

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4482622号  
(P4482622)

(45) 発行日 平成22年6月16日(2010.6.16)

(24) 登録日 平成22年4月2日(2010.4.2)

(51) Int.Cl. F I  
GO 1 R 31/3183 (2006.01) GO 1 R 31/28 Q

請求項の数 7 (全 16 頁)

(21) 出願番号	特願2008-522609 (P2008-522609)	(73) 特許権者	503360115 独立行政法人科学技術振興機構 埼玉県川口市本町四丁目1番8号
(86) (22) 出願日	平成19年6月27日(2007.6.27)	(73) 特許権者	504174135 国立大学法人九州工業大学 福岡県北九州市戸畑区仙水町1番1号
(86) 国際出願番号	PCT/JP2007/062929	(73) 特許権者	504125469 株式会社システム・ジェイディー 福岡県福岡市早良区百道浜三丁目8番33号
(87) 国際公開番号	W02008/001818	(74) 代理人	100116573 弁理士 羽立 幸司
(87) 国際公開日	平成20年1月3日(2008.1.3)	(72) 発明者	梶原 誠司 福岡県飯塚市大字川津680番地の4号 国立大学法人九州工業大学内 最終頁に続く
審査請求日	平成21年6月15日(2009.6.15)		
(31) 優先権主張番号	特願2006-180655 (P2006-180655)		
(32) 優先日	平成18年6月30日(2006.6.30)		
(33) 優先権主張国	日本国(JP)		
早期審査対象出願			

(54) 【発明の名称】 変換装置、変換方法、変換方法をコンピュータに実行させることが可能なプログラム、及び、このプログラムを記録した記録媒体

(57) 【特許請求の範囲】

【請求項1】

論理回路に対して予め与えられた遷移遅延故障を検出する複数のテストベクトルを有するテストパターンを前記論理回路に印加する前に論理値が異なるビット構成の前記遷移遅延故障を検出する新たなテストパターンに変換する変換装置において、

前記予め与えられたテストパターンの構成要素は連続して前記論理回路に印加されることが予定された $N$ 個( $N \geq 2$ )のテストベクトルであり、且つ、前記構成要素を構成する $N$ 個( $N \geq 2$ )のテストベクトルは互いのビット間の関係が反転関係に限られず且つそれぞれのビット配列が交互にビット反転するものに限られないテストベクトルであり、

前記構成要素の印加により検出される前記論理回路の遷移遅延故障の検出条件を $N$ 時刻展開回路( $N \geq 2$ )により定め、この検出条件を満たす前記予め与えられたテストパターン中の論理値の組み合わせであってドントケアとできないビットについての論理値の組合せを決定する決定手段を備え、

前記決定手段は前記遷移遅延故障の検出条件である信号値変化前に必要となる初期条件及び変化後の信号値を確認する条件の二つの条件のいずれをも満たすテストパターン中の論理値の組合せを決定する、変換装置。

【請求項2】

前記決定手段により決定された論理値の組合せに含まれないビットの論理値をドントケアとするドントケア決定手段を備えた、請求項1記載の変換装置。

【請求項3】

10

20

論理回路に対して予め与えられた所定の故障を検出する複数のテストベクトルを有するテストパターンを前記論理回路に印加する前に論理値が異なるビット構成の前記故障を検出する新たなテストパターンに変換する変換方法において、

前記予め与えられたテストパターンの構成要素は連続して前記論理回路に印加されることが予定された $N$ 個( $N \geq 2$ )のテストベクトルであり、且つ、前記構成要素を構成する $N$ 個( $N \geq 2$ )のテストベクトルは互いのビット間の関係が反転関係に限られず且つそれぞれのビット配列が交互にビット反転するものに限られないテストベクトルであり、

決定手段が前記構成要素の印加により検出される前記論理回路の故障の検出条件を満たす前記予め与えられたテストパターン中の論理値の組み合わせであってドントケアとできないビットについての論理値の組合せを決定する決定ステップを含み、

前記決定ステップは、前記論理回路の故障の検出条件を $N$ 時刻展開回路( $N \geq 2$ )により定め、この検出条件を満たす前記予め与えられたテストパターン中の論理値の組合せを決定する、変換方法。

#### 【請求項 4】

前記決定ステップは前記決定手段が前記故障の検出条件である信号値変化前に必要となる初期条件及び変化後の信号値を確認する条件の二つの条件のいずれをも満たすテストパターン中の論理値の組合せを決定する、請求項 3 記載の変換方法。

#### 【請求項 5】

前記決定ステップにより前記決定手段が決定した論理値の組合せに含まれないビットの論理値をドントケアとするドントケア決定ステップを含む、請求項 3 又は 4 記載の変換方法。

#### 【請求項 6】

請求項 3 から 5 のいずれかに記載の変換方法をコンピュータに実行させることが可能なプログラム。

#### 【請求項 7】

請求項 6 記載のプログラムをコンピュータが実行することが可能にて記録した記録媒体。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、変換装置、変換方法、変換方法をコンピュータに実行させることが可能なプログラム、及び、このプログラムを記録した記録媒体に関し、特に、論理回路に対して予め与えられたテストパターンを論理値が異なるビット構成のテストパターンに変換する変換装置等に関する。

#### 【背景技術】

#### 【0002】

以下に、従来から提案されてきた縮退故障検出を前提とした、テストパターン内のドントケアの抽出手法について説明する。

#### 【0003】

図 9 は、一般的な論理回路におけるフルスキャン順序回路の模式図である。

#### 【0004】

一般に、半導体論理回路は主に順序回路である。順序回路は、アンド(AND)ゲート、ナンド(NAND)ゲート、オア(OR)ゲート、ノア(NOR)ゲート等の論理素子からなる組合せ回路部 1201 と、回路の内部状態を記憶するフリップフロップ 1203 とを備える。この場合、組合せ回路部 1201 は、外部入力線(PI)、フリップフロップの出力線である擬似外部入力線(PPi)、外部出力線(PO)、フリップフロップの入力線である擬似外部出力線(PPo)を有する。組合せ回路部 1201 への入力は、外部入力線より直接与えられるものと、擬似外部入力線を介して与えられるものがある。また、組合せ回路部 1201 からの出力は、外部出力線に直接現れるものと擬似外部出力線に現れるものがある。

10

20

30

40

50

## 【 0 0 0 5 】

しかし、順序回路のフリップフロップ 1 2 0 3 の出力線（擬似外部入力線）と入力線（擬似外部出力線）は一般に外部より直接アクセスできない。従って、組合せ回路部 1 2 0 1 をテストするためには、擬似外部入力線の可制御性及び擬似外部出力線の可観測性に問題がある。

## 【 0 0 0 6 】

上述の組合せ回路部 1 2 0 1 のテストにおける可制御性及び可観測性の問題を解決する主な手法として、フルスキャン設計がある。フルスキャン設計とは、フリップフロップをスキャンフリップフロップに置き換えた上で、それらを用いて 1 本または複数本のスキャンチェーンを形成することである。スキャンフリップフロップの動作はスキャンインネーブル（SE）信号線で制御される。例えば、SE = 0 のとき、従来のフリップフロップと同じ動作をし、クロックパルスが与えられると、組合せ回路部 1 2 0 1 から値でスキャンフリップフロップの出力値が更新され、また、SE = 1 のとき、同じスキャンチェーンにある他のスキャンフリップフロップと 1 つのシフトレジスタを形成し、クロックパルスが与えられると、外部から新しい値がスキャンフリップフロップにシフトインされると同時に、スキャンフリップフロップに現存の値が外部へシフトアウトされる。一般に、同じスキャンチェーンにあるスキャンフリップフロップは同じスキャンインネーブル（SE）信号線を共有するが、異なるスキャンチェーンのスキャンインネーブル（SE）信号線は同一の場合もあれば異なる場合もある。

## 【 0 0 0 7 】

フルスキャン順序回路の組合せ回路部のテストはスキャンシフトとスキャンキャプチャを繰り返すことによって行われる。スキャンシフトは、スキャンインネーブル（SE）信号が論理値 1 にされているシフトモードで行われる。シフトモードにおいては、1 つまたは複数のクロックパルスが与えられ、外部から 1 つまたは複数の新しい値がスキャンチェーン内のスキャンフリップフロップにシフトインされる。また、それと同時に、そのスキャンチェーン内のスキャンフリップフロップに現存の 1 つまたは複数の値が外部へシフトアウトされる。スキャンキャプチャは、スキャンインネーブル（SE）信号が論理値 0 にされているキャプチャモードで行われる。キャプチャモードにおいては、1 つのスキャンチェーンにあるすべてのスキャンフリップフロップに同時に 1 つのクロックパルスが与えられ、組合せ回路部の擬似外部出力線の値がすべてのスキャンフリップフロップに取り込まれる。

## 【 0 0 0 8 】

スキャンシフトは、擬似外部入力線を介して組合せ回路部 1 2 0 1 へテストベクトルを印加するためと、擬似外部出力線を介して組合せ回路部 1 2 0 1 からテスト応答を観測するために用いられる。また、スキャンキャプチャは、組合せ回路部 1 2 0 1 のテスト応答をスキャンフリップフロップに取り込むために用いられる。すべてのテストベクトルに対して、スキャンシフトとスキャンキャプチャを繰り返すことによって、組合せ回路部をテストすることができる。このようなテスト方式はスキャンテスト方式という。

## 【 0 0 0 9 】

スキャンテスト方式では、組合せ回路部 1 2 0 1 へのテストベクトルの印加は、外部入力から直接行われる部分と、スキャンシフトによって行われる部分とがある。スキャンシフトによって、任意の論理値を任意のスキャンフリップフロップに設定することができるので、擬似外部入力線の可制御性の問題が解決される。組合せ回路部 1 2 0 1 からテスト応答の観測は、外部出力から直接行われる部分と、スキャンシフトによって行われる部分とがある。スキャンシフトによって、任意のスキャンフリップフロップの出力値を観測することができるため、擬似外部出力線の可観測性の問題が解決される。このように、スキャンテスト方式においては、自動テストパターン生成（以下、ATPGとする）プログラムを用いてテストベクトル及び期待テスト応答を求めるだけで十分である。

## 【 0 0 1 0 】

図 1 0 は、テスト入力とテスト応答との関係を示した模式図である。

## 【 0 0 1 1 】

図 1 0 において、テストベクトルにドントケア ( X ) が存在する場合、それによってテスト応答にもドントケアが現れる。ドントケアが存在する原因は、1 つまたは複数の縮退故障を検出するために、テスト入力の一部のビットのみに論理値が決まれば十分であることによる。このようなドントケアをもつテストベクトルは、テストキューブとも呼ばれ、A T P G またはドントケア発見手法によって求められる。ドントケアには、自由に論理値 0 または 1 を割り当てることができる。

## 【 0 0 1 2 】

ドントケアを抽出する方法についての技術がいくつか開示されている ( 非特許文献 1 ~ 非特許文献 7 )。ここでは、非特許文献 1 と非特許文献 2 の技術について簡単に説明する

10

## 【 0 0 1 3 】

非特許文献 1 では、ビット独立と呼ばれる手法に基づいた方法について述べられている。まず、テストベクトル  $v$  は、 $v$  のみで検出可能なすべての故障  $F ( v )$  を決定するために故障シミュレーションが実行される。次に、 $v$  の最初のビットを仮にドントケアとし、3 値故障シミュレーションで  $F ( v )$  が実際に検出可能かどうかを検証する。検出可能であればそのままドントケアとし、そうでなければ本来の値に戻す。これを全ビットについて繰り返すことにより、ドントケアを抽出する。

## 【 0 0 1 4 】

非特許文献 2 では、故障シミュレーションと A T P G の処理手順の一部を用いて、可能な限り多くのドントケアの組合せを求める。ただし、初期テストパターンの故障検出率は維持される。以下に例を用いて簡単に説明する。図 1 1 は、テストパターン中のドントケアを決定する場合の例を示す回路図である。

20

## 【 0 0 1 5 】

縮退故障に対する初期テストパターンが与えられたとすると、その初期テストパターンの中のビットには反対の論理値に変えても故障検出率を低下させないビットがある。そのようなビットはドントケアとすることができる。例えば、図 1 1 において、テストベクトル  $\langle a, b, c, d \rangle = \langle 1, 0, 0, 1 \rangle$  が与えられ、それは信号線  $e$  の 1 縮退故障のみを検出するテストベクトルであるとする。テストベクトル  $\langle a, b, c, d \rangle = \langle 1, 0, 0, 1 \rangle$  は、信号線  $( a, b, c, d )$  に値  $( 1, 0, 0, 1 )$  がそれぞれ印加されることを表す。テストベクトル  $\langle a, b, c, d \rangle = \langle 1, 0, 0, 1 \rangle$  が信号線  $e$  の 1 縮退故障のみを検出できればよいため、信号線  $b$  または  $c$  の論理値 0 は不必要であり、どちらかの論理値はドントケアとすることができる。縮退故障を検出する条件は、故障の活性化と故障の伝搬を保証することである。

30

## 【 0 0 1 6 】

まず、図 1 1 の回路を故障の活性化について見てみると、信号線  $e$  の 1 縮退故障を検出するには信号線  $e$  に論理値 0 を割り当てる必要がある。そのため信号線  $b$  または  $c$  のどちらか一方を論理値 0 にすることが必要であるが、もう一方の信号線は論理値 0 と 1 のどちらの値になっても活性化に影響を与えることはない。よって、その信号線の値はドントケアに変えることができる。この例では、信号線  $c$  に印加する論理値をドントケアとする。

40

## 【 0 0 1 7 】

次に故障の伝搬について見てみると、図 1 1 の回路図では故障箇所  $e$  から外部出力  $( h$  と  $i )$  まで 2 つの伝搬経路  $( e - f - h$  と  $e - g - i )$  がある。このように複数の伝搬経路がある場合、任意の基準で伝搬経路を選択することができる。この例では、伝搬経路  $e - f - h$  を選択する。この伝搬経路を活性化するために、信号線  $a$  には論理値 1 を印加する必要がある。それにより信号線  $e$  の 1 縮退故障は外部出力  $h$  に伝搬でき、検出が可能になる。よって残りの信号線  $d$  はドントケアとすることができる。結果得られたテストキューブ  $\langle a, b, c, d \rangle = \langle 1, 0, X, X \rangle$  はテストベクトル  $\langle a, b, c, d \rangle = \langle 1, 0, 0, 1 \rangle$  と同じように、信号線  $e$  の 1 縮退故障を検出することが可能である。この例のようにして、テストパターン中のいくつかのビットは故障検出率を低下させずにド

50

ントケアに変えることができる場合がある。この非特許文献2の方法は、テストパターン中の全てのビットについてシミュレーションを行なうわけではないので、実行時間においては非特許文献1で提案された手法よりも格段に高速である。

【0018】

【非特許文献1】R.Sankaralingam,N.A.Touba, “Controlling peak power during scan testing”、Proceedings of the IEEE VLSI Test Symposium、pp.153-159、2002.

【非特許文献2】S.Kajihara,K.Miyase, ”On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits”、ICCAD-2001、pp.364-369、Nov.2001.

【非特許文献3】A.El-Maleh,and,A.Al-Suwaiyan, “An efficient test relaxation technique for combinational & full-scan sequential circuits.”、Proceedings of IEEE VLSI Test Symposium、pp.53-59、April 2002.

【非特許文献4】A.El-Maleh,and,K.Al-Utaibi, “An efficient test relaxation technique for synchronous sequential circuits.”、Proceedings of IEEE VLSI Test Symposium、pp.179,185、April 2003.

【非特許文献5】Y.Higami,S.Kajihara,S.Kobayashi,Y.Takamatsu,and,I.Pomeranz, “A method to find don't care values in test sequences for sequential circuits.”、Proceedings of IEEE International Conference on Computer Design、pp.397-399、Oct.2003.

【非特許文献6】B.Koenemann,et.al., “A smartBIST variant with guaranteed encoding.”、Proceedings of 10th Asian Test Symposium、pp.325-330、November 2001.

【非特許文献7】H.-G.Liang,S.Hellebrand,and,H.-J.Wunderlich, “Two-dimensional test data compression for scan-based deterministic BIST.”、Proceedings of IEEE International Test Conference、pp.894-901、November 2001.

【発明の開示】

【発明が解決しようとする課題】

【0019】

近年、半導体回路の大規模微細化に伴い、VLSIのタイミングに関する欠陥が増大しつつある。このため、VLSIのテストにおいても実動作速度での性能を保証することが重要となってきた。タイミングに関する欠陥は、縮退故障モデルのもとで作成される縮退故障検出用テストパターンでは検出できない。そのため、遷移遅延故障モデルやパス遅延故障モデルを用いた実動作速度でのテストが必要である。しかしながら、遅延故障モデルに対する実動作速度テストでは、上述した従来の縮退故障のテスト以上に、テストパターンのサイズが膨大となり、それに伴いテスト印加時間も増大する。また、テスト時の消費電力が増大する問題もある。

【0020】

1つの縮退故障を検出するためには1つのテストベクトルで検出可能だが、1つの遅延故障を検出するためには2つのテストベクトルが必要である。これが縮退故障検出用のテストパターンのサイズに比べて遅延故障検出用のテストパターンのサイズが増大する理由である。テストパターンのサイズが増大する問題を解決するための有力な手掛りとしてテストパターン内のドントケアを利用したテストパターン圧縮手法が提案されている。

【0021】

しかしながら、非特許文献1の技術は、テストパターン中のビットの数だけ故障シミュレーションを繰り返すので、処理時間が故障数、テストベクトル数、及び、テストベクトルのビット数に比例する。したがって、大規模回路への適用は処理時間の面で実用的でない。

【0022】

また、故障シミュレーションとATPGを応用する非特許文献2の手法は、非特許文献

10

20

30

40

50

1 に比べて実行時間が短いのが特徴である。しかし、縮退故障検出を前提とするものであって、遅延故障検出のためにドントケアを抽出する手法はまだ提案されていないのが現状である。

【0023】

上記と同様に非特許文献3～7についても、いずれも縮退故障検出用のテストパターンを対象とする手法であり、遅延故障検出用テストパターンに対しては適用可能性を保障できない。

【0024】

ゆえに、本発明は、上記問題を解決するため、あらかじめ与えられた初期テストパターンを、その構成要素で検出できる遷移遅延故障の故障検出率を下げずに、論理値が異なるビット構成のテストパターンに変換する変換装置、変換方法、変換方法をコンピュータに実行させることが可能なプログラム、及び、このプログラムを記録した記録媒体を提供することを目的とする。

【課題を解決するための手段】

【0025】

請求項1に係る発明は、論理回路に対して予め与えられた遷移遅延故障を検出する複数のテストベクトルを有するテストパターンを前記論理回路に印加する前に論理値が異なるビット構成の前記遷移遅延故障を検出する新たなテストパターンに変換する変換装置において、前記予め与えられたテストパターンの構成要素は連続して前記論理回路に印加されることが予定されたN個(N<sub>2</sub>)のテストベクトルであり、且つ、前記構成要素を構成するN個(N<sub>2</sub>)のテストベクトルは互いのビット間の関係が反転関係に限られず且つそれぞれのビット配列が交互にビット反転するものに限られないテストベクトルであり、前記構成要素の印加により検出される前記論理回路の遷移遅延故障の検出条件をN時刻展開回路(N<sub>2</sub>)により定め、この検出条件を満たす前記予め与えられたテストパターン中の論理値の組み合わせであってドントケアとできないビットについての論理値の組合せを決定する決定手段を備え、前記決定手段は前記遷移遅延故障の検出条件である信号値変化前に必要となる初期条件及び変化後の信号値を確認する条件の二つの条件のいずれをも満たすテストパターン中の論理値の組合せを決定する変換装置である。

【0027】

請求項2に係る発明は、請求項1記載の変換装置であって、前記決定手段により決定された論理値の組合せに含まれないビットの論理値をドントケアとするドントケア決定手段を備える。

【0028】

請求項3に係る発明は、論理回路に対して予め与えられた所定の故障を検出する複数のテストベクトルを有するテストパターンを前記論理回路に印加する前に論理値が異なるビット構成の前記故障を検出する新たなテストパターンに変換する変換方法において、前記予め与えられたテストパターンの構成要素は連続して前記論理回路に印加されることが予定されたN個(N<sub>2</sub>)のテストベクトルであり、且つ、前記構成要素を構成するN個(N<sub>2</sub>)のテストベクトルは互いのビット間の関係が反転関係に限られず且つそれぞれのビット配列が交互にビット反転するものに限られないテストベクトルであり、決定手段が前記構成要素の印加により検出される前記論理回路の故障の検出条件を満たす前記予め与えられたテストパターン中の論理値の組み合わせであってドントケアとできないビットについての論理値の組合せを決定する決定ステップを含み、前記決定ステップは、前記論理回路の故障の検出条件をN時刻展開回路(N<sub>2</sub>)により定め、この検出条件を満たす前記予め与えられたテストパターン中の論理値の組合せを決定する変換方法である。

【0029】

請求項4に係る発明は、請求項3記載の変換方法であって、前記決定ステップは前記決定手段が前記故障の検出条件である信号値変化前に必要となる初期条件及び変化後の信号値を確認する条件の二つの条件のいずれをも満たすテストパターン中の論理値の組合せを決定する。

10

20

30

40

50

## 【0030】

請求項5に係る発明は、請求項3又は4記載の変換方法であって、前記決定ステップにより前記決定手段が決定した論理値の組合せに含まれないビットの論理値をドントケアとするドントケア決定ステップを含む。

## 【0031】

請求項6に係る発明は、請求項3から5のいずれかに記載の変換方法をコンピュータに実行させることが可能なプログラムである。

## 【0032】

請求項7に係る発明は、請求項6記載のプログラムをコンピュータが実行することが可能にて記録した記録媒体である。

10

## 【発明の効果】

## 【0033】

本発明によれば、遷移遅延故障検出用のテストパターンを圧縮により削減する有効な手段を実現しているため、遷移遅延故障検出用のテストパターンのサイズが膨大になるという問題が改善され、遷移遅延故障のテスト印加時間を短縮することができる。

## 【0034】

また、本発明によれば、テスト時の消費電力量が削減されるようなテストパターンに変換することで、テスト時の消費電力量を削減できる。したがって、歩留まりの低下を防ぐのに役立つ。

## 【0035】

さらに、本発明によれば、予め与えられた初期テストパターンの故障検出率を低下させずに、テストパターンを圧縮する等の新しい特徴を持たせることができる。

20

## 【0036】

さらに、本発明によれば、遅延故障検出用のテストパターンのサイズが小さくなるため、従来はテストパターンのサイズが膨大なためにテスト印加時間が増大し、十分にテストできなかった遅延故障に対するテストが可能となる。従って、これまでテストが不十分のため本来不良品なのに良品として出荷していた半導体装置のテストが可能となるため、半導体装置の品質が向上する。

## 【0037】

さらに、本発明の手法は、スキャン回路の方式及びクロックの種類、クロックの数に依存しないため、フルスキャン設計を適用した半導体装置全般に対して適用できる。すなわち、複数本のスキャンチェーンが存在する半導体装置、スキャン回路を駆動するクロックが複数個存在する半導体装置、または、スキャン回路に組合せ回路部分からフリップフロップに値を取り込むタイミングが異なる半導体装置に対しても適用可能である。

30

## 【図面の簡単な説明】

## 【0038】

【図1】本発明の概要を示す図である。

【図2】被検査対象回路を2時刻展開した場合の回路構造を示す図である。

【図3】本発明の実施の形態に係る変換装置及び圧縮装置の構成の一例を示したブロック図である。

40

【図4】本発明の実施の形態に係る変換装置及び圧縮装置の処理の一例を示したフロー図である。

【図5】テストパターンの論理値の組合せの決定方法及びドントケアの決定方法の一例を説明するための簡単な回路図である。

【図6】テスト圧縮の処理の例を示す図である。

【図7】初期テストパターンに対してドントケアの決定を行った実験結果を示す表である。

【図8】図7に示す実験でドントケアが決定されたテストパターンに対して、テスト圧縮を行った場合の実験結果を示す表である。

【図9】一般的な論理回路におけるフルスキャン順序回路の模式図である。

50

【図10】テスト入力とテスト応答との関係を示した模式図である。

【図11】テストパターン中のドントケアを決定する場合の例を示す回路図である。

【符号の説明】

【0039】

100a 初期テストパターン

100b 中間テストパターン

【発明を実施するための最良の形態】

【0040】

本発明についての概要を説明する。本発明は、VLSIのテストにおいて実動作速度での性能を保証するためのテスト手法であり、本質的には予め与えられた初期テストパターンの各テストベクトルで検出できる遷移遅延故障の情報をもとにしてテストパターン内にドントケアを決定する手法であり、故障検出率を下げずに各テストベクトルをマージし、与えられたテストパターンを圧縮する等の応用が可能である。また、テストパターンを圧縮することに使用されるだけでなく、スキャンフリップフロップのキャプチャ動作時の消費電力の増大に伴う電源電圧降下に起因する誤テスト回避のためのテストパターンの変換など様々な目的に使用可能である。

10

【0041】

図1は、本発明の概要を示す図である。

【0042】

図1において、初期テストパターン100aは、予め与えられたATPG等の手法により生成されたテストデータである。初期テストパターン100aは、複数の構成要素を備え、各構成要素は少なくとも2つのテストベクトルを備える。図1においては、2つのテストベクトルを備える。1つの遷移遅延故障を検出するには、このように連続する2つのテストベクトルが必要である。初期テストパターン100aは、変換装置100により、図に示すように、故障の検出条件を満たす論理値の組合せ(図中に示す)が決定され、同時にドントケア(図中に示す)が決定される。

20

【0043】

変換装置100は、中間テストパターン100bを生成する。中間テストパターン100bは、複数の構成要素を備え、各構成要素は2つのテストキューブを備える。

【0044】

論理値割当装置101は、生成された中間テストパターン100bのドントケアに対して論理値を割り当てる。論理値割当装置101は、具体的にはテストデータを圧縮する装置や、テスト時の消費電力を削減するための装置である。それぞれの目的に応じて適した論理値をドントケアに割り当てる。

30

【0045】

全てのドントケアに論理値が割り当てられたテストデータは、最終テストパターン100cとして変換される。最終テストパターン100cは、初期テストパターン100aと同様に、複数の構成要素を備え、各構成要素は2つのテストベクトルを備える。しかし、データの内容は初期テストパターン100aとは異なるものとなっている。この最終テストパターン100cは、初期テストパターン100aと故障検出率は同じであるが、圧縮によりデータ量が削減されていたり、テスト時の消費電力が削減できるようなテストパターンに変換されている。

40

【0046】

次に、本発明の実施の形態について説明する。

【0047】

本発明では、遷移遅延故障を検出するテストパターンの変換を行うが、遷移遅延故障を検出する方法として、2つの方法が広く知られている。一方はlaunch-off-shift方式であり、もう一方は、ブロードサイドテスト(launch-off-capture方式)である。本発明では、遷移遅延故障を検出するためにブロードサイドテスト(launch-off-capture方式)を用いる場合を想定する。遷移遅延

50



故障モデルに対する実動作速度テストに広く使われ始めたブロードサイドテストについて概説する。実動作速度テストでは、クロックのタイミングの制御が容易であるためブロードサイドテストが使用される。遅延テストは通常2パターンテストを必要とする。始めのパターンは*initialization*パターンと呼ばれ、次のパターンは*launch*パターンと呼ばれる。ブロードサイドテストを用いた場合、*launch*パターンは*initialization*パターンに対するテスト応答から導出する。つまり、*launch*パターンは*initialization*パターンに大きく依存していることになる。それゆえ、必要な*launch*パターンが導出できない場合が起こり得る。しかし、ブロードサイドテストはクロックタイミングの制御が*launch-off-shift*方式のテストよりも容易であるので、遅延故障を検出するためにブロードサイドテストの  
10  
ほうが一般に用いられている。ブロードサイドテストを用いた遷移遅延故障を、背景技術に示す縮退故障の検出の技術と同様の方式で取扱うために2時刻展開回路が用いられる。

【0048】

本発明では、ブロードサイドテストに基づいた遷移遅延故障用の初期テストパターンに対して、ドントケアを決定する。まず、遷移遅延故障を背景技術に示す縮退故障の検出の技術と同様に取扱うための2時刻展開について説明する。これにより縮退故障検出用のテストパターンに対するドントケアの決定を遷移遅延故障検出用のテストパターンに対するドントケアの決定に拡張することができる。

【0049】

ブロードサイドテストを用いた場合にドントケアの決定を行うために、被検査対象回路を2時刻に展開する。図2に2時刻展開した場合の回路構造を示す。2時刻展開回路は、一つの回路の各時刻の論理動作を空間的に展開したものである。展開した回路ではクロックの概念がなくなり、一つの大きな組合せ回路のように表される。1時刻目は*initialization*段階として取扱うことができ、2時刻目は*launch*段階として取扱うことができる。*launch*パターンは*initialization*パターンから導出される。  
20

【0050】

次に2時刻展開回路を用いた場合、遅延故障の検出が可能となるための条件を定義する。この遅延故障の検出の条件は明確に定式化できる。この定式を用いると、遷移遅延故障用の初期テストパターンに対してもドントケアを決定することができる。  
30

【0051】

遷移遅延故障の検出条件について説明する。信号線*a*の遷移遅延故障*f*を検出するためには、信号線*a*の2時刻目に相当する信号線*a*2で縮退故障の検出が必要である。2時刻目の回路で故障の活性化を保証するため、信号線*a*の値は1時刻目と2時刻目で反対の論理値でなければならない。そのため1時刻目の信号線*a*1には、*a*2とは反対の論理値を割当てて。 *a*2に1縮退故障を仮定した場合の故障の検出は、立上り遷移遅延故障の検出と見なせる。*a*2に0縮退故障を仮定した場合の故障の検出は、立下り遷移遅延故障の検出と見なせる。それゆえ、遷移遅延故障の検出条件は以下のように定式化することができる。  
40

【0052】

信号線*a*の立上り故障*f*の検出条件

- (1) 信号線*a*2の0縮退故障を検出する。
- (2) 信号線*a*1に*a*2と反対の論理値割り当てをする。

【0053】

信号線*a*の立下り故障*f*の検出条件

- (1) 信号線*a*2の1縮退故障を検出する。
- (2) 信号線*a*1に*a*2と反対の論理値割り当てをする。

【0054】

上記背景技術で示した縮退故障検出の手法に、上記の検出条件を加えることで、遷移遅延故障用の初期テストパターンに対して、その故障検出率を維持したままドントケアの決  
50

定が行えるようになる。

【 0 0 5 5 】

なお、上記条件とは、上記(1)、(2)の条件を満たしさえすれば、対象となる遷移遅延故障の検出ができる条件である。

【 0 0 5 6 】

次に、2時刻展開回路中には構造的に検出が不可能であると分かる故障が存在するため、それらはあらかじめ対象故障から除外する。この対象故障の削減は本発明において処理時間の短縮に直接つながる。

【 0 0 5 7 】

実速度テストの場合、launchパターンの導出に制約が存在する場合がある。たとえば、1時刻目と2時刻目で外部入力に印加する論理値を変化させないことである。これにより、1時刻目と2時刻目の値が常に同じになる信号線がある。これは外部入力に印加される論理値によってのみ値が導出される信号線が存在することにより起こる。それゆえ、少なくとも1つ以上の擬似外部入力に印加される論理値により信号線値が導出される場合のみ、その信号線の遷移遅延故障が検出される可能性があると考えられる。その他の故障はこの制約のもとで検出されることは不可能である。そのため、本発明においては、構造的に検出不可能な故障があるため、そのような故障は対象故障からは除外する。これにより本発明での検出対象故障は、前提とする実速度テスト環境下で検出される可能性のある故障のみとなる。

【 0 0 5 8 】

以下に、本発明の実施の形態に係る変換装置及び圧縮装置について説明する。

【 0 0 5 9 】

図3は、本発明の実施の形態に係る変換装置及び圧縮装置の構成の一例を示したブロック図である。

【 0 0 6 0 】

変換装置200は、メモリ部210と処理部220と入出力インターフェース240とデータ記憶部250とを備える。メモリ部210は、故障リスト部212とテストパターン部216とを備える。処理部220は、故障抽出部222とテストパターン変換部225とを備える。テストパターン変換部225は、テストパターン論理値組合せ決定部226とドントケア決定部232と分布調整部230とを備える。データ記憶部250は、初期パターン252と中間テストパターン254を備える。

【 0 0 6 1 】

圧縮装置201は、変換装置200と同様に、メモリ部260と処理部270と入出力インターフェース280とデータ記憶部290とを備える。メモリ部260は、マージ情報部262を備える。処理部270は、マージ処理部272と論理値割当部274とを備える。データ記憶部290は、最終テストベクトル集合292を備える。

【 0 0 6 2 】

変換装置200と圧縮装置201の処理の概要を簡潔に説明する。ATPG等の手法により生成された初期テストパターン252は、変換装置200の入出力インターフェース240を介して故障抽出部222に入力される。故障抽出部222は、入力された初期テストパターン252の故障リストを抽出し、メモリ部210の故障リスト部212に保持する。テストパターン変換部225は、故障リスト部212に保持された情報を基に、ドントケアの分布を調整しながらテストパターンの論理値の組合せを決定する。その決定されたテストパターンはテストパターン部216に保持される。そして、テストパターン部216に保持された情報を基に、ドントケア決定部232によりドントケアが決定され、ドントケアを含む中間テストパターン254がデータ記憶部250に保持される。

【 0 0 6 3 】

圧縮装置201は、変換装置200により生成された中間テストパターン254のデータをデータ記憶部290に記憶しており、入出力インターフェース280を介して処理部270のマージ処理部272に入力される。マージ処理部272は、入力されたデータを

10

20

30

40

50

基にテストベクトルをマージし、そのマージされたデータは、メモリ部 260 のマージ情報部 262 に保持される。論理値割当部 274 は、マージ情報部 262 が保持する情報のうちドントケアに論理値を割り当てて、最終テストパターン 292 が生成される。

【0064】

図 4 は、本発明の実施の形態に係る変換装置及び圧縮装置の処理の一例を示したフロー図である。

【0065】

まず、ステップ ST301 で、図 3 の初期テストパターン 252 のデータを用いて故障抽出部 222 により検出の対象となる対象故障が抽出される。抽出された対象故障のリストは故障リスト部 212 に保持される。ステップ ST302 では、故障リスト部 212 に保持された情報に基づいて、テストパターン論理値組合せ決定部 226 により、テストパターンの論理値の組合せが決定される。同時に、分布調整部 230 により対象故障の分布の調整が行われる。ステップ ST303 では、ドントケア決定部 232 が、ステップ ST302 で決定された論理値の組合せに含まれないビットの論理値をドントケアとして決定する。

10

【0066】

ここで、テストパターンの論理値の組合せの決定及びドントケアの決定について詳細に説明する。

【0067】

図 5 は、テストパターンの論理値の組合せの決定方法及びドントケアの決定方法の一例を説明するための簡単な回路図である。図 5 (a) の回路に対して、図 5 (b) は 2 時刻展開した場合の回路である。この図 5 の回路を用いて、テストパターンに対してどのようにドントケアを決定するかを示す。

20

【0068】

まずテストベクトル  $\langle a_1, b_1, c_1, d_1 \rangle = \langle 1, 1, 0, 0 \rangle$  が、外部入力 (a, c, d) と疑似外部入力 (b) に対して印加される場合を考える。このテストベクトル  $\langle a_1, b_1, c_1, d_1 \rangle = \langle 1, 1, 0, 0 \rangle$  の検出対象故障は、信号線 b の立下り遷移遅延故障のみとする。立下り遷移遅延故障を検出するために、1 時刻目の信号線 b 1 には論理値 1 を割当てなければならない。信号線 b 2 で 1 縮退故障の活性化とその伝搬を保証しなければならない。故障の影響を外部出力まで伝搬するために、信号線 a 2 には論理値 1 を割当てなければならない。また信号線 b 2 の 1 縮退故障の活性化には信号線 b 2 を論理値 0 にしなければならないので、信号線 c 1 か d 1 に論理値 0 を割当てる必要がある。そして残りの信号線に印加される論理値はドントケアとすることができる。この例では、信号線 c 1 に論理値 0 を割当てている。信号線 d 1 の値はドントケアに変えることができる。テストキューブ  $\langle a_1, b_1, c_1, d_1 \rangle = \langle 1, 1, 0, X \rangle$  はテストベクトル  $\langle a_1, b_1, c_1, d_1 \rangle = \langle 1, 1, 0, 0 \rangle$  から導出でき、かつ同じ故障を検出する。もし信号線 b の立下り遷移遅延故障を検出するテストベクトル  $\langle a_1, b_1, c_1, d_1 \rangle = \langle 1, 0, 1, 1 \rangle$  に対してドントケアの決定を行ったとすると、それは 1 つのドントケアも見つけずに終了する。その理由はテストベクトルの全ての論理値が信号線 b の立下り遷移遅延故障を検出するのに必要であるからである。

30

40

【0069】

図 4 に戻って、ステップ ST303 でドントケアが決定され、ドントケアを含む中間テストパターン 254 が生成されると、ステップ ST304 で、図 3 の圧縮装置 201 のマージ処理部 272 によりマージ処理が行われる。このマージ処理により中間テストパターン 254 は圧縮されてマージ情報部 262 に保持される。

【0070】

ここで、テスト圧縮について説明する。

【0071】

中間テストパターン中の決定されたドントケアは様々な用途に用いることができる。その有効な利用法のひとつに、テスト圧縮にドントケアを利用する方法がある。テストパタ

50

ーンを圧縮することはテストパターンのサイズを削減し、それによりテスト印加時間も同様に削減できるため、非常に有効な利用法である。

【 0 0 7 2 】

ドントケアを含む中間テストパターンが与えられると、いくつかのテストキューブは故障検出率を低下させることなくマージすることができる。テスト圧縮を行うと、2つかそれ以上のテストキューブがマージされ、1つのテストキューブとして扱うことができ、それによりテストパターンのサイズが削減できる。ドントケアを含むテストパターンがどのようにマージされるかを、図6を用いて説明する。

【 0 0 7 3 】

図6は、テスト圧縮の処理の例を示す図である。

10

【 0 0 7 4 】

テストキューブ  $t_1 < 0, X, 1, X >$  はテストキューブ  $t_3 < X, 1, 1, X >$  とマージすることが可能である。2つのテストキューブをマージしてテストキューブ  $< 0, 1, 1, X >$  が導出される。そして、そのテストキューブ  $< 0, 1, 1, X >$  はさらにテストキューブ  $t_6 < 0, 1, X, 0 >$  とマージすることができ、テストキューブ  $t'_1 < 0, 1, 1, 0 >$  が導出される。図6に示されるようにテストキューブ  $t'_1 < 0, 1, 1, 0 >$  はテストキューブ  $t_1, t_3, t_6$  によって検出される全ての故障を検出できる。よって故障検出率を低下させることなくテストキューブ数を削減することができる。同様の方法で、テストキューブ  $t'_2 < 1, 1, 0, 1 >$  はテストキューブ  $t_2 < X, 1, 0, 1 >$  とテストキューブ  $t_4 < 1, X, 0, X >$  をマージすることで導出される。テストキューブ  $t'_3 < 1, 0, 1, X >$  はテストキューブ  $t_5 < 1, 0, 1, X >$  と同一のものである。なぜならテストキューブ  $t_5$  はマージすることができるテストキューブがないため、そのまま  $t'_3$  として残されている。そして、最終的にテストパターン  $(t'_1, t'_2, t'_3)$  が求まる。このテストパターンの故障検出率は、テストパターン  $(t_1, t_2, t_3, t_4, t_5, t_6)$  と同じである。しかし、テストキューブ数は6から3に削減されているので、テストパターンのサイズは元のテストパターンの半分に削減することができる。

20

【 0 0 7 5 】

他にも、より高い圧縮の効果の得られるテストキューブをマージする組合せが存在するかもしれないが、最も良いマージの組合せを見つけることはNP-完全問題である。そして良い組合せを見つける有効なヒューリスティックなアルゴリズムも存在するが、それでも多くの計算時間を要する。従って、本発明の実施の形態では、テストパターンを単純に上から降順に探索し、マージ可能なものをマージしていく方法で行った。この処理であれば  $O(n^2)$  で終了し、比較的短い時間で良い圧縮の効果を得ることができる。

30

【 0 0 7 6 】

図4に戻って、ステップST305で、ドントケアに論理値が割り当てられて最終テストパターン292が生成されて処理を終了する。

【 0 0 7 7 】

以下に、本発明の実験結果を説明する。

【 0 0 7 8 】

本実験では、PC (OS: FreeBSD 4.11 Release, CPU: Athlon XP 1800+ 1.53GHz, memory: 768MB) 上にCプログラミング言語を用いて実装し、ISCAS'89ベンチマーク回路に対して行った。実験で使用したテストパターンは、ブロードサイドテストを用いて遷移遅延故障に対して生成されたテストパターンを使用する。そのテストパターンは以下の方法で導出している。なお、1時刻目と2時刻目で外部入力に印加する論理値を変化させないことはできず、また、故障検出は疑似外部出力でのみ可能で外部出力ではできずとする。

40

【 0 0 7 9 】

まず1つの対象故障を選択し、その故障に対してテストパターンの生成を行い、生成したテストパターンで故障シミュレーションを実行する。これにより、そのテストパターン

50

で検出される故障はテスト生成を行う必要はなくなり、テスト生成を行う回数を少なくする。この処理をテスト生成する必要があり、且つ、未処理の対象故障がなくなるまで実行することにより、テストパターンを導出している。また、対象故障集合は、initialization段階とlaunch段階で値が変わる可能性のある信号線上の故障のみで構成される。その他の故障はブロードサイドテストを用いる限り、常に検出不可能な故障として対象故障集合からは除外する。

#### 【0080】

図7は、初期テストパターンに対してドントケアの決定を行った実験結果の表を示す。図7の表中の最初の列は回路名を示す。それに続く3列は、外部入力数、擬似外部入力数、与えられたテストパターンのテストベクトル数をそれぞれ示す。次の3列はそれぞれのテストパターン中で決定されたドントケアの平均の割合、最も多くのドントケアを含むテストベクトル中のドントケアの割合、最も少ないドントケアを含むテストベクトル中のドントケアの割合を、それぞれ%で示す。続く2列は、対象故障数と与えられたテストパターンで検出された故障数をそれぞれ示す。最後の列は、CPUの処理時間を秒で示す。

10

#### 【0081】

与えられたテストパターンに対して、本発明ではテストパターン中に74%以上のドントケアを決定している。また、列“%X-max”はドントケア決定後、全てドントケアで構成されたテストキューブを含んでいることを示す。そのようなテストキューブは削除するか、または、テスト品質を高めるために用いることが可能である。また、列“%X-min”は、いくつかの回路のテストパターンが全くドントケアを含まないテストベクトルを含んでいることと示す。このようなテストベクトルでは、全ての入力値が遷移遅延故障を検出するために用いられていることを意味する。CPU処理時間は、与えられたテストパターンのテストベクトル数と回路規模に依存する。それは本発明の方法がテストベクトルをシリアルに処理しているからである。

20

#### 【0082】

図8は、図7に示す実験でドントケアが決定されたテストパターンに対して、テスト圧縮を行った場合の実験結果の表を示す。この実験で、テスト圧縮には上記で示したように単純に上から降順に探索する方法を用いている。図8の表の最初の列は回路名を示す。次の列“#tests”は、圧縮前と圧縮後のテストパターン中のテストベクトル数を示す。列“%(圧縮率)”は与えられたテストパターンに対するテスト圧縮後のテストベクトル数の割合を%で示す。この結果は与えられたテストパターンを67%まで圧縮することができたことを示す。CPU処理時間については、図8に示される大規模回路に対しても数秒の計算時間で終了する。

30

#### 【0083】

上記実験結果から、本発明の方法がテストパターン中に平均して74%以上のドントケアを決定できたことを示した。また、実験結果はドントケアが決定されたテストパターンに対して良いテスト圧縮の効果を示せた。それはテストパターン中にドントケアを決定することの有効性を示している。

#### 【0084】

なお、上記では2時刻展開回路を用いて、2つの連続して印加されるテストベクトルにより遅延故障を検出しているが、2時刻展開に限らずN時刻展開回路を用いてもよい。また、N個の連続して印加されるテストベクトルにより故障を検出してもよい。その場合の故障は、遅延故障に限定されない。

40

#### 【0085】

また、本発明の実施の形態では、遷移遅延故障の検出を行っているが、パス遅延故障の他、クロストーク、スタックオープン、ブリッジ等、複数テストパターンで検出できる故障であれば応用可能である。

#### 【0086】

さらに、上記条件を満たすテストパターンの論理値の組合せは、そのテストパターンの全てであっても、一部であってもよい。

50

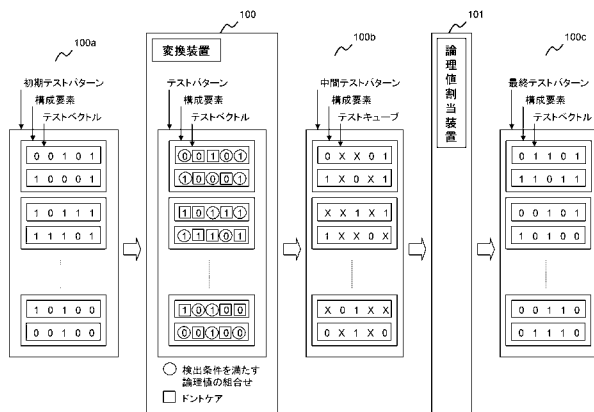
【0087】

さらに、変換装置と圧縮装置を分けて示しているが、変換装置と圧縮装置を1つの装置と捉えてもよい。

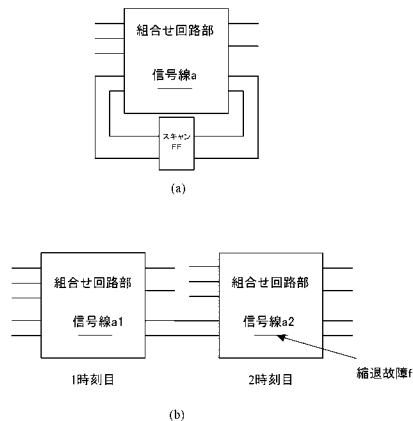
【0088】

さらに、上記実施の形態における圧縮装置は、テスト時の最大電力を削減するようにドントケアに論理値を割り当てるための装置やテスト時の平均電力を削減するようにドントケアに論理値を割り当てるための装置や対象故障をN回検出するようにドントケアに論理値を割り当てる装置等であってもよい。この場合、これらの装置と変換装置を1つの装置と捉えてもよい。

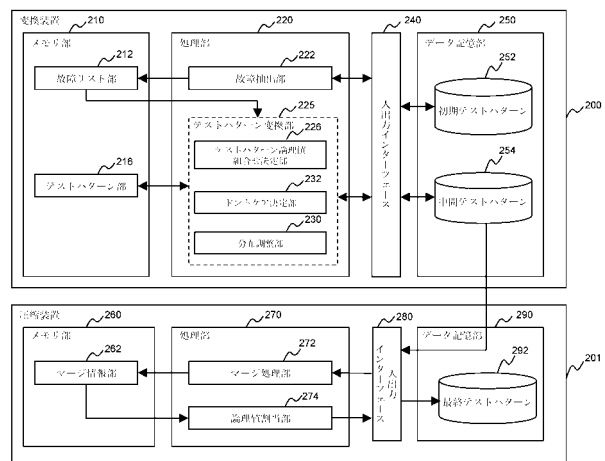
【図1】



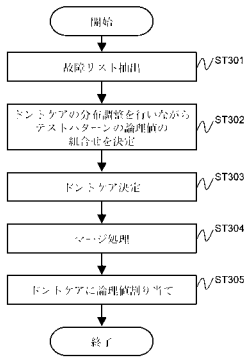
【図2】



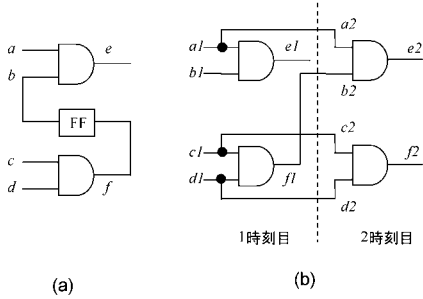
【図3】



【図 4】



【図 5】



【図 6】

テスト キューブ	検出対象 故障	テスト キューブ	検出対象 故障
$t_1$ 0x1x	$f_1, f_5$	$t'_1$ 0110	$f_1, f_3, f_5, f_9, f_{10}$
$t_2$ x101	$f_2, f_4$	$t'_2$ 1101	$f_2, f_4, f_6$
$t_3$ x11x	$f_3$	$t'_3$ 101x	$f_7, f_8$
$t_4$ 1x0x	$f_6$		
$t_5$ 101x	$f_7, f_8$		
$t_6$ 01x0	$f_9, f_{10}$		

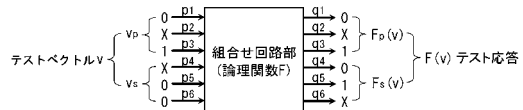
【図 7】

故障名	件数	100%	10%	*%max	*%min	%min	検出率	検出割合	検出率
<510>	19	6	79	75.13	100	49	660	571	0.83
<520>	3	21	86	56.3	100	25	474	555	0.83
<641>	35	19	57	75.54	100	55.56	356	288	0.81
<713>	35	19	54	74.51	100	59.26	512	560	0.83
<820>	18	5	85	59.51	100	64.78	786	531	0.64
<852>	18	5	81	58.62	100	54.78	824	558	0.63
<858>	35	32	55	85.16	100	62.69	780	406	0.82
<953>	16	29	94	75.63	100	22.22	1254	1041	0.27
<1106>	14	18	34	66.64	100	31.25	716	398	0.83
<1218>	14	18	36	66.93	100	31.25	716	398	0.83
<1423>	17	74	166	69.83	100	17.98	2244	3925	0.2
<1488>	8	6	131	57.73	100	0	1309	3069	0.33
<1494>	8	6	129	54.5	100	0	1226	3080	0.33
<1538>	35	179	862	85.08	100	19.63	4924	4398	0.76
<1654>	19	228	992	86.62	100	46.36	16949	9632	31.3
<1337>	31	669	1636	96.51	100	38.71	14796	13759	23.64
<15888>	14	597	372	94.56	100	44.52	17568	12544	42.69
<15932>	35	1728	199	91.4	100	2.32	53340	44012	39.46
<18417>	36	1636	3260	96.35	100	51.67	49988	48016	38.564
<18984>	12	1452	2411	96.91	100	26.81	52812	43301	372.26
<合計>				74.04	100.00	52.65			

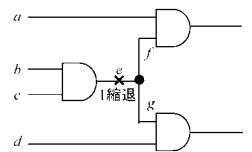
【図 8】

故障名	#tests		平均検出率
	検出割合	検出率	
<510>	78	43	78.2
<520>	961	45	73.3
<641>	57	35	61.1
<713>	51	31	61.0
<820>	83	47	81.7
<852>	81	45	81.5
<858>	55	32	61.0
<953>	51	71	78.7
<1106>	31	29	71.5
<1218>	36	27	77.5
<1423>	195	111	68.7
<1488>	131	103	78.6
<1494>	129	100	82.2
<1538>	192	272	67.7
<1654>	162	439	62.1
<18417>	1469	555	54.9
<18490>	772	178	61.9
<18532>	133	72	37.3
<18417>	2830	1452	57.8
<18784>	2113	1459	69.2
合計			67.0

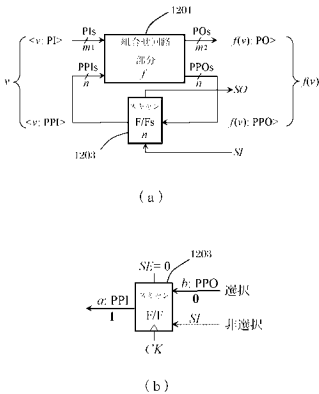
【図 10】



【図 11】



【図 9】



---

フロントページの続き

- (72)発明者 宮瀬 紘平  
福岡県福岡市早良区昭代一丁目6番27-402号
- (72)発明者 温 暁青  
福岡県飯塚市大字川津680番地の4号 国立大学法人九州工業大学内
- (72)発明者 皆本 義弘  
福岡県北九州市八幡西区三ヶ森二丁目8番18-403号
- (72)発明者 伊達 博  
福岡県福岡市早良区百道浜三丁目8番33号 株式会社システム・ジェイディー内

審査官 神谷 健一

- (56)参考文献 特開2001-004710(JP,A)  
Seiji Kajihara, Kenjiro Taniguchi, Kohei Miyase, Irith Pomeranz, Sudhakar M. Reddy, Test Data Compression Using Don't-Care Identification and Statistical Encoding, Proceedings of the 11th Asian Test Symposium(ATS'02), 米国, IEEE, 2002年11月20日, pp. 67 - 72
- (58)調査した分野(Int.Cl., DB名)  
G01R 31/28-31/3193