【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第１区分
【発行日】平成24年6月28日(2012.6.28)

【公表番号】特表2008-531982(P2008-531982A)
【公表日】平成20年8月14日(2008.8.14)
【年通号数】公開・登録公報2008-032
【出願番号】特願2007-540439(P2007-540439)
【国際特許分類】
Ｇ０１Ｒ　31/28　　(2006.01)
Ｇ０６Ｆ　 9/44　　(2006.01)
【ＦＩ】
Ｇ０１Ｒ　31/28　　Ｈ
Ｇ０６Ｆ　 9/06　　６２０Ｋ

【誤訳訂正書】
【提出日】平成24年5月14日(2012.5.14)
【誤訳訂正１】
【訂正対象書類名】特許請求の範囲
【訂正対象項目名】全文
【訂正方法】変更
【訂正の内容】
【特許請求の範囲】
【請求項１】
オープンアーキテクチャテストシステムにおいて、複数のソフトウェアコンポーネントと複数のベンダハードウェアモジュールバージョンとを関連付ける方法であって、
ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取る段階と、
前記ベンダハードウェアモジュールによってサポートされた一連のソフトウェアコンポーネントを受け取る段階と、
前記一連のハードウェアバージョンを、一のマスク値によりマスクすることにより等価となる複数のハードウェアバージョン番号の等価集合として処理する前記一連のハードウェアバージョンを処理するための段階と、
前記ベンダハードウェアモジュールのハードウェアバージョンをユーザに選択させる段階と、
ユーザが選択した前記ハードウェアバージョンが前記等価集合の要素であるか否かに応じて、ユーザの前記ハードウェアバージョンの選択が有効であるかどうかを決定する段階と、
ユーザによる前記ハードウェアバージョンの選択に従って、システムプロファイルを作成する段階と
を含む方法。
【請求項２】
前記一連のソフトウェアコンポーネントは、
一以上のベンダソフトウェア開発キットと、
一以上のユーザソフトウェア開発キットと、
前記ベンダハードウェアモジュールの前記一連のハードウェアバージョンに対応する一以上のベンダドライバソフトウェアバージョンと
を含む請求項１に記載の方法。
【請求項３】
前記一連のハードウェアバージョンを処理する段階は、
前記マスク値を定義する段階であって、前記マスク値は前記ハードウェアバージョンの

番号がマスクされた番号およびマスクを含み、前記マスクは前記一連のハードウェアバージョンの番号のビット列範囲内のデータパターンであり、前記データパターンは、少なくとも０で表現されるドントケアビットおよび１で表現されるマスクされないビットを含み、前記マスク値と前記ハードウェアバージョンの番号とを論理演算して前記マスクされた番号を算出して定義する段階と、

前記マスクおよび前記マスクされた番号に従って、前記一連のハードウェアバージョンのうち、ソフトウェアコンポーネントによりサポートされる複数のハードウェアバージョンを、前記ソフトウェアコンポーネント毎に複数のハードウェアバージョン番号の等価集合を特定する段階と

を含む請求項１または２に記載の方法。

【請求項４】

前記複数のハードウェアバージョン番号の等価集合を特定する段階は、

前記マスクにおけるそれぞれの前記ドントケアビット毎に、前記マスクされた番号における対応ビットをマスクし、前記等価集合の全ての要素を前記対応ビットのビット位置において１または０で表す段階と、

前記マスクにおけるそれぞれのマスクされないビットは、前記等価集合の前記マスクされた番号における前記対応ビットと等しくさせる段階と

を含む請求項３に記載の方法。

【請求項５】

前記ハードウェアバージョンの選択が有効であるかどうかを決定する段階は、

ユーザが選択した前記ハードウェアバージョンの番号が前記マスク値によって覆われる場合に、前記マスク値に対応する前記等価集合の全体が選択されたとして示す段階を含む請求項３または４に記載の方法。

【請求項６】

前記ハードウェアバージョンの選択が有効であるかどうかを決定する段階は、

ユーザが選択したハードウェアバージョンのそれぞれに対して、ハードウェアバージョンの番号および前記マスクのビット単位の論理積が前記マスクされた番号と等しい場合のみ、ユーザによる前記ハードウェアバージョンの選択を有効として表す段階を含む請求項５に記載の方法。

【請求項７】

前記システムプロファイルを生成する段階は、

ユーザの前記ハードウェアバージョンの選択に従って、一連の互換性のあるソフトウェアコンポーネントを決定する段階と、

ユーザの前記ハードウェアバージョンの選択のそれぞれによってソフトウェアコンポーネントのバージョンを選択する段階と、

ユーザの前記ハードウェアバージョンの選択のそれぞれに対して、ユーザに選択された複数のソフトウェアコンポーネントのリストを生成する段階と、

ユーザに選択された複数のソフトウェアコンポーネントの前記リストを格納する段階と

を含む請求項１から６のいずれか一項に記載の方法。

【請求項８】

前記ソフトウェアコンポーネントのバージョンを選択する段階は、

前記一連の互換性のあるソフトウェアコンポーネントからソフトウェアコンポーネントのバージョンを選択することをユーザに問い合わせる段階と、

前記ソフトウェアコンポーネントの前記バージョンのユーザ選択のそれぞれに対応する、ユーザが選択した前記ハードウェアバージョンの共通部分を計算する段階であって、サポートされた複数のハードウェアバージョンのリストを生成するための段階と

を含む請求項７に記載の方法。

【請求項９】

前記共通部分を計算する段階は、

前記ユーザが選択した前記ハードウェアバージョンのそれぞれを表現する複数の等価集

合に含まれる複数のマスクについて、ビット単位の論理和を演算することで前記共通部分のマスクを計算する段階と、
前記複数のマスクに対応する複数の前記マスクされた番号について、ビット単位の論理和を演算することで、前記共通部分のマスクされた番号を計算する段階と
を含む請求項８に記載の方法。
【請求項１０】
前記オープンアーキテクチャテストシステムは、
システムコントローラ、一以上のサイトコントローラ、一以上のベンダハードウェアモジュール、および一以上の被試験デバイス（ＤＵＴ）を含み、
前記ベンダハードウェアモジュールの、複数のハードウェアバージョン番号の前記等価集合の一要素である新規ハードウェアバージョンを受け取り、
前記システムプロファイルを利用する前記ベンダハードウェアモジュールの前記新規ハードウェアバージョンをテストする
請求項１から９のいずれか一項に記載の方法。
【請求項１１】
システムコントローラと、
前記システムコントローラに結合されるサイトコントローラと、
前記サイトコントローラに結合される一以上のベンダハードウェアモジュールと、
ベンダハードウェアモジュールに同時に結合される一以上の被試験デバイス（ＤＵＴ）と
を備え、
前記ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取るための手段と、
前記ベンダハードウェアモジュールによってサポートされた一連のソフトウェアコンポーネントを受け取るための手段と、
前記一連のハードウェアバージョンを、一のマスク値によりマスクすることにより等価となる複数のハードウェアバージョン番号の等価集合として処理する前記一連のハードウェアバージョンを処理するための手段と、
前記ベンダハードウェアモジュールのハードウェアバージョンをユーザに選択させるための手段と、
ユーザが選択した前記ハードウェアバージョンが前記等価集合の要素であるか否かに応じて、ユーザの選択が有効であるかどうかを決定するための手段と、
ユーザによる前記ハードウェアバージョンの選択に従って、システムプロファイルを生成するための手段と
を含むオープンアーキテクチャテストシステム。
【請求項１２】
前記一連のソフトウェアコンポーネントは、
一以上のベンダソフトウェア開発キットと、
一以上のユーザソフトウェア開発キットと、
前記ベンダハードウェアモジュールの前記一連のハードウェアバージョンに対応する一以上のベンダドライバソフトウェアバージョンと
を含む請求項１１に記載のオープンアーキテクチャテストシステム。
【請求項１３】
前記一連のハードウェアバージョンを処理するための手段は、
前記マスク値を定義するための手段であって、前記マスク値は前記ハードウェアバージョンの番号がマスクされた番号およびマスクを含み、前記マスクは前記一連のハードウェアバージョンの番号のビット列範囲内のデータパターンであり、前記データパターンは、少なくとも０で表現されるドントケアビットおよび１で表現されるマスクされないビットを含み、前記マスク値と前記ハードウェアバージョンの番号とを論理演算して前記マスクされた番号を算出して定義する手段と、

前記マスクおよび前記マスクされた番号に従って、前記一連のハードウェアバージョンのうち、ソフトウェアコンポーネントによりサポートされる複数のハードウェアバージョンを、前記ソフトウェアコンポーネント毎に複数のハードウェアバージョン番号の等価集合を特定するための手段と

を含む請求項１１または１２に記載のオープンアーキテクチャテストシステム。

【請求項１４】

前記複数のハードウェアバージョン番号の等価集合を特定するための前記手段は、

前記マスクにおけるそれぞれの前記ドントケアビット毎に、前記マスクされた番号における対応ビットをマスクし、前記等価集合の全ての要素を前記対応ビットのビット位置において１または０で表す手段と、

前記マスクにおけるそれぞれのマスクされないビットは、前記等価集合の前記マスクされた番号における前記対応ビットと等しくさせる手段と

を含む請求項１３に記載のオープンアーキテクチャテストシステム。

【請求項１５】

前記ハードウェアバージョンの選択が有効であるかどうかを決定するための手段は、

ユーザが選択した前記ハードウェアバージョンの番号が前記マスク値によって覆われる場合に、前記マスク値に対応する前記等価集合の全体が選択されたとして示す手段を含む請求項１３または１４に記載のオープンアーキテクチャテストシステム。

【請求項１６】

前記複数のハードウェアバージョンの選択が有効であるかどうかを決定するための前記手段は、

ユーザが選択したハードウェアバージョンのそれぞれに対して、ハードウェアバージョンの番号および前記マスクのビット単位の論理積が前記マスクされた番号と等しい場合のみ、ユーザによる前記複数のハードウェアバージョンの選択を有効として表すための手段を含む請求項１５に記載のオープンアーキテクチャテストシステム。

【請求項１７】

前記システムプロファイルを生成するための手段は、

ユーザの前記ハードウェアバージョンの選択に従って、一連の互換性のあるソフトウェアコンポーネントを決定するための手段と、

ユーザの前記ハードウェアバージョンの選択のそれぞれに対して、複数のソフトウェアコンポーネントのバージョンを選択するための手段と、

ユーザの前記ハードウェアバージョンの選択のそれぞれに対して、ユーザに選択された複数のソフトウェアコンポーネントのリストを生成するための手段と、

ユーザに選択された複数のソフトウェアコンポーネントの前記リストを格納するための手段と

を含む請求項１１から１６のいずれか一項に記載のオープンアーキテクチャテストシステム。

【請求項１８】

前記ソフトウェアコンポーネントのバージョンを選択するための手段は、

前記一連の互換性のあるソフトウェアコンポーネントからソフトウェアコンポーネントのバージョンを選択することをユーザに問い合わせるための手段と、

前記ソフトウェアコンポーネントの前記バージョンのユーザ選択のそれぞれに対応する、ユーザが選択した前記ハードウェアバージョンの共通部分を計算するための手段であって、サポートされた複数のハードウェアバージョンのリストを生成するための手段と

を含む請求項１７に記載のオープンアーキテクチャテストシステム。

【請求項１９】

前記共通部分を計算するための手段は、

前記ユーザが選択した前記ハードウェアバージョンのそれぞれを表現する複数の等価集合に含まれる複数のマスクについて、ビット単位の論理和を演算することで前記共通部分のマスクを計算するための手段と、

前記複数のマスクに対応する複数の前記マスクされた番号について、ビット単位の論理和を演算することで、前記共通部分のマスクされた番号を計算するための手段と
を含む請求項１８に記載のオープンアーキテクチャテストシステム。
【請求項２０】
前記ベンダハードウェアモジュールの、複数のハードウェアバージョン番号の前記等価集合の一要素である新規ハードウェアバージョンを受け取るための手段と、
前記システムプロファイルを利用する前記ベンダハードウェアモジュールの前記新規ハードウェアバージョンをテストするための手段と
をさらに含む請求項１１から１９のいずれか一項に記載のオープンアーキテクチャテストシステム。
【請求項２１】
システムコントローラ、サイトコントローラ、一以上のベンダハードウェアモジュール、および一以上の被試験デバイス（ＤＵＴ）を含むオープンアーキテクチャテストシステムにおける複数のベンダハードウェアモジュールバージョンを表すための方法であって、
前記ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取る段階と、
前記一連のハードウェアバージョンを、一のマスク値によりマスクすることにより等価となる複数のハードウェアバージョン番号の等価集合として処理する前記一連のハードウェアバージョンを処理するための段階と、
前記マスク値は前記ハードウェアバージョンの番号がマスクされた番号およびマスクを含み、
前記マスクは、前記一連のハードウェアバージョンの番号のビット列範囲内のデータパターンであり、
前記データパターンは、少なくとも０で表現されるドントケアビットおよび１で表現されるマスクされないビットを含む方法。
【請求項２２】
前記一連のハードウェアバージョンを表す段階は、
前記マスクにおけるそれぞれの前記ドントケアビット毎に、前記マスクされた番号における対応ビットをマスクし、前記等価集合の全ての要素を前記対応ビットのビット位置において１または０で表す段階と、
前記マスクにおけるそれぞれのマスクされないビットは、前記等価集合の前記マスクされた番号における前記対応ビットと等しくさせる段階と
をさらに含む請求項２１に記載の方法。
【請求項２３】
前記ベンダハードウェアモジュールのハードウェアバージョンをユーザに選択させる段階と、
ユーザによる前記ベンダハードウェアモジュールの前記ハードウェアバージョンの選択が有効であるかどうかを決定する段階と、
をさらに含む請求項２１または２２に記載の方法。
【請求項２４】
前記ハードウェアバージョンの選択が有効であるかどうかを決定する段階は、
ユーザが選択したハードウェアバージョンのそれぞれに対して、ハードウェアバージョンの番号および前記マスクのビット単位の論理積が前記マスクされた番号と等しい場合のみ、ユーザによる前記ハードウェアバージョンの選択を有効として表す段階と、
ユーザによる前記ハードウェアバージョンの選択が有効と決定された場合、前記マスク値に対応する等価集合の全体が選択されたとして示す段階と
をさらに含む請求項２３に記載の方法。
【請求項２５】
前記ベンダハードウェアモジュールの、複数のハードウェアバージョン番号の前記等価集合の一要素である新規なハードウェアバージョンを受け取る段階と、

複数のハードウェアバージョン番号の前記等価集合を利用する前記ベンダハードウェアモジュールの前記新規なハードウェアバージョンをテストする段階と
をさらに含む請求項２１から２４のいずれか一項に記載の方法。
【誤訳訂正２】
【訂正対象書類名】明細書
【訂正対象項目名】全文
【訂正方法】変更
【訂正の内容】
【発明の詳細な説明】
【発明の名称】オープンアーキテクチャテストシステムにおけるベンダハードウェアモジュール更新のコンパクト表現および方法
【技術分野】
【０００１】
本発明は、半導体テストのための自動試験装置（ＡＴＥ）の分野に関する。特に、本発明は、効率的なフィールド配置のためのオープンアーキテクチャテストシステムにおけるベンダハードウェアモジュールバージョンのコンパクト表現に関する。
【背景技術】
【０００２】
オープンアーキテクチャテストシステムにおいて、ソフトウェアは、ユーザの特定のシステムを構成する全てのベンダハードウェアモジュール、及びこれらのベンダハードウェアモジュールに付属する全ての異なるモジュールソフトウェアコンポーネントについて、各バージョンの付属モジュールソフトウェアで動作可能なベンダハードウェアモジュールのバージョンのリストを示すベンダ仕様をアクセスする必要がある。そしてシステムは、この複雑な情報を効果的に分析して、システムのソフトウェアコンテンツの候補として、ユーザに提供すべきソフトウェアコンポーネントの選択肢を決定しなければならない。
【０００３】
理想的には、ユーザが与えられた選択肢の中から採用する任意の完全なソフトウェアコンポーネントのセットが一体として機能し、かつユーザが選択したハードウェア構成で動作するという全般的な制約の下で、ソフトウェアコンポーネントの選択肢は、可能な限り大きいセットであるべきである。この問題と関連して、一の方法は、整数のコンマ区切りされたリストのように、ソフトウェアコンポーネントそれぞれに対して互換性のあるハードウェアバージョンを一覧にし、ハードウェアバージョン番号（連続している番号を示すための範囲指示子とともに）を表現することである。この方法は、多数の問題を有する。
【０００４】
第１に、ハードウェアバージョン番号は、例えば３２ビット程度といった大きさとなり、かつ番号内の異なるビットフィールドはハードウェアベンダによって異なる独自の重要性を持つ。当該番号の中のビットの特定のパターンについて全ての可能な値を有する全てのバージョンを単一のソフトウエアバージョンと互換性があるとして、ベンダが認証する場合、このことは下記の結果に至る。ａ）範囲指示子を通して効果的に表すことができない、バージョン番号の連続しない配列、および／またはｂ）記憶および／または処理の目的に対して極めて実用的でないかもしれないとても長いリストを生じる、非常に大きなバージョン番号。第２に、この方法は非常に拘束的なものであり、特にハードウェア製品の更新が非常に頻繁である場合には、リストへの頻繁な追加が求められ、結果としてこの追加を組み込むためにユーザの現場においてシステムプロファイルの再生成を行う面倒で頻繁な作業が必要になる。従って、効果的なフィールド配置のために、オープンアーキテクチャテストシステムにおいて、ベンダハードウェアモジュールバージョンのコンパクト表現が必要である。
【発明の開示】
【課題を解決するための手段】
【０００５】

一の方法は、ハードウェアモジュールベンダが、コンパクトな方法で、オープンアーキテクチャテストシステムにおいて、関連するモジュールソフトウェアの与えられたバージョンとともに機能することを補償するハードウェアバージョン番号の任意の配列を特定できるようなものとして開示される。加えて、ハードウェアバージョン番号は、ベンダハードウェアモジュールに対して最適な一連のソフトウェアコンポーネントの選択肢を、ユーザに提供するように処理され、現場に展開するための有効なシステム設定を導く。

【０００６】

一の実施例において、オープンアーキテクチャテストシステムにおけるベンダハードウェアモジュールバージョンとソフトウェアコンポーネントとを関連付けるための方法は、ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取る段階と、ベンダハードウェアモジュールによってサポートされた一連のソフトウェアコンポーネントを受け取る段階と、マスク値を利用するハードウェアバージョン番号の等価集合として表される一連のハードウェアバージョンを処理する段階と、ベンダハードウェアモジュールのハードウェアバージョンのユーザ選択を得る段階と、ベンダハードウェアモジュールのハードウェアバージョンのユーザ選択を有効にする段階と、ハードウェアバージョンのユーザ選択に従って、システムプロファイルを生成する段階とを含む。

【０００７】

他の実施例において、オープンアーキテクチャテストシステムは、システムコントローラ、システムコントローラに結合されたサイトコントローラ、サイトコントローラに結合された一または複数のハードウェアモジュール、およびハードウェアモジュールの群によって同時に情報が提供される一または複数の被試験デバイス（ＤＵＴ）を含む。システムは、ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取るための手段、ベンダハードウェアモジュールによってサポートされた一連のソフトウェアコンポーネントを受け取る手段、マスク値を利用するハードウェアバージョン番号の等価集合として表される一連のハードウェアバージョンを処理する手段と、ベンダハードウェアモジュールのハードウェアバージョンのユーザ選択を得る手段と、ベンダハードウェアモジュールのハードウェアバージョンのユーザ選択を有効にする手段と、ハードウェアバージョンのユーザ選択に従って、システムプロファイルを生成する手段とをさらに含む。

【０００８】

さらにもう一つの実施形態において、オープンアーキテクチャテストシステムにおけるベンダハードウェアモジュールバージョンを表すための方法は、ベンダハードウェアモジュールの一連のハードウェアバージョンを受け取る段階と、マスク番号およびマスクを包含するマスク値を利用するハードウェアバージョン番号の等価集合として、一連のハードウェアバージョンを表す段階とを含む。マスクは、一連のハードウェアバージョンの範囲内のデータパターンであり、データパターンは、少なくともひとつのドントケアビットおよびマスクされないビットを含む。

【発明を実施するための形態】

【０００９】

方法およびシステムは、オープンアーキテクチャテストシステムにおけるベンダハードウェアモジュールバージョンのコンパクト表現のために提供される。以下の説明は、本発明を製造することおよび本発明を利用することを当業者に可能にするために提示される。特定の実施形態および用途の説明は、単に例として提供される。ここに述べられた例の様々な変更および組み合わせは、当業者に容易に明確になり、ここで定義された一般的な原理は、他の例および用途に適用されてもよい。従って、本発明は、述べられおよび示される例を限定する目的ではなく、ここに記載された原理および技術と調和する最も広い範囲を許容される。

【００１０】

再構成可能なオープンアーキテクチャテストシステムは、サードパーティのソフトウェアおよびハードウェア（ベンダハードウェアモジュールとしても知られる）を開発し、個々に認証を受け、オープンアーキテクチャテストシステムに確実に統合するソリューショ

ンを提供する。図１ａは、本発明の実施例による一または複数のＳｉｔｅＣと、多数のＤＵＴの同時テストに対するオープンアーキテクチャテストシステムを示す。システムは、システムコントローラ１０２、複数のサイトコントローラ１０４、複数のベンダハードウェアモジュール１０６、および対応するＤＵＴ１１０の群を含み、ＤＵＴの群のそれぞれは、一または複数の個々のＤＵＴ１１２を含んでもよい。テストヘッド１０８は、ＤＵＴ群１１０を保持する。一の実施形態において、システムコントローラおよびサイトコントローラは、多数の地理的場所における多数のコンピュータに分配されたコンピュータシステムにより実施されてもよい。他の実施形態において、システムコントローラおよびサイトコントローラは、単一コンピュータにより実施されてもよい。オープンアーキテクチャテストシステムの詳細な説明は、米国特許出願番号６０／４４７，８３９、"半導体集積回路に対してテストプログラムを展開する方法および構造"、および米国特許出願番号６０／５７３，５７７、"オープンアーキテクチャテストシステムにおけるソフトウェア展開"により提供される。加えて、オープンアーキテクチャテストシステムは、"モジュラオープンアーキテクチャテストシステムにおけるテストプログラミング環境"、プラマニックその他（Ａ．Ｐｒａｍａｎｉｃｋ　ｅｔ　ａｌ．）、２００４年ＩＥＥＥインターナショナルテスト会議、２００４年１０月、４１３－４２２頁にも述べられている。

【００１１】

そのようなオープンアーキテクチャテストシステムにおいて、サードパーティソリューションが統合フレームワークの要求に順応する限り、モジュラユニットのそれぞれ、ソフトウェアおよびハードウェアの両方は、サードパーティユニットによって置き換えることができる。従って、サードパーティハードウェアは、デジタルピンカード、アナログカード、デバイス電源などのような任意の機能ユニットでもよい。ベンダハードウェアモジュール（または略してモジュール）の用語は、ハードウェア機能ユニットのように参照されて利用され、さらにベンダハードウェアモジュールのサポートにおける関連するベンダソフトウェアコンポーネントはモジュールソフトウェアと呼ぶ。後者は、モジュール制御ドライバソフトウェア、校正および診断ソフトウェア、エミュレーションソフトウェア、ベンダ特定パターンコンパイラ、等のような、いくつかの独特なソフトウェアコンポーネントを含む。

【００１２】

オープンアーキテクチャテストシステムは、テストシステムの動作だけではなく、テスト環境を制御すること、及び新規ベンダハードウェアモジュールを全てのそれらに付随するソフトウェアとともに導入することも提供する。図１ｂは、オープンアーキテクチャテストシステムにおいて、多数のベンダハードウェアモジュールバージョン１２８およびそれらに対応するソフトウェアコンポーネントバージョンの例を示す。ソフトウェアコンポーネントは、ベンダソフトウェアドライバ１２４の多数のバージョンと同様に、テスタオペレイティングシステム（ＴＯＳ）１２０の多数のバージョンおよびその関連するベンダソフトウェア開発キット（ＳＤＫ）１２２およびユーザＳＤＫ１２６のバージョンを含む。オープンアーキテクチャテストシステムにおける、ＴＯＳ、ベンダＳＤＫ、およびユーザＳＤＫのような多数のベンダハードウェアモジュールバージョンおよびソフトウェアコンポーネントバージョンの管理のための方法は、本願の出願人に譲渡された、米国特許出願番号６０／６３５，０９４、"テスタ計測モジュールのインストールおよび設定管理を実施するための方法およびシステム"により提供される。

【００１３】

ソフトウェアのバージョンｎが、与えられたシステム設定において共存する可能性のある他のバージョンのベンダハードウェアモジュールに互換性があると、あるハードウェアモジュールＭが認証される限りにおいて、あるハードウェアモジュールＭに対するモジュールソフトウェアの特定のバージョンｎが、ある特定のシステム構成に互換性があるものとしてテストシステムによって認証されることができるように、テストシステムは、十分にフレキシブルであることが要求される。ベンダは、Ｍに対するモジュールソフトウェアのバージョンｎとともに機能することが補償されるモジュールＭに対するハードウェアバ

ージョンのリストを公開することが要求される。
【００１４】
ユーザが、特定のバージョンのベンダハードウェアモジュールおよびそれらに関連するモジュールソフトウェアでオープンアーキテクチャテストシステムを構成する場合、ユーザの選択が、実際に動作可能とするシステムをもたらすことを確認することは、システムソフトウェアに委ねられる。従って、大きなシステムが、それぞれが多数の関連するソフトウェアコンポーネントを伴った数十の異なるベンダハードウェアモジュールを有することができるとするならば、そのようなオープンアーキテクチャテストシステムは、ユーザにプロファイル生成の間、ハードウェアおよびソフトウェアの両方のシステム設定について有効な選択のみをユーザに強いるシステムからユーザへの適切なガイダンスを用いてユーザにより生成されるシステムプロファイル情報に依存する必要がある。従って、不必要にユーザを制限することなしに、システムプロファイルを生成するユーザへの適切なガイダンスを提供するタスクは、とても複雑である。
【００１５】
一般的に、効果的な現場展開のためにオープンアーキテクチャテストシステムにおけるベンダハードウェアモジュールバージョンのコンパクト表現を提供する方法は、２つの主要なこと、１）ハードウェアバージョンを特定すること、２）ハードウェアバージョン情報を処理してユーザが有効なシステムプロファイルを生成することを可能にすることを含む。当業者は、オープンアーキテクチャテストシステムにおいて、ＴＯＳ、ベンダＳＤＫ、およびユーザＳＤＫのバージョンのようなソフトウェアコンポーネントバージョンを表すことに適用できる方法を理解できるであろう。
【００１６】
一の実施例において、互換性のあるハードウェアバージョン仕様は、ハードウェアバージョン番号における任意の一連の（ベンダが特定された）ビットをマスキングすることによって達成される。例えば、ハードウェアバージョン番号の構文は、仕様ファイルにおいて、以下、１０９５１８８７８４、０ｘ０１４０３５００／０ｘ０ｆｆ０ｆｆ００，８８０００００１１１のように表現されてもよく、そこで、番号"１０９５１８８７８４"および"８８０００００１１１"は、マスクされていない、単一ハードウェアバージョンであることを示し、番号"０ｘ０１４０３５００／０ｘ０ｆｆ０ｆｆ００"は、マスクされた番号を示し、等価ハードウェアバージョン番号のファミリ（集合）全体を表す。
【００１７】
この例において、マスクされた値は、１６進数フォーマットにおいて表され、構文"ｎ／ｍ"において'ｎ'は、マスクされた番号を表し、'ｍ'はマスクを表す。表現"ｎ／ｍ"において、マスクされた番号ｎに対するマスクｍは、"ＥＱＣ（ｎ／ｍ）"と示されたｎに基づく番号の等価集合（ＥＱＣ）を特定する、以下のようなビットパターンとして定義される。１）ｍにおいてドントケア（本例において"０"ビットにより表現される）であるビット毎に、ｎにおける対応ビットはマスクされ、等価集合の全ての要素がそのビット位置において１または０を有することができることを表す。２）ｍにおいてマスクされないビット（本例において"１"ビットにより表現される）であるビット毎に、等価集合の任意の番号における対応ビットは、ｎにおける対応ビットに等しい。
【００１８】
図２は、本発明の一実施形態によるマスクされた番号へのマスクの適用を示す。図２において、タプル"０１０／１１０"（説明の容易のために１６進数フォーマットの代わりにバイナリフォーマットにおいて表現される）を検討する。マスクｍ２０２は、バージョン番号２０６の等価集合を生成するために番号ｎ２０４に適用される。
【００１９】
上記で与えられたマスクの定義に適用して、ＥＱＣ（０１０／１１０）の一要素のような番号ｘは、ビット位置２および３のそれぞれ（右からまたは最小桁から、左または最上位桁へ３ビットを数えて）において、'１'および'０'を有する。これは、マスク"１１０"におけるビット位置２および３が、１（すなわち、マスクされていない）であり、これは

、ｘにおいて、これら２つのビットが、ｎにおける対応ビットの正確な値を有していることを意味する。しかしながら、マスク"１１０"におけるビット－１の値が'０'（すなわち、マスクされている）ことから、ｘは、ビット位置１において'１'または'０'を有することができる。従って、ＥＱＣ（０１０／１１０）は、｛０１０，０１１｝の組、すなわち｛２，３｝の組、であり、従って、ｘは２または３をとることが可能である。

【００２０】

オープンアーキテクチャテストシステムにおいて、共通に実施される動作は、ユーザによって提供されるハードウェアバージョン番号ｘがマスクされた等価集合の一要素かどうかを判断することである。マスクされた等価集合における構成要素であるかを判断する方法のひとつは、"＆"がビット単位の論理積演算子であるとすれば、ｘ＆ｍ＝ｎである場合にのみ、番号ｘが、等価集合（ｎ／ｍ）の一要素であると見なすことである。この特性は、要求された決定を実施することの簡単な方法を提供する。

【００２１】

次に、多数の連続性の無いバージョン番号の等価集合のマスクされた表現を有するシステムプロファイルを生成するための方法が述べられる。一般的に、方法は、１）一連のマスクされたハードウェアバージョンを処理する段階、２）ベンダハードウェアモジュールのユーザの選択したバージョンを実行する、互換性のあるソフトウェアコンポーネントバージョンの選択をユーザに提供する段階、３）モジュールに対して互換性のあるソフトウェアコンポーネントバージョンのユーザの選択から生じる、ベンダハードウェアモジュールのそれぞれに対して一連のハードウェアバージョンのコンパクト表現を計算する段階と、４）ユーザシステムプロファイルにおけるコンパクト表現を格納する段階とを含む。

【００２２】

一連のマスクされたハードウェアバージョンを処理するために、ベンダハードウェアモジュールおよびそれらの関連するソフトウェアコンポーネントのバージョンを表現するために利用される表記表は、以下のように述べられる。第１に、特別なベンダハードウェアモジュールをＭとして示させ、ＡＴＥにおけるハードウェアモジュールＭを制御するために必要とされるソフトウェアコンポーネントを｛ＭＸ｝の組によって示させ、そこで、ＸはＡ、Ｂ、Ｃ等とすることができる。従って、例えば、Ｍが全部で４つのソフトウェアコンポーネントを有している場合、それらは、ＭＡ、ＭＢ、ＭＣ、およびＭＤとして表される。次に、ベンダハードウェアモジュールＭに対してソフトウェアコンポーネントＸの特別なバージョンｋをＭＸ．ｋとして示される。例えば、モジュールＭに対するコンポーネントＡのバージョン１は、ＭＡ．１として示される。さらには、ベンダハードウェアモジュールＭに対するソフトウェアコンポーネントＸのｋ番目のバージョンによりサポートされた一連のハードウェアバージョンは、ＨＷＲ（ＭＸ．ｋ）として示される。例えば、ＨＷＲ（ＭＡ．１）は、｛３，５，０１０／０１１｝の組を表す。簡単のために留意することは、マスクされた番号がバイナリフォーマットにおいて表されることである。通常の当業者は、マスクされた番号が１６進数または他のフォーマットで表されてもよいことを理解できるであろう。

【００２３】

図３は、本発明の一実施形態によるオープンアーキテクチャテストシステムおける異なるベンダハードウェアモジュールバージョンをテストするためにシステムプロファイルを生成するための方法を示す。当該方法は、開始状態（３０２）において始まり、その後、ステップ１（３０４）へと移動し、そこで、方法は、テストシステムにおけるモジュールＭに対するハードウェアバージョン番号の選択をユーザに問い合わせる。上述したように、ハードウェアバージョンに対して任意のマスクされた値および結果として生じたハードウェアバージョン番号のファミリを利用することは、ユーザに対して等価集合ＥＱＣ（ｎ／ｍ）で表すという課題を伴う。特にｍにおいて特定されたマスクされたビットの数を考慮すると、ｘがＥＱＣ（ｎ／ｍ）の一要素であるような全ての可能な値ｘの完全な列挙（および次の保守）は、指数関数的に複雑になり得る。ｍの値の大きさが中くらいである場合（例えば、ｍが３２より小さいまたは等しい場合）であってさえ、完全な列挙の方法は

明らかに非効率的かつ非実用的である。従って、ユーザは、ハードウェアバージョン値のユーザ選択を入力することが指示される。

【００２４】

ステップ２（３０６）において、方法は、ユーザ入力のそれぞれが有効な選択であるかどうかを決定する。決定は、ハードウェアバージョン番号がＥＱＣの一要素であるかどうかを決定することにおいて、上述した方法を利用して実施される。ユーザ入力が有効の場合（３０６＿ｙｅｓ）、方法は、ステップ３へと続く。そうでない場合（３０６＿ＮＯ）、方法は、ステップ１を繰り返し、ハードウェアバージョン番号を再入力することをユーザに問い合わせる。

【００２５】

ステップ３（３０８）において、ハードウェアバージョン番号のユーザ入力が成功すると、方法は、ＨＷＲ（ＭＸ．ｋ）（Ｍに対するハードウェアバージョンのユーザの選択）に互換性のあるＸのソフトウェアコンポーネントｋの一連の互換性のあるバージョンを決定する。それぞれのステップにおいて、マスクされた値に則してハードウェアバージョン番号をテストするために上述した方法が利用される。ユーザの選択のそれぞれに対して、それがマスクされた値によって覆われている場合、そのマスクされた値に対応する全体の等価集合は、選択されたものとみなされる。この特性は、ハードウェアバージョン番号の効果的なコンパクト表現を得るために大切である。

【００２６】

ステップ４（３１０）において、方法は、上記で決定された一連の互換性のあるソフトウェアコンポーネントからソフトウェアコンポーネントの一のバージョンを選択するために、ユーザに問い合わせをする。

【００２７】

ステップ５（３１２）において、ステップ１で得られたハードウェアバージョン選択のそれぞれに対して、方法は、ＨＷＲ（ＭＸ．ｐ）の全てのＸにわたり、共通部分を計算する。ここで、ｐは、Ｘからソフトウェアコンポーネントの選択されたバージョンを示す。共通部分の計算の詳細な論議は、図４の論議に関連して下記の通りである。ソフトウェアコンポーネントＸのユーザ選択バージョンｐのそれぞれに対して、このステップは、全てのＨＷＲ（ＭＸ．ｐ）の組の共通部分を計算することにより、全ての互換性のあるハードウェアバージョンの全リストを評価する。このステップは、ユーザのハードウェアバージョン選択のそれぞれ（単一値または単一ＥＱＣにおいて結果として生じるユーザ選択のそれぞれとともに）に対してこの計算を繰り返すとともに、結果を収集する。

【００２８】

ステップ６（３１４）において、方法は、全ての選択されたハードウェアバージョンにわたるステップ５から結果の収集を利用し、ユーザのソフトウェアバージョン選択に互換性がある、Ｍに対するハードウェアバージョンの最終リストを計算する。

【００２９】

ステップ７（３１６）において、方法は、ユーザのプロファイルに上記で計算された最終リストを格納する。方法は、終了状態（３１８）において終了する。

【００３０】

図４は、本発明の一実施形態によるマスクされた表現の共通部分を計算するための方法を示す。図４に示すように、二つのコンポーネントm_Iおよびm_Jの共通部分が計算される。なお、それぞれの選択はマスクされたもの（すなわち、等価集合の全体）と仮定する。通常の当業者は、ハードウェアバージョン番号の、マスクされていない選択とマスクされた選択の共通部分、またはマスクされていない選択とマスクされていない選択の共通部分の計算のための方法に拡張することが可能であるのを理解できるであろう。図４に示されるように、m_Iに対するユーザ選択は、ＥＱＣ（n_I／m_I）によって表現され、m_Jに対するユーザ選択は、ＥＱＣ（n_J／m_J）によって表現される。共通部分の結果はＥＱＣ（ｎ／ｍ）とされる。

【００３１】

方法は、開始状態（４０２）において始まり、その後ステップ１（４０４）へと移動し、そこで方法は、結果として生じるマスクを計算する。記号"｜"は、ビット単位の論理和演算子を表す。ビット単位の論理和演算子の定義およびマスクの定義により、m_Iおよびm_Jの両方により満足される単一のマスクであるので、m_Iおよびm_Jのビット単位の論理和の結果は、実際に結果により生じるマスクである。留意することは、ｍが空（すなわちすべて１）である場合、その時、ＥＱＣ（ｎ／ｍ）は単一の値ｎを構成する。ステップ２（４０６）において、方法は、結果として生じるマスクされた番号ｎを計算する。方法は、終了状態（４０８）において終了する。通常の当業者は、共通部分の上記計算を２つより多い要素に拡張することができ、計算において、次の繰り返しのための開始値としてステップ２において計算されたｎおよびｍの値を利用することが可能であることなどは、理解できるであろう。

【００３２】

以下の例は、上記図３および４において述べられた方法に適用する。特別なベンダハードウェアモジュールMは、三つのソフトウェアコンポーネント、コンポーネントＭＡ、コンポーネントＭＢ、およびコンポーネントＭＣを有すると仮定する。特定のユーザがこれらのコンポーネントの利用可能なソフトウェアバージョンに対して有する一連の選択および、それらに対応するサポートされたハードウェアバージョンは、テーブル１において与えられることも仮定する。ここで、表記法"ＭＸ．ｎ"は、ソフトウェアコンポーネントＭＸのｎ番目のバージョンを示す。説明の容易さのために、マスクされた番号は、１６進数に代えてバイナリにおいて表現される。加えて、一連の全ての可能なハードウェアバージョンは三つのビットにより表されることができると仮定する。

【００３３】

【表１】

モジュールmに対するSWコンポーネント	SWコンポーネントによりサポートされるハードウェアバージョン
MA.1	3, 5, 010/011
MA.2	5, 000/100
MA.3	000/110
MB.1	010/010, 5
MB.2	000/000
MC.1	5, 010/110

テーブル１

【００３４】

留意することは、最終的な目標は、モジュールＭに対するハードウェアバージョンの選択とともに動作させるために用意された各ソフトウェアコンポーネント（すなわち、ＭＡ、ＭＢ、およびＭＣのそれぞれの一バージョン）について、ユーザが正確に一つのバージョンを選択できるようにすることである。上記例に図３の方法を適用する。

【００３５】

【表２】

１．ユーザ選択は、全体のＭＸ．ｎの組に対してテストされ、すなわち、

｛３，５，０１０／０１１，０００／１００，０００／１１０，０１０／０１０，０００／０００，０１０／１１０｝

ユーザ選択を、Ｍに対するハードウェアバージョン‘２’および‘５’と仮定する。以下の関連においてこの結果は、

ａ．ハードウェアバージョン‘２’に対して、関連は（すなわち、ＨＷＲ（ＭＸ．ｎ）$_2$の組は）、

MA.1 ⇔ 010/011

MA.2 ⇔ 000/100

MA.3 ⇔ Ø（‘２’はＭＡ．３によってサポートされない）

MB.1 ⇔ 010/010

MB.2 ⇔ 000/000

MC.1 ⇔ 010/110

ｂ．ハードウェアバージョン‘５’に対して、関連は（すなわち、ＨＷＲ（ＭＸ．ｎ）$_5$の組は）、

MA.1 ⇔ 5

MA.2 ⇔ 5

MA.3 ⇔ Ø（‘５’はＭＡ．３によってサポートされない）

MB.1 ⇔ 5

MB.2 ⇔ 000/000

MC.1 ⇔ 5

【００３６】

【表３】

２．上記の関連命令であって、ユーザに対して利用可能なソフトウェアコンポーネント選択は、

一つの{*MA*.1, *MA*.2},

一つの{*MB*.1, *MB*.2}，および

一つの {*MC*.1}.

【００３７】
【表４】
３．以下、ＭＡ．２、ＭＢ．２およびＭＣ．１である、ユーザ（上記２におけるコンポーネント群のそれぞれから一つを選択する）選択と仮定する。その時、
ａ．ハードウェアバージョン選択‘２’に対して、サポートされたハードウェアバージョンのリストは、ＨＷＲ（ＭＡ．２）$_2$、ＨＷＲ（ＭＢ．２）$_2$およびＨＷＲ（ＭＣ．１）$_2$の組の共通部分によって与えられ、以下のステップによれば、
ｉ．図４の方法を以下のように利用して、ＨＷＲ（ＭＢ．２）$_2$とＨＷＲ（ＭＡ．２）$_2$との共通部分を計算し：000/100 ∩ 000/000 ⇒ 000/100 （そこで 000 & 100 = 000, および 000 & 000 = 000)、
ｉｉ．ＨＷＲ（ＭＣ．１）$_2$と０００／１００の共通部分を計算し：000/100 ∩ 010/110 ⇒ 010/110 （そこで 010 & 100 = 000, および 010 & 110 = 010)、
ｉｉｉ．従って、ハードウェアバージョン‘２’の選択に対する最終的な等価集合は、０１０／１１０であり、すなわち｛２，３｝である。
ｂ．ハードウェアバージョン選択‘５’に対して、サポートされたハードウェアバージョンのリストは、ＨＷＲ（ＭＡ．２）$_5$、ＨＷＲ（ＭＢ．２）$_5$、ＨＷＲ（ＭＣ．１）$_5$の組の共通部分によって与えられ、
ｉ．第１に、以下のように図３の方法を利用して、ＨＷＲ（ＭＢ．２）$_5$とＨＷＲ（ＭＡ．２）$_5$との共通部分を計算し：５ ∩ 000/000 ⇒ 101/111 ∩ 000/000 ⇒ 101/111、
ｉｉ．その時、ＨＷＲ（ＭＣ．１）$_5$と１０１／１１１との共通部分を計算し：101/111 ∩ 5 ⇒ 101/111 ∩ 101/111 ⇒ 101/111 ⇒ 5、
ｉｉｉ．従って、ハードウェアバージョン‘５’の選択に対する最終値は、５である。

【００３８】
【表５】
４．上記の３ａおよび３ｂから、図３にステップ６を適用し、ソフトウェアコンポーネントＭＡ．２、ＭＢ．２およびＭＣ．１のその選択に対してユーザのプロファイルに含められるサポートされるハードウェアバージョンのリストは、従って、｛０１０／１１０，５｝、すなわち｛２，３，５｝である。

【００３９】
従って、ハードウェアモジュールベンダは、オープンアーキテクチャテストシステムにおける関連するモジュールソフトウェアの与えられたバージョンとともに動作することが補償されたハードウェアバージョン番号の任意の配列を特定することができる。加えて、ハードウェアバージョン番号は、ベンダハードウェアモジュールに対して最適な一連のソフトウェアコンポーネント選択がユーザに提供されるように処理され、現場に展開するための有効なシステム設定に導く。
【００４０】
本発明は、少なくとも以下の利益を提供する。第１に、サポートされたモジュールハードウェアバージョンのコンパクト表現が採用され、ベンダハードウェアモジュールに対してサポートされた実際の製品バージョンのファミリ全体を表現する。第２に、記載された方法は、ベンダが互換性のあるソフトウェアコンポーネントに対してハードウェアバージ

ョン仕様を公開することを可能にし、同時に、特定された値により暗に示された全てのハードウェアバージョンをサポートする。第３に、ベンダが、正しく連続するバージョン番号を選択することに限定されないように、任意のビットパターンは、ベンダにより特定されることができる。第４に、ベンダが特定した値は、集合を形成する個々のハードウェアバージョン番号の全てのコンパクト集合としてオープンアーキテクチャテストシステムソフトウェアによって処理されることができる。第５に、開示された方法は、ユーザの選択されたベンダハードウェアモジュールと関連する多数のソフトウェアモジュールに対する多数の互換性のあるハードウェアバージョンリストを効率的に分析し、正確なシステム動作を補償しつつ、選択すべき最適な互換性のあるソフトウェアコンポーネントのセットをユーザに提示する。最後に、記載された方法は、ユーザのシステムプロファイルにおいて、選択されたモジュールそれぞれに対する等価ハードウェアバージョンの全体のファミリの最適なコンパクト表現を計算し、格納する。従って、同じファミリから他のベンダハードウェアモジュールを置換することは、ハードウェアバージョンリストの再発行およびシステムプロファイルの再生成のどちらも要求しない。

【００４１】

当然のことながら、明瞭のための上記開示は、異なる機能ユニットおよびプロセッサに関連して本発明の実施形態を記載した。しかしながら、異なる機能ユニットまたはプロセッサの間の機能性の任意の適切な分配は、本発明から損なうことなく利用可能であることは明白である。例えば、別々のプロセッサまたはコントローラによって実施されることで示された機能性は、同じプロセッサまたはコントローラによって実施されうる。従って、特定の機能ユニットへの関連は、厳格な論理、もしくは物理的な構造または組織の指標よりもむしろ、記載された機能性を提供するための適切な方法に関連するようにのみ理解される。

【００４２】

本発明は、ハードウェア、ソフトウェア、ファームウェアまたはこれらの任意の結合を含む任意の適切なフォームにおいて実施される。本発明は、一または複数のデータプロセッサおよび／またはデジタル信号プロセッサ上で実行するコンピュータソフトウェアとして部分的に随意に実施されてもよい。本発明の一実施形態の要素およびコンポーネントは、任意の適する方法で、物理的、機能的、および論理的に実施されてもよい。実際に機能性は、単一ユニットにおいて、複数のユニットにおいて、または他の機能ユニットの一部として実施されてもよい。そのように、本発明は、単一ユニットで実施されてよく、または異なるユニットおよびプロセッサの間で物理的および機能的に分配されてもよい。

【００４３】

関連ある当業者は、同じ根底にある基礎機構および手法をさらに採用しつつ、開示された実施形態の多数の可能な改良および結合が利用できることを理解できる。前述の記載は、説明の目的のために、特定の実施形態に関連して記載されている。しかしながら、上記の図示する議論は、記述された種類に緻密に網羅的ではなく、または本発明を限定することを目的とするものではない。多数の改良および変形は、上記の技術に照らして可能である。実施形態は、本発明およびそれらの実用的な用途の原則を説明するとともに、他の当業者に意図された特別な利用に合わされるような様々な改良とともにある本発明および様々な実施形態の最適な利用を可能にするように選択および記載されている。

【図面の簡単な説明】

【００４４】

本発明の上記特徴および有利な点も、その付加的な特徴および有利な点も、以下の図面と併せて本発明の実施形態の詳細な説明を読んだ後、さらに明確に理解できる。

【００４５】

【図１ａ】本発明の実施形態による、一またはそれ以上のサイトコントローラ（ＳｉｔｅＣ）と、多数の被試験デバイス（ＤＵＴ）との同時テストのためのオープンアーキテクチャテストシステムを示す。

【図１ｂ】オープンアーキテクチャテストシステムにおける多数のベンダハードウェアモ

ジュールバージョンおよびそれらの対応するソフトウェアコンポーネントバージョンの例を示す。
【図２】本発明の一実施形態による、マスクされた番号へのマスクの適用を示す。
【図３】本発明の一実施形態による、オープンアーキテクチャテストシステムにおける異なったベンダハードウェアモジュールバージョンをテストするための、システムプロファイルを生成するための方法を示す。
【図４】本発明の一実施例によるマスクされた表現の共通部分を計算するための方法を示す。