(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2004-46865
(P2004-46865A)
(43)公開日　平成16年2月12日(2004.2.12)

(51)Int.Cl.[7]	F I	テーマコード（参考）
G06F 17/50	G06F 17/50 672R	5B046

審査請求 有　請求項の数 1　OL　（全 12 頁）

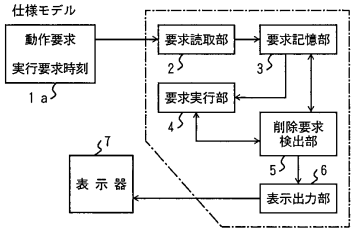
(21)出願番号　特願2003-193157(P2003-193157)
(22)出願日　平成15年7月7日(2003.7.7)
(62)分割の表示　特願平5-51709の分割
原出願日　平成5年3月12日(1993.3.12)

(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100058479
弁理士　鈴江　武彦
(74)代理人　100091351
弁理士　河野　哲
(74)代理人　100088683
弁理士　中村　誠
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100084618
弁理士　村松　貞男
(74)代理人　100092196
弁理士　橋本　良郎

最終頁に続く

(54)【発明の名称】ハードウェアシミュレーション装置

(57)【要約】
【課題】ハードウェアの動作を記述した仕様モデルのシミュレーション実行過程の各種情報を記憶して解析する手段を用いて、設計者のシミュレーション環境を向上させ、仕様モデルの静的な解析だけでは発見できないバグを取り除くための支援を行う。
【解決手段】仕様モデルに記述された各動作要求及び各実行要求時刻を読取る過程で、先に記憶されている各動作要求のうち各実行要求時刻相互の関係に起因して実行されずに削除される動作要求を検出し、この検出した削除動作要求を表示出力して、動作に反映されない信号代入要求を設計の初期段階で設計者に知らせるようにしている。
【選択図】　図１



【特許請求の範囲】
【請求項１】
ハードウェアの動作を記述した仕様モデルに記述された各信号代入要求を含む信号処理要求を読取る要求読取手段と、この要求読取手段にて読取られた各信号処理要求を記憶する要求記憶手段と、この要求記憶手段にて記憶された各信号処理要求を順次実行する要求実行手段とを有するハードウェアシミュレーション装置において、
前記要求実行手段にて実際に一つの信号に対する信号処理要求が実行される毎に、その時点までの該当信号の信号値の最大値及び最小値を検出する信号値検出手段と、
この信号値検出手段にて検出された最大値及び最小値をそれぞれ上限値及び下限値として記憶する信号限界値記憶手段と、
前記要求実行手段にて実際に前記信号に対する信号代入要求が実行される毎に、信号代入値と前記信号限界値記憶手段に記憶された上限値及び下限値各値と比較する比較手段と、
前記信号代入値が前記上限値より大きいとき、上限値を代入値に更新し、前記信号代入値が前記下限値より小さいとき、下限値を代入値に更新する信号限界値更新手段と、
前記信号限界値記憶手段に記憶された上限値及び下限値を表示する限界値表示手段と
を備えたハードウェアシミュレーション装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明はハードウェアの設計を支援するＣＡＤ（計算機支援設計）システムに係わり、特に、作成されたハードウェアの動作を示す仕様モデルの動作をシミュレーションするハードウェアシミュレーション装置に関する。
【０００２】
【従来の技術】
例えばＣＡＤ装置で作成された集積回路に組込まれる論理回路等のハードウェアの動作をシミュレーションするハードウェアシミュレーション装置は、このハードウェアの仕様モデルに記述にされた各動作要求を読取って順番に実行していく。
【０００３】
例えば、ハードウェアの仕様モデルが複数の動作単位（プロセス）で構成されていた場合には、第１の動作単位に与えられた信号を入力して、この動作単位の出力を変化さる。次に、この第１の動作単位の変化した出力信号を第２の動作単位へ入力させて、この第２の動作単位に含まれる各動作要求を実行する。そして、この第２の動作単位の出力値を得る。
【０００４】
そして、ハードウェアの設計者は、そのシミュレーション動作を途中で止めたり、内部状態を参照したり、内部状態を書換えたりしながらシミュレーションの時刻を進めていく。
【０００５】
【発明が解決しようとする課題】
しかしながら、上述したようなハードウェアシミュレーション装置においては、信号に設定した値の範囲について妥当性を知ることができないという技術的課題がある。すなわち、シミュレーション実行過程中に各信号が実際に取る値の範囲を記憶しておき、作成された仕様モデルにおいてその信号に与えられている値の範囲と比較する機能は提供されていない。したがって、ある信号において実際はある一定範囲の値しか取り得ないにも拘らず、仕様モデル上においては広い範囲の値を取るとして構築した場合、実際の動作中には現れない信号値等に対する処理回路も設計し、無駄な回路の設計を行なってしまう問題がある。
【０００６】
また、設計した仕様モデルが正常な動作をすることを複数のテストデータを使って、検証するのが普通である。この時、ある信号に注目した時、仕様モデルの検証が十分であるか否かの判断基準として、その信号の取り得る値の全範囲に亘って検証が終了したか否かを

確認する必要がある。
【０００７】
しかし、上述したシミュレーション装置においては、シミュレーション実行過程でその信号が実際にとった値の上限と下限等を記憶保持していないので、設計者は、膨大なテスト結果を処理して判断するなど、検証の十分性を判断するのが困難であった。
【０００８】
本発明はこのような事情に鑑みてなされたもので、シミュレーション実行過程における信号値や実行時刻等の種々の履歴を記憶保持することによって、設計者がハードウェアのシミュレーションを行なう上での回路のデバッグ作業効率の向上を図ることができるハードウェアシミュレーション装置を提供することを目的とする。
【０００９】
【課題を解決するための手段】
上記課題を解消するために本発明は、ハードウェアの動作を記述した仕様モデルに記述された各信号代入要求を含む信号処理要求を読取る要求読取手段と、要求読取手段にて読取られた各信号処理要求を記憶する要求記憶手段と、要求記憶手段にて記憶された各信号処理要求を順次実行する要求実行手段とを有するハードウェアシミュレーション装置において、要求実行手段にて実際に一つの信号に対する信号処理要求が実行される毎に、その時点までの該当信号の信号値の最大値及び最小値を検出する信号値検出手段と、信号値検出手段にて検出された最大値及び最小値をそれぞれ上限値及び下限値として記憶する信号限界値記憶手段と、要求実行手段にて実際に信号に対する信号代入要求が実行される毎に、信号代入値と信号限界値記憶手段に記憶された上限値及び下限値各値と比較する比較手段と、信号代入値が上限値より大きいとき、上限値を代入値に更新し、信号代入値が前記下限値より小さいとき、下限値を代入値に更新する信号限界値更新手段と、信号限界値記憶手段に記憶された上限値及び下限値を表示する限界値表示手段と備えている。
【００１０】
【作用】
このように構成されたハードウェアシミュレーション装置においては、実際に一つの信号に対する信号処理要求が実行される毎に、その時点までの該当信号の信号値の最大値及び最小値が検出されて、上限値及び下限値として記憶される。そして、この信号に対する代入値が上限値及び下限値を越えると、上限値又は下限値がこの代入値で更新される。設計者による信号の上限値，下限値の提示要求にて、これらの値が表示される。
【００１１】
これにより、設計者は、作成された仕様モデルが実際に流れるデータの大きさに較べてオーバースペックであるとか、ある信号について、全ての取り得る値を考慮したような検証がまだ終了していない等を簡単に確認できる。
【００１２】
【発明の実施の形態】
以下、図面を参照しながら本発明の実施形態を説明する。
【００１３】
図１は第１の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図である。なお、このハードウェアシミュレーション装置は例えばコンピュータ等の情報処理装置で構成されている。
【００１４】
例えばＣＡＤ装置にて作成されたハードウェアの動作を記述した仕様モデル１ａには、図２に示すように、多数の動作要求と各動作要求を実行するタイミングを示す実行要求時刻とが記述されている。
【００１５】
この仕様モデル１ａに記載されている動作要求及び実行要求時刻は要求読取部２で読取られたのち要求記憶部３へ一旦登録される。なお、この要求記憶部３は、要求読取部２で読取られた各動作要求を順次登録する過程で、先に登録されている各動作要求のうち各実行

要求時刻相互の関係に起因して実行できない動作要求が発生すれば、この動作要求を削除する。要求実行部４は要求記憶部３に記憶さている各動作要求をこの動作要求に指定された実行要求時刻に達した時点で実際に実行する。

【００１６】

削除要求検出部５は要求記憶部３にて削除される動作要求を検出する。削除要求検出部５で検出された削除動作要求は表示出力部６にてＣＲＴ表示装置等で構成された表示器７に表示出力する。

【００１７】

このように構成されたハードウェアシミュレーション装置の具体的動作を図２に示すハードウェアの動作を記述した仕様モデル１ａがシミュレーションされる場合を例にとって説明する。

【００１８】

図２はＥ１というモジュールの動作内容をＢ１の動作記述で書いている仕様モデルである。すなわち、この仕様モデル１ａの動作は２つの動作単位（プロセス）Ｐ０，Ｐ１で構成されている。第１の動作単位Ｐ０は、７行目の動作要求に記述されているように、信号ＣＬＫを３ｎｓ毎に反転させる。第２の動作単位Ｐ０は、８行乃至１１行目の各動作要求に記述されているように、信号ＣＬＫが変化すると、信号Ａに１ｎｓ後に５を代入し、６ｎｓ後に１０を代入する信号代入要求を発生させる。

【００１９】

このような記述によって表される仕様モデル１ａにおいては、第１の動作単位Ｐ０でもってＣＬＫ信号値に変化が起こる（８行目）毎に、第２の動作単位Ｐ１の処理が開始され、この動作単位Ｐ１の中に記述されている２つの信号代入要求（１０，１１行目）が予約登録される。

【００２０】

しかし、２番目の信号代入（１１行目）が実行されるより前に、必ず、信号ＣＬＫの値に変化が起こり（７行目）、動作単位Ｐ１が実行され、次の信号代入　（１０行目）が予約される。この時、残っていた２番目の信号代入予約（１１行目）の実行予定時刻よりも、早い時刻での代入の予約が行なわれるため、この残っていた信号代入予約は削除（キャンセル）される。

【００２１】

つまり、図２に示すハードウェアの仕様モデル１ａの場合、動作単位Ｐ１に記述されている２番目の信号代入要求（１１行目）は全く実施されない。このことは、信号代入に対する遅延時間とセンシティビティリストの信号の変化のタイミングにバグがあることを示す。このような動作要求の記述は、文法上のエラーではないためそのような回路を表す記述を静的に解析しただけではこのバグは発見できず、シミュレーションを実行していく中でしか発見できない。

【００２２】

しかし、図１に示す実施例装置においては、信号代入要求が削除（キャンセル）される時には削除要求検出部５が、その信号代入要求削除を検出して、表示出力部６が表示器７に表示出力する。これによって記述の該当部分にタイミングのバグが含まれている可能性があると設計者は知ることができ、シミュレーションの初期の段階でことようなバグを除去することが可能である。

【００２３】

なお、信号代入等の各動作要求を特定するために、各動作要求名対して、対象となる信号とその信号が属するモジュール、それは仕様モデル１ａの記述のどこに相当するか（図面のファイルならそのファイル名と該当する信号の描かれている位置、ハードウェア記述言語で書かれているならそのファイル名と代入文の行番号など）などの情報などを持たせる手法が必要である。

【００２４】

例えば、例では２番目の代入予約に対してモジュールＥ１の１１行目であり、モジュール

Ｅ１の内部信号Ａに対するものであると、信号代入要求に関する情報の要求記憶部３に記憶しておくことで、設計者はこれらの記述の該当部分に注目して設計の修正を行なえる。

【００２５】

図３は第２の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図である。なお、図１に示した実施例装置と同一部分には同一符号が付してある。したがって、重複する部分の詳細説明は省略する。

【００２６】

この実施例装置においては、仕様モデル１ｂに記述された信号代入要求等の信号処理要求及び実行要求時刻は要求読取部２で読取られて一旦要求記憶部３へ登録される。その後、要求実行部４で実際に実行される。そして、要求表示部８は、図４に示すように、要求記憶部３に登録された各信号処理要求を該当実行要求時刻が到来したタイミングで表示器７に表示する。また、信号表示部９は、要求実行部４にて実際に実行された信号処理要求後における信号値を先の信号処理要求と同一画面に表示する。

【００２７】

このように、図４に示すように、仕様モデル１ｂに記述された各信号処理要求と実際に実行された後の信号状態とを表示器７の同一画面に表示することによって、設計者は信号処理要求があったが、実際にはその信号処理要求が実行されなかったことを簡単に把握できる。

【００２８】

なお、信号値を図４に示すように波形で表示すると非常に理解しやすい。また、図示するように、削除された信号処理要求に印をつけて表示することによって、各信号処理要求の実行・不実行がより明確になる。さらに、信号処理要求の発生した順番が容易に理解できるように番号をつけて表示することも可能である。また、この場合、実際に実行された信号処理要求後における信号値を表示させなくとも、その値を知ることができる。

【００２９】

図５は第３の実施例に係わるハードウェアシミュレーション装置の構成を示すブロック図である。なお、図１に示した実施例装置と同一部分には同一符号が付してある。したがって、重複する部分の詳細説明は省略する。

【００３０】

この実施例装置においては、仕様モデル１ｃに記述された信号代入要求等の動作要求及び実行要求時刻は要求読取部２で読取られて一旦要求記憶部３へ登録される。その後、要求実行部４で実際に実行される。最終実行時刻検出手段としての信号時刻記憶部１０ａは、要求実行部４において実際に信号代入要求等の各動作単位に含まれる各動作要求が実行された時の実行時刻を記憶する。判定部１１は、信号時刻記憶部１０ａに記憶された各実行時刻が、該当動作単位に指定された時刻範囲に入るか否かか判定する。動作要求表示部１２は、該当動作単位中の指定された時刻範囲に入る動作要求を表示器７に表示する。

【００３１】

このように構成されたハードウェアシミュレーション装置の具体的動作を図６に示すハードウェアの動作を記述した仕様モデル１ｃがシミュレーションされる場合を例にとって説明する。

【００３２】

図６に示す仕様モデル１ｃにおけるＢ１に動作が記述されているモジュールＥ１は、２つの動作単位Ｐ０，Ｐ１で構成されている。第１の動作単位Ｐ１は、第１９行目に示すように、信号ＣＡＳＥＮＵＭを３ｎｓ毎に１ずつ増加させる。第２の動作単位Ｐ０は、信号ＣＡＳＥＮＵＭが変化すると（７行目）、１０行目乃至１６行目に示すように、信号ＣＡＳＥＮＵＭが０，１，…．５と変化する毎に、信号Ａにそれぞれの遅延時間６０ｎｓ～１０ｎｓが経過した時点で、０，１，…．５，の値を代入する。

【００３３】

このような仕様モデル１ｃにおいては、信号ＣＡＳＥＮＵＭの信号値に変化が起こるたびに動作単位Ｐ０の処理が開始され、この動作単位Ｐ０の中に記述されている信号代入が要

求されるが、実際に１番目の信号代入（１０行目）が実行される開始時刻＋６０ｎｓより前に、開始時刻＋３ｎｓに再び信号ＣＡＳＥＮＵＭの値が変化する。その結果、２番目（１１行目）の信号代入要求（開始時刻＋５３ｎｓ）が発生する。この時２番目の信号代入予約は、残っていた１番目の信号代入予約の実行予定時刻よりも、早い時刻での代入の予約であるため、残っていた１番目（１０行目）の信号代入予約は削除（キャンセル）される。

【００３４】

つまり、図６のように記述された仕様モデル１ｃの場合、動作単位Ｐ０に書かれている１番目の代入要求は全く実行されないことになる。同様に２番目から６番目までの各信号代入要求も実行されない。すなわち、シミュレーション開始時刻から６３ｎｓ経過するまでは信号Ａは変化しないことになる。このような記述は、文法上のエラーではないので、このハードウェアを表す仕様モデル１ｃの記述を解析するのみではこのバグは発見できず、シミュレーションを実行していく過程でしか発見できない。

【００３５】

そこで、実施例装置においては、各信号に対して実行された最後の信号代入の時刻が信号時刻記憶部１０ａに記憶されている。すると、上述した仕様モデル１ｃの場合、信号Ａに相当する時刻記憶部ではシミュレーション開始から６２ｎｓ経過するまでずっとシミュレーション開始時刻が記憶されたままである。判定部１１は信号Ａについての信号時刻記憶部１０ａの値が変化していないと判定する。動作要求表示部１２は、信号Ａに対する信号代入要求が全く実行されていないことを表示器７に表示する。これによって設計者はシミュレーションの早い時点で回路の動作に反映されていない動作要求の記述を発見することができる。

【００３６】

なお、信号時刻記憶部１０ａに記憶するするものは時刻以外にシミュレーションサイクルも考えられる。

【００３７】

また、前述した図２の仕様モデル１ａのように、信号代入要求の一部が実行されない記述についてはこの信号時刻記憶部１０ａに記憶された時刻からは判断し難いが、このような場合も時刻を記憶する代りに各信号に代入予約の数と、代入実行の数を記録しておき、その比率を判定することで、シミュレーションの早い時点で回路の動作に反映されない動作要求記述を表示器７に表示することが可能である。

【００３８】

さらに、実施例装置では指定された時刻の範囲内に入る動作要求を表示するとしたが、範囲外の動作要求を表示することも可能である。

【００３９】

図７は第４の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図である。図５に示した実施例装置と同一部分には同一符号が付してある。したがって、重複する部分の詳細説明は省略する。

【００４０】

この実施例装置においては、図５の実施例装置の信号時刻記憶部１０ａの代りに、並列動作記憶部１０ｂが組込まれている。この並列動作記憶部１０ｂは、要求実行部４におけるシミュレーション実行過程において並列動作が起動される毎にその時の時刻を記憶する。判定部１１は並列動作記憶部１０ｂに記憶された刻が指定時刻の範囲に入るかどうか判定する。そして動作要求表示部１２は指定された時刻の範囲に入る並列動作単位を表示器７に表示する。

【００４１】

このように構成されたハードウェアシミュレーション装置の具体的動作を図８に示すハードウェアの動作を記述した仕様モデル１ｄがシミュレーションされる場合を例にとって説明する。

【００４２】

図８に示す記述によって表される仕様モデル１ｄにおいては、信号Ｂは全く変化しないので、並列動作単位Ｐ０も実行されない。このように、全く起動されない並列動作単位Ｐ０も文法上はエラーでないため、シミュレーション過程でないとこのバグは発見できない。

【００４３】

実施例装置においては、各並列動作単位について最新に実行されたシミュレーションサイクルを記憶する並列動作時刻記憶部１０ｂが設けられている。したがって、上記並列動作単位Ｐ０に対して並列動作時刻記憶部１０ｂには常にシミュレーション開始時刻が記憶されている。判定部１１が、シミュレーション開始時刻が並列動作時刻記憶部１０ｂに記憶されている並列動作単位を判定すると、動作要求表示部１２がこの並列動作単位Ｐ０が指定された時刻に入るため表示器７にに表示する。

【００４４】

設計者は、この表示内容を見て、この動作単位を起動する信号、動作単位Ｐ０については信号Ｂに注目して記述のバグを取り除くことができる。

【００４５】

次に、図９のように非常に多くの並列動作単位を含む仕様モデル１ｅをシミュレーションする場合を説明する。

【００４６】

この仕様モデル１ｅは、１０１個の要素を持つ配列信号Ｔのそれぞれの要素の値を決定する１０１個の動作単位と、そのそれぞれの信号値の変化によって動作するＰ０～Ｐ１００の１０１個の単位動作との合計２０２個の動作単位で構成されている。なお、説明を簡単にするために記述を省略して示してある。

【００４７】

この仕様モデル１ｅにおいては、並列動作単位は多いが、同じ時刻で動作する動作単位は少ない。一般の回路もこの仕様モデル１ｅのように同一時刻で起動される並列動作単位は一部分である。この仕様モデル１ｅのシミュレーション動作過程で、回路に異常動作が現れたとする。この場合、これらの全ての動作単位のうち異常動作の原因となった動作単位を同定するのは困難であり、非常に多くのデバッグ時間を要する。

【００４８】

しかし、実施例装置にように、ある時刻で起動した並列動作単位だけを表示器７に表示すれば、それらの中から、異常動作をを引き起こした動作単位を限定すれば良い。例えば、動作単位Ｐ４９が異常動作を引き起こしたとすると、これと同じ時刻で起動した動作単位は全部の動作単位のごく一部であるため、動作単位Ｐ４９が原因であることを簡単に発見できる。このように、異常動作の原因を究明する際の時間を短縮することができる。

【００４９】

図１０は第５の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図である。なお、図１に示した実施例装置と同一部分には同一符号が付してある。したがって、重複する部分の詳細説明は省略する。

【００５０】

この実施例装置においては、仕様モデル１ｆに記述された信号処理要求等の動作要求及び実行要求時刻は要求読取部２で読取られて一旦要求記憶部３へ登録される。その後、要求実行部４で実際に実行される。

【００５１】

信号値検出部１３は要求実行部４にて実際に一つの信号に対する信号処理要求が実行される毎に、その時点までの該当信号の信号値の最大値及び最小値を検出する。信号限界値記憶部１４は信号値検部１３にて検出された最大値及び最小値をそれぞれ上限値及び下限値として記憶する。比較部１５は要求実行部４にて実際に信号に対する信号代入要求が実行される毎に、信号代入値と信号限界値記憶部１４に記憶された上限値及び下限値各値と比較する。信号限界値更新部１６は信号代入値が上限値より大きいとき、上限値を代入値に更新し、信号代入値が下限値より小さいとき、下限値を代入値に更新する。さらに、限界値表示部１７は、例えば設計者の表示要求に応じて、信号限界値記憶部１４に記憶された

上限値及び下限値を表示器７に表示する。

【００５２】

このように構成されたハードウェアシミュレーション装置の具体的動作を図１１に示すハードウェアの動作を記述した仕様モデル１ｆがシミュレーションされる場合を例にとって説明する。

【００５３】

図１１に示す記述によって表される仕様モデル１ｆにおいては、モジュールＥ１　の入力ポートに外部から入力される信号ＣＡＳＥＮＵＭの値が変化する毎（６行目）に、その時点における信号ＣＡＳＥＮＵＭ値と他の入力ポートＴＢＬの値から出力ポートＭＢＡの値を計算して出力する。例えば、入力信号ＣＡＳＥＮＵＭ値が２の場合、出力ポートＭＢＡの値は、入力ポートＴＢＬの値に２を乗算した値である（１０行目）。

【００５４】

このような仕様モデル１ｆに対して、入力信号ＣＡＳＥＮＵＭに対して複数種類のテストデータを代入してシミュレーションを実行する場合における条件として、入力信号ＣＡＳＥＮＵＭの下限値が４であり、上限値が７であったと仮定する。この場合、９行目乃至１１行目における信号ＣＡＳＥＮＵＭ値が１から３の時と、１６行目の８以上の時のシミュレーションは実行されない。

【００５５】

この場合、

（ａ）　実際はそれらの場合のテストも実行しなくてはならず、現在行われたシミュレーションでは十分幅広いテストが行われていない、

（ｂ）　シミュレーション対象のハードウェアの性質から実際は信号ＣＡＳＥＮＵＭは４以上の値しか取り得ないにも拘らず、仕様モテル１ｆはそれ以外の場合を考慮した記述になっており、無駄が多い、

という２つの可能性が考えられる。前者の場合、これに気付かないと十分に検証が行われていないハードフェアを既に検証済としてしまう可能性もあり、設計時間，費用ともに損失を被ることがある。

【００５６】

また、後者の場合、記述を参考にして論理回路の設計を行う場合、実際に使用されない無駄な回路を設計してしまう可能性があり、チップの面積などに影響を受ける。

【００５７】

実施例装置においては、入力信号ＣＡＳＥＮＵＭ値を複数種類に変更した場合におけるシミュレーションを実行した後で、入力信号ＣＡＳＥＮＵＭの上限値及び下限値からなる信号限界が表示器７に表示される。したがって、設計者は、表示器７の表示情報を読取ることによって、その信号の取り得る値に対して、十分幅広いシミュレーションが終了したか否かを判断できる。

【００５８】

また、実際にはある範囲の値しか取り得ない信号に対して広い範囲の値を取る可能性を考えた無駄な動作要求を記述しているか否かを簡単に判断できる。

【００５９】

なお、本発明は上記実施形態そのままに限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で構成要素を変形して具体化できる。また、上記実施形態に開示されている複数の構成要素の適宜な組み合わせにより、種々の発明を形成できる。例えば、実施形態に示される全構成要素から幾つかの構成要素を削除してもよい。さらに、異なる実施形態にわたる構成要素を適宜組み合わせてもよい。

【００６０】

【発明の効果】

以上説明したように本発明のハードウェアシミュレーション装置においては、シミュレーション実行過程における信号値や実行時刻等の種々の履歴を記憶保持している。そして、これにの情報を必要に応じて表示出力している。したがって、設計者がハードウェアのシ

ミュレーションを行なう上でのハードウェア回路におけるデバッグ作業効率の向上を図ることができる。

【図面の簡単な説明】

【図１】本発明の一実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図。

【図２】同実施例装置でシミュレーションされるハードウェアの仕様モデルを示す図。

【図３】本発明の他の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図。

【図４】同実施例装置で得られる信号波形及び信号処理要求の出力図。

【図５】本発明のさらに別の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図。

【図６】同実施例装置でシミュレーションされるハードウェアの仕様モデルを示す図。

【図７】本発明のさらに別の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図。

【図８】同実施例装置でシミュレーションされるハードウェアの仕様モデルを示す図。

【図９】同実施例装置でシミュレーションされるハードウェアの仕様モデルを示す図。

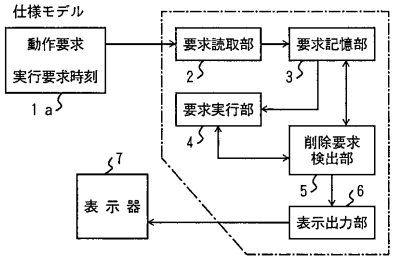
【図１０】本発明のさらに別の実施例に係わるハードウェアシミュレーション装置の概略構成を示すブロック図。

【図１１】同実施例装置でシミュレーションされるハードウェアの仕様モデルを示す図。

【符号の説明】

１ａ，１ｂ，１ｃ，１ｄ，１ｅ，１ｆ…仕様モデル、２…要求読取部、３…要求記憶部、４…要求実行部、５…削除要求検出部、６…表示出力部、７…表示器、８…要求表示部、９…信号表示部、１０ａ…信号時刻記憶部、１０ｂ…並列動作時刻記憶部、１１…判定部、１２…動作要求表示部、１３…信号値検出部、１４…信号限界値記憶部、１５…比較部、１６…信号限界値更新部、１７…限界値表示部。

【図１】



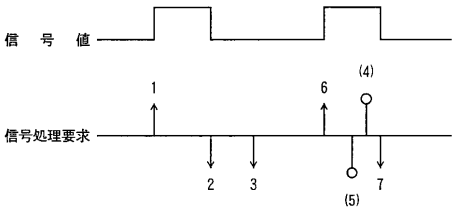
【図２】

```
1    ENTITY E1 IS
2    END E1;

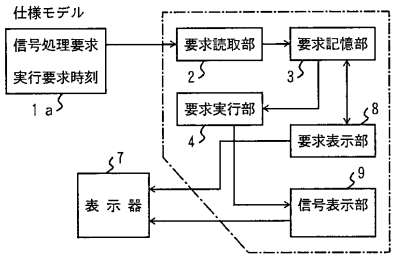
3    ARCHITECTURE B1 OF E1 IS
4    SIGNAL CLK : Bit;
5    SIGNAL A   : Integer;
6    BEGIN -- B1
7    P0 : CLK <= not CLK after 3 ns;
8    P1 : PROCESS (CLK)
9    BEGIN -- PROCESS P1
10     A <=  5 after 1 ns,
11           10 after 1 ns;
12   END PROCESS P1;
13 END B1;
```

1a

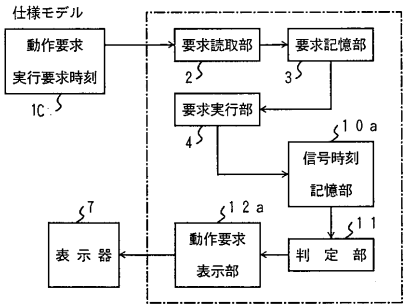
【図３】



【図４】



【図５】



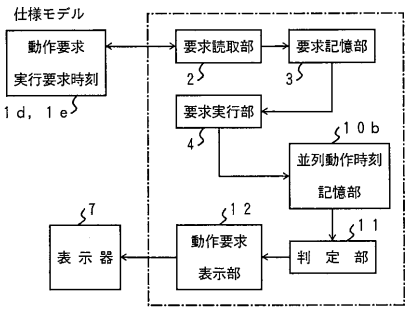
【図６】

```
1  ENTITY E1 IS
2  END E1;

3  ARCHITECTURE B1 OF E1 IS
4  SIGNAL CASENUM : Integer:= -1;
5  SIGNAL A : Integer;
6  BEGIN -- B1
7   P0 : PROCESS (CASENUM)
8   BEGIN -- PROCESS P0
9     CASE CASENUM IS
10    WHEN 0 => A <= 0 after 60 ns;
11    WHEN 1 => A <= 1 after 50 ns;
12    WHEN 2 => A <= 2 after 40 ns;
13    WHEN 3 => A <= 3 after 30 ns;
14    WHEN 4 => A <= 4 after 20 ns;
15    WHEN 5 => A <= 5 after 10 ns;
16    WHEN others => A <= 6 after 0 ns;
17    END CASE;
18  END PROCESS P0;
19  P1 : CASENUM <= CASENUM + 1 after 3 ns;
20 END B1;
```

1c

【図７】



【図８】

```
1  ENTITY E1 IS
2  END E1;

3  ARCHITECTURE B1 OF E1 IS
4  SIGNAL CLK : Bit;
5  SIGNAL A,B : Integer;
6  BEGIN -- B1
7    P0 : PROCESS (B)
8    BEGIN -- PROCESS P0
9     A <= B + A;
10   END PROCESS P0;
11 END B1;
```

1d

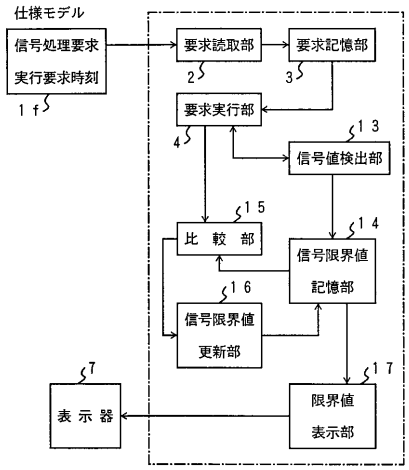
【図９】

```
ENTITY E2 IS
END E2;
ARCHITECTURE B2 OF E2 IS
SIGNAL S,T : Bit_Vector (0 to 100);
BEGIN -- B2
T(0)<= S(0) and T(100) after 0 ns;
T(1)<= S(1) and T(99)  after 1 ns;
      :
T(50)<= S(50) and T(50) after 50 ns;
      :
T(100)<= S(100) and T(0) after 100 ns;

P0 : PROCESS (T(0))
BEGIN -- PROCESS P0
-- statement P0
END PROCESS P0;
      :
P49 : PROCESS (T(49))
BEGIN -- PROCESS P49
-- statement P49 contains something wrong
END PROCESS P49;
      :
P100 : PROCESS (T(100))
BEGIN -- PROCESS P100
-- statement P100
END PROCESS P100;
END B2;
```

1e

【図１０】



【図１１】

```
1  ENTITY E5 IS
2    PORT (CASENUM:in Integer;TBL:in Integer;MBA:out Integer);
3  END E5;
4  ARCHITECTURE B5 OF E5 IS

5  BEGIN

6   P0  PROCESS (CASENUM)                                    ～1f
7   BEGIN
8   CASE CASENUM IS
9     WHEN 1 => MBA <= TBL;
10    WHEN 2 => MBA <= TBL*2;
11    WHEN 3 => MBA <= TBL*3;
12    WHEN 4 => MBA <= TBL*5;
13    WHEN 5 => MBA <= TBL*8;
14    WHEN 6 => MBA <= TBL*10;
15    WHEN 7 => MBA <= TBL*12;
16    WHEN 8 => MBA <= TBL*4;
17    WHEN others => MBA <= TBL*6;
18  END CASE;
19  END PROCESS P0 ;
20 END B5;
```

フロントページの続き

(72)発明者　高橋　奈穂美
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(72)発明者　酒井　浩
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(72)発明者　黒澤　雄一
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
Ｆターム(参考) 5B046 AA08　BA03　JA05