設 計 自 動 化 87－4
（1998．1．30）

信号経路に対するタイミング制約を考慮した回路分割手法

南淳一朗　小出哲士　若林真一
広島大学工学部
〒739-8527 東広島市鏡山一丁目4番1号
E-mail:{south, koide, wakaba}@ecs.hiroshima-u.ac.jp

近年の集積回路の動作速度の飛躍的な向上に伴い，回路分割を行なう際にタイミング制約を考慮することが必要となってきた．本稿では，ゲートの遅延を考慮した一般的な遅延モデルを用い，順序回路のレジスタ間のタイミング制約を考慮した回路分割手法を提案する．提案手法は反復改良法に基づいており，大きく分けて2つのフェーズから構成される．フェーズ1でタイミング制約を考慮したクラスタリングを行ない，フェーズ2でFiducciaらの分割手法を拡張した拡張FM法により解の反復改良を行なう．拡張FM法ではタイミング制約を考慮した新しいゲインを導入しており，タイミング制約のある回路分割問題を扱うことが可能である．さらに，拡張FM法適用後にタイミング制約違反を取り除くために，タイミング違反パスに基づくノードの移動により違反除去を行なう．

A Circuit Partitioning Method under Path Delay Constraints

Jun'ichiro MINAMI, Tetsushi KOIDE and Shin'ichi WAKABAYASHI
Faculty of Engineering, Hiroshima University
4-1, Kagamiyama 1 chome, Higashi-Hiroshima 739-8527 JAPAN

Recent progress of operational speed of VLSI circuits has caused the designers to take timing constraints into account during circuit partitioning. In this paper, we adopt a general delay model in which gate delays of the circuit are considered, and propose a partitioning method that explicitly takes timing constraints among registers in the circuit into consideration. The proposed method is based on an iterative improvement method and consists of two phases. In phase 1, we perform a clustering so as to decrease the circuit size, and in phase2, iteratively improve an initial partition with a new partitioning method which is an extension of Fidducia-Mattheyses(FM) method. Since we extended the FM method, we can explicitly deal with circuit partitioning problems with timing constraints. After phase 2, we also apply a new path-based violation removal algorithm (PBVR) so as to remove all the timing violations.

1 はじめに

近年の半導体技術の進歩により，集積回路はますます大規模化の傾向にある．そのため，VLSIチップ設計の配置を行なう場合や大規模システムを複数のチップを用いて実現する場合などに回路分割が行なわれる [1]．

一般に，回路分割手法は組立て法に基づく手法と反復改良法に基づく手法の2種類に大きく分類することができる [1]．前者には固有ベクトルに基づくアルゴリズム [2] やネットワークフローに基づいて構築的に分割を求めるアルゴリズム [14] などがある．後者のゲイン等を用いたノードの移動による反復改良法は，(1) 面積制約の下でノードの面積が一定でないような問題を容易に扱える，(2) カット数の最小化と計算時間の短縮のトレードオフの考慮が容易である，(3) 分割の部分的な改良が容易に行なえる，などの利点がある．代表的なものとして，Kernighan-Lin(KL) 法 [10] や Fidducia-Mattheyses(FM) 法 [6] がある．FM法は，ノードの移動によるカット数の減少数，すなわちゲインが最大となるノードを移動させ，カット数の最小化を行なう反復改良法である．しかし，FM法を単独で用いると局所解に陥りやすいという欠点があるため，クラスタリング等が併せて用いられることが一般的である [1]．FM法では，解の質がノードの移動順序に依存するため，近年では，より効率の良い順番にノードが移動するようFM法のゲインを改良した手法が提案されており，非常に良い解が得られている [3–5]．

このように様々な回路分割手法が研究されているが，近年の回路の動作速度の高速化やチップ全体の遅延のうち配線遅延の占める割合の増大などによって，チップ設計の際にタイミング制約を考慮することが必要となってきた．従来のタイミング制約を扱った分割問題の遅延モデルとしては，単位遅延モデル (unit delay model) [12] や，一般遅延モデル (general delay model) [17] などがある．単位遅延モデルで [12] は異なるチップを接続する配線のみに一定の遅延を設けている．しかし，チップ内のゲートの遅延を全く考慮にいれていないため，遅延の見積りが正確ではない．一方，一般遅延モデル [17] は各ゲートの遅延を考慮し，チップ内の配線遅延は無視し，異なるチップ間をまたぐ配線のみ一定の遅延を考慮する遅延モデルである．そのため，単位遅延モデルより正確に遅延を見積もることが可能である．文献 [17] などでは，この一般遅延モデルを用い，I/O ピン数と面積制約の下で，チップの入力から出力までの最大遅延が最小となるような分割を求めるアルゴリズムを提案している．しかし，このアルゴリズムは扱える問題が組合せ回路に限定されており，順序回路を含む回路に適用する場合には，回路中のサイクルを除去して，アサイクリックグラフに変換する必要がある．そのため，順序回路の端子間にタイミング制約を持つような問題を正確に扱うことは難しいと考えられる．文献 [16] ではMCMに対するシステム分割手法が提案されている．タイミング制約はネットの端子間に与えてあり，遅延は配線の距離によって見積もられる．しかし，ゲート遅延やチップ内部の配線の遅延は全く考慮されていない．

また，これまでに著者らは，全てのレジスタ間のパスのカットが2回以内で行なえる分割手法 [7] や，MCMに対する分割手法を提案している [13]．しかし，文献 [7] ではゲートの遅延を考慮していないため，ゲートの遅延を陽に考慮する必要があるVLSIチップ設計には，そのまま用いることは難しい．また，文献 [13] では，タイミング制約を隣り合うモジュール間のみとしているため，パスにタイミング制約を持つような問題でタイミング制約を正確に扱うことは困難である．

そこで，本稿では一般遅延モデルを用い，順序回路のレジスタ間の遅延制約を考慮した回路分割手法を提案する．また，大規模回路に対応するため，タイミング制約を考慮したクラスタリング手法についても述べる．

以降では，2節で問題の定式化を行ない，3節でアルゴリズムの詳細を述べる．そして4節でシミュレーション実験の結果を示し，最後に5節でまとめと今後の課題について述べる．

2 タイミング制約を考慮した回路分割

本稿で取り扱うグラフモデルは，回路素子をノード，ネットを枝で表したハイパーグラフ $G=(V,H)$ で，各ノード $v_i \in V$ は面積 $a(v_i)$ と遅延 $d(v_i)$ を持つ．ハイパー枝 $h_i \in H$ は，1つのソースノード $S(h_i)$ とそれ以外のノード集合 $D(h_i)$ から構成される $(\{S(h_i)\} \cup D(h_i) = h_i)$．ハイパー枝 h_i を有向枝 $e_k = (S(h_i), v)(\forall v \in D(h_i))$ に2端子分解したグラフを $G'=(V,E)$ とする．

配線遅延モデルとしてはゲートの遅延も考慮に入れた一般遅延モデル [17] を用い，カットされる枝には一定の配線遅延 $d_c \gg d(v_i)$ を設ける．タイミング制約は全てのレジスタの入出力端子と回路の入出力端子間の全てのパスに設ける．レジスタノード集合を $R \subset V$，レジスタノードを $r_i \in R$，レジスタノード r_i，$r_j (\in R, i \neq j)$ 間のパス p_{ij} は，レジスタノードを始点と終点とするノードと有向枝の系列 $p_{ij} = \{r_i, e'_{k_{1ij}}, v'_{k_{2ij}}, \ldots, r_j\}$ と表すことができる (ただし，系列中のノードにレジスタノードは含まない)．パス p_{ij} に含まれるノード集合を $V(p_{ij})$，パス p_{ij} のカットされる回数を $ncut(p_{ij})$ とす

ると，パス $p_{ij} \in P$の配線遅延制約は，

$d(p_{ij}) = \sum_{v_i \in V(p_{ij})} d(v_i) + d_c \times ncut(p_{ij})$

で表される．よってパス p_{ij}のタイミング制約は $d(p_{ij}) \leq D$で与えられる．ここで Dはパス遅延の上限を表す定数である．

図1は一般遅延モデルを用いて r_1, r_2間の配線遅延を示したものである．カットされる枝の遅延は分割の目的に応じて変化させる．例えば，図2(a)のように，配置に分割を用いる場合は d_cを小さくし，(b)のように，大規模システムを複数のチップで実現するような場合には，d_cを大きな値に設定する．

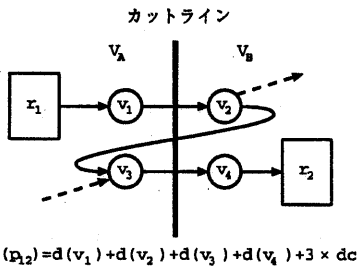

図 1 パスの遅延

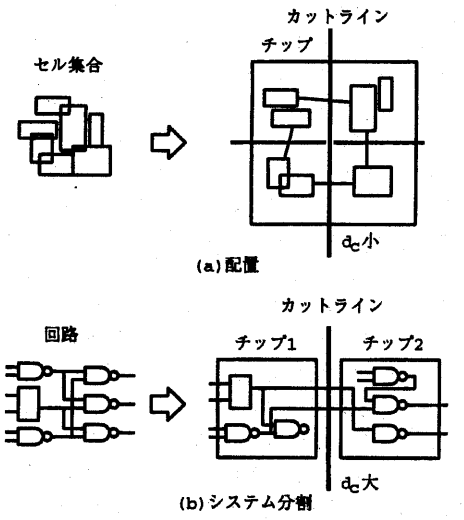

図 2 カットされる枝の遅延

本稿では回路の2分割を考える．ノードの集合 Vを互いに素な Vの2つの部分集合 V_Aと V_Bに分割することを回路の2分割という．ただし，V_A，V_Bは

$\sum_{v_i \in V_j} a(v_i) \leq \alpha \times \sum_{v_i \in V} a(v_i)$ $(j \in \{A, B\},\ 0.5 \leq \alpha \leq 1.0)$

を満足するものとする．これを面積制約という．

タイミング制約を考慮した回路2分割問題とは先に述べた面積制約とタイミング制約の下でカット数，すなわち V_A，V_Bにノードを持つハイパー枝の総数が最小となるような分割を求めることである．

【タイミング制約を考慮した回路2分割問題】

入力	$G = (V, H)$
出力	2分割 $\Phi : V \rightarrow \{V_A, V_B\}$ $(V = V_A \cup V_B, V_A \cap V_B = \emptyset)$
目的関数	カット数の最小化
制約	$\sum_{v_i \in V(p_{ij})} d(v_i) + d_c \times ncut(p_{ij}) \leq D$ (全てのパス $p_{ij} \in P$)
	$\sum_{v_i \in V_j} a(v_i) \leq \alpha \times \sum_{i=1}^{n} a(v_i)$ $(j \in \{A, B\},\ 0.5 \leq \alpha \leq 1.0)$

3 提案分割手法

まず，提案手法の概要について述べる．提案手法は大きく分けて次の2つのフェーズからなる．

フェーズ1：クラスタリング
フェーズ2：拡張FM法による分割

提案手法ではまず大規模回路を実用的な時間で分割するためにノード数を減少させる．一般にノード数を減少させるための方法としてクラスタリングが知られている [1]．本稿で扱うモデルはレジスタの入出力端子間にタイミング制約を設けているため，クラスタリングの際，レジスタ間の配線遅延を考慮する必要がある．また，フェーズ2においても，反復改良の際，配線遅延を考慮したノードの移動が要求される．以下に，各フェーズの詳細について説明する．

3.1 フェーズ1:クラスタリング

大規模回路に対して良い分割を実用的な計算時間で得るために，フェーズ1ではクラスタリングを行なう．また，クラスタリングを行なうとノード数が減少するため，計算時間は短縮される．クラスタリングとは，ノード集合 $V = \{v_1, v_2, \cdots, v_N\}$ を k個のクラスタ集合 $C = \{c_1, c_2, \cdots, c_k\}$ $(1 \leq k \leq N)$ に割り当てることである．クラスタ c_iは Vの部分集合で与えられ，$V = \bigcup_{i=1}^{k} c_i$，$c_i \cap c_j = \emptyset$ $(1 \leq i, j \leq k,\ i \neq j)$ である．フェーズ1では，以下の2つのステップでクラスタリングを行なう．

ステップ1：タイミング制約に基づくクラスタリング

ステップ 2 ：2 つの制約を考慮したクラスタリング

ステップ 1 ではレジスタ間のパスがカットされると必ずタイミング延制約違反が生じるパス，すなわち，

$$\sum_{v_i \in V(p_{ij})} d(v_i) + d_c > D$$

となるパスに含まれるノード集合 $V(p_{ij})$ をクラスタリングする (図 3).

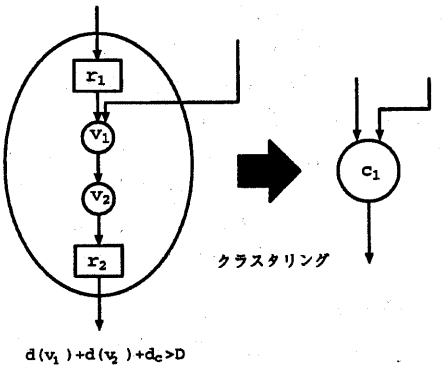

図 3 ステップ 1 のクラスタリング

ステップ 2 では，タイミング制約と面積制約を考慮したクラスタリングを行なう．$plen(e_k)$ を枝 $e_k \in E$を通るパスの最大遅延とし，$ncp_n(e_k)$ を枝 $e_k \in E$を通るパスの中で n 回までカットできるパスの数とする．すなわち枝 e_kを通る全てのパス集合を $P(e_k) \subset P$とすると，

$ncp_n(e_k) = |\{p_{ij} | \lfloor (D - d(p_{ij}))/d_c \rfloor = n, \forall p_{ij} \in P(e_k)\}|$

となる．図 4に $ncp_1(e_k)$，$ncp_2(e_k)$ を求めるアルゴリズムを示す．ncp_nを求めるためには，e_kの両端のノード v_i，v_jから全てのレジスタまでのパス集合 P_i，P_jの遅延が必要である．そのため，あらかじめ全てのレジスタからノードまでの遅延を計算しておく．各ノードまでの遅延の計算はレジスタからトポロジカルソートの順に行なえば効率良く行うことが可能である．

次に，これらの枝の情報をもとに以下の計算式を用いて各枝の重み $w_1(e_k)$ を計算する．

$w_1(e_k) = \kappa \times plen(e_k) \times (2 \times ncp_1(e_k) + ncp_2(e_k))$.

$w_1(e_k)$ はタイミング制約の厳しいパスが多く通る程大きな値になる．この枝の重み $w_1(e_k)$ を用いて枝 e_k に接続するノード v_i, v_jをクラスタリングするための $cost1(v_i, v_j)$ を

$$cost1(v_i, v_j) = w_1(e_k) + \eta \times \frac{1}{a(v_i) \times a(v_j)}$$

とする．$cost1$ の第 2 項は，クラスタサイズを均等化するためのものであり，面積の小さいノードのペアほど

```
手続き calc_ncp
{
  e_k = (v_i, v_j);
  ncp_1(e_k) = ncp_2(e_k) = ncp_m(e_k) = 0;
  for each p_ki ∈ P_i
    for each p_jl ∈ P_j
      if (d(p_ki) + d(p_jl) < D - 3d_c)
        ncp_m(e)++;
      else if (d(p_ki) + d(p_jl) < D - 2d_c)
        ncp_2(e)++;
      else
        ncp_1(e)++;
}
```

図 4 手続き $calc\_ncp(e_k)$

どクラスタリングされやすくなる．ステップ 2 では $cost1(v_i, v_j)$ の大きいノードのペアから順に全体のノード数が一定数以下になるまでクラスタリングを行う．$w_1(e_k)$ と $cost1(v_i, v_j)$ はクラスタリングを行うごとに再計算する．

次に，手続き $calc\_ncp$ の計算量について述べる．まず，$calc\_ncp$ では各ノードに対し，レジスタからのパス遅延を計算する．これは全てのレジスタからトポロジカルソートの順にノードまでの遅延を計算すればよいから，レジスタ数を $\mathcal{R}$，ノード数を $\mathcal{N}$とすると計算時間は $O(\mathcal{R} \times \mathcal{N})$ である．次に，枝 e_kを通るパスの遅延を求めるのに枝の両端のノードまでの全てのパス遅延の組合せについて調べる必要がある．枝 e_kを通るパスの平均本数を $\mathcal{P}_{ave}$，グラフの枝数を $\mathcal{E}$とすると，これは $O(\mathcal{P}_{ave} \times \mathcal{E})$ となる．よって，$calc\_ncp$ 全体の計算時間は $O(\mathcal{R} \times \mathcal{N}) + O(\mathcal{P}_{ave} \times \mathcal{E})$ である．

3.2 フェーズ 2:拡張 FM 法による反復改良

フェーズ 2 は拡張 FM 法による反復改良フェーズであり，全体として図 5のようなフローとなっている．タイミング延制約は，FM 法のゲインの計算の際に考慮する．また，配線遅延制約違反がとれなかった場合は，PBVR(Path Based Violation Removal) アルゴリズムを用いて違反パスの除去を行なう．

[初期分割]

一般に FM 法は最終結果が初期解に依存しやすいため，配線遅延制約とカット数を考慮して初期分割を求める．

まず，入力として与えられるグラフ G'から，任意の

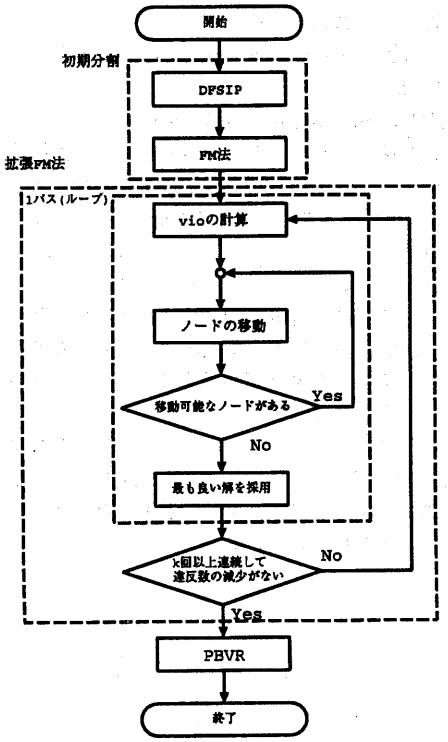

図 5 フェーズ2のフロー

レジスタノード r_iを起点とし，深さ優先探索を行なう．このときの探索の範囲はレジスタノードまでとし，パスの途中にレジスタノードは含まないようにする (図6)．探索されたノード v_jを含むクラスタ $c_k(v_j \in c_k)$ を V_Aに加える．探索終了後，V_Aに含まれるレジスタノードのうち最もクリティカルなパスの起点となっているノードを1つ選択する．図6では，r_2，r_3，r_4の中で最もクリティカルなノードを選択する．ここでクリティカルなノードとは，関数 $max\_path\_delay(v_i)$ が最大となるノードを意味する．$max\_path\_delay(v_i)$ は，ノード v_iを通る全てのパスのうち最大のパス遅延を示す．図6において，r_4が最もクリティカルなノードとして選ばれたものとすると，このノード r_4を根とし同様に深さ優先探索を行ない V_Aにノードを加えていく．図7に手続き DFSIP(Depth First Search-based Initial Partitioning) を示す．このようにして求められた分割に対し，FM 法を適用してカット数を最小化したものを拡張 FM 法の初期解とする．

[拡張 FM 法]

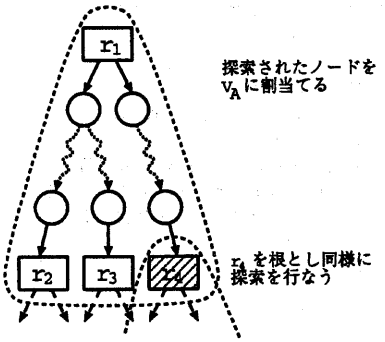

図 6 深さ優先探索

次に拡張 FM 法について述べる．拡張 FM 法は FM 法のゲインにタイミング制約違反を考慮したゲインを加えたものであり，タイミング制約違反を考慮したゲインは違反の度合が大きなパスの遅延が改善されやすくなるようにしている．また，回路全体のタイミング制約違反の度合の評価は制約違反数ではなく，以下に述べる別の尺度を用いて行なう．

まず，タイミング制約違反の度合を表すために，各枝 $e_k \in E$について $vio(e_k)$ を求める．$vio(e_k)$ は以下のように定義する．

$$vio(e_k) = \sum_{d(p_{ij})>D, \forall p_{ij} \in P(e_k)} (\lceil (d(p_{ij}) - D)/d_c \rceil)$$

すなわち，枝 e_kを通る全てのパス $p_{ij} \in P(e_k)$ で，タイミング制約に違反しているパス p_{ij}の違反パスカット数 $\lceil (d(p_{ij}) - D)/d_c \rceil$ の総和を表す．例えば，図8において，e_2，e_3には p_{13}，p_{23}の2本の違反パスが通っている．パス p_{13}はカット可能回数1回に対して3回カットされているので，違反パスカット数は2，また，p_{23}の違反パスカット数は1となる．よって，e_2と e_3は共に $vio(e_1) = vio(e_3) = 3$ となる．

また，制約違反を侵しやすいパスを優先的に改善が行えるよう $penalty(e_k)$ を用いて枝の重みを動的に変化させる．$penalty(e_k)$ は，拡張 FM 法の実行における直前の x 回 (現在 $x = 3$) のループにおいて制約違反を満たされなかった回数を示し，何度もタイミング制約を侵しているパスに含まれる枝ほど重みが大きくなる．$penalty(e_k)$ を用いることによって同じパスが何度も連続してタイミング制約違反を侵すことを防ぐことが可能である．そして，フェーズ2での枝 e_kの重み $w_2(e_k)$ を $w_2(e_k) = (penalty(e_k) + 1) \times vio(e_k) + 1$ とする．拡張 FM 法は，ノード v_iのゲインを $g(v_i)$ =(ノード v_i

```
手続き DFSIP
{
    V_A = V_B = ∅;
    ランダムに v_i ∈ R を1個選択する;
    V_A = {v_i};
    while(area(V_A) < α × area(V)){
        max_path_delay(r_i) が最大となる
            v_i を選択 (v_i ∈ V_A ∩ R);
        DFS(v_i) ;
    }
    V_B = V − V_A;
}

DFS(v_i){
    v_i を含むクラスタノードを C_l とする;
    V_A = V_A ∪ {v_k | v_k ∈ c_l};
    for each ({v_j|v_i に接続するノード })
        if (v_j ∉ V_A  &&  v_j ∉ V_R)
            DFS(v_j);
}
```

図 7 手続き DFSIP

を移動した時に減少するカットされる枝の重み $w_2(e_k)$ の総和) としている.

移動可能なノード v_iのうち, ゲイン $g(v_i)$ の値の最も大きいノードを選び, そのノードを移動してロックする. 移動可能ノードが存在しなければそのパスのうちで最も違反度の少ない解を最良解とし, 次のループ (FM 法の 1 パス) の初期解とする. 違反度を表す尺度として以下の *infeasibilty* を用いる.

$infeasibilty = \sum_{\forall e_i \in E} vio(e_i)$

違反度は直接に制約違反数を表していないため, 違反数の増加も予想されるが, 局所解を避けるため解の採用は違反度を選択の尺度とした.

以上の動作を解の改善が k回連続してみられなくなるまで行なう.

[PBVR(Path Based Violation Removal)]

次に, 違反のとれなかったパスについて強制的に改良を行う PBVR アルゴリズムについて説明する. 違反パスに含まれる全てのノードの集合を V'とする. また, V'に含まれているノードで, V_A, V_Bに含まれるノードの集合をそれぞれ, $V'_A(=V' \cap V_A)$, $V'_B(=V' \cap V_B)$ とする. 次に, V'_A, V'_Bを頂点とする誘導部分グラフをそれぞれ, $G_A = G(V'_A)$, $G_B = G(V'_B)$ と定義する. このとき, ブロック集合 Bの要素 b_kは, G_A, G_Bの連結成分で与えられる (図 9). つまり, カットラインで区切られる違反パスに含まれるノードの集合をブロック $b_k \in B$とする. ブロック b_kは, 移動させると違反パスについては必ずパスカット数が 1 または 2 減少するノードの集合を表している.

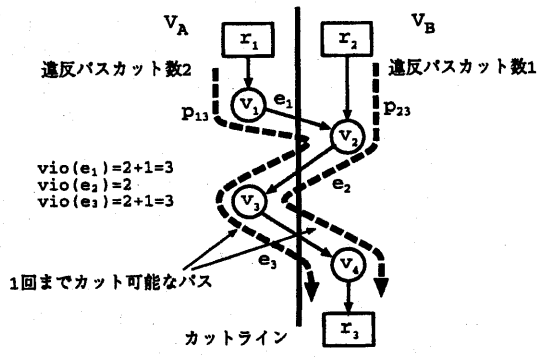

図 8 タイミング制約違反の度合 $vio(e_k)$ の計算

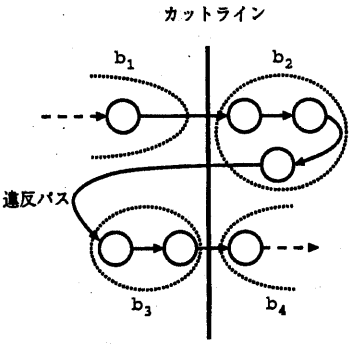

図 9 ブロック

次に, ブロック b_iを移動することにより減少する制約違反数を $dec\_vio(b_i)$ とする. 面積制約を満たし, ブロック b_iを移動させた時の違反パスの減少数 $dec\_vio(b_i)$ の最も大きいブロックを移動させ, 再びブロックの再構築を行う. 以上の動作を制約違反の改善が見られなくなるまで, 繰り返し行なう.

4 シミュレーション実験

提案手法を ULTRA COMP station model 170 上に C 言語を用いて実現し, シミュレーション実験を行なった. 実験で用いたデータは, ISCAS ベンチマークデータを SIS2.1 [15] を用いて論理合成したものを用いた (表 4). 面積制約$\alpha = 0.55$, 遅延制約 $D = 24$[ns] のもとで, カットされる枝の遅延 d_c=6, 8[ns] で実験を行なった. 提案手法は, データ No.1〜5 はクラスタノー

ド数を500，データNo.6～8はクラスタノード数を800とし実験を行なった．

表1 ISCASベンチマークデータ

No	データ名	ノード数	ネット数	レジスタ数	制約数	P_{ave}
1	C1	566	534	73	1353	283
2	C2	804	745	215	1133	30
3	C3	754	732	72	737	77
4	C5	1382	1259	301	2852	44
5	C6	1338	1215	301	2945	49
6	s35932	12193	11873	2083	6715	17
7	s38417	7706	7600	1569	28541	65
8	s38584	9306	9002	1730	17316	18

表2，表3に提案手法とFM法の比較結果を示す．FM法と提案手法に対して，それぞれ100個，10個の異なる初期解に対して分割を行い，平均のカット数($\#C_{ave}$)，平均の制約違反パス数($\#V_{ave}$)，制約違反の最も少ないときの最小カット数($\#C_{best}$)，また，そのときの制約違反数($\#V_{best}$)，平均の計算時間(T_{ave}[sec])について比較を行なった．FM法は，非常に高速であるため実行回数を増やすことによって，制約違反の少ない解を求めることが可能である．しかし，制約を満たす解はごく稀でほとんどの場合が制約を満たすことができなかった．これに対して，提案手法は全体的に制約違反数が少なく，制約を満たす解を得る可能性はFM法と比べて非常に高い．

表2 FM法との比較1($dc = 6$[ns])

No	手法	$\#C_{ave}$	$\#V_{ave}$	$\#C_{best}$	$\#V_{best}$	T_{ave}
1	FM	45	366	22	24	1
	提案	49	24	49	0	416
2	FM	45	6	21	0	0
	提案	52	0	33	0	28
3	FM	74	74	54	1	1
	提案	78	1	68	0	182
4	FM	59	3	44	0	1
	提案	84	6	50	0	71
5	FM	59	4	43	0	1
	提案	90	3	47	0	172
6	FM	110	0	59	0	51
	提案	86	0	60	0	288
7	FM	254	557	167	0	27
	提案	130	0	75	0	603
8	FM	152	53	78	0	39
	提案	191	0	116	0	414

次に，表4は提案手法の初期解生成アルゴリズムの有効性を確認するために，初期解をランダムに生成したもの(RND)と，DFSIP(DFS)を用いたて生成した場合とで比較を行なった実験結果である．表中の–は計算時間が非常にかかり，計測出来なかったことを意味する．また，データNo.6～No.8のRNDについては，現在，実験中である．DFSIPを用いることにより，計算時間の大幅な短縮がみられた．また，カット数と違反数においてもDFSIPが優れいることから，DFSIPは非常に有効であるといえる．

表3 FM法との比較2($dc = 8$[ns])

No	手法	$\#C_{ave}$	$\#V_{ave}$	$\#C_{best}$	$\#V_{best}$	T_{ave}
1	FM	45	482	20	64	1
	提案	53	47	49	0	610
2	FM	45	40	29	0	0
	提案	49	1	33	0	43
3	FM	74	131	55	9	1
	提案	89	3	77	0	668
4	FM	59	11	44	0	1
	提案	107	21	49	0	189
5	FM	59	31	54	0	1
	提案	108	3	40	0	459
6	FM	110	0	59	0	50
	提案	77	0	60	0	323
7	FM	254	826	171	1	28
	提案	173	0	87	0	1020
8	FM	152	170	85	2	39
	提案	396	53	358	3	1290

表4 RNDIPとDFSIPの比較

No	手法	$\#C_{ave}$	$\#V_{ave}$	$\#C_{best}$	$\#V_{best}$	T_{ave}
1	RND	97	550	76	370	–
	DFS	53	47	49	0	610
2	RND	133	11	93	2	145
	DFS	49	1	33	0	43
3	RND	222	198	212	95	1078
	DFS	89	3	77	0	668
4	RND	212	27	197	0	674
	DFS	107	21	49	0	189
5	RND	257	91	174	4	624
	DFS	68	4	50	0	234
6	RND					
	DFS	77	0	60	0	323
7	RND					
	DFS	173	0	87	0	1020
8	RND					
	DFS	396	53	358	3	1290

表5は，PBVRによるカット数と違反数の平均の改善率(%)を示す．プラスの値はカット数，違反数の減少を表し，マイナスの値は増加を表す．実験結果より，PBVRが違反除去に非常に効果的であることが分かる．

5 あとがき

本稿では，タイミング制約を考慮した回路分割手法を提案した．提案手法は，タイミング制約を考慮したクラスタリングとタイミング制約を考慮したゲインに基づく拡張FM法から構成されている．また，シミュ

表 5 PBVR の効果

No	$\#C_{ave}$の改善率 (%)	$\#V_{ave}$の改善率 (%)
1	−3.9	45
2	−4.1	86
3	−3.5	68
4	−9.1	64
5	−5.9	55
6	0	0
7	−7.5	14
8	−5.1	19

レーション実験により有効性を示した．本稿では，タイミング制約を取り除くことのみに重点をおいてきたが，タイミング制約を満たしたうえで，さらにカット数の少ない解を得るよう改良することが今後の課題である．

参考文献

[1] C. J. Alpert and A. B. Kahng: "Recent direction in netlist partitioning: A survey," INTEGRATION, the VLSI journal, Vol.1-2, No.19, pp. 1–81 (1996).

[2] C. J. Alpert and S.-Z. Yao: "Spectral partitioning: The more eigenvectors, the better," Proc. 32nd Design Automation Conference, pp. 1–93 (1995).

[3] J. Cong, H. P. Li, S. K. Lim, T. Shibuya and D. Xu: "Large scale circuit partitioning with loose/stable net removal and signal flow based clustering," Proc. IEEE/ACM International Conference on CAD (1997).

[4] S. Dutt and W. Deng: "A probability-based approach to VLSI circuit partitioning," Proc. 33rd Design Automation Conference, pp. 100–105 (1996).

[5] S. Dutt and W. Deng: "VLSI circuit partitioning by cluster-removal using iterative improvement, " Proc. IEEE/ACM International Conference on CAD, pp. 194–200 (1996).

[6] C. M. Fiduccia and R. M. Mattheyses: "A linear-time heuristic for improving network partitions," Proc. 19th Design Automation Conference, pp. 175–181 (1982).

[7] S. Hiratani, S. Wakabayashi and T. Koide: "A hypergraph partitioning algorithm considering path-cut constraints for circuit partitioning," in Paper of Technical Group. IEICE, VLD96-555, pp. 49–56 (1997), in Japanese.

[8] F. M. Johannes: "Partitioning of VLSI circuits and systems," Proc. 33rd Design Automation Conference, pp. 83–86 (1996).

[9] G. Karypis, R. Aggarwal, V. Kumar and S. Shekhar: "Multilevel hypergraph partitioning: Application in VLSI domain," Proc. 34th Design Automation Conference, pp. 526–529 (1997).

[10] B. Kernighan and S. Lin: "An efficient heuristic procedure of partitioning of electrical circuts," Bell System Technical Journal, Vol.49, No.2, pp. 291–307 (1970).

[11] T. Koide, M. Ono, Y. Nishimaru, S. Wakabayashi and N. Yoshida: "A new performance driven placement method with the Elmore delay model for row based VLSIs," Proc. Asia and South Pacific Design Automation Conference 95, pp. 405–412 (1995).

[12] R. Murgai, R. K. Brayton and A. L. Sangiovanni-Vincentelli: "On clustering for minimum delay/area," Proc. IEEE/ACM International Conference on CAD, pp. 6–9 (1991).

[13] 大平, 南, 小出, 若林："非線形計画法に基づくタイミングドリブン MCM システム分割手法," 第 9 回回路とシステム軽井沢ワークショップ, pp. 241–247 (1996).

[14] B. Saucier, D. Brasen and J. P. Hiol: "Partitioning with cone structures," Proc. IEEE/ACM International Conference on CAD, pp. 236–239 (1993).

[15] E. M. Sentovich, et al.: "SIS: A system for sequential circuit synthesis," Tech. Electronics Research Laboratory (1992).

[16] M. Shih, E. S. Kuh and R.-S. Tsay: "Timing-driven system partitioning by constraints decoupling method," Proc. of IEEE Multi-Chip Module Conference, pp. 164–169 (1993).

[17] H. Yang and D. F. Wong: "Circuit clustering for delay minimization under area and pin constraints," Proc. European Design & Test Conference 95, pp. 65–70 (1995).