

配線混雑度指向のMin-Cut 分割による
タイミング駆動配置手法

3H-2

高橋 一浩 寺井 正幸

三菱電機株式会社 半導体基盤技術統括部 EDA 技術部

1. はじめに

LSIの微細化に伴って、配線負荷が信号の伝播遅延に大きく影響するようになったため、タイミング駆動配置技術が必須になっている。これまでに発表されているタイミング駆動配置手法は、クリティカルパス(タイミング制約を課せられたパス)に含まれるネットまたは端子対間の仮想配線長のみを制御の対象としてきた。しかしながら、ゲート遅延に比べて配線遅延が大きな比重を占めるディープサブミクロンLSIでは、配線集中による迂回配線や隣接配線間容量の増大によって、配線終了後にタイミング制約違反が生じる恐れがあるため、配線長だけでなく配線混雑度の低減にも注力する必要がある。

本稿では、従来のMin-Cut分割手法[1]にクリティカルネットの周囲の配線混雑度を低減する機能を組み込んだ独自の分割手法によるタイミング駆動配置手法を提案する。

2. 対象とする配置問題とMin-Cut配置手法

対象とするのは、基板サイズが固定されており、配線集中による迂回配線が生じやすいEA(Embedded Array)の配置問題である。回路はセルと、セルの端子間の接続を表わすネットからなる。セルには、I/Oセル、メガセル、標準セルの3種類があるが、I/Oセル、メガセルは予め配置されており、標準セルのみを配置する。基板上には、セル配置のためのスロットと、配線のための配線格子線が定義されている(図1参照)。配線格子線の間隔は、格子線上にある配線線分と、他の格子線上にある同層の配線線分とが設計規則を満たすように決定する。従って、ある領域内の配線格子線数から、その領域内を通過可能な配線線分の数がわかる。

タイミング制約は、任意の二端子間のパス(信号伝播経路)上での信号伝播遅延時間(以下、遅延)の上限値として与えられる。パス上の入力端子から次段ゲートの入力端子までの経路をパスセグメントと呼ぶ(図2参照)。ディープサブミクロンLSIでは配線遅延が無視できないので、パスセグメン

t_{ij}の遅延t_{ij}は、[2]同様、ゲート遅延と配線遅延の和として表わす。ゲート遅延は入力波形の鈍り具合sとセグメントが含まれるネットの負荷容量Wの関数で表わされる。本手法では、種々のs、Wに対するゲート遅延が予め算出され、表形式で保持されているものとする。配線遅延の計算式としてはElmore遅延がよく使われる。しかし、配置時にElmore遅延を求めるための正確な配線経路を予測することは困難であるため、我々はKuhら[3]の導出したElmore遅延の上限値を表わす式を用いる。パスの遅延は、そのパスに含まれるパスセグメントの遅延の和となる。

我々の目的は、配線終了後に抽出したパスの実遅延がタイミング制約を満たすようにセルを配置することである。このため、配線処理においてクリティカルパスに含まれるネットの遅延を最小にする配線経路が生成し易いように、そのネットの配線長だけでなく、ネットの配線が通過する可能性が高い範囲の配線混雑度も低減する必要がある。

我々は、配線混雑度を低減するための配置手法としてMin-Cut配置手法を選択した。基板上に、複数の、カットラインと呼ばれる線分を設定し、その1本々々について「基板領域をカットラインに沿って二分し、カットラインと交差するネット数(カット数と呼ぶ)が最小になるように、分割によって生じた領域にセルを割り付ける」という処理を行ない、配置を決定する手法である。階層的に基板領域及びセル集合の分割を繰り返すため、タイミング制約をネットの重みに変換するタイミング駆動配置手法に適用し易く、しばしば利用される。しかし、Min-Cut配置手法には、分割が後回しとなる領域で配線集中が生じやすい、セル内パタン(配線障害物)を考慮していない等の問題があり、その混雑度低減の性能は十分ではない。次章では、この問題を解決するための、我々が提案する手法について述べる。

以下では、基板全体を二つの領域に分ける処理をレベル1の分割と呼び、レベルiの分割によって生じたすべての領域を二分する処理をレベルi+1の分割と呼ぶ。

3. 提案するタイミング駆動配置手法

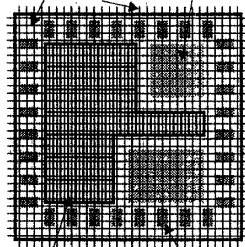
3.1 全体処理フロー

まず最初に、提案する手法の全体処理フローを示す。

- 1: L ← 1
- 2: while(未処理のカットラインがある){
- 3: タイミング制約を無視してレベルLの分割を行なう
- 4: 仮想遅延値を見積もり、パスのスラック[4]を求める
- 5: スラックを分配し、パスセグメント毎の遅延の上限値を求める
- 6: 遅延の上限値を考慮してレベルLの分割を改善する
- 7: L ← L + 1
- 8: }

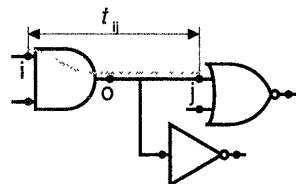
タイミング制約が配線混雑度と与える影響を小さくするため、配線混雑度のみを考慮した分割(3行目)を行なった後

配線格子線 メガセル



スロット I/Oセル

図1 配置配線モデル



$$t_{ij} = f(s, W) + r_{oj} \cdot W$$

f(s, W): ゲート遅延算出関数
r_{oj}: oj間の配線抵抗

図2 パスセグメントの遅延計算

に、仮想遅延値を見積もり、タイミング制約違反が生じている部分を中心に分割改善を行なう(6行目)。スラックの分配は[4]と同様の手法を用いる。以下では、3行目の分割、及び6行目の分割改善について述べる。

3. 2 配線混雑度指向の Min-Cut 分割

我々の分割手法の特長は、目的関数を分割によって生じる二つの領域内の配線格子線使用率(領域内の仮想配線長+領域内セルの内部パターンが塞ぐ格子線長)/領域内の配線格子線長の最大値の最小化としたことである。領域内の格子線使用率の低減により、領域内の配線混雑度を低減できる。

セルの集合Cを、領域Rを二分することによって生じる領域R_A、R_Bのそれぞれに割り付けるべきC_A、C_Bに分割する処理のフローを示す。なお、初期分割結果C_{A0}、C_{B0}は与えられるものとする。

1: 分割前処理

- ・C_A ← C_{A0}, C_B ← C_{B0}. C内の全セルに移動可の印を付ける
- ・最大格子線使用率uの初期値u₀を求める
- u = MAX{(e_A+o_A)/g_A, (e_B+o_B)/g_B}、ただし、
- e_A(e_B): R_A(R_B)内の仮想配線長
- o_A(o_B): C_A(C_B)のセル内パターンが塞ぐ配線格子線長
- g_A(g_B): R_A(R_B)内の配線格子線長

2: 入れ替えセルの選択

- ・R_AまたはR_B内のセル密度(領域内セルが占めるスロット数/領域内のスロット数)が上限値d_{limit}を越えるとき、セル密度の高い領域から、移動可の印が付いたセルの中でゲインが最大のものを選ぶ
- ・R_AまたはR_B内のセル密度がd_{limit}以下のとき、格子線使用率の高い領域から、移動可の印が付いたセルの中でゲインが最大のものを選ぶ
- ・該当セルが存在すれば3へ、存在しなければ4へ

3: 選択セルの入れ替え

- ・2で選択されたセルを反対側の領域へ移し(即ちC_A、C_Bを更新し)、移動したセルに移動不可の印を付ける
- ・e_A、o_A、e_B、o_Bを更新した後、最大格子線使用率uを求め、2へ戻る

4: 終了判定

- ・3で求めた格子線使用率の最小値 min u < u₀ のとき、C_{A0}(C_{B0}) ← min u が得られたときの C_A(C_B)、1へ
- ・min u ≥ u₀ のとき、C_A ← C_{A0}、C_B ← C_{B0}、終了

ネットの仮想配線長は、セルが割り付けられた領域の中心にあるものとしてネットにつながるセルを含む最小矩形を求め、その半周囲長とファンアウト数を基に算出する。領域内の仮想配線長は、各ネットの[仮想配線長×(最小矩形と領域の重なる面積/最小矩形の面積)]の和とする。

上記のゲインとは、移動セルを選択するための関数値であり、通常そのセルが移動した際の目的関数の減少量とする([1]ではカット数の減少量)。しかし、本手法の目的関数である最大格子線使用率の減少量[MAX{(e_A+o_A)/g_A, (e_B+o_B)/g_B} - MAX{(e'_A+o'_A)/g_A, (e'_B+o'_B)/g_B}] (ただし、e'_A、e'_Bはセル移動後の領域内の仮想配線長、o'_A、o'_Bはセル移動後の領域内セルの内部パターンが塞ぐ配線格子線長)をゲインとすると、e'_A、e'_B、o'_A、o'_Bはセルが1個移動しただけで変化するため、セルが1個移動する度に全セルのゲインを更新しなければならない。これでは処理時間が増大するため、我々は[1]と同様にカット数の減少量をゲインとして採用する。

本手法では、常に格子線使用率の高い領域からセルを選択し、他方に移動させる。このとき、セルの移動元領域ではセル内パターンが塞ぐ配線格子線長が減少する。また、前述した領域内の仮想配線長の見積もり方法では、カット数が減少する限り移動元領域内の仮想配線長も減少する。このため、カット数を減少させるセルの移動は、格子線使用率の高い領域の使用率を下げるように作用し、二つの領域の格子線使用率は二つの領域の平均格子線使用率 (e_A+o_A+e_B+o_B)/(g_A+g_B) に近い値となる。ここで、カット数を減少させるセルの移動はe_A+e_Bを減少させるため、平均格子線使用率は減少する(o_A+o_B、g_A+g_Bは一定)。従って、カット数の減少量をゲインとしても、最大格子線使用率を小さくすることができる。

3. 3 タイミング駆動分割改善

本処理のフローは、前節で述べたフローと同じである。相違点は、クリティカルなネットの周囲の配線混雑度が低減されるように、仮想遅延が上限を越える(あるいは上限に近い)パスセグメントを多く含むネットの仮想配線長及びカット数に対して重み付けを行なう点である。

ネットnの重みw_nは、ネットnを駆動するゲートの入力端子(fanin(n)と記す)とネットnに接続する入力端子(fanout(n)と記す)間のパスセグメントの仮想遅延t_{ij}及び遅延上限値Uから次式を使って求める。

$$w_n = h(\sum_{i \in \text{fanin}(n), j \in \text{fanout}(n)} p_{ij}^2 / (|\text{fanin}(n)| \times |\text{fanout}(n)|))$$

$$p_{ij} = \begin{cases} t_{ij} / \alpha \cdot U_{ij} & (t_{ij} > \alpha \cdot U_{ij}) \\ 1 & (t_{ij} \leq \alpha \cdot U_{ij}) \end{cases}$$

ここで、|fanin(n)|、|fanout(n)|はそれぞれの端子数、h(x)は単調増加関数、αは0以上1以下の実数である。

この重みは、セルが移動する度に更新する。このため、カット数が変化しない場合でも、タイミング制約の達成度が向上すれば、ゲインは正となる。図3に示す例において、t_{ij}が小さくなることによって制約の達成度が向上する場合、セル3の移動によってネットnの重みが減少するため重み付けカット数は減少する。よって、セル3はセル2より優先して移動される。このように動的に重みを決定することによって、タイミング制約の達成度が向上が期待できる。

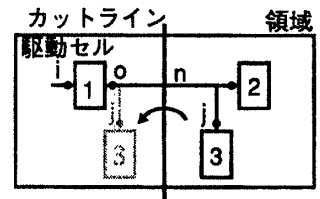


図3 遅延の減少によるゲイン

4. おわりに

本書では、配線混雑度指向のMin-Cut分割を用いたタイミング駆動配置手法について述べた。今後、本手法を実現し、評価を行なう予定である。

参考文献

[1]C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in Proc. 19th DAC, June 1982, pp.175-181.
 [2]L. N. Kannan, P. R. Suaris, and H. Fang, "A methodology and algorithms for post-placement delay optimization," in Proc. 31st DAC, June 1994, pp.327-332
 [3]E.S.Kuh and M.Shih, "Recent advances in timing-driven physical design," in Proc. APCCAS, December 1992, pp.23-28
 [4]W. K. Luk, "A fast physical constraints generator for timing driven layout," in Proc. 28th DAC, June 1991, pp.626-631