

GA/CBIC自動レイアウトシステムGALETにおける
タイミングドリブン自動配置手法

多和田 茂芳 水牧 俊博† 田形 充†
袖 美樹子* 岩元 圭一郎** 水沼 貞幸†

日本電気(株) 〒183 東京都府中市日新町 1-10
* 〒216 神奈川県川崎市宮前区宮崎 4丁目 1-1
** 〒211 神奈川県川崎市中原区下沼部 1753

†NEC ソフトウェア北陸(株) 〒920-21 石川県石川郡鶴来町安養寺 1 番地

あらまし

ゲートアレイ/セルベースICの自動レイアウトシステムGALETの自動配置手法について、特にタイミングドリブン配置機能を中心に報告する。階層クラスタリングによるミニカット処理、2次元改良処理の中で、与えられたパス遅延制約を満足するようにパス遅延を検証しながら配線性の改善とパス遅延の改善を同時に行うアルゴリズムを紹介する。2次元改良ではパスグループという手法を用いて強力なパス遅延改善機能を実現した。高性能なサブミクロンLSIの設計に適用した実験結果から、タイミング考慮なしの配置結果に対して総配線長や配線混雑度の増大による配線性を悪化させることなくパス遅延制約を満足する結果を得ることができ、本手法の有効性が確認できた。

和文キーワード 自動配置、タイミングドリブン、タイミング保証、パスグループ、パス遅延制約

Timing driven placement algorithm of the GA/CBIC layout system GALET
using path grouping technique

Shigeyoshi Tawada Toshihiro Mizumaki† Mitsuru Tagata†
Mikiko Sode* Keiichiro Iwamoto** Sadayuki Mizunuma†

NEC Corporation 1-10, Nisshincho, Fuchu, Tokyo, 183 Japan

*1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa, 216 Japan

**1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211 Japan

†NEC Software Hokuriku, Ltd 1, Anyoji, Tsurugimachi, Isikawa-gun, Ishikawa-ken, 920-21 Japan

Abstract

This paper presents an automatic placement method of the gate array / cell base IC automatic layout system GALET, emphasizing its timing driven placement algorithm. In order to improve path delay and routing congestion simultaneously, our proposed algorithm has clustering-based mincut placement and 2-dimensional improvement method with timing assurance technique, called "path grouping". Experimental results on submicron high performance chips show that our algorithm is able to find a solution which satisfies the path delay constraints without degrading routing length and congestion.

key words automatic placement, timing driven, timing assurance, path grouping, path delay constraint

1 はじめに

回路の信号遅延は論理素子の内部の遅延（ゲート遅延）と論理素子間の配線によって生じる配線遅延とから成る。LSIの製造プロセスの微細化に伴い、設計回路が大規模化し信号遅延に占める配線遅延の割合が相対的に増大してきており、高速/高性能なLSIを設計する上でレイアウト設計における配線遅延の最適化の重要度が高まっている。

そのようなニーズに基づいて配線遅延をコントロールするための自動配置での種々の方法が提案されている。予測線長に基づく遅延解析の結果からクリティカルパス上のネットに重み付けして配置する方法 [1] は、重み値と配置結果の配線長との間の関係が不明瞭なためその重み設定が難しく更に結果についての保証が無いので重みを変えながらの繰り返し実行になる。パスの遅延時間制限を満足できるようにネットに配線長制限を与えそれを満足するように配置する方法 [2] は、パスとしての遅延時間制限をネット毎の配線長制限として割り振るので配置改良の自由度が低下し配置のもう一つの目的関数である配線収容性改善の効果を必要以上に妨げることになる。パスとしての遅延制限値を扱い配置の改良毎に遅延解析結果を更新しながらタイミングを保証していく方法 [3, 4] は、一般的に処理時間がかかるが遅延時間制限を満足して更に配線収容性の高い良質な配置結果を得ることができる。

本論文で提案する配置手法は、上述の3番目の方法の一つで、パスとしての遅延時間制限を扱い配置改良時にインクリメンタルに遅延解析を行って遅延保証を行う。配置処理中のミニカット、2次元改良という二つのフェーズでタイミングを考慮しそれぞれに高速化を図って実用化している。

2 タイミング制約とパストレース構造

本論文で対象とするタイミング制約はパスの最大遅延時間の制限として与えられ信号のRISE/FALLを考慮することができる。FF-FF間、チップ入力端子-FF間、FF-チップ出力端子間のパスについて個別に、あるいは一括してタイミング制約を与えることができる。指定についてはパス上の全経由点を記述する指定の他にパスの始終点指定（さらに特定の経由点の指定、特定の経由点不通過の指定も可能）のみで与えることも可能である。後者の指定を行った場合には始終点間のパストレースを行い始終点間の全経路についてタイミング制約を考慮できる。

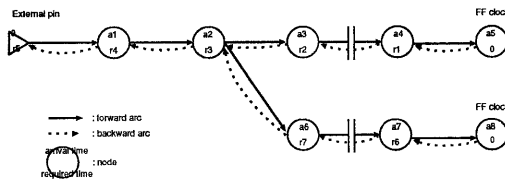


図 1: パストレース構造

タイミング制約入力後に、自動配置中にインクリ

メンタルに遅延解析を行うためのパストレース構造（図1）を生成する。パストレース構造はパス上端子（以後ノード）と信号伝搬方向を考慮したそれら端子間の接続（以後アーク）を表現した構造である。ノードは、その前段の全FFクロック入力端子あるいはチップ入力端子からのワーストのアライバルタイム（a）と、後段の全FFクロック入力端子あるいはチップ出力端子からのワーストのリクワイアードタイム（r）をRISE/FALL別に保持している。アークは、前進と後退の両方向存在し、素子遅延時間と配置の進行につれて変化する予測線長に基づく配線遅延時間を保持する。素子の配置の変化に伴ってそれに接続する配線の長さ（形状）が変化すると配線遅延時間が更新されるその遅延時間差分を伝搬してアライバルタイム、リクワイアードタイムがインクリメンタルに更新される。配線遅延は、最初はネットのファンアウト数やネットの属するグループ領域のサイズや位置から見積もられる予測線長によって計算され、配置の進行に伴って素子の配置位置の詳細化が進むにつれて随時更新されていく。

3 タイミング保証自動配置手法

3.1 基本アルゴリズム

GALETの配置の基本アルゴリズムは既に報告済みの「階層クラスタリング法を用いたミニカット手法」[5]を用いている。このアルゴリズムはミニカットアルゴリズムの改良版で、素子の接続関係に基づいて階層的にクラスタを生成し、クラスタの階層を崩しながら各階層レベルでカットラインを跨いだクラスタの交換を行う（図2）。素子の交換はこのクラスタ同士の交換によって行われ、交換個数の削減による高速化と各階層レベルでのクラスタサイズの均一化による交換容易化を図っている。

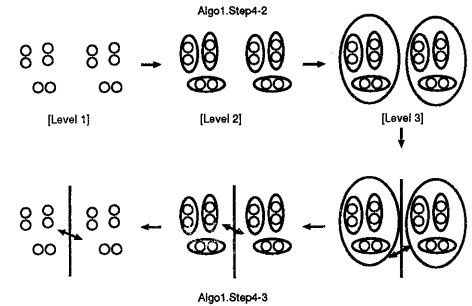


図 2: 階層クラスタリングとミニカット

1次元交換によるミニカットの配置改良はカットラインの生成順序に対する解の依存性が高い。そこで、配線性を考慮してカットラインを横切る配線数を2次元的に平坦化して解を補正するために2次元改良配置によるクラスタの多段反復交換処理を行う（図3）。2次元改良配置においても高速化、交換容易化のために階層クラスタリング法を用いている。2次元改良配置はミニカットの任意の時点で実行可能だが処理時間と効果を考慮して4×4分割、16

×16分割・・・等の全カット領域サイズがそろって縦横の領域数が同等になるタイミングで行っている。

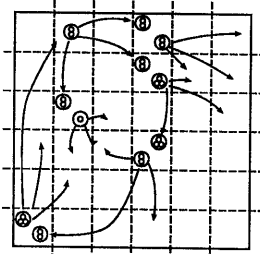


図3: 2次元改良

ミニカット、2次元改良により配置改良が行われカット領域に割り当てられた素子について、下地の制約を守って各領域内での詳細な配置位置、配置方向をマッピング処理(図4)により決定する。マッピング処理では、2次計画法の手法(SCG法)を用いている。

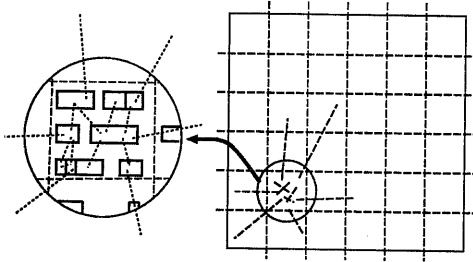


図4: マッピング

3.2 ミニカットにおけるタイミング保証

ミニカット配置でのタイミング保証は[6]で既に報告しているアルゴリズムを用いている。カットライン生成毎に素子が属するカット領域の位置からネットの配線形状を見積り配線遅延を再計算して遅延解析結果を更新する。次にスラック値[7] ($s = r - a$)を求め、スラックが負であるネットについて以下のような考慮を行う。

- 階層クラスタリング時にスラックが負であるネットにつながる素子を優先的にクラスタリングする。(Algo1.Step4-2)
- ペア交換時にスラックが負であるネットの配線長が短くなるような交換を優先する(図5)。(Algo1.Step4-3-1)

ミニカットでのタイミング保証ではスラックをネットに分配しそれを考慮してペア交換を行うので、遅延解析更新回数はカットライン数分であり高速性を大きく損なわない。

Algo1. タイミング保証ミニカットアルゴリズム

- Step1. バストレー構造を生成する。
- Step2. 初期遅延解析を行いスラック値を求める。
- Step3. 初期クラスタリングを行う。
- Step4. for (各カットラインに対して) |
 - Step4-1. 遅延解析を行いスラック値を更新する。
 - Step4-2. ネットスラックを考慮して階層クラスタリングを行う。
 - Step4-3. for (クラスタの階層を崩しながら) |
 - Step4-3-1. ネットスラックを考慮してクラスタのペア交換を行う。
 - Step4-4. if (特定カット数) 2次元改良を行う。
- Step5. 下地へのマッピングを行う。

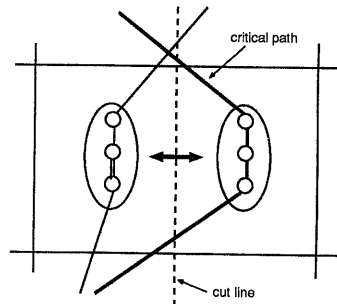


図5: スラックを考慮したペア交換

3.3 2次元改良におけるタイミング保証

2次元改良(Algo1.Step4-4)でのタイミング保証は遅延時間制限のある全てのバス(クリティカルバス、場合によっては全バス)の遅延時間制限を網羅的に保証することができる。さらに、バス遅延改善による配線性の悪化を最小限にとどめることができる。

以下にアルゴリズムの詳細についての説明を行う。ミニカット処理により各素子とその時点でのカット領域のどこかに割り当てられた結果が2次元改良に渡されると、Algo2.Step1により各ネットを構成する素子の属するカット領域の位置関係を考慮してネットの配線形状を見積り配線遅延時間を計算して遅延解析結果が更新される。

Algo2.Step2-1により全カット領域の素子について階層クラスタリングを行い、以後クラスタの階層を崩しながらクラスタ単位に改良処理を行う(Algo2.Step2-2)。

Algo2.Step2-2-1ではバス遅延エラーを改善するためのバsgループとネットグループを生成する。バsgグループとは遅延エラーを起こしているバスすなわちスラックが負となるバス上の連続したクラスタの集合(図6)で、バス始点(FFあるいはチップ入力端子)一終点(終点はFFあるいはチップ出力端子)間の負スラックをバsgトレースしながら求めていく。バsgグループは1つのエラーバス(負スラック

Algo2. タイミング保証2次元改良アルゴリズム

- Step1. 遅延解析結果を更新する。
 Step2. for (パラメータを変えた改良条件数分) |
 Step2-1. 全カット領域で階層クラスタリングを行う。
 Step2-2. for (クラスタの階層を崩しながら) |
 Step2-2-1. パスグループ、ネットグループを作成する。
 Step2-2-2. クラスタ、グループの仮想ゲインを算出し、降順ソートされたバケットにセットする。
 Step2-2-3. for (バケット内のクラスタ、グループの最大ゲインが規定値以下になるまで) |
 Step2-2-3-1. バケットからクラスタまたはグループを取り出す。
 Step2-2-3-2. グループの場合仮移動してクラスタ(グループクラスタ)にする。
 Step2-2-3-3. クラスタの多段反復移動の最適パターンを探索する。
 Step2-2-3-4. 最適移動パターンの移動処理を行い、同時に遅延解析結果をインクリメンタルに更新する。
 Step2-2-3-5. クラスタの移動結果に基づいてバケット内仮想ゲインを更新する。
 Step2-2-3-6. グループクラスタの場合、クラスタを解体してグループに戻す。

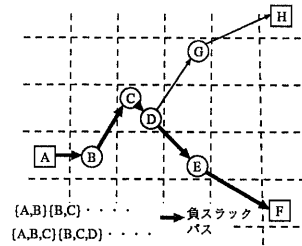


図 6: パスグループ

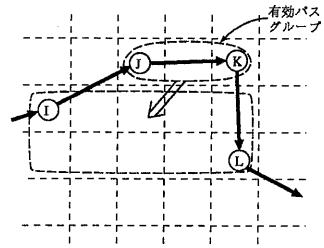


図 7: 有効パスグループ

クパス) 上のクラスタが n 個ある場合最悪 $n(n-1)/2$ 個存在することになるが改善効果のあるパスグループのみを有効パスグループとして抽出する。改善効果については、図 7 に示すようにそのパスグループを通るエラーパスの前段と次段のクラスタが囲む矩形領域外にそのパスグループが存在するとき限り有効とする。また、必要以上に大きなパスグループが生成されないように、パスグループのサイズ制約としてその時点でのカット領域の許容サイズ(例えばカット領域サイズに全体の面積使用率をかけたもの)を設定しそれを越えないグループのみを有効にする。次に、ネットグループとはスラックが負となるパス上ネットにつながるクラスタの集合でパスグループと同様のサイズ制約を満たして有効となる。ネットグループはエラーパス上に無いクラスタを移動することでネット配線長を小さくして配線遅延を改善する効果を得る目的で生成する(図 8)。

Algo2.Step2-2-2 ではクラスタ、パスグループ、ネットグループの仮想ゲインを計算し、それをキーにクラスタおよびグループを降順にソートしたバケットを作成する。仮想ゲインは以下の式で求められ、グループについてはグループ内のクラスタのゲインの和で求められる。移動に関しては各々のクラスタの最適な移動が行われた場合を想定してゲインは計算される。

$$\alpha \cdot \Delta Ct + \beta \cdot \Delta Sz + \gamma \cdot S1$$

(α, β, γ は係数)

ΔCt : クラスタの移動によるカット数の改善度

ΔSz : クラスタの移動によるカット領域の許容サイズオーバーの改善度

$S1$: 最悪スラック値

ΔSz (サイズ制約ゲイン) はクラスタが属するカット領域が許容サイズオーバーを起こしているときのそのクラスタが移動することによる改善効果を数値化したものである。 $S1$ (最悪スラック) の係数 γ はスラックが正のときは 0 でスラックが負のときは他のゲインの項に比べて大きめに設定される(図 9)。この最悪スラックの項により遅延エラーパス上のクラスタやグループがバケットの先頭の方に現れ早い段階で改良が行われる。

作成されたバケットの先頭から順にクラスタあるいはグループを取り出して多段反復改良配置を行う。このとき、改善効果の低いクラスタやグループの処理を行わないために規定値(ゲインの最低基準値)を終了条件に加えている。バケットは配置改良に従って効率的に順序更新が行われる。

クラスタはある 1 つのカット領域に存在するが、

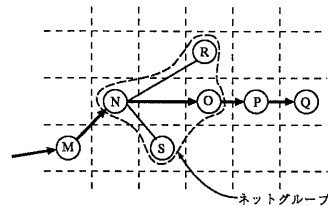


図 8: ネットグループ

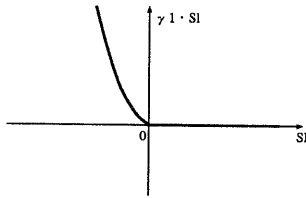


図 9: 最悪スラックの係数

グループは複数のクラスタの集合体なのでそれぞれに別のカット領域に存在している場合がある。そこでグループがバケットから改良対象として選ばれた場合には Algo2.Step2-2-3-2 によりグループ内のクラスタを 1 つのカット領域に仮移動して結合 (クラスタ化) し通常の 1 つのクラスタと同様にして改良処理を行う。仮移動の際には最もサイズに余裕のあるカット領域が選択され、仮移動後にゲインは再計算される。また、グループから作られたクラスタは最適移動が行われた後で Algo2.Step2-2-3-6 により再び解体されグループに戻る。

Algo2.Step2-2-3-3 のクラスタの最適移動パターンの探索は再帰的に行われ、再帰呼出の深さ (回数) と移動候補領域の選択数を制限して処理時間の増大を抑える。

Algo3. クラスタの最適移動パターン探索アルゴリズム

```

Step1. クラスタを受け取る。
Step2. 移動のゲインの大きい移動候補領域を規定数選択する。
Step3. for (移動候補領域規定数分) {
  Step3-1. クラスタを移動し遅延解析結果をインクリメンタルに更新する。
  Step3-2. 移動のゲインを累計する。
  Step3-3. if (累計ゲインが登録されている移動パターンの累計ゲインよりも大きい) {
    Step3-3-1. そのパターンを登録する。
  }
  Step3-4. if (再帰呼びだしの深さ制限以内) {
    Step3-4-1. 移動先の最大ゲインのクラスタを規定数選択する。
    Step3-4-2. for (クラスタの規定数分) {
      Step3-4-2-1. if (クラスタのゲインが最低基準値以上) {
        Step3-4-2-1-1. 選択されたクラスタについて
          Step1. を再帰的に呼び出す。
      }
    }
  }
}
Step3-5. クラスタを元の移動前の領域へ戻し、遅延解析結果も再更新する。

```

Algo3.Step2 で選択されるクラスタの移動候補領域は、普通のクラスタの場合にはそれにつながる全てのネットを囲む矩形領域となる。グループの場合はその前段と次段のクラスタを囲む矩形領域 (図 7)、すなわちバス遅延時間の改善が見込まれる領域となる。クラスタの移動候補領域の中でクラスタ移動ゲ

インの大きい移動候補領域が規定数 (4 つ程度) 選択され再帰的に最適な多段反復改良配置移動パターンの探索が行われる。クラスタの移動候補領域を決定するためのクラスタ移動ゲインは、以下のように求められる。

$$\alpha \cdot \Delta C t + \beta \cdot \Delta S z + \gamma 2 \cdot \Delta S 1$$

(α 、 β 、 $\gamma 2$ は係数)

$\Delta C t$: クラスタの移動によるカット数の改善度
 $\Delta S z$: クラスタの移動によるカット領域の許容サイズオーバーの改善度
 $\Delta S 1$: スラックの改善度

$\Delta S 1$ (スラックゲイン) は、移動前の最悪スラック値を $S 11$ 、移動後の最悪スラック値を $S 12$ とすると、以下のように計算される。

$$\begin{aligned} S 11, S 12 \geq 0 \text{ のとき、} & 0 \\ S 11 \geq 0, S 12 < 0 \text{ のとき、} & -(S 12)^2 \\ S 11 < 0, S 12 \geq 0 \text{ のとき、} & S 11^2 \\ S 11, S 12 < 0 \text{ のとき、} & S 11^2 - S 12^2 \end{aligned}$$

スラックゲインの係数 $\gamma 2$ を正の大きな値に設定することにより、スラックが悪化 (例えば正から負へ変化) するような移動はゲインが小さくなるため選択されない。

前述したバスグループやネットグループはサイズが揃えられていたクラスタを結合するのであるから他のクラスタに比べて当然サイズが大きくなり動きにくくなるように見えるが、スラックゲインがその係数 $\gamma 2$ によって他のゲイン ($\Delta C t$ 、 $\Delta S z$) より大きく設定されているためその改善効果により移動先のカット領域でのサイズ超過によるゲインの低下を十分に吸収できる程大きな移動ゲインが得られ、移動が行われる。移動後はその領域内でのサイズ制約ゲインが大きくなり領域から出やすくなるためグループ内の個々のクラスタがバケットで選ばれたときに遅延エラーを悪化させない (スラックが悪化しない) ようなカット領域に追い出されていくことになり、カット領域サイズのバランスが改善される。

最適移動パターンでの多段反復移動をバケットに現れるクラスタやグループについて繰り返して配線性とバス遅延エラーが改善されていく。

2 次元改良でのタイミング保証ではバスとしての遅延エラーの改善を優先するが同時に 2 次元改良での本来の機能であるカット数平坦化による配線性の改善も行うので、配線設計における迂回配線による遅延エラーの発生を抑制できる。

4 実験結果

本論文で提案したタイミング保証配置についての実データによる性能評価結果を以下に示す。

表 1 は、下地ゲート数 2 0 2 K G、素子数 1 2 8 5 2、ゲート使用率 4 9. 5 % のチップデータで、制約バス遅延値は 1 6 n s、クリティカルパス 9 2 8 本についてバス遅延制約を与えた結果である。タイミング保証無しの場合の配置結果 (NoTD) に対して、タイミング保証を使用した結果 (2DTD) ではタイミング違反はほとんど無くなり、カット数、総配線長に関しても同等であるが、配置時間は 2.4 倍に増加している。2DTD において残った 4 つの違反は、配置

プログラム中の予測配線に基づく遅延検証結果ではエラーとはなっていないが、配線遅延の見積り誤差のため詳細配線後の遅延解析結果ではエラーとなった。これについては、高ドライブ素子への置き換えと設計変更処理によるインクリメンタルレイアウトで対応した。

表 1: チップ評価結果

	NoTD	2DTD
違反パス	888	4
最大遅延	19.42ns	16.09ns
最大Vカット数	960	928
最大Hカット数	862	882
総配線長	18.555m	18.429m
CPU 時間	34m	1h22m

表 2、表 3 は高性能なサブマイクロン L S I を階層的にレイアウトする設計法において、マクロレベルの設計に適用した例である。この場合、クリティカルパスが多いため全パスにパス遅延制約を与えている。表 2 のデータは素子数 1 2 4 9 2、F F 間の総パス数は 3 8 1 2 3 で、そのすべてのパスに 1 0 0 timeunit (timeunit は時間の単位とする) の制限値を与えた結果である。NoTD で 1 1 9 8 0 パスあったエラーパスは 2DTD より 4 2 に減少した。最大遅延パスも 1 4 5 timeunit から 1 0 4 timeunit となり 2 8 % 改善された。このデータについても配置プログラム中の予測配線に基づく遅延検証結果ではエラーであったが、マクロレイアウトエリアの設定が小さかったため配線迂回が起これ詳細配線後の遅延解析結果では 4 2 エラーとなった。これについては、優先配線によりパス上ネットの配線迂回を抑えることによって最終的にはエラー無しとなった。表 3 のデータは素子数 1 4 9 6、F F 間総パス数 6 9 4 6 で、表 2 のデータと同様に全パスに 1 0 0 timeunit の制限値を与えた結果である。NoTD で 1 1 9 あったエラーパスは 2DTD より 3 9 パスに減少した。最大遅延パスは 1 1 4 timeunit から 1 0 8 timeunit と 5 % 改善された。このデータについても配置プログラム中の予測配線に基づく遅延検証結果ではエラーであった。これについても、パス上ネットの優先配線により最終的にはエラー無しとなった。

表 2: マクロ評価結果 (1)

	NoTD	2DTD
違反パス	11980	42
最大遅延	145timeunit	104timeunit
最大Vカット数	989	833
最大Hカット数	703	631
総配線長	4.344m	4.105m
CPU 時間	16m	5h38m

表 3: マクロ評価結果 (2)

	NoTD	2DTD
違反パス	119	39
最大遅延	114timeunit	108timeunit
最大Vカット数	185	157
最大Hカット数	190	191
総配線長	0.297m	0.280m
CPU 時間	3m	16m

5 まとめ

G A L E T のタイミング保証自動配置についてアルゴリズムを紹介し、実験結果をもってその有効性を示した。

本論文で述べたアルゴリズムのタイミング改善機能は強力で、遅延制約の厳しい高性能なサブマイクロン L S I のレイアウト設計においても十分に効果を発揮している。しかし、処理時間の点では現状まだ満足いく結果ではない。そこで、今後は特にタイミング保証 2 次元改良配置の更なる高速化改善を行っていき、大規模で遅延制約の厳しい L S I についても一括レイアウトして全パス遅延保証が行えるようにしていきたい。また、配線迂回等による配線遅延見積り誤差による遅延エラーパスの発生を無くすように、優先配線機能を用いた配線プログラムとのリンク機能や配線性を保証した配置機能を実現していきたい。

6 謝辞

本発表の機会を与えてくださった第一コンピュータ事業本部コンピュータ事業部 C A D 技術部の野村部長、尾藤課長ならびに L S I 技術部の矢部課長に感謝いたします。また、ご助言をいただいた C & C 研究所吉村 C A D 担当部長ならびに L S I 事業本部 U L S I システム開発研究所 C A D 基礎開発部の石塚主任にも感謝いたします。

参考文献

- [1] A.E.Dunlop, V.D.Agrawal, D.N.Deutsch, M.F.Jukl, P.Kozak, M.Wiesel, "CHIP LAYOUT OPTIMIZATION USING CRITICAL PATH WEIGHTING," *21st DAC*, pp.133-136, 1984
- [2] M.Terai, K.Takahashi, K.Sato, "A New Mincut Placement Algorithm for Timing Assurance Layout Design Meeting Net Length Constraint," *27th DAC*, pp.96-101, 1990.
- [3] Michael A., B.Jackson, E.S.Kuh, "Performance-Driven Placement of Cell Based IC's," *26th DAC*, pp.370-375, 1989.
- [4] W.E.Donath, R.J.Norman, B.K.Agrawal, A.E.Bello, S.Y.Han, J.M.Kurtzberg, P.Lowy, R.I.McMillan, "TIMING DRIVEN PLACEMENT USING COMPLETE PATH DELAY," *27th DAC*, pp.84-89, 1990.
- [5] 枝廣, 吉村, "階層クラスタリング法を用いたセル列型 LSI のための配置手法," *VLD90-62*, pp.9-15, 1990.
- [6] M. S. Tanaka, Y. Miyazawa, H. Aizawa and M. Minowa, "Minimum Delay Placement with Influence of Nets and Hierarchical Clustering," *Proc. CICC*, pp663-666, 1994.
- [7] P.S.Hauge, R.Nair, J.Yoffa, "Circuit Placement for Predictable Performance," *Proc. of ICCAD*, pp.88-91, 1987.