グリッドフリーチャネル配線の一手法
An Algorithm for Grid-free Channel Routing

米田 浩，小嶋 格，林 正樹，久保 登，千葉 徹
犬伏 恒雄　　（シャープ株式会社）

1．まえがき

近年の半導体加工技術の進歩は目覚しく、LSIチップ上に搭載される素子の数も飛躍的に増加している。この様なLSIの高集積化に伴ない、チップ内の高密度配置・配線を実現する自動レイアウトシステムの必要性が増大している。LSI自動レイアウトシステムの最も重要な目的の一つはチップの面積を最小化することにあり、従って配線に対してはその配線領域を最小にする必要がある。配線手法としては、通常配線経路をチャネルに割当てる段階（グローバル配線）と各チャネル内での配線経路を詳細に決定する段階（チャネル配線）に分けて取り扱うのが効果的な方法の一つとして良く知られている。

このうち、後者のチャネル配線手法に対しては、チップ全面にわたり予め固定したグリッド（格子）上のみ配線を許すという条件の下で、チャネル幅を最小にする為の種々の理論的考察及び実験が行なわれてきた。[1][2] しかし、グリッドを固定する配線手法は処理時間も速く、プログラムが容易である反面、一般にコンタクト－コンタクト間クリアランスがコンタクト－ライン間クリアランスよりも大きいことよりグリッドの間隔をどちらに取っても配線領域の面積が増大してしまう。

この様なグリッドを固定する場合の弊害をなくし、より密度の高い配線を行なうことを目的として提案されたのがグリッドフリー（格子無し）配線手法であり、この手法の有用性を示す実験結果も最近報告されている。[3]

本文においては、グリッドフリーの配線を行う際に考慮されなければならない様々な制約条件を重み付制約グラフ G_W を用いて表現し、このグラフを用いて、ライン・コンタクト等の間隔をすべて最小許容範囲にして、高密度な配線を実現するアルゴリズムを提案する。

2．準備

2.1 チャネル配線問題

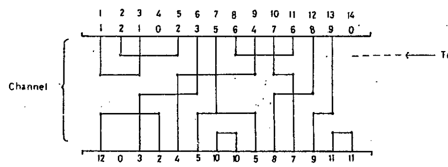


図1．チャネルと配線要求例

チャネル配線問題は、図1に示す様に2行の端子列上に与えられた配線要求を、それらにはさまれた領域（チャネル）において、2つの配線層を用いて実現する問題である。ここで、配線要求は上下の端子列に対応した2つの整数の列で示され、同じ正整数が与えられた端子の集合（ネット）は同電位になる様に結線されねばならない。ただし 0を持つ端子は他のどの端子とも結線する必要のない空端子であるものとする。

2層を用いて配線を行う際は、通常一つの層には端子列に平行な水平線分（幹線）が、他の層には垂直な線分（支線）が置かれ、それらはコンタクトによって接続される。又、支線は各端子の存在する垂直な列の上に、幹線は水平トラックと呼ばれる水平線上に置かれ、与えられた水平トラック数（チャネル容量）内で配線要求を実現することがこの問題の目的となる。

またネットを2端子対単位に分解したものをサブネットと呼び、サブネットに対応する水平線分を2端子幹線と呼ぶ。

2.2 前提条件

本手法においては、文献[3]と同様に次の前提条件を用いる。

(1).配線層は縦横2層を用い、その方向も固定である。

(2).各種クリアランス(設計基準)が以下に示すように与えられているものとする。ここで、クリアランスとは、コンタクト及びラインを配置する際に互いに離さなければならない最小の距離(整数)を示すものであり、すべて中心線又は中心点間の距離で表わされる。(図2参照)

1. 水平コンタクトーコンタクト間クリアランス(HCC)
2. 垂直コンタクトーコンタクト間クリアランス(VCC)
3. 水平コンタクトーライン間クリアランス(HCL)
4. 垂直コンタクトーライン間クリアランス(VCL)

(3).各クリアランスの最大公約数をトラックのピッチとし(2)のクリアランスはトラックのピッチの倍数で表わす。

(4).ここでは問題を単純化する為に、チャネルの上下辺には凸凹が存在しないものとする。

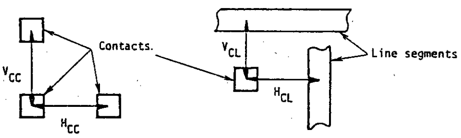


図2. 各種クリアランス

2.3 幹線間の制約条件[3]

従来の配線手法における制約条件には、異なるネットの間の短絡を避ける為に、幹線が互いに重ならないこと、支線が重ならないことの2種類がある。ここで、支線が重ならないとは、上辺の端子が持つ値と下辺の端子が持つ値が異なり、かつこれら2つの端子が同一の端子列上に置かれている時、上辺の端子に接続されるネットの幹線は、下辺のそれよりも上に置かなければならないということになる。

グリッドフリーチャネル配線においては以上の制約に加え、グリッドを固定した場合には自動的に満足されていたクリアランス条件についても考慮する必要がある。以下に、考慮すべき幹線間の制約条件について述べる。

(1).上下制約

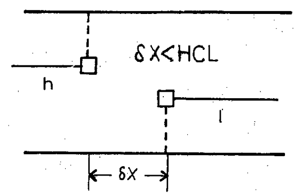
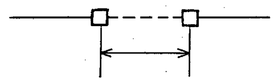


図3. 上下制約

図3に示す様に、チャネルの上下辺に異なる値を持ったネットが接続されており端子間隔δxが水平コンタクトーライン間クリアランス(HCL)より小さい場合、幹線h、l間には上下制約が生じる。

(2).非重複制約



δX ≧ Hcc

図4. 非重複制約

異なる値を持つネットの幹線を同一のトラックに置く場合、図4に示す様に端子の間隔δxは水平コンタクトーコンタクト間クリアランス(HCC)以上離れていなければならない。

(3).クリアランス条件

幹線を割当てる場合には、従来の配線手法とは異なり、2.2で示したクリアランスをすでに割当てられた幹線との間で満足させる必要がある。

3. 処理手順

本手法においては、2で述べた種々の制約条件を含んだ制約グラフである重み付制約グラフ $G_W = [V_W, E_W \cup A_W]$ を導入し、この G_W を用いて配線を行なう。ここでは、一般性を失うことなく、2端子幹線に対応するグラフのみを対象とすることとする。

良く知られている様に、上下制約グラフ中にループが存在している場合、ドグレグ配線が必要となるため、本手法では、前処理として上下制約ループの解除を行なっている。

本手法は大きく分けて、前処理・重み付制約グラフ作成・幹線割当の3段階から成り、以下に詳述する。ここで問題を簡単に表現する為チャネルは水平であるとし、図5に示す様な座標系を取るものとする。図中のN_iはサブネットを表わし、$v_i \in V_w$はグラフ上でサブネットN_iに対応する節点、$Y(v_i)$はサブネットN_iが割当てられるトラックのY座標である。

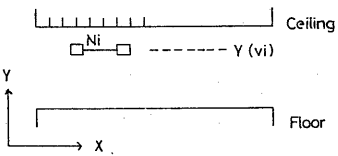


図.5 ネットN_iとトラック位置$Y(v_i)$

3.1 前処理

前処理においてはネットをサブネットに分割する処理と、上下制約ループの解除を行なう処理が行なわれる。
サイクル範囲とは上下制約グラフ内でループを構成するサブネットが、チャネル内で占める範囲であるとすると、ループ解除は次の3つの方法で行なわれる。[4]

(1). 方法1.：ループを構成しているサブネットをそのサイクル範囲中で分割する（図6）

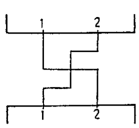


図6. 方法1

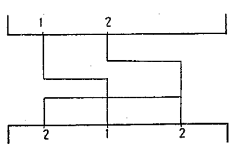


図7. 方法2

(2). 方法2.：ループを構成しているサブネットを、そのサイクル範囲外の同一の値を持つ端子まで延長する。（図7）.

(3). 方法3.：ループを構成しているサブネットをそのサイクル範囲外の適当な端子の位置まで延長する（図8）

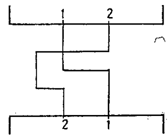


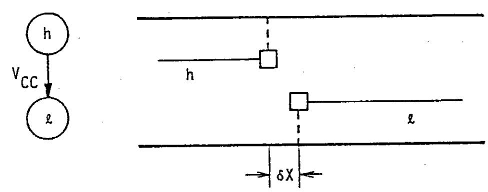
図8. 方法3.

前処理のアルゴリズムでは、上記の方法1,2,3,を順次適用して、上下制約ループを解除する。

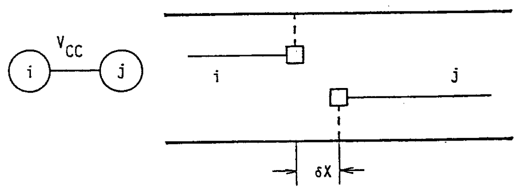
3.2 重み付制約グラフ作成

重み付制約グラフ $G_w = [V_w, A_w \cup E_w]$ は次の様に定義される。

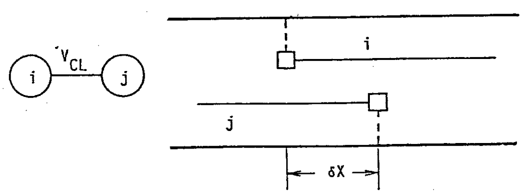
(1). サブネットN_hとN_lの端点間の距離δxがHCL未満の場合、節点v_hとv_lの間には重さ$w(a) = w(v_h, v_l) =$ VCCを持つ有向枝$a = \langle v_h, v_l \rangle \in A_w$を付加する。（図9(a)）



(a) $\delta X < H_{CL}$



(b) $H_{CL} \le \delta X < H_{CC}$



(c) $H_{CC} \le \delta X$

図9. 重み付制約グラフ中の有向枝及び無向枝

(2). サブネット N_i と N_j の端点間の距離 δx が HCL 以上 HCC 未満の場合．節点 v_i, v_j の間には．重み $w(e) = w(v_i, v_j) = VCC$ を持つ無向枝 $e = (v_i, v_j) \in E_w$ を付加する（図9(b)）

(3). サブネット N_i と N_j の端点間の距離 δx が HCC 以上で幹線が重複している場合．節点 v_i, v_j の間には重み $w(e) = w(v_i, v_j) = VCL$ を持つ無向枝 $e = (v_i, v_j) \in E_w$ を付加する。（図9(c)）

なお例外として．同一の値を持つサブネット対に対しては上記の処理は行なわず．枝は付加しない。（図10）

図10. 枝が付加されない場合

以上の定義により．図11の配線要求に対する重み付制約グラフの例を図12に示す

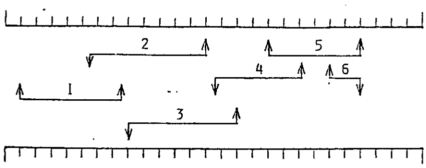


図11. 配線要求

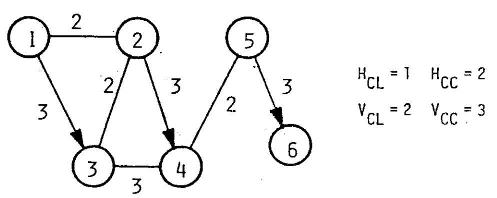


図12. 重み付制約グラフ(例)

〈重み付制約グラフ作成〉

STEP.1：サブネットの集合を N_s とし N_s をサブネットの左端の座標でソーティングする。

STEP.2：N_s 中のすべてのサブネット N_i に対して．STEP.3，STEP.4 の処理を繰返す。

STEP.3：サブネット N_i の左端の x 座標と右端の x 座標 + HCC の間に．左端が存在するサブネット N_j を選出し．それらの集合 N_c を作る。

STEP.4：サブネット N_i と $N_j \in N_c$ に対し前記の(1)(2)(3)の定義に従い．有向枝．無向枝を付加する。

以上の処理により作成したグラフ G_w を用いて．幹線割当を行なうわけであるが．ここでグラフ G_w を用いたグリッドフリーチャネル配線について考えてみる。

まずチャネル幅Wは．以前に定義した $Y(v_i)$ を用いて

$$W \triangleq \underset{i}{MAX}(Y(v_i)) - \underset{i}{MIN}(Y(v_i))$$

と表わされる。グリッド固定のチャネル配線においては．このWを最小化することを目的としているわけであるが．グリッドフリーチャネル配線に対しては．さらに条件が加わる。それを考慮して定式化を行なうと．次の様に表わされる。

［グリッドフリーチャネル配線問題］

「重み付制約グラフ G_w が与えられた場合．以下の2つの条件を満足し．チャネル幅Wを最小とする配線を行なえ

(1). すべての $a_{hl} \in A_w$ に対し

$$Y(v_h) - Y(v_l) \geq w(a_{hl})$$

(2). すべての $e_{ij} \in E_w$ に対し

$$|Y(v_i) - Y(v_j)| \geq w(e_{ij})$$ 」

3.3 幹線割当

以下では3.2で定式化したグリッドフリーチャネル配線問題の発見的なアルゴリズムを示す。本アルゴリズムは．重み付制約グラフによって表わされた配線要求に対して，チャネルの上方より順次ネットをクリアランスを満足しながらトラックに割当てるものである。

同一トラックに割当てられる幹線（グラフ上では節点）の集合を決定する際．唯一つの手法で可配線集合を選出するのではなく．3種の候補者節点の集合 W_j ($j=1,2,3$) を作成し．評価関数

$F(W_j)$が最大となるものを可配線集合として決定している。

評価の対象となる候補者節点の集合をW_j $(j=1,2,3)$とすると評価関数$F(W_j)$は次式で表わされる。

$$F(W_j) = C_\infty * H(W_j) + \sum_i E(v_i),\ C_\infty \gg 1.$$

$$H(W_j) = \underset{i}{MAX}(U(v_i)) + \underset{i}{MAX}(D(v_i)) - \underset{i}{MAX}(U(v_i)+D(v_i))$$

$$v_i \in W_j \ (j=1.2.3)$$

ここで

$E(v_i)$：節点v_iが持つ無向枝の数。

$U(v_i)$：節点v_iを含む最長パス上で有向枝が入ってこない節点（ソース）からv_iまでの有向枝の数。

$D(v_i)$：節点v_iから有向枝が出て行かない節点（シンク）までの有向枝の数（図13）

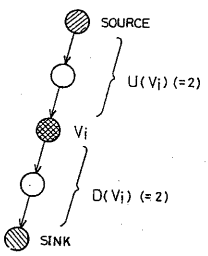


図13. $U(v_i)$と$D(v_i)$

$H(V_i)$はW_jに含まれる節点を同一のトラックに割当てた場合の最長パス長の増加分であり（図14）[1] $\sum_i E(v_i)$はW_j中の節点が持つ無向枝の総数である。

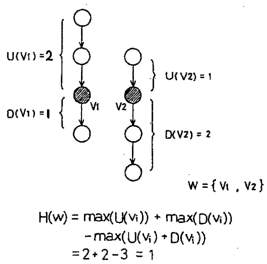


図14. $H(W)$

この評価関数$F(W_j)$が最大となる候補者集合を可配線集合とする。この様な集合は上方のトラックに割当てられずに、後のすなわち$Y(v_i)$の小さいトラックに割当てられた場合、最長パスの増加分が大きくなると予測される。従って、上方のトラックに割当てられる場合と比べ、チャネル幅Wが増大する可能性が高くなると考えられる。この様に、割当を後回しにすると、それ以後の割当に悪影響を及ぼすであろうと考えられる集合を決定する為に、上記の評価関数を用いているといえる。

〈幹線割当〉

STEP.1：集合S ← {グラフG_W中で有向枝が入ってこない節点}

STEP.2：S中のすべての節点v_iに対してサブネットN_iが割当てられる最上のトラックのy座標$Y(v_i)$を算出する。

STEP.3：$Y(v_i)$の最大値Y_{MAX} $(= \underset{i}{MAX}(Y(v_i)))$を求め、$Y(v_i) = Y_{MAX}$となる$v_i$をS中から選出し、部分集合Uを作る。
U中のすべての節点の間に無向枝が存在しなければ、可配線集合$Z \leftarrow U$としてSTEP 6.へ.

STEP.4：U中から可配線集合Zとなり得る候補者として、次の3種の集合W_1, W_2, W_3を選出する。

W_1：レフトエッジ法を用いて選出した候補者集合.

W_2：ライトエッジ法を用いて選出した候補者集合

W_3：最長パス上の節点を中心に、右端からはレフトエッジ法を、左端からはライトエッジ法を用いて選出した候補者集合

STEP.5：候補者集合W_j $(j=1,2,3)$に対し、評価関数$F(W_j)$を適用し値が最大となる集合を可配線集合Zとする。

STEP.6：Z中の節点v_iに対応するサブネットN_iを$Y(v_i)(=Y_{MAX})$のy座標を持つトラックに割当てる。
$V_W \leftarrow V_W - Z$

STEP.7：V_Wが空になれば終る。空でない場合、V_Wに従ってグラフG_Wを更新し、G_W中で有向枝が入ってこない節点の集合Sを選び、STEP.2へ

4. 実験結果

本手法を、FORTRAN を用いてプログラムし、VAX11/780 上でいくつかの実験を行なった。

4.1 グリッドフリーとグリッド固定の比較

表1にグリッドフリー、とグリッッド固定の場合の実験結果を示す。グリッドフリーの場合のクリアランスは HCC = VCC = 8、HCL = VCL = 7、固定の場合はすべて8である。data1～4 は実際のデータの中のチャネルである。

	data 1	data 2	data 3	data 4
グリッドフリー	85	96	95	91
固定	96	104	104	112

表1. グリッドフリー、グリッド固定に場合のチャネル幅　単位はピッチ

コンタクト－コンタクト間クリアランスとコンタクト－ライン間のクリアランスの差が少ないこの様な場合においても チャネル幅Wはグリッドフリーで配線すると 8～18%減少する。この差が大きい場合は、さらにチャネル幅は減少すると考えられる。

4.2 評価関数の有効性

本手法においては、候補者集合 W_j ($j=1,2,3$) 中から可配線集合 Z を決定する際、評価関数を用いず、Z を W_1 あるいは W_2, W_3 のどれに対しても固定することが可能である。例えば Z を W_1 に固定するとレフトエッジのグリッドフリー配線が行なわれることになる。4.1のデータに対して、Z を W_1, W_2, W_3 に固定した場合に、得られた実験結果を表2に示す

	data 1	data 2	data 3	data 4
W_1に固定	93	99	99	91
W_2に固定	86	101	106	92
W_3に固定	92	100	102	98

表2. Z を W_1, W_2, W_3 に固定した場合のグリッドフリー配線結果(チャネル幅)、単位はピッチ

ここで、クリアランスは 4.1 と同様 HCC = VCC = 8、HCL = VCL = 7 である。

いずれのデータに対しても評価関数 $F(W_j)$ を用いた場合が最も良い結果を得ていることがわかる。

さらに、文献[5]で "Difficult Example" と呼ばれている例題に対しても同様のテストを行なったので、その結果を表3に示す。

本配線手法	163
Z を W_1 に固定	172
Z を W_2 に固定	182
Z を W_3 に固定	176

表3 "Difficult Example" に対する実験結果(チャネル幅) 単位はピッチ
HCC = VCC = 8、HCL = VCL = 7

このうち本手法を適用した場合の配線結果を図15に示す。この例に対しても評価関数を用いた本手法が最も良い結果を得ている。

4.3 実際のデータに対する結果

表4に実際のデータに対する結果(処理時間)を示す。このうち CHIP2 に対する配線結果を図16に示す。

5. あとがき

本文では、グリッドフリーチャネル配線問題に対し、重み付制約グラフという概念を導入し、それを用いて幹線割当を行なう手法を提案し、実験結果を示した。

本手法は既に当社のゲートアレイ自動レイアウトシステム[6]において採用されている。前処理として上下制約ループの解除を行なっている為、未配線は発生していない。

またビルディングブロック方式のLSIレイアウトの際に問題となるチャネル上下辺の凸凹に対しても図17の様に割当てられる最上のトラックの発見を容易にする包絡線(Envelope)を定義する

ことにより．対応が可能となる。

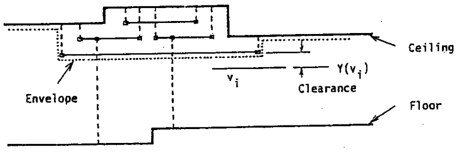


図 17．Envelope の導入

謝辞

最後に．本研究の機会を与えて下さった．集積回路事業部副事業部長　木村征二氏に深く感謝いたします。また本文をまとめるにあたり．貴重な助言及び批判を戴いた．集積回路事業部．第6技術部　神戸尚志氏，谷貞宏氏　に心から感謝いたします。

参考文献

[1] T. Yoshimura and E.S. Kuh : " Efficient Algorithm for Channel Routing ", IEEE Trans. on CAD, vol. CAD-1, no.1. pp. 25-35 (1982)

[2] 福井正博．築山修治．白川功．尾崎弘 :" チャネル配線の一手法 " 信学技報 CAS81-121 pp. 91-98 (1981)

[3] 下山博義．佐藤興二．八原俊彦 : " 配線格子の無いチャネル配線ルーチンの開発 " 信学技報 CAS79-44 pp. 35-39 (1979)

[4] M.L. Liu : " An Algorithm for Two-layer Channel Routing with Cyclic Constraints " UCB/ERL M81/82 (Nov. 1981).

[5] G. Persky et al. : " LTX-A Minicomputer-based System for Automationed LSI Layout" , J. Design Auto. & Fault-Tolerant Computing Vol. 1, 3, pp. 217-255 (1977)

[6] T. Chiba, N. Okuda, T. Kambe, I. Nishioka T. Inufushi, and S. Kimura : " SHARPS : A hierarchical layout system for VLSI, " Proc. 18th Design Auto. Conf., pp. 820-827 (1981)

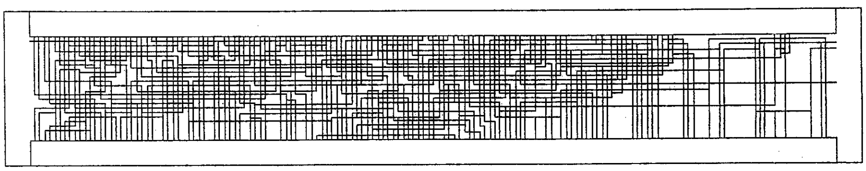


図 15 "Difficult Example" に対する配線結果 (HCC=VCC=8, HCL=VCL=7)

	CHIP 1	CHIP 2	CHIP 3	CHIP 4
ブロック数	187	664	1204	1315
ピン数	1238	4872	4185	9543
ネット数	201	643	675	1246
チャネル数	9	17	23	23
CPU 時間	1'48.26"	8'45.09"	8'37.61"	19'47.58"

表 4. 実際のデータに対する処理時間
(CPU 時間はチャネル配線のみ)

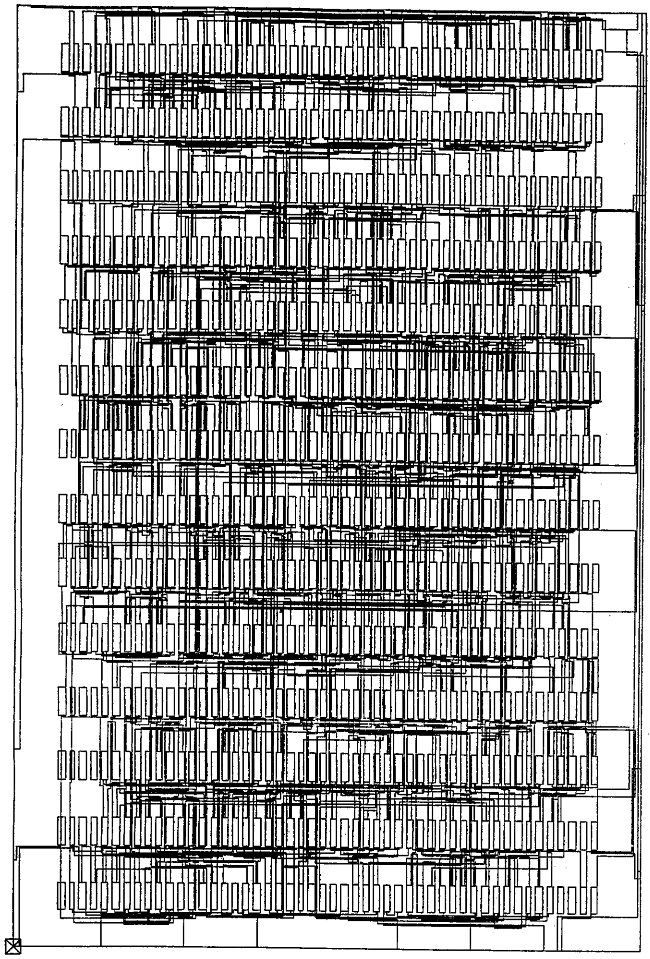


図.16　CHIP 2 に対する配線結果
（HCC＝VCC＝8，HCL＝VCL＝7）