number of LUTs of the LUT network is minimum. The proposed algorithm exactly solves the problem of obtaining LUT network whose number of LUTs is minimum, and the level of each node is minimum. The experiments shows that the proposed algorithm derives LUT networks whose number of LUTs are smaller than the number of LUTs of network which are derived by the existing algorithms.

**Key words** EDA, FPGA, Logic Synthesis, Technology Mapping

### 1. はじめに

少量生産の LSI (Large Scale Integration) を低コストで実現するための手法の一つに、再構成可能デバイスを用いた設計がある。再構成可能デバイスは、チップの製造後にチップが実現する機能を変更することができる。現在、再構成可能デバイスとして LUT (Look-Up Table) 型 FPGA (Field Programmable Gate Array) が広く用いられている。LUT 型 FPGA の問題点は、性能と消費電力である。LUT 型 FPGA は再構成のための

機構を持つため、設計ごとに製造した LSI に比べて性能や消費電力の点で劣っていることが多い。FPGA の性能、消費電力を向上させるための課題は、ハードウェア・アーキテクチャと、ハードウェア上に設計を実現するための回路合成技術である。

遅延最小な回路合成を行うために重要な要素技術のひとつとして、テクノロジ・マッピングのアルゴリズムがある。LUT 型 FPGA におけるテクノロジ・マッピングは論理式を表すネットワークを入力として、LUT からなる論理的に等価なネットワークを生成する。LUT とは、決められた入力数以下の任意の論

理論関数を実現できる素子である。遅延が小さい回路を生成するためには、テクノロジー・マッピングにおいて深さが最小で LUT 個数が少ないネットワークを生成することが望ましい。深さととは、ネットワークの出力における最大段数のことである。LUT 型 FPGA の遅延は、LUT 内部の遅延と LUT 間の配線部分の遅延からなる。個々の LUT の遅延はほぼ一定である。テクノロジー・マッピングの段階では LUT 間の配線部分の遅延を見積もることができないため、各々の LUT 間の配線に一定の固有遅延を仮定するのが一般的である。これらに基づき、ネットワークの深さで遅延の見積もりを行うことが出来る。また、遅延最小な回路を生成するためには、深さが同じならば LUT の個数が少ない方が望ましい。これは、配線の自由度を高めるためである。

深さが最小かつ LUT の個数が最小な解を求める問題は、難しいと考えられる。深さに限って言えば、厳密最小解を求めるアルゴリズムが存在する [1] [3] [4]。一方、LUT の個数最小解を得る問題は DAG (Directed Acyclic Graph: 非循環有向グラフ) カバリング問題 [2] として定式化できるが、この問題は NP 困難と呼ばれる問題のクラスに相当する。LUT の個数最小解を得る多項式時間のアルゴリズムは見つかっていない。発表者が知る限り深さが最小で個数が最小な解を求める問題の難しさは証明されていないが、DAG カバリング問題と同等かそれ以上に難しい問題であると考えられる。

著者らは、ネットワークの各部が段数最小である制約の下で LUT の個数が最小な解を得る厳密なアルゴリズムを考案した。著者らが考案したアルゴリズムをプログラム Tmap として実装し、既存のアルゴリズム FlowMap [1]、DAOmap [3] と比較する実験を行った。実験の結果、Tmap の解の LUT の個数は FlowMap の解の LUT の個数と比較して最小で 58%、平均で 79%、DAOmap の解の LUT の個数と比較して最小で 75%、平均で 99% であった。

本稿は、以下の構成をとる。2 章で準備として、本稿で用いる語句の定義を説明する。3 章で、深さ最小かつ LUT の個数が少ない解を生成するテクノロジー・マッピングの既存のアルゴリズムを紹介する。4 章で、提案手法を説明する。5 章で、実験について述べる。6 章で本稿をまとめる。

## 2. 準備

LUT 型 FPGA 向けのテクノロジー・マッピングは、論理式を変えネットワークが与えられ、 $k$  入力以下の LUT からなる論理的に等価なネットワークを生成する処理である。 $k$  は LUT の最大入力数であり、FPGA のアーキテクチャによって決まる。

テクノロジー・マッピングの入力として与えられるネットワークは、 $K$  バウンデッド・ネットワークである。 $K$  バウンデッド・ネットワークとは、各々のノードが  $k$  入力以下であるような DAG である。DAG とは閉路を持たない有向グラフのことであり、ノードの集合  $V$  および辺の集合  $E$  からなる。各々のノードは、 $k$  入力以下の論理関数を持つ。ノード  $i \in V$  の出力がノード  $j \in V$  における論理関数の入力であるとき、有向辺  $(i, j) \in E$  が存在する。入力辺を持たないノードをプライマリ・

インプット (PI)、出力辺を持たないノードをプライマリ・アウトプット (PO) と呼ぶ。

ネットワーク  $N(V(N), E(N))$  におけるノード  $v$  のファンイン集合とは、 $v$  の入力辺に接続するノードの集合であり、 $INPUT(v)$  と記す。さらに、 $v$  のトランシティブ・ファンイン・グラフ  $TFI_v = (V(TFI_v), E(TFI_v))$  とは  $v$  から全ての PI へ辿ることで得られる  $N(V(N), E(N))$  の部分グラフを表す。 $V(TFI_v), E(TFI_v)$  はそれぞれ次のように定義される。

$$V(TFI_v) = \bigcup_{u \in INPUT(v)} v \cup V(TFI_u)$$

$$E(TFI_v) = \{e \mid e \in E(N) \wedge e \in V(TFI_v) \times V(TFI_v)\}$$

カットを定義する。 $K$  バウンデッド・ネットワーク  $N = (V(N), E(N))$  が与えられたとき、 $t \in V(N)$  におけるカット  $(X_t, \bar{X}_t)$  とは、 $V(TFI_t)$  の分割  $X, \bar{X}$  のうち、 $V(TFI_t)$  に含まれる PI の集合を  $PI_s$  としたとき、 $PI_s \subseteq X_t, t \in \bar{X}_t$  であるようなものである (ノードの集合  $V$  に対し、 $X \subseteq V, \bar{X} \subseteq V, X \cap \bar{X} = \emptyset$  であるような  $X, \bar{X}$  を  $V$  の分割と呼ぶ)。このとき、 $t \in V(N)$  をカット  $(X_t, \bar{X}_t)$  の根と呼ぶ。ただし本稿では、特に次の 2 つの条件を満たすものをコンベックスなカットとする。まず、 $\bar{X}_t$  からなる誘導部分グラフが連結であることである。次に、 $\bar{X}_t$  に含まれるノードが  $X_t$  に含まれるノードへ出力しないこと、すなわち  $\forall i \in \bar{X}_t, \forall j \in X_t, (i, j) \notin E(TFI_t)$  を満たすことである。 $X_t$  に含まれるノードのうち、 $\bar{X}_t$  のいずれかのノードと接続するものをカット・ノードセットと呼び  $BORDER(X_t, \bar{X}_t)$  で表す。 $BORDER(X_t, \bar{X}_t) = \{v \mid v \in X_t, \exists w \in \bar{X}_t, (v, w) \in E(TFI_t)\}$  である。 $|BORDER(X_t, \bar{X}_t)| \leq k$  であるような、コンベックスなカットを  $K$  フィージブル・カットと呼ぶ。 $K$  が 3 である場合の、 $K$  フィージブル・カットの例を図 1 に示す。 $K$

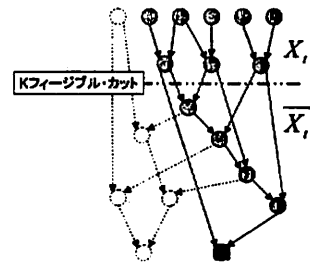


図 1  $K$  フィージブル・カットの例 ( $K=3$ )

フィージブル・カット  $(X_t, \bar{X}_t)$  が与えられたとき、 $\bar{X}_t$  の誘導部分グラフは、 $BORDER(X_t, \bar{X}_t)$  を入力とする論理的に等価な LUT で被覆することが出来る。

LUT 型 FPGA 向けのテクノロジー・マッピングは、 $K$  バウンデッド・ネットワークの全てのノードを、 $K$  フィージブル・カットの  $\bar{X}_t$  で被覆する問題であると見なすことが出来る。すなわち、テクノロジー・マッピングにおいて考えなければならない問題は、 $K$  バウンデッド・ネットワークから得られる全ての  $K$  フィージブル・カットから、 $\bar{X}_t$  で全てのノードを被覆する

ような K フィージブル・カットの組み合わせを選択するというものである。ただし、LUT の入力が他の LUT の出力であるための条件として、選択した K フィージブル・カット  $(X, \bar{X})$  の  $v \in BORDER(X, \bar{X})$  が選択した他の K フィージブル・カット  $(X_v, \bar{X}_v)$  の根となっていなければならない。

選択した K フィージブル・カットの  $\bar{X}_i$  から得られる誘導部分グラフを、論理的に等価な K 入力以下の LUT で置換することによって、各々のノードが LUT を表すネットワークが得られる。各々のノードが LUT を表すネットワークを、LUT ネットワークと呼ぶことにする。LUT ネットワークは、テクノロジー・マッピングの解である。LUT ネットワークに対し、段数と深さを定義する。LUT ネットワークのノード  $u$  における段数とは、全ての PI から  $u$  までのパスのうち、最長なパスの長さである。また、LUT ネットワークの深さとは、全てのノードにおける段数の最大値である。本稿では特に、K バウンデッド・ネットワークから得られる LUT ネットワークのうち、全ての LUT が段数最小であるような LUT ネットワークを、ホール・コンプレストなネットワークと呼ぶ。

K バウンデッド・ネットワーク  $N(V, E)$  の各々のノードに対し、ラベルという値を定義する。ノード  $t$  におけるラベルとは、 $N$  の部分グラフ  $N_t(V(TFI_t), E(TFI_t))$  を深さが最小になるように LUT で被覆した場合の LUT ネットワークの深さのことである。 $t$  における K フィージブル・カットの集合を  $C(t)$  と表す。さらに K フィージブル・カット  $(X_i, \bar{X}_i)$  に対して、 $BORDER(X_i, \bar{X}_i)$  に含まれるノードのラベルのうち最大値を  $m(X_i, \bar{X}_i)$  と表す。このとき、 $t$  におけるラベル  $label(t)$  は以下の式に従って計算することが出来る。

$$label(t) = \min_{(X_i, \bar{X}_i) \in C(t)} m(X_i, \bar{X}_i) + 1 \quad (1)$$

$t \in V$  において、 $m(X_i, \bar{X}_i)$  が最小であるような K フィージブル・カット  $(X, \bar{X}_i)$  を、ラベルに対応する K フィージブル・カットと呼ぶ。

### 3. 関連研究

#### 3.1 FlowMap

FlowMap [1] は、深さが最小かつ LUT 個数が少ない LUT ネットワークを生成するテクノロジー・マッピングのアルゴリズムである。FlowMap は、まずラベルに対応する K フィージブル・カットを列挙する。ラベルに対応する K フィージブル・カットを、PO 側から PI 側へヒューリスティックな手法で選択していくことによって、ホール・コンプレストな LUT ネットワークを生成する。次に、ホール・コンプレストであることを保ったまま LUT の個数を減らすヒューリスティックな後処理を行う。FlowMap が生成する解は、ホール・コンプレストであることにより深さ最小であることが保証されるが、ホール・コンプレストな制約下において LUT 個数が最小である保証はない。

#### 3.2 DAOmap

DAOmap [3] は、深さが最小かつ LUT 個数が少ない LUT ネットワークを生成するテクノロジー・マッピングのアルゴリズムのうち、現在のところ最も LUT の個数が少ない解を生成で

きるとされるものの 1 つである。DAOmap は、段数に関してクリティカルなパスにおいてはラベルに対応する K フィージブル・カットを選択する。一方、段数に余裕があるパスにおいては、深さが増大しない範囲で段数が増えることを許容し、LUT 個数が少ない LUT ネットワークを導くような K フィージブル・カットを選択する。DAOmap が生成する解は、深さ最小であることは保証されるが、カットの選択においてヒューリスティックな指標を用いているため、LUT の個数に関して最小である保証はない。

### 4. 提案手法

提案手法は、ホール・コンプレストな制約の下で、LUT の個数が厳密に最小な LUT ネットワークを生成するものである。まず K バウンデッド・ネットワークの全てのノードにおける K フィージブル・カットの列挙を行い、各々のノードのラベルを計算する。さらに、ラベルに対応する K フィージブル・カットの集合から導かれる DAG カバリング問題を厳密に解くことによって、解を得る。

#### 4.1 提案手法のアルゴリズム

##### 4.1.1 K フィージブル・カットの列挙とラベルづけ

K バウンデッド・ネットワークの PI から PO へ向かうトポロジカルな順序で、各々のノードにおける K フィージブル・カットの列挙、およびラベルづけを行う。

##### a) K フィージブル・カットの列挙

$v \in V(N)$  における K フィージブル・カットの列挙を行う。まず  $INPUT(v)$  の全てのノードに対して、それぞれに保持された K フィージブル・カットのうちの一つ  $(X, \bar{X})$  を用い、それらの  $\bar{X}$  の和集合をとってカットを生成する。すなわち、 $INPUT(v) = \{v_0, v_1, v_2, \dots, v_n\}$ 、 $i$  において列挙された K フィージブル・カットの集合を  $C_i$  と表すと、

$$x \in C_{v_0} \times C_{v_1} \times C_{v_2} \times \dots \times C_{v_n}$$

$$(X_i, \bar{X}_i) \in C_i : x \text{ の } i \text{ 番目の要素}$$

$$\bar{X}_v = \{v\} + \bigcup_{v_i \in INPUT(v)} \bar{X}_i$$

$$X_v = V(TFI_v) - \bar{X}_v$$

に従ってカットの集合  $\{(X_v, \bar{X}_v)\}$  を列挙する。上記の計算により、 $v$  において最大で  $\prod_{i \in INPUT(v)} |C_i|$  個のカットを列挙する

ことになる。次に列挙したカット  $(X_v, \bar{X}_v)$  のうち、以下の 2 つの条件を満たすものを K フィージブル・カットとして保持する。まず、 $|BORDER(X_v, \bar{X}_v)|$  が  $k$  以下であることである。次に、コンベックスであること、すなわち  $\bar{X}_v$  に含まれるノードが  $X_v$  に含まれるノードへ出力しないことである。カットがコンベックスであるためには、 $\bar{X}_v$  からなる誘導部分グラフが連結であるという条件も必要であるが、上記の計算によって得たカットがこれを満たすことは自明である。

##### b) ラベルづけ

ノードにおいて列挙した K フィージブル・カットの集合を用い、(1) 式よりノードのラベルの値を計算する。PI に対して

はラベルとして 0 を割り当てる。また、ラベルに対応する K フィージブル・カットの集合を保持する。

#### 4.1.2 DAG カバリング問題としての定式化

K バウンデッド・ネットワークと K フィージブル・カットの集合が与えられたときに、LUT の個数最小解を生成する問題を、DAG カバリング問題として定式化する。

与えられた K バウンデッド・ネットワークを  $N(V(N), E(N))$  とする。与えられた K フィージブル・カットの集合を  $C(N)$ 、解として選択した K フィージブル・カットの集合を  $C_{LUT}(N) \subseteq C(N)$  とする。K フィージブル・カット  $c \in C(N)$  が  $C_{LUT}(N)$  に含まれる場合に 1、そうでない場合に 0 となる論理変数  $x_c$  を導入する。K フィージブル・カット  $c(X, \bar{X}) \in C(N)$  に対して、 $\bar{X}$  に被覆されるノードの集合を  $COVER(c)$  と表す。K フィージブル・カット  $c$  の根にあたるノードを  $ROOT(c)$  と表す。また、ノード  $v$  を  $\bar{X}$  で被覆できる K フィージブル・カットの集合を  $\{q_1^v, q_2^v, \dots\}$ 、ノード  $v$  を根とする K フィージブル・カットの集合を  $\{r_1^v, r_2^v, \dots\}$  と表す。

$C_{LUT}(N)$  から生成される LUT ネットワークが、 $N(V(N), E(N))$  と論理的に等価な DAG であるためには、次の 2 つの条件を満たすことが必要である。

- 制約 1: 全てのノードの被覆

$C_{LUT}(N)$  に含まれる  $c$  の  $\bar{X}$  によって、全ての  $v \in V(N)$  が被覆されていないといけない。

- 制約 2: LUT の入出力関係

$C_{LUT}(N)$  に含まれる全ての  $c$  の  $BORDER(c)$  における他の K フィージブル・カットもまた、 $C_{LUT}(N)$  に含まれていなければならない。

DAG カバリング問題とは、これら 2 つの条件を満たす  $C_{LUT}(N)$  のうち、最も要素数が少ないものを求める問題である。

- a) 制約 1: 全てのノードの被覆

$C_{LUT}(N)$  に含まれる  $c$  の  $\bar{X}$  によって、全ての  $v \in V(N)$  が被覆されているとは、以下が成立することを言う。

$$\forall v \in V, \exists c \in C_{LUT}(N), v \in COVER(c)$$

この条件を表す論理式は、次のようになる。

$$\prod_{v \in V} (x_{q_1^v} + x_{q_2^v} + \dots) \quad (2)$$

- b) 制約 2: LUT の入出力関係

$C_{LUT}(N)$  に含まれる全ての  $c$  の  $BORDER(c)$  における他の K フィージブル・カットもまた、 $C_{LUT}(N)$  に含まれていなければならない。この条件は、以下のように表される。

$$\forall c \in C_{LUT}(N), \forall u \in BORDER(c).$$

$$\exists l \in C_{LUT}(N), ROOT(l) = u$$

この条件を表す論理式は、以下のようになる。

$$\prod_{c \in C_{LUT}(N)} \left( \prod_{u \in BORDER(c)} (x_{\bar{r}_1^u} + x_{\bar{r}_2^u} + \dots) \right) \quad (3)$$

- c) 目的関数: LUT 個数の最小化

目的関数は次式で表される。

$$\sum_{i \in C(N)} x_i \quad (4)$$

DAG カバリング問題は、(2),(3) 式を満たす  $\{x_i\}$  ( $i \in C(N)$ ) の値の組み合わせのうち、(4) 式の値が最小となるものを見つける問題である。

#### 4.2 提案手法の解の深さと LUT 個数

次の定理を証明する。

定理: ラベルに対応する K フィージブル・カットのみを用いて DAG カバリング問題を解くことで得られる解は、ホール・コンプレストな制約下での LUT 個数最小解である。

証明: まず、ラベルに対応する K フィージブル・カットのみの集合から得られる解が、ホール・コンプレストであることを証明する。ラベルに対応する K フィージブル・カット  $(X, \bar{X})$  は、全ての  $v \in BORDER(X, \bar{X})$  が段数最小な LUT で被覆される場合のみ、 $\bar{X}$  に対応する LUT が段数最小であることを保証するものである。

- d) PI のみに接続する LUT

$BORDER(X, \bar{X})$  の要素が PI のみである K フィージブル・カット  $(X, \bar{X})$  は、 $\bar{X}$  に対応する LUT が段数最小であることが自明である。

- e) 中間の LUT

ラベルに対応する K フィージブル・カット  $(X, \bar{X})$  について考える。全ての  $v \in BORDER(X, \bar{X})$  が、段数最小な LUT で被覆されていると仮定する。このとき、K フィージブル・カット  $(X, \bar{X})$  の性質から、 $\bar{X}$  に対応する LUT は段数最小であることが保証される。

d), および e) より、ラベルに対応する K フィージブル・カットのみの集合から生成される LUT ネットワークは、全ての LUT に関して段数最小、すなわちホール・コンプレストである。

また、ホール・コンプレストな LUT ネットワークに含まれる LUT に対応する K フィージブル・カットが、ラベルに対応する K フィージブル・カットであることは自明である。

よって、ラベルに対応する K フィージブル・カットのみを用いて DAG カバリング問題を解くことで得られる解は、ホール・コンプレストな制約下における LUT 個数最小解となっている。

## 5. 実 験

提案手法のアルゴリズムをプログラム Tmap として実装し、FlowMap [1], DAOmap [3] との比較を行った。Tmap, FlowMap, DAOmap に対し、MCNC ベンチマークの各ネットワークから生成した K バウンデッド・ネットワークを入力として与えて解を得た。Tmap は、DAG カバリング問題を整数線形計画問題として定式化し、ILOG 社の線形計画問題ソルバーである CPLEX を用いて解を得ている。Tmap, FlowMap, DAOmap はいずれも、解が深さ最小であることを保証するものなので、同じ入力に対する解の深さは等しい。各プログラムの解を評価するために、LUT の個数で比較を行った。

実験結果は、表 1 の通りである。図において bench. の列は、MCNC ベンチマークの各ネットワーク名である。Fmap,Dmap,Tmap の列は、それぞれ FlowMap,DAOmap,Tmap のプログラムが生成した解の LUT 個数である。T/F, T/D の列は、それぞれ Tmap と FlowMap, Tmap と DAOmap の解の LUT 個数比を示している。Tmap は NP 困難である DAG カバリング問題を厳密に解くものなので、ベンチマークの幾つかに関して計算時間がかかりすぎるために解を得ることができなかった。ここでは、MCNC ベンチマークのうち、Tmap が DAG カバリング問題の解を得ることが出来たものについてのみ結果を示している。

表 1 Tmap,FlowMap,DAOmap の解における LUT 個数

bench.	FMap	Dmap	Tmap	T/F	T/D
C17	2	2	2	100%	100%
att1	4	4	4	100%	100%
att9	17	16	16	94%	100%
att19	16	16	16	100%	100%
att17	27	21	18	67%	86%
att5	26	20	20	77%	100%
att2	32	28	21	66%	75%
att12	55	39	38	69%	97%
b9	53	45	41	77%	91%
att18	69	52	49	71%	94%
att11	74	59	60	81%	102%
9sym1	75	60	59	79%	98%
C432	104	82	77	74%	94%
z4m1	44	43	41	93%	95%
apex7	112	85	87	78%	102%
att4	90	69	69	77%	100%
f51a	58	52	51	88%	98%
C880	170	116	120	71%	103%
C499	80	66	66	83%	100%
att7	172	128	125	73%	98%
att3	41	41	41	100%	100%
C1355	80	66	66	83%	100%
C1908	190	150	147	77%	98%
att20	212	160	169	80%	106%
att14	297	244	244	82%	100%
att13	276	180	190	69%	106%
apex6	291	198	209	72%	106%
att23	405	221	241	60%	109%
rot	626	400	412	66%	103%
att22	435	279	297	68%	106%
att8	746	626	618	83%	99%
att16	606	442	512	84%	116%
att15	1068	655	616	58%	94%
Ave.				79%	99%

### 5.1 FlowMap との比較

Tmap は FlowMap と比較して LUT の個数が最小で 58%、平均で 79% の解を生成した。両者の LUT 個数比が 100% であるようなベンチマークも見られるが、これはネットワークの規

模が小さく解空間そのものが小さいベンチマークで、FlowMap が LUT 個数最小解を生成することに成功しているものである。ほとんどのベンチマークにおいて、Tmap は FlowMap よりも LUT 個数が少ない解を生成できることを確認できた。

Tmap と FlowMap が同じホール・コンプレストな制約下での解空間を探索するものであり、かつ両者の LUT 個数の比が大きいことから、FlowMap が行う K フィジブル・カット選択のヒューリスティックは LUT 個数削減に対して十分に効果的ではないことがわかる。MCNC ベンチマークに限れば、FlowMap には LUT の個数を平均で約 20% ほど改善できる余地があることが確認できる。

### 5.2 DAOmap との比較

Tmap は DAOmap と比較して、LUT の個数がほぼ同等の解を生成した。ほとんどのベンチマークに対し、両者の LUT 個数比はプラスマイナス 10% 未満である。

Tmap は、DAOmap よりも小さい解空間を探索している。Tmap はホール・コンプレストな制約における解空間を探索しているが、これは深さ最小なテクノロジ・マッピングの解空間の部分集合である。Tmap はホール・コンプレストな解を生成した後に、さらに大きな解空間を探索する後処理を行う余地が残っている。Tmap に追加する後処理として具体的には、解のホール・コンプレストな制約を取り除き、段数がクリティカルでないパスに関して、LUT ネットワークの深さが増大しない範囲で LUT の個数をさらに減らすものが考えられる。Tmap にこのような後処理を追加した場合、Tmap は DAOmap よりも LUT 個数が少ない解を生成できる可能性がある。

## 6. おわりに

本稿では、ラベルに対応する K フィジブル・カットのみを用いて DAG カバリング問題を解くことで、ホール・コンプレストな制約下での LUT 個数最小解が求まることを証明した。実験では提案手法をプログラム Tmap として実装し、LUT 個数が FlowMap と比較して平均 79%、DAOmap と比較して平均 99% の解を得た。実験結果より、FlowMap が行う K フィジブル・カットの選択方法には改善の余地があること、Tmap が DAOmap と同等の LUT 個数で解を生成できることを確認した。

今後の課題は、段数に関してクリティカルでないパスにおいて、LUT ネットワークの深さが増大しない範囲で LUT の個数を削減する方法の開発である。また、Tmap は NP 困難のクラスに属する DAG カバリング問題を厳密に解くため、規模が大きいベンチマークの解を求めることができないという問題点がある。DAG カバリング問題のヒューリスティックな解法を開発することもまた、今後の課題である。

## 謝 辞

本研究の一部は、科学研究費補助金(学術創成研究費(2))(課題番号: 14GS0218)による。

## 文 献

- [1] Jason Cong, Y.Ding, "FlowMap: An Optimal Technology

- Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *IEEE Transactions on Computer Aided Design*, 13(1):1-12, Jan. 1994.
- [2] Richard Rudell L., "Logic synthesis for VLSI design," Ph.D.thesis, University of California, Berkeley, 1989.
  - [3] Deming Chen, Jason Cong, "DAOmap: A Depth-optimal Area Optimization Mapping Algorithm for FPGA Designs," *IEEE International Conference on Computer Aided Design*, pp.752-757, 2004.
  - [4] Maxim Teslenko, Elena Dubrova, "Hermes:LUT FPGA Technology Mapping Algorithm for Area Minimization with Optimal Depth," *IEEE International Conference on Computer Aided Design*, 2004.