

論理設計誤りの追跡と修正の自動化†

富田 昌宏^{††} 蔣 洪海^{†††} 山本 保^{†††}

論理回路の論理設計段階において、種々の設計自動化が行われているが、完全な自動設計は極めて困難であり、部分的な変更および変更後の検証が不可欠である。よって、QTAT (Quick Turn Around Time) を実現するためには、論理検証で設計誤りと判定された場合の、誤り追跡・修正の自動化が必要である。現在提案されている誤り追跡手法では限定範囲が広く、修正の具体的な指針となりにくい。そこで本論文では、誤りを顕在化し得る入力パターンに着目し、ゲートレベルの設計誤りに対する追跡と修正の自動化手法を提案する。組合せ回路や単一クロックを用いた同期式順序回路において、ゲート機能の誤りや信号線欠落などのゲートレベルの設計誤りが単一に存在する場合を対象とする。本手法は、ある入力パターンに対する設計回路の出力が機能記述と一致しない場合に、その入力パターンとハミング距離が1で、かつ回路出力が機能記述と一致する別の入力パターンを発見し、外部出力まで伝搬すべき外部入力进行を特定する。特定された外部入力のみをブール変数に置換した入力パターンを用い、伝搬が停止したゲートから後方追跡を行う。回路出力が機能記述と一致しない複数の入力パターンについての追跡結果から、誤り箇所の特定と修正法の提示を行う。ISCAS ベンチマーク回路に論理設計誤りを仮定した回路に対して、本手法を適用した結果、すべての例で誤りの追跡と修正法の提示が可能となった。

1. はじめに

近年の論理回路の大規模化、複雑化に伴い、論理設計に対する計算機による支援が不可欠となっている。種々の論理設計自動化手法が提案・実用化されているが、すべての設計について適用可能とはいえず、回路規模や性能に対する要求が厳しい場合は人手設計に頼らざるを得ない。人手で設計または修正された部分については論理検証^{1)~6)}が不可欠であると同時に、検証によって誤りの存在が確認された際の誤り追跡・修正を自動化し、論理設計全体に要する時間を短縮する必要がある。

現在提案されている誤り追跡の手法^{6)~8)}は、誤り箇所の限定範囲が広く、修正を行う場合に具体的な指針となりにくい点に問題がある。そこで本論文では、誤りの具体的な修正方法を、ゲートレベルで設計者に提示できる手法を提案する。

本手法は、ある入力パターンについて、設計された回路の出力が機能記述に対して不一致を起こすとき、その入力パターンとハミング距離が1で、かつ回路出力が機能記述と一致する別の入力パターンを見つける。これら2つのパターンより、外部出力まで伝搬すべき外部入力进行を特定する。特定された外部入力のみを

ブール変数に置換した入力パターンを用い、そのブール変数の伝搬状況を伝搬が停止したゲートから後方追跡する。不一致を生じる複数の入力パターンについての追跡結果から、誤り箇所の特定と、修正法の提示を行う。

以下、2章で論理設計誤りの定義を行う。3章で誤り追跡入力を定義し、その特徴を示した後、4章で実際の誤り追跡法と修正法を述べる。5章で、ISCAS ベンチマーク回路⁹⁾に対して本手法を適用した実験について述べ、考察を加える。

2. 論理設計誤り

本章では、本手法で対象とする回路および論理設計誤りの定義を行う。対象とする回路は、組合せ論理回路と単一クロックを用いた同期式順序回路である。多出力回路は1出力の組合せ回路(以後、コーン回路)に分解可能である。対象とする回路はコーン回路に帰着できるので、以下ではコーン回路のみを対象とする。

機能記述から導かれたブール関数を Fs 、回路から導かれたブール関数を Fg とする。ここで、全入力パターンから入力禁止項を除いたパターンの集合を考える。この集合の要素 v が $Fg(v) = Fs(v)$ を満たすとき、このパターンを一致入力と呼び、満たさないとき、これを不一致入力と呼ぶことにする。このとき、論理設計とは不一致入力の存在しない回路を設計することであり、設計された回路は機能を満たすといえる。

論理設計誤りは、実際に設計した回路(以後、実回路)が、機能記述を満たさない場合に定義される。図

† A Technique for Automatic Logic Error Location and Correction by MASAHIRO TOMITA (The Graduate School of Science and Technology, Kobe University), HONGHAI JIANG and TAMOTSU YAMAMOTO (Department of Precision Engineering, Faculty of Engineering, University of Tokyo).

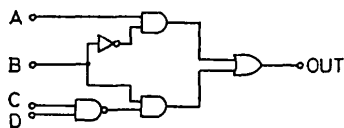
†† 神戸大学大学院自然科学研究科

††† 東京大学工学部精密機械工学科

AB	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	1	1	0	1
10	1	1	1	1

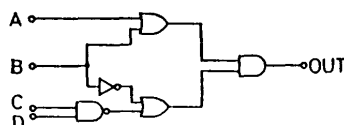
(a) 機能記述

(a) Functional description.



(b) 回路例 (1)

(b) Circuit example (1).



(c) 回路例 (2)

(c) Circuit example (2).

図 1 回路構成の多様性

Fig. 1 Variety of circuit construction.

表 1 単純な論理設計誤り

Table 1 Simple design errors.

設計誤り	理想回路	誤りを含む回路
ゲート誤り	ゲート機能選択	
	インバータ過剰	
	インバータ欠落	
	ゲート過剰	
	ゲート欠落	
信号線誤り	信号線過剰	
	信号線欠落	

1に示すように、ある特定の機能に対して、それを満たす回路は一意的には定まらない。実回路を修正して機能記述を満たす回路(以後、理想回路⁶⁾)を設計する場合に、論理設計誤りは、理想回路と実回路との差異として定義できる。論理設計誤りは、表1に示す単純な論理設計誤り単独、あるいは、それらの組合せで表現可能である⁶⁾。

なお、現状ではすべての論理設計誤りについて、誤り追跡と修正法の提示を行うまでは至っていない。本手法では、単純な論理設計誤りが単一に存在するときのみ、つまり実回路に対して単一の修正のみで理

想回路が得られるときのみを対象とする。自動論理生成によって得られた回路に部分修正を加えた場合、単一誤りの発生する可能性が高いと考えられ、本手法の適用が有効である。また、すべての設計誤りは単一誤りの組合せで表現可能であるため、まず基本となる単一誤りに対する追跡・修正法を確立することが重要である。

3. 誤り追跡入力

本手法は、2章で示した論理設計誤りに対して、誤りの追跡と修正法の提示を行うために、誤り追跡入力と呼ぶ入力パターンを用いる点を特徴とする。本章では、まず誤り追跡入力と、追跡を行う際に重要な用語を定義し、その後、誤り追跡入力の性質と生成法を示す。

3.1 誤り追跡と伝搬停止ゲート

まず、誤り追跡入力を以下に定義する。

定義 n 入力コーン回路において、機能記述から導かれたブール関数を F_s 、回路から導かれたブール関数を F_g とする。ブール変数 x と、その補 $\bar{x}$ を導入した場合、以下の条件を満たす入力パターン $vt = (a_1, \dots, a_{i-1}, x/\bar{x}, a_{i+1}, \dots, a_n)$ を誤り追跡入力と定義する。ただし $x/\bar{x}$ は x または $\bar{x}$ の一方を表し、 a_j ($j=1, \dots, n; j \neq i$) および a は定数 (0 または 1) を表す。

$$F_s(vt) = x \quad (1)$$

$$F_g(vt) = a \quad (2)$$

図2に示すように、機能記述を満たす回路では、誤り追跡入力に対する出力値は、ブール変数 x の値に依存するが、実回路では出力値は x に依存しない。よって、実回路が x を出力するように修正することが、機

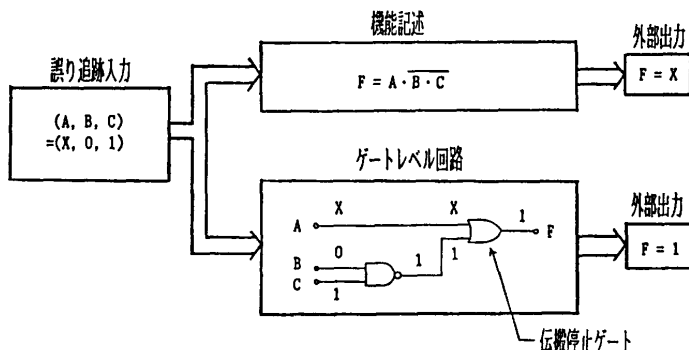


図 2 誤り追跡入力の伝搬概念図

Fig. 2 Input pattern for locating design errors and $x/\bar{x}$ propagation.

能記述を満たすためには必要である。

次に、この誤り追跡入力を用いて追跡を行うための重要な用語を定義する。

定義 $x/\bar{x}$ の伝搬が停止するゲート、つまり x と $\bar{x}$ の少なくとも一方を入力としてもち、かつ 0 または 1 を出力するゲートを伝搬停止ゲートと定義する。

定義 ある誤り追跡入力に対して存在するすべての伝搬停止ゲートについて、各伝搬停止ゲートを出力ゲートとするコーン回路を考える。少なくとも 1 個以上のコーン回路に含まれるゲート、外部入力、信号線のみから構成される部分回路を、その誤り追跡入力に対する伝搬部分回路と定義する。

3.2 誤り追跡入力の性質

本節では、前節で定義した用語を用いて、誤り追跡入力の性質を述べる。すべて定理証明が可能であるが、本稿では概略を述べるにとどめる。これらは追跡と修正法の提示を行う際の根拠となる。

(1) 一致入力と不一致入力に対する回路動作の差異の明確化

x に 0 を代入した入力パターンを $vt0$ 、 x に 1 を代入した入力パターンを $vt1$ とする。 a は 0 または 1 なる定数であるから、式(1)、式(2)より以下の 2 式(式(3)、式(4))のうち一方のみが成り立つ。

$$\begin{aligned} Fs(vt0) &= Fg(vt0) \quad \text{かつ} \\ Fs(vt1) &\neq Fg(vt1) \end{aligned} \quad (3)$$

$$\begin{aligned} Fs(vt0) &\neq Fg(vt0) \quad \text{かつ} \\ Fs(vt1) &= Fg(vt1) \end{aligned} \quad (4)$$

つまり、 $vt0$ と $vt1$ の 2 つの入力パターンは、一方は一致入力、他方は不一致入力である。よって、 vt によるシミュレーションは一致入力によるシミュレーションと不一致入力によるシミュレーションを同時に行うことを意味し、両者による回路動作の差異を明確化することが可能となる。

(2) 伝搬停止ゲートの存在

誤り追跡入力に対して、実回路中に必ず伝搬停止ゲートが存在することを以下に示す。外部入力中に $x/\bar{x}$ が存在するため、実回路中には $x/\bar{x}$ を入力に持つゲートが必ず存在する。もし、実回路中に伝搬停止ゲートが存在しない場合、 $x/\bar{x}$ を入力とするゲートは必ず $x/\bar{x}$ を出力するため、外部出力は $x/\bar{x}$ となる。ところがこれは誤り追跡入力の定義に反する。よって、伝搬停止ゲートが必ず存在する。

(3) 伝搬部分回路中の論理設計誤りの存在

いかなるゲート演算も、入力に $x/\bar{x}$ をもたなければ

ば、 $x/\bar{x}$ を出力することはない。また、信号値が $x/\bar{x}$ となる信号線は伝搬部分回路中にしか存在しない。よって、伝搬部分回路より出力側の修正のみでは、外部出力が x となることはない。ゆえに、伝搬部分回路に論理設計誤りが必ず存在する。

(4) 修正の必要条件

信号値が $x/\bar{x}$ である信号線は伝搬部分回路中にしか存在しないため、外部出力を x とするには、伝搬部分回路が出力側へ拡張するように修正する必要がある。この場合、以下の 2 通りの修正が考えられる。

i) 1 つ以上の伝搬停止ゲートが $x/\bar{x}$ を出力するように修正する。

ii) 信号値 $x/\bar{x}$ の信号線の分岐先を伝搬部分回路外へ接続する。

i) の修正で外部出力 x が得られる場合、その誤りを伝搬停止ゲート関連誤りと呼び、ii) の修正で外部出力 x が得られる場合、その誤りを伝搬信号線欠落誤りと呼ぶことにする。

3.3 誤り追跡入力の生成法

本手法は、論理検証システム¹⁾⁻⁶⁾で、不一致入力 that 得られていることを前提とする。誤り追跡入力を生成するためには必ずしも厳密にすべての不一致入力を求める必要はない。以下に、追跡入力の生成法を述べる。3.1 節で示した定義より、誤り追跡入力は、ハミング距離が 1 である 2 つの入力パターンに対する機能記述出力と回路出力の 4 つの値に着目する必要がある。そこで、全入力パターンを機能記述出力と回路出力に着目して分類する。概念図を図 3 に示す。

$$V00 = \{v \mid Fs(v) = 0 \wedge Fg(v) = 0\}$$

$$V01 = \{v \mid Fs(v) = 0 \wedge Fg(v) = 1\}$$

$$V11 = \{v \mid Fs(v) = 1 \wedge Fg(v) = 1\}$$

$$V10 = \{v \mid Fs(v) = 1 \wedge Fg(v) = 0\}$$

$$Vdc = \{v \mid v \text{ は入力禁止項}\}$$

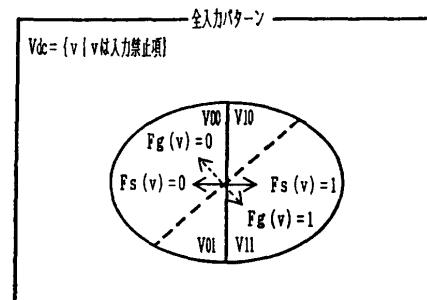


図 3 入力パターンの分類

Fig. 3 Classification of input patterns.

このとき、 $V01$ の要素と $V10$ の要素が不一致入力である。追跡入力、ある不一致入力 v に対してハミング距離が1のパターン v' が以下の条件を満たすとき生成可能である。

$$Fs(v) \neq Fs(v') \quad \text{かつ}$$

$$Fg(v) = Fg(v')$$

これには、(a) $v \in V10$ に対して $v' \in V00$ が存在する場合と、(b) $v \in V01$ に対して $v' \in V11$ が存在する場合の2通りがあり得る。よって、誤り追跡入力の存在条件は、(a)または(b)に対応する v, v' の組が1組以上存在することである。存在条件を満たさない例もあり得るが、それらに対する考察は5.4節で行う。

誤り追跡入力は、定義に従った網羅的なシミュレーションによって生成可能であるが、不一致入力を用いることで、生成が容易となる。すべての不一致入力が与えられれば、存在するすべての誤り追跡入力の生成が可能である。以下に上記(a)に対応する生成手順を、外部入力数を n として示す。(b)についても同様である。

- (1) 論理検証システムより得られた不一致入力のうち $V10$ の要素すべてについて(2)を行う。
- (2) $i=1$ から $i=n$ まで順に(3)から(5)を行う。
- (3) $v \in V10$ の第 i 桁を反転させたパターン v' に着目する。 v と v' のハミング距離は1である。
- (4) もし $v' \in V00$ であれば、(5)を行う。
- (5) v の第 i 桁が $Fs(v)$ と等しいならば、 v の第 i 桁を x に置換したパターンが誤り追跡入力であり、等しくない場合は $\bar{x}$ に置換したパターンが誤り追跡入力である。

追跡入力生成に要する処理時間の大部分は、上記の処理のうち(4)の判定処理である。この判定処理には論理シミュレーションが必要であり、逐次的なソフトウェアシミュレータの利用を前提とすると、処理時間は実回路全体のゲート数に比例する。よって、論理検証システムより得られた不一致入力の総数を m とし、実回路のゲート数を g とした場合、追跡入力の生成時間は $O(n * m * g)$ である。

4. 誤り追跡と修正

本章では、誤り追跡と修正法提示の手順について述べる。処理概要を示した後、処理の主要部を成す後方追跡について

具体例を交えて詳細に述べる。

4.1 誤り追跡と修正の処理概要

本節では、誤り追跡の処理概要を述べる。

定義 誤りを含む実回路に変更を加えた結果、その回路が誤り追跡入力に対して x を出力する回路に変更される場合、その変更を誤り候補と呼ぶ。

単一論理設計誤りを修正した場合は、すべての入力に対して機能記述を満たす出力が得られるため、誤り追跡入力に対しては外部出力 x を得る。よって、各誤り追跡入力に対して、すべての誤り候補が求められる場合、図4に示す包含関係が成り立つ。全誤り追跡入力に対して誤り候補とされたものを最終的な誤り候補とすることで、それらが、単一論理設計誤りである可能性を高めることができる。

処理概要を図5に従って説明する。処理(5)(6)は、各誤り追跡入力に対して誤り候補の集合を求める処理であり、次節で詳述する。

- (1) 不一致入力をもとに、誤り追跡入力を求める。
- (2) すべての誤り追跡入力について(3)から(7)を行う。
- (3) 誤り追跡入力に対する伝搬停止ゲートを検出する。

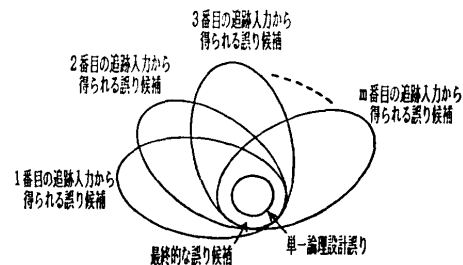


図4 誤り追跡入力と単一論理設計誤りの包含関係
Fig. 4 Relation of inclusion between error candidates and single design errors.

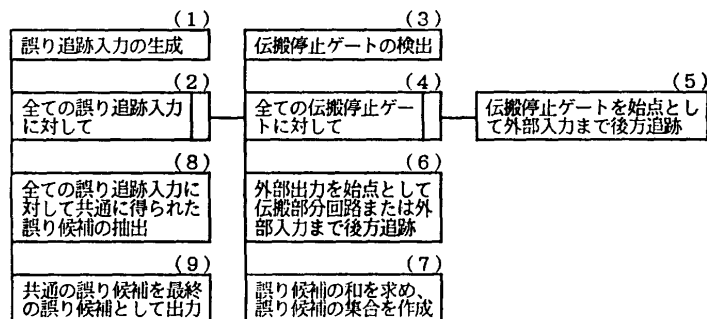


図5 処理概要

Fig. 5 Outline of error location and correction.

- (4) 検出されたすべての伝搬停止ゲートについて(5)を行い、伝搬停止ゲート関連誤り候補をすべて求める。
- (5) 伝搬停止ゲートを始点として、外部入力まで後方追跡を行う。追跡の過程で伝搬停止ゲート関連誤り候補を求める。
- (6) 外部入力を始点として、伝搬部分回路または外部入力まで後方追跡を行い、伝搬信号線欠落誤り候補をすべて求める。
- (7) (5)(6)で求められた誤り候補の和集合を作成し、その誤り追跡入力に対する誤り候補の集合を得る。
- (8) すべての誤り追跡入力について共通に得られた誤り候補を抽出する。
- (9) (8)で得られた誤り候補を最終的な誤り候補として出力する。

以上の処理では、(3)から(7)が処理時間の大部分を占める。各誤り追跡入力に対して同じ処理であるから、処理時間は用いる誤り追跡入力数にほぼ比例することが予想できる。

4.2 後方追跡

本節では、前節の処理(5)(6)に対応する後方追跡部について詳しく述べた後、具体例を示す。

4.2.1 後方追跡処理

前節の処理で、最終的な誤り候補がすべての単一論理設計誤りを含むことを保証するためには、各追跡入力に対して、存在する誤り候補をすべて求める必要がある。ここでは、追跡入力に対する誤り候補をすべて求めるための処理を示す。

3.2 節で述べた修正の必要条件を網羅するために、必要条件の必要条件を求める処理を繰り返す。求めた必要条件が外部出力を実際に x とするか否かは、シミュレーションで調べる。

あるゲートの出力値が変化することによって、外部出力が x となる可能性がある場合、その値をそのゲートの出力期待値と呼ぶ。また、あるゲートの出力値を出力期待値にする可能性のある入力信号値を入力期待値と呼ぶ。例として、AND ゲートの場合の出力期待値に対する入力期待値の関係を、表2に示す。追跡は、入力信号線の信号値が入力期待値になる可能性を

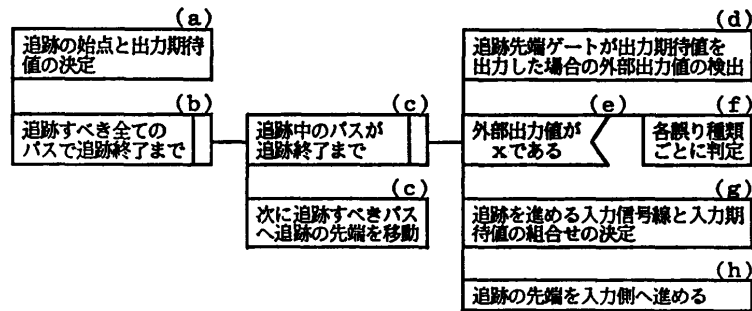


図6 後方追跡処理

Fig. 6 Backtracing algorithm.

表2 出力期待値と入力期待値の関係 (AND ゲートの場合)

Table 2 Output desired value and input desired values.

出力期待値	入力期待値	
0	{0}	{0, 1, x }
	{0, 1}	{0, 1, $\bar{x}$ }
	{0, x }	{0, x , $\bar{x}$ }
	{0, $\bar{x}$ }	{1, x , $\bar{x}$ }
	{ x , $\bar{x}$ }	{0, 1, x , $\bar{x}$ }
1	{1}	
x	{ x }	{1, x }
$\bar{x}$	{ $\bar{x}$ }	{1, $\bar{x}$ }

求めて後方（入力側）へ進む。

後方追跡の処理フローを図6を用いて順に述べる。

- (a) 伝搬停止ゲート関連誤り候補を求める後方追跡（前節(5)）では、伝搬停止ゲートを始点として、 x と $\bar{x}$ のそれぞれを始点における出力期待値として追跡を開始する。伝搬信号線欠落誤り候補を求める後方追跡（前節(6)）では、外部出力ゲートの出力期待値を x としてトレースを開始する。これにより、3.2 節の修正の必要条件を網羅する。
- (b) すべての追跡対象パスに関して(c)から(h)を行う。
- (c) 追跡中のパスについて、後方追跡の先端が外部入力に到達するまで(d)から(h)を行う。追跡中のパスが外部入力まで到達したら、次に追跡すべきパスに追跡の先端を移す。
- (d) 追跡先端ゲートが出力期待値を出力した場合の外部出力値を求める。
- (e) (d)で外部出力値が x となった場合、そのゲートに論理設計誤りの可能性がある。

(f) 誤りの可能性を判定する。判定方法を以下に記す。

i) インバータ欠落・過剰誤りは、期待出力値が実際の出力の反転値であるとき、その可能性がある。欠落ゲートから分岐がある場合も含めて検討する。

ii) ゲート機能誤りは、ゲート機能を変更することによって、出力値が、出力期待値に変更されるとき、その可能性がある。

iii) 信号線欠落誤りは、そのゲートに新たな入力信号線を追加し、ある信号値を加えられたときに出力値が出力期待値をみたすとき、その可能性がある。信号線欠落誤りについては、欠落信号線の始点と終点を対にして提示する。

iv) 信号線過剰誤りは各入力信号線について検討する。その信号線を除くことによって、ゲート出力値が出力期待値を満たすとき、信号線過剰誤りの可能性がある。

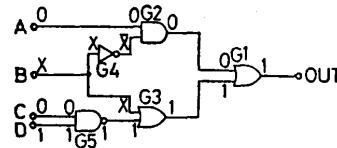
(g) 追跡先端ゲートのすべての入力信号線について、ゲート機能と出力期待値によって、追跡を進める入力信号線と入力期待値を決定する。追跡すべき組合せが複数ある場合は、depth-firstで追跡する。

(h) (g)で決めた入力信号線を出力とするゲートに後方追跡の先端を進める。

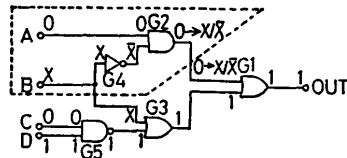
以上の処理では、(d)が処理時間を決める最大因子である。具体的な処理時間は実験例に依存するが、回路規模をもとに予測を行う。(d)が実行される回数は、伝搬部分回路の大きさと、信号線の分岐数に依存する。伝搬部分回路の大きさは追跡入力ごとに異なるが、平均的には実回路全体のゲート数に、ほぼ比例すると予想できる。また、欠落インバータからの出力分岐を考慮すると、分岐数 k の信号線に対してインバータ欠落の可能性を調べるには 2^k-1 通りの場合分けが必要であり、最大分岐点の分岐数に着目する必要がある。一方(d)を1回処理するために必要な時間は、逐次的なソフトウェアシミュレータの利用を前提とすると、実回路全体のゲート数に比例する。以上より、実回路全体のゲート数を g 、最大分岐数を f とし、 a を定数とすると、処理時間は $O(g * (g + a2^f))$ と予測できる。

4.2.2 後方追跡の具体例

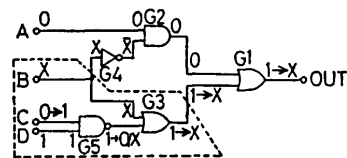
図7を用いて、誤り候補を求めるための後方追跡の具体例を示す。



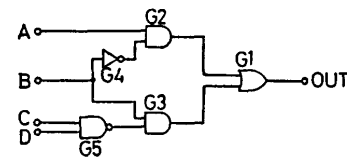
(a) 伝搬停止ゲートの検出
(a) Finding out $x/\bar{x}$ propagation stopped gates.



(b) G2 から外部入力への後方追跡
(b) Backtracing from G2 to primary inputs.



(c) G3 から外部入力への後方追跡
(c) Backtracing from G3 to primary inputs.



(d) 得られた理想回路
(d) An acquired ideal circuit.

図7 後方追跡の例

Fig. 7 Example of backtracing.

- (a) この例では、伝搬停止ゲートは G2 と G3 である。
- (b) G2 を後方追跡の始点とする。G2 が $x/\bar{x}$ を出力しても外部出力は x にはならない。
- (c) G3 を後方追跡の始点とする。G3 が x を出力した場合、外部出力は x である。そこで、まず G3 のゲート機能の誤り可能性を求める。 x , 1 を入力として x を出力するゲートは AND, XNOR が考えられる。次に、G3 の機能は正しいが入力信号値が誤っている場合を想定して後方追跡を行う。G5 が 0 または x を出力すれば G3 は x を出力する。0, 1 を入力として 0 を出力するゲートは AND, NOR, XNOR が考えられるが、 x を出力するゲートは考えられない。さらに、G5 が 0 または x を出力する条件を求めて後方追跡を行う。C と G5 の間に

INV が欠落している場合が考えられる。

以上より、この追跡入力に対しては以下の誤り候補が列挙される。

- i) G3 は AND または XNOR である。
 - ii) G5 は AND または NOR または XNOR である。
 - iii) CとG5の間に INV が欠落している。
- (d) 上記の誤り候補のうち、他のすべての追跡入力パターンに対しても誤り候補とされるものは G3 が AND である場合だけである。よって、最終的に誤り候補としてそれを出力する。

5. 実験と考察

有効性評価のため、本手法を計算機上で実現し、ISCAS 85 のベンチマーク回路⁹⁾を対象として実験を行った。実験は、ベンチマーク回路中に単純論理設計誤りを故意に挿入し、その論理設計誤りを追跡する場合の、誤り候補数と処理時間にもとづいて評価する。

実験結果に考察を加えた後、複数誤りへの対応について検討する。

5.1 評価項目

評価項目を以下に示す。

(1) 限定率

本手法では、誤り候補が実際に論理設計誤りか否か

を判定するには、修正後の論理検証が必要である。実回路中に単一論理設計誤りが存在する場合、必ず誤り候補に含まれるので、誤り候補に従った修正と論理検証を順に行えば、短時間で修正を行うことが可能である。単一論理設計誤りは、複数存在し得るので、誤り候補のうち単一論理設計誤りの占める割合（限定率）が高いほど、修正がより短時間で済むことが期待される。

(2) 誤り追跡入力数と限定率、処理時間の相関

適用する誤り追跡入力の順番を固定させて考える場合、用いる誤り追跡入力数に対して、限定率は単調増加する。そこで全誤り追跡入力中、実際の追跡に用いる入力数を制限して、誤り追跡入力数と限定率との相関を求める。実験で用いる誤り追跡入力の順番は、生成された順番である。追跡に用いる誤り追跡入力数を制限すれば処理時間は低減するので、この実験を通して処理時間と限定率の相関が得られる。

誤り追跡入力数の影響を、以下の項目にもとづいて評価する。

i) 限定率

ii) 処理時間

(3) 回路規模と処理時間の相関

4章で予想した相関と、実験結果を比較する。

表 3 実験結果
Table 3 Experimental results.

回路コード	回路名 外部出力名	外部 入力数	ゲート数	誤り種類	限定率 (%)	追 入 力 数	処理時間 (sec)
S1	C2670 No. 156	8	32	a インバータ欠落	100	144	253
				b インバータ過剰	100	64	87
				c ゲート機能選択	100	256	350
				d 信号線欠落	100	24	31
				e 信号線過剰	75	128	346
S2	C7552 No. 397	8	40	a インバータ欠落	100	176	263
				b インバータ過剰	100	120	275
				c ゲート機能選択	100	176	212
				e 信号線過剰	67	124	206
S3	C2670 No. 227	8	46	a インバータ欠落	100	256	809
				b インバータ過剰	100	128	183
				c ゲート機能選択	100	256	343
				d 信号線欠落	100	192	511
				e 信号線過剰	100	192	574
S4	C3540 No. 351	8	63	a インバータ欠落	100	256	847
				b インバータ過剰	100	256	833
				c ゲート機能選択	100	256	542
				e 信号線過剰	100	256	1,567

5.2 対象回路と実験環境

本節では、対象回路と実験環境について述べる。対象回路は、ベンチマーク回路から選んだコーン回路のそれぞれに、単一単純論理設計誤りを挿入した回路である。実験環境を以下に示す。

(1) プログラム規模

論理設計誤り追跡のプログラムは、以下の3部から成っており、C言語を用いている。

- i) 論理シミュレータ部 2,000 行
- ii) 誤り追跡部 1,300 行
- iii) 画面出力部 1,000 行

(2) 使用計算機

日立ワークステーション 2050/32
(CPU: MC 68020, 約 1 MIPS)

5.3 実験結果

実験回路と実験結果を表3に示す。回路例S2とS4には入力数3以上のゲートが存在しないため、信号線欠落誤りの仮定は行っていない。

(1) 限定率

限定率は、18例中16例で100%である。誤り候補が実際に単一論理設計誤りである割合が、極めて高いことがわかる。限定率が高い理由を以下に述べる。3.2節で述べたように、誤り追跡入力による追跡は、誤り追跡入力の生成に用いた、不一致入力と一致入力の両入力に対するシミュレーションを同時に行うことになる。不一致入力に対してのみ着目して、正しく出力する修正を行う場合、一致入力に対して不一致を引き起こす修正法を提示する場合が多い。しかし本手法では、このような修正法は削減される。

また、本手法は、各誤り追跡入力に対して得られた誤り候補の集合の積を最終的な誤り候補としている。

(2) 誤り追跡入力数と限定率、処理時間の相関

本手法では、限定率は使用する誤り追跡入力数に対して単調に増加するが、全追跡入力をいなくとも限定率が上限に達する場合が多い。図8に18の実験例について、用いた追跡入力数と限定率が上限に達した実験回路数の関係を示す。

図9に回路S3について、用いた追跡入力数と追跡に要した時間（誤り追跡入力生成時間は含まない）の関係を示す。処理時間は使用する誤り追跡入力数には

ほぼ比例する。他の回路例でも同様の結果を得ている。

誤り追跡入力の生成時間は、不一致入力数に比例する。全実験例について不一致入力数に比べて生成された誤り追跡入力数のほうが多く、平均で2.2倍である。1追跡入力当たりの生成時間は、追跡時間に比べて1%未満であり、無視できる。

誤り候補数が、候補すべてに対して修正・論理検証可能な個数に減った段階で誤り追跡を終了し、提示された修正法に従って修正・論理検証を行うことで、全体の時間短縮が期待される。

(3) 回路規模と処理時間の相関

4章での処理時間の予測に基づくと、処理時間を T 秒、用いた追跡入力数を m 、実回路全体のゲート数を g 、最大分岐数を f とし、 a 、 b を定数としたときに、これらの関係は式(5)で表し得る。

$$T/m = b(g * (g + a2')) \quad (5)$$

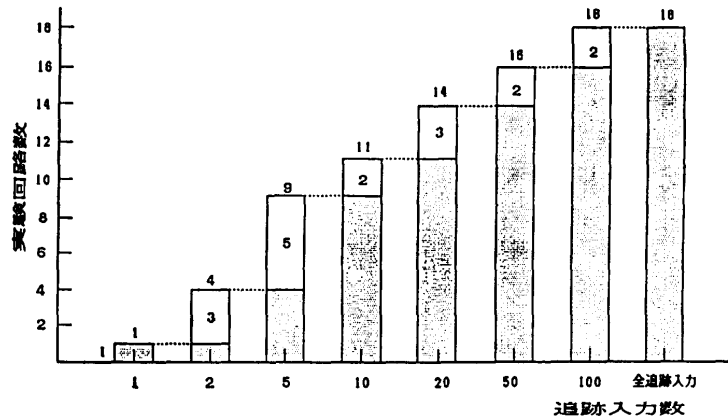


図8 用いた追跡入力数と限定率が上限に達した実験回路数
Fig. 8 Number of circuits with final hit rate value vs. number of input patterns for locating design errors.

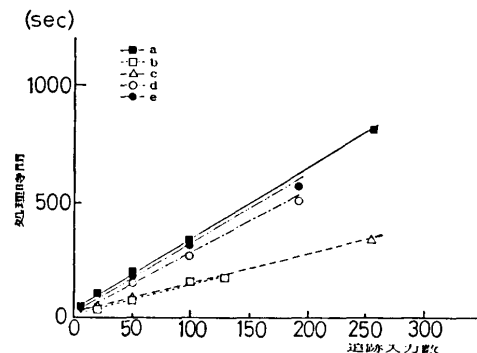


図9 追跡入力数と処理時間 (回路S3について)
Fig. 9 Processing time vs. number of input patterns for locating design errors of circuit S3.

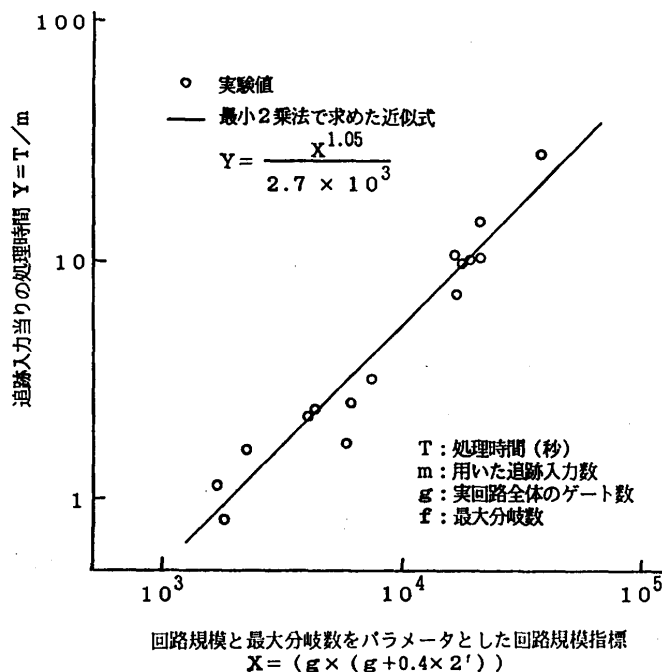


図 10 回路規模指標と処理時間の相関
Fig. 10 Processing time vs. circuit size.

ISCAS の別の回路例も加えて、回路規模と追跡に要した処理時間の相関を図 10 に示す。誤りの種類はインパタ欠落に限定した。全例で、図 6 (d) の処理を行う時間が全体の約 6~7 割を占めた。図 10 は、追跡入力当りの処理時間 $Y = T/m$ 、回路規模と最大分岐数をパラメータとした回路規模指標 $X = (g * (g + a2^f))$ と置いたときの XY の関係を両対数グラフで表している。ただし、 a には、各点が直線に最も近づく値であった 0.4 を代入している (相関係数 0.96)。図上の点を、 $Y = bX^c$ と近似した場合の c を最小 2 乗法で求めると、 $c = 1.05$ となり、式 (5) にほぼ従うと判断される。

5.4 誤り追跡入力の存在可能性

本手法は誤り追跡入力生成可能であることを前提としている。3.3 節で述べた条件を満たす入力存在せず、誤り追跡入力生成不可能となる機能記述と実回路の組合せは理論上存在する。しかし、実験的には、特殊な場合を除いて追跡入力は生成可能であった。

生成不可能となる典型的な例は、すべての入力パターンが不一致入力となる場合である。この場合、外部出力にインパタを挿入すれば理想回路が得られるため、誤り追跡は必要ない。その他、一致入力と不一致入力の方が極めて少ない場合に誤り追跡入力存在

しない可能性が高くなると考えられるが、5.3 節で示した実験の範囲では、このような例を得ていない。

5.5 複数の論理設計誤りへの対応

本論文では、単一誤りのみを対象としたが、今後は、複数の論理設計誤りが同時に存在する場合に対象を拡張する必要がある。複数誤りに対しても 3 章で述べた誤り追跡入力の性質は失われない。よって、各誤り追跡入力に対する伝搬部分回路内に、誤りが少なくとも 1 つは存在する。現在は、単一誤りを高い限定率で求めるために、図 4 で示したように、全追跡入力に対して誤り候補とされた箇所のみを最終的な誤り候補としている。しかし、複数誤りを想定した場合は、各追跡入力に対して誤り候補とされた頻度を用いて、誤りが限定できる可能性がある。

6. おわりに

本論文では、単一論理設計誤りを対象とし、その自動追跡・修正法提示を目的として誤り追跡入力を用いた追跡システムを開発し、システムの評価を行った。ベンチマーク回路を用いて実験した結果、すべての例で誤りの追跡と修正法の提示が可能であった。これにより、設計者が単一論理設計誤りを含む回路を設計した場合に、修正を短時間で行うことが可能となった。今後の課題として、複数誤りへの対応が挙げられる。

謝辞 本研究を進めるにあたり、日頃より数多くの御指導を頂いた、故小田原豪太郎教授に深く感謝いたします。ここに、先生への深謝の意を記するとともに、謹んで先生の御冥福をお祈り申し上げます。

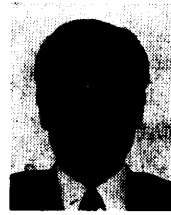
参 考 文 献

- 1) Smith, G. L. et al.: Boolean Comparison of Hardware and Flowchart, *IBM J. Res. Dev.*, Vol. 26, No. 1, pp. 106-116 (1982).
- 2) Odawara, G. et al.: A Logic Verifier Based on Boolean Comparison, *Proc. of 23rd Design Automation Conf.*, pp. 208-214 (1986).
- 3) Wei, R.-S. et al.: PROTEUS: A Logic Verification System for Combinational Circuits, *Proc. of Int. Test Conf.*, pp. 350-359 (1986).
- 4) Fujita, M. et al.: Evaluation and Improvements of Boolean Comparison Method Based on Binary Decision Diagrams, *ICCAD-88*, pp.

- 2-5 (1988).
- 5) Madre, J. C. et al.: Proving Circuit Correctness Using Formal Comparison between Expected and Extracted Behavior, *Proc. 25th Design Automation Conf.*, pp. 205-210 (1988).
- 6) Abadir, M. S. et al.: Logic Design Verification via Test Generation, *IEEE Trans. on CAD*, Vol. 7, No. 1, pp. 138-148 (1988).
- 7) Tamura, K. A.: Locating Functional Errors in Logic Circuits, *Proc. of 26th Design Automation Conf.*, pp. 185-191 (1989).
- 8) Madre, J. C. et al.: Automating the Diagnosis and the Rectification of Design Errors with PRIAM, *ICCAD-89*, pp. 30-33 (1989).
- 9) Brglez, F. and Fujiwara, H.: A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translation in FORTRAN, Special Session on ATPG and Fault Simulation, *IEEE Int. Symp. on Circuits and Systems* (1985).

(平成2年5月7日受付)

(平成3年1月11日採録)



田中 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. 1986年東京大学工学部精密機械工学科助手. 1989年同学科講師. 1990年5月より神戸大学大学院自然科学研究科講師. 工学博士. 専門は論理装置のCAD, ソフトウェア工学. IEEE, プリント回路学会各会員.



蒋 洪海 (正会員)

1961年生. 1982年中国上海交通大学電工及電子計算機科学系卒業. 1984年同大学院修士課程修了. 1985年上海交通大学電氣工学系助手. 1988年東京大学大学院工学系研究科精密機械工学専攻博士課程に入学, 1991年3月修了見込み. 論理設計CADの研究に従事.



山本 保 (学生会員)

1966年生. 1989年東京大学工学部精密機械工学科卒業. 1991年3月, 東京大学大学院工学系研究科修士課程修了見込み. LSIの論理設計支援の研究に従事. IEEE 学生会員.