設 計 自 動 化 53－4
（１９９０．７．１０）

チップ周辺領域の配線手法

雑賀俊二[1)]　　豊永昌彦[2)]
1) 松下電器産業（株）半導体研究センター
2) 松下電子工業（株）ＤＡ開発研究所

ＬＳＩレイアウトの開発コスト、開発期間削減のためにコムやパッケージの共通化を目指す標準チップ方式に適した、チップ周辺配線手法について述べる。本周辺配線手法では、まず、周辺パッド四辺と、コアモジュール間の配線混雑度と配線最短化を力学モデル化し、配置を決定する方法と、線分探索法と迷路配線法の相互の利点を組み合わせたＭＯＬＥ配線法による高速配線手法からできている。ＭＯＬＥ配線法におけるグリッドに関して、チップ周辺配線を意識した配線グリッドのメモリー節約法も提案する。テストチップにおける実験の結果、巨大なチップ周辺配線を高速に処理できることがわかった。

A chiplevel peripheral routing method

Shunji Saika[1)], and Masahiko Toyonaga[2)]
1) Semiconductor Research Center Matsushita Electric Industrial Co.,Ltd.
2) Design Automation Dev.Lab. Matsushita Electronics Corporation

We present a chiplevel peripheral routing method on the standard chip including the core-module placement, where the standard chip is to reduce the development cost and the turn-around-time. The process has two steps, one is to assign the core module by dynamic model calculation, and the second is to route between pins around core and I/O pads by using MOLE(Modified Lee Expansion) routing method, that is based on line-search method and maze routing method. Consideing the gigantic obstacle core, we set only peripheral area around the core to compress the size of wiring area memory. As the experimental result for the gigantic test chip data,we could observe the effectiveness of our method.

１．はじめに

近年、集積回路の製造技術の向上によるデザインルールの微細化が進み、大規模な回路を１チップ上に作成することが可能となってきた。１チップ上で実現するシステムの巨大化に伴い、ＬＳＩレイアウト設計及び周辺部の開発に要する期間、コストの削減が問題となってきている。とくにＡＳＩＣ（特定ユーザー向けＬＳＩ）などにおいては、その傾向が強い。我が社では、この問題を解決する方法として、標準チップ方式を用いてきた。

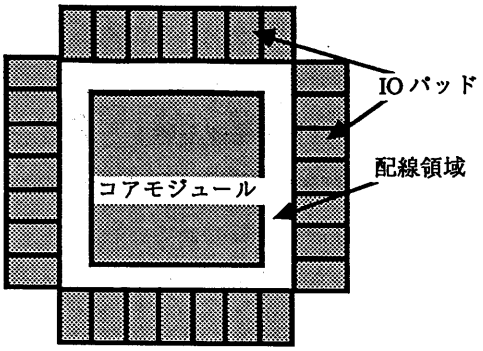


図１　　標準チップ

標準チップ方式とは、図１に示すように、周辺のパッド部をライブラリー化し、そのコム、パッケージを共有化するものである。チップ内部の論理を実現したコアモジュールをレイアウト完了後、この標準チップ内に配置し、周辺配線を行なうことにより、チップが完成する。

従来のチップ周辺配線では、コアモジュールとパッド間を、４つのチャネル領域に分割して、各々をチャネル配線して完成させる方法であった。この方法に拠れば、与えられたコアモジュールに対して最小のパッド枠を得ることができる。

一方、標準チップ方式では、パッドの位置があらかじめ限定されているところにコアモジュールを配置して配線するため、巨大なスイッチボックス問題となる。

我々は、本報告において、コアモジュールの二次元的配置法と、チップ周辺配線を意識したメモリー節約表現法を用いたＭＯＬＥ配線法（Modified Lee Expansion Method）によるチップ周辺配線手法について述べる。

本手法は、周辺パッド四辺とコアモジュール間の配線混雑度の考慮と配線最短化を、力学モデルによる斥力と引力でモデル化し、コアモジュールを二次元的に配置する方法と、コア部を考慮してメモリーを削減するメモリー表現法、そして線分探索法と迷路配線法の相互の利点を組み合わせたＭＯＬＥ配線法による配線の完成方法、から構成されている。

以下、まず第２章では、ＭＯＬＥ配線法のアルゴリズムについて述べる。つづいて第３章では、チップコアモジュールの２次元的配置手法について説明する。第４章では、メモリーを削減したグリッドモデルの表現法について説明する。第５章では、本手法によるチップ周辺配線の実験結果を示し、第６章で考察する。第７章に、報告のまとめ及び今後の展開について述べる。

２．迷路配線手法

レイアウト設計の配線処理では、従来から多くのアルゴリズムが研究されているが、その中でも迷路配線法[1]と線分探索法[2]は最も普及している。線分探索法とは配線領域を線分で分割し、それら線分について交点から交点へと探索して行くことにより、配線経路を求めていくものである。本手法では単純な経路については、非常に高速に求めることができる利点を持つ一方、経路が複雑なった場合には、十分な探索ができなくなることが指摘される。また、本手法による配線結果で、最短経路は保証されない。

迷路配線法は、配線領域を配線間のデザインルールを満たす最小の格子（グリッド）に分割して、その格子を探索することにより配線経路を求める方法である。処理が単純で、経路が存在すれば必ず最短経路を発見することができ、しかも最短距離を保証する利点を持つ。しかし、全配線領域に格子を設定するため、処理に要するメモリー空間及び、検索に必要になる処理時間は、膨大となる。両者の利点をうまく合わせた方法として、高速迷路配線法が、提案されている[3]。しかし、この方法においても、配線経路の発見が難しいときには、迷路配線法が作用する仕組みになっており、とくにチップ周辺配線のように領域が膨大になった場合には、処理時間が問題となる。そこで我々は、

やはり、線分探索法と、迷路配線法の利点を用いた高速な配線法、ＭＯＬＥ配線法開発した。

なお、以下では配線領域は水平及び垂直な線分で囲まれた領域（複合長方形領域）とする。また、配線は2層配線とし、求める配線経路は斜線を含まないとする。

２．１　ＭＯＬＥ配線法

ＭＯＬＥ配線法とは、線分探索法の高速性と、迷路配線法の柔軟性を取り入れた配線手法である。まず、配線領域におけるグローバル情報（障害物の位置、及び形状、さらに始点、終点の位置）を、配線禁止領域の各頂点から、及び始点、終点から、水平・垂直に引かれた補助線を用いて表現し、その補助線上で迷路法による探索を行なうものである。

２．１．１　補助線について

ここで言う補助線とは、配線禁止領域の各辺に平行に隣接した直線のグリッド列の集合を指す。この補助線グリッドに、配線通過可能フラグを立てておき、迷路探索は、このフラグの立つグリッドのみを対象とする。既存の配線もまた一種の配線禁止領域と見なし、その各線分に沿って補助線を発生する。配線対象となる端子については、別に補助線を設ける。

２．１．２　アルゴリズム

以下に、アルゴリズムを示す。ここでは簡単のために、１グリッド幅の経路を縦横ルールに従って作成するものとする。

補助線処理

ステップ1　グリッドの設定。
各グリッドにＤ、Ｃを定義する。

ステップ2　ピン情報、障害物情報の読み込み。
Ｄに、配線禁止、配線可能、既配線コードを入れる。

ステップ3　障害物補助線の設定。
Ｄに、配線禁止領域に沿って補助線コードを設定する。水平・垂直の補助線が交わるグリッドについては、そのＣに交点コードを入れる。

ステップ4　端子補助線の設定。
終点Ｔを通る水平および垂直補助線を発生させ、さらに始点Ｓからも水平および垂直の補助線を発生させる。

配線経路探索処理

ステップ1
探索起点リストＯＱ、探索点リストＮＱを空にする。

ステップ2
始点ＳをＯＱの最後尾に登録する。その際、始点ＳのＤに始点のピン方向をいれる。（ただし、方向コードには、" 1"（ＵＰ）、"2"（ＲＩＧＨＴ）、"3"（ＤＯＷＮ）、"4"（ＬＥＦＴ）の４つを用いる。）

ステップ3
ＯＱの先頭のグリッドＧＯについて、ＤをDO,CをＣＯとし、ＤＯ方向に隣接するグリッドを探索グリッドＧＮ１、もしも、ＣＯが、交点コードであれば、ＤＯ方向の左右に隣接するグリッドをＧＮ２、ＧＮ３とする。ＯＱからＧを抹消する。
ただし、ＯＱが空であれば、ステップ７に　行く。

ステップ4
ＧＮ１について、またＧＮ２、ＧＮ３があれば、それらについても、ＧＮ１から順にそれぞれ以下を行なう。ＧＮｉ（ｉ＝１～３）のＤをＤＮとして、
１）ＤＮ＝補助線コードならば、ＧＮｉのＤにＤＯを入れ、ＧＮｉをＮＱの最後尾に登録する。
２）ＤＮ＝ＤＯに垂直な既配線であれば、ＧＮｉのＤにＤＯを入れ、ＧＮｉをＮＱの最後尾に登録する。
３）ＤＮ＝Ｄコードに平行な既配線又は、配線禁止領域であれば、なにもしない。
４）隣接グリッドＧＮｉ＝終点であれば、ＧＮｉのＤにＤＯを入れ、ステップ６に進む。

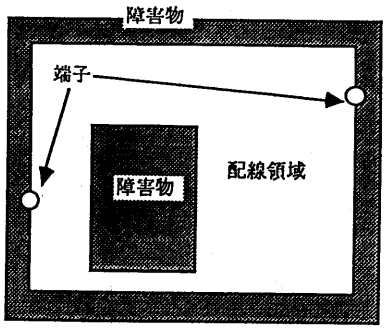
ステップ5　ＯＱの更新
ＮＱを探索起点キューＯＱにコピーする。その後、ＮＱを空にする。

ステップ6　配線の作成
終点Ｔから出発して、各グリッドのDに書き込まれている方向コードに従って探索のときと逆に辿り始点から終点までの経路を得て、終了。

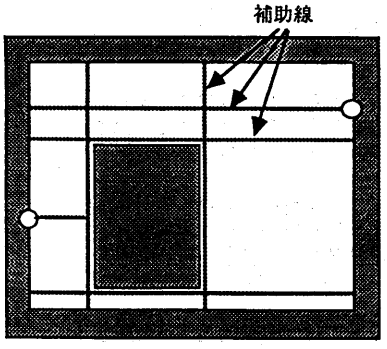
ステップ7　未配線処理
未配線のメッセージを出力し、終了。

以上がＭＯＬＥ配線法のアルゴリズムである。配線処理の様子を図２（ａ）～（ｄ）に示す。本手法による経路探索が一次元的に行なわれている様子がよくわかる。本手法に於ては、補助線にたいして、Ｌｅｅ迷路法を用いていることから、ＭＯＬＥ配線法で求められる経路もまた最短経路であることが保証される。また、ステップ４でも説明した通り直進する向きの探索を優先するため、得られる経路の形状は、折れ曲がりの少ないものとなる。

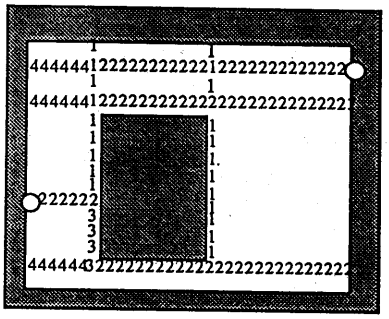
多端子ネットの場合には、求められた経路をつぎつぎに経路探索対象としていくことにより配線できる。この場合、次の配線処理への準備としては、その経路上のグリッド全てにわたって終点コ



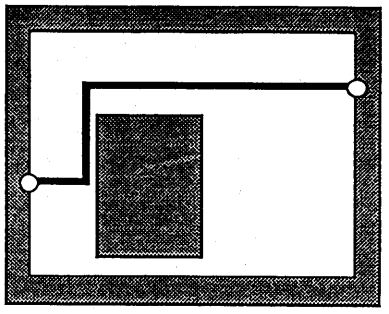
(a) 障害物と端子の読み込み



(b) 補助線の作成



(c) 経路の探索



(d) 配線結果

図2　MOLE 配線法

ードを書き込むだけでよい。一方、他のネットの配線結果は、次の配線処理にとって探索の障害物となる。ＭＯＬＥ配線法で必要な処理としては、既存の補助線に加え、新たに作成された配線を障害物として囲んだ補助線を書き加えるのみでよい。

我々は、本ＭＯＬＥ配線法を採用することにより、処理速度の問題を解決し、チップ周辺のコアモジュールとＩ／Ｏパッド間の配線処理を実用時間内に行なえると考える。

３．チップコア部の位置決定

本手法のチップ周辺配線では、前提として、配線を行なう前にチップコアモジュールの位置が決定していなければならない。しかし、コアモジュールの位置決定を行なうこと自体が非常に難しい問題である。この点、例えばモートルータ[4]においては、最初、全辺同一の幅を持つチャネル領域をチップコアモジュールの周囲に設定し、配線後、不必要なトラックの削減を行なうことにより、配置の最終位置を決定する方法をとっている。我々の目指すチップ周辺配線（図３（ａ）参照）では、Ｉ／Ｏパッドの大きさ・形状が初期段階において決定されているため、変形することができない。そこで我々は、以下に述べる仮定の下で、コアモジュールの二次元的な配置決定を行なうための力学モデルを提案する。これは、コアモジュールを配置する際に考慮するべき要素を、パッドが囲む領域内でコアモジュールに与えられる力学的ポテンシャルとしてモデル化して、その力学的安定点を求めようとするものである。本手法の目的は、配線密度を考慮して、より短い配線で周辺配線が完結するように、コアモジュールの位置を、縦横同時に考慮して求めることである。

３．１　計算のための前提

仮定０

Ｉ／Ｏパッドが囲む領域とコアモジュールは、共に形状固定の長方形で、相対する辺が平行になるように、あらかじめ与えられた向きで配置するとする。

仮定１　各辺に面する配線領域について、その辺の配線密度分の幅を確保するため、パッドとコアモジュールの相対辺の間に、ピン幅の和に比例した斥力ポテンシャルＵｒを想定する。

$$Ur = r \cdot M / R$$

ここで、Mは、パッド側、コア側それぞれの辺の総ピン幅の平均であり、Rは、パッドとコアモジュールの相対辺の間の距離である。

仮定2　各辺に面する配線領域について、総配線長を短くするため、パッドとコアモジュールの相対辺の間に、ピン幅の和に比例した引力ポテンシャルUaが存在すると想定する。

$$Ua = a \cdot M \cdot L^2$$

ここで、Lは、パッド側とコア側それぞれのピンの重み中心（幅を重みとしている）間の、辺に平行な方向の距離成分である。

上記で仮定した斥力及び引力のポテンシャルを用いて、パッドとコアモジュール間に働く全ての力を考慮したポテンシャルは、次で表される。

$$U = \sum_{\text{全辺}} (Ua + Ur)$$

いま、パッド枠の左下の頂点を原点にとり、コアモジュールの左下の頂点の座標をx、yとすると（図3（b））、ポテンシャルUは独立変数x、yの関数となる。コアモジュールに働く力は、x方向に働く力をFx、y方向に働く力をFyとすると、それぞれ以下の様に与えられる。

$$Fx = - \partial U(x,y) / \partial x$$
$$Fy = - \partial U(x,y) / \partial y$$

我々の目的は、力学的安定点を求めることであるから、力のつり合いの方程式

$$Fx(x) = 0 , \quad Fy(y) = 0$$

を解けばよい。その際に、次のようにスケール規格化変換をしておくことにより、チップサイズの大小によらずに、全体に対する比率としてコアモジュールの位置を求められる。

$$x \text{ -----> } x/W , \quad y \text{ -----> } y/H$$

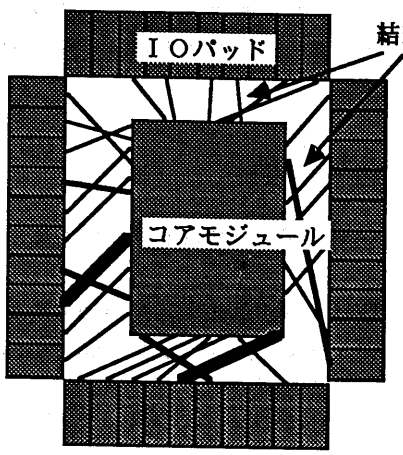
ここで、Wは、I／Oセルのつくる配線領域の横幅からコアモジュールの横幅を引いたものであり、Hは、高さについての同様な量である。上記変数変換に対して、力の比例定数は、

x方向）$A = 2aW$、$R = r / W^2$

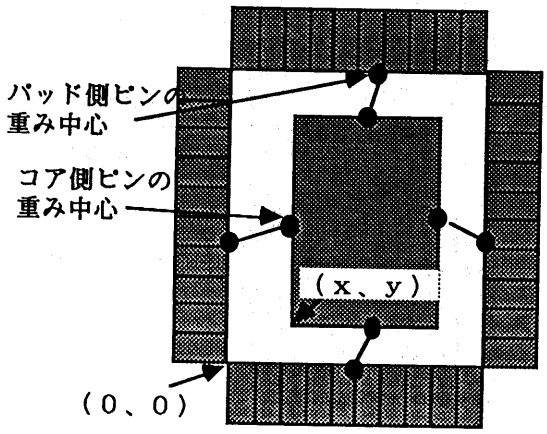
y方向）$A = 2aH$、$R = r / H^2$

となる。

なお、具体的に解x、y
を求める方法としては、ニュートン法を用いている。



（a）実際の問題



（b）本モデル

図3　コアモジュールの二次元配置モデル

4．メモリーを削減したグリッドモデル

グリッドとして取り扱うには、余りにも巨大となるチップ周辺の配線をＭＯＬＥ配線法でおこなう場合であったとしても、使用メモリーが膨大になるという問題は、避けられない。しかしながら、チップ周辺の場合の配線領域は中央部に巨大なコアモジュールという障害物が存在する。従って、チップ全体をグリッド表現した場合には、実際配線領域として使用される部分はその内のごく一部にしか過ぎない。我々は、巨大な配線障害物でであるチップコアモジュールの存在を意識し、その領域に対してグリッドメモリーを用意しないことで、使用メモリーを大幅に削減する方法をとる。

以下に、メモリーを削減するグリッドモデルの表現方法を述べる。

4．1　メモリ表現

対象とするチップ周辺配線問題に対して、次の仮定を置く。

1）Ｉ／Ｏパッドが形成する配線領域の外枠、及びコアモジュールが形成する配線領域の内枠は、共に形状固定の長方形であるとする。

（図４．１）に示す。

2）配線領域は、斜線で示された輪状領域である。また、グリッド座標ｘｙを外枠の左下を原点として設定する。

このような仮定の下で、二次元配列の一次元配列変換を行なう。ｘｙ座標で指定された各グリッドと、一次元配列の要素との対応関係は、関数により求める。

一次元配列は、もとの配線領域をｙ軸に平行な線分に分割し、座標原点グリッドから指定されたグリッドまで、これらの線分を下から上に繰り返し順次たどっていきそのグリッド距離によりｘ、ｙ座標を表わす。二次元情報をより速いアクセス時間で、読みだし書き込みを行なうために、中間にセグメント管理メモリを設定している。指定された座標からｘ方向で管理されるセグメント管理メモリを呼び出し、ｙ情報を用いて、一次元配列の位置を割りだすため、非常に高速に処理できる。図4.2に、各メモリのポインタ関係を示す。

5．計算機実験

5．1　データ

本チップ周辺配線の効果を評価するために、表１に示すようなチップデータを用いいて計算機実験を行なった。

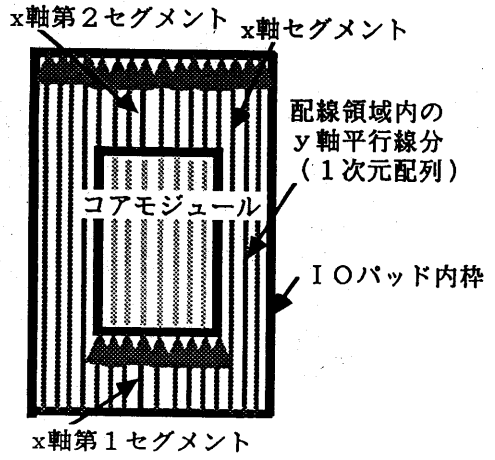


図4.1　二次元配列と１次元配列（１）

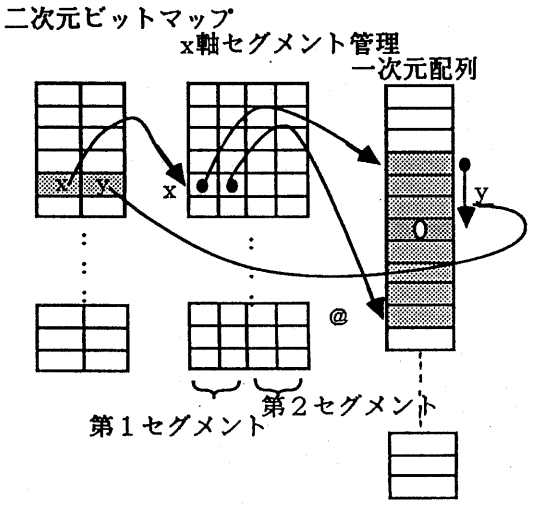


図4.2　二次元配列と１次元配列の対応（２）

表1　評価データ一覧

コアモジュールサイズ：1650 x 1650
内部配線領域サイズ　：2000 x 2000
配線幅：10～40　配線スペース幅：10

データ名	外部ピン数	IOセル数	ネット数
CHIP1	68	20 x 20	63
CHIP2	56	20 x 20	51
CHIP3	40	20 x 20	38

注意
CHIP1 の各辺ピン数　L=17、T=17、R=17、B=17
CHIP2 の各辺ピン数　L=5 、T=17、R=17、B=17
CHIP2 の各辺ピン数　L=5 、T=17、R=17、B=5

5.2　実験条件

斥力、引力のの重みパラメータは、A = R とした。また、各辺に、電源に相当するピンを設け、他の配線の4倍幅に設定した。

5.3　実験結果

各チップデータに対する実験結果を図5（a）～（c）に示す。また、コアモジュールの位置、及び処理時間について表2に示す。本処理は、ソルボーン社製ワークステーションシリーズ4／600で行なわれた。

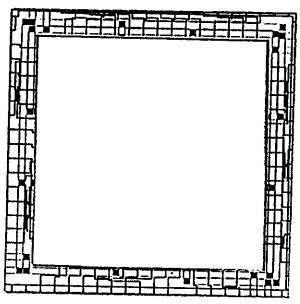
表2　実験結果

データ名	コアモジュール座標(比)	処理時間(sec)
CHIP1	0.505 , 0.505	0.01 + 15.74
CHIP2	0.394 , 0.512	0.01 + 13.50
CHIP3	o.343 , o.399	0.02 + 10.43

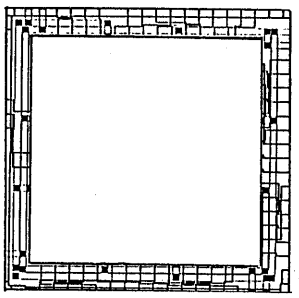
6．考察

従来チップのコアモジュールの配置は、経験的に、中央部に置き、配線後の配線密度から人手により移動配線を繰り返していた。これは、配線領域に閉める配線の混雑をあらかじめ予想することが難しいからであった。本報告で用いた配線混雑度のモデルは、その精度の点から見て十分とはいえない。しかし、このモデルの意義は、二次元的に四辺の混雑度を意識して配置することにより、水平垂直各々を独立に評価する以上に有効な位置を求めることができる点にある。ここで用いたMOLE配線法は、補助線の生成の後のメイズ展開を用いているために、最短経路を保証し、また、配線領域が巨大になっても、その検索領域は、小さいために、高速性を保ったまま処理が可能と考えられる。（補助線上のみを探索するので、nグリッド離れた2点間の経路探索問題をO（n）で処理することができる。）

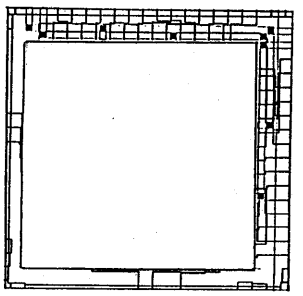
また、本報告では、説明していないが、配線幅を可変にして処理ができる改良も、なされている。



（a）CHIP 1の実行結果



（b）CHIP2の実行結果



（c）CHIP3の実行結果

図5　実験結果

コアモジュールのに次元的配置において、斥力と引力の距離依存性を変えた理由は、単調な解を避けるためである。また、そのモデルの妥当性は、今後の課題である。

一次元メモリー表現による節約効果は、例えば、１格子点あたり２バイトを用意するとして、全領域の大きさを４０００ｘ４０００グリッドとすると、３２Mバイトであるのが、コアモジュールを除く本手法では、約１０Mバイト程度であり、２／３の節約効果がある。この大きさは、現在のワークステーションにおいて十分用意されているメモリーサイズである。

７．むすび

標準チップ方式に適した、チップ周辺配線手法について述べた。本周辺配線手法では、まず、周辺パッド四辺とコアモジュール間の、配線混雑度と配線最短化を力学モデル化し、配置を決定する方法と、線分探索法と迷路配線法の相互の利点を組み合わせたＭＯＬＥ配線法による高速配線手法からできている。ＭＯＬＥ配線法におけるグリッドに関して、チップ周辺配線を意識した配線グリッドのメモリー節約法も提案した。テストチップにおける実験の結果、巨大なチップ周辺配線を高速に処理できることがわかった。

８．謝辞

著者らは、本研究を進めるのに当たりご助言ご協力をいただいた、秋濃、福井、川上、佐藤、奥出以下次世代レイアウトシステムＭＡＰＬＥＳ開発メンバーの各位と、本研究の機会を与えていただいた、松下電子ＤＡ開発研究所の泉所長、千村室長、および半導体研究センタの堀内取締役、間野所長に感謝いたします。

［参考文献］

[1]C.Y. Lee: "An algorithm for path connections and its applications" ,IRE Trans. on Electronic Computers VEC-10, 1961.

[2] D.W.Hightower : "A solution to the line routing problem on a continuous plane" in Proc. 6th Design Automation Workshop, June (1969).

[3] J.Soukup : "Fast Maze Router",in Proc. 15-th Design Automation Conference,pp100 - 102 June (1978).

[4]Richard K. Mcgehee: "A Prctical Moat Router", in Proc. 24th Design Automation Conference,1987 .