(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2005-141624
(P2005-141624A)
(43)公開日　平成17年6月2日(2005.6.2)

| (51)Int.Cl.[7] | F I | | テーマコード（参考） |
|---|---|---|---|
| G06F 17/50 | G06F 17/50 | 664A | 5B042 |
| G06F 11/28 | G06F 17/50 | 664G | 5B046 |
| | G06F 17/50 | 672C | |
| | G06F 11/28 | 340C | |

審査請求 未請求　請求項の数 10　ＯＬ　（全 16 頁）

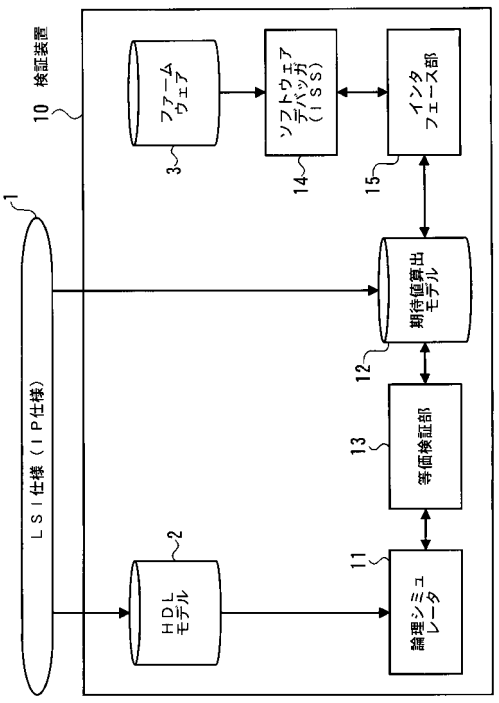
(21)出願番号　特願2003-379532 (P2003-379532)
(22)出願日　平成15年11月10日 (2003.11.10)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100092152
弁理士　服部　毅巖
(72)発明者　岩本　正美
神奈川県川崎市高津区坂戸３丁目２番１号　富士通エルエスアイテクノロジ株式会社内
(72)発明者　小澤　裕一
神奈川県川崎市高津区坂戸３丁目２番１号　富士通エルエスアイテクノロジ株式会社内

Ｆターム(参考)　5B042 GA13　GC15　HH06　HH07
5B046 AA08　BA03　JA05

(54)【発明の名称】検証装置、検証方法およびプログラム

(57)【要約】
【課題】　システムＬＳＩ開発におけるハードウェアおよびソフトウェアの検証を精度良く効率的に行う。
【解決手段】　ハードウェア検証時には、ＨＤＬモデル２の論理シミュレータ１１によるシミュレーション結果と、期待値算出モデル１２が生成する期待値とが、等価検証部１３で比較、検証される。ソフトウェア検証時には、期待値算出モデル１２がインタフェース部１５を介して利用され、ファームウェア３がソフトウェアデバッガ１４により検証される。期待値算出モデル１２は、ハードウェア検証時には期待値生成モデルとして用いられ、ソフトウェア検証時にはハードウェアのＣモデルとして用いられる。このように期待値算出モデル１２をハードウェア検証とソフトウェア検証の双方に用いることにより、精度良く効率的に検証が行えるようになる。
【選択図】　図１



【特許請求の範囲】
【請求項１】
ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証装置において、
検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする検証装置。
【請求項２】
前記モデルは、複数の異なる検証レベルに対応する数のタイミング情報を有し、検証時の検証レベルに応じた前記タイミング情報に従って入出力データの転送方式を切り替えることを特徴とする請求項１記載の検証装置。
【請求項３】
前記モデルは、検証時の入出力データを変換するデータ変換手段を有し、検証時に入力されるデータを前記データ変換手段によって変換して演算を実行し、演算後のデータを前記データ変換手段によって再変換して出力することを特徴とする請求項１記載の検証装置。
【請求項４】
前記モデルは、検証時の検証レベルに応じたタイミング情報に従って入出力データを転送するインタフェースと、前記インタフェースから転送されるデータを用いて演算を実行するハードウェアモデルと、前記ハードウェアモデルによって演算され前記インタフェースによって出力されるデータを格納するスタックメモリと、を有することを特徴とする請求項１記載の検証装置。
【請求項５】
ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証方法において、
検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする検証方法。
【請求項６】
前記モデルは、複数の異なる検証レベルに対応する数のタイミング情報を有し、検証時の検証レベルに応じた前記タイミング情報に従って入出力データの転送方式を切り替えることを特徴とする請求項５記載の検証方法。
【請求項７】
前記モデルは、検証時に入力されるデータを変換して演算を実行し、演算後のデータを再変換して出力することを特徴とする請求項５記載の検証方法。
【請求項８】
コンピュータを用いて、ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証するためのプログラムにおいて、
コンピュータに、
検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを用いてハードウェア検証を行い、
前記モデルを用いてソフトウェア検証を行う処理を実行させることを特徴とするプログラム。
【請求項９】
ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、
前記モデルは、前記ソフトウェア検証に使用されるタイミング情報以外のタイミング情報を有することを特徴とする検証装置。
【請求項１０】
ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検

証装置において、
前記モデルは、前記ソフトウェア検証に使用されるタイミング情報について、複数種別のソフトウェア検証に対応する数有することを特徴とする検証装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は検証装置、検証方法およびプログラムに関し、特にシステムＬＳＩ（Large Scale Integrated Circuit）のハードウェアおよびソフトウェアの開発・検証を行うための検証装置、検証方法およびプログラムに関する。
【背景技術】
【０００２】
近年のシステムＬＳＩの開発においては、その大規模化、多機能化に伴い、開発期間の長期化やコストの増加が問題となっており、短期間で効率良く、かつ、低コストで所望の高品質システムＬＳＩを製造することのできるハードウェアおよびソフトウェアの開発・検証環境の構築が要望されている。
【０００３】
図５は従来のハードウェアおよびソフトウェアの開発・検証環境の概念図である。
システムＬＳＩ開発では、まず、ＬＳＩ仕様の設計後、そのＬＳＩ仕様を基にアーキテクチャ設計を行ってシステムを機能モジュールに分割し、ハードウェアとそのハードウェアを制御するソフトウェアとの切り分けを行う。
【０００４】
ハードウェアは、例えばシステムＬＳＩに搭載されるＣＰＵ（Central Processing Unit）／ＭＰＵ（Micro Processing Unit）の周辺回路であり、通常、その論理機能のビヘイビアレベルでの記述を基に、ハードウェア記述言語（Hardware Description Language, ＨＤＬ）でその動作記述が行われる。
【０００５】
そして、この動作記述からハードウェアの論理機能を表現した設計データがレジスタ・トランスファ・レベル（Resistor Transfer Level, ＲＴＬ）で記述される。ハードウェア検証では、このようにＨＤＬを用いてＲＴＬ記述されたハードウェアモデル（ＲＴＬモデル）１０１を、テストパターン１０２を用い、論理シミュレータ１０３でシミュレートする。
【０００６】
さらに、ハードウェア検証では、そのシミュレーション結果と、ＲＴＬモデル１０１の基になったビヘイビアレベルのハードウェアモデル（期待値モデル）１０４からテストパターン１０２を用いて得られる期待値との等価性が等価検証部１０５で検証される。シミュレーション結果と期待値とが等価であるか否かによって、ＲＴＬモデル１０１の機能の検証が行われる。
【０００７】
なお、ここで用いられる期待値モデル１０４には、通常、ハードウェアにおける処理のタイミングといった時間的概念は考慮されない。ハードウェア検証後は、論理合成やレイアウト設計等の工程へと進む。
【０００８】
一方、ソフトウェアは、通常、Ｃ／Ｃ＋＋等のプログラミング言語で記述される。ソフトウェア検証では、ＣＰＵ／ＭＰＵの周辺回路を動作記述した周辺回路ブロック１０６を、ＬＳＩ仕様に合わせてＣ言語等で記述したバスインタフェース１０７およびバス１０８を介して、ＣＰＵ／ＭＰＵを動作記述したＣＰＵ／ＭＰＵブロック１０９に接続し、さらに、検証すべきソフトウェア（ファームウェア）１１０をメモリブロック１１１に格納した仮想プロトタイプが使用される。
【０００９】
この仮想プロトタイプを用いたソフトウェア検証では、周辺回路ブロック１０６として

Ｃ言語で記述されたハードウェアモデルが必要になるが、通常はＲＴＬモデル１０１をＣ言語に書き換えたプログラム（Ｃモデル）をソフトウェア検証用に作成し、これをソフトウェア検証時のハードウェアモデルとして使用する。このようなＣモデルを用い、ＣＰＵ／ＭＰＵの命令動作を実行する命令セットシミュレータ（Instruction Set Simulator, ＩＳＳ）をソフトウェアデバッガとして使用し、ほぼ実回路と同じ環境でソフトウェアのデバッグを行う。

【００１０】

デバッグ後は、検証後のソフトウェアを、ハードウェア検証を基に形成された実回路に搭載し、システム全体の検証が行われる。

このようなシステムＬＳＩ開発においては、最近では、開発期間の短縮を目的として、ソフトウェアデバッガの動作と同期させながらＲＴＬモデル１０１や期待値モデル１０４のシミュレーションを行って、ソフトウェアデバッガおよび論理シミュレータ１０３での各実行結果を比較する協調検証も行われている。

【００１１】

また、従来は、ハードウェア検証時のシミュレーションを高速化して開発期間を短縮する目的で、ＲＴＬモデルを用いたときのような詳細な（時刻精度が細かい）低速シミュレーションとビヘイビアレベルのモデルを用いたときのような簡略な（時刻精度が粗い）高速シミュレーションを、時刻精度と指定した切り替え時刻に基づいてモデルを切り替えて行うようにした提案もなされている（例えば特許文献１参照）。

【特許文献１】特開平１０－２６１００２号公報（段落番号〔００３７〕～〔００９２〕、図１）

【発明の開示】

【発明が解決しようとする課題】

【００１２】

しかし、従来のシステムＬＳＩ開発では、ソフトウェア検証のために、ＨＤＬで記述されたＲＴＬモデルをＣ言語に書き換えたＣモデルを作成する際、ＣモデルとＲＴＬモデルの間の等価性に問題があった。

【００１３】

そのため、ソフトウェア検証を行ったときに、仕様通りの結果が得られないなど、不具合が生じた場合には、その不具合がＲＴＬモデルに等価であるはずのＣモデルに起因するものであるのか、検証しているソフトウェアに起因するものであるのか、その特定が非常に困難になる。原因特定のためには、ＲＴＬモデルやＣモデルの再検証、システム全体の再検証が必要になり、ＣモデルとＲＴＬモデルの等価性は、システムＬＳＩ開発期間の長期化を招く一因となっていた。

【００１４】

また、ソフトウェア検証に用いるＣモデルは、検証精度を重視するか検証スピードを重視するかといった検証目的に応じた検証レベルごとに別々に作成していた。このようにＣモデルを検証レベルごとに別々に作成すると、モデル作成やそのモデルを用いたソフトウェアデバッグに大変な労力と時間が必要になり、これによってもシステムＬＳＩ開発の効率化が損なわれていた。

【００１５】

本発明はこのような点に鑑みてなされたものであり、システムＬＳＩ開発におけるハードウェアおよびソフトウェアの検証を精度良く効率的に行うことのできる検証装置を提供することを目的とする。

【００１６】

また、本発明は、システムＬＳＩ開発におけるハードウェアおよびソフトウェアの検証を精度良く効率的に行う検証方法を提供することを目的とする。

また、本発明は、システムＬＳＩ開発におけるハードウェアおよびソフトウェアの検証を精度良く効率的に行うための処理をコンピュータに実行させるためのプログラム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】
【００１７】
本発明では上記課題を解決するために、図１に示す構成で実現可能な検証装置が提供される。本発明の検証装置は、ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証装置において、検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする。
【００１８】
図１に示した検証装置１０によれば、ＨＤＬモデル２の検証の際には、ＨＤＬモデル２の論理シミュレータ１１によるシミュレーション結果と、期待値算出モデル１２が生成する期待値とが、等価検証部１３で比較、検証される。そして、ファームウェア３の検証の際には、このＨＤＬモデル２の検証に用いられる期待値算出モデル１２がインタフェース部１５を介して利用され、ファームウェア３がソフトウェアデバッガ１４により検証される。すなわち、期待値算出モデル１２は、ハードウェア検証の際には期待値の生成モデルとして用いられ、ソフトウェア検証の際にはハードウェアのＣモデルとして用いられる。この期待値算出モデル１２は、検証時の検証レベルを示す複数のタイミング情報を有し、ハードウェアまたはソフトウェアの検証の際には、そのタイミング情報を基に、検証レベルに応じたタイミングでデータを転送する。このように期待値算出モデル１２をハードウェア検証とソフトウェア検証の双方に用いることにより、精度良く効率的に検証が行えるようになる。
【００１９】
また、本発明では、ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証方法において、検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする検証方法が提供される。
【００２０】
このような検証方法によれば、検証時の検証レベルを示す複数のタイミング情報を有するモデルが、ハードウェア検証とソフトウェア検証の双方に用いられ、精度良く効率的に検証が行えるようになる。
【００２１】
また、本発明では、コンピュータを用いて、ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証するためのプログラムにおいて、コンピュータに、検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを用いてハードウェア検証を行い、前記モデルを用いてソフトウェア検証を行う処理を実行させることを特徴とするプログラムが提供される。
【００２２】
このようなプログラムによれば、検証時の検証レベルを示す複数のタイミング情報を有する期待値算出モデルがハードウェア検証に用いられ、さらにソフトウェア検証にも用いられるようになるので、精度良く効率的に検証が行えるようになる。
【００２３】
また、本発明では、ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、前記モデルは、前記ソフトウェア検証に使用されるタイミング情報以外のタイミング情報を有することを特徴とする検証装置が提供される。
【００２４】
このような検証装置によれば、複数のタイミング情報を有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いて、精度良く効率的に検証が行える。
また、本発明では、ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、前記モデルは、前記ソフトウェア検証に使用されるタイミング情報について、複数種別のソフトウェア検証に対応する数有することを特徴とする検証装置が提供される。

【００２５】
このような検証装置によれば、複数のタイミング情報を有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いて、精度良く効率的に検証が行える。
【発明の効果】
【００２６】
本発明によれば、検証時の検証レベルを示す複数のタイミング情報を有するモデルを用いてハードウェア検証とソフトウェア検証が行われる。タイミング情報を基に、ひとつのモデルで異なる検証レベルの複数の検証に対応することができるので、ソフトウェア検証用にハードウェアモデル（Ｃモデル）を作成することが不要になり、また、不具合の原因を効率良く見つけ出すことが可能になる。これにより、システムＬＳＩの開発効率を向上させ、高品質なシステムＬＳＩを短期間でかつ低コストで開発・製造することが可能になる。
【発明を実施するための最良の形態】
【００２７】
以下、本発明の実施の形態を図面を参照して詳細に説明する。
図１は本発明の原理説明図である。
システムＬＳＩ開発では、まず、ＬＳＩ仕様（ＩＰ（Intellectual Property）仕様）１が設計され、アーキテクチャの設計を行ってハードウェアとそのハードウェアを制御するためのソフトウェアが切り分けられ、ハードウェア、ソフトウェアそれぞれの詳細設計が行われる。
【００２８】
ハードウェアは、例えば、システムＬＳＩに搭載されるＣＰＵ／ＭＰＵの周辺回路等である。ハードウェアについては、その論理機能がＶＨＤＬ（Very High Speed Integrated Circuit HDL）やＶｅｒｉｌｏｇ－ＨＤＬ等のＨＤＬを用いて記述されたハードウェアモデル（ＨＤＬモデル）２が設計される。
【００２９】
また、ソフトウェアは、ＣＰＵ／ＭＰＵやその周辺回路等に所定の処理動作を実行させるためのプログラムからなるファームウェア３である。ソフトウェアについては、Ｃ／Ｃ＋＋等のプログラミング言語を用いて、システムＬＳＩに搭載されるファームウェア３が設計される。
【００３０】
本発明の検証装置１０は、ＨＤＬモデル２およびファームウェア３を有するとともに、ＨＤＬモデル２の機能を検証するための論理シミュレータ１１、および論理シミュレータ１１のシミュレーション結果と期待値算出モデル１２によって生成される期待値との等価性を検証する等価検証部１３を有している。さらに、この検証装置１０は、ファームウェア３の処理動作を検証するためのソフトウェアデバッガ１４、およびこのソフトウェアデバッガ１４と期待値算出モデル１２とを接続するインタフェース部１５を有している。
【００３１】
このような検証装置１０において、論理シミュレータ１１は、ＬＳＩ仕様１に基づく所定のテストパターンを用いてＨＤＬモデル２をシミュレートし、シミュレーション結果を生成する。
【００３２】
期待値算出モデル１２は、ＬＳＩ仕様１に基づくハードウェアの機能仕様に従ってＣ／Ｃ＋＋等のプログラミング言語を用いて作成されたハードウェアモデルを含んでおり、このハードウェアモデルは、ＨＤＬモデル２の検証における期待値の生成モデルとして用いられる。さらに、このハードウェアモデルは、ファームウェア３の検証においては、システムＬＳＩのハードウェアを記述したＣモデルとして用いられる。
【００３３】
等価検証部１３は、ＨＤＬモデル２のシミュレーション結果と、所定のテストパターンを用いて期待値算出モデル１２から生成される期待値とを比較し、その等価性を検証する

。なお、シミュレーション結果が期待値と等価であるＨＤＬモデル２は、ソフトウェア検証、あるいは論理合成、レイアウト設計、タイミング検証、テープアウト等の各工程で利用される。

【００３４】

ソフトウェアデバッガ１４は、システムＬＳＩに搭載されるＣＰＵ／ＭＰＵの命令動作をシミュレートするＩＳＳであり、ソフトウェア検証の際には、このＩＳＳがＣＰＵ／ＭＰＵに代わってファームウェア３を実行する。

【００３５】

インタフェース部１５は、ソフトウェアデバッガ１４とのインタフェースを有し、さらに、ＡＰＩ（Application Program Interface）を用いた期待値算出モデル１２とのインタフェースを有する。これにより、ソフトウェアデバッガ１４は、このインタフェース部１５を介して、ハードウェアのＣモデルとして期待値算出モデル１２を利用し、ファームウェア３の検証を行うことができるようになっている。

【００３６】

このように、期待値算出モデル１２は、ハードウェア検証においては、システムＬＳＩのＣＰＵ／ＭＰＵの周辺回路等を示したＨＤＬモデル２の期待値生成モデルとして用いられ、ソフトウェア検証においては、ＣＰＵ／ＭＰＵの周辺回路等を示したＣモデルとして用いられるようになっている。

【００３７】

このようなハードウェア検証およびソフトウェア検証を行う検証装置１０は、例えば、一般的なパーソナルコンピュータやワークステーションを用いて構成することができる。その場合、ＨＤＬモデル２、ファームウェア３、論理シミュレータ１１、期待値算出モデル１２、等価検証部１３、ソフトウェアデバッガ１４、インタフェース部１５およびテストパターン等は、プログラムとして構成され、検証装置１０が有するハードディスク等の記憶装置あるいは検証装置１０内のメモリに格納される。検証装置１０では、これらのプログラムが検証装置１０のＣＰＵ等によって実行されることで、所定の処理機能が実現され、ハードウェア検証およびソフトウェア検証が行えるようになっている。

【００３８】

図２はハードウェアおよびソフトウェアの開発・検証環境の概念図である。

ハードウェア開発・検証環境は、期待値算出モデル１２、ＲＴＬモデル２ａ、テストパターン１６、論理シミュレータ１１、等価検証部１３で構成される。

【００３９】

ここで、ＲＴＬモデル２ａは、ＬＳＩ仕様（ＩＰ仕様）１に基づき、システムＬＳＩのハードウェアの論理機能がＨＤＬを用いてＲＴＬ記述されたハードウェアモデルである。また、テストパターン１６は、このシステムＬＳＩが使われるアプリケーションで想定される命令や命令の組み合わせを示したテストベンチである。

【００４０】

ソフトウェア開発・検証環境は、図２に示すように、期待値算出モデル１２のほか、期待値算出モデル１２にバス１７およびバスインタフェース１８を介して接続されるＣＰＵ／ＭＰＵの動作を記述したＣＰＵ／ＭＰＵブロック１９、およびファームウェア３が格納されるメモリブロック２０を有する仮想プロトタイプとして構成することができる。

【００４１】

ここで、バスインタフェース１８は、ＵＳＢ（Universal Serial Bus）、ＰＣＩ（Peripheral Component Interconnect）、ＩＥＥＥ（Institute of Electrical and Electronics Engineers）１３９４などの規格に従い、ＬＳＩ仕様１に合わせてＣ言語で記述され、ＣＰＵ／ＭＰＵブロック１９をバス１７に接続する。

【００４２】

ＣＰＵ／ＭＰＵブロック１９としては、ソフトウェアデバッガ１４であるＩＳＳが用いられる。ＩＳＳは、ＣＰＵ／ＭＰＵに代わってメモリブロック２０に格納されたＣ言語のファームウェア３を実行し、ＣＰＵ／ＭＰＵの命令動作をシミュレートするようになって

いる。

【００４３】

ＩＳＳは、ファンクション・アキュレート・レベル（Function Accurate Level）、トランザクション・アキュレート・レベル（Transaction Accurate Level）、サイクル・アキュレート・レベル（Cycle Accurate Level）のいずれかの検証レベルでＣＰＵ／ＭＰＵの命令動作をシミュレートする。

【００４４】

ここで、ファンクション・アキュレート・レベルとは、時間概念のない、機能単位でデータを転送して高速にシミュレーションが実行される検証レベルをいう。トランザクション・アキュレート・レベルとは、バスアクセス等のイベント単位でデータを転送してシミュレーションが実行される検証レベルをいう。サイクル・アキュレート・レベルとは、クロック単位でデータを転送してシミュレーションが実行される検証レベルをいう。

【００４５】

ソフトウェア検証の際には、検証精度を重視するか、検証スピードを重視するかといった検証目的に応じて、用いるＩＳＳが選択され、選択されるＩＳＳによってソフトウェア検証の検証レベルが決まる。

【００４６】

メモリブロック２０は、システムＬＳＩにおいてＲＡＭあるいはＲＯＭとして機能し、ＣＰＵ／ＭＰＵやその周辺回路を動作させるファームウェア３が格納される。

期待値算出モデル１２は、後述のバスインタフェースおよびバス１７を介して、ＣＰＵ／ＭＰＵブロック１９と接続されるようになっている。

【００４７】

また、期待値算出モデル１２は、ファンクション・アキュレート・レベル、トランザクション・アキュレート・レベル、サイクル・アキュレート・レベルをそれぞれ示すタイミング情報を有する。このタイミング情報は、期待値算出モデル１２において、例えばＩＳＳに応じ、ユーザが検証レベルを設定することによって決定される。期待値算出モデル１２では、このタイミング情報によってシミュレーションの検証レベルが認識され、それによってシミュレーション時のデータ転送のタイミング（機能単位、イベント単位、クロック単位）が制御されるようになっている。

【００４８】

このような構成のハードウェアおよびソフトウェアの開発・検証環境において、ハードウェア検証の際には、ＨＤＬを用いて記述されたＲＴＬモデル２ａとテストパターン１６を用い、論理シミュレータ１１でＲＴＬモデル２ａの出力をシミュレートする。さらに、この論理シミュレータ１１では、Ｃ言語で記述された期待値算出モデル１２とテストパターン１６を用い、期待値算出モデル１２の出力がシミュレートされ、期待値が生成される。その際、期待値は、ファンクションレベルの演算によって生成される。

【００４９】

この期待値算出モデル１２は、テストパターン１６を用いたＲＴＬモデル２ａのシミュレーションにおける論理シミュレータ１１のコントロール情報に基づいて演算を行うようにすることで、ＲＴＬモデル２ａと動作の同期が行える。このように、期待値算出モデル１２を、ＲＴＬモデル２ａと同期させて同じ動作をするかどうかを比較することで、ハードウェアの不具合を早い段階で見つけ出すことが可能になる。

【００５０】

また、期待値算出モデル１２は、ＬＳＩ仕様１に基づくハードウェアの機能仕様に従って作成されているので、ＲＴＬモデル２ａとの動作比較基準として好適である。すなわち、従来のハードウェア検証では、ＲＴＬモデル２ａが適正に動作しているか否かの判定基準がなかったために、ＲＴＬモデル２ａが仕様通りに動作しているか否かの見極めが難しかった。しかしながら、期待値算出モデル１２をハードウェアの機能仕様に従って作成することにより、期待値算出モデル１２をＲＴＬモデル２ａの動作が適正か否かの判断基準として利用することができる。

【００５１】
ＲＴＬモデル２ａのシミュレーション結果と期待値算出モデル１２の期待値とは、等価検証部１３でその等価性が検証される。シミュレーション結果と期待値とが一致する場合には、ソフトウェア検証や、そのＲＴＬモデル２ａを用いた論理合成等、所定の次工程に移行する。シミュレーション結果と期待値とが不一致の場合には、ＲＴＬモデル２ａおよび期待値算出モデル１２を再検証し、不具合の生じる原因を究明することとなる。
【００５２】
このようにして期待値算出モデル１２の出力とＲＴＬモデル２ａの出力とが等価となった後に、用いるＩＳＳによって決まる検証レベルでのソフトウェア検証が行われる。ソフトウェア検証の際には、ファームウェア３が期待値算出モデル１２のハードウェアモデル（Ｃモデル）と共にＩＳＳによって正しく動作するかが検証される。
【００５３】
ここで、上記のようにハードウェア検証およびソフトウェア検証で用いられる期待値算出モデル１２について、より詳細に説明する。
図３は期待値算出モデルの構成例を示す図である。
【００５４】
期待値算出モデル１２は、バス１７に接続されるＵＳＢ、ＰＣＩ、ＩＥＥＥ１３９４などの規格に従ったバスインタフェース１２ａ、バスインタフェース１２ａから転送される入力データを用いて所定の演算を実行するハードウェアモデル１２ｂ、およびハードウェアモデル１２ｂの演算結果が格納されるスタックメモリ１２ｃを有している。
【００５５】
バスインタフェース１２ａは、ファンクション・アキュレート・レベル、トランザクション・アキュレート・レベル、サイクル・アキュレート・レベルの各検証レベルに対応したタイミング情報に応じて、バス１７から入力されてくるデータのハードウェアモデル１２ｂへの転送方式を切り替えることができるように構成されている。
【００５６】
すなわち、バスインタフェース１２ａは、ファンクション・アキュレート・レベルでは、ハードウェアモデル１２ｂが演算に必要とする入力データのすべてを転送する。トランザクション・アキュレート・レベルでは、ハードウェアモデル１２ｂにバスアクセス等のイベントごとに入力データを転送する。サイクル・アキュレート・レベルでは、ハードウェアモデル１２ｂに１クロックごとに入力データを転送する。
【００５７】
また、バスインタフェース１２ａは、タイミング情報に応じて、スタックメモリ１２ｃからバス１７への出力データの転送方式を切り替えることができるように構成されている。
【００５８】
すなわち、バスインタフェース１２ａは、ファンクション・アキュレート・レベルでは、スタックメモリ１２ｃに格納されているデータをすべてバス１７に転送する。トランザクション・アキュレート・レベルでは、スタックメモリ１２ｃに格納されているデータをイベントごとにバス１７に転送する。サイクル・アキュレート・レベルでは、スタックメモリ１２ｃに格納されているデータを１クロックごとにバス１７に転送する。
【００５９】
期待値算出モデル１２にこのようなタイミング情報に基づくデータ転送方式の切り替え機能を備えることにより、ＲＴＬモデル２ａの検証のほか、ＩＳＳの種類に依らずにファームウェア３の検証を行うことが可能になっている。また、別のソフトウェア検証装置を用いる場合であっても、その種類に依らずにファームウェア３の検証が可能になる。
【００６０】
また、ハードウェアモデル１２ｂは、先のハードウェア検証においてＲＴＬモデル２ａと等価なファンクションレベルのモデルであり、例えばＡＮＳＩ規格のＣ言語等を用いて作成される。ハードウェアモデル１２ｂでは、ＲＴＬモデル２ａの検証あるいはファーム

ウェア３の検証の際には、バスインタフェース１２ａから入力データが転送され、所定の演算処理に必要な入力データがすべて揃ったときに、その演算が実行され、演算結果のデータがスタックメモリ１２ｃに転送される。スタックメモリ１２ｃに転送される演算結果のデータが複数のデータの集合からなる場合には、そのデータの集合がスタックメモリ１２ｃに一気に転送される。

【００６１】

ここで、ハードウェアモデル１２ｂがＡＮＳＩ規格に従うＣ言語で作成されている場合には、ハードウェア検証の際、テストパターンやシミュレーションに用いられる４値（０，１，Ｘ，Ｚ）のうちＸ（不定）およびＺ（ハイインピーダンス）の値はそのままでは演算に用いることができない。そのため、このバスインタフェース１２ａは、ユーザ定義により、４値表現の入力データを０，１の２値表現に変換してハードウェアモデル１２ｂに転送するように構成されている。例えば、バスインタフェース１２ａは、不定を示すＸを０または１に、ハイインピーダンスを示すＺを１または０に、それぞれ変換してハードウェアモデル１２ｂに転送する。

【００６２】

ハードウェアモデル１２ｂでは、０，１の２値で演算が実行され、その演算結果のデータがスタックメモリ１２ｃに転送される。

また、その場合、バスインタフェース１２ａは、バス１７に転送するデータが、入力データ転送時のデータ幅やビット幅に満たない場合には、４値表現から２値表現に変換されたデータであると認識し、そのデータを、ユーザ定義により、０または１をＸまたはＺに再変換し、４値表現でデータをバス１７に出力する。

【００６３】

期待値算出モデル１２に、このような４値／２値変換機能を備えることで、ＡＮＳＩ規格に従うＣ言語で作成されハードウェアモデル１２ｂを、ＲＴＬモデル２ａの検証およびファームウェア３の検証の双方に容易に用いることが可能になる。

【００６４】

また、検証装置１０は、ハードウェア検証で使用する４値表現のモデルを、ソフトウェア検証に使用する２値表現に変換する４値／２値変換手段を有している。

なお、期待値算出モデル１２は、その入力データに０，１，Ｘ，Ｚの４値以外の値（Ｕ，Ｗ，Ｌ，Ｈなど）が含まれている場合でも、同様にして最終的に０，１の２値に変換し、ハードウェアモデル１２ｂによる演算を実行した後、再変換して出力するように構成することも可能である。

【００６５】

また、この期待値算出モデル１２においては、演算結果のデータまたはデータの集合がスタックメモリ１２ｃに格納される際、その演算結果のデータまたはデータの集合にフラグが付加されるようになっている。

【００６６】

図４はスタックメモリのデータ格納例を示す図である。

期待値算出モデル１２のスタックメモリ１２ｃには、ハードウェアモデル１２ｂによる演算結果のデータが一気に転送され、各アドレスに格納される。期待値算出モデル１２では、このように各アドレスに格納されるデータについて、検証レベルに応じて転送すべきデータ単位に、それぞれ目安となるフラグが定義されている。

【００６７】

図４に示したデータ格納例の場合、アドレス０ｘ０～０ｘＦまでのアドレスに格納されるデータについて、検証レベルがファンクション・アキュレート・レベルのときには、アドレス０ｘ０～０ｘ７，０ｘ８～０ｘＦのデータの集合にそれぞれ０，１のフラグが付加されている。バスインタフェース１２ａでは、バス１７に転送するデータ単位および転送順がこのフラグによって識別され、バス１７へのデータ転送が制御される。

【００６８】

すなわち、バスインタフェース１２ａは、検証レベルがファンクション・アキュレート

・レベルのときには、スタックメモリ１２ｃからのデータ転送時に、まず、０のフラグが付加されているアドレス０ｘ０～０ｘ７のデータの集合を転送し、次いで、１のフラグが付加されているアドレス０ｘ８～０ｘＦのデータの集合を転送する。

【００６９】

また、検証レベルがトランザクション・アキュレート・レベルのときには、０ｘ０～０ｘ３，０ｘ４～０ｘ７，０ｘ８～０ｘＢ，０ｘＣ～０ｘＦのデータの集合にそれぞれ０～３のフラグが付加されている。さらにまた、検証レベルがサイクル・アキュレート・レベルのときには、０ｘ０～０ｘＦの１６個の各データに０～１５のフラグがそれぞれ付加されている。トランザクション・アキュレート・レベル、サイクル・アキュレート・レベルのときもファンクション・アキュレート・レベルのときと同様、バスインタフェース１２ａでは、バス１７に転送するデータ単位および転送順がこのフラグによって識別され、バス１７へのデータ転送が制御される。

【００７０】

このように、検証レベルに応じて、転送するデータ単位にフラグを付加して格納しておくことにより、検証レベルに応じたバス１７へのデータ転送が可能になる。したがって、この期待値算出モデル１２では、ソフトウェア検証がファンクション・アキュレート・レベル、トランザクション・アキュレート・レベル、サイクル・アキュレート・レベルのいずれであっても、その検証レベルに応じてバス１７にデータまたはデータの集合を転送することができる。

【００７１】

従来はソフトウェア検証がファンクション・アキュレート・レベルかトランザクション・アキュレート・レベルかサイクル・アキュレート・レベルかに依って、すなわちソフトウェア検証に用いるＩＳＳに依って、別々にハードウェアモデル（Ｃモデル）を作成し、別々にシミュレーションを行う必要があった。これに対し、上記期待値算出モデル１２によれば、ひとつのハードウェアモデル（Ｃモデル）で複数の検証レベルあるいはソフトウェア検証装置に対応することができるので、より低コストで効率的にシミュレーションを行うことが可能になる。

【００７２】

以上説明したように、上記検証装置１０によれば、ハードウェア検証において作成された期待値算出モデル１２を、ソフトウェア検証に再利用するので、従来のようにソフトウェア検証用にハードウェアモデル（Ｃモデル）を作成することが不要になる。そのため、期待値算出モデル１２が、ＲＴＬモデル２ａと等価である限りは、ソフトウェア検証で生じる不具合の原因はファームウェア３にあると考えることができ、ソフトウェア検証の際にハードウェアモデルの検証を行う必要がなくなる。これにより、システムＬＳＩの開発効率を向上させ、高品質なシステムＬＳＩを短期間でかつ低コストで開発・製造することが可能になる。

【００７３】

また、期待値算出モデル１２をソフトウェア検証に再利用するに当たり、期待値算出モデル１２をソフトウェア検証の検証レベルに対応したタイミング情報に従ってデータの転送方式を切り替えるようにしたので、検証レベルの異なる複数種のＩＳＳに対応することができる。すなわち、システムＬＳＩに搭載するＣＰＵ等が別のものに替わっても、期待値算出モデル１２は特別な変更を加えることなくＩＰとして以後も利用していくことができ、非常に汎用性が高い。

【００７４】

また、期待値算出モデル１２をハードウェア検証とソフトウェア検証の双方に利用できるようにするため、ハードウェアモデル１２ｂをＡＮＳＩ規格のＣ言語を用いて作成し、期待値算出モデル１２に４値／２値変換機能を備えたことにより、期待値算出モデル１２の汎用性がいっそう高まるとともに、ソフトウェア検証への再利用が容易に行えるようになる。

【００７５】

なお、検証装置１０は、ハードウェア検証の際にはタイミング情報を有していない期待値算出モデル１２を用い、ソフトウェア検証の際に期待値算出モデル１２に複数のタイミング情報を付加し、これを用いるようにすることができる。あるいは、あらかじめ複数のタイミング情報を有する期待値算出モデル１２を、ハードウェア検証とソフトウェア検証の双方に用いるようにしてもよい。

【００７６】

なお、上記の処理機能は、コンピュータによって実現することができる。その場合、検証装置１０が有すべき機能の処理内容を記述したプログラムが提供される。そのプログラムをコンピュータで実行することにより、上記処理機能がコンピュータ上で実現される。処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリなどがある。磁気記録装置には、ハードディスク装置（ＨＤＤ）、フレキシブルディスク（ＦＤ）、磁気テープなどがある。光ディスクには、ＤＶＤ(Digital Versatile Disc)、ＤＶＤ－ＲＡＭ(Random Access Memory)、ＣＤ－ＲＯＭ(Compact Disc Read Only Memory)、ＣＤ－Ｒ(Recordable)／ＲＷ(ReWritable)などがある。光磁気記録媒体には、ＭＯ(Magneto-Optical disk)などがある。

【００７７】

プログラムを流通させる場合には、例えば、そのプログラムが記録されたＤＶＤ、ＣＤ－ＲＯＭなどの可搬型記録媒体が販売される。また、プログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することもできる。

【００７８】

プログラムを実行するコンピュータは、例えば、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、自己の記憶装置に格納する。そして、コンピュータは、自己の記憶装置からプログラムを読み取り、プログラムに従った処理を実行する。なお、コンピュータは、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータは、サーバコンピュータからプログラムが転送されるごとに、逐次、受け取ったプログラムに従った処理を実行することもできる。

【００７９】

（付記１）　ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証装置において、

検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする検証装置。

【００８０】

（付記２）　前記モデルは、複数の異なる検証レベルに対応する数のタイミング情報を有し、検証時の検証レベルに応じた前記タイミング情報に従って入出力データの転送方式を切り替えることを特徴とする付記１記載の検証装置。

【００８１】

（付記３）　前記モデルは、検証時の入出力データを変換するデータ変換手段を有し、検証時に入力されるデータを前記データ変換手段によって変換して演算を実行し、演算後のデータを前記データ変換手段によって再変換して出力することを特徴とする付記１記載の検証装置。

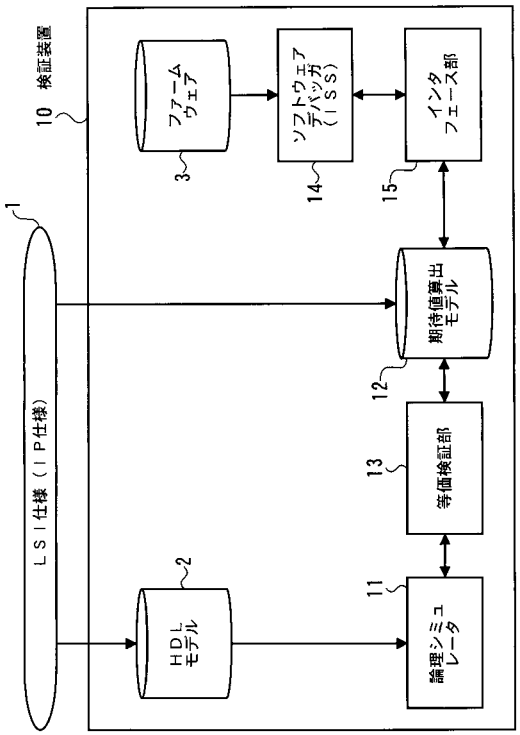
【００８２】

（付記４）　前記モデルは、検証時の検証レベルに応じたタイミング情報に従って入出力データを転送するインタフェースと、前記インタフェースから転送されるデータを用いて演算を実行するハードウェアモデルと、前記ハードウェアモデルによって演算され前記インタフェースによって出力されるデータを格納するスタックメモリと、を有することを特徴とする付記１記載の検証装置。

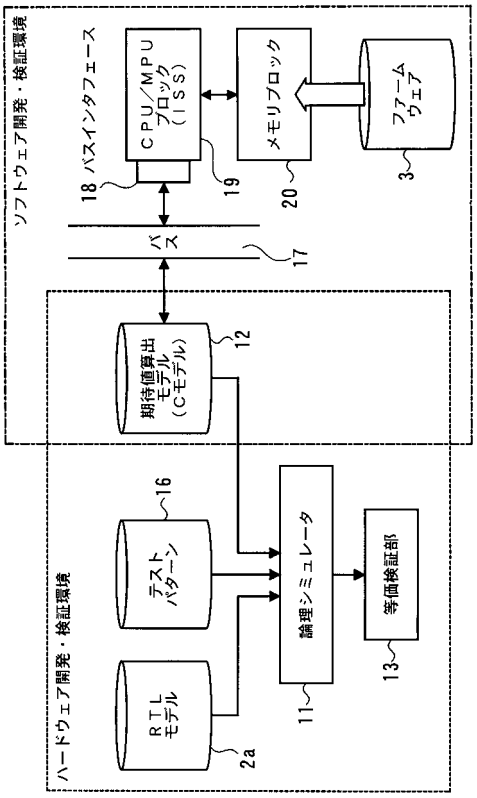
【００８３】
（付記５）　ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証する検証方法において、
検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを、ハードウェア検証とソフトウェア検証の双方に用いることを特徴とする検証方法。
【００８４】
（付記６）　前記モデルは、複数の異なる検証レベルに対応する数のタイミング情報を有し、検証時の検証レベルに応じた前記タイミング情報に従って入出力データの転送方式を切り替えることを特徴とする付記５記載の検証方法。
【００８５】
（付記７）　前記モデルは、検証時に入力されるデータを変換して演算を実行し、演算後のデータを再変換して出力することを特徴とする付記５記載の検証方法。
（付記８）　前記モデルは、入力されるデータを検証時の検証レベルに応じたタイミング情報に従ってインタフェースによってハードウェアモデルに転送し、前記インタフェースから転送されたデータを用いてハードウェアモデルによって演算を実行し、前記ハードウェアモデルによって演算されたデータをスタックメモリに格納し、前記スタックメモリに格納されたデータを前記検証時の検証レベルに応じたタイミング情報に従って出力することを特徴とする付記５記載の検証方法。
【００８６】
（付記９）　コンピュータを用いて、ハードウェアとハードウェアを制御するソフトウェアとをモデルを用いて検証するためのプログラムにおいて、
コンピュータに、
検証時の検証レベルを示す一のタイミング情報と前記一のタイミング情報と異なる他のタイミング情報とを有するモデルを用いてハードウェア検証を行い、
前記モデルを用いてソフトウェア検証を行う処理を実行させることを特徴とするプログラム。
【００８７】
（付記１０）　前記コンピュータに、
検証時の検証レベルに応じたタイミング情報に従って前記モデルの入出力データの転送方式を切り替える処理を実行させることを特徴とする付記９記載のプログラム。
【００８８】
（付記１１）　前記コンピュータに、
検証時に前記モデルに入力されるデータを変換して演算を実行し、演算後のデータを再変換して出力する処理を実行させることを特徴とする付記９記載のプログラム。
【００８９】
（付記１２）　前記コンピュータに、
前記モデルに入力されるデータを検証時の検証レベルに応じたタイミング情報に従ってインタフェースによってハードウェアモデルに転送し、前記インタフェースから転送されたデータを用いてハードウェアモデルによって演算を実行し、前記ハードウェアモデルによって演算されたデータをスタックメモリに格納し、前記スタックメモリに格納されたデータを前記検証時の検証レベルに応じたタイミング情報に従って出力する処理を実行させることを特徴とする付記９記載のプログラム。
【００９０】
（付記１３）　ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、
前記モデルは、前記ソフトウェア検証に使用されるタイミング情報以外のタイミング情報を有することを特徴とする検証装置。
【００９１】
（付記１４）　ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるよ

うに作成する検証装置において、
前記モデルは、前記ソフトウェア検証に使用されるタイミング情報について、複数種別のソフトウェア検証に対応する数有することを特徴とする検証装置。
【００９２】
（付記１５） ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、
前記モデルは、前記ソフトウェア検証に使用されるタイミング情報について、複数のソフトウェア検証装置に対応する数有することを特徴とする検証装置。
【００９３】
（付記１６） ハードウェア検証に使用されるモデルを、ソフトウェア検証を行えるように作成する検証装置において、
前記ハードウェア検証で使用する４値表現のモデルを、前記ソフトウェア検証に使用する２値表現に変換する４値／２値変換手段を有することを特徴とする検証装置。
【図面の簡単な説明】
【００９４】
【図１】本発明の原理説明図である。
【図２】ハードウェアおよびソフトウェアの開発・検証環境の概念図である。
【図３】期待値算出モデルの構成例を示す図である。
【図４】スタックメモリのデータ格納例を示す図である。
【図５】従来のハードウェアおよびソフトウェアの開発・検証環境の概念図である。
【符号の説明】
【００９５】
１ ＬＳＩ仕様
２ ＨＤＬモデル
２ａ ＲＴＬモデル
３ ファームウェア
１０ 検証装置
１１ 論理シミュレータ
１２ 期待値算出モデル
１２ａ，１８ バスインタフェース
１２ｂ ハードウェアモデル
１２ｃ スタックメモリ
１３ 等価検証部
１４ ソフトウェアデバッガ
１５ インタフェース部
１６ テストパターン
１７ バス
１９ ＣＰＵ／ＭＰＵブロック
２０ メモリブロック

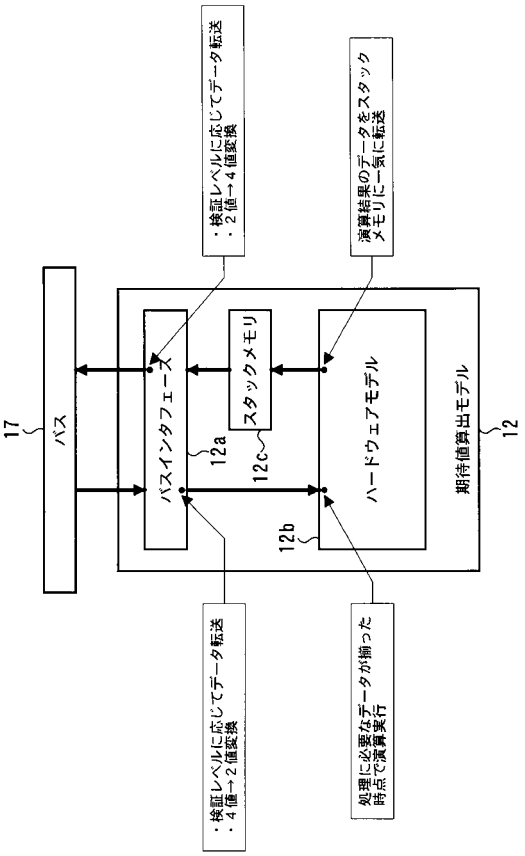
【図１】




【図２】



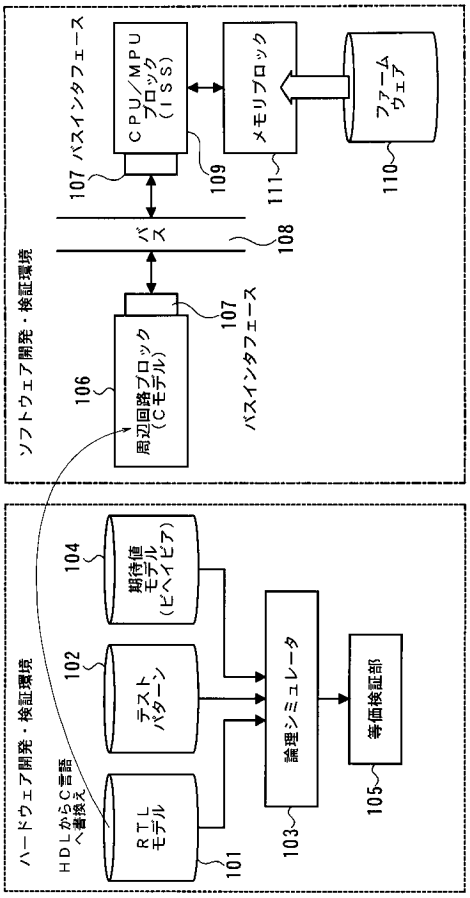

【図３】




【図４】

| フラグ | | | アドレス | データ |
|---|---|---|---|---|
| ファンクション・アキュレート・レベル | トランザクション・アキュレート・レベル | サイクル・アキュレート・レベル | | |
| 0 | 0 | 0 | 0x0 | 80 |
| | | 1 | 0x1 | bb |
| | | 2 | 0x2 | 7c |
| | | 3 | 0x3 | bc |
| | 1 | 4 | 0x4 | 80 |
| | | 5 | 0x5 | bd |
| | | 6 | 0x6 | 7c |
| | | 7 | 0x7 | be |
| 1 | 2 | 8 | 0x8 | 71 |
| | | 9 | 0x9 | 29 |
| | | 10 | 0xA | 83 |
| | | 11 | 0xB | 28 |
| | 3 | 12 | 0xC | 7a |
| | | 13 | 0xD | 32 |
| | | 14 | 0xE | 82 |
| | | 15 | 0xF | 3a |

【図５】


ソフトウェア開発・検証環境
107 バスインタフェース
CPU/MPU ブロック (ISS)
109
メモリブロック
111
ファームウェア
110
バス
108
107
バスインタフェース
106
周辺回路ブロック (Cモデル)
ハードウェア開発・検証環境
HDLからC言語へ書換え
104
期待値モデル (ビヘイビア)
102
テストパターン
RTL モデル
101
論理シミュレータ
103
等価検証部
105