5X-2

VLSIのブロック配置における矩形重なり除去問題について

大村 道郎　藤井 隆志　菊野 亨　吉田 典可
広　島　大　学

1．まえがき

VLSIレイアウト設計における機能ブロック(以降，ブロックと略称する)の配置の決定は通常，自動配置プログラムによる配置の計算と，その配置からデッドスペースを減らすための人手による編集操作からなる[2]．この編集操作においては，ブロック間に重なりが生じることがある．

本稿では，与えられたブロックの配置に対し，ブロック間の相対位置関係を保持したままブロック間の重なりを除去する問題について議論する．ここでは，ブロックの形状を矩形に限定する．

2．ブロック間の相対位置関係

与えられる初期配置に含まれるブロック(矩形)の集合を $B=\{b_i | 1 \le i \le n\}$ で表す．各 b_i に対して，4つの頂点のx座標及びy座標の値をそれぞれ平均して求まる座標に位置する点を中心と定める．

2つのブロック b_i，b_j の中心がそれぞれ (x_i, y_i)，(x_j, y_j) であるとする．このとき，$x_i \le x_j$ 且つ $y_i \le y_j$ ならば b_j は b_i の右上に（逆に，b_i は b_j の左下に）位置すると言う．右下，左上の位置関係も同様に定義し，これらの位置関係をブロック間の相対位置関係と総称する．

3．重なり除去問題

初期(あるいは，最終)配置において，2つのブロック $b_i, b_j \in B$ の中心の間の距離を $d_I(b_i, b_j)$ ($d_F(b_i, b_j)$) で表す．Bに属する全てのブロックを囲む最小の矩形の面積を S_I (S_F) で表す．初期配置上で重なりを持つブロックの対の集合を $P=\{(b_i, b_j) | b_i, b_j \in B,\ i<j,$ 且つ，b_i と b_j は重なりを持つ$\}$ とする．配置の変更に伴う変化量Lを次式で定義する．

$$L = \sum_{(b_i, b_j) \in P} |d_F(b_i, b_j) - d_I(b_i, b_j)|$$

重なり除去問題Pを次に定義する[1]．

［問題P］　ブロックの集合 $B=\{b_i\}$ の初期配置と2つの定数 c_1，c_2 が与えられる．このとき，次の条件 i)－iii) を満足するBの最終配置を求めよ．

i) 重なりを持たない．

ii) 相対位置関係が保持されている．

iii) 目的関数を $Z=c_1 L + c_2 |S_F - S_I|$ とするとき，Zの値が最小である．

［例1］　図1に示すブロックの集合 $B=\{b_1, b_2, \cdots, b_7\}$ の初期配置，2つの定数の値 $c_1=c_2=1$ について考える．このとき，図2に示す最終配置が求まり，Z=50.8である．

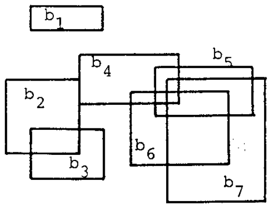


図1　初期配置

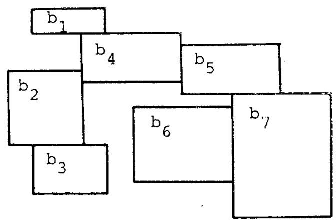


図2　最終配置

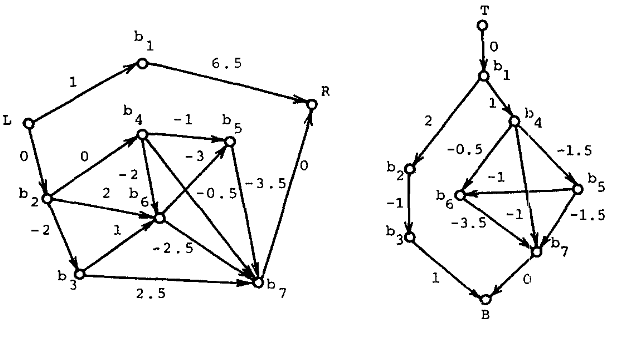
4．配置グラフ

ブロックの配置をx軸，y軸に平行な直線群の上で考え，左右あるいは上下の位置関係で表した配置グラフ $G_x=(V_x, E_x)$，$G_y=(V_y, E_y)$ を導入する[1]．G_x，G_y は重み付き有向グラフであって，各有向枝 $(b_i, b_j) \in E_x$ の重みdはブロック b_i の右辺がブロック b_j の左辺よりも右方向に距離 $|d|$ だけ離れていることを示す．特に，$d<0$ なら，

On the resolution problem of overlapping rectangular blocks in VLSI
Michirou OHMURA, Takashi FUJII, Tohru KIKUNO and Noriyoshi YOSHIDA
Hiroshima University

b_iとb_jが重なりを持つことを意味する．有向枝$(b_i, b_j) \in E_y$についても同様とする．

［例２］　例１で述べたBの初期配置に対する配置グラフG_x，G_yを図３に示す．節点L,R,T,Bはそれぞれチップの左辺，右辺，上辺，下辺に対応している．



(a) G_x　　(b) G_y

図３　配置グラフ

５．重なり除去アルゴリズム

集合PをG_x，G_yに基づいて２つの部分集合P_X，P_Yに分割する．$P_X=\{(b_i, b_j) \in P \mid G_x, G_y$上の$b_i, b_j$間の枝の重みをそれぞれ$d_x, d_y$とするとき，$d_x \geq d_y$である.$\}$とし，$P_Y=P-P_X$とする．

集合Bに属するブロックを中心のx座標の値の非減少順に並べた系列を$B_x=(b_{s(1)}, b_{s(2)}, \cdots, b_{s(n)})$とする．このとき，各ブロック$b_{s(i)}$を４つのタイプA,B,C,Dに分類する．図４は左右方向に関する分類を示している．タイプA,Bは$b_{s(i)}$がブロック$b_{s(i-1)}$と重なりを持たない場合である．タイプCは$(b_{s(i-1)}, b_{s(i)}) \in P_X$，タイプDは$(b_{s(i-1)}, b_{s(i)}) \in P_Y$の場合である．なお，系列$B_y$，上下方向の分類に関しても同様とする．

(a) ﾀｲﾌﾟA　　(b) ﾀｲﾌﾟB

(c) ﾀｲﾌﾟC　　(d) ﾀｲﾌﾟD

図４　ブロックの分類

以上の準備の下に，ヒューリスティックアルゴリズムRを次に与える．詳細は文献[1]を参照されたい．

ステップ１：配置グラフG_x，G_yと集合P，P_X，P_Yを構成する．

ステップ２：ブロックの系列$B_x=(b_{s(1)}, b_{s(2)}, \cdots, b_{s(n)})$を求める．

ステップ３：各$b_{s(i)}$ $(1 \leq i \leq n)$を４つのタイプに分類する．

ステップ４：系列B_x中の順番に，各ブロック$b_{s(i)}$のx方向の移動量$\Delta x_{s(i)}$を$b_{s(i)}$のタイプに基づいて計算する．

ステップ５：ステップ２－４をy方向について行い，$\Delta y_{s(i)}$を計算する．

ステップ６：各ブロックb_iの中心の座標(x_i, y_i)を$(x_i+\Delta x_i, y_i+\Delta y_i)$に変換する．

アルゴリズムRの時間計算量は$O(n^2)$である．既に，日本・データゼネラル社のECLIPSE MV/4000上でＣ言語を用いて実現している．図５の初期配置(n=50)に対して，CPU時間 2.6(秒)で，図６に示す最終配置を得た．

現在，ブロックの形状を一般のマンハッタン図形に拡張した場合の重なり除去問題についても考察中である．

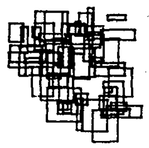


図５　初期配置 (n=50)

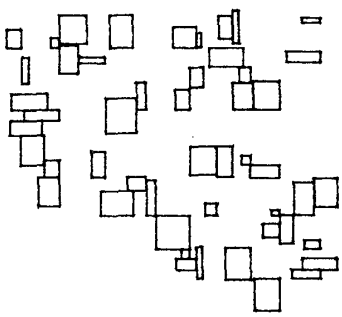


図６　最終配置 (n=50)

謝辞　本研究を進めるに当たりご協力頂いた㈱日本電気 超LSI CAD技術本部 開発部 川西宏博士，西口信行博士に深謝致します．

文献

[1] 大村道郎："矩形重なり除去問題について(3)", ECS Tech. Rep. No.86-07 (1986).
[2] 可児他："超LSI CADの基礎", ｵｰﾑ社 (1983).