

## レジスタに着目したタイミング制約を考慮するセル配置手法

佐藤秀則 河合健治

NTT LSI 研究所

神奈川県厚木市森の里若宮 3-1

LSI の性能を表すタイミング制約は、フリップフロップの位置に大きく左右される。フリップフロップは、通常レジスタとして固まりで扱うことができる。このことから我々は、クリティカルパス遅延制約を考慮するセル配置手法として、レジスタとその他のセルの配置を独立に行なう手法を考案した。レジスタの配置には、クリティカルパスを考慮しなければならないレジスタ間の接続関係を表す、レジスタ間ネットを用いる。実験の結果、レジスタを考慮しながら手で行なった配置には及ばなかったものの、従来レジスタを考慮しない配置手法よりも、総配線長で良い結果が得られた。

## A New Placement Algorithm Considering Register Position

Hidenori SATO and Kenji KAWAI

NTT LSI Laboratories

3-1, Morinosato Wakamiya, Atsugi-Shi, Kanagawa Pref.

A timing constraint which represents performance of LSI depends on the positions of Flip Flops. We usually treat Flip Flops as a register. In this paper, we present a new placement algorithm in which register placement and cell placement are performed mutually. To calculate the register position, we use the connectivities of registers which are defined by the timing constraint of the critical path. The experimental result using this algorithm was better than that of using conventional method concerning total net length.

## 1 はじめに

近年、LSIの進歩は著しく、ますます微細化、高速化、大規模化してきている。それに伴い、遅延時間の配線に依存する部分の割合が大きくなってきており、いわゆるタイミング制約に関する問題が生じてきている。そのような中で、このタイミング制約に関する問題を考慮できる設計CADツールの開発が必須となってきている。

タイミング制約を考慮するセル配置問題は、従来からの総配線長の最小化に主眼を置いている net-oriented な問題とは違い、入力から出力までの信号の流れの中で、その信号の遅延時間に関する制約を満たさなければならないという、いわゆる path-oriented な問題である。

これに関する多くの研究が今までなされている。そのアプローチの仕方を大きく分類すると、(1) path-oriented な問題を、適当に net-oriented な問題に直して、従来手法に近い形の問題として解く手法 [1, 2, 3]、(2) path-oriented な問題のまま、バスに対する制約として扱う手法 [4, 5]、に分けることができる。

[1, 2] では、タイミング制約から与えられる各ネット長に上限値を設けその範囲を越えないようにセルの配置を行っている。[5] では、バス単位のタイミング制約まで含めた配置問題を線形問題として表し、それを解くことによってセルの配置を行っている。

一般に、タイミング制約は、クリティカルパス遅延値とクロック・スキューとして与えられる。これらはそれぞれ、フリップフロップ間の信号伝搬時間の最大値、各フリップフロップへのクロック信号到達時間の最大、最小の差である。すなわちタイミング制約を考慮するには、フリップフロップの位置の考慮が重要であるといえる。このフリップフロップの配置に着目するセル配置手法に関する研究結果も報告されている [6, 3]。

論理設計の段階で、フリップフロップはその機能毎にレジスタとして固まりで扱われる (図 1)。このことから、我々はクリティカルパスを考慮するセル配置手法として、レジスタに着目した配置を行なうことにした。そのため、セル同士の接続を表すネットに加え、クリティカルパスを考慮しなければならないレジスタ同士の接続を表すネットを用いる配置を行なうことにした。これらふたつのネットリストに関する線形問題の融合として、レジスタとその他のセルの位置を計算する配置アルゴリズムを考案し、配置実験を行なったので、その結果について報告する。

フリップフロップをレジスタとしてまとめて扱い、配置を行なうことによって、

1. クラスタリング効果による配置対象セル数の減少
2. 同じ機能をもつフリップフロップを近くに置くことによるクロック・スキューの減少。
3. クリティカルパスを考慮するレジスタ間同士の接続関係を陽に考慮することによる、信号の流れに従った配置結果。

が、期待できる。

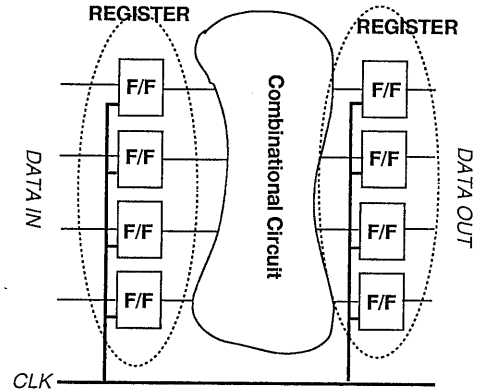


図 1: レジスタと組合せ回路

## 2 レジスタ間の接続関係

### 2.1 遅延時間の計算

時間  $t_s$  に、あるセルに入力した信号が次のセルに到達する時間  $t_a$  は、次の式で計算することができる。

$$t_a = t_s + t_{cell} + t_{f/o} + t_w \quad (1)$$

$t_{cell}$  : セルの固有遅延  
 $t_{f/o}$  : ファンアウトによる付加遅延  
 $t_w$  : 配線遅延

$t_{cell}$  と  $t_{f/o}$  は、ネットリストから計算でき、 $t_w$  は単位配線長あたりの遅延時間に配線長をかけた時間で計算される。結局、 $t_a$  はレイアウト結果に依存し、次式で計算される。

$$t_a = t_s + t_{fix} + l \cdot t_u \quad (2)$$

$t_{fix}$  : ネットリストから計算される遅延時間  
 (=  $t_{cell} + t_{f/o}$ )  
 $l$  : 配線長  
 $t_u$  : 単位配線長あたりの遅延時間

ここから、あるレジスタ間バスの遅延時間  $t_{path}$  は、 $n$  をそのバス上のネットとして、

$$t_{path} = \sum_{n \in path} (l_n \cdot t_{nu}) + Const. \quad (3)$$

$l_n$  : ネット  $n$  の長さ  
 $t_{nu}$  : ネット  $n$  の単位長さあたりの遅延時間

で計算される。定数 Const. はバス上の全セルの  $t_{fix}$  の和である。

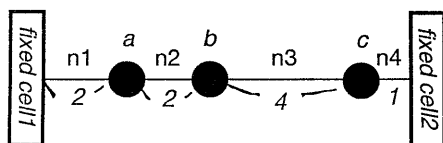


図 2: 重心法による一次元配置

## 2.2 レジスタ間の接続関係の定義

本手法では、配置にレジスタ同士の接続関係を示すネットを用いる。

この接続関係はクリティカルパスを考慮しなければならないレジスタ対及び、レジスタと入出力ピンに対して与える。このレジスタ同士の接続関係を表すネットを、通常のネットに対して、レジスタ間ネットとよぶ。

レジスタ間ネットの結合度は、式 (3) から計算されるクリティカルパス遅延値の逆数とする。この値には、パス上のセルの固有遅延も含まれているので、パスの段数も同時に考慮することになる。

配置アルゴリズム上ではこのレジスタ間ネットは、設計者に指定される。

## 3 アルゴリズム

### 3.1 重心法

配置問題のコスト関数は、セルの集合を  $\epsilon$ 、セル  $i$  と  $j$  を結ぶネットの結合度を  $C_{ij}$  として、

$$L = \frac{1}{2} \sum_{i,j \in \epsilon} C_{ij} \{ (x_i - x_j)^2 + (y_i - y_j)^2 \} \quad (4)$$

で、表される。ここで、 $(x_i, y_i)$  はセル  $i$  の座標を表している。配置は  $L$  が最小になるように行なう。

あるセル  $i$  の座標が決まっていない時、 $L$  を最小とする  $i$  の  $x$  座標  $x_i$  は、 $x$  方向と  $y$  方向で独立とすれば、

$$x_i = \frac{\sum_{j \neq i \in \epsilon} C_{ij} x_j}{\sum_{j \neq i \in \epsilon} C_{ij}} \quad (5)$$

となつて、これは  $i$  に関する重心位置を求める式に他ならない。すなわち  $L$  を最小化するための  $x$  座標は重心を求める式から計算される。 $y$  座標についても同じことがいえる。

セル座標が未定の全セルについて、 $xy$  両方向について重心を求める式 (5) を書き下して、左辺に座標未定のセルに関する項、右辺に座標確定のセルに関する項を集めれば、それらは、確定座標を基準として、全セルの座標を求めることができる連立一次方程式となる。この連立方程式を解いて座標を求める手法を重心法とよぶ。

図 2 に、上記重心法を用いて配置を行なった場合の例を示す。

この例では、セル 1 と 2 の位置が決まっています、その間にセル  $a, b, c$  があり、それらはネット  $n1, n2, n3, n4$  で接続されている。各ネットの結合度を

$$C_{1a} = 2; C_{ab} = 2; C_{bc} = 1; C_{c2} = 4$$

とし、1 と 2 の距離を 9 とする。

重心法により座標を計算すると、各ネットの長さは、

$$n1 = 2; n2 = 2; n3 = 4; n4 = 1$$

となる。

本手法ではこの重心法を、レジスタ間ネットとセル間ネットに交互に適用し、配置を行なう。連立一次方程式を解くアルゴリズムには、SOR 法 [7] を用いた。

### 3.2 全体のアルゴリズム

配置のアルゴリズムの全体の流れを下に示す。各ステップでの詳細な内容は次節以降に示す。

#### (step0) セル間ネットリストの変更

座標または方向が指定されていない入出力ピンと、それに鎖状に接続しているセルをネットリストから除外する。

#### (step1) レジスタの展開順序の決定

配置アルゴリズム内では、点として扱うレジスタの座標から、構成しているフリップフロップの座標を決める。その時にはレジスタの座標を中心にしてフリップフロップを横一列に並べる。これをレジスタの展開とよぶ。重心法によりレジスタのフリップフロップへの展開順序を決定する。

#### (step2) 初期配置

レジスタもセルも点として扱い、重心法により初期配置を行なう。

#### (step3) セル列への割り当て

繰り返し、領域分割と配置を行ない、セルをセル列へ割り当てていく。

#### (step4) 配置改善

PI により配置改善を行なう。

### 3.3 ネットリストの変更

重心法により配置を行なうにはチップの入出力ピンの座標が与えられていなければならない。

本手法では、レジスタ間ネットの入出力ピンの座標は必ず与えられなければならないが、それ以外のピンについては、配置の過程で座標を決めることができるようにした。

そのために、座標または方向が与えられていない入出力ピンについてはネットリストから除外し、配置の過程で、ピンの座標を決めることにした。同時に、そこから鎖状に接続のあるセルに関するネットも除外する。このステップで除外されるセルを不定セルとよぶ。

不定セルは、各入出力ピンと、フリップフロップを始点とするネット探索により判別できる。

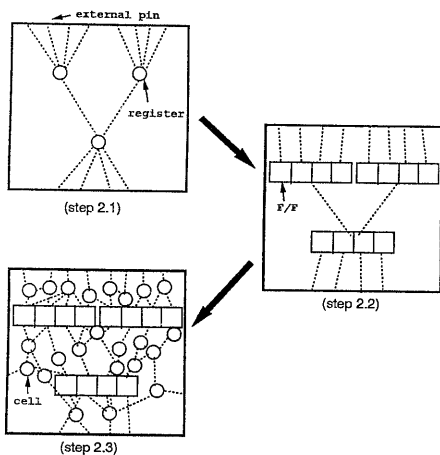


図 3: 初期配置のイメージ

### 3.4 レジスタの展開順序の決定

本手法では、レジスタは、同一セル列内でフリップフロップへと展開するものとする。展開するフリップフロップ列の中心座標はレジスタの座標である。この各レジスタ内でのフリップフロップの展開順序が配置結果を大きく左右する。

本手法では、重心法を利用してフリップフロップの展開順序を決定する。

このステップで決める展開順序は最後まで用いられる。

#### (step1.1) $x$ 座標の計算

セル間のネットリストを用いて、全セルの  $x$  座標を重心法により計算する。

#### (step1.2) ソーティング

計算した座標順にセルをソーティングする。

#### (step1.3) 展開順序の決定

各レジスタを構成しているフリップフロップをソーティング順に探し、それを展開順序とする。

### 3.5 初期配置

重心法を利用して初期配置を行なう。

配置のイメージを図 3 に示す。図では 2 次元で表示しているが、実際には、次節で述べるカットラインの最初の方向に従い、 $x$  か  $y$  方向のどちらかの座標だけ計算すれば良い。

以下の順序で初期配置を行なう。

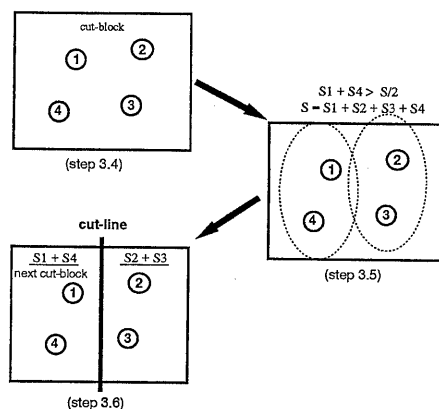


図 4: カットラインによるカットブロック生成

#### (step2.1) レジスタの座標の計算

レジスタ間ネットを用いて、レジスタの座標を重心法で計算する。

#### (step2.2) フリップフロップへの展開

決められた展開順序で、全レジスタをそれを構成しているフリップフロップへ展開する。

#### (step2.3) セルの座標の計算

セル間ネットを用いて、フリップフロップ位置を固定として、他のセルの座標を重心法で計算する。

### 3.6 セル列への割り当て

初期配置では、セルはチップの中心に集まってしまう、セル列への割り当てが全く考慮されていない。そこで、カットラインで配置領域を 2 分割しながらセル列への割り当てを行なう。以下ではカットラインによって分割された配置領域をカットブロックと呼ぶことにする。

カットラインによるカットブロック生成の手順を図 4 に示した。

#### (step3.1) セルの座標の決定

カットブロック内のセル列の段数が一段だったら、そこに所属するセルの座標を  $x$  方向の重心法により決定する。違うカットブロックを選び (step3.1) へ。そうでなければ (step3.2) へ。

#### (step3.2) ネットの復元

セル列の段数が少なく、かつそのブロック内に不定セルがある場合には、それとネットを元に戻す。そうでなければ (step3.3) へ。

### (step3.3) カットライン方向の決定

カットラインの方向を決定する。ここで決まる方向は、(step3.6)で、今のカットブロックをどちらの方向に切って2分割するか、である。

方向は、基本的に前回の垂直方向とするが、カット後のブロックの幅が幅最大のレジスタを越えないような方向にする。

### (step3.4) 座標計算

(step3.3) のカットラインに垂直方向に、カットブロック内での座標を計算。座標計算は初期配置と同じ手順で重心法により行なう。

ここでは、カットブロック外にあるセルについては領域上に座標を投影し、確定座標とみなして計算する。

### (step3.5) セル面積の加算

計算した座標順にセルの面積を加算していく。この時、不定セルも加算の対象とする。その座標は、不定セルに接続上最も近いセルの座標とする。

### (step3.6) カットブロックの生成

加算面積計が全セルの面積計の半分を越えたところで、(step3.3) のカットラインにより、新しいカットブロックを作る。

(step3.1) へ。

## 3.7 配置改善

このステップでは、フリップフロップ以外のセルを対象としたPIによる配置改善を行なう。フリップフロップが理想的な座標に配置されていれば、PIだけでもかなり良い改善結果が期待できる。

## 4 配置実験

### 4.1 プログラミング

前章のアルゴリズムを基に配置プログラムを作成し、配置実験を行なった。プログラムは、C言語で、Sparc Station2上で作成した。

### 4.2 実験モデル

本手法では、設計者がレジスタ間ネットを作成する必要がある。実験にはレジスタ間の信号の流れがはっきりとわかるように、レジスタ間の組合せ回路部の段数が少なく、かつレジスタが多いモデルを選んだ。

配置実験モデルは、横20縦24段のゲートアレイである。フリップフロップ一つで、レジスタを構成しているものもある。論理設計段階では、図5のような信号の流れに沿って、レジスタが置かれることをイメージして設計されてい

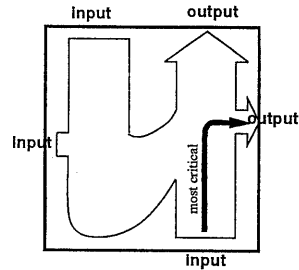


図 5: 実験モデルの信号の流れ

セル数	ネット数	外部ピン数	F/F数	レジスタ数
415	568	51	206	43

表 1: 配置実験モデル

る。右下の入力から右端の出力への信号は特にクリティカルになっている。主な仕様を表 1 に示す。

### 4.3 実験

配置実験は、比較のため、表 2 に示す 3 つの手法で行なった。

手法	条件
本手法	一括配置。 レジスタを考慮。
手配置	論理設計結果を基にしたブロック分割を利用。 レジスタを考慮。
PI	論理設計結果を基にしたブロック分割を利用。 レジスタの考慮なし。

表 2: 配置実験を行なった手法

### 4.4 結果

#### 4.4.1 面積、ネット長

配置結果の、高さ、総配線長の比較を表 3 に示す。

セルはゲートアレイだが、配線結果までの面積を比較するため、スタンダードセルの配線と同様に、配線に使用した分だけ配線領域をとるようにした。

ネット長の分布を示すグラフを図 6 に示した。手配置での結果が、総配線長で一番短く、ネット長の分布をみても長いネットが少ないことがわかる。本手法による結果は、手配置とPIの間の結果が得られた。PIでは、長いネットが多い。

手法	CPU time(s)	高さ ( $\mu\text{m}$ )	総配線長 ( $\mu\text{m}$ )(比)
本手法	40.0	4616.5	712620.2(1.00)
手配置	—	4616.5	625126.2(0.88)
分割+PI	36.0	4620.0	747110.0(1.05)

表 3: 面積、総配線長の比較

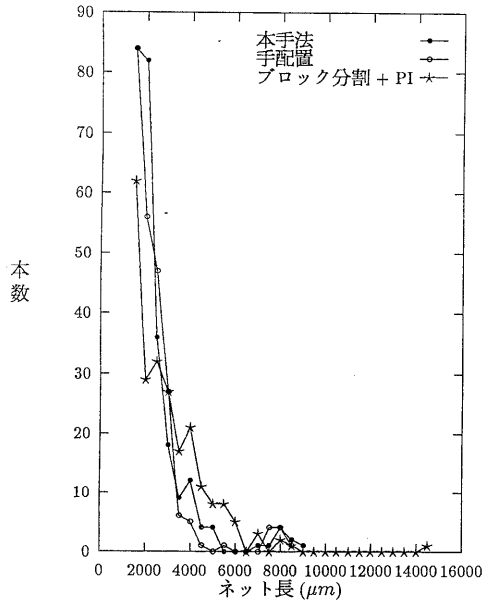


図 6: ネット長分布の比較

#### 4.4.2 レジスタ位置

各手法の配置結果のレジスタ位置による比較も行なった。位置を示す図を図7から9に示す。

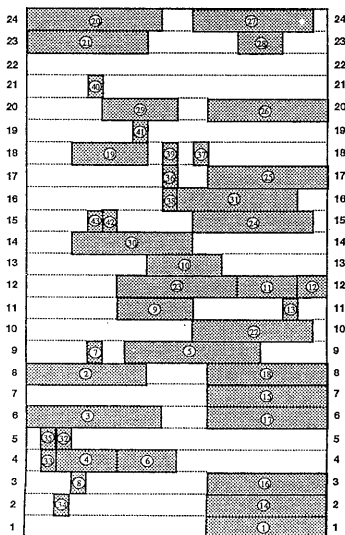


図 7: 本手法によるレジスタ位置

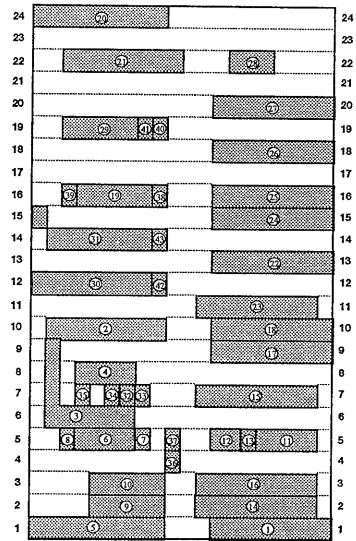


図 8: 手配置によるレジスタ位置

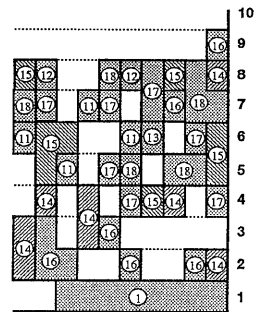


図 9: PIによるレジスタ位置 (右下の2ブロックのみ)

PIの結果に関しては、一番クリティカルな右下の2ブロック分だけを図示する。レジスタの考慮をしていないので、各レジスタが離散的に配置されているのがわかる。例えば、図中で、斜線で塗りつぶしたレジスタ14とレジスタ15は、14が出力側、15が入力側の8ビットの信号の流れによる接続関係があるが、配置結果には何の規則性も見られていない。

一方、本手法による配置結果をみても、レジスタ同士が、与えられた接続関係に従い、手配置の場合に近い形で配置されていることがわかる。しかし、左半分のレジスタについては、中央に寄ってしまっている。これは、左下に入出力ピンが少なかったためと思われる。

最後に、配線結果の図を図10から12に示す。レジスタ位置から予想されるように、手配置による結果がレジスタに沿った配線がなされているのがわかる。

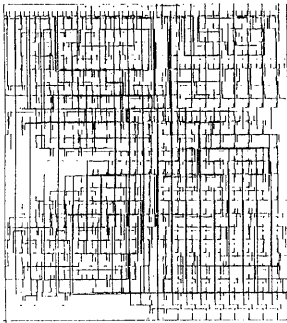


図 10: 本手法による配線結果

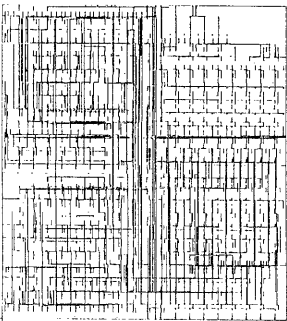


図 11: 手配置による配線結果

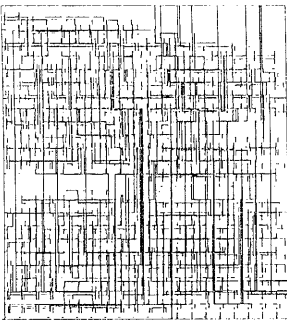


図 12: PI による配線結果

## 5 まとめ

タイミング制約を考慮するセル配置手法として、レジスタに着目する配置アルゴリズムを考案した。アルゴリズム内では、レジスタ間ネットを用いることにより、レジスタ間の接続関係を陽に考慮した。

実験の結果、レジスタを考慮し、手で配置をした結果には及ばなかったが、従来のレジスタを考慮しない配置より良い結果が得られた。

今後は、クリティカルパス遅延値を考慮できる配置改善手法の検討を行なう予定である。その時には、クロック・スキューの考慮も行なう。

## 謝辞

最後に、本研究を行なうにあたり、多大な御指導、御助言を頂いた、NTT LSI 研究所の小野沢晃主任研究員、並びに北沢仁志主幹研究員に感謝致します。

## 参考文献

- [1] Y. Ogawa, T. Ishii et al.: "Efficient Placement Algorithms Optimizing Delay For High-Speed ECL Master Masterslice LSI's", Proc. 23rd DAC, pp. 404-410(1986).
- [2] T. Gao, P.M.Vaidya and C.L.Liu: "A New Performance Driven Placement Algorithm", Proc. of ICCAD, pp. 44-47(1991).
- [3] T. Hasegawa: "A New Placement Algorithm Minimizing Path Delays", Proc. of ISCAS, pp. 2052-2055(1991).
- [4] M. A. B. Jackson and E. S. Kuh: "Performance-Driven Placement of Cell Based IC's", Proc. 26th DAC, pp. 370-375(1989).
- [5] A. Srinivasan: "RITUAL: A Performance Driven Placement Algorithm for Small Cell ICs", Proc. of ICCAD, pp. 48-51(1991).
- [6] D. A. Joy and J. Ciesielski: "Placement For Clock Period Minimization With Multiple Wave Propagation", Proc. of 28th DAC, pp. 640-643(1991).
- [7] 戸川隼人: "詳解 数値計算演習"(1980).
- [8] R.-S. Tsay, E. S. Kuh and C.-P. Hsu: "PROUD: A Fast Sea-Of-Gates Placement Algorithm", Proc. 25th DAC, pp. 318-323(1988).