

5 J-5
 メタヒューリスティックに基づくタイミング制約を考慮した
 フロアプランニング手法

山崎 晋哉* 中矢 真吾* 若林 真一* 小出 哲士**

*広島大学大学院工学研究科 **広島大学ナノデバイス・システム研究センター

1 はじめに

半導体技術の進歩により、VLSI レイアウト設計におけるパフォーマンスの考慮の重要性が益々増大している。このようなディープサブミクロン ULSI チップにおいては回路遅延全体に占める配線遅延の割合が支配的となるため、機能論理設計における回路遅延の短縮だけでは対処できず、配線遅延を考慮したレイアウト設計手法が不可欠となる。本稿ではタイミング制約が与えられたビルディングブロックレイアウトに対するフロアプランニング手法に注目し、フロアプラン、およびバッファ挿入と配線幅調整を考慮した概略配線を同時に求める手法を提案する。

提案手法では、配置モデルとしてスライシング構造を採用し、ソフトモジュールも考慮する。配線遅延モデルとしては Elmore 遅延モデルを仮定する。本手法は実行過程を 3 段階に分け、それぞれの段階においてフロアプランを評価するコスト関数を変えることにより解の探索範囲と評価精度を制御し、質のよい解を高速に求めることを可能にする。また、解探索手法として SA (Simulated Annealing) と TA (Tabu Search) の 2 種類のメタヒューリスティック手法を用いており、これらを単独に、または組み合わせで適用し、それぞれの場合のパフォーマンスを実験的に評価する。

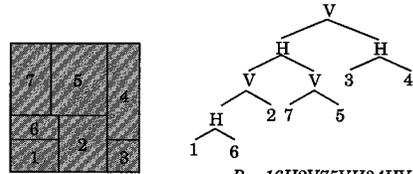
2 準備

2.1 スライシング木

本稿では、配置モデルとして配置領域を水平、垂直の線分で再帰的に分割して得られるスライシング構造を採用し、スライシング構造の表現としてスライシング木 [3] を用いる。スライシング構造はソフトモジュールを扱うことが容易であり、コーディングが簡潔でチップ面積とモジュールの座標も容易に求めることができるという特徴を持つ [4]。図 1(a), (b) にそれぞれスライシング構造とそれに対応するスライシング木の例を示す。図 1(b) で表されるようにスライシング木はポーランド記法により文字列 P で表現できる。

2.2 問題の定式化

入力: モジュールの集合 M (ハードモジュール M_H , ソフトモジュール M_S , $M = M_H \cup M_S$), 許容されるソフトモジュールの形状 M'_S , ネットリスト \mathcal{N} , タイミング制約 $Cons(\mathcal{N})$, バッファ (b_i), 配線幅 (w_i) の種類
 出力: 各モジュール $m_i (m_i \in M)$ の位置, 配線幅調整とバッファ挿入位置と配線経路



(a) スライシング構造

(b) スライシング木

図 1 スライシング木

目的: チップ面積 A_{chip} , 配線面積 A_{wire} , 配線遅延逆反 $Total_{vio}$, バッファ面積 A_{buf} の重み付き和の最小化

制約: チップの許容アスペクト比 $AS_{req} = \frac{H}{W}$ (H : チップ高さ, W : チップ幅), 各ソフトモジュール $m_j (m_j \in M_S)$ の許容アスペクト比 as_{m_j} , 1 ネットあたりの挿入バッファ数 B_{max} , タイミング制約 $T(\mathcal{N})$

3 提案フロアプランニング手法

提案手法は、ランダムに生成された初期解に対し、2 種類のメタヒューリスティック手法 (SA, TS) を単独に、もしくは組み合わせで適用することで、従来手法よりも高い解探索能力を実現する。

提案手法では解を評価するためのコスト関数が必要である。コスト関数の評価のために、提案手法では与えられた許容解 S に対し、文献 [4] で提案されている手法によりソフトモジュールの形状を決定し、 S が表すフロアプランのトポロジに対するチップ面積最小のフロアプランを求める。次に、各モジュールの端子位置を決定し、端子位置に基づいて各ネットの概略配線の評価する。ここで、各ソフトモジュールの形状の総数は各モジュールの面積に比例して設定する。概略配線の評価においては、バッファ挿入、配線幅調整を考慮する。ただし、バッファの挿入位置等の制約をすべて考慮したフロアプランと概略配線を求めることは多大な計算時間を必要とし、また、探索の初期においては解空間の大域的探索が重要であり、解の評価に対しては高い精度を必要としないため、文献 [1] の手法と同様、SA の実行途中でコスト関数を変更することにより計算時間の短縮を図る。提案手法では実行過程を 3 段階に分け、それぞれにおいて異なるコスト関数を採用し、徐々にフロアプランを詳細化していく。また、提案手法では多数のネットを高速に評価するため、ネットの端子位置に基づいて配線評価テーブルを作成し、テーブル検索により各ネットの配線の評価する [2]。

†“A Timing-Driven Floorplanning Method Based on Meta-Heuristics”, Shinya YAMASAKI*, Shingo NAKAYA*, Shin-ichi WAKABAYASHI*, Tetsushi KOIDE**, *Graduate School of Engineering, Hiroshima University, **Research Center for Nanodevices and Systems, Hiroshima University. e-mail: {shinya.wakaba}@ecs.hiroshima-u.ac.jp, koide@sxsys.hiroshima-u.ac.jp

3.1 コスト関数

以下では提案手法の3つの段階をそれぞれステージ1, ステージ2, ステージ3と呼ぶ。ステージ1では解空間の大域的探索を重視してなるべく多くの解の評価を可能とするために配線を大まかに見積ることでフロアプランの評価時間を短縮する。ステージ2ではより精度の高い配線評価を行うため、概略配線経路の配線面積と挿入されるバッファ数を見積る。最後にステージ3では全てのネットがタイミング制約を満たすようにモジュールの配置を行う。

各ステージのコスト関数は以下のように与えられる。SAを用いる場合のSAの温度パラメータはステージが進みコスト関数が増加するごとに再設定を行う。TSを用いる場合も同様に、ステージごとに近傍解の個数、タブ期間等の各パラメータを再設定する。

以下に各ステージにおけるコスト関数を示す。 α_1, β_2 などはユーザがあらかじめ与えるパラメータである。
[コスト関数1]

$$f_{1, \text{cost}} = A + \alpha_1 W \quad (1)$$

$$W = \sum_{i=1}^{N_{\text{net}}} \sum_{j=1}^{N_{\text{pin}}^i} \frac{l(t_j^i)^2}{(N_{\text{pin}}^i - 1)} \quad (2)$$

A をチップ面積, W はネット数を N_{net} , ネット n_i のソースを s^i , シンクを $t_j^i (1 \leq j \leq N_{\text{pin}}^i - 1, N_{\text{pin}}^i$ はネット n_i のピン数)とし, ソース s^i とシンク t_j^i 間のマンハッタン距離を $l(t_j^i)$ とする。また, ソフトモジュールはすべて正方形とし, ソフトモジュールの端子はすべてソフトモジュールの中心にあるものとする。
[コスト関数2]

$$f_{2, \text{cost}} = (A + A_{\text{buffer}}) + \alpha_2 A_{\text{wire}} + \beta_2 T_{\text{slack}} \quad (3)$$

A_{buffer} を挿入バッファの総面積, A_{wire} を配線総面積, T_{slack} を違反スラックの総和とする。ここでは, ソフトモジュールの形状を文献[4]の方法で決定し, 求めたモジュールの位置からテーブル検索により配線を求める。
[コスト関数3]

$$f_{3, \text{cost}} = (A + A_{\text{buffer}}) + \alpha_3 A_{\text{wire}} + \beta_3 SV \quad (4)$$

$$SV = \sum_{i=1}^{N_{\text{net}}} (V_i)^2 \quad (5)$$

SV を各ネットの違反スラック (V_i)の2乗和とする。ここでも, コスト関数2と同様にテーブル検索で配線を求める。

3.2 TSを用いたステージ3

各ステージでSAを用いる場合はコスト関数が異なるだけで解の探索方法は同じである。一方, TSを用いる場合は, ステージ1, 2においては近傍解をランダムに生成するのに対し, ステージ3では, タイミング違反のネット数を削減することを目的として近傍解を生成する。すなわち, 最大タイミング違反ネットを選択し, このネットがタイミング制約を満たすようにモジュールの再配置を繰り返すことにより探索を進める。この場合, 選択したネットとモジュールをタブとして設定し, これらが一定期間内に再び近傍解生成オペレータの対象となることを禁止する。こうして, フロアプランニングの最終段階ですべてのネットのタイミングを満たすような局所的改良を行うことで制約を満足する許容解を生成する。終了条件には, 反復回数とタイミング違反ネットが削減した場合の2つを設定する。

4 実験的評価

提案手法をC言語により実現し, UltraSPARC-II(450MHz)ワークステーション上でシミュレーション実験を行った。実験には, モジュール数46(ソフトモジュールを31含む), ネット数2564, 面積下界値24.93[mm²]のデータを用いて行なった。配線幅は0.18, 0.25, 0.50[μm]の3通りとし, バッファは1種類としている。また, ネットのタイミング制約はすべてのネット共通とし400[ps]とした。また, 従来手法($W-L$)としてSAに基づくWong-Liuの手法を用いる。この手法では, コスト関数をチップ面積と配線長(半周近似)の線形和で見積り, 得られた解に対して提案手法の配線評価手法を適用することにより最終解を得る。提案手法は, ステージ1と2にSAを単独で用いた場合(*Pro2.SA*)と, ステージ1にSAを用い, ステージ2にTSを組み合わせて用いた場合(*Pro2.HB*), さらに, ステージ1と2をSAを用いて評価した解をステージ3でSAとTSをそれぞれ適用して評価した場合(*Pro3.SA, Pro3.TS*)の結果を示す。実験では各手法に対しそれぞれ10回ずつ試行した。表1に実験結果を示す。

実験結果より, 提案手法の全ての組み合わせで大幅な配線遅延違反の改善が行なわれていることがわかる。特に全てのネットがタイミング制約を満たすようにステージ3を用いて評価した場合には, 全てのステージでSAを用いた*Pro3.SA*よりも, ステージ3にTSを用いて最大タイミング違反ネットを選択し, このネットがタイミング制約を満たすように再配置を行う*Pro3.TS*の方が, 優れた値を生成していることがわかる。

また, *Pro3.TS*は半数の試行で全てのネットがタイミング制約を満たす許容解を生成している。このことより, TSは局所改良に優れていると思われ, TSを最終ステージで使用することで, それまでにSAで十分な探索を行った最良解をさらに改良し, 優れた許容解を生成可能であることが予想される。今後, さらに実験と考察を行なう予定である。

表1 実験結果 (平均値)

Method	St.	A_{chip}	A_{wire}	T_{vio}	N_{vio}	CPU
<i>W-L</i>	-	25.64	3.57	646.07	18.4	10529
<i>Pro2.SA</i>	1	27.28	3.78	943.54	23.6	472
	2	26.02	3.38	136.03	7.3	12358
<i>Pro2.HB</i>	2	25.80	3.41	121.40	7.7	11808
	3	26.17	3.52	30.05	4.2	16140
<i>Pro3.SA</i> (best)	3	26.13	3.31	0.00	0	16392
	3	25.97	3.13	55.51	3.3	16450
<i>Pro3.TS</i> (best)	3	25.78	2.84	0.00	0	16984

A_{chip} : チップ面積 [mm²], A_{wire} : 配線面積 [mm²], T_{vio} : 違反スラックの総和 [ps], N_{vio} : 違反ネット数, CPU: CPU時間 [sec]

5 あとがき

今後の課題としては, 大規模データへの適用, プログラムの高速化などがある。

文献

- [1] H. Chen, H. Zhou, F. Y. Young, D. F. Wong, H. H. Yang and N. Sherwani: "Integrated floorplanning and interconnect planning," Proc. International Conference on Computer Aided Design, pp.354-357 (1999).
- [2] 中矢真吾, 山崎晋哉, 若林真一, 小出哲士: "概略配線を考慮したフロアプランニングに対するメタヒューリスティック手法," DAシンポジウム2001, pp.169-174 (2001).
- [3] S. M. Sait and H. Youssef: "VLSI Physical Design Automation," IEEE Press (1995).
- [4] L. Stockmeyer: "Optimal orientations of cells in slicing floorplan designs," Information and Control, Vol.57, pp.91-101 (1983).