

# 高密度マクロセルジェネレータ MOSAIC

鈴木 五郎<sup>1</sup> 山本 哲也<sup>1</sup> 夏目幸一郎<sup>1</sup> 岡村 芳雄<sup>2</sup>  
<sup>1</sup> (株)日立製作所・日立研究所 <sup>2</sup> 同デバイス開発センター

マクロセルを高密度でレイアウトするマクロセルジェネレータMOSAICを開発した。リーフセルをタイル状に隙間なく配置し、あらかじめリーフセル内に埋め込んだバス配線を使ってリーフセル間の配線を行う方式を採用している。レイアウトを強く意識した特別な回路図が必要無く、またリーフセルを汎用的に用いるために膨大な種類のリーフセルを準備する工数が不要である。試行結果では7kトランジスタ/m<sup>2</sup> (1 $\mu$ プロセス使用時)のレイアウト密度を実現している。

## *A High Packing Density Macro-cell Generator* MOSAIC

Goro Suzuki Tetsuya Yamamoto Kouichirou Natsume<sup>1</sup>  
Yoshio Okamura<sup>2</sup>

<sup>1</sup> Hitachi Research Lab. <sup>2</sup> Hitachi Device Development Center

In this paper, we propose a new high packing density macro-cell layout generator. This system places lots of leaf cells like mosaic and routes them using bus wiring patterns which are placed in a leaf cell beforehand. There are two main features. One is that it is not necessary to prepare special circuit schematics. The other is this system needs very few kinds of leaf cell. Experimental results show that 7000 transistors can be placed in 1mm $\times$ 1mm area using 1 $\mu$  process.

## 1. まえがき

高性能マイコンのデータパスのうち、整数演算部や浮動小数点演算部などは、チップサイズ及び性能を大きく支配することから大規模なセルつまりマクロセルでレイアウトすることが常識である。大規模なものは100k個程度のトランジスタから構成されるものもあり、例えば5kトランジスタ/mm<sup>2</sup> (1μプロセス使用時)の高密度なレイアウトが要求される場合がある。

マクロセルをレイアウトするDAツールは1979年ごろから盛んに研究されてきているが、処理方式の観点から、(I)ゲートアレー方式<sup>1)</sup>、(II)レイアウト形状記述言語コンパイル方式<sup>2)</sup>、(III)タイル張り方式<sup>3)</sup>の3通りに分類できる。

(I)は、拡散層の分離数が最少になるように拡散層を共有化できるトランジスタ群を求め、NMOSとPMOSトランジスタを列状に配置する。そして、トランジスタ列間にあらかじめ確保されている配線チャネルを用いて自動配線する方式である。回路情報から自動的にレイアウトすることから設計のTATは極めて短いが、トランジスタ列間に1~10本程度の配線チャネルが必要ことから、高密度なレイアウトができない(たかだか1~3kトランジスタ/mm<sup>2</sup>、1μプロセス使用時)のが現状である。又チャネル配線では、配線遅延を細かく制御するレイアウトがなかなか難しい。

(II)は雛型となる回路のレイアウト形状をある言語であらかじめ記述しておき、その都度入力されたトランジスタのサイズや個数から所望のレイアウトを生成する方式である。ただし、この方式でレイアウトするのはたかだか20~30トランジスタ規模のいわゆるリーフセルであり、このリーフセルを列状に自動(あるいは人手)で配置し、リーフセル間をチャネル配線する方式が一般的である。この方式はレイアウト形状を言語表現することから、雛型回路を用意するのに手間がかかり、又チャネル配線を併用することから(I)の方式と同様に、高密度を実現することと配線遅延の制御が難しい。

(III)はレイアウトの基本となるリーフセルを何種類か用意しておき、それらを隙間無くタイル状に配置し、リーフセル間を自動配線することによってマクロセル全体をレイアウトする方式である。この方式ではリーフセルの上空全面を配線領域として使用できるため、高密度なレイアウトが可能になる。例えば文献<sup>3)</sup>によれば、3kトランジスタのALUを約5kトランジスタ/mm<sup>2</sup> (1μプロセス使用時)でレイアウトしている。今回我々が開発したMOSAIC (Module Synthesis Aid for VLSI Chip)でも基本的にはこのタイル張り方式を採用している。

このタイル張り方式には2つの大きな問題点があった。第1の問題点は回路図の読み易さである。従来は回路図上での素子シンボルの相対位置関係つまり上下左右関係からレイアウト上でリーフセル(素子シンボルに対応)を配置する位置関係を決めていることから、素子シンボルの相対位置関係がレイアウト上のリーフセルの相対位置関係と完全に一対一対応した回路図を必要としていた。このような図面は一般的には読みにくく、回路設計の効率が劣化してしまう。MOSAICではこの問題を解決すべく、回路図の読み易さの

水準を保ちながらリーフセルを設計者の意図する通りに配置できる方式を開発した。第2の問題点は膨大な種類のリーフセルを準備することである。MOSAICでは少ない種類のリーフセルを汎用的に使用できる方式を開発し、設計のTATをできるだけ短くしている。

本論文では、まずMOSAICの概要を説明し、次に新しいタイル張り手法を述べ、最後に試行結果を示す。

## 2. MOSAICの概要

### 2. 1システム構成とレイアウトモデル

MOSAICは図1に示すように論理/回路図エディタDASH<sup>4)5)</sup>及びレイアウトエディタSPACEと<sup>6)7)</sup>一体化されており、一つのファイルインターフェースとグラフィックインターフェースを共有している。MOSAIC、DASH、SPACEは3つのタスクに分けているが、タスク間通信を利用することによってシステム間を自由自在に行き来しながら作業ができるようにしている。これらのシステムは、OSはUNIX、言語はC、及びグラフィックスはGKSと標準的なシステム環境で動く。

マクロセルの回路図を図2(a)に示す。つまりビット方向とそれと直交するビット列方向に最大でA1サイズの図面になるようページ分割した回路図をDASHで作成している。この回路図単位(以下ではサブマクロと呼ぶ)にMOSAICでレイアウトを行う。図2(b)にマクロセルのレイアウトモデルを示すが、MOSAICを用いてサブマクロ単位にレイアウトした後、SPACEを用いて人手でサブマクロのレイアウトをタイル状に隙間無く配置し、マクロセル全体のレイアウトを完成させる。また、サブマクロの中にはビット方向に同一サイズであるリーフセルがタイル状に隙間無く配置されている。リーフセル、サブマクロ、マクロセルとも外形は矩形としている。

### 2. 2読み易い回路図

MOSAICで使用する回路図の一例を図3に示す。この例では全ての素子シンボルは下位図面としてトランジスタレベルの回路を持つ階層シンボルとしている。図面の縦方向がビット方向であり、それと直交するビット列がはっきり分かる図面となっている。この図面は信号の流れに沿って書かれており、充分読み易い回路図になっている。

設計者はDASHを用いてこの回路図を表示し、どの部分をどんなリーフセルでレイアウトするかを図のように直角多角形(以下リーフセル割り当て図形と呼ぶ)で指示する作業を対話形式で行う。

MOSAICはこのリーフセル割り当て図形の上下左右関係を尊重してリーフセルを配置するが、ビット列方向に限定して次の操作を可能にしている。

- (1) 図4(I)のように、リーフセルどうしを入れ替えて配置する。
- (2) 図4(II)のように、論理図/回路図上で離ればなれな位置に存在する複数の素子シンボルあるいは配線を一つのリーフセルに対応させて配置する。

これらの機能をうまく利用することにより、充分読み易い回路図を基にして、設計者の意図する通りにリ

リーフェルを配置することができる。

### 2. 3リーフェルの汎用化

リーフェルの汎用化を図るために、あらかじめ用意するリーフェルの入出力信号の位置は固定せず、サブマクロ中にリーフェルを配置した時点で周囲の状況からそれらの位置を決定する方式とした。

MOSAICを動かす前にあらかじめSPACEを用いて人手でレイアウトされたリーフェルのレイアウトモデルを図5に示す。リーフェル間の配線はあらかじめ埋め込んでおいた縦横(図5は横方向のみ)に走る複数本のバス配線パターンを利用するが、バス配線パターンは意図的にセル枠に接しないようにしており、若干の間隙をあけている。リーフェルを隙間無く並べるだけでバス配線パターンどうしが接続されることを避けるためである。又ゲートに接続される配線(ここではゲート配線と呼ぶ)はバス配線とは接続せず、わざわざ浮いた状態にしてある。

MOSAICでは以下の手順(図6参照)でレイアウトを行う。

- (S1) リーフセルを自動配置する。  
リーフェル割り当て図形に関する図形包含矩形(割り当て図形を完全に包含する面積最小な矩形)の左下頂点座標でリーフェル配置位置を決定する。
- (S2) バス配線にサブマクロ信号名を割り当てる。
- (S3) 信号名の割り当てられたバス配線とセル枠との間のギャップにALパターンを補充。
- (S4) リーフセル割り当て図形で切り出されたサブマクロの回路とリーフェルの回路(特に信号)の対応付けを行い、ゲート配線にサブマクロの信号名(s1~s3)を割り当てる。
- (S5) 同一信号名が割り当てられたバス配線とゲート配線の交点にコンタクトを配置する。

ここで、(S2)~(S5)の処理を自動モディファイと呼んでいる。

### 3. 新しいタイル張り手法

前節で説明した自動モディファイ処理において、主に次の2つの手法を開発した。一つは限られた本数のバス配線にサブマクロの全ての信号名を余り無く割り当てることを可能にする手法であり、もう一つは妥当な時間でゲート配線に対してサブマクロの信号名を誤り無く割り当てる手法である。前者に関しては限られた本数の配線チャンネル(バス配線が存在するX/Y軸方向の直線状の連続領域。正確に言うとセル枠との間に間隙があることからバス配線は不連続であるが、隣合うリーフェルで同じ位置にバス配線がある場合は連続していると言う)を用いて信号名を割り当てる手法を開発し、後者に関しては回路のトポロジを有効に利用して、処理時間が信号数の1~2乗に比例し、設計者の頭で処理した場合と同様な成功率を保證する回路図対応付け手法を考案した。

#### 3. 1 チャンネル割り当て手法

サブマクロの外部との切り口となる信号の位置は固

定されているが、サブマクロ内部で閉じる信号はバス配線を利用して自由な位置に配線することができる。図7に配線チャンネルとそこに割り当てるサブマクロ信号の一例を示す。配線チャンネルの長さが不揃いであるため、チャンネル配線で使われるLeft Edge アルゴリズム<sup>8)</sup>だけでは対応できない。

MOSAICでは、次のアルゴリズムによってバス配線チャンネルにサブマクロ信号を割り当てている。ここでは、X軸方向の配線チャンネルにサブマクロ信号を割り当てる手順を説明するが、Y軸方向の配線チャンネルへの割り当て処理もほぼ同様の手順である。

- (S1) サブマクロ信号に注目し、割り当て可能な配線チャンネルがただ一つであるものが存在する場合には、該当チャンネルに該当信号を割り当てる。
- (S2) 逆に配線チャンネルに注目し、割り当て可能なサブマクロ信号がただ一つであるものが存在する場合には、該当チャンネルに該当サブマクロ信号を割り当てる。
- (S3) 割り当て処理ができなくなるまで(S1)(S2)を繰り返す。
- (S4) 全てのサブマクロ信号が割り当てられれば終了。
- (S5) 残りのサブマクロ信号に注目し、回路図上の左端点X座標がもっとも小さいものを選ぶ。
- (S6) IF 使用済みチャンネルに割り当て可能 THEN 該当チャンネルに該当サブマクロ信号を割り当てる。  
ELSE 最も上側に存在する未使用チャンネルに該当サブマクロ信号を割り当てる。
- (S7) 全てのサブマクロ信号が割り当てられるまで(S1)~(S6)を繰り返す。

#### 3. 2 回路図対応付け手法

次に、回路トポロジによって2つの回路、つまりリーフェル割り当て図形で切り出されたサブマクロの回路(以下回路図1と呼ぶ)とリーフェルの回路(以下回路図2と呼ぶ)を対応付けする手法を説明する。

計算機の内部では図8のように信号をノードに、トランジスタをブランチに対応させ、ノードの集合Nとブランチの集合Bからなる無向グラフ $G = (N, B)$ で回路を表現する。2つの回路の対応付けは、グラフの同形判定問題に帰着される。この問題に関しては現在まで数多くの解法が考案<sup>9)10)</sup>されてきたが、本システムではその内の一つである集合分割法を採用している。つまり、同一条件で2つのグラフのノードの集合 $N^1$ と $N^2$ を分割し、分割系列 $\Pi^1$ と $\Pi^2$ を求める

$$\Pi^1 = (N^1_1, \dots, N^1_k)$$
$$\Pi^2 = (N^2_1, \dots, N^2_k)$$

$N^1_i$ と $N^2_j$ 内にただ一つだけノードを含む場合にはノード $n^1_i$ とノード $n^2_j$ は一対一に対応付けられる。集合分割の方法としては距離分割<sup>11)</sup>が最も一般的である。これは、あらかじめ対応付けられているノードからの最短経路の距離で分割する方法であるが、回路図どりにレイアウトされているかどうかの結核チェック用に開発されたもので、対応付けできないノ

ードを精度良く指摘することを目的にしている。MOSAICで必要なのは同形かどうかの判定と同形の場合のノードの対応付けだけであることから、従来よりも簡単な方法を採用し、高速処理することができる。以下に、今回新しく考案した手法を説明する。

回路図1のグラフ表現がグラフ1であり、回路図2のグラフ表現がグラフ2である。又電源及びグラウンドだけは2つの回路で同一信号名が与えられていることを前提としている。

- (S1) 同一信号名を持つノードを対応付けする。
- (S2) 残りのノードに関して、ブランチの数、ブランチ上のトランジスタ数、対応済みの隣接ノード数を基準としてノードの集合を分割する。分割集合を構成するノードの数が一つで、分割の基準が等しい集合つまりノードを対応付けする。図8の例題では2つのグラフでノードS1とS2は既に対応済みであり、グラフ1とグラフ2のノードを上記基準で分割した結果、S3とS4及びT1とT2は共にノードの数が一つの分割集合になり、S3はT1に、S4はT2に対応付けされる。
- (S3) (S2)で対応がとれなかった残りの分割集合に関して、対応済みの隣接ノードがどのような信号名であるかを基準にして再分割する。分割集合を構成するノードの数が一つで、分割の基準が等しい分割集合つまりノードを対応付けする。図9の例題では2つのグラフでノードS1とS2は既に対応済みであり、グラフ1とグラフ2のノードを上記基準で分割した結果、S3とS4及びT1とT2は共にノードの数が一つの分割集合になり、S3はT1に、S4はT2に対応付けされる。
- (S4) (S2)で対応がとれなかった残りの分割集合に関して、ブランチの属性、つまりPMOS/NMOS、あるいはチャネル長/チャネル幅などのトランジスタの属性を基準にして再分割する。分割集合を構成するノードの数が一つで、分割の基準が等しい分割集合つまりノードを対応付けする。図10の例題では2つのグラフでノードS1とS2は既に対応済みであり、グラフ1とグラフ2のノードを上記基準で分割した結果、それぞれのグラフの3つのノードは別々の分割集合になり、S3はT1に、S4はT2に、S5はT3に対応付けされる。
- (S5) (S4)において分割集合どうしの対応付けはできるが、その構成ノード数が1でない場合は強制的に対応付けしてしまう。全く同一なトランジスタが並列に接続されている場合のように、交換可能なノードだからである。分割集合どうしの対応付けができない場合は、2つのグラフは合同でない」と判断する。

実際には、リーフセル割り当て図形で切り出されたサブマクロの回路において図11のようにトランジスタのゲートを電源につったり、グラウンドに落ちたりする場合がある。この場合、(S1)だけでは対応付けができないために、電源やグラウンドであっても(S2)以降の処理の対象としている。

信号数の異なるいくつかの回路を使い、回路図対応付けプログラムの性能を評価した結果を図12に示す。横軸が信号数であり、縦軸がCPU処理時間(約8M

IPSのEWS使用)である。又前記した各ステップ(S1)~(S4)において、全体のノードの何%が対応付けでき、どれほどの処理時間((S1)(S2)を基準にした相対比で表している)がかかったかを表1(全て平均値)に示した。(S1)と(S2)で全体の75%のノードを対応付けしている。これらの処理は信号数の2乗に比例するがコードや数に注目した高速処理にしているため、全体の処理時間の1/56を支配するにすぎない。一方(S3)では(S2)で対応がとれなかった残りの分割集合をさらに分割するが、分割の対象となる集合を構成するノード数が充分少なくなっていることから、信号数のほぼ1乗に比例する処理で済んでいる。(S4)も同様の理由で信号数のほぼ1乗に比例する処理となる。(S3)と(S4)で全体の処理時間の55/56を支配しているため、結果として回路図対応付け全体の処理時間は信号数の1~2乗に比例したものとなっている。グラフの同形判定はNP問題に属するが、電源・グラウンドは共通の名称を持ち、トランジスタは各種の属性を持つことから、これらの特長を利用して多項式のオーダーでこの問題を解くことが可能となった。又、いずれの回路においても、100%の対応付け成功率を実現している。

#### 4. 試行結果

MOSAICを用いてサブマクロを設計した例を示す。図13は代表的なリーフセルのレイアウト図である。AL1層とAL2層にそれぞれ6本ずつのバス配線が配置してあり、また各層に2本ずつの電源を配置してある。電源配線はセル枠まで到達させて配置してあり、リーフセルをタイル状に配置しただけで接続されるようになっている。図14がリーフセルを自動配置し、自動モディファイして出来上がった図3の回路図(全体で2.8kトランジスタ)に対応するレイアウト図である。縦方向がビット方向である。

約8MIPSのEWS上でMOSAICを動かした場合、リーフセル自動配置から自動モディファイまで約15分のコマンド応答時間で処理ができる。レイアウト密度は7kトランジスタ/mm<sup>2</sup>(1μプロセス使用時)である。第2.1節で述べたように、サブマクロを人手でタイル状に配置してマクロセル全体をレイアウトすることから、マクロセル全体つまり100k個程度のトランジスタから構成されるマクロセルにおいても平均的にこの程度のレイアウト密度が実現できる。

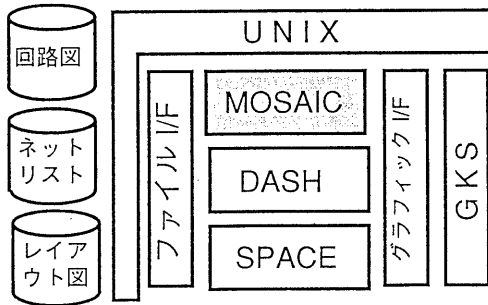
#### 5. むすび

マクロセルを高密度でレイアウトするマクロセルージェネレータMOSAICを開発した。リーフセルをタイル状に隙間なく配置し、あらかじめリーフセル内に埋め込んだバス配線を使ってリーフセル間の配線を行う方式である。レイアウトを強く意識した特別な回路図が必要無く、またリーフセルを汎用的に用いるために膨大な種類のリーフセルを準備する工数が不要である、という特長を持つ。試行結果では7kトランジスタ/mm<sup>2</sup>(1μプロセス使用時)のレイアウト密度を実現している。

謝辞 本研究の機会を与えていただいた当社平沢副所長、前島部長、細坂部長、喜田部長、大野副工場長はじめ関係諸氏に深く感謝します。

参考文献

- 1) T. Uehara et al.: Optimal Layout of CMOS Functional Arrays, Proc. of 16th Design Automation Conference, pp. 287-289 (1979, Jun.)
- 2) M. Burich: Programming language makes silicon compilation a tailored affair, Electronic Design, pp. 135-142 (1985, Dec.)
- 3) N. Matsumoto et al.: Data Path Generator Based on Gate-level Symbolic Layout, Proc. of 27th Design Automation Conference, pp. 287-289 (1990, Jun.)
- 4) 鈴木 五郎他: VLSI用階層論理設計システム(DASH)、通信学会研究会SSD84-79, pp. 43-49 (1984, Oct.)
- 5) 鈴木 五郎他: VLSI用階層論理設計システム、情報処理学会第30回全国大会, pp. 1921 (1985, Mar.)
- 6) 鈴木 五郎他: ASIC用対話型セル設計システムSPACE、情報処理学会設計自動化研究会46-7, pp. 49-55 (1987, Feb.)
- 7) G. Suzuki et al.: A Practical Online Design Rule Checking System, Proc. of 27th Design Automation Conference, pp. 246-252 (1990, Jun.)
- 8) A. Hashimoto et al.: Wire Routing Channel Assignment within Large Apertures, Proc. of 8th Design Automation Workshop, pp. 155-169 (1971, Jun.)
- 9) C. Ebeling et al.: Validating VLSI Circuit Layout by Wirelest Comparison, Proc. of ICCAD (1983, Nov.)
- 10) N. Kubo et al.: A Fast Algorithm for Testing Graph Isomorphism, Proc. of ISCAS, (1979, Jun.)
- 11) 久保 登他: 1対1対応を部分的に持つグラフ間の同形判定に関する一手法、情報処理, pp. 1167-1172 (1978, Dec.)



UNIXはATTベル研究所の登録商標

図1 システム構成  
Fig.1 System configuration

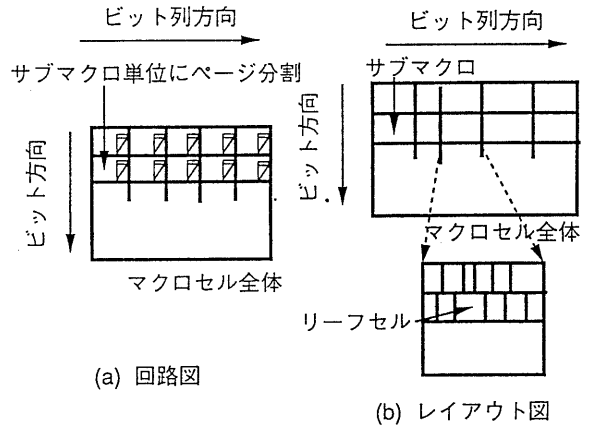


図2 回路図とレイアウトモデル  
Fig.2 Circuit and layout model

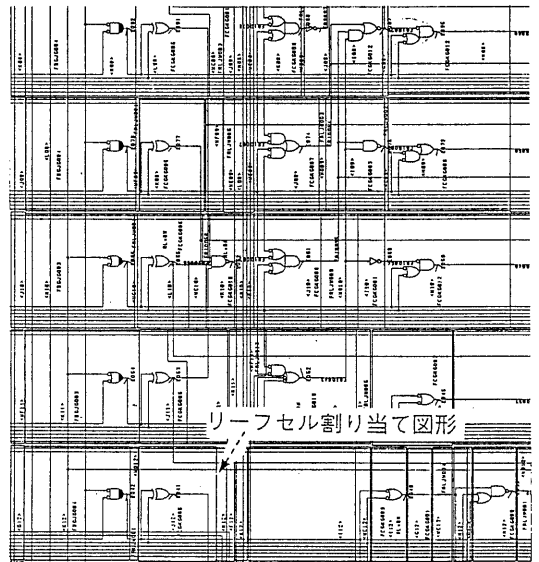


図3 回路図  
Fig.3 Circuit example

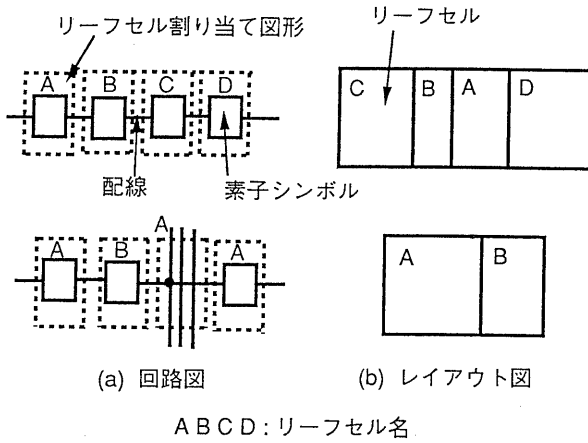


図4 回路図とレイアウトの関係  
Fig.4 Leaf cell arrangement

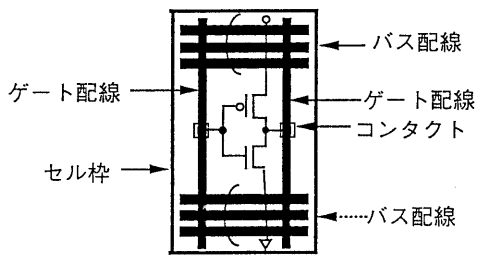


図5 リーフセルのレイアウトモデル  
Fig.5 Leaf cell layout model

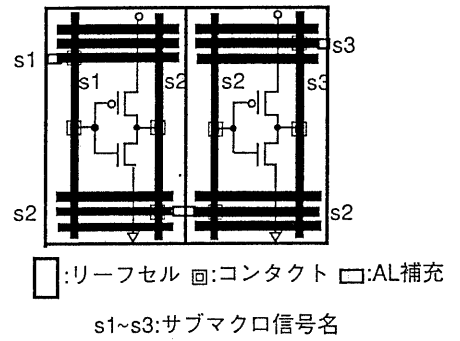


図6 リーフセルの自動モディファイ  
Fig.6 Leaf cell modification

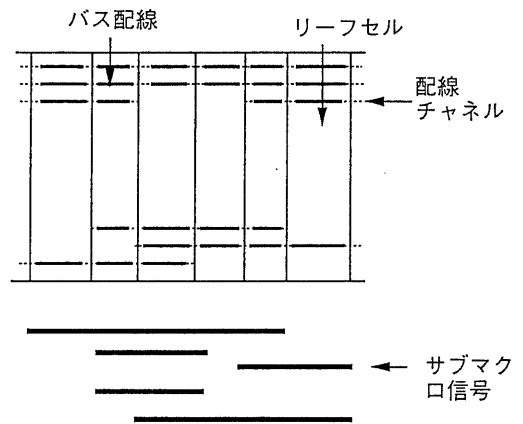
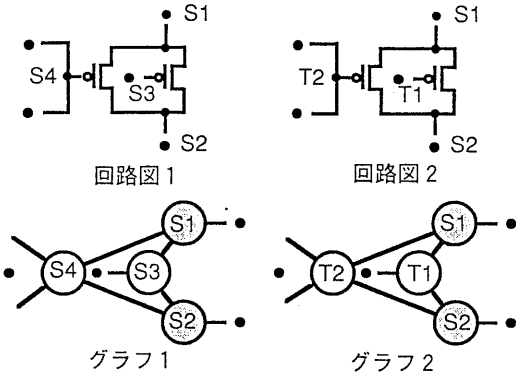


図7 配線チャンネル  
Fig.7 Channel for signal assignment

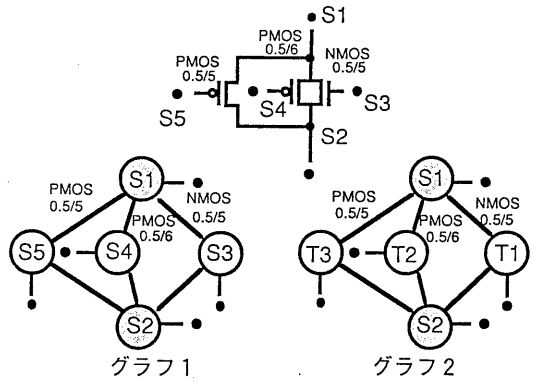
ブランチ数/対応済みノード数



$(S3)$  は  $(T1)$  に対応  
 $(S4)$  は  $(T2)$  に対応

図8 回路図対応付け手法(1)  
Fig.8 Graph isomorphism (1)

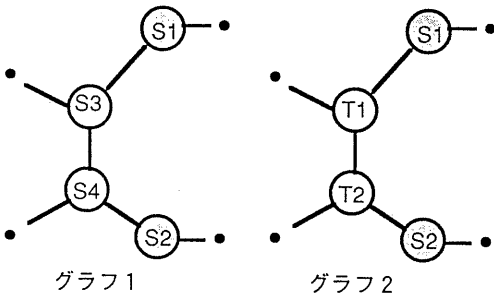
素子の種類とチャンネル長/チャンネル幅



$(S3)$  は  $(T1)$  に対応  
 $(S4)$  は  $(T2)$  に対応  
 $(S5)$  は  $(T3)$  に対応

図10 回路図対応付け手法(3)  
Fig.10 Graph isomorphism (3)

対応済み隣接ノードの信号名

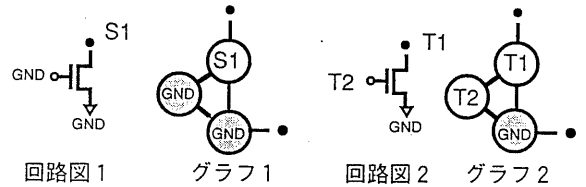


$(S3)$  は  $(T1)$  に対応  
 $(S4)$  は  $(T2)$  に対応

図9 回路図対応付け手法(2)  
Fig.9 Graph isomorphism (2)

問題点

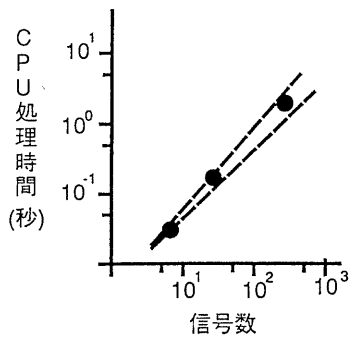
同一信号名だけでは対応付け不可能



対応策

グラフの形状で対応付け

図11 対応付けの問題点  
Fig.11 Problem of isomorphism



対応付け成功率 100%

図12 回路図対応付け性能評価  
Fig.12 Evaluation of isomorphism

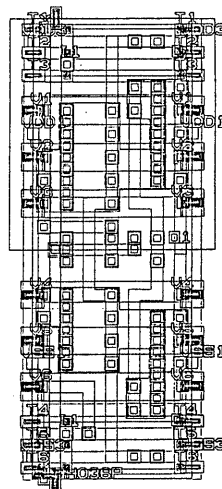


図13 リーフセルレイアウト図  
Fig.13 Leaf cell layout example

表1 対応付処理時間の内訳

Table1 Isomorphism time

	対応付けノードの割合	処理時間相対比	規模依存性
(S1) (S2)	75%	1	信号数 <sup>2</sup>
(S3)	20%	5	信号数 <sup>1</sup>
(S4)	5%	50	信号数 <sup>1</sup>

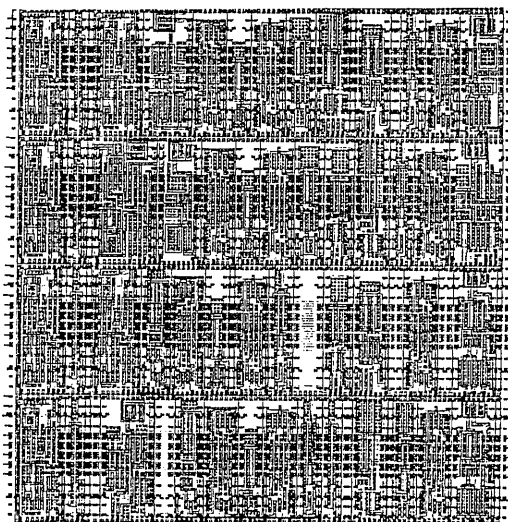


図14 サブマクロレイアウト図  
Fig.14 Submacro layout example