1V-4　複数行にわたるブロックを考慮したセル敷き詰め型ゲートアレイの配置手法

山本 知　若林 真一　宮尾 淳一　吉田 典可
広島大学工学部

1．まえがき

セル敷き詰め型ゲートアレイ(Sea-of-Gates Array)では，低開発コスト，短い開発期間という従来型ゲートアレイの利点を維持しつつ，レイアウト設計時の自由度が飛躍的に向上している．セル敷き詰め型ゲートアレイチップのレイアウト設計に関して現在使われているブロック配置手法は，従来型ゲートアレイに対する手法と同様に，高さが一定のブロックのみを対象としているものがほとんどである．しかし，レイアウト設計の自由度の大きさを利用して，高さが不均一な機能ブロックの集合を扱えれば一層の高集積化が期待できる．本稿では，高さが1の基本セル列から成るブロックに加えて，数セル程度の高さを持つブロックも配置の対象としたセル敷き詰め型ゲートアレイの配置手法を提案する．

2．配置問題

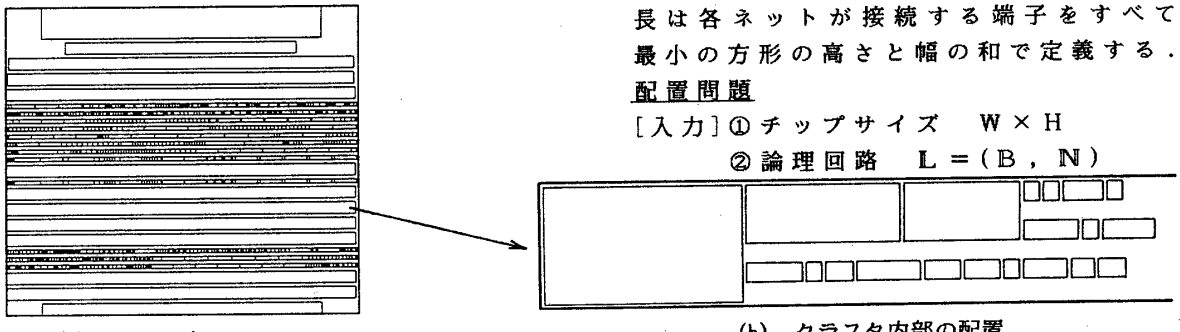
2.1　チップモデル　セル敷き詰め型ゲートアレイでは，あらかじめ基本セルが全面に配置された配置領域(マスタスライス)に対し，与えられた論理回路に基づいて配線を行うことにより，所望の機能を実現する．

セル敷き詰め型ゲートアレイのレイアウト設計は，通常，どの基本セルを論理回路のゲートとして使用するかを決める配置設計と，セル間の結線を行う配線設計の2段階で行われる．チップモデルとして，基本セルが水平方向にW個，垂直方向にH個配列された配置領域を考える．配線はメタル2層もしくは3層を用いて行われ，機能ブロック内部の配線に使われている部分を除き，配置領域の全面を配線に用いることができる．

2.2　配置モデル　チップ上に実現する論理回路として，論理回路を構成する機能ブロックの集合$\mathbb{B}=\{B_1,B_2,\cdots,B_m\}$と，機能ブロック間の接続情報であるネットリスト$\mathbb{N}=\{N_1,N_2,\cdots,N_n\}$が与えられる．機能ブロックは数個から数十個の基本セルで構成される方形であり，高さが数行にわたるものも許す．また，各ネットは接続される機能ブロックの集合で与えられる．論理回路は，ネットリストに基づいていくつかのクラスタに分割され，各クラスタはそれぞれ図1(a)のように行を構成して配置される．各クラスタ内では，図1(b)のように高さの異なる機能ブロックが混在することを許す．この配置モデルは，従来型ゲートアレイのような方形チャネルを確保することによって配線設計におけるチャネルルータの適用を可能としている．セル敷き詰め型ゲートアレイの機能ブロックの配置問題を次のように定式化する．ただし，配置問題の目的関数として，仮想配線長の総和を採用する．多端子ネットを仮定しているため，仮想配線長は各ネットが接続する端子をすべて囲む最小の方形の高さと幅の和で定義する．

配置問題

[入力]①チップサイズ　W×H
②論理回路　$\mathbb{L}=(\mathbb{B},\mathbb{N})$



(a)　チップ上のクラスタ　　(b)　クラスタ内部の配置

図1　配置モデル

A placement method for Sea-of-Gates array with multiple-row block placement
Satoshi YAMAMOTO, Shin'ichi WAKABAYASHI, Jun'ichi MIYAO, and Noriyoshi YOSHIDA
Hiroshima University

[出力]目的関数を最小にする機能ブロックの配置．

[目的関数]

$$Z = \sum_{i=1}^{n} (N_i \text{の仮想配線長})$$

3．アルゴリズム

3.1 処理の流れ セル敷き詰め型ゲートアレイの配置問題を解くアルゴリズムを以下に示す．

ﾌｪｰｽﾞ1:(論理回路の分割とクラスタの配置順序の決定)
論理回路を分割してクラスタを構成し，チップ上における配置順序を決定する．

ﾌｪｰｽﾞ2:(クラスタ内の機能ブロックの配置)
仮想配線長を見積もりながらクラスタ内での機能ブロックの配置を求める．

ﾌｪｰｽﾞ3:(クラスタ位置の決定)
クラスタ間の接続情報に基づいてクラスタ間のチャネルの高さを決定する．

3.2 クラスタリングアルゴリズム フェーズ1では，論理回路の各機能ブロックを機能ブロック間の接続情報に基づいてクラスタ化し，更に，チップ上における各クラスタの配置順序を決定する．総仮想配線長を最小にするため，クラスタリングでは総カット数を最小にすることを目的とする．

クラスタリングアルゴリズムは文献[1]の手法を拡張した．文献[1]では，ブロックの行割当ての際に入力全体に対して最小カット法[2]を適用している．最小カット法のアルゴリズムは反復改良法に基づいており，機能ブロック数に比例した計算時間の改良ステップを解が改善されなくなるまで繰り返す．このため，入力のサイズが大きくなると改良ステップの適用回数が増加し，その結果，計算時間が増大する場合がある．ここでは，段階的にクラスタを構成しつつ，それらとの接続に基づいて最小カット法の適用範囲を制限し，計算時間の改善を図っている．また，本クラスタリング手法では文献[1]と同様にクラスタリングと同時にクラスタの配置順序も決定している．各クラスタのサイズは，クラスタに属する機能ブロックのうち最も大きなもののサイズに基づいて動的に決める．クラスタのサイズの上限を以下に示す式で与える．ここでWは与えられたチップサイズの幅である．

$$Smax = \begin{cases} W(\text{高さ1の機能ﾌﾞﾛｯｸからなるｸﾗｽﾀ}) \\ W \times hmax \times p\ (hmax:\text{ｸﾗｽﾀに属する機能ﾌﾞﾛｯｸのうち最も大きなものの高さ}, \\ \quad p:\text{ｾﾙ使用率}) \end{cases}$$

論理回路のクラスタリングを行うアルゴリズムを次に示す．

S1：機能ブロックを任意に1個選ぶ．このブロックに接続するブロックの集合よりクラスタC_0を構成する．

S2：C_0と接続を持つ機能ブロックの集合B_0を構成する．

S3：最小カット法によりB_0を2個のクラスタC_1とC_{-1}に分割する．C_1とC_{-1}をそれぞれクラスタC_0の上下に配置する．

S4:C_1,C_{-1}との接続に基づき，まだクラスタ化されていない機能ブロックをB_1，B_{-1}に分割する．

S5：B_1とB_{-1}から，それぞれ残りのクラスタを$C_2,\cdots,C_i,C_{-2},\cdots,C_{-j}$の順に構成する．

4．実験結果

フェーズ1の論理回路を分割するクラスタリングアルゴリズムをSun3/60M(3MIPS)上でC言語を用いて実現した．実験は表1に示すような2個の実データに対して行い，文献[2]の手法との比較を行った．結果を表2に示す．同程度の解を約30％減の計算時間で求めており，本手法が有効であることがわかる．

5．あとがき

セル敷き詰め型ゲートアレイの配置問題に対して，高さが不均一な機能ブロックの集合を扱うことができる配置手法を提案した．現在，フェーズ2以降のアルゴリズムを開発中である．

表1　実験データ

	データ1	データ2
ﾁｯﾌﾟｻｲｽﾞ W×H(ｾﾙ)	269×79=21251	362×106=38372
セル使用率(%)	42.2	40.4
ブロック数	1600	4717
ﾌﾞﾛｯｸを構成するｾﾙ数(平均)	5.61	3.29
ネット数	2093	6137
ネットの端子数(平均)	3.01	3.21

表2　実験結果

		従来法	本手法
ﾃﾞｰﾀ1	CPU時間(秒)	30.3	22.7
	総カット数	7713	7716
ﾃﾞｰﾀ2	CPU時間(秒)	121.2	80.8
	総カット数	35604	36043

文献 [1]H.G.Cho et al.:"A heuristic standard cell placement algorithm using constrained multistage graph model",IEEE Trans.CAD,7,11,pp.1205-1214(1988).
[2]C.M.Fiduccia et al.:"A linear-time heuristic for improving network partitions",Proc.19th DAC,pp.175-181(1982).