

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-155339

(P2007-155339A)

(43) 公開日 平成19年6月21日(2007.6.21)

(51) Int. Cl.	F I	テーマコード (参考)
GO 1 R 31/3183 (2006.01)	GO 1 R 31/28	2 G 1 3 2
GO 1 R 31/28 (2006.01)	GO 1 R 31/28	G

審査請求 未請求 請求項の数 16 O L (全 25 頁)

(21) 出願番号	特願2005-346613 (P2005-346613)	(71) 出願人	503360115 独立行政法人科学技術振興機構 埼玉県川口市本町4丁目1番8号
(22) 出願日	平成17年11月30日(2005.11.30)	(71) 出願人	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
		(71) 出願人	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
		(74) 代理人	100116573 弁理士 羽立 幸司
		(72) 発明者	温 暁青 福岡県飯塚市大字川津680-4 九州工業大学内

最終頁に続く

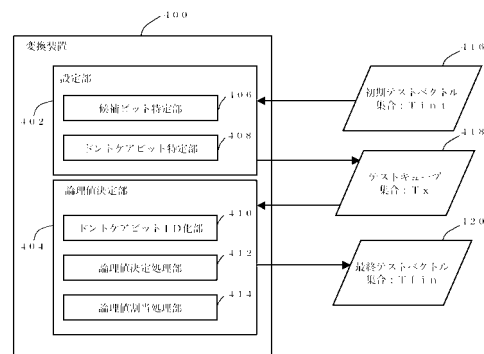
(54) 【発明の名称】 変換装置、変換方法、変換方法をコンピュータに実行させることが可能なプログラム、及び、このプログラムを記録した記録媒体

(57) 【要約】

【課題】 フルスキャン順序回路に含まれるスキャンセルの出力について、スキャンキャプチャの前後において発生する論理値の相違が低減されるようにテストベクトル集合の変換を行う変換装置等を提供する。

【解決手段】 フルスキャン順序回路に対するテストベクトル集合を変換する変換装置400であって、テストベクトル集合の各テストベクトルからドントケアとできるドントケアビットを特定するために、所定の制約条件であって論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビット、及び、ドントケアビットにしてはならない固定ビットを設定する設定部402と、前記設定部402により特定されたドントケアビットを含むテストキューブに対して、複数のビットペア間の関係を考慮して、論理値をドントケアビットに決定する論理値決定部404とを備える。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

論理回路に対するテストベクトル集合を変換する変換装置であって、
前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビット、及び、ドントケアビットにしてはならない固定ビットを設定する設定手段と、

前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定手段とを備えた、変換装置。

10

【請求項 2】

前記決定手段は、前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブに代えて他の手段で得られたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する、請求項 1 記載の変換装置。

【請求項 3】

論理回路に対するテストベクトル集合を変換する変換装置であって、
前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビットを設定する設定手段を備えた、変換装置。

20

【請求項 4】

前記設定手段は、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、前記候補ビットを設定することに加えて、ドントケアビットにしてはならない固定ビットを設定する、請求項 3 記載の変換装置。

【請求項 5】

前記論理回路はフルスキャン順序回路であり、
前記制約条件は、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後の論理値の相違数が削減されるように定められる条件である、請求項 1 から 4 のいずれかに記載の変換装置。

30

【請求項 6】

論理回路に対するテストベクトル集合を変換する変換装置であって、
前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定手段を備えた、変換装置。

40

【請求項 7】

前記論理回路はフルスキャン順序回路であり、
前記目的は、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後の論理値の相違数が効果的に削減されることである、請求項 1、2 又は 6 記載の変換装置。

【請求項 8】

論理回路に対するテストベクトル集合を変換する変換方法であって、
設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビッ

50

トにしてもよい候補ビット、及び、ドントケアビットにはならない固定ビットを設定する設定ステップと、

決定手段が、前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブ、またはその他の手段で得られたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定ステップとを含む、変換方法。

【請求項 9】

論理回路に対するテストベクトル集合を変換する変換方法であって、

設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビットを設定する設定ステップを含む、変換方法。

10

【請求項 10】

前記設定ステップは、前記設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、前記候補ビットを設定することに加えて、ドントケアビットにはならない固定ビットを設定するステップを含む、請求項 9 記載の変換方法。

【請求項 11】

前記論理回路はフルスキャン順序回路であり、

20

前記設定ステップは、

前記テストベクトル集合の各テストベクトルに対して、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後で論理値の相違が発生するビットと相違が発生しないビットとを識別する第 1 のステップと、

前記相違が発生しないビットを固定ビットとする第 2 のステップと、

前記相違が発生するビットのうちから所定の条件を満たした前記候補ビットを特定するとともに前記候補ビットに特定されなかった残余の前記相違が発生するビットを固定ビットとする第 3 のステップとを含み、

前記候補ビットからドントケアとできるドントケアビットを特定し、前記固定ビットからドントケアビットを特定しないことを特徴とする、請求項 8 又は 10 に記載の変換方法

30

【請求項 12】

前記第 3 のステップは、

各テストベクトルにおける候補ビットの予定総数を決定するステップと、

前記相違が発生するすべてのビットに対してドントケアを割り当て、3 値論理シミュレーションを行って、その結果によって、入力ビットにドントケアがありそれに対応する出力ビットにドントケアがないタイプ 1 のドントケアビットペア、及び、入力ビットとそれに対応する出力ビットの両方にドントケアがあるタイプ 2 のドントケアビットペアを特定するステップと、

前記特定されたタイプ 1 のドントケアビットペアを優先的に候補ビットとして選択し、タイプ 1 のドントケアビットペアがなくなりかつ既に選択された候補ビットの数が予定総数に達していない場合、タイプ 2 のドントケアビットペアを所定の基準に従って候補ビットとして選択するステップを含む、請求項 11 記載の変換方法。

40

【請求項 13】

論理回路に対するテストベクトル集合を変換する変換方法であって、

前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定ステップを含む、変換方法。

【請求項 14】

50

前記論理回路はフルスキャン順序回路であって、
前記決定ステップは、

前記テストキューブにおいて、前記フルスキャン順序回路に含まれるスキャンセルの出力に対応するドントケアビットをそれぞれ異なるものと見なし、このような個々のドントケアビット及びその反転が前記フルスキャン順序回路に含まれるスキャンセルの入力への出現状況を調べるステップと、

前記フルスキャン順序回路に含まれるスキャンセルの出力に対応するドントケアビットと、前記フルスキャン順序回路に含まれるスキャンセルの入力への前記ドントケアビット又はその反転の出現状況に基づいて、スキャンキャプチャの前と後で論理値の相違数が最小化することを実現させる論理値を前記ドントケアビットに決定するステップとを含む

10

【請求項 15】

請求項 8 から 14 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 16】

請求項 15 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。

20

【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、論理回路に対するテストベクトル集合を変換する変換装置等に関する。

【背景技術】

【0002】

図 14 に示すように、半導体論理回路は、設計、製造、テストの三段階を経て出荷される。ここで、テストとは、製造された半導体論理回路に対してテストベクトルを印加し、半導体論理回路からテスト応答を観測し、それを期待テスト応答と比較して良品、不良品の判別を行う。その良品率を歩留りと呼び、歩留りは半導体論理回路の品質、信頼性及び製造コストを大きく左右する。

30

【0003】

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と、回路の内部状態を記憶するフリップフロップとよりなる。この場合、組合せ回路部は、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からなる。また、組合せ回路部からの出力は、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からなる。

40

【0004】

順序回路の組合せ回路部をテストするために、組合せ回路部の外部入力線と擬似外部入力線から所要のテストベクトルを印加し、組合せ回路部の外部出力線と擬似外部出力線からテスト応答を観測する必要がある。1つのテストベクトルは、外部入力線と擬似外部入力線に対応するビットからなる。また、1つのテスト応答は、外部出力線と擬似外部出力線に対応するビットからなる。

50

【 0 0 0 5 】

しかし、順序回路のフリップフロップの出力線（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、組合せ回路部からの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。

10

【 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。

20

30

【 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。組合せ回路部からの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（ATPG）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。

40

【 0 0 1 0 】

上述のスキャンテスト方式が有効性を有しているにもかかわらず、通常動作時よりテスト時の消費電力が非常に大きいという問題点が存在する。半導体論理回路がCMOS回路

50

で構成されていれば、消費電力としては、漏れ電流による静的消費電力と、論理ゲートやフリップフロップのスイッチング動作による動的消費電力とがある。さらに、後者の動的消費電力は、シフト操作時におけるシフト消費電力と、キャプチャ操作時におけるキャプチャ消費電力とがある。

【0011】

1つのテストベクトルに対して、スキャンシフト時に与えられるクロックパルス数は一般に多い。例えば、あるスキャンチェーン内のすべてのスキャンフリップフロップに新しい値を設定するために、最大の場合にスキャンフリップフロップ個数分のクロックパルスを与える必要がある。このため、シフト消費電力が大きくなり、過度な発熱を引き起こすことがある。それによって、半導体論理回路装置を損壊する恐れがある。シフト消費電力の低減手法が盛んに研究されている。

10

【0012】

一方、1つのテストベクトルに対して、スキャンキャプチャ時に必要なクロックパルス数は一般に1つのスキャンチェーンにつき1つである。そのため、スキャンキャプチャ消費電力による発熱は問題にならない。しかし、キャプチャモードにおいて、擬似外部出力線に現れる組合せ回路部のテスト応答がスキャンフリップフロップに取り込まれるとき、テスト応答値とスキャンフリップフロップの現在値が異なれば、対応するスキャンフリップフロップの出力値が変化する。このような出力変化スキャンフリップフロップの数が多ければ、論理ゲートとスキャンフリップフロップのスイッチング動作によって、電源電圧が一時的に低下する。この現象はIR（I：電流、R：抵抗）ドロップ現象とも呼ばれる。IRドロップ現象により回路が誤動作し、誤ったテスト応答値がスキャンフリップフロップに取り込まれることがある。これによって、通常時には正常に動作できる半導体論理回路は、テスト時に不良品として判定されてしまうという誤テストが発生する。その結果として、歩留りが低下する。特に、半導体論理回路が超大規模化、超微細化、低電源電圧化した場合、誤テストによる歩留り低下は顕著である。従って、キャプチャ消費電力の低減が必要である。

20

【0013】

テスト時に単一クロック信号を用いる場合には、クロックゲーティング手法を用いてスキャンキャプチャ消費電力を低減することができるが、半導体論理回路の物理設計への影響が大きい。また、テスト時に多重クロック信号を用いる場合には、ワンホット手法もしくは多重クロック手法でスキャンキャプチャ消費電力を低減することができるが、前者はテストデータ量が著しく増大し、後者はテストベクトル生成に膨大なメモリ消費が必要になるなどATPGへの負担が大きい。従って、スキャンキャプチャ消費電力の低減においては、物理設計への影響、テストデータ量の増加、及びATPGへの負担が小さい手法が望ましい。

30

【0014】

他方、ドントケアビットを有するテストキューブはATPGプログラムによるテストベクトルの生成過程で現れることが多い。これに対して、論理ビット（論理値0又は論理値1を持つビット）のみを含み、ドントケアビットを含まないテスト入力をテストベクトルと呼ばれる。また、ドントケアビットを有しないテストベクトルの集合が与えられる場合、その集合の故障検出率を変えずに、一部のテストベクトルの一部ビットをドントケアビットにすることができる。つまり、ドントケアビット特定プログラムによってテストキューブを得ることもできる。テストキューブが存在する原因は、フルスキャン順序回路の組合せ回路部内の1つ又は複数の対象故障を検出するために、外部入力線と擬似外部入力線における一部のビットに必要な論理値を設定すれば十分であることが多いからである。その残りのビットに0を設定しても1を設定しても、その対象故障の検出に影響を与えないため、そのようなビットはその対象故障にとってドントケアビットになる。

40

【0015】

非特許文献1から3はドントケアビットを有しないテストベクトルの集合に対して、その集合の故障検出率を変えずに、一部のテストベクトルの一部ビットをドントケアビット

50

にする技術である。

【0016】

非特許文献1では、テストベクトルごとにドントケアビットを特定するため、Bit-Strippingという各ビットに対して順番にドントケアビットになれるかどうかのチェックを行う手法を使っている。この手法では、テストベクトル間の相関関係を完全に無視している。また、この手法は、ビット数に比例して処理時間が長くなる欠点もある。

【0017】

非特許文献2では、XIDと呼ばれる手法に基づきドントケアビットを識別する。非特許文献1の技術と異なって、XID手法はテストベクトルごとではなく、与えられるテストベクトル集合内の全てのテストベクトルに対して同時に処理する。具体的には、まず各テストベクトルでしか検出できない故障（必須故障と呼ぶ）を求める。次に、全ての必須故障を検出するために必要な論理値設定をATPGの含意操作と論理正当化の手法を応用して求める。その結果、その他の論理ビットをドントケアビットにする。この方法では、全部の入力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ため、実行時間においては前述の非特許文献1で提案された手法よりも効率的で高速である。しかし、このドントケアビット手法は制限条件を受けていない。つまり、この手法では、どの論理ビットもドントケアビットにされる可能性がある。

10

【0018】

非特許文献3では、前述の非特許文献2の技術と同様、テストベクトルごとではなく、与えられるテストベクトル集合内の全てのテストベクトルに対して同時に処理する。前述の非特許文献2の技術との違いは、どの論理ビットもドントケアビットにされることを許さず、一部の論理ビット（候補ビットと呼ぶ）のみからドントケアビットを特定することにある。候補ビット以外の論理ビット（固定ビットと呼ぶ）からはドントケアビットを特定しない。非特許文献3では、候補ビットと固定ビットからなる制約条件のもとでドントケアビットを特定する。前述の非特許文献2と同様で高速である他、所定の目的を達成するための効率的なドントケアビット特定を行うことができる。明らかに、このような目標達成効率は特定されたドントケアビットの位置に関わるため、候補ビットと固定ビットからなる制約条件を目標に合わせて設定することが大事である。

20

【0019】

ドントケアビットを有するテストキューブはあくまでドントケアビットを有しないテストベクトルを生成する過程で現れる中間物である。このため、テストキューブ内のドントケアビットに最終的には論理値0または論理値1を埋め込む必要がある。その埋め込みに際しては、何らかの目的を達成するために必要な論理値（0又は1）をドントケアビットに決定することが一般的に行われる。非特許文献4はスキャンキャプチャ消費電力の低減を目的として、テストキューブのドントケアビットに論理値を決定する技術である。

30

【0020】

非特許文献4では、フルスキャン順序回路の組合せ回路部において、様々な手法で得られたドントケアビットを含むテストキューブに対して、3値（論理値0，論理値1，及びドントケアを表すX）シミュレーションを行い、そのテストキューブに対するテスト応答をまず求める。次に、擬似入力線ビットと擬似出力線ビットからなるビットペアを、擬似入力線ビットにのみドントケアがあるタイプAビットペア、擬似出力線ビットにのみドントケアがあるタイプBビットペア、及び、擬似入力線ビットと擬似出力線ビットの両方にドントケアがあるタイプCビットペアに分類する。更に、これらのビットペアを順番に1つずつ処理していく。その処理において、タイプAビットペアの場合、擬似入力線ビットのドントケアビットに対応する擬似出力線ビットの論理値を割り当てることにし、タイプBビットペアの場合、擬似出力線ビットのドントケアビットに対応する擬似入力線ビットの論理値が現れるように、正当化操作を行って、テストキューブ内のドントケアビットの論理値を決めることにし、タイプCビットペアの場合、擬似入力線ビットと擬似出力線ビットの両方にあるドントケアビットに同じ論理値（0又は1）が現れるように、擬似入力

40

50

線に対する論理値の割り当てと擬似出力線に対する正当化操作を行って、テストキューブ内のドントケアビットの論理値を決めることにする。明らかに、非特許文献4の埋め込み技術の特徴は、テストキューブのドントケアビットに論理値を決める際、1つの擬似入力線ビットと1つの擬似出力線ビットからなる1つのビットペアしか考慮していない。このように決定された論理値は全体的に見て必ずしも最適と言えない。

【0021】

【非特許文献1】R.Sankaralingam and N.A.Touba, "Controlling Peak Power During Scan Testing," Proceedings of IEEE VLSI Test Symposium, pp.153-159,2002.

【非特許文献2】K.Miyase and S.Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," IEEE Transactions on Computer-Aided Design, Vol.23, pp.321-326,2004. 10

【非特許文献3】K.Miyase, S.Kajihara, I.Pomeranz, and S.Reddy, "Don't Care Identification on Specific Bits of Test Patterns," Proceedings of IEEE/ACM International Conference on Computer Design, pp.194-199,2002.

【非特許文献4】X.Wen, H.Yamashita, S.Kajihara, L.-T.Wang, K.Saluja, and K.Kinoshita, "On Low-Capture-Power Test Generation for Scan Testing," Proceedings of IEEE VLSI Test Symposium, pp.265-270,2005.

【発明の開示】

【発明が解決しようとする課題】 20

【0022】

前述のように、フルスキャン順序回路のテストの場合、キャプチャモードにおいて、擬似外部出力線に現れる組合せ回路部のテスト応答がスキャンフリップフロップに取り込まれるとき、テスト応答値とスキャンフリップフロップの現在値が異なれば、対応するスキャンフリップフロップの出力値が変化する。このような出力変化スキャンフリップフロップの数が多ければ、論理ゲートとスキャンフリップフロップのスイッチング動作によって、電源電圧が一時的に低下するIRドロップ現象が起こり、それにより回路が誤動作し、誤ったテスト応答値がスキャンフリップフロップに取り込まれることがある。これによって、通常時には正常に動作できる半導体論理回路は、テスト時に不良品として判定されてしまうという誤テストが発生する。その結果として、歩留りが低下する。特に、半導体論理回路が超大規模化、超微細化、低電源電圧化した場合、誤テストによる歩留り低下は顕著である。従って、キャプチャ消費電力の低減が必要である。 30

【0023】

上記問題が起こらないようにするためには、図15において組合せ回路部102からスキャンフリップフロップ104に取り込まれる論理値と、スキャンフリップフロップ104が現在持っている論理値との間での相違数ができるだけ小さい低キャプチャ時消費電力のテストベクトルをテストに用いることが重要である。このようなテストベクトルを生成するために、何らかの手法でドントケアビットを含むテストキューブを作り、更にその中に含まれるドントケアビットに最適な論理値を決定し埋め込む形で最終テストベクトルに仕上げる必要がある。 40

【0024】

テストキューブはATPGプログラムによるテストベクトルの生成過程で現れることが多い。また、ドントケアビットを有しないテストベクトルの集合が与えられる場合、その集合の故障検出率を変えずに、一部のテストベクトルの一部のビットをドントケアビットにすることができる。特に、与えられたテストベクトルの集合からドントケアビットを特定することによってテストキューブを作るとは、テスト生成フローに与える影響が少ないため、応用において有利である。

【0025】

しかしながら、テストキューブを作るための従来技術は、様々な欠点を持っている。非特許文献1の技術は、実行時間が長く、作られたテストキューブも必ずしもキャプチャ時 50

消費電力を効果的に低減することができない。非特許文献2の技術は、テスト集合全体を処理対象にするため実行時間が比較的早く、より多くのドントケアビットを特定することができるが、作られたテストキューブは必ずしもキャプチャ時消費電力を効果的に低減することができない。非特許文献3の技術は、テスト集合の論理ビットの一部のみよりドントケアビットを特定することによって、キャプチャ時消費電力を効果的に低減することができるテストキューブを作ることが可能であるが、候補ビットと固定ビットで制約されるドントケアビット特定範囲を如何に定めるかについては何も示されていない。

【0026】

更に、キャプチャ時消費電力を効果的に低減するために、テストキューブ内のドントケアビットに論理値0または論理値1を埋め込む技術がある。特に、非特許文献4の技術は、擬似入力線ビットのみならず擬似出力線ビットをも考慮して、対応する擬似入力線ビットと擬似出力線ビットの間の論理値の相違を削減するため、他の技術より有利である。しかし、非特許文献4の技術は、ドントケアビットに論理値(0又は1)を決定するとき、一回に1つの擬似出力線ビットと1つの擬似出力線ビットからなる1つのビットペアしか考慮していない。このため、このように決定された論理値は全体的に見て必ずしも最適と言えない。

10

【0027】

なお、ここまではキャプチャ時の消費電力低減の問題を説明したが、テストデータ量削減、或いはテストデータの欠陥検出においても、制約が必要な同様の問題がある。

20

【0028】

ゆえに、本発明は、上記問題を解決するため、例えば、フルスキャン順序回路に含まれるスキャンセルの出力について、スキャンキャプチャの前後において発生する論理値の相違が低減されるようなテストベクトル集合の変換を行う変換装置、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】**【0029】**

請求項1に係る発明は、論理回路に対するテストベクトル集合を変換する変換装置であって、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビット、及び、ドントケアビットにはならない固定ビットを設定する設定手段と、前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定手段とを備えた変換装置である。

30

【0030】

これにより、ドントケアビットを含まないテストベクトルの集合を最終的なテストベクトル集合とできる。この最終的なテストベクトル集合は、本来のテストの目的(故障検出率)を達成しつつ、他の目的としての例えば誤テストを回避することが可能なものである。

40

【0031】

請求項2に係る発明では、請求項1の決定手段が、前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブに代えて他の手段で得られたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する。

【0032】

請求項3に係る発明は、論理回路に対するテストベクトル集合を変換する変換装置であ

50

って、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビットを設定する設定手段を備えた変換装置である。

【0033】

請求項4に係る発明では、請求項3の設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、前記候補ビットを設定することに加えて、ドントケアビットにしてはならない固定ビットを設定する。

【0034】

請求項5に係る発明では、請求項1から4のいずれかの論理回路がフルスキャン順序回路であり、前記制約条件が、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後の論理値の相違数が削減されるように定められる条件である。

【0035】

これにより、フルスキャン順序回路に含まれるスキャンセルの出力について、スキャンキャプチャの前と後の論理値の相違数を効果的に削減可能となり、スキャンキャプチャ時の消費電力を抑えることで、誤テストを回避できる。

【0036】

請求項6に係る発明は、論理回路に対するテストベクトル集合を変換する変換装置であって、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定手段を備えた変換装置である。

【0037】

請求項7に係る発明では、請求項1、2又は6の論理回路がフルスキャン順序回路であり、前記目的が、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後の論理値の相違数が効果的に削減されることである。

【0038】

請求項8に係る発明は、論理回路に対するテストベクトル集合を変換する変換方法であって、設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビット、及び、ドントケアビットにしてはならない固定ビットを設定する設定ステップと、決定手段が、前記設定手段が設定した候補ビットのみから特定されたドントケアビットを含むテストキューブ、またはその他の手段で得られたドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定ステップとを含む、変換方法である。

【0039】

請求項9に係る発明は、論理回路に対するテストベクトル集合を変換する変換方法であって、設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを特定するようにするために、所定の制約条件であって前記論理回路における入出力関係に基づく条件によりドントケアビットにしてもよい候補ビットを設定する設定ステップを含む、変換方法である。

【0040】

請求項10に係る発明では、請求項9の設定ステップは、前記設定手段が、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちの一部のみからドントケアとできるドントケアビットを特定するようにするために、前記候補ビットを設定することに加えて、ドントケアビットにしてはならない固定ビットを設定するステップを含む。

10

20

30

40

50

【0041】

請求項11に係る発明は、請求項8又は10において、前記論理回路がフルスキャン順序回路であり、前記設定ステップが、前記テストベクトル集合の各テストベクトルに対して、前記フルスキャン順序回路に含まれるスキャンセルの出力において、スキャンキャプチャの前と後で論理値の相違が発生するビットと相違が発生しないビットとを識別する第1のステップと、前記相違が発生しないビットを固定ビットとする第2のステップと、前記相違が発生するビットのうちから所定の条件を満たした前記候補ビットを特定するとともに前記候補ビットに特定されなかった残余の前記相違が発生するビットを固定ビットとする第3のステップとを含み、前記候補ビットからドントケアとできるドントケアビットを特定し、前記固定ビットからドントケアビットを特定しないことを特徴とする変換方法である。

10

【0042】

請求項12に係る発明は、請求項11において、前記第3のステップが、各テストベクトルにおける候補ビットの予定総数を決定するステップと、前記相違が発生するすべてのビットに対してドントケアを割り当て、3値論理シミュレーションを行って、その結果によって、入力ビットにドントケアがありそれに対応する出力ビットにドントケアがないタイプ1のドントケアビットペア、及び、入力ビットとそれに対応する出力ビットの両方にドントケアがあるタイプ2のドントケアビットペアを特定するステップと、前記特定されたタイプ1のドントケアビットペアを優先的に候補ビットとして選択し、タイプ1のドントケアビットペアがなくなりかつ既に選択された候補ビットの数が予定総数に達していない場合、タイプ2のドントケアビットペアを所定の基準に従って候補ビットとして選択するステップを含む、変換方法である。

20

【0043】

この請求項11に係る発明と請求項12に係る発明では、例示的ではあるが、具体的な手法によって、請求項10に係る発明を実現でき、フルスキャン順序回路に含まれるスキャンセルの出力について、スキャンキャプチャの前と後の論理値の相違数が効果的に削減されるという目的を解決できる。

【0044】

請求項13に係る発明は、論理回路に対するテストベクトル集合を変換する変換方法であって、前記テストベクトル集合の各テストベクトルを構成する論理ビットのうちからドントケアとできるドントケアビットを含むテストキューブに対して、入力ビットと出力ビットからなる複数のビットペア間の関係を考慮して、所定の目的を達成するために必要な論理値をドントケアビットに決定する決定ステップを含む、変換方法である。

30

【0045】

請求項14に係る発明は、請求項8又は13において、前記論理回路がフルスキャン順序回路であり、前記決定ステップが、前記テストキューブにおいて、前記フルスキャン順序回路に含まれるスキャンセルの出力に対応するドントケアビットをそれぞれ異なるものと見なし、このような個々のドントケアビット及びその反転が前記フルスキャン順序回路に含まれるスキャンセルの入力への出現状況を調べるステップと、前記フルスキャン順序回路に含まれるスキャンセルの出力に対応するドントケアビットと、前記フルスキャン順序回路に含まれるスキャンセルの入力への前記ドントケアビット又はその反転の出現状況に基づいて、スキャンキャプチャの前と後で論理値の相違数が最小化することを実現させる論理値を前記ドントケアビットに決定するステップとを含む、変換方法である。

40

【0046】

請求項15に係る発明は、請求項8から14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。

【0047】

請求項16に係る発明は、請求項15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。

【発明の効果】

50

【0048】

本発明によれば、例えば、論理回路におけるフルスキャン順序回路に含まれるスキャンセルの出力について、スキャンキャプチャの前後において発生する論理値の相違がより効果的に低減させることができる。それにより、スキャンキャプチャ時の消費電力を抑えることで、誤テストを回避できる。従って、本来正常な動作を行う半導体論理回路をテストにおいて不良品として廃棄するような良品率の低下を解消することができる。

【0049】

また、本発明における変換装置及び変換方法は、論理回路のテスト設計フローを変更することもハードウェア追加によって回路面積を増加させることもない。このため、本発明における変換装置及び変換方法は、キャプチャ時の誤テストを回避するために非常に効果的であると言える。

10

【0050】

さらに、本発明における変換装置及び変換方法は、クロックの種類に依存しないので、テスト時に単一クロック信号を用いる場合にクロックゲーティング手法を採用する場合のようにテストデータ量が著しく増大することはなく、テストデータ量の低減にも効果がある。

【0051】

さらに、本発明における変換装置及び変換方法は、論理回路の故障検出率の低下はないため、テストデータの欠陥検出能力の向上に効果がある。

【発明を実施するための最良の形態】

20

【0052】

本発明の実施の形態を以下に示す。

【0053】

図1を参照して、本発明の背景技術となる一般的なフルスキャン回路について説明する。

【0054】

図1(a)は、一般的なフルスキャン回路の構成を示した模式図である。このフルスキャン回路は、組合せ回路部分300とフルスキャン順序回路のスキャンフリップフロップ302とから構成される。組合せ回路部分300は、外部入力線(P I)、スキャンフリップフロップの出力線である擬似外部入力線(P P I)、外部出力線(P O)、フリップフロップの入力線である擬似外部出力線(P P O)を有する。組合せ回路部分300へのテストベクトル v は、外部入力線より直接与えられる部分 $\langle v : P I \rangle$ と、擬似外部入力線を介して与えられる部分 $\langle v : P P I \rangle$ からなる。 $\langle v : P P I \rangle$ はスキャンシフトによってスキャンフリップフロップ302に設定される。また、組合せ回路部分300からの出力はテストベクトル v に対するテスト応答 $f(v)$ であり、外部出力線に直接現れる部分 $\langle f(v) : P O \rangle$ と、擬似外部出力線に現れる部分 $\langle f(v) : P P O \rangle$ からなる。 $\langle f(v) : P P O \rangle$ はスキャンキャプチャによってスキャンフリップフロップ302に取り込まれる。

30

【0055】

図1(b)は図1(a)におけるスキャンフリップフロップ302におけるスキャンキャプチャの前と後において論理値に相違が発生する場合の一例を示す。

40

【0056】

図1(b)において、テストベクトル $\langle v : P P I \rangle$ の要素である一つのビット a と、それに対応するテスト応答 $\langle f(v) : P P O \rangle$ が、スキャンフリップフロップ302で異なる論理値を取ると、キャプチャモードの際に論理値の相違(以下、遷移とする)が発生する。ある一つのテストベクトルに対する遷移の数は、そのテストベクトルを原因とした組合せ回路部分300を含めた回路全般で発生する消費電力と深く関係しているため、テストベクトルに対するキャプチャ時の遷移の数を削減することで、キャプチャ時の消費電力を低減することができる。

【0057】

50

次に、本発明の実施の形態に係る変換装置の構成を説明する。図2は、本発明の実施の形態に係る変換装置の概略ブロック図である。

【0058】

変換装置400は、設定部402と論理値決定部404とから構成されている。設定部402は、候補ビット特定部406とドントケアビット特定部408とから構成され、論理値決定部404は、ドントケアID化部410と論理値決定処理部412と論理値割当処理部414とから構成されている。また、入出力データとして初期テストベクトル集合416(Tint)と中間テストベクトル集合(テストキューブ集合)418(Tx)と最終テストベクトル集合420(Tfin)があり、初期テストベクトル集合416は予めATPG等の手段により生成されたテストベクトル集合である。設定部402が、初期

10

【0059】

図3は、本発明の実施の形態に係る変換装置の処理を示すフロー図である。

【0060】

まず、ステップ500で図2に示す初期テストベクトル集合416(Tint)をATPG等の手法により生成し用意する。ただし、Tintの生成中に、中間的なテストキューブ中のドントケアビットに対して論理値を決定する必要があるかもしれないが、この段階での論理値の決定は、テストベクトル数削減のための新たな故障の検出を対象としたものである。また、c_limit(各テストベクトルにおける遷移ビット数の上限の値)を同時に設定しておく。

20

【0061】

ここで、c_limitとはスキャンフリップフロップにおけるスキャンキャプチャの前と後において論理値に相違が発生するビット(以下遷移ビットとする)の数の上限値であり、遷移ビットの数がこれ以下であれば、誤テストを起こらないと想定できる数値であり、遷移ビットの数がこれ以上であれば誤テストを起こす要因となり得る数値である。この各テストベクトルにおける遷移ビットの数の上限の値(c_limit)は回路設計時の消費電力量の見積もりまたは経験則等から決定される。c_limitが大きすぎると消費電力を減少することにほとんど有効性を示すことができない。また、c_limitが小さすぎると候補ビットが多数存在するため、効果的なドントケアビットの特定できず、有効性を示すことができない。

30

【0062】

図3に戻って、ステップ500でTintが生成された後は、次のステップ502で、図2に示す候補ビット特定部406により候補ビットの特定が行われ、ステップ503で、図2に示すドントケアビット特定部408によりドントケアビットが特定される。このステップ502及びステップ503では特定したドントケアビットの分布がキャプチャ時の遷移ビット数の削減に対して有効であるドントケアビットが特定される。このステップ502及びステップ503の処理について詳細に説明する。ここでは簡単のため、すべてのスキャンフリップフロップが1本のスキャンチェーンを構成し、また、単一のクロックドメインの中で動作するものとし、シングルキャプチャ方式を仮定する。

40

【0063】

なお、本発明においてはスキャンフリップフロップが複数本のスキャンチェーンで構成されてもよく、また、同様に複数のクロックドメインの中で動作するようにしても構わない。さらに、ダブルキャプチャ方式でも構わない。

【0064】

特定されるドントケアビットの数を最大化するための正当化操作、含意操作を利用して、全てのテストベクトルを同時に処理する手法が知られているが、本発明の実施の形態においては、候補ビットによってドントケアビットの探索を制限し、一方で残りのビットに

50

対しては固定ビットとして論理値を残したままにする。

【0065】

図4は、候補ビットの中からドントケアとできるドントケアビットが特定されたテストベクトル集合の例を示す。

【0066】

図4(a)は、初期テストベクトル集合によりテストが行われる回路の例を示した概念図であり、図4(b)は、回路内に仮定された故障を検出するために生成した初期テストベクトル集合である。図4(c)は、対象となる回路に対するテストベクトル集合の制約表であり、「*」は、図2に示す候補ビット特定部406により特定される候補ビットで、「-」は、値を変更しない固定ビットである。候補ビットの特定についての詳細は後述する。図4(d)は、初期テストベクトル集合に対して候補ビットの中からドントケアとできるドントケアビットの特定を行った結果を示している。図中に示された「X」は、特定された候補ビットの中でドントケアとできるドントケアビットであり、図2に示したドントケアビット特定部408により特定される。

10

【0067】

次に、候補ビットの特定及び候補ビットの中からドントケアとできるドントケアビットを特定する手順を説明する。

【0068】

制約付ドントケアビットの特定で重要とされることは、効果的な制約表を生成することである。スキャンフリップフロップにおけるスキャンキャプチャの前と後において論理値に相違が発生しないビット(以下、非遷移ビットとする)は固定ビットとされるべきであり、一方、スキャンフリップフロップにおけるスキャンキャプチャの前と後において論理値に相違が発生するビット(以下、遷移ビットとする)は、候補ビットとするべきである。また、テストベクトル集合Tに対する候補ビットを特定する際には、特定されたドントケアビットの分布が、キャプチャ時の遷移ビット数の最大値MCT(T)を効果的に削減するように保障されたドントケアビットを特定すべきである。以下に、本発明の実施の形態における制約表の作成手順について説明する。

20

【0069】

(手順1) 入力テストベクトルに対して2値論理シミュレーションを実行し、遷移ビットと非遷移ビットを識別する。

30

【0070】

(手順2) 手順1で識別したビットにおいて非遷移ビットを固定ビットとする。

【0071】

(手順3) 一つのテストベクトルに対して、遷移ビットの集合を選択し、候補ビットを特定する。これを全てのテストベクトルに対して繰り返す。

【0072】

(手順4) 手順3において、候補ビットとして特定されなかった残余の遷移ビットを固定ビットとする。

【0073】

ここで、本発明の実施の形態における制約表について説明する。図5は、制約表の生成のために用いる概念図である。

40

【0074】

図5に示すように、あるテストベクトルに対してスキャンフリップフロップにおける遷移ビット数を n とし、遷移ビット数の上限を c_limit で表すと、候補ビット数 d は、 $d = n - c_limit$ となる。この候補ビット数 d の値は、図4(c)における「*」のビット数と一致している。

【0075】

図5において、入力テストベクトル v_i では遷移ビット数 n が遷移ビット数の上限 c_limit を上回っているため、遷移ビット数 n から遷移ビット数の上限 c_limit を減算した値 d 個を候補ビット数とすることができる。また、 v_j では遷移ビット数 n が

50

遷移ビット数の上限 c_limit を下回っているため、候補ビットは存在しない。

【0076】

従って、候補ビット数 d の値は c_limit の決め方に依存するため、本発明を効果的に実現するためには、適切な c_limit の設定が必要不可欠である。

【0077】

図5に示す概念により、ある一つのテストベクトルに対する候補ビットが多数存在することによる非効率的なドントケアビットの特定、または、行う必要のないドントケアビットの特定を避けることができる。従って、その他のテストベクトルにおいて新たに効果的なドントケアビットの特定が可能となる。その結果、テストベクトル集合全体に対して、キャプチャ時の遷移ビット数の削減を達成できる。

10

【0078】

図5において候補ビットの数が算出された後は、何れのビットを候補ビットとして特定するかを識別する必要がある。

【0079】

図6は、候補ビットの特定方法を示した図である。

【0080】

図6における候補ビットの特定方法は、後の処理であるドントケアビットに論理値を割り当てる処理を容易にするための候補ビットの特定方法である。入力テストベクトル v に対して2値論理シミュレーションを行った結果を $f(v)$ とする。 $f(v)$ には遷移ビットが2個含まれており、それぞれの遷移ビットを a 及び b とする。ここで、 c_limit の値を1と設定すると、特定する候補ビット数は、遷移ビット数の2から c_limit の1を差し引いた1である。

20

【0081】

次に、 v の遷移ビットをすべて X に置き換えたものを v' とする。この v' に対して0、1及び X の3値を使用した3値論理シミュレーションを行い、その結果を $f(v')$ とすると、 a 及び b に対応するそれぞれのドントケアビットペアは2個存在する。ここで、それぞれのドントケアビットペアについて、図7を参照して説明する。

【0082】

図7は、ドントケアビットを含むビットペアをタイプ毎に分類して表にしたものである。

30

【0083】

図7におけるタイプAは、入力値がドントケアビットであるが、それに応答する出力値は固定値であるタイプである。このタイプAは出力値が決まっているため、容易にドントケアビットに対して論理値を割り当てることできる。タイプBは、入力値が固定値であるが、それに応答する出力値がドントケアビットであるタイプである。このタイプは出力値がドントケアビットであるため、出力側のドントケアビットに対する論理値の割り当ての際に出力値に影響を及ぼしている入力値を識別し、入力側のドントケアビットに論理値を割り当てる必要がある。このタイプBは、タイプAに比べて容易にドントケアビットに対して論理値を割り当てることできない。タイプCは、入力値及び出力値が共にドントケアビットであり、タイプA及びタイプB双方の処理を行わなければならないため複雑な処理が必要となる。

40

【0084】

図6における a に対応するドントケアビットペアは、図7のタイプCに該当し、図6における b に対応するドントケアビットペアは、図7のタイプAに該当する。図7によると、タイプAに該当するドントケアビットペアに対して行うドントケアビットに対する論理値の割り当ては、タイプCに該当するドントケアビットペアに対して行う論理値の割り当てよりも難易度が低く容易である。従って、図6における b のタイプのドントケアビットペアを特定することが望ましい。

【0085】

以下に、候補ビットの特定手順を示す。

50

【0086】

(手順1) 入力テストベクトルに対して事前に2値論理シミュレーションを実行し、遷移ビットと非遷移ビットを識別する。

【0087】

(手順2) 手順1にて識別された遷移ビットをドントケアビットに置換し、3値論理シミュレーションを行い、ドントケアビットペアを特定する。

【0088】

(手順3) 候補ビット数を越えないように、最初にタイプAのドントケアビットペアに該当する遷移ビットを候補ビットとして選択する。候補ビット数を満たさない場合は、タイプCのドントケアビットペアに該当する遷移ビットを候補ビットとして選択する。

10

【0089】

上記手順3において、複数のタイプCが存在する場合、出力側のドントケアビットから到達可能な入力側のドントケアビット数が多いドントケアビットペアを優先して選択する。到達可能な入力側のドントケアビット数が多いほど、出力側のドントケアビットに論理値を割り当て易くなるからである。

【0090】

以上のような手順で処理を行うことにより、候補ビットの特定及び候補ビットの中からドントケアとできるドントケアビットが特定される。

【0091】

図3の処理フローに戻って、ここまでで特定されたドントケアビットがステップ504に示す中間テストベクトル集合(Tx)である。

20

【0092】

次に、ステップ506で図2における論理値決定部404が、中間テストベクトル集合418(Tx)である特定したドントケアビットに対して、キャプチャ時に遷移ビット数が削減されるような適切な論理値の割り当てを行う。その手法を図8に示す。この手法は、従来手法である、キャプチャ時に遷移ビット数が削減されるように割当操作、正当化操作及びそれらの組合せを行う点で類似しているが、本発明の実施の形態においては、論理値の決定方法という点が改良されている。

【0093】

図8は、特定したドントケアビットに対して論理値の割り当てを行う場合の処理を示したフロー図である。

30

【0094】

まず、図2における中間テストベクトル集合418において、全てのドントケアビットを含むテストベクトルに対して処理が行われたか否かを判定する(ステップ1000、1002)。全てのドントケアビットを含むテストベクトルに対して行われていれば、ステップ1018で最終テストベクトル集合Tfinが得られ、処理を終了する。全てのドントケアビットを含むテストベクトルに対して行われていなければステップ1004に進む。ステップ1004ではドントケアビットを含むテストベクトルを1つ選択し、ステップ1006でステップ1004で選択されたテストベクトル内の全てのドントケアビットペアの相互間の依存関係を調べる。このドントケアビットペアの相互間の依存関係を調べる処理は、図2に示すドントケアビットID化部410により行われる。

40

【0095】

ステップ1008で、全てのドントケアビットペアに対して相互間の依存関係を調べたか否かを判定し、全てのドントケアビットペアに対して相互間の依存関係を調べたらステップ1002に戻る。全てのドントケアビットペアに対して相互間の依存関係を調べていなければステップ1010に進む。ステップ1010では、ドントケアビットペアを1つ選択し、ステップ1012で、選択したドントケアビットペアのドントケアビットに割り当てる論理値を決定する。このステップ1012の処理は、図2に示す論理値決定処理部412により行われる。次に、ステップ1012で決定した論理値を、ステップ1014で正当化することによりドントケアビットに割り当てを行い、ステップ1016で割り当

50

てが成功するか否かが判定される。成功しなければステップ1012に戻り新たな論理値を決定する。成功すればステップ1008に戻り、全てのドントケアビットペアに対して処理が行われるまで繰り返して処理が行われる。

【0096】

なお、ステップ1014における正当化処理は行われる必要はない場合もある。

【0097】

以下に、図8におけるステップ1006(A)のドントケアビットを含むテストベクトル内の全てのドントケアビットペアの相互間の依存関係を部分的記号Xシミュレーションにより求める処理についての手順を説明する。ドントケアビットペアの相互間の依存関係とは、出力側のドントケアビットが何れの入力側のドントケアビットから伝搬したかの関係をいい、依存関係を調べることにより適切な論理値の決定を行うことができる。また、部分的記号Xシミュレーションとは、従来故障診断に用いられてきた記号シミュレーションの一部である。

10

【0098】

ここで、全ての論理値が0または1である初期テストベクトルから特定されたm個のドントケアビットを含むテストベクトルを v' とする。 v' に対する部分的記号Xシミュレーションは以下のように処理される。まず、 v' 中のm個のドントケアビットを相異なるm個のX記号(X_1, X_2, \dots, X_m)で置き換えたものを v'' とする。その結果 v'' はX記号を含み、X記号の伝搬処理は次の2つの規則を基に繰り返し行われる。

【0099】

20

(規則1) 論理演算の結果が X_i の否定となる場合は、 X_i とする(は否定論理を表す)。

【0100】

(規則2) もし、論理演算の結果が確定値でなく、且つ、X記号 X_i または X_i でもなければ、新しいX記号が生成される。

【0101】

これらの規則は、X記号の伝搬処理において論理否定のみを保存し、その他の論理演算に関する情報は無視する。その目的は、出来る限り有効な情報を保ちながらメモリの使用量を削減するためである。

【0102】

30

図9に、上記部分的記号Xシミュレーションを実行した結果の一例を示す。

【0103】

図9において、入力テストベクトル v' には3個のドントケアビットが存在する。上記規則に従い v' における3個のドントケアビットに新しく記号 X_1, X_2 及び X_3 を割り振った入力テストベクトルが v'' である。X記号の伝搬処理は規則1及び規則2に従って繰り返し行われる。例えば、ゲートG2の出力は X_2 の否定になるため、 X_2 と表される。また、ゲートG4の出力は X_1 と X_2 の論理和であるため、新しく記号 X_4 と表される。

【0104】

一般的に、部分的記号Xシミュレーションは図9の例が示すように3値論理シミュレーションよりも依存関係の情報を求めることができる。ここでは3値論理シミュレーションの結果においては、信号線 $\langle e, z \rangle$ 上のドントケアビットペアは $\langle X, X \rangle$ となり、依存関係が得られない。しかし、部分的記号Xシミュレーションを行った結果、信号線 $\langle e, z \rangle$ 上のドントケアビットペアは $\langle X_3, X_3 \rangle$ となり、信号線 $\langle e, z \rangle$ は同じ論理値を取り得ないことを示している。この情報は3値論理シミュレーションでは得ることができない。

40

【0105】

次に、図8におけるステップ1012(B)及びステップ1014のドントケアビットペアのドントケアビットに割り当てる論理値を決定し、割り当てる処理について説明する。ここで、決定される論理値は、上記で調べた依存関係に基づいて、遷移ビットの数を削

50

減するようにドントケアビットに設定される値である。適した論理値が決定され、割り当てられることができれば、誤テストの原因となる遷移ビット数が削減され、誤テストを回避でき得る。

【0106】

ドントケアビットを含むテストベクトル v' に対して論理値の割り当てが必要であり、 v' と $f(v')$ のドントケアビットに対して割り当てるべき論理値を決定する必要がある場合を仮定する。以下にその決定の手順を示す。

【0107】

(手順1) ドントケアビットを含む入力テストベクトル v'' に対応する X 記号を含む v'' に対して部分的記号 X シミュレーションを行う。全ての X 記号のビットペアは $\langle v'' : PPI \rangle$ と $\langle f(v'') : PPO \rangle$ を比較することにより求めることができる。ただし $f(v'')$ は v'' の出力側のテストベクトルである。 10

【0108】

(手順2) ビットペア $\langle X_i, b_2 \rangle$ 、 $\langle b_1, X_i \rangle$ 及び $\langle b_1, X_i \rangle$ のそれぞれの X 記号のビットペアに対して図 10 に示される規則に従い X_i に適合すると思われる論理値を算出する。ここで、 b_1 及び b_2 は論理値である。ただし、 X_i に対する 0 を優先的に割り当てる指標を示す値及び 1 を優先的に割り当てる指標を示す値は 0 とする。

【0109】

(手順3 - a) ビットペア $\langle X_i, b_2 \rangle$ 、 $\langle b_1, X_i \rangle$ 及び $\langle b_1, X_i \rangle$ の X_i に対して論理値を決定する場合、もし、0 (1) を優先的に割り当てる指標を示す値が 1 (0) を優先的に割り当てる指標を示す値より大きければ 0 (1) に決定する。 20

【0110】

(手順3 - b) ビットペア $\langle a, b \rangle = \langle X_i, X_i \rangle$ の場合、0 を優先して決定し、その決定による論理値の割り当てが失敗であれば 1 を割り当てる。

【0111】

(手順3 - c) ビットペア $\langle a, b \rangle = \langle X_i, X_i \rangle$ の場合、 a に 0 を決定し、 b に 1 を決定する。その決定による論理値の割り当てが失敗であれば a に 1 を、 b に 0 を割り当てる。

【0112】

(手順3 - d) ビットペア $\langle a, b \rangle = \langle X_i, X_j \rangle$ または $\langle X_i, X_j \rangle$ の場合、 a 及び b に同じ論理値を決定し、その決定による論理値の割り当てが失敗であれば a 及び b に異なる論理値を割り当てる。 30

【0113】

手順1は、部分的記号 X シミュレーションにより X 記号のビットペアの依存関係の情報を求める処理である。手順2は、ドントケアビットを決定するために X 記号のビットペアの依存関係を考慮する。手順3は、4つの異なる論理値の決定についての処理である。これらの処理の結果から、効果的にドントケアビットに論理値を決定することができる。この手順を行った場合の例を、図 11 を参照して説明する。

【0114】

図 11 は、ドントケアビットに論理値を決定する場合の例を示した図である。 40

【0115】

図 11 において、3つの X 記号のビットペア $\langle a, x \rangle = \langle X_1, 0 \rangle$ 、 $\langle b, y \rangle = \langle 1, X_1 \rangle$ 及び $\langle c, z \rangle = \langle 1, X_1 \rangle$ がある。図 10 の表に従うと X_1 に対する 0 を優先的に割り当てる指標を示す値は 1 となり、1 を優先的に割り当てる指標を示す値は 2 となる。従って、X 記号のビットペア $\langle a, x \rangle = \langle X_1, 0 \rangle$ の X_1 に対する論理値を決定する場合、論理値は 1 に決定される。これは X 記号のビットペア $\langle a, x \rangle$ では、キャプチャ時に遷移が発生するが、 $\langle b, y \rangle$ 及び $\langle c, z \rangle$ では、キャプチャ時に遷移が発生しない。

【0116】

従来の手法においては、ドントケアビットペア間の相互の関係を考慮せずに $\langle a, x \rangle$ 50

の関係のみから X 1 の論理値を決定していたため、X 1 には 0 が決定され遷移ビットの数が削減される効果が十分とは言えなかった。しかし、本発明の実施の形態に係るドントケアビットへの論理値の決定手法を行うことで、ドントケアビットペア間の相互の関係が考慮されるため、遷移ビットの数が効果的に削減され、誤テストを回避することができる。

【0117】

上記手順で処理を行うことにより、図3におけるステップ508の最終テストベクトル集合 (T f i n) が得られ、処理を終了する。

【0118】

以下に、本発明の実験結果を示す。

【0119】

本実験では、本発明の提案手法を C 言語により実装し、I S C A S ' 8 9 ベンチマーク回路を対象とし、C P U 1 . 0 G H z のペンティアム (登録商標) III プロセッサ、メモリ 5 1 2 M B の計算機により行われた。図12は実験結果を示した表である。

【0120】

図12 (a) に、実験に使用した回路の基本情報を示す。

【0121】

外部入力数、外部出力数及び F / F s (フリップフロップ) の数がそれぞれ「# o f P I s」、 「# o f P O s」及び「# o f F / F s」に示される。A T P G プログラムにより縮退故障用テストベクトル集合を生成し、テストベクトル数及び各回路の故障検出率はそれぞれ「# o f V e c t o r」及び「F a u l t C o v e r a g e」に示される。さらに、各回路のオリジナルの遷移ビットの最大数 (初期テストベクトル集合における遷移ビットの最大数) が「O r i g e n a l M a x T r a n s」に示される。

【0122】

図12 (b) は、従来手法である、候補ビットの特定を行わないドントケアビットの特定手法に基づくキャプチャ時の消費電力低減手法による実験結果を示した表である。

【0123】

各回路において、候補ビットの特定を行わずに特定されたドントケアビットの割合及び CPU 時間を「U n c o n s t r a i n e d X - I d e n t i f i c a t i o n」の「X (%)」及び「C P U (S e c .)」に示す。最下行には平均値を示しており、全て 0 または 1 であるテストベクトル集合中の平均 6 3 . 5 % が故障検出率を低下させずにドントケアビットとして特定されている。

【0124】

従来手法 (L C P (L o w C a p t u r e P o w e r)) では、特定されたドントケアビットに対する論理値の割り当ては、ドントケアビットペアの相互間の依存関係を考慮しない。各回路の遷移ビットの最大数、遷移ビットの減少比率及び CPU 時間を「O l d L C P X - F i l l i n g」の「M a x T r a n s .」、 「R e d R a t e (%)」及び「C P U (S e c .)」に示す。また、比較のためにドントケアビットにランダムに論理値を割り当てた場合の結果を「R a n d o m X - F i l l i n g」の「M a x T r a n s .」に示す。図12 (b) の表より、明らかに、従来手法によるドントケアビットへのランダムな論理値の割り当てよりキャプチャ時の消費電力の低減効果があることがわかる。

【0125】

図12 (c) は、本発明の実施の形態における実験結果を示した表である。

【0126】

本発明の実施の形態では、候補ビットの特定及び候補ビットからドントケアとできるドントケアビットの特定は制約表を生成して行われ、ドントケアビットへの論理値の割り当てはドントケアビットペアの相互間の依存関係を考慮してキャプチャ時の消費電力を低減する手法により行われた。図12 (c) の表に示す各列は図12 (b) の表に示す各列と同意義である。

10

20

30

40

50

【0127】

図12(b)より、従来技術によると最大遷移ビット数の減少比率は平均して33.2%である。また、図12(c)より、本発明の実施の形態における手法によると最大遷移ビット数の減少比率は平均して43.3%である。このことから従来技術に比べて本発明の実施の形態における手法が遷移ビット数の減少比率を30.4%改善していることになる。

【0128】

遷移ビットの数は、回路中のノード遷移の数と深い相互関係があり、そのノード遷移の数はキャプチャ時の消費電力の浪費を直接反映するものであるため、本発明において提案された手法がキャプチャ時の消費電力の浪費を減少させることに効果的である。

10

【0129】

本発明の実施の形態の手法においては、遷移ビット数の上限である c_limit を設定する必要がある。本実験においては、初期テストベクトル集合 T_{int} の遷移ビット数の最大値の10%を c_limit に設定して行った。

【0130】

図12(b)における「Unconstrained X-Identification」の平均値は65.3%を示しており、図12(c)における「Unconstrained X-Identification」の平均値は21.9%を示している。これは、本発明の実施の形態における手法が、制約を受けるドントケアビットの数が小さいにも関わらず、キャプチャ時の消費電力の浪費を減少させる上で明らかに有効であることを示している。従って、ドントケアビットの数だけでなく、ドントケアビットの分布が、キャプチャ時の消費電力の浪費を減少させる効果を向上させる重要な役割を果たすことがわかる。

20

【0131】

なお、上記まででは、図1に示すようなフルスキャン順序回路を用いて説明したが、図13に示すようなパーシャルスキャン回路にも適用可能である。ここで、パーシャルスキャン回路とは、一部の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本若しくは複数本のスキャンチェーンを形成するものである。

【0132】

具体的には、組合せ回路部分3000と順序回路のフリップフロップ3001とパーシャルスキャン順序回路のスキャンフリップフロップ3002とから構成される。組合せ回路部分3000は図1の組合せ回路部分300と同様ではあるが、その入力は $\langle v : PPI \rangle$ がスキャンシフトによってスキャンフリップフロップ3002に設定される部分 $\langle v : PPI_{n2} \rangle$ とフリップフロップ3001に設定される部分 $\langle v : PPI_{n1} \rangle$ からなる。また、組合せ回路部分3000からの出力は同じく図1の組合せ回路部分300と同様であるが、テストベクトル v に対するテスト応答 $f(v)$ としての外部出力線に直接現れる部分 $\langle f(v) : PO \rangle$ と擬似外部出力線に現れる部分 $\langle f(v) : PPO \rangle$ のうち、 $\langle f(v) : PPO \rangle$ はスキャンキャプチャによってスキャンフリップフロップ3002に取り込まれる部分 $\langle f(v) : PPO_{n2} \rangle$ とフリップフロップ3001に取り込まれる部分 $\langle f(v) : PPO_{n1} \rangle$ からなる。

30

40

【0133】

図13(b)は図13(a)におけるスキャンフリップフロップ3002におけるスキャンキャプチャの前と後において論理値に相違が発生する場合の一例を示す。

【0134】

図1(b)の説明と同様であるが、図13(b)において、テストベクトル $\langle v : PPI_{n2} \rangle$ の要素である一つのビット a と、それに対応するテスト応答 $\langle f(v) : PPO_{n2} \rangle$ が、スキャンフリップフロップ3002で異なる論理値を取ると、キャプチャモードの際に論理値の相違(以下、遷移とする)が発生する。ある一つのテストベクトルに対する遷移の数は、そのテストベクトルを原因とした組合せ回路部分3000を含めた回路全般で発生する消費電力と深く関係しているため、テストベクトルに対するキャプチャ時

50

の遷移の数を削減することで、キャプチャ時の消費電力を低減することができる。

【図面の簡単な説明】

【0135】

【図1】本発明の背景技術となる一般的なフルスキャン回路の構成を示した模式図である。

【図2】本発明の実施の形態に係る変換装置の概略ブロック図である。

【図3】本発明の実施の形態に係る変換装置の処理を示すフロー図である。

【図4】候補ビットの中からドントケアとできるドントケアビットが特定されたテストベクトル集合の例を示した図である。

【図5】制約表の生成のために用いる概念図である。

10

【図6】候補ビットの特定方法を示した図である。

【図7】ドントケアビットを含むビットペアをタイプ毎に分類した表である。

【図8】特定したドントケアビットに対して論理値の割り当てを行う場合の処理を示したフロー図である。

【図9】部分的記号Xシミュレーションを実行した結果の一例を示す。

【図10】論理値を決定する規則を示した図である。

【図11】ドントケアビットに論理値を決定する場合の例を示した図である。

【図12】本発明における実験結果を示した表である。

【図13】一般的なパーシャルスキャン回路の模式図を図nに示す。

【図14】半導体論理回路が市場に出荷されるまでの工程を示した模式図である。

20

【図15】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。

【符号の説明】

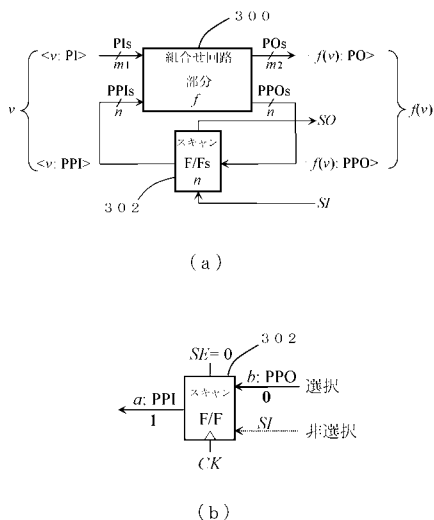
【0136】

400 変換装置

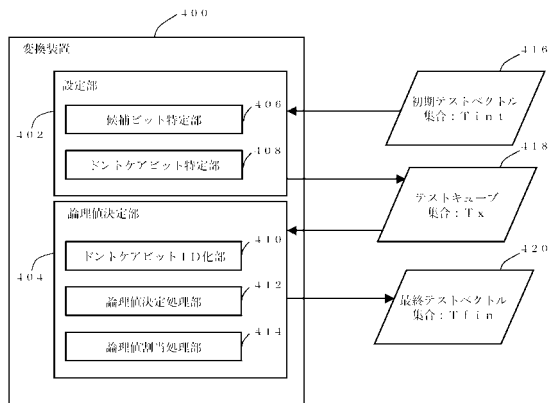
402 設定部

404 論理値決定部

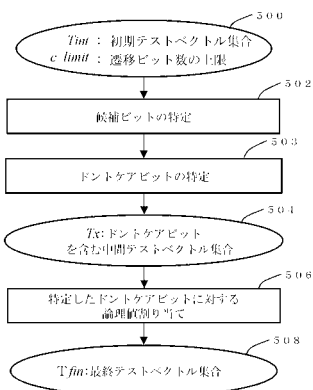
【図1】



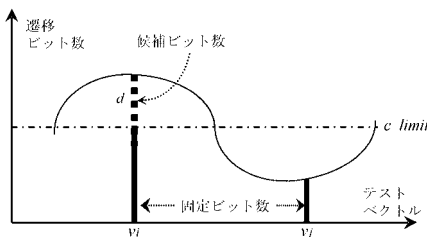
【図2】



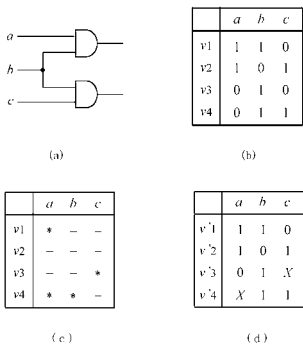
【図3】



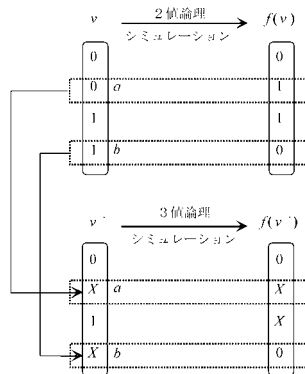
【図5】



【図4】



【図6】

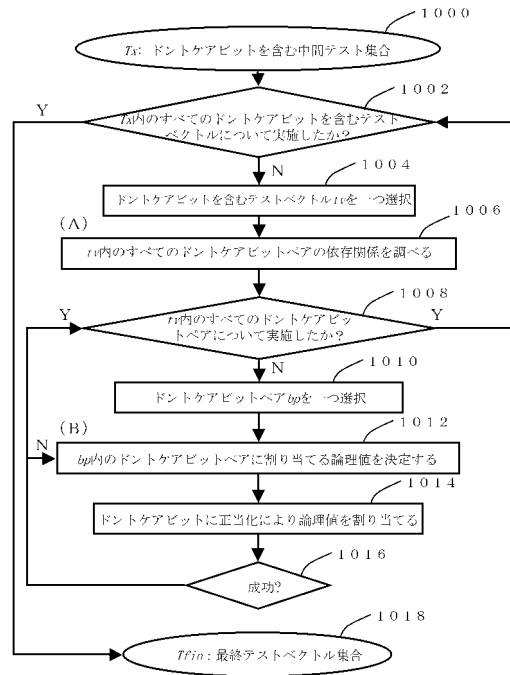


【 図 7 】

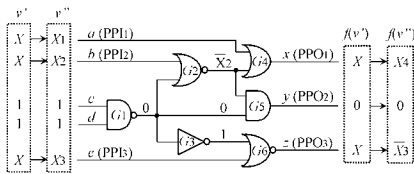
	$v : \text{PPI} \rangle$	$f(v) : \text{PPO} \rangle$	ドントケアビットに対する論理軌跡	
			X_i に相当する論理値	難易度
タイプ-A	X	b_2	b_2	低
タイプ-B	b_1	X	b_1	中
タイプ-C	X	X	0-0 1-1	高

($b_1, b_2 : 0 \text{ or } 1$)

【 図 8 】



【 図 9 】



【 図 12 】

Circuit	-of PIs	-of FFS	-of FFS	-of FFS	Final Success	Original Min. Trms.
s1258	14	14	18	175	94.9	18
s1423	17	5	74	24	99.1	49
s378	35	49	479	100	99.1	102
s12017	31	121	669	255	98.5	380
s15850	14	87	597	97	96.7	282
s3932	35	329	1728	12	89.8	1548
s38417	28	106	1636	87	99.5	590
s38384	12	278	1452	114	95.9	925

(a)

【 図 10 】

$v'' : \text{PPI} \rangle$	$f(v'') : \text{PPO} \rangle$	X_i	
		0-割り当て指標	1-割り当て指標
X_i	b_2	+1 if $b_2 = 0$	+1 if $b_2 = 1$
b_1	X_i	+1 if $b_1 = 0$	+1 if $b_1 = 1$
b_1	\bar{X}_i	+1 if $b_1 = 1$	+1 if $b_1 = 0$

($b_1, b_2 : 0 \text{ or } 1$)

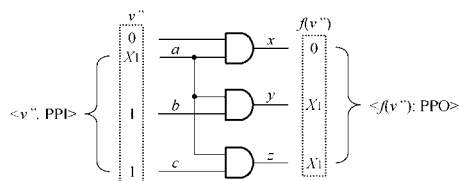
Circuit	Unconstrained X-identification		Random X-filling		Old ICP X-filling	
	X (%)	CPU (Sec.)	Max. Trms.	Max. Fms.	Min. Fms. (%)	CPU (Sec.)
s1258	55.0	0.1	1.1	9	50.0	0.0
s1423	41.1	0.1	4.0	3.1	20.6	0.0
s378	71.0	3.3	109	91	19.8	0.2
s12017	91.6	8.6	395	244	35.8	5.9
s15850	76.1	5.3	262	173	38.7	1.8
s3932	54.4	4.0	1532	1517	2.0	0.3
s38417	73.4	13.8	600	323	45.3	24.7
s38384	79.7	17.5	802	437	52.8	22.3
avr	65.3	-	-	-	33.2	-

(b)

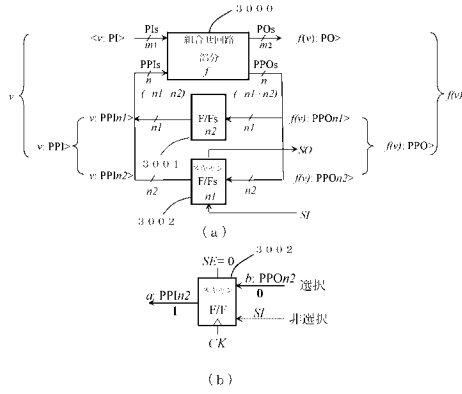
Circuit	Constrained X-identification		Random X-filling		New ICP X-filling	
	X (%)	CPU (Sec.)	Max. Trms.	Min. Fms.	Min. Fms. (%)	CPU (Sec.)
s1258	32.6	0.1	14	5	72.2	0.0
s1423	19.8	0.1	43	31	30.6	0.0
s378	25.2	1.2	108	70	31.4	0.5
s12017	33.1	10.3	333	198	47.9	5.5
s15850	20.2	5.2	252	132	53.2	1.8
s3932	12.8	3.7	1533	1482	4.3	0.6
s38417	20.3	15.3	592	295	50.0	5.6
s38384	20.6	14.9	785	390	57.2	7.0
avr	21.9	-	-	-	43.3	-

(c)

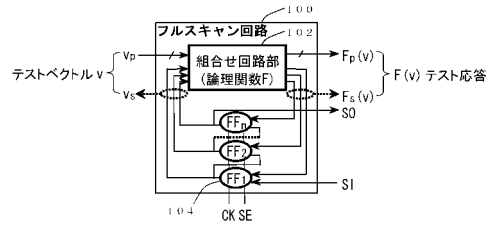
【 図 11 】



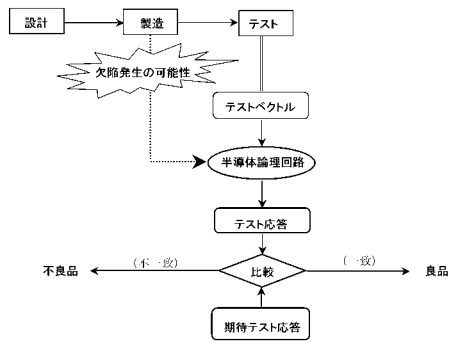
【 図 1 3 】



【 図 1 5 】



【 図 1 4 】



フロントページの続き

- (72)発明者 梶原 誠司
福岡県飯塚市大字川津 6 8 0 - 4 九州工業大学内
- (72)発明者 宮瀬 紘平
福岡県福岡市早良区昭代 1 - 6 - 2 7 4 0 2 号
- (72)発明者 皆本 義弘
福岡県北九州市八幡西区三ヶ森 2 - 8 - 1 8 4 0 3 号
- (72)発明者 伊達 博
福岡県福岡市早良区百道浜 3 - 8 - 3 3 株式会社システム・ジェイディー内
- Fターム(参考) 2G132 AA01 AB01 AC03 AC14 AD06 AG11 AG12 AL11 AL40