

## Rent's Rule に基づく概略配線問題生成手法

高辻 和秀 白石 洋一

群馬大学工学部 〒376-8515 群馬県桐生市天神町 1-5-1

E-mail: (kazuhide, siraisi)@keim.cs.gunma-u.ac.jp

あらまし 本論文では、Rent's Rule に基づいて概略配線問題を自動生成する手法について述べる。VLSI チップの自動レイアウト処理において、概略配線処理はチップのレイアウト結果を左右する重要な処理である。概略配線アルゴリズムの評価では、従来、Web 上などで公開されているベンチマークデータを使用していた。しかし、DSM プロセスにおける超大規模チップに対する概略配線アルゴリズム開発においては、現在、使用可能な超大規模概略配線問題は存在しない。また、たとえ超大規模な論理データを作成したとしても、概略配線問題をを得るためには配置処理を実行しなければならず、さらに、得られた配置結果は配置アルゴリズムの性能に大きく依存する。本論文では、Rent's Rule に基づいて、チップの性能を推定する既存の手法を拡張して、従来得られていない規模の概略配線問題を生成する手法を示す。

キーワード VLSI, Rent's Rule, 概略配線, ベンチマークデータ, 配置, 配線

## A Global Routing Problem Generation Method based on Rent's Rule

Kazuhite TAKATSUJI and Yoichi SHIRAIISHI

Faculty of Engineering, Gunma University 1-5-1 Tenjin-cho, Kiryu, Gunma, 376-8515 Japan

E-mail: (kazuhide, siraisi)@keim.cs.gunma-u.ac.jp

**Abstract** This paper presents a method which generates global routing problems based on Rent's Rule. In the layout synthesis of a VLSI chip, a global routing is a very important process which greatly affects the performances of a generated chip design. Conventionally, benchmark data published, for example, on Web pages are used in evaluating the performances of a global routing algorithm. However, there are no such benchmark data that can be used for developing a global routing algorithm applied in designing a next-generation VLSI chip in the DSM fabrication process. Even if the logical data are generated, a placement process must be executed for generating a global routing problem and moreover, the generated problem is strongly dependent of the performances of the placement program. In this paper, a global routing problem generation method based on Rent's Rule is suggested by extending the conventional method for estimating the chip performances. This method is applicable for generating such very large scale global routing problems that have not yet been generated before.

**Keyword** VLSI, Rent's Rule, Global Router, Benchmark Data, Placement, Routing

### 1. 緒言

VLSI チップの超大規模化、高性能化、および製造プロセスの微細化により [1]、自動配置配線アルゴリズムの開発はより困難さを増してきている。また、アルゴリズムを開発してもそれを評価すべきベンチマークデータを作成することも非常に困難である。現在、特に Web ページなどで入手可能なベンチマークデータ [2] は、次世代配置配線アルゴリズム評価には規模の点では不十分である。このような状況の下、次世代配置アルゴリズムを開発するために使用可能な超大規模配置問題生成手法が提案され、生成された配置問題が使用され始めている [3, 9]。しかし概略配線アルゴリズムの評価のためにこの問題をそのまま使用することはできない。なぜなら、配置処理を実行して部品の配置座標

を決定することが必要であることと、さらに決定的なのは、ネットはすべて 2 端子ネットのみで概略配線アルゴリズムの実用レベルの評価にはならない。今後、超大規模な論理設計データが公開されたとしても、概略配線問題をを得るためには配置処理を実行しなければならず、そのような問題に対して配置処理が実行可能かどうか、高性能な、すなわち概略配線可能な配置結果を生成可能かどうかの保証はない。

そこで本論文では、問題の規模、数種類のパラメータ、およびデザインルールを元に直接概略配線問題を生成する手法を提案する。これらの入力データをもとにチップのサイズ、消費電力などの性能を推定する手法はすでに提案されている [4, 5]。この手法は Rent's Rule に基づいて総配線長を理論的に計算し、それを元にチップの性能を推定するもので、配置、配線問題を

生成することを目的としたものではない。本論文ではこの手法を拡張して配線問題のネットリストを自動生成し、端子の位置を決定して概略配線問題を生成する。この手法をプログラム化し、実際の概略配線問題と比較して生成された問題が実問題に近いことを示す。

本論文の構成を示す。第2節では概略配線問題生成問題を定義し、続いて概略配線問題生成手法の全体の流れについて述べる。第3節ではDavisらによるチップ性能推定手法について述べる。第4節では第2節で示した概略配線問題生成のアルゴリズムを示す。第5節では、生成した概略配線問題の生成時間、実問題との混雑度比較における評価結果を述べる。第6節では概略配線問題以外への応用を示し、最後に第7節で全体を結言としてまとめる。

## 2. 概略配線問題生成と生成手法の構成

本節では概略配線問題を生成する問題を定義し、それを解くための概略配線問題生成手法の概要を示す。

### 2.1. 概略配線問題生成問題

本論文で取り扱う概略配線問題を生成する問題を次で定義する。

#### 定義1 概略配線問題生成問題

入力 ゲート数、平均ファンアウト数、Rent's パラメータ、ゲートピッチサイズ、配線層数、配線ピッチサイズ、配線効率係数 [4]、

出力 ネットリスト、チップサイズ、

目的関数 生成した問題の混雑度分布と実際の配線問題の混雑度分布の差の最小化。

### 2.2. 概略配線問題生成手法の流れ

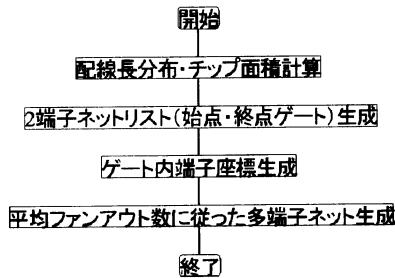


図1: 概略配線問題生成手法の流れ

図1に、本論文で提案する概略配線問題生成手法の全体の流れを示す。まず、Rent's Rule に従って配線長毎の2端子ネット数分布とチップサイズを求める [4, 5]。続く処理で、各2端子ネットの始点、終点ゲートを決定する [4]。次にゲート内の端子位置を決定するが、これは、端子位置をセル外枠上などに限定せず、

ゲート内の任意の位置にランダムに決定する。最後に、入力として指定される平均ファンアウト数に従って2端子ネットから多端子ネットを生成する。

## 3. チップ性能推定方法

本節では Davis ら [4, 5] によって提案されたチップの性能推定方法に関して示す。

### 3.1. 配線長分布計算式

Davis らの方法に従って配線長分布の計算を行う。 $l=a$ 、 $l=b$ の間の配線長を持つ配線本数  $I(a < l < b)$  は次式で与えられる。

$$I(a < l < b) = \int_a^b i(l) dl.$$

ここで、 $i(l)$  は配線密度関数で、次の2式で与えられる。

$$\text{領域 I : } 1 \leq l \leq \sqrt{N}$$

$$\frac{\alpha k}{2} \Gamma \left( \frac{l^3}{3} - 2\sqrt{N}l^2 + 2Nl \right) l^{2p-4}$$

$$\text{領域 II : } \sqrt{N} \leq l \leq 2\sqrt{N}$$

$$\frac{\alpha k}{6} \Gamma (2\sqrt{N} - l)^3 l^{2p-4}$$

ここで、 $l$  は配線長 (ゲートピッチ単位)、 $N$  は論理ゲート数、 $p$  は Rent's 指数、 $\alpha$  はシンク端子数で、これは平均ファンアウト数  $f.o.$  と次の関係を持つ。

$$\alpha = \frac{f.o.}{f.o. + 1}$$

また  $\Gamma$  は次式である。

$$\Gamma = \frac{2N(1-N^{p-1})}{\left( -N^p \frac{1+2p-2^{2p-1}}{p(2p-1)(p-1)(2p-3)} - \frac{1}{6p} + \frac{2\sqrt{N}}{2p-1} - \frac{N}{p-1} \right)}$$

図2はこれらの計算式から得られた配線長分布の一例である。

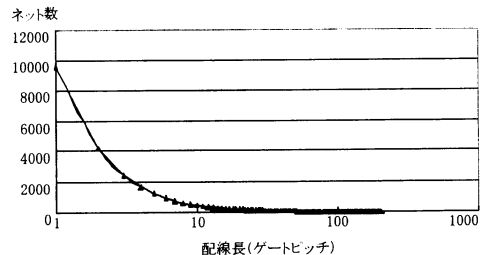


図2:  $N=10000$ 、 $p=0.75$ 、 $k=4.0$ 、 $f.o.=3.0$  における配線長分布

### 3.2. チップサイズ計算

Davis らの方法に従って配線領域面積の計算を行う。レイアウトに必要な配線領域面積 ( $A_{required}$ ) は下記の式で与えられる。

$$A_{required} = \sqrt{\frac{A_c}{N}} p \chi L_{total}$$

ここで、 $A_c$  はチップサイズ(ゲート領域+配線領域)、 $N$  はゲート数、 $p$  は配線ピッチ ( $\mu m$ ) で、配線幅/2×2+ (配線スペーシング) で計算される値である (同じ記号  $p$  を使用しているが、3.1 で使用している Rent's 指数ではない)。 $\chi$  はファンアウト数を考慮した補正式  $\chi = 4/(f.o.+3)$ 、 $L_{total}$  は配線長合計 (ゲートピッチ) である。

文献 [5] では  $A_c$  と  $A_{required}$  との間に下記の関係性を求めている。

$$A_{required} = A_c e_w n_{levels}$$

ここで、 $e_w$  は配線効率係数、 $n_{levels}$  は配線層数である。配線効率係数はチップサイズに占める配線領域サイズの割合で、例えば文献 [5] では 0.4 としている。これらにより下記のチップサイズ推定式を得る。

$$A_c = \frac{A_{required}}{e_w n_{levels}} = \frac{\sqrt{\frac{A_c}{N}} p \chi L_{total}}{e_w n_{levels}}$$

この式より本実験で使用する下記のチップサイズ推定式を得る。

$$A_c = \frac{p^2 \cdot \chi^2 \cdot L_{total}^2}{N \cdot e_w^2 \cdot n_{levels}^2}$$

### 4. 概略配線問題生成アルゴリズム

本節で示す概略配線問題生成アルゴリズムは、図 1 の 2 端子ネットリスト生成、ゲート内端子座標生成、および平均ファンアウト数に従った多端子ネット生成の各処理を行うアルゴリズムである。

#### 4.1. 2 端子ネットリスト生成

3.1 で求めた配線長分布に従って 2 端子ネットを生成する。Davis の考え方 (文献 [4] appendix) に基づいてネット生成手法を使用する。図 3 では、左上隅のゲート ( $S_1$ ) から順に見てネットを生成する状況を示している。このゲートについてネットを生成し終えたら、左上隅のゲートを削除して同様の処理を行う。

具体的にネットの生成方法を述べる。あるゲート配置において、注目ゲート数  $N_A$ 、注目ゲートと注目配線長を持つゲート間のゲート数  $N_B$ 、注目配線長を持つゲート数  $N_C$  を数え上げる。図 3 の左の例の場合、注目ゲートを  $A=\{S_1\}$ 、注目配線長を 3 とすると、 $B=\{1, 2$  のラベルを持つゲート)、 $C=\{3$  のラベルを持つゲート)と

なり  $N_A=1$ 、 $N_B=5$ 、 $N_C=2$  である。従って Rent's Rule を繰り返し適用して、ゲート A から C への配線本数  $I_{a-to-c}$  は次式で与えられる。

$$I_{a-to-c} = \alpha \cdot k \left[ (N_A + N_B)^p - (N_B)^p + (N_B + N_C)^p - (N_A + N_B + N_C)^p \right]$$

$p$ 、 $k$  および  $\alpha$  は 3.1 で示した係数である。

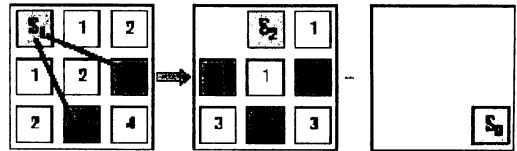


図 3: 9 ゲートにおける 2 端子ネットの生成例

図 3 に示したすべてのゲート、 $S_1, S_2, \dots, S_9$  に対して  $I_{a-to-c}$  を求めた結果を表 1 に示す。パラメータは、 $N=9$ 、 $p=0.75$ 、 $k=4.0$ 、 $f.o.=3.0$  である。表 1 では、配線長降順、 $I_{a-to-c}$  昇順でソートしている。

表 1: 生成したネット一覧表

注目ゲート	NA	NB	NC	Ia-to-c	配線長	選択
S3	1	5	1	0.0600	4	
S1	1	7	1	0.0420	4	
S3	1	3	2	0.1770	3	
S6	1	2	1	0.1460	3	
S1	1	5	2	0.1100	3	
S2	1	5	2	0.1100	3	
S4	1	4	1	0.0760	3	
S3	1	1	2	0.3990	2	*
S1	1	2	3	0.3230	2	*
S2	1	2	3	0.3230	2	
S6	1	1	1	0.2520	2	
S7	1	1	1	0.2520	2	
S4	1	4	1	0.0760	2	
S5	1	2	2	0.2470	2	
S1	1	0	2	1.2070	1	*
S2	1	0	2	1.2070	1	*
S4	1	0	2	1.2070	1	*
S5	1	0	2	1.2070	1	*
S3	1	0	1	0.9550	1	*
S6	1	0	1	0.9550	1	*
S7	1	0	1	0.9550	1	*
S8	1	0	1	0.9550	1	*

この表においてどのネットを採択するかを決定する。3.1 で示した配線長分布の計算の結果、配線長 1: 9 本、配線長 2: 2 本が得られ、これに基づいて行う。ネットの採否は、同一配線長を持つネットに関しては  $I_{a-to-c}$  の値の降順とする。なぜなら、 $I_{a-to-c}$  の値が大きいほどネットの存在確率が高いからである。ここで表 1 において、配線長 1 を持つネットは 8 本しか存在しない。しかし要求は 9 本である。この場合、配線長 1 を持ち  $I_{a-to-c}$  が最大値となるネットをコピーして 9 本とする。選択したネットは選択のカラムに \* を付している。採択したネットを図示して図 4 を得る。

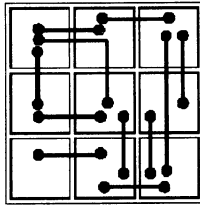


図 4: 表 1 で採択された 2 端子ネット. 端子位置はゲート内任意

#### 4.2. ゲート内端子座標生成

上で求めた 2 端子ネットの始点, 終点はゲートである. ゲート内の座標決定をこの処理で行う. 通常, ゲートの端子位置は上下辺などに固定されているが, 概略配線処理では厳密な位置は必要ではない. そのため, ゲート内の 1 点をランダムに選択して端子座標とする.

#### 4.3. 平均ファンアウト数に従った多端子ネット生成

4.1 で生成した 2 端子ネットリストを平均ファンアウト数に従った多端子ネットリストへ変換する. 基本的な考え方は, “あるファンアウト数を持つネットの配線長を満たすように 2 端子ネットから多端子ネットを生成する” と言うことである. また, これまでにファンアウト数を考慮した多端子生成アルゴリズムは提案されていない.

まず, あるファンアウト数を持つネットの配線長を推定する. Davis らの方法 [4] ではファンアウト数を考慮した配線長の補正を行っている. そこで使用しているファンアウトモデルは図 5 に示すリニア配線ネットモデルである. ここで  $s = \text{配線長} / 2 \text{ 端子配線本数}$  である. このモデルに従って多端子ネットを生成する.

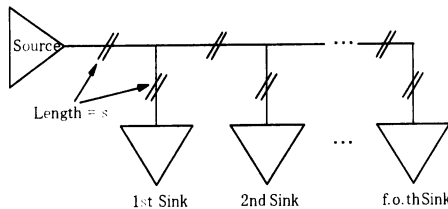


図 5: リニア配線ネットモデル

$s$  の値より, 各ファンアウト数  $f.o.$  を持つネットの平均配線長は  $(2 \cdot f.o.)s$  である. この配線長を満たすように, かつ図 5 に示したリニア配線ネットモデルに従って 2 端子ネットの端子をマージすることにより, 多端子ネットを生成する.

##### 4.3.1. ファンアウト数毎のネットの配線長

あるファンアウト数を持つ多端子ネットの配線長に合わせて 2 端子ネットから多端子ネットを構成する. 従ってファンアウト数毎の平均配線長が必要である.

これは図 5 に示す従来データの分布から求める. 図 5 に示すように分布曲線の形状は問題の規模に関して相似であると考えられる. 従って, 問題の入力で与えられる平均ファンアウト数と図 2 の配線長分布よりファンアウト数毎の平均配線長を求めることができる.

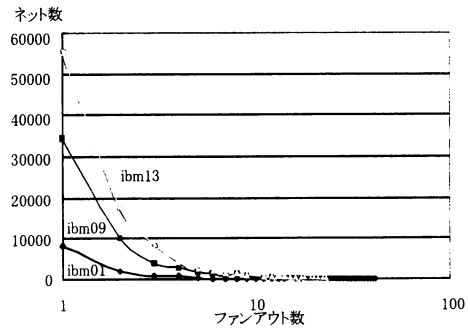


図 5: ファンアウト数毎のネット数分布 [2]

#### 4.3.2. 多端子ネット生成アルゴリズム

アルゴリズムは下記に示す greedy 法である.

【 $f.o. = k$  の多端子ネットを生成する場合】

1. このファンアウト数の配線長を求める (4.3.1).
2. ベースになる長さ  $(f.o. + 1)s$  以下の配線をランダムに選択する.
3.  $k = 1$  なら終了. そうでない場合は, 下記を繰り返す.
  - 3.1 ベースになる配線の端子と同一ゲート内に端子を持つ長さ  $s$  以下の配線をランダムに選択し, 選択した配線を結合する.
  - 3.2  $k = k - 1$ .

#### 5. 実験と評価

提案する概略配線問題生成手法を評価した結果を示す. 生成した問題は, ネットリストとして正しく, また, データフォーマットが合っていさえすれば概略配線問題として使用可能である. しかし, 実際の概略配線問題にどの程度類似しているかどうかを評価することが必要である. 以下では, 新たに評価基準を提案して, それにより問題の質を評価する.

##### 5.1. 開発プログラムと実験環境

本概略配線問題生成手法を C++ 言語, LEDA クラスライブラリを用いてプログラム化し, 実験, 評価した. プログラムの実行は, 3GHz MPU, 4GB メインメモリマシン上である.

##### 5.2. 問題生成におけるパラメータ

生成した概略配線問題は既存のベンチマークデータ ISPD98 [2] のベンチマークデータに近くなるようにゲート数, 2 端子ネット数, 平均ファンアウト数を設定したものである. これはベンチマークデータと今回生成した概略配線問題との質の比較を行うためである.

生成する問題のパラメータを表2に示す。Ex. 1, Ex. 2, Ex. 3はそれぞれ ibm01, ibm13, ibm18に対応する問題を生成するためのパラメータである。Ex. 4は ISPD98の最大規模ベンチマークデータ (ibm18) を超えるネット数を持つ配線問題生成に対応する。その他のパラメータは、 $p=0.75$ ,  $k=4.0$ ,  $e_w=0.4$ ,  $n_{levels}=2$  である。

表2: 生成した概略配線問題のパラメータ

問題名	ゲート数	2端子ネット数	平均ファンアウト数
Ex. 1	13,930	36,455	2.58
Ex. 2	94,696	257,409	2.58
Ex. 3	215,074	617,777	3.05
Ex. 4	393,779	1,143,372	3.10

### 5.3. 生成した問題の質の評価

提案した概略配線問題生成手法の配線長合計とチップサイズはそれぞれ実問題に類似していることがすでに実証されている [5]。以下では、これに加えて混雑度分布がどの程度実問題に類似しているかを比較する。概略配線問題の難易度を示す指標として混雑度は最も重要な要素の一つである。概略配線アルゴリズムの目的関数の一つは混雑度準化である。混雑度が100%に近い、あるいは超えている領域を持つ概略配線問題(配置結果)を入力して、迂回配線を含めて混雑度を完全配線可能な程度に低下させることが可能かどうか概略配線アルゴリズムの性能を測る有効な尺度である。例えば、高性能な概略配線アルゴリズムを適用する場合、全体の混雑度の上位20%の平均値が80%程度の問題は完全配線可能であり、混雑度が85%を超えると経験的に配線不可能であることが示されている [6]。また、経験的に中心付近の領域の混雑度が高く周辺領域では徐々に混雑度が低下していく。ここで、混雑度は確率的に径路を推定するアルゴリズムによって求める [7]。

表3に、提案した概略配線問題生成手法によって生成した概略配線問題の評価結果を示す。チップサイズは推定した値を使用したもので、「実際にチップレイアウト可能」な問題に対応するものとして生成した結果である。表に示すように、上の混雑度分布基準から判断するといずれも配線可能であり、最大混雑度も100%以下であることから「比較的易しい」とみなすことができる。

表3: 生成した概略配線問題の混雑度と生成時間

問題名	最大混雑度	高混雑度 20%平均	生成時間 (分)
Ex. 1	95	67.0	6
Ex. 2	78	59.8	393
Ex. 3	69	56.9	2,764
Ex. 4	70	50.3	12,653

表3は全体のマクロな評価であることから、より詳細な混雑度が実問題に近いかどうかを検証する必要がある。混雑度分布のプロファイルの比較結果を図6に

示す。ネット数、チップサイズ、Rent'sパラメータなどの違いから分布形状は一致しないが、混雑度が高い辺数から低い辺数への減少傾向は近いと考えられる。ここで辺数とは、その混雑度を持つタイルの辺数である。

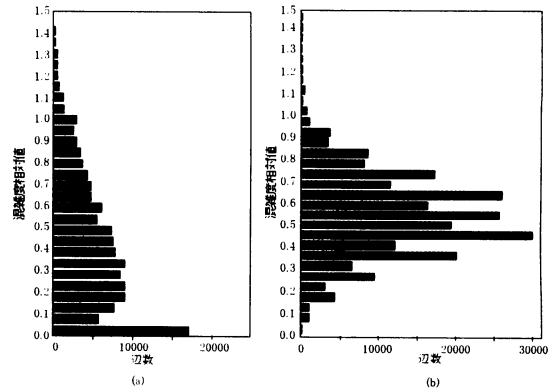


図6: 混雑度分布プロファイル比較 (a)実問題 (696,377 ネット) [8] (b)Ex. 1

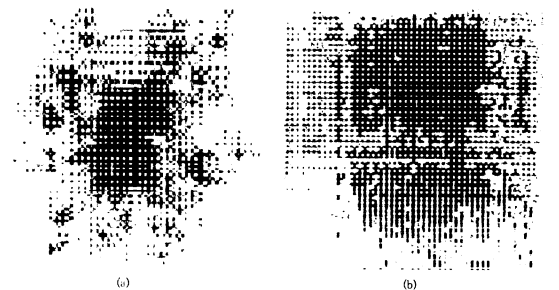


図7: 混雑度分布 (50x50 タイル分割) (a)実問題 (73,273 ネット) [6] (b)Ex. 2

同等の規模の概略配線問題に対して、混雑度の大小をグラデュエーションで示した結果を図7に示す。生成した問題の高混雑度領域は中心付近からずれてはいるが、全体の分布は実問題に類似している。

問題の生成時間に関しては、問題規模の2乗程度のオーダで増大する。最大のネット数114万の問題生成 (Ex. 4) では8.7日間かかるが、この規模の問題生成を論理設計と配置処理実行を経由して行うよりも小さく、また概略配線問題はベンチマークデータとして数回生成すればよいことから、提案手法は問題の生成時間の点で実用レベルにあると考えられる。

本提案手法のメリットであるさまざまな難易度の問題を生成可能なことを実証する。具体的には、計算で求めたチップサイズの推定値 (3.2参照) を縮小、または拡大することで問題の難易度を変更することで、問題 Ex. 1のチップサイズを120%に拡大、また

は80%に縮小した場合のそれぞれの混雑度を表4に示す。

表4：概略配線問題の難易度変更

サイズ比	1辺の格子数	最大混雑度	高混雑度20%平均
120%	3015	84	67.0
100%	2513	146	85.1
80%	2010	179	95.2

先に述べた混雑度による問題の難易度の尺度において、問題の難易度が対応して変化することが示されている。

本実験、評価では混雑度を評価基準として、生成した概略配線問題の質を比較した。比較対象があれば、さらに信号伝播遅延などの電気的特性を実現するための制約を付加して概略配線問題を生成することが可能である。今後の課題として、これらの制約を含めて提案手法の実験、評価を行う。

## 6. 他の分野への応用可能性

提案した概略配線問題生成手法の下記の2分野への応用も考えられる。今後の課題としてこれらの分野への応用を検討する。

### 6.1. 配置問題生成への応用

配置問題では、ネットリストとセルの情報が必要である。提案手法では部品をゲートしていたが、セルライブラリのセルとゲート数のデータが与えられれば、それをもとにしてセルを定義し、複数ゲートから1セルを生成することができる。これにより配置問題のベンチマークデータを生成することが可能である。

### 6.2. 上流設計におけるチップ特性の高精度推定への応用

VLSIチップの設計支援は、より上流の設計フェーズである動作設計へと焦点を移しつつある。現在のチップ設計では、チップの性能である信号伝播遅延、ノイズ、発熱などを上流の設計フェーズで推定して対策を考慮していくことが必須である。従って上流設計ではレイアウトレベルの性能を高精度に推定することが求められている。そのためには少なくとも概略配線径路の推定が必要であるが、本提案手法は、ゲート数と実装パラメータをもとに概略配線径路までを推定することが可能である。これにより、上流設計でのチップの性能を高精度に推定する可能性を持つ。

## 7. 結言

Rent's Ruleに基づく手法を元に、概略配線問題を生成する手法を提案した。提案手法の新規性は、ネットリストを生成するために、ネット数分布推定、多端子ネット生成のそれぞれのアルゴリズムを提案したことにある。

実問題との類似性を比較するための尺度として混雑度を適用し、既存の実概略配線問題との比較を行った。その結果、生成した概略配線問題が実問題に高い

度合いで類似していることを示した。さらに、チップサイズを変更することでさまざまな難易度を持つ概略配線問題を生成可能なことを示した。

本報告では電気的特性を実現するための制約を含んだ概略配線問題の生成までを対象としていないが、この種の問題生成をも今後、対象としていく。

さらに、本手法の他の分野への応用として、配置問題生成とVLSIチップの上流の設計フェーズにおけるチップの性能の高精度推定への応用可能性を示した。これらの分野への応用検討は今後の課題である。

## 文 献

- [1] Semiconductor Industry Association, "International Roadmap for Semiconductors, 2002 Update", p.159, 2002.
- [2] "The ISPD98 Circuit Benchmark Suite," <http://vlsicad.cs.ucla.edu/~cheese/ispd98.html>, 1998.
- [3] C. C. Chang and Min Xie, "PEKO Suite (Placement Example with Known Optimal Wirelength)", <http://cadlab.cs.ucla.edu/pubbench/placement>, 2003.
- [4] J. A. Davis, V. K. De and J. D. Meindl, "A Stochastic Wire-Length Distribution for Gigascale Integration (GSI) - Part I: Derivation and Validation", IEEE Trans. on Electron Devices, (45)3:580-589, March 1998.
- [5] J. A. Davis, V. K. De and J. D. Meindl, "A Stochastic Wire-Length Distribution for Gigascale Integration (GSI) - Part II: Applications to Clock Frequency, Power Dissipation, and Chip Size Estimation", IEEE Trans. on Electron Devices, 45(3):590-597, March 1998.
- [6] U. Brenner and A. Rohe, "An Effective Congestion Driven Placement Framework," Proc. of the ISPD02, pages 6-11, April 2002.
- [7] J. Lou, S. Krishnamoorthy and H. Sheng, "Estimating Routing Congestion using Probabilistic Analysis," Proc. of the ISPD01, pages 112-117, April 2001.
- [8] C. Albrecht, "Global Routing by New Approximation Algorithms for Multicommodity Flow," IEEE Trans. on CAD/ICAS, 20(5):622-632, May 2001.
- [9] C. C. Chang, J. Cong and M. Xie, "Optimality and Scalability of Existing Placement Algorithms," Proc. of the ASP-DAC, pages 621-627, January 2003.