

高密度マクロセルジェネレータ MOSAIC†

鈴木 五郎^{††} 山本 哲也^{††}
夏目 幸一郎^{††} 岡村 芳雄^{†††}

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なく、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/mm² (1 μ m プロセス使用時) のレイアウト密度を実現している。

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、チップサイズおよび性能を大きく支配することから大規模な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/mm² (1 μ m 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。このような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¹⁾、(II) レイアウト形状記述言語コンパイル方式²⁾、(III) タイル張り方式³⁾ 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し、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ン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できることから設計の TAT は極めて短いが、トランジスタ列間に 1~10 本程度の配線チャンネルが必要なことから、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/mm², 1 μ m プロセス使用時) のが現状である。

(II) は雛型となる回路のレイアウト形状の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ン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、またチャン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、それらを隙間な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 3) によれば、3k トランジスタの ALU を約 5k トランジスタ/mm² (1 μ m プロセス使用時) でレイアウトしている。今回われわれが開発した 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。

従来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やすさである。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。そこで MOSAIC ではこの問題を解決すべく、回路図の読みやすさを保ちながらリーフセルを設計者の意図するお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。

† A High Packing Density Macro-cell Generator MOSAIC by GORO SUZUKI, TETSUYA YAMAMOTO, KOUICHIRO NATSUME (Hitachi Research Lab.) and YOSHIO OKAMURA (Hitachi Device Development Center).

†† (株)日立製作所日立研究所

††† (株)日立製作所デバイス開発センター

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^(4),5) およびレイアウトエディタ SPACE^(6),7) と一体化されており、1つのファイルインタフェースとグラフィックインタ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。OS は UNIX, 言語は C, およびグラフィックスは GKS と標準的なシステム環境でこれらのシステムを動かしている。

大規模マクロセル全体の回路図を図 2 (a) に示す。つまりビット方向 (縦方向) と、それに直交するビット列方向 (横方向) に最大で A1 サイズの図面になるように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 (以下ではこの単位をマクロセルと呼ぶ) に MOSAIC でレイアウトを行う。図 2 (b) に大規模マクロセル全体のレイアウトモデルを示すが、MOSAIC を用いてマクロセル単位にレイアウトした後、SPACE を用いて人手でマクロセルをタイル状に隙間なく配置し、大規模マクロセル全体のレイアウトを完成させる。また、マクロセル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なく配置されている。リーフセル、マクロセル、大規模マクロセルとも外形は矩形としている。

2.2 読みやすい回路図

MOSAIC で使用する回路図の一例を図 3 に示す。この例ではすべ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わ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

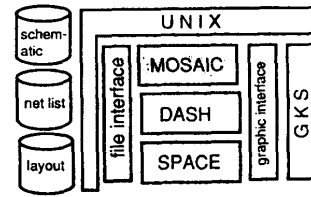
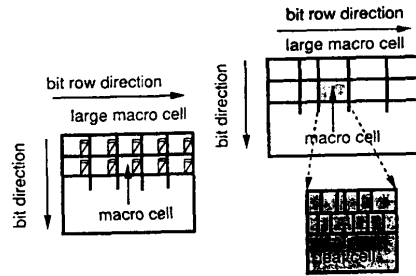


図 1 システム構成
Fig. 1 System configuration.



(a) 回路図 (b) レイアウト図
(a) Schematic. (b) Layout.

図 2 大規模マクロセルの回路図とレイアウトモデル
Fig. 2 Large macro cell schematic and layout model.

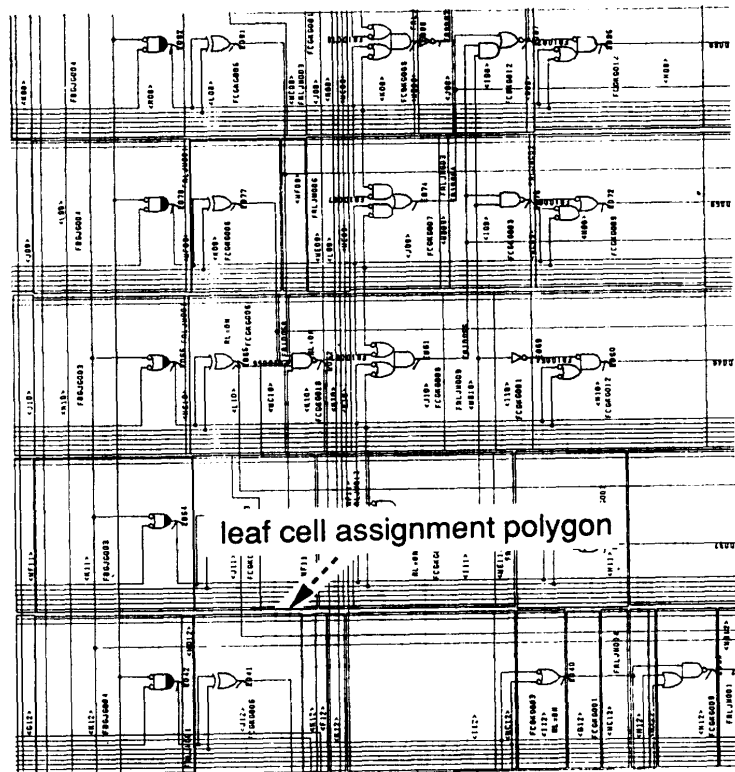


図 3 回路図の例
Fig. 3 Schematic example.

かれており、十分読みやす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 は原則として、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 (A) のように、素子どうしを入れ替えてレイアウトする。
- (2) 図 4 (B) のように、論理図/回路図上で離れ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 1 つのリーフセルに対応させて配置する。

リーフセル自動配置は次の手順に従う。リーフセ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少な矩形）の回路図上での左下頂点座標でリーフセルを配置する順序を決定し、各リーフセルの横/縦幅からレイアウト上での配置位置を決定している。ただし、図 4 (A), (B) のようにリーフセル割り当て図形にグループ名（該当ビット列内でユニークな名称）が与えられている場合、共通のグループ名が与えられた部分を同一リーフセルに割り当て、リーフセル名が与えられているリーフセル割り当て図形でリーフセルを配置する順序を決定している。図 5 に、リーフセル割り当て図形が書き込まれた回路図とリーフセルを配置した例題を示す。上記した (1), (2) の機能をサポートしたことから、信号の流

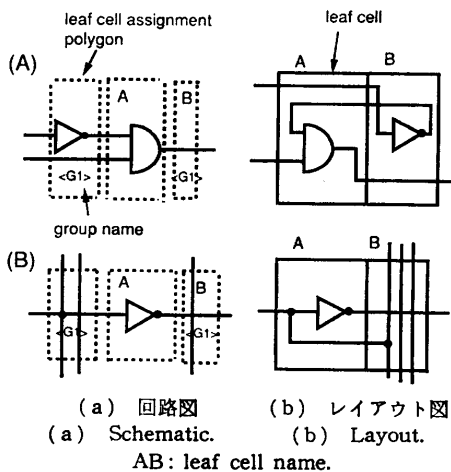


図 4 リーフセル配置時の工夫
Fig. 4 Leaf cell arrangement.

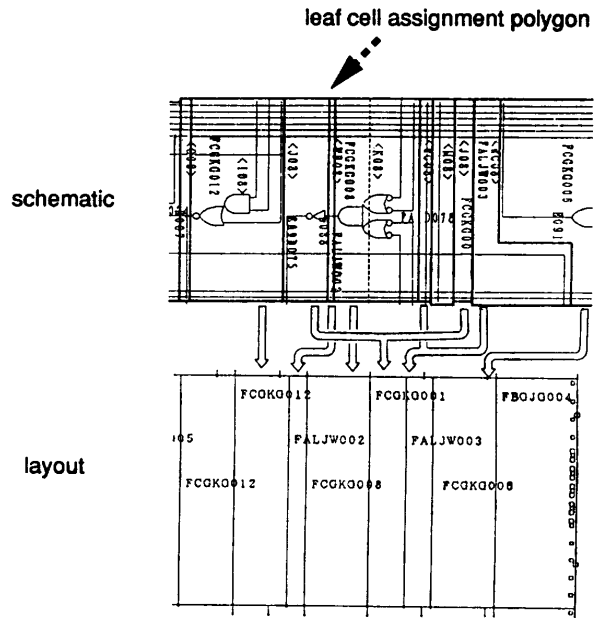


図 5 回路図とレイアウト図
Fig. 5 Schematic and layout.

れに沿って書かれた回路図を基にして設計者の意図するとおりにリーフセルを配置することが可能となった。

2.3 リーフセルの汎用化

リーフセルの汎用化を図るために、あらかじめ用意するリーフセルの入出力信号の位置は固定せず、マクロセル中にリーフセ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にした。

MOSAIC を動かす前に、SPACE を用いて人手でレイアウトするリーフセルのレイアウトモデルを図 6 に示す。リーフセル間の配線はあらかじめ埋め込んでおいた縦横（図 6 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。リーフセルを隙間な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。またリーフセルの入出力になる配線（ここでは入出力配線パターンと呼ぶ縦方向の配線）はバス配線

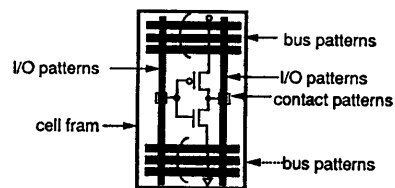


図 6 リーフセルのレイアウトモデル
Fig. 6 Leaf cell layout model.

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 7 参照) でレイアウトを行う。

- (S1) リーフセルを自動配置する。
- (S2) バス配線にマクロセルのすべての信号を割り当てる。図 7 では、インバータの入出力信号 $n1$, $n2$ をリーフセル A の 2 本バス配線に、また $n2$, $n3$ をリーフセル B の 2 本のバス配線にそれぞれ割り当てている。
- (S3) 信号が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 層パターンを補充。
- (S4) リーフセル割り当て図形で切り出されたマクロセルの回路とリーフセルの回路の対応付けを行い、入出力配線にマクロセルのすべての信号を割り当てる。図 7 では、上段の回路図と下段に示したレイアウトと等価な回路図の 2 つの回路 (上段の回路図はインバータで表現されているが、レベルを合わせるためにトランジスタ回路に変換する) を比較することによって、リーフセル A の左側の入出力配線は $n1$, また右側の入出力配線は $n2$ に対応することがわかる。同様にリーフセル B の入出力配線に $n2$ と $n3$ を割り当てている。
- (S5) 同一信号名が割り当てられたバス配線と入出力配線の交点にコンタクトを配置する。図 7 では、合計 4 個のコンタクトを配置している。

ここで、(S2)~(S5)の処理を自動モディファイと呼ぶ。図 8 にリーフセルレイアウト図の例を示す。横方向に AL1 層/AL3 層各 6 本ずつのバス配線と縦方向に AL2 層 2 本の入出力配線が配置してあり、バス配線の下は、無駄なくすべてトランジスタ領域になっている。

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。1 つは限られた本数のバス配線にマクロセルのすべての信号を余りなく割り当てることができる手法であり、もう 1 つは妥当な時間で入出力配線に対してマクロセルの信号を誤りなく割り当てることができる手法である。前者に関しては限られた本数の配線トラック (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

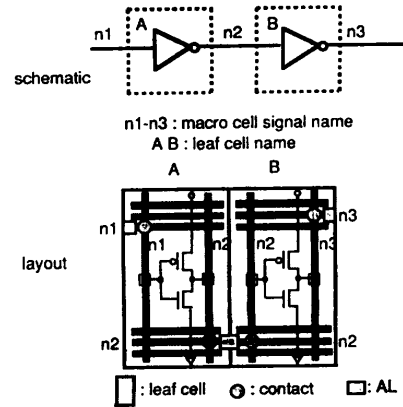


図 7 リーフセルの自動モディファイ
Fig. 7 Leaf cell modification.

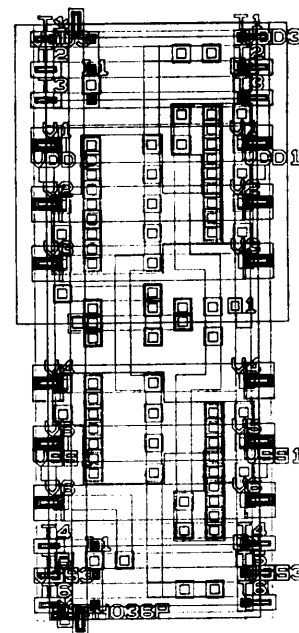


図 8 リーフセルレイアウト図の例
Fig. 8 Leaf cell layout example.

うリーフセ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) を用いて信号を割り当てるチャンネル割り当て手法を開発し、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 乗に比例した合同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

マクロセルの切り口となる信号端子の位置は、隣接するマクロセルの同一信号端子と同じ位置になるようにレイアウトを行う前に人手であらかじめ合わせているが、マクロセル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 9 に配線

トラックとそこに割り当てるマクロセル信号の一例を示す。配線トラックの長さが不揃いであるため、いわゆるチャンネル配線の幹線割り当てで使われる Left Edge アルゴリズム⁹⁾だけでは対応できない。図 10 にその例題を示す。(a)の場合、一番下の信号が割り当て可能なトラックは一番上のトラックただ 1 つであり、(b)の場合はこれとは逆に、一番下のトラックに割り当て可能な信号は一番上の信号ただ 1 つである。いずれの場合も Left Edge アルゴリズムだけで信号を割り当てると、最後の信号が割り当て不可能になる。そこで、このような事態を回避するため、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マクロセル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マクロセル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) マクロセル信号に注目し、割り当て可能な配線トラック（使用済みの場合も含む）がただ 1 つである信号が存在する場合には、該当配線トラック（使用済みの場合も含む）に該当信号を割り当てる。
- (S2) 逆に配線トラック（使用済みの場合も含む）に注目し、割り当て可能なマクロセル信号がただ 1 つである配線トラック（使用済みの場合も含む）が存在する場合には、該当トラックに該当マクロセル信号を割り当てる。
- (S3) 割り当て処理ができなくなるまで(S1)(S2)

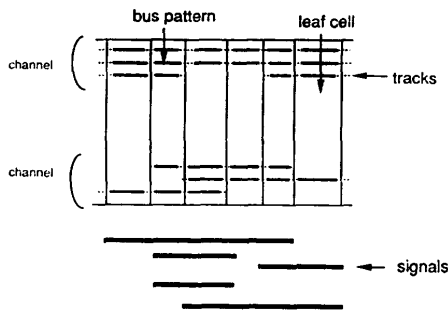


図 9 配線チャンネル

Fig. 9 Channel for signal assignment.

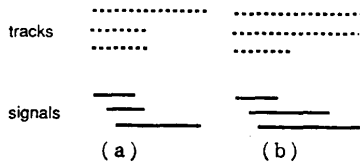


図 10 信号割り当てアルゴリズム

Fig. 10 Signal assignment algorithm.

を繰り返す。

- (S4) すべてのマクロセル信号が割り当てられれば終了。
- (S5) 残りのマクロセル信号に注目し、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 使用済み配線トラックに割り当て可能
THEN 該当配線トラックに該当マクロセル信号を割り当てる。
ELSE 最も上側に存在する未使用配線トラックに該当マクロセル信号を割り当てる。
- (S7) すべてのマクロセル信号が割り当てられるまで(S1)~(S6)を繰り返す。

3.2 合同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、つまりリーフセル割り当て図形で切り出されたマクロセルの回路（以下回路図 1 と呼ぶ）とリーフセルの回路（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 12 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たグラフで回路を表現する。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^{9),10)}されてきたが、本システムではその内の 1 つである集合分割法を採用している。つまり、図 11 のように 2 つのグラフのノードの集合 N と M を同一条件で分割する。

$$N = \bigcup_i N_i; \quad M = \bigcup_i M_i;$$

N_i と M_i がただ 1 つだけノードを含む場合にはノード $n (\in N_i)$ とノード $m (\in M_i)$ を一対一に対応付けする。対応付けできなかった集合に関しては、別な条件でさらに再分割して同様な処理を繰り返す方法である。集合分割の方法としては距離分割法¹¹⁾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、回路図どおりにレイアウトされているかどうかの結線チェック用に開発されたもので、対応付けできな

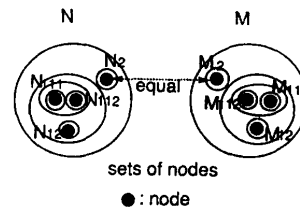


図 11 集合分割法による同形判定

Fig. 11 Nodes sets partition method.

いノードを精度良く指摘することを目的としている。MOSAIC が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。また電源およびグランドだけは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。

$$N = \cup_i N_i \quad M = \cup_i M_i$$

($N=N_i, M=M_i$ の場合もある.)

分割集合 N_i, M_i を構成するノードの数が1つで、分割の基準が等しい分割集合のノード $n \in N_i$ とノード $m \in M_i$ を対応付けする。

対応する分割集合が存在しなかったり、存在しても要素数が異なる場合には同形でないと判定する。

図12の例題では2つのグラフでノード $n1$ と $n2$ は既に対応済みとする。グラフ1とグラフ2のノードの集合を上記基準で分割した結果、各分割集合はただ1つのノード $n3$ と $n4$ および $m1$ と $m2$ を含むことになり、 $n3$ を $m1$ に、 $n4$ を $m2$ に対応付けできる。

- (S3) (S2) でノードの対応はとれなかったが分割集合としての対応がとれている N_i, M_i 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。

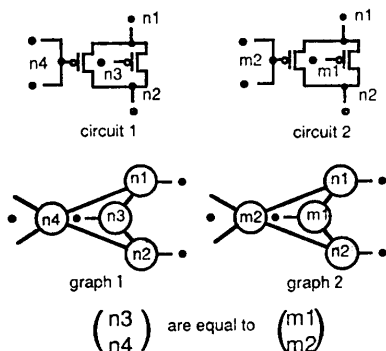


図12 同形判定アルゴリズム(1)
Fig. 12 Graph isomorphism (1).

$$N_i = \cup_j N_{ij} \quad M_i = \cup_j M_{ij}$$

($N_i=N_{ij}, M_i=M_{ij}$ の場合もある.)

再分割集合 N_{ij}, M_{ij} を構成するノードの数が1つで、分割の基準が等しい再分割集合のノード $n \in N_{ij}$ とノード $m \in M_{ij}$ を対応付けする。

対応する再分割集合が存在しなかったり、存在しても要素数が異なる場合には同形でないと判定する。

図13の例題ではノード $n1$ と $n2$ は既に対応済みとする。(S2) まででノードの対応がとれなかったグラフ1とグラフ2の分割集合 $N_i = \{n3, n4\}, M_i = \{m1, m2\}$ を上記基準で再分割すると、各再分割集合はただ1つのノード $n3$ と $n4$ および $m1$ と $m2$ を含むことになり、 $n3$ を $m1$ に、 $n4$ を $m2$ に対応付けできる。

- (S4) (S3) でノードの対応はとれなかったが再分割集合としての対応がとれている N_{ij}, M_{ij} 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々分割する。

$$N_{ij} = \cup_k N_{ijk} \quad M_{ij} = \cup_k M_{ijk}$$

($N_{ij}=N_{ijk}, M_{ij}=M_{ijk}$ の場合もある.)

再々分割集合 N_{ijk}, M_{ijk} を構成するノードの数が1つで、分割の基準が等しい再々分割集合のノード $n \in N_{ijk}$ とノード $m \in M_{ijk}$ を対応付けする。

対応する再々分割集合が存在しなかったり、存在しても要素数が異なる場合には同形でないと判定する。

また、対応する再々分割集合 N_{ijk} と M_{ijk}

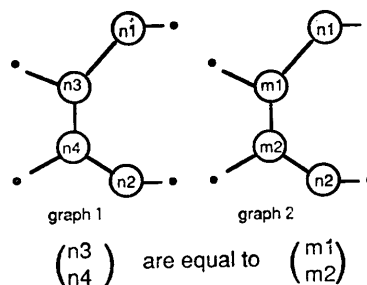


図13 同形判定アルゴリズム(2)
Fig. 13 Graph isomorphism (2).

を構成するノード数が1でない場合は強制的にそれらのノードを対応付けする. 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.

図 14 の例題では 2 つのグラフでノード $n1$ と $n2$ は既に対応済みとする. (S3) まででノードの対応がとれなかったグラフ 1 とグラフ 2 の再分割集合 $N_{i,j} = \{n3, n4, n5\}$, $M_{i,j} = \{m1, m2, m3\}$ を上記基準で再々分割すると, 各再々分割集合はただ 1 つのノードを含むことになり, $n3$ を $m1$ に, $n4$ を $m2$ に, $n5$ を $m3$ に対応付けできる.

実際には, リーフセル割り当て図形で切り出されたマクロセルの回路において図 15 のようにトランジスタのゲートを電源につったり, グランドに落とし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 16 に示す. 横軸が信号数であり, 縦軸が CPU 処理時間 (約 8 MIPS の EWS 使用) である. また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) がかったかを表 1 (すべて平均値) 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, コード化された信号名に注目し, またブランチ数など数に注目した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十分少なくなっていることから, 本来信号数の 2 乗に比例する処理であるが, 実質的に信号数のほぼ 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. また, 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 を用いてマクロセルを設計した例を示す. 図 17 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 (縦方向がビット方

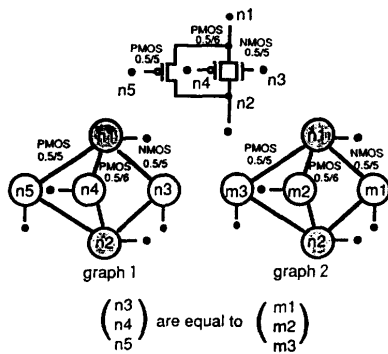


図 14 同形判定アルゴリズム (3)
Fig. 14 Graph isomorphism (3).

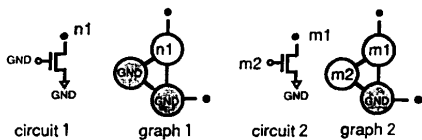


図 15 同形判定の問題点
Fig. 15 Problem of isomorphism.

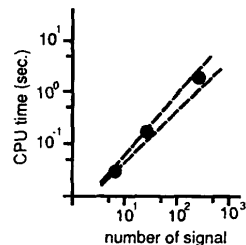


図 16 同形判定の処理速度
Fig. 16 Evaluation of the algorithm.

表 1 同形判定処理時間の内訳
Table 1 Isomorphism processing time analysis.

	How many nodes can correspond	Relative computing time	Computational complexity
(S1) (S2)	75%	1	NS*
(S3)	20%	5	NS*
(S4)	5%	50	NS*

NS: number of signal.

向)である。本例題では、図8に示したものを合わせて6種類のリーフセルを使用している。すべてのリーフセルを専用で作成した場合は約600種類作成することになり、リーフセルを作成する工数は1/100に低減したことになる。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/mm²(1 μ mプロセス使用時)である。第2.1節で述べたように、マクロセルを手でタイル状に配置して大規模マクロセル全体をレイアウトすることから、大規模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大規模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。

リーフセルを汎用化したことによる、エリア上のオーバヘッドに関して考察する。第2.3節で述べたように、リーフセル全面に配置された横方向バス配線の下はトランジスタ領域であることから、縦方向には無駄なエリアは存在していない。一方横方向でみると、隣にどんなリーフセルが配置されているか予想できないために、一番外側の縦配線はセル枠から設計規則の1/2だけ内側に配置している。そこで、該当リーフセルの隣接部分にデットエリア(マクロセル全体に関して最悪5%程度)が生じる可能性がある。大規模マクロセルはその外形を矩形としていることから、その条件を満足するように二次元コンパクションでデットエリアを取り除くとエリアの節約になる。ただし、実際にはほとんどのリーフセルが縦配線を一番外側まで一杯に配置しており、このようなデットエリアはほとんど生じていない。つまり、われわれの経験ではリーフセルを汎用化したことによるエリア上のオーバヘッドはほとんど生じていない。

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

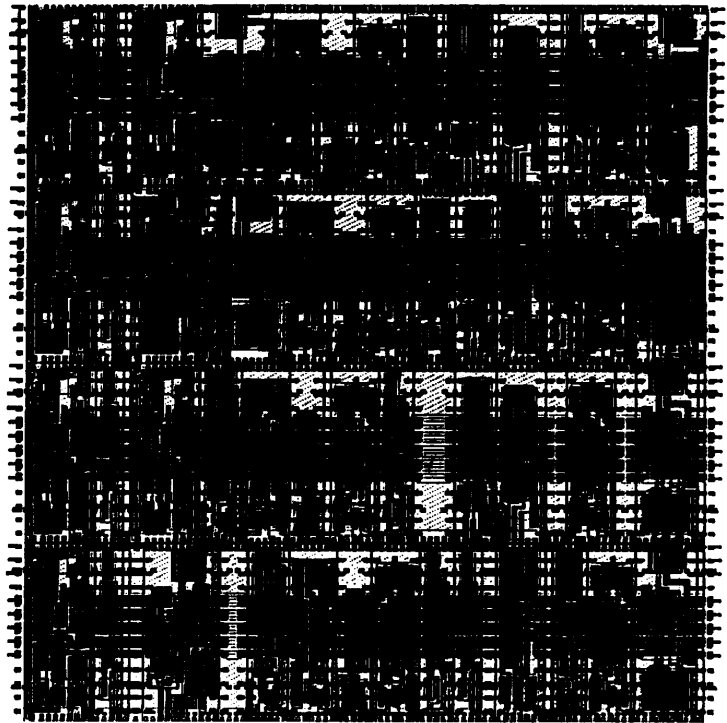


図17 マクロセルレイアウト図の例
Fig. 17 Macro cell layout example.

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なく、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/mm²(1 μ m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主管技師長、前島部長、佐藤部長、細坂部長、喜田副センター長、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) Uehara, T. et al.: Optimal Layout of CMOS Functional Arrays, *Proc. of 16th Design Automation Conference*, pp. 287-289 (1979).
- 2) Burich, M.: Programming Language Makes Silicon Compilation a Tailored Affair, *Electronic Design*, Vol. 33, No. 29, pp. 135-142 (1985).
- 3) Matsumoto, N. et al.: Datapath Generator Based on Gate-level Symbolic Layout, *Proc. of 27th Design Automation Conference*, pp. 388-393 (1990).
- 4) 鈴木五郎ほか: VLSI 用階層論理設計システム

- (DASH), 電子通信学会研究会, SSD 84-79, pp. 43-49 (1984).
- 5) 鈴木五郎ほか: VLSI 用階層論理設計システム, 第30回情報処理学会全国大会論文集, pp. 1921-1922 (1985).
 - 6) 鈴木五郎ほか: 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).
 - 7) Suzuki, G. et al.: A Practical Online Design Rule Checking System, *Proc. of 27th Design Automation Conference*, pp. 246-252 (1990).
 - 8) Hashimoto, A. et al.: Wire Routing Channel Assignment within Large Apertures, *Proc. of 8th Design Automation Workshop*, pp. 155-169 (1971).
 - 9) Ebeling, C. et al.: Validating VLSI Circuit Layout by Wirelist Comparison, *Proc. of ICCAD*, pp. 172-173 (1983).
 - 10) Kubo, N. et al.: A Fast Algorithm for Testing Graph Isomorphism, *Proc. of ISCAS*, pp. 641-644 (1979).
 - 11) 久保 登ほか: 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*, Vol. 19, No. 12, pp. 1167-1172 (1978).

(平成3年6月20日受付)

(平成3年11月5日採録)



鈴木 五郎 (正会員)

昭和50年慶応義塾大学理工学部電気工学科卒業。同年(株)日立製作所入社。以来 VLSI・DA/CAD システムの研究開発に従事。計算幾何学に興味を持ち、レイアウト・コンパクタ, 設計規則検証, 回路接続検証, 図面エディタ, モジュール・ジェネレータを開発。現在同社日立研究所第8部主任研究員。IEEE, 電子情報通信学会各会員。



山本 哲也 (正会員)

昭和54年千葉工業大学電子工学科卒業。同年(株)日立製作所入社。以来, CAD 端末システム, VLSI・DA/CAD システムの研究開発に従事。マスクパターン・エディタ, モジュール・ジェネレータを開発。現在同社日立研究所第8部研究員。



夏目幸一郎 (正会員)

平成元年筑波大学第一学群自然科学類卒業。同年(株)日立製作所入社。以来 VLSI・DA/CAD システムの研究に従事し, モジュール・ジェネレータを開発。現在同社日立研究所第8部所属。



岡村 芳雄

昭和50年東京工業大学工学部電子工学科卒業。同52年, 同学科修士修了。同年(株)日立製作所入社。以来 LSI マスクパターンレイアウト, 検証用 DA/CAD システムの開発に従事。マスクパターンからの回路抽出と特性検証, マスクパターンデータベース・マスクパターンエディタを開発。現在同社デバイス開発センター DA 開発部主任技師。