

## ゲートの仮想除去を用いた大規模回路向けトランスダクション法

1K-8

永井 裕 上林 彌彦

京都大学工学部

## 1 まえがき

近年, 論理回路の設計規模はますます大規模化し, 計算機支援による設計は欠かせなくなっている. 計算機支援による回路の最適化手法にトランスダクション法<sup>[4]</sup>があるが, この手法は許容関数という概念に基づき結線の追加・削除によって回路変換を行う.

一方, 設計回路の大規模化に伴って, 計算時間や記憶容量の問題のため, 目的回路から部分回路を抽出しその部分回路に対して最適化を行う手法が要求されている. 部分回路を抽出しトランスダクション法を適用する手法には, 関数のBDDサイズに基づいて部分回路を抽出する方法<sup>[1]</sup>や木状に部分回路を抽出する方法<sup>[2]</sup>がある.

本稿では, 木状に部分回路を抽出しトランスダクション法を適用する場合の問題点について述べる. またその問題点を解決するため, 回路変換の過程で除去されるゲートを仮想的なものとし, 一旦除去されたゲートを再利用することによって回路変形の可能性を高める手法について述べる.

## 2 基本的事項

## 2.1 許容関数

あるゲート(入力端子または結線)の実現する関数 $f$ を, 論理関数 $f'$ で置換えても回路の出力に変化がないとき, 関数 $f'$ をそのゲート(入力端子または結線)の許容関数(Permissible Function)という. また許容関数の集合を考えた時, 同時に置換え可能なものからなる集合をCSPF(Compatible Set of Permissible Functions)と呼び,  $0 \cdot 1 \cdot *$ (*don't care*)の3値をとる関数 $G$ として表す.

## 2.2 許容関数集合による回路変換

NORゲートのみで構成される組合せ回路を考えると, 回路中のある結線の許容関数集合に恒偽関数が含まれるとき, この結線は削除可能(Disconnectable)である. また, ある結線をゲート $v$ に接続した時に新たに $v$ の実現する関数が $v$ の許容関数集合に含まれる時, この接続によって回路の出力は変化しない. ゲート $v_j$ の出力をゲート $v_i$ の入力に接続可能(Connectable)である条件は次の通りである.(ただし $v_j$ は $v_i$ から出力側に辿っていった時到達可能でない.)

$$G^{on}(v_i) \cap f^{on}(v_j) = \phi \quad (1)$$

Transduction Method for Large Networks Using Virtual Elimination of Removable Gates  
Hiroshi NAGAI, Yahiko KAMBAYASHI  
Faculty of Engineering, Kyoto University

## 2.3 手続き Connectable/Disconnectable

手続き Connectable/Disconnectable (C/DC) の概要は次の通りである. まず各ゲートで実現される論理関数を計算する. 次に出力端子に近いゲートから, そのゲートへの入力結線のCSPFの計算(冗長結線の切断), 接続可能なゲートの接続, CSPFの再計算(冗長結線の切断)を行う. これを回路コスト(ゲート数, 結線数)の改良がなくなるまで繰り返す.

## 3 部分回路抽出によるトランスダクション法

本章では, 大規模回路から部分回路を抽出しトランスダクション法を適用する方法について述べる.

## 3.1 BDDサイズに基づく部分回路の抽出

文献<sup>[1]</sup>では, 各ゲートに対して実現される論理関数のBDDサイズがある閾値以内の部分回路を抽出する手法が示されている. この手法では閾値とするBDDサイズをうまく設定すると, 数千ゲート規模のベンチマーク回路に対して現実的な時間で最適化できることが示されている.

## 3.2 木状に部分回路を抽出する方法

またこの他に, あるゲートに着目しそのゲートへの入力回路を部分回路として最適化を行う手法<sup>[2]</sup>がある. この場合, 抽出した部分回路は木状になる.

木状に部分回路を抽出した場合は, 回路によっては結果が良いものもあったが, 多くの場合文献<sup>[1]</sup>の手法に比べて計算時間, 結果ともに良くなかった.

## 4 仮想的なゲートの除去

## 4.1 木状に抽出する手法の問題点

木状に抽出する手法が, BDDサイズに基づいて抽出する手法に比べて結果が良くない原因には, 図1の様に, 先に最適化されることによって除去されてしまうゲートから後の最適化ステップにおいて結線を接続することができないことが考えられる.

## 4.2 仮想的に除去されるゲート

上述の原因を解消するために, ファンアウトがなくなり除去されるゲートを仮想的なものとし, データ構造自体は保存しておく. 仮想的に除去することでC/DCの手続き中において接続の候補に考えることができ, 回路が変形される可能性が高まる. ただし, 仮想的に除去されたゲートから接続するときには無闇に接続するのではなく, 次のコストを評価する.

$$P = P_{rm} - P_{rv} \quad (2)$$

$P_{rm}$ : 接続によって除去されるゲート数

$P_{rv}$ : 接続によって復活するゲート数

表 1: 仮想除去を考慮した場合と考慮しない場合の比較

Circuit	仮想除去を考慮しない場合				仮想除去の場合			
	BDD サイズ抽出		木状抽出		BDD サイズ抽出		木状抽出	
	G/C/L	CPU	G/C/L	CPU	G/C/L	CPU	G/C/L	CPU
C1908	528/1093/32	13.1	534/1092/32	81.1	528/1093/32	16.4	533/1091/32	136
C2670	887/1709/25	26.6	894/1709/26	93.7	885/1706/25	42.7	885/1701/26	158
C6288	2324/4779/120	40.0	2333/4776/119	232	2324/4779/120	50.0	2321/4770/119	412
C7552	2397/4850/36	288	2432/4787/36	1086	2396/4844/36	407	2385/4776/36	1980
alu2	231/564/27	10.3	236/554/27	25.3	221/549/27	16.6	222/532/26	47.7
dalu	1050/2415/24	344	1124/2562/26	917	1047/2410/24	487	1098/2520/25	1643
des	4213/8849/20	671	4286/9064/20	1651	4213/8849/20	842	4283/9061/18	2966
frg2	804/1967/14	71.1	771/1838/15	492	801/1959/14	112	752/1807/15	767
i8	1673/4016/14	155	1643/4062/14	707	1673/4016/14	214	1624/4059/14	1270
rot	579/1145/25	16.9	589/1168/25	58.8	577/1144/25	29.2	584/1166/25	97.7
term1	162/349/10	8.4	158/345/10	21.5	153/331/10	11.7	138/292/10	28.2
too_large	563/1301/24	16.5	577/1348/24	89.0	561/1298/24	27.5	557/1310/24	147

(ゲート数 / 結線数 / 段数)

(sec.)

※ 太字は仮想除去の場合の両抽出法の比較を表す

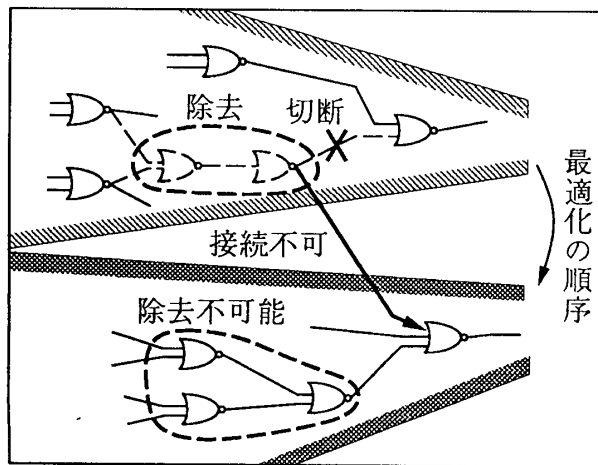


図 1: 木状に抽出する手法の問題点

このコスト  $P$  が正の場合のみ接続し、その他の場合は接続しない。

## 5 実験結果

前章で述べた手法を C 言語を用いて実装し SS5 上で実験を行った。プログラム中の SBDD 処理は、NTT の湊真一氏による SBDD パッケージ (最大ノード数 100 万) を使用した。また初期回路は、LGSynth'91 多段ベンチマーク回路をファンイン 4 までの NOR ゲートにマッピングしたものをを用いた。

ここでは BDD サイズに基づき部分回路を抽出する手法と木状に抽出する手法について、仮想的に除去する場合とそうでない場合をそれぞれ比較した結果の一部を表 1 に示す。表中の G/C/L の欄は最適化結果 (ゲート数 / 結線数 / 段数) を表し、CPU の欄は CPU 時間 (秒) を表す。また仮想除去の場合の両抽出法の結果を比較しその数の小さいものを太字で表す。

これによると、両抽出法について多くの回路で仮想除去を考慮した場合は考慮しない場合に比べてゲート数や結線数が減少している。特に C6288 や C7552 など仮想除去を考えない時に BDD サイズに基づいて抽出した方が結果が良かったものが、仮想除去により木状抽出の結果の方が良くなる場合があり、木状抽出の場合に対して特に仮想除去が有効であることが示されている。ただし、計算時間は木

状抽出の場合で仮想除去を考慮しない場合の 1.5 ~ 2 倍の時間がかかるため、この点の改良が今後の課題である。

## 6 あとがき

本稿では、トランスダクション法において最適化の途中で除去されるゲートを仮想的な除去とし、そのデータ構造を保存しておき、再びそのゲートから結線を接続する手法について述べた。また、ベンチマーク回路に対して適用した結果その有効性が示された。

この手法は木状に部分回路を抽出する場合だけでなく、BDD サイズに基づく場合や回路全体を対象とする場合についても同様に適用でき、回路が変形される可能性が高まると考えられる。

## 謝辞

本手法について有益な御示唆を頂いたイリノイ大学の室賀三郎教授、御討論下さった室賀研究室の皆様、SBDD パッケージの使用を快諾して頂いた矢鳥脩三教授並びに矢鳥研究室の皆様へ深謝する。

本研究は、新エネルギー・産業技術総合開発機構 (NEDO) の提案公募型・最先端分野研究開発のプロジェクトとしての支援を受けている。

## 参考文献

- [1] 石垣博康, 上林弥彦: “回路抽出による大規模回路へのトランスダクション法の適用”, 情報処理学会第 48 回全国大会, 5B-8, 1994.
- [2] 永井裕, 上林弥彦: “大規模回路向けトランスダクション法における部分回路の抽出”, 情報処理学会第 51 回全国大会, 7P-3, 1995.
- [3] S. Minato, N. Ishiura, S. Yajima: “Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation”, *Proc. 27th Design Automat. Conf.*, pp. 52-57, 1990.
- [4] S. Muroga, Y. Kambayashi, H. C. Lai, J. N. Culliney: “The Transduction Method - Design of Logic Networks Based on Permissible Functions”, *IEEE Transactions on Computers*, Vol. 38, No. 10, pp. 1404-1424, 1989.