

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3825572号
(P3825572)

(45) 発行日 平成18年9月27日(2006.9.27)

(24) 登録日 平成18年7月7日(2006.7.7)

(51) Int. Cl. F I
G06F 17/50 (2006.01) G O 6 F 17/50 6 7 4
HO1L 21/82 (2006.01) G O 6 F 17/50 6 6 4 G
 HO 1 L 21/82 C

請求項の数 14 (全 16 頁)

(21) 出願番号	特願平11-9373	(73) 特許権者	000003078
(22) 出願日	平成11年1月18日(1999.1.18)		株式会社東芝
(65) 公開番号	特開2000-207440(P2000-207440A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成12年7月28日(2000.7.28)	(74) 代理人	100058479
審査請求日	平成15年1月16日(2003.1.16)		弁理士 鈴江 武彦
		(74) 代理人	100084618
			弁理士 村松 貞男
		(74) 代理人	100068814
			弁理士 坪井 淳
		(74) 代理人	100092196
			弁理士 橋本 良郎
		(74) 代理人	100091351
			弁理士 河野 哲
		(74) 代理人	100088683
			弁理士 中村 誠

最終頁に続く

(54) 【発明の名称】 半導体集積回路の設計検証装置、方法及び記憶媒体

(57) 【特許請求の範囲】

【請求項1】

変更前後の回路記述を入力する入力手段と、
前記入力手段からの前記変更前後の回路記述におけるレジスタの出力及び信号の入出力ノードの配置関係を表すキーポイントの不一致箇所に応じて変更箇所を特定する特定手段を有し、

前記特定手段による特定の結果、前記キーポイントの一致を保証できる回路記述に対して形式検証を行い、前記特定手段による前記特定の結果、前記変更前後の回路記述において前記キーポイントの一致を保証できない回路記述、もしくは前記形式検証により論理的不一致が発生した場合、論理的な変更箇所を含む回路記述に対してはイベントドリブン方式のシミュレーションを利用して回路の機能検証を行う検証機構を備えることを特徴とした半導体集積回路の設計検証装置。

【請求項2】

変更前後の回路記述を入力する入力手段と、
入力された前記回路記述中においてレジスタの出力および信号の入出力ノードの配置関係を表すキーポイントのマッピングを行い、前記変更前と変更後の回路記述の不一致箇所から、最も近くの前記変更前と変更後の回路記述の一致しているキーポイントを外部入出力とし、変更箇所を含む回路部をモジュール化する変更箇所特定手段と、

前記モジュール化した前記変更前と変更後の回路記述中の回路部それぞれに対しイベントドリブン方式のシミュレーションを利用した回路の機能検証を行うシミュレーション検

証手段と、

前記モジュール化した各回路部において外部入出力扱いとしたノードのみと前記変更前と変更後の回路記述それぞれに対して形式検証を利用し回路の機能を検証する形式検証手段とを具備し、

前記形式検証手段において、論理的不一致が発生した場合、前記変更前と変更後の回路記述に対し論理的な変更箇所があるとして前記形式検証に伴うコーンのうち前記論理的な変更箇所および前記外部入出力扱いとしたノードを含んだコーンの入出力を外部入出力扱いとして前記論理的な変更箇所を含む回路記述を切り出しモジュール化する機能を有し、このモジュール化した前記変更前と変更後の回路記述中の回路部それぞれの機能が前記シミュレーション検証手段で検証されることを特徴とする半導体集積回路の設計検証装置。

10

【請求項 3】

前記イベントドリブン方式のシミュレーションを利用するため、前記モジュール化された各回路部の入力キーポイントに繰り返し入力されるシミュレーション時間の区切られたランダムなテストベクトルを生成するためのテストベンチ生成手段を含むことを特徴とする請求項 2 に記載の半導体集積回路の設計検証装置。

【請求項 4】

前記シミュレーション検証手段における前記変更後の回路記述に関する期待値を入力できる機能を備え、

前記シミュレーション検証手段からの実際のシミュレーション結果と前記期待値とを比較する比較判定手段を具備することを特徴とする請求項 1 乃至 3 いずれか一つに記載の半導体集積回路の設計検証装置。

20

【請求項 5】

前記比較判定手段は、前記シミュレーション検証手段からの実際のシミュレーション結果により、前記変更前と変更後の回路記述に不一致の発生したテストベクトルとこのテストベクトルを用いた前記変更後の回路記述に応じた出力を前記期待値と比較することにより、期待した変更がなされているか否か判定を行うことを特徴とする請求項 4 に記載の半導体集積回路の設計検証装置。

【請求項 6】

前記期待値は、RT レベルの回路記述段階における変更箇所を指定することでその回路記述を解析し、前記変更前と変更後の回路記述で異なる実行結果が得られるテストベクトル（入力信号）とその実行結果から作成することを特徴とする請求項 4 に記載の半導体集積回路の設計検証装置。

30

【請求項 7】

前記不一致が発生したテストベクトル（入力信号）についてこのテストベクトルを用いた前記変更後の回路記述に関する回路部の出力を新たな期待値として登録できる機能をさらに具備し、前記イベントドリブン方式のシミュレーションの続行を可能とすることを特徴とする請求項 5 に記載の半導体集積回路の設計検証装置。

【請求項 8】

入力手段により変更前後の回路記述を入力し、

特定手段により、前記入力手段により入力された変更前後の回路記述におけるレジスタの出力及び信号の入出力ノードの配置関係を表すキーポイントの不一致箇所に応じて変更箇所を特定し、

40

検証機構により、前記特定手段による特定の結果、前記キーポイントの一致を保証できる回路記述に対して形式検証を行い、前記特定手段による前記特定の結果、前記変更前後の回路記述において前記キーポイントの一致を保証できない回路記述、もしくは前記形式検証により論理的不一致が発生した場合、論理的な変更箇所を含む回路記述に対してはイベントドリブン方式のシミュレーションを利用して回路の機能検証を行うことを特徴とする半導体集積回路の設計検証方法。

【請求項 9】

入力手段により変更前と変更後の回路記述を入力する過程と、

50

変更特定手段により、前記入力手段により入力された前記回路記述中においてレジスタの出力および信号の入出力ノードの配置関係を表すキーポイントのマッピングを行い、前記変更前と変更後の回路記述の不一致箇所から、最も近くの前記変更前と変更後の回路記述の一致しているキーポイントを外部入出力とし、変更箇所を含む回路部をモジュール化する変更箇所特定過程と、

シミュレーション検証手段により、前記モジュール化した前記変更前と変更後の回路記述中の回路部それぞれに対しイベントドリブン方式のシミュレーションを利用した回路の機能検証を行うシミュレーション検証過程と、

形式検証手段により、前記モジュール化した各回路部において外部入出力扱いとしたノードのみと前記変更前と変更後の回路記述それぞれに対して形式検証を利用し回路の機能を検証する形式検証過程とを具備し、

10

前記形式検証過程で論理的不一致が発生した場合、論理の変更箇所特定手段により、前記変更前と変更後の回路記述に対し論理的な変更箇所があるとして前記形式検証に伴うコーンのうち前記論理的な変更箇所および前記外部入出力扱いとしたノードを含んだコーンの入出力を外部入出力扱いとして前記論理的な変更箇所を含む回路記述を切り出しモジュール化する論理の変更箇所特定過程を具備し、

前記モジュール化した前記変更前と変更後の回路記述中の回路部それぞれに対し前記シミュレーション検証過程が実行されることを特徴とする半導体集積回路の設計検証方法。

【請求項 10】

期待値入力手段により、前記シミュレーション検証過程における前記変更後の回路記述に関する期待値を入力する過程と、

20

比較判定手段により、前記シミュレーション検証過程における実際のシミュレーション結果と前記期待値とを比較する比較判定過程とを具備することを特徴とする請求項 8 又は 9 に記載の半導体集積回路の設計検証方法。

【請求項 11】

比較判定手段は、前記シミュレーション検証過程において前記変更前と変更後の回路記述に不一致が発生したテストベクトル（入力信号）についてこのテストベクトルを用いた前記変更後の回路記述に関する回路部の出力を新たな期待値として登録し、前記イベントドリブン方式のシミュレーションの続行を可能とする過程とを具備したことを特徴とする請求項 10 に記載の半導体集積回路の設計検証方法。

30

【請求項 12】

コンピュータに、

変更前と変更後の回路記述を入力させる過程と、

前記入力された前記回路記述中においてレジスタの出力および信号の入出力ノードの配置関係を表すキーポイントのマッピングを行わせ、前記変更前と変更後の回路記述の不一致箇所から、最も近くの前記変更前と変更後の回路記述の一致しているキーポイントを外部入出力とし、変更箇所を含む回路部をモジュール化させる変更箇所特定過程と、

前記モジュール化された前記変更前と変更後の回路記述中の回路部それぞれに対しイベントドリブン方式のシミュレーションを利用した回路の機能検証を行わせるシミュレーション検証過程と、

40

前記モジュール化した各回路部において外部入出力扱いとしたノードのみと前記変更前と変更後の回路記述それぞれに対して形式検証を利用し回路の機能を検証する形式検証過程と、

前記形式検証過程で論理的不一致が発生した場合、前記変更前と変更後の回路記述に対し論理的な変更箇所があるとして前記形式検証に伴うコーンのうち前記論理的な変更箇所および前記外部入出力扱いとしたノードを含んだコーンの入出力を外部入出力扱いとして前記論理的な変更箇所を含む回路記述を切り出してモジュール化させる論理の変更箇所特定過程を具備し、

前記モジュール化された前記変更前と変更後の回路記述中の回路部それぞれに対し前記シミュレーション検証過程を実行させるプログラムを記憶したコンピュータ読み取り可能

50

な記憶媒体。

【請求項13】

前記コンピュータに、前記シミュレーション検証過程における前記変更後の回路記述に関する期待値を入力させる過程と、

前記シミュレーション検証過程における実際のシミュレーション結果と前記期待値とを比較させる比較判定過程とを実行させるプログラムを記憶したことを特徴とする請求項12に記載のコンピュータ読み取り可能な記憶媒体。

【請求項14】

前記コンピュータに、前記シミュレーション検証過程において前記変更前と変更後の回路記述に不一致が発生したテストベクトル（入力信号）についてこのテストベクトルを用いた前記変更後の回路記述に関する回路部の出力を新たな期待値として登録させ、前記イベントドリブン方式のシミュレーションの続行を可能にする過程とを実行させるプログラムを記憶したことを特徴とする請求項13に記載のコンピュータ読み取り可能な記憶媒体。

10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、LSIの設計検証に係り、特に大規模、複雑な回路の機能検証について、高速で検証確度の高い機能検証が要求される半導体集積回路の設計検証装置、方法及び記憶媒体に関する。

20

【0002】

【従来の技術】

半導体集積回路の論理設計段階において、イベントドリブンシミュレータは、イベントが回路を通過してどのように処理されていくかをシミュレーションしていくものである。イベントとは検証対象の回路に対して準備される複数種類の入力信号をいう。

【0003】

イベントドリブンシミュレータによる機能検証方法は、集積回路の規模が大きくなるほど、扱われるイベントが著しく増加し、シミュレーション時間が膨大となる。この結果、設計期間を長期化させる。

【0004】

また、イベントドリブンシミュレータは、検証したい動作をするための回路への入力信号の組み合わせ（イベント）を設計者が考えなければならない。設計者は、上記イベントを、回路の個々の事象に関しテストベクトルとして作成する。

30

【0005】

複雑な機能を有する回路は、その機能を検証する際、全ての事象を網羅したようなテストベクトルを人手で作成することは不可能となってきた。このため、イベントドリブンシミュレータは、大規模で複雑な機能を有する回路に対し、検証確度は低下する傾向にある。

【0006】

一方、形式検証は、イベントドリブンシミュレータにおける上記問題点を解決する技術として注目されている。形式検証は、設計した回路記述を論理式に変換し、その論理式を用いて機能検証を行う。形式検証は、検証対象の設計回路を反映した論理式から機能検証が行われる（等価性検証機能）ので、テストベクトルが不要であり、検証確度は100%である。

40

【0007】

形式検証は、主に2種類の技法を有する。第1の技法は、回路全体をそのまま論理式に変換する技法である。第2の技法は、回路をコーンと呼ばれる小さい部分に分割し、それぞれのコーン単位で論理式に変換する技法である。図7に、一つのコーンの概念を示す。すなわち、機能ブロック近傍のレジスタ出力、外部入出力を単位に論理式に対応する回路部を検証比較するものである。

50

【 0 0 0 8 】

上記において、第1の技法は、論理式が大きくなりすぎ、メモリ容量の問題により、大規模集積回路に適用するのは事実上不可能である。また、第2の技法は上記問題を解消し得る。しかし、第2の技法は、コン単位で検証することが前提となるので、回路に含まれるレジスタ構成が同じでないと、設計検証比較ができないという制約が発生する。

【 0 0 0 9 】

例えば、RTレベル設計（レジスタ転送レベル設計）等、レジスタへ変換される設計入力などのレベル（段階）の上位設計から、ゲートレベルの記述に移行した結果、タイミング調整の必要性からバッファを挿入したい箇所ができたとする。このとき、レジスタ構成を変更せずに設計変更が完了できれば、形式検証による等価性検証機能を利用できる。

10

【 0 0 1 0 】

すなわち、形式検証は、上位に対して下位の設計回路がある場合、これら2つの回路の等価性を調べるという等価性検証機能を用いるツールである。バグ修正のために機能を変更するような場合、形式検証では、機能の不一致が発生した箇所と、不一致が発生するときの入力の組み合わせが出力される。よって、設計者はある程度の変更には対応できる。

【 0 0 1 1 】

しかし、この形式検証は、回路記述を論理的に変換するという特性上、時間の概念が無い。このため、設計者が行った機能変更を既知のものとして検証対象とするような設定を、時間の概念を含めて設定することはできない。

【 0 0 1 2 】

【 発明が解決しようとする課題 】

形式検証は、等価性検証機能を用いるツールである。従って、設計変更に対処しにくく、現在の設計フローの中では、形式検証の適用可能範囲は限られている。RTレベル設計時は、機能やレジスタ構成の変更が頻繁に行われる。従って、最も機能検証の必要とされるRTレベル設計等の上位設計において効果的に導入できない状況にある。

20

【 0 0 1 3 】

この発明は、上記事情を考慮し、その課題は、上位設計においても形式検証を効果的に導入でき、高速で検証確度の高い機能検証環境を構築する半導体集積回路の設計検証装置、方法及び記憶媒体を提供することにある。

【 0 0 1 4 】

【 課題を解決するための手段 】

この発明の半導体集積回路の設計検証装置、方法及び記憶媒体は、変更前後の回路記述が入力され、特定手段により、前記入力手段により入力された変更前後の回路記述におけるレジスタの出力及び信号の入出力ノードの配置関係を表すキーポイントの不一致箇所に応じて変更箇所を特定し、前記特定手段による特定の結果、前記キーポイントの一致を保証できる回路記述に対して形式検証を行い、前記特定手段による前記特定の結果、前記変更前後の回路記述において前記キーポイントの一致を保証できない回路記述、もしくは前記形式検証により論理的不一致が発生した場合、論理的な変更箇所を含む回路記述に対してはイベントドリブン方式のシミュレーションを利用して回路の機能検証を行うことを特徴とする。

30

40

【 0 0 1 5 】

この発明によれば、検証対象となる半導体集積回路に関し、変更箇所以外は形式検証を利用するので高速な機能検証が実現される。さらに、変更箇所のみを切り出したのイベントドリブン方式のシミュレーションは高速である。よって、大規模な集積回路に対しても高速な機能検証を実現可能とする環境が得られる。

【 0 0 1 6 】

【 発明の実施の形態 】

図1は、この発明の第1の実施形態に係る半導体集積回路の設計検証方法を示すフローチャートである。この発明の設計検証は、まず、図中少なくとも次の(A)、(B)に示す処理フローが重要である。

50

【 0 0 1 7 】

(A) 変更前後の回路記述 1 , 2 を入力し、不一致箇所に応じて変更箇所を特定する。すなわち、処理 3 に示すように、回路記述 1 , 2 両者における入出力信号ノードの配置関係を表わすキーポイントのマッピングを行う。変更の程度によって全キーポイントに関してマッピングできる場合もあれば一部できない場合もある。このマッピングにより、不一致箇所が認められれば、不一致箇所に応じて変更箇所への入出力を外部入出力扱いとする。

【 0 0 1 8 】

すなわち、不一致箇所から最も近くの上記変更前後の回路記述の一致しているキーポイントを外部入出力扱いとする。全キーポイントに関して一部がマッピングされていなくても、最も近くのマッピングされている上記変更前後の回路記述の一致しているキーポイントを指定し、外部入出力扱いとすることができる。つまり、変更箇所を含む回路記述部分のみをモジュール化し、変更箇所の特定をする。

10

【 0 0 1 9 】

また、処理 6 に示すように、非変更箇所となつて、形式検証された回路部のうち、論理的な不一致が発生する場合があります。この論理的な変更箇所を含む回路部をモジュール化し、論理的変更箇所の特定をする。

【 0 0 2 0 】

上記処理 6 では、形式検証に伴うコーンのうち、特定された論理的な変更箇所およびその周辺における形式検証前に外部入出力扱いとしたノードを含むコーンの入出力を外部入出力扱いとして、この論理的な変更箇所を含む回路記述を切り出しモジュール化する。

20

【 0 0 2 1 】

(B) 上記変更前後の回路記述 1 , 2 においてキーポイントの一致を保証できる回路記述に対して形式検証を行い(処理 4 , 5)、かつ上記キーポイントの一致、もしくは等価性を保証できない回路記述に対してはイベントドリブン方式のシミュレーションを利用して検証を行う(処理 7 , 8 , 9 , 1 0)。

【 0 0 2 2 】

すなわち、処理 4 では、前記変更前後の回路記述それぞれに対して形式検証を利用し回路を検証する。このとき、上記処理 3 でモジュール化した回路部があるなら、各回路部において上記外部入出力扱いとしたノードのみを反映させる。つまり、モジュール化した変更箇所を含む回路部はブラックボックス扱いとしながら、変更前後の回路記述それぞれに対して全体的な形式検証を行う。

30

【 0 0 2 3 】

処理 5 では、上記形式検証の結果、すなわち、等価性が判定される。論理的な不一致箇所が無ければ、変更箇所以外の全体の形式検証は等価とみなされる(検証終了(17))。また、論理的な不一致箇所がある場合は、変更箇所として特定される(処理 6)。

【 0 0 2 4 】

処理 7 では、処理 8 のイベントドリブン方式のシミュレーションを利用に際し、上述の処理 3 (さらに必要なら処理 6)によりモジュール化された各回路部の入力キーポイントにシミュレーション時間の区切られたランダムなテストベクトル(入力信号)を繰り返し回路に入力するためのテストベンチを生成する。

40

【 0 0 2 5 】

処理 8 では、テストベンチに応じて、上記モジュール化された変更前後の各回路部の入力キーポイントにテストベクトル(入力信号)が繰り返し入力され、イベントドリブン方式のシミュレーションを利用したシミュレーション検証を行う。

【 0 0 2 6 】

処理 9 では、上記モジュール化された変更前後の回路部におけるシミュレーション結果を比較する。そして、処理 1 0 では、上記シミュレーション結果が等価であるか不一致であるかが判定される。この判定で等価と判定されると検証終了である(17)。

【 0 0 2 7 】

50

上記実施形態の方法によれば、検証対象となる半導体集積回路に関し、変更箇所以外は形式検証を利用するので高速な機能検証ができる。さらに、変更箇所のみを切り出してのイベントドリブン方式のシミュレーションは高速である。よって、大規模な集積回路に対しても高速な機能検証を実現可能とする。

【0028】

RTレベルの機能設計においてレジスタ等の機能ブロックが構成されている場合、変更箇所を自動的に判定し、変更された箇所のみ切り出しモジュール化することができる。もちろん変更箇所が複数コーンに隣接していたり重なり合っている場合でも変更箇所を切り出してモジュール化することができる。

【0029】

すなわち、レジスタ構成の変更されていない回路記述部分は、形式検証によって高速に機能検証が行われる。レジスタを越えて組合せ回路を移動するような場合にも対応できる。

【0030】

また、上記処理7の利点は次のようである。自動生成されるランダムなテストベクトルを用いることによってテストベクトルの作成ミスによる検証抜けの防止、テストベクトル作成作業の省略をすることができる。

【0031】

イベントドリブンシミュレータ用のテストベンチを自動生成することによってテストベンチの作成ミスの防止、テストベンチ作成作業の省略に寄与する。さらに、シミュレーション時間を区切ったランダムなテストベクトルを繰り返し入力する方式を用いるので、不一致が発生した際のデバッグ効率を向上させる。

【0032】

処理8では、変更箇所のみ切り出した小規模な回路に対して自動生成されたランダムなテストベクトルでのシミュレーション検証が行われる。このため、大量なテストベクトルを実行することが可能となり、検証確度を向上させる。

【0033】

さらに、上で説明した主なフローに付随する各処理について再び図1を参照して説明する。

上記処理9の変更前後のモジュール化された回路部についてのシミュレーション結果の比較において、結果の異なる値になった場合、それが設計者の予測どおりの値となることがある。

【0034】

処理11では、このようなシミュレーション結果の予測される期待値を例えば波形表示ツールのGUI(グラフィック・ユーザ・インタフェイス)を用いて入力する。このような期待値は、例えばRTレベルの回路記述段階における変更前後の回路の変更箇所を指定することでその回路記述を解析し、変更前後で異なる実行結果が得られるテストベクトル(入力信号)とその実行結果から作成してもよい。

【0035】

さらに、処理12において、上記処理8,9を経た実際のシミュレーション結果から、不一致の発生したテストベクトル(入力信号)とこのテストベクトルを用いた前記変更後の回路の出力を上記期待値と比較する。その後、上述の処理10につながり、期待した変更がなされているか否か判定が行われる。

【0036】

また、処理13では、上述した処理10のシミュレーション結果の判定を経て、不一致の発生したテストベクトル(入力信号)とその結果出力をファイル出力または波形表示出力する。

【0037】

さらに処理14では、上記処理13において表示された変更前後の回路部の不一致が発生した箇所の確認、判定が行われる。ここでの判定とは、表示された不一致が、設計者の期待したとおりの変更であるかが判断される。

10

20

30

40

50

【 0 0 3 8 】

処理 1 4 において、変更が設計者の期待したとおりである場合には、この不一致に至るテストベクトル（入力信号）およびその変更後の回路部の出力を新たな期待値として追加登録し（15）、イベントドリブンシミュレータでのシミュレーションの続行を可能とする。また、表示された不一致が、設計者の期待したとおりの変更でない場合、シミュレーション結果を参照してデバッグすることになる（16）。

【 0 0 3 9 】

上記実施形態の方法において、処理 9 によれば、変更前後の回路でシミュレーション結果を比較することによって、機能を変更した部分のテストベクトルと回路出力のみ選別できる。

10

【 0 0 4 0 】

また、処理 1 1 によれば、変更箇所の記述を解析し、それに基づいて期待値を作成することにより、期待値の作成ミス、作成抜けを防止する。そして、処理 1 2 によれば、機能変更を行った際、期待値と、変更前後の回路で不一致となったテストベクトル、回路出力を比較することにより、期待した変更のみ実施されたかどうかを自動的に判定できる。

【 0 0 4 1 】

また、処理 1 3 によれば、イベントドリブンシミュレータよりシミュレーション結果の出力を波形表示する機能を持つ。これにより、GUIを用いたデバッグ環境を提供する。

【 0 0 4 2 】

さらに処理 1 4、1 5 によれば、不一致箇所の検証を行うための環境を自動生成することになる。これにより、ケアレスミスや設計者の意図していない機能変更によるバグ検出あるいはバグ混入が防止される。また、期待値の作成ミスを無くし、迅速に期待値の設定を行うことができるのでデバッグ効率が向上する。

20

【 0 0 4 3 】

図 2 は、この発明の第 2 の実施形態に係る半導体集積回路の設計検証装置を示すブロック図である。この設計検証装置は、上記第 1 の実施形態で示した検証方法のフローを実現する。

【 0 0 4 4 】

データ入力部 2 1 は、変更前後の回路記述（1, 2）を入力する。処理部 2 0 0 中のマッピング変更箇所特定手段 2 2 は、変更前後の回路記述中においてレジスタの出力および信号の入出力に対応したキーポイントのマッピングを行う。不一致箇所がある場合は、不一致箇所から最も近くの一一致しているキーポイントを外入出力とし、変更箇所を含む回路部をモジュール化する。

30

【 0 0 4 5 】

キーポイントのマッピングは、変更の程度によっては全キーポイントに対し一部がマッピングできなくなることもある。しかし、最も近くのマッピングされている上記変更前後の回路記述の一一致しているキーポイントを指定し、外部入出力扱いとすることができる。つまり、変更箇所を含む回路記述部分のみをモジュール化し、変更箇所の特定をする。

【 0 0 4 6 】

処理部 2 0 0 中の形式検証機構 2 3 は、上記モジュール化した各回路部においては上記外部入出力扱いとしたノードのみを反映させながら前記変更前後のそれぞれに対して形式検証を利用し回路全体を検証する。

40

【 0 0 4 7 】

ここで、論理的不一致箇所特定手段 2 4 は、上記形式検証機構 2 3 で論理的不一致が発生した場合、形式検証に伴うコーンのうち、特定された論理的な変更箇所およびその周辺における形式検証前に外部入出力扱いとしたノードを含んだコーンの入出力を外入出力扱いとして回路記述を切り出しモジュール化する。

【 0 0 4 8 】

処理部 2 0 0 中のイベントドリブンシミュレータ 2 5 は、モジュール化された変更前後の回路部それぞれに対し、シミュレーションする。モジュール化された回路部とは、上記マ

50

ッピング変更箇所特定手段 2 2 によってモジュール化された回路部であり、また、論理的変更箇所特定手段 2 4 によってモジュール化された回路部もあれば加える、上記シミュレーションは、入力キーポイントにテストベンチに応じてシミュレーション時間の区切られたランダムなテストベクトル（入力信号）を繰り返し入力するイベントドリブン方式のシミュレーションを利用した検証である。

【 0 0 4 9 】

変更前後の回路記述について、上記のシミュレーション結果を比較する比較判定機構 2 6 が設けられる。出力部 2 7 では、この比較判定機構 2 6 により不一致が発生した場合、不一致の発生したテストベクトル（入力信号）とその結果出力を得る。この結果出力は例えばファイル出力や波形表示等モニタ出力である。

10

【 0 0 5 0 】

さらに、図 2 中に付随する機能について説明する。

期待値入力手段 3 1 は、シミュレーション結果の予測される期待値を例えば波形表示ツールの GUI（グラフィック・ユーザ・インタフェース）を用いて入力する。期待値は、上記に限らず、例えば RT レベルの回路記述段階における変更前後の回路の変更箇所を指定することでその回路記述を解析し、変更前後で異なる実行結果が得られるテストベクトル（入力信号）とその実行結果から作成してもよい。

【 0 0 5 1 】

比較判定機構 2 6 は、実際のシミュレーション結果から、不一致の発生したテストベクトル（入力信号）とこのテストベクトルを用いた前記変更後の回路の出力を上記期待値と比較する。すなわち、期待した変更がなされているか否か判定が行われる。

20

【 0 0 5 2 】

出力部 2 7 では、上述したシミュレーション結果の判定を経て、不一致の発生したテストベクトル（入力信号）とその結果出力をファイル出力または波形表示出力する。

【 0 0 5 3 】

さらに比較判定機構 2 6 では、変更前後の回路部の不一致が発生した箇所の確認、判定が行われる。すなわち、上記表示された不一致が、設計者の期待したとおりの変更である場合には、この不一致に至るテストベクトル（入力信号）およびその変更後の回路部の出力を新たな期待値として追加登録し、イベントドリブンシミュレータでのシミュレーションの続行を可能とする。また、表示された不一致が、設計者の期待したとおりの変更でない

30

【 0 0 5 4 】

上記実施形態によれば、第 1 の実施形態と同様の効果を有する。すなわち、検証対象となる半導体集積回路に関し、変更箇所以外は形式検証を利用するので高速な機能検証ができる。さらに、変更箇所のみを切り出してのイベントドリブン方式のシミュレーションは高速である。よって、大規模な集積回路に対しても高速な機能検証を実現可能とする。

【 0 0 5 5 】

RT レベルの機能設計においてレジスタ等の機能ブロックが構成されている場合、変更箇所を自動的に判定し、変更された箇所のみ切り出しモジュール化することができる。もちろん変更箇所が複数コーンに隣接していたり重なり合っている場合でも変更箇所を切り出してモジュール化することができる。

40

【 0 0 5 6 】

すなわち、レジスタ構成の変更されていない回路記述部分は、形式検証によって高速に機能検証が行われる。レジスタを越えて組合せ回路を移動するような場合にも対応できる。

【 0 0 5 7 】

比較判定機構 2 6 では、変更前後の回路でシミュレーション結果を比較する。これにより、機能を変更した部分のテストベクトルと回路出力のみ選別できる。また、期待値入力手段 3 1 は、期待値の作成は、期待値の作成ミス、作成抜けを防止する。そして、機能変更を行った際、期待値と、変更前後の回路で不一致となったテストベクトル、回路出力を比較することにより、期待した変更のみ実施されたかどうかを自動的に判定できる。

50

【 0 0 5 8 】

さらに、比較判定機構 26 では、不一致箇所の検証を行うための環境を自動生成することになる。これにより、ケアレスミスや設計者の意図していない機能変更によるバグ検出あるいはバグ混入が防止される。また、期待値の作成ミスを無くし、迅速に期待値の設定を行うことができるのでデバッグ効率が向上する。

【 0 0 5 9 】

また、出力部 27 は、イベントドリブンシミュレータよりシミュレーション結果の出力を波形表示する機能を持つ。これにより、GUI を用いたデバッグ環境を提供する。

【 0 0 6 0 】

上記第 1、第 2 の実施形態を踏まえ、この発明の設計検証方法について (i) ~ (iii) のケースを例に、より具体的に説明する。なお、図 3 (a), (b) それぞれは、ケース (ii) で用いる変更前の組合せ回路部、変更後の組合せ回路部を示す。図 4 (a), (b) それぞれは、ケース (ii) で用いる変更前回路部の入出力波形、変更後回路部の入出力波形を示す。図 5 は、RT レベル設計の変更前後のソースコードからの変更箇所の期待値の生成の様子を示す説明図である。図 6 (a), (b) それぞれは、ケース (iii) で用いる変更前の組合せ回路部、変更後の組合せ回路部を示す。

10

【 0 0 6 1 】

(i) <レジスタ構成変更：無し、機能変更：無し>

例：ゲートレベル記述のバッファ挿入、セルの置き換え、タイミングチューニング。

【 0 0 6 2 】

(1) 入力された変更前後の回路記述における全キーポイント（レジスタ出力、外部入出力）に関してマッピングできる（処理 3）。

(2) 形式検証実行（処理 4）。

上記 2 つの回路は等価であるというレポート出力（処理 5）。

(3) 機能検証終了。

すなわち、上記 (i) のケースでは、形式検証のみで所望の設計検証は達成できる。

20

【 0 0 6 3 】

(ii) <レジスタ構成変更：無し、機能変更：有り>

例 1：ゲートレベル記述のバッファ挿入、セルの置き換え、タイミングチューニングでバグが混入した場合。

30

例 2：RT レベル記述でレジスタ間の組合せ回路の一部を他のレジスタ間へ移動（リソースシェリング）。

例 3：RT レベル記述で組合せ回路記述変更時にバグが混入した場合。

【 0 0 6 4 】

上記各例のうち、例 2 の場合で説明する。

(1) 入力された変更前後の回路記述における全キーポイント（レジスタ出力、外部入出力）に関してマッピングできる（処理 3）。

(2) 形式検証実行（処理 4）。

(3) 不一致箇所がレポート出力される。図 3 では変更前回路の組合せ回路 A と変更後回路の組合せ回路 A, B、変更前回路の組合せ回路 B, C と変更後回路の組合せ回路 C の 2 箇所が機能不一致としてレポート出力される（処理 5）。

40

【 0 0 6 5 】

(4) 複数コーンが不一致であった場合、それらの回路を全て切り出すため、外部入出力信号から順次信号をトレースしていき、論理的不一致点の入出力に最初に至った点を外部入出力とし、回路の切り出しを行う。この図 3 では、変更前後それぞれの回路部に関し、ノード a, b, e を外部入力、ノード d, f を外部出力として点線内部の回路を切り出し、モジュール化する（処理 6）。

【 0 0 6 6 】

(5) 切り出された回路をイベントドリブンシミュレーションするためのテストベンチと、入力に一定のシミュレーション時間で区切られたランダムな信号を繰り返し入力するシ

50

ミュレーション記述を作成する。ランダムな信号を入力する際、シミュレーション時間を区切って、繰り返し行う利点は、後でデバッグをする際にテストベクトルを取り扱うのに効率がよいからである（処理 7）。

【 0 0 6 7 】

(6) シミュレーション実行（処理 8）。

(7) 切り出した変更前後の 2 つの回路出力を順次比較する。v c d ファイルやシステムタスクを使用し、シミュレーション時間毎の信号変化をファイル出力させ、比較を行う（処理 9）。

【 0 0 6 8 】

(8) 不一致が発生した場合、入力してある期待値があれば、その期待値と比較する。期待値と一致している場合はカバレッジを向上させるためシミュレーションを続ける（処理 1 2）。期待値は、例えば設計者が変更箇所を認識しており、動作が異なる場合の入出力状態が既知の場合、シミュレーション途中でも登録できる。

10

【 0 0 6 9 】

(9) 期待値と一致していない、もしくは期待値が入力されていない場合は不一致の発生する際のテストベクトルと出力結果をレポートし、波形表示可能なファイル形式で不一致の発生したランダム入力信号と出力信号を出力する（処理 1 3）。

【 0 0 7 0 】

(10) 波形表示ツールとテストベンチ、および出力されたテストベクトルを用いてデバッグする。図 4 の場合、波形表示ツールの比較機能を使用して変更前後の 2 つの波形を比較すると、斜線部分が不一致箇所であると判定できる。設計者はその波形を見て、設計者の変更のとおりであるかを判断する（処理 1 4）。

20

【 0 0 7 1 】

上記判断により結果期待どおりの結果であれば、斜線部分を期待値として登録し（処理 1 5）、シミュレーションを続行する。確度が十分になったら検証を終了する。期待どおりでなければ、デバッグ作業（処理 1 6：図示せぬ別フロー）に入る。

【 0 0 7 2 】

ここで、上記のような期待値の入力方法を説明する。例えば、波形表示ツールの G U I を使用し、グラフィカルに入力する技法である。図 4 の場合は、斜線内の入出力信号ベクトルが期待値となるので、このパターンを予め入力しておく。

30

【 0 0 7 3 】

また、例えば、R T レベル設計の変更前後のソースコードから変更箇所の期待値を生成する技法である。図 5 のように、入力記述が R T レベルの場合、ソースコードの変更部分をクリックすることで、その変更部分を実行するための条件式を解析し、期待値ベクトルを作成する。

【 0 0 7 4 】

この図 5 の場合、i f 文の条件式が変更可能であるので、C L K 立ち上がり時に I N P U T A = 1 かつ I N P U T B または C のどちらかが 1 である場合は O U T P U T B が b になるベクトルを期待値として設定する。

【 0 0 7 5 】

40

(iii) < レジスタ構成変更：無し、機能変更：有り >

例：R T レベル記述でのバグ修正。仕様変更、または高速化のための記述変更。

(1) 入力される変更前後の回路記述においてマッピングできないキーポイントが発生する。複数のコーンで不一致箇所が認められた場合、それらの回路を全て切り出すため、外部入出力信号から順次信号をトレースしていき、マッピングできないキーポイントに最も近いマッピングされているキーポイントを外部入出力として回路を切り出し、モジュール化を行う（処理 3）。例えば図 6 では、ノード a , b を外部入力、ノード c , d を外部出力として回路を切り出しモジュール化する。

【 0 0 7 6 】

(2) 切り出された回路をイベントドリブンシミュレーションするためのテストベンチと

50

、入力に一定のシミュレーション時間で区切られたランダムな信号を繰り返し入力するシミュレーション記述を作成する。ランダムな信号を入力する際、シミュレーション時間を区切って、繰り返し行う利点は、後でデバッグをする際にテストベクトルを取り扱うのに効率がよいからである（処理 7）。

【 0 0 7 7 】

(3) シミュレーション実行（処理 8）。

(4) 切り出した変更前後の 2 つの回路出力を順次比較する。v c d ファイルやシステムタスクを使用し、シミュレーション時間毎の信号変化をファイル出力させ、比較を行う（処理 9）。

【 0 0 7 8 】

(5) 不一致が発生した場合、入力してある期待値があれば、その期待値と比較する。期待値と一致している場合はカバレッジを向上させるためシミュレーションを続ける（処理 1 2）。期待値は、例えば設計者が変更箇所を認識しており、動作が異なる場合の入出力状態が既知の場合、シミュレーション途中でも登録できる。なお、期待値の入力方法は、上記具体例（ii）で述べたのと同様である。

【 0 0 7 9 】

(6) 期待値と一致していない、もしくは期待値が入力されていない場合は不一致の発生する際のテストベクトルと出力結果をレポートし、波形表示可能なファイル形式で不一致の発生したランダム入力信号と出力信号を出力する（処理 1 3）。

【 0 0 8 0 】

(7) 波形表示ツールとテストベンチ、および出力されたテストベクトルを用いてデバッグする。例えば、波形表示ツールの比較機能を使用して変更前後の 2 つの波形を比較し、設計者の期待どおりの結果であれば、必要な部分を期待値として登録し、シミュレーションを続行する（処理 1 4、1 5）。

【 0 0 8 1 】

(8) 一方、上記(1)で切り出されモジュール化された回路部への入出力を外部入出力として取り扱う。モジュール化した変更箇所を含む回路部はブラックボックス扱いとしながら、変更前後の回路記述それぞれに対して全体的な形式検証を行う（処理 4）。

【 0 0 8 2 】

(9) 不一致点がある場合は、上記具体例（ii）における（3）以降の処理と同様の処理を経て検証を終了する。

【 0 0 8 3 】

以上のような設計検証により、検証対象となる半導体集積回路に関し、変更箇所以外は形式検証により高速な機能検証が、さらに、変更箇所のみを切り出して高速にかつ検証確度の高いイベントドリブン方式のシミュレーションが実現できる。

【 0 0 8 4 】

また、コンピュータを動作させるための、少なくとも半導体集積回路の設計検証プログラムを記憶した、コンピュータで読み取り可能な記憶媒体において、前記図 1 の処理フローに含まれる本発明に係る基本的な設計検証方法がプログラムされているものも本発明の範囲内である。

【 0 0 8 5 】

すなわち、本発明に係る記憶媒体は、変更前後の回路記述を入力させ、不一致箇所に応じて変更箇所を特定させて、変更前後の回路記述においてレジスタの出力及び信号の入出力に対応したキーポイントの一致を保証できる回路記述に対しては形式検証を行うように、また、キーポイントの一致もしくは等価性を保証できない回路記述に対してはイベントドリブン方式のシミュレーションを利用して検証を行うようにプログラムされている。

【 0 0 8 6 】

【 発明の効果 】

以上、説明したように、この発明によれば、最も機能検証の必要とされる R T レベル設計等の上位設計において、形式検証を効果的に導入できる。すなわち、変更前後の回路記述

10

20

30

40

50

において、変更箇所以外は形式検証を利用するので高速な機能検証ができる。

【0087】

さらに、変更箇所のみを切り出したイベントドリブン方式のシミュレーションは高速化、確度向上が可能である。よって、大規模な集積回路に対しても高速で信頼性の高い機能検証環境を実現可能とする半導体集積回路の設計検証装置、方法及び記憶媒体が提供できる。

【図面の簡単な説明】

【図1】この発明の第1の実施形態に係る半導体集積回路の設計検証方法を示すフローチャート。

【図2】この発明の第2の実施形態に係る半導体集積回路の設計検証装置を示すブロック図。

【図3】(a), (b)それぞれは、半導体集積回路の設計検証の具体例を説明するための変更前の組合せ回路部、変更後の組合せ回路部を示す第1の回路ブロック図。

【図4】(a), (b)それぞれは、半導体集積回路の設計検証の具体例を説明するための変更前回路部の入出力波形、変更後回路部の入出力波形を示すタイミングチャート。

【図5】RTLレベル設計の変更前後のソースコードからの変更箇所の期待値の生成の様子を示す説明図。

【図6】(a), (b)それぞれは、半導体集積回路の設計検証の具体例を説明するための変更前の組合せ回路部、変更後の組合せ回路部を示す第2の回路ブロック図。

【図7】形式検証におけるコーンを示す概念図。

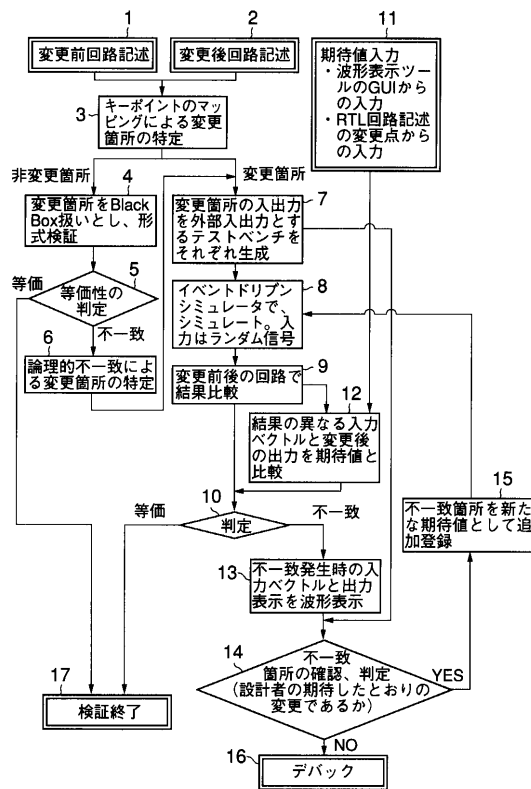
【符号の説明】

1 ~ 17 ... 設計検証のための各処理

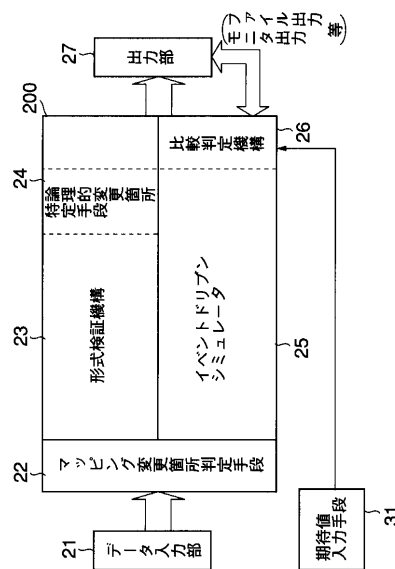
10

20

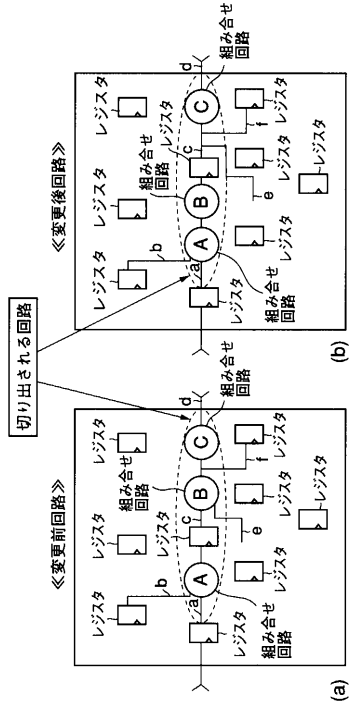
【図1】



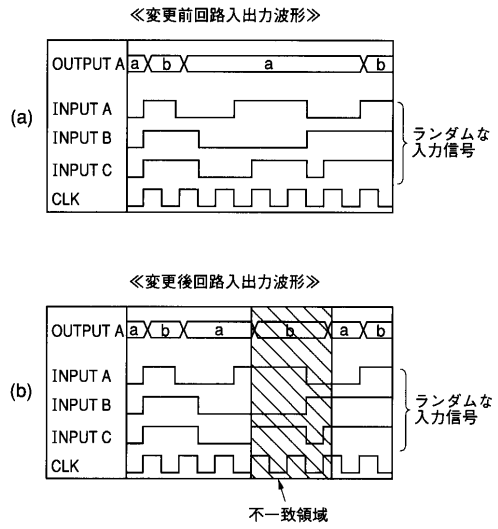
【図2】



【 図 3 】



【 図 4 】



【 図 5 】

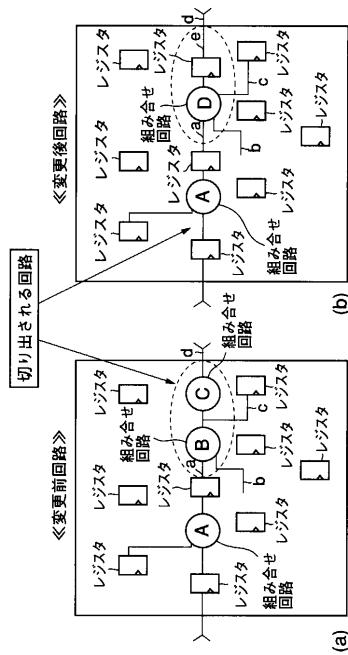
```

<<<変更前RTLソースコード>>>
always @(posedge CLK) begin
  if (INPUT A && INPUT B || INPUT C)
    OUTPUT A <= b
  else
    OUTPUT A <= a
end
変更箇所
<<<変更後RTLソースコード>>>
always @(posedge CLK) begin
  if (INPUT A && (INPUT B || INPUT C))
    OUTPUT A <= b
  else
    OUTPUT A <= a
end

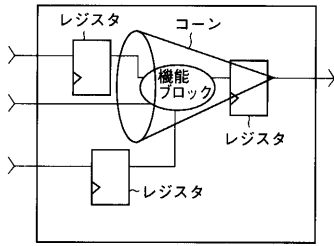
```

・上記記述から生成される変更箇所の期待値
 入力 : INPUT A, INPUT B, INPUT C
 出力 : OUTPUT A
 クロック : CLK
 期待値 1 : 1 0 1
 期待値 2 : 1 1 0

【 図 6 】



【 図 7 】



フロントページの続き

(74)代理人 100070437

弁理士 河井 将次

(72)発明者 土屋 丈彦

神奈川県川崎市幸区堀川町580番1号 株式会社東芝半導体システム技術センター内

審査官 松浦 功

(56)参考文献 特開平09-204460(JP,A)

特開平03-164978(JP,A)

特開平10-269271(JP,A)

特開平10-334131(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 17/50

H01L 21/82