

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4867708号
(P4867708)

(45) 発行日 平成24年2月1日(2012.2.1)

(24) 登録日 平成23年11月25日(2011.11.25)

(51) Int. Cl.

F I

G06F 17/50 (2006.01)
H01L 21/82 (2006.01)

G O 6 F 17/50 6 5 8 U
G O 6 F 17/50 6 5 8 A
G O 6 F 17/50 6 5 8 E
G O 6 F 17/50 6 5 4 G
H O 1 L 21/82 C

請求項の数 10 (全 20 頁)

(21) 出願番号 特願2007-45618 (P2007-45618)
(22) 出願日 平成19年2月26日 (2007.2.26)
(65) 公開番号 特開2008-210109 (P2008-210109A)
(43) 公開日 平成20年9月11日 (2008.9.11)
審査請求日 平成22年1月19日 (2010.1.19)

(73) 特許権者 000004237
日本電気株式会社
東京都港区芝五丁目7番1号
(74) 代理人 100123788
弁理士 官崎 昭夫
(74) 代理人 100106138
弁理士 石橋 政幸
(74) 代理人 100127454
弁理士 緒方 雅昭
(72) 発明者 岡本 匠
東京都港区芝五丁目7番1号 日本電気株式会社社内
審査官 伊知地 和之

最終頁に続く

(54) 【発明の名称】 半導体集積回路の設計方法および設計装置

(57) 【特許請求の範囲】

【請求項1】

設計対象の半導体集積回路における、部品情報と、配線接続情報と、フリップフロップ間のタイミング制約と、に基づいて前記半導体集積回路を設計する設計装置が行う半導体集積回路の設計方法であって、

前記部品情報に基づいて複数の論理ゲートと複数のフリップフロップが配置されたチップ領域を、複数の領域に分割する分割ステップと、

入力部および出力部として異なる前記フリップフロップが用いられ前記入力部と前記出力部の間にはフリップフロップが接続されず少なくとも前記論理ゲートのいずれかが接続される論理回路を、前記配線接続情報に基づいて、前記チップ領域から抽出し、1つの前記領域内に全体が含まれる前記論理回路をコア論理ブロックとし、2つ以上の前記領域にまたがる前記論理回路をインタフェイス論理ブロックとする設定ステップと、

前記配線接続情報と前記タイミング制約に基づいて、前記コア論理ブロックのレイアウト設計を行うコア論理ブロック設計ステップと、

前記配線接続情報に基づいて、前記領域間の配線のレイアウト設計を行う領域間配線設計ステップと、

前記配線接続情報と前記タイミング制約と前記配線のレイアウト設計の結果に基づいて、前記インタフェイス論理ブロックのレイアウト設計を行うインタフェイス論理ブロック設計ステップと、を含む半導体集積回路の設計方法。

【請求項2】

10

20

請求項 1 記載の半導体集積回路の設計方法において、

前記チップ領域に、コア論理ブロックレイアウト領域と、インタフェース論理ブロックレイアウト領域と、領域間の配線レイアウト領域と、を設定する領域設定ステップをさらに含み、

前記コア論理ブロック設計ステップでは、前記コア論理ブロックのレイアウト設計を、前記コア論理ブロックレイアウト領域で行い、

前記領域間配線設計ステップでは、前記領域間の配線のレイアウト設計を、前記領域間の配線レイアウト領域で行い、

前記インタフェース論理ブロック設計ステップでは、前記インタフェース論理ブロックの設計を、前記インタフェース論理ブロックレイアウト領域で行う、半導体集積回路の設計方法。

10

【請求項 3】

請求項 1 記載の半導体集積回路の設計方法において、

前記分割ステップでは、前記部品情報に基づいて前記複数の論理ゲートおよび前記複数のフリップフロップをチップ配置領域に仮配置して前記チップ領域を生成し、前記チップ領域を格子状に分割することにより前記複数の領域を生成し、

前記設定ステップでは、前記領域において、当該領域に全体が含まれる前記論理回路の占める割合が大きくなるように、前記領域間で、前記論理回路を移動し、前記 1 つの領域内に全体が含まれる前記移動された論理回路を前記コア論理ブロックとし、前記 2 つ以上の領域にまたがる前記移動された論理回路を前記インタフェース論理ブロックとする、半導体集積回路の設計方法。

20

【請求項 4】

請求項 3 記載の半導体集積回路の設計方法において、

前記設定ステップでは、前記 1 つの領域に全体が含まれる前記移動された論理回路と、前記 2 つ以上の領域にまたがる前記移動された論理回路と、の間に重複論理ゲートが存在する場合、当該重複論理ゲートを 2 重化して、前記 1 つの領域に全体が含まれる前記論理回路と、前記 2 つ以上の領域にまたがる前記論理回路と、を互いに排他的になるように分離し、前記 1 つの領域内に全体が含まれる前記排他的な論理回路を前記コア論理ブロックとし、前記 2 つ以上の領域にまたがる前記排他的な論理回路を前記インタフェース論理ブロックとする、半導体集積回路の設計方法。

30

【請求項 5】

請求項 1 記載の半導体集積回路の設計方法において、

前記インタフェース論理ブロック設計ステップでは、前記配線のレイアウト設計の結果から得られるブロック間配線遅延を考慮して、全てのインタフェース論理ブロックのレイアウト設計を一括して行う、半導体集積回路の設計方法。

【請求項 6】

設計対象の半導体集積回路における、部品情報と、配線接続情報と、フリップフロップ間のタイミング制約と、に基づいて前記半導体集積回路を設計する、半導体集積回路の設計装置であって、

前記部品情報に基づいて複数の論理ゲートと複数のフリップフロップが配置されたチップ領域を、複数の領域に分割する分割手段と、

40

入力部および出力部として異なる前記フリップフロップが用いられ前記入力部と前記出力部の間にはフリップフロップが接続されず少なくとも前記論理ゲートのいずれかが接続される論理回路を、前記配線接続情報に基づいて、前記チップ領域から抽出し、1 つの前記領域内に全体が含まれる前記論理回路をコア論理ブロックとし、2 つ以上の前記領域にまたがる前記論理回路をインタフェース論理ブロックとする設定手段と、

前記配線接続情報と前記タイミング制約に基づいて、前記コア論理ブロックのレイアウト設計を行うコア論理ブロックレイアウト設計手段と、

前記配線接続情報に基づいて、前記領域間の配線のレイアウト設計を行う配線レイアウト設計手段と、

50

前記配線接続情報と前記タイミング制約と前記配線のレイアウト設計の結果に基づいて、前記インタフェイス論理ブロックのレイアウト設計を行うインタフェイス論理ブロックレイアウト設計手段と、を含む半導体集積回路の設計装置。

【請求項 7】

請求項 6 記載の半導体集積回路の設計装置において、

前記チップ領域に、コア論理ブロックレイアウト領域と、インタフェイス論理ブロックレイアウト領域と、領域間の配線レイアウト領域と、を設定する領域設定手段をさらに含み、

前記コア論理ブロックレイアウト設計手段は、前記コア論理ブロックのレイアウト設計を、前記コア論理ブロックレイアウト領域で行い、

前記配線レイアウト設計手段は、前記領域間の配線のレイアウト設計を、前記領域間の配線レイアウト領域で行い、

前記インタフェイス論理ブロックレイアウト設計手段は、前記インタフェイス論理ブロックの設計を、前記インタフェイス論理ブロックレイアウト領域で行う、半導体集積回路の設計装置。

【請求項 8】

請求項 6 記載の半導体集積回路の設計装置において、

前記分割手段は、

前記部品情報に基づいて、前記複数の論理ゲートおよび前記複数のフリップフロップをチップ配置領域に仮配置して前記チップ領域を生成する仮ゲート配置手段と、

前記チップ領域を格子状に分割することにより前記複数の領域を生成する分割格子設定手段と、を含み、

前記設定手段は、

前記配線接続情報に基づいて、前記論理回路を前記チップ領域から抽出し、前記領域において、当該領域に全体が含まれる前記論理回路の占める割合が大きくなるように、前記領域間で、前記論理回路を移動する移動手段と、

前記 1 つの領域内に全体が含まれる前記移動された論理回路を前記コア論理ブロックとし、前記 2 つ以上の領域にまたがる前記移動された論理回路を前記インタフェイス論理ブロックとする論理ブロック決定手段と、を含む、半導体集積回路の設計装置。

【請求項 9】

請求項 8 記載の半導体集積回路の設計装置において、

前記設定手段は、さらに、前記 1 つの領域に全体が含まれる前記移動された論理回路と、前記 2 つ以上の領域にまたがる前記移動された論理回路と、の重複論理ゲートが存在する場合、当該重複論理ゲートを 2 重化して、前記 1 つの領域に全体が含まれる前記論理回路と、前記 2 つ以上の領域にまたがる前記論理回路と、を互いに排他的になるように分離する論理ブロック分割手段を含み、

前記論理ブロック決定手段は、前記 1 つの領域内に全体が含まれる前記排他的な論理回路を前記コア論理ブロックとし、前記 2 つ以上の領域にまたがる前記排他的な論理回路を前記インタフェイス論理ブロックとする、半導体集積回路の設計装置。

【請求項 10】

請求項 6 記載の半導体集積回路の設計装置において、

前記インタフェイス論理ブロックレイアウト設計手段は、前記配線のレイアウト設計の結果から得られるブロック間配線遅延を考慮して、全てのインタフェイス論理ブロックのレイアウト設計を一括して行う、半導体集積回路の設計装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、半導体集積回路の設計方法および設計装置に関し、特に、論理回路の遅延時間を考慮して半導体集積回路を設計する設計方法および設計装置に関する。

【背景技術】

10

20

30

40

50

【 0 0 0 2 】

クロック同期式半導体回路（以後「LSI」とも表記する）の設計、特に、論理回路の遅延時間を考慮したタイミング設計について、図 1 1 を参照して説明する。

【 0 0 0 3 】

クロック同期式半導体回路では、フリップフロップ（以後「FF」とも表記する）と呼ばれるクロック同期用素子101-01の間に、論理回路が設けられる。論理回路は、論理ゲート101-02と配線（信号線）101-03からなる。

【 0 0 0 4 】

クロック同期式半導体回路設計では、クロック同期用素子間の論理回路を通る信号の遅延時間が制約値（タイミング制約101-05）以内に収まるように、FF間の論理回路のレイアウト設計（「配置配線設計」とも言う）を行う必要がある。

10

【 0 0 0 5 】

詳細には、論理回路を通る信号の遅延時間は、論理ゲート101-02の遅延時間と論理ゲート間を接続する配線101-03の遅延時間からなり、この合計を制約値以内に収める必要がある。

【 0 0 0 6 】

例えば、100MHzで動作する回路を設計するためには、FF間の全ての論理パス101-04の遅延時間を10nano秒以内に収める必要がある。

【 0 0 0 7 】

以下、FF101-01の入力端子から、論理回路を信号の流れと逆方向に辿って他のFF101-01に到達するまでに現れる信号線101-03および論理ゲート101-02の集合を、FFの入力論理コーン（または、論理コーン）101-06と呼ぶ。また、「遅延時間」を単に「遅延」とも呼ぶ。

20

【 0 0 0 8 】

従来、この種の半導体回路の設計手法のひとつとして用いられている「フラット設計手法」を図 1 2 に示す。

【 0 0 0 9 】

フラット設計手法では、論理ゲート101-02と論理ゲート間信号線102-02のレイアウト設計領域を区別することなく、チップ全体をひとつのレイアウト対象領域として、全ての論理回路のレイアウト設計が一括して行われる。

30

【 0 0 1 0 】

レイアウトに関する領域的な制約がなく、自由度および柔軟性が高いため、チップ面積を最小化するためには、フラット設計手法は適している。

【 0 0 1 1 】

しかし、フラット設計手法では、配置領域と配線領域が混在するため、論理ゲートが障害になり、論理ゲート間の配線がどのような経路になるか予測が付きにくい、かつ、経路の制御が難しい、という問題点がある。

【 0 0 1 2 】

図 1 2 に示した迂回配線102-03は、論理ゲート101-02が密に配置された領域を避けるために生じる。

40

【 0 0 1 3 】

超微細プロセス半導体では、論理パス遅延において配線遅延の占める割合が大きく、このような配線迂回は、半導体回路のタイミング設計を困難にする大きな要因となっている。

【 0 0 1 4 】

また、フラット設計手法では、扱える論理ゲート数に限界があり、超大規模回路の設計には適していない、という問題点も存在する。

【 0 0 1 5 】

上述のフラット設計手法の問題点である「配線経路の予測および制御の困難性」および「扱える回路規模の制限」を解決する設計手法として、従来用いられている「分割設計手

50

法」を図13に示す。

【0016】

分割設計手法では、回路全体が複数の論理ブロック集合に分割され(図13では4つの論理ブロック集合に分割している)、集積回路の設計は、論理ブロック内設計と論理ブロック間配線設計の2つに分けて行われる。

【0017】

論理ブロック間を接続する長い配線103-03に対しては専用のレイアウト領域103-02が設けられているため、「論理ゲートが障害になり配線迂回が生じる」という問題が生じにくい。そのため、早期設計段階でのブロック間配線遅延の見積もりが容易であり、また、ブロック間配線領域を変更することにより、配線幅、配線層、リピータ挿入等のブロック間配線遅延を最適化するためのブロック間配線構造の制御を容易に行うことが可能である。

10

【0018】

また、設計対象を小規模の論理ブロック集合に分割して設計を行うため、フラット設計では扱えない規模の回路の設計を、小メモリ、短時間で効率よく行なうことができる、という効果がある。従来階層設計システムの一例が、特許文献1ないし3に記載されている。

【特許文献1】特開2004-192227号公報

【特許文献2】特開2004-302818号公報

【特許文献3】特開2006-338090号公報

【発明の開示】

20

【発明が解決しようとする課題】

【0019】

従来フラット設計手法には、上述したように、配置領域と配線領域が混在するため、論理ゲートが障害になり、論理ゲート間の配線がどのような経路になるか予測が付き難い、経路の制御が難しい、半導体回路のタイミング設計を困難にする大きな要因となっている配線迂回が生じやすい、という問題点が存在する。この問題を解決するために、配置配線を融合したフラット設計手法も提案されているが、この場合、扱える回路の規模に制限がある、という問題点が存在する。

【0020】

分割設計手法は、上述のフラット設計手法の問題点を解決することは可能であるが、問題を分割して解くために、図14に示すように、複数の論理ブロックにまたがる論理コンが存在する場合、各ブロックに含まれる部分論理回路に対して、本来のタイミング制約を分割して設定する必要がある。

30

【0021】

例えば、本来の制約が10nano秒であった場合、ここから、論理ブロック間の配線遅延を減じた値を4分割する、などの処理が必要である。

【0022】

各ブロックに適切なタイミング制約を設定することは困難であるため、本来一括して最適化すべき問題が分割されることにより、結果の全体的な最適性が失われてしまうという問題、所謂、分割損の問題が存在する。

40

【0023】

また、従来階層設計では、論理設計階層を基に論理ブロックを生成するため、論理ブロックサイズにバラつきが生じることが多く、論理ブロック間に規則的な配線構造を用いるのに適していない。

【0024】

本発明の目的は、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にする、半導体集積回路の設計方法および設計装置を提供することにある。

【課題を解決するための手段】

【0025】

50

上記目的を達成するために、本発明の半導体集積回路の設計方法は、設計対象の半導体集積回路における、部品情報と、配線接続情報と、フリップフロップ間のタイミング制約と、に基づいて前記半導体集積回路を設計する設計装置が行う半導体集積回路の設計方法であって、前記部品情報に基づいて複数の論理ゲートと複数のフリップフロップが配置されたチップ領域を、複数の領域に分割する分割ステップと、入力部および出力部として異なる前記フリップフロップが用いられ前記入力部と前記出力部の間にはフリップフロップが接続されずに少なくとも前記論理ゲートのいずれかが接続される論理回路を、前記配線接続情報に基づいて、前記チップ領域から抽出し、1つの前記領域内に全体が含まれる前記論理回路をコア論理ブロックとし、2つ以上の前記領域にまたがる前記論理回路をインタフェイス論理ブロックとする設定ステップと、前記配線接続情報と前記タイミング制約に基づいて、前記コア論理ブロックのレイアウト設計を行うコア論理ブロック設計ステップと、前記配線接続情報に基づいて、前記領域間の配線のレイアウト設計を行う領域間配線設計ステップと、前記配線接続情報と前記タイミング制約と前記配線のレイアウト設計の結果に基づいて、前記インタフェイス論理ブロックのレイアウト設計を行うインタフェイス論理ブロック設計ステップと、を含む。

10

【0026】

また、本発明の半導体集積回路の設計装置は、設計対象の半導体集積回路における、部品情報と、配線接続情報と、フリップフロップ間のタイミング制約と、に基づいて前記半導体集積回路を設計する、半導体集積回路の設計装置であって、前記部品情報に基づいて複数の論理ゲートと複数のフリップフロップが配置されたチップ領域を、複数の領域に分割する分割手段と、入力部および出力部として異なる前記フリップフロップが用いられ前記入力部と前記出力部の間にはフリップフロップが接続されずに少なくとも前記論理ゲートのいずれかが接続される論理回路を、前記配線接続情報に基づいて、前記チップ領域から抽出し、1つの前記領域内に全体が含まれる前記論理回路をコア論理ブロックとし、2つ以上の前記領域にまたがる前記論理回路をインタフェイス論理ブロックとする設定手段と、前記配線接続情報と前記タイミング制約に基づいて、前記コア論理ブロックのレイアウト設計を行うコア論理ブロックレイアウト設計手段と、前記配線接続情報に基づいて、前記領域間の配線のレイアウト設計を行う配線レイアウト設計手段と、前記配線接続情報と前記タイミング制約と前記配線のレイアウト設計の結果に基づいて、前記インタフェイス論理ブロックのレイアウト設計を行うインタフェイス論理ブロックレイアウト設計手段と、を含む。

20

30

【0027】

上記発明によれば、コア論理ブロック、インタフェイス論理ブロック、および、領域間の配線のレイアウト設計は、個別に行われる。このため、領域間配線時、論理ゲート配置が障害にならず、配線幅、配線間隔、リピータ挿入等の、領域間配線遅延を最小化するための配線構造の制御を行うことが容易になる。また、設計対象を分割して設計するため、大規模の集積回路を、小メモリ、短時間で行うことができる。

【0028】

コア論理ブロックは、入出力部が全てFFとなっているため、FF間のタイミング制約を分割することなく、各コア論理ブロックの設計を、他の論理ブロックと独立に行うことが可能になる。

40

【0029】

インタフェイス論理ブロックの設計については、例えば、実際の領域間配線結果に基づく正確な遅延を考慮して、複数の領域にまたがるインタフェイス論理ブロック集合全体を一括して最適化できるため、従来の分割設計が持つ、分割損による、タイミング設計収束性悪化問題を解決することができる。

【0030】

よって、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にすることが可能になる。

【0031】

50

なお、前記設計装置は、前記チップ領域に、コア論理ブロックレイアウト領域と、インタフェイス論理ブロックレイアウト領域と、領域間の配線レイアウト領域と、を設定する領域設定手段をさらに含み、前記コア論理ブロックレイアウト設計手段は、前記コア論理ブロックのレイアウト設計を、前記コア論理ブロックレイアウト領域で行い、前記配線レイアウト設計手段は、前記領域間の配線のレイアウト設計を、前記領域間の配線レイアウト領域で行い、前記インタフェイス論理ブロックレイアウト設計手段は、前記インタフェイス論理ブロックの設計を、前記インタフェイス論理ブロックレイアウト領域で行うことが望ましい。

【0032】

上記発明によれば、コア論理ブロックと領域間の配線とインタフェイス論理ブロックとが、それぞれに対応するレイアウト領域で設計される。このため、領域間配線時、論理ゲート配置が障害にならず、配線幅、配線間隔、リピータ挿入等の、領域間配線遅延を最小化するための配線構造の制御を行うことが容易となる。

10

【0033】

また、前記分割手段は、前記部品情報に基づいて、前記複数の論理ゲートおよび前記複数のフリップフロップをチップ配置領域に仮配置して前記チップ領域を生成する仮ゲート配置手段と、前記チップ領域を格子状に分割することにより前記複数の領域を生成する分割格子設定手段と、を含み、前記設定手段は、前記配線接続情報に基づいて、前記論理回路を前記チップ領域から抽出し、前記領域において、当該領域に全体が含まれる前記論理回路の占める割合が大きくなるように、前記領域間で、前記論理回路を移動する移動手段と、前記1つの領域内に全体が含まれる前記移動された論理回路を前記コア論理ブロックとし、前記2つ以上の領域にまたがる前記移動された論理回路を前記インタフェイス論理ブロックとする論理ブロック決定手段と、を含むことが望ましい。

20

【0034】

上記発明によれば、インタフェイス論理ブロックのサイズを小さくすることが可能になる。よって、インタフェイス論理ブロック全体の一括最適化を容易にすることが可能になる。

【0035】

また、前記設定手段は、さらに、前記1つの領域に全体が含まれる前記移動された論理回路と、前記2つ以上の領域にまたがる前記移動された論理回路と、の重複論理ゲートが存在する場合、当該重複論理ゲートを2重化して、前記1つの領域に全体が含まれる前記論理回路と、前記2つ以上の領域にまたがる前記論理回路と、を互いに排他的になるように分離する論理ブロック分割手段を含み、前記論理ブロック決定手段は、前記1つの領域内に全体が含まれる前記排他的な論理回路を前記コア論理ブロックとし、前記2つ以上の領域にまたがる前記排他的な論理回路を前記インタフェイス論理ブロックとすることが望ましい。

30

【0036】

上記発明によれば、論理回路の重複論理ゲートが存在しても、それらの論理機能を変更することなく論理回路を適宜分離することが可能となり、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にすることが可能になる。

40

【0037】

また、前記インタフェイス論理ブロックレイアウト設計手段は、前記配線のレイアウト設計の結果から得られるブロック間配線遅延を考慮して、全てのインタフェイス論理ブロックのレイアウト設計を一括して行うことが望ましい。

【0038】

上記発明によれば、実際の領域間配線結果に基づく正確な遅延を考慮して、複数の領域にまたがるインタフェイス論理ブロック集合全体を一括して最適化するため、従来の分割設計が持つ、分割損による、タイミング設計収束性悪化問題を解決することができる。

【発明の効果】

50

【0039】

本発明によれば、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にすることが可能になる。

【発明を実施するための最良の形態】

【0040】

次に、本発明の実施例について図面を参照して詳細に説明する。

【0041】

図1は、本発明の一実施例の、半導体集積回路の設計装置（以下「設計装置」と表記する）1を示したブロック図である。

【0042】

図1において、設計装置1は、チップ情報入力手段1-01と、仮ゲート配置手段1-02と、分割格子設定手段1-03と、FF入力論理コーンと分割格子を基準としたゲートグルーピング手段（以下「グルーピング手段」と表記する）1-04と、コア論理ブロックとインタフェース論理ブロック分割手段（以下「論理ブロック分割手段」と表記する）1-05と、コア論理ブロックとインタフェース論理ブロックのレイアウト領域決定手段（以下「レイアウト領域決定手段」と表記する）1-06と、コア論理ブロックレイアウト設計手段1-07と、ブロック間配線レイアウト設計手段1-08と、インタフェース論理ブロックレイアウト設計手段1-09と、レイアウト設計統合手段1-10と、を含む。

【0043】

仮ゲート配置手段1-02と分割格子設定手段1-03とで、分割手段1-11が構成され、グルーピング手段1-04と論理ブロック分割手段1-05とレイアウト領域決定手段1-06とで、設定手段1-12が構成され、レイアウト領域決定手段1-06とブロック間配線レイアウト設計手段1-08とで、領域設定手段1-13が構成される。

【0044】

なお、設計装置1は、制御装置、記憶装置、入力装置および表示装置からなる一般的なコンピュータにより構成される。これらの各部については図示しない。

【0045】

上記の各手段は、記憶装置に格納されたプログラムにより動作する制御装置により、ROM、RAMなどの記憶装置上に構築されて制御される。

【0046】

チップ情報入力手段1-01は、設計対象となる半導体集積回路チップのゲートレベルのネットリストに関する情報（チップデータ）を、記憶装置（具体的には、仮ゲート配置手段1-02）に入力する。

【0047】

このネットリストに関する情報は、例えば、集積回路チップ内の部品の情報（例えば、複数の論理ゲートおよび複数のフリップフロップの情報）と、配線接続情報と、FF間のタイミング制約（制約値）とを含む。設計装置1は、ネットリストに関する情報に基づいて、半導体集積回路を設計する。

【0048】

仮ゲート配置手段1-02は、チップ情報入力手段1-01から受け付けたチップデータ（具体的には、部品情報）に基づいて、集積回路チップ内の論理ゲートおよびFFをチップ配置領域に仮配置して、チップ領域を生成する。

【0049】

分割格子設定手段1-03は、論理ゲートおよびFFが仮配置されたチップ領域全体を格子状の複数の領域（格子）に分割する分割線を設定する。

【0050】

グルーピング手段1-04は、移動手段の一例であり、ネットリストに関する情報（具体的には、配線接続情報）に基づいて、チップ領域全体に含まれる論理コーンを全て抽出し、各論理コーンについて、1つの格子内に含まれるか、複数の格子にまたがるかを判定する。

。

10

20

30

40

50

【 0 0 5 1 】

なお、論理コーンは、入力部および出力部として異なるFFが用いられ入力部と出力部の間にはFFが接続されずに少なくとも論理ゲートのいずれかが接続される論理回路の一例である。

【 0 0 5 2 】

1つの格子に含まれる論理コーン（以下「コア論理コーン」と表記する）は、コア論理ブロックを構成する要素の候補となり、複数の格子にまたがる論理コーン（以下「インタフェイス論理コーン」と表記する）は、インタフェイス論理ブロックを構成する要素の候補となる。

【 0 0 5 3 】

グルーピング手段1-04は、格子（領域）において、コア論理コーンの占める割合が大きくなるように、格子（領域）間で、論理コーンを移動する。

【 0 0 5 4 】

論理ブロック分割手段1-05は、コア論理コーンとインタフェイス論理コーンの重複部分回路（重複論理ゲート）が存在する場合、その重複部分回路を2重化することにより、コア論理コーンとインタフェイス論理コーンを排他的な論理集合（論理回路）に分離する。

【 0 0 5 5 】

レイアウト領域決定手段1-06は、論理ブロック決定手段の一例であって、1つの領域内に全体が含まれる、移動された論理コーンをコア論理ブロックとし、2つ以上の領域にまたがる、移動された論理コーンをインタフェイス論理ブロックとする。

【 0 0 5 6 】

また、レイアウト領域決定手段1-06は、1つの領域内に全体が含まれる排他的な論理コーンをコア論理ブロックとし、2つ以上の領域にまたがる排他的な論理コーンをインタフェイス論理ブロックとする。

【 0 0 5 7 】

また、レイアウト領域決定手段1-06は、コア論理ブロックおよびインタフェイス論理ブロックに基づいて、チップ領域に、コア論理ブロックレイアウト領域と、インタフェイス論理ブロックレイアウト領域と、を設定する。

【 0 0 5 8 】

コア論理ブロックレイアウト設計手段1-07は、ネットリストに関する情報（具体的には、配線接続情報とタイミング制約）に基づいて、コア論理ブロックのレイアウト設計をコア論理ブロックレイアウト領域で行う。

【 0 0 5 9 】

配線レイアウト設計手段1-08は、集積回路全体の領域に、領域間の配線レイアウト領域を設定し、ネットリストに関する情報（具体的には、配線接続情報）に基づいて、領域間の配線のレイアウト設計を領域間の配線レイアウト領域で行う。

【 0 0 6 0 】

インタフェイス論理ブロックレイアウト設計手段1-09は、ネットリストに関する情報（具体的には、配線接続情報とタイミング制約）と配線のレイアウト設計の結果に基づいて、インタフェイス論理ブロックのレイアウト設計を行う。

【 0 0 6 1 】

レイアウト設計統合手段1-10は、コア論理ブロックレイアウトと、インタフェイス論理ブロックレイアウトと、論理ブロック間レイアウトとを統合し、チップレイアウト設計を完了する。

【 0 0 6 2 】

分割手段1-11は、部品情報に基づいて複数の論理ゲートと複数のフリップフロップが配置されたチップ領域を、複数の領域に分割する。

【 0 0 6 3 】

設定手段1-12は、配線接続情報に基づいて、チップ領域から論理コーンを抽出し、1つの領域内に全体が含まれる論理コーンをコア論理ブロックとし、2つ以上の領域にまたが

10

20

30

40

50

る論理コーンをインタフェイス論理ブロックとする。

【0064】

領域設定手段1-13は、チップ領域に、コア論理ブロックレイアウト領域と、インタフェイス論理ブロックレイアウト領域と、領域間の配線レイアウト領域と、を設定する。

【0065】

次に、動作を説明する。

【0066】

具体的には、上記の各手段それぞれの動作を図2、図3、図4、図5、図6、図7、図8、図9および図10を用いて詳細に示す。

【0067】

チップ情報入力手段1-01は、設計対象となる回路チップのゲートレベルネットリストに関するチップデータを受け付けると、そのチップデータを仮ゲート配置手段1-02に提供する。

【0068】

仮ゲート配置手段1-02は、チップ情報入力手段1-01からチップデータを受け付けると、そのチップデータに基づいて、回路チップ内の論理ゲートおよびFFの仮配置をチップ配置領域に行き、論理ゲートおよびFFが仮配置されたチップ領域を生成する。

【0069】

この際、仮ゲート配置手段1-02は、回路に含まれる全論理ゲートおよび全FFを対象とした仮配置を一括処理する。この処理は、以降の手段で行う回路分割処理に、回路分割指針を与えるための概略的な仮配置であり、詳細な最適化処理を要しない。そのため、大規模回路でも一括処理可能である。

【0070】

仮ゲート配置手段1-02は、仮配置を終了すると、チップ領域とチップデータを分割格子設定手段1-03にて提供する。

【0071】

分割格子設定手段1-03は、チップ領域とチップデータを受け付けると、仮配置が実行されたチップ領域全体を格子状の複数の領域に分割する分割線を設定する。ここでの1格子のサイズを変更することにより、以降の処理で生成される論理ブロックのサイズを制御することが可能である。

【0072】

図2は、仮ゲート配置手段1-02と分割格子設定手段1-03の動作を説明するための図である。図2では、仮ゲート配置手段1-02は、FFと論理ゲートを一括してチップ上に配置し、分割格子設定手段1-03は、チップ領域全体に2×2の格子を設定している。

【0073】

分割格子設定手段1-03は、複数の領域を設定すると、その複数の領域とチップデータをグルーピング手段1-04に提供する。

【0074】

グルーピング手段1-04は、複数の領域とチップデータを受け付けると、チップデータに基づいて、複数の領域からなるチップ領域から論理コーンを全て抽出し、各論理コーンについて、1格子内に含まれる(コア論理コーン)か、複数の格子にまたがる(インタフェイス論理コーン)か、を判定する。

【0075】

この段階では、通常、コア論理コーンとインタフェイス論理コーンには部分的重複が存在する。

【0076】

タイミング設計を容易化するためには、コア論理ブロックに含まれるゲート数が多い方が望ましいため、グルーピング手段1-04は、コア論理コーンとなる論理コーンのサイズを増やすことを目的に、格子間のゲート移動を、論理コーン単位で行う。

【0077】

10

20

30

40

50

図3および図4は、グルーピング手段1-04の動作を説明するための図である。

【0078】

この段階では、図3に示すようにコア論理コーン3-01とインタフェイス論理コーン3-02には重複が生じている。この重複は、後段の論理2重化処理で除去される。

【0079】

図4は、コア論理コーンのサイズを増やすための動作、換言すると、領域において、コア論理コーンが占める割合が大きくなるための動作を説明するための図である。

【0080】

グルーピング手段1-04は、図4(a)に示した処理例1や図4(b)に示した処理例2を実行することによって、格子間の論理コーン移動を行う。このため、図4では、初期状態では、インタフェイス論理コーンであった論理コーンがコア論理コーンに変更されている。

10

【0081】

この処理では、コア論理コーンからインタフェイス論理コーンに変更される論理コーンも生じるが、グルーピング手段1-04は、この処理によって、コア論理コーンサイズの増減を計算し、コア論理コーンサイズが増えるようにゲートの移動を行う。

【0082】

グルーピング手段1-04は、論理コーンの移動を終了すると、その移動結果とチップデータを論理ブロック分割手段1-05に提供する。

【0083】

20

論理ブロック分割手段1-05は、移動結果とチップデータを受け付けると、コア論理コーンとインタフェイス論理コーンの重複部分回路を2重化することにより、コア論理コーンとインタフェイス論理コーンを排他的な論理集合に分離する。

【0084】

図5は、論理ブロック分割手段1-05の動作を説明するための図である。

【0085】

図5(a)において、論理ブロック分割手段1-05は、論理2重化対象回路5-01を2重化することにより、論理回路の動作を変更することなく、コア論理コーン5-02とインタフェイス論理コーン5-03を排他的な論理回路に分離する。

【0086】

30

より詳細には、図5(b)に示したように、論理ブロック分割手段1-05は、コア論理コーン5-02とインタフェイス論理コーン5-03の重複論理ゲート(2重化対象ゲート)5-04を多重化することにより、論理機能を変更することなく、コア論理コーン5-02とインタフェイス論理コーン5-03を分離する。

【0087】

なお、論理2重化対象回路5-01が存在しない場合、論理ブロック分割手段1-05は、論理2重化処理を行なわない。

【0088】

その後、論理ブロック分割手段1-05は、処理結果とチップデータを、レイアウト領域決定手段1-06に提供する。

40

【0089】

レイアウト領域決定手段1-06は、その処理結果とチップデータを受け付けると、コア論理ブロックとインタフェイス論理ブロックのそれぞれのレイアウト領域を設定する。

【0090】

具体的には、レイアウト領域決定手段1-06は、まず、各格子内のコア論理コーン集合をひとつにまとめた論理回路をコア論理ブロックとする。

【0091】

続いて、レイアウト領域決定手段1-06は、生成されたコア論理ブロックのサイズを基に、格子領域の中心にコア論理ブロックレイアウト領域を設定する。

【0092】

50

続いて、レイアウト領域決定手段1-06は、複数の格子領域にまたがるインタフェイス論理コーンに含まれる論理回路を、仮ゲート配置手段1-02で得られた配置位置を基に複数の格子領域に分割し、各格子領域内のインタフェイス論理コーンの一部分をまとめた論理回路をインタフェイス論理ブロックとする。

【0093】

続いて、レイアウト領域決定手段1-06は、インタフェイス論理ブロックレイアウト領域を、インタフェイス論理ブロックのサイズを基に、コア論理ブロックレイアウト領域の周辺に設定する。

【0094】

続いて、レイアウト領域決定手段1-06は、生成されたコア論理ブロックとインタフェイス論理ブロックの接点を、全てFFにする。

10

【0095】

図6は、レイアウト領域決定手段1-06の動作を説明するための図である。

【0096】

レイアウト領域決定手段1-06は、各格子領域内でコア論理コーン集合をまとめてコア論理ブロックを作成し、格子領域中心にコア論理ブロックレイアウト領域6-01を設定する。

【0097】

続いて、レイアウト領域決定手段1-06は、複数格子にまたがるインタフェイス論理コーンを格子間に分割し（分割されたインタフェイス論理コーン6-03参照）、各格子領域内でインタフェイス論理ブロックとしてまとめ、コア論理ブロックレイアウト領域6-01周辺に、インタフェイス論理ブロック用のレイアウト領域6-04を設定する。

20

【0098】

レイアウト領域決定手段1-06は、コア論理ブロックとインタフェイス論理ブロックの接点を全てFF（例えば、コア論理ブロック境界FF6-02）とする。このため、コア論理ブロックとインタフェイス論理ブロックをまたがるFF間論理パスは存在しない。

【0099】

その後、レイアウト領域決定手段1-06は、自己の処理結果とチップデータを、コア論理ブロックレイアウト設計手段1-07とブロック間配線レイアウト設計手段1-08に提供する。

【0100】

コア論理ブロックレイアウト設計手段1-07は、その処理結果とチップデータを受け付けると、その処理結果とチップデータ（配線接続情報とタイミング制約）に基づいて、各コア論理ブロック内のレイアウト設計を行う。

30

【0101】

コア論理ブロックの境界は全てFFになっているため、図14に示した論理パス104-03のような、他の論理ブロックに接続する論理パスは存在しない。そのため、コア論理ブロックレイアウト設計手段1-07は、チップデータに含まれるタイミング制約を分割する必要はなく、全てのコア論理ブロックについて、他のブロックと独立にレイアウト設計を行うことが可能である。

【0102】

各コア論理ブロック内のレイアウト設計は、そのサイズが集積回路全体に比べ小規模であるため、小メモリかつ短時間で行うことが可能である。

40

【0103】

図7は、コア論理ブロックレイアウト設計手段1-07の動作を説明するための図である。

【0104】

図7において、論理回路全体から、FFを入出力境界とする4つの小規模コア論理ブロックが生成され、コア論理ブロックレイアウト設計手段1-07は、各コア論理ブロックのレイアウト設計について、本来のタイミング制約を分割することなく、独立に行う。

【0105】

一方、ブロック間配線レイアウト設計手段1-08は、レイアウト領域決定手段1-06から処理結果とチップデータを受け付けると、その処理結果とチップデータ（配線接続情報）に

50

基づいて、論理ブロック間配線のレイアウト設計を行う。

【0106】

具体的には、ブロック間配線レイアウト設計手段1-08は、従来の「フラット設計手法」で障害となる論理ゲート配置が存在しないブロック間配線専用領域（領域間の配線レイアウト領域）を設け、そこで領域間の配線設計を行う。このため、ブロック間配線レイアウト設計手段1-08は、配線経路および遅延の制御を容易に行うことができる。配線経路および遅延の制御としては、例えば、配線遅延最小化を目的としての、迂回が少ない配線経路設計、配線幅の変更、配線間隔の変更、リピータの挿入、ブロック間配線要求量に応じてのブロック間隔の調整等が挙げられる。

【0107】

図8は、ブロック間配線レイアウト設計手段1-08の動作を説明するための図である。

【0108】

ブロック間配線レイアウト設計手段1-08は、論理ブロックをブラックボックス化（ブラックボックス化された論理ブロック8-02参照）し、論理ブロック間に配線専用領域（領域間の配線レイアウト領域）8-03を設定し、その領域8-03で、配線接続情報に基づいて、ブロック間配線経路を最短で配線し、遅延を最適化するためにリピータ102-01を挿入する。

【0109】

その後、ブロック間配線レイアウト設計手段1-08の処理は、その配線結果とチップデータを、インタフェース論理ブロックレイアウト設計手段1-09に提供する。

【0110】

インタフェース論理ブロックレイアウト設計手段1-09は、配線結果とチップデータを受け付けると、ブロック間配線レイアウト設計手段1-08が生成した論理ブロック間配線結果を固定し、配線接続情報と、タイミング制約と、ブロック配線結果から得られるブロック間配線遅延と、を考慮して、インタフェース論理ブロック集合全体のレイアウト設計を一括して行う。

【0111】

すなわち、インタフェース論理ブロックレイアウト設計手段1-09は、固定された論理ブロック間配線から計算される論理ブロック間遅延を本来のタイミング制約から減じた値をタイミング制約として、インタフェース論理ブロックのレイアウト設計を分割することなく一括で行う。

【0112】

例えば、本来のタイミング制約が10nano秒で、ブロック間配線遅延が4nano秒の場合、インタフェース論理ブロックレイアウト設計手段1-09は、6nano秒をタイミング制約として、インタフェース論理ブロックのレイアウト設計を行う。このとき、各インタフェース論理ブロックには、それが属する論理ブロック内のインタフェース論理ブロックレイアウト領域内で配置配線されるという制約が課される。

【0113】

図9は、インタフェース論理ブロックレイアウト設計手段1-09の動作を説明するための図である。

【0114】

インタフェース論理ブロックレイアウト設計手段1-09は、ブロック間レイアウト（配線）を固定し、その部分の遅延を考慮して、インタフェース論理ブロック全体のレイアウト設計を行う。この段階では、論理ブロックの大部分は、コア論理ブロックとしてブラックボックス化されているため、インタフェース論理ブロックのサイズは削減されており、このような一括処理が可能である。

【0115】

インタフェース論理ブロックレイアウト設計手段1-09の処理が終了すると、レイアウト設計統合手段1-10は、前段までの処理で得られたコア論理ブロックレイアウト、インタフェース論理ブロックレイアウト、論理ブロック間レイアウトを統合し、チップレイアウト設計を完了する。

10

20

30

40

50

【 0 1 1 6 】

図 1 0 は、レイアウト設計統合手段1-10の動作を説明するための図である。

【 0 1 1 7 】

レイアウト設計統合手段1-10は、コア論理ブロックレイアウト、インタフェース論理ブロックレイアウト、論理ブロック間レイアウトを統合し、チップ全体のレイアウト設計を完了している。

【 0 1 1 8 】

本実施例によれば、以下に記載する効果を奏する。

【 0 1 1 9 】

回路全体を複数の論理ブロック（領域）に分割し、論理ブロックレイアウト領域と論理ブロック間レイアウト領域を設定して、レイアウト設計を行うため、ブロック間配線時、論理ゲート配置が障害にならず、配線幅、配線間隔、リピータ挿入等の、ブロック間配線遅延を最小化するための論理ブロック間配線構造の制御を行うことが容易となる。

10

【 0 1 2 0 】

コア論理ブロックは、入出力切り口が全てFFとなっており、図 1 4 に示すようなコア論理間にまたがる論理パスは存在しないため、FF間のタイミング制約を分割することなく、各コア論理ブロックの設計を、他の論理ブロックと独立に行うことが可能である。

インタフェース論理ブロックの設計については、実際のブロック間配線結果に基づく正確な遅延を考慮して、複数の論理ブロックにまたがるインタフェース論理ブロック集合全体を一括して最適化できるため、従来の分割設計が持つ、分割損による、タイミング設計収束性悪化問題を解決することができる。

20

【 0 1 2 1 】

この一括最適化は、論理ブロックをコア論理ブロックとインタフェース論理ブロックに分割し、インタフェース論理ブロックのサイズを削減しているため可能となる。

【 0 1 2 2 】

また、本実施例によれば、以下の作用効果を奏する。

【 0 1 2 3 】

本実施例によれば、コア論理ブロック、インタフェース論理ブロック、および、領域間の配線のレイアウト設計は、個別に行われる。このため、領域間配線時、論理ゲート配置が障害にならず、配線幅、配線間隔、リピータ挿入等の、領域間配線遅延を最小化するための配線構造の制御を行うことが容易になる。また、設計対象を分割して設計するため、大規模の集積回路を、小メモリ、短時間で行うことができる。

30

【 0 1 2 4 】

コア論理ブロックは、入出力部が全てFFとなっているため、FF間のタイミング制約を分割することなく、各コア論理ブロックの設計を、他の論理ブロックと独立に行うことが可能になる。

【 0 1 2 5 】

インタフェース論理ブロックの設計については、例えば、実際の領域間配線結果に基づく正確な遅延を考慮して、複数の領域にまたがるインタフェース論理ブロック集合全体を一括して最適化できるため、従来の分割設計が持つ、分割損による、タイミング設計収束性悪化問題を解決することができる。

40

【 0 1 2 6 】

よって、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にすることが可能になる。

【 0 1 2 7 】

また、本実施例では、コア論理ブロックと領域間の配線とインタフェース論理ブロックとが、それぞれに対応するレイアウト領域で設計される。

【 0 1 2 8 】

この場合、領域間配線時、論理ゲート配置が障害にならず、配線幅、配線間隔、リピータ挿入等の、領域間配線遅延を最小化するための配線構造の制御を行うことが容易となる

50

。

【0129】

また、グルーピング手段1-04は、領域に全体が含まれる論理コーンの占める割合が大きくなるように、領域間で論理コーンを移動する。

【0130】

この場合、インタフェース論理ブロックのサイズを小さくすることが可能になる。よって、インタフェース論理ブロック全体の一括最適化を容易にすることが可能になる。

【0131】

また、論理ブロック分割手段1-05は、1つの領域に全体が含まれる論理コーンと、2つ以上の領域にまたがる論理コーンと、の重複論理ゲートが存在する場合、その重複論理ゲートを2重化して、1つの領域に全体が含まれる論理コーンと、2つ以上の領域にまたがる論理コーンと、を互いに排他的になるように分離する。

10

【0132】

この場合、重複論理ゲートが存在しても、それらの論理機能を変更することなく論理コーンを適宜分離することが可能となり、論理回路レイアウト設計の全体最適性を失うことなく、タイミング設計で重要となるゲート間配線の見積もり、制御性を容易にすることが可能になる。

【0133】

また、本実施例では、インタフェース論理ブロックレイアウト設計手段1-09は、配線のレイアウト設計の結果から得られるブロック間配線遅延を考慮して、全てのインタフェース論理ブロックのレイアウト設計を一括して行う。

20

【0134】

この場合、実際の領域間配線結果に基づく正確な遅延を考慮して、複数の領域にまたがるインタフェース論理ブロック集合全体を一括して最適化するため、従来の分割設計が持つ、分割損による、タイミング設計収束性悪化問題を解決することができる。

【0135】

以上説明した実施例において、図示した構成は単なる一例であって、本発明はその構成に限定されるものではない。

【図面の簡単な説明】

【0136】

30

【図1】本発明の実施形態に係る半導体集積回路の階層設計装置の一実施例の構成を示すブロック図である。

【図2】図1の半導体集積回路の設計装置の仮ゲート配置手段1-02と分割格子設定手段1-03の動作を示す図である。

【図3】図1の半導体集積回路の設計装置のFF入力論理コーンと分割格子を基準としたゲートグルーピング手段1-04の動作を示す図である。

【図4】図1の半導体集積回路の設計装置のFF入力論理コーンと分割格子を基準としたゲートグルーピング手段1-04の動作を示す図である。

【図5】図1の半導体集積回路の設計装置のコア論理ブロックとインタフェース論理ブロック分割手段1-05の動作を示す図である。

40

【図6】図1の半導体集積回路の設計装置のコア論理ブロックとインタフェース論理ブロックのレイアウト領域決定手段1-06の動作を示す図である。

【図7】図1の半導体集積回路の設計装置のコア論理ブロックレイアウト設計手段1-07の動作を示す図である。

【図8】図1の半導体集積回路の設計装置のブロック間配線レイアウト設計手段1-08の動作を示す図である。

【図9】図1の半導体集積回路の設計装置のインタフェース論理ブロックレイアウト設計手段1-09の動作を示す図である。

【図10】図1の半導体集積回路の設計装置のレイアウト設計統合手段1-10の動作を示す図である。

50

【図 1 1】LSIのタイミング設計と本明細書で使用する用語に関する説明図である。

【図 1 2】従来手法のひとつであるフラットレイアウト設計手法とその課題を示す図である。

【図 1 3】従来手法のひとつである分割設計手法を示す図である。

【図 1 4】従来手法のひとつである分割設計手法の課題を示す図である。

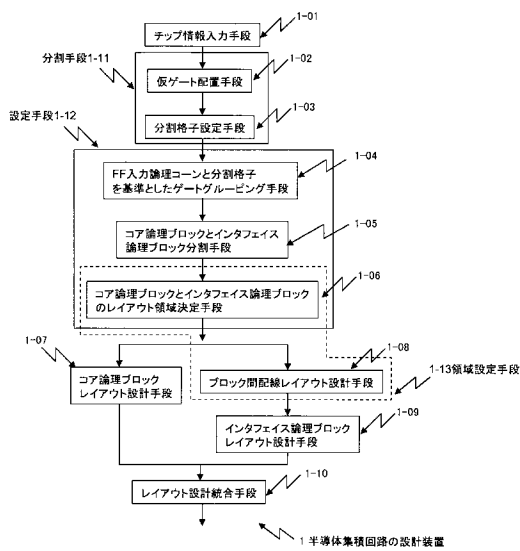
【符号の説明】

【 0 1 3 7 】

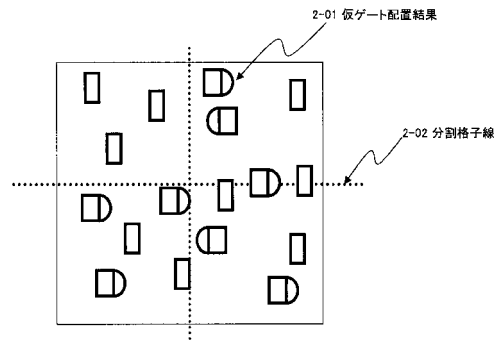
101-01	フリップフロップ(FF)	
101-02	論理ゲート	
101-03	信号線	10
101-04	論理パス	
101-05	タイミング制約	
101-06	フリップフロップ(FF)入力論理コーン	
102-01	リピータ	
102-02	信号配線	
102-03	迂回配線	
103-01	論理ブロックレイアウト領域	
103-02	論理ブロック間配線領域	
103-03	論理ブロック間信号配線	
104-01	始点FF	20
104-02	終点FF	
104-03	複数の論理ブロックを跨るパス	
1	半導体集積回路の設計装置	
1-01	チップ情報入力手段	
1-02	仮ゲート配置手段	
1-03	分割格子設定手段	
1-04	FF入力論理コーンと分割格子を基準としたゲートグルーピング手段	
1-05	コア論理ブロックとインタフェイス論理ブロック分割手段	
1-06	コア論理ブロックとインタフェイス論理ブロックのレイアウト領域決定	
手段		30
1-07	コア論理ブロックレイアウト設計手段	
1-08	ブロック間配線レイアウト設計手段	
1-09	インタフェイス論理ブロックレイアウト設計手段	
1-10	レイアウト設計統合手段	
1-11	分割手段	
1-12	設定手段	
1-13	領域設定手段	
2-01	仮ゲート配置結果	
2-02	分割格子線	
3-01	格子領域内に閉じた論理コーン(コア論理コーン)	40
3-02	格子領域を跨る論理コーン(インタフェイス論理コーン)	
4-01	コア論理コーンサイズを増やすために移動する論理回路	
4-02	論理コーンの移動による格子領域内に閉じた論理コーンサイズ増加	
5-01	論理2重化対象回路	
5-02	コア論理コーン	
5-03	インタフェイス論理コーン	
5-04	2重化対象ゲート	
6-01	コア論理ブロックレイアウト領域	
6-02	コア論理ブロック境界フリップフロップ	
6-03	分割されたインタフェイス論理コーン	50

- 6-04 インタフェイス論理ブロックレイアウト領域
- 6-05 ブロック間信号
- 7-01 コア論理ブロックレイアウト設計
- 8-01 ブロック間配線レイアウト設計
- 8-02 ブラックボックス化された論理ブロック
- 9-01 ブラックボックス化されたコア論理ブロック
- 9-02 インタフェイス論理ブロックレイアウト設計
- 9-03 固定ブロック間配線レイアウト
- 9-04 同時最適化対象となるインタフェイス論理ブロック
- 10 統合されたレイアウト設計

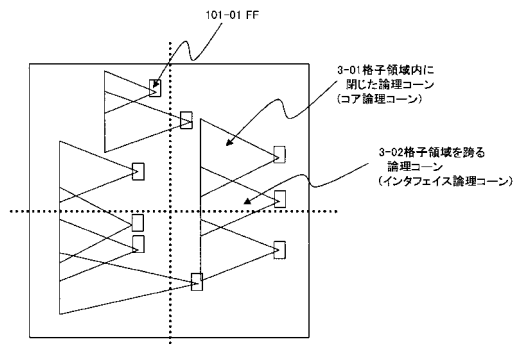
【図1】



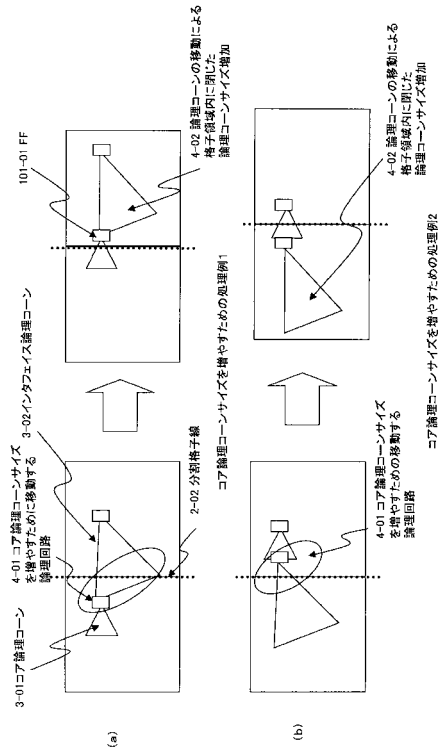
【図2】



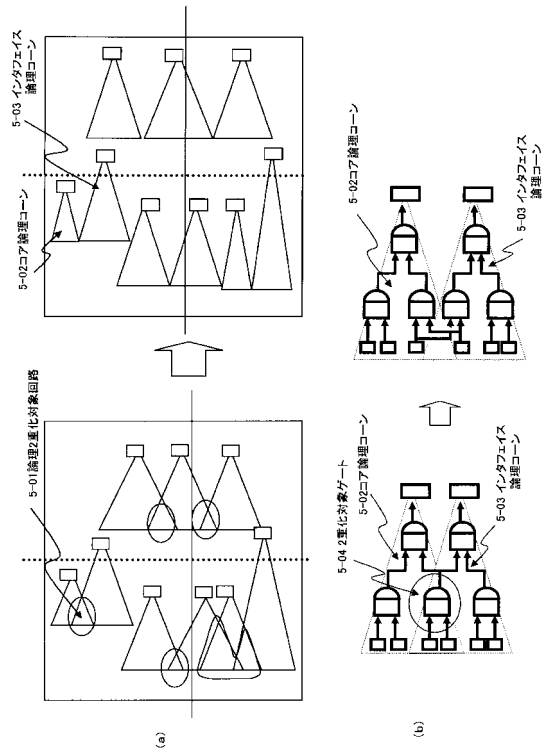
【図3】



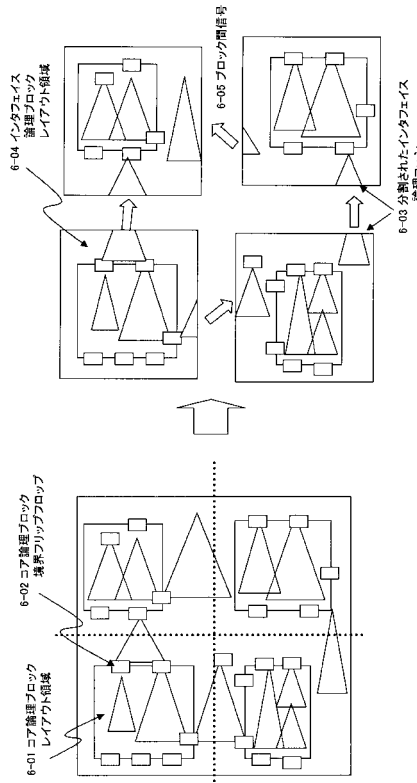
【 図 4 】



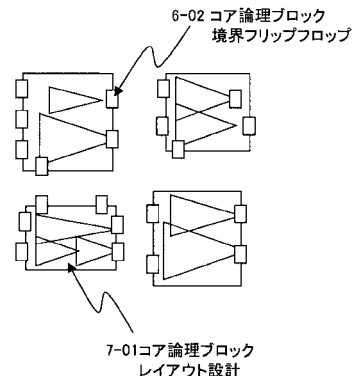
【 図 5 】



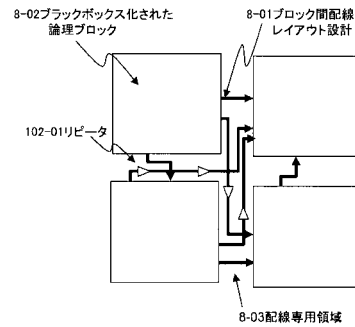
【 図 6 】



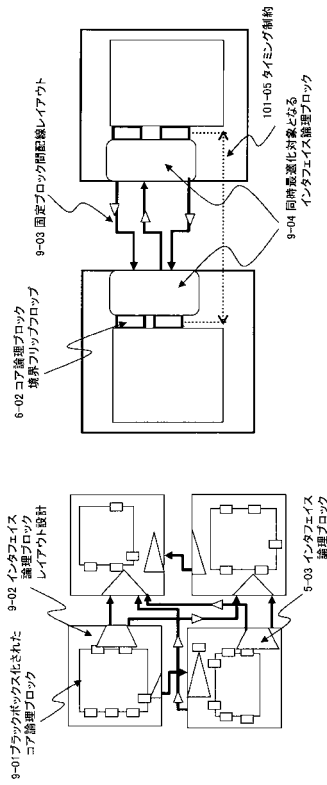
【 図 7 】



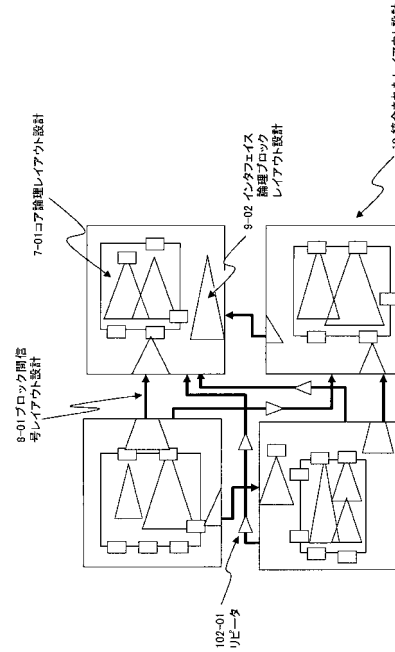
【 図 8 】



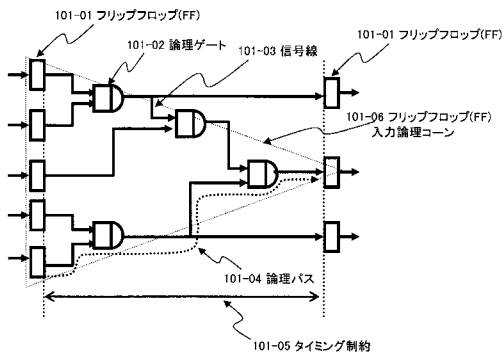
【 図 9 】



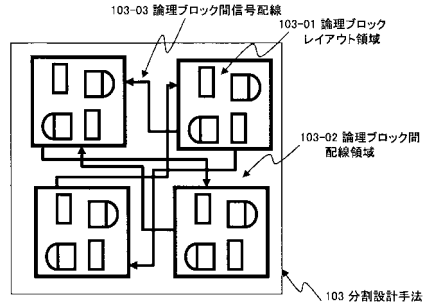
【 図 10 】



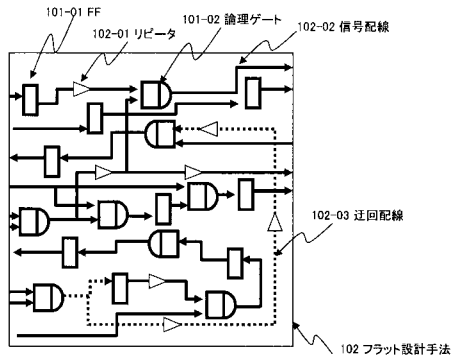
【 図 11 】



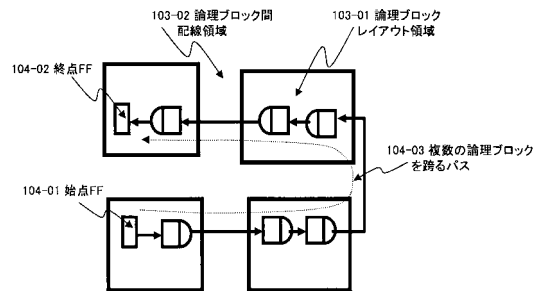
【 図 13 】



【 図 12 】



【 図 14 】



フロントページの続き

- (56)参考文献 特開2004-192227(JP,A)
特開2004-302819(JP,A)
特開2006-338090(JP,A)
特開2004-13248(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 17/50
CSDB(日本国特許庁)
JSTPlus(JDreamII)