真理値シミュレーションに基づくLUT論理診断手法

沼 昌宏[†] 井上 宏[†,††] 皆見利行[†]
黒木修隆[†] 山本啓輔[†]

LUT（Look-Up Table）で表現された回路に含まれる多重設計誤りを自動的に修正する手法の1つであるEXL（Extended X-algorithm for LUT-based circuits）法[1)]では，処理時間の問題により，多重度3を超える設計誤りの修正は困難であった．本論文では，修正可能な設計誤りの多重度を向上させるために，LUT内部を真理値変数と呼ぶ変数で表現し，誤り追跡入力を外部入力に与えた際の外部出力関数を求める真理値シミュレーションによって，考慮すべき組合せ箇所を効率良く絞り込むとともに，その結果を利用してLUT機能修正を短時間で行う論理診断手法EXL_{TV}を提案する．3個の設計誤りを含む回路に対する実験の結果，LUT機能修正において考慮すべき組合せ箇所の数を真理値シミュレーションによって従来の3.1%にまで削減可能となった．さらに，その結果をもとに可能ならば真理値変数を論理定数，または他の真理値変数で置換することで，LUT機能修正に必要なBDDノード数を従来の23%に削減した．以上の効果により，3個の設計誤りを修正する処理全体に要する時間を約7分の1に短縮可能となった．また，従来は修正が困難であった4個の設計誤りを含む回路に対しても，現実的な処理時間での修正が可能となった．

An LUT-based Multiple Error Diagnosis Technique Using Symbolic Simulation with Truth Variables

Masahiro Numa,[†] Hiroshi Inoue,[†,††] Toshiyuki Kaimi,[†]
Nobutaka Kuroki[†] and Keisuke Yamamoto[†]

Rectification of four or more design errors in LUT-based circuits based on the EXL-algorithm[1)] has been difficult due to processing time. For breaking through this problem, we present an error diagnosis technique: EXL_{TV} employing TV simulation, a novel symbolic simulation method using truth variables to represent LUT contents. Based on the simulation result, EXL_{TV} reduces the number of error location sets efficiently. In addition, the result is also used to reduce the number of BDD nodes needed for rectification of LUT contens. The experimental results for LUT-based circuits including 3 design errors demonstrate that TV simulation has reduced the number of combinations of LUT's to be rectified to 3.1% of conventional one. The number of BDD nodes needed for rectification is also reduced to 23% by replacing truth variables by Boolean constants or other ones based on the results of TV simulation. EXL_{TV} has shortened the processing times for rectification of 3 design errors to about one-seventh of EXL. Moreover, EXL_{TV} has performed rectification of four design errors within paractical processing time.

1. はじめに

論理回路の設計時間短縮のために，論理合成ツールを利用した設計が一般的となっている．しかし，合成された回路がタイミング等の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．変更にと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，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に限らず，仕様変更（ECO: Engineering Change Order）に対応したインクリメンタル合成[2),3)]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．

LUT（Look-Up Table）は真理値表により任意の機能を実現できる素子である．このLUTにより構成される回路（LUT回路）を対象とする論理診断をLUT論理診断と呼ぶ．素子の種類が限られるゲートレベ

† 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

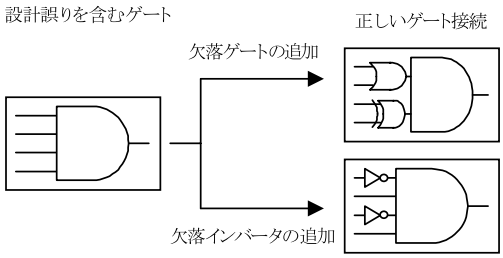

図 1 LUT 回路モデルを利用した複雑な設計誤りの修正
Fig. 1 Rectification of complex error types using LUT-based circuit model.

ル回路と比べて LUT 回路は高い自由度を持つため，LUT 論理診断手法はゲートレベル回路に対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が可能となる．よって，一般のセルベース LSI を対象とした，複合セルの置換を含むインクリメンタル合成への応用も考えられる．

一方で，LUT 回路が持つ高い自由度のために，ゲートレベル回路を対象とする場合と比べて論理診断はきわめて困難になる．たとえば，LUT 数 n の回路に対して多重度 m の誤りを想定して論理診断を行う場合，修正を考慮すべき LUT の組合せ，すなわち組合せ箇所の数は $O(n^m)$ となる．そのうえ，k 入力 LUT 1 個に対する修正方法は $2^{2^k}-1$ 通りとなり，$k=4$ に対して $t=65,535$ にも及ぶ．もし処理の初期段階から LUT 内部の真理値割当て，すなわち修正方法まで考慮するならば，$O(n^m t^m)$ の可能性について調べる必要性が生じるので現実的ではない．この結果，LUT 回路に対する論理診断については，限られた手法[4),5)]が提案されたのみであった．Kukimoto ら[4)] はブール関係に基づく手法を提案しているが，大規模回路には未対応である点，同一パス上に存在する複数の誤りには対応できない点に問題があった．一方で Pomeranz ら[5)] は，複数の入力組合せに対する回路の出力値と理想出力値の比較に基づく手法を提案しているが，外部入力数 n に対して最悪の場合は 2^n 個の組合せについて調べる必要があり，現実的ではなかった．

これらの問題点を解決するために，我々は，LUT 回路に含まれる多重設計誤りを自動的に修正する EXL 法[1)] を提案した．EXL 法は次の処理からなる．

i) 誤り追跡入力を用いた組合せ箇所の抽出と絞り込み
ii) 論理関数処理に基づく LUT 機能修正

まず i) により，誤り追跡入力[6)] と呼ぶ入力パターンを用いて，修正方法を特定しない組合せ箇所の段階で，その抽出と絞り込みを行う．残存する各組合せ箇所について，該当する各 LUT の機能を ii) の論理関数処理に基づいて修正する．EXL 法は，i) と ii) を組み合わせることにより，現実的な処理時間で 100%の誤り限定率を達成した．しかしながら，多重度の増加にともなって i) の後に残存する組合せ箇所の数が増えるため，ii) の処理に要する時間の増加が問題となっていた．このため，多重度 3 を超える設計誤りの診断は困難であった．

本論文では EXL 法をもとに，修正可能な設計誤りの多重度を向上させた論理診断手法 $\mathrm{EXL}_{\mathrm{TV}}$ を提案する．LUT 内部を真理値変数と呼ぶ変数で表現したうえで，誤り追跡入力に対する外部出力関数を求める真理値シミュレーションを行い，解となる可能性のない組合せ箇所を効率的に排除するとともに，その結果を LUT 機能修正にも利用する点を特徴とする．

2. EXL 法の概要

必要な用語を定義したうえで，従来の EXL 法[1)] について概要を述べる．なお本論文では，複数解の存在可能性と，手法の適用分野が設計誤りの修正に限定されない点を考慮して，文献 1) における誤り箇所，誤り種類，誤り候補，真の組合せ誤り，の各用語について，それぞれ修正箇所，真理値割当て，修正候補，修正解と呼ぶことにする．

2.1 用語の定義

定義 1 $\boldsymbol{f}_s = (f_{s1}, \cdots, f_{sp})$ および $\boldsymbol{f}_g = (f_{g1}, \cdots, f_{gp})$ を，n 入力変数ベクトル $\boldsymbol{x} = (x_1, \cdots, x_n)$ に対する p 出力ブール関数ベクトルとする．ここで f_s は機能仕様 S に，f_g は設計誤りを含む LUT 回路 G の外部出力関数に対応するとする．1 つのブール変数 X または $\overline{X}$ を含む入力ベクトル $\boldsymbol{\alpha} = (a_1, \cdots, a_{i-1}, X/\overline{X}, a_{i+1}, \cdots, a_n)$ 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[6)] と呼ぶ．論理診断に n_α 個の誤り追跡入力を適用するものとし，それぞれ $\boldsymbol{\alpha}_q$ $(1 \le q \le n_\alpha)$ で識別する． □

定義 2 $B = \{0, 1\}$，$T = \{0, 1, *\}$ とする．ここで，* はドントケアを表す．LUT は，真理値表で示される任意の機能を実現する論理素子である．その機能は，LUT 入力ベクトル $\boldsymbol{y} = (y_1, \cdots, y_k)$ に対する LUT 関数 $h : B^k \to B$ で表現される．本論文では，最大 4 入力 $(k \le 4)$ の LUT を扱うが，拡張は可能である．さらに，修正法に関する自由度を表すため，LUT

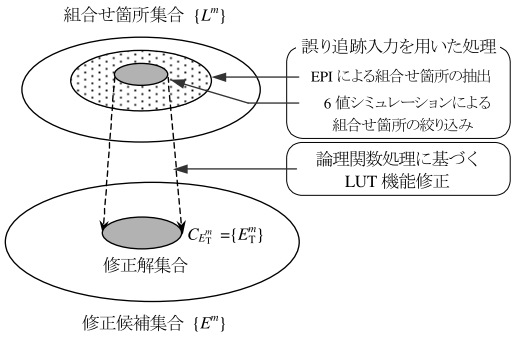

図 2 EXL 法の概要
Fig. 2 Overview of EXL-algorithm.

関数を不完全記述関数 $h : B^k \rightarrow T$ として扱う．また，この LUT と外部入力，外部出力，さらにそれらを接続するループのない結線で表される回路を，LUT 回路と呼ぶ． □

定義 3 対応する LUT 関数に誤りを想定する 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 $L^m = \{l_i \mid i = 1, \cdots, m\}$ と呼ぶ． □

定義 4 修正箇所 l_i における真理値表の割当てを，真理値割当て $\boldsymbol{t}_i = (t_{i,0}, \cdots, t_{i,2^{k_i}-1})$ と呼ぶ．$t_{i,0}, \cdots, t_{i,2^{k_i}-1}$ は，それぞれ各入力割当て $\boldsymbol{y}_a \in B^{k_i}$ の真理値に対応する． □

定義 5 修正候補 e_i は，修正箇所 l_i と真理値割当て t_i の組 $(l_i, \boldsymbol{t}_i)$ で表される．多重度 m の修正候補 $E^m = \{e_i \mid i = 1, \cdots, m\}$ は，m 個の修正候補からなる空でない集合である．ここで，すべての i $(1 \leq i \leq m)$ について，$e_i \in E^m$ の要素 l_i が L^m に含まれるとき，E^m を L^m における修正候補という．この場合，L^m は E^m に含まれるともいい，$L^m = L(E^m)$ と表す． □

定義 6 その修正によって機能仕様を満たす回路が得られる多重度 m の修正候補を，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． □

定義 7 修正解に含まれる組合せ箇所を，真の組合せ箇所 L_T^m と呼ぶ．1 つ以上の修正解 $E_T^m \in C_{E_T^m}$ に対して，$L_T^m = L(E_T^m)$ が成り立つ． □

EXL 法では，次の論理診断問題を対象とする．

論理診断問題：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． □

2.2 EXL 法による論理診断処理

従来の EXL 法[1] による論理診断処理の概要を**図 2** に示す．以下，各処理について説明する．

（1） EPI（Error Possibility Index）

論理診断においては，すべての不一致外部出力，すなわち機能仕様と異なる値を示す外部出力について，正しい出力値に変化するような修正法を求める必要がある．誤り可能性の指標 EPI は，不一致外部出力に対する可制御性を表す．ある誤り追跡入力 $\boldsymbol{\alpha}_q$ に対して発生する各不一致外部出力ごとに，各 LUT の EPI を設定する．いかなる修正を施しても着目する不一致外部出力の信号値を変化させない組合せ箇所に対しては，EPI の和が 1 未満となるよう文献 1) の手法に従い設定する．EPI の和が 1 以上となる組合せ箇所のみを候補として抽出する．

（2） 6 値シミュレーション

EPI を用いた判断は，外部出力に接続するゲートの出力値変化のみに着目している．次の段階では，変化後の外部出力値が機能仕様を満たす理想回路の出力値と等しくなる可能性に着目し，より厳密な判断を行う．網羅的な処理を回避するため，修正後の LUT 機能を特定せずに信号値を評価する．誤りを想定する各箇所について，可能性のある真理値割当てのそれぞれが存在する場合をまとめて評価するために，不定信号値 D と E を利用する．D は $\{0,1\}$，E は $\{0,1,X,\overline{X}\}$ のいずれかをとる可能性を表す．これらの信号値に，誤り追跡入力の 0，1，X，$\overline{X}$ の 4 値を加えた 6 値を用いて，外部出力まで X が伝搬する可能性の有無を判断する．理想回路で X が出力される外部出力について，実回路に関する 6 値シミュレーションの結果が X または E とならない場合は修正不可能と判断して，誤りを想定した組合せ箇所を削除する．

（3） 論理関数処理に基づく LUT 機能修正

6 値シミュレーションの結果，残存する組合せ箇所に対して，論理関数処理に基づいて LUT 機能の修正を試みる．診断の最終段階では論理関数処理を行うことで，一般に入力パターンに基づく手法では必要となる修正後の再検証が不要となる．その一方で，各外部入力および想定する組合せ箇所に含まれる LUT 出力を変数とした外部出力関数を求めるため，処理すべき組合せ箇所数が多い場合は時間を要する点が課題となっていた．

3. 提 案 手 法

前章で述べた EXL 法の課題を解決するには，6 値シミュレーションの後に残存する組合せ箇所を，さらに絞り込む必要がある．そこで，真理値シミュレーションによる組合せ箇所の絞り込みと，その結果を利用した LUT 機能修正法を提案する．

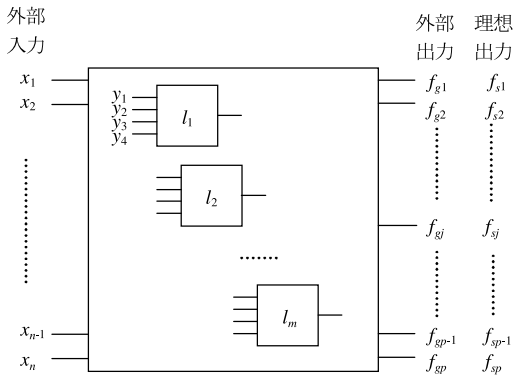

図 3 対象とする LUT 回路
Fig. 3 LUT-based circuit.

組合せ箇所に含まれる各 LUT 内部を真理値変数で表現したうえで，誤り追跡入力に対する外部出力値を記号シミュレーションの一種とも考えられる真理値シミュレーションによって求める．このとき外部入力に与えられる誤り追跡入力は，1 つの変数 X または $\overline{X}$ を除いてすべて論理定数（0 or 1）で表されるため，真理値シミュレーションにおいて扱うべき論理変数の数が一般的に少なく抑えられ，効率良く組合せ箇所を絞り込める点に特徴がある．

以下では，**図 3** に示す n 入力 p 出力の LUT 回路を考える．

3.1 用語の定義

定義 8 修正箇所 l_i における k_i 入力 LUT ($k_i \leq 4$) について，各入力割当て $\boldsymbol{y}_a \in B^{k_i}$ に対する出力値をそれぞれ $v_{i,0}, v_{i,1}, \cdots, v_{i,2^{k_i}-1}$ の変数で表すとき，この変数を真理値変数と呼ぶ．また，組合せ箇所 $L^m = \{l_1, \cdots, l_m\}$ 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 $\boldsymbol{v}^m = (v_{1,0}, \cdots, v_{1,2^{k_1}-1}, \cdots, v_{m,0}, \cdots, v_{m,2^{k_m}-1})$ と呼ぶ． □

定義 9 外部入力変数ベクトル $\boldsymbol{x} = (x_1, \cdots, x_n)$ と $\boldsymbol{v}^m$ に対して定義される外部出力関数 $f_{gj}(\boldsymbol{x}, \boldsymbol{v}^m)$ は，j 番目の外部出力の論理を表す．各外部出力関数 f_{gj} から成るベクトルを，外部出力関数ベクトル $\boldsymbol{f}_g = (f_{g1}, \cdots, f_{gp})$ と呼ぶ． □

定義 10 ある誤り追跡入力 $\boldsymbol{\alpha}_q = (a_{q,1}, \cdots, a_{q,i-1}, X/\overline{X}, a_{q,i+1}, \cdots, a_{q,n})$ を外部入力に与えた場合，全外部出力関数が機能仕様と一致する条件は，

$$c_{\boldsymbol{\alpha}_q} = c'_{\boldsymbol{\alpha}_q}\big|_{X=0} \cdot c'_{\boldsymbol{\alpha}_q}\big|_{X=1} = 1, \tag{3}$$

$$c'_{\boldsymbol{\alpha}_q} = \prod_{j=1}^{p} \left(f_{gj}(\boldsymbol{\alpha}_q, \boldsymbol{v}^m) \equiv f_{sj}(\boldsymbol{\alpha}_q)\right) \tag{4}$$

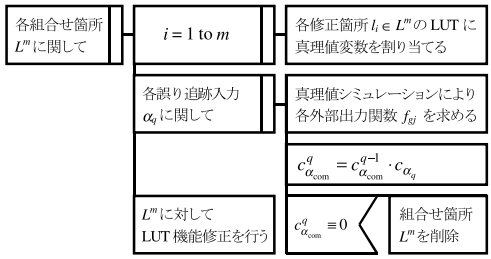

図 4 真理値シミュレーションによる組合せ箇所の絞り込み
Fig. 4 Screening error location sets with TV (truth variables) simulation.

で表される．この $c_{\boldsymbol{\alpha}_q}$ を，誤り追跡入力 $\boldsymbol{\alpha}_q (1 \leq i \leq q)$ に対する一致関数と呼ぶ． □

定義 11 各誤り追跡入力 $\boldsymbol{\alpha}_i$ に対する一致関数 $c_{\boldsymbol{\alpha}_i}$ の論理積

$$c^q_{\alpha_{\rm com}} = \prod_{i=1}^{q} c_{\boldsymbol{\alpha}_i} = c_{\boldsymbol{\alpha}_1} \cdot c_{\boldsymbol{\alpha}_2} \cdots c_{\boldsymbol{\alpha}_q} \tag{5}$$

を，$\boldsymbol{\alpha}_1$〜$\boldsymbol{\alpha}_q$ に関する共通一致関数と呼ぶ． □

補題 1 真の組合せ箇所 $L^m_{\rm T}$ に対応する真理値変数ベクトル $\boldsymbol{v}^m$ に関して，

$$c^q_{\alpha_{\rm com}} = 1 \tag{6}$$

を満たす割当てが存在する． □

この 式 (6) を，共通一致関数充足条件と呼ぶ．

定義 12 n 入力ブール関数 $f(x_1, \cdots, x_n)$ の $\boldsymbol{x}_r = (x_{i_1}, \cdots, x_{i_r})$ $(r < n)$ によるスムース演算 $S_{\boldsymbol{x}_r} f$ は，

$$S_{\boldsymbol{x}_r} f = S_{x_{i_1}} S_{x_{i_2}} \cdots S_{x_{i_r}} f, \tag{7}$$

$$S_{x_{i_j}} f = f\big|_{x_{i_j}=0} + f\big|_{x_{i_j}=1} \tag{8}$$

で表される． □

3.2 真理値シミュレーションによる絞り込み

6 値シミュレーション処理後に残存する各組合せ箇所 L^m に対して，**図 4** に示す処理によって組合せ箇所を絞り込む．まず，L^m に含まれる各修正箇所 l_i の k_i 入力 LUT に，対応する 2^{k_i} 個の真理値変数を与える．次に，各誤り追跡入力 $\boldsymbol{\alpha}_q$ に対する一致関数 $c_{\boldsymbol{\alpha}_q}$ をもとに，$\boldsymbol{\alpha}_1$〜$\boldsymbol{\alpha}_q$ に関する共通一致関数

$$c^q_{\alpha_{\rm com}} = c^{q-1}_{\alpha_{\rm com}} \cdot c_{\boldsymbol{\alpha}_q} \tag{9}$$

を求める．ただし，$c^0_{\alpha_{\rm com}} = 1$ とする．ここで，$c^q_{\alpha_{\rm com}} \equiv 0$ ならば，補題 1 より L^m は真の組合せ箇所でないと判断できるので，L^m を削除する．適用した誤り追跡入力 $\boldsymbol{\alpha}_q, \cdots, \boldsymbol{\alpha}_{n_\alpha}$ に対して $c^{n_\alpha}_{\alpha_{\rm com}} \not\equiv 0$ が満足されると，L^m に含まれる各 LUT について機能修正を行う．

3.3 LUT 機能の修正法

真理値シミュレーションの結果，残存する組合せ箇所 L^m に関する LUT 機能の修正法，すなわち LUT

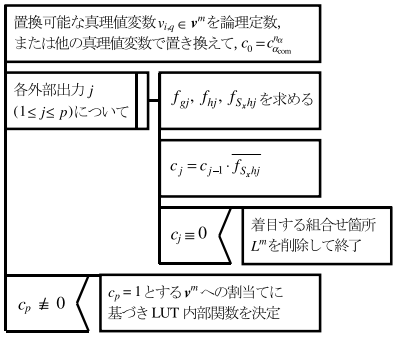

図 5 LUT 内部関数の決定
Fig. 5 Assignment of LUT function.

内部関数を決定する手法を**図 5** に示す．

この段階で残存している組合せ箇所については，適用した全誤り追跡入力 $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{n_\alpha}$ に関する共通一致関数充足条件 $c^{n_\alpha}_{\alpha_{\rm com}} = 1$ を満たす $\boldsymbol{v}^m$ の割当てが存在する．割当てが論理定数 0 または 1 のいずれか一方に特定される真理値変数については，その定数で置換する．さらに，他の真理値変数の値によって割当てが一意に決定される真理値変数についても置換する．例として，$v_{2,0} = \overline{v_{1,3}}$ 等があげられる．このように，共通一致関数充足条件を用いて可能な限り残存する真理値変数を減らしておくことで，外部入力変数ベクトル $\boldsymbol{x}$ に対する外部出力関数を表す BDD のノード数の増加を抑える．

次に，残存する真理値変数をもとに，外部出力関数を計算する．その結果を，

$$f'_{gj} = f_{gj}(\boldsymbol{x}, \boldsymbol{v}^m) \tag{10}$$

で表す．ここで，各外部出力に関する不一致関数

$$f_{hj} = f'_{gj} \oplus f_{sj} \tag{11}$$

の外部入力変数ベクトル x に関するスムース演算

$$f_{S_{\boldsymbol{x}}h_j} = S_{\boldsymbol{x}} f_{hj} = S_{x_1} S_{x_2} \cdots S_{x_n} f_{hj} \tag{12}$$

は，外部入力変数の全割当てに対する外部出力関数 f'_{gj} の，機能仕様 f_{sj} との不一致の可能性を示す．f'_{gj} が f_{sj} と一致するための必要十分条件は

$$f_{S_{\boldsymbol{x}}h_j} = 0 \tag{13}$$

である．よって，共通一致関数 $c^{n_\alpha}_{\alpha_{\rm com}}$ で初期設定された外部出力一致関数 c_j ($c_0 = c^{n_\alpha}_{\alpha_{\rm com}}$) を

$$c_j = c_{j-1} \cdot \overline{f_{S_{\boldsymbol{x}}h_j}} \tag{14}$$

のように求める．いずれかの j に対して $c_j \equiv 0$ となれば，真理値変数について $c_p = 1$ とする解が存在しないので，そのような組合せ箇所は候補から削除する．全外部出力に関する式 (14) の演算結果が $c_p \not\equiv 0$ となれば，解が存在する．$c_p = 1$ とする $\boldsymbol{v}^m$ への割当てに基づき，各箇所 $l_i \in L^m$ の LUT 内部関数について，すべての解を求める．

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

回路名	ゲート数	LUT 数	外部入力数	外部出力数
C432	160	109	36	7
C499	202	94	41	32
C880	383	184	60	26
C1355	546	166	41	32
C1908	880	346	33	25
C2670	1,193	598	233	140
C3540	1,669	742	50	22
C5315	2,307	1,146	178	123
C7552	3,513	1,324	207	108

4. 実験評価

本論文で提案した $\mathrm{EXL_{TV}}$ 法を C 言語（gcc ver. 2.95.2）を用いて計算機（Sun Ultra2 / model 2200）上に実装するとともに，実験評価を行った．

実験には，**表 1** に示す ISCAS'85 ベンチマーク回路[7] に対して無作為にゲート機能誤りを 1 個から 4 個挿入したうえで LUT にマッピングした回路例を用いた．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 $4 \times 20 \times 9 = 720$ 例について実験を行った．その際，BDD ノード数を約 840 万（$= 2^{23}$）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．

表 2，**表 3** にそれぞれ多重度 3，4 に対する実験結果を示す．また，処理時間に関する従来手法[1] との比較を**表 4** に示す．これら多重度は，挿入した誤りの個数ではなく，修正可能となった最小多重度を示す．表中の値は，すべて幾何平均による結果である．“組合せ箇所数” の欄は，考慮すべき組合せ箇所が各処理によって削減された過程を示している．すなわち，“EPI”，“6 値”，“TV”，“Final” は，それぞれ EPI を用いて抽出された個数，6 値シミュレーションによる絞り込み後の個数，真理値シミュレーション後に残存する個数，および真の組合せ箇所 $L^m_{\rm T}$ の個数を示す．また，表中の “1st” は最初の修正解が発見されるまでの処理時間を表す．また，“$\#E^m_{\rm T}$” は修正解の個数を表す．

実験の結果，$\mathrm{EXL_{TV}}$ 法によって従来の EXL 法と同様に存在するすべての解を求めることが可能であることを確認した．さらに，$\mathrm{EXL_{TV}}$ 法は EXL 法と比べて，多重度 1 の例で約 1/3，多重度 3 の例では約 1/7 の短い処理時間内での論理診断を可能とし，従来は対応が困難であった多重度 4 の設計誤りの修正を実現した．その要因として，次の 2 点が考えられる．

表 2 多重度 3 の設計誤りに対する実験結果（$m = 3$）
Table 2 Results for multiple errors ($m = 3$).

回路名	組合せ箇所数					$\#E_{\mathrm{T}}^m$	平均処理時間 (s)		最大
	Total	EPI	6 値	TV	Final		1st	Total	処理時間 (s)
C432	2.16×10^5	922.1	29.5	1.27	1.23	23.0	1.5	1.6	5.7
C499	1.39×10^5	129.3	90.9	1.24	1.24	10.7	4.1	4.5	26.6
C880	1.04×10^6	13.7	3.9	1.73	1.34	30.4	1.1	1.2	7.7
C1355	7.63×10^5	69.0	51.0	1.53	1.41	53.8	4.5	4.9	69.8
C1908	6.90×10^6	6,103	727.6	3.45	2.55	61.5	26.6	27.9	966.9
C2670	3.56×10^7	1,862	171.3	3.91	2.31	6,995	39.8	44.1	5,990
C3540	6.81×10^7	827.8	78.0	2.13	1.55	2,721	30.5	31.3	473.5
C5315	2.51×10^8	224.4	27.6	1.74	1.67	2,328	17.7	18.1	167.3
C7552	3.87×10^8	7,520	123.0	1.99	1.70	725.3	76.0	80.1	785.3
Ave.	6.95×10^6	495.1	63.8	1.95	1.61	209.9	10.3	11.0	145.6

表 3 多重度 4 の設計誤りに対する実験結果（$m = 4$）
Table 3 Results for multiple errors ($m = 4$).

回路名	組合せ箇所数					$\#E_{\mathrm{T}}^m$	平均処理時間 (s)		最大
	Total	EPI	6 値	TV	Final		1st	Total	処理時間 (s)
C432	5.78×10^6	33,159	1,753	1.71	1.41	462.4	27.8	29.2	675.1
C499	3.19×10^6	6,340	4,462	2.33	2.11	494.9	33.5	40.4	825.6
C880	4.73×10^7	45.9	10.0	1.43	1.23	35.6	6.1	6.2	20.5
C1355	3.13×10^7	936	651.4	2.49	2.49	2,889	19.7	23.3	1,402
C1908	5.80×10^8	77,505	1,430	4.21	1.71	10.6	333.2	335.8	43,909
C2670	5.31×10^9	5,822	182.4	5.10	3.29	5,302	470.5	474.0	1,705
C3540	1.26×10^{10}	2.93×10^5	2,188	2.83	2.13	6,335	4,601	4,626	19,087
C5315	7.17×10^{10}	1,716	164.6	2.71	2.44	61,733	3,533	3,537	15,563
C7552	1.28×10^{11}	1.57×10^6	2,783	4.07	1.37	3,495	11,322	11,344	59,809
Ave.	5.97×10^8	11,388	618.6	2.74	1.93	1,068	245.5	257.7	3,027

表 4 各多重度の設計誤りに対する処理時間と従来比
Table 4 Processing times for single to triple errors and ratios to conventional ones.

m	1		2		3	
回路名	平均 [s](比)	最大 [s](比)	平均 [s](比)	最大 [s](比)	平均 [s](比)	最大 [s](比)
C432	0.5 (0.17)	0.8 (0.22)	0.6 (0.18)	1.6 (0.28)	1.6 (0.26)	5.7(0.07)
C499	2.6 (0.33)	9.2 (0.67)	3.4 (0.13)	9.0 (0.01)	4.5 (0.05)	26.6(0.01)
C880	0.9 (0.27)	1.1 (0.14)	1.0 (0.20)	1.9 (0.17)	1.2 (0.14)	7.7(0.12)
C1355	2.4 (0.28)	5.7 (0.40)	3.1 (0.16)	5.5(0.004)	4.9 (0.05)	69.8(0.01)
C1908	1.9 (0.21)	2.8 (0.23)	3.4 (0.16)	35.9 (0.02)	27.9 (0.04)	966.9(0.06)
C2670	6.7 (0.46)	17.8 (0.03)	14.0 (0.16)	294.9 (0.02)	44.1 (0.08)	5,990(0.16)
C3540	6.4 (0.25)	7.6 (0.17)	8.5 (0.22)	54.4 (0.01)	31.3 (0.27)	473.5(0.03)
C5315	5.3 (0.36)	7.6 (0.41)	7.0 (0.37)	278.6 (0.63)	18.1 (0.41)	167.3(1.03)
C7552	10.1 (0.98)	29.3 (1.81)	16.4 (0.71)	59.6 (0.29)	80.1 (0.57)	785.3(0.58)
Ave.	2.8 (0.32)	5.4 (0.27)	4.1 (0.22)	22.4 (0.05)	11.0 (0.14)	145.6(0.08)

i) 真理値シミュレーションによる組合せ箇所の削減効果
ii) 真理値シミュレーション結果の利用による LUT 機能修正時の BDD ノード数削減効果

i) について，表 2 より $m = 3$ の場合の組合せ箇所数が “TV” では “6 値” の約 3.1%に削減され，$m = 4$ の場合は表 3 より約 0.44%にまで削減されている．ここで，各処理に要する時間の割合を**表 5** に示す．表中の “追跡入力”，“EPI”，“6 値”，“TV”，“機能修正” は，それぞれ誤り追跡入力の生成，EPI 処理，6 値シミュレーション，真理値シミュレーション，LUT 機能修正に要した処理時間の割合を示す．LUT 機能修正

表 5 各処理に要する処理時間の割合（%）
Table 5 Percentage of processing times for each procedure.

m	追跡入力	EPI	6 値	TV	機能修正
1	65.1	20.0	2.8	3.1	9.0
2	54.9	17.6	9.8	4.6	13.1
3	30.8	25.1	26.8	6.2	11.1
4	5.6	59.3	25.1	6.6	3.4

表 6 論理関数処理における BDD の最大ノード数
Table 6 Maximum numbers of BDD nodes and ratios to conventional ones.

m	1	2	3	4
回路名	ノード数 (比)	ノード数 (比)	ノード数 (比)	ノード数
C432	50 (0.88)	64 (0.57)	72 (0.15)	554
C499	434 (0.64)	525 (0.35)	719 (0.35)	2,703
C880	97 (0.66)	126 (0.32)	151 (0.19)	161
C1355	322 (0.51)	419 (0.29)	479 (0.24)	1,900
C1908	161 (0.44)	265 (0.24)	356 (0.17)	961
C2670	301 (0.37)	823 (0.21)	946 (0.19)	2,028
C3540	435 (0.78)	717 (0.75)	705 (0.36)	945
C5315	170 (0.86)	176 (0.44)	197 (0.22)	197
C7552	827 (1.17)	653 (0.32)	998 (0.25)	1,378
Ave.	311 (0.70)	419 (0.39)	514 (0.23)	1,203

各ノード数について，1,000 ノードを単位として示す．

に要する時間の割合は $m = 3$ の場合に 11.1%となっている．単純に処理すべき組合せ箇所数のみ考慮すると，真理値シミュレーションを行わない場合は約 33 倍に増加すると考えられ，i) の効果が認められる．

ii) に関して，**表 6** に示す論理関数処理により使用した BDD の最大ノード数（1,000 ノード単位）を示す．$m = 1$〜3 については，従来手法との比もあわせて示す．たとえば，$m = 3$ の場合の C432 では，提案手法における BDD ノード数は 72,000 ノードであり，これは従来手法で要したノード数の 15%であることを表している．表 6 から，$m = 3$ の場合には BDD のノード数が従来の EXL 法の 23%に抑えられていることが分かる．この結果，EXL 法ではノード数の制限を超えて修正不能となった 3 個の誤りを含む C2670 の 7 例，C7552 の 4 例についても修正可能となった．

その一方で表 5 より，多重度 4 の設計誤りを含む実験例では，処理時間の大半を EPI による組合せ箇所抽出，ならびに 6 値シミュレーションによる絞り込みに費やしていることが分かる．今後，修正可能な設計誤りの多重度をさらに向上させるために，これら組合せ箇所の抽出と絞り込みの効率化のほか，部分回路を対象とした修正を繰り返す手法[8] の適用可能性についても検討の余地がある．

5. ま と め

本論文では，EXL 法をもとに，真理値シミュレーションによって組合せ箇所を効率良く絞り込み，さらにその結果をもとに短時間で LUT 機能修正を行う $\mathrm{EXL}_{\mathrm{TV}}$ 法を提案した．処理時間を平均で約 1/7〜1/3 に短縮する効果が確認されるとともに，多重度 4 の設計誤りへの対応が可能となった．

今後の課題として，組合せ箇所の抽出・絞り込み処理の高速化，部分回路を対象とする反復修正等により，修正可能な設計誤りの多重度をさらに向上させることがあげられる．

謝辞 本研究の一部は，日本学術振興会科学研究費補助金（特別研究員奨励費）による．

参 考 文 献

1) 沼 昌宏, 井上 宏, 黒木修隆, 山本啓輔：誤り追跡入力と論理関数処理を併用した FPGA 対応論理診断手法，電子情報通信学会論文誌，Vol.J84-D-I, No.10, pp.1474–1483 (2001).
2) Watanabe, Y. and Brayton, R.K.: Incremental synthesis for engineering changes, *ICCAD-91*, pp.40–43 (1991).
3) Fujita, M., Tamiya, Y., Kukimoto, Y. and Chen, K.C.: Application of Boolean unification to combinational locig synthesis, *ICCAD-91*, pp.510–513 (1991).
4) Kukimoto, Y. and Fujita, M.: Rectification method for lookup-table type FPGA's, *ICCAD-92*, pp.54–61 (1992).
5) Pomeranz, I. and Reddy, M.: On error correction in macro-based circuits, *IEEE Trans. CAD*, Vol.16, No.10, pp.1088–1100 (1997).
6) Tomita, M., Suganuma, N. and Hirano, K.: Pattern generation for locating design errrors, *IEICE Trans.*, Vol.E77-A, No.5, pp.881–893 (1994).
7) Brglez, F. and Fujiwara, H.: A neutral netlist of 10 combinational benchmark circuits and a target translation in FORTRAN, *ISCAS-85* (1985).
8) 井上 宏, 増田浩平, 黒木修隆, 沼 昌宏：修正可能な設計誤りの多重度を向上させた論理診断手法，情報処理学会論文誌，Vol.41, No.4, pp.970–973 (2000).

(平成 13 年 9 月 25 日受付)
(平成 14 年 3 月 14 日採録)

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．

井上　宏

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．

皆見　利行

1977 年生．2000 年神戸大学工学部電気電子工学科卒業．同年，同大学大学院修士課程進学．論理診断，画像処理に関する研究に従事．

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．

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．