

タイミング制約を考慮したスタンダードセル概略配線の一手法[†]

5L-3

鈴木 武志

小出 哲士

若林 真一

吉田 典可

広島大学 工学部

1 まえがき

近年のVLSI設計技術の進歩により、単位長当たりの配線抵抗、及び配線容量による信号の伝搬遅延のクロック周期に対する割合がかなり大きくなっている。このため、タイミング制約を満たし、チップ面積を最小化するレイアウト設計手法が必要である。文献[3]において、セルベースのレイアウト設計における配線遅延を考慮した概略配線手法が提案されているが、2つのレジスタ間のパスに対してタイミング解析を陽には行っていないという問題点がある。そこで本稿では、パスに対してタイミング解析を行ない、かつチャンネル密度の最小化を目的とする概略配線アルゴリズムを提案する。

2 準備

本稿で取り扱うスタンダードセルモデルでは、セルはすべて同じ高さとし、各端子はセルの上下辺に等電位端子を持つものとする。アルゴリズムで用いるグリッドグラフ $G = (V, E)$ を図1のように定義する。ここで、節点 $v \in V$ はチップ上の各チャンネルにおける端子に対応する。枝集合 E は、垂直方向枝 E_V と水平方向枝 E_H に分け、枝 $(u, u') \in E_V$ は、セルの両端の定数個の等電位端子対に対応し、枝 $(u, v) \in E_H$ は各チャンネルの節点間に存在するものとする。

配線遅延モデルとして、ネット n のソース s からロード i までの遅延 $del(n, s, i)$ を次式を用いて表す[4]。

$$del(n, s, i) = (c_{M1} \cdot W_n^{M1} + \sum_j C_j^n)(R_o^n + r_{M1} \cdot L_i^{M1}) + (c_{M2} \cdot W_n^{M2} + \sum_j C_j^n)(R_o^n + r_{M2} \cdot L_i^{M2})$$

ここで、

- $W_n^{M1(2)}$: 第1(2)層のネット n のスタイナー木の長さ、
- $L_i^{M1(2)}$: 第1(2)層のソースからロード i のパスの長さ、
- $c_{M1(2)}$: 第1(2)層の単位長当たりの配線容量、
- $r_{M1(2)}$: 第1(2)層の単位長当たりの配線抵抗、
- $\sum_j C_j^n$: ネット n のロード容量の総和、
- R_o^n : ネット n に入力するセルの出力等価抵抗。

以下に本稿で取り扱う概略配線問題を示す。

【概略配線問題】 スタンダードセルの配置、ネットリスト N 、及び配線遅延の上限値が与えられた時、タイミング制約を満たし、チップ面積を最小とするような概略配線経路を求めよ。 □

[†]“A Standard Cell Global Routing Method Considering Timing Constraint,” by Takeshi SUZUKI, Tetsushi KOIDE, Shin'ichi WAKABAYASHI, and Noriyoshi YOSHIDA, Faculty of Engineering, Hiroshima University.

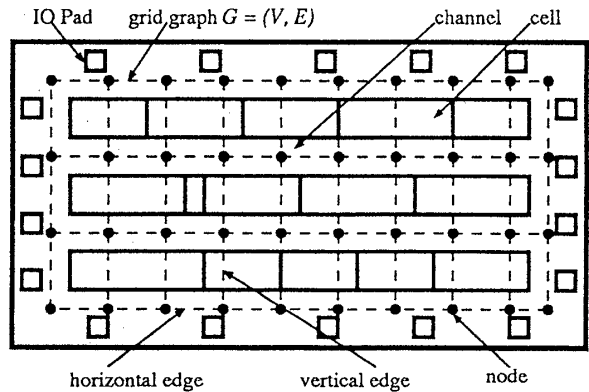


図1 グリッドグラフ $G = (V, E)$

3 提案アルゴリズム

提案アルゴリズムは大きく分けて3つのフェーズからなる。フェーズ1では、初期概略配線経路の決定、及びフィードスルーの挿入を行なう。フェーズ2では、配線遅延の制約を満たすように概略配線経路を改良し、最後にフェーズ3で、配線遅延の制約を満たした上で、チャンネル密度の最小化を目的とする概略配線経路の改良を行なう。以降では、アルゴリズムの各フェーズについて述べる。

3.1 フェーズ1: 初期概略配線

フェーズ1では、各ネットの初期概略配線経路を決定する。まず、すべてのネットに対して等電位端子対を節点とする完全グラフを構成し、枝の重みとして節点間のマンハッタン距離を与える。次に各ネットの配線遅延の下界値を配置より計算し、遅延の小さなネットから順に端子を節点とする完全グラフ上で木を構成していく。木を構成するアルゴリズムは、文献[1]のDijkstraの最短経路木を求めるアルゴリズムとPrimの最小木を求めるアルゴリズムを組み合わせたものを用い、以下の重みの小さな枝を順に取り込み各ネットの木を構成する(図2(a)-(c))。

$$\alpha \cdot L_i + l_{ij}$$

ここで、 L_i はソースからロード i までのパス長、 l_{ij} は枝 (i, j) のマンハッタン距離、 α は定数 $(0 \leq \alpha \leq 1)$ とする。その後、この木を配線混雑度を考慮しながらグリッドグラフ G 上へ埋め込み、 G 上でスタイナー木を構成し、垂直方向の枝に対応するセル列にフィードスルーを挿入する(図2(d))。

3.2 フェーズ2: 配線遅延に基づく配線改良

フェーズ2では、タイミング制約が与えられている2つのレジスタ間のすべてのパスに対して、タイミング制約を満たすように概略配線経路の改良を行なう。まず、

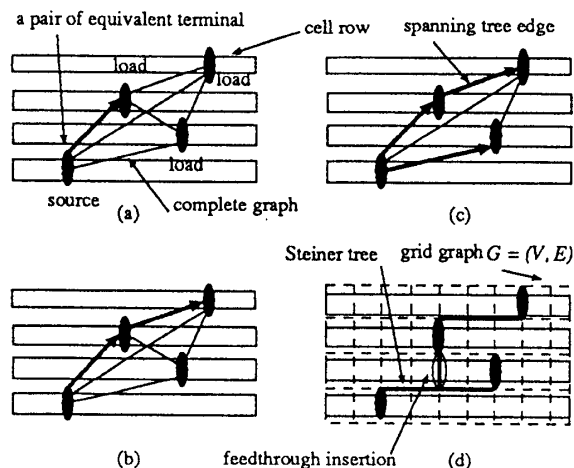


図2 各ネットのスタイナ木の構成

タイミング制約に違反しているバス上の各ネットの経路に対して、(1) ネット n の経路 s, v 間の遅延 $d_c(n, s, v)$, (2) その経路の遅延の下界値 $d_l(n, s, v)$, (3) その経路を通っているバス数 $Inf(n, s, v)$ [5], に基づく重みを以下の式により与える。

$$w(n, s, v) = Inf(n, s, v) \times (d_c(n, s, v) - d_l(n, s, v))$$

$d_l(n, s, v)$ は、配線遅延モデルの W を最小木の長さ、 L をソースからロードまでの最短経路長とすることにより求める。

この重みの最も大きい経路に対して、まずソースからロードまでを最短経路で結び、その他の枝に対してフェーズ1と同じ重み付けを行わない木を求め、グリッドグラフ上に埋め込む。この枝 (s, v 間の経路) は以後定数回の改良の間はロックし、改良される枝として選ばれないようにする。この操作を、すべてのバスがタイミング制約を満たすまで繰り返し行なう。以下にフェーズ2の各ステップを示す。

Step2.1 制約に違反しているバスを求める。

Step2.2 バス上の各ネットの経路に対して、 $w(n, s, v)$ を計算する。

Step2.3 $w(n, s, v)$ の最も大きいネットを配線遅延と配線混雑度を考えながら再配線する。

Step2.4 すべてのバスがタイミング制約を満たしていれば終了。そうでなければ Step2.1 へ。

3.3 フェーズ3: チャンネル密度に基づく配線改良

フェーズ3では、タイミング制約を満たす範囲でチャンネル密度が減少するように概略配線経路の改良を行なう。まず、文献 [2] のゼロスラックアルゴリズムにより各ネットのソース、ロード間 (s, v 間) に対してスラックを求め、それを s, v 間の許容配線遅延と考え、許される配線長を計算する。そして、許容配線長を超えずチャンネル密度に基づくコストが最も小さくなるような概略配線経路を選択し、この2点間の経路を変更する (図4)。以下にフェーズ3の各ステップを示す。

Step3.1 バス上の各経路に対して、要求遅延時間と実際の遅延時間の差 ($Slack(n, s, v)$) を計算する。

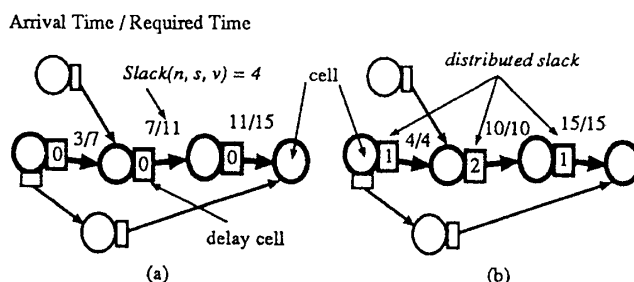


図3 ゼロスラックアルゴリズム

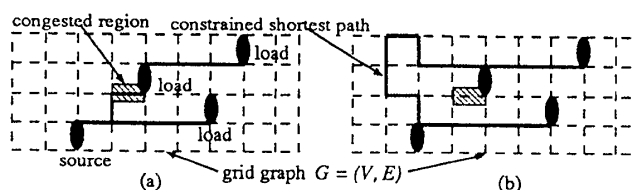


図4 概略配線経路の改良

Step3.2 経路上で最もスラックの値が小さい経路 (n, s, v) を選択する。

Step3.3 経路 (n, s, v) を通り、バス上のスラックの値がすべて $Slack(n, s, v)$ と等しいパス ($n', s, u \rightarrow \dots \rightarrow (n, s, v) \rightarrow \dots \rightarrow (n'', s, w)$) を求める (図3(a))。

Step3.4 パス ($n', s, u \rightarrow \dots \rightarrow (n, s, v) \rightarrow \dots \rightarrow (n'', s, w)$) 上の経路に対して、チャンネル密度に基づくコストを与え、そのコストに基づきスラック $Slack(n, s, v)$ の値を分散させる (図3(b))。

Step3.5 分散されたスラックに基づき、チャンネル密度に基づくコストを最小とする制約つき最短経路を求める。

Step3.6 影響された他の経路の値を更新。

Step3.7 すべての経路の処理が終わっていない場合には Step3.2 へ。

Step3.8 すべての経路のスラックが0になるか、定数回改良が行なわれない場合には終了。そうでなければ、Step3.1 へ。

4 あとがき

今後の課題としては、提案手法を計算機上で実現し、シミュレーション実験を行なうことがある。

参考文献

- [1] C. J. Alpert, et al.: "A direct combination of the Prim and Dijkstra constructions for improved performance-driven global routing," Proc. ISCAS, pp. 1869-1872 (1993).
- [2] P. S. Hauge, et al.: "Circuit placement for predictable performance," Proc. ICCAD, pp. 88-91 (1987).
- [3] J. Huang, et al.: "An efficient timing-driven global routing algorithm," Proc. 30th DAC, pp. 596-600 (1993).
- [4] E. S. Kuh, et al.: "Recent advances in timing-driven physical design," Proc. APCCAS, pp. 23-28 (1992).
- [5] M. S. Tanaka, et al.: "Minimum delay placement with influence of nets and hierarchical clustering," Proc. CICC, pp. 663-666 (1994).