社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2007－ＳＬＤＭ－132 (32)
2007／11／22

LUT型FPGA向けテクノロジ・マッピングにおける深さ制約下のLUT数削減手法

高田　大河†　　松永　裕介††

† 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
†† 九州大学 大学院システム情報科学研究院 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，深さ最小の制約下でLUT数が少ないLUTネットワークを生成するLUT型FPGA向けテクノロジ・マッピングにおける，LUT数削減のための後処理Cut Substitutionを提案する．深さ最小の制約下でLUT数最小なネットワークを得る問題はNP困難なクラスと同等かそれ以上に難しい問題と考えられ，効率の良い厳密アルゴリズムは見つかっていない．Cut Substitutionは，LUTネットワークの深さを保ち余分なLUTを取り除くことで，局所的な最適解を生成する．提案手法と既存手法の比較実験を行い，提案手法の優位性を確認した．
キーワード　EDA, FPGA, 論理合成，テクノロジ・マッピング

Area Recovery under Depth Constraint by Cut Substitution for Technology Mapping for LUT-based FPGAs

Taiga TAKATA† and Yusuke MATSUNAGA††

† Graduate School of Information Science and Electrical Engineering, Kyushu University
744 Motooka, Nishiku, Fukuoka 819-0395 JAPAN
†† Faculty of Information Science and Electrical Engineering, Kyushu University
744 Motooka, Nishiku, Fukuoka 819-0395 JAPAN
E-mail: †{taiga,matsunaga}@c.csce.kyushu-u.ac.jp

Abstract　In this paper we present the post-processing algorithm, Cut Substitution, for technology mapping for LUT-based FPGAs to minimize the area under depth minimum constraint. The problem to generate a LUT network whose area is minimum under depth minimum costraint seems to be very difficult. Cut Substitution is the process to generate a local optimum solution by eliminating redundant LUTs while the depth of LUT network is maintained. The experiments shows that the proposed method derives LUT networks whose number of LUTs are smaller than the number of LUTs of network which are deribed by the existing algorithms.
Key words　EDA, FPGA, Logic Synthesis, Technology Mapping

1. はじめに

少量生産のLSIを低コストで実現するための手法の1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とは，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デバイスである．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．

遅延最小な回路を実現するために重要な回路合成の要素技術として，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，サブジェクト・グラフ（論理関数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処理である．LUT

とは，ある入力数以下の任意の論理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・マッピングにおいて深さが最小で LUT 数が少ないネットワークを生成することが望ましい．深さとは，ネットワークにおける最長パスの長さである．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．LUT 型 FPGA においては配線部分の遅延が支配的なため，個々の LUT 内部の遅延を一定であると見なす．また，テクノロジ・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，テクノロジ・マッピング後の処理が理想的と仮定して各々の LUT 間の配線に一定の固有遅延を仮定する．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．さらに，配置配線の自由度を高めることを狙い，同じ深さにおいては LUT 数が少ないネットワークを生成することを目的とする．

テクノロジ・マッピングにおいて LUT 数最小なネットワークを生成する問題は NP 困難のクラスに属し [3]，深さ最小の制約下で LUT 数最小なネットワークを生成する問題も難しいと考えられる．従来，深さ最小なネットワークの生成を保証するアルゴリズム (FlowMap [1], FlowMap-r [2], Tmap [6], Hermes [5], DAOmap [4], Imap [9]) が提案されてきたが，いずれも LUT 数に関しては最小を保証していない．

本稿では，LUT 型 FPGA 向けテクノロジ・マッピングの後処理として，LUT からなるネットワークを入力として深さを保つ制約下で LUT 数を削減するヒューリスティックなアルゴリズム Cut Substitution を提案する．Cut Substitution の考え方は，LUT からなるネットワークの局所的な変換によって，余分な LUT を直接取り除くというものである．既存研究において主流となっているアルゴリズムは，LUT の候補にあたる K フィージブル・カット（本稿では単にカットと呼ぶ）の選択に基づくものであり，選択したカットの数が LUT からなるネットワークの LUT 数にあたる．Cut Substitution は，LUT からなるネットワークにおける余分な LUT に対応するカット（余分なカット）を特定し，余分なカットを必要とするカットを余分なカットを必要としないカットに置き換えて選択することによって，余分なカットを選択から外す．カットの置き換えは LUT からなるネットワークの深さが増大しない範囲で行うため，LUT からなるネットワークの深さは保たれる．

実験において，Cut Substitution が単純なテクノロジ・マッピングのアルゴリズム Ddmap によって生成されたネットワークの LUT 数を平均で 8%削減することを確認した．Ddmap と Cut Substitution の組み合わせは，既存手法 DAOmap と比較して深さが等しく，LUT 数が平均で 8%少ないネットワークを生成した．Ddmap と Cut Substitution の組み合わせは，実行時間において DAOmap とほぼ同等であった．

本稿は，以下の構成からなる．二章において，準備としていくつかの定義を行う．三章において，k フィージブル・カットの選択に基づくテクノロジ・マッピングの主なアルゴリズムの流れを示す．四章において，Cut Substitution を説明する．五章において実験結果を示し，六章において本稿をまとめる．

2. 準　　備

テクノロジ・マッピングにおいて，与えられる入力はサブジェクト・グラフと呼ばれるグラフであり，出力は LUT ネットワークと呼ばれるグラフである．サブジェクト・グラフと LUT ネットワークは，各ノードが k 入力以下の DAG(Directed Acyclic Graph) である．k は LUT の入力数制約であり，デバイスのアーキテクチャに依存する値である．サブジェクト・グラフの各ノードはプリミティブな論理関数を表しており，LUT ネットワークの各ノードは k 入力以下の LUT を表している．サブジェクト・グラフや LUT ネットワークにおいて，ノード v_i がノード v_j の入力である場合に有向辺 (v_i, v_j) が存在する．

DAG $N(V,E)$ において，$v \in V$ のファンインとは v の入力辺に接続するノードを指す．$v \in V$ のファンインの集合は $fanin(v) = \{u \in V | \exists(u,v) \in E\}$ として定義される．また同様に，v のファンアウトは v の出力辺に接続するノードを指し，v のファンアウトの集合は $fanout(v) = \{w \in V | \exists(v,w) \in E\}$ で定義される．$fanin(v) = \phi$ であるような $v \in V$ をプライマリ・インプット (PI)，$fanout(v) = \phi$ であるような $v \in V$ をプライマリ・アウトプット (PO) と呼ぶ．DAG の深さは，全ての PI から PO までのパスのうち，最長パスの長さである．$v \in V$ の推移的ファンイン (TFI) とは，全ての PI から v までのパスに含まれるノードの集合である．より正確には，推移的ファンイン $TFI(v)$ は以下の式で定義される．

$$TFI(v) = fanin(v) \cup \bigcup_{u \in fanin(v)} TFI(u)$$

$v \in V$ の推移的ファンイン・グラフとは，$TFI(v)$ によって導かれる N の誘導部分グラフのことであり，$TFIG(v)$ と表す．

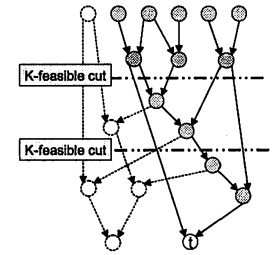

図 1　k-feasible cut (example if k=3)

$v \in V$ のカットとは，$TFI(v)$ の分割のことであり，$(X_v, \bar{X_v})$ と表される．$X \subseteqq V, \bar{X} \subseteqq V, X \cap \bar{X} = \phi$ である．ただしカット $(X_v, \bar{X_v})$ は単純なノードの分割でなく，$\forall pi \in TFI(v), pi \in X, v \in \bar{X_v}$ を満たしていなければならない．pi は，PI を表す．また，v を $(X_v, \bar{X_v})$ の根と呼ぶ．カット $(X_v, \bar{X_v})$ に対し，カット・セット $input((X_v, \bar{X_v}))$ を以下のように定義する．

$$input((X_v, \bar{X_v})) = \{v_i | v_i \in X_v, \exists v_j \in \bar{X_v}, \exists(v_i, v_j) \in E\}$$

以下の条件を満たすカット $(X_v, \bar{X_v})$ を，K フィージブル・カットと呼ぶ．まず，$\bar{X_v}$ によって誘導される部分グラフが連結である

こと．また，$\bar{X_v}$ に含まれるノードのファンアウトに X_v のノードが含まれないこと，すなわち $\forall v_i \in \bar{X_v}, \forall v_j \in X_v, (i,j) \notin E$．さらに，カット・セットの要素数が k 以下であること，すなわち $|\ input((X_v, \bar{X_v}))\ | \leqq k$．図1に，$k=3$ の場合の t を根とするKフィージブル・カットの例を示す．本稿の以下においては，Kフィージブル・カットでないカットは取り扱わないため，Kフィージブル・カットを単にカットと呼ぶ．

カット $(X_v, \bar{X_v})$ の $\bar{X_v}$ により誘導される部分グラフを，K フィージブル・コーン (KFC) と呼び $kfc_{(X_v,\bar{X_v})}$ で表す．$kfc_{(X_v,\bar{X_v})}$ に対し，v を $kfc_{(X_v,\bar{X_v})}$ の根と呼び，$input(X_v, \bar{X_v})$ を $kfc_{(X_v,\bar{X_v})}$ の葉と呼ぶ．kfc の葉を $input(kfc)$ で表す．サブジェクト・グラフにおける KFC kfc は，$input(kfc)$ を入力とする論理的に等価な k 入力 LUT で実現することができる．カット $(X_v, \bar{X_v})$ と $kfc_{(X_v,\bar{X_v})}$ は一対一対応であるため，$(X_v, \bar{X_v})$ によって生成される LUT は一意に決まる．$(X_v, \bar{X_v})$ に対応する LUT を $LUT_{(X_v,\bar{X_v})}$ と表す．

LUT 型 FPGA 向けテクノロジ・マッピングの問題は，与えられたサブジェクト・グラフを KFC で被覆する問題であると考えることができる．つまり，テクノロジ・マッピングの問題は，与えられたサブジェクト・グラフにおける全てのカットの集合 C から，サブジェクト・グラフの全てのノードを $\bar{X_v}$ で被覆するような C の部分集合を求める問題であると見なすことができる．C の部分集合 C' が LUT ネットワークとして実現可能であるためには，C' におけるカットのカット・セットが C' における他のカットの根であるか PI でなければならない．すなわち，以下の条件を満たしていなければならない．

$$\forall (X_v, \bar{X_v}) \in C', \forall v_j \in input((Xv, \bar{X_v})),$$
$$\exists (X_{v_j}, \bar{X_{v_j}}) \in C' \ or \ v_j \in PI$$

上記の条件を満たすカットの集合を，フィージブルであるという．

テクノロジ・マッピングにおいて深さ最小の制約下で LUT 数最小な LUT ネットワークを得るる問題は，LUT ネットワークの深さ最小を保証するフィージブルなカットの集合のうち，要素数が最小のものを求める問題であると見なすことができる．

3. カットの選択に基づくテクノロジ・マッピング

深さ最小の制約下で LUT 数最小な LUT ネットワークを得るテクノロジ・マッピング問題に対する既存のヒューリスティックなアルゴリズムは，主に次の3つの処理段階からなる．すなわち，カット列挙，コスト計算，カバリングである．カット列挙 [4] はサブジェクト・グラフにおけるカットを全列挙する処理である．カット列挙は LUT ネットワークの質に影響しないため，本稿ではカット列挙の説明を省略する．

コスト計算は，各ノードにおけるカットの深さや LUT 数に関するコストを計算する [4], [5], [7]．サブジェクト・グラフを $N(V,E)$ とする．$v \in V$ を根とするカット c_v の深さに関するコストとは，推移的ファンイン・グラフ $TFIG(v)$ において，c_v が解の一部として選択される場合の最小深さである．またカット c_v の LUT 数に関するコストとは，推移的ファンイン・グラフ $TFIG(v)$ において，c_v が解の一部として選択される場合の最小 LUT 数である．正確な最小 LUT 数を現実的な計算時間で求める方法が見つかっていないため，既存手法は LUT 数に関するコストとして，ヒューリスティックな手法による見積もり値を用いている．

カバリングは，LUT として実現するカットを選択する処理である．サブジェクト・グラフのノードに対し，0 または 1 つのカットを選択する．カットの選択は，サブジェクト・グラフの PO から PI へトポロジカル・オーダーで行われる．ノード $v \in V$ におけるカットの選択は，次のように行うのが一般的である．v までの処理において選択したカットの集合を C' とする．v が C' の全ての要素 c に対して $v \notin input(c)$ であった場合，何も行わない．v が C' のいずれかの要素 c に対して $v \in input(c)$ であった場合，v における Min コストなカット c'_v を C' へ追加する．v における Min コストなカットとは，v を根とするカットの中で深さに関するコストが v において要求される深さ以下であるもののうち，LUT 数に関するコストが最小のものである．上記の処理は，PO から処理を開始し，カット c の選択を行い，以下 $v \in input(c)$ で再帰的にカット選択の処理を行う，ということを全ての PO に対して行うことに等しい．

カバリングの処理は本質的に PO の TFIG ごとに独立に行われるため，余分なカットが選択される場合がある．すなわちカバリングにおいては，PO の集合を PO として，$po \in PO$ に対し，ノード $v \in V(TFIG(po))$ を根とするカットの選択を行う際に，$w \notin V(TFIG(po))$ で選択するカットと入力を共有することを考慮していない．図2に，余分なカットの例を示す．図2において，カット c は kfc_c を意味する台形で表されている．実線の台形は $TFIG(t)$ を被覆するためのカット選択によるものであり，破線の台形は $TFIG(y)$ を被覆するためのカット選択によるものである．t において c_t が Min コストのカットである場合，c_t が選択された後に u, v, w を根とするカットがそれぞれ選択される．t においてもし $\{u, x, w\}$ をカット・セットとするカットを選択したなら，x を根とするカットは $TFIG(y)$ を被覆するために選択が確定している一方で，v を根とするカットが必要なくなり，結果として選択するカットの総数が減ることになる．従来手法における多くのカバリングは，この点を考慮していない．

カバリング後に LUT 数を減らそうとするアルゴリズムはいくつか存在する [2], [4]．DAOmap [4] は，カバリング後に各カットの LUT 数に関するコストを修正し，カバリングをやり直す．LUT 数に関するコストの修正は，選択する複数のカット間でカット・セットのノードを共有することを狙って行われる．

4. Cut Substitution

Cut Substitution はテクノロジ・マッピングの後処理であり，LUT ネットワークを入力として深さを保つ制約下で LUT 数を削減するヒューリスティックな処理である．Cut Substitution は，テクノロジ・マッピングのカバリングにおいて選択されたカットの集合から，余分なカットを直接取り除く．

$S(V,E)$ をサブジェクト・グラフとし，C を $S(V,E)$ における

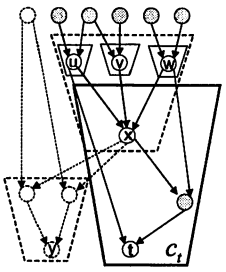

図 2 An excessive cut c_v is selected if c_t is selected(K=3)

全てのカットの集合とする．また，$C' \subseteqq C$ をテクノロジ・マッピングのカバリングにおいて選択されたカットの集合とする．Cut Substitution の入力は，$S(V,E), C, C'$ である．あるカットの集合 Cut から導かれる LUT ネットワークを，$N(V_{Cut}, E_{Cut})$ と表す．$lut \in V_{Cut}$ に対し，段数 $lev(lut)$，要求段数 $req(lut)$ を定義する．段数 $lev(lut)$ とは全ての PI から lut までのパスの最長長さであり，以下のように求められる．

$$lev(lut) = \max_{lut_i \in fanin(lut)} lev(lut_i) + 1$$

PI の段数は 0 である．PI でないノードの段数は $N(V_{Cut}, E_{Cut})$ における PI から PO へのトポロジカル・オーダーで求めることができる．また，要求段数 $req(lut)$ とは LUT ネットワークの深さを保つために lut に対応するカットの根で要求される段数のことであり，以下のように求められる．

$$req(lut) = \min_{lut_i \in fanout(lut)} req(lut_i) - 1$$

PO に対する要求段数は，与えられた LUT ネットワークの深さである．PO でない各ノードの要求段数は，$N(V_{Cut}, E_{Cut})$ において PO から PI へのトポロジカルな順序で求まる．

```
Cut Substitution( S(V,E), C, C' ){
  while (1) {
      C_nosy := Nosy_Cut_Enumeration(C', C);
      if C_nosy is empty {
          break;
      }
      c^best = Choice_Best_Cut( C_nosy );
      N(V_C', E_C')
        := Cut_Elimination( c^best, N(V_C', E_C') );
  }
  return N(V_C', E_C');
}
```

図 3 Pseudo code for Cut Substitution

図 3 に，Cut Substitution の擬似コードを示す．図 3 において，大文字の C はカットの集合，小文字の c はカットを表す．Cut Substitution は反復処理からなり，各反復は 3 つの処理からなる．すなわち，Nosy Cut Enumeration，Choise of Best Cut，Cut Elimination である．始めに，Nosy Cut Enumeration において，選択されたカットのうち余分なカットを全て特定する．$c' \in C'$ かつ $v \in input(c')$ であるような c' を $c_v \in C'$ の Fo-Cut と呼んだとき，余分なカット c_v とは，c_v の全ての Fo-Cut を v をカット・セットとしない他のカットで置き換えることが可能なカットのことである．次に，余分なカットの集合 C_{nosy} から，カットを取り除いたときに減る LUT 数に関するヒューリスティックな指標を用いて最良のカット c^{best} を選択する．次に，c^{best} の全ての Fo-Cut を他のカットに置き換え，c^{best} を選択したカットの集合から削除する．この反復処理は，余分なカットがなくなるまで実行される．

4.1 Nosy Cut Enumeration

Nosy Cut Enumeration において，C' における余分なカットが全て列挙される．余分なカット $c_v \in C'$ は，$v \in input(c_w)$ であるようなカット $c_w \in C'$ を全て，$v \notin input(c_w)$ であるようなカット $c'_w \notin C'$ で置き換えることが可能な c_v のことである．カット c_w の置き換えは，根が同じカット c'_w で行うことに注意する．上記の c_w を c_v の Fo-Cut と呼び，c'_w を c_w の Sub-Cut と呼ぶ．

カット c_w は，次の 2 つの条件が満たされる場合にのみ c'_w で置き換えることが可能である．C' から c_w を取り除いて c'_w を加えたカットの集合を C'' とする．最初の条件は，C'' のフィージブル性である．PI の集合を PIs と表すと，C'' がフィージブルであるためには以下の条件が満たされなければならない．

$$\forall v_i \in input(c'_w), v_i \in PIs \ or \ \exists c_{v_i} \in C''$$

もう一方の条件は，$N(V_{C''}, E_{C''})$ が深さ制約を満たすことである．$N(V_{C''}, E_{C''})$ が深さ制約を満たすためには，以下の条件が満たされなければならない．

$$\forall v_i \in input(c'_w), \exists c_{v_i} \in C'', \ lev(LUT_{c_{v_i}}) + 1 \leqq req(LUT_{c'_w})$$

Nosy Cut Enumeration において，上記の条件を調べることによって全ての余分なカットを列挙する．余分なカットおよび余分なカットに対応する Fo-Cut の集合，Sub-Cut の集合を保持する．

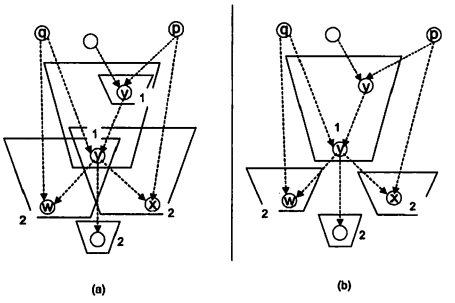

図 4 An example of an elimination of nosy cut c_v (K=3)

図 4 に，余分なカットの例を示す．図において，台形は C' に含まれるカットを表している．見易さのため，サブジェクト・グラフの辺は破線で示している．ノード i の隣の数字はカット c_i に対応する LUT LUT_{c_i} の段数を表している．図 4(a) にお

いて，ノード v，y は複製されている．c_v が余分なカットであることを確認する．$\{y,p\}$ をカット・セットとするようなカット c'_x が存在する．$y \in input(c'_x)$ に対しては $c_y \in C'$ が存在し，$p \in input(c'_x)$ に対しては p が PI なので，C' の c_x を c'_x で置き換えた C'' はフィージブルである．$lev(LUT_{c_y}) = 1$ であるため，C'' に関する深さの制約もまた満たされる．よって c_x は c'_x で置き換え可能である．カット c_w もまた同様に $\{y,q\}$ をカット・セットとするカット c'_w で置き換え可能である．全ての Fo-Cut $\{c_x, c_w\}$ が，選択したカットのフィージブル性や深さの制約を満たす範囲で他のカットによって置き換え可能であるので，c_v は余分なカットである．

4.2 Choise of Best Cut

Choise of Best Cut において，余分なカットのうち削除した場合に最も LUT 数削減につながると予想されるものを計算する．

カット $c \in C'$ に対し，ゲインを定義する．カット c のゲイン $g(c)$ は，以下の式に従い計算する．

$$g(c) = \sum_{v_i \in input(c)} g'(c_{v_i}) + 1 \quad (1)$$

$$g'(c) = \begin{cases} g(c) & (c\ is\ fanout\ free\ cut) \\ 0 & (c\ is\ not\ fanout\ free\ cut) \end{cases} \quad (2)$$

上記の式において，fanout free cut とは $N(V_{C'}, E_{C'})$ において $|fanout(LUT_c)| = 1$ であるような $c \in C'$ を指す．カット $c \in C'$ に対して $v_i \in input(c)$ を根とする fanout free cut $c_{v_i} \in C'$ が存在した場合，$fanout(LUT_{c_{v_i}}) = \{LUT_c\}$ であるため，c が C' から取り除かれた場合は c_{v_i} も取り除くことが可能である．

式 (1)，(2) に従い，$N(V_{C'}, E_{C'})$ の PI から PO へトポロジカルな順序で C' 各要素のゲインを求める．Choise of Best Cut は，余分なカットの集合におけるゲインが最大のカットを返す．

4.3 Cut Elimination

Cut Elimination において，カットの置き換えを行いゲインが最大のカットを C' から取り除く．具体的には，ゲインが最大のカット c に対する全ての Fo-Cut を C' から取り除いて，C' に Sub-Cut を加える．さらに，c を C' から取り除き，入力辺に接続する fanout free cut を再帰的に C' から取り除く．生成した新たな C' に従い，LUT ネットワーク $N(V_{C'}, E_{C'})$ を生成する．$V_{C'}$ における各ノードの段数，要求段数を計算する．

図 4(b) に，図 4(a) の余分なカット c_v を取り除いた LUT ネットワークを示す．余分なカット c_v および Fo-Cut $\{c_x, c_w\}$ が C' から取り除かれ，Sub-Cut $\{c'_x, c'_w\}$ が C' に加えられる．

5. 実　　験

Cut Substitution のアルゴリズムを，九州大学の論理合成システム Magus にプログラムとして組み込んだ．用いた言語は C++である．実験を行い，提案手法と既存手法 DAOmap の比較を行った．用いたマシンの CPU は IntelXeon 3.0GHz，メインメモリは 15GB である．

Cut Substitution は Ddmap の後処理として組み込まれた．Ddmap は LUT 型 FPGA 向けテクノロジ・マッピングの単純なアルゴリズムであり，[10] における近似アルゴリズムの応用である．Ddmap は深さが最小な LUT ネットワークを生成するが，LUT 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アルゴリズムである．多くの伝統的なアルゴリズムは，LUT 数に関するコストをファンアウト数で割るというヒューリスティックを用いてきた．このヒューリスティックは，分岐したパスが再収斂した際に LUT 数に関するコストを重複して足しあわせてしまうことを避けるものである．しかしこのヒューリスティックは，LUT 内部におけるファンアウトを無視している．すなわち，$kfc(v)$ が LUTl によって被覆されているとき，TFIG(v) の LUT 数に関するコストは v のファンアウト数で割られる．$w \in kfc(v)$ から $x \notin TFI(v)$ へのファンアウトが存在し，かつ w, x を含むパスと w, v を含むパスが再収斂した場合に，LUT 数に関するコストの一部が重複して数えられる場合があるが，単純に v のファンアウト数で割るヒューリスティックはこれを無視している．LUT 内部におけるファンアウトを考慮するために，Ddmap はファンアウト数によってサブジェクト・グラフの全ての辺に重みをつけ，パスに含まれる辺の重みと LUT 数のコストを掛け合せるヒューリスティック [10] を用いている．

表 1　Comparison with Cut Substitution and DAOmap on MCNC benchmarks (k = 5)

bench	DaoMap		Dd	DdMap + CS			
marks	LUT	time	LUT	LUT	time	vs. Dd	vs. Da
C1908	150	0.16	159	143	0.2	90%	95%
C3540	368	0.28	366	325	0.35	89%	88%
C5315	438	0.51	448	425	0.56	95%	97%
C6288	560	2.70	894	631	3.70	71%	113%
C880	116	0.05	122	111	0.06	91%	96%
apex6	198	0.07	202	195	0.07	97%	98%
apex7	84	0.03	88	84	0.05	95%	100%
f51m	52	0.03	52	52	0.04	100%	100%
rot	385	0.15	379	359	0.16	95%	93%
vda	455	0.23	455	422	0.22	93%	93%
att10	935	0.57	932	882	0.43	95%	94%
att6	455	0.23	455	422	0.22	93%	93%
att8	619	0.22	598	598	0.17	100%	97%
C2670	205	0.27	208	190	0.25	91%	93%
des	2079	1.90	2140	1971	1.50	92%	95%
C7552	634	1.11	662	608	1.23	92%	96%
Total	7733	8.50	8160	7418	9.26	91%	96%

表 1，表 2 に実験結果を示す．表 1 は MCNC ベンチマークセット，表 2 は ITC'99 ベンチマークセットの各ベンチマークに対し，DAOmap，Ddmap，および提案手法（Ddmap と Cut Substitution の組み合わせ）が生成した LUT ネットワークの LUT 数の比較である．本実験における LUT の入力数制約は 5 である．MCNC ベンチマークセット，ITC'99 ベンチマークセットのうち規模が小さいベンチマークは省略した．"Dd" は Ddmap，"Ddmap+CS" は提案手法を現している．"LUT" の

表 2 Comparison with Cut Substitution and DAOmap on ITC'99 benchmarks (k = 5)

bench	DaoMap		Dd	DdMap + CS			
marks	LUT	time	LUT	LUT	time	vs. Dd	vs. Da
b14	2265	3.67	2390	2097	3.74	88%	93%
b14_1	1749	2.43	1823	1632	2.53	90%	93%
b15	2944	3.08	3175	2695	3.59	85%	92%
b15_1	3571	5.52	3438	3145	3.92	91%	88%
b17	9457	12.4	10208	8724	11.5	85%	92%
b17_1	10747	16.4	10578	9678	12.2	91%	90%
b20	4932	8.18	4824	4275	8.33	89%	87%
b20_1	3437	5.46	3484	3085	5.86	89%	90%
b21	5069	8.32	5066	4498	8.47	89%	89%
b21_1	3553	5.55	3603	3217	6.03	89%	91%
b22	6860	11.7	7149	6306	12.1	88%	92%
b22_1	5084	8.12	5363	4728	9.07	88%	93%
Total	59668	90.8	61101	54080	87.3	89%	91%

列は生成した LUT ネットワークのノード数，"time" は秒単位の実行時間である．"vs Dd"，"vs Da" はそれぞれ提案手法と Ddmap，提案手法と DAOmap における生成した LUT ネットワークのノード数の比である．ベンチマークが同じであれば LUT ネットワークの深さはいずれも等しいため，LUT ネットワークの深さは省略した．

C6288 を除く全てのベンチマークにおいて，提案手法は DAOmap と LUT 数が同等かより少ない LUT ネットワークを生成した．生成した全ての LUT ネットワークの LUT 数において，提案手法は DAOmap よりも平均 9%少ない結果を得た．表 1 において，Ddmap は DAOmap よりも平均 6%ほど多くの LUT 数で LUT ネットワークを生成した．Cut Substitution は Ddmap が生成した LUT ネットワークの LUT を平均で 9%削減し，結果として DAOmap に比べて平均 4%少ない LUT で LUT ネットワークを生成した．図 2 において，Ddmap は DAOmap よりも平均 2%ほど多くの LUT 数で LUT ネットワークを生成した．Cut Substitution は Ddmap が生成した LUT ネットワークの LUT を平均 11%削減し，結果として DAOmap に比べて平均 9%少ない LUT 数で LUT ネットワークを生成した．提案手法の実行時間は，DAOmap に比べて平均 3%短かった．

6. 終わりに

本稿において，LUT 型 FPGA 向けテクノロジ・マッピングにおいて深さ最小の制約下で LUT 数を削減するための後処理 Cut Substitution を提案した．Cut Substitution の考え方は，LUT からなるネットワークの局所的な変換の反復によって，余分な LUT を直接取り除くというものである．Cut Substitution は LUT ネットワークの生成方法に依存しない処理であり，テクノロジ・マッピングの全てのヒューリスティックなアルゴリズムの後処理として実装することが可能である．実験の結果，Cut Substitution は深さ最小な LUT ネットワークを生成する単純なヒューリスティック Ddmap が生成した LUT ネットワークの LUT を平均で 11%削減することを確認した．また，Ddmap と Cut Substitution の組み合わせが，既存手法 DAOmap と比較して平均で 9%少ない LUT 数で LUT ネットワークを生成することを確認した．Ddmap と Cut Substitution の組み合わせは，DAOmap と比較して平均で 3%短い時間で LUT ネットワークを生成した．

今後の課題の 1 つは，Cut Substitution が有効に働くような LUT ネットワークを調査することである．DAOmap など他のテクノロジ・マッピングのアルゴリズムが生成した LUT ネットワークに対し，後処理として Cut Substitution を適用する．また，Cut Substitution は LUT ネットワークの局所的な変換を行うものであるが，今後より大域的な変換を一度に行う後処理を検討する．前処理で生成した LUT ネットワークの LUT に対応するカットの根によって，同じ根を持つカットのみ使用する制約の下のテクノロジ・マッピングは，Cut Substitution の処理をサブジェクト・グラフの複数の領域へ同時に適用することと同等の処理である．また，深さ制約下で LUT 数最小な LUT ネットワークを生成する問題の厳密アルゴリズムを開発することも今後の課題である．厳密解の LUT 数は LUT 数の下限値と見なせるため，厳密アルゴリズムによって限られた規模のベンチマークに対する厳密解が求まれば，ヒューリスティックなアルゴリズムの性能評価に用いることができる．

謝　辞

本研究の一部は，科学研究費補助金 (基盤研究 (A))(課題番号: 19200004) による．

文　献

[1] J.Cong, Y.Ding, "FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 1994.

[2] J.Cong, Y.Ding, "On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping," IEEE Transactions on Very Large Scale Integration Systems, June, 1994.

[3] Richard L.Rudell, "Logic synthesis for VLSI design," Ph.D.thesis, University of California, Berkeley, 1989.

[4] D.Chen, J.Cong, "DAOmap: A Depth-optimal Area Optimization Mapping Algorithm for FPGA Designs," IEEE International Conference on Computer Aided Design, 2004.

[5] Maxim Teslenko, Elena Dubrova, "Hermes: LUT FPGA Technology Mapping Algorithm for Area Minimization with Optimal Depth," IEEE International Conference on Computer Aided Design, 2004.

[6] 高田 大河, 松永 裕介, "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," , 信学技報, 第 VLD2006-59(2006-11) 巻, 2006.

[7] Touati Herve Jacques, "Performance-oriented Technology Mapping," Ph.D.thesis, University of California, Berkeley, 1990.

[8] J.Cong and Y.HWang, "Simultaneous Depth and Area Minimization in LUT-Based FPGA Mapping," International Symposium on Field Programmable Gate Arrays, Feb. 1995.

[9] Valavan Manohararajah, Stephen D. Brown, Zvonko G. Vranesic, "Heuristics for Area Minimization in LUT-based FPGA Technology Mapping, " IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.25, No.11, November 2006.

[10] 松永 裕介, "DAG カバリング問題の下限とそれを用いた厳密アルゴリズムについて", 信学技報, 第 VLD2007-8 巻, 2007.