

## 論理回路の最適化 及び 仕様変更 のための

5B-1

## エラー補償型トランスダクション法

山下茂 幸田武範 上林彌彦

京都大学工学部

## 1 はじめに

VLSI の設計において CAD は、単なる回路の解析支援システムから回路の自動合成システムへと発展してきた。その過程において、数多くの解析・最適化システムが開発され、実用化されてきた。

しかし、実際の VLSI の設計過程において回路動作の誤りの修正に膨大な時間と労力がかかる現状にもかかわらず、仕様変更支援システムの開発はほとんどされていない。

本稿では、論理回路最適化のために開発されたトランスダクション法<sup>[1]</sup>の一手法であるエラー補償法<sup>[1][2]</sup>を利用して入力を部分的に回路で置き換える方法について検討する。このためエラー補償の改良も行なったためその結果についても報告する。

## 2 基本的事項

本章では、エラー補償法に関する基本的事項について簡単な説明を行なう。

トランスダクション法に許容関数という概念があり、その部分集合を CSPF(Compatible Set of Permissible Functions)とよび、0、1、\* (don't care) の3値を要素として持つ<sup>[1]</sup>。エラー補償法では、この CSPF を拡張した CSPFE (誤りつき許容関数)を用いる。CSPFE は 0、1、\*、0'、1' の5値を持つ。ここで、0' (0 エラー) は現在は 0 であるが本来は 1 が要求されている値であることを表し、1' (1 エラー) は現在 1 で本来は 0 が要求されている値を表す。

エラー補償法は、あるゲートを削除しその結果回路の出力に起こったエラーを計算し、各ゲートごとにエラーを結線の削除・置換・追加を行なうことでエラーを補償していく手法である。一度エラーが補償された後も、更に違うゲートを削除し結果が改良される限り上記のアルゴリズムを適用して回路を最適化する。

## 3 エラー補償法の改良

論文<sup>[1][2]</sup>に示されるエラー補償法の基本アルゴリズムに対して、以下の様な改良を加えた。

- i) エラー補償法は、回路の最適化能力については強力であるが計算に非常に時間がかかるため、トランス

ダクション法の他の手法と比べあまり実用的とはいえなかった。そのため、冗長な結線を削除する方式の変更を行ない、各ゲートにおけるエラー補償成功の可能性を増やすことで、回路全体の計算時間を減少させる様にした。

- ii) CSPFE 計算時にあるゲートの出力にエラーありの値とエラーなしの値が伝搬されてきた時、エラーなしの値を採用 (手法 1) していたが、ここでエラーありを採用する手法<sup>[2]</sup> (手法 2)、どちらかを選ぶ手法 (手法 3) の3種類を用意し、後述の仕様変更への適用時に使えるように回路形状変更能力を高めた。

## 4 エラー補償法の改良前後の比較

3章で述べた手法を実装し、MCNC ベンチマーク回路に対して実験を行なった。プログラム内部での関数表現には、湊氏による SBDD パッケージ<sup>[3]</sup>を用いている。

- 冗長結線削除方式の変更についての実験の結果  
冗長結線削除方式の変更についての実験を 31 回路について行なった。この結果、ゲート数では 5 例が平均 8% 減少し、5 例で平均 3% 増加した。結線数では 7 例が平均 5% 減少し、増加したのはわずか 3 例で平均 2% であった。計算時間についてはすべて短縮され、短縮効果の大きかった 9symml などでは 69% も短縮された。平均としては 23% 短縮された。以上の結果より、最適化能力を失うことなく目的の計算の高速化を達成することができたことがわかった。
- 各 CSPFE 計算法による最適化結果の比較  
表 1 に各 CSPFE 計算法による最適化結果の比較を示す。この結果より、回路によって各手法の最適化の能力が異なることがわかり、これらの手法を組み合わせる可能性が大きくなる。ここで太字は 3 手法のうちで最良の結果を示している。

## 5 エラー補償法の仕様変更への適用

まず、仕様変更について述べる。一般に、回路設計においては回路の設計仕様を変更することは少なくない。このような場合の修正は、回路のかなり大きな部分に及ぶことも多く最悪の場合回路そのものの再設計を余儀なくされる場合もある。

しかし、再設計された回路がもとの回路と非常に似ていることも多く、元の回路を修正して利用した方がより実用

表 1: 各 CSPFE 計算法による最適化結果の比較

回路名	手法 1	手法 2	手法 3
cm150a	48/103/155	51/101/265	<b>42/91/197</b>
cm162a	<b>47/91/99.0</b>	49/94/67.7	49/94/52.5
cm42a	<b>23/54/2.36</b>	25/54/1.83	23/54/2.34
cm82a	<b>30/49/1.28</b>	27/46/1.12	<b>30/49/1.24</b>
f51m	93/187/328	78/157/115	<b>72/139/131</b>
mux	64/120/271	<b>54/112/166</b>	56/116/107
sct	<b>80/163/406</b>	84/167/244	94/188/497
x2	43/95/22.5	41/89/23.4	<b>40/83/21.9</b>
z4ml	39/73/10.2	40/70/6.80	<b>34/62/12.1</b>

(ゲート数 / 結線数 / 計算時間 (秒))

的である場合もある。既存回路の再利用について次に述べる方法が考えられる。

i) 設計者が設計中の回路に修正を加える場合

このような場合、元の回路内で結線の追加・削除などを行なうことで、回路そのものを再設計せずに済ませることができる。しかし計算機能力が飛躍的に向上している現在では、回路の再設計に要する時間とコストは、回路の再利用のためにかかる時間とコストと比べてあまり差がないため、再利用の実用性は少ない。

ii) 既にチップなどになっている既存回路の仕様を変更して使いたい時

これは、既存回路の出力を自分の目的にあうように変更したい場合などである。もちろん、チップ化されているので回路内部を変更することはできないので、その既存回路の入力、または出力に別の回路を繋ぐことにより全体としての仕様を変更しようとするものである。これは、多くのチップを用いて回路を設計する際に、デバッグの終わった既存の回路を利用する際などに使えるため、かなり有用であるといえる。

以下、エラー補償法を用いて上記の二つの場合の仕様変更(回路の再利用)を行なう手法を示す。

前者は、要求により変更された部分を元の回路に対する「エラー」として、そのエラーを補償するべくエラー補償法をかければよいだけなので、実験は行なわなかった。

後者の場合の様に既存回路の仕様を変更して使いたい時、入力補正の方が困難な問題といえる。このため、ここでは入力補正問題のみを取り上げる。まず、エラー補償法の CSPFE 計算を利用して、求める回路出力と元の回路出力の差をエラーとして途中のゲートで補償をせずに回路の入力まで伝える。この結果、計算された入力端の CSPFE からそれぞれの入力端の「補正入力」を計算する。ここで補正入力とは、その論理関数をその回路の入力端に与えてやれば、要求する出力が回路の出力端から得られるような論理関数である。この補正入力計算により見つかった場合、元の入力から補正入力を生成する様な論理

回路を生成し、元の回路と繋げばよい。そうすれば、既にチップなどになっている既存回路であっても、似たような仕様であれば多少の論理回路を追加するだけで利用することができる。

## 6 エラー補償法の仕様変更への適用の有用性検証実験の結果とその考察

エラー補償法の仕様変更への適用の有用性検証実験を行った。本実験では、各試行ごとに異なる補正入力の候補が計算されるよう各結線の CSPFE 計算時にランダムに計算法を選択する様にした。計算された補正入力の候補が必ずしも補正入力であるとは限らないため、上記の様に計算法を選択することで補正入力の条件を満たすものが見つかるまで試行を繰り返すことができ、かなり高い確率で正しい補正入力をみつけられるようになった。(ただし本実験では、試行回数の上限を 100 回とした。)

また本実験では、元の真理値表の各要素の論理を変更し、その変更に対して補正入力計算できるかを調べた。28 回路について変更要素の数を 0 個から 100 個まで変化させて実験を行なった結果、1 要素のみの変更の場合ほぼ成功した。それ以上では回路の形状によって大きく異なり、結果の悪いものでは 2、3 要素くらいからほとんど再利用できなくなるが、cm150a など結果の良いものでは 100 要素変更しても再利用できることがわかった。平均としては、10 要素くらいまでなら変更要素の位置しただいでは補正入力を見つけることができる事がわかった。

以上の有用性検証実験の結果から、わずかの仕様変更に対してなら、本手法を用いることで回路を再利用できる可能性が高いことが検証できた。なお補正入力を作るのに要する回路の複雑さの評価や出力の補正などの方法との併用が残された課題である。

## 謝辞

有益な御助言、御指導下さるイリノイ大学計算機科学科室賀三郎教授、ならびに SBDD パッケージを使用させていただいた矢島研究室の皆様へ深謝する。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] Y.Kambayashi, H.C.Lai, J.N.Culliney, S.Muroga, "NOR Network Transduction Based on Error-Compensation (Principles of NOR Network Transduction Programs NETTRA-E1, NETTRA-E2 and NETTRA-E3)", Report No. UIUCDCS-R-75-737, Dept. of Comp.Sci. Univ. of Illinois (June 1975).
- [2] 澤田 直、桑野浩二、上林弥彦: "エラー補償型トランスタクション法による回路の最適化について"、情報処理学会第 48 回全国大会、5B-10(1994-3).
- [3] S.Minato, N.Ishiura, S.Yajima, "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation", Proceedings of 27th Design Automation Conference(1990), 52-57.