

非明示的集合表現に基づく修正箇所シミュレーションによる論理診断処理の効率化 An Improved Error Diagnosis Technique Based on Location Variable Simulation Employing Implicit Representation

津山 勲貴[†] 黒木 修隆[†] 沼 昌宏[†]
Hiroki Tsuyama[†] Nobutaka Kuroki[†] Masahiro Numa[†]

1. はじめに

大規模化・複雑化が進む LSI 設計・製造工程において、仕様変更や設計誤りの混入、タイミング制約違反などにより、設計変更要求 (ECO: Engineering Change Order) が生じる場合がある。設計期間やコストの増加を抑制しつつ ECO に対応する手法として、回路に含まれる論理設計誤りを自動的に修正する論理診断手法 [1],[2] に基づき、配線変更のみで修正を実現する論理再合成手法 [3],[4] が提案されている。従来手法である、充足可能性判定に基づく誤り追跡入力生成と機能特定を用いた論理診断手法 [2] のうち、EPI, 6 値シミュレーションに基づく組合せ箇所抽出・絞り込み処理において、明示的な集合表現を用いるため、多重度や回路規模の増加によって処理時間が大きく増加する点が課題となっていた。

本稿では、修正箇所を変数とする論理関数を用いて各信号線が取る各値の可能性を表現することで、修正すべき組合せ箇所の集合を非明示的表現に基づいて求める、修正箇所シミュレーションを提案することで、組合せ箇所の抽出に要する処理時間を大幅に短縮する。

2. 論理診断手法の概要

必要な用語の定義を行った上で、本稿で扱う論理診断手法の概要について述べる。

定義 1 機能仕様, LUT 回路, 診断対象回路

論理回路として実現すべき機能仕様 S を、 n 入力変数ベクトル $\mathbf{x} = (x_1, \dots, x_n)$ に対する p 出力論理関数ベクトル $\mathbf{f}_s = (f_{s1}, \dots, f_{sp})$ で表す。真理値表 (LUT: Look-Up Table) を素子とする LUT 回路が機能誤りを含むとき、これを診断対象回路 G と呼び、 G の外部出力関数をベクトル $\mathbf{f}_g = (f_{g1}, \dots, f_{gp})$ で表す。□

定義 2 誤り追跡入力

一つの変数 X または \bar{X} を含む入力ベクトル $\boldsymbol{\alpha} = (a_1, \dots, a_{i-1}, X/\bar{X}, a_{i+1}, \dots, a_n)$ に対して

$$f_{sj}(\boldsymbol{\alpha}) = X \quad (1)$$

$$f_{gj}(\boldsymbol{\alpha}) = a \quad (2)$$

が成立するとき、 $\boldsymbol{\alpha}$ を誤り追跡入力と呼ぶ。ただし、 X/\bar{X} は X または \bar{X} の一方を表し、 $a_k \in B = \{0, 1\}$ ($1 \leq k \leq n; k \neq i$), $a \in B$ とする。論理診断には、 n_α 個の誤り追跡入力を適用するものとし、必要に応じてそれぞれ $\boldsymbol{\alpha}_q$ ($1 \leq q \leq n_\alpha$) で識別する。□

定義 3 修正箇所, 組合せ箇所

設計誤りを含むと想定する一つの LUT を、修正箇所 l_i と呼ぶ。 m 個の修正箇所からなる空でない集合を、多重度 m の組合せ箇所と呼び、 $L^m = \{l_i | i = 1, \dots, m\}$ で表す。□

図 1 に示す従来の論理診断手法 [2] では、SAT ソルバを用いて生成した誤り追跡入力をもとに、EPI と 6 値シ

ミュレーションを用いて、具体的な修正方法を特定せずに修正候補となる組合せ箇所の抽出と絞り込みを行う。絞り込まれた組合せ箇所の集合に対して、該当する LUT の内容を真理値変数で表現した真理値シミュレーション、さらに最終的には SAT ソルバを用いた機能特定処理によって、修正解を得ている。

3. 修正箇所シミュレーション

本章では、修正箇所シミュレーションにおいて必要な用語の定義を行ったうえで、概要について述べる。

定義 4 修正箇所変数, 信号値関数

診断対象回路 G に含まれる各 LUT に対して与えられ、その機能修正の有無を表す論理変数を修正箇所変数 l_v と呼び、機能修正がある場合は $l_v = 1$ 、ない場合は $l_v = 0$ で表す。誤り追跡入力 $\boldsymbol{\alpha}$ が外部入力に与えられた場合の、ある信号線 y の値が 4 値: $0, 1, X, \bar{X}$ をとるための必要条件を、それぞれ修正箇所変数による信号値関数 $v_0(y)$, $v_1(y)$, $v_X(y)$, $v_{\bar{X}}(y)$ で表す。□

定義 5 修正条件

誤り追跡入力 $\boldsymbol{\alpha}$ に対して、 $f_{sj}(\boldsymbol{\alpha}) \neq f_{gj}(\boldsymbol{\alpha})$ となる不一致外部出力 poj に関する理想出力値を $s_j = f_{sj}(\boldsymbol{\alpha})$ で表すとき、 poj の値が s_j と等しくなるための必要条件 $v_{s_j}(poj)$ のことを、 poj に対する修正条件と呼び、 $f_{c_j} = v_{s_j}(poj)$ で表す。また、誤り追跡入力 $\boldsymbol{\alpha}$ に対するすべての不一致外部出力 poj に関する修正条件の論理積を、 $\boldsymbol{\alpha}$ に関する修正条件 $F_c(\boldsymbol{\alpha}) = \prod f_{c_j}$ で表す。さらに、適用するすべての誤り追跡入力 $\boldsymbol{\alpha}_i$ に関する修正条件の論理積を、適用する全誤り追跡入力に対する修正条件 $F_{c_{all}} = \prod F_c(\boldsymbol{\alpha}_i)$ と呼ぶ。□

定義 6 多重度指定関数, 多重度制限関数

多重度指定関数 F_m は、 N 個の修正箇所変数から m 個を選ぶ組合せを表す論理関数である。多重度制限関数 $F_{lmt(m_{max})} = \sum F_m$ は、 N 個の修正箇所変数から m 個以下の変数を選ぶ組合せを表す論理関数であり、 $1 \leq m \leq m_{max}$ の各多重度 m に関する多重度指定関数 F_m の論理和で表現される。□

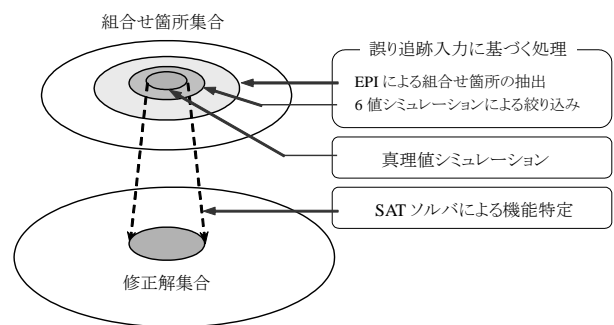


図 1 従来の論理診断処理の概要

[†] 神戸大学, Kobe University

修正条件 $F_{c_{all}}$ と、多重度指定関数 F_m もしくは多重度制限関数 $F_{lmt(m_{max})}$ との論理積をとることで、それぞれ多重度 m もしくは m 以下の組合せ箇所限定した修正条件が得られる。

図 2 に、提案する修正箇所シミュレーションの処理手順を示す。修正箇所シミュレーションは、従来の論理診断手法のうち、EPI, 6 値シミュレーションに基づく組合せ箇所の抽出・絞り込み処理の代わりに行う。

修正箇所シミュレーションでは、まず多重度制限関数で修正条件 $F_{c_{all}-lmt(m_{max})}$ を初期化する。次に、各誤り追跡入力に対して、外部入力値 $0, 1, X, \bar{X}$ に対応する信号値関数を設定する。この際、外部入力の信号値関数は、ノードをもたない BDD (定数) である。そして、外部入力に近い LUT から順に、入力の信号値関数を利用して、LUT の真理値表に基づいて計算した出力の信号値関数を、不一致外部出力まで伝搬する。伝搬された信号値関数より、理想出力値 $s_j = f_j(\alpha)$ と等しい外部出力値を得るための必要条件を修正条件 $f_j = v_j(p_o_j)$ として求める。修正条件 $F_{c_{all}-lmt(m_{max})}$ について、各誤り追跡入力の各不一致外部出力における修正条件 f_j との論理積で更新する。

さらに、BDD ノード数の削減による必要メモリ容量削減および処理時間短縮を目的として、ある論理関数によって定義域を限定するコファクタ演算を修正箇所シミュレーションにおける信号値関数に適用する。多重度制限関数 $F_{lmt(m_{max})}$ または修正条件 $F_{c_{all}-lmt(m_{max})}$ を定義域指定関数としてコファクタ演算を行い、信号値関数のノード数を削減することで、演算に要する時間がノード数にほぼ比例する BDD の処理時間を短縮する。

4. 実験結果

従来手法および提案手法を計算機 (CPU: Core i7-3770 3.40 GHz, RAM: 32 GB) 上に実装し、実験評価を行った。

表 1 に示す、C7552 回路、b14_opt_C 回路に対して 4, 5 箇所の素子機能誤りを無作為に挿入した回路を各 10 例ずつ、合計 40 例について実験を行った。提案手法については、コファクタ演算に用いる定義域指定関数に関して、

- 提案手法 A : 多重度制限関数 $F_{lmt(m_{max})}$ を適用
- 提案手法 B : 修正条件 $F_{c_{all}-lmt(m_{max})}$ を適用
- 提案手法 C : 提案手法 A, B を適応的に切替え

の 3 種類の設定をもとに実験を行い、SAT ソルバに関しては minisat2.2 を用いた。評価項目は、組合せ箇所の抽出・絞り込みに要した処理時間とする。

表 2 に、各手法による処理時間の幾何平均値を示す。提案手法 B, C ではすべての回路例に対して、従来手法よ

表 1 ベンチマーク回路

回路	ゲート数	LUT 数	外部入力数	外部出力数
C7552	3,513	805	207	108
b14_opt_C	5,923	3,380	277	299

表 2 組合せ箇所抽出・絞り込み処理時間 [s]

回路	m	従来手法	提案手法 A	提案手法 B	提案手法 C
C7552	4	0.89	0.29	0.28	0.28
	5	13.57	0.59	0.71	0.62
b14_opt_C	4	35.60	238.86	19.42	19.90
	5	10,477.62	2,876.83	629.76	414.06

りも大幅に処理時間を短縮できた。特に提案手法 C では b14_opt_C の多重度 5 において、処理時間を従来比で 96.0% 短縮できた。また、提案手法 A において、C7552 の多重度 5 では処理時間を 95.7% 短縮する一方、b14_opt_C の多重度 4 では 6.71 倍の処理時間を要した。これは、定義域指定関数に用いた多重度制限関数の、ノード数は少ないが絞り込める修正条件の範囲が狭い性質によると考えられる。

次に、C7552, b14_opt_C の多重度 4, 5 における処理時間短縮率の平均は、提案手法 A が 59.8%, 提案手法 B では 84.8%, 提案手法 C では 86.7% であり、提案手法 C が最も処理時間を短縮できた。最も効果の大きかった C7552 のある回路例では 99.984% の短縮効果を得た。一方で、提案手法 C が、提案手法 B より処理時間を要している回路例も存在した。

5. まとめ

本稿では、充足可能性判定に基づく論理診断手法における、組合せ箇所抽出・絞り込み処理の効率向上による処理時間短縮を目的として、修正箇所シミュレーションを提案した。提案手法では、BDD による非明示的なグラフ表現によって、組合せ箇所を効率的に表現することで、処理時間の大幅な短縮を可能とした。

提案手法を計算機上に実装して評価実験を行った結果、4, 5 箇所の素子機能誤りを含む C7552, b14_opt_C 回路に対して、提案手法 C によって処理時間が平均 86.7% 短縮され、提案手法による処理時間短縮効果を確認した。

今後の課題としては、コファクタ演算に用いる論理関数を適応的に切り替える条件についての考察を深め、処理時間のさらなる短縮を図ることが挙げられる。

参考文献

- [1] 沼 昌宏, 井上 宏, 皆見利行, 黒木修隆, 山本啓輔, “真理値シミュレーションに基づく LUT 論理診断手法”, 情報処理学会論文誌, vol. 43, no. 5, pp. 1252-1259, 2002.
- [2] 片山直樹, 松山友紀, 渡辺浩介, 廣瀬哲也, 黒木修隆, 沼 昌宏, “充足可能性判定に基づく誤り追跡入力生成と機能特定を用いた論理診断手法,” DA シンポジウム 2012, pp. 13-18, 2012.
- [3] H. Inoue, T. Iwasaki, T. Sugane, M. Numa, and K. Yamamoto, “Application of error diagnosis technique to incremental synthesis,” IEICE Trans. Fundamentals, vol. E86-A, no. 12, pp. 3214-3217, 2003.
- [4] 下野友大, 天満 健, 千崎弘人, 廣瀬哲也, 黒木修隆, 沼 昌宏, “メタル配線により再構成可能なセルと論理再合成への応用,” DA シンポジウム 2012, pp. 7-12, 2012.

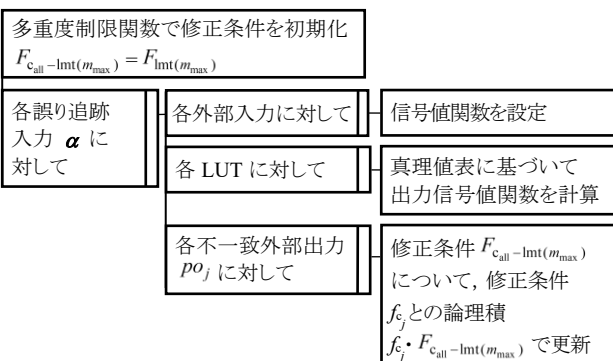


図 2 修正箇所シミュレーションの処理手順