(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5111659号
(P5111659)
(45)発行日　平成25年1月9日(2013.1.9)　(24)登録日　平成24年10月19日(2012.10.19)

(51)Int.Cl.　F I
G06F 17/50 (2006.01)　G06F 17/50 652C

請求項の数 4　(全 16 頁)

(21)出願番号　特願2011-509120(P2011-509120)
(86)(22)出願日　平成21年4月15日(2009.4.15)
(86)国際出願番号　PCT/JP2009/057565
(87)国際公開番号　WO2010/119520
(87)国際公開日　平成22年10月21日(2010.10.21)
審査請求日　平成23年10月21日(2011.10.21)

(73)特許権者　302062931
ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部１７５３番地
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　塚本　美智子
神奈川県川崎市中原区下沼部１７５３番地
ルネサスエレクトロニクス株式会社内
(72)発明者　中島　隆
神奈川県川崎市中原区下沼部１７５３番地
ルネサスエレクトロニクス株式会社内
(72)発明者　宮西　篤史
神奈川県川崎市中原区下沼部１７５３番地
ルネサスエレクトロニクス株式会社内

最終頁に続く

(54)【発明の名称】マスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステムおよびマスクパターンを作成するための方法

(57)【特許請求の範囲】
【請求項１】
マスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステムであって、
第１の情報処理装置と、
第２の情報処理装置とを備え、
前記第１の情報処理装置は、
製造すべきマスクＲＯＭの設計パラメータを受付けるためのモジュールと、
前記設計パラメータに応じた第１のコードをランダムに生成するためのモジュールとを含み、前記第１のコードは、前記製造すべきマスクＲＯＭに記憶される第２のコードから独立して定められており、
前記第１のコードに対応する第１の設計情報を生成するためのモジュールと、
命令が格納されたプログラムファイルを出力するためのモジュールとを含み、前記プログラムファイルには、前記第１のコードおよび前記第１の設計情報との関連付けを示す識別情報が付与されており、
前記プログラムファイルに格納された前記命令が前記第２の情報処理装置によって実行されると、前記第２の情報処理装置は以下のモジュールを含むように構成され、前記以下のモジュールは、
前記第２のコードを受付けるためのモジュールと、
前記第１のコードおよび前記第１の設計情報に基づいて、前記第２のコードに対応す

る第２の設計情報を生成するためのモジュールと、
　前記識別情報に基づいて、関連付けられた前記第１のコードおよび前記第１の設計情報が存在しない場合には前記第２の設計情報の生成を禁止するモジュールとを含む、システム。
【請求項２】
　前記第１のコードを生成するためのモジュールは、前記第１のコードをランダムに決定する、請求項１に記載のシステム。
【請求項３】
　前記第１の情報処理装置と前記第２の情報処理装置とはデータ通信可能にネットワーク接続されており、
　前記第１の情報処理装置は、前記第１のコードと、前記第１の設計情報と、前記プログラムファイルとを、ネットワークを介して前記第２の情報処理装置へ配信する、請求項１に記載のシステム。
【請求項４】
　第１の情報処理装置および第２の情報処理装置を含むシステムを用いて、マスクＲＯＭの製造に向けられたレイアウトパターンを作成するための方法であって、
　前記第１の情報処理装置によって、製造すべきマスクＲＯＭの設計パラメータを受付けるステップと、
　前記第１の情報処理装置によって、前記設計パラメータに応じた第１のコードをランダムに生成するステップとを含み、前記第１のコードは、前記製造すべきマスクＲＯＭに記憶される第２のコードから独立して定められており、
　前記第１の情報処理装置によって、前記第１のコードに対応する第１の設計情報を生成するステップと、
　前記第１の情報処理装置によって、命令が格納されたプログラムファイルを出力するステップとを含み、前記プログラムファイルには、前記第１のコードおよび前記第１の設計情報との関連付けを示す識別情報が付与されており、
　前記第２の情報処理装置によって、前記プログラムファイルに格納された前記命令が実行されるステップと、
　前記第２の情報処理装置によって、前記第２のコードを受付けるステップと、
　前記第２の情報処理装置によって、前記第１のコードおよび前記第１の設計情報に基づいて、前記第２のコードに対応する第２の設計情報を生成するステップと、
　前記第２の情報処理装置によって、前記識別情報に基づいて、関連付けられた前記第１のコードおよび前記第１の設計情報が存在しない場合には前記第２の設計情報の生成を禁止するステップとを含む、方法。
【発明の詳細な説明】
【技術分野】
【０００１】
　本発明は、マスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステム、そのシステムを用いて製造されたマスクＲＯＭ、およびレイアウトパターンを作成するための方法に関する。
【背景技術】
【０００２】
　従来から、不揮発的に情報を記憶するデバイスとして、マスクＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）が知られている。このようなマスクＲＯＭは、半導体上に所定の回路パターンを形成することで、特定の情報（以下、「ＲＯＭコード」とも称す。）を記憶する。このようなマスクＲＯＭは、典型的には、半導体基板に対して、回路パターンを記述したフォトマスクを用いて露光処理を行なうことで製造される。したがって、このようなフォトマスクに描画すべき回路パターン（以下、「レイアウトパターン」とも称す。）を効率的に作成する必要がある。

このようなマスクＲＯＭのレイアウトパターンを作成する先行技術として、特開平０６－２１５０７０号公報（特許文献１）、特開平０６－１３９３０９号公報（特許文献２）、および特開平０５－１８９５２１号公報（特許文献３）などが知られている。

【０００３】

近年、情報通信技術における急速な進歩に伴って、このようなマスクＲＯＭについても、開発の短期間化、すなわちＴＡＴ（Ｔｕｒｎ　Ａｒｏｕｎｄ　Ｔｉｍｅ）の短縮が特に要求されている。一方で、マスクＲＯＭが搭載される製品の開発過程においては、マスクＲＯＭに記憶されるべきＲＯＭコードが頻繁に変更されることも多い。

【先行技術文献】

【特許文献】

【０００４】

【特許文献１】特開平０６－２１５０７０号公報

【特許文献２】特開平０６－１３９３０９号公報

【特許文献３】特開平０５－１８９５２１号公報

【発明の概要】

【発明が解決しようとする課題】

【０００５】

従来のレイアウトパターンを作成するシステムにおいては、ＲＯＭコードが変更されるたびに、レイアウトパターンをすべて作成し直すといった作業が発生していた。なお、特定のマスクＲＯＭにおいては、上述の特開平０５－１８９５２１号公報（特許文献３）に開示されるレイアウトパターン発生装置を用いることで、ＴＡＴを短縮できる場合もある。しかしながら、同一の半導体基板上にプロセッサおよびメモリが搭載されるＳＯＣ（Ｓｉｌｉｃｏｎ　Ｏｎ　ａ　Ｃｈｉｐ）などにおいては、マスクＲＯＭの配置位置自体が頻繁に変更されるため、特開平０５－１８９５２１号公報（特許文献３）に開示されるレイアウトパターン発生装置をそのまま適用することはできなかった。

【０００６】

また、近年、情報セキュリティ上の観点から、ＲＯＭコードの機密性を保ちたいという要求も高まっている。そこで、本来のＲＯＭコードとは異なる暫定的なＲＯＭコードを用いて、装置の設計開発を行なった上で、最終工程で本来のＲＯＭコードに変更し、レイアウトパターンを決定するといった工程が採用される場合も多くなってきている。

【０００７】

しかしながら、従来のレイアウトパターンを作成するシステムにおいては、上述のような工程に十分に対応できるものではなかった。

【０００８】

この発明は、このような課題を解決するためになされたものであって、その目的は、マスクＲＯＭに記憶すべきコードの機密性を保持しつつマスクＲＯＭのレイアウトパターンを作成することが可能なシステムを提供することである。また、この発明の別の目的は、このようなシステムによって製造されるマスクＲＯＭを提供することである。さらに、この発明のさらに別の目的は、マスクＲＯＭに記憶すべきコードの機密性を保持しつつマスクＲＯＭのレイアウトパターンを作成することが可能なレイアウトパターンを作成するための方法を提供することである。

【課題を解決するための手段】

【０００９】

この発明のある局面に従えば、マスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステムを提供する。本システムは、第１の情報処理装置と、第２の情報処理装置とを含む。第１の情報処理装置は、製造すべきマスクＲＯＭの設計パラメータを受付けるためのモジュールと、第１のコードを生成するためのモジュールとを含む。第１のコードは、製造すべきマスクＲＯＭに記憶される第２のコードから独立して定められている。本システムは、さらに、第１のコードに対応する第１の設計情報を生成するためのモジュールと、命令が格納されたプログラムファイルを出力するためのモジュールとを含

む。プログラムファイルには、第１のコードおよび第１の設計情報との関連付けを示す識別情報が付与されている。プログラムファイルに格納された命令が第２の情報処理装置によって実行されると、第２の情報処理装置は以下のモジュールを含むように構成される。以下のモジュールは、第２のコードを受付けるためのモジュールと、第１のコードおよび第１の設計情報に基づいて、第２のコードに対応する第２の設計情報を生成するためのモジュールと、識別情報に基づいて、関連付けられた第１のコードおよび第１の設計情報が存在しない場合には第２の設計情報の生成を禁止するモジュールとを含む。

【００１０】

好ましくは、第１のコードを生成するためのモジュールは、第１のコードをランダムに決定する。

【００１１】

好ましくは、第１の情報処理装置と第２の情報処理装置とはデータ通信可能にネットワーク接続されており、第１の情報処理装置は、第１のコードと、第１の設計情報と、プログラムファイルとを、ネットワークを介して第２の情報処理装置へ配信する。

【００１２】

この発明の別の局面に従えば、上述のシステムを用いて製造されたマスクＲＯＭを提供する。

【００１３】

この発明のさらに別の局面に従えば、第１の情報処理装置および第２の情報処理装置を含むシステムを用いて、マスクＲＯＭの製造に向けられたレイアウトパターンを作成するための方法を提供する。本方法は、第１の情報処理装置によって、製造すべきマスクＲＯＭの設計パラメータを受付けるステップと、第１の情報処理装置によって、第１のコードを生成するステップとを含む。第１のコードは、製造すべきマスクＲＯＭに記憶される第２のコードから独立して定められている。本方法は、さらに、第１の情報処理装置によって、第１のコードに対応する第１の設計情報を生成するステップと、第１の情報処理装置によって、命令が格納されたプログラムファイルを出力するステップとを含む。プログラムファイルには、第１のコードおよび第１の設計情報との関連付けを示す識別情報が付与されている。本方法は、さらに、第２の情報処理装置によって、プログラムファイルに格納された命令が実行されるステップと、第２の情報処理装置によって、第２のコードを受付けるステップと、第２の情報処理装置によって、第１のコードおよび第１の設計情報に基づいて、第２のコードに対応する第２の設計情報を生成するステップと、第２の情報処理装置によって、識別情報に基づいて、関連付けられた第１のコードおよび第１の設計情報が存在しない場合には第２の設計情報の生成を禁止するステップとを含む。

【発明の効果】

【００１４】

この発明によれば、マスクＲＯＭに記憶すべきコードの機密性を保持しつつマスクＲＯＭのレイアウトパターンを作成することができる。

【図面の簡単な説明】

【００１５】

【図１】この発明の実施の形態に従う一般的な製品設計フローを示す概略図である。

【図２】この発明の実施の形態に従うシステムの概略構成を示す図である。

【図３】図２に示すホストサーバおよびワークステーションのハードウェア構成を示す概略構成図である。

【図４】この発明の実施の形態に従うシステムにおける処理の概要を説明するための図である。

【図５】この発明の実施の形態に従うホストサーバの機能ブロック図である。

【図６】この発明の実施の形態に従うワークステーションの機能ブロック図である。

【図７】この発明の実施の形態に従うシステムにおける処理手順を示すフローチャートである。

【発明を実施するための形態】

【００１６】
この発明の実施の形態について、図面を参照しながら詳細に説明する。なお、図中の同一または相当部分については、同一符号を付してその説明は繰返さない。
【００１７】
＜製品設計フロー＞
まず、本実施の形態に従うマスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステムについて説明する前に、その位置付けについての理解を容易にするために、半導体チップを含む一般的な製品設計フローについて説明する。なお、図１に示す製品設計フローは、顧客から特定の半導体チップについての要求を受けて、チップメーカがマスクＲＯＭを含めた一連の半導体チップを設計開発する場合の処理手順を示す。
【００１８】
図１は、この発明の実施の形態に従う一般的な製品設計フローを示す概略図である。図１を参照して、まず、顧客が対象となる半導体チップを含めた製品全体についてのシステム設計を行なう（ステップＳ２）。より具体的には、入出力データの内容や論理の内容などが決定される。
【００１９】
このシステム設計が完了すると、顧客もしくはチップメーカが、論理設計および検証を行なう（ステップＳ４）。この工程は、レジスタ転送レベル（ＲＴＬ：Ｒｅｇｉｓｔｅｒ　Ｔｒａｎｓｆｅｒ　Ｌｅｖｅｌ）とも称され、ハードウェアレジスタとブール論理回路との間のデータの授受に着目して、対象とする半導体チップにおける回路動作が設計される。続いて、チップメーカが、複数の論理設計を１つの半導体チップにまとめる論理合成を行なう（ステップＳ６）。
【００２０】
続いて、チップメーカが、ステップＳ６において合成された論理に応じたテスト設計を行なう（ステップＳ８）。より具体的には、スキャン手法やＢＩＳＴ（Ｂｕｉｌｔ　Ｉｎ　Ｓｅｌｆ－Ｔｅｓｔ）手法などを用いて、故障検出率の高いテストパターンを設計する。続いて、チップメーカが、基板上のレイアウトを決定するレイアウト設計を行なう（ステップＳ１０）。
【００２１】
上述のような一連の設計が完了すると、チップメーカは、サインオフ検証を行なう（ステップＳ１２）。このサインオフ検証では、最終的な論理機能およびタイミングが検証される。典型的には、このサインオフ検証には、スタティックタイミング解析（ＳＴＡ：Ｓｔａｔｉｃ　Ｔｉｍｉｎｇ　Ａｎａｌｙｓｉｓ）、シグナルインテグリティ（Ｓｉｇｎａｌ　Ｉｎｔｅｇｒｉｔｙ）解析、および実負荷シミュレーションなどが用いられる。
【００２２】
サインオフ検証が完了すると、チップメーカは、レイアウト検証を行なう（ステップＳ１４）。このレイアウト検証では、マスクのレイアウトパターンが検証される。典型的には、このレイアウト検証では、ＤＲＣ（Ｄｅｓｉｇｎ　Ｒｕｌｅ　Ｃｈｅｃｋｉｎｇ）検証と、ＬＶＳ（Ｌａｙｏｕｔ　Ｖｅｒｓｕｓ　Ｓｃｈｅｍａｔｉｃ）検証とが実行される。ＤＲＣ検証では、製造装置の制約から決まる幾何学的な設計ルールを満足しているかどうかが検証される。また、ＬＶＳ検証では、論理設計および回路設計段階で作成されたエレメントやエレメント間の電気的な接続が、レイアウト設計で正しく実装されているかが検証される。
【００２３】
上述のような検証工程を経て、実際のフォトマスクが作成される（ステップＳ１６）。
ところで、上述のステップＳ２～Ｓ１４までの工程は、コンピュータ上で実現されることになる。これらの工程のうちステップＳ６～Ｓ１４については、予め部品ライブラリＬＩＢを用意しておき、この部品ライブラリＬＩＢを逐次参照しながら、各種の処理が実行される。この部品ライブラリＬＩＢは、メモリに関するものとして、ＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）ライブラリおよびＲＯＭライブラリを含む。

【００２４】
このようなメモリに関する設計手順として、まず、顧客もしくはチップメーカが、メモリ種類の検討を行なう（ステップＳ３０）。続いて、顧客もしくはチップメーカが、必要なメモリの種類毎にその容量などを定めるメモリアサインを行なう（ステップＳ３２）。続いて、顧客もしくはチップメーカが、定められた設計パラメータに基づいて、ＲＡＭコンパイラおよびＲＯＭコンパイラを用いて、ＲＡＭライブラリおよびＲＯＭライブラリをそれぞれ生成する（ステップＳ３４およびＳ３６）。
【００２５】
＜概要＞
本実施の形態に従うマスクＲＯＭの製造に向けられたレイアウトパターンを作成するためのシステムは、主として、図１に示すステップＳ３６のＲＯＭライブラリに関する処理に向けられる。より具体的には、本実施の形態に従うシステムにおいては、顧客の保有するＲＯＭコードの機密性を高めたいという要求などに応じるために、チップメーカによるサインオフ検証（図１に示すステップＳ１２）やレイアウト検証（図１に示すステップＳ１４）を暫定的なＲＯＭコード（以下、「仮ＲＯＭコード」とも称す。）を用いて行なうとともに、一連の設計および検証が完了した時点で、仮ＲＯＭコードを顧客の保有する本来のＲＯＭコード（以下、「本ＲＯＭコード」とも称す。）に変更される。そして、この変更されたＲＯＭコードに基づいて設計された情報に基づいて、フォトマスクが作成される。
【００２６】
このような構成を採用することで、本ＲＯＭコードについての顧客以外の者に対する情報セキュリティ機能を高めることができる。さらに、ＲＯＭコードが変更されても、既に行なわれた製品設計フローの工程をやり直す必要がないため、コード変更に伴うＴＡＴを短縮することができる。
【００２７】
＜システム構成＞
図２は、この発明の実施の形態に従うシステムの概略構成を示す図である。図２を参照して、本実施の形態に従うシステムは、典型的には、チップメーカ側に設けられたホストサーバＳＲＶおよびワークステーションＷＳ１と、顧客側に設けられたワークステーションＷＳ２－１，ＷＳ２－２，・・・とを含む。ホストサーバＳＲＶと、ワークステーションＷＳ１，ＷＳ２－１，ＷＳ２－２，・・・（以下、「ワークステーションＷＳ」とも総称する。）とは、互いにデータ通信可能に接続された情報処理装置である。なお、ホストサーバＳＲＶとワークステーションＷＳ１との間は、典型的には、社内ＬＡＮを介してデータ通信可能に接続されており、ホストサーバＳＲＶとワークステーションＷＳ２－１，ＷＳ２－２，・・・との間は、典型的には、インターネットなどのネットワークＮＷを介してデータ通信可能に接続されている。なお、ワークステーションＷＳ２－１，ＷＳ２－２，・・・の各々は、互いに独立した顧客側に設けられることが想定されている。代替的に、チップメーカ自身が半導体チップを含む製品などを直接に製造販売する場合などには、顧客側にワークステーションＷＳが存在しなくともよい。
【００２８】
＜ハードウェア構成＞
図３は、図２に示すホストサーバＳＲＶおよびワークステーションＷＳのハードウェア構成を示す概略構成図である。図３には、ホストサーバＳＲＶおよびワークステーションＷＳが汎用的なコンピュータによって実現される構成を示すが、専用のハードウェアを用いてもよい。
【００２９】
図３を参照して、ホストサーバＳＲＶは、本体部１０１と、表示部としてのモニタ１０２と、入力部としてのキーボード１０３およびマウス１０４とを含む。モニタ１０２、キーボード１０３およびマウス１０４は、本体部１０１に接続される。
【００３０】

本体部１０１は、演算部としてのＣＰＵ（Ｃｅｎｔｒａｌ Ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）１０５と、記憶部としてのメモリ１０６および固定ディスク１０７と、通信インターフェース１０９と、データ読出装置としてのＦＤ（Ｆｌｅｘｉｂｌｅ Ｄｉｓｋ）駆動装置１１１およびＣＤ－ＲＯＭ（Ｃｏｍｐａｃｔ Ｄｉｓｋ－Ｒｅａｄ Ｏｎｌｙ Ｍｅｍｏｒｙ）駆動装置１１３とを含む。これらの各部は、バスを介して相互接続されている。

【００３１】

ホストサーバＳＲＶでは、典型的には、ＣＰＵ１０５がメモリ１０６などのコンピュータハードウェアを用いてプログラムを実行することで、後述する各種機能が提供される。一般的に、このようなプログラムは、ＦＤ１１２やＣＤ－ＲＯＭ１１４などの記憶媒体に格納されて、またはネットワークなどを介して流通する。そして、このようなプログラムは、ＦＤ駆動装置１１１やＣＤ－ＲＯＭ駆動装置１１３などにより記憶媒体から読取られて、または通信インターフェース１０９にて受信されて、固定ディスク１０７に格納される。さらに、このようなプログラムは、固定ディスク１０７からメモリ１０６に読出されて、ＣＰＵ１０５により実行される。

【００３２】

なお、ＣＰＵ１０５が、コンピュータのオペレーティングシステム（ＯＳ）の一部として提供されるプログラムモジュールのうち、必要なモジュールを所定の配列および／またはタイミングで呼出して処理を実行することで、ホストサーバＳＲＶにより提供される機能の全部もしくは一部を実現する場合もある。このような場合には、本発明に係るプログラム自身には上記のようなモジュールが含まれることなく、代替的に、ＯＳによって提供される上記のようなモジュールを利用するための命令を含む。

【００３３】

ＣＰＵ１０５は、各種の数値論理演算を行なうプロセッサであり、プログラムされた命令を順次実行することで、後述するような処理を実行する。メモリ１０６は、ＣＰＵ１０５のプログラム実行に応じて、各種の情報を格納する。

【００３４】

モニタ１０２は、ＣＰＵ１０５が出力する各種情報を表示する。モニタ１０２は、一例として、ＬＣＤ（Ｌｉｑｕｉｄ Ｃｒｙｓｔａｌ Ｄｉｓｐｌａｙ）やＣＲＴ（Ｃａｔｈｏｄｅ Ｒａｙ Ｔｕｂｅ）などからなる。

【００３５】

マウス１０４は、クリックやスライドなどの動作に応じたユーザから指令を受付ける。キーボード１０３は、入力されるキーに応じたユーザから指令を受付ける。

【００３６】

通信インターフェース１０９は、ワークステーションＷＳまたは他のホストサーバＳＲＶなどとの間でデータ通信を行なう。

【００３７】

ワークステーションＷＳのハードウェア構成についても、ホストサーバＳＲＶと同様であるので、詳細な説明は繰返さない。

【００３８】

＜全体機能構成＞

図４は、この発明の実施の形態に従うシステムにおける処理の概要を説明するための図である。なお、図４には、チップメーカの設計部門および顧客のいずれもが設計開発に携わる場合について例示する。

【００３９】

図４を参照して、まず、ホストサーバＳＲＶは、製造すべきマスクＲＯＭの設計パラメータ３２を受付ける。すなわち、顧客は、自身の要求するマスクＲＯＭについての、（１）ＲＯＭタイプ（種類）、および（２）Ｗｏｒｄ数／Ｂｉｔ数などを、ホストサーバＳＲＶ（チップメーカ）へ伝える。ホストサーバＳＲＶでは、ＲＯＭコンパイラ１０が実行可能にインストールされている。このＲＯＭコンパイラ１０が実行されることで、入力され

た設計パラメータ３２に基づいて、仮ＲＯＭコードファイル２４が生成される。この仮ＲＯＭコードファイル２４には、顧客がマスクＲＯＭに記憶させるべきＲＯＭコード（正ＲＯＭコード）とは独立して定められる仮ＲＯＭコードが格納される。この仮ＲＯＭコードは、ランダムに決定されることが好ましい。これは、サインオフ検証やレイアウト検証において、検証カバレッジを向上させるためである。すなわち、「０」もしくは「１」の連続するようなビット列が仮ＲＯＭコードとして生成されると、論理設計上のエラー、レイアウト上のエラー、および回路接続上のエラーなどを発見できない場合があるためである。

【００４０】

より具体的には、仮ＲＯＭコードがすべて「０」もしくは「１」であれば、異なるアドレスから出力される結果が同じになってしまい、検証カバレッジを向上させることができない。これに対して、仮ＲＯＭコードをランダムに設定することで、アドレスが異なれば、出力の期待値が異なるため、検証カバレッジを向上させることができる。

【００４１】

また、仮ＲＯＭコードがすべて「０」もしくは「１」であれば、ＲＯＭコードに対応するレイアウトが画一化（コンタクトが存在するもののみ、あるいはコンタクトが存在しないもののみ）する。そのため、コンタクトの有無の組み合わせを利用したレイアウト上のエラーを発見できない場合がある。これに対して、仮ＲＯＭコードをランダムに設定することで、レイアウトの組み合わせが多様化するため、レイアウト上のエラーの発見確率を高めることができる。

【００４２】

同様に、仮ＲＯＭコードがすべて「０」もしくは「１」であれば、ネットリストが画一化するため、回路接続上のエラーを発見できない場合がある。これに対して、仮ＲＯＭコードをランダムに設定することで、ネットリストが多様化するため、回路接続上のエラーの発見確率を高めることができる。

【００４３】

なお、この仮ＲＯＭコードは、典型的には、論理シミュレータで用いられるハードウェア記述言語（Ｖｅｒｉｌｏｇ　ＨＤＬ）で記述される（図４に示す「Ｖｅｒｉｌｏｇモデル」）。この仮ＲＯＭコードファイル２４の内容の一例は、以下のようになる。

【００４４】

```
／／ＢＲＭＡ２４Ｐ２：８ｂｉｔｓ，１０２４ｗｏｒｄｓ
@０　ＦＥ
@１　ＡＥ
@２　ＢＢ
・・・
```

上述の記述の１行目には、仮ＲＯＭコードファイル２４に与えられるモジュール名、Ｂｉｔ数、Ｗｏｒｄ数が定義される。続く２行目以降からは、「@」に引き続く数字が記憶されるべきマスクＲＯＭ上のアドレスを示し、さらにそれに続く「ＦＥ」といった記述が記憶されるべきマスクＲＯＭコードを示す。

【００４５】

また、ホストサーバＳＲＶは、生成された仮ＲＯＭコードに対応する設計情報ファイル２６を生成する。この設計情報ファイル２６は、エレメントについての電気的な接続状態を記述した回路設計データ（ＣＤＬ：Ｃｉｒｃｕｉｔ　Ｄｅｓｉｇｎ　ｌａｎｇｕａｇｅ）と、エレメントについての幾何学的位置を記述したレイアウト設計データ（ＧＤＳＩＩ：Ｇｒａｐｈｉｃａｌ　Ｄａｔａ　ｓｙｓｔｅｍ　ＩＩ）とを含む。

【００４６】

さらに、ホストサーバＳＲＶは、上述の仮ＲＯＭコードファイル２４および設計情報ファイル２６の生成に伴って、専用ＲＯＭコンパイラ２２および専用ＲＯＭコンパイラ２２に関連付けられた中間ファイル２８を生成する。この専用ＲＯＭコンパイラ２２は、後述するように、ワークステーションＷＳ上で、正ＲＯＭコードファイル４４、および正ＲＯ

Ｍコードに対応する設計情報ファイル４６を生成するための一種のプログラムである。また、中間ファイル２８は、対応する専用ＲＯＭコンパイラ２２と同時に生成された仮ＲＯＭコードファイル２４および設計情報ファイル２６との間の関連付けを示す。すなわち、中間ファイル２８に含まれる識別情報は、専用ＲＯＭコンパイラ２２がワークステーションＷＳ上で実行された際に、その内容を変更可能な範囲を、当該専用ＲＯＭコンパイラ２２と同時に生成された特定の仮ＲＯＭコードファイル２４および設計情報ファイル２６のみに限定するために用いられる。

【００４７】

上述のように生成されるファイルセット２０（専用ＲＯＭコンパイラ２２、中間ファイル２８、仮ＲＯＭコードファイル２４、および設計情報ファイル２６）は、設計パラメータ３２の入力元に対応するワークステーションＷＳへネットワークを介して配信される。このファイルセット２０は、典型的には、チップメーカの設計部門に設けられるワークステーションＷＳおよび／または顧客側に設けられるワークステーションＷＳへ配信される。

【００４８】

チップメーカの設計部門に設けられるワークステーションＷＳでは、ホストサーバＳＲＶから配信された設計情報ファイル４６に基づいて、サインオフ検証やレイアウト検証などが実行される。

【００４９】

一方、顧客側に設けられたワークステーションＷＳでは、専用ＲＯＭコンパイラ２２が実行され、設計情報ファイル２６の内容を正ＲＯＭコードに応じたものに変更される。すなわち、専用ＲＯＭコンパイラ２２がワークステーションＷＳによって実行されることで、ワークステーションＷＳは、正ＲＯＭコード４２を受付けるとともに、ホストサーバＳＲＶから受信した仮ＲＯＭコードに対応する設計情報ファイル２６を、正ＲＯＭコード４２に対応する設計情報ファイル４６に変更して出力する。このとき、専用ＲＯＭコンパイラ２２を実行するワークステーションＷＳは、正ＲＯＭコードファイル４４を同時に出力してもよい。

【００５０】

また、専用ＲＯＭコンパイラ２２を実行するワークステーションＷＳは、当該専用ＲＯＭコンパイラ２２に関連付けられた仮ＲＯＭコードファイル２４および設計情報ファイル２６が存在しない場合には、上述のような設計情報ファイル４６の生成を禁止する。すなわち、専用ＲＯＭコンパイラ２２は、特定の設計パラメータ３２および仮ＲＯＭコード（仮ＲＯＭコードファイル２４の内容）に関連付けられた設計情報ファイル２６のみを変更可能なように専用化される。なお、専用ＲＯＭコンパイラ２２を実行するワークステーションＷＳは、実行中の専用ＲＯＭコンパイラ２２に付与された中間ファイル２８に含まれる識別情報を参照して、対象の設計情報ファイル２６を特定する。このような専用化処理によって、異なるパラメータ、異なるバージョン、および異なる正ＲＯＭコードに基づく設計情報ファイル４６が生成されてしまうことを回避できる。

【００５１】

このような専用化を行なう典型的な手法として、本実施の形態に従うシステムにおいては、上述の中間ファイル２８に含まれる識別情報が用いられる。より具体的には、中間ファイル２８には、対応する専用ＲＯＭコンパイラ２２に関連付けられたモジュール名のリストが記述される。また、仮ＲＯＭコードファイル２４および設計情報ファイル２６についても、それぞれ生成時のモジュール名が付与されている。なお、このモジュール名は、本実施の形態に従うシステムにおいてユニークな値であることが好ましい。そして、ワークステーションＷＳが専用ＲＯＭコンパイラ２２を実行すると、まず、対応する中間ファイル２８が参照され、その中に記述されているモジュール名に対応する仮ＲＯＭコードファイル２４および設計情報ファイル２６が存在するか否かが判断される。そして、両方のファイルが存在している場合に限って、上述の正ＲＯＭコード４２の受付処理や設計情報ファイル４６の生成処理が実行可能になる。

【００５２】
＜情報セキュリティ＞
上述したように、ホストサーバＳＲＶにインストールされているＲＯＭコンパイラ１０から分離して、専用ＲＯＭコンパイラ２２を各ワークステーションＷＳへ配信することで、情報セキュリティ機能を高めることができる。より具体的な形態としては、ホストサーバＳＲＶをチップメーカ側に設ける構成に代えて、ベンダー側に設けてもよい。そして、正ＲＯＭコードは、チップメーカ側のセキュアな領域もしくは顧客側に格納することが好ましい。
【００５３】
すなわち、ＲＯＭコンパイラ１０および正ＲＯＭコードの格納場所を物理的に隔離することで、正ＲＯＭコードについての情報セキュリティ機能を高めることができる。
【００５４】
＜機能ブロック図＞
図５は、この発明の実施の形態に従うホストサーバＳＲＶの機能ブロック図である。図６は、この発明の実施の形態に従うワークステーションＷＳの機能ブロック図である。
【００５５】
図５を参照して、ホストサーバＳＲＶは、その制御構造として、入力モジュール２０２と、仮コード生成モジュール２０４と、言語変換モジュール２０６と、中間ファイル生成モジュール２０８と、専用ＲＯＭコンパイラ出力モジュール２１０と、レイアウト設計データ生成モジュール２１２と、回路設計データ生成モジュール２１４と、配信モジュール２１６とを含む。
【００５６】
入力モジュール２０２は、製造すべきマスクＲＯＭについてのＲＯＭタイプ（種類）やＷｏｒｄ数／Ｂｉｔ数といった設計パラメータを受付ける。
【００５７】
仮コード生成モジュール２０４は、入力モジュール２０２を介して設定された設計パラメータに応じたＲＯＭコードをランダムに決定し、仮ＲＯＭコードとして出力する。
【００５８】
言語変換モジュール２０６は、仮コード生成モジュール２０４によって生成された仮ＲＯＭコードをハードウェア記述言語（Ｖｅｒｉｌｏｇ　ＨＤＬ）で示されるデータに変換し、仮ＲＯＭコードファイルとして、配信モジュール２１６へ出力する。
【００５９】
中間ファイル生成モジュール２０８は、仮ＲＯＭコードファイル２４に与えられるモジュール名といった識別情報を含む中間ファイルを生成する。専用ＲＯＭコンパイラ出力モジュール２１０は、中間ファイル生成モジュール２０８によって生成された中間ファイルを付与した専用ＲＯＭコンパイラを配信モジュール２１６へ出力する。
【００６０】
レイアウト設計データ生成モジュール２１２は、仮コード生成モジュール２０４によって生成された仮ＲＯＭコードに対応するレイアウト設計データを生成し、配信モジュール２１６へ出力する。また、回路設計データ生成モジュール２１４は、仮コード生成モジュール２０４によって生成された仮ＲＯＭコードに対応する回路設計データを生成し、配信モジュール２１６へ出力する。
【００６１】
配信モジュール２１６は、いずれかのワークステーションＷＳからの要求に応答して、上述の各部で生成された、仮ＲＯＭコードファイル、専用ＲＯＭコンパイラ、中間ファイル、レイアウト設計データ、および回路設計データを配信する。
【００６２】
ワークステーションＷＳは、ホストサーバＳＲＶから配信された専用ＲＯＭコンパイラを実行することで、図６に示すような制御構造を実現する。図６を参照して、ワークステーションＷＳは、その制御構造として、入力モジュール３０２と、言語変換モジュール３

０４と、レイアウト設計データ変更モジュール３０６と、回路設計データ変更モジュール３０８と、監視モジュール３１０とを含む。

【００６３】

入力モジュール３０２は、製造すべきマスクＲＯＭに記憶すべき正ＲＯＭコードを受付ける。

【００６４】

言語変換モジュール３０４は、入力モジュール３０２を介して入力される正ＲＯＭコードをハードウェア記述言語（Ｖｅｒｉｌｏｇ　ＨＤＬ）で示されるデータに変換し、正ＲＯＭコードファイルとして出力する。

【００６５】

レイアウト設計データ変更モジュール３０６は、ホストサーバＳＲＶから受信したレイアウト設計データに対して、入力モジュール３０２を介して入力される正ＲＯＭコードに応じた変更を行なうことで、正ＲＯＭコードに対応するレイアウト設計データを生成する。同様に、回路設計データ変更モジュール３０８は、ホストサーバＳＲＶから受信した回路設計データに対して、入力モジュール３０２を介して入力される正ＲＯＭコードに応じた変更を行なうことで、正ＲＯＭコードに対応する回路設計データを生成する。

【００６６】

監視モジュール３１０は、対応するワークステーションＷＳにおいて実行中の専用ＲＯＭコンパイラに関連付けられた仮ＲＯＭコードファイルおよび当該仮ＲＯＭコードに対応する設計情報ファイル（レイアウト設計データおよび回路設計データ）の存在を監視する。そして、いずれかのファイルが存在しない場合には、言語変換モジュール３０４、レイアウト設計データ変更モジュール３０６、および回路設計データ変更モジュール３０８における生成処理を禁止する。

【００６７】

＜処理手順＞

図７は、この発明の実施の形態に従うシステムにおける処理手順を示すフローチャートである。図７に示すステップは、ホストサーバＳＲＶまたはワークステーションＷＳのＣＰＵによって実行される。

【００６８】

図７を参照して、ホストサーバＳＲＶのＣＰＵ１０５は、製造すべきマスクＲＯＭについてのＲＯＭタイプ（種類）やＷｏｒｄ数／Ｂｉｔ数といった設計パラメータを受付ける（ステップＳ１００）。このとき、顧客側もしくはチップメーカ側のユーザが、ホストサーバＳＲＶに対して、所望の設計パラメータを入力する。

【００６９】

続いて、ホストサーバＳＲＶのＣＰＵ１０５は、入力された設計パラメータに応じたＲＯＭコードをランダムに決定し、仮ＲＯＭコードファイルとして出力する（ステップＳ１０２）。このとき、ホストサーバＳＲＶのＣＰＵ１０５は、仮ＲＯＭコードファイルに与えるモジュール名を予め決定する。さらに、ホストサーバＳＲＶのＣＰＵ１０５は、仮ＲＯＭコードファイルに与えられるモジュール名などの識別情報を含む中間ファイルを生成する（ステップＳ１０４）。

【００７０】

また、ホストサーバＳＲＶのＣＰＵ１０５は、ステップＳ１０２において決定した仮ＲＯＭコードに対応するレイアウト設計データを生成する（ステップＳ１０６）とともに、仮ＲＯＭコードに対応する回路設計データを生成する（ステップＳ１０８）。

【００７１】

最終的に、ホストサーバＳＲＶのＣＰＵ１０５は、ステップＳ１０２において生成した仮ＲＯＭコードファイル、ステップＳ１０６において生成したレイアウト設計データ、ステップＳ１０８において生成した回路設計データ、ステップＳ１０４において生成した中間ファイル、および専用ＲＯＭコンパイラをワークステーションＷＳへ配信する（ステップＳ１１０）。

【００７２】
ワークステーションＷＳのＣＰＵ１０５は、仮ＲＯＭコードファイルなどを含むデータセットをホストサーバＳＲＶから受信すると、それらをメモリ１０６または固定ディスク１０７に格納する（ステップＳ１１２）。
【００７３】
続いて、ワークステーションＷＳのＣＰＵ１０５は、専用ＲＯＭコンパイラの実行が指示されたか否かを判断する（ステップＳ１１４）。専用ＲＯＭコンパイラの実行が指示されていなければ（ステップＳ１１４においてＮＯ）、ステップＳ１１４の処理が繰返される。
【００７４】
一方、専用ＲＯＭコンパイラの実行が指示されると（ステップＳ１１４においてＹＥＳ）、ワークステーションＷＳのＣＰＵ１０５は、実行中の専用ＲＯＭコンパイラに関連付けられた仮ＲＯＭコードファイルおよび当該仮ＲＯＭコードに対応する設計情報ファイル（レイアウト設計データおよび回路設計データ）が存在しているか否かを判断する（ステップＳ１１６）。実行中の専用ＲＯＭコンパイラに関連付けられた仮ＲＯＭコードファイルおよび当該仮ＲＯＭコードに対応する設計情報ファイルのいずれかが存在していなければ（ステップＳ１１６においてＮＯ）、専用ＲＯＭコンパイラの実行は中止される。
【００７５】
これに対して、実行中の専用ＲＯＭコンパイラに関連付けられた仮ＲＯＭコードファイルおよび当該仮ＲＯＭコードに対応する設計情報ファイルがいずれも存在していれば（ステップＳ１１６においてＹＥＳ）、ワークステーションＷＳのＣＰＵ１０５は、正ＲＯＭコードが入力されたか否かを判断する（ステップＳ１１８）。正ＲＯＭコードが入力されていなければ（ステップＳ１１８においてＮＯ）、ステップＳ１１８の処理が繰返される。
【００７６】
これに対して、正ＲＯＭコードが入力されると（ステップＳ１１８においてＹＥＳ）、ワークステーションＷＳのＣＰＵ１０５は、入力された正ＲＯＭコードをハードウェア記述言語で示されるデータに変換し、正ＲＯＭコードファイルとして出力する（ステップＳ１２０）。続いて、ワークステーションＷＳのＣＰＵ１０５は、ホストサーバＳＲＶから受信したレイアウト設計データに対して、入力された正ＲＯＭコードに応じた変更を行なうことで、正ＲＯＭコードに対応するレイアウト設計データを生成する（ステップＳ１２２）。また、ワークステーションＷＳのＣＰＵ１０５は、ホストサーバＳＲＶから受信した回路設計データに対して、入力された正ＲＯＭコードに応じた変更を行なうことで、正ＲＯＭコードに対応する回路設計データを生成する（ステップＳ１２４）。そして、処理は終了する。
【００７７】
＜作用効果＞
この発明の実施の形態に従うシステムによれば、正ＲＯＭコードに対応する設計情報ファイルを生成するコンパイラが、製造すべきマスクＲＯＭの設計パラメータに応じた設計情報ファイルを生成するコンパイラから分離される。すなわち、本ＲＯＭコードを保有するユーザ以外の開発者は正ＲＯＭコードにアクセスすることなく、一連の設計開発を行なうことができる。そのため、本ＲＯＭコードに対する情報セキュリティ機能を高めることができる。
【００７８】
また、この発明の実施の形態に従うシステムによれば、ＲＯＭコードが変更されたとしても、変更前に行なわれた工程をやり直す必要がない。そのため、ＲＯＭコード変更時のＴＡＴを短縮することができる。
【００７９】
今回開示された実施の形態は、すべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて請求の範囲によって示され、

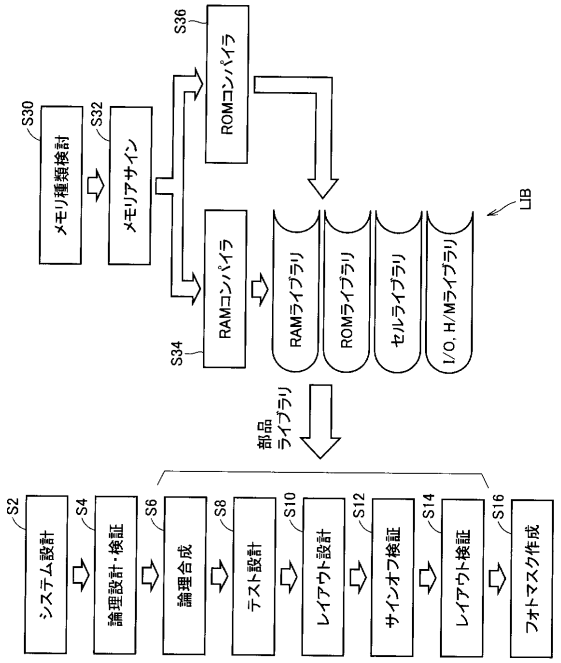
請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【符号の説明】

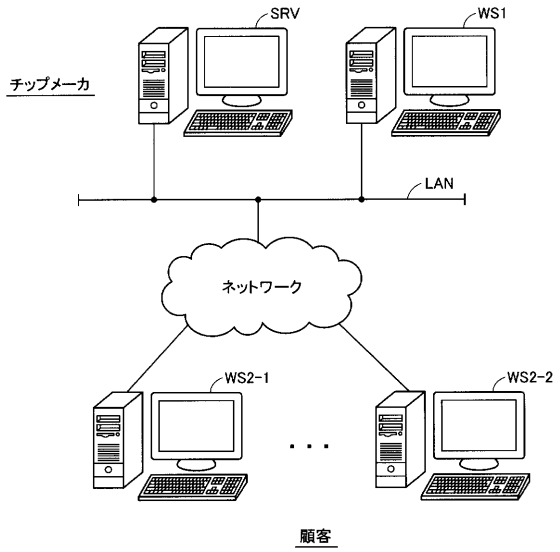
【００８０】

１０　ＲＯＭコンパイラ、２０　ファイルセット、２２　専用ＲＯＭコンパイラ、２４　仮ＲＯＭコードファイル、２６　設計情報ファイル、２８　中間ファイル、３２　設計パラメータ、４２　正ＲＯＭコード、４４　正ＲＯＭコードファイル、４６　設計情報ファイル、１０１　本体部、１０２　モニタ、１０３　キーボード、１０４　マウス、１０６　メモリ、１０７　固定ディスク、１０９　通信インターフェース、１１１　ＦＤ駆動装置、１１３　ＣＤ－ＲＯＭ駆動装置、２０２　入力モジュール、２０４　仮コード生成モジュール、２０６　言語変換モジュール、２０８　中間ファイル生成モジュール、２１０　コンパイラ出力モジュール、２１２　レイアウト設計データ生成モジュール、２１４　回路設計データ生成モジュール、２１６　配信モジュール、３０２　入力モジュール、３０４　言語変換モジュール、３０６　レイアウト設計データ変更モジュール、３０８　回路設計データ変更モジュール、３１０　監視モジュール、ＬＩＢ　部品ライブラリ、ＮＷ　ネットワーク、ＳＲＶ　ホストサーバ、ＷＳ，ＷＳ１，ＷＳ２　ワークステーション。

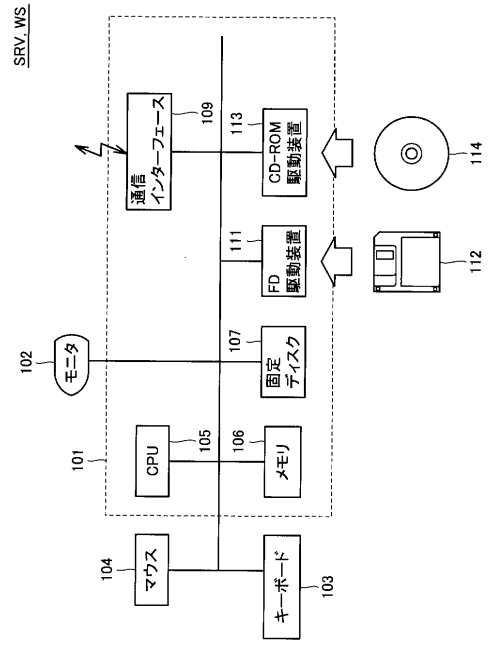
【図１】



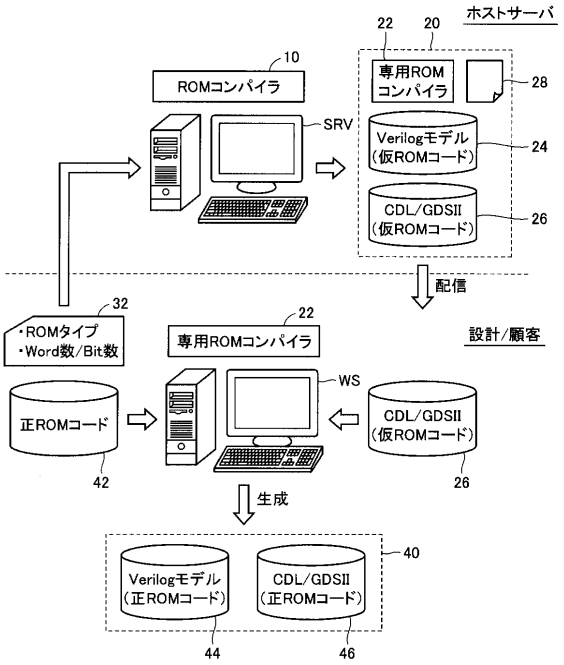
【図２】



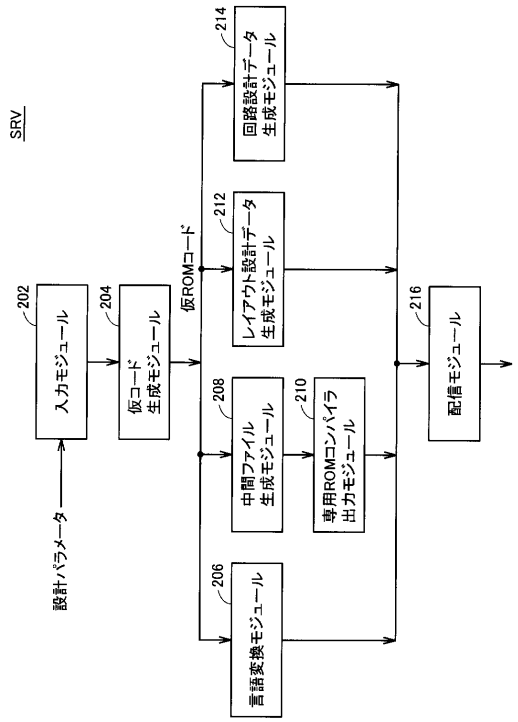
【図３】


SRV_WS
通信インターフェース
109
113
CD-ROM駆動装置
114
111
FD駆動装置
112
102
モニタ
107
固定ディスク
101
CPU
105
106
メモリ
104
マウス
キーボード
103

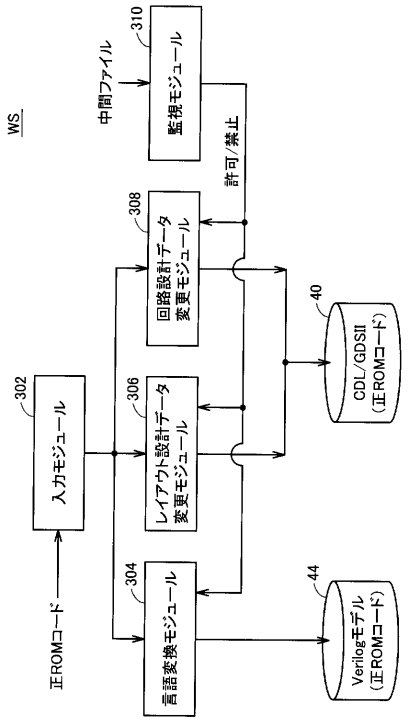
【図４】


ホストサーバ
22
20
10
ROMコンパイラ
専用ROMコンパイラ
28
SRV
Verilogモデル（仮ROMコード）
24
CDL/GDSII（仮ROMコード）
26
配信
設計/顧客
32
・ROMタイプ
・Word数/Bit数
22
専用ROMコンパイラ
WS
正ROMコード
CDL/GDSII（仮ROMコード）
42
26
生成
40
Verilogモデル（正ROMコード）
CDL/GDSII（正ROMコード）
44
46

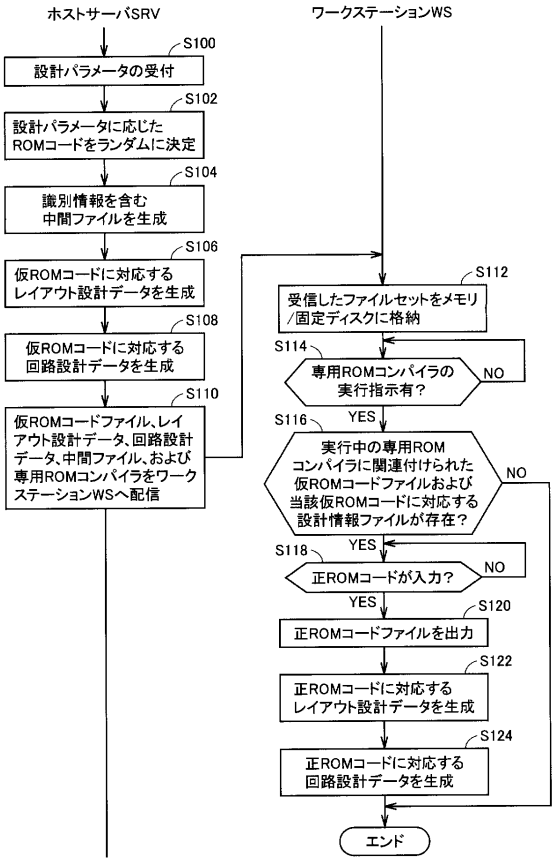
【図５】


SRV
202
入力モジュール
204
仮コード生成モジュール
仮ROMコード
214
回路設計データ生成モジュール
212
レイアウト設計データ生成モジュール
208
中間ファイル生成モジュール
210
専用ROMコンパイラ出力モジュール
216
配信モジュール
206
言語変換モジュール
設計パラメータ

【図６】


WS
中間ファイル
310
監視モジュール
許可/禁止
308
回路設計データ変更モジュール
40
CDL/GDSII（正ROMコード）
302
入力モジュール
306
レイアウト設計データ変更モジュール
304
言語変換モジュール
44
Verilogモデル（正ROMコード）
正ROMコード

【図７】


ホストサーバSRV
ワークステーションWS
S100
設計パラメータの受付
S102
設計パラメータに応じた
ROMコードをランダムに決定
S104
識別情報を含む
中間ファイルを生成
S106
仮ROMコードに対応する
レイアウト設計データを生成
S108
仮ROMコードに対応する
回路設計データを生成
S110
仮ROMコードファイル、レイアウト設計データ、回路設計データ、中間ファイル、および専用ROMコンパイラをワークステーションWSへ配信
S112
受信したファイルセットをメモリ/固定ディスクに格納
S114
専用ROMコンパイラの実行指示有？
NO
YES
S116
実行中の専用ROMコンパイラに関連付けられた仮ROMコードファイルおよび当該仮ROMコードに対応する設計情報ファイルが存在？
NO
YES
S118
正ROMコードが入力？
NO
YES
S120
正ROMコードファイルを出力
S122
正ROMコードに対応する
レイアウト設計データを生成
S124
正ROMコードに対応する
回路設計データを生成
エンド

フロントページの続き

審査官　平野　崇

(56)参考文献　特開平０３－０１６０９９（ＪＰ，Ａ）
特開２００２－１６３３１０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 17/50
IEEE Xplore
CiNii
JSTPlus(JDreamII)