

⑤Int. Cl.⁴

識別記号

庁内整理番号

⑬公開 平成1年(1989)12月13日

G 06 F 15/60

3 7 0

A-8125-5B

審査請求 未請求 請求項の数 30 (全23頁)

⑭発明の名称 ASIC用計算機支援設計システム

⑮特 願 平1-7550

⑯出 願 平1(1989)1月12日

優先権主張 ⑰1988年1月13日⑱米国(US)⑲143,821

⑳発明者 小林 秀 昭 アメリカ合衆国 29201 サウス・カロライナ、コロンビア、メイン・ストリート1401番、スイート890 インターナショナル・チップ・コーポレーション内

㉑発明者 進 藤 晶 弘 東京都大田区中馬込1丁目3番6号 株式会社リコー内

㉒出願人 インターナショナル・チップ・コーポレーション アメリカ合衆国 29201 サウス・カロライナ、コロンビア、メイン・ストリート1401番、スイート890

㉓出願人 株式会社リコー 東京都大田区中馬込1丁目3番6号

㉔代理人 弁理士 青山 葆 外1名

明 細 書

1. 発明の名称

ASIC用計算機支援設計システム

2. 特許請求の範囲

(1)集積回路の機能的仕様から直接的に特定用途向集積回路を設計するための計算機支援設計システムであって、

集積回路についての機能の仕様を定めるために使用者によって操作可能な入力手段と、

上記機能的仕様を集積回路の構造レベルの定義に変換するコンピュータで作動する変換手段とを備えたことを特徴とする計算機支援設計システム。

(2)機能的仕様は一連の動作と条件とを有し、構造レベルの定義はブロックと、ブロック間の相互接続とを有している請求(1)項記載のシステム。

(3)入力手段は、集積回路の機能の仕様を記述した一連の動作と条件を表わす要素を有するフローチャートを作成するフローチャートエディタを備えている請求(1)項記載のシステム。

(4)コンピュータで作動する手段は、動作と条件

を実現する集積回路ハードウェア素子を定める素子ライブラリと、構造的な定義に対するブロックに対応する適切なハードウェア素子を素子ライブラリから選択する素子選択手段とを含む請求(1)項記載のシステム。

(5)集積回路の機能的仕様から直接的に特定用途向集積回路を設計するための計算機支援設計システムであって、

動作と条件を定めるマクロライブラリと、集積回路についての機能の仕様を定めるために使用者によって操作可能な入力手段であって、この機能的仕様は一連の動作と条件とを有し、さらに入力手段は使用者がマクロライブラリから選択されたマクロを各限定された一連の動作と条件に含まれる動作と条件に対して特定することが可能にする手段を含む入力手段と、

動作と条件を実現する集積回路ハードウェア素子を定める素子ライブラリと、上記仕様によって特定されたマクロについて素子ライブラリからマクロによって決定された動作または条件を実行す

るためにハードウェア素子を素子ライブラリから選択する素子選択手段とを含む計算機支援設計システム。

(6)素子選択手段は、素子ライブラリからハードウェア素子を選択する規則を含む知識ベースと、

上記知識ベースの規則にしたがって素子ライブラリから適切なハードウェア素子を選択するインターフェースエンジン手段とを備えている請求(5)項に記載のシステム。

(7)入力手段は一連の動作と条件とを表わす要素を有するフローチャートを含むフローチャートエディタを備えた請求(5)項記載のシステム。

(8)集積回路の動作を使用者が検証し得るためにフローチャート内で定められた機能を模擬するフローチャートシュミレータを備えている請求(7)項記載のシステム。

(9)入力手段は一連の動作と条件を定めるリストの使用による入力を受信する手段を備えている請求(5)項記載のシステム。

(10)素子選択手段は素子選択手段と協働してこの

請求(13)項記載のシステム。

(15)集積回路の機能的仕様から直接的に特定用途向集積回路を設計するための計算機支援設計システムであって、

一連の動作と条件を定めるマクロライブラリと、

一連の動作と条件を表わす要素を有するフローチャートを作成するフローチャートエディタであって、フローチャートで表される各動作と条件に対して、マクロライブラリから選択されたマクロを使用者が特定することを可能にしているマクロ特定手段を含むフローチャートエディタと、

動作と条件を実現する集積回路ハードウェア素子を定める素子ライブラリと、

マクロによって決定された動作または条件を実行するためにハードウェア素子を素子ライブラリから選択する素子選択手段と、

素子選択手段と協働してこの素子選択手段によって選択されたハードウェア素子についてのデータ通路を生成するデータ通路発生手段と

を備えたことを特徴とする計算機支援設計システム。

素子選択手段によって選択されたハードウェア素子についてのデータ通路を生成するデータ通路発生手段を含む請求(5)項記載のシステム。

(11)データ通路発生手段はハードウェア素子間あるいはハードウェア素子とインターフェースエンジン間のデータ通路を選択する規則を含む知識ベースを含み、素子選択手段によって選択されたハードウェア素子間のデータ通路を、上記知識ベースの規則と特定マクロの注書とにしたがって選択する請求(10)項記載のシステム。

(12)セル選択手段によって選択されたハードウェア素子についてのコントローラと制御通路を生成する制御発生器を含む請求(10)項記載のシステム。

(13)素子選択手段と協働するネットリスト生成手段を備え、集積回路の機能的条件を完成するに必要なハードウェア素子を決定するネットリストを出力するものである請求(5)項記載のシステム。

(14)ネットリストから、特定の機能的条件を有する集積回路を製造するために必要なマスクデータを生成するマスクデータ生成手段を備えている請求(5)項記載のシステム。

ム。

(16)素子選択手段は素子ライブラリからハードウェア素子を選択するための規則を含む知識ベースとインターフェースエンジンとを備え、

知識ベースの規則にしたがって素子ライブラリからハードウェア素子を選択し、

データ通路発生手段はハードウェア素子間あるいはハードウェア素子とインターフェースエンジン間のデータ通路を選択する規則を含む知識ベースを含み、素子選択手段によって選択されたハードウェア素子間のデータ通路を、上記知識ベースの規則と特定マクロの注書とにしたがって選択する請求(15)項記載のシステム。

(17)セル選択手段によって選択されたハードウェア素子についてのコントローラと制御通路を生成する制御発生器を含む請求(15)項記載のシステム。

(18)集積回路の機能的仕様から直接的に特定用途向集積回路を設計するための計算機支援設計システムであって、

動作を表わす四角な箱形と条件を表わすひし形

と、動作と条件との間の遷移を示す矢印を有する線を有するフローチャートを使用者によって作成し得るとともに、各箱形やひし形に関して実行すべき特定の動作や条件を与える手段を含むフローチャート編集手段と、

動作と条件を実現する集積回路ハードウェア素子を定める素子ライブラリと、

素子ライブラリからハードウェア素子を選択する規則およびハードウェア素子用のデータおよび制御通路を生成する規則を含む知識ベース

および上記知識ベースと協働して、フローチャート編集手段で定められたフローチャートを、特定の機能要件を備えた集積回路に要求される所要のハードウェア素子とデータ用および制御用通路とを決めるネットリストに変換するエキスパートシステムとを備えたことを特徴とする計算機支援設計システム。

(19) ネットリストから、特別の機能要件を備えた集積回路を製造するに要するマスクデータを生成する手段を備えた請求(18)項記載のシステム。

データ通路を生成する工程を含む請求(21)項記載の方法。

(23) データ通路を生成する工程は知識ベースにストアされたデータ通路規則を選択された素子に適用することおよびデータ通路を生成することを含む請求(22)項記載の方法。

(24) 選択された集積回路のハードウェア素子用の制御通路を生成する工程を含む請求(23)項記載の方法。

(25) 集積回路の所望の機能を実行するのに必要なハードウェア素子を決定するネットリストおよび各素子間の接続条件を選択された集積回路ハードウェア素子に対して生成する工程を含む請求(20)項記載の方法。

(26) 所望の機能を有する集積回路を製造するために必要なマスクデータをネットリストから生成することを含む請求(25)項記載の方法。

(27) 所望の機能を実行する特定用途向集積回路を設計する知識ベース設計方法であって、

所要の動作と条件を定義するマクロをマクロラ

(20) 所望の機能を実行する特定用途向集積回路を設計する計算機支援設計方法であって、

動作と条件の定義をストアすること、

ストアされた動作と条件を実行するために、利用可能な集積回路のハードウェア素子を記述したデータをストアすること、

要求される特定用途向集積回路に対して動作と条件とを記述すること、

記述された動作と条件に対して、実行すべき所要の動作と条件に対応する上記ストアされた定義を特定すること、

特定用途向集積回路の所望の機能を実行するため、特定の定義に対して、ストアされたデータから集積回路ハードウェア素子を選択すること
を備えた計算機支援設計方法。

(21) ハードウェア素子を選択する過程は実行すべき動作と条件の特定の定義に対して知識ベースにストアされている素子選択規則を付与するものである請求(20)項記載の設計方法。

(22) 選択された集積回路のハードウェア素子用の

イブラリにストアすること、

動作と条件を実行するために集積回路の利用可能なハードウェア素子を素子ライブラリにストアすること、

ストアされた、マクロによって定義される動作と条件を実行するために素子ライブラリから選択されたハードウェア素子用の規則を知識ベースにストアすること、

所望の特定用途向集積回路について集積回路によって実行されるべき機能を行う一連の動作と条件を提案された特定用途向集積回路について記述すること、

各記述された動作と条件について動作と条件に対応してマクロライブラリから一連の選択されたマクロを特定すること

および特定用途向集積回路の所望の機能を実行するに必要なハードウェア素子を素子ライブラリから選択するために特定のマクロに対して知識ベースの規則を適用することを含むことを特徴とする知識ベース設計方法。

(28)動作と条件を記述する工程は動作と条件を表す要素を含むフローチャートを生成することを含む請求(27)項記載の方法。

(29)ハードウェア素子間のデータ通路を生成する規則を知識ベースにストアすることおよび知識ベースの規則を選択されたハードウェア素子に対してデータ通路を生成するために知識ベースの規則を特定する手段に適用することを含む請求(27)項記載の方法。

(30)選択されたハードウェア素子に対してコントローラと制御通路を生成する工程を含む請求(29)項記載の方法。

3. 発明の詳細な説明

産業上の利用分野

この発明は集積回路の設計に関し、特に集積回路を設計する計算機支援自動設計方法と設計装置に関するものである。

従来技術

特定用途向集積回路(application specific integrated circuit 以下ASICという)はマ

たとえば回路図、回路パラメータ、あるいは特別なハードウェア記述語(HDL)などの形式で示される。構造レベルでの設計仕様書から、ハードウェア部品と相互接続についての記述が、集積回路チップの実際の微細構造の特徴を敘述する具体的チップ配置レベルの記述に変換される。この具体的チップ配置レベルの記述によって当該チップを製造するのに必要なマスクデータを供給する。

大規模集積回路(VLSI)技術の発展によって非常に複雑な回路システムが単一のチップに形成されるようになってきている。その複雑さと早期にVLSIチップを設計するようにという要求とで、多量性と、特別の用途の増加とに対してコンピュータ支援設計即ちCAD技術を用いる必要がある。CAD技術は、集積回路の構造レベルと物理的配置レベルとの両面において成功裡に集積回路の設計と検査とに用いられている。たとえばCADシステムはVLSIの構造レベルの記述を実際の製造に必要な物理的配置(レイアウト)レベルでの微細構造のマスクデータに変換する作業を支援する

マイクロプロセッサやメモリチップなどの汎用集積回路チップとは異なり、特定の機能を実行するように設計された集積回路チップのことである。VLSI回路設計に特別な知識を有する高級な設計技術者はASICの設計を担当させられる。設計過程においては、VLSI設計技術者は実現すべき特別の方針ないしは目的と当該集積回路によって行すべき作業(タスク)を考慮し、各コンポーネント間の必要な相互接続とともに、所望の機能や作用を実行するに要する種々のハードウェア部品を決定するための、構造レベルでの設計仕様書を作成する。これらの部品間の同期動作を行なうためにシステム制御装置をも設計しなければならない。この作業は、必要な素子間の接続、信号レベルの互換性、タイミング互換性、物理的配置等と同様に所望の目的や方針を達成するために要する種々のハードウェア部品の全般的なかつ豊富な知識を必要とする。各設計過程において、設計者は長い、単調な解析をしなければならない。VLSI設計技術者によって作成された設計仕様書は、

ことに用いられている。現存するCADシステムはVLSIの設計工程を容易にするのに役立っているが、現在の実務は高度のVLSI設計技術者が必要な構造レベルのハードウェア記述を作成することが必要である。

そのような集積回路のハードウェア記述を作成できる高度の技術を身につけた技術者は少ししかない。利用可能なVLSI CADツールを用いてもこの種の設計は長時間を要し、また人的作業を伴うから誤りが発生する可能性もある。カスタム集積回路のより良い、かつ安価な方法が要望されている。

課題を解決する手段と作用

この発明のCADシステムと方法は理解容易な機能レベルの表現を用いて、目的とする集積回路の機能的必要事項を使用者が決定でき、かつ特有の機能的機能を実行することができるASICを直接作成するに要する詳細な情報を上記機能的必要事項から発生し得る。このようにして、本発明は、高度のVLSI設計技術が有している特別な

専門家の知識を有しないASIC設計者、技術者でなくとも、設計と製造を可能にしたものである。

上述の所望ASICの機能的な仕様は、リスト形式や好ましくはフローチャートの形式などの適宜な仕方決定することができる。フローチャートは種々の熟練レベルと訓練度とを有するソフトウェア技術者やハードウェア技術者が容易に理解でき、論理動作の順序を記述するのに極めて有効である。本発明のシステムと方法はフローチャートから(或いは他の機能的な仕様書から)、機能的仕様を、ASICを直接製造するのに使用できる構造レベルの定義に翻訳する。この構造レベルの定義は、所定の機能の仕様を完成するために必要な集積回路のハードウェアセルのリストをも含んでいる。これらのセルは種々の機能と技術的仕様あるいは明細を実現するための予じめ設計されたハードウェア素子を記憶させた素子ライブラリから選択される。さらに、この発明はシステムコントローラと選択された集積回路用ハードウェア素子の制御経路をも発生する。ハードウェア素子の

ASIC設計者から抽出した知識データベースを備えた知識ベースエキスパートシステムを用いたものであり、フローチャートから、選択されたハードウェア素子と各素子の相互接続条件を記述したネットリストを発生するものである。

実施例

第1a図、第1b図、第1c図は集積回路設計を表わす3つの異なったレベルを示す。第1a図はフローチャート形式の機能的あるいは動作上の表現を示す。フローチャートはアルゴリズムの図形的表現であり、2種のブロックまたは状態、即ち動作と条件(あるいは決定)から成り立っている。フローチャートでは動作は矩形のブロックにて表わされ、条件は菱形で表わされる。各動作及び各条件間の移行は矢印付の線で示される。第1b図は集積回路の構造レベルあるいは論理表現レベルを示す。この表現においては、ブロックは種々の機能を実行する集積回路のハードウェア素子を示すために用いられ、ブロックを結ぶ線は各ブロック間のデータの流れや制御信号の通路(バス)を示

リストとその相互接続の必要条件はネットリストの形式で表わされる。公知のマニュアル操作または既存のVLSI CAD配置システムを用いて、ネットリストから、チップ形式のASICを製造するのに必要な、詳細なチップレベルのマスクデータなどの幾何学的図形情報を発生させることができる。

以下に詳細に述べる本発明のシステムと方法の好ましい実施例においては、知識ベースシリコンコンパイラ(Knowledge based silicon compiler)(略してKBSCという))と称する。KBSCは人工知能(artificial Cartificeal intelligence)とエキスパート(expert)システム技術に基いてなされたASICの設計技術である。KBSCの使用者のインターフェイスはフローチャートエディタ(flow chart editor)であり、このフローチャートエディタは、設計者がフローチャート形式でVLSIシステムを表わすことができるものである。KBSCはVLSI設計における高度のエキスパート化技術を有するエキスパートA

す。ブロックはたとえば、加算器、比較器レジスタ、システムコントローラなどのハードウェア素子を示す。第1c図は集積回路設計の具体的レイアウトレベルの表現であり、集積回路を構成する装置と接続導体を実際に製造するのに必要な詳細なマスクデータを提供する。

既に述べたように、構造レベルでの集積回路設計は高度でVLSI設計について専門化した設計技術者が望まれる。しかしながら、この発明のKBSCシステムによればVLSI設計のエキスパート化がなされ、また該エキスパート化が本発明により応用されて、集積回路は機能レベルで設計され得る。論理回路の図形表現化に代えてフローチャートで設計者が作業を行なえるようにしたことによってカスタム集積回路の設計作業を容易にかつ迅速化して、かつ安価化するとともに、信頼度を高め得る。動作レベルの簡単なフローチャートを用いて設計者はアルゴリズムを取り扱い、設計者は目的の仕事(task)を完成するための具体的手段を知るよりはむしろその仕事をするに必要な

論理ステップだけを知ればよいのである。フローチャートを用いて設計することによって、テストにおける作業が軽減することができる。なぜならば、フローチャートによる作業は設計者にとっては、アルゴリズムにより近いからである。一方、既存のVLSI設計ツールは設計者が、構造レベルでの複雑な回路図でアルゴリズムを表現する必要があり、テストのための作業を要する。回路図は、ハードウェアと機能とについての考慮を混合するので、目的とする設計に導入することを要する演算機能に回路図が打ち勝つのは困難である。カスタム設計集積回路を設計するために、フローチャートを使用することは以前はハードウェアの記述を得るために、少数の設計者だけが知識を有していたVLSI技術に多数のシステム設計者が関与することを可能にする。

本発明に用いられる全体的なシステムの流れを第2図に示す。使用者は回路の機能(動作)仕様をフローチャート11の形式でKBSC(知識ベースシリコンコンパイラ)10に入力する。KBSC

ーチャートを作りかつ編集するフローチャートエディタ21とそのフローチャートを模擬し、検査するフローチャートシミュレータ22を含んでいる。フローチャートにおいて各四角形によって実行される動作はマクロライブラリ23から選択される。PSCS(path synthesizer and cell selector 通路同期器および素子選択器)と呼ばれるプログラム30はデータおよび制御信号同期器モジュール31を含み、同期器モジュール31はデータおよび制御信号同期用の知識ベースシステムである。PSCSはさらにシステム設計に必要な素子を選択するための素子選択器32を含んでいる。この素子選択器32は以前に設計されたハードウェア素子の素子ライブラリ34から、フローチャートに示される各動作と条件に必要な適宜な素子を選択する。制御信号発生器33は他のハードウェア素子の動作を制御するためのカスタム設計システム制御信号を発生する。知識ベース35はデータ通路同期器および素子選択に必要なASIC設計エキスパート用の知識を含ん

C10はフローチャート11からネットリスト15を発生する。ネットリスト15はカスタム生成システムコントローラ、必要な動作をさせるために必要な、他のハードウェア素子および上記ハードウェア素子間およびシステムコントローラを接続する相互接続情報を含んでいる。このネットリストは幾何図形的レイアウト用のマスクデータ18を生成するために既存のVLSIレイアウトならびにルーティング(routing)ツール16への入力として使用することができる。

システム概観

KBSCシステムを有する基本的な素子あるいはモジュールを第3図に示す。この実施例において、これらの素子あるいはモジュールはソフトウェアプログラムの形式であり、当業者にとってはこれらの素子を他の適宜な形態のハードウェアに具現することは容易であろう。

第3図を参照して、KBSCシステム10はEDSIMと呼ばれるプログラム20を含むKBSCシステムが示されている。このシステムはフロ

ーチャートによりPSCSはシステム制御信号を発生し、他のすべてのハードウェア素子を選択し、さらにデータと制御通路を発生し、さらにすべての設計情報を記載したネットリストを発生する。

KBSCシステムは第4図に示したようにハイアラキ素子選択ASIC設計アプローチを用いている。スクラッチ(scratch)即ち所望の機能を果たす種々の素子をかき集めた群からそれぞれの必要なハードウェア素子を発生するよりもKBSCシステムは以前に設計されたまた試験され証明された種々の形のあるいは与えられた形態の種々の動作能力を持つハードウェア素子についての素子ライブラリ34を備えている。マクロライブラリ23はフローチャートにおいて具体化される種々の動作を定めるマクロを含んでいる。マクロライブラリ23内の各マクロ機能に対して特有の機能を実行する別の幾何図形的および特性の素子ライブラリ34内にいくつかのハードウェア素子をも有する。エキスパートASIC設計者から引き出

した知識ベース35を有する規則ベースエキスパートシステムを用いることによってKBSCシステムは素子ライブラリ34から所望の機能を実行するための最適の素子を選択する。

第3図を参照して、素子選択器32によって選択された素子、制御信号発生器33により発生した制御情報およびデータ/制御通路同期器31により発生したデータおよび制御通路はすべてネットリスト15を発生するためにPSCSプログラム30によって用いられる。ネットリストは回路内における各ブロックおよび各ブロックの入力端子と出力端子の間の相互接続を示すリストである。このネットリストは集積回路を製造するのに必要なすべての情報を提供する。市販の素子配置および素子接続用の計算機支援設計システムは上記ネットリストデータを入力として受信し、チップ内のそれぞれの素子の配置(レイアウト)をなし、又必要な接続を定め、集積回路の製造におけるチップ工場(foundry)によって直接用いられるマスクデータを発生する。

ート、論理など)、データベース(素子ライブラリ)および知識ベースを記憶する。CPUはフローチャートの生成と模擬および設計の自動同期を実行する。

フローチャートの例

ひとつのネットリストに対するフローチャートの図形表示を行うために第5図に示すような大型のシステムの一部としての第5図に示すフローチャート例を考える。この図示されたフローチャートにおいて二つの変数VAL1およびVAL2が比較され、もし両者が等しければ両者が合計される。この例において第1の動作(Action 1)は変数VAL1をレジスタAへ移す動作を含む。第2の動作は変数VAL2をレジスタBへ移す動作を含む。条件1(condition 1)はレジスタAとBの各値を比較する(Compare)動作を含む。第3の動作(Action 3)はレジスタAとBの値を加算(Add)し、その結果をレジスタCに蓄える(Store)。

第5図に示された機能を実行するための集積回路を製造するに際してKBSCはハードウェア素

システムに要求される条件

KBSCシステムは適当なプログラムされた汎用デジタルコンピュータによって操作される。例えばKBSCシステムの一例においてはsun3 and VAX Station-11/GPX Running UNIX Operating System and X Window Managerなどのワークステーションで操作される。このワークステーションは8メガバイトの主記憶部と20メガバイトのハードディスクとを少なくとも必要とする。使用されたモニターとしては8ビット平面のカラーテレビである。ソフトウェアはCプログラミング言語とINGRESリレーショナルデータベースを用いる。

人とのインターフェイスは主としてポップアップメニュー、ボタンおよび特別の目的のコマンド言語を用いる。集積回路設計の永久データは抽出と更新を容易にするためにデータベースの形でストアされる。主記憶部は一時的な次回データ(next data temporarily)、エグゼキュータブルコード(executable code)、設計データ(フローチャ

子間の相互接続条件についてのシステムの動作のフローチャート記述を図示する。これらのハードウェア素子はすべての制御信号を発生するシステムコントローラによって制御される。システムコントローラに含まれる変数としては次の2種類がある。

(1)入力変数:入力変数はハードウェア素子によって作られるか、あるいはコントローラへの外部からの入力である。これらはフローチャートにおける条件に対応する。

(2)出力変数:出力変数はシステムコントローラによって作られ、フローチャートにおける動作に対応する。

第6図はハードウェア素子についての第5図のフローチャートを図示する結果を示す。フローチャートの動作と条件は素子選択とデータならびに制御通路同期化に用いられる。VAL1レジスタおよびVAL2レジスタおよび各レジスタから導かれるデータ通路は現実実施例における動作1の以前に生じている。動作1はレジスタAのデータを発

生させる。同様に動作2はレジスタBのデータを位置付けさせる。比較器は条件1における比較動作の結果として位置付けられる。この比較動作は1)比較器の素子を選ぶこと、2)レジスタAとBに対する比較器素子の入力を定めること、3)比較器をレジスタAとBに接続する通路を定めること、4)等しいか、より大きいか、より小さいかにしたがって、システムコントローラ用の入力変数を発生させる。同様に動作3における加算動作は加算素子の選択を行いレジスタに対する加算器のパラメータを作りデータ通路を生成する。この過程の後ブロックリストがフローチャートに対して生成される。このブロックリストはシステムコントローラおよび必要な動作を実行するのに要する種々のブロックについての情報を含み各ブロックはデータ通路で接続され、かつ制御通路を介してシステムコントローラによって制御される。これらのブロックは素子リストを作るために素子ライブラリから素子を選択するために作られる。

相互動作フローチャートエディタおよびシミュレ

ータ

フローチャートの作成と検証とはVLSI設計手法の第1段階である。ひとつのアルゴリズムからひとつの等価なフローチャートへの変換はフローチャートエディタ21(第3図)によってなされる。編集されたフローチャートの検証はフローチャートシミュレータ22によってなされる。フローチャートエディタとシミュレータは親密なインターフェイスの設計者によってフローチャート編集動作を行うために同一の作業環境内に一体的に設けられる。

EDSIMはフローチャートエディタ21とフローチャートシミュレータ22とを含むプログラムである。このEDSIMはフローチャートをロードし、あるいはセーブするなどの機能を提供する。EDSIMは状態リストと呼ばれる各フローチャートに対する中間ファイルを生成する。この中間ファイルはネットリストを生成するためにPSCSプログラム30によって使用される。

フローチャートエディタ

フローチャートエディタ21はフローチャートを表示し、作成し、編集するために使用されるソフトウェアモジュールである。このモジュールは第7図に示されたフローチャート編集窓を介して制御される。編集機能とともにフローチャートエディタは設計エラーのチェックを行う。

フローチャートエディタの動作を以下に説明する。主編集機能は状態と条件と遷移の生成と編集と削除とを含んでいる。生成動作においては設計者が新しい状態と条件あるいは遷移をフローチャートに追加することを許可する。編集は設計者がフローチャートの状態の位置、条件、遷移を変更することを許可し、削除は設計者が現存のフローチャートから状態と条件と遷移を削除することを許可する。動作を含む状態は四角の箱で示され、条件は菱形で示され、遷移は遷移の方向を示す矢印を伴った線で示される。

編集動作は設計者が各箱に対して動作を付与することを許可する。これらの動作はマクロ名および注書き(Argument)を含んでいる。注書きの例

は外部信号の設定およびクリアなどである。マクロライブラリ23における利用可能な基本的なマクロのリストを表1に示す。

表 1

Macro(マクロ)	Description(記述)
ADD (A, B, C)	c = A + B
SUB (A, B, C)	c = A - B
MULT (A, B, C)	c = A * B
DIV (A, B, C)	c = A div B
DECR (A)	A = A - 1
INCR (A)	A = A + 1
CLR (A)	A = 0
REG (A, B)	B = A
CMP (A, B)	Compare A to B and set EQ, LT, GT signals (AとBを比較し、EQ, LT, GT信号を設定)
CMP0 (A)	Compare A to 0 and set EQ, LT, GT signals (Aと0を比較し、EQ, LT, GT信号を設定)
NEGATE (A)	A = NOT (A)
MOD (A, B, C)	c = A Modulus B
POW (A, B, C)	c = A ^ B
DC2 (A, S1, S2, S3, S4)	Decode A into S1, S2, S3, S4 (AをS1, S2, S3, S4にデコード)
EC2 (S1, S2, S3, S4, A)	Encode S1, S2, S3, S4, into A (S1, S2, S3, S4をAにデコード)
MOVE (A, B)	B = A
CALL sub-flowchart (A, B, ...)	Call a sub-flowchart. Pass A, B... (サブフローチャート呼び出す。A, B...を通過)
START (A, B, ...)	Beginning state of a sub-flowchart (サブフローチャートの状態の始まり)
STOP (A, B, ...)	Ending state of a sub-flowchart (サブフローチャートの状態の終わり)

フローチャートエディタはフローチャートシミュレータがフローチャートを模擬するにしたがってフローチャートをグラフィック的に表示する。このグラフィック表示は第7図に示すような箱と菱形と線とを含んでいる。すべてが表示スクリーン上に画かれ従来のフローチャートと同様な形に表示される。シミュレーションの間に表示スクリーン上にフローチャートを表示することによって設計者がフローチャートを同時に設計し検証することを許可する。

フローチャートシミュレータ

フローチャートシミュレータ22はフローチャートをシミュレート(模擬)するのに使用されるソフトウェアモジュールである。このモジュールは第8図に示したシミュレータ窓を介して制御される。フローチャートシミュレータはフローチャートの状態とあるいは条件間での遷移をシミュレートする。フローチャートシミュレータの動作を以下に示す。

データ編集—レジスタあるいはメモリ内の値を交

更

状態設定—シミュレートされるべき次の状態を設定

詳細あるいは要約のディスプレイの設定—シミュレーションの間の詳細なあるいは要約の情報を表示

破断設定—破断点を設定

破断のクリア—すべての破断点をクリア

破断の表示—現在の破断点の表示

ステップ—ひとつの遷移を歩進

実行—フローチャートの実行

停止—フローチャートの実行を停止

ヒストリーオンあるいはヒストリーオフ—ヒストリー(履歴)の記録のオンあるいはオフ

キャンセル—実行中の動作のキャンセル

ヘルプ(help)—ヘルプ(help)画面の表示

クローズ(close)—シミュレータ窓を閉じる(close)

シミュレーションの結果はシミュレータ窓に表示される。またエディタ窓はフローチャートがシミュ

レートされるにしたがってシミュレートしている部分を追跡する。フローチャートの追跡はエラーが発見された時フローチャートの編集を容易にする。

素子選択

素子選択器32はVLSIシステムを実現するために素子ライブラリ34から最適な素子の一群を選択するための知識ベースシステムである。この選択はフローチャート内の機能の記述に基づいてなされるものでフローチャート内において表示されている各動作に対して割り当てられたマクロによって具体化される。VLSIシステムを作成するために選択される素子は種々の条件例えば素子の機能、製造技術の種類、使用される電力の制限、遅延時間などに依存する。素子選択器は素子選択を行うためのVLSI設計専門技術者(エキスパート)から抽出した知識ベースを用いる。

使用者の利用におけるフローチャートの記載からVLSIシステムを設計するためにフローチャート内の機能と素子ライブラリからの素子との間

の整合が必要である。この整合を行うためには、素子選択工程が複雑であり多数の設計パラメータおよび拘束に基づいてなされるものであるから、人口知能技術を用いる必要がある。

素子選択に用いられる技術的思想はソフトウェアにおけるコンパイルで用いられる技術的思想と同様である。ソフトウェアにおけるコンパイルにおいては多数のサブルーチンがライブラリから結合される。VLSIシステムの設計においては機能的マクロがライブラリ素子に表される。

第4図はハイアラキの素子選択の技術的思想を示す。素子選択工程は二つのステップで構成される。

(1)機能的マクロの選択

(2)幾何図形的素子の選択

基本的なマクロの一群を表1に示す。一つのマクロはフローチャート内の一つの動作に対応する。例えばAとBを加算し、その結果Cをストアする動作を考える。この機能は加算マクロADD(X, Y, Z)として表される。フローチャートエディタ

(4)二つの素子の併合

(5)エラーの診断

上記情報は規則として知識ベース35にストアされる。

素子リスト作成

第9図は素子リスト作成の段階を示す。

素子リスト作成の第1の段階はフローチャート記載の素子選択器によって利用できる構造への変換である。この変換は状態リストと呼ばれる。ブロックリストはインターフェイスエンジンによって状態リストから作成される。このブロックリストは集積回路において使用される機能ブロックのリストを含んでいる。この段階においては以下に述べる形態の規則が利用される。

- データ通路への注書きの作成
- マクロに対する動作の作成
- 各ブロックの接続

規則はこのレベルにおける素子の最適化とエラー診断とを用意する。

素子選択器は素子ライブラリ34から選択され

とフローチャートシミュレータはフローチャートの方形の箱と菱形および線を引くために用いられ、さらにマクロライブラリ23から選択されたマクロをフローチャート内に示される各動作に割り当てることならびにフローチャート内での機能を検証するために用いられる。フローチャートは中間の形状(状態リスト)に変換され素子選択器に入力される。

素子選択器は各動作を実行するための適当な素子を選択するための規則ベースエキスパートシステムを用いる。もし素子ライブラリが、マクロによって特定される動作を実行するための異なる形状の多数の素子を有しているならば、素子機能、使用されるプロセス技術、遅延時間、電力制限値などの条件に基づいて素子の選択が行われる。

素子選択器32の知識ベースは以下の各項に係する情報あるいは規則を含んでいる。

(1)マクロの選択

(2)二つのマクロの併合

(3)マクロと素子の連結

た素子に対するブロックを作成する。また、素子選択器は一つのブロックに対して最適の素子を選択する。この動作は素子ライブラリからの適当な素子の選択のための規則の型式化をも含んでいる。各素子には四つの形式の情報をストアしている。

(1)機能レベル情報:レジスタ転送レベルでの素子についての記述

(2)論理レベル情報:フリップフロップとゲートに関する記述

(3)回路レベル情報:トランジスタレベルでの記述

(4)レイアウトレベル情報:幾何図形マスクレベルの明細

素子の属性は

- 素子名
- 記述
- 機能
- 幅
- 高さ
- 状態

- 技術
- 最小遅延時間
- 典型的な遅延時間
- 最大遅延時間
- 電力
- ファイル
- 設計者
- 日付
- コメント
- 検査者

素子選択工程において、上述の情報を使用することができる。素子に対するマクロを生成するために用いることができるいくつかのパラメータは

- (1)マクロ名
- (2)実行すべき機能
- (3)チップの複雑さ
- (4)製造技術
- (5)可能な遅延時間
- (6)消費電力
- (7)マクロデータ通路のビットサイズ

る。PSCS操作の最高レベルの記述を以下に示す。

- (1)フローチャート中間ファイルを読み状態リストを形成する
- (2)現在のコンテキスト(context)=START
- (3)インターフェイスエンジンを起動し現在のコンテキスト規則をロード
- (4)現在のコンテキストにしたがって以下のうちのいずれかを実行する
 - (a)正しい履行のために状態リストを修正
 - (b)ブロックのリストとマクロリストとデータ通路とを作成
 - (c)ブロックリストとデータ通路を最適化し、エラーチェックを実行
 - (d)ブロックを素子に変換
 - (e)素子リストを最適化し、エラーチェックを実行
 - (f)ネットリストの作成
 - (g)ネットリストの最適化とエラーチェックの実行、その完了後7へ行く(go to 7)

ネットリスト生成

ネットリストは素子がPSCSによって選択された後に生成される。PSCSはまた素子の端子を他の素子に接続するためのマクロの定義を使用する。PSCSはネットリストを生成するために例えば状態リストなどのフローチャートの表示形式の中間リストから状態から状態への遷移を使用する。PSCSはネットリスト生成のための以下の知識を含んでいる。

- 1)データ通路同期
- 2)データ通路最適化
- 3)マクロの定義
- 4)素子ライブラリ
- 5)エラー検出および訂正

上述の情報は知識ベース35に規則としてストアされる。知識技術者はASIC設計エキスパートから上述の規則を作成することを補助する。マクロライブラリ23と素子ライブラリ34はKBSCのデータベースにストアされる。

PSCSによっていくつかの操作がおこなわれ

(5)もし現在のコンテキストが変化したならば新しいコンテキスト規則をロード

(6)4へ行く(go to 4)

(7)ネットリストファイルとslfファイルを出力しストップ

次の章においてステップ4において述べた操作が記述される。規則言語およびPSCS表示が記述される。

規則言語PSCSの規則言語は宣言的でありかつ規則の編集を容易にするように設計される。専門家にとって知識ベースの構造を理解できるように規則言語は知識表示をするための手段を提供する。この動作はデータ構造の形式(フォーマット)を規則ベースで述べられるようにし、専門家がデータ構造を参照でき、システムによって利用される種々の構造を理解できるようにする。例えばエキスパートは配線の構造を解析でき、構成部分を決定できる。エキスパートはそれらの構成部分を規則の中へ取り込むことができる。新しい対象物が決められたときエキスパートは新しい構造を宣

言し、この新しい構造に既存の構造をリンプさせるように修正する。この方法においてデータ構造の成長がエキスパートによってより良く視覚化される。そして次に、これによって設計者が規則を修正し、あるいは追加することができる。

以下の態様が規則言語の中に含まれる。

- i) レコード構造の形状の知識表示
- ii) 規則の前文における状態の表示
- iii) 規則動作における構造の作成および破壊の容易化
- iv) 規則の動作における任務の記述
- v) 規則動作における入力と出力の容易化
- vi) 規則動作からのC機能の刺激の容易化

使用される規則フォーマットは以下の通りである。

規則	番号	コンテキスト
If	if clause	
Then	Then clause	
Where	number	規則番号

規則がデータベース内にストアされる。この動作を実行するためにエキスパートは種々の知識構造とインターフェイス工程とを熟知していることが必要である。もしこれが不可能ならば、知識技術者の援助が必要である。

PSCSは種々のオプションを設定できるメニューを提供する。種々のデバッグフラグを設定するための機構、および表示オプションの機構、さらにPSCSの全体的な制御を行うための機構が設けられる。

使用者によって作成されたブロックリストをセーブし、表示するための機構も設けられる。使用者によって作成されたブロックリストの形状はファイルにセーブすることができ、後にブロックによって印刷することができる。PSCS表示工程を再スタートさせるためにリセットすることができる。

PSCS 規則例

規則 1

IF もしブロックが存在しなければ
THEN システムコントローラを発生させる

context この規則は活性である
というコンテキスト

インターフェイス方式

インターフェイス方式は高速パターンマッチングアルゴリズムに基づく。この規則はネットワーク中にストアされ規則を通して繰り返して述べるための必要条件是回避される。これによって実行の速度を上げることができる。コンフリクトレソリューション(conflict resolution)方式は以下のものに基づく。

(1) 最も近い時点のデータが含まれている規則が選択される。

(2) 最も複雑な条件を有する規則が選択される。

(3) 最初に宣言された規則が選択される。

規則エディタ

PSCSはエキスパートが規則のセットを更新できるように相互作用をする規則エディタを提供する。この規則はデータベースパッケージの編集能力が規則の編集に用いることができるように、

規則 2

IF もしマクロを有する状態が存在し、
AND (および) このマクロがブロックに作られていなければ
THEN ライブラリ内で対応するマクロを見出し、このマクロに対するブロックを生成する。

規則 3

IF もし二つの状態間で遷移があればAND (および) 同じ注書きを用いた状態でマクロがあれば
THEN 最初のマクロに対応するレジスタから第2のマクロに対応する他のレジスタへ接続を作成する。

規則 4

IF もし一つのレジスタが他の一つのレジスタから一つだけ接続されているならば
THEN これらのレジスタを一つのレジスタに結合する

規則 5

IF もし二つの比較器がありAND(および)入力データ幅が同じサイズでありAND(および)それらの一つの入力が同じでありAND(および)比較器の出力が同じ動作を行うのに用いられるならば

THEN これらの比較器を一つの比較器に結合する

規則 6

IF もしレジスタなしでデータがあれば

THEN 一つのレジスタをこのデータのために設ける

規則 7

IF もしすべてのブロックが既に相互に接続されAND(および)一つのブロックが接続されないいくつかの端子を有しているならば

THEN これらのブロックとその端子を除きあるいはエラーメッセージを出力する

プレクサの出力を上記一つのブロックへ接続する

他の規則の要点は、以下に示すように、

- 他の素子の出力を用いることによって置換できる素子は除去する
 - 多重のトリートを簡略化する
 - 素子からのファンアウトを利用する
- などである。

ソフト飲料自動販売機用コントローラの設計例

以下の例は特定用途向集積回路(A S I C)の設計において本発明の前述の態様が以下に用いられるかを示す。この例においてはA S I Cは自動販売機のコントローラの設計に用いたものである。自動販売機のコントローラは硬貨受け取り部に硬貨が投入されるごとに信号を受ける。受け取った硬貨の総額が適正な値になると、その硬貨の金額が記録され、コントローラは一つの飲料を搬出する信号を出力する。飲料の価格よりも高額な金額の硬貨が受け取られるとコントローラは正しい金額になるようにつり銭を出す。

る

規則 8

IF もしメモリが使用され、しかしそのメモリのためのブロックが作成されていないならば

THEN データ、アドレス、データおよび制御用の読出、書込端子とともにメモリブロックを作成する

規則 9

IF もしレジスタがカウンタへの一つの接続を有しているならば

THEN レジスタとカウンタを結合してレジスタとその端子を除去する

規則 10

IF もし異なるいくつかのブロックから一つのブロックの一つの端子へ接続があるならば

THEN マルチプレクサを挿入し、端子への接続を除去し、それらの端子をマルチプレクサの入力へ接続し、マルチ

この自動販売機のコントローラの例においてはデジタルシステムコントローラを教えるテキストを用いた後にパターンが作られる。Fletcher, William L., An Engineering Approach to Digital Design, Prentice-Hall, Inc., pp. 491-505 参照。このテキストを参照して自動販売機のコントローラの要件をより完全に説明し、コントローラのハードウェアを設計するためには、本発明の以前においてはいかにその設計手法が複雑であったかを理解することができよう。

第10図は自動販売機のコントローラのフローチャートを示す。このフローチャートはフローチャートエディタを介して使用者により、KBSCシステムに入力される。フローチャートを一見すると硬貨が硬貨受け取り部に存在するときコントローラが硬貨有りの信号を受ける。State0およびCond0は硬貨の投入を待っていることを示すウェイト状態を定義している。CPは硬貨有りを示しており、!CPは硬貨無しを示す。State1お

よびCond1は硬貨が硬貨受け取り部からクリアされたときを決定する。State20では硬貨を受け取った後にマクロインストラクションADD3.1(1c,cv,sum)がシステムに1c(last coin)およびcv(coin value)を指示し、その合計をストアする。state21と協働するマクロインストラクションはレジスタsumの値をレジスタcvへ移動させる。state22におけるマクロCMP.1はレジスタcvの値をPR(飲料の価格price of soft drink)と比較して信号EQ,GTおよびLTを返す。条件cond2は比較動作CMP.1の結果を検査する。その結果が"大きくない"(!GT,CMP.1)であれば条件cond3は結果が"等しい"(EQ,CMP.1)であるかどうかをみるための検査を行う。もし結果が"等しくない"(!EQ,CMP.1)であれば制御は状態state0へ戻り他の硬貨の投入を待つ。もしcond3がEQであれば、状態state4は飲料を搬出するための制御信号を発生し、マクロインストラクションCLR.1(cv)はcvを0にリセットして、他の顧客を待つ。

示す。比較マクロCMP(A,B)は、値Aをストアするレジスタ、Bをストアするレジスタ、比較器のブロックを生成し、さらに比較の結果発生するEQ,LT,GT信号用のシステムコントローラのための制御通路を生成する。別のマクロADD(A,B,C)は各入力値AおよびB用のレジスタ、出力値C用のレジスタおよび加算器(adder)ブロックを生成する。マクロDECR(A)はカウンタ(counter)ブロックを生成する。PSCSプログラム30は第10図のフローチャートに用いられたマクロを対応するハードウェア素子へ生成し、第12図に示すハードウェアブロックを生成する。図示されたブロックの生成においてPSCSプログラム30はすでに述べた規則の例のうち規則1と規則2とによっている。

第13図はデータ通路と制御通路をともに示した第12図のブロックの接続を示す。データおよび制御通路の生成においてデータ/制御シンセサイザプログラム31により規則3を用いた。

第14図は冗長なレジスタを消去するために規

則4を適用して回路を最適化した結果を示す。この規則の適用の結果、第13図のレジスタR2,R3,R7,R8,R9が除かれた。第15図は冗長な比較器を省略したさらなる最適化を行った結果のブロックダイアグラムを示す。この最適化は規則5の適用によりPSCSプログラムで完成した。

もし投入された硬貨の合計金額が飲料価格を越えているときは、状態state30は動作"return coin"(硬貨返却)を出力する。さらにマクロDECR.1(cv)は返却された硬貨の合計金額だけcvの値を減算する。状態state31でcvとPRが再び比較される。もしcvがまだPRよりも大きければ、制御は状態state30へとび、他の硬貨を返却する。条件cond5はCMP.2の結果が等しいかどうかを検査し、飲料を搬出するか(droppop)あるいは状態state0へ戻り他の硬貨の投入を待つ。状態と協働する第10図に示すマクロは上述の表1に定義したものに对应し、それぞれの状態において実行されるべき特定の動作を決定する。

付表Aは第10図に示したフローチャートで作成される中間ファイルあるいは状態リストを示す。この状態リストはEDSIMプログラム20からの出力として作成され、PSCSプログラム30(第3図)への入力として用いられる。

第11図は第10図のフローチャートに用いられる各マクロと対応するハードウェアブロックを

示す。システムコントローラブロックが決定したので、集積回路の他の必要なハードウェアブロックとデータおよび制御通路およびPSCSプログラム30はこれらのハードウェア部分および必要な接続を決定するネットリスト15を生成する。このネットリストから集積回路を製造するマスクデータが利用可能なVLSI CADツールを用いて直接的に作られる。

システムコントローラブロックが決定したので、集積回路の他の必要なハードウェアブロックとデータおよび制御通路およびPSCSプログラム30はこれらのハードウェア部分および必要な接続を決定するネットリスト15を生成する。このネットリストから集積回路を製造するマスクデータが利用可能なVLSI CADツールを用いて直接的に作られる。

APPENDIX A

```

name rpop;
data path @ic<0:5>,cv<0:5>,sum<0:5>,@pr<0:5>;
{
state4 : state0;
state30 : state31;
state21 : state22;
state20 : state21;

state0 :: !cp state0;
state0 :: cp state1;
state1 :: cp state1;
state1 :: !cp state20;
state22 :: GT.CMP.1 state30;
state22 :: !GT.CMP.1*EQ.CMP.1 state4;
state22 :: !GT.CMP.1*!EQ.CMP.1 state0;
state31 :: GT.CMP.2 state30;
state31 :: !GT.CMP.2*EQ.CMP.2 state4;
state31 :: !GT.CMP.2*!EQ.CMP.2 state0;
state30 :: returncoin;
state30 :: DECR.1(cv);
state4 :: droppop;
state4 :: CLR.1(cv);
state31 :: CMP.2(cv,pr);
state22 :: CMP.1(cv,pr);
state21 :: MOVE.1(sum,cv);
state20 :: ADD3.1(ic,cv,sum);
}

```

を示す図、

第6図は第5図において定義された集積回路用のハードウェアブロックおよび相互接続条件を示した構造を表す図、

第7図はフローチャートエディタ窓を示す図、

第8図はフローチャートシミュレータ窓を示す図、

第9図は素子リスト生成に含まれるステップを示す図、

第10図は自動販売機システムの一例を示すフローチャート、

第11図は第10図のフローチャートに用いられる三つのマクロのそれぞれに対応するハードウェア部分を示す図、

第12図は第10図のフローチャートで決定される集積回路のハードウェア部分を示す初期のブロックダイアグラム、

第13図は各ブロック間の相互接続を示す、第12図に対応するブロックダイアグラム、

第14図はレジスタの最適化後の第13図に対

4. 図面の簡単な説明

第1a図はフローチャート形式で示された所望の回路の一部分についての機能レベルでの設計の表現を示す図、

第1b図は集積回路の構造レベルの設計表現を示す図、

第1c図は集積回路チップの製造において用いられる物理的なレイアウトレベルの回路の設計表現を示す図、

第2図は集積回路が本発明のKBSCシステムによってフローチャート記載からどのように集積回路のマスキングデータが作られるかを示すブロックダイアグラム、

第3図はKBSCシステムの最初の部分の詳細を示す図、

第4図は本発明のASIC設計システムがどのようにして素子ライブラリから集積回路ハードウェア素子を抽出するかを示す図、

第5図は集積回路によって実行される機能的動作のシーケンスを決定するフローチャートの一例

を示すブロックダイアグラム、

第15図はさらに最適化を行った後の第14図に対応するブロックダイアグラムである。

10…KBSC、11…フローチャート、
15…ネットリスト、16…CADツール、
18…マスキングデータ、23…マクロライブラリ、
34…素子ライブラリ、35…知識ベース。

特許出願人 インターナショナル・チップ・

コーポレーション 外1名

代理人 弁理士 青山 稔 外1名

図面の浄書(内容に変更なし)

第1a図

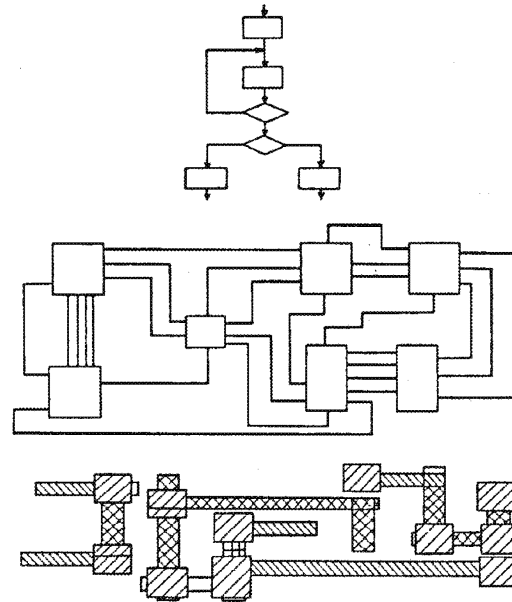
Functional Level
機能レベル

第1b図

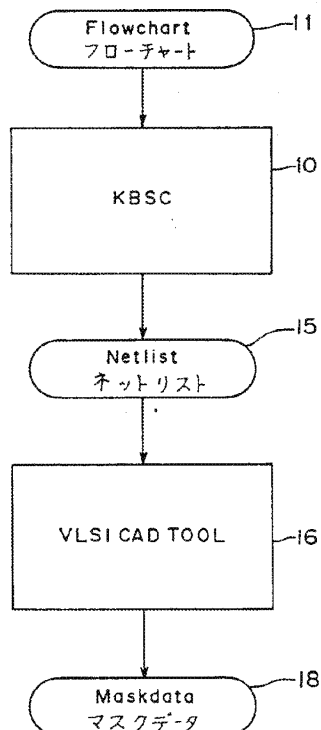
Structural Level
構造レベル

第1c図

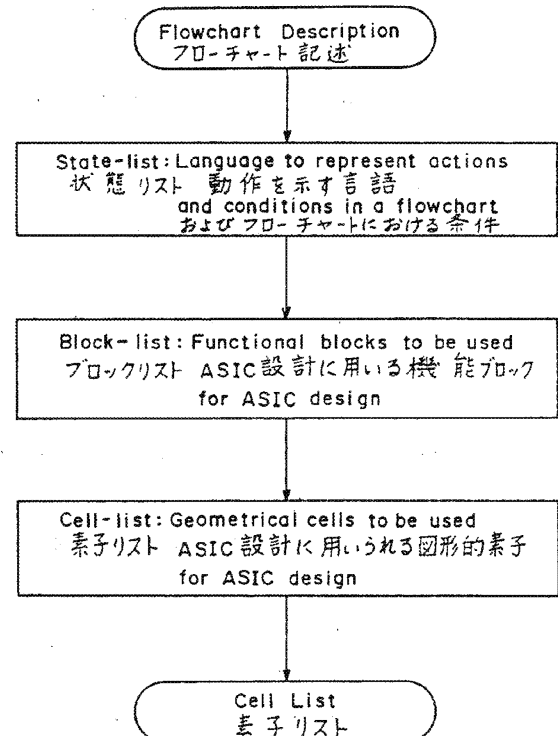
Physical Layout Level
物理的レイアウトレベル



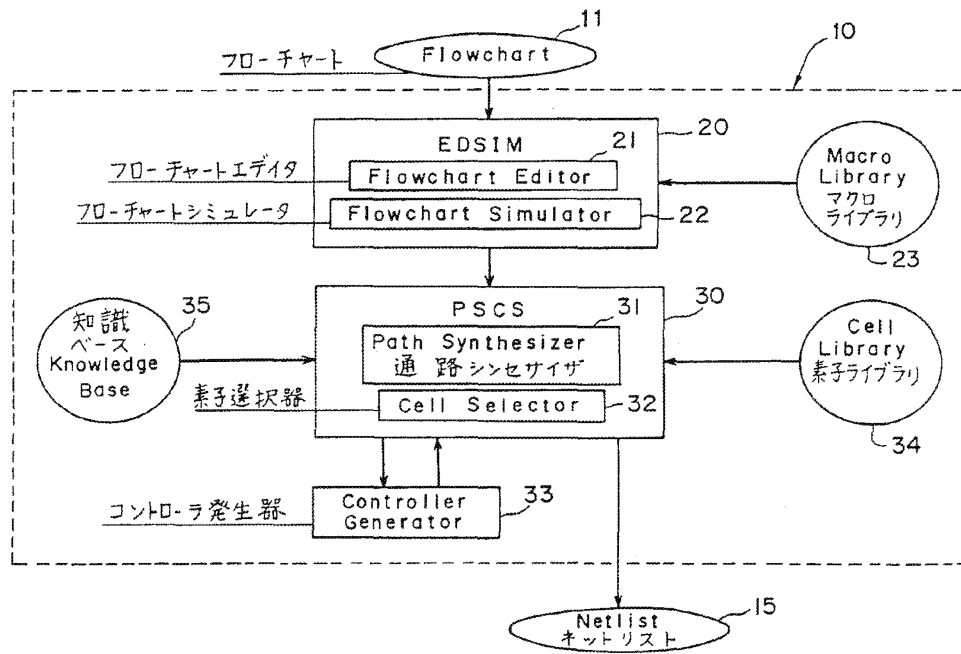
第2図



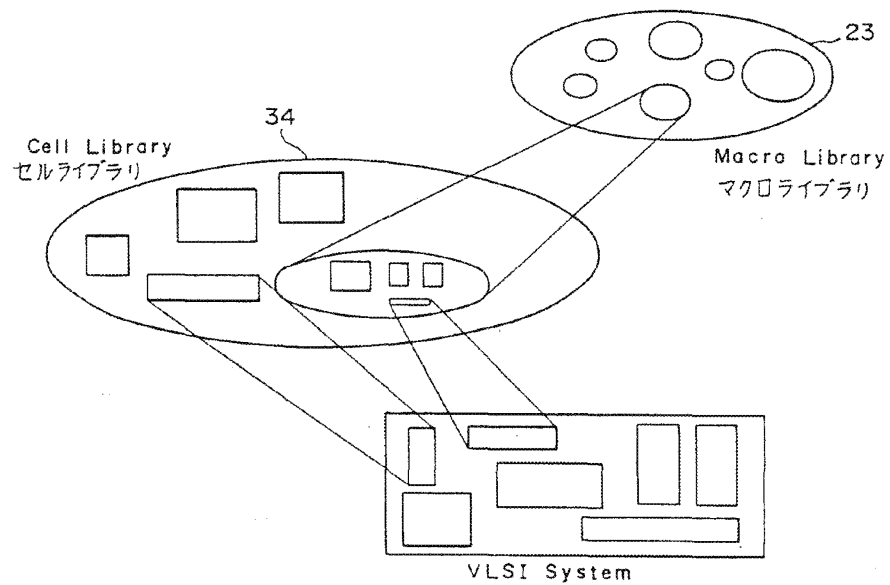
第9図



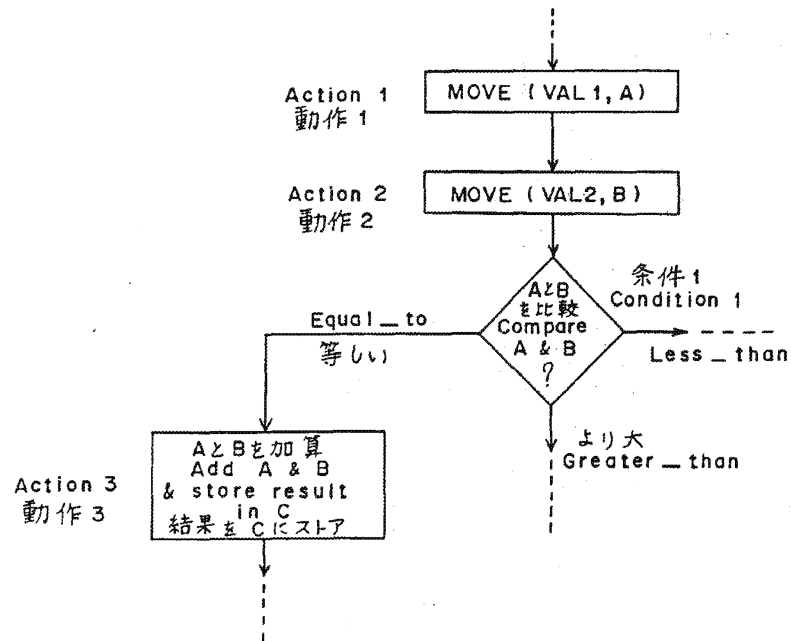
第 3 区



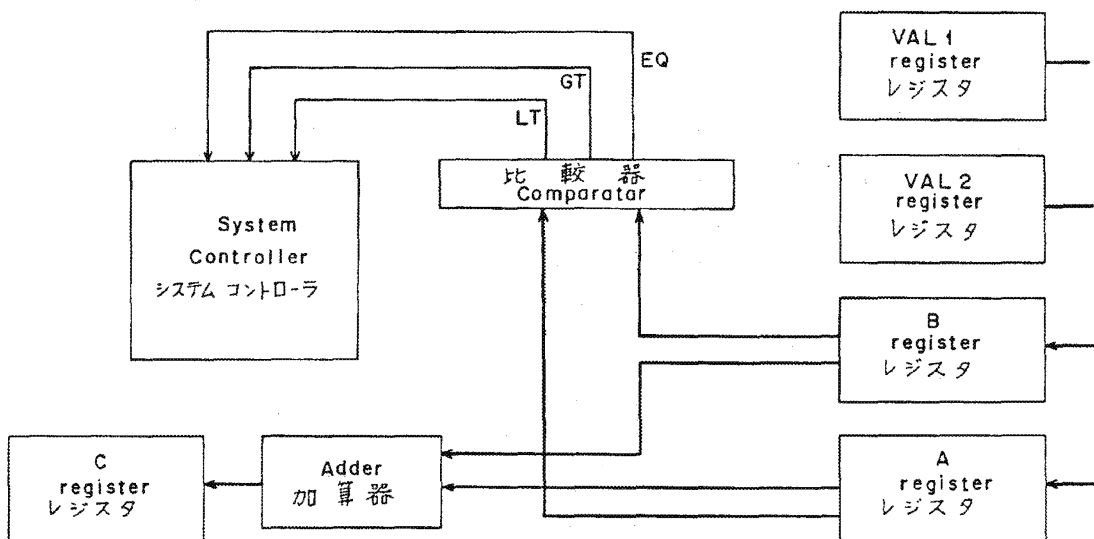
第 4 図



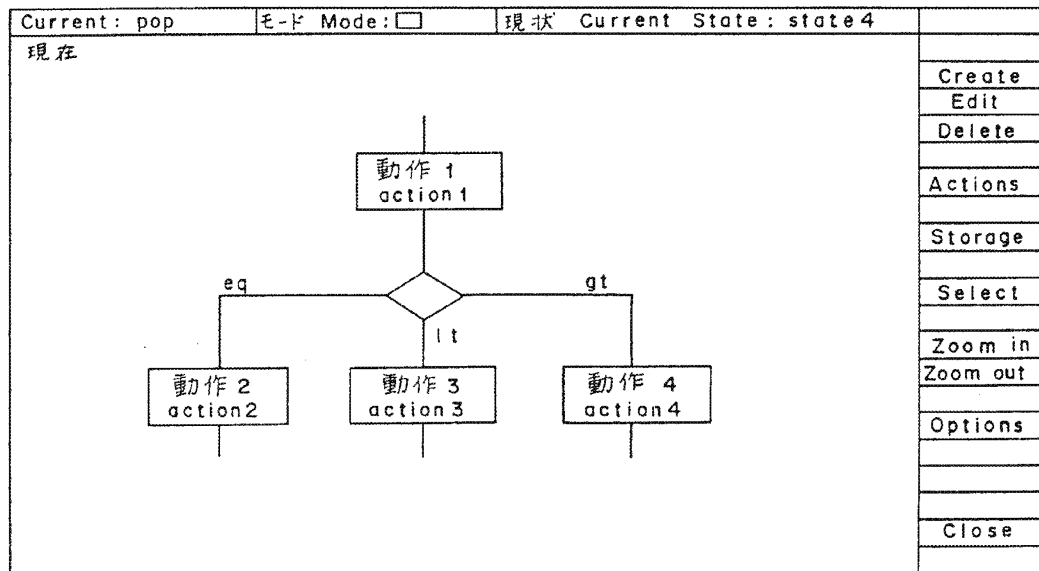
第 5 図



第 6 図



第 7 図

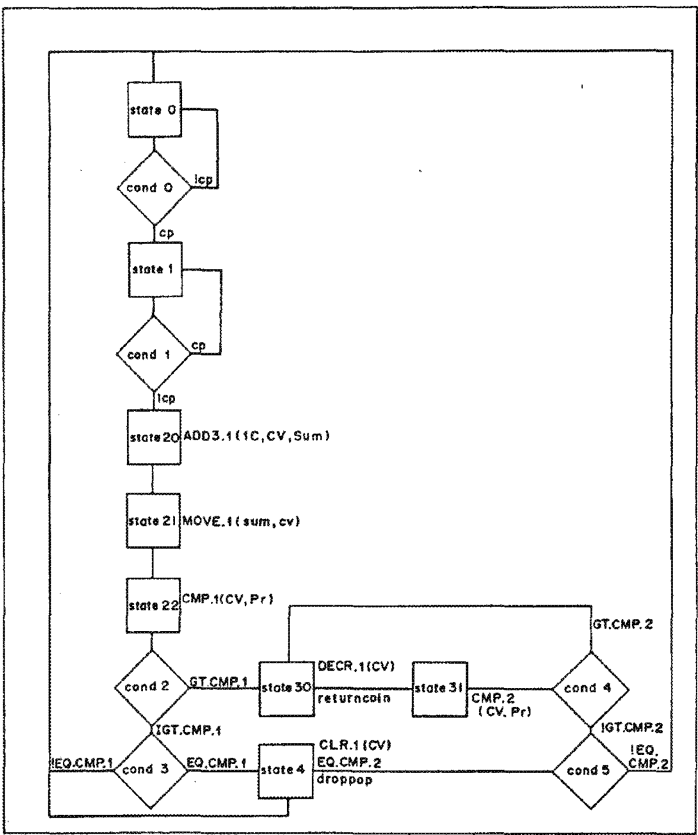


第 8 図

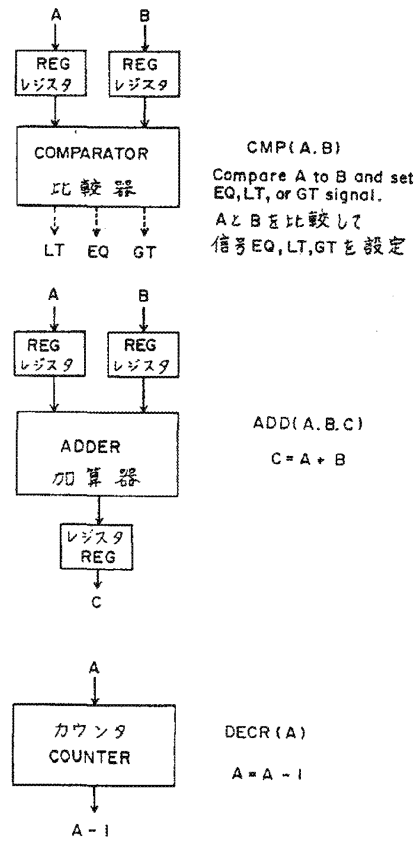
Edit Data	Set Breaks	Step	History ON	Cancel
Show Data	Clear Breaks	Execute	Detail	Help
Set State	Show Breaks	Stop		Close

*** Ready ***

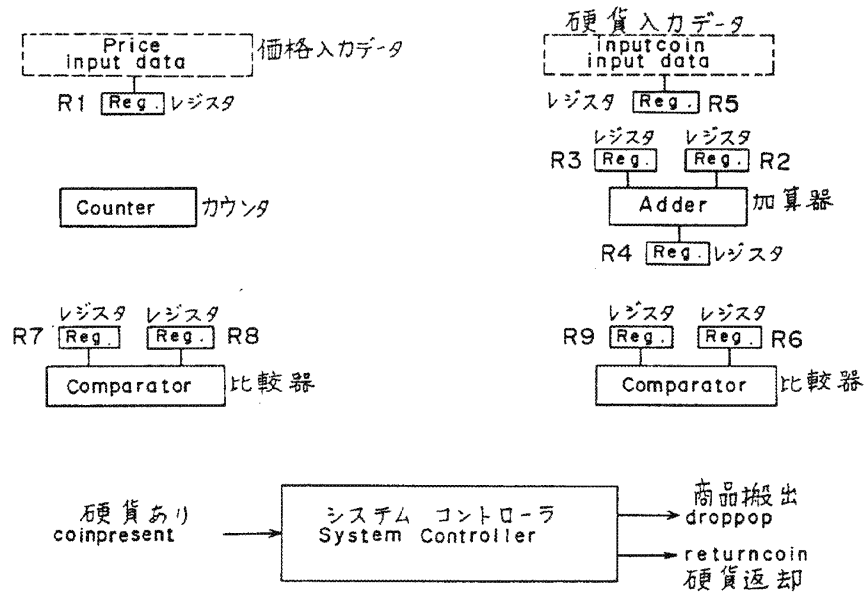
第10図



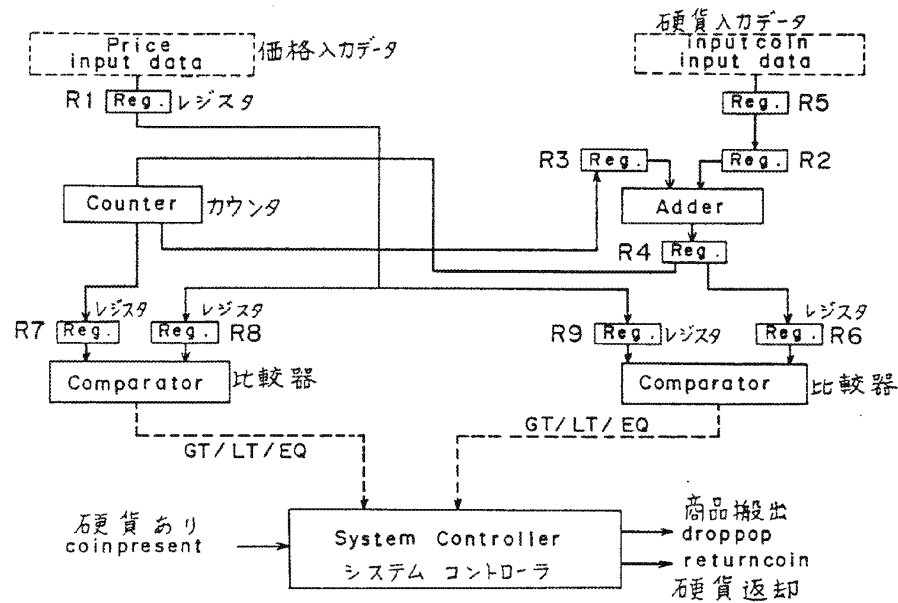
第11図



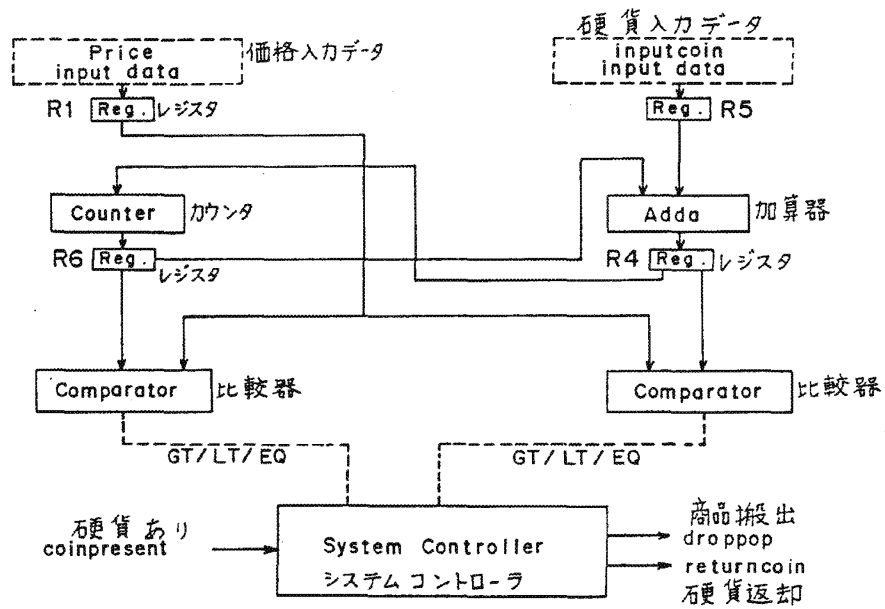
第12図



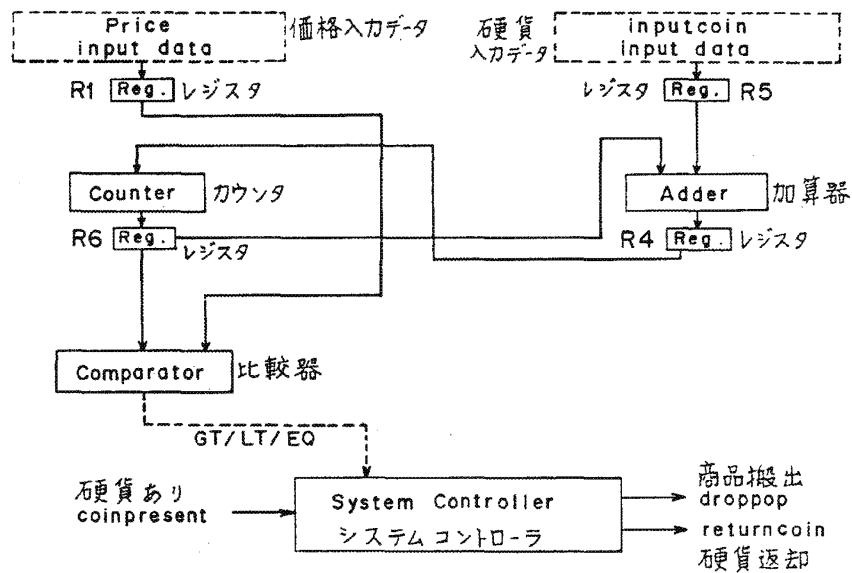
第13図



第14図



第15図



手続補正書 (方式)

特許庁長官 殿

平成 1 年 5 月 1 8 日

1. 事件の表示

平成1年特許願第007550号



2. 発明の名称

A S I C 用 計 算 機 支 援 設 計 シ ス テ ム

3. 補正をする者

事件との関係 特許出願人

名称 インターナショナル・チップ・コーポレーション

ほか1名

4. 代 理 人

住所 〒540 大阪府大阪市中央区城見2丁目1番61号
ツイン21 MIDタワー内 電話(06)949-1261

氏名 弁理士(6214) 青 山 稔



5. 補正命令の日付

平成1年4月25日(発送日)

6. 補正の対象 : 図面

7. 補正の内容 : 願書に最初に添付した図面の浄書・

別紙の通り(内容に変更なし)

