

## タイミング制約を考慮した非線形計画法に基づく スタンダードセル配置手法

楠元 寛史 若林 真一 小出 哲士 吉田 典可

広島大学工学部

〒724 広島県東広島市鏡山一丁目4番1号

本稿では、VLSIチップのレイアウト設計において、与えられたタイミング制約を満たすセル配置手法を提案する。パフォーマンスを考慮する上で配線遅延は無視できないが、本手法では遅延が配線長の2乗に比例する配線遅延モデルを用いる。アルゴリズムでは、まず、タイミング制約を考慮した高速な手法で初期配置を行い、次に、非線形計画法に基づく手法で配置改良をする。この配置改良手法はクリティカルパスの伝搬遅延時間の制約の下で総配線長を最小化する問題を非線形計画法で解く手法である。実用性を考えて、配置改良は制約がクリティカルな部分回路に対して反復して実行される。最後に、タイミングを考慮した列割り当て手法でセルを列状に配置する。

## Performance Driven Standard Cell Placement Based on Nonlinear Programming

Hiroshi KUSUMOTO Shin'ichi WAKABAYASHI Tetsushi KOIDE Noriyoshi YOSHIDA

Faculty of Engineering, Hiroshima University

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

Hiroshima 724 Japan

In this paper, we present a new performance driven placement method for standard cell VLSIs. Our method employs an interconnection delay model, in which the delay time is proportional to the square of wire length. At first of our algorithm, initial placement is got by an efficient performance driven placement method. Next, an iterative improvement method by nonlinear programming improves the layout. The placement improvement is formulated as the problem of minimizing the total wire length subject to critical path delays. From the practical consideration, the placement improvement procedure is applied to a part of given circuit, which is critical in timing. Finally, row assignment considering timing constraint is performed.

## 1. まえがき

近年の半導体技術の進歩はめざましく、微細化によって大規模な回路を実現したVLSIが広く用いられるようになった。また、VLSIにとってパフォーマンスは重要な評価基準の一つである。しかし、ゲートや配線を小さくすると、スケーリング則<sup>[2]</sup>に従って信号の伝搬遅延そのものは減少するがゲートのスイッチング速度に対して配線遅延の割合が大きくなるという現象が生じる。このため、チップ上に実現される回路の信号伝搬遅延時間を改善するためには配線遅延を考慮することが必要である。VLSIチップのレイアウト設計においてもパフォーマンスを考慮した手法が盛んに研究されている<sup>[1,3,6,9,11,25]</sup>。レイアウト設計は配置・配線と大きく2つに分けられるが、特に配置設計におけるパフォーマンスの考慮がより重要である。これは、一度配置が決まってしまうと配線設計だけで配線長を大きく変更することが難しく、大幅なパフォーマンスの改善には限界があるからである。従って、パフォーマンスを考慮した配置設計手法についての研究が盛んになされている。

パフォーマンスを考慮した配置手法はi)伝搬遅延モデル、ii)パフォーマンスの考慮の方法、iii)配置アルゴリズムの3つの観点から分類できる。伝搬遅延モデルとして多くの手法ではElmoreモデル<sup>[4]</sup>の近似式を用いている。アルゴリズムの中でスタイナ木を構築しながら配置することは困難であるので、配線遅延を集中定数回路で近似し、ネットの端子を囲む最小矩形(バウンディングボックス)の半周長を近似配線長とする場合がほとんどである。配線遅延の近似式としては配線容量のみを考慮し、配線遅延が配線長に比例するモデルもあるが、配線容量と配線抵抗の積で配線長の2乗に比例するモデルがより正確な式として採用されることが多い。

配置アルゴリズムの中でのパフォーマンスの扱い方には大きく2つの方法がある。いわゆるネットベース手法<sup>[1,9,13,16,17,20,22-24]</sup>とパスベース手法<sup>[3,6,8,11,12,15,19,21,25]</sup>である。ネットベース手法はネット毎に許される配線遅延の上限を与える方法である。しかし、回路のクロックサイクルはフリップフロップ間の組み合わせ回路の伝搬遅延に依存するため、パフォーマンスはパスの遅延によって決定される。従って、我々が本来考えるべき問題は与えられたパス遅延の制約を満足することである。ネットベース手法では、パス全体から見れば伝搬遅延には余裕があるにもかかわらず、安全を考えて配線遅延の上限を小さく見積るためにレイアウトの自由度を不必要に制約する可能性がある。これに対して、パスの伝搬遅延を考慮する方法をパスベース手法と言う。パスベース手法はネットベース手法よりもパフォーマンスを改善でき、かつ、総配線長も小さくできる可能性があるが、

アルゴリズムの中でのパスの扱い方が複雑になり、計算時間が大きくなる傾向がある。さらに、これら2つの手法はそれぞれ制約を重みに変換する方法<sup>[1,8,13,19,24,25]</sup>とそのまま制約として扱う方法<sup>[2,6,9,11,12,15,17,20-23]</sup>に分けられる。遅延の上限を重みに変換する方法はアルゴリズム中で制約が扱いやすくなる一方、制約が必ずしも満たされない場合が生じやすい。これに対して、制約そのものを扱う場合は正確に制約を考慮できるが処理が煩雑になる。

最後に、用いられている配置アルゴリズムにはヒューリスティックな手法と解析的な手法とがある。ヒューリスティックな手法にはFM法やシミュレーテッドアニリング法に基づいた回路分割手法などを階層的に適用して配置を求める手法<sup>[1,19,23,25]</sup>や、コンストラクティブな手法<sup>[15]</sup>などがある。これに対してパフォーマンスを考慮した配置問題を数理計画法に定式化して問題を解く方法<sup>[6,11,12,18,21]</sup>が最近盛んに研究されている。これらの手法の場合、最適解を求めることは可能であるが、大規模な回路全体を一度に配置することは困難なので分割した回路をそれぞれ定式化して解くなどの工夫がなされている。

以上の議論から、望ましい手法の一つとして正確な遅延モデルを用い、パスベース手法で数理計画法を適用するようなものが考えられる。しかし、このような手法の場合、大規模な回路への適用が難しく、計算時間が大変掛かると考えられる。そのような手法でありながらこれらの困難を克服し非常に良いレイアウトを出力する手法の一つにRITUAL<sup>[21]</sup>がある。RITUALは階層的な分割に基づいているが分割には2次計画法を適用している。しかし、パスベースで階層的に分割を繰り返す場合、一度パスの要素が異なる分割に分けられてしまうと再びそのパス上の要素が一度に配置されることはなく、ネットベース手法と同じような問題が生じることになる。

本稿では与えられたタイミング制約を満たすためにパス全体の配置改良を反復する配置手法を提案する。我々の提案する手法では配線をRC集中定数回路で近似した、配線遅延が配線長の2乗に比例する配線遅延モデルを仮定する。また、非線形計画法を適用することで最適解に近い解を求めることができる。大規模な回路への適用も可能にするために、まず、パフォーマンスを考慮した高速なアルゴリズムで初期配置を求め、次に、反復改良を行う。この反復改良では遅延の大きいパスとその周辺の部分回路に対してパス遅延を制約として配線長を最小化する問題に定式化し、非線形計画法を適用する。最後に、タイミング制約を考慮した列割り当てアルゴリズムでセルを列状に配置する。我々は本手法を計算機上を実現し、ベンチマークデータに対して実験を行った。実験の結果、提案手法が有効であることを確認できた。

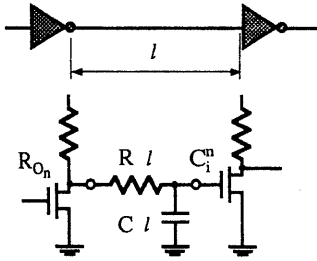


図1 配線の等価回路

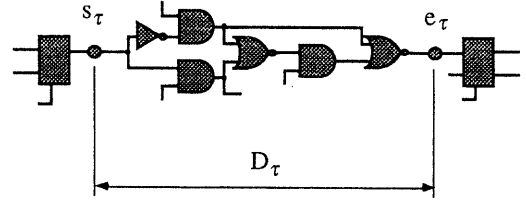


図2 タイミング制約の例

## 2. 準備

### 2.1 配線遅延モデル

本稿では配線遅延モデルとして図1のようなRC集中定数回路を仮定し、次の近似式を用いる。式の前半は水平方向（メタル第1層）の配線遅延、後半は垂直方向（メタル第2層）の配線遅延を表している。また、配線長はネットのバウンディングボックスのマンハッタン距離とする。

$$d(n, l_x, l_y) = (R_x l_x + R_{O_n}) \left( C_x l_x + \sum_{i=1}^{f_o} C_i^n \right) + (R_y l_y + R_{O_n}) \left( C_y l_y + \sum_{i=1}^{f_o} C_i^n \right) \quad (1)$$

ただし、

- $l_x$  : ネット $n$ の第1層の配線長
- $l_y$  : ネット $n$ の第2層の配線長
- $C_x$  : 第1層の単位長あたりの配線容量
- $C_y$  : 第2層の単位長あたりの配線容量
- $R_x$  : 第1層の単位長あたりの配線抵抗
- $R_y$  : 第2層の単位長あたりの配線抵抗
- $R_{O_n}$  : ネット $n$ に入力するセルの出力等価抵抗
- $C_i^n$  : ネット $n$ の端子 $i$ に接続するセルの入力容量
- $f_o$  : ネット $n$ のファンアウト数

### 2.2 タイミング制約

タイミング制約とは回路が満足すべき信号伝搬遅延の制約であり、primary inputからフリップフロップの入力端子間、フリップフロップの出力端子からフリップフロップの入力端子間、および、フリップフロップの出力端子からprimary output間に与えられる。制約は図2のように3項組 $t_\tau = (s_\tau, e_\tau, D_\tau)$ の集合 $T$ で与えられる。ただし、 $s_\tau, e_\tau$ は端子対、 $D_\tau$ はそれらの端子の間の最大許容遅延時間、すなわち、端子間に許される信号の伝搬遅延

時間の最大値である。全ての最大許容伝搬遅延時間を等しい値にすると、同期回路のクロックサイクル時間の制約と等価になる。

### 2.3 問題の定式化

タイミング制約を考慮したセル配置問題を次のように定式化する。

- 【入力】 1) 論理回路  $L = (M, N)$   
 $M$ : セル集合,  $N$ : ネット集合  
 2) タイミング制約集合  $T$   
 3) 物理的パラメータ
- 【出力】  $M$ の配置
- 【目的関数】 総配線長  $\rightarrow$  最小化
- 【制約条件】 1) レイアウトモデルに従う  
 2) タイミング制約を満たす  $\square$

$T$ は予め回路シミュレーションによって与えられているものとする。また、物理的パラメータとは、(1)式の配線遅延を与える各パラメータを指す。仮定するレイアウトモデルはポリセル型のスタンダードセル、または、ゲートアレイであり、ゲートアレイ型の場合、入力にチップのサイズやスロットの座標をさらに加えなければならない。セルは列状に並べるか、与えられたスロットに割り当て、セル同士が重なりあってはならない。また、セルの中には位置が固定されたものがあるが、マクロは含まない。配線は2層HVモデルを仮定する。

ここで、幾つかの定義をしておく。各タイミング制約 $t_\tau \in T$ に対し、始点が $s_\tau$ 、終点が $e_\tau$ であるような任意のパスをクリティカルパスと呼び、次のように定義する。

$$p_\pi = (M_\pi, N_\pi)$$

ただし、

- $M_\pi$  :  $p_\pi$ に含まれるセルの集合
- $N_\pi$  :  $p_\pi$ に含まれるネットの集合

また,

- $P$  : クリティカルパス集合
- $P_\tau$  :  $t_\tau \in T$  に対するクリティカルパス集合
- $dreq_\pi$  :  $p_\pi \in P$  の最大許容遅延時間
- $dact_\pi$  :  $p_\pi \in P$  の実伝搬遅延時間

タイミング制約  $t_\tau \in T$  に対し, 次のような部分回路を被制約回路と定義する.

$$L_\tau = (M_\tau, N_\tau)$$

ただし,

$$M_\tau = \bigcup_{p_\pi \in P_\tau} M_\pi \quad : L_\tau \text{ のセル集合}$$

$$N_\tau = \bigcup_{p_\pi \in P_\tau} N_\pi \quad : L_\tau \text{ のネット集合}$$

また,

$$dact_\tau = \max_{p_\pi \in P_\tau} dact_\pi \quad : t_\tau \in T \text{ の実伝搬遅延時間}$$

### 3. アルゴリズム

#### 3.1 アルゴリズムの概要

提案アルゴリズムは3つのフェーズで構成されている。フェーズ1ではタイミング制約を考慮しながら高速にセルの配置を求める手法<sup>[19]</sup>を適用することにより, セルの初期配置を求める。フェーズ2ではフェーズ1で得られたセルの配置を改良する。このフェーズは反復改良法であり, 1回の試行では, 制約の違反が大きいクリティカルパスを含む部分回路に対して非線形計画法を適用する。最後にフェーズ3ではフェーズ2の配置結果に基づき, タイミング制約を考慮してセルを列に割り当てる。なお, レイアウトモデルにスタンダードセル方式を仮定した場合はフェーズ1の前に仮のチップサイズとセル列数を与えられた回路から算出しておく。

以下の節で各フェーズについて説明する。

#### 3.2 初期配置

フェーズ1の目的はタイミング制約をできるだけ満足するようなセルの概略配置を求めることである。タイミング制約を全て満足するような配置を求めることはフェーズ2で行うので, ここでは, 制約を大体満足し, セルが配置領域上に均一に分散すればよい。

このフェーズには以前我々が開発したタイミング制約を考慮した配置手法<sup>[19]</sup>を適用する。この手法はパスベースの重み付け手法であり, 今回提案する手法と同じ配線遅延モデルを採用している。アルゴリズムはオーソドックスな階層的4分割手法に基づいている。これは, 1つの領域に割り当てられたセルを左右2つの領域に分

割し, さらに, 各領域をそれぞれ同様に縦に2分割して元の領域を4分割する操作を階層的に反復することによってセルの詳細な位置を求める方法である。このとき, セルは割り当てられた領域の中心に配置する。ここで用いる2分割にはFiducciaらのいわゆるFM法<sup>[5]</sup>に基づいた手法を適用する。この手法は, 選択したセルを現在の領域から他方の領域へ移動した場合に領域間のカットラインを横切るネットの本数の減少分をゲインとして, ゲインが最も大きいセルから順に移動してゆく方法である。ここではこのゲインをカットゲイン  $gc$  と呼ぶ。

ここで, クリティカルパス  $p_\pi \in P$  に対するスラックを以下のように定義する。

$$sl_\pi = \begin{cases} dreq_\pi - dact_\pi & dact_\pi > dreq_\pi \\ 0 & dact_\pi \leq dreq_\pi \end{cases} \quad (2)$$

この式は次のようなことを意味している。すなわち, クリティカルパスの伝搬遅延の上限を実際の伝搬遅延が越えている場合は越えている分をスラックとし, 上限を越えていない, つまり, 制約を満足している場合はスラックはゼロである。2分割の際, セルを移動した後のクリティカルパスのスラックの減少分をスラックゲインと呼び, 次式で与える。

$$gs_\pi = sl_\pi' - sl_\pi \quad (3)$$

ただし,

$$sl_\pi \quad : \text{セルを移動する前のパス } p_\pi \text{ のスラック}$$

$$sl_\pi' \quad : \text{セルを移動した後のパス } p_\pi \text{ のスラック}$$

この手法では, 次式のようにカットゲインと全てのクリティカルパスのスラックゲインの総和の線形和をゲインとしてFM法に基づく2分割を行う。

$$g = \alpha \cdot gc + \beta \cdot \sum_{p_\pi \in P} gs_\pi \quad (4)$$

ただし,

$$\alpha, \beta \quad : \text{正の定数}$$

#### 3.3 反復改良手法

フェーズ2ではフェーズ1の配置結果を初期配置として反復改良を行う。良質な解を得るためには数値計画法の適用が望ましいが, 一般に数値計画法は計算時間やメモリの点で大規模な回路への適用が困難であるため, 部分回路への適用を考える。ここで, 配置改良の対象となるセルを可動セルと呼び, 可動セルの集合を  $M_{mov}$  と表す。可動セル以外の移動しないセルは固定セルと呼び, 固定セル集合を  $M_{fix}$  と表す。また, ネットの端子に1つでも可動セルが接続しているものがあればそのネットを

可動ネットと呼び、可動ネット集合を $N_{mov}$ と表す。それ以外のネットは固定ネットと呼び、配置中に変化しない。固定ネット集合は $N_{fix}$ と表す。

数理計画法を配置問題に適用するためには定式化において可動セルの $X, Y$ 座標を表す変数の他に可動ネットのバウンディングボックスを表す変数が必要である。ネットのバウンディングボックスを表す変数は従来の手法でも用いられており、これらの手法ではバウンディングボックスの左下の角の $X, Y$ 座標と右上の角の $X, Y$ 座標の4変数が1つのネットにつき与えられる。しかし、本手法では1つのネット $n_i$ に対してバウンディングボックスの横方向の長さ $w_i$ と縦方向の長さ $h_i$ の2変数のみを用いる。これらの変数を用いて定式化した場合、ネットのバウンディングボックスを特定するための制約式の数に大きな差はないが、変数の数が減少し、配線長の記述が従来に比べて簡潔になる。ネットのバウンディングボックスはこれらの変数を用いて次のような制約式で表すことができる。

$$CN1: \left. \begin{array}{l} x_j - x_k \leq w_i \\ x_k - x_j \leq w_i \\ y_j - y_k \leq h_i \\ y_k - y_j \leq h_i \end{array} \right\} \begin{array}{l} \forall m_j \neq m_k \in M_i \cap M_{mov}, \\ \forall n_i \in N_{mov} \end{array}$$

ただし、 $M_i$ はネット $n_i$ に接続するセルの集合である。ネットが固定セルに接続している場合は次の制約も必要である。

$$CN2: \left. \begin{array}{l} x_j - Xmin_i \leq w_i \\ Xmax_i - x_j \leq w_i \\ Xmax_i - Xmin_i \leq w_i \\ y_j - Ymin_i \leq h_i \\ Ymax_i - y_j \leq h_i \\ Ymax_i - Ymin_i \leq h_i \end{array} \right\} \begin{array}{l} \forall m_j \in M_i \cap M_{mov}, \\ \forall n_i \in N_{mov} \end{array}$$

ただし、

$$\begin{array}{l} Xmin_i : M_i \cap M_{fix} \text{の任意のセルの最小の} X \text{座標} \\ Xmax_i : M_i \cap M_{fix} \text{の任意のセルの最大の} X \text{座標} \\ Ymin_i : M_i \cap M_{fix} \text{の任意のセルの最小の} Y \text{座標} \\ Ymax_i : M_i \cap M_{fix} \text{の任意のセルの最大の} Y \text{座標} \end{array}$$

数理計画法を用いた同様の配置手法のRITUALでは、階層的に分割された部分回路に対して2次計画法を適用して回路を分割している<sup>[21]</sup>。しかし、回路を幾つにも分割すると、クリティカルパスも分割されてしまい、分

割の階層が進むに従ってクリティカルパス全体の修正ができなくなるといった問題が生じる。

この問題を解決するために本手法ではクリティカルパス自身を配置改良の対象となる部分回路に選ぶ。しかし、パス1本だけではかなり局所的であるため、十分には配置が改良されない可能性があるため、そのクリティカルパスが属している被制約回路の中から別のクリティカルパスを何本か選び、それらが覆う部分回路に対して非線形計画法を適用する。同じ被制約回路のクリティカルパス同士は元々同じタイミング制約に起因するので、それらを同時に改良することはその制約の改善により望ましい結果を生じると思われる。また、トポロジ的に近い位置にあるクリティカルパスを選ぶことで本来改良したいクリティカルパスの周囲のネットワークを同時に改良することになるので、タイミング、配線長共に大きな改良が期待できる。このようにして、選択したクリティカルパスの集合を $P_{mov}$ とすると、パス遅延の制約は(1)式を用いて次のように表せる。

$$CP: \sum_{\forall n_i \in N_\pi} d(n_i, w_i, h_i) + \sum_{\forall m_j \in M_\pi} dswitch_j \leq dreq_\pi \quad \forall P_\pi \in P_{mov}$$

ただし、

$$dswitch_j : \text{セル} m_j \text{のスイッチング遅延}$$

これらのクリティカルパスが選ばれるとそれらに含まれる部分回路が決定する。従って、可動セル集合は

$$M_{mov} = \bigcup_{\forall P_\pi \in P_\tau} M_\pi \quad (5)$$

と、表すことができる。

いま述べたようにして選んだクリティカルパスのみを制約に加えて定式化すると、可動セルを含むにもかかわらず選ばれなかったクリティカルパスが存在するために、それらのクリティカルパスの遅延が増大する可能性がある。従って、これらのパスも制約として考慮しなければならない。従って、制約CPは次のように変更される。

$$CP': \sum_{\forall n_i \in N_\pi \cap N_{mov}} d(n_i, w_i, h_i) + \sum_{\forall n_i \in N_\pi \cap N_{fix}} d(n_i, Xmax_i - Xmin_i, Ymax_i - Ymin_i) + \sum_{\forall m_j \in M_\pi} dswitch_j \leq dreq_\pi \quad \forall P_\pi \in P \mid M_\pi \cap M_{mov} \neq \emptyset$$

目的関数は基本的には配線長の総和の最小化とする。しかし、総配線長を最小化する問題を数理計画法で解い

た場合、セル同士が一箇所に集中してしまうことがしばしばある。これでは、列割り当てをするときにセルを元の位置から大きく動かさなければならなくなり、配置改良の意味がなくなってしまう。これを避けるためにセルを配置領域に均一に分散させるという制約なり目的関数なりを設定すればよいのだが、関数の凸性を保ちながら定式化することは難しい。従って、ここでは配線長の2乗の総和を最小化するように目的関数を定める。この目的関数の下ではネットの長さが平均するようになるため、ネット長の和が同じでも結果として総配線長を最小化する場合よりもセルが分散しやすいと思われる。目的関数は次式のとおりである。

$$\sum_{\forall n_i \in N_{mov}} (w_i^2 + h_i^2) \rightarrow \text{minimize}$$

以上の議論から問題を次のように定式化する。

【目的関数】

$$\sum_{\forall n_i \in N_{mov}} (w_i^2 + h_i^2) \rightarrow \text{minimize} \quad (6)$$

【制約条件】

$$CN1 \cup CN2 \cup CP'$$

この問題は、目的関数と制約式の双方に2次の項があるため非線形計画問題となる。本手法ではこの問題を解くのにシンプルで従来手法でも用いられている双対法<sup>[10]</sup>の代わりに解の収束が速いと言われる乗数法<sup>[10]</sup>を適用する。

以下にフェーズ2の概略を示す。

【反復改良アルゴリズム】

- 1)  $\forall t_r \in T$ についてタイミング解析をする,  $k=0$ .
- 2) 違反がないか,  $k=(\text{ある定数})$ なら終了.
- 3) 違反が最も大きい $t_r \in T$ を選ぶ.
- 4)  $\forall p_r \in P_r$ の中で遅延が最大のものを選ぶ.
- 5)  $P_{mov}$ を生成する.
- 6)  $\forall p_r \in P$ の中で $M_r \cap M_{mov} \neq \phi$ を満たすものを全て選ぶ.
- 7) 問題を定式化し, 非線形計画法を用いて解く.
- 8) 変化があった被制約回路についてタイミング解析を行う,  $k=k+1$ , 2)へ.

最初に、全ての被制約回路についてタイミング解析を行い、次のループで反復改良する。改良の対象とするクリティカルパスは制約を違反している被制約回路をランダムに1つ選び、その中で最も違反の大きいパスとする。クリティカルパス集合 $P_{mov}$ は、可動セルと可動ネットの数を元に定式化したときの変数の数を計算しながら、

変数の数が与えられた上限を越えないようにパス集合とセルを共有する割合が大きいパスから順に $P_{mov}$ に加えて生成する。問題を非線形計画法で解いたあとは可動ネットを含んでいる被制約回路についてタイミング解析を再度実行し、再びループを反復する。そして、タイミング制約の違反がなくなるか、ループの反復回数が与えられた上限に達すると、フェーズ2を終了する。

### 3. 4 列割り当て

このフェーズでは配置領域上に不規則に分散したセルを列状に配置する。セル列は入力与えられているか、あるいはこのアルゴリズムの前処理で算出されているものとし、セル列の数を $r$ 、セル列を $R_1, R_2, \dots, R_r$ と表す(図3)。各セル列には容量が決められており、セルの面積の総和を $S$ とすると、各セル列の容量は

$$\alpha S / r \quad \text{ただし, } \alpha \geq 1$$

とする。また、各セル列にはY方向の座標が与えられている。

アルゴリズムの中では図4のようなセル集合 $A_1, A_2, \dots, A_{r+1}$ を用いる。これは、次のようにして作る。始めにセル集合をY方向にソートする。そして上から順にセルの面積の和が $S/(2r)$ になるまでセルを取りそのセルの部分集合を $A_1$ の要素とする。 $A_2$ は $A_1$ に含まれなかった最初のセルからセルの面積の和が $S/r$ になるまで取る。以下同様に $A_r$ までの独立な集合を作り、最後の $A_{r+1}$ は $A_1$ と同じ面積の集合になる。

アルゴリズムは以下に示す通りである。

【列割り当てアルゴリズム】

- 1) 各セルを $A_1, A_2, \dots, A_{r+1}$ に割り当てる.
- 2)  $R_1 \leftarrow A_1, \forall m_i \in A_1$ をY方向に $R_1$ へ移動,  $A_1 = \phi$ .
- 3)  $R_r \leftarrow A_{r+1}, \forall m_i \in A_{r+1}$ をY方向に $R_r$ へ移動,  $A_{r+1} = \phi$ .
- 4)  $k=2$ .
- 5)  $\forall m_i \in A_k$ について $R_{k-1}$ へ移動した場合のコストを計算する.
- 6)  $m_s \in A_k$ をコストが最も減少するセルとする.
- 7)  $R_{k-1} \leftarrow m_s, m_s$ をY方向に $R_{k-1}$ へ移動,  $A_k = A_k - \{m_s\}$ .
- 8)  $m_s$ に接続するセルのコストを更新する.
- 9)  $R_{k-1}$ に含まれるセル面積の総和が容量以下なら6)へ.
- 10)  $R_k \leftarrow A_k, k < r$ なら $k=k+1$ として4)へ.
- 11) 各セル列でセルの重なりを除く.

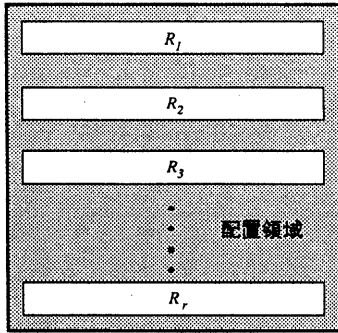


図3 セル列

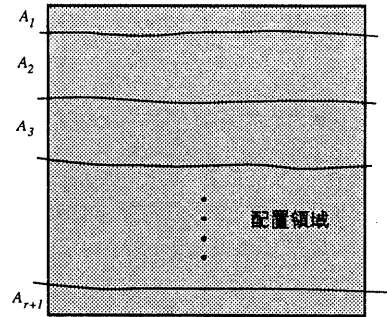


図4 セルの $A_1, A_2, \dots, A_{r+1}$ への割り当て

5)においてセルを移動するコストは次のようにして計算される。まず、全てのネット $n_i$ に対して $A_1, A_2, \dots, A_{r+1}$ の中で最初に $n_i$ に接続するセルが現われる集合を $Au_i$ 、最後に現われる集合を $Al_i$ とする。そして、 $n_i$ に接続するセル $m_j$ に対して次のような値を与える。

$$u_{ij} = \begin{cases} 0 & m_j \notin Au_i \\ 1 & m_j \in Au_i \end{cases} \quad (7)$$

$$l_{ij} = \begin{cases} 0 & m_j \notin Al_i \\ 1 & m_j \in Al_i \end{cases}$$

これを用いて次式のようにセル $m_j$ のコストを計算する。

$$cost_j = \sum_{\forall n_i \in M_n} \omega_i (l_{ij} - u_{ij}) \quad (8)$$

ただし、 $M_i$ はネット $n_i$ に接続するセルの集合を表す。また、 $\omega_i$ は $n_i$ のクリティカリティを表し、 $n_i$ を含む任意の被制約回路を $L_\tau$ とすると、

$$\omega_i = \begin{cases} 1 & dact_\tau \leq D_\tau \\ 1 + \alpha \cdot \max\left(\frac{dact_\tau}{D_\tau} - 1\right) & otherwise \end{cases} \quad (9)$$

で与えられる。ただし、

$\alpha$ : 正パラメータ

であり、 $\alpha$ が大きいほどクリティカルネットが重視される。5)では $A_k$ に属する全てのセルのついて式を計算するが8)では移動したセルについてのみコストを更新すればよい。

#### 4. 実験結果

提案手法をSun SPARCstation2上でC言語を用いて実現し、実験を行った。表1は実験に用いたMCNCベ

ンチマークデータfract-dbである。タイミング制約は最長パス上のセルのスイッチング遅延の総和を数倍した値を用いた。表2は提案手法の実験結果でフェーズ1、フェーズ2、フェーズ3終了後の配置結果を示している。表中の違反数とは違反のあったタイミング制約の数である。また、最大違反率とは全てのタイミング制約に対して実伝搬遅延時間を最大許容遅延時間で割った値の最大値を表しており、この値が1以下なら制約を満足していることになる。フェーズ1の出力は制約の違反がまだ15残っていたが、フェーズ2の結果では違反のある制約が2つ残っているものの最大違反率がほぼ1.0になり、本手法が与えられたタイミング制約を満たすセル配置を求めることが確認できた。なお、実験ではフェーズ2のループを10回反復し、1回の試行では非線形計画法の性能の都合上、定式化したときの変数の数が200以下になるよう可動セル集合を選んだ。フェーズ3の列割り当てでは、違反が多少増え、総配線長はかなり増加した。このフェーズについては今後改良の余地があると思われる。

#### 5. まとめ

本研究では、タイミング制約を考慮した非線形計画法に基づくセル配置手法を提案した。今後の課題として、他のベンチマークデータに対しての実験、他のタイミング制約を考慮した配置手法との比較実験が挙げられる。また、タイミング制約を考慮した配線手法を開発し、詳細配線までを行って評価することを考えている。

#### 謝辞

プログラムの開発にご協力頂いた本学学部生小野光博君に感謝します。本研究の成果一部は文部省科学研究費補助金一般研究(C) (課題番号05680274) による。

表1 fract-db

セル数	ネット数	IOパッド数	制約数
125	163	24	38

表2 実験結果

	違反数	最大違反率	総配線長( $\lambda$ )	CPU時間(s)
フェーズ1	15	1.14	53331	3
フェーズ2	2	1.03	50169	2413
フェーズ3	4	1.03	53862	6

## 文献

- [1] M.Burstein et al., "Timing influenced layout design", Proc. 22nd D.A.C., pp.124-130 (1985).
- [2] R.H.Dennard et al., "Design of ion implanted MOSFET's with very small physical dimensions", IEEE J. of Solid State Circ., SC-9, pp.256-268 (1974).
- [3] W.E.Donath et al., "Timing driven placement using complete path delays", Proc. 27th D.A.C., pp.84-93 (1990).
- [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, Jan. (1948).
- [5] C.M.Fidduccia et al., "A linear-time heuristic for improving network partitions", Proc. 19th D.A.C., pp.175-181 (1982).
- [6] T.Gao et al., "A new performance driven placement algorithm", Proc. ICCAD91, pp.44-47 (1991).
- [7] T.Hamada et al., "Prime: A timing-driven placement tool using a piecewise linear resistive network approach", Proc. 30th D.A.C., pp.531-536 (1993).
- [8] T.Hasegawa et al., "A new placement algorithm minimizing path delay", Proc. ISCAS91, pp.2052-2055 (1991).
- [9] P.S.Hauge et al., "Circuit placement for predictable performance", Proc. ICCAD87, pp.88-91 (1987).
- [10] 今野, 他, "非線形計画法", 日科技連出版社 (1978).
- [11] M.A.B.Jackson et al., "Performance-driven placement for cell based IC's", Proc. 27th D.A.C., pp.370-375 (1989).
- [12] M.A.B.Jackson et al., "A fast algorithm for performance-driven placement", Proc. ICCAD90, pp.328-331 (1990).
- [13] S.Kim et al., "APT: An area-performance-testability driven placement algorithm", Proc. 29th D.A.C., pp.147-152 (1992).
- [14] E.S.Kuh et al., "Recent advances in timing-driven physical design", Proc. APCCAS, pp.23-238 (1992).
- [15] I.Lin et al., "Performance-driven constructive placement", Proc. 27th D.A.C., pp.103-106 (1990).
- [16] R.-B.Lin et al., "Fuzzy logic approach to placement problem", Proc. 29th D.A.C., pp.153-158 (1992).
- [17] M.Marek-Sadowska et al., "Timing driven placement", Proc. ICCAD89, pp.94-97 (1989).
- [18] S.Mayrhofer et al., "Timing driven placement: why a unified net- and path-based approach pays off", Proc. APCCAS, pp.29-34 (1992).
- [19] 三島, 他, "タイミング制約を考慮したセルの一配置手法", 信学技報, VLD91-82, pp.17-24 (1991).
- [20] Y.Ogawa et al., "Timing-driven placement for general cell layout", Proc. ISCAS90, pp.872-875 (1990).
- [21] A.Srinivasan et al., "RITUAL: A performance driven placement algorithm for small cell IC's", Proc. ICCAD91, pp.48-51 (1991).
- [22] S.Sutanthavibul et al., "An adaptive timing-driven layout for high speed VLSI", Proc. 27th D.A.C., pp.90-95 (1990).
- [23] M.Terai et al., "A new min-cut placement algorithm for timing assurance layout design meeting net length constraint", Proc. 27th D.A.C., pp.96-102 (1990).
- [24] R.S.Tsay et al., "An analytic net weighting approach for performance optimization in circuit placement", Proc. 28th D.A.C., pp.620-625 (1991).
- [25] S.Wakabayashi et al., "Gate array placement based on mincut partitioning with path delay constraints", Proc. ISCAS93, pp.2059-2062 (1993).