中間変数を用いた大規模回路向けトランスダクション法

5B－2

永井 裕　上林弥彦

京都大学工学部

1　まえがき

VLSI技術の目覚しい進歩による回路の大規模化，複雑化に伴って，計算機支援による論理回路設計は非常に重要になっている．本稿で用いる回路の最適化手法であるトランスダクション法[3]は，許容関数と呼ばれる回路内の冗長性を表す概念を用いて回路変換を行うものであり，近年の計算機能力の飛躍的な向上と共有二分決定グラフ(SBDD)の開発[2]により，広く用いられるようになった．

しかし回路がますます大規模化するにつれ，記憶容量，計算時間の点から，回路全体ではなく部分回路に対して最適化を行う手法が求められてきている．

そこで本稿では，各ゲートの論理関数を表すBDDのノード数によって部分回路を抽出し，抽出した部分回路の出力を中間変数で表現することにより対象外となった回路の部分も考慮して許容関数を計算し，トランスダクション法を適用して大規模回路の最適化を行う手法を提案する．また，本手法による実験を行い，大規模な回路に対して比較的短時間に処理が可能で，文献[1]の手法に比べて良好な最適化結果が得られたことを示す．

2　基本的事項

本章では，本手法で用いるトランスダクション法の基本的な概念について述べる．ここでは論理回路を構成するゲートとして，NORゲートのみを扱っているが，一般化は容易であると考えられる．

2.1　許容関数

ある入力端子，ゲートまたは結線の実現する関数fを，論理関数f'で置換えても回路全体としての出力に変化がないとき，そのようなf'を入力端子，ゲートまたは結線の許容関数(Permissible Function)であるという．またそのような許容関数の集合の中で同時に置換え可能なものからなる部分集合をCSPF(Compatible Set of Permissible Functions)と呼ぶ．

許容関数集合は，入力に対して$0 \cdot 1 \cdot *(don't\ care)$の3値をとる関数として表すことができ，$n$個の外部変数$x_1 \sim x_n$に対するCSPFを$G(x_1, x_2..x_n)$と表すこととする．

2.2　許容関数集合による回路変換

ある回路中の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に接続した時の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，以下のように表すことができる．

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

$$v_i \notin TFO(v_j) \quad (2)$$

ここで$TFO(v_j)$は，ゲートv_jから出力端子側に向けて到達可能なゲート集合である．

2.3　手続き Connectable/Disconnectable

ここでは，上述の条件を用いた回路変換手法Connectable/Disconnectable(C/DC)のアルゴリズムの概要を示す．

step 1 各ゲートの出力関数を計算する．
step 2 各ゲートを出力端子に近い方から順序付けする．
step 3 step 3.1～3.4をファンイン制限付きで，step 2で求めた順序で適用する．
- **step 3.1** ゲートの入力結線のCSPFを求め，冗長な結線があれば切断する．
- **step 3.2** このゲートに接続可能なゲートを回路中から探索し，接続する．
- **step 3.3** このゲートの入力結線のCSPFを再計算し，冗長な結線を取り除く．
- **step 3.4** もしファンイン制限を超えていれば，step 3.1が終了した状態の入力結線に戻す．

step 4 step 1～3を回路コストの改良がなくなるまで繰り返す．

3　中間変数を用いた手法

大規模回路向けにBDDのノード数で対象とする回路を制限する手法[1]では，閾値をできるだけ大きくした方が得られる結果は良いが，計算時間や必要となるBDDノード数は膨大になる．本章では，手続きC/DCの対象となる回路は比較的小さくし，中間変数を用いて対象外となった回路の部分も考慮してCSPFの計算する手法を示す．

3.1　BDDノード数による部分回路の抽出

文献[1]では，手続きC/DCのstep1で各ゲートの論理関数を計算していく際に，BDDのノード数がある閾値を越えると，そのゲート及びTFOに含まれるゲートの論理関数は計算せず，論理関数が計算されている部分回路に対してstep2以降を適用する．

3.2　中間変数を用いた手法

まず，ゲートv_kの論理関数を表すBDDのノード数がある閾値Tを越えた時，ゲートv_kの入力結線の始点となっているゲート$v_{j_1}, v_{j_2}, \cdots, v_{j_n}$を境界ゲートと呼ぶ．

BDDノード数によって部分回路を抽出する手法では，この境界ゲートを回路の出力ゲートとみなし，それらの論

Transduction Method for Large Networks Using Intermediate Variables
Hiroshi NAGAI　Yahiko KAMBAYASHI
Faculty of Engineering, Kyoto University

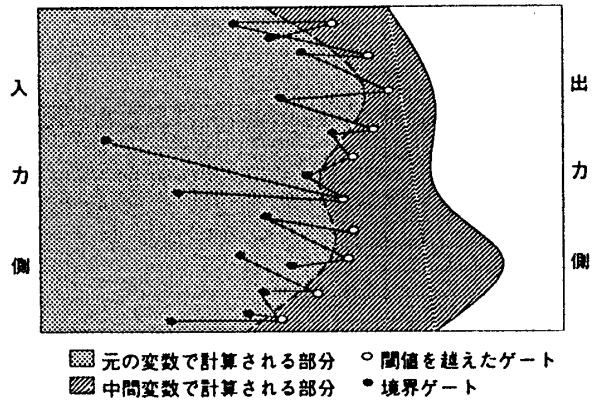

図 1: 中間変数で表される部分回路

理関数を変えずに最適化を行うものであった．しかし，目的回路の出力が変化しないように境界ゲートの出力を変更することが可能である．

そこで，境界ゲートまでの論理関数を計算した後，このような境界ゲートの出力を新たに中間変数で置換えて，対象外となっている元の回路中の各ゲートの論理関数を計算する．ここでも BDD ノード数によって計算範囲の制限を行う．次に，それらのゲートの CSPF を計算していく．これらの論理関数，CSPF は中間変数で表されている．BDD ノードの消費を少なくするため，これらの論理関数，CSPF の計算は一度しか行わず，境界ゲートではそれらの BDD は保持し続け，境界ゲート以外ではすべて解放する．手続き C/DC の step3 で境界ゲートより前段のゲートの CSPF を計算するために，まず，境界ゲートの CSPF を元の変数で置換えていく．その後，境界ゲートより前段の CSPF を計算する．以後，最適化を進める際には，まず境界ゲートまでの各ゲートの論理関数を計算し，次に境界ゲートの CSPF を元の変数で表し，それより前段の CSPF を計算する．

3.3 問題点とその対処

上述の手法にはいくつかの問題点がある．

1) 境界ゲートの CSPF の変数の置換えに時間がかかる．
2) 置換えによって CSPF を表す BDD のノード数が膨大になる．

1 については，置換えを行った後，*don't care* の含まれる割合がある一定値より小さい時には，以降はそのゲートの論理関数に置換える．2 については，文献[1] と同様に閾値を設け，その値を越えた場合には，CSPF の代わりに実現されている論理関数で代用して計算を続ける．

4 実験結果

上述した手法に基づいて，トランスダクション法を用いた論理回路最適化プログラムを C 言語を用いて作成し，SS10 上で実験を行なった．この時生成される回路のファンインは 4 に制限した．

初期回路としては，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用いた．なお本プログラムの SBDD 処理は，NTT の湊真一氏による SBDD パッケージを使用している．

表 1: 実験結果

	文献[1] の手法		本手法	
回路	CPU	Gate/Conn/Lev	CPU	Gate/Conn/Lev
C1355	9.7	575/1065/25	621m	575/1065/25
C1908	37.6	536/1112/32	96.9	537/1117/32
C3540	177.9	1081/2327/40	248.1	1081/2316/40
C432	7.8	155/331/21	1412	153/329/21
C499	9.0	515/993/22	9099	515/993/22
C5315	2224	1557/3462/40	3468	1567/3467/40
C6288	184.5	2302/4765/119	195.6	2302/4765/119
C7552	3915	2413/4869/35	-	-/-/-
C880	6.5	375/750/22	9.8	374/751/22
alu2	31.9	234/569/28	46.8	202/503/24
alu4	118.6	508/1209/33	154.5	498/1191/33
apex6	296.9	771/1508/19	293.3	764/1577/19
dalu	543.4	1093/2483/26	319.3	950/2162/22
des	4348	4215/8852/20	9098	4217/8855/20
frg2	237.0	751/1841/14	303.3	746/1821/12
i10	2305	2214/4692/44	486m	2211/4623/44
i8	220.3	1596/3925/14	1882	1591/3914/14
pair	1105	1451/3078/20	1474	1446/3048/20
t481	1112	1402/3305/20	920.7	1286/3031/20

表 1に本手法および文献[1] の手法による最適化結果を示す．どちらも BDD ノードの閾値を 100 とし，本手法の中間変数で計算される論理関数についても BDD ノード数 100 を閾値とした．CPU 時間の単位は秒で表されているが，m が付加されているものの単位は分である．結果はゲート数 / 結線数 / 段数 で表されている．これにより，本手法は全体的に時間はややかかったものの最適化結果は良好で，特に，回路全体の最適化が困難である dalu は，本手法ではゲート数が初期回路の 50%(文献[1] の手法では 57%) になることが示された．

5 あとがき

本稿では，各ゲートの論理関数を表す BDD のノード数によって抽出した部分回路に対して，中間変数を用いて対象外となった回路の部分も考慮して許容関数を計算し，大規模回路の最適化を行う手法を提案した．また，本手法による実験を行い，大規模な回路に対して短時間で比較的良好な最適化結果が得られたことを示した．

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，SBDD パッケージの使用を快諾して頂いた矢島研究室の皆様に深謝する．なお本研究は文部省科学研究費 (国際学術研究) によるものである．

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] 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.

[3] S.Muroga, Y.Kambayashi, H.C. Lai, J. Niel, Culliney: “The Transduction Method - Design of Logic Networks Based on Permissible Functions”, *IEEE Transactions on Computers*, Vol.38, No.10, pp.1404-1424, 1989.