

信号線誤りに対応した LUT 論理診断手法

井上 宏^{†,††} 黒木 修 隆[†]
沼 昌 宏[†] 山 本 啓 輔[†]

LUT (Look-Up Table) で表現された組合せ回路に含まれる設計誤りの修正に関して、新たに信号線誤りの修正を可能とする論理診断手法を提案する。信号線誤りは、信号線過剰誤りと信号線欠落誤りに分類される。これらのうち、信号線過剰誤りは LUT 機能誤りの修正問題に帰着することにより修正法を求めることが可能である。一方で、信号線欠落誤りの修正は、信号線欠落が発生している誤り箇所に対して接続すべき信号線を求めるとともに、LUT 内部の機能をも正しく決定する必要がある。このとき、決定された LUT 機能によって欠落信号線が満たすべき条件が変化するため、あらゆる LUT 機能割当てに対して条件を求める必要があるという点で、信号線欠落誤りの修正は困難であった。本手法では、外部入力変数と LUT 内部論理を表現する真理値変数のブール関係に対応する特徴関数をもとに、欠落信号線の満たすべき条件を表現する。特徴関数を用いることで、あらゆる修正法に対する条件を非明示的に表現できるため、存在するすべての解を求めることが可能となる。

An LUT-based Multiple Error Diagnosis Technique Extended for Single Line Errors

HIROSHI INOUE,^{†,††} NOBUTAKA KUROKI,[†] MASAHIRO NUMA[†]
and KEISUKE YAMAMOTO[†]

This paper presents an approach to rectify multiple logic design errors including single line errors in LUT-based combinational circuits. Single line errors are classified into missing-line errors and extra-line errors. Although the conventional technique called EXLTV is able to rectify extra-line errors only by correcting LUT contents, but is not applicable to missing-line errors. Rectification of missing-line errors requires both correcting LUT contents and finding a line to be connected to the LUT. Since the condition to be satisfied with the missing-line depends on the LUT contents at the error location, rectification of missing lines is very difficult. The proposed algorithm introduces the condition derived from the characteristic function to represent Boolean relation among the primary input variables and the truth variables assigned to each entry of LUT. Since the characteristic function implicitly represents the condition for any possible assignment of truth variables, every solution is obtained.

1. はじめに

論理合成ツールを利用した論理回路設計が一般的となっているが、合成された回路がタイミング仕様を満足しない場合、人手で回路に変更を加えることがある。変更にもなって回路中に設計誤りが混入する可能性があり、論理検証が不可欠となる。検証により設計誤りの存在が確認された際に、その修正の自動化を目的とするのが論理診断である。論理診断手法は設計誤りの修正のための設計変更要求 (ECO: Engineering

Change Order) に限らず、仕様の変更にもなう設計変更要求に対するインクリメンタル合成^{1),2)}にも応用できる。すなわち、以前の仕様に基づいて設計・合成された回路が誤りを含むと見なして論理診断手法を適用することにより、最小の修正で仕様変更に対応しようとする。

論理診断手法について、これまでにゲートレベル回路を対象とする手法^{3)~5)}ならびに LUT (Look-Up Table) で構成される LUT 回路を対象とする、LUT 論理診断手法^{6)~8)}が提案されてきた。LUT 論理診断は、LUT 型 FPGA (Field Programmable Gate Array) で実現される回路の修正のみならず、ゲート回路を対象とする論理診断における設計誤りモデルの一般化にも有用である。具体的には、図 1 に示すような複数のゲート欠落誤りやインバータ欠落誤りへの対応

[†] 神戸大学工学部
Faculty of Engineering, Kobe University

^{††} 日本学術振興会特別研究員
Research Fellow of the Japan Society for the Promotion of Science

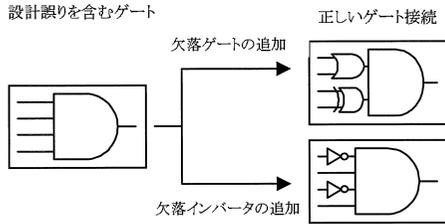


図1 LUT 回路モデルを利用した複雑な設計誤りの修正
Fig. 1 Rectification of complex error types using LUT-based circuit model.

を可能とする。

従来の LUT 論理診断手法のうち、ブール関係に基づく手法⁶⁾は、i) 修正すべき箇所を効率良く特定できない点、ii) 同一パス上に存在する複数の設計誤りには対応できない点、に問題があった。一方で、複数の入力組合せに対する回路の出力値を理想出力値と比較する手法⁷⁾では、100%の限定率を保証することが困難な点に問題があった。

これらの問題を解決すべく提案した EXL_{TV} (Extended X-algorithm for LUT-based circuits with Truth Variables) 法⁸⁾は、i) パターンに基づく処理と、ii) BDD (Binary Decision Diagram)^{9),10)}に基づく論理関数処理、を併用する点に特徴がある。パターンに基づく処理では、誤り追跡入力¹¹⁾と呼ぶ入力パターンを用いて考慮すべき組合せ箇所を効率良く削減する。続いて、論理関数処理で LUT 内部機能を決定することで、現実的な処理時間で 100%の限定率を達成した。さらに、LUT 内部を真理値変数と呼ぶ変数で表現することによって、同一パス上の複数の設計誤りに対する修正をも実現した。その一方で、LUT 機能誤りのみを修正対象とするため、設計変更要求に対応したインクリメンタル合成に対応する場合、信号線の追加をともしない修正は不可能であった。

そこで本論文では、EXL_{TV} 法をもとに、信号線誤りに対応した論理診断を行う EXL_{SL} (Extended X-algorithm for LUT-based circuits including multiple function errors and Single Line errors) 法を提案する。

2. 用語と問題の定義

$B = \{0, 1\}$, $T = \{0, 1, *\}$ とする。* は、ドントケアを表す。LUT は、 k 入力と 1 出力を持つ真理値表によって表される任意の機能を実現する素子である。その機能は、LUT 入力変数ベクトル $y = (y_1, \dots, y_k)$ に対する LUT 関数 $h: B^k \rightarrow B$ で表現される。

診断対象とする LUT 回路は、非巡回有向グラフ

$G = (V, E)$ で表される。ここでノードの集合 V は、1 個以上の LUT、 n 個の外部入力、 p 個の外部出力を要素とする。各ノードは 1 本以上の有向エッジ $e \in E$ で他のノードと接続され、接続のない LUT 入力・出力および外部入力・出力は考慮しない。

実現すべき機能仕様 S を、 n 入力変数ベクトル $x = (x_1, \dots, x_n)$ に対する p 出力論理関数ベクトル $f_s = (f_{s1}, \dots, f_{sp})$ で表す。設計誤りを含む LUT 回路 G の外部出力関数ベクトルを $f_g = (f_{g1}, \dots, f_{gp})$ で表す。

対応する LUT 関数の修正を想定する LUT のことを、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 L^m と呼ぶ。修正箇所 l_i に関して各入力割当て $y_a \in B^{k_i}$ に対する出力値をそれぞれ $v_{i,0}, \dots, v_{i,2^{k_i}-1}$ の変数で表すとき、この変数を真理値変数と呼ぶ。また、 $v_i = (v_{i,0}, \dots, v_{i,2^{k_i}-1})$ を修正箇所 l_i における真理値変数ベクトルと呼ぶ。 v_i の各変数への割当てを l_i における真理値割当て $t_i \in B^{|v_i|}$ と呼ぶ。この t_i は、LUT 機能の修正法、すなわち正しい LUT 機能 (LUT 関数 h) と 1 対 1 に対応する。組合せ箇所 $L^m = \{l_1, \dots, l_m\}$ に含まれる各箇所に対応した真理値変数から成るベクトルを、真理値変数ベクトルを $v^m = (v_{1,0}, \dots, v_{1,2^{k_1}-1}, \dots, v_{m,0}, \dots, v_{m,2^{k_m}-1})$ と定義する。 v^m の各変数への割当てを真理値割当てベクトル $t^m \in B^{|v^m|}$ と呼ぶ。

1 つの修正箇所 $l_i \in L^m$ ($i = 1, \dots, m$) に信号線欠落を想定する組合せ箇所 L^m のことを、信号線欠落を想定した組合せ箇所 $L_{ML}^m(i)$ と呼ぶ。信号線欠落を想定する修正箇所 l_i に関して、欠落入力線に対応する LUT 入力変数を欠落入力変数 y_{ML} と呼ぶ。ただし、 y_{ML} は信号線欠落を想定する LUT 入力をも表すものとする。さらに、 $L_{ML}^m(i)$ における真理値変数ベクトル v^m と欠落入力変数 y_{ML} を考慮した j 番目の外部出力関数を、 $f_{gj}(x, v^m, y_{ML})$ と表す。

信号線欠落を想定した組合せ箇所 $L_{ML}^m(i)$ に対する修正候補 $E_{ML}^m(i) = (L_{ML}^m(i), t^m, w)$ は、 $L_{ML}^m(i)$ と真理値割当てベクトル t^m 、および修正箇所 l_i に接続する信号線 w により表される。その修正によって機能仕様 S を満たす回路が得られる $E_{ML}^m(i)$ を、修正解 E_T^m と呼ぶ。多重度 m に対して存在するすべての E_T^m からなる集合を、修正解集合 $C_{E_T^m} = \{E_T^m\}$ と呼ぶ。

信号線誤りは、信号線過剰誤りと信号線欠落誤りに分類されるが、前者は LUT 機能誤りの修正問題に帰着できる。よって提案する EXL_{SL} 法では、以下のように信号線欠落誤りの修正を想定した論理診断問題を



図2 信号線誤りを考慮した論理診断処理
Fig.2 Rectification of design errors including single line errors.

扱う。

論理診断問題：複数の設計誤りを含む LUT 回路 G と機能仕様 S をもとに、 $C_{E_T}^m \neq \phi$ を満たす最小の多重度 m に対して修正解集合 $C_{E_T}^m$ を求める。ただし、信号線欠落誤りはたかだか 1 カ所に存在するものとする。 □

3. 処理概要

EXLSL 法の処理概要は、次のようになる。

- Step 1 誤り追跡入力¹¹⁾の生成
- Step 2 可制御性指標 EPI による組合せ箇所の抽出
- Step 3 6 値シミュレーションによる組合せ箇所の絞り込み
- Step 4 真理値シミュレーションによる組合せ箇所の絞り込み
- Step 5 信号線誤りを考慮した LUT 機能修正

診断の初期段階においては、修正すべき組合せ箇所の特定を行う必要がある。そこで、まず Step 1 で組合せ箇所の抽出と絞り込みに用いる誤り追跡入力の生成を行う。続いて、Step 2~4 の処理を行うことにより組合せ箇所を絞り込み、Step 5 で残存する組合せ箇所に対して信号線誤りを考慮した LUT 機能修正を行う。上記の処理のうち、Step 1 および Step 2 に関しては、EXLTV 法⁸⁾からの変更点はなく、従来どおりの処理を行う。以降の処理については信号線誤りの修正への対応が必要となる。

Step 3 の 6 値シミュレーションについては、信号線欠落誤りに対応するために次の点を変更した。信号線欠落誤りを想定しない場合、シミュレーションにおける修正箇所の出力値は信号線の追加を考慮せずに決定できる。EXLTV 法では、効率良く組合せ箇所の絞り込みを行うため、修正箇所の入力値による場合分けにより出力値を決定していた。提案手法では、修正箇所に対して信号線の追加を考慮するため、文献 4) で提案された手法と同様に、その出力値には不定信号値 E を割り当てる。

6 値シミュレーション処理後に残存する組合せ箇所

L^m に信号線欠落を想定した $L_{ML(i)}^m$ ($i = 1, \dots, m$) に対して、図 2 に示す処理 (Step 4, 5) を行う。Step 4 では、真理値シミュレーション⁸⁾を行い、修正解が得られる見込みのない $L_{ML(i)}^m$ を削除する。信号線欠落誤りに対応するための変更点については、LUT 機能の修正法とともに次章で詳述する。

最後に Step 5 で、LUT 機能の変更のみによる修正の可能性について判定する。もし修正可能であれば、その修正法に基づき回路を修正する。LUT 機能の変更だけでは修正不可能である場合には、信号線欠落誤りを考慮した修正を試みる。

4. 信号線誤りに対応した真理値シミュレーションと LUT 機能修正

本章では、誤り追跡入力を用いた処理により抽出された各組合せ箇所に対して適用する、真理値シミュレーションおよび LUT 機能修正法について述べる。

まず、4.1 節で必要な定義を行う。続いて 4.2 節で信号線欠落誤りに対応するための真理値シミュレーションに関する変更点について説明し、4.3 節で、信号線欠落誤りに対応した LUT 機能修正法を提案する。

4.1 諸定義

論理関数 $f(x_1, \dots, x_i, \dots, x_n)$ に対して $x_i = 0, 1$ を代入した結果を、それぞれ f の \bar{x}_i に関するコファクタ $f|_{\bar{x}_i}$ 、 x_i に関するコファクタ $f|_{x_i}$ と呼ぶ。論理関数 $f(x_1, \dots, x_n)$ について、その関数が真に依存する変数の集合を依存変数集合 (support) と呼び、 $sup(f)$ と表す。すなわち、 $\forall x \in sup(f) : f|_{\bar{x}} \neq f|_x$ かつ $\forall x \notin sup(f) : f|_{\bar{x}} = f|_x$ である。

不完全記述関数 $\hat{h} : B^n \rightarrow T$ が、 $h^L \leq h^U$ ($h^L \cdot \overline{h^U} \equiv 0$) を満たす 2 つの完全記述関数 $h^L, h^U : B^n \rightarrow B$ との間に

$$\hat{h}(x) = \begin{cases} 0 & (x|h^L(x) = h^U(x) = 0) \\ 1 & (x|h^L(x) = h^U(x) = 1) \\ * & (x|h^L(x) = 0, h^U(x) = 1) \end{cases} \quad (1)$$

の関係が成り立つとき、 $\hat{h} = [h^L, h^U]$ と表す。また、完

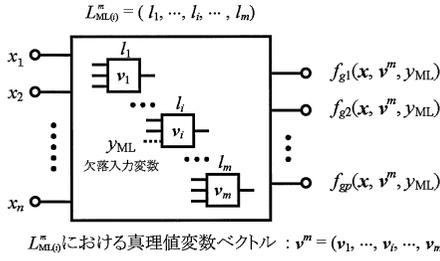


図3 信号線誤りを想定した組合せ箇所 $L_{ML(i)}^m$ に対する真理値変数ベクトル v^m と欠落入力変数 y_{ML} の割当て

Fig.3 Assignment of v^m and y_{ML} for $L_{ML(i)}^m$.

全記述関数 h について $h^L \leq h \leq h^U$ が成り立つことを, h は不完全記述関数 $\hat{h} = [h^L, h^U]$ に compatible であるといい, $h \prec [h^L, h^U]$ と表す¹²⁾.

n 入力論理関数 $f(x_1, \dots, x_n)$ に関して $x_q = (x_{i_1}, \dots, x_{i_q})$ ($q \leq n$) によるスムーズ演算 S_{x_q} およびコンセンサス演算 C_{x_q} は, それぞれ

$$S_{x_q} f = S_{x_{i_1}} S_{x_{i_2}} \cdots S_{x_{i_q}} f \quad (2)$$

$$C_{x_q} f = C_{x_{i_1}} C_{x_{i_2}} \cdots C_{x_{i_q}} f \quad (3)$$

で表される. ただし, 演算子 $S_{x_{i_j}}$, $C_{x_{i_j}}$ は,

$$S_{x_{i_j}} f = f|_{x_{i_j}} + f|_{\overline{x_{i_j}}} \quad (4)$$

$$C_{x_{i_j}} f = f|_{x_{i_j}} \cdot f|_{\overline{x_{i_j}}} \quad (5)$$

で定義される. $C_{x_q} f$ は全称作用とも呼ばれ,

$$C_{x_q} f = \prod_{a \in B^q} f|_{x_q=a} \quad (6)$$

と書けることから, 次の補題が成立する.

補題 1 n 入力変数ベクトル $x = (x_1, \dots, x_n)$ に含まれる各変数を, $x_p = (x_{i_1}, \dots, x_{i_p})$ と $x_q = (x_{j_1}, \dots, x_{j_q})$ に分割する ($p + q = n$). このとき, $\forall x_q \in B^q : f|_{x_p=a} = 1$ となる x_p への割当て a に関する必要十分条件は, $C_{x_q} f|_{x_p=a} = 1$ である. また, そのような x_p への割当て a が存在するための必要十分条件は, $C_{x_q} f \neq 0$ である. \square

信号線欠落を想定した組合せ箇所 $L_{ML(i)}^m$ に対して, 真理値変数ベクトル v^m と欠落入力変数 y_{ML} が図3のように割り当てられるとき, すべての外部出力が機能仕様と一致する条件は,

$$c = \prod_{j=1}^p (f_{gj}(x, v^m, y_{ML}) \equiv f_{sj}(x)) \quad (7)$$

で表される. この c を一致関数と呼ぶ. ここで,

$$c_{u0}(x, v^m) = \overline{c}|_{y_{ML}} \quad (8)$$

$$c_{u1}(x, v^m) = \overline{c}|_{\overline{y_{ML}}} \quad (9)$$

とすると, 一致関数は次のように変形できる.

$$c(x, v^m, y_{ML}) = y_{ML} \cdot \overline{c_{u0}} + \overline{y_{ML}} \cdot \overline{c_{u1}} \quad (10)$$

4.2 信号線欠落誤りに対応した真理値シミュレーション

真理値シミュレーション⁸⁾では, LUT 内部を真理値変数によって表現し, 外部入力に誤り追跡入力¹¹⁾を与えた際の外部出力関数を求める. 具体的には, 各誤り追跡入力 α_q ($1 \leq q \leq n_\alpha$) に対して, 各外部出力の信号値をシミュレーションにより求める. このとき, j 番目の外部出力の信号値は真理値変数 v^m と誤り追跡入力に含まれる X , そして欠落入力変数 y_{ML} によって $f_{gj}(\alpha_q, v^m, y_{ML})$ と表される.

誤り追跡入力 α_q に対する一致関数 c_{α_q} は,

$$c_{\alpha_q} = C_X S_{y_{ML}} c'_{\alpha_q}, \quad (11)$$

$$c'_{\alpha_q} = \prod_{j=1}^p (f_{gj}(\alpha_q, v^m, y_{ML}) \equiv f_{sj}(\alpha_q)) \quad (12)$$

と表される. 式 (11) では, c'_{α_q} に対して y_{ML} に関するスムーズ演算を行うことにより, y_{ML} が 0 あるいは 1 のいずれかの場合に f_{gj} と f_{sj} が一致する可能性を求める. 以降は従来手法⁸⁾と同様に, c_{α_q} をもとに誤り追跡入力 $\alpha_1 \sim \alpha_q$ に関する共通一致関数を求め, 組合せ箇所を絞り込む.

4.3 信号線欠落誤りに対応した LUT 機能修正

信号線欠落誤りの修正は, 次の 3 段階の処理により行う.

- (1) 信号線欠落誤りの存在判定
- (2) 欠落信号線候補の絞り込み
- (3) 欠落信号線の条件判定と LUT 機能修正

以下, 4.3.1 項において手法の適用例を通して処理概要を示した後, 上記 (1)~(3) の各処理について 4.3.2~4.3.4 の各項で述べる.

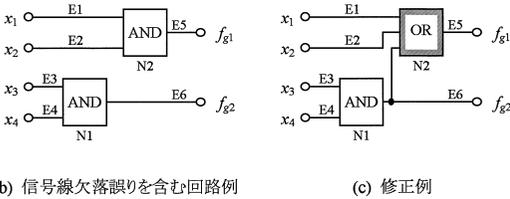
4.3.1 信号線欠落誤りの修正例

信号線欠落誤りに対応した LUT 機能修正の例を図4に示す. 多重度 $m = 1$, 信号線を想定した組合せ箇所 $L_{ML(1)}^1 = \{N2\}$, すなわち $N2$ に信号線欠落誤りを想定する場合を考える. まず, 欠落入力変数 y_{ML} , および修正箇所 $N2$ における真理値変数ベクトル $v_1 = (v_{1,0}, \dots, v_{1,7})$ を図5のように割り当てる. この後, 真理値シミュレーションを行う. 8 個の誤り追跡入力 $(x_1, x_2, x_3, x_4) = (X, 0, 0, 0), (X, 0, 0, 1), (X, 0, 1, 0), (0, X, 0, 0), (0, X, 0, 1), (0, X, 1, 0), (0, 0, X, 1), (0, 0, 1, X)$ に対する真理値シミュレーションの結果, 共通一致関数 $c_{\alpha_{com}}^8 = (\overline{v_{1,0}} + \overline{v_{1,1}})(v_{1,0} + v_{1,1})(v_{1,2} + v_{1,3})(v_{1,4} + v_{1,5})(v_{1,6} + v_{1,7})$ が得られる. $c_{\alpha_{com}}^8 \neq 0$ であるので, $L_{ML(1)}^1$ に対し

$$f_{g1} = x_1 + x_2 + x_3x_4$$

$$f_{g2} = x_3x_4$$

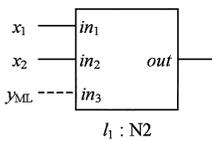
(a) 機能仕様



(b) 信号線欠落誤りを含む回路例 (c) 修正例

図 4 信号線欠落誤りの修正例

Fig. 4 Example of rectification of a single line error.



in_1	in_2	in_3	out
0	0	0	$v_{1,0}$
0	0	1	$v_{1,1}$
0	1	0	$v_{1,2}$
0	1	1	$v_{1,3}$
1	0	0	$v_{1,4}$
1	0	1	$v_{1,5}$
1	1	0	$v_{1,6}$
1	1	1	$v_{1,7}$

(a) 欠落入力変数 y_{ML} の割当て (b) 真理値変数の割当て

図 5 N2 に対する変数割当て

Fig. 5 Variable assignment for N2.

て信号線誤りを想定した LUT 機能修正に進む。また、共通一致関数からは $v_{1,1} = \overline{v_{1,0}}$ の関係が導かれるため、真理値変数 $v_{1,1}$ を $\overline{v_{1,0}}$ で置き換える。

信号線欠落誤りの修正では、修正箇所 N2 に接続する信号線 w と N2 の真理値割当て t_1 を決定する。具体的な修正解として、図 4(c) に示すように、 $w = E6$ と OR 機能に対応する真理値割当てベクトル $t_1 = (0, 1, 1, 1, 1, 1, 1, 1)$ が考えられる。

提案手法では、まず (1) 信号線欠落誤りの存在判定、を行う。すなわち、信号線欠落を想定した組合せ箇所 $L_{ML(1)}^1 = \{N2\}$ に対して解が存在する可能性を評価する。もし任意の論理を実現可能な信号線を N2 に接続したとしても回路の機能仕様を満足できないならば、この段階で $L_{ML(1)}^1 = \{N2\}$ を棄却する。存在判定のための評価式については、4.3.2 項で詳述する。

$L_{ML(i)}^m$ におけるすべての修正解を求めるために、LUT 回路内の各信号線に対して信号線欠落誤りを想定した LUT 機能修正を行う。しかし、全信号線を対象とするのは効率的ではない。そこで、(2) 欠落信号線候補の絞り込み、を行う。図 4(b) の例の場合、信号線 E5 を N2 に接続すると回路にループが生じるため、E5 を対象とする必要はない。また、N2 に接続する信号線の論理は x_3, x_4 を依存変数として持つ必要があることが容易に分かる。よって、E1, E2, E3, E4 も

解とはなりえず、E6 だけが欠落信号線候補となる。

欠落信号線候補の絞り込み後、残存する信号線 E6 に対して、(3) 欠落信号線の条件判定と LUT 機能修正、を行う。欠落信号線の論理 g_{ML} に関して解が存在するための条件は、N2 の真理値割当て t_1 を用いて、 $c_{u1}|_{t_1} \leq g_{ML} \leq \overline{c_{u0}}|_{t_1}$ と表せる。たとえば、真理値割当てベクトル $t_1 = (0, 1, 1, 1, 1, 1, 1, 1)$ に対しては、 $\overline{x_1} \overline{x_2} x_3 x_4 \leq g_{ML} \leq x_1 + x_2 + x_3 x_4$ となる。よって、着目する信号線 E6 が、前述の条件を満たすか否かを一致関数に基づいて評価するとともに、対応する N2 の真理値割当てを求める。4.3.4 項で詳述するが、提案手法では真理値割当てに対応する特徴関数を計算することによって解を求める。

4.3.2 信号線欠落誤りの存在判定

LUT 機能修正に際しては、処理の効率化のために、まず信号線欠落誤りの存在判定を行う。

まず、 y_{ML} に接続する信号線 w_{ML} を特定しない段階で、式 (10) で表される一致関数 c が満たすべき条件を求める。具体的には、一致関数における y_{ML} を独立変数、すなわち x および v^m に依存しない変数と仮定したうえで、 $L_{ML(i)}^m$ における真理値割当てベクトル t^m の修正法が存在するか否かを判定する。一致関数 c の論理値は、解となる真理値割当てベクトル $v^m = t^m$ のもとでは y_{ML} の値を適宜決定することにより、任意の x に対して 1 となる。よって、求める条件は、

$$\exists v^m \forall x \exists y_{ML} : c(x, v^m, y_{ML}) = 1 \quad (13)$$

$$\Leftrightarrow \exists v^m \forall x : S_{y_{ML}} c = 1 \quad (14)$$

となる。さらに補題 1 を適用すると、式 (14) は、

$$s_{ML}(v^m) = C_x S_{y_{ML}} c \neq 0 \quad (15)$$

と表される。この s_{ML} は、解となりうる真理値割当てベクトル t^m を非明示的に表す特徴関数となる。ここで、本論文で扱う論理診断問題においては、接続すべき信号線 w_{ML} を LUT 回路 G に含まれるノード $v \in V$ の出力から選択することに注意する。この制約を外せば、式 (15) は解の存在に関する必要十分条件となる。

図 4 の例では、 $s_{ML} = (v_{1,2} + v_{1,3})(v_{1,4} + v_{1,5})(v_{1,6} + v_{1,7}) \neq 0$ となり、条件を満足する。このとき、解となりうる真理値割当ては s_{ML} を充足する真理値割当てに制限される。

4.3.3 欠落信号線候補の絞り込み

以下では、回路構造や論理関数の性質を利用することで、解の存在しない信号線を削除する方法を提案する。欠落信号線候補を絞り込むために、次の条件を適用する。

条件 1 修正後にループを生じない。

条件 2 対象となる信号線の論理関数が^g、修正のために必須の入力変数すべてに依存する。

条件 1 より、信号線欠落誤りを想定する修正箇所から外部出力へ至るパス上の信号線は候補としない。また条件 2 について、ある入力変数 x_i が欠落信号線の論理関数に必須の依存変数であるならば、信号線の探索範囲は外部入力 x_i からいずれかの外部出力に至るパス上のすべてのノード集合からなる $TFO(x_i)$ の要素に限定できる。また、必須変数が複数存在する場合には、各必須外部入力 x_i に対する $TFO(x_i)$ の共通部分が探索範囲となる。以下では、欠落信号線想定箇所に接続する信号線の論理 $g_{ML}(x, v^m)$ に関して、 v^m の割当て t^m にかかわらず必須となる入力変数の集合 $ESML$ (Essential Support for Missing Line) を求める方法について述べる。

まず、 $v^m = t^m$ のもとで欠落信号線の論理 $g_{ML}|_{t^m}$ に必須となる入力変数の集合 $ESML(t^m)$ を求める。ここで、 $ESML(t^m)$ は、解となる任意の信号線が実現する論理 $g_{ML}|_{t^m}$ に関して $x_i \in sup(g_{ML}|_{t^m})$ となる x_i の集合と定義する。

式 (10) より $v^m = t^m$ のもとで一致関数は、 $c_{t^m} = c|_{t^m} = y_{ML} \cdot \overline{c_{u0}}|_{t^m} + \overline{y_{ML}} \cdot \overline{c_{u1}}|_{t^m}$ と表される。ここで、2 つの関数 $\overline{c_{u0}}|_{t^m}$ 、 $\overline{c_{u1}}|_{t^m}$ は、

$$c_{u0}|_{t^m} = \overline{c_{t^m}}|_{y_{ML}} \quad (16)$$

$$c_{u1}|_{t^m} = \overline{c_{t^m}}|_{\overline{y_{ML}}} \quad (17)$$

と表され、それぞれ欠落入力変数 y_{ML} が 1 あるいは 0 である場合に、外部出力信号値が機能仕様と一致しない外部入力割当ての集合を表す。すなわち、 $c_{u0}|_{t^m}$ は、 $y_{ML} = 0$ を、 $c_{u1}|_{t^m}$ は $y_{ML} = 1$ を割り当てなければならない外部入力割当ての条件を表す。よって、回路内のある信号線の論理 g_{ML} に対して真理値割当てベクトル t^m が解となるための必要十分条件は、

$$g_{ML}|_{t^m} \prec [c_{u1}|_{t^m}, \overline{c_{u0}}|_{t^m}] \quad (18)$$

となる。

$ESML(t^m)$ を求めるために、次の補題を利用する。

補題 2 3 つの論理関数 $f_1(x)$ 、 $f_2(x)$ 、 $g(x)$ が $g \prec [f_1, \overline{f_2}]$ 、または $g \prec [f_2, \overline{f_1}]$ を満たすとき、

$$f_1|_{x_i} \cdot f_2|_{x_i} + f_1|_{x_i} \cdot \overline{f_2|_{x_i}} \neq 0 \Rightarrow x_i \in sup(g) \quad (19)$$

が成り立つ。

(証明) 論理関数 $f_1(x)$ 、 $f_2(x)$ 、 $g(x)$ に関して、 $g \prec [f_1, \overline{f_2}]$ とすると、

$$f_1|_{x_i} \leq g|_{x_i} \leq \overline{f_2|_{x_i}} \quad (20)$$

$$f_1|_{x_i} \leq g|_{x_i} \leq \overline{f_2|_{x_i}} \quad (21)$$

が成り立つ。式 (20) より $f_1|_{x_i} \leq g|_{x_i}$ 、式 (21) より $f_2|_{x_i} \leq \overline{g|_{x_i}}$ であるので、

$$f_1|_{x_i} \cdot f_2|_{x_i} \leq g|_{x_i} \cdot \overline{g|_{x_i}} \quad (22)$$

となる。同様に、式 (20) より $f_2|_{x_i} \leq \overline{g|_{x_i}}$ 、式 (21) より $f_1|_{x_i} \leq g|_{x_i}$ であるので、

$$f_1|_{x_i} \cdot f_2|_{x_i} \leq \overline{g|_{x_i}} \cdot g|_{x_i} \quad (23)$$

が成り立つ。よって、式 (22) および式 (23) より、次式が成り立つ。

$$f_1|_{x_i} \cdot f_2|_{x_i} + f_1|_{x_i} \cdot \overline{f_2|_{x_i}} \leq g|_{x_i} \cdot \overline{g|_{x_i}} + \overline{g|_{x_i}} \cdot g|_{x_i} \quad (24)$$

(左辺) $\neq 0$ であるので、式 (24) は、

$$0 \leq g|_{x_i} \oplus g|_{x_i} \quad (25)$$

と表せる。ゆえに、

$$g|_{x_i} \neq g|_{x_i} \quad (26)$$

が成り立つので、 $x_i \in sup(g)$ である。

論理関数 $f_1(x)$ 、 $f_2(x)$ 、 $g(x)$ が^g、 $g \prec [f_2, \overline{f_1}]$ を満足する場合も同様である。□

補題 2 を式 (18) に適用すると、

$$c_{u0}|_{t^m x_i} \cdot c_{u1}|_{t^m x_i} + c_{u0}|_{t^m x_i} \cdot \overline{c_{u1}|_{t^m x_i}} \Rightarrow x_i \in sup(g_{ML}|_{t^m}) \quad (27)$$

となる。ここで、 $S_{x_i} c_{u0}$ 、 c_{u1} は、それぞれ

$$S_{x_i} c_{u0} = c_{u0}|_{x_i} + c_{u0}|_{\overline{x_i}} \quad (28)$$

$$c_{u1} = x_i \cdot c_{u1}|_{x_i} + \overline{x_i} \cdot c_{u1}|_{\overline{x_i}} \quad (29)$$

と表せるので、関数 $S_{x_i} c_{u0} \cdot c_{u1}$ は、

$$S_{x_i} c_{u0} \cdot c_{u1} = x_i (c_{u0}|_{x_i} \cdot c_{u1}|_{x_i}) + \overline{x_i} (c_{u0}|_{\overline{x_i}} \cdot c_{u1}|_{\overline{x_i}}) \quad (30)$$

のように変形できる。よって、

$$S_{x_i} c_{u0}|_{t^m} \cdot c_{u1}|_{t^m} \neq 0 \Leftrightarrow x_i \in ESML(t^m) \quad (31)$$

となる。

図 4 の例において、真理値割当てベクトル $t_1 = (0, 1, 1, 1, 1, 1, 1, 1)$ に対して補題 2 を適用すると、 $f_1 = c_{u1}|_{t^m} = \overline{x_1} \overline{x_2} x_3 x_4$ 、 $f_2 = c_{u0}|_{t^m} =$

$x_1 + x_2 + x_3x_4 = \overline{x_1}\overline{x_2}(\overline{x_3} + \overline{x_4})$ とおける $x_1 \sim x_4$ の各変数に対して, $f_1|_{x_i} \cdot f_2|_{x_i} + f_1|_{\overline{x_i}} \cdot f_2|_{\overline{x_i}}$ を計算すると次のようになる.

$$f_1|_{x_1} \cdot f_2|_{x_1} + f_1|_{\overline{x_1}} \cdot f_2|_{\overline{x_1}} \equiv 0 \quad (32)$$

$$f_1|_{x_2} \cdot f_2|_{x_2} + f_1|_{\overline{x_2}} \cdot f_2|_{\overline{x_2}} \equiv 0$$

$$f_1|_{x_3} \cdot f_2|_{x_3} + f_1|_{\overline{x_3}} \cdot f_2|_{\overline{x_3}} = \overline{x_1}\overline{x_2}x_4 \neq 0$$

$$f_1|_{x_4} \cdot f_2|_{x_4} + f_1|_{\overline{x_4}} \cdot f_2|_{\overline{x_4}} = \overline{x_1}\overline{x_2}x_3 \neq 0$$

よって, $ESML(0, 1, 1, 1, 1, 1, 1) = \{x_3, x_4\}$ を得る.

続いて, t^m にかかわらず必須となる変数の集合 $ESML$ を求める. $ESML$ は特徴関数 $s_{ML}(t^m) = 1$ を満足するすべての t^m に関して, $x_i \in ESML(t^m)$ となる変数 x_i の集合である. よって, 式 (31) より,

$$\forall t^m \text{ s.t. } s_{ML}(t^m) = 1 \exists x_i : S_{x_i}c_{u0} \cdot c_{u1} = 1 \quad (33)$$

が成り立てば, $x_i \in ESML$ であるので,

$$S_x(S_{x_i}c_{u0} \cdot c_{u1}) \geq s_{ML} \Leftrightarrow x_i \in ESML \quad (34)$$

となる.

図 4 の例における欠落信号線の絞り込みでは, 条件 1 により信号線 E5 が候補から削除され, 条件 2 により信号線 E1, E2, E3, E4 が削除される.

4.3.4 欠落信号線の条件

4.3.2 項では, 一致関数 c に対する必要条件を求めた. ここでは, 修正の可能性がある $L_{ML(i)}^m$ に対して, 信号線 w_{ML} が欠落信号線であるか否かを判定する必要十分条件を求めることにより, 信号線欠落誤りを考慮した LUT 機能修正を行う.

y_{ML} に接続される信号線 w_{ML} の論理を $g_{ML}(x, v^m)$ とする. 4.3.1 項では, 欠落信号線が満たすべき条件を具体的に示したが, 論理が $g_{ML}(x, v^m)$ である信号線 w_{ML} がこの条件を満たすか否かは, 一致関数 c において $y_{ML} = g_{ML}$ を代入することにより求めることができる. すなわち, $L_{ML(i)}^m$ における真理値割当てベクトル $v^m = t^m$ に対して,

$$c(x, t^m, g_{ML}(x, t^m)) \equiv 1 \quad (35)$$

が成り立つとき, w_{ML} は欠落信号線の条件を満足し, $E_{ML(i)}^m = (L_{ML(i)}^m, t^m, w_{ML})$ が求める解の 1 つである. よって, 解が存在するための必要十分条件は,

$$\exists v^m \forall x : c(x, v^m, g_{ML}(x, v^m)) = 1 \quad (36)$$

となる. ここで,

$$c_{ML}(v^m) = C_x c(x, v^m, g_{ML}(x, v^m)) \quad (37)$$

とし, 補題 1 を適用すると, 式 (36) は,

$$c_{ML}(v^m) \neq 0 \quad (38)$$

と表される. この c_{ML} は, 解となる真理値割当てベクトルの集合を表す特徴関数となる. すなわち, 信号線 w_{ML} に対して $c_{ML}(t^m) = 1$ を満たす真理値割当て

表 1 ISCAS'85 ベンチマーク回路
Table 1 ISCAS'85 benchmark circuits.

Circuit	#Gate	#LUT	#PI	#PO
C432	160	129	36	7
C499	202	132	41	32
C880	383	234	60	26
C1355	546	180	41	32
C1908	880	467	33	25
C3540	1,669	921	50	22
C5315	2,307	1,386	178	123

てベクトル t^m が求める解である.

図 4(b) の例について, 対象とする信号線が E6, すなわち $g_{ML} = x_3x_4$ である場合, $c_{ML} = \overline{v_{1,0}}v_{1,2}v_{1,3}v_{1,4}v_{1,5}v_{1,6}v_{1,7} \neq 0$ が得られる. よって, N2 の真理値割当ては, c_{ML} を充足する真理値割当て $t_1 = (0, 1, 1, 1, 1, 1, 1)$ となる. この結果, 図 4(c) に示す修正法が求められた.

5. 実験結果

提案した EXL_{SL} 法について, C 言語 (gcc ver. 2.95.2) を用いて IBM-PC/AT 互換機 (CPU: Pentium III 600 MHz) の Linux 上に実装するとともに, 実験を行った.

実験には, 表 1 に示す ISCAS'85 ベンチマーク回路¹³⁾ に対して最大 3 入力の LUT にマッピングした回路を対象とした. これらの LUT 回路に無作為に設計誤りを 1 個から 3 個挿入した回路例を用いた. いずれの回路例に対しても設計誤りのうち 1 つは信号線欠落誤りとし, 修正後の LUT が 4 入力になることを許した. 1 種類の回路に対して挿入した誤りの個数ごとに 20 例, 合計 $3 \times 20 \times 7 = 420$ 例について実験を行った.

論理診断においては, BDD のノード数を 800 万に制限した. この制限によって正しい外部出力関数の表現が不可能となった C6288 については実験対象から除いた. また, C2670, C7552 については, 外部出力関数の BDD 表現は可能であったが, 論理関数処理においてノード制限を超えたため, 実験対象から除いた. また, 2 個の誤りを含む C5315 のうち 1 例でノード数制限を超えて修正不可能となったが, その他の例では, すべて修正可能であることを確認した. 修正可能な回路例のうち 30 例の回路については, LUT 機能の修正のみで機能仕様が満足されたため, 以下に示す結果から除いた.

表 2 に組合せ箇所と修正解の個数に関する結果を示す. 表中の結果は全回路例に関する結果の幾何平均である. 本論文では, ある組合せ箇所 L^m のうちの

表 3 処理時間に関する実験結果 [s]
Table 3 Processing times [s].

回路名	Ave.			Max.		
	m = 1	m = 2	m = 3	m = 1	m = 2	m = 3
C432	0.5	1.0	6.5	3.7	5.4	603
C499	11.4	80.9	518.2	29.6	4,091.0	132,480
C880	4.5	6.1	9.3	349.2	422.3	258
C1355	9.9	48.5	159.1	23.8	7,275.8	50,727
C1908	3.5	11.2	150.2	26.2	75.4	8,354
C3540	26.4	42.1	109.4	109.5	248.7	26,112
C5315	4.4	5.7	21.4	15.6	57.6	3,417
Ave.	5.2	13.0	56.1	32.8	256.3	6,945

表 2 組合せ箇所数に関する実験結果
Table 2 Numbers of error location sets.

m	#L ^m			#E _T ^m
	Total	aft. Screening	Final	
1	330	3.0	1.5	15.6
2	5.4 × 10 ⁵	6.4	1.8	30.2
3	6.0 × 10 ⁸	17.2	2.0	59.8

表 4 欠落信号線候補数に関する結果
Table 4 Numbers of missing line candidates.

m	欠落信号線候補数			修正可能信号線数
	Total	回路構造を利用した絞り込み後	ESMLによる絞り込み後	
1	2,662	2,492	296	6.4
2	6,000	5,362	305	7.2
3	38,748	32,931	2,511	9.4

1 つの修正箇所 l_i に信号線欠落誤りを想定する場合の修正法を提案した。表中の“Total”は、LUT 数 n に対する多重度 m 以下の組合せ箇所の総数 $\sum_{k=1}^m {}_n C_k = {}_n C_1 + \dots + {}_n C_m$ の値を示す。“aft. Screening”は入力パターンを用いた組合せ箇所の削減を行った後の組合せ箇所数，“Final”は LUT 機能修正後の個数、すなわち解が存在する組合せ箇所の個数を表す。また、“#E_T^m”は、最終的に得られた修正解の個数を表すが、真理値割当てベクトルのみ異なる解も別解として個数を求めた。

処理時間に関する結果を表 3 に示す。“Ave.”および“Max.”はそれぞれ処理時間の幾何平均、最大値を表す。 m は、修正された設計誤りの多重度を示す。実験結果より、回路例ごとの幾何平均では $m = 3$ の場合でも 60s 程度に収まっている。

表 4 に欠落信号線候補数に関する結果を示す。ただし、表中の数値は全回路例に関する結果の算術平均である。実験では条件 1 に基づく回路構造を利用した候補の絞り込みを行ったうえで、4.3.3 項で提案した必須変数集合に基づく欠落信号線候補の絞り込み手法を適用した。表中の“欠落信号線候補”の欄に欠落信号線候補の総数“Total”と各処理を適用した後の候補数を示す。また、“修正可能信号線数”は解の存在した候補の個数を表す。実験結果より、2 種類の絞り込み手法を適用することにより、92%の候補数削減が可能となった。その内訳は、回路構造を利用した絞り込み(条件 1)が 11%、必須変数集合に基づく絞り込み(条件 2)が 81%である。この結果から、必須変数集

合に基づく絞り込みの有効性が示された。

提案手法では最大 2,300 ゲートの組合せ回路に対する論理診断が実現されている一方で、BDD ノード数の制限で修正が不可能であった回路例が存在した。C2670 や C7552 に関しては外部出力関数の BDD 表現は可能であるため、すべての解を求めることはできないものの、次のような実装上の工夫により対応可能であると考えられる。すなわち、ノード数が制限を超えた段階で着目している候補に対する処理を終了し、次の候補に対する処理を行うことにより解を求める。しかし、さらに大規模な回路に対応するためには、分割診断処理の実現が期待される。

6. ま と め

本論文では、LUT で表現された組合せ回路に含まれる設計誤りの修正に関して、信号線誤りの修正を可能とする EXL_{SL} 法を提案した。本手法を計算機上に実装し実験を行った結果、信号線欠落誤りの修正に有効であることを確認した。

今後の課題として、本手法に関する具体的な設計変更要求の例への適用可能性について検証することがあげられる。

謝辞 本研究の一部は、文部科学省科学研究費補助金(特別研究員奨励費)による。

参 考 文 献

1) Watanabe, Y. and Brayton, R.K.: Incremental

- tal synthesis for engineering changes, *ICCD-91*, pp.40–43 (1991).
- 2) Fujita, M., Tamiya, Y., Kukimoto, Y. and Chen, K.C.: Application of Boolean unification to combinational logic synthesis, *ICCAD-91*, pp.510–513 (1991).
 - 3) Abadir, M.S., Ferguson, J. and Kirkland, T.: Logic design verification via test generation, *IEEE Trans. CAD*, Vol.7, No.1, pp.138–148 (1988).
 - 4) Tomita, M., Yamamoto, T., Sumikawa, F. and Hirano, K.: Rectification of multiple logic design errors in multiple output circuits, *31st DAC*, pp.212–217 (1994).
 - 5) Boppana, V., Mukherjee, R., Jain, J. and Fujita, M.: Multiple error diagnosis based on Xlists, *36th DAC*, pp.660–665 (1999).
 - 6) Kukimoto, Y. and Fujita, M.: Rectification method for lookup-table type FPGA's, *ICCAD-92*, pp.54–61 (1992).
 - 7) Pomeranz, I. and Reddy, M.: On error correction in macro-based circuits, *IEEE Trans. CAD*, Vol.16, No.10, pp.1088–1100 (1997).
 - 8) 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).
 - 9) Akers, S.B.: Binary decision Diagrams, *IEEE Trans. Comput.*, Vol.27, No.6, pp.509–516 (1978).
 - 10) Bryant, R.E.: Graph-based algorithms for Boolean function manipulation, *IEEE Trans. Comput.*, Vol.C-35, No.8, pp.677–691 (1986).
 - 11) Tomita, M., Suganuma, N. and Hirano, K.: Pattern generation for locating design errors, *IEICE Trans.*, Vol.E77-A, No.5, pp.881–893 (1994).
 - 12) Sawada, H., Suyama, T. and Nagoya, A.: Logic synthesis for Look-Up Table based FPGAs using functional decomposition and support minimization, *ICCAD-95*, pp.353–358 (1995).
 - 13) Brglez, F. and Fujiwara, H.: A neutral netlist of 10 combinational benchmark circuits and a target translation in FORTRAN, *ISCAS-85* (1985).

(平成 14 年 10 月 23 日受付)

(平成 15 年 3 月 4 日採録)



井上 宏 (学生会員)

1976 年生。1999 年神戸大学工学部電気電子工学科卒業。2001 年同大学大学院修士課程修了。同年同大学大学院博士課程進学。日本学術振興会特別研究員。論理診断の研究に従事。



黒木 修隆

1965 年生。1990 年神戸大学工学部電子工学科卒業。1995 年同大学大学院自然科学研究科博士課程修了。同年同大学工学部電気電子工学科助手。工学博士。画像処理, LSI 設計に関する研究に従事。IEEE, 電子情報通信学会, 電気学会会員。



沼 昌宏 (正会員)

1960 年生。1983 年東京大学工学部精密機械工学科卒業。1985 年同大学大学院修士課程修了。同大学助手を経て 1989 年同大学講師。工学博士。1990 年 5 月より神戸大学大学院自然科学研究科講師。1995 年同大学工学部電気電子工学科助教授。1996 年文部省在外研究員として米国カリフォルニア大学サンタバーバラ校に派遣。主に LSI CAD, アクセラレータ, 画像処理に関する研究に従事。IEEE, ACM, 電子情報通信学会会員。



山本 啓輔

1939 年生。1962 年神戸大学工学部電気工学科卒業。同年松下電器産業(株)入社。主としてテレビ受信機の開発, 研究に従事。2000 年神戸大学工学部電気電子工学科教授。2001 年同大学共同研究開発センター教授。工学博士。放送と通信の融合, 画像処理, LSI CAD に関する研究に従事。映像情報メディア学会会員。