設計自動化　40－3
(1987. 12. 17)

モジュールの再利用を考慮したフロアプラン設計手法の提案
－初期フロアプランニング－

A Floorplanning Method for Custom VLSI Chips Including Reused Modules
－Initial Floorplanning Phase－

大村 道郎† 出本 浩† 藤井 隆志† 菊野 亨‡ 吉田 典可†
Michiroh OHMURA Hiroshi IZUMOTO Takashi FUJII Tohru KIKUNO Noriyoshi YOSHIDA

†広島大学 工学部
Faculty of Engineering
Hiroshima University

‡大阪大学 基礎工学部
Faculty of Engineering Science
Osaka University

あらまし　ＶＬＳＩチップのレイアウト設計において，既に設計されているチップのモジュールを再利用することは，設計期間の短縮につながる．本稿では，筆者らにより提案されている階層化フロアプランニング手法における初期フロアプランの段階について，①モジュールを再利用することを前提とし，ハードモジュールの形状を凸ＸＹ多角形に拡張し，②チップ周辺の入出力パッドの位置を考慮した．本手法では，先ず，モジュール間の結合度に応じて初期配置を行う．次に，各モジュールの向きを決定する．最後に，相対位置関係を保存しながら，チップの縦横比を満たし，且つ面積が最小となるように重なりを除去する．

Abstract This paper discusses the initial phase of the floorplanning method proposed by the authors. First, convex rectilinear polygons are introduced in order to treat reused modules. Second, the positions of I/O pads are taken into consideration. In our method, a placement of modules is obtained based on the connectivity between modules. Then, orientations of modules are determined. Finally, overlaps of modules are resolved preserving relative positions, satisfying the aspect ratio, and minimizing the chip area.

1．まえがき

近年の集積度の増大に伴い，ＶＬＳＩは特定用途向け，すなわちＡＳＩＣ化の傾向が強くなってきた．ＡＳＩＣでは，生産量は少ないが，多くの品種を開発する必要があり，信頼度の高いＶＬＳＩを短期間に，安いコストで設計するＣＡＤ技術が望まれる[6]．

設計期間の短縮，更には設計コストの低減につながる１つの方法として，新しく作るチップの一部に，既存チップの一部を再利用することが挙げられる．一般に，再利用する部分は矩形になるとは限らないので，形状の制約を緩めた設計手法が必要となる．

一方，レイアウト設計の最初に行われる配置設計に，フロアプランニングと言う重要なステップがある．フロアプランニングとは，チップ上を幾つかの領域に分割し，各領域内に配置すべきモジュールの集合を決定することを言い，その結果をフロアプランと呼ぶ．フロアプランの良し悪しは，最終的なチップの面積，性能などに大きく影響する[1][4]．

本稿では，筆者らによって既に提案されている階層化フロアプランニング手法[3]の最初のフェーズ(初期フロアプランニング)について，モジュールの再利用を考慮し，以下の点で拡張を試みた．

Ｅ１：比較的大規模なハードモジュールの形状として凸ＸＹ多角形を許す．

Ｅ２：チップの外部との接続に用いられる入出力パッドの位置が，当初に与えられることを考慮する．

本研究における初期フロアプランニングでは，モジュール間の接続数と，チップの形状，面積を特に考慮

している．提案するアルゴリズムの流れを概説する．先ず，モジュール間の接続数に基づいた結合度を計算し，各モジュールの初期配置を行う．次に，各モジュールについて，その向きを決定する．最後に，モジュール間の相対位置関係を保存しながら，チップの縦横比を満たし，且つ面積が最小となるように重なりを除去して，初期フロアプランを得る．

本稿では，２.で初期フロアプランニングの概要を説明し，３.でモジュールの初期配置について述べる．次に４.でモジュールの向きの決定について説明し，５.で重なり除去について述べる．最後に６.で今後の課題について述べる．

２．初期フロアプランニング

２.１　再利用モジュールの形状

既に設計の終っているチップ上の幾つかのモジュールについて，モジュール間の配線領域も含めた配置の形状を，そのまま新たなチップ設計に再利用することを考える．ここでは，再利用される既存チップの一部(幾つかのモジュールと，それらの間の配線領域からなる)の形状を以下に定義する凸ＸＹ多角形に限定した１つのモジュールとして取り扱う．

[定義１] Ｘ軸，Ｙ軸に平行な線分のみから構成されている単純な多角形(以降，ＸＹ多角形と呼ぶ)Ｐに対して次の条件が成立するとき，図形Ｐを凸ＸＹ多角形と呼ぶ．

(条件１) 図形Ｐを同時に３つ以上のＸＹ多角形に分割するＸ軸，Ｙ軸に平行な１本の直線が存在しない(図１参照)．　□

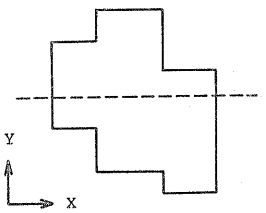


図１　凸ＸＹ多角形

２.２　論理回路

チップ上に実現される回路は，形状の決まった比較的規模の大きいハードモジュール(以降，Ｈモジュールと呼ぶ)の集合$\mathbb{M}h=\{Mh_i\}$と，形状が未定のままの小規模なソフトモジュール(Ｓモジュール)の集合$\mathbb{M}s=\{Ms_i\}$から構成される．Ｓモジュールの集合$\mathbb{M}s$は機能分割に基づいて幾つかのグループに分割されており，各グループに含まれる複数のＳモジュールを，仮想的に正方形の形状を持つ１つの大規模なＳモジュールと見なすことができる[3]．これを仮想モジュール(Ｖモジュール)と呼び，その集合を$\mathbb{M}v=\{Mv_i\}$とする．

本稿で考察する初期フロアプランニングでは，上述のＨモジュールとＶモジュールに分類されるモジュールの集合を対象とする．すなわち入力の一部であるモジュールの集合は，$\mathbb{M}=\mathbb{M}h\cup\mathbb{M}v$と表される．$Mh_i$に対しては①形状(凸ＸＹ多角形)，②面積，③端子位置が，また，Mv_iに対しては①形状(正方形)，②面積がそれぞれ与えられる．但し，Ｖモジュールの端子位置は正方形の中心に代表させる．さらに，入出力パッドの集合$\mathbb{M}e=\{Me_i\}$を含めたモジュールとパッドの集合を$\mathbb{M}^+(=\mathbb{M}\cup\mathbb{M}e)$とする．

集合$\mathbb{M}^+$に属するモジュール間の配線に関する結線要求は，ネットリスト$\mathbb{N}=\{n_i\}$によって与えられる．各ネットn_iは互いに接続すべき端子の集合によって表される．但し，多端子ネットは，文献[5]の方法で２端子ネットに変換し，以降は全て２端子ネットとする($n_i=\{t,t'\}$)．

ここでは，論理回路Ｌをモジュールとパッドの集合$\mathbb{M}^+$とネットリスト$\mathbb{N}$よりなる２項組$L=(\mathbb{M}^+,\mathbb{N})$として表す．

[例１] 論理回路$L=(\mathbb{M}^+,\mathbb{N})$の例を図２に示す．同図中の実線はネットを，外枠の破線は仮想のチップ領域を，そして"○"は端子，または入出力パッドを表す．　□

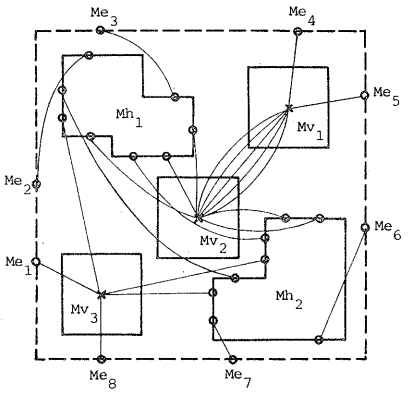


図２　論理回路Ｌ

２.３　モジュールの結合度

初期フロアプランニングでは，モジュール$M(\in\mathbb{M}^+)$間の結合度に基づいて配置を行う．以下，この結合度について説明する．

先ず，論理回路$L=(\mathbb{M}^+,\mathbb{N})$に対して，次のような無向グラフ$G=(V,E)$(以降，結合度グラフと呼ぶ)を導入する．

ｉ) $V=\{M\mid M\in\mathbb{M}^+\}$,

ii) $E=\{(M_i,M_j)\mid M_i,M_j\in M^+$，且つ$N_{ij}\subset N\}$，ここで，$N_{ij}$は論理回路Lにおけるモジュール$M_i$，$M_j$間のネットの集合である．

iii) 各無向枝$(M_i,M_j)\in E$には，重み$w(M_i,M_j)$が付けられる．その値は次式で与えられる．

$$w(M_i,M_j)=\frac{1}{|N_{ij}|}$$

[定義2] 結合度グラフ$G=(V,E)$上のモジュールM_i，$M_j$$(i\neq j)$に対して，結合度$b_{ij}$を$M_i$，$M_j$間の最短パスの長さ(パス上の枝の重みの総和)の逆数と定義する．□

任意のモジュール間の結合度b_{ij}を行列で表したとき，それは上三角行列となる．この行列を結合度行列Bと呼ぶ．

[例2] 図2の論理回路Lに対する，結合度行列Bの一部(入出力パッドを除いたモジュールのみ)を表1に示す．□

表1 結合度行列B

	Mh_1	Mh_2	Mv_1	Mv_2	Mv_3
Mh_1		4.03	1.88	3.00	1.33
Mh_2			1.43	2.00	2.00
Mv_1				5.03	0.83
Mv_2		0			1.00
Mv_3					

2.4 アルゴリズムの概要

初期フロアプランニングにおけるモジュールの配置では，以下に示す2点を考慮する．

a) 相対的に接続数の多いモジュール同士は近くに，少ないものは遠くに配置する（結合度に応じた距離に配置する）．

b) 全てのモジュールを囲む最小の矩形は，与えられたチップの縦横比を満たし，且つ面積が小さくなるようにする．

上述のa)，b)を考慮した初期フロアプランニングのアルゴリズムIFの概要を以下に示す．入力として，モジュールの集合$M=Mh\cup Mv$，ネットリストN，チップの縦横比，及び入出力パッドの集合Meとその位置が与えられ，チップの重なりを持たないモジュールの配置(初期フロアプラン)を求める．

[アルゴリズムIF]

Phase1：(モジュールの初期配置)

モジュール間の結合度に応じて，モジュールの初期配置を決定する(アルゴリズムIP)．

Phase2：(Hモジュールの向き決定)

Phase1の初期配置における各HモジュールMh_iに形状を与え，その向きを端子間のマンハッタン距離の総和が最小となるように決定する．Vモジュールは面積に応じた正方形の形状を与える(アルゴリズムDI)．

Phase3：(重なり除去)

Phase2の出力におけるモジュール間の重なりを，相対位置関係を保存しながら(5.1で説明)，チップの縦横比を満たし，且つ面積が最小となるように除去する(アルゴリズムRO)．

Phase1，Phase2，Phase3については，それぞれ**3.**，**4.**，**5.**で詳しく述べる．

[例3] 図2の論理回路Lで与えられる入力に対して，アルゴリズムIFを適用した結果(初期フロアプラン)を図3に示す．□

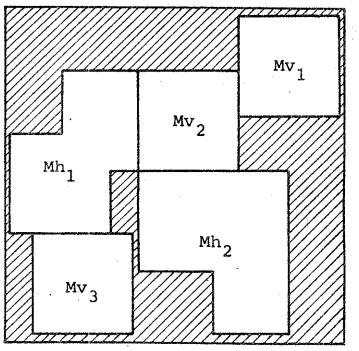


図3 初期フロアプラン

3. モジュールの初期配置

3.1 結合度に基づくモジュール間の距離

2.3で定義した結合度の逆数に定数Cをかけることにより，任意の2つのモジュール$M_i\in M,M_j\in M^+$間の理想的な距離dp_{ij}を計算する．定数Cは，モジュールの面積の総和，チップの縦横比，及びモジュールの数を考慮して決定される．モジュールの配置における任意のモジュール間のユークリッド距離をdr_{ij}とすると，ここではdp_{ij}とdr_{ij}の差の総和が最小となるような配置が望ましいと考える．任意のモジュール間の理想距離dp_{ij}を行列で表したとき，それは上三角行列となる．この行列を理想距離行列(Dp=$[dp_{ij}]$)と言う．

[例4] 図2の論理回路における，理想距離行列Dpの一部(入出力パッドを除いたモジュールのみ)を表2に示す．□

表2　理想距離行列D_p

	Mh_1	Mh_2	Mv_1	Mv_2	Mv_3
Mh_1		0.35	0.38	0.24	0.54
Mh_2			0.50	0.36	0.36
Mv_1				0.14	0.86
Mv_2		0			0.72
Mv_3					

3.2　初期配置問題IP

与えられたチップの縦横比を満たし，単位面積を持つ矩形を，仮想チップ領域と呼ぶことにする．モジュール$M_i \in \mathbb{M}$の各頂点の座標の平均値を$\mathbb{M}$の中心と呼ぶ．尚，モジュールの形状が凸XY多角形で各頂点の平均値がモジュールの外に出る場合は，その点と最も近いモジュールの頂点をその中心とする．この問題では，モジュールをこの中心で代表させる．

モジュールの初期配置問題IPの定式化を与える．

[問題IP]　入力として，①モジュールの集合$\mathbb{M}$，②理想距離行列$Dp=[dp_{ij}]$，③仮想チップ領域A，④仮想チップ周上の入出力パッド$Me_i \in Me$とその位置が与えられる．このとき，以下に示す目的関数Zの値が最小となるようなA内に存在するモジュールの配置を求めよ．

$$Z = \sum_{M_i \in \mathbb{M}, M_j \in \mathbb{M}^+} |dp_{ij} - dr_{ij}|$$

但し，dr_{ij}は配置結果におけるモジュール$M_i \in \mathbb{M}$，$M_j \in \mathbb{M}^+$間のユークリッド距離である．　□

3.3　アルゴリズムIP

3.2で述べた問題IPに対するヒューリスティック解法として，初期配置を逐次改良していく反復改良法[6]を用いる．初期配置には単純に乱数を発生させて配置を行うランダム配置法[6]を採用する．反復改良法には，入力で与えられる理想距離とのずれを小さくするような改良を行う方法をとる．

アルゴリズムIPの概要を以下に示す．

[アルゴリズムIP]

Step1：(初期配置)

ランダム配置法により，仮想チップ領域A内にモジュールを配置する；

Step2：$i \leftarrow 1$；

Step3：(逐次改良)

モジュール$M_i \in \mathbb{M}$と他の全てのモジュール$M_j \in \mathbb{M}^+$間のユークリッド距離dr_{ij}を求め，それぞれについて理想距離とのずれ$dp_{ij} - dr_{ij}$を計算する(但し，M_iの座標を(x_i, y_i)，M_jの座標を(x_j, y_j)で表す)；

Step4：モジュールM_iを次に示す新座標(x_i', y_i')に移動する；

$$x_i' = x_i + \frac{\sum_{M_i \in \mathbb{M}, M_j \in \mathbb{M}^+} (dp_{ij} - dr_{ij}) * (x_j - x_i) / dr_{ij}}{|\mathbb{M}| - 1}$$

$$y_i' = y_i + \frac{\sum_{M_i \in \mathbb{M}, M_j \in \mathbb{M}^+} (dp_{ij} - dr_{ij}) * (y_j - y_i) / dr_{ij}}{|\mathbb{M}| - 1}$$

Step5：$i \leq |\mathbb{M}|$ならばStep3へ；

$i > |\mathbb{M}|$ならば終了；

3.4　適用例

図2の論理回路$L=(\mathbb{M}^+, \mathbb{N})$に対する，アルゴリズムIPの適用例を図4に示す．図はアルゴリズムIPを10回適用した結果である．"●"は各モジュールの中心座標を，外枠の破線は仮想チップ領域Aを表す．ここで，ランダム配置法で求めた初期配置での目的関数Zの値は12.8であり，図4の配置では10.3である．この結果から，アルゴリズムIPの適用によって，モジュール間の距離が理想距離に近づいていることが分かる．

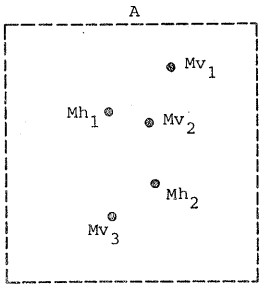


図4　モジュールの初期配置

4．モジュールの向きの決定

4.1　モジュールの向き

Phase1の結果である仮想チップ領域Aを$\mathbb{M}$に属するモジュールの面積の総和に等しい大きさに拡大する．新しく求まったモジュールの配置位置に，実際のモジュールの形を与える．このとき，Hモジュールは凸XY多角形で，且つ端子位置も与えられているので，その向きを決める必要がある．向きは端子位置を考慮して決定すべきであると考えられるが，全てのHモジュールを一度に決めることは時間的に困難である．本手法では，各HモジュールMh_iごとに，それ自身の端子位置のみを考慮し，他のモジュールはHモジュールも

含めて１点であるとして向きを決定する．

向きは，上，下，左，右，表，裏をの８通りが考えられ，Mh_iの端子と直接接続のあるモジュール及び入出力パッドとのマンハッタン距離の総和が最小となるものを採用する．尚，形状を与えて求まったモジュールの配置を$L(M)$とする．

４．２　アルゴリズムDI

２．４で述べたアルゴリズムＩＦにおけるPhase2の詳細をアルゴリズムDIとして以下に示す．

[アルゴリズムDI]

Step1：(仮想チップ領域Ａの拡大)

各モジュールの配置(中心の位置)を含んだ仮想チップ領域ＡをＭに属するモジュールの面積の総和に等しい大きさに拡大する．

Step2：(Ｈモジュールの向きの決定)

Mh_iをPhase1で決定した位置に配置し，８通りの向きのうち，接続のある他のモジュール及び入出力パッドとのマンハッタン距離の総和が最小となる向きに決定する．これを全てのMh_iについて行う．

Step3：(形状を持たせたＶモジュールの配置)

正方形の仮想的な形状を与えたMv_iを，Phase1で決定した位置に配置する．

４．３　適用例

図４の入力に対するアルゴリズムDIの出力を図５に示す．

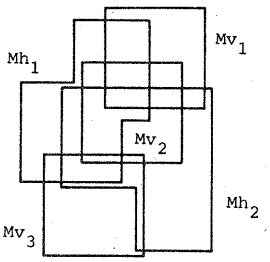


図５　アルゴリズムDIの出力

５．重なり除去

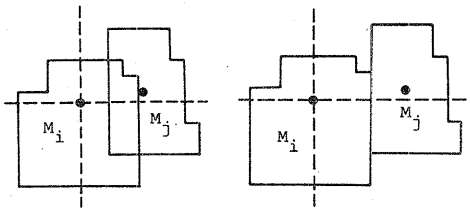
５．１　相対位置関係の保存

Phase2のモジュールの配置結果には，一般に，重なりが存在する．この重なりを除去する際には，除去前のモジュール間の相対的な位置を保存することが望ましい[2]．相対位置関係の保存を次のように定義する．

[定義３] モジュールの集合Ｍの配置$L_1(M)$に含まれるモジュールに対する操作として，ここでは平行移動だけを許す．その結果として，新しい配置$L_2(M)$が決まったとする．このとき，次のｉ)，ii)が成立するなら，$L_2(M)$は$L_1(M)$における相対位置関係を保存していると言う．

ｉ) $L_1(M)$上において重なりを持つ任意のモジュール対M_i，M_jに対し，それぞれの中心のＸ座標，及びＹ座標に関する大小関係が$L_2(M)$上でも成立する(図６)．

ii) $L_1(M)$上で左右(上下)に隣接する任意のモジュール対M_i，M_jに対し，中心のＸ座標(Ｙ座標)に関する大小関係が$L_2(M)$上でも成立する．□



(a)　$L_1(M)$　　(b)　$L_2(M)$

図６　相対位置関係の保存

５．２　制約条件

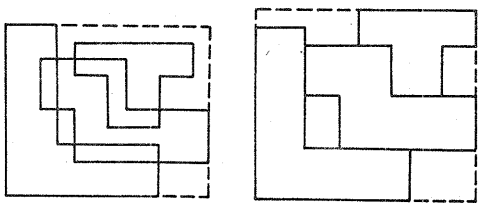
ここでは，重なりを除去する際の制約条件について述べる．以下に示すａ)～ｃ)が制約条件である．尚，重なり除去前の配置を配置$L_I(M)$，除去後の配置を配置$L_F(M)$とする．$L_I(M)$，$L_F(M)$を囲む最小の矩形(以降，チップ領域Ｃとする)を，それぞれ，$C(L_I(M))$，$C(L_F(M))$とする(図７)．また，$C(L_I(M))$及び$C(L_F(M))$の高さと幅をそれぞれh_I, w_I，及び，h_F, w_Fで表し；面積を$S(L_I(M))$，$S(L_F(M))$で表す．更に，入力で与えられる縦横比をａｒで表す．

ａ) $L_F(M)$は$L_I(M)$における相対位置関係を保存する．

ｂ) $C(L_F(M))$が入力で与えられる縦横比ａｒを満たす．

ｃ) $S(L_F(M))$が最小となる．

ここでｂ)はチップ形状に関する条件で，ｃ)は面積最小化に関する条件である．



(a)　$C(L_I(M))$　　(b)　$C(L_F(M))$

図７　モジュールを囲む最小矩形

以上，a)～c)までの条件を満たすように，モジュール間の重なりを除去する問題の定式化を，5.3で行う．

5.3　重なり除去問題RO

重なり除去問題ROの定式化を行う．

[問題RO]　入力として，①凸XY多角形の形状をしたモジュールの集合M，②モジュール間に重なりを持つ配置$L_I(M)$，③チップの縦横比arが与えられるとき，以下に示す条件を満足するモジュールの配置$L_F(M)$と，それを囲むチップ領域Cを求めよ．

i) $L_F(M)$上の任意のモジュールM_i, $M_j \in M$ $(i \neq j)$は重なりを持たない．

ii) $L_F(M)$は$L_I(M)$における相対位置関係を保存する．

iii) $C(L_F(M))$の縦横比ar_Fに対し，$|ar_F - ar| \leq \gamma$が成り立つ．但し，γは定数である．

iv) $S(L_F(M))$が最小である．　□

5.4　アルゴリズムRO

5.3の重なり除去問題ROに対するヒューリスティックアルゴリズムROについて述べる．準備として，配置グラフG_x, G_y，及び重なりを持つモジュール対の集合Pの説明を(A)で行い，(B)でアルゴリズムROを示す．

(A) 準備

(1)配置グラフG_x, G_y

モジュールはX軸，Y軸と平行に配置するとし，左右あるいは上下の位置関係を表す配置グラフ$G_x=(V_x, E_x)$, $G_y=(V_y, E_y)$を導入する．G_x, G_yは重み付き有向グラフであり，$V_x = M \cup M_L \cup M_R$, $V_y = M \cup M_T \cup M_B$, $E_x = \{(M_i, M_j) \mid M_i, M_j \in M$, [$M_i$は$M_j$の左に隣接している，あるいは，$M_i$と$M_j$は重なりを持つ]$\} \cup \{(M_L, M_i) \mid M_L$は$M_i$の左に隣接している$\} \cup \{(M_j, M_R) \mid M_j$は$M_R$の左に隣接している$\}$．$E_y$についても同様とする．尚，$M_L$, M_R, M_T, M_Bは仮想的なモジュールである．

有向枝$(M_i, M_j) \in E_x$($(M_i, M_j) \in E_y$)の重みを$w_x(M_i, M_j)$($w_y(M_i, M_j)$)とし，隣接しているモジュールM_i, M_jに対してはX軸(Y軸)方向のモジュールM_i, M_j間の最小距離をその値とする．このときの$w_x(M_i, M_j)$($w_y(M_i, M_j)$)は正の値とする．また，重なりを持つM_i, M_j間の重みの値は，X軸(Y軸)方向にモジュールを平行移動して重なりが除去できる最小の距離とする．このときの$w_x(M_i, M_j)$($w_y(M_i, M_j)$)は負の値とする．

[例5]　図6のモジュール配置に対する配置グラフG_x, G_yを図8に示す．　□

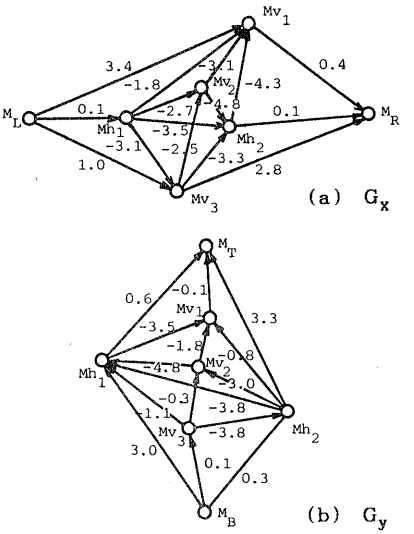


図8　配置グラフ

(2)集合P

集合Pは，配置$L_I(M)$における，重なりを持つモジュール対$P(M_i, M_j)$(但し，$M_i, M_j \in M$)の集合である．モジュール間の重なりは，X方向への平行移動またはY方向への平行移動のどちらか一方によって除去する．ここでX方向への移動で重なりを除去するモジュール対の集合をP_x，Y方向への移動で重なりを除去するモジュール対の集合をP_yとする($P = P_x \cup P_y$)．

今，配置$L_I(M)$において，Pの2つの部分集合PI_x, PI_y($P = PI_x \cup PI_y$)を次のように定める．

$$PI_x = \{(M_i, M_j) \in P \mid |w_x(M_i, M_j)| \leq |w_y(M_i, M_j)|\},$$

$$PI_y = P - PI_x$$

以下，これらを直観的に説明する．モジュール対$(M_i, M_j) \in PI_x$に対しては，モジュールM_iあるいはM_jをX方向に移動する方が，Y方向に移動するよりもモジュールの移動量が少なくて重なりを除去できる．逆に$(M_i, M_j) \in PI_y$ならば，モジュールM_iあるいはM_jをY方向に移動すれば少ない移動量で重なりが除去できる．

(B) アルゴリズムRO

最初に，アルゴリズムROの基本方針について説明する．アルゴリズムROでは，先ず$P_x = PI_x$, $P_y = PI_y$として重なりを除去する．しかし，この方法では一般に重なり除去後の$C(L_F(M))$の縦横比ar_Fが$|ar_F - ar| \leq \gamma$を満たすとは限らない．そこで，重なり除去後の全モジュールを囲む最小の矩形の縦横比ar_Fが，$ar_F < ar$ならば，P_xに属するモジュールの幾

つかを$\mathbb{P}_y$に入れて重なりを除去する．$ar_F > ar$ならば，$\mathbb{P}_y$に属するモジュールの幾つかを$\mathbb{P}_x$に入れて重なりを除去して改良する．このとき，$\mathbb{P}I_x$(あるいは$\mathbb{P}I_y$)のどの要素を$\mathbb{P}I_y$($\mathbb{P}I_y$)に入れるかが問題となる．

今，$ar_F < ar$の場合を考える．要素の入換えにより求まったC($L_F(\mathbb{M})$)の縦の長さを$h_F = h_I + \beta$ ($0 \leq \beta$)，横の長さを$w_F = w_I - \alpha$ ($0 \leq \alpha < w_I$)とする．このとき$\beta \geq \alpha$が成立する．ここでは，$S(L_F(\mathbb{M}))$をなるべく小さくしたいので$\beta = \alpha$の場合が理想である．一般に，

$$\begin{cases} \alpha = \beta, \\ \dfrac{h_I + \beta}{w_I - \alpha} = ar \end{cases}$$

を満たすような(複数の)要素の入換えが必要となる．$ar_F > ar$の場合も同様である．アルゴリズムROでは，X方向の重なりとY方向の重なりの差が小さなものから優先的に入換えを行って縦横比を改良する．

重なり除去問題ROに対するヒューリスティックアルゴリズムROの概要を示す．ここで，配置グラフの構成，及び各モジュールの移動量の計算については，詳細を文献[2]に譲る．

[アルゴリズムRO]

Step1：配置$L_I(\mathbb{M})$に対する配置グラフG_x，G_yを構成し，同時に重なりを持つモジュール対の集合$\mathbb{P}$を求める；

Step2：G_x，G_yより，$\mathbb{P}$を$\mathbb{P}I_x$，$\mathbb{P}I_y$に分割し，$\mathbb{P}_x \leftarrow \mathbb{P}I_x$，$\mathbb{P}_y \leftarrow \mathbb{P}I_y$とする；

Step3：$\mathbb{P}_x$に属するモジュール対(ここでは，$P_{x(k)}$と表す)を，Y方向の重なりとの差により非減少順にソートし，$\mathbb{P}_x$上の系列S_xを求める；$\mathbb{P}_y$(=$\{P_{y(k)}\}$)に対しても同様に，$\mathbb{P}_y$上の系列S_yを求める；

Step4：$L_I(\mathbb{M})$の重なりを除去し，$ar_F(=h_F/w_F)$を求める(手続きRemove)．

Step5：(改良の必要性の判定)

i)$|ar_F - ar| \leq \gamma$ならば，終了(改良不要)；

ii)$|ar_F - ar| > \gamma$且つ$ar_F < ar$ならば(X方向の縮小による改良)，

Case←X，$m \leftarrow |\mathbb{P}_x|$，$n \leftarrow |\mathbb{P}_y|$，$k \leftarrow 1$；

iii)$|ar_F - ar| > \gamma$且つ$ar_F > ar$ならば(Y方向の縮小による改良)，

Case←Y，$m \leftarrow |\mathbb{P}_y|$，$n \leftarrow |\mathbb{P}_x|$，$k \leftarrow 1$；

Step6：($\mathbb{P}_x$(または$\mathbb{P}_y$)の変更)

CaseX：S_xのk番目の要素に対応する$P_{x(k)}$を$\mathbb{P}_x$から除去して$\mathbb{P}_y$に入れる；

CaseY：S_yのk番目の要素に対応する$P_{y(k)}$を$\mathbb{P}_y$から除去して$\mathbb{P}_x$に入れる；

Step7：$L_I(\mathbb{M})$の重なりを除去し，$ar_F(=h_F/w_F)$を求める(手続きRemove)；

Step8：(縦横比のチェック)

i)$k \leq m$のとき，

CaseX：$ar_F > ar$ならば，$\mathbb{P}_y$に入れた$P_{x(k)}$を除去し，再び$\mathbb{P}_x$に戻す；　$k \leftarrow k+1$としてStep6へ；

CaseY：$ar_F < ar$ならば，$\mathbb{P}_x$に入れた$P_{y(k)}$を除去し，再び$\mathbb{P}_y$に戻す；　$k \leftarrow k+1$としてStep6へ；

ii)$k > m$のとき，

1)$|ar_F - ar| \leq \gamma$ならば，終了；

2)$|ar_F - ar| > \gamma$ならば，モジュールを余分に移動することによって$|ar_F - ar| = \gamma$とし，終了；

[手続きRemove]

Step1：$\mathbb{P}_y$に属するモジュール対(M_i，M_j)に対し，G_x上の$w_x(M_i, M_j)$の値を0に変換し，変換後のグラフをG_x'で表す；

Step2：G_x'上でM_Lから各M_iへの最短パスの長さを基にして，各モジュールM_iのX方向の移動距離を決定する；

Step3：各モジュールをStep2で決めた距離だけX方向に移動し，新しく求まった配置を$L_T(\mathbb{M})$で表す；

Step4：$L_T(\mathbb{M})$に対して配置グラフG_y'を構成する；

Step5：G_y'上でM_Bから各M_iへの最短パスの長さを基にして，各モジュールM_iのY方向の移動量を決定する；

Step6：各モジュールM_iをStep5で求めた距離だけY方向に移動して，配置$L_F(\mathbb{M})$を求める；

5.5　適用例

図5のモジュールの配置，チップの縦横比$ar = 1$を入力とするとき，アルゴリズムROの出力は図3に示す通りである．

6．あとがき

本稿では，モジュールの再利用を考慮したフロアプラン設計における，初期フロアプランの手法を提案した．今後は，各段階で用いるアルゴリズムの開発，及び初期フロアプランニング後のフェーズを含めたフロアプランシステムの実現を行う予定である．

謝辞

本稿の作成に当り，献身的なご協力を頂いた本学学部生 横山和俊君に感謝します．

文献

[1] A.R. Newton and A.L. Sangiovanni-Vincentelli :"Computer-aided design for VLSI circuit," Computer, 19, 4, pp.38-60 (1986).

[2] 大村，藤井，菊野，吉田:"ＶＬＳＩレイアウト設計におけるブロック間の重なり除去," 信学技報, COMP86-43 (1986).

[3] 大村，藤井，菊野，吉田:"カスタムＶＬＳＩのための機能分割に基づく階層化フロアプラン設計について," 信学技報, VLD87-56 (1987).

[4] D.P. LaPotin and S.W. Director : "Mason: A global floorplanning approach for VLSI design," IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst. CAD-5, 4, pp.477-489 (1986).

[5] K. Ueda, H. Kitazawa and I. Harada:"CHAMP: Chip floor plan for hierarchical VLSI layout design," IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst. CAD-4, 1, pp.12-22 (1985).

[6] 渡辺，浅田，可児，大附:"ＶＬＳＩの設計Ｉ," 岩波書店 (1985).