

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6265788号
(P6265788)

(45) 発行日 平成30年1月24日(2018.1.24)

(24) 登録日 平成30年1月5日(2018.1.5)

(51) Int.Cl. F I
G 0 6 F 17/50 (2006.01)
 G 0 6 F 17/50 6 6 4 A
 G 0 6 F 17/50 6 6 4 K

請求項の数 6 (全 15 頁)

(21) 出願番号	特願2014-45607 (P2014-45607)	(73) 特許権者	000006013 三菱電機株式会社 東京都千代田区丸の内二丁目7番3号
(22) 出願日	平成26年3月7日(2014.3.7)	(74) 代理人	100099461 弁理士 溝井 章司
(65) 公開番号	特開2015-170221 (P2015-170221A)	(74) 代理人	100152881 弁理士 山地 博人
(43) 公開日	平成27年9月28日(2015.9.28)	(72) 発明者	竹内 友美 東京都千代田区丸の内二丁目7番3号 三 菱電機株式会社内
審査請求日	平成28年11月28日(2016.11.28)	(72) 発明者	峯岸 孝行 東京都千代田区丸の内二丁目7番3号 三 菱電機株式会社内
		審査官	合田 幸裕

最終頁に続く

(54) 【発明の名称】 シミュレーション装置及びインタフェースモジュール生成装置及びプログラム

(57) 【特許請求の範囲】

【請求項1】

ハードウェアモジュール又はバスモデルの複数の仕様のうちの1つである第1の仕様に従ったハードウェアモジュール又はバスモデルである第1のモデルと、前記複数の仕様のうちの1つであって前記第1の仕様とは異なる第2の仕様に従ったハードウェアモジュール又はバスモデルである第2のモデルとの間の通信のシミュレーションを行うシミュレーション装置であって、

前記第1のモデルにより生成された前記第1の仕様に従った通信データを解析し、前記通信データから、前記複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を抽出する第1の通信データ管理部と

10

前記第1の通信データ管理部により抽出された共通データ要素を用いて前記第2の仕様に従った前記第2のモデル宛ての通信データを生成する第2の通信データ管理部とを有することを特徴とするシミュレーション装置。

【請求項2】

前記第1の通信データ管理部は、

前記第1の仕様に従った通信データから、複数の属性の複数の共通データ要素を抽出し

前記シミュレーション装置は、更に、

共通データ要素の属性ごとに設けられた複数の記憶領域と、

20

前記第1の通信データ管理部により抽出された前記複数の共通データ要素を属性ごとに分類し、前記複数の共通データ要素を、それぞれ、対応する属性の記憶領域に格納するとともに、前記複数の記憶領域から前記複数の共通データ要素を読み出す共通データ要素管理部とを有し、

前記第2の通信データ管理部は、

前記共通データ要素管理部により前記複数の記憶領域から読み出された前記複数の共通データ要素を用いて前記第2の仕様に従った通信データを生成することを特徴とする請求項1に記載のシミュレーション装置。

【請求項3】

ハードウェアモジュール又はバスモデルの複数の仕様のうちの1つである第1の仕様に従ったハードウェアモジュール又はバスモデルである第1のモデルと、前記複数の仕様のうちの1つであって前記第1の仕様とは異なる第2の仕様に従ったハードウェアモジュール又はバスモデルである第2のモデルとの間の通信のシミュレーションを行うコンピュータに、

10

前記第1のモデルにより生成された前記第1の仕様に従った通信データを解析し、前記通信データから、前記複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を抽出する第1の通信データ管理処理と、

前記第1の通信データ管理処理により抽出された共通データ要素を用いて前記第2の仕様に従った前記第2のモデル宛ての通信データを生成する第2の通信データ管理処理とを実行させることを特徴とするプログラム。

20

【請求項4】

前記第1の通信データ管理処理において、

前記コンピュータに、

前記第1の仕様に従った通信データから、複数の属性の複数の共通データ要素を抽出させ、

前記プログラムは、更に、

前記第1の通信データ管理処理により抽出された前記複数の共通データ要素を属性ごとに分類し、前記複数の共通データ要素を、それぞれ、共通データ要素の属性ごとに設けられた前記コンピュータ内の複数の記憶領域のうちの対応する属性の記憶領域に格納するとともに、前記複数の記憶領域から前記複数の共通データ要素を読み出す共通データ要素管理処理を、前記コンピュータに実行させ、

30

前記第2の通信データ管理処理において、

前記コンピュータに、

前記共通データ要素管理処理により前記複数の記憶領域から読み出された前記複数の共通データ要素を用いて前記第2の仕様に従った通信データを生成させることを特徴とする請求項3に記載のプログラム。

【請求項5】

ハードウェアモジュール又はバスモデルの複数の仕様のうちの1つである第1の仕様に従ったハードウェアモジュール又はバスモデルである第1のモデルと、前記複数の仕様のうちの1つであって前記第1の仕様とは異なる第2の仕様に従ったハードウェアモジュール又はバスモデルである第2のモデルとの間の通信のシミュレーションが行われる際に、前記第1のモデルと前記第2のモデルとのインタフェースとして機能するプログラムであるインタフェースモジュールを生成するインタフェースモジュール生成装置であって、

40

前記第1の仕様を解析し、前記第1のモデルにより前記第1の仕様に従って生成される通信データに含まれるデータ要素のうち、前記複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を判別し、判別結果に基づき、前記インタフェースモジュールに含まれるプログラムコードであって、前記通信データから前記共通データ要素を抽出するプログラムコードを、第1のプログラムコードとして生成する第1のプログラムコード生成部と、

50

前記第2の仕様を解析し、前記インタフェースモジュールに含まれるプログラムコードであって、前記第1のプログラムコードの実行により抽出された共通データ要素を用いて前記第2の仕様に従った前記第2のモデル宛ての通信データを生成するプログラムコードを、第2のプログラムコードとして生成する第2のプログラムコード生成部とを有することを特徴とするインタフェースモジュール生成装置。

【請求項6】

ハードウェアモジュール又はバスモデルの複数の仕様のうちの1つである第1の仕様に従ったハードウェアモジュール又はバスモデルである第1のモデルと、前記複数の仕様のうちの1つであって前記第1の仕様とは異なる第2の仕様に従ったハードウェアモジュール又はバスモデルである第2のモデルとの間の通信のシミュレーションが行われる際に、前記第1のモデルと前記第2のモデルとのインタフェースとして機能するプログラムであるインタフェースモジュールを生成するコンピュータに、

10

前記第1の仕様を解析し、前記第1のモデルにより前記第1の仕様に従って生成される通信データに含まれるデータ要素のうち、前記複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を判別し、判別結果に基づき、前記インタフェースモジュールに含まれるプログラムコードであって、前記通信データから前記共通データ要素を抽出するプログラムコードを、第1のプログラムコードとして生成する第1のプログラムコード生成処理と、

前記第2の仕様を解析し、前記インタフェースモジュールに含まれるプログラムコードであって、前記第1のプログラムコードの実行により抽出された共通データ要素を用いて前記第2の仕様に従った前記第2のモデル宛ての通信データを生成するプログラムコードを、第2のプログラムコードとして生成する第2のプログラムコード生成処理とを実行させることを特徴とするプログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、LSI開発において、RTLやSystemC(トランザクションレベル、ピンレベル)等でそれぞれ設計された既存モジュールを用い、早期の段階でシステム全体の検証を容易に実施可能とするための、インタフェース(I/F)モジュール等に関するものである。

30

【背景技術】

【0002】

従来、LSI開発では、Verilog-HDLやVHDLといったハードウェア記述言語による設計(以下、RTL設計と呼ぶ)、検証をメインで行ってきた。

しかし、近年のLSIの大規模化に伴いRTL設計されたハードウェアモジュール(以下、RTLモジュールと呼ぶ)を用いてのシステム全体検証は多大な時間を要することが問題となっており、ハードウェア記述言語よりも抽象度の高いC言語、C++言語、SystemC言語などの高級言語を用いて設計(以下、高位設計と呼ぶ)、検証を行うことで、検証時間の短縮を図る技術が実用化されてきている。

さらに、高位設計されたハードウェアモジュール(以下、高位モジュールと呼ぶ)と既存のRTLモジュールを混在させての検証を可能とする技術もあり、設計の早期段階でシステム全体検証が可能となってきた。

40

【0003】

しかし、開発するLSIのプロトコルと、既存のRTLモジュールのプロトコルが異なる場合や、設計早期の段階でプロトコルが決定しておらず、高位モジュール(ハードウェアモジュールやバスモデル)ではトランザクションレベルでの通信を行い、RTLモジュールのピンレベルでの通信と抽象度が異なる場合などは、そのままでは各モジュール間の接続ができなかった。

そこで、抽象度やプロトコルの異なる各モジュール間に、変換アダプタ(ブリッジ、ラッパ、等と呼ばれることもある)を挿入する技術が一般的に用いられる。

50

しかし、変換アダプタの処理は、抽象度やプロトコル毎に異なるため、接続するモジュールおよびバスモデル間の抽象度とプロトコルの組み合わせの数だけ、変換アダプタが必要となる。

変換アダプタの作成は、ハードウェアモジュールの設計と同等の時間を要する場合もあるため、バス周辺構成の変更や、プロトコルの変更があると、その分作成期間を要することになり、変換アダプタ作成が、システム全体検証のボトルネックになってしまう。

【0004】

従来技術は、上記のような問題点を解決するため、バスモデルと同一のプロトコル仕様であるハードウェアモジュール1と、バスモデルと異なるプロトコル仕様であるハードウェアモジュール2との通信を可能とする、データ受信手段と、データ変換手段と、データ送信手段を備えるラッパモデルを自動生成していた(例えば、特許文献1)。

10

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2010-231633号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

従来技術では、バスモデルと異なるプロトコルを実装したハードウェアモジュール2を、バスモデルに接続可能とするラッパモデルの自動生成時に、ハードウェアモジュール2の実装情報からバス接続情報を取得し、ハードウェアモジュール2の仕様書からトランザクション情報(プロトコル)を抽出し、それらの情報とバスモデルの仕様に基づき、データ受信手段、データ変換手段、データ送信手段を備えるラッパモデルを生成していた。

20

しかし、仕様書は作成者毎に記述方法がまちまちであり、一様に情報を抽出可能とは限らない。

また、早期段階でのシステム全体検証は、高位モジュールとRTLモジュールが混在した環境で行う場合があるが、従来技術のラッパモデル自動生成方法は、ハードウェアモジュール2およびバスモデルがどちらもピンレベルで設計されている場合のみ開示されており、抽象度が異なる場合の自動生成方法については開示されていない。

【0007】

30

本発明は、このような課題を解決し、抽象度やプロトコルといった仕様が異なるモジュールをフレキシブルに接続し、設計早期の段階で、容易にシステム全体のシミュレーションを実行可能とすることを主な目的とする。

【課題を解決するための手段】

【0008】

本発明に係るシミュレーション装置は、

ハードウェアモジュール又はバスモデルの複数の仕様のうちの1つである第1の仕様に従ったハードウェアモジュール又はバスモデルである第1のモデルと、前記複数の仕様のうちの1つであって前記第1の仕様とは異なる第2の仕様に従ったハードウェアモジュール又はバスモデルである第2のモデルとの間の通信のシミュレーションを行うシミュレーション装置であって、

40

前記第1のモデルにより生成された前記第1の仕様に従った通信データを解析し、前記通信データから、前記複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を抽出する第1の通信データ管理部と、

前記第1の通信データ管理部により抽出された共通データ要素を用いて前記第2の仕様に従った前記第2のモデル宛ての通信データを生成する第2の通信データ管理部とを有することを特徴とする。

【発明の効果】

【0009】

50

本発明によれば、第1の仕様に従った通信データから、全ての仕様のシミュレーションにおいて共通に利用される共通データ要素を抽出し、抽出した共通データ要素を用いて第2の仕様に従った通信データを生成するため、仕様異なるモジュールをフレキシブルに接続し、設計早期の段階で、容易にシステム全体のシミュレーションを実行することができる。

【図面の簡単な説明】

【0010】

【図1】実施の形態1に係るI/Fモジュールの構成例を示す図。

【図2】実施の形態1に係るI/Fモジュールを用いたバス周辺構成を示す図。

【図3】実施の形態1に係るシミュレーション装置の構成例を示す図。

【図4】実施の形態1に係るマスタ・スレーブ生成装置の構成例を示す図。

【図5】実施の形態1に係るマスタ・スレーブ生成フローを示すフローチャート図。

【図6】実施の形態1に係るパラメータ設定ファイルの記述例を示す図。

【図7】実施の形態1に係るマスタ部の処理タイミングを示す図。

【図8】実施の形態1に係るシミュレーション装置及びマスタ・スレーブ生成装置のハードウェア構成例を示す図。

【発明を実施するための形態】

【0011】

実施の形態1.

図1は、実施の形態1によるI/Fモジュールの構成を示したものである。

【0012】

I/Fモジュール1は、マスタのハードウェアモジュールと、スレーブのハードウェアモジュール又はバスモデルとの間の通信のシミュレーションを行うコンピュータにより実行されるプログラムである。

I/Fモジュール1は、データ通信のマスタとなるハードウェアモジュールとの接続を行うマスタ部2と、データ通信のスレーブとなるハードウェアモジュールまたはバスモデルとの接続を行うスレーブ部3と、マスタ部2とスレーブ部3を仲介する共通部4で構成される。

以下では、ハードウェアモジュールまたはバスモデルのプロトコル、抽象度を仕様と表記する。

また、以下では、マスタとなるハードウェアモジュールの仕様（プロトコル、抽象度）を、マスタ仕様と表記する。

更に、スレーブとなるハードウェアモジュールまたはバスモデルの仕様（プロトコル、抽象度）を、スレーブ仕様と表記する。

マスタ仕様、スレーブ仕様は、それぞれ、ハードウェアモジュールまたはバスモデルの複数の仕様のうちの1つである。

【0013】

図1において、マスタ部2は、マスタ仕様に対応しており、マスタのハードウェアモジュールにより生成されたマスタ仕様に従った通信データを解析し、当該通信データから、複数の仕様の全ての仕様に対する通信のシミュレーションにおいて共通に利用されるデータ要素である共通データ要素を抽出する。

そして、マスタ部2は、抽出した共通データ要素を、共通部4に渡す。

より具体的には、マスタ部2は、共通データ要素として、通信データから、コマンド関連情報と、データ部分（ペイロード）と、第1の通信データのうちの有効箇所を指定するバイトイネーブルを抽出する。

【0014】

共通部4は、第1の通信データから抽出されたコマンド関連情報を格納するコマンド用バッファ5と、データ部分（ペイロード）を格納するデータ用バッファ6と、第1の通信データのうちの有効箇所を指定するバイトイネーブルを格納するバイトイネーブル用バッファ7と、バッファ制御部8から構成される。

10

20

30

40

50

図 1 では、説明の便宜上、コマンド用バッファ 5、データ用バッファ 6、バイトイネーブル用バッファ 7 を I / F モジュール 1 の構成要素としているが、現実には、これらのバッファは I / F モジュール 1 を実行するコンピュータ内の複数の記憶領域である。

バッファ制御部 8 は、マスタ部 2 により抽出された複数の共通データ要素を属性ごとに分類し、複数の共通データ要素を、それぞれに対応する属性のバッファに格納する。

つまり、バッファ制御部 8 は、第 1 の通信データから抽出されたコマンド関連情報をコマンド用バッファ 5 に格納し、データ部分（ペイロード）をデータ用バッファ 6 に格納し、バイトイネーブルをバイトイネーブル用バッファ 7 に格納する。

また、バッファ制御部 8 は、各バッファから共通データ要素を読み出し、読み出した共通データ要素をスレーブ部 3 に渡す。

10

【 0 0 1 5 】

スレーブ部 3 は、スレーブ仕様に対応している。

スレーブ部 3 は、バッファ制御部 8 により読み出された複数の共通データ要素を用いて、スレーブとなるハードウェアモジュールまたはバスモデル宛てのスレーブ仕様に従った通信データを生成する。

【 0 0 1 6 】

信号線 9 は、マスタとなるハードウェアモジュールからマスタ部 2 へ送信される読み出し・書き込み要求関連の情報、マスタ部 2 からハードウェアモジュールへ送信される応答情報、および、通信データ等である。

信号線 10 は、スレーブ部 3 からスレーブとなるハードウェアモジュールへ送信される読み出し・書き込み要求関連の情報、ハードウェアモジュールからスレーブ部 3 へ送信される応答情報、および、通信データ等である。

20

信号線 11 は、マスタ部 2 と共通部 4 のバッファ制御部 8 で送受信される各種抽象度・プロトコルに共通の情報であり、信号線 12 は、スレーブ部 3 と共通部 4 のバッファ制御部 8 で送受信される各種抽象度・プロトコルに共通の情報であり、信号線 13、14、15 は、それぞれ、コマンド関連情報、データ、バイトイネーブルである。

【 0 0 1 7 】

マスタとなるハードウェアモジュールから I / F モジュール 1 を介してスレーブとなるハードウェアモジュール又はバスモデルへ通信データを送信するシミュレーションでは、マスタ部 2 が第 1 の通信データ管理処理に対応し、スレーブ部 3 が第 2 の通信データ管理処理に対応する。

30

また、マスタとなるハードウェアモジュールが第 1 のモデルに対応し、スレーブとなるハードウェアモジュール又はバスモデルが第 2 のモデルに対応する。

また、マスタ仕様が第 1 の仕様に対応し、スレーブ仕様が第 2 の仕様に対応する。

スレーブとなるハードウェアモジュール又はバスモデルから I / F モジュール 1 を介してマスタとなるハードウェアモジュールへ通信データを送信するシミュレーションでは、スレーブ部 3 が第 1 の通信データ管理処理に対応し、マスタ部 2 が第 2 の通信データ管理処理に対応する。

また、スレーブ仕様が第 1 の仕様に対応し、マスタ仕様が第 2 の仕様に対応する。

また、いずれのシミュレーションにおいても、バッファ制御部 8 が共通データ要素管理処理に対応する。

40

【 0 0 1 8 】

図 2 に、実施の形態 1 による I / F モジュール 1 を用いたバス周辺構成を示す。

プロトコル A が実装されているバスモデル 20 に、プロトコル A が実装されているモジュール 1 (21)、トランザクションレベルで実装されているモジュール 2 (22)、プロトコル A で実装されているモジュール 3 (23)、プロトコル B で実装されているモジュール 4 (24) が接続されている。

このとき、バスモデル 20 と同じプロトコルで実装されているモジュール 1 (21) とモジュール 3 (23) は、直接接続可能であるが、バスモデル 20 とは異なる抽象度、または、プロトコルが実装されているモジュール 2 (22) およびモジュール 4 (24) は

50

、直接は接続できないため、それぞれ I / F モジュール 1 の挿入が必要である。

【 0 0 1 9 】

図 3 は、図 1 に示した I / F モジュール 1 を実行するコンピュータであるシミュレーション装置 1 0 0 の構成例を示す。

【 0 0 2 0 】

I / F モジュール実行部 1 0 1 は、I / F モジュール 1 を実行する。

バッファ領域 1 0 2 は、図 1 に示すコマンド用バッファ 5、データ用バッファ 6、バイトイネーブル用バッファ 7 に対応する記憶領域である。

マスタ・スレーブ実行部 1 0 3 は、マスタとなるハードウェアモジュール、スレーブとなるハードウェアモジュールまたはバスモデルを実行する。

プログラム記憶部 1 0 4 は、I / F モジュール 1、マスタとなるハードウェアモジュール、スレーブとなるハードウェアモジュールまたはバスモデルのそれぞれに対応するプログラムを記憶する。

【 0 0 2 1 】

I / F モジュール実行部 1 0 1 において、マスタ部実行部 1 0 1 1 は、I / F モジュール 1 のマスタ部 2 を実行する。

すなわち、マスタ部実行部 1 0 1 1 は、マスタ部 2 を実行して、マスタ仕様に従った通信データから共通データ要素を抽出し、抽出した共通データ要素をバッファ制御部 8 (バッファ制御部実行部 1 0 1 2) に渡す処理を行う。

また、マスタ部実行部 1 0 1 1 は、マスタ部 2 を実行して、バッファ制御部 8 (バッファ制御部実行部 1 0 1 2) から渡された共通データ要素を用いてマスタ仕様に従った通信データを生成する処理を行う。

マスタ部実行部 1 0 1 1 は、マスタのハードウェアモジュールからスレーブのハードウェアモジュールまたはバスモデルへの通信のシミュレーションでは、第 1 の通信データ管理部として機能する。

また、マスタ部実行部 1 0 1 1 は、スレーブのハードウェアモジュールまたはバスモデルからマスタのハードウェアモジュールへの通信のシミュレーションでは、第 2 の通信データ管理部として機能する。

【 0 0 2 2 】

バッファ制御部実行部 1 0 1 2 は、I / F モジュール 1 のバッファ制御部 8 を実行する。

すなわち、バッファ制御部実行部 1 0 1 2 は、バッファ制御部 8 を実行して、マスタ部 2 (マスタ部実行部 1 0 1 1) から渡された共通データ要素を、それぞれに対応するバッファ (コマンド用バッファ、データ用バッファ、バイトイネーブル用バッファ) に格納し、また、各バッファ内の共通データ要素をスレーブ部 3 (スレーブ部実行部 1 0 1 3) に渡す処理を行う。

また、バッファ制御部実行部 1 0 1 2 は、バッファ制御部 8 を実行して、スレーブ部 3 (スレーブ部実行部 1 0 1 3) から渡された共通データ要素を、それぞれに対応するバッファ (コマンド用バッファ、データ用バッファ、バイトイネーブル用バッファ) に格納し、また、各バッファ内の共通データ要素をマスタ部 2 (マスタ部実行部 1 0 1 1) に渡す処理を行う。

バッファ制御部実行部 1 0 1 2 は、共通データ要素管理部の例に相当する。

【 0 0 2 3 】

スレーブ部実行部 1 0 1 3 は、I / F モジュール 1 のスレーブ部 3 を実行する。

すなわち、スレーブ部実行部 1 0 1 3 は、スレーブ部 3 を実行して、バッファ制御部 8 (バッファ制御部実行部 1 0 1 2) から渡された共通データ要素を用いてスレーブ仕様に従った通信データを生成する処理を行う。

また、スレーブ部実行部 1 0 1 3 は、スレーブ部 3 を実行して、スレーブ仕様に従った通信データから共通データ要素を抽出し、抽出した共通データ要素をバッファ制御部 8 (バッファ制御部実行部 1 0 1 2) に渡す処理を行う。

10

20

30

40

50

スレーブ部実行部 1013 は、マスタのハードウェアモジュールからスレーブのハードウェアモジュールまたはバスモデルへの通信のシミュレーションでは、第 2 の通信データ管理部として機能する。

また、スレーブ部実行部 1013 は、スレーブのハードウェアモジュールまたはバスモデルからマスタのハードウェアモジュールへの通信のシミュレーションでは、第 1 の通信データ管理部として機能する。

【0024】

図 4 は、図 1 に示した I / F モジュール 1 のマスタ部 2 及びスレーブ部 3 を生成するマスタ・スレーブ生成装置 30 の構成例を示す。

マスタ・スレーブ生成装置 30 は、I / F モジュール生成装置の例に相当する。

10

【0025】

マスタ・スレーブ生成装置 30 において、パラメータ解析部 31 は、外部から入力されるパラメータ設定ファイルから、マスタ部 2 が接続するハードウェアモジュール、およびスレーブ部 3 が接続するハードウェアモジュールまたはバスモデルについての抽象度やプロトコルに関する情報を解析する。

【0026】

I / O 解析部 32 は、外部から入力される接続モジュールの I / O 記述ファイルから、マスタ部 2 が接続するハードウェアモジュール、およびスレーブ部 3 が接続するハードウェアモジュールまたはバスモデルの I / O 情報を解析する。

【0027】

生成部 33 は、パラメータ解析部 31 と I / O 解析部 32 からの情報に従ってマスタ部記述（マスタ部 2 に対応するプログラムコード）およびスレーブ部記述（スレーブ部 3 に対応するプログラムコード）を生成する。

20

【0028】

ライブラリ 36 は、各種抽象度におけるマスタ部 2 およびスレーブ部 3 のテンプレート記述と、一般的に用いられるプロトコルにおけるマスタ部記述とスレーブ部記述を予め格納する。

【0029】

ライブラリ読み出し部 35 は、パラメータ設定ファイルの抽象度やプロトコルに関する情報に基づき、ライブラリ 36 よりマスタ部 2 およびスレーブ部 3 のテンプレート記述、または一般的に用いられるプロトコルにおけるマスタ部記述およびスレーブ部記述を読み出し、生成部 33 または出力部 34 に渡す。

30

【0030】

出力部 34 は、生成部 33 またはライブラリ読み出し部 35 より渡されたマスタ部記述およびスレーブ部記述を外部に出力する。

【0031】

信号線 37 は、予めライブラリ 36 に書き込まれる各種テンプレート記述、マスタ部記述、および、スレーブ部記述である。

信号線 38 は、パラメータ設定ファイルから入力される各種パラメータ情報である。

信号線 39 は、I / O 記述ファイルから入力される I / O 情報である。

40

信号線 40 は、生成部 33 からパラメータ解析部 31 へのパラメータ解析命令である。

信号線 41 は、パラメータ解析部 31 から出力されるパラメータ解析結果である。

信号線 42 は、生成部 33 から I / O 解析部 32 への I / O 解析命令である。

信号線 43 は、I / O 解析部 32 から出力される I / O 解析結果である。

信号線 44 は、生成部 33 からライブラリ読み出し部 35 へのライブラリ読み出し命令である。

信号線 45 は、ライブラリ 36 からライブラリ読み出し部 35 に読み出されるテンプレート記述、マスタ部記述、および、スレーブ部記述である。

信号線 46 は、ライブラリ読み出し部 35 から生成部 33 または出力部 34 へ渡すテンプレート記述、マスタ部記述、および、スレーブ部記述である。

50

信号線 47 は、生成部 33 が生成したマスタ部記述およびスレーブ部記述である。

信号線 48 は、出力部 34 から外部へ出力されるマスタ部記述である。

信号線 49 は、出力部 34 から外部へ出力されるスレーブ部記述である。

【 0032 】

図 4 において、生成部 33 が生成するマスタ部記述及びスレーブ部記述は、第 1 のプログラムコード及び第 2 のプログラムコードに対応する。

また、パラメータ解析部 31、I/O 解析部 32 及び生成部 33 は、第 1 のプログラムコード生成部及び第 2 のプログラムコード生成部の例に相当する。

【 0033 】

図 5 は、実施の形態 1 によるマスタ・スレーブ生成装置 30 で実行する、マスタ・スレーブ生成のフロー図であり、マスタ部 2、スレーブ部 3 それぞれについて本フローが実行される。

10

【 0034 】

P01 では、生成部 33 が、パラメータ解析部 31 へ抽象度パラメータの解析命令を出力し、パラメータ解析部 31 はパラメータ設定ファイルより抽象度パラメータを抽出し解析する。

【 0035 】

P02 では、P01 で解析した抽象度がトランザクションレベル (TLM) を示す識別子 (例えば、Loosely Timed の LT や、Approximately Timed の AT) であるか、ピンレベルを示す識別子 (例えば、PIN) であるかによって分岐する。

20

【 0036 】

P11 では、P02 で TLM を示す識別子ならば、生成部 33 が、ライブラリ読み出し部 35 へ TLM 用テンプレートのライブラリ読み出し命令を出力し、ライブラリ読み出し部 35 はライブラリ 36 から対応するテンプレート記述を読み出し、生成部 33 へ渡す。

【 0037 】

P12 では、P11 に続き、生成部 33 がパラメータ解析部 31 へ TLM 用のパラメータ解析命令を出力し、パラメータ解析部 31 はパラメータ設定ファイルより TLM 用の各種パラメータを抽出し解析する。

30

【 0038 】

P13 では、生成部 33 が、P12 で得られたパラメータ解析結果に基づき、P11 で取得したテンプレート記述を書き換え、出力部 34 へ渡す。

【 0039 】

P03 では、P02 でピンレベルを示す識別子ならば、生成部 33 がパラメータ解析部 31 へ IF 分類パラメータ解析命令を出力し、パラメータ解析部 31 がパラメータ設定ファイルより IF 分類のパラメータを抽出し解析する。

【 0040 】

P04 では、P03 で解析した IF 分類が AXI や AHB 等の一般的なプロトコルであり既にライブラリに用意しているプロトコルであるか、それ以外のプロトコル (OTHER) であるかによって分岐する。

40

【 0041 】

P21 では、P04 でそれ以外のプロトコルを示す識別子 (OTHER) ならば、生成部 33 が、ライブラリ読み出し部 35 へ PIN 用テンプレートのライブラリ読み出し命令を出力し、ライブラリ読み出し部 35 はライブラリ 36 から対応するテンプレート記述を読み出し、生成部 33 へ渡す。

【 0042 】

P22 では、P21 に続き、生成部 33 が I/O 解析部 32 へ I/O 解析命令を出力し、I/O 解析部 32 は I/O 記述ファイルより I/O 情報を抽出し解析する。

【 0043 】

50

P 2 3では、P 2 2に続き、生成部 3 3がパラメータ解析部 3 1へピンレベル用のパラメータ解析命令を出力し、パラメータ解析部 3 1はパラメータ設定ファイルよりピンレベル用の各種パラメータを抽出し解析する。

【 0 0 4 4 】

P 2 4では、生成部 3 3が、P 2 2とP 2 1で得られたI / O解析結果およびパラメータ解析結果に基づき、P 2 1で取得したテンプレート記述を書き換え、出力部 3 4へ渡す。

【 0 0 4 5 】

P 3 1では、P 0 4で一般的なプロトコルであり既にライブラリに用意しているプロトコルを示す識別子ならば、生成部 3 3が、ライブラリ読み出し部 3 5へ対応するマスタ部記述またはスレーブ部記述のライブラリ読み出し命令を出力し、ライブラリ読み出し部 3 5はライブラリ 3 6から対応するマスタ部記述またはスレーブ部記述を読み出し、出力部 3 4へ渡す。

【 0 0 4 6 】

P 0 5では、出力部 3 4が、P 1 3、P 2 4、または、P 3 1で得られたマスタ部記述またはスレーブ部記述を外部へ出力する。

【 0 0 4 7 】

続いて、図 6 (a)、(b)、(c)は、実施の形態 1 によるパラメータ解析部 3 1が解析するパラメータ設定ファイル内の記述例である。

【 0 0 4 8 】

図 6 (a)は、マスタ部 2 に接続するハードウェアモジュールの抽象度がL Tであることを示しており、以下A R G U M E N T、T L _ C M D、T L _ A D D R、T L _ L E N等は、抽象度がT L Mである場合のパラメータであり、共通部 4 に格納するコマンド関連情報、データ等がどの変数に対応しているのかを示す。

一方、スレーブ部 3 に接続するハードウェアモジュールまたはバスモデルの抽象度はピンレベルであることを示している。

【 0 0 4 9 】

図 6 (b)は、マスタ部 2 に接続するハードウェアモジュールの抽象度がピンレベルであり、I F分類がA X Iであり、既にライブラリ 3 6にマスタ部記述が用意されていることを示している。

【 0 0 5 0 】

図 6 (c)は、マスタ部 2 に接続するハードウェアモジュールの抽象度がピンレベルであり、I F分類がO T H E Rであることを示している。

以下、W C M D、R C M D、R E S P、W D A T A等は抽象度がピンレベルである場合のパラメータであり、共通部 4 に格納するコマンド関連情報、データ等がどの信号に対応しているのかを示す。

【 0 0 5 1 】

図 7 は、図 6 (c) のパラメータ設定である場合のマスタ部の処理をタイミング図で示したものである。

【 0 0 5 2 】

“ W C M D = { M a d d r } @ (M C m d = 1) ” は、M C m d が 1 のときに書き込み要求としてM a d d rを取得することを意味する。

“ W D A T A = { M d a t a } @ (M C m d = 1) ” は、M c m d が 1 のときに書き込みデータとしてM d a t aを取得することを意味する。

このため、図 7 の 5 0 のM c m d = 1 のタイミングにおいて、マスタ部 2 は、M A d d r とM d a t aをハードウェアモジュールから取得し、共通部 4 へ送信する。

共通部 4 は、書き込み要求を正常に受理すると、受理したことを示すA C C E P T信号をマスタ部 2 へ送信する。

“ R E S P = { S C m d A c c e p t = 1 } @ A C C E P T ” は、A C C E P T信号がアサートされたら応答としてS C m d A c c e p tに1をアサートすることを意味してい

10

20

30

40

50

るので、マスタ部2は図7の51でS C m d A c c e p tを1にする。

【0053】

以上のように、本実施の形態に係るI/Fモジュール1を用いれば、ハードウェアモジュールまたはバスモデルの各種抽象度や各種プロトコルで共通に情報を処理する共通部4を設けているため、抽象度が異なる場合の変換も可能である。

つまり、I/Fモジュール1をマスタ部2、スレーブ部3、および、共通部4で構成し、各種抽象度やプロトコルで共通に必要な情報をバッファに格納するようにしたこと、および、パラメータ設定ファイルにおいて抽象度の指定ができるようにしたこと、従来技術では開示されていなかった、異なる抽象度間の変換にも対応可能となる。

また、パラメータ設定ファイルから、抽象度やプロトコルに関連する情報を指定可能としたことで、作成者によらず一様に、一意にI/Fモジュール1のマスタ部2およびスレーブ部3を自動生成することができるので、設計の早期段階でのシステム全体検証を容易に実施可能となる。

つまり、作成者によらず一様に、一意に情報を取得可能となるので、作成者によって記述方法が異なる仕様書から一様に情報を抽出することが困難であった従来技術の課題を解決できる。

【0054】

最後に、本実施の形態に示したシミュレーション装置100及びマスタ・スレーブ生成装置30のハードウェア構成例を図8を参照して説明する。

シミュレーション装置100及びマスタ・スレーブ生成装置30はコンピュータであり、シミュレーション装置100及びマスタ・スレーブ生成装置30の各要素をプログラムで実現することができる。

シミュレーション装置100及びマスタ・スレーブ生成装置30のハードウェア構成としては、バスに、演算装置901、外部記憶装置902、主記憶装置903、通信装置904、入出力装置905が接続されている。

【0055】

演算装置901は、プログラムを実行するCPU(Central Processing Unit)である。

外部記憶装置902は、例えばROM(Read Only Memory)やフラッシュメモリ、ハードディスク装置である。

主記憶装置903は、RAM(Random Access Memory)である。

通信装置904は、例えばNIC(Network Interface Card)である。

入出力装置905は、例えばマウス、キーボード、ディスプレイ装置等である。

図3のバッファ領域102は、例えば、主記憶装置903で実現される。

また、図3のプログラム記憶部104、図4のライブラリ36は、例えば、外部記憶装置902で実現される。

【0056】

プログラムは、通常は外部記憶装置902に記憶されており、主記憶装置903にロードされた状態で、順次演算装置901に読み込まれ、実行される。

プログラムは、図3及び図4に示す「~部」(プログラム記憶部104を除く、以下も同様)として説明している機能を実現するプログラムである。

更に、外部記憶装置902にはオペレーティングシステム(OS)も記憶されており、OSの少なくとも一部が主記憶装置903にロードされ、演算装置901はOSを実行しながら、図3及び図4に示す「~部」の機能を実現するプログラムを実行する。

また、本実施の形態の説明において、「~の抽出」、「~の解析」、「~の生成」、「~の検知」、「~の設定」、「~の選択」、等として説明している処理の結果を示す情報やデータや信号値や変数値が主記憶装置903にファイルとして記憶されている。

【0057】

なお、図8の構成は、あくまでもシミュレーション装置100及びマスタ・スレーブ生

10

20

30

40

50

成装置 30 のハードウェア構成の一例を示すものであり、シミュレーション装置 100 及びマスタ・スレーブ生成装置 30 のハードウェア構成は図 8 に記載の構成に限らず、他の構成であってもよい。

【符号の説明】

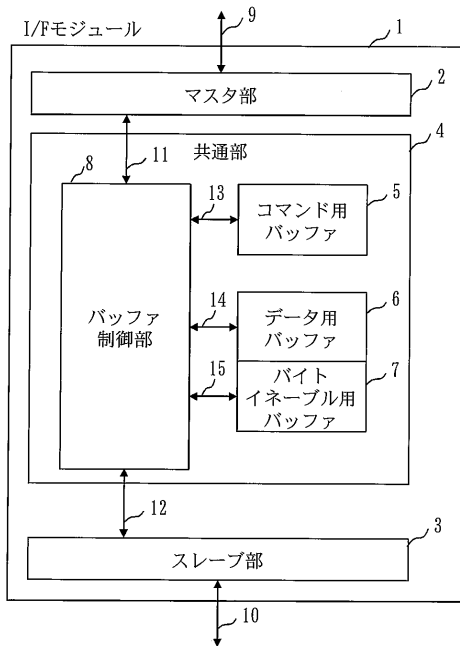
【0058】

1 I/Fモジュール、2 マスタ部、3 スレーブ部、4 共通部、5 コマンド用バッファ、6 データ用バッファ、7 バイトイネーブル用バッファ、8 バッファ制御部、9 信号線、10 信号線、11 信号線、12 信号線、13 信号線、14 信号線、15 信号線、30 マスタ・スレーブ生成装置、31 パラメータ解析部、32 I/O解析部、33 生成部、34 出力部、35 ライブラリ読み出し部、36 ライブラリ、37 信号線、38 信号線、39 信号線、40 信号線、41 信号線、42 信号線、43 信号線、44 信号線、45 信号線、46 信号線、47 信号線、48 信号線、49 信号線、100 シミュレーション装置、101 I/Fモジュール実行部、102 バッファ領域、103 マスタ・スレーブ実行部、104 プログラム記憶部、1011 マスタ部実行部、1012 バッファ制御部実行部、1013 スレーブ部実行部。

10

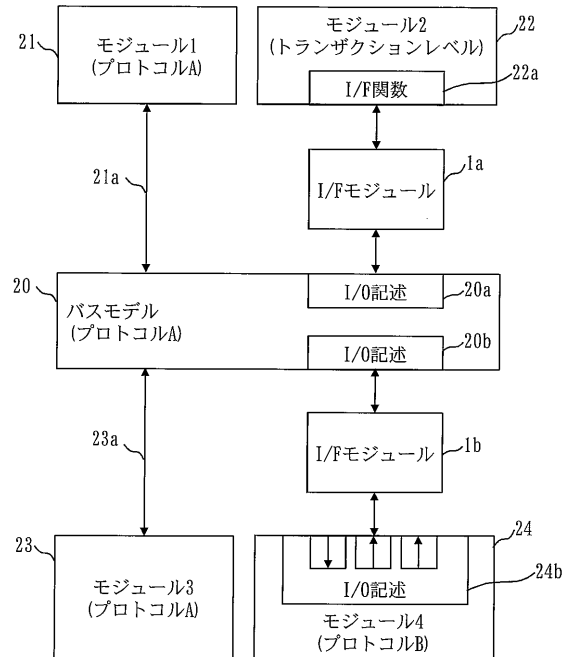
【図1】

実施の形態1によるI/Fモジュールの構成

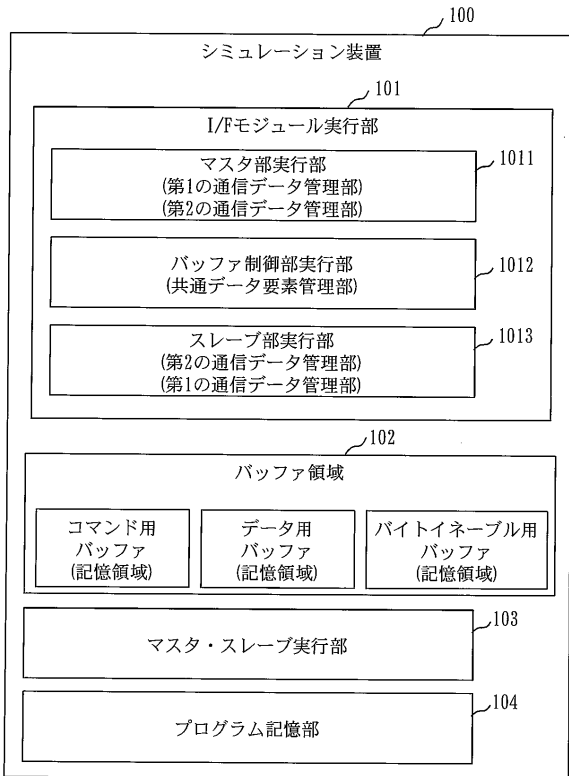


【図2】

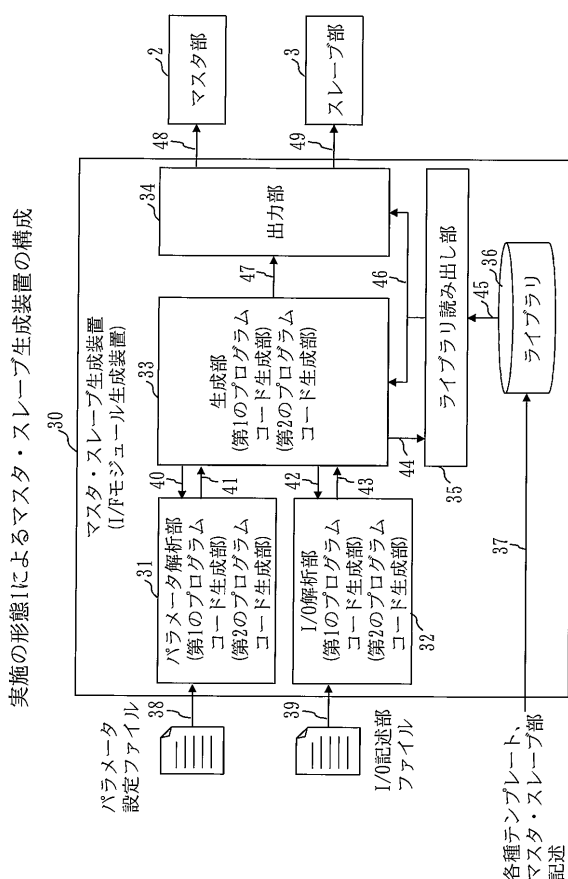
実施の形態1によるI/Fモジュールを用いたバス周辺構成



【図3】

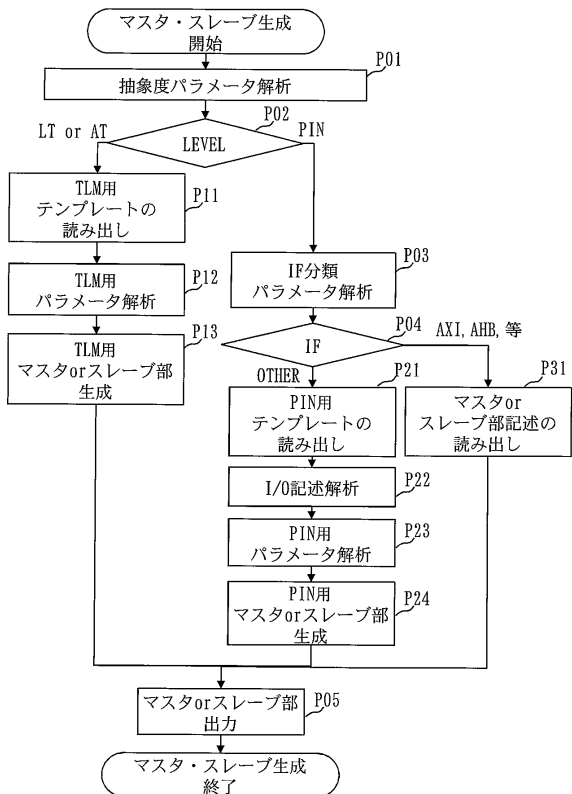


【図4】



【図5】

実施の形態1によるマスタ・スレーブ生成装置およびプログラムで実行するマスタ・スレーブ生成フロー



【図6】

パラメータ設定ファイル記述例

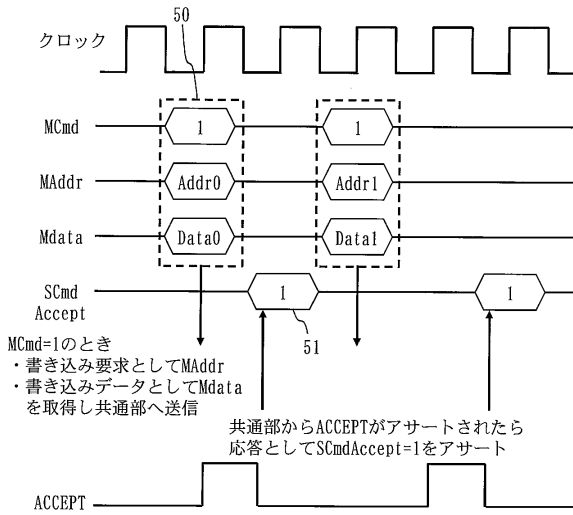
```
(a)
MASTER //以下、マスタ部の設定
LEVEL=LT //抽象度レベル=TLM (Loosely Time)
ARGUMENT=trn //I/F関数の引数
TL_CMD=trn.command //コマンドとしてtrn.cmdを取得
TL_ADDR=trn.address //アドレスとしてtrn.addrを取得
TL_LEN=trn.length //バースト長としてtrn.lengthを取得
...
SLAVE //以下、スレーブ部の設定
LEVEL=PIN //抽象度レベル=PIN
...
```

```
(b)
MASTER //以下、マスタ部の設定
LEVEL=PIN //抽象度レベル=ピンレベル
IF=AXI //IF分類=AXI
SLAVE //以下、スレーブ部の設定
LEVEL=PIN //抽象度レベル=PIN
...
```

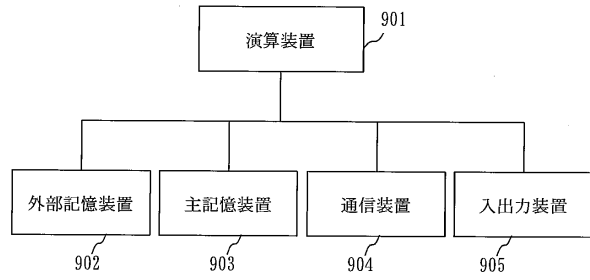
```
(c)
MASTER //以下、マスタ部の設定
LEVEL=PIN //抽象度レベル=ピンレベル
IF=OTHER //IF分類=その他
WCMD= (Maddr) @ (MCmd=1) //MCmd信号が1のとき、書き込み要求としてMAddrを取得
RCMD= (Maddr) @ (MCmd=2) //MCmd信号が2のとき、読み出し要求としてMAddrを取得
RESP= (SCmdAccept=1) @ ACCEPT //応答はWCMD/RCMD受信時に、SCmdAcceptとして出力
WDATA= (Mdata) @ (MCmd=1) //MCmd=1のとき、書き込みデータとしてMDataを取得
...
SLAVE //以下、スレーブ部の設定
LEVEL=PIN //抽象度レベル=PIN
...
```

【図7】

図6(c)のパラメータ設定である場合のマスタ部の処理タイミング



【図8】



フロントページの続き

- (56)参考文献 特開2010-231633(JP,A)
特開2006-343942(JP,A)
特開2004-013227(JP,A)
酒井 皓太, 垣内 洋介, 浜口 清治, トランザクション識別子を伴うバスプロトコル間の変換器
自動生成手法, 研究報告システムLSI設計技術(SLDM), 日本, 情報処理学会, 2011年 3
月11日, 第2011-SLDM-149巻/第32号, 第1-6頁
- (58)調査した分野(Int.Cl., DB名)
G06F 17/50
IEEE Xplore
JSTPlus(JDreamIII)