

トランスダクション法を用いた回路の置き換えによる論理回路遅延最適化について  
6K-1

石垣博康 澤田直 上林弥彦

京大工学部 九大工学部 京大工学部

## 1 まえがき

トランスダクション法[1]は、許容関数という概念に基づき回路内の潜在的なドントケアを有効に利用して形状変換と冗長部分の削除を行なう手法であり、回路の単純化や遅延最適化に有効であることが知られている[?]。しかし回路が大きくなるにつれて必要となる記憶容量と計算時間が増大するため大規模な回路全体に対して適用することは困難である。

一方、近年のVLSI技術の発展に伴い、大規模回路に対する設計自動化の必要性が高まってきている。その際にはさまざまな制約が加えられるが、特に遅延に関する制約の要求が最近大きくなりつつある。

そこで本稿では許容関数の性質を用いて回路中の複数のゲートとそれに付随する部分回路を同時に別の段数の少ない回路に置換することにより、回路の段数削減をはかる手法について述べる。

## 2 用語と記号

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  $f'$  で置き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き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  $0 \cdot 1 \cdot *$  の3値をとるような関数として表すことができ、 $n$  個の外部変数  $x_0 \sim x_n$  に対する CSPF を  $G(x_0, x_1, \dots, x_n)$  で表すこととする。本稿では、許容関数  $G$  を表現するために共有二分決定グラフ(SBDD)[3][4]を用いているが、その3値を表すために一つの許容関数に対して二つのSBDDを使用している。

許容関数は出力段から入力段に向けて計算されていくが、入力段に近付くにつれてドントケアが増えていくという性質を持っている。

## 3 許容関数を用いた部分回路置換

本章では CSPF を用いた、回路の置換による段数削減手法を提案する。本手法は、(1) 置換されるゲート集合の選択、(2) 置換するための部分回路の生成と置換、の2つの部分から構成される。3.1節ではその部分回路の設計手法、3.2節では置換されるゲート集合の選択方法、3.3節では置換方法について述べる。

## 3.1 部分回路設計手法

本稿では部分回路を生成するために、文献[2]の手法を使用する。この手法は束(lattice)を用いて NOR ゲートからなる3段回路を生成し、それに対して段数をできるだけ増加させないようにファンイン制限を加えていくというものである。この手法はドントケアを含む多出力回路を少ない段数で設計するという意味で本稿の目的に適しているが、束をもとにしているのであまり多くの外部変数を必要とする関数には効率が悪くなるという欠点が存在する。そのため必要となる変数の数はなるべく少ないことが望まれる。

## 3.2 置換されるゲート集合の選択

置換されるゲート集合は段数削減のため最長経路上のあるレベルのゲートと、また他の部分でそれと同レベルのゲートの一部が選ばれる。選択される最長経路上のレベルは、そこに存在するゲート各々に必要な外部変数の数によって決定される。ここであるゲートに必要な外部変数は、次のようにして決めることができる。

もし変数  $x_i$  について次式が空でなかった場合、変数  $x_i$  は CSPF を表現するのに不要であるのでこれにより取り除き、関数  $G$  のかわりに縮退された  $G'$  を用いることとする。これを繰り返し適用することにより必要な外部変数を減らすことができる。

$$G'(x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = G(x_0, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \cap G(x_0, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

Delay Optimization by Network Substitution with Transduction Method

Hiroyasu ISHIGAKI, Sunao SAWADA and Yahiko KAMBAYASHI  
Kyoto Univ. Kyushu Univ. Kyoto Univ.

このようにして得られた外部変数の数が小さければ、生成される置換用の回路も小さくなり易いので、最長経路上のあるレベルのゲート各々に必要な外部変数の数がすべてある決められた値よりも小さかった場合にそのレベルを置換の対象とすることとする。またその際、最長経路以外の部分で同レベルのゲートに対しても必要な外部変数の数を求め、それが上に述べた値よりも小さかった場合、同様にそれらも置換対象とする。また目的ゲートの計算は出力段より行なうこととする。

### 3.3 部分回路の生成と置換

目的となるゲート集合が決まれば、その縮退された CSPF に対して部分回路を生成する。部分回路は、上述したように束を用いて3段 NOR 回路を生成しそれに対しファンイン制限を加えたものを用いる。この回路の段数が置換されるゲート集合のレベルよりも小さかった場合に置換を行なう。置換は生成された部分回路のゲート各々に対し次の手続きを用いておこなう。

step1 元の回路のゲートに同じ論理関数があるか調べる。  
あれば次のステップへ行く。なければ新しくゲートを作り手続きを終る。

step2 見つめられた元の回路のゲートのレベルと、部分回路における目的のゲートのレベルとを比べ、元の回路の方が小さければそのゲートの出力を用い、そうでなければ新しくゲートを作り、見つめられた元の回路のゲートもそれによって置き換える。

また、部分回路は元の回路に比べて小さいのでトランسدクション法を適用することも比較的容易になると考えられる。

## 4 実験結果

上述した手法に基づいて、論理回路の遅延最適化プログラムをC言語を用いて作成し、オムロン社のLUNA88K上で実行した。この時、生成する部分回路はファンイン数を4に制限し、ファンアウト数の制限はしていない。また置換対象を決定するための外部変数の数は5個とし、対象の選択は出力段から始め、入力段に達するまで繰り返し置換を行なっている。

なお本プログラムのSBDD処理は、現NTT湊真一氏によるSBDDパッケージを使用している。

表1に、このプログラムによる論理回路の遅延最適化結果を示す。表中のInは回路の外部入力数、Outは外部出力数を表している。またCPU時間の単位は秒で表されている。

この結果によると、いくつかの回路においてはそれほどの面積の増加もなく段数が削減されていることがわかる。しかし逆に面積が大きく増加してしまった回路も存在する。

今回は比較的小さい回路に対してのみ実験を行なったが、本来より大きな回路を対象とすることが目的なので、そのような回路に対する実験も行なっていく予定である。

回路	In	Out	初期回路	適用結果	CPU
5xp1	7	10	105/213/34	97/207/22	3.0
Z9sym	9	1	154/322/16	154/327/15	3.8
add6	12	7	105/190/27	110/207/23	3.7
adr4	8	5	62/112/19	65/118/15	2.0
alu1	12	8	37/67/4	38/79/3	2.3
alu2	10	8	78/149/17	79/154/12	2.2
alu3	10	8	73/143/15	74/152/10	2.7
apla	10	12	84/177/14	86/183/13	2.1
con1	7	2	23/40/6	20/40/5	1.8
dc2	8	7	97/202/39	122/273/18	4.4
dk17	10	11	57/114/13	67/134/12	2.0
f51m	8	8	111/225/37	118/250/20	3.8
misex1	8	7	47/89/18	45/88/11	1.9
mlp4	8	8	219/492/41	223/511/37	5.4
radd	8	5	57/104/19	61/112/13	2.0
risc	8	31	71/137/15	82/180/6	3.0
root	8	5	103/228/28	121/275/24	4.1
rot8	8	5	106/225/27	106/231/24	3.2
sao2	10	4	101/216/22	102/221/21	3.2
sex	9	14	44/83/9	44/87/6	1.8
sqn	7	3	86/179/24	100/232/18	3.7
sqr6	6	11	106/218/24	126/278/12	3.9
z4	7	4	53/99/16	70/153/12	3.7

(ゲート数 / 結線数 / 段数)

表 1: 実験結果

## 5 あとがき

本稿ではCSPFを用いた部分回路置換による遅延最適化手法を提案し実験を行なった。その結果回路によっては面積の増加も少なく、段数削減を行なうことができた。今後は段数削減に、より効果的な置換部分の発見や大規模な回路への適用手法について検討していく予定である。

## 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に深謝致します。

## 参考文献

- [1] S.Muroga, Y.Kmbayashi, 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, October 1989.
- [2] S.Sawada, Y.Kambayashi, S.Muroga: Generation of Fan-in Restricted Initial Networks for Transduction Method, *Proc. SASIMI'92*, April 1992.
- [3] Graph-based algorithms for boolean function manipulation, *IEEE Trans. Comput.*, C-35(8), August 1986.
- [4] S.Minato, N.Ishiura, S.Yajima: Shared binary decision diagram with attributed edges for efficient boolean function manipulation, In *Proc. 27th Design Automat. Conf.*, June 1990.