設計自動化　39－5
（1987. 10. 15）

論理動作抽出による配置制御手法について

服部俊洋　　三浦地平　　宮本俊介

（株）日立製作所　中央研究所

従来、論理自動生成システムと自動レイアウトシステムは各々別個に実行されていたが、より高品質のＬＳＩを自動設計するためには両者の有機的な結合が必要である。本論文では、論理設計者が意識している論理動作、論理構造を示す情報を論理自動生成システムから得て、レイアウトを最適化するための配置制約を出力し、結果として面積縮小をねらった試みについて延べる。評価実験により、データ転送が主たる機能である論理については、ブロック面積を９０％に削減できた。

A Logic Constraints Dicision Algorithm with Synthesized Logic Behavior

Toshihiro HATTORI, Chihei MIURA, Shunsuke MIYAMOTO

Central Research Laboratory, Hitachi Ltd.
1-280, Higashi-Koigakubo, Kokubunji, Tokyo 185, Japan

In a conventional DA system, logic synthesizer and automatic layout system are handled separately. However, those systems should utilize mutual output imformation to minimize LSI chip area. In this article, chip area reduction algorithm is described. Placement constraints are automatically generated from logic behavior and structure imformations extracted from logic synthesizer. In our experiment, block area is reduced to 90% using this method.

１．はじめに

近年、論理ＬＳＩや計算機の高集積化に伴い、ＤＡシステムはＬＳＩ設計にとって不可欠なものとなっている。特に、自動レイアウトシステムは人手設計と同等の性能を発揮し、実用化されている。[1] 一方、論理設計工数の増大に対処するため、論理設計の自動化を図る論理自動生成システムの研究が盛んに進められている。[2-7]

各設計工程各々の自動化はそれなりの効果はあげるが、それだけではＬＳＩ設計工程全体の効率化は図れない。例えば高木氏らによる研究では、人手により論理構造を指定して自動レイアウト設計を行い、機能ブロックの面積が削減出来ることが示されている。[8-9] 論理設計時の情報を考慮したレイアウト、実装条件を考慮した論理設計は過去、設計者が行ってきた方法である。

我々のねらいは、論理自動生成システムと自動レイアウトシステムを有機的に結合させ、ＬＳＩ設計全体の効率化を図り、最適なレイアウト結果を得ることにある。

本稿ではブロック内レイアウトを対象に、論理構造を保存する配置制約決定処理について、処理方式と評価結果を報告する。本処理では、データレジスタやセレクタ等のセルの概略初期配置位置を決定することによりスタンダードセル方式のブロック内セルレイアウトの最適化が可能である。

２．自動ＬＳＩ設計システム

２．１　概要

本節では、自動ＬＳＩ設計の概要と、現状の問題点について述べる。自動ＬＳＩ設計の概略フローと、各設計工程でのＤＡシステムを図２．１に示す。基本的に自動ＬＳＩ設計システムは２つの構成要素から構成されている。

第１の要素は論理自動生成システムである。我々のシステムでは、ブロック図とタイムチャートレベルに対応する高位論理記述から、ブール式レベル論理を自動生成する。[2-4] 本分野の研究は盛んに行なわれているが、現在のところは、生成結果のブール式レベルまたはゲートレベル論理を得ることが最大の目的であり、後続のシステムを制御しようとする試みはなされていない。

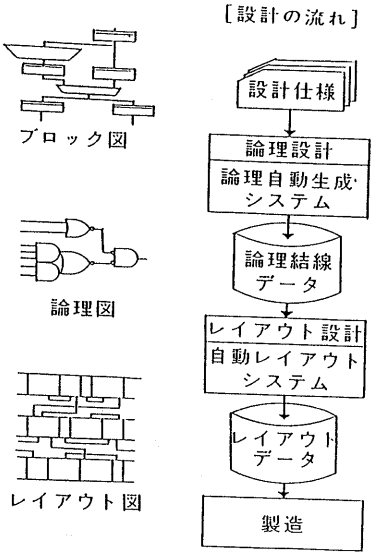

図２．１　ＬＳＩ設計のフロー

しかしながら、高位論理記述には、論理設計者の設計意図が自然な形で反映されている。人手でＬＳＩ設計を行う際には、設計者はこの情報を暗黙のうちにレイアウト設計で使っているといえる。

第２の要素は自動レイアウトシステムである。[1] 論理設計の結果である結線データを入力として、自動配置、自動配線を行う。現在の論理ＬＳＩのレイアウト設計においては、スタンダードセル方式による設計が主流である。スタンダードセル方式では、論理要素であるＮＡＮＤやインバータ等のセルをあらかじめ設計しておき、論理設計の結果に従ってセルを配置し、端子間を配線する。また、レイアウト設計においては階層化設計が行なわれている。つまり、全体論理をブロックに分割し、各々をスタンダードセル方式によってレイアウトする。それ等のブロックを配置し、配線してチップ全体のレイアウトを得ている。ブロック間、ブロック内の自動配置では、要素間の結線の本数に基づいて要素の配置を決定するのが一般的である。

しかしながら、設計者が人手でレイアウトを行う場合は、結線本数だけでなく各要素の論理的な意味合いを利用して配置を決定している。よって、論理設計時の情報も用いて、レイアウトを行うようにすれば、より人手設計に近い最適化されたＬＳＩ設計が可能となる。

以上の理由から、我々は自動論理設計と自動レイアウト設計を有機的に結合するシステムの研究を行って

いる。有機的な結合の本来の意味は、単に論理自動生成システムの出力する設計結果がそのまま自動レイアウトシステムの入力となるように結合するというだけでなく、論理設計時の情報とレイアウト設計時の情報を各々のシステムが相互利用することによりＬＳＩ設計全体の最適化を図ることにある。

２．２　論理動作に基づくレイアウト設計の必要性

本節では、論理動作、構造を保存するレイアウトの必要性について述べる。

現在の自動レイアウトシステムは、論理要素の記述と論理要素の端子間の結線関係から、結合度の大きなものを近くに配置することで、よりよいレイアウト結果を得ることを目的としている。ブロックレイアウトとブロック内セルレイアウトの処理は基本的な考え方は同じであるので、以下ではブロック内レイアウトについて検討して行く。

現在のブロック内セル自動配置システムでは、

（１）結線関係に基づいてセルをグループ化し（クラスタ生成）、セルの初期配置位置を決定する、
（２）繰り返し改善法によって、仮想配線長が極小になるように配置改善を行う、

という処理を行っている。２次元クラスタリング手法によるセル配置では、配置配線後のブロック面積から見て、人手設計と同等の性能が得られている。[1)]

しかし現在のレイアウトシステムにおいても、対象とする論理回路によっては、人手設計したブロックよりも自動レイアウトしたブロックの面積が大きくなってしまう場合が存在する。その原因を考えてみると、設計者は結線関係だけに基づいて配置を行っているのではなく、論理設計時の情報にも基づいて論理要素の初期配置を決定していると考えられる。したがって、論理設計情報を利用する自動レイアウトを行うと、短い処理時間でよりよいレイアウト結果が得られるはずである。

以上述べたことは、ブロック配置処理においてより顕著である。フロアプランとよばれるブロック配置処理では、論理設計者が論理情報に基づいてブロックの概略的な配置位置を決定しているのが現状である。ブロック配置処理を自動化するためには、自動論理分割を含め論理情報の活用が重要である。

この様な論理情報、すなわち設計対象の論理動作や設計結果の論理構造を設計者に記述させるのは煩雑である。しかし、論理自動生成システムの構築により、その入力である高位論理記述や自動生成時の内部情報から容易に論理動作、構造を抽出することが可能になってきた。

以上述べた背景に基づき、我々は論理自動生成システムから論理動作や論理構造を自動抽出して、自動レイアウトシステムを最適化するための制御情報生成方法の研究を行っている。

２．３　論理設計情報の抽出

本節では、論理設計者が行う配置手法を自動的に行うために、抽出すべき論理設計情報と各々の内容について説明する。

（１）論理属性
論理回路上の論理要素がデータの転送のために用いられるデータ系論理要素であるのか、または、制御を司るために用いられる制御系論理要素であるかを示す属性である。データ系論理要素の例としては、レジスタ、データ転送の競合阻止のため自動生成されたセレクタなどが挙げられる。レジスタ等については論理自動生成システムの入力記述であるＢ²ＤＬ（Block-diagram and Behavior Oriented Discription Language）記述内で明示宣言されるため、抽出は容易である。セレクタは自動生成システム内で作り出された際に識別される。

（２）束情報
論理要素であるラッチ群が同一の制御線によって制御されるレジスタを構成していることを示す情報である。例えば、８ビット幅のレジスタＡはＢ²ＤＬ記述上では一括してＡ（０：７）という形で記述されている。従来は、自動レイアウトシステムの入力となるゲートレベル論理に展開された際に個々のラッチとして表現されており、どのラッチが元々同じレジスタを構成していたかを示す情報が欠落していた。

（３）信号伝達方向
信号がどの論理要素から、どの論理要素に向かって流れているかを示す情報である。従来の結線情報では端子間の結線関係のみ

を記述しており信号の流れる方向は明記されていなかった。本処理ではデータの流れる方向を明記した形で論理情報を得る。

これ等の情報は論理自動生成システムの入力である高位論理記述を解析することによって得られるため、特別にゲートレベル論理記述から再構成することは無駄な処理である。この他にも、論理動作を表現する方法はあるかもしれないが、論理動作を論理構造に集約した形で、後の処理に役立てる。

3．論理構造を保存する配置制約決定処理

本章では、論理構造を保存するための配置制約を求める処理方式について述べる。

本処理では論理結線情報と論理設計情報とに基づいて、論理構造を保存する配置制約決定処理を行う。今回の実験ではデータ系論理要素を対象に、ブロック内での概略的な初期配置位置の制約を与えることとした。この配置制約と論理結線情報を、自動レイアウトシステムの入力とし、データ系論理要素の初期配置位置は配置制約に従い、制御系論理要素の初期配置位置は従来法どおりの手法で求める方法をとった。この初期配置に対して繰り返し改善処理を行うことにより最適なセル配置を決定することができる。

3．1　処理方法の概略

論理構造保存配置処理のフローを図3．1に示す。本処理は3つの部分から構成される。

第1の処理は、論理結線情報と論理設計情報を入力として、データフロー構造を出力するデータフロー構成処理である。データフロー構造とは、データ系論理におけるデータの流れを優先度の大きいものから階層的に構成したものである。第2の処理では、このデータフロー構造に基づいて、最も近接して配置すべき論理要素から近くに、しかも、データの流れを保存するように順序付けを行う。これが1次元配置順序決定処理である。この処理の結果、全てのデータ系要素についての1次元配置順序が得られる。第3の処理として、この1次元配置順序に基づいてデータ系要素のブロック内の概略的な配置制約を決定する2次元配置制約決定処理を行う。以下の節で各処理を詳細に説明する。

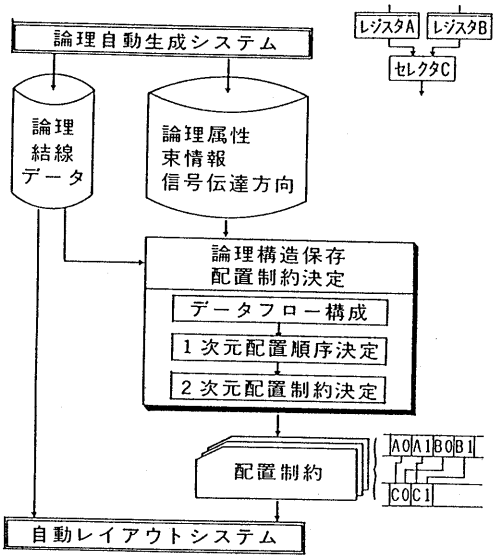

図3．1　配置制約決定処理のフロー

3．2　データフロー構成処理

本節では、データの流れを表現するデータフロー構造を生成するデータフロー構成処理の概要について述べる。

今回は、レイアウト設計に活用すべき論理構造として、データ系論理におけるデータの流れと同一レジスタを構成するラッチ群の集合情報を考えた。ここで、

（1）データ系論理要素か否かの情報、
（2）同一レジスタを構成するラッチ群の情報、

は論理設計情報として論理自動生成システムから得ている。図3．2に論理自動生成システムの入力であるB^2DL記述の一例を示す。この記述中でA（0：7）はレジスタAが8ビット幅のレジスタであることを表現している。このB^2DL記述中に明記されたレジスタは全てデータ系論理要素である。

UFPLANでは、データの流れを示すものとして基本構造を定義した。基本構造は以下の3つの種類に分類される。

（1）直列
データ系論理要素間にデータが一対一で転送される構造。

（2）分岐
データが1つの論理要素から複数の論理要素に転送される構造。
（3）収れん
データが複数の論理要素から、選択的に1つの論理要素に転送される構造。

図3．3は、図3．2で示したB²DL記述が表現するブロック図上での基本構造を図示したものである。例えば、レジスタFはレジスタHと直列の基本構造を構成している。また、レジスタAとレジスタBは収れんの基本構造を構成している。そして、収れんの基本構造を構成するレジスタA，レジスタBのグループと、直列の基本構造を構成するセレクタC，レジスタDのグループは直列の基本構造を構成していると言える。この様に、基本構造は階層的に定義することが可能である。

そこで、データフロー構成処理では、より近接して配置すべき基本構造から優先的にグループ化を行ない、階層的に木構造を作る。以上により、データの流れを表現するデータフロー構造が構成される。データフロー構成処理では、以下の2つのルールに従ってデータフロー構造を構成する。

（1） 基本構造が重なりあった場合、直列、分岐、収れんの順で優先する。
（2） 同種の基本構造では、構成論理要素のビット幅の大きいものを優先する。

処理に従って、図3．3の回路から構成したデータフロー構造を図3．4に示す。データフロー構造はグループ化による階層構造と同時に、データの流れる方向を情報として持っている。

3．3 1次元配置順序決定処理

本節では、データフロー構造に基づいて論理要素の1次元配置順序を求める処理の内容を記述する。

図3．4に示したデータフロー構造は、データの流れる逐次方向とデータが並行して流れる並列方向との2方向からなる2次元の位置関係を示している。この2次元の位置関係から直接、矩形のブロック内での配置位置を最適に決定することは困難である。2次元にセルを配置した後、セルを移動させブロック形状を整形する方法では、優先的に近接配置すべきセル群が離

```
BLOCK: REI;
  INPUT:     A1(0:7)  :DATA,
             A2(0:7)  :DATA,
             SEL      :COMMAND,
             TK0      :COMMAND;
  OUTPUT:    O1(0:7)  :DATA;
  REGISTER:  A(0:7),
             B(0:7),
             D(0:7),
             E(0:7),
             F(0:7),
             G(0:7),
             H(0:7);
  SEQUENCE: REISEQ;
       << CONSTANT >>
    #1:TK0    A:=IN1;
              B:=IN2;
    #2:TK0    IF SEL THEN D:=A;
                      ELSE D:=B;
    #3:TK0    E:=D;
              F:=D;
              G:=D;
    #4:TK0    H:=F;
    #5:TK0    O1:=H;
    END;
END: REI;
```

図3．2 B²DL記述例

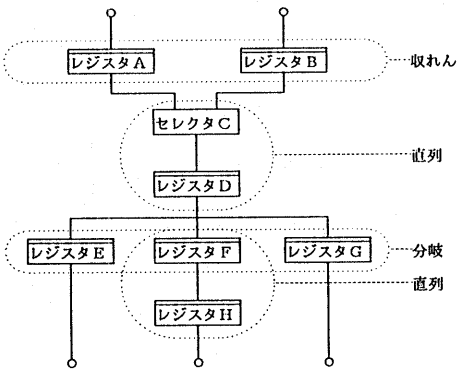

図3．3 基本構造の例

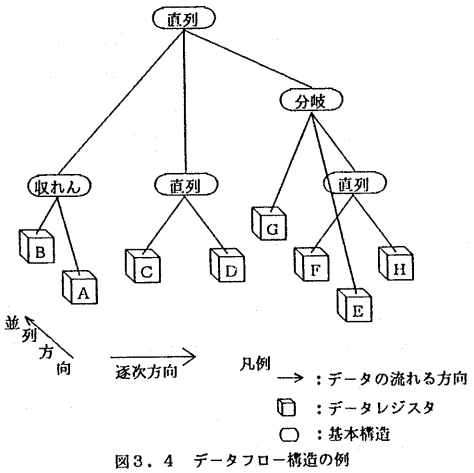

図3．4 データフロー構造の例

れて配置されることがある。そこで本実験では、得られた２次元の位置関係に基づいて、先ず最適な１次元の順序を求める。

並行関係も含む基本構造を１次元的に如何に順序付けるかを表３．１に示す。１次元配置順序決定処理では、この表に従って基本構造を構成する要素を順序付けていく。この際に、階層化されたデータフロー構造の下層の部分、つまり、より優先的にグループ化すべき基本構造から順序付けることによって、一意的に１次元配置順序を決定出来る。図３．４のデータフロー構造から決定されるラッチ群の１次元配置順序を図３．５に示す。

表３．１　基本構造の種類と一次元配置順序の対応

名称	基本構造の概略図	一次元配置順序
直列	A B C	A B C
分岐	A B　C	A B C
収れん	A　B C	A B C

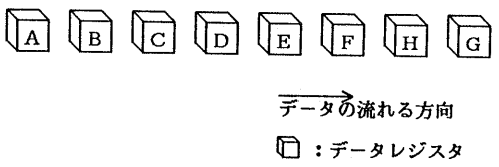

図３．５　１次元配置順序の例

３．４　２次元配置制約決定処理

本節では、今まで述べた処理で得られた１次元配置順序に従いデータ系論理要素のブロック上の概略的な配置制約を決定する処理について述べる。

スタンダードセル方式において、レジスタ単位の１次元配置順序が与えられているとき、それを２次元的に如何にブロック上に配置すれば、最終的なブロック面積を小さく出来るかには選択の予知は多い。その方法には、以下の２つの代表的な選択肢がある。

（１）データ線の方向
　（ａ）データ線をセル列に平行に（一般的には、横方向に）する。
　（ｂ）データ線をセル列に垂直に（一般的には、縦方向に）する。
（２）レジスタグループの扱い
　（ａ）レジスタを構成しているラッチ群を集中して配置する。（制御線が短くなる。）
　（ｂ）ビット位置が同じラッチ群を集中して配置する。（データ線が短くなる）

各方法を比較検討した結果、本処理においては、

（１）データ線をセル列に垂直にする、
（２）ビット位置が同じラッチ群を集中して配置する、

という手法で２次元配置を決定する。図３．５に２次元配置方法の概要を示す。

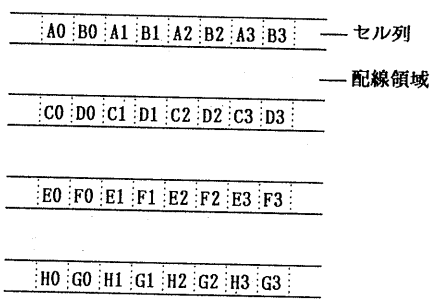

図３．６　２次元配置制約の例

４．評価

４．１　実験対象

今回提案した配置制約に基づいたセルレイアウト結果と、従来の２次元クラスタリング法によるセルレイアウト結果を比較し、論理構造に基づく配置制約決定処理の有効性を評価した。実験対象回路としてＣＭＯＳマイクロプロセッサの部分論理を選んだ。実験対象回路の諸元を表４．１にまとめる。

表４．１　評価実験対象回路の諸元

実験対象回路	マイクロプロセッサ
回路の機能	データ転送論理と付随する制御論理
ゲート数	１０００
セル数	４００
レジスタ数	２００

実験の手順は以下のとおりである。

（１）対象回路の機能をB^2DL言語で記述する。

（２）論理自動生成システムにより論理を生成する。同時に、論理設計情報も出力する。

（３）配置制約を自動生成する。

（４）論理結線情報と配置制約を入力として自動レイアウトを行う。

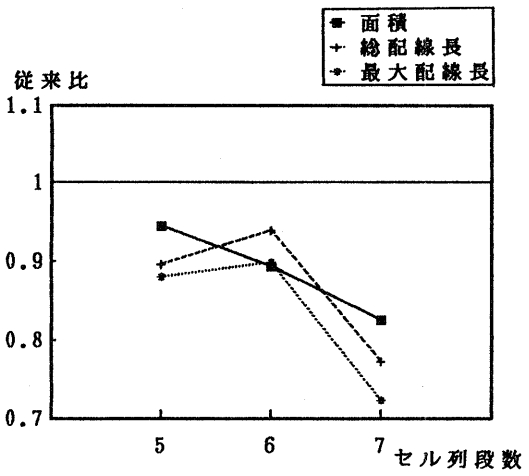

図４．１　評価実験結果

４．２　実験結果

今回実験した配置制約に基づくレイアウト結果を、論理結線情報のみを入力とした従来レイアウト結果と比較する。従来法による自動レイアウトは２次元クラスタリング法とネットバランス法を用いている。[1)] ブロック面積、総配線長、最大配線長の従来比を図４．１に示す。フロアプランによりブロックの最適形状が変わるため、種々のセル列段数において従来法と比較した。レイアウトモデルは信号線３層モデルである。

平均すると、ブロック面積では従来比８９．８％、総配線長では従来比８４．５％、最大配線長では従来比７９．７％に削減することができ、データ系要素が主な構成要素となる論理での論理構造保存配置制約決定処理の有効性が確認できた。

４．３　計算機処理時間

新手法を用いたときのセル初期配置に要する計算機時間を従来法と比較した。今回の実験では、実験システムによるデータ系論理の配置制約決定処理と、従来法による制御論理の初期配置処理の処理時間の和を初期配置決定に要するＣＰＵ時間とした。結果を表４．２に示す。明らかに論理構造を利用した新手法が優位である。

表４．２　初期配置決定に必要な計算機処理時間比

手法	処理時間比
従来法 （クラスタリング処理）	１．０
新手法 （論理構造保存配置制約決定 ＋クラスタリング処理）	０．０１６

５．おわりに

本稿では、論理構造を保存する配置制約決定処理について述べた。新手法では論理自動生成システムが出力する論理設計情報を入力としてレイアウトを最適化するための配置制約を出力する。１Ｋゲート規模の論理を対象とした実験ではブロック面積を従来の９０％に削減し、初期配置決定に必要な計算時間を１．６％までに削減できた。

現段階では、論理動作、論理構造の一部をレイアウト設計に活用しているに過ぎない。今後は、より抽象的な論理動作を利用した、論理設計とレイアウト設計の有機的な結合方式の検討を試みる予定である。

参考文献

[1] T. Kozawa, et al.: Automatic Placement Algorithms for High Packing density VLSI; proc. of 20th DAC, pp175-181, (1983)

[2] T. Shimizu, et al.: High level Logic Synthesis Algorithm with Global Minimization Process; ICCD'86, pp. 15-18, (1986)

[3] T. Shimizu, et al.: A Logic Synthesis Algorithm for the Design of A High Performance Processor; ISCAS'85, pp. 407-410, (1985)

[4] 宮本、他:高位論理記述言語B²DLの現状と将来;信学技報、VLD87-37, (1987)

[5] R. K. Brayton, et al.:The YORKTOWN Silicon Compiler; ISCAS'85, pp391-394, (1985)

[6] T. J. Kowalski, et al.:The VLSI design automation assistant:From algorithms to silicon; IEEE Design and Test of Computers, vol. 2, no. 4, pp. 33-43, (1986. 8)

[7] H. Trickey, et al.: Flamel:A High-Level Hardware Compiler; IEEE Trans. on CAD, vol. CAD-6, NO. 2, PP. 259-269, (1987)

[8] 高木、他:論理構造情報に基づくトップダウン配置手法;電子情報通信学会論文誌、Vol. J70-C, no. 1, pp. 11-20, (1987)

[9] 仲林、他:論理構造を保存したレイアウト手法;信学技報、CAS85-4, (1985)

[10] 平出、他:設計情報および回路構造に注目したゲートアレイの配置手法;情報処理学会第31回全国大会講演論文集、pp. 1621-1622, (1985)