THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le titre de

DOCTEUR INGENIEUR

Spécialité : Electronique

par

Philippe TYCHON

CONCEPTION DU DESSIN DES CIRCUITS INTEGRES PAR UNE APPROCHE TOPOLOGIQUE

Soutenue le 19 Avril 1985 devant le Jury composé de :

M. CONSTANT	Président
M. PERNOT	Rapporteur
M. DECARPIGNY M. GOBERT M. JESPERS M. ROLLAND	Examinateurs

REMERCIEMENTS

Je tiens à remercier Monsieur le Professeur E. CONSTANT qui me fait l'honneur de présider le jury de cette thèse et qui m'a soutenu par ses encouragements tout au long de sa réalisation.

Je tiens à exprimer à Monsieur J.M. PERNOT, responsable des circuits intégrés spéciaux chez BULL, ma profonde reconnaissance pour avoir été l'instigateur de ce travail et m'avoir aidé par ses précieux conseils et ses nombreuses suggestions durant la première partie de cette thèse.

Je remercie très vivement les membres du jury,

Monsieur le Professeur P. JESPERS de l'Université de Louvain-La-Neuve, qui a été à l'origine de l'intérêt que je porte au domaine des circuits intégrés,

Monsieur J. GOBERT, responsable de la division "Architecture des Microprocesseurs et V.L.S.I." aux Laboratoires d'Electronique et de Physique Appliquée (LEP), qui m'a permis de mener à bien la présente étude,

Monsieur ROLLAND, Maître de Conférences à l'EUDIL,

Monsieur J.N. DECARPIGNY, directeur des études à l'ISEN.

Enfin, ma reconnaissance va également à mes collègues du LEP et à mes proches qui m'ont offert leur contribution et m'ont soutenu dans la réalisation de cette thèse.

TABLE DES MATIERES

LISTE DES FIGURES

INTRODUCTION

La conception assistée par ordinateur a pris ces dernières années une importance non négligeable pour les fabriquants de circuits intégrés. De nombreux programmes ont été développés qui tentent de faire face à la complexité croissante des circuits. C'est dans ce cadre que cette étude propose une nouvelle approche du dessin des masques : sachant qu'une connexion est une liaison physique reliant deux points du circuit, il s'agit de définir un système tel que, connaissant la topologie des connexions, le passage à un dessin symbolique soit automatique.

Dans le premier chapitre, nous présentons les algorithmes classiques concernant le placement et l'interconnexion des éléments constitutifs des circuits. Les objectifs de la nouvelle approche sont ensuite énoncés.

La nouvelle méthode est décrite dans le second chapitre sans l'appliquer à des cas concrets. Elle comporte deux grandes étapes: la première est une recherche topologique qui permet de définir les positions des connexions les unes par rapport aux autres; la seconde étape génère un dessin symbolique des connexions.

Cette nouvelle méthode est appliquée au problème du "Channel routing", c'est-à-dire à l'interconnexion d'un domaine fermé généralement rectangulaire; le cas bijectif, une connexion relie deux points, et son extension au cas général sont traités. C'est l'objet du troisième chapitre. Dans le dernier chapitre, on présente une façon d'utiliser également cette méthode pour générer automatiquement le dessin d'une cellule MOS.

1.0 LE "LAYOUT" DES CIRCUITS INTEGRES

Depuis plusieurs années, la complexité et la taille des circuits intégrés ne cessent d'augmenter. Ce développement a nécessité la mise en oeuvre de méthodes de travail et d'outils informatiques qui facilitent le travail de conception. Parmi les méthodes de travail, on peut citer en particulier la conception hiérarchique des circuits. Quant aux outils informatiques, ce sont essentiellement des programmes de conception assistée par ordinateur. Ce besoin s'est fait sentir indépendammemt à tous les niveaux, architecture, logique, électrique, "layout", test, mais aussi dans les liaisons entre ces différents niveaux de conception.

La génération du "layout" d'un circuit intégré revient à placer sur une surface, l'ensemble des transistors dimensionnés suivant les résultats des simulations électriques et reliés de manière à réaliser les fonctions logiques désirées. Différentes techniques sont couramment utilisées pour dessiner un circuit intégré. Elles seront rapidement décrites dans le paragraphe 1.1 . Ces méthodes nécessitent l'aide de programmes informatiques qui seront évoqués dans le paragraphe 1.2 .

1.1 LES METHODES DE "LAYOUT"

Toutes les techniques visent un même objectif : optimiser le produit (TC * S) où :

- o TC est le temps de réalisation du layout
- o S est la surface du circuit.

En effet, le dessin manuel permet d'optimiser la surface mais demande un temps de conception considérable. Par contre d'autres méthodes, les prédiffusés, les "gate array", permettent une réalisation rapide au détriment de la surface.

1.1.1 Dessin manuel complet

C'est la méthode la plus simple qui donne les meilleurs résultats en ce qui concerne la surface du circuit. Les masques sont dessinés à l'aide d'un éditeur graphique plus ou moins perfectionné . Mais au vu de l'accroissement du nombre de transistors sur une puce, cette méthode sera de moins en moins utilisée à cause d'un coût de développement trop élevé.

1.1.2 Dessin symbolique

Dans ce cas, le concepteur introduit également manuellement des données; mais il ne génère que la topologie du dessin : on dessine librement des lignes de largeur nulle; ces lignes remplacent les pistes; les transistors et les contacts, représentés par des symboles, sont placés aux intersections de ces lignes (Figure 1 page 5). Une fois la topologie déterminée, un programme compacte le dessin en fonction des règles de la technologie et l'on peut ainsi obtenir un "layout" assez dense. Plusieurs systèmes de ce genre existent : STICK (1), CABBAGE (2).

1.1.3 Cellules standard

On crée une bibliothèque de cellules pour une technologie donnée. Toutes les cellules ont la même hauteur. Chaque cellule réalise une fonction logique et est traversée par au moins deux bus, un bus pour la Masse et un bus pour l'Alimentation de la cellule, placés à la même ordonnée dans toutes les cellules ce qui permet de les accoler deux à deux. En effet, elles sont ensuite assemblées et forment de longues rangées. Un programme d'interconnexion automatique réalise alors les liaisons entre les rangées ou entre cellules d'une même rangée (Figure 2 page 5). C'est une méthode presque entièrement automatique mais qui donne des surfaces assez grandes. On peut citer deux systèmes basés sur cette approche : CALMOS (3), LTX (4).

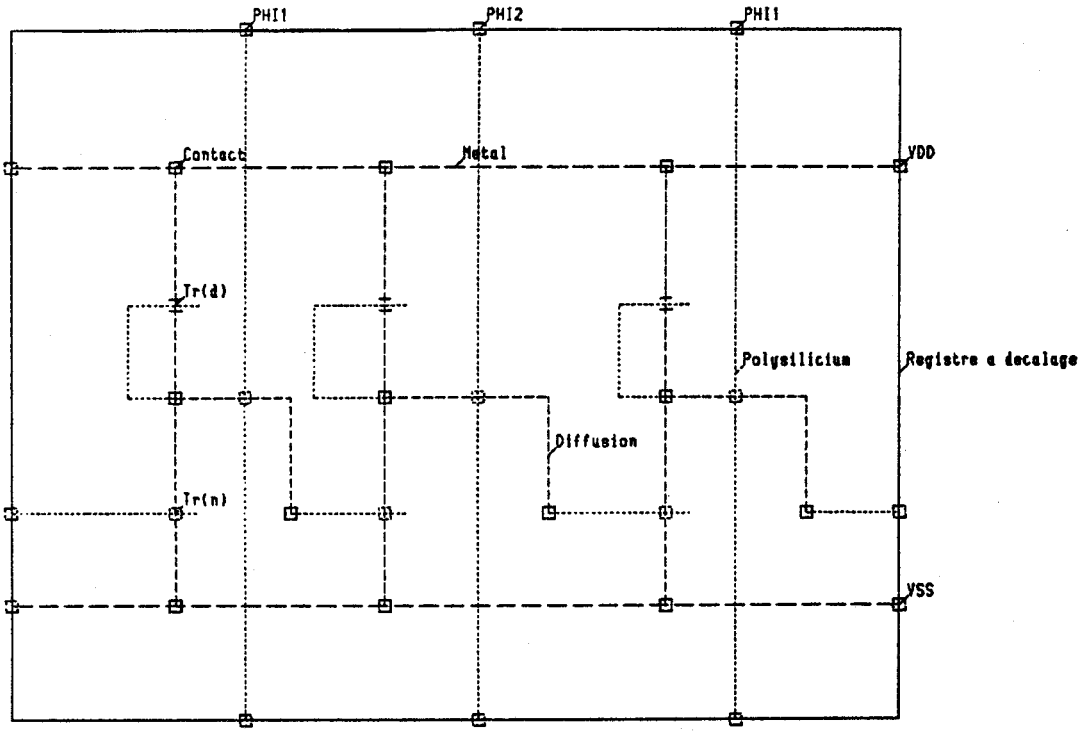


Figure 1. Dessin symbolique

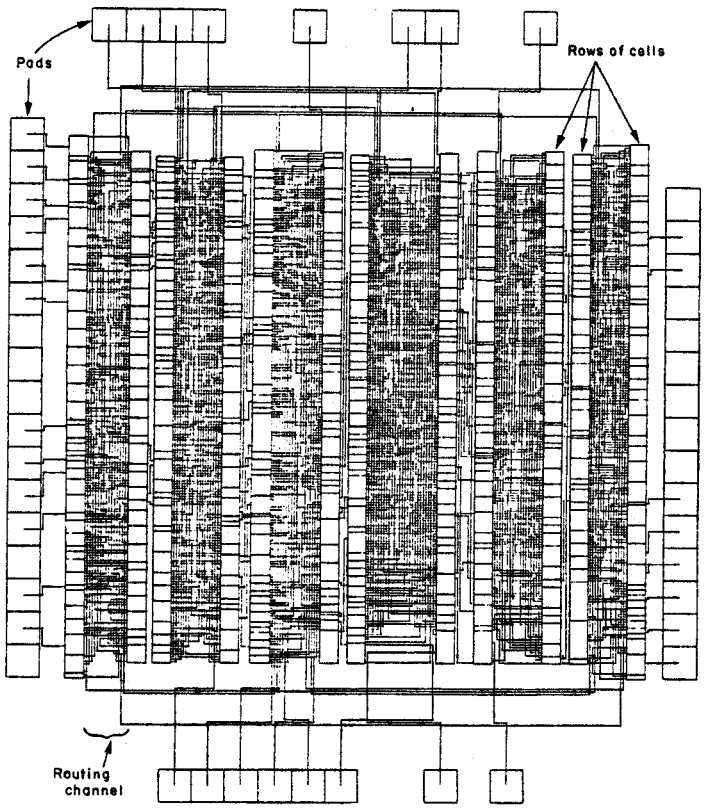


Figure 2. Cellules standard

1.1.4 Compilateurs de silicium

La méthode développée par Johansen s'applique à tous les niveaux de la conception, depuis l'architecture jusqu'au "layout" (5). On n'évoquera que ce dernier. Des cellules sont conçues pour être accolées deux à deux. Elles sont paramétrables : suivant les paramètres entrés, on peut obtenir pour une même cellule différentes vitesses de fonctionnement, différentes puissances consommées. La cellule peut être agrandie suivant une dimension : il est ainsi possible de relier directement les sorties de cette cellule avec les entrées de celle qui lui est accolée. Johansen a appliqué cette méthode pour réaliser la partie opérative, souvent appelée "data path", d'un microprocesseur. Cette technique semble donc très intéressante et pourra sûrement remplacer dans l'avenir le dessin manuel avec de bons résultats.

1.1.5 "Gate array"

Le point de départ de cette méthode est un tableau d'éléments déjà dessinés. Chaque élément réalise une fonction logique assez simple. Leur placement est fixe. Le seul travail à faire est de les interconnecter pour réaliser les fonctions logiques du circuit. Cette méthode est très rapide, par contre la surface n'est pas du tout optimisée.

1.2 GENERALITES SUR LA C.A.O APPLIQUEE AU "LAYOUT"

D'une manière générale, le "layout" d'un circuit intégré se fait actuellement de la manière suivante :

1. on dessine les cellules de base en les optimisant.

2. on assemble ces cellules pour former des blocs compacts: l'unité arithmétique et logique, les registres, des PLA (programmable logic array),de la RAM interne etc... Il reste souvent à dessiner manuellement des parties en lo-

gique dite "sauvage" : leur caractéristique est de ne pas contenir de structures régulières.

3. les blocs doivent ensuite être placés et interconnectés sur le circuit. Un bon placement est un placement qui minimise la longueur des connexions et la surface totale du circuit.

Depuis plusieurs années, des programmes informatiques sont utilisés lors des différentes étapes présentées ci-dessus. Ils visent un double objectif:

1. diminuer le temps et donc le coût de la conception

2. diminuer le coût de la fabrication en minimisant la surface du circuit, ce qui améliore le rendement en fabrication.

Ces outils informatiques vont du simple éditeur graphique au système entièrement automatisé qui réalise un dessin du circuit en fonction des spécifications logiques et électriques. En général, ces systèmes comprennent deux programmes:

- un programme de placement

- un programme d'interconnexion

Différents algorithmes concernant ces deux applications sont présentés aux paragraphes 1-2-1 et 1-2-2.

1.2.1 Algorithmes de placement.

Les algorithmes de placement permettent de positionner les éléments d'un circuit sur une surface rectangulaire. Beaucoup d'algorithmes ont été développés; ils diffèrent suivant le type des éléments à placer, suivant la méthode utilisée et suivant la façon de mesurer la qualité du placement.

Le placement n'est pas le même si l'on traite des "gate arrays", des cellules standard, avec une dimension fixe et l'autre variable, ou des blocs de forme quelconque dans lesquels les deux dimensions peuvent varier. Dans le premier de ces trois cas, seules les permutations de cellules sont possibles puisqu'on utilise une matrice de cellules. Dans le second, le placement détermine la rangée dans laquelle chaque cellule doit être placée et comment les rangées sont placées les unes par rapport aux autres. Un exemple de ce système est LTX (4). Le dernier cas est celui qui donne le plus de liberté mais est également le plus difficile à résoudre.

1.2.1.1 Critères utilisés

Le but du placement est de permettre l'interconnexion de tous les éléments sur la surface la plus petite possible. Il n'a pas encore été possible de le modéliser mathématiquement: à défaut on peut seulement utiliser différents critères pour mesurer la qualité du placement:

- la longueur totale des connexions. En utilisant uniquement ce critère, certaines régions d'interconnexions, encore appelées canaux, en particulier celles occupant le centre du circuit, sont très saturées. On doit donc utiliser aussi d'autres critères.

- la mesure de la "densité" des connexions. L'idée est de quantifier le nombre de connexions par unité de surface et d'identifier les points de congestion éventuelle. Une technique souvent utilisée est de calculer le nombre de connexions rencontrées par une droite fictive traversant

ce circuit suivant une horizontale ou une verticale. On effectue des mesures suivant plusieurs droites. Le but recherché est de minimiser la valeur maximale trouvée. Ceci a été étudié en particulier par Breuer (6). Cette méthode permet de répartir les connexions sur toute la surface du circuit; on évite la saturation de certaines zones.

- la surface totale du circuit; c'est une valeur estimée car il n'est pas toujours possible de déterminer de façon exacte la place occupée par les interconnexions.

1.2.1.2 Méthodes

Recherche exhaustive

Pour chaque bloc, on essaie toutes les positions possibles sur la surface ainsi que toutes les orientations. Cette méthode permet de trouver la meilleure solution possible mais elle entraîne des temps de calcul prohibitifs : si l'on ne tient pas compte des orientations, le nombre de permutations sur un ensemble de N blocs est N! . Cette méthode n'est utilisée que pour des problèmes simples c'est-à-dire pour un nombre de blocs inférieur à 15. Les algorithmes couramment pratiqués sont donc heuristiques.

Placement constructif

Cette méthode s'apparente à la croissance d'un cristal. On place un bloc sur la surface, en général dans un des quatre coins du circuit. L'algorithme recherche ensuite parmi les blocs restants celui qui est le plus connecté avec le bloc déjà placé. Ce bloc est placé dans la meilleure position possible. Ce processus est répété jusqu'à ce que tous les blocs soient placés.

Le critère utilisé pour mesurer la qualité du placement est la longueur des connexions. Le résultat obtenu dépend beaucoup du premier bloc placé. Cette méthode aboutit à un placement qui n'est en général pas satisfaisant, puisque la

position d'un élément donné est choisie seulement en fonction des éléments déjà placés. Le placement constructif a été étudié en particulier par Schweikert (7).

Bipartition

Considérons l'ensemble des blocs à placer. Cet ensemble peut être divisé en 2 sous-ensembles tels que :

- la surface occupée par chacun ait la même taille
- le nombre de connèxions entre les deux ensembles soit minimisé.

Ensuite chacun des deux ensembles est partitionné à son tour de la même manière. Cet algorithme est réitéré jusqu'à ce que les blocs formés soient de petite taille ou soient des blocs de base. Il donne généralement de meilleurs résultats que le placement constructif car il permet une meilleure répartition des connexions sur toute la surface. Il s'inscrit bien également dans une approche hiérarchique de la conception d'un circuit. Cette méthode a été developpée entre autres par Breuer (6).

Recuit simulé.

La méthode d'optimisation par recuit simulé (on parle plus souvent de "simulated annealing") a également été appliquée au placement par Jespen et Gelatt (8). L'algorithme fonctionne par analogie avec le recuit physique d'un matériau. Il permet de placer sur un circuit des blocs de forme rectangulaire dont la taille et le facteur de forme peuvent varier.

Chacun des algorithmes présentés précédemment est généralement suivi d'une phase d'amélioration. Lorsque l'on traite des cellules standard, les changements sont des permutations de cellules au niveau local: on déplace les proches voisines. Dans le cas des blocs rectangulaires de forme quelconque, on enlève les blocs un à un et on les replace dans la meilleure position possible, en général dans un voisinage rapproché de sa position d'origine (9).

Le placement une fois terminé, tous les blocs sont positionnés. Les blocs sont séparés par des canaux d'interconnexion, dont la taille a été estimée en fonction du nombre de fils qui doivent y passer. La Figure 3 page 11 présente un exemple de circuit. Il s'agit maintenant de relier les points qui appartiennent à différents blocs et qui sont sur une même équipotentielle : c'est la phase dite du "routing" ou encore interconnexion.

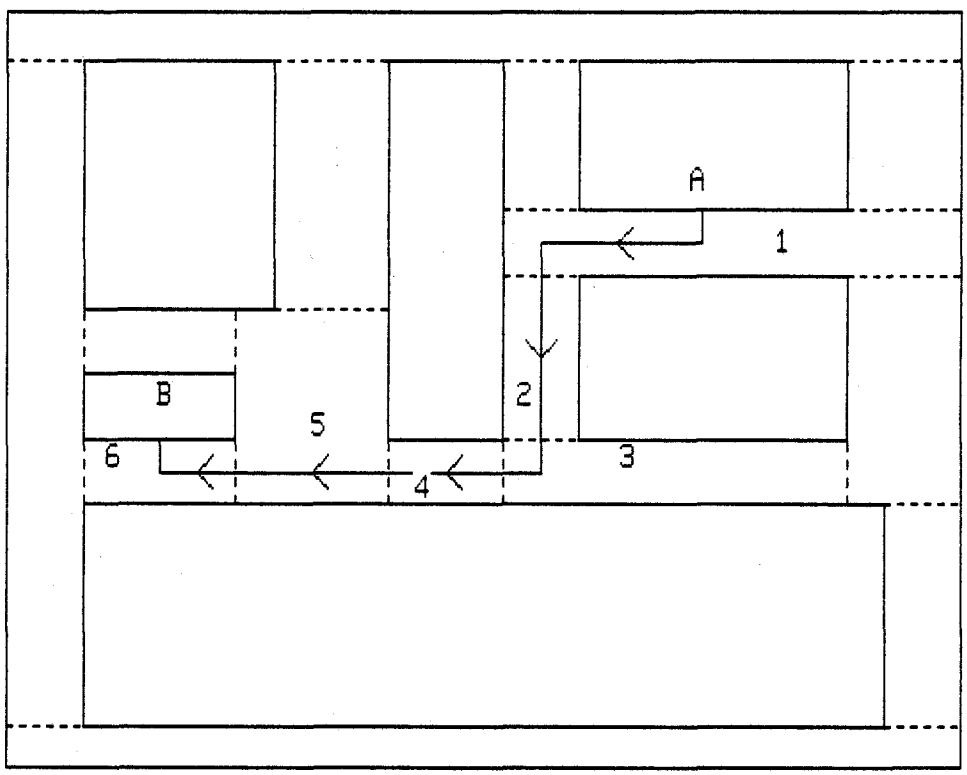


Figure 3. Blocs et canaux d'interconnexion

1.2.2 Algorithmes de "routing"

Actuellement le "routing" est généralement divisé en deux étapes :

1. le "global routing": cette étape prépare le travail de la seconde étape. Elle choisit, pour chaque connexion, les canaux qui vont être traversés.

2. le "channel routing": dans chaque canal, on assigne à chaque connexion sa position exacte.

Il faut rappeler, cependant, que lorsque le niveau de complexité du circuit était plus faible, l'interconnexion se faisait en une seule étape au moyen d'algorithmes tels que ceux de LEE ou HIGHTOWER. Ces méthodes seront d'abord présentées, ensuite on développera plus particulièrement les algorithmes utilisés actuellement pour le "global" et le "channel routing".

1.2.2.1 Algorithme de LEE.

Cet algorithme (10) est le plus souvent utilisé sur une grille rectangulaire, chaque case pouvant être vide, contenir un obstacle fixe, ou une connexion. Pour chercher un chemin entre deux points A et B disposés sur un plan, le processus est initialisé avec l'un des points à relier. Une onde est développée autour du point choisi, A par exemple : un 1 est inscrit dans chaque case vide adjacente à A. Puis un 2 est inscrit dans chaque case vide adjacente aux cases contenant un 1. Et ainsi de suite jusqu' à ce que l'une des deux situations suivantes se produise :

1. le processus se bloque; au Kième pas, on ne trouve plus de cases libres adjacentes à celles contenant K-1. Il n'existe dans ce cas aucun chemin entre A et B.

2. le point B est atteint : le plus court chemin entre A et B est constitué d'une suite de cases adjacentes partant de A et numérotées dans un ordre strictement croissant ; il peut en exister plusieurs.

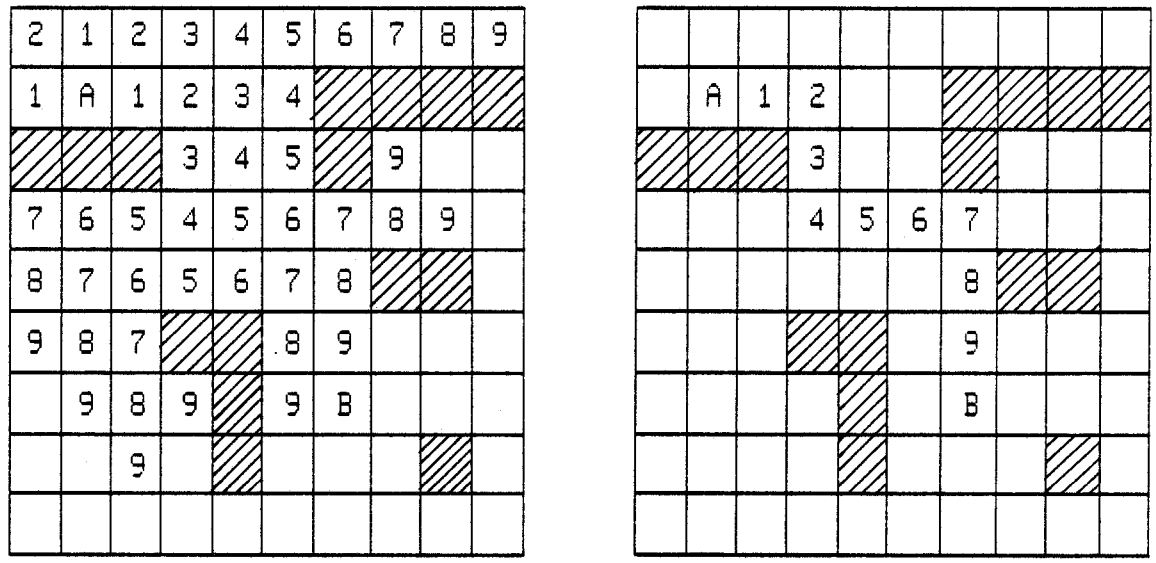


Figure 4. Algorithme de LEE

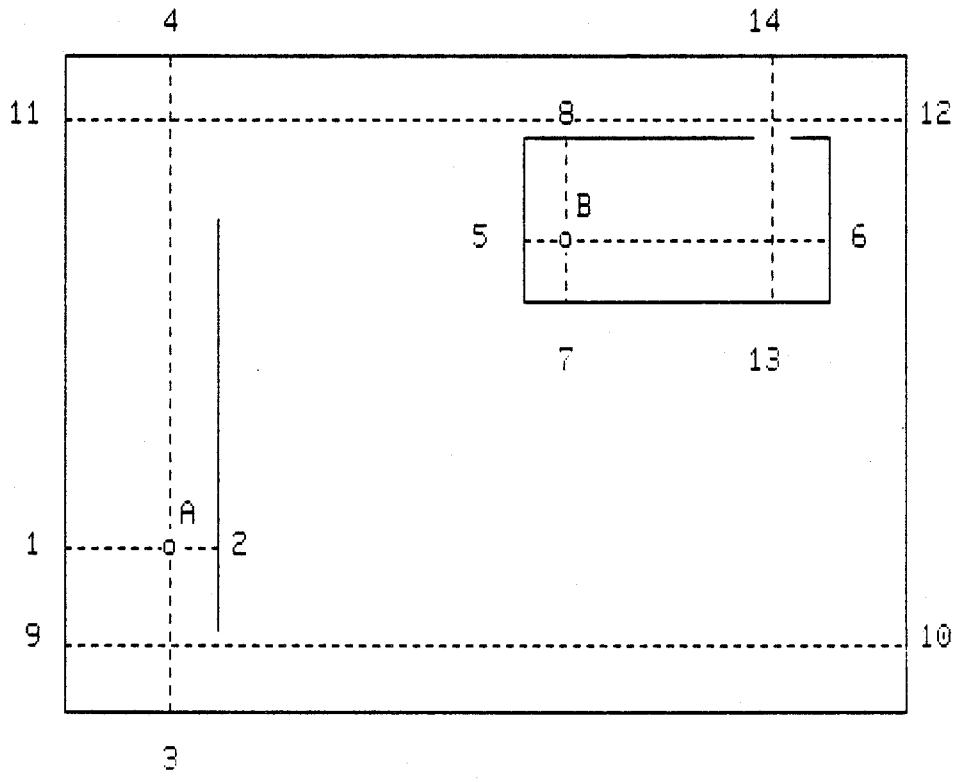


Figure 5. Algorithme de HIGHTOWER

La Figure 4 page 13 présente un exemple d'application.

La méthode est très coûteuse en temps CPU et en place mémoire occupée car on mémorise le plan sous forme d'une grille. La vitesse de l'algorithme peut être accrue de plusieurs façons :

- deux ondes sont developpées, une du point A et une du point B.
- la zone de propagation de l'onde est limitée à un rectangle dont une des diagonales est A-B.

1.2.2.2 Algorithme de HIGHTOWER

La recherche d'un chemin n'est pas effectuée point par point comme dans l'algorithme de LEE, mais au moyen d'un balayage par lignes, ce qui ne nécessite pas le stockage du plan.

L'algorithme construit des lignes d'expansion verticales et horizontales à partir des 2 points à connecter. Dans l'exemple de la Figure 5 page 13, il trace les droites 1-2,3-4,5-6 et 7-8; puis pour chacune de ces droites, il recherche les droites perpendiculaires les plus longues et il choisit les plus proches du point origine: ce sont les droites 9-10,11-12 et 13-14 sur l'exemple. Ce processus est répété jusqu'à ce que deux lignes d'expansion se croisent. La liaison est réalisée entre les deux points.

Cet algorithme génère dans la plupart des cas un chemin comportant un minimum d'angles. Il est très rapide pour des exemples simples; mais pour des problèmes compliqués, sa vitesse décroît vite, il demande une taille mémoire importante et ne permet pas toujours de réaliser toutes les liaisons.

Ces deux premières méthodes ont été un peu délaissées car les résultats étaient souvent peu satisfaisants : en effet, les connexions étant placées successivement, il est possible qu'une connexion déjà placée empêche le placement d'une nouvelle connexion. C'est pour ces raisons, entre au-

tres, que le problème du "routing" a été divisé en deux parties : le "global routing" et le "channel routing".

1.2.2.3 "Global routing".

Le "global routing" décide à travers quels canaux les connexions vont passer. Une étape préliminaire permet de définir les canaux d'interconnexion. Plusieurs méthodes ont été étudiées : elles reviennent toutes à diviser la zone d'interconnexion réservée pendant le placement en plusieurs rectangles. La méthode développée par les laboratoires Bell a été appliquée à l'exemple de la Figure 3 page 11.

Une fois ces canaux déterminés, l'algorithme proprement dit fonctionne de la manière suivante : prenons l'exemple de la liaison de 2 terminaux appartenant à des canaux différents. Partant d'un des deux terminaux, on cherche un chemin, une liste de canaux à traverser, le reliant au second terminal de manière analogue à l'algorithme de Lee avec les données suivantes :

- chaque canal correspond à une case chez Lee
- chaque canal est affecté d'un coefficient correspondant à sa longueur

On recherche le chemin le plus court entre les deux extrémités de la connexion. La méthode peut être améliorée si l'on donne des pénalités aux canaux critiques , c'est-à-dire ceux qui sont assez saturés et ceux pour lesquels le "channel routing" sera très difficile.

Le résultat du "global routing" est une division du problème d'interconnexions en sous-problèmes, chacun correspondant à un rectangle à interconnecter connaissant les fils qui doivent y passer.

1.2.2.4 "Channel routing".

Les nombreux algorithmes de "channel routing" traitent en général le problème d'un domaine rectangulaire, qui possède des terminaux fixes sur deux de ses côtés; ce sont des côtés opposés. Les connexions peuvent sortir du domaine sur les deux côtés restants mais c'est l'algorithme qui fixe leurs positions. Le but de l'algorithme est de minimiser la surface du canal; la longueur du canal, déterminée par la taille des blocs adjacents peut être considérée comme fixe. On minimise donc la hauteur, ce qui revient à minimiser le nombre de pistes utilisées.

Il est possible d'utiliser l'algorithme de Lee pour régler ce problème, mais de nouveaux algorithmes ont été étudiés et sont nettement plus performants que ce soit pour la surface utilisée ou pour le temps CPU. Trois parmi ces algorithmes importants seront précisés : le premier étudié par Hashimoto et Stevens (12), le "dogleg router" (13), et le dernier développé par Yoshimura et Kuh (14).

Algorithme de Hashimoto-Stevens.

C'est le premier algorithme qui ait traité le problème du "channel routing". Le but recherché est d'interconnecter un ensemble de boîtiers placés sur une carte. Ces boîtiers sont disposés sous forme de matrice. Un exemple en est donné sur la Figure 6 page 17 .

Les surfaces libres permettant l'interconnexion sont divisées en deux ensembles : les canaux horizontaux et les canaux verticaux. Un canal est un domaine qui s'étend d'un bord à l'autre de la carte et qui a une hauteur égale à la distance laissée entre deux rangées ou deux colonnes de boîtiers.

L'interconnexion se fait en directions privilégiées c'est-à-dire :

- tous les segments horizontaux sont placés sur le premier niveau d'interconnexion ou dans ce cas, la première face de la carte.

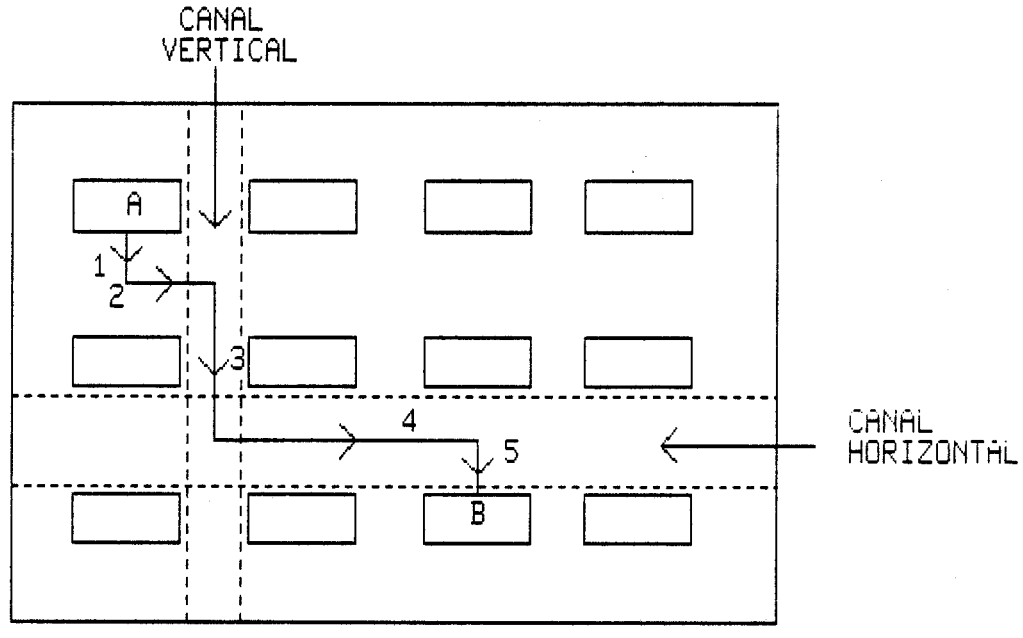


Figure 6. Carte à interconnecter

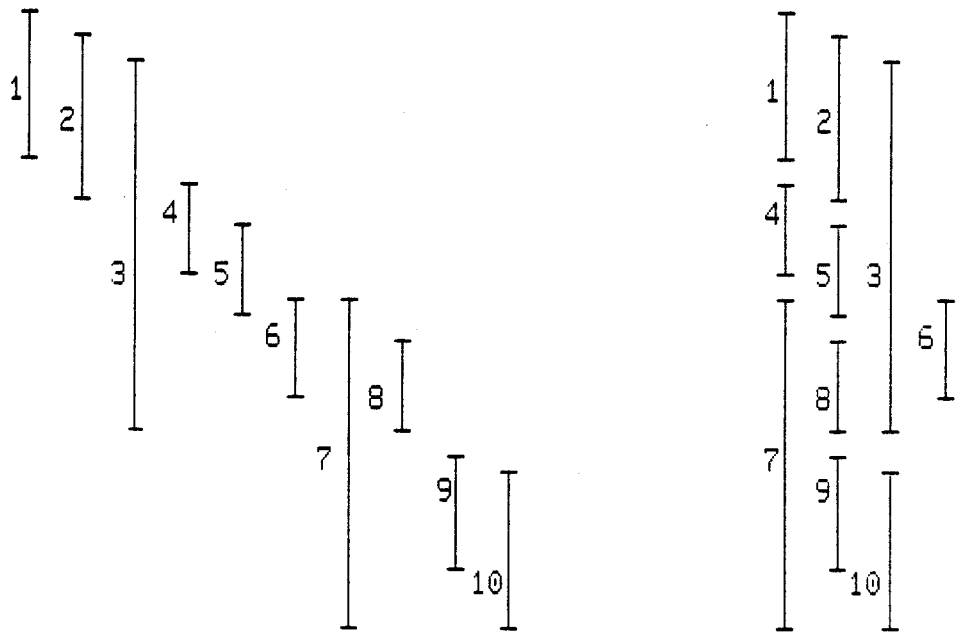


Figure 7. Algorithme de Hashimoto-Stevens

- tous les segments verticaux sont disposés sur le second niveau.

La méthode comprend deux étapes :

1. un "global routing" classique : chaque connexion est divisée en au moins 5 segments (Figure 6 page 17) ; on choisit pour chaque segment le canal où il doit être placé.

2. un "channel routing" : chaque canal vertical ou horizontal comprend un ensemble de segments ; chaque segment est défini par une borne inférieure et une borne supérieure: c'est une abscisse ou une ordonnée suivant le cas. L'algorithme cherche à placer tous ces segments dans le canal en minimisant le nombre de pistes nécessaires, ce qui revient à minimiser la hauteur du canal. Il fonctionne de la manière suivante : pour chaque piste, on effectue les opérations :

   - on recherche le segment qui a la borne supérieure la plus grande; ce segment est placé sur la piste en question.

   - parmi les segments restants, on recherche le segment qui possède la borne supérieure la plus grande tout en étant inférieure à la borne inférieure du dernier segment placé. Ce segment est placé.

   - cette dernière opération est répétée jusqu'à ce que la piste traitée soit remplie.

   Le traitement est terminé lorsqu'il n'y a plus de segment à traiter. Un exemple est donné sur la Figure 7 page 17.

"Dogleg channel router"

Avant de traiter le "dogleg" proprement dit, on introduira plusieurs notions liées au fait d'utiliser des directions privilégiées :

1. le canal est supposé horizontal. Un système de coordonnées est choisi: les abscisses x définissent l'horizontale, les ordonnées y la verticale. L'unité définie sur l'axe des abscisses correspond à la distance minimale entre deux terminaux; l'unité définie en ordonnée correspond au passage d'une piste.

2. la DENSITE du canal : c'est le nombre minimum de pistes nécessaires à l'interconnexion dans le canal. La densité se définit comme suit (Figure 8 page 20) :

   - o le segment horizontal de chaque connexion comprend une borne inférieure et une borne supérieure correspondant respectivement au premier et au dernier terminal de la connexion.

   - o à chaque abscisse du canal, on associe une valeur appelée densité locale qui est égale au nombre de segments horizontaux passant à cet endroit du canal.

   - o on appelle densité du canal la valeur maximale des densités locales. Toute solution nécessite un nombre de pistes égal ou supérieur à la densité: on appellera solution optimum une solution où le nombre de pistes est égal à la densité.

3. le fait d'utiliser des directions privilégiées entraîne également des CONTRAINTES VERTICALES. Ceci est illustré par la Figure 9 page 20 : si deux connexions distinctes possèdent des terminaux à la même abscisse dans le canal, le segment horizontal correspondant au terminal supérieur doit être placé au-dessus du segment horizontal correspondant au terminal inférieur. Un cas particulier de contrainte est la contrainte cyclique. Les résultats donnés par l'algorithme de Hashimoto-Stevens ne sont pas optimaux s'il y a des contraintes verticales et il n'y a pas de solution s'il y a des contraintes cycliques.

Pour expliquer le principe du "dogleg", on peut prendre la Figure 10 page 22 : l'exemple 1 correspond à une interconnexion faite par un algorithme type Hashimoto; il n'est pas optimum puisque le nombre de pistes est 3 alors que la densité est 2. L'exemple 2 résoud le problème : on parle de

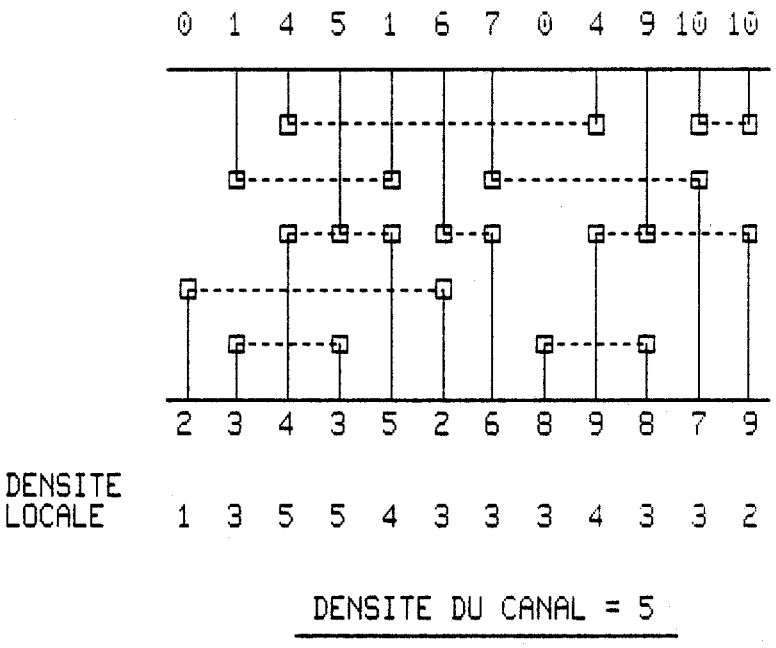


Figure 8. Densité d'un canal

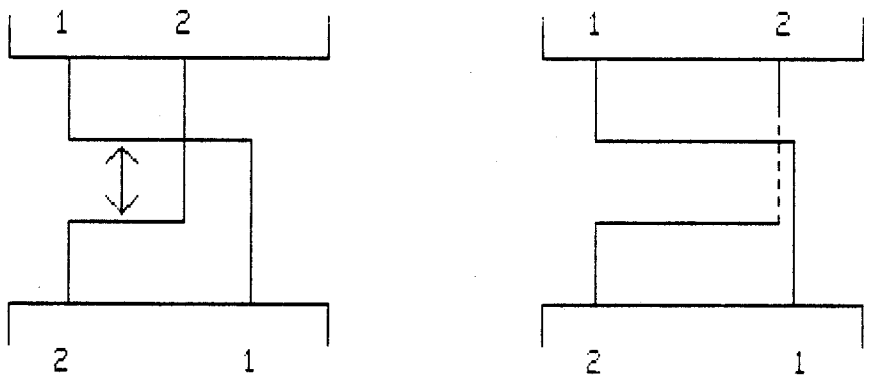


Figure 9. Contraintes verticale et cyclique

"dogleg" ; il y a transfert d'une partie du segment horizontal sur la piste précédente. Le "dogleg" permet également de résoudre le problème des contraintes cycliques au prix d'une piste supplémentaire; l'exemple 3 illustre ce cas.

Cette méthode a aussi des désavantages : on crée des contacts supplémentaires; d'autre part, si l'on réalise un "dogleg" à une position du canal où la densité locale est égale à la densité du canal, le nombre de pistes nécessaires ne peut plus être minimum.

L'algorithme est appliqué à des positions du canal où se trouvent des terminaux de la connexion traitée, car il ne faut pas trop utiliser le "dogleg" pour les raisons précédemment données. La méthode proprement dite est la suivante:

- chaque segment horizontal est décomposé en sous-segments de longueur unité: c'est la distance entre deux terminaux consécutifs dans le canal.

- on applique l'algorithme de Hashimoto-Stevens à l'ensemble des sous-segments avec la remarque suivante : pour minimiser les "doglegs", on oblige un certain nombre de sous-segments d'une connexion donnée à être placés sur une même piste.

Cet algorithme donne de meilleurs résultats que l'algorithme de Hashimoto-Stevens pour le nombre de pistes utilisées.

<u>Algorithme de Yoshimura et Kuh.</u>

Les hypothèses de départ sont les mêmes que pour le "dogleg router" : on a deux rangées de terminaux, on utilise des directions privilégiées. Mais cette méthode ne peut pas traiter les cas cycliques; il faut dans ce cas décaler une des rangées par rapport à l'autre.

Cet algorithme tient compte de deux types de contraintes:

1. les contraintes verticales définies précédemmment : deux segments verticaux ne peuvent se superposer.

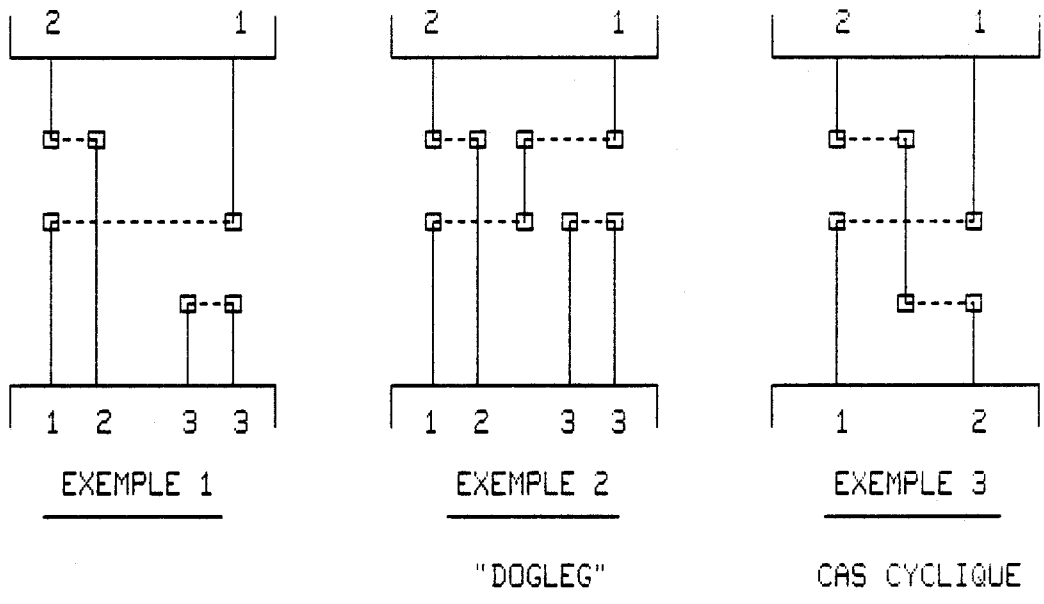


Figure 10. "Dogleg"

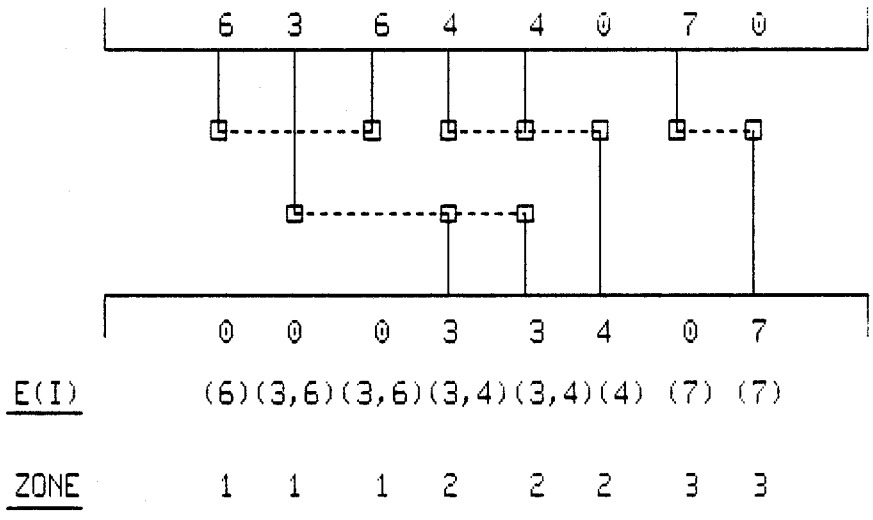


Figure 11. Définition de zones.

2. les contraintes horizontales : deux segments horizontaux appartenant à des connexions différentes et qui se superposent horizontalement doivent être assignés à des pistes différentes.

A partir des contraintes verticales, on crée un graphe Gv défini de la manière suivante :

- les noeuds du graphe sont les connexions (équipotentielles) du problème.
- il y a une arête entre deux noeuds(connexions) A et B si la connexion A doit être placée au-dessus de la connexion B.

Ce graphe doit être acyclique.

Les contraintes horizontales permettent de définir des zones : soit E(i) l'ensemble des connexions dont les segments horizontaux interceptent la colonne i du canal; on appelle zone une région contenant un ensemble de connexions distinctes appartenant à des E(i) correspondant à des colon-

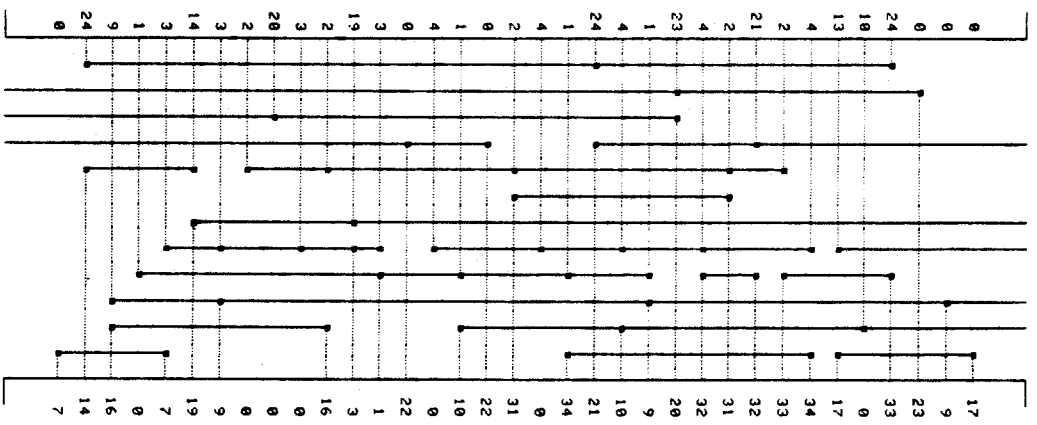


Figure 12. Algorithme Yoshimura-Kuh

nes consécutives et telle que pour deux colonnes adjacentes i et i+1, E(i) est inclus dans E(i+1) ou l'inverse. Pour préciser cette notion, prenons l'exemple résolu de la Figure 11 page 22 : le problème est défini par les deux séquences de terminaux (6 3 6 4 4 0 7 0) et (0 0 0 3 3 4 0 7); le nombre 0 corresond à l'absence de connexion. L'algorithme génère trois zones.

Le principe de l'algorithme est de réunir sur chacune des pistes deux ou plusieurs connexions qui appartiennent à des zones différentes tout en vérifiant dans le graphe Gv que les contraintes verticales sont satisfaites. La Figure 12 page 23 donne un exemple assez complexe de canal traité par cet algorithme.

Les résultats obtenus donnent pour des problèmes difficiles un nombre de pistes égal ou légèrement supérieur à la densité, avec des temps de calcul très corrects. On peut considérer que le problème du "channel routing" à directions privilégiées est correctement traité par cet algorithme.

Il nous a paru intéressant dans le cadre de cette thèse de reprendre ce problème des interconnexions mais en nous démarquant des approches classiques, en particulier du "channel routing" à directions privilégiées . Notre méthode comprend deux étapes:

- rechercher une solution topologique
- réaliser le passage à la grille de cette solution.

Cette méthode devrait permettre également de dessiner automatiquement les cellules de base des circuits intégrés.

1.3 PRESENTATION DU PROBLEME

1.3.1 Exemples

Le problème que l'on se propose de résoudre est donc le suivant: considérons un domaine fermé sur la frontière duquel sont placés des points dans un ordre prédéterminé à des positions bien précises. Ces points doivent être reliés pour réaliser une fonction donnée. Précisons cette idée en présentant les trois applications réelles qui seront traitées.

1.3.1.1 Le "Channel Routing" bijectif

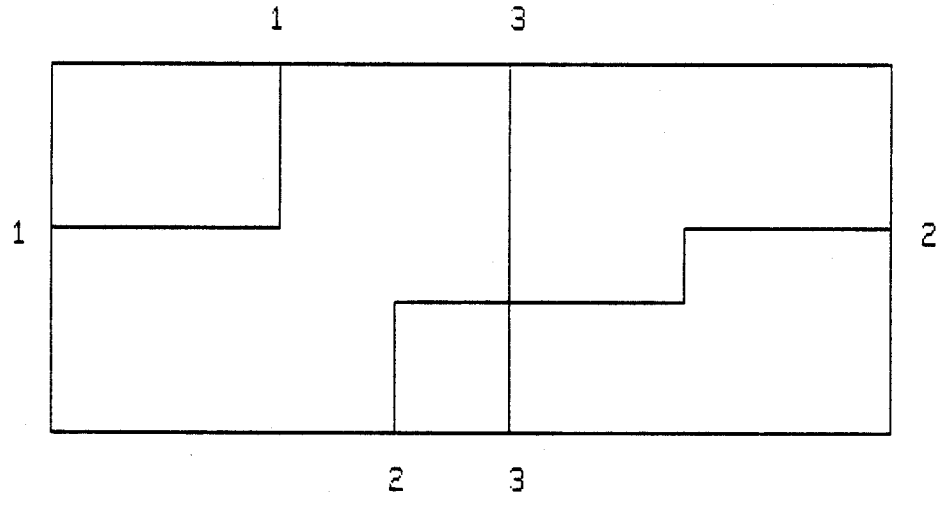


Figure 13. Channel routing bijectif

Deux points situés sur la frontière et affectés du même numéro doivent être connectés électriquement.

La fonction réalisée est donc la connectivité électrique.

1.3.1.2 Le "Channel Routing" général

Dans ce cas, plus de deux points situés sur la frontière peuvent être connectés électriquement. Cette application est plus proche de la réalité car l'observation des circuits intégrés a montré qu'en moyenne trois points sont connectés sur un même noeud électrique.

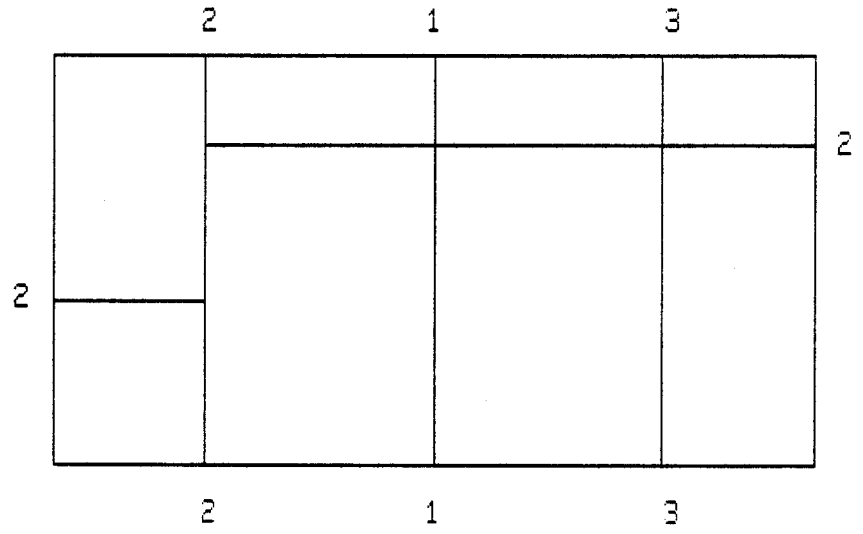


Figure 14. Channel routing général

1.3.1.3 Le dessin des cellules MOS

On appelle cellule MOS un ensemble de transistors MOS placés dans un domaine fermé qui permette de réaliser une ou plusieurs fonctions logiques. Une cellule est caractérisée par:

- ses fonctions logiques
- ses signaux d'entrée et de sortie
- ses performances en fonctionnement : niveaux électriques, vitesse et puissance consommée.

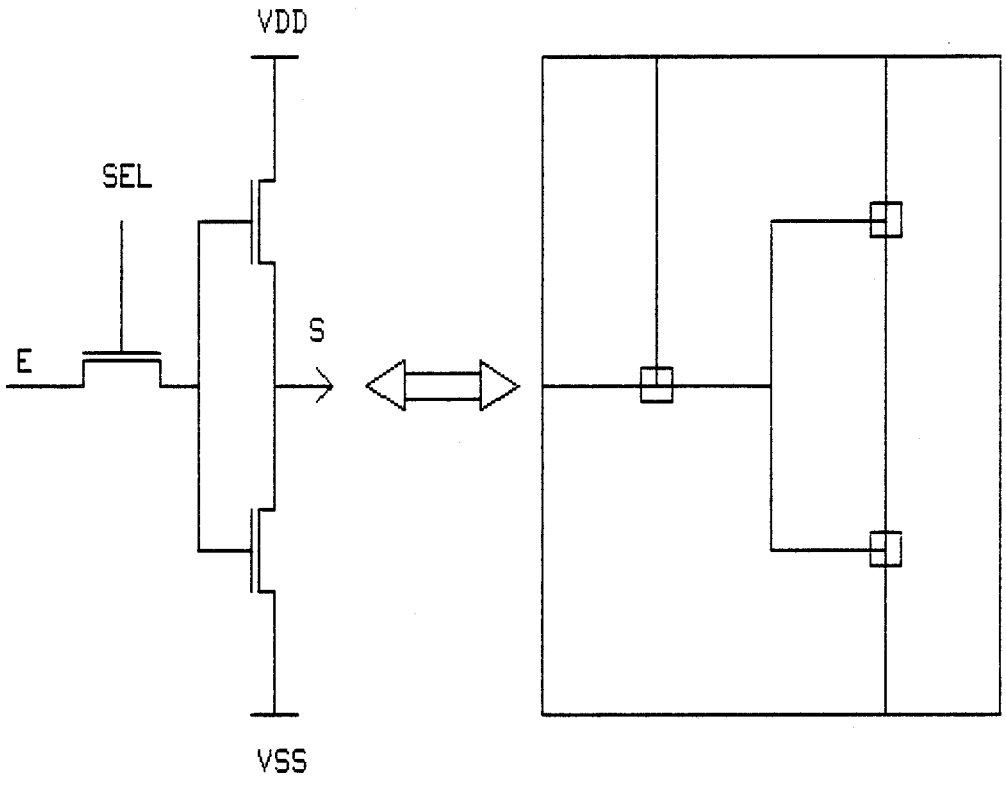


Figure 15. Dessin de cellule MOS

Au schéma logique, fonctionnel, de la cellule correspond un schéma électrique constitué par une énumération de transistors (description du type SPICE). Chacun est défini par:

o son drain,sa source et sa grille

o sa taille : longueur(L) et largeur(W) du canal.

La fonction réalisée lors de l'implantation d'une cellule MOS est donc la connexion des transistors entre eux ainsi qu'avec les points d'entrée et de sortie, ce qui revient à placer les transistors dans le domaine tout en respectant la connectivité des points appartenant au même noeud électrique.

Les points d'entrée et de sortie doivent pouvoir être placés n'importe où sur la frontière de la cellule.

1.3.2 La méthode

La méthode présentée dans cette thèse est originale car d'une part elle diffère complètement des algorithmes classiques et d'autre part elle permet de traiter de manière analogue les problèmes de "Channel routing" et de layout d'une cellule MOS.

Ce genre de problème nécessite de résoudre trois problèmes :

1. définir les positions relatives des connexions les unes par rapport aux autres. Ici intervient une notion de topologie.

2. réaliser le tracé symbolique de ces connexions; des contraintes géométriques influent sur ce tracé.

3. générer les masques en fonction des règles de dessin de la technologie.

Dans les approches classiques, les deux premières étapes sont mêlées. Il a paru intéressant de les dissocier de sorte que la seconde étape soit indépendante de la première. Le problème est alors focalisé sur une recherche de solution topologique sans tenir compte des contraintes géométriques.

1.3.2.1 La recherche topologique

Elle permet de placer les connexions les unes par rapport aux autres en essayant d'optimiser différents paramètres: le nombre de contacts, la longueur des chemins. Ce placement est fictif et ne fait intervenir aucune notion de dimension.

En effet, un solution topologique sera définie par un ensemble de zones, la réunion de ces zones constituant le domaine initial. Chaque zone sera définie par :

- les différentes connexions qui forment sa frontière

o les zones qui lui sont contiguës

Si on prend l'exemple de la Figure 16, la zone 4 a pour zones contiguës les zones 2,3 et 5.

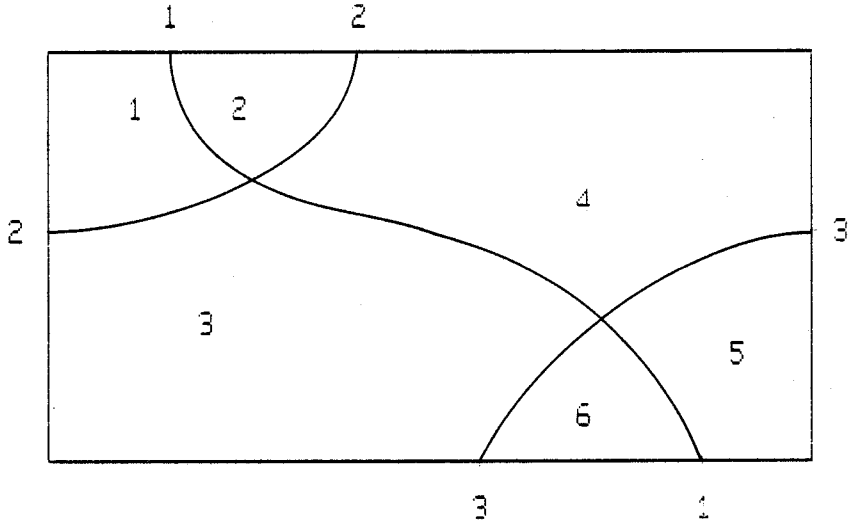


Figure 16. Recherche topologique

Ajouter une connexion revient à trouver un chemin entre 2 zones, la zone du point origine de la connexion et celle du point destination, c'est-à-dire déterminer une suite de zones à traverser.

1.3.2.2 Le passage à la grille.

Il s'agit ici de donner une consistance physique à la solution topologique. Toutes les zones doivent être orientées et dimensionnées. Le résultat obtenu sera un dessin symbolique de la cellule ou du canal d'interconnexion. Le dessin symbolique sera du type "Stick Diagram".

Si l'on reprend l'exemple présenté plus haut, on peut obtenir la solution suivante (Figure 17 page 30):

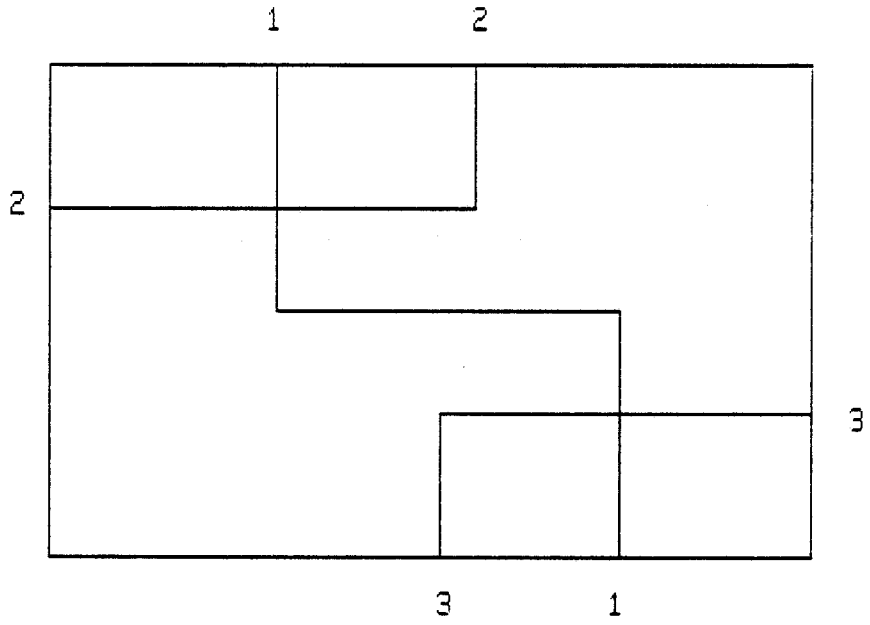


Figure 17. Passage à la grille

1.3.2.3 Le passage aux masques.

A partir d'un dessin symbolique, il est possible de passer aux masques réels par élargissement des lignes en respectant les règles de dessin de la technologie. De nombreuses études ont été réalisées sur ce sujet et donnent de bons résultats. Cette étape pourra également s'appeler compaction de la cellule ou du canal.

1.3.3 Remarques

1. Dans chacune des applications, on essaiera d'optimiser certains paramètres:

   - pour les connexions dans un canal, la hauteur du canal et le nombre de contacts devront être minimisés
   - pour les cellules MOS, il s'agira de minimiser la surface et de donner un facteur de forme au domaine

2. Dans les exemples décrits précédemment, tous les domaines initiaux sont des rectangles. Il devra également être possible de traiter d'autres types de domaines avec la restriction suivante: tous les points anguleux de ce domaine doivent être des angles droits. Ce choix est lié aux algorithmes utilisés et sera expliqué plus tard.

3. Les technologies MOS utilisées seront du type NMOS ou CMOS à un niveau de métal et un niveau de polysilicium.

1.3.4 Guide de la thèse

Les principes de base de la nouvelle méthode seront développés en détail dans le chapitre 2 sans appliquer vraiment les algorithmes à des cas précis.

Les chapitres suivants seront l'occasion de particulariser la méthode aux trois applications présentées plus haut: les "channel routing" bijectif et général, le dessin d'une cellule MOS. Signalons que toutes les définitions concernant les graphes sont données dans l'Annexe A.

2.0 METHODE GENERALE

2.1 APPROCHE DE LA METHODE

2.1.1 Le canal d'interconnexion

Considérons dans un premier temps un canal d'interconnexion placé entre deux blocs d'un circuit intégré. En technologie MOS classique, on peut considérer que l'on dispose de deux plans parallèles, le plan du métal et le plan du polysilicium, pour réaliser les connexions,la transition entre ces deux plans se faisant par des contacts métal-polysilicium. Le plan de la diffusion, dénommée également oxyde mince, est plus rarement utilisé que le polysilicium pour les connexions car les lignes de diffusion sont généralement plus capacitives que les lignes de polysilicium. Le métal est toujours utilisé à cause de sa très faible résistivité.

2.1.2 Le graphe des connexions

Prenons comme exemple la Figure 18 page 34. Dans chacun des deux plans, une connexion est matérialisée par un ensemble de segments de droite, deux connexions distinctes n'ayant aucun segment en commun.

La projection du schéma des connexions (2 plans) sur un plan unique (Figure 19 page 34) peut être modélisée par un graphe planaire simple dont les noeuds seraient :

- les points de croisement des connexions
- les points de dérivation sur les lignes
- les points d'entrée dans le canal

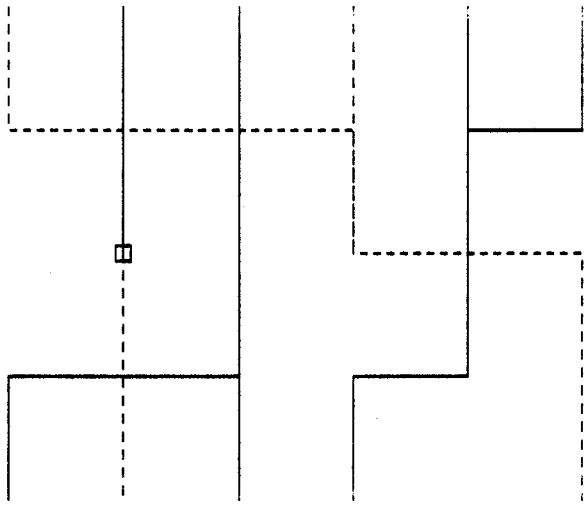


Figure 18. Exemple de canal d'interconnexion

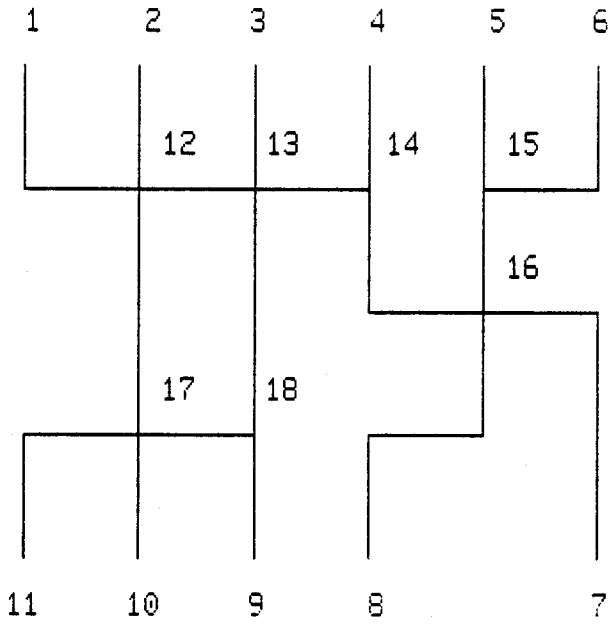


Figure 19. Projection du canal sur un seul plan

et dont les arêtes seraient les segments de droite qui constituent les connexions, plusieurs segments d'une même connexion pouvant appartenir à une même arête.

La Figure 21 page 36 représente le graphe des connexions généré à partir de la Figure 18 page 34.

On peut ajouter deux remarques au sujet de cette figure:

1. Les contacts métal-polysilicium ne constituent pas des noeuds du graphe représenté. En effet, une fois le schéma projeté sur un plan unique, les contacts qui faisaient la liaison entre les deux plans disparaissent. La Figure 20 donne deux exemples de suppression de contact. Dans le second exemple, le contact était placé sur une dérivation qui, elle, constitue un noeud du graphe.

2. Le graphe présenté n'est pas complet. En effet, le canal d'interconnexion est limité par des frontières, chaque frontière étant formée de segments reliant, soit deux points d'entrée dans le canal, soit un point d'entrée et une de ses extrémités, soit deux extrémités du canal. La Figure 22 page 36 présente le graphe modifié suite à cette remarque.

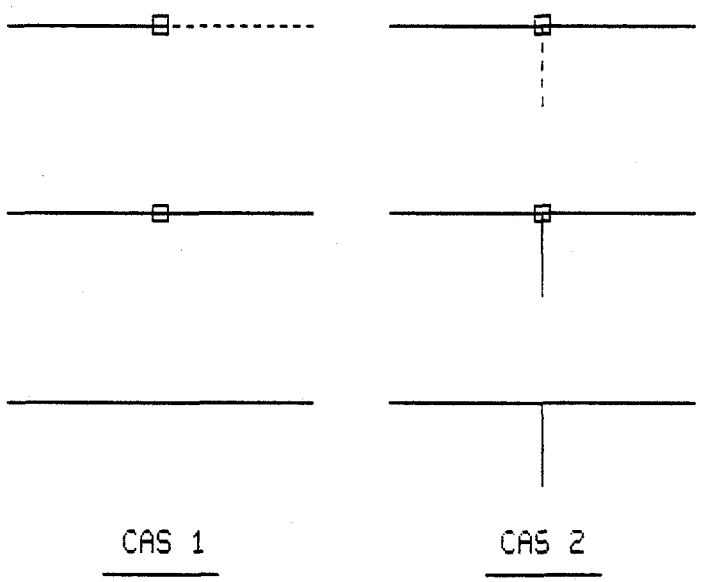


Figure 20. Exemple de suppression de contact

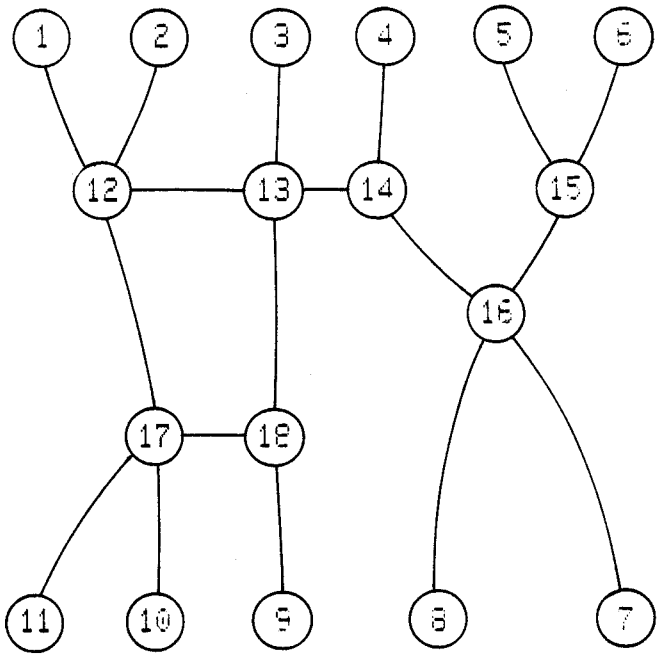


Figure 21. Graphe des connexions

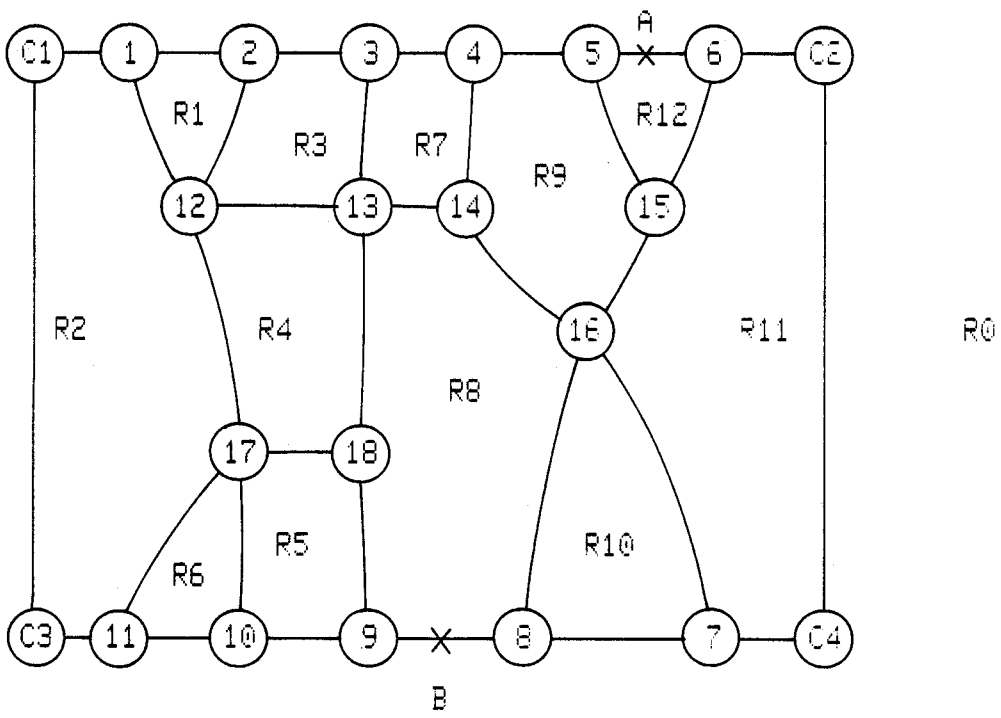


Figure 22. Graphe complet des connexions

2.1.3 Le graphe des régions

Supposons que l'on veuille ajouter une connexion reliant les points A et B dans l'exemple précédent. Le graphe des connexions est un graphe planaire et connexe; on peut donc lui faire correspondre son graphe dual (Figure 23 page 37) défini par :

- ses noeuds; ce sont les faces du graphe des connexions numérotées de R0 à R12. Chaque face correspond à une région.

- ses arêtes; une arête relie deux noeuds, deux régions du graphe si ces deux régions possèdent un segment, c'est-à-dire une frontière, en commun.

Pour connecter les deux noeuds, il suffit de les identifier à leurs régions respectives RGA et RGB et de chercher un chemin dans le graphe dual entre ces deux noeuds.

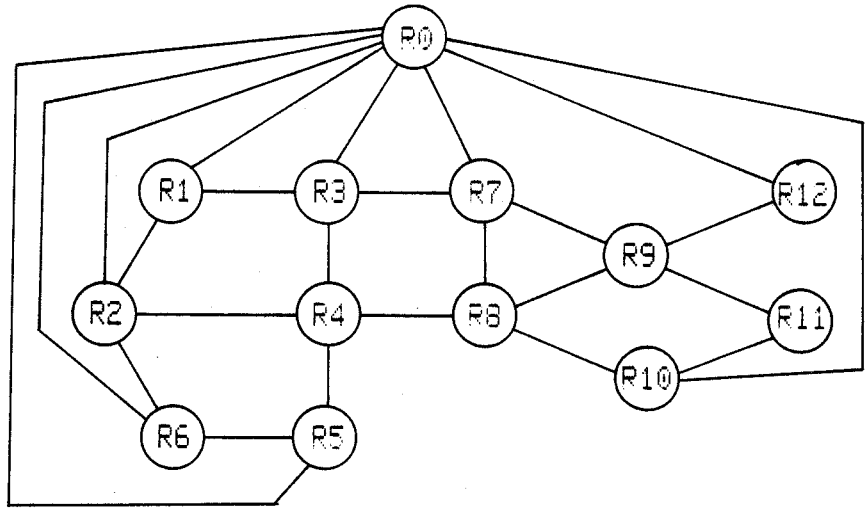


Figure 23. Graphe des régions

Dans le graphe dual, la région RGA est le noeud R12, RGB le noeud R8. Une des solutions possibles pour relier R12 et R8 est le chemin : R12-R9-R8 .

Le graphe dual que l'on appellera par la suite "graphe des régions" semble donc intéressant car il facilite la recherche d'un chemin pour une connexion que l'on veut créer.

2.1.4 Méthodologie

Partant du schéma d'un canal d'interconnexion, on a construit successivement deux graphes qui le caractérisent :

- le graphe des connexions
- le graphe des régions, graphe dual du précédent.

On est ainsi passé d'une configuration concrète à une configuration abstraite. On a pu également remarquer qu'il était possible de générer de nouvelles connexions dans le graphe des régions.

Au vu de ces deux constatations, on se propose dans cette thèse d'appliquer la méthode inverse :

- générer le graphe des régions en connaissant uniquement les points d'entrée des connexions sur la frontière du canal.
- passer de ce graphe au schéma du canal d'interconnexion.

La première étape s'appellera " Recherche topologique d'une solution ", la seconde " Passage à la grille ".

2.1.5 Application aux cellules MOS

De la même manière, considérons une cellule MOS. Classiquement on utilise trois niveaux technologiques:

- la diffusion (ou oxyde mince)
- le polysilicium
- le métal

Si l'on regroupe les niveaux deux par deux, on obtient trois couples :

- Métal-Diffusion
- Métal-Polysilicium
- Diffusion-Polysilicium

Les deux premiers ne posent pas de problèmes car on retrouve le cas du canal d'interconnexion. Le dernier, Diffusion-Polysilicium , n'entre pas dans ce cadre car la Diffusion et le Polysilicium ne jouent pas des rôles équivalents à ceux que jouent le Métal et le Polysilicium pour les interconnexions; en effet, le croisement d'une ligne de diffusion et d'une ligne de polysilicium génère un transistor.

La génération du graphe des régions se fera comme pour les interconnexions mais en traitant à part le cas des transistors. Le passage à la grille s'effectue suivant des méthodes similaires.

2.2 RECHERCHE TOPOLOGIQUE - TRAITEMENT GENERAL

La donnée du problème est un domaine fermé sur la frontière duquel seront placés les points à connecter. Au départ, le graphe des régions comprend donc deux noeuds: le noeud correspondant à l'intérieur du domaine et le noeud correspondant à l'extérieur, reliés par une arête qui représente la frontière du domaine. En fait, on ne conservera que le premier de ces noeuds dans le graphe origine car toutes les connexions doivent être placées physiquement à l'intérieur du domaine fermé.

A chaque noeud du graphe des régions, on associera deux informations:

1. la liste orientée des points définissant le contour de la région; l'orientation choisie pour chaque zone est le sens trigonométrique.

2. la liste des régions voisines, avec pour chacune d'entre elles, le segment qui fait office de frontière commune; un segment est formé de deux points consécutifs du contour. Cette seconde liste définit les arêtes du graphe partant de ce noeud.

Dans le cas où le domaine origine est un rectangle, ses quatre coins définissent le contour. Dans la suite de ce rapport les termes "région" et "zone" seront employés indifféremment.

2.2.1 Opération élémentaire

Considérons l'exemple le plus simple qui soit : un domaine fermé dans lequel on veuille relier deux points distincts placés sur sa frontière(Figure 24 page 42).

Au départ le graphe des régions comprend une seule région symbolisée par le noeud 1. Le contour de la région est la suite de points (C1 C2 C3 C4). En ajoutant deux points distincts sur ce contour, A entre C4 et C1, B entre

C2 et C3, et en les reliant, on divise la région en deux , ce qui équivaut au niveau du graphe à :

- o créer un nouveau noeud 2 dont le contour sera (C4 A B C3)

- o modifier le contour du noeud 1 qui devient (A C1 C2 B)

- o créer une arête reliant les deux noeuds pour symboliser la frontière commune A-B , ce qui revient à indiquer dans la description du noeud 1 (respectivement 2) qu'il a pour voisin le noeud 2 (resp. 1). Dans chaque région, le segment de contiguïté est orienté de la même manière que les points du contour. Dans la zone 1, le segment de contiguïté est B-A, dans la zone 2 c'est A-B. Par la suite, on utilisera indifféremment A-B ou B-A comme segment de contiguïté.

Le traitement décrit ci-dessus sera l'opération élémentaire appliquée à un noeud du graphe des régions. On parlera par la suite de "division d'une région".

2.2.2 Exemple de création d'un chemin

Reprenons l'exemple de la Figure 24 page 42 et ajoutons une connexion suppplémentaire reliant les points C et D placés sur la frontière, C entre C3 et C4, D entre C1 et C2.

Le point C appartient au contour du domaine représenté par le noeud 2 du graphe; de même le point D appartient au contour associé au noeud 1. Pour connecter les deux points, il suffit de trouver dans le graphe des régions un chemin reliant les noeuds 2 et 1. Ce chemin est unique ; il comprend le noeud 2, l'arête A-B, le noeud 1.

Créer physiquement une liaison entre C et D revient donc à: (Figure 25 page 42)

- o créer un point commun I sur la frontière commune A-B

- o traverser la zone 2 du point C au point I

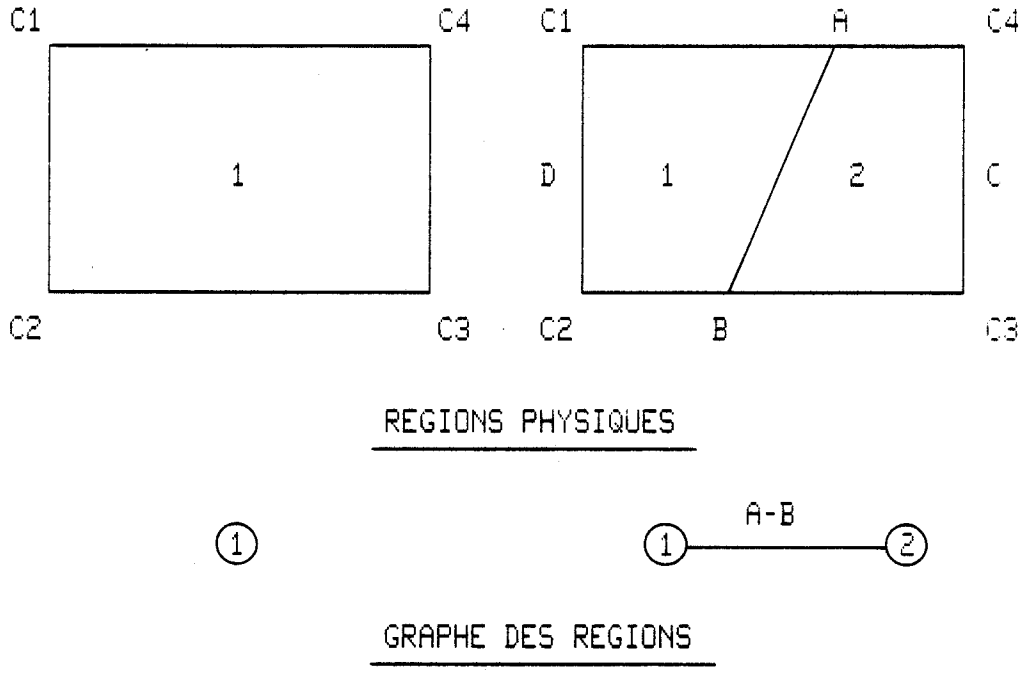


Figure 24. Connexion élémentaire

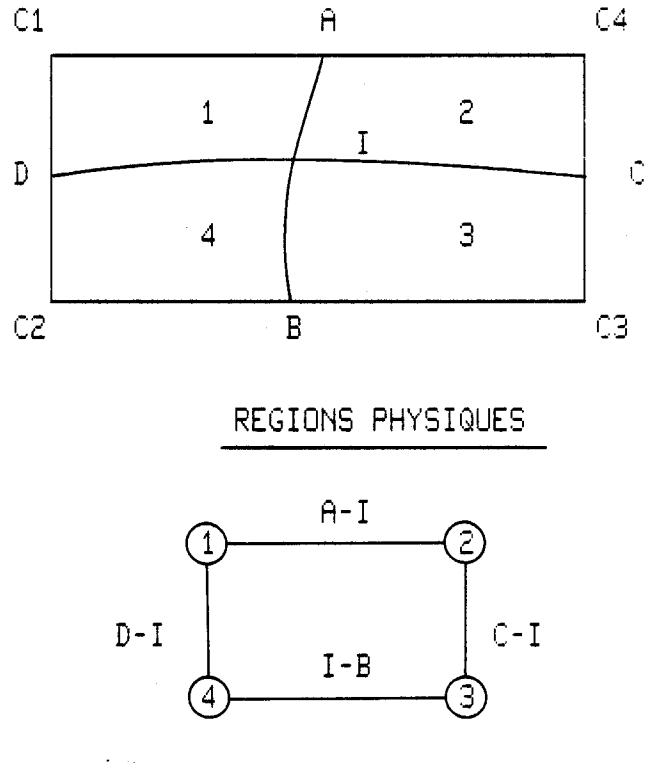


Figure 25. Connexion complexe

o traverser la zone 1 du point I au point D

La correspondance sur le graphe est immédiate. Le traitement comporte plusieurs étapes :

1. modifier le contour associé au noeud 2 en ajoutant le point I entre A et B.

2. faire la division du noeud 2 représentant la zone 2 suivant C-I; le noeud 3 est créé.

3. remplacer l'arête A-B entre les noeuds 1 et 2 par deux arêtes: A-I entre les noeuds 1 et 2, I-B entre 1 et 3.

4. modifier le contour associé au noeud 1 en ajoutant le point I entre B et A.

5. faire la division du noeud 1 représentant la zone 1 suivant I-D; le noeud 4 est créé.

6. placer l'arête I-B entre les noeuds 2 et 4 du graphe.

Les étapes 2 et 5 sont deux opérations élémentaires.

2.2.3 Généralisation

On peut généraliser cette méthode : considérons un domaine origine divisé en N zones auquel correspond un graphe des régions comportant N noeuds. Cette configuration est le résultat obtenu après le traitement de plusieurs connexions. Pour ajouter une nouvelle connexion, il faut effectuer les opérations suivantes :

1. Rechercher la région dont la frontière contient le point origine de la connexion

2. Rechercher la région dont la frontière contient le point destination de la connexion

Les régions origine et destination correspondent respectivement à un noeud origine dénommé Org et à un noeud destination dénommé Dest dans le graphe des régions.

3. Deux cas se présentent :

   a. Les noeuds Org et Dest sont confondus. Les deux extrémités de la connexion appartiennent à la même région. On effectue simplement une division de la région; c'est une opération élémentaire.

   b. Les noeuds Org et Dest sont distincts. Le traitement, plus complexe, nécessite :

      1) La recherche sous certaines contraintes d'un chemin reliant les noeuds origine et destination dans le graphe des régions.

      2) La transposition de ce chemin au niveau du graphe, ce qui revient à appliquer le traitement présenté dans le paragraphe précédent non plus à deux noeuds mais à autant de noeuds qu'il y a de régions dans le chemin.

Précisons les deux opérations essentielles:

2.2.3.1 La recherche du chemin.

Elle s'apparente à celles faites classiquement dans les graphes. Elle dépend du cas traité et ne fera donc pas l'objet d'un développement dans ce chapitre. On peut cependant ajouter deux précisions. Le chemin trouvé est la liste des régions à traverser successivement. A chaque élément de la liste, on ajoute une information supplémentaire : le segment qui constitue la frontière commune entre la région pointée par cet élément et sa suivante dans le chemin.

2.2.3.2 La transformation du graphe.

On effectue pour chaque noeud du graphe un traitement similaire. Considérons une région quelconque R à traverser. (Figure 26 page 46). Elle est définie par :

- son contour I0,I1,..Ii,Ij1,I(i+1),....Ij,I(j+1),...In
- les régions qui ont un segment en commun avec elle à savoir R0(I0-I1)...R(i-1),Ri(Ii-Ij1),Rj1(Ij1-I(i+1))....Rj(Ij-I(j+1))....Rn(In-I0) .

La nouvelle connexion qui traverse cette région y entre par le point Ij1 placé sur l'ancien segment Ii-I(i+1) et en sort par le segment Ij-I(j+1). Ceci a été déterminé lors de la recherche du chemin. Les modifications du graphe sont les suivantes :

1. insertion d'un nouveau point Ij2 dans le contour qui devient I0,I1,..Ii,Ij1,I(i+1),....Ij,Ij2,I(j+1),...In. Ce point est le point de sortie de la connexion. Ce point doit également être ajouté dans le contour de la région voisine Rj.
2. division de la région R, ce qui entraîne :
   - la création d'un nouveau noeud S dans le graphe dont le contour est I0,I1,..Ii,Ij1,Ij2,I(j+1),...In.
   - la modification du contour de l'ancienne région R qui devient Ij1,I(i+1),....Ij,Ij2.
   - la création d'une arête entre R et S
3. mise à jour des anciennes contiguïtés du noeud R avec les régions voisines. Trois cas se présentent :
   a. les deux points du segment de contiguïté appartiennent au nouveau contour de R. C'est le cas des arêtes Ij1-I(i+1).....I(j-1)-Ij . Ces arêtes du graphe sont inchangées.

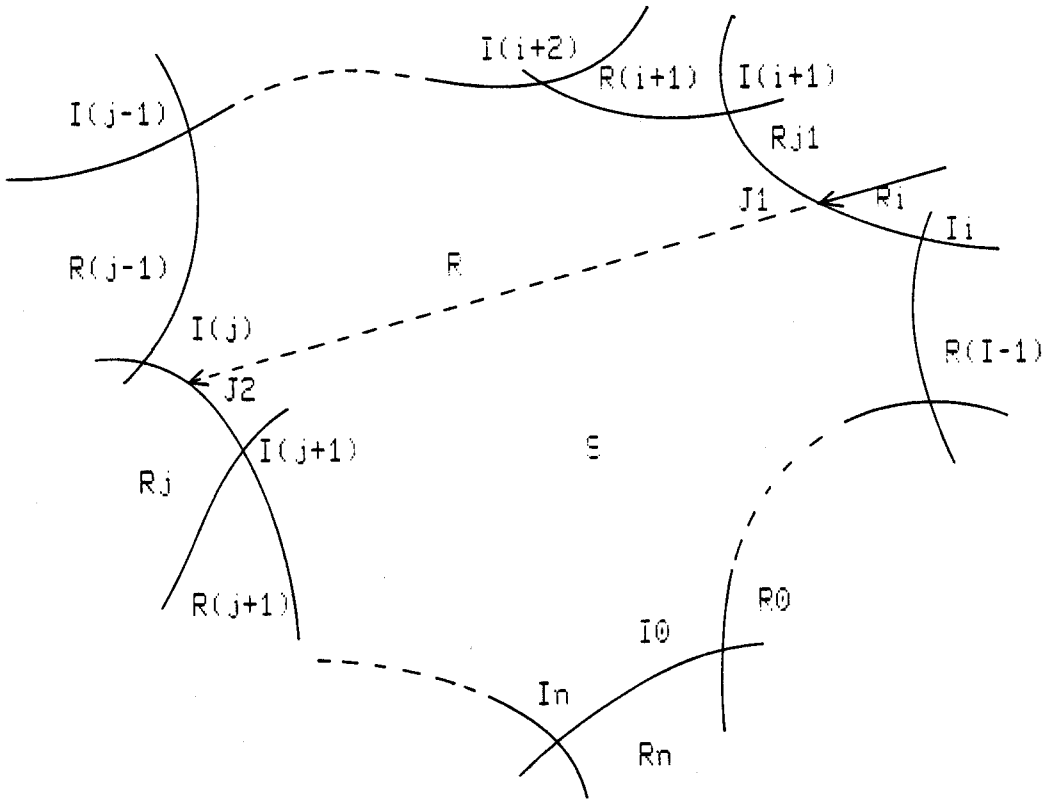


Figure 26. Traitement d'un noeud du graphe

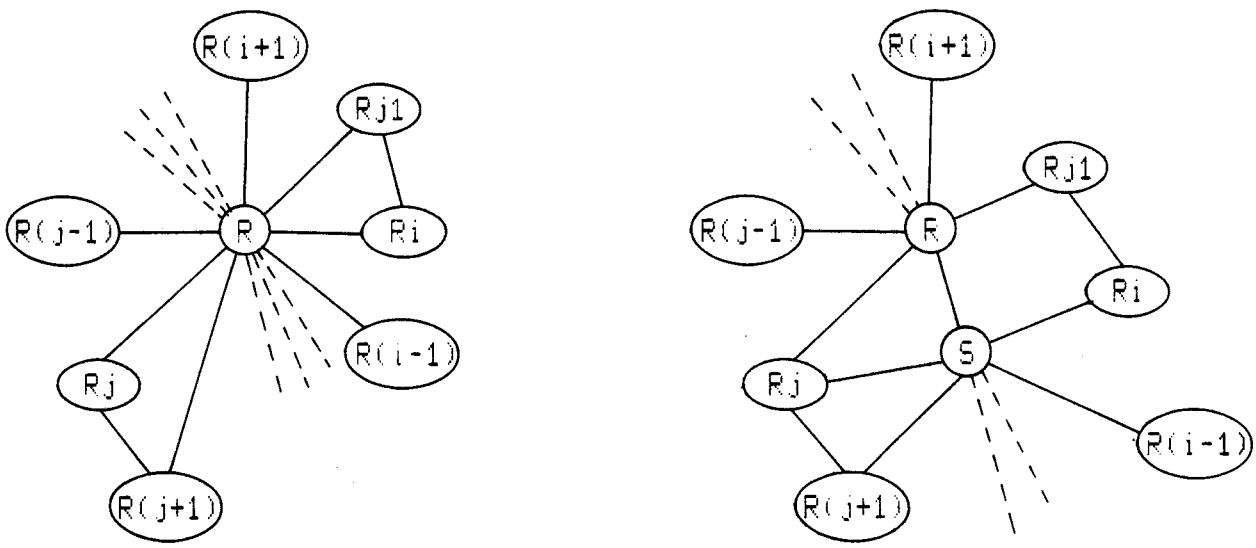


Figure 27. Graphe original et graphe modifié

b. les deux points du segment de contiguïté appartiennent au contour de S. Les arêtes qui reliaient R et les régions R(j+1),....R0,...,R(i-1),Ri deviennent des arêtes reliant S et ces mêmes régions.

c. un point du segment appartient au nouveau contour de R et un au contour de S. C'est le segment Ij-I(j+1). L'arête entre S et Rj est créée avec pour segment de contiguïté Ij2-I(j+1) ; le segment de contiguïté entre R et Rj est modifié et devient Ij-Ij2.

Le graphe original et le graphe modifié sont donnés sur la Figure 27 page 46

La recherche topologique génère donc un graphe correspondant au placement des connexions les unes par rapport aux autres. Chaque noeud ou région est défini par son contour et ses relations avec les régions qui lui sont contiguës. Cette phase du traitement ne fait intervenir aucune métrique. A partir de ce graphe qui n'a rien de concret, il s'agit de placer physiquement ces régions en respectant les données du graphe.

2.3 LE PASSAGE A LA GRILLE

Le " passage à la grille " est la deuxième étape du traitement. Partant d'un graphe des régions, il s'agit de générer un schéma physique du canal d'interconnexion ou de la cellule, ce qui revient à définir une fonction sur l'ensemble des noeuds ou régions du graphe à valeurs dans l'ensemble des polygones.

Chaque région est définie par une liste de points. cette liste correspond au contour orienté de la zone. Pour transformer la suite de points en un polygone, deux opérations sont nécessaires :

- donner une géométrie à ce contour (triangle,rectangle, hexagone,etc...).
- dimensionner les côtés du polygone.

Indépendamment de la description explicite du contour de la région, plusieurs facteurs influent sur le choix de sa géométrie et sur son dimensionnement :

- les régions qui lui sont contiguës, définies par les arêtes du graphe
- la forme du canal ou de la cellule
- les positions des points d'entrée dans le canal.

Les difficultés pour réaliser le passage à la grille se situent essentiellement au niveau du choix de la géométrie. Connaissant la géométrie des régions, leur dimensionnement, qui tient compte du graphe des régions, est assez simple.

Trois approches de ce problème ont été étudiées; elles sont développées dans ce paragraphe:

- la première favorise la création d'un noyau interne.
- la seconde utilise trois types de polygones prédéfinis.
- la dernière ne génère que des rectangles.

2.3.1 Géométrie des régions

2.3.1.1 Polygones autorisés

Dans cette étude, seuls les segments horizontaux et verticaux seront autorisés. La possibilité offerte par la plupart des technologies de dessiner des segments à 45 degrés ne sera pas utilisée. Ceci a une conséquence directe sur le choix des polygones : ils doivent comporter un nombre pair de côtés ou d'angles droits.

En effet, considérons d'une part un vecteur parcourant successivement tous les côtés d'un polygone quelconque. Ce vecteur subit des rotations telles que la somme des angles de rotation est un multiple de 360 degrés lorsque tous les côtés ont été parcourus une fois.

D'autre part, supposons qu'il existe un polygone fermé possédant un nombre impair d'angles droits ou de côtés. Si un vecteur parcourt de la même manière ce polygone, la somme des angles de rotation sera un nombre impair de fois 90 degrés; ce qui est en contradiction avec la règle énoncée plus haut. Donc si un polygone ne possède que des angles droits, le nombre de côtés de ce polygone est pair.

Les polygones autorisés seront les rectangles, les hexagones, les octogones,etc... ne comportant que des angles droits.

On peut ajouter une remarque importante : il n'y pas de relation directe entre le nombre de points définissant le contour du domaine et le nombre d'angles du polygone associé. En effet si la zone contient un point de dérivation (cas d'une connexion dont une des extrémités est placée sur une autre connexion), celui-ci ne deviendra pas nécessairement un angle.

2.3.1.2 Traitement privilégié des zones internes

On appelle zone interne une région du graphe qui ne comporte aucun point placé sur la frontière du canal ou de la cellule et zone externe une région touchant cette frontière.

Dans cette première méthode, les zones internes sont particularisées. Le but recherché est de créer un noyau interne optimisé, les sorties de ce noyau étant ensuite reliées aux points d'entrée dans le canal. On parle de noyau interne optimisé lorsque son centre est formé de zones à 4 points, des zones à 3 points pouvant être placées sur son pourtour. Ce choix est fait car le passage à la grille est facile pour ce type de zones.

Pour optimiser le noyau, la recherche topologique ne doit donc générer que des régions internes à 3 ou 4 points; les régions internes comportant des dérivations peuvent cependant posséder des points supplémentaires.

L'algorithme de recherche du chemin permet de répondre à ces contraintes; il fonctionne de la manière suivante (Figure 28 page 51) :

- o si la nouvelle connexion traverse une zone interne, deux cas se présentent :
  - c'est une zone à 3 points; elle est divisée en 2 zones, une zone à 3 points et une zone à 4 points
  - c'est une zone à 4 points; elle est divisée en 2 zones à 4 points
- o si la nouvelle connexion traverse une zone externe et si elle crée une zone interne, celle-ci ne doit contenir que 3 ou 4 points.

Pour illustrer la constitution d'un noyau, considérons l'exemple de la Figure 29 page 52 . Il s'agit de connecter 2 rangées de terminaux (1 2 3 4 5) et (5 4 3 2 1) placés sur les côtés opposés d'un canal. Le noyau central est constitué des régions internes R1 à R6 .

Le passage à la grille est très simple pour les régions du noyau. En ne tenant pas compte des points de dérivation,

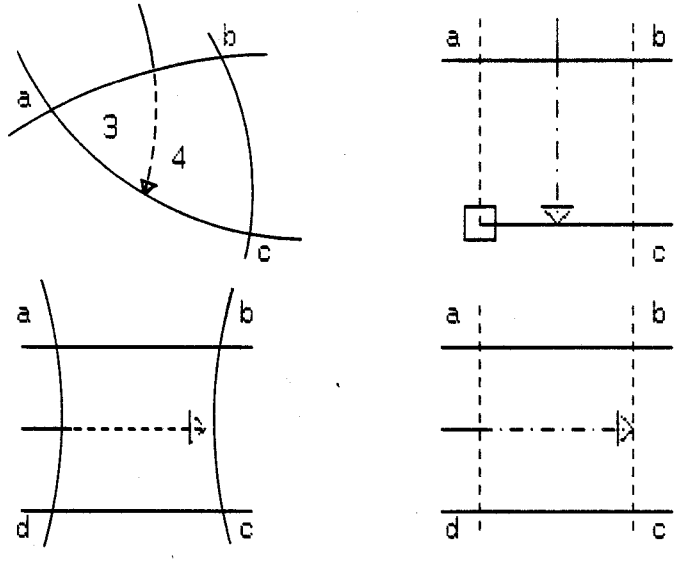
les zones à 4 points deviennent des rectangles; il en est de même pour les zones à 3 points: on leur ajoute 1 point sur un des segments contigus à une zone externe. Toutes les régions s'assemblent ensuite facilement et l'on obtient une sorte de damier (Figure 30 page 52). Donner une géométrie aux zones externes revient à relier les points d'entrée dans le canal et les points de sortie du noyau.

Mais cette approche ne donne pas de bons résultats car une des deux rangées de terminaux se trouve décalée par rapport à l'autre. Dans le cas où le traitement topologique génère plusieurs noyaux, les liaisons entre les noyaux posent d'autres problèmes.

Cette méthode présente donc assez peu d'intérêt. Deux raisons peuvent l'expliquer :

1. lors du traitement topologique, la recherche du chemin se fait en fonction de critères géométriques facilitant le passage à la grille, alors que le chemin trouvé doit dépendre d'autres critères : le nombre de contacts,etc....

2. le passage à la grille ne tient pas vraiment compte, ni de la forme du canal, ni de la position des entrées sur le contour. Les zones externes ne sont pas prises en compte comme les zones internes.



RECHERCHE D'UN CHEMIN PASSAGE A LA GRILLE

Figure 28. Traversée d'une région.

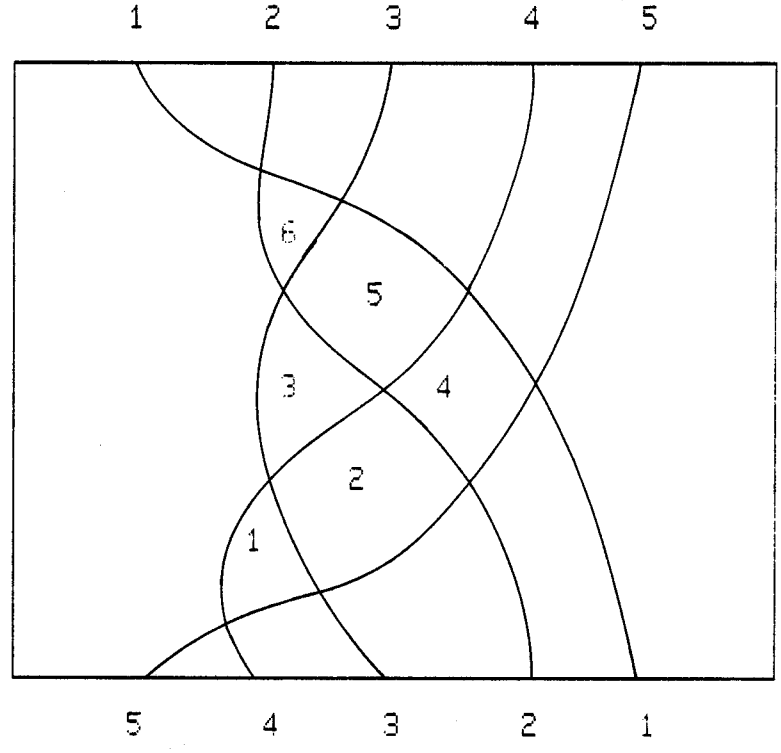


Figure 29. Traitement topologique des zones internes

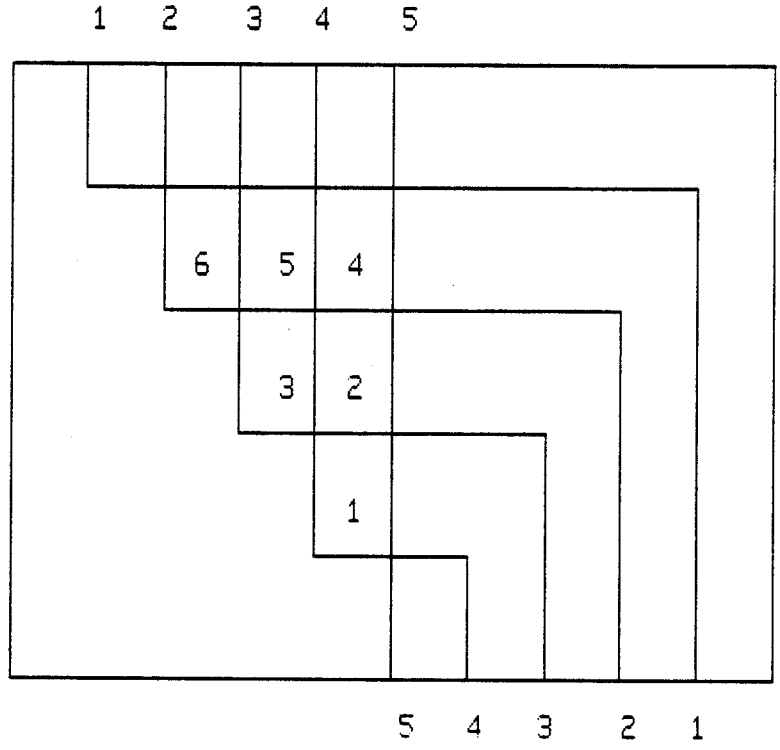


Figure 30. Passage à la grille

2.3.1.3 Régions à 4,6 ou 8 côtés.

Par rapport à la première, la seconde méthode reprend le problème à l'envers. Considérant que toutes les zones, qu'elles soient internes ou externes, doivent être traitées de la même manière, on se définit un ensemble de formes géométriques qui peuvent être utilisées pour chaque région.

Les polygones autorisés comprennent 4,6 ou 8 côtés (Figure 31 page 54). Pour les polygones à 8 côtés, plusieurs géométries sont possibles. La limitation à 8 côtés s'explique simplement : plus le nombre de côtés augmente, plus le nombre de formes géométriques différentes s'accroît. Or chaque forme géométrique nécessite un dimensionnement particulier, et donc un accroissement de la taille des programmes.

Chacun des 5 polygones peut être associé à un type de région :

- le rectangle à une zone à 4 points
- l'hexagone à une zone à 5 points résultant de la division d'une zone à 4 points en une zone à 3 points et une zone à 5 points
- le premier octogone à une zone à 4 points traversée par deux connexions ayant créé deux zones à 3 points
- etc...

La réciproque est plus difficile à faire : dans les régions comportant un nombre impair de points, les points de dérivation non compris, il faut ajouter des points fictifs pour obtenir une des géométries autorisées. Ceci ne peut se faire efficacement que pendant la recherche topologique car, une fois le graphe généré, la correspondance entre polygone et région n'est plus immédiate. On peut noter que dans cette approche, la recherche topologique fait intervenir la métrique cartésienne sans pour autant parler de longueur de connexion puisque les connexions ne peuvent se déplacer que suivant deux directions.

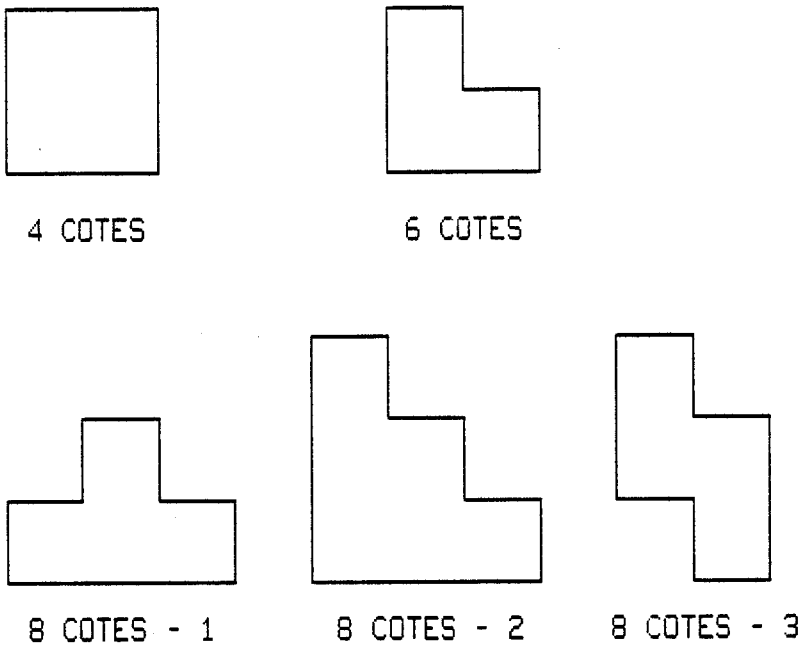


Figure 31. Géométries utilisées

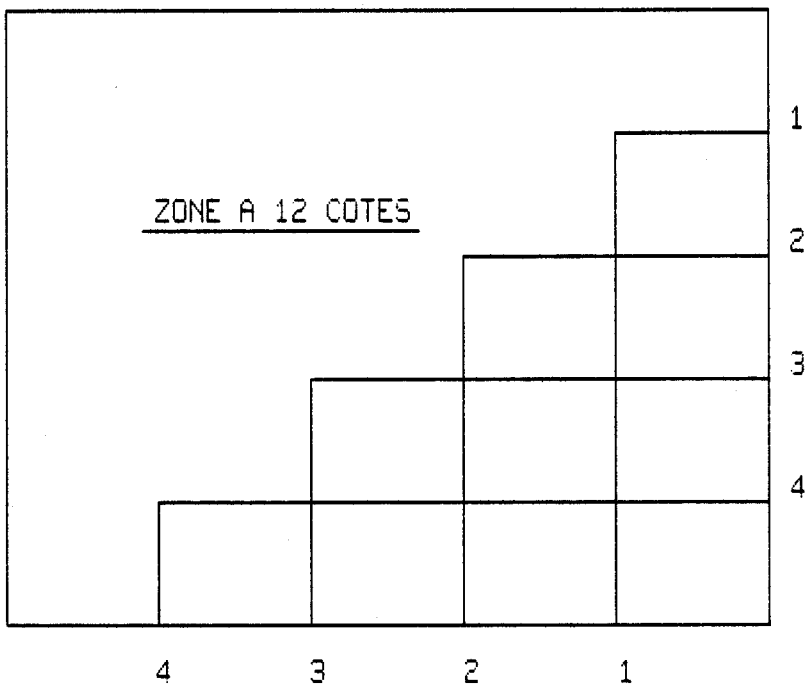


Figure 32. Exemple non solvable

Cette approche paraît plus satisfaisante que la première mais elle comporte un défaut important : en choisissant une borne supérieure pour le nombre de côtés d'un polygone, on limite les cas traités. L'exemple de la Figure 32 page 54 ne peut être réalisé, car il comprend une zone à 12 côtés.

2.3.1.4 Zones à 4 côtés.

Les deux premières méthodes s'étant avérées peu satisfaisantes, une troisième méthode a été envisagée ; elle se base sur un constat simple : le passage à la grille est facile si l'on ne traite que des zones à 4 côtés ou 4 angles. A chaque noeud du graphe, il suffit d'associer un rectangle. Or la recherche topologique générale présentée précédemment peut générer n'importe quelle zone ; une modification du traitement s'impose.

Suite à ce constat, on peut compléter les définitions données précédemment pour une région du graphe. Chaque zone doit être un domaine fermé convexe; il peut être décrit comme un pavé de la manière suivante :

$$Z = \{ A(x,y) \;/\; (z1 - X/2 < x < z1 + X/2) \text{ et } (z2 - Y/2 < y < z2 + Y/2) \}$$

z1 et z2 sont les coordonnées du centre Oz de la zone. Les valeurs de X,Y,z1 et z2 sont inconnues pendant la recherche topologique ; elles seront déterminées au moment du passage à la grille.

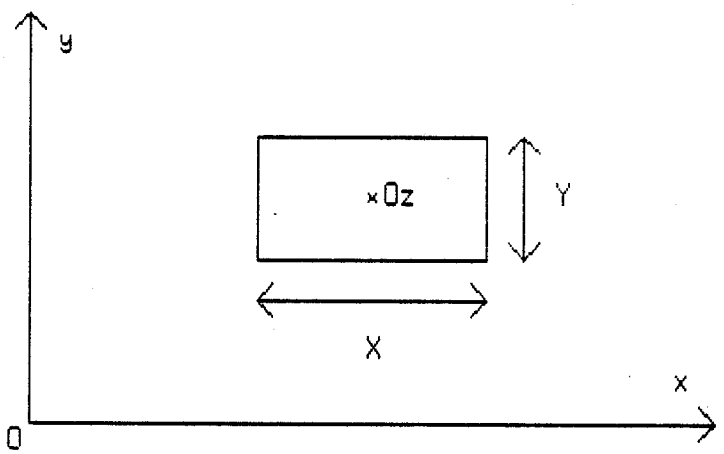


Figure 33. Présentation d'une zone à 4 côtés

Créer un chemin revient à répéter plusieurs fois la même operation : traverser un noeud du graphe. Considérons une zone à 4 côtés et une connexion entrant par un des côtés : elle peut ressortir de cette zone de 5 manières différentes (Figure 34 page 58); le choix en est fait suivant les

critères de l'algorithme (contacts, longueur du chemin, etc...).

En fait, on a seulement trois cas à traiter : les traversées 0-2 et 0-4 d'une part, les traversées 0-1 et 0-5 d'autre part demandent des traitements similaires. Précisons les :

1. la connexion sort par le point 3 ; la zone à 4 côtés est simplement divisée en deux zones à 4 côtés.

2. la connexion sort par le noeud 2 ou le noeud 4. Dans les deux premières méthodes, on avait création d'une zone à 3 côtés et d'une zone à 5 côtés qui , après transformation, devenaient des zones à 4 et 6 côtés. Dans cette approche, traverser la zone revient à faire un recouvrement de celle-ci uniquement par des domaines fermés convexes : la solution minimale est le recouvrement par 3 régions, 2 de ces régions comprenant un côté transparent (Figure 35 page 58). Les zones 2 et 3 réunies correspondent en fait à une zone à 6 côtés.

3. la connexion sort par le point 1 ou le point 5. Plusieurs solutions de recouvrement sont possibles. La solution retenue est la suivante; elle nécessite deux étapes :

   a. diviser la zone origine en 3 zones numérotées 1,2 et 3 sur la Figure 36 page 59. Les régions 2 et 3 possèdent un côté transparent en commun.

   b. diviser la zone 1 en deux zones 1 et 4; les zones 2 et 4 sont également séparées par un segment transparent.

Cette méthode fait intervenir une nouvelle notion. Chaque région était jusqu'alors définie par son contour et les régions qui lui sont contiguës. Le contour était une simple liste de points. Deux qualificatifs doivent maintenant être donnés à chacun des points:

1. sa nature : transparent ou continu . Cette notion s'applique en fait au segment reliant ce point et son suivant dans la liste. On peut l'interpréter d'une autre manière : ce segment appartient ou non à une connexion.

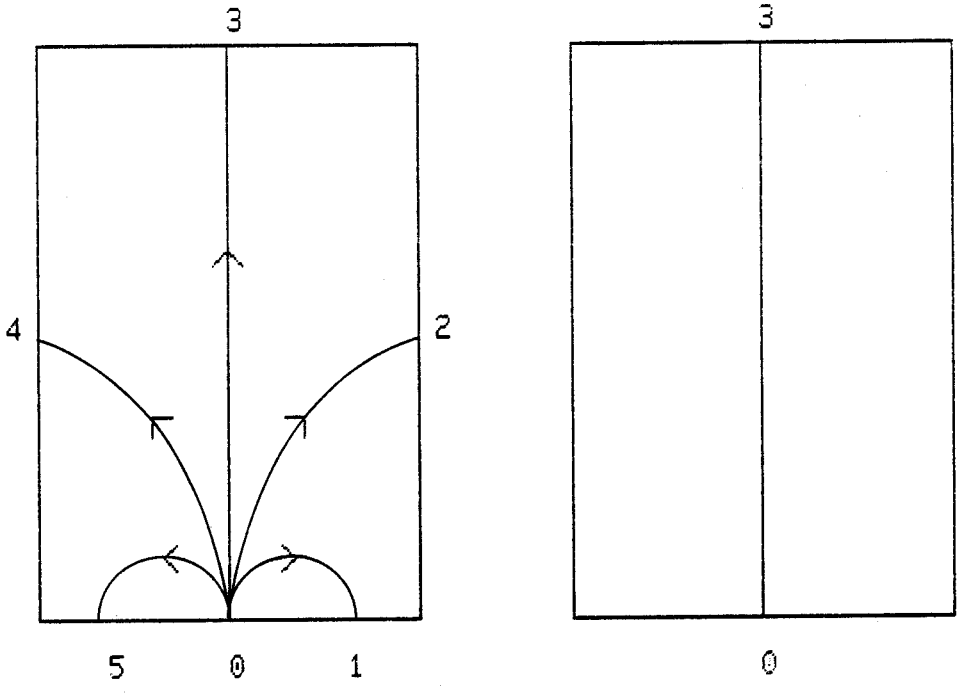


Figure 34. Traversées possibles d'une région

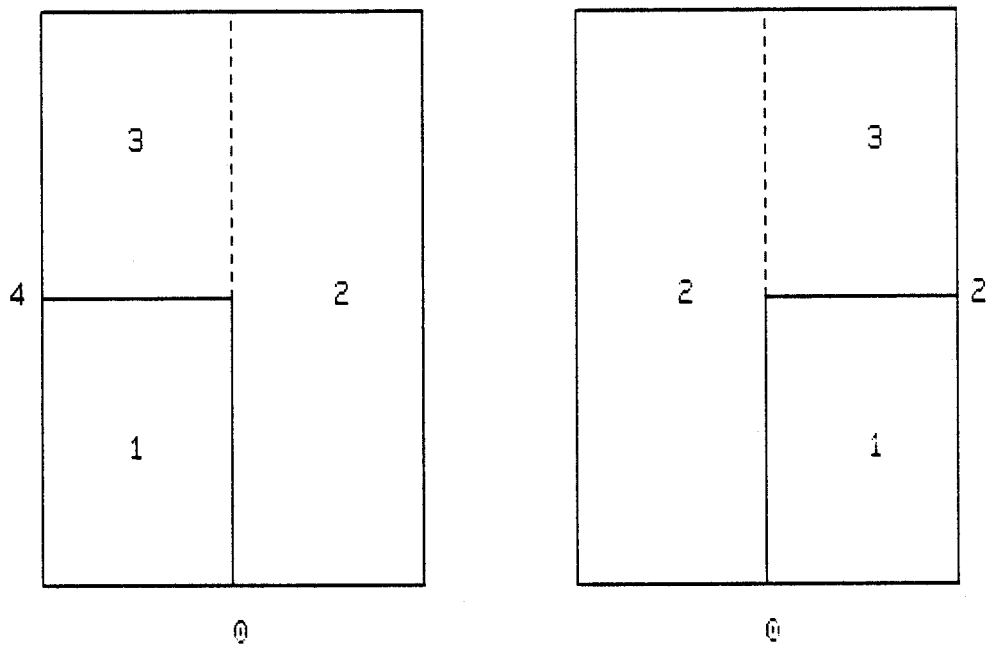


Figure 35. Traversée créant trois zones

2. sa contribution à la géométrie de la région à savoir: constitue-t-il ou non un point anguleux de la zone ?

Ce traitement répond assez bien aux critiques qui avaient été addressées aux deux premières méthodes :

- le choix de la géométrie n'influe pas sur l'algorithme de recherche du chemin.
- toutes les régions traversées, qu'elles soient internes ou externes, sont traitées de la même manière.
- il n'y a pas de borne supérieure au nombre de côtés d'une zone, en ce sens que la réunion de zones comportant un côté transparent permet d'obtenir n'importe quelle forme géométrique.

Cependant, il faudra tenir compte de la nature des segments traversés pendant la recherche du chemin. C'est une des seules difficultés posées par cette méthode.

Cette façon de procéder déplace le problème du choix de la géométrie dans la recherche topologique.

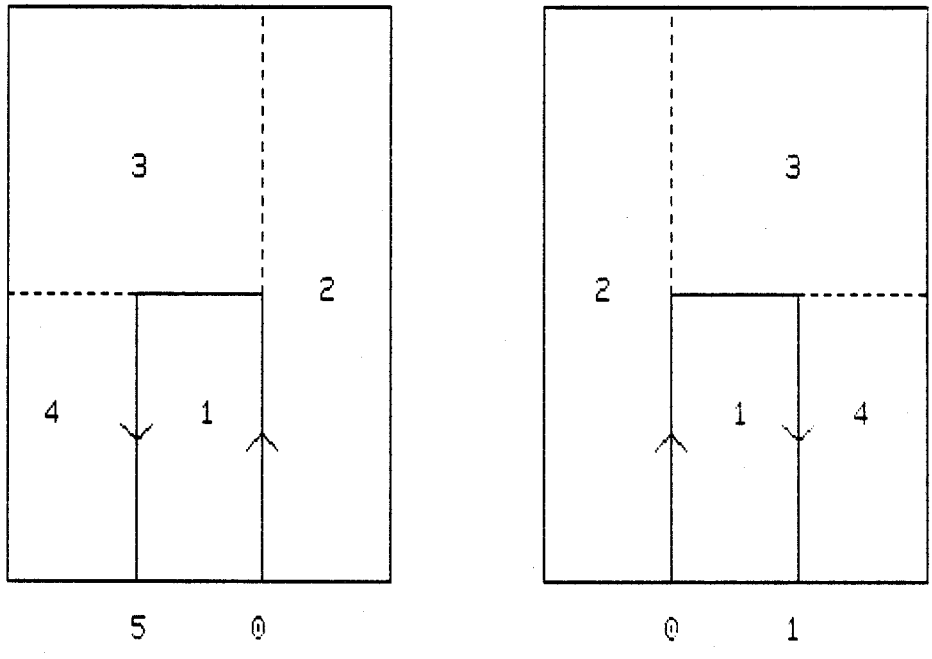


Figure 36. Traversée créant quatre zones

2.3.1.5 Méthode choisie

La dernière méthode présentée a été retenue car le choix de la géométrie est fait automatiquement pendant la recherche topologique et ne pose donc pas de problème. Le passage à la grille se limite ainsi au dimensionnement des rectangles, chaque rectangle étant associé à un noeud du graphe des régions.

Chaque segment du contour d'une région est soit continu, soit transparent selon son appartenance ou non à une connexion. Cette propriété offre de nouvelles possibilités pour le domaine origine.

Tous les exemples présentés jusqu'à maintenant prenaient pour domaine de départ un rectangle. Il est cependant possible de traiter des formes différentes : des polygones quelconques ne comportant que des angles droits ; il suffit de diviser le domaine en plusieurs rectangles, les frontières entre ces rectangles étant transparentes. L'exemple de la Figure 37 illustre cette remarque. Le graphe de départ n'est plus limité à un noeud mais comprend un noeud par rectangle du domaine origine.

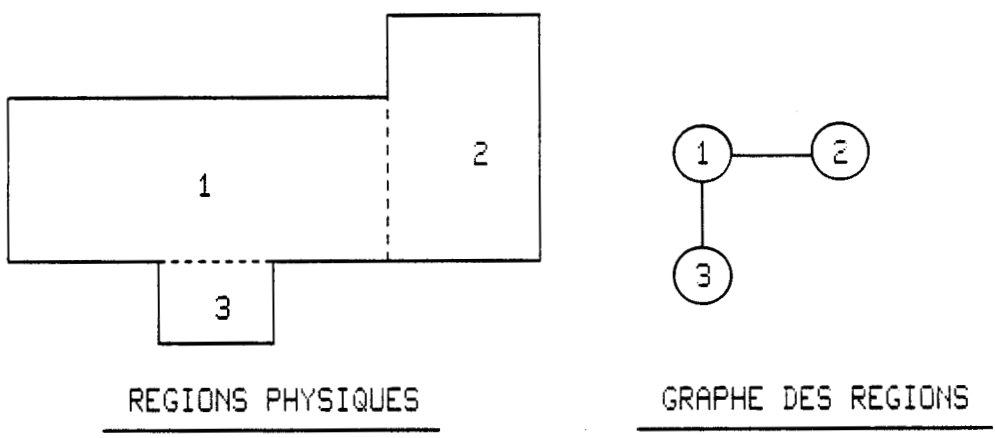


Figure 37. Domaine origine complexe

2.3.2 Dimensionnement des régions.

C'est la dernière étape du traitement. Toutes les régions doivent être placées physiquement dans le domaine initial. Chaque région est définie par :

1. la liste des régions voisines.

2. la liste des points formant son contour;

Dimensionner chaque région revient à déterminer les coordonnées de chacun de ses points. L'origine du repère utilisé est un des points angle du domaine initial noté I0. Ce point appartient à la région R0. Le point I0 est choisi de telle manière que le segment I0-N, N étant le point suivant I0 dans la liste des points de la zone R0, soit orienté suivant l'axe Ox.

On peut distinguer 4 phases pour arriver à ce résultat:

1. orienter chacune des zones.

2. rendre cohérente chacune des zones.

3. rendre cohérentes les zones les unes par rapport aux autres. Les valeurs de X et Y, longueur et largeur, définies précédemment sont déterminées pour chaque zone.

4. positionner chaque zone.

Cet algorithme sera appliqué à un exemple; le canal à traiter est donné sur la Figure 38 page 62 : il s'agit de réaliser cinq connexions. Une recherche topologique a été effectuée qui a généré un graphe des régions. Ce graphe est caractérisé par l'ensemble de ses noeuds et par les relations entre ses noeuds.

L'ensemble des noeuds est défini par une liste de zones, chaque zone est elle-même définie par une liste de points notés Ni. Un point peut être point angle ou non; dans ce dernier cas, le qualificatif 0 lui sera donné.

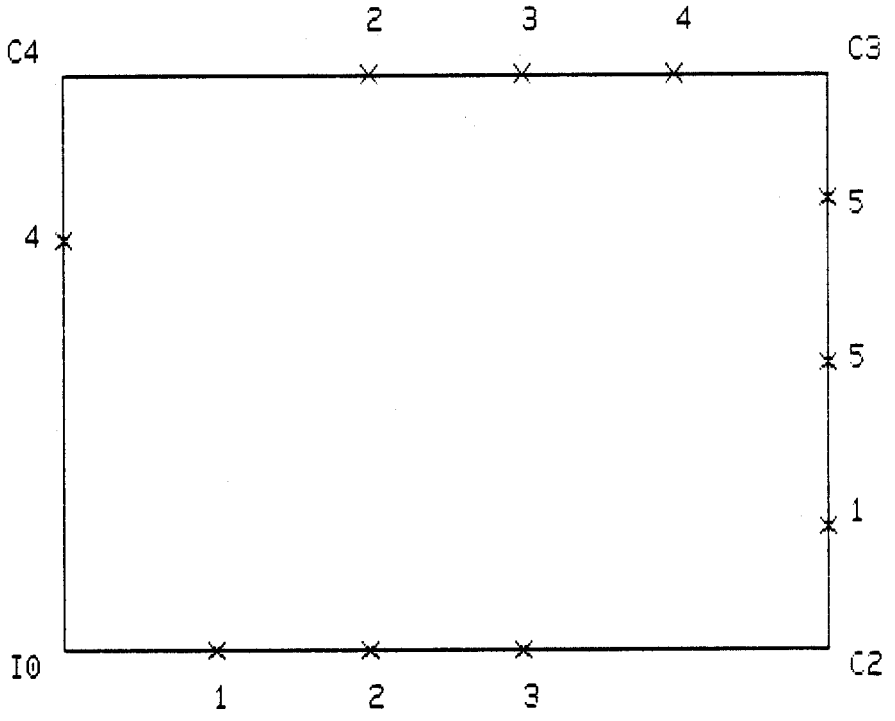


Figure 38. Exemple de canal à interconnecter

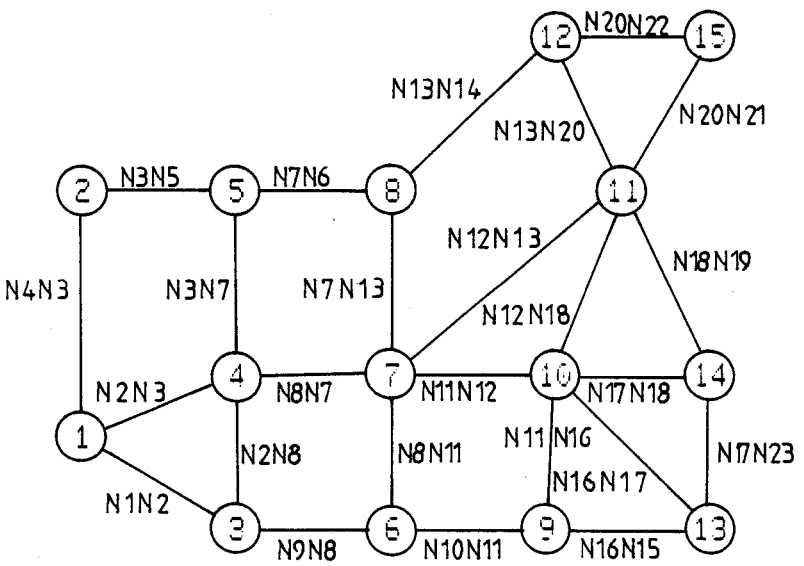


Figure 39. Données pour le dimensionnement

La liste des zones est la suivante (pour plus de précision, on pourra se reporter à la Figure 40 page 66) :

1 : I0-N1-N2(D)-N3-N4
2 : N4-N3-N5-C4
3 : N1-N9-N8-N2
4 : N2-N8-N7-N3
5 : N3-N7-N6-N5
6 : N9-N10-N11-N8
7 : N8-N11-N12(D)-N13-N7
8 : N7-N13-N14-N6
9 : N10-C2-N15-N16(D)-N11
10 : N11-N16-N17(D)-N18-N12
11 : N12-N18(D)-N19-N21-N20(D)-N13
12 : N13-N20-N22-N14
13 : N16-N15-N23-N17
14 : N17-N23-N19-N18
15 : N20-N21-C3-N22

Le graphe des régions correspondant est donné sur la Figure 39 page 62 . Pour chacune des arêtes, on a précisé le segment de contiguïté qui sépare deux régions.

2.3.2.1 Orientation des zones

La recherche topologique associe à chaque région une liste de points (D1 N1 .... D2 Ni .... D3 ... D4 ...); quatre d'entre eux D1 à D4 sont des points angles , puisque par construction chaque région est un rectangle . Cependant, on ne dispose d'aucune information sur l'orientation des segments (D1-D2),(D2-D3),(D3-D4),(D4-D1) dans le repère orthonormé (O,Ox,Oy). Sont-ils orientés suivant l'axe Ox, direction EW (Est-Ouest) ou suivant l'axe Oy, direction NS (Nord-Sud) ? Il s'agit donc dans chaque zone d'associer à chaque segment une des deux directions.

On recherche la zone contenant le point I0. Sachant que le segment suivant le point I0 est orienté EW, on détermine en parcourant la liste des points :

o les 4 côtés; chaque côté relie deux points.

- la longueur de chaque côté. La longueur est égale au nombre de points reliant les deux points angles moins un.

- l'orientation de chacun des côtés. On part d'une direction EW et elle change chaque fois que l'on rencontre un point angle dans la liste.

La zone contenant R0 est donc orientée. Les orientations associées a chaque côté de cette région vont servir à orienter les zones qui lui sont contiguës.Il suffit en se servant du graphe des régions de créer une liste, chaque élément de cette liste comprenant :

- un pointeur vers une zone voisine de R0.

- le segment de contiguïté proprement dit.

- l'orientation de ce segment

A partir de ces trois informations, il est possible de traiter chacune des zones contiguës de la même façon que la zone origine.

Le processus est répété jusqu'à ce que toutes les zones soient orientées. Cette méthode, similaire à la propagation d'une onde dans toutes les zones, sera également utilisée pendant le positionnement des zones. Prenons l'exemple des zones de la Figure 40 page 66 et précisons à chaque passe les zones traitées:

- passe 1 : la zone 1 comprenant le point I0

- passe 2 : les zones 2,3,4 voisines de la zone 1

- passe 3 : les zones 5,6,7 voisines

- passe 4 : les zones 8,9,10,11

- passe 5 : les zones 12,13,14,15.

2.3.2.2 Cohérence de chaque zone.

Chaque zone est un rectangle. Donc les côtés opposés de chaque région pris 2 à 2 doivent avoir la même longueur. Or cette longueur a été calculée en fonction du nombre de points rencontrés entre les deux points angles, ce qui nécessite de redimensionner certains côtés. C'est le cas de la zone 10 dans l'exemple Figure 40 page 66. La taille d'un des segments NS doit être augmentée.

Pour chaque zone le traitement est donc le suivant :

1. on fait la différence des longueurs des 2 segments de direction NS;

2. si le résultat trouvé est non nul, on augmente de cette différence le segment le plus petit.

L'opération est faite également pour les segments de direction EW. Cette phase transforme donc chaque zone en un vrai rectangle au sens métrique du terme.

2.3.2.3 Cohérence générale.

Dans la phase précédente, chaque région du graphe a été rendue cohérente indépendamment des autres régions et en particulier des régions qui lui étaient contiguës. Il est maintenant nécessaire de traiter tous les arcs du graphe : un arc relie deux régions suivant un segment A-B. Ce segment A-B appartient aux deux régions et sa longueur doit être la même dans ces deux zones ce qui n'est pas le cas de la liaison entre les zones 7 et 10 sur la Figure 40 page 66 .

Pour régler ce problème, la méthode est simple. Considérons chaque segment de contiguïté :

1. on calcule la différence entre sa longueur dans une des zones contiguës et sa longueur dans l'autre.

2. si ces deux valeurs ne sont pas les mêmes, on augmente de la différence le segment le plus petit. Il faut en-

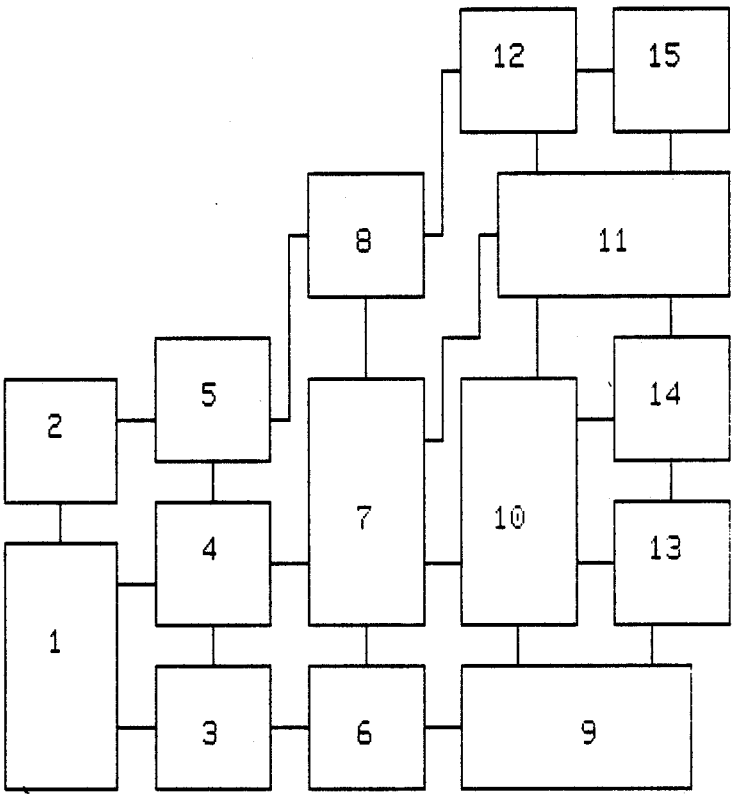


Figure 40. Résultat après orientation et cohérence

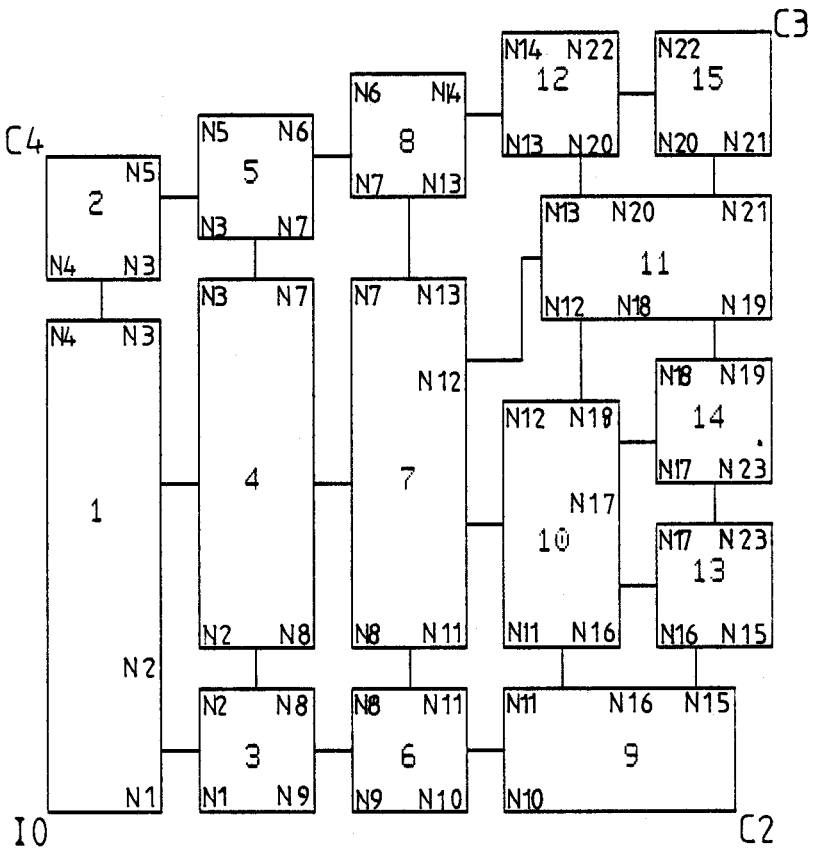


Figure 41. Résultat après la cohérence générale.

suite rendre cohérente la zone qui contient ce segment (augmenter la longueur du segment opposé).

On répète ce traitement jusqu'à ce que toutes les différences soient nulles. La taille de chaque zone est ainsi déterminée. Il reste à positionner ces zones. Les résultats obtenus après cette étape sont donnés sur la Figure 41 page 66 .

2.3.2.4 Positionnement.

Connaissant les dimensions de chaque région et le graphe des régions, il est possible de calculer, pour chaque région les coordonnées de tous ses points. L'algorithme est similaire à celui utilisé pour l'orientation :

1. on positionne la zone R0.

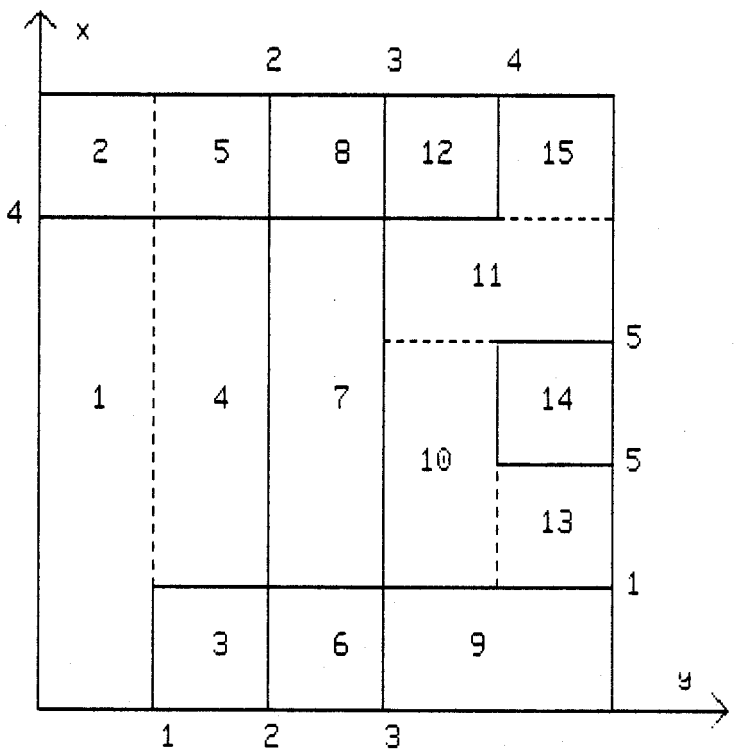


Figure 42. Résultat après positionnement

2. on génère à partir du graphe une liste des zones voisines de R0. Chaque élément de cette liste comprend :

   a. un pointeur vers la zone voisine de R0

   b. le segment de contiguïté

   c. Les coordonnées des extrémités de ce segment.

3. à partir de cette liste, les zones contiguës sont positionnées sans problème et une nouvelle liste de zones contiguës aux zones de la première liste est créée.

4. ceci est répété jusqu'à ce que toutes les zones soient positionnées.

Pour positionner les points d'une région, on parcourt la liste des points à partir d'un des points du segment de contiguïté dont on connaît les coordonnées.

2.3.3 Conclusion

Dans ce chapitre, une nouvelle méthode a été présentée ; elle comprend deux parties :

- la première comporte des algorithmes permettant de modifier une configuration en fonction du chemin à créer.

- la seconde est composée d'algorithmes qui réalisent le passage à la grille, c'est-à-dire qui génèrent une solution physique à partir d'une liste de zones. Ces zones définissent la configuration.

On peut maintenant appliquer cette méthode à des exemples concrets : quel que soit le cas, il s'agit toujours de créer des liaisons entre deux ou plusieurs points. Créer une liaison revient à rechercher un chemin satisfaisant certaines contraintes et ensuite à appliquer les différents algorithmes présentés dans ce chapitre. Un chemin est constitué d'une liste de zones à traverser.

3.0 APPLICATION DE LA METHODE AU "CHANNEL ROUTING".

Ce problème se définit simplement par les données suivantes :

- un domaine généralement rectangulaire séparant des blocs d'un circuit intégré. Dans notre cas, on ne se limitera pas à des rectangles.

- une liste de connexions, chaque connexion reliant plusieurs points placés sur le contour du domaine. On parlera de terminaux. Si le nombre de terminaux est égal à deux, on parlera de "Channel routing" bijectif.

- un ensemble de règles à respecter:

    1. on peut utiliser deux niveaux technologiques : le métal et le polysilicium.

    2. le passage d'un niveau à l'autre se fait par des contacts.

    3. des règles sont imposées pour une technologie donnée: deux lignes de polysilicium ou deux lignes de métal sont séparées par une distance minimale,etc...

Trouver une solution à ce problème revient à donner pour chaque connexion une liste des segments reliant les terminaux; pour chaque segment il faudra spécifier sa position dans le canal et son niveau technologique.

Les méthodes classiques ont été présentées dans le premier chapitre : algorithmes de Lee, Hightower, Hashimoto, du "dogleg" et de Yoshimura.

La plupart de ces méthodes réalisent l'interconnexion avec directions privilégiées; ceci signifie que tous les segments horizontaux sont sur un niveau (métal) et tous les verticaux sur l'autre (polysilicium) ou l'inverse. Il peut être intéressant d'autoriser la présence des segments horizontaux et verticaux sur les deux niveaux tout en évitant la super-

position sur de longues distances de deux connexions car il peut y avoir des problèmes capacitifs entre les deux pistes.

Les algorithmes classiques utilisent beaucoup de contacts. Minimiser ce nombre permettrait d'améliorer plusieurs facteurs :

- o le rendement en fabrication du circuit
- o la surface du circuit car un contact occupe en général plus de place qu'une ligne : la distance entre deux lignes, de métal ou de polysilicium, ne peut être minimale si un contact est placé sur une de ces lignes.

Le "channel routing" a été envisagé avec les deux objectifs ci-après:

1. ne plus utiliser systématiquement des directions privilégiées
2. minimiser le nombre de contacts.

Tandis que le présent travail était réalisé, une autre étude basée sur des principes similaires était faite à l'université de Berkeley par Chi-Ping HSU. Elle concerne le "Channel routing" bijectif et sera exposée dans un premier temps, ensuite seront présentés les algorithmes propres à cette thèse: ils permettent de traiter le "Channel routing" bijectif ainsi que le "Channel routing" général, ce dernier étant considéré comme une extension du "Channel routing" bijectif.

3.1 ALGORITHME DE HSU.

Hsu (15-16) divise le problème en deux parties :

1. un "routing" topologique qui recherche une solution topologique comportant un nombre minimum de contacts. C'est la liste orientée des terminaux qui détermine le nombre d'intersections entre connexions et donc le nombre de contacts nécessaires.

2. un passage à la grille qui transcrit la solution topologique sur un plan muni de la métrique classique. Ce sont les contraintes géométriques qui jouent.

3.1.1 "Routing" topologique.

3.1.1.1 Bipartition sur l'ensemble des connexions.

Le contour de la région est représenté par un cercle sur lequel sont placés les terminaux. Seul l'ordre des terminaux importe pour générer le graphe d'intersection des connexions Gi qui est défini comme suit :

- un noeud du graphe correspond à une connexion.
- un arc relie deux noeuds du graphe si les deux connexions se croisent.

Le principe de l'algorithme est simple : un ensemble de connexions peut être réalisé sans contact si l'on peut effectuer une partition de cet ensemble en deux sous-ensembles qui vérifient chacun la propriété suivante : deux connexions quelconques d'un sous-ensemble ne se croisent pas. Dans ce cas, toutes les connexions d'un sous-ensemble peuvent être placées sur un même niveau technologique. Un telle partition de l'ensemble des connexions est possible si et seulement si le graphe Gi d'intersection des connexions est biparti.

Donc pour traiter un problème d'interconnexion quelconque, on recherche dans un premier temps le nombre maximum de connexions pouvant être placées sans contact, ce qui revient à trouver le nombre minimum de connexions à supprimer dans le graphe Gi pour qu'il soit biparti. Puis dans un second temps, on pourra placer les connexions restantes qui nécessitent l'utilisation de contacts.

Reprenons l'exemple donné dans l'article (15) de trois fils qui se croisent. La Figure 43 page 72 présente le

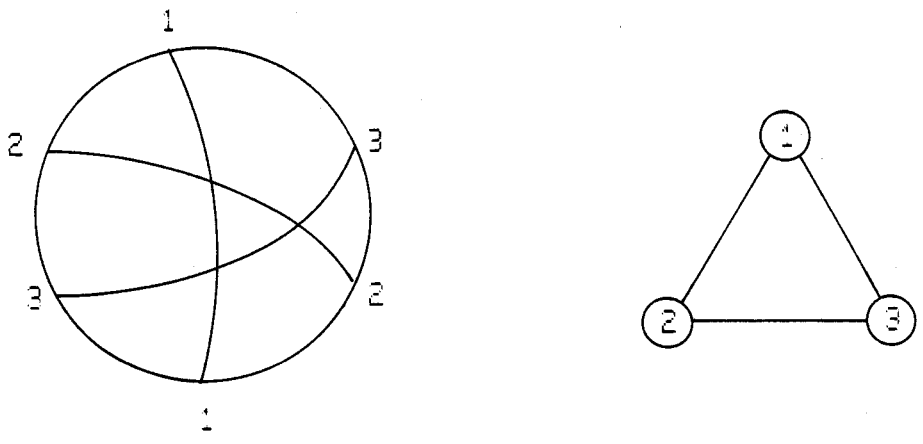


Figure 43. Cercle de Hsu et graphe d'intersection Gi

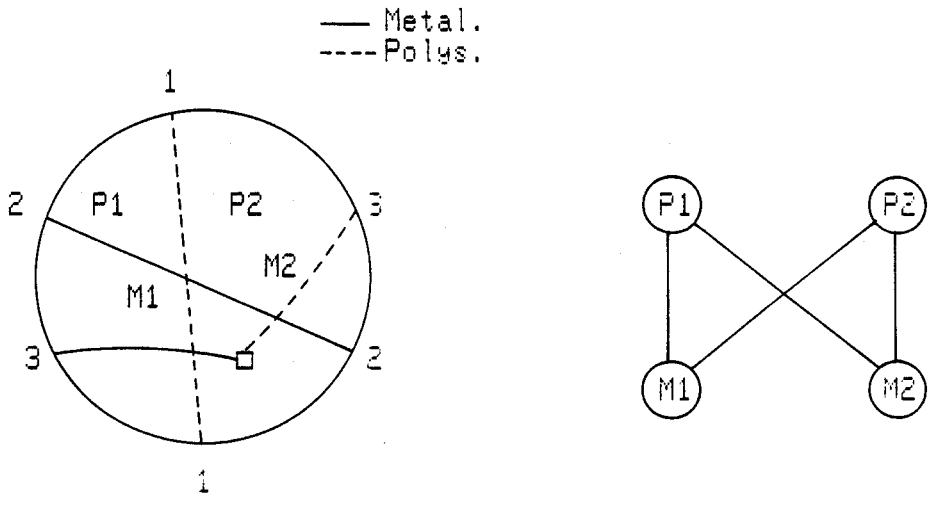


Figure 44. Régions et graphe des régions

cercle et le graphe Gi correspondant. La bipartition du graphe est très simple : on supprime la connexion 3 du graphe et les connexions 1 et 2 peuvent être placées sans contact; par exemple, la connexion 1 peut se faire en polysilicium, la connexion 2 en métal.

3.1.1.2 Création des connexions

Les deux sous-ensembles de connexions obtenus à partir du graphe biparti sont affectés chacun à un des niveaux technologiques. Lorsque l'on place les connexions d'un des ensembles sur le plan correspondant, celui-ci est divisé en régions. La Figure 44 page 72 illustre cette notion de région : dans l'exemple présenté précédemment, la connexion 1 (respectivement 2) divise le plan du métal (resp. polysilicium) en deux régions M1 et M2 (resp. P1 et P2).

A partir de cette notion de région, Hsu crée un graphe Gr de recouvrement des régions défini ainsi :

- chaque noeud représente une région
- il y a une arête entre deux noeuds si les deux régions se superposent.

Ce graphe est utile pour traiter les connexions qui nécessitent des contacts : chaque connexion relie deux terminaux; tous les deux appartiennent à deux régions du graphe, une région sur chaque niveau. Relier les deux terminaux revient à chercher le chemin le plus court dans le graphe Gr entre une des régions de départ et une des régions d'arrivée. Le nombre d'arêtes parcourues dans le graphe sera égal au nombre de contacts puisqu'une arête correspond à la superposition de deux régions.

Dans l'exemple traité, la connexion 3 a son origine dans une des régions (M1,P1) et sa destination dans (M2,P2); les deux chemins les plus courts sont M1-P2 et P1-M2. Pour chaque connexion nécessitant des contacts, c'est la méthode adoptée.

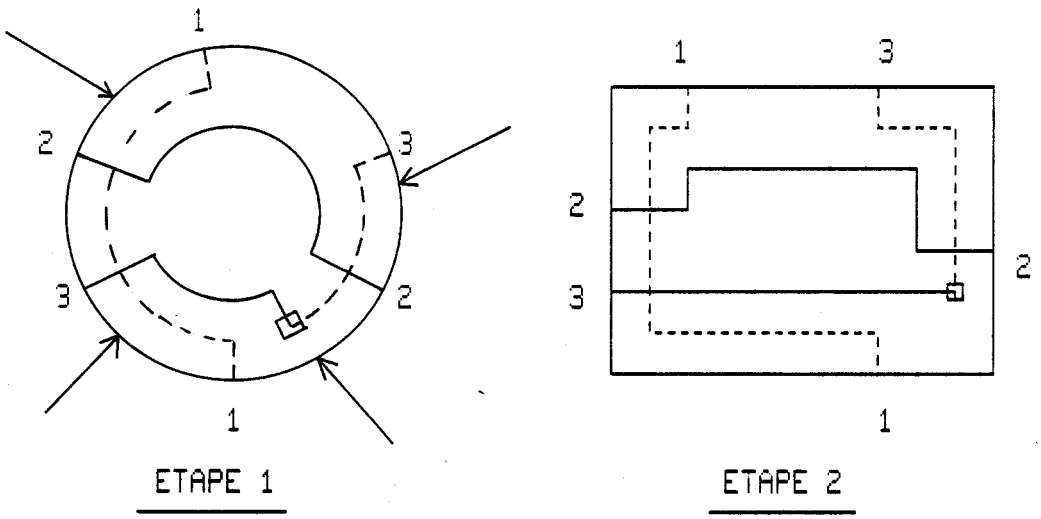


Figure 45. Passage à la grille.

3.1.1.3 Optimisation.

Dans certains cas, plusieurs solutions de bipartition sont possibles et suivant le choix effectué, le nombre de contacts utilisés est plus ou moins grand. De nombreux algorithmes sont développés dans l'article de Hsu (15) qui permettent de réaliser une bipartition optimale.

3.1.1.4 Passage à la grille.

Partant de la solution topologique, les connexions sont placées sur des pistes circulaires comme le montre l'exemple de la Figure 45 . La position des quatre coins du domaine est repérée par des flèches sur le cercle. Par des transformations appropriées tenant compte des quatre coins, les pistes circulaires deviennent des pistes verticales et horizontales. Une fois ce résultat obtenu, des algorithmes de compaction sont appliqués sur ce schéma.

3.2 METHODE ETUDIEE.

La méthode que nous avons étudiée est présentée d'abord pour le "Channel routing" bijectif; son extension au "Channel routing" général est donnée au paragraphe 3.3. Cette méthode comprend de manière analogue à celle de Hsu deux grandes étapes que l'on peut préciser ici :

1. la recherche topologique. Elle s'effectue de la manière suivante:

   - o trouver dans l'ensemble des connexions un sous-ensemble dont les connexions pourront être placées sans contact.

   - o pour chaque connexion, chercher un chemin reliant la zone origine et la zone destination. Pour les connexions nécessitant des contacts, il s'agit de minimiser leur nombre. Le graphe des régions est ensuite modifié en fonction de ce chemin ; on utilise les algorithmes présentés dans le chapitre précédent.

2. Le passage à la grille. Il comprend essentiellement le dimensionnement des régions du graphe. Cependant, il est précédé d'une étape supplémentaire: il s'agit d'affecter chacun des segments formant le contour des régions à un niveau technologique et également de déterminer la position des contacts. Cette phase s'appelera la phase du coloriage des régions.

3.2.1 Comparaison avec l'algorithme de Hsu.

A part la première partie de la recherche topologique, cette méthode se différencie assez bien de celle de Hsu et ce particulièrement sur deux points :

1. la recherche topologique ne fait pas intervenir la notion de niveau technologique. Il n'est pas nécessaire de trouver une bipartition optimale de l'ensemble de

connexions comme le fait Hsu; le problème est résolu automatiquement par le coloriage des zones.

2. les deux étapes sont liées car le choix de la géométrie est fait implicitement dans la recherche topologique. Notre méthode divise le domaine à interconnecter en régions alors que l'algorithme de Hsu divise chaque niveau technologique en régions. Ces régions ne lui servent que pour la création de connexions. Son passage à la grille est indépendant de ces régions.

3.2.2 Algorithmes propres au "Channel Routing"

Les algorithmes concernant la modification du graphe des régions lors de la création d'une connexion et le dimensionnement des régions ont été développés en détail dans le chapitre précédent. Dans cette partie, trois algorithmes seront présentés :

1. l'algorithme permettant de trouver les connexions pouvant être placées sans contact.

2. l'algorithme de recherche d'un chemin.

3. l'algorithme réalisant le "coloriage" des connexions.

Pour les généralités concernant les graphes et leurs propriétés, on pourra se reporter à l'annexe A.

3.2.2.1 Algorithme de bipartition.

L'algorithme est le même que celui utilisé par Hsu et déjà évoqué dans le paragraphe 3.1.1.1 . Le contour de la région est représenté par un cercle sur lequel sont placés les terminaux. Le graphe d'intersection des connexions est ensuite généré. Ce graphe tel qu'il a été décrit précédemment est un graphe cercle.

Le but de l'algorithme est d'extraire de ce graphe un sous-graphe biparti, c'est-à-dire d'enlever un minimum de noeuds (connexions) du graphe de telle manière qu'il soit biparti, que les connexions puissent être placées sans contact. Pour certains types de graphe, des mathématiciens (17) ont montré que ce problème était NP-complet c'est-à-dire que le temps de calcul nécessaire pour résoudre le problème n'est pas une fonction polynomiale de la taille du problème. La question n'a pas encore été tranchée pour les graphes cercles.

Donc au lieu de chercher la solution exacte, on va approximer la solution en se servant d'une propriété des graphes cercles mise en évidence par GAVRIL (18) : sachant qu'un ensemble de noeuds d'un graphe est dit indépendant s'il n'existe pas d'arête entre deux quelconques de ses noeuds, GAVRIL a montré que l'ensemble indépendant le plus grand dans un graphe cercle peut être trouvé en un temps qui varie polynomialement avec la taille du problème.

Si C est l'ensemble des connexions et si I définit l'ensemble des croisements entre connexions, le graphe cercle G(C,I) peut être traité de la manière suivante :

1. trouver un sous-ensemble indépendant maximum nommé C1 dans le graphe G(C,I).

2. trouver un sous-ensemble indépendant maximum nommé C2 dans le sous-graphe de G(C,I) engendré par (C - C1). Le sous-graphe de G engendré par (C1 U C2) est biparti.

3. pour chaque connexion appartenant à (C - C1UC2), on essaiera de l'ajouter à (C1 U C2) de telle manière que le sous-graphe engendré par (C1 U C2) soit encore biparti.

Prenons l'exemple de la Figure 46 page 78; C est l'ensemble des connexions numérotées de 1 à 15, l'ensemble C1 est égal à (1,5,7,9,10,13), C2 est égal à (3,4,14,15). Dans l'Annexe A, l'algorithme de GAVRIL sera expliqué en détail.

Cette première étape dans la recherche topologique a été rajoutée car elle permet d'accélérer la recherche d'un chemin pour une connexion donnée. Mais elle n'est pas vraiment nécessaire car la recherche d'un chemin telle qu'elle sera

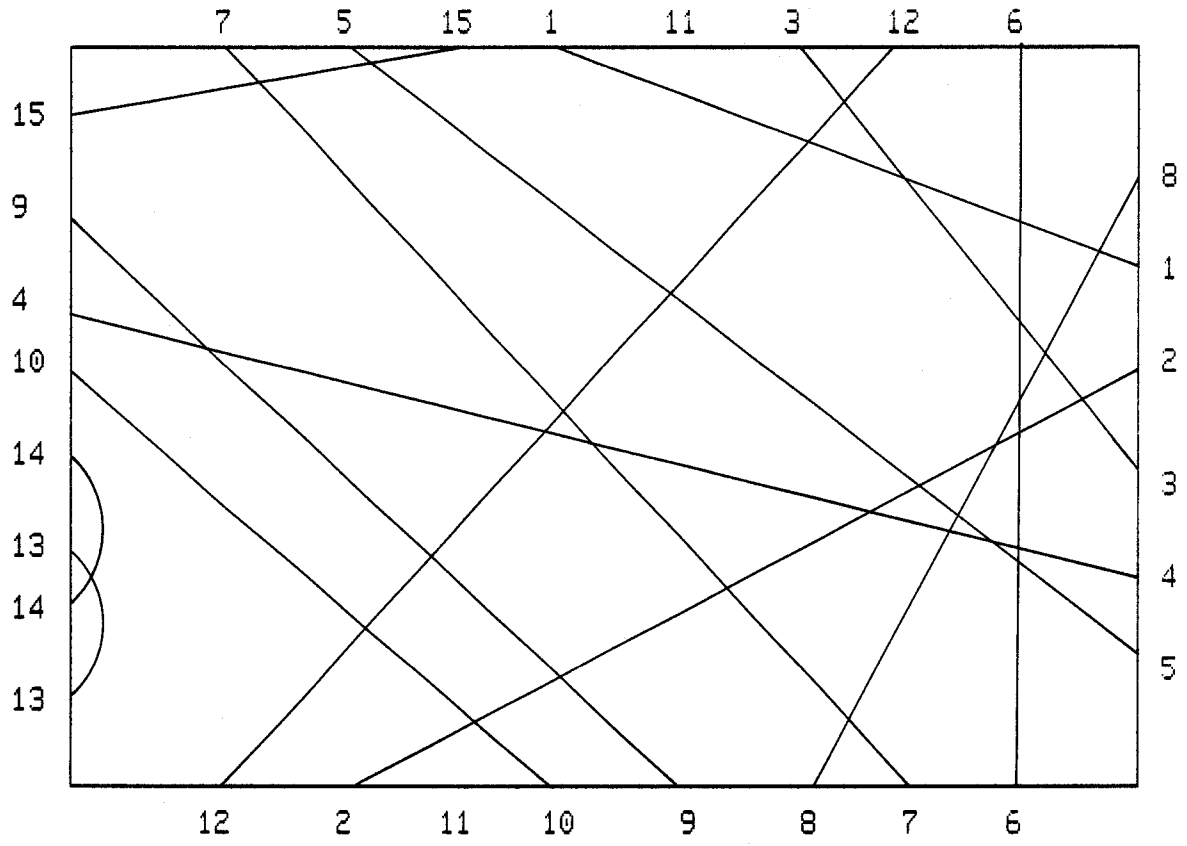


Figure 46. Exemple pour la bipartition.

définie par la suite permet de générer toutes les solutions possibles en exigeant toutefois des temps de calcul nettement plus plus importants. On traite donc les connexions dans l'ordre suivant: celles appartenant à C1 puis celles appartenant à C2 et enfin les autres qui nécessitent des contacts.

3.2.2.2 Recherche d'un chemin.

Le problème s'énonce simplement. Une connexion est définie par ses deux terminaux; chaque terminal appartient à une région du graphe des régions : pour relier les deux terminaux, dans le cas non trivial où les deux régions origine et destination sont distinctes, il faut trouver un ou plusieurs chemins qui vérifient une ou plusieurs contraintes. Le chemin trouvé sera une liste des régions à traverser.

Critères utilisés pendant la recherche

La recherche se fait en parcourant le graphe des régions; arrivant dans une région, on détermine quelles sont les régions contiguës qui pourront ensuite être traversées. On examine en particulier la nature des segments de contiguïté la séparant des autres zones. Ceci se fait en fonction de plusieurs critères:

1. ne pas traverser deux fois une connexion déjà placée. Cette contrainte est utilisée pour une raison simple : si l'on traverse deux fois une même connexion, son chemin en sera allongé. Certes, on peut obtenir de meilleurs résultats pour ce qui est du nombre de contacts utilisés si on l'autorise mais c'est au détriment de la surface. Sadowska (19) a, dans une étude prolongeant celle de Hsu, réusssi à diminuer le nombre de contacts mais en autorisant les traversées multiples.

2. ne pas traverser deux fois la même région.

3. tenir compte du graphe des intersections G(C,I). Seules sont autorisées les traversées de connexion devant vraiment croiser la connexion pour laquelle on cherche on chemin. Cette règle est un peu redondante de la première mais elle permet d'accélérer le processus.

4. lorsque l'on traverse une région fermée, c'est-à-dire ne comportant aucun segment transparent sur sa frontière, la traversée doit être directe. En raison des critères précédents, les régions fermées sont limitées à deux types:

   a. la région comprend quatre côtés appartenant à quatre connexions différentes.

   b. deux côtés consécutifs de la région appartiennent à la même connexion.

   La traversée est directe si, entrant par un des côtés de la région, on en sort par le côté opposé. Cette règle a pour but de créer un ou plusieurs noyaux internes identiques à ceux présentés précédemment (paragraphe 2.3, traitement privilégié des zones internes).

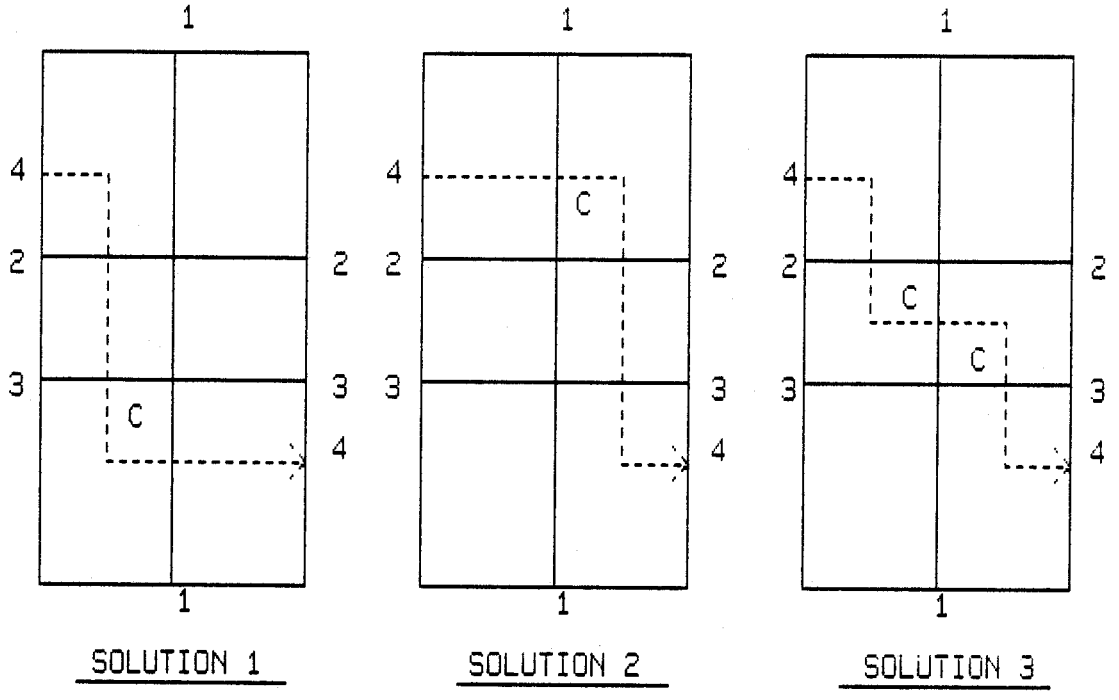


Figure 47. Choix d'un chemin.

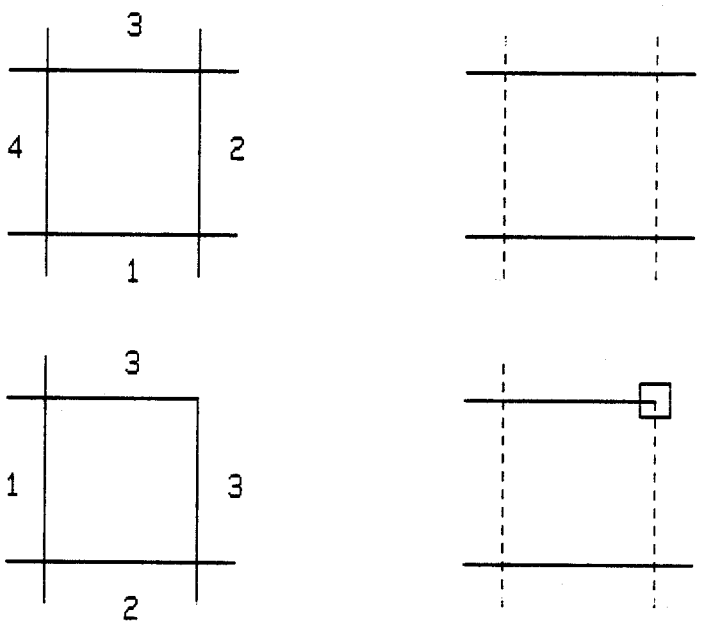


Figure 48. Coloriage des zones fermées.

Choix des chemins

Appliquant ces critères pendant la recherche d'un chemin dans le graphe, on trouve un ou plusieurs chemins reliant la zone origine et la zone destination. Parmi ces chemins, il faut choisir ceux qui créent le minimum de contacts. La notion de contact n'a pas encore été évoquée jusqu'à présent : en fait, il y a contact chaque fois que l'on a une zone fermée du deuxième type (deux côtés consécutifs appartenant à la même région). La raison en sera expliquée plus tard.

Pour illustrer ce choix du minimum de contacts, considérons les exemples de la Figure 47 page 80: supposons que l'on cherche un chemin reliant les deux terminaux de numéro 4; trois solutions sont possibles : les solutions 1 et 2 créent un seul contact puisqu'il y a une seule zone à trois côtés. La solution 3 en crée deux, donc elle ne doit pas être retenue.

Le nombre de régions traversées est également minimisé. Ceci permet d'obtenir le chemin le plus direct. Cette contrainte a beaucoup moins d'importance que le nombre de contacts, car de toute façon, chaque connexion doit croiser un ensemble bien défini de connexions. Seul l'ordre dans lequel les connexions sont rencontrées change d'un chemin à un autre.

Un ou plusieurs chemins sont ainsi déterminés. Chaque chemin est une liste de zones à traverser; on a en fait une liste d'enregistrements, chacun comprenant :

- la zone à traverser,
- le segment de contiguïté entre cette zone et sa suivante dans la liste,
- des informations concernant ce segment : sa nature, le numéro de la connexion auquel il appartient.

Le graphe des régions est ensuite modifié en fonction du chemin trouvé. S'il y a plusieurs chemins satisfaisant les contraintes établies, plusieurs solutions sont générées.

3.2.2.3 Coloriage des zones.

La génération du graphe des régions n'a pas fait intervenir la notion de niveau technologique. Avant de dimensionner les régions du graphe tel qu'on l'a décrit dans le chapitre précédent, il s'agit d'affecter chaque segment formant la frontière d'une région à un des deux niveaux.

Zones fermées.

Avant d'expliquer l'algorithme du coloriage, on peut préciser l'intérêt des zones fermées. Les zones fermées du premier type (quatre côtés appartenant à quatre connexions différentes) ne nécessitent pas de contact : il suffit de donner aux côtés opposés le même niveau technologique (Figure 48 page 80). Pour le second type de zone fermée, on assigne également les côtés opposés au même niveau; mais pour assurer la continuité de la connexion appartenant à deux côtés consécutifs, il faut utiliser un contact.

Initialisation

L'algorithme fonctionne de manière un peu analogue à ceux utilisés pour l'orientation et le positionnement des régions en ce sens qu'il y a propagation de l'algorithme à travers les zones à la manière d'une onde. La source de l'onde peut être soit une région fermée, soit une région dont un des côtés appartient à la frontière du domaine à interconnecter.

Si la zone est fermée, on colorie tous ses côtés comme on l'a montré précédemment: on part d'un des points angles et l'on parcourt tous les points du contour en changeant de couleur chaque fois que l'on rencontre un point angle.

Si la zone origine n'est pas fermée, on recherche dans cette zone un point placé sur la frontière du domaine. Puis on parcourt la liste des points de cette région jusqu'à trouver un point correpondant à un côté transparent ou à la frontière du domaine; on change de niveau chaque fois que l'on change de connexion (Figure 49 page 83).

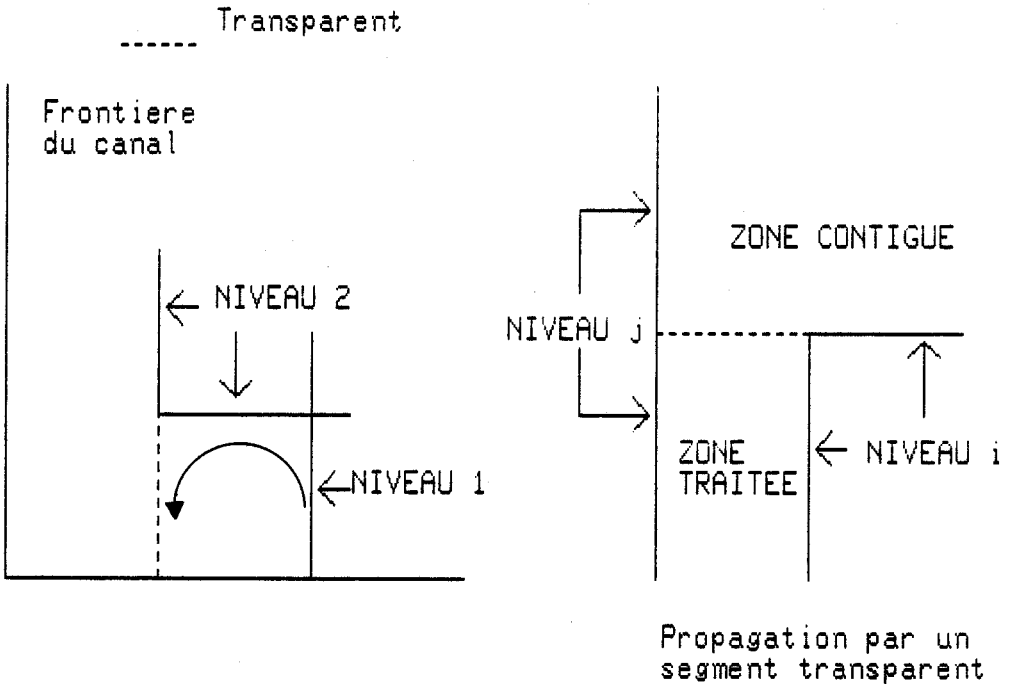


Figure 49. Zone origine non fermée.

La zone origine est donc coloriée. Les couleurs associées à chacun de côtés de cette région vont servir à colorier les zones qui lui sont contiguës. Il suffit en se servant du graphe des régions de créer une liste, chaque élément de cette liste comprenant :

- un pointeur vers une zone voisine de la zone source.
- le segment de contiguïté.
- la couleur de ce segment.

Si le segment est transparent, on le met également dans cette liste mais on ajoute dans ce cas deux informations :

1. la couleur du côté précédant ce segment dans la zone traitée.
2. la couleur du côté suivant ce segment.

Ces deux couleurs correspondent à des connexions qui appartiennent à la zone traitée et qui se prolongent dans la zone contiguë suivant le segment transparent (Figure 49).

Propagation

A partir des informations comprises dans la liste décrite ci-dessus, il est possible de traiter chacune des zones contiguës de la même façon que la zone source : partant du segment de contiguïté, on parcourt tous les points de la région; on assigne un niveau technologique à chacun en changeant de niveau chaque fois que l'on change de connexion (sauf cas particulier des zones fermées créant des contacts); on s'arrête lorsque l'on rencontre un point transparent. Lorsque le segment de contiguïté est transparent, on parcourt de la même manière les points précédant et les points suivant ce segment.

Le processus est répété jusqu'à ce que toutes les zones soient coloriées. Les régions sont ensuite orientées et positionnées comme on l'a décrit dans le chapitre 2.

3.2.3 Résultats.

3.2.3.1 Exemples traités.

Dans les pages qui suivent sont présentés quelques problèmes d'interconnexion résolus par la nouvelle méthode présentée dans ce rapport. Pour les deux premiers exemples, on donne deux schémas: le premier est le résultat obtenu en appliquant simplement la méthode, le second est le résultat obtenu après compaction de la figure suivant les deux directions.

La Figure 50 page 85 illustre le problème suivant: 9 connexions reliant des terminaux placés sur les côtés opposés d'un canal doivent être réalisées; les deux séquences de terminaux sont inversées l'une par rapport à l'autre. Le nombre de contacts obtenus est minimum et est égal à 7. Le temps de calcul est de 4,5 secondes CPU.

Le second exemple se rapproche assez du premier: on traite le cas de deux séquences de 6 terminaux également

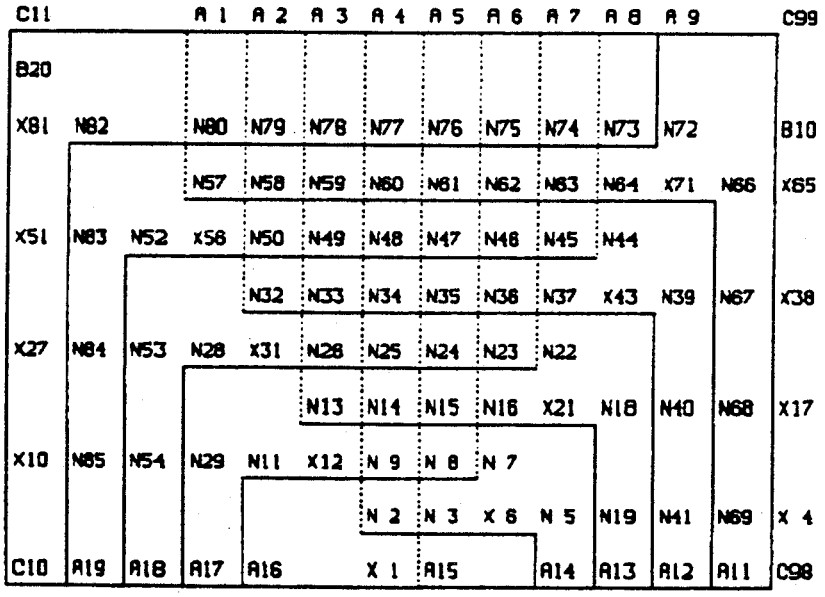


Figure 50. Exemple 1.

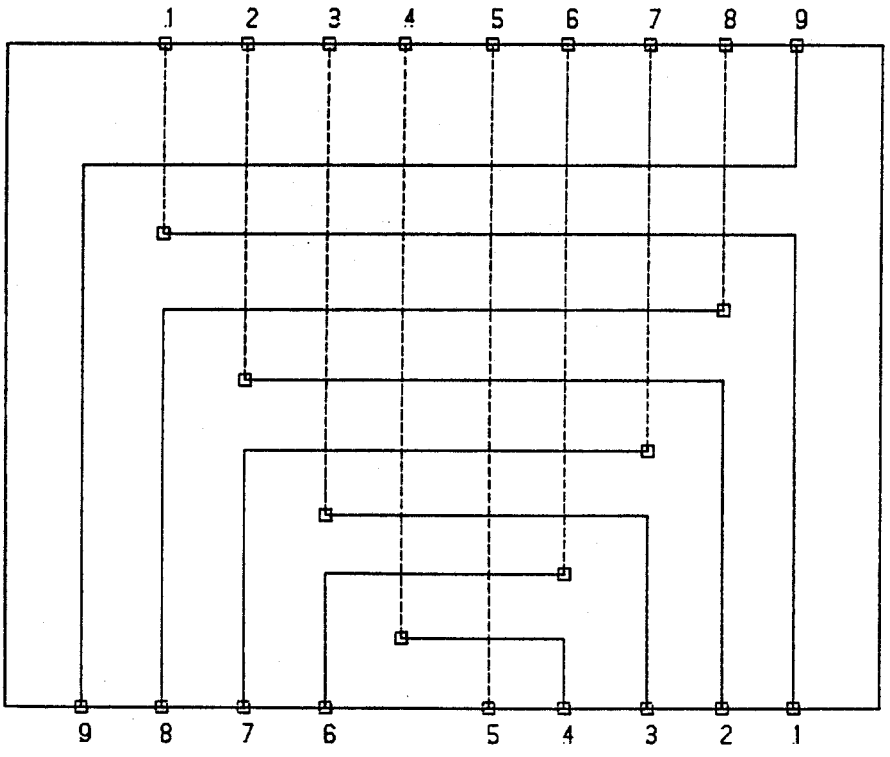


Figure 51. Exemple 1 compacté.

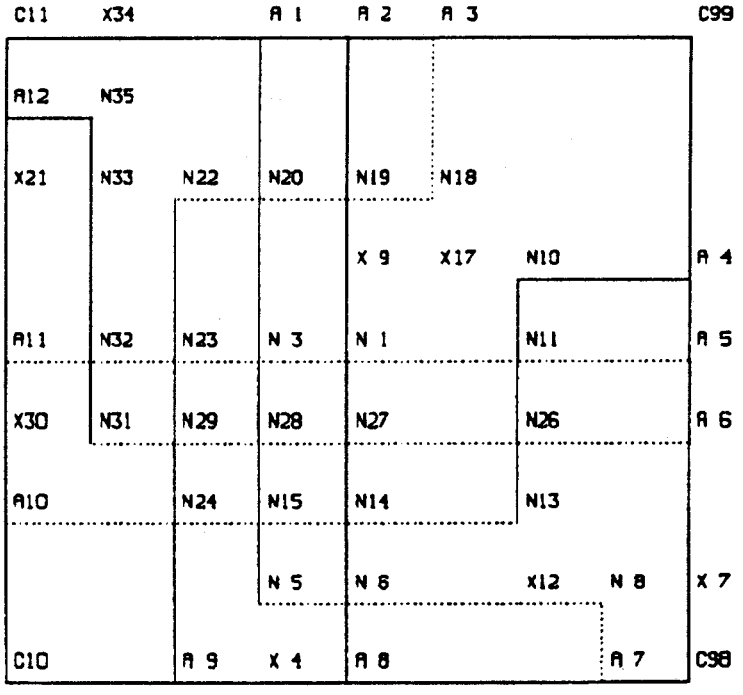


Figure 52. Exemple 2.

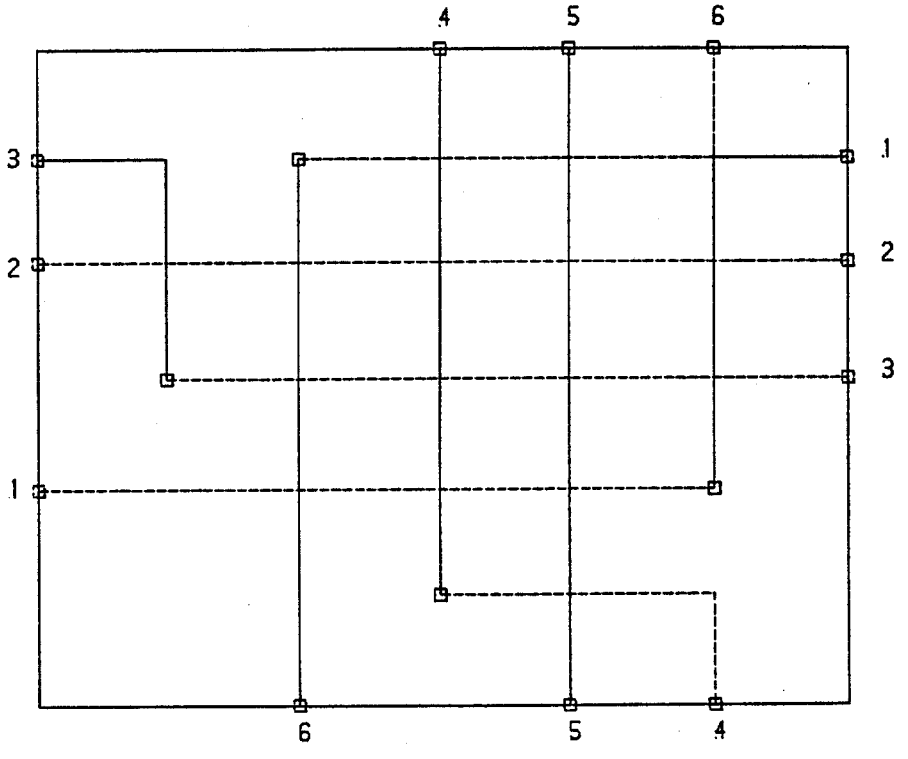


Figure 53. Exemple 2 compacté.

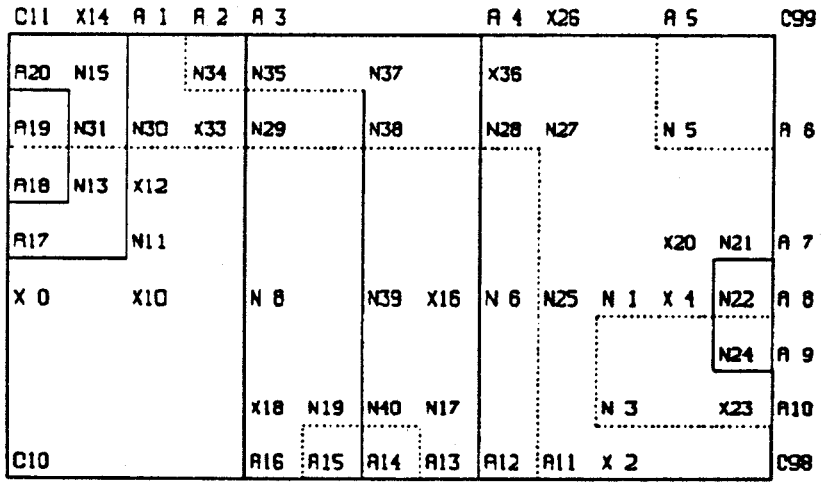


Figure 54. Exemple 3.

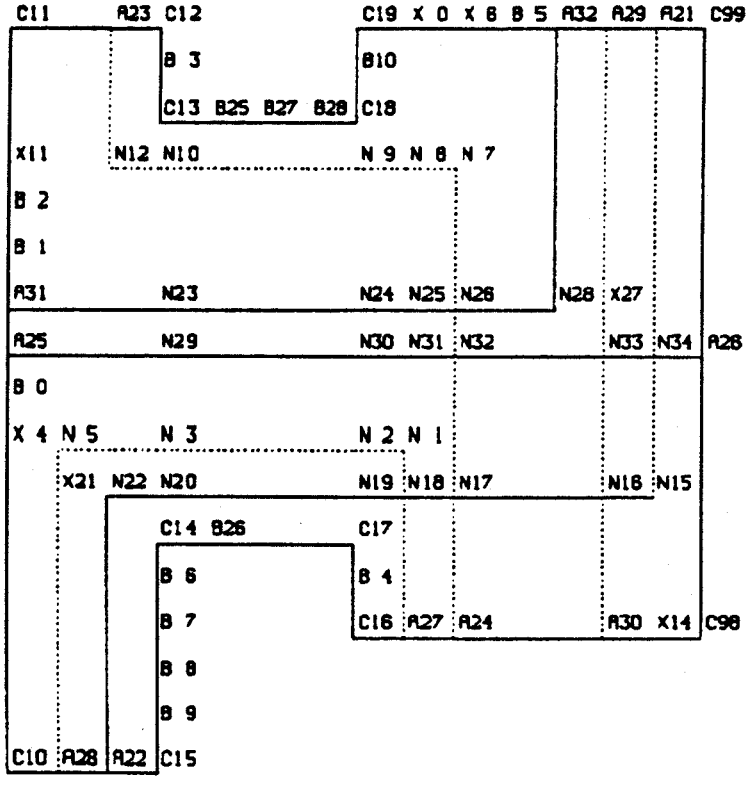


Figure 55. Exemple 4.

inversées l'une par rapport à l'autre. Cependant dans ce cas, une séquence de terminaux est placée sur deux côtés consécutifs du canal. On a minimisé le nombre de contacts qui est égal à 4. Ce problème a nécessité 1,5 s. de temps CPU (Figure 52 page 86).

Le troisième exemple est tiré de l'article de Hsu (15). Il illustre le fait que le coloriage permet de résoudre automatiquement le problème des contacts et qu'il n'est pas nécessaire dans notre méthode de réaliser une bipartition optimale du graphe d'intersection des connexions. Pour colorier, on part de la région créant un contact et l'algorithme de coloriage se propage sur les zones voisines assignant automatiquement le niveau technologique correct aux différents segments (Figure 54 page 87).

Le dernier exemple illustre la possibilité de ne pas se limiter à des domaines d'interconnexion rectangulaire. Dans ce cas, le domaine origine est formé de trois zones (Figure 55 page 87).

3.2.3.2 Minimisation des contacts.

Dans la plupart des exemples traités, le nombre de contacts est minimal. Si N est le nombre de connexions à réaliser, il existe une solution comportant (N-2) contacts. Montrons le en prenant le cas le plus complexe: considérons un ensemble de N connexions tel que chaque connexion croise les (N-1) autres connexions; les terminaux correspondants sont placés sur les côtés opposés d'un canal.

Prenons les deux couples de terminaux situés le plus près de l'extrémité du canal et traitons les deux connexions correspondantes: chacune d'elles est placée sur un niveau technologique. Les deux plans affectés aux deux niveaux sont chacun divisés en deux régions. Pour chacune des (N-2) connexions restantes, les deux terminaux n'appartiennent pas à la même région; chaque connexion nécessite un contact qui permet en changeant de plan de relier la région origine et la région destination de la connexion. Donc (N-2) contacts sont suffisants pour traiter le cas le plus complexe.

Dans les deux premiers exemples, la complexité est maximale puisque chaque connexion croise toutes les autres connexions. On trouve bien un nombre de contacts égal à (N-2) lorsque le nombre de connexions est égal à N.

3.2.3.3 Remarques.

Deux critiques peuvent être faites sur cette méthode:

1. l'algorithme impose la position des terminaux sur la frontière du canal.

2. l'algorithme impose le niveau technologique des terminaux des connexions.

Position des points de sortie.

L'ordre des terminaux sur la frontière du canal est correct après le traitement. Cependant dans certains cas, la position donnée par l'algorithme n'est pas la même que celle imposée par l'environnement extérieur représenté par les blocs qui entourent le canal: le problème est donc de relier les points de sortie donnés par l'algorithme et les points de sortie imposés par l'environnemnt extérieur.

Ce problème a déjà été traité en particulier par Pinter (20) et Hsu (21) sous le nom de "River routing"; c'est un problème de "Channel routing" bijectif avec la restriction suivante : deux connexions ne se croisent jamais. Pour le problème qui nous intéresse, on définit une seconde frontière pour le canal qui entoure la première (Figure 56 page 90). Sur cette seconde frontière sont placés les points de sortie définis par l'environnement extérieur. Le domaine à traiter est donc la surface comprise entre les deux frontières.

Pour traiter ce problème, on divise le domaine en quatre domaines : deux horizontaux et deux verticaux. Dans chaque domaine, on applique un algorithme de "River routing" classique. Le problème peut ainsi se résoudre, cependant il est

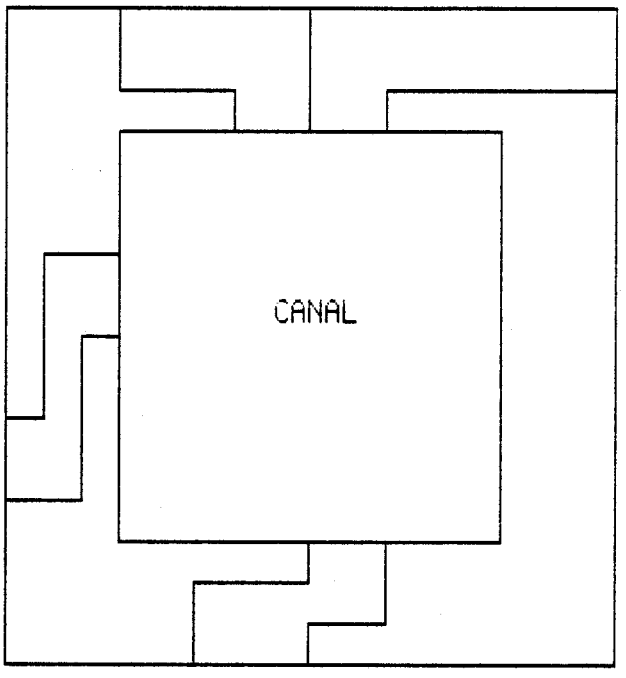


Figure 56. "River routing".

possible de mieux tenir compte de la position des terminaux sur la frontière.

En effet, sur le contour initial du canal, on peut rajouter des points fictifs lorsque deux terminaux consécutifs du contour ne sont pas séparés par la distance mimimale autorisée entre deux terminaux. Ces points ne servent à rien pendant la recherche topologique mais sont utilisés pendant le dimensionnement des régions.

Niveau technologique des points de sortie.

Le fait de vouloir imposer le niveau technologique des terminaux va à l'encontre d'un des objectifs de l'algorithme qui est la minimisation du nombre de contacts. Si l'on impose les niveaux technologiques, il n'est pas toujours possible de trouver la solution topologique optimale.

3.3 EXTENSION AU "CHANNEL ROUTING" GENERAL.

Le "Channel Routing" général se distingue aisément du "Channel routing" bijectif: chaque connexion, on parlera d'équipotentielle, peut comporter plus de deux terminaux. La méthode développée pour cette application essaie de tirer parti des résultats obtenus pour le "Channel Routing" bijectif. Elle comporte deux étapes:

1. la première considère chaque équipotentielle et génère pour chacune une liste des connexions à réaliser; c'est la division d'une équipotentielle.

2. la seconde effectue le traitement proprement dit: on réalise le placement de toutes les connexions associées aux équipotentielles.

3.3.1 Division des équipotentielles.

Définissons deux termes associés à ce problème: connexion principale et dérivation.

On appelle connexion principale une connexion dont les deux extrémités sont des terminaux de l'équipotentielle. C'est la racine de l'équipotentielle sur laquelle vont se greffer les dérivations. Il y a une connexion principale par équipotentielle.

On appelle dérivation une connexion dont une connait une extrémité placée sur la frontière du canal et dont l'autre extrémité est un point appartenant à la connexion principale de l'équipotentielle. Donc si une équipotentielle comporte N terminaux, le nombre de dérivations est égal à (N-2).

Considérons une équipotentielle comportant N terminaux. Dans le cas simple où N est égal à 2, l'équipotentielle ne pose pas de problème car c'est une connexion principale. Si N est supérieur à 2, il faut choisir parmi les N terminaux, deux terminaux qui constitueront les extrémités de la connexion principale. La règle utilisée pour choisir ces deux terminaux est la suivante: se plaçant sur un point au

milieu de la frontière supérieure du canal, on recherche, de part et d'autre de ce point, le terminal le plus proche appartenant à l'équipotentielle traitée: ce terminal constitue une des extrémités de la connexion principale. Pour la seconde extrémité, on répète la même procédure en se plaçant au milieu de la frontière inférieure.

On réalise de cette façon une sorte de rateau vertical. Cette règle est appliquée lorsque le facteur de forme du canal est inférieur à un; le facteur de forme d'un canal est égal au rapport de sa longueur sur sa hauteur. Dans le cas contraire, le même principe est utilisé en se plaçant au milieu des frontières gauche et droite du canal afin de créer un rateau horizontal.

Cette manière de procéder est un peu semblable aux méthodes classiques: dans un canal, qu'il soit horizontal ou vertical, chaque équipotentielle est définie par une borne inférieure et par une borne supérieure; entre ces deux bornes, on a un segment sur lequel viennent se greffer toutes les dérivations issues de terminaux compris entre les deux bornes (Méthode de Yoshimura-Kuh).

3.3.2 Traitement.

Après avoir réalisé la division de toutes les équipotentielles, on dispose de deux ensembles:

1. un ensemble de connexions principales; son cardinal est égal au nombre d'équipotentielles.

2. un ensemble de dérivations: pour chacune d'entre elles, on connaît une de ses extrémités et l'équipotentielle à laquelle elle appartient.

Ces deux ensembles sont traités successivement.

Traitement des connexions principales.

Chaque connexion principale est formée de deux terminaux. Donc le problème à résoudre est un problème de

"Channel routing" bijectif tel qu'on l'a présenté dans le paragraphe précédent. Il suffit de reprendre les algorithmes et de les appliquer à cet ensemble. Un graphe des régions est défini.

Traitement des dérivations.

On génère successivement toutes les dérivations. De la même manière que pour une connexion principale, on recherche un chemin. Ce chemin ne relie plus deux zones connues à l'avance mais la zone du terminal origine et une zone qui comporte sur son contour un segment appartenant à la même équipotentielle que la dérivation.

Les critères de recherche sont les mêmes que pour le "Channel routing" bijectif. Quant au choix d'un chemin, il se fait essentiellement en fonction du nombre de contacts et en fonction du nombre d'équipotentielles traversées. Ce dernier nombre doit être minimisé. Ce critère n'existe pas pour le "Channel routing" bijectif car le nombre de connexions à traverser est défini par la séquence des terminaux sur le contour du canal.

Après avoir déterminé un chemin pour une dérivation, le graphe des régions est modifié en conséquence.

Passage à la grille.

Une fois toutes les connexions principales et toutes les dérivations traitées, on peut réaliser, de manière identique au "Channel routing" bijectif, le coloriage des régions ainsi que leur dimensionnement à un petit détail près : les extrémités des dérivations peuvent être source de contact; dans le cas où la région qui contient l'extrémité de la dérivation liée à l'équipotentielle est une région fermée, l'extrémité de la dérivation génère un contact.

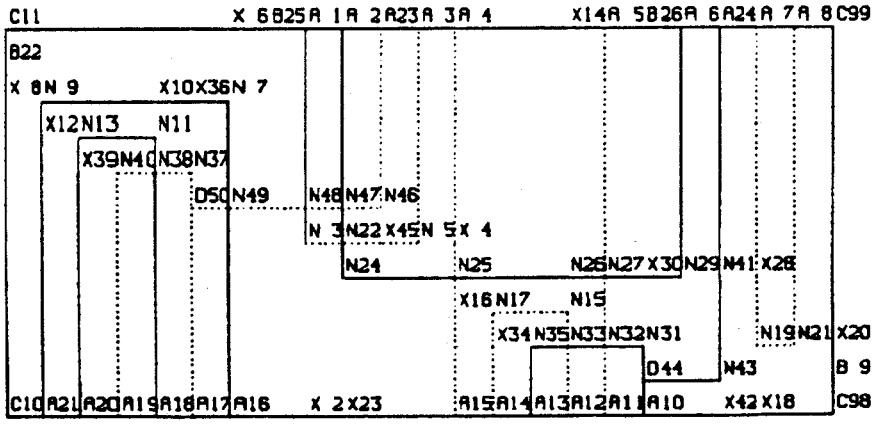


Figure 57. Exemple de "Channel routing " général.

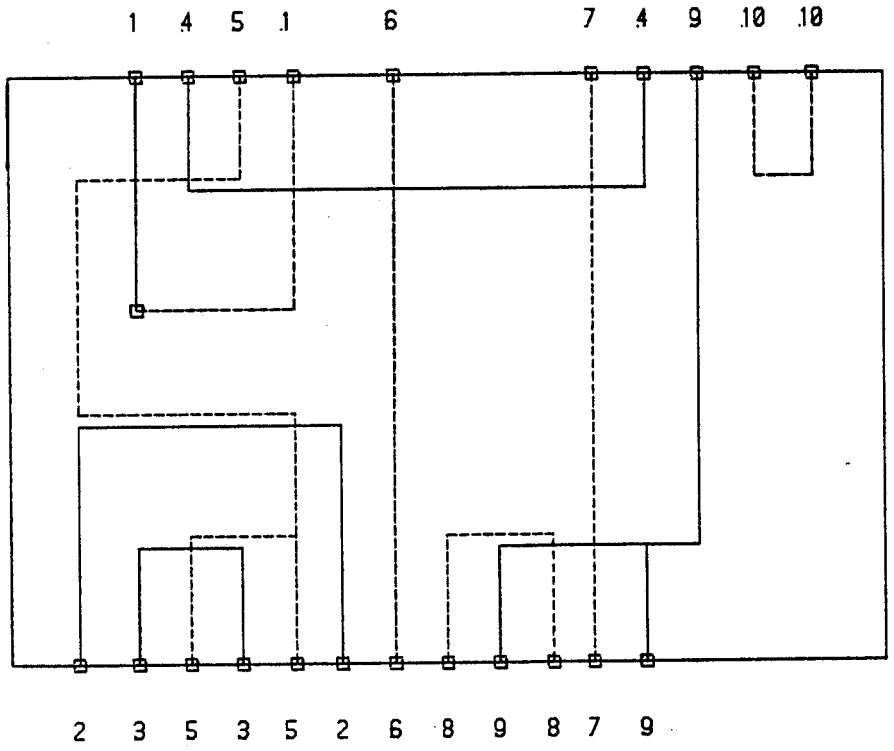


Figure 58. Solution compactée.

3.3.3 Exemple.

Pour appliquer cette méthode, on a repris un problème donné dans le premier chapitre (Figure 8 page 20), constitué de deux séquences de 12 terminaux à interconnecter. Cet exemple comporte deux dérivations: une pour l'équipotentielle 5 et une pour l'équipotentielle 9. Le résultat est donné sur la Figure 57 page 94. Il comporte un seul contact. Ce résultat a été compacté et a subi quelques transformations géométriques pour donner le résultat de la Figure 58 page 94. Par rapport à la solution donnée par Yoshimura, on a une piste de moins que la densité et un nombre de contacts nettement inférieur.

4.0 APPLICATION AU DESSIN DES CELLULES MOS.

Le but recherché dans cette application est de générer automatiquement un dessin symbolique d'une cellule MOS décrite par deux types d'information :

- une description électrique de la cellule. Chaque transistor y est défini par son drain, sa source, sa grille ainsi que sa taille: longueur (L) et largeur (W) du canal.

- une liste des points d'entrée et de sortie de la cellule.

Ce problème est beaucoup plus complexe que celui du "Channel routing". Dans ce dernier, les liaisons à réaliser sont définies au départ: il s'agit de relier, soit deux terminaux, soit un terminal et une équipotentielle. Dans le cas de la cellule, on rencontre un troisième type de liaison: relier deux noeuds électriques qui n'apparaissent pas sur le contour du domaine. Donc, une liste de liaisons à réaliser doit être générée en fonction des données entrées avant d'entamer le traitement proprement dit.

Avant de développer notre méthode, présentons deux méthodes qui s'attaquent également au dessin automatique de cellules MOS.

4.1 METHODES CONCURRENTES.

La première méthode dénommée DUMBO (22) a été développée à l'Université de Stanford, tandis que la seconde, TOPOLOGIZER (23) a été étudiée aux laboratoires Bell. Les deux méthodes prennent les mêmes données en entrée que celles données ci-dessus.

DUMBO.

DUMBO effectue successivement les opérations suivantes:

- o les transistors sont placés dans la cellule par un algorithme de placement analogue à ceux opérant sur les blocs.
- o les transistors sont ensuite orientés.
- o chaque équipotentielle reliant plusieurs terminaux est divisée en branches, chaque branche reliant deux noeuds.
- o pour chaque branche, DUMBO réalise l'interconnexion.

Le système DUMBO est assez souple car il permet à l'utilisateur d'intervenir. Par rapport à un dessin manuel, la surface est environ doublée.

TOPOLOGIZER

TOPOLOGIZER, quant à lui, utilise les mêmes principes que DUMBO: il place les transistors et ensuite il les interconnecte. En fait, TOPOLOGIZER est un système expert auquel on a appris un ensemble de règles inspirées du dessin manuel. Il traite seulement la technologie CMOS.

4.2 ESSAI DE GENERATION DES CONNEXIONS.

La méthode développée dans cette thèse a peu de rapport avec les deux méthodes présentées ci-dessus. La nouvelle approche donnant la possibilité de réaliser des connexions dans le graphe des régions, il faut définir la liste des connexions devant être créées. L'ordre de cette liste est très important car certaines connexions, reliant des points internes, ne pourront être réalisées qu'après la création de leurs extrémités sur des connexions préalablement placées. Il faut donc essayer de décomposer la cellule.

Graphe des transistors.

Pour définir tous les noeuds électriques du circuit, il suffit de considérer:

- les drains ou les sources des transistors,
- les grilles attaquées par des noeuds externes. Les noeuds externes sont des noeuds placés sur le contour de la cellule par opposition aux noeuds internes.

En effet, les autres grilles de transistor sont les sorties, drain ou source, d'autres transistors.

Sachant que toutes les connexions arrivant sur des grilles pourront être traitées sans problème une fois les drains et les sources de tous les transistors placés, on ne considère que les drains et les sources. Les premières connexions à traiter ne feront intervenir que les drains et les sources des transistors.

Considérons le graphe G appelé graphe des transistors. G est défini par:

- E l'ensemble des noeuds du circuit qui sont un drain ou une source d'un transistor.
- V l'ensemble des arcs du graphe. Il y a un arc entre deux noeuds s'il existe un transistor entre ces deux noeuds. L'arc est particularisé par le nom du transistor (il y peut y avoir plusieurs arcs entre deux noeuds).

Sous-graphes.

Le premier traitement effectué sur le graphe des transistors est la détection des rebouclages. Sur la Figure 59 page 100, le transistor M18 est un transistor de rebouclage. Des boucles étant détectées, on supprime les transistors créant ces boucles du graphe des transistors.

Ensuite on extrait du graphe des transistors des sous-graphes de la manière suivante: on se place sur un

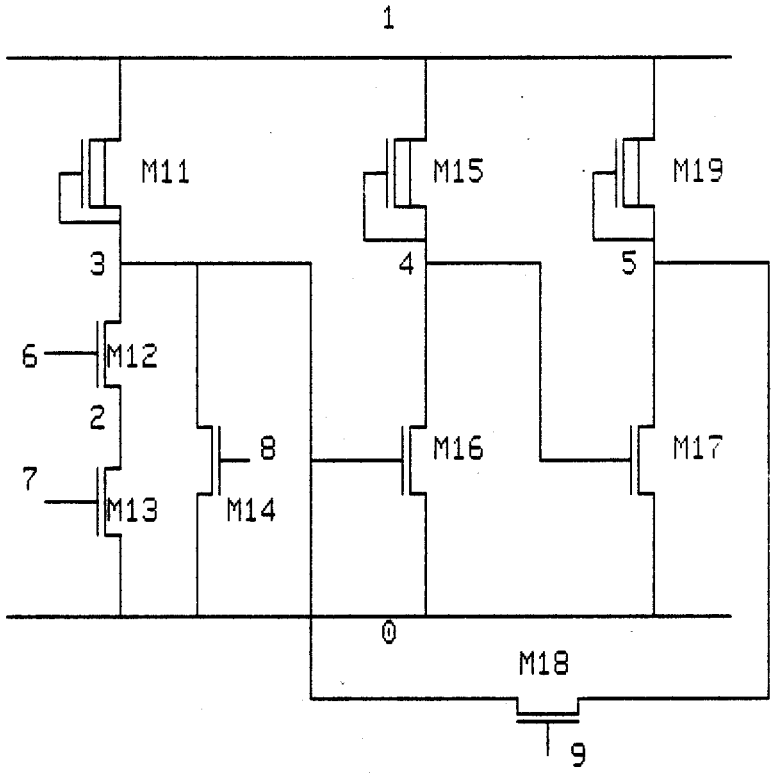


Figure 59. Cellule à traiter.

noeud interne du graphe des transistors; on parcourt toutes les branches du graphe partant de ce point. On s'arrête lorsqu'on ne rencontre plus que des noeuds externes. Le sous-graphe est alors le graphe engendré par l'ensemble des noeuds parcourus. Ce processus est répété jusqu'à ce que tous les noeuds internes aient été traversés une fois. On obtient plusieurs sous-graphes.

Pour préciser ces notions, considérons l'exemple de la Figure 59. Trois sous-graphes peuvent être extraits du graphe des transistors:

1. le sous-graphe 1 constitué des noeuds 0,1,2,3

2. le sous-graphe 2 constitué des noeuds 0,1,4

3. le sous-graphe 3 constitué des noeuds 0,1,5

Chacun des sous-graphes peut être caractérisé par l'ensemble des côtés qu'il touche et par l'ensemble des sous-graphes auxquels il est connecté.

Création des connexions.

Puisque chaque sous-graphe est indépendant des autres sous-graphes, on peut maintenant définir pour chaque sous-graphe les connexions qui devront être réalisées. Par exemple, pour le sous-graphe 1 de l'exemple, on peut considérer qu'il y a deux connexions : une connexion principale du noeud 1 vers le noeud 0 traversant les transistors M11, M12 et M13 et une dérivation, la branche du noeud 0 vers le noeud 3 traversant le transistor 14. La connexion principale sera d'abord traitée puis le noeud 3 sera placé sur cette connexion; il sera alors possible de traiter la dérivation. Les connexions extraites des sous-graphes sont les premières dans la liste des connexions à tracer. Ensuite, on traitera les connexions qui créent des rebouclages et les connexions dont une extrémité est une grille. De cette manière, on peut générer la liste des connexions.

4.3 GENERALITES SUR LE TRAITEMENT.

Une fois la liste ordonnée des connexions déterminée, on peut réaliser le traitement proprement dit. On opère successivement sur les trois ensembles de connexions suivants:

- l'ensemble des connexions faisant partie des sous-graphes,
- l'ensemble des connexions correspondant à des rebouclages,
- l'ensemble des connexions dont une extrémité est une grille de transistor.

La différence par rapport au "Channel routing" est que l'on introduit les niveaux technologiques pendant la recherche topologique. Une raison explique cette démarche: pour chaque transistor, sa source et son drain sont en diffusion, sa grille est en polysilicium. Donc il n'est pas possible de choisir indifféremment un des trois niveaux technologiques pour ces noeuds.

Lorsqu'on trace les connexions de chaque sous-graphe, tout peut se faire sur un seul niveau technologique, c'est-à-dire la diffusion puisqu'il n'y a pas de rebouclage et que les sous-graphes sont indépendants. Cette règle est valable en technologie NMOS. Par contre en technologie CMOS, les diffusions des transistors n et p sont distinctes et il est nécessaire d'utiliser une liaison en métal pour relier les deux diffusions appartenant à un même noeud électrique. Donc pour chaque sous-graphe comprenant un noeud lié à la fois à un transistor p et à un transistor n, il faut rajouter sur la connexion correspondante un noeud permettant de changer de niveau.

Pour toutes les connexions des deux derniers ensembles, on connaît leur origine et leur destination et les niveaux technologiques de ces deux noeuds. Ces noeuds appartiennent chacun à une ou deux régions du graphe. Pour créer la liaison, on recherche un chemin entre la région origine et la région destination en tenant compte des niveaux technologiques. La traversée d'un segment d'une connexion donnée par une autre connexion est possible si:

- les deux connexions ne sont pas sur le même niveau technologique
- le croisement n'entraîne pas la création d'un transistor.

Si la traversée du segment est nécessaire, il faut changer de niveau technologique en rajoutant un point sur la connexion. Le chemin retenu est celui qui entraine le moins de contacts.

Ce rapide aperçu de la génération automatique des cellules montre que les problèmes à régler sont nombreux et difficiles. Cependant, on peut penser que cette approche pourra donner de très bons résultats. L'autre intérêt de cette méthode est de pouvoir générer différentes topologies et donc des cellules de formes différentes pour une même structure logique.

CONCLUSION

La génération du dessin des circuits intégrés par l'approche topologique présentée dans cette thèse permet de traiter les problèmes de dessin en se démarquant des approches classiques: dans ces dernières, deux étapes sont généralement confondues; celle qui définit la topologie des connexions et celle qui définit leur position exacte dans un repère donné. L'approche développée ici les sépare et permet donc de se polariser sur la topologie des connexions, c'est-à-dire leur positionnement relatif les unes par rapport aux autres et de s'affranchir des contraintes liées à la géométrie.

La méthode générale développée peut s'appliquer indifféremment aux problèmes de "Channel routing" et de dessin de cellules MOS. Elle est basée sur le graphe des régions généré par le placement des connexions dans le domaine à traiter. Lorsqu'une connexion est ajoutée ce graphe est modifié en fonction du chemin trouvé reliant les deux extrémités de la connexion. Une fois toutes les connexions traitées, les régions du graphe final sont dimensionnées.

Son application au problème du "Channel routing" a donné des résultats assez bons; l'objectif essentiel a été de minimiser le nombre de contacts. Des problèmes subsistent dans certains cas pour obtenir le facteur de forme désiré. C'est dans son application au dessin automatique de cellules de circuit intégré que l'approche paraît la plus prometteuse. Il devra être possible à partir d'une description électrique d'une cellule MOS de réaliser automatiquement un dessin symbolique de celle-ci.

Cette étude pourrait être poursuivie dans plusieurs directions: utiliser des géométries comprenant des angles à 45 degrés, réaliser l'interconnexion sur trois niveaux technologiques ou plus.

APPENDIX A. RAPPELS SUR LES GRAPHES

Nous donnons dans cette annexe quelques définitions et propriétés concernant les graphes qui sont utilisés dans cette étude. Ces définitions sont reprises entre autres de l'ouvrage de Berge sur les graphes (24). Ensuite, l'algorithme de Gavril permettant de trouver un ensemble indépendant maximum sur un graphe cercle sera exposé.

A.1 DEFINITIONS ET PROPRIETES DES GRAPHES.

A.1.1 Définition d'un graphe.

Un graphe G = (X,V) est le couple constitué par :

1. un ensemble X = (x1,x2,...,xn)

2. une famille V = (v1,v2,...,vn) d'éléments du produit cartésien X x X = ((x,y) / x e X, y e X).

On parle de noeuds ou sommets pour les éléments de X et d'arêtes ou arcs pour les éléments de V. On parle d'arc orienté (x,y) lorsqu'on parle du couple (x,y) du produit cartésien et non pas du couple (y,x).

A.1.2 Sous-graphe engendré par A inclus dans X

Etant donné A inclus dans X, le sous-graphe engendré par A est le graphe Ga dont les sommets sont les éléments de A et dont les arêtes sont les arêtes de G ayant leurs deux extrémités dans A.

A.1.3 Graphe complémentaire.

Etant donné un graphe G = (X,V), le complémentaire H = (X,W) a le même ensemble de sommets que G et comme arêtes les arêtes complémentaires à V : si (i,j) appartient à V (resp. à W), (i,j) n'appartient pas à W (resp. à V).

A.1.4 Chemin dans un graphe.

Une chaîne ou chemin dans un graphe de longueur q est une séquence de q arêtes L = (v1,v2,...,vq) avec

```
v1 = (x0,x1)
v2 = (x1,x2)
.....
vq = (xq-1,xq)
```

Le sommet x0 est l'extrémité initiale du chemin; le sommet xq est l'extrémité terminale du chemin L.

A.1.5 Cycle.

Un cycle est une chaîne (v1,v2,...,vn) telle que:

- le même arête ne figure pas deux fois dans la séquence;
- les deux sommets aux extrémités de la chaîne coïncident.

A.1.6 Graphe connexe.

Un graphe est dit connexe si, pour tout couple de sommets (i,j), il existe un chemin reliant i et j.

A.1.7 Graphe planaire.

On dit qu'un graphe G est planaire s'il est possible de le représenter sur un plan de sorte que les sommets soient des points distincts, les arêtes des courbes simples, et que deux arêtes ne se rencontrent pas en dehors de leurs extrémités. La représentation de G sur un plan conformément aux conditions imposées s'appelle un graphe planaire topologique.

A.1.8 Graphe dual topologique.

Soit G un graphe planaire topologique; une face de G est par définition une région du plan limitée par des arêtes et telle que deux points arbitraires de cette région puissent toujours être reliés par un trait continu ne rencontrant ni arêtes, ni sommets.

Considérons un graphe planaire G, connexe; on lui fait correspondre un graphe planaire G* défini de la manière suivante : à l'intérieur de chaque face s de G, plaçons un sommet s* de G*; à toute arête v de G, on fait correspondre une arête v* de G* qui relie les sommets s1* et s2* correspondant aux faces s1 et s2 qui se trouvent de part et d'autre de l'arête v dans G. Le graphe G* ainsi défini est planaire et connexe. On l'appelle le graphe dual topologique de G.

A.1.9 Sommets adjacents.

Deux sommets x1 et x2 du graphe G = (X,V) sont adjacents s'il existe une arête de V reliant x1 et x2.

A.1.10 Graphe biparti.

Un graphe est biparti si l'ensemble X de ses sommets peut être partitionné en deux classes X1 et X2 de sorte que deux sommets de la même classe ne soient jamais adjacents.

A.1.11 Ensemble indépendant.

Un ensemble A de sommets d'un graphe G = (X,V) (A inclus dans X) est dit indépendant s'il n'existe aucune arête reliant deux de ses éléments.

Un ensemble indépendant maximum est un ensemble indépendant qui comprend le plus grand nombre de sommets parmi l'ensemble des ensembles indépendants du graphe G.

A.1.12 Clique.

Un ensemble A de sommets d'un graphe G = (X,V), A inclus dans X, est une clique si, quel que soit le couple de sommets (i,j) appartenant à A, il existe une arête reliant i et j.

Une clique maximale est une clique qui comprend le plus grand nombre de sommets d'un graphe.

A.1.13 Graphe pondéré.

Un graphe pondéré est un graphe dans lequel chaque sommet i est affecté d'un poids w(i).

On définit également une clique maximale pondérée de la manière suivante : c'est une clique telle que la somme des poids des sommets qui en font partie soit maximale.

On définit également un ensemble indépendant maximal pondéré de la manière suivante : c'est un ensemble indépendant tel que la somme des poids des sommets qui en font partie soit maximale.

A.1.14 Graphe cercle.

Un graphe cercle se définit de la manière suivante : soit un cercle et un ensemble de cordes traversant ce cercle;

- chaque corde correspond à un sommet du graphe;
- il y a une arête entre deux sommets si les deux cordes correspondantes se croisent.

A.1.15 Graphe orienté transitif.

Un graphe orienté G = (X,V) est dit transitif si :

- il ne contient pas de cycle;
- quels que soient les sommets u,v et w tels que :
  - u et v sont adjacents,
  - v et w sont adjacents,

  alors les sommets u et w sont adjacents.

Si X = (x1,x2,.....,xn), on peut renuméroter les sommets 1,2,...n de telle manière que tout arc orienté aille d'un sommet de numéro inférieur vers un sommet de numéro supérieur.

A.2 ALGORITHMES DE GAVRIL.

Le but de l'algorithme développé par Gavril est de trouver un ensemble indépendant maximum dans un graphe cercle.

A.2.1 Clique maximale dans un graphe orienté transitif.

Soit G=(X,V) un graphe orienté transitif avec X=(1,2,...,n). Soit Gi le sous-graphe de G engendré par (1,2..,i) 1<=i<=n. Pour chaque sommet i du graphe, on définit :

- C(i) une clique maximale contenant i dans Gi.
- P(i) le cardinal de C(i) (nombre de sommets dans C(i))
- K(i) l'ensemble des sommets de G tel qu'il existe un arc orienté entre tout élément de cet ensemble et i.

Pour trouver une clique maximale dans un graphe G orienté transitif, la démarche est la suivante :

1. On a C(1) = (1) et P(1) = 1.

2. Supposons C(1),C(2),..,C(i-1) et P(1),P(2),..,P(i-1) connus :

   a. Si K(i) est l'ensemble vide, alors C(i) = (i) et P(i) = 1.

   b. Si K(i) n'est pas l'ensemble vide, l'ensemble K(i) est égal à (i(1),i(2),....,i(k)) avec i(1),i(2),....i(k) < i. Quel que soit i(j) appartenant à K(i), C(i(j)) U (i) est une clique de Gi car Gi est un graphe transitif..

   c. Chaque clique contenant i est obtenue en ajoutant i à une clique de Gi(j). Soit i(r) le sommet tel que : P(i(r)) = Max P(j) avec j appartenant à K(i).

      Alors C(i(r)) U (i) est une clique maximale contenant i dans Gi et P(i) = P(i(r)) + 1.

3. En supposant avoir déterminé C(1),...,C(n) et P(1),..,P(n) par la méthode présentée ci-dessus, la clique maximale C(k) est la clique telle que P(k) = Max P(i) avec 1<=i<=n.

Cette méthode peut s'appliquer à un graphe pondéré en modifiant les définitions de P(i) : P(1)=w(1), P(i)=P(i(r))+w(i).

Si n est le nombre de sommets du graphe G, le nombre de pas de programme est au maximum n*n, chaque pas permettant de déterminer une clique.

A.2.2 Ensemble indépendant d'un graphe des intervalles.

Considérons la Figure 60 page 112, une droite A-B et une famille d'intervalles placés au-dessus de la ligne A-B. Soit G le graphe des intervalles défini ainsi :

- l'ensemble des sommets est l'ensemble F des intervalles.
- deux sommets i et j sont reliés par une arête lorsqu'il y a intersection entre i et j.

Soit G' le graphe complémentaire de G; si une arête i-j relie les sommets i et j, alors les intervalles i et j sont disjoints.

Le graphe G' peut être orienté transitivement par la relation suivante : il y a une arête orientée de i vers j si l'intervalle i est à la gauche de l'intervalle j sur la droite A-B.

Un ensemble de sommets est un ensemble indépendant maximum pour G si et seulement si cet ensemble est une clique pour le graphe complémentaire, donc on peut trouver un ensemble indépendant maximum pour un graphe des intervalles car on sait déterminer une clique maximale pour son complémentaire qui peut être orienté transitivement.

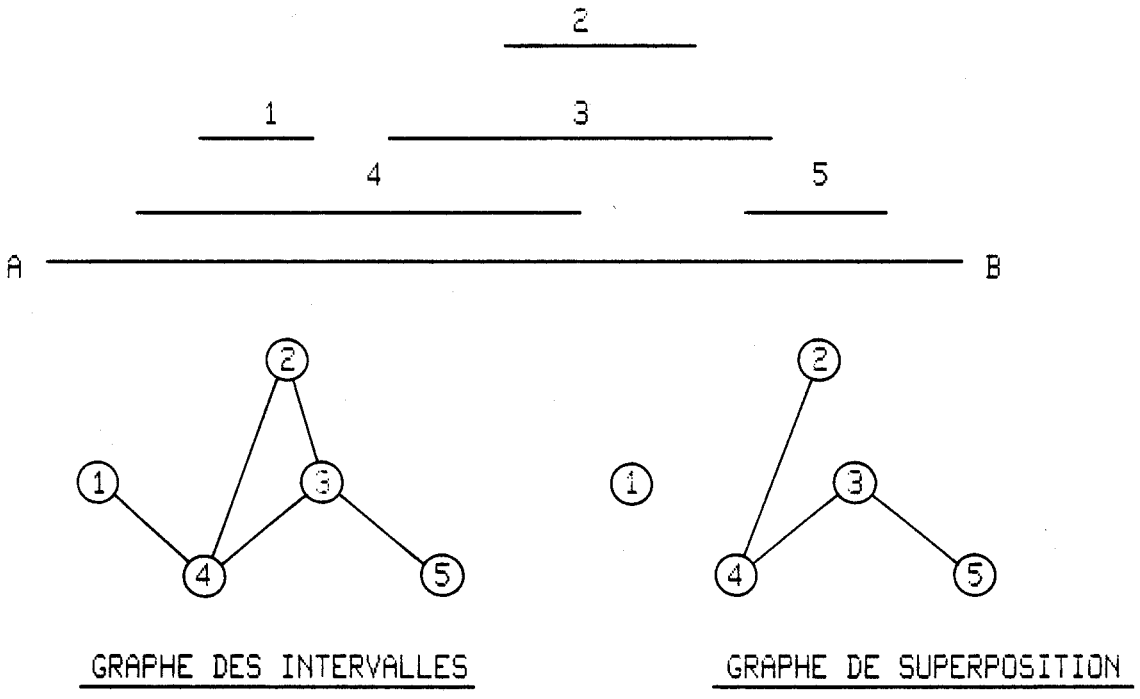


Figure 60. Graphe des intervalles

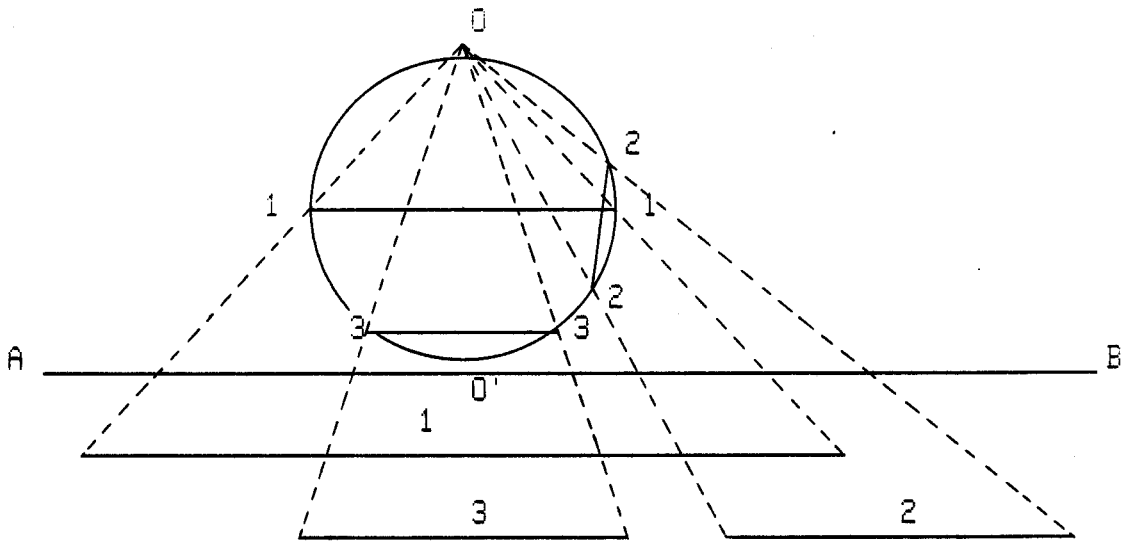


Figure 61. Graphe de superposition et graphe cercle

A.2.3 Graphe de superposition et graphe cercle.

Considérons une famille d'intervalles placés sur une droite A-B (Figure 60 page 112). On dit que deux intervalles i et j se superposent s'il y a intersection entre i et j et s'il n'y a pas inclusion de i dans j, ni de j dans i. On peut alors définir un graphe de superposition :

- les sommets du graphe sont les intervalles.
- deux sommets i et j sont reliés si i et j se superposent.

Ce graphe est différent du graphe des intervalles car dans ce dernier si un intervalle est inclus dans un autre, il y a une arête reliant les deux sommets correspondant à ces intervalles.

Considérons un graphe cercle et une famille de cordes traversant ce cercle (Figure 61 page 112). Soit O et O' deux points opposés du cercle n'étant pas des extrémités d'une corde et soit A-B la tangente au cercle en O'.

Si l'on projette les cordes sur la droite A-B par la projection de centre O, on obtient une famille d'intervalles telle que deux intervalles se superposent si et seulement si les cordes correspondantes se croisent.

Donc un graphe cercle défini par une famille de cordes est isomorphe à un graphe de superposition défini par la famille des intervalles projections de ces cordes sur une tangente au cercle.

Trouver un ensemble indépendant maximum pour un graphe cercle revient à trouver un ensemble indépendant maximum pour le graphe de superposition correspondant. La méthode de Gavril consiste à essayer de se rapporter à des graphes des intervalles pour lesquels on sait trouver un ensemble indépendant maximum.

A.2.4 Ensemble indépendant maximum pour un graphe de superposition.

Nous prendrons les notations suivantes :

- P(G) le cardinal d'un ensemble indépendant maximum pour le graphe G.
- Pw(G) la somme des poids des sommets faisant partie d'un ensemble indépendant maximum dans le cas où le graphe G est pondéré.
- G' le graphe des intervalles correspondant au graphe de superposition G défini sur l'ensemble d'intervalles F.
- v' est l'intervalle correspondant au sommet v dans le graphe de superposition.

Nous donnons ici l'énoncé du lemme démontré par Gavril permettant de trouver un ensemble indépendant maximum pour un graphe de superposition :

- Soit G un graphe de superposition et F la famille d'intervalles correspondante.
- Soit Uv = (u' / u' est inclus dans v');
- Soit Gv le graphe de superposition défini sur Uv;
- Soit w(v) le poids affecté à chaque sommet v de G' avec w(v)=P(Gv)+1;
- Soit (v(1),v(2),.....,v(r)) un ensemble indépendant de poids maximum dans G';
- Soit Dj un ensemble indépendant maximum de Gv(j) pour tout j compris entre 1 et r.

Alors P(G) = Pw(G') et

$$\bigcup_{j=1}^{r} (Dj \cup (vj))$$ est un ensemble indépendant maximum de G.

Pour la démonstration de ce lemme, on pourra se reporter à l'article de Gavril (18).

A.2.5 Application du lemme.

Gavril a décrit une méthode pour affecter à chaque sommet v (intervalle) du graphe de superposition un poids w(v) et un ensemble Dv tels que :

- Dv est un ensemble maximum indépendant de Gv,
- w(v) = P(Gv) + 1.

Définissons une famille de sous-ensembles de F de la manière suivante :

- A(1) = (u'/ il n'existe pas v' de F inclus dans u')
- A(2) = (u'/u' n'appartient pas à B(2) et si v' est inclus dans u', alors v' appartient à B(2))
- .
- .
- A(k) = (u'/u' n'appartient pas à B(k) et si v' est inclus dans u', alors v' appartient à B(k))

B(i) = UA(j) pour j variant de 1 à i-1 et F = UA(i) pour i variant de 1 à k.

Un intervalle u' appartient à un ensemble A(i) si et seulement si :

- il contient au moins un intervalle de A(i-1).
- tous les intervalles inclus dans u' sont des éléments de la réunion des A(i) pour i variant de 1 à i-1.

Donc si deux intervalles u' et v' appartiennent à un même sous-ensemble A(i), soit u' et v' se superposent, soit u' et v' sont disjoints.

On applique alors la méthode suivante:

1. Quel que soit v appartenant à A(1), Gv et donc Dv sont des ensembles vides ; il s'ensuit que w(v)=1.

2. Supposons que pour tous les u, tels que u' appartienne à la réunion des A(j) (j variant de 1 à i-1), on leur ait affecté :

a. un poids w(u) = P(Gu) + 1,
b. un ensemble indépendant maximum Du dans le graphe Gu.

Soit v tel que v' appartient à A(i) et Gv le sous-graphe de superposition défini à partir de Uv. On peut donc dire que :

a. Quel que soit u appartenant à Gv, l'intervalle u' correspondant est inclus dans v' et appartient à la réunion des A(i) (i variant de 1 à i-1). Donc on a affecté à u un poids w(u) et un ensemble indépendant maximum Du.

b. Soit Gv' le graphe des intervalles défini sur la famille d'intervalles Uv. On affecte à chacun des éléments u' de Gv' le poids w(u). Puisque Gv' est un graphe des intervalles, on peut trouver un ensemble indépendant de poids maximum (u(1),u(2),..,u(t)).

c. On peut appliquer le lemme énoncé précédemment au graphe de superposition Gv; on a :

$$P(Gv) = Pw(Gv') \text{ et}$$

$$Dv = \bigcup_{j=1}^{t} (Du(j) \cup (u(j)))$$

est un ensemble indépendant maximum de Gv.

De cette manière, on peut assigner à tout sommet v de A(i) un poids w(v) = P(Gv) + 1 et un ensemble Dv. On applique cette méthode successivement à tous les ensembles A(j).

3. A chaque sommet de G, graphe de superposition, on a donc affecté un poids w(v) et un ensemble Dv. On affecte à chacun des éléments de G' le poids w(v). Puisque G' est un graphe des intervalles, on peut trouver un ensemble maximum indépendant (v(1),v(2),...,v(r)) dans G'. On applique le lemme à G et l'on obtient :

$$P(G) = Pw(G') \text{ et}$$

r
U (Dv(j)U(v(j)) est un ensemble indépendant maximum de G.
j=1

Si n est le nombre de sommets du graphe, alors le nombre de pas de programme est au plus n*n*n puisque pour trouver un ensemble indépendant maximum il faut au plus n*n pas (en fait pour trouver une clique maximale dans un graphe orienté transitif).

A.2.6 Exemple

Reprenons l'exemple présenté dans le chapitre 3. Les extrémités des cordes sont placées sur le graphe cercle dans l'ordre suivant (Figure 62 page 118) : 7 5 15 1 11 3 12 6 8 1 2 3 4 5 6 7 8 9 10 11 2 12 13 14 13 14 10 4 9 15.

Les ensembles A(i) sont égaux à :

1. A(1) = (1 2 3 6 8 13 14)

2. A(2) = (5 10 11 12)

3. A(3) = (4 7 9)

4. A(4) = (15)

Dans un premier temps, on détermine pour chaque sommet v un poids w(v) et un ensemble indépendant maximum Dv dans Gv qui est le graphe de superposition défini à partir de Uv avec Uv = (u' / u' est inclus dans v'). On traite successivement les ensembles A(j) :

1. Traitement de A(1).

   Pour tout élément v de A(1), Uv est l'ensemble vide; il en est donc de même pour Gv et Dv. Et w(v) = P(Gv) + 1 = 0 + 1 = 1. On en déduit :

   o w(1)=w(2)=w(3)=w(6)=w(8)=w(13)=w(14) = 1

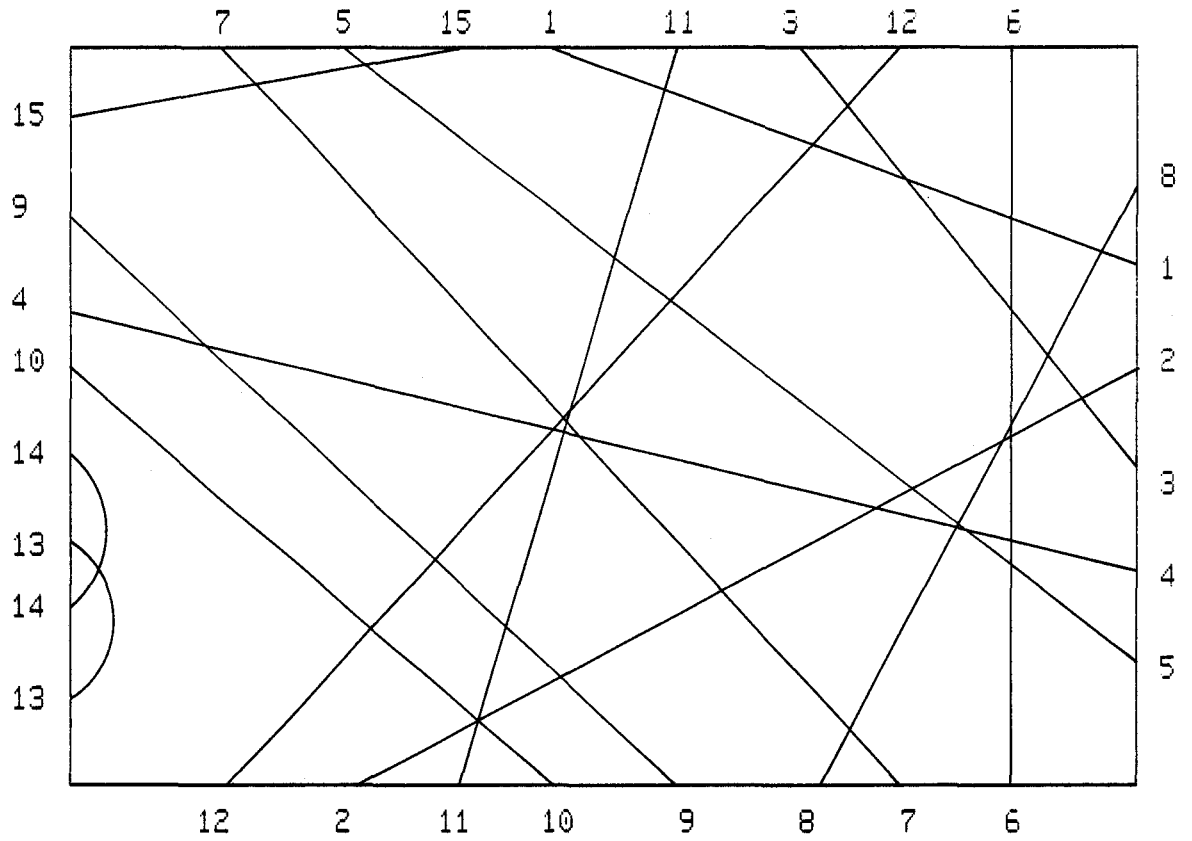


Figure 62. Exemple de graphe cercle à traiter.

- D(1),D(2),D(3),D(6),D(8),D(13) et D(14) sont des ensembles vides.

2. Traitement de A(2).

- Etude du sommet 5 G5 = (1,3). Un ensemble indépendant pondéré maximum de G5' est formé uniquement du sommet 1. Donc D(5)= D(1) U (1) = (1) et w(5)=Pw(G1)+1=2.

- Etude du sommet 10 G10 = (13,14). Un ensemble indépendant pondéré maximum de G10' est formé uniquement du sommet 13. Donc D(10)= D(13) U (13) = (13) et w(10)=Pw(G13)+1=2.

- Etude du sommet 11 G11 = (3,6,8). Un ensemble indépendant pondéré maximum de G11' est formé uniquement du sommet 3. Donc D(11)= D(3) U (3) = (3) et w(11)=Pw(G3)+1=2.

- Etude du sommet 12 G12 = (2,6,8). Un ensemble indépendant pondéré maximum de G12' est formé uniquement du sommet 2. Donc D(12)= D(2) U (2) = (2) et w(12)=Pw(G2)+1=2.

3. Traitement de A(3).

   - Etude du sommet 4 G4 = (10,13,14). Un ensemble indépendant pondéré maximum de G4' est formé du sommet 10. Donc D(4)= D(10) U (10) = (13) U (10) = (10,13). w(4)=Pw(G4)+1=2 + 1 = 3.

   - Etude du sommet 7 G7 = (1,3,5,6). Un ensemble indépendant pondéré maximum de G7' est formé du sommet 5. Donc D(7)= D(5) U 5 = (1) U (5) =(1,5). w(7)=Pw(G7)+1=2+1 = 3.

   - Etude du sommet 9 G9 = (10,13,14). Un ensemble indépendant pondéré maximum de G9' est formé du sommet 10. Donc D(9)= D(10) U (10) = (13) U (10) = (10,13). w(9)=Pw(G9)+1=2 + 1 = 3.

4. Traitement de A(4). Cet ensemble ne comprend que le sommet 15 G15 = (1,2,3,4,6,8,9,10,11,12,13,14). Un ensemble indépendant pondéré maximum de G15' est formé des sommets 1 et 4. Donc D(15) = D(1) U (1) U D(4) U (4) = (1) U (10,13) U (4) = (1,4,10,13). Et w(15)=Pw(G15) + 1 = 5.

A chaque sommet de G, graphe de superposition, on a donc affecté un poids w(v) et un ensemble Dv. Pour trouver l'ensemble indépendant maximum dans G, on détermine l'ensemble indépendant pondéré maximum dans le graphe des intervalles G'. Cet ensemble est égal à (7,9). On applique le lemme et l'on obtient comme ensemble indépendant maximum D pour le graphe G :

- D = D(7) U (7) U D(9) U (9) = (1,5) U (7) U (10,13) U (9) soit D = (1,5,7,9,10,13).

- w(v) = 3 + 3 =6.

APPENDIX B. IMPLANTATION INFORMATIQUE.

Nous présentons dans cette annexe un rapide aperçu des programmes informatiques réalisés : après avoir décrit les structures de données utilisées, on précisera une opération importante affectant la structure de données : la modification d'un noeud du graphe des régions.

B.1 LE LANGAGE DE PROGRAMMATION.

Les algorithmes présentés dans cette thèse ont été implantés sur un ordinateur IBM 4381 fonctionnant sous le système d'exploitation VM/SP. Le langage de programmation est le Pascal; il a été choisi pour deux raisons :

1. le Pascal est le langage couramment utilisé pour les programmes de conception assistée par ordinateur des circuits intégrés car d'une part c'est un langage très structuré et d'autre part il est très facile d'emploi.

2. le Pascal permet de faire de la programmation dynamique ce qui facilite le traitement des graphes qui sont très utilisés dans les algorithmes développés ici.

Certes les langages à structure de liste du genre LISP, PROLOG seraient mieux adaptés pour ce type de traitement mais à cause de leur manque de lisibilité, de leur programmation plus difficile, on a préféré utiliser le langage Pascal.

B.2 LA STRUCTURE DE DONNEES.

La structure de données comprend essentiellement le graphe des régions; en effet, toutes les opérations effectuées dans les différents algorithmes concernent ce graphe:

- la recherche d'un chemin pour une connexion donnée se fait dans le graphe des régions.

- créer une connexion revient à modifier le graphe des régions.

- le dimensionnement des régions se fait uniquement en fonction des données du graphe.

Pour décrire le graphe des régions, on utilise des variables dynamiques; ces variables sont créées pendant l'exécution du programme et sont référencées par des variables pointeurs. Cette allocation dynamique permet de n'utiliser que la mémoire nécessaire au traitement mais elle nécessite une programmation plus complexe : il faut s'allouer de la mémoire quand on en a besoin et la rendre au système lorsque l'on ne s'en sert plus afin de ne pas dépasser la capacité de la mémoire virtuelle qui est allouée. Cette allocation dynamique est une des facilités offerte par le langage Pascal.

Structure générale

Un graphe est défini par ses sommets et les arêtes reliant les sommets entre eux. L'ensemble des sommets du graphe est représenté par une liste d'enregistrements correspondant chacun à une région. Chaque enregistrement comporte plusieurs informations:

1. la liste des points formant le contour de la région. Pour chaque région, ce contour est orienté suivant le sens de rotation trigonométrique.

2. la liste des contiguïtés. Cette liste permet de définir les arêtes du graphe.

3. le type de cette région; on distingue quatre types de régions :

   a. les régions touchant la frontière du domaine,
   b. les régions fermées comportant trois connexions différentes sur leur contour,
   c. les régions fermées en comportant quatre,

d. les régions non fermées qui ne touchent pas la frontière du domaine.

Informations liées à un point.

A chacun des points du contour, on associe également un enregistrement qui comprend les informations définissant ce point :

- un identificateur, c'est le nom donné à ce point par le programme.
- une variable indiquant si ce point est angle ou non.
- le numéro de la connexion à laquelle appartient ce point.
- le niveau de ce point. Pendant la recherche topologique, un point peut soit appartenir au contour extérieur du domaine, soit être transparent, soit appartenir à une connexion. Lors du passage à la grille, on assigne les points appartenant à une connexion à un niveau technologique, ceci dans le cas où l'on traite le problème du "Channel Routing".

Informations liées à une contiguïté.

De même, chaque contiguïté est définie par un enregistrement qui contient les informations suivantes :

- un pointeur vers l'enregistrement correspondant à la région contiguë en question. Ce pointeur est l'adresse de cet enregistrement.
- le segment formé de deux points du contour qui sépare les deux régions contiguës. Ce segment est orienté de la même manière que le contour de la région.
- le numéro de la connexion à laquelle appartient ce segment. Sa valeur est égale à zéro dans le cas où le segment est transparent.

De cette manière, chaque arête du graphe des régions est définie avec précision.

La Figure 63 page 125 présente la structure générale des données.

B.3 TRAITEMENT D'UN NOEUD DU GRAPHE DES REGIONS

Sachant que l'opération élémentaire appliquée à un noeud du graphe des régions est la modification de ce graphe suite à la traversée de la région par une connexion, les données passées à la procédure qui réalise l'opération élémentaire sont les suivantes:

- un pointeur vers la région à modifier, c'est-à-dire l'adresse de l'enregistrement correspondant à cette région,
- le point d'entrée de la connexion dans la zone,
- le segment qui devra être traversé pour sortir de cette région,
- le numéro de la connexion traitée.

La procédure modifie le graphe en fonction de ces données et renvoie plusieurs informations et, en particulier le noeud de sortie.

La procédure comporte deux phases : l'initialisation et le traitement proprement dit. Pour chacune, on énoncera les différentes opérations effectuées.

B.3.1 Initialisation.

Elle comporte plusieurs actions:

1. Ajouter le point d'entrée sur le contour de la zone si cette zone n'est pas la zone origine de la connexion.

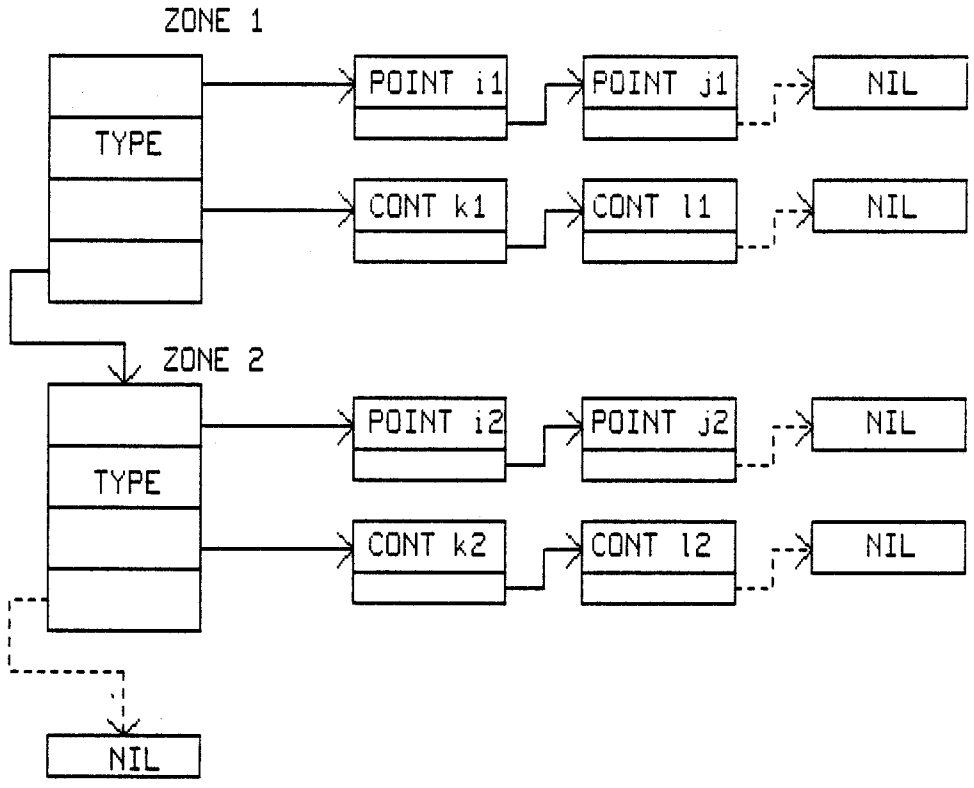


Figure 63. Structure générale des données.

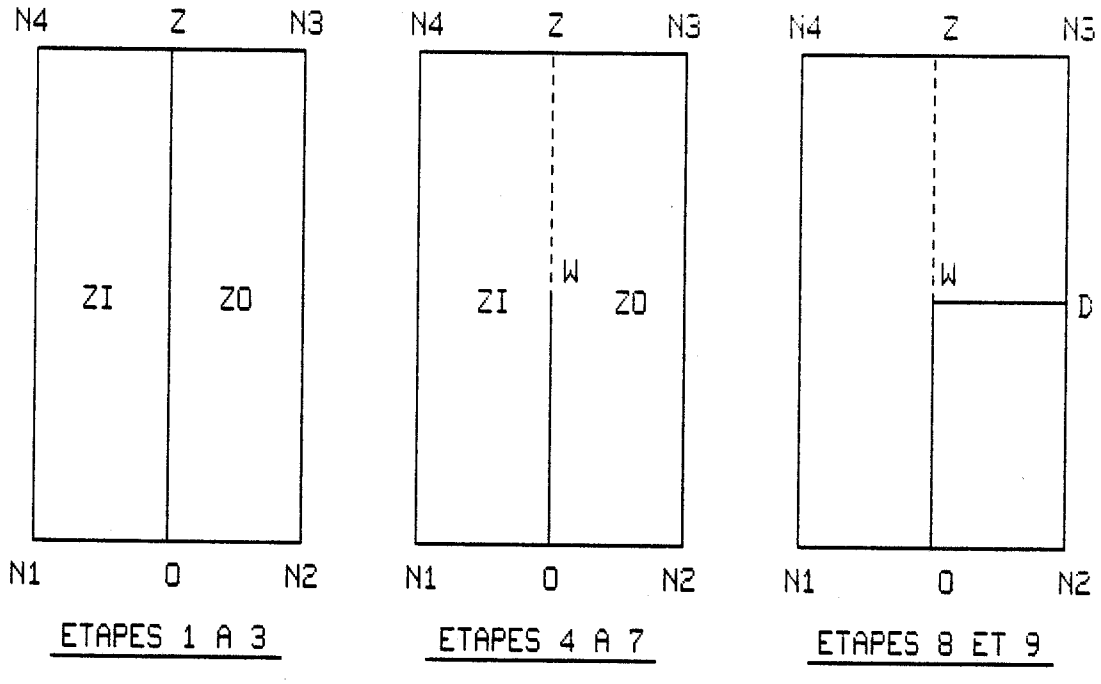


Figure 64. Traitement d'une région.

2. Déterminer la position du point de sortie par rapport au point d'entrée dans la région. On définit une variable DIF qui est égale au nombre de points angles séparant le point d'entrée et le point de sortie. La variable DIF peut prendre les valeurs 0 à 4; suivant la valeur trouvée, un traitement approprié est appliqué :

   a. si DIF = 2 la traversée est directe

   b. si DIF = 1 ou 3, la traversée comporte un angle droit; le traitement est similaire pour ces deux valeurs.

   c. si DIF = 0 ou 4, la traversée comporte deux angles droits.

B.3.2 Traitement.

On présentera le cas où la variable DIF prend la valeur 1. Cet exemple permet d'expliquer la notion de segment transparent. Le contour de la région est la suite de points (N1 O N2 N3 N4). Le point d'entrée est noté O, le segment de sortie est N2-N3. La méthode comporte neuf étapes qui sont résumées sur la Figure 64 page 125 :

1. modifier le contour en rajoutant le point Z; le point Z est placé après le point N3 dans le contour. Le point Z est un point angle.

2. ajouter le point Z sur le contour de la zone contiguë à la région traitée suivant le segment N3-N4. Ce point n'est pas un point angle pour la région contiguë.

3. diviser la région en deux régions suivant le segment O-Z. Cela se traduit au niveau du graphe par les modifications suivantes :

   - créer un nouveau noeud appelé ZI dont le contour est (O Z N4 N1),

   - modifier le contour de l'ancien noeud ZO qui devient (Z O N2 N3),

- mettre à jour les contiguïtés de l'ancienne zone ZO,

- créer une contiguïté entre ZO et ZI.

4. Modifier la zone ZI en rajoutant le point W sur le contour de ZI après le point O. W n'est pas un point angle.

5. Modifier la zone ZO en rajoutant le point W sur le contour de ZO après le point Z. W est un point angle.

6. Transformer la contiguïté entre ZO et ZI en tenant compte du point W qui a été rajouté dans les deux zones.

7. Dans ZO, rendre le point Z transparent;
   Dans ZI, rendre le point W transparent.

8. modifier le contour de ZO en rajoutant le point d'arrivée O entre les deux points du segment de sortie si on n'est pas dans la zone d'arrivée de la connexion.

9. diviser la zone ZO suivant le segment W-O.

le traitement, dans le cas où DIF est égal à 3, est similaire puisqu'on crée également un seul angle droit. Par contre si DIF est égal à 0 ou à 4, les sept premières étapes sont identiques; ensuite on applique la même méthode que lorsque DIF est égal à 1 ou à 3 avec les caractéristiques d'entrée suivantes:

- la zone traitée est ZO,
- le point d'entrée est W, le segment de sortie O-N1.

BIBLIOGRAPHIE

(1) J. D. Williams, "STICKS - A graphical Compiler for High-Level LSI Design", AFIPS Conference Proceedings, Vol. 47, JUNE 1978, p. 289-295.

(2) M. Y. Hsueh, "Symbolic Layout Compaction",Computer Design Aids for VLSI Circuits,édité par Antognetti, Pederson et De Man, Sijthoff and Noordhoff, 1981, p. 499-541.

(3) H. Beke, W. M. C. Sansen et R. Van Overstraeten, "CALMOS: A Computer-Aided Layout Program for MOS/LSI", IEEE Journal of Solid-State Circuits, vol. SC-12, no. 3, Juin 1977, p. 281-282.

(4) G. Persky, D.N. Deutsch et D.G. Schweikert, "LTX - A Minicomputer-Based System for Automated LSI Layout", Journal of Design Automation and Fault Tolerant Computing, vol. 1, no. 3, May 1977, p. 217-255.

(5) D. L. Johansen, "Silicon Compilation",Technical Report no. 4530 - Departement of Computer Science, California Institute of Technology, Pasadena, 1981.

(6) M. A. Breuer "Min-Cut Placement", Journal of Design Automation and Fault Tolerant Computing, vol. 1, no. 4, Octobre 1977. p. 343-362

(7) D. G. Schweikert, "A 2-dimensionnal placement algorithm for the layout of electrical circuits",Proceedings Design Automation Conference(San Fransisco),1976, p.408-416.

(8) D. W. Jepsen et C. D. Gelatt, "Macro placement by Monte Carlo Annealing", International Conference on Computer Design: VLSI in Computers, New-York, 1983, p.495-498.

(9) E. A. Slutz, "Shape Determination and Placement Algorithms For Hierarchical Integrated Circuit Layout", Stanford Electronics Laboratories, Technical Report no. 199, Aout 1980.

(10) C. Y. Lee, "An algorithm for path connections and its applications", IEEE Transactions on Electronic Computers, Sept. 1961, p. 346-365.

(11) D. Hightower, "A solution to the line routing problem on the continuous plane", Proceedings 6th Design Automation Workshop, June 1969, p. 1-24.

(12) A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignement within large apertures", Proceedings 8th Design Automation Workshop, 1971, p.155-169.

(13) D. N. Deutsch, "A "DOGLEG" channel router", Proceedings 13th Design Automation Conference, Juin 1976, p. 425-433.

(14) T. Yoshimura and E. Kuh, "Efficient algorithms for channel routing", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 1, no. 1, Janvier 1982.

(15) Chi-Ping Hsu, "Minimum-Via Topological Routing", IEEE Transactions on Computer-Aided Design of Integrated Circuits, Octobre 1983, vol. 2, no. 4 p. 235-246.

(16) Chi-Ping Hsu, "Minimum-Via Two-Layer Two-Dimensionnal Routing", IEEE International Conference on Computer-Aided Design, 12-15 Sept. 1983., p. 119-120.

(17) M. R. Garey, D. S. Johnson and L. Stockmeyer, "Some Simplified NP-Complete graph problems", Theoretical Computer Science, 1976, vol. 1, p. 237-267.

(18) F. Gavril, "Algorithms for a Maximum Clique and a Maximum Independent Set of a Circle Graph", Networks, vol. 3, p. 262-273, 1973.

(19) M. Marek-Sadowska, "An unconstrained Topological Via Minimization Problem for Two-Layer Routing", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 3, no. 3, Juillet 1984, p. 184-190.

(20) Ron Y. Pinter, "River Routing: Methodology and Analysis", Third Caltech Conference on VLSI, Pasadena, 21-23 mars 1983, p 141-163.

(21) Chi-Ping Hsu, "General River Routing Algorithm", 20th Design Automation Conference, Miami Beach, 27-29 juin 1983, p. 578-583.

(22) W. Wolf, J. Newkirk, R. Mathews and R. Dutton, "Dumbo, A Schematic-to-Layout-Compiler", Third Caltech Conference on VLSI, Pasadena, 21-23 mars 1983, p. 379-393.

(23) P. W. Kollaritsch and N. H. E. Weste, "TOPOLOGIZER: An Expert System Translator Of Transistor Connectivity to Symbolic Cell Layout", ESSIRC, Septembre 1984, Edimbourg, p.175-178.

(24) C. Berge, "Graphes", Editions Gauthier-Villars, 1983.

RESUME

Une nouvelle approche du dessin des circuits intégrés est présentée dans cette thèse. L'étude aborde principalement le problème de l'interconnexion d'un canal séparant des blocs.

Le traitement comporte deux étapes. Dans la première, la topologie des connexions est définie à l'aide d'un graphe des régions, le passage des connexions divisant le domaine initial en régions; l'objectif de la recherche topologique est la minimisation du nombre de contacts et de la surface du canal. La seconde étape génère à partir de ce graphe un dessin symbolique des connexions en dimensionnant les régions du graphe.

Le cas bijectif, chaque connexion relie deux terminaux, et le cas général sont traités. Une extension de la méthode au dessin des cellules MOS est présentée.

MOTS CLES

Circuit intégré, conception assistée par ordinateur, canal d'interconnexion, approche topologique, dessin symbolique.