

## 配線幅とバッファ挿入を考慮したタイミングドリブン概略配線手法

西丸 由貴 小出 哲士 若林 真一

広島大学 工学部

〒739 広島県東広島市鏡山一丁目4番1号

\*E-mail: yuki@ecs.hiroshima-u.ac.jp

本稿では、VLSI チップのレイアウト設計において、スタンダードセル配置に対する概略配線手法を提案する。配線の際、配線幅の拡大とバッファ挿入によって配線遅延を減少させ、与えられたタイミング制約を満たす配線を実現する。提案配線手法は、まず部分配線ごとに配線遅延を最小にする配線形状を求め、各ネットの配線木を生成する。次に各ネットに対して、タイミング制約に対するクリティカル度や配線幅拡大・バッファ挿入の効果を考慮して、配線幅拡大・バッファ挿入の処理を行う。そして、最初に得た配線形状に近い形で実配線を行う。

## A Performance-Driven Global Routing Method with Wire-Widening and Buffer-Insertion

Yutaka NISHIMARU Tetsushi KOIDE Shin'ichi WAKABAYASHI

†Faculty of Engineering, Hiroshima University

4-1, Kagamiyama 1 chome, Higashi-Hiroshima 739, JAPAN

\*E-mail: yuki@ecs.hiroshima-u.ac.jp

In this paper, we present a performance driven global routing method with wire-widening and buffer-insertion for large scale standard cell layout. In routing, widening the width of a wire segment and inserting buffers allow us to reduce the path delay so that we can obtain a high-performance layout. In the proposed routing method, first, a routing tree is obtained with minimizing the delay of a net from the source of a net to each root of subtrees of the tree. Next, check the criticality of each net to determine which nets are suitable for wire-widening and buffer-insertion. Finally, HV-routing of nets is performed based on the routing trees obtained in the first phase, considering timing constraint.

## 1 まえがき

VLSI の設計は大変複雑であるため、幾つかの段階に分けて行われる。その中で回路をチップ上に実現する設計段階をレイアウト設計、実装設計といい、配置設計と配線設計とから成る [10, 11, 13]。配線設計とはチップ上に配置された素子間の配線を行うことであるが、この際、仕様で与えられたタイミング制約を満たし、かつ全ての配線を行わなければならない。

近年、半導体技術の進歩による微細化技術によって、大規模な回路を実現した VLSI が作られるようになってきている。現在は  $0.25\mu\text{m}$  ルールが実用化されているがこのように非常に小さなデザインルールの場合、配線抵抗が非常に大きくなり、それによって配線遅延が増加するために、可能最小幅での配線だけでは高速動作するチップが実現できなくなっている。そのため、配線幅調整 (Wire-sizing) やバッファ挿入といった技術が使われるようになってきており、スタンダードセル方式では、クロック周波数が  $100\text{MHz}$  以上、搭載ゲート数が数百万というチップが実現されている。

配線幅調整 [2, 3, 5, 6, 7] とは、タイミングのクリティカルなネットの配線幅を大きくし配線抵抗を小さくすることで配線遅延の減少を実現する方法で、配線面積の増加によるチップ面積の増加や、配線可能性を考慮しながら行う。

またバッファ挿入 [8] とは、同じくタイミングのクリティカルなネットに対して、その途中に新たにバッファを挿入し、遅延に関係する配線・ゲートキャパシタンスを調節し遅延を小さくするもので、これは素子数が増加することからチップ面積や、素子の配置などを考慮しながら行う必要がある。

これまでの従来手法の中には、最適配線幅・挿入バッファを決定する手法もあるが、それらは複数のネットの実配線を想定しておらず、非常に計算時間がかかるため、回路全体に対しての適用は不可能である。また、ほとんどが配線の後処理であり、かつ実配線を想定していないため、配線幅を太くしたネットや、新しく挿入したバッファのレイアウトを考慮できていない。

そこで、本研究では配線幅を 2 種類に限定し、それぞれ専用の XY 方向配線層を設け、配線を行う手法を提案する。この際、遅延を小さくする配線形状を見積もり、配線幅・バッファ挿入を考慮し、複数ネットの実配線を行っていくことにより高速動作可能なチップを実現する。

## 2 Delay model

本研究で取り扱う遅延モデルは以下のような Elmore モデル [4] とする。

$$c_{n_v} = \alpha l_{n_v} \cdot w_n + cf \cdot l_{n_v} \quad (1)$$

$$r_{n_v} = \beta l_{n_v} / w_n \quad (2)$$

$$d_{n_v} = r_{n_v} \left( \frac{c_{n_v}}{2} + c(T_v) \right) \quad (3)$$

$$D_{n_v} = \sum_{j \in \text{path}\{0,v\}} d_{n_j} \quad (4)$$

ネット  $n$  について、 $c_{n_v}, r_{n_v}$  はそれぞれノード  $v$  に入ってくるネットセグメントのキャパシタンスとレジスタンスで、 $l_{n_v}$  ( $v$  に入ってくるセグメントの配線長) と、 $w_n$  (ネットの配線幅) の関数になっている。

$c(T_v)$  はノード  $v$  を root とする配線木のキャパシタンスの和である。 $D_{n_v}$  はネット  $e$  の root からノード  $v$  までの遅延である。また、ゲートの出力等価抵抗を  $R_g$ 、ゲートキャパシタンスを  $C_g$ 、バッファのゲート遅延を  $D_b$  とする。遅延パラメータは表 1 [1, 2] を用いる。ディープサブミクロンの設計ルールでは、これまでは無視できていたフリンジ容量の割合が大きくなってきているためこれを考慮することが重要である。

Parameters	value
Min Driver Resistance ( $\Omega$ )	310
Min Gate Capacitance ( $fF$ )	1.0
Wire Resistance $\alpha$ ( $\Omega/\square$ )	0.180
Wire Capacitance $\beta$ ( $fF$ )	0.030
Fringing Capacitance $cf$ ( $aF/\mu\text{m}$ )	150
Buffer Gate Delay ( $nsec$ )	0.5

表 1 遅延パラメータ ( $0.5\mu\text{m}$  CMOS モデル)

## 3 配線幅、バッファ挿入による遅延の変化

ここでは簡単な形状の配線について、配線遅延を Elmore モデルで見積もった際の、配線幅やバッファ挿入位置に対しての配線遅延の変化を示す。

図 1 のような直線セグメントについて配線幅とバッファの挿入位置を変化させた。その際の遅延の変化を示したものが図 2, 3 であり、ここでは  $l_1 + l_2$  = 一定として各配線セグメントの長さを変化させた。グラフの縦軸は  $N1 \sim N2$  間の遅延、横軸は  $l_1$  の割合を示す。

まず図 2 は配線幅を  $0.5\mu\text{m}$  としてバッファを挿入した場合としない場合の遅延の変化を示したものである。グラフから配線長の長いものにおいて効果が大きくなることがわかる。 $10000\mu\text{m}$  を越えると

1nsec以上の遅延減少が実現可能である。また挿入位置に関しては中央部分で大きな効果が得られる。図3は配線幅を $1.0\mu\text{m}$ としてバッファ挿入をおこなったものである。図2と比べると、この場合も配線長の長いものにおいて減少量が大きく、割合でみると10~30%程度の改善がなされている。



図1 配線例

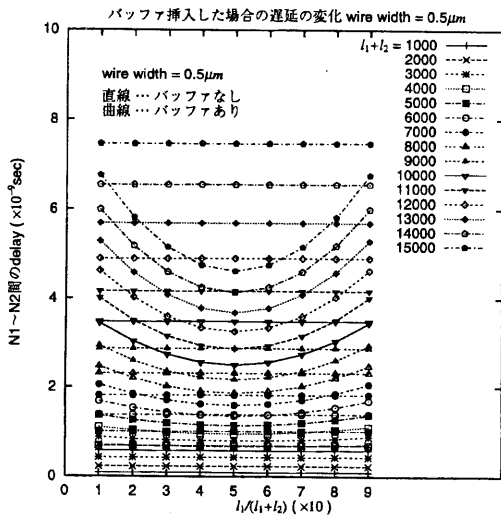


図2 バッファ挿入の効果(1)

#### 4 レイアウトモデル

本稿で取り扱うレイアウトモデルは図4のようなもので、素子はスタンダードセルとし、最下段の層(素子層)に列状に並べる。セル間配線はその上に専用4層メタル配線層を用意する。最初の1,2層は細い配線幅で配線を行う層で、その上の3,4層は太い配線幅で配線を行う層である。ただし、素子内配線を第1,2配線層で実現しその部分を配線禁止領域として指定することによって、全体で4層の配線モデルも取り扱うことが可能である。

また配線処理中に挿入されたバッファは素子層に加えられるものとする。

#### 5 問題の定式化

本手法で取り扱う配線問題を定式化する。

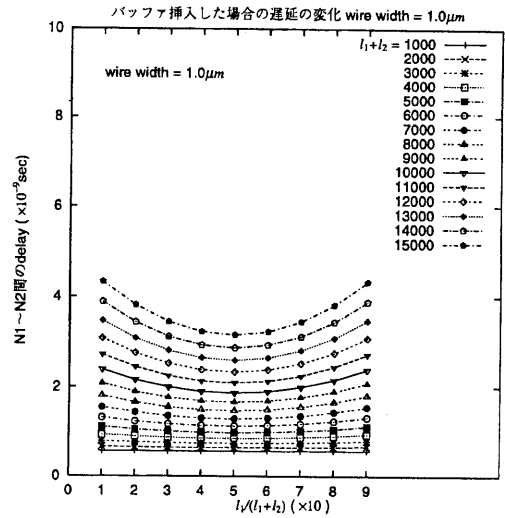


図3 バッファ挿入の効果(2)

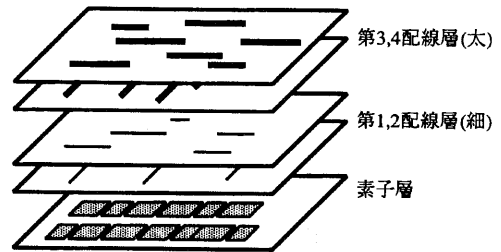


図4 レイアウトモデル

【定義】セル集合  $m_i \in \mathcal{M}$

ネット集合  $n_i \in \mathcal{N}$

タイミング制約集合  $\mathcal{T}$

遅延パラメータ集合  $\mathcal{P}$

ネット  $n_i$  の配線幅  $w_{n_i} = Wf(n_i)$

$Wf: \{n_i\} \rightarrow \mathcal{W}$

配線幅集合  $\mathcal{W} = \{W, M_w \cdot W\}$  ( $M_w$  は定数)

配線間隔  $s_1, s_2$

列許容伸長率  $R$

最小 grid 幅  $g_{w_{min}}$

【入力】ネットリスト  $(\mathcal{M}, \mathcal{N})$ ,  $\mathcal{P}$ ,

セル情報(位置, 大きさ, PIN 位置)

$\mathcal{T}$ , 各 PAD の位置, 各層の配線幅・配線間隔,

$R, g_{w_{min}}$

【目的】100%配線, 配線長最小化

【制約】タイミング制約 $T$

【出力】セル位置 (挿入 buffer も含む), 各ネットの配線経路・層

各ネットは配線幅を  $W$  から選択し, それぞれ専用の層へ配線する. 問題の入力はネットリスト, セル情報, 遅延パラメータ, タイミング制約等で, 目的は100%配線, 配線長最小化である. 出力はセル位置, 各ネットの配線経路・配線層である.

## 6 手法の概要

本研究では4層配線を考慮した配線手法を提案する.

入力は既に配置が終わったレイアウトデータとする. 配線層は2種類に分かれ, 最初の2層は配線幅の狭い線を配線するXY方向の層で, 後の2つは配線幅の広い線を配線するXY方向の層である. 素子を配置する層と配線層とは基本的に分けられているので, それぞれの層の配線領域は全面使用可能であるが, マクロセルがある部分などをユーザが配線禁止領域として指定することもできる.

配線幅は  $W$  と  $M_w \cdot W$  の2種類とし, それぞれの配線層に対応させる. 配線幅  $W$  の層と  $M_w \cdot W$  の層とでは, 配線グリッドの1格子の大きさは  $M_w$  倍の差がある. 各ネットはそのネットを通るパスの遅延や, 配線混雑度を考慮して割り当てられる層が決まる. また, 配線幅を太くするだけでは遅延の改善が不十分な場合は, タイミング違反率の大きなパス上のネットにバッファを挿入し遅延の改善を行う. この時挿入されたバッファは, 適切な場所に配置され, 既に配置されている素子と共に配置改良が行われる.

## 7 グリッド

グリッドはレイアウト平面上に等間隔 ( $Gw$ ) でXY方向それぞれに設ける. ここでこのグリッドによるグラフを  $GG = (GV, GE)$  とする. 節点はXY方向のグリッドの交点とし,  $gv_{ij} \in GV$  を左上からみてX方向に  $i$ , Y方向に  $j$  のところに位置する節点とする. また,  $gv_{ij}$  と  $gv_{kl}$  を結ぶ枝を  $ge_{ijkl}$  とする. 1 grid の幅は  $Gw$  としユーザが指定するものとする. ただし  $w_i, s_i$  をそれぞれ  $i$  層の配線幅, 配線間隔とすると,  $Gw$  は  $(w_1 + s_1)$  と  $(w_2 + s_2)$  の公倍数でなければならない. つまり,  $D_{max_i} = \frac{Gw}{w_i + s_i}$  が  $i$  層の 1grid あたりの配線可能本数となる.

## 8 アルゴリズムの概要

### 8.1 アルゴリズム

提案配線アルゴリズムの概要は以下の通りである.

- Step 1: 各ネットについて, ネットを囲む最小矩形の半周近似で配線長を見積る.
- Step 2: タイミング解析を行い, 各ネットの遅延スラックを計算する.
- Step 3: Step 2 で計算されたスラックを考慮し, 配線順序, 幅広配線, バッファ挿入等の決定を行い, 配線をする.
- Step 4: 全てのネットの配線終了後, タイミング解析を行い, 違反をしているパスがあればそのパス上の全てのネット, もし配線ができなかったネットがあればそのネットの端子を含む最小矩形に入っている配線をいくらか引き剥がし, 配線できなかったネットから順に同じ処理を行い再配線していく.
- Step 5: Step 2 ~ 5 を全てのネットがタイミング違反なしに配線されるまで行う.

以下で, 主要ステップを詳しく述べる.

### 8.2 Step 2 (遅延スラック計算)

ここでは, 最初のネットの配線順序を決定するために各ネットに対してタイミング制約に対するスラックを定義する. まず Step 1 で得られた配線長を基に, 被制約回路の各ゲートにおいて source からそのゲートまでの最大遅延時間を計算する. そして次に sink 側からさかのぼることによって各ゲートにおける要求遅延時間を計算する. そして各ゲートにおいてこの差を計算しその値をそのゲートに入ってくるネットの source からその sink までのスラックとする. Step 3 での配線は, このスラックのもっとも小さな値を持つパスを求め, そのパス上のネットから順に行う. 図5の例では, 上のパスのスラックが1で最小のためこのパス上のネットから配線を行っていく. ネット  $n$  の source から sink  $v$  までのスラック  $Sl(n, v)$  を以下のように定式化する.

$$Sl(n, v) = \min_{p_j \in P_n} rq_j(v) - \max_{p_j \in P_n} ar_j(v) \quad (5)$$

$P_n$  : ネット  $n$  を通るパスの集合.

$rq_j(v)$  : パス  $p_j$  の sink から  $v$  までの要求遅延.

$ar_j(v)$  : パス  $p_j$  の source から  $v$  までの実遅延.

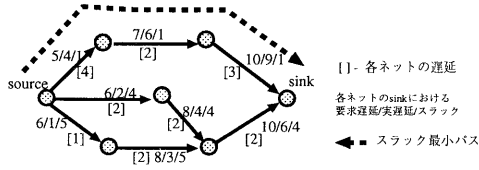


図 5 スラックの計算

### 8.3 Step 3

ここで配線を行っていく。各時点でのグリッド  $ge_{ijkl}$  の配線本数を  $NR_{ijkl}$  とする。

まずネット  $n$  の routing-graph ( $RG(n) = (RV(n), RE(n))$ ) (図 6) を定義する。このグラフは  $GG = (GV, GE)$  を基に作られる。ネット  $n$  の各端子を通る水平・垂直線の交点を節点  $rv_{ij}$  とし、さらに grid 間隔が最小間隔  $gw_{min}$  以上の場合、その部分の間隔が  $gw_{min}$  以下になるようにグリッドラインを付け加える。この際できるだけ既配線量の少ないラインを選択する。そのグリッドラインによってできた節点も  $RV(n)$  に加える。これは、グリッド数が少ないため候補となる配線経路数が少なくなり、配線不可能な場合が出るのを防ぐためである。ただし、大きな routing-graph を作成しなければならない場合は、問題の複雑さとグリッドを細かくする無意味さを避けるために最小間隔を  $X, Y$  方向それぞれ  $\frac{BX(n)}{F} \cdot gw_{min}, \frac{BY(n)}{F} \cdot gw_{min}$  とする。ここで  $BX(n), BY(n)$  はそれぞれ  $n$  の端子を囲む最小矩形の  $X$  方向、 $Y$  方向の長さ、 $F$  は基準となる最小矩形の大きさを表す定数である。つまり  $RG(n)$  がある大きさを越えるとそれ以降はノード数が一定の  $RG(n)$  となる。そして水平・垂直方向で隣接する節点間  $rv_{ij} - rv_{kl}$  に枝  $re_{ijkl}$  を設けてそれらの集合を  $RE(n)$  とする。各枝はその長さをコストとして持つ。ただし、既に配線容量が飽和している部分には枝をつけない。

配線手順は以下のように3つのフェーズからなる。

- Phase 1: 配線形状の決定。
- Phase 2: 配線幅拡大 or バッファの挿入の選択。
- Phase 3: 実際に配線。

以下で各フェーズについての説明を行う。

#### 8.3.1 Phase 1 (配線形状の決定)

ここでは、各ネットに対してその配線形状を決定する。ここで提案する手法は [12] の手法に基づくものである。

あるパスの遅延は式 4 のように表されるが、こ

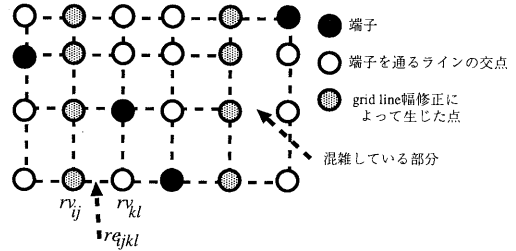


図 6 routing-graph ( $RG(n)$ )

れは

$$D_{n_v} = r_{n_0} \beta L_n + r_{n_0} \sum_{k \in Sink(n)} C_k + \alpha \sum_{k \in Sink(n)} C_k Lkn_v + \alpha \beta \sum_{j \in path\{1, v\}} l_{n_j} L(T_{n_j}) \quad (6)$$

のように書換えることができる。ここで  $Sink(n)$  はネット  $n$  の sink 集合、 $C_k$  は sink  $k$  のキャパシタンス、 $Lkn_v$  は source から sink  $k$  までと  $n_v$  までのパスの共通部分の長さ、 $L(T_{n_j})$  は  $n_j$  以降の tree の総配線長である。この式の第 2 項は各ネットについて一定である。また第 3 項より大きなキャパシタンス  $C_k$  にかかる  $Lkn_v$  は小さくしなければならないこと、第 4 項からは各部分木の長さ  $L(T_{n_j})$  を最小にしなければならないことがわかる。つまり、キャパシタンスの大きな部分木は root に近くしなければならないということがいえる。しかしこうするとネットの総配線長が長くなり、第 1 項の値が大きくなるのでこの項も考慮にいれておかなければならない。以上の考察がこのフェーズの処理を進めていく上での基礎となる。

図 7 の例で説明する。まず各 sink を source からみた sink の位置で circular order をつける (図 7(a))。例では  $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow E$  の順である。この順で隣り合ったノード同士を組み合わせさせていき tree を作成していく。この際 alphabetic tree [12] の考えに基づきキャパシタンスの小さなもの同士を組み合わせさせていくが、その際に作成される部分木の配線キャパシタンスも小さくなるようにする。まず、この順序で最小のキャパシタンスを持つ sink から  $\pm k$  の範囲で隣り合う sink に対してその sink のキャパシタンスと、2 点のマンハッタン距離の配線キャパシタンスとの和が最小になる sink を選択する。そして図 8 のようにその選ばれた 2 つの sink

と source を接続する点 R を決定するのであるが、これは式 6 と対応させて、

$$\begin{aligned} & r_{n_0} \beta (d(A, R) + d(B, R) + d(C, R)) \\ & + \alpha (C_B + C_C) d(A, R) \\ & + \alpha \beta d(A, R) (d(B, R) + d(C, R)) \quad (7) \end{aligned}$$

を最小にする点 R の座標を求める。ここで  $d(A, R)$  は A-R 間の距離で最小矩形の半周近似で見積もる。これを解いて R の座標を得て、それを部分木の root としそのノードにはその tree のキャパシタンスを重みとしてつける。この重みは以降この部分木をさらに連結していく際に式 7 の  $C_B, C_C$  の値として用いる。以降各 subtree に対して同様の連結操作を行い、配線長の見積もりのされた配線トポロジを得る (図 7(b))。

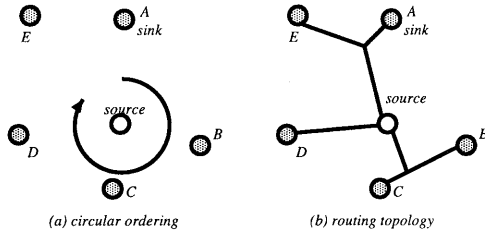


図 7 sink の順序付け

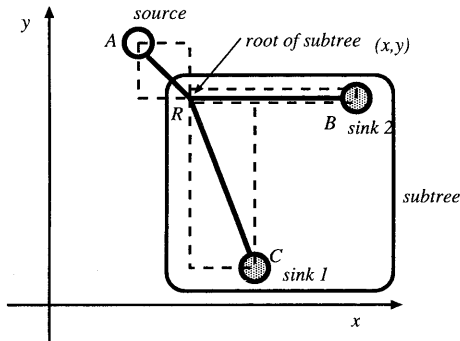


図 8 2 つの sink(subtree) の連結

### 8.3.2 Phase 2 (配線幅拡大・バッファ挿入の決定)

このフェーズでは、各ネットを配線幅を広げるか、バッファを挿入するかのどちらによって遅延を

改善するかを決定する。この処理はフェーズ 1 で求めた配線トポロジを基に行う。

ここでの目的は各ネットをどちらの方法で遅延改善を行うかを定めることで、それを 2 つの評価値に基づいて行う。その評価値で各ネットをソートするのであるが、まず、配線幅拡大の評価順を扱うキューとバッファ挿入の評価順を扱うキューの 2 つのキューを用意する。前者の評価値は、

$$\max_{j \in \text{Sink}(n)} \max_{i \in \text{path}(0, j)} \frac{(\alpha L_{ii} - R_b)(C(T_i) - C_b)}{SI(n, j)} \quad (8)$$

である。これはバッファ挿入において、その点以降の遅延減少量が

$$(\alpha L_{ii} - R_b)(C(T_i) - C_b) - R_b C_b \quad (9)$$

で表されるためである。ここで  $R_b, C_b$  はそれぞれバッファの出力等価抵抗、ロード容量である。この順に各ネットをキューに入れ、配線時にこの式によって、バッファ挿入の最大効果が得られかつそのネットの Slack が小さなものが選ばれる。

また後者の評価値は、

$$\frac{Ln \cdot |\text{Sink}(n)|}{\sum_{i \in \text{Sink}(n)} SI(n, i)} \quad (10)$$

である。これは式 1,2 より幅が  $k$  倍になると配線抵抗は  $1/k$  倍になるが配線容量はフリッジ容量のために  $k$  倍にはならない。そのため、式 3,4 より配線長が長いネットの遅延減少量が大きくなると言える。よって式 10 によって、配線時に配線幅拡大の効果が大きくかつ Slack の小さなネットから選択されていく。

### 8.3.3 Phase 3

ここでは phase 2 のキューに入っているネット順を参照し実際に配線を行う。配線順序はまず配線幅キューに入っているネットから始め、次にバッファキューに入っているネットを配線していく。その際は列長の伸長率を考慮し、バッファ挿入を決定する。もし、後の方のネットの処理中に幅広配線できないネットが生じた場合はそのネットはバッファキューの方で配線処理を行う。

配線は routing graph 上で、phase 1 でトポロジを決めた際と同じように sink の方から配線していく。それぞれのノードから迷路法により配線を伸ばしていき (図 9)、2 つの sink を連結する internal node 付近で共有節点を決定し、できた tree を subtree と

する。つづいてまた同様に、それらの subtree の共有節点を決定し root に向けて配線を行っていく。

2つの sink の接続手順は、まず図のように1つの sink と internal node の接続を行う。maze route expansion は処理時間短縮のため両ノードから行っていく。その経路が見つかったら、その経路を障害物とみなしてもう1つの sink の接続を行う。ただし第3,4層で配線を行ってるときに、Phase 1で見積もった配線長の1倍以内で経路が見つからない場合は、最初の見積もりと遅延が大きく異なってしまうので、配線を取りやめ、あとで第1,2層での配線を行う。また、最初に決まった経路を障害物としたために2つ目の sink に対する経路が見つからなかった場合は、最初の経路に隣接するノードに2つ目の maze route expansion が到達しているかどうかをチェックし、その部分で2つ目の sink の接続を行う。つまり配線長を最小にすることを考え、既配線のセグメントに対して最短経路で結線する。このチェックは maze route expansion 時に行う。

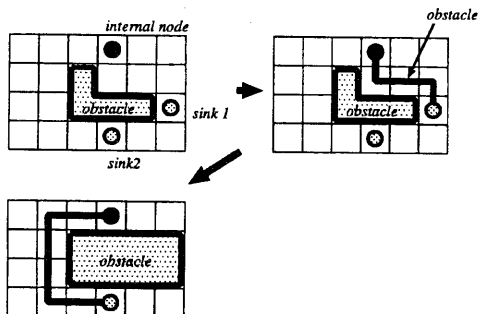


図9 maze routing

#### 8.4 Step 4

タイミング解析を行い、違反をしているパスがあればそのパス上の全てのネット、もし配線ができなかったネットがあればそのネットの端子を含む最小矩形に入っている配線がある程度引き剥がし、配線できなかったネットから同じ処理を行い再配線していく。

タイミング違反なしに全てのネットが配線できれば、処理を終了する。

#### 8.5 Step 7

最後にバッファを配置する。これは、挿入された位置にもっとも近いセル列に配置し、その場所のセルは左右にシフト移動させる。配線時に列の伸長率の制約を考慮しているので、ここでは決定されたバッファを挿入していく。

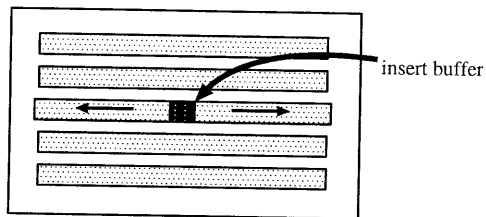


図10 buffer 挿入

## 9 あとがき

本稿では幅広配線、バッファ挿入を考慮した概略配線手法を提案した。現在、提案手法をワークステーション上にプログラムとして実現中であり、今後、提案手法の実験的評価を行う予定である。さらにアルゴリズムの高速化、1つのネットに対する複数バッファの挿入の考慮など、提案手法の更なる改良を行う予定である。

## 参考文献

- [1] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins: "Rectilinear steiner tree with minimum Elmore delay," Proc. of the 31st ACM/IEEE Design Automation Conference (1994).
- [2] J. Cong and C.-K. Koh: "Simultaneous driver and wire sizing for performance and power optimization," Proc. of International Conference on Computer-Aided Design (1994).
- [3] J. Cong, B. Preas and C. L. Liu: "Physical models and efficient algorithms for over-the cell routing in standard cell design," IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., 12, 5 (1993).
- [4] W. C. Elmore: "The transient response of damped linear networks with particular regard to wideband amplifiers," J. Appl. Phys., Vol.19, pp. 55-63 (1948).
- [5] T. D. Hodes, B. A. McCoy and G. Robins: "Dynamically-wiresized Elmore-based routing constructions," Proc. of International Symposium on Circuits and Systems (1994).
- [6] J. Lillis, C.-K. Cheng, T.-T. Y. Lin and C.-Y. Ho: "New performance driven routing techniques with explicit area/delay trade-off and simultaneous wire sizing," Proc. of the 33rd ACM/IEEE Design Automation Conference (1996).
- [7] N. Menezes, S. Pullela and L. T. Pileggi: "Simultaneous gate and interconnect sizing for circuit-level delay optimization," Proc. of the 32nd ACM/IEEE Design Automation Conference (1995).
- [8] S. Pullela, N. Menezes and L. T. Pileggi: "Post-processing of clock trees via wiresizing and buffering for robust design," IEEE Trans. on Comput.-Aided Design Integrated Circuits & Syst., Vol.15, No.6, pp. 691-701 (1996).

- [9] C. Sechen, K.-W. Lee, B. Swartz, D. Chen and M. Lee: "The TimberWolfSC Standard Cell Placement and Global Routing Package, User's guide for version 4.2c," Yale University (1987).
- [10] K. Shahookar and P. Mazumder: "VLSI cell placement techniques," ACM Computing Surveys, Vol. 23, acm press, pp. 143-220 (1991).
- [11] 菅野, 堀口: "ULSI 設計技術," 1-2 章, pp. 1-65, 社団法人 電子情報通信学会 (1993).
- [12] A. Vittal and M. Marek-Sadowska: "Minimal delay interconnect design using alphabetic trees," Proc. of the 31st ACM/IEEE Design Automation Conference, pp. 392-396 (1994).
- [13] 渡辺, 浅田, 可児, 大附: "VLSI の設計 I (岩波講座マイクロエレクトロニクス)," 1,4 章, pp. 1-21, 161-232, 岩波書店 (1985).